Государственное образовательное учреждение высшего профессионального образования
"Нижегородский государственный архитектурно-строительный университет" (ННГАСУ)

Кафедра информационных систем и технологий

ИСПОЛЬЗОВАНИЕ ЯЗЫКА СТРУКТУРИРОВАННЫХ ЗАПРОСОВ SQL

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к расчетной работе для студентов направлений 230200.62 Информационные системы, 080500.62 Менеджмент и специальности 080502.65 Экономика и управление на предприятии (в строительстве)

Нижний Новгород
2010

УДК681.3
ИСПОЛЬЗОВАНИЕ ЯЗЫКА СТРУКТУРИРОВАННЫХ ЗАПРОСОВ SQL. Методические указания к расчетной работе для студентов направлений 230200.62 Информационные системы, 080500.62 Менеджмент и специальности 080502.65 Экономика и управление на предприятии (в строительстве)
Нижний Новгород, ННГАСУ, 2010

Методические указания содержат краткое описание языка структурированных запросов SQL , варианты заданий и состав расчетной работы по SQL.

Составили: канд.техн.наук, доцент А.Я. Лахов,
канд.техн.наук, доцент К.А. Сафонов
Редактор: д-р.физ.-мат.наук, профессор А.Н. Супрун

© ННГАСУ, 2010

Введение

Общие сведения

В настоящее время информационные системы (ИС) нашли широкое применение во всех отраслях промышленности и экономики. Одними из важнейших составных частей ИС являются базы данных (БД) и системы управления базами данных (СУБД).

БД – совокупность специальным образом организованных данных, хранимых в памяти вычислительной системы, отображающих состояние объектов и их взаимосвязей в рассматриваемой предметной области.

СУБД – это комплекс языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.

В функции этих компонентов ИС входят:

1) хранение и модификация больших объемы информации,
2) обеспечение возможности поиска информации по различным критериям,
3) обеспечение возможности реорганизации информации,
4) предоставление информации в удобной форме.

Принято различать настольные и серверные СУБД. Настольные размещаются, как правило, на одном компьютере. Серверные СУБД основаны на архитектуре клиент/сервер и размещаются на разных компьютерах. Сервер базы данных используется и для хранения информации, и для обработки запросов к БД. Запросы клиентской рабочей станции обрабатываются сервером базы данных и обратно возвращаются только результаты выполнения запроса. Такой подход уменьшает трафик в сети. Обработка запросов сервером базы данных, как правило, осуществляется быстрее, чем на рабочей станции, так как:

1) в качестве сервера, как правило, используется более мощный компьютер,
2) СУБД, используемая в качестве сервера базы данных, обладает более мощными средствами обработки данных.

В ряде случаев клиентская и серверная части являются компонентами одной СУБД, например SQL Plus и Oracle. В других случаях, в качестве клиентской части используется приложение, написанное, например на C++, а в качестве сервера базы данных – мощная СУБД, например Oracle.

В качестве серверов баз данных в большинстве случаев используются реляционные СУБД, для которых стандартным языком доступа к данным является структурированный язык запросов SQL. Настоящие методические указания посвящены изучение этого языка.

1 Реляционные базы данных

Модели данных – средство определения структур данных и процессов над этими данными. Типы структур данных различают по правилам композиции структур данного типа из структур другого типа и правилам обработки структур данного типа.

Различают иерархическую, сетевую, реляционную, многомерную и объектно-ориентированную модели данных. В иерархической модели для описания данных используют древовидную структуру. В сетевой модели для описания используют графы общего вида. В многомерной модели данные представляются в виде многомерных кубов. Объектно-ориентированная модель основана на теории объектно-ориентированного программирования.

Реляционной считается такая база данных, в которой все данные представлены в виде двумерных таблиц, связанных между собой, и все операции над базой данных сводятся к манипуляциям с таблицами. Каждая таблица должна иметь уникальное имя: Студент, Кадры, Товары и др. Эти таблицы на этапе проектирования БД называются сущностями. Если данные меняются в одной таблице, то они автоматически меняются во всех таблицах, связанных с первой.

При использовании реляционной модели данных БД – это совокупность двумерных таблиц для хранения той или иной информации.

Основы теории реляционных баз данных заложил британский учёный Эдгар Кодд. Также им определен математический аппарат реляционной алгебры, на котором основываются реляционные БД.

К свойствам реляционной БД относят:

1) каждая таблица состоит из строк и столбцов;
2) каждая таблица имеет имя, уникальное внутри БД;
3) все элементы данного столбца однородны (как, например, только текст, только дата, только число и т.д.).

При проектировании БД используют следующие термины: поле (столбец), запись (строка), отношение (таблица), первичный и внешний ключ.

Поле – наименьшая, поименованная единица данных к которой можно непосредственно обращаться (используется для представления атрибутов). Значение поля может относиться к различным типам данных: числовому, символьному, дата/время и т.д.

Запись – поименованная совокупность полей (используется для представления сущности). Например, сущность "Товар" может иметь запись из следующих полей и их значений: "Наименование товара" – Стол, "Цена" – 2850, "Количество" – 26 и т.п.

Отношение – совокупность экземпляров записей определенного типа.

Отношения обладают следующими свойствами:

1) Значения атрибутов атомарны, т.е. их нельзя расчленить на несколько значений. Например, если требуется атрибут "Стол" разделить на "Стол письменный" и "Стол обеденный", то следует создать 2 записи.

2) Каждая запись в отношении уникальна, т.е. не существует двух записей с полностью совпадающим набором значений его атрибутов.
3) Каждый атрибут имеет уникальное имя в пределах отношения.
4) Последовательность атрибутов в отношении произвольная.
5) Последовательность записей в отношении произвольная.

Первичный ключ – это минимальный набор атрибутов, уникально идентифицирующий сущность. Другими словами – первичный ключ задает индивидуальное значение каждой записи в таблице. Например, если в некотором текстовом поле отношения "Товары" каждая запись поля "Наименование товара" имеет уникальное значение, например "Стол письменный", то атрибут " Наименование товара" может являться первичным ключом. В противном случае для идентификации записей первичный ключ может состоять из нескольких атрибутов. Например, "Наименование товара" и "Группа товаров".

Связи между таблицами в реляционной модели осуществляются за счет внешних ключей. Атрибут, значения которого ссылаются на значения первичного ключа другого отношения, называется **внешним ключом.** Например, таблицы "Сотрудник" и "Руководитель" (см. рисунок) имеют связь Многие-к-Одному. В этой связи участвуют первичный ключ "Номер" из таблицы "Руководитель" и атрибут "Номер руководителя" из таблицы "Сотрудник", который может принимать только те значения, которые существуют в указанном первичном ключе. Поэтому атрибут " Номер руководителя " называется внешним ключом.

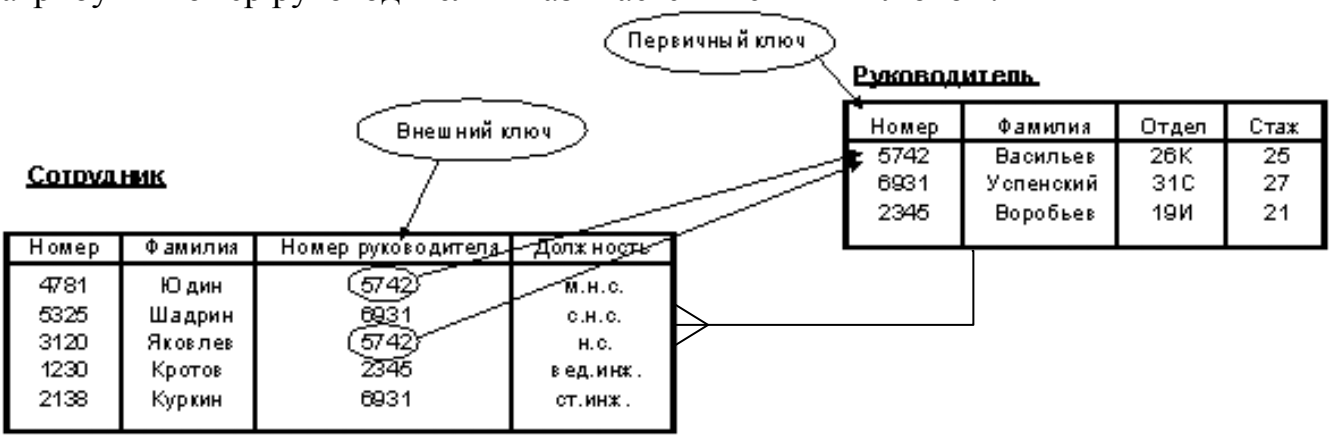


Рис.1. Связь между таблицами

Достоинства реляционной модели:

1) простота модели и модификации данных,
2) независимость данных и простота реализации,
3) запросы формируются без учета системной реализации,
4) хорошая теоретическая база в виде реляционной алгебры.

