ФГБОУ ВПО «Воронежский государственный
технический университет»

Кафедра систем информационной безопасности

338-2014

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к лабораторным работам № 1–5 по дисциплинам
«Основы построения защищенных СУБД»,
«Безопасность систем баз данных»
для студентов специальностей 090301
«Компьютерная безопасность»,
090303 «Информационная безопасность
автоматизированных систем»
очной формы обучения

Воронеж 2015

Составитель канд. техн. наук Д. Г. Плотников

УДК 004.056.5

Методические указания к лабораторным работам № 1–5 по дисциплинам «Основы построения защищенных СУБД», «Безопасность систем баз данных» для студентов специальностей 090301 «Компьютерная безопасность», 090303 «Информационная безопасность автоматизированных систем» очной формы обучения / ФГБОУ ВПО «Воронежский государственный технический университет»; сост. Д. Г. Плотников. – Воронеж, 2015. 55 с.

Выполняя лабораторные работы, студенты получают знания и навыки по современным базам данных и системам управления базами данных, в частности для построения систем баз данных, управления данными с помощью языка SQL и других средств современных СУБД. В издании рассматриваются реляционные базы данных и системы управления базами данных.

Методические указания подготовлены в электронном виде в текстовом редакторе MS Word 2013 и содержатся в файле Плотников_ЛР_БД_1-5.pdf.

Табл. 3. Ил. 5. Библиогр.: 7 назв.

Рецензент д-р техн. наук, проф. А.Г. Остапенко

Ответственный за выпуск зав. кафедрой д-р техн. наук, проф. А.Г. Остапенко

Издается по решению редакционно-издательского совета Воронежского государственного технического университета

Лабораторная работа № 1
Работа в среде интерактивного SQL

Цель работы

Ознакомление с возможностями сред интерактивного SQL СУБД MS SQL Server и приобретение начальных навыков работы в этих средах.

Темы для предварительной проработки

- Структура базы данных «Учебная база данных» (см. приложения 1 и 2: Концептуальная схема, Структура таблиц).
- Оператор SELECT.

Выполнение работы

- Ознакомиться с индивидуальным заданием и сформулировать SQL-запросы, соответствующие Вашему варианту индивидуального задания.
- Запустить Microsoft Query Analyzer и ознакомиться со средой выполнения, им обеспечиваемой.
- Соединиться с рабочей базой данных.
- Выполнить подготовленные запросы в среде Microsoft Query Analyzer и убедиться в том, что их результаты правильно отражают поставленное задание. (Для того, чтобы и убедиться в правильности результатов, выполнить более простые запросы и сопоставить их результаты с полученными)
- Сохранить в файле запросы и результаты их выполнения.
- Закончить работу с Microsoft Query Analyzer.

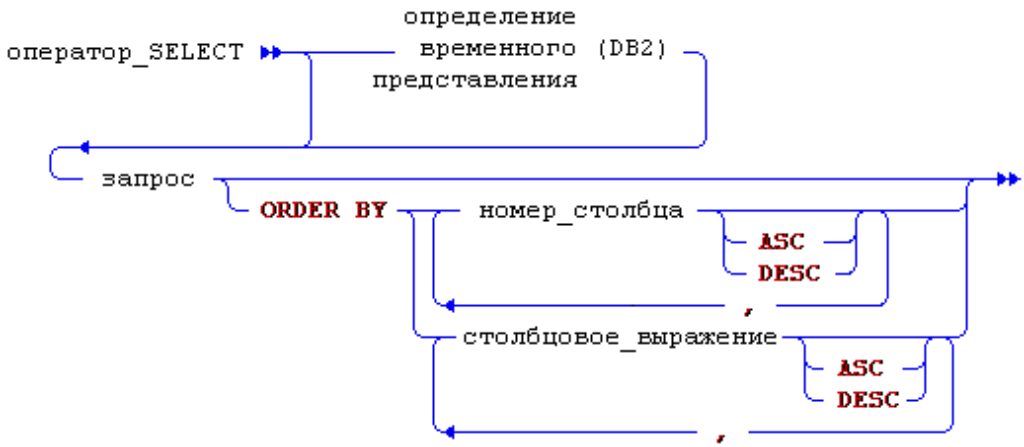
Содержание отчета

- Протокол выполнения запросов в среде Microsoft Query Analyzer.
- Краткие выводы о навыках, приобретенных в ходе выполнения работы.

Теоретические сведения

Операторы манипулирования данными

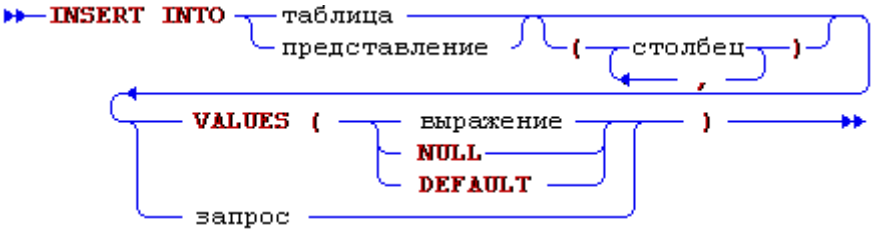
Оператор SELECT выполняет выборку данных из таблиц базы данных. Синтаксис оператора:



ORDER BY не является реляционной операцией, поэтому применяется только последней - после всех реляционных действий. Если аргументом является список, то сортировка ведется по первому элементу списка, при равенстве значений - по второму и т.д.

Стандарт требует упорядочивания только по номеру столбца в результирующей таблице, но все СУБД допускают и вторую форму. Столбцовое выражение в этом случает должно быть в точности то же, что и в списке выборки. Использование здесь псевдонимов не допускается.

Оператор INSERT вставляет данные в таблицу или представление

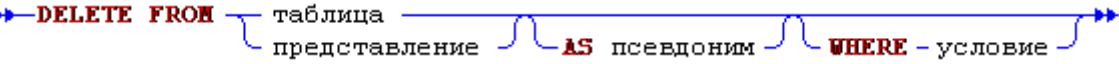


Таблица, в которую происходит вставка, должна быть базовой таблицей.

Список столбцов может быть опущен в том случае, если в фразе VALUES задаются значения для всех столбцов в том порядке, в каком столбцы были описаны при создании таблицы.

Если вставляемые данные задаются запросом, то оператором INSERT может быть вставлено более одной строки.

Оператор DELETE удаляет из таблицы или представления одну или несколько строк.

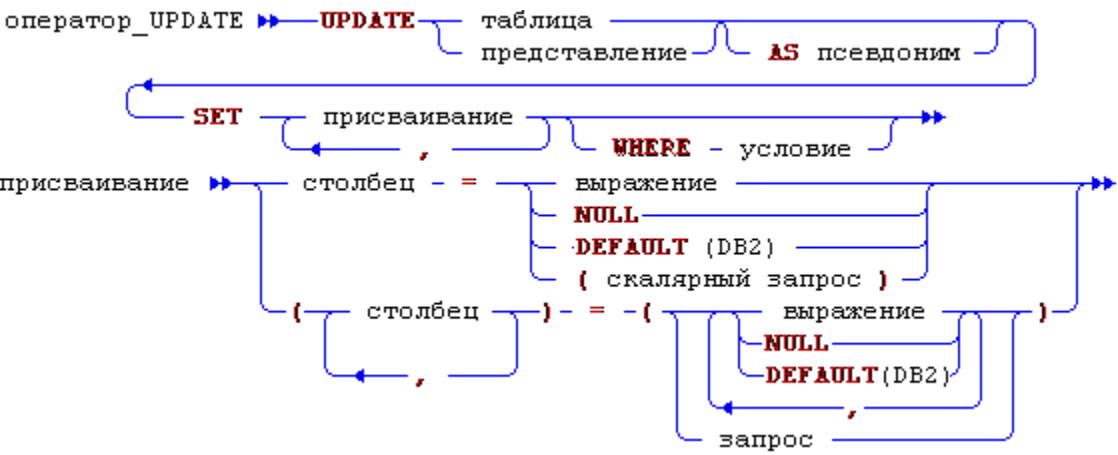


Таблица, из которой происходит удаление, должна быть базовой таблицей.

Синтаксис фразы WHERE в операторе DELETE - тот же, что и в запросе.

Присвоенный таблице/представлению псевдоним может использоваться в выражениях фразы WHERE.

Оператор UPDATE изменяет значения в заданных столбцах удаляет одной или нескольких строк таблицы или представления.



Таблица, в которой происходит изменения, должна быть базовой таблицей.

Синтаксис фразы WHERE в операторе UPDATE - тот же, что и в запросе.

Присвоенный таблице/представлению псевдоним может использоваться в выражениях фразы WHERE.

Контрольные вопросы

1. Объяснить, как работают написанные запросы.
2. Рассказать про операцию соединения (JOIN) и различные её разновидности.
3. Рассказать про агрегатные функции, предложения GROUP BY и HAVING.
4. Как выбрать только уникальные значения какого-либо столбца?
5. Как осуществить сортировку по возрастанию/убыванию по значению какого-либо столбца?
6. Как агрегатные функции ведут себя по отношению к неопределённым значениям?
7. Рассказать о теоретико-множественных операциях в SQL.
8. Чем отличаются UNION и UNION ALL?

9. Чем отличаются COUNT(*) и COUNT(field)?
10. Как подсчитать количество уникальных значений столбца?
11. Как можно осуществить проверку на неопределенное значение?
12. Рассказать про предикат LIKE.
13. Как можно выбрать только определенное количество строк?
14. Чем SQL-таблица отличается от отношения?

Дополнительные вопросы

1. Исправить неверно работающий запрос (запросы).
2. Упростить один или несколько запросов.
3. Округлить результирующее значение до 3 знаков после точки.
4. Округлить вещественное число до целого без нулей после точки.
5. Переписать запрос, не используя функцию MAX (MIN).
6. Изменить формат вывода данных (например, формат даты и времени).

Лабораторная работа № 2
Создание таблиц

Цель работы

Ознакомление со способами создания таблиц в режимах интерактивного SQL и мастеров СУБД MS SQL Server и приобретение начальных навыков работы в средах мастеров.

Темы для предварительной проработки

- Оператор CREATE TABLE, типы данных SQL, декларативные ограничения целостности.

Подготовка к работе

- Изучить ER-диаграмму, приведенную в вашем варианте индивидуального задания, при необходимости - модифицировать ее.
- Заданная ER-диаграмма дает лишь ориентировочное представление о структуре данных заданной Вам предметной области, Вы можете модифицировать ее (но не в сторону упрощения!), добавив новые сущности, атрибуты, связи.
- Конвертировать ER-диаграмму в концептуальную схему, отображаемую на реляционные таблицы.
- Составить SQL-скрипт для создания таблиц базы данных, включив в него все требуемые логикой предметной области декларативные ограничения целостности (первичные и внешние ключи, проверочные ограничения и т.п.).
- Утвердить результаты предварительной подготовки у преподавателя.

Выполнение работы

- Запустить Microsoft Query Analyzer и соединиться с локальной базой данных.
- Выполнить часть подготовленных операторов CREATE TABLE в среде Microsoft Query Analyzer.
- Сохранить в файле оператор и результаты его выполнения.
- Запустить Microsoft Enterprise Manager и открыть локальную базу данных. Ознакомиться со средой выполнения Microsoft Enterprise Manager.
- Создать остальные таблицы подготовленной базы данных средствами мастера таблиц Microsoft Enterprise Manager.
- Закончить работу с MS SQL Server.

Содержание отчета

- Протокол создания таблицы в среде Microsoft Query Analyzer.
- Краткие выводы о навыках, приобретенных в ходе выполнения работы.

Теоретические сведения

Таблицы создаются оператором SQL CREATE TABLE.

Для целей нашего лабораторного практикума в большинстве случаев может быть достаточна общая форма оператора, синтаксис которой показан на следующей диаграмме (Рис. 1).

Некоторые диалектные различия в применении оператора описаны в табл. 1.

Для целей нашего лабораторного практикума в большинстве случаев можно обойтись типами данных, представленными в табл. 2.

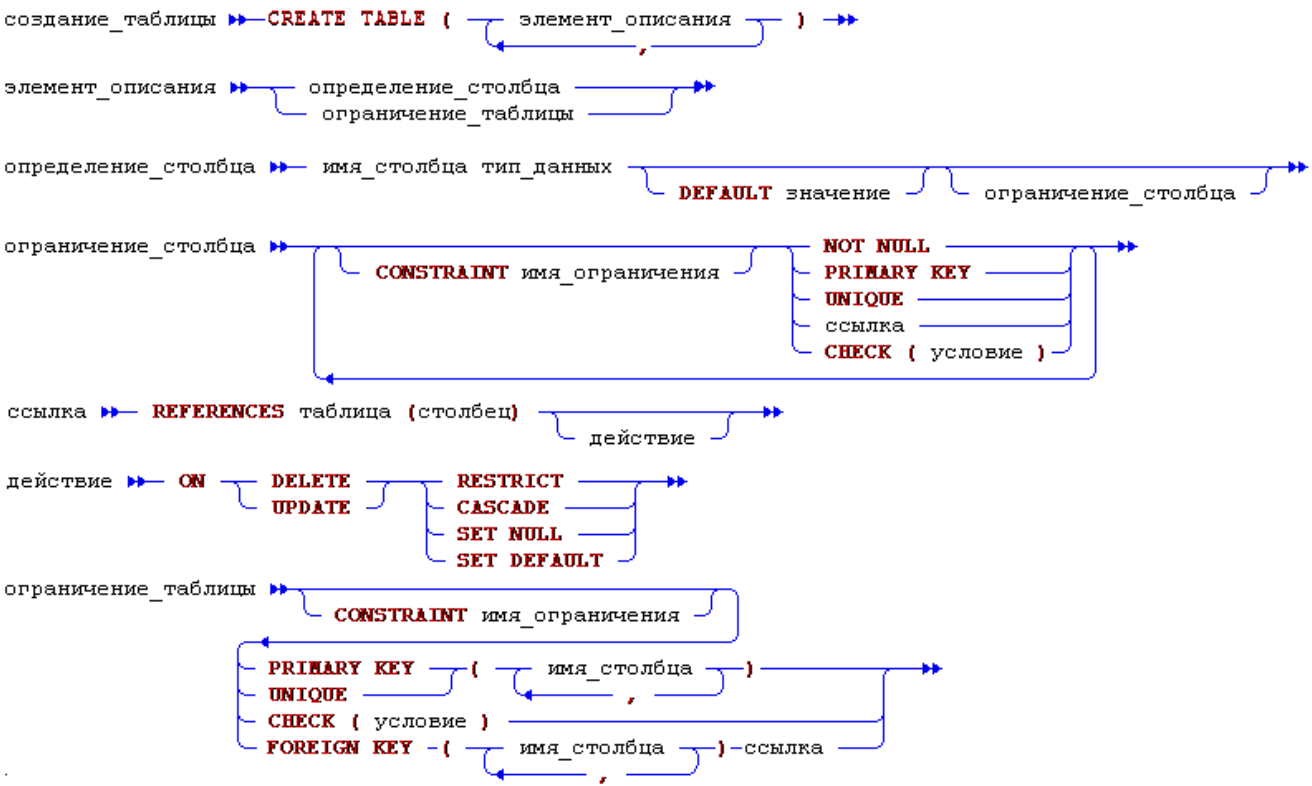


Рис. 1. Общая форма оператора SQL CREATE TABLE

Таблица 1

Некоторые диалектные различия в применении оператора

MSSQL	**Oracle**	**DB2**
PRIMARY KEY подразумевает NOT NULL	PRIMARY KEY подразумевает NOT NULL	PRIMARY KEY употребляется только вместе с NOT NULL
ON DELETE CASCADE - (задается отдельно), RESTRICT - не задается.	ON DELETE CASCADE / RESTRICT (по умолчанию)	ON DELETE NO ACTION, RESTRICT (по умолчанию) / CASCADE / SET NULL Разница между NO ACTION и RESTRICT: RESTRICT проверяется перед всеми остальными ограничениями,NO ACTION - после.
ON UDATE CASCADE (задается отдельно)	ON UPDATE - не задается.	ON UPDATE NO ACTION / RESTRICT (по умолчанию).

Таблица 2

Типы данных

Тип данных	Стандарт ANSI	DB2	Oracle	MSSQL	Примечания
Символьные строки постоянной длины	CHARACTER (n)	CHAR(n)	CHAR(n)	CHAR(n)	В DB2 максимальный размер типа CHAR - 256 символов, в Oracle - 2000, в MSSQL - 8000
Символьные строки переменной длины	CHARACTER VARIYNG(n)	VARCHAR (n)	VARCHAR2(n) VARCHAR(n)	VARCHAR (n)	В DB2 максимальный размер типа VARCHAR - 32К символов, в Oracle - 4000, в MSSQL - 8000. VARCHAR2 и VARCHAR в Oracle пока являются синонимами. В будущих версиях для них, возможно, будут введены разные правила сравнения

Продолжение табл. 2

Символьные строки в национальной кодировке	-	GRAPHIC(n) VARGRAPHIC(n)	NCHAR(n) NAVARRCHAR2(n)	NCHAR(n) NVARCHAR(n)	Тип данных для представления строк национальных алфавитов в 2-байтной кодировке. Ограничения длин для DB2 и Oracle - те же, что и для CHAR, VARCHAR, а для MSSQL - 4000 символов. n - число символов
Двоичные данные	BIT(n) BIT VARIYNG(n)	CHAR(n) FOR BIT DATA VARCHAR(n) FOR BIT DATA	- RAW(n)	BIT	Двоичные данные могут использоваться для тех же целей, что и строковые, но эти типы не сравниваются. При импорте/экспорте двоичные данные, в отличие от символьных, не перекодируются

Продолжение табл. 2

Числовые данные	NUMERIC (p,s) DECIMAL (p,s) SMALLINT INTEGER FLOAT(p) REAL DOUBLE PRECISION	NUMERIC(p,s) DECIMAL (p,s) SMALLINT INTEGER BIGINT FLOAT(p) REAL DOUBLE	NUMBER(p,s)	NUMERIC(p,s) DECIMAL(p,s) SMALLINT INT TINYINT BIGINT FLOAT(p) REAL DOUBLE	В Oracle существует единственный тип числовых данных, но Oracle «понимает» также и все ANSI-типы, переводя их в свой тип NUMBER. Размеры целочисленных типов: TINYINT (только MSSQL) - 1 байт, SMALLINT - 2 байта, INTEGER - 4 байта, BIGINT (только DB2) - 8 байтов.
Дата и время	DATE TIME TIMESTAMP INTERVAL	DATE TIME TIMESTAMP	DATE	DATETIME SMALLDATETIME TIMESTAMP	

Информацию о структуре созданной таблицы можно получить:

- В DB2 Command Center - команда:
- DESCRIBE TABLE *имя таблицы*
- В Oracle8 SQL*Plus - команда:
- DESC *имя таблицы*
- В Microsoft Query Analyzer - выборка из каталога метаданных:
-
- SELECT so.name AS TABLE_NAME,
- sc.name AS COLUMN_NAME,
- st.name AS TYPE_NAME,
- sc.length,
- sc.isnullable
- FROM syscolumns sc,sysobjects so, systypes st
- WHERE sc.id=so.id AND
- sc.xtype=st.xtype AND
- so.xtype='U' AND
- so.name='*имя таблицы*'

ORDER BY 1,2DESC

Контрольные вопросы

1. Объяснить, что делают написанные запросы.
2. В чем различие типов CHAR и VARCHAR? VARCHAR и NVARCHAR?
3. Что такое внешний ключ?
4. Какие существуют способы поддержания ссылочной целостности?
5. Что такое уникальный ключ?
6. Что такое нормализация?
7. Рассказать о нормальных формах.
8. Что такое IDENTITY?
9. Рассказать о значениях по умолчанию и неопределенных значениях.
10. Рассказать о вычисляемых столбцах.
11. Как можно представить значение булевского типа?
12. Как можно хранить даты и время?
13. Рассказать о числовых типах данных.
14. Каким образом можно вставить несколько строк с помощью одного оператора INSERT?
15. Как ведет себя оператор INSERT, если в списке столбцов перечислены не все столбцы?

