

БАЗЫ ДАННЫХ

**Учебно-методическое пособие
к лабораторной работе**

«ФОРМИРОВАНИЕ ПРОСТЫХ ЗАПРОСОВ SELECT»

**для студентов направления
09.03.01 «Информатика и вычислительная техника»**

Составитель: А. А. Будаева

Владикавказ 2018

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ГОРНО-МЕТАЛЛУРГИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ)»

Кафедра автоматизированной обработки информации

БАЗЫ ДАННЫХ

Учебно-методическое пособие
к лабораторной работе

«ФОРМИРОВАНИЕ ПРОСТЫХ ЗАПРОСОВ SELECT»

для студентов направления
09.03.01 «Информатика и вычислительная техника»

Составитель А. А. Будаева

Допущено
редакционно-издательским советом
Северо-Кавказского горно-металлургического института
(государственного технологического университета).
Протокол заседания РИСа № от _____ г.

Владикавказ 2018

УДК 004.65
ББК 32.073.26-0.18.2
Б90

Рецензент: кандидат технических наук, доцент СКГМИ (ГТУ)
Соколова Е. А.

Б90 Базы данных: Учебно-методическое пособие к лабораторной работе «Формирование простых запросов SELECT» для студентов направления 09.03.01 «Информатика и вычислительная техника» / Сост.: А. А. Будаева; Северо-Кавказский горно-металлургический институт (государственный технологический университет). – Владикавказ: Северо-Кавказский горно-металлургический институт (государственный технологический университет). Изд-во «Терек», 2018. – 32 с.

В учебном пособии рассматриваются особенности использования оператора SELECT языка SQL; описаны правила использования специальных условий, правила написания многотабличных запросов, а также приемы форматирования списка вывода SELECT-запросов. Приведены индивидуальные задания и контрольные вопросы.

УДК 004.65
ББК 32.073.26-0.18.2

© Составление. ФГБОУ ВПО «Северо-Кавказский
горно-металлургический институт
(государственный технологический университет)», 2018
© Будаева А. А., составление, 2018

Подписано в печать _____ Формат 60x84 ¹/₁₆. Бумага офсетная. Гарнитура «Таймс».
Печать на ризографе. Усл. п.л. _____. Уч.-изд. л. 4,56. Тираж 20 экз. Заказ № _____.
Северо-Кавказский горно-металлургический институт
(государственный технологический университет). Издательство «Терек».
Отпечатано в отделе оперативной полиграфии СКГМИ (ГТУ).
362021, г. Владикавказ, ул. Николаева, 44.

Оглавление

Цель работы: 5

Теоретические сведения	5
1. Оператор SELECT	5
2. Форматирование списