

IBM® DB2® Life Sciences Data Connect



Руководство по планированию, установке и настройке

Версия 8

IBM® DB2® Life Sciences Data Connect



Руководство по планированию, установке и настройке

Версия 8

Перед тем как использовать данный документ и продукт, описанный в нем, прочтите общие сведения под заголовком *Замечания*.

Этот документ содержит информацию, которая является собственностью IBM. Она предоставляется в соответствии с лицензионным соглашением и защищена законами об авторском праве. Информация в данной публикации не включает никаких гарантий на продукт и никакое из утверждений в данном руководстве не следует понимать подобным образом.

Заказать публикации IBM можно через Интернет или у местного представителя IBM.

- Чтобы заказать публикации через Интернет, перейдите на Web-страницу Центра публикаций IBM (IBM Publications Center): www.ibm.com/shop/publications/order
- Чтобы найти местное представительство IBM, перейдите на страницу IBM Directory of Worldwide Contacts по адресу www.ibm.com/planetwide

Чтобы заказать публикации DB2 через отдел DB2 Marketing and Sales в Соединенных Штатах или Канаде, позвоните по телефону 1-800-IBM-4YOU (426-4968).

Отсылая информацию IBM, вы тем самым даете IBM неисключительное право использовать или распространять эту информацию любым способом, как фирма сочтет нужным, без каких-либо обязательств перед вами.

© Copyright International Business Machines Corporation 2001, 2002. Все права защищены.

Содержание

Об этой книге	v
Для кого предназначена эта книга	v
Что нового в Версии 8?	v
Электронная информация.	vi
Соглашения	vi
Как читать синтаксические диаграммы	vii
Как отослать ваши комментарии	ix

Глава 1. Что такое DB2 Life Sciences Data Connect?	1
DB2 Life Sciences Data Connect	1
IBM Life Sciences DiscoveryLink	2

Глава 2. Установка DB2 Life Sciences Data Connect	5
Установка DB2 Life Sciences Data Connect	5
Перед установкой DB2 Life Sciences Data Connect	7
Установка DB2 Life Sciences Data Connect на серверах AIX, HP-UX, Linux и операционной среды Solaris	7
Установка DB2 Life Sciences Data Connect на серверах Windows.	8
После установки DB2 Life Sciences Data Connect	10

Глава 3. Файлы с табличной структурой как источники данных	13
Что такое файлы с табличной структурой?	13
Типы файлов с табличной структурой.	14
Сортированные файлы	14
Несортированные файлы	14
Как DB2 Life Sciences Data Connect работает с файлами с табличной структурой	14
Добавление файлов с табличной структурой в систему объединения	16
Регистрация оболочки файлов с табличной структурой	16
Задание переменной среды DB2_DJ_COMM для оболочки файлов с табличной структурой	17
Регистрация сервера для файлов с табличной структурой	18
Регистрация псевдонимов для файлов с табличной структурой.	19
Ограничения и особенности оболочки для файлов с табличной структурой.	23

Ограничения и особенности файлов для оболочки для файлов с табличной структурой	24
Модель управления доступом к файлам для оболочки для файлов с табличной структурой	25
Советы по оптимизации для оболочки для файлов с табличной структурой.	25
Сообщения для оболочки для файлов с табличной структурой.	26

Глава 4. Documentum как источник данных	31
Что такое Documentum?	31
Добавление Documentum в систему объединения	33
Компоновка с библиотеками клиента Documentum (только для AIX и операционной среды Solaris).	34
Указание файла dmcl.ini клиента Documentum	35
Регистрация оболочки Documentum	36
Задание переменной среды DB2_DJ_COMM для оболочки Documentum.	37
Регистрация сервера для источников данных Documentum	38
Аргументы	39
Опции	39
Отображение пользователей (оболочка Documentum).	40
Регистрация псевдонимов для источников данных Documentum	41
Опции столбцов	42
Опции столбца псевдонима	42
Опции псевдонима	43
Понятие псевдостолбцов	44
Пример CREATE NICKNAME	48
Регистрация пользовательских функций для источников данных Documentum	50
Правила для строк аргументов пользовательской функции	51
Использование пользовательских функций в запросах	51
Таблица пользовательских функций	52
Выполнение запросов для источников данных Documentum	57
Что такое утилита CreateNicknameFile для оболочки Documentum?	58

Установка утилиты CreateNicknameFile (оболочка Documentum)	58	Аргументы	103
Конфигурирование утилиты CreateNicknameFile (оболочка Documentum)	59	Опции.	103
Отображение типа объектов DM_ID в зарегистрированные таблицы Documentum	60	Регистрация псевдонимов для источников данных BLAST	104
Двойное определение для атрибутов с возможностью повторения (оболочка Documentum)	61	Опции столбца псевдонима	105
Ограничения и особенности оболочки Documentum	62	Опции псевдонима	106
Управление доступом для оболочки Documentum	64	Анализ строки определения.	107
Сообщения оболочки Documentum	64	Фиксированные столбцы	107
Глава 5. Excel как источник данных	73	Пример CREATE NICKNAME	111
Что такое Excel?	73	Построение запросов SQL BLAST	112
Предварительное требование для оболочки Excel	75	Примеры запросов BLAST	113
Добавление Excel в систему объединения.	75	Советы по оптимизации оболочки BLAST	115
Регистрация оболочки Excel	75	Сообщения оболочки BLAST	115
Регистрация сервера для источника данных Excel	76	Глава 7. XML как источник данных	119
Определения аргументов	76	Что такое XML?	119
Регистрация псевдонимов для источников данных Excel	77	Добавление XML в систему объединения	123
Синтаксис оператора CREATE NICKNAME (для Excel)	77	Регистрация оболочки XML	124
Определения опций	78	Задание переменной среды DB2_DJ_COMM для оболочки XML	124
Выполнение запросов для источников данных Excel	79	Регистрация сервера для источника данных XML	125
Пример сценария оболочки Excel	79	Регистрация псевдонимов для источников данных XML	126
Ограничения оболочки Excel.	82	Создание производных таблиц объединения для некорневых псевдонимов (оболочка XML)	131
Ограничения файлов Excel	82	Запуск запросов к источникам данных XML	133
Модель управления доступом к файлам для оболочки Excel	82	Ограничения и особенности оболочки XML	134
Сообщения оболочки Excel	83	Сообщения оболочки XML	135
Глава 6. BLAST как источник данных	91	Глава 8. Задание опций стоимости псевдонима	143
Что такое BLAST?	91	Глава 9. Изменение псевдонимов	145
Добавление BLAST в систему объединения	96	Изменение псевдонимов	145
Проверка правильности установленной версии выполняемого файла blastall и файлов матриц.	97	Изменение типа данных	145
Конфигурирование демона BLAST.	97	Изменение опции псевдонима	146
Запуск демона BLAST	100	Замечания	147
Регистрация оболочки BLAST	101	Товарные знаки	150
Задание переменной среды DB2_DJ_COMM для оболочки BLAST.	102	Библиография	153
Регистрация сервера для источника данных BLAST	103	Индекс	155
		Как связаться с IBM	157
		Информация о продукте.	157

Об этой книге

Эта книга содержит:

- Введение в DB2 Life Sciences Data Connect и описание его связей с предложением IBM Life Sciences DiscoveryLink, содержательным набором программ и служб для задач биологии, социологии и медицины
- Инструкции по установке DB2 Life Sciences Data Connect
- Инструкции по добавлению источников данных к системе объединения и регистрации оболочек. Оболочки - это модули, которые позволяют вам или прикладным программам связываться с источниками данных при помощи операторов SQL.

Технические изменения в тексте отмечены вертикальной чертой слева от изменения.

Для кого предназначена эта книга

Эта книга предназначена для администраторов, которые конфигурируют среду базы данных объединения для данных исследований по биологии, социологии и медицине, и для прикладных программистов, разрабатывающих программы для такой среды.

Что нового в Версии 8?

В число новых возможностей DB2 Life Sciences Data Connect Версии 8 входит:

Общие возможности

- Изменены имена библиотек оболочек.
- Добавлены опции стоимости псевдонима для нереляционных источников данных.

Улучшено планирование запросов для нереляционных источников данных

Оболочки DB2 Life Sciences Data Connect переписаны так, чтобы поддерживать глобальный процесс планирования запроса, улучшающий стратегию доступа для запроса к одному из поддерживаемых источников данных. Эта новая функция планирования повышает производительность запросов, направляемых оболочкам нереляционных источников данных.

Оболочка XML

Добавлена оболочка XML. Она обеспечивает доступ систем объединения к источникам данных XML. XML добавлен к списку

нереляционных оболочек, появившихся в DB2 Universal Database Версии 7 и включающих в себя BLAST, Documentum, Excel и файлы с табличной структурой.

Оболочка для файлов с табличной структурой

- Опции сервера TYPE, VERSION и NODE более не требуются.
- Добавлена опция псевдонима SORTED.

Оболочка Documentum

- Добавлена опция псевдонима ALL_VALUES.
- Следующие пользовательские функции Версии 7 теперь рассматриваются как псевдостолбцы:
 - GET_FILE
 - GET_FILE_DEL
 - GET_RENDITION
 - GET_RENDITION_DEL
 - HITS
 - SCORE
- Добавлена пользовательская функция RENDITION_FORMAT.

Оболочка Excel

- Для источников данных Excel97 и Excel2000 теперь используется единая оболочка.
- Опции сервера TYPE, VERSION и NODE более не требуются.

Электронная информация

С этом разделе приводятся адреса Web и электронной почты, связанные с продуктом.

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

Сайт продукта DB2 Life Sciences Data Connect

<http://www.ibm.com/solutions/lifesciences/discoverylink.html>

Сайт DiscoveryLink

<http://www.ibm.com/solutions/lifesciences/>

Сайт IBM Life Sciences

ls@us.ibm.com

Адрес электронной почты IBM Life Sciences

Соглашения

В этой книге используются следующие выделения:

Жирный шрифт

Выделяет команды и элементы графического интерфейса пользователя, такие как имена полей, папок и пункты меню.

Моноширинный шрифт

Примеры кода или текст, который должны ввести вы.

Курсив Выделяет переменные, вместо которых надо подставить ваши значения.

Курсив используется также для названий книг и выделенных слов.

ВЕРХНИЙ РЕГИСТР

Ключевые слова SQL и имена объектов (например, таблиц, производных таблиц и серверов).

Как читать синтаксические диаграммы

В этой книге синтаксис описывается по следующей схеме:

Читайте синтаксические диаграммы слева направо и сверху вниз по линиям диаграмм.

Символ \blacktriangleright — означает начало оператора.

Символ \longrightarrow означает, что оператор продолжается на следующей строке.

Символ \blacktriangleright — означает, что данная строка - продолжение оператора с предыдущей строки.

Символ $\longrightarrow\blacktriangleleft$ означает конец оператора.

Обязательные элементы появляются на горизонтальной линии (основном пути).

\blacktriangleright —STATEMENT—*обязательный элемент*— \longrightarrow

Необязательные элементы выводятся под главным путем.

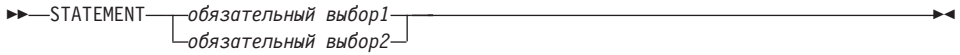
\blacktriangleright —STATEMENT— $\left[$ *необязательный элемент* $\right]$ — \longrightarrow

Необязательный элемент может выводиться над главным путем; это не влияет на выполнение оператора и применяется только для удобства чтения.

\blacktriangleright —STATEMENT— $\left[$ *необязательный элемент* $\right]$ — \longrightarrow

Если возможен выбор из двух или более элементов, они указываются один над другим.

Если один из этих элементов выбрать *необходимо*, этот вертикальный ряд будет находиться на главном пути.



Если можно не выбирать ни один из элементов, весь вертикальный ряд размещается под главным путем.



Если один из элементов выбирается по умолчанию, он будет помещен над главным путем, а остальные варианты - внизу.



Обратная стрелка над главным путем означает, что элемент можно повторить. В таком случае элементы должны отделяться друг от друга одним или несколькими пробелами.



Если стрелка повтора содержит запятую, необходимо разделять повторяемые элементы запятой.



Стрелка повтора над вертикальным рядом означает, что можно выбрать несколько элементов из ряда или повторить один элемент несколько раз.

Ключевые слова показаны в верхнем регистре (например, FROM). Их надо писать в точности так, как на диаграмме. Переменные записываются в нижнем регистре (например, имя-столбца). Они представляют имена или значения в синтаксисе, задаваемые пользователем.

Если показаны знаки препинания, круглые скобки, арифметические операторы или другие подобные символы, необходимо ввести их как часть синтаксиса.

Иногда одна переменная представляет набор параметров. Например, на следующей диаграмме переменную блок-параметров можно заменить на любую реализацию диаграммы под заголовком **блок-параметров**:



блок-параметров:



Соседние сегменты между значками (●) можно задавать в любом порядке.



В этой диаграмме показано, что элемент2 и элемент3 могут идти в любом порядке. Оба следующих варианта допустимы:

STATEMENT элемент1 элемент2 элемент3 элемент4
 STATEMENT элемент1 элемент3 элемент2 элемент4

Как отослать ваши комментарии

Ваши отзывы помогут ИВМ повысить качество информации. Шлите нам ваши комментарии по поводу этой книги и прочих публикаций по DB2. Можно использовать любой из следующих методов отправки комментариев:

- Послать ваши комментарии из Web. Форму для отправки комментариев для читателей IBM Data Management смотрите по адресу <http://www.ibm.com/software/data/rcf>
- Посылать отзывы электронной почтой по адресу comments@vnet.ibm.com. Не забудьте указать название продукта, номер версии продукта, название и номер книги (если есть). Если вы высылаете комментарий к определенному тексту, укажите положение этого текста (например, главу и название раздела, номер таблицы, номер страницы или заголовок темы справки.)

Глава 1. Что такое DB2 Life Sciences Data Connect?

В этой главе вы познакомитесь с предлагаемым продуктом DB2 Life Sciences Data Connect; здесь описываются также основные шаги настройки системы для запросов данных по биологии, медицине и социологии.

DB2 Life Sciences Data Connect

IBM® DB2® Life Sciences Data Connect позволяет интегрировать в системе объединения DB2 информацию по химии, биологии, генетике и другие научные данные из распределенных источников. Система объединения DB2 - это распределенная вычислительная система, состоящая из сервера DB2 Universal Database™ и нескольких источников данных, из которых сервер DB2 Universal Database получает данные.

Система объединения позволяет вам или прикладной программе с помощью операторов SQL запрашивать, получать или объединять данные, которые могут находиться в нескольких разнородных источниках, например, реляционных базах данных IBM, Oracle, Sybase и Microsoft или нереляционных источниках данных, таких как файлы с табличной структурой. На рис. 1 показано, как система объединения обращается с помощью DB2 Life Sciences Data Connect к нескольким источникам научных данных.

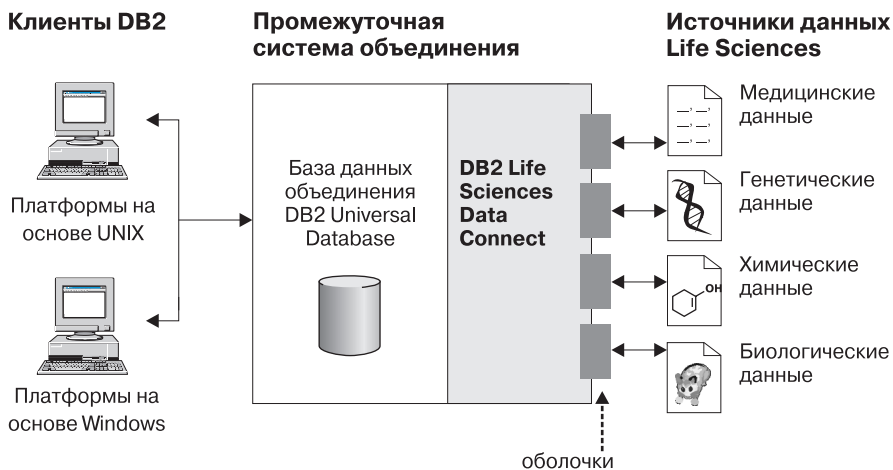


Рисунок 1. Доступ к данным биологии, социологии и медицины с помощью DB2 Life Sciences Data Connect

В систему объединения DB2 входят клиенты, база данных (называемая базой данных объединения), куда клиенты посылают запросы, интерфейс, через который база данных объединения взаимодействует с источниками данных, и сами источники данных.

Механизм, посредством которого сервер объединения взаимодействует с источником данных, называется *оболочкой*. Для реализации оболочки сервер использует подпрограммы из библиотеки, называемой *модулем оболочки*. Эти подпрограммы позволяют серверу выполнять такие операции, как соединение с источником данных и получение данных от него.

Когда система объединения сконфигурирована, к информации источников данных можно обращаться так, как если бы она находилась в одной большой базе данных. Пользователи и прикладные программы посылают запросы в одну базу данных объединения, получающую информацию от многих источников данных. Прикладные программы работают с базой данных объединения, как и с любой другой базой данных DB2.

Дополнительную информацию о системах объединения смотрите в руководстве *DB2 SQL Reference*.

Понятия, связанные с данным:

- “IBM Life Sciences DiscoveryLink” на стр. 2

IBM Life Sciences DiscoveryLink

Предлагаемый продукт DiscoveryLink - это набор промежуточного программного обеспечения и служб, специально приспособленный для интеграции данных из нескольких разнородных источников в интересах развития исследований в области биологии и смежных наук.

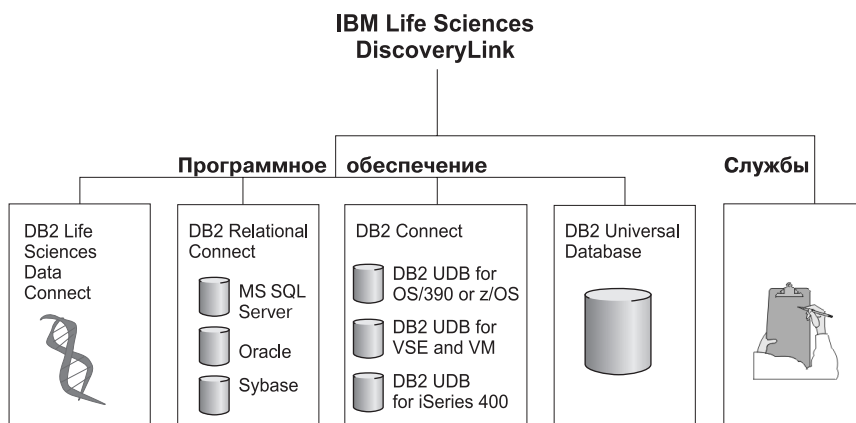


Рисунок 2. IBM Life Sciences DiscoveryLink

Например, DiscoveryLink позволяет с помощью единственного оператора SQL интегрировать данные об аминокислотных последовательностях белков из базы данных Oracle в Швейцарии, информацию о химической структуре соединений из базы данных Sybase в Японии и спектроскопические данные из плоских файлов с табличной структурой, хранящихся в вашей локальной сети. Данные при этом выглядят как данные одной виртуальной базы данных.

В число программных компонентов входит:

DB2[®] Life Sciences Data Connect

Для доступа к данным по биологии, социологии и медицине.

DB2 Relational Connect

Для доступа к реляционным базам данных Oracle, Sybase и Microsoft[®].
Дополнительную информацию о DB2 Relational Connect смотрите в руководстве *Federated Systems Guide*.

DB2 Connect[™]

Для доступа к серверам баз данных DB2 на системах хоста.
Дополнительную информацию о DB2 Connect смотрите в руководстве *DB2 Connect User's Guide*.

DB2 Universal Database[™]

Для оптимизации запросов и интеграции результатов, полученных из нескольких разнородных источников данных. Дополнительную информацию о DB2 Universal Database смотрите в руководстве *DB2 Administration Guide*.

Дополнительные сведения о программах и службах DiscoveryLink смотрите ниже в главе "Электронная информация" в разделе ссылок.

Понятия, связанные с данным:

- “DB2 Life Sciences Data Connect” на стр. 1

Глава 2. Установка DB2 Life Sciences Data Connect

В этой главе приводятся подробности о платформах, поддерживаемых для каждой оболочки, инструкции по установке для оболочек систем на основе Unix и Windows и подробности о библиотеках оболочек, помещаемых в вашу систему после завершения установки.

Установка DB2 Life Sciences Data Connect

Чтобы запрашивать и получать научные данные при помощи DB2 Life Sciences Data Connect, надо установить оболочки и зарегистрировать каждую оболочку для их добавления в систему объединения.

Оболочки не предъявляют дополнительных требований сверх требований DB2 Universal Database, и работают при любых конфигурациях системы, поддерживаемых DB2 Universal Database.

В Табл. 1 приведены оболочки DB2 Life Sciences Data Connect для каждой операционной системы. Указания по регистрации каждой научной оболочки приведены в темах, перечисленных ниже в соответствующем разделе ссылок.

Таблица 1. Оболочки DB2 Life Sciences Data Connect для различных операционных систем

Оболочка	Windows	AIX	HP-UX	Linux	Операционная среда Solaris
Файлы с табличной структурой	X	X	X	X	X
Documentum	X	X			X
Excel	X				
BLAST	X	X		X	X
XML	X	X	X	X	X

В ходе установки выберите один из трех возможных компонентов: Scientific, Structured files и Applications. Перечень возможных компонентов установки и оболочек, включенных в каждый компонент, приведен в Табл. 2 на стр. 6.

Таблица 2.

Имя компонента установки	Описание	Содержит оболочки
Scientific	Эти источники данных разработаны специально для работы с информацией социологии, биологии и медицины, такими как базы генов, белков, биологических и химических данных.	BLAST
Structured files	Эти источники данных содержат научные данные, хранимые в файлах с определенной повторяющейся структурой.	Файл с табличной структурой, Excel, XML
Applications	Эти источники данных используют прикладную программу для доступа к данным по социологии, биологии и медицине. Сами эти данные могут быть в различных стандартных и нестандартных форматах.	Documentum

Процедура:

Чтобы установить DB2 Life Sciences Data Connect:

1. Перед установкой DB2 Life Sciences Data Connect.
2. Установите DB2 Life Sciences Data Connect на серверах AIX, HP-UX, Linux и операционной среды Solaris.
3. Установите DB2 Life Sciences Data Connect на серверах Windows
4. После установки DB2 Life Sciences Data Connect.

Задачи, связанные с данной темой:

- “Перед установкой DB2 Life Sciences Data Connect” на стр. 7
- “Установка DB2 Life Sciences Data Connect на серверах AIX, HP-UX, Linux и операционной среды Solaris” на стр. 7
- “Установка DB2 Life Sciences Data Connect на серверах Windows” на стр. 8
- “После установки DB2 Life Sciences Data Connect” на стр. 10
- “Установка утилиты CreateNicknameFile (оболочка Documentum)” на стр. 58

Перед установкой DB2 Life Sciences Data Connect

Эта задача - часть большей задачи *Установка DB2 Life Sciences Data Connect*.

Процедура:

Перед установкой DB2 Life Sciences Data Connect на сервере объединения:

- Убедитесь, что на вашем сервере объединения установлена DB2 Universal Database Enterprise Server Edition.
- Убедитесь, что в базе данных включена поддержка систем баз данных объединения. Чтобы проверить этот параметр, выполните команду из командной строки DB2:

```
GET DATABASE MANAGER CONFIGURATION
```

В окне команд появится список всех параметров базы данных и их текущих значений. Убедитесь, что для параметра FEDERATED задано значение YES.

Если параметр FEDERATED равен NO, введите в командной строке DB2:

```
UPDATE DATABASE MANAGER CONFIGURATION USING FEDERATED YES
```

Следующая задача в этой последовательности задач - *Установка DB2 Life Sciences Data Connect на серверах AIX, HP-UX, Linux и операционной среды Solaris*.

Задачи, связанные с данной темой:

- “Установка DB2 Life Sciences Data Connect на серверах Windows” на стр. 8
- “После установки DB2 Life Sciences Data Connect” на стр. 10
- “Установка DB2 Life Sciences Data Connect на серверах AIX, HP-UX, Linux и операционной среды Solaris” на стр. 7

Установка DB2 Life Sciences Data Connect на серверах AIX, HP-UX, Linux и операционной среды Solaris

Эта задача - часть большей задачи *Установка DB2 Life Sciences Data Connect*.

Предварительные требования:

Посмотрите в разделе Задачи, связанные с данной, “Перед установкой DB2 Life Sciences Data Connect”.

Порядок действий:

Чтобы установить DB2 Life Sciences Data Connect на серверах объединения AIX, HP-UX, Linux и операционной среды Solaris, воспользуйтесь утилитой db2setup.

Примечание: Содержимое экранов, появляющихся во время работы утилиты db2setup, зависит от того, какие программные продукты уже

установлены на сервере объединения. В следующих шагах предполагается, что DB2 Life Sciences Data Connect не установлена.

1. Зарегистрируйтесь как пользователь с полномочиями root.
2. Вставьте и смонтируйте компакт-диск с DB2 Life Sciences Data Connect. Информацию о монтировании компакт-диска смотрите в руководстве *DB2 for UNIX Quick Beginnings*.
3. Перейдите в каталог, где смонтирован компакт-диск, введя команду **cd /cdrom**, где *cdrom* - точка монтирования компакт-диска с продуктом.
4. Введите команду:

```
./db2setup
```

Откроется окно Установка DB2.
5. Следуйте подсказкам программы установки.
Когда установка завершится, DB2 Life Sciences Data Connect будет установлена в каталоге вместе с другими продуктами DB2.
 - На серверах DB2 for AIX этот каталог называется /usr/opt/db2_08_01.
 - На серверах DB2 для операционной среды Solaris этот каталог называется /opt/IBM/db2/V8.1
 - На серверах DB2 for HP-UX этот каталог называется /opt/IBM/db2/V8.1
 - На серверах DB2 for Linux этот каталог называется /opt/IBM/db2/V8.1

Следующая задача в этой последовательности задач - *Установка DB2 Life Sciences Data Connect на серверах Windows*.

Задачи, связанные с данной темой:

- “Перед установкой DB2 Life Sciences Data Connect” на стр. 7
- “Установка DB2 Life Sciences Data Connect на серверах Windows” на стр. 8
- “После установки DB2 Life Sciences Data Connect” на стр. 10

Установка DB2 Life Sciences Data Connect на серверах Windows

Эта задача - часть большей задачи *Установка DB2 Life Sciences Data Connect*.

Предварительные требования:

Посмотрите в разделе Задачи, связанные с данной, “Перед установкой DB2 Life Sciences Data Connect”.

Процедура:

Чтобы установить DB2 Life Sciences Data Connect на серверах объединения Windows, используйте программу установки.

1. Зарегистрируйтесь на сервере объединения с учетной записью пользователя, которую вы создали для установки DB2 Universal Database.
2. Завершите все запущенные программы, чтобы программа настройки могла изменить файлы, как это требуется.
3. Вызовите программу установки. Программу установки можно вызвать автоматически или вручную. Если программа установки не запускается автоматически, или если вы хотите запустить ее на другом языке, вызовите ее вручную.
 - Чтобы вызвать программу установки автоматически, вставьте в дисковод компакт-диск DB2 Life Sciences Data Connect. Функция автозапуска автоматически запустит программу установки. Будет определен язык системы и загрузится программа установки для этого языка.
 - Чтобы вызвать программу установки вручную:
 - a. Нажмите кнопку **Пуск** и выберите **Выполнить**.
 - b. В окне **Открыть** введите команду:
`x:\setup /i язык`

где:

x: - Буква дисковода компакт-дисков.

язык - код вашего языка (например, RU для русского).
 - c. Нажмите кнопку **ОК**.

Откроется панель запуска установки.

4. Чтобы начать установку, нажмите кнопку **Установить**.
5. Следуйте подсказкам программы установки.
Когда установка завершится, DB2 Life Sciences Data Connect будет установлена в каталоге установки вместе с другими продуктами DB2.
Каталог установки по умолчанию - C:\Program Files\IBM\SQLLIB.

Следующая задача в этой последовательности задач - *После установки DB2 Life Sciences Data Connect*.

Задачи, связанные с данной темой:

- “Перед установкой DB2 Life Sciences Data Connect” на стр. 7
- “После установки DB2 Life Sciences Data Connect” на стр. 10
- “Установка DB2 Life Sciences Data Connect на серверах AIX, HP-UX, Linux и операционной среды Solaris” на стр. 7

После установки DB2 Life Sciences Data Connect

Эта задача - часть большей задачи *Установка DB2 Life Sciences Data Connect*. После установки в систему записываются файлы библиотек оболочек. Эти библиотеки используются в процессе регистрации оболочки.

Порядок действий:

Чтобы проверить правильность установки, убедитесь, что в каталогах установки записаны библиотеки оболочек по умолчанию.

Имена файлов всех библиотек по умолчанию для платформ Windows приведены в Табл. 3, а для платформ UNIX - в Табл. 4.

Таблица 3. Имена библиотек оболочек по умолчанию на платформах Windows

Оболочка	Windows
Файлы с табличной структурой	db2lsfile.dll
Documentum	db2lsdctm.dll
Excel97 / Excel2000	db2lsxls.dll
BLAST	db2lsblast.dll
XML	db2lsxml.dll

В Табл. 4 приведены имена библиотек оболочек на поддерживаемых платформах UNIX.

Таблица 4. Имена библиотек оболочек по умолчанию на платформах UNIX

Оболочка	AIX	HP-UX	Linux	Solaris Operating Environment
Файлы с табличной структурой	libdb2lsfile.a	libdb2lsfile.sl	libdb2lsfile.so	libdb2lsfile.so
Documentum	libdb2lsdctm.a			libdb2lsdctm.so
BLAST	libdb2lsblast.a		libdb2lsblast.so	libdb2lsblast.so
XML	libdb2lsxml.a	libdb2lsxml.sl	libdb2lsxml.so	libdb2lsxml.so

Примечание: Для Documentum на всех платформах, кроме Windows, эти библиотеки создаются после того, как они скомпонованы с библиотеками клиента Documentum. После установки в вашей системе появится файл с именем `libdb2lsdctmF.a` в AIX или `libdb2lsdctmF.so` в операционной среде Solaris.

Это последняя задача данной последовательности.

Задачи, связанные с данной темой:

- “Перед установкой DB2 Life Sciences Data Connect” на стр. 7
- “Установка DB2 Life Sciences Data Connect на серверах Windows” на стр. 8
- “Добавление файлов с табличной структурой в систему объединения” на стр. 16
- “Установка DB2 Life Sciences Data Connect на серверах AIX, HP-UX, Linux и операционной среды Solaris” на стр. 7
- “Добавление Documentum в систему объединения” на стр. 33
- “Добавление Excel в систему объединения” на стр. 75
- “Добавление BLAST в систему объединения” на стр. 96
- “Добавление XML в систему объединения” на стр. 123

Глава 3. Файлы с табличной структурой как источники данных

В этой главе объясняется, что такое файлы с табличной структурой и как добавить их в качестве источников данных в вашу систему объединения, и приводятся сообщения об ошибках, связанные с оболочкой для файлов с табличной структурой.

Что такое файлы с табличной структурой?

Файл с табличной структурой состоит из ряда записей с одинаковым числом полей, разделенных какими-либо разделителями. Пустые значения представляются двумя разделителями подряд.

Ниже приводится пример содержимого файла DRUGDATA1.TXT. Этот файл содержит три записи, каждая из трех полей, разделенных запятыми:

```
234, ЛекарствоА, Производитель1  
332, ЛекарствоВ, Производитель2  
333, ЛекарствоС, Производитель2
```

Первое поле содержит уникальный номер - ID лекарства. Второе поле содержит название этого лекарства. Третье поле содержит название производителя этого лекарства.

Понятия, связанные с данным:

- “Типы файлов с табличной структурой” на стр. 14
- “Как DB2 Life Sciences Data Connect работает с файлами с табличной структурой” на стр. 14
- “Что такое Documentum?” на стр. 31
- “Что такое Excel?” на стр. 73
- “Что такое BLAST?” на стр. 91
- “Что такое XML?” на стр. 119

Задачи, связанные с данной темой:

- “Добавление файлов с табличной структурой в систему объединения” на стр. 16

Типы файлов с табличной структурой

Файлы с табличной структурой могут быть сортированными или несортированными.

Сортированные файлы

Файл DRUGDATA1.TXT содержит сортированные записи. Он отсортирован по первому полю с уникальными номерами ID лекарств. Это поле представляет собой первичный ключ, так как оно уникально для каждого лекарства.

Сортированные файлы должны быть отсортированы в восходящем порядке.

234, ЛекарствоА, Производитель1

332, ЛекарствоВ, Производитель2

333, ЛекарствоС, Производитель2

Несортированные файлы

Файл DRUGDATA2.TXT содержит несортированные записи. Эти записи располагаются в файле в произвольном порядке.

332, ЛекарствоВ, Производитель2

234, ЛекарствоА, Производитель1

333, ЛекарствоС, Производитель2

Оболочка может вести поиск в файлах с сортированными данными гораздо эффективнее, чем в файлах с несортированными данными.

Понятия, связанные с данным:

- “Что такое файлы с табличной структурой?” на стр. 13
- “Как DB2 Life Sciences Data Connect работает с файлами с табличной структурой” на стр. 14

Задачи, связанные с данной темой:

- “Добавление файлов с табличной структурой в систему объединения” на стр. 16

Как DB2 Life Sciences Data Connect работает с файлами с табличной структурой

С помощью модуля, называемого оболочкой, DB2 Life Sciences Data Connect может обрабатывать операторы SQL, запрашивающие данные в файлах с табличной структурой, как если бы данные содержались в обычной реляционной таблице или производной таблице. Это позволяет объединять данные в файлах с табличной структурой с реляционными данными или с данными других таких же файлов. Процесс такого объединения изображен на рис. 3 на стр. 15.

Клиент DB2

База данных объединения

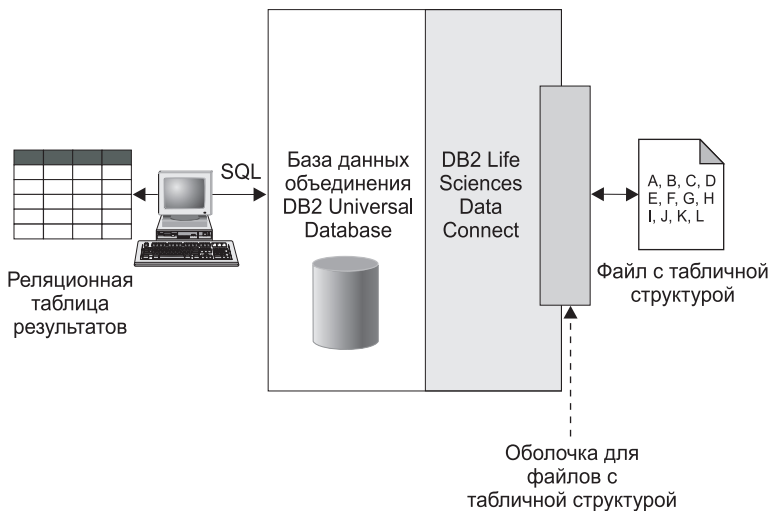


Рисунок 3. Как работает оболочка для файлов с табличной структурой

Например, допустим, что файл с табличной структурой DRUGDATA2.TXT находится на компьютере в вашей лаборатории. Запрашивать эти данные, сопоставляя их затем с табличными данными из других используемых вами источников, может быть утомительно.

После регистрации DRUGDATA2.TXT в DB2 Life Sciences Data Connect с этим файлом можно работать как с реляционным источником данных. К нему теперь можно посылать запросы наряду с другими реляционными и нереляционными источниками данных и совместно анализировать результаты этих запросов.

Например, можно запустить такой запрос:

```
SELECT * FROM DRUGDATA2 ORDER BY DCODE
```

По этому запросу будут найдены следующие результаты.

Dcode	Лекарство	Производитель
234	ЛекарствоА	Производитель1
332	ЛекарствоВ	Производитель2
333	ЛекарствоС	Производитель2

Понятия, связанные с данным:

- “Что такое файлы с табличной структурой?” на стр. 13

- “Типы файлов с табличной структурой” на стр. 14

Задачи, связанные с данной темой:

- “Добавление файлов с табличной структурой в систему объединения” на стр. 16

Добавление файлов с табличной структурой в систему объединения

Порядок действий:

Чтобы добавить источник данных в виде файла с табличной структурой на сервер объединения:

1. Зарегистрируйте оболочку при помощи команды CREATE WRAPPER.
2. Необязательно: для повышения производительности запросов задайте переменную среды DB2_DJ_COMM.
3. Зарегистрируйте сервер при помощи команды CREATE SERVER.
4. Зарегистрируйте псевдонимы при помощи команды CREATE NICKNAME для всех файлов с табличной структурой.

Эти команды можно запустить из командной строки DB2.

Задачи, связанные с данной темой:

- “Регистрация оболочки файлов с табличной структурой” на стр. 16
- “Задание переменной среды DB2_DJ_COMM для оболочки файлов с табличной структурой” на стр. 17
- “Регистрация сервера для файлов с табличной структурой” на стр. 18
- “Регистрация псевдонимов для файлов с табличной структурой” на стр. 19
- “Добавление Documentum в систему объединения” на стр. 33
- “Добавление Excel в систему объединения” на стр. 75
- “Добавление BLAST в систему объединения” на стр. 96
- “Добавление XML в систему объединения” на стр. 123

Регистрация оболочки файлов с табличной структурой

Эта задача - частью большей задачи *Добавление файлов с табличной структурой в систему объединения*. Чтобы можно было обращаться к источникам данных, необходимо зарегистрировать оболочку. Оболочка - это механизм, посредством которого серверы объединения подключаются к источникам данных и получают от них данные. Оболочки устанавливаются в системе в виде файлов библиотек.

Порядок действий:

Чтобы зарегистрировать оболочку, воспользуйтесь оператором CREATE WRAPPER для указания, какая оболочка будет использоваться для доступа к файлам с табличной структурой.

Например, чтобы зарегистрировать оболочку в AIX, выполните оператор:

```
CREATE WRAPPER laboratory_flat_files LIBRARY 'libdb2sfile.a'  
  OPTIONS(DB2_FENCED 'N');
```

В этом примере `laboratory_flat_files` - имя, выбранное для оболочки. Это имя должно быть уникальным в базе данных, в которой выполняется регистрация оболочки. Имя требуемой библиотеки для оболочки файлов с табличной структурой в AIX - `libdb2sfile.a`.

Имя `libdb2sfile.a` присваивается библиотеке при установке по умолчанию, если не было задано другого имени. Узнайте правильное имя у своего системного администратора.

Таблица имен файлов по умолчанию для библиотеки оболочки файлов с табличной структурой для поддерживаемых платформ приведена далее, в подразделе "После установки DB2 Life Sciences Data Connect" раздела Задачи, связанные с данной. Дополнительную информацию об операторе CREATE WRAPPER смотрите в книге *DB2 SQL Reference*.

Следующая задача в этой последовательности цепочке задач - *Задание переменной среды DB2_DJ_COMM для оболочки файлов с табличной структурой*.

Задачи, связанные с данной темой:

- "Задание переменной среды DB2_DJ_COMM для оболочки файлов с табличной структурой" на стр. 17
- "Регистрация оболочки Documentum" на стр. 36
- "Регистрация оболочки Excel" на стр. 75
- "Регистрация оболочки BLAST" на стр. 101
- "Регистрация оболочки XML" на стр. 124

Задание переменной среды DB2_DJ_COMM для оболочки файлов с табличной структурой

Эта задача - частью большей задачи *Добавление файлов с табличной структурой в систему объединения*. Чтобы повысить производительность доступа к файлам с табличной структурой, задайте переменную среды DB2_DJ_COMM. Эта переменная определяет, будет ли сервер объединения загружать оболочку при инициализации.

Порядок действий:

Чтобы задать переменную среды DB2_DJ_COMM, выполните команду db2set, указав в ней библиотеку оболочки, заданную в соответствующем операторе CREATE WRAPPER.

Например:

```
db2set DB2_DJ_COMM='libdb2lsfile.a'
```

Слева и справа от знака равенства (=) не должно быть пробелов.

При загрузке библиотек оболочек при запуске базы данных возникают дополнительные затраты времени. Чтобы избежать этих затрат, просто задайте библиотеки, к которым нужно обращаться.

Дополнительную информацию о переменной DB2_DJ_COMM смотрите в книге *DB2 Administration Guide*.

Следующая задача в этой последовательности задач - *Регистрация сервера для файлов с табличной структурой*.

Задачи, связанные с данной темой:

- “Регистрация оболочки файлов с табличной структурой” на стр. 16
- “Регистрация сервера для файлов с табличной структурой” на стр. 18
- “Задание переменной среды DB2_DJ_COMM для оболочки Documentum” на стр. 37
- “Задание переменной среды DB2_DJ_COMM для оболочки BLAST” на стр. 102
- “Задание переменной среды DB2_DJ_COMM для оболочки XML” на стр. 124

Регистрация сервера для файлов с табличной структурой

Эта задача - часть большей задачи *Добавление файлов с табличной структурой в систему объединения*. Зарегистрировав оболочку, необходимо зарегистрировать соответствующий сервер.

Порядок действий:

Для регистрации сервера файл с табличной структурой в системе объединения используйте оператор CREATE SERVER. Например:

```
CREATE SERVER biochem_lab WRAPPER laboratory_flat_files
```

В этом примере biochem_lab - имя, присвоенное серверу файлов с табличной структурой. Имя должно быть уникальным в базе данных, в которой выполняется регистрация сервера.

Дополнительную информацию об операторе CREATE SERVER смотрите в руководстве *DB2 SQL Reference*.

Следующая задача в этой последовательности задач - *Регистрация псевдонимов для файлов с табличной структурой*.

Задачи, связанные с данной темой:

- “Задание переменной среды DB2_DJ_COMM для оболочки файлов с табличной структурой” на стр. 17
- “Регистрация псевдонимов для файлов с табличной структурой” на стр. 19
- “Регистрация сервера для источников данных Documentum” на стр. 38
- “Регистрация сервера для источника данных Excel” на стр. 76
- “Регистрация сервера для источника данных BLAST” на стр. 103
- “Регистрация сервера для источника данных XML” на стр. 125

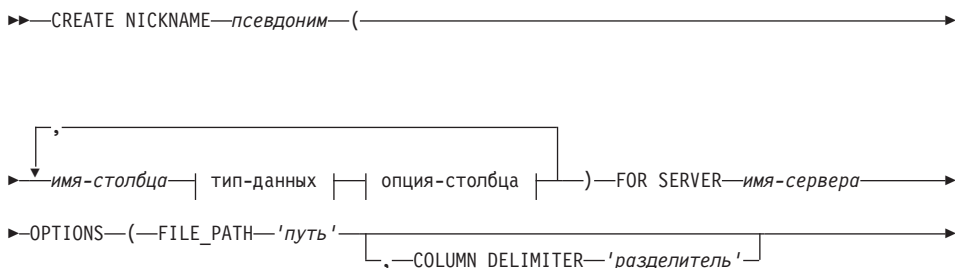
Регистрация псевдонимов для файлов с табличной структурой

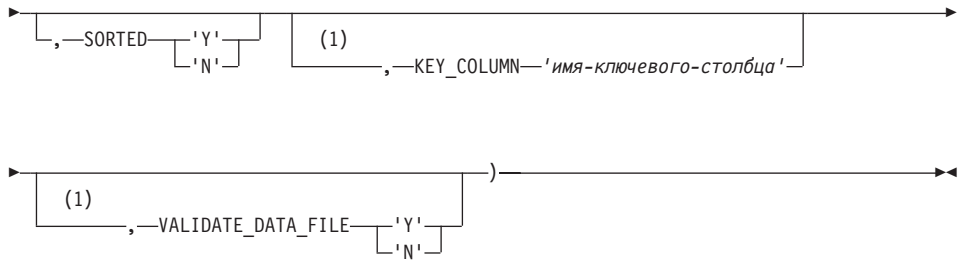
Эта задача - частью большей задачи *Добавление файлов с табличной структурой в систему объединения*. Зарегистрировав сервер, необходимо зарегистрировать соответствующий псевдоним. Псевдонимы используются для обозначения источника данных - файла с табличной структурой в запросе.

Порядок действий:

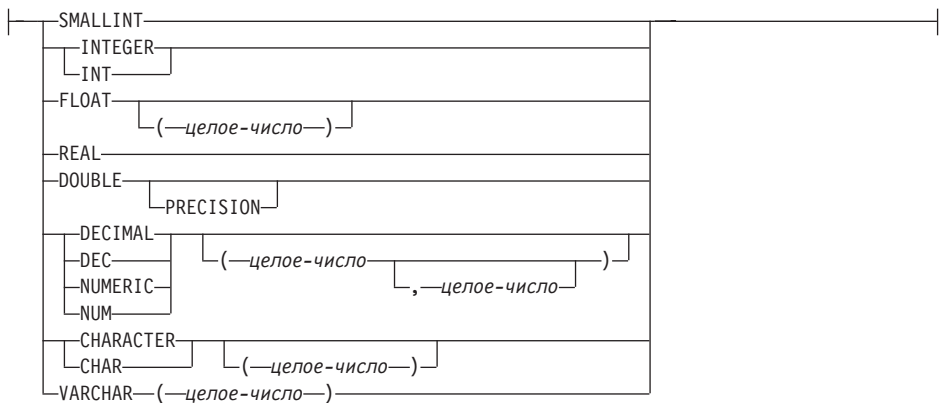
При помощи оператора CREATE NICKNAME зарегистрируйте псевдонимы для всех файлов с табличной структурой, к которым хотите обращаться.

Синтаксис оператора CREATE NICKNAME:





тип-данных:



опция-столбца:



Примечания:

- 1 Не допускается для несортированных файлов. Необязательна для сортированных файлов.

псевдоним

Уникальный псевдоним для обращения к файлу с табличной структурой. Он должен отличаться от остальных псевдонимов и имен таблиц и производных таблиц в той схеме, в которой его регистрируют.

имя-столбца

Уникальное имя, присваиваемое каждому полю в файле с табличной структурой. После каждого имени столбца указывайте тип его данных. Поддерживаются только столбцы типов CHAR, VARCHAR, SMALLINT, INTEGER, FLOAT, DOUBLE, REAL и DECIMAL.

SMALLINT

Для маленького целого числа.

INTEGER или INT

Для большого целого числа.

FLOAT(целое)

Для числа с плавающей точкой одинарной или двойной точности в зависимости от значения *целое*. Значение *целое* должно находиться в диапазоне от 1 до 53. Значения от 1 до 24 указывают на одинарную точность, значения от 25 до 53 указывают на двойную точность.

REAL Для числа с плавающей точкой одинарной точности.

DOUBLE или DOUBLE PRECISION

Для числа с плавающей точкой двойной точности.

FLOAT

Для числа с плавающей точкой двойной точности.

DECIMAL(точность, масштаб) или DEC(точность, масштаб)

Для десятичного числа.

Первое целое значение - точность числа, то есть общее число цифр. Это значение должно находиться в диапазоне от 1 до 31.

Второе целое значение - масштаб числа, то есть число цифр справа от десятичной точки. Допускаются значения в диапазоне от 0 до точности данного числа.

Если точность и масштаб не заданы, используются значения по умолчанию 5, 0.

Слова **NUMERIC** и **NUM** можно использовать как синонимы для **DECIMAL** и **DEC**.

CHARACTER(целое), CHAR(целое), CHARACTER или CHAR

Для символьной строки с фиксированной длиной *целое* от 1 до 254. Если длина не указана, предполагается 1 символ.

VARCHAR(целое)

Для символьной строки переменной длины с максимальной длиной *целое* от 1 до 32672.

NOT NULL

Запрещает наличие в столбце пустых значений.

имя-сервера

Сервер, зарегистрированный при помощи оператора **CREATE SERVER**. Дополнительную информацию об операторе **CREATE SERVER** смотрите далее, в разделе *Связанные ссылки*. Этот сервер будет использоваться для обращения к файлу с табличной структурой.

'путь' Полный путь файла с табличной структурой, к которому выполняется доступ, в одинарных кавычках. Файл данных может быть стандартным файлом или символической ссылкой, но не конвейером или иным нестандартным типом файла. Файлы данных должны быть доступны для чтения владельцем экземпляра DB2. Дополнительную информацию о владельцах экземпляра смотрите в руководстве *DB2 Administration Guide*.

SORTED

Указывает, отсортирован ли файл источника данных. Возможные значения опции - 'Y', 'y', 'n' и 'N'. Ее значение по умолчанию - 'N'.

Примечание: Если вы указали, что источник данных отсортирован, рекомендуется задавать значение 'Y' для `VALIDATE_DATA_FILE`.

'разделитель'

Разделитель столбцов в файле с табличной структурой, заключенный в одинарные кавычки. Допускаются только односимвольные разделители. Если разделитель столбцов не задан, по умолчанию принимается запятая. Разделитель не может быть одинарной кавычкой. Разделитель столбца не может совпадать с допустимыми данными столбца. Например, запятую нельзя использовать как разделитель столбцов, если в одном из столбцов данные содержат запятые.

'имя-ключевого-столбца'

Имя столбца, по которому отсортирован этот файл, в одинарных кавычках. Эта опция используется только для сортированных файлов.

Поддерживаются только ключи, состоящие из одного столбца. Значение должно быть именем столбца, заданного в операторе `CREATE NICKNAME`. Столбец должен быть отсортирован по возрастанию. Если это значение не задано для сортированного псевдонима, по умолчанию используется первый столбец файла, заданного псевдонимом.

Рекомендуется запретить наличие пустых значение в ключевом столбце, добавив к его определению в операторе `CREATE NICKNAME` опцию `NOT NULL`. Например:

```
CREATE NICKNAME tox (tox_id INTEGER NOT NULL, toxicity VARCHAR(100))
FOR SERVER tox_server1
  OPTIONS (FILE_PATH'/tox_data.txt', SORTED 'Y')
```

```
CREATE NICKNAME weights (mol_id INTEGER, wt VARCHAR(100) NOT NULL)
FOR SERVER wt_server
  OPTIONS (FILE_PATH'/wt_data.txt', SORTED 'Y', KEY_COLUMN 'WT')
```

Примечание: Эта опция регистрозависима, однако DB2 преобразует имена столбцов в верхний регистр, если только они не заданы в двойных кавычках. Следующий пример не будет работать правильно, поскольку столбец `empno` будет

преобразован DB2 в верхний регистр, а ключевой столбец empno задан в нижнем регистре. Поэтому столбец, обозначенный в качестве ключевого, не будет найден.

```
CREATE NICKNAME depart (  
    empno char(6) NOT NULL)  
FOR SERVER DATASTORE  
OPTIONS(FILE_PATH'data.txt', SORTED 'Y',  
    KEY_COLUMN'empno');
```

VALIDATE_DATA_FILE

Для сортированных файлов эта опция задает, будет ли оболочка проверять, что столбец ключа отсортирован по возрастанию, а также проверять наличие в нем пустых ключей. Для этой опции допустимы только значения 'Y' и 'N', указанные в одинарных кавычках. Проверка выполняется один раз при регистрации. Если опция не задана, проверка не производится.

Ниже приводится пример использования оператора CREATE NICKNAME для файла с табличной структурой DRUGDATA1.TXT, описанного в разделе "Что такое файлы с табличной структурой?", указанном далее в разделе Связанные ссылки:

```
CREATE NICKNAME DRUGDATA1(Dcode Integer NOT NULL, Drug CHAR(20),  
    Manufacturer CHAR(20))  
FOR SERVER biochem_lab OPTIONS(FILE_PATH '/usr/pat/DRUGDATA1.TXT',  
    COLUMN_DELIMITER ',', SORTED 'Y', KEY_COLUMN 'DCODE', VALIDATE_DATA_FILE 'Y')
```

Дополнительную информацию об операторе CREATE NICKNAME смотрите в справочнике *DB2 SQL Reference*. Дополнительную информацию о псевдонимах смотрите в руководстве *DB2 Administration Guide*.

Это последняя задача данной последовательности.

Задачи, связанные с данной темой:

- “После установки DB2 Life Sciences Data Connect” на стр. 10
- “Регистрация сервера для файлов с табличной структурой” на стр. 18
- “Регистрация псевдонимов для источников данных Documentum” на стр. 41
- “Регистрация псевдонимов для источников данных Excel” на стр. 77
- “Регистрация псевдонимов для источников данных BLAST” на стр. 104
- “Регистрация псевдонимов для источников данных XML” на стр. 126
- Глава 8, “Задание опций стоимости псевдонима” на стр. 143

Ограничения и особенности оболочки для файлов с табличной структурой

- При использовании этой оболочки не допускаются сквозные сеансы.
- Не разрешены многостолбцовые ключи.

- Сортированные файлы должны быть отсортированы только в восходящем порядке. Сортировка в нисходящем порядке не поддерживается.
- оболочка не применяет ограничение NOT NULL, но DB2 применяет его. Если создать псевдоним и задать ограничение NOT NULL для столбца и затем выбрать строку, содержащую пустое значение в этом столбце, DB2 выдаст ошибку SQL0407N, указывая, что нельзя присвоить пустое значение столбцу с ограничением NOT NULL.

Исключение из этого правила - псевдонимы сортированных файлов.

Ключевой столбец сортировки в таком псевдониме не может содержать пустых значений. Если в ключевом столбце сортированного псевдонима обнаружено пустое значение, появится сообщение об ошибке SQL1822N, указывающее, что отсутствует значение ключевого столбца.

- В DB2 Universal Database Enterprise Server Edition любой файл с табличной структурой, для которого создан псевдоним, должен быть доступен на каждом узле по одному и тому же имени пути. Этот файл не обязательно должен быть на узле DB2 Universal Database, если к нему возможен доступ с любого узла по общему пути.

Ссылки, связанные с данной темой:

- “Ограничения и особенности файлов для оболочки для файлов с табличной структурой” на стр. 24
- “Ограничения и особенности оболочки Documentum” на стр. 62
- “Ограничения оболочки Excel” на стр. 82
- “Ограничения файлов Excel” на стр. 82
- “Ограничения и особенности оболочки XML” на стр. 134

Ограничения и особенности файлов для оболочки для файлов с табличной структурой

- Файлы должны содержать одну запись в строке.
- Каждая запись должна содержать одно и то же число столбцов, отделенных разделителями.
- Каждая запись должна завершаться символом (символами) конца строки, стандартным для платформы, где установлена оболочка.
- Во всем файле должен использоваться один и тот же разделитель столбцов.
- Пустое значение представляется двумя идущими подряд разделителями или разделителем, после которого идет символ конца строки, если пустое значение находится в последнем поле строки.
- Отделитель дробной части определяется элементом RADIXCHAR категории LC_NUMERIC поддержки национального языка.
- Сортированные источники данных должны быть отсортированы в восходящем порядке в соответствии с последовательностью упорядочения

для текущей национальной версии, определяемой значениями категории LC_COLLATE поддержки национального языка.

- Кодовая страница базы данных должна совпадать с кодовой страницей файла; в противном случае вы можете получить неожиданные результаты.
- Файлы с многобайтными символами не поддерживаются.
- Если нечисленное поле содержит слишком длинное значение для типа данных его столбца, избыточные данные отсекаются.
- Если значение десятичного поля содержит после десятичной точки больше цифр, чем разрешено значением параметра масштаба типа данных его столбца, избыточные данные отсекаются.
- Максимальная длина строки - 32768.

Ссылки, связанные с данной темой:

- “Ограничения и особенности оболочки для файлов с табличной структурой” на стр. 23
- “Ограничения и особенности оболочки Documentum” на стр. 62
- “Ограничения файлов Excel” на стр. 82
- “Ограничения и особенности оболочки XML” на стр. 134

Модель управления доступом к файлам для оболочки для файлов с табличной структурой

Для доступа к файлам с табличной структурой СУБД использует полномочия владельца экземпляра DB2. Оболочка может обращаться только к файлам, доступным для этого ID пользователя (или ID группы). ID авторизации прикладной программы (ID, под которым установлено соединение с базой данных объединения) при этом не применяется.

Ссылки, связанные с данной темой:

- “Управление доступом для оболочки Documentum” на стр. 64
- “Модель управления доступом к файлам для оболочки Excel” на стр. 82

Советы по оптимизации для оболочки для файлов с табличной структурой

- В сортированных файлах данных система может выполнять поиск более эффективно, чем в несортированных.
- Для сортированных файлов можно повысить производительность, задав в запросе значение или диапазон для ключевого столбца.
- Статистику для псевдонимов файлов с табличной структурой необходимо обновлять вручную, изменяя производные таблицы SYSSTAT и SYSCAT. Дополнительную информацию о ручном обновлении производных таблиц SYSSTAT и SYSCAT смотрите в книге *DB2 Administration Guide*.

Ссылки, связанные с данной темой:

- “Советы по оптимизации оболочки BLAST” на стр. 115

Сообщения для оболочки для файлов с табличной структурой

В этом разделе перечисляются и описываются сообщения, которые вы можете получить при работе с оболочкой для файлов с табличной структурой. Дополнительная информация о сообщениях приведена в *Справочнике по сообщениям DB2*.

Таблица 5. Сообщения, генерируемые оболочкой для файлов с табличной структурой

Код ошибки	Сообщение	Объяснение
SQL0405N	Неверный числовой литерал "<литерал>" - значение вне допустимого диапазона.	Столбец в файле данных или предикат оператора SQL содержит значение, выходящее за границы допустимого диапазона значений для этого типа данных. Исправьте файл данных или определите для столбца другой, более подходящий тип данных.
SQL0408N	Значение несовместимо с типом данных назначения. Имя назначения - "<имя_столбца>".	Столбец в файле данных содержит символы, недопустимые для этого типа данных. Исправьте файл данных или определите для столбца другой, более подходящий тип данных.
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина - "Data source path is NULL" - Путь источника данных имеет пустое значение.)	Обратитесь в службу программной поддержки IBM.
SQL0901N	Оператор SQL завершился неудачно из-за существенной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина - "Key Column retrieval failure" - Ошибка получения ключевого столбца.)	Обратитесь в службу программной поддержки IBM.

Таблица 5. Сообщения, генерируемые оболочкой для файлов с табличной структурой (продолжение)

Код ошибки	Сообщение	Объяснение
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина - "STAT failed on data source. ERRNO = <номер_ошибки>" - Ошибка STAT для источника данных. ERRNO = <номер_ошибки>.)	Убедитесь, что у вас есть соответствующие полномочия для каталога. Убедитесь, что файл существует.
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина - "No column info found" - Не найдена информация столбцов.)	Обратитесь в службу программной поддержки IBM.
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина - "Unsupported operator" - Неподдерживаемый оператор.)	Обратитесь в службу программной поддержки IBM.
SQL1816N	Модуль оболочки "<имя_оболочки>" нельзя использовать для обращения к источнику данных "тип" ("<тип>" ""), который вы пытались определить для базы данных объединения.	Неверный тип сервера. Не задавайте тип сервера в операторе CREATE SERVER. Удалите ключевое слово TYPE и его значение и запустите этот оператор еще раз.

Таблица 5. Сообщения, генерируемые оболочкой для файлов с табличной структурой (продолжение)

Код ошибки	Сообщение	Объяснение
SQL1822N	От источника данных "<имя_сервера>" получен неожиданный код ошибки "ERRNO = <номер_ошибки>". Соответствующие текст и элементы - "Unable to read file" (Невозможно прочесть файл).	Проверьте значение номера ошибки. Убедитесь, что владелец экземпляра DB2 может читать этот файл. Затем запустите эту команду SQL еще раз.
SQL1822N	От источника данных "<имя_сервера>" получен неожиданный код ошибки "Data Error" (Ошибка данных). Соответствующие текст и элементы - "Data source is a non-standard file" (Источник данных - нестандартный файл).	Файл источника данных - это каталог, гнездо или FIFO. В качестве источников данных могут использоваться только обычные файлы. Измените опцию FILE_PATH, чтобы она указывала на правильный файл, и повторите команду SQL.
SQL1822N	От источника данных "<имя_сервера>" получен неожиданный код ошибки "ERRNO = <номер_ошибки>". Соответствующие текст и элементы - "File open error" (Ошибка открытия файла).	Модуль оболочки не смог открыть файл. Проверьте номер ошибки, чтобы определить ее причину. Исправьте проблему с источником данных и повторите команду SQL.
SQL1822N	От источника данных "<имя_сервера>" получен неожиданный код ошибки "Data Error" (Ошибка данных). Соответствующие текст и элементы - "Key column missing" (Ключевой столбец отсутствует).	В записи, полученной от источника данных, нет ключевого поля. Ключевой столбец должен иметь непустое значение. Исправьте данные или зарегистрируйте этот файл как несортированный псевдоним.
SQL1822N	От источника данных "<имя_сервера>" получен неожиданный код ошибки "Data Error" (Ошибка данных). Соответствующие текст и элементы - "File not sorted" (Несортированный файл).	Файл не был сортирован по ключевому столбцу. Выполните одно из следующих действий: измените опцию KEY_COLUMN, чтобы она указывала на правильный столбец; выполните сортировку файла данных; зарегистрируйте этот псевдоним как несортированный.

Таблица 5. Сообщения, генерируемые оболочкой для файлов с табличной структурой (продолжение)

Код ошибки	Сообщение	Объяснение
SQL1822N	От источника данных "<имя_сервера>" получен неожиданный код ошибки "Data Error" (Ошибка данных). Соответствующие текст и элементы - "Key exceeds definition size" (Превышение размера определения ключа).	Длина полученного от источника данных поля ключевого столбца превышает длину, заданную в определении столбца DB2, что может нарушить нормальную работу функций поиска оболочки. Исправьте данные или исправьте определение псевдонима и зарегистрируйте этот псевдоним снова.
SQL1822N	От источника данных "<имя_сервера>" получен неожиданный код ошибки "Data Error" (Ошибка данных). Соответствующие текст и элементы - "Line in data file exceeds 32k" (Длина строки в файле данных больше 32 Кбайт).	Длина строки файла данных превышает максимально разрешенную оболочкой длину строки. Длина строки не может быть больше 32768. Уменьшите длину строки в файле данных.
SQL1823N	Для типа данных "<тип-данных>" от сервера "<имя-сервера>" нет отображения типа данных.	В определении псевдонима задан неподдерживаемый тип данных. Переопределите псевдоним, используя только поддерживаемые типы данных.
SQL1881N	"<имя_опции>" - недопустимая опция "<компонента>" для объекта "<имя_столбца>".	Указанное значение не является допустимой опцией для указанного объекта. Удалите или измените неверную опцию и повторите оператор SQL.
SQL1882N	Для опции "Nickname" "COLUMN_DELIMITER" нельзя задать значение "<разделитель>" для объекта "<имя_псевдонима>".	Разделитель столбцов длиннее одного символа. Задайте в этой опции один символ. Затем выполните команду оператора SQL снова.
SQL1882N	Для опции "Nickname" "KEY_COLUMN" нельзя задать значение "<имя_столбца>" для объекта "<имя_псевдонима>".	Столбец, выбранный в качестве ключевого, не определен для этого псевдонима. Исправьте опцию KEY_COLUMN, задав в ней один из столбцов сортировки для этого псевдонима, и затем повторите команду SQL.
SQL1882N	Для опции "Nickname" "VALIDATE_DATA_FILE" нельзя задать значение "<значение_опции>" для объекта "<имя_псевдонима>".	Неверное значение опции. Допустимые значения: "Y" или "N". Исправьте опцию и зарегистрируйте псевдоним снова.

Таблица 5. Сообщения, генерируемые оболочкой для файлов с табличной структурой (продолжение)

Код ошибки	Сообщение	Объяснение
SQL1883N	"<имя_опции>" - обязательная опция "<компонента>" для сервера "<имя_объекта>".	В операторе SQL не задана обязательная опция для этой оболочки. Добавьте эту опцию и повторите оператор SQL.
SQL30090N	Операция недопустима в среде выполнения прикладной программы. Код причины = "21".	Вы пытались использовать сквозной сеанс. Оболочка для файлов с табличной структурой не поддерживает сквозные сеансы.

Ссылки, связанные с данной темой:

- “Сообщения оболочки Documentum” на стр. 64
- “Сообщения оболочки Excel” на стр. 83
- “Сообщения оболочки BLAST” на стр. 115
- “Сообщения оболочки XML” на стр. 135

Глава 4. Documentum как источник данных

В этой главе объясняется, что такое система Documentum, как добавить источники данных Documentum в вашу систему объединения, и приводятся сообщения об ошибках, связанные с оболочкой Documentum.

Что такое Documentum?

Documentum - это программный продукт, позволяющий управлять содержимым документов и атрибутами, поддерживая такие функции, как активирование, резервирование, рабочий поток и управление версиями. Documentum - это трехуровневая система клиент-сервер, настроенная над реляционной базой данных.

Docbase - это репозиторий Documentum, где хранится содержимое документов, атрибуты, отношения, версии, соответствия, форматы, рабочий поток и защита. Для запросов к данным Documentum используется язык запросов Documentum (Documentum Query Language, DQL) - расширенный диалект SQL. Docbase эквивалентна экземпляру Oracle или базе данных DB2[®] вместе с файлами содержимого документов. Метаданные хранятся в основной системе управления реляционными базами данных (relational database management system, RDBMS), а содержимое хранится в базе данных в виде двоичных больших объектов или в файловой системе сервера в виде файлов. Дополнительную информацию о Documentum смотрите в руководствах по Documentum.

Оболочка для Documentum позволяет добавлять источник данных Documentum в систему объединения DB2. Добавив источник данных Documentum в систему объединения, можно с помощью операторов SQL обращаться к объектам и зарегистрированным таблицам в Docbase Documentum и делать запросы к ним. Эти данные можно тем самым объединять с данными других источников в системе объединения без извлечения их непосредственно из источников данных. Оболочка Documentum использует для взаимодействия с сервером Documentum библиотеку клиента. Она дает доступ к двум версиям сервера Documentum: EDMS 98 (другое обозначение - версия 3) и 4i. На рис. 4 на стр. 32 показано, как работает оболочка Documentum.

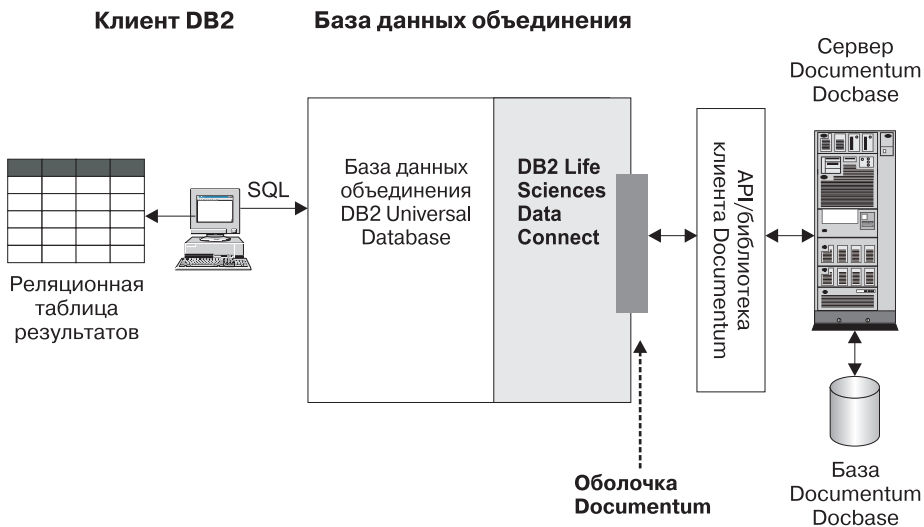


Рисунок 4. Как работает оболочка Documentum

Зарегистрировав оболочку Documentum, можно отображать объекты и зарегистрированные таблицы Docbase Documentum как реляционные таблицы. Для этого атрибуты Docbase отображают на имена столбцов в реляционной таблице DB2.