В качестве недостатка реляционная модели можно указать на то, что она обладает избыточностью (одно и тоже значение несколько раз присутствует в различных записях).

2 Краткие сведения о языке структурированных запросов SQL

Приложение-клиент при работе по технологии клиент-сервер формирует запрос к серверу, на котором расположена БД, на языке SQL, являющемся стандар-

том для реляционных БД. Удаленный сервер принимает запрос и переадресует его SQL-серверу БД. SQL-сервер интерпретирует запрос, выполняет его в базе данных, формирует результаты выполнения запроса и выдает его приложению-клиенту.

SQL (Structured Query Language) переводится как *Структурированный Язык Запросов.* Он предназначен для работы с реляционными БД. SQL сам определяет, какую наиболее эффективную последовательность операций следует использовать для их получения – эти детали не требуется указывать в запросе к базе данных при использовании SQL.

По технологии "Клиент-Сервер", запросы пользовательских персональных компьютеров (Клиентов) обрабатываются на специальных серверах баз данных (Серверах), а на пользовательские компьютеры возвращаются лишь результаты обработки запроса. При этом, естественно, нужен единый язык общения с Сервером и в качестве такого языка используется SQL. Поэтому все современные версии реляционных СУБД (DB2, Oracle, Ingres, Informix, Sybase) используют технологию "Клиент-Сервер" и язык SQL.

В настоящее время существует две разновидности языка SQL: интерактивный и программный (встроенный).

Интерактивный SQL используется для функционирования непосредственно в базе данных. В этом случае введенная пользователем команда выполняется немедленно.

Программный (встроенный) SQL предназначен для встраивания запросов в прикладную программу. Существует несколько видов программного SQL: статический, динамический и API-интерфесы.

Статический SQL – разновидность программного SQL, предназначенная для встраивания SQL-операторов в текст программы на языке программирования высокого уровня.

Динамический SQL – разновидность программного SQL, предназначенная для встраивания SQL-операторов в текст программы на языке программирования высокого уровня, допускающая динамическое формирование и выполнение запросов во время работы программы.

API-интерфейсы – разновидность программного SQL, основанная на использовании библиотек функций, разработанных для обеспечения связи прикладной программы с СУБД посредством выполнения SQL-запросов.

Существует несколько основных типов операторов SQL.

1. DDL (Data Definition Language) – язык определения данных. Он дает возможность создавать различные объекты БД и переопределять их структуру, например создавать или удалять таблицы. Примеры операторов:
   CREATE ALTER DROP RENAME
2. DML (Data Manipulation Language) – язык манипуляции данными. Он дает возможность манипулировать данными внутри объектов реляционной БД. Примеры операторов:
   INSERT UPDATE DELETE
3. DQL (Data Query Language) – язык запросов к данным. Он используется для построения запросов к реляционным БД. Примеры операторов:
   SELECT

4. DCL (Data Control Language) – язык управления данными. Он позволяет осуществлять контроль над возможностью доступа пользователей к данным внутри БД, а также для назначений пользователям привилегий доступа. Примеры операторов:
   ALTER PASSWORD GRANT REVOKE
5. DAL (Data Administration Language) – язык администрирования данных. Он дает возможность выполнять аудит и анализ операций внутри БД. Примеры операторов:
   START AUDIT STOP AUDIT
6. Команды управления транзакциями. Примеры операторов:
   COMMIT ROLLBACK SAVE POINT SET TRANSACTION

Несмотря на то, что стандарт языка SQL определяется **ANSI** (*Американским Национальным Институтом Стандартов*) и **ISO** (*Международной Организацией по Стандартизации*), разработчики коммерческих СУБД без уведомления расширяют SQL ANSI, интегрирую дополнительные возможности в этот язык. Поэтому в настоящее время существует большое количество диалектов SQL.

3 Типы данных

Понятие тип данных в БД полностью адекватно понятию типа данных в языках программирования.

Тип данных определяет:

- каков размер области памяти, занимаемой объектом;
- как интерпретируется эта память;
- какие действия можно выполнять с этим объектом.

В SQL используются следующие основные типы данных, форматы которых могут несколько различаться для разных СУБД:

а) Символьные типы данных - содержат буквы, цифры и специальные символы.

- **CHAR** или **CHAR(n)** -символьные строки фиксированной длины. Длина строки определяется параметром **n. CHAR** без параметра соответствует **CHAR(1)**. Для хранения таких данных всегда отводится **n** байт вне зависимости от реальной длины строки.
- **VARCHAR(n)** - символьная строка переменной длины. Для хранения данных этого типа отводится число байт, соответствующее реальной длине строки.

б) Целые типы данных - поддерживают только целые числа (дробные части и десятичные точки не допускаются). Над этими типами разрешается выполнять арифметические операции и применять к ним агрегирующие функции (определение максимального, минимального, среднего и суммарного значения столбца реляционной таблицы).

- **INTEGER** или **INT**- целое, для хранения которого отводится, как правило, 4 байта. *(Замечание: число байт, отводимое для хранения того или иного числового типа данных зависит от используемой СУБД и аппаратной платформы.)* Интервал значений от - 2147483647 до + 2147483648.
- **SMALLINT** - короткое целое (2 байта), интервал значений от - 32767 до +32768

в) Вещественные типы данных - описывают числа с дробной частью.

- **FLOAT** и **SMALLFLOAT** - числа с плавающей точкой (для хранения отводится обычно 8 и 4 байта соответсвенно).
- **DECIMAL(p)** - тип данных аналогичный **FLOAT** с числом значащих цифр **p**.
- **DECIMAL(p,n)** - аналогично предыдущему, **p** - общее количество десятичных цифр, **n** - количество цифр после десятичной запятой.

г) Денежные типы данных - описывают, естественно, денежные величины. Если ваша система такого типа данных не поддерживает, то используйте DECIMAL(p,n).

- **MONEY(p,n)** - все аналогично типу **DECIMAL(p,n)**. Вводится только потому, что некоторые СУБД предусматривают для него специальные методы форматирования.

д) Дата и время - используются для хранения даты, времени и их комбинаций. Большинство СУБД умеет определять интервал между двумя датами, а также уменьшать или увеличивать дату на определенное количество времени.

- **DATE** - тип данных для хранения даты.
- **TIME** - тип данных для хранения времени.
- **DATETIME** - тип данных для хранения моментов времени (год + месяц + день + часы + минуты + секунды + доли секунд).

е) Двоичные типы данных - позволяют хранить данные любого объема в двоичном коде (оцифрованные изображения, исполняемые файлы и т.д.). Определения этих типов наиболее сильно различаются от системы к системе, часто используются ключевые слова:

- **BINARY,**
- **BYTE,**
- **BLOB.**

4 Технология работы с языком SQL в MS Access

Для имитации работы в среде клиент/сервер можно использовать настольную СУБД MS Access. В качестве примеров, приведем последовательность действий пользователя в MS Access трех версий.

4.1 Microsoft Access 2003

Запустите СУБД MS Access. Выполните команду Пуск-Программы-MS Access. Запустится данная программа. Далее выполните команду Файл-Создать.

Откроется окно задания названия новой базы данных. Наберите – grNNNN_Фамилия.mdb

Откроется окно новой базы данных. Перейдите на вкладку Запросы. Для создания запроса (все команды SQL в данном случае будут запросами) выполните следующие действия:

1. Нажмите кнопку Создать на вкладке Запросы окно базы данных и выберите пункт Конструктор. Затем нажмите кнопку OK.
2. В окне конструктора закройте диалоговое окно Добавление таблицы без добавления таблиц в окно конструктора.

3. В меню запрос выберите команду Запрос SQL нажмите пункт Объединение.
4. Откроется окно ввода команды SQL. Введите инструкции SQL .

Например, SELECT CNAME, CITY
FROM CUSTOMER;

5. При закрытии этого окна с введенной инструкцией SQL появится вопрос Введите имя запроса. Наберите подходящее имя, например, Запрос1 и нажмите OK.
6. Для выполнения созданного запроса необходимо щелкнуть на нем на вкладке Запросы окна базы данных MS Access.

4.2 Microsoft Access 2007, 2010

Запустите СУБД MS Access. Выполните команду Пуск-Программы-MS Access. Запустится данная программа. Далее выполните команду Системное меню -Создать.

Откроется окно задания названия новой базы данных. В поле "Имя файла" наберите – grNNNN_Фамилия и нажмите "Создать".

Откроется окно новой базы данных. Перейдите на панель Создание. Для создания запроса (все команды SQL в данном случае будут запросами) выполните следующие действия.

