

IBM® DB2® Universal Database



Iniciación al SQL

Versión 7

IBM® DB2® Universal Database



Iniciación al SQL

Versión 7

Antes de utilizar esta información y el producto al que da soporte, asegúrese de leer la información general incluida en el "Apéndice C. Avisos" en la página 115.

Este documento contiene información sobre productos patentados de IBM. Se proporciona de acuerdo con un contrato de licencia y está protegido por la ley de la propiedad intelectual. La presente publicación no incluye garantías de producto y las declaraciones que contiene no deben interpretarse como tales.

Puede solicitar publicaciones a través del representante de IBM o sucursal de IBM de su localidad, o bien llamando a los números de teléfono 1-800-879-2755, en los Estados Unidos, o 1-800-IBM-4YOU, en Canadá.

Cuando envía información a IBM, otorga a IBM un derecho no exclusivo para utilizar o distribuir dicha información en la forma en que IBM considere adecuada, sin contraer ninguna obligación con el remitente.

© Copyright International Business Machines Corporation 1993, 2000. Reservados todos los derechos.

Contenido

Bienvenido	v	Capítulo 5. Expresiones y subconsultas.	35
Documentación afin para este manual.	v	Selecciones completas escalares	35
Convenios tipográficos de resaltado	vi	Conversión de tipos de datos	35
Capítulo 1. Bases de datos relacionales y SQL	1	Expresiones CASE	36
Capítulo 2. Organización de los datos	3	Expresiones de tabla	37
Tablas	3	Expresiones de tabla anidadas	38
Vistas	4	Expresiones de tabla comunes	39
Esquemas	4	Nombres de correlación	40
Tipos de datos	5	Subconsultas correlacionadas	41
Capítulo 3. Creación de tablas y vistas	9	Implementación de una subconsulta correlacionada	43
Creación de tablas	9	Capítulo 6. Utilización de operadores y predicados en consultas	47
Inserción de datos	10	Combinación de consultas mediante operadores de conjunto	47
Cambio de datos	12	Operador UNION	47
Supresión de datos.	13	Operador EXCEPT	48
Creación de vistas	13	Operador INTERSECT	49
Utilización de vistas para manejar datos	15	Predicados	50
Capítulo 4. Utilización de sentencias SQL para acceder a datos	17	Utilización del predicado IN	50
Conexión a una base de datos	18	Utilización del predicado BETWEEN.	50
Investigación de errores	18	Utilización del predicado LIKE.	51
Selección de columnas	19	Utilización del predicado EXISTS	51
Selección de filas	20	Predicados cuantificados	52
Clasificación de filas	22	Capítulo 7. SQL avanzado.	53
Supresión de filas duplicadas	24	Aplicación de reglas de negocio con restricciones y desencadenantes	53
Orden de operaciones.	24	Claves	54
Utilización de expresiones para calcular valores.	25	Restricciones de unicidad	54
Asignación de nombres a las expresiones	26	Restricciones de integridad referencial	54
Selección de datos de más de una tabla	26	Restricciones de comprobación de tabla.	55
Utilización de una subconsulta.	27	Desencadenantes	56
Utilización de funciones	28	Uniones	60
Funciones de columna	29	Consultas complejas	65
Funciones escalares	30	Consultas ROLLUP y CUBE	65
Funciones de tabla	30	Consultas recursivas	65
Agrupación	31	Funciones OLAP	66
Utilización de una cláusula WHERE con una cláusula GROUP BY.	31	Capítulo 8. Personalización y mejora del manejo de datos	67
Utilización de la cláusula HAVING después de la cláusula GROUP BY	32	Tipos definidos por el usuario	67
		Funciones definidas por el usuario	68
		Objetos grandes.	69

Manejo de objetos grandes (LOB)	70
Registros especiales	70
Introducción a las vistas de catálogo	71
Selección de filas en catálogos del sistema	71

Foto de Adamson	90
Currículum de Adamson	90
Foto de Walker	91
Currículum de Walker	92

Apéndice A. Tablas de base de datos de ejemplo 73

La base de datos de ejemplo	74
Para crear la base de datos de ejemplo	74
Borrado de la base de datos de ejemplo	74
Tabla CL_SCHED	75
Tabla DEPARTMENT	75
Tabla EMPLOYEE	75
Tabla EMP_ACT	79
Tabla EMP_PHOTO	81
Tabla EMP_RESUME	81
Tabla IN_TRAY	82
Tabla ORG	82
Tabla PROJECT	83
Tabla SALES	84
Tabla STAFF	85
Tabla STAFFG	86
Archivos de ejemplo con el tipo de datos BLOB y CLOB	87
Foto de Quintana	87
Currículum de Quintana	87
Foto de Nicholls	88
Currículum de Nicholls	89

Apéndice B. Utilización de la biblioteca de DB2 93

Archivos PDF y manuales impresos sobre DB2	93
Información sobre DB2	93
Impresión de los manuales PDF	104
Solicitud de los manuales impresos	105
Documentación en línea de DB2	106
Acceso a la ayuda en línea	106
Visualización de información en línea	108
Utilización de los asistentes de DB2	111
Configuración de un servidor de documentos	112
Búsqueda de información en línea	113

Apéndice C. Avisos. 115

Marcas registradas	118
------------------------------	-----

Índice 121

Cómo ponerse en contacto con IBM	125
Información del producto	125

Bienvenido

Este manual está pensado para proporcionar a los usuarios información básica sobre el Lenguaje de Consulta Estructurada (Structured Query Language, SQL) y las bases de datos relacionales. En el presente manual, el lector encontrará lo siguiente:

- Una descripción de los conceptos básicos utilizados en el producto DB2.
- Una explicación de cómo realizar tareas de manejo de bases de datos.
- Una demostración de las tareas mediante ejemplos sencillos.

Si usted es el administrador del sistema, antes de poner en práctica cualquiera de los ejemplos de este manual, debe realizar lo siguiente:

- Instalar y configurar el servidor tal como se describe en el manual *Guía rápida de iniciación* correspondiente a su sistema operativo. Crear la base de datos SAMPLE utilizando la opción "Primeros Pasos". La base de datos SAMPLE también se puede crear desde un indicador de línea de mandatos. Vea el manual *Consulta de SQL* para obtener detalles. Nota: No coloque datos propios en la base de datos SAMPLE de DB2.
- Crear el ID de usuario del administrador de DB2, tal como se describe en el manual *Guía rápida de iniciación*.

Si usted no es el administrador del sistema, asegúrese de que tiene un ID de usuario válido y la autorización y los privilegios apropiados para acceder a la base de datos SAMPLE.

Documentación afín para este manual

Las publicaciones siguientes pueden serle de utilidad:

<i>Guía rápida de iniciación</i>	Contiene información necesaria para instalar y utilizar el gestor de bases de datos.
<i>Consulta de SQL</i>	Contiene información de consulta sobre SQL.
<i>Administration Guide</i>	Contiene información necesaria para diseñar, implementar y mantener una base de datos, a la que se accede de forma local o en un entorno cliente/servidor.
<i>Application Development Guide</i>	Explica el proceso de desarrollo de aplicaciones y cómo codificar, compilar y ejecutar programas de aplicación que utilizan SQL incorporado para acceder a la base de datos, o se ejecutan como procedimientos almacenados DB2 utilizando el lenguaje de procedimientos del SQL (u otros lenguajes de programación soportados).

Convenios tipográficos de resaltado

En el presente manual se utilizan los convenios siguientes:

Negrita	En los ejemplos, indica mandatos y palabras claves predefinidas por el sistema.
----------------	---

<i>Cursiva</i>	Se utiliza para indicar algo de lo siguiente: <ul style="list-style-type: none">• La primera aparición de un término nuevo• Término que remite a otra fuente de información.
----------------	---

MAYÚSCULAS	Se utiliza para indicar algo de lo siguiente: <ul style="list-style-type: none">• Mandatos y palabras claves predefinidas por el sistema.• Ejemplos de valores datos determinados o nombres de columnas.
-------------------	---

Capítulo 1. Bases de datos relacionales y SQL

En una *base de datos relacional*, los datos se almacenan en *tablas*. Una tabla es un conjunto de *filas* y *columnas*. La Figura 1 en la página 4 es un ejemplo gráfico de una tabla. En la figura, las columnas (vertical) y las filas (horizontal) aparecen marcadas. El *Lenguaje de Consulta Estructurada (SQL)* se utiliza para recuperar o actualizar datos mediante la especificación de columnas, tablas y las diversas relaciones que existen entre ellas.

El SQL es un lenguaje estandarizado para definir y manejar datos en una base de datos relacional. Las sentencias de SQL se ejecutan mediante un *gestor de bases de datos*. Un gestor de bases de datos es un programa de software que maneja los datos.

Una base de datos relacional *particionada* es una base de datos relacional en la que los datos se manejan entre varias particiones (también llamadas *nodos*). Una forma sencilla de visualizar conceptualmente las particiones es considerar que cada partición es una máquina física. En el presente manual, centraremos nuestra atención en las bases de datos de partición única.

Puede acceder a la base de datos de ejemplo (SAMPLE) y ensayar los ejemplos de este manual mediante el SQL *interactivo*, utilizando una interfaz tal como el Procesador de línea de mandatos (CLP) o el Centro de mandatos (CC).

Capítulo 2. Organización de los datos

Este capítulo ofrece descripciones conceptuales importantes de *tablas*, *vistas* y *esquemas*. Se da una visión general que muestra la conexión existente entre los diferentes componentes de una base de datos relacional. La última sección contiene un breve análisis de algunos de los tipos de datos importantes de uso más frecuente.

Tablas

Las tablas son estructuras lógicas formadas por un número definido de *columnas* y un número variable de *filas*. Una columna es un conjunto de valores pertenecientes a un mismo tipo de datos. Una fila es una sucesión de valores que forman un registro individual de la tabla. Las filas no están necesariamente ordenadas dentro de una tabla. Para ordenar el conjunto resultante, es necesario especificar explícitamente una ordenación en la sentencia SQL que selecciona datos de la tabla. En la intersección de cada fila con cada columna se encuentra un dato elemental determinado que se denomina *valor*. En la Figura 1 en la página 4, 'Sanders' es un ejemplo de un valor contenido en la tabla.

Una *tabla base* contiene datos de usuario y se crea mediante la sentencia CREATE TABLE. Una *tabla resultante* es un conjunto de filas que el gestor de bases de datos selecciona o genera a partir de una o más tablas base para satisfacer una consulta.

La Figura 1 en la página 4 muestra una sección de una tabla. Las columnas y filas aparecen marcadas.

	ID	NAME	DEPT	J
Fila {	10	Sanders	20	Mg
	20	Pernal	20	Sa
	30	Marenghi	38	Mg
	40	O'Brien	38	Sa
	50	Hanes	15	Mg
	60	Quigley	38	Sa

Figura 1. Visualización de una tabla

Vistas

Una *vista* proporciona una forma alternativa de examinar los datos de una o más tablas. Es una ventana dinámica sobre tablas.

Las vistas permiten que varios usuarios vean presentaciones diferentes de los mismos datos. Por ejemplos, varios usuarios pueden estar accediendo a una tabla de datos sobre los empleados de una empresa. Un jefe de departamento puede ver datos sobre sus empleados, pero no los datos referidos a empleados de otro departamento. Un jefe de selección de personal puede ver las fechas de contratación de todos los empleados, pero no sus salarios, mientras que un jefe de finanzas ve los salarios pero no las fechas de contratación. Cada uno de estos usuarios trabaja con una vista obtenida a partir de una tabla real. Cada vista tiene el aspecto de una tabla y un nombre propio.

La ventaja de utilizar vistas es que pueden servir para controlar el acceso a datos confidenciales. De esta forma, personas diferentes pueden tener acceso a columnas o filas diferentes de los datos.

Esquemas

Un *esquema* es un conjunto de objetos con nombre (tablas y vistas, por ejemplo). Un esquema proporciona una clasificación lógica de los objetos de la base de datos.

Un esquema se crea implícitamente cuando el usuario crea una tabla, una vista o cualquier otro objeto con nombre. O bien, puede crear un esquema explícitamente utilizando la sentencia CREATE SCHEMA.

Cuando crea un objeto con nombre, puede *calificar* (asociar) su nombre con el nombre del esquema determinado. El nombre de un objeto con nombre tiene dos partes, la primera de las cuales es el nombre del esquema al que está asignado el objeto. Si no especifica un nombre de esquema, el objeto se asigna al esquema por omisión. (El nombre del esquema por omisión es el *ID de autorización* del usuario que ejecuta la sentencia.)

En el SQL interactivo, que es el método utilizado para ejecutar los ejemplos de este manual, el ID de autorización es el ID de usuario especificado mediante la sentencia CONNECT. Por ejemplo, si el nombre de una tabla es STAFF, y el ID de usuario especificado es USERXYZ, entonces el nombre de tabla calificado es USERXYZ.STAFF. Vea “Conexión a una base de datos” en la página 18 para obtener detalles sobre la sentencia CONNECT.

Algunos nombres de esquema son reservados. Por ejemplo, las *funciones incorporadas* pertenecen al esquema SYSIBM, mientras que las *funciones definidas por usuario* preinstaladas pertenecen al esquema SYSFUN. Vea el manual *Consulta de SQL* para obtener detalles sobre la sentencia CREATE SCHEMA.

Tipos de datos

Los tipos de datos definen valores aceptables para constantes, columnas, variables de lenguaje principal, funciones, expresiones y registros especiales. Esta sección describe los tipos de datos a los que hacen referencia los ejemplos. Para obtener una lista y descripción completas de otros tipos de datos, vea el manual *Consulta de SQL*.

Serie de caracteres

Una *serie de caracteres* es una secuencia de bytes. La longitud de la serie es el número de bytes de la secuencia. Si la longitud es cero, el valor se denomina *serie vacía*.

Serie de caracteres de longitud fija

CHAR(x) es una serie de caracteres de longitud fija. El atributo de longitud x debe estar comprendido entre 1 y 254, inclusive.

Serie de caracteres de longitud variable

Las series de caracteres de longitud variable son de tres tipos: VARCHAR, LONG VARCHAR y CLOB.

Los tipos VARCHAR(x) son series de caracteres de longitud variable, por lo que una serie de longitud 9 se puede insertar en VARCHAR(15), pero la longitud de la serie será 9.

Vea “Objetos grandes” en la página 69 para obtener detalles sobre CLOB.

Serie gráfica de caracteres

Una *serie gráfica* es una secuencia de datos de caracteres de doble byte.

Serie gráfica de longitud fija

GRAPHIC(x) es una serie de caracteres de longitud fija. El atributo de longitud x debe estar comprendido entre 1 y 127, inclusive.

Serie gráfica de longitud variable

Las series gráficas de longitud variable son de tres tipos: VARGRAPHIC, LONG VARGRAPHIC y DBCLOB. Vea “Objetos grandes” en la página 69 para obtener detalles sobre DBCLOB.

Serie binaria de caracteres

Una *serie binaria* es una secuencia de bytes. Se utiliza para representar datos no tradicionales, tales como imágenes. Binary Large Object (BLOB) es una serie binaria. Vea “Objetos grandes” en la página 69 para obtener más información.

Números

Todos los números tienen un signo y una *precisión*. La precisión es el número de bits o dígitos excluido el signo.

SMALLINT

SMALLINT (“small integer”) es un número entero de dos bytes con una precisión de 5 dígitos.

INTEGER

INTEGER (“large integer”) es un número entero de cuatro bytes con una precisión de 10 dígitos.

BIGINT

BIGINT (“big integer”) es un número entero de ocho bytes con una precisión de 19 dígitos.

REAL *REAL* (número de coma flotante y precisión simple) es una aproximación de 32 bits de un número real.

DOUBLE

DOUBLE (número de coma flotante y precisión doble) es una aproximación de 64 bits de un número real. DOUBLE también se denomina FLOAT.

DECIMAL(p,s)

DECIMAL es un número decimal. La posición de la coma decimal está determinada por la *precisión* (*p*) y la *escala* (*s*) del número. La precisión es el número total de dígitos y debe ser menor que 32. La escala es el número de dígitos de la parte fraccionaria y siempre es menor o igual que el valor de la precisión. Si no se especifican la precisión ni la escala, por omisión toman los valores de 5 y 0, respectivamente.

Valores de fecha y hora

Los valores de fecha y hora son representaciones de fechas, horas e indicaciones de fecha y hora (serie de caracteres de 14 dígitos que representa una fecha y hora válidas en el formato *aaaaxxddhhmmss*). Los valores de fecha y hora se pueden utilizar en determinadas operaciones aritméticas y operaciones sobre series de caracteres; son compatibles con determinadas series de caracteres, pero no son ni series ni números.¹

Fecha Una *fecha* es un valor que consta de tres partes (año, mes y día).

Hora Una *hora* es un valor que consta de tres partes (hora, minuto y segundo) y designa una hora del día de acuerdo con un reloj de 24 horas.

Indicación de fecha y hora

Una *indicación de fecha y hora* es un valor que consta de siete partes (año, mes, día, hora, minuto, segundo y microsegundo) y designa una fecha y una hora.

Valor nulo

El valor *nulo* es un valor especial que es distinto de todos los valores no nulos. Significa la ausencia de cualquier otro valor en la columna y fila en cuestión. El valor nulo existe para todos los tipos de datos.

La tabla siguiente destaca las características de los tipos de datos utilizados en los ejemplos. Todos los tipos de datos numéricos están definidos en un rango determinado. El rango de los tipos de datos numéricos también se incluye en la tabla. Puede utilizar esta tabla como guía de consulta rápida para conocer la utilización correcta de los tipos de datos.

1. En el presente manual, se hace referencia a las representaciones ISO de los valores de fecha y hora.

Tipo de datos	Tipo	Característica	Ejemplo o rango
CHAR(15)	serie de caracteres de longitud fija	longitud máxima igual a 254	'día soleado'
VARCHAR(15)	serie de caracteres de longitud variable	longitud máxima igual a 32672	'día soleado'
SMALLINT	número	2 bytes de longitud 5 dígitos de precisión	rango de -32768 a 32767
INTEGER	número	4 bytes de longitud 10 dígitos de precisión	rango de -2147483648 a 2147483647
BIGINT	número	8 bytes de longitud 19 dígitos de precisión	rango de -9223372036854775808 a 9223372036854775807
REAL	número	aproximación de 32 bits de coma flotante y precisión simple	rango de -3,402E+38 a -1,175E-37 o de 1,175E-37 a 3,402E+38 o cero
DOUBLE	número	aproximación de 64 bits de coma flotante y precisión doble	rango de -1,79769E+308 a -2,225E-307 o de 2,225E-307 a 1,79769E+308 o cero
DECIMAL(5,2)	número	la precisión es 5 la escala es 2	rango de -10**31+1 a 10**31-1
DATE	fecha y hora	valor de tres partes	1991-10-27
TIME	fecha y hora	valor de tres partes	13.30.05
TIMESTAMP	fecha y hora	valor de siete partes	1991-10-27-13.30.05.000000

Vea la tabla Compatibilidad de los Tipos de Datos, en el manual *Consulta de SQL*, para obtener más información.

Capítulo 3. Creación de tablas y vistas

Este capítulo describe cómo puede crear y manejar tablas y vistas en DB2 Universal Database. Mediante diagramas y ejemplos se examina la relación entre tablas y vistas.

Este capítulo trata los temas siguientes:

- Creación de tablas y Creación de vistas
- Inserción de datos
- Cambio de datos
- Supresión de datos
- Utilización de vistas para manejar datos

Creación de tablas

Para crear sus propias tablas utilice la sentencia `CREATE TABLE`, especificando los nombres y tipos de las columnas, así como *restricciones*. Las restricciones se analizan en “Aplicación de reglas de negocio con restricciones y desencadenantes” en la página 53.

La sentencia siguiente crea una tabla llamada `PERS`, que es similar a la tabla `STAFF`, pero tiene una columna más para la fecha de nacimiento.

```
CREATE TABLE PERS
( ID          SMALLINT          NOT NULL,
  NAME        VARCHAR(9),
  DEPT        SMALLINT WITH DEFAULT 10,
  JOB         CHAR(5),
  YEARS       SMALLINT,
  SALARY      DECIMAL(7,2),
  COMM        DECIMAL(7,2),
  BIRTH_DATE  DATE)
```

Esta sentencia crea una tabla que no contiene datos. La sección siguiente describe cómo insertar datos en una tabla nueva.

Tal como muestra el ejemplo, se especifica un nombre y un tipo de datos para cada columna. Los tipos de datos se tratan en “Tipos de datos” en la página 5. `NOT NULL` es opcional y se puede especificar para indicar que no se permiten valores nulos en una columna. Los valores por omisión también son opcionales.

Existen otras muchas opciones que puede especificar en una sentencia `CREATE TABLE`, tales como *restricciones de unicidad o restricciones referenciales*.

Para obtener más información sobre todas las opciones, vea la sentencia CREATE TABLE en el manual *Consulta de SQL*.

Inserción de datos

Cuando crea una nueva tabla, no contiene ningún dato. Para entrar nuevas filas en una tabla, debe utilizar la sentencia INSERT. Esta sentencia tiene dos formatos generales:

- En uno de los formatos, se utiliza una cláusula VALUES para especificar valores para las columnas de una o más filas. Este formato general es el utilizado en los tres ejemplos siguientes para insertar datos en las tablas.
- En el otro formato, en lugar de especificar valores, se especifica una *selección completa* para identificar columnas pertenecientes a filas de otras tablas y/o vistas.

La selección completa es una sentencia de selección que se utiliza en las sentencias INSERT o CREATE VIEW, o a continuación de un predicado. Cuando una selección completa está encerrada entre paréntesis suele denominarse *subconsulta*.

Dependiendo de las opciones por omisión que haya elegido al crear la tabla, para cada fila que inserte, debe proporcionar un valor a cada columna o aceptar un valor por omisión. Los valores por omisión de los diversos tipos de datos se tratan en el manual *Consulta de SQL*.

La sentencia siguiente utiliza una cláusula VALUES para insertar una fila de datos en la tabla PERS:

```
INSERT INTO PERS
VALUES (12, 'Harris', 20, 'Sales', 5, 18000, 1000, '1950-1-1')
```

La sentencia siguiente utiliza la cláusula VALUES para insertar tres filas en la tabla PERS, en la que sólo se conocen los ID, los nombres y los puestos de trabajo. Si una columna está definida como NOT NULL y no tiene valor por omisión, debe especificar un valor para ella.

Puede utilizar las palabras WITH DEFAULT para ampliar la cláusula NOT NULL de una definición de columna en una sentencia CREATE TABLE. Si una columna está definida como NOT NULL WITH DEFAULT o como valor por omisión fijo, tal como WITH DEFAULT 10, y no especifica la columna en la lista de columnas, el valor por omisión se coloca en esa columna en la fila insertada. Por ejemplo, en la sentencia CREATE TABLE, se especificó un valor por omisión sólo para la columna DEPT y se definió como igual a 10. Por lo tanto, el número de departamento (DEPT) se establece en 10 y cualquier otra columna cuyo valor no se proporcione explícitamente se establece en NULL.

```

INSERT INTO PERS (NAME, JOB, ID)
VALUES ('Swagerman', 'Prgmr', 500),
       ('Limoges', 'Prgmr', 510),
       ('Li', 'Prgmr', 520)

```

La sentencia siguiente devuelve el resultado de las inserciones:

```

SELECT *
FROM PERS

```

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM	BIRTH_DATE
12	Harris	20	Sales	5	18000.00	1000.00	01/01/1950
500	Swagerman	10	Prgmr	-	-	-	-
510	Limoges	10	Prgmr	-	-	-	-
520	Li	10	Prgmr	-	-	-	-

Observe que, en este caso, no se han especificado valores para cada columna. Los valores NULL se visualizan en forma de guión (-). Para que este esquema funcione, la lista de nombres de columnas debe corresponder, en orden y en tipo de datos, con los valores proporcionados en la cláusula VALUES. Si se omite la lista de nombres de columnas (como ocurre en el primer ejemplo), la lista de los valores de datos que siguen a VALUES deben estar en el mismo orden que las columnas de la tabla en la que se insertan, y el número de valores debe ser igual al número de columnas de la tabla.

Cada valor debe ser compatible con el tipo de datos de la columna donde se inserta el valor. Si la definición de una columna indica que la columna puede contener un valor nulo y no se especifica un valor para esa columna, entonces se da el valor NULL a esa columna en la fila insertada.

El ejemplo siguiente inserta el valor nulo en YEARS, COMM y BIRTH_DATE, pues no se han especificado valores para esas columnas de la fila.

```

INSERT INTO PERS (ID, NAME, JOB, DEPT, SALARY)
VALUES (410, 'Perna', 'Sales', 20, 20000)

```

El segundo formato de la sentencia INSERT es muy práctico para llenar una tabla con valores procedentes de filas de otra tabla. Tal como se mencionó, en lugar de especificar valores, se especifica una selección completa para identificar columnas pertenecientes a filas de otras tablas y/o vistas.

El ejemplo siguiente selecciona datos de la tabla STAFF correspondientes a miembros del departamento 38 y los inserta en la tabla PERS:

```

INSERT INTO PERS (ID, NAME, DEPT, JOB, YEARS, SALARY)
SELECT ID, NAME, DEPT, JOB, YEARS, SALARY
FROM STAFF
WHERE DEPT = 38

```

Después de esta inserción, la siguiente sentencia SELECT produce un resultado igual a la selección completa de la sentencia INSERT.

```
SELECT ID, NAME, DEPT, JOB, YEARS, SALARY
FROM PERS
WHERE DEPT = 38
```

El resultado es:

ID	NAME	DEPT	JOB	YEARS	SALARY
30	Marenghi	38	Mgr	5	17506.75
40	O'Brien	38	Sales	6	18006.00
60	Quigley	38	Sales	-	16808.30
120	Naughton	38	Clerk	-	12954.75
180	Abrahams	38	Clerk	3	12009.75

Cambio de datos

Utilice la sentencia UPDATE para modificar los datos de una tabla. Mediante esta sentencia, puede cambiar el valor de una o más columnas para cada fila que cumpla la condición de búsqueda impuesta por la cláusula WHERE.

El ejemplo siguiente actualiza información referente al empleado cuyo ID es 410:

```
UPDATE PERS
SET JOB='Prgmr', SALARY = SALARY + 300
WHERE ID = 410
```

La cláusula SET especifica las columnas que se deben actualizar y proporciona los valores.

La cláusula WHERE es opcional y especifica las filas que se deben actualizar. Si se omite la cláusula WHERE, el gestor de bases de datos actualiza cada fila de la tabla o vista con los valores proporcionados por el usuario.

En este ejemplo, primero se menciona la tabla (PERS), después se especifica una condición para la fila que se debe actualizar. La información correspondiente al empleado número 410 ha cambiado: su puesto de trabajo ahora es Prgmr y su salario ha aumentado en \$300.

Puede modificar los datos de varias filas incluyendo una cláusula WHERE que sea aplicable a dos o más filas. El ejemplo siguiente aumenta el salario de cada vendedor en un 15%:

```
UPDATE PERS
SET SALARY = SALARY * 1.15
WHERE JOB = 'Sales'
```

Supresión de datos

Utilice la sentencia DELETE para suprimir filas de datos de una tabla basándose en la condición de búsqueda especificada por la cláusula WHERE. El ejemplo siguiente suprime la fila donde el ID de empleado es 120:

```
DELETE FROM PERS
WHERE ID = 120
```

La cláusula WHERE es opcional y especifica las filas que se deben suprimir. Si se omite la cláusula WHERE, el gestor de bases de datos suprime todas las filas de la tabla o vista.

Puede utilizar la sentencia DELETE para suprimir más de una fila. El ejemplo siguiente suprime todas las filas donde el departamento del empleado es 20:

```
DELETE FROM PERS
WHERE DEPT = 20
```

Cuando suprime una fila, suprime la fila completa, no valores determinados de ella.

Para suprimir la definición de una tabla así como su contenido, emita la sentencia DROP TABLE, tal como se describe en el manual *Consulta de SQL*.

Creación de vistas

Tal como se describe en “Vistas” en la página 4, una vista proporciona una forma alternativa de examinar los datos de una o más tablas. Mediante la creación de vistas, puede restringir la información a la que podrán acceder los diversos usuarios. El diagrama siguiente muestra la relación entre vistas y tablas.

En la Figura 2 en la página 14, View_A permite el acceso sólo a las columnas AC1 y AC2 de TABLE_A.

View_AB permite el acceso a la columna AC3 de TABLE_A y a la columna BC2 de TABLE_B.

Mediante View_A, restringe el acceso que los usuarios tienen a TABLE_A, y mediante VIEW_AB, limita el acceso a determinadas columnas de ambas tablas.