Например, в Табл. 6 перечислены атрибуты поднабора для dm_document, типа документа по умолчанию Docbase Documentum, и связанные с этим поднабором данные. Допустим, приведенный поднабор атрибутов важен для вас, и вы собираетесь включить эти атрибуты в свою систему баз данных объединения. Вы присвоили этому поднабору данных имя DrugAB_data.

Таблица 6. DrugAB_data

Title	Subject	Authors	Keywords
Влияние лекарства А на кроликов	Лекарство А	Curran, L.	кролики, лекарство А
Результаты испытания токсичности лекарства А	Лекарство А	Abelate, P., McMurtrey, K.	токсичность, лекарство А
Сочетаемость лекарства В	Лекарство В	DeNiro, R., Stone, S.	сочетаемость, лекарство В
Химическое строение лекарства В	Лекарство В	Boyslim, F.	строение, лекарство В

После регистрации оболочки Documentum данные можно запрашивать с помощью операторов SQL.

Ниже показан пример запроса заголовков и авторов публикаций по теме Лекарство А. Результаты показаны на Табл. 7.

```
SELECT title, authors
FROM drugAB_data
WHERE subject = 'Лекарство А'
```

Таблица 7. Результаты запроса

Title	Authors
Влияние лекарства А на кроликов	Curran, L.
Результаты испытания токсичности лекарства А	Abelite, P., McMurtrey, K.

Понятия, связанные с данным:

- “Что такое файлы с табличной структурой?” на стр. 13
- “Что такое Excel?” на стр. 73
- “Что такое BLAST?” на стр. 91
- “Что такое XML?” на стр. 119

Задачи, связанные с данной темой:

- “Добавление Documentum в систему объединения” на стр. 33

Добавление Documentum в систему объединения

Порядок действий:

Чтобы добавить на сервер объединения источник данных Documentum:

1. Выполните компоновку с библиотеками клиента Documentum.
2. Задайте указание на файл dmcl.ini клиента Documentum
3. Зарегистрируйте оболочку при помощи оператора CREATE WRAPPER.
4. Необязательно: для повышения производительности запросов задайте переменную среды DB2_DJ_COMM.
5. Зарегистрируйте сервер при помощи оператора CREATE SERVER.
6. Предоставьте пользователям доступ к источнику данных при помощи оператора CREATE USER MAPPING.
7. Зарегистрируйте псевдонимы с помощью оператора CREATE NICKNAME.
8. Создайте пользовательские функции при помощи оператора CREATE FUNCTION.

Операторы можно вводить в командной строке DB2. После регистрации можно выполнять запросы для этого источника данных.

Задачи, связанные с данной темой:

- “Компоновка с библиотеками клиента Documentum (только для AIX и операционной среды Solaris)” на стр. 34
- “Указание файла dmcl.ini клиента Documentum” на стр. 35
- “Регистрация оболочки Documentum” на стр. 36
- “Задание переменной среды DB2_DJ_COMM для оболочки Documentum” на стр. 37
- “Регистрация сервера для источников данных Documentum” на стр. 38
- “Отображение пользователей (оболочка Documentum)” на стр. 40
- “Регистрация псевдонимов для источников данных Documentum” на стр. 41
- “Регистрация пользовательских функций для источников данных Documentum” на стр. 50
- “Добавление файлов с табличной структурой в систему объединения” на стр. 16
- “Добавление Excel в систему объединения” на стр. 75
- “Добавление BLAST в систему объединения” на стр. 96
- “Добавление XML в систему объединения” на стр. 123

Компоновка с библиотеками клиента Documentum (только для AIX и операционной среды Solaris)

Эта задача - часть большей задачи *Добавление Documentum в систему объединения*. Для доступа к источникам данных Documentum систему объединения DB2 нужно скомпоновать с библиотеками клиента. В процессе компоновки создается библиотека оболочки для каждого источника данных, с которым связывается сервер объединения. Библиотека оболочки Documentum создается, когда вы запускаете сценарий `djxlinkDctm`.

Порядок действий:

Чтобы запустить сценарий `djxlinkDctm`:

1. Задайте переменную среды `LSDC_DMCL`, чтобы она указывала на каталог, где расположена библиотека клиента Documentum.

Например:

```
export LSDC_DMCL=/usr/documentum/product/3.1.7
```

2. Введите от имени пользователя `root` команду:

```
ksh djxlinkDctm
```

Примечание: Команду `djxlinkDctm` необходимо повторно запустить после установки пакета исправлений DB2 Universal Database.

Следующая задача в этой последовательности задач - *Указание на файл dmcl.ini клиента Documentum.*

Задачи, связанные с данной темой:

- “Указание файла dmcl.ini клиента Documentum” на стр. 35

Указание файла dmcl.ini клиента Documentum

Эта задача - часть большей задачи *Добавление Documentum в систему объединения.* Управление доступом к базам документов Documentum осуществляется с помощью файла dmcl.ini клиента Documentum. Чтобы экземпляр DB2 получил доступ к базе документов Documentum, надо задать его переменные среды в файле dmcl.ini клиента Documentum.

Порядок действий:

Чтобы задать переменные среды:

1. Отредактируйте файл db2dj.ini, задав одну из следующих переменных среды:
DOCUMENTUM=<путь>
DMCL_CONFIG=<путь>/dmcl.ini

где <путь> - полное имя каталога, содержащего файл dmcl.ini, который вы хотите использовать.

По умолчанию файл dmcl.ini Documentum находится в каталоге /pkgs/documentum. Если в файл включены обе строки, используется DMCL_CONFIG.

В AIX и операционной среде Solaris файл db2dj.ini находится в каталоге \$HOME/sql11ib/cfg.

В Windows файл db2dj.ini находится в каталоге x:\sql11ib\cfg, где x: - диск, на котором расположен каталог sql11ib.

Примечание: Проверьте, что имя посредника, которому посылают отчеты все доступные базы документов для экземпляра DB2, указано в файле dmcl.ini, как показано на рис. 5 на стр. 36.

```
##### ФАЙЛ КОНФИГУРАЦИИ КЛИЕНТА DOCUMENTUM #####
#
# Copyright Documentum 1994.
# Версия 3.1 сервера Documentum.
#
# Сгенерированный файл инициализации клиента для сервера Documentum.
#
# Единственная НЕОБХОДИМАЯ информация в этом файле -
# раздел [DOCBROKER_PRIMARY] и запись для хоста.
# Значением хоста должно быть имя хоста, на котором
# работает общесетевой посредник документов

[DOCBROKER_PRIMARY]
host = server16.comp2.big.com
```

Рисунок 5. Пример файла dtcl.ini с указанным именем посредника

2. Перезапустите экземпляр DB2, введя команды:

```
db2stop
db2start
```

Следующая задача в этой последовательности задач - *Регистрация оболочки Documentum*.

Задачи, связанные с данной темой:

- “Задание переменной среды DB2_DJ_COMM для оболочки файлов с табличной структурой” на стр. 17
- “Компоновка с библиотеками клиента Documentum (только для AIX и операционной среды Solaris)” на стр. 34
- “Задание переменной среды DB2_DJ_COMM для оболочки Documentum” на стр. 37
- “Задание переменной среды DB2_DJ_COMM для оболочки BLAST” на стр. 102
- “Задание переменной среды DB2_DJ_COMM для оболочки XML” на стр. 124

Регистрация оболочки Documentum

Эта задача - часть большей задачи *Добавление Documentum в систему объединения*. Чтобы можно было обращаться к источникам данных, необходимо зарегистрировать оболочку. Оболочка - это механизм, посредством которого серверы объединения подключаются к источникам данных и получают от них данные. Оболочки устанавливаются в системе в виде файлов библиотек.

Порядок действий:

Чтобы зарегистрировать оболочку Documentum, используйте оператор CREATE WRAPPER.

Например, чтобы создать в AIX из файла библиотеки по умолчанию libdb21sdctm.a оболочку Documentum с именем Dctm_Wrapper , нужен следующий оператор:

```
CREATE WRAPPER Dctm_Wrapper LIBRARY 'libdb21sdctm.a'  
  OPTIONS(DB2_FENCED 'N');
```

Таблицу имен файлов библиотек по умолчанию для оболочки Documentum на поддерживаемых платформах смотрите в описании задачи "После установки DB2 Life Sciences Data Connect" по ссылке, приведенной далее в разделе Задачи, связанные с данной. Дополнительную информацию об операторе CREATE WRAPPER смотрите в книге *DB2 SQL Reference*.

Следующая задача в этой последовательности задач - *Задание переменной среды DB2_DJ_COMM для оболочки Documentum*.

Задачи, связанные с данной темой:

- “После установки DB2 Life Sciences Data Connect” на стр. 10
- “Регистрация оболочки файлов с табличной структурой” на стр. 16
- “Задание переменной среды DB2_DJ_COMM для оболочки Documentum” на стр. 37
- “Регистрация оболочки Excel” на стр. 75
- “Регистрация оболочки BLAST” на стр. 101
- “Регистрация оболочки XML” на стр. 124

Задание переменной среды DB2_DJ_COMM для оболочки Documentum

Эта задача - часть большей задачи *Добавление Documentum в систему объединения*. Чтобы улучшить производительность при обращении к источникам данных Documentum, задайте переменную среды DB2_DJ_COMM. Эта переменная определяет, будет ли сервер объединения загружать оболочку при инициализации.

Порядок действий:

Чтобы задать переменную среды DB2_DJ_COMM, выполните команду db2set, указав в ней библиотеку оболочки, заданную в соответствующем операторе CREATE WRAPPER.

Например:

```
db2set DB2_DJ_COMM='libdb21sdctm.a'
```

Слева и справа от знака равенства (=) не должно быть пробелов.

При загрузке библиотек оболочек при запуске базы данных возникают дополнительные затраты времени. Чтобы избежать этих затрат, просто задайте библиотеки, к которым нужно обращаться.

Дополнительную информацию о переменной DB2_DJ_COMM смотрите в книге *DB2 Administration Guide*.

Следующая задача в этой последовательности задач - *Регистрация сервера для источников данных Documentum*.

Задачи, связанные с данной темой:

- “Задание переменной среды DB2_DJ_COMM для оболочки файлов с табличной структурой” на стр. 17
- “Регистрация оболочки Documentum” на стр. 36
- “Регистрация сервера для источников данных Documentum” на стр. 38
- “Задание переменной среды DB2_DJ_COMM для оболочки BLAST” на стр. 102
- “Задание переменной среды DB2_DJ_COMM для оболочки XML” на стр. 124

Регистрация сервера для источников данных Documentum

Эта задача - часть большей задачи *Добавление Documentum в систему объединения*. Зарегистрировав оболочку, необходимо зарегистрировать соответствующий сервер.

Порядок действий:

Для регистрации сервера Documentum в системе объединения используйте оператор CREATE SERVER.

Например, предположим, что есть сервер с именем Dctm_Server1 для оболочки Dctm_Wrapper созданной соответствующим оператором CREATE WRAPPER. Предположим, что на этом сервере находится база документов, работающая в AIX и использующая для хранения данных Oracle. Чтобы зарегистрировать этот сервер, выполните следующий оператор:

```
CREATE SERVER Dctm_Server1
  TYPE DCTM
  VERSION 3
  WRAPPER Dctm_Wrapper
  OPTIONS( NODE 'Dctm_Docbase',
           OS_TYPE 'AIX',
           RDBMS_TYPE 'ORACLE');
```

Аргументы

TYPE Задаёт тип источника данных. Тип для Documentum - DCTM. Это обязательный аргумент.

VERSION

Задаёт версию источника данных. Для EDMS98 это значение - '3'. Для 4i значение - '4'. Это обязательный аргумент.

WRAPPER

Задаёт имя оболочки, связанной с этим сервером. Это обязательный аргумент.

Опции

CONTENT_DIR

Задаёт имя локально доступного корневого каталога для хранения файлов содержимого, полученных псевдосто́лбцами GET_FILE, GET_FILE_DEL, GET_RENDITION и GET_RENDITION_DEL. Он должен быть доступен для записи всем пользователям, которые могут использовать эти псевдосто́лбцы. Значение по умолчанию - /tmp. Эта опция необязательна.

NODE Задаёт реальное имя базы документов Documentum. Это обязательная опция.

OS_TYPE

Указывает операционную систему сервера базы документов. Допустимые значения - AIX, SOLARIS и WINDOWS. Это обязательная опция.

RDBMS_TYPE

Задаёт реляционную СУБД, используемую базой документов. Допустимые значения: DB2, INFORMIX, ORACLE, SQLSERVER или SYBASE. Это обязательная опция.

TRANSACTIONS

Задаёт режим транзакций сервера. Допустимые значения:

- NONE – транзакции запрещены.
- QUERY – транзакции разрешены только для методов Dctm_Query.
- ALL – транзакции разрешены для метода Dctm_Query. В данном выпуске ALL выполняет такую же функцию, что и QUERY.

Значение по умолчанию - QUERY. Эта опция необязательна.

Дополнительную информацию об операторе CREATE SERVER смотрите в книге *DB2 SQL Reference*.

Следующая задача в этой последовательности задач - *Отображение пользователей (оболочка Documentum)*.

Задачи, связанные с данной темой:

- “Регистрация сервера для файлов с табличной структурой” на стр. 18
- “Задание переменной среды DB2_DJ_COMM для оболочки Documentum” на стр. 37
- “Отображение пользователей (оболочка Documentum)” на стр. 40
- “Регистрация сервера для источника данных Excel” на стр. 76
- “Регистрация сервера для источника данных BLAST” на стр. 103
- “Регистрация сервера для источника данных XML” на стр. 125

Отображение пользователей (оболочка Documentum)

Эта задача - часть большей задачи *Добавление Documentum в систему объединения*. Чтобы предоставить пользователям доступ к источнику данных, необходимо отобразить их на ранее определенные серверы.

Порядок действий:

Для отображения пользователей на серверы объединения используется оператор CREATE USER MAPPING.

Например, следующий оператор CREATE USER MAPPING отображает пользователя Chuck на пользователя Charles на сервере Dctm_Server1.

```
CREATE USER MAPPING FOR Chuck SERVER Dctm_Server1  
OPTIONS(REMOTE_AUTHID 'Charles', REMOTE_PASSWORD 'Charles_pw');
```

Можно также определить собственные отображения пользователей. В следующем примере USER - ключевое слово, обозначающее текущего пользователя, а не пользователя с именем USER.

```
CREATE USER MAPPING FOR USER SERVER Dctm_Server1  
OPTIONS(REMOTE_AUTHID 'Lisa', REMOTE_PASSWORD 'Lisa_pw')
```

Дополнительную информацию об операторе CREATE USER MAPPING смотрите в руководстве *DB2 SQL Reference*.

Следующая задача в этой последовательности задач - *Регистрация псевдонимов для источников данных Documentum*.

Задачи, связанные с данной темой:

- “Регистрация сервера для источников данных Documentum” на стр. 38
- “Регистрация псевдонимов для источников данных Documentum” на стр. 41

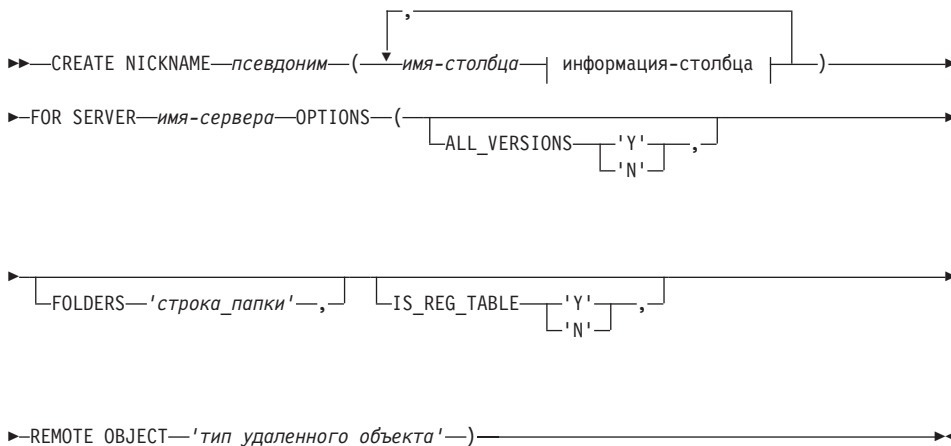
Регистрация псевдонимов для источников данных Documentum

Эта задача - часть большей задачи *Добавление Documentum в систему объединения*. После регистрации сервера и отображения пользователей на этот сервер необходимо зарегистрировать соответствующие псевдонимы. Псевдонимы используются для обозначения источников данных Documentum в запросе.

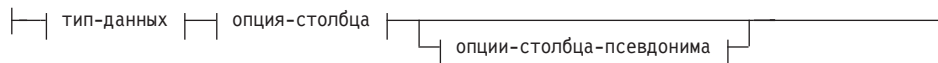
Порядок действий:

При регистрации псевдонимов оператор CREATE NICKNAME используется для создания псевдонимов каждой базы Docbase для всех необходимых типов объектов или зарегистрированных таблиц.

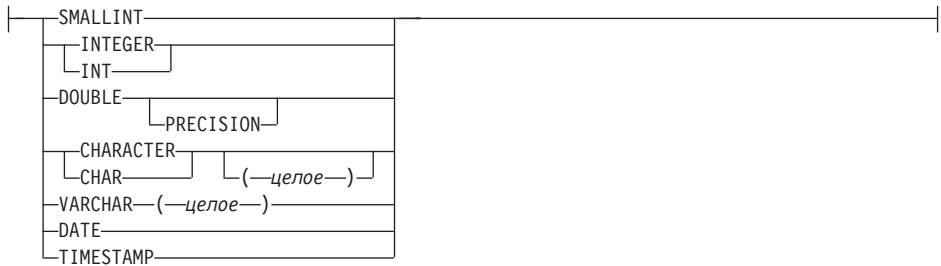
Синтаксис оператора CREATE NICKNAME для Documentum:



информация-столбца:



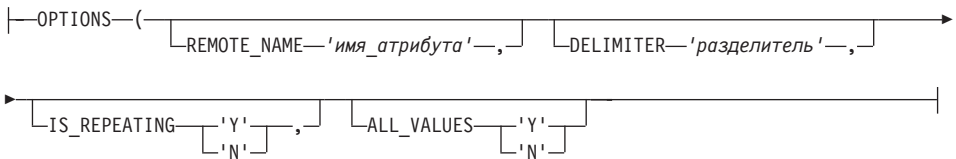
тип-данных:



опция-столбца:



опции-столбца-псевдонима:



Дополнительную информацию об операторе CREATE NICKNAME смотрите в руководстве *DB2 SQL Reference*.

Опции столбцов

NOT NULL

Все столбцы с одним значением, кроме определенных как TIMESTAMP и DATE, должны быть определены как NOT NULL. Атрибуты с возможностью повторения в псевдонимах не должны быть определены как NOT NULL.

Опции столбца псевдонима

Значения опции столбца псевдонима должны быть заключены в одинарные кавычки.

ALL_VALUES

Указывает, что все значения атрибута с возможностью повторения будут возвращены один за другим с использованием указанного разделителя. Если эта опция отсутствует или для нее указано значение 'N', будет возвращено только последнее значение атрибута с возможностью повторения. Как уже указывалось в описании DELIMITER, ALL_VALUES можно указывать только для столбцов VARCHAR, для которых у опции IS_REPEATING установлено значение 'Y' (и нельзя, когда IS_REG_TABLE = 'Y').

DELIMITER

Задаёт строку разделителя, которая будет использоваться при конкатенации нескольких значений атрибута с возможностью повторения. Этот разделитель может состоять из одного или нескольких символов. Разделитель по умолчанию - запятая. Эта опция применяется только для атрибутов объектов с типом данных VARCHAR, если для опции IS_REPEATING задано 'Y'. Эта опция необязательна.

IS_REPEATING

Указывает, что столбец допускает несколько значений. Допустимые значения: 'Y' или 'N'. Значение по умолчанию - 'N'. Эта опция необязательна.

REMOTE_NAME

Задаёт имя соответствующего атрибута столбца Documentum. Эта опция отображает имена удалённых атрибутов или столбцов на имена локальных столбцов DB2. По умолчанию используются имена столбцов DB2. Это необязательная опция.

Опции псевдонима

Значения опции псевдонима должны быть заключены в одинарные кавычки.

ALL_VERSIONS

Указывает, будет ли производиться поиск для всех версий объектов. Допустимые значения - 'y', 'Y', 'n' и 'N'. По умолчанию используется 'N', что означает, что при обработке запроса учитываются только текущие версии объектов. Опция не действует, если IS_REG_TABLE = 'Y'. Эта опция необязательна.

FOLDERS

Задаёт строку, содержащую один или несколько логически скомбинированных и синтаксически правильных предикатов FOLDER Documentum. Задание предикатов FOLDER ограничивает набор документов, представленный псевдонимом, до документов, содержащихся в указанных папках.

При задании этой опции заключите все значение опции FOLDERS в одинарные кавычки, а внутри строки вместо одинарных кавычек используйте двойные.

Например, если вы хотите вставить:

```
FOLDER('/Tools',DESCEND) OR FOLDER('/Cars')
```

Задайте следующую опцию FOLDERS:

```
FOLDERS 'FOLDER("/Tools",DESCEND) OR FOLDER("/Cars")'
```

Опция не действует, если IS_REG_TABLE = 'Y'. Это необязательная опция.

IS_REG_TABLE

Указывает, является ли объект, указанный опцией REMOTE_OBJECT, зарегистрированной таблицей Documentum. Допустимые значения - 'y', 'Y', 'n' и 'N'. Значение по умолчанию - 'N'. Это необязательная опция.

Примечание: Изменяя эту опцию в операторе ALTER NICKNAME, нельзя заменить псевдоним объекта Documentum на зарегистрированную таблицу (и наоборот). Вместо этого необходимо для этого псевдонима выполнить операторы DROP и затем CREATE.

REMOTE_OBJECT

Задаёт связанное с псевдонимом имя типа объекта Documentum. Это имя может быть любым типом объекта или зарегистрированной таблицей Documentum. В случае зарегистрированной таблицы ему должно предшествовать имя владельца таблицы. Если зарегистрированная таблица принадлежит владельцу базы документов, в качестве имени владельца можно использовать dm_dbo. Это обязательная опция.

Примечание: Использование ALTER NICKNAME для изменения значения опции REMOTE_OBJECT приведет к ошибкам, если структура нового объекта не идентична структуре исходного объекта.

Понятие псевдостолбцов

Оператор CREATE NICKNAME определяет также 6 псевдостолбцов. Эти столбцы используются для доступа к содержимому объекта и другой информации.

Эти псевдостолбцы и их значения перечислены в Табл. 8.

Таблица 8. Имена и определения псевдостолбцов.

Имя псевдостолбца	Определение
GET_FILE	VARCHAR (255)
GET_FILE_DEL	VARCHAR (255)
GET_RENDITION	VARCHAR (255)
GET_RENDITION_DEL	VARCHAR (255)
HITS	INTEGER
SCORE	DOUBLE

В Табл. 9 перечислены псевдостолбцы для условий SELECT.

Таблица 9. Псевдостолбцы для условий SELECT

Имя псевдостолбца	Описание
GET_FILE	<p>Получает для текущей строки в дополнение к значениям столбца файл содержимого.</p> <p>Расширение для файла содержимого - это его имя в формате Documentum. Если существует файл с таким же именем, он будет перезаписан.</p> <p>GET_FILE пытается получить базовый формат объекта. Его значение в строке представляет собой a_content_type для объекта. Если у объекта нет файла содержимого, его значение - это строка символов "no_content".</p> <p>Например:</p> <pre>SELECT object_name, DCTM.GET_FILE FROM ...</pre> <p>Файл содержимого помещается в каталог сервера, указанный опцией сервера CONTENT_DIR. В этом каталоге файл помещается в подкаталог, имя которого - локальное имя пользователя DB2. Если такой каталог не существует, он будет создан.</p> <p>Расширением файла будет его расширение DOS, определенное в базе документов для типа формата документов. Например, для документов MS Word используется расширение ".doc".</p> <p>Возвращает строку символов "no_content" или полное имя файла.</p>
GET_FILE_DEL	<p>Эта функция аналогична GET_FILE, но GET_FILE_DEL сначала удаляет файл, полученный для предыдущей строки, если она есть в данном запросе. Возвращает строку символов "no_content" или полное имя файла.</p>

Таблица 9. Псевдостолбцы для условий SELECT (продолжение)

Имя псевдостолбца	Описание
GET_RENDITION	<p>Для текущей строки в дополнение к значениям столбца считывает файл содержимого для этого соответствия - копию исходного документа в другом формате.</p> <p>Расширение для файла содержимого - это его имя в формате Documentum. Если существует файл с таким же именем, он будет перезаписан.</p> <p>Чтобы указать формат соответствия, в условии WHERE надо задать предикат в форме DCTM.RENDITION_FORMAT(<format) = 1.</p> <p>Например: <pre>SELECT object_name, get_rendition FROM ... WHERE DCTM.RENDITION_FORMAT('pdf')=1</pre> </p> <p>GET_RENDITION пытается получить указанное соответствие для объекта. Его значение в строке - это a_content_type объекта, или строка символов "no_content", если у объекта нет файла содержимого, или строка символов "not_found", если соответствия не существует.</p> <p>Файл содержимого помещается в каталог сервера, указанный опцией сервера CONTENT_DIR. В этом каталоге файл помещается в подкаталог, имя которого - локальное имя пользователя DB2. Если такой каталог не существует, он будет создан.</p> <p>Расширением файла будет его расширение DOS, определенное в базе документов для типа формата документов. Например, для документов MS Word используется расширение ".doc".</p> <p>Возвращает строку символов "no_content", "not found" или полное имя файла.</p>
GET_RENDITION_DEL	<p>Эта функция аналогична GET_RENDITION, но GET_RENDITION_DEL сначала удаляет файл, полученный для предыдущей строки, если она есть в данном запросе. Возвращает строку символов "no_content", "not found" или полное имя файла.</p>

В Табл. 10 на стр. 47 перечислены псевдостолбцы для условий SELECT в запросах, содержащих условия поиска.

Таблица 10. Псевдостолбцы для условий *SELECT* в запросах, содержащих условия поиска

Имя псевдостолбца	Описание
HITS	<p>Возвращает целое число - количество мест в документе, для которых выполняется условие поиска.</p> <p>Например:</p> <pre data-bbox="438 401 1161 475">SELECT r_object_id, object_name, hits FROM std_doc WHERE DCTM.SEARCH_WORDS (''workflow'' OR ''flowchart'')=1</pre> <p>Для каждого из возвращенных документов суммарное число вхождений слов "workflow" и "flowchart" в содержимом этого документа возвращается как значение HITS.</p> <p>Псевдостолбец HITS удобен, если у документа только один файл содержимого. Такая ситуация является типичной. Этот столбец можно использовать в спецификаторе условия WHERE для оператора SELECT. Однако в этом случае его надо указать также в условии SELECT.</p>
SCORE	<p>Возвращает ранг релевантности объекта.</p> <p>Этот псевдостолбец используется в сочетании с оператором Documentum ACCRUE. Вместе они позволяют определить, сколько из заданных слов найдено в каждом из возвращенных документов.</p> <p>Например:</p> <pre data-bbox="438 956 1233 1083">SELECT object_name, score FROM std_doc WHERE DCTM.SEARCH_TOPIC('<ACCRUE>("document","management","workflow")')=1 AND SCORE >=75</pre> <p>Этот оператор возвращает все документы, в содержимом которых найдено два или три из указанных слов. Если в документе найдено только одно из этих слов, ему присваивается ранг 50, что не удовлетворяет критерий условия WHERE, и такой документ не возвращается. Если найдено два из трех слов, документу присваивается ранг 75. Если найдено все три слова, ранг документа - 88.</p> <p>Псевдостолбец SCORE используется для документов, у которых только один файл содержимого. Такая ситуация является типичной.</p> <p>SCORE можно использовать в условии SELECT, только если в условии WHERE есть функция SEARCH_WORDS() или SEARCH_TOPIC(). В условии WHERE он используется в сочетании с оператором ACCRUE.</p> <p>Информацию об операторе ACCRUE смотрите в документации к Documentum.</p>

Пример CREATE NICKNAME

Следующий оператор CREATE NICKNAME определяет псевдоним std_doc. Std_doc связан с базой документов Documentum с типом объектов dm_document. Табл. 11 отображает атрибуты и типы данных Documentum на имена и типы данных реляционных столбцов DB2, которые потом используются для построения оператора CREATE NICKNAME.

Таблица 11. Отображение атрибутов Documentum на столбцы DB2 для std_doc nickname

Имя атрибута Documentum	Тип данных Documentum	Имя столбца DB2	Тип данных DB2	Допускает повторы?	Допускает пустые значения?
object_name	string(255)	object_name	varchar	Нет	Нет
r_object_id	ID	object_id	char(16)	Нет	Нет
r_object_type	string(32)	object_type	varchar	Нет	Нет
title	string(255)	title	varchar	Нет	Нет
subject	string(128)	subject	varchar	Нет	Нет
authors	string(32)	author	varchar	Да	Да
keywords	string(32)	keyword	varchar	Да	Да
r_creation_date	time	creation_date	timestamp	Нет	Да
r_modify_date	time	modified_date	timestamp	Нет	Да
a_status	string(16)	status	varchar	Нет	Нет
a_content_type	string(32)	content_type	varchar	Нет	Нет
r_content_size	double	content_size	integer	Нет	Нет
owner_name	string(32)	owner_name	varchar	Нет	Да

Табл. 12 описывает все атрибуты Documentum, использованные в псевдониме.

Таблица 12. Описание атрибутов Documentum для псевдонима std_doc

Имя атрибута Documentum	Описание
object_name	Определенное пользователем имя объекта.
r_object_id	Уникальный идентификатор этого объекта, заданный во время создания.
r_object_type	Тип объекта, заданный во время создания объекта.
title	Определенный пользователем заголовок объекта.
subject	Определенная пользователем тема объекта.
authors	Определенный пользователем список авторов для объекта.

Таблица 12. Описание атрибутов Documentum для псевдонима std_doc (продолжение)

Имя атрибута	Описание
Documentum	
keywords	Список определенных пользователем ключевых слов для объекта.
r_creation_date	Дата и время создания объекта.
r_modify_date	Дата и время последнего изменения объекта.
a_status	Устанавливается сервером при передаче задачи маршрутизатора. Это значение берется из значений, присваиваемых attached_task_status в объекте маршрутизатора.
a_content_type	Формат файла содержимого объекта.
r_content_size	Число байтов содержимого. Для многостраничных документов этот атрибут отражает размер первого содержимого, связанного с таким документом.
owner_name	Имя владельца объекта (пользователя, создавшего объект).

Атрибуты Табл. 11 на стр. 48 преобразуются в следующий оператор CREATE NICKNAME.

```
CREATE NICKNAME std_doc (
  object_name varchar(255) not null,
  object_id char(16) not null OPTIONS(REMOTE_NAME 'r_object_id'),
  object_type varchar(32) not null OPTIONS(REMOTE_NAME 'r_object_type'),
  title varchar(255) not null,
  subject varchar(128) not null,
  author varchar(32) OPTIONS(REMOTE_NAME 'authors', IS_REPEATING 'Y'),
  keyword varchar(32) OPTIONS(REMOTE_NAME 'keywords', IS_REPEATING 'Y'),
  creation_date timestamp OPTIONS(REMOTE_NAME 'r_creation_date'),
  modified_date timestamp OPTIONS(REMOTE_NAME 'r_modify_date'),
  status varchar(16) not null OPTIONS(REMOTE_NAME 'a_status'),
  content_type varchar(32) not null OPTIONS(REMOTE_NAME 'a_content_type'),
  content_size integer not null OPTIONS(REMOTE_NAME 'r_content_size'),
  owner_name varchar(32))
FOR SERVER Dctm_Server2 OPTIONS (REMOTE_OBJECT 'dm_document', IS_REG_TABLE 'N')
```

После выполнения оператора CREATE NICKNAME можно использовать псевдоним std_doc для запроса в системе объединения. Можно также выполнить объединение псевдонима std_doc с другими псевдонимами и таблицами в системе объединения.

Примечание: Из-за псевдостолбцов число столбцов для этого псевдонима в каталоге на 6 больше, чем указано в операторе CREATE NICKNAME.

Можно использовать утилиту CreateNicknameFile, чтобы автоматически отобразить типы Documentum на типы DB2 и создать первоначальный оператор

CREATE NICKNAME. Дополнительную информацию об утилите CreateNicknameFile смотрите по приведенным ниже ссылкам на задачи, связанные с данной.

Следующая задача в этой последовательности задач - *Регистрация пользовательских функций для источников данных Documentum*.

Задачи, связанные с данной темой:

- “Регистрация псевдонимов для файлов с табличной структурой” на стр. 19
- “Отображение пользователей (оболочка Documentum)” на стр. 40
- “Регистрация пользовательских функций для источников данных Documentum” на стр. 50
- “Регистрация псевдонимов для источников данных Excel” на стр. 77
- “Регистрация псевдонимов для источников данных BLAST” на стр. 104
- “Регистрация псевдонимов для источников данных XML” на стр. 126
- Глава 8, “Задание опций стоимости псевдонима” на стр. 143

Регистрация пользовательских функций для источников данных Documentum

Эта задача - часть большей задачи *Добавление Documentum в систему объединения*. Для регистрации пользовательских функций нужно использовать оператор CREATE FUNCTION. Эти функции можно использовать для доступа к некоторым особым возможностям Documentum, например, к полнотекстовому поиску или получению в запросах содержимого документов.

Пользовательские функции для предикатов перечислены в Табл. 13 на стр. 52.

DB2 не поддерживает тип данных BOOLEAN. Поэтому для создания правильных операторов SQL нужно явно проверять значение каждой пользовательской функции. Реализация оболочки поддерживает только семантику "DSTM.<функция>(<аргументы>) = 1", независимо от примененного для проверки оператора сравнения.