1) Нажмите кнопку "Конструктор запросов".
2) В окне конструктора закройте диалоговое окно "Добавление таблицы" без добавления таблиц в окно конструктора.
3) На панели нажмите кнопку Режим SQL.
4) Откроется окно ввода команды SQL. Введите инструкции SQL

Например, SELECT CNAME, CITY
FROM CUSTOMER;

5) При закрытии этого окна с введенной инструкцией SQL появится вопрос "Введите имя запроса". Наберите имя, например, Запрос1 и нажмите "OK".
6) Для выполнения созданного запроса необходимо щелкнуть на нем на вкладке Запросы окна базы данных MS Access, либо в режиме SQL нажать кнопку "Выполнить".

5 Конструкции языка SQL

5.1 Создание таблиц

Таблицы создаются командой **CREATE TABLE**. Эта команда создает пустую таблицу. Команда CREATE TABLE определяет имя таблицы и саму таблицу в виде описания набора имен столбцов, указанных в определенном порядке. Она также определяет типы данных и размеры столбцов. Каждая таблица должна иметь, по крайней мере, один столбец.

Синтаксис команды CREATE TABLE:

```
CREATE TABLE <table-name>
(<column name> <data type>[(<size>)],
<column name> <data type>[(<size>)], ... );
```

Значение аргумента size зависит от типа данных. Если вы его не указываете, СУБД будет устанавливать значение автоматически.

Пример:

В дальнейшем будем рассматривать базу данных, состоящую из 3х таблиц, приведенных ниже.

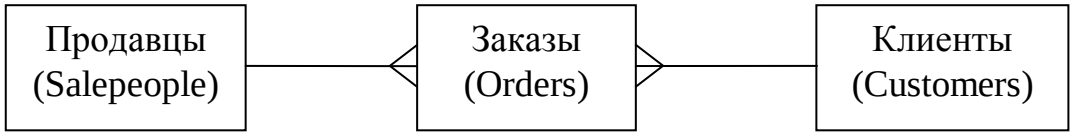


Таблица Продавцы (Salepeople):

SNum	SName	City	Comm
11	Brown	London	0.12
12	Serres	San Jose	0.13
14	Smith	London	0.11
17	Sanches	Barcelona	0.15
13	Collins	New York	0.10

SNum - уникальный номер, назначенный каждому продавцу,

SName - имя продавца,

City - расположение продавца (город),

Comm - комиссионные продавцов в десятичном виде.

Таблица Клиенты (Customers):

CNum	CName	City	Rating	Snum
21	Hoffman	Berlin	100	11
22	Giovanni	Katmandu	200	13
23	Liu	Caracas	200	12
24	Grass	N.Novgorod	300	12
26	Clemens	New York	100	11
28	Cisneros	Caracas	300	17
27	Pereira	Katmandu	100	14

CNum - уникальный номер, назначенный каждому клиенту,

CName - имя клиента,

City - расположение клиента (город),

Rating - код указывающего уровень предпочтения данного клиента перед другими (рейтинг),

SNum - номер продавца назначенного этому клиенту.

Таблица Заказы (Orders):

Onum	AMT	ODate	CNum	SNum
1	18.69	2008.10.03	28	17
2	767.19	2008.10.03	21	11
3	1900.10	2008.10.03	27	14
4	5160.45	2008.10.03	23	12
5	1098.16	2008.10.03	28	17
6	1713.23	2008.10.04	22	13
7	100.0	2008.10.05	24	12
8	200.0	2008.10.06	26	11

ONum - уникальный номер данный каждому приобретению,

AMT - значение суммы приобретений,
ODate - дата приобретения,
CNum - номер клиента делающего приобретение,
SNum - номер продавца, продающего приобретение.

Для примера рассмотрим, как создать таблицу продавцов:

```
CREATE TABLE Salepeople
( SNum      integer,
  SName     char (10),
  City      char (10),
  Comm      number );
```

5.2 Удаление таблиц

Удалить можно только пустую таблицу. Заполненная таблица с находящимися в ней строками не может быть удалена, т. е. таблица перед удалением должна быть очищена. Команда на удаление таблицы имеет следующий вид:

```
DROP TABLE < table name >;
```

Например: **DROP TABLE Salepeople;**

5.3 Изменение таблицы, после того как она была создана

Для изменения таблиц используется команда **ALTER TABLE.** Она может добавлять столбцы к таблице, удалять столбцы, изменять их размеры, а также добавлять или удалять ограничения. Эта команда не часть стандарта ANSI, поэтому в разных системах она имеет разные возможности.

Типичный синтаксис чтобы добавить столбец к таблице:

```
ALTER TABLE <table name> ADD <column name>
<data type> <size>;
```

Например:

ALTER TABLE Salepeople ADD Phone CHAR(7);

5.4 Введение ограничений

Когда вы создаете или изменяете таблицу, вы можете устанавливать ограничения на значения, вводимые в поля. В этом случае SQL будет отклонять любые данные, которые нарушают заданные ограничения. Ограничение может устанавливаться как отдельно для каждого конкретного столбца, так и на всю таблицу в целом. Для определения ограничения на столбец, Вы вставляете специальное выражение в конец имени столбца после типа данных и перед запятой. Далее показан синтаксис для команды CREATE TABLE с учетом ограничений:

```
CREATE TABLE < table name >
( <column name> <data type> <column constraint>,
<column name> <data type> <column constraint> ...
;
```

С помощью ограничений (CONSTRAINT) можно устанавливать индексы (структуры БД, представляющие собой указатель на конкретную запись в таблице) для полей таблицы, первичный и внешний ключи, определять отношение и целостность данных.

Чтобы предохранить поле таблицы от ввода в него *пустых* (NULL) значений, нужно при создании таблицы указать для соответствующего столбца ключевое выражение **NOT NULL**.

Например, очевидно, что первичные ключи никогда не должны быть *пустыми*, поэтому наш пример по созданию таблицы Salepeople может быть модифицирован следующим образом:

```
CREATE TABLE Salepeople
( SNum    integer NOT NULL,
SName   char (10),
City      char (10),
Comm    number);
```

Достаточно часто требуется быть уверенным, что все значения, введенные в столбец, отличаются друг от друга. Если вы помещаете ограничение столбца UNIQUE в поле при создании таблицы, база данных отклонит любую попытку ввода в это поле для одной из строк, значения, которое уже представлено в другой строке. Например:

```
CREATE TABLE Salepeople
( SNum    integer UNIQUE,
SName   char (10),
City      char (10),
Comm    float);
```

5.5 Создание первичных ключей

Первичный ключ создается с помощью выражения **PRIMARY KEY**. Таблица может содержать только один первичный ключ. Существует два способа создания первичного ключа:

1) в виде ограничения на столбец (для простого первичного ключа);

2) в виде ограничения на таблицу (для простого или составного первичного ключа).

1-й способ:

```
CREATE TABLE Salepeople
( SNum   integer PRIMARY KEY,
SName   char (10),
City      char (10),
Comm   float);
```

2-й способ:

```
CREATE TABLE Orders
(Onum      integer,
AMT        double,
ODate      date,
CNum       integer,
SNum       integer,
PRIMARY KEY (CNum, SNum));
```

5.6 Создание внешних ключей

Внешний ключ создается с помощью выражения **FOREIGN KEY**. Назначение FOREIGN KEY – это ограничение допустимых значений поля множеством значений первичного ключа, ссылка на который указывается при описании данного ограничения. Как и для первичного, для внешнего ключа также существует два способа его назначения.

1-й способ: в виде ограничения на столбец.

```
CREATE TABLE Orders
(Onum      integer,
 AMT       double,
 ODate     date,
 CNum      integer REFERENCES Customers (CNum),
 SNum      integer REFERENCES Salepeople (SNum));
```

2-й способ: в виде ограничения на таблицу.

```
CREATE TABLE Orders
(Onum      integer,
 AMT       double,
 ODate     date,
 CNum      integer,
 SNum      integer,
FOREIGN KEY (CNum) REFERENCES Customers (CNum),
FOREIGN KEY (SNum) REFERENCES Salepeople (SNum));
```

Два примера, приведенные выше, определяют поля CNum и SNum как внешние ключи таблицы Orders, ссылающиеся на первичные ключи CNum и SNum таблиц Customers и Salepeople соответственно.

Если в родительской таблице у первичного ключа уже указано ограничение PRIMARY KEY, то при задании ограничения FOREIGN KEY, накладываемого на таблицу или на столбцы, можно в скобках не указывать список столбцов первичного ключа.