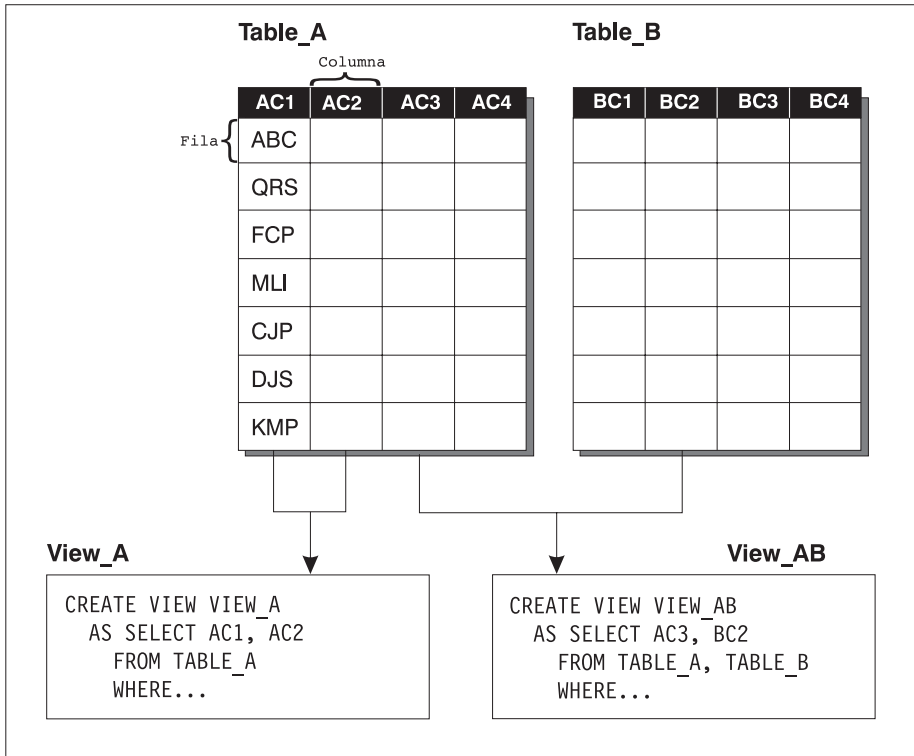


Figura 2. Relación entre tablas y vistas

La sentencia siguiente crea una vista de los empleados no directivos del departamento 20 de la tabla STAFF, donde no se muestra ni el salario ni la comisión contenidos en la tabla base.

```
CREATE VIEW STAFF_ONLY
AS SELECT ID, NAME, DEPT, JOB, YEARS
FROM STAFF
WHERE JOB <> 'Mgr' AND DEPT=20
```

Después de crear la vista, la sentencia siguiente visualiza el contenido de la vista:

```
SELECT *
FROM STAFF_ONLY
```

Esta sentencia produce el resultado siguiente:

ID	NAME	DEPT	JOB	YEARS
20	Pernal	20	Sales	8
80	James	20	Clerk	-
190	Sneider	20	Clerk	8

Como ejemplo adicional, podemos utilizar las tablas STAFF y ORG para crear una vista que lista el nombre de cada departamento y el nombre del jefe de ese departamento. La sentencia siguiente crea dicha vista:

```
CREATE VIEW DEPARTMENT_MGRS
AS SELECT NAME, DEPTNAME
FROM STAFF, ORG
WHERE MANAGER = ID
```

Puede limitar la posibilidad de hacer inserciones y actualizaciones en una tabla mediante una vista utilizando la cláusula WITH CHECK OPTION al crear una vista. Esta cláusula hace que el gestor de bases de datos verifique que cualquier actualización o inserción en la vista se ajuste a la definición de la vista, y rechace las que no sean conformes. Si se omite esta cláusula, las inserciones y actualizaciones no se comparan con la definición de la vista. Para obtener detalles sobre cómo opera la cláusula WITH CHECK OPTION, vea la sentencia CREATE VIEW en el manual *Consulta de SQL*.

Utilización de vistas para manejar datos

Al igual que la sentencia SELECT, las sentencias INSERT, DELETE y UPDATE se aplican a una vista como si ésta fuera una tabla real. Las sentencias manejan los datos contenidos en la(s) tabla(s) base asociada(s). Cuando se accede de nuevo a la vista, se evalúa utilizando las tablas base más actuales. Si no se utiliza la cláusula WITH CHECK OPTION, los datos modificados mediante una vista pueden no aparecer en los nuevos accesos a la vista, pues puede que los datos ya no sean conformes a la definición original de la vista.

El ejemplo siguiente actualiza la vista FIXED_INCOME:

```
CREATE VIEW FIXED_INCOME (LNAME, DEPART, JOBTITLE, NEWSALARY)
AS SELECT NAME, DEPT, JOB, SALARY
FROM PERS
WHERE JOB <> 'Sales' WITH CHECK OPTION

UPDATE FIXED_INCOME
SET NEWSALARY = SALARY * 1.10
WHERE LNAME = 'Li'
```

La actualización realizada en la vista anterior equivale (salvo por la opción de validación) a actualizar la tabla base PERS:

```
UPDATE PERS
SET SALARY = SALARY * 1.10
WHERE NAME = 'Li'
AND JOB <> 'Sales'
```

Observe que debido a que la vista se crea utilizando WITH CHECK OPTION para la restricción JOB <> 'Sales' en CREATE VIEW FIXED_INCOME, la actualización siguiente no se permitirá cuando el empleado Limoges pase a Ventas:

```
UPDATE FIXED_INCOME
SET JOBTITLE = 'Sales'
WHERE LNAME = 'Limoges'
```

Las columnas definidas por expresiones tales como SALARY + COMM o SALARY * 1.25 no se pueden actualizar. Si define una vista que contiene una o más de tales columnas, el propietario no recibe el privilegio de actualización para dichas columnas. Las sentencias INSERT no están permitidas en las vistas que contienen tales columnas, pero sí que pueden utilizarse sentencias DELETE.

Considere una tabla PERS donde ninguna columna está definida como NOT NULL. Podría insertar filas en la tabla PERS mediante la vista FIXED_INCOME, aunque la vista no contenga las columnas ID, YEARS, COMM ni BIRTHDATE de la tabla PERS en la que se basa. Las columnas no visualizables mediante la vista se establecen en NULL o en el valor por omisión, según corresponda.

Sin embargo, la tabla PERS tiene la columna ID definida como NOT NULL. Si intenta insertar una fila mediante la vista FIXED_INCOME, el sistema intenta insertar valores nulos en todas las columnas de PERS que la vista “no puede visualizar”. Debido a que la columna ID no está incluida en la vista y no permite valores nulos, el sistema no autoriza la inserción mediante la vista.

Para conocer las reglas y restricciones referentes a la modificación de vistas, vea la sentencia CREATE VIEW en el manual *Consulta de SQL*.

Capítulo 4. Utilización de sentencias SQL para acceder a datos

Esta sección describe cómo conectarse a una base de datos y recuperar datos utilizando sentencias SQL.

Los ejemplos muestran la sentencia a entrar seguida (en la mayoría de los casos) por los resultados que se obtendrían al emitir la sentencia para la base de datos de ejemplo. Observe que aunque las sentencias aparecen escritas en mayúsculas, se pueden entrar según una combinación cualquiera de caracteres en mayúsculas y minúsculas (excepto cuando la sentencia está encerrada entre comillas simples (') o comillas dobles ('')).

La base de datos SAMPLE, incluida con DB2 Universal Database, consta de varias tablas, que están listadas en el “Apéndice A. Tablas de base de datos de ejemplo” en la página 73. Esta base de datos se puede crear utilizando el panel de instalación “Primeros Pasos”. También puede crear la base de datos SAMPLE desde la línea de mandatos. Vea el manual *Consulta de SQL* para obtener más detalles.

Observe que DB2 Universal Database incluye otras bases de datos de ejemplo para mostrar la funcionalidad del Data Warehouse Center y del OLAP Starter Kit. Los ejemplos del presente manual sólo utilizan la base de datos general SAMPLE.

Dependiendo de cómo se haya configurado la base de datos, puede ser necesario que califique los nombres de tablas utilizados anteponiendo como prefijo el nombre de esquema y un punto. Para los ejemplos de este manual, se supone que el esquema por omisión es USERID. Por tanto, puede hacer referencia a la tabla ORG como USERID.ORG. Consulte al administrador si esto es necesario o no.

Este capítulo trata los temas siguientes:

- Conexión a una base de datos
- Investigación de errores
- Selección de columnas y Selección de filas
- Clasificación de filas y Supresión de filas duplicadas
- Orden de operaciones
- Utilización de expresiones para calcular valores
- Asignación de nombres a las expresiones

- Selección de datos de más de una tabla
- Utilización de una subconsulta
- Utilización de funciones
- Agrupación

Conexión a una base de datos

Es necesario que se conecte a una base de datos para poder utilizar sentencias SQL a fin de consultar o manejar la base de datos. La sentencia `CONNECT` asocia una conexión de base de datos con un nombre de usuario.

Por ejemplo, para conectarse a la base de datos `SAMPLE`, escriba el mandato siguiente en el procesador de línea de mandatos de DB2 :

```
CONNECT TO SAMPLE USER USERID USING PASSWORD
```

(Asegúrese de que elige un ID de usuario y una contraseña que sean válidos en el sistema servidor).

En este ejemplo, `USER` es `USERID` y `USING` es `PASSWORD`.

El mensaje siguiente le indica que ha establecido una conexión satisfactoria:

```
Database Connection Information
```

```
Database product      = DB2/NT 7.1.0
SQL authorization ID  = USERID
Local database alias  = SAMPLE
```

Una vez conectado, puede empezar a manejar la base de datos. Para obtener más detalles sobre las conexiones, consulte la sentencia `CONNECT` en el manual *Consulta de SQL*.

Investigación de errores

Cada vez que comete un error al escribir cualquiera de los ejemplos o se produce un error durante la ejecución de una sentencia SQL, el gestor de bases de datos devuelve un mensaje de error. El mensaje de error consta de un identificador de mensaje, una breve explicación y un estado SQL (`SQLSTATE`).

Los errores `SQLSTATE` son códigos de error comunes a la familia de productos DB2. Los errores `SQLSTATE` se ajustan al estándar `SQL92` de `ISO/ANSI`.

Por ejemplo, si el ID de usuario o la contraseña eran incorrectos en la sentencia `CONNECT`, el gestor de bases de datos devuelve el identificador de mensaje `SQL1403N` y el estado `SQL 08004`. El mensaje es el siguiente:

SQL1403N El nombre de usuario y/o la contraseña proporcionados son incorrectos. SQLSTATE=08004

Puede obtener más información sobre el mensaje de error escribiendo un signo de interrogación (?) seguido del identificador de mensaje o el estado SQL:

```
? SQL1403N
0 BIEN
? SQL1403
0 BIEN
? 08004
```

Observe que la penúltima línea de la descripción del error SQL1403N indica que el código SQL es -1403. El código SQL es un código de error específico del producto. Los identificadores de mensaje que terminan con una N (Notificación) o una C (Crítico) representan un error y tienen un código SQL negativo. Los identificadores de mensaje que terminan con una W (de "Warning", Aviso) representan un aviso y tienen un código SQL positivo.

Selección de columnas

Utilice la sentencia SELECT para seleccionar determinadas columnas de una tabla. En la sentencia, especifique una lista de nombres de columnas separados por comas. Esta lista se denomina *lista de selección*.

La sentencia siguiente selecciona nombres de departamento (DEPTNAME) y números de departamento (DEPTNUMB) de la tabla ORG de la base de datos SAMPLE:

```
SELECT DEPTNAME, DEPTNUMB
FROM ORG
```

Esta sentencia produce el resultado siguiente:

DEPTNAME	DEPTNUMB
Head Office	10
New England	15
Mid Atlantic	20
South Atlantic	38
Great Lakes	42
Plains	51
Pacific	66
Mountain	84

Utilizando un asterisco (*) puede seleccionar todas las columnas de la tabla. El ejemplo siguiente lista todas las columnas y filas de la tabla ORG:

```
SELECT *
FROM ORG
```

Esta sentencia produce el resultado siguiente:

DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
10	Head Office	160	Corporate	New York
15	New England	50	Eastern	Boston
20	Mid Atlantic	10	Eastern	Washington
38	South Atlantic	30	Eastern	Atlanta
42	Great Lakes	100	Midwest	Chicago
51	Plains	140	Midwest	Dallas
66	Pacific	270	Western	San Francisco
84	Mountain	290	Western	Denver

Selección de filas

Para seleccionar determinadas filas de una tabla, a continuación de la sentencia SELECT utilice la cláusula WHERE para especificar la condición o condiciones que debe cumplir una fila para ser seleccionada. Un criterio para seleccionar filas de una tabla es una *condición de búsqueda*.

Una condición de búsqueda consta de uno o más *predicados*. Un predicado especifica una condición que es verdadera o falsa (o desconocida) referente a una fila. Puede especificar condiciones en la cláusula WHERE utilizando los predicados básicos siguientes:

Predicado	Función
x = y	x es igual a y
x <> y	x no es igual a y
x < y	x es menor que y
x > y	x es mayor que y
x <= y	x es menor o igual que y
x >= y	x es mayor o igual que y
IS NULL/IS NOT NULL	verifica la existencia de valores nulos

Cuando defina condiciones de búsqueda, tenga la precaución de realizar operaciones aritméticas sólo sobre tipos de datos numéricos y hacer comparaciones sólo entre tipos de datos compatibles. Por ejemplo, no puede comparar series de texto con números.

Si selecciona filas basándose en un valor de tipo carácter, ese valor debe estar encerrado entre comillas simples (por ejemplo, WHERE JOB = 'Clerk') y cada carácter se debe escribir exactamente tal como existe en la base de datos. Si el

valor de datos está escrito en minúsculas en la base de datos y lo escribe en mayúsculas, no se seleccionará ninguna fila. Si selecciona filas basándose en un valor numérico, ese valor no debe estar encerrado entre comillas (por ejemplo, WHERE DEPT = 20).

El ejemplo siguiente selecciona sólo las filas de la tabla STAFF correspondientes al departamento 20:

```
SELECT DEPT, NAME, JOB
FROM STAFF
WHERE DEPT = 20
```

Esta sentencia produce el resultado siguiente:

DEPT	NAME	JOB
20	Sanders	Mgr
20	Pernal	Sales
20	James	Clerk
20	Sneider	Clerk

El ejemplo siguiente utiliza AND para especificar más de una condición. Puede especificar tantas condiciones como desee. El ejemplo selecciona los oficinistas del departamento 20 de la tabla STAFF:

```
SELECT DEPT, NAME, JOB
FROM STAFF
WHERE JOB = 'Clerk'
AND DEPT = 20
```

Esta sentencia produce el resultado siguiente:

DEPT	NAME	JOB
20	James	Clerk
20	Sneider	Clerk

Se genera un valor nulo cuando no se especifica ningún valor y la columna no acepta un valor por omisión. También puede darse cuando el valor se establece específicamente en un valor nulo. Sólo puede ocurrir en las columnas que están definidas para aceptar valores nulos. La definición y aceptación de valores nulos en tablas se trata en “Creación de tablas” en la página 9.

Utilice los predicados IS NULL y IS NOT NULL para verificar un valor nulo.

La sentencia siguiente lista empleados cuya comisión no se conoce:

```
SELECT ID, NAME
FROM STAFF
WHERE COMM IS NULL
```

Esta sentencia produce el resultado siguiente:

ID	NAME
10	Sanders
30	Marenghi
50	Hanes
100	Plotz
140	Fraye
160	Molinare
210	Lu
240	Daniels
260	Jones
270	Lea
290	Quill

El valor cero no es lo mismo que el valor nulo. La sentencia siguiente selecciona los empleados de una tabla cuya comisión es cero:

```
SELECT ID, NAME
FROM STAFF
WHERE COMM = 0
```

Debido a que no existen valores iguales a cero en la columna COMM de la tabla de ejemplo, el conjunto resultante devuelto esta vacío.

El ejemplo siguiente selecciona todas las filas de la tabla STAFF para las que el valor de YEARS es mayor que 9:

```
SELECT NAME, SALARY, YEARS
FROM STAFF
WHERE YEARS > 9
```

Esta sentencia produce el resultado siguiente:

NAME	SALARY	YEARS
Hanes	20659.80	10
Lu	20010.00	10
Jones	21234.00	12
Quill	19818.00	10
Graham	21000.00	13

Clasificación de filas

Puede desear que la información devuelta esté en un orden determinado. Utilice la cláusula ORDER BY para clasificar la información según los valores de una o más columnas.

La sentencia siguiente visualiza los empleados del departamento 84 clasificados por el número de años en el empleo:

```
SELECT NAME, JOB, YEARS
FROM STAFF
WHERE DEPT = 84
ORDER BY YEARS
```

Esta sentencia produce el resultado siguiente:

NAME	JOB	YEARS
Davis	Sales	5
Gafney	Clerk	5
Edwards	Sales	7
Quill	Mgr	10

Especifique **ORDER BY** como última cláusula de la sentencia **SELECT** completa. Las columnas mencionadas en esta cláusula pueden ser expresiones o cualquier columna de la tabla. Los nombres de columnas indicados en la cláusula **ORDER BY** no es necesario especificarlos en la lista de selección.

Puede ordenar las filas por orden ascendente o descendente, especificando explícitamente **ASC** o **DESC** en la cláusula **ORDER BY**. Si no se especifica ningún orden, las filas se ordenan automáticamente por orden ascendente. La sentencia siguiente visualiza los empleados del departamento 84 clasificados por orden descendente según el número de años en el empleo:

```
SELECT NAME, JOB, YEARS
FROM STAFF
WHERE DEPT = 84
ORDER BY YEARS DESC
```

Esta sentencia produce el resultado siguiente:

NAME	JOB	YEARS
Quill	Mgr	10
Edwards	Sales	7
Davis	Sales	5
Gafney	Clerk	5

Las filas se pueden ordenar por valores de tipo carácter así como por valores numéricos. La sentencia siguiente visualiza los empleados del departamento 84 clasificados por orden alfabético del nombre:

```
SELECT NAME, JOB, YEARS
FROM STAFF
WHERE DEPT = 84
ORDER BY NAME
```

Esta sentencia produce el resultado siguiente:

NAME	JOB	YEARS
Davis	Sales	5
Edwards	Sales	7
Gafney	Clerk	5
Quill	Mgr	10

Supresión de filas duplicadas

Puede desear que la sentencia `SELECT` no le devuelva información duplicada. Por ejemplo, `STAFF` tiene una columna `DEPT` con varios números de departamento que están listados más de una vez, y una columna `JOB` en la que aparecen descripciones repetidas de puestos de trabajo.

Para eliminar filas duplicadas, utilice la opción `DISTINCT` en la cláusula `SELECT`. Por ejemplo, si inserta `DISTINCT` en la sentencia, cada puesto de trabajo de un departamento se lista una sola vez:

```
SELECT DISTINCT DEPT, JOB
FROM STAFF
WHERE DEPT < 30
ORDER BY DEPT, JOB
```

Esta sentencia produce el resultado siguiente:

DEPT	JOB
10	Mgr
15	Clerk
15	Mgr
15	Sales
20	Clerk
20	Mgr
20	Sales

`DISTINCT` ha suprimido todas las filas que contienen datos duplicados en el conjunto de columnas especificadas en la sentencia `SELECT`.

Orden de operaciones

Es importante tener en cuenta el orden de las operaciones. El resultado de una cláusula es la entrada para la siguiente, como se indica en la lista mostrada más abajo. En la sección “Asignación de nombres a las expresiones” en la página 26 se muestra un ejemplo donde el orden de las operaciones es importante.

La secuencia siguiente de operaciones no es necesariamente la forma en que las operaciones se realizan dentro del código de DB2. Esta explicación sencilla simplemente permite tener una visión más intuitiva de las consultas de bases de datos. La secuencia de operaciones es la siguiente:

1. Cláusula FROM
2. Cláusula WHERE
3. Cláusula GROUP BY
4. Cláusula HAVING
5. Cláusula SELECT
6. Cláusula ORDER BY

Utilización de expresiones para calcular valores

Una *expresión* es un cálculo o función incluido en una sentencia. La sentencia siguiente calcula los salarios que recibirían los empleados del departamento 38 si cada uno recibiera una prima de \$500:

```
SELECT DEPT, NAME, SALARY + 500
      FROM STAFF
      WHERE DEPT = 38
      ORDER BY 3
```

El resultado es:

DEPT	NAME	3

38	Abrahams	12509.75
38	Naughton	13454.75
38	Quigley	17308.30
38	Marenghi	18006.75
38	O'Brien	18506.00

Observe que el nombre de la tercera columna es un número. Este número lo genera el sistema, pues `SALARY+500` no especifica un nombre de columna. Posteriormente, la cláusula `ORDER BY` utiliza este número para hacer referencia a la tercera columna. La sección “Asignación de nombres a las expresiones” en la página 26 describe cómo dar nombres significativos a las expresiones.

Puede formar expresiones aritméticas utilizando los operadores aritméticos básicos correspondientes a la suma (+), la resta (-), la multiplicación (*) y la división (/).

Los operadores pueden actuar sobre valores numéricos contenidos en diversos tipos de operandos, algunos de los cuales son:

- Nombres de columnas (como en `RATE * HOURS`)
- Valores constantes (como en `RATE * 1.07`)
- Funciones escalares (como en `LENGTH(NAME) + 1`).

Asignación de nombres a las expresiones

La cláusula opcional AS le permite asignar un nombre significativo a una expresión, lo cual hace más fácil hacer referencia a la expresión. Puede utilizar una cláusula AS para dar un nombre a un elemento cualquiera de la lista de selección.

La sentencia siguiente visualiza todos los empleados cuyo salario más comisión es menor que \$13.000. La expresión SALARY + COMM se denomina PAY:

```
SELECT NAME, JOB, SALARY + COMM AS PAY
FROM STAFF
WHERE (SALARY + COMM) < 13000
ORDER BY PAY
```

Esta sentencia produce el resultado siguiente:

NAME	JOB	PAY
-----	-----	-----
Yamaguchi	Clerk	10581.50
Burke	Clerk	11043.50
Scoutten	Clerk	11592.80
Abrahams	Clerk	12246.25
Kermisch	Clerk	12368.60
Ngan	Clerk	12714.80

Mediante la cláusula AS, puede hacer referencia a un nombre de columna determinado en la cláusula ORDER BY, en lugar de utilizar el número generado por el sistema. En este ejemplo se compara (SALARY + COMM) con 13000 en la cláusula WHERE, en lugar de utilizar el nombre PAY. Esto es debido al orden de las operaciones. La cláusula WHERE se evalúa antes de asignar el nombre PAY a (SALARY + COMM), debido a que la sentencia SELECT se ejecuta después de la cláusula WHERE. Por ello, no se puede utilizar PAY en el predicado.

Selección de datos de más de una tabla

Puede utilizar la sentencia SELECT para crear informes que contengan información procedente de dos o más tablas. Esta operación se suele denominar *unión*. Por ejemplo, puede unir datos de las tablas STAFF y ORG para formar una nueva tabla. Para unir dos tablas, especifique en la cláusula SELECT las columnas que desea que se visualicen, indique los nombres de las tablas en una cláusula FROM y especifique la condición de búsqueda en la cláusula WHERE. La cláusula WHERE es opcional.

El ejemplo siguiente asocia el nombre de cada jefe de departamento con un nombre de departamento. Es necesario seleccionar información de dos tablas, pues la información sobre empleados (tabla STAFF) y la información sobre departamentos (tabla ORG) están almacenadas por separado. La consulta

siguiente selecciona las columnas NAME y DEPTNAME de las tablas STAFF y ORG, respectivamente. La condición de búsqueda limita la selección a las filas donde los valores de la columna MANAGER son los mismos que los valores de la columna ID:

```
SELECT DEPTNAME, NAME
FROM ORG, STAFF
WHERE MANAGER = ID
```

La Figura 3 muestra cómo se comparan columnas de dos tablas diferentes. Los valores contenidos en recuadrados indican una coincidencia con la condición de búsqueda.

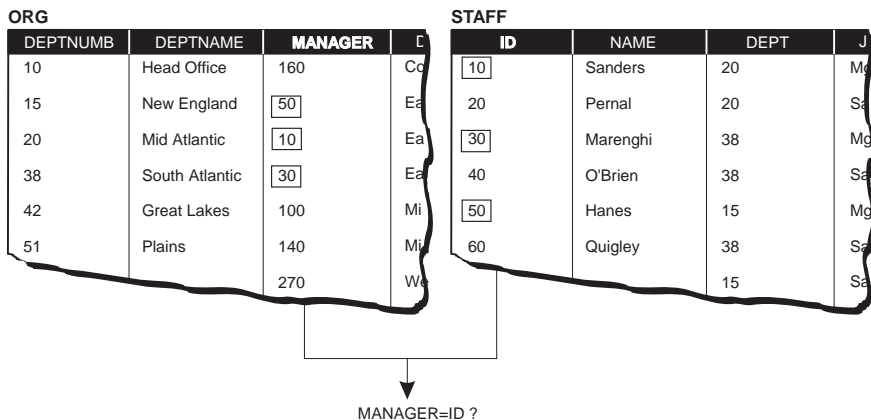


Figura 3. Selección en las tablas STAFF y ORG

La sentencia SELECT produce el resultado siguiente:

DEPTNAME	NAME
Mid Atlantic	Sanders
South Atlantic	Marenghi
New England	Hanes
Great Lakes	Plotz
Plains	Fraye
Head Office	Molinare
Pacific	Lea
Mountain	Quill

El resultado lista el nombre cada jefe y su departamento.

Utilización de una subconsulta

Cuando escribe una sentencia SELECT de SQL, puede colocar sentencias SELECT adicionales dentro de la cláusula WHERE. Cada SELECT adicional inicia una subconsulta.

A su vez, una subconsulta puede incluir otra subconsulta separada, cuyo resultado se coloca en la cláusula WHERE de la subconsulta original. Además, una cláusula WHERE puede incluir subconsultas en más de una condición de búsqueda. La subconsulta puede hacer referencia a tablas y columnas que son diferentes de las utilizadas en la consulta principal.

La sentencia siguiente selecciona en la tabla ORG la división y la ubicación del empleado cuyo ID en la tabla STAFF es 280:

```
SELECT DIVISION, LOCATION
FROM ORG
WHERE DEPTNUMB = (SELECT DEPT
                  FROM STAFF
                  WHERE ID = 280)
```

Cuando DB2 procesa una sentencia, primero determina el resultado de la subconsulta. En este ejemplo, el resultado de la subconsulta es 66, pues el empleado cuyo ID es 280 está en el departamento 66. A continuación, el resultado final se toma de la fila de la tabla ORG cuyo valor de columna para DEPTNAME es 66. El resultado final es:

```
DIVISION  LOCATION
-----  -
Western   San Francisco
```

Cuando se utiliza una subconsulta, el gestor de bases de datos la evalúa y coloca directamente el valor resultante en el cláusula WHERE.

En “Subconsultas correlacionadas” en la página 41 se dan más detalles sobre las subconsultas.

Utilización de funciones

Esta sección proporciona una breve información básica sobre las funciones que se utilizarán en los ejemplos a lo largo de todo el manual. Una *función de base de datos* es una relación entre un conjunto de valores de datos de entrada y un valor resultante.

Las funciones pueden estar *incorporadas* o *definidas por el usuario*. DB2 Universal Database proporciona muchas funciones incorporadas y funciones preinstaladas definidas por el usuario.

Las funciones incorporadas están el esquema SYSIBM; y las funciones preinstaladas definidas por el usuario se encuentran en el esquema SYSFUN. SYSIBM y SYSFUN son esquemas reservados.

Las funciones incorporadas y las funciones preinstaladas definidas por el usuario no satisfacen nunca todas las necesidades del usuario. Por lo tanto, los desarrolladores de aplicaciones pueden necesitar crear su propia colección

de funciones, destinadas específicamente a sus aplicaciones. Las funciones definidas por el usuario hacen que esto sea posible, ampliando el ámbito de DB2 Universal Database para incluir, por ejemplo, funciones personalizadas de gestión o científicas. En “Funciones definidas por el usuario” en la página 68 se proporciona más información sobre este tema.

Funciones de columna

Las *funciones de columna* actúan sobre un conjunto de valores de una columna y obtienen un valor resultante individual. A continuación siguen algunos ejemplos de funciones de columna. Para obtener una lista completa, consulte la publicación *Consulta de SQL*.

AVG	Devuelve la suma de un conjunto de valores dividida por el número de valores de ese conjunto
COUNT	Devuelve el número de filas o valores de un conjunto de filas o valores.
MAX	Devuelve el valor más alto de un conjunto de valores.
MIN	Devuelve el valor más bajo de un conjunto de valores.

La sentencia siguiente selecciona el salario máximo en la tabla STAFF:

```
SELECT MAX(SALARY)
FROM STAFF
```

Esta sentencia devuelve el valor 22959.20 de la tabla de ejemplo STAFF.

El ejemplo siguiente selecciona los nombres y salarios de los empleados cuyo salario es mayor que el salario promedio, pero han estado trabajando en la empresa un tiempo menor que el número promedio de años.

```
SELECT NAME, SALARY
FROM STAFF
WHERE SALARY > (SELECT AVG(SALARY) FROM STAFF)
AND YEARS < (SELECT AVG(YEARS) FROM STAFF)
```

Esta sentencia produce el resultado siguiente:

```
NAME      SALARY
-----
Marenghi  17506.75
Daniels   19260.25
Gonzales  16858.20
```

En la cláusula WHERE del ejemplo anterior, la función de columna está especificada en una *subconsulta*, en lugar de estar aplicada directamente (por ejemplo: WHERE SALARY > AVG(SALARY)). Las funciones de columna no se pueden especificar en la cláusula WHERE. Esto es debido al orden de las operaciones. Considere que la cláusula WHERE se evalúa antes que la cláusula SELECT. En consecuencia, cuando se evalúa la cláusula WHERE, la

función de columna no tiene acceso al conjunto de valores. La cláusula `SELECT` selecciona este conjunto de valores en un momento posterior.

Puede utilizar la palabra clave `DISTINCT` como parte del argumento de una función de columna para eliminar los valores duplicados antes de aplicar una función. De este modo, `COUNT(DISTINCT WORKDEPT)` calcula el número de departamentos diferentes.

Funciones escalares

Una *función escalar* realiza una operación sobre un valor individual y devuelve otro valor individual. A continuación siguen algunos ejemplos de funciones escalares proporcionadas por DB2 Universal Database.

ABS	Devuelve el valor absoluto de un número.
HEX	Devuelve la representación hexadecimal de un valor.
LENGTH	Devuelve el número de bytes de un argumento (en el caso de una serie gráfica de caracteres devuelve el número de caracteres de doble byte.)
YEAR	Extrae la porción del año en un valor de fecha y hora.

