

ФЕДЕРАЛЬНОЕ АГЕНСТВО ПО ОБРАЗОВАНИЮ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П.КОРОЛЕВА»

DB2 UNIVERSAL DATABASE – СРЕДСТВА
СОЗДАНИЯ И УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

Методические указания к лабораторному практикуму

САМАРА
Издательство СГАУ
2006

Составитель *Л.В. Логанова*

УДК 519.6(076)

Db2 universal database – средства создания и управления базами данных : метод. указания / Самар. аэрокосм. ун-т ; сост. Л.В. Логанова. – Самара, 2006. – 36 с.

Методические указания предназначены для студентов, обучающихся по направлению 010600 «Прикладные математика и физика».

Содержат краткие теоретические сведения и задания для лабораторных работ с db2 universal database по курсу «Базы данных и экспертные системы». СУБД DB2 поставляется в нескольких вариантах: персональный сервер (Personal Edition), сервер рабочих групп (Workgroup Edition) и сервер масштаба предприятия (Enterprise Edition). Фирма изготовитель гарантирует полную переносимость приложений, разработанных с использованием персонального сервера, на сервера более высокого уровня. DB2 функционирует под управлением различных операционных систем. Среди них: Windows, Linux, Solaris и др.

Подготовлены на кафедре технической кибернетики.

Печатаются по решению Редакционно-издательского совета Самарского государственного аэрокосмического университета.

Рецензент канд. техн. наук., доц. *Е. И. Ч и г а р и н а*

Работа выполнена при поддержке Министерства образования и науки РФ, а также Американского фонда гражданских исследований и развития (CRDF Project SA-014-02) в рамках Российско-американской программы «Фундаментальные исследования и высшее образование» (BRHE).

Лабораторная работа № 1

Тема: Создание базы данных

Для того чтобы иметь возможность создавать базу данных, необходимо обладать полномочиями SYSADM, т.е. ваше имя должно входить в группу *Администраторы*. Один из способов создания базы данных предполагает применение мастера (Create Database Wisard). В этом случае необходимо выполнить следующую последовательность действий.

Запустить *Центр управления*, если он еще не запущен.

Разворачивать объекты в дереве объектов до тех пор, пока не увидите папку *Базы данных*. В контекстном меню для данной папки выбрать пункт *Создать* (Create). Далее – *Базы данных* при помощи мастера (Database using Wisard).

Заполнить поля первой страницы, учитывая следующие правила:

Имя базы данных (Database name) – набор символов от А до Z, от 0 до 9 и не может начинаться с цифры. Длина не более 8 символов.

Диск по умолчанию (Default Drive) выбирается с учетом свободного места на диске, ввиду того что со временем данные будут разрастаться.

При оценке размера базы данных следует учитывать размер таблиц системного каталога (вначале он составляет 1600 КБ, а затем увеличивается по мере добавления объектов и полномочий в БД), размер таблицы каждого пользователя, пространство индексного указателя, пространство файла журнала и временное рабочее пространство.

Алиас (Alias) – псевдоним, присвоенный базе данных. По умолчанию ему будет присвоено имя БД.

Комментарий (Comment). В этом текстовом поле можно ввести необязательное описание базы данных, длиной до 30 символов.

Щелкнуть на кнопке *Готово* (Finish).

На данный момент БД пуста. И потому на следующем шаге следует создавать таблицы, добавить данные, определить ссылочные ограничения.

Создание таблиц

Таблицы также можно создавать с помощью *Центра управления*, используя для этого мастер (Create Table Wisard). Для этого следует:

Запустить *Центр управления*, если он еще не запущен.

В дереве объектов найти папку нужной БД и щелкнуть на ней.

В контекстном меню папки *Таблицы* (Tables) выбрать пункт *Создать* (Create), чтобы запустить мастер по созданию таблиц.

Приступить к заполнению первого появившегося диалогового окна, учитывая следующее:

При указании *Схемы таблицы* (Table Shema) можно выбрать уже существующую или ввести имя новой схемы. Начальным значением является используемое имя пользователя.

Имя таблицы (Table name) должно быть уникальным в пределах используемой схемы и может содержать до 128 символов. Однако в качестве имени не могут быть выбраны слова: SYSIBM, SYSCAT, SYSFUN, SYSSTAT.

Комментарий (Comment) – необязательное текстовое поле длиной до 30 символов, содержащее описание таблицы.

На следующей странице мастера *Столбцы* (Columns) можно добавить заранее заданные столбцы (Add Predefined) или создать новые, щелкнув на кнопке *Добавить* (Add). При этом

Имя столбца (Column Name) не превышает 30 символов.

Тип данных (Data Type) может быть выбран из широкого множества типов данных, используемых DB2, а может представлять тип данных, созданных вами (см. Приложение).

Можно задать необязательное значение, используемое по умолчанию.

Допускаются пустые значения. Флажок устанавливается в том случае, если данное поле может быть пустым. По умолчанию упомянутый флажок установлен.

По завершении щелкните на кнопке ОК. (Определение характеристик каждого столбца требует тщательного планирования. Большинство столбцов трудно изменить по завершении создания таблицы. Обычно для этого нужно удалить таблицу и создать ее заново.)

На странице мастера *Табличное пространство* (Table Spase) можно создавать табличные пространства для хранения данных таблицы.

На странице *Ключи* (Key) можно на основе списка столбцов, существующих в таблице, выбором соответствующих кнопок определить первичный, уникальный и внешний ключи.

На странице *Измерения* (Dimensions) мастера можно настроить кластеризацию данных, что может повлиять на скорость выполнения запросов.