5.7 Ввод значений в таблицы

Все строки в SQL вводятся с использованием команды модификации **INSERT**. Предложение INSERT имеет один из следующих форматов:

```
INSERT  INTO   <table name | view name>
[(column [,column] ...)]
VALUES ( <value> [,<value>] ... );
```

или

```
INSERT  INTO   <table name | view name>
[(column [,column] ...)]
подзапрос;
```

Так, например, чтобы ввести строку в таблицу Продавцов, вы можете использовать следующее условие:

INSERT INTO Salepeople
VALUES (11, 'Brown', 'London', 0.12);

Вы можете также указывать столбцы, куда вы хотите вставить значение имени. Это позволяет вам вставлять имена в любом порядке. Например:

INSERT INTO Salepeople (SName, Comm, SNum)
VALUES ('Brown',0.12, 11);

Вы можете также использовать команду INSERT чтобы получать или выбирать значения из одной таблицы и помещать их в другую, чтобы использовать их вместе с запросом. Чтобы сделать это, вы просто заменяете предложение VALUES (из предыдущего примера) на соответствующий запрос:

INSERT INTO Londonstaff
SELECT * FROM Salespeople
WHERE City = 'London';

5.8 Удаление строк

Вы можете удалять строки из таблицы командой модификации - **DELETE**. Она может удалять только введённые строки, а не индивидуальные значения полей. Предложение DELETE имеет следующий формат:

```
DELETE FROM  <table name | view name>
[WHERE search-condition];
```

Например, чтобы удалить все содержание таблицы Продавцы, вы можете ввести следующее условие:

DELETE FROM Salepeople;

Чтобы определить какие строки будут удалены, вы должны использовать условие. Например, чтобы удалить продавца Collins из таблицы, вы можете ввести:

DELETE FROM Salepeople
WHERE SNum = 13;

5.9 Изменение значения поля

Это выполняется командой **UPDATE**. Эта команда содержит предложение UPDATE в которой указано имя используемой таблицы и предложение SET которое указывает на изменение которое нужно сделать для определенного столбца. Предложение UPDATE имеет два формата. Первый из них:

```
UPDATE  <table name | view name>
SET column = expression [, column = expression] ...
[WHERE search-condition]
```

где expression - это столбец | выражение | константа | переменная.

Второй вариант:

```
UPDATE  <table name>
SET  column = expression, ...
[ FROM  table-list ]
[ WHERE search-condition ]
```

Например, чтобы изменить оценки всех клиентов на 200, вы можете ввести:

UPDATE Customers
SET Rating = 200;

Для модифицирования определенных строк надо использовать условие, как в DELETE. Вот как например можно выполнить изменение одинаковое для всех клиентов продавца Brown (имеющего SNum=11):

UPDATE Customers
SET Rating = 200
WHERE SNum = 11;

Предложение SET может назначать любое число столбцов, отделяемых запятыми. Все указанные назначения могут быть сделаны для любой табличной строки, но только для одной в каждый момент времени. Например:

UPDATE Salepeople
SET SName = 'Gibson', City = 'Boston', Comm = 0.10
WHERE SNum = 14;

Вы можете использовать арифметические выражения в предложении SET команды UPDATE. Например:

UPDATE Salepeople
SET Comm = Comm * 2;

5.10 Извлечение информации из таблицы (простейшие запросы)

Для извлечения информации из таблиц применяется команда **SELECT**. В самой простой форме, в команде указываются столбцы, которые вы бы хотели видеть, и имя таблицы из которой они будут извлекаться. Например, вы могли бы вывести таблицу Продавцов без столбца SNum, введя следующую команду:

SELECT SName, City, Comm
FROM Salepeople;

Чтобы вывести все столбцы таблицы Salepeople:

SELECT *
FROM Salepeople;

Если вы не хотите чтобы какие-либо значения дублировались в списке вывода, надо воспользоваться аргументом **DISTINCT** (отличие) - который обеспечивает вас способом устранять двойные значения из вашего предложения SELECT. DISTINCT может указываться только один раз в данном предложении SELECT. Пример:

SELECT DISTINCT SNum
FROM Orders;

SQL дает возможность вам устанавливать критерии, определяющие выбор строк для вывода. Предложение **WHERE** команды SELECT позволяет устанавливать условия, значения которых может быть верным или неверным для любой строки таблицы. Команда извлекает только те строки из таблицы, для которой такое утверждение верно. Например, вы хотите видеть имена и комиссионные всех продавцов в Лондоне. Вы можете ввести такую команду:

SELECT SName, Comm
FROM Salepeople
WHERE City = ‘London’;

Работа с несколькими таблицами

5.11 Объединение таблиц

Объединение является одним из видов операций в реляционных базах данных. Оно определяет связи между несколькими таблицами и выводит информацию из них в терминах этих связей. В виду того, что в разных таблицах могут быть столбцы с одинаковыми именами, для идентификации нужного столбца используется префикс имени таблицы.

При объединении таблицы представленные списком в предложении FROM запроса, отделяются запятыми. Условие запроса может ссылаться на любой столбец любой связанной таблицы и, следовательно, может использоваться для связи между ними. Обычно сравниваются значения в столбцах различных таблиц, чтобы определить, удовлетворяет ли WHERE установленному условию. Самый простой способ объединения -это декартово произведение, его можно выполнить следующим образом:

SELECT Customers.*, Salepeople.*
FROM Salepeople, Customers;

Но в данном случае получится бесполезная таблица, с множеством ненужной информации. Если из декартова произведения убрать ненужные строки и столбцы, то можно получить нужные данные. Это реализуется с помощью WHERE фразы.

Например, если вы хотите увидеть все комбинации продавцов и клиентов для данного города, то вы должны ввести следующее:

SELECT Customers.CName, Salepeople.SName, Salepeople.City
FROM Salepeople, Customers
WHERE Salepeople.City = Customers.City;

Что SQL в основном делает в объединении - так это исследует каждую комбинацию строк двух или более возможных таблиц и проверяет эти комбинации по их условиям. Чаще всего используется объединение через целостность.

Пример: показ имен всех клиентов соответствующих продавцам, которые их обслуживают

SELECT Customers.CName, Salepeople.SName
FROM Customers, Salepeople
WHERE Salepeople.SNum = Customers.SNum;

Объединения, которые используют условия, основанные на равенствах, называются *объединениями по равенству*, это наиболее общий вид объединения, но имеются и другие.

5.12 Объединение таблицы с собой

Для объединения таблицы с собой, вы можете сделать каждую строку таблицы, одновременно, и комбинацией ее с собой и комбинацией с каждой другой строкой таблицы. Вы затем оцениваете каждую комбинацию в терминах условия. Это объединение - такое же, как и любое другое объединение между двумя таблицами, за исключением того, что в данном случае обе таблицы идентичны.

Когда вы объединяете таблицу с собой, все повторяемые имена столбца, заполняются префиксами имени таблицы. Чтобы ссылаться к этим столбцам внутри запроса, вы должны иметь два различных имени для этой таблицы. Вы можете сделать это с помощью определения временных имен, называемых - ***псевдонимами.*** Они определяются через пробел после имени таблицы в предложении FROM запроса.

Пример: нахождение всех пар клиентов, имеющих один и тот же рейтинг.

SELECT a.CName, b.CName, a.Rating
FROM Customers a, Customers b
WHERE a.Rating = b.Rating;

В данном случае, SQL ведет себя так, как если бы он соединял две таблицы называемые a и b. Псевдоним существует только на время выполнения команды. В выше приведенном примере есть избыточные строки, два значения для каждой комбинации. Значение **A** в псевдониме сначала выбирается в комбинации со значением **B** во втором псевдониме, а затем значение **A** во втором псевдониме выбирается в комбинации со значением **B** в первом псевдониме, При этом каждая строка сравнивается сама с собой. Простой способ избежать этого заключается в том, чтобы налагать ограничения на два значения, так чтобы один мог быть меньше другого или предшествовал ему в алфавитном порядке. В этом случае те же самые значения в обратном порядке выбраны снова не будут.

Пример:

SELECT a.CName, b.CName, a.Rating
FROM Customers a, Customers b
WHERE a.Rating = b.Rating
AND a.CName < b.CName;

В данном примере, если первая комбинация удовлетворяет второму условию, то она выводится, но та же комбинация в обратном порядке уже не будет ему удовлетворять и наоборот. Вы можете использовать псевдонимы и тогда, когда вы хотите создать альтернативные имена для ваших таблиц в команде. Вы можете использовать любое число псевдонимов для одной таблицы в запросе. Вам не всегда

обязательно использовать каждый псевдоним или таблицу, которые упомянуты в предложении FROM запроса или в предложении SELECT. Вы можете также создать объединение, которое включает и различные таблицы и псевдонимы одиночной таблицы.

5.13 Простые вложенные подзапросы

С помощью SQL вы можете вкладывать запросы внутрь друг друга. Обычно, внутренний запрос генерирует значение, которое проверяется во внешнем запросе, определяющем верно оно или нет.

