

CONOCIENDO EL

DB2 Express-C

Un libro de la comunidad para la comunidad

RAUL CHONG, IAN HAKES, RAV AHUJA

PROLOGO POR DR. ARVIND KRISHNA



SEGUNDA EDICION

Segunda Edición (Abril 2008)

**Esta edición ha sido actualizada para IBM® DB2® Express-C Version 9.5 for Linux®,
UNIX® and Windows®.**

© Copyright IBM Corporation, 2007, 2008. Todos los derechos son reservados.

Tabla de Contenido

Sobre este libro	8
Avisos y Marca Registrada.....	8
A quién se le recomienda leer este libro?.....	9
Cómo está el libro estructurado?.....	9
Un libro para la comunidad.....	9
Colaboradores.....	10
Agradecimientos.....	11
Prólogo.....	12
PARTE I – VISION GENERAL Y CONFIGURACION	13
Capítulo 1 – Qué es DB2 Express-C?	15
1.1 Libre para desarrollar, implementar y distribuir... ¡sin límites!	15
1.2 Asistencia al usuario y soporte técnico.....	16
1.3 Servidores DB2	16
1.4 Clientes DB2 y controladores	17
1.5 Libertad en el desarrollo de aplicaciones.....	18
1.6 Versiones DB2 contra Ediciones DB2	19
1.7 Moviéndose a otra edición DB2.....	19
1.8 Mantenimiento de DB2 Express-C	20
1.9 Software Libre relacionado	20
1.9.1 IBM Data Studio	21
1.9.2 DB2 9.5 Net Search Extender	21
1.9.3 Starter Toolkit for DB2 on Rails	21
1.9.4 Web 2.0 Starter Toolkit for DB2.....	21
1.9.5 WebSphere Application Server – Community Edition	21
Capítulo 2 – Características y productos relacionados	23
2.1 Características incluidas con la licencia de suscripción de DB2 Express-C	26
2.1.1 Fix Packs.....	26
2.1.2 Alta disponibilidad de Recuperación Crítica (HADR)	26
2.1.3 Réplica de Datos	27
2.2 Características no disponibles con DB2 Express-C.....	28
2.2.1 Particionamiento de la Base de Datos.....	28
2.2.2 Concentrador de conexión.....	29
2.2.3 Geodetic Extender.....	29
2.2.4 Workload Management (WLM).....	29
2.3 Productos pagados relacionados con DB2.....	29
2.3.1 DB2 Connect.....	29
2.3.2 WebSphere Federation Server	30
2.3.3 WebSphere Replication Server	31
Capítulo 3 – Instalación de DB2	33
3.1 Requisitos para la instalación	33
3.2 Permisos en el Sistema Operativo para la instalación	33
3.3 Instalación con ayuda gráfica guiada (Wizards)	34
3.4 Silent Install (Instalación silenciosa)	38
QuickLab #1: Instala DB2 Express-C & crea la base datos SAMPLE.....	40

Capítulo 4 – El Ambiente DB2	43
QuickLab #2 – Crear una nueva base de datos	54
4.1.1 Variables de ambiente.....	56
4.1.2 Archivo de configuración del manejador de base de datos (dbm cfg).....	56
4.1.3 Archivo de configuración de la base de datos (db cfg)	58
4.1.4 Registro de perfil DB2	59
4.2 El Servidor de Administración DB2 (DB2 Administration Server).....	60
QuickLab #3 – Trabajando con instancias, bases de datos y configuración	62
Capítulo 5 – Herramientas de DB2	65
5.1 Centro de Control (Control Center).....	67
5.2 El Editor de Comandos DB2.....	70
5.3 SQL Assist Wizard.....	72
5.4 El botón Show SQL	74
QuickLab #4: Poblando la base de datos EXPRESS usando scripts.....	76
5.5 Uso de guiones (Scripting)	77
5.5.1 Scripts SQL	77
5.5.2 Scripts de Sistema Operativo o shell scripts.....	78
QuickLab #5: Crear un script de instalación para la Base de datos EXPRESS	80
5.6 Centro de Tareas (Task Center).....	83
5.6.1 La base de datos de Catalogo de la herramienta	83
5.7 Reporte (Journal)	84
5.8 Monitor de Salud (Health Monitor).....	86
5.8.1 Centro de Salud (Health Center)	87
PARTE II – APRENDIENDO DB2: ADMINISTRACION DE BASE DE DATOS.....	90
Capítulo 6 - La Arquitectura de DB2	91
6.1 El Modelo de Procesos.....	91
6.2 El Modelo de Memoria.....	94
6.3 El Modelo de Almacenamiento	95
6.3.1 Páginas y Extensiones	95
6.3.2 Buffer pools	95
6.3.3 Espacios de Tabla (table spaces).....	98
Capítulo 7 – Conectividad desde un cliente DB2.....	103
7.1 Directorios de DB2	103
7.2 El asistente de Configuración (Configuration Assistant)	104
7.2.1 La configuración requerida en el Servidor	104
7.2.2 Configuración requerida en el cliente	107
7.2.3 Creación del perfil del Servidor y el Cliente	112
QuickLab #6: Usando el Asistente de Configuración.....	116
Capítulo 8 – Trabajando con Objetos de la Base de Datos	120
8.1 Esquema	120
8.2 Tablas	121
8.2.1 Tipos de datos	121
8.2.2 Columnas de Identidad.....	124
8.2.3 Objetos Secuencia (SEQUENCE)	124

8.2.4 Tablas del catálogo del sistema	125
8.2.5 Tablas temporales declaradas.....	125
QuickLab #7: Creando una nueva tabla	127
8.3 Vistas	130
8.4 Indices	130
8.4.1 Consejero de diseño.....	130
8.5 Integridad referencial.....	132
Capítulo 9 – Utilidades para el movimiento de datos	135
9.1 La utilidad EXPORT	136
9.2 La utilidad IMPORT	137
9.3 La utilidad LOAD	138
9.4 La utilidad db2move	139
9.5 La utilidad db2look.....	140
Quicklab #8 – Extrayendo DDL para la base de datos EXPRESS.....	142
Capítulo 10 – Seguridad en la Base de Datos	145
10.1 Autenticación.....	146
10.2 Autorización.....	147
10.3 Autoridad DBADM	150
10.4 El grupo PUBLIC	150
10.5 Las sentencias GRANT y REVOKE.....	151
10.6 Chequeo de Autorizaciones y privilegios	151
10.7 Consideraciones sobre privilegios del grupo	153
Quicklab #9 – Otorgando y quitando permisos al usuario.....	154
Capítulo 11 – Backup y Recupero	157
11.1 Registro de cambios o Logging de la Base de datos.....	157
11.2 Tipos de logs	158
11.3 Tipos de logging	159
11.3.1 Logging Circular	159
11.3.2 Logging de Archivado o Retención de log	159
11.4 Logging de Bases de datos desde el Centro de Control.....	160
11.5 Parámetros de Logging	162
11.6 Backup de Base de datos.....	163
Quicklab #10 – Programando un backup.....	165
11.7 Recupero de la Base de Datos.....	168
11.7.1 Tipos de Recupero	168
11.7.2 Restauración de Base de datos.....	169
11.8 Otras operaciones con BACKUP y RESTORE	169
Capítulo 12 – Tareas de Mantenimiento	171
12.1 REORG, RUNSTATS, REBIND.....	171
12.1.1 El comando REORG	172
12.1.2 El comando RUNSTATS	172
12.1.3 BIND / REBIND	173
12.1.4 Tareas de mantenimiento desde el Centro de control	174
12.2 Las opciones de Mantenimiento	175
Quicklab #11 – Configuración del mantenimiento automatizado	178

Capítulo 13 – Concurrencia y Bloqueo	181
13.1 Transacciones	181
13.2 Concurrencia	182
13.3 Problemas sin control de concurrencia	183
13.3.1 Actualizaciones perdidas	183
13.3.2 Lecturas no confirmadas o Uncommitted read	184
13.3.3 Lecturas no repetidas	185
13.3.4 Lectura fantasma	186
13.4 Niveles de aislamiento	187
13.4.1 Lectura no confirmada o Uncommitted read	187
13.4.2 Estabilidad de cursor o Cursor stability	187
13.4.3 Estabilidad de lectura o Read stability	187
13.4.4 Lecturas repetidas	188
13.4.5 Comparando niveles de aislamiento	188
13.4.6 Fijando los niveles de aislamiento	189
13.5 Escalamiento de Bloqueos	190
13.6 Monitoreo de bloqueos	191
13.7 Espera de Bloqueo	192
13.8 Bloqueo Mortal o Deadlock, causas y detección	192
13.9 Mejores prácticas para concurrencia y bloqueo	194
 PARTE III – APRENDIENDO DB2: DESARROLLO DE APLICACIONES	 195
 Capítulo 14 – Procedimientos Almacenados SQL PL	 197
14.1 IBM Data Studio	198
14.1.2 Creación de procedimiento almacenado en IBM Data Studio	199
14.2 Introducción a los procedimientos almacenados SQL PL	204
14.2.1 Estructura de un procedimiento almacenado	204
14.2.2 Atributos opcionales del procedimiento almacenado	205
14.2.3 Parámetros	205
14.2.4 Comentarios en un procedimiento almacenado SQL PL	206
14.2.5 Sentencias compuestas (Compound statements)	206
14.2.6 Declaración de variables.	207
14.2.7 Sentencia de asignación de estados	207
14.3 Cursores	207
14.4 Flujo de control	208
14.5 Llamando a procedimientos almacenados	208
14.6 Errores y manejo de condiciones	210
14.7 SQL Dinámico	212
 Capítulo 15 – SQL PL en-línea, Disparadores (Triggers) y UDFs	 215
15.1 SQL PL en-línea	215
15.2 Triggers (Disparadores)	217
Quicklab #12 – Creación de un disparador en el Centro de Control	220
15.3 Funciones definidas por el Usuario (UDFs)	223
15.3.1 Funciones escalares	223
15.3.2 Funciones de Tabla	224
Quicklab #13 – Creación de un UDF utilizando el IBM Data Studio	226

Capítulo 16 – DB2 pureXML	229
16.1 Uso de XML en bases de datos.....	230
16.2 Bases de datos XML	230
16.2.1 Bases de datos XML-enabled.....	230
16.2.2 Bases de datos Nativa XML	231
16.3 XML en DB2.....	232
16.3.1 Ventajas tecnológicas de pureXML	233
16.3.2 XPath básico	236
16.3.3 XQuery: Definición.....	239
16.3.4 Inserción de documentos XML	240
16.3.5 Consulta de datos XML	244
16.3.6 Unir con SQL/XML.....	249
16.3.7 Unir con XQuery	250
16.3.8 Operaciones de actualización y eliminación	251
16.3.9 Indexación XML.....	253
QuickLab #14 - SQL/XML y XQuery	255
Capítulo 17 –Desarrollo con Java, PHP, y Ruby	257
17.1 Desarrollo de aplicaciones en Java	257
17.1.1 Driver JDBC tipo 2.....	257
17.1.2 Driver JDBC tipo 4.....	258
17.2 Desarrollo de aplicaciones en PHP	259
17.2.1 Opciones de conexión de DB2 para PHP	259
17.2.2 Zend Core para IBM	261
17.3 Desarrollo de aplicaciones en Ruby on Rails.....	264
17.3.1 Startup Toolkit para DB2 on Rails.....	264
Apéndice A – Localización de fallas	265
A.1 Obteniendo más información acerca de los códigos de error.	266
A.2 SQLCODE y SQLSTATE.....	266
A.3. Bitácora de notificaciones de administración.....	267
A.4 db2diag.log	267
A.5 Trazas CLI.....	268
A.6 Defectos y arreglos de DB2	268
Recursos	269
Web sites:.....	269
Libros	270

Sobre este libro

Avisos y Marca Registrada

© Copyright IBM Corporation 2007, 2008
Todos los derechos son reservados.
IBM Canada
8200 Warden Avenue
Markham, ON
L6G 1C7
Canada

La presente documentación no puede copiarse ni reproducirse, total o parcialmente, de ningún modo ni a través de ningún medio así como tampoco traducirse a otro idioma sin el consentimiento previo de todos los propietarios del Copyright mencionados anteriormente.

IBM no ofrece garantías ni realiza manifestaciones respecto a este contenido y renuncia específicamente a cualquier garantía implícita de comercialización o idoneidad para cualquier fin determinado. IBM no asume ninguna responsabilidad por los errores que puedan aparecer en el presente documento, incluidos, aunque sin limitarse a ellos, los errores de traducción. La información contenida en el presente documento está sujeta a cambios sin previo aviso. IBM se reserva el derecho a realizar cualquier cambio de este tipo sin obligación de notificar a ninguna persona dichas revisiones o dichos cambios. IBM no se compromete a mantener actualizada la información aquí contenida.

La información contenida en el presente documento relacionada con los productos de IBM se ha obtenido del proveedor o de los proveedores de dichos productos. IBM no ha probado dichos productos y no puede confirmar la precisión del rendimiento, la compatibilidad ni ninguna otra afirmación relacionada con los productos de IBM. Las consultas sobre las capacidades de los productos de IBM se deben dirigir al proveedor o a los proveedores de dichos productos.

IBM, el logo de IBM, AIX, DB2, DB2 Connect, DB2 Universal Database, i5/OS, Informix, pureXML, Tivoli, WebSphere, and z/OS son marcas o marcas registradas de International Business Machines Corporation en los Estados Unidos, otros países, o ambos.

Java y todas las marcas registradas basadas en Java son marcas de Sun Microsystems, Inc. en los Estados Unidos, otros países, o ambos.

Microsoft y Windows son marcas registradas de Microsoft Corporation en los Estados Unidos, otros países, o ambos.

Linux es una marca registrada de Linus Torvalds en los Estados Unidos, otros países, o ambos.

UNIX es una marca registrada de The Open Group en los Estados Unidos, otros países, o ambos.

Otras empresas, productos, o nombres de servicios pueden ser marcas registradas o marcas de servicio de otros.

Las referencias realizadas en esta publicación a productos o a servicios de IBM no implican que IBM tenga la intención de comercializarlos en todos los países en los que IBM realiza operaciones comerciales.

A quién se le recomienda leer este libro?

Este libro puede ser leído por cualquier persona que trabaja o tiene la intención de trabajar con bases de datos, como administradores de base de datos (DBAs), desarrolladores, consultores, arquitectos de software, gerentes de producto, profesores y estudiantes.

Cómo está el libro estructurado?

En la parte I, Visión General y Configuración, se explica de qué se trata la edición Express-C de DB2, se introduce la familia de productos de DB2 y sus funcionalidades, se cubre instalación y creación de bases de datos, y se explora las herramientas disponibles con DB2.

En la parte II, Aprendiendo DB2: Administración de bases de datos, la intención es familiarizar al lector con el ambiente de DB2, su arquitectura, conexiones de clientes remotos, objetos de la base de datos, movimiento de datos (import/export/load), seguridad, backup y recuperación, concurrencia y bloqueo, y otros tareas comunes de mantenimiento.

En la parte III - Aprendiendo DB2: Desarrollo de Aplicaciones, se cubren procedimientos almacenados, funciones definidas por el usuario, triggers, SQL/XML, XQuery, desarrollo en Java™, PHP y Ruby.

El apéndice contiene información útil sobre cómo solucionar problemas.

Ejercicios que llamamos “Quicklabs” son incluidos en la mayoría de los capítulos; cualquier archivo requerido en esos ejercicios se pueden encontrar en el archivo zip **expressc_book_quicklabs_9.5.zip** que acompañan a este libro, o que se pueden bajar de la página Web de IBM® DB2 Express-C: www.ibm.com/db2/express.

El contenido de este libro también se utiliza en cursos ofrecidos como parte del programa “DB2 on Campus”, y en los videos disponibles en www.channelDB2.com/oncampus. Este libro también te puede preparar para los exámenes del programa DB2 on Campus que son ofrecidos por la comunidad. Puedes leer más sobre este programa en la página Web www.ibm.com/db2/oncampus.

Nota:

Para mas información sobre el programa DB2 on Campus, mira el video <http://www.channeldb2.com/video/video/show?id=807741:Video:3902>

Un libro para la comunidad

Este libro fue creado por miembros del equipo de DB2 Express-C y ofrecido gratuitamente a la comunidad DB2 Express-C. Muchos miembros de la comunidad a nivel mundial han traducido este libro a muchos idiomas. Si deseas ofrecer tus comentarios, contribuir con nuevo material, mejorar el material existente, o ayudar a traducir este libro a otro idioma,

por favor envíanos un correo electrónico a db2x@ca.ibm.com con el tipo de contribución que deseas hacer, y utilizando como título del correo "DB2 Express-C book changes."

Colaboradores

Quisieramos agradecer a los voluntarios de la comunidad de DB2 por el increíble trabajo de traducción! Ellos trabajaron por muchas horas y durante fines de semana para hacer de este trabajo una realidad. La tabla de abajo muestra la lista de colaboradores voluntarios:

Nombre del Colaborador	Empresa / Universidad	Cargo	Contribución
Calixto Maldonado	Universidad Tecnológica Nacional. Córdoba, y UESiglo21, Argentina	Docente de Base de Datos en la UTN - Córdoba, y tutor de las carreras de informática en la UESiglo21	Líder del equipo de traducción. Tradujo los capítulos 7, 10,11,12,13, Apéndice A
Ricardo Castañeda Muñoz	Universidad Nacional Autónoma de México (UNAM)	Miembro de la comunidad DB2 - UNAM	Tradujo los capítulos 8, 9, 15, Apéndice A
Diego Cruz Castillo	Pontificia Universidad Católica del Perú	Estudiante	Tradujo los capítulos 1, 2, 3 y 6
Antonio Guzmán Fernández	Universidad Nacional Autónoma de México (UNAM)	Miembro de la comunidad DB2 - UNAM	Tradujo los capítulos 4 y 5
Mariano Molina	Universidad Tecnológica Nacional. Córdoba, Argentina	Miembro Laboratorio IBM. Grupo Investigación DB2	Tradujo los capítulos 14, 16, 17
Jorge Rozas	Universidad Tecnológica Nacional. Córdoba, Argentina	Miembro Laboratorio IBM. Grupo Investigación DB2	Tradujo los capítulos 14, 16, 17
Gustavo F. Ruiz	Universidad Tecnológica Nacional. Córdoba, Argentina	Coordinador Laboratorio IBM	Tradujo los capítulos 14, 16, 17
Juan Enrique Sanchez Hidalgo	Pontificia Universidad Católica del Perú	Estudiante	Tradujo los capítulos 1, 2, 3 y 6
Pablo A. Torres Navarrete	Pontificia Universidad Católica del Perú	Estudiante	Tradujo los capítulos 1, 2, 3 y 6
Raúl F. Chong	Laboratorio de IBM en Toronto, Canadá	Gerente del programa DB2 on Campus	Verificó que la traducción al español fue correcta y arreglo el formato.

Agradecimientos

Agradecemos mucho a las siguientes personas por la ayuda prestada en la creación de los materiales utilizados en este libro:

- Ted Wasserman, Clara Liu y Paul Yip del laboratorio de IBM Toronto quienes crearon el material que sirvió de base para este libro.
- Don Chamberlin y Cindy Saracco por sus artículos en IBM developerWorks sobre XQuery, y Matthias Nicola por sus presentaciones sobre pureXML™.
- Kevin Czap y Grant Hutchison por sus materiales de instrucción técnica en DB2.
- Katherine Boyachok por el diseño de la portada del libro.
- Susan Visser por su ayuda en la publicación del libro.

Prólogo

Innovación es la piedra fundamental del progreso en el área tecnológica. En IBM, innovación ha sido parte fundamental en la evolución de nuestros servidores de datos. Habiendo sido pioneros en tecnologías de manejo de datos en los sesentas y setentas, hemos continuado haciendo disponibles tecnologías innovadoras en el área de manejo de datos. Esto se demuestra por las miles de patentes en esta área creados por especialistas de IBM. Como resultado, muchas de las grandes empresas del mundo de hoy en día cuentan en los productos de IBM, como DB2, para manejar sus soluciones que requieren de mucho poder y capacidad y que son de misión crítica.

Sin embargo, DB2 ya no es solo para grandes empresas. Con el ofrecimiento de DB2 Express-C, la tecnología de punta de DB2 es ahora también disponible para pequeñas y medianas empresas y de manera gratuita! Aun cuando existen otros servidores de datos libres en el mercado, DB2 Express-C ofrece ventajas únicas.

Existen muchas innovaciones presentes en DB2 Express-C. Algunas de estas innovaciones tienen como objetivo el tener más capacidad de almacenamiento, otras a reducir el costo administrativo, otras a mejorar la performance, y otras a reducir el costo de infraestructura. No discutiré todos estos temas ahora con la expectativa de que leas el libro, pero en manera breve, quiero describir una de ellas.

DB2 Express-C ha sido construido en base a la tecnología 'Viper', que lo hace el primer servidor de datos híbrido para manejar tanto datos relacionales, como XML de manera nativa. Esto hace a DB2 ideal para ser el motor de aplicaciones SOA y Web 2.0 donde datos XML circulan en abundancia. Mientras que otros servidores de datos de otras compañías comerciales tienen límites en la cantidad de datos a almacenar, DB2 Express-C no los tiene. Tampoco tiene límite en el número de base de datos que puedes crear en el sistema. Por supuesto, si necesitas de soporte técnico de IBM, ayuda esta a tu disposición con un simple clic.

Este libro sirve como una guía para conocer y empezar a usar DB2 Express-C. El libro te ayudará a comprender conceptos de DB2 y te ayudará a desarrollar habilidades de administración y desarrollo de aplicaciones con DB2. La habilidad y el conocimiento ganado con la ayuda de este libro son relevantes para otras ediciones avanzadas de DB2 en Linux, UNIX, y Windows.

Aun cuando DB2 Express-C no es un producto de código abierto (open-source), en IBM creemos mucho en apoyar y promover iniciativas de la comunidad. Me alegra mucho que este libro creado por la comunidad de DB2 Express-C sea ofrecido de manera gratuita. Te aliento a enriquecer y ayudar a actualizar el libro con tu know-how, experiencias, y asistas a traducir este libro a otros idiomas para que otros también se beneficien con tu conocimiento.



Arvind Krishna
Vice President, Data Servers
Information Management, IBM Software Group

PARTE I – VISION GENERAL Y CONFIGURACION

1

Capítulo 1 – Qué es DB2 Express-C?

DB2 Express-C es un miembro de la familia IBM DB2 de poderosas aplicaciones de servidores de datos para manejar tanto datos relacionales como XML. DB2 Express-C es una edición de DB2 libre, sin límites y fácil de usar. La 'C' en DB2 Express-C significa Comunidad. Una comunidad de usuarios DB2 Express-C que se juntan para ayudarse unos a otros, tanto en línea como fuera de ella. La comunidad DB2 Express-C consiste en una variedad de personas y compañías que diseñan, desarrollan, implementan o utilizan soluciones de base de datos, como:

- Desarrolladores de aplicaciones que requieren un software de base de datos de estándar abierto para construir aplicaciones standalone, cliente-servidor, web y empresariales.
- ISVs, vendedores de hardware, vendedores de infraestructura de pila y proveedores de otros tipos de solución que quieran incluir o empotrar un completo servidor de datos como parte de sus soluciones.
- Consultores, administradores de base de datos y arquitectos IT que necesiten un servidor de datos robusto para entrenamiento, desarrollo de habilidades, evaluación y prototyping.
- Startups, pequeñas y medianas compañías que necesitan un servidor de datos confiable para sus aplicaciones y operaciones.
- Estudiantes, profesores y otros usuarios académicos que quieran un servidor de datos altamente versátil para enseñanza, courseware, proyectos e investigaciones.

DB2 Express-C comparte el mismo núcleo de funcionalidad y código base como las ediciones pagadas de DB2 para Linux, UNIX y Windows. DB2 Express-C puede correr en sistemas de 32-bits y 64-bits con sistemas operativos Windows o Linux. Puede correr en un sistema que tenga cualquier cantidad de núcleos y de memoria. No tiene ningún requerimiento especial de almacenamiento o de configuración del sistema que sean especiales. DB2 Express-C también incluye pureXML sin costo. pureXML es la tecnología única de DB2 para almacenar y procesar documentos XML nativamente.

1.1 Libre para desarrollar, implementar y distribuir... ¡sin límites!

Esta oración resume las ideas centrales de DB2 Express-C:

- **Libre para desarrollar:** Si eres un desarrollador de aplicaciones y necesitas una base de datos para tu aplicación, tú puedes usar DB2 Express-C.

- **Libre para implementar:** Si estás trabajando en un ambiente de producción y necesitas una base de datos para almacenar tus registros vitales, tú puedes usar DB2 Express-C.
- **Libre para distribuir:** Si estás desarrollando una aplicación o herramienta que requiera un servidor de datos empotrado, tú puedes incluir DB2 Express-C. Aún si DB2 Express-C está empotrado en tu aplicación y distribuido cada vez que vendes tu aplicación, este sigue siendo libre. Es requerido que te registres con IBM para poder redistribuir DB2 Express-C; sin embargo este registro también es libre.
- **Sin límites:** Mientras que otros competidores de base de datos establecen límites en el tamaño de la base de datos, con DB2 Express-C NO hay límite de datos. Tu base de datos puede crecer y crecer sin violar el acuerdo de licencia. También no hay límites en términos del número de conexiones de usuarios por servidor.

Nota:

Para aprender más acerca de DB2 Express-C y su rol en el mundo de la información por demanda y la Web 2.0, échale un vistazo a este video presentación (en inglés):

<http://www.channeldb2.com/video/video/show?id=807741:Video:3922>

1.2 Asistencia al usuario y soporte técnico

Si tienes preguntas técnicas sobre DB2 Express-C, puedes publicar tus preguntas en el foro de DB2 Express-C. Este foro libre es monitoreado por un equipo dedicado de DB2 Express-C, aunque es la comunidad quien provee la mayor parte de las respuestas de manera voluntaria. IBM también ofrece a los usuarios la opción de adquirir una suscripción anual de bajo costo (también conocido como Suscripción y Licencia de 12 Meses o Licencia de Términos Establecidos o FTL, siglas en inglés). Esta suscripción de DB2 Express-C viene con el apoyo de IBM para soporte técnico 24x7 y actualizaciones de software. Por un pago de bajo costo renovable anual (\$2,995 por Servidor por Año en los Estados Unidos – puede variar en otros países) no solo recibes soporte y mantenimiento de software para tu servidor DB2 Express-C, sino también dos principales add-on: HADR (Recuperación de Desastres de Alta Disponibilidad, por sus siglas en inglés) y replicación SQL (para replicar información con otros servidores DB2).

1.3 Servidores DB2

Todas las ediciones de servidores DB2 contienen el mismo núcleo de componentes; están empaquetados de tal manera que los usuarios puedan escoger las funciones que necesitan al precio correcto. Figura 1.1 (en inglés) ilustra las diferentes ediciones de productos DB2.

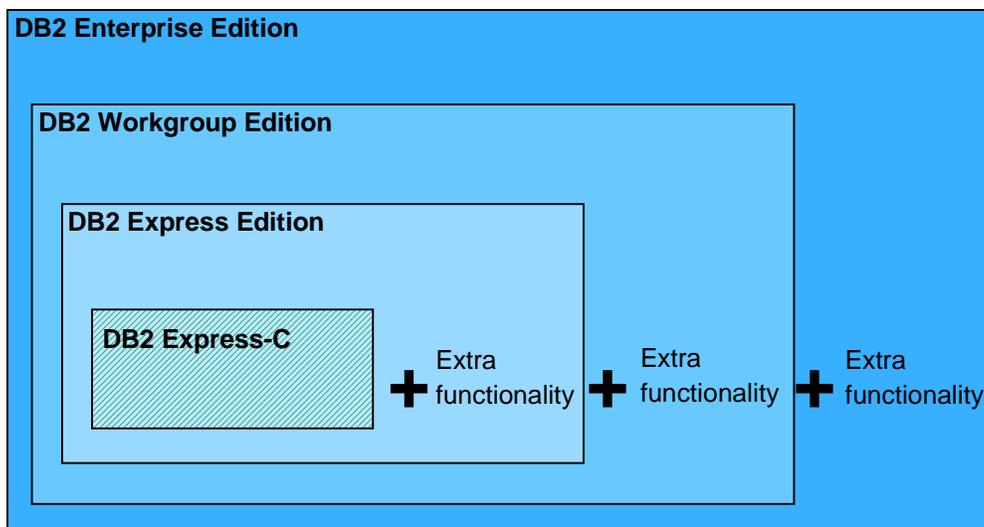


Figura 1.1 – Servidores DB2

Como se muestra en la Figura 1.1, DB2 Express-C es igual que DB2 Express solo que sin algunos componentes. DB2 Express-C es libre a la comunidad. Como se mencionó antes, asistencia técnica esta disponible a través de un foro libre en línea, o puedes recibir el soporte técnico oficial de IBM 24x7 si adquieres la licencia de suscripción de 12 meses.

La Figura 1.1 también explica porqué es tan fácil actualizar desde DB2 Express-C. Si deseas actualizar a cualquier otro servidor DB2 en el futuro, todos los servidores DB2 tiene el mismo núcleo de componentes. Esto también significa que una aplicación desarrollada para una edición trabajará, sin ninguna modificación, en otra edición. Y cualquier habilidad que aprendas en una edición aplicará a las otras ediciones.

1.4 Clientes DB2 y controladores

Un cliente DB2 incluye las funcionalidades necesarias para conectarse a un servidor DB2, sin embargo, un cliente no siempre tiene que ser instalado. Por ejemplo, una aplicación JDBC Type 4 puede conectarse directamente a un servidor DB2, previsto del correcto controlador ya cargado. Los clientes DB2 vienen en diferentes sabores:

- Cliente Servidor de Datos IBM (IBM Data Server Client): el más completo, incluye herramientas GUI, controladores.
- Cliente de Tiempo de Ejecución de Servidor de Datos (IBM Data Server Runtime Client): funcionalidad básica para conectarse e incluye controladores.
- Módulos fusionados de Cliente de Tiempo de Ejecución DB2 para Windows (DB2 Runtime Client Merge Modules for Windows): principalmente usado para incluir un cliente de Tiempo de Ejecución DB2 como parte de la instalación de una aplicación Windows.

Figura 1.2 muestra los diferentes clientes DB2 y controladores disponibles.

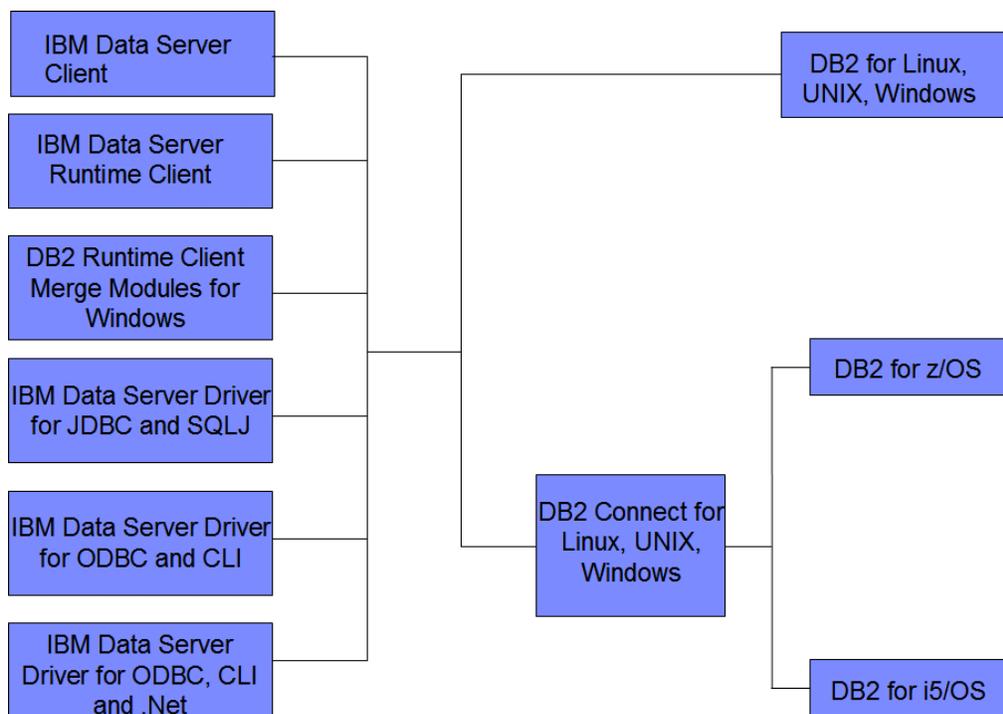


Figura 1.2 – Clientes DB2 y controladores

Al lado izquierdo de la Figura 1.2, todos los clientes DB2 y controladores son mostrados. Aunque todos los clientes DB2 incluyen sus controladores requeridos, a partir de DB2 9 proveemos de controladores individuales también. Todos los clientes DB2 y controladores son gratis y disponibles para descargar desde el sitio web de DB2 Express-C. Los clientes y controladores pueden ser usados para conectarse a un servidor DB2 en Linux, UNIX o Windows. Para conectarse a servidores DB2 para z/OS® o DB2 para i5/OS® necesitarás ir a través de un servidor DB2 Connect (mostrado en el medio de la Figura 1.2). Discutiremos acerca del software DB2 Connect en el Capítulo 2.

1.5 Libertad en el desarrollo de aplicaciones

DB2 ofrece un ambiente de desarrollo de aplicaciones que es basado en estándares y es transparente a lo largo de la familia DB2. La estandarización SQL a lo largo de los productos DB2 provee un grupo común de interfaces de programación de aplicaciones para acceso a base de datos.

Adicionalmente, cada producto DB2 provee precompiladores SQL que permiten a los desarrolladores que empotren SQL estático y dinámico en aplicaciones portátiles. DB2 inclusive tiene un proveedor manejado en .NET nativo e integración con herramientas de Microsoft® Visual Studio. Los lenguajes y estándares que puedes usar en DB2 son:

- Ruby on Rails
- C/C++ (ODBC y SQL empotrado)
- JDBC y SQLJ
- COBOL
- Borland
- Python
- PHP
- Perl
- Lenguajes .NET
- OLE-DB
- ADO
- Servicios Web
- SQL
- MS Office: Excel, Access, Word

1.6 Versiones DB2 contra Ediciones DB2

Si eres nuevo en DB2, debes estar un poco confundido con respecto a la diferencia entre una versión DB2 y una edición DB2.

Cada pocos años IBM lanza públicamente una nueva Versión DB2. Una Versión incluye nuevas características y mejoras significativas al producto. Actualmente, DB2 Versión 8 y DB2 Versión 9 están soportados oficialmente por IBM. Una Versión también puede tener algunas Releases que pueden incluir alguna funcionalidad nueva pero usualmente no significativa como para garantizar una nueva Versión. Por ejemplo, 8.1 y 8.2 son niveles Releases para DB2 Versión 8. Viendo el pasado historial, IBM saca un nuevo Release de DB2 aproximadamente cada año; sin embargo, nuevas Versiones están espaciadas típicamente 2 o 3 años. El Release más reciente es V9.5 (anteriormente apodado como DB2 'Viper 2') y se convirtió Disponible en General (GA, por sus siglas en inglés) en Octubre del 2007. Cada Release puede tener también varios niveles Modification, que típicamente contienen parches o corresponden a niveles de paquetes de parches, y contienen nueva funcionalidad. Al momento de escribir esto el más reciente nivel Versión, Release, Modification (V,R,M) de DB2 Express-C es 9.5.0 que corresponde a 9.5 con Fix pack 0, que significa que es está en nivel GA.

Por otro lado, las ediciones son empaquetamientos o agrupamiento de paquetes dentro de cada versión. Como ha sido discutido anteriormente, una edición es un empaquetamiento de diferentes funciones por un precio y licencia dados. DB2 Versión 9.5 (también conocido como DB2 9.5) tiene varias ediciones: por ejemplo, DB2 Express-C 9.5, DB2 Express 9.5, DB2 Workgroup 9.5, y DB2 Enterprise 9.5 (ver Figura 1.1).

1.7 Moviéndose a otra edición DB2

Mientras tu base de datos necesite crecer, puedes necesitar actualizar a una edición DB2 que soporte una mayor configuración de hardware. Si la situación se presenta, es fácil actualizar a otra edición DB2:

- Si estás actualizando a otra edición de DB2 en un mismo sistema de computadora, instala la nueva edición de DB2 encima de DB2 Express-C, y la nueva licencia correspondiente. Tu base de datos no será borrada (pero una copia de seguridad siempre es recomendable).
- Si estás actualizando DB2 donde la nueva edición será instalada en una computadora diferente y más grande usando el mismo sistema operativo: instala la nueva edición DB2 en la computadora grande, haz un backup de la base de datos de la computadora más pequeña, mueve los backup a la computadora grande y restaura desde los backup la base de datos en esta. También puedes necesitar las propiedades de configuración (dbm cfg) de la computadora pequeña y aplicar esta configuración a la grande. Los comandos de restauración y backup son discutidos en más detalle en el Capítulo 11, Backup y Recuperación. El dbm cfg es discutido en más detalle en el Capítulo 4, El Ambiente DB2.
- En cualquier caso tu aplicación no necesitará modificación.

1.8 Mantenimiento de DB2 Express-C

Como ha sido discutido anteriormente, existen dos opciones de soporte para DB2 Express-C:

1. Comprar la licencia de suscripción de 12 meses. Esto te provee de cobertura a tiempo completo del Soporte Técnico IBM DB2, te da la habilidad de instalar software actualizadores DB2 (también llamados fixpacks).
2. Usar el foro en línea de la comunidad DB2 Express-C. Es totalmente gratis, pero viene sin soporte oficial de IBM. A su vez, bajo esta opción, IBM no se compromete a proveer nuevas características y parches a las fechas dadas. El concepto de fixpacks que es discutido en el Capítulo 2, no se aplica tampoco; en cambio, refrescos de toda la imagen de DB2 Express-C se hacen disponibles de tiempo en tiempo. Mientras nuevas versiones salen, puedes esperar típicamente que imágenes refrescadas de DB2 Express-C estén disponibles para estas nuevas versiones en vez de las antiguas.

1.9 Software Libre relacionado

Todo el software que está disponible para descarga desde el sitio de descarga de DB2 Express-C (www.ibm.com/db2/express/download.html) está libre de cargos. Además de las imágenes de DB2 Express-C (para Linux y Windows, en arquitecturas de 32 y 64-bits), hay otros software útiles que pueden ser descargados y usados gratuitamente:

- IBM Data Studio
- DB2 9 Embedded Application Server
- DB2 9.5 Net Search Extender

También existen starter toolkits adicionales basados en DB2 Express-C y disponibles para descarga desde el sitio web de IBM Alphaworks (www.alphaworks.ibm.com/datamgmt) que puedes encontrar útiles:

- Starter Toolkit for DB2 on Rails
- Web 2.0 Starter Toolkit for DB2

Si buscas una versión ligera de un servidor de aplicaciones, y que sea gratis, IBM ofrece:

- WebSphere® Application Server – Community Edition (WAS CE)

1.9.1 IBM Data Studio

IBM Data Studio es una herramienta basada en Eclipse que permite diseñar, desarrollar, implementar y manejar tu información, base de datos y aplicaciones de base de datos a lo largo de todo el ciclo de vida del manejo de información. Data Studio reemplaza la solución previamente disponible DB2 Developer WorkBench 9.1.

IBM Data Studio ayuda a desarrollar funciones definidas por el usuario, procedimientos almacenados, XQuery, declaraciones SQL, e incluye un debugger integrado. Adicionalmente, Data Studio permite trabajar con diagramas de Modelado de Datos Físicos para entender las relaciones de entidades entre tablas. También puede servir para desarrollar y publicar información como servicio Web sin programar. Discutiremos de Data Studio en el Capítulo 14, Procedimientos Almacenados de SQL PL.

1.9.2 DB2 9.5 Net Search Extender

Con DB2 9.5 Net Search Extender puedes ejecutar rápidas y detalladas búsquedas de texto en documentos de texto, incluyendo cualquier documento XML almacenado nativamente en DB2 9.5.

1.9.3 Starter Toolkit for DB2 on Rails

Starter Toolkit for DB2 on Rails es un conjunto de productos y tecnologías convenientemente empaquetadas que permite la rápida creación de un entorno para construir aplicaciones Web DB2 usando tecnología Ruby on Rails. Todos el software requerido está incluido: DB2 Express-C, DB2 driver for Ruby, DB2 adapter for Rails; junto con tutoriales, ejemplos y otros materiales de aprendizaje. Discutiremos de Ruby on Rails más adelante en el Capítulo 17, Desarrollo en Java, PHP y Ruby.

1.9.4 Web 2.0 Starter Toolkit for DB2

Web 2.0 Starter Toolkit for DB2 es una manera fácil de empezar con DB2, PHP y Dojo. Ayuda a implementar el software requerido, te conecta con tutoriales e incluye aplicaciones demo. Dos de las aplicaciones demo son Atom Services Control Panel, que genera Atom feeds de tablas DB2; y Web Services Control Panel, que crea envolturas de servicios web REST alrededor de tus tablas DB2. Ambos se apoyan en Dojo por sus significativas capacidades en Ajax y widgets.

1.9.5 WebSphere Application Server – Community Edition

IBM WebSphere Application Server Community Edition es un servidor de aplicaciones Java EE 5 ligero disponible de manera gratuita. Construido en base a la tecnología Apache Geronimo, incluye las últimas innovaciones de la comunidad de software de

código abierto para ofrecer un entorno integrado, fácilmente accesible y flexible para desarrollar y desplegar aplicaciones Java. Soporte técnico es opcional para WAS CE a través de una suscripción anual.

2

Capítulo 2 – Características y productos relacionados

Este capítulo describe características de DB2 incluidas con la compra de DB2 Express-C con licencia de suscripción de 12 meses. También incluye otras características no disponibles con DB2 Express-C, pero como parte de otras ediciones de DB2, en algunos casos, con pagos adicionales.

Las características incluidas en la versión gratuita de DB2 Express-C son:

- La funcionalidad base de DB2
- Control Center, Data Studio y otras herramientas de manejo
- pureXML
- Utilización de recursos hasta 2GB y 2 núcleos
- Disponible en Linux, Windows, y Solaris (x86)

Las capacidades disponibles con la licencia de 12 meses de DB2 Express-C son:

- Fix packs
- Alta disponibilidad y Recuperación Crítica (HADR)
- Data Replication (Homogenous SQL)
- Utilización de recursos hasta 4GB y 4 núcleos (en 2 tomas)

La siguiente table lista características del producto y si están o no incluidas con las distintas ediciones de DB2. Características que pueden ser compradas separadamente están listadas por nombre para la edición correspondiente.

Funcionalidad	Express-C fixed term license	DB2 Express Edition	DB2 Workgroup Server Edition	DB2 Enterprise Server Edition
Replicación SQL Homogénea	Sí	Sí	Sí	Sí
Net Search Extender	Sí	Sí	Sí	Sí

Funcionalidad	Express-C fixed term license	DB2 Express Edition	DB2 Workgroup Server Edition	DB2 Enterprise Server Edition
Spatial Extender	Sí	Sí	Sí	Sí
Tecnología pureXML™	Sí	pureXML	pureXML	pureXML
Alta disponibilidad y Recuperación Crítica	Sí	Alta Disponibilidad	Sí	Sí
Tivoli® System Automation	Sí	Alta Disponibilidad	Sí	Sí
Advanced Copy Services	No	Alta Disponibilidad	Sí	Sí
Reorganización en línea	No	Alta Disponibilidad	Sí	Sí
Federación Homogénea	No	Federación Homogénea	Federación Homogénea	Federación Homogénea
MQT	No	No	Optimización de consultas	Sí
MDC	No	No	Optimización de consultas	Sí
Paralelismo de consultas	No	No	Optimización de consultas	Sí
Conexión concentrada	No	No	No	Sí
Partición de tablas	No	No	No	Sí
DB2 Governor	No	No	No	Sí
Compresión a nivel fila	No	No	No	Optimización de almacenamiento

Funcionalidad	Express-C fixed term license	DB2 Express Edition	DB2 Workgroup Server Edition	DB2 Enterprise Server Edition
Compresión: backup	No	No	No	Optimización de almacenamiento
Label-based access control (LBAC)	No	No	No	Advanced Access Control
Geodetic Extender	No	No	No	Geodetic Data Management
Query Patroller	No	No	No	Optimización de Performance
DB2 workload management	No	No	No	Optimización de Performance
Performance Expert	No	No	No	Optimización de Performance
Replicación Q Homogénea	No	No	No	Homogeneous Replication
Particionamiento de base de datos	No	No	No	No

Table 2.1: Ediciones de DB2 Versión 9.5: características y funcionalidad soportada

Características incluidas en otras ediciones de DB2:

Características pagadas en la edición DB2 Express

- pureXML
- Alta disponibilidad
- Federación Homogénea

Características incluidas sin costo en la edición DB2 Workgroup:

- Alta disponibilidad
- Disponible en AIX®, Solaris, y HP-UX además de Linux y Windows

Características pagadas en la edición DB2 Workgroup

- pureXML
- Optimización de Performance (MQT, MDC, Query Parallelism)
- Federación Homogénea

Características incluidas sin costo en la edición DB2 Enterprise:

- Particionamiento de Tablas (Range)
- Materialized Query Tables (MQT)
- Multi-dimensional Clustering (MDC)
- Alta disponibilidad y Recuperación Crítica (HADR)
- Conexión Concentrada

Características pagadas en la edición DB2 Enterprise

- pureXML
- Optimización de almacenamiento. (incluido compresión)
- Control de acceso avanzado (seguridad avanzada)
- Optimización del Performance (Performance Experto, Query Patroller)
- Administración de datos Geodetic
- Federación Homogénea DB2

Productos pagados relacionados a DB2:

- DB2 Connect
- DB2 Warehouse Editions
- WebSphere® Federation Server
- WebSphere Replication Server

2.1 Características incluidas con la licencia de suscripción de DB2 Express-C

Esta sección explica los Fix packs de DB2, HADR y réplica de datos.

2.1.1 Fix Packs

Un fix pack de DB2 es un paquete de código corregido aplicado en un producto DB2 instalado, para corregir publicaciones diferentes después de que el producto fue liberado. Con una licencia de suscripción, la corrección de paquetes es gratis de descargar e instalar. Ellos están típicamente disponibles cada tres meses.

Para bajar los últimos fix packs, revisa esta página del soporte técnico de DB2 en http://www.ibm.com/software/data/db2/support/db2_9/

2.1.2 Alta disponibilidad de Recuperación Crítica (HADR)

Alta disponibilidad de Recuperación Crítica (HADR) es una característica de fiabilidad de base de datos que proporciona una alta disponibilidad y la solución de recuperación de datos ante desastres de sitio completos así como parciales. Un ambiente HADR generalmente consiste en dos servidores de datos, el primario y el secundario (que puede estar en posiciones geográficamente distantes). El servidor primario es donde la fuente de base de datos es almacenada y accesada por aplicaciones de clientes. Como las transacciones son procesadas sobre la base de datos primaria, los registros de la base de datos automáticamente son transportados al servidor secundario a través de la red. El servidor secundario tiene una copia clonada de la base de datos primaria, por lo general creada por el back-up de la base de datos primaria y restaurándolo sobre la base de datos secundaria. Cuando los registros de la base de datos primaria son recibidos ellos son

reproducidos de nuevo y aplicados a la base de datos secundaria. Por la repetición continua de los registros, la base de datos secundaria mantiene una réplica en sincronización de la base de datos primaria que puede asumir si la base de datos primaria falla.

Una solución completa de DB2-supported HADR nos da lo siguiente:

- Capacidad rápida de solución ante fallas, con transparencia completa para clientes y aplicación de cliente
- Atomicidad completa en las transacciones para prevenir pérdida de datos
- Habilidad de actualizar sistemas o aplicaciones sin interrupción de servicios visibles
- Sistema remoto de fallas, proporcionando recuperación completa de desastre local que puede dañar el centro de datos
- Administración fácil con herramientas graficas de DB2
- Todos esto con un impacto insignificante sobre funcionamiento total del sistema

Nota:

Para ver una demostración (en inglés) de cómo trabaja el HADR, puedes visitar:

<http://www-306.ibm.com/software/data/db2/express/demo.html>

2.1.3 Réplica de Datos

Esta característica permite la réplica de datos entre un servidor fuente donde los cambio de datos con capturados, y un servidor objetivo donde el cambio de datos son aplicados. Figura 2.1 proporciona una descripción de cómo trabaja la réplica.

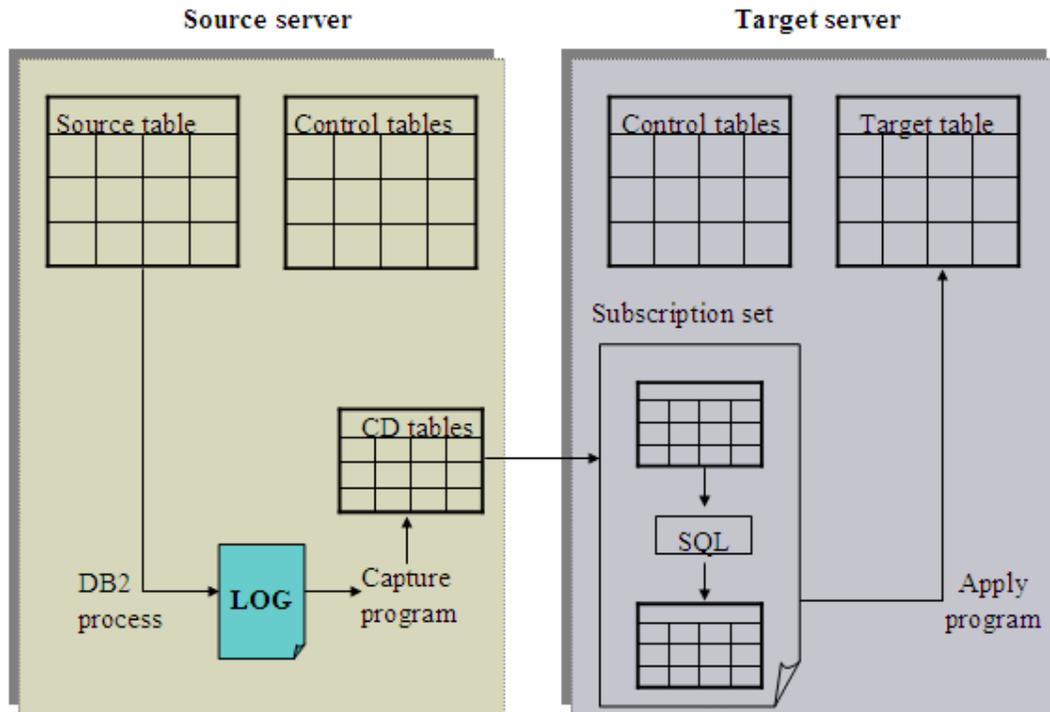


Figura 2.1 –Réplica de SQL

En la Figura 2.1 hay dos servidores, un servidor fuente y un servidor objetivo. En el servidor fuente, el programa “Capture” captura el cambio hecho en la base de datos. En el servidor objetivo, un programa “Apply” aplica los cambios de la base de datos replica. La réplica es útil para una variedad de propósitos que requieren replica de datos, incluyendo alivio de capacidad, sosteniendo el data warehouse y el mercado de datos, revisando el historial de cambios. Usando las características de la réplica SQL puedes replicar datos entre DB2 Express-C y otros servidores DB2, incluyendo aquellos en otros sistemas Linux, UNIX, z/OS, y i5/OS.

2.2 Características no disponibles con DB2 Express-C

Esta sección describe alguna de las características disponibles en otras ediciones de DB2 pero no en DB2 Express-C.

2.2.1 Particionamiento de la Base de Datos

La característica de particionamiento de la base de datos (DPF) es solo disponible con DB2 Enterprise Edition con la licencia de pago adicional. Esto permite a la base de datos ser extendido a través de múltiples particiones los cuales pueden estar alojados en varias computadoras. DPF está basado en una arquitectura shared-nothing. Cada computadora, como es añadido al grupo de partición, trae el poder adicional de procesamiento de datos con sus propias CPUs y memoria. DPF es en particular útil en ambientes de servidor de datos grandes como data warehouses donde las sentencias de los sistemas de apoyo de decisión (DSS) son controlados.

2.2.2 Concentrador de conexión

El concentrador de conexiones es una característica que permite el soporte de un largo número de usuarios conectados simultáneamente. Previamente, toda conexión a la base de datos requiere un “agente” de base de datos. El concentrador de conexiones introduce el concepto de un “agente lógico”, permitiendo un agente para mantener muchas conexiones. El concepto de “agentes” serán discutidos con más detalle en el Capítulo 6, Arquitectura DB2.

2.2.3 Geodetic Extender

DB2 Geodetic Extender esta disponible con pago adicional para DB2 Enterprise Edition. Este extender hace desarrollos para aplicaciones de inteligencia de negocios y e-government que requieren un análisis de localización geográfica mucho más sencillo. DB2 Geodetic Extender puede construir un globo mundial virtual a cualquier escala. La mayoría de información de localización es tomada usando sistemas worldwide, por ejemplo el sistema de satélites de posicionamiento global (GPS), y puede ser representado en coordenadas de latitud/longitud (geocódigo). Los datos de negocios, como direcciones, pueden ser convertidos a geocódigo con DB2 Geodetic Extender y aplicaciones a nivel empresarial trabajan mejor cuando mantienen los datos en esta forma sin proyección, saliendo las proyecciones del mapa (tierra a mapa plano) donde ellos pertenecen, en la capa de presentación, para mostrar e imprimir mapas.

2.2.4 Workload Management (WLM)

Maneja cargas de trabajo en una base de datos en base a prioridades de un usuario y de aplicaciones combinadas con la disponibilidad de recursos y límites. Permite regular la carga de trabajo de tu base de datos y consultas, de tal manera que consultas importantes y de alta prioridad pueden correr inmediatamente, y , and previene consultas espontáneas que no tienen buen rendimiento de monopolizar los recursos del sistema de manera que tu sistema puede correr eficientemente. WLM is Nuevo en DB2 9.5 y prove mas poder que las características de las herramientas Query Patroller y del DB2 Governor que eran disponibles en versiones anteriores de DB2.

2.3 Productos pagados relacionados con DB2

2.3.1 DB2 Connect

DB2 Connect es un software pagado que permite al cliente DB2 para Linux, UNIX o Windows conectarse a un servidor DB2 para z/OS o para i5/OS, como se muestra en la Figura 2.2. DB2 Connect no es requerido cuando la conexión ocurre en la dirección opuesta: cuando conectas desde DB2 para z/OS o i5/OS hacia DB2 para Linux, UNIX o Windows. DB2 Connect viene en dos ediciones principales dependiendo de tus necesidades de conexión: DB2 Connect Personal Edition y DB2 Connect Enterprise Edition.

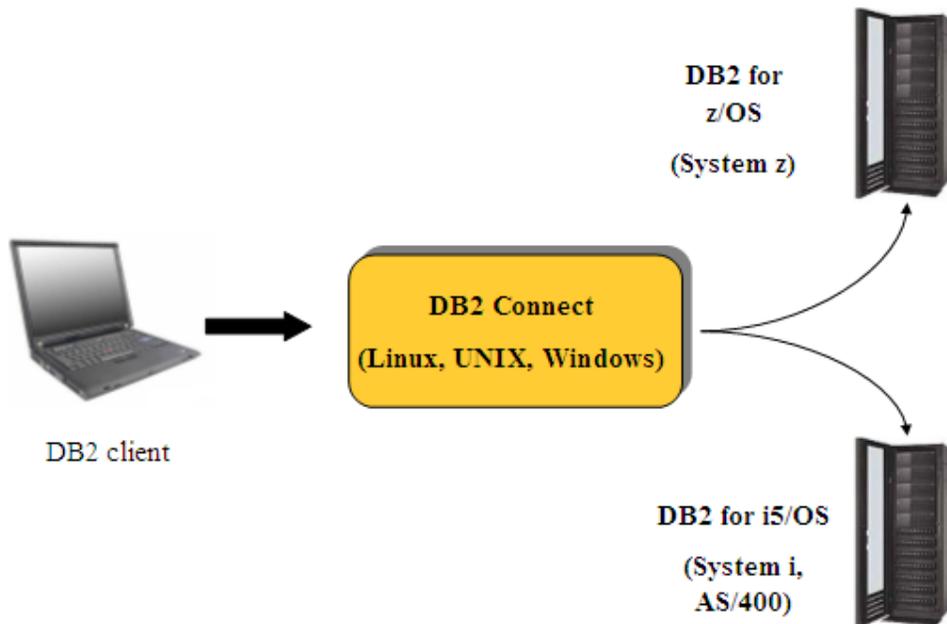


Figura 2.2 – DB2 Connect

2.3.2 WebSphere Federation Server

Antiguamente conocido como WebSphere Information Integrator (para soporte de federación), WebSphere Federation Server permite una federación de base de datos; significando que puedes correr consultas de base de datos que puedan trabajar con objetos de diferentes sistemas de base de datos relacional. Por ejemplo, si compras WebSphere Federation Server, puedes correr la siguiente consulta:

```
SELECT *
FROM   Oracle.Table1 A
       DB2.Table2 B
       SQLServer.Table3 C
WHERE  A.col1 < 100
       and B.col5 = 1000
       and C.col2 = 'Test'
```

Figura 2.3 provee una ilustración donde WebSphere Federation Server es usado.

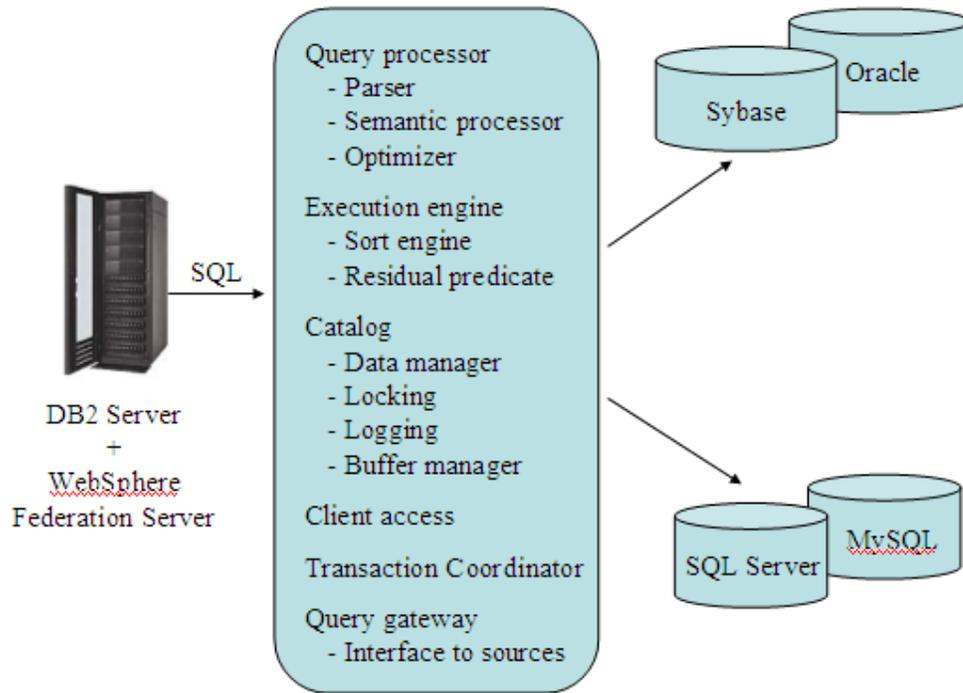


Figure 2.3 – WebSphere Federation Server

Para sistemas de manejo de base de datos relacionales que son parte de la familia IBM, soporte de federaciones está incluido en DB2 Express-C. Esto significa que WebSphere Federation Server no es requerido cuando, por ejemplo, puedes correr una consulta entre dos base de datos diferentes, o entre una base de datos DB2 y una base de datos Informix (Informix es parte de la familia IBM).

2.3.3 WebSphere Replication Server

Antiguamente conocido como WebSphere Information Integrator (para soporte de replicación), WebSphere Replication Server permite replicación SQL en registros de base de datos cuando servidores de datos que no son de IBM están involucrados. Esto también incluye una característica conocida como Q-Replication para replicar información usando cola de mensajes.

3

Capítulo 3 – Instalación de DB2

Para instalar la edición DB2 Express-C tanto en Linux como en Windows, asegúrate que tu sistema cumpla con los requisitos para la instalación.

3.1 Requisitos para la instalación

Con respecto a la versión del sistema operativo y el nivel de requerimientos, DB2 Express-C esta disponible en Linux, Solaris (x64), y Windows 2003, 2000, XP, y Vista. La arquitectura del procesador esta disponible para 32-bit, 64-bit y PowerPC (Linux). Si necesita correr DB2 sobre otra plataforma (como UNIX), usted debe comprar una de las ediciones de servidor de datos diferentes descritas en capítulos anteriores en este libro. Los requerimientos del sistema operativo para todas las ediciones DB2 también son descritos en este documento:

<http://www.ibm.com/software/data/db2/udb/sysreqs.html>

En términos de recursos de hardware, DB2 Express-C puede ser instalado sobre sistemas con cualquier número de núcleos de CPU y memoria, sin embargo, esto sólo utilizará hasta 2 núcleos y 2GB de memoria para la versión de licencia gratis y sin garantías, y hasta 4 núcleos y 4GB de memoria para la licencia de suscripción de 12 meses y versión con soporte. Los sistemas pueden ser físicos, o virtuales creados por particionamiento o controlando el software de máquina virtual. Usted, desde luego puede correr sobre sistemas más pequeños si usted prefiere, por ejemplo un sistema de un solo procesador con 1GB de memoria.

Para la última información sobre el requerimiento de hardware para DB2 Express-C, revisa la pagina web de DB2 Express-C:

<http://www-306.ibm.com/software/data/db2/express/getstarted.html>

3.2 Permisos en el Sistema Operativo para la instalación

Para instalar DB2 Express-C en Linux o Windows, debes ser un usuario con los permisos suficientes para poder instalar aplicaciones sobre el Sistema Operativo.

Para **Linux**, se necesita ser root (o superusuario) para poder instalar DB2 Express-C. Además puedes instalar DB2 Express-C sin necesidad de ser usuario root, sin embargo esto te limita con lo que puedas hacer con el producto. Por ejemplo, bajo una instalación no-root, no puedes crear más instancias que el creado por defecto al momento de la instalación.

Para **Windows**, la cuenta del usuario debe pertenecer al grupo Administrador en las maquinas donde instalarás el producto. Alternativamente, una cuenta de usuario no-Administrador puede ser usada, siempre y cuando un Administrador de Windows primero configure los privilegios de Windows que permitan al no-Administrador realizar la instalación.

Para cuentas de dominio de Windows, si se quiere verificar al usuario ID sobre el servidor DB2, el usuario ID de instalación debe pertenecer al grupo de Dominio de Administradores sobre el dominio donde las cuentas van a ser creadas. Usted también puede usar la cuenta de built-in Local System para correr la instalación para todos los productos.

La cuenta de usuario tiene que tener el derecho para "Accede a esta computadora desde cualquier lugar".

Nota:

Mira este video (en inglés) sobre la instalación de DB2 Express-C en este link:
<http://www.channeldb2.com/video/video/show?id=807741:Video:4442>

3.3 Instalación con ayuda gráfica guiada (Wizards)

Existen muchos métodos para instalar DB2 Express-C, el método más fácil es usar el Wizard de instalación de DB2 (DB2 Installation Wizard) de interface gráfica (GUI-based). Luego de bajar y descomprimir la imagen DB2 Express-C, podemos seguir con el Wizard que nos guiará en la instalación:

- Windows: ejecuta el archivo setup.exe en el directorio EXP/image
- Linux: ejecuta el comando db2setup (en consola) en el directorio exp/disk1

DB2 Express-C es muy fácil de instalar siguiendo las instrucciones del DB2 installation wizard. En la mayoría de casos, las configuraciones por defecto son suficientes, por lo que lo único que debe hacer es aceptar la licencia, clic en el botón "Next" varias veces, y clic en el botón "Finish". Después de poco tiempo tu instalación estará completada y DB2 esta ¡listo para ser usada!

La Figura 3.1 muestra el DB2 Setup Launchpad. Clic en "Install a product" y luego elige "Install New" para instalar una nueva copia de DB2 Express-C en tu sistema.

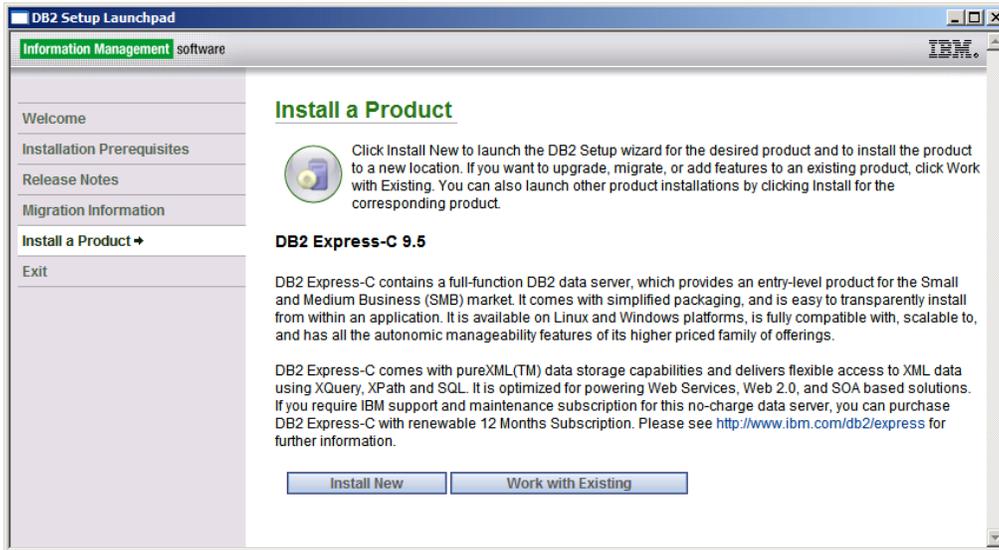


Figura 3.1 – El DB2 Setup Launchpad

Después de aceptar la Licencia, usualmente es suficiente elegir la instalación por defecto “Typical” como se muestra en la Figura 3.2

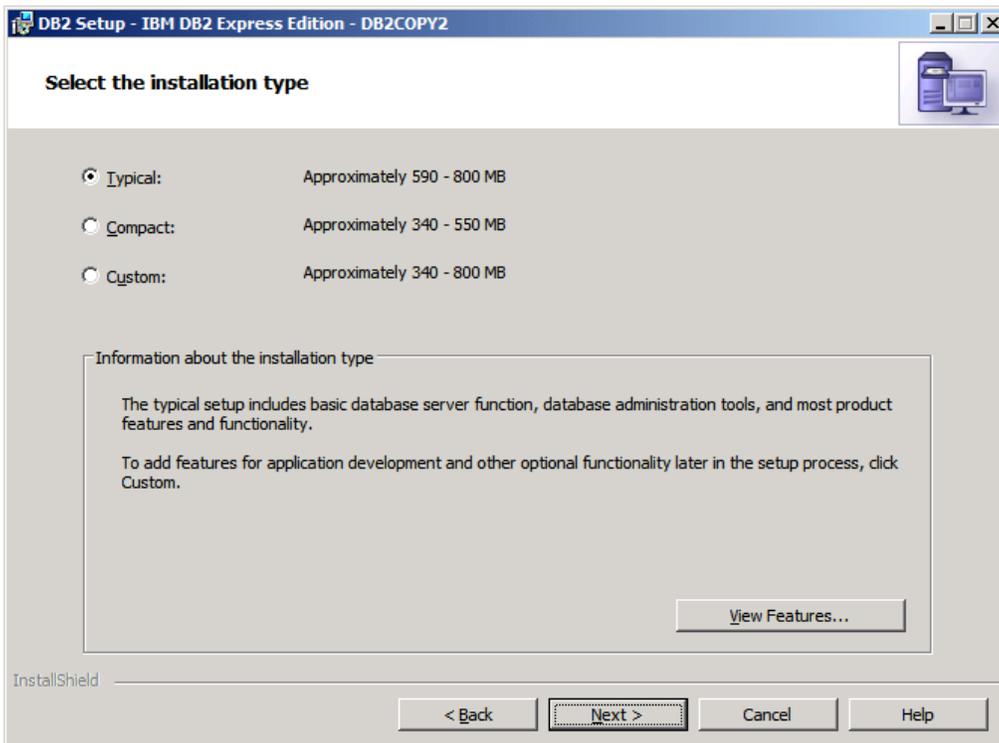


Figura 3.2 – Tipos de instalación.

En la Figura 3.3, usted tiene la opción para instalar el producto, crear un archivo de respuesta, o ambos. Los archivos de respuesta se explican en la sección 3.4, Silent Install. Eligiendo la opción por defecto (Instala IBM DB2 Express Edition en esta computadora y guarda mis configuraciones en un archivo de respuesta) es más que suficiente.

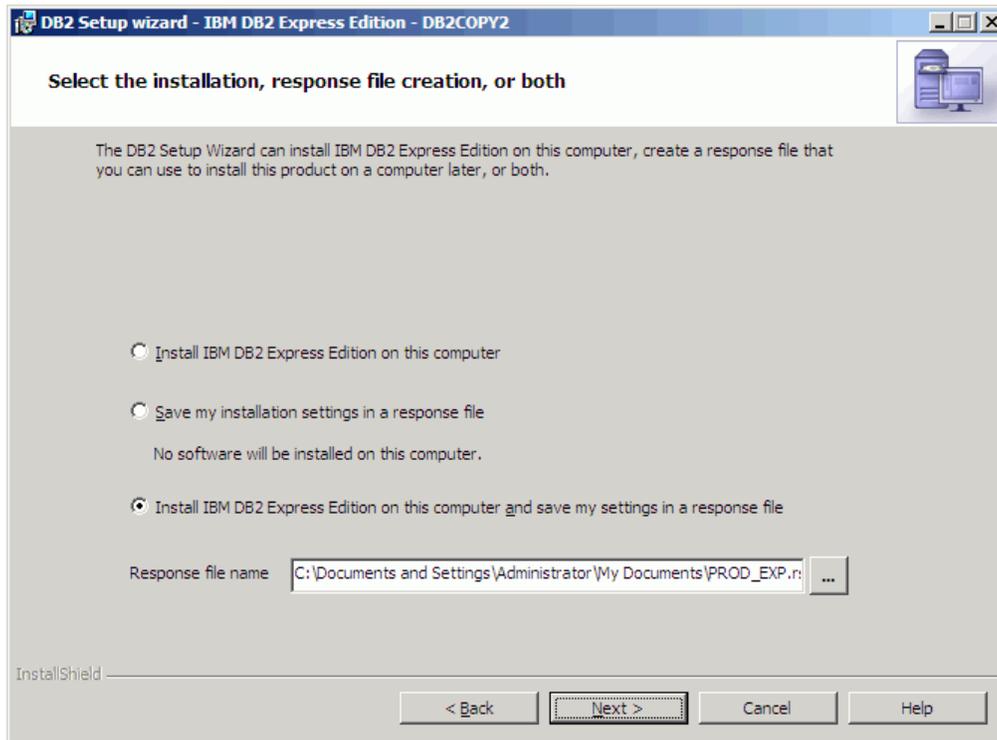


Figura 3.3 – Seleccionando la instalación.

Escoja valores por defecto para las siguientes pantallas. Cuando usted se pone a la ventana mostrada en la Figura 3.4, usted puede introducir a un usuario existente que estará acostumbrado trabajar con la instancia y otros servicios. Este usuario debe ser parte del grupo de Administrador Local en Windows. Si el ID de usuario que usted introduce no pertenece a un usuario existente, será creado como un Administrador Local. Usted puede dejar el campo de dominio en blanco si el ID de usuario no pertenece a un dominio. EL ID de usuario por defecto a crear en Windows es llamado db2admin. En el caso de Linux, el ID de usuario por defecto creado es llamado db2inst1.

DB2 Setup - IBM DB2 Express Edition - DB2COPY2

Set user information for the default DB2 instance

Specify the required user information that the DB2 instance, DB2_01, will use to log on to your system.

It is highly recommended that you use a local user or domain user account instead of the LocalSystem account. Further details are available by clicking Help.

Local user or Domain user account

LocalSystem account

Use the same user name and password for the remaining DB2 services

InstallShield

< Back Next > Cancel Help

Figura 3.4 – Especificando la información de usuario por la instancia por defecto de DB2

Finalmente, en la Figura 3.5, la installation wizard muestra un resumen de lo que será instalado, y las diferentes configuraciones propuestas en los pasos anteriores. Cuando das clic en “Finish”, la instalación comenzará, y los archivos de programa serán alojados en tu sistema.