Para obtener una lista y descripción detalladas de las funciones escalares, consulte el manual *Consulta de SQL*.

La sentencia siguiente devuelve los nombres de departamento de la tabla `ORG` y la longitud de cada uno de estos nombres:

```
SELECT DEPTNAME, LENGTH(DEPTNAME)
FROM ORG
```

Esta sentencia produce el resultado siguiente:

DEPTNAME	2
Head Office	11
New England	11
Mid Atlantic	12
South Atlantic	14
Great Lakes	11
Plains	6
Pacific	7
Mountain	8

Nota: Debido a que no se utilizó la cláusula `AS` para dar un nombre significativo a `LENGTH(DEPTNAME)`, en la segunda columna aparece un número generado por el sistema.

Funciones de tabla

Las *funciones de tabla* devuelven columnas de una tabla, obteniendo una tabla similar a la creada por una sentencia `CREATE TABLE` sencilla.

Las funciones de tabla sólo se pueden utilizar en la cláusula FROM de una sentencia SQL.

Actualmente, la única función de tabla que puede utilizarse en DB2 Universal Database es SQLCACHE_SNAPSHOT.

SQLCACHE_SNAPSHOT

Devuelve en forma de tabla los resultados de una instantánea de la antememoria de sentencias del SQL dinámico de DB2.

Agrupación

DB2 Universal Database tiene la capacidad de analizar datos basándose en determinadas columnas de una tabla.

El usuario puede organizar las filas de una tabla de acuerdo con una estructura de agrupación definida en una cláusula GROUP BY. En su forma más simple, un *grupo* es un conjunto de filas, cada una de las cuales tiene el mismo valor en las columnas especificadas por "GROUP BY". Los nombres de columnas indicados en la cláusula SELECT deben ser una columna de agrupación o una función de columna. Las funciones de columna devuelven un valor para cada grupo definido por la cláusula GROUP BY. Cada grupo está representado por una fila individual en el conjunto resultante. El ejemplo siguiente lista el salario máximo de cada número de departamento:

```
SELECT DEPT, MAX(SALARY) AS MAXIMUM
FROM STAFF
GROUP BY DEPT
```

Esta sentencia produce el resultado siguiente:

DEPT	MAXIMUM
10	22959.20
15	20659.80
20	18357.50
38	18006.00
42	18352.80
51	21150.00
66	21000.00
84	19818.00

Observe que MAX(SALARY) se calcula para cada departamento, que es un grupo definido por la cláusula GROUP BY, no para la empresa completa.

Utilización de una cláusula WHERE con una cláusula GROUP BY

Una consulta de agrupación puede utilizar una cláusula WHERE convencional que suprime las filas que no reúnen los requisitos antes de que se formen los grupos y se calculen las funciones de columna. Es necesario especificar la cláusula WHERE *antes* que la cláusula GROUP BY. Por ejemplo:

```

SELECT WORKDEPT, EDLEVEL, MAX(SALARY) AS MAXIMUM
FROM EMPLOYEE
WHERE HIREDATE > '1979-01-01'
GROUP BY WORKDEPT, EDLEVEL
ORDER BY WORKDEPT, EDLEVEL

```

El resultado es:

WORKDEPT	EDLEVEL	MAXIMUM
D11	17	18270.00
D21	15	27380.00
D21	16	36170.00
D21	17	28760.00
E11	12	15340.00
E21	14	26150.00

Observe que cada nombre de columna especificado en la sentencia SELECT también aparece en la cláusula GROUP BY. Si no se mencionan los nombres de columna en ambos lugares, se obtiene un error. La cláusula GROUP BY devuelve una fila para cada combinación exclusiva de WORKDEPT y EDLEVEL.

Utilización de la cláusula HAVING después de la cláusula GROUP BY

Puede aplicar una condición restrictiva a los grupos, de forma que DB2 devuelva un resultado sólo para los grupos que cumplen la condición. Para hacer esto, incluya una cláusula HAVING *después* de la cláusula GROUP BY. Una cláusula HAVING puede contener uno o más predicados unidos por los operadores AND y OR. Cada predicado compara una propiedad del grupo (tal como AVG(SALARY)) con uno de estos elementos:

- Otra propiedad del grupo

Por ejemplo:

```
HAVING AVG(SALARY) > 2 * MIN(SALARY)
```

- Una constante

Por ejemplo:

```
HAVING AVG(SALARY) > 20000
```

Por ejemplo, la consulta siguiente busca el salario máximo y mínimo de los departamentos con más de 4 empleados:

```

SELECT WORKDEPT, MAX(SALARY) AS MAXIMUM, MIN(SALARY) AS MINIMUM
FROM EMPLOYEE
GROUP BY WORKDEPT
HAVING COUNT(*) > 4
ORDER BY WORKDEPT

```

Esta sentencia produce el resultado siguiente:

WORKDEPT	MAXIMUM	MINIMUM
D11	32250.00	18270.00
D21	36170.00	17250.00
E11	29750.00	15340.00

Es posible (pero no habitual) que una consulta tenga una cláusula HAVING, pero ninguna cláusula GROUP BY. En este caso, DB2 trata la tabla completa como un solo grupo. Debido a que la tabla es tratada como un grupo individual, el resultado contendrá una fila como máximo. Si la condición HAVING se cumple para la tabla completa, se devuelve el resultado seleccionado (que debe constar sólo de funciones de columna); de lo contrario, no se devuelve ninguna fila.

Capítulo 5. Expresiones y subconsultas

DB2 proporciona flexibilidad para expresar subconsultas. Este capítulo describe algunos métodos que pueden utilizarse para expresar subconsultas complejas.

Este capítulo proporciona una descripción completa de lo siguiente:

- Selecciones completas escalares
- Conversión de tipos de datos
- Expresiones CASE
- Expresiones de tabla
- Nombres de correlación

Selecciones completas escalares

Una selección completa es una modalidad de consulta que se puede utilizar en sentencias SQL. Una selección completa escalar es una selección completa que devuelve una fila individual que contiene un solo valor. Las selecciones completas escalares son útiles para recuperar valores de datos en una base de datos y utilizarlos en una expresión.

- El ejemplo siguiente lista nombres de empleados que tienen un salario mayor que el salario promedio de todos los empleados. La selección completa escalar incluida en la consulta es la sentencia SELECT encerrada entre paréntesis.

```
SELECT LASTNAME, FIRSTNAME
FROM EMPLOYEE
WHERE SALARY > (SELECT AVG(SALARY)
FROM EMPLOYEE)
```

- Este ejemplo busca el salario promedio de los empleados en la tabla STAFF y en la tabla EMPLOYEE.

```
SELECT AVG(SALARY) AS "Average_Employee",
(SELECT AVG(SALARY) AS "Average_Staff" FROM STAFF)
FROM EMPLOYEE
```

Conversión de tipos de datos

A veces puede necesitar convertir valores de un tipo de datos a otro; por ejemplo, convertir un valor numérico en una serie de caracteres. Para convertir un valor a un tipo diferente, utilice la especificación CAST.

Otra finalidad posible de una especificación CAST es truncar una serie de caracteres muy larga. En la tabla EMP_RESUME, la columna RESUME es CLOB(5K). Puede desear visualizar sólo los primeros 370 caracteres que contienen la información personal del candidato. Para visualizar los primeros 370 caracteres del formato ASCII de los curriculums de la tabla EMP_RESUME, emita la consulta siguiente:

```
SELECT EMPNO, CAST(RESUME AS VARCHAR(370))
FROM EMP_RESUME
WHERE RESUME_FORMAT = 'ascii'
```

Se emite un aviso para informarle de que los valores mayores que 370 caracteres están truncados.

Puede convertir valores nulos en otros tipos de datos que sean más apropiados para su manejo en una consulta. La sección “Expresiones de tabla comunes” en la página 39 es un ejemplo del uso de la conversión de datos con esta finalidad.

Expresiones CASE

Puede utilizar expresiones CASE en sentencias SQL para manejar fácilmente la representación de datos de una tabla. Esto proporciona una potente herramienta para representar expresiones condicionales, que conceptualmente es similar a las sentencias CASE utilizadas en algunos lenguajes de programación.

- La consulta siguiente cambia los números de departamento de la columna NOMBREDEPT de la tabla ORG por palabras significativas:

```
SELECT DEPTNAME,
CASE DEPTNUMB
WHEN 10 THEN 'Marketing'
WHEN 15 THEN 'Research'
WHEN 20 THEN 'Development'
WHEN 38 THEN 'Accounting'
ELSE 'Sales'
END AS FUNCTION
FROM ORG
```

El resultado es:

DEPTNAME	FUNCTION
Head Office	Marketing
New England	Research
Mid Atlantic	Development
South Atlantic	Accounting
Great Lakes	Sales
Plains	Sales
Pacific	Sales
Mountain	Sales

- Puede utilizar expresiones CASE para evitar excepciones tales como la división por cero. En el ejemplo siguiente, si el empleado no recibe ninguna prima ni comisión, la condición de la sentencia evita un error al impedir la operación de división:

```
SELECT LASTNAME, WORKDEPT FROM EMPLOYEE
WHERE(CASE
      WHEN BONUS+COMM=0 THEN NULL
      ELSE SALARY/(BONUS+COMM)
      END ) > 10
```

- Puede utilizar una expresión CASE para obtener el cociente entre la suma de un subconjunto de valores de un columna y la suma de todos los valores de esa columna. Este cociente puede estar contenido en una sentencia individual que hace uso de una expresión CASE; esta expresión requiere una sola pasada de proceso por los datos. Si no se utiliza una expresión CASE, como mínimo son necesarios dos pasadas para realizar el mismo cálculo.

El ejemplo siguiente calcula el cociente entre la suma de los salarios del departamento 20 y la suma de todos los salarios, utilizando para ello una expresión CASE:

```
SELECT CAST(CAST (SUM(CASE
                    WHEN DEPT = 20 THEN SALARY
                    ELSE 0
                    END) AS DECIMAL(7,2))/
          SUM(SALARY) AS DECIMAL (3,2))
FROM STAFF
```

El resultado es 0.11. Observe que las funciones CAST preservan la precisión del resultado.

- Puede utilizar una expresión CASE para evaluar una función simple en lugar de invocar a la propia función, lo cual requeriría más actividad general de proceso. Por ejemplo:

```
CASE
  WHEN X<0 THEN -1
  WHEN X=0 THEN 0
  WHEN X>0 THEN 1
END
```

Esta expresión tiene el mismo resultado que la función definida por el usuario SIGN, perteneciente al esquema SYSFUN.

Expresiones de tabla

Si solo necesita la definición de una vista para una consulta individual, puede utilizar una *expresión de tabla*.

Las expresiones de tabla son temporales y sólo tienen validez durante la ejecución de la sentencia SQL; no se pueden compartir como las vistas, pero permiten una mayor flexibilidad que las vistas.

Esta sección describe cómo utilizar expresiones de tabla comunes y expresiones de tabla anidadas en consultas.

Expresiones de tabla anidadas

Una expresión de tabla anidada es una vista temporal donde la definición está *anidada* (definida directamente) en la cláusula FROM de la consulta principal.

La sentencia siguiente utiliza una expresión de tabla anidada para buscar la paga total promedio, el nivel de educación y el año de contratación, correspondientes a los empleados con un nivel de educación mayor que 16:

```
SELECT EDLEVEL, HIREYEAR, DECIMAL(AVG(TOTAL_PAY),7,2)
      FROM (SELECT EDLEVEL, YEAR(HIREDATE) AS HIREYEAR,
                  SALARY+BONUS+COMM AS TOTAL_PAY
            FROM EMPLOYEE
            WHERE EDLEVEL > 16) AS PAY_LEVEL
GROUP BY EDLEVEL, HIREYEAR
ORDER BY EDLEVEL, HIREYEAR
```

El resultado es el siguiente:

EDLEVEL	HIREYEAR	3
17	1967	28850.00
17	1973	23547.00
17	1977	24430.00
17	1979	25896.50
18	1965	57970.00
18	1968	32827.00
18	1973	45350.00
18	1976	31294.00
19	1958	51120.00
20	1975	42110.00

Esta consulta utiliza una expresión de tabla anidada para primero obtener el año de contratación en la columna HIREDATE, para que luego se pueda utilizar en la cláusula GROUP BY. Puede que no desee crear esto en forma de vista, si piensa realizar consultas similares utilizando valores diferentes de EDLEVEL.

En este ejemplo se utiliza la función incorporada escalar DECIMAL. DECIMAL devuelve la representación decimal de un número o serie de caracteres. Para obtener más detalles sobre funciones, consulte el manual *Consulta de SQL*.

Expresiones de tabla comunes

Una *expresión de tabla común* es una expresión de tabla que se crea para utilizarla a lo largo de toda una consulta compleja. La expresión de tabla común se define y recibe un nombre al principio de la consulta, utilizando una cláusula WITH. Las referencias repetidas a una expresión de tabla común utilizan el mismo conjunto resultante. En cambio, si utilizara expresiones de tabla anidadas o vistas, el conjunto resultante se regeneraría cada vez, con resultados posiblemente diferentes.

El ejemplo siguiente lista todos los empleados de la empresa que tienen un nivel de educación mayor que 16, y que reciben una paga promedio menor que las personas contratadas en el mismo año y con el mismo nivel de educación. Las partes de la consulta se describen con más detalle a continuación de la consulta.

1

```
WITH
  PAYLEVEL AS
    (SELECT EMPNO, EDLEVEL, YEAR(HIREDATE) AS HIREYEAR,
      SALARY+BONUS+COMM AS TOTAL_PAY
     FROM EMPLOYEE
     WHERE EDLEVEL > 16),
```

2

```
  PAYBYED (EDUC_LEVEL, YEAR_OF_HIRE, AVG_TOTAL_PAY) AS
    (SELECT EDLEVEL, HIREYEAR, AVG(TOTAL_PAY)
     FROM PAYLEVEL
     GROUP BY EDLEVEL, HIREYEAR)
```

3

```
SELECT EMPNO, EDLEVEL, YEAR_OF_HIRE, TOTAL_PAY, DECIMAL(AVG_TOTAL_PAY,7,2)
FROM PAYLEVEL, PAYBYED
WHERE EDLEVEL = EDUC_LEVEL
      AND HIREYEAR = YEAR_OF_HIRE
      AND TOTAL_PAY < AVG_TOTAL_PAY
```

1

Esta es una expresión de tabla común llamada PAYLEVEL. La tabla resultante incluye un número de empleado, el año en que se contrató a la persona, su paga total y su nivel de educación. Sólo se seleccionan las filas correspondientes a los empleados con un nivel de educación mayor que 16.

2

Esta es una expresión de tabla común llamada PAYBYED (es decir, paga según nivel de educación). Esta expresión utiliza la tabla PAYLEVEL, creada en la expresión de tabla común anterior, para determinar el nivel de educación, el año de contratación y la paga promedio de los empleados dentro de cada nivel de educación, contratados en el mismo año. Las columnas devueltas por esta tabla

tienen nombres diferentes (por ejemplo, EDUC_LEVEL) que los nombres de columna utilizados en la lista de selección. Esto produce un conjunto resultante llamado PAYBYED que es igual que el resultado generado en el ejemplo de Expresión de Tabla Anidada.

- Finalmente, llegamos a la consulta propiamente dicha que produce el resultado deseado. Las dos tablas (PAYLEVEL, PAYBYED) están unidas para determinar los empleados cuya paga total es menor que la paga promedio de las personas contratadas en el mismo año. Observe que PAYBYED está basado en PAYLEVEL. Por tanto, en realidad se accede a PAYLEVEL dos veces en la sentencia completa. En ambas ocasiones se utiliza el mismo conjunto de filas para evaluar la consulta.

El resultado es el siguiente:

EMPNO	EDLEVEL	YEAR_OF_HIRE	TOTAL_PAY	5
000210	17	1979	20132.00	25896.50

Nombres de correlación

Un *nombre de correlación* es un identificador que se utiliza para distinguir los diversos usos de un objeto. Un nombre de correlación se puede definir en la cláusula FROM de una consulta y en la primera cláusula de una sentencia UPDATE o DELETE. Se puede asociar a una tabla, vista o a una expresión de tabla anidada, pero sólo dentro del contexto en el que está definido.

Por ejemplo, la cláusula FROM STAFF S, ORG O establece S y O como nombres de correlación para STAFF y ORG, respectivamente.

```
SELECT NAME, DEPTNAME
FROM STAFF S, ORG O
WHERE O.MANAGER = S.ID
```

Una vez definido un nombre de correlación, *sólo* puede utilizarlo para calificar el objeto. Por ejemplo, en el ejemplo anterior, si se especificara ORG.MANAGER=STAFF.ID, se obtendría un error.

También puede utilizar un nombre de correlación como nombre abreviado para hacer referencia a un objeto de base de datos. Escribir sólo una S es más sencillo que escribir STAFF.

Mediante nombres de correlación, puede crear **duplicados** de un objeto. Esto es útil si necesita comparar entradas de una tabla con ella misma. En el ejemplo siguiente, la tabla EMPLOYEE se compara con otra instancia de ella

misma para determinar los jefes de todos los empleados. El ejemplo visualiza el nombre de los empleados que no son diseñadores, el nombre de su jefe y el número de departamento.

```
SELECT E2.FIRSTNME, E2.LASTNAME, E2.JOB, E1.FIRSTNME AS MGR_FIRSTNAME,
       E1.LASTNAME AS MGR_LASTNAME, E1.WORKDEPT
FROM EMPLOYEE E1, EMPLOYEE E2
WHERE E1.WORKDEPT = E2.WORKDEPT
      AND E1.JOB = 'MANAGER'
      AND E2.JOB <> 'MANAGER'
      AND E2.JOB <> 'DESIGNER'
```

Esta sentencia produce el resultado siguiente:

FIRSTNME	LASTNAME	JOB	MGR_FIRSTNAME	MGR_LASTNAME	WORKDEPT
DOLORES	QUINTANA	ANALYST	SALLY	KWAN	C01
HEATHER	NICHOLLS	ANALYST	SALLY	KWAN	C01
JAMES	JEFFERSON	CLERK	EVA	PULASKI	D21
SALVATORE	MARINO	CLERK	EVA	PULASKI	D21
DANIEL	SMITH	CLERK	EVA	PULASKI	D21
SYBIL	JOHNSON	CLERK	EVA	PULASKI	D21
MARIA	PEREZ	CLERK	EVA	PULASKI	D21
ETHEL	SCHNEIDER	OPERATOR	EILEEN	HENDERSON	E11
JOHN	PARKER	OPERATOR	EILEEN	HENDERSON	E11
PHILIP	SMITH	OPERATOR	EILEEN	HENDERSON	E11
MAUDE	SETRIGHT	OPERATOR	EILEEN	HENDERSON	E11
RAMLAL	MEHTA	FIELDREP	THEODORE	SPENSER	E21
WING	LEE	FIELDREP	THEODORE	SPENSER	E21
JASON	GOUNOT	FIELDREP	THEODORE	SPENSER	E21

Subconsultas correlacionadas

Una *subconsulta correlacionada* es una subconsulta que puede hacer referencia a cualquiera de las tablas mencionadas anteriormente. También se dice que la subconsulta tiene una *referencia correlacionada* a una tabla de la consulta principal.

El ejemplo siguiente utiliza una subconsulta no correlacionada para listar el número de empleado y el nombre de los empleados del departamento 'A00' cuyo salario es mayor que el salario promedio del departamento:

```
SELECT EMPNO, LASTNAME
FROM EMPLOYEE
WHERE WORKDEPT = 'A00'
      AND SALARY > (SELECT AVG(SALARY)
                   FROM EMPLOYEE
                   WHERE WORKDEPT = 'A00')
```

Esta sentencia produce el resultado siguiente:

```

EMPNO  LASTNAME
-----  -----
000010  HAAS
000110  LUCCHESI

```

Si desea conocer el salario promedio de cada departamento, es necesario evaluar la subconsulta una sola vez para cada departamento. Puede hacer esto utilizando la capacidad de correlación del SQL, que le permite escribir una subconsulta y ejecutarla varias veces, una vez para cada tabla identificada en la consulta de nivel exterior.

El ejemplo siguiente utiliza una subconsulta correlacionada para listar todos los empleados cuyo salario es mayor que el salario promedio de sus respectivos departamentos:

```

SELECT E1.EMPNO, E1.LASTNAME, E1.WORKDEPT
FROM EMPLOYEE E1
WHERE SALARY > (SELECT AVG(SALARY)
                FROM EMPLOYEE E2
                WHERE E2.WORKDEPT = E1.WORKDEPT)
ORDER BY E1.WORKDEPT

```

En esta consulta, la subconsulta se evalúa una sola vez para cada departamento. El resultado es:

```

EMPNO  LASTNAME  WORKDEPT
-----  -----  -----
000010  HAAS      A00
000110  LUCCHESI  A00
000030  KWAN      C01
000060  STERN     D11
000150  ADAMSON   D11
000170  YOSHIMURA D11
000200  BROWN     D11
000220  LUTZ      D11
000070  PULASKI   D21
000240  MARINO    D21
000270  PEREZ     D21
000090  HENDERSON E11
000280  SCHNEIDER E11
000100  SPENSER   E21
000330  LEE       E21
000340  GOUNOT    E21

```

Para escribir una consulta con una subconsulta correlacionada, utilice el mismo formato básico de una consulta externa ordinaria con una subconsulta. Sin embargo, en la cláusula FROM de la consulta externa, justo a continuación del nombre de tabla, coloque un nombre de correlación. Entonces la subconsulta puede contener referencias a columna calificadas por el nombre de correlación. Por ejemplo, si E1 es un nombre de correlación, entonces E1.WORKDEPT representa el valor WORKDEPT de la fila actual de la tabla

en la consulta externa. La subconsulta se reevalúa (conceptualmente) para cada fila de la tabla en la consulta externa.

Cuando se utiliza una subconsulta correlacionada, el sistema libera de trabajo al usuario y disminuye la cantidad de código que es necesario escribir en la aplicación.

DB2 permite utilizar referencias correlacionadas no calificadas. Por ejemplo, la tabla EMPLOYEE tiene una columna llamada LASTNAME, en cambio la tabla SALES tiene una columna llamada SALES_PERSON, y ninguna columna llamada LASTNAME.

```
SELECT LASTNAME, FIRSTNAME, COMM
FROM EMPLOYEE
WHERE 3 > (SELECT AVG(SALES)
FROM SALES
WHERE LASTNAME = SALES_PERSON)
```

En este ejemplo, el sistema busca una columna LASTNAME en la cláusula FROM más interna. Al no encontrar ninguna, examina la siguiente cláusula FROM más interna (que en este caso es la cláusula FROM externa). Aunque no siempre es necesario, es recomendable calificar las referencias correlacionadas para mejorar la legibilidad de la consulta y asegurar que se obtiene el resultado deseado.

Implementación de una subconsulta correlacionada

¿Cuándo es necesario utilizar una subconsulta correlacionada? El uso de una función de columna es a veces una pista.

Supongamos que desea listar los empleados cuyo nivel de educación es mayor que el nivel promedio de sus respectivos departamentos.

Primero, debe determinar los elementos de la lista de selección. El problema consiste en “Listar los empleados”. Esto significa que LASTNAME de la tabla EMPLOYEE debería ser suficiente para identificar de forma inequívoca a los empleados. El problema también indica como condiciones el nivel de educación (EDLEVEL) y los departamentos de los empleados (WORKDEPT). Aunque el problema no solicita explícitamente que se visualicen columnas, el incluirlas en la lista de selección ayudará a mostrar la solución. Ahora se puede escribir parte de la consulta:

```
SELECT LASTNAME, WORKDEPT, EDLEVEL
FROM EMPLOYEE
```

A continuación, es necesaria una condición de búsqueda (cláusula WHERE). El enunciado del problema dice “...cuyo nivel de educación es mayor que el nivel promedio de sus respectivos departamentos”. Esto significa que para cada empleado de la tabla, se debe calcular el nivel promedio de educación del departamento de ese empleado. Este enunciado se ajusta a la descripción

de una subconsulta correlacionada. Cierta propiedad desconocida (el nivel promedio de educación del departamento actual del empleado) se calcula para cada fila. Es necesario un nombre de correlación para la tabla EMPLOYEE:

```
SELECT LASTNAME, WORKDEPT, EDLEVEL
FROM EMPLOYEE E1
```

La subconsulta necesaria es sencilla. Esta subconsulta calcula el nivel promedio de educación para cada departamento. La sentencia SQL completa es:

```
SELECT LASTNAME, WORKDEPT, EDLEVEL
FROM EMPLOYEE E1
WHERE EDLEVEL > (SELECT AVG(EDLEVEL)
FROM EMPLOYEE E2
WHERE E2.WORKDEPT = E1.WORKDEPT)
```

El resultado es:

LASTNAME	WORKDEPT	EDLEVEL
HAAS	A00	18
KWAN	C01	20
PULASKI	D21	16
HENDERSON	E11	16
LUCCHESSI	A00	19
PIANKA	D11	17
SCOUTTEN	D11	17
JONES	D11	17
LUTZ	D11	18
MARINO	D21	17
JOHNSON	D21	16
SCHNEIDER	E11	17
MEHTA	E21	16
GOUNOT	E21	16

Suponga que en lugar de listar el número de departamento del empleado, desea listar el nombre del departamento. La información necesaria (DEPTNAME) está en una tabla separada (DEPARTMENT). La consulta de nivel externo que define una variable de correlación también puede ser una consulta de unión (vea “Selección de datos de más de una tabla” en la página 26 para obtener detalles).

Cuando utilice uniones en una consulta de nivel externo, liste en la cláusula FROM las tablas a unir y coloque el nombre de correlación junto al nombre de tabla apropiado.

Para modificar la consulta a fin de que liste el nombre del departamento en lugar de su número, sustituya WORKDEPT por DEPTNAME en la lista de selección. Ahora la cláusula FROM también debe incluir la tabla DEPARTMENT y la cláusula WHERE debe expresar la condición de unión apropiada.

Esta es la consulta modificada:

```
SELECT LASTNAME, DEPTNAME, EDLEVEL
FROM EMPLOYEE E1, DEPARTMENT
WHERE E1.WORKDEPT = DEPARTMENT.DEPTNO
AND EDLEVEL > (SELECT AVG(EDLEVEL)
FROM EMPLOYEE E2
WHERE E2.WORKDEPT = E1.WORKDEPT)
```

Esta sentencia produce el resultado siguiente:

LASTNAME	DEPTNAME	EDLEVEL
HAAS	SPIFFY COMPUTER SERVICE DIV.	18
LUCCHESSI	SPIFFY COMPUTER SERVICE DIV.	19
KWAN	INFORMATION CENTER	20
PIANKA	MANUFACTURING SYSTEMS	17
SCOUTTEN	MANUFACTURING SYSTEMS	17
JONES	MANUFACTURING SYSTEMS	17
LUTZ	MANUFACTURING SYSTEMS	18
PULASKI	ADMINISTRATION SYSTEMS	16
MARINO	ADMINISTRATION SYSTEMS	17
JOHNSON	ADMINISTRATION SYSTEMS	16
HENDERSON	OPERATIONS	16
SCHNEIDER	OPERATIONS	17
MEHTA	SOFTWARE SUPPORT	16
GOUNOT	SOFTWARE SUPPORT	16

Los ejemplos anteriores muestran que el nombre de correlación utilizado en una subconsulta debe estar definido en la cláusula FROM de una consulta donde reside la subconsulta correlacionada. Si embargo, esta inclusión de un elemento dentro de otro puede suponer varios niveles de anidamiento.

Suponga que algunos departamentos tienen solo unos pocos empleados y por tanto su nivel promedio de educación puede ser engañoso. Puede decidir que es necesario que haya un mínimo de cinco empleados en un departamento para que el nivel promedio de educación sea un valor significativo con el que poder comparar un empleado. Por tanto, ahora tenemos que listar los empleados cuyo nivel de educación es mayor que el nivel promedio de sus respectivos departamentos y considerar sólo los departamentos con un mínimo de cinco empleados.

El problema implica otra subconsulta, pues para cada empleado indicado en la consulta de nivel externo, se debe contar el número total de empleados del departamento de esa persona:

```
SELECT COUNT(*)
FROM EMPLOYEE E3
WHERE E3.WORKDEPT = E1.WORKDEPT
```

Sólo se debe calcular un valor promedio si la cuenta de empleados es mayor o igual que 5:

```

SELECT AVG(EDLEVEL)
  FROM EMPLOYEE E2
 WHERE E2.WORKDEPT = E1.WORKDEPT
 AND 5 <= (SELECT COUNT(*)
           FROM EMPLOYEE E3
           WHERE E3.WORKDEPT = E1.WORKDEPT)

```

Finalmente, sólo se incluyen los empleados cuyo nivel de educación es mayor que el nivel promedio para ese departamento:

```

SELECT LASTNAME, DEPTNAME, EDLEVEL
  FROM EMPLOYEE E1, DEPARTMENT
 WHERE E1.WORKDEPT = DEPARTMENT.DEPTNO
 AND EDLEVEL >
 (SELECT AVG(EDLEVEL)
  FROM EMPLOYEE E2
 WHERE E2.WORKDEPT = E1.WORKDEPT
 AND 5 <=
 (SELECT COUNT(*)
  FROM EMPLOYEE E3
 WHERE E3.WORKDEPT = E1.WORKDEPT))

```

Esta sentencia produce el resultado siguiente:

LASTNAME	DEPTNAME	EDLEVEL
PIANKA	MANUFACTURING SYSTEMS	17
SCOUTTEN	MANUFACTURING SYSTEMS	17
JONES	MANUFACTURING SYSTEMS	17
LUTZ	MANUFACTURING SYSTEMS	18
PULASKI	ADMINISTRATION SYSTEMS	16
MARINO	ADMINISTRATION SYSTEMS	17
JOHNSON	ADMINISTRATION SYSTEMS	16
HENDERSON	OPERATIONS	16
SCHNEIDER	OPERATIONS	17

Capítulo 6. Utilización de operadores y predicados en consultas

En DB2 Universal Database puede combinar consultas con diferentes *operadores de conjunto* y construir sentencias condicionales complejas con *predicados cuantificados*.