Пример: предположим, что мы знаем имя продавца: Smith, но не знаем значение его поля SNum и хотим извлечь все заказы из таблицы Заказов. Сделать это можно следующим образом:

```
SELECT *
FROM Orders
WHERE SNum =
( SELECT SNum FROM Salepeople
WHERE SName = 'Smith' );
```

Сначала выполняется внутренний запрос, а затем его результаты используются для формирования внешнего запроса (SNum сравнивается с результатом подзапроса). В данном случае подзапрос должен выбрать один и только один столбец, а тип данных этого столбца должен совпадать с типом того значения, с которым он будет сравниваться в предикате. Вы можете в некоторых случаях использовать DISTINCT, чтобы подзапрос генерировал одиночное значение.

Пример: предположим, что мы хотим найти все заказы для тех продавцов, которые обслуживают Hoffman (CNum=21).

```
SELECT * FROM Orders
WHERE SNum =
( SELECT DISTINCT SNum
FROM Orders
WHERE CNum = 21 );
```

В данном случае подзапрос выведет только одно значение 11, но в принципе может быть много подобных значений, и тогда DISTINCT выберет только одно.

Один тип функций, который автоматически может производить одиночное значение для любого числа строк - агрегатная функция, ее и можно использовать в подзапросе.

Например, вы хотите увидеть все заказы, имеющие сумму приобретений выше средней на 4-е октября:

```
SELECT * FROM Orders
WHERE AMT >
( SELECT AVG (AMT) FROM Orders
WHERE ODate = ‘2008.10.04’ );
```

Вы можете использовать подзапросы, которые производят любое число строк, если вы используете специальный оператор IN. Когда вы используете IN с подзапросом, SQL просто формирует набор значений для IN из вывода подзапроса.

Пример: показать все заказы для продавцов находящихся в Лондоне.

```
SELECT * FROM Orders
WHERE SNum IN
( SELECT SNum
FROM Salepeople
WHERE City = 'London' );
```

Действия основных команд языка SQL

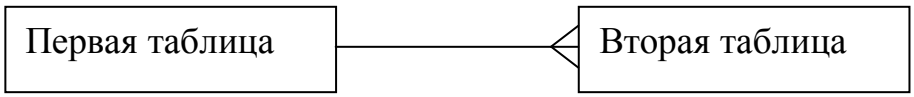
Команда SQL	Выполняемое действие
CREATE TABLE	Создание таблицы
DROP TABLE	Удаление таблицы
ALTER TABLE	Изменение таблицы
NOT NULL	Ограничение на использование пустых значений
UNIQUE	Ограничение на использование одинаковых значений
PRIMARY KEY	Создание первичного ключа
FOREIGN KEY	Создание внешнего ключа
INSERT	Добавление строк в таблицу
DELETE	Удаление строк из таблицы
UPDATE	Изменение значений в таблице
SELECT	Извлечение информации из таблиц

6 Задания к расчетной работе

6.1 Общее задание

1 Напишите запросы на создание таблиц, приведенных в задании, учитывая что:

а) первая таблица связана со второй связью "один ко многим";



б) значения первого поля в каждой таблице должно быть уникальными и не содержать значений NULL;

в) первые поля в таблицах являются первичными ключами,

г) последнее поле во второй таблице является внешним ключом.

2 Напишите запросы на добавление приведенных в задании данных в созданные таблицы.

6.2 Варианты заданий

Вариант 1. Строительная организация состоит из нескольких подразделений. В базе данных должны содержаться сведения о:

а) подразделениях строительной организации (подразделение представляется номером подразделения, названием, специализацией);

б) сотрудниках (данными о служащих являются его табельный номер, ФИО, год рождения, должность, подразделение в котором он работает).

Подразделение

№_подразделения	Название	Количество_ПК	Специализация
15	Плановый отдел	12	Составление планов работ
21	Сметно-договорной отдел	15	Выполнение и проверка сметных расчетов
23	Цех железобетонных изделий	1	Изготовление ЖБ изделий
48	СМУ-1	2	Производство СМР
52	СМУ-2	3	Производство СМР

Сотрудник

Таб_№	ФИО	Год_рождения	Должность	Подразделение
5383	Сидоров Иван Михайлович	1958	Экономист	15
6852	Иванов Петр Сергеевич	1960	Начальник цеха	23
6578	Морозова Анастасия Андреевна	1975	Экономист	15
4852	Бирюков Леонид Ильич	1980	Начальник отдела	15
6548	Волков Дмитрий Александрович	1955	Прораб	48
3216	Зайцев Кирилл Викторович	1965	Мастер	48
6536	Касатонова Юлия Олеговна	1983	Сметчик	21

3 Напишите запрос, который увеличивает Количество_ПК во всех подразделениях на 5 шт.
4 Напишите запрос, переводящий сотрудников СМУ-1 в СМУ-2.
5 Напишите запрос, который выводит №_Подразделения, Название и Специализацию из таблицы Подразделение.
6 Напишите запрос, который вывел бы список всех сотрудников Планового отдела.
7 Напишите запрос, который вывел бы таблицу Сотрудник со столбцами в обратном порядке.
8 Напишите запрос, извлекающий из таблицы Сотрудник список подразделений, в которых работают сотрудники. Подразделения не должны повторяться.
9 Напишите запрос, считающий средний возраст сотрудников.
10 Напишите запрос на создание списка, состоящего из ФИО сотрудника и названия его подразделения для всех подразделений, в которых количество компьютеров меньше 10.
11 Напишите запрос на удаление всех сотрудников, работающих в подразделении №23.

Вариант 2. Строительная организация ведет работы на нескольких объектах. В базе данных должны содержаться сведения о:

а) заказчиках (данными о заказчике являются номер заказчика, его наименование, адрес, количество сотрудников);
б) объектах (данными об объекте являются его номер, наименование, сметная стоимость работ, планируемая дата окончания работ, заказчик).

Заказчик

№_заказчика	Наименование	Адрес	Количество_сотрудников
15	ОАО Парус	ул.Тимирязева, 30	30
21	ЗАО Берег	пр. Ленина, 45	52
23	ООО Корвет	пр. Гагарина, 28	108
48	ОАО Консул	пер. Союзный, 4	24
52	ГОУ ВПО ННГАСУ	ул.Ильинская, 65	1850

Объект

№_объекта	Наименование	Сметная_стоимость	Дата_окончания	Заказчик
265	Поликлиника	150	01.04.2010	21
546	Школа	82	01.05.2009	21
845	Жилой дом по ул.Тимирязева	136	01.08.2011	52
828	Котельная №1	25	01.06.2009	15
145	Жилой дом по ул.Гоголя	140	01.03.2009	52
548	Жилой дом по пр.Гагарина	250	01.11.2012	23
753	Котельная №1	23	01.07.2010	48

3 Напишите запрос, который сокращает Количество_сотрудников у всех заказчиков на 5.
4 Напишите запрос, передающий объекты от заказчика ОАО Консул к ООО Корвет.
5 Напишите запрос, который выводит Наименование, Адрес и Количество_сотрудников из таблицы Заказчик.
6 Напишите запрос, который вывел бы список всех объектов заказчика ЗАО Берег.
7 Напишите запрос, который вывел бы таблицу Объект со столбцами в обратном порядке.
8 Напишите запрос, извлекающий из таблицы Объект список заказчиков этих объектов. Заказчики не должны повторяться.
9 Напишите запрос, выводящий наименование и сметную стоимость самого дорогого объекта.
10 Напишите запрос на создание списка, состоящего из Наименования объекта и Наименования его заказчика для всех заказчиков, у которых работает более 100 человек.
11 Напишите запрос на удаление всех объектов заказчика №21.

Вариант 3. У строительной организации несколько складов. В базе данных должны содержаться сведения о:

а) складах (данными о складе являются его номер, адрес, вид хранящихся строительных материалов, расстояние до областного центра);

б) строительных материалах (данными о материалах являются его номер, наименование, единица измерения, остаток, склад).

Склад

№_склада	Адрес	Вид_материалов	Расстояние
1	д.Крутово	сыпучие	5
2	пос.Веканово	отделочные	10
3	пос.Заскочиха	отделочные	15
4	д.Орлово	отделочные	8
5	д.Комарово	кирпич	12

Стройматериал

№_материала	Наименование	Ед_изм	Остаток	Склад
5466	Цемент	кг	680	1
7898	Краска	кг	350	4
1232	Шпатлевка	кг	260	2
4565	Кирпич глиняный	м3	68	5
7535	Песок	т	250	1
1595	Известь	т	9	3
8542	Кирпич силикатный	м3	120	5

3 Напишите запрос, который уменьшает остаток всех строительных материалов на 10%.