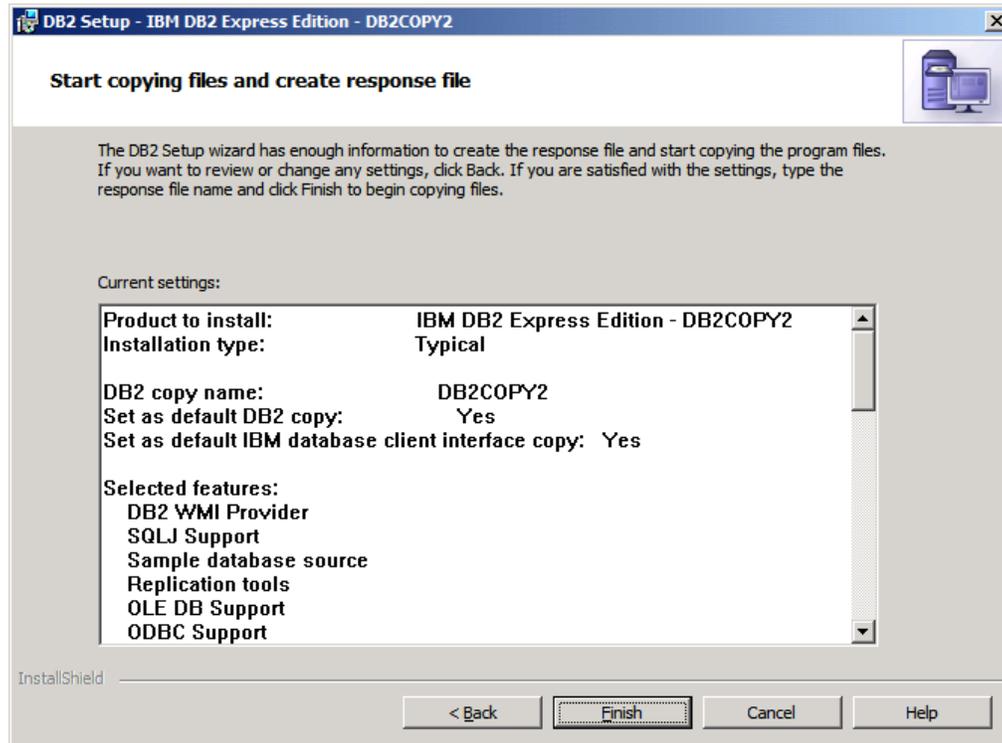


Figura 3.5 – Resumen de lo que será instalado.

3.4 Silent Install (Instalación silenciosa)

Existen situaciones donde necesitas instalar un cliente DB2 en múltiples computadoras; o necesitas incrustar un DB2 data server como parte de tu aplicación, y te gustaría instalarlo como parte del proceso de instalación de tu aplicación. Para estos casos, un silent install es la forma ideal para instalar DB2.

DB2 permite silent install a través del uso de los archivos de texto de respuesta que almacenan la información de la instalación. Lo siguiente muestra un pequeño ejemplo de un archivo de respuesta.

Ejemplo del código de un pequeño archivo de respuesta.

```
PROD=UDB_EXPRESS_EDITION
```

```
LIC_AGREEMENT=ACCEPT
```

```
FILE=C:\Program Files\IBM\SQLLIB\
```

```
INSTALL_TYPE=TYPICAL
LANG=EN
INSTANCE=DB2
DB2.NAME=DB2
DEFAULT_INSTANCE=DB2
DB2.SVCENAME=db2c_DB2
DB2.DB2COMM=TCPIP
...
```

Existen un número de formas de generar un archivo de respuesta.

- ▶ Instala DB2 Express-C usando el Wizard para la instalación de DB2. Una de las primeras opciones del Wizard le permite seleccionar el checkbox para guardar su respuestas de instalación a un archivo de respuesta. Al final del Wizard, un archivo de respuesta será generado en un directorio y el nombre del archivo que usted especifique. Esto es un archivo de texto, por lo que usted puede editarlo después. Esto lo mostramos en la Figura 3.3.
- ▶ Edita el archivo de respuesta de ejemplo empaquetado con la imagen DB2. Este archivo de ejemplo (denotado con una extensión .rsp file extension) está localizado en el directorio *db2/platform/samples/*
- ▶ Para Windows, además usas el comando generador de archivo de respuesta:
`db2rspgn -d <output directory>`

Luego para instalar DB2 usando un archivo de respuesta, usa en Windows el comando:
`setup -u <response filename>`

Para Linux usa el comando:
`db2setup -r <response filename>`

QuickLab #1: Instala DB2 Express-C & crea la base datos SAMPLE

Objetivos

Antes de comenzar a explorar todas las características y herramientas que vienen con DB2 Express-C, primero debes instalarlo en tu sistema. En este Quicklab, realizaremos una instalación básica de DB2 Express-C en Windows. La misma installation wizard está disponible en Linux; por lo tanto los pasos son similares en ambas plataformas.

Procedimiento

1. **Obtén las imagen de DB2 Express-C.** Descarga la imagen apropiada de DB2 Express-C, u ordena el Discovery Kit DVD que incluye la imagen del DB2 Express-C de la página web www.ibm.com/db2/express. Descomprime los archivos en el directorio que desees.
2. **Localiza los archivos.** Dirígete hacia el directorio (o unidad) en donde se encuentran los archivos de instalación que has descomprimido.
3. **Corre el Launchpad.** Ejecuta el DB2 Launchpad haciendo doble clic en el archivo `setup.exe`. En Linux, ejecuta el commando `db2setup` como root. De el Launchpad, clic en la opción *Install Product* en el panel izquierdo de la ventana.
4. **Corre el DB2 setup wizard.** El DB2 setup wizard revisa si cumple todos los requerimientos del sistema y ve si existe otra instalación de DB2. Clic en el botón Next para seguir con la instalación.
5. **Revisa los términos de licencia.** Lee y acepta los términos de licencia (selecciona el radio button "*I Accept...*") y clic al botón Next para continuar.
6. **Elige el tipo de instalación.** Para este ejercicio, selecciona la opción *Typical* (opción por defecto). La opción *Compact* realiza una instalación básica, mientras la opción *Custom* permite personalizar la configuración específica que usted quiere instalar. Clic en el botón Next para continuar.
7. **Selecciona la carpeta para la instalación.** Esta pantalla te permite elegir la unidad y el directorio donde el código de DB2 es instalado en tu sistema. Asegúrate que exista espacio suficiente para la instalación. Usa la unidad y el directorio por defecto para este ejemplo (mostrado a continuación):

Unidad: C:

Directorio: C:\Archivos de Programa\IBM\SQLLIB

Clic en el botón Next para continuar.

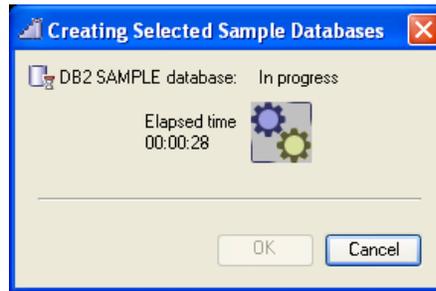
8. **Configura la información del usuario.** Una vez que DB2 Express-C está instalado, asegúrate que los procesos de DB2 están corriendo como Servicios del Sistema. Estos servicios requieren una cuenta del sistema operativo para poder correrlo. En el ambiente de Windows, usar la cuenta por defecto **db2admin** es recomendada. Si esta cuenta aun no existe, DB2 la creará por ti en el Sistema Operativo. También puedes especificar una cuenta existente, pero esa cuenta

debe de tener los permisos de administrador local. Recomendamos usar las opciones sugeridas. Asegúrese de especificar el password para la cuenta. En Linux usamos el UserID por defecto **db2inst1** (dueño de la instancia), **db2fenc1** (usuario para rutinas limitadas) y **dasusr1** (usuario del DB2 Administrator Server, DAS). Clic en el botón Next para continuar.

9. **Configura la instancia DB2.** Piense en el concepto de instancia DB2 como un contenedor para bases de datos. Una instancia debe existir antes de que una base de datos pueda ser creada dentro de ella. En una instalación en Windows, la instancia llamada *DB2* se creará automáticamente. En Linux, el nombre por defecto de la instancia es db2inst1. Cubriremos ambos casos mas adelante en este libro.

Por defecto, la instancia *DB2* esta configurada para escuchar conexiones TCP/IP en el puerto 50000. El protocolo por defecto y el puerto pueden ser cambiados dándole clic en *Protocols* y el botón *Startup*, respectivamente. Recomendamos usar las opciones por defecto para este ejemplo. Clic en el botón *Next* para continuar.

10. **Comenzar con la instalación.** Revisa las opciones de instalación seleccionadas previamente. Clic el botón *Install* para comenzar a copiar los archivos a la carpeta destino de instalación. DB2 también realizará algunos procesos de configuración iniciales.
11. **Primeros pasos.** Luego que la instalación ha terminado, otra herramienta llamada First Steps arranca automaticamente. First Steps también puede ser arrancada con el comando db2fs.
12. La base de datos SAMPLE es una base datos que puedes usar para probar DB2. Esta es creada automáticamente luego de la instalación de DB2. Verifica que la base de datos exista abriendo la herramienta DB2 Control Center. Para abrir esta herramienta, del botón Inicio de Windows elige: *Inicio -> Programas -> IBM DB2 -> DB2COPY1 (Default) -> General Administration Tools -> Control Center*. Ademas puedes iniciar el Control Center con el comando db2cc.
13. Si la base de datos SAMPLE se muestra en el Control Center, te puedes saltar al paso 16. Si no se muestra, selecciona el item *Refresh* del menu Control Center *View* para asegurar que puedes ver toda la información refrescada. Si la base de datos SAMPLE sigue sin aparecer, es porque no ha sido creada. Lo crearemos manualmente del First Steps. Elige donde dice "Database Creation", y luego guiate con el Wizard para crear la base de datos SAMPLE. Asegúrate que los objetos XML y SQL y las opciones de datos son seleccionados, y luego clic en OK. Esta última opción creará una base de datos UNICODE el cual fue requerido en la version 9 para soportar XML puro (pureXML), pero no es necesitado en DB2 9.5.
14. La siguiente ventana aparece cuando la base de datos esta siendo creada. (Este procedimiento puede tomar algunos minutos). Cuando la base de datos es creada, clic en el botón *OK* y cierra la herramienta First Steps.



15. Regresa al Control Center y verifica que la base de datos SAMPLE aparezca en el panel Object Tree. Tienes que refrescar la vista del Control Center para ver los nuevos cambios efectuados. Para hacer esto, selecciona el ítem *Refresh* del menú Control Center *View*.
16. **Reinicia la computadora.** A pesar que este paso no está mencionado en la documentación oficial de instalación DB2, recomendamos reiniciar el sistema (si es posible, por lo menos en Windows) para asegurarnos que todos los procesos comiencen satisfactoriamente y limpien cualquier recurso de memoria que no han sido eliminados correctamente. Esto es OPCIONAL.

4

Capítulo 4 – El Ambiente DB2

En este capítulo discutiremos el ambiente DB2. La figura 4.1 provee un resumen acerca de DB2, y la elipse roja destaca el área de estudio de este capítulo. El lado izquierdo de la figura muestra los diferentes comandos DB2, SQL, SQL/XML, y sentencias XQuery que pueden ser creadas para interactuar con un servidor de datos DB2. A la mitad de la figura se muestra los nombres de diferentes herramientas usadas para interactuar con un servidor de datos DB2. El lado derecho de la figura muestra el ambiente básico de DB2 que consiste en una instancia, una base de datos, y los archivos asociados de configuración.

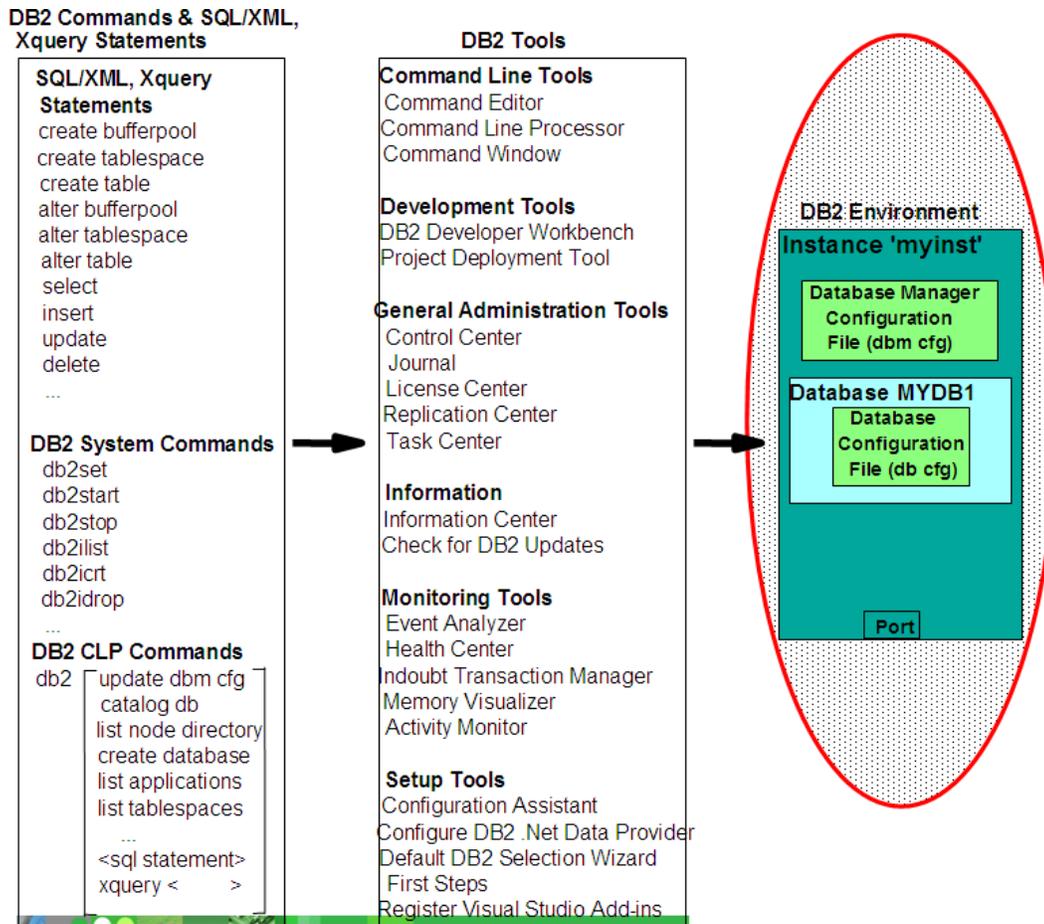


Figura 4.1 – La vista general de DB2: El ambiente DB2

Nota:

Vea las presentaciones en video acerca del ambiente DB2 en estos links:
<http://www.channeldb2.com/video/video/show?id=807741:Video:4029>
<http://www.channeldb2.com/video/video/show?id=807741:Video:4042>

Para describir el ambiente DB2, vamos a describir cada elemento componente paso a paso. La figura 4.2 muestra una representación de un servidor de datos DB2 después de instalar DB2 Express-C 9.5.

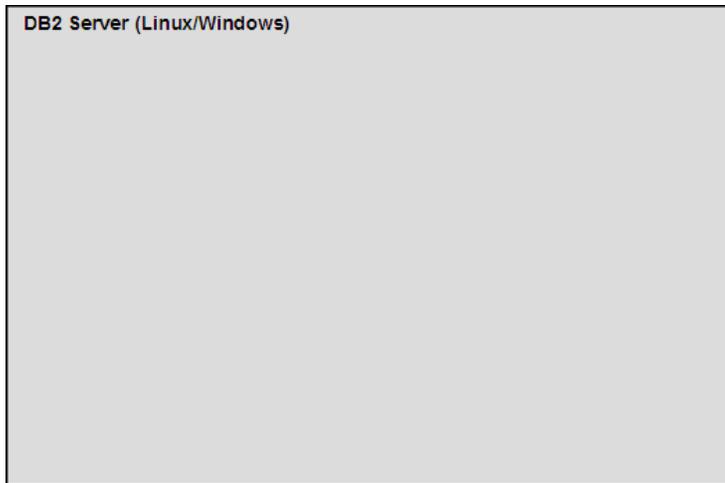


Figura 4.2 – Representación de un Server DB2 después de instalar DB2 Express-C 9.5

Como parte de la instalación en Windows, una instancia predeterminada llamada "DB2" ("db2inst1" en Linux) es creada. Está representada por la caja verde en la figura 4.3. Una instancia es simplemente un ambiente independiente donde las aplicaciones pueden correr, y las bases de datos pueden ser creadas. Se pueden crear múltiples instancias en un servidor de datos, y usarlas para diferentes propósitos. Por ejemplo, una instancia puede ser usada para mantener bases de datos en producción, otra instancia puede ser usada para un ambiente de prueba de bases de datos, y otra para un ambiente de desarrollo. Todas esas instancias son independientes es decir que las operaciones de una instancia no afectarán a las operaciones de otras instancias.

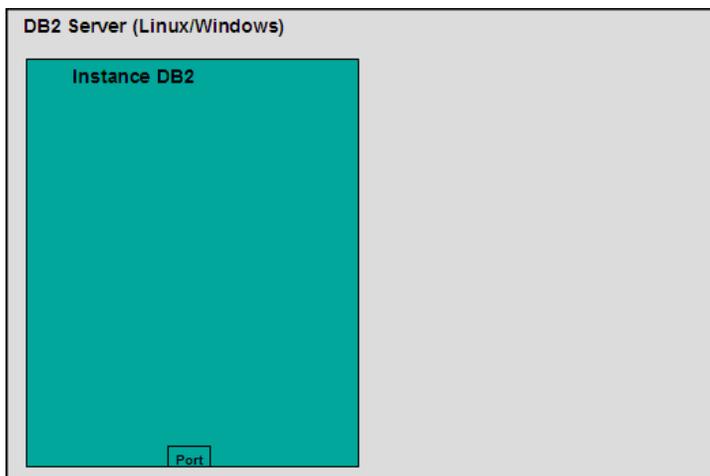


Figura 4.3 – La instancia predeterminada DB2 es creada

Para crear una nueva instancia, usa el comando `db2icrt <nombre de la instancia>`, donde `<nombre de la instancia>` es reemplazado con cualquier nombre de 8 caracteres. Por ejemplo, para crear la instancia `myinst`, nosotros usamos este comando: `db2icrt myinst`.

La Figura 4.4 muestra una nueva instancia llamada `myinst` como una caja verde separada.

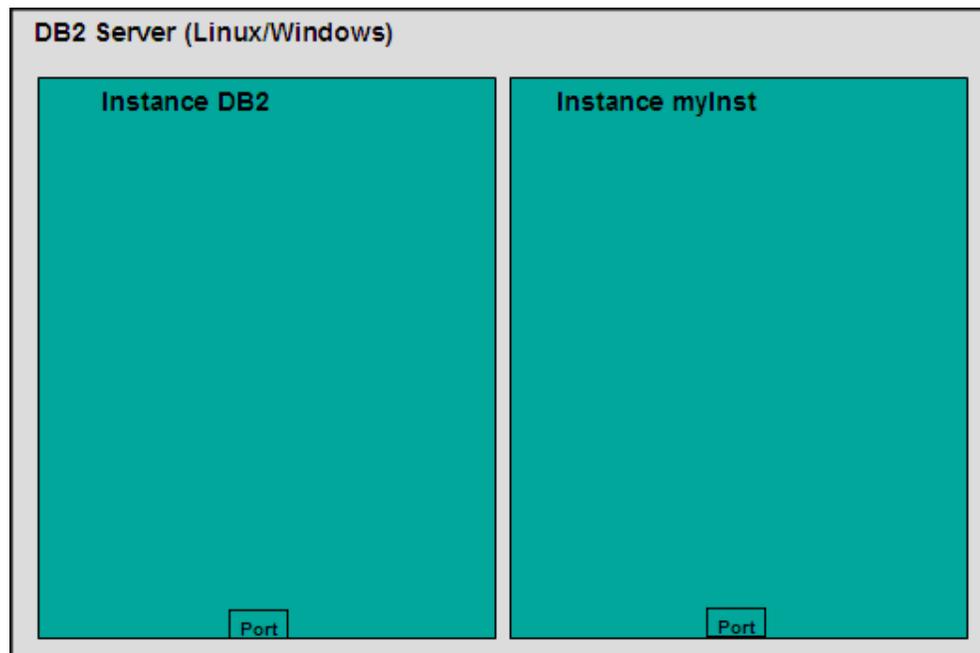


Figura 4.4 – Un servidor DB2 con dos instancias

Note que cada instancia tiene un único número de puerto. Esto ayuda a distinguir entre las instancias cuando se requiere conectar a una base de datos a través de una instancia dada desde un cliente remoto. Si se usa la Ventana de Comandos de DB2, se puede hacer que cualquier instancia de DB2 esté activa usando este comando de sistema operativo en Windows:

```
set db2instance=myinst
```

En este ejemplo, al crearse una base de datos desde la Ventana de Comandos, ésta será creada en la instancia `myinst`.

Para listar las instancias, corre el comando:

```
db2ilist
```

En Linux, una instancia debe corresponder con un usuario de sistema operativo; por lo tanto, para cambiar entre instancias puedes simplemente cambiar usuarios (con el comando `su`).

La tabla 4.1 muestra algunos comandos útiles a nivel de instancia.

Comando	Descripción
<code>db2start</code>	Inicia la instancia actual
<code>db2stop</code>	Detiene la instancia actual
<code>db2icrt</code>	Crea una nueva instancia
<code>db2idrop</code>	Borra una instancia
<code>db2ilist</code>	Lista las instancias existentes en el sistema
<code>db2 get instance</code>	Lista la instancia activa actualmente

Tabla 4.1 – Comandos útiles de DB2 a nivel de instancia

Algunos de los comandos de arriba pueden ser ejecutados a través del Centro de Control . Por ejemplo, en el Centro de Control , al expandir la carpeta *Instances* y a continuación clic derecho en la instancia que se desee y luego se elige *Start*, lo cual es equivalente a usar el comando `db2start` desde el DB2 Ventana de Comandos, o *Stop*, el cual es el equivalente a ejecutar el comando `db2stop` como se muestra en la figura 4.5.

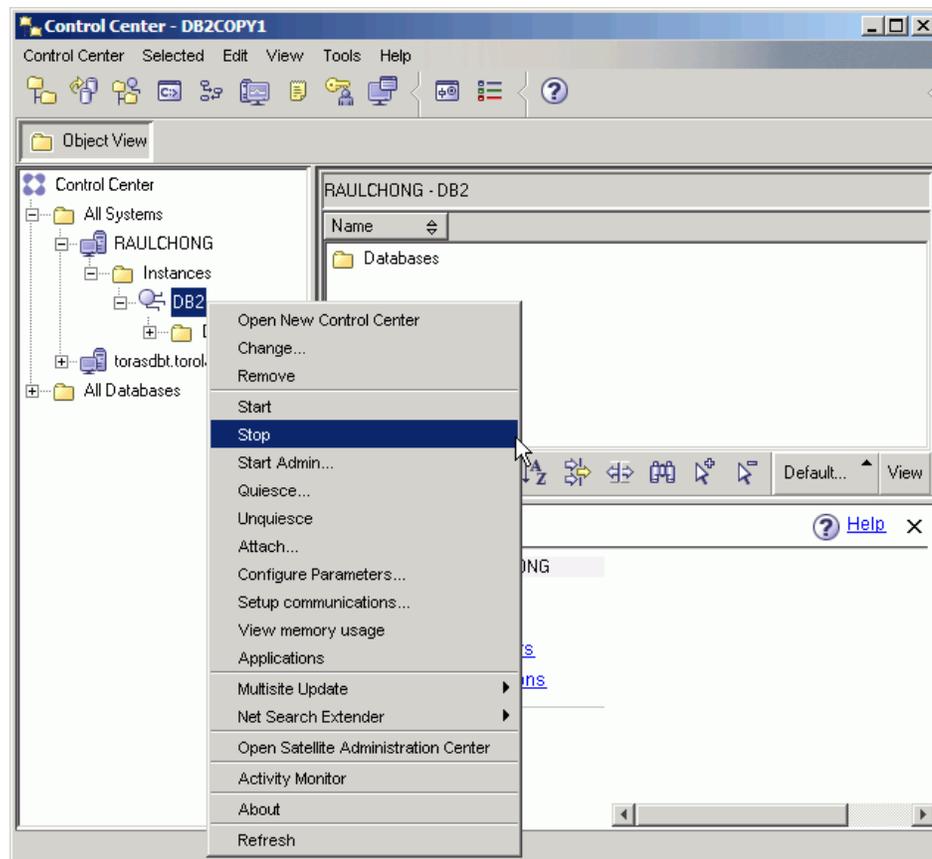


Figura 4.5 – Comandos de instancia desde el Centro de Control

Para crear una base de datos en una instancia activa, use este comando desde DB2 Ventana de Comandos:

```
db2 create database mydb1
```

Para listar todas las bases de datos creadas, ejecuta el comando:

```
db2 list db directory
```

Dentro de cualquier instancia, puedes crear muchas bases de datos. Una base de datos es una colección de objetos como tablas, vistas, índices, etc. Las bases de de datos son unidades independientes, y es por ello, que no comparten objetos con otras bases de datos. La figure 4.6 muestra la representación de una base de datos “MYDB1” creada dentro de la instancia “DB2”.

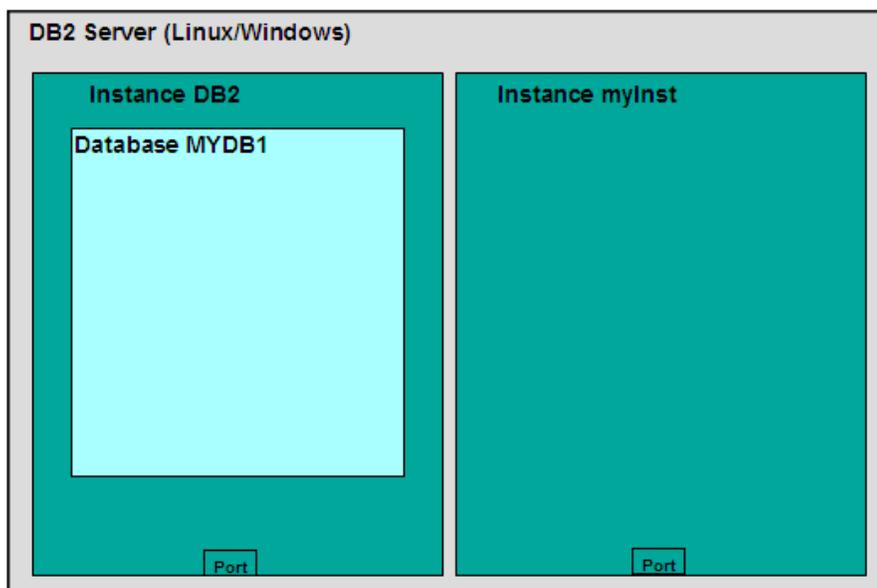


Figura 4.6 – Base de datos “MYDB1” creada en la instancia “DB2”

La tabla 4.2 muestra algunos comandos que puedes usar a nivel de la base de datos.

Comando/sentencia SQL	Descripción
db2 create database	Crea una nueva base de datos
db2 drop database	Borra una base de datos
db2 connect to <database_name>	Conecta hacia una base de datos
db2 create table/create view/create index	Sentencia SQL para crear una tabla, vista e índices respectivamente

Tabla 4.2 – Comandos/Sentencias SQL a nivel de la base de datos

Si queremos crear otra base de datos con el mismo nombre (MYDB1) pero en la instancia “myinst”, los siguientes comandos, desde la ventana de comandos del DB2, podrían ser usados:

```
db2 list db directory
set db2instance=myinst
db2 create database mydb1
set db2instance=db2
```

La figura 4.7 muestra la nueva base de datos “MYDB1” creada en la instancia “myInst”.

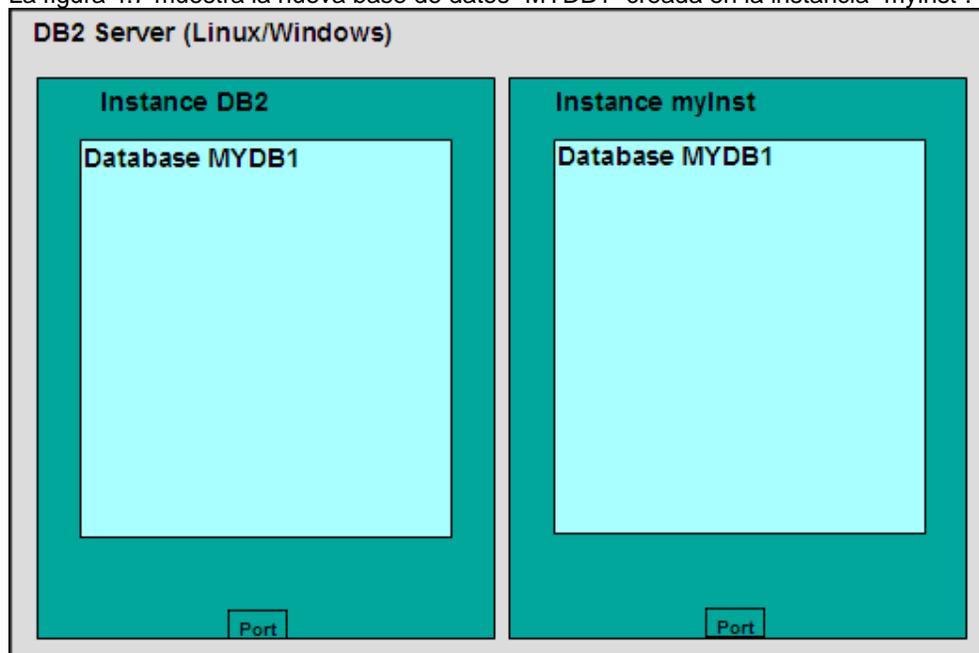


Figura 4.7 – Base de datos “MYDB1” creada en la instancia “myInst”

Cuando una base de datos es creada, hay varios objetos que también se crean de forma predeterminada: los espacios de tablas o Tablespaces, las tablas, un buffer pool (área de memoria) y archivos log. Crear esos objetos toma un poco tiempo, ése es el porqué de los minutos que requiere el comando `create database` para ser procesado. La figura 4.8 muestra tres tablespaces creados de forma predeterminada. Los tablespaces se discutirán con más detalle en el Capítulo 6, Arquitectura DB2; pero por ahora, piense en los tablespaces como la capa lógica entre tablas (construcción lógica) y los recursos físicos como discos y memoria.

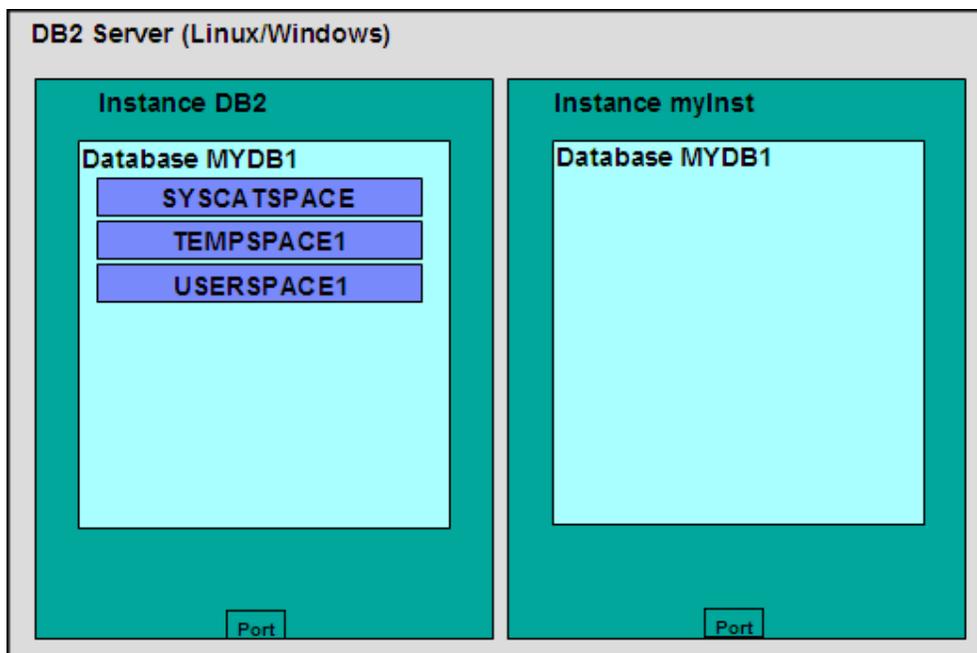


Figura 4.8 –Tablespaces creados de forma predeterminada cuando una base de datos es creada

El tablespace SYSCATSPACE contiene el catálogo de tablas. El Catálogo también es conocido en otros manejadores de bases de datos relacionales como el diccionario de datos. Esto contiene básicamente el sistema de información que no debería ser modificado o borrado; de otra forma, la base de datos no trabajaría correctamente. El tablespace TEMPSPACE1 es usado por DB2 cuando necesita espacio adicional para realizar una operación como un ordenamiento. El tablespace USERSPACE1 es usado normalmente para almacenar tablas correspondientes al usuario si es que no se especifica un tablespace cuando se crea una tabla.

Se pueden crear tablespaces para las tablas de una aplicación particular usando la sentencia CREATE TABLESPACE. La figura 4.9 muestra el tablespace MYTBLS1 creado dentro de la base de datos MYDB1 en la instancia DB2. Al crear un tablespace, se deben especificar los discos y la memoria (buffer pool) a usar. Por lo tanto, si es el caso de una tabla que usas frecuentemente (se la puede denominar tabla caliente o Hot table), se podrán asignar los discos más rápidos y asignar la mayoría de la memoria para el tablespace que contenga la tabla con esas características.

En la figura 4.9, se muestran dos objetos creados de forma predeterminada: Un buffer pool llamado IBMDEFAULTBP y los archivos log.

Un buffer pool es básicamente memoria cache usada por la base de datos. Puede crearse uno o más buffer pools, pero siempre debe existir un buffer pool con un tamaño de página que corresponda al tamaño de la página de un tablespace existente. Páginas y tamaño de página serán discutidos a mayor detalle en el Capítulo 6, Arquitectura DB2.

Los archivos log son usados para recuperación. Cuando se trabaja sobre una base de datos, no sólo es almacenada la información en los discos para la base de datos, sino que mientras trabajas sobre la base de datos, los archivos log almacenan todas las operaciones ejecutadas sobre los datos. Se considera a los archivos log como archivos temporales donde una operación de “auto guardado” es realizada. Los logs son discutidos con más detalle en el Capítulo 11: Respaldo y Recuperación.

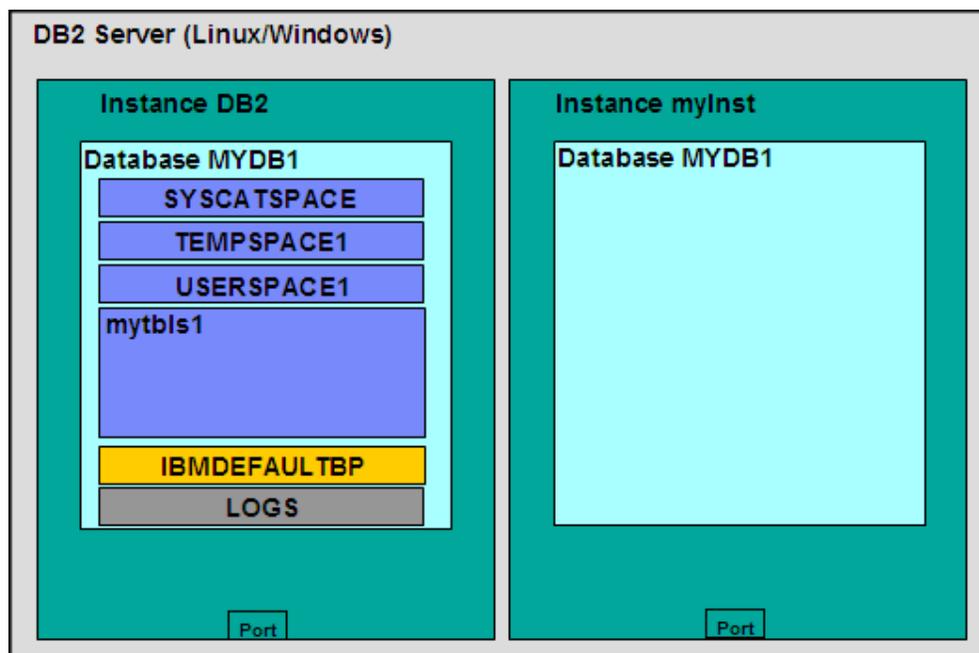


Figura 4.9 – Buffer pool y logs creados de forma predeterminada

Se discutió que las instancias son ambientes independientes, y por ello, una base de datos con el mismo nombre podría ser creada en varias instancias. Así como las instancias, las bases de datos también son unidades independientes; por lo tanto, los objetos en una base de datos no tienen relación con los objetos en otra base de datos. La figura 4.10 muestra el tablespace “mytbls1” dentro de ambas base de datos MYDB1 y la base de datos SAMPLE, dentro de la instancia DB2. Esto es válido porque las bases de datos son unidades independientes. Note que la figura 4.10 no muestra los otros objetos predeterminados de las base de datos SAMPLE debido a la escasez de espacio en la imagen.

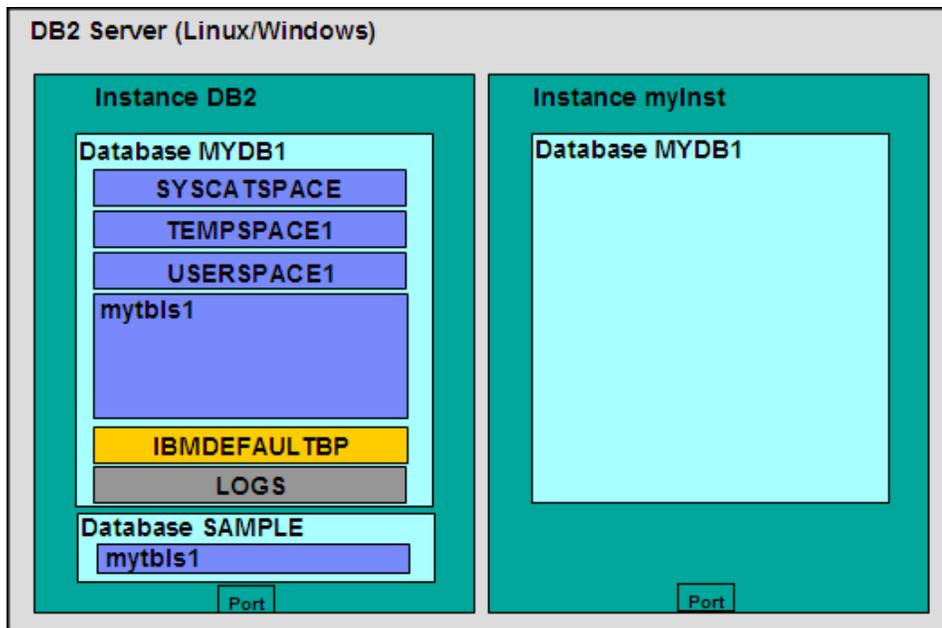


Figura 4.10 – Tablespaces con el mismo nombre en diferentes bases de datos.

Una vez creado un tablespace, se pueden crear objetos dentro del tablespace como tablas, vista e índices. Esto es ilustrado en la figura 4.11.

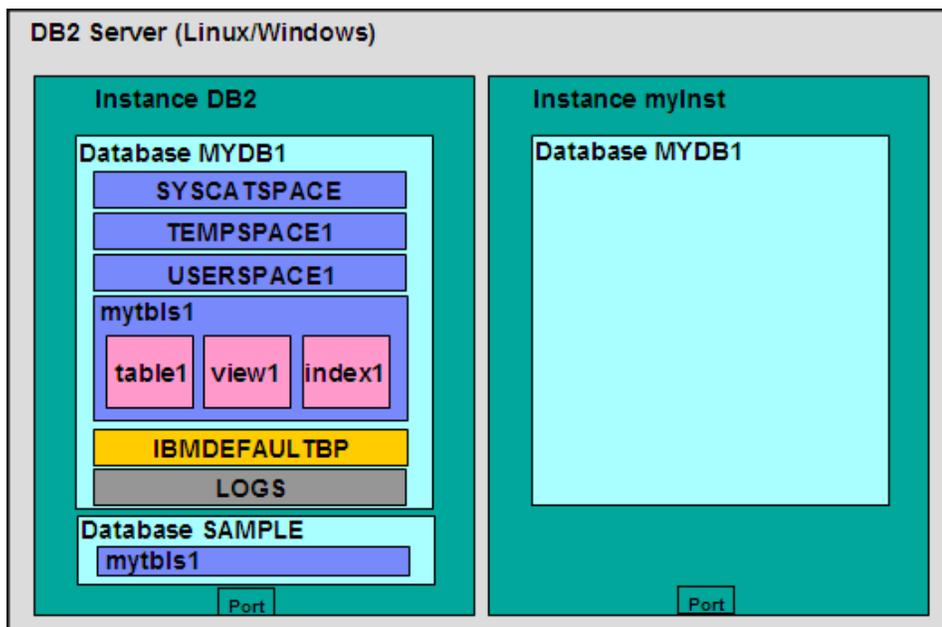


Figura 4.11 – Tablas, vistas e índices creados dentro del tablespace

QuickLab #2 – Crear una nueva base de datos

Objetivo

En este ejercicio, se deberá crear una nueva base de datos usando el Create Database wizard (asistente para creación de bases de datos) en el Centro de Control .

Procedimiento

1. Desde el panel del Control, el árbol del objetos del Centro de Control (Control Center Object Tree), haciendo clic derecho en la carpeta *All Databases (todas las bases de datos)*, seleccionar el elemento *Create Database (crea una base de datos)*, y elegir el elemento *With Automatic Maintenance (con mantenimiento automático)*. Esto lanzará el Create Database Wizard.
2. Especificar el nombre de la base de datos y la ubicación en la página *Name* del asistente. Usar los siguientes valores:

Nombre de la base de datos: EXPRESS
Dispositivo predeterminado (Windows): C:
Ruta predeterminada: (Linux): /home/db2inst1
Alias: De forma predeterminada es EXPRESS si se deja vacío
Comentario: Esto es opcional y puede ser dejado vacío

Dar clic en el botón *Next* (siguiente) para continuar con el asistente.
3. En la página *Specify where to store your data* (especifica dónde almacenarás tus datos), no hacer cambios, y hacer clic en *Next*.
4. En la página *Select your maintenance strategy* (selecciona tu estrategia de mantenimiento), dejar los valores predeterminados (*Yes, I can specify an offline ...*), y dar clic en *Next*.
5. Especificar la ventana de tiempo de mantenimiento fuera de línea en la página *Timing* del asistente. Especificar dos o más horas a la semana en las que DB2 pueda realizar tareas de mantenimiento automáticamente para preservar la salud de la base de datos. Por ahora, configura la ventana para iniciar a la 1AM cada Lunes hasta el jueves por períodos de 6 horas. Dar clic en el botón *Next*.
6. Configurar la notificación en la página *Mail Server* (servidor de correo) del asistente. DB2 puede enviar automáticamente un correo electrónico o una página si hay un problema o se detectan condiciones de riesgo. Si se deseara configurar esto, indicar un servidor SMTP para que DB2 envíe un correo electrónico. Para este ejercicio no tenemos servidor SMTP, así que dejar vacía esa opción y dar clic en *Next*.
7. Revisa las opciones seleccionadas en la página *Summary* (resumen) del asistente. Da clic en el botón *Finish* para empezar la creación de la base de datos. Usualmente, la creación de la base de datos toma unos cuantos minutos, tiempo durante el cual se despliega un indicador del progreso.

4.1 Configuración DB2

Los parámetros de DB2 pueden ser configurados usando el Configuration Advisor Tool. Para acceder al Configuration Advisor a través el Centro de Control, da clic derecho sobre la base de datos y elegir "Configuration Advisor". Basado en las respuestas que des a algunas preguntas acerca de los recursos de tu sistema y carga de trabajo, el Configuration Advisor proveerá una lista de parámetros de DB2 que podrían ser modificados con valores sugeridos para cada cual. Si quieres más detalle sobre la configuración DB2, continua leyendo; de otra forma, usa el Configuration Advisor y estarás listo para trabajar con DB2

Un servidor DB2 se puede configurar en cuatro diferentes niveles:

- Variables de ambiente
- Archivo de configuración del manejador de base de datos (dbm cfg)
- Archivo de configuración de la base de datos (db cfg)
- Registro de perfil DB2

Esto también se muestra en la figura 4.12. En la figura, notar donde se ubican cada uno de los niveles anteriores. Por ejemplo, las variables de ambiente son un conjunto a nivel del sistema operativo del server, mientras los parámetros de archivos de configuración del manejador de base de base de datos son un conjunto a nivel de instancia. Los parámetros de configuración de la base de datos son un conjunto a nivel de la base de datos, y el registro de perfiles de DB2 es un conjunto que puede estar a nivel de sistema operativo o de instancia.

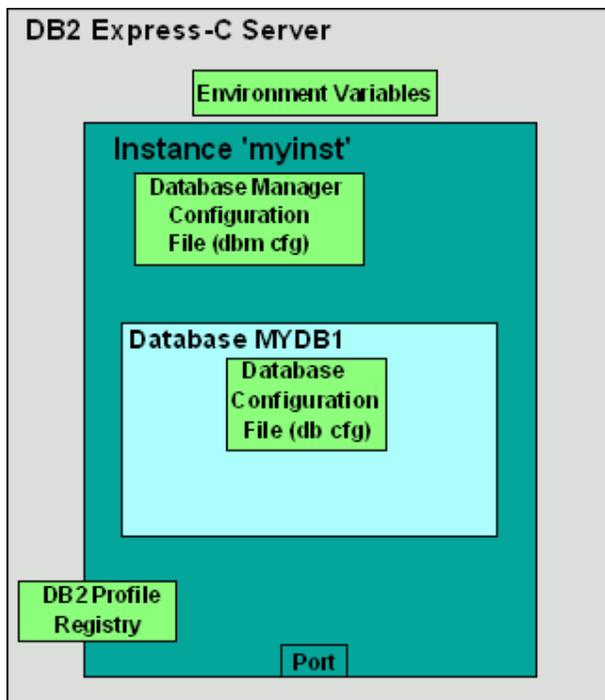


Figura 4.12 – Configuración DB2**4.1.1 Variables de ambiente**

Las variables de ambiente son un conjunto de variables a nivel de sistema operativo. Una variable de ambiente es DB2INSTANCE. Esta variable indica la instancia activa con la cual se está trabajando, y para la cual pueden usarse comando DB2. Por ejemplo, para fijar la instancia activa en "myinst" en Windows, puedes ejecutar este comando de sistema operativo: `set db2instance=myinst`

4.1.2 Archivo de configuración del manejador de base de datos (dbm cfg)

El archivo de configuración del manejador de base de datos (cfg dbm) incluye los parámetros que afectan la instancia y todas las bases de datos contenidas. El archivo de configuración del manejador de base de datos se puede ver o modificar usando la línea de comando, o a través del DB2 Control Center.

Para trabajar con el DBM CFG desde El Centro de Control, selecciona el objeto instancia de la carpeta de instancia en El Centro de Control, da clic derecho para mostrar el menú emergente y selecciona Configure Parameters (configura parámetros). Esto se muestra en la figura 4.13.

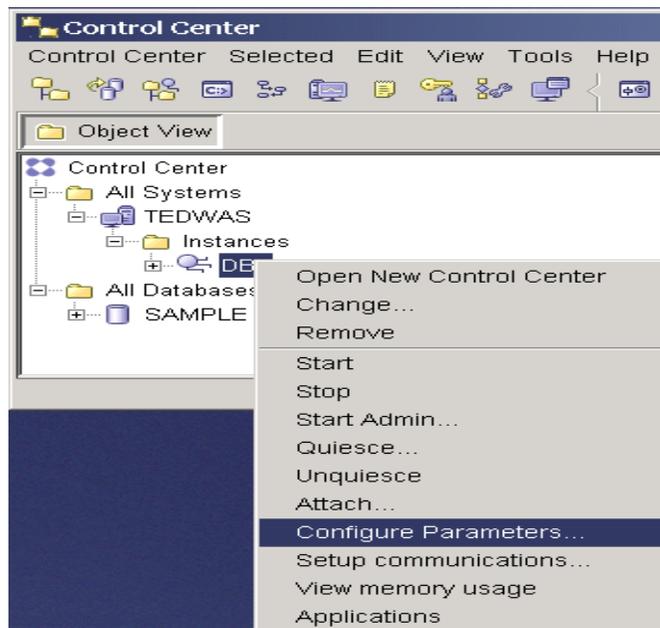


Figura 4.13 – Configuración del dbm cfg en El Centro de Control .

Después de elegir Configure Parameters, la pantalla mostrada en la figura 4.14 será desplegada con la lista de los parámetros del cfg dbm.

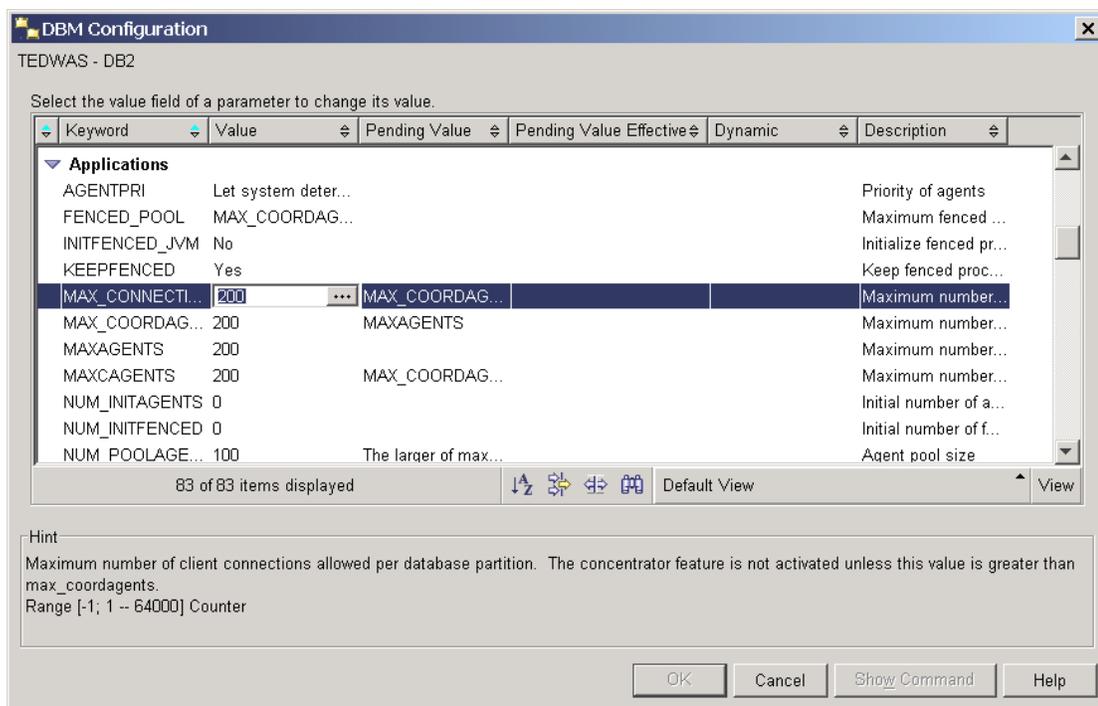


Figura 4.14 –El diálogo dbm cfg

Muchos parámetros cambian inmediatamente su valor pues el efecto de la modificación es inmediato; sin embargo, los cambios a algunos parámetros pueden requerir detener e iniciar la instancia. Desde la línea de comandos, esto se puede hacer usando los comandos `db2stop` y `db2start`.

Antes que una instancia pueda ser detenida, todas las aplicaciones se deben desconectar. Si quieres parar la instancia forzosamente, puedes usar el comando `db2stop force`.

Una instancia también puede ser detenida a través del Centro de Control dando clic en el objeto instancia y seleccionando Stop (detener) o Start (iniciar).

La tabla 4.3 muestra algunos comandos útiles para manejar los dbm cfg desde la línea de comandos.

Comando	Descripción
<code>db2 get dbm cfg</code>	Devuelve información acerca de los dbm cfg
<code>db2 update dbm cfg using <parameter_name> <value></code>	Actualiza los valores de un parámetro de un dbm cfg

Tabla 4.3 - Comandos para manipular los dbm cfg

4.1.3 Archivo de configuración de la base de datos (db cfg)

El archivo de configuración de la base de datos (db cfg) incluye parámetros que afectan a una base de datos en particular. El archivo de configuración puede ser visto o modificado usando la línea de comandos, o a través del DB2 Control Center.

Para trabajar con el DB CFG en el Centro de Control, seleccionar la carpeta base de datos en el Centro de Control, dar clic derecho para desplegar el menú y seleccionar Configure Parameters. Esto se muestra en la figura 4.15.

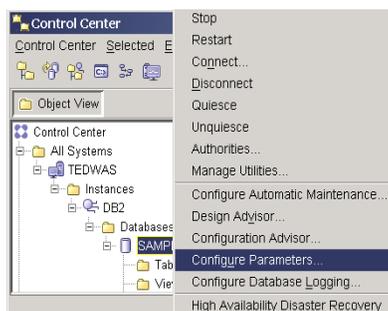


Figura 4.15 – Configurando el db cfg en El Centro de Control.

Después de elegir Configure Parameters, la pantalla mostrada en la figura 4.16 será desplegada con la lista de parámetros db cfg.

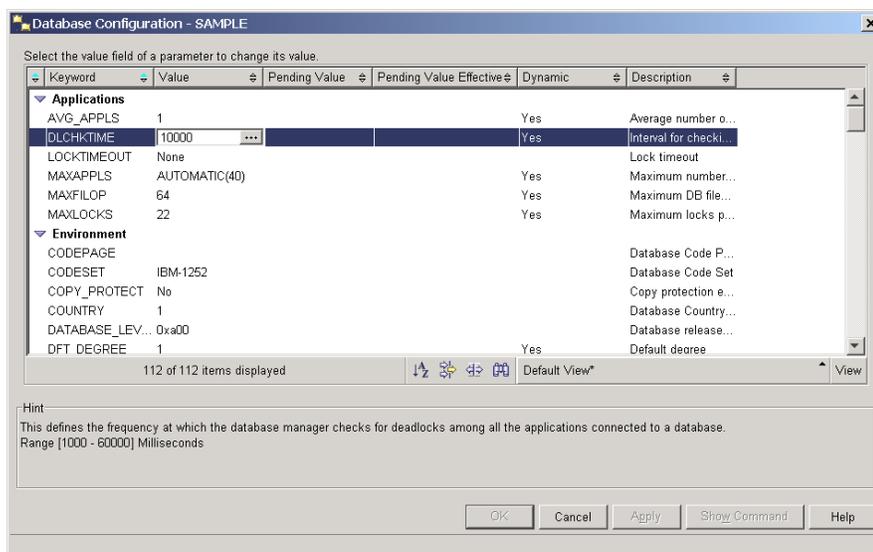


Figura 4.16 – Los parámetros disponibles en db cfg

La tabla 4.4 muestra algunos comandos útiles para manejar el db cfg desde la línea de comandos.

Comando	Descripción
<code>get db cfg for <database_name></code>	Devuelve información acerca del db cfg para una base de datos dada
<code>update db cfg for <database_name> using <parameter_name> <value></code>	Actualiza el valor de un parámetro db cfg

Tabla 4.4 - Comandos para manipular el db cfg

4.1.4 Registro de perfil DB2

Las variables de registro de perfil DB2 incluyen parámetros que pueden ser específicos de una plataforma y pueden ser asignados globalmente (afectando todas las instancias), o a nivel de instancia (afectando una instancia en particular).

La tabla 4.5 muestra algunos comandos útiles para manipular el registro del perfil DB2

Comando	Descripción
<code>db2set -all</code>	Lista todas las variables de registro de perfil DB2 que actualmente están asignadas

<code>db2set -lr</code>	Lista todas las variables de registro de perfil DB2
<code>db2set <parameter>=<value></code>	Asigna un parámetro con un valor dado

Tabla 4.5 - Comandos para manipular registro de perfil DB2

La tabla 4.6 muestra algunos de las variables más comúnmente usadas de registro DB2

Variable de Registro	Descripción
DB2COMM	Especifica los manejadores de comunicación que fueron iniciados cuando el manejador de base de datos inició.
DB2_EXTSECURITY	En Windows, previene accesos no autorizados a DB2 por bloqueo del sistema de archivos de DB2
DB2_COPY_NAME	Almacena el nombre de la copia actualmente en uso de DB2. Para cambiar a una copia instalada diferente de DB2, ejecuta el comando <code>installpath\bin\db2envvars.bat</code> . Esta variable no puede ser utilizada para este propósito

Tabla 4.6 – Variables comúnmente usadas del registro de perfil DB2

Por ejemplo, para permitir comunicación usando TCP/IP, asigna a la variable de registro DB2COMM el valor TCPIP como se muestra abajo:

```
db2set db2comm=tcPIP
```

4.2 El Servidor de Administración DB2 (DB2 Administration Server)

El DB2 Administration Server (DAS) es un proceso 'demonio' del sistema operativo que se ejecuta en el servidor DB2 para permitir a clientes remotos administrar gráficamente el servidor DB2. Sólo hay uno por cada computadora física como se muestra en la figura 4.16.

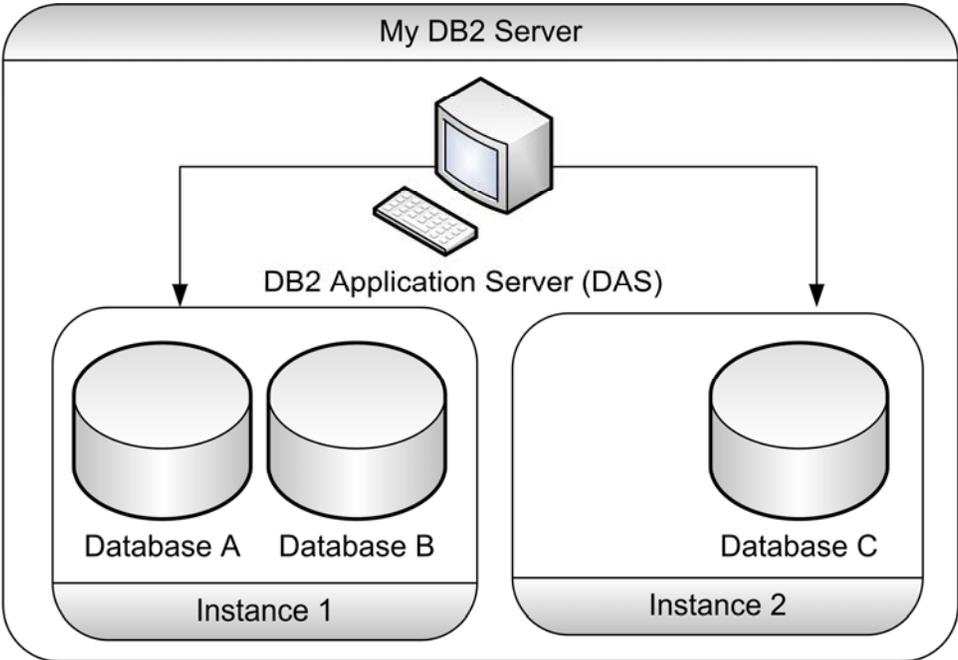


Figure 4.16 –El DB2 Administration Server (DAS)

QuickLab #3 – Trabajando con instancias, bases de datos y configuración

Objetivo

En este ejercicio, se creará una nueva instancia, una base de datos, y se cambiarán los parámetros de configuración en un servidor DB2 en Windows. Lo puedes hacer desde El Centro de Control o la Ventana de Comandos (Command Window). En este ejercicio, proveeremos las instrucciones como si fueran ejecutadas desde la Ventana de Comandos.

Procedimiento

1. Desde la Ventana de Comandos, crear una nueva instancia llamada *newinst*
`db2icrt newinst`
2. En la nueva instancia *newinst*, crear una base de datos *newdb* con los valores de defecto.
`set db2instance=newinst`
`db2start`
`db2 create database newdb`
3. Listar todas las instancias en tu servidor
`db2ilist`
4. Cambiar la instancia a *DB2* y con el siguiente comando, asegurarse que se haya modificado
`set db2instance=db2`
`db2 get instance`
5. Cambiar el parámetro `dbm cfg FEDERATED` al valor YES en lugar de NO y se verifica que el cambio haya ocurrido.
`db2 update dbm cfg using FEDERATED YES`
`db2 force applications all`
`db2 terminate`
`db2stop`
`db2start`
`db2 get dbm cfg`
6. Conectarse a la base de datos *SAMPLE* con el `userID/psw` con el que se haya ingresado al sistema operativo
`db2 connect to sample user <userID> using <psw>`

7. Revisar cuántas aplicaciones están ejecutándose en tu instancia actual
`db2 list applications show detail`
8. Abrir otra DB2 Ventana de Comandos y nuevamente conectarse a la base de datos SAMPLE sin especificar un userID/psw. Entonces revisar cuántas conexiones existen ahora.
`db2 connect to sample`
`db2 list applications`
9. Terminar la ejecución, de manera forzada, de una de las Ventana de Comandos de DB2
`db2 force application (<se obtiene el nombre de la aplicación a través del comando para listado de aplicaciones "db2bp.exe">)`
10. Borrar la instancia *newinst*
`db2idrop newinst`
11. Borrar y crear nuevamente el DAS, y se re inicia.
`db2admin stop`
`db2admin drop`
`db2admin create`
`db2admin start`
12. Asignar a la variable de Registro DB2COMM el valor tcpip y npipe a la instancia
`db2set db2comm=tcpip,npipe`
`db2stop`
`db2start`
13. Elimina el valor asignado a la variable de registro DB2COMM
`db2set db2comm=`
`db2stop`
`db2start`
14. Verificar el valor actual del parámetro db cfg LOGSECOND, y entonces cambiarlo a un valor de 5 y verificar el nuevo valor
`db2 connect to sample`
`db2 get db cfg`
`db2 update db cfg using LOGSECOND 5`
`db2 get db cfg`

5

Capítulo 5 – Herramientas de DB2

En este capítulo, describiremos algunas de las herramientas que puedes usar con DB2. La elipse roja en la figura 5.1 muestra el área en la que se centra este capítulo.

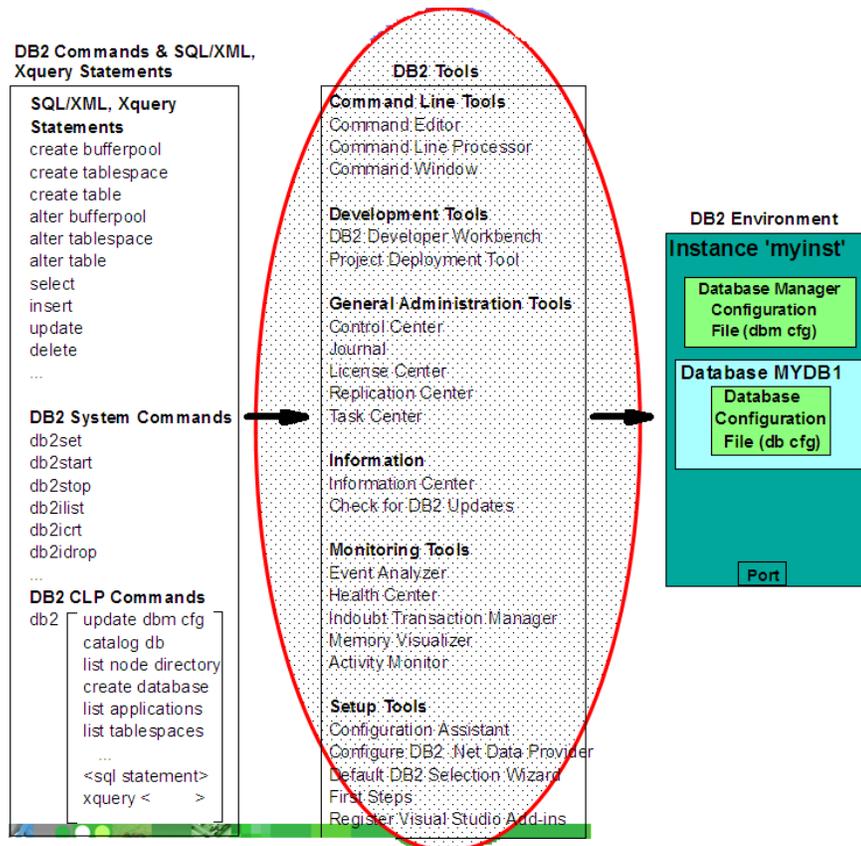


Figura 5.1 – La gran imagen de DB2: herramientas DB2

Nota:

Ver las presentaciones en video acerca de las herramientas DB2 y de scripting en estos links:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4202>

<http://www.channeldb2.com/video/video/show?id=807741:Video:4182>

La figura 5.2 lista todas las herramientas disponibles en el menú de inicio de DB2. La mayoría de esas herramientas son las mismas en Linux y Windows.

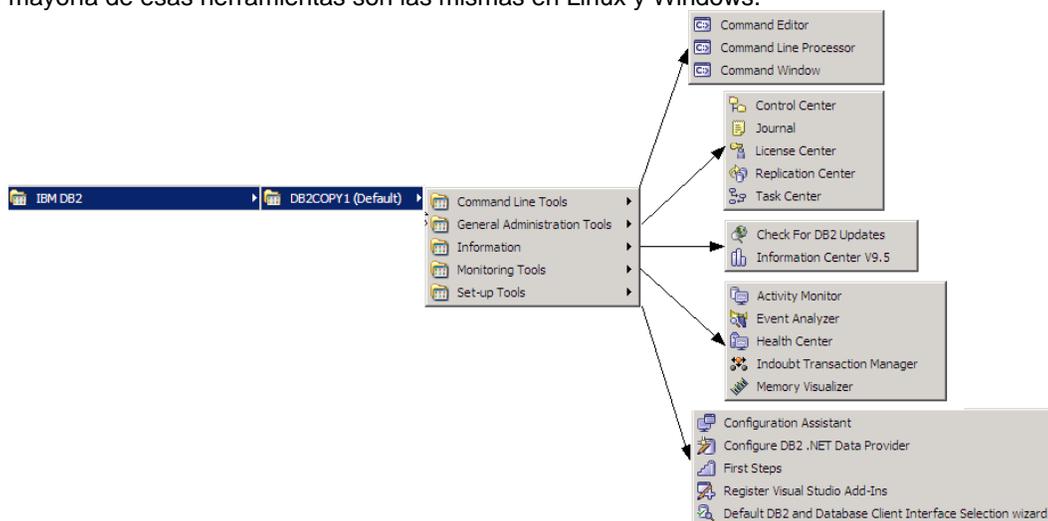


Figura 5.2 – Herramientas DB2 desde el menú de inicio de DB2

La tabla 5.1 provee una lista de comandos que pueden ser usados para iniciar algunas de las herramientas más populares tanto en Linux o Windows.

Nombre de la herramienta	Comando
Editor de Comandos	db2ce
Command Line processor	db2
Ventana de Comandos (Sólo disponible en Windows)	db2cmd
Control Center	db2cc
Task Center	db2tc

Health Center	db2hc
Configuration Assistant	db2ca
First Steps	db2fs

Tabla 5.1 – Comandos para iniciar algunas herramientas DB2

5.1 Centro de Control (Control Center)

La herramienta primaria para la administración de DB2 es el Centro de Control, ilustrado en la figura 5.3.

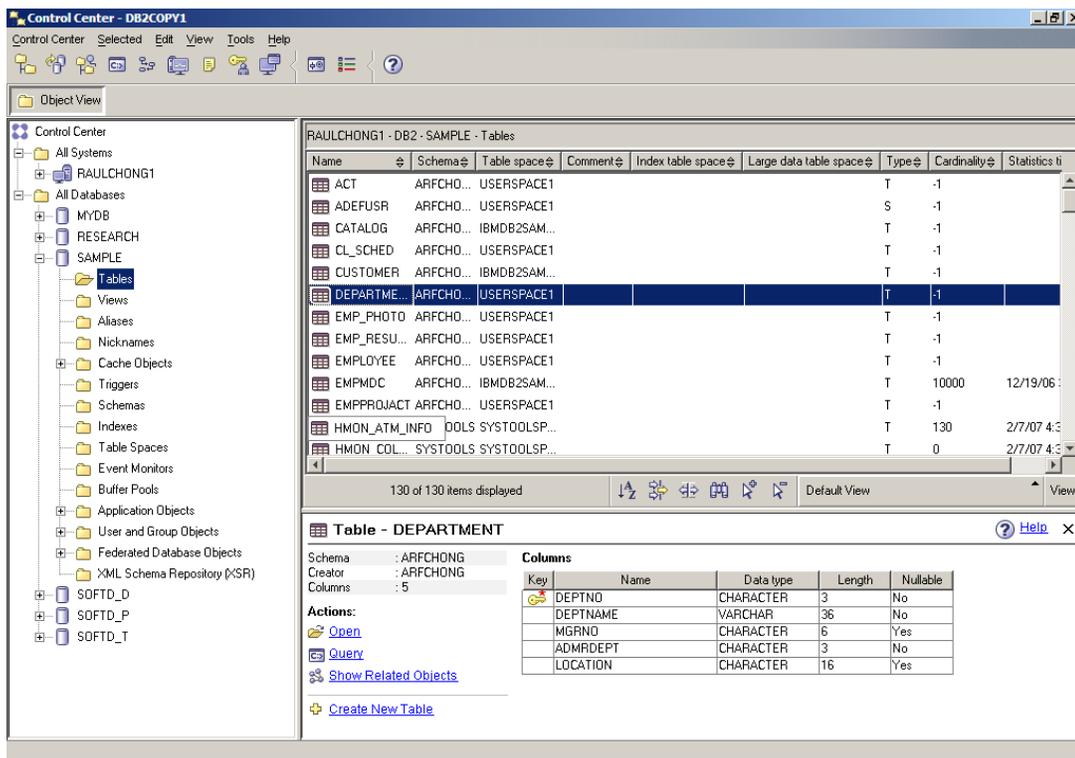


Figura 5.3 – El Centro de Control de DB2

El Centro de Control es una herramienta de administración centralizada que te permite:

- Ver el sistema, instancias, bases de datos y objetos de la base de datos;
- Crear, modificar y manejar objetos de las bases de datos;
- Lanzar otras herramientas gráficas de DB2

El panel del lado izquierdo provee una visión jerárquica de los objetos de la base de datos en el sistema, teniendo una carpeta para Tablas, Vistas, etc. Al dar doble clic sobre una carpeta (por ejemplo, como se muestra en la figura 5.3), del lado derecho listaremos todos los objetos relacionados, en este caso, todas las tablas asociadas con la base de datos SAMPLE. Si se selecciona una tabla en la parte alta del panel, en la parte baja del lado derecho del panel se proveerá información más específica acerca de esa tabla.

Hacer clic derecho sobre los diferentes objetos/carpetas en el árbol de objetos para ver el menú con las opciones aplicables a los objetos/carpetas. Por ejemplo, dar clic derecho sobre una instancia y elegir "Configure parameters" que te permite ver y actualizar el archivo de configuración del manejador de base de datos. De forma similar, si das clic derecho sobre una base de datos y se elige "Configure parameters", se podrá ver y actualizar el archivo de configuración de la base de datos. Los parámetros de ambiente y configuración DB2 se discuten a mayor detalle en el capítulo 4, El Ambiente DB2.

La primera vez que se lanza el Centro de Control, preguntará qué vista se prefiere usar. La elección de la vista determina qué tipo de opciones y objetos de la base de datos son expuestos. La figura 5.4 muestra la caja de diálogo de la vista de el Centro de Control.

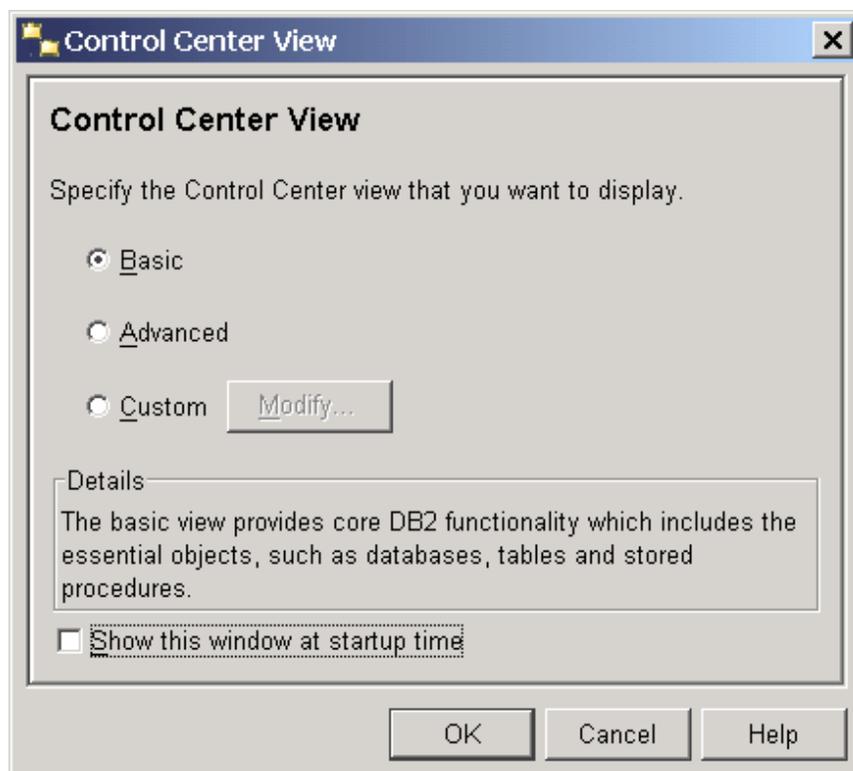


Figura 5.4 – La caja de diálogo de el Centro de Control DB2

La vista básica provee la funcionalidad elemental.
La vista avanzada muestra más opciones y mejoras.