Примечание: Ссылки на функцию TOPIC - это ссылки на функцию Documentum, входящую в полнотекстовую систему индексации, поставляемую компанией Verity.

Порядок действий:

Для регистрации пользовательских функций используйте оператор CREATE FUNCTION.

Все пользовательские функции должны регистрироваться с именем схемы DCTM. Полное имя каждой функции - DCTM.<имя_функции>.

В следующем примере регистрируется пользовательская функция ANY_EQ.

```
CREATE FUNCTION DCTM.ANY_EQ (CHAR(), CHAR()) RETURNS INTEGER
AS TEMPLATE DETERMINISTIC NO EXTERNAL ACTION
```

Каждую пользовательскую функцию необходимо зарегистрировать один раз для каждой базы данных DB2, для которой установлена оболочка Documentum.

Чтобы помочь вам регистрировать пользовательские функции, поставляется файл примера create_function_mappings.ddl, который находится в каталоге sqllib/samples/lifesci. Этот файл содержит определения для каждой пользовательской функции. Можно выполнить этот файл ddl, чтобы зарегистрировать пользовательские функции для каждой базы данных DB2, для которой установлена оболочка Documentum.

Правила для строк аргументов пользовательской функции

Все аргументы, передаваемые в виде строк, должны удовлетворять следующим правилам:

- Каждая строка заключается в апострофы (одинарные кавычки).
- Апострофы внутри строки представляются двумя символами апострофа.

Использование пользовательских функций в запросах

В следующих примерах показано использование пользовательских функций в запросах.

Чтобы вывести имя объекта и авторов для документов из псевдонима std_doc, у которых среди авторов есть 'Dave Winters':

```
SELECT object_name,authors FROM std_doc
WHERE DCTM.ANY_EQ(authors,'Dave Winters')=1
```

Чтобы вывести имя объекта и автора для документов из псевдонима std_doc, у которых среди авторов есть 'Dave Winters' или 'Jon Doe':

```
SELECT object_name,authors FROM std_doc
WHERE DCTM.ANY_IN(authors,'Dave Winters','Jon Doe')=1
```

Чтобы вывести имя объекта и r_object_id и получить содержимое файла для документов из псевдонима std_doc, содержащих строки, подобные 'Dave Win%' в столбце авторов:

```
SELECT object_name, r_object_id, get_file FROM std_doc
WHERE DCTM.ANY_LIKE(authors,'Dave Win%')=1
```

Таблица пользовательских функций

В Табл. 13 перечислены пользовательские функции для предикатов.

Таблица 13. Пользовательские функции для предикатов

Имя функции	Описание
ANY_EQ(arg1, arg2)	<p>Проверяет, что у атрибута с возможностью повторения есть значение, равное заданному. Получает два обязательных аргумента:</p> <p>arg1 Задаёт имя столбца, представляющего атрибут с возможностью повторения.</p> <p>arg2 Задаёт значение для сравнения.</p> <p>Например:</p> <pre>... WHERE DCTM.ANY_EQ(authors, 'Dave Winters')=1</pre>
ANY_NE(arg1, arg2)	<p>Проверяет, что у атрибута с возможностью повторения есть значение, не равное заданному. Получает два обязательных аргумента:</p> <p>arg1 Задаёт имя столбца, представляющего атрибут с возможностью повторения.</p> <p>arg2 Задаёт значение для сравнения.</p> <p>Например:</p> <pre>... WHERE DCTM.ANY_NE(authors, 'Dave Winters')=1</pre>
ANY_LT(arg1, arg2)	<p>Проверяет, что у атрибута с возможностью повторения есть значение меньше заданного. Получает два обязательных аргумента:</p> <p>arg1 Задаёт имя столбца, представляющего атрибут с возможностью повторения.</p> <p>arg2 Задаёт значение для сравнения.</p> <p>Например:</p> <pre>... WHERE DCTM.ANY_LT(num_approvers, 4)=1</pre>
ANY_GT(arg1, arg2)	<p>Проверяет, что у атрибута с возможностью повторения есть значение больше заданного. Получает два обязательных аргумента:</p> <p>arg1 Задаёт имя столбца, представляющего атрибут с возможностью повторения.</p> <p>arg2 Задаёт значение для сравнения.</p> <p>Например:</p> <pre>... WHERE DCTM.ANY_GT(num_approvers, 3)=1</pre>

Таблица 13. Пользовательские функции для предикатов (продолжение)

Имя функции	Описание
ANY_LE(arg1, arg2)	<p>Проверяет, что у атрибута с возможностью повторения есть значение не больше заданного. Получает два обязательных аргумента:</p> <p>arg1 Задаёт имя столбца, представляющего атрибут с возможностью повторения.</p> <p>arg2 Задаёт значение для сравнения.</p> <p>Например:</p> <pre>... WHERE DCTM.ANY_LE(num_approvers,2)=1</pre>
ANY_GE(arg1, arg2)	<p>Проверяет, что у атрибута с возможностью повторения есть значение не меньше заданного. Получает два обязательных аргумента:</p> <p>arg1 Задаёт имя столбца, представляющего атрибут с возможностью повторения.</p> <p>arg2 Задаёт значение для сравнения.</p> <p>Например:</p> <pre>... WHERE DCTM.ANY_GE(num_approvers,1)=1</pre>
ANY_IN(arg1, arg2 – arg11)	<p>Проверяет, что у атрибута с возможностью повторения есть значение, равное какому-либо из десяти значений в списке. Получает от 3 до 11 аргументов одного типа данных:</p> <p>arg1 Задаёт имя столбца, представляющего атрибут с возможностью повторения.</p> <p>arg2–arg11 Список значений для сравнения, разделённых запятыми.</p> <p>Например:</p> <pre>... WHERE DCTM.ANY_IN(authors,'Crick','Watson')=1</pre>

Таблица 13. Пользовательские функции для предикатов (продолжение)

Имя функции	Описание
ANY_LIKE(arg1, arg2)	<p>Проверяет, что у атрибута с возможностью повторения есть значение, подобное заданному. Получает два обязательных аргумента:</p> <p>arg1 Задаёт имя столбца, представляющего атрибут с возможностью повторения.</p> <p>arg2 Задаёт шаблон для сравнения с подстроками; шаблон задаётся в апострофах.</p> <p>Например:</p> <pre>... WHERE DCTM.ANY_LIKE(authors, 'Dave Win%')=1 OR DCTM.ANY_LIKE(keywords, '%_%')=1</pre> <p>Примечание: Условие ESCAPE не поддерживается в предикатах ANY_LIKE().</p>
ANY_NOT_LIKE(arg1, arg2)	<p>Проверяет, что у атрибута с возможностью повторения есть значение, не подобное заданному. Получает два обязательных аргумента:</p> <p>arg1 Задаёт имя столбца, представляющего атрибут с возможностью повторения.</p> <p>arg2 Задаёт шаблон для сравнения с подстроками; шаблон задаётся в апострофах.</p> <p>Например:</p> <pre>... WHERE DCTM.ANY_NOT_LIKE(authors, 'Dave Win%')=1 OR DCTM.ANY_NOT_LIKE(keywords, '%_%')=1</pre> <p>Примечание: Условие ESCAPE не поддерживается в предикатах ANY_NOT_LIKE().</p>
ANY_NULL(arg)	<p>Проверяет атрибут с возможностью повторения на условие IS NULL. Получает один обязательный аргумент - имя атрибута с возможностью повторения или атрибута DATE или TIMESTAMP с одним значением.</p> <p>Например:</p> <pre>... WHERE DCTM.ANY_NULL(authors)=1</pre>
ANY_NOT_NULL(arg)	<p>Проверяет атрибут с возможностью повторения на условие IS NOT NULL. Получает один обязательный аргумент - имя атрибута с возможностью повторения.</p> <p>Например:</p> <pre>... WHERE DCTM.ANY_NOT_NULL(authors)=1</pre>

Таблица 13. Пользовательские функции для предикатов (продолжение)

Имя функции	Описание
ANY_SAME_INDEX(arg1 – arg10)	<p>Проверяет, совпадают ли индексы найденных значений атрибутов с возможностью повторения. В качестве аргументов получает от двух до десяти функций ANY_xx().</p> <p>Следующий пример ищет документы, у которых хотя бы один из авторов имеет имя Ken и при этом не работает в Калифорнийском университете в Дэвисе (UCD - University of California at Davis).</p> <pre>... WHERE DCTM.ANY_SAME_INDEX(ANY_EQ(author_name, 'Ken'), DCTM.ANY_NE(author_affiliation, 'UCD'))=1</pre>
CABINET(arg) и CABINET_TREE(arg)	<p>Получает один обязательный аргумент - полное имя шкафа базы Docbase.</p> <p>Например:</p> <pre>... WHERE DCTM.CABINET('/Tools')=1 ... WHERE DCTM.CABINET_TREE('/MyDocs')=1</pre> <p>Чтобы задать несколько шкафов, используйте несколько функций CABINET и CABINET_TREE.</p> <p>Например:</p> <pre>... WHERE DCTM.CABINET('/Tools')=1 OR DCTM.CABINET_TREE('/Parts')=1</pre>
FOLDER(arg) и FOLDER_TREE(arg)	<p>Получает один обязательный аргумент - полное имя папки или шкафа базы Docbase.</p> <p>Например:</p> <pre>... DCTM.FOLDER('/Tools/Drills')=1 ... DCTM.FOLDER_TREE('/MyDocs/WhitePapers')=1</pre> <p>Чтобы задать несколько папок, используйте несколько функций FOLDER и FOLDER_TREE.</p> <p>Например:</p> <pre>... DCTM.FOLDER('/Tools/Drills')=1 OR DCTM.FOLDER_TREE('/Animals/Horses')=1</pre>

Таблица 13. Пользовательские функции для предикатов (продолжение)

Имя функции	Описание
RENDITION_FORMAT (формат)	<p>Работает с псевдостолбцами GET_RENDITION и GET_RENDITION_DEL, чтобы установить формат структуры получаемых данных. Получает один аргумент - символьную строку, задающую формат.</p> <p>Следующий пример получает документ в формате PDF:</p> <pre>SELECT get_rendition FROM WHERE DCTM.RENDITION_FORMAT('pdf')=1</pre>
USER(1)	<p>Сравнивает ID автора Documentum с ID текущего пользователя. Получает один фиктивный аргумент, который должен быть числом 1.</p> <p>Например:</p> <pre>... WHERE approver = DCTM.USER(1)</pre> <p>Примечание: Чтобы установить соответствие между ID автора Documentum и ID автора DB2, используйте оператор CREATE USER MAPPING. Дополнительную информацию об отображениях пользователей смотрите в разделе, указанном ниже в ссылках.</p>
SEARCH_WORDS(agr)	<p>Получает один обязательный строчный аргумент - список отдельных слов, заключенных в простые кавычки; в этом списке могут использоваться операции AND, OR или NOT и скобки для задания порядка их применения. Слова не могут содержать пробелы и должны быть заключены в простые кавычки.</p> <p>Например:</p> <pre>... DCTM.SEARCH_WORDS(''yeast'' AND (''bread'' OR ''cake'') AND NOT ''wedding'')=1</pre>
SEARCH_TOPIC(agr)	<p>Получает один обязательный строчный аргумент - оператор запроса Verity TOPIC, который без изменений передается Documentum и Verity.</p> <p>Например:</p> <pre>... WHERE DCTM.SEARCH_TOPIC('"quick"')=1</pre>

Дополнительную информацию об операторе CREATE FUNCTION смотрите в книге *DB2 SQL Reference*.

Это последняя задача данной последовательности.

Задачи, связанные с данной темой:

- “Регистрация псевдонимов для источников данных Documentum” на стр. 41

Выполнение запросов для источников данных Documentum

Зарегистрировав оболочку, можно выполнять запросы SQL для источника данных Documentum. В этом разделе приведены некоторые примеры запросов.

Порядок действий:

Для выполнения запросов нужно использовать в операторах SQL псевдоним и определенные для него столбцы, аналогично тому, как используются имя обычной таблицы и ее столбцов.

Следующий запрос выводит все документы базы Docbase с названиями 'Test Document':

```
SELECT object_name
FROM std_doc
WHERE object_name='Test Document';
```

Следующий запрос использует пользовательскую функцию ANY_EQ для вывода всех документов, среди авторов которых есть 'Joe Doe'.

```
SELECT object_name
FROM std_doc
WHERE DCTM.ANY_EQ(author, 'Joe Doe')=1
```

Следующий запрос использует функции FOLDER_TREE и SEARCH_WORDS для поиска в шкафу Approved всех документов, содержащих текст "protein".

```
SELECT object_name
FROM std_doc
WHERE DCTM.FOLDER_TREE('/Approved')=1
      AND DCTM.SEARCH_WORDS('protein')=1
```

Следующий запрос использует псевдостолбец GET_FILE и пользовательские функции FOLDER_TREE и ANY_IN для получения имен файлов на сервере DB2, в которые было записано содержимое документов в шкафу Approved, среди авторов которых есть кто-либо из указанных.

```
SELECT object_name, object_id, get_file
FROM std_doc
WHERE DCTM.FOLDER_TREE('/Approved')=1
      AND DCTM.ANY_IN(author, 'Mary Black', 'Joe Carson', 'Peter Miller')=1
```

Задачи, связанные с данной темой:

- “Выполнение запросов для источников данных Excel” на стр. 79
- “Запуск запросов к источникам данных XML” на стр. 133

Что такое утилита CreateNicknameFile для оболочки Documentum?

Бесплатно загружаемая утилита Docbasic CreateNicknameFile служит для создания файла ASCII, содержащего полные определения любых объектов или зарегистрированных таблиц Docbase. Выходной файл можно отредактировать, чтобы:

- Определить пользовательские локальные имена для столбцов и атрибутов. Изначально в качестве локальных и удаленных имен используются имена из Docbase.
- Удалить ненужные столбцы и атрибуты. В единственном предварительно определенном типе документов Documentum (dm_document) 59 атрибутов для версии EDMS98 и 76 атрибутов для версии 4i. Большинство из них содержит метаданные для управления документами низкого уровня и разработки прикладных программ. Удаление не представляющих интереса атрибутов может повысить эффективность операторов SQL SELECT * без ухудшения производительности.
- Добавить значение для опции FOLDERS, чтобы ограничить поиск по данному псевдониму конкретными папками Documentum.
- Если нужно, изменить отображение DATE в TIMESTAMP. Эта утилита генерирует отображение DQL DATE в DB2[®] DATE, что представляется наиболее полезным.
- Изменить отображения CHAR в VARCHAR или наоборот в зависимости от нужд прикладной программы.

Эту утилиту надо установить в Docbase и запускать ее через графический пользовательский интерфейс Documentum Windows[®]. Файлы, генерируемые этой утилитой, специфичны для Docbase, где утилита установлена.

Задачи, связанные с данной темой:

- “Установка утилиты CreateNicknameFile (оболочка Documentum)” на стр. 58
- “Конфигурирование утилиты CreateNicknameFile (оболочка Documentum)” на стр. 59
- “Отображение типа объектов DM_ID в зарегистрированные таблицы Documentum” на стр. 60

Установка утилиты CreateNicknameFile (оболочка Documentum)

Утилита CreateNicknameFile помогает написать операторы CREATE NICKNAME для источников данных Documentum.

Порядок действий:

Чтобы установить эту утилиту:

1. Загрузите утилиту CreateNicknameFile из раздела загрузки сайта продукта DB2 Life Sciences Data Connect по адресу:
<http://www.ibm.com/software/data/db2/lifesciencesdataconnect/>
2. Используйте графический пользовательский интерфейс EDMS98 Workspace или 4i Desktop Client, чтобы импортировать утилиту - файл CreateNicknameFile.txt. Утилиту можно импортировать как тип процедуры в шкаф или папку базы Dcbase и дать ей любое имя.
3. Включите переключатель **Can be run by user** (Может запускаться пользователем) на странице свойств для импортированного объекта CreateNicknameFile.txt.

Понятия, связанные с данным:

- “Что такое утилита CreateNicknameFile для оболочки Documentum?” на стр. 58

Задачи, связанные с данной темой:

- “Конфигурирование утилиты CreateNicknameFile (оболочка Documentum)” на стр. 59
- “Отображение типа объектов DM_ID в зарегистрированные таблицы Documentum” на стр. 60

Конфигурирование утилиты CreateNicknameFile (оболочка Documentum)

Утилита CreateNicknameFile помогает написать операторы CREATE NICKNAME для источников данных Documentum.

Предварительные требования:

Перед конфигурированием утилиты CreateNicknameFile ее необходимо установить. Дополнительную информацию об установке утилиты CreateNicknameFile смотрите в разделе “Установка утилиты CreateNicknameFile (оболочка Documentum)”, ссылка на который есть ниже в разделе Задачи, связанные с данной.

Порядок действий:

Чтобы сконфигурировать эту утилиту после ее установки:

1. Дважды щелкните по значку утилиты, чтобы запустить ее.
2. Введите имя документа/типа объекта Documentum. Имя по умолчанию - dm_document.

Примечание: Если нужно создать файл псевдонима для зарегистрированной таблицы, задайте имя dm_registered. Если задано имя dm_registered, вас попросят ввести полное имя таблицы в

формате <владелец>.<имя_таблицы>. Если таблицей владеет владелец базы Docbase (типичный случай), в качестве имени владельца можно задать dm_dbo.

Утилита предполагает использование соглашения об именовании для имен псевдонимов зарегистрированных таблиц. По этому соглашению имя таблицы дополняется префиксом "rt_", указывающим на то, что это "зарегистрированная таблица". Если вы не хотите использовать это соглашение, можете изменить предложенный утилитой псевдоним.

3. Введите имя сервера, связанное с создаваемым псевдонимом.
4. Введите имя псевдонима.

Используйте понятные имена, которые уникальны внутри экземпляра DB2. Утилита предполагает использование соглашения об именовании - <имя_сервера>.<тип_объекта>, поскольку может понадобиться определить один и тот же <тип_объекта> на нескольких серверах. Если вы не хотите следовать этому соглашению, можете изменить предложенный утилитой псевдоним.

5. Введите имя выходного файла.

Имя по умолчанию - C:\Temp\nickname.txt. Каталог для выходного файла должен уже существовать, и у пользователя, запустившего утилиту, должны быть права на запись в этот каталог.

Когда вы введете всю запрошенную информацию, будет создан файл псевдонима, затем этот файл будет открыт в текстовом редакторе.

Понятия, связанные с данным:

- “Что такое утилита CreateNicknameFile для оболочки Documentum?” на стр. 58

Задачи, связанные с данной темой:

- “Регистрация псевдонимов для источников данных Documentum” на стр. 41
- “Установка утилиты CreateNicknameFile (оболочка Documentum)” на стр. 58

Отображение типа объектов DM_ID в зарегистрированные таблицы Documentum

Определения столбцов, созданные утилитой CreateNicknameFile, соответствуют требованиям оболочки Documentum, включая правильное отображение каждого типа данных на соответствующий тип данных DB2. Единственное исключение - Documentum не поддерживает тип данных DM_ID в зарегистрированных таблицах. Утилита предполагает, что столбец зарегистрированной таблицы используется для ID объекта, если он определен как строка длиной 16 символов

и его имя оканчивается на "_id". Столбец с типом данных DM_ID утилита отображает на тип данных DB2 CHAR(16). Во всех других случаях все столбцы строчных типов и типа VARCHAR отображаются на тип данных DB2 VARCHAR.

Порядок действий:

Чтобы гарантировать правильное отображение типов данных:

1. Проверьте определения типов данных столбцов в выходном файле, созданном утилитой CreateNicknameFile.
2. Если утилита задала неверное отображение типа данных столбца Documentum на тип данных DB2, измените тип данных DB2 перед использованием этого файла для регистрации псевдонима для DB2.

Понятия, связанные с данным:

- “Что такое утилита CreateNicknameFile для оболочки Documentum?” на стр. 58

Задачи, связанные с данной темой:

- “Установка утилиты CreateNicknameFile (оболочка Documentum)” на стр. 58
- “Конфигурирование утилиты CreateNicknameFile (оболочка Documentum)” на стр. 59

Двойное определение для атрибутов с возможностью повторения (оболочка Documentum)

Для наилучшего выполнения запросов оболочкой нужно определить для каждого атрибута полностью эквивалентный ему тип данных DB2. Это значит, что целочисленные значения Documentum должны быть определены как целочисленные значения DB2 и так далее. Однако такие определения мешают получать несколько значений для атрибутов с возможностью повторения, если их тип - не VARCHAR. Для таких столбцов возвращается только значение index[0].

Это ограничение вызвано тем, что, где это возможно, оболочка возвращает одну строку результатов для одного объекта Docbase. Это ограничение становится проблемой только при выборе атрибутов с возможностью повторения. Однако можно определить второй столбец - для того же удаленного атрибута с возможностью повторения, но с типом данных VARCHAR.

Используйте это имя столбца в списке SELECT, чтобы получить все значения в виде списка значений с разделителями. (Используемый разделитель задается опцией DELIMITER.)

Для таких столбцов с многими значениями лучше использовать стандартизованные локальные имена. Для этого в качестве локальных имен столбцов с многими значениями можно использовать локальные имена столбцов с точными типами данных, добавив перед ними префикс "m_".

Например, предположим, для атрибута с возможностью повторения Documentum есть столбец псевдонима с именем approval_dates (даты одобрения), определенный с типом данных TIMESTAMP. Можно создать второй столбец псевдонима с именем m_approval_dates и определить его как столбец типа данных VARCHAR. Затем можно использовать имя m_approval_dates в списке SELECT, чтобы получить все даты одобрения в виде списка с разделителями.

Нет необходимости использовать двойное определение для атрибутов с возможностью повторения, точный тип данных которых - VARCHAR.

Ограничения и особенности оболочки Documentum

В этом разделе представлен список ограничений и особенностей, связанных с использованием оболочки Documentum.

- Ограничения на возвращение значений атрибутов с возможностью повторения: Возвращается только последнее значение для
 - Атрибутов с возможностью повторения всех типов, кроме VARCHAR
 - Столбцов VARCHAR, когда задана опция ALL_VALUES 'N'

Чтобы обойти это ограничение, можно создать двойное определение столбца атрибута с возможностью повторения. Дополнительную информацию о создании двойного определения для атрибутов с возможностью повторения смотрите в разделе, указанном ниже в ссылках.

Кроме того, несколько значений атрибутов с возможностью повторения, определенных как VARCHAR, возвращаются в виде одной строки с разделителями значений. Разделитель задается опцией DELIMITER псевдонима в операторе CREATE NICKNAME.

- Сквозные сеансы не поддерживаются.
- Для каждого соединения прикладной программы DB2 с базой данных DB2 оболочка Documentum может поддерживать не более 10 одновременных сеансов Documentum, а каждый сеанс может одновременно управлять не более, чем 10 запросами Documentum. Для одной прикладной программы DB2 одновременно могут выполняться несколько запросов; выполнение запроса начинается при его передаче системе DB2 и заканчивается при закрытии соответствующего указателя набора результатов. В любой момент времени для всего набора выполняемых в этот момент запросов должны выполняться следующие ограничения:

- Все используемые в запросах псевдонимы должны находится не более чем на 10 различных сервера Documentum.
- Нельзя использовать более 10 псевдонимов с одного сервера Documentum.

Псевдонимы, используемые в нескольких запросах или упомянутые несколько раз в одном запросе, нужно учитывать столько раз, сколько они упоминаются.

- Оболочка Documentum использует клиентскую библиотеку Версии 3.1.7a для AIX. Для работы с Documentum 4i нужно получить эту версию клиентской библиотеки от Documentum (если она еще не установлена).
- Поскольку DB2 не поддерживает тип данных Boolean, для большинства пользовательских функций (кроме USER), используемых в условии WHERE, нужно выполнять проверку на "=1", так как эти функции возвращают целочисленное значение.

Например:

```
"... WHERE DSTM.ANY_EQ(authors,'Dave Winters')=1"
```

- Из-за ограничений DB2 пользовательская функция USER определена с целочисленным аргументом, который не используется.
- Все серверы, работающие для одного экземпляра DB2, должны использовать один файл параметров конфигурации Documentum dmcl.ini.
- Максимальное число значений в пользовательской функции ANY_IN для атрибутов с возможностью повторения - 10 для одного оператора. Однако можно объединить несколько операторов при помощи операции OR.
- Для пользовательской функции ANY_SAME_INDEX максимальное число проверок значений в одном индексе для атрибутов с возможностью повторения равно 10. Эти проверки должны быть проверками типа AND, выполняемыми слева направо.
- У оболочки нет возможностей, специфичных для конкретных кодовых страниц.

Ссылки, связанные с данной темой:

- “Ограничения и особенности оболочки для файлов с табличной структурой” на стр. 23
- “Ограничения и особенности файлов для оболочки для файлов с табличной структурой” на стр. 24
- “Ограничения оболочки Excel” на стр. 82
- “Ограничения файлов Excel” на стр. 82
- “Ограничения и особенности оболочки XML” на стр. 134

Управление доступом для оболочки Documentum

Для запросов применяются права пользователей в базе Docbase. В результаты запроса включаются только документы, для которых пользователь обладает по крайней мере правами на чтение.

Ссылки, связанные с данной темой:

- “Модель управления доступом к файлам для оболочки для файлов с табличной структурой” на стр. 25
- “Модель управления доступом к файлам для оболочки Excel” на стр. 82

Сообщения оболочки Documentum

В этом разделе перечисляются и описываются сообщения, которые вы можете получить при работе с оболочкой Documentum. Дополнительную информацию о сообщениях смотрите в книге *Справочник по сообщениям DB2*.

Таблица 14. Сообщения, генерируемые оболочкой Documentum

Код ошибки	Сообщение	Объяснение
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "sqlno_crule_save_plans [100]:rc (-2144272209) Empty plan list detect" - Обнаружен пустой список планов.)	Запрос SQL, переданный в DB2, нельзя обработать оболочкой. Исправьте синтаксис и повторите обработку оператора.

Таблица 14. Сообщения, генерируемые оболочкой Documentum (продолжение)

Код ошибки	Сообщение	Объяснение
SQL0901N	Оператор SQL завершился неудачно из-за несущественной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "dmAPI exec failed: [DM_QUERY_E_BAD_QUAL] error: "The attribute qualifier, A0, for attribute <column_name>, is not a valid qualifier."" - Ошибка выполнения dmAPI: [DM_QUERY_E_BAD_QUAL] ошибка: "Спецификатор атрибута, A0, для атрибута <имя_столбца> не является правильным спецификатором".)	Для опции REMOTE_OBJECT псевдонима заданы неправильные тип Documentum или зарегистрированная таблица. Измените псевдоним, чтобы он использовал правильные тип объектов Documentum или зарегистрированную таблицу.
SQL0901N	Оператор SQL завершился неудачно из-за несущественной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Invalid null column specified" - Задан неверный пустой столбец.)	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL0901N	Оператор SQL завершился неудачно из-за несущественной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Nickname specification is empty" - Определение псевдонима пусто.)	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL0901N	Оператор SQL завершился неудачно из-за несущественной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "The Output object is empty or incomplete" - Выходной объект пуст или не завершен.)	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.

Таблица 14. Сообщения, генерируемые оболочкой Documentum (продолжение)

Код ошибки	Сообщение	Объяснение
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Unexpected number of columns requested" - Запрошено неожиданное число столбцов.)	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "No column information found" - Не найдена информация столбцов.)	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Unsupported column type requested" - Запрошен неподдерживаемый тип столбца.)	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Incorrect Column definition" - Неверное определение столбца.)	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.

Таблица 14. Сообщения, генерируемые оболочкой Documentum (продолжение)

Код ошибки	Сообщение	Объяснение
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Inconsistent type; DB2 request != nickname type" - Несовместимый тип; запрос DB2 не соответствует типу псевдонима.)	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Output parameter is not NULL" - Выходной параметр имеет непустое значение.)	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Query output variable is not NULL" - Выходная переменная запроса имеет непустое значение.)	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Invalid timestamp length" - Неверная длина отметки времени.)	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.

Таблица 14. Сообщения, генерируемые оболочкой Documentum (продолжение)

Код ошибки	Сообщение	Объяснение
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Inconsistent number of columns" - Несовместимое число столбцов.)	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Could not access data when converting values" - Нет доступа к данным при преобразовании значений.)	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Failed to initialize the DMCL client" - Ошибка инициализации клиента DMCL.)	Не удалось инициализировать клиент Documentum. Обратитесь к администратору системы.
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Get_User returned NULL" - Функция Get_User вернула пустое значение.)	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.

Таблица 14. Сообщения, генерируемые оболочкой Documentum (продолжение)

Код ошибки	Сообщение	Объяснение
SQL0901N	Оператор SQL завершился неудачно из-за несущественной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина - "Get_Local_User returned NULL" - Функция Get_Local_User вернула пустое значение.)	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL0901N	Оператор SQL завершился неудачно из-за несущественной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Begin Transaction failed" - Не удалось начать транзакцию.)	Documentum сообщает, что не смог начать транзакцию. Обратитесь к администратору системы.
SQL0901N	Оператор SQL завершился неудачно из-за несущественной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Input parameter was not NULL" Входной параметр имел непустое значение.)	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL0901N	Оператор SQL завершился неудачно из-за несущественной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Dctm functions must be like Dctm.function(...)=1" - Функции DCTM должны использоваться в форме DCTM.функция(...)=1.)	Пользователь не указал =1 в правой части предиката для функции DCTM. Исправьте синтаксис и повторите запрос.

Таблица 14. Сообщения, генерируемые оболочкой Documentum (продолжение)

Код ошибки	Сообщение	Объяснение
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Invalid column number requested" - Затребован неверный номер столбца.)	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL1881N	"DELIMITER" - недопустимая опция "COLUMN" для объекта "<имя-столбца>"	Для столбца "<имя-столбца>" задана опция DELIMITER, но не задана опция IS_REPEATING.
SQL1882N	Для опции "SERVER" "DEFAULT" нельзя задать значение "<значение-опции>" для объекта "<имя-сервера>".	Для опции сервера RDBMS_TYPE задано недопустимое значение. Допустимы следующие значения: DB2, INFORMIX, ORACLE, SQLSERVER и SYBASE.
SQL1882N	Для опции "SERVER" "TRANSACTIONS" нельзя задать значение "<значение-опции>" для объекта "<имя-сервера>".	Для опции сервера TRANSACTIONS задано недопустимое значение. Допустимы следующие значения: NONE, QUERY, PASSTHRU и ALL.
SQL1882N	Для опции "NICKNAME" "IS_REG_TABLE" нельзя задать значение "<значение-опции>" для объекта "<псевдоним>".	Для опции псевдонима IS_REG_TABLE задано недопустимое значение. Допустимы значения 'Y' и 'N'.
SQL1882N	Для опции "NICKNAME" "ALL_VERSIONS" нельзя задать значение "<значение-опции>" для объекта "<псевдоним>".	Для опции псевдонима ALL_VERSIONS задано недопустимое значение. Допустимы значения 'Y' и 'N'.
SQL1882N	Для опции "SERVER" "OS_TYPE" нельзя задать значение "<значение-опции>" для объекта "<имя-сервера>".	Для опции сервера OS_TYPE задано недопустимое значение. Допустимы значения AIX, HPUX, SOLARIS и WINDOWS.
SQL1882N	Для опции "NICKNAME" "FOLDERS" нельзя задать значение "<значение-опции>" для объекта "<псевдоним>".	Для опции псевдонима FOLDERS задано недопустимое значение. Его нельзя задавать для таблицы, для которой опция IS_REG_TABLE имеет значение 'Y'.

Таблица 14. Сообщения, генерируемые оболочкой Documentum (продолжение)

Код ошибки	Сообщение	Объяснение
SQL1882N	Для опции "NICKNAME" "VERSIONS" нельзя задать значение "<значение-опции>" для объекта "<псевдоним>".	Для опции псевдонима VERSIONS задано недопустимое значение. Допустимы значения 'Y' и 'N'. Кроме того, опцию VERSIONS 'Y' нельзя задавать для таблицы, для которой опция IS_REG_TABLE имеет значение 'Y'.
SQL30090N	Операция недопустима в среде выполнения прикладной программы. Код причины = "Invalid column name, IS_REG_TABLE, or IS_REPEATING specified in nickname" (В псевдониме задано неверное имя столбца или значение опции IS_REG_TABLE или IS_REPEATING).	Проверьте, правильно ли заданы в операторе псевдонима опции IS_REG_TABLE, IS_REPEATING и REMOTE_NAME и имена столбцов.
SQL30090N	Операция недопустима в среде выполнения прикладной программы. Код причины = "db2dj.ini missing DOCUMENTUM or DMCL_CONFIG env var" (В db2dj.ini отсутствует переменная среды DOCUMENTUM или DMCL_CONFIG).	Не заданы обязательные переменные среды. Задайте их в файле db2dj.ini.
SQL30090N	Операция недопустима в среде выполнения прикладной программы. Код причины = "Failed to open log file for debugging" (Не удалось открыть файл журнала для отладки).	Недоступен файл журнала, используемый для устранения неисправностей. Обратитесь к администратору системы.
SQL30090N	Операция недопустима в среде выполнения прикладной программы. Код причины = "Only one search condition may be specified" (Можно задать только одно условие поиска).	В одном запросе можно задавать только одну пользовательскую функцию поиска.

Таблица 14. Сообщения, генерируемые оболочкой Documentum (продолжение)

Код ошибки	Сообщение	Объяснение
SQL30090N	Операция недопустима в среде выполнения прикладной программы. Код причины = "Failed to create content directory" (Не удалось создать каталог содержимого).	Убедитесь, что агент DB2 может писать в каталог назначения.
SQL30090N	Операция недопустима в среде выполнения прикладной программы. Код причины = "Failed to change permissions on content file" (Не удалось изменить разрешения для файла содержимого).	Убедитесь, что агент DB2 может писать в каталог назначения.