Дополнительные вопросы

1. Добавить какие-либо ограничения целостности.
2. Добавить IDENTITY.

Лабораторная работа № 3
Манипулирование данными

Цель работы

Освоение способов манипулирования данными в средах интерактивного SQL СУБД Microsoft SQL Server.

Темы для предварительной проработки

- Операторы манипулирования данными языка SQL.

Подготовка к работе

- Подготовить данные для заполнения таблиц, созданных при выполнении лабораторной работы № 2. Объем подготовленных данных должен составлять не менее 10 экземпляров для каждой из стержневых сущностей и 20 экземпляров для каждой из ассоциативных.
- Подготовить SQL-скрипты для внесения данных в базу данных.
- Составить SQL-скрипты для выполнения выборок, заданных в Вашем варианте индивидуального задания, а также еще 3-4 выборок, имеющих осмысленное значение для предметной области.
- Сформулировать 3-4 запроса на изменение и удаление из базы данных. Запросы должны быть сформулированы в терминах предметной области. Среди запросов обязательно должны быть такие, которые будут вызывать срабатывание ограничений целостности. Составить SQL-скрипты для выполнения этих запросов.

Выполнение работы

- Запустить Microsoft Query Analyzer и соединиться с локальной базой данных.
- Выполнить скрипт внесения данных в базу. Сохранить протокол выполнения в файле.
- Выполнить подготовленные выборки. Сохранить протокол выполнения в файле.
- Выполнить подготовленные запросы на модификацию данных. Убедиться в правильном срабатывании ограничений целостности. Выполнить выборки, подтверждающие правильность выполнения модификаций. Сохранить протокол выполнения в файле.
- Закончить работу с Microsoft SQL Server.

Содержание отчета

- Протокол работы в среде Microsoft Query Analyzer.
- Краткие выводы о навыках, приобретенных в ходе выполнения работы.

Контрольные вопросы

- Объяснить, как работают написанные запросы.
- Примеры вопросов по оператору SELECT см. в задании №1.

Дополнительные вопросы

- Исправить неверно работающий запрос (запросы).
- Упростить один или несколько запросов.
- Написать или модифицировать запрос по сформулированному заданию

Лабораторная работа № 4
Создание и использование представлений

Цель работы

Ознакомление со способами создания таблиц в режимах интерактивного SQL и мастеров СУБД Microsoft SQL Server и приобретение навыков работы с представлениями.

Темы для предварительной проработки

- Оператор создания представления языка SQL.

Подготовка к работе

- Подготовить SQL-скрипт для создания представлений, заданных в Вашем варианте индивидуального задания.
- Сформулировать 3-4 запроса на выполнение выборок из представлений и составить соответствующие SQL-скрипты.

Выполнение работы

- Запустить Microsoft Query Analyzer и соединиться с локальной базой данных.
- Выполнить скрипты создания представлений.
- Выполнить подготовленные выборки. Сохранить протокол выполнения в файле.
- Уничтожить созданные представления.
- Запустить Microsoft Enterprise Manager и создать те же представления средствами мастера представлений.
- Выполнить выборки, подтверждающие адекватность представлений.
- Закончить работу с Microsoft SQL Server

Содержание отчета

- Протокол работы в среде Microsoft Query Analyzer.
- Краткие выводы о навыках, приобретенных в ходе выполнения работы.

Теоретические сведения

Представление – это виртуальная таблица, которая сама по себе не существует, но для пользователя выглядит таким образом, как будто она существует. Представление определяется перечнем тех столбцов таблиц и признаками тех их строк, которые хотелось бы в ней увидеть. Представление не поддерживается его собственными физическими хранимыми данными. Вместо этого в каталоге таблиц хранится определение, оговаривающее, из каких столбцов и строк других таблиц оно должно быть сформировано при реализации SQL-предложения на получение данных из представления или на модификацию таких данных. В каталоге БД представление сохраняется в виде того запроса, который его определяет. При трансляции запроса, содержащего обращение к представлению, SQL-машина подставляет вместо него обращение к базовой таблице/таблицам и добавляет те условия, которые определяют представление.

Представление создается с помощью предложения CREATE VIEW:

Список имен столбцов должен быть обязательно определен лишь в тех случаях, когда:

- хотя бы один из столбцов подзапроса не имеет имени (создается с помощью выражения, SQL-функции или константы);
- два или более столбцов запроса имеют одно и то же имя; если же список отсутствует, то представление наследует имена столбцов из запроса.

Во всех выражениях выборки представление может быть использовано вместо таблицы без всяких ограничений. Сложнее обстоит дело с модификацией представления.

По стандарту SQL/92 представления являются изменяемыми, если:

- в списке выборки не указано ключевое слово DISTINCT;
- каждое арифметическое выражение в списке выборки представляет собой одну спецификацию столбца, и спецификация одного столбца не появляется более одного раза;
- в разделе FROM указана только одна таблица, являющаяся либо базовой таблицей, либо изменяемой представляемой таблицей;
- в условии выборки раздела WHERE не используются подзапросы;
- в табличном выражении отсутствуют разделы GROUP BY и HAVING.

Фраза WITH CHECK OPTION в определении представления имеет смысл только в случае определения изменяемой представляемой таблицы, которая определяется спецификацией запроса, содержащей фразу WHERE. При наличии этой фразы не допускаются изменения представляемой таблицы, которые приводят к появлению в базовой таблице строк, не видимых в представляемой таблице (т.е. таких строк, которые не удовлетворяют условию фразы WHERE спецификации запроса). Если WITH CHECK OPTION в определении представления отсутствует, такой контроль не производится. Вычисляемые столбцы могут использоваться в качестве ключей, но не могут вставляться/изменяться.

Изменяемость представлений в DB2 в основном следует стандарту. Однако:

- допускаются подзапросы в выражении WHERE;
- допускается изменение (delete, update) представлений, содержащих UNION ALL.

Изменяемость представлений в Oracle

- допускает подзапросы в выражении WHERE;

- не допускает изменение представлений, содержащих UNION ALL;
- дает частичную возможность изменения соединенных таблиц.

Изменяемость представлений в MSSQL

- допускает подзапросы в выражении WHERE;
- не допускает изменение представлений, содержащих TOP, GROUP BY, UNION;
- дает частичную возможность изменения соединенных таблиц.

Таблица Oracle является сохраняемой ключом (key-preserved), если ключ таблицы может также являться и ключом представления. В простейшем случае - если первичный ключ или уникальное поле входит в представление. Свойство key-preserved не зависит от конкретной конфигурации данных, а только от определений ограничений целостности в соединяемых таблицах.

Контрольные вопросы

1. Объяснить, как работают написанные запросы.
2. Рассказать о CHECK OPTION.
3. Рассказать о модификации данных через представления.
4. Рассказать о вставке данных через представления.
5. Примеры вопросов по оператору SELECT см. в лабораторной работе № 1.

Дополнительные вопросы

1. Исправить неверно работающий запрос (запросы).
2. Упростить один или несколько запросов.
3. Продемонстрировать изменение и вставку данных через представления.
4. Написать или модифицировать запрос по сформулированному заданию.

Лабораторная работа № 5
Управление доступом

Цель работы

Освоение способов управления доступом в средах СУБД Microsoft SQL Server.

Темы для предварительной проработки

- Управление доступом в СУБД.

Подготовка к работе

- Скооперироваться с кем-то из своих товарищей по группе для доступа к таблицам друг друга. Ознакомиться со структурой базы данных своего партнера.
- Подготовить SQL-скрипты для доступа к таблицам своего партнера.

Выполнение работы

- Запустить Microsoft Enterprise Manager и Microsoft Query Analyzer, соединиться с локальной базой данных.
- В среде Microsoft Enterprise Manager был создан пользователь test/test. Необходимо дать ему доступ к локальной базе данных (назначить ему роль уровня базы данных public). Для этого надо в среде Microsoft Query Analyzer выполнить следующую команду T-SQL: exec sp_grantdbaccess 'test'
- В среде Microsoft Query Analyzer выполнить скрипты присвоения новому пользователю прав доступа к таблицам, созданным в лабораторной работе № 2. При этом права доступа к различным таблицам должны быть различными, а именно:

- по крайней мере для одной таблицы новому пользователю присваиваются права SELECT, INSERT, UPDATE в полном объеме
- по крайней мере для одной таблицы новому пользователю присваиваются права SELECT и UPDATE только избранных столбцов.
- по крайней мере для одной таблицы новому пользователю присваивается только право SELECT.

• В среде Microsoft Enterprise Manager присвоить новому пользователю право доступа (SELECT) к представлению, созданному в лабораторной работе № 4.

• В среде Microsoft Enterprise Manager создать стандартную роль уровня базы данных, присвоить ей право доступа (UPDATE на некоторые столбцы) к представлению, созданному в лабораторной работе № 4, назначить новому пользователю созданную роль.