La vista personalizada permite ajustar las mejoras específicas, opciones y objetos que se mostrarán.

Para volver a invocar el diálogo Control Center View, selecciona la opción “Customize Control Center” desde el menú Tools como se muestra en la figura 5.5.

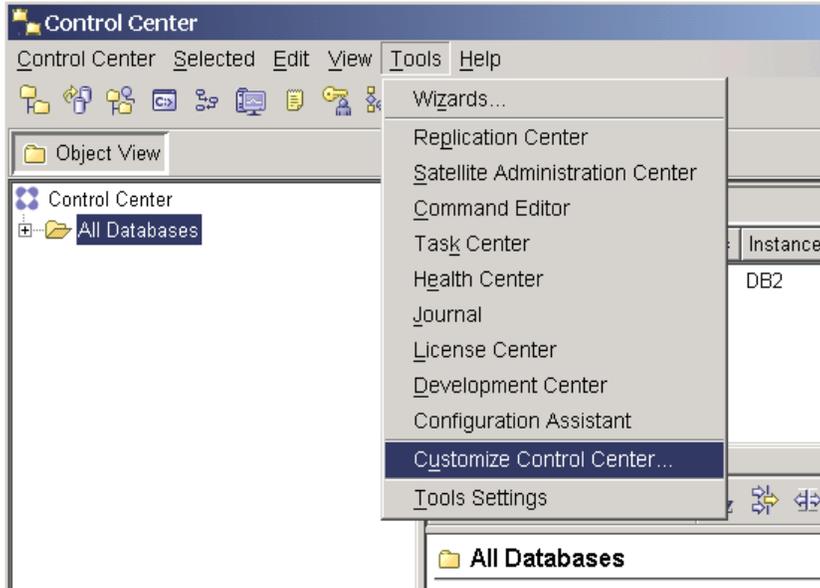


Figura 5.5 – Personalizando el Centro de Control

Lanzando el Centro de Control

Hay varias formas de lanzar el Centro de Control :

- Navegando a través del menú de inicio de Windows
- Ejecutando `db2cc` en la línea de comandos
- Dando clic en el icono de el Centro de Control  en la barra de herramientas de cualquier otra herramienta gráfica de DB2
- Desde el icono DB2 en Windows como se muestra en la figura 5.6 (Da clic sobre el icono verde DB2 y selecciona la opción Central de Control de DB2)

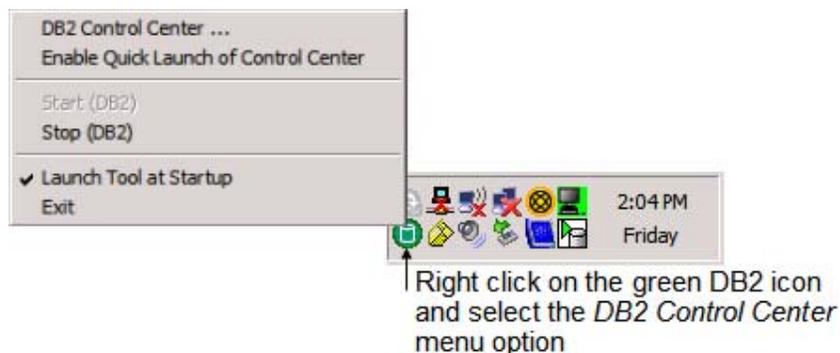


Figura 5.6 – lanzando el Centro de Control de DB2 desde la bandeja de Sistemas de Windows

5.2 El Editor de Comandos DB2

Usando el Editor de Comandos DB2 (editor de comandos), se pueden ejecutar comandos DB2, sentencias SQL y XQuery, analizar el plan de ejecución de una sentencia, y ver o actualizar resultados de consultas.

La figura 5.7 muestra el Editor de Comandos con una descripción de sus elementos.

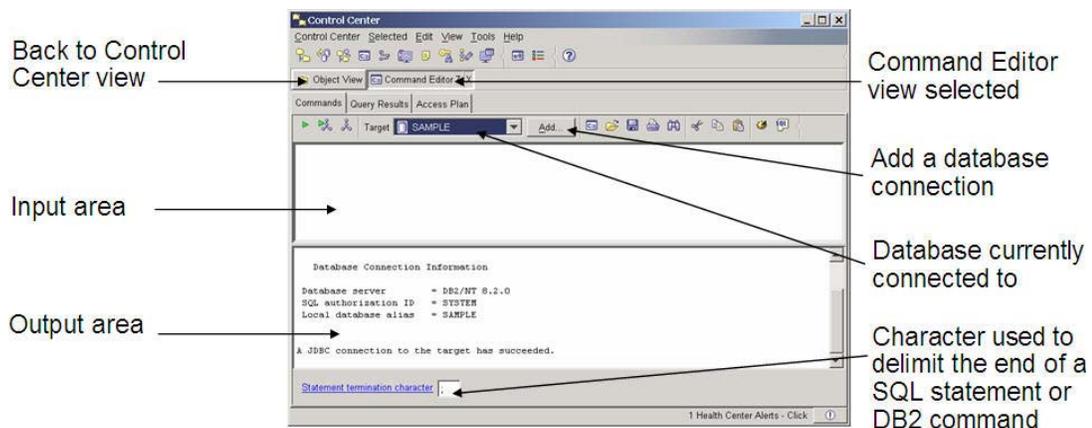


Figura 5.7 – El Editor de Comandos DB2

En el área de entrada, puedes ingresar varias sentencias, tan largas hasta que se presente el caracter de finalización. Si se presiona el botón ejecutar (ver figura 5.8), las sentencias serán ejecutadas una después de otra. Si tú seleccionas explícitamente una sentencia, sólo esa sentencia será ejecutada. Debe existir una conexión a la base de datos para devolver sentencias SQL, sin embargo, una de esas sentencias puede ser una sentencia de conexión.

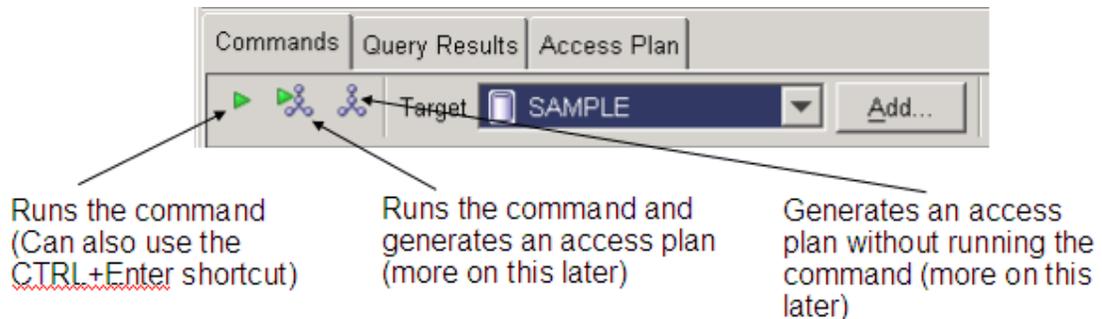


Figura 5.8 – El Editor de Comandos – Pestaña de comandos

Lanzando el Editor de Comandos

Se podrá lanzar el Editor de Comandos de varias formas:

- ▶ Desde el menú de inicio de Windows:
Start -> Programs -> IBM DB2 -> DB2COPY1 (Default) -> Command Line Tools -> Editor de Comandos
- ▶ Desde la línea de comandos, escribe `db2ce`
- ▶ Desde el menú herramientas en El Centro de Control
- ▶ Dentro de el Centro de Control
 - Dar clic derecho en el icono de la base de datos SAMPLE en el árbol de objetos del panel de el Centro de Control y selecciona el elemento del menu *Query*
 - Siempre que un objeto consultable esté seleccionado (base de datos, tabla, etc.), se puede lanzar el Editor de Comandos dando clic en la liga de *Query* que está en el panel de detalle de objetos de el Centro de Control
- ▶ Desde el Centro de Control , dar clic en el icono  de Editor de Comandos en la barra de herramientas como muestra la figura 5.9



Figura 5.9 – El icono Editor de Comandos en El Centro de Control

Agregando una conexión a una base de datos

Para agregar una conexión a una base de datos, dar clic en el botón Add (agregar). Un diálogo como en la figura 5.10 aparecerá.

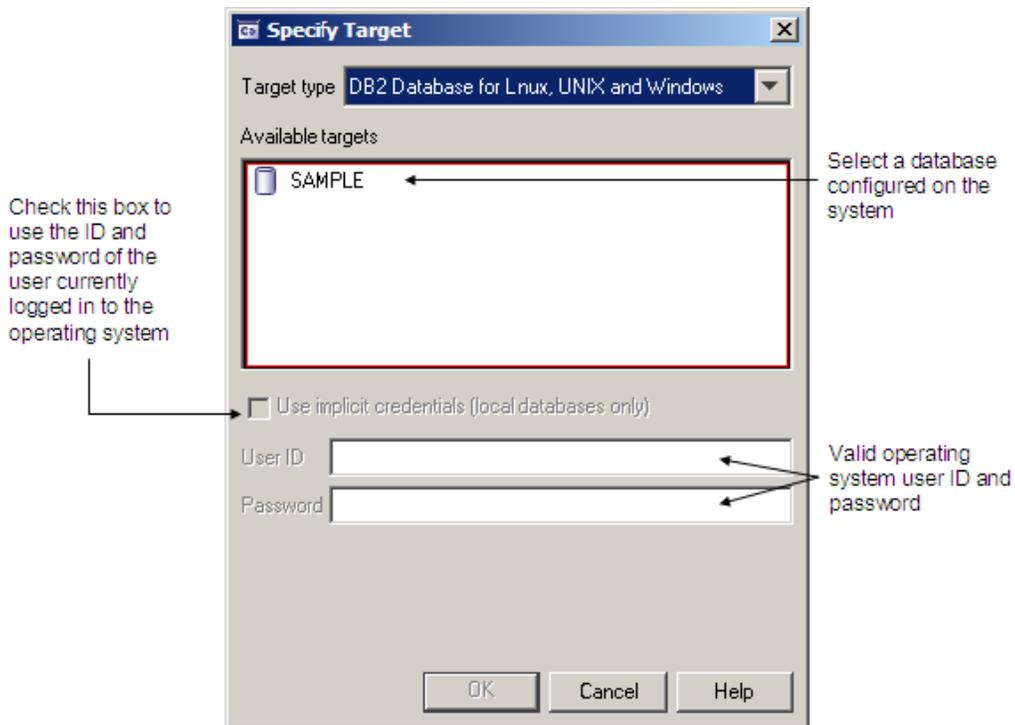


Figure 5.10 – Agregar una conexión a la base de datos

5.3 SQL Assist Wizard

Si no está familiarizado con el lenguaje SQL se podría usar un asistente para generar código SQL, el SQL Assist Wizard está disponible en el Editor de Comandos para ayudarte. Como se muestra en la figura 5.11, invócalo desde el Editor de Comandos dando clic sobre le último icono con el símbolo SQL.

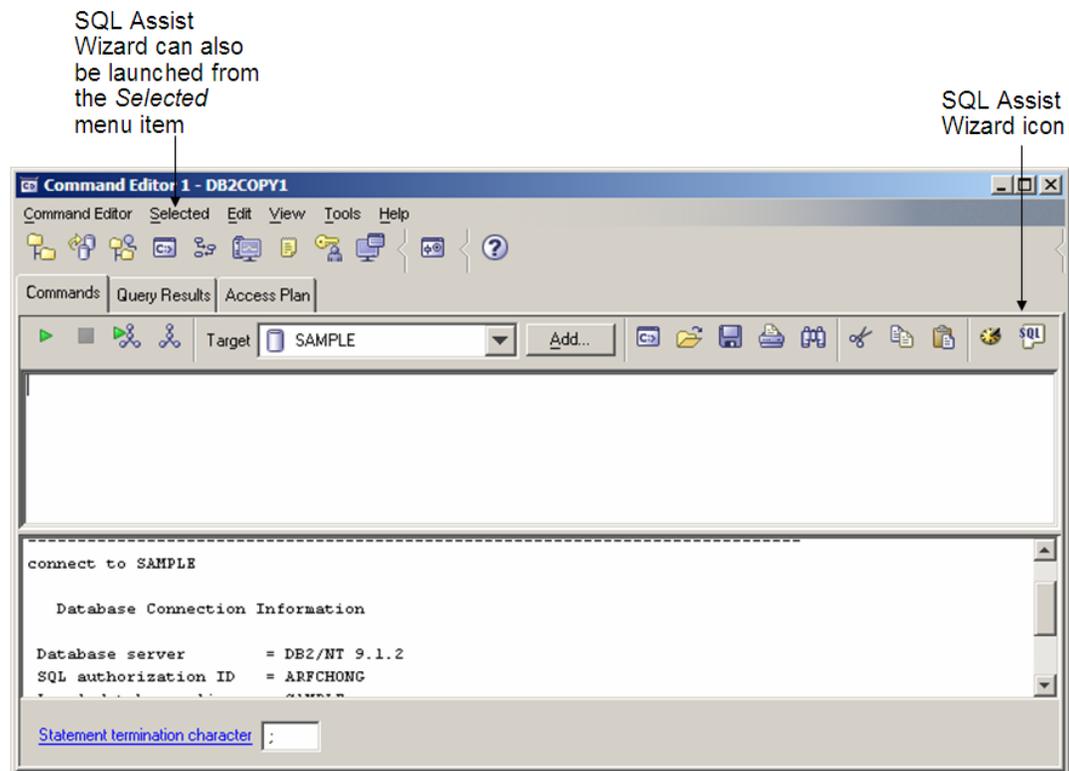


Figure 5.11 – Invocando el SQL Assist Wizard

La figura 5.12 muestra el SQL Assist Wizard. Es bastante claro su uso. Primero indica el tipo de sentencia SQL para la cual necesitas asistencia (SELECT, INSERT, UPDATE, DELETE). Dependiendo de que sentencia se haya elegido, diferentes opciones aparecerán. En la parte baja de la ventana verás cómo la sentencia SQL es construida a partir de diferentes elecciones en el asistente.

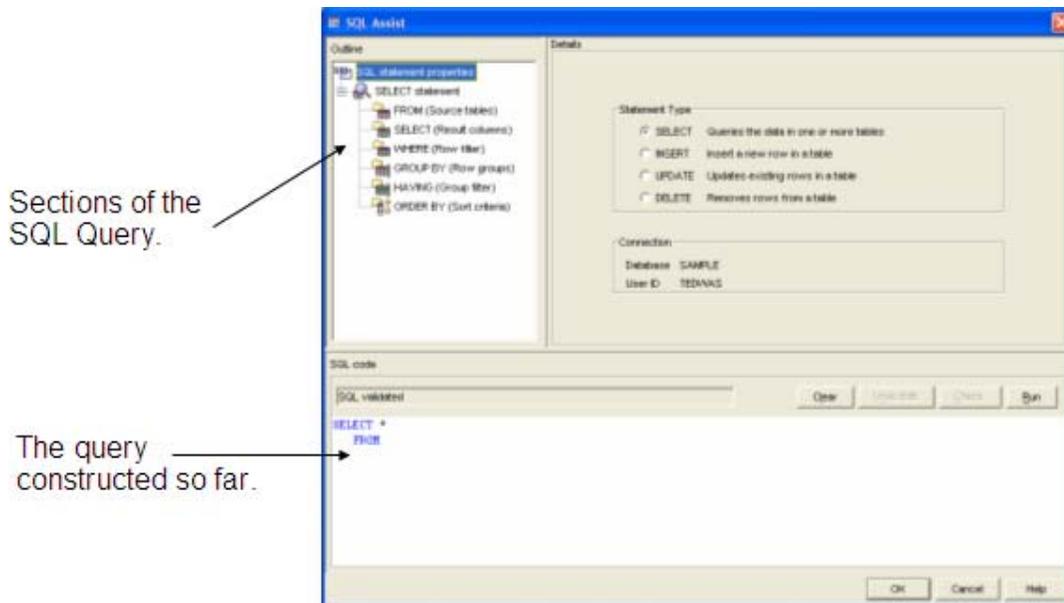


Figura 5.12 – El SQL Assist wizard

5.4 El botón Show SQL

La mayoría de las herramientas gráficas y los asistentes en DB2 permiten revisar el comando actual o la sentencia SQL que es creada como resultado de usar la herramienta para realizar tal acción. Para ver esto, da clic en el botón Show SQL en la herramienta en la que estás trabajando, como lo muestra la figura 5.13 y figura 5.14



Figura 5.13 – El botón Show SQL

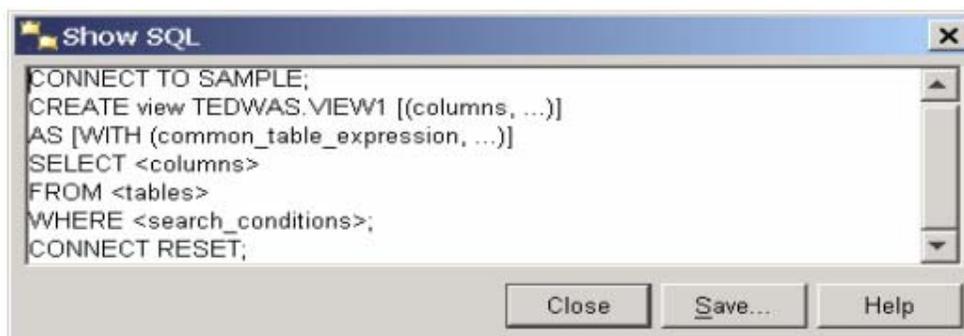


Figura 5.14 – La salida de un botón Show SQL

La posibilidad de revisar las sentencias SQL y comandos es muy útil para aprender sintaxis SQL, y para guardar comandos o sentencias para su posterior uso. También se puede construir scripts re usando esos comandos y sentencias generados.

QuickLab #4: Poblado la base de datos EXPRESS usando scripts

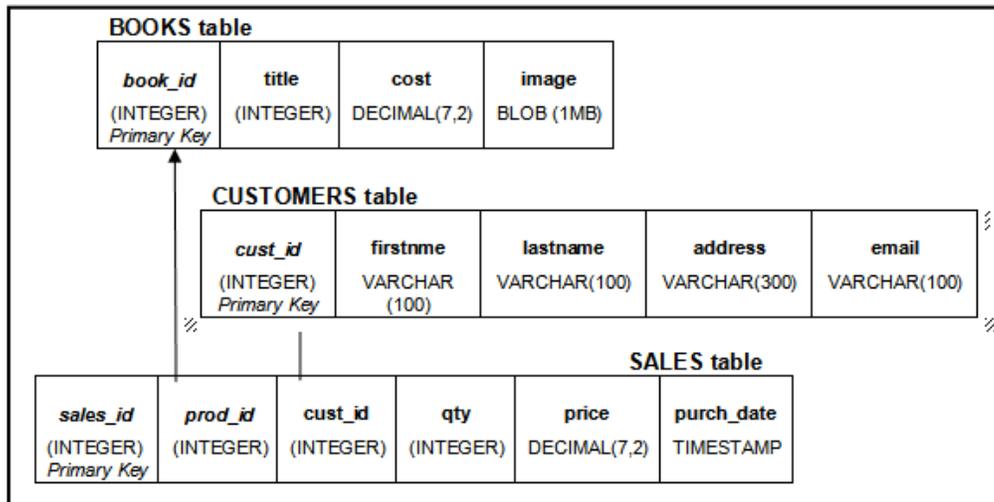
Objetivo

En este ejercicio, poblarás la base de datos express usando el Editor de Comandos y dos scripts.

Procedimiento

1. En el QuickLab #2 se creó la base de datos EXPRESS; ahora es necesario poblarla con unas cuantas tablas y algunos datos. Por conveniencia, dos scripts, llamados `quicklab4.db2` y `quicklab4.dat` han sido creados para hacer esto. El script `quicklab4.db2` contiene los comandos usados para crear las tablas y por lo tanto debe ser ejecutado primero. El script `quicklab4.dat` contiene sentencias que insertan datos en las tablas. Ambos scripts pueden ser encontrados en el archivo zip `quicklabs` que acompaña este libro. Para ejecutar esos scripts, abrir el Editor de Comandos. Asegúrate que la nueva base de datos que creaste está seleccionada en el drop-down list de la barra de herramientas. Si la nueva base de datos no aparece en la lista, agrega una conexión usando el botón *Add*.
2. Dar clic en el menu *Selected* → *Open en el* Editor de Comandos y navegar a la carpeta donde los scripts están almacenados. Selecciona el archivo `quicklab4.db2` y da clic en el botón *OK*. El contenido del archivo debería ser desplegado en el área de entrada del Editor de Comandos. Da clic en el botón *Run para ejecutar el* script. Verifica que no se encontraron errores durante la ejecución del script.
3. Repite el paso 2 para el archivo `quicklab4.dat`.

La nueva base de datos creada es para una librería muy simple de Internet. La tabla LIBROS (BOOKS en inglés) contiene toda la información sobre los libros que la tienda lleva. La tabla CLIENTES (CLIENTS en inglés) contiene la información sobre cada uno de los clientes de la tienda. Finalmente, la tabla VENTAS (SALES en inglés) contiene datos de las ventas. Siempre que un cliente compre un libro, se hace un registro en la tabla VENTAS. El diagrama de abajo muestra el diseño y la relación entre las tablas.



5.5 Uso de guiones (Scripting)

Siempre es útil poder crear archivos de guión o scripts que ejecutan varios comandos DB2 o sentencias SQL repetidamente. Por ejemplo un DBA podría requerir que se ejecuten un script dado todos los días para contar la cantidad de filas de una tabla importante.

Hay dos formas generales de scripting:

1. Scripts de SQL
2. Scripts del Sistema Operativo, habitualmente llamados shell-scripts.

5.5.1 Scripts SQL

Incluyen sentencias de consultas y comandos de la base de datos. Estos scripts son relativamente sencillos de entender y son independientes de la plataforma. Sin embargo, las variables o parámetros no son soportados.

Por ejemplo, los siguientes comandos son guardados en un archivo llamado script1.db2

```
CONNECT TO EXPRESS;
CREATE TABLE user1.mytable
    ( col1 INTEGER NOT NULL,
      col2 VARCHAR(40),
      col3 DECIMAL(9,2) );
SELECT * FROM user1.mytable FETCH FIRST 10 ROWS ONLY;
COMMIT;
```

Archivo script1.db2

En el script anterior, todas las sentencias son SQL, y cada una esta separada por un delimitador de sentencias, en este caso es el punto y coma. El nombre del archivo no necesita una extensión especial como “db2”. Se puede usar cualquier extensión.

Ejecutando Scripts SQL

Un script SQL puede ser ejecutado desde el Editor de comandos o desde la ventana de Comandos DB2 en Windows o a través de un shell Linux. Para ejecutar el script previo desde la ventana de Comandos DB2 en Windows o con un shell Linux, se puede usar el siguiente comando:

```
db2 -t -v -f script1.db2 -z script1.log
o:
db2 -tvf script1.db2 -z script1.log
```

En éste comando

- t indica que la sentencia usa la terminación de sentencia default (punto y coma)
- v indica modo 'detallado' (verbose) que causa que db2 muestre los comandos en ejecución
- f indica que el nombre de archivo indicado a continuación es el archivo script.
- z indica que se usara el nombre de archivo que sigue para contener la salida de pantalla del proceso para su posterior análisis (aunque es opcional, es recomendable).

Cuando la marca **-t** es usada y no hay delimitador especificado se asume que es el punto y coma. Puede haber situaciones donde otro delimitador es requerido. Por ejemplo un script que contenga código SQL PL que requiere un caracter de terminación diferente al punto y coma, debido a que éste es usado dentro de las definiciones de objetos SQL PL para finalizar las sentencias procedimentales.

Por ejemplo, en el archivo de script de abajo "functions.db2", que contiene las sentencias DDL para crear una función, se necesita un punto y coma al final de la sentencia SELECT requerida dentro de esta. Para finalizar la sentencia de CREATE FUNCTION estamos usando un signo de admiración (!). Si hemos usado un punto y coma, habrá un conflicto en tiempo de ejecución, resultando en un error reportado por DB2.

```
CREATE FUNCTION f1()
  SELECT ... ;
...
END!
```

Archivo functions.db2

Para informarle a DB2 que se usará un caracter de terminación de sentencia diferente, se usará la bandera **-d**, seguida por el carácter elegido:

```
db2 -td! -v -f functions.db2 -z functions.log
```

La descripción de las demás banderas pueden ser obtenidas desde la Ventana de Comandos o shell Linux usando este comando:

```
db2 list command options
```

5.5.2 Scripts de Sistema Operativo o shell scripts

Los scripts de Sistemas operativos proveen gran flexibilidad y poder, al ofrecer la posibilidad de agregar logica programática. Estos son dependientes de la plataforma y

pueden soportar parámetros y variables. Abajo se puede observar un ejemplo de script de Windows.

```
set DBPATH=c:
set DBNAME=PRODEXPR
set MEMORY=25
db2 CREATE DATABASE %DBNAME% ON %DBPATH% AUTOCONFIGURE USING
    MEM_PERCENT %MEMORY% APPLY DB AND DBM
db2 CONNECT TO %DBNAME% USER %1 USING %2
del schema.log triggers.log app_objects.log
db2 set schema user1
db2 -t -v -f schema.db2 -z schema.log
db2 -td@ -v -f triggers.db2 -z triggers.log
db2 -td@ -v -f functions.db2 -z functions.log
```

Archivo create_database.bat

Para ejecutar este script de sistema operativo desde la línea de comando, se debería hacer lo siguiente en Windows:

```
create_database.bat db2admin ibmdb2
```

En Windows al usar la extensión “bat” le indica al Sistema Operativo que es un archivo ejecutable en lotes o batch.

En Linux, es necesario cambiarle el modo al archivo para indicarle si es ejecutable usando un comando como `chmod +x`. Luego de esto se puede ejecutar de la misma manera que se indica arriba.

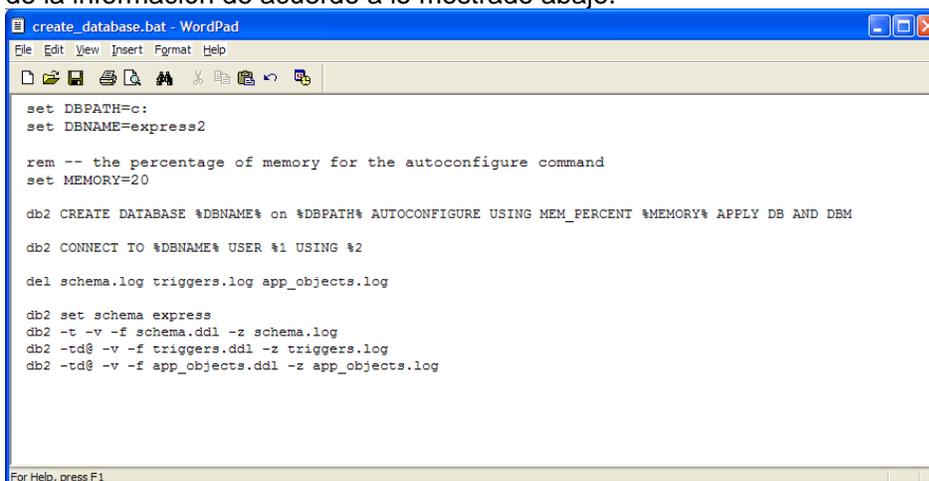
QuickLab #5: Crear un script de instalación para la Base de datos EXPRESS

Objetivo

Los Scripts son un mecanismo potente para realizar tareas repetitivas como la colección de estadísticas, backups e implementación de Bases de Datos. Los scripts de Sistema operativo tienen la ventaja de soportar parámetros, haciéndolos más flexibles. En este ejercicio, se creará un script de Sistema operativo para implementar la base de datos *EXPRESS*. El script llamará los scripts SQL de los objetos de la base de datos, previamente generado para los objetos de Base de Datos. Para ahorrar espacio este ejercicio mostrará los scripts y comandos específicos para la plataforma Windows. Si prefiere trabajar en Linux, se debe asegurar que se modifiquen las instrucciones a continuación.

Procedimiento

1. Abrir un editor de texto, tal como Notepad o Wordpad y realizar el ingreso de la información de acuerdo a lo mostrado abajo.



```
create_database.bat - WordPad
File Edit View Insert Format Help
[Icons]
set DBPATH=c:
set DBNAME=express2

rem -- the percentage of memory for the autoconfigure command
set MEMORY=20

db2 CREATE DATABASE %DBNAME% on %DBPATH% AUTOCONFIGURE USING MEM_PERCENT %MEMORY% APPLY DB AND DBM

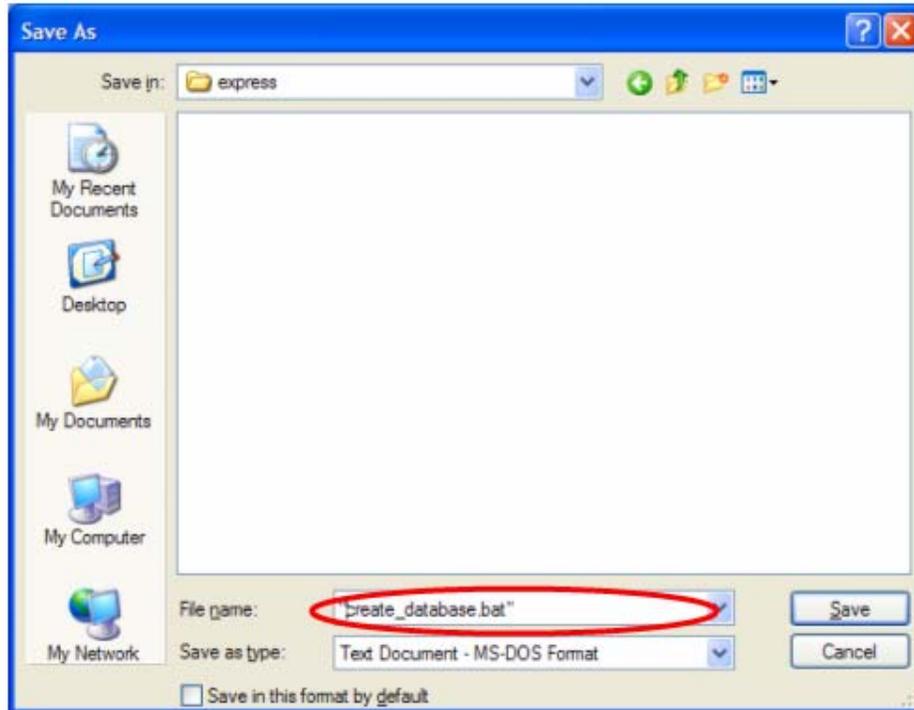
db2 CONNECT TO %DBNAME% USER %1 USING %2

del schema.log triggers.log app_objects.log

db2 set schema express
db2 -t -v -f schema.ddl -z schema.log
db2 -td@ -v -f triggers.ddl -z triggers.log
db2 -td@ -v -f app_objects.ddl -z app_objects.log

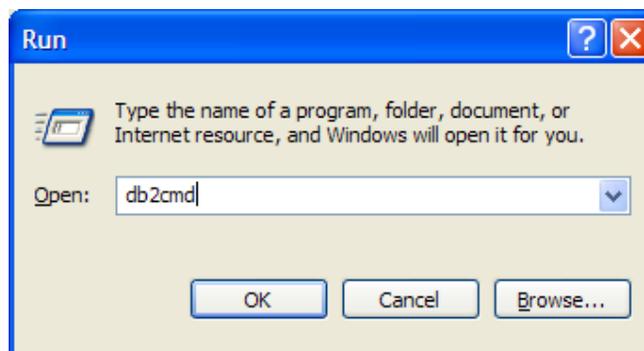
For Help, press F1
```

2. Guardar el archivo script en un directorio y llamarlo `create_database.bat`. En la ventana diálogo *Save As* se debe asegurarse de elegir la opción *MS-DOS Format* (en Wordpad). Si se guarda el archivo con diferente formato, Wordpad podría introducir caracteres invisibles que pueden causar problemas en la ejecución del script. También se debe agregar comillas en el nombre para asegurarse que Windows no agregue la extensión `.TXT` como se muestra en la figura abajo.



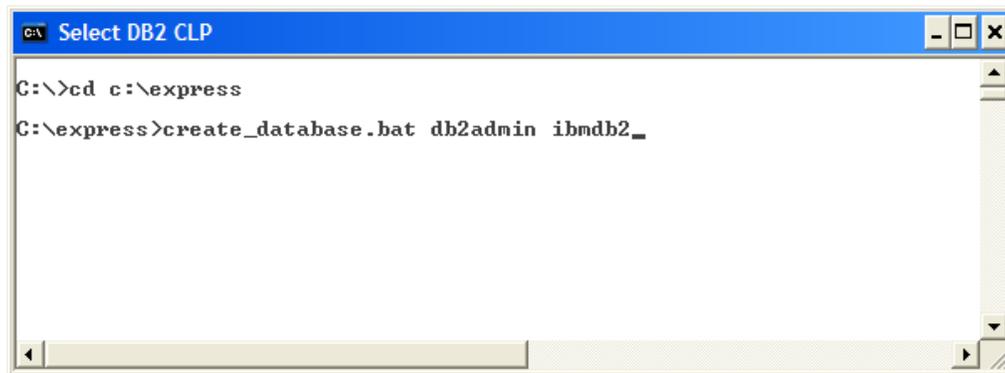
3. Para correr scripts que interactúan con DB2, se debe contar con un ambiente de línea de comando DB2. Para abrir una ventana de comandos DB2, ir a *Start > Program Files > IBM DB2 > DB2COPY1 (default) > Command Line Tools > Command Window*.

Alternativamente, se puede usar *Start > Run*, tipeando `db2cmd` y presionar *enter* según lo mostrado.



4. Luego para ejecutar el script, ingrese los siguientes comandos:

```
cd C:\express
create_database.bat db2admin ibmdb2
```



5. Tome una pausa y familiarícese con el script recientemente creado. ¿Puede entender cada línea?
6. Trate de responder las siguientes preguntas:
- ¿En que línea se establece la conexión a la base de datos?
 - ¿Qué significan %1 y %2?
 - ¿Qué hace la siguiente línea de código? ¿Dónde es usada? ¿Para qué?
SET DBPATH=C:
 - ¿Que hace la siguiente línea de código?
del schema.log, triggers.log, app_objects.log
 - ¿Que pasa cuando el script es llamado sin ningún parámetro?
 - ¿Porqué el script de SQL llamado contiene la sentencia CONNECT TO? ¿Cómo se conectan a la base de datos?

5.6 Centro de Tareas (Task Center)

La herramienta de Interfaz Gráfica de Usuario llamada Centro de Tareas (Task Center) permite crear tareas que son un conjunto de operaciones como la ejecución de comandos DB2, comandos o scripts de Sistema Operativo. Las acciones subsiguientes pueden ser realizadas si la tarea falla o es exitosa. Por ejemplo, si una tarea que involucra realizar un respaldo de una base de datos importante a las 3:00am es exitosa, un e-mail puede ser enviado al DBA para proveer esta información. Por otro lado, si la tarea falla, el Centro de Tareas puede enviar un mensaje al pager del DBA. La figura 5.15 muestra el Centro de Tareas.

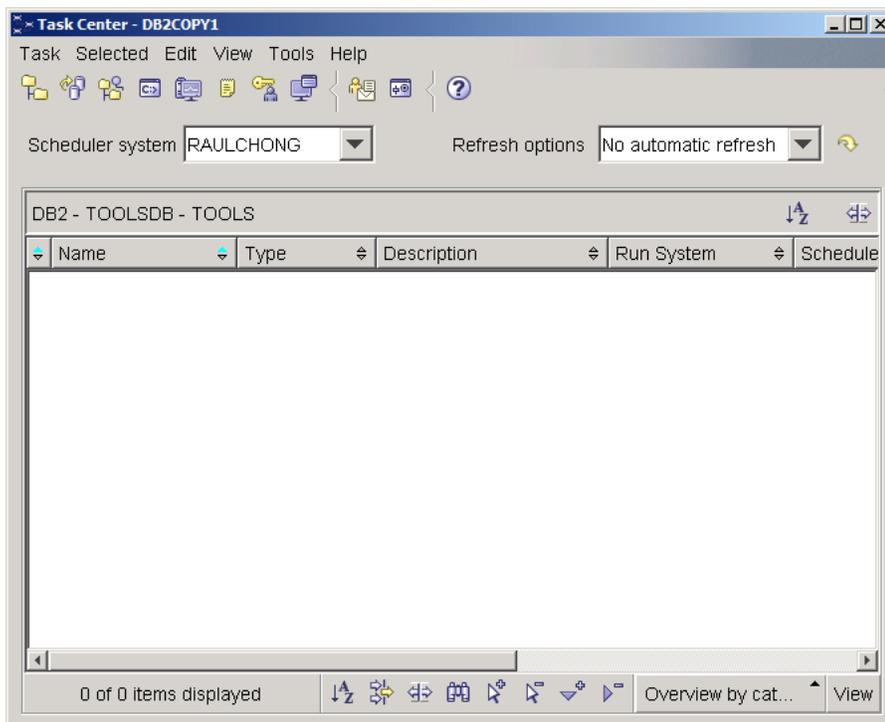


Figura 5.15 – El centro de tareas o Task Center

5.6.1 La base de datos de Catálogo de la herramienta

Todos los detalles sobre las tareas y su programación son almacenados en una Base de datos DB2 separada llamada el Catálogo de Herramientas (Tools Catalog database). Estas deben existir anticipadamente a programar las tareas. Para crear una base de datos como Catálogo de herramientas se debe usar este comando:

```
CREATE TOOLS CATALOG systools CREATE NEW DATABASE toolsdb
```

En el ejemplo anterior, `sys tools` es el nombre de esquema de todas las tablas en la base de datos y el nombre de la base de datos es `toolsdb`. Hablaremos mas sobre esquemas en el Capítulo 8, Trabajando con Objetos de Base de datos.

Iniciando el Centro de Tareas

Desde el Centro de Control haciendo clic en *Tools > Task Center*, según la Figura 5.16. Alternativamente, se puede arrancar esta ventana desde el boton Inicio de Windows: Inicio > Programas > IBM DB2 > DB2COPY1 > General Administration Tools > Task Center

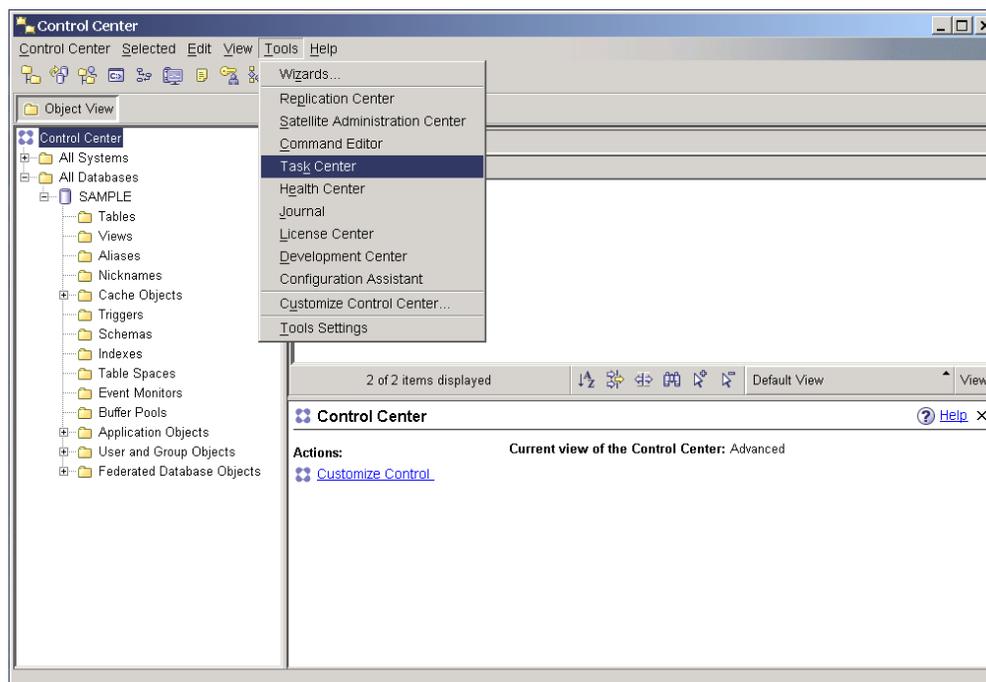


Figura 5.16 – Iniciando el Centro de Tareas

Programando con el Centro de Tareas

Cualquier tipo de script puede ser programado usando el Centro de Control (aunque no haya sido creado con la herramienta gráfica de DB2). Las tareas son ejecutadas en el momento programado desde el Sistema donde se ha creado la base de datos de catalogo. Alentamos que explore el Centro de Tareas por su cuenta, la creación de una tarea es muy sencilla.

5.7 Reporte (Journal)

La herramienta de Interfaz gráfica de reportes de DB2 le brinda al DBA un reporte de actividades en un formulario en línea. La figura 5.17 muestra el reporte y la Tabla 5.2 muestra la información que se puede obtener de él.

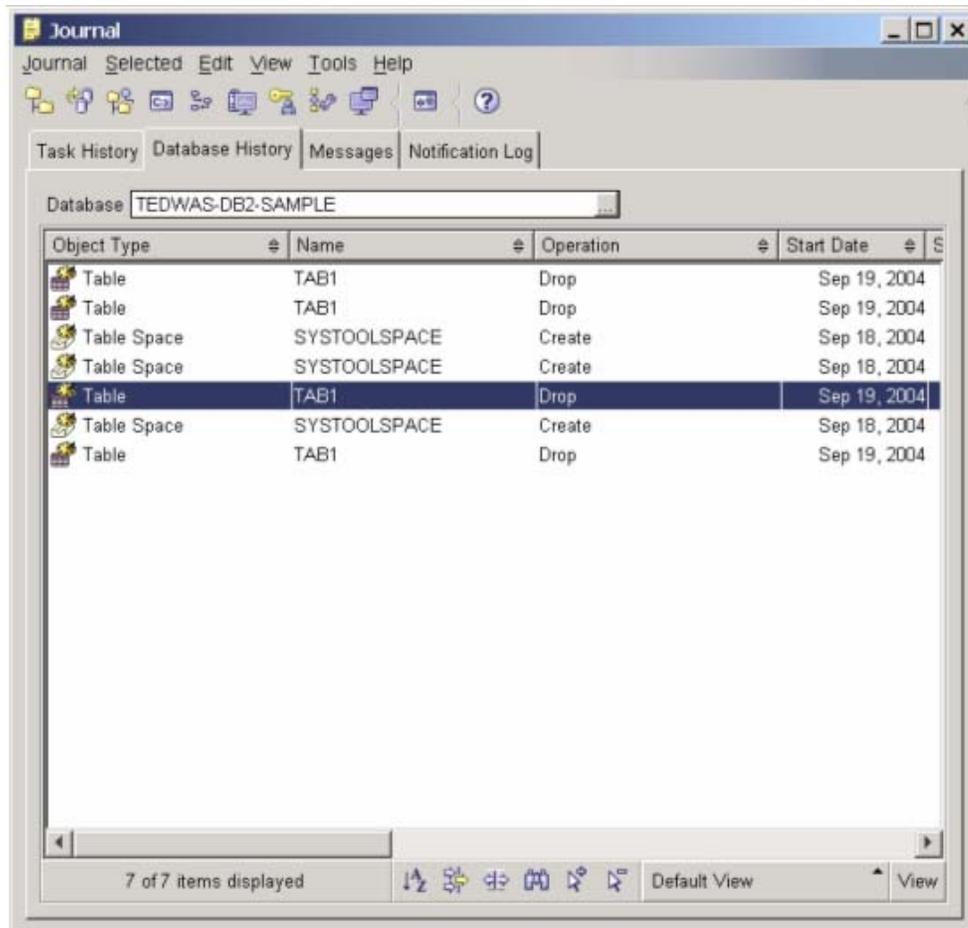


Figura 5.17 – El Reporte

Tipo de Información	Descripción
Historia de la Tarea (Task History)	Todas las tareas programadas y su estado de éxito
Historia de la Base de datos (Database History)	Un registro de las actividades ejecutadas en la base de datos (backup, restore, REORGs, etc.)
Mensajes (Messages)	La historia de los mensajes retornados por las herramientas de DB2. Es útil si se quiere recordar y comparar los mensajes anteriores o si se cierra una caja de dialogo demasiado rápido o por accidente.
Registro de notificaciones (Notification Log)	Contiene mensajes a nivel de system. Los errores críticos son registrados aquí.

Tabla 5.2 – Información disponible con la herramienta Reporte (Journal)

Iniciando el reporte

Se puede iniciar el Journal desde el Centro de control, haciendo clic en *Tools > Journal*, como se muestra en la Figure 5.18. Alternativamente, se puede arrancar esta herramienta desde el botón Inicio de Windows: *Inicio > Programas > IBM DB2 > DB2COPY1 > General Administration Tools > Journal*.

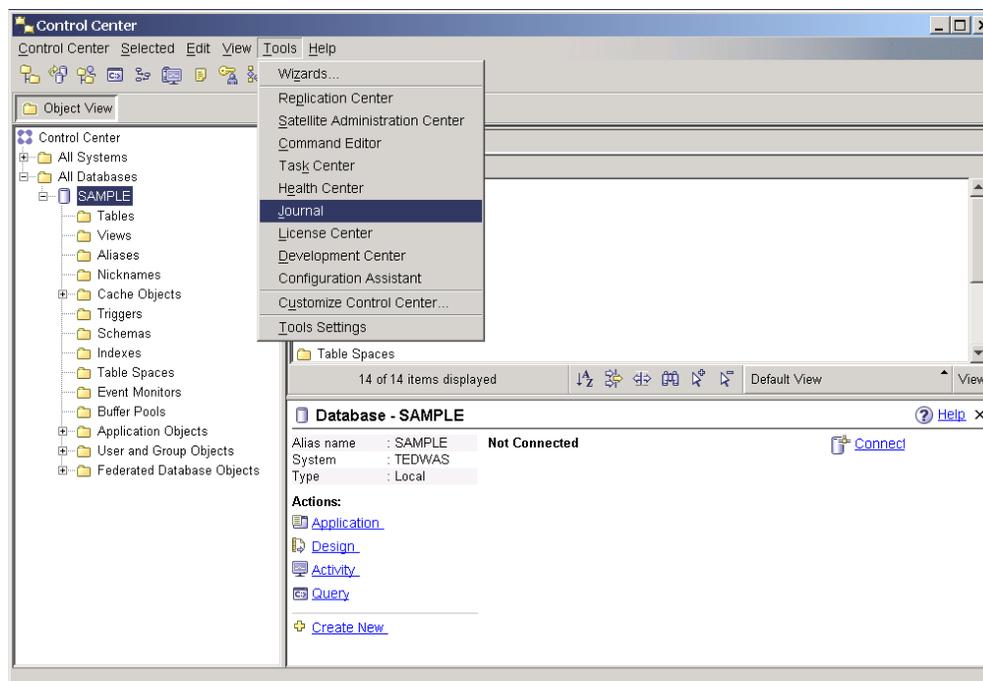


Figura 5.18 – Iniciando el reporte

5.8 Monitor de Salud (Health Monitor)

El Monitor de Salud es un agente que por defecto corre dentro del motor DB2, monitoreando todos los aspectos de la salud de una base de datos (memoria, administración de espacio, actividades automatizadas previamente definidas, etc.). Cuando algún aspecto de DB2 está operando fuera del conjunto de parámetros, una excepción es levantada y puesta bajo la atención del DBA. Hay tres tipos de estados de alerta:

- ▶ Atención: un estado no normal
- ▶ Advertencia: un estado no crítico que no requiere atención inmediata; puede indicar un sistema no óptimo
- ▶ Alarma: una condición crítica que requiere una acción inmediata

El monitor de Salud puede ser encendido o apagado usando el parámetro de configuración del administrador de base de datos HEALTH_MON.

5.8.1 Centro de Salud (Health Center)

El Centro de Salud es una herramienta gráfica para interactuar con el Monitor de Salud. Esta herramienta separa las alertas en un sistema por niveles como instancia, base de datos y Tablespace. La figura 5.19 muestra el Centro de Salud.

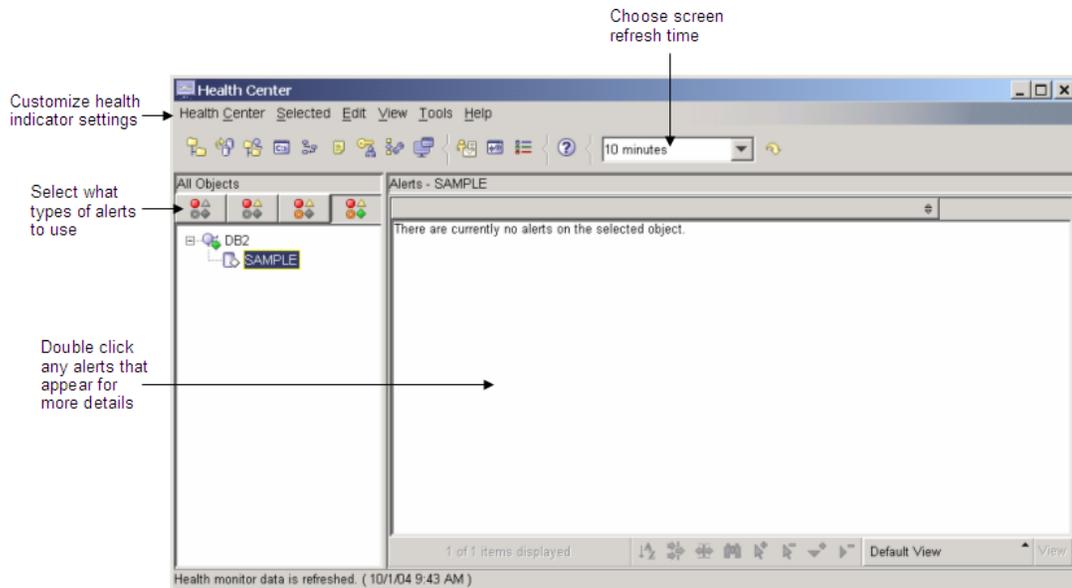


Figura 5.19 – el Centro de Salud

Iniciando el Centro de Salud

Se puede iniciar el Centro de Salud desde el Centro de comando haciendo clic en el Menu Herramientas y eligiendo el Centro de Salud. Esto es mostrado en la Figura 5.20. También se puede iniciar desde *Inicio > Programas > IBM DB2 > DB2COPY1 >Monitoring Tools > Health Center*

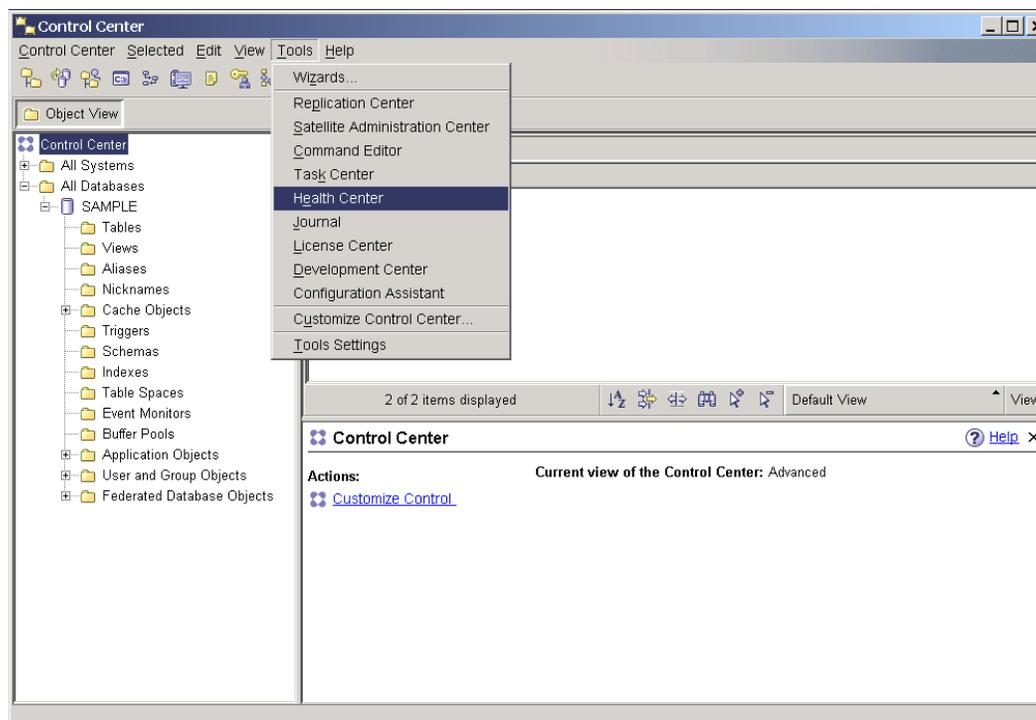


Figura 5.20 – Iniciando el Centro de Salud

Configuración de la Notificación de Alerta de Salud

Una vez que se ha iniciado el Centro de Salud, se podrá configurar la notificación de Alerta haciendo clic en el menú del *Centro de Salud* > *Configurar* > *Notificación de Alerta* como se muestra en la Figura 5.21. La notificación de Alerta permite ingresar los nombres de notificación con direcciones de e-mail o número de pagers de las personas que deben ser contactadas si se presenta un alerta.

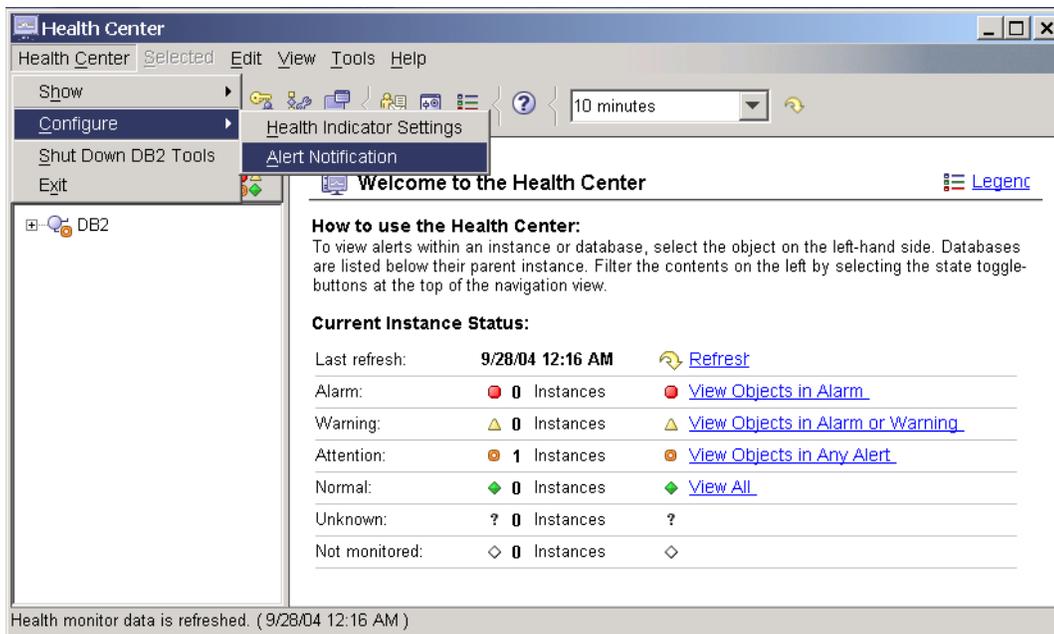


Figura 5.21 – Notificación de Alerta

PARTE II – APRENDIENDO DB2: ADMINISTRACION DE BASE DE DATOS

6

Capítulo 6 - La Arquitectura de DB2

En este capítulo se discute brevemente la arquitectura de DB2:

- El modelo de procesos
- El modelo de memoria
- El modelo de almacenamiento

Nota:

Para más información acerca de la arquitectura de DB2, se recomienda el siguiente video: <http://www.channeldb2.com/video/video/show?id=807741:Video:4482>

6.1 El Modelo de Procesos

La figura 6.1 representa el modelo de procesos de DB2. En ella, los rectángulos representan procesos, mientras que las elipses representan hilos. El proceso principal de DB2 se llama db2sysc. Existen muchos hilos para este proceso, de los cuales el primario también se llama db2sysc. Este es el hilo principal que crea a los otros. Cuando una aplicación remota intenta conectarse al servidor usando la sentencia SQL CONNECT, los observadores remotos del proceso de comunicaciones reciben dicha solicitud y contactan a un agente coordinador DB2 (db2agent). Un agente de este tipo actúa como un representante de DB2 que efectúa operaciones en su nombre. Cuando la aplicación es local, es decir, corre en el mismo servidor que DB2, los pasos son muy similares, con la diferencia que es un agente db2ipccm quien procesa la solicitud en lugar de un hilo db2tcpcm. En algunos casos, como cuando está permitido el paralelismo, un agente db2agent puede crear otros agentes, que aparecen como hilos db2agntp. Algunos otros agentes mostrados en la figura, como db2pfchr, db2loggr, db2dlock pueden ser usados para otros fines. Los procesos más comunes se describen en la tabla 6.1, los hilos más comunes, en la 6.2.

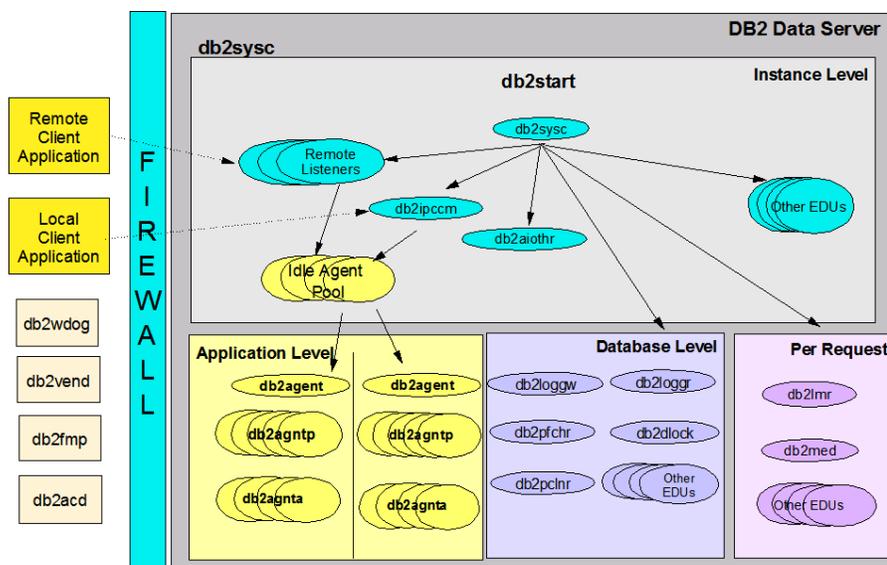


Figura 6.1 – El Modelo de Procesos de DB2

Nombre del Proceso	Descripción
db2sysc (Linux) db2syscs (Win)	El principal controlador/motor del sistema DB2. En DB2 9.5, solo existe un motor principal multihilos para toda la partición. Todas las unidades despachables por el motor (EDUs: Engine Dispatchable Units) son hilos dentro de este proceso. Sin este, el servidor no puede funcionar.
db2acd	Demonio de computación autónoma. Es usado para tareas automáticas del lado cliente, como monitoreo de salud, utilidades de mantenimiento automático y el planificador del admin. Este proceso solía llamarse db2hmon.
db2wdog	El watchdog de DB2. Es el padre de db2sysc. Se encarga de la liberación de recursos en caso el proceso principal finalice anormalmente.
db2vend	El proceso de vendedores cercados, nuevo en DB2 9.5. Todo código de terceros corre en este proceso, fuera del motor. Dichos códigos son programas ajenos a IBM que pueden interactuar con DB2; por ejemplo, el archivamiento de logs puede ser tercerizado especificando el parámetro de rutina de salida de usuario.
db2fmp	Procesos cercados que ejecutan código de usuarios fuera del firewall tanto para procedimientos almacenados como para funciones definidas por el usuario. Este proceso reemplaza tanto al db2udf como al db2dari, procesos de versiones anteriores.

Tabla 6.1 – Procesos Comunes en DB2

Nombre del Hilo	Descripción
db2sysc	El hilo controlador del sistema. Es responsable de las configuraciones iniciales y finales, así como la gestión de la instancia en curso.
db2tcpcom	Observador de comunicaciones TCP/IP.
db2agent	Agente coordinador que efectúa operaciones sobre la base de datos en nombre de aplicaciones (una por conexión cuando menos, dependiendo de si Connection Concentrator está activado).
db2agntp	Sub-agente activo creado si el valor de INTRA_PARALLEL es YES. Ejecuta operaciones sobre la base de datos en nombre de la aplicación. El trabajo de los diferentes sub-agentes es coordinado por db2agent.
db2pfchr	Pre-recogedor de datos de entrada/salida (NUM_IOSERVERS). Funciona asincrónicamente.
db2pclnr	Escritor de datos de entrada/salida (NUM_IOCLEANERS). Funciona asincrónicamente.

Tabla 6.2 – Hilos Comunes en DB2

6.2 El Modelo de Memoria

El modelo de memoria de DB2 consiste en diferentes áreas en memoria en los niveles de instancia, base de datos, aplicación y agente, como se muestra en la figura 6.2. No se detallarán cada una de las áreas de memoria en este libro, sólo serán descritas brevemente.

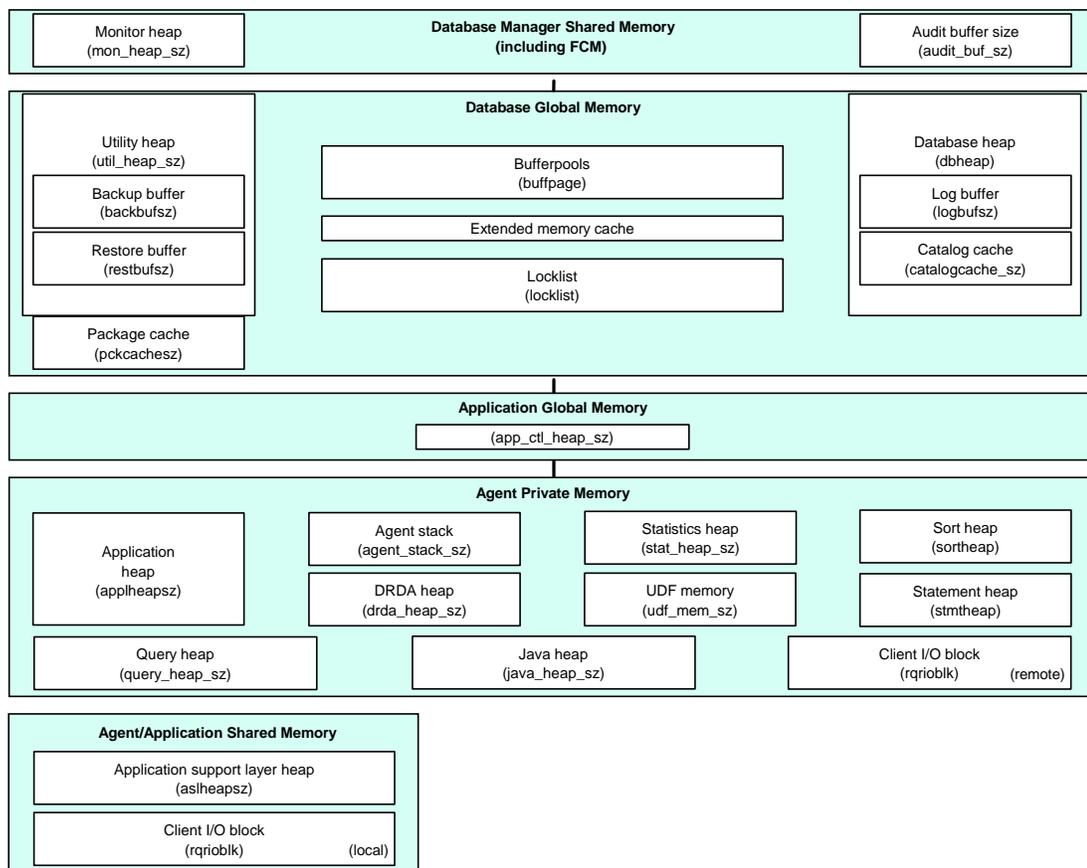


Figura 6.2 – El Modelo de Memoria de DB2

Cuando se inicia a una instancia, se reserva la memoria compartida del manejador de base de datos (Database Manager Shared Memory). Normalmente, no se requiere de mucho espacio para dicha reserva. Al momento de la primera conexión con la base de datos, el Database Global Memory (Memoria Global de Base de Datos) es reservada. En este bloque, la piscina de buffers (bufferpools) es una de las partes más importantes, especialmente para la mejora del desempeño de consultas. El tamaño de la piscina de buffers determina qué tan grande será el Database Global Memory en su totalidad.

La memoria privada de agente (Agent private memory) es aquella usada por cada agente DB2. Sin usar Connection Concentrator, cada conexión requiere un agente. Típicamente,

un agente usa entre 3 y 5 MB. Con Connection Concentrator, varias conexiones pueden usar el mismo agente, de manera que se reduce la cantidad de memoria física requerida.

6.3 El Modelo de Almacenamiento

En esta sección, analizaremos los siguientes conceptos:

- Páginas y extensiones
- Buffer pool (piscina de buffer)
- Espacio de tablas (Tablespace)

6.3.1 Páginas y Extensiones

Una página es la mínima unidad de almacenamiento en DB2. Los tamaños de página permitidos son: 4K, 8K, 16K y 32K. Una extensión es un conjunto de páginas. Trabajar con una página a la vez sería costoso en términos de desempeño, por esta razón, DB2 trabaja con varias extensiones a la vez. El tamaño de página y extensión son definidos al trabajar con buffer pools y espacios de tablas, como se verá en las siguientes secciones.

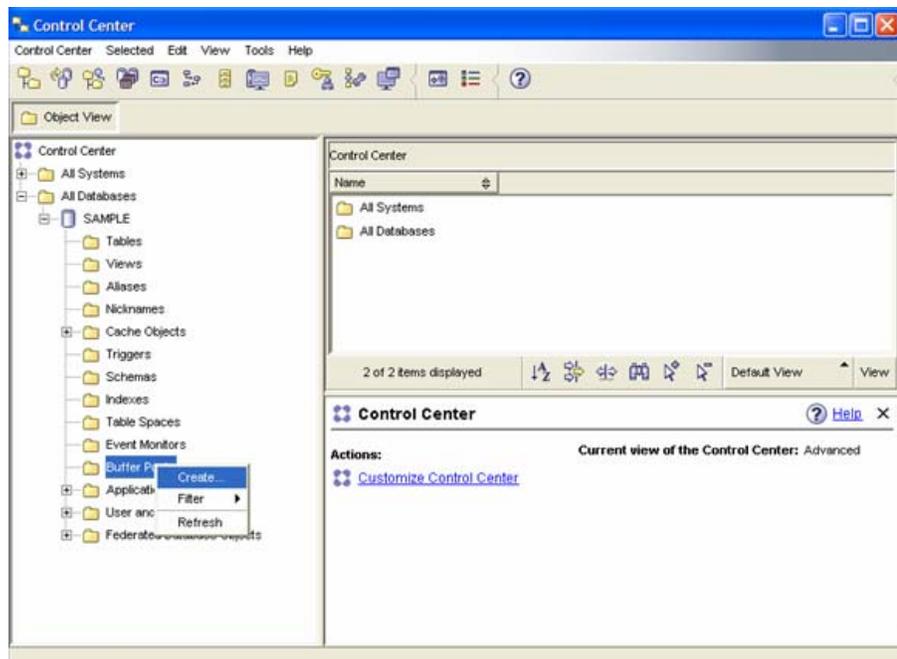
6.3.2 Buffer pools

Un buffer pool es un verdadero cache de memoria para datos de tabla e índices. Mejora el desempeño al reducir la cantidad de entradas/salidas secuenciales directas, y promueve la lectura (pre-recoger) y escritura asíncronas. En otras palabras, DB2 determina anticipadamente qué páginas serán necesitadas, las pre-recoge del disco y las coloca en el buffer pool, de tal modo que están listas para ser usadas.

Los buffer pool se alojan en unidades de memoria de 4K, 8K, 16K y 32K páginas. Por cada base de datos, deberá existir cuando menos un buffer pool, así como una para cada espacio de tabla de un tamaño de página determinado.

Creando un Buffer Pool

Para crear un buffer pool, se puede usar la sentencia CREATE BUFFERPOOL. Alternativamente, usando el Centro de Control, se puede dar clic derecho en el folder de Buffer pool de una base de datos y elegir Crear (Create), como muestra la figura 6.3.



Figure

Figura 6.3 – Creando un buffer pool

Luego de dar clic en Crear, el diálogo "Crear Buffer Pool" aparecerá, tal y como en la figura 6.4.

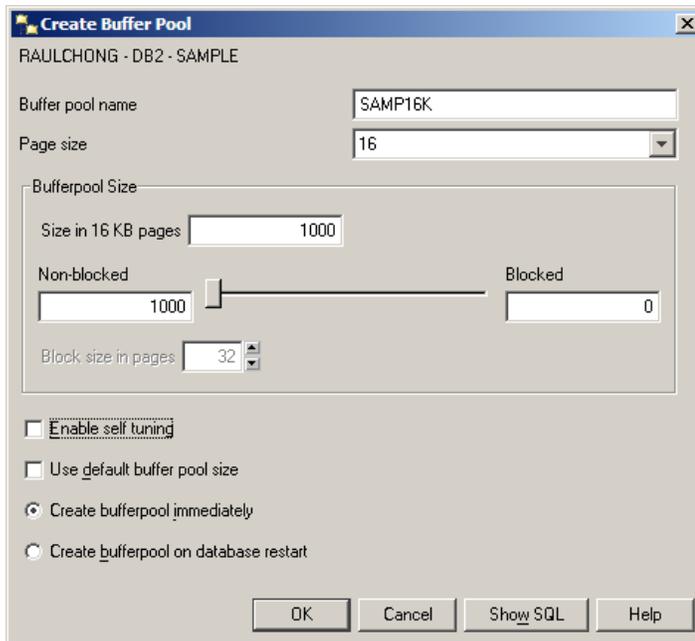


Figura 6.4 – La caja de diálogo para crear un buffer pool

La mayoría de objetos en la figura 6.4 se explican a sí mismos. Los campos "Non-blocked" y "Blocked" se refieren a la cantidad de páginas que deberían existir como no-agrupadas y agrupadas, respectivamente. Los buffer pools basados en tablas agrupadas aseguran que páginas contiguas en disco sean llevadas al buffer pool también de manera contigua a un área agrupada. Esto podría mejorar el desempeño. El número de páginas no debe ser mayor al 98% de la cantidad de páginas para el buffer pool. El especificar el valor de 0 deshabilita esta característica.

Una vez que el buffer pool ha sido creado, será mostrado en el centro de control, como muestra la figura 6.5.

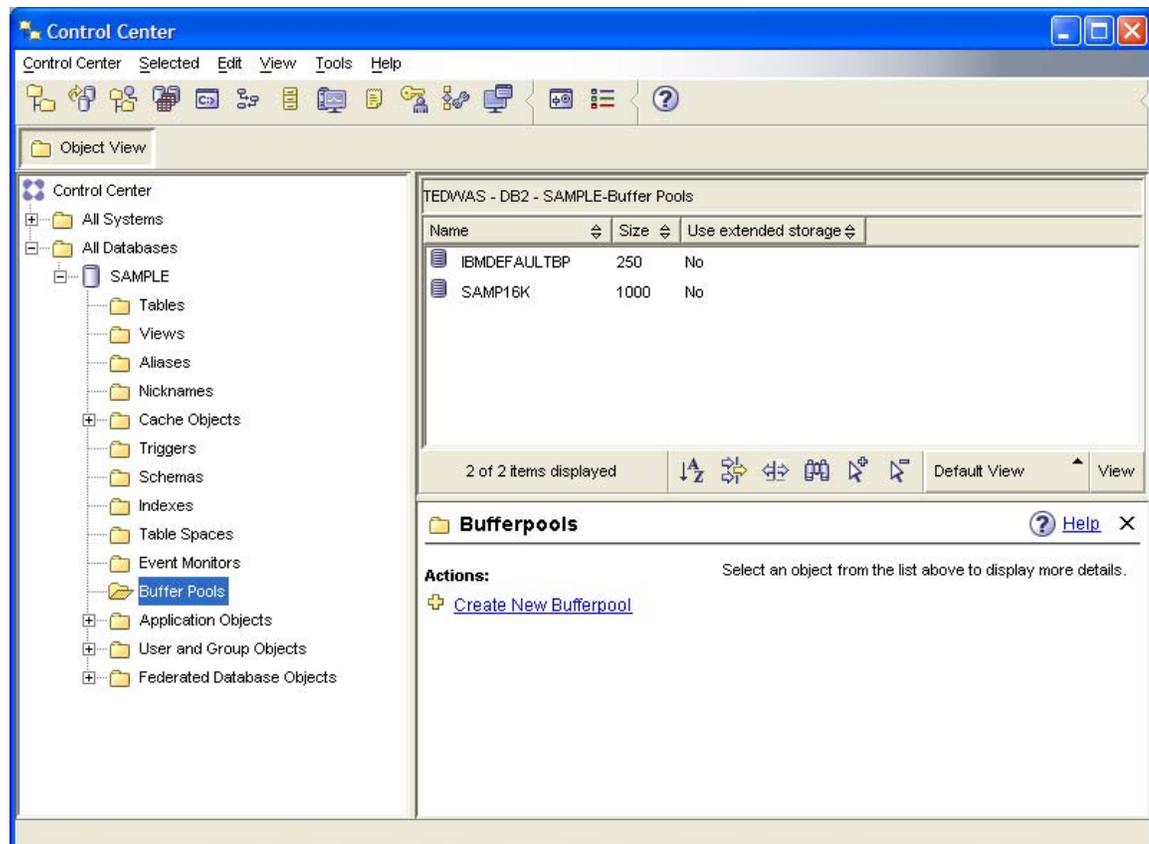


Figura 6.5 – El centro de control luego de haber creado el buffer pool "SAMP16K"

6.3.3 Espacios de Tabla (table spaces)

Los espacios de tabla son una interfaz lógica entre tablas lógicas y la memoria física (buffer pool) y los contenedores (disco) del sistema. Se usa la sentencia CREATE TABLESPACE para crearlos. Es posible especificar:

- El tamaño de página para el espacio de tabla (4K, 8K, 16K y 32K). El tamaño de página debe corresponder al de algún buffer pool.
- El nombre del buffer pool asociado a esta tabla.
- El tamaño de la extensión
- El tamaño de pre-recogida.