Ссылки, связанные с данной темой:

- “Сообщения для оболочки для файлов с табличной структурой” на стр. 26
- “Сообщения оболочки Excel” на стр. 83
- “Сообщения оболочки BLAST” на стр. 115
- “Сообщения оболочки XML” на стр. 135

Глава 5. Excel как источник данных

В этой главе объясняется, что такое система Excel, как добавить источники данных Excel в вашу систему объединения, и приводятся сообщения об ошибках, связанные с оболочкой Excel.

Что такое Excel?

Электронная таблица или "книга" Excel - это файл с расширением xls, созданный с помощью программы Microsoft® (MS) Excel. DB2 Life Sciences Data Connect поддерживает электронные таблицы Excel 97 и Excel 2000. На рис. 6 показано, как оболочка Excel соединяет электронные таблицы с вашей системой объединения.

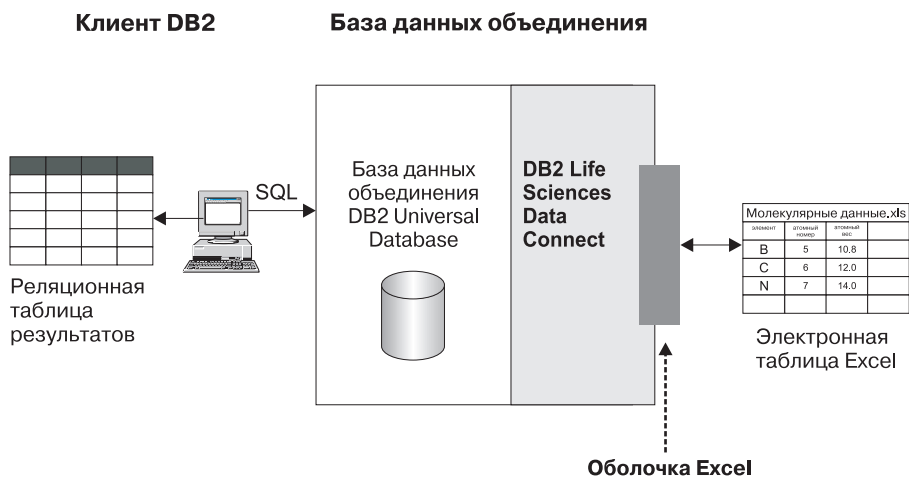


Рисунок 6. Как работает оболочка Excel

В оболочке Excel для отображения столбцов электронной таблицы Excel в столбцы системы объединения DB2® используется оператор CREATE NICKNAME. В Табл. 15 показан пример данных электронной таблицы, хранящихся в файле Compound_Master.xls.

Таблица 15. Пример электронной таблицы Compound_Master.xls

	A	B	C	D
1	соединение_A	1,23	367	проверено
2	соединение_G		210	

Таблица 15. Пример электронной таблицы Compound_Master.xls (продолжение)

	A	B	C	D
3	соединение_F	0,000425536	174	проверено
4	соединение_Y	1,00256		проверено
5	соединение_Q		1024	
6	соединение_B	33,5362		
7	соединение_S	0,96723	67	проверено
8				
9	соединение_O	1,2		проверено

При использовании стандартных команд SQL эта информация обычно недоступна. Установив и зарегистрировав оболочку Excel, можно обращаться к этим данным как к обычному реляционному источнику данных. Например, если нужно узнать данные о всех соединениях с молекулярным весом, превышающим 100, достаточно запустить следующий запрос SQL:

```
SELECT * FROM compound_master WHERE mol_count > 100
```

Результаты запроса показаны в Табл. 16.

Таблица 16. Результаты запроса

COMPOUND_NAME	WEIGHT	MOL_COUNT	WAS_TESTED
соединение_A	1,23	367	проверено
соединение_G		210	
соединение_F	0,000425536	174	проверено
соединение_Q		1024	

Понятия, связанные с данным:

- “Что такое файлы с табличной структурой?” на стр. 13
- “Что такое Documentum?” на стр. 31
- “Что такое BLAST?” на стр. 91
- “Что такое XML?” на стр. 119

Задачи, связанные с данной темой:

- “Добавление Excel в систему объединения” на стр. 75

Ссылки, связанные с данной темой:

- “Предварительное требование для оболочки Excel” на стр. 75

Предварительное требование для оболочки Excel

Предварительное требование для использования оболочки источников данных Excel:

- Прежде, чем можно будет использовать оболочку Excel, необходимо установить программу MS Excel на сервере, где установлен DB2 Life Sciences Data Connect.

Задачи, связанные с данной темой:

- “Добавление Excel в систему объединения” на стр. 75
- “Регистрация оболочки Excel” на стр. 75

Добавление Excel в систему объединения

Порядок действий:

Чтобы добавить источник данных Excel в систему объединения:

1. Зарегистрируйте оболочку с помощью оператора CREATE WRAPPER.
2. Зарегистрируйте сервер с помощью оператора CREATE SERVER.
3. Зарегистрируйте псевдонимы с помощью оператора CREATE NICKNAME для каждой электронной таблицы Excel, к которой требуется доступ.

Эти команды можно запустить из командной строки DB2.

Задачи, связанные с данной темой:

- “Регистрация оболочки Excel” на стр. 75
- “Регистрация сервера для источника данных Excel” на стр. 76
- “Регистрация псевдонимов для источников данных Excel” на стр. 77
- “Добавление файлов с табличной структурой в систему объединения” на стр. 16
- “Добавление Documentum в систему объединения” на стр. 33
- “Добавление BLAST в систему объединения” на стр. 96
- “Добавление XML в систему объединения” на стр. 123

Регистрация оболочки Excel

Эта задача - часть большей задачи *Добавление Excel в систему объединения*.

Чтобы можно было обращаться к источникам данных, необходимо зарегистрировать оболочку. Оболочка - это механизм, посредством которого серверы объединения подключаются к источникам данных и получают от них данные. Оболочки устанавливаются в системе в виде файлов библиотек.

Порядок действий:

Чтобы зарегистрировать оболочку для источников данных Excel, выполните оператор CREATE WRAPPER.

Чтобы создать оболочку Excel для Excel 97 с именем Excel_9x_Wrapper, использующую файл библиотеки db21xls.dll, выполните оператор:

```
CREATE WRAPPER Excel_9x_Wrapper LIBRARY 'db21xls.dll'  
  OPTIONS(DB2_FENCED 'N');
```

Дополнительную информацию об операторе CREATE WRAPPER смотрите в книге *DB2 SQL Reference*.

Следующая задача в этой последовательности задач - *Регистрация сервера для источника данных Excel*.

Задачи, связанные с данной темой:

- “Регистрация сервера для источника данных Excel” на стр. 76

Ссылки, связанные с данной темой:

- “Предварительное требование для оболочки Excel” на стр. 75

Регистрация сервера для источника данных Excel

Эта задача - часть большей задачи *Добавление Excel в систему объединения*. Зарегистрировав оболочку, необходимо зарегистрировать соответствующий сервер.

Порядок действий:

Для регистрации сервера Excel в системе объединения используйте оператор CREATE SERVER.

Например, для создания сервера с именем biochem_lab с именем узла biochem_node1, который регистрируется как сервер для оболочки Excel_2000_Wrapper, созданной оператором CREATE WRAPPER, выполните оператор:

```
CREATE SERVER biochem_lab WRAPPER Excel_2000_Wrapper;
```

Определения аргументов

WRAPPER

Задаёт имя оболочки, зарегистрированной соответствующим оператором CREATE WRAPPER. Это обязательный аргумент.

Дополнительную информацию об операторе CREATE SERVER смотрите в книге *DB2 SQL Reference*.

Следующая задача в этой последовательности задач - *Регистрация псевдонимов для источников данных Excel*.

Задачи, связанные с данной темой:

- “Регистрация сервера для файлов с табличной структурой” на стр. 18
- “Регистрация сервера для источников данных Documentum” на стр. 38
- “Регистрация оболочки Excel” на стр. 75
- “Регистрация псевдонимов для источников данных Excel” на стр. 77
- “Регистрация сервера для источника данных BLAST” на стр. 103
- “Регистрация сервера для источника данных XML” на стр. 125

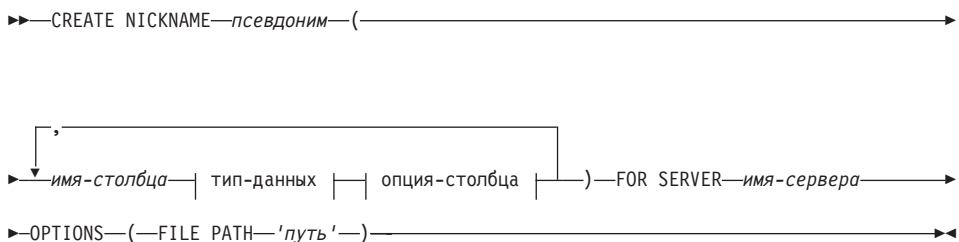
Регистрация псевдонимов для источников данных Excel

Эта задача - часть большей задачи *Добавление Excel в систему объединения*. Зарегистрировав сервер, необходимо зарегистрировать соответствующий псевдоним. Псевдонимы используются в запросах для ссылок на источники данных Excel.

Порядок действий:

Чтобы отобразить источник данных Excel на реляционные таблицы, создайте псевдоним с помощью оператора CREATE NICKNAME.

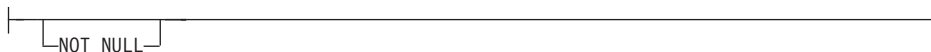
Синтаксис оператора CREATE NICKNAME (для Excel)



тип-данных:



опция-столбца:



Дополнительную информацию об операторе CREATE NICKNAME смотрите в книге *DB2 SQL Reference*.

FOR SERVER

Указывает сервер, зарегистрированной соответствующим оператором CREATE SERVER. Этот сервер используется для обращения к электронной таблице Excel. Задайте имя сервера.

Определения опций

FILE_PATH

Задаёт полный путь каталога и имя файла электронной таблицы Excel, к которой вы хотите обращаться.

Оператор в следующем примере создает псевдоним Compounds (химические вещества) для файла электронной таблицы Excel с именем CompoundMaster.xls. Этот файл содержит три столбца данных, определяемых в системе объединения как Compound_ID, CompoundName и MolWeight (ID соединения, имя соединения и его молекулярный вес).

```

CREATE NICKNAME Compounds (
  Compound_ID INTEGER,
  CompoundName VARCHAR(50),
  MolWeight FLOAT)
FOR SERVER biochem_lab
OPTIONS(FILE_PATH 'C:\My Documents\CompoundMaster.xls');
  
```

Это последняя задача данной последовательности.

Задачи, связанные с данной темой:

- “Регистрация псевдонимов для файлов с табличной структурой” на стр. 19
- “Регистрация псевдонимов для источников данных Documentum” на стр. 41
- “Регистрация сервера для источника данных Excel” на стр. 76
- “Регистрация псевдонимов для источников данных BLAST” на стр. 104
- “Регистрация псевдонимов для источников данных XML” на стр. 126
- Глава 8, “Задание опций стоимости псевдонима” на стр. 143

Выполнение запросов для источников данных Excel

В этом разделе приводятся некоторые примеры запросов к электронным таблицам Excel; в этих примерах используется псевдоним Compounds (химические вещества).

Порядок действий:

Для выполнения запросов нужно использовать в операторах SQL псевдоним и определенные для него столбцы, аналогично тому, как используются имя обычной таблицы и ее столбцы.

Следующий запрос выводит все значения compound_ID для веществ, у которых молекулярный вес (значение MolWeight) больше 2000:

```
SELECT compound_ID
FROM Compounds
WHERE MolWeight > 2000;
```

Следующий запрос выводит все записи, в которых имя соединения или молекулярный вес имеют пустые значения:

```
SELECT *
FROM Compounds
WHERE CompoundName IS NULL
OR MolWeight IS NULL;
```

Следующий запрос выводит все записи, в которых имя соединения содержит строку ase, а молекулярный вес больше или равен 300:

```
SELECT *
FROM Compounds
WHERE CompoundName LIKE 'ase%'
AND MolWeight >=300;
```

Задачи, связанные с данной темой:

- “Выполнение запросов для источников данных Documentum” на стр. 57
- “Пример сценария оболочки Excel” на стр. 79
- “Запуск запросов к источникам данных XML” на стр. 133

Пример сценария оболочки Excel

В этом разделе показан пример реализации оболочки Excel_2000, обращающейся к электронной таблице Excel 2000 в каталоге C:\Data. В этом сценарии регистрируется оболочка, сервер, а также один псевдоним для обращения к электронной таблице. Операторы в этом сценарии вводятся через процессор командной строки DB2. Зарегистрировав оболочку, можно запускать запросы для этой электронной таблицы.

Сценарий начинается с обращения к электронной таблице химических соединений *Compound_Master.xls* с четырьмя столбцами и девятью строками. Полное имя файла этой таблицы - *C:\Data\Compound_Master.xls*. Ее содержимое показано в Табл. 17.

Таблица 17. Пример электронной таблицы *Compound_Master.xls*

	A	B	C	D
1	соединение_A	1,23	367	проверено
2	соединение_G		210	
3	соединение_F	0,000425536	174	проверено
4	соединение_Y	1,00256		проверено
5	соединение_Q		1024	
6	соединение_B	33,5362		
7	соединение_S	0,96723	67	проверено
8				
9	соединение_O	1,2		проверено

Порядок действий:

Для обращения к электронной таблице Excel 2000 при помощи оболочки Excel:

1. Зарегистрируйте оболочку Excel_2000:

```
db2 => CREATE WRAPPER Excel_2000 LIBRARY 'db21sxls.dll'  

OPTIONS(DB2_FENCED 'N')
```
2. Зарегистрируйте сервер:

```
db2 => CREATE SERVER biochem_lab WRAPPER Excel_2000
```
3. Зарегистрируйте псевдоним, ссылающийся на электронную таблицу Excel:

```
db2 => CREATE NICKNAME Compound_Master (compound_name VARCHAR(40),  

weight FLOAT, mol_count INTEGER, was_tested VARCHAR(20))  

FOR biochem_lab  

OPTIONS ( FILE_PATH 'C:\Data\Compound_Master.xls')
```

Процесс регистрации завершен. Источник данных Excel теперь входит в систему объединения и его можно использовать в запросах SQL.

Ниже приводятся примеры запросов SQL к источнику данных Excel и полученных результатов.

- Пример запроса SQL: "Вывести данные по всем соединениям, чей молекулярный вес (*mol_count*) превышает 100"

```
SELECT * FROM compound_master WHERE mol_count > 100
```

Результат: Все поля для строк 1, 2, 3, 5 и 7.

- Пример запроса SQL: "Вывести название соединения (compound_name) и молекулярный вес (mol_count) для всех соединений, чей молекулярный вес еще не определялся.

```
SELECT compound_name, mol_count FROM compound_master
WHERE mol_count IS NULL
```

Результат: Поля compound_name и mol_count строк 4, 6 и 9 электронной таблицы.

- Пример запроса SQL: "Подсчитать число непроверенных соединений с весом (поле weight) больше 1."

```
SELECT count(*) FROM compound_master
WHERE was_tested IS NULL AND weight > 1
```

Результат: Одна запись, представляющая строку 6 электронной таблицы, которая удовлетворяет критериям поиска.

- Пример запроса SQL: "Вывести название соединения (compound_name) и молекулярный вес (mol_count) для всех соединений, чей молекулярный вес определялся и меньше, чем средний молекулярный вес.

```
SELECT compound_name, mol_count
FROM compound_master
WHERE mol_count IS NOT NULL
AND mol_count < (SELECT AVG(mol_count) FROM compound_master
WHERE mol_count IS NOT NULL AND was_tested IS NOT NULL)
```

Подзапрос возвращает среднее значение 368 главному запросу, который возвращает Табл. 18:

Таблица 18. Результаты запроса

COMPOUND_NAME	MOL_COUNT
соединение_A	367
соединение_G	210
соединение_F	174
соединение_S	67

Задачи, связанные с данной темой:

- "Добавление Excel в систему объединения" на стр. 75
- "Выполнение запросов для источников данных Excel" на стр. 79

Ограничения оболочки Excel

- Оболочки Excel доступны только для операционных систем Microsoft Windows, поддерживающих DB2 Universal Database Enterprise Server Edition.
- При использовании оболочек Excel не допускаются сквозные сеансы.
- Данные электронных таблиц Excel можно только читать, но не записывать.
- Оболочка поддерживает для типа данных DATE диапазон дат с 1 января 1970 года по 18 января 2038 года.

Ссылки, связанные с данной темой:

- “Ограничения и особенности оболочки для файлов с табличной структурой” на стр. 23
- “Ограничения и особенности файлов для оболочки для файлов с табличной структурой” на стр. 24
- “Ограничения и особенности оболочки Documentum” на стр. 62
- “Ограничения файлов Excel” на стр. 82
- “Ограничения и особенности оболочки XML” на стр. 134

Ограничения файлов Excel

- В каждом столбце должны быть совместимые типы данных и типы данных столбцов должны быть правильно описаны при регистрации псевдонима.
- Оболочки Excel могут обращаться только к первому листу электронной таблицы в книге Excel.
- Пустые клетки электронных таблиц считаются содержащими пустые значения.
- Электронная таблица может содержать до 10 идущих подряд пустых строк, которые будут включены в набор данных. Более 10 последовательных пустых строк считаются концом набора данных.
- Электронная таблица может содержать пустые столбцы. Однако эти столбцы должны быть зарегистрированы и описаны как правильные поля, даже если они не будут использоваться.

Ссылки, связанные с данной темой:

- “Ограничения оболочки Excel” на стр. 82

Модель управления доступом к файлам для оболочки Excel

Для доступа к файлам Excel СУБД использует полномочия, определяемые свойством LOG ON AS службы базы данных DB2. Это значение можно увидеть на странице свойств LOG ON для экземпляра DB2. Страницу свойств можно открыть из панели управления Службы Windows NT.

Ссылки, связанные с данной темой:

- “Модель управления доступом к файлам для оболочки для файлов с табличной структурой” на стр. 25
- “Управление доступом для оболочки Documentum” на стр. 64

Сообщения оболочки Excel

В этом разделе перечисляются и описываются сообщения, которые могут появиться при работе с оболочкой Excel. Дополнительная информация о сообщениях приведена в *Справочнике по сообщениям DB2*.

Таблица 19. Сообщения, генерируемые оболочкой Excel

Код ошибки	Сообщение	Объяснение
SQL1817N	В операторе CREATE SERVER не идентифицирован "ТИП-ИЛИ-ВЕРСИЯ" источника данных, который вы хотите определить для базы данных объединения.	В операторе CREATE SERVER не был задан параметр VERSION. Исправьте оператор SQL и запустите его снова.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1000.<внутренний программный код>". Соответствующие текст и элементы - "Memory allocation error" (Ошибка выделения памяти)	Обратитесь в службу программной поддержки IBM.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1001.<внутренний программный код>". Соответствующие текст и элементы - "Unknown option" (Неизвестная опция).	Заданная в операторе DDL опция не поддерживается. Исправьте оператор SQL и запустите его снова.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1002.<внутренний программный код>". Соответствующие текст и элементы - "Creation of DELTA object failed" (Не удалось создать дельта-объект).	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.

Таблица 19. Сообщения, генерируемые оболочкой Excel (продолжение)

Код ошибки	Сообщение	Объяснение
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1100.<внутренний программный код>". Соответствующие текст и элементы - "Wrapper options are not supported" (Опции оболочки не поддерживаются).	Эта оболочка не поддерживает OPTIONS для оболочки. Исправьте оператор SQL и запустите его снова.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1200.<внутренний программный код>". Соответствующие текст и элементы - "<опция> - неподдерживаемая опция сервера". (<опция> - неподдерживаемая опция сервера).	Заданная опция не поддерживается этой оболочкой. Исправьте оператор SQL и запустите его снова.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1201.<внутренний программный код>". Соответствующие текст и элементы - (Error obtaining server name) "Ошибка при получении имени сервера".	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1209.<внутренний программный код>". Соответствующие текст и элементы - "Error converting VARCHAR data" (Ошибка преобразования данных VARCHAR).	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1211.<внутренний программный код>". Соответствующие текст и элементы - "Error converting INTEGER data" (Ошибка преобразования данных INTEGER).	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.

Таблица 19. Сообщения, генерируемые оболочкой Excel (продолжение)

Код ошибки	Сообщение	Объяснение
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1212.<внутренний программный код>". Соответствующие текст и элементы - "Error converting FLOAT data" (Ошибка преобразования данных FLOAT).	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1400.<внутренний программный код>". Соответствующие текст и элементы - "<опция> is an unsupported User option" (<опция> - неподдерживаемая опция пользователя).	Заданная опция не поддерживается этой оболочкой. Исправьте оператор SQL и запустите его снова.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1401.<внутренний программный код>". Соответствующие текст и элементы - "Creation of USER Delta object failed" (Не удалось создать пользовательский дельта-объект).	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1500.<внутренний программный код>". Соответствующие текст и элементы - "<опция> is an unsupported Nickname option" (<опция> - неподдерживаемая опция псевдонима).	Заданная опция не поддерживается этой оболочкой. Исправьте оператор SQL и запустите его снова.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1501.<внутренний программный код>". Соответствующие текст и элементы - "Required option PATH not specified" (Не задана обязательная опция PATH).	Для регистрации псевдонима необходима опция PATH. Исправьте оператор SQL и запустите его снова.

Таблица 19. Сообщения, генерируемые оболочкой Excel (продолжение)

Код ошибки	Сообщение	Объяснение
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1502.<внутренний программный код>". Соответствующие текст и элементы - "Creation of NICKNAME Delta object failed" (Не удалось создать дельта-объект псевдонима).	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1503.<внутренний программный код>". Соответствующие текст и элементы - "Error obtaining Nickname column type" (Ошибка при получении типа столбца псевдонима).	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1504.<внутренний программный код>". Соответствующие текст и элементы - "Error obtaining Nickname column type name" (Ошибка при получении имени типа столбца псевдонима).	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1505.<внутренний программный код>". Соответствующие текст и элементы - "received from data source "Excel Wrapper" (От источника данных "Excel Wrapper" получен неизвестный тип данных).	Заданный <тип данных> не поддерживается этой оболочкой. Исправьте оператор SQL и запустите его снова.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1506.<внутренний программный код>". Соответствующие текст и элементы - "Error obtaining Nickname column info" (Ошибка при получении информации о столбце псевдонима).	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.

Таблица 19. Сообщения, генерируемые оболочкой Excel (продолжение)

Код ошибки	Сообщение	Объяснение
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1507.<внутренний программный код>". Соответствующие текст и элементы - "<option> option cannot be dropped" (Невозможно отбросить опцию <опция>).	Заданную опцию нельзя отбросить, так как это обязательная опция.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1508.VANI". Соответствующие текст и элементы - "Column names cannot be altered" (Невозможно изменить имена столбцов).	Оболочка Excel не допускает изменения имен столбцов.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1701.<внутренний программный код>". Соответствующие текст и элементы - "Error parsing SQL" (Ошибка при анализе SQL).	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1702.<внутренний программный код>". Соответствующие текст и элементы - "Error accessing NICKNAME object" (Ошибка при обращении к объекту псевдонима).	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1703.<внутренний программный код>". Соответствующие текст и элементы - "Error building data storage area" (Ошибка при построении области хранения данных).	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.

Таблица 19. Сообщения, генерируемые оболочкой Excel (продолжение)

Код ошибки	Сообщение	Объяснение
SQL1822N	От источника данных "Excel Wgapper" получен неожиданный код ошибки "-1704.<внутренний программный код>". Соответствующие текст и элементы - "Error linking SQL to Nickname Data" (Ошибка связывания SQL с данными псевдонима).	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.
SQL1822N	От источника данных "Excel Wgapper" получен неожиданный код ошибки "-1705.<внутренний программный код>". Соответствующие текст и элементы - "Excel application startup failed" (Ошибка при запуске прикладной программы Excel).	Запуск прикладной программы Excel завершился неудачно. Убедитесь, что Excel установлена в системе и зарегистрирована с нужной версией оболочки. Проверьте свойство LOG ON AS для экземпляра DB2 на панели управления службами Windows NT. Это свойство определяет полномочия, которые будут использоваться для обращения к прикладной программе Excel. Убедитесь, что указанный пользователь имеет нужные права, или измените это свойство, указав учетную запись с нужными полномочиями, затем перезапустите DB2 и снова запустите запрос SQL.
SQL1822N	От источника данных "Excel Wgapper" получен неожиданный код ошибки "-1706.<внутренний программный код>". Соответствующие текст и элементы - "Error opening source spreadsheet" (Ошибка при открытии исходной электронной таблицы).	Ошибка возникла при открытии электронной таблицы, заданной псевдонимом в запросе SQL. Убедитесь, что файл существует по пути PATH, заданном в операторе CREATE NICKNAME во время регистрации.
SQL1822N	От источника данных "Excel Wgapper" получен неожиданный код ошибки "-1707.<внутренний программный код>". Соответствующие текст и элементы - "Error accessing DL output storage area" (Ошибка при обращении к области хранения вывода DL).	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.

Таблица 19. Сообщения, генерируемые оболочкой Excel (продолжение)

Код ошибки	Сообщение	Объяснение
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1708.<внутренний программный код>". Соответствующие текст и элементы - "Excel application end failed" (Ошибка при завершении прикладной программы Excel).	Внутренняя программная ошибка. Если эта ошибка повторится при нескольких запросах, обратитесь в службу программной поддержки IBM.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1711.<внутренний программный код>". Соответствующие текст и элементы - "Error during fetch, possible data/col type mismatch" (Ошибка выборки, возможно несоответствие типов данных или столбцов).	Тип данных в выборке при запросе SQL отличен от типа данных, заданного во время регистрации псевдонима. Исправьте данные в исходной электронной таблице или зарегистрированный тип данных в псевдониме. Если это не поможет исправить ошибку, обратитесь в службу программной поддержки IBM.
SQL1822N	От источника данных "Excel Wrapper" получен неожиданный код ошибки "-1900.<внутренний программный код>". Соответствующие текст и элементы - "Memory allocation error" (Ошибка выделения памяти)	Внутренняя программная ошибка. Обратитесь в службу программной поддержки IBM.

Ссылки, связанные с данной темой:

- “Сообщения для оболочки для файлов с табличной структурой” на стр. 26
- “Сообщения оболочки Documentum” на стр. 64
- “Сообщения оболочки BLAST” на стр. 115
- “Сообщения оболочки XML” на стр. 135

Глава 6. BLAST как источник данных

В этой главе объясняется, что такое система BLAST, как добавить источники данных BLAST в вашу систему объединения, и приводятся сообщения об ошибках, связанные с оболочкой BLAST.

Что такое BLAST?

BLAST (Basic Local Alignment Search Tool - базовый инструментарий поиска локальных блоков) - это утилита, применяемая Национальным центром биотехнологической информации (National Center for Biotechnology Information, NCBI). BLAST используется для просмотра баз данных аминокислотных или нуклеотидных последовательностей для выявления совпадений. Совпадение, найденное BLAST, содержит одну или несколько пар сегментов с высоким показателем совпадения (high-scoring segment pairs, HSP). HSP - это пара фрагментов последовательностей, чей показатель сходства превышает заданное пороговое значение при их оптимальном локальном совмещении. NCBI предлагает специальный исполняемый файл blastall для поиска BLAST в подходящих для этого источниках данных, например, GenBank и SWISS-PROT.

Оболочка BLAST поддерживает все пять типов поиска BLAST: BLASTn, BLASTp, BLASTx, tBLASTn и tBLASTx. Эти типы описываются ниже в Табл. 20.

Таблица 20. Типы поиска BLAST, поддерживаемые оболочкой BLAST

Тип поиска BLAST	Описание
BLASTn	В этом типе поиска BLAST нуклеотидная последовательность сравнивается с содержимым базы данных нуклеотидных последовательностей для поиска областей, гомологичных областям в исходной последовательности.
BLASTp	В этом типе поиска BLAST аминокислотная последовательность сравнивается с содержимым базы данных аминокислотных последовательностей для поиска областей, гомологичных областям в исходной последовательности.

Таблица 20. Типы поиска BLAST, поддерживаемые оболочкой BLAST (продолжение)

Тип поиска BLAST	Описание
BLASTx	В этом типе поиска BLAST нуклеотидная последовательность сравнивается с содержимым базы данных аминокислотных последовательностей для поиска областей, гомологичных областям в исходной последовательности. Последовательность запроса транслируется во всех шести рамках считывания, и поиск в базе данных ведется по каждой из полученных в результате последовательностей.
tBLASTn	В этом типе поиска BLAST аминокислотная последовательность сравнивается с содержимым базы данных нуклеотидных последовательностей для поиска областей, гомологичных областям в исходной последовательности. Последовательности из базы данных транслируются во всех шести рамках считывания, и для полученных последовательностей ведется поиск соответствий с последовательностью запроса.
tBLASTx	В этом типе поиска BLAST нуклеотидная последовательность сравнивается с содержимым базы данных нуклеотидных последовательностей для поиска областей, гомологичных областям в исходной последовательности. При поиске tBLASTx в шести рамках считывания происходит трансляция и последовательности запроса, и последовательностей из базы данных, а поиск гомологичных областей ведется между полученными последовательностями.

На рис. 7 на стр. 93 показано, как BLAST работает с системой объединения.

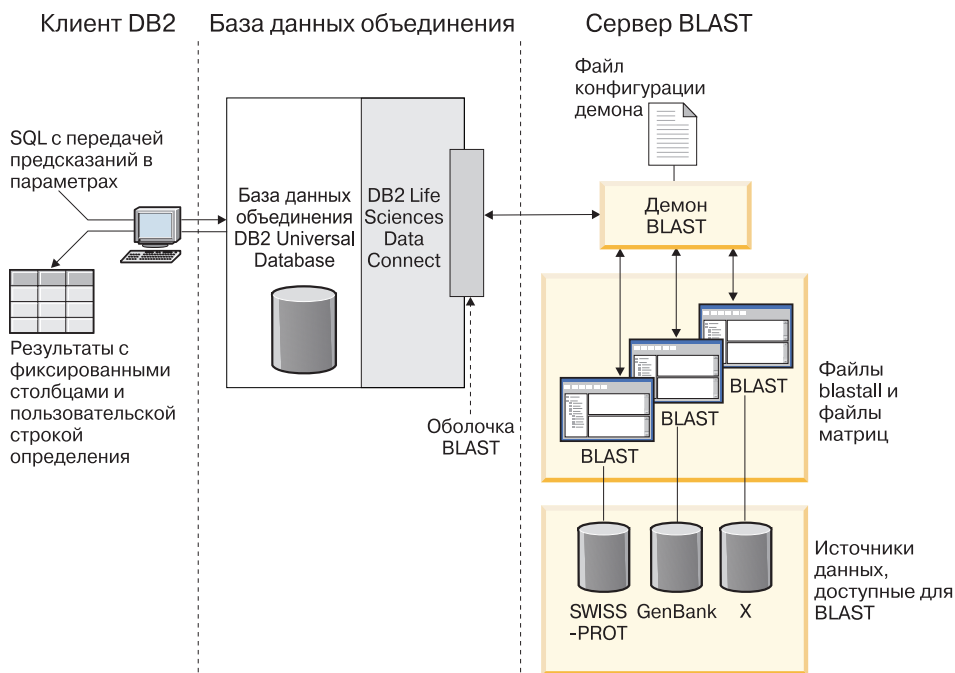


Рисунок 7. Как работает оболочка BLAST

На стороне клиента пользователи и прикладные программы передают на выполнение операторы SQL со специфическими для BLAST предикатами передачи параметров, отображаемыми в стандартные опции BLAST. Операторы SQL со входными предикатами посылаются в систему базы данных объединения DB2® Universal Database, в которой установлена оболочка BLAST.

Оболочка BLAST преобразует запрос в формат, понятный программе BLAST, и посылает преобразованный запрос на ваш сервер BLAST. Этот сервер и сервер системы объединения могут находиться на разных компьютерах. На сервере BLAST запускается специальный демон. Используя информацию своего файла конфигурации, этот демон получает требование на запрос и передает его программе BLAST. После этого программа BLAST запускается для соответствующего источника данных обычным способом.

Полученные результаты возвращаются в программу BLAST и затем в демон. Демон возвращает полученные данные в оболочку BLAST. Оболочка преобразует эти данные в формат реляционной таблицы и возвращает эту таблицу пользователю или прикладной программе. Возвращаемые данные состоят из двух частей:

- Ряд стандартных столбцов, знакомых пользователям BLAST, и
- Пользовательская информация строки определения.

В приведенном ниже примере показано, как извлекается реляционная информация из источников данных, с которыми работает BLAST. Данные перемещаются из формата необработанного файла fasta набора данных для BLAST в реляционную таблицу, которую можно объединить с другими источниками данных в вашей системе объединения.