• В среде Microsoft Query Analyzer соединиться с локальной базой данных под именем нового пользователя.

• Выполнить от имени нового пользователя некоторые выборки из таблиц и представления, подготовленные для лабораторных работ №№ 2, 4. Убедиться в правильности контроля прав доступа. Сохранить протокол выполнения в файле.

• Выполнить от имени нового пользователя операторы изменения таблиц с ограниченными правами доступа. Убедиться в правильности контроля прав доступа. Сохранить протокол выполнения в файле.

• Закончить работу с Microsoft SQL Server.

Содержание отчета

• Протокол работы в среде Microsoft Query Analyzer.

• Краткие выводы о навыках, приобретенных в ходе выполнения работы.

Теоретические сведения

Средства управления доступом являются одним из наиболее трудных для описания аспектов функционирования СУБД, так как в этой части рассматриваемые нами СУБД при обеспечении в принципе эквивалентных возможностей используют различные подходы и различную терминологию. Ниже мы пытаемся прежде всего «привести к общему знаменателю» некоторые средства управления доступом в DB2, Oracle и MSSQL с последующим указанием различий между ними.

Аутентификация и авторизация

Доступ к данным и возможностям СУБД персонифицирован. Для каждого действия, выполняемого в системе, определено, каким пользователем оно выполняется. Пользователь (user) - это некоторое имя, определенное в базе данных и связанное с некоторым субъектом, который может соединяться с базой данных и выполнять доступ к ее объектам.

Логически пользователь понимается рассматриваемыми СУБД одинаково, хотя с принципиально различными реализациями этого понятия.

Процедура аутентификации (authentication) состоит в представлении пользователя системе при установлении соединения с базой данных и подтверждении его подлинности. Подтверждение подлинности выполняется паролем (password) (см. рис. 2-4). Процедура аутентификации выполняется при запуске пользователем приложения.

Рис. 2. Аутентификация в DB2

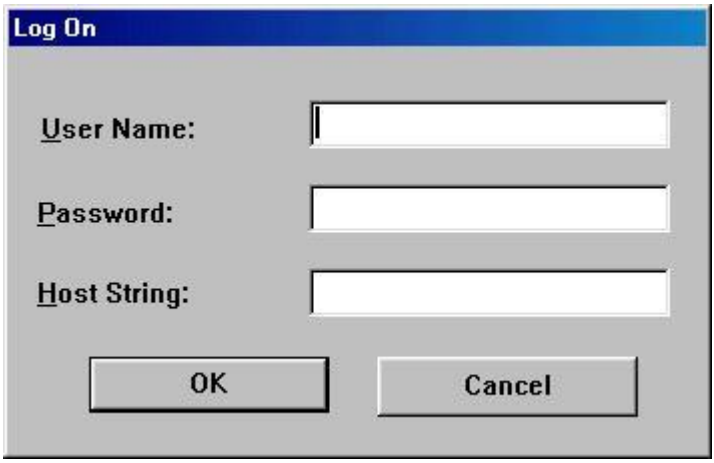
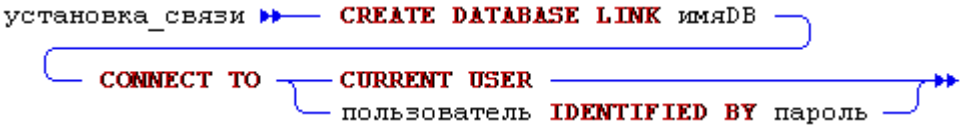


Рис. 3. Аутентификация в Oracle

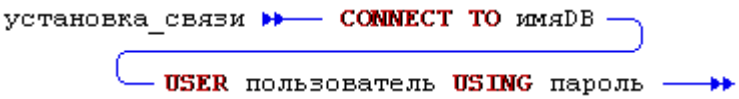
Рис. 4. Аутентификация в MSSQL

Связь с базой данных устанавливается приложением при помощи SQL-оператора (показаны только минимальный наборы необходимых составляющих):

- в Oracle:



- в DB2:



- в MSSQL:

```
CONNECT TO {[имя_сервера.] имя_базы_данных}
[AS имя_соединения]
USER [логин[.пароль] | $integrated]
```

Рассматриваемые СУБД используют термин *привилегии (privileges)*, хотя и с некоторыми разночтениями в его понимании. Описывая общие свойства, мы будем понимать под привилегией некоторый поддерживаемый системой признак, который определяет, разрешение не выполнение какой-либо конкретной операции (общей или относящейся к конкретному объекту). Каждое действие, выполняемое пользователем в базе данных, сопровождается процедурой авторизации *(authorization)*, которая заключается в проверке того, может ли данный пользователь выполнять запрошенное действие над данным объектом, т.е., имеет ли данный пользователь данную привилегию. По результатам авторизации запрос на доступ выполняется или отклоняется.

Методы аутентификации

Для аутентификации могут использоваться внутренние или внешние методы.

При применении внутренних методов данные аутентификации пользователи являются объектами базы данных и их имена и пароли хранятся в СУБД.

При применении внешних методов аутентификация пользователя при соединении с базой данных производится по тому же имени и паролю, которые используются для его аутентификации в операционной системе или сетевых службах. В этом случае явная аутентификация при соединении с базой данных может и не требоваться - для нее используются результаты аутентификации пользователя при начале его сеанса в операционной системе/сети.

Oracle по умолчанию использует внутренний метод. В соответствии с этим, в Oracle пользователь является объектом базы данных, т.е., информация о пользователе хранится в базе данных (в системном каталоге) и пользователи могут создаваться / изменяться / удаляться средствами языка SQL. Ниже приведен синтаксис основной части оператора CREATE USER:

Фраза IDENTIFIED BY пароль – определяет аутентификацию пользователя внутренними методами - с сохранением имени и пароля в базе данных.

Фраза IDENTIFIED EXTERNALLY – определяет аутентификацию внешними средствами.

Фраза IDENTIFIED GLOBALLY AS внешнее_имя – определяет аутентификацию пользователя средствами Oracle Security Service.

Информация о пользователе может также изменяться (ALTER USER) и удаляться (DROP USER).

DB2 использует внешний метод - аутентификацию внешними средствами, например, операционной системой. В связи с этим, пользователь не является объектом базы данных, в базе данных информация сохраняется не о пользователе, а только о данных ему грантах - разрешениях использования привилегий. Поэтому пользователь не управляется средствами языка SQL. При работе с DB2 Command Center может создаться первое впечатление, что мы можем создать нового пользователя так же, как и в Oracle Navigator. Однако, обратите внимание на следующие обстоятельства:

- при добавлении (Add) нового пользователя на странице Database соответствующего Мастера поле имени предоставляет возможность выбора имени пользователя из имен, зарегистрированных в системе;
- в Мастере отсутствует поле для ввода пароля пользователя;
- кнопка 

Мастера не станет доступной, пока не будет дан хотя бы один грант для создаваемого пользователя.

Это происходит потому, что при добавлении пользователя на самом деле в базу данных заносится не информация о пользователе, а только информация о его грантах. Поэтому в DB2 возможна даже такая "странная" ситуация: пользователь создан в базе данных и ему назначены некоторые привилегии, но работать с базой данных он не может, так как, не будучи зарегистрирован в системе, не может пройти аутентификацию.

MSSQL может использовать как аутентификацию средствами операционной системы, так и внутренний метод аутентификации, при котором информация о пользователе хранится в базе данных (в системном каталоге) и пользователи могут создаваться / изменяться / удаляться командами языка T-SQL. Ниже приведен синтаксис команды T-SQL для заведения новой учетной записи пользователя:

```
sp_addlogin [ @loginame = ] 'имя'
   [ , [ @passwd = ] 'пароль' ]
   [ , [ @defdb = ] 'база данных' ]
   [ , [ @deflanguage = ] 'язык' ]
```

Параметр @loginame - определяет имя создаваемой учетной записи.

Параметр @passwd - определяет пароль учетной записи.

Параметр @defdb - определяет базу данных по умолчанию (используется клиентскими средствами для подкличения к ней после аутентификации) для создаваемого пользователя.

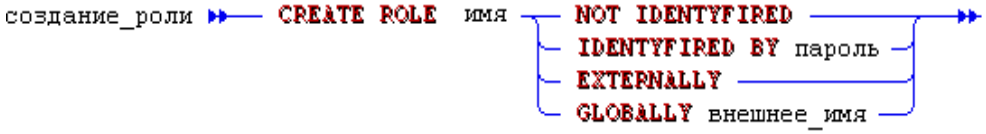
Параметр @deflanguage - определяет используемый по умолчанию язык.

Учетная запись может также удаляться (sp_droplogin [@loginame =] 'имя').

Роли и группы

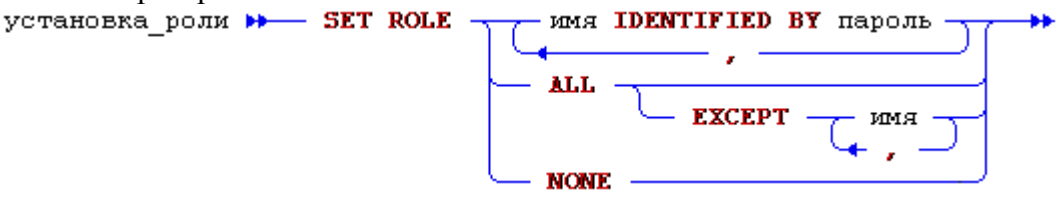
Для сокращения объема информации по управлению доступом и обеспечения более гибких возможностей управления рассматриваемые СУБД применяют группирование привилегий - возможность одним действием администратора предоставить разным пользователям одинакового набора привилегий. Однако, представления концепции группирования различны в наших СУБД.