Este capítulo describe cómo:

- Combinar tablas diferentes mediante los operadores de conjunto UNION, EXCEPT e INTERSECT
- Construir condiciones complejas para consultas con predicados cuantificados. Los predicados básicos se trataron brevemente en “Selección de filas” en la página 20.

Combinación de consultas mediante operadores de conjunto

Los operadores de conjunto UNION, EXCEPT e INTERSECT le permiten combinar dos o más consultas de nivel externo para formar una consulta individual. Cada consulta conectada por estos operadores de conjunto se ejecuta y los resultados individuales se combinan. Cada operador produce un resultado diferente.

Operador UNION

El operador UNION obtiene una tabla resultante combinando otras dos tablas resultantes (por ejemplo, TABLA1 y TABLA2) y eliminando las filas duplicadas de las tablas. Cuando se utiliza ALL con UNION (es decir, UNION ALL), no se eliminan las filas duplicadas. En ambos casos, cada fila de la tabla obtenida es una fila procedente de TABLA1 o TABLA2.

En el ejemplo siguiente del operador UNION, la consulta devuelve los nombres de los empleados cuyo salario es mayor que \$21.000, o que tienen cargos directivos y llevan trabajando menos de 8 años en la empresa:

1

```
SELECT ID, NAME FROM STAFF WHERE SALARY > 21000  
UNION
```

2

```
SELECT ID, NAME FROM STAFF WHERE JOB='Mgr' AND YEARS < 8  
ORDER BY ID
```

Los resultados de las consultas individuales son:

1

ID	NAME
140	Fraye
160	Molinare
260	Jones

2

ID	NAME
10	Sanders
30	Marenghi
100	Plotz
140	Fraye
160	Molinare
240	Daniels

El gestor de bases de datos combina los resultados de ambas consultas, elimina los elementos duplicados y devuelve el resultado final en orden ascendente.

ID	NAME
10	Sanders
30	Marenghi
100	Plotz
140	Fraye
160	Molinare
240	Daniels
260	Jones

Si utiliza la cláusula `ORDER BY` en una consulta junto con cualquier operador de conjunto, debe escribir la cláusula después de la última consulta. El sistema aplica la ordenación al resultado combinado.

Si el nombre de columna es diferente en las dos tablas, la tabla combinada resultante carece de nombres para las columnas correspondientes. En cambio, las columnas están numeradas por el orden en que aparecen. Por tanto, si desea que la tabla resultante esté ordenada, debe especificar el número de columna en la cláusula `ORDER BY`.

Operador EXCEPT

El operador `EXCEPT` obtiene una tabla resultante que incluye todas las filas de `TABLA1`, pero no las de `TABLA2`, y elimina todas las filas duplicadas. Si utiliza `ALL` con `EXCEPT` (`EXCEPT ALL`), no se eliminan las filas duplicadas.

En el ejemplo siguiente del operador EXCEPT, la consulta devuelve los nombres de los empleados cuyo salario es mayor que \$21.000, no ocupan puestos directivos y llevan trabajando 8 años como mínimo en la empresa:

```
SELECT ID, NAME FROM STAFF WHERE SALARY > 21000
EXCEPT
SELECT ID, NAME FROM STAFF WHERE JOB='Mgr' AND YEARS < 8
```

Los resultados de las consultas individuales se muestran en la sección correspondiente a UNION. La sentencia anterior produce el resultado siguiente:

ID	NAME
260	Jones

Operador INTERSECT

El operador INTERSECT obtiene una tabla resultante que incluye todas las filas existentes a la vez en TABLA1 y TABLA2, y elimina todas las filas duplicadas. Si utiliza ALL con INTERSECT (INTERSECT ALL), no se eliminan las filas duplicadas.

En el ejemplo siguiente del operador INTERSECT, la consulta devuelve el nombre y el ID de los empleados cuyo salario es mayor que \$21.000, ocupan puestos directivos y llevan trabajando menos de 8 años en la empresa.

```
SELECT ID, NAME FROM STAFF WHERE SALARY > 21000
INTERSECT
SELECT ID, NAME FROM STAFF WHERE JOB='Mgr' AND YEARS < 8
```

Los resultados de las consultas individuales se muestran en la sección correspondiente a UNION. El resultado de las dos consultas con INTERSECT es:

ID	NAME
140	Fraye
160	Molinare

Cuando utilice los operadores UNION, EXCEPT e INTERSECT, tenga en cuenta lo siguiente:

- Todos los elementos correspondientes de las listas de selección de las consultas utilizadas por los operadores deben ser compatibles. Vea la tabla sobre compatibilidad de los tipos de datos, en el manual *Consulta de SQL*, para obtener más información.
- Si se utiliza una cláusula ORDER BY, debe estar a continuación de la última consulta que tenga un operador de conjunto. El nombre de columna sólo se puede utilizar en la cláusula ORDER BY si el nombre de columna es igual a los elementos correspondientes en la lista de selección de las consultas utilizadas por cada operador.

- Las operaciones entre columnas que tienen el mismo tipo de datos y la misma longitud producen una columna de ese tipo y longitud. Vea en el manual *Consulta de SQL* las reglas sobre los tipos de datos del resultado aplicables a los operadores de conjunto UNION, EXCEPT e INTERSECT.

Predicados

Los predicados le permiten definir condiciones para que sólo se procesen las filas que cumplen esas condiciones. Los predicados básicos se tratan en “Selección de filas” en la página 20. IN, BETWEEN, LIKE, EXISTS y los predicados cuantificados se examinan en la presente sección.

Utilización del predicado IN

Utilice el predicado IN para comparar un valor con varios otros valores. Por ejemplo:

```
SELECT NAME
FROM STAFF
WHERE DEPT IN (20, 15)
```

Este ejemplo es equivalente a:

```
SELECT NAME
FROM STAFF
WHERE DEPT = 20 OR DEPT = 15
```

Puede utilizar los operadores IN y NOT IN cuando una subconsulta devuelve un conjunto de valores. Por ejemplo, la consulta siguiente lista el apellido de los empleados encargados de los proyectos MA2100 y OP2012:

```
SELECT LASTNAME
FROM EMPLOYEE
WHERE EMPNO IN
  (SELECT RESPEMP
   FROM PROJECT
   WHERE PROJNO = 'MA2100'
   OR PROJNO = 'OP2012')
```

La subconsulta se evalúa una sola vez y la lista resultante se coloca directamente en la consulta de nivel externo. Por ejemplo, si la subconsulta anterior selecciona los números de empleado 10 y 330, la consulta de nivel externo se evalúa como si su cláusula WHERE fuera:

```
WHERE EMPNO IN (10, 330)
```

La lista de valores devuelta por la subconsulta puede contener ninguno, uno o más valores.

Utilización del predicado BETWEEN

El predicado BETWEEN compara un valor individual con un rango inclusivo de valores (mencionado en el predicado BETWEEN).

El ejemplo siguiente determina los nombres de los empleados cuyo salario está comprendido entre \$10.000 y \$20.000:

```
SELECT LASTNAME
FROM EMPLOYEE
WHERE SALARY BETWEEN 10000 AND 20000
```

Esto es equivalente a:

```
SELECT LASTNAME
FROM EMPLOYEE
WHERE SALARY >= 10000 AND SALARY <= 20000
```

El ejemplo siguiente determina los nombres de los empleados cuyo salario es menor que \$10.000 o mayor que \$20.000:

```
SELECT LASTNAME
FROM EMPLOYEE
WHERE SALARY NOT BETWEEN 10000 AND 20000
```

Utilización del predicado LIKE

Utilice el predicado LIKE para buscar series de caracteres que tienen una composición determinada. La composición se especifica mediante signos de porcentaje y de subrayado.

- El carácter de subrayado (_) representa un carácter individual cualquiera.
- El signo de porcentaje (%) representa una serie de varios caracteres o la ausencia de caracteres.
- Cualquier otro carácter se representa por sí mismo.

El ejemplo siguiente selecciona los nombres de empleado que tienen siete letras de longitud y comienzan con la letra 'S':

```
SELECT NAME
FROM STAFF
WHERE NAME LIKE 'S _ _ _ _ _ _ _'
```

El ejemplo siguiente selecciona los nombres de empleado que no comienzan con la letra 'S':

```
SELECT NAME
FROM STAFF
WHERE NAME NOT LIKE 'S%'
```

Utilización del predicado EXISTS

Puede utilizar una subconsulta para verificar la *existencia* de una fila que cumple una determinada condición. En este caso, la subconsulta se enlaza con la consulta de nivel externo mediante el predicado EXISTS o NOT EXISTS.

Cuando enlaza una subconsulta con una consulta externa mediante un predicado EXISTS, la subconsulta no devuelve un valor. En lugar de esto, el predicado EXISTS es verdadero si el conjunto resultante de la subconsulta contiene una o más filas, y es falso si no contiene ninguna fila.

El predicado EXISTS a menudo se utiliza con subconsultas correlacionadas. El ejemplo siguiente lista los departamentos que actualmente no tienen entradas en la tabla PROJECT:

```
SELECT DEPTNO, DEPTNAME
FROM DEPARTMENT X
WHERE NOT EXISTS
      (SELECT *
       FROM PROJECT
        WHERE DEPTNO = X.DEPTNO)
ORDER BY DEPTNO
```

Puede conectar los predicados EXISTS y NOT EXISTS a otros predicados utilizando AND y OR en la cláusula WHERE de la consulta de nivel externo.

Predicados cuantificados

Un predicado cuantificado compara un valor con un conjunto de valores. Si una selección completa devuelve más de un valor, debe modificar los operadores de comparación del predicado añadiendo el sufijo ALL, ANY o SOME. Estos sufijos determinan cómo el predicado de nivel externo trata los valores devueltos. En el ejemplo se utiliza el operador de comparación > (los comentarios que siguen son aplicables también a los demás operadores):

expresión > ALL (selección completa)

El predicado es verdadero si la expresión es mayor que cada valor individual devuelto por la selección completa. Si la selección completa no devuelve ningún valor, el predicado es verdadero. El resultado es falso si la relación especificada es falsa para un valor como mínimo. Observe que el predicado cuantificado <>ALL es equivalente al predicado NOT IN.

El ejemplo siguiente utiliza una subconsulta y una comparación > ALL para determinar el nombre y la profesión de los empleados que ganan más que todos los jefes de departamento:

```
SELECT LASTNAME, JOB
FROM EMPLOYEE
WHERE SALARY > ALL
      (SELECT SALARY
       FROM EMPLOYEE
        WHERE JOB='MANAGER')
```

expresión > ANY (selección completa)

El predicado es verdadero si la expresión es mayor que al menos uno de los valores devueltos por la selección completa. Si la selección completa no devuelve ningún valor, el predicado es falso. Observe que el operador cuantificado =ANY es equivalente al predicado IN.

expresión > SOME (selección completa)

SOME es sinónimo de ANY.

Para obtener más información sobre los predicados y operadores, consulte el manual *Consulta de SQL*.

Capítulo 7. SQL avanzado

Este capítulo describe algunas características de DB2 Universal Database que le permiten diseñar consultas de forma más efectiva, y al mismo tiempo adaptarlas a sus necesidades. Los temas de este capítulo suponen que el lector ha asimilado completamente las nociones dadas hasta este momento.

Este capítulo trata los temas siguientes:

- Aplicación de reglas de negocio con restricciones y desencadenantes
- Uniones
- Consultas ROLLUP y CUBE y Consultas recursivas
- Funciones OLAP

Aplicación de reglas de negocio con restricciones y desencadenantes

En el ámbito empresarial, es necesario asegurarse de que determinadas reglas se apliquen siempre. Por ejemplo, un empleado que trabaja en un proyecto debe estar en la lista de nóminas. O bien, quizás es deseable que determinados sucesos se produzcan automáticamente. Por ejemplo, si un vendedor realiza una venta, su comisión se debe aumentar.

Con este fin, DB2 Universal Database proporciona un conjunto útil de métodos:

- Las *restricciones de unicidad* impiden la existencia de valores duplicados en una o más columnas de una tabla.
- Las *restricciones de integridad referencial* aseguran la coherencia de los datos en las tablas especificadas.
- Las *restricciones de comprobación de tabla* son reglas que establecen restricciones respecto a los valores que puede tomar una columna. Las inserciones y actualizaciones fallan si un valor asignado a una columna no cumple las restricciones de comprobación para esa columna.
- Los *desencadenantes* definen un conjunto de acciones que se ejecutan, o disparan, mediante una operación de supresión, inserción o actualización en una tabla especificada. Los desencadenantes se pueden utilizar para escribir en otras tablas, para modificar valores de entrada y para emitir mensajes de alerta.

La primera sección proporciona un resumen conceptual de claves. Posteriormente, se examinan, mediante ejemplos y diagramas, la integridad referencial, las restricciones y los desencadenantes.

Claves

Una *clave* es un conjunto de columnas que se puede utilizar para identificar o acceder a una fila o filas determinadas.

Cuando una clave consta de más de una columna se denomina *clave compuesta*. En una tabla con una clave compuesta, la ordenación de las columnas en la clave compuesta no se corresponde necesariamente a su ordenación en la tabla.

Claves exclusivas

Una *clave exclusiva* se define como una columna (o conjunto de columnas) en la que no hay dos valores iguales. Las columnas de una clave exclusiva no pueden contener valores nulos. El gestor de bases de datos aplica la restricción durante la ejecución de las sentencias INSERT y UPDATE. Una tabla puede contener varias claves exclusivas. Las claves exclusivas son opcionales y pueden estar definidas en sentencias CREATE TABLE o ALTER TABLE.

Claves primarias

Una *clave primaria* es una clave exclusiva que forma parte de la definición de la tabla. Una tabla no puede tener más de una clave primaria, y las columnas de una clave primaria no pueden contener valores nulos. Las claves primarias son opcionales y pueden estar definidas en sentencias CREATE TABLE o ALTER TABLE.

Claves externas

Una *clave externa* se especifica en la definición de una restricción referencial. Una tabla puede tener una o más claves externas o carecer de ellas. El valor de la clave externa compuesta es nulo si cualquier componente del valor es nulo. Las claves externas son opcionales y pueden estar definidas en sentencias CREATE TABLE o ALTER TABLE.

Restricciones de unicidad

Las restricciones de unicidad aseguran que los valores de una clave sean exclusivos dentro de una tabla. Las restricciones de unicidad son opcionales, y puede definir las en una sentencia CREATE TABLE o ALTER TABLE especificando la cláusula PRIMARY KEY o UNIQUE. Por ejemplo, puede definir una restricción de unicidad sobre la columna del número de empleado de una tabla para asegurar que cada empleado tenga un número exclusivo.

Restricciones de integridad referencial

Mediante la definición de restricciones de unicidad y claves externas puede definir relaciones entre tablas y consecuentemente aplicar determinadas reglas de negocio. La combinación de restricciones de clave de exclusiva y de clave externa se denomina habitualmente restricciones de integridad referencial. Una restricción de unicidad referenciada por una clave externa se denomina *clave principal*. Una clave externa hace referencia o está asociada a una clave

principal determinada. Por ejemplo, una norma puede establecer que cada empleado (tabla EMPLOYEE) debe pertenecer a un departamento existente (tabla DEPARTMENT). Por tanto, se define el número de departamento como clave externa en la tabla EMPLOYEE y como clave primaria en la tabla DEPARTMENT. El diagrama siguiente describe visualmente las restricciones de integridad referencial.

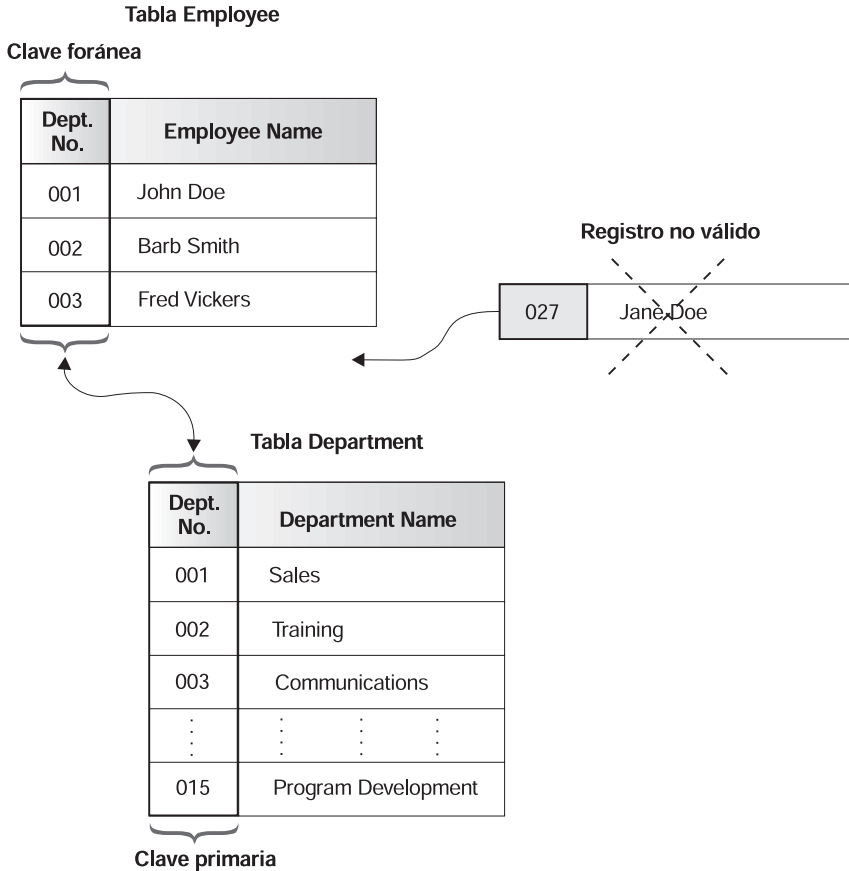


Figura 4. Las restricciones de clave externa y de clave primaria definen relaciones y protegen los datos

Restricciones de comprobación de tabla

Las *restricciones de comprobación de tabla* especifican condiciones que se evalúan para cada fila de una tabla. Puede especificar restricciones de comprobación para columnas individuales. Puede añadir las restricciones utilizando las sentencias CREATE TABLE o ALTER TABLE.

La sentencia siguiente crea una tabla con las restricciones siguientes:

- Los valores del número de departamento deben estar dentro del rango de 10 a 100
- El puesto de trabajo de un empleado sólo puede ser uno de los siguientes: “Sales”, “Mgr” o “Clerk”
- Los empleados contratados antes de 1986 deben tener un salario mayor que \$40.500.

```

CREATE TABLE EMP
  (ID          SMALLINT NOT NULL,
   NAME       VARCHAR(9),
   DEPT       SMALLINT CHECK (DEPT BETWEEN 10 AND 100),
   JOB        CHAR(5) CHECK (JOB IN ('Sales', 'Mgr', 'Clerk')),
   HIREDATE   DATE,
   SALARY     DECIMAL(7,2),
   COMM       DECIMAL(7,2),
   PRIMARY KEY (ID),
   CONSTRAINT YEARSAL CHECK
     (YEAR(HIREDATE) >= 1986 OR SALARY > 40500) )

```

Una restricción se viola sólo si la evaluación de la condición da un resultado falso. Por ejemplo, si DEPT is NULL para una fila insertada, la inserción prosigue sin error, aunque los valores para DEPT deban estar comprendidos entre 10 y 100, tal como está definido en la restricción.

La sentencia siguiente añade una restricción llamada COMP a la tabla EMPLOYEE según la cual la retribución total de un empleado debe ser mayor que \$15.000:

```

ALTER TABLE EMP
  ADD CONSTRAINT COMP CHECK (SALARY + COMM > 15000)

```

Las filas existentes de la tabla se comprobarán para asegurarse de que no violan la nueva restricción. Puede diferir esta comprobación utilizando la sentencia SET CONSTRAINTS del modo siguiente:

```

SET CONSTRAINTS FOR EMP OFF
ALTER TABLE EMP ADD CONSTRAINT COMP CHECK (SALARY + COMM > 15000)
SET CONSTRAINTS FOR EMP IMMEDIATE CHECKED

```

Primero, se utiliza la sentencia SET CONSTRAINTS para diferir la comprobación de la restricción para la tabla. Después, se pueden añadir una o más restricciones a la tabla sin comprobar las restricciones. A continuación se vuelve a emitir la sentencia SET CONSTRAINTS para activar de nuevo la comprobación de restricción y realizar la comprobación diferida de restricción que sea necesaria.

Desencadenantes

Un *desencadenante* define un conjunto de acciones. Los desencadenantes se activan por operaciones que modifican los datos de una tabla base especificada.

Estas son algunas aplicaciones de los desencadenantes:

- validar datos de entrada
- generar automáticamente un valor para una fila recién insertada
- leer en otras tablas para establecer referencias cruzadas
- escribir en otras tablas para realizar seguimientos de auditoría
- dar soporte a alertas mediante mensajes de correo electrónico

El uso de desencadenantes permite un desarrollo más rápido de las aplicaciones, una aplicación global de las reglas de negocio y un mantenimiento más fácil de las aplicaciones y los datos.

DB2 Universal Database da soporte a varios tipos de desencadenantes. Los desencadenantes se pueden definir para que actúen antes o después de una operación DELETE, INSERT o UPDATE. Cada desencadenante incluye un conjunto de sentencias SQL llamado *acción desencadenada* que puede contener una condición opcional de búsqueda.

Los *desencadenantes posteriores* se pueden definir con más detalle para que la acción desencadenada se ejecute para cada fila o una sola vez para la sentencia. Los *desencadenantes anteriores* siempre ejecutan la acción desencadenada para cada fila.

Utilice un desencadenante precediendo a una sentencia INSERT, UPDATE o DELETE para comprobar la existencia de determinadas condiciones antes de realizar una operación desencadenante o para cambiar los valores de entrada antes de que se almacenen en la tabla.

Utilice un desencadenante posterior para propagar valores según sea necesario o para realizar otras tareas, tales como enviar un mensaje, que pueden ser necesarias como parte de la operación de activación.

El ejemplo siguiente muestra una utilización de los desencadenantes anteriores y posteriores. Considere una aplicación que registra y hace un seguimiento de los cambios en la cotización de las acciones. La base de datos contiene dos tablas, CURRENTQUOTE (cotización actual) y QUOTEHISTORY (historia de cotizaciones), que están definidas del modo siguiente:

```
CREATE TABLE CURRENTQUOTE
(SYMBOL VARCHAR(10),
QUOTE DECIMAL(5,2),
STATUS VARCHAR(9))
```

```
CREATE TABLE QUOTEHISTORY
(SYMBOL VARCHAR(10),
QUOTE DECIMAL(5,2),
TIMESTAMP TIMESTAMP)
```

Cuando la columna QUOTE de CURRENTQUOTE se actualiza utilizando una sentencia tal como:

```
UPDATE CURRENTQUOTE
  SET QUOTE = 68.5
  WHERE SYMBOL = 'IBM'
```

La columna STATUS de CURRENTQUOTE se debe actualizar para reflejar si las acciones:

- Están subiendo de valor
- Han alcanzado una nueva cotización máxima del año
- Están disminuyendo de valor
- Han alcanzado una nueva cotización mínima del año
- Están en un valor estable.

Esto se realiza mediante este desencadenante anterior:

1

```
CREATE TRIGGER STOCK_STATUS
  NO CASCADE BEFORE UPDATE OF QUOTE ON CURRENTQUOTE
  REFERENCING NEW AS NEWQUOTE OLD AS OLDQUOTE
  FOR EACH ROW MODE DB2SQL
```

2

```
SET NEWQUOTE.STATUS =
```

3

```
CASE
```

4

```
  WHEN NEWQUOTE.QUOTE >=
    (SELECT MAX(QUOTE)
     FROM QUOTEHISTORY
     WHERE SYMBOL = NEWQUOTE.SYMBOL
     AND YEAR(TIMESTAMP) = YEAR(CURRENT DATE) )
  THEN 'High'
```

5

```
  WHEN NEWQUOTE.QUOTE <=
    (SELECT MIN(QUOTE)
     FROM QUOTEHISTORY
     WHERE SYMBOL = NEWQUOTE.SYMBOL
     AND YEAR(TIMESTAMP) = YEAR(CURRENT DATE) )
  THEN 'Low'
```

6

```
  WHEN NEWQUOTE.QUOTE > OLDQUOTE.QUOTE
  THEN 'Rising'
  WHEN NEWQUOTE.QUOTE < OLDQUOTE.QUOTE
```

```

        THEN 'Dropping'
    WHEN NEWQUOTE.QUOTE = OLDQUOTE.QUOTE
        THEN 'Steady'
END

```

- 1 Este bloque de código define un desencadenante llamado STOCK_STATUS que se debe activar antes de actualizar la columna QUOTE de la tabla CURRENTQUOTE. La segunda línea especifica que la acción desencadenada se debe aplicar antes de cualquier cambio producido en la base de datos por la actualización propiamente dicha de la tabla CURRENTQUOTE. La cláusula NO CASCADE significa que la acción desencadenada no provocará la activación de ningún otro desencadenante. La tercera línea especifica los nombres que se deben utilizar como calificadores del nombre de columna para los valores nuevos (NEWQUOTE) y para los valores antiguos (OLDQUOTE). Los nombre de columna calificados con estos nombres de correlación (NEWQUOTE y OLDQUOTE) se denominan *variables de transición*. La cuarta línea indica que la acción desencadenada se debe ejecutar para cada fila.
- 2 Esto señala el inicio de la primera y única sentencia SQL contenida en la acción desencadenada de este desencadenante. La sentencia con la variable de transición SET se utiliza en un desencadenante para asignar un valor a una columna en la fila de la tabla que se está actualizando mediante la sentencia activada por el desencadenante. Esta sentencia asigna un valor a la columna STATUS de la tabla CURRENTQUOTE.
- 3 La expresión utilizada en el lado derecho de la asignación es una expresión CASE. La expresión CASE abarca hasta la palabra clave END.
- 4 El primer bloque de la expresión CASE comprueba si la nueva cotización (NEWQUOTE.QUOTE) sobrepasa el valor máximo correspondiente al símbolo de las acciones en el año natural actual. La subconsulta utiliza la tabla QUOTEHISTORY que es actualizada por el desencadenante posterior que sigue a continuación.
- 5 El segundo bloque de la expresión CASE comprueba si la nueva cotización (NEWQUOTE.QUOTE) es menor que el valor mínimo correspondiente al símbolo de las acciones en el año natural actual. La subconsulta utiliza la tabla QUOTEHISTORY que es actualizada por el desencadenante posterior que sigue a continuación.
- 6 Los últimos tres bloques de CASE comparan la nueva cotización (NEWQUOTE.QUOTE) con la cotización que había en la tabla (OLDQUOTE.QUOTE) para determinar si es mayor, menor o igual. La sentencia con la variable de transición SET finaliza aquí.

Además de actualizar la entrada en la tabla CURRENTQUOTE, es necesario crear un registro de auditoría en la tabla QUOTEHISTORY, mediante la copia de la nueva cotización acompañada de una indicación de fecha. Esto se realiza mediante este desencadenante posterior:

1

```
CREATE TRIGGER RECORD_HISTORY
AFTER UPDATE OF QUOTE ON CURRENTQUOTE
REFERENCING NEW AS NEWQUOTE
FOR EACH ROW MODE DB2SQL
BEGIN ATOMIC
```

2

```
INSERT INTO QUOTEHISTORY
VALUES (NEWQUOTE.SYMBOL, NEWQUOTE.QUOTE, CURRENT_TIMESTAMP);
END
```

1

Este bloque de código define un desencadenante llamado RECORD_HISTORY que se debe activar antes de actualizar la columna QUOTE de la tabla CURRENTQUOTE. La tercera línea especifica el nombre que se debe utilizar como calificador del nombre de columna para el nuevo valor (NEWQUOTE). La cuarta línea indica que la acción desencadenada se debe ejecutar para cada fila.

2

La acción desencadenada de este desencadenante contiene una sentencia SQL individual que inserta una fila en la tabla QUOTEHISTORY utilizando los datos de la fila que se ha actualizado (NEWQUOTE.SYMBOL y NEWQUOTE.QUOTE) y la indicación de fecha actual.

CURRENT_TIMESTAMP es un registro especial que contiene la indicación de la fecha. La sección “Registros especiales” en la página 70 proporciona una lista y una explicación.

Uniones

El proceso de combinar datos de dos o más tablas se denomina unión de tablas. El gestor de bases de datos forma todas las combinaciones de filas procedentes de las tablas especificadas. Para cada combinación, comprueba la *condición de unión*. Una condición de unión es una condición de búsqueda, con algunas restricciones. Para obtener una lista de restricciones, vea el manual *Consulta de SQL*.

Observe que no es necesario que las columnas que intervienen en la unión tengan el mismo tipo de datos, pero deben ser compatibles. La condición de unión se evalúa de la misma manera que cualquier otra condición de búsqueda, y son aplicables las mismas reglas para las comparaciones.

Si no se especifica una condición de unión, se forman todas las combinaciones de filas de las tablas listadas en la cláusula FROM, aunque las filas puedan no tener ninguna relación entre sí. El resultado se denomina *producto cruzado* de las dos tablas.