На рис. 8 показан пример файла fasta, содержащего четыре строки определения и записи нуклеотидных последовательностей.

```
>7:4986 PMON5744
GTTCTTCCCAGTGCCCAAGTCCATTCTGACATCAATGAAGAAGGTAAAAATCCCTGCGTGATCCCTCTGCC
AAGATGTCGAAATCGACCCGGATAAACTAGCTGCTGTCAGAATAACAGACAGCCCGGAGGAGATCGTGC
AGAAGTTCCGCAAGGCTGTGACGGACTTCACCTCGGAGGTCACCTACGACCCGGCCAGGCAGGAGGCGT
GTCCAATTGGTGGCCATCCACGCGCAGTGACCGGACTCCCGTGGAGGAGGTGGTCCGCCGAAGTGCT
GGCATCAACACCCTGGCTACAAGTTGGTGGTGGCGGAGGCTGTGATTGAGAGATTTGCACCAATTAAGA
GTGAAATTGAAAATGAAGAGGAACAAGGACCACCTAGAGAAGGTTTTACAAGTTGGGTCCGCAAAAGC
CAAAGAATTAGCATATCCCGTGTGCCAGGAGGTGAAGAAATGGTGGGGTTTCTATAGGCAGTCTCACCT
AGTCCSAGAAAATGTTTTTATCTTGTGGTCTGCTTGACACTCAGTCTAATAAAGGCAGCTTTCCSTAAG
ACGCCAACAATCCAGTTGGGGATGCTTAGTTACT
>8:9747 PMON5699
AAGAAGTCTTGTAGAACTTCCACCTCCGGCTTCCCCTCCACCTCTCTTACTGTCCCAACCTTCTGAG
ACGCTTTTTCTCTCCCGAGGATTTATCTTTCTCTCTCTCTCTCTCTTTTTTTTTTTTCCCCT
TTTTCCCCCGAGGCTGGTTTTGCTTTGGGGAGGGGGGTTTTTTAAAGGGCCCGGGGGCCCCCTTT
TCCCCCTAATGGGGTAAATTAATAATGGGGGGGGGGTTTTTTTTTTTAAACCCCTATTTGGTCCGG
CCCGGGATTTCCCCCCCCCTTCCCGGTTCCGGGGCCCGAGGAGGGGGGAAAGGGCGGGAA
CCTTTGGTAGTTTCCCCTCGAAAAAATTTTTCGGGGGGAAACCTCCCT
>13:6512 PMON5498
GATAAGAGGCAGAATAGAAGACTGGACTACTTCTCTCTAAAAACACATTTAAACTAAGCCTGAGCAAT
CTCCACCCAAATGGACCGGAAACCTTAAAAAGAATCCTACTCCTGAAGAAAAGAGGAGGACACATCAA
AGGTAGAAAGGGCGATTTTCATGATATAAACAACCCCATACCTCCAGAGTGGGAAGCTCCACAGACTGAA
AACTAAGTGTTCACAGAACTCACCTACAGGAGTGAGCCCCACATCAAACCTCGAATGTGGGGATCTG
GCACTGGTAGAAAGAGCCCTGGAGCATCTGGCATTGAAGGCCAGTGGGCTTGTGTGCAGGAGATCCAC
AGGACTAGGGGAAACGAGACCCCATTTCTTAAAGGTGCACACAGACTTTTACGTGCACTGGTCCCAG
TGCAAGCAAAGTCTCCATAGGAATCTGGGTCAAACCTGACTGCAGTTCTTGGAGACCTCCTGGGAAAG
CAAGGGTGAATGTGGCTTCTGTGGGAAAGACATTGGAAGCAAAGCTCTTGGGAATATTCATCAGTGT
GC
>15:8924 PMON5426
GGAGAACTGACTCCTGAGCAGCTGCAATTCATGCGGCAGGTGCAGCTCGCCAGTGGCAGAAGACGCTG
CCACAGCGCGGACCCGGAACATCGTGACCGGCCTGGGCATCGGGGCGCTGGTGTGGCAATTTGATCC
GTTTGGACTGTAGACTCAGGGAGACCGCATTTAGGGGAACAGGAAGGGCAGCAGGGGCGTGTAGGAGGGC
AGTGTGGGGTGGTAGAAGGAGCCGAGATATGAAAACCTTGGCTCCTTTTAACTCTGAATCAAGCGTTT
GGTGTACCTTACGTTGTCTTTTTAAAGGTGATTTTTAGTATAAATGATTAATGATTACGGAGTCGGGTGA
GGGCTCCAGGAGCAGACGGCAGAAGATCGAATTTGGGAGGATGATCAGCAGCGGTGGTTGAGCAAGTGT
GGGAAAAGGGAATCGCACATTCACGTGGTTTCTTGAACCCACCTCCCAGATGGTTACACCTTCTACT
CGGTGCCAGGAGCGTTTCTTGGATGAGCTGGAGGATGAGGCCAAAGCTGCTC
```

Рисунок 8. Пример файла fasta, nucleo1

Стандартная прикладная программа formatdb преобразует файл fasta в набор данных для BLAST. Теперь, если установлена и зарегистрирована оболочка BLAST, данные можно запрашивать с помощью SQL через систему объединения.

Ниже показано, как запрос от пользователя или прикладной программы на стороне клиента преобразуется оболочкой BLAST. Этот запрос выполняется для набора данных для BLAST.

```
SELECT Unique_ID, Experiment_Number, Organism_Number, HSP_Info, Score
FROM nucleol
WHERE BlastSeq = 'ACATTCTTATAGAGTATTGCTACTCTCCAGGATAGAGTCATCTCT
GGTCTCCAGAGCCACCGCTGGCTACAAGTTGGTGGTGGCGGAGGCTGTGATTGAGAGATTTG
CACCAATACAGAAACTCACSTACAGGAGTGAGCGGGTGGTAGAAGGAGCCCGAGATATGAAA
ACSTTGTTCAGACCCCATTTGTCACCGGGG';
```

Результаты этого запроса преобразуются оболочкой BLAST в формат реляционной таблицы, показанный в Табл. 21.

Таблица 21. BLAST возвращает результаты в виде реляционной таблицы, интегрированной в вашу систему объединения.

Unique ID	Experiment number	Organism number	HSP_INFO	SCORE
PMON5744	4986	7	Identities = 57/201 (28%), Positives = 57/201 (28%), Gaps = 0/201 (0%)	+1.13487000000000E+002
PMON5426	8924	15	Identities = 35/201 (17%), Positives = 35/201 (17%), Gaps = 0/201 (0%)	+6.98754000000000E+001
PMON5498	6512	13	Identities = 26/201 (13%), Positives = 26/201 (13%), Gaps = 0/201 (0%)	+5.20342000000000E+001

Данные находятся в полностью реляционной форме и могут объединяться с данными из других источников данных, используемых в вашей лаборатории. Комбинирование результатов, полученных по нескольким источникам данных, может помочь решить проблемы, неразрешимые без использования системы объединения.

Понятия, связанные с данным:

- “Что такое файлы с табличной структурой?” на стр. 13
- “Что такое Documentum?” на стр. 31
- “Что такое Excel?” на стр. 73

- “Что такое XML?” на стр. 119

Задачи, связанные с данной темой:

- “Добавление BLAST в систему объединения” на стр. 96

Добавление BLAST в систему объединения

Порядок действий:

Чтобы добавить источник данных BLAST на сервер объединения:

1. Убедитесь, что установлена правильная версия выполняемого файла blastall и файлов матриц.
2. Сконфигурируйте демон BLAST.
3. Запустите демон BLAST.
4. Зарегистрируйте оболочку с помощью оператора CREATE WRAPPER.
5. Необязательно: Для повышения производительности запросов задайте переменную среды DB2_DJ_COMM.
6. Зарегистрируйте сервер с помощью оператора CREATE SERVER.
7. Зарегистрируйте псевдонимы с помощью оператора CREATE NICKNAME.

Эти операторы можно выполнить из командной строки DB2. Добавив оболочку BLAST в систему объединения, можно выполнять запросы для источников данных BLAST.

Задачи, связанные с данной темой:

- “Проверка правильности установленной версии выполняемого файла blastall и файлов матриц.” на стр. 97
- “Конфигурирование демона BLAST” на стр. 97
- “Запуск демона BLAST” на стр. 100
- “Регистрация оболочки BLAST” на стр. 101
- “Задание переменной среды DB2_DJ_COMM для оболочки BLAST” на стр. 102
- “Регистрация сервера для источника данных BLAST” на стр. 103
- “Регистрация псевдонимов для источников данных BLAST” на стр. 104
- “Добавление файлов с табличной структурой в систему объединения” на стр. 16
- “Добавление Documentum в систему объединения” на стр. 33
- “Добавление Excel в систему объединения” на стр. 75
- “Добавление XML в систему объединения” на стр. 123

Проверка правильности установленной версии выполняемого файла blastall и файлов матриц.

Эта задача - часть большей задачи *Добавление BLAST в систему объединения*. Проверьте, что на компьютере сервера BLAST установлены сама свежая версия выполняемого файла blastall и файлов матриц BLOSUM62, BLOSUM80, PAM30 и PAM70. Если это не так, нужно установить эти двоичные файлы и файлы матриц. Файлы матриц надо установить в тот же каталог, что и выполняемый файл blastall.

Порядок действий:

Чтобы проверить уровень версии выполняемого файла blastall и файлов матриц:

1. Из командной строки запустите поиск BLAST и в выходном файле посмотрите номер версии.
2. Если у вас нет самой последней версии выполняемого файла blastall и файлов матриц, загрузите их с сайта NCBI: <ftp://ftp.ncbi.nih.gov/blast/executables>.

Следующая задача в этой последовательности задач - *Конфигурирование демона BLAST*.

Задачи, связанные с данной темой:

- “Конфигурирование демона BLAST” на стр. 97

Конфигурирование демона BLAST

Эта задача - часть большей задачи *Добавление BLAST в систему объединения*. Оболочка BLAST требует, чтобы на компьютере на основе UNIX, доступном из системы объединения DB2 Universal Database через TCP/IP, работал демон BLAST. Этот демон работает отдельно от оболочки и системы DB2 Universal Database и ожидает запросов заданий BLAST от оболочки. Выполняемый файл демона, db2blast_daemon, может находиться в любом каталоге на компьютере сервера BLAST.

При установке DB2 Universal Database выполняемый файл демона помещается в каталог /usr/opt/db2_08_01/bin в AIX и в каталог /opt/IBM/db2/V8.1/bin на других платформах Unix на компьютере, на который устанавливается сервер объединения. Если в вашей среде BLAST работает на другом компьютере, нужно скопировать демон в любое место на этом компьютере.

Демон BLAST должен обладать:

- Полномочиями выполнения двоичного файла blastall, чтобы выполнять поиск BLAST.

- Полномочиями доступа к каталогу для записи временных файлов.
- Полномочиями чтения для хотя одного источника данных, для которого может выполняться поиск BLAST. Выполняемый файл blastall должен обладать полномочиями чтения для файла данных и для файлов индекса BLAST, генерируемых программой formatdb.

Демону BLAST требуется файл конфигурации. Пример файла конфигурации демона с именем `BLAST_DAEMON.config` находится в каталоге `DB2PATH/samples/lifesci`, где `DB2PATH` - каталог установки DB2 Universal Database. По умолчанию этот файл конфигурации называется `BLAST_DAEMON.config`.

Скопируйте файл конфигурации в любое место, доступное для демона, переименуйте его, если хотите, и отредактируйте для работы с вашим источником данных. По умолчанию `blast_daemon` ищет свой файл конфигурации в рабочем каталоге, из которого он запущен.

Порядок действий:

Чтобы сконфигурировать демон, задайте в файле конфигурации следующие опции. Для опций, в которых нужно задать путь, можно задавать относительный путь. Относительный путь определяется относительно каталога, из которого был запущен процесс демона.

DAEMON_PORT

Сетевой порт, на котором демон будет ожидать запросов заданий BLAST от оболочки.

MAX_PENDING_REQUESTS

Максимальное число запросов заданий BLAST, объединяемых демоном в блок какой-то момент времени. Это число не отражает число одновременно выполняемых заданий BLAST, а только число заблокированных запросов заданий. Рекомендуется задать значение больше пяти. Демон BLAST не ограничивает число одновременно выполняемых заданий BLAST.

DAEMON_LOGFILE_DIR

Каталог, где демон создаст свой файл журнала. Этот файл будет содержать полезную информацию о состоянии и ошибках, генерируемую демоном BLAST.

Q_SEQ_DIR_PATH

Каталог, где демон создаст временный файл данных последовательности запросов. Этот временный файл очищается после завершения задания BLAST.

BLAST_OUT_DIR_PATH

Каталог, где демон создаст временный файл для хранения выходных

данных BLAST. Данные будут читаться из этого файла и направляться оболочке через сетевое соединение; затем демон очистит этот временный файл.

BLASTALL_PATH

Полное имя выполняемого файла BLAST на компьютере, где работает демон.

запись спецификации базы данных

Задаёт положение доступного для BLAST источника данных. Для нормальной работы демона необходимо, чтобы каждое из имен таких записей спецификации, используемых в файле конфигурации, было задано в опции DATASOURCE оператора CREATE NICKNAME при создании псевдонима для источника данных. Дополнительную информацию об операторе CREATE NICKNAME смотрите в разделе "Регистрация псевдонимов для источников данных BLAST", ссылка на который есть ниже в разделе Задачи, связанные с данной.

Файл конфигурации должен содержать хотя бы одну запись спецификации базы данных в следующем формате:

имя_записи = путь_доступного-для-BLAST-источника-данных

Например, чтобы задать доступный для BLAST источник данных GenBank, добавьте в файл конфигурации демона строку:

```
genbank=/dsk/1/nuc1_data/genbank
```

Путь, задаваемый в записи спецификации базы данных, должен содержать следующие файлы:

- Исходные данные в формате fasta
- Три файла индексов.
 - Для источников данных по нуклеотидам расширения этих индексных файлов:
 - .nhr
 - .nin
 - .nsq
 - Для источников данных по аминокислотам расширения этих индексных файлов:
 - .phr
 - .pin
 - .psq

В записи спецификации базы данных должно быть указано имя файла, содержащего исходные данные в формате fasta. У трех файлов индексов должны быть то же основное имя файла, что и у файла с исходными данными в формате Fasta.

Файл конфигурации должен заканчиваться символом новой строки.

Пример:

Ниже показано содержимое примера файла конфигурации, в котором заданы обязательные опции и спецификации доступных для BLAST источников данных GenBank и SWISS-PROT.

```
DAEMON_PORT=4007
MAX_PENDING_REQUESTS=10
DAEMON_LOGFILE_DIR=./
Q_SEQ_DIR_PATH=./
BLAST_OUT_DIR_PATH=./
BLASTALL_PATH=./blastall
genbank=/dsk/1/nuc1_data/genbank
swissprot=/dsk/1/prot_data/swissprot
```

Следующая задача в этой последовательности задач - *Запуск демона BLAST*.

Задачи, связанные с данной темой:

- “Проверка правильности установленной версии выполняемого файла blastall и файлов матриц.” на стр. 97
- “Запуск демона BLAST” на стр. 100
- “Регистрация псевдонимов для источников данных BLAST” на стр. 104

Запуск демона BLAST

Эта задача - часть большей задачи *Добавление BLAST в систему объединения*. Перед обращением к источнику данных BLAST необходимо запустить демон BLAST.

Предварительные требования:

Перед запуском демона BLAST нужно иметь доступ для записи ко всем путям, перечисленным в записях DAEMON_LOGFILE_DIR, BLAST_OUT_DIR_PATH и Q_SEQ_DIR_PATH в файле конфигурации.

Порядок действий:

Если вы находитесь в каталоге установки демона BLAST, имя файла конфигурации демона не изменено и этот файл находится в том же каталоге, что и выполняемый файл демона, для запуска демона введите в командной строке команду:

```
db2blast_daemon
```


Этот выполняемый файл запустит новый процесс, в котором будет работать демон BLAST.

Если имя файла конфигурации демона изменено или вы находитесь не в каталоге, где находится этот файл, для запуска демона BLAST нужно использовать в команде демона оболочки опцию `-c`, чтобы указать выполняемому файлу демона новое имя или положение файла конфигурации.

Например, следующая команда заставит демон оболочки искать свой файл конфигурации в файле с именем `BLAST_D.config` в подкаталоге `cfg`.

```
db2blast_daemon -c cfg/BLAST_D.config
```

Следующая задача в этой последовательности задач - *Регистрация оболочки BLAST*.

Задачи, связанные с данной темой:

- “Конфигурирование демона BLAST” на стр. 97
- “Регистрация оболочки BLAST” на стр. 101

Регистрация оболочки BLAST

Эта задача - часть большей задачи *Добавление BLAST в систему объединения*. Чтобы можно было обращаться к источникам данных, необходимо зарегистрировать оболочку. Оболочка - это механизм, посредством которого серверы объединения подключаются к источникам данных и получают от них данные. Оболочки устанавливаются в системе в виде файлов библиотек.

Порядок действий:

Для регистрации оболочки BLAST выполните оператор `CREATE WRAPPER`.

Например, чтобы создать в AIX оболочку BLAST с именем `my_blast`, используя файл библиотеки по умолчанию, `libdb2lsblast.a`, выполните оператор:

```
CREATE WRAPPER my_blast LIBRARY 'libdb2lsblast.a'  
  OPTIONS(DB2_FENCED 'N');
```

Список имен файлов библиотек по умолчанию для оболочки BLAST для разных поддерживаемых платформ смотрите в разделе “После установки DB2 Life Sciences Data Connect”, ссылка на который есть ниже в разделе Задачи, связанные с данной. Дополнительную информацию об операторе `CREATE WRAPPER` смотрите в книге *DB2 SQL Reference*.

Следующая задача в этой последовательности задач - *Задание переменной среды DB2_DJ_COMM для оболочки BLAST*.

Задачи, связанные с данной темой:

- “Регистрация оболочки файлов с табличной структурой” на стр. 16
- “Регистрация оболочки Documentum” на стр. 36
- “Регистрация оболочки Excel” на стр. 75
- “Запуск демона BLAST” на стр. 100
- “Задание переменной среды DB2_DJ_COMM для оболочки BLAST” на стр. 102
- “Регистрация оболочки XML” на стр. 124

Задание переменной среды DB2_DJ_COMM для оболочки BLAST

Эта задача - часть большей задачи *Добавление BLAST в систему объединения*. Чтобы улучшить производительность при обращении к источникам данных BLAST, задайте переменную среды DB2_DJ_COMM. Эта переменная определяет, будет ли сервер объединения загружать оболочку при инициализации.

Порядок действий:

Чтобы задать переменную среды DB2_DJ_COMM, выполните команду `db2set`, указав в ней библиотеку оболочки, заданную в соответствующем операторе `CREATE WRAPPER`.

Например:

```
db2set DB2_DJ_COMM='libdb2lsblast.a'
```

Слева и справа от знака равенства (=) не должно быть пробелов.

При загрузке библиотек оболочек при запуске базы данных возникают дополнительные затраты времени. Чтобы избежать этих затрат, просто задайте библиотеки, к которым нужно обращаться.

Дополнительную информацию о переменной DB2_DJ_COMM смотрите в книге *DB2 Administration Guide*.

Следующая задача в этой последовательности задач - *Регистрация сервера для источника данных BLAST*.

Задачи, связанные с данной темой:

- “Задание переменной среды DB2_DJ_COMM для оболочки файлов с табличной структурой” на стр. 17
- “Задание переменной среды DB2_DJ_COMM для оболочки Documentum” на стр. 37

- “Регистрация оболочки BLAST” на стр. 101
- “Регистрация сервера для источника данных BLAST” на стр. 103
- “Задание переменной среды DB2_DJ_COMM для оболочки XML” на стр. 124

Регистрация сервера для источника данных BLAST

Эта задача - часть большей задачи *Добавление BLAST в систему объединения*. Зарегистрировав оболочку, необходимо зарегистрировать соответствующий сервер.

Порядок действий:

Для регистрации сервера BLAST в системе объединения используйте оператор CREATE SERVER.

На каждом компьютере в вашей среде, где установлены выполняемый файл и демон BLAST, нужно зарегистрировать один сервер для каждого типа поиска BLAST, который вы хотите выполнять при помощи этих экземпляров выполняемого файла и демона BLAST.

Например, чтобы зарегистрировать сервер с именем blast_server1, который будет использоваться для поиска BLASTn, для оболочки my_blast, созданной оператором CREATE WRAPPER, выполните оператор:

```
CREATE SERVER blast_server1
  TYPE blastn
  VERSION 2.1.2
  WRAPPER my_blast
  OPTIONS (NODE 'big_rs.company.com', PORT '4007')
```

Аргументы

TYPE Определяет тип поиска BLAST, выполняемый при помощи данного сервера. Это обязательный аргумент. Он должен иметь одно из следующих значений: blastn, blastp, blastx, tblastn, tblastx.

VERSION

Указывает версию используемого сервера. В качестве значения нужно задать используемую версию blastall. Это обязательный аргумент.

WRAPPER

Задаёт имя оболочки, зарегистрированной соответствующим оператором CREATE WRAPPER. Это обязательный аргумент.

Опции

NODE Задаёт имя хоста системы, где работает процесс демона BLAST. Это обязательная опция.

PORT Задаёт номер порта, на котором демон будет ожидать запросов заданий

BLAST. Этот номер порта должен совпадать с номером порта, заданным в опции `daemon_port` в файле конфигурации демона. Порт по умолчанию - 4007. Это необязательная опция.

Дополнительную информацию об операторе `CREATE SERVER` смотрите в книге *DB2 SQL Reference*.

Следующая задача в этой последовательности задач - *Регистрация псевдонимов для источников данных BLAST*.

Задачи, связанные с данной темой:

- “Регистрация сервера для файлов с табличной структурой” на стр. 18
- “Регистрация сервера для источников данных Documentum” на стр. 38
- “Регистрация сервера для источника данных Excel” на стр. 76
- “Задание переменной среды `DB2_DJ_COMM` для оболочки BLAST” на стр. 102
- “Регистрация псевдонимов для источников данных BLAST” на стр. 104
- “Регистрация сервера для источника данных XML” на стр. 125

Регистрация псевдонимов для источников данных BLAST

Эта задача - часть более общей задачи *Добавление BLAST в систему объединения*. После регистрации сервера необходимо зарегистрировать соответствующий псевдоним. Псевдонимы служат для ссылки на источник данных BLAST в запросе.

Порядок действий:

Чтобы зарегистрировать псевдоним BLAST, воспользуйтесь оператором `CREATE NICKNAME`. Поскольку каждый тип поиска BLAST обрабатывается отдельным сервером, для разных типов поиска, которые вы собираетесь запустить для конкретного источника данных BLAST, надо определить свои псевдонимы.

Псевдоним задает информацию столбца для части источника данных, связанной со строкой определения. Все другие столбцы фиксированы. Дополнительную информацию об анализе строки определения смотрите в разделе “Анализ строки определения” на стр. 107. Дополнительную информацию о фиксированных столбцах смотрите в разделе “Фиксированные столбцы” на стр. 107.

Синтаксис оператора `CREATE NICKNAME` для BLAST:

```
► CREATE NICKNAME—псевдоним—(—имя-столбца— информация-столбца —)►
```

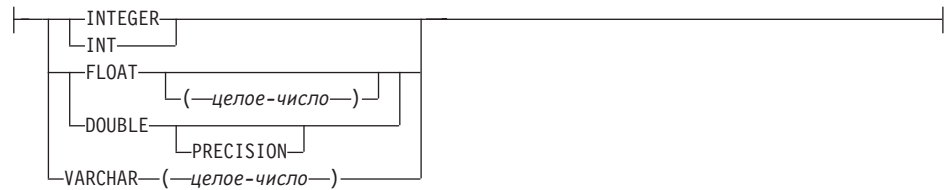
►FOR SERVER—*имя-сервера*—OPTIONS—(—DATASOURCE—'*имя_источника_данных*'—,———►

►TIMEOUT—'*срок_ожидания*'—)———►

информация-столбца:

| тип-данных | | опция-столбца | | опции-столбца-псевдонима | |

тип-данных:



опция-столбца:

| NOT NULL | |

опции-столбца-псевдонима:

►OPTIONS—(—INDEX—'*номер_индекса*'—,—DELIMITER—'*разделитель*'———►
| DEFAULT—'*новое_значение_по_умолчанию*' |)———►

Дополнительную информацию об операторе CREATE NICKNAME смотрите в руководстве *DB2 SQL Reference*.

Опции столбца псевдонима

Значения опции столбца псевдонима должны быть заключены в одинарные кавычки.

INDEX

Порядковый номер столбца, в котором эта опция встречается в группе столбцов строки определения. Это обязательная опция.

Дополнительную информацию об анализе строки определения смотрите в разделе “Анализ строки определения” на стр. 107.

DELIMITER

Символ разделителя, при помощи которого следует задать конечную точку информации строки определения для столбца, где встречается эта опция. Если значение этой опции содержит несколько символов, первое вхождение любого из них указывает на конец данных этого поля. По умолчанию предполагается конец строки. Эта опция обязательна для всех столбцов, кроме последнего заданного столбца, если нужно, чтобы в нем содержался остаток строки определения. Дополнительную информацию об анализе строки определения смотрите в разделе “Анализ строки определения” на стр. 107.

DEFAULT

Задаёт новое значение по умолчанию для следующих входных фиксированных столбцов:

- E_value
- QueryStrands
- GapAlign
- NMisMatchPenalty
- NMatchReward
- Matrix
- FilterSequence
- NumberOfAlignments
- GapCost
- ExtendedGapCost
- WordSize
- ThresholdEx

Это новое значение переопределяет предопределённые значения по умолчанию. Тип нового значения по умолчанию должен быть таким же, как у значения, указанного для данного столбца. Дополнительную информацию о входных фиксированных столбцах смотрите в разделе “Входные фиксированные столбцы” на стр. 107. Это необязательная опция.

Опции псевдонима

Значения опции псевдонима должны быть заключены в одинарные кавычки.

DATASOURCE

Имя источника данных, для которого будет выполняться поиск BLAST. Используемая здесь строка должна в точности повторяться в файле конфигурации демона BLAST. Дополнительную информацию об этом файле конфигурации смотрите ниже в теме “Конфигурирование демона BLAST” раздела Задачи, связанные с данной. Это обязательная опция.

TIMEOUT

Максимальное время в минутах, в течение которого оболочка BLAST ожидает результатов от демона. Значение по умолчанию - 60. Это необязательная опция.

Анализ строки определения

Строка определения подобна ключу для каждой последовательности в источнике данных для BLAST и возвращается как часть каждого совпадения BLAST.

Если требуется включить информацию строки определения в вашу таблицу результатов, надо задать в операторе CREATE NICKNAME столбцы строки определения. В спецификации каждого столбца должна быть задана опция INDEX. Опция DELIMITER должна быть задана для всех столбцов, кроме последнего, если нужно, чтобы в нем содержался остаток строки определения.

Дополнительную информацию об опциях INDEX и DELIMITER смотрите в разделе “Опции столбца псевдонима” на стр. 105.

Поля строки определения должны иметь тип INTEGER, FLOAT, DOUBLE или VARCHAR.

Примечание: Если в поле Accession Number совпадения BLAST будут найдены данные, эти данные вставляются перед данными в поле определения для совпадения BLAST. Полученная в результате строка определения, в которой за данными поля Accession Number следуют данные поля определения, анализируется оболочкой.

Пример задания столбцов строки определения в операторе CREATE NICKNAME смотрите в разделе “Пример CREATE NICKNAME” на стр. 111.

Фиксированные столбцы

Оператор CREATE NICKNAME автоматически создает фиксированные столбцы. Эти столбцы не входят в сам оператор CREATE NICKNAME, но являются частью определения псевдонима, и на них можно ссылаться в запросах SQL. Фиксированные столбцы бывают двух типов - входные и выходные.

Входные фиксированные столбцы

Входные фиксированные столбцы используются в запросах SQL как предикаты передачи параметров. Они передают в BLAST стандартные переключатели BLAST. Затем BLAST запускается для заданного источника данных с использованием этих переключателей. На входные фиксированные столбцы можно также ссылаться в списке SELECT запроса и их можно возвращать как часть таблицы результатов. Эти столбцы перечислены в Табл. 22 на стр. 108.

Таблица 22. Входные фиксированные столбцы

Имя	Тип данных	Описание
BlastSeq	varchar(32000)	Передаёт последовательность запроса в оболочку BLAST.
E_Value	double	Это и входной, и выходной параметр. В качестве входного параметра этот столбец указывает оболочке BLAST верхний предел ожидаемых значений, которые должен вернуть blastall.
QueryStrands	integer	Задаёт, какие цепи должны сравниваться при выполнении поиска BLASTn. Значение 1 указывает, что должна использоваться верхняя цепь, 2 - что должна использоваться нижняя цепь, а 3 - что в сравнении участвуют обе цепи.
GapAlign	char(1)	Указывает оболочке, разрешены ли в выходе BLAST пропуски совмещений.
Matrix	varchar(50)	Определяет, какая матрица замещения используется blastall для определения степени сходства между парами аминокислотных последовательностей. Этот предикат используется только в тех поисках BLAST, где сравниваются две аминокислотные последовательности.
NMisMatchPenalty	integer	Задаёт величину, вычитаемую blastall из значения показателя совмещения при несовпадении одной пары нуклеотидов в гомологичных областях. Этот предикат используется только в тех поисках BLAST, где сравниваются две нуклеотидные последовательности.
NMatchReward	integer	Задаёт величину, добавляемую blastall к значению показателя совмещения для каждой совпадающей пары нуклеотидов в гомологичных областях. Этот предикат используется только в тех поисках BLAST, где сравниваются две нуклеотидные последовательности.
FilterSequence	char(1)	Указывает blastall, отфильтровывать ли из последовательности запроса малоинтересные с биологической точки зрения сегменты. При типе поиска BLASTn используется фильтр DUST. В остальных случаях используется SEG.
NumberOfAlignments	integer	Задаёт количество совмещений HSP, включаемых в выход BLAST.

Таблица 22. Входные фиксированные столбцы (продолжение)

Имя	Тип данных	Описание
GapCost	integer	Задаёт величину, вычитаемую blastall из значения показателя совмещения, если в последовательность запроса или последовательность результата требуется ввести пропуск для увеличения длины совмещаемых участков последовательностей.
ExtendedGapCost	integer	Задаёт величину, вычитаемую blastall из значения показателя совмещения, если пропуск, введенный ранее в последовательность запроса или последовательность результата, нужно увеличить на один нуклеотид или аминокислоту для увеличения длины совмещения.
WordSize	integer	Задаёт blastall исходную длину совпадений при поиске в базе данных.
ThresholdEx	integer	Задаёт порог показателя совпадения, ниже которого BLAST прекращает попытки наращивать длину совмещаемых участков.

Поддерживаемые типы поиска и переключатели BLAST для каждого входного фиксированного столбца перечислены в Табл. 23

Таблица 23. Типы поиска и переключатели BLAST, поддерживаемые входными фиксированными столбцами

Имя	Типы поиска BLAST	Переключ. BLAST	Обяз.?	По умолчанию
BlastSeq	n, p, x, tn, tx	-l	Да	Нет
E_Value	n, p, x, tn, tx	-e	Нет	10
QueryStrands	n	S	Нет	3
GapAlign	n, p, x, tn, tx	-g	Нет	T
Matrix	p, x, tn, tx	-n	Нет	BLOSUM62
NMismatchPenalty	n	-q	Нет	-3
NMatchReward	n	-r	Нет	1
FilterSequence	n, p, x, tn, tx	-F	Нет	T
NumberOfAlignments	n, p, x, tn, tx	-b	Нет	250
GapCost	n, p, x, tn, tx	-G	Нет	11
ExtendedGapCost	n, p, x, tn, tx	-E	Нет	1

Таблица 23. Типы поиска и переключатели BLAST, поддерживаемые входными фиксированными столбцами (продолжение)

Имя	Типы поиска BLAST	Переключ. BLAST	Обяз.?	По умолчанию
WordSize (для Blastn не допускаются значения меньше 7)	n, p, x, tn, tx	-W	Нет	11 -BLASTn 3 -BLASTp
ThresholdEx	n, p, x, tn, tx	-f	Нет	0

Выходные фиксированные столбцы

Выходные фиксированные столбцы возвращаются в таблице результатов запроса и могут использоваться как предикаты. Эти столбцы перечислены в Табл. 24.

Таблица 24. Выходные фиксированные столбцы

Имя	Тип данных	Описание
Score	double	Вычисленный для HSP показатель, представленный в результатах BLAST.
E_value	double	Это и входной, и выходной параметр. В качестве выходного параметра этот столбец содержит вычисленный для HSP показатель, представленный в результатах BLAST.
Длина	integer	Длина совпадающей последовательности в результатах BLAST.
HSP_Info	varchar(100)	Строка информации для данного HSP, выдаваемая BLAST. Эта строка содержит информацию о числе нуклеотидов или аминокислот, совпавших в последовательности запроса и последовательности результата.
HSP_Q_Start	integer	Номер положения первого гомологичного нуклеотида или аминокислоты в последовательности запроса.
HSP_Q_End	integer	Номер положения последнего гомологичного нуклеотида или аминокислоты в последовательности запроса.
HSP_Q_Seq	varchar(32000)	Сегмент последовательности запроса, начинающийся с HSP_Q_Start и заканчивающийся HSP_Q_End.

Таблица 24. Выходные фиксированные столбцы (продолжение)

Имя	Тип данных	Описание
HSP_H_Start	integer	Номер положения первого гомологичного нуклеотида или аминокислоты в последовательности результата.
HSP_H_End	integer	Номер положения последнего гомологичного нуклеотида или аминокислоты в последовательности результата.
HSP_H_Seq	varchar(32000)	Сегмент последовательности результата, начинающийся с HSP_H_Start и заканчивающийся HSP_H_End.
HSP_Midline	varchar(32000)	Строка, выдаваемая BLAST для указания степени гомологии между аминокислотами или нуклеотидами в каждом положении гомологичных областей последовательностей запроса и результата.

Пример CREATE NICKNAME

В приведенном ниже примере оператора CREATE NICKNAME определяется псевдоним genbank.

Пример основан на следующей информации, содержащейся в поле Определение результата BLAST:

>276342 15:8924 PMON5426

где:

276342 Поле Accession результата BLAST.

15:8924 PMON5426

Поле определения результата BLAST, содержащее номер организма, за которым следуют номер эксперимента и уникальный идентификатор.

На основе указанной информации создан следующий псевдоним:

```
CREATE NICKNAME genbank (
  acc_num integer  OPTIONS(INDEX '1', DELIMITER ' '),
  org_num integer  OPTIONS(INDEX '2', DELIMITER ':'),
  exp_num integer  OPTIONS(INDEX '3', DELIMITER ' '),
  u_id  varchar(10)  OPTIONS(INDEX '4'))
FOR SERVER blast_server1
  OPTIONS(DATASOURCE 'genbank', TIMEOUT '300');
```

Столбец `acc_num` должен содержать 276342, столбец `org_num` должен содержать 15, столбец `exp_num` должен содержать 8924, а столбец `u_id` должен содержать PMON5426.