Tipos de Espacios de Tabla

Existen tres tipos de espacios de tabla:

- Regular
Usadas para tablas de usuario. Por ejemplo, el espacio de tabla USERSPACE1, creado por defecto, es regular.

- Grande
Usados para separar opcionalmente datos LOB en su propio espacio de tabla. También son usados para almacenar datos XML para bases de datos que los soportan (aquellas que fueron creadas como UNICODE y que usan el tipo de dato XML para las columnas). Los espacios de tabla grandes se escogen por defecto.
- Temporales
Se clasifican en dos tipos:
 - ▶ Del Sistema
Usados por DB2 para operaciones internas. Por ejemplo, el espacio de tabla TEMPSPACE1, creado siempre que se crea una base de datos, es un espacio de tabla temporal del sistema.
 - ▶ Del Usuario
Usados para crear tablas globales, temporales y definidas or el usuario (almacenadas en memoria). Se confunden fácilmente con las anteriores.

Gestión de Espacios de Tabla

Los espacios de tabla pueden ser clasificados de acuerdo a cómo se gestionan. El método de gestión puede ser especificado en la sentencia CREATE TABLESPACE:

Manejo por el Sistema

Este tipo de espacio de tabla se conoce como Almacenamiento Gestionado por el Sistema (SMS - System Managed Storage). Es el sistema operativo quien maneja este tipo de espacios de tabla. Son fáciles de gestionar, y los contenedores son directorios del sistema de archivos. El espacio no se reserva con anticipación, más bien, los archivos crecen dinámicamente. Una vez especificados los contenedores, la creación los fija y no es posible agregar otros luego, a menos que se utilice una restauración redirigida. Cuando se usen espacios de tabla SMS, los datos, datos LOB e índices de la tabla no pueden desperdigarse a través de distintos espacios de tabla.

Manejo por la Base de Datos

Este tipo de espacio de tabla se conoce como Almacenamiento Manejado por la Base de Datos (DMS - Database Managed Storage). Es DB2 quien maneja este tipo de espacio de tabla. La gestión del espacio requiere de más intervención manual por parte de un DBA. Pueden reservarse archivos o dispositivos crudos con anticipación para actuar como contenedores. Para dispositivos crudos, los datos se escriben directamente, sin caching de sistema operativo.

Los contenedores pueden ser borrados y redimensionados; nuevos pueden ser agregados. Espacios de tabla DMS presentan mejor desempeño, y datos, datos LOB e índices pueden repartirse en varios espacios de tabla, lo cual mejora el desempeño.

Manejo por Almacenamiento Automático

Este tipo de espacio de tabla es gestionado por almacenamiento automático, y se benefician del uso fácil similar al de espacios de tablas SMS, pero con el poder y flexibilidad de los DMS. Es por eso que, empezando con DB2 9, este es el espacio de tabla por defecto. Para estos, el usuario primero especifica un grupo lógico de dispositivos de almacenamiento. No se proveen definiciones explícitas de contenedores. Estos se

crean automáticamente a lo largo de las rutas de almacenamiento. El crecimiento de los contenedores existentes y la adición de nuevos son completamente manejados por DB2.

Para poder soportar el almacenamiento automático, es necesario primero crear una base de datos con dicha opción activada (este es el comportamiento por defecto) y asociar a ella una serie de rutas de almacenamiento. Si es necesario, es posible redefinir dichas rutas luego de la creación; esto se logra con una operación RESTORE. Luego, es posible crear espacios de tabla con almacenamiento automático (nuevamente, este es el comportamiento por defecto).

Ejemplo de Almacenamiento Automático

Primero se debe crear la base de datos con almacenamiento automático activado, como se muestra a continuación:

```
El almacenamiento automático está activado por defecto:  
CREATE DATABASE DB1
```

```
También puede ser especificado explícitamente:  
CREATE DATABASE DB1 AUTOMATIC STORAGE YES
```

```
No se especifica el almacenamiento automático, pero se dan las rutas:  
CREATE DATABASE DB1 ON /data/path1, /data/path2
```

```
El almacenamiento automático se deshabilita explícitamente:  
CREATE DATABASE DB1 AUTOMATIC STORAGE NO
```

Luego, se crea el espacio de tabla con almacenamiento automático, como se muestra en las siguientes sentencias:

```
El almacenamiento automático para espacios de tabla también está activado por defecto:  
CREATE TEMPORARY TABLESPACE TEMPTS
```

```
También puede ser especificado explícitamente:  
CREATE TABLESPACE TS2 MANAGED BY AUTOMATIC STORAGE
```

```
El almacenamiento automático se especifica explícitamente y se reservan el tamaño inicial, el incremento y el tamaño máximo:
```

```
CREATE TABLESPACE TS1  
    INITIALSIZE 500 K  
    INCREASESIZE 100 K  
    MAXSIZE 100 M
```

Cómo se Almacenan los Datos en los Espacios de Tabla

Por defecto, DB2 escribirá a la vez en varias extensiones de disco, repartidas a lo largo de distintos contenedores. Por ejemplo, para un espacio de tabla de 4K con un tamaño de extensión de 8 que usa 3 contenedores crudos en un espacio de tabla DMS, 32K de datos

(4K x 8 páginas por extensión = 32K) se escribirán en un disco antes de pasar al siguiente. La figura 6.6 ilustra este hecho. Notar que distintas tablas no comparten extensiones.

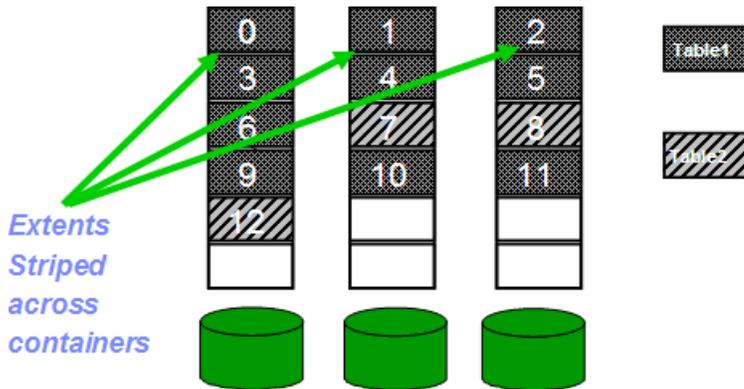


Figura 6.6 – Escribiendo Datos en Espacios de Tabla

Creando un Espacio de Tabla Usando el Centro de Control

Dar clic derecho en el folder "Table Spaces" dentro de una base de datos y seleccionar "Create", como muestra la figura 6.7. El ayudante para crear espacios de tabla aparecerá, como muestra la figura 6.8.

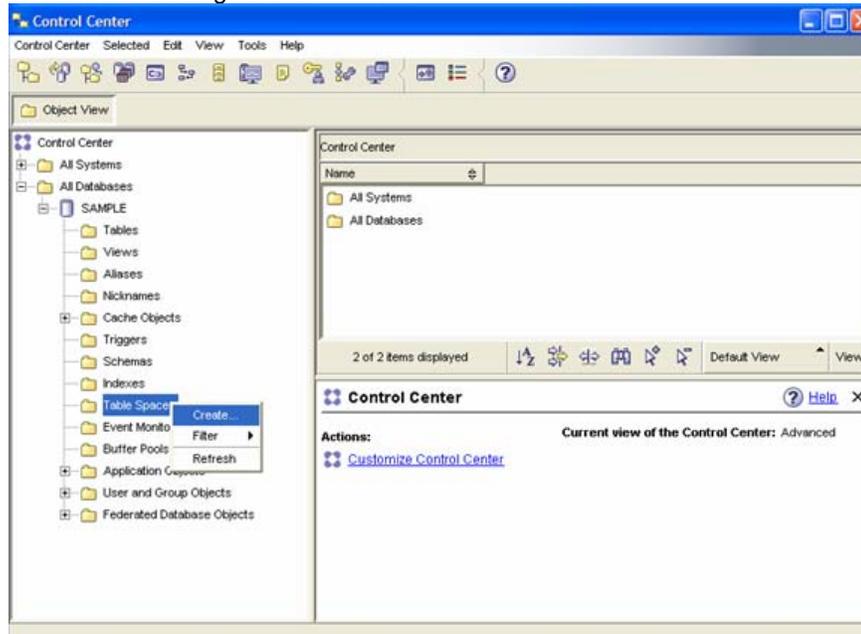


Figura 6.7 – Creando un Espacio de Tabla desde el Centro de Control

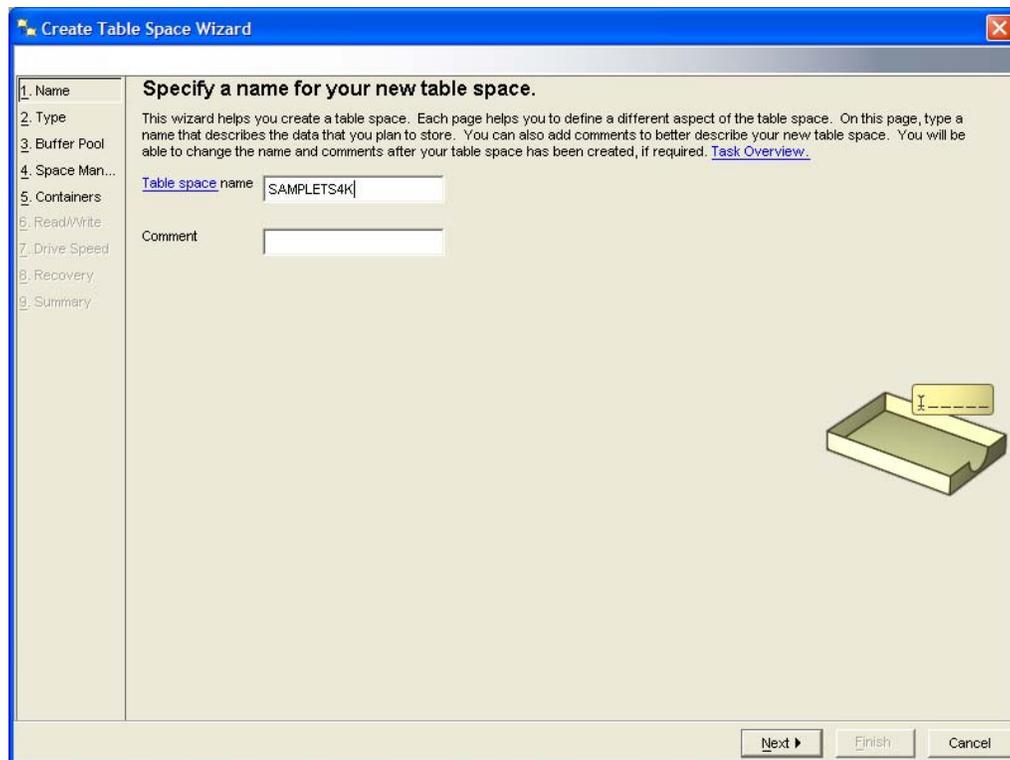


Figura 6.8 – El Ayudante de Creación de Espacios de Tabla

7

Capítulo 7 – Conectividad desde un cliente DB2

Este capítulo cubre la configuración requerida para conectar un cliente DB2 a un servidor de base de datos DB2 usando TCPIP. Note que un servidor DB2 viene con un componente cliente, así que un servidor DB2 puede comportarse como un cliente para conectarse a otro servidor DB2. Hay varios modos de configurar la conectividad de un cliente DB2; sin embargo, en este capítulo discutimos solamente el método más fácil que es usando el Asistente de Configuración.

Notar:

Para más información acerca de la conectividad desde un cliente DB2, observar este video: <http://www.channeldb2.com/video/video/show?id=807741:Video:4222>

7.1 Directorios de DB2

Los directorios de DB2 son archivos binarios que almacenan información acerca de cuales bases de datos se pueden conectar desde un cliente de DB2. Hay cuatro directorios:

1. Directorio Sistema de las Bases de Datos (System Database Directory)

El Directorio Sistema de las Bases de Datos es como la tabla de contenidos de un libro, muestra todas las bases de datos, ya sea que sean locales o remotas a las que se puede conectar. Para una base de datos local, tendrá un puntero al *Directorio de las Bases de Datos Locales*. Para una base de datos remota, tendrá un puntero a una entrada en el *Directorio de Nodo*. Para revisar los contenidos de este directorio suministrar el comando:

```
list db directory
```

2. Directorio de las Bases de Datos Locales (Local Database Directory)

Contiene información acerca de las bases de datos con las que se puede conectar y que residen en la máquina local. Para revisar los contenidos de este directorio se debe ejecutar este comando:

```
list db directory on <drive/path>
```

3. Directorio de Nodo

Este directorio incluye información sobre como conectarse a una base de datos dada. Por ejemplo, si se usa el protocolo TCP/IP, un nodo TCP/IP incluirá la dirección IP del servidor donde reside la base de datos DB2 al cual se está tratando de conectar y el puerto de la

instancia donde reside ésta base de datos. Para revisar este directorio ingrese el comando siguiente:

```
list node directory
```

4. Directorio DCS

Este directorio solamente aparecerá si está instalado el software DB2 Connect para conectarse a DB2 en z/OS (mainframe) o DB2 en i5/OS. Para revisar los contenidos de este directorio ingresar el comando:

```
list dcs directory
```

Uno puede revisar y actualizar el contenido de estos directorios con la herramienta Gráfica Asistente de configuración (Configuration Assistant).

7.2 El asistente de Configuración (Configuration Assistant)

Usando la herramienta gráfica Configuration Assistant, fácilmente se puede configurar la conectividad entre un cliente y un Servidor DB2.

Para iniciar el Asistente de configuración en Windows, se deberá elegir: *Inicio > Programas > IBM DB2 > DB2COPY1 > Set-up Tools > Configuration Assistant.*

Desde la línea de comando, se puede iniciar la herramienta con el comando `db2ca`. La figura 7.1 muestra el Asistente de Configuración.

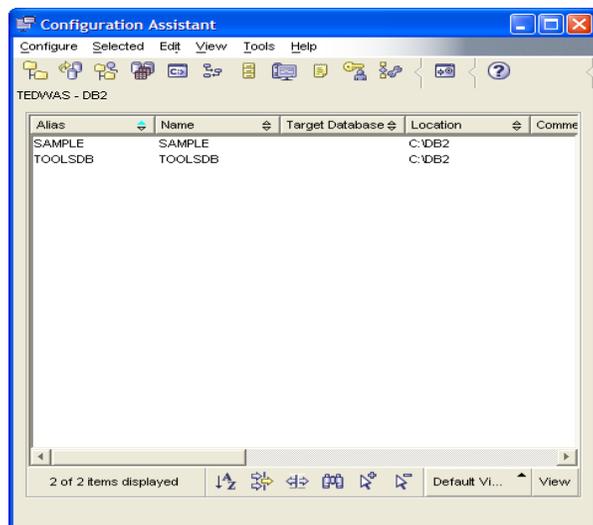


Figura 7.1 – El asistente de Configuración

7.2.1 La configuración requerida en el Servidor

Hay dos cosas que son necesarias de configurar en el servidor:

1) DB2COMM

Esta variable del registro determina cual Listeners de protocolo de comunicaciones deberían monitorear los requerimientos de los clientes. Típicamente TCPIP es el protocolo de comunicación mas usado. Cambiando este parámetro requiere un re inicio de la instancia. Para ver y cambiar el valor de DB2COMM desde el Asistente de Configuración elegir: *Configure -> DB2 Registry* según lo mostrado en la Figura 7.2 y en la Figura 7.3.

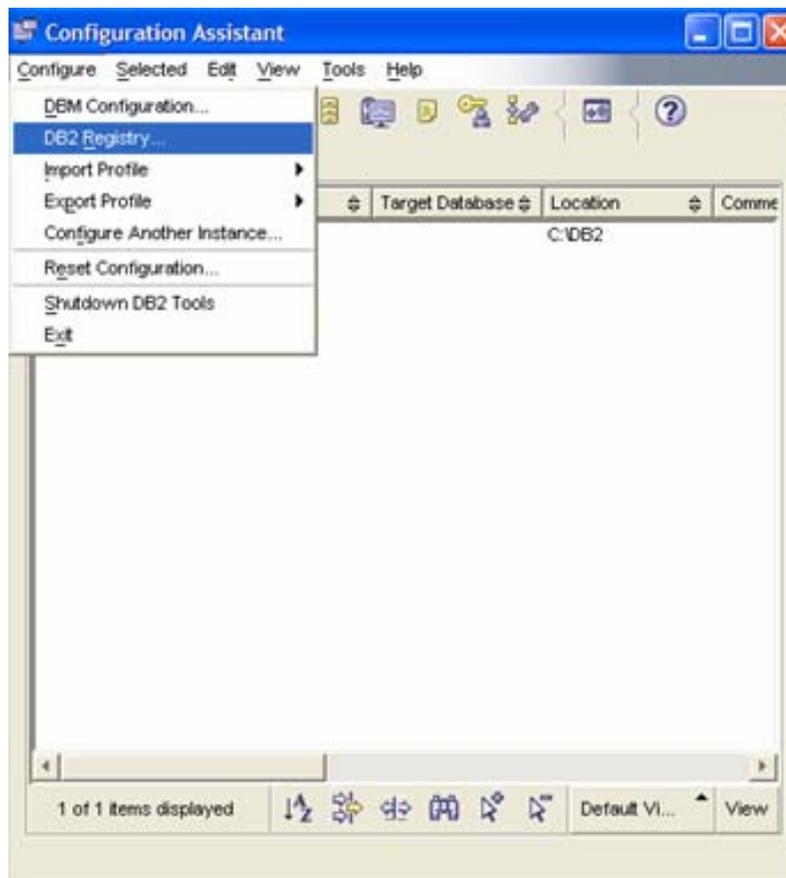


Figura 7.2 – Acceso al Registro de DB2

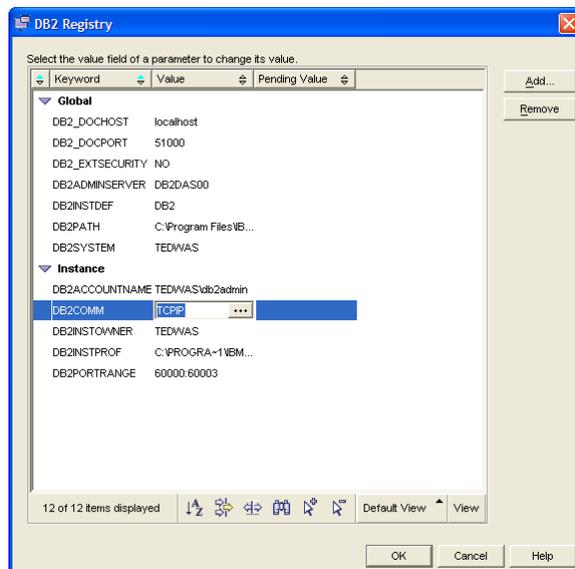


Figura 7.3 –Verificación del valor de las variables de registro DB2COMM DB2

2) SVCENAME

Este parámetro dbm cfg debe contener el nombre del servicio (según lo definido en el archivo de servicios TCPIP) o el número de puerto para usar cuando se quiere acceder a la base de datos de esta instancia. Desde el Asistente de Configuración, elegir *Configure > DBM configuration* según se ve en la Figura 7.4

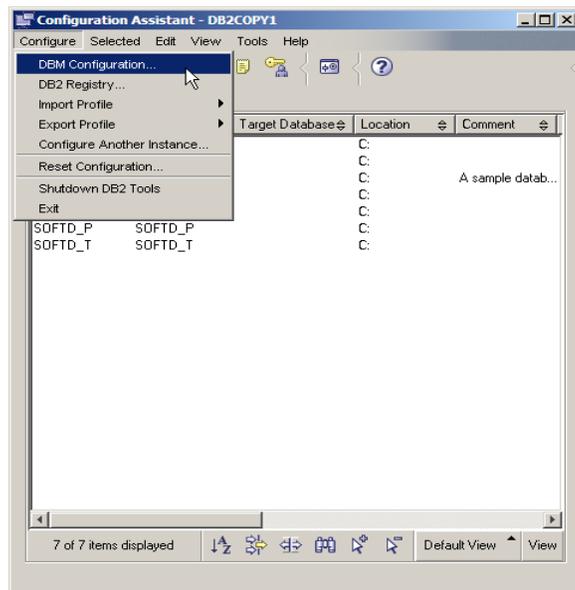


Figura 7.4 –Revisando el dbm cfg desde el Asistente de Configuración

Una vez abierta la ventana de configuración DBM, en la sección de Comunicaciones, buscar SVCENAME. Se puede cambiar el valor a un string o a un número de puerto si es lo requerido. Esto se muestra en la Figura 7.5.

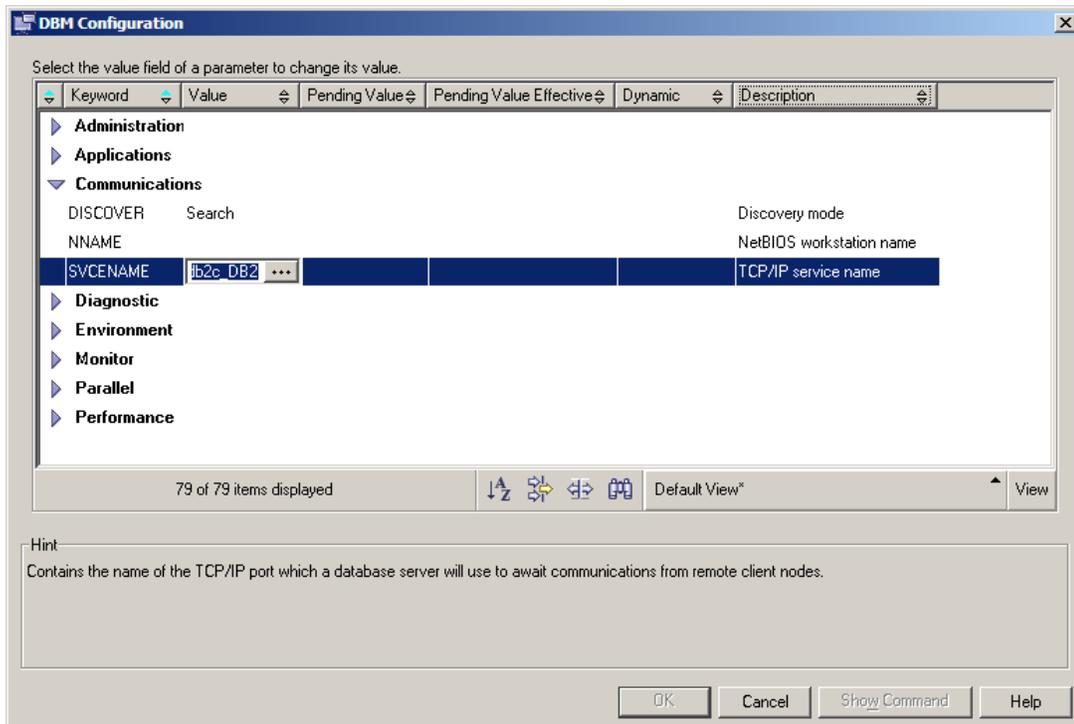


Figura 7.5 –Revisión del parametro dbm cfg SVCENAME

7.2.2 Configuración requerida en el cliente

En el cliente, se necesita saber esta información de antemano:

1. El nombre de la base de datos a la que se quiere conectar
2. El número de puerto de la instancia DB2 en el servidor donde la base de datos reside. Se puede usar también un nombre de servicio, mientras exista una entrada coincidente en los archivos de servicios de TCPIP
3. El ID del usuario de sistema operativo y la contraseña para conectarse a la base de datos. Este ID de usuario debe haber sido previamente definido en el servidor.

La información anterior será ingresada al cliente DB2 usando el Asistente de Configuración. Primero, lance el Wizard de Agregar Base de datos (Add Database Wizard) haciendo clic en *Selected -> Add Database Using Wizard*, como se muestra en la Figure 7.6

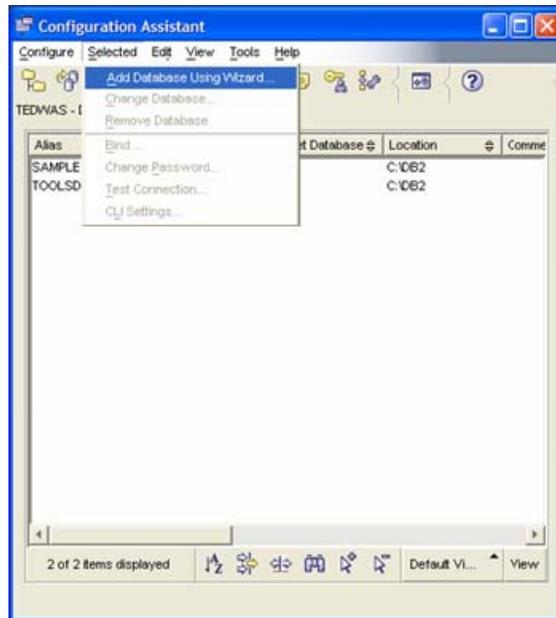


Figura 7.6 – Invocando al Asistente para agregar una Base de Datos (Add Database Wizard)

También se puede tomar este asistente haciendo clic en el espacio en blanco en el Asistente de Configuración y eligiendo la opción *Add Database Using Wizard*.

Figura 7.7 muestra el Asistente para agregar una Base de Datos.

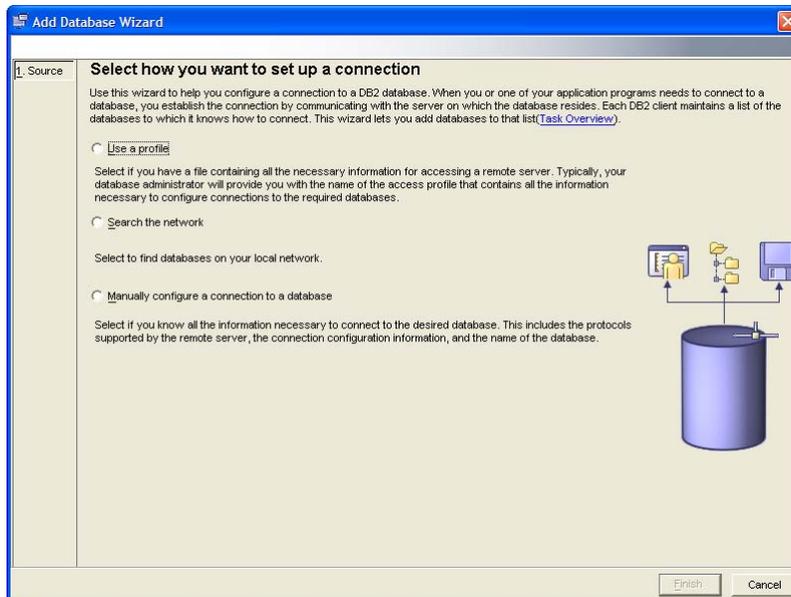


Figura 7.7 –Asistente de Agregar una Base de datos

En el Asistente de agregar una base de datos hay tres opciones:

Usar un Perfil (Use a profile)

Hay muchas situaciones cuando es necesario configurar muchos clientes para conectar al mismo servidor DB2. En estas situaciones, es conveniente realizar todas las configuraciones desde un cliente y almacenar la información en un archivo de Perfil (“profile” file). Con este archivo, se podrá cargar toda la información directamente a otros clientes. En la Figura 7.7, si es elegido “Use a Profile” se estaría cargando la información desde un archivo de perfil ya existente. Más detalles se proveerán mas tarde en este capítulo, describiendo como crear perfiles de clientes y servidores.

Buscar en la Red (Search the network)

Este método, también conocido como Descubrimiento o “Discovery”, le dice a DB2 que busque en la red un servidor determinado, una instancia y una base de datos. Para que este método funcione, el DAS debe estar ejecutándose en cada servidor DB2 donde estén las bases a ser descubiertas. Con este método, hay dos formas de realizar la búsqueda:

- **Buscar (Search):**
Busca en la red completa. Esto no es recomendado si la red es grande, con multiples centros (hubs) ya que tomaría mucho tiempo en obtener los datos de cada sistema en la red.
- **Conocer (Known):**
Buscar un servidor conocido en una dirección que se proveería.

Los dos métodos son ilustrados en la Figura 7.8

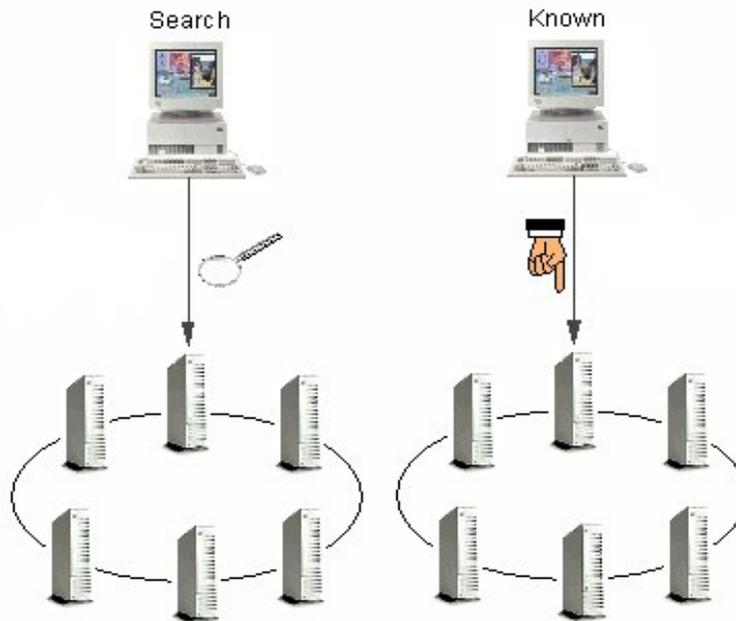


Figura 7.8 –Los métodos Buscar (Search) y Conocer (Known) para buscar o Descubrir

Pueden haber circunstancias en las que un administrador no desea que los clientes hagan búsquedas en la red de bases de datos con datos confidenciales. Esto puede ser prevenido a nivel de Base de Datos, Instancia o DAS. En la figura 7.9 se proveen detalles sobre este tema.

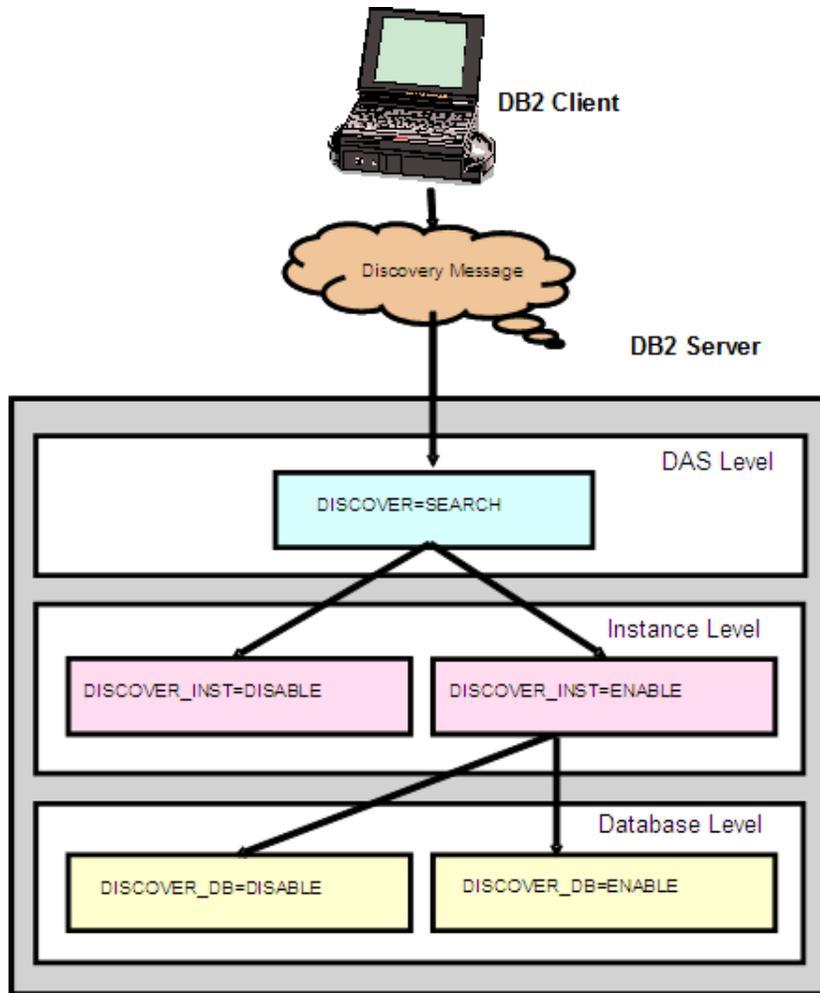


Figura 7.9 – Configuración de parámetros para permitir el descubrimiento

La figura 7.9 muestra los diferentes niveles donde se puede activar o desactivar la posibilidad de descubrir la base de datos. En el nivel de DAS, se puede darle al parámetro DISCOVER un valor de SEARCH o KNOWN. A nivel de instancia, el parámetro dbm cfg DISCOVER_INST puede fijarse en DISABLE o ENABLE. Finalmente, a nivel de Base de datos, el parámetro DISCOVER_DB también puede ser fijado como ENABLE o DISABLE. Fijando estos parámetros convenientemente se proveerá la suficiente granularidad para el descubrimiento de la base de datos.

Configurar manualmente una conexión a una base de datos (Manually configure...)

Usando este método, manualmente se puede agregar el nombre del host name, número de Puerto y la información de la base de datos al Asistente de Configuración, el cual generará el comando “catalog”, para ejecutar la configuración de conectividad. El Asistente de Configuración no revisará si la información es correcta. Se sabrá si es

incorrecta si no se puede conectar al servidor. También, asegurarse que el ID de usuario y contraseña que se provee para conectarse a la base de datos remota son correctos. Por defecto la autenticación se lleva a cabo en el Servidor DB2 en el que se desea conectar, por lo tanto se deberá proveer un ID de usuario y una contraseña definida en ese servidor.

7.2.3 Creación del perfil del Servidor y el Cliente

Si se desea configurar un gran número de servidores o clientes con la misma información, en vez de configurar cada uno individualmente, se puede configurar sólo uno y exportar un perfil o profile de éste. Luego se puede aplicar este perfil a los correspondientes clientes o servidores. Esto ahorra mucho tiempo en la configuración.

Para crear un perfil desde el Asistente de Configuración, haga clic en el Menu de configuración y luego seleccione Export Profile => Customize, como se muestra en la Figura 7.10

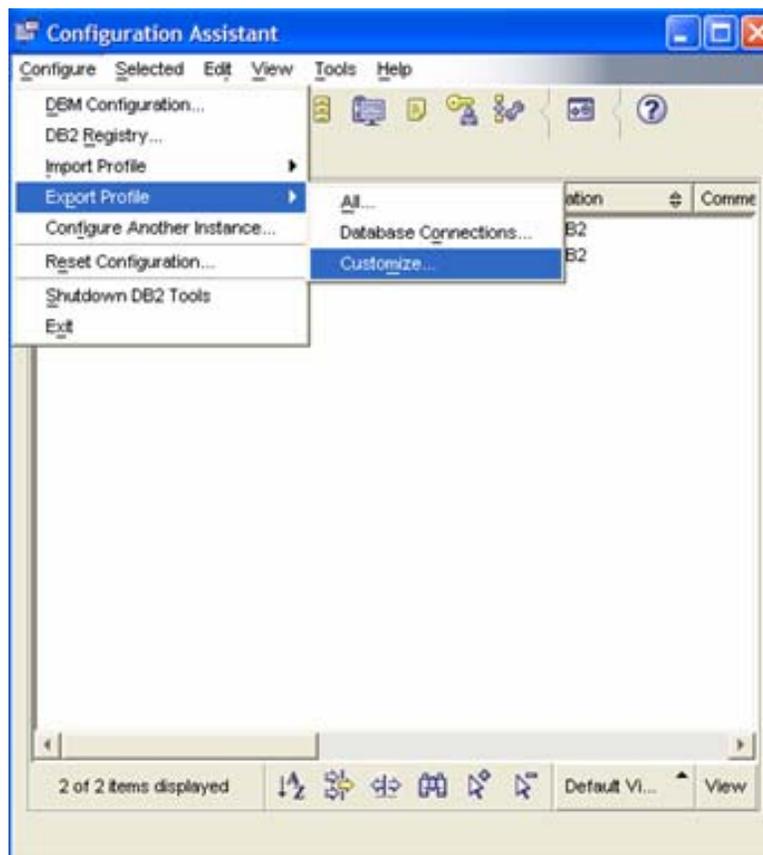


Figura 7.10 – Exportando un Profile

La Figura 7.11 muestra los campos que se necesitan completar para exportar un profile

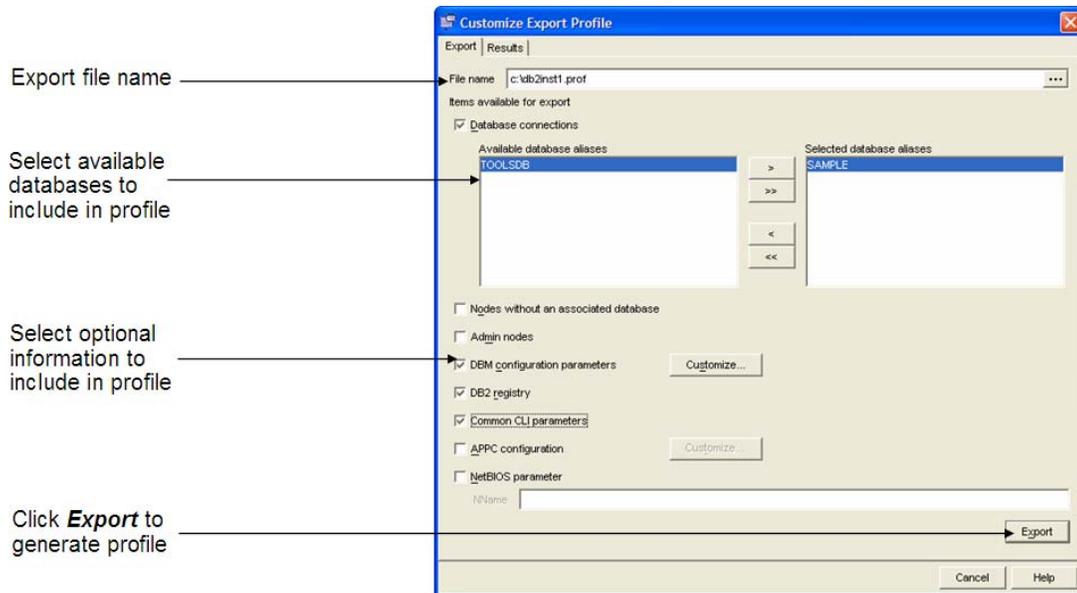


Figura 7.11 – Ventana de dialogo para exportar un Profile

Figura 7.12 muestra el resultado al hacer clic en “Export” en la ventana de dialogo para exportar un Profile.

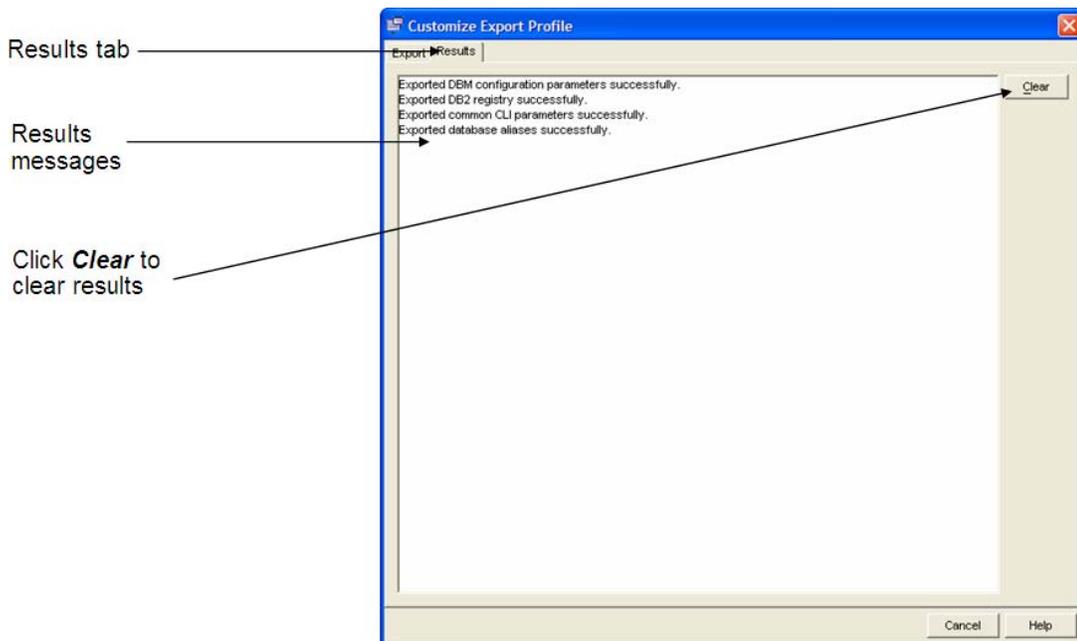


Figura 7.12 – Resultado de Exportar un Profile

Para importar un profile, desde el Asistente de Configuración haga clic en el menú de Configurar, luego seleccione Import Profile => Customize, como se vé en la Figura 7.13

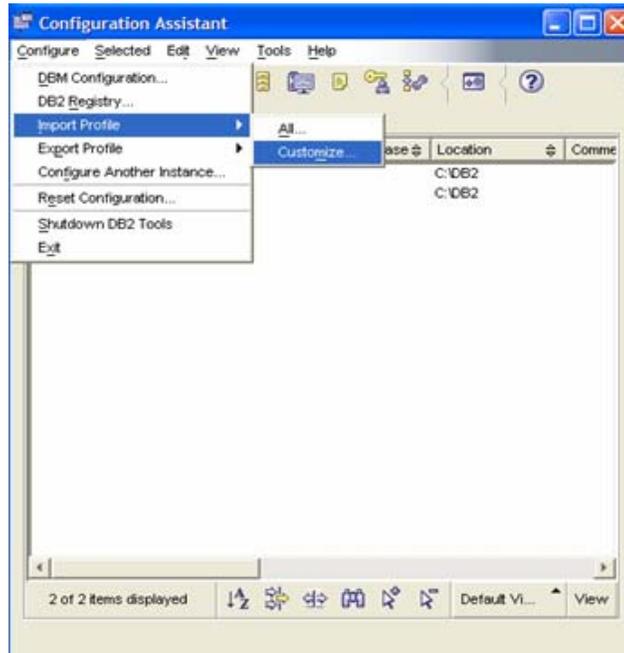


Figura 7.13 – Importación de un profile

La Figura 7.14 muestra los campos que necesitan ser completado para importar un profile

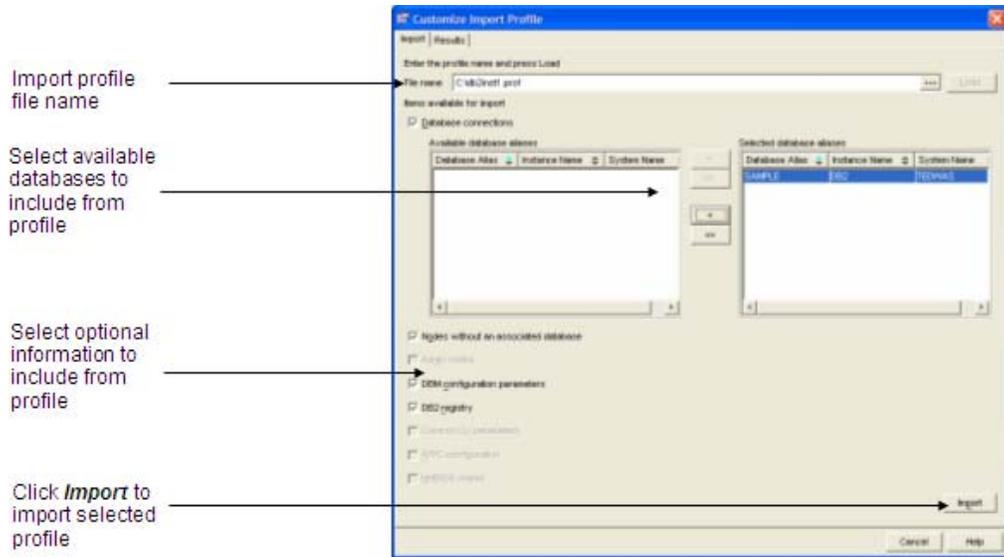


Figura 7.14 – Ajustar la importación de un Profile

6. En la página *TCP/IP* del asistente, ingresar el nombre completo del servidor o la dirección IP que fue escrita en el paso (1). Ingrese el número de puerto escrito en el paso (1). Normalmente por defecto, el puerto de la instancia de defecto es 50000. Hacer Clic en el botón *Next* para pasar a la siguiente página del asistente.

Nota: La opción de *Service Name* puede ser usada si tiene una entrada en el archivo *local Services* con un número de puerto definido, correspondiente al puerto en el que la instancia del servidor remoto esta atendiendo pedidos. Cuando se usa esta opción, DB2 buscará en los archivos de servicios en la máquina local y no en el servidor. Se debe agregar una entrada en este archivo si quiere usar esta opción.

7. En la página *Database* del asistente, ingresar el nombre de la base de datos definida en el servidor remoto que es el mismo que el definido en el paso (1) en el campo nombre de base de datos. Notar como el alias de Base de datos es completado automáticamente con el mismo valor. El alias de Base de datos es un nombre que las aplicaciones locales usarán para conectarse a esta base de datos. Ya que existe una base de datos llamada *SAMPLE*, DB2 no permitirá catalogar otra base de datos con el mismo nombre. Se debe, por lo tanto usar un alias diferente. Para este ejemplo, cambie el alias a *SAMPLE1*. Se puede ingresar un comentario adicional para esta base de datos si se desea. Hacer Clic en el botón *Next* para pasar a la siguiente página del asistente.
8. En la página *Data Source* del asistente, puede, opcionalmente registrar esta nueva base de datos como una fuente de datos ODBC. Esto registrará automáticamente en el administrador de ODBC de Windows. Para este ejemplo, de-seleccionar la opción *Register this database for ODBC* ya que no se usará ODBC. Hacer Clic en el botón *Next* para pasar a la siguiente página del asistente.
9. En la página *Node Options* del asistente, especificar el Sistema Operativo del servidor donde la base de datos remota está localizada. En el campo Instance name fijar el valor *DB2*, asumiendo que ese es el nombre de la instancia que contiene a la base de datos a la que se quiere conectar. Hacer Clic en el botón *Next* para pasar a la siguiente página del asistente.
10. Esta página de *System Options* ofrece la oportunidad de asegurar que el sistema y el nombre del host son correctos, y para verificar la configuración del Sistema operativo. Hacer Clic en el botón *Next* para pasar a la siguiente página del asistente.
11. La página *Security Options* del wizard permite especificar donde se desea que la autenticación sea llevada a cabo y qué método quiere usarse.

Seleccionar el valor *“Use authentication value”* en la configuración DBM de Servers. Esto hará que se use el método especificado por el parámetro AUTHENTICATION en el archivo de configuración remoto. Hacer Clic en el botón *Finish* para catalogar la base de datos remota y cerrar el Asistente. Una ventana de confirmación aparecerá y se deberá hacer clic en el botón *Test Connection* para asegurarse que se puede conectar exitosamente con la Base de datos. También, asegúrese que el nombre de usuario y contraseña provista es la valida en el servidor remoto (ya que por defecto, el parámetro AUTHENTICATION en el servidor es SERVER). Si el test de conexión es exitoso, tenemos correctamente catalogada la base de datos remota. Si esto no es exitoso, se deberá retornar al Asistente y asegurarse que todos los valores correctos están especificados. (Haga clic en el botón de *Change* para volver a través de las configuraciones del Asistente).

12. Abrir el Centro de Control y tratar de ver las diferentes tablas en la recientemente catalogada Base de Datos remota.
13. Si está conectado en una red, vuelva al Asistente de Configuración y pruebe de catalogar una base de datos diferente, esta vez usando la opción *Search the Network*. Avanzar a través del asistente de la misma forma que se hizo manualmente configurando la conexión. Note que en redes grandes el descubrimiento podría llevar mucho tiempo en devolver resultados.

8

Capítulo 8 – Trabajando con Objetos de la Base de Datos

Este capítulo trata sobre objetos de la Base de datos como esquemas, tablas, vistas, indexes, secuencias y otros. Algunos objetos de aplicaciones de Bases de Datos avanzadas como los triggers, las Funciones definidas por el usuario (sus siglas en inglés son UDF) y procedimientos almacenados son tratados en el capítulo 14, SQL Procedimientos Almacenados PL y en el capítulo 15, Inline SQL PL, UDFs y Triggers.

Nota:

Para mayor información sobre el trabajo con objetos de la Base de Datos observar el videos en: <http://www.channeldb2.com/video/video/show?id=807741:Video:4242>

8.1 Esquema

Los esquemas son espacios de nombres para una colección de objetos de Bases de Datos. Son usados primariamente para:

- Proveer una indicación de pertenencia del objeto o la relación a una aplicación
- Agrupar lógicamente a objetos relacionados

Todos los objetos de bases de datos en DB2 tienen un nombre con dos partes para ser completamente calificado. El esquema es la primera parte del nombre:

<nombre_del_esquema>.<nombre_del_objeto>

El nombre completamente calificado de un objeto debe ser único. Cuando se conecta a una base de datos y se crea o referencia un objeto sin especificar el esquema, DB2 usa el ID de usuario con el que se ha conectado como el nombre del esquema. Por ejemplo, si se ha conectado a la base de datos SAMPLE con el usuario "arfchong", y se crea una tabla usando la sentencia CREATE TABLE:

```
CREATE TABLE artists ...
```

El nombre completamente calificado de la tabla será arfchong.artists.

8.2 Tablas

Una tabla es una colección de datos relacionados y lógicamente arreglados en columnas y filas. La sentencia de abajo provee un ejemplo de cómo se crea una tabla con la sentencia CREATE TABLE.

```
CREATE TABLE artists
(artno          SMALLINT    not null,
 name          VARCHAR(50) with default 'abc',
 classification CHAR(1)     not null,
 bio           CLOB(100K)   logged,
 picture       BLOB(2M)     not logged compact
)
IN mytbls1
```

En las siguientes secciones, describiremos las principales partes de esta sentencia CREATE TABLE

8.2.1 Tipos de datos

La Figura 8.1 lista los tipos de datos soportados en DB2

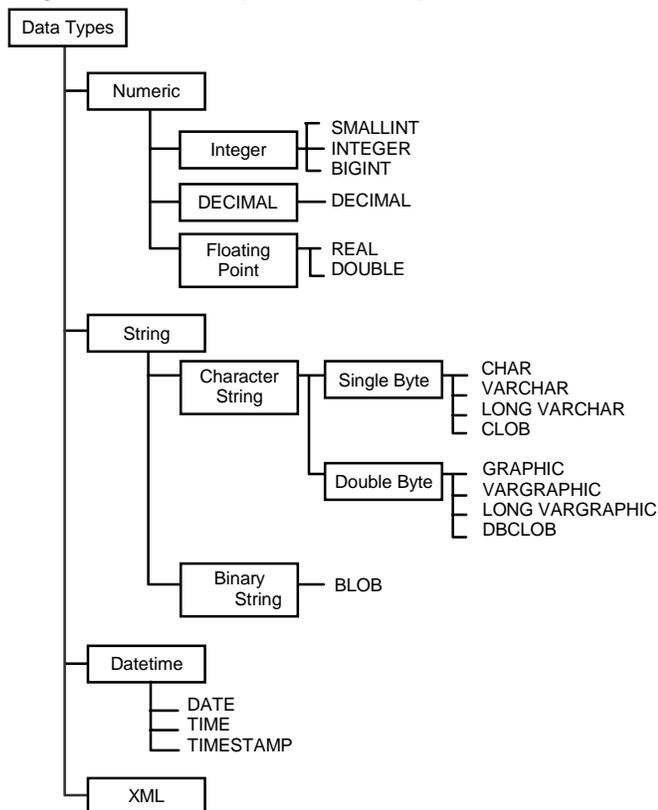


Figura 8.1 – Tipos de datos incorporados de DB2

Tipos de datos de Objetos Grandes(LOB)

Los tipos de datos de objeto grande son usados y almacenados para almacenar cadenas de caracteres o de binarios de gran tamaño y también archivos como se vé en la figura 8.2

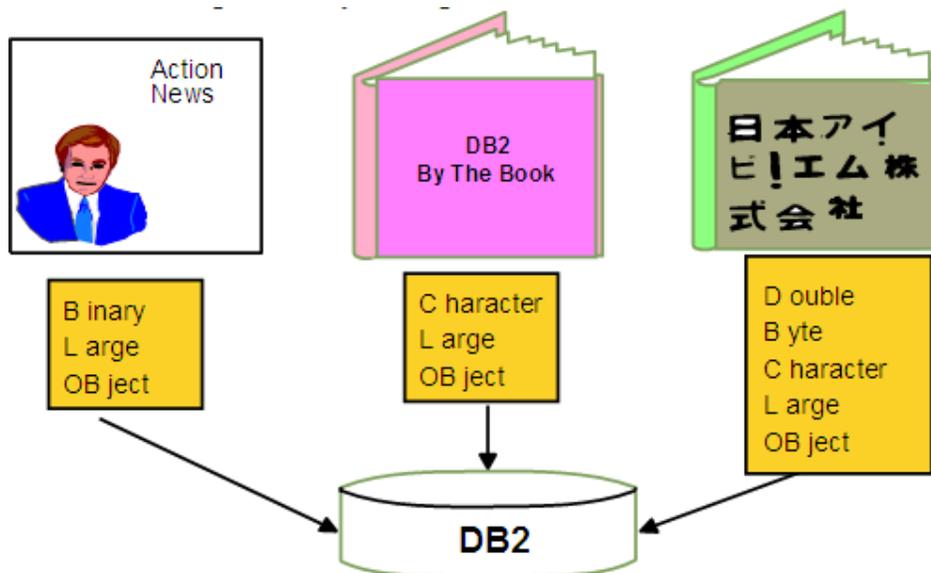


Figura 8.2 – Tipos de datos LOB

Estos objetos binarios grandes reciben como nombre las siglas LOB para claridad, se los distingue como BLOB para los objetos binarios grandes, para los objetos de caracteres se usa CLOB y un objeto grande de doble byte es conocido también como DBCLOB.

Tipos definidos por el usuario (User-defined types)

DB2 permite al desarrollador definir sus propios tipos de datos, basados en los tipos de datos incorporados. Esto es conocido como Tipos definidos por el usuario (ó UDT). Estos UDTs son útiles cuando:

- Hay una necesidad para establecer el contexto para los valores
- Hay una necesidad para que DB2 fuerce el tipo de datos

Las siguientes sentencias ilustran un ejemplo de como y cuando usar UDTs:

```
CREATE DISTINCT TYPE POUND AS INTEGER WITH COMPARISONS
```

```
CREATE DISTINCT TYPE KILOGRAM AS INTEGER WITH COMPARISONS
```

```
CREATE TABLE person
  (f_name  VARCHAR(30),
   weight_p POUND NOT NULL,
   weight_k KILOGRAM NOT NULL )
```

En este ejemplo se han creado dos UDTs: POUND (Libra) y KILOGRAM (Kilogramo). Ambas son construidas basadas en el tipo de dato INTEGER. La cláusula WITH COMPARISONS definida como parte de la sintaxis causará que funciones de conversión o casting con el mismo nombre que los tipos de datos, también sean creadas.

La tabla `person` usa los dos nuevos UDTs en las columnas `weight_p` y `weight_k`, respectivamente. Si ahora ejecutamos las siguientes sentencias:

```
SELECT F_NAME FROM PERSON
WHERE weight_p > weight_k
```

Se recibirá un error debido a que las dos columnas tienen diferentes tipos de datos y no pueden ser comparadas. Aun cuando `weight_p` y `weight_k` usen respectivamente los tipos de datos POUND y KILOGRAM, y que ambos fueron creados en base al tipo de dato INTEGER, la comparación es imposible. Esto es exactamente lo buscado por que estas dos columnas no son comparables por que no se debe poder comparar libras y kilogramos.

En el próximo ejemplo, se quiere comparar la columna `weight_p` con un entero; sin embargo, estos dos tipos son diferentes, y por lo tanto se recibirá un error a menos que se use una función de traducción o cast.

Así como se ve en la sentencia de abajo, se puede usar la función de traducción POUND() así se puede realizar la comparación. Como se indicó antes la función de cast, POUND() fue creada con UDT cuando es invocado con la cláusula WITH COMPARISONS de la sentencia CREATE DISTINCT TYPE.

```
SELECT F_NAME FROM PERSON
WHERE weight_p > POUND(30)
```

Valores Null

Un valor null representa un estado desconocido. Sin embargo la sentencia CREATE TABLE puede definir una columna usando la cláusula NOT NULL. Esto asegura que la columna contiene un valor de datos conocido. Puede especificarse un valor default si una columna es definida como NOT NULL. La siguiente sentencia provee ejemplo de este comportamiento:

```
CREATE TABLE Staff (
  ID          SMALLINT NOT NULL,
  NAME       VARCHAR(9),
  DEPT       SMALLINT NOT NULL with default 10,
  JOB        CHAR(5),
  YEARS      SMALLINT,
  SALARY     DECIMAL(7,2),
  COMM       DECIMAL(7,2) with default 15
)
```

8.2.2 Columnas de Identidad

Una columna de identidad es una columna numérica que automáticamente genera un valor numérico para cada fila insertada. Puede haber sólo una columna identidad por tabla.

Hay dos modos de generar valores para una columna de identidad, dependiendo de como fue definida:

- **Generated always:** los valores son siempre generados por DB2. Las aplicaciones no tienen permiso para proveer un valor explícito.
- **Generated by default:** los valores pueden ser explícitamente provistos por una aplicación o, si no se les da un valor, entonces DB2 generará uno. DB2 no puede garantizar unicidad. Esta opción está orientada para la propagación de datos y para la carga y recarga de una tabla.

Examinemos el siguiente ejemplo:

```
CREATE TABLE subscriber(subscriberID INTEGER GENERATED ALWAYS AS
                        IDENTITY (START WITH 100
                        INCREMENT BY 100),
                        firstname VARCHAR(50),
                        lastname  VARCHAR(50) )
```

En el ejemplo, la columna subscriberID es un INTEGER definido como columnas de identidad que son generadas siempre. El valor generado empezará en 100, y serán incrementados de a 100.

8.2.3 Objetos Secuencia (SEQUENCE)

Los objetos secuencia generan un número único a través de la base de datos. A diferencia de las columnas identidad, las secuencias son independientes de las tablas. Los siguientes enunciados proveen un ejemplo:

```
CREATE TABLE t1 (salary int)

CREATE SEQUENCE myseq
  START WITH 10
  INCREMENT BY 1
  NO CYCLE

INSERT INTO t1 VALUES (nextval for myseq)
INSERT INTO t1 VALUES (nextval for myseq)
INSERT INTO t1 VALUES (nextval for myseq)

SELECT * FROM t1

SALARY
-----
      10
```

```

        11
        12
3 record(s) selected.

SELECT prevval for myseq FROM sysibm.sysdummy1

1
-----
        12
1 record(s) selected

```

PREVVAL provee el valor actual de la secuencia, mientras que NEXTVAL provee el siguiente valor.

El ejemplo anterior utiliza la tabla SYSIBM.SYSDUMMY1. Esta tabla del catálogo del sistema contiene una fila y una columna y fue creada para utilizarla en situaciones como la del ejemplo, donde una consulta requiere una salida basada en solamente un valor. Las tablas del catálogo del sistema son descritas en la siguiente sección.

8.2.4 Tablas del catálogo del sistema

Cada base de datos tiene sus propias tablas y vistas del catálogo del sistema. Estas almacenan metadatos acerca de los objetos de la base de datos. Se puede consultar estas tablas como cualquier otra tabla normal de una base de datos. Tres esquemas son usados para identificar las tablas del catálogo del sistema:

- SYSIBM: tablas base, optimizadas para el uso de DB2.
- SYSCAT: vistas basadas en las tablas de SYSIBM, optimizadas para un fácil uso.
- SYSSTAT: estadísticas de la base de datos.

Las siguientes son algunos ejemplos de vistas del catálogo:

- SYSCAT.TABLES
- SYSCAT.INDEXES
- SYSCAT.COLUMNS
- SYSCAT.FUNCTIONS
- SYSCAT.PROCEDURES

8.2.5 Tablas temporales declaradas

Las tablas temporales declaradas son tablas creadas en memoria y son usadas por una aplicación y borradas automáticamente cuando la aplicación termina. Estas tablas solamente pueden ser accedidas por la aplicación que las crea. El acceso a este tipo de tablas provee un rendimiento muy eficiente por que no existen entradas en ninguna tabla del catálogo de DB2, no hay bloqueo de filas, no hay generación de bitácoras por defecto (que son opcionales), y no hay verificación de autoridad. Es eficiente también por que estas tablas temporales soportan el uso de índices. Se puede también ejecutar RUNSTATS en estas tablas.

Las tablas temporales declaradas residen dentro de un espacio de tablas temporal, el cual debe ser definido previamente a crear cualquier tabla temporal declarada. El siguiente enunciado muestra un ejemplo de como crear tres tablas temporales declaradas.

```
CREATE USER TEMPORARY TABLESPACE apptemps
    MANAGED BY SYSTEM USING ('apptemps');

DECLARE GLOBAL TEMPORARY TABLE tempemployees
    LIKE employee NOT LOGGED;

DECLARE GLOBAL TEMPORARY TABLE tempdept
    (deptid CHAR(6), deptname CHAR(20))
    ON COMMIT DELETE ROWS NOT LOGGED;

DECLARE GLOBAL TEMPORARY TABLE tempprojects
    AS ( fullselect ) DEFINITION ONLY
    ON COMMIT PRESERVE ROWS NOT LOGGED
    WITH REPLACE IN TABLESPACE apptemps;
```

Cuando una tabla temporal declarada es creada, su esquema es SESSION, y debe ser especificado. El ID de usuario empleado para crear una tabla temporal declarada tendrá todos los privilegios en la tabla. Cada aplicación que crea una tabla temporal tendrá su copia independiente como es mostrado en la figura 8.5.



Figura 8.5 – Alcance de las tablas temporales declaradas.

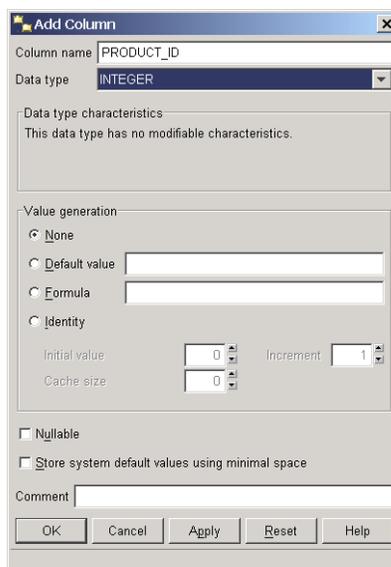
QuickLab #7: Creando una nueva tabla

Objetivo

Hasta ahora, se han utilizado las tablas existentes en la base de datos SAMPLE para ilustrar los conceptos. Eventualmente, se necesitará crear tablas propias en la base de datos. En este laboratorio, se utilizará el asistente para crear dos nuevas tablas en la base de datos SAMPLE.

Procedimiento

- Lanzar el asistente de creación de tablas como se ha mostrado anteriormente en este libro. (Control Center -> All Databases -> SAMPLE -> (clic derecho) Tables object -> Create).
- Definir el nombre de la tabla, definiciones de columnas y algunas restricciones. La tabla que será utilizada para almacenar información acerca de artículos de oficina utilizados por un proyecto utilizado en la base de datos sample. Cada vez que un artículo es comprado, una fila será agregada a esta tabla. La tabla tendrá seis columnas:
 - ▶ product_id: identificador único del artículo que está siendo comprado.
 - ▶ description: descripción del artículo.
 - ▶ quantity: la cantidad comprada.
 - ▶ cost: el costo del artículo.
 - ▶ image: una imagen del artículo (si está disponible)
 - ▶ project_num: el proyecto de este artículo ha sido comprado por.
- En la primera página del asistente, para el nombre del esquema, indique el ID de usuario con el que actualmente entró al sistema, y use el siguiente nombre de tabla: SUPPLIES. Opcionalmente puede indicar un comentario. Continúe con la siguiente página del asistente.
- Desde esta página, se pueden agregar columnas a la tabla. Clic en el botón ADD para agregar columnas.



Ingrese el nombre de la columna “product_id” y seleccione el tipo de dato: INTEGER. Deseleccione Nullable y de clic en el botón Apply para definir la columna.

Repita este paso para las columnas restantes de la tabla utilizando las opciones mostradas en la tabla siguiente. Una vez que todas las columnas han sido agregadas, hacer clic en el botón OK y la lista de las columnas creadas deberá ser resumida. Continuar con la siguiente página del asistente.

Column Name	Attributes
product_id (completed)	INTEGER, NOT NULL
Description	VARCHAR, length 40, NOT NULL
Quantity	INTEGER, NOT NULL
Cost	DECIMAL, Precision 7, Scale 2, NOT NULL
Image	BLOB, 1MB, NULLABLE, NOT LOGGED
Project_num	CHAR, length 6, NOT NULL

Nota: La opción NOT LOGGED puede ser especificada cuando se declaran columnas LOB. Esto es obligatorio para columnas mayores a 1GB en tamaño. Esto es también generalmente recomendado para LOBs mayores a 10MB como cambios de tamaño de columnas que pueden rápidamente llenar el archivo de bitácora (log). Aun si NOT LOGGED es utilizado, cambios en los archivos LOB durante una transacción pueden ser deshechos (rolled back). También nótese que la columna imagen (image) es la única definida como “NULLABLE”. Porqué crees que este es el caso?

- Ahora se tiene toda la información obligatoria para la creación de una tabla. En las siguientes páginas, seleccione los valores por defecto. Se puede siempre agregar

llaves y restricciones después de que una tabla ha sido creada.

- Agregar una restricción a la tabla para restringir valores en la cantidad. En la página Constraint del asistente, de clic en el botón ADD. En el campo Check Name, introduzca: valid_quantities. En el campo Check Condition, introduzca: quantity > 0.

Dar clic en el botón OK. Deberá mostrarse un resumen de las constantes que han sido agregadas en la página Constraints del asistente. Continúe con la siguiente página del asistente.

- Se puede continuar a través del asistente, cambiando los otros parámetros de la tabla. Alternativamente, se puede saltar a la página Summary, o simplemente dar clic en el botón Finish para crear la tabla.
- Desde el centro de control, de clic en el directorio Tables bajo la base de datos SAMPLE en el panel de árbol de objetos. La tabla creada debe ahora aparecer en esta lista. Podría ser necesario refrescar la vista del centro de control para observar los cambios.

8.3 Vistas

Una vista es una representación de los datos en las tablas. Los datos para las vistas no son almacenados separadamente, porque son obtenidos cuando la vista es invocada. Las vistas anidadas, que son vistas basadas en otra vistas, son soportadas. Toda la información acerca de las vista es mantenida en las siguientes vistas del catálogo de DB2: SYSCAT.VIEWS, SYSCAT.VIEWDEP, y SYSCAT.TABLES. Este es un ejemplo de como crear y utilizar una vista.

```
CONNECT TO MYDB1;

CREATE VIEW MYVIEW1
  AS SELECT ARTNO, NAME, CLASSIFICATION
  FROM ARTISTS;
```

```
SELECT * FROM MYVIEW1;
```

Resultado:

ARTNO	NAME	CLASSIFICATION
-----	-----	-----
10	HUMAN	A
20	MY PLANT	C
30	THE STORE	E
...		

8.4 Índices

Un índice es un conjunto ordenado de claves cada una de las cuales apunta a una fila en la tabla. Un índice permite unicidad, y también mejora el rendimiento. Algunas de las características que se pueden definir en los índices son:

- El orden del índice que puede ser ascendente o descendente.
- Las claves del índice pueden ser únicas o no únicas.
- Diversas columnas pueden ser utilizadas para el índice (esto es llamado un índice compuesto)
- Si el índice y los datos físicos son agrupados en una secuencia de índice similar, ellos son un índice agrupado.

Por ejemplo:

```
CREATE UNIQUE INDEX artno_ix ON artists (artno)
```

8.4.1 Consejero de diseño

El consejero de diseño (design advisor) es una excelente herramienta para el óptimo diseño de la base de datos para la carga de trabajo dada. El consejero de diseño puede ayudar con el diseño de los índices, Tablas Materializadas (MQTs). Agrupación multidimensional (MDC), y la característica de particionamiento de bases de datos. El

consejero de diseño es invocado desde el centro de control dando clic derecho en una base de datos y seleccionando “Design Advisor” como se muestra en la figura 8.6.

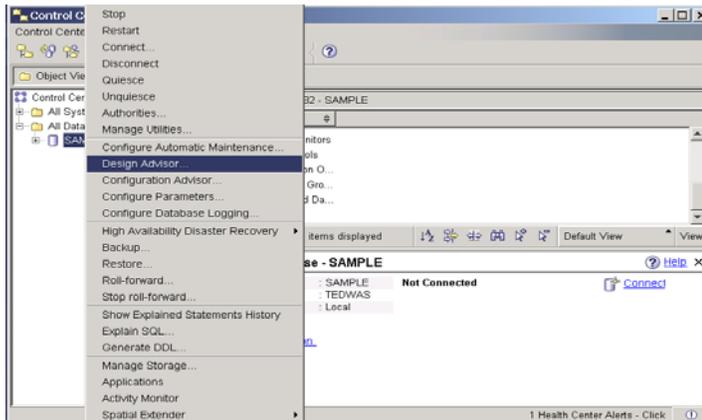


Figura 8.6 – Invocando al consejero de diseño desde el centro de control.

La figura 8.7 muestra al consejero de diseño. Siga los pasos del asistente para obtener las recomendaciones de diseño desde DB2.

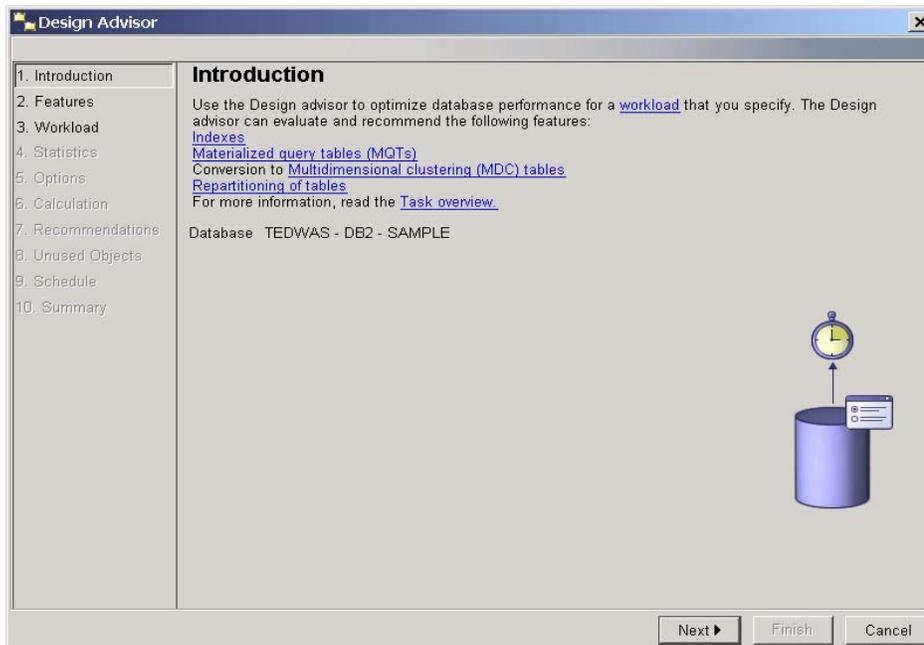


Figura 8.7 – El consejero de diseño (Design advisor).