Los ejemplos de esta sección están basados en las dos tablas siguientes. Estas tablas son simplificaciones de las tablas contenidas en la base de datos de ejemplo, pero no existen en ella. Se utilizan para tratar cuestiones interesantes sobre las uniones en general. SAMP_STAFF lista el nombre de los empleados que no están empleados como contratistas y su puesto de trabajo; SAMP_PROJECT lista el nombre de los empleados (con contrato y dedicación completa) y los proyectos en los que trabajan.

Las tablas son las siguientes:

NAME	PROJ
Haas	AD3100
Thompson	PL2100
Walker	MA2112
Lutz	MA2111

Figura 5. Tabla SAMP_PROJECT

NAME	JOB
Haas	PRES
Thompson	MANAGER
Lucchessi	SALESREP
Nicholls	ANALYST

Figura 6. Tabla SAMP_STAFF

El ejemplo siguiente genera el producto cruzado de las dos tablas. No se especifica una condición de unión, por lo que se muestran todas las combinaciones de filas:

```
SELECT SAMP_PROJECT.NAME,  
       SAMP_PROJECT.PROJ, SAMP_STAFF.NAME, SAMP_STAFF.JOB  
FROM SAMP_PROJECT, SAMP_STAFF
```

Esta sentencia produce el resultado siguiente:

NAME	PROJ	NAME	JOB
-----		-----	-----
Haas	AD3100	Haas	PRES
Thompson	PL2100	Haas	PRES
Walker	MA2112	Haas	PRES
Lutz	MA2111	Haas	PRES
Haas	AD3100	Thompson	MANAGER
Thompson	PL2100	Thompson	MANAGER
Walker	MA2112	Thompson	MANAGER
Lutz	MA2111	Thompson	MANAGER
Haas	AD3100	Lucchessi	SALESREP
Thompson	PL2100	Lucchessi	SALESREP
Walker	MA2112	Lucchessi	SALESREP
Lutz	MA2111	Lucchessi	SALESREP
Haas	AD3100	Nicholls	ANALYST
Thompson	PL2100	Nicholls	ANALYST
Walker	MA2112	Nicholls	ANALYST
Lutz	MA2111	Nicholls	ANALYST

Los dos tipos principales de uniones son las *uniones internas* y las *uniones externas*. Hasta el momento, en todos los ejemplos hemos utilizado la unión interna. Las uniones internas sólo conservan las filas del producto cruzado que cumplen la condición de unión. Si una fila existe en una tabla, pero no en la otra, la información no se incluye en la tabla resultante.

El ejemplo siguiente produce la unión interna de las dos tablas. La unión interna lista los empleados con dedicación completa que están asignados a un proyecto:

```

SELECT SAMP_PROJECT.NAME,
       SAMP_PROJECT.PROJ, SAMP_STAFF.NAME, SAMP_STAFF.JOB
FROM SAMP_PROJECT, SAMP_STAFF
WHERE SAMP_STAFF.NAME = SAMP_PROJECT.NAME

```

Como método alternativo, puede especificar la unión interna del modo siguiente:

```

SELECT SAMP_PROJECT.NAME,
       SAMP_PROJECT.PROJ, SAMP_STAFF.NAME, SAMP_STAFF.JOB
FROM SAMP_PROJECT INNER JOIN SAMP_STAFF
ON SAMP_STAFF.NAME = SAMP_PROJECT.NAME

```

El resultado es:

NAME	PROJ	NAME	JOB
-----		-----	-----
Haas	AD3100	Haas	PRES
Thompson	PL2100	Thompson	MANAGER

Observe que el resultado de la unión interna consta de filas que tienen valores coincidentes para la columna NAME en la tablas derecha e izquierda - 'Haas' y 'Thompson' están en tabla SAMP_STAFF, que lista los empleados con

dedicación completa, y en la tabla SAMP_PROJECT, que lista los empleados con contrato y dedicación completa asignados a un proyecto.

Las uniones externas son una concatenación de la unión interna y de las filas ausentes en la unión interna que proceden de la tabla izquierda, la tabla derecha o ambas tablas. Cuando se realiza una unión externa de dos tablas, una tabla se designa arbitrariamente como tabla izquierda y la otra como tabla derecha. Existen tres tipos de uniones externas:

1. La **unión externa izquierda** incluye la unión interna y las filas de la tabla izquierda no contenidas en la unión interna.
2. La **unión externa derecha** incluye la unión interna y las filas de la tabla derecha no contenidas en la unión interna.
3. La **unión externa total** incluye la unión interna y las filas de la tabla derecha no contenidas en la unión interna.

Utilice la sentencia SELECT para especificar las columnas que se deben mostrar. En la cláusula FROM, especifique el nombre de la primera tabla seguido de las palabras clave LEFT OUTER JOIN, RIGHT OUTER JOIN o FULL OUTER JOIN. A continuación, especifique la segunda tabla seguida de la palabra clave ON. Después de la palabra clave ON, especifique la condición de unión para expresar una relación entre las tablas a unir.

En el ejemplo siguiente, SAMP_STAFF está designada como tabla derecha y SAMP_PROJECT como tabla izquierda. Mediante LEFT OUTER JOIN, se lista el nombre y número de proyecto de los empleados con contrato y dedicación completa (listados en SAMP_PROJECT) y su puesto de trabajo, si son empleados con dedicación completa (listados en SAMP_STAFF):

```
SELECT SAMP_PROJECT.NAME, SAMP_PROJECT.PROJ,
       SAMP_STAFF.NAME, SAMP_STAFF.JOB
FROM SAMP_PROJECT LEFT OUTER JOIN SAMP_STAFF
ON SAMP_STAFF.NAME = SAMP_PROJECT.NAME
```

Esta sentencia produce el resultado siguiente:

NAME	PROJ	NAME	JOB
Haas	AD3100	Haas	PRES
Lutz	MA2111	-	-
Thompson	PL2100	Thompson	MANAGER
Walker	MA2112	-	-

El resultado de la unión interna son las filas que tienen valores en todas las columnas. Estas son las filas que cumplen la condición de búsqueda: 'Haas' y 'Thompson' aparecen en SAMP_PROJECT (tabla izquierda) y en SAMP_STAFF (tabla derecha). Para las filas que no cumplen la condición de búsqueda, aparece el valor nulo en las columnas de la tabla derecha: 'Lutz' y 'Walker' son empleados con contrato que aparecen en la tabla

SAMP_PROJECT, pero no en la tabla SAMP_STAFF. Observe que el conjunto resultante incluye todas las filas de la tabla izquierda.

En el ejemplo siguiente, SAMP_STAFF está designada como tabla derecha y SAMP_PROJECT como tabla izquierda. Mediante RIGHT OUTER JOIN, se lista el nombre y puesto de trabajo de los empleados con dedicación completa (listados en SAMP_STAFF) y su número de proyecto, si están asignados a un proyecto (listado en SAMP_PROJECT):

```
SELECT SAMP_PROJECT.NAME,
       SAMP_PROJECT.PROJ, SAMP_STAFF.NAME, SAMP_STAFF.JOB
FROM SAMP_PROJECT RIGHT OUTER JOIN SAMP_STAFF
ON SAMP_STAFF.NAME = SAMP_PROJECT.NAME
```

El resultado es:

NAME	PROJ	NAME	JOB
Haas	AD3100	Haas	PRES
-	-	Lucchessi	SALESREP
-	-	Nicholls	ANALYST
Thompson	PL2100	Thompson	MANAGER

Como en el caso de la unión externa izquierda, el resultado de la unión interna son las filas que tienen valores en todas las columnas. Estas son las filas que cumplen la condición de búsqueda: 'Haas' y 'Thompson' aparecen en SAMP_PROJECT (tabla izquierda) y en SAMP_STAFF (tabla derecha). Para las filas que no cumplen la condición de búsqueda, aparece el valor nulo en las columnas de la tabla derecha: 'Lucchessi' y 'Nicholls' son empleados con dedicación completa que no están asignados a un proyecto. Aunque estos empleados aparecen en SAMP_STAFF, no están en SAMP_PROJECT. Observe que el conjunto resultante incluye todas las filas de la tabla derecha.

El ejemplo siguiente utiliza FULL OUTER JOIN con las tablas SAMP_PROJECT y SAMP_STAFF. La sentencia lista el nombre de todos los empleados con dedicación completa, incluidos los no asignados a un proyecto, y los empleados con contrato:

```
SELECT SAMP_PROJECT.NAME, SAMP_PROJECT.PROJ,
       SAMP_STAFF.NAME, SAMP_STAFF.JOB
FROM SAMP_PROJECT FULL OUTER JOIN SAMP_STAFF
ON SAMP_STAFF.NAME = SAMP_PROJECT.NAME
```

El resultado es:

NAME	PROJ	NAME	JOB
Haas	AD3100	Haas	PRES
-	-	Lucchessi	SALESREP
-	-	Nicholls	ANALYST

Thompson	PL2100	Thompson	MANAGER
Lutz	MA2111	-	-
Walker	MA2112	-	-

Este resultado comprende la unión externa izquierda, la unión externa derecha y la unión interna. Se listan todos los empleados con dedicación completa y contrato. Al igual que en la unión externa izquierda y la unión externa derecha, aparece un valor nulo para los valores de columna que no cumplen la condición de unión. El conjunto resultante comprende todas las filas de SAMP_STAFF y SAMP_PROJECT.

Consultas complejas

DB2 Universal Database le permite agrupar, fusionar y visualizar varias columnas en un conjunto resultante individual mediante la utilización de ROLLUP y CUBE. Esta nueva y potente característica mejora y simplifica el análisis de datos basado en el SQL.

Existen diversos métodos para extraer información útil de la base de datos. Puede aplicar consultas recursivas para producir tablas resultantes a partir de los conjuntos de datos existentes.

Consultas ROLLUP y CUBE

Puede especificar las operaciones ROLLUP y CUBE en la cláusula GROUP BY de una consulta.

ROLLUP produce un conjunto resultante que contiene las filas agrupadas normales y *filas de subtotal*. CUBE produce un conjunto resultante que contiene las filas obtenidas por ROLLUP y filas de tabulación cruzada.

Por tanto, con ROLLUP, podría obtener las ventas por persona y mes, junto con los totales mensuales de ventas y un total general. Con CUBE, se obtendrían filas adicionales para el total de ventas por persona.

Vea el manual *Consulta de SQL* para obtener más detalles.

Consultas recursivas

Una *consulta recursiva* es una consulta que utiliza iterativamente datos resultantes para determinar más resultados. Puede equipararlo conceptualmente a recorrer un árbol o un grafo.

Son ejemplos prácticos de esto los sistemas de reservas y la planificación de redes.

Las consultas recursivas se escriben utilizando una expresión de tabla común que incluye una referencia a su propio nombre.

En el manual *Consulta de SQL* encontrará ejemplos de consultas recursivas.

Funciones OLAP

Las funciones de Proceso Analítico en Línea (OnLine Analytical Processing (OnLine Analytical Processing (OLAP)) realizan una operación de función de columna sobre una *ventana* de datos. Esta ventana puede especificar un particionamiento de filas, una ordenación de filas dentro de particiones o un *grupo de agregación*. El grupo de agregación permite al usuario especificar qué filas, con respecto a la fila actual, intervienen en el cálculo. La utilización de una ventana de esta clase permite realizar operaciones tales como sumas acumulativas y promedios móviles.

Además de permitir al usuario especificar una ventana para funciones de columna existentes (tales como SUM y AVG), las funciones OLAP pueden realizar operaciones de ordenación (RANK y DENSE_RANK) y numeración de filas (ROW_NUMBER), para un particionamiento y una ordenación de filas determinados.

La consulta de ejemplo siguiente proporciona la categoría de un empleado dentro del departamento, basándose en el salario, y muestra la suma acumulativa de los salarios dentro del departamento (para los departamentos 15 y 38):

```
SELECT NAME, DEPT,  
       RANK () OVER (PARTITION BY DEPT ORDER BY SALARY DESC) AS RANK,  
       SUM (SALARY) OVER (PARTITION BY DEPT  
                          ORDER BY SALARY DESC  
                          ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)  
       AS CUMULATIVE_SUM  
FROM STAFF  
WHERE DEPT IN (15,38)  
ORDER BY DEPT, RANK
```

Esta sentencia produce el resultado siguiente:

NAME	DEPT	RANK	CUMULATIVE_SUM
Hanes	15	1	20659.80
Rothman	15	2	37162.63
Ngan	15	3	49670.83
Kermisch	15	4	61929.33
O'Brien	38	1	18006.00
Marenghi	38	2	35512.75
Quigley	38	3	52321.05
Naughton	38	4	65275.80
Abrahams	38	5	77285.55

Capítulo 8. Personalización y mejora del manejo de datos

Este capítulo proporciona una breve introducción a las *extensiones orientadas a objetos* de DB2 Universal Database. Utilizar extensiones orientadas a objetos tiene muchas ventajas. Los *tipos definidos por el usuario* (*user-defined types, UDT*) permiten utilizar más tipos de datos en las aplicaciones del usuario, mientras que las *funciones definidas por el usuario* (*user-defined functions, UDF*) permiten crear funciones específicas de la aplicación. Las UDF actúan como *métodos* para los UDT al proporcionar un comportamiento y una encapsulación uniformes de los tipos.

A continuación se tratan los *registros especiales* y los *catálogos del sistema*. Los registros especiales son áreas de almacenamiento definidas por el gestor de bases de datos; se utilizan para almacenar información utilizada por las sentencias SQL. Los registros especiales se definen durante la conexión y son específicos del proceso de la aplicación. Los catálogos del sistema contienen información sobre la estructura lógica y física de los objetos de la base de datos.

Este capítulo trata los temas siguientes:

- Tipos definidos por el usuario
- Funciones definidas por el usuario
- Objetos grandes
- Registros especiales
- Introducción a las vistas de catálogo

Queda fuera del ámbito de este libro la exposición detallada de los temas indicados anteriormente; estos temas se tratan con detalle en los manuales *Consulta de SQL* y *Administration Guide*.

Tipos definidos por el usuario

Un *tipo diferenciado* es un tipo de datos definido por el usuario que tiene la misma representación interna que un tipo existente (su tipo “fuente”), pero se considera que es un tipo diferente e incompatible para la mayoría de las operaciones. Por ejemplo, se puede definir un tipo edad, un tipo peso y un tipo estatura, todos los cuales tienen semánticas completamente diferentes, pero que utilizan el tipo de datos incorporado INTEGER para sus representaciones internas.

El ejemplo siguiente muestra la creación de un tipo diferenciado llamado PAY:

CREATE DISTINCT TYPE PAY AS DECIMAL(9,2) WITH COMPARISONS

Aunque PAY tiene la misma representación que el tipo de datos incorporado DECIMAL(9,2), se considera que es un tipo diferente que no es compatible con DECIMAL(9,2) ni con ningún otro tipo. Sólo se puede comparar con el mismo tipo diferenciado. Además, los operadores y funciones que pueden actuar sobre DECIMAL no serían aplicables aquí. Por ejemplo, un valor cuyo tipo de datos es PAY no se puede multiplicar por un valor cuyo tipo de datos es INTEGER. Por tanto, es necesario escribir funciones que sólo son aplicables al tipo de datos PAY.

La utilización de tipos de datos diferenciados limita los errores accidentales. Por ejemplo, si la columna SALARY de la tabla EMPLOYEE está definida como perteneciente al tipo de datos PAY, no se podrá sumar a COMM aunque tengan el mismo tipo fuente.

Los tipos de datos diferenciados permiten la conversión del tipo de datos. Un tipo fuente se puede convertir a un tipo de datos diferenciado y viceversa. Por ejemplo, si la columna SALARY de la tabla EMPLOYEE está definida como perteneciente al tipo de datos PAY, el ejemplo siguiente permite el uso del operador de comparación:

```
SELECT * FROM EMPLOYEE
WHERE DECIMAL(SALARY) = 41250
```

DECIMAL(SALARY) devuelve un tipo de datos decimal. Inversamente, un tipo de datos numérico se puede convertir a un tipo PAY. Por ejemplo, puede convertir el número 41250 utilizando PAY(41250).

Funciones definidas por el usuario

Tal como se mencionó en “Utilización de funciones” en la página 28, DB2 Universal Database proporciona funciones incorporadas y funciones definidas por el usuario (UDF). Sin embargo, este conjunto de funciones no satisface nunca todas las necesidades del usuario. A menudo, es necesario crear funciones personalizadas para tareas determinadas. Las funciones definidas por el usuario le permiten crear funciones personalizadas.

Hay cuatro tipos de funciones definidas por el usuario: *funciones fuente* (o modelo), *funciones escalares externas*, *funciones de tabla externa* y *funciones de tabla externa DB OLE*.

Esta sección trata las funciones fuente y las funciones escalares externas. Para obtener más información sobre las funciones de tabla externa y las funciones de tabla DB OLE, vea el manual *Consulta de SQL*.

Las funciones fuente definidas por el usuario permiten que tipos definidos por el usuario hagan referencia de forma selectiva a otra función incorporada o definida por el usuario que ya es conocida para la base de datos. Puede utilizar funciones escalares y de columna a la vez.

En el ejemplo siguiente, se crea una función definida por el usuario (llamada MAX) que está basada en la función de columna incorporada MAX, la cual acepta el tipo de datos DECIMAL como entrada. La UDF MAX acepta un tipo PAY como entrada y devuelve un tipo PAY como salida.

```
CREATE FUNCTION MAX(PAY) RETURNS PAY
SOURCE MAX(DECIMAL)
```

Las funciones externas definidas por el usuario se escriben en un lenguaje de programación. Existen *funciones escalares externas* y *funciones de tabla externas*; ambas se examinan en el manual *Consulta de SQL*.

Supongamos, por ejemplo, que ha escrito una función que cuenta el número de palabras de una serie de caracteres; puede registrar esta función en la base de datos utilizando la sentencia CREATE FUNCTION junto con el nombre WORDCOUNT. Después, esta función se puede utilizar en sentencias SQL.

La sentencia siguiente devuelve los números de empleado y el número de palabras de sus currículums en formato ASCII. WORDCOUNT es una función escalar externa que el usuario ha registrado en la base de datos y ahora se utiliza en la sentencia.

```
SELECT EMPNO, WORDCOUNT(RESUME)
FROM EMP_RESUME
WHERE RESUME_FORMAT = 'ascii'
```

Para obtener información más detallada sobre la escritura de funciones definidas por el usuario, consulte el manual *Application Development Guide*.

Objetos grandes

El término *objeto grande* y su acrónimo inglés *LOB* (large object) se utilizan para hacer referencia a tres tipos de datos: BLOB, CLOB y DBCLOB. Estos tipos pueden contener grandes cantidades de datos, correspondientes a objetos tales como audio, fotos y documentos.

Un *gran objeto binario* (binary large object, BLOB) es una serie de caracteres de longitud variable, medida en bytes, que puede tener hasta 2 gigabytes de longitud. Un BLOB se utiliza principalmente para contener información no convencional, tal como imágenes, voz y soportes mixtos de datos.

Un *gran objeto de caracteres* (character large object, CLOB) es una serie de caracteres de longitud variable, medida en bytes, que puede tener hasta 2 gigabytes de longitud. Un CLOB se utiliza para almacenar datos del juego de caracteres de un solo byte, tales como documentos. Se considera que un CLOB es una serie de caracteres.

Un *gran objeto de caracteres de doble byte* (double-byte character large object, DBCLOB) es una serie de caracteres de doble byte de longitud variable que puede tener hasta 2 gigabytes de longitud (1.073.741.823 caracteres de doble byte). Un DBCLOB se utiliza para almacenar volúmenes grandes de datos del juego de caracteres de doble byte, tales como documentos. Se considera que un DBCLOB es una serie gráfica.

Manejo de objetos grandes (LOB)

Debido a que los valores LOB pueden ser muy grandes, su transferencia desde el servidor de bases de datos al programa de aplicación cliente puede exigir mucho tiempo. Sin embargo, generalmente los valores LOB se procesan en porciones individuales, en lugar de como un todo. Cuando una aplicación no necesita (o desea) almacenar el valor LOB completo en la memoria de la aplicación, puede hacer referencia a este valor mediante una *variable localizadora de gran objeto*.

Las sentencias subsiguientes pueden luego utilizar los localizadores para realizar operaciones en los datos sin necesidad de recuperar el gran objeto completo. Las variables localizadoras se utilizan para disminuir las necesidades de memoria de las aplicaciones, y para mejorar el rendimiento disminuyendo el flujo de datos entre el cliente y el servidor.

Otro mecanismo son las *variables de referencia a archivos*. Se utilizan para recuperar un gran objeto directamente en un archivo o para actualizar un gran objeto de una tabla directamente desde un archivo. Las variables de referencia a archivos se utilizan para disminuir las necesidades de memoria de las aplicaciones, pues éstas no necesitan almacenar los datos del gran objeto. Para obtener más información, consulte los manuales *Application Development Guide* y *Consulta de SQL*.

Registros especiales

Un *registro especial* es un espacio de memoria que el gestor de bases de datos define para una conexión y sirve para almacenar información que puede ser utilizada en sentencias SQL. A continuación siguen algunos ejemplos de los registros especiales de uso más habitual. Para obtener una lista de todos los registros especiales e información más detallada, consulte el manual *Consulta de SQL*.

- CURRENT DATE: Contiene la fecha de acuerdo con el reloj del sistema en el momento de ejecución de la sentencia SQL.

- **CURRENT FUNCTION PATH:** Contiene un valor que especifica la vía de acceso de función que se utiliza para resolver las referencias a funciones y tipos de datos.
- **CURRENT SERVER:** Especifica el servidor de aplicaciones actual.
- **CURRENT TIME:** Contiene la hora de acuerdo con el reloj del sistema en el momento de ejecución de la sentencia SQL.
- **CURRENT TIMESTAMP:** Especifica una indicación de fecha y hora de acuerdo con el reloj del sistema en el momento de ejecución de la sentencia SQL.
- **CURRENT TIMEZONE:** Especifica la diferencia entre la Hora Universal Coordinada y la hora local en el servidor de aplicaciones.
- **USER:** Especifica el ID de autorización de ejecución.

Puede visualizar el contenido de un registro especial mediante la sentencia **VALUES**. Por ejemplo:

```
VALUES (CURRENT TIMESTAMP)
```

También podría utilizar:

```
SELECT CURRENT TIMESTAMP FROM ORG
```

y esto devolverá la indicación de fecha y hora para cada fila de la tabla.

Introducción a las vistas de catálogo

DB2 crea y mantiene un conjunto amplio de tablas de catálogo del sistema para cada base de datos. Estas tablas contienen información sobre la estructura lógica y física de objetos de base de datos, tales como tablas, vistas, paquetes, relaciones de integridad referencial, funciones, tipos diferenciados y desencadenantes. Estas tablas se crean cuando se crea la base de datos y se actualizan durante las operaciones normales. El usuario no puede crearlas ni eliminarlas explícitamente, pero puede consultarlas y visualizar su contenido.

Para obtener más información, consulte el manual *Consulta de SQL*.

Selección de filas en catálogos del sistema

Las vistas de catálogo son como cualquier otra vista de base de datos. Puede utilizar sentencias SQL para examinar los datos, exactamente de la misma manera que lo haría para cualquier otra vista del sistema.

En el catálogo SYSCAT.TABLES encontrará información muy útil sobre tablas. Para determinar los nombres de tablas existentes que ha creado, emita una sentencia como la siguiente:

```

SELECT TABNAME, TYPE, CREATE_TIME
FROM SYSCAT.TABLES
WHERE DEFINER = USER

```

Esta sentencia produce el resultado siguiente:

TABNAME	TYPE	CREATE_TIME
ORG	T	1999-07-21-13.42.55.128005
STAFF	T	1999-07-21-13.42.55.609001
DEPARTMENT	T	1999-07-21-13.42.56.069001
EMPLOYEE	T	1999-07-21-13.42.56.310001
EMP_ACT	T	1999-07-21-13.42.56.710001
PROJECT	T	1999-07-21-13.42.57.051001
EMP_PHOTO	T	1999-07-21-13.42.57.361001
EMP_RESUME	T	1999-07-21-13.42.59.154001
SALÉS	T	1999-07-21-13.42.59.855001
CL_SCHED	T	1999-07-21-13.43.00.025002
IN_TRAY	T	1999-07-21-13.43.00.055001

La lista siguiente indica las vistas de catálogo relativas a temas tratados en el presente manual. Existen otras muchas vistas de catálogo; están descritas con detalle en los manuales *Consulta de SQL* y *Administration Guide*.

Descripción	Vista de catálogo
restricciones de comprobación	SYSCAT.CHECKS
columnas	SYSCAT.COLUMNS
columnas referenciadas por restricciones de comprobación	SYSCAT.COLCHECKS
columnas utilizadas en claves	SYSCAT.KEYCOLUSE
tipos de datos	SYSCAT.DATATYPES
parámetros de función o resultado de una función	SYSCAT.FUNCPARMS
restricciones referenciales	SYSCAT.REFERENCES
esquemas	SYSCAT.SCHEMATA
restricciones de tabla	SYSCAT.TABCONST
tablas	SYSCAT.TABLES
desencadenantes	SYSCAT.TRIGGERS
funciones definidas por el usuario	SYSCAT.FUNCTIONS
vistas	SYSCAT.VIEWS

Apéndice A. Tablas de base de datos de ejemplo

Este apéndice muestra la información contenida en las tablas de ejemplo de la base de datos de ejemplo SAMPLE y describe cómo instalarlas y eliminarlas.

Se proporcionan más bases de datos de ejemplo con DB2 Universal Database para demostrar las funciones de la inteligencia de la empresa y se utilizan en la guía de aprendizaje de la inteligencia de la empresa. Sin embargo, sólo se describe en este apéndice el contenido de la base de datos de ejemplo SAMPLE. Consulte *Centro de depósito de datos Administration Guide* para obtener más información acerca de las bases de datos de ejemplo de inteligencia de la empresa.

Las tablas de ejemplo se utilizan en los ejemplos del presente manual y en otros manuales de esta biblioteca. Además, se muestran los datos contenidos en los archivos de ejemplo cuyos tipos de datos son BLOB y CLOB.

Este apéndice contiene las secciones siguientes:

- “La base de datos de ejemplo” en la página 74
- “Para crear la base de datos de ejemplo” en la página 74
- “Borrado de la base de datos de ejemplo” en la página 74
- “Tabla CL_SCHED” en la página 75
- “Tabla DEPARTMENT” en la página 75
- “Tabla EMPLOYEE” en la página 75
- “Tabla EMP_ACT” en la página 79
- “Tabla EMP_PHOTO” en la página 81
- “Tabla EMP_RESUME” en la página 81
- “Tabla IN_TRAY” en la página 82
- “Tabla ORG” en la página 82
- “Tabla PROJECT” en la página 83
- “Tabla SALES” en la página 84
- “Tabla STAFF” en la página 85
- “Tabla STAFFG” en la página 86
- “Archivos de ejemplo con el tipo de datos BLOB y CLOB” en la página 87
- “Foto de Quintana” en la página 87
- “Currículum de Quintana” en la página 87
- “Foto de Nicholls” en la página 88
- “Currículum de Nicholls” en la página 89
- “Foto de Adamson” en la página 90
- “Currículum de Adamson” en la página 90
- “Foto de Walker” en la página 91
- “Currículum de Walker” en la página 92.

Tablas de base de datos de ejemplo

En las tablas de ejemplo, un guión (-) denota un valor nulo.

La base de datos de ejemplo

Los ejemplos del presente manual utilizan una base de datos de ejemplo. Para utilizar estos ejemplos, debe crear la base de datos SAMPLE. Para manejar la base de datos, debe instalar el gestor de bases de datos.

Para crear la base de datos de ejemplo

La base de datos de ejemplo se crea mediante un archivo ejecutable.² Para crear una base de datos debe tener autorización SYSADM.

- **Cuando se utilizan plataformas basadas en UNIX**

Si está utilizando el indicador de mandatos del sistema operativo, escriba:

```
sql1lib/bin/db2samp1 <path>
```

desde el directorio inicial del propietario de la instancia del gestor de bases de datos, donde *vía_de_acceso* es un parámetro opcional que especifica la vía de acceso donde debe crearse la base de datos de ejemplo. Pulse Intro.³ El esquema de DB2SAMPL es el valor del registro especial CURRENT SCHEMA.

- **Cuando se utilizan plataformas OS/2 o Windows**

Si está utilizando el indicador de mandatos del sistema operativo, escriba:

```
db2samp1 e
```

donde *e* es un parámetro opcional que especifica la unidad donde debe crearse la base de datos. Pulse Intro.⁴

Si no está conectado a la estación de trabajo mediante Gestión de perfiles de usuarios, se le solicitará que se conecte.

Borrado de la base de datos de ejemplo

Si no necesita acceder a la base de datos de ejemplo, puede borrarla utilizando el mandato DROP DATABASE:

```
db2 drop database sample
```

2. Para obtener información referente a este mandato, vea el mandato DB2SAMPL en el manual *Consulta de mandatos*.

3. Si no se especifica el parámetro de vía de acceso, la base de datos de ejemplo se crea en la vía de acceso por omisión especificada por el parámetro DFTDBPATH en el archivo de configuración del gestor de base de datos.