4 Напишите запрос, переводящий строительные материалы, находящиеся на складе в пос.Веканово на склад в пос.Заскочиха.

5 Напишите запрос, который выводит Адрес, Вид_материалов и Расстояние из таблицы Склад.

6 Напишите запрос, который вывел бы список всех стройматериалов, находящихся на складе в д.Комарово.

7 Напишите запрос, который вывел бы таблицу Стройматериал со столбцами в обратном порядке.

8 Напишите запрос, извлекающий из таблицы Стройматериал список складов, где хранятся эти стройматериалы. Склады не должны повторяться.

9 Напишите запрос, выводящий наименование и номер склада стройматериала с самым большим остатком.

10 Напишите запрос на создание списка, состоящего из Наименования стройматериала и Адреса склада, где он хранится для всех складов, расположенных на расстоянии не менее 12 км от областного центра.

11 Напишите запрос на удаление всех стройматериалов, хранящихся на складе №1.

Вариант 4. В управлении механизации несколько типов машин (бульдозеры, автокраны, и т.д.). В базе данных должны содержаться сведения о:

а) типах машин (данными о типе являются его номер, название, дальность перегона (км), назначение);

б) машинах (данными о машине являются инвентарный номер, название, местонахождение базы, количество, тип).

Тип_машины

№_Типа	Название	Даль-ность_перегона	Вид_работ
1	Землеройные	30	Земляные работы
2	Грузовые	1000	Перевозка грузов
3	Бетоносмесительные	200	Перевозка растворов
4	Специализированные	100	Строительные работы
5	Сваебойные	30	Забивка свай

Машина

№_Машины	Название	База	Количество	Тип
234	Бульдозер	Москва	2	1
345	Трактор	Калуга	3	4
654	Грейдер	Обнинск	1	1
642	Автобетоносмеситель	Новгород	3	3
854	Грузовик	Москва	4	2
321	Тягач	Новгород	1	2
643	Одноковшовый погрузчик	Калуга	2	1

3 Напишите запрос, который увеличивает Дальность_перегона у всех типов машин на 10 км.

4 Напишите запрос, меняющий тип специализированных машин на грузовые.

5 Напишите запрос, который выводит Название, Дальность_перегона и Вид_работ для всех типов машин.

6 Напишите запрос, который вывел бы список всех машин, имеющих тип "Землеройные".

7 Напишите запрос, который вывел бы таблицу Машина со столбцами в обратном порядке.

8 Напишите запрос, извлекающий из таблицы Машина список типов этих машин. Типы не должны повторяться.

9 Напишите запрос, выводящий название машины и базу, имеющую самое большое количество.

10 Напишите запрос на создание списка, состоящего из Названия машины и Названия ее типа для всех типов, дальность перегона которых не менее 200 км.

11 Напишите запрос на удаление всех машин с типом 1.

Вариант 5. В строительной организации несколько бригад. В базе данных должны содержаться сведения о:

а) бригадах (данными о бригаде являются код бригады, фамилия бригадира, число работников, вид выполняемых работ);

б) работниках (данными о работнике являются его табельный номер, ФИО, год рождения, разряд, код бригады).

Бригада

Код_бригады	Бригадир	Число_работников	Вид_работ
1	Сидоров	10	Малярные
2	Петров	9	Штукатурные
3	Иванов	5	Сантехнические
4	Кузнецов	8	Каменные
5	Смирнов	10	Электротехнические

Работник

Таб_№	ФИО	Год_рождения	Разряд	Бригада
3258	Морозова Анастасия Андреевна	1960	3	1
4342	Павлинов Алексей Владимирович	1975	4	2
5428	Пигалов Максим Александрович	1985	3	1
5321	Рязанцев Максим Вячеславович	1961	5	5
4418	Смирнов Сергей Александрович	1963	5	3
3021	Чистова Ксения Марковна	1950	4	5
5152	Чуважова Анастасия Сергеевна	1981	3	2

3 Напишите запрос, который увеличивает Число_работников во всех бригадах на 2 человека.

4 Напишите запрос, переводящий работников из бригады №2 в бригаду №1.

5 Напишите запрос, который выводит Бригадира, Число_работников, Вид_работ из таблицы Бригада.

6 Напишите запрос, который вывел бы список всех работников бригады, выполняющей каменные работы.

7 Напишите запрос, который вывел бы таблицу Работник со столбцами в обратном порядке.

8 Напишите запрос, извлекающий из таблицы Работник список кодов бригад. Коды не должны повторяться.

9 Напишите запрос, считающий средний возраст работников.

10 Напишите запрос на создание списка, состоящего из ФИО работника и Вида_работ бригады для всех бригад, в которых число работников больше 8.

11 Напишите запрос на удаление всех работников бригады №1.

Вариант 6. При изготовлении строительных изделий требуется несколько видов ресурсов (цемент, гравий, металлопрокат и т.п.). В базе данных должны содержаться сведения о:

а) ресурсах (данными о ресурсе являются код ресурса, наименование, единица измерения, необходимый запас);

б) потребностях ресурсов при изготовлении изделий (данными о потребности являются код, наименование, дата поставки необходимого ресурса, код данного ресурса).

Ресурс

Код	Наименование	Ед_изм	Запас
2346	Цемент	т	120
2364	Гравий	т	245
5678	Песок	т	330
6753	Металлический профиль	м	560
3783	Арматура	т	300

Потребность

Код	Изделие	Дата	Расход	Ресурс
9087	Лестничный марш	02.10.2009	1,5	2346
7453	Лестничный марш	03.10.2009	2	5678
8754	Плита перекрытия	15.10.2009	1,9	2346
6543	Стеновая панель	01.11.2009	1,8	5678
8765	Металлическая ферма	10.11.2009	9	6753
7645	Стеновая панель	01.11.2009	0,8	2346
9876	Плита перекрытия	10.10.2009	2,4	3783

3 Напишите запрос, который сокращает Запас всех ресурсов на 5%.

4 Напишите запрос, меняющий дату поставки ресурса в таблице Потребность на 05.11.2009 для тех записей, где используется Песок.

5 Напишите запрос, который выводит Наименование, Ед_изм и Запас из таблицы Ресурс.

6 Напишите запрос, который вывел бы список всех записи из таблицы Потребность, где используется цемент.

7 Напишите запрос, который вывел бы таблицу Потребность со столбцами в обратном порядке.

8 Напишите запрос, извлекающий из таблицы Потребность список ресурсов. Ресурсы не должны повторяться.

9 Напишите запрос, выводящий наименование изделия и дату поставки соответствующего ресурса, расход которого максимален.

10 Напишите запрос на создание списка, состоящего из Наименования изделия и Наименования соответствующего ресурса для ресурсов, которые измеряются в тоннах.

11 Напишите запрос на удаление записей из таблицы Потребность, где используется ресурс №2346.

Вариант 7. На заводе железобетонных конструкций существует несколько технологических линий изготовления ж.б. конструкций и изделий. В базе данных должны содержаться сведения о:

а) технологических линиях (данными о линии являются номер, название, число рабочих мест, дата очередного профилактического обслуживания);

б) железобетонных изделиях (данными об изделии являются код, наименование, цена изделия (тыс. руб.), план выпуска в смену (шт.), технологическая линия).

Линия

№_линии	Название	Число_мест	Дата
1	Плиты	4	01.03.2010
2	Стеновые панели	5	15.03.2010
3	Лестничные марши	3	01.12.2009
4	Фермы	7	01.11.2009
5	Спец. изделия	5	30.11.2009

Изделие

Код	Наименование	Цена	План	Линия
9087	Плита перекрытия 6м	80	20	1
7453	Плита перекрытия 12м	120	25	1
8754	Ж.б. ферма 12м	240	10	4
6543	Стеновая панель	75	25	2
8765	Лестничный марш	60	18	3
7645	Ж.б. ферма 6м	150	13	4
9876	Фундаментный блок	50	30	5

3 Напишите запрос, который сокращает Число_мест на всех технологических линиях на 1.

4 Напишите запрос, увеличивающий стоимость изделий, которые изготавливаются на технологической линии "Плиты" на 1 тысячу руб.

5 Напишите запрос, который выводит Название, Число_мест и Дата из таблицы Линия.

6 Напишите запрос, который вывел бы список всех изделий, изготавливаемых на технологической линии "Фермы".

7 Напишите запрос, который вывел бы таблицу Изделие со столбцами в обратном порядке.

8 Напишите запрос, извлекающий из таблицы Изделие список номеров технологических линий. Номера не должны повторяться.
9 Напишите запрос, выводящий наименование и цену самого дорогого изделия.
10 Напишите запрос на создание списка, состоящего из Наименования изделия и Названия технологической линии для всех линий, число рабочих мест которых не менее 4.
11 Напишите запрос на удаление всех изделий, изготавливаемых на технологической линии №1.