8.5 Integridad referencial

La integridad permite a la base de datos manejar las relaciones entre tablas. Es posible establecer el tipo de relaciones padre-hijo de las relaciones entre las tablas como se muestra en la figura 8.8. En la figura, hay dos tablas, DEPARTMENT y EMPLOYEE, relacionadas por el nombre de departamento. La columna WORKDEPT en la tabla EMPLOYEE puede solamente contener un número de departamento que ya exista en la tabla DEPARTMENT. Esto es porque en este ejemplo, la tabla DEPARTMENT es la tabla padre, y la tabla EMPLOYEE es el hijo, o tabla dependiente. La figura también muestra el enunciado CREATE TABLE necesario que la tabla EMPLOYEE necesitaba para establecer la relación.

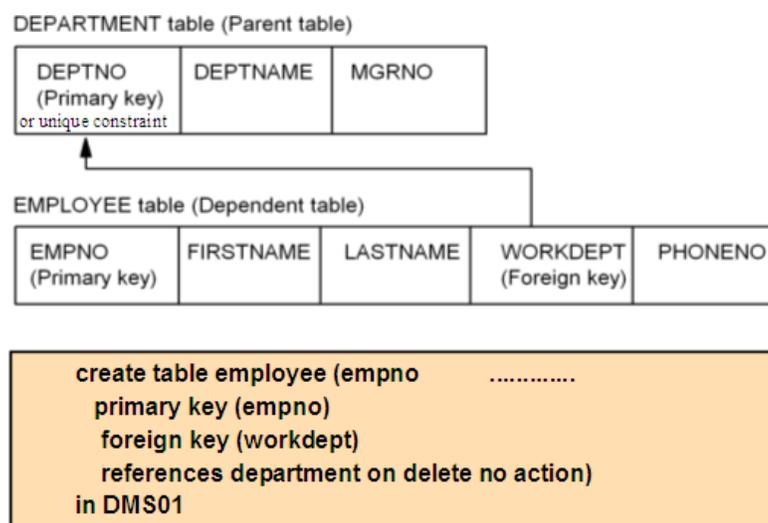


Figura 8.8 – Un ejemplo de la integridad referencial entre tablas.

En integridad referencial, se utilizan frecuentemente los siguientes conceptos.

Concepto	Descripción
Tabla padre (Parent table)	Una tabla de control en la cual las llaves padre existen
Tabla dependiente (Dependent table)	Una tabla dependiente de los datos de la tabla padre. También puede contener una clave foránea. Para una fila que existe en la tabla dependiente, una fila coincidente debe existir previamente dentro de la tabla padre.
Clave primaria (Primary Key)	Define la llave padre en la tabla padre. No puede contener valores nulos y los valores deben ser únicos. Una clave primaria consiste en una o más columnas dentro de la tabla.
Clave foránea (Foreign Key)	Referencia la clave foránea a la tabla padre

Los datos en las tablas pueden ser relacionados a datos de una o mas tablas con integridad referencial. Las restricciones también son impuestas en valores de datos para formar una determinada propiedad o regla de negocio. Por ejemplo, si una tabla almacena

el sexo de una persona, la restricción puede forzar a que solo los valores permitidos sean M para masculino y F para femenino.

9

Capítulo 9 – Utilidades para el movimiento de datos

Las herramientas y comandos descritos en esta sección son utilizados para mover datos dentro de la misma base de datos u otras bases de datos en la misma o en diferentes plataformas. La figura 9.1 provee una descripción de las utilidades de movimiento de datos.

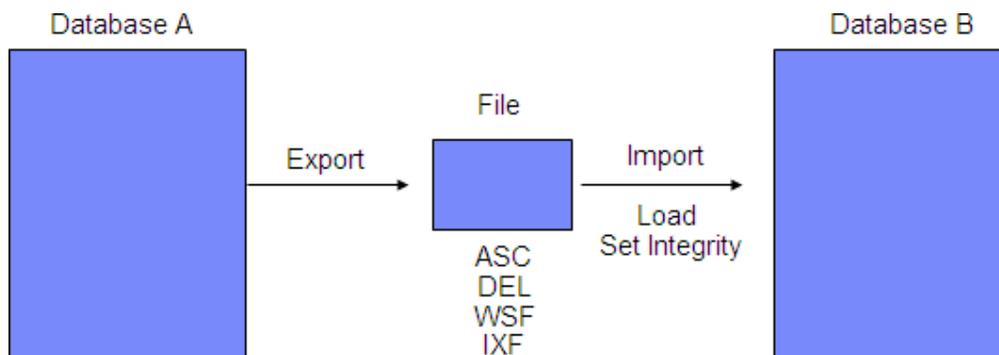


Figura 9.1 – Utilidades de movimiento de datos

En la figura 9.1 hay dos bases de datos, base de datos A y B. Utilizando la herramienta EXPORT, uno puede exportar los datos de una tabla a un archivo. El archivo puede tener cualquiera de estos formatos:

ASC = ASCII
DEL = ASCII delimitado
WSF = Worksheet format
IXF = Formato Integrado de Cambio

Los archivos ASC y DEL son archivos de texto que pueden ser abiertos y revisados en un editor de texto. WSF es un formato que puede ser utilizado para mover datos a hojas de cálculo como Excel o Lotus® 1-2-3. IXF es un formato que no solamente puede incluir los datos sino también el lenguaje de definición de datos (DDL) de la tabla en cuestión. Esto es conveniente porque cuando la tabla necesita ser reconstruida, esto puede ser hecho directamente desde un archivo con formato IXF, mientras esto no sería posible si se utilizaran otros formatos.

Una vez que los datos han sido exportados hacia un archivo, la utilidad IMPORT puede ser usada para importar los datos desde el archivo hacia alguna tabla. La tabla debe existir de

antemano para los formatos ASC, DEL y WSF, pero no es necesario que existan para el formato IXF. Otro método para cargar los datos en una tabla es utilizar la herramienta LOAD. LOAD es mas rápida y va directamente a las páginas de la base de datos sin interactuar con el motor de DB2, sin embargo este método no hace revisión de las restricciones y los disparadores (triggers) no serán lanzados. Para garantizar la consistencia de los datos luego de usar la utilidad LOAD, el comando SET INTEGRITY debe ser utilizado.

La siguiente sección describe las utilidades EXPORT, IMPORT y LOAD en mas detalle.

Nota:

Para mas información acerca de como trabajar con las utilidades de movimiento de datos, vea este video: <http://www.channeldb2.com/video/video/show?id=807741:Video:4262>

9.1 La utilidad EXPORT

La utilidad EXPORT es utilizada para extraer datos desde una tabla a un archivo como fue discutido anteriormente. Detrás de escena, este proceso es una operación SQL de SELECT. El siguiente ejemplo exporta 10 filas de la tabla employee al archivo employee.ixf en formato IXF.

```
EXPORT to employee.ixf OF IXF
  SELECT * FROM employee
  FETCH FIRST 10 ROWS ONLY
```

La tabla employee es parte de la base de datos SAMPLE. Si desea probar la sentencia del ejemplo anterior, necesitaría conectarse primero a esta base de datos.

Si se prefiere trabajar con las herramientas GUI, la utilidad EXPORT también puede ser invocada desde el Centro de Control como se muestra en la figura 9.2

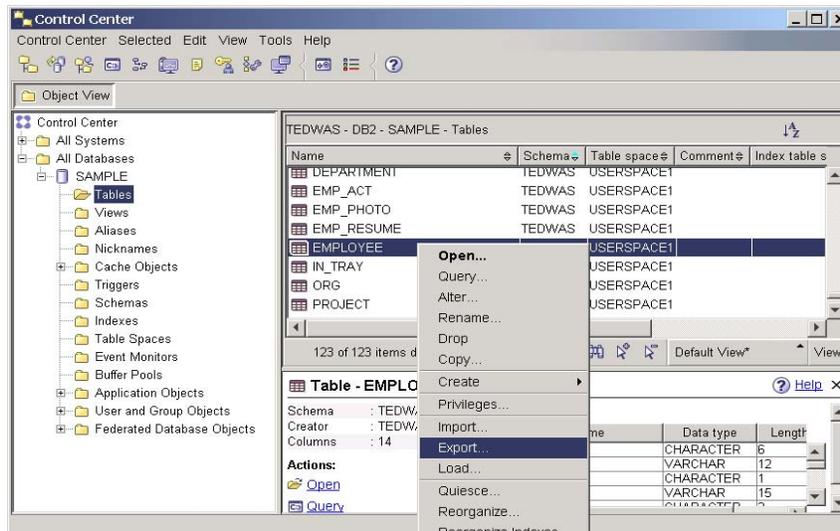


Figura 9.2 – Lanzando el dialogo de tabla EXPORT

Como se muestra en la figura, primero se debe seleccionar la tabla EMPLOYEEES dando clic una vez, y luego con clic derecho en la tabla para obtener un menu emergente desde donde se puede seleccionar la opción Export. Después de seleccionar esta opción, el asistente aparecerá. Simplemente siga los pasos del asistente para completar la operación.

9.2 La utilidad IMPORT

La utilidad IMPORT es utilizada para cargar datos desde un archivo hacia las tablas como se discutió anteriormente. Detrás de escenas, una operación INSERT está siendo en realidad ejecutada. Como una operación INSERT está siendo ejecutada, triggers son activados, restricciones son forzadas inmediatamente y el bufferpool de la base de datos es usado. El siguiente ejemplo carga todos los datos desde el archivo *employee.ixf* con formato IXF dentro de la tabla *employee_copy*. La opción REPLACE_CREATE es una de muchas opciones disponibles en la utilidad IMPORT. Esta opción reemplazará el contenido de la tabla *employee_copy* si la tabla existe previamente antes de que la utilidad IMPORT fuera ejecutada, o creara la tabla y cargara los datos si la tabla no existía aun. Se sugiere probar el ejemplo siguiente, pero es necesario haber ejecutado la utilidad EXPORT en la sección previa.

```
IMPORT FROM employee.ixf OF IXF
REPLACE_CREATE
INTO employee_copy
```

Si se prefiere trabajar desde el Control Center, se puede lanzar la utilidad IMPORT seleccionando cualquier tabla, dando clic derecho en ella, y seleccionando la opción Import como se muestra en la figura 9.3

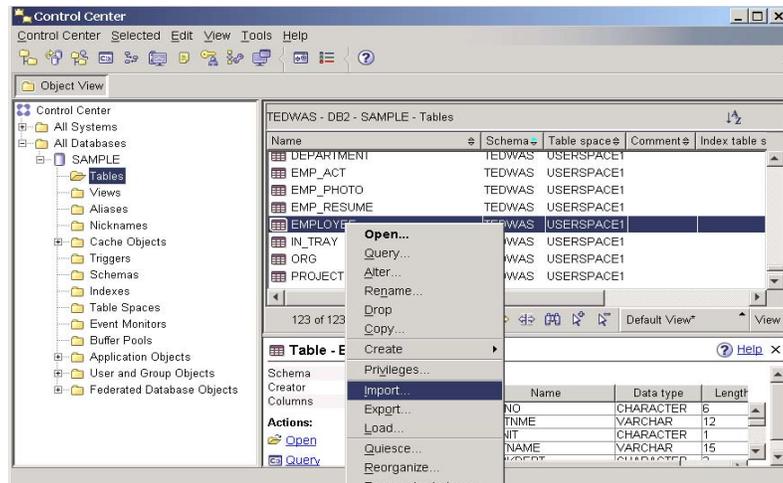


Figura 9.3 – Lanzando el diálogo IMPORT

9.3 La utilidad LOAD

La utilidad LOAD es una manera rápida de cargar datos desde un archivo en la tabla. Como se discutió antes, la utilidad LOAD no va a través del motor de DB2, por lo tanto los triggers no son activados, el bufferpool no es utilizado y las restricciones pueden ser forzadas pero solamente como un paso de separación. Por otro lado, una operación LOAD es más rápida que un IMPORT por ser un cargador de datos de nivel más bajo que accede directamente las páginas en disco. Esto funciona en tres fases: LOAD (carga), BUILD (Construcción) y DELETE (Borrado).

El siguiente ejemplo carga todos los datos desde el archivo *employee.ixf* con formato IXF dentro de la tabla *employee_copy*. La opción REPLACE es una de muchas opciones disponibles con LOAD. En este caso LOAD es usado para reemplazar todo el contenido de la tabla *employee_copy*.

```
LOAD FROM employee.ixf OF IXF
  REPLACE INTO employee_copy
```

Después de ejecutar el comando anterior, el espacio de tablas (table space) donde la tabla reside puede haber sido colocado en un estado CHECK PENDING (verificación pendiente). Esto significa que es necesario ejecutar el comando SET INTEGRITY para verificar la consistencia de los datos. El siguiente ejemplo muestra como hacerlo:

```
SET INTEGRITY FOR employee_copy
  ALL IMMEDIATE UNCHECKED
```

Si se prefiere trabajar desde el Centro de Control, se pueden lanzar las utilidades LOAD y SET INTEGRITY, como se muestra en las figuras 9.4 y 9.5 respectivamente.

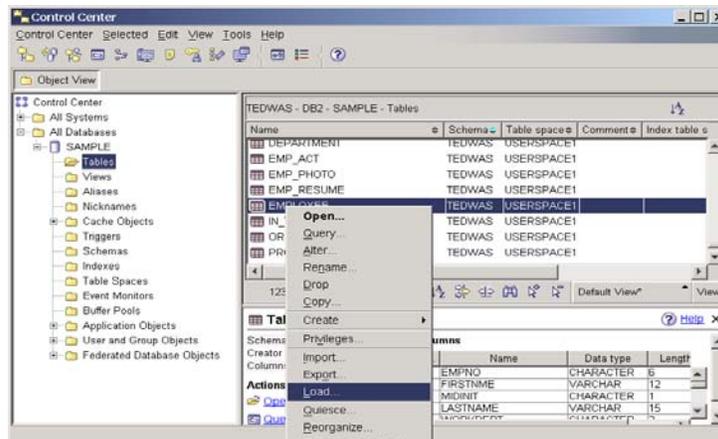


Figura 9.4 – Lanzando el asistente LOAD

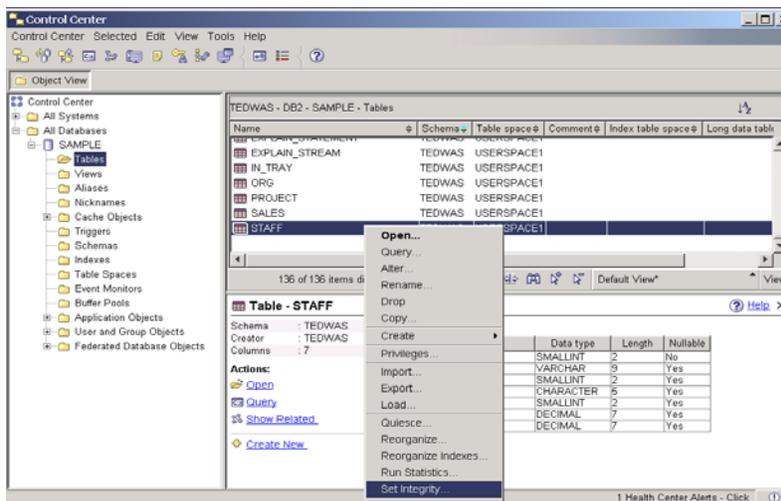


Figura 9.5 – Lanzando el asistente SET INTEGRITY

9.4 La utilidad db2move

Las utilidades EXPORT, IMPORT y LOAD trabajan en una tabla a la vez. Aunque se podría escribir un script para generar los comandos anteriores por cada tabla en una base de datos, la utilidad *db2move* puede realizar esta tarea. La utilidad *db2move* puede trabajar solamente con archivos IXF, y los nombres de los archivos son automáticamente generados por *db2move*. Los siguientes ejemplos muestran como ejecutar *db2move* con las opciones de exportación e importación respectivamente, utilizando la base de datos SAMPLE.

```
db2move sample export
db2move sample import
```

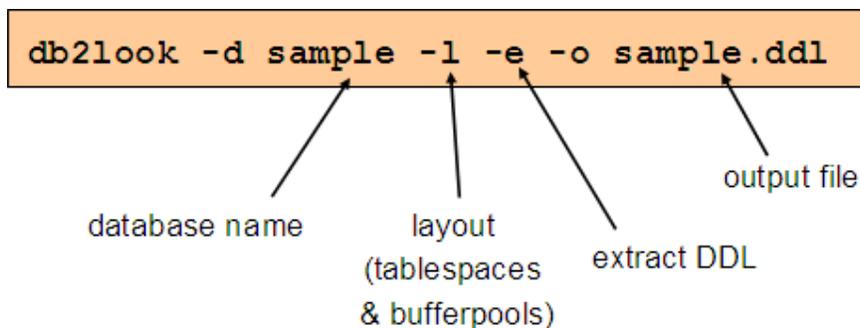
El Centro de control no tiene una opción para db2move.

9.5 La utilidad db2look

Mientras que EXPORT, IMPORT, LOAD y db2move permiten mover datos de una tabla a otra, dentro de una base de datos o a través de diferentes bases de datos, la utilidad db2look puede ser usada para extraer las sentencias DDL, estadísticas de la base de datos y características del espacio de tablas para una base de datos y almacenarlos en un archivo de script que pueda ser mas tarde ejecutado en otro sistema. Por ejemplo, si se quisiera clonar una base de datos desde un servidor DB2 corriendo en Linux a un servidor DB2 corriendo en Windows, se podría primero ejecutar la utilidad db2look en el servidor DB2 en Linux para obtener la estructura de la base de datos y almacenarla en un archivo de script. Entonces se copiaría este archivo al servidor DB2 en Windows, y se ejecutaría el script para comenzar la construcción de la base de datos clonada. En este punto, la estructura de la base de datos ha sido clonada. El siguiente paso sería ejecutar la utilidad db2move con la opción EXPORT en el servidor Linux, y entonces copiar todos los archivos generados al servidor Windows, entonces ejecutar db2move con la opción IMPORT o LOAD. Una vez hecho esto, la base de datos sería completamente clonada de un servidor a otro en plataformas diferentes.

El escenario anterior puede ser necesitado cuando se trabaja con bases de datos en diferentes plataformas como Linux y Windows. Si ambos servidores estan corriendo en la misma plataforma, se preferiría utilizar los comandos backup y restore, los cuales hacen el proceso más sencillo y más directo. Los comandos backup y restore son discutidos con más detalle en el siguiente capítulo de este libro.

El siguiente ejemplo extrae el espacio de tablas y el bufferpool, junto con los enunciados DDL de la base de datos SAMPLE, y los almacena dentro del archivo sample.ddl. Se sugiere ejecutar el comando siguiente y revisar la salida al archivo de texto "sample.ddl".



El comando db2look tiene muchas opciones. Utilice la bandera -h para obtener una breve descripción de las opciones disponibles.

```
db2look -h
```

La utilidad db2look puede también ser invocada desde el Control Center como se muestra en la Figura 9.6

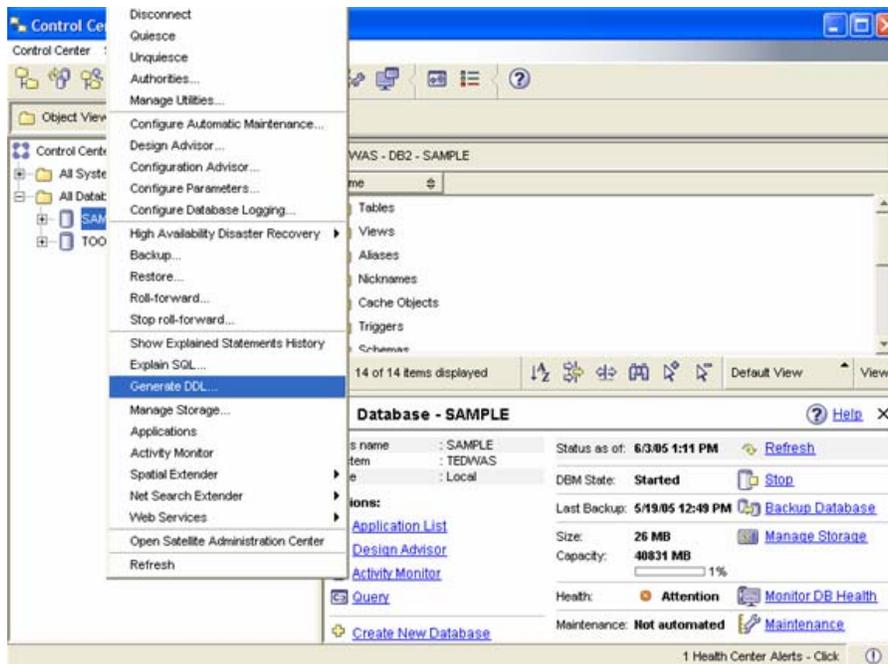


Figura 9.6 – Extrayendo DDL desde el Control Center

En la figura 9.6, seleccione la base de datos desde la cual se quiere obtener el DDL, clic derecho en ella y seleccione “Generate DDL”. Una ventana aparecerá, mostrando varias opciones, como se muestra en la Figura 9.7.

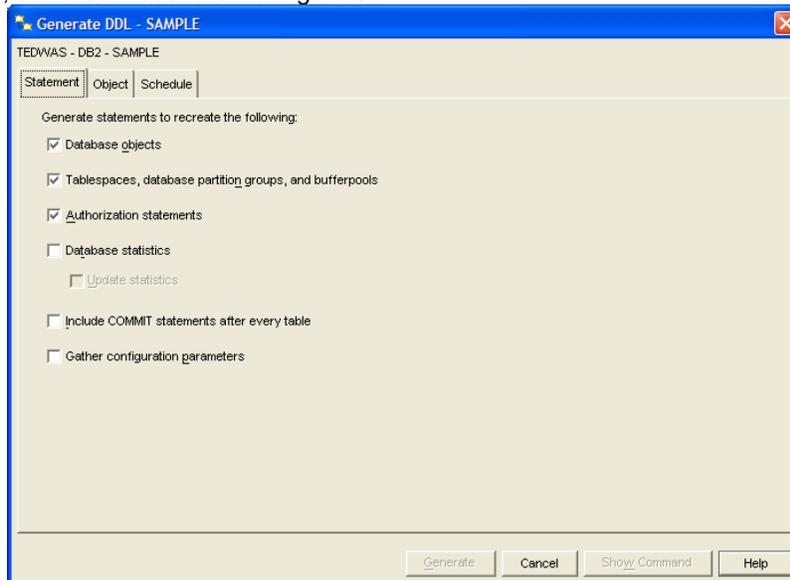


Figura 9.7 – Extrayendo DDL desde el Centro de Control.

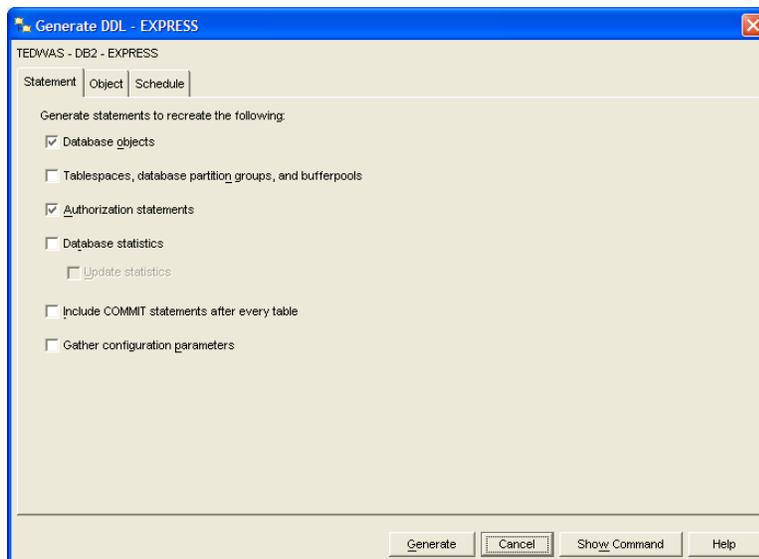
Quicklab #8 – Extrayendo DDL para la base de datos EXPRESS

Objetivo

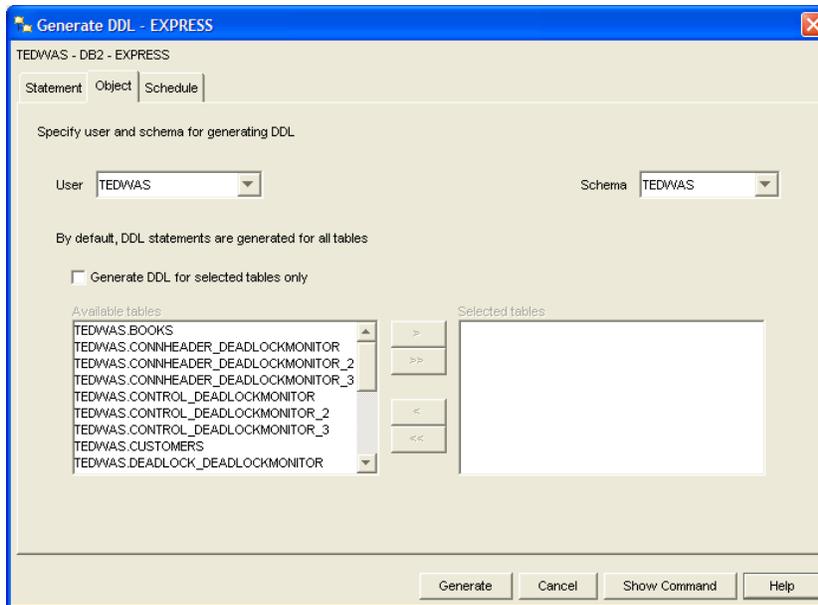
Cuando se clona una base de datos, el objetivo debe ser re-crear la base de datos tan fiel y repetidamente como sea posible. Esto es normalmente hecho utilizando scripts SQL. En este Quicklab, se extraerán las definiciones de los objetos de la base de datos EXPRESS (creada en el Quicklab #2) utilizando el Centro de Control.

Procedimiento

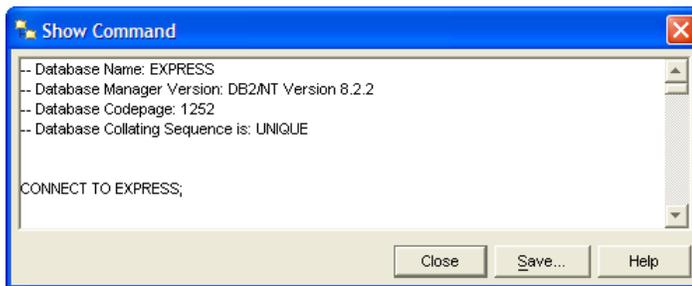
- Abrir el Control Center
- Clic derecho en la base de datos EXPRESS en el árbol de objetos y seleccionar el menu Generate DDL . Esto lanzará una ventana de diálogo.
- En la ventana, especificar las opciones para el DDL generado, como se muestra a continuación. Si se crearon objetos adicionales en el entorno, como son espacios de tablas, buffer pools, etc, se seleccionaran aquí. Las estadísticas de la base de datos no han sido incluidas debido a que el entorno de producción probablemente contendrá un conjunto diferente de estadísticas que el entorno de desarrollo. Similarmente los parámetros de configuración probablemente serán diferentes también. En su entorno, si todo es configurado de forma idéntica al que será desplegado, se puede incluir las opciones adicionales.



- Seleccionar la pestaña Object. Aquí será posible seleccionar de cuales objetos se quiere generar el DDL. En este caso, seleccionar el usuario y el esquema que ha sido usado para crear todos los objetos y generar el DDL de todos los objetos en ese esquema. Clic en el botón *Generate* para comenzar la generación del DDL.



- Revisar el DDL resultante. El resultado del paso anterior es un solo script con todos los enunciados SQL para los objetos seleccionados. Ahora se puede organizar este script en grupos lógicos.
- Crear un directorio llamado [c:\express](#) en el sistema de archivos y guardar el archivo DDL generado en el nuevo directorio en un archivo llamado schema.ddl (clic en el botón Save).



- Abrir el archivo salvado en el Command Editor.
- Aunque en realidad solo se quiere el DDL para las tablas, se puede notar DDL para otros objetos de la base de datos incluidos también. Mover todos los enunciados CREATE TRIGGER a un archivo separado llamado triggers.ddl. Aun cuando quisieramos crear un solo disparador, es en general buena práctica separar los objetos por tipos.
- Ahora se recomienda remover todos los siguientes enunciados:
 4. Enunciados CONNECT TO
 5. Enunciados DISCONNECT
 Se deberían tener dos scripts en este punto:
DDL para tablas, vistas, índices y restricciones

```
C:\express\schema.ddl
DDL para disparadores.
C:\express\triggers.ddl
```

Límpiese el script para el despliegue

5. Remover comentarios innecesarios
6. Separar las funciones y procedimientos en sus propios archivos (útil cuando se tiene una gran cantidad de funciones y procedimientos). Se podría también agruparlos por función o aplicación (e.g. billing.ddl, math.ddl, stringfunc.ddl, etc.)

Se puede haber notado que un carácter especial está siendo usado para delimitar el final de los disparadores, funciones y procedimientos (@). Esto es necesario para diferenciar el final del enunciado CREATE <object> con el final de un enunciado procedural dentro del objeto.

10

Capítulo 10 – Seguridad en la Base de Datos

Este capítulo expone como la seguridad es manejada en DB2. La figura 10.1 provee una visión básica del tema.

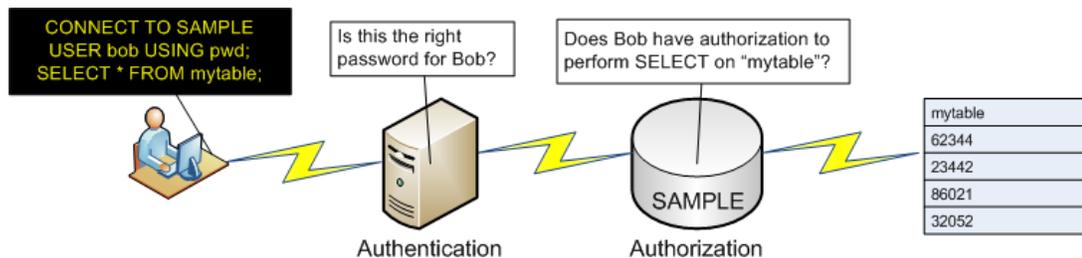


Figura 10.1 – Visión general de la seguridad DB2

Según se muestra en la figura 10.1, la seguridad en DB2 consiste en dos partes:

Autenticación (Authentication en inglés)

Este es el proceso con el que la identidad del usuario es verificada. La Autenticación es realizada por una facilidad de seguridad externa a DB2 (típicamente por el sistema operativo, o por un método de autenticación de red o por un plug-in de autenticación construido por el cliente). La autenticación por defecto es la basada en el sistema operativo. Cuando se usa esta verificación basada en el Sistema Operativo, la identificación del usuario y la contraseña son enviadas al servidor de Base de Datos (por ejemplo, como parte de la sentencia de conexión). El servidor de base de datos entonces invoca la autenticación del Sistema Operativo para verificar el id de usuario y la contraseña.

Autorización (Authorization en inglés)

En esta etapa, DB2 chequea si el usuario autenticado puede realizar la operación requerida. La información de autorización es almacenada en el catálogo de DB2 y en el archivo de configuración DBM.

Por ejemplo, en la Figura 10.1, el usuario "bob" se conecta a la base de datos SAMPLE con esta sentencia:

```
CONNECT TO sample USER bob USING pwd
```

Ambos, "bob" y "psw" son pasados al sistema operativo o a la aplicación externa de autenticación para solicitar la aprobación de autenticación, verificando que existe un usuario definido como "bob" y que la contraseña coincide con la del usuario. Si esta parte es cumplida exitosamente, el sistema operativo retornará el control de seguridad a DB2. Como siguiente paso el usuario "bob" ejecutará una sentencia tal como:

```
SELECT * FROM mytable
```

Ahora DB2 toma el control de seguridad para realizar el chequeo de autorización y confirma que el usuario "bob" tiene privilegios de realizar un SELECT en la tabla "mytable". Si el chequeo de autorización falla, DB2 retornará un mensaje de error, de otra forma la sentencia será ejecutada.

Nota:

Para mas información sobre como funciona la seguridad en DB2, observe este video:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4267>

10.1 Autenticación

Aunque la autenticación es ejecutada por el sistema operativo (u otra facilidad externa de seguridad), DB2 decide en que nivel ocurrirá esta autenticación.

El parámetro del DBM CFG llamado "AUTHENTICATION", en el servidor DB2, tiene un rango de valores posibles. Por ejemplo, cuando el parámetro es fijado en SERVER (que es el de defecto), la autenticación es realizada por el Sistema Operativo o la aplicación externa de seguridad en el servidor. Pero si AUTHENTICATION es fijado en CLIENT, la autenticación es realizada por el Sistema Operativo o la aplicación externa de seguridad en el cliente. Esto se muestra en la Figura 10.2.

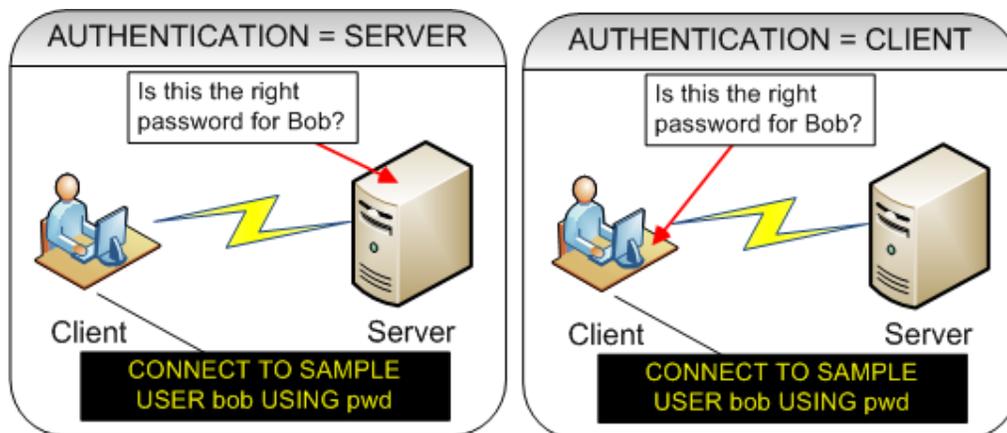


Figura 10.2 – Donde se lleva a cabo la autenticación

El parámetro AUTHENTICATION puede fijarse en cualquiera de los valores listados en la Tabla 10.1

Comando	Descripción
SERVER (default)	La autenticación es realizada en el servidor
CLIENT	La autenticación es realizada en el cliente
SERVER_ENCRYPT	Es como SERVER excepto que el Id del usuario y las passwords son transmitidas encriptadas.
KERBEROS	La Autenticación se realiza con un mecanismo de seguridad Kerberos.
SQL_AUTHENTICATION_DATAENC	La autenticación del Server mas las conexiones deben usar encriptación de datos
SQL_AUTHENTICATION_DATAENC_CMP	Es como la anterior, excepto que la encriptación es usada solamente cuando está disponible
GSSPLUGIN	La Autenticación usa un mecanismo externo de seguridad plug-in basado en la API GSS

Tabla 10.1 – Valores validos del parámetro AUTHENTICATION

10.2 Autorización

La autorización consiste en otorgar privilegios y autoridades a usuarios o grupos, y son almacenados en tablas del sistema administrados por DB2.

Un privilegio permite a un usuario ejecutar un tipo simple de operación contra la base de datos, como CREATE, UPDATE, DELETE, INSERT, etc.

Una autoridad es un rol predefinido consistente en varios privilegios. La figura 10.3 muestra las diferentes autoridades y privilegios en DB2.

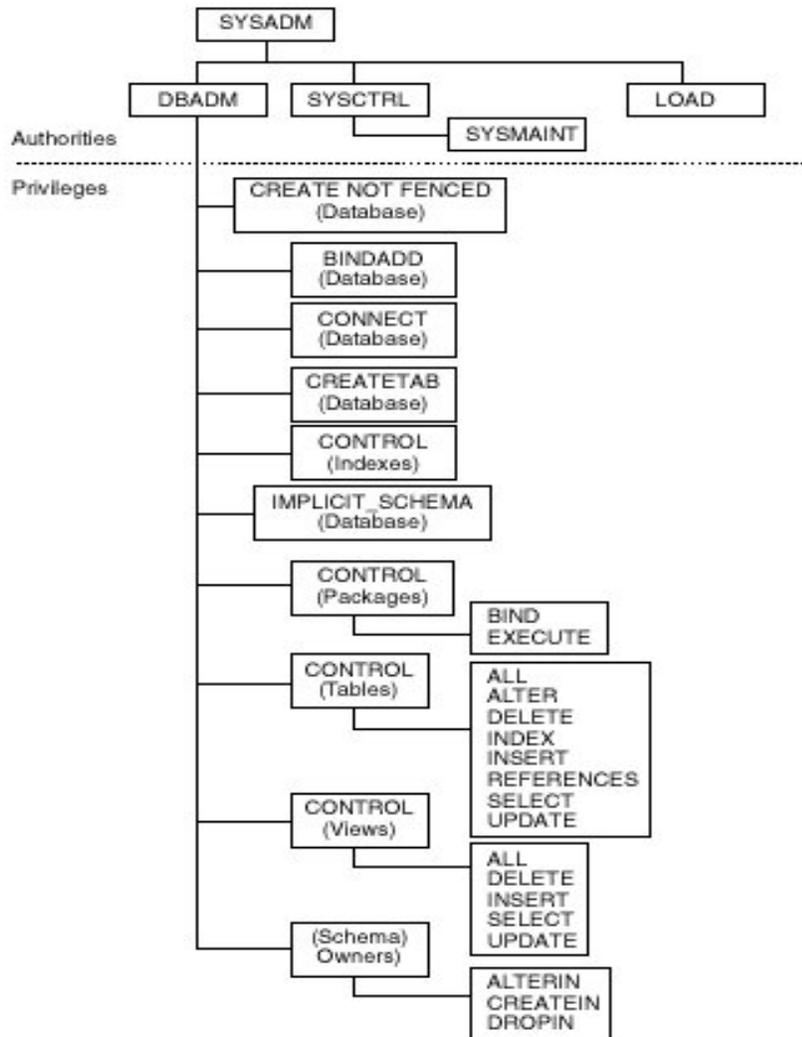


Figura 10.3 – Autoridades y privilegios

La Tabla 10.2 muestra las diferentes funciones que cada autoridad puede ejecutar. Como se puede ver, SYSADM tiene la mayor autoridad, mientras SYSMON tiene la menor.

Función	SYSADM	SYSCTRL	SYSMAINT	SYSMON	DBADM	LOAD
Actualizar los DBM CFG	Y					
Grant/revoke DBADM	Y					
Establecer/cambiar SYSCTRL	Y					
Establecer/cambiar SYSMAINT	Y					
Establecer/cambiar SYSMON	Y					
Forzar a usuarios a salir de la base	Y	Y				
Create/drop database	Y	Y				
Restaurar una nueva database	Y	Y				
Actualizar los DB CFG	Y	Y	Y			
Hacer Backup database/table space	Y	Y	Y			
Restaurar a una database existente	Y	Y	Y			
Realizar recuperacion con roll-forward	Y	Y	Y			
Iniciar/Detener instante	Y	Y	Y			
Restaurar table space	Y	Y	Y			
Ejecutar trace	Y	Y	Y	Y		
Obtener snapshots de monitoreo	Y	Y	Y			
Consultar el estado del table space	Y	Y	Y			
Podar los archivos de historia de log	Y	Y	Y			
Hacer 'Quiesce' sobre un table space	Y	Y	Y		Y	Y
Ejecutar LOAD de tablas	Y				Y	Y

Poner/quitar estado de chequeo pendiente	Y		Y
Create/drop monitores de eventos	Y		Y

Tabla 10.2 – Autoridades DB2 y privilegios

Para otorgar las autoridades de SYSADM, SYSCTRL o SYSMANT a un grupo, los parámetros del DBM CFG: SYSADM_GROUP, SYSCTRL_GROUP, y SYSMANT_GROUP, pueden ser asignados a un grupo de sistema operativo.

Por ejemplo, para darle la autoridad de SYSADM al grupo “db2admns” del sistema operativo, se puede usar este comando:

```
update dbm cfg using SYSADM_GROUP db2admns
```

Cada instancia DB2 tiene su propia definición de grupo de autoridades.

En Windows, estos parámetros están vacíos por defecto, lo que significa que el grupo local Administrators de Windows será SYSADM. En Linux el grupo dueño de la instancia será SYSADM.

10.3 Autoridad DBADM

La autoridad DBADM (DataBase ADMinistrator) es un super usuario para la base de datos. No es autoridad al nivel de la instancia, por lo tanto no se lo incluyó en la sección anterior. Para otorgar la autoridad de DBADM, use la sentencia GRANT como se muestra en el ejemplo abajo.

```
connect to sample
grant DBADM on database to user <userid>
```

En el ejemplo de arriba, es necesario primero conectarse a la base, en este ejemplo, la base de datos “SAMPLE”, y entonces se puede otorgar DBADM a un usuario. Para otorgar la autoridad DBADM es necesario ser SYSADM.

Note que un DBADM no puede crear table spaces, pese a que son objetos dentro de una base de datos, debido a que un Table space maneja contenedores o discos y buffer pools o memoria que son los recursos físicos del sistema.

10.4 El grupo PUBLIC

DB2 define un grupo interno llamado PUBLIC. Cualquier usuario identificado por el sistema operativo o servicio de autenticación de red, es implícitamente un miembro del grupo PUBLIC. Cuando una base de datos es creada, ciertos privilegios son otorgados automáticamente a PUBLIC:

- CONNECT,
- CREATETAB,
- IMPLICIT_SCHEMA,
- BINDADD

Si no se desea que PUBLIC tenga estos privilegios, recomendamos revocar todos los privilegios del grupo, como se muestra abajo:

```
REVOKE CONNECT          ON DATABASE FROM PUBLIC
REVOKE CREATETAB       ON DATABASE FROM PUBLIC
REVOKE IMPLICIT_SCHEMA ON DATABASE FROM PUBLIC
REVOKE BINDADD         ON DATABASE FROM PUBLIC
```

10.5 Las sentencias GRANT y REVOKE

Las sentencias GRANT y REVOKE son parte de los estándares de SQL y se utilizan para otorgar o quitar privilegios de un usuario o grupo. A continuación hay algunos ejemplos de estas sentencias:

Para otorgar el privilegio de SELECT sobre la tabla T1 al usuario USER1:

```
GRANT SELECT ON TABLE T1 TO USER user1
```

Para otorgar todos los privilegios sobre la tabla T1 al grupo GROUP1:

```
GRANT ALL ON TABLE T1 TO GROUP group1
```

Para quitar todos los privilegios sobre la tabla T1 del grupo GROUP1:

```
REVOKE ALL ON TABLE T1 FROM GROUP group1
```

Para otorgar el privilegio EXECUTE sobre el procedimiento p1 al usuario USER1:

```
GRANT EXECUTE ON PROCEDURE p1 TO USER user1
```

Para quitar el privilegio EXECUTE sobre el procedimiento p1 del usuario USER1:

```
REVOKE EXECUTE ON PROCEDURE p1 FROM USER user1
```

10.6 Chequeo de Autorizaciones y privilegios

El modo más sencillo para chequear autorizaciones y privilegios es a través del Centro de Control. En la figura 10.4 se muestra como lanzar el diálogo de Privilegios de tabla para la tabla EMPLOYEE desde el Centro de control.

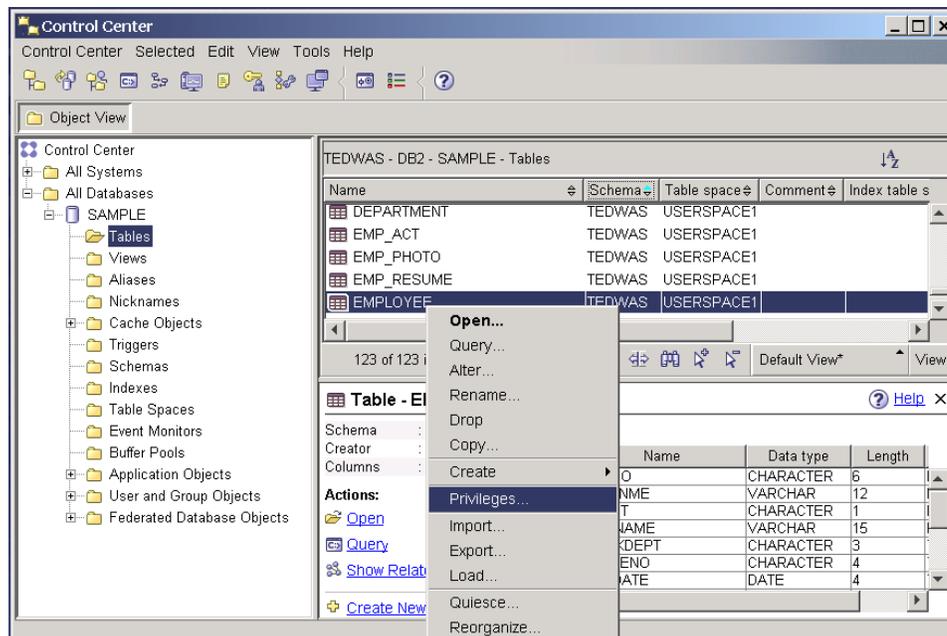


Figura 10.4 – Lanzando el dialogo de Privilegios de Tabla.

Como se ve en la figura 10.4, se selecciona la tabla deseada, con clic en el botón derecho, y se elige Privilegios. Una vez elegido la caja de dialogo aparece como se ve en la figura 10.5. En esta figura también explica los diferentes campos y elementos de esta caja de dialogo.

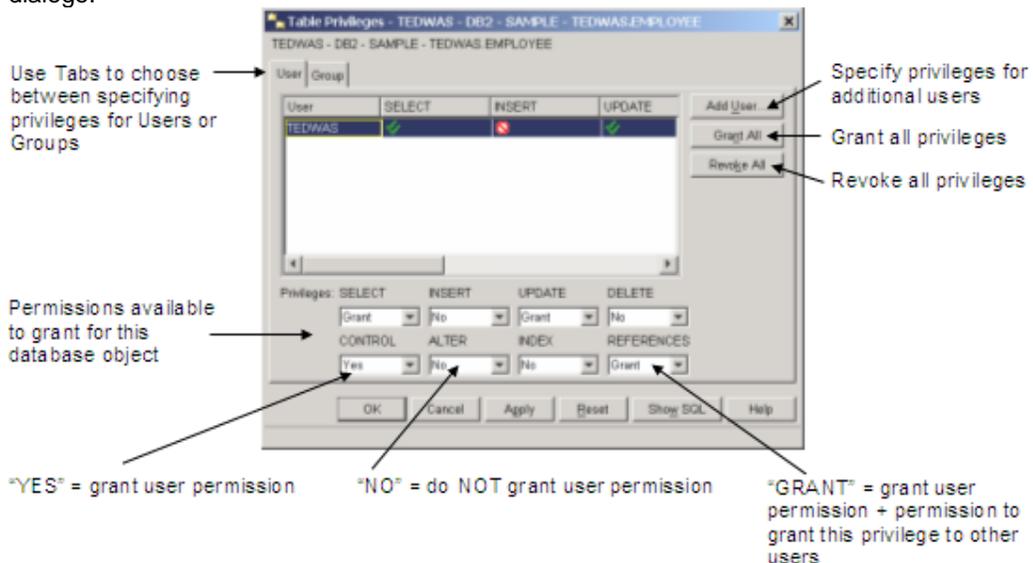


Figura 10.5 – La caja de dialogo de Privilegios de tabla

Alternativamente, se puede consultar las vistas SYSCAT del catálogo de DB2 que contienen la información de las autorizaciones. Por ejemplo, si se quisiera conocer si el usuario DB2ADMIN tiene el privilegio de hacer SELECT en la tabla T2, y se desea conocer también, quién ha otorgado este privilegio, se puede ejecutar una consulta como esta:

```
SELECT grantor, grantee, selectauth
   FROM syscat.tabauth
  WHERE tabname = 'T2'
```

GRANTOR	GRANTEE	SELECTAUTH
ARFCHONG	DB2ADMIN	Y

En el ejemplo, el usuario ARFCHONG otorgó el privilegio de SELECT al usuario DB2ADMIN.

10.7 Consideraciones sobre privilegios del grupo

Para hacer mas sencilla la administración de DB2, distribuya a los usuarios en grupos, y entonces ejecute la sentencias grant para aquellos grupos que lo requieran.

Cuando a un grupo se le otorgan privilegios, a los miembros del grupo se les otorgan 'implícitamente' los privilegios heredados a través de la pertenencia al grupo.

Cuando un usuario es removido de un grupo, pierde los privilegios implícitos, pero mantiene los privilegios que les fueron explícitamente otorgados previamente. Los privilegios que fueran expresamente otorgados a un usuario deben ser explícitamente revocados del usuario.

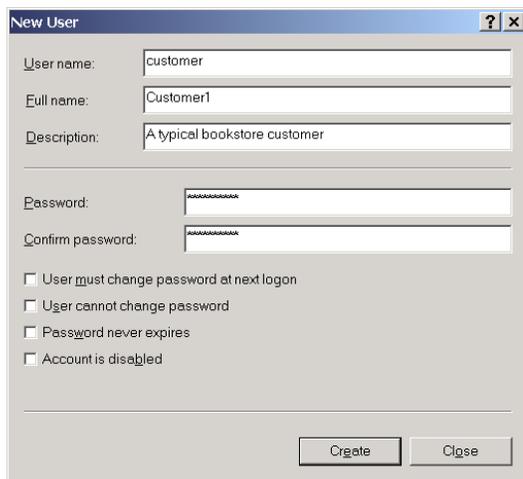
Quicklab #9 – Otorgando y quitando permisos al usuario

Objetivo

Antes se usó la cuenta de administrador de instancia (SYSADM) para realizar todos los comandos de la base de datos. Esta cuenta tiene acceso completo a todas las utilidades, datos y objetos de la base de datos. Por lo tanto es muy importante resguardar esta cuenta para evitar pérdidas accidentales o deliberadas. En la mayoría de los casos, es deseable crear diferentes cuentas del usuario o grupos con un conjunto limitado de permisos. En este laboratorio, se creará una cuenta nueva de usuario y luego le asignará sus privilegios específicos.

Procedimiento

1. Abra la consola de Administrador de Windows haciendo clic-derecho en el ícono *Mi Computadora* en el escritorio y seleccione el ítem Administrar del menú.
2. Expanda la selección de *Herramientas del sistema* en el árbol del panel izquierdo de la ventana y luego expanda la carpeta *Usuarios y Grupos Locales*. Haga Clic-derecho en la carpeta *Usuario* y seleccione el ítem *Nuevo Usuario*.
3. En la ventana de diálogo *Nuevo Usuario*, ingrese la siguiente información: En el campo *Nombre del usuario*, ingrese "customer" y en el campo *Nombre completo*, ingrese "Customer1". En el campo *Descripción*, ingrese "Un cliente típico de almacén de libros". En los campos *Contraseña* y *Confirme la contraseña*, ingrese "ibmdb2". Remueva el check de la opción *El usuario debe cambiar la contraseña en el próximo logon*, y haga clic en el botón *Crear* para crear el nuevo usuario.

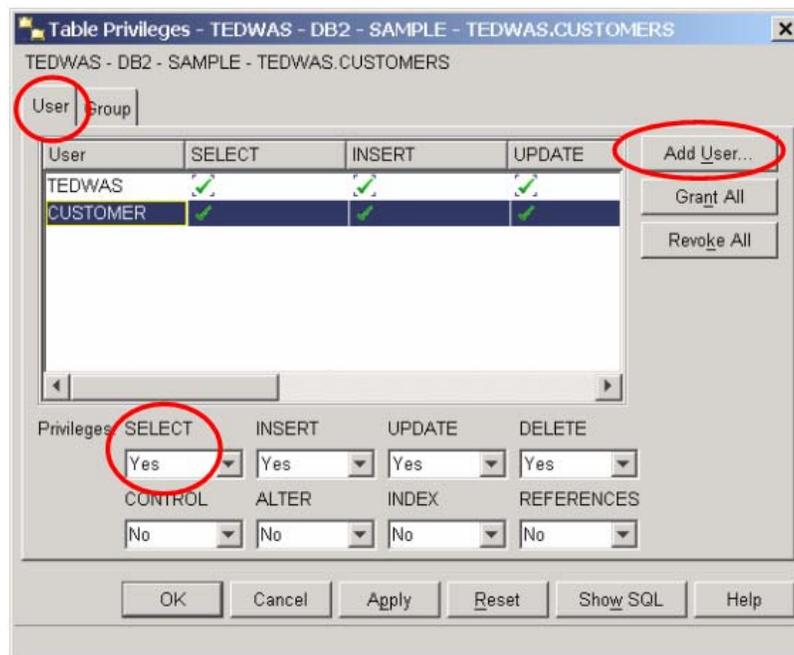


The image shows a 'New User' dialog box with the following fields and options:

- User name: customer
- Full name: Customer1
- Description: A typical bookstore customer
- Password: [masked]
- Confirm password: [masked]
- User must change password at next logon
- User cannot change password
- Password never expires
- Account is disabled

Buttons: Create, Close

4. Asegúrese que está utilizando la vista avanzada del Centro de control de DB2. Para cambiar a la vista avanzada, seleccione el ítem de menú *Ajustar el Centro de Control Center* desde el menú de Centro de Control. Seleccione la opción *Avanzada* y haga clic en el botón *OK*.
5. Expanda el árbol de objetos del Centro de Control en el panel izquierdo a *Todas las Bases de Datos > EXPRESS > Tablas*.
6. Otorgue con Grant los privilegios requeridos al usuario recientemente creado. Desde la lista de tablas en la base de datos *EXPRESS*, haga clic-derecho en la tabla *CUSTOMERS*, y seleccione el ítem *Privilegios* para ver la ventana de diálogo de Privilegios de Tabla.
7. Haga clic en el botón *Agregar Usuario* y seleccione al usuario *customer* recién creado. Haga clic en el botón *OK* para cerrar el dialogo *Agregar Usuario*.
8. Notará que el usuario *customer* ha sido agregado a la lista de usuarios, pero que no tiene privilegios asignados. Para otorgarle con grant los privilegios SELECT, INSERT, UPDATE y DELETE al usuario, cambie cada caja a Yes. Un cliente de internet debería poder ver, agregar, actualizar los datos de su cuenta. No le otorgaremos al usuario otros permisos, debido a que no los requerirá. Haga clic en el botón *OK* para cerrar la ventana de dialogo de Privilegios de Tabla y aceptar los cambios realizados.



9. Repita los pasos 7-9 para las tablas *BOOKS* y *SALES*. Para la tabla *BOOKS*, solamente otorgue con grant el privilegio de *SELECT* porque el usuario *customer* debería no serle posible modificar ninguno de los datos del inventario de libros. Para la tabla *SALES*, debería solamente otorgarle con grant los privilegios de *SELECT* e *INSERT*. El usuario *customer* no debería tener los privilegios de *DELETE* o *UPDATE* debido a que solamente los empleados del almacén de libros deberían acceder a modificar las transacciones de ventas.

10. Conectarse a la Base de Datos usando el usuario *customer* creado anteriormente. Trate de hacer *SELECT* a la tabla cliente. Qué sucede? Trate de hacer *DELETE* o *UPDATE* sobre la tabla *SALES*. Qué sucede?

En esta práctica, solamente hemos creado un usuario, sin embargo, su aplicación puede tener diferentes tipos de usuarios. Experimente creando otros usuarios y asignándoles privilegios. Usted puede también crear grupos de usuarios asignándoles privilegios a los grupos, en vez de a cada usuario individualmente.

11

Capítulo 11 – Backup y Recupero

En este capítulo discutimos el registro de cambios o “logging” de la base de datos DB2, como hacer una copia completa o parcial de su base de datos usando la utilidad BACKUP y como recuperar sus datos usando la utilidad RESTORE.

Nota:

Para mayor información sobre el registro de cambios o logging, backup y recovery, mire este video:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4282>

11.1 Registro de cambios o Logging de la Base de datos

Si se esta trabajando con un editor de texto, cada vez que quiere asegurarse que un documento es guardado, se hace clic en el botón “Guardar”. En el mundo de las bases de datos la sentencia COMMIT hace exactamente eso. Cada vez que una sentencia SQL como UPDATE, DELETE o INSERT es ejecutada, se garantiza que cualquier cambio hecho a los datos son guardados con un COMMIT.

De modo similar, cuando se trabaja en un documento de texto, algunas veces puede ver en el rincón inferior derecho un breve mensaje que dice “Auto Guardando” o “Auto Saving”. En el mundo de las bases de datos, esto sucede también, debido a que cualquier cambio realizado a los datos, tales como un UPDATE, INSERT o DELETE, serán guardado en algún lugar mientras se realizan.

Ese “algún lugar” del párrafo anterior se refiere a los logs de la base de datos (database logs). Los logs de la base de datos son almacenados en disco y son usados para registrar las acciones de las transacciones. Si hay un crash de la base de datos o del sistema (el sistema se cae), los logs se utilizan para reproducir nuevamente o re hacer (redo) las transacciones confirmadas (las que fueron COMMIT) durante el recupero.

La figura 11.1 provee una visión gráfica de lo que sucede cuando se esta trabajando con una base de datos en términos de logging.

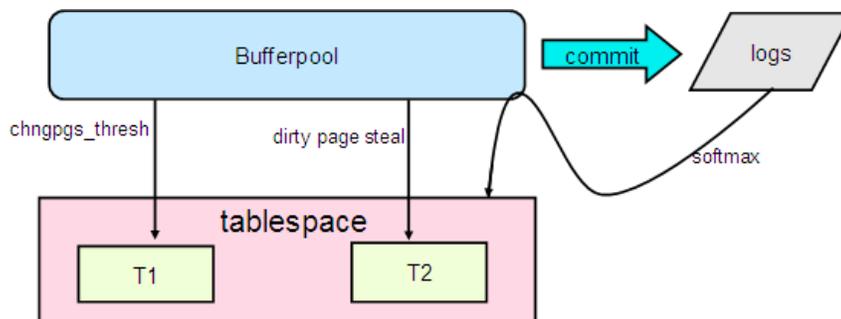


Figura 11.1 – El proceso de logging de la Base de Datos

En la figura 11.1, vemos un table space y la estructura llamada log. Ambos residen en discos, aunque recomendamos que no sean mantenidos en el mismo disco. Cuando una operación de UPDATE se lleva a cabo por ejemplo, las paginas de la/s fila/s involucrada/s son leídas y puestas en el buffer pool (memoria). Los cambios del update serán realizados en este área de memoria y los viejos y nuevos valores se almacenarán en los archivos de logs; algunas veces inmediatamente y en otras oportunidades, cuando los buffers de logs (no mostrados en la figura, pero también en memoria) se llenan. Si se ejecuta una sentencia COMMIT luego del UPDATE, los valores nuevos y los anteriores se almacenarán en los archivos de log inmediatamente. Este proceso es repetido para muchas otras operaciones SQL realizadas en la base de datos. Sólo cuando ciertas condiciones son cumplidas, tal como alcanzar un límite predefinido de páginas cambiadas, especificado con el parámetro CHNGPGS_THRES, las páginas en el buffer pool se “externalizarán” o serán escritas en el disco del table space. El parámetro CHNGPGS_THRES indica el porcentaje del buffer pool con páginas “sucias”, esto es, páginas conteniendo cambios.

Desde un punto de vista de performance, no tiene sentido ejecutar dos escrituras por cada operación de escritura para cada operación de COMMIT: una al escribir en los logs y otra en escribir en el disco del table space; por eso la “externalización” de los datos hacia el disco del table space solamente ocurre cuando el valor del parámetro de límite predefinido “chngpgs_thres” es alcanzado.

11.2 Tipos de logs

Hay dos tipos de logs:

Logs Primario

Estos están pre-alojados y el número de logs primarios disponibles esta determinados por el parámetro LOGPRIMARY del DB CFG.

Logs Secundario

Estos son dinámicamente creados cuando son necesarios para DB2. El número máximo de logs secundarios esta fijado por el parámetro LOGSECOND del DB CFG. Crear dinámicamente un log es costoso; por lo tanto, para las operaciones diarias, asegúrese que se la mayoría de transacciones entre en los logs primarios. Los archivos de log secundarios son borrados cuando todas las conexiones a la base son terminadas.

El logging Infinito es posible si se fija el parámetro LOGSECOND a un valor de -1; sin embargo, esto no es recomendado y puede quedarse sin espacio en el file system.

11.3 Tipos de logging

Hay dos tipos de logging: circular (el de defecto) y archivado.

11.3.1 Logging Circular

La figura 11.2 demuestra como funcionan los logging circular.

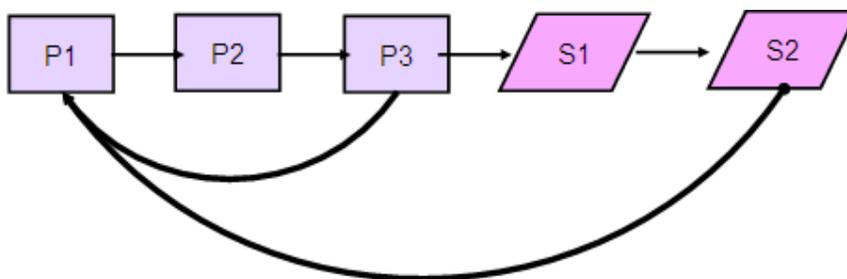


Figure 11.2 – Trabajando con logs primarios y secundarios

En la figura 11.2 hay 3 logs primarios P1, P2 y P3, por lo que podemos afirmar que LOGPRIMARY ha sido configurado a 3. Para simplicidad, digamos que solamente una transacción está siendo ejecutada en este ejemplo. A medida que la transacción esta siendo ejecutada, el espacio comienza a ser llenado en el log file P1 y luego en el P2. Si un COMMIT ocurriese y la información es externalizada al disco del table space, entonces P1 y P2 pueden ser reescritas, debido a que la información no es mas necesaria para un recuperio de un caída de sistema (lo que será discutido en mas detalle mas adelante). Por otro lado, si la transacción es tan grande que usa los log files P1, P2, P3 y aún necesita mas espacio de log, porque aún no ha sido COMMIT ni externalizada, entonces un log secundario (S1 en la figura) sera dinámicamente creada. Si la transacción aún continúa, mas logs secundarios son creados hasta llegar a la máxima cantidad especificada por el parámetro LOGSECOND. Si se necesitan aún mas logs, un mensaje de error indicando que se ha llegado a una condición de logs llenos, se retornará al usuario y la transacción será deshecha con rollback.

11.3.2 Logging de Archivado o Retención de log

En el logging de archivado, también conocido como log de retención, los archivos de log no son sobre escritos sino que son mantenidos en línea o fuera de línea. Los logs de archivado Online son mantenidos con los logs activos que son necesarios para recuperación de una caída del sistema. Los logs de archivado Offline son movidos a otro medio como cintas, y esto puede ser hecho con rutinas del USEREXIT. Para activar el logging de archivado se debe fijar el parámetro db cfg LOGRETAIN en YES.

El logging de archivado es normalmente usado en sistemas de producción, y ya que los logs son mantenidos, esto permite un recuperado de base de datos hasta el último cambio almacenado en el último archivo de log en la mayoría de las situaciones. Con este modo de logging, un DBA puede recuperarse (hasta alguna extensión) de errores causados por humanos. Por ejemplo, si un usuario inadvertidamente inicia la ejecución de un proceso que realiza una transacción errónea por días, cuando el problema es detectado mas tarde, el DBA podría restaurar el sistema hacia atrás en el tiempo antes que el problema fuera introducido. Sin embargo, puede haber alguna manipulación manual requerida para re-ejecutar la transacción correctamente.

El logging de archivado es requerido para hacer recuperado hacia adelante o roll forward recovery y backup en línea. Figure 11.3 muestra el proceso de logging de archivado.

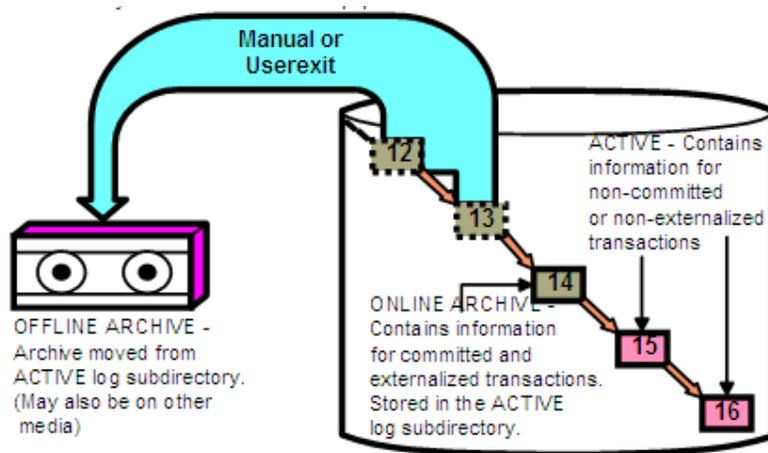


Figura 11.3 – Logging de Archivado

11.4 Logging de Bases de datos desde el Centro de Control

Se puede configurar el logging de la base de datos desde la Centro de control haciendo clic derecho en la base de datos en cuestión y seleccionando “Configurar el Logging de Base de datos”. Esto se muestra en la figura 11.4

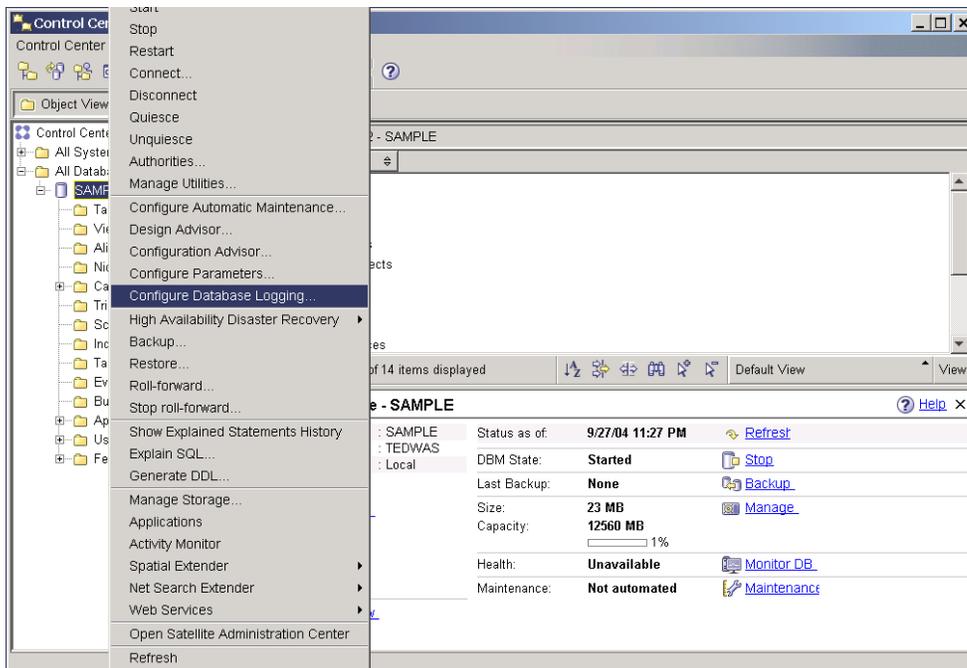


Figura 11.4 – Configuración del logging de la base de datos desde la centro de control.

La figura 11.5 muestra el asistente de Logging de la Base de datos, donde se puede elegir el logging circular o de archivado.

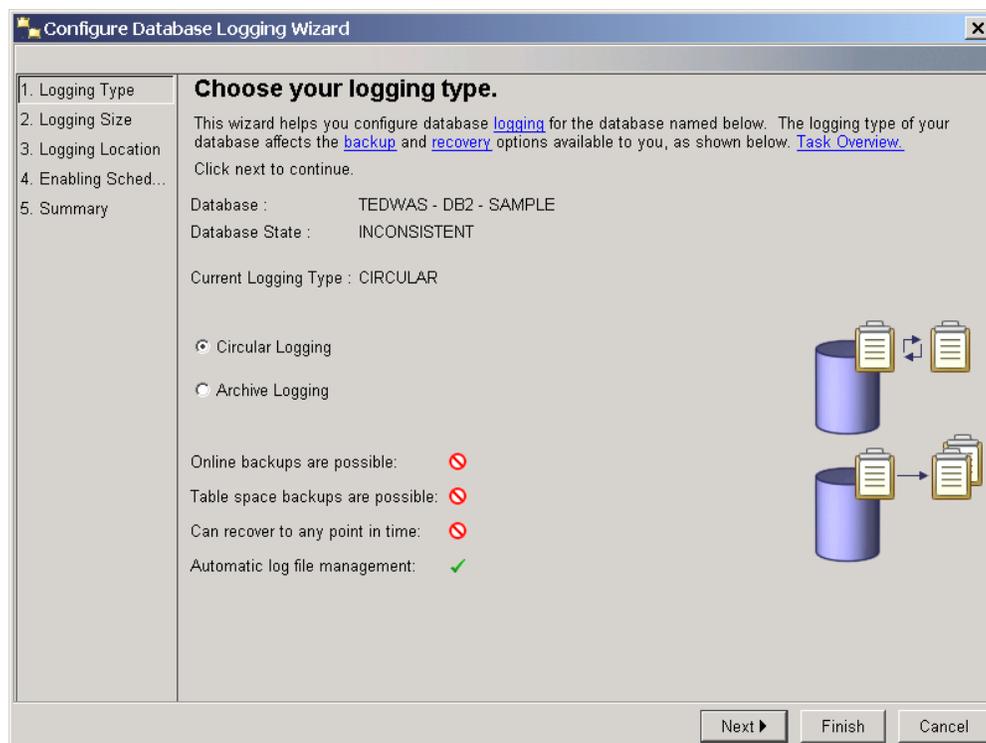


Figura 11.5 – Asistente de Logging de Base de datos

11.5 Parámetros de Logging

Hay una importante cantidad de parámetros DB CFG relacionados con logging. La tabla 11.1 lista los parámetros principales.

Parámetro	Descripción
logbufsz	El monto de la memoria para ser usada como buffer para los registros de log antes de escribirlos en disco
logfilsz	El tamaño de cada log configurado, en cantidades de paginas de 4KB
logprimary	La cantidad de logs primarios de tamaño logfilsz que seran creados
logsecond	El número de archivos de log secundarios que son creados y usados para recupero si es necesario.
logpath/newlogpath	La ubicación en el cual los logs activos y los logs en el futuro archivados son ubicados.
mirrorlogpath	Para proteger los logs en el camino o path de los logs primarios de fallas de disco o borrados accidentales, se pueden especificar que un conjunto identico de logs sea mantenido en una senda o camino secundario o espejo.
loghead	El nombre del archivo log que es que está actualmente activo

userexit	Permite a los programas userexit copiar los logs fuera de línea
softmax	Limita el costo de un recuperado de una caída de sistema
logretain	Activa el modo de Logging de Archivado
overflowlogpath	Similar a la opción OVERFLOW LOG PATH del comando ROLLFORWARD; sin embargo, en vez de especificar la opción OVERFLOW LOG PATH para cada comando ROLLFORWARD, se puede fijar este parámetro de configuración una sola vez.
blk_log_dsk_ful	Fijado, permite prevenir errores de disco lleno cuando DB2 no puede crear un nuevo log file en la senda o path activo. En cambio, DB2 intentará crear el archivo de log cada 5 minutos hasta que pueda realizarlo. Las operaciones de SQL de sólo lectura podrán continuar una vez desbloqueado.
max_log	Porcentaje de espacio activo máximo de un log por transacción
num_log_span	Número de log files activos por cada UOW (unidad de trabajo) activo

Tabla 11.1 – parámetros de Logging

11.6 Backup de Base de datos

El comando de Backup de DB2 permite tomar una copia instantánea de su base de datos en el momento en que el comando es ejecutado. La sintaxis mas simple que puede usar para ejecutar este comando es:

```
BACKUP DATABASE <nombre de la base de datos> [ TO <ruta> ]
```

La mayoría de los comandos y utilidades pueden ser ejecutados en línea o fuera de línea. En línea implica que otros usuarios pueden estar conectados y realizando operaciones en la base de datos mientras se ejecuta el comando. Fuera de línea significa que ningún otro usuario esta conectado a la base de datos mientras se realiza esta operación. Para permitir una operación backup en línea, agregue la palabra ONLINE a la sintaxis del comando, de otro modo, por defecto el comando asumirá que se esta ejecutando fuera de línea.