На странице *Ограничения* (Constraints) можно задать ограничения проверки или бизнес-правила, определяющие значения для каждой из строк таблицы. Ограничения контролируются при каждой вставке или обновлении данных.

На странице *Сводка* (Summary) мастера отображается сводка всех параметров, выбранных во время использования мастера по созданию таблиц.

Добавление данных в таблицу

В среде DB2 данные в таблицы можно добавлять несколькими способами. В большинстве случаев существует приложение, из которого можно вводить данные в базу данных. Это могут быть приложения, написанные с использованием таких полярных языков программирования, как, например, C++, Visual Basic, Java.

При отсутствии приложения данные можно ввести, используя *Командный центр* и простые SQL операторы.

```
Insert into <название табл.> (<наименование поля1>
[,<наименование поля2>,<наименование поля3 >]) values (<Значение
поля1> [,<Значение поля2>,<Значение поля3>]) [ (<Значение
поля1> [,<Значение поля2>,<Значение поля3>]),(<Значение по-
ля1> [,<Значение поля2>,<Значение поля3>]);
```

Для просмотра введенных данных можно выполнить инструкцию Select:

```
Select * from <название табл.>
```

Операторы Insert и Select могут быть составлены с помощью Ассистента SQL (Assis SQL).

Задание на лабораторную работу

1. Создать базу данных Base, хранящую информацию о результатах сессий студентов одной группы.
2. Создайте таблицу с именем Stud, в которой хранятся сведения о студентах:

номер зачетной книжки	Num
фамилия, имя, отчество	Fio
год поступления в университет	God
дата рождения	Dr
3. Вид обучения (платное или бесплатное, по умолчанию бесплатное) Plata
средний бал при поступлении Srb

стоимость обучения в платной группе
адрес

Money
Adr

4. Выберите в качестве первичного ключа столбец *Num*.
5. Заполнить 4-5 записями созданную таблицу (без и с использованием *Ассистента SQL*).
6. Добавить, удалить записи таблицы.

Лабораторная работа № 2

Тема: Индексирование и связывание таблиц

Индексы могут существенно повысить производительность, уменьшая необходимость просмотров таблиц и количество столбцов, выбираемых по данному запросу. Лучше всего работают индексы, значения которых уникальны (при большом числе отношений), и расположенные на быстрых дисковых устройствах. Создать индекс можно с помощью мастера индексов.

Находясь в *Центре управления*, в контекстном меню папки *Индексы* (Indexes) выбрать пункт *Создать* (Create). При заполнении открывшегося диалогового окна учитывать следующее:

1. *Уникальный* (Unique). Создает уникальный индекс. Причем можно повысить производительность запросов, устранив необходимость сортировки.
2. *Кластер* (Cluster). Выбор данного параметра позволяет улучшить производительность при вставке данных, группируя вновь вставляемые строки в соответствии с индексом. Уменьшается расщепление страниц и необходимость реорганизации данных.
3. *Разрешить обратный просмотр* (Allow Reverse Scans). Индексы могут быть определены в порядке возрастания или

убывания. Этот параметр позволяет разрешить просмотр индекса в направлении, обратном заданному при определении индекса; в противном случае при запросе данных в обратном порядке индекс не используется.

4. *Процент свободного места, необходимый на страницах индекса (Percentage of Free Spase to Be left on Index Pages).* Позволяет задать значение параметра PCTFREE в виде процента свободного места, которое должно остаться на индексной странице, чтобы можно было в дальнейшем вставлять данные, что может повысить производительность приложений, требующих вставки данных.
5. *Процент минимального количества используемого места, оставляемого на страницах индекса (Percentage of Minimum Amount of Used Spase to Be Left on Indexes Pages).* Этот параметр задает значение параметра MINPCTUSED в виде процента от размера индексной страницы. При достижении этого предельного значения DB2 автоматически определяет, можно ли объединить две смежные концевые страницы. Это процесс называется оперативной дефрагментацией индекса (online index defragmentation).

Во время создания индексов следует воспользоваться некоторыми рекомендациями, а именно:

- Старайтесь ограничить количество столбцов, по которым будет создаваться индекс, поскольку это дополнительные затраты для приложений, изменяющих данные.
- Выбирайте столбцы, по которым часто производится объединение таблиц.
- Для таблиц, определенных со ссылочной целостностью, создание индексов по внешним ключам улучшает производительность операций удаления и изменения данных для родительской таблицы.

- Если индекс создан на нескольких столбцах, определяющих частичный индекс, задание подмножества существующего индекса обычно не улучшает производительность.

Задание на лабораторную работу

1. Создать таблицу *Session*, хранящую результаты сессий, включающую поля: номер зачетной книжки, номер сессии и оценки по трем предметам.
2. В качестве внешнего ключа поле *Num*.
3. Проиндексировать таблицу *Session* по номеру зачетной книжки и № сессии, определив тип уникальный.
4. Заполнить 4-5 записями созданные таблицы .

Лабораторная работа № 3

Тема: Создание и использование запросов к базе данных и представлений

Естественным языком в рамках технологий реляционных баз данных является язык структурированных запросов (Structured Query Language – SQL). Язык SQL состоит из трех основных компонентов:

- Язык манипулирования данными (Data Manipulation Language – DML). Этот компонент используется для выбора, изменения и удаления объектов из базы данных.
- Язык определения данных (Data Definition language – DDL). Применяется для создания, изменения и удаления объектов базы данных.
- Язык управления данными (Data Control Language – DCL). Используется для предоставления или отзыва прав доступа к базам данных.