Вариант 8. Процесс возведения здания можно расчленить на множество бригадных процессов и определить набор ресурсов используемых в каждом процессе. База данных должна содержать сведения о:
а) бригадных процессах (данными о процессе являются код, наименование, трудоемкость (чел/час), дата начала работ);
б) выполнение работ (данными о выполнении работ являются его код, наименование бригады и количество человек в ней, минимальный разряд членов бригады, код выполняемого процесса).

Процесс

Код	Наименование	Трудоемкость	Дата
1	Укладка плит перекрытия	16	01.03.2009
2	Возведение стен	18	05.03.2009
3	Установка лестничных маршей	3	05.03.2009
4	Установка оконных и дверных блоков	12	01.04.2009
5	Прокладка инженерных систем	25	15.04.2009

Выполнение

Код	Бригада	Количество	Разряд	Процесс
234	Монтажники	5	3	1
345	Крановщик	1	5	1
654	Сантехники	2	3	5
642	Отделочники	4	4	4
854	Монтажники	4	3	2
321	Крановщик	1	5	2
643	Монтажники	3	4	3

3 Напишите запрос, который сокращает Трудоемкость всех процессов на 1 чел/час.
4 Напишите запрос, повышающий минимальный разряд членов бригады, участвующей в процессе "Возведение стен".
5 Напишите запрос, который выводит Наименование, Трудоемкость и Дату из таблицы Процесс.
6 Напишите запрос, который вывел бы все записи из таблицы Выполнение, которые связаны с процессом " Укладка плит перекрытия ".

7 Напишите запрос, который вывел бы таблицу Выполнение со столбцами в обратном порядке.
8 Напишите запрос, извлекающий из таблицы Выполнение список кодов процессов. Процессы не должны повторяться.
9 Напишите запрос, выводящий название бригады и минимальный разряд ее рабочих для бригады с самым большим количеством человек.
10 Напишите запрос на создание списка, состоящего из Наименования бригады и и Наименования процесса для всех процессов, трудоемкость которых не менее 18 чел/часов.
11 Напишите запрос на удаление всех записей в таблице Выполнение, связанных с процессом №1.

Вариант 9. Строительное подразделение ведет работу на нескольких объектах. В базе данных должны содержаться сведения:
а) об объектах (данными об объекте являются его номер, наименование, сметная стоимость работ (млн. руб.), процент выполнения работ);
б) о поставках ресурсов (данными о поставке ресурсов являются код поставки, наименование ресурса, единица измерения, количество, объект).

Объект

№_объекта	Наименование	Стоимость	Выполнение
1	Поликлиника	150	90
2	Школа	82	80
3	Жилой дом по ул.Тимирязева	136	50
4	Котельная №1	25	20
5	Жилой дом по ул.Гоголя	140	30

Поставка_ресурсов

Код	Ресурс	Ед_изм	Количество	Объект
3258	Цемент	кг	680	1
4342	Краска	кг	350	4
5428	Шпатлевка	кг	260	2
5321	Кирпич глиняный	м3	68	5
4418	Песок	т	250	1
3021	Известь	т	9	3
5152	Кирпич силикатный	м3	120	5

3 Напишите запрос, который увеличивает выполнение по всем объектам на 1%.
4 Напишите запрос, переводящий ресурсы, предназначенные для объекта " Жилой дом по ул.Тимирязева" на объект "Школа".
5 Напишите запрос, который выводит Наименование, Стоимость, Выполнение из таблицы Объект.
6 Напишите запрос, который вывел бы список всех поставок ресурсов для объекта Поликлиника.

7 Напишите запрос, который вывел бы таблицу Поставка_ресурсов со столбцами в обратном порядке.
8 Напишите запрос, извлекающий из таблицы Поставка_ресурсов список объектов. Объекты не должны повторяться.
9 Напишите запрос, выводящий наименование объекта и процент выполнения, для объекта имеющего самую высокую сметную стоимость.
10 Напишите запрос на создание списка, состоящего из Наименования ресурса и Наименования объекта, где он используется для всех объектов, выполнение по которым не более 60%.
11 Напишите запрос на удаление всех записей из таблицы Поставка_ресурсов, предназначенных для объекта №1.

Вариант 10. Завод-изготовитель поставляет нескольким получателям строительные изделия и конструкции. В базе данных должны содержаться сведения о:
а) получателях (данными о получателе являются его код, наименование, адрес, удаленность от завода);
б) поставках (данными о поставке являются ее шифр, наименование изделия, единица измерения, цена ед. измерения, получатель).

Получатель

Код	Наименование	Адрес	Удаленность
5241	ООО Гранит	ул.Ильинская, 30	102
3820	ЗАО Протект	ул.Должанская, 1	50
2450	ЧП Кулик	ул.Архангельская, 28	92
3054	ОАО Маяк	пр.Ленина, 49	72
1568	АО Строй-НН	пр.Гагарина, 37	28

Поставка

Шифр	Изделие	Ед_изм	Цена	Получатель
1238	Кирпич	шт.	8	2450
1237	Плитка	м2	420	1568
1247	ГВЛ	шт.	400	3820
7421	Ламинат	м2	999	3054
1241	Стеклопакет	шт.	14999	2450
5421	Гвозди	Кг	70	2450
3248	Шифер	шт.	230	3820

3 Напишите запрос, который увеличивает Цену всех поставок на 10 руб.
4 Напишите запрос, передающий поставки от "ЗАО Протект" в "ООО Гранит".
5 Напишите запрос, который выводит Наименование, Удаленность и Адрес из таблицы Получатель.
6 Напишите запрос, который вывел бы всю информацию о поставках "ЧП Кулик".
7 Напишите запрос, который вывел бы таблицу Поставка со столбцами в обратном порядке.

8 Напишите запрос, извлекающий из таблицы Поставка список получателей. Получатели не должны повторяться.
9 Напишите запрос, считающий среднюю цену поставок.
10 Напишите запрос на создание списка, состоящего из названия Изделия, и Наименования его получателя для всех получателей, которые расположены далее 70 км от завода.
11 Напишите запрос на удаление всех поставок получателя с кодом 2450.

Вариант 11. Строительная организация получает строительные изделия и материалы от нескольких поставщиков. В базе данных должны содержаться сведения о:
а) поставщиках (данными о поставщике являются его номер, индекс, наименование, адрес);
б) получаемых изделиях (данными об изделии являются его шифр, наименование, единица измерения, количество, поставщик).

Поставщик

№_поставщика	Наименование	Адрес	Количество_сотрудников
5241	ООО Гранит	ул.Ильинская, 30	30
3820	ЗАО Протект	ул.Должанская, 1	52
2450	ЧП Кулик	ул.Архангельская, 28	108
3054	ОАО Маяк	пр.Ленина, 49	24
1568	АО Строй-НН	пр.Гагарина, 37	185

Изделие

Шифр	Наименование	Ед_изм	Количество	Поставщик
1238	Шлакоблок	шт	50	1568
1237	Цемент	т	18	3820
1247	Стеновая панель	шт.	50	3820
7421	Труба	м	320	3054
1241	Дверной блок	шт.	260	2450
5421	Плита перекрытия	шт.	82	3820
3248	Оконный блок	шт.	240	2450

3 Напишите запрос, который сокращает Количество_сотрудников у всех поставщиков на 2.
4 Напишите запрос, увеличивающий количество изделий поставщика ЧП Кулик на 5 шт.
5 Напишите запрос, который выводит Наименование, Адрес и Количество_сотрудников из таблицы Поставщик.
6 Напишите запрос, который вывел бы список всех изделий, поставляемых ЗАО Протект.
7 Напишите запрос, который вывел бы таблицу Изделие со столбцами в обратном порядке.

8 Напишите запрос, извлекающий из таблицы Изделие список номеров поставщиков. Номера не должны повторяться.
9 Напишите запрос, выводящий наименование и адрес поставщика, где работает минимальное количество сотрудников.
10 Напишите запрос на создание списка, состоящего из Наименования изделия и его поставщика для всех поставщиков, количество сотрудников которых не менее 100.
11 Напишите запрос на удаление всех изделий, поставляемых поставщиком №2450.

Вариант 12. Строительные изделия и конструкции поставляются с нескольких заводов-изготовителей. В базе данных должны содержаться сведения:
а) об изделиях (данными об изделии являются его код, наименование, единица измерения, цена за единицу измерения (руб.));
б) поставках (данными о поставке являются ее шифр, наименование получателя, дата поставки, количество изделий в единице измерения, код изделия).