Por ejemplo, si quiere hacer backup de la base simple en la senda C:\BACKUPS, se puede realizar este comando en la ventana de comandos DB2 en Windows:

```
db2 BACKUP DB sample TO C:\BACKUPS
```

Note que el directorio C:\BACKUPS debe existir antes de ejecutar el comando. También asegúrese que no hay conexiones a la base de datos cuando se ejecute el comando anterior, por que de otro modo recibirá un mensaje de error, debido a que no se puede realizar un backup fuera de línea cuando hay conexiones.

Para encontrar si hay conexiones a la bases de datos en una instancia, provea este comando desde la ventana de comandos de DB2 o desde el shell en Linux:

```
db2 list applications
```

Para forzar todas las conexiones de todas las bases de datos en una instancia, ejecute este comando:

db2 force applications all

Probablemente éste no es un buen comando a utilizar en un entorno de producción, de otro modo va a recibir las llamadas furiosas de sus compañeros de trabajo! Note también que el último comando corre asincrónicamente, lo que significa que cuando trate de ejecutar el comando backup inmediatamente después, pueda que reciba un error indicando que aún hay conexiones a la base de datos. Espere unos segundos y repita el comando backup si recibió un error la primera vez.

Luego de una exitosa ejecución del comando de backup, un nuevo archivo conteniendo una imagen de backup se habrá creado. El nombre de este archivo sigue la convención mostrada en la Figura 11.6.

Linux/UNIX/Windows

Diagrama que muestra la convención de nombres de archivos de backup de DB2. El nombre del archivo es DBALIAS.0.DB2INST.NODE0000.CATN0000.20060314131259.001. Las partes del nombre están etiquetadas: Alias (DB), Instance (2), Type (0), Node (INST), Catalog Node (NODE0000), Year (2006), Day (03), Month (14), Hour (13), Minute (12), Second (59), y Sequence (001).

Figura 11.6 – Convención del nombre de la imagen de Backup

Un tipo "0" significa que el backup es completo. El nodo es fijado en NODE0000 para bases de datos no particionadas, que es el caso de todas las ediciones de DB2 excepto DB2 Enterprise Edition con la característica DPF. El nodo del catálogo es también fijado en CATN0000. Refiérase a los manuales de DB2 para más detalles.

Cuando varios backups son tomados y almacenados en el mismo directorio, la marca de hora o timestamp al final del nombre del archivo es usada para distinguir entre las imágenes de backup. Como veremos en la próxima sección, el comando RESTORE puede usar la marca de hora para restaurar desde un archivo específico.

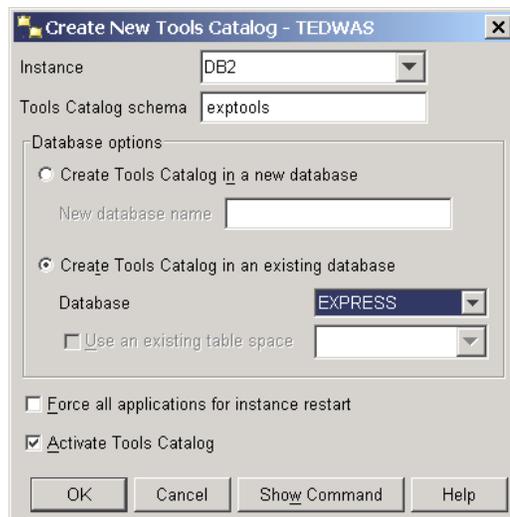
Quicklab #10 – Programando un backup

Objetivo

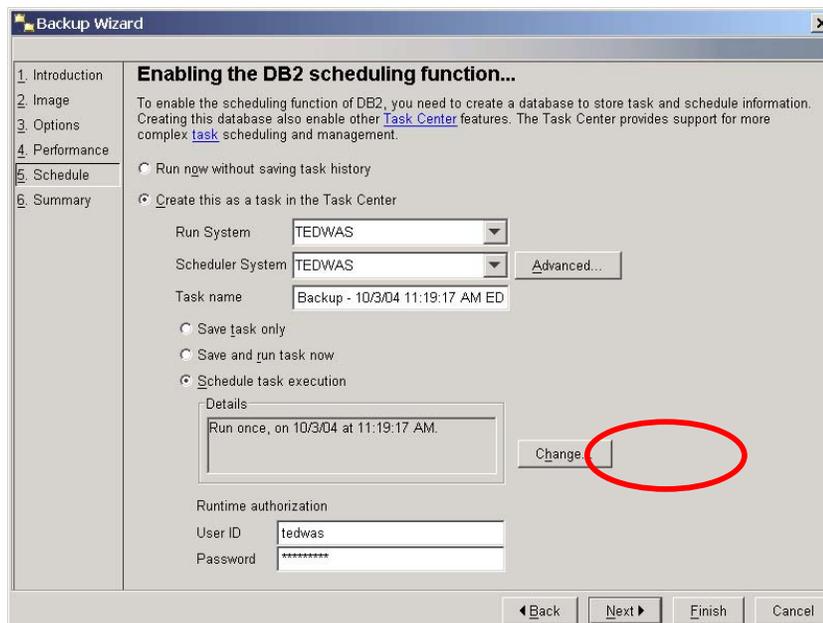
Aunque DB2 es capaz de automatizar varias actividades de mantenimiento, en algunos momentos se desea indicar específicamente cuando se deben realizar ciertas actividades. En este Quicklab, usted creará un programa específico para el usuario (customized) de backup nocturno para la base de datos *EXPRESS*.

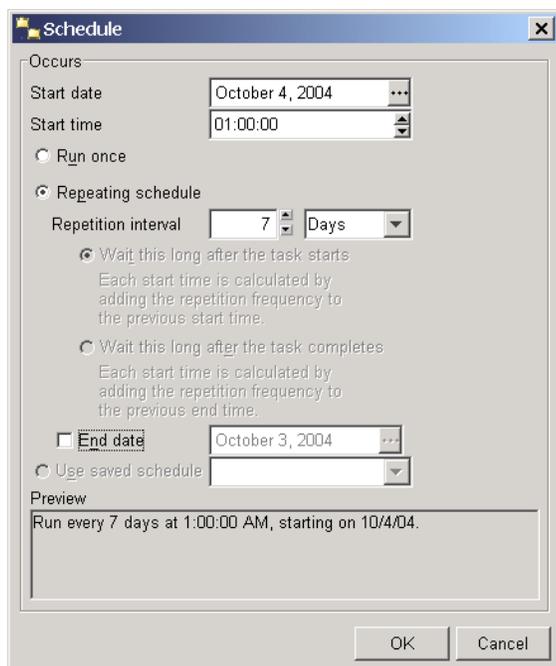
Procedimiento

1. Desde el árbol de objetos en el Centro de control, navegue a Centro de control => Todas las Bases de datos. Haga clic-derecho en la base de datos *EXPRESS* y seleccione el ítem de *Backup*. Esto inicia el asistente de *Backup*.
2. La página de introducción del asistente resume el estado actual de la base de datos, incluyendo la fecha del último backup realizado y el método de logging actualmente utilizado. Haga clic en el botón *Next* para moverse a la próxima página del asistente.
3. En la página *Imagen* del asistente, seleccione el destino de la Imagen del Backup. Usualmente seleccionará un dispositivo diferente al que almacena a la base. Por ahora cree una nueva carpeta en el file system llamada *C:\db2backup*, y especifique que carpeta será la ubicación del backup. En el asistente, seleccione el ítem de *File System* desde la lista *Tipos de Medio*. Haga clic en el botón *Add*, seleccione la carpeta recientemente creada, luego haga clic en el botón *OK*. Haga clic en el botón *Siguiente* para pasar a la siguiente página del asistente.
4. Se puede revisar las páginas *Opciones* y *Performance*, pero las opciones de defecto son usualmente suficientes ya que DB2 automáticamente realizará el backup del modo mas óptimo posible. Navegue a la página *Schedule* cuando finalice la exploración.
5. En la página *Schedule*, si el programador (scheduler) no esta activado, elija para activarlo ahora. Seleccione el sistema para crear el catalogo de herramientas y cree uno nuevo. Especifique un esquema para el catálogo de herramientas y elija crearlo en la base de datos *EXPRESS*. Las herramientas del catálogo contiene los metadatos sobre las tareas programadas. Haga clic en el botón *OK* para continuar. Haga clic en el botón *Siguiente* para pasar a la siguiente pagina del asistente, un vez que el catalogo de herramientas ha sido creado.



6. En la página *Schedule*, elija crear un programa o schedule para la ejecución de las tareas. Programe el backup para correr cada día, comenzando a las 1AM. Haga clic en el botón *Siguiente* para moverse a la próxima página.





7. En la página *Sumario*, usted puede revisar las tareas programadas que serán creadas. Cuando haya revisado los cambios haga clic en el botón *Finalizar* para crear la tarea.
8. Inicie el Centro de Tareas (Task Center) para ver o modificar la tarea de Backup recientemente creada.

11.7 Recupero de la Base de Datos

Un recuperado de base de datos implica restaurar la base de datos desde un Backup y/o los logs. Si solamente restaura de un backup, estaría recreando la base de datos tal cual existió en el momento en que fue realizado el backup.

Si el logging de archivado está activo antes del backup, se podría no solamente restaurar desde la imagen del backup, sino que también desde los logs. Como veremos en la próxima sección, un recuperado roll-forward permite restaurar desde un backup y luego aplicar (roll-forward) los logs hasta el fin de estos, o hasta un determinado punto específico en el tiempo.

Note que el término “recupero” es usado muchas veces en esta sección, pero el comando utilizado para el recuperado es ‘RESTORE’.

11.7.1 Tipos de Recuperado

Hay tres tipos de recuperado:

- **Recupero de una caída o de un reinicio**

Asuma que está trabajando en una computadora de escritorio, ejecutando una transacción importante en una base de datos. Repentinamente un corte de luz o alguien desconecta el cable de electricidad: ¿qué le pasará a la base?

La próxima vez que arranque la computadora e inicie DB2, el recuperado de la caída se ejecutará automáticamente. En este recuperado, DB2 automáticamente correrá el comando RESTART DATABASE y leerá los logs activos. DB2 rehará (redo) y deshará transacciones basadas en estos logs. Cuando se completa este comando, DB2 garantiza que las bases de datos están consistentes, esto es, lo que haya sido confirmado con commit estará guardado y aquello que no haya sido confirmado se habrá deshecho.

- **Recupero de una imagen o Versión**

Este tipo de recuperado implica que está restaurando solamente de una imagen de Backup, por lo tanto su base de datos será puesta en el estado que estaba al momento de realizar el backup. Cualquier transacción realizada en la base de datos después de que el backup fuera tomado, se habrá perdido.

- **Recupero con Roll-forward**

Con este tipo de recuperado, no solamente se hace RESTORE desde una imagen de backup, sino que también se corre el comando ROLLFORWARD para aplicar los logs sobre el backup. De esta manera, se puede recuperar a un punto especificado en el tiempo. Este tipo de recuperado minimiza la pérdida de datos.

11.7.2 Restauración de Base de datos

Use el comando RESTORE para recuperar una base de datos desde una imagen de Backup. La siguiente sintaxis es la más simple que puede ser usada para este comando:

```
RESTORE DATABASE <nombre de la base> [from <ruta>] [taken at
<timestamp>]
```

Por ejemplo, si se tiene un archivo de imagen de backup de la base *Sample* con este nombre:

Alias	Instance			Year	Day	Minute	Sequence
SAMPLE.0	DB2INST	NODE0000	CATN0000	2006	0314	131259	.001
	Type	Node	Catalog Node	Month	Hour	Second	

Puede ejecutar el siguiente comando:

```
RESTORE DB sample FROM <ruta> TAKEN AT 20060314131259
```

11.8 Otras operaciones con BACKUP y RESTORE

La siguiente es una lista de algunas de las cosas que también puede hacer con los comandos BACKUP y RESTORE. Le sugerimos que revise los manuales DB2 para detalles adicionales.

- Hacer Backup de una base de datos en una instancia de 32-bit, y restaurarla en una instancia de 64-bit
- Restaurar sobre una base de datos existente
- Uso de una restauración redirigida cuando se esta restaurando dentro de un sistema donde hay diferentes cantidades de discos de los especificados en la imagen del backup
- Backup o Recupero solo de un tablespace, en vez restaurar toda la base de datos.
- Los Backups Delta e Incremental están permitidos. Los backup delta registran sólo los cambios desde el ultimo backup al presente, mientras que el backup incremental registra todos los cambios y los acumula en cada imagen del backup.
- Backup desde una copia flash (requiere hardware adecuado)
- Recuperar tablas eliminadas con Drop (si la opción fue activada para una tabla determinada)
- Realizar un backup desde una plataforma (por ejemplo, Windows) y ser restaurada en otra plataforma (por ejemplo, Linux) no es posible y debe usarse db2look y db2move para este escenario.

12

Capítulo 12 – Tareas de Mantenimiento

Este capítulo discute alguna de las tareas requeridas para mantener en buena forma a su base de datos. En general, en DB2 Express-C al igual que en las otras ediciones de DB2, se automatiza la mayoría de estas tareas. Esta capacidad de auto administración es un gran beneficio para compañías de pequeño y mediano tamaño, que no cuentan con suficientes recursos económicos para contratar a un Administrador de Bases de Datos (DBA). Por otro lado, si un DBA es contratado, este tendrá mas tiempo libre para realizar actividades avanzadas que agregarán valor a la compañía.

Nota:

Para mas información sobre tareas de mantenimiento, observe este video:
<http://www.channeldb2.com/video/video/show?id=807741:Video:4302>

12.1 REORG, RUNSTATS, REBIND

Estas son las tres tareas principales en DB2, como se describe en la Figura 12.1: REORG, RUNSTATS y REBIND.

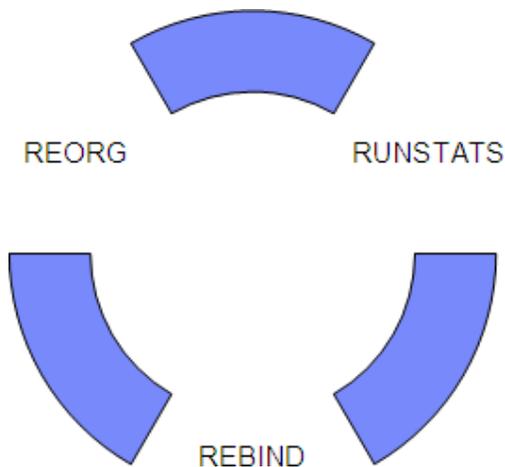


Figure 12.1 – Tareas de Mantenimiento: REORG, RUNSTATS, REBIND

La Figura 12.1 nos muestra que las tareas de mantenimiento son realizadas de manera circular. Si se ejecuta un REORG, es recomendable correr también un RUNSTATS, seguido por un REBIND. En general, las tablas en una base de datos se ven modificadas por operaciones de UPDATE, DELETE e INSERT en manera constante. Luego de algún tiempo es bueno correr estas operaciones de mantenimiento. El ciclo de mantenimiento puede ser iniciado con un REORG.

12.1.1 El comando REORG

Con el tiempo, las operaciones INSERT, UPDATE y DELETE pueden causar fragmentación de los datos guardados en una base de datos. Esto significa que los datos aparecerán en diversas páginas internas en el disco y no de manera contigua. El comando REORG toma el espacio desperdiciado y reorganiza los datos para hacer la extracción más eficiente. La mayoría de las tablas que son frecuentemente modificadas se beneficiarán al ejecutar REORG. Usted puede hacer REORG tanto en índices como en tablas y este comando REORG puede ser ejecutado en línea (online) o fuera de línea (offline).

El comando REORG es más rápido y eficiente si es offline, pero no permite el acceso a la tabla, mientras que la versión online de REORG permite el acceso a la tabla, pero puede consumir una gran cantidad de recursos del sistema; trabaja mucho mejor para tablas pequeñas.

Sintaxis:

```
REORG TABLE <nombre de tabla>
```

Ejemplo:

```
REORG TABLE empleado
```

El comando REORGCHK puede usarse antes del comando REORG para determinar si una tabla o índice necesitan ser reorganizados.

12.1.2 El comando RUNSTATS

El optimizador de DB2 es el "cerebro" de DB2. Este encuentra el camino de acceso más eficiente para localizar y extraer los datos. El optimizador se basa en el costo de la localización y extracción de datos en el sistema y usa estadísticas que son almacenadas en tablas de catálogo para calcular el camino de menor costo. Las tablas de catálogo tienen estadísticas sobre cuántas columnas tiene una tabla, cuántas filas hay, cuántos y que tipos de índices hay por cada tabla y otra información al respecto.

La información de las estadísticas no es actualizada dinámicamente. Esto es por diseño, ya que no es deseable que DB2 actualice las estadísticas para cada operación realizada en la base por que afectaría negativamente la performance general. En cambio, DB2 provee el comando RUNSTATS para actualizar estas estadísticas. Correr este comando es esencial para mantener las estadísticas actualizadas. El optimizador DB2 puede hacer cambios radicales en el camino de acceso si sabe que la tabla tiene una fila o si tiene un millón de filas. Cuando las estadísticas de la base de datos están actualizadas, DB2 puede elegir el mejor plan de acceso. La frecuencia en que se recomienda correr el RUNSTATS depende de que tan seguido se realizan cambios en los datos.

Sintaxis:

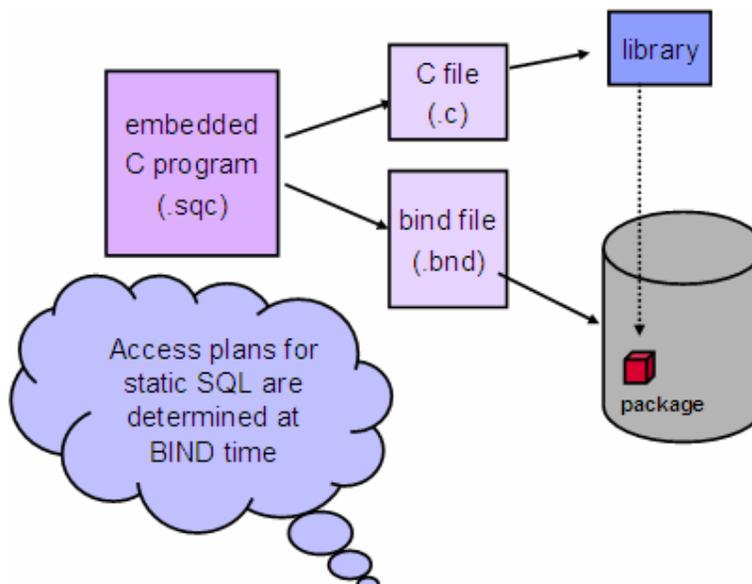
```
RUNSTATS ON TABLE <esquema.nombredetabla>
```

Ejemplo:

```
RUNSTATS ON TABLE miesquema.empleado
```

12.1.3 BIND / REBIND

Luego de correr exitosamente un comando RUNSTATS, no todas las consultas usarán las últimas estadísticas. Los planes de acceso del SQL estático son definidos cuando se ejecuta un comando BIND. Las estadísticas usadas cuando se ejecutó el BIND no serán siempre las mismas que las actuales. La figura 12.2 ayuda a ilustrar esta idea.



La Figura 12.2 – Proceso bind de SQL Estático

En la figura 12.2 un programa C embebido (almacenado como un archivo con una extensión “sqc”) es precompilado. Luego de la pre-compilación, dos files son generados, un archivo “.c” que contiene el código C con todas las sentencias SQL comentadas; y un archivo “.bnd” conteniendo todas las sentencias SQL. El archivo C con la extensión “.c” es compilado con cualquier compilador C, creando una “biblioteca” o “library” como se muestra en el vértice superior derecho de la figura. El archivo “.bnd” es ‘compilado’ en forma similar con un comando BIND, generando un paquete que se almacena en la base de datos. “Binding” es equivalente a compilar las sentencias SQL donde el mejor plan de acceso es determinado, basándose en las estadísticas en ese momento y luego guardadas en el paquete.

Ahora, ¿qué pasa si son insertadas un millón de filas en la tabla que se usa en el SQL del programa C embebido? Luego de la inserción, si un RUNSTATS es ejecutado, las

estadísticas serán actualizadas, sin embargo el package no será actualizado para recalculer el camino de acceso basado en las últimas estadísticas. El comando `db2rbind` puede utilizarse para 'rebind' todos los paquetes existentes para que tomen en cuenta las últimas estadísticas.

Sintaxis:

```
db2rbind database_alias -l <logfile>
```

Ejemplo:

Para que todos los paquetes de la base de datos *sample* sean alcanzados por un rebind general y almacene el resultado en el archivo `mylog.txt`, ejecute este comando:

```
db2rbind sample -l mylog.txt
```

12.1.4 Tareas de mantenimiento desde el Centro de control

Desde el Centro de Control usted puede correr REORG y RUNSTATS. La figura 12.3 muestra cómo.

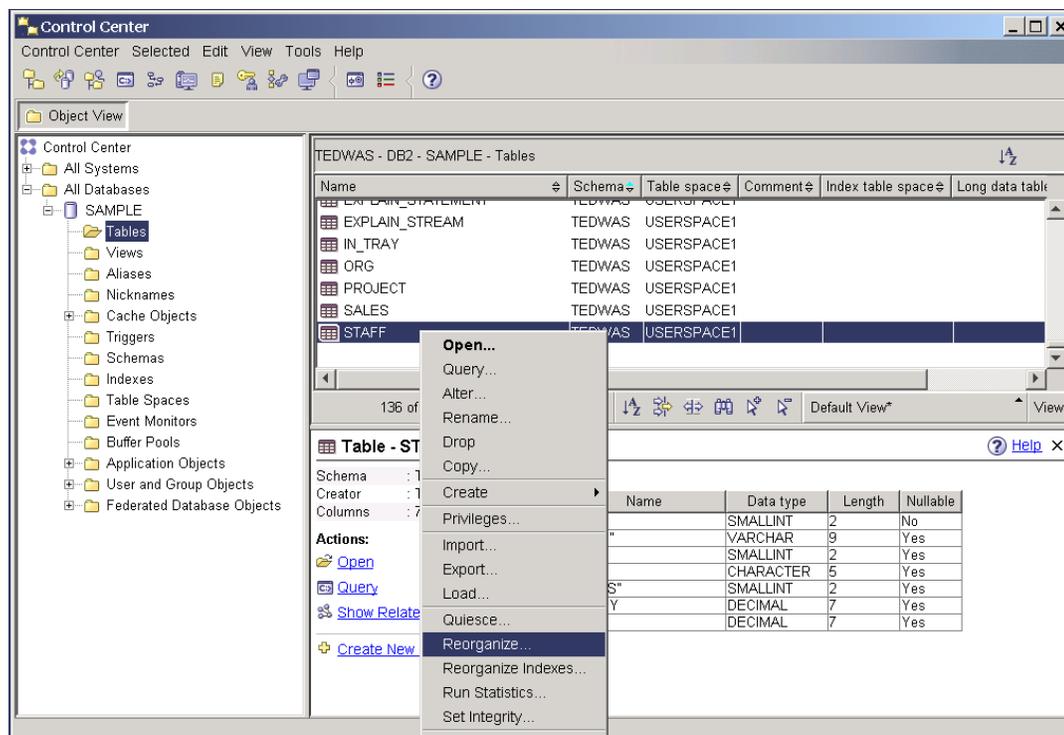


Figura 12.3 – REORG y RUNSTATS desde el Centro de control

Al elegir la tabla sobre la que quiere operar, haga clic botón derecho y seleccione Reorganizar (para ejecutar REORG) o Correr Estadísticas (para ejecutar RUNSTATS).

La vista operacional de la base de datos

Cuando selecciona una base de datos, la vista operacional de la base de datos en el lado derecho de el Centro de control proveerá de información sobre la Base de Datos, tales como su tamaño, cuando fue por ultima vez respaldada y si el mantenimiento automático esta configurado, etc. Esta vista permite identificar rápidamente las necesidades de mantenimiento para su Base de Datos. La figura 12.4 muestra esta información.

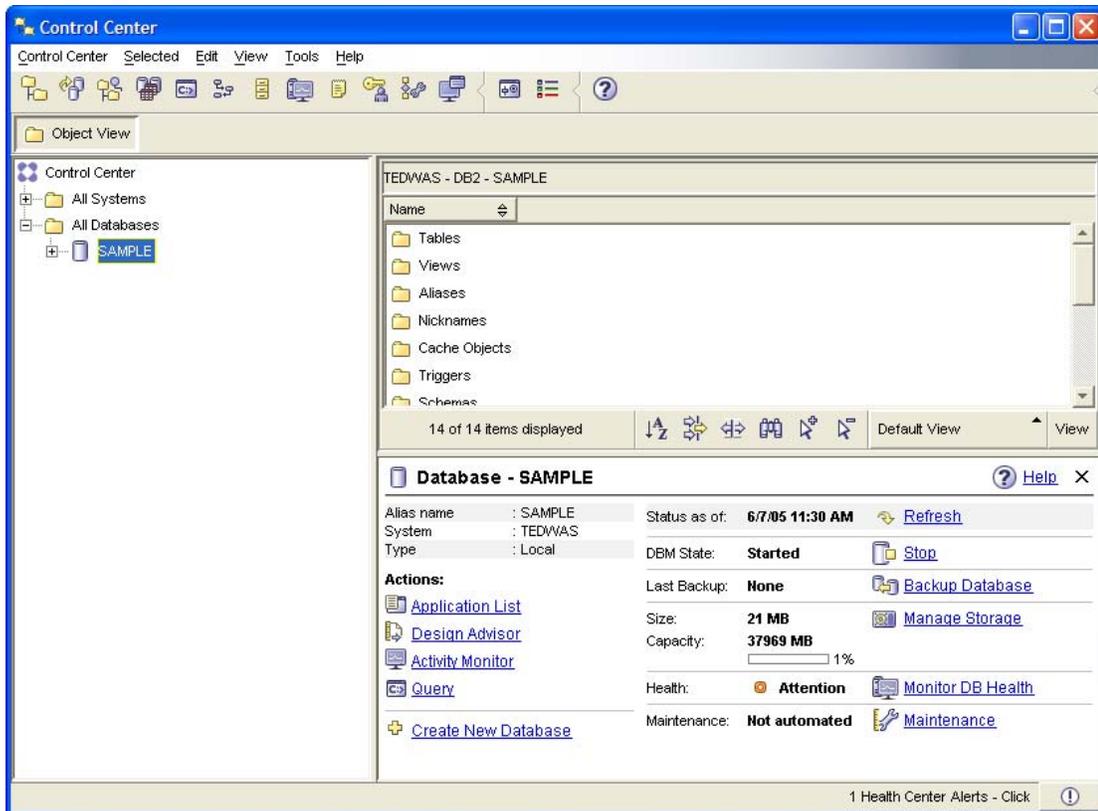


Figura 12.4 – La vista operacional de la Base de datos desde el Centro de Control

12.2 Las opciones de Mantenimiento

Hay tres modos para realizar tareas de mantenimiento:

1. **Mantenimiento Manual**
 Cuando el mantenimiento es requerido, se hace manualmente en el momento que se necesita.
2. **Crear scripts para realizar el mantenimiento**
 Se pueden crear scripts con los comandos de mantenimiento, y programar su ejecución periódica.

3. Mantenimiento Automatizado

Haga que DB2 automáticamente realice el mantenimiento por si mismo (REORG, RUNSTATS, BACKUP)

En esta sección nos concentraremos en el mantenimiento automatizado.

El mantenimiento automático consiste en lo siguiente:

- El usuario define una *ventana de mantenimiento* donde las tareas pueden ser ejecutadas con una mínima disrupción del funcionamiento operativo. Por ejemplo, si el sistema tiene menor actividad los domingos de 2:00am a 4:00am, este marco de tiempo funcionará como ventana de mantenimiento.
- Hay dos ventanas de mantenimiento: una para las operaciones en línea (online) y otra para las operaciones fuera de línea (offline).
- DB2 ejecutará las operaciones de mantenimiento automáticamente solamente cuando sea necesario durante la ventana de mantenimiento.

Desde el Centro de control, se puede lanzar el asistente de configuración del Mantenimiento automático como se muestra en la figura 12.5.

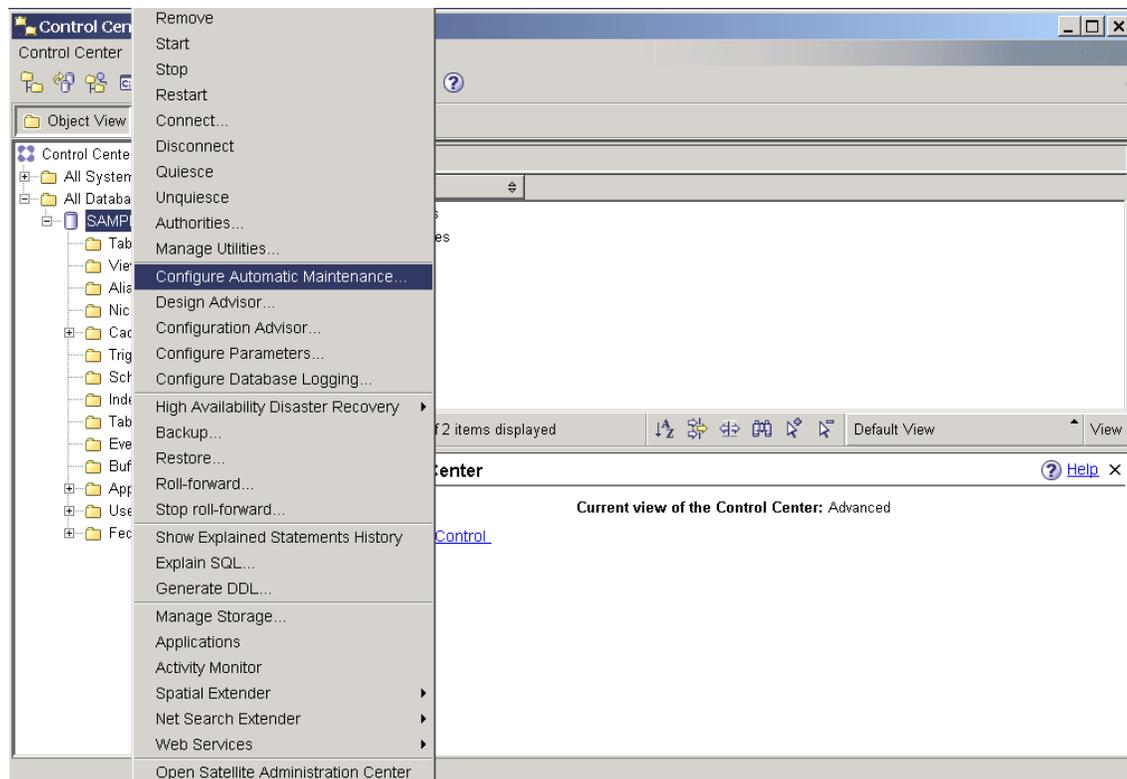


Figura 12.5 – Lanzando el Asistente de Configuración de Mantenimiento automatizado

La figura 12.6 muestra el Asistente de Configuración de Mantenimiento automatizado.

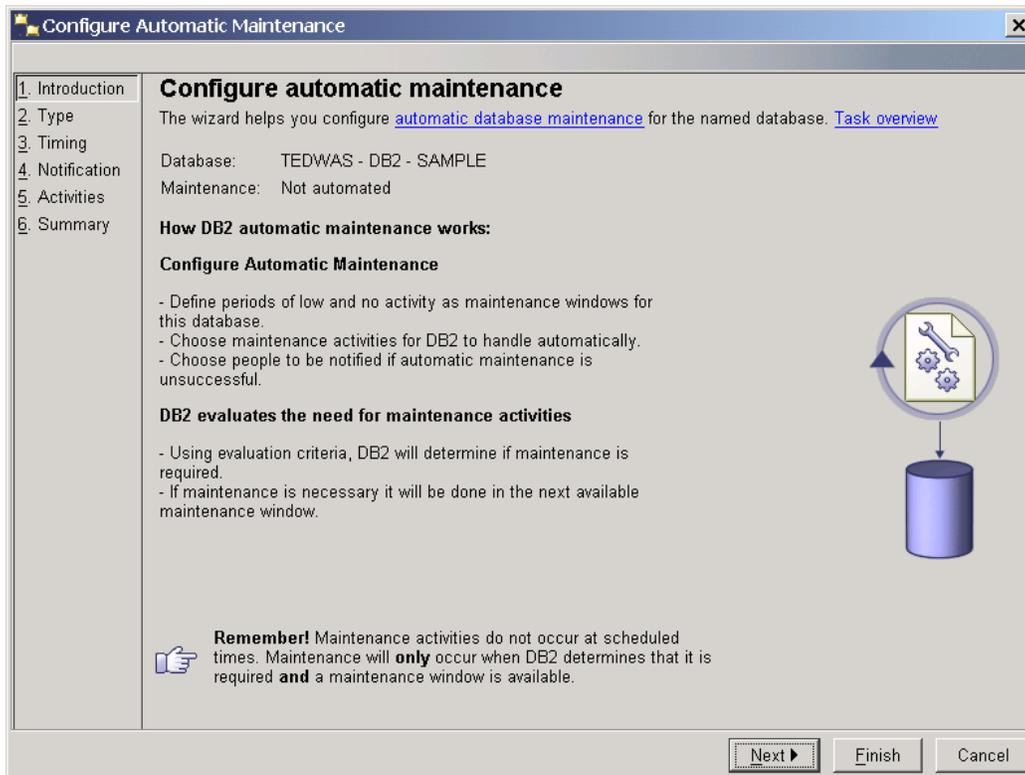


Figura 12.6 – El Asistente de Configuración de Mantenimiento Automatizado

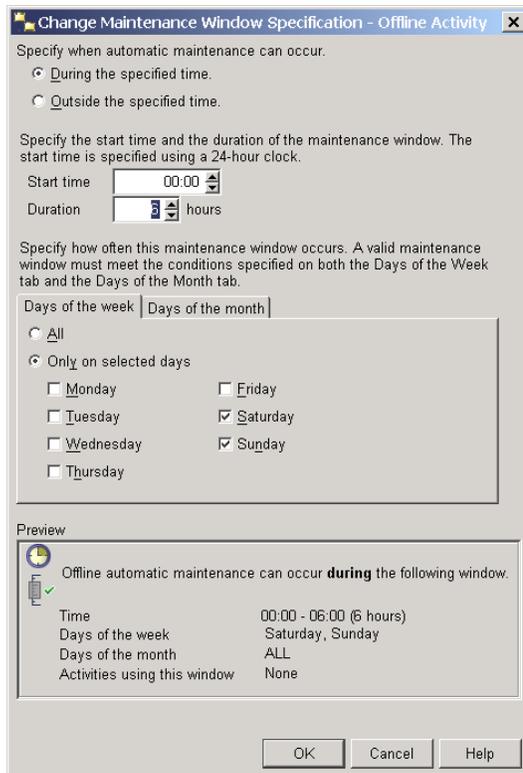
Quicklab #11 – Configuración del mantenimiento automatizado

Objetivo

En este Quicklab, con unos pocos y simples pasos, se configurará el mantenimiento automático en la base de datos SAMPLE de DB2

Procedimiento

1. Desde el árbol de objetos del Centro de Control, hacer clic-derecho en la base de datos *SAMPLE* y seleccione el ítem del menú Configuración de Mantenimiento Automático. Esto dispara el Asistente de Configuración de Mantenimiento Automático.
2. La página de *Introducción* del Asistente muestra la configuración de Mantenimiento automático actual. Si ha creado la base de datos *SAMPLE* con la opción de Mantenimiento Automático, entonces este modo de mantenimiento ya está configurado. Puede usar este asistente para re-configurar las opciones de mantenimiento automatizado. Haga clic en el botón *Siguiente* para moverse a la siguiente página del Asistente.
3. La página *Tipo* del Asistente requiere que elija entre desactivar todo el mantenimiento automático, o cambiar las configuraciones de mantenimiento automático. Seleccione la opción para cambiar las configuraciones de mantenimiento automático, haga clic en *Siguiente*.
4. La página de frecuencia (*Timing*) del asistente requiere que se especifique las ventanas de mantenimiento. Configure la ventana de apagado (*Offline window*) para que cada Sábado y Domingo a la noche desde las 0:00 hasta las 6 AM como se muestra abajo. Haga clic en el botón de Cambio, junto al panel de visualización previa de ventana de mantenimiento Fuera de Línea y elija las horas desde-hasta deseadas. Al finalizar de ingresar la información haga clic en el botón *OK* para volver al Asistente. Deje la ventana online como está (así el mantenimiento en línea puede realizarse en cualquier momento). Haga clic en el botón *Next*.



5. En la página de *Notificación* del Asistente, se puede fijar un contacto en caso de que una actividad de mantenimiento automatizado falle. Salteemos este paso por ahora. Haga clic en el botón *Siguiente*

6. En la pagina de *Actividades* del Asistente, puede elegir individualmente el automatizar o no, algunas de las tareas específica tanto como ser notificado de actividades particulares. En este ejemplo, asegúrese que todos los checkboxes de *Automatice* están chequeados y que los checkboxes *Notificar* estén des-chequeados. Haga clic en el botón *Siguiente*.

7. Antes de pasar a la próxima pagina del Asistente, debería configurar la ubicación del respaldo de la base de datos. Idealmente se debería almacenar los respaldos en un dispositivo diferente en caso de falla de disco. En la pagina *Actividades*, seleccione la opción *Backup database option*, y luego haga clic en el botón *Configuración de Seteos*.

8. En la oreja de *Criterio de Backup* de la ventana de dialogo de Configuración de Seteos, elija la opción *Balance de Recuperabilidad de Base de datos con Performance*. En la oreja de *Ubicación de Backup*, elija una ubicación preexistente y haga clic en el botón *Cambiar*. Especifique una diferente ubicación para realizar un respaldo backup (asegure que haya suficiente espacio en el disco). En la oreja *Modo del Backup*, asegúrese que *Backup fuera de línea* esté seleccionado. Haga clic en el botón *OK* para cerrar este tab. Haga clic en el botón *Siguiente*.

9. La página de *Sumario del Asistente de Configuración de Mantenimiento Automatizado* contiene un resumen de las opciones que ha elegido en el proceso. Haga clic en el botón de *Finalizar* para aceptar e implementar el cambio.

13

Capítulo 13 – Concurrencia y Bloqueo

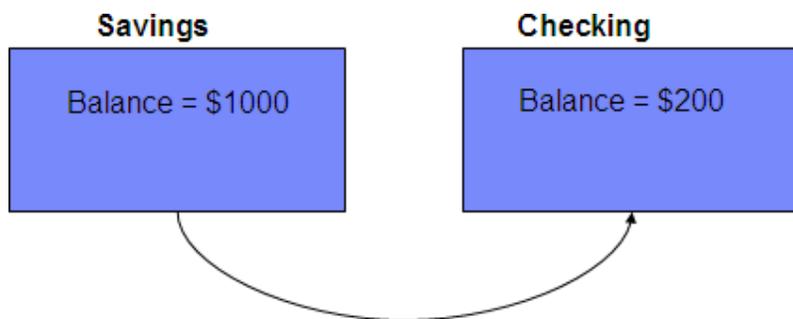
Este capítulo discute cómo permitir que múltiples usuarios tengan acceso a la misma base de datos al mismo tiempo sin interferir uno a otro, y mantener consistencia en sus operaciones. Discutiremos los conceptos de transacciones, de la concurrencia y del bloqueo.

Nota:

Para más información acerca de concurrencia y bloqueo, vea este vídeo:
<http://www.channeldb2.com/video/video/show?id=807741:Video:4322>

13.1 Transacciones

Una transacción o una unidad del trabajo consiste en una o más sentencias SQL que, cuando se ejecutan, se deben considerar como una sola unidad; es decir, si una de las sentencias en la transacción falla, la transacción entera falla, y cualquier sentencia ejecutada hasta el punto de la falla se restaura a un estado previo íntegro o consistente (se hace un *rollback*). Una transacción termina con una sentencia COMMIT, que también significa el comienzo de una nueva transacción. La figura 13.1 proporciona un ejemplo de una transacción.



Transfer \$100 from Savings to Checking:

- Debit \$100 from Savings account
- Credit \$100 to Checking account

Figura 13.1 – Un ejemplo de transacción

En la figura 13.1, por ejemplo, usted quiere transferir 100 dólares de su cuenta de ahorros a su cuenta de cheques. La siguiente secuencia de eventos puede ser requerida para completar esa tarea:

Realizar un débito de \$100 de la cuenta de ahorros y luego acreditar este monto en la cuenta de cheques.

Si la secuencia de eventos no se trata como una sola unidad de trabajo, una transacción, se imagina qué sucedería si ocurriera un apagón después de la operación de débito de la cuenta de ahorros, pero antes de que se acredite la operación en la cuenta de cheques. ¡Usted perdería \$100!

13.2 Concurrencia

Concurrencia implica que varios usuarios puedan trabajar al mismo tiempo sobre los mismos objetos de la base de datos. DB2 fue diseñada como una base de datos multiusuarios. El acceso a los datos debe ser coordinado correctamente y transparentemente usando un mecanismo para asegurar integridad y consistencia de los datos. Considere la figura 13.2 como un ejemplo.

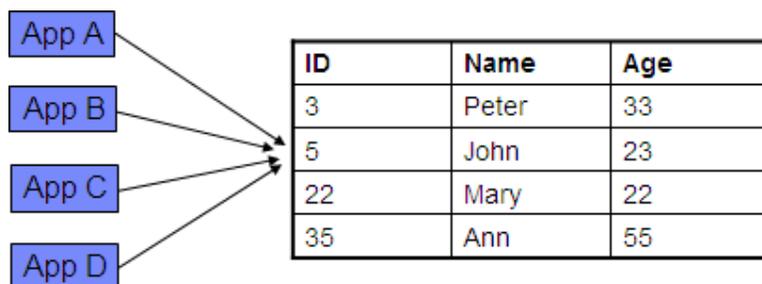


Figura 13.2 –Un ejemplo de concurrencia, y la necesidad por control de concurrencia

En la figura 13.2, hay cuatro aplicaciones, App A, App B, App C, y App D que están intentando tener acceso a la misma fila (fila 2) en una tabla. Sin ningún control de concurrencia, todas las aplicaciones podrían realizar operaciones en la misma fila. Asumiendo que todas las aplicaciones están actualizando la columna *Age* para la fila 2 con diversos valores, la aplicación que realiza la última actualización será probablemente el "ganador" en esta situación. Debe ser obvio en este ejemplo que una cierta clase de control de concurrencia es requerida para garantizar resultados consistentes. Este control de concurrencia se basa en el uso de bloqueos.

Los conceptos de bloqueo y concurrencia van de la mano. El bloqueo temporal impide a otras aplicaciones realizar sus operaciones hasta que otra operación termine. A mayor bloqueo que haya en un sistema, menor concurrencia será posible. Del otro lado, a menor bloqueo que haya en un sistema, mayor concurrencia será posible.

Los bloqueos ocurren automáticamente pues son necesarios para apoyar una transacción y se eliminan cuando la transacción termina (con un comando del COMMIT o del ROLLBACK). Los bloqueos se pueden hacer en las tablas o las filas. Hay dos tipos básicos de bloqueos:

- Bloqueos Compartidos (S locks) – Se usa cuando una aplicación quiere leer y prevenir que otros actualicen la misma fila
- Bloqueos (X locks) – Se usan cuando una aplicación actualiza, inserta, o borra una fila

Ahora considere la figura 13.3, la cual es similar a la figura 13.2, pero esta muestra un bloqueo.

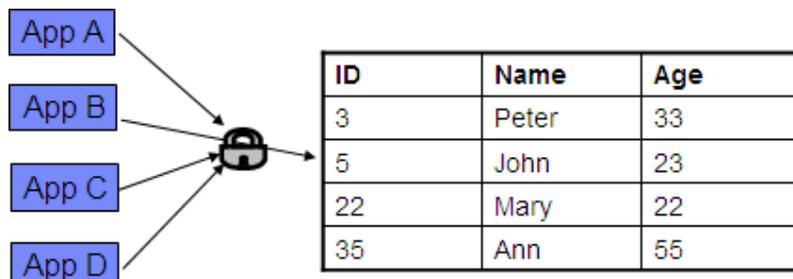


Figura 13.3 –Un ejemplo de concurrencia, y la necesidad de bloqueos

Por ejemplo, en la figura 13.3, si la App B es la primera aplicación en acceder a la fila 2, y está realizando un UPDATE, App B mantiene un bloqueo de X en la fila. Cuando la App A, la App C y la App D intentan tener acceso a la misma fila, no podrán hacer un UPDATE debido al bloqueo de X. Este control permite consistencia y la integridad de los datos.

13.3 Problemas sin control de concurrencia

Sin alguna forma de control de concurrencia, se pueden presentar los siguientes problemas

- Actualizaciones perdidas
- Lecturas no confirmadas o Uncommitted read
- Lecturas no repetidas
- Lecturas fantasmas

13.3.1 Actualizaciones perdidas

Las actualizaciones perdidas son un problema similar al que se explicó previamente en esta sección donde las aplicaciones realizaron actualizaciones y la última en hacerlo sería la “ganadora”.

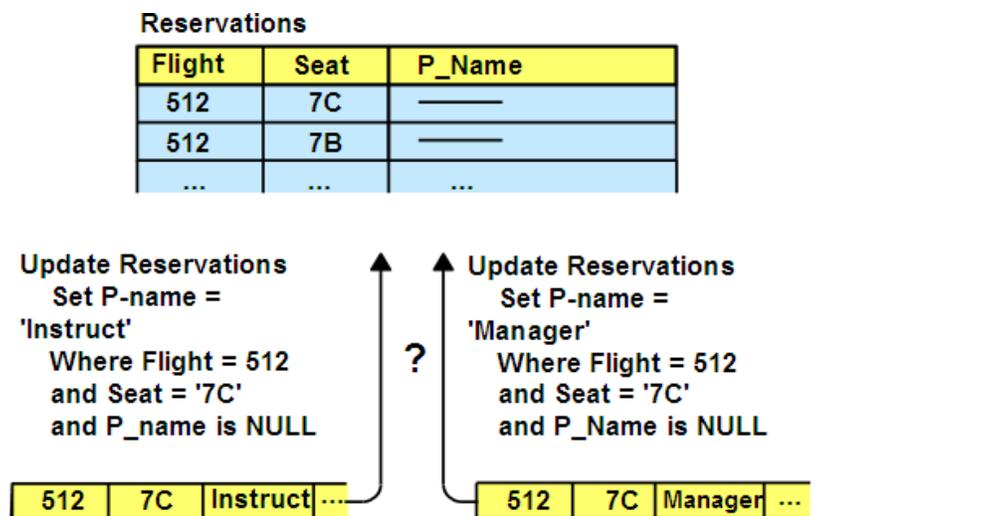


Figura 13.4 – Actualización perdida

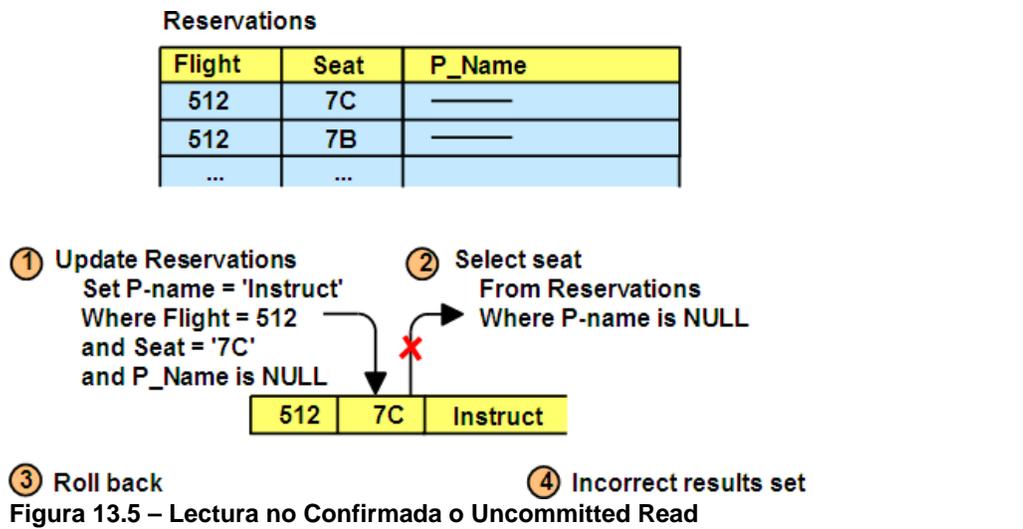
En la figura 13.4 hay dos aplicaciones intentando actualizar la misma fila. La aplicación App1 está del lado izquierdo, aplicación App2 del lado derecho. Entonces, la secuencia de eventos es:

1. App1 actualiza una fila
2. App2 actualiza la misma fila
3. App1 hace commit
4. App2 hace commit

La actualización de la aplicación App1 se pierde cuando la aplicación App2 hace su actualización, y de ahí el término “Actualización Perdida”.

13.3.2 Lecturas no confirmadas o Uncommitted read

Una lectura no confirmada, o también llamada una “lectura sucia”, permite a una aplicación leer información que no ha sido confirmada, y por lo tanto no es necesariamente correcta.



En la figura 13.5 se sigue esta secuencia de eventos:

- La aplicación App1 actualiza una fila
- La aplicación App2 lee los nuevos valores desde esa fila
- La aplicación App1 hacer roll back sobre el cambio de esa fila

La aplicación App2 está leyendo datos no confirmados, y por ello son datos inválidos, por eso este problema es llamado una “Lecturas no confirmadas”

13.3.3 Lecturas no repetidas

Una lectura no repetida implica que usted no puede obtener el mismo resultado después de realizar la misma lectura en la misma operación.

FLIGHT	SEAT	NAME	DESTINATION	ORIGIN
512	7B	_____	DENVER	DALLAS
....				
....				
814	8A	_____	SAN JOSE	DENVER
....				
134	1C	_____	HONOLULU	SAN JOSE
....			

Figura 13.6 – Lectura no repetida

En la figura 13.6, considere que usted esté intentando reservar un vuelo de Dallas a Honolulu. La secuencia de eventos es:

- La aplicación App1 abre un cursor (también conocido como un conjunto de resultados o result set) obteniendo lo que usted ve en la figura 13.6
- La aplicación App2 borra una fila que calificó para el cursor (por ejemplo, la fila con destino "San Jose")
- La aplicación App2 confirma los cambios
- La aplicación App1 cierra y vuelve a abrir el cursor

En este caso, ya que la aplicación App1 no consiguió los mismos datos en la lectura, no puede reproducir el conjunto de resultados; a este problema es al que se le llama "lectura no repetida".

13.3.4 Lectura fantasma

La lectura fantasma es un problema similar al problema de lectura no repetida, pero la diferencia está en que las subsecuentes tomas o "fetchs", usted puede obtener filas adicionales. La figura 13.7 proporciona un ejemplo de este problema.

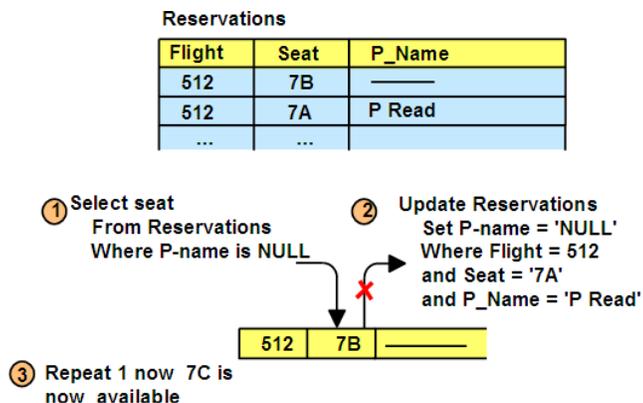


Figura 13.7 – Lectura fantasma

La figura 13.7 muestra la siguiente secuencia de eventos:

1. La aplicación App1 abre un cursor
2. La aplicación App2 agrega una fila a la base de datos que podría calificar para el cursor
3. La aplicación App2 commits los cambios
4. La aplicación App1 cierra y vuelve a abrir el cursor

En este caso, la aplicación App1 podría no conseguir los mismos datos en una lectura repetida, esta podría traer más filas, por esto este problema es llamado "lectura fantasma".

13.4 Niveles de aislamiento

Usted puede considerar los niveles de aislamiento como políticas de bloqueo donde, dependiendo del nivel de aislamiento elegido, usted puede conseguir diferentes comportamientos para una base de datos bloqueada con una aplicación.

DB2 provee diferentes niveles de protección para aislamientos de datos:

- Lectura no confirmada o Uncommitted Read (UR)
- Estabilidad de cursor o Cursor Stability (CS)
- Estabilidad de Lectura o Read Stability (RS)
- Lecturas repetidas o Repeatable Read (RR)

13.4.1 Lectura no confirmada o Uncommitted read

La lectura no confirmada también es conocida como una lectura “sucia”. Este es el menor nivel de aislamiento, y provee el mayor nivel de concurrencia. En operaciones de lectura, las filas no son bloqueadas, a menos que otra aplicación intente eliminar o modificar la tabla; y en operaciones de actualización, el comportamiento es como si se usara el nivel de aislamiento de Estabilidad de Cursor o cursor stability.

Los problemas que aún persisten con este nivel de aislamiento son:

- Lectura no confirmada
- Lecturas no repetidas
- Lecturas fantasmas

Problemas prevenidos con este nivel de aislamiento son:

- Pérdidas de actualizaciones

13.4.2 Estabilidad de cursor o Cursor stability

Estabilidad de cursor o Cursor stability es el nivel de aislamiento predeterminado. Esto provee el mínimo nivel de bloqueo. Básicamente, con este nivel de aislamiento la fila "actual" de un cursor está bloqueada. Si la fila es sólo leída, el bloqueo es mantenido hasta que una nueva fila es tomada (fetched) o la unidad de trabajo es terminada. Si la fila es actualizada, el bloqueo es mantenido hasta que la unidad de trabajo es terminada.

Los problemas que persisten con este nivel de aislamiento son:

- Lecturas no repetidas
- Lecturas fantasmas

Los problemas prevenidos con este nivel de aislamiento son:

- Pérdida de actualizaciones
- Lectura no confirmada

13.4.3 Estabilidad de lectura o Read stability

Con estabilidad de lectura, todas las filas que una aplicación devuelve dentro de una unidad de trabajo son bloqueadas. Para un cursor dado, el bloqueo se da sobre todas las filas que califican para el cursor. Por ejemplo, si tu tienes una tabla con 10,000 filas y la

consulta regresa 10 filas, entonces sólo esas 10 filas son bloqueadas. Estabilidad de Lectura usa un moderado grado de bloqueo.

Los problemas persistentes con este nivel de aislamiento son:

- Lectura fantasma

Los problemas prevenidos con este nivel de aislamiento son:

- Pérdida de actualizaciones
- Lectura no confirmada
- Lecturas no repetidas

13.4.4 Lecturas repetidas

Lecturas repetidas es el más alto nivel de aislamiento. Proporciona el grado más alto de bloqueo, y la menor concurrencia. Los bloqueos se mantienen en todas las filas procesadas para construir el result set; es decir, las filas que no necesariamente están en el conjunto de resultados final pueden ser bloqueadas. Ninguna otra aplicación puede actualizar, borrar, o insertar una fila que afectaría el resultado bloqueado hasta que la unidad de trabajo termina. Las lecturas repetidas garantizan que la misma consulta hecha por una aplicación, en una unidad del trabajo, dará el mismo resultado cada vez.

Problemas que persisten con este nivel de aislamiento son:

- ninguno

Problemas prevenidos con este nivel de aislamiento son:

- Pérdida de actualizaciones
- Lecturas no confirmadas
- Lecturas no repetidas
- Lecturas fantasmas

13.4.5 Comparando niveles de aislamiento

La figura 13.8 compara los diferentes niveles de aislamiento para un fetch. En la figura, nosotros vemos que el nivel de aislamiento lectura no confirmada o uncommitted read (UR) no tiene bloqueos. El nivel de aislamiento estabilidad de cursor o cursor stability (CS) tiene un bloqueo por la fila 1 cuando ésta está siendo tomada por el Fetch, pero la libera tan pronto como este proceso toma o hace fetch de la fila 2, y así continúa. Para el nivel de aislamiento Estabilidad de lectura o read stability (RS) o lecturas repetidas (RR), cualquier fila que es tomada con fetch será bloqueada, y el bloqueo no es liberado hasta el final de la transacción (cuando se ejecuta un commit).

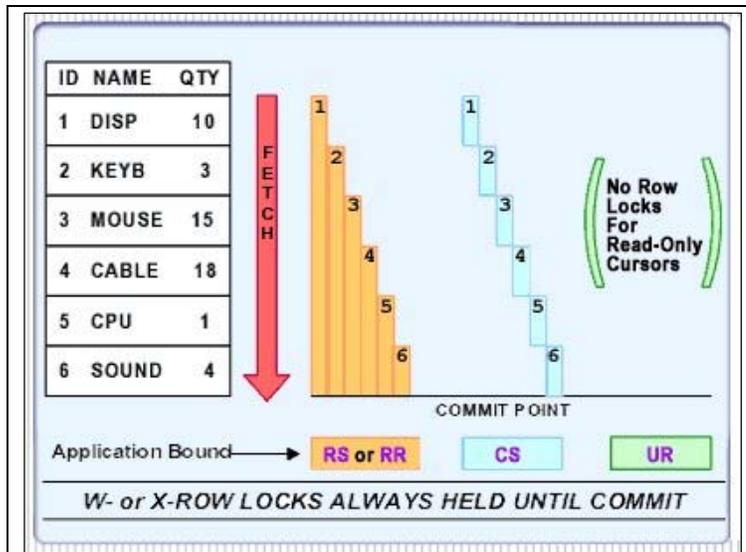


Figura 13.8 – Comparando niveles de aislamiento por un fetch

13.4.6 Fijando los niveles de aislamiento

Los niveles de aislamiento pueden ser especificados en muchos niveles:

- Sesión (aplicación)
- Conexión
- Sentencia

El nivel de aislamiento es definido normalmente en la sesión o en el nivel de aplicación. Si no es especificado un nivel de aislamiento en su aplicación, es puesto por default en Estabilidad de cursor. Por ejemplo, la tabla 13.1 muestra los posibles niveles de aislamiento para un programa .NET o JDBC y cómo esas propiedades corresponden con un nivel de aislamiento de DB2.

DB2	.NET	JDBC
Uncommitted Read (UR)	ReadUncommitted	TRANSACTION_READ_UNCOMMITTED
Cursor Stability (CS)	ReadCommitted	TRANSACTION_READ_COMMITTED
Read Stability (RS)	RepeatableRead	TRANSACTION_REPEATABLE_READ
Repeatable Read (RR)	Serializable	TRANSACTION_SERIALIZABLE

Tabla 13.1 - Comparación de la terminología para los de niveles de aislamiento

A nivel sentencia, el aislamiento puede ser fijado usando la cláusula WITH {nivel de aislamiento}. Por ejemplo:

```
SELECT ... WITH {UR | CS | RS | RR}
```

Escenario de ejemplo:

Asuma que una aplicación necesita conseguir una cuenta aproximada del número de filas que hay en una tabla. La prioridad es tener el mejor performance. En este ejemplo, se puede utilizar el nivel de aislamiento de Estabilidad de cursor para toda la aplicación, pero para la sentencia usada para obtener el número de filas de la tabla se puede usar aislamiento UR como se muestra:

```
SELECT COUNT(*) FROM tab1 WITH UR
```

Para SQL embebido, el nivel es fijado cuando se relacionan las variables en la compilación (bind time). Para SQL dinámico, el nivel se fija en tiempo de ejecución.

La elección del nivel de aislamiento depende de la aplicación. Si su aplicación no necesita cuentas exactas como en el ejemplo previo, elija el aislamiento UR. Si su aplicación necesita un control preciso de los datos con los que trabaja, elija aislamiento RR.

13.5 Escalamiento de Bloqueos

Cada bloqueo hecho por DB2 consume algo de memoria. Cuando el optimizador cree que es mejor tener un bloqueo en toda la tabla, en vez de múltiples filas bloqueadas, ocurre el fenómeno llamado Escalamiento de bloqueos. La figura 13.9 ilustra esto.

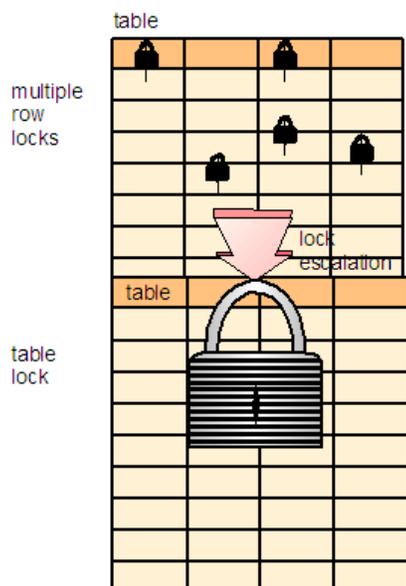


Figura 13.9 – Escalamiento de Bloqueos

Hay dos parámetros de configuración principales de una base de datos relacionadas al Escalamiento de lo bloqueos:

- LOCKLIST – La cantidad de memoria (en páginas de 4k) reservada para manejar bloqueos para todas las aplicaciones conectadas. El valor predeterminado es cincuenta páginas de 4K (200 K) en Windows.
- MAXLOCKS – El máximo porcentaje de la lista entera de bloqueos que una sola aplicación puede usar. El valor predeterminado es 22%.

Por lo tanto, si el valor predeterminado es usado, ocurre escalamiento de bloqueo cuando una aplicación requiere más de 44K de memoria bloqueada (200 K * 22% = 44K). Si un escalamiento de bloqueo ocurre frecuentemente con esas condiciones, incremente el valor de LOCKLIST y MAXLOCKS. El fenómeno escalamiento de bloqueo no es bueno para el rendimiento pues reduce concurrencia. El archivo de diagnóstico de DB2 (db2diag.log, que normalmente está localizado en el directorio C:\Program Files\IBM\SQLLIB\DB2) puede ser usado para determinar si está ocurriendo escalamiento de bloqueo.

13.6 Monitoreo de bloqueos

Usted puede monitorear el uso de bloqueos en DB2 usando la utilidad “snapshot” de DB2 que puede capturar instantáneamente lo que ocurre en su aplicación. Para habilitar la instantánea o snapshot para bloqueos, use este comando:

UPDATE MONITOR SWITCHES USING LOCK ON

Después de habilitar la opción, la información de monitoreo será recolectada. Para obtener un reporte de los bloqueos en un momento dado, ejecute este comando:

GET SNAPSHOT FOR LOCKS FOR APPLICATION AGENT ID <handle>

La figura 13.9 muestra la salida para un ejemplo de la aplicación instantánea de bloqueo.

Figura 13.9 – Aplicación instantánea de bloqueo

```

Application Lock Snapshot

Snapshot timestamp                = 11-05-2002
00:09:08.672586

Application handle                 = 9
Application ID                    = *LOCAL.DB2.00B9C5050843
Sequence number                   = 0001
Application name                   = db2bp.exe
Authorization ID                   = ADMINISTRATOR
Application status                 = UOW Waiting
Status change time                = Not Collected
Application code page              = 1252
Locks held                        = 4
Total wait time (ms)              = 0

List Of Locks
Lock Name                        = 0x050007000480010000000000052
Lock Attributes                   = 0x00000000
Release Flags                     = 0x40000000
Lock Count                        = 255
Hold Count                        = 0
Lock Object Name                  = 98308
Object Type                       = Row
Tablespace Name                   = TEST4K
Table Schema                      = ADMINISTRATOR
Table Name                        = T2
Mode                              = X
    
```

13.7 Espera de Bloqueo

Cuando dos o más aplicaciones necesitan realizar una operación sobre el mismo objeto, uno de ellos puede tener que esperar para obtener el bloqueo necesario. De forma predeterminada, una aplicación esperará indefinidamente. El tiempo que una aplicación espera por un bloqueo está controlado por el parámetro de configuración de la base de datos LOCKTIMEOUT. El valor predeterminado de este parámetro es -1 (espera infinita).

El registro CURRENT LOCK TIMEOUT puede ser usado para fijar el tiempo de Espera de bloqueo para una conexión dada. De forma predeterminada, este registro tiene el valor de LOCKTIMEOUT. Use la sentencia SET LOCK TIMEOUT para cambiar su valor. Una vez que el valor de este registro es fijado para una conexión, persistirá a través de las transacciones. Por ejemplo:

```
SET LOCK TIMEOUT=WAIT n
```

13.8 Bloqueo Mortal o Deadlock, causas y detección

Un bloqueo mortal o deadlock ocurre cuando dos o más aplicaciones conectadas a la misma base de datos esperan indefinidamente por un recurso. La espera nunca es resuelta porque cada aplicación está esperando un recurso que la otra necesita. En la mayoría de los casos los Deadlocks son un problema de diseño de la aplicación. La figura 13.10 ilustra un deadlock.

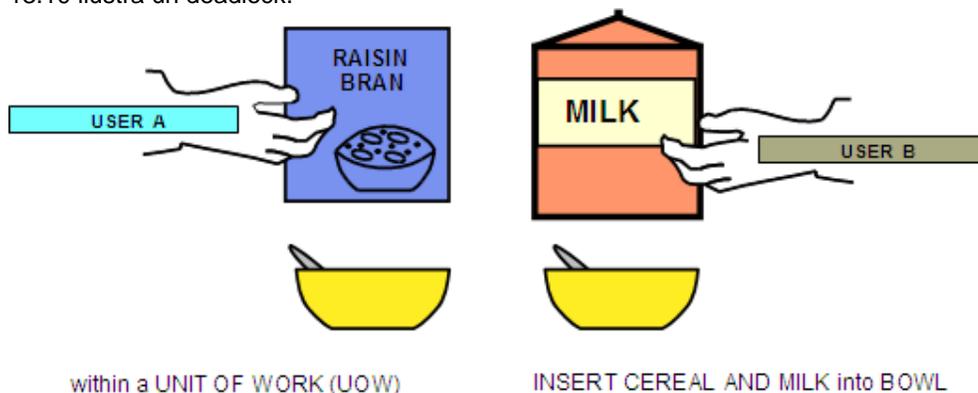


Figura 13.10 – Escenario de un Deadlock

En la figura 13.10, el usuario A está sosteniendo el cereal *raisin bran* y no lo soltará hasta que él tenga la leche. En la otra mano, el usuario B está sosteniendo la leche, y no la dejará hasta que tenga el cereal *raisin bran*. Por lo tanto, tenemos una situación de bloqueo mortal o deadlock.

Para simular una situación de deadlock en DB2, siga estos pasos:

1. Abra dos ventanas de comandos de DB2 (las cuales serán llamadas "CLP1" y "CLP2", respectivamente) representando dos diferentes aplicaciones que se conectan a la base de datos.

2. Desde CLP1 ejecute los comandos:

```
db2 connect to sample
db2 +c update employee set firstnme = 'Mary' where empno = '000050'
```

Primero nos conectamos a la base de datos SAMPLE, y luego ejecutamos una sentencia UPDATE en la fila con “empno = 50000” en la tabla employee. La opción “+c” en la sentencia indica que nosotros no queremos que la ventana de comandos de DB2 automáticamente ejecute un COMMIT luego de esta sentencia. Esto lo hacemos deliberadamente porque queremos mantener el bloqueo.

3. Desde CLP2 ejecute los comandos:

```
db2 connect to sample
db2 +c update employee set firstnme = 'Tom' where empno = '000030'
```

En la ventana CLP2, que representa la segunda aplicación, estamos conectando a la base de datos SAMPLE, pero estamos actualizando otra fila en la tabla employee.

4. Desde CLP1 ejecute:

```
db2 +c select firstnme from employee where empno = '000030'
```

Después de presionar *Enter* para ejecutar la sentencia SELECT, puede parecer que el SELECT se queda en un estado de no respuesta. Realmente no lo está, si no que está esperando por el liberamiento de un bloqueo exclusivo (o exclusive lock) que fue tomado por el CLP2 en esa fila en el paso 3. En este punto, si LOCKTIMEOUT ha sido dejado con su valor predeterminado de -1, la aplicación CLP1 podría esperar eternamente.

5. Desde CLP2 ejecute:

```
db2 +c select firstnme from employee where empno = '000050'
```

Al ejecutar la sentencia SELECT, estamos creando un deadlock. Esta sentencia SELECT también parecerá quedarse en un estado de no respuesta, porque está esperando por el liberamiento del bloque exclusivo (o exclusive lock) que fue tomado por CLP1 en esta fila en el paso 2.

En este ejemplo de deadlock, DB2 verifica el valor del parámetro de configuración DLCHKTIME en la base de datos. Este parámetro fija el intervalo de tiempo para verificar si un deadlock ocurrió. Por ejemplo, si este parámetro es fijado a 10 segundos, DB2 verificará cada 10 segundos si un deadlock ha ocurrido. Si un deadlock ha ocurrido, DB2 usará un algoritmo interno para determinar cuál de las dos transacciones podría ser rolled back (deshecha), y cuál debería continuar.

Si usted está experimentando numerosos deadlocks, usted debería examinar sus transacciones existentes y ver si una reestructuración es posible.

13.9 Mejores prácticas para concurrencia y bloqueo

Siguiendo los consejos siguientes, usted puede mejorar la concurrencia de sus aplicaciones:

1. Crear las transacciones lo mas cortas como sea posible. Esto puede ser posible utilizando constantemente la sentencia COMMIT (aun para transacciones de solo lectura) cuando la lógica de su aplicación lo permita.

2. Registrar (log) la información de las transacciones sólo cuando sea requerida.

3. Purgar los datos rápidamente usando:

```
ALTER TABLE ACTIVATE NOT LOGGED INITIALLY WITH EMPTY TABLE
```

4. Realizar modificaciones de datos en lotes o grupos. Por ejemplo:

```
DELETE FROM (  
    SELECT * FROM tedwas.t1 WHERE c1 = ... FETCH FIRST 3000 ROWS  
    ONLY)
```

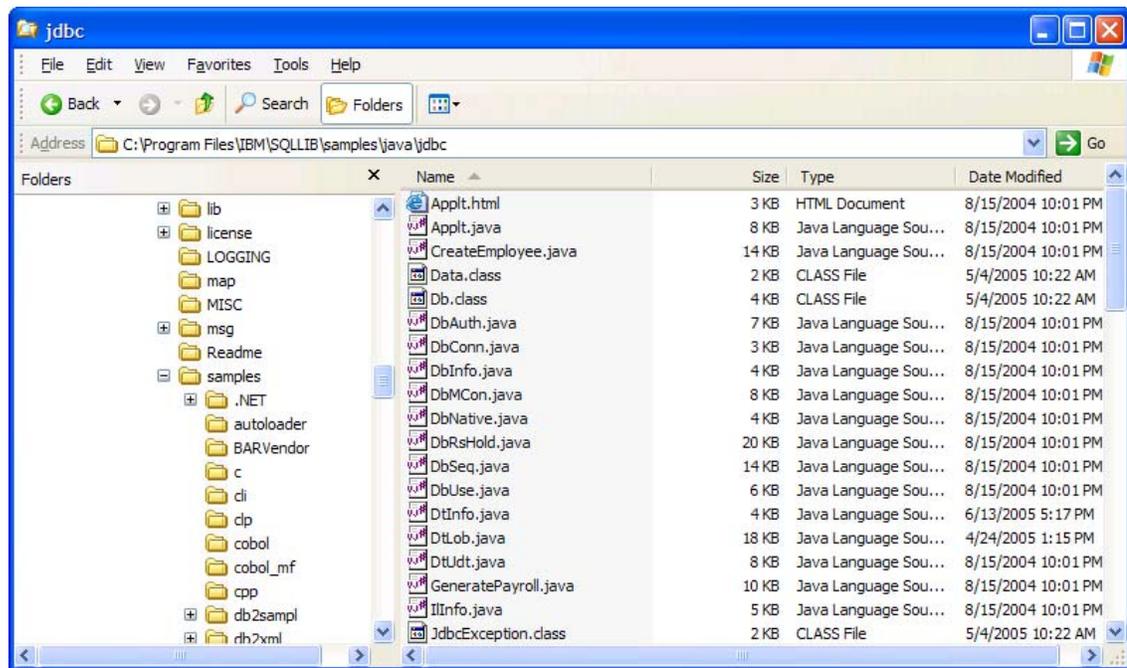
5. Use las características de concurrencia en las herramientas de movimiento de datos de DB2.

6. Fije el nivel del parámetro LOCKTIMEOUT (el tiempo sugerido está entre 30-120 segundos). No lo deje en su valor predeterminado de -1. Usted también puede usar tiempo de bloqueo basado en una sesión.

7. No regrese más datos de los que son requeridos. Por ejemplo, use la cláusula FETCH FIRST n ROWS ONLY en sentencias SELECT.

PARTE III – APRENDIENDO DB2: DESARROLLO DE APLICACIONES

En parte III de este libro, discutimos en detalle objetos de aplicación de la base de datos tales como: procedimientos de almacenado, funciones definidas por el usuario (con siglas en inglés (UDFs), y disparadores. Usted puede aprender como programar en distintos lenguajes de programación utilizando a DB2 como el servidor de datos si inspecciona el código de ejemplo que se incluye con la instalación de DB2 y se localizan por default en el directorio SQLLIB\samples. La figura siguiente muestra los programas de ejemplo con Java y DB2 en Windows.