Oracle использует для этих целей роли. *Роль (role)* - это объект базы данных, представляющий собой именованный набор привилегий, который может предоставляться пользователю или другой роли. Администратор может создавать новые роли и изменять состав ролей. Как объект базы данных, роль могут управляться средствами языка SQL (CREATE ROLE, ALTER ROLE, DROP ROLE). Минимальный синтаксис оператора CREATE ROLE показан ниже:



Применение роли может быть защищено паролем и требовать аутентификации - такой же, как и аутентификация пользователя.

Роли могут применяться динамически в течение сеанса связи с базой данных. Установка / отмена ролей производится оператором SET ROLE:



В Oracle есть несколько предустановленных ролей:

- роли CONNECT, RESOURCE и DBA обеспечиваются для совместимости с предыдущими версиями;

- роли SELECT_CATALOG_ROLE, EXECUTE_CATALOG_ROLE, DELETE_CATALOG_ROLE разрешают доступ к представлениям словаря данных;
- роли EXP_FULL_DATABASE и IMP_FULL_DATABASE разрешают использование утилит импорта и экспорта.

Для тех же целей в DB2 применяются группы пользователей. *Группа (group)* - набор пользователей, которым даются совместные привилегии. Группе могут даваться / отбираться привилегии. Как и в случае с пользователями, объектами базы данных являются гранты, даваемые группе. Создание же групп, равно как и включение в них пользователей выполняется внешними средствами.

Несмотря на столь очевидные различия, роли и группы обеспечивают одинаковую функциональность. Так, если в Oracle роль R составляет некоторый набор привилегий P, связанных с выполнением некоторой производственной функции F, и мы можем назначить пользователя U на роль R, то в DB2 можно создать группу пользователей G, выполняющую ту же функцию F, назначить для группы тот же набор привилегий P и включить пользователя U в группу G.

Однако, в отличие от ролей, группы не являются объектами базы данных и не могут создаваться средствами SQL.

Как и в случае Oracle, для группировки привилегий MSSQL использует понятие роли. Роли могут управляться командами языка T-SQL (sp_addrole, sp_droprole). Минимальный синтаксис команды sp_addrole показан ниже:

sp_addrole [@rolename =] 'название'
[, [@ownername =] 'владелец']

Роли уровня приложений могут быть защищены паролем и требовать аутентификации - такой же, как и аутентификация пользователя.

В MSSQL существует два вида ролей:

- Роли уровня сервера.
- Роли уровня базы данных.

Роли уровня сервера предоставляют различные степени доступа к операциям и задачам сервера баз данных. Они не зависят от конкретных баз данных и модифицировать их нельзя. Существуют следующие типы ролей уровня сервера:

- sysadmin - возможность исполнить любое действие в SQL Server;
- serveradmin - возможность изменить параметры SQL Server и завершить его работу;
- setupadmin - возможность инсталлировать систему репликации и управлять выполнением расширенных хранимых процедур;
- securityadmin - возможность контролировать параметры учетных записей для подключения к серверу и предоставлять права доступа к базам данных;
- processadmin - возможность управлять ходом процессов в MSSQL Server;
- dbcreator - возможность создавать и модифицировать базы данных;
- diskadmin - возможность управлять файлами баз данных на диске.

Роли уровня базы данных позволяют назначить набор прав доступа для работы с конкретной базой данных отдельному пользователю или группе. В MSSQL существует три типа таких ролей:

- Заранее определенные роли.
- Определяемые пользователем роли.

Существует следующий набор заранее определенных ролей уровня базы данных:

- db_owner - имеет полный доступ ко всем объектам базы данных;
- db_accessadmin - осуществляет контроль за доступом к базе данных путем добавления и удаления пользователей;
- db_datareader - имеет полный доступ к выборке данных;
- db_datawriter - может выполнять операторы INSERT, DELETE и UPDATE для любой таблицы базы данных;
- db_ddladmin - имеет возможность создавать, модифицировать и удалять объекты базы данных;
- db_securityadmin - управляет системой безопасности базы данных;
- db_backupoperator - позволяет создавать резервные копии базы данных;
- db_denydatareader - отказывает в разрешении на выполнение оператора SELECT для всех таблиц базы данных;
- db_denydatawriter - отказывает в разрешении на выполнение операторов модификации данных (INSERT, DELETE, UPDATE) для любых таблиц базы данных;
- public - каждому пользователю при открытии доступа к базе данных автоматом назначается роль public.

Роли, определяемые пользователем, позволяют группировать пользователей и назначать каждой группе конкретную функцию безопасности. Эти роли зависят от базы данных, в то время как роли уровня сервера зависят от сервера баз данных.

Более полную информацию по работе системы безопасности MSSQL можно получить из [1].

Уровни объектов и привилегии

Как мы сформулировали выше, в широком понимании привилегия - это разрешение на выполнение конкретной операции над конкретным объектом. Однако, объекты, на которые распространяются привилегии имеют сложную структуру, показанную на рис. 5:

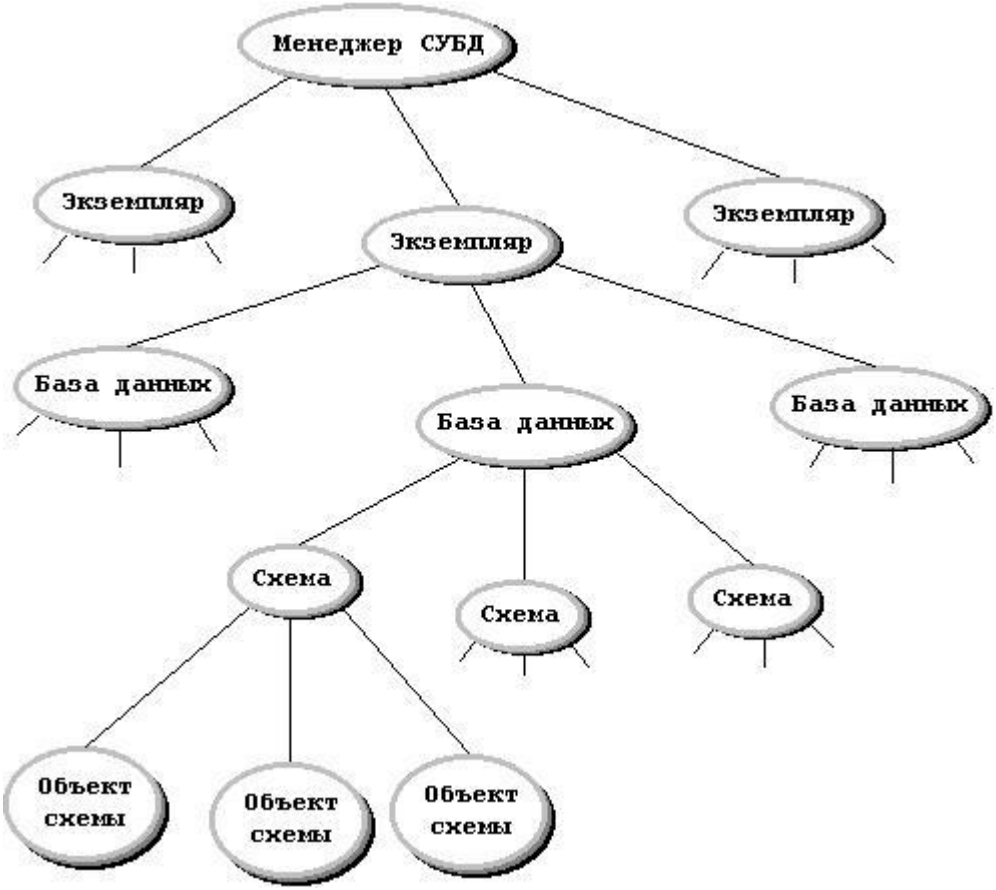


Рис. 5. Структура объекта, на которые распространяются привилегии

Менеджер СУБД (database manager) является собственно программным кодом СУБД. Для каждого

менеджера может быть создан один или несколько экземпляров.

Экземпляр (instance) менеджера СУБД - логическая среда, которая характеризуется своими ресурсами и параметрами. Экземпляр, однако, по-разному понимается разными СУБД.

С точки зрения Oracle экземпляр - это среда времени выполнения, его ресурсы составляют набор выполняющихся процессов Oracle и область оперативной памяти (SGA - system global area). В каждом экземпляре Oracle «монтируется» и «открывается» одна база данных и экземпляр выполняет функции монитора этой базы данных.

DB2 и MSSQL понимают под экземпляром логическую среду, включающую в себя несколько баз данных и постоянно хранимые параметры. Экземпляр, таким образом, является монитором нескольких баз данных. Экземпляры выполняются совершенно независимо друг от друга. Такой механизм позволяет осуществлять раздельное администрирование и настройку параметров экземпляров, повышает безопасность и надежность обработки данных.

На понятии *базы данных (datebase)* рассматриваемые СУБД опять сходятся. Базой данных называется коллекция данных, рассматриваемая как единое целое. База данных характеризуется занимаемой ею памятью (табличными пространствами) и каталогом. В каждой базе данных (с позиций DB2 и Oracle) может быть создано несколько схем. MSSQL же не поддерживает выделения в базе данных схем. С точки зрения СУБД MSSQL база данных является единым (неделимым) хранилищем данных.

Схема (schema) с точки зрения СУБД DB2 и Oracle - набор объектов базы данных. Каждая схема связывается с определенным пользователем (т.е., это та схема данных, которую видит данный пользователь), и имя схемы совпадает с именем пользователя. По умолчанию каждый пользователь работает с объектами в своей схеме и создает объекты в своей схеме. Однако, пользователь может (если это ему позволено)

иметь доступ и к объектам в другой схеме или менять схему, в которой он работает. В MSSQL (как было замечено выше) понятие схемы отсутствует. Область видимости объектов базы данных для каждого пользователя формируется путем раздачи соответствующих прав на ее ресурсы.