После передачи на выполнение оператора `CREATE NICKNAME` можно запрашивать вашу систему объединения с помощью псевдонима `genbank`. Этот псевдоним можно также объединить с другими псевдонимами и таблицами в вашей системе объединения.

Это последняя задача данной последовательности.

Задачи, связанные с данной темой:

- “Регистрация псевдонимов для файлов с табличной структурой” на стр. 19
- “Регистрация псевдонимов для источников данных Documentum” на стр. 41
- “Регистрация псевдонимов для источников данных Excel” на стр. 77
- “Конфигурирование демона BLAST” на стр. 97
- “Регистрация сервера для источника данных BLAST” на стр. 103
- “Регистрация псевдонимов для источников данных XML” на стр. 126
- Глава 8, “Задание опций стоимости псевдонима” на стр. 143

Построение запросов SQL BLAST

SQL для источников данных BLAST должен содержать только специальные входные предикаты для передачи стандартных переключателей BLAST исполняемому файлу `blastall`.

Ограничения:

Каждый допустимый запрос к оболочке BLAST должен содержать хотя бы входной предикат `BlastSeq`. Все остальные предикаты необязательны.

Порядок действий:

Чтобы построить запрос BLAST, используйте входные предикаты в условии `WHERE` вашего оператора SQL.

В приведенном ниже примере показаны три входных предиката: `BlastSeq`, `GapCost` и `NMismatchPenalty`.

```
Select * from blast b where  
BlastSeq = 'GTCCAGCC...' AND  
GapCost = -10 AND  
NMismatchPenalty = -4;
```

Список типов данных, описаний, переключателей BLAST и типов поиска, поддерживаемых для каждого входного предиката, смотрите ниже в теме "Регистрация псевдонимов для источников данных BLAST" раздела Задачи, связанные с данной.

Задачи, связанные с данной темой:

- "Примеры запросов BLAST" на стр. 113

Примеры запросов BLAST

Ниже приводится несколько примеров запросов BLAST, показывающих, как строятся запросы для источников данных BLAST.

Порядок действий:

Для запуска запросов руководствуйтесь приведенными ниже примерами.

В этих запросах имя, использованное для каждого псевдонима, обозначает тип поиска BLAST и источник данных. Это необходимо, чтобы не перечислять операторы регистрации в каждом примере запроса. Кроме того, чтобы показать в этих примерах поведение оболочки при объединении источников данных, в некоторых запросах упомянуты другие гипотетические источники данных.

Запрос 1

```
select *
from blastn_genbank
where BlastSeq =
  'caaccctccagccgagttgtcaatggcgaggaagctgttccccac';
```

При выполнении этого оператора SQL оболочка начнет поиск BLASTn указанной последовательности в GenBank. Оболочка будет возвращать все доступные столбцы, включая столбцы входных параметров и столбцы результатов BLAST.

Запрос 2

```
select *
from blastn_genbank
where BlastSeq =
  'caaccctccagccgagttgtcaatggcgaggaagctgttccccac'
and GapCost = 8 and NmisMatchPenalty = -4;
```

По выполнении этого оператора SQL оболочка начнет поиск BLASTn указанной последовательности в GenBank. Кроме того, оболочка передаст демону два указанных параметра, который, в свою очередь, передаст их в командную строку blastall. Оболочка будет возвращать все доступные столбцы, включая столбцы входных параметров и столбцы результатов BLAST.

Запрос 3

```
select blp.*
from blastp_swissprot blp, protein_db prdb
where prdb.keyword = 'malic enzyme'
and blp.BlastSeq = prdb.sequence;
```

При выполнении этого оператора SQL оболочка предпримет ноль или более поисков BLASTp в базе данных SWISS-PROT, в зависимости от числа последовательностей, возвращенных из гипотетической базы данных по секвенированным белкам. DB2 разобьет этот оператор на два отдельных запроса, и для каждой цепи, возвращенной из гипотетической базы данных по белкам, будет выполнено по одному поиску BLASTp. Оболочка будет возвращать все доступные столбцы, включая столбцы входных параметров и столбцы результатов BLAST.

Запрос 4

```
select Score, E_Value, HSP_Info, HSP_Q_Seq, HSP_H_Seq, HSP_Midline
from blastx_swissprot
where BlastSeq = 'gagttgtcaatggcgagg'
and GapCost = 8;
```

По выполнении этого оператора SQL оболочка начнет поиск BLASTx указанной последовательности в SWISS-PROT. В этом случае blastall будет транслировать входную последовательность во всех шести рамках считывания и выполнять поиск гомологий в каждом из шести воссозданных фрагментов аминокислотных последовательностей. HSP в результатах будут основаны на совмещении пар аминокислот, а не нуклеотидов. Полученный параметр будет передан демону и затем через командную строку - исполняемому файлу blastall. Оболочка при этом возвратит только столбцы, указанные в запросе.

Запрос 5

```
select tblx.Score, tblx.E_Value, tblx.HSP_Info tblx.HSP_Q_Seq,
HSP_H_Seq, HSP_Midline
from tblastx_genbank tblx, gen_exp_database gedb
where tblx.BlastSeq = gedb.sequence
and gedb.organism = 'interesting organism'
and GapCost = 8
and FilterSequence = 'F';
```

При выполнении этого оператора SQL оболочка предпримет ноль или более поисков tBLASTx в базе данных GenBank, в зависимости от числа последовательностей, возвращенных из гипотетической базы данных по экспрессии генов. DB2 разобьет этот оператор на два отдельных запроса, и для каждой цепи, возвращенной из гипотетической базы данных по экспрессии генов, будет выполнено по одному поиску tBLASTx. В этом случае blastall будет транслировать входную последовательность и все последовательности из GenBank во всех шести рамках считывания и выполнять поиск гомологий в

каждом из шести воссозданных фрагментов аминокислотных последовательностей запроса и воссозданных аминокислотных последовательностях базы данных. HSP в результатах будут основаны на совмещении пар аминокислот, а не нуклеотидов. Полученные параметры будут переданы демону и затем через командную строку - исполняемому файлу blastall. Оболочка при этом возвратит только столбцы, указанные в запросе.

Задачи, связанные с данной темой:

- “Выполнение запросов для источников данных Documentum” на стр. 57
- “Выполнение запросов для источников данных Excel” на стр. 79
- “Запуск запросов к источникам данных XML” на стр. 133

Советы по оптимизации оболочки BLAST

Запуск оболочки и демона на одном сервере может предотвратить возможные узкие места сетевых соединений.

Ссылки, связанные с данной темой:

- “Советы по оптимизации для оболочки для файлов с табличной структурой” на стр. 25

Сообщения оболочки BLAST

В этом разделе перечисляются и описываются сообщения, которые могут появиться при работе с оболочкой BLAST. Дополнительная информация о сообщениях приведена в *Справочнике по сообщениям DB2*.

Таблица 25. Сообщения, генерируемые оболочкой BLAST

Код ошибки	Сообщение	Объяснение
SQL0901N	Оператор SQL завершился неудачно из-за несущественной ошибки системы. Последующие операторы SQL могут обрабатываться. (Reason "sqlno_crule_save_plans [100]:rc (-2144272209) Empty plan list detect".) (Причина "sqlno_crule_save_plans [100]:rc (-2144272209) Обнаружен пустой список планов".)	Запрос SQL, переданный в DB2, нельзя обработать оболочкой. Исправьте синтаксис и повторите обработку оператора.

Таблица 25. Сообщения, генерируемые оболочкой BLAST (продолжение)

Код ошибки	Сообщение	Объяснение
SQL1816N	Модуль оболочки "BLAST_WRAPPER" нельзя использовать для обращения к источнику данных "тип" (" <code><тип сервера></code> " ""), который вы пытались определить для базы данных объединения.	В операторе CREATE SERVER использован недопустимый тип. Надо использовать один из поддерживаемых типов BLAST.
SQL1817N	В операторе CREATE SERVER не идентифицирован "тип-или-версия" источника данных, который вы хотите определить для базы данных объединения.	В операторе CREATE SERVER не задана версия.
SQL1822N	От источника данных "Blast Wrapper" получен неожиданный код ошибки "Unspecified Error" (Неуточненная ошибка). Соответствующие текст и элементы - "Unable to connect to daemon" (Невозможно соединиться с демоном).	Оболочка blast не смогла соединиться с демоном. Возможно, демон не запущен или неправильно сконфигурирован. Возможно, недоступен компьютер, на котором запущен демон.
SQL1822N	От источника данных "Blast Wrapper" получен неожиданный код ошибки "Unspecified Error" (Неуточненная ошибка). Соответствующие текст и элементы - "Blast daemon timeout expired" (Истек срок ожидания демона Blast).	От демона за время, заданное в операторе CREATE NICKNAME, не получены результаты. Увеличьте срок ожидания или проверьте, правильно ли работает демон.
SQL1822N	От источника данных "Blast Wrapper" получен неожиданный код ошибки "Unspecified Error" (Неуточненная ошибка). Соответствующие текст и элементы - "Blast Daemon Failed" (Ошибка демона Blast).	Демон перестал передавать информацию или формат возвращенных результатов неверен.

Таблица 25. Сообщения, генерируемые оболочкой BLAST (продолжение)

Код ошибки	Сообщение	Объяснение
SQL1822N	От источника данных "Blast Wrapper" получен неожиданный код ошибки "Unspecified Error" (Неуточненная ошибка). Соответствующие текст и элементы - "Unkown error from the blast daemon" (Неизвестная ошибка от демона blast).	Оболочка blast получила от демона неизвестный ей код ошибки. Возможно, версия демона несовместима с версией оболочки.
SQL1822N	От источника данных "Blast Wrapper" получен неожиданный код ошибки "Unspecified Error" (Неуточненная ошибка). Соответствующие текст и элементы - "Column rename not allowed" (Переименование столбцов не разрешено).	Вы пытаетесь переименовать один из столбцов при помощи оператора ALTER NICKNAME. Переименование столбцов не разрешено.
SQL1822N	От источника данных "Blast Wrapper" получен неожиданный код ошибки "Unspecified Error" (Неуточненная ошибка). Соответствующие текст и элементы - "XML parser error" (Ошибка анализа XML).	Анализатор Xerces находится в недопустимом состоянии или сигнализировал об исключительной ситуации.
SQL1823N	Для типа данных "<имя типа данных>" от сервера "<имя сервера>" нет отображения типа данных.	Заданный тип данных не поддерживается этим столбцом.
SQL1881N	"DEFAULT" - недопустимая опция "COLUMN" для объекта "<имя-столбца>"	Опция DEFAULT использована для столбца, который ее не поддерживает. У столбцов только для вывода и столбцов определений нет значений по умолчанию.
SQL1882N	Для опции "COLUMN" "DEFAULT" нельзя задать значение "<значение-опции>" для объекта "<имя-столбца>".	Значение, заданное для опции DEFAULT, несовместимо с типом столбца, или его формат неверен.

Ссылки, связанные с данной темой:

- “Сообщения для оболочки для файлов с табличной структурой” на стр. 26
- “Сообщения оболочки Documentum” на стр. 64
- “Сообщения оболочки Excel” на стр. 83

- “Сообщения оболочки XML” на стр. 135

Глава 7. XML как источник данных

В этой главе объясняется, что такое система XML, как добавить источники данных XML в вашу систему объединения, и приводятся сообщения об ошибках, связанные с оболочкой XML.

Что такое XML?

Язык XML (Extensible Markup Language - расширяемый язык разметки) - это универсальный формат для структурированных документов и данных. Файлы XML имеют расширение xml. Как и в HTML, в языке XML для структурирования данных документов используются теги (слова в угловых скобках '<' и '>'). Пример документа XML показан на рис. 9.

```
<doc>
  <customer id='123'>
    <name>...</name>
    <address>...</address>
    ...
    <order>
      <amount>...</amount>
      <date>...</date>
      <item quant='12'>
        <name>...</name>
      </item>
      <item quant='4'>...</item>
      ...
    </order>
    <order>...</order>
    ...
    <payment>
      <number>...</number>
      <date>...</date>
    </payment>
    <payment>...</payment>
    ...
  </customer>
  <customer id='124'>...</customer>
</doc>
```

Рисунок 9. Пример документа XML

Оболочка XML позволяет использовать операторы SQL для запросов внешних документов XML, которые хранятся как файлы. На рис. 10 показано, как XML работает с системой объединения.

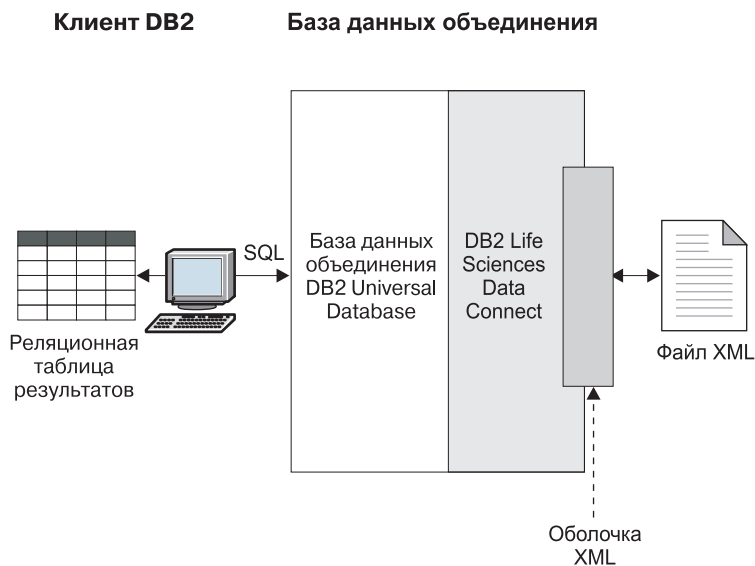


Рисунок 10. Как работает оболочка XML

Оболочка XML позволяет отображать данные XML из внешних источников данных в реляционную схему, состоящую из набора псевдонимов. Структура документа XML логически эквивалентна реляционной схеме, где вложенные и повторяющиеся элементы смоделированы как отдельные таблицы с внешними ключами.

Псевдонимы, соответствующие документу XML, организованы в дерево, где дочерние псевдонимы моделируют элементы, вложенные в элементы, соответствующие родительским псевдонимам.

Есть два основных случая, когда вложенный элемент должен моделироваться как отдельный псевдоним:

- Повторяющиеся элементы
- Элементы, различные по сути, со сложной структурой

Дочерние и родительские псевдонимы связаны первичными/внешними ключами, генерируемыми оболочкой.

Для отображения документа XML в реляционную схему, состоящую из набора псевдонимов, используются выражения XPath. XPath - это механизм адресации

для идентификации частей файла XML, например, групп узлов и атрибутов в дереве документа XML. Основа синтаксиса XPath напоминает адресацию файловой системы.

Каждый псевдоним определяется выражением XPath, идентифицирующим элементы XML, представляющие отдельные кортежи, и набором выражений XPath, задающих, как извлекать значения столбцов из каждого элемента.

Пример:

В приведенном ниже примере показано, как документ XML, изображенный на рис. 9 на стр. 119, отображается в набор псевдонимов, как родительские и дочерние отношения моделируются при помощи первичного и внешнего ключей, как выражения XPath позволяют определить индивидуальные кортежи и столбцы в каждом элементе документа и как можно запустить запрос для документа XML после его регистрации в системе объединения.

Документ XML в этом примере содержит набор элементов `customer` (заказчик) в каждом из которых есть несколько элементов `order` (заказ) и `payment` (платеж).

Элементы `order` включают несколько элементов `item` (пункт).

Отношения между этими элементами показаны на рис. 11.

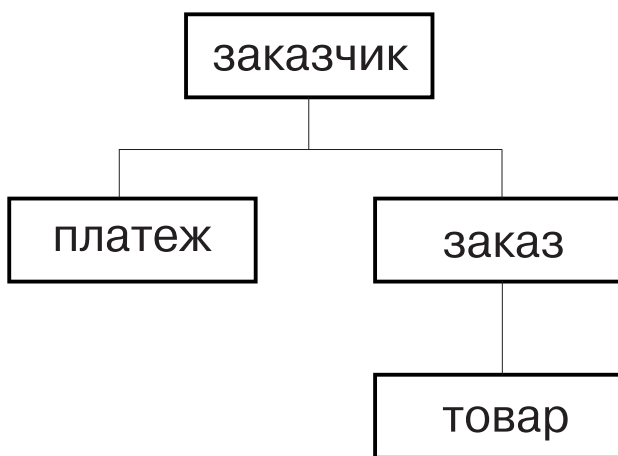


Рисунок 11. Иерархическая структура примера документа XML

Из этой структуры можно при помощи оператора `CREATE NICKNAME` отобразить документ XML в реляционную схему с четырьмя псевдонимами:

- `customers` (заказчики)
- `orders` (заказы)

- payments (платежи)
- items (пункты)

Отношения между псевдонимами определяются заданием каждого псевдонима как родительского или дочернего с помощью специальных опций столбца псевдонима первичного или, соответственно, внешнего ключа. У каждого родительского псевдонима должен быть специальный столбец с назначенной ему опцией столбца первичного ключа. Дочерние псевдонимы определяются специальным столбцом, который ссылается на столбец первичного ключа родительского псевдонима с помощью опции столбца внешнего ключа. Указанные столбцы первичного и внешних псевдонимов не соответствуют данным в вашем документе XML, поскольку в этих столбцах псевдонимов будут содержаться ключи, сгенерированные оболочкой. У псевдонима может быть несколько дочерних псевдонимов, но только один родительский, за исключением корневого псевдонима, у которого нет родительского псевдонима.

В примере документа XML для псевдонима customers определен первичный ключ, а для псевдонимов orders, payments и items определены внешние ключи, указывая на их родительский псевдоним. У псевдонимов orders и payments должны быть внешние ключи, указывающие на псевдоним customers, а у псевдонима items - внешний ключ, указывающий на псевдоним orders.

Для идентификации элементов XML, представляющих отдельные кортежи, создается одно выражение XPath. В этом примере на все элементы customer можно ссылаться при помощи выражения XPath `//customer`, а на все элементы order - при помощи выражения XPath `./order`.

Набор выражений XPath создается, чтобы задать, как извлекать значения каждого столбца из каждого элемента. В этом примере на атрибут id элементов customer, ставший определенным в псевдониме столбцом, можно ссылаться при помощи выражения XPath `./@id`. На элемент name (имя) элементов customer можно ссылаться при помощи выражения XPath `./name`, а на элемент address (адрес) элементов customer - при помощи выражения XPath `./address/@street`.

После того, как документ XML отображен в набор псевдонимов с помощью оператора CREATE NICKNAME так, что каждый псевдоним определен как родительский или дочерний при помощи первичного и внешних ключей, а выражения XPath определяют индивидуальные кортежи и столбцы в каждом элементе документа, для этого документа XML можно запускать запросы SQL.

Более подробную информацию о создании псевдонимов и о синтаксисе оператора CREATE NICKNAME смотрите ниже в теме 'Регистрация псевдонимов для источников данных XML' раздела Задачи, связанные с данной.

Понятия, связанные с данным:

- “Что такое файлы с табличной структурой?” на стр. 13
- “Что такое Documentum?” на стр. 31
- “Что такое Excel?” на стр. 73
- “Что такое BLAST?” на стр. 91

Задачи, связанные с данной темой:

- “Добавление XML в систему объединения” на стр. 123
- “Регистрация псевдонимов для источников данных XML” на стр. 126

Добавление XML в систему объединения

Порядок действий:

Чтобы добавить источник данных XML на сервер объединения:

1. Зарегистрируйте оболочку при помощи оператора CREATE WRAPPER.
2. Необязательно: Задайте переменную среды DB2_DJ_COMM, чтобы повысить производительность запросов.
3. Зарегистрируйте сервер при помощи оператора CREATE SERVER.
4. Зарегистрируйте псевдонимы с помощью оператора CREATE NICKNAME.
5. Создайте производные таблицы для некорневых псевдонимов

Эти операторы можно выполнить из командной строки DB2. Добавив оболочку XML в систему объединения, вы сможете выполнять запросы к источнику данных XML.

Задачи, связанные с данной темой:

- “Регистрация оболочки XML” на стр. 124
- “Задание переменной среды DB2_DJ_COMM для оболочки XML” на стр. 124
- “Регистрация сервера для источника данных XML” на стр. 125
- “Регистрация псевдонимов для источников данных XML” на стр. 126
- “Создание производных таблиц объединения для некорневых псевдонимов (оболочка XML)” на стр. 131
- “Добавление файлов с табличной структурой в систему объединения” на стр. 16
- “Добавление Documentum в систему объединения” на стр. 33
- “Добавление Excel в систему объединения” на стр. 75
- “Добавление BLAST в систему объединения” на стр. 96

Регистрация оболочки XML

Эта задача - часть большей задачи *Добавление XML в систему объединения*. Чтобы можно было обращаться к источникам данных, необходимо зарегистрировать оболочку. Оболочка - это механизм, посредством которого серверы объединения подключаются к источникам данных и получают от них данные. Оболочки устанавливаются в системе в виде файлов библиотек.

Порядок действий:

Чтобы зарегистрировать оболочку XML, выполните оператор CREATE WRAPPER.

Например, чтобы создать оболочку XML в AIX под именем `my_xml` из файла библиотеки по умолчанию `libdb2lxml.a`, введите:

```
CREATE WRAPPER my_xml LIBRARY 'libdb2lxml.a'  
  OPTIONS(DB2_FENCED 'N');
```

Таблица имен файлов библиотеки по умолчанию для оболочки XML на разных поддерживаемых платформах приведена в подразделе 'После установки DB2 Life Sciences Data Connect' в разделе Задачи, связанные с данной. Дополнительную информацию об операторе CREATE WRAPPER смотрите в книге *DB2 SQL Reference*.

Следующая задача в этой последовательности задач - *Задание переменной среды DB2_DJ_COMM для оболочки XML*.

Задачи, связанные с данной темой:

- “Регистрация оболочки файлов с табличной структурой” на стр. 16
- “Регистрация оболочки Documentum” на стр. 36
- “Регистрация оболочки Excel” на стр. 75
- “Регистрация оболочки BLAST” на стр. 101
- “Задание переменной среды DB2_DJ_COMM для оболочки XML” на стр. 124

Задание переменной среды DB2_DJ_COMM для оболочки XML

Эта задача - часть большей задачи *Добавление XML в систему объединения*. Чтобы повысить производительность при обращении к документам XML, задайте переменную среды DB2_DJ_COMM. Эта переменная определяет, будет ли сервер объединения загружать оболочку при инициализации.

Порядок действий:

Чтобы задать переменную среды DB2_DJ_COMM, выполните команду db2set, указав в ней библиотеку оболочки, заданную в соответствующем операторе CREATE WRAPPER.

```
db2set DB2_DJ_COMM='libdb21sxml.a'
```

Слева и справа от знака равенства (=) не должно быть пробелов.

При загрузке библиотек оболочек при запуске базы данных возникают дополнительные затраты времени. Чтобы избежать этих затрат, просто задайте библиотеки, к которым нужно обращаться.

Дополнительную информацию о переменной DB2_DJ_COMM смотрите в книге *DB2 Administration Guide*.

Следующая задача в этой последовательности задач - *Регистрация сервера для источника данных XML*.

Задачи, связанные с данной темой:

- “Задание переменной среды DB2_DJ_COMM для оболочки файлов с табличной структурой” на стр. 17
- “Задание переменной среды DB2_DJ_COMM для оболочки Documentum” на стр. 37
- “Задание переменной среды DB2_DJ_COMM для оболочки BLAST” на стр. 102
- “Регистрация оболочки XML” на стр. 124
- “Регистрация сервера для источника данных XML” на стр. 125

Регистрация сервера для источника данных XML

Эта задача - часть большей задачи *Добавление XML в систему объединения*. После регистрации оболочки надо зарегистрировать соответствующий сервер.

Порядок действий:

Чтобы зарегистрировать сервер XML в системе объединения, используйте оператор CREATE SERVER:

```
CREATE SERVER xml_server WRAPPER my_xml
```

где:

WRAPPER

Оболочка, которую вы зарегистрировали соответствующим оператором CREATE WRAPPER. Это обязательный аргумент.

Примечание: Оболочка XML не использует ключевые слова TYPE и VERSION. При попытке использовать эти ключевые слова в операторе CREATE SERVER возникнет ошибка.

Следующая задача в этой последовательности задач - *Регистрация псевдонимов для источников данных XML*.

Задачи, связанные с данной темой:

- “Регистрация сервера для файлов с табличной структурой” на стр. 18
- “Регистрация сервера для источников данных Documentum” на стр. 38
- “Регистрация сервера для источника данных Excel” на стр. 76
- “Регистрация сервера для источника данных BLAST” на стр. 103
- “Задание переменной среды DB2_DJ_COMM для оболочки XML” на стр. 124
- “Регистрация псевдонимов для источников данных XML” на стр. 126

Регистрация псевдонимов для источников данных XML

Эта задача - часть большей задачи *Добавление XML в систему объединения*. Вам надо создать псевдонимы, моделирующие структуру дерева вашего источника данных XML. Надо создать родительский псевдоним, моделирующий корень дерева. Надо создать псевдонимы-потомки, моделирующие элементы, вложенные в элемент, которому соответствует родительский псевдоним.

Родительские и дочерние псевдонимы связаны первичными и внешними ключами, заданными в операторе CREATE NICKNAME.

Каждый псевдоним определяется выражениями XPath, которые:

- указывают элементы XML, представляющие отдельные кортежи;
- задают, как следует извлекать значения столбцов из каждого элемента.

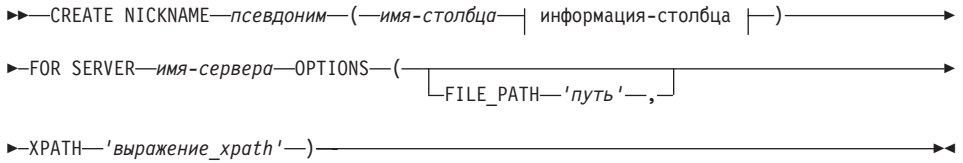
Псевдонимы связаны с вашими документами XML одним из двух способов:

- фиксированно (при помощи опции псевдонима FILE_PATH). При использовании этой опции псевдоним представляет данные определенного документа XML;
- при помощи имени файла, задаваемого во время запроса (при помощи опции столбца псевдонима DOCUMENT). При использовании этой опции псевдоним можно использовать для представления данных из любого документа XML, схема которого соответствует определению псевдонима.

Дополнительная информация по этим опциям приводится ниже в разделе о порядке действий.

Порядок действий:

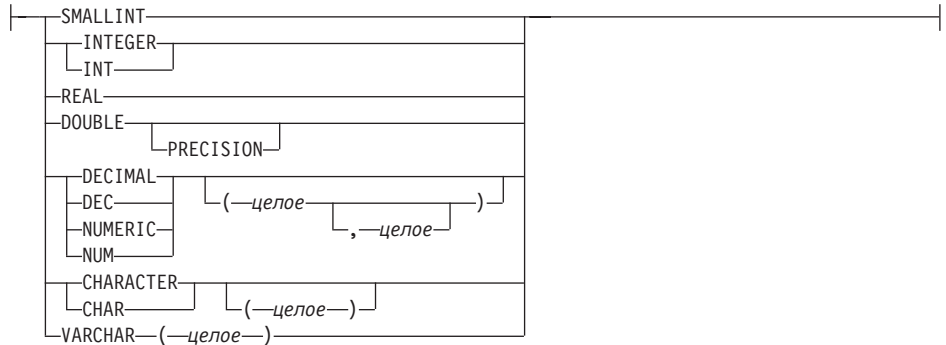
Чтобы отобразить источник данных XML в реляционные таблицы, надо создать псевдонимы при помощи оператора CREATE NICKNAME.



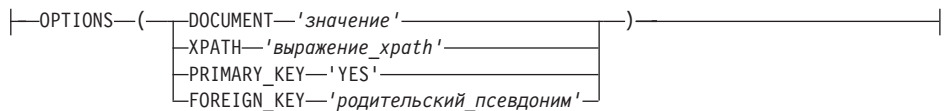
информация-столбца:



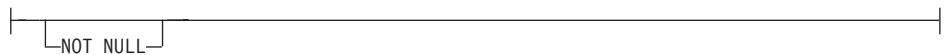
тип-данных:



опции-столбца-псевдонима:



опция-столбца:



Опции псевдонима

FILE_PATH

Задаёт путь файла документа XML. Если задана эту опцию псевдонима, не следует задавать опции столбца псевдонима DOCUMENT. Эта опция допустима только для основного псевдонима (псевдонима, указывающего элементы на верхнем уровне документа XML).

XPATN

Задаёт выражение XPath, которое указывает элементы XML, представляющие отдельные кортежи. Опция псевдонима XPATN для дочернего псевдонима вычисляется в контексте пути, заданного опцией родительского псевдонима XPATN. Это выражение XPath используется как контекст для вычисления значений столбцов, указанных опциями столбцов псевдонима XPATN.

Опции столбца псевдонима

DOCUMENT

Задаёт разновидность данных XML. В настоящее время оболочка XML поддерживает только разновидность FILE. Эта опция допустима только для основного псевдонима (псевдонима, указывающего элементы на верхнем уровне документа XML). Только один столбец можно задать при помощи опции DOCUMENT в одном псевдониме. Столбец, связанный с опцией DOCUMENT, должен иметь тип данных VARCHAR.

Использование опции столбцов псевдонима DOCUMENT вместо опции псевдонима FILE_PATH означает, что соответствующий псевдониму документ будет задан во время выполнения запроса. Если значение опции DOCUMENT - "FILE", во время выполнения запроса будет предоставляться имя содержащего документ файла. Следующий пример CREATE NICKNAME иллюстрирует использование опции столбцов псевдонима DOCUMENT:

```
CREATE NICKNAME customers
(
  doc      VARCHAR(100)  OPTIONS(DOCUMENT 'FILE'),
  name     VARCHAR(16)   OPTIONS(XPATH './name'),
  address  VARCHAR(30)  OPTIONS(XPATH './address/@street'),
  cid      VARCHAR(16)  OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER xml_server
OPTIONS(XPATH '//customer');
```

Теперь, задавая положение документа XML в условии WHERE, можно выполнить следующий запрос, использующий псевдоним customers:

```
SELECT * FROM customers WHERE doc='/home/db2user/Customers.xml';
```

XPATN

Задаёт в документе XML выражение XPath для поиска данных, соответствующих этому столбцу. Это выражение XPath применяется после вычисления выражения XPath, заданного в опции псевдонима XPATN.

PRIMARY_KEY

Указывает, что это родительский псевдоним. Тип данных столбца должен быть VARCHAR(16). У псевдонима может быть не более одной опции столбца PRIMARY_KEY. Единственное допустимое значение - 'YES'. Столбец, заданный в этой опции, содержит ключ,

сгенерированный оболочкой. Значение столбца нельзя получить в списке SELECT, и для этого столбца не должна задаваться опция XPATH. Этот столбец можно использовать только для объединения родительских и дочерних псевдонимов.

FOREIGN_KEY

Указывает, что этот псевдоним - дочерний, и задает имя соответствующего родительского псевдонима. У псевдонима может быть не более одной опции столбца FOREIGN_KEY. Значение этой опции задается с учетом регистра символов. Столбец, заданный в этой опции, содержит ключ, сгенерированный оболочкой. Значение столбца нельзя получить в списке SELECT, и для этого столбца не должна задаваться опция XPATH. Этот столбец можно использовать только для объединения родительских и дочерних псевдонимов.

Примеры псевдонимов

В следующих примерах показана процедура создания псевдонимов для источников данных XML с использованием примера файла XML, содержащийся в примере документа из раздела рис. 12.

```
<doc>
  <customer id='123'>
    <name>...</name>
    <address>...</address>
    ...
    <order>
      <amount>...</amount>
      <date>...</date>
      <item quant='12'>
        <name>...</name>
      </item>
      <item quant='4'>...</item>
      ...
    </order>
    <order>...</order>
    ...
    <payment>
      <number>...</number>
      <date>...</date>
    </payment>
    <payment>...</payment>
    ...
  </customer>
  <customer id='124'>...</customer>
</doc>
```

Рисунок 12. Пример файла XML

Чтобы создать родительский псевдоним customers, введите оператор:

```
CREATE NICKNAME customers
(
  id          VARCHAR(5)   OPTIONS(XPATH './@id'),
  name       VARCHAR(16)  OPTIONS(XPATH './name'),
  address    VARCHAR(30)  OPTIONS(XPATH './address/@street'),
  cid        VARCHAR(16)  OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER xml_server
OPTIONS( XPATH '//customer',
        FILE_PATH '/home/db2user/Customers.xml');
```

Чтобы создать псевдонимы для потомков псевдонима customer – orders, payments и items – введите следующие три оператора псевдонимов:

Для псевдонима orders:

```
CREATE NICKNAME orders
(
  amount  INTEGER          OPTIONS(XPATH './amount'),
  date    VARCHAR(10)      OPTIONS(XPATH './date'),
  oid     VARCHAR(16)      OPTIONS(PRIMARY_KEY 'YES'),
  cid     VARCHAR(16)      OPTIONS(FOREIGN_KEY 'CUSTOMERS'))
FOR SERVER xml_server
OPTIONS( XPATH '//order');
```

Для псевдонима payments:

```
CREATE NICKNAME payments
(
  number  INTEGER          OPTIONS(XPATH './number'),
  date    VARCHAR(10)      OPTIONS(XPATH './date'),
  cid     VARCHAR(16)      OPTIONS(FOREIGN_KEY 'CUSTOMERS'))
FOR SERVER xml_server
OPTIONS( XPATH '//payment');
```

Для псевдонима items:

```
CREATE NICKNAME items
(
  name     VARCHAR(20)     OPTIONS(XPATH './name'),
  quantity INTEGER         OPTIONS(XPATH './@quant'),
  oid     VARCHAR(16)     OPTIONS(FOREIGN_KEY 'ORDERS'))
FOR SERVER xml_server
OPTIONS( XPATH '//item');
```

Следующая задача в этой последовательности задач - *Создание производных таблиц объединения для некорневых псевдонимов (для оболочки XML)*.