Изделие

Код	Наименование	Ед_изм	Цена
1247	Кирпич	1000 шт.	30000
7421	Плитка	м2	420
1241	ГВЛ	шт.	400
5421	Ламинат	м2	1000
3248	Стеклопакет	шт.	14000

Поставка

Шифр	Получатель	Дата	Количество	Изделие
5241	ЗАО Берег	01.12.2008	20	1247
3820	ЗАО Протект	01.12.2008	550	1241
2450	ЧП Кулик	03.12.2008	50	7421
3054	ОАО Маяк	03.12.2008	300	1241
1568	АО Строй-НН	04.12.2008	60	7421
1248	ОАО Парус	04.12.2008	70	5421
7452	ЗАО Берег	05.12.2008	230	3248

3 Напишите запрос, который увеличивает Цену всех изделий на 10 руб.
4 Напишите запрос, меняющий кирпич на плитку в соответствующих поставках.
5 Напишите запрос, который выводит Наименование, Ед_изм и Цену из таблицы Изделие.
6 Напишите запрос, который вывел бы всю информацию о поставках плитки.
7 Напишите запрос, который вывел бы таблицу Поставка со столбцами в обратном порядке.
8 Напишите запрос, извлекающий из таблицы Поставка список кодов изделий. Коды не должны повторяться.

9 Напишите запрос, считающий среднюю цену изделий.
10 Напишите запрос на создание списка, состоящего из названия получателя и наименования изделия для всех изделий, цена которых не более 500 руб.
11 Напишите запрос на удаление всех поставок изделия с кодом 7421.

Вариант 13. В строительном вузе преподаватели проводят занятия. В базе данных должны содержаться сведения о:
а) преподавателях (данными о преподавателе являются табельный номер, Ф.И.О., должность, оклад);
б) занятиях (данными о занятии являются номер занятия, название дисциплины, почасовая ставка оплаты (руб.), день недели, преподаватель).

Преподаватель

Таб_№	ФИО	Должность	Оклад
5241	Сидоров В.В.	профессор	20000
3820	Петров В.П.	доцент	12000
2450	Лисин А.Н.	доцент	12000
3054	Киров Д.О..	ст. преподаватель	10000
1568	Королева О.М.	ассистент	8000

Занятие

№_занятия	Дисциплина	Ставка	День_недели	Преподаватель
1238	Теоретическая механика	115	понедельник	2450
1237	Сопротивление материалов	135	понедельник	5241
1247	Математика	115	среда	3820
7421	Делопроизводство	75	среда	1568
1241	Теоретическая механика	115	вторник	2450
5421	Строительные материалы	95	вторник	3054
3248	Сопротивление материалов	135	среда	5241

3 Напишите запрос, который увеличивает почасовую ставку оплаты на 5 руб. для всех занятий.
4 Напишите запрос, передающий занятия от преподавателя Петрова В.П. Сидорову В.В.
5 Напишите запрос, который выводит ФИО, Должность и Оклад из таблицы Преподаватель.
6 Напишите запрос, который вывел бы информацию обо всех занятиях Сидорова В.В.

7 Напишите запрос, который вывел бы таблицу Занятие со столбцами в обратном порядке.
8 Напишите запрос, извлекающий из таблицы Занятие список номеров преподавателей. Номера не должны повторяться.
9 Напишите запрос, считающий средний оклад преподавателей.
10 Напишите запрос на создание списка, состоящего из названия Дисциплины и Дня недели, когда проводится занятие для всех преподавателей, оклад которых не менее 12 000 руб.
11 Напишите запрос на удаление всех занятий преподавателя с табельным номером 2450.

Вариант 14. В строительной компании ведется учет рабочего времени. Необходимо обеспечить начисление заработной платы. В базе данных должны содержать сведения о:
а) рабочих (данными о служащем являются табельный номер, ФИО, должность, тариф (руб./час);
б) карточки учета рабочего времени (они содержат номер карточки, дату, день недели, количество отработанных часов, номер рабочего).

Рабочий

Таб_№	ФИО	Должность	Тариф
3258	Морозова Анастасия Андреевна	штукатур	100
4342	Павлинов Алексей Владимирович	монтажник	120
5428	Пигалов Максим Александрович	монтажник	120
5321	Рязанцев Максим Вячеславович	крановщик	250
4418	Смирнов Сергей Александрович	каменщик	150

Карточка

№_карточки	Дата	День_недели	Часы	Рабочий
1238	01.12.2008	понедельник	8	3258
1237	01.12.2008	понедельник	8	4342
1247	03.12.2008	среда	8	5321
7421	03.12.2008	среда	4	4342
1241	04.12.2008	четверг	4	5321
5421	04.12.2008	четверг	8	4342
3248	05.12.2008	пятница	7	4448

3 Напишите запрос, который увеличивает тариф на 10 руб./час для всех рабочих.
4 Напишите запрос, меняющий записи в таблице Карточка, где указан в качестве исполнителя рабочий Павлинов Алексей Владимирович, на рабочего Пигалова Максима Александровича.
5 Напишите запрос, который выводит ФИО, Должность и Тариф из таблицы Рабочий.

6 Напишите запрос, который вывел бы все записи из таблицы Карточка, связанные с Рязанцевым Максимом Вячеславовичем.
7 Напишите запрос, который вывел бы таблицу Карточка со столбцами в обратном порядке.
8 Напишите запрос, извлекающий из таблицы Карточка список табельных номеров рабочих. Номера не должны повторяться.
9 Напишите запрос, считающий средний тариф рабочих.
10 Напишите запрос на создание списка, состоящего из названия Даты и Дня недели, когда работали рабочие с окладом не более 150 руб.
11 Напишите запрос на удаление всех записей из таблицы карточка, связанных с рабочим, имеющим табельный номер 3258.

Вариант 15. Поставщик продает товары различных производителей. Необходимо обеспечить работу системы обработки заказов. В базе данных должны содержаться сведения о:
а) товарах (данными о товаре являются код, наименование, единица измерения, цена единицы);
б) заказах (данными о заказе являются код, дата заказа, стоимость заказа, вид оплаты, код товара,).

Товар

Код	Наименование	Ед_изм	Цена
3258	Плита перекрытия	шт.	60000
4342	Стеновая панель	шт.	60000
5428	Лестничный марш	шт.	50000
5321	Ферма металлическая	шт.	350000
4418	Оконный блок	шт.	9000

Заказ

Код	Дата	Количество	Вид_оплаты	Товар
5466	01.12.2008	50	нал.	3258
7898	01.12.2008	18	безнал.	5321
1232	03.12.2008	50	безнал.	3258
4565	03.12.2008	320	нал.	4418
7535	04.12.2008	260	безнал.	4418
1595	04.12.2008	82	нал.	5428
8542	05.12.2008	240	безнал.	4342

3 Напишите запрос, который увеличивает цену всех товаров на 1 000 руб.
4 Напишите запрос, увеличивающий количество лестничных маршей во всех заказах, где они встречаются, на 5 шт.
5 Напишите запрос, который выводит Наименование, Ед_изм и Цену из таблицы Товар.
6 Напишите запрос, который вывел бы список всех заказов Оконных блоков.

7 Напишите запрос, который вывел бы таблицу Заказ со столбцами в обратном порядке.
8 Напишите запрос, извлекающий из таблицы Заказ список кодов товаров. Коды не должны повторяться.
9 Напишите запрос, выводящий дату и количество заказа для самого дешевого товара.
10 Напишите запрос на создание списка, состоящего из Даты заказа и Наименования товара для всех товаров не дороже 60000 руб.
11 Напишите запрос на удаление всех заказов товаров с кодом 3258.

Литература

1. Дунаев В. В., Базы данных. Язык SQL для студента. – СПб.: БХВ-Петербург, 2008. – 1280 с.
2. Харитонова И., Рудикова Л.,Microsoft Office Access 2007. – СПб.: БХВ-Петербург, 2008. – 312 с.
3. Корнелюк В.К., Веккер З.Е., Зиновьев Н.Б. ACCESS 97. – М: СОЛОН 1998, 482 с.
4. Горев А., Ахаян Р., Макашарипов С. Эффективная работа с СУБД – СПб.: Питер, 1997. – 704 с.

Содержание

Андрей Яковлевич Лахов,
Константин Анатольевич Сафонов

ИСПОЛЬЗОВАНИЕ SQL.

Методические указания к расчетной работе для студентов направлений 230200.62 Информационные системы, 080500.62 Менеджмент и специальности 080502.65 Экономика и управление на предприятии (в строительстве)

Подписано к печати _____. Бумага газетная. Формат 60 90 1/16. Печать офсетная. Уч-изд. л. _____. Усл. печ. л. _____. Тираж ___. экз. заказ №_____. Нижегородский Архитектурно-Строительный Университет, 603600, Н. Новгород , Ильинская, 65. Полиграфический центр ННГАСУ, 603600, Н. Новгород, Ильинская, 65.