4. Si no se especifica en el parámetro de unidad, la base de datos de ejemplo se crea en la misma unidad que DB2.

Tabla CL_SCHED

Name:	CLASS_CODE	DAY	STARTING	ENDING
Type:	char(7)	smallint	time	time
Desc:	Class Code (room:teacher)	Day # of 4 day schedule	Class Start Time	Class End Time

Tabla DEPARTMENT

Name:	DEPTNO	DEPTNAME	MGRNO	ADMRDEPT	LOCATION
Type:	char(3) not null	varchar(29) not null	char(6)	char(3) not null	char(16)
Desc:	Department number	Name describing general activities of department	Employee number (EMPNO) of department manager	Department (DEPTNO) to which this department reports	Name of the remote location
Values:	A00	SPIFFY COMPUTER SERVICE DIV.	000010	A00	-
	B01	PLANNING	000020	A00	-
	C01	INFORMATION CENTER	000030	A00	-
	D01	DEVELOPMENT CENTER	-	A00	-
	D11	MANUFACTURING SYSTEMS	000060	D01	-
	D21	ADMINISTRATION SYSTEMS	000070	D01	-
	E01	SUPPORT SERVICES	000050	A00	-
	E11	OPERATIONS	000090	E01	-
	E21	SOFTWARE SUPPORT	000100	E01	-

Tabla EMPLOYEE

Names:	EMPNO	FIRSTNME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE
Type:	char(6) not null	varchar(12) not null	char(1) not null	varchar(15) not null	char(3)	char(4)	date
Desc:	Employee number	First name	Middle initial	Last name	Department (DEPTNO) in which the employee works	Phone number	Date of hire
JOB	EDLEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM	
char(8)	smallint not null	char(1)	date	dec(9,2)	dec(9,2)	dec(9,2)	
Job	Number of years of formal education	Sex (M male, F female)	Date of birth	Yearly salary	Yearly bonus	Yearly commission	

Tablas de base de datos de ejemplo

Vea la página siguiente para conocer los valores de la tabla EMPLOYEE.

EMPNO	FIRSTNAME	MID INIT	LASTNAME	WORK DEPT	PHONE NO	HIREDATE	JOB	ED LEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM
char(6) not null	varchar(12) not null	char(1) not null	varchar(15) not null	char(3)	char(4)	date	char(8)	smallint not null	char(1)	date	dec(9,2)	dec(9,2)	dec(9,2)
000010	CHRISTINE	I	HAAS	A00	3978	1965-01-01	PRES	18	F	1933-08-24	52750	1000	4220
000020	MICHAEL	L	THOMPSON	B01	3476	1973-10-10	MANAGER	18	M	1948-02-02	41250	800	3300
000030	SALLY	A	KWAN	C01	4738	1975-04-05	MANAGER	20	F	1941-05-11	38250	800	3080
000050	JOHN	B	GEYER	E01	6789	1949-08-17	MANAGER	16	M	1925-09-15	40175	800	3214
000060	IRVING	F	STERN	D11	6423	1973-09-14	MANAGER	16	M	1945-07-07	32250	500	2580
000070	EVA	D	PULASKI	D21	7831	1980-09-30	MANAGER	16	F	1953-05-26	36170	700	2893
000090	EILEEN	W	HENDERSON	E11	5498	1970-08-15	MANAGER	16	F	1941-05-15	29750	600	2380
000100	THEODORE	Q	SPENSER	E21	0972	1980-06-19	MANAGER	14	M	1956-12-18	26150	500	2092
000110	VINCENZO	G	LUCCHESSI	A00	3490	1958-05-16	SALESREP	19	M	1929-11-05	46500	900	3720
000120	SEAN	O'CONNELL	A00	2167	1963-12-05	1963-12-05	CLERK	14	M	1942-10-18	29250	600	2340
000130	DOLORES	M	QUINTANA	C01	4578	1971-07-28	ANALYST	16	F	1925-09-15	23800	500	1904
000140	HEATHER	A	NICHOLLS	C01	1793	1976-12-15	ANALYST	18	F	1946-01-19	28420	600	2274
000150	BRUCE	ADAMSON	D11	4510	1972-02-12	1972-02-12	DESIGNER	16	M	1947-05-17	25280	500	2022
000160	ELIZABETH	R	PIANKA	D11	3782	1977-10-11	DESIGNER	17	F	1955-04-12	22250	400	1780
000170	MASATOSHI	J	YOSHIMURA	D11	2890	1978-09-15	DESIGNER	16	M	1951-01-05	24680	500	1974
000180	MARILYN	S	SCOUTTEN	D11	1682	1973-07-07	DESIGNER	17	F	1949-02-21	21340	500	1707
000190	JAMES	H	WALKER	D11	2986	1974-07-26	DESIGNER	16	M	1952-06-25	20450	400	1636
000200	DAVID	BROWN	D11	4501	1966-03-03	1966-03-03	DESIGNER	16	M	1941-05-29	27740	600	2217
000210	WILLIAM	T	JONES	D11	0942	1979-04-11	DESIGNER	17	M	1953-02-23	18270	400	1462
000220	JENNIFER	K	LUTZ	D11	0672	1968-08-29	DESIGNER	18	F	1948-03-19	29840	600	2387
000230	JAMES	J	JEFFERSON	D21	2094	1966-11-21	CLERK	14	M	1935-05-30	22180	400	1774
000240	SALVATORE	M	MARINO	D21	3780	1979-12-05	CLERK	17	M	1954-03-31	28760	600	2301
000250	DANIEL	S	SMITH	D21	0961	1969-10-30	CLERK	15	M	1939-11-12	19180	400	1534
000260	SYBIL	P	JOHNSON	D21	8953	1975-09-11	CLERK	16	F	1936-10-05	17250	300	1380
000270	MARIA	L	PEREZ	D21	9001	1980-09-30	CLERK	15	F	1953-05-26	27380	500	2190
000280	ETHEL	R	SCHNEIDER	E11	8997	1967-03-24	OPERATOR	17	F	1936-03-28	26250	500	2100
000290	JOHN	R	PARKER	E11	4502	1980-05-30	OPERATOR	12	M	1946-07-09	15340	300	1227
000300	PHILIP	X	SMITH	E11	2095	1972-06-19	OPERATOR	14	M	1936-10-27	17750	400	1420
000310	MAUDE	F	SETRIGHT	E11	3332	1964-09-12	OPERATOR	12	F	1931-04-21	15900	300	1272
000320	RAMLAL	V	MEHTA	E21	9990	1965-07-07	FIELDREP	16	M	1932-08-11	19950	400	1596

Tablas de base de datos de ejemplo

EMPNO	FIRSTNME	MID INIT	LASTNAME	WORK DEPT	PHONE NO	HIREDATE	JOB	ED LEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM
000330	WING		LEE	E21	2103	1976-02-23	FIELDREP	14	M	1941-07-18	25370	500	2030
000340	JASON	R	GOUNOT	E21	5698	1947-05-05	FIELDREP	16	M	1926-05-17	23840	500	1907

Tabla EMP_ACT

Name:	EMPNO	PROJNO	ACTNO	EMPTIME	EMSTDATE	EMENDATE
Type:	char(6) not null	char(6) not null	smallint not null	dec(5,2)	date	date
Desc:	Employee number	Project number	Activity number	Proportion of employee's time spent on project	Date activity starts	Date activity ends
Values:	000010	AD3100	10	.50	1982-01-01	1982-07-01
	000070	AD3110	10	1.00	1982-01-01	1983-02-01
	000230	AD3111	60	1.00	1982-01-01	1982-03-15
	000230	AD3111	60	.50	1982-03-15	1982-04-15
	000230	AD3111	70	.50	1982-03-15	1982-10-15
	000230	AD3111	80	.50	1982-04-15	1982-10-15
	000230	AD3111	180	1.00	1982-10-15	1983-01-01
	000240	AD3111	70	1.00	1982-02-15	1982-09-15
	000240	AD3111	80	1.00	1982-09-15	1983-01-01
	000250	AD3112	60	1.00	1982-01-01	1982-02-01
	000250	AD3112	60	.50	1982-02-01	1982-03-15
	000250	AD3112	60	.50	1982-12-01	1983-01-01
	000250	AD3112	60	1.00	1983-01-01	1983-02-01
	000250	AD3112	70	.50	1982-02-01	1982-03-15
	000250	AD3112	70	1.00	1982-03-15	1982-08-15
	000250	AD3112	70	.25	1982-08-15	1982-10-15
	000250	AD3112	80	.25	1982-08-15	1982-10-15
	000250	AD3112	80	.50	1982-10-15	1982-12-01
	000250	AD3112	180	.50	1982-08-15	1983-01-01
	000260	AD3113	70	.50	1982-06-15	1982-07-01
	000260	AD3113	70	1.00	1982-07-01	1983-02-01
	000260	AD3113	80	1.00	1982-01-01	1982-03-01
	000260	AD3113	80	.50	1982-03-01	1982-04-15
	000260	AD3113	180	.50	1982-03-01	1982-04-15
	000260	AD3113	180	1.00	1982-04-15	1982-06-01
	000260	AD3113	180	.50	1982-06-01	1982-07-01
	000270	AD3113	60	.50	1982-03-01	1982-04-01
	000270	AD3113	60	1.00	1982-04-01	1982-09-01
	000270	AD3113	60	.25	1982-09-01	1982-10-15
	000270	AD3113	70	.75	1982-09-01	1982-10-15
	000270	AD3113	70	1.00	1982-10-15	1983-02-01

Tablas de base de datos de ejemplo

Name:	EMPNO	PROJNO	ACTNO	EMPTIME	EMSTDATE	EMENDATE
	000270	AD3113	80	1.00	1982-01-01	1982-03-01
	000270	AD3113	80	.50	1982-03-01	1982-04-01
	000030	IF1000	10	.50	1982-06-01	1983-01-01
	000130	IF1000	90	1.00	1982-01-01	1982-10-01
	000130	IF1000	100	.50	1982-10-01	1983-01-01
	000140	IF1000	90	.50	1982-10-01	1983-01-01
	000030	IF2000	10	.50	1982-01-01	1983-01-01
	000140	IF2000	100	1.00	1982-01-01	1982-03-01
	000140	IF2000	100	.50	1982-03-01	1982-07-01
	000140	IF2000	110	.50	1982-03-01	1982-07-01
	000140	IF2000	110	.50	1982-10-01	1983-01-01
	000010	MA2100	10	.50	1982-01-01	1982-11-01
	000110	MA2100	20	1.00	1982-01-01	1982-03-01
	000010	MA2110	10	1.00	1982-01-01	1983-02-01
	000200	MA2111	50	1.00	1982-01-01	1982-06-15
	000200	MA2111	60	1.00	1982-06-15	1983-02-01
	000220	MA2111	40	1.00	1982-01-01	1983-02-01
	000150	MA2112	60	1.00	1982-01-01	1982-07-15
	000150	MA2112	180	1.00	1982-07-15	1983-02-01
	000170	MA2112	60	1.00	1982-01-01	1983-06-01
	000170	MA2112	70	1.00	1982-06-01	1983-02-01
	000190	MA2112	70	1.00	1982-02-01	1982-10-01
	000190	MA2112	80	1.00	1982-10-01	1983-10-01
	000160	MA2113	60	1.00	1982-07-15	1983-02-01
	000170	MA2113	80	1.00	1982-01-01	1983-02-01
	000180	MA2113	70	1.00	1982-04-01	1982-06-15
	000210	MA2113	80	.50	1982-10-01	1983-02-01
	000210	MA2113	180	.50	1982-10-01	1983-02-01
	000050	OP1000	10	.25	1982-01-01	1983-02-01
	000090	OP1010	10	1.00	1982-01-01	1983-02-01
	000280	OP1010	130	1.00	1982-01-01	1983-02-01
	000290	OP1010	130	1.00	1982-01-01	1983-02-01
	000300	OP1010	130	1.00	1982-01-01	1983-02-01
	000310	OP1010	130	1.00	1982-01-01	1983-02-01
	000050	OP2010	10	.75	1982-01-01	1983-02-01
	000100	OP2010	10	1.00	1982-01-01	1983-02-01
	000320	OP2011	140	.75	1982-01-01	1983-02-01

Tablas de base de datos de ejemplo

Name:	EMPNO	PROJNO	ACTNO	EMPTIME	EMSTDATE	EMENDATE
	000320	OP2011	150	.25	1982-01-01	1983-02-01
	000330	OP2012	140	.25	1982-01-01	1983-02-01
	000330	OP2012	160	.75	1982-01-01	1983-02-01
	000340	OP2013	140	.50	1982-01-01	1983-02-01
	000340	OP2013	170	.50	1982-01-01	1983-02-01
	000020	PL2100	30	1.00	1982-01-01	1982-09-15

Tabla EMP_PHOTO

Name:	EMPNO	PHOTO_FORMAT	PICTURE
Type:	char(6) not null	varchar(10) not null	blob(100k)
Desc:	Employee number	Photo format	Photo of employee
Values:	000130	bitmap	db200130.bmp
	000130	gif	db200130.gif
	000130	xwd	db200130.xwd
	000140	bitmap	db200140.bmp
	000140	gif	db200140.gif
	000140	xwd	db200140.xwd
	000150	bitmap	db200150.bmp
	000150	gif	db200150.gif
	000150	xwd	db200150.xwd
	000190	bitmap	db200190.bmp
	000190	gif	db200190.gif
	000190	xwd	db200190.xwd

- “Foto de Quintana” en la página 87 muestra la foto del empleado Delores Quintana.
- “Foto de Nicholls” en la página 88 muestra la foto del empleado Heather Nicholls.
- “Foto de Adamson” en la página 90 muestra la foto del empleado Bruce Adamson.
- “Foto de Walker” en la página 91 muestra la foto del empleado James Walker.

Tabla EMP_RESUME

Name:	EMPNO	RESUME_FORMAT	RESUME
Type:	char(6) not null	varchar(10) not null	clob(5k)
Desc:	Employee number	Resume Format	Resume of employee
Values:	000130	ascii	db200130.asc

Tablas de base de datos de ejemplo

Name:	EMPNO	RESUME_FORMAT	RESUME
	000130	script	db200130.scr
	000140	ascii	db200140.asc
	000140	script	db200140.scr
	000150	ascii	db200150.asc
	000150	script	db200150.scr
	000190	ascii	db200190.asc
	000190	script	db200190.scr

- “Currículum de Quintana” en la página 87 muestra el currículum del empleado Delores Quintana.
- “Currículum de Nicholls” en la página 89 muestra el currículum del empleado Heather Nicholls.
- “Currículum de Adamson” en la página 90 muestra el currículum del empleado Bruce Adamson.
- “Currículum de Walker” en la página 92 muestra el currículum del empleado James Walker.

Tabla IN_TRAY

Name:	RECEIVED	SOURCE	SUBJECT	NOTE_TEXT
Type:	timestamp	char(8)	char(64)	varchar(3000)
Desc:	Date and Time received	User id of person sending note	Brief description	The note

Tabla ORG

Name:	DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
Type:	smallint not null	varchar(14)	smallint	varchar(10)	varchar(13)
Desc:	Department number	Department name	Manager number	Division of corporation	City
Values:	10	Head Office	160	Corporate	New York
	15	New England	50	Eastern	Boston
	20	Mid Atlantic	10	Eastern	Washington
	38	South Atlantic	30	Eastern	Atlanta
	42	Great Lakes	100	Midwest	Chicago
	51	Plains	140	Midwest	Dallas
	66	Pacific	270	Western	San Francisco
	84	Mountain	290	Western	Denver

Tabla PROJECT

Name:	PROJNO	PROJNAME	DEPTNO	RESPEMP	PRSTAFF	PRSTDATE	PRENDATE	MAJPROJ
Type:	char(6) not null	varchar(24) not null	char(3) not null	char(6) not null	dec(5,2)	date	date	char(6)
Desc:	Project number	Project name	Department responsible	Employee responsible	Estimated mean staffing	Estimated start date	Estimated end date	Major project, for a subproject
Values:	AD3100	ADMIN SERVICES	D01	000010	6.5	1982-01-01	1983-02-01	-
	AD3110	GENERAL ADMIN SYSTEMS	D21	000070	6	1982-01-01	1983-02-01	AD3100
	AD3111	PAYROLL PROGRAMMING	D21	000230	2	1982-01-01	1983-02-01	AD3110
	AD3112	PERSONNEL PROGRAMMING	D21	000250	1	1982-01-01	1983-02-01	AD3110
	AD3113	ACCOUNT PROGRAMMING	D21	000270	2	1982-01-01	1983-02-01	AD3110
	IF1000	QUERY SERVICES	C01	000030	2	1982-01-01	1983-02-01	-
	IF2000	USER EDUCATION	C01	000030	1	1982-01-01	1983-02-01	-
	MA2100	WELD LINE AUTOMATION	D01	000010	12	1982-01-01	1983-02-01	-
	MA2110	W L PROGRAMMING	D11	000060	9	1982-01-01	1983-02-01	MA2100
	MA2111	W L PROGRAM DESIGN	D11	000220	2	1982-01-01	1982-12-01	MA2110
	MA2112	W L ROBOT DESIGN	D11	000150	3	1982-01-01	1982-12-01	MA2110
	MA2113	W L PROD CONT PROGS	D11	000160	3	1982-02-15	1982-12-01	MA2110
	OP1000	OPERATION SUPPORT	E01	000050	6	1982-01-01	1983-02-01	-
	OP1010	OPERATION	E11	000090	5	1982-01-01	1983-02-01	OP1000
	OP2000	GEN SYSTEMS SERVICES	E01	000050	5	1982-01-01	1983-02-01	-
	OP2010	SYSTEMS SUPPORT	E21	000100	4	1982-01-01	1983-02-01	OP2000
	OP2011	SCP SYSTEMS SUPPORT	E21	000320	1	1982-01-01	1983-02-01	OP2010
	OP2012	APPLICATIONS SUPPORT	E21	000330	1	1982-01-01	1983-02-01	OP2010
	OP2013	DB/DC SUPPORT	E21	000340	1	1982-01-01	1983-02-01	OP2010
	PL2100	WELD LINE PLANNING	B01	000020	1	1982-01-01	1982-09-15	MA2100

Tablas de base de datos de ejemplo

Tabla SALES

Name:	SALES_DATE	SALES_PERSON	REGION	SALES
Type:	date	varchar(15)	varchar(15)	int
Desc:	Date of sales	Employee's last name	Region of sales	Number of sales
Values:	12/31/1995	LUCCHESSI	Ontario-South	1
	12/31/1995	LEE	Ontario-South	3
	12/31/1995	LEE	Quebec	1
	12/31/1995	LEE	Manitoba	2
	12/31/1995	GOUNOT	Quebec	1
	03/29/1996	LUCCHESSI	Ontario-South	3
	03/29/1996	LUCCHESSI	Quebec	1
	03/29/1996	LEE	Ontario-South	2
	03/29/1996	LEE	Ontario-North	2
	03/29/1996	LEE	Quebec	3
	03/29/1996	LEE	Manitoba	5
	03/29/1996	GOUNOT	Ontario-South	3
	03/29/1996	GOUNOT	Quebec	1
	03/29/1996	GOUNOT	Manitoba	7
	03/30/1996	LUCCHESSI	Ontario-South	1
	03/30/1996	LUCCHESSI	Quebec	2
	03/30/1996	LUCCHESSI	Manitoba	1
	03/30/1996	LEE	Ontario-South	7
	03/30/1996	LEE	Ontario-North	3
	03/30/1996	LEE	Quebec	7
	03/30/1996	LEE	Manitoba	4
	03/30/1996	GOUNOT	Ontario-South	2
	03/30/1996	GOUNOT	Quebec	18
	03/30/1996	GOUNOT	Manitoba	1
	03/31/1996	LUCCHESSI	Manitoba	1
	03/31/1996	LEE	Ontario-South	14
	03/31/1996	LEE	Ontario-North	3
	03/31/1996	LEE	Quebec	7
	03/31/1996	LEE	Manitoba	3
	03/31/1996	GOUNOT	Ontario-South	2
	03/31/1996	GOUNOT	Quebec	1
	04/01/1996	LUCCHESSI	Ontario-South	3
	04/01/1996	LUCCHESSI	Manitoba	1
	04/01/1996	LEE	Ontario-South	8
	04/01/1996	LEE	Ontario-North	-
	04/01/1996	LEE	Quebec	8
	04/01/1996	LEE	Manitoba	9
	04/01/1996	GOUNOT	Ontario-South	3

Tablas de base de datos de ejemplo

Name:	SALES_DATE	SALES_PERSON	REGION	SALES
	04/01/1996	GOUNOT	Ontario-North	1
	04/01/1996	GOUNOT	Quebec	3
	04/01/1996	GOUNOT	Manitoba	7

Tabla STAFF

Name:	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
Type:	smallint not null	varchar(9)	smallint	char(5)	smallint	dec(7,2)	dec(7,2)
Desc:	Employee number	Employee name	Department number	Job type	Years of service	Current salary	Commission
Values:	10	Sanders	20	Mgr	7	18357.50	-
	20	Pernal	20	Sales	8	18171.25	612.45
	30	Marenghi	38	Mgr	5	17506.75	-
	40	O'Brien	38	Sales	6	18006.00	846.55
	50	Hanes	15	Mgr	10	20659.80	-
	60	Quigley	38	Sales	-	16808.30	650.25
	70	Rothman	15	Sales	7	16502.83	1152.00
	80	James	20	Clerk	-	13504.60	128.20
	90	Koonitz	42	Sales	6	18001.75	1386.70
	100	Plotz	42	Mgr	7	18352.80	-
	110	Ngan	15	Clerk	5	12508.20	206.60
	120	Naughton	38	Clerk	-	12954.75	180.00
	130	Yamaguchi	42	Clerk	6	10505.90	75.60
	140	Fraye	51	Mgr	6	21150.00	-
	150	Williams	51	Sales	6	19456.50	637.65
	160	Molinare	10	Mgr	7	22959.20	-
	170	Kermisch	15	Clerk	4	12258.50	110.10
	180	Abrahams	38	Clerk	3	12009.75	236.50
	190	Sneider	20	Clerk	8	14252.75	126.50
	200	Scoutten	42	Clerk	-	11508.60	84.20
	210	Lu	10	Mgr	10	20010.00	-
	220	Smith	51	Sales	7	17654.50	992.80
	230	Lundquist	51	Clerk	3	13369.80	189.65
	240	Daniels	10	Mgr	5	19260.25	-
	250	Wheeler	51	Clerk	6	14460.00	513.30
	260	Jones	10	Mgr	12	21234.00	-
	270	Lea	66	Mgr	9	18555.50	-
	280	Wilson	66	Sales	9	18674.50	811.50

Tablas de base de datos de ejemplo

Name:	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
	290	Quill	84	Mgr	10	19818.00	-
	300	Davis	84	Sales	5	15454.50	806.10
	310	Graham	66	Sales	13	21000.00	200.30
	320	Gonzales	66	Sales	4	16858.20	844.00
	330	Burke	66	Clerk	1	10988.00	55.50
	340	Edwards	84	Sales	7	17844.00	1285.00
	350	Gafney	84	Clerk	5	13030.50	188.00

Tabla STAFFG

Nota: STAFFG sólo se crea para páginas de códigos de doble byte.

Name:	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
Type:	smallint not null	vargraphic(9)	smallint	graphic(5)	smallint	dec(9,0)	dec(9,0)
Desc:	Employee number	Employee name	Department number	Job type	Years of service	Current salary	Commission
Values:	10	Sanders	20	Mgr	7	18357.50	-
	20	Pernal	20	Sales	8	18171.25	612.45
	30	Marenghi	38	Mgr	5	17506.75	-
	40	O'Brien	38	Sales	6	18006.00	846.55
	50	Hanes	15	Mgr	10	20659.80	-
	60	Quigley	38	Sales	-	16808.30	650.25
	70	Rothman	15	Sales	7	16502.83	1152.00
	80	James	20	Clerk	-	13504.60	128.20
	90	Koonitz	42	Sales	6	18001.75	1386.70
	100	Plotz	42	Mgr	7	18352.80	-
	110	Ngan	15	Clerk	5	12508.20	206.60
	120	Naughton	38	Clerk	-	12954.75	180.00
	130	Yamaguchi	42	Clerk	6	10505.90	75.60
	140	Fraye	51	Mgr	6	21150.00	-
	150	Williams	51	Sales	6	19456.50	637.65
	160	Molinare	10	Mgr	7	22959.20	-
	170	Kermisch	15	Clerk	4	12258.50	110.10
	180	Abrahams	38	Clerk	3	12009.75	236.50
	190	Sneider	20	Clerk	8	14252.75	126.50
	200	Scoutten	42	Clerk	-	11508.60	84.20
	210	Lu	10	Mgr	10	20010.00	-
	220	Smith	51	Sales	7	17654.50	992.80

Tablas de base de datos de ejemplo

Name:	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
	230	Lundquist	51	Clerk	3	13369.80	189.65
	240	Daniels	10	Mgr	5	19260.25	-
	250	Wheeler	51	Clerk	6	14460.00	513.30
	260	Jones	10	Mgr	12	21234.00	-
	270	Lea	66	Mgr	9	18555.50	-
	280	Wilson	66	Sales	9	18674.50	811.50
	290	Quill	84	Mgr	10	19818.00	-
	300	Davis	84	Sales	5	15454.50	806.10
	310	Graham	66	Sales	13	21000.00	200.30
	320	Gonzales	66	Sales	4	16858.20	844.00
	330	Burke	66	Clerk	1	10988.00	55.50
	340	Edwards	84	Sales	7	17844.00	1285.00
	350	Gafney	84	Clerk	5	13030.50	188.00

Archivos de ejemplo con el tipo de datos BLOB y CLOB

Esta sección muestra los datos contenidos en los archivos EMP_PHOTO (fotos de empleados) y en los archivos EMP_RESUME (currículums de empleados).

Foto de Quintana



Figura 7. Delores M. Quintana

Currículum de Quintana

El texto siguiente se encuentra en los archivos db200130.asc y db200130.scr.

Resume: Delores M. Quintana

Personal Information

Tablas de base de datos de ejemplo

Address: 1150 Eglinton Ave Mellonville, Idaho 83725
Phone: (208) 555-9933
Birthdate: September 15, 1925
Sex: Female
Marital Status: Married
Height: 5'2"
Weight: 120 lbs.

Department Information

Employee Number: 000130
Dept Number: C01
Manager: Sally Kwan
Position: Analyst
Phone: (208) 555-4578
Hire Date: 1971-07-28

Education

1965 Math and English, B.A. Adelphi University

1960 Dental Technician Florida Institute of Technology

Work History

10/91 - present Advisory Systems Analyst Producing documentation tools for engineering department.

12/85 - 9/91 Technical Writer Writer, text programmer, and planner.

1/79 - 11/85 COBOL Payroll Programmer Writing payroll programs for a diesel fuel company.

Interests

- Cooking
- Reading
- Sewing
- Remodeling

Foto de Nicholls



Figura 8. Heather A. Nicholls

Currículum de Nicholls

El texto siguiente se encuentra en los archivos db200140.asc y db200140.scr.

Resume: Heather A. Nicholls

Personal Information

Address:	844 Don Mills Ave Mellonville, Idaho 83734
Phone:	(208) 555-2310
Birthdate:	January 19, 1946
Sex:	Female
Marital Status:	Single
Height:	5'8"
Weight:	130 lbs.

Department Information

Employee Number:	000140
Dept Number:	C01
Manager:	Sally Kwan
Position:	Analyst
Phone:	(208) 555-1793
Hire Date:	1976-12-15

Education

1972	Computer Engineering, Ph.D. University of Washington
1969	Music and Physics, M.A. Vassar College

Work History

Tablas de base de datos de ejemplo

2/83 - present

Architect, OCR Development Designing the architecture of OCR products.

12/76 - 1/83

Text Programmer Optical character recognition (OCR) programming in PL/I.

9/72 - 11/76

Punch Card Quality Analyst Checking punch cards met quality specifications.

Interests

- Model railroading
- Interior decorating
- Embroidery
- Knitting

Foto de Adamson



Figura 9. Bruce Adamson

Currículum de Adamson

El texto siguiente se encuentra en los archivos db200150.asc y db200150.scr.

Resume: Bruce Adamson

Personal Information

Address:	3600 Steeles Ave Mellonville, Idaho 83757
Phone:	(208) 555-4489
Birthdate:	May 17, 1947
Sex:	Male
Marital Status:	Married
Height:	6'0"
Weight:	175 lbs.

Department Information

Tablas de base de datos de ejemplo

Employee Number: 000150
Dept Number: D11
Manager: Irving Stern
Position: Designer
Phone: (208) 555-4510
Hire Date: 1972-02-12

Education

1971 Environmental Engineering, M.Sc. Johns Hopkins University

1968 American History, B.A. Northwestern University

Work History

8/79 - present Neural Network Design Developing neural networks for machine intelligence products.

2/72 - 7/79 Robot Vision Development Developing rule-based systems to emulate sight.

9/71 - 1/72 Numerical Integration Specialist Helping bank systems communicate with each other.

Interests

- Racing motorcycles
- Building loudspeakers
- Assembling personal computers
- Sketching

Foto de Walker



Figura 10. James H. Walker

Tablas de base de datos de ejemplo

Currículum de Walker

El texto siguiente se encuentra en los archivos db200190.asc y db200190.scr.

Resume: James H. Walker

Personal Information

Address: 3500 Steeles Ave Mellonville, Idaho 83757
Phone: (208) 555-7325
Birthdate: June 25, 1952
Sex: Male
Marital Status: Single
Height: 5'11"
Weight: 166 lbs.

Department Information

Employee Number: 000190
Dept Number: D11
Manager: Irving Stern
Position: Designer
Phone: (208) 555-2986
Hire Date: 1974-07-26

Education

1974 Computer Studies, B.Sc. University of Massachusetts
1972 Linguistic Anthropology, B.A. University of Toronto

Work History

6/87 - present Microcode Design Optimizing algorithms for mathematical functions.
4/77 - 5/87 Printer Technical Support Installing and supporting laser printers.
9/74 - 3/77 Maintenance Programming Patching assembly language compiler for mainframes.

Interests

- Wine tasting
- Skiing
- Swimming
- Dancing

Apéndice B. Utilización de la biblioteca de DB2

La biblioteca de DB2 Universal Database consta de ayuda en línea, manuales (PDF y HTML) y programas de ejemplo en formato HTML. Esta sección describe la información proporcionada y cómo puede acceder a ella.