Объекты схемы (schema objects) - таблицы, представления, индексы и т.д. - скалярные (не составные) объекты базы данных.

Рассматриваемые нами СУБД по разному подходят к назначению привилегий контейнерных и скалярных объектов. В DB2 привилегии, относящиеся к экземпляру и, частично, к базе данных, выделены в отдельное понятие полномочий. Прочие привилегии, как для контейнерных, так и для скалярных объектов, также различаются.

Oracle объединяет привилегии, относящиеся ко всем контейнерным объектам, в понятие системные привилегии, а все остальные - в привилегии объектов схемы.

На уровне MSSQL существует два типа прав доступа (привилегий): объектные и командные. Объектные права доступа определяют, кто может получать доступ и работать с данными в таблицах и представлениях и кто может запускать хранимые процедуры. Командные права доступа определяют, кто может удалять и создавать объекты в базе данных.

DB2: полномочия и привилегии

В DB2 разрешения пользователю выполнять те или иные действия делятся на полномочия и привилегии. *Полномочие (authority)* является методом группирования привилегий и возможности управлять менеджером базы данных и системными утилитами. Полномочия, связанные с базой данных, записываются в каталог соответствующей базы данных, полномочия, связанные с управлением системой записываются в файл конфигурации менеджера СУБД для данного экземпляра.

Полномочия в DB2 являются предустановленными. Имеются следующие виды полномочий

Полномочие SYSADM - администрирования системы - является высшим уровнем полномочий и обеспечивает над всеми ресурсами, которыми управляет менеджер базы данных. Полномочие SYSADM включает в себя также полномочия DBADM (для всех баз данных в экземпляре), SYSCTRL иSYSMAINT, а также разрешение давать и забирать полномочия DBADM. Пользователь с полномочием SYSADM имеет полный доступ ко всем объектам всех баз данных в данном экземпляре.

Полномочие DBADM - администрирования базы данных - определяется для отдельной базы данных. Оно включает в себя разрешение на полный доступ ко всем объектам этой базы данных посредством операторов SQL, на создание объектов и на выполнение команд базы данных. Полномочие DBADM также включает в себя возможность давать и забирать привилегию CONTROL и другие привилегии.

Полномочие SYSCTRL - управления системой - высший уровень управления системой, применяется только к системным ресурсам. Оно не дает разрешения на прямой доступ к данным, но позволяет создавать базы данных, изменять их конфигурацию и уничтожать их, создавать и уничтожать табличные пространства. Полномочие SYSCTRL включает в себя также полномочияSYSMAINT.

Полномочие SYSMAINT - сопровождения системы - второй уровень управления системой. Оно дает возможность выполнять опебрации по сопровождению всех баз данных в данном экземпляре: изменять конфигурацию баз данных, выполнять их копирование/восстановление, и управлять базами данных. Оно не дает разрешения на прямой доступ к данным.

Полномочия SYSADM, SYSCTRL, SYSMAINT даются группам, имена которых определяются в конфигурационных параметрах sysadm_group, sysctrl_group, sysmaint_group соответственно. Членство в этих группах управляется вне менеджера базы данных - средствами безопасности той платформы, на которой функционирует СУБД.

Привилегия (privilege) дает пользователю или группе право выполнять определенный вид доступа к определенному объекту. В DB2 имеется несколько видов привилегий, некоторые из них рассматриваются ниже.

Привилегии базы данных - включают в себя действия над базой данных в целом, в том числе привилегии:

- CONNECT - разрешение на доступ к базе данных;
- BINDADD - разрешение на создание новых пакетов в базе данных;
- CREATETAB - разрешение на создание новых таблиц в базе данных.

Привилегии схемы - включают в себя действия над определенной схемой в базе данных, в том числе привилегии:

- CREATEIN - разрешение на создание объектов в схеме;
- ALTERIN - разрешение на изменение объектов в схеме;
- DROPIN - разрешение на уничтожение объектов в схеме.

Указанные привилегии могут даваться с правом передачи или без него.

Все остальные виды привилегий являются объектными привилегиями - они даются по отношению к отдельным объектам базы данных.

Привилегии таблиц и представлений - включают в себя разрешение на действия над определенными таблицами / представлениями, в том числе привилегии:

- CONTROL - привилегия владельца, включает в себя все другие привилегии для таблицы / представления, а также разрешения удалять ее и давать / забирать привилегии для данной таблицы / представления другим пользователям / группам. Создатель таблицы автоматически получает для нее привилегию CONTROL, создатель представления получает привилегию CONTROL только в том случае, если он имеет эту привилегию для всех таблиц, входящих в

определение представления. Привилегия CONTROL может быть дана / отобрана пользователем с полномочиями SYSADM или DBADM (это справедливо и для всех объектных привилегий, описанных ниже).

- ALL - все привилегии, перечисленные ниже.
- ALTER - разрешение на изменение таблицы (добавление столбцов, добавление / изменение / уничтожение ограничений целостности, создание триггеров для таблицы). Однако, для выполнения таких изменений, которые затрагивают другие таблицы / представления, пользователь должен иметь также соответствующие привилегии по отношению к этим другим таблицам / представлениям.
- DELETE - разрешение на удаление строк из таблицы / представления.
- INDEX (только таблица) - разрешение на создание индекса к таблице.
- INSERT - разрешение на вставку строк в таблицу / представление и выполнение утилиты IMPORT.
- REFERENCES (только таблица) - разрешение на создание внешних ключей, ссылающихся на данную таблицу. Эта привилегия может быть ограничена выбранными столбцами.
- SELECT - разрешение на выборку строк из таблицы / представления, создание представления на базе данной таблицы / представления таблицы и выполнение утилиты EXPORT.
- UPDATE - разрешение на изменение значений в столбцах таблицы / представления. Эта привилегия может быть ограничена выбранными столбцами.

Указанные привилегии могут даваться с правом передачи или без него.

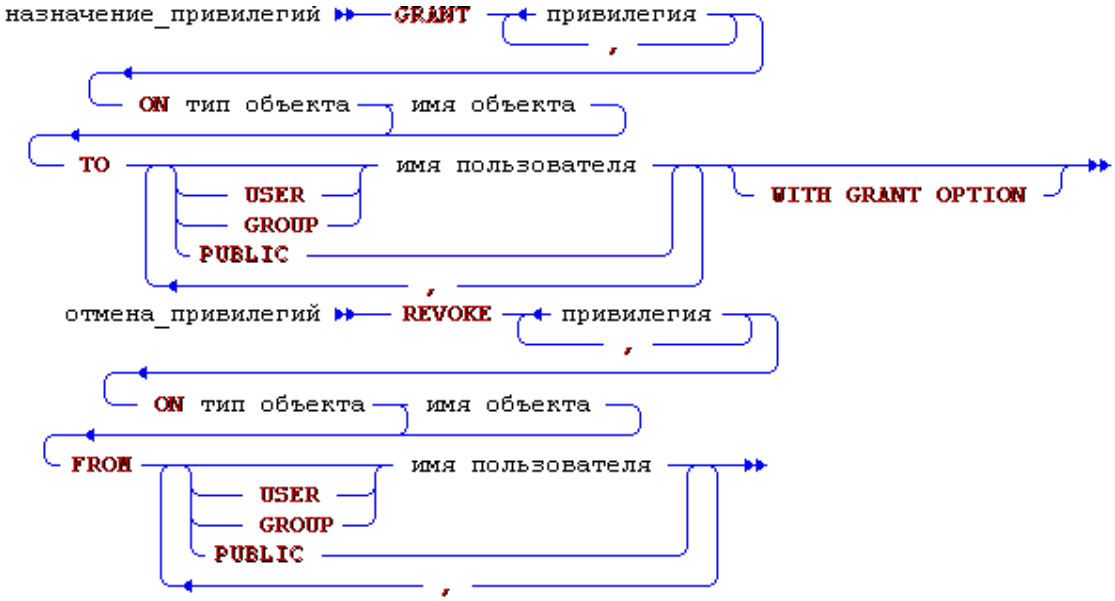
Привилегии пакета - включают в себя разрешение на действия над определенными пакетами - объектами базы данных, содержащими информацию, используемую

менеджером базы данных для эффективного доступа к данным из определенного приложения. Привилегии пакета следующие.

- CONTROL - привилегия владельца, включает в себя все другие привилегии пакета, а также разрешение удалять его и давать / забирать привилегии для него другим пользователям / группам.
- BIND - разрешение на пересоздание пакета.
- EXECUTE - разрешение на выполнение пакета.

Привилегии пакета - создатель индекса автоматически получает для него привилегию CONTROL, что дает ему разрешение на удаление индекса.

Все названные выше привилегии назначаются операторами GRANT, а забираются операторами REVOKE, обобщенная форма которых показана ниже:



Примечания:

- *типы объектов* - DATABASE, SCHEMA, TABLE, VIEW и т.д.;

- *имя объекта* не указывается для *типа объекта* DATABASE;
- набор возможных привилегий зависит от *типа объекта*;
- опция WITH GRANT OPTION означает разрешение на передачу привилегий и возможна только для *типов объектов* TABLE, VIEW, SCHEMA.