К числу основных операторов, которые используются для извлечения данных из БД относится оператор Select. Он запрашивает данные из одной или нескольких таблиц. Одним из способов составления данной инструкции является обращение к возможностям, открывающимся при использовании SQL Assist из командного центра. Эта возможность позволяет составлять SQL-запросы, осуществляя предварительную проверку синтаксиса данного оператора, и лишь после этого посылать запрос на выполнение.

Оператор Select можно использовать для создания отчетов, содержащих информацию из двух и более таблиц. Обычно эту операцию называют соединением (join).

Чтобы соединить две таблицы в выражении Select потребуется указать столбцы, которые должны быть отображены, в выражении From задать имена таблиц, а в выражении Where – условие поиска.

Функции столбцов

Функции столбцов (column functions) действуют в отношении набора значений столбца с целью получения единого результирующего значения. В качестве функций могут применяться, например, следующие: avg, count, max, min.

Строки можно организовывать в соответствии со структурой группирования, определенной в конструкции GROUP BY. Функции столбцов возвращают значение для каждой группы, определенной в GROUP BY. В результирующем наборе каждая группа представляется одной строкой. К определению групп можно добавить уточняющее условие, чтобы система DB2 возвращала результат только для тех групп, которые удовлетворяют условию. Для этого используется конструкция Having после конструкции GROUP BY. Она может содержать один или более предикатов, связанных операциями AND и OR. Предикаты могут быть использованы как в конструкции Having, так и в предложении WHERE.

Представления

Для ограничения доступа к данным большую пользу приносит использование представлений. *Представление* – это поименован-

ная динамически поддерживаемая выборка из одной или нескольких таблиц. Формирование представления происходит каждый раз при обращении к нему. Представления (VIEW), или в контексте системы DB2 производная таблица, позволяет просматривать лишь часть данных, которая доступна и необходима для работы какой-либо группе.

Для создания новой производной таблицы (VIEW) следует воспользоваться *Центром управления* и выполнить следующие действия:

1. В контекстном меню папки *Производные таблицы* выбрать пункт *Создать* (Create).
2. И в открывшемся диалоговом окне ввести имя представления, длиной не более 18 символов и уникальное в рамках используемой схемы; SQL-оператор, определяющий производную таблицу.
3. В разделе *Опции проверки* (Check Options) укажите соответствие между ограничениями и определением производной таблицы.
 - 3.1. Если определение представления не используется при проверке операций вставки и обновления, необходимо использовать переключатель *Нет* (None), установленный по умолчанию.
 - 3.2. Если ограничение новой производной таблицы наследует условия поиска в производной таблице, доступной для обновления, от которой зависит новая производная таблица, выберите переключатель *Каскадно* (Cascaded).
 - 3.3. Выберите переключатель *Локально* (Local) для указания того, что используемое для создания представления условие поиска используется в качестве ограничения для операций вставки и обновления.

4. Чтобы увидеть пример содержимого представления, в контекстном меню данного представления выберите пункт *Пример содержимого (Sample Contents)*.
5. Производная таблица представляет средства управления доступом или расширения полномочий работы с таблицей, обеспечивая доступ только к определенным столбцам таблицы или только к определенному поднабору ее строк.

Задание на лабораторную работу

Создать следующие запросы к базе данных «Результаты сессии студентов», использовать различные направления вывода результатов запроса:

1. *Вывести сведения о студентах, год поступления которых в институт равен заданному.*
 - 1.1. *Вывести фамилии студентов и размер оплаты за обучение для студентов, обучающихся платно.*
 - 1.2. *Вывести сведения о студентах, обучающихся платно и средний балл при поступлении которых меньше заданного.*
 - 1.3. *Вывести фамилии студентов в алфавитном порядке, указанной даты рождения.*
 - 1.4. *Вывести год и количество студентов, поступивших в каждом году.*
 - 1.5. *Вывести фамилии студентов, сдавших сессию указанного номера с оценками не ниже 4 по всем предметам.*
 - 1.6. *Создать запрос с информацией о студентах, имеющих хотя бы одну неудовлетворительную оценку, используя тип объединения внутренних.*

2. Создать представления для базы данных «Результаты сессии студентов».
 - 2.1. Сведения о студентах, сдавших последнюю сессию на «отлично».
 - 2.2. Фамилии студентов, поступивших в институт в указанном году и обучающихся в платной группе.
 - 2.3. Сведения о студентах, добавить нового студента.
 - 2.4. Создать представление с информацией о студентах, не сдававших сессию ни разу.

Лабораторная работа № 4

Тема: Настройка экземпляров DB2

Экземпляр (instance) – это логическая среда баз данных, которая может быть настроена на конкретную группу баз данных, содержащихся внутри экземпляра. Например, стало распространенной практикой создавать один экземпляр для тестовых баз данных и другой для производственных, т.е. окончательных баз данных.

В среде своей системы можно создавать один или большее число экземпляров DB2. Программные файлы физически хранятся в одном месте на конкретном компьютере. Каждый созданный экземпляр ссылается на это место, потому нет необходимости в дублировании программных файлов для каждого нового экземпляра. Помещать в отдельный экземпляр имеет смысл несколько логически связанных баз данных.

Для выполнения более тонкой настройки экземпляра на конкретную базу данных, следует каждую базу данных создавать в отдельном экземпляре.

Для создания дополнительного экземпляра необходимо запустить *Центр управления*, в контекстном меню папки *Экземпляры* (instance) выбрать пункт *Создать* (Create).