Задачи, связанные с данной темой:

- “Регистрация псевдонимов для файлов с табличной структурой” на стр. 19
- “Регистрация псевдонимов для источников данных Documentum” на стр. 41
- “Регистрация псевдонимов для источников данных Excel” на стр. 77

- “Регистрация псевдонимов для источников данных BLAST” на стр. 104
- “Регистрация сервера для источника данных XML” на стр. 125
- Глава 8, “Задание опций стоимости псевдонима” на стр. 143

Создание производных таблиц объединения для некорневых псевдонимов (оболочка XML)

Эта задача - часть большей задачи *Добавление XML в систему объединения*. Рекомендуется определить производные таблицы объединения, охватывающие иерархию псевдонимов, которую описывает документ XML. Определение производных таблиц объединения гарантирует правильное выполнение запросов, объединяющих части иерархии псевдонимов XML, и не включающих корень, а также запросов, объединяющих столбцы помимо особых столбцов PRIMARY_KEY и FOREIGN_KEY.

Порядок действий:

Чтобы определить производные таблицы объединения, включающие все обязательные предикаты и полный путь корня, выполните следующие шаги:

1. Определите производную таблицу для каждого некорневого псевдонима как объединение всех псевдонимов пути корня.
2. В условии WHERE создайте предикаты join со столбцами PRIMARY_KEY и FOREIGN_KEY.
3. В список SELECT включите все столбцы некорневого псевдонима, кроме одного, заданного с опцией столбца псевдонима FOREIGN_KEY.
4. В список SELECT включите столбец родительского псевдонима, определенного с опцией PRIMARY_KEY.

Примеры производных таблиц:

В следующем примере показано использование производных таблиц. В этом примере предполагается, что предварительно были созданы псевдонимы customers, orders, payments и items из примера файла, приведенного в разделе рис. 13 на стр. 132.

```

<doc>
  <customer id='123'>
    <name>...</name>
    <address>...</address>
    ...
    <order>
      <amount>...</amount>
      <date>...</date>
      <item quant='12'>
        <name>...</name>
      </item>
      <item quant='4'>...</item>
      ...
    </order>
    <order>...</order>
    ...
    <payment>
      <number>...</number>
      <date>...</date>
    </payment>
    <payment>...</payment>
    ...
  </customer>
  <customer id='124'>...</customer>
</doc>

```

Рисунок 13. Пример файла XML.

После этого создайте производные таблицы для некорневых псевдонимов `order`, `payment` и `item`:

Для псевдонима `order`:

```

CREATE FEDERATED VIEW order_view AS
SELECT o.amount, o.date, o.oid, c.cid
FROM customers c, orders o
WHERE c.cid = o.cid;

```

Для псевдонима `payment`:

```

CREATE FEDERATED VIEW payment_view AS
SELECT p.amount, p.date, c.cid
FROM customers c, payments p
WHERE c.cid = p.cid;

```

Для псевдонима `item`:

```

CREATE FEDERATED VIEW item_view AS
SELECT it.quantity, it.name, o.oid
FROM customers c, orders o, items i
WHERE c.cid = o.cid AND o.oid = i.oid;

```


Запросы к этим производным таблицам обрабатываются правильно, поскольку присутствует путь объединения для корня.

Например, следующий запрос объединяет суммы заказов и платежи с той же датой:

```
SELECT o.amount, p.amount
FROM order_view o, payment_view p
WHERE p.date = o.date AND
      p.cid = o.cid;
```

Это последняя задача данной последовательности.

Задачи, связанные с данной темой:

- “Регистрация псевдонимов для источников данных XML” на стр. 126

Запуск запросов к источникам данных XML

В этом разделе приведено несколько примеров запросов с использованием псевдонимов `customers`, `orders` и `items`, созданных при помощи оператора `CREATE NICKNAME`.

Порядок действий:

Для запуска запросов используйте как образец следующие примеры.

Запрос, выводящий имена всех клиентов:

```
SELECT name FROM customers;
```

Запрос, выводящий все записи с именем клиента 'Smith':

```
SELECT * FROM customers where name='Smith';
```

Запрос, выводящий сумму и имя клиента для каждого заказа каждого клиента:

```
SELECT c.name, o.amount FROM customers c, orders o where c.cid=o.cid;
```

Объединение `c.cid=o.cid` необходимо, чтобы указать связь потомок-родитель между псевдонимами заказов и клиентов.

Следующий запрос выбирает суммы заказов, имена товаров и адрес клиента для каждого заказа и товара каждого клиента:

```
SELECT c.address, o.amount, i.name FROM customers c, orders o, items i
WHERE c.cid=o.cid AND o.oid=i.oid;
```

И здесь два объединения необходимы для того, чтобы задать отношения потомок-родитель.

Следующие два примера показывают, как написать запросы, использующие псевдоним, который не задает опцию псевдонима FILE_PATH, но задает опцию столбца псевдонима DOCUMENT. Соответствующий оператор CREATE NICKNAME, используемый для создания псевдонима customers:

```
CREATE NICKNAME customers
(
  doc      VARCHAR(100)  OPTIONS(DOCUMENT 'FILE'),
  name     VARCHAR(16)   OPTIONS(XPATH './name'),
  address  VARCHAR(30)   OPTIONS(XPATH './address/@street'),
  cid     VARCHAR(16)   OPTIONS(PRIMARY_KEY 'YES')
FOR SERVER xml_server
OPTIONS(XPATH 'T/customer');
```

Следующий запрос выбирает все данные из файла XML Customers.xml в каталоге /home/db2user/:

```
SELECT * FROM customers WHERE doc='/home/db2user/Customers.xml';
```

Следующий запрос выбирает имена клиентов и даты их заказов, когда заказы превышают сумму 1000, из файла XML Customers.xml в каталоге /home/db2user/:

```
SELECT c.name, o.date FROM customers c, orders o
WHERE c.doc='/home/db2user/Customers.xml' AND o.amount > 1000;
```

Задачи, связанные с данной темой:

- “Выполнение запросов для источников данных Documentum” на стр. 57
- “Выполнение запросов для источников данных Excel” на стр. 79

Ограничения и особенности оболочки XML

В этом разделе содержится перечень ограничений и особенностей, связанных с использованием оболочки XML.

- Не поддерживается возможность сквозных сеансов.
- Документы XML можно только читать.

Ссылки, связанные с данной темой:

- “Ограничения и особенности оболочки для файлов с табличной структурой” на стр. 23
- “Ограничения и особенности файлов для оболочки для файлов с табличной структурой” на стр. 24
- “Ограничения и особенности оболочки Documentum” на стр. 62
- “Ограничения оболочки Excel” на стр. 82
- “Ограничения файлов Excel” на стр. 82

Сообщения оболочки XML

В этом разделе перечисляются и описываются сообщения, которые вы можете получить при работе с оболочкой XML. Дополнительную информацию о сообщениях смотрите в книге *Справочник по сообщениям DB2*.

Таблица 26. Сообщения, генерируемые оболочкой XML

Код ошибки	Сообщение	Объяснение
SQL0405N	Неверный числовой литерал "<имя_столбца>" - значение вне допустимого диапазона.	Заданный числовой литерал выходит за границы допустимого диапазона. Проверьте тип данных столбца в операторе CREATE NICKNAME.
SQL0408N	Значение несовместимо с типом данных назначения. Имя назначения - "<имя_столбца>".	Тип данных значения, назначаемого столбцу, несовместим с объявленным типом данных объекта назначения. Проверьте тип данных столбца в операторе CREATE NICKNAME.
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Error creating wrapper object" - Ошибка создания объекта оболочки.)	Ошибка при создании нового объекта оболочки. Обратитесь в службу поддержки DB2.
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Xerces initialization error" - Ошибка инициализации Xerces.)	Исключительная ситуация при инициализации анализатора Xerces. Обратитесь в службу поддержки DB2.
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина - "<сообщение_об_ошибке_xalan>".)	Ошибка при вызове функции Xalan. Проверьте документ XML. Если структура документа правильна, посмотрите дополнительные сведения об этом сообщении об ошибке в документации Xalan.

Таблица 26. Сообщения, генерируемые оболочкой XML (продолжение)

Код ошибки	Сообщение	Объяснение
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "XalanDOMException: exception code is <код_исключительной_ситуации>" - XalanDOMException: код исключительной ситуации - <код_исключительной_ситуации>.)	Исключительная ситуация XalanDOMException. Значение кода исключительной ситуации смотрите в документации по Xalan.
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Error getting node value" - Ошибка получения значения узла.)	Программа Xalan пыталась обратиться к недопустимому узлу. Обратитесь в службу поддержки DB2.
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Error parsing XML document" - Ошибка анализа документа XML.)	Ошибка при анализе документа XML. Проверьте документ XML.
SQL0901N	Оператор SQL завершился неудачно из-за незначительной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Error getting root element of XML document" - Ошибка получения корневого элемента документа XML.)	После анализа документа XML программа Xalan не смогла получить корневой элемент. Проверьте документ XML.

Таблица 26. Сообщения, генерируемые оболочкой XML (продолжение)

Код ошибки	Сообщение	Объяснение
SQL0901N	Оператор SQL завершился неудачно из-за несущественной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Unspecified exception while evaluating XPath expression" Неуточненная исключительная ситуация при вычислении выражения XPath.)	Неуточненная исключительная ситуация сгенерирована программой Xalan при вычислении выражения XPath. Проверьте документ XML и посмотрите документацию по Xalan.
SQL0901N	Оператор SQL завершился неудачно из-за несущественной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Unspecified exception while getting node value" - Неуточненная исключительная ситуация при получении значения узла.)	Неуточненная исключительная ситуация сгенерирована программой Xalan при получении значения узла. Проверьте документ XML и посмотрите документацию по Xalan.
SQL0901N	Оператор SQL завершился неудачно из-за несущественной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Unspecified exception while building DOM tree" - Неуточненная исключительная ситуация при построении дерева DOM.)	Неуточненная исключительная ситуация сгенерирована программой Xalan при построении дерева DOM для документа XML. Проверьте документ XML и посмотрите документацию по Xalan.
SQL0901N	Оператор SQL завершился неудачно из-за несущественной ошибки системы. Последующие операторы SQL могут обрабатываться. (Причина "Memory allocation error" - Ошибка выделения памяти".)	Ошибка при выделении памяти.

Таблица 26. Сообщения, генерируемые оболочкой XML (продолжение)

Код ошибки	Сообщение	Объяснение
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<точка_трассировки>". Соответствующие текст и элементы - "Column data type not supported" (Недопустимый тип данных столбца).	Недопустимый тип данных столбца псевдонима. Проверьте оператор CREATE NICKNAME.
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<код_трассировки>". Соответствующие текст и элементы - "TYPE clause not supported" (Условие TYPE не поддерживается).	Оператор CREATE SERVER содержит условие TYPE. Это условие не поддерживается оболочкой XML. Удалите его.
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<код_трассировки>". Соответствующие текст и элементы - "VERSION clause not supported" (Условие VERSION не поддерживается).	Оператор CREATE SERVER содержит условие VERSION. Это условие не поддерживается оболочкой XML. Удалите его.
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<код_трассировки>". Соответствующие текст и элементы - "Invalid use of predicate with DOCUMENT column" (Недопустимое условие предиката со столбцом DOCUMENT).	Запрос содержит предикат с некорректными операндами. Проверьте предикаты в запросе.
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<код_трассировки>". Соответствующие текст и элементы - "Invalid use of predicate with FOREIGN_KEY column" (Недопустимое условие предиката со столбцом FOREIGN_KEY).	Запрос содержит предикат с некорректными операндами. Проверьте предикаты в запросе.

Таблица 26. Сообщения, генерируемые оболочкой XML (продолжение)

Код ошибки	Сообщение	Объяснение
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<код_трассировки>". Соответствующие текст и элементы - "Invalid use of predicate with PRIMARY_KEY column" (Недопустимое условие предиката со столбцом PRIMARY_KEY).	Запрос содержит предикат с некорректными операндами. Проверьте предикаты в запросе.
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<код_трассировки>". Соответствующие текст и элементы - "XPATH and DOCUMENT options not compatible" (Опции XPATH и DOCUMENT несовместимы).	Оператор CREATE NICKNAME задан неправильно. Проверьте его синтаксис.
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<код_трассировки>". Соответствующие текст и элементы - "XPATH and FOREIGN_KEY options not compatible" (Опции XPATH и FOREIGN_KEY несовместимы).	Оператор CREATE NICKNAME задан неправильно. Проверьте его синтаксис.
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<код_трассировки>". Соответствующие текст и элементы - "XPATH and PRIMARY_KEY options not compatible" (Опции XPATH и PRIMARY_KEY несовместимы).	Оператор CREATE NICKNAME задан неправильно. Проверьте его синтаксис.

Таблица 26. Сообщения, генерируемые оболочкой XML (продолжение)

Код ошибки	Сообщение	Объяснение
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<код_трассировки>". Соответствующие текст и элементы - "DOCUMENT and FOREIGN_KEY options not compatible" (Опции DOCUMENT и FOREIGN_KEY несовместимы).	Оператор CREATE NICKNAME задан неправильно. Проверьте его синтаксис.
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<код_трассировки>". Соответствующие текст и элементы - "DOCUMENT and PRIMARY_KEY options not compatible" (Опции DOCUMENT и PRIMARY_KEY несовместимы).	Оператор CREATE NICKNAME задан неправильно. Проверьте его синтаксис.
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<код_трассировки>". Соответствующие текст и элементы - "FOREIGN_KEY and PRIMARY_KEY options not compatible" (Опции FOREIGN_KEY и PRIMARY_KEY несовместимы).	Оператор CREATE NICKNAME задан неправильно. Проверьте его синтаксис.
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<код_трассировки>". Соответствующие текст и элементы - "Column option missing" (Отсутствует опция столбца).	Оператор CREATE NICKNAME задан неправильно. Проверьте его синтаксис.

Таблица 26. Сообщения, генерируемые оболочкой XML (продолжение)

Код ошибки	Сообщение	Объяснение
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<код_трассировки>". Соответствующие текст и элементы - "DOCUMENT column option not unique" (Опция столбца DOCUMENT не уникальна).	Оператор CREATE NICKNAME задан неправильно. Проверьте его синтаксис.
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<код_трассировки>". Соответствующие текст и элементы - "FOREIGN_KEY column option not unique" (Опция столбца FOREIGN_KEY не уникальна).	Оператор CREATE NICKNAME задан неправильно. Проверьте его синтаксис.
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<код_трассировки>". Соответствующие текст и элементы - "PRIMARY_KEY column option not unique" (Опция столбца PRIMARY_KEY не уникальна).	Оператор CREATE NICKNAME задан неправильно. Проверьте его синтаксис.
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<код_трассировки>". Соответствующие текст и элементы - "Invalid DOCUMENT option value" (Недопустимое значение опции DOCUMENT).	В операторе CREATE NICKNAME задано недопустимое значение опции DOCUMENT: допускается только 'FILE'. Проверьте оператор CREATE NICKNAME.

Таблица 26. Сообщения, генерируемые оболочкой XML (продолжение)

Код ошибки	Сообщение	Объяснение
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<код_трассировки>". Соответствующие текст и элементы - "Invalid PRIMARY_KEY option value" (Недопустимое значение опции PRIMARY_KEY).	В операторе CREATE NICKNAME задано недопустимое значение опции PRIMARY_KEY: допускается только 'YES'. Проверьте оператор CREATE NICKNAME.
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<код_трассировки>". Соответствующие текст и элементы - "Invalid FOREIGN_KEY option value" (Недопустимое значение опции FOREIGN_KEY).	В операторе CREATE NICKNAME задано недопустимое значение опции FOREIGN_KEY: не найден родительский псевдоним, соответствующий значению опции. Проверьте оператор CREATE NICKNAME.
SQL1822N	От источника данных "XML Wrapper" получен неожиданный код ошибки "<код_трассировки>". Соответствующие текст и элементы - "FILE_PATH and DOCUMENT options not compatible" (Опции FILE_PATH и DOCUMENT несовместимы).	Оператор CREATE NICKNAME задан неправильно: нельзя одновременно задавать опции FILE_PATH и DOCUMENT. Проверьте синтаксис CREATE NICKNAME.
SQL1881N	"<имя_опции>" - недопустимая опция "<тип_опции>" для объекта "<имя_столбца>".	Заданная опция не существует или недопустима для указанного источника данных. Проверьте оператор CREATE NICKNAME.
SQL1883N	"<имя_опции>" - обязательная опция "<тип_опции>" для сервера "<имя_объекта>".	Не задана обязательная опция DB2. Проверьте оператор CREATE NICKNAME.

Ссылки, связанные с данной темой:

- “Сообщения для оболочки для файлов с табличной структурой” на стр. 26
- “Сообщения оболочки Documentum” на стр. 64
- “Сообщения оболочки Excel” на стр. 83
- “Сообщения оболочки BLAST” на стр. 115

Глава 8. Задание опций стоимости псевдонима

Оптимизатор генерирует набор разных планов и для каждого плана оценивает требуемые ресурсы, чтобы создавать эффективные планы выполнения. Для оценки используется план, который требует наименьших ресурсов.

Оценки времени для частей запроса, выполняемых внешними источниками данных, предоставляются оболочкой. При этом используются формулы, учитывающие три основных параметра; эти параметры можно изменить для той или иной установки.

Эти параметры задаются как следующие опции псевдонимов:

RESET_COST

Задаёт в миллисекундах время обращения к внешнему серверу и получения результата.

ADVANCE_COST

Задаёт в миллисекундах время получения строки.

BIND_COST

Задаёт в миллисекундах время передачи параметра от оболочки внешнему источнику.

Все значения должны задаваться как целые. Значения по умолчанию приведены в публикации *DB2 Life Sciences Data Connect Release Notes Version 8*.

Порядок действий:

Чтобы задать опцию псевдонима стоимости:

1. Проанализируйте вашу установку, определив, может ли настройка опций оценки улучшить обработку запросов объединения в условиях вашей компании.
2. Если это так, добавьте одну или несколько опций оценки в оператор оболочки CREATE NICKNAME в качестве опции псевдонима.
3. Выполните оператор CREATE NICKNAME.

Дополнительную информацию об операторе CREATE NICKNAME смотрите в справочнике *DB2 SQL Reference*.

Задачи, связанные с данной темой:

- “Регистрация псевдонимов для файлов с табличной структурой” на стр. 19
- “Регистрация псевдонимов для источников данных Documentum” на стр. 41

- “Регистрация псевдонимов для источников данных Excel” на стр. 77
- “Регистрация псевдонимов для источников данных BLAST” на стр. 104
- “Изменение псевдонимов” на стр. 145
- “Регистрация псевдонимов для источников данных XML” на стр. 126

Глава 9. Изменение псевдонимов

В этой главе объясняется, как использовать оператор ALTER NICKNAME для изменения ранее зарегистрированных псевдонимов.

Изменение псевдонимов

Оператор ALTER NICKNAME можно использовать для изменения представления источника данных или производной таблицы в базе данных объединения.

Ограничения:

Оператор ALTER NICKNAME нельзя использовать для изменения имен столбцов для какой-либо оболочки DB2 Life Sciences Data Connect.

Порядок действий:

Для изменения значений столбцов псевдонима используйте оператор ALTER NICKNAME, чтобы:

- Изменить локальные типы данных для таких столбцов
- Добавить, изменить или удалить опции для таких столбцов

Дополнительную информацию об операторе ALTER NICKNAME смотрите в справочнике *DB2 SQL Reference*.

Задачи, связанные с данной темой:

- “Изменение типа данных” на стр. 145
- “Изменение опции псевдонима” на стр. 146

Изменение типа данных

С помощью оператора ALTER NICKNAME можно изменить тип данных столбца.

Порядок действий:

Чтобы изменить тип данных столбца, воспользуйтесь оператором ALTER NICKNAME.

В приведенном ниже примере оператор ALTER NICKNAME изменяет локальный тип данных столбца DRUG на CHAR(30). Первоначально с помощью оператора CREATE NICKNAME столбец DRUG был определен как CHAR(20). Псевдоним DRUGDATA1 ссылается на локальный файл с табличной структурой drugdata1.txt.

```
ALTER NICKNAME DRUGDATA1
  ALTER COLUMN DRUG
  LOCAL TYPE CHAR(30)
```

Задачи, связанные с данной темой:

- “Изменение псевдонимов” на стр. 145
- “Изменение опции псевдонима” на стр. 146

Изменение опции псевдонима

С помощью оператора ALTER NICKNAME можно изменить опцию псевдонима.

Порядок действий:

Чтобы изменить опцию псевдонима, воспользуйтесь оператором ALTER NICKNAME.

В приведенном ниже примере оператор ALTER NICKNAME изменяет полный путь к файлу с табличной структурой drugdata1.txt. Первоначально путь был задан в операторе CREATE NICKNAME как '/user/pat/drugdata1.txt'. Псевдоним DRUGDATA1 ссылается на локальный файл с табличной структурой drugdata1.txt.

```
ALTER NICKNAME DRUGDATA1
  OPTIONS (SET FILE_PATH '/usr/kelly/data/drugdata1.txt')
```

Задачи, связанные с данной темой:

- “Изменение псевдонимов” на стр. 145
- “Изменение типа данных” на стр. 145

Замечания

IBM может предлагать описанные продукты, услуги и возможности не во всех странах. Сведения о продуктах и услугах, доступных в настоящее время в вашей стране, можно получить в местном представительстве IBM. Любые ссылки на продукты, программы или услуги IBM не означают явным или неявным образом, что можно использовать только продукты, программы или услуги IBM. Разрешается использовать любые функционально эквивалентные продукты, программы или услуги, если при этом не нарушаются права IBM на интеллектуальную собственность. Однако ответственность за оценку и проверку работы любых продуктов, программ и услуг других фирм лежит на пользователе.

Фирма IBM может располагать патентами или рассматриваемыми заявками на патенты, относящимися к предмету данного документа. Получение этого документа не означает предоставления каких-либо лицензий на эти патенты. Запросы по поводу лицензий следует направлять в письменной форме по адресу:

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

По поводу лицензий, связанных с использованием наборов двухбайтных символов (DBCS), обращайтесь в отдел интеллектуальной собственности IBM в вашей стране/регионе или направьте запрос в письменной форме по адресу:

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

Следующий абзац не применяется в Великобритании или в любой другой стране/регионе, где подобные заявления противоречат местным законам: КОРПОРАЦИЯ INTERNATIONAL BUSINESS MACHINES ПРЕДСТАВЛЯЕТ ДАННУЮ ПУБЛИКАЦИЮ “КАК ЕСТЬ” БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, ЯВНЫХ ИЛИ ПОДРАЗУМЕВАЕМЫХ, ВКЛЮЧАЯ ПРЕДПОЛАГАЕМЫЕ ГАРАНТИИ СОВМЕСТИМОСТИ, РЫНОЧНОЙ ПРИГОДНОСТИ И СООТВЕТСТВИЯ ОПРЕДЕЛЕННОЙ ЦЕЛИ, НО НЕ ОГРАНИЧИВАЯСЬ ИМИ. В некоторых странах для определенных сделок подобные оговорки не допускаются; таким образом, это утверждение может не относиться к вам.

Данная информация может содержать технические неточности и типографские опечатки. Периодически в информацию вносятся изменения, они будут включены в новые издания этой публикации. Фирма IBM может в любое время без уведомления вносить изменения и усовершенствования в продукты и программы, описанные в этой публикации.

Любые ссылки в данной информации на Web-сайты, не принадлежащие IBM, приводятся только для удобства и никоим образом не означают поддержки IBM этих Web-сайтов. Материалы этих Web-сайтов не являются частью данного продукта IBM, и вы можете использовать их только на собственную ответственность.

IBM может использовать или распространять присланную вами информацию любым способом, как фирма сочтет нужным, без каких-либо обязательств перед вами.

Если обладателю лицензии на данную программу понадобятся сведения о возможности: (i) обмена данными между независимо разработанными программами и другими программами (включая данную) и (ii) совместного использования таких данных, он может обратиться по адресу:

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

Такая информация может быть предоставлена на определенных условиях (в некоторых случаях к таким условиям может относиться оплата).

Лицензированная программа, описанная в данном документе, и все лицензированные материалы, доступные с ней, предоставляются IBM на условиях IBM Customer Agreement (Соглашения IBM с заказчиком), Международного соглашения о лицензиях на программы IBM или эквивалентного соглашения.

Приведенные данные о производительности измерены в контролируемой среде. Таким образом, результаты, полученные в других операционных средах, могут существенно отличаться от них. Некоторые показатели измерены получены в системах разработки и нет никаких гарантий, что в общедоступных системах эти показатели будут теми же. Более того, некоторые результаты могут быть получены путем экстраполяции. Реальные результаты могут отличаться от них. Пользователи должны проверить данные для своих конкретных сред.

Информация о продуктах других фирм получена от поставщиков этих продуктов, из их опубликованных объявлений или из других общедоступных

источников. Фирма IBM не проверяла эти продукты и не может подтвердить точность измерений, совместимость или прочие утверждения о продуктах других фирм. Вопросы о возможностях продуктов других фирм следует направлять поставщикам этих продуктов.

Все утверждения о будущих планах и намерениях IBM могут быть изменены или отменены без уведомлений, и описывают исключительно цели фирмы.

Эта информация может содержать примеры данных и отчетов, иллюстрирующие типичные деловые операции. Чтобы эти примеры были правдоподобны, в них включены имена лиц, названия компаний и товаров. Все эти имена и названия вымышлены и любое их сходство с реальными именами и адресами полностью случайно.

ЛИЦЕНЗИЯ НА КОПИРОВАНИЕ:

Эта информация может содержать примеры прикладных программ на языках программирования, иллюстрирующих приемы программирования для различных операционных платформ. Разрешается копировать, изменять и распространять эти примеры программ в любой форме без оплаты фирме IBM для целей разработки, использования, сбыта или распространения прикладных программ, соответствующих интерфейсу прикладного программирования операционных платформ, для которых эти примера программ написаны. Эти примеры не были всесторонне проверены во всех возможных условиях. Поэтому IBM не может гарантировать их надежность, пригодность и функционирование.

Каждая копия программ примеров или программ, созданных на их основе, должна содержать следующее замечание об авторских правах:

© (название вашей фирмы) (год). Части этого кода построены на основе примеров программ IBM Corp. © Copyright IBM Corp. *вставьте год или годы*. Все права защищены.

Товарные знаки

Следующие термины, используемые по крайней мере в одном из документов библиотеки документации DB2 UDB, являются товарными знаками корпорации International Business Machines в Соединенных Штатах и/или в других странах.

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

Следующие термины, используемые по крайней мере в одном из документов библиотеки документации DB2 UDB, являются товарными знаками или зарегистрированными товарными знаками других компаний:

Microsoft, Windows, Windows NT и логотип Windows - товарные знаки Microsoft Corporation в Соединенных Штатах и в других странах.

Intel и Pentium - товарные знаки Intel Corporation в Соединенных Штатах и/или других странах.

Java и все товарные знаки на основе Java - товарные знаки Sun Microsystems, Inc. в Соединенных Штатах и/или в других странах.

UNIX - зарегистрированный товарный знак The Open Group в Соединенных Штатах и в других странах.

Названия других компаний, продуктов и услуг могут быть товарными знаками или марками сервиса других фирм.

Библиография

Эта библиография содержит публикации по DB2 Universal Database, которые могут оказаться полезными при работе с DB2 Life Sciences Data Connect.

- *DB2 Connect. Руководство пользователя* (SC09-2954)
- *DB2 for UNIX Quick Beginnings* (GC09-2970)
- *DB2 SQL Reference* (SC09-2974)
- *Руководство администратора DB2: Планирование* (SC09-2946)
- *Руководство администратора DB2: Реализация* (SC09-2944)
- *Руководство администратора DB2: Производительность* (SC09-2945)
- *Справочник по сообщениям DB2* (GC09-2978)
- *IBM DB2 Universal Database Federated Systems Guide* (GC27-1224)
- *DB2 Life Sciences Data Connect Release Notes Version 8*

Индекс

В

BLAST

- добавление в систему объединения
- запуск демона BLAST 100
- оператор CREATE NICKNAME 104
- оператор CREATE SERVER 103
- оператор CREATE WRAPPER 101
- проверка правильности установленного исполняемого файла blastall 97
- проверка правильности установленных файлов матриц 97
- регистрация оболочки 101
- регистрация псевдонимов 104
- регистрация сервера 103
- установка и конфигурирование демона BLAST 97
- файл конфигурации BLAST 97
- описание 91
- сообщения 115

D

DiscoveryLink 2

Documentum

- двойное определение атрибутов с возможностью повторения 61
- добавление в систему объединения компоновка с библиотеками клиента Documentum (только для AIX и операционной среды Solaris) 34
- оператор CREATE FUNCTION 50
- оператор CREATE NICKNAME 41
- оператор CREATE SERVER 38
- оператор CREATE USER MAPPING 40
- оператор CREATE WRAPPER 36
- отображение пользователей 40

Documentum (продолжение)

- добавление в систему объединения (продолжение)
- регистрация оболочки 36
- регистрация пользовательских функций 50
- регистрация псевдонимов 41
- регистрация сервера 38
- указание файла dmcl.ini клиента Documentum 35
- утилита CreateNicknameFile 58
- доступ пользователей к документам 64
- ограничения и особенности 62
- описание 31
- пример 31
- сообщения 64
- утилита CreateNicknameFile 58

I

источники данных по биологии, социологии и медицине 1

O

- оболочка определение 1
- оболочки для биологии, социологии и медицины, по платформам 5
- имена библиотек по умолчанию по платформам 10
- оператор CREATE FEDERATED VIEW XML 131
- оператор CREATE FUNCTION Documentum 50
- оператор CREATE NICKNAME BLAST 104 Documentum 41 XML 126
- файлы Excel 77
- файлы с табличной структурой 19
- оператор CREATE SERVER BLAST 103 Documentum 38 XML 125
- файлы Excel 76

оператор CREATE SERVER (продолжение)

- файлы с табличной структурой 18
- оператор CREATE USER MAPPING Documentum 40
- оператор CREATE WRAPPER BLAST 101 Documentum 36 XML 124
- файлы Excel 75
- файлы с табличной структурой 16
- оптимизация BLAST 115
- файлы с табличной структурой 25
- опции стоимости псевдонима 143

П

- переменная среды DB2_DJ_COMM 17, 37, 102, 124
- примеры запросов 57
- BLAST построение 104, 112
- примеры 113
- Documentum 57
- Excel 79
- XML 133
- псевдонимы задание опций стоимости 143
- изменение 145
- изменение опции псевдонима 146
- изменение типа данных 145

C

- сообщения оболочка BLAST 115
- оболочка Documentum 64
- оболочка Excel 83
- оболочка XML 115
- оболочки для файлов с табличной структурой 26

У

- утилита CreateNicknameFile, Documentum 58
- конфигурирование 59
- отображение типа объектов DM_ID 60

утилита CreateNicknameFile,
Documentum *(продолжение)*
установка 58

Ф

файлы Excel

- добавление в систему объединения
 - оператор CREATE NICKNAME 77
- оператор CREATE SERVER 76
- регистрация оболочки 75
- регистрация псевдонимов 77
- регистрация сервера 76
- модель управления доступом к файлам 82
- ограничения и особенности 82
- описание 73
- пример 73
- пример пользовательского сценария 79
- сообщения 83

файлы с табличной структурой

- добавление в систему объединения
 - регистрация оболочки 16
 - регистрация псевдонимов 19
 - регистрация сервера 18
- доступ из DB2 Life Sciences Data Connect 14
- модель управления доступом к файлам 25
- обзор 13
- ограничения и особенности 23, 24
- оптимизация 25
- пример 13
- сообщения 26
- типы 14

Я

язык XML

- добавление в систему объединения 124
 - оператор CREATE FEDERATED VIEW 131
- оператор CREATE NICKNAME 126
- оператор CREATE SERVER 125
- оператор CREATE WRAPPER 124
- регистрация оболочки 124
- регистрация псевдонимов 126
- регистрация сервера 125

язык XML *(продолжение)*

- добавление в систему объединения *(продолжение)*
 - Создание производных таблиц
 - объединения для некорневых псевдонимов 131
- ограничения и особенности 134
- описание 119
- сообщения 135

Как связаться с IBM

В Соединенных Штатах позвоните по одному из следующих номеров:

- 1-800-237-5511, чтобы обратиться в службу поддержки заказчиков
- 1-888-426-4343, чтобы узнать о доступных формах обслуживания.
- 1-800-IBM-4YOU (426-4968), чтобы обратиться в отдел маркетинга и продаж DB2

В Канаде позвоните по одному из следующих номеров:

- 1-800-IBM-SERV (1-800-426-7378), чтобы обратиться в службу поддержки заказчиков
- 1-800-465-9600, чтобы узнать о доступных формах обслуживания.
- 1-800-IBM-4YOU (1-800-426-4968), чтобы обратиться в отдел маркетинга и продаж DB2

Адрес отделения IBM в вашей стране или регионе можно найти на странице IBM Directory of Worldwide Contacts в Интернете по адресу www.ibm.com/planetwide

Информация о продукте

Информацию о продуктах DB2 Universal Database можно получить по телефону или в Интернете по адресу www.ibm.com/software/data/db2/udb

Этот сайт содержит свежую информацию по технической библиотеке, заказу книг, загружаемые клиенты, группы новостей, пакеты FixPaks, новости и ссылки на ресурсы в Интернете.

Если вы находитесь в США, позвоните по одному из следующих номеров:

- 1-800-IBM-CALL (1-800-426-2255), чтобы заказать продукты или получить общую информацию.
- 1-800-879-2755, чтобы заказать публикации.

Информацию о том, как связаться с IBM из других стран, смотрите на странице IBM Worldwide по адресу www.ibm.com/planetwide



Код изделия: CT16FNA

Напечатано в Дании

GH43-0205-00



(1P) P/N: CT16FNA



Spine information:



IBM® DB2® Life Sciences
Data Connect

Руководство по планированию, установке
и настройке DB2 LSDC

Версия 8