Programas Java de ejemplo que se incluye con la instalación de DB2

14

Capítulo 14 – Procedimientos Almacenados SQL PL

En este capítulo vamos hablar sobre procedimientos almacenados. Un procedimiento almacenado es un objeto de aplicación de una base de datos que puede encapsular declaraciones SQL y lógica de negocio. Manteniendo parte de la lógica de la aplicación en la base de datos provee mejoras de rendimiento pues se reduce la cantidad de tráfico en la red entre la aplicación y la base de datos considerablemente. Mas aún, los procedimientos almacenados proveen una localización centralizada para almacenar su código, de esta manera, otras aplicaciones pueden reutilizar los mismos procedimientos.

Los procedimientos almacenados de DB2 pueden ser escritos usando SQL PL, C/C++, Java, Cobol, CLR (Common Language Runtime o Runtime de Lenguaje Común), y OLE. En este capítulo, nos enfocaremos en procedimientos SQL PL porque son los más populares y simples. La Figura 14.1 muestra cómo trabajan los procedimientos almacenados.

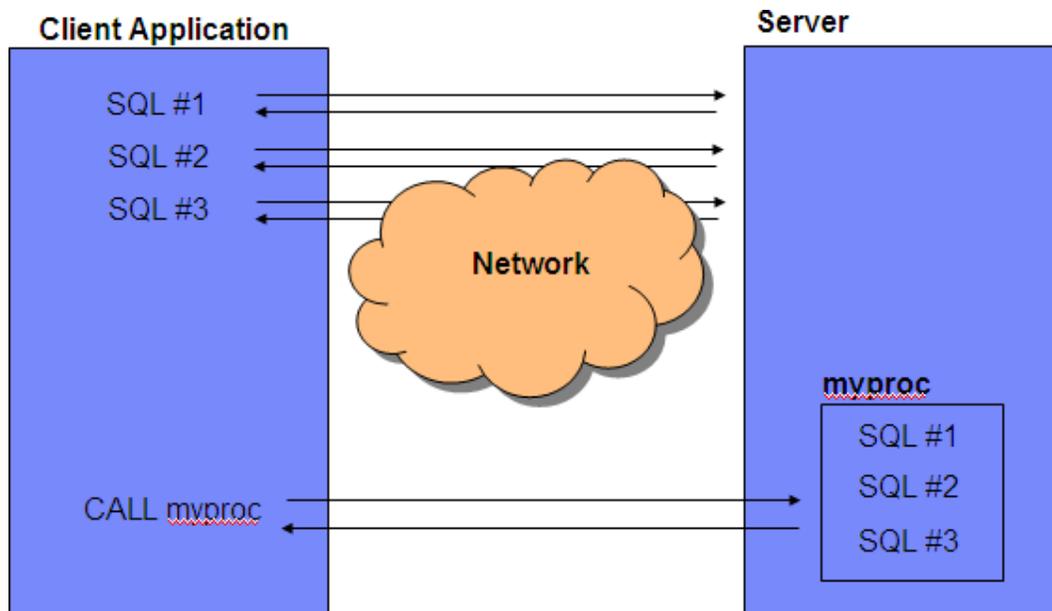


Figura 14.1 – Tráfico de red reducido con procedimientos almacenados.

En la esquina izquierda superior de la figura, se observan varias sentencias ejecutadas una después de otra. Cada sentencia SQL es enviada desde el cliente al servidor, y el servidor seguidamente devuelve el resultado al cliente. Muchas sentencias SQL son ejecutadas de este modo, entonces el tráfico de red se incrementa. Por el otro lado, en el vértice derecho inferior, se puede observar el procedimiento almacenado "myproc" mantenido en el servidor de datos, el cual contiene algunas sentencias SQL; y luego en el cliente (en el lado izquierdo), la declaración CALL es usada para llamar al procedimiento. Este segundo método de llamar el procedimiento es más eficiente, ya que hay una sola declaración de llamada que viaja a través de la red, y solo un resultado retorna al cliente.

Los procedimientos almacenados también pueden ser provechosos para propósitos de seguridad en su base de datos. Por ejemplo, usted puede dejar acceder a los usuarios a tablas o a vistas solamente a través de procedimientos almacenados; esto ayuda a preservar el acceso al servidor e impedir a usuarios tener acceso a la información a la que no tienen permiso de acceso. Esto es posible porque los usuarios no requieren privilegios explícitos sobre las tablas o vistas, para tener acceso a través de procedimientos almacenados, así los usuarios necesitan solamente los privilegios necesarios para invocar los procedimientos almacenados.

Nota:

Para mas información sobre procedimientos almacenados de SQL PL, mira este video: <http://www.channeldb2.com/video/video/show?id=807741:Video:4343>

14.1 IBM Data Studio

IBM Data Studio es una solución comprensiva que le ayuda a desarrollar y manejar las aplicaciones de base de datos durante todo el ciclo de vida de la gestión de datos. Algunas actividades que pueden ser realizadas con el IBM Data Studio son:

- Crear, modificar, y eliminar objetos de base de datos DB2 (con el análisis de impacto).
- Explorar y editar datos relacionales y XML.
- Construir visualmente declaraciones SQL y XQuery.
- Optimizar consultas utilizando Visual Explain.
- Desarrollar, eliminar fallos, y desplegar procedimientos almacenados SQL y Java™.
- Desarrollar funciones definidas por el usuario (UDFs).
- Desarrollar aplicaciones SQLJ.
- Desarrollar consultas y rutinas para aplicaciones pureXML.
- Realizar tareas de movimiento de datos.
- Colaborar en proyectos como miembro de un grupo.
- Rápidamente construir Web Services SOAP y REST
- Descubrir relaciones de objeto de base de datos con modelos de datos físicos (diagrama).
- Visualizar la distribución de datos a través de tablas.

IBM Data Studio está basado en la plataforma Eclipse. Es una imagen separada (no forma parte de la imagen de instalación de DB2), y también es gratuito. La imagen de Data Studio

puede ser descargada desde el botón “Download” en sitio web ibm.com/db2/express. La figura 14.2 muestra el IBM Data Studio.

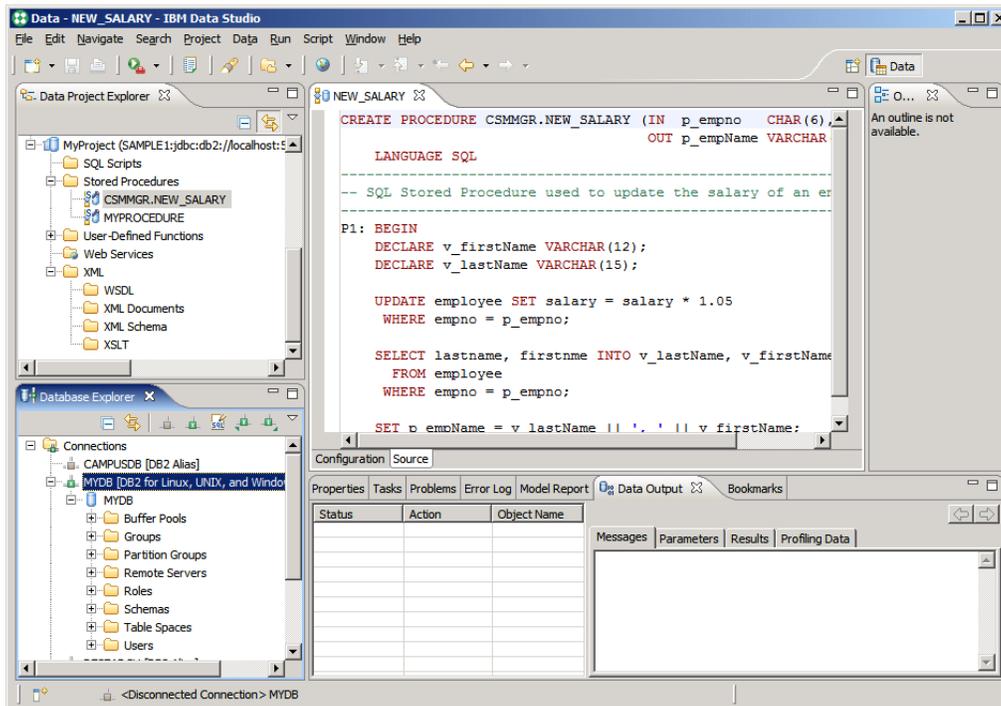


Figure 14.2 – El IBM Data Studio

14.1.2 Creación de procedimiento almacenado en IBM Data Studio

Para crear procedimientos almacenados Java o SQL PL en el IBM Data Studio, tiene que seguir los pasos siguientes. Note que procedimientos almacenados escritos en otros lenguajes no pueden ser creados usando el IBM Data Studio.

Paso 1: Cree un proyecto en el IBM Data Studio

Desde el menú de IBM Data Studio, escoja Archivo -> Nuevo -> Proyecto y escoja Proyecto de Desarrollo de Datos. Esto se muestra en la Figura 14.3.

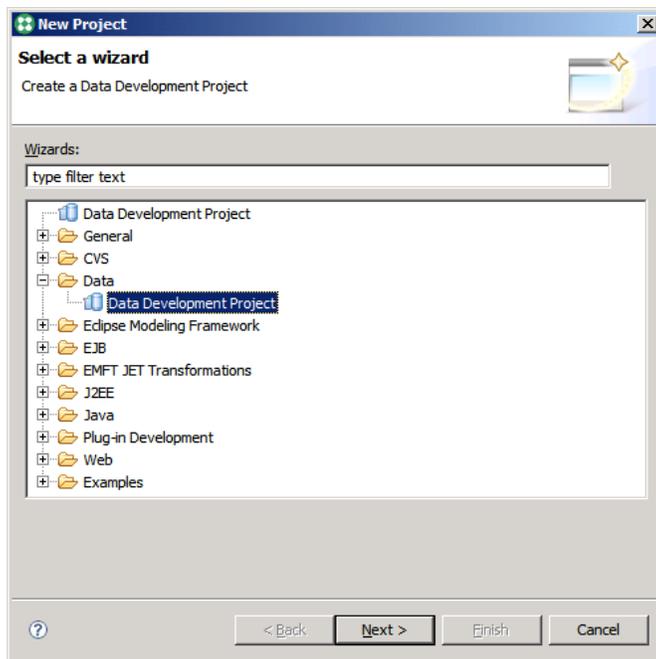


Figure 14.3 – El proyecto de desarrollo de datos

Siga los pasos del asistente para introducir un nombre para su proyecto, indicar a cual base de datos usted quiere unirse como parte de su proyecto, y especifique cual es el directorio JDK (el proporcionado por defecto es por lo general el correcto).

Paso 2: Cree un procedimiento almacenado

Cuando el proyecto es creado, el lado izquierdo de la perspectiva de datos mostrará su proyecto. En la Figura 14.4 usted puede ver el proyecto de "myProject" creado y expandido.

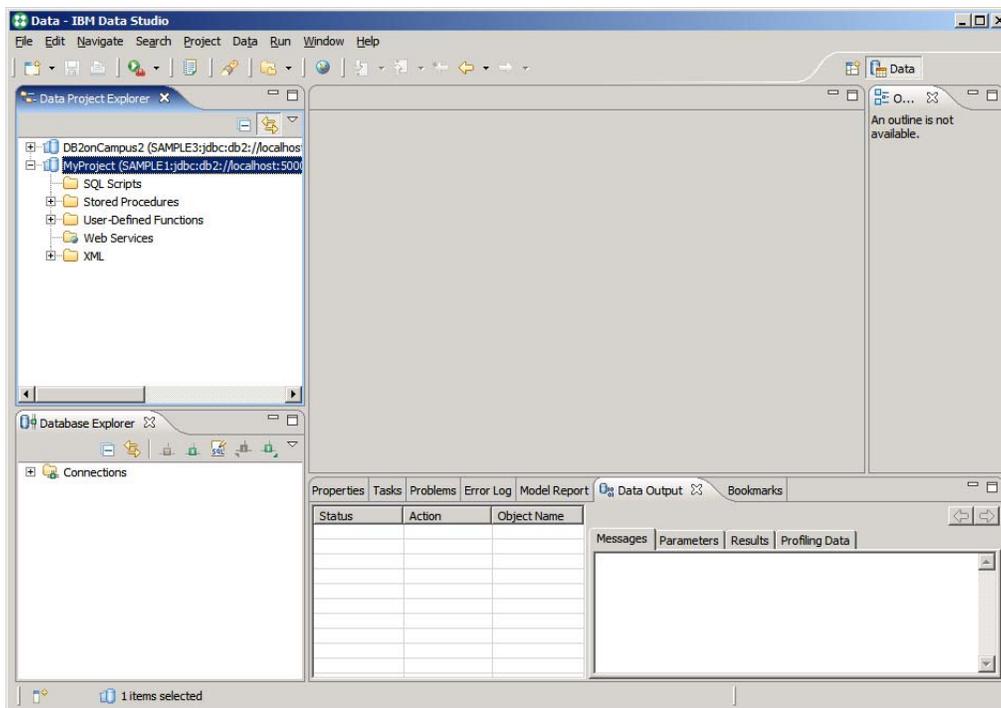


Figura 14.4 – El proyecto “myProject”.

La Figura 14.4 muestra las diferentes carpetas de su proyecto. Cuando usted desea crear un procedimiento almacenado, debe hacer clic con el botón derecho sobre la carpeta de Procedimientos Almacenada y escoge Nuevo -> Procedimiento Almacenado. Complete la información solicitada por el asistente para el "Nuevo Procedimiento Almacenado", como el nombre del proyecto a asociar el procedimiento, el nombre y lenguaje del procedimiento (Note que solamente SQL PL y Java son soportados por el IBM Data Studio de IBM), y las sentencias SQL que se usan en el procedimiento. Por defecto, el IBM Data Studio provee de ejemplos de sentencias SQL. En este punto, puede hacer clic en “Terminar” y el procedimiento almacenado es creado usando algún código de plantilla y las sentencias SQL proporcionada antes como un ejemplo. Esto se muestra en la Figura 14.5.

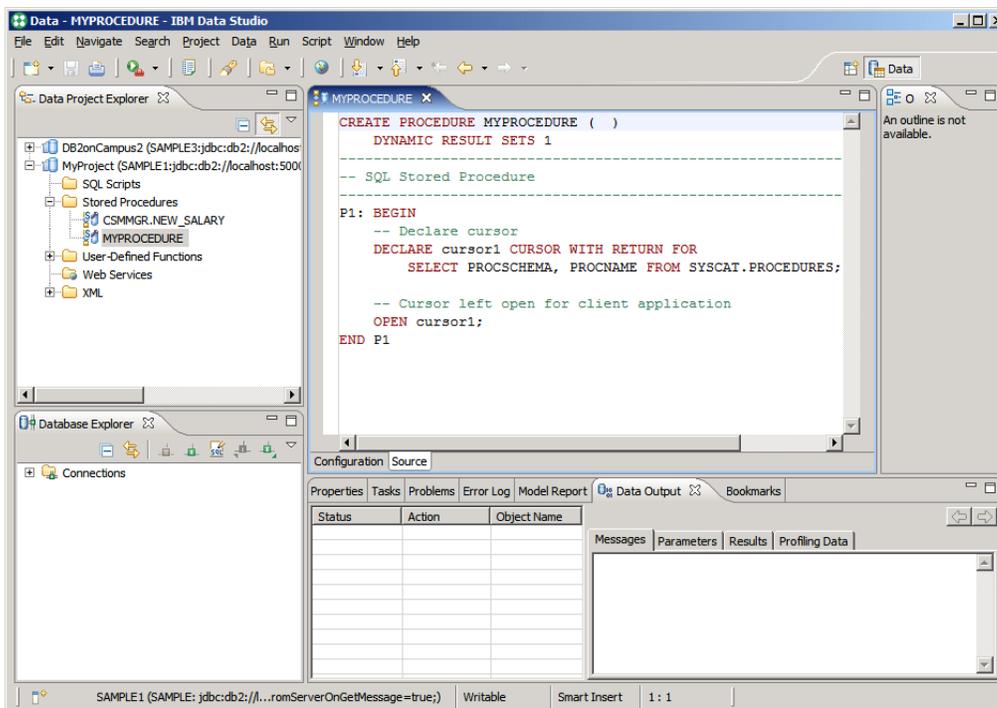


Figura 14.5 – Un ejemplo de procedimiento almacenado.

En la figura 14.5, el código del procedimiento almacenado "MYPROCEDURE" fue generado. Usted puede sustituir todo este código por su propio código. Para simplificar, vamos a continuar con este procedimiento almacenado de muestra como si se usted lo hubiera escrito.

Paso 3: Compilar e implementar un procedimiento almacenado

Una vez que el procedimiento almacenado se crea, se vuelve a compilar y desplegar, pulsando con el botón derecho sobre ella en el panel izquierdo y elegir "Desplegar". La Figura 14.6 ilustra este paso.

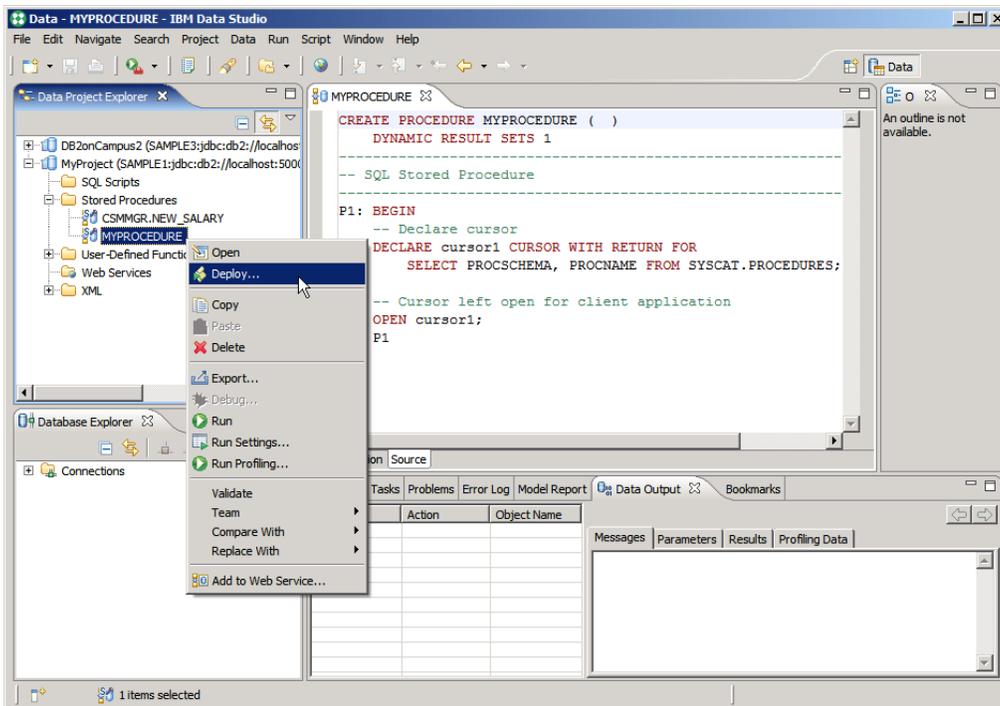


Figure 14.6 – Deploying a stored procedure

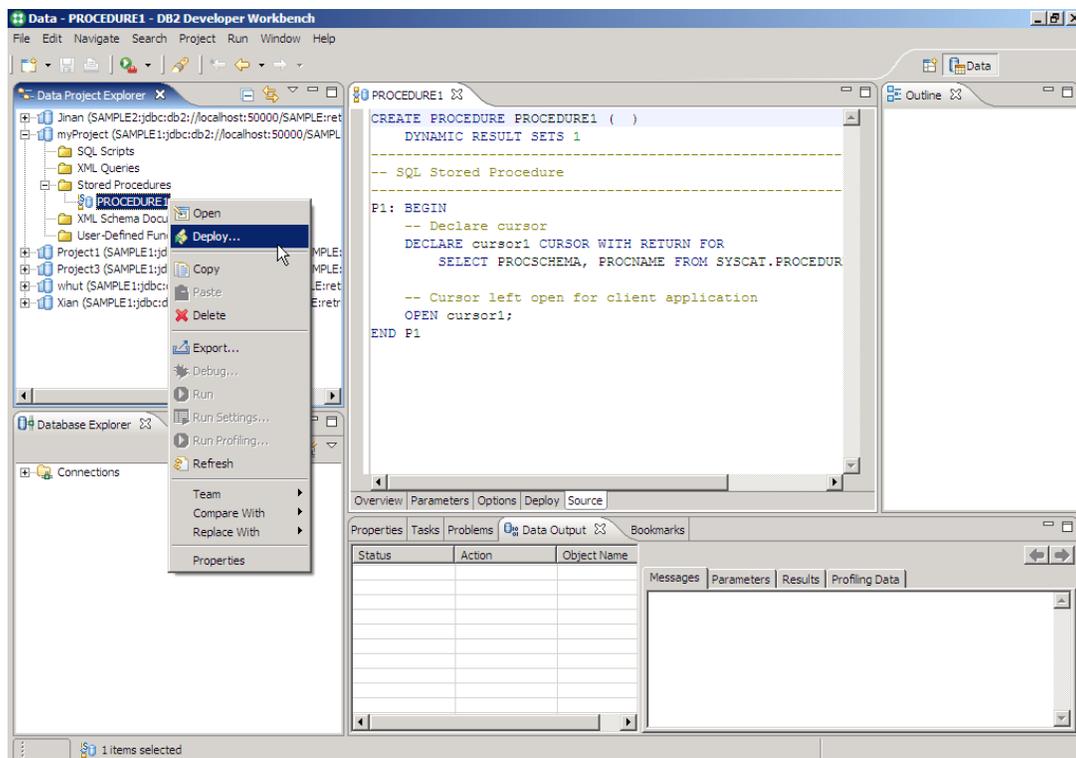


Figura 14.6 – Expandiendo un procedimiento almacenado.

Paso 4: Ejecutando un procedimiento almacenado.

Una vez que el procedimiento almacenado se ha desplegado, puede ejecutarlo haciendo clic derecho sobre él y seleccione "Ejecutar". Los resultados aparecen en la ficha "Resultados", en la esquina inferior derecha de la ventana. Para ejecutar un procedimiento almacenado desde la ventana de comandos o el comando del editor, puede utilizar `CALL <nombre del procedimiento>`.

14.2 Introducción a los procedimientos almacenados SQL PL

Los procedimientos almacenados SQL PL son fáciles de crear y aprender. Ellos tienen el mejor rendimiento en DB2. Procedimientos almacenados SQL PL (o simplemente "los procedimientos almacenados de SQL") son el foco de este capítulo.

14.2.1 Estructura de un procedimiento almacenado

La sintaxis básica de un procedimiento almacenado se muestra aquí.

```
CREATE PROCEDURE proc_name [( {parametros opcionales} )]
    [atributos opcionales del procedimiento] <statement>
```

Donde <statement> es una sentencia individual, o un conjunto de sentencias agrupadas por `BEGIN [ATOMIC]... END`

14.2.2 Atributos opcionales del procedimiento almacenado

A continuación se describen algunos de los atributos del procedimiento almacenado:

- `LANGUAGE SQL` (Lenguaje SQL)
Este atributo indica el idioma del procedimiento almacenado que se va a utilizar. `LANGUAGE SQL` es el valor por defecto. Para otros idiomas, tales como Java o C utilizar un `LANGUAGE JAVA` o `LANGUAGE C`, respectivamente.
- `RESULT SETS <n>` (Conjunto de Resultados <n>)
Esto es necesario si el procedimiento almacenado devuelve un conjunto n de resultados.
- `SPECIFIC my_unique_name` (Especificar mi_unico_nombre)
Este es un nombre único que se le puede dar a un procedimiento. Un procedimiento almacenado puede estar sobrecargado, es decir, varios procedimientos almacenados pueden tener el mismo nombre, pero con diferente número de parámetros. Mediante el uso de la palabra clave específica puede proporcionar un nombre único para cada uno de estos procedimientos almacenados, y esto puede facilitar la administración de procedimientos almacenados. Por ejemplo, para eliminar un procedimiento almacenado utilizando la palabra clave específica, puede utilizar esta declaración: `DROP SPECIFIC` (procedimiento específico). Si la palabra clave `SPECIFIC` (específica) no se hubiera utilizado, usted tendría que utilizar un procedimiento de declaración `DROP` y poner el nombre del procedimiento con todos los parámetros a fin de que DB2 sepa cual procedimiento almacenado sobrecargado quería eliminar.

14.2.3 Parámetros

Existen tres tipos de parámetros en un procedimiento almacenado SQL PL:

- `IN` – parámetro de entrada
- `OUT` – parámetro de salida
- `INOUT` – parámetro de entrada y salida

Por ejemplo:

```
CREATE PROCEDURE proc (IN p1 INT, OUT p2 INT, INOUT p3 INT)
```

Al llamar al procedimiento, todos los parámetros deben estar siempre en la sentencia de llamada `CALL`. Por ejemplo, para llamar al anterior procedimiento almacenado se debe especificar lo siguiente:

```
CALL proc (10,?,4)
```

El signo de interrogación (?) Se utiliza para los parámetros de salida en la declaración de llamada.

Aquí hay otro ejemplo de un procedimiento almacenado con los parámetros que usted puede probar:

14.2.6 Declaración de variables.

Para declarar una variable, se debe usar la sentencia DECLARE:

```
DECLARE <var_name> <data type> [DEFAULT value];
```

Aquí unos ejemplos:

```
DECLARE temp1 SMALLINT DEFAULT 0;
DECLARE temp2 INTEGER DEFAULT 10;
DECLARE temp3 DECIMAL(10,2) DEFAULT 100.10;
DECLARE temp4 REAL DEFAULT 10.1;
DECLARE temp5 DOUBLE DEFAULT 10000.1001;
DECLARE temp6 BIGINT DEFAULT 10000;
DECLARE temp7 CHAR(10) DEFAULT 'yes';
DECLARE temp8 VARCHAR(10) DEFAULT 'hello';
DECLARE temp9 DATE DEFAULT '1998-12-25';
DECLARE temp10 TIME DEFAULT '1:50 PM';
DECLARE temp11 TIMESTAMP DEFAULT '2001-01-05-12.00.00';
DECLARE temp12 CLOB(2G);
DECLARE temp13 BLOB(2G);
```

14.2.7 Sentencia de asignación de estados

Para asignar un valor a una variable, se debe usar la sentencia SET. Por ejemplo:

```
SET total = 100;
```

La declaración anterior es equivalente a:

```
VALUES (100) INTO total;
```

Adicionalmente, cualquier variable puede ser definida como NULL:

```
SET total = NULL;
```

En los casos siguientes, una excepción se lanza si se retorna más de una fila:

```
SET total = (select sum(c1) from T1);
SET first_val = (select c1 from T1 fetch first 1 row only)
```

También se puede asignar a variables valores basados en propiedades de una base de datos:

```
SET sch = CURRENT SCHEMA;
```

14.3 Cursores

Un cursor es un conjunto de resultados que es consecuencia de un comando SELECT. La sintaxis para declarar, abrir, buscar y cerrar un cursor es el siguiente:

```
DECLARE <nombre del cursor> CURSOR [WITH RETURN <return target>]
    <SELECT statement>;
OPEN <nombre del cursor>;
FETCH <nombre del cursor> INTO <variables>;
CLOSE <nombre del cursor>;
```

Cuando un cursor es declarado, la cláusula de retorno puede tener uno de estos valores:

- CLIENT: el conjunto de resultados volverá a la aplicación cliente.
- CALLER: el resultado se devuelve al cliente o procedimiento almacenado que hizo la llamada.

Aquí se tiene un ejemplo de un procedimiento almacenado utilizando un cursor:

```
CREATE PROCEDURE set()
DYNAMIC RESULT SETS 1
LANGUAGE SQL
BEGIN
DECLARE cur CURSOR WITH RETURN TO CLIENT
        FOR SELECT name, dept, job
           FROM staff
           WHERE salary > 20000;
OPEN cur;
END
```

14.4 Flujo de control

Al igual que en muchos otros lenguajes de programación, SQL PL tiene varios comandos que pueden utilizarse para controlar el flujo de la lógica. A continuación se lista algunas de las declaraciones de flujo de control soportados:

CASE (selecciona un camino de ejecución (búsqueda simple))
 IF
 FOR (ejecuta el cuerpo para cada fila de la tabla)
 WHILE
 ITERATE (potencia siguiente iteración. Analogía a continuar en C)
 LEAVE (hojas de un bloque o bucle. "organización estructurada Goto")
 LOOP (bucle infinito)
 REPEAT
 GOTO
 RETURN
 CALL (procedimiento de llamada)

14.5 Llamando a procedimientos almacenados

Los siguientes fragmentos de código muestran cómo llamar a procedimientos almacenados utilizando diferentes lenguajes de programación.

Ejemplo de llamada a un procedimiento almacenado desde una aplicación CLI/ODBC

```
SQLCHAR *stmt = (SQLCHAR *)
"CALL MEDIAN_RESULT_SET( ? )" ;
SQLDOUBLE sal = 20000.0; /* Cuerpo del parametro marcado
en stmt*/
SQLINTEGER salind = 0; /* Indicador de la variable para
sal */
```

```

        sqlrc = SQLPrepare(hstmt, stmt, SQL_NTS);
        sqlrc = SQLBindParameter(hstmt, 1, SQL_PARAM_OUTPUT,
            SQL_C_DOUBLE, SQL_DOUBLE, 0, 0, &sal, 0, &salind);
        SQLExecute(hstmt);

if (salind == SQL_NULL_DATA)
    printf("Median Salary = NULL\n");
else
    printf("Median Salary = %.2f\n\n", sal );

/* Obtención del primer conjunto de resultados */
    sqlrc = StmtResultPrint(hstmt);
/* Check for another result set */
    sqlrc = SQLMoreResults(hstmt);
if (sqlrc == SQL_SUCCESS) {
    /* Aquí otro conjunto de resultados */
    sqlrc = StmtResultPrint(hstmt);
}

```

Para más detalles, consulte el archivo de ejemplo de DB2: `sqllib/samples/sqlproc/rsultset.c`

Ejemplo de llamada a un procedimiento almacenado desde una aplicación VB.NET

```

Try
    `Crear un DB2Command para ejecutar el procedimiento almacenado
    Dim procName As String = "TRUNC_DEMO"
    Dim cmd As DB2Command = conn.CreateCommand()
    Dim parm As DB2Parameter

    cmd.CommandType = CommandType.StoredProcedure
    cmd.CommandText = procName

    `Registro de la salida de parámetros para la DB2Command
    parm = cmd.Parameters.Add("v_lastname", DB2Type.VarChar)
    parm.Direction = ParameterDirection.Output
    parm = cmd.Parameters.Add("v_msg", DB2Type.VarChar)
    parm.Direction = ParameterDirection.Output

    ` Llamada a un procedimiento almacenado
    Dim reader As DB2DataReader = cmd.ExecuteReader

Catch myException As DB2Exception
    DB2ExceptionHandler(myException)
Catch
    UnhandledExceptionHandler()
End Try

```

Ejemplo de llamada a un procedimiento almacenado desde una aplicación Java

```
try
{
    // Conectar a base de datos de ejemplo
    String url = "jdbc:db2:sample";
    con = DriverManager.getConnection(url);

    CallableStatement cs = con.prepareCall("CALL trunc_demo(?, ?)");

    // registro de parámetros de salida
    callStmt.registerOutParameter(1, Types.VARCHAR);
    callStmt.registerOutParameter(2, Types.VARCHAR);

    cs.execute();
    con.close();
}
catch (Exception e)
{
    /* Aquí va el manejo de excepciones lógica */
}
```

14.6 Errores y manejo de condiciones

En DB2, las palabras claves SQLCODE y SQLSTATE se usan para determinar el éxito o fracaso de la ejecución de una sentencia SQL. Estas palabras claves deben ser explícitamente declaradas en el ámbito externo de la aplicación que ejecuta el procedimiento de la siguiente manera:

```
DECLARE SQLSTATE CHAR(5);
DECLARE SQLCODE INT;
```

DB2 asignará automáticamente los valores de las palabras claves anteriores después de cada operación SQL. Por el SQLCODE, los valores se asignan de la siguiente manera:

- = 0, con éxito.
- > 0, con éxito de alerta
- <0, que pierda
- = 100, no se encontró datos. (por ejemplo: FETCH declaración no ha devuelto ningún dato)

Por el SQLSTATE, los valores se asignan de la siguiente manera:

- éxito: SQLSTATE'00000 '

- no encontrada: SQLSTATE'02000 '
- advertencia: SQLSTATE'01XXX '
- excepción: todos los demás valores

El SQLCODE es específico a un RDBMS y más detallado que el SQLSTATE. El SQLSTATE es norma entre los RDBMSs pero es muy general en naturaleza. Varios SQLCODEs pueden adaptarse a un SQLSTATE. SQLCODEs y SQLSTATEs se discuten en más detalle en el capítulo 18, Solución de problemas.

Una condición puede ser planteada por cualquier sentencia SQL y que coincida con un SQLSTATE. Por ejemplo, una condición específica como SQLSTATE'01004 ' es planteada cuando un valor se trunca durante la operación de un SQL. En lugar de utilizar SQLSTATE'01004 ' para la prueba de esta condición, el nombre de la condición puede ser utilizada. En este ejemplo, el nombre "trunc" pueden asignarse a condición SQLSTATE'01004 ', como se indica a continuación.

```
DECLARE trunc CONDITION FOR SQLSTATE '01004'
```

Otros condiciones predefinidas generales son las siguientes:

- SQLWARNING (Advertencia)
- SQLEXCEPTION (Exepción)
- NOT FOUND (No encontrado)

Manejo de condiciones

Para manejar una condición, debe crear una condición de manejo donde se especifica:

- Que condiciones que manejará
- Dónde reanudar la ejecución (basado en el tipo de manejador de la condición: CONTINUE, EXIT o UNDO)
- Las acciones a realizar para manejar la condición. Las acciones pueden ser cualquier sentencia, incluyendo las estructuras de control.

Si una condición SQLEXCEPTION se lanza, y no hay manejador, el procedimiento termina y devuelve al cliente con un error.

Tipos de manipuladores

Hay tres tipos de manipuladores:

CONTINUE - Este manejador se usa para indicar que después de que una excepción se lanza, y el manejador se encarga de la condición, el flujo continúe con la siguiente sentencia después de la declaración que lanzó la condición.

EXIT - Este manejador se usa para indicar que, después de que una excepción se lanza, y el manejador se encarga de la condición, el flujo irá al final del procedimiento.

UNDO - Este manejador se usa para indicar que después de que una excepción se lanza, y el manejador se encarga de la condición, el flujo va al final del procedimiento, y desechará las sentencias realizadas.

La figura 14.8 muestra las diferentes condiciones de los manipuladores y su comportamiento.

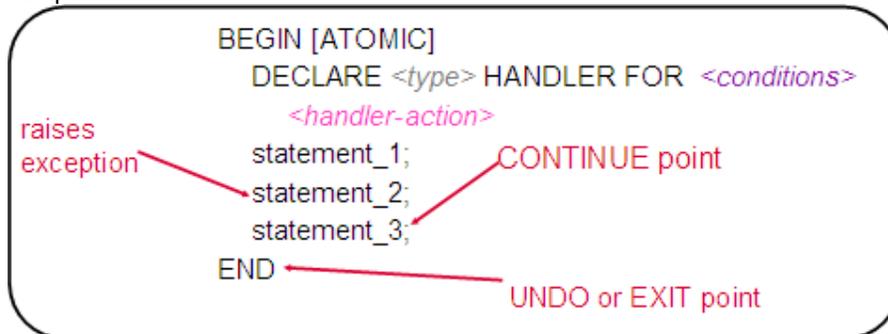


Figura 14.8 - Tipo de manipuladores de condiciones

14.7 SQL Dinámico

En SQL dinámico, en contraposición a SQL estático, no todos los elementos de la sentencia SQL se conocen si no hasta el tiempo de ejecución. Por ejemplo, si col1 y tabname son variables en la siguiente sentencia, estaríamos tratando con un SQL dinámico:

```
'SELECT ' || col1 || ' FROM ' || tabname;
```

SQL dinámico en un procedimiento almacenado es también recomendado para ejecutar sentencias DDL para la creación de objetos. Esto evita problemas de dependencias e invalidación de paquetes. SQL dinámico es también necesario para lógica de recursividad.

SQL dinámico puede ser ejecutado utilizando dos enfoques:

- Uso de la declaración EXECUTE IMMEDIATE - ideal para una sola ejecución de SQL
- Uso de la declaración PREPARE junto con la declaración EXECUTE - ideal para la ejecución de muchas sentencias SQL.

El siguiente código muestra un ejemplo de SQL dinámico usando los dos enfoques. El ejemplo asume que la tabla T2 ha sido creada con esta definición:

```

CREATE TABLE T2 (c1 INT, c2 INT)

CREATE PROCEDURE dyn1 (IN value1 INT, IN value2 INT)
SPECIFIC dyn1
BEGIN
DECLARE stmt varchar(255);
  
```

```
DECLARE st STATEMENT;  
  
SET stmt = 'INSERT INTO T2 VALUES (?, ?)';  
  
PREPARE st FROM stmt;  
  
EXECUTE st USING value1, value1;  
EXECUTE st USING value2, value2;  
  
SET stmt = 'INSERT INTO T2 VALUES (9,9)';  
EXECUTE IMMEDIATE stmt;  
END
```


15

Capítulo 15 – SQL PL en-línea, Disparadores (Triggers) y UDFs

En este capítulo, se discutirá el SQL PL en-línea y otros objetos de aplicación de la base de datos como son las funciones definidas por el usuario (user defined functions, UDFs) y los disparadores (triggers).

Note:

Para más información acerca de disparadores y UDFs vea estos videos:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4367>

<http://www.channeldb2.com/video/video/show?id=807741:Video:4362>

15.1 SQL PL en-línea

En el capítulo 14 se discutió la creación de procedimientos almacenados utilizando el lenguaje SQL PL. El lenguaje SQL PL puede ser utilizado “en-línea”, esto es que puede ser utilizado sin necesitar crear un procedimiento almacenado. El SQL PL utilizado en las UDFs u los disparadores es también en-línea debido a que el código es agregado en-línea con el código de la UDF o el disparador, y es SQL dinámico. El SQL PL en-línea es solamente un subconjunto de todos los enunciados SQL PL. A continuación se muestran las palabras clave soportadas.

```
DECLARE <variable>
SET
CASE
FOR
GET DIAGNOSTICS
GOTO
IF
RETURN
SIGNAL
WHILE
ITERATE
LEAVE
```

Estas son las palabras no soportadas:

```
ALLOCATE CURSOR
```

```

ASSOCIATE LOCATORS
DECLARE <cursor>
DECLARE ...HANDLER
PREPARE
EXECUTE
EXECUTE IMMEDIATE
LOOP
REPEAT
RESIGNAL
CALL
COMMIT/ROLLBACK

```

Este es un ejemplo de SQL dinámico utilizando SQL PL en línea. Si quiere probarlo, debe ponerlo como parte de un archivo de script, y asegurarse de crear las siguientes tablas.

```

CREATE TABLE T1 (c1 INT)
CREATE TABLE T3 (c1 INT)

```

```

BEGIN ATOMIC
  DECLARE cnt          INT DEFAULT 0;
  DECLARE sumevens INT DEFAULT 0;
  DECLARE err_msg VARCHAR(1000) DEFAULT '';
  WHILE (cnt < 100) DO
    IF mod(cnt,2) = 0 THEN
      SET sumevens = sumevens + cnt;
    END IF;
    SET cnt=cnt+1;
  END WHILE;
  INSERT INTO T3 values (sumevens);
  SET cnt = (SELECT 0 FROM SYSIBM.SYSDUMMY1);
  FOR cur1 AS SELECT * FROM T1 DO
    IF cur1.c1 > 100 THEN
      SET cnt = cnt + 1;
    END IF;
  END FOR;

  SET err_msg = 'Rows with values > 100 is:' || char(cnt);
  SIGNAL SQLSTATE '80000' SET MESSAGE_TEXT = err_msg;
END!

```

Si se guarda el código anterior en un archivo llamado "myScript.txt" se podría ejecutar de esta manera:

```
db2 -td! -vf myScript.txt
```

15.2 Triggers (Disparadores)

Los disparadores son objetos de la base de datos asociados con una tabla que define operaciones que ocurren cuando una operación INSERT, UPDATE o DELETE es realizada en la tabla. Son activados automáticamente. Las operaciones que causan lanzar los disparadores son llamados “triggering SQL statements”

15.2.1 Tipos de disparadores

Hay tres tipos de disparadores: “before” o “antes”, “after” o “después” e “instead”o “en lugar de”.

Before triggers

Este tipo de disparadores son activados antes que una fila sea insertada, actualizada o borrada. Las operaciones realizadas por este disparador no puede activar otros disparadores (entonces las operaciones INSERT, DELETE y UPDATE no son permitidas).

Un ejemplo de un before trigger simple es mostrado en la figura 15.1.

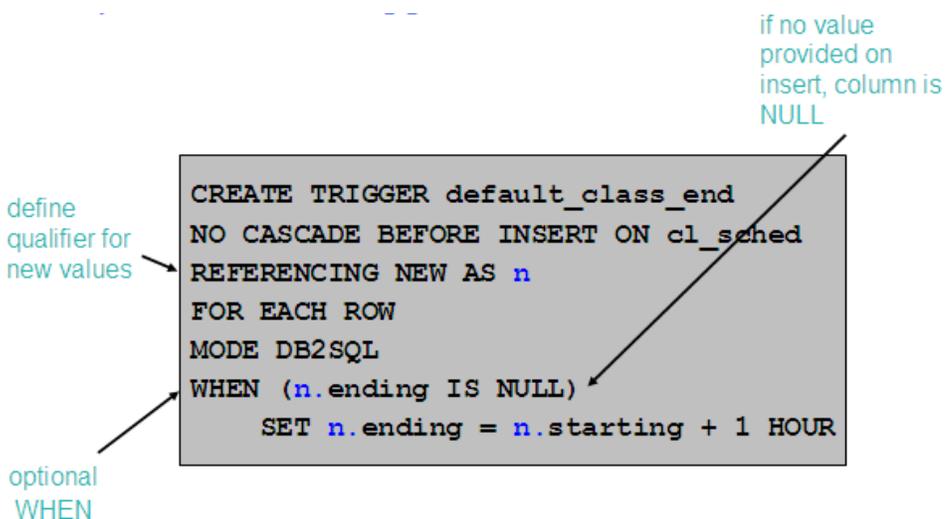


Figura 15.1 – Ejemplo de un before trigger.

En la figura 15.1 el disparador “default_class_end” sera disparado antes de que la sentencia INSERT se efectúe en la tabla cl_sched. Esta tabla es parte de la base de datos SAMPLE, por lo cual usted puede crear y probar este disparador mientras se este conectado a esta base de datos.

La variable “n” en la definición del trigger representa el nuevo valor en un INSERT, esto es, el valor que esta siendo insertado. El disparador verificará la validez de lo que está siendo insertado a la tabla. Si la columna “ending” no tiene un valor durante la inserción, el disparador asegurará que tenga el valor de la columna “starting” mas 1 hora.

Los siguientes enunciados muestran como probar el disparador.

```

C:\Program Files\IBM\SQLLIB\BIN>db2 insert into cl_sched
(class_code, day, starting) values ('abc',1,current time)
DB20000I The SQL command completed successfully.

C:\Program Files\IBM\SQLLIB\BIN>db2 select * from cl_sched

CLASS_CODE DAY      STARTING ENDING
-----
042:BF      4 12:10:00 14:00:00
553:MJA     1 10:30:00 11:00:00
543:CWM     3 09:10:00 10:30:00
778:RES     2 12:10:00 14:00:00
044:HD     3 17:12:30 18:00:00
abc        1 11:06:53 12:06:53

6 record(s) selected.

```

El disparador “validate_sched” mostrado a continuación extiende la funcionalidad de el disparador “default_class_end” previamente descrito. Nuevamente, usted puede crear y probar este disparador en la base de datos SAMPLE.

```

CREATE TRIGGER validate_sched
NO CASCADE BEFORE INSERT ON cl_sched
REFERENCING NEW AS n
FOR EACH ROW
MODE DB2SQL
BEGIN ATOMIC
-- supply default value for ending time if null
IF (n.ending IS NULL) THEN
    SET n.ending = n.starting + 1 HOUR;
END IF;

-- ensure that class does not end beyond 9pm
IF (n.ending > '21:00') THEN
    SIGNAL SQLSTATE '80000'
    SET MESSAGE_TEXT='class ending time is beyond 9pm';
ELSEIF (n.DAY=1 or n.DAY=7) THEN
    SIGNAL SQLSTATE '80001'
    SET MESSAGE_TEXT='class cannot be scheduled on a weekend';
END IF;
END

```

After triggers

Estos disparadores son activados después que la sentencia SQL se han ejecutado exitosamente. La operaciones realizadas por este disparador pueden activar otros

disparadores (Las operaciones en cascada son permitidas hasta en 16 niveles). Estos disparadores soportan las operaciones INSERT, UPDATE y DELETE. A continuación se muestra un ejemplo de uno de estos disparadores.

```
CREATE TRIGGER audit_emp_sal
AFTER UPDATE OF salary ON employee
REFERENCING OLD AS o NEW AS n
FOR EACH ROW
MODE DB2SQL
  INSERT INTO audit VALUES (
    CURRENT TIMESTAMP, ' Employee ' || o.empno || ' salary changed
from ' || CHAR(o.salary) || ' to ' || CHAR(n.salary) || ' by ' ||
USER)
```

En este ejemplo, el disparador `audit_emp_sal` es utilizado para realizar auditoria en la columna “salary” de la tabla “employee”. Cuando alguien hace un cambio en esta columna, el disparador será lanzado para escribir información acerca del cambio hecho en el salario en otra tabla llamada “audit”. La línea “OLD as o NEW as n” indica que el prefijo “o” será utilizado para representar el valor anterior o el existente en la tabla, y el prefijo “n” será utilizado para representar el nuevo valor que viene de la operación UPDATE. Por lo que “o.salary” representa el valor anterior o el actual del salario y “n.salary” representa el valor actualizado para la columna salario.

“Instead of” triggers

Este tipo de disparadores son definidos en vistas. La lógica definida en los disparadores es ejecutada en lugar de la sentencia SQL. Por ejemplo, si se realiza una operación UPDATE en la vista, el disparador realizará una actualización en las tablas base que forman la vista.

Los disparadores no pueden ser creados desde el IBM Data Studio. Ellos pueden ser creados desde el Control Center o desde las herramientas de línea de comandos (Command Window, Command Line Processor, o el Command Editor).

Quicklab #12 – Creación de un disparador en el Centro de Control

Objetivo

Los Triggers son objetos de la Base de Datos usados para ejecutar lógica de negocios, cuando una operación en la tabla ocurre. En este ejercicio, se creará un trigger usando el Centro de Control. Este trigger mantendrá un registro o log de los cambios hechos a la tabla *SALES* con fines de auditoría. Se registrará el Id del usuario que hizo el cambio, junto con la hora en que el cambio fue hecho.

Procedimiento

1. Abrir el Centro de Control.
2. Para este ejercicio, es necesario crear una tabla adicional para ser usada para guardar el registro de los cambios. Crear esta tabla con las siguientes características:

Nombre: `saleslog`

Columna1:

Nombre: `userid`

Tipo de dato: `VARCHAR(128)`

Otros atributos: `NOT NULL`

Columna 2:

Nombre: `daytime`

Tipo de dato: `TIMESTAMP`

Otros atributos: `NOT NULL`

Consejo: Crear esta tabla usando la sentencia `CREATE TABLE` en el Editor de comandos, o use el asistente de `Create Table` en el Centro de Control.

3. Desde el Centro de Control, expandir la carpeta Base de Datos *EXPRESS*. Haga clic derecho en la carpeta *Triggers* y seleccionar la opción *Create*. Se abrirá la ventana de diálogo *Create Trigger*.
4. Llenar la siguiente información en el dialogo:

Trigger de esquema: el Id de Usuario con el que el usuario se ha conectado (es el seteo por default)

Nombre del Trigger: `audit_sales`

Tabla o vista de esquema: El ID del Usuario conectado

Nombre de la Vista o de la Tabla: *SALES*

Momento de la accion del trigger: *After*

Operación que causa que el trigger sea ejecutado: *Update de columnas*
(no especificar ninguna columna por que es deseable que el trigger se ejecute cuando cualquiera de las columnas sean actualizadas).

Comentario: Registrará todos los updates sobre la tabla *Sales*.

5. En la solapa *Triggered action*, seleccionar la opción *For Each STATEMENT*. Usar el siguiente código para la acción a realizar:

```
WHEN ( 1=1 )
BEGIN ATOMIC
    INSERT INTO saleslog (userid, daytime) VALUES (CURRENT
USER, CURRENT TIMESTAMP);
```

END

(Nota: La acción del trigger se ejecuta una sola vez luego que la sentencia que activa al trigger ha sido completada. Un trigger de fila significa que la acción del trigger se ejecutará cada vez que la sentencia SQL que dispara el trigger afecta a una fila.)

Hacer Clic en el botón *OK* para crear el trigger.

6. Ahora debería ser posible ver el trigger en la carpeta *Triggers* en el Centro de Control.
7. Consultar la tabla *saleslog* para asegurarse que no habrá datos en el. Haga Delete sobre cualquier fila que puede haber en ésta (`DELETE FROM saleslog`).
8. Tratar de actualizar una fila en la tabla *sales*. (Consejo: usar el Editor de Comand o el Asistente de SQL).
9. Revisar los contenidos de la tabla *saleslog* nuevamente. Cuantas filas hay en ella?

15.3 Funciones definidas por el Usuario (UDFs)

Una función definida por el usuario o en sus siglas en inglés UDF es una aplicación de base de datos que transforma un conjunto de datos de entrada en un conjunto de valores de salida. Por ejemplo, una función puede tomar una medida en pulgadas y devolver el resultado convertido en centímetros.

DB2 soporta crear funciones usando SQL PL, C/C++, Java, CLR (Common Language Runtime) y OLE (Object Linking and Embedding). En este libro, se focalizará en las funciones de SQL PL debido a su simplicidad, popularidad y performance.

Hay cuatro tipos de funciones: escalar (scalar), de tabla (table), de fila (row) y de columna (column). En este capítulo solo cubriremos las UDFs escalares y de tabla.

15.3.1 Funciones escalares

Las funciones escalares devuelven un valor simple. No pueden incluir sentencias SQL que cambien el estado de la Base de datos, o sea que las sentencias INSERT, UPDATE y DELETE no son permitidas. Algunas funciones escalares son SUM(), AVG(), DIGITS(), COALESCE() y SUBSTR().

DB2 permite construir funciones especializadas de usuario donde se encapsularán la lógica frecuentemente usada. Por ejemplo, considere una migración de su aplicación de Oracle a DB2. En su aplicación, se usa comúnmente la función de Oracle NVL(). La función equivalente en DB2 se llama COALESCE. En vez de renombrar en la aplicación se podría crear una UDF llamada NVL que invoque a la función COALESCE según el siguiente código.

```
CREATE FUNCTION NVL (p_var1 VARCHAR(30),
                    p_var2 VARCHAR(30))
SPECIFIC nvlvarchar30
RETURNS VARCHAR(30)
RETURN COALESCE(p_var1, p_var2)
```

La Función COALESCE retornará así el primer argumento que no es null.

A continuación hay otro ejemplo de funciones escalares. La función es llamada "deptname" y retorna el número de departamento de un empleado basándose en el id de empleado.

```
CREATE FUNCTION deptname(p_empid VARCHAR(6))
RETURNS VARCHAR(30)
SPECIFIC deptname
BEGIN ATOMIC
  DECLARE v_department_name VARCHAR(30);
  DECLARE v_err VARCHAR(70);
  SET v_department_name = (
    SELECT d.deptname FROM department d, employee e
```

```

        WHERE e.workdept=d.deptno AND e.empno= p_empid);
    SET v_err = 'Error: employee ' || p_empid || ' was not found';
    IF v_department_name IS NULL THEN
        SIGNAL SQLSTATE '80000' SET MESSAGE_TEXT=v_err;
    END IF;
RETURN v_department_name;
END

```

Para probar la función, realizar lo siguiente desde la Ventana de comandos o shell de Linux/UNIX:

```
db2 "values (deptname ('000300'))"
```

Invocando las UDF escalares

Las UDF escalares pueden ser invocadas en sentencias SQL donde cualquier valor escalar puede ubicarse o en una cláusula VALUES. Aquí hay dos ejemplos mostrando una llamada a la función escalar COALESCE:

```

SELECT DEPTNAME, COALESCE(MGRNO, 'ABSENT') FROM DEPARTMENT
VALUES COALESCE('A', 'B')

```

15.3.2 Funciones de Tabla

Las funciones de tabla retornan una tabla. Por esto pueden ser usadas en la cláusula FROM de una consulta. Las funciones de Tabla, opuestamente a las funciones escalares, pueden cambiar el estado de la Base de datos, por lo tanto las sentencias INSERT, UPDATE y DELETE son permitidas. Algunas funciones de tabla incluidas en DB2 son SNAPSHOT_DYN_SQL() y MQREADALL(). Las funciones de Tabla son similares a las vistas, pero ya que permiten sentencias de modificación (INSERT, UPDATE y DELETE) son mas poderosas. Típicamente son usadas para retornar y mantener registros de auditoria.

A continuación, mostramos un ejemplo de una función de tabla que enumera un conjunto de empleados de departamentos:

```

CREATE FUNCTION getEnumEmployee(p_dept VARCHAR(3))
RETURNS TABLE
    (empno CHAR(6),
     lastname VARCHAR(15),
     firstnme VARCHAR(12))
SPECIFIC getEnumEmployee
RETURN
    SELECT e.empno, e.lastname, e.firstnme
    FROM employee e
    WHERE e.workdept=p_dept

```

Para probar la función de arriba, ejecutar:

```
db2 "SELECT * FROM table(getEnumEmployee('D11')) AS t"
```

Invocación de las UDF de tabla

Una UDF de tabla debe ser invocada en la cláusula FROM de una sentencia SQL. La función TABLE() debe ser aplicada y aplicarle un alias. En la Figura 15.2 se aprecia un ejemplo de como invocar la función "getEnumEmployee" probada anteriormente.

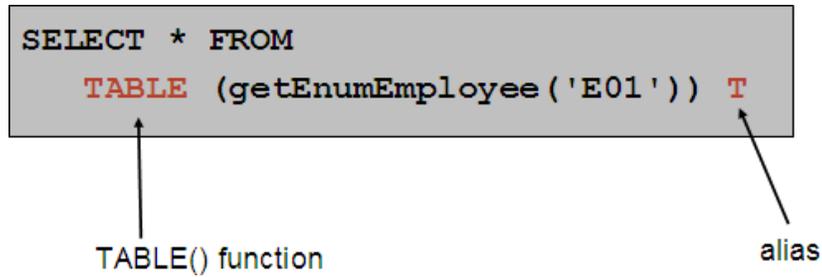


Figura 15.2 – Invocación de una función de tabla.

Quicklab #13 – Creación de un UDF utilizando el IBM Data Studio

Objetivo

En este laboratorio, se creará una función escalar definida por el usuario en IBM Data Studio. Esto proporcionará mayor experiencia con Data Studio, así mismo mejorará el conocimiento del lenguaje SQL PL para funciones definidas por el usuario.

Procedimiento

1. Abrir el IBM Data Studio.
2. Desde el Explorador de Proyectos de Datos, elegir el proyecto creado en el laboratorio anterior y seleccionar Open Project para abrir el proyecto.
3. Dar un clic derecho en el directorio User Defined Function- Seleccionar el menu New. Seleccionar el menu SQL User-Defined Function. Se podrá tener alternativamente seleccionado el elemento User-Defined Function using Wizard si se quiere ser guiado a través del proceso utilizando un asistente.

La vista Editor debe abrirse una función esqueleto, Modificar el código como se muestra:

```
CREATE FUNCTION booktitle(p_bid INTEGER)
RETURNS VARCHAR(300)
-----
SQL UDF (Scalar)
-----
SPECIFIC booktitle
F1: BEGIN ATOMIC
DECLARE v_book_title VARCHAR(300);
DECLARE v_err VARCHAR(70);
SET v_book_title = (SELECT title FROM books WHERE p_bid =
book_id);
SET v_err = 'Error: The book with ID ' || CHAR(p_bid) || '
was not found.';
IF v_book_title IS NULL THEN SIGNAL SQLSTATE '80000' SET
MESSAGE_TEXT=v_err;
END IF;
RETURN v_book_title;
END
```

4. Construir la función dando un clic derecho en la función y eligiendo Deploy.
5. Ejecutar la función dando clic en el botón Run en la barra de herramientas.
6. Como la función acepta un parámetro de entrada, una ventana de diálogo aparece pidiendo llenar con un valor para el parámetro.

Introducir el valor: 80002

¿Cuál es el resultado?

Probar de nuevo con el valor: 1002

¿Qué pasó esta vez? (Consejo: Mire en la sección Mensajes de la vista Output)

7. Cerrar el IBM Data Studio cuando se finalice.

16

Capítulo 16 – DB2 pureXML

En este capítulo discutiremos pureXML, la nueva tecnología disponible a partir de DB2 9 para soportar el almacenamiento nativo de datos XML. Muchos de los ejemplos y conceptos discutidos en este capítulo han sido tomados de los libros rojos (redbooks) de IBM: *DB2 9: pureXML overview and fast start*. Vea la sección Recursos para más información. La Figura 16.1 esboza cual sección de DB2 “Big Picture” vamos a discutir en este capítulo.

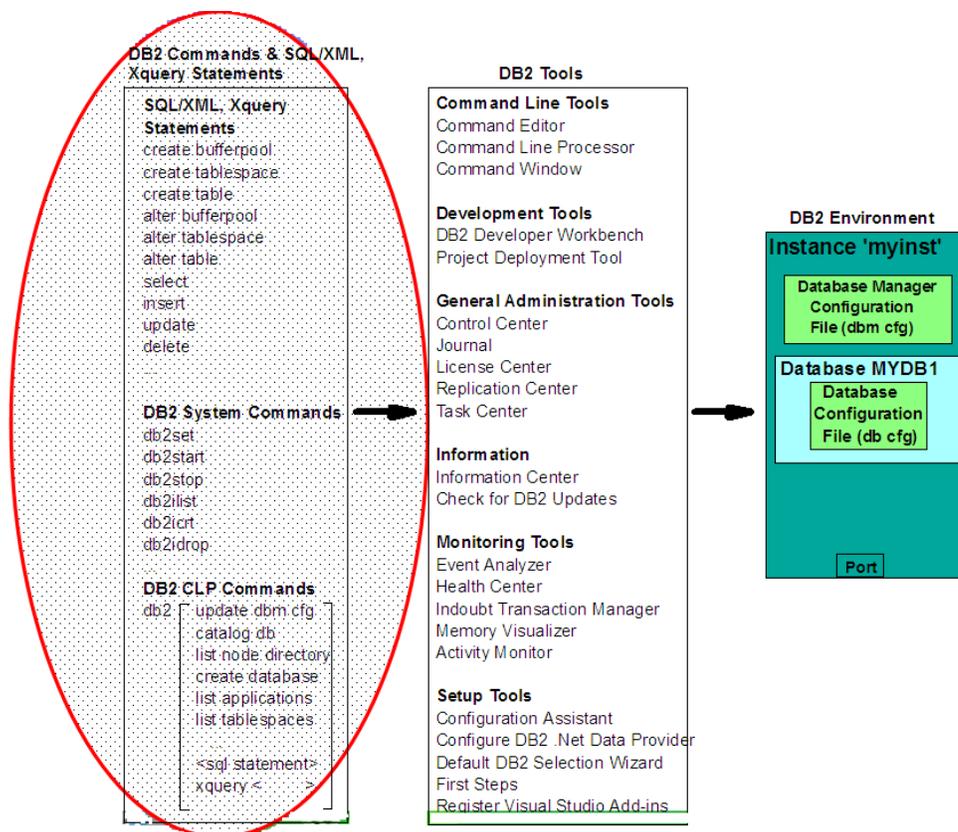


Figura 16.1 – DB2 Visión General: Comandos DB2, SQL/XML y XQuery

Nota:

Para más información acerca de pureXML, vea este video:
<http://www.channeldb2.com/video/video/show?id=807741:Video:4382>

16.1 Uso de XML en bases de datos

Los documentos XML pueden ser almacenados en archivos de texto, repositorios XML, o base de datos. Hay dos razones principales por las que muchas compañías proponen almacenarlos en una base de datos:

- La gestión de largos volúmenes de datos XML es un problema de base de datos. Datos en XML son datos como cualquier otros, solo que en diferente formato. Los mismos criterios para almacenar datos relacionales en las bases de datos se aplican a datos XML: Las bases de datos ofrecen búsquedas y recuperaciones eficientes, soporte robusto para persistencia de datos, back up y recuperación, soporte de transacción, rendimiento y escalabilidad.
- Integración: Al almacenar datos relacionales y documentos XML juntos, usted puede integrar los datos XML con datos relacionales existentes. Por otro lado, los datos relacionales pueden ser publicados como XML y viceversa. A través de la integración, las bases de datos pueden soportar aplicaciones Web, SOA, y servicios Web.

16.2 Bases de datos XML

Hay dos tipos de bases de datos para almacenar datos XML:

- Bases de datos XML-enabled
- Bases de datos Native XML

16.2.1 Bases de datos XML-enabled

Una base de datos XML-enabled utiliza un modelo relacional para el centro de datos del modelo de almacenamiento. Esto requiere un mapeo entre el modelo de datos XML (jerárquico) y el modelo de datos relacional o bien almacenar los datos XML como un objeto carácter largo. Si bien esto puede considerarse como una “vieja” tecnología, aun esta siendo usado por muchos proveedores de bases de datos. La figura 16.2 explica con más detalle las dos opciones para bases de datos XML-enabled.

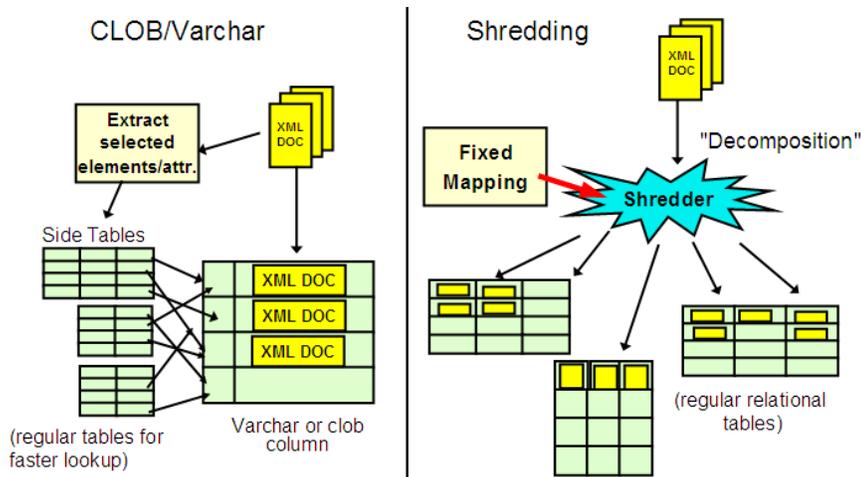


Figura 16.2 – Dos opciones para almacenar XML en una base de datos XML-enabled

EL lado izquierdo de la figura 16.2 muestra el método "CLOB/Varchar" de almacenamiento de documentos XML en una base de datos. Usando este método, un documento XML es almacenado como una cadena no parseada tanto en base de datos CLOB como Varchar. Si el documento XML es almacenado como cadena, cuando se quiere recuperar parte del documento XML, su programa tiene que recuperar la cadena, y efectuar el parseo para encontrar lo que desea. Este método no es muy flexible.

La otra opción para bases de datos XML-enabled se llama Trituración o descomposición y esta ilustrada en el lado derecho de la figura 16.2. Usando este método, un documento XML entero es desmenuzado en partes más pequeñas que son almacenadas en tablas. Usando este método, el modelo jerárquico de un documento XML es forzado dentro del modelo relacional. Esto tampoco es bueno para la flexibilidad: un cambio en el documento XML no es fácilmente propagable en las correspondientes tablas y muchas otras tablas pueden necesitar ser creadas. Este método tampoco es bueno para el rendimiento: si necesita obtener el documento XML original de vuelta, tendrá que realizar una costosa operación de SQL, lo que aumenta aun más cuando hay más tablas unidas.

16.2.2 Bases de datos Nativa XML

Las bases de datos Nativa XML usan el modelo de datos XML jerárquico para almacenar y procesar XML internamente. El formato de almacenamiento es el mismo que el formato de proceso: no existe mapeo para el modelo relacional, y los documentos XML no son almacenados como imágenes. Cuando las declaraciones XPath o XQuery son usadas, son procesadas en forma nativa por el motor, y no se convierten a SQL. Esta es la razón por la que las bases de datos son conocidas como bases de datos XML "nativa". DB2 comenzando con la versión 9 es actualmente el único servidor comercial que provee este soporte.

16.3 XML en DB2

La figura 16.3 describe a continuación como datos relacionales y datos jerárquicos (documentos XML) son almacenados en DB2 comenzando con la versión 9. En la figura, asumimos que la tabla dept esta definida como sigue:

```
CREATE TABLE dept (deptID CHAR(8),..., deptdoc XML);
```

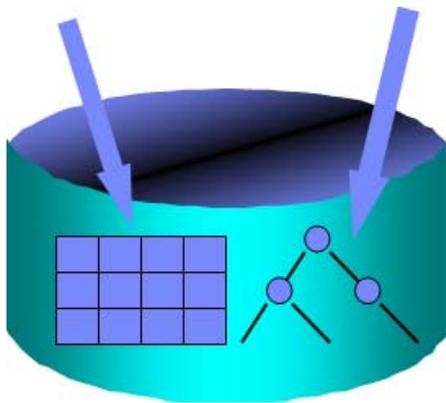


Figura 16.3 – XML en DB2

Nótese que la definición de la tabla usa un nuevo tipo de dato, XML, para la columna `deptdoc`. La flecha de la izquierda en la figura indica la columna relacional `deptID` almacenada en formato relacional (tabla), mientras que la columna XML `deptdoc` esta almacenada en formato jerárquico parseado.

La figura 16.4 ilustra que comenzando con DB2 9, hay ahora cuatro nuevos caminos para acceder a los datos.

- Uso de SQL para acceder a datos relacionales
- Uso de SQL con extensión XML para acceder a datos XML
- Uso de XQuery para acceder a datos XML
- Uso de XQuery para acceder a datos relacionales

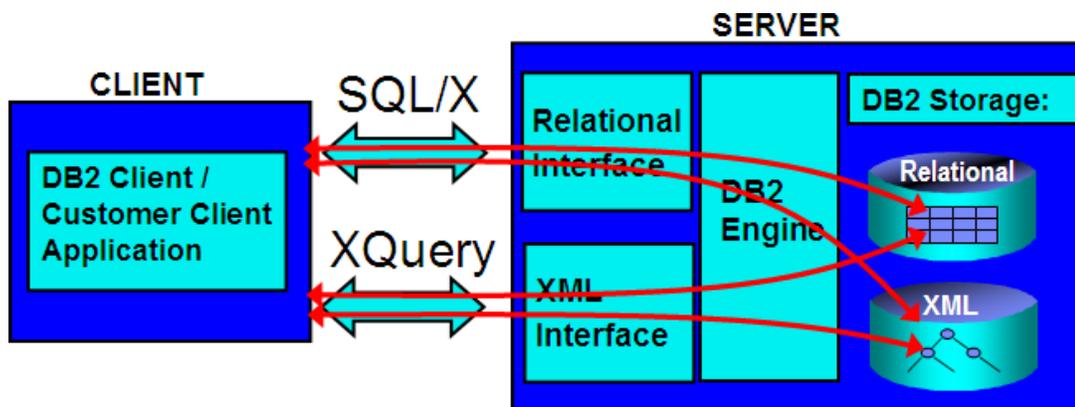


Figura 16.4 – Cuatro caminos para acceder a los datos en DB2

Por lo tanto, mientras una persona con habilidad en SQL puede ver a DB2 como un RDBMS de clase mundial que también soporta XML, una persona con habilidad en XML vería a DB2 como un repositorio de XML de clase mundial que también soporta SQL.

Nótese que IBM usa el termino **pureXML** en lugar de “XML nativo” para describir esta tecnología. Mientras que otros proveedores aun utilizan la vieja tecnología de CLOB/varchar o trituración para almacenar documentos XML, ellos llaman a esas viejas tecnologías “XML nativo”. Para evitar confusiones, IBM decidió usar el termino pureXML, y registra este nombre de manera que ninguna otra base de datos o proveedor pueda utilizar el mismo termino para referirse a diferencias tecnológicas. En DB2 9.1, el soporte pureXML fue proporcionado solo cuando la base de datos era creada como base de datos Unicode. Esta restricción ha sido levantada con DB2 9.5, donde se puede usar pureXML en bases de datos Unicode o no unicode.

16.3.1 Ventajas tecnológicas de pureXML

Muchas ventajas son proporcionadas por el uso de la tecnología pureXML.

1. Puede perfectamente aprovechar su inversión relacional, dado que los documentos XML son almacenado en las columnas de las tablas usando el nuevo tipo de datos XML.
2. Puede reducir la complejidad del código. Por ejemplo, la figura 16.5 ilustra un script PHP escrito usando y sin usar pureXML. Usando XML (la caja verde) las líneas de código son reducidas. Esto no solo significa que el código es menos complejo, el rendimiento general se mejora ya que hay menos líneas para analizar y mantener en el código.

```

<?php
$conn = db2_connect($dbname, $dbuser, $dbpass);

/* Insert Customer Documents */

$stmt = db2_prepare($conn, "VALUES (NEXT VALUE FOR
Cid)");
db2_execute($stmt);
list($Cid) = db2_fetch_array($stmt);

$fileContents = file_get_contents
("customers/c1.xml");

$stmt = db2_prepare($conn, "INSERT INTO xmlicustomer
(Cid, Info) VALUES (?, ?)");
if(!db2_execute($stmt, array($Cid, $fileContents)))

    echo db2_stmt_errormsg($stmt);
}

/* Insert Product Documents */

$fileContents = file_get_contents
("products/p1.xml");
$dom = simplexml_load_string($fileContents);

$prodID = (string) $dom["pid"];

$stmt = db2_prepare($conn, "INSERT INTO xmlproduct
(Pid, Description) VALUES (?, ?)");
if(!db2_execute($stmt, array($prodID,

```

```

Cid) ??
db2_execute($stmt);
list($Cid) = db2_fetch_array($stmt);

$fileContents = file_get_contents
("customers/c1.xml");
$dom = simplexml_load_string($fileContents);

$custName = (string) $dom->name;
$custCountry = (string) $dom->addr["country"];
$custStreet = (string) $dom->addr->street;
$custCity = (string) $dom->addr->city;
$custProvince = (string) $dom->addr->("prov-state");
$custZip = (string) $dom->addr->("pcode-zip");
$custPhone = (string) $dom->phone;

$stmt = db2_prepare($conn, "INSERT INTO sqlcustomer
(Cid, Name, Country, Street, City, Province, Zip,
Phone, Info) VALUES (?, ?, ?, ?, ?, ?, ?, ?)");
if(!db2_execute($stmt, array($Cid, $custName,
$custCountry, $custStreet, $custCity, $custProvince,
$custZip, $custPhone, $fileContents) )) {
    echo db2_stmt_errormsg($stmt);
}

/* Insert Product Documents */

$fileContents = file_get_contents
("products/p1.xml");
$dom = simplexml_load_string($fileContents);

$prodID = (string) $dom["pid"];

```

Figura 16.5 – Complejidad de código con y sin pureXML

- Los cambios en su esquema son más fáciles usando tecnología pureXML. La figura 16.6 ilustra un ejemplo de este aumento de flexibilidad. En la figura, suponga que tiene una base de datos consistente de las tablas Empleado y Departamento. Típicamente con una base de datos no-XML, si su gerente le pidió almacenar no solo un número telefónico por empleado (el número telefónico del hogar), sino también un segundo número telefónico (un número de teléfono móvil), entonces puede agregar una columna extra a la tabla Empleado y almacenar el número de teléfono móvil en esta nueva columna. Sin embargo, este método estaría en contra de las reglas de normalización de las bases de datos relacionales. Si desea conservar estas normas, debería crear una nueva tabla de teléfono y mover toda la información telefónica a esta nueva tabla. Puede añadir a continuación los teléfonos móviles también. Crear una nueva tabla "teléfono" es costoso, no solo por las grandes cantidades de datos pre-existentes que necesitan ser movidos, sino también porque todas las SQL en su aplicación deberían ser cambiadas para apuntar a la nueva tabla.