С другой стороны, тот же результат можно получить, выдав системную команду

```
db2icrt <имя экземпляра>, в командном окне DB2.
```

В процессе инсталляции, по умолчанию, создается экземпляр с именем DB2. Параметры DB2INSTANCE (устанавливает значение DB2 для переменной среды), DB2INSTDEF (устанавливает значение DB2 для переменной реестра) позволяют определить DB2 как экземпляр по умолчанию.

Установить переменную db2instance для текущего сеанса работы можно из командного окна, следующей командой:

```
Set db2instance = <имя экземпляра>.
```

Чтобы узнать, какое значение переменной db2instance установлено для текущего сеанса, следует воспользоваться командой:

```
Set db2instance.
```

Для установки переменной db2instance для всех сеансов работы необходимо изменить ее значение в диалоговом окне *Переменные среды* Панели свойств системы.

Глобальная установка переменной db2instance производится в случае, если db2instance вообще не установлена. Тогда она устанавливается на глобальном уровне реестра с помощью *Ассистента конфигурирования* (Configuration Assistant) или команды:

```
Db2set db2instdef = <имя нового экземпляра> -g .
```

Следует отметить, что значение, установленное для переменной среды DB2INSTANCE для уровня сеанса или системы, будет перекрывать значение DB2INSTDEF.

Для выполнения ряда задач (подключить базу данных к экземпляру, выполнить перекомпиляцию приложения и т.д.) необходимо запустить экземпляр сервера DB2.

Для этого необходимо, находясь в *Центре управления*, щелкнуть правой кнопкой мыши на экземпляре, который требуется запустить, и выбрать в контекстном меню пункт *Запуск (Start)*. Или, находясь в командном окне DB2, ввести системную команду Db2start.

Для останова выполняющегося экземпляра, соответственно, в контекстном меню экземпляра выбрать пункт *Остановить (Stop)* или ввести команду Db2stop.

Возможно запустить сразу несколько экземпляров, если они принадлежат программному коду одной и той же версии. Чтобы запустить сразу несколько экземпляров параллельно, можно поступить следующим образом:

- Находясь в *Центре управления*, запустить тот экземпляр который необходимо.
- Находясь в командном окне DB2, командой set db2instance присвоить переменной db2instance имя другого экземпляра, который должен быть запущен. Запустить его командой db2start, а затем повторно запустить сервер администратора DB2 Administration Server командами DB2admin stop, DB2admin start.

Каждый экземпляр может быть сконфигурирован отдельно. Стало распространенной практикой создавать один экземпляр для тестовых заданий, другой – для окончательных баз данных.

Параметры конфигурации, установленные на уровне менеджера баз данных или на уровне экземпляра, оказывают влияние на все базы в этом экземпляре. Конфигурационные параметры разбиты на восемь категорий: Управление (Administration), Связь (Communication), Диагностика (Diagnostic), Среда (Environment), Программы (Miscellaneous), Производительность (Perfomance), Монитор (Monitor) и Параллельно (Concurrent). Центр управления можно использовать для просмотра и изменений значений параметров, хранящихся в файле конфигурации менеджера данных (в контекст-

ном меню экземпляра следует выбрать *Конфигурировать параметры* [Configuration parameters]).

Установка конфигурационных параметров на уровне базы данных позволяет производить настройку каждой базы данных в отдельности. Для чего необходимо из *Центра управления* выбрать базу данных. Затем в контекстном меню последней указать пункт *Конфигурировать параметры* [Configuration parameters]).

Переменные реестра также оказывают помощь в управлении операционной средой как на глобальном уровне, так и на уровне экземпляра. Для просмотра и изменения переменных реестра нужно запустить *Ассистент конфигурирования* (Configuration Assistant), после чего выбрать пункт меню *Реестр DB2* (DB2 Registry).

Соединение с экземплярами необходимо для того, чтобы выполнить задачи сопровождения и обслуживания, которые решаются только на уровне экземпляра. В число таких задач входит создание базы данных, отключение приложений, мониторинг базы данных и т.д. Для соединения с экземпляром достаточно из *Центра управления*, в контекстном меню экземпляра выбрать пункт *Подсоединить* (Attach). Если необходимо, ввести имя пользователя и пароль для системы.

Чтобы удалить некоторый экземпляр DB2, следует выполнить следующие действия:

- Завершить все приложения, которые в данный момент использует удаляемый экземпляр.
- Остановить выполнение экземпляра, щелкнув правой кнопкой мыши на экземпляре и выбрав в контекстном меню пункт *Остановить* (Stop).
- При необходимости сохраните копии файлов, размещенных в каталоге `\sqlib\<имя удаляемого экземпляра>`.

- Удалить экземпляр, выбрав в контекстном меню экземпляра пункт *Удалить* (Remove).

Задание на лабораторную работу

1. Создать экземпляр *Examp*.
2. Выполнить установку переменной *DB2INSTANCE* для текущего сеанса.
3. Выполнить установку переменной *DB2INSTDEF* с помощью Ассистента конфигурирования.
4. Запустить экземпляр.
5. Создать базу данных в новом экземпляре.
6. Остановить экземпляр.
7. Запустить оба экземпляра параллельно.
8. Изменить значение параметра *Discover* с первоначального *Search* на *Known* (на уровне экземпляра).
9. На уровне базы данных изменить значение параметра *Discover* с первоначального *Включить* (*Enable*) на *Отключить* (*Disable*).
10. Изменить переменную реестра *DB2Comm*, добавив в список протокол *NETBIOS*.
11. Удалить экземпляр.

Тема: Управление данными в табличных пространствах

При работе с особо крупными базами данных, содержащими большие объекты, или с базами данных, к которым предъявляются специальные требования высокой производительности, для хранения данных необходимо использовать более сложные методы. Для определения способа хранения данных в системе DB2 предоставляется возможность использования табличных пространств, контейнеров и буферных пулов.