Para acceder "en línea" a información de productos, puede utilizar el Centro de Información. Para obtener más información, consulte el apartado "Acceso a información mediante el Centro de Información" en la página 109. En la Web puede visualizar información sobre tareas, manuales de DB2, resolución de problemas, programas de ejemplo e información sobre DB2.

Archivos PDF y manuales impresos sobre DB2

Información sobre DB2

La tabla siguiente clasifica los manuales de DB2 en cuatro categorías:

Información de guía y consulta sobre DB2

Estos manuales contienen información básica sobre DB2 para todas las plataformas.

Información de instalación y configuración sobre DB2

Estos manuales están pensados para un sistema DB2 que se utiliza en una plataforma determinada. Por ejemplo, existen manuales de *Guía rápida de iniciación* diferentes para DB2 sobre OS/2, Windows y plataformas basadas en UNIX.

Programas de ejemplo en HTML para varias plataformas

Estos ejemplos son la versión HTML de los programas de ejemplo que se instalan con el Cliente de desarrollo de aplicaciones. Están pensados para fines informativos y no sustituyen a los programas propiamente dichos.

Notas de release

Estos archivos contienen información de última hora que no se pudo incluir en los manuales de DB2.

Los manuales de instalación, las notas de release y las guías de aprendizaje son visualizables directamente en formato HTML desde el CD-ROM del producto. La mayoría de los manuales pueden visualizarse en formato HTML desde el CD-ROM del producto y pueden visualizarse e imprimirse en formato PDF (Adobe Acrobat) desde el CD-ROM de publicaciones de DB2.

Puede también solicitar un ejemplar impreso a IBM; vea “Solicitud de los manuales impresos” en la página 105. La tabla siguiente lista los manuales que se pueden solicitar.

En las plataformas OS/2 y Windows, puede instalar los archivos HTML en el directorio `sqllib\doc\html`. La información sobre DB2 está traducida a varios idiomas, pero no toda la información está disponible en todos los idiomas. Cuando la información no está disponible en un idioma determinado, se proporciona en el idioma inglés.

En las plataformas UNIX, puede instalar los archivos HTML en varios idiomas, en los directorios `doc/%L/html`, donde `%L` representa el entorno nacional. Para obtener más información, consulte el manual *Guía rápida de iniciación*.

Puede obtener manuales de DB2 y acceder a la información de varias maneras:

- “Visualización de información en línea” en la página 108
- “Búsqueda de información en línea” en la página 113
- “Solicitud de los manuales impresos” en la página 105
- “Impresión de los manuales PDF” en la página 104

Tabla 1. Información sobre DB2

Nombre	Descripción	Número de documento	Directorio de HTML
		Nombre de archivo PDF	
Información de guía y consulta sobre DB2			
<i>Administration Guide</i>	<p><i>Administration Guide: Planning</i> proporciona una visión general de conceptos sobre bases de datos, información sobre cuestiones de diseño (tal como el diseño lógico y físico de una base de datos) y una exposición sobre el tema de la alta disponibilidad.</p>	SC09-2946 db2d1x70	db2d0
	<p><i>Administration Guide: Implementation</i> proporciona información sobre cuestiones de implantación, tales como la implantación del diseño de base de datos, el acceso a bases de datos, la auditoría, la copia y recuperación.</p>	SC09-2944 db2d2x70	
	<p><i>Administration Guide: Performance</i> proporciona información sobre el entorno de base de datos y la evaluación y ajuste del rendimiento de aplicaciones.</p>	SC09-2945 db2d3x70	
	<p>En Norteamérica, puede solicitar los tres volúmenes del manual <i>Administration Guide</i>, en lengua inglesa, utilizando el número de documento SBOF-8934.</p>		
<i>Administrative API Reference</i>	<p>Describe las interfaces de programación de aplicaciones (las API) de DB2 y las estructuras de datos que puede utilizar para gestionar las bases de datos. Este manual también explica cómo invocar las API desde las aplicaciones.</p>	SC09-2947 db2b0x70	db2b0
<i>Application Building Guide</i>	<p>Proporciona información para configurar el entorno e instrucciones paso a paso para compilar, enlazar y ejecutar aplicaciones DB2 en Windows, OS/2 y plataformas basadas en UNIX.</p>	SC09-2948 db2axx70	db2ax
<i>Códigos de detección de APPC, CPI-C y SNA</i>	<p>Proporciona información general sobre APPC, CPI-C y los códigos de detección SNA que pueden aparecer al utilizar productos DB2 Universal Database.</p>	Sin número de documento db2apx70	db2ap
	Solo está disponible en formato HTML.		

Tabla 1. Información sobre DB2 (continuación)

Nombre	Descripción	Número de documento	Directorio de HTML
		Nombre de archivo PDF	
<i>Application Development Guide</i>	Explica cómo desarrollar aplicaciones que acceden a bases de datos DB2 mediante SQL incorporado o Java (JDBC y SQLJ). Los temas tratados incluyen la escritura de procedimientos almacenados, la escritura de funciones definidas por el usuario, la creación de tipos definidos por el usuario, la utilización de desencadenantes y el desarrollo de aplicaciones en entornos particionados o mediante sistemas federados.	SC09-2949 db2a0x70	db2a0
<i>CLI Guide and Reference</i>	Explica la forma de desarrollar aplicaciones que acceden a bases de datos DB2 a través de la Interfaz de Nivel de Llamada de DB2, que es una interfaz SQL invocable que es compatible con la especificación ODBC de Microsoft.	SC09-2950 db2l0x70	db2l0
<i>Consulta de mandatos</i>	Explica cómo utilizar el procesador de línea de mandatos y describe los mandatos de DB2 que puede utilizar para gestionar la base de datos.	GC10-3495 db2n0x70	db2n0
<i>Connectivity Supplement</i>	Proporciona información de configuración y consulta sobre cómo utilizar DB2 para AS/400, DB2 para OS/390, DB2 para MVS o DB2 para VM como peticionarios de aplicaciones DRDA con servidores DB2 Universal Database. Este manual también describe cómo utilizar servidores de aplicaciones DRDA con peticionarios de aplicaciones DB2 Connect.	Sin número de documento db2h1x70	db2h1
	Solo está disponible en los formatos HTML y PDF.		
<i>Data Movement Utilities Guide and Reference</i>	Explica cómo utilizar los programas de utilidad de DB2, tales como import, export, load, AutoLoader y DPROF, los cuales facilitan el movimiento de los datos.	SC09-2955 db2dmx70	db2dm

Tabla 1. Información sobre DB2 (continuación)

Nombre	Descripción	Número de documento	Directorio de HTML
		Nombre de archivo PDF	
<i>Centro de depósito de datos Administration Guide</i>	Proporciona información sobre cómo crear y mantener un depósito de datos utilizando el Centro de depósito de datos.	SC26-9993 db2ddx70	db2dd
<i>Centro de depósito de datos Application Integration Guide</i>	Proporciona información para ayudar a los programadores a integrar aplicaciones mediante el Centro de depósito de datos y el Gestor del Catálogo de Información.	SC26-9994 db2adx70	db2ad
<i>DB2 Connect User's Guide</i>	Proporciona conceptos, información sobre programación e información general de utilización sobre los productos DB2 Connect.	SC09-2954 db2c0x70	db2c0
<i>DB2 Query Patroller Administration Guide</i>	Proporciona una visión general sobre el funcionamiento del sistema Query Patroller de DB2, información específica de utilización y administración e información sobre tareas para los programas de utilidad administrativos de la interfaz gráfica de usuario.	SC09-2958 db2dwx70	db2dw
<i>DB2 Query Patroller User's Guide</i>	Describe cómo utilizar las herramientas y funciones de DB2 Query Patroller.	SC09-2960 db2wwx70	db2ww
<i>Glosario</i>	Proporciona definiciones de términos utilizados en DB2 y en sus componentes. Está disponible en formato HTML y en la publicación <i>Consulta de SQL</i> .	Sin número de documento db2t0x70	db2t0
<i>Image, Audio, and Video Extenders Administration and Programming</i>	Proporciona información general sobre los expansores de DB2, e información sobre la administración y configuración de los expansores de imagen, audio y vídeo, y su utilización en la programación. Incluye información de consulta, información de diagnóstico (con mensajes) y ejemplos.	SC26-9929 dmbu7x70	dmbu7
<i>Information Catalog Manager Administration Guide</i>	Proporciona información de guía para la gestión de catálogos de información.	SC26-9995 db2dix70	db2di

Tabla 1. Información sobre DB2 (continuación)

Nombre	Descripción	Número de documento	Directorio de HTML
		Nombre de archivo PDF	
<i>Information Catalog Manager Programming Guide and Reference</i>	Proporciona definiciones para las interfaces con arquitectura del Gestor del Catálogo de Información.	SC26-9997 db2bix70	db2bi
<i>Information Catalog Manager User's Guide</i>	Proporciona información sobre la utilización de la interfaz de usuario del Gestor del Catálogo de Información.	SC26-9996 db2aix70	db2ai
<i>Suplemento de instalación y configuración</i>	Sirve de guía para planificar, instalar y configurar clientes DB2 específicos de una plataforma. Este suplemento contiene información sobre la creación de enlaces, la configuración de comunicaciones de cliente y servidor, herramientas de GUI para DB2, DRDA AS, la instalación distribuida, la configuración de peticiones distribuidas y el acceso a fuentes de datos heterogéneas.	GC10-3487 db2iyx70	db2iy
<i>Consulta de mensajes</i>	Contiene los mensajes y códigos que emite DB2, el Gestor del Catálogo de Información y el Centro de depósito de datos, y describe las acciones que el usuario debe emprender. En Norteamérica, puede solicitar ambos volúmenes del manual Consulta de mensajes, en lengua inglesa, utilizando el número de documento SBOF-8932.	Volumen 1 GC10-3493 db2m1x70 Volumen 2 GC10-3494 db2m2x70	db2m0
<i>OLAP Integration Server Administration Guide</i>	Explica cómo utilizar el componente Gestor de Administración del Servidor de Integración de OLAP.	SC27-0787 db2dpx70	n/d
<i>OLAP Integration Server Metaoutline User's Guide</i>	Explica cómo crear y llenar con datos "metaoutlines" OLAP utilizando la interfaz estándar de Metaoutline OLAP (no mediante el Asistente de Metaoutline).	SC27-0784 db2upx70	n/d
<i>OLAP Integration Server Model User's Guide</i>	Explica cómo crear modelos OLAP utilizando la Interfaz de Modelos de OLAP (no mediante el Asistente de Modelos).	SC27-0783 db2lpx70	n/d

Tabla 1. Información sobre DB2 (continuación)

Nombre	Descripción	Número de documento	Directorio de HTML
		Nombre de archivo PDF	
<i>OLAP Setup and User's Guide</i>	Proporciona información de configuración e instalación sobre el Kit de arranque de OLAP.	SC27-0702 db2ipx70	db2ip
<i>OLAP Spreadsheet Add-in User's Guide for Excel</i>	Describe cómo utilizar el programa de hoja de cálculo Excel para analizar datos de OLAP.	SC27-0786 db2epx70	db2ep
<i>OLAP Spreadsheet Add-in User's Guide for Lotus 1-2-3</i>	Describe cómo utilizar el programa de hoja de cálculo Lotus 1-2-3 para analizar datos de OLAP.	SC27-0785 db2tpx70	db2tp
<i>Replication Guide and Reference</i>	Proporciona información sobre la planificación, configuración, administración y utilización de las herramientas de replicación de IBM que se ofrecen con DB2.	SC26-9920 db2e0x70	db2e0
<i>Spatial Extender Guía del usuario y de consulta</i>	Proporciona información sobre la instalación, configuración, administración, programación y resolución de problemas para el Expansor Espacial. También proporciona descripciones importantes sobre conceptos de datos espaciales y ofrece información de consulta (mensajes y SQL) que es específica del Extensor Espacial.	SC27-0701 db2sbx70	db2sb
<i>Guía de iniciación de SQL</i>	Proporciona conceptos básicos sobre SQL y ofrece ejemplos de muchas estructuras sintácticas y tareas.	GC10-3496 db2y0x70	db2y0
<i>Consulta de SQL, Volumen 1 y Volumen 2</i>	Describe la sintaxis, la semántica y las normas del lenguaje SQL. Este manual también incluye información sobre las incompatibilidades entre releases, los límites del producto y las vistas de catálogo. En Norteamérica, puede solicitar ambos volúmenes del manual <i>SQL Reference</i> , en lengua inglesa, utilizando el número de documento SBOF-8933.	Volumen 1 GC10-3497 db2s1x70 Volumen 2 GC10-3497 db2s2x70	db2s0

Tabla 1. Información sobre DB2 (continuación)

Nombre	Descripción	Número de documento	Directorio de HTML
		Nombre de archivo PDF	
<i>System Monitor Guide and Reference</i>	Describe cómo recoger distintos tipos de información sobre bases de datos y el gestor de bases de datos. Este manual explica cómo utilizar la información para comprender la actividad de una base de datos, mejorar su rendimiento y determinar la causa de los problemas.	SC09-2956	db2f0
		db2f0x70	
<i>Text Extender Administración y programación</i>	Proporciona información general sobre los expansores de DB2, e información sobre la administración y configuración del expansor de texto y su utilización en la programación. Incluye información de consulta, información de diagnóstico (con mensajes) y ejemplos.	SC26-9930	desu9
		desu9x70	
<i>Troubleshooting Guide</i>	Le ayuda a determinar la causa de los errores, realizar la recuperación para un problema y utilizar herramientas de diagnóstico en colaboración con el Servicio de Asistencia al Cliente de DB2.	GC09-2850	db2p0
		db2p0x70	
<i>Novedades</i>	Describe las nuevas características, funciones y mejoras de DB2 Universal Database, Versión 7.	GC10-3498	db2q0
		db2q0x70	
Información de instalación y configuración sobre DB2			
<i>DB2 Connect Enterprise Edition para OS/2 y Windows Guía rápida de iniciación, Versión 7</i>	Proporciona información sobre la planificación, migración, instalación y configuración de DB2 Connect Enterprise Edition en los sistemas operativos OS/2 y Sistemas operativos Windows de 32 bits. Este manual también contiene información sobre la instalación y configuración de muchos clientes a los que se da soporte.	GC10-3486	db2c6
		db2c6x70	
<i>DB2 Connect Enterprise Edition para UNIX Guía rápida de iniciación</i>	Ofrece información sobre la planificación, migración, instalación, configuración y realización de tareas para DB2 Connect Enterprise Edition en plataformas basadas en UNIX. Este manual también contiene información sobre la instalación y configuración de muchos clientes a los que se da soporte.	GC10-3485	db2cy
		db2cyx70	

Tabla 1. Información sobre DB2 (continuación)

Nombre	Descripción	Número de documento	Directorio de HTML
		Nombre de archivo PDF	
<i>DB2 Connect Personal Edition Quick Beginnings</i>	Proporciona información sobre la planificación, migración, instalación, configuración y realización de tareas para DB2 Connect Personal Edition en el OS/2 y Sistemas operativos Windows de 32 bits. Este manual también contiene información sobre la instalación y configuración de todos los clientes a los que se da soporte.	GC09-2967	db2c1
DB2 Connect Personal Edition Quick Beginnings for Linux	Proporciona información sobre la planificación, instalación, migración y configuración de DB2 Connect Personal Edition en todas las distribuciones Linux soportadas.	GC09-2962	db2c4
<i>DB2 Data Links Manager Guía rápida de iniciación</i>	Proporciona información sobre la planificación, instalación, configuración y realización de tareas en DB2 Data Links Manager para los sistemas operativos AIX y Windows de 32 bits.	GC10-3488	db2z6
<i>DB2 Enterprise - Extended Edition for UNIX Quick Beginnings</i>	Ofrece información sobre la planificación, instalación y configuración de DB2 Enterprise - Extended Edition en plataformas basadas en UNIX. Este manual también contiene información sobre la instalación y configuración de muchos clientes a los que se da soporte.	GC09-2964	db2v3
<i>DB2 Enterprise - Extended Edition for Windows Quick Beginnings</i>	Proporciona información sobre la planificación, instalación, configuración de DB2 Enterprise - Extended Edition para los sistemas operativos Windows de 32 bits. Este manual también contiene información sobre la instalación y configuración de muchos clientes a los que se da soporte.	GC09-2963	db2v6

Tabla 1. Información sobre DB2 (continuación)

Nombre	Descripción	Número de documento	Directorio de HTML
		Nombre de archivo PDF	
<i>DB2 para OS/2 Guía rápida de iniciación</i>	Ofrece información sobre la planificación, instalación, migración y configuración de DB2 Universal Database en el sistema operativo OS/2. Este manual también contiene información sobre la instalación y configuración de muchos clientes a los que se da soporte.	GC10-3489 db2i2x70	db2i2
<i>DB2 para UNIX Guía rápida de iniciación</i>	Ofrece información sobre la planificación, instalación, migración y configuración de DB2 Universal Database en plataformas basadas en UNIX. Este manual también contiene información sobre la instalación y configuración de muchos clientes a los que se da soporte.	GC10-3491 db2ixx70	db2ix
<i>DB2 para Windows Guía rápida de iniciación</i>	Proporciona información sobre la planificación, instalación, migración y configuración de DB2 Universal Database en Sistemas operativos Windows de 32 bits. Este manual también contiene información sobre la instalación y configuración de muchos clientes a los que se da soporte.	GC10-3492 db2i6x70	db2i6
<i>DB2 Personal Edition Guía rápida de iniciación</i>	Proporciona información sobre la planificación, instalación, migración y configuración de DB2 Universal Database Personal Edition en el OS/2 y Sistemas operativos Windows de 32 bits.	GC10-3490 db2i1x70	db2i1
DB2 Personal Edition Quick Beginnings for Linux	Proporciona información sobre la planificación, instalación, migración y configuración de DB2 Universal Database Personal Edition en todas las distribuciones Linux soportadas.	GC09-2972 db2i4x70	db2i4
<i>DB2 Query Patroller Installation Guide</i>	Proporciona información sobre la instalación de DB2 Query Patroller.	GC09-2959 db2iwx70	db2iw

Tabla 1. Información sobre DB2 (continuación)

Nombre	Descripción	Número de documento	Directorio de HTML
		Nombre de archivo PDF	
DB2 Warehouse Manager Installation Guide	Proporciona información sobre la instalación de agentes de depósito, transformadores de depósito y el Gestor del Catálogo de Información.	GC26-9998	db2id
Programas de ejemplo en HTML para varias plataformas			
Programas de ejemplo en HTML	Proporciona los programas de ejemplo en formato HTML para los lenguajes de programación de todas las plataformas soportadas por DB2. Los programas de ejemplo se ofrecen sólo con fines informativos. No todos los programas de ejemplo están disponibles en todos los lenguajes de programación. Los ejemplos en formato HTML sólo pueden utilizarse si está instalado DB2 Application Development Client. Para obtener más información sobre los programas, consulte el manual <i>Application Building Guide</i> .	Sin número de documento	db2hs
Notas de release			
<i>DB2 Connect Release Notes</i>	Proporciona información de última hora que no se pudo incluir en los manuales de DB2 Connect.	Véase nota #2.	db2cr
<i>DB2 Installation Notes</i>	Proporciona información de última hora, específica de la instalación, que no se pudo incluir en los manuales de DB2.	Sólo disponible en el CD-ROM del producto.	
<i>DB2 Release Notes</i>	Proporciona información de última hora, referente a todos los productos y características de DB2, que no se pudo incluir en los manuales de DB2.	Véase nota #2.	db2ir

Notas:

1. El carácter x que ocupa la sexta posición en el nombre de archivo indica el idioma en que está escrito el manual. Por ejemplo, el nombre de archivo db2d0e70 identifica la versión inglesa del manual *Administration Guide* y el nombre de archivo db2d0f70 identifica la versión francesa del mismo manual. En la posición sexta de los nombres de archivo se utilizan las letras siguientes para indicar el idioma del manual:

Idioma	Identificador
Portugués brasileño	b
Búlgaro	u
Checo	x
Danés	d
Holandés	q
Inglés	e
Finés	y
Francés	f
Alemán	g
Griego	a
Húngaro	h
Italiano	i
Japonés	j
Coreano	k
Noruego	n
Polaco	p
Portugués	v
Ruso	r
Chino simplificado	c
Esloveno	l
Español	z
Sueco	s
Chino tradicional	t
Turco	m

2. La información de última hora que no se pudo incluir en los manuales de DB2 se encuentra en las Notas de Release, en formato HTML y en forma de archivo ASCII. La versión en formato HTML puede consultarse desde el Centro de Información y en los CD-ROM del producto. Para visualizar el archivo ASCII:

- En las plataformas basadas en UNIX, vea el archivo Release.Notes. Este archivo está situado en el directorio DB2DIR/Readme/%L, donde %L representa el entorno nacional y DB2DIR representa:
 - /usr/lpp/db2_07_01 en AIX
 - /opt/IBMdb2/V7.1 en HP-UX, PTX, Solaris, y Silicon Graphics IRIX
 - /usr/IBMdb2/V7.1 en Linux.
- En otras plataformas, vea el archivo RELEASE.TXT. Este archivo reside en el directorio donde está instalado el producto. En las plataformas OS/2, puede también hacer una doble pulsación sobre la carpeta **IBM DB2** y luego sobre el icono **Release Notes**.

Impresión de los manuales PDF

Si prefiere tener copias impresas de los manuales, puede imprimir los archivos PDF contenidos en el CD-ROM de publicaciones de DB2. Mediante Adobe

Acrobat Reader, puede imprimir el manual completo o un rango específico de páginas. Para conocer el nombre de archivo de cada manual de la biblioteca, vea la Tabla 1 en la página 95.

Puede obtener la última versión de Adobe Acrobat Reader en el sitio Web de Adobe, que se encuentra en <http://www.adobe.com>.

Los archivos PDF contenidos en el CD-ROM de publicaciones de DB2 tienen PDF como extensión de archivo. Para acceder a los archivos PDF:

1. Inserte el CD-ROM de publicaciones de DB2. En las plataformas basadas en UNIX, monte el CD-ROM de publicaciones de DB2. Consulte el manual *Guía rápida de iniciación* para conocer los procedimientos de montaje del CD-ROM.
2. Arranque Acrobat Reader.
3. Abra el archivo PDF deseado que se encuentra en una de las ubicaciones siguientes:
 - En las plataformas OS/2 y Windows:
el directorio `x:\doc\idioma`, donde `x` representa la unidad de CD-ROM e `idioma` representa el código de país de dos caracteres correspondiente al idioma del usuario (por ejemplo, EN para el inglés).
 - En plataformas basadas en UNIX:
el directorio `/cdrom/doc/%L` del CD-ROM, donde `/cdrom` representa el punto de montaje del CD-ROM y `%L` representa el entorno nacional deseado.

Puede también copiar los archivos PDF del CD-ROM a una unidad local o de red y leerlos desde allí.

Solicitud de los manuales impresos

Puede solicitar los manuales impresos de DB2 en forma individual o como colección de manuales (en Norteamérica sólo), utilizando en este segundo caso un número de documento SBOF (sold bill of forms). Para solicitar manuales, consulte al concesionario o representante de ventas autorizado de IBM, o llame a los números 1-800-879-2755 (Estados Unidos) o 1-800-IBM-4YOU (Canadá). Puede también solicitar manuales desde la página Web de publicaciones, situada en <http://www.elink.ibm.com/pbl/pbl>.

Puede adquirir dos colecciones de manuales. SBOF-8935 proporciona información de consulta y de utilización sobre DB2 Warehouse Manager. SBOF-8935 proporciona información de consulta y de utilización sobre todos los demás productos y características de DB2 Universal Database. La tabla siguiente lista el contenido de cada colección de manuales:

Tabla 2. Pedido de los manuales impresos

Número SBOF	Manuales incluidos
SBOF-8931	<ul style="list-style-type: none"> • Administration Guide: Planning • Administration Guide: Implementation • Administration Guide: Performance • Administrative API Reference • Application Building Guide • Application Development Guide • CLI Guide and Reference • Command Reference • Data Movement Utilities Guide and Reference • Data Warehouse Center Administration Guide • Data Warehouse Center Application Integration Guide • DB2 Connect User's Guide • Installation and Configuration Supplement • Image, Audio, and Video Extenders Administration and Programming • Message Reference, Volúmenes 1 y 2 • OLAP Integration Server Administration Guide • OLAP Integration Server Metaoutline User's Guide • OLAP Integration Server Model User's Guide • OLAP Integration Server User's Guide • OLAP Setup and User's Guide • OLAP Spreadsheet Add-in User's Guide for Excel • OLAP Spreadsheet Add-in User's Guide for Lotus 1-2-3 • Replication Guide and Reference • Spatial Extender Administration and Programming Guide • SQL Getting Started • SQL Reference, Volúmenes 1 y 2 • System Monitor Guide and Reference • Text Extender Administration and Programming • Troubleshooting Guide • What's New
SBOF-8935	<ul style="list-style-type: none"> • Information Catalog Manager Administration Guide • Information Catalog Manager User's Guide • Information Catalog Manager Programming Guide and Reference • Query Patroller Administration Guide • Query Patroller User's Guide

Documentación en línea de DB2

Acceso a la ayuda en línea

Existe ayuda en línea para todos los componentes de DB2. La tabla siguiente describe los diversos tipos de ayuda.

Tipo de ayuda	Contenido	Cómo acceder...
Ayuda para mandatos	Explica la sintaxis de los mandatos del procesador de línea de mandatos.	<p>Desde el procesador de línea de mandatos en modalidad interactiva, especifique:</p> <p style="padding-left: 40px;"><i>? mandato</i></p> <p>donde <i>mandato</i> representa una palabra clave o el mandato completo.</p> <p>Por ejemplo, <i>? catalog</i> visualiza ayuda para todos los mandatos CATALOG, mientras que <i>? catalog database</i> visualiza ayuda para el mandato CATALOG DATABASE.</p>
Ayuda para el Asistente de configuración del cliente	Explica las tareas que el usuario puede realizar en una ventana o cuaderno. La ayuda incluye información general e información sobre los requisitos previos que debe conocer, y describe cómo utilizar los controles de una ventana o cuaderno.	Desde una ventana o cuaderno, pulse el botón Ayuda o pulse la tecla F1 .
Ayuda para el Centro de mandatos		
Ayuda para Centro de control		
Ayuda para Centro de depósito de datos		
Ayuda para el Analizador de sucesos		
Ayuda para el Gestor del catálogo de información		
Ayuda para el Centro de administración de satélites		
Ayuda para el Centro de scripts		

Tipo de ayuda	Contenido	Cómo acceder...
Ayuda para mensajes	Describe la causa de un mensaje y la acción que debe realizar el usuario.	<p>Desde el procesador de línea de mandatos en modalidad interactiva, especifique:</p> <pre>? XXXnnnnn</pre> <p>donde <i>XXXnnnnn</i> representa un identificador válido de mensaje.</p> <p>Por ejemplo, ? SQL30081 muestra ayuda sobre el mensaje SQL30081.</p> <p>Para ver la ayuda sobre mensajes pantalla a pantalla, especifique:</p> <pre>? XXXnnnnn more</pre> <p>Para guardar la ayuda sobre el mensaje en un archivo, especifique:</p> <pre>? XXXnnnnn > nombreachivo.ext</pre> <p>donde <i>nombreachivo.ext</i> representa el archivo en el que desea guardar la ayuda referente al mensaje.</p>
Ayuda para SQL	Explica la sintaxis de las sentencias de SQL.	<p>Desde el procesador de línea de mandatos en modalidad interactiva, especifique:</p> <pre>help sentencia</pre> <p>donde <i>sentencia</i> representa una sentencia de SQL.</p> <p>Por ejemplo, help SELECT visualiza ayuda sobre la sentencia SELECT.</p> <p>Nota: En las plataformas basadas en UNIX no existe ayuda para SQL.</p>
Ayuda para SQLSTATE	Explica los estados y códigos de clase del SQL.	<p>Desde el procesador de línea de mandatos en modalidad interactiva, especifique:</p> <pre>? estado_sql o ? código_clase</pre> <p>donde <i>estado_sql</i> representa un estado SQL válido de cinco dígitos y <i>código_clase</i> representa los dos primeros dígitos del estado SQL.</p> <p>Por ejemplo, ? 08003 visualiza ayuda para el estado SQL 08003, mientras que ? 08 visualiza ayuda para el código de clase 08.</p>

Visualización de información en línea

Los manuales que se incluyen con el presente producto están en copia software, en el formato HTML (Hypertext Markup Language). El formato en copia software le permite buscar o examinar información y proporciona

enlaces de hipertexto con información afín. También facilita la utilización compartida de la biblioteca en el sitio Web.

Puede visualizar los manuales en línea o programas de ejemplo mediante cualquier navegador que cumpla las especificaciones de HTML Versión 3.2.

Para visualizar manuales en línea o programas de ejemplo:

- Si está ejecutando herramientas de administración de DB2, utilice el Centro de Información.
- Desde un navegador, pulse **Archivo** —> **Abrir página**. La página que se abre contiene descripciones y enlaces que conducen a información sobre DB2.
 - En las plataformas basadas en UNIX, abra la página siguiente:

```
INSTHOME/sqllib/doc/%L/html/index.htm
```

donde %L representa el entorno nacional.

- En otras plataformas, abra la página siguiente:

```
sqllib\doc\html\index.htm
```

La vía de acceso se encuentra en la unidad donde está instalado DB2.

Si no ha instalado el Centro de Información, puede abrir la página efectuando una doble pulsación sobre el icono **Información de DB2**. Según cuál sea el sistema que esté utilizando, el icono se encuentra en la carpeta principal del producto o en el menú Inicio de Windows.