Oracle: системные и объектные привилегии

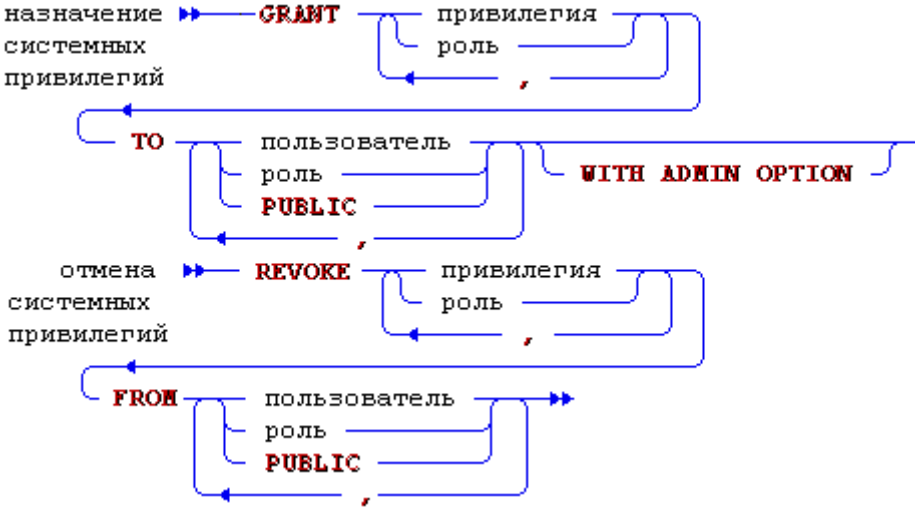
В Oracle все даваемые пользователю разрешения сводятся к привилегиям. Различаются два вида привилегий - *системные привилегии (system privileges)* и *привилегии объектов схемы (schema object privileges)* или просто объектные. Управление обоими видами привилегий производится одинаковым образом. Привилегии могут даваться / забираться пользователю или роли.

Системные привилегии дают пользователю возможность выполнять общесистемные действия или определенные действия над всеми объектами определенного типа. Существует более 40 системных привилегий, которые можно разделить на несколько основных групп:

- привилегии, связанные с выполнением некоторых общесистемных действий (аналог полномочий в DB2), например:
  - SYSOPER - разрешение на запуск и останов Server Manager, изменение конфигурации баз данных, копирование и восстановление.
  - SYSDBA - то же, плюс разрешение на создание баз данных.
  - CREATE USER, CREATE ROLE - разрешение на создание пользователей, ролей.
  - CREATE SESSION - разрешение на соединение с базой данных.
- привилегии, связанные с выполнением определенных действий над всеми объектами определенного типа, например:

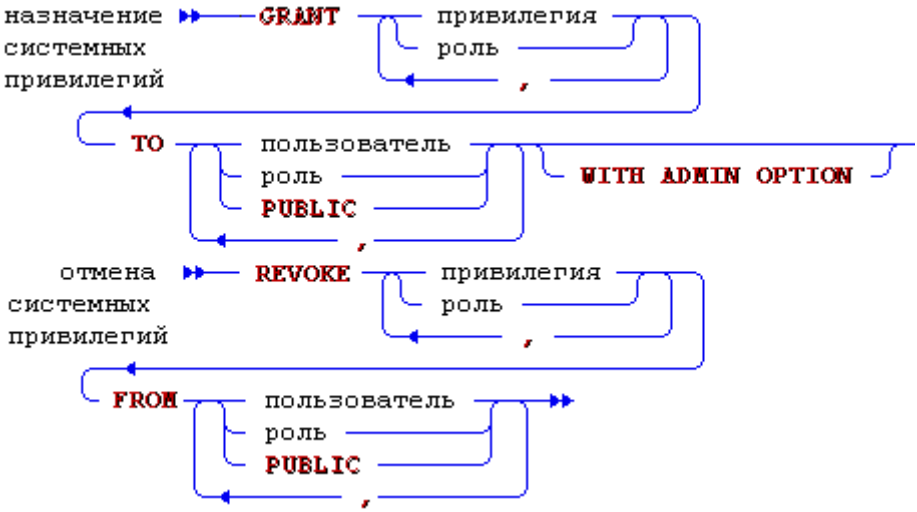
  - CREATE ANY TABLE, CREATE ANY VIEW, CREATE ANY INDEX... - разрешение на создание любых таблиц, представлений, индексов ... в любых схемах.
  - ALTER ANY TABLE, ALTER ANY INDEX... - разрешение на изменение любых таблиц, индексов ... в любых схемах.
  - DROP ANY TABLE, DROP ANY VIEW, DROP ANY INDEX... - разрешение на уничтожение любых таблиц, представлений, индексов ... в любых схемах.
- привилегии, связанные с выполнением определенных действий в определенной схеме, например:
  - CREATE TABLE, CREATE VIEW... - разрешение на создание таблиц, представлений ... в определенной схеме.

Системные привилегии могут даваться с правом передачи другому пользователю / роли или без него. Они назначаются / забираются GRANT / REVOKE:



WITH ADMIN OPTION определяет возможность передачи привилегий.

Объектные привилегии концептуально совпадают с таковыми в DB2, они разрешают пользователю выполнять определенные действия над определенным объектом. Основными объектами, для которых даются объектные привилегии являются таблицы и представления, привилегии для них - ALTER, INDEX, INSERT, REFERENCES, SELECT, UPDATE, ALL - как и в DB2. Указанные привилегии могут даваться с правом передачи или без него. Операторы GRANT / REVOKE для объектных привилегий имеют следующий синтаксис:



MSSQL: объектные и командные права доступа

Как было сказано выше, в MSSQL все права доступа подразделяются на две категории: объектные и командные.

Объектные права доступа позволяют контролировать права доступа для таблиц, столбцов таблиц, представлений и хранимых процедур. Существуют следующие типы объектных прав доступа (табл. 3).

Таблица 3

Типы объектных прав доступа

Тип объекта	**Возможные команды**
Таблица	SELECT, UPDATE, DELETE, INSERT, REFERENCE
Столбец	SELECT, UPDATE
Представление	SELECT, UPDATE, DELETE, INSERT
Хранимая процедура	EXECUTE

Ниже приведены команды T-SQL, которые используются для управления разрешениями на работу с объектами:

Назначение прав:

```
GRANT
   { ALL [ PRIVILEGES ] | разрешение [ ,...n ] }
   {
      [ ( столбец [ ,...n ] ) ] ON { таблица | представление }
      | ON { таблица | представление } [ ( столбец [ ,...n ] ) ]
      | ON { хранимая_процедура | расширенная_процедура }
      | ON { определенныя_пользователем_функция }
   }
TO учетная_запись [ ,...n ]
[ WITH GRANT OPTION ]
[ AS { группа | роль } ]
```

Отмена прав:

```
REVOKE [ GRANT OPTION FOR ]
   { ALL [ PRIVILEGES ] | разрешение [ ,...n ] }
   {
      [ ( столбец [ ,...n ] ) ] ON { таблица | представление }
      | ON { таблица | представление } [ ( столбец [ ,...n ] ) ]
      | ON { хранимая_процедура | расширенная_процедура }
      | ON { определенныя_пользователем_функция }
```

}
{ TO | FROM }
учетная_запись [,...n]
[CASCADE]
[AS { группа | роль }]

Командные права доступа определяют, кто может выполнять административные действия. Командные права могут быть назначены только системным администраторам, пользователям, которым назначена роль sysadmin, или владельцам баз данных. Ниже приведены командные права доступа, которые можно предоставить или аннулировать:

- CREATE DATABASE - право создания базы данных.
- CREATE DEFAULT - право создания стандартного значения для столбца таблицы.
- CREATE PROCEDURE - право создания хранимой процедуры.
- CREATE RULE - право создания правила для столбца таблицы.
- CREATE TABLE - право создания таблицы.
- CREATE VIEW - право создания представления.
- BACKUP DATABASE - право создания резервной копии базы данных.
- BACKUP TRANSACTION - право создания резервной копии журнала транзакций.

Ниже приведены команды T-SQL, которые используются для управления командными правами доступа:

Назначение прав:
GRANT { ALL | оператор [,...n] }
TO учетная_запись [,...n]

Отмена прав:
REVOKE { ALL | оператор [,...n] }
TO учетная_запись [,...n]

Контрольные вопросы

1. Рассказать об аномалиях доступа к БД.
2. Перечислить аномалии, возникающие на каждом из уровней изолированности.
3. Рассказать о свойствах транзакций.
4. Рассказать об управлении транзакциями.
5. Что такое тупики? Как бороться с тупиками?
6. На каком уровне изолированности возможны тупики?
7. Как обеспечивается изолированность транзакций в СУБД?
8. Как бороться с проблемой фантомов?
9. Что такое гранулированные блокировки?
10. Как можно избежать блокировки при конфликте Read-Write?
11. Что такое журнал транзакций?
12. Как обеспечивается постоянство хранения (durability) в СУБД?

Дополнительные вопросы

1. Продемонстрировать откат транзакции при возникновении ошибок.
2. Продемонстрировать возникновение тупика.
3. Исправить неверные сценарии проверки аномалий.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кузнецов, С. Д. Основы баз данных [Текст] / С. Д. Кузнецов. – М.: Бином. Лаборатория знаний, Интернет-университет информационных технологий, 2007.

2. Microsoft SQL Server 2005 Books Online. [Электронный ресурс] – Режим доступа: http://msdn.microsoft.com/en-us/library/ms130214%28v=sql.90%29.aspx

3. Гарсиа-Молина, Г. Системы баз данных. Полный курс [Текст] / Г. Гарсиа-Молина, Дж. Ульман, Д. Уидом. – М.: Вильямс, 2004.

4. Ульман, Дж. Основы реляционных баз данных [Текст] / Дж. Ульман, Д. Уидом. – М.: Лори, 2006.