Логически базы данных организованы в виде *табличных пространств* (Table spaces). Табличные пространства состоят из физических устройств хранения, называемых контейнерами. Одно табличное пространство может быть распределено по нескольким контейнерам. *Буферный пул* (buffer pool) – это область памяти, используемая для кэширования таблиц и для индексирования страниц данных при их считывании с диска либо изменении.

Для создания таблиц в базе данных вовсе не обязательно создавать табличное пространство, контейнер, буферный пул. Достаточно при создании базы данных и таблицы в базе данных принять параметры этих объектов по умолчанию. По умолчанию создаются три табличных пространства:

- SYSCATSPACE. Обычное табличное пространство, используемое для хранения таблиц системного каталога.
- TEMPSPACE1. Временное табличное пространство, используемое для хранения и реорганизации таблиц, для создания индексов и объединения таблиц.
- USERSPACE1 Обычное табличное пространство, используемое для хранения данных и индексов таблиц.

Использование табличных пространств для хранения данных обеспечивает гибкость в присвоении таких составляющих, как дан-

ные, индексы и длинные поля данных различным табличным пространствам. В результате пользователь получает возможность назначать различные устройства хранения в зависимости от содержимого каждого табличного пространства. Можно выполнять резервное копирование и восстановление табличного пространства как единого целого. Следовательно, копирование некоторых табличных пространств, содержащих более часто обновляемые данные, можно выполнять чаще.

Табличное пространство может быть создано с помощью мастера, для этого нужно в контекстном меню папки *Табличные пространства* (Table Space) выбрать пункт *Создать* (Create). При создании табличного пространства вам необходимо указать его тип:

- *Обычное* (Regular). Этот тип используется для всех видов табличных данных, за исключением временных таблиц. При создании табличного пространства с помощью мастера по созданию таблиц этот тип является единственным доступным.
- *Большое* (Large). Этот тип используется для создания пространств для хранения больших объектов данных. При выборе этого типа система DB2 использует для такого табличного пространства высокопроизводительный метод управления.
- *Системное временное* (System Temporary). Этот тип позволяет создавать табличные пространства для хранения временных системных таблиц. Временное пространство используется во время выполнения сортировок, объединений и ряда других операций. По умолчанию оно имеет имя Tempspase1.
- *Пользовательское временное* (User Temporary). Этот тип применяется при создании табличного пространства для хранения объявленных временных таблиц.

При определении управления тем, как и где будут храниться табличные пространства, следует выбрать:

- *Пространство управляемое системой (SMS)*. В этом случае управление осуществляется со стороны операционной системы. Эта опция недоступна, если до этого был выбран тип табличного пространства *Большой (Large)*. Для хранения множества небольших таблиц в табличном пространстве следует использовать тип SMS.
- *Пространство управляемое базой данных (Database – Managed Spase)*. Этот тип управления предоставляет множество возможностей настройки, в том числе предварительное определение размера каждого табличного пространства и добавление контейнеров при увеличении объема данных. Одно из преимуществ DMS – возможность отделения индексов и длинных данных от остальных данных. Резервное копирование, восстановление и настройку производительности каждого табличного пространства можно выполнять отдельно от остальных, что позволяет обеспечить более высокую производительность.

Также во время создания табличного пространства необходимо определить *Контейнеры (Containers)*. Контейнер определяет место хранения данных. При его определении может потребоваться задать следующее:

- *Имя контейнера, диск и каталог*. Выберите каталог, который не будет использоваться какими-либо другими компонентами системы. Следует указать имя несуществующего файла, поскольку он будет создан автоматически.
- *Размер файла*. Для DMS необходимо определить размер контейнера (в Мб или в страницах по 4 кб). Минимальный размер контейнера равен 5-кратному объему табличного пространства плюс одна страница размером 4 кб. (По умолчанию $(32 \times 4 \text{ кб}) \times 5 + 1 = 161$ страница).
- *Файл*. В качестве контейнера может быть выбран файл или непосредственное устройство. Использование файла проще, однако производительность будет ниже. В качестве непосредственного устройства можно использовать неформати-

рованный логический раздел или не разбитый на логические разделы диск (DB2 обращается непосредственно к диску).

Для конкретного табличного пространства данные равномерно распределяются по контейнерам. Размер экстенда (extent size) позволяет определить количество страниц, которые DB2 записывает в один контейнер, прежде чем использовать другой. Размер предварительной выборки (prefetch size) показывает, какой объем данных считывается с диска в ожидании использования.

Стратегия добавления пространства для табличных пространств DMS и SMS различна. Для достигшего предельной емкости табличного пространства DMS можно изменить размеры существующих контейнеров либо добавить дополнительные контейнеры.

В общем случае увеличить объем табличного пространства SMS нельзя, не увеличивая размера файловой системы, на основе которой оно создано. Это объясняется тем, что емкость SMS зависит от объема доступного пространства файловой системы и максимального размера файла, поддерживаемого операционной системой.

Буферный пул, или пул буферов (buffer pool), - это область основной памяти, в которую временно считываются или изменяются страницы базы данных. Его определение позволяет влиять на производительность системы баз данных. Размер страницы табличного пространства должен быть равен размеру страницы, связанного с ним пула. Размер страницы пула буферов нельзя изменить после его создания. Чтобы увеличить его размер следует удалить и снова создать буферный пул.