En la parte izquierda de la figura, mostramos como esto puede hacerse usando XML. Si la empleada "Cristina" tiene un número de teléfono móvil, una nueva etiqueta se puede añadir a esta información. Si el empleado "Michael" no posee teléfono móvil, solamente lo dejamos como está.

```

<DEPARTMENT deptid="15" deptname="Sales">
  <EMPLOYEE>
    <EMPNO>10</EMPNO>
    <FIRSTNAME>CHRISTINE</FIRSTNAME>
    <LASTNAME>SMITH</LASTNAME>
    <PHONE>408-463-4963</PHONE>
    <PHONE>415-010-1234</PHONE>
    <SALARY>52750.00</SALARY>
  </EMPLOYEE>
  <EMPLOYEE>
    <EMPNO>27</EMPNO>
    <FIRSTNAME>MICHAEL</FIRSTNAME>
    <LASTNAME>THOMPSON</LASTNAME>
    <PHONE>406-463-1234</PHONE>
    <SALARY>41250.00</SALARY>
  </EMPLOYEE>
</DEPARTMENT>
    
```

Requires:

- Normalization of existing data !
- Modification of the mapping
- Change of applications

Phone

EMPNO	PHONE
27	406-463-1234
10	415-010-1234
10	408-463-4963

Department

DEPTID	DEPTNAME
15	Sales

Costly!

Employee

DEPTID	EMPNO	FIRSTNAME	LASTNAME	PHONE	SALARY
15	27	MICHAEL	THOMPSON	406-463-1234	41250
15	10	CHRISTINE	SMITH	408-463-4963	52750

Figura 16.6 – Mayor flexibilidad de datos usando XML

4. Puede mejorar el rendimiento de su aplicación XML. Pruebas realizadas usando tecnología pureXML mostraron enormes mejoras en el rendimiento de varias aplicaciones XML. La figura 16.7 muestra el resultado de la prueba para una compañía que pasa a pureXML desde viejas tecnologías. La columna del medio muestra el resultado usando el viejo método trabajando con XML, y la tercera columna muestra el resultado usando XML.

Tarea	Otra BD relacional	DB2 con pureXML
Desarrollo de búsqueda y recuperación de procesos de negocios	CLOB: 8 hrs Shred: 2 hrs	30 min.
Líneas relativas a código de E/S	100	35 (65% reducción)
Añadir a campo de esquema	1 semana	5 min.
Consultas	24 - 36 hrs	20 seg. - 10 min

Figura 16.7 – Aumento de rendimiento utilizando tecnología pureXML

16.3.2 XPath básico

Xpath es un lenguaje que puede ser usado para consultar documentos XML. La figura 16.8 muestra un documento XML, y la figura 16.9 ilustra el mismo documento representado en formato parseo-Jerárquico (también llamado “hoja” o “nodo”). Nosotros vamos a usar el formato Parseo-Jerárquico para explicar como funciona XPath.

```

<dept bldg="101">
  <employee id="901">
    <name>John Doe</name>
    <phone>408 555 1212</phone>
    <office>344</office>
  </employee>
  <employee id="902">
    <name>Peter Pan</name>
    <phone>408 555 9918</phone>
    <office>216</office>
  </employee>
</dept>

```

Figura 16.8 – Un documento XML

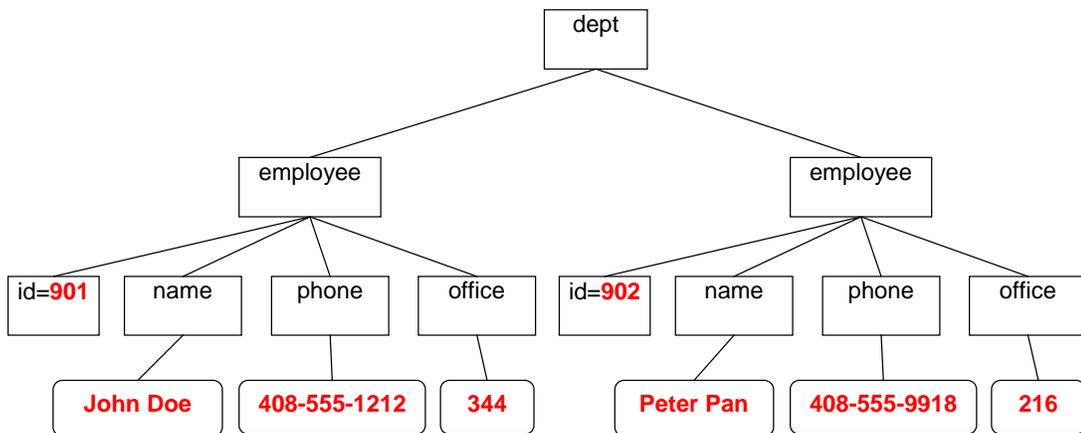


Figure 16.9 – Representación Parseo-Jerárquica del documento XML en la Fig. 16.8

Un rápido camino para aprender XPath es compararlo con el comando `change directory` (CD) en MS-DOS o Linux/UNIX. Con el uso del comando CD puede recorrer el árbol del directorio como sigue:

```
CD /directorio1/directorio2/...
```

De manera similar, en XPath usted usa barras para ir de un elemento a otro dentro de un documento XML. Por ejemplo, usando el documento en la figura 16.9 en XPath puede recuperar los nombres de todos los empleados usando esta consulta:

```
/dept/employee/name
```

Expresiones XPath

Las expresiones XPath utilizan caminos totalmente calificados para especificar elementos y/o atributos. Un signo “@” se usa para especificar un atributo. Para recuperar solo el valor (nodo texto) de un elemento, usar la función “text()”. La tabla 16.1 muestra consultas XPath y los correspondientes resultados usando el documento XML de la figura 16.9

XPath	Resultado
/dept/@bldg	101
/dept/employee/@id	901 902
/dept/employee/name	<name>Peter Pan</name> <name>John Doe</name>
/dept/employee/name/text()	Peter Pan John Doe

Tabla 16.1 – Ejemplo de expresiones XPath**Comodines XPath**

Existen dos comodines principales en XPath:

- “*” coincide con cualquier final de nombre
- “//” es el comodín “descendente o propio”

La tabla 16.2 proporciona más ejemplos usando documentos XML de la figura 16.6

XPath	Resultados
/dept/employee/*/text()	John Doe 408 555 1212 344 Peter Pan 408 555 9918 216
/dept*/@id	901 902
//name/text()	Peter Pan John Doe
/dept//phone	<phone>408 555 1212</phone> <phone>408 555 9918</phone>

Tabla 16.2 – Ejemplo de comodines XPath**Predicados XPath**

Los predicados están encerrados entre corchetes []. Como una analogía, puede pensar en ellos como el equivalente a la cláusula WHERE en SQL. Por ejemplo [@id="902"] puede ser leído como “WHERE atributo id es igual a 902”. Pueden haber múltiples predicados en una expresión XPath. Para especificar un predicado posicional, use [n] que significa que el enésimo niño sería seleccionado. La tabla 16.3 proporciona más ejemplos.

XPath	Resultados
/dept/employee[@id="902"]/name	<name>Peter Pan</name>
/dept[@bldg="101"]/employee[office > "300"]/name	<name>John Doe</name>
//employee[office="344" OR office="216"]/@id	901 902
/dept/employee[2]/@id	902

Tabla 16.3 – Ejemplos de predicados XPath

XPath: El eje padre.

Al igual que en MS-DOS o Linux/UNIX, puede usar un "." (Punto) para indicar en la expresión, que usted se refiere al contexto actual, y ".." (Punto punto) para referirse al contexto padre. La tabla 16.4 proporciona más ejemplos.

XPath	Resultados
/dept/employee/name[../@id="902"]	<name>Peter Pan</name>
/dept/employee/office[.>"300"]	<office>344</office>
/dept/employee[office > "300"]/office	<office>344</office>
/dept/employee[name="John Doe"]/../@bldg	101
/dept/employee/name[.="John Doe"]/../@bldg	101

Tabla 16.4 – Eje padre en XPath

16.3.3 XQuery: Definición

XQuery es un lenguaje de consulta creado para XML. XQuery soporta expresiones de rutas para navegar la estructura jerárquica de XML. De hecho, XPath es un subconjunto de XQuery; por lo que todo lo aprendido anteriormente sobre XPath se aplica también a XQuery. XQuery soporta datos con tipo y datos sin tipo. XQuery carece de valores nulos porque los documentos XML omiten la falta de datos o los datos desconocidos. XQuery devuelve secuencias de datos XML. Es importante notar que las expresiones de XQuery y XPath distinguen entre mayúsculas y minúsculas.

XQuery soporta la expresión FLWOR. Si usamos SQL para analogía, es equivalente a la expresión SELECT-FROM-WHERE. La sección siguiente describe FLWOR con más detalle.

XQuery: La expresión FLWOR

Significado de FLWOR:

- FOR (Para): itera a través de una secuencia, une una variable a los elementos.
- LET (Permitir): Une una variable a una secuencia.
- WHERE (Donde): elimina los elementos de la iteración.
- ORDER (Orden): reordena los elementos de la iteración.
- RETURN (retorno): Construye resultados de consultas.

Es una expresión que permite la manipulación de documentos XML, permitiendo devolver otra expresión. Por ejemplo, suponga que tiene una tabla con esta definición.

```
CREATE TABLE dept(deptID CHAR(8),deptdoc XML);
```

Y el siguiente documento XML es insertado en la columna deptdoc:

```

<dept bldg="101">
  <employee id="901">
    <name>John Doe</name>
    <phone>408 555 1212</phone>
    <office>344</office>
  </employee>
  <employee id="902">
    <name>Peter Pan</name>
    <phone>408 555 9918</phone>
    <office>216</office>
  </employee>
</dept>

```

Entonces las siguiente sentencia XQuery utilizando la expresión FLWOR se podría ejecutar:

```

xquery
for $d in db2-fn:xmlcolumn('dept.deptdoc')/dept
let $emp := $d//employee/name
where $d/@bldg > 95
order by $d/@bldg
return
  <EmpList>
    {$d/@bldg, $emp}
  </EmpList>

```

Esto devolvería lo siguiente:

```

<EmpList bldg="101">
  <name>
    John Doe
  </name>
  <name>
    Peter Pan
  </name>
</EmpList>

```

16.3.4 Inserción de documentos XML

La Inserción de documentos XML en una base de datos DB2 se puede realizar utilizando la sentencia INSERT SQL, o la utilidad IMPORT. XQuery no puede utilizarse para este fin ya que esto aun no ha sido definido en el standard.

Vamos a examinar el siguiente script, que puede ser ejecutado desde el comando DB2 en los shell de Windows o Linux usando la siguiente declaración:

```
db2 -tvf table_creation.txt
```

table_creation.txt

```
-- (1)
```

```
drop database mydb
;

-- (2)
create database mydb using codeset UTF-8 territory US
;

-- (3)
connect to mydb
;

-- (4)
create table items (
  id          int primary key not null,
  brandname   varchar(30),
  itemname    varchar(30),
  sku         int,
  srp         decimal(7,2),
  comments    xml
);

-- (5)
create table clients(
  id          int primary key not null,
  name        varchar(50),
  status      varchar(10),
  contact     xml
);

-- (6)
insert into clients values (77, 'John Smith', 'Gold',
  '<addr>111 Main St., Dallas, TX, 00112</addr>')
;

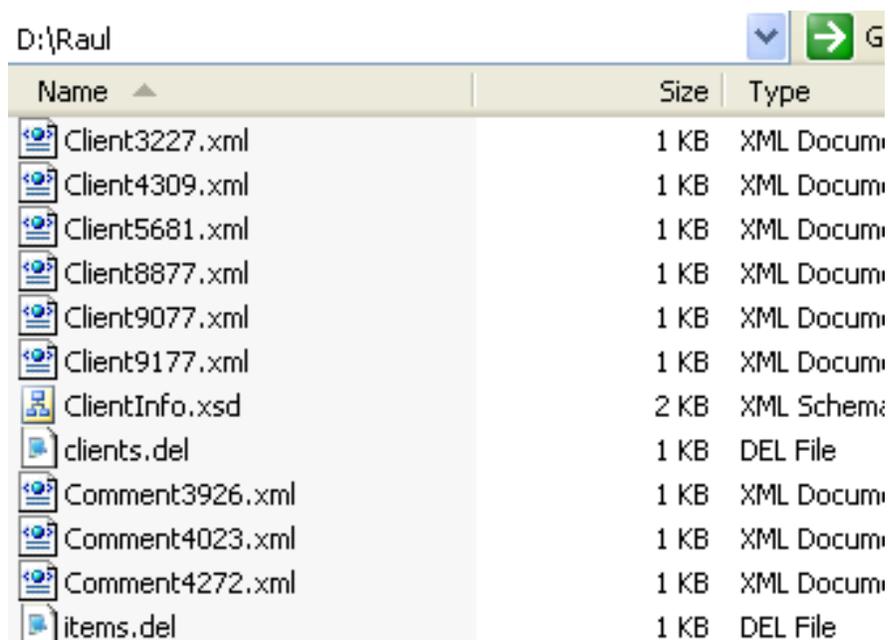
-- (7)
IMPORT FROM "D:\Raul\clients.del" of del xml from "D:\Raul" INSERT
INTO CLIENTS (ID, NAME, STATUS, CONTACT)
;

-- (8)
IMPORT FROM "D:\Raul\items.del" of del xml from "D:\Raul" INSERT
INTO ITEMS (ID, BRANDNAME, ITEMNAME, SKU, SRP, COMMENTS)
;
```

Tenga en cuenta que este archivo de secuencia de comandos y archivos relacionados son proporcionados en el archivo zip **expressc_book_quicklabs.zip** que acompaña a este libro. Siga a lo largo la descripción de cada línea del script.

1. Eliminar la base de datos "mydb". Esto se hace normalmente en los archivos script para realizar una limpieza antes de crear nuevos objetos de la base de datos. Si "mydb" no existía, recibirá un mensaje de error, pero es correcto.
2. Crear la base de datos "mydb" utilizando los códigos UTF-8. Se requiere una base de datos UNICODE para soportar pureXML, por lo que este paso es necesario para crear la base de datos como una base de datos UNICODE. Si esta usando la version 9.5, crear la base de datos como UNICODE es opcional.
3. Conectar a la base datos "mydb" recién creada. Esto es necesario para crear objetos dentro de la base de datos.
4. Crear la tabla "ítems". Tenga en cuenta que la ultima columna de la tabla (columna "comentarios") esta definida como una columna XML usando el nuevo tipo de dato XML.
5. Creamos la tabla "clientes". Tenga en cuenta que la ultima columna en la tabla (columna "contacto") también esta definida con el nuevo tipo de datos XML.
6. Usando esta sentencia SQL INSERT puede insertar un documento XML en una columna XML. En la sentencia INSERT usted puede pasar al documento XML como una cadena encerrada entre comillas simples.
7. Usando el comando IMPORT, puede insertar o importar varios documentos XML junto a los datos relacionales hacia la base de datos. En (7) esta importando el dato desde el archivo clients.del (un archivo ascii delimitado), y también indica donde se encuentra el dato XML referenciado por el archivo clients.del (para este ejemplo, en D:/Raul)

Veamos con mas detalle el archivo clients.del, pero primero, veamos el contenido del directorio D:/Raul (figura 16.10)



Name	Size	Type
Client3227.xml	1 KB	XML Document
Client4309.xml	1 KB	XML Document
Client5681.xml	1 KB	XML Document
Client8877.xml	1 KB	XML Document
Client9077.xml	1 KB	XML Document
Client9177.xml	1 KB	XML Document
ClientInfo.xsd	2 KB	XML Schema
clients.del	1 KB	DEL File
Comment3926.xml	1 KB	XML Document
Comment4023.xml	1 KB	XML Document
Comment4272.xml	1 KB	XML Document
items.del	1 KB	DEL File

Figura 16.10 - Contenido del directorio D:/Raul con documentos XML

Estos son los contenidos del archivo clients.del

clients.del

```
3227,Ella Kimpton,Gold,<XDS FIL='Client3227.xml' />,
8877,Chris Bontempo,Gold,<XDS FIL='Client8877.xml' />,
9077,Lisa Hansen,Silver,<XDS FIL='Client9077.xml' />
9177,Rita Gomez,Standard,<XDS FIL='Client9177.xml' />,
5681,Paula Lipenski,Standard,<XDS FIL='Client5681.xml' />,
4309,Tina Wang,Standard,<XDS FIL='Client4309.xml' />
```

En el archivo clients.del, “XDS FIL=” es usado para apuntar a un archivo documento XML.

La figura 16.11 muestra el Centro de Control luego de ejecutar el script de arriba.

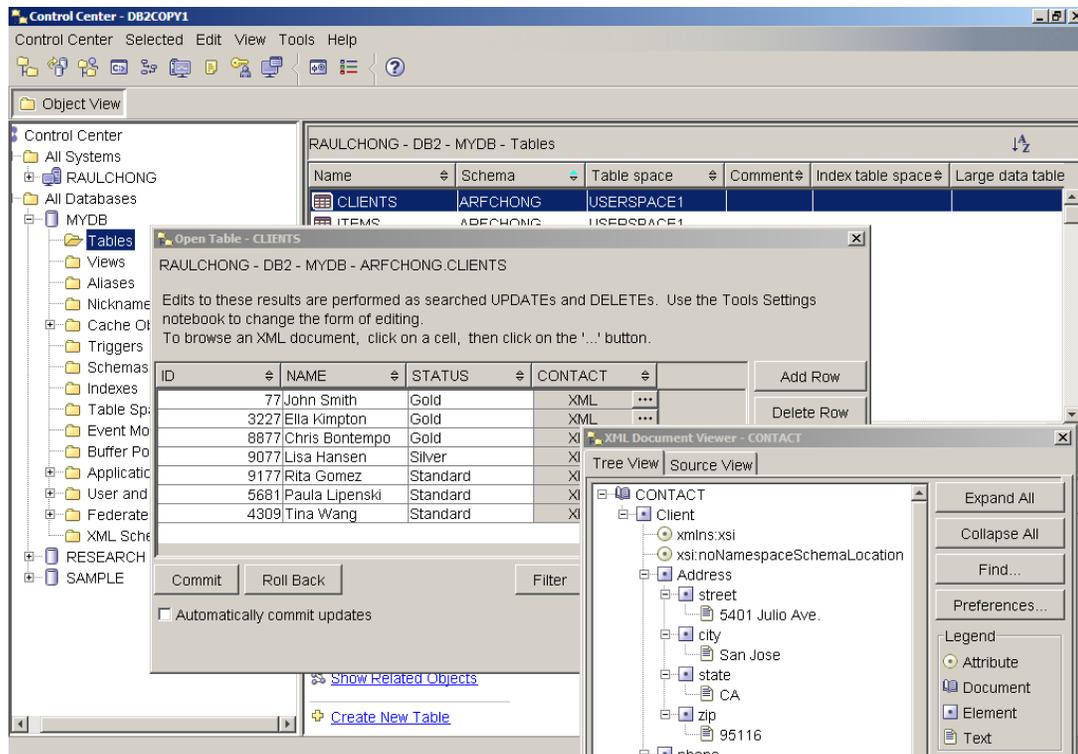


Figura 16.11 –El Centro de control después de ejecutar table_creation.txt

Tenga en cuenta que en la figura se muestra el contenido de la tabla CLIENTS. La última columna “Contact” es una columna XML. Al hacer clic en el botón con tres puntos, se abre

otra ventana que mostrara el contenido del documento XML. Esto se muestra en la esquina inferior derecha en la figura 16.11

16.3.5 Consulta de datos XML

Hay dos caminos para consultar datos XML en DB2:

- Usar SQL con extensión XML (SQL/XML)
- Usar XQuery

En ambos casos, DB2 sigue las normas internacionales.

Consulta de datos XML con SQL/XML

El uso de sentencias SQL permite trabajar con filas y columnas. Una sentencia SQL puede ser usada para trabajar con todo el documento XML; pero si se deseara obtener solo parte del documento XML, entonces se necesitaría usar SQL con extensión XML (SQL/XML).

La tabla 16.5 describe algunas de las funciones SQL/XML disponibles con el estándar SQL 2006.

Nombre de la función	Descripción
XMLPARSE	Parseo de caracter dato binario de objeto largo, produce valores XML
XMLSERIALIZE	Convierte un valor XML en carácter o dato binario de objeto largo.
XMLVALIDATE	Valida el valor XML contra un esquema XML y anota el tipo de valor XML
XMLEXISTS	Determina si una XQuery retorna un resultado (es decir, una secuencia de uno o mas ítems)
XMLQUERY	Ejecuta una XQuery y devuelve la secuencia resultado.
XMLTABLE	Ejecuta una XQuery devuelve el resultado secuencia como una tabla relacional(si es posible)
XMLCAST	Castea desde o hacia un tipo XML

Tabla 16.5 – Funciones SQL/XML

Los siguientes ejemplos pueden ser probados usando la base de datos “mydb” creada anteriormente.

Ejemplo 1

Este es un ejemplo de una consulta. Imagine que necesita localizar los nombres de todos los clientes que viven en un determinado código postal. La tabla “clients” almacena las direcciones de los clientes, incluyendo código postal en una columna XML. Usando XMLEXISTS, puede buscar en la columna XML el código postal deseado y luego limitar el resultado retornado como corresponda.

```
SELECT name FROM clients
WHERE xmlexists(
```

```
'$c/Client/Address[zip="95116"]'
passing clients.contact as "c"
)
```

La primera línea es una cláusula SQL que especifica que necesita recuperar la información en la columna “name” de la tabla “clients”

La cláusula WHERE invoca a la función XMLEXISTS, especificando la expresión XPath que induce a DB2 a navegar hasta el elemento “zip” y chequear el valor 95116.

La cláusula “\$c/Client/Address” indica la ruta dentro de la jerarquía del documento XML donde DB2 puede localizar el elemento “zip”. El signo dólar (\$) se utiliza para especificar una variable; por lo que “c” es una variable. Esta variable se define por esta línea: `passing clients.contact as "c"`. Aquí, “clients” es el nombre de la tabla y “contact” es el nombre de la columna con un dato de tipo XML. En otras palabras, estamos pasando el documento XML a la variable “c”.

DB2 inspecciona los datos XML contenidos en la columna “contact”, navega desde la raíz, nodo “Client”, al nodo “Address” y luego al nodo “zip” y finalmente determina si el cliente vive en el código postal buscado. La función XMLEXISTS evalúa por “verdadero (true)” y DB2 retorna el nombre del cliente asociado con esa fila.

En DB2 9.5, las consultas pueden simplificarse como sigue:

```
SELECT name FROM clients
WHERE xmlexists(
    '$CONTACT/Client/Address[zip="95116"]'
)
```

Una variable con el mismo nombre que la columna de tipo XML es creada automáticamente por DB2. En el ejemplo anterior, la variable CONTACT es creada automáticamente por DB2. Este nombre coincide con la columna CONTACT de tipo XML.

Ejemplo 2

Vamos a estudiar la forma de resolver el problema de como crear un informe con lista de direcciones e-mail de clientes de estado “Gold”. La siguiente consulta podría ejecutar:

```
SELECT xmlquery('$c/Client/email' passing contact as "c")
FROM clients
WHERE status = 'Gold'
```

La primera línea indica que queremos retornar la dirección de e-mail que es un elemento del documento XML (no una columna relacional). Como en el ejemplo anterior, “\$c” es una variable que contiene el documento XML. En este ejemplo usamos la función XMLQUERY que puede ser usada después de SELECT, mientras la función XMLEXISTS puede ser usada después de una cláusula WHERE.

Ejemplo 3

Puede haber situaciones en la que le gustaría presentar datos XML como tablas. Esto es posible con la función XMLTABLE como se muestra en el ejemplo debajo.

```
SELECT t.comment#, i.itemname, t.customerID, Message
FROM items i,
xmltable('$c/Comments/Comment' passing i.comments as "c"
columns Comment# integer path 'CommentID',
CustomerID integer path 'CustomerID',
Message varchar(100) path 'Message') AS t
```

La primera línea especifica las columnas que deben incluirse en el conjunto de resultados. Las columnas con el prefijo “t” están basadas en valores de elementos XML.

La segunda línea invoca la función XMLTABLE para especificar la columna XML que contiene el dato buscado (“i.comments”) y la ruta dentro de la columna de documentos XML donde los elementos de interés están localizados.

La cláusula “columns”, que abarca desde la línea 4 a la 6, identifican los elementos específicos XML que van a ser mapeados a la salida de la columna en el conjunto resultado SQL especificado en la línea 1. Parte de este mapeo implica especificar en el tipo de dato que los valores de los elementos XML serán convertidos. En este ejemplo todo dato XML es convertido en el tradicional tipo de datos SQL.

Ejemplo 4

Ahora vamos a estudiar un ejemplo sencillo donde usted incluye una expresión XQuery FLWOR dentro de una función XMLQUERY SQL/XML

```
SELECT name, xmlquery(
'for $e in $c/Client/email[1] return $e'
passing contact as "c"
)
FROM clients
WHERE status = 'Gold'
```

La primera línea especifica que los nombres de los clientes y la salida desde la función XMLQUERY estarán incluidos en el conjunto de resultados. La segunda línea indica el primer sub-elemento “email” del elemento “client” Será retornado. La tercera línea identifica la fuente del dato XML (columna “contact”). La cuarta línea nos dice que esta columna proviene de la tabla “clients”; y la quinta línea indica que solo los clientes “Gold” son de interés.

Ejemplo 5

Este ejemplo muestra nuevamente la función XMLQUERY que toma una expresión XQuery FLWOR, pero note que esta vez retornamos no solo XML, sino también HTML.

```
SELECT xmlquery('for $e in $c/Client/email[1]/text()
return <p>{$e}</p>'
passing contact as "c")
FROM clients
WHERE status = 'Gold'
```

La cláusula return de XQuery permite transformar la salida XML como necesite. Usando la función text() en la primer línea, indica que solo la representación del texto de la primera dirección de email de clientes calificados son de interés. La segunda línea especifica que esta información estará encerrada por las etiquetas de párrafo HTML.

Ejemplo 6

El siguiente ejemplo utiliza la función XMLEMENT para crear una serie de elementos ítems, cada uno contiene valores para sub-elementos ID, nombre de marca, y la unidad de mantenimiento de existencias (stock keeping unit (SKU)) obtenidos de las correspondientes columnas en la tabla "ítems". Básicamente puede usar la función XMLEMENT cuando desee convertir dato relacional a dato XML.

```
SELECT
  xmlelement (name "item", itemname),
  xmlelement (name "id", id),
  xmlelement (name "brand", brandname),
  xmlelement (name "sku", sku)
FROM items
WHERE srp < 100
```

La consulta anterior provee una salida como esta:

```
<item>
  <id>4272</id>
  <brand>Classy</brand>
  <sku>981140</sku>
</item>
...
<item>
  <id>1193</id>
  <brand>Natural</brand>
  <sku>557813</sku>
</item>
```

Consulta de datos XML con XQuery

En la sección anterior, vimos como consultar datos XML utilizando SQL con extensiones XML. SQL fue siempre el método de consulta primario, y XPath estaba embebido dentro de SQL. En esta sección, estudiaremos como consultar datos XML utilizando XQuery. Esta vez, XQuery será el método primario de consulta, y en algunos casos, usaremos SQL embebido dentro de XQuery (usando la función "db2-fn:sqlquery"). Cuando usamos XQuery, invocaremos algunas funciones, y utilizaremos también la expresión FLWOR.

Ejemplo 1

Una simple XQuery para retornar dato del contacto cliente.

```
xquery db2-fn:xmlcolumn('CLIENTS.CONTACT')
```

Siempre prefije cualquier expresión XQuery con el comando “xquery” así DB2 sabrá que tiene que utilizar XQuery, de lo contrario DB2 asumirá que esta tratando de ejecutar una expresión SQL. La función **db2-fn:xmlcolumn** es una función que recupera documentos XML de la columna especificada por parámetro. Es equivalente a la siguiente sentencia SQL, que recupera el contenido de la columna completa:

```
SELECT contact FROM clients
```

Ejemplo 2

En este ejemplo, usamos la expresión FLWOR para recuperar el dato del fax del cliente.

```
xquery
  for $y in db2-fn:xmlcolumn('CLIENTS.CONTACT')/Client/fax
  return $y
```

La primera línea invoca a el parseador XQuery. La segunda línea instruye a DB2 a iterar a través del sub-elemento fax contenido en la columna CLIENTS.CONTACT. Cada elemento fax esta rodeado por la variable \$y. La tercera línea indica que por cada iteración, el valor “\$y” es retornado.

La salida de esta consulta es similar a lo siguiente (puede incluir el nombre del espacio, o namespace, por defecto, pero no lo mostramos, de lo contrario esta salida seria más difícil de leer, ya que podría abarcar varias líneas):

```
<fax>4081112222</fax>
<fax>5559998888</fax>
```

Ejemplo 3

El siguiente ejemplo consulta datos XML y retorna el resultado como HTML.

```
xquery
  <ul> {
    for $y in db2-fn:xmlcolumn('CLIENTS.CONTACT')/Client/Address
    order by $y/zip
    return <li>{$y}</li>
  }
</ul>
```

La salida HTML se ve así:

```
<ul>
<li>
<address>
  <street>9407 Los Gatos Blvd.</street>
  <city>Los Gatos</city>
  <state>ca</state>
  <zip>95302</zip>
</address>
</li>
<address>
<street>4209 El Camino Real</street>
  <city>Mountain View</city>
```

```

    <state>CA</state>
    <zip>95302</zip>
</address>
</li>
...
</ul>

```

Ejemplo 4

El siguiente ejemplo muestra como embeder SQL dentro de XQuery usando la función db2-fn:sqlquery. La función db2-fn:sqlquery ejecuta una consulta SQL y retorna solo el dato XML seleccionado. La consulta SQL que paso a db2-fn:sqlquery solo debe retornar datos de tipo XML. Este dato XML puede ser procesado mas aun por XQuery

```

xquery
  for $y in
    db2-fn:sqlquery(
      'select comments from items where srp > 100'
    )/Comments/Comment
  where $y/ResponseRequested='Yes'
  return (
    <action>
      {
        $y/ProductID
        $y/CustomerID
        $y/Message
      }
    </action>
  )

```

En el ejemplo, la consulta SQL filtra líneas basadas en la condición que la columna “srp” tiene un valor mayor a 100. De estas líneas filtradas, seleccionara la columna “comments”, que es la columna de tipo XML. Luego XQuery (o XPath) es aplicado para ir a los sub-elementos.

Nota: DB2 no distingue mayúsculas de minúsculas y trata a todos los nombres de tablas y columnas en mayúsculas mientras XQuery si distingue estas. Las funciones que hemos discutido hasta ahora, son funciones de XQuery, así que todo nombre de tablas y columnas deben ser pasados en mayúsculas. Si se utilizaran minúsculas puede resultar en un error de un objeto indefinido.

16.3.6 Unir con SQL/XML

Esta sección describe como llevar a cabo operaciones JOIN entre dos columnas XML de diferentes tablas, o entre una columna XML y una columna relacional. Suponga que tiene creada dos tablas con estas sentencias:

```

CREATE TABLE dept (unitID CHAR(8), deptdoc XML)

CREATE TABLE unit (unitID CHAR(8) primary key not null,
                   name CHAR(20),
                   manager VARCHAR(20),
                   ...
                   )

```

Puede realizar una operación JOIN de dos maneras:

Método 1:

```
SELECT u.unitID
FROM dept d, unit u
WHERE XMLEXISTS (
  '$e//employee[name = $m]'
  passing d.deptdoc as "e", u.manager as "m")
```

En la línea 3 de esta sentencia se muestra que la operación JOIN se produce entre el elemento "name" que es sub-elemento de la columna XML "deptdoc" en la tabla "dept", y la columna relacional "manager" en la tabla "unit".

Método 2:

```
SELECT u.unitID
FROM dept d, unit u
WHERE u.manager = XMLCAST(
  XMLQUERY('$e//employee/name '
  passing d.deptdoc as "e")
  AS char(20))
```

En este modo alternativo, la columna relacional se encuentra en el lado izquierdo del JOIN. Si la columna relacional esta en el lado izquierdo de la ecuación, un índice relacional será utilizado en vez de un índice XML.

16.3.7 Unir con XQuery

Suponga que las siguientes tablas han sido creadas:

```
CREATE TABLE dept(unitID CHAR(8), deptdoc XML)
CREATE TABLE project(projectDoc XML)
```

Si usamos SQL/XML, la consulta se vería como sigue:

```
SELECT XMLQUERY (
  '$d/dept/employee' passing d.deptdoc as "d")
FROM dept d, project p
WHERE XMLEXISTS (
  '$e/dept[@deptID=$p/project/deptID]'
  passing d.deptdoc as "e", p.projectDoc as "p")
```

La consulta equivalente usando XQuery sería:

```
xquery
for $dept in db2-fn:xmlcolumn("DEPT.DEPTDOC")/dept
for $proj in db2-fn:xmlcolumn("PROJECT.PROJECTDOC")/project
where $dept/@deptID = $proj/deptID
return $dept/employee
```

Este segundo método es más fácil de interpretar – la variable “\$dept” tiene el documento XML de la columna XML “deptdoc” en la tabla “dept”. La variable “\$proj” tiene el documento XML de la columna XML “projectdoc” en la tabla “project”. Luego la línea 4 lleva a cabo la operación JOIN entre un atributo del primer documento XML y un elemento del segundo documento XML.

16.3.8 Operaciones de actualización y eliminación

Operaciones de actualización y eliminación en datos XML se pueden hacer de dos maneras:

- ▶ Usando sentencias SQL UPDATE y DELETE
- ▶ Usando la expresión TRANSFORM

En la primera forma, usando sentencias SQL UPDATE y DELETE, la actualización o la eliminación se produce a nivel documento; es decir, todo el documento XML se sustituye con la actualización. Por ejemplo, si en el ejemplo siguiente todo lo que nos gustaría cambiar es el elemento <state>, en realidad todo el documento XML se sustituye.

```
UPDATE clients SET contact=(
  xmlparse(document
    '<Client>
      <address>
        <street>5401 Julio ave.</street>
        <city>San Jose</city>
        <state>CA</state>
        <zip>95116</zip>
      </address>
      <phone>
        <work>4084633000</work>
        <home>4081111111</home>
        <cell>4082222222</cell>
      </phone>
      <fax>4087776666</fax>
      <email>newemail@someplace.com</email>
    </Client>')
  )
WHERE id = 3227
```

Para la segunda forma, puede actualizar sub-documentos usando la expresión TRANSFORM, la cual es un poco más eficiente. Esto le permite reemplazar, insertar, eliminar o renombrar nodos en un documento XML. También puede cambiar el valor de un nodo sin sustituir el nodo en sí, normalmente para cambiar un elemento o valor de un atributo, que es un tipo de actualización muy común. Esto es nuevo con DB2 9.5.

La expresión TRANSFORM es parte del lenguaje XQuery, puede usarla en cualquier lugar donde usa XQuery, por ejemplo en una expresión FLWOR o en la función XMLQUERY en una sentencia SQL/XML. El uso más típico es en sentencias SQL UPDATE para modificar un documento XML es una columna XML.

Esta es la sintaxis de la expresión TRANSFORM:

```
>>-transform--| copy clause |--| modify clause |--| return clause
|-><
```

copy clause

```

      .-,-----
      v                                     |
|--copy----$VariableName---:---CopySourceExpression-+-----|
```

modify clause

```
|--modify--ModifyExpression-----|
```

return clause

```
|--return--ReturnExpression-----|
```

La cláusula `copy` se usa para asignar a una variable el documento XML que quiere procesar. En la cláusula `modify`, puede invocar una expresión `insert`, `delete`, `rename` o `replace`. Estas expresiones le permiten actualizar el documento XML. Por ejemplo, si quiere añadir nuevos nodos al documento, debería usar la expresión `insert`, para eliminar nodos desde un documento XML, use la expresión `delete`, para renombrar un elemento o atributo en el documento XML, use la expresión `rename`, y para reemplazar un nodo existente con un nuevo nodo o secuencia de nodos, usa la expresión `replace`. El valor reemplazado de la expresión puede ser usado solamente para cambiar el valor de un elemento o atributo. La cláusula `return` retorna el resultado de una expresión `transform`.

A continuación hay un ejemplo de una sentencia `UPDATE` usando la expresión `TRANSFORM`.

```
(1)-- UPDATE customers
(2)-- SET contactinfo = xmlquery( 'declare default element namespace
(3)--                               "http://posample.org";
(4)--   transform
(5)--   copy $newinfo := $c
(6)--       modify do insert <email2>my2email.gm.com</email2>
(7)--               as last into $newinfo/customerinfo
(8)--   return $newinfo' passing contactinfo as "c")
(9)-- WHERE id = 100
```

En el ejemplo anterior, las líneas (1), (2), y (9) son parte de la sintaxis SQL `UPDATE`. En la línea (2) la función `XMLQUERY` es invocada, y llama a la expresión `transform` en la línea (4). El bloque de la expresión `transform` va desde la línea (4) a la línea (8), y es usada para insertar un nuevo nodo dentro del documento XML que contiene el elemento `email2`.

Tenga en cuenta que la actualización de los elementos en un documento XML a través de una vista no está soportado.

Eliminar documentos XML completos desde tablas es tan simple como cuando se utiliza la sentencia SELECT en SQL/XML. Use la sentencia SQL DELETE y especifique cualquier predicado WHERE necesario.

16.3.9 Indexación XML

En un documento XML, la indexación puede ser creada para elementos, atributos, o para valores (nodos de texto). Abajo hay algunos ejemplos. Suponga que la siguiente tabla está creada.

```
CREATE TABLE customer(info XML)
```

Y suponga que este es uno de los documentos XML almacenados:

```
<customerinfo Cid="1004">
  <name>Matt Foreman</name>
  <addr country="Canada">
    <street>1596 Baseline</street>
    <city>Toronto</city>
    <state>Ontario</state>
    <pcode>M3Z-5H9</pcode>
  </addr>
  <phone type="work">905-555-4789</phone>
  <phone type="home">416-555-3376</phone>
  <assistant>
    <name>Peter Smith</name>
    <phone type="home">416-555-3426</phone>
  </assistant>
</customerinfo>
```

1) Esta sentencia crea un índice en el atributo "Cid"

```
CREATE UNIQUE INDEX idx1 ON customer(info)
GENERATE KEY USING
xmlpattern '/customerinfo/@Cid'
AS sql DOUBLE
```

2) Esta sentencia crea un índice en el elemento "name"

```
CREATE INDEX idx2 ON customer(info)
GENERATE KEY USING
xmlpattern '/customerinfo/name'
AS sql VARCHAR(40)
```

3) Esta sentencia crea un índice en todos los elementos "name"

```
CREATE INDEX idx3 ON customer(info)
```

```
GENERATE KEY USING  
xmlpattern '//name'  
AS sql VARCHAR(40);
```

4) Esta sentencia crea un índice en todos los nodos de texto (todos los valores). Esto no es recomendable, ya que sería demasiado costoso mantener el índice en una operación de actualización, eliminación o inserción, y el índice sería demasiado grande.

```
CREATE INDEX idx4 ON customer(info)  
GENERATE KEY USING  
xmlpattern '//text()'  
AS sql VARCHAR(40);
```

QuickLab #14 - SQL/XML y XQuery

Objetivo

Hasta ahora ya ha visto varios ejemplos de sintaxis SQL/XML y XQuery y se ha introducido al DB2 Command Editor e IBM Data Studio. En este laboratorio, pondrá a prueba su conocimiento sobre SQL/XML y XQuery mientras adquiere experiencia con estas herramientas. Usaremos la base de datos “mydb” creada usando el archivo script **table_creation.txt** que fue explicado anteriormente en este capítulo.

Procedimiento

1. Crear la base de datos “mydb” y cargar el dato XML, como fue examinado anteriormente en este capítulo.
2. Usar ya sea el Command Editor o IBM Data Studio:
 - a) Recuperar todos los comentarios del documento XML desde la tabla ITEMS de dos maneras, pero utilizando solo XQuery.
 - b) ¿Porque la emisión de esta sentencia SQL no retorna exactamente la misma salida?:

```
SELECT comments FROM items
```
 - c) Recuperar el ID y BRANDNAME de los archivos cuyos documentos XML tienen un elemento ResponseRequested con el valor “NO”

SOLUCION

2a)

```
xquery db2-fn:xmlcolumn('ITEMS.COMMENTS')
xquery db2-fn:sqlquery("select comments from items")
```

2b)

La salida es diferente porque SQL retorna valores NULL cuando un valor no esta presente, mientras XQuery no retorna nada.

2c)

```
SELECT id, brandname FROM items WHERE
  XMLEXISTS(' $c/Comments/Comment[ResponseRequested="No" ] '
  passing ITEMS.COMMENTS as "c
```


17

Capítulo 17 –Desarrollo con Java, PHP, y Ruby

Este capítulo trata de las aplicaciones básicas desarrolladas en Java, PHP y Ruby on Rails usando un servidor DB2. El propósito de este capítulo no es enseñar este lenguaje, sino proveer información pertinente al uso de ellos con DB2.

Nota:

Para mas información sobre este tema, observe este video:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4402>

17.1 Desarrollo de aplicaciones en Java

El driver de IBM DB2 para JDBC (también conocido como driver JCC) ha sido optimizado para trabajar en todos los servidores DB2 en todas las plataformas. El archivo jar db2jcc.jar (com.ibm.db2.jcc) incluye drivers tipo 2 y tipo 4. El archivo db2jcc.jar está incluido con cualquier cliente de DB2 o puede ser obtenido (el driver IBM DB2 para JDBC y SQLJ) desde el sitio web de DB2 Express-C (ibm.com/db2/express).

17.1.1 Driver JDBC tipo 2

El driver JDBC tipo 2 requiere que un cliente DB2 sea instalado donde una aplicación JDBC esta corriendo. La figura 17.1 ilustra una aplicación JDBC usando el driver tipo 2.

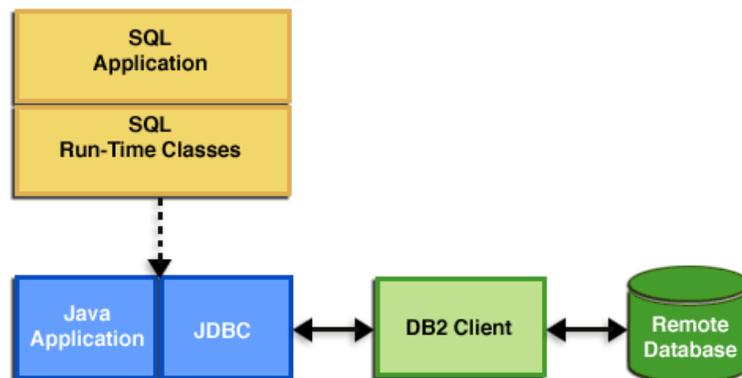


Figure 17.1 – El driver JDBC tipo 2.

La figura 17.2 proporciona una lista que muestra como establecer la conexión utilizando el driver JDBC tipo 2. Nótese que la URL no incluye nombre de Host o información del puerto ya que éstos son obtenidos del cliente DB2.

```

...
public static final String DB_URL = "jdbc:db2:sample";
Properties connectProperties = new Properties();
connectProperties.put("user", "db2admin");
connectProperties.put("password", "ibmdb2");
Connection connection = null
try
{
    Class.forName("com.ibm.db2.jcc.DB2Driver").newInstance();
    connection = DriverManager.getConnection(url, connectProperties)
}
catch (Exception e)
throw e;
}
...

```

Figure 17.2 – Estableciendo una conexión usando el driver JDBC tipo 2.

17.1.2 Driver JDBC tipo 4

El driver JDBC tipo 4 no necesita un cliente DB2 para conectar a un servidor DB2. La figura 17.3 ilustra una aplicación JDBC usando el driver tipo 4.

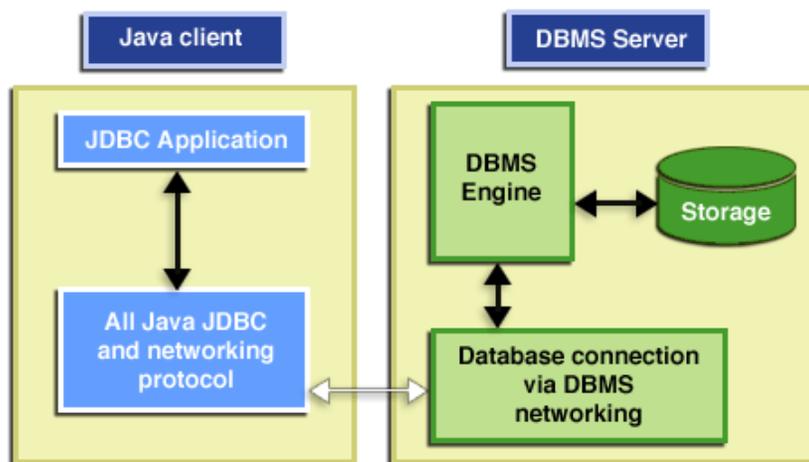


Figure 17.3 – El driver JDBC tipo 4

La figura 17.4 proporciona una lista que muestra como establecer la conexión utilizando el driver JDBC tipo 4. Nótese que la URL incluye nombre de Host o información del puerto.

```

...
public static final String DB_URL = "jdbc:db2://server1:50000/sample";
Properties connectProperties = new Properties();
connectProperties.put("user", "db2admin");
connectProperties.put("password", "ibmdb2");
Connection connection = null
try
{
    Class.forName("com.ibm.db2.jcc.DB2Driver").newInstance();
    connection = DriverManager.getConnection(url,connectProperties)
}
catch (Exception e)
    throw e;
}
...

```

Figure 17.4 – Estableciendo una conexión usando el driver JDBC tipo 4.

17.2 Desarrollo de aplicaciones en PHP

PHP (PHP Preprocesador hipertexto) es un lenguaje de fuente abierta, plataforma de escritura independiente designado para desarrollo de aplicaciones Web. Es uno de los lenguajes web mas extensamente desplegado en el mundo actualmente. La popularidad de PHP esta bastada en las siguientes características del lenguaje:

- Ciclos de desarrollo rápidos e iterativos con una curva baja de aprendizaje.
- Robusto, de alto rendimiento y escalable.
- Estable y seguro.
- Una alternativa para J2EE™ y .NET en la Web.
- Fácil integración en entornos heterogéneos / sistemas
- Demostrado a través del despliegue generalizado.
- Vibrante comunidad bien establecida.

PHP es parte de la pila LAMP (Linux, Apache HTTP Server, MySQL, **PHP** / Perl / Python). Esta es una pila de código abierto, a veces disponible en ISPs por cuotas mensuales razonables.

17.2.1 Opciones de conexión de DB2 para PHP

IBM soporta acceso a la base de datos DB2 desde aplicaciones PHP a través de dos extensiones.

ibm_db2:

La extensión `ibm_db2` ofrece un procedimiento de interfaz de programación de aplicaciones para crear, leer, actualizar y escribir operaciones de base de datos además de amplio acceso a la meta data de la base de datos. Puede ser compilado para trabajar con PHP 4 o PHP 5. La extensión esta disponible en el repositor PECL bajo la licencia Apache 2.0. Fue desarrollado y es soportado por IBM. Tiene soporte completo para los procedimientos almacenados y LOBs, es rápido ya que ha sido optimizado para DB2.

PDO_ODBC:

PDO_ODBC es un driver para la extensión del Data Object de PHP (PDO) y ofrece acceso a bases de datos DB2 a través de una de una norma orientada a objetos de interfaz de base de datos introducida en PHP 5.1. Puede ser compilado directamente contra librerías de DB2. Proporciona un estándar de interfaz de acceso a datos para PHP. Es rápido, ligero de peso, y orientado a objetos. La extensión PDO_ODBC utiliza las bibliotecas de DB2 para el acceso nativo, y se ha construido en PHP 5,1. Para obtener más información, consulte estos sitios:

- <http://pecl.php.net/package/pdo>
- http://pecl.php.net/package/PDO_ODBC

Conectar a una base de datos DB2 no catalogada.

El listado 17.1 muestra como conectar a una base de datos DB2 usando las dos extensiones previamente descritas.

```
$host = 'localhost';
$port = 50000;
$DSN = "DRIVER={IBM DB2 ODBC DRIVER}; PORT=$port;
        HOSTNAME=$host; DATABASE=$database; PROTOCOL=TCPIP;
        USER=$user; PWD=$password";

-- If using the ibm_db2 extension --
$suconn = db2_connect($DSN, null, null);

-- If using the PDO_ODBC extension --
try {
    $suconn = new PDO("odbc:$DSN", null, null);
}
catch (PDOException $e) { print $e->errormsg(); }
```

Listado 17.1 – Conectar a una base de datos DB2 no catalogada

El listado 17.2 proporciona un ejemplo de una simple aplicación PHP usando la extensión `ibm_db2`

```
<?php
$sql = "SELECT name, breed FROM ANIMALS WHERE weight < ?";
$conn = db2_connect($database, $user, $password);
$stmt = db2_prepare($conn, $sql);
$res = db2_execute($stmt, array(10));
while ($row = db2_fetch_assoc($stmt)) {
    print "{$row['NAME']} is a {$row['BREED']}\n";
}
?>
```

Listado 17.2 – Una simple aplicación PHP usando extensión `ibm_db2`**Configuración de PHP para `ibm_db2`**

En Linux o UNIX es posible que tenga que modificar el archivo `php.ini` como sigue:

```
extension=ibm_db2.so  
ibm_db2.instance_name=<instance name>
```

En Windows, modificar el archivo php.ini como sigue:

```
extension=php_ibm_db2.dll
```

Si lo prefiere, puede descargar e instalar el Zend Core para la suite de aplicaciones de IBM tal como se describe en la siguiente sección y no tiene que preocuparse por problemas de configuración.

17.2.2 Zend Core para IBM

Zend Core es un paquete de desarrollo en PHP del tipo “Out-of-the-box” y de entorno de producción para empresas de aplicaciones web críticas. Ofrece fiabilidad, productividad y flexibilidad necesarias para ejecutar las aplicaciones PHP. Se puede descargar y usar gratuitamente desde:

<http://ibm.com/software/data/info/zendcore>

Zend Core para IBM instala DB2 y clientes IDS, una opción de Apache HTTP Server, PHP 5, y extensiones populares de PHP tal como `ibm_db2`, y `PDO_INFORMIX`. Zend Core para IBM puede instalar opcionalmente DB2 Express-C Server, IBM Cloudscape™ Server, el manual completo de PHP y aplicaciones de muestra para DB2. Viene con gran facilidad de uso y configuración de entorno PHP, como se muestra en la figura 17.5, 17.6 y 17.7.

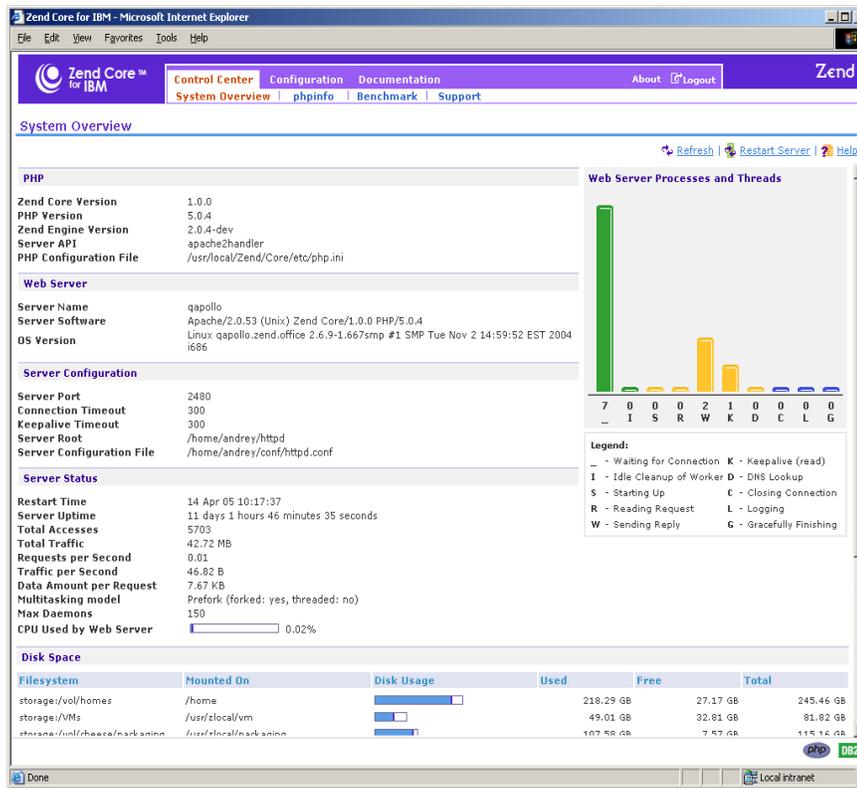


Figura 17.5 - Zend Core interfaz de gestión y control

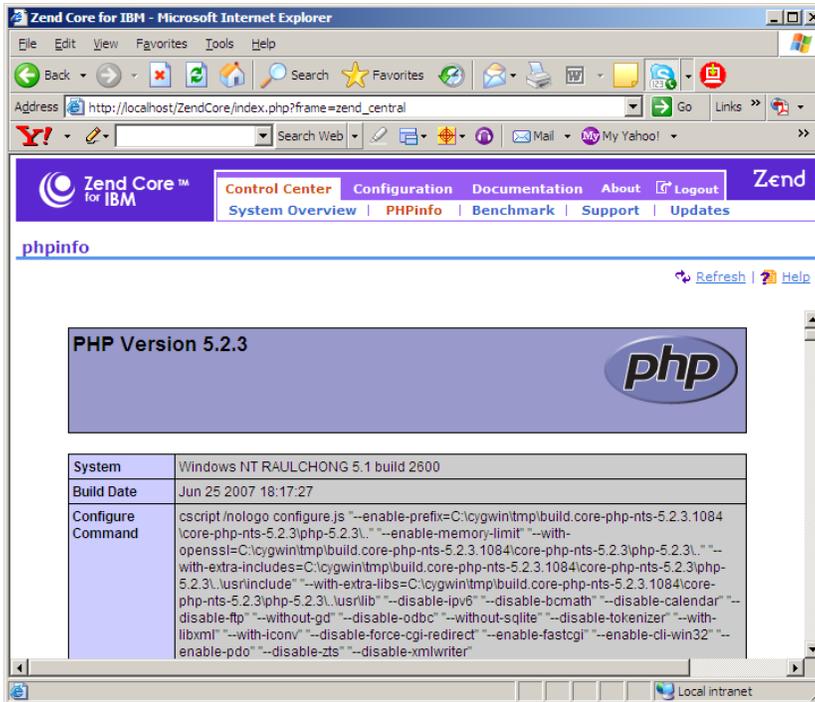


Figura 17.6 - Zend Core interfaz de configuración de PHP

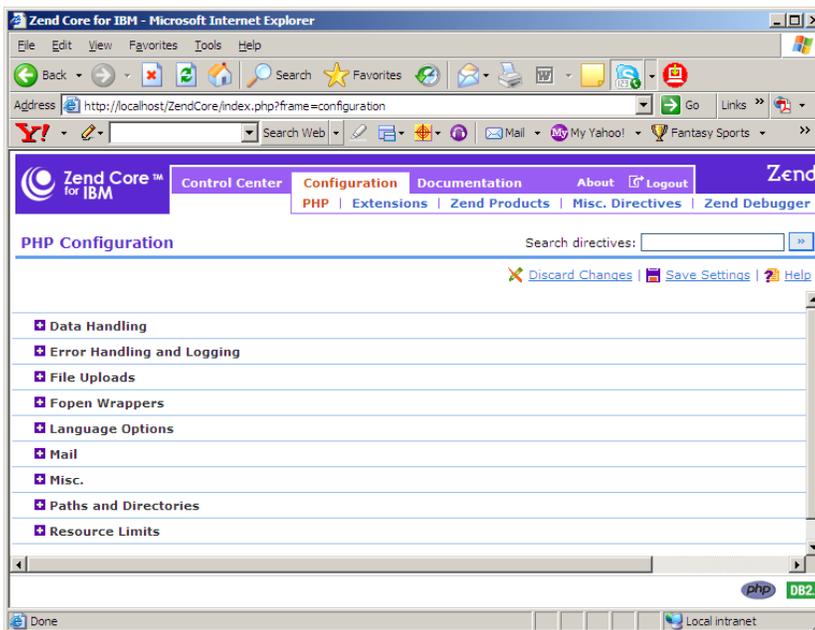


Figura 17.7 - Zend Core, interfaz de configuración de PHP

17.3 Desarrollo de aplicaciones en Ruby on Rails

Ruby es un lenguaje orientado a objetos, dinámico y con scripts que funcionan en varias plataformas. Facilita el rápido desarrollo e incluye una rica librería. Ruby es un sencillo y divertido lenguaje de programación orientado a objetos, inventado por Yukihiro Matsumoto ("Matz") en 1995.

Rails es un completo marco (Framework) para la aplicaciones escritas en Ruby que utilizan a una base de datos. Implementa la arquitectura modelo-vista-control (MVC). Es increíblemente productivo y fácil de usar. Rails es uno de los frameworks para la Web que estan emergiendo rapidamente para el desarrollo en la web y fue inventado por David Heinemeier Hansson en el 2004.

17.3.1 Startup Toolkit para DB2 on Rails

IBM reconoce la importancia de Ruby on Rails en la comunidad de desarrollo, por lo tanto ha creado un paquete llamado **Startup Toolkit for DB2 on Rails**. Se trata de un instalador integrado que crea un completo entorno de desarrollo DB2 Ruby on Rails. Puede ser descargado y utilizado gratuitamente desde el sitio Web IBM alphaWorks: <http://www.alphaworks.ibm.com/tech/db2onrails>.

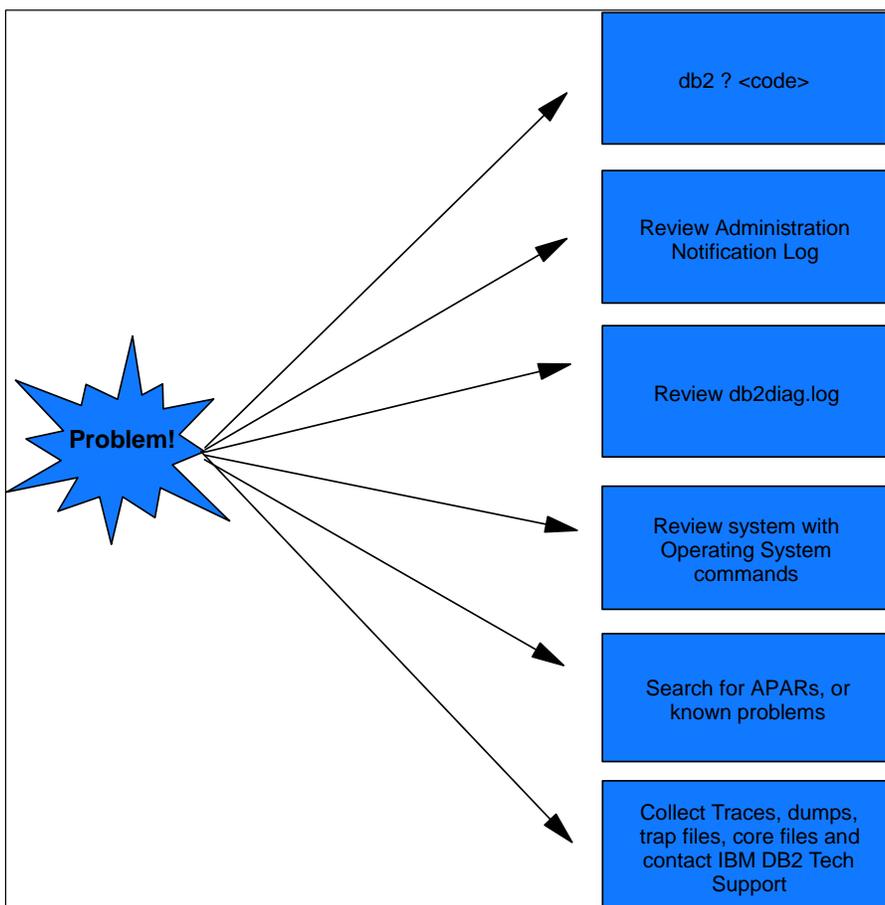
El paquete **Startup Toolkit for DB2 on Rails**:

- ▶ Incluye un instalador integrado
- ▶ Ayuda a la fácil instalación y configuración de Ruby and Rails
- ▶ Instala DB2 Express – C 9 y herramientas.
- ▶ Incluye un desarrollado driver IBM DB2 Ruby y Adaptador DB2 Rails.
- ▶ Incluye varios demos y tutoriales.

A

Apéndice A – Localización de fallas

Este apéndice discute como localizar y gestionar problemas que pudieran ser encontrados cuando se trabaja con DB2. La figura A 1 provee un breve panorama de las acciones a tomar si un problema surgiera.



Nota:

Para más información al respecto, vea este video:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4462>

A.1 Obteniendo más información acerca de los códigos de error.

Para obtener más información acerca de un código de error recibido, escribir el código antecedido por un signo de interrogación en el Editor de Comandos y dar clic en el botón Execute. Esto es mostrado en la Figura A.2

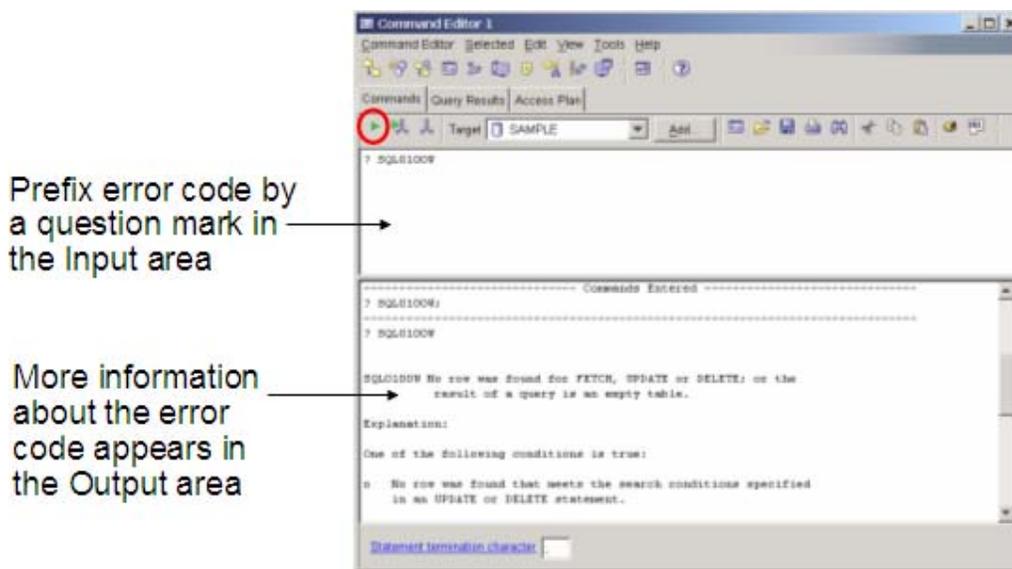


Figura A.2 – Encontrando más información acerca de los códigos de error de DB2

El signo de interrogación (?) invoca el comando help de DB2. Los siguientes ejemplos muestran como invocar este comando si recibe el código de error SQL "-104". Todos los ejemplos son equivalentes:

```
db2 ? SQL0104N
db2 ? SQL104N
db2 ? SQL-0104
db2 ? SQL-104
db2 ? SQL-104N
```

A.2 SQLCODE y SQLSTATE

Un SQLCODE es un código recibido después de que cualquier sentencia SQL es ejecutada. Los significados de los valores son resumidos abajo.

SQLCODE = 0, el comando fue exitoso
SQLCODE > 0, el comando fue exitoso pero regresó una advertencia
SQLCODE < 0, el comando no fue exitoso y regresó un error.

El SQLSTATE es una cadena de 5 caracteres conforme al estándar ISO/ANSI SQL92. Los primeros dos caracteres son conocidos como la clase de código SQLSTATE.

Una clase de código 00 significa que el comando fue exitoso.
Una clase de código 01 implica una advertencia.
Una clase de código 02 implica una condición no encontrada.
Todas las demás clases son considerados errores.

A.3. Bitácora de notificaciones de administración

La bitácora de notificaciones de administración (DB2 administration notification log) provee información de diagnóstico acerca de errores en el punto de falla. En las plataformas Linux/UNIX, es un archivo de texto llamado <nombre_instancia>.nfy (por ejemplo db2inst.nfy). En Windows, todos los mensajes de notificación son escritos en el Windows Event Log.

El parámetro de configuración notifylevel permite a los administradores especificar el nivel de información a ser grabado:

- 0 – Los mensajes no son capturados (no recomendado)
- 1 – Errores fatales o irrecuperables
- 2 – Acción inmediata requerida
- 3 – Información importante, no se requiere acción inmediata (por defecto)
- 4 – Mensajes informativos.

A.4 db2diag.log

Provee información más detallada que la bitácora de notificaciones de administración. Es normalmente utilizado solo por el soporte técnico de IBM DB2 o por DBAs experimentados. La información en el db2diag.log incluye:

- El código de localización reportando un error.
- Los identificadores de aplicación que permiten verificar las entradas pertenecientes a una aplicación en el archivo db2diag.log de los servidores y clientes
- Un mensaje de diagnóstico (comenzando con “DIA”) explicando la razón del error.
- Cualquier dato disponible de apoyo, como las estructuras de datos SQLCA y apuntadores a la localización de cualquier volcado extra o archivos de trampa.

En Windows, el archivo db2diag.log es localizado por defecto en el directorio

```
C:\Archivos de programa\IBM\sqlllib\\db2diag.log
```

En Linux/UNIX, este archivo es localizado por defecto en el directorio.

```
/home/<instance_owner>/sqlllib/db2dump/db2diag.log
```

La cantidad de información de diagnóstico es determinada por el parámetro de configuración DIAGLEVEL. El rango es de 0 a 4, donde 0 es el que menos texto produce y 4 el que más. El nivel por defecto es 3.

A.5 Trazas CLI

Para las aplicaciones CLI y Java, se puede activar la utilidad de trazas CLI para el descubrimiento de fallas en la aplicación. Esto puede ser hecho haciendo cambios en el archivo db2cli.ini en el servidor donde la aplicación esté corriendo. Típicamente las entradas en el archivo db2cli.ini son como las mostradas a continuación.

```
[common]
trace=0
tracerefreshinterval=300
tracepathname=/path/to/writeable/directory
traceflush=1
```

El nivel detallado de traza (db2trc) es también disponible, pero este es típicamente utilizado solo por el soporte técnico de DB2.

A.6 Defectos y arreglos de DB2

Algunas veces un problema encontrado puede ser causado por un defecto en DB2. IBM regularmente proporciona paquetes de arreglo (fix packs) los cuales contienen código para corregir esos defectos (APARs). La documentación de los fix packs contienen una lista de los ajustes contenidos en el. Cuando se desarrollan nuevas aplicaciones, se recomienda siempre utilizar el último fix pack para beneficiarse de los últimos ajustes. Para ver la versión actual y el nivel del fix pack: desde el Centro de Control, seleccionar la opción About del menu Help; de la Ventana de Comandos, teclear db2level. Note que los fix packs y el soporte técnico de IBM DB2 son solo ofrecidos en la versión DB2 Express-C si se compra una licencia de 12 meses.

Recursos

Web sites:

1. Página Web de DB2 Express-C:
www.ibm.com/db2/express
Use esta pagina para bajar la imagen del servidor de datos DB2 Express-C, los clientes de DB2, los drivers de DB2, manuales, acceso al blog del equipo DB2 Express-C, etc.
2. Foro de DB2 Express-C:
www.ibm.com/developerworks/forums/dw_forum.jsp?forum=805&cat=19
Use este foro para publicar preguntas técnicas cuando usted no puede encontrar las respuestas en los manuales.
3. DB2 Information Center
<http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp>
El centro de información proporciona los manuales de DB2 en línea. Contiene la información mas reciente.
4. developerWorks
<http://www-128.ibm.com/developerworks/db2>
Este sitio Web es un recurso excelente para desarrolladores y DBAs, dando acceso a artículos recientes, tutoriales, etc. de manera gratuita.
5. alphaWorks
<http://www.alphaworks.ibm.com/>
Este sitio Web proporciona acceso a tecnologías emergentes de IBM. Es un lugar donde uno puede encontrar los últimos avances en tecnología de los centros de investigación de IBM.
6. planetDB2
www.planetDB2.com
Este es un agregador de blogs de muchos contribuidores de blogs sobre DB2.
7. DB2 Technical Support
Si compró la licencia de subscripción de 12 meses de DB2 Express-C, usted puede bajar los fixpacks desde este sitio Web.
http://www.ibm.com/software/data/db2/support/db2_9/

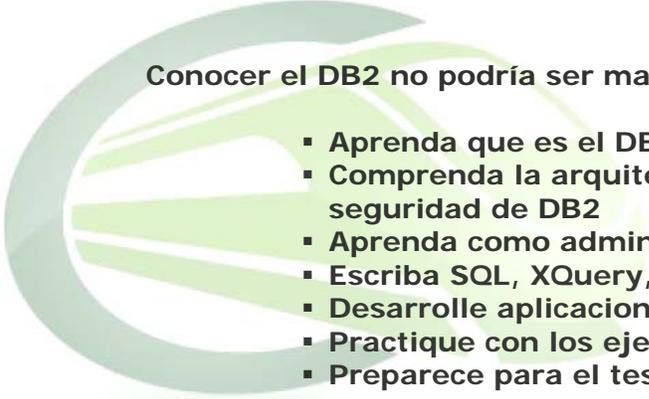
Libros

1. Redbook: DB2 Express-C: The Developer Handbook for XML, PHP, C/C++, Java, and .NET
Whei-Jen Chen, John Chun, Naomi Ngan, Rakesh Ranjan, Manoj K. Sardana,
Agosto 2006 - SG24-7301-00
<http://www.redbooks.ibm.com/abstracts/sq247301.html?Open>
2. Understanding DB2 – Learning Visually with Examples V9.5
Raul F. Chong, et all. January 2008
ISBN-10: 0131580183
3. DB2 9: pureXML overview and fast start by Cynthia M. Saracco, Don Chamberlin,
Rav Ahuja Junio 2006 SG24-7298
<http://www.redbooks.ibm.com/abstracts/sq247298.html?Open>
4. DB2® SQL PL: Essential Guide for DB2® UDB on Linux™, UNIX®, Windows™,
i5/OS™, and z/OS®, 2nd Edition
Zamil Janmohamed, Clara Liu, Drew Bradstock, Raul Chong, Michael Gao, Fraser
McArthur, Paul Yip
ISBN: 0-13-100772-6
5. Redbook: DB2 pureXML Guide
Whei-Jen Chen, Art Sammartino, Dobromir Goutev, Felicity Hendricks, Ippei Komi,
Ming-Pang Wei, Rav Ahuja, Matthias Nicola. August 2007
<http://www.redbooks.ibm.com/abstracts/sq247315.html?Open>
6. Information on Demand - Introduction to DB2 9 New Features
Paul Zikopoulos, George Baklarz, Chris Eaton, Leon Katsnelson
ISBN-10: 0071487832
ISBN-13: 978-0071487832
7. Redbook: Developing PHP Applications for IBM Data Servers.
Whei-Jen Chen, Holger Kirstein, Daniel Krook, Kiran H Nair, Piotr Pietrzak
May 2006 - SG24-7218-00
<http://www.redbooks.ibm.com/abstracts/sq247218.html?Open>

emails de Contacto

Mailbox general de DB2 Express-C: db2x@ca.ibm.com

Mailbox general del programa DB2 on Campus: db2univ@ca.ibm.com



Conocer el DB2 no podría ser mas fácil. Lea este libro y:

- **Aprenda que es el DB2 Express-C**
- **Comprenda la arquitectura, herramientas, y seguridad de DB2**
- **Aprenda como administrar bases de datos DB2**
- **Escriba SQL, XQuery, procedimientos almacenados**
- **Desarrolle aplicaciones para DB2**
- **Practique con los ejercicios que se incluyen**
- **Prepárese para el test de DB2 on Campus**

La rápida adopción de XML para la integración de aplicaciones, Web 2.0, y SOA está empujando la necesidad de servidores de datos híbridos. DB2 Express-C de IBM es un servidor de datos híbrido, sin costo, sin límites, y capaz de manejar tanto datos XML como datos relacionales sin problemas. Sin costo significa que DB2 Express-C es gratis para bajar, gratis para desarrollar aplicaciones, gratis para desplegar en producción, y gratis para distribuir con su aplicativo. DB2 no pone límites artificiales en el tamaño de la base de datos, el número de bases de datos, o el número de usuarios.

DB2 Express-C corre en sistemas Windows y Linux e incluye los drivers de muchos lenguajes de programación tales como C/C++, Java, .NET, PHP, Perl, y Ruby. Opcionalmente, a un costo bajo puede comprar una suscripción de soporte técnico y otras características. Si requiere de mayor escalabilidad o funcionalidades mas avanzadas, usted puede sin problemas desplegar las aplicaciones que creó usando DB2 Express-C a otras ediciones de DB2 tal como el DB2 Enterprise.

Esta edición gratis de DB2 es ideal para desarrolladores, consultores, distribuidores, administradores de base de datos (DBAs), estudiantes, o cualquiera que tenga la intención de desarrollar, probar, desplegar, o distribuir aplicaciones de bases de datos. Únete a la comunidad de usuarios de DB2 Express-C que crece cada día y prueba el DB2 Express-C. Empieza a descubrir como puedes crear la siguiente generación de aplicativos y desarrolle soluciones innovadoras.