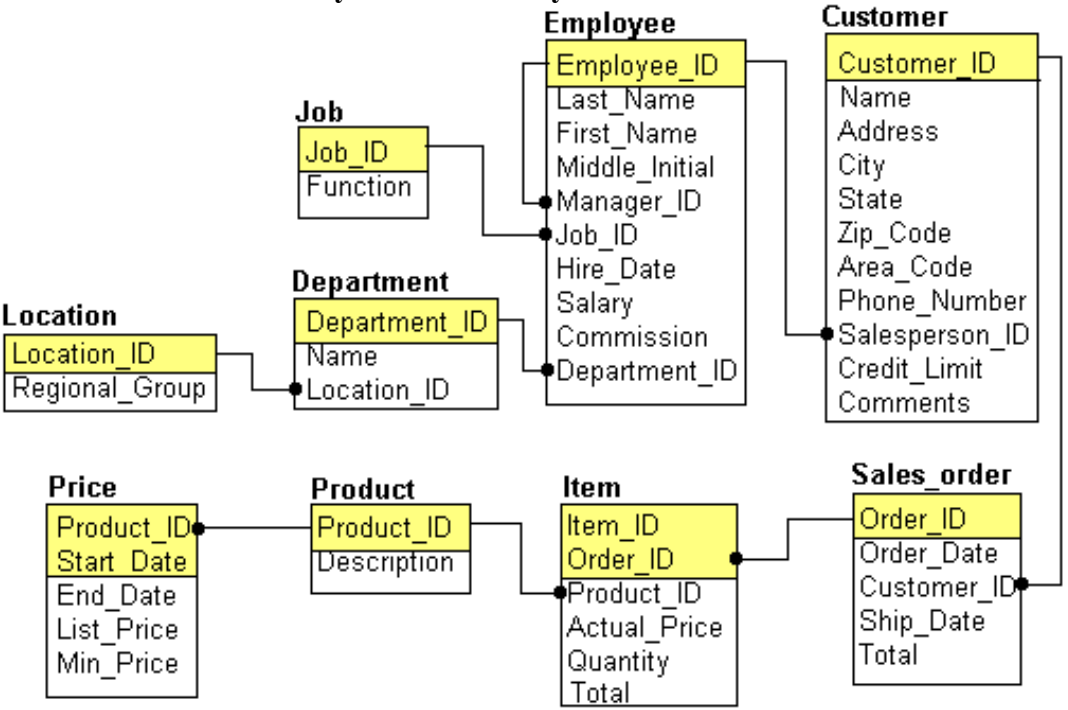
5. Коннолли, Т. Базы данных: проектирование, реализация и сопровождение. Теория и практика [Текст] / Т. Коннолли, К. Бегг. – М.: Вильямс, 2003.

6. Станек У. Р. Microsoft SQL Server 2005. Справочник администратора [Текст] / У. Р. Станек. – М.: Русская редакция, 2008.

7. Нильсен, П. SQL Server 2005. Библия пользователя [Текст] /П. Нильсен. – М.: Вильямс, 2008.

ПРИЛОЖЕНИЕ 1

Концептуальная схема учебной базы данных



ПРИЛОЖЕНИЕ 2

Таблицы базы данных

N пп	Имя столбца	Тип данных Oracle	Тип данных DB2(MSSQL)	Комментарий
Таблица EMPLOYEE - сотрудники фирмы				
1	employee_id	NUMBER(4,0)	SMALLINT	Код сотрудника
2	last_name	VARCHAR2(15)	VARCHAR(15)	Фамилия
3	first_name	VARCHAR2(15)	VARCHAR(15)	Имя
4	middle_initial	VARCHAR2(1)	VARCHAR(1)	Средний инициал
5	manager_id	NUMBER(4,0)	SMALLINT	Код начальника
6	job_id	NUMBER(3,0)	SMALLINT	Код должности
7	hire_date	DATE	DATE(DATETIME)	Дата поступления в фирму
8	salary	NUMBER(7,2)	NUMERIC(7,2)	Зарплата
9	commission	NUMBER(7,2)	NUMERIC(7,2)	Комиссионные
10	department_id	NUMBER(2,0)	SMALLINT	Код отдела
CREATE TABLE EMPLOYEE (employee_id SMALLINT NOT NULL PRIMARY KEY,				

```
last_name VARCHAR(15),
first_name VARCHAR(15),
middle_initial VARCHAR(1),
manager_id SMALLINT,
job_id SMALLINT,
hire_date DATETIME,
salary NUMERIC(7,2),
commission NUMERIC(7,2),
department_id SMALLINT  );
```

Таблица DEPARTMENT - отделы фирмы

1	department_id	NUMBER(2,0)	SMALLINT	Код отдела
2	name	VARCHAR2(14)	VARCHAR(14)	Название отдела
3	location_id	NUMBER(3,0)	SMALLINT	Код места размещения

```
CREATE TABLE DEPARTMENT (
 department_id SMALLINT NOT NULL PRIMARY KEY,
 name VARCHAR(14),
 location_id SMALLINT );
```

Таблица LOCATION - места размещения отделов				
1	location_id	NUMBER(3,0)	SMALLINT	Код места размещения
2	regional_group	VARCHAR2(20)	VARCHAR(20)	Город

```
CREATE TABLE LOCATION (
  location_id SMALLINT NOT NULL PRIMARY KEY,
  regional_group VARCHAR(20) );
```

Таблица JOB - должности в фирме				
1	job_id	NUMBER(3,0)	SMALLINT	Код должности
2	function	VARCHAR2(30)	VARCHAR(30)	Название должности

```
CREATE TABLE JOB  (
  job_id SMALLINT NOT NULL PRIMARY KEY,
  [function] VARCHAR(30) );
```

Таблица CUSTOMER - фирмы-покупатели				
1	customer_id	NUMBER(6,0)	INTEGER	Код покупателя
2	name	VARCHAR2(45)	VARCHAR(45)	Название покупателя
3	address	VARCHAR2(40)	VARCHAR(40)	Адрес
4	city	VARCHAR2(30)	VARCHAR(30)	Город
5	state	VARCHAR2(2)	VARCHAR(2)	Штат
6	zip_code	VARCHAR2(9)	VARCHAR(9)	Почтовый код
7	area_code	NUMBER(3,0)	SMALLINT	Код региона
8	phone_number	NUMBER(7,0)	INTEGER	Телефон
9	salesperson_id	NUMBER(4,0)	SMALLINT	Код сотрудника-продавца, обслуживающего данного покупателя
10	credit_limit	NUMBER(9,2)	NUMERIC(9,2)	Кредит для покупателя
11	comments	LONG	VARCHAR(500)	Примечания

```
CREATE TABLE CUSTOMER (
  customer_id INTEGER NOT NULL PRIMARY KEY,
  name VARCHAR(45),
  address VARCHAR(40),
```

```
  city VARCHAR(30),
  state VARCHAR(2),
  zip_code VARCHAR(9),
  area_code SMALLINT,
  phone_number INTEGER,
  salesperson_id SMALLINT,
  credit_limit NUMERIC(9,2),
  comments VARCHAR(500) );
```

Таблица SALES_ORDER - договоры о продаже

1	order_id	NUMBER(4,0)	SMALLINT	Код договора
2	order_date	DATE	DATE(DATETIME)	Дата договора
3	customer_id	NUMBER(6,0)	INTEGER	Код покупателя
4	ship_date	DATE	DATE(DATETIME)	Дата поставки
5	total	NUMBER(8,2)	NUMERIC(8,2)	Общая сумма договора

```
CREATE TABLE SALES_ORDER (
  order_id SMALLINT NOT NULL PRIMARY KEY,
  order_date DATETIME,
  customer_id INTEGER,
  ship_date DATETIME,
  total NUMERIC(8,2) );
```

Таблица ITEM - акты продаж				
1	order_id	NUMBER(4,0)	SMALLINT	Код договора, в состав которого входит акт
2	item_id	NUMBER(4,0)	SMALLINT	Код акта
3	product_id	NUMBER(6,0)	INTEGER	Код продукта
4	actual_price	NUMBER(8,2)	NUMERIC(8,2)	Цена продажи
5	quantity	NUMBER(8,0)	INTEGER	Количество
6	total	NUMBER(8,2)	NUMERIC(8,2)	Общая сумма

```
CREATE TABLE ITEM (
  order_id SMALLINT NOT NULL,
  item_id SMALLINT NOT NULL,
  product_id INTEGER,
  actual_price NUMERIC(8,2),
  quantity INTEGER,
  total NUMERIC(8,2),
PRIMARY KEY (order_id,item_id)
```

Таблица PRODUCT - товары				
1	product_id	NUMBER(6,0)	INTEGER	Код продукта
2	description	VARCHAR(30)	VARCHAR(30)	Название продукта

```
CREATE TABLE PRODUCT (
  product_id INTEGER NOT NULL PRIMARY KEY,
  description VARCHAR(30) );
```

Таблица PRICE – цены				
1	product_id	NUMBER(6,0)	INTEGER	Код продукта
2	list_price	NUMBER(8,2)	NUMERIC(8,2)	Объявленная цена
3	min_price	NUMBER(8,2)	NUMERIC(8,2)	Минимально возможная цена
4	start_date	DATE	DATE(DATETIME)	Дата установления цены
5	end_date	DATE	DATE(DATETIME)	Дата отмены цены

```
CREATE TABLE PRICE (
  product_id INTEGER NOT NULL,
  list_price NUMERIC(8,2),
  min_price NUMERIC(8,2),
  start_date DATETIME NOT NULL,
  end_date DATETIME,
  PRIMARY KEY (product_id,start_date)
);
```

СОДЕРЖАНИЕ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к лабораторным работам № 1–5 по дисциплинам
«Основы построения защищенных СУБД»,
«Безопасность систем баз данных»
для студентов специальностей 090301
«Компьютерная безопасность»,
090303 «Информационная безопасность
автоматизированных систем»
очной формы обучения

Составитель
Плотников Денис Геннадьевич

В авторской редакции

Подписано к изданию 20.04.2015
Уч.- изд. л. 3,4

ФГБОУВПО «Воронежский государственный
технический университет»
394026 Воронеж, Московский просп., 14