Задание на лабораторную работу

- 1. Создать табличное пространство DMS.*
- 2. Добавить несколько контейнеров к диску, на котором хранятся данные (например, F:\container1, F:\container2).*

3. Создать буферный пул.
4. Создать табличное пространство для индексов (имя – CDINDEX, Тип – обычное, Управление пространством – DMS, Размер файла – Файл 2 Мб, Контейнер – F:\CDINDEX).
5. Создайте новую таблицу Ses (о сессиях) в табличном пространстве, созданном в п 1:

NS – номер сессии;

PrS – признак сессии (весенняя или осенняя).

Лабораторная работа № 6

Тема: Доступ к данным

Получать доступ к информации, хранящейся в базе данных DB2, можно с помощью следующих средств:

- Центра управления;
- Командного центра;
- Приложений ODBC, таких, как Microsoft Access;
- Приложений JDBC, реализованных на языке Java;
- Приложений, написанных на C, C++ или других языках программирования.

Доступ к данным посредством *Центра управления* осуществляется выбором пункта меню *Пример содержимого* (Sample Contents) из контекстного меню интересующей таблицы. Данное средство позволяет только просматривать строки выбранной таблицы.

Окно командного центра позволяет решать следующие задачи:

- Выдавать SQL-запросы, команды DB2 и системные команды MVS.
- Просматривать результаты выполнения одного или нескольких SQL-операторов и команд DB2 в окне результатов. При этом можно выполнять прокрутку результатов и сохранять выходную информацию в файле.
- Сохранять последовательность SQL-операторов и команд DB2 в файле сценария. Повторно вызывать и запускать сценарий.
- Просматривать план выполнения и статистическую информацию, связанную SQL-оператором, перед его выполнением.
- С помощью Центра сценариев отображать все командные сценарии, известные системе, с выводом обобщенной информации по каждому из них.
- Получать доступ к локальным и удаленным базам данных.

Командный центр позволяет вводить SQL-операторы для просмотра и манипулирования данными в базах данных.

Microsoft Access – это интерфейс баз данных, который позволяет осуществить доступ к данным. Это приложение можно установить как в той же системе, в которой размещаются базы данных, так и в системе удаленного клиента.

Чтобы получить доступ к базам данных DB2 с помощью Microsoft Access, предварительно нужно целевую базу данных зарегистрировать как источник данных ODBC. Это выполняется из окна *Ассистент конфигурирования* (Configuration Assistant) выбором пункта контекстного меню целевой базы данных *Изменить базу*

данных (Change Database). Далее, следуя указаниям мастера по изменению таблиц, установить соединение.

Для просмотра и манипулирования данными запустить приложение Microsoft Access. Выбрать в меню Файл(File) – *Открыть* (Open). В открывшемся диалоговом окне в поле *Тип файлов* (Files of Type) выбрать из списка ODBC Databases, далее, из списка всех системных источников данных на компьютере, выбрать нужный и установить стандартное соединение. После чего выбранные таблицы базы данных будут представлены в формате Access. Для выполнения дополнительных задач над данными DB2 следует воспользоваться инструкциями, предоставляемыми Microsoft Access.

Средства доступа к данным, такие, как приложения JDBC, реализованные на языке Java; приложения, написанные на С, С++, на лабораторных работах нами рассматриваться не будут, они могут быть изучены самостоятельно.

Задание на лабораторную работу

- 1. Осуществить просмотр информации, хранящейся в базе данных с помощью Центра управления.*
- 2. Осуществить доступ к информации, хранящейся в базе данных с помощью Командного центра, произвести модификацию данных.*
- 3. Используя Microsoft Access получить доступ к данным.*
- 4. Создать формы заполнения таблиц Stud, Session, Ses.*
- 5. Сформировать отчет об успеваемости студентов.*

Тема: Ограничения целостности

В любой производственной среде данные часто должны соответствовать определенным ограничениям или правилам. Например, номер зачетной книжки студента должен быть уникальным. В качестве средства реализации таких правил DB2 предоставляет *Ограничения* (constraints). В DB2 существуют следующие типы ограничений:

- *Ограничение Not Null* предотвращает ввод нулевых значений в столбце.
- *Ограничение уникальности* обеспечивает, чтобы значения в столбце были уникальными и ненулевыми для всех строк таблицы.
- *Проверочное ограничение* – правило базы данных, которое определяет значения, допустимые в одном или более столбцах каждой строки таблицы.
- *Первичный ключ*. Каждая таблица может содержать один первичный ключ. Ограничения первичного и внешнего ключа служат для определения отношений между таблицами.
- *Внешний ключ* (ограничение целостности ссылок или реляционное ограничение) позволяет определить требуемые отношения между и внутри таблиц.

Ограничения целостности ссылок или ссылочная целостность предполагает выполнение некоторых правил при операциях удаления, изменения данных. Так, например, при определении внешнего ключа таблицы необходимо указать ограничения на связь с родительской таблицей при выполнении вышеуказанных операций.

Выбрав ограничение No Action (*Действие отсутствует*) или Restrict (*Ограничить*) для операции удаления, система будет выво-

дить сообщение об ошибке при попытке удаления строки в родительской таблице, если имеются зависимые строки, и никакие строки удаляться не будут. При выборе действия Cascade (*Каскадировать*) строки в родительской таблице удаются вместе с зависимыми строками в зависимых таблицах. Выбор действия Set Null предполагает, что в случае удаления строки в родительской таблице все столбцы в любых зависимых строках будут заменяться столбцами со значением Null. Если это допускается определениями таблиц.

При изменении таблиц действия выбираются аналогично.