Instalación del navegador Netscape

Si no tiene todavía un navegador Web instalado, puede instalar Netscape desde el CD-ROM proporcionado con el producto. Para obtener instrucciones detalladas sobre cómo instalarlo, siga los pasos siguientes:

1. Inserte el CD-ROM de Netscape.
2. Si utiliza una plataforma basada en UNIX, monte el CD-ROM. Consulte el manual *Guía rápida de iniciación* para conocer los procedimientos de montaje del CD-ROM.
3. Para obtener instrucciones sobre la instalación, consulte el archivo CDNAVnn.txt, donde nn representa el identificador de dos caracteres correspondiente a su idioma. El archivo está situado en el directorio raíz del CD-ROM.

Acceso a información mediante el Centro de Información

El Centro de Información proporciona acceso rápido a información sobre los productos DB2. El Centro de Información está disponible en todas las plataformas en las que pueden utilizarse las herramientas de administración de DB2.

Para abrir el Centro de Información, haga una doble pulsación sobre su icono. Según cuál sea el sistema que esté utilizando, el icono se encuentra en la carpeta principal del producto o en el menú **Inicio** de Windows.

También puede acceder al Centro de Información utilizando la barra de herramientas y el menú **Ayuda** en la plataforma Windows para DB2.

El Centro de Información proporciona seis tipos de información. Pulse la pestaña adecuada para consultar el tipo de información correspondiente.

Tareas Tareas esenciales que puede realizar mediante DB2.

Consulta Información de consulta sobre DB2, tal como palabras clave, mandatos y las API.

Manuales Manuales de DB2.

Resolución de problemas

Categorías de mensajes de error y sus acciones de recuperación.

Programas de ejemplo

Programas de ejemplo que se proporcionan con el Cliente de Desarrollo de Aplicaciones de DB2. Si no instaló el Cliente de Desarrollo de Aplicaciones de DB2, esta pestaña no se visualiza.

Web

Información sobre DB2 disponible en la World Wide Web. Para acceder a esta información, debe tener una conexión con la Web desde su sistema.

Cuando selecciona un elemento de una de estas listas, el Centro de Información abre un visor para mostrar la información. El visor puede ser el visor de ayuda del sistema, un editor o un navegador Web, dependiendo del tipo de información que seleccione.

El Centro de Información proporciona una función de búsqueda, que le permite buscar un tema determinado sin examinar las listas.

Para realizar una búsqueda de texto completa, siga el enlace de hipertexto del Centro de Información que conduce al formulario de búsqueda **Buscar información en línea sobre DB2**.

Normalmente, el servidor de búsqueda HTML arranca automáticamente. Si una búsqueda en la información HTML no funciona, puede que deba arrancar el servidor de búsqueda siguiendo uno de los métodos siguientes:

En Windows

Pulse **Inicio** y seleccione **Programas** —> **IBM DB2** —> **Información** —> **Iniciar servidor de búsqueda HTML**.

En OS/2

Haga una doble pulsación sobre la carpeta **DB2 para OS/2** y luego sobre el icono **Iniciar servidor de búsqueda HTML**.

Consulte las notas de release si tiene cualquier otro problema al buscar la información HTML.

Nota: La función de búsqueda no puede utilizarse en los entornos Linux, PTX ni Silicon Graphics IRIX.

Utilización de los asistentes de DB2

Los asistentes ("wizards") le ayudan a realizar tareas de administración determinadas mediante instrucciones a paso a paso. Puede acceder a los asistentes mediante el Centro de control y el Asistente de configuración de cliente. La tabla siguiente lista los asistentes y describe su función.

Nota: Los asistentes para Crear base de datos, Crear índice, Configurar actualización múltiple y Configuración del rendimiento pueden utilizarse en el entorno de base de datos particionada.

Asistente	Le ayuda a...	Cómo acceder...
Añadir base de datos	Catalogar una base de datos en una estación de trabajo cliente.	En el Asistente de configuración del cliente, pulse Añadir .
Hacer copia de seguridad de base de datos	Determinar, crear y planificar un plan de copia de seguridad.	En el Centro de Control, pulse con el botón derecho del ratón sobre la base de datos que desea copiar y seleccione Copia de seguridad —> Base de datos utilizando asistente .
Configurar actualización múltiple	Realizar una actualización múltiple, una transacción distribuida o una operación de confirmación de dos fases.	En el Centro de Control, pulse con el botón derecho del ratón sobre la carpeta Bases de datos y seleccione Actualización múltiple .
Crear base de datos	Crear una base de datos y realizar algunas tareas básicas de configuración.	En el Centro de Control, pulse con el botón derecho del ratón sobre la carpeta Bases de datos y seleccione Crear —> Base de datos utilizando asistente .
Crear tabla	Seleccionar tipos de datos básicos y crear una clave primaria para la tabla.	En el Centro de Control, pulse con el botón derecho del ratón sobre el icono Tablas y seleccione Crear —> Tabla utilizando asistente .

Asistente	Le ayuda a...	Cómo acceder...
Crear espacio de tablas	Crear un nuevo espacio de tablas.	En el Centro de Control, pulse con el botón derecho del ratón sobre el icono Espacios de tablas y seleccione Crear —> Espacio de tablas utilizando asistente .
Crear índice	Determinar qué índices crear y eliminar para cada consulta.	En el Centro de Control, pulse con el botón derecho del ratón sobre el icono Índice y seleccione Crear —> Índice utilizando asistente .
Configuración del rendimiento	Ajustar el rendimiento de una base de datos actualizando los parámetros de configuración de acuerdo con sus necesidades.	En el Centro de Control, pulse con el botón derecho del ratón sobre la base de datos que desea ajustar y seleccione Configurar rendimiento utilizando asistente . Si utiliza un entorno de base de datos particionada, desde la vista Particiones de base de datos, pulse con el botón derecho del ratón sobre la primera partición de base de datos que desea ajustar y seleccione Configurar rendimiento utilizando asistente .
Restaurar base de datos	Recuperar una base de datos después de una anomalía. Le ayuda a determinar qué copia de seguridad se debe utilizar y qué archivos de anotaciones se deben aplicar.	En el Centro de Control, pulse con el botón derecho del ratón sobre la base de datos que desea restaurar y seleccione Restaurar —> Base de datos utilizando asistente .

Configuración de un servidor de documentos

Por omisión, la información sobre DB2 se instala en el sistema local. Esto significa que cada una de las personas que deba acceder a la información sobre DB2 debe instalar los mismos archivos. Para que la información sobre DB2 se almacene en una única ubicación, siga los pasos siguientes:

1. Copie todos los archivos y subdirectorios del directorio `\sqlib\doc\html`, del sistema local, en un servidor Web. Cada manual tiene su propio subdirectorio que contiene todos los archivos HTML y archivos GIF necesarios que forman el manual. Asegúrese de que la estructura de directorios permanece igual.
2. Configure el servidor Web para que busque los archivos en la nueva ubicación. Si desea obtener más información, consulte el Apéndice sobre NetQuestion que se encuentra en la publicación *Suplemento de instalación y configuración*.

3. Si está utilizando la versión Java del Centro de Información, puede especificar un URL base para todos los archivos HTML. Debe utilizar el URL para acceder a la lista de manuales.
4. Una vez que pueda visualizar los archivos del manual, puede marcar los temas que consulte con frecuencia. Probablemente deseará marcar las páginas siguientes:
 - Lista de manuales
 - Tablas de contenido de manuales utilizados con frecuencia
 - Temas consultados con frecuencia, tales como ALTERAR TABLA
 - El formulario de búsqueda

Para obtener información sobre cómo puede proporcionar los archivos de documentación en línea de DB2 Universal Database desde una máquina central, consulte el Apéndice sobre NetQuestion del manual *Suplemento de instalación y configuración*.

Búsqueda de información en línea

Para buscar información en los archivos HTML, siga uno de los métodos siguientes:

- Pulse **Buscar** en el panel superior. Utilice el formulario de búsqueda para buscar un tema determinado. La función de búsqueda no puede utilizarse en los entornos Linux, PTX ni Silicon Graphics IRIX.
- Pulse **Índice** en el panel superior. Utilice el índice para buscar un tema determinado en el manual.
- Visualice la tabla de contenido o índice de la ayuda o del manual HTML y luego utilice la función de búsqueda del navegador Web para buscar un tema determinado en el manual.
- Utilice la función de marcaje de documentos del navegador Web para volver rápidamente a un tema determinado.
- Utilice la función de búsqueda del Centro de Información para buscar temas determinados. Vea “Acceso a información mediante el Centro de Información” en la página 109 para obtener detalles.

Apéndice C. Avisos

Puede que IBM no comercialice algunos productos, servicios o características descritos en este manual. Consulte a su representante local de IBM para obtener información sobre los productos y servicios que están disponibles actualmente en su región geográfica. Cualquier referencia a un producto, programa o servicio de IBM no pretende afirmar ni implicar que sólo se puede utilizar dicho producto, programa o servicio de IBM. En su lugar se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no infrinja ninguno de los derechos de propiedad intelectual de IBM. Pero es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patentes en tramitación que afecten al tema tratado en este documento. La posesión de este documento no confiere ninguna licencia sobre dichas patentes. Puede realizar consultas escribiendo a:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

En el caso de consultas sobre licencias referentes a información de doble byte (DBCS), consulte al Departamento de Propiedad Intelectual de IBM en su país o realice consultas escribiendo a:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

El párrafo siguiente no es aplicable al Reino Unido ni a ningún país en el que tales disposiciones sean incompatibles con la legislación local:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL", SIN GARANTÍA DE NINGUNA CLASE, NI EXPLÍCITA NI IMPLÍCITA, INCLUIDAS, PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE NO VULNERACIÓN DE DERECHOS, COMERCIALIZABILIDAD O IDONEIDAD PARA UN FIN DETERMINADO. Algunos estados no permiten la exclusión de garantías expresas o implícitas en determinadas transacciones, por lo que es posible que esta declaración no sea aplicable en su caso.

Esta publicación puede contener inexactitudes técnicas o errores tipográficos. Periódicamente se efectúan cambios en la información aquí contenida; dichos cambios se incorporarán a las nuevas ediciones de la publicación. IBM puede efectuar, en cualquier momento y sin previo aviso, mejoras y/o cambios en los productos y/o programas descritos en esta publicación.

Las referencias hechas en esta publicación a sitios Web que no son de IBM se proporcionan sólo para la comodidad del usuario y no constituyen un aval de esos sitios Web. La información contenida en esos sitios Web no forma parte de la información del presente producto IBM y el usuario es responsable de la utilización de esos sitios Web.

Cuando envía información a IBM, IBM puede utilizar o distribuir dicha información en la forma en que IBM considere adecuada, sin contraer por ello ninguna obligación con el remitente.

Los licenciatarios de este programa que deseen obtener información sobre él con el fin de habilitar: (i) el intercambio de información entre programas creados de forma independiente y otros programas (incluido este) y (ii) el uso mutuo de la información intercambiada, deben ponerse en contacto con:

IBM Canada Limited
Office of the Lab Director
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

Dicha información puede estar disponible, sujeta a los términos y condiciones apropiados, incluido en algunos casos, el pago de una tarifa.

El programa bajo licencia descrito en este manual y todo el material bajo licencia asociado a él los proporciona IBM según los términos del Convenio del Cliente IBM, el Convenio Internacional de Licencia de Programas de IBM o cualquier convenio equivalente entre el usuario e IBM.

Los datos de rendimiento contenidos en este documento se obtuvieron en un entorno controlado. Por tanto, los resultados obtenidos en otros entornos operativos pueden variar significativamente. Algunas mediciones pueden haberse hecho en sistemas experimentales y no es seguro que estas mediciones sean las mismas en los sistemas disponibles comercialmente. Además, algunas mediciones pueden haberse calculado mediante extrapolación. Los resultados reales pueden variar. Los usuarios del presente manual deben verificar los datos aplicables para su entorno específico.

La información referente a productos que no son de IBM se ha obtenido de los proveedores de esos productos, de sus anuncios publicados o de otras

fuentes disponibles públicamente. IBM no ha probado esos productos y no puede confirmar la exactitud del rendimiento, la compatibilidad ni cualquier otra afirmación referente a productos no IBM. Las preguntas sobre las prestaciones de productos no IBM deben dirigirse a los proveedores de esos productos.

Todas las declaraciones de intenciones de IBM están sujetas a cambio o cancelación sin previo aviso, y sólo representan objetivos.

Esta publicación puede contener ejemplos de datos e informes que se utilizan en operaciones comerciales diarias. Para ilustrarlos de la forma más completa posible, los ejemplos incluyen nombre de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier similitud con nombres y direcciones utilizados por una empresa real es totalmente no intencionada.

LICENCIA DE COPYRIGHT:

Este manual puede contener programas de aplicación de ejemplo escritos en lenguaje fuente, que muestra técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo de la forma que desee, sin pago alguno a IBM, con los fines de desarrollar, utilizar, comercializar o distribuir programas de aplicación de acuerdo con la interfaz de programación de aplicaciones correspondiente a la plataforma operativa para la que están escritos los programas de ejemplo. Estos ejemplos no se han probado exhaustivamente bajo todas las condiciones. Por tanto, IBM no puede asegurar ni implicar la fiabilidad, utilidad o función de estos programas.

Cada copia o porción de estos programas de ejemplo o cualquier obra derivada debe incluir una nota de copyright como la siguiente:

© (nombre de la empresa) (año). Partes de este código derivan de programas de ejemplo de IBM Corp. © Copyright IBM Corp. - especifique el año u años. Reservados todos los derechos.

Marcas registradas

Los términos siguientes, que pueden estar indicados por un asterisco (*), son marcas registradas de International Business Machines Corporation en los Estados Unidos y/o en otros países.

ACF/VTAM	IBM
AISPO	IMS
AIX	IMS/ESA
AIX/6000	LAN DistanceMVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
AS/400	OS/2
BookManager	OS/390
CICS	OS/400
C Set++	PowerPC
C/370	QBIC
DATABASE 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/DS
DB2 Extenders	SQL/400
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView
DRDA	VisualAge
eNetwork	VM/ESA
Extended Services	VSE/ESA
FFST	VTAM
First Failure Support Technology	WebExplorer
	WIN-OS/2

Los términos siguientes son marcas registradas de otras empresas:

Microsoft, Windows y Windows NT son marcas registradas de Microsoft Corporation.

Java, y las marcas registradas y logotipos basados en Java, y Solaris son marcas registradas de Sun Microsystems, Inc. en los Estados Unidos y/o en otros países.

Tivoli y NetView son marcas registradas de Tivoli Systems Inc. en los Estados Unidos y/o en otros países.

UNIX es una marca registrada en los Estados Unidos y/o en otros países bajo licencia exclusiva de X/Open Company Limited.

Otros nombres de empresas, productos o servicios, que pueden estar indicados por un doble asterisco (**), pueden ser marcas registradas o marcas de servicio de otras empresas.

Índice

A

ADD CONSTRAINT, sentencia 55
ALTER TABLE, sentencia 55
ALL, uso en una consulta 52
anidamiento de subconsultas correlacionadas 45
ANY, palabra clave 52
añadir base de datos, asistente para 111, 112
AS, cláusula 26
asistente
 restaurar base de datos 112
asistentes
 añadir base de datos 111, 112
 configuración del rendimiento 112
 configurar actualización múltiple 111
 copiar base de datos 111
 crear base de datos 111
 crear espacio de tablas 111
 crear tabla 111
 índice 112
 realización de tareas 111
ayuda en línea 106

B

base de datos de ejemplo 73
 borrado 74
 creación 74
base de datos relacional, definición 1
base de datos relacional multinodal, definición 1
base de datos relacional particionada, definición 1
BETWEEN, predicado 50
biblioteca de DB2
 asistentes 111
 ayuda en línea 106
 buscar información en línea 113
 Centro de Información 109
 configuración de un servidor de documentos 112
 estructura de 93
 identificador de idioma para manuales 103
 imprimir manuales PDF 104
 información de última hora 104

biblioteca de DB2 (*continuación*)
 manuales 93
 pedido de manuales impresos 105
 visualización de información en línea 108
BIGINT, tipo de datos 5
BLOB, serie 69
BLOB, tipo de datos 69
borrado de la base de datos de ejemplo 74
buscar
 información en línea 110, 113

C

calificación de objetos 4, 17
case, expresión
 descripción 36
 función SIGN 36
catálogos del sistema 71
Centro de Información 109
CL_SCHED, tabla de ejemplo 75
clasificación de filas 22
clave
 compuesta 54
 definición 54
 exclusiva 54
 externa 54
 primaria 54
clave exclusiva 54
 restricción de unicidad 54
clave externa 54
clave primaria 54
clave principal, definición 54
CLOB, serie 69
CLOB, tipo de datos 69
columna
 ASC, clasificación por orden ascendente 23
 definición de 3
 DESC, clasificación por orden descendente 23
columna de agrupación, definición 31
combinación de consultas 47
comprobar la existencia 51
compuesta, clave 54
condición de búsqueda 20
condición de unión 60
conexión de consultas 49
configuración de un servidor de documentos 112
configuración del rendimiento, asistente de 112
configurar actualización múltiple, asistente para 111
CONNECT, sentencia 18
 explícita 18
 implícita 18
consulta de nivel externo, correlación 43
Consulta de SQL v
consultas, conexión 49
consultas recursivas, descripción 65
conversión de datos
 condiciones de unión 62
 operadores de conjunto 49
conversión de tipos de datos
 descripción 35
copiar base de datos, asistente para 111
correlación
 descripción 40
 nombre 43
 subconsulta 41
 subconsultas utilizando uniones 44
creación de la base de datos de ejemplo 74
crear base de datos, asistente para 111
crear espacio de tablas, asistente para 111
crear tabla, asistente para 111
CREATE DISTINCT TYPE 67
CREATE FUNCTION 68
CREATE TABLE, sentencia 9
 NOT NULL/NOT NULL WITH DEFAULT, valor de columna 9
CREATE TRIGGER 56
CREATE VIEW, sentencia 13
 WITH CHECK OPTION 13
CUBE 65
 filas de subtotal 65
 filas de tabulación cruzada 65
CURRENT DATE, registro especial 70
CURRENT FUNCTION PATH, registro especial 71

CURRENT SERVER, registro
especial 71
CURRENT TIME, registro
especial 71
CURRENT TIMESTAMP, registro
especial 71
CURRENT TIMEZONE, registro
especial 71

CH

CHAR, tipo de datos 5

D

DATE, tipo de datos 5
DATETIME, tipo de datos 5
DBLOB, serie 69
DBLOB, tipo de datos 69
decimal, descripción 5
DECIMAL, tipo de datos 5
DELETE, sentencia 13
DEPARTMENT, tabla de
ejemplo 75
desencadenantes
CREATE TRIGGER 56
definición 53
descripción 56
desencadenante anterior 56
desencadenante posterior 56
variables de transición 59
DISTINCT, palabra clave 24, 30
Documentación relacionada v
DOUBLE, tipo de datos 5

E

EMP_ACT, tabla de ejemplo 79
EMP_PHOTO, tabla de ejemplo 81
EMP_RESUME, tabla de ejemplo 81
EMPLOYEE, tabla de ejemplo 75
entero binario, descripción 5
esquema
definición de 4
esquemas reservados 4
estructura de datos
columna 3
fila 3
valor 3
EXCEPT, operador 48
ordenación de resultados 49
restricciones de uso referidas
a 49
tipos de datos 49
EXCEPT ALL 48
existencia, comprobación de la 51
EXISTS, predicado 51
expresión de tabla común
descripción 39

expresiones 25
expresiones, asignación de
nombres 26
expresiones de tabla
descripción 37
expresiones de tabla anidadas,
descripción 38

F

fecha y hora, descripción de
valores 5
fila
definición de 3
selección 20
filas de subtotal 65
filas de tabulación cruzada 65
FLOAT, tipo de datos 5
FROM, cláusula 19
FULL OUTER, unión 60
función
de columna 28
de tabla 30
definida por el usuario 28
descripción 28
escalar 28
incorporada 28
Online Analytical Processing
(OLAP) 66
función de columna 28
AVG 29
COUNT 29
MAX 29
MIN 29
función de tabla
SQLCACHE_SNAPSHOT 30
función de tabla externa 68
función de tabla externa DB
OLE 68
función escalar 28
ABS 30
DECIMAL 38
HEX 30
LENGTH 30
SIGN 30
YEAR 30
función escalar externa 68
función fuente 68
funciones de columna 29
funciones definidas por el
usuario 68
definición 68
función de tabla externa 68
función de tabla externa DB
OLE 68
función escalar externa 68
función fuente 68

funciones OLAP 66
grupo de agregación 66
ordenación de filas en 66
particionamiento de filas en 66
fusión del resultado de
consultas 47

G

gestor de bases de datos 1
GROUP BY 24
GROUP BY, cláusula
columna de agrupación 31
con cláusula HAVING 32
Guía de administración v
Guía de desarrollo de
aplicaciones v
Guía rápida de iniciación v

H

HAVING 24
HAVING, cláusula
descripción 32
HTML
programas de ejemplo 103

I

ID de autorización 4
identificador de idioma
manuales 103
imprimir manuales PDF 104
IN, predicado 50
IN_TRAY, tabla de ejemplo 82
indicador de posición 70
índice, asistente de 112
información de última hora 104
información en línea
buscar 113
visualizar 108
INSERT, sentencia 10
NOT NULL/NOT NULL WITH
DEFAULT, valor de
columna 10
instalación
navegador Netscape 109
INTEGER, tipo de datos 5
INTERSECT, operador 49
ordenación de resultados 49
restricciones de uso referidas
a 49
tipos de datos 49
INTERSECT ALL 49

L

LEFT OUTER, unión 60
Lenguaje de Consulta Estructurada
(SQL), definición 1
LIKE, predicado 51

lista de selección 19

LOB

indicador de posición,
definición 70

serie de caracteres, definición 70

localización de objetos grandes,
definición 70

M

manuales 93, 105

mensajes de error

identificador de mensaje 18

SQLSCODE 18

SQLSTATE 18

modificar tablas mediante una
vista 15

WITH CHECK OPTION 15

N

navegador Netscape

instalación 109

nombre de correlación

normas de 40

referencia calificada de nombre
de columna 40

NOT BETWEEN, predicado 50

NOT EXISTS, predicado 51

NOT IN, predicado 50

NOT LIKE, predicado 51

notas de release 104

números, descripción 5

O

operador de comparación en una
subconsulta 52

operadores aritméticos 25

orden de operaciones 24, 29

ORDER BY, cláusula 22

operadores de conjunto 49

ORG, tabla de ejemplo 82

P

PDF 104

precisión, como atributo
numérico 5

predicado

IS NOT NULL 20

IS NULL 20

predicado de nivel externo 52

procesador de línea de mandatos 1

proceso analítico en línea 66

producto cruzado 60

programas de ejemplo

HTML 103

para varias plataformas 103

PROJECT, tabla de ejemplo 83

R

REAL, tipo de datos 5

recuperación de datos 19

referencia correlacionada,
descripción 41

registro especial 70

CURRENT DATE 70

CURRENT DEGREE 70

CURRENT FUNCTION
PATH 70

CURRENT PATH 70

CURRENT SERVER 70

CURRENT TIME 71

CURRENT TIMESTAMP 71

CURRENT TIMEZONE 71

USER 71

relación entre tablas y vistas 13

restauración, asistente de 112

restricción de unicidad 54

restricciones

para operadores de conjunto 49

restricciones de unicidad 9

restricciones referenciales 9

restricciones de comprobación de
tabla

comprobación diferida de
restricción 55

definición 53

descripción 55

restricciones de integridad

referencial

clave externa 54

clave principal 54

definición 53

descripción 54

restricciones de unicidad

definición 53

RIGHT OUTER, unión 60

ROLL-UP

filas de subtotal 65

ROLLUP 65

S

SALES, tabla de ejemplo 84

selección completa 35

ALL, palabra clave 52

ANY, palabra clave 52

con sentencia INSERT 10

subconsulta 10, 52

selección completa, definición 10

selecciones completas escalares

descripción 35

SELECT, sentencia 19

serie de caracteres

como tipo de datos 5

serie de caracteres (*continuación*)

LOB 69

longitud fija 5

longitud variable 5

serie gráfica

longitud fija 5

longitud variable 5

SET, cláusula

con sentencia UPDATE 12

SET CONSTRAINTS, sentencia 55

signo, como atributo numérico 5

SMALLINT, tipo de datos 5

SmartGuides

asistentes 111

SOME, palabra clave 52

SQL, lenguaje de procedimientos
de v

SQL interactivo, definición 1

STAFF, tabla de ejemplo 85

STAFFG, tabla de ejemplo 86

subconsulta

definición 27

subconsulta correlacionada

cuándo utilizarla 43

descripción 41

supresión de filas duplicadas 24

T

tabla

base de datos de ejemplo 73

clave exclusiva 54

clave externa 54

clave primaria 54

combinar datos (unión) 26

como calificador de nombre de

columna 40

definición de 3

funciones 30

restricción de unicidad 54

tabla base 3

tabla resultante 3

tabla base 3

tabla resultante 3

tablas de ejemplo 73, 93

TIME, tipo de datos 5

TIMESTAMP, tipo de datos 5

tipo de datos

diferenciado 67

tipo de datos diferenciado 67

tipos de datos

BIGINT 5

CHAR 5

DATE 5

DATETIME 5

DECIMAL 5

DOUBLE 5

tipos de datos (*continuación*)

- FLOAT 5
- INTEGER 5
- REAL 5
- SMALLINT 5
- TIME 5
- TIMESTAMP 5
- VARCHAR 5

WHERE, cláusula 20 (*continuación*)

- consideraciones sobre la agrupación 31

WITH, cláusula 39

WITH CHECK OPTION 15

U

unión

- condiciones de unión 60
- conversión de datos 62
- definición 26
- producto cruzado 60
- sin condiciones de unión 60
- subconsultas correlacionadas 44

UNION, operador 47, 48

- descripción 47
- ordenación de resultados 48
- restricciones de uso referidas a 49
- tipos de datos 49

UNION ALL 47

unión externa

- descripción 60
- FULL OUTER, unión 60
- LEFT OUTER, unión 60
- RIGHT OUTER, unión 60

unión interna 60

UPDATE, sentencia 12

USER, registro especial 71

V

valor

- definición de 3

valor en SQL 5

valor nulo 47

- suprimir valor de columna 12

valor nulo, descripción 5

VALUES, cláusula

- con sentencia INSERT 10

VARCHAR, tipo de datos 5

vista

- como calificador de nombre de columna 40

Vista

- descripción 4
- ventajas 4

visualizar

- información en línea 108

W

WHERE, cláusula 20

- combinar datos de tablas (unión) en sentencia SELECT 26

Cómo ponerse en contacto con IBM

Si tiene un problema técnico, repase y lleve a cabo las acciones que se sugieren en la *Guía de resolución de problemas* antes de ponerse en contacto con el Centro de Asistencia al Cliente de DB2. Dicha guía sugiere información que puede reunir para ayudar al Centro de Asistencia a proporcionarle un mejor servicio.

Para obtener información o para solicitar cualquiera de los productos de DB2 Universal Database, consulte a un representante de IBM de una sucursal local o a un concesionario autorizado de IBM.

Si reside en los Estados Unidos, puede llamar a uno de los números siguientes:

- 1-800-237-5511 para obtener soporte técnico
- 1-888-426-4343 para obtener información sobre las opciones de servicio técnico disponibles

Información del producto

Si reside en los Estados Unidos, puede llamar a uno de los números siguientes:

- 1-800-IBM-CALL (1-800-426-2255) o 1-800-3IBM-OS2 (1-800-342-6672) para solicitar productos u obtener información general.
- 1-800-879-2755 para solicitar publicaciones.

<http://www.ibm.com/software/data/>

Las páginas Web de DB2 ofrecen información actual sobre DB2 referente a novedades, descripciones de productos, planes de formación, etc.

<http://www.ibm.com/software/data/db2/library/>

La biblioteca técnica de servicio y de productos DB2 ofrece acceso a preguntas frecuentemente formuladas (FAQ), arreglos de programa, manuales e información técnica actualizada sobre DB2.

Nota: Puede que esta información sólo esté disponible en inglés.

<http://www.elink.ibm.com/pbl/pbl/>

El sitio Web para el pedido de publicaciones internacionales proporciona información sobre cómo hacer pedidos de manuales.

<http://www.ibm.com/education/certify/>

El Programa de homologación profesional contenido en el sitio Web

de IBM proporciona información de prueba de homologación para diversos productos de IBM, incluido DB2.

ftp.software.ibm.com

Conéctese como anónimo (anonymous). En el directorio /ps/products/db2 encontrará programas de demostración, arreglos de programa, información y herramientas referentes a DB2 y a muchos otros productos.

comp.databases.ibm-db2, bit.listserv.db2-1

En estos foros de discusión de Internet los usuarios pueden explicar sus experiencias con los productos DB2.

En Compuserve: GO IBMDB2

Entre este mandato para acceder a los foros referentes a la familia de productos DB2. Todos los productos DB2 tienen soporte a través de estos foros.

Para conocer cómo ponerse en contacto con IBM desde fuera de los Estados Unidos, consulte el Apéndice A del manual **IBM Software Support Handbook**. Para acceder a este documento, vaya a la página Web siguiente: <http://www.ibm.com/support/> y luego seleccione el enlace "IBM Software Support Handbook", cerca del final de la página.

Nota: En algunos países, los distribuidores autorizados de IBM deben ponerse en contacto con su organización de soporte en lugar de acudir al Centro de Asistencia de IBM.



Número Pieza: CT7YHES

Impreso en España

GC10-3496-00



CT7YHES