Триггеры сложнее и потенциально обеспечивают большие возможности по сравнению с ограничениями. Они определяют набор действий, которые выполняются в сочетании с операторами Insert, Update, Delete в указанной базовой таблице или запускаются данными операторами. Примерами применения триггеров служат:

- Проверка допустимости вводимых данных;
- Генерирование значения для вставляемой строки;
- Считывание данных из других таблиц для реализации перекрестных ссылок.
- Запись данных в другие таблицы для целей аудита.

Создаются триггеры выбором соответствующего пункта в контекстном меню папки *Триггеры* (Triggers). При этом имя триггера должно содержать менее 18 символов. Время запуска триггера может принимать следующие значения:

- Before (*до*) активизирует триггер до выполнения действий Insert, Update, Delete;
- After (*после*) активизирует триггер после выполнения действий Insert, Update, Delete.

Кроме этого следует указать операцию, которая активизирует триггер. На странице *Действие триггера* (Triggered Action) можно указать внутриоператорные имена старых и новых строк и временные таблицы для старых и новых строк. Внутриоператорные имена и временные таблицы необходимы в случае, если нужно сослаться на строки до и после их обновления. Следует также указать, должен ли триггер запускаться при каждом выполнении операции над строкой или только один раз после выполнения операции над всеми строками. В текстовой области *Действие триггера* (Triggered Action) введите SQL-оператор, определяющий триггер.

Задание на лабораторную работу

- 1. В таблицу, хранящую сведения о студентах на уровне поля, добавить ограничения на уровне поля: номер зачетной книжки должен быть больше 1000.*
- 2. Задать ограничение на уровне записи для таблицы, хранящей результаты сессии, – сумма оценок по трем предметам не должна превышать 15.*
- 3. Задать ограничение целостности ссылок для таблиц Stud и Session для операции удаления и обновления. В первом случае Cascade, во втором Restrict.*
- 4. С помощью триггера задать ограничение на год рождения.*
- 5. Создать триггер, который позволяет изменить значение поля.*

Лабораторная работа № 8

Тема: Создание и использование сценария команд DB2

Мини-приложение, называемое сценарием (script), – это набор из одной или более команд, созданных с помощью *Центра управления* или *Центра задач*. Сохранение часто применяемых запросов

и команд DB2 в сценариях команд DB2 позволяет упростить и более четко организовать работу. Несколько задач можно объединить в группу, которая выполняется регулярно или только при успешном или неуспешном завершении другой задачи. Результаты можно отправить по электронной почте. Для создания сценария команд необходимо выполнить следующее:

- Запустить *Центр задач* (можно из *Центра управления* соответствующей пиктограммой или *Запуск/ Программы/ IBM DB2/Общие инструменты управления*).
- На вкладке *Задача* (Task) задать имя сценария, тип создаваемой задачи, описание сценария, категорию задачи, систему выполнения, экземпляр и раздел DB2, в котором будет запускаться сценарий.
- На вкладке *Командный сценарий* (Command Script) ввести команды, из которых состоит сценарий и путь к рабочему каталогу, в котором будут сохранены результаты работы сценария.
- На вкладке *Параметры запуска* (Run Properties) выбрать или создать набор кодов успешного завершения, задающий условия, при которых задача считается успешно завершённой. По умолчанию успешным считается код возврата 0.
- На вкладке *Расписание* (Schedule) задать информацию о времени выполнения задач.
- На вкладке *Уведомление* (Notification) можно задать отправку уведомления о результатах выполнения сценария по электронной почте или записать результат в виде сообщения в Журнале.
- На вкладке *Действия задач* (Task Actions) можно задать действия после выполнения сценария.

- Вкладка *Защита* (Security) позволяет назначить каждому пользователю право читать, запустить или записать сценарий.

Сохраненный сценарий или расписание можно запустить позже в другое время или другой день. Это можно выполнить, выбрав из контекстного меню сценария пункт *Запустить сейчас* (Run Now). Для просмотра результатов выполнения сценария следует выбрать пункт *Показать результаты* (Show Results) из контекстного меню сценария.

Для просмотра всей информации, сгенерированной Центром управления и его компонентами, можно использовать журнал. Журнал предусматривает несколько режимов просмотра:

- *Хронология задач* позволяет просматривать информацию о задачах DB2, выполненных на данном компьютере.
- *Хронология базы данных* позволяет увидеть хронологию резервных копий базы данных, выполненных на данном компьютере.
- *Сообщения* позволяет увидеть список всех прошлых сообщений, касающихся DB2.
- *Журнал уведомлений* позволяет увидеть уведомления, сгенерированные Монитором работоспособности.

Журнал хранит и может отображать всю информацию о задачах, действиях и операциях баз данных, а также действиях, сообщениях и предупреждениях Центра управления.

Открыть журнал можно из Центра управления, выбрав соответствующую пиктограмму в панели инструментов (или Пуск/Программы/IBM DB2/Общие инструменты управления/ Журнал).

Задание на лабораторную работу

1. *Создать сценарий подключения к базе данных Base, просмотра списка таблиц базы и выборки содержимого таблицы Stud.*

*(connect to Base;
List tables;
Select * from Stud;)*

2. *Данный сценарий выполнять ежемесячно в определенный день в течение 3 месяцев.*
3. *Запустить сценарий сейчас, посмотреть результаты.*
4. *С помощью Журнала убедиться в выполнении сценария.*

Лабораторная работа № 9

Тема: Резервное копирование и восстановление

Одним из преимуществ систем управления базами данных по сравнению с хранением данных в файлах является наличие процедур резервного копирования и восстановления, с помощью которых можно воссоздать базу данных при каком-либо ее повреждении.

При создании базы автоматически создаются файлы журналов восстановления и файл хронологии восстановлений. DB2 назначает для каждой создаваемой базы данных первичные и вторичные файлы журналов, необходимые для операций восстановления. Первичные файлы журналов обеспечивают заранее выделенное фиксированное количество памяти для файлов журнала восстановления. Вторичные файлы журналов используются при переполнении первичных файлов журналов и выделяются по одному по мере необходимости. По умолчанию система настроена на циклическое ведение журналов. При таком типе ведения журналов разрешены только

полные автономные резервные копирования. При полном резервном копировании база данных должна пребывать в автономном состоянии (быть недоступной пользователям). Циклическая запись использует «кольцо» оперативных журналов для обеспечения восстановления при ошибках проведения транзакций и авариях системы. Журналы используются и сохраняются лишь до точки, гарантирующей целостность текущих транзакций. В этом случае поддерживаются только Аварийное восстановление и Восстановление версии. Архивное ведение журналов используется специально для восстановления с повтором транзакций. Для архивирования журналов вы можете указать DB2, оставляя файлы журналов в активном каталоге и затем архивировать их вручную, или же вы можете автоматизировать архивирование, установив пользовательскую программу-обработчик. Архивные журналы – это журналы, которые раньше были активными, но теперь для аварийного восстановления не требуются. Во время аварийного восстановления откатываются незафиксированные изменения, записанные в этих журналах. Изменения, зафиксированные, но незаписанные из памяти, на диск повторяются. Активные журналы размещаются в каталоге журналов базы данных. Изменить способ конфигурирования журнала можно с помощью *Мастера по конфигурированию записи базы данных*, для запуска которого в контекстном меню определенной для копирования базы выбрать пункт *Конфигурировать запись базы данных в журнал* (Configure Database Logging). А далее, в последовательно открывающихся окнах, выставить необходимые параметры.

Резервное копирование баз данных предполагает создание копии информации и сохранение ее на другом носителе на случай сбоя или повреждения оригинала.

Для облегчения резервного копирования и восстановления баз данных служит *Мастер резервного копирования* Backup Wizard. Для его запуска достаточно, находясь в Центре управления, из контекстного меню базы данных выбрать пункт *Резервная копия* (Backup). Создание резервной копии базы данных позволит в дальнейшем восстановить данные в случае их утери в результате проблем с внешними носителями и памятью, сбоев в питании и оши-

бок в приложениях. В случае повреждения базы данных вы можете воссоздать ее с помощью *Мастера по восстановлению данных* (Restore Data Wizard).

DB2 автоматически обрабатывает проблемы, возникающие из-за сбоев питания. Она автоматически перезапускает и возвращает базу данных в состояние, в котором она находилась при завершении последней транзакции. Сбой носителей и ошибки приложений – это более серьезный случай, который должен быть предусмотрен с точки зрения восстановления данных.

Мастер по резервному копированию необходим лишь только при создании копии базы данных первый раз. Он позволяет выбрать значения параметров, необходимых при данной операции. Для запуска в контекстном меню базы данных следует выбрать пункт *Резервная копия* (Backup). И далее, в открывающихся друг за другом окнах, выставить нужные параметры.

Мастер по восстановлению данных помогает восстановить базу данных в состояние до момента последнего резервного копирования базы или до последней завершенной транзакции. С помощью Мастера по восстановлению данных можно произвести полное восстановление базы данных, но на последнем шаге указать дату и время, до которых должен осуществляться повтор транзакций с помощью журналов. Чтобы запустить мастер в контекстном меню базы данных, выберите пункт *Восстановить* (Restore); и в появляющихся диалоговых окнах укажите необходимые параметры.

Каждая операция резервного копирования включает копию файла хронологии восстановления, связанного с базой данных. Этот файл содержит сведения о резервных копированиях, которые могут быть использованы в случае необходимости полного или частичного восстановления базы данных до заданного момента времени. Информация, содержащаяся в файле, включает:

- описание, какая часть базы была скопирована и как;
- время создания копии;

- место расположения копии;
- момент времени, когда проводилось последнее восстановление.

Каждая резервная копия DB2 содержит копию этого файла, поэтому он может быть восстановлен из любой резервной копии. Для просмотра файла хронологии базы откройте *Журнал* (Journal) [пиктограмма в панели инструментов Центра управления] и выберите соответствующую вкладку.

Задание на лабораторную работу

- 1. С помощью Мастера по конфигурированию записи базы данных в журнал изменить стандартную циклическую запись на архивную.*
- 2. С помощью Мастера по резервному копированию создать копию.*
- 3. Исследуйте доступные опции резервного копирования и восстановления.*
- 4. Просмотрите файл хронологии базы данных.*
- 5. Восстановите копию.*

Учебное издание

**DB2 UNIVERSAL DATABASE – СРЕДСТВА
СОЗДАНИЯ И УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ**

Методические указания

Составитель **Логанова Лилия Владимировна**

Редактор **О.Ю. Дьяченко**
Компьютерная верстка **О.А. Ананьев**

Подписано в печать 13.11.2006 г. Формат 60x84 1/16.
Бумага офсетная. Печать офсетная.
Усл. печ. л. 2,09; Усл. кр.-отт. 2,21; Уч.-изд.л. 2,25.
Тираж 100 экз. Заказ 25. Арт. 14(Д1)/2006

Самарский государственный аэрокосмический
университет имени академика С.П. Королева
443086, Самара, Московское шоссе, 34

Издательство Самарского государственного
аэрокосмического университета
443086, Самара, Московское шоссе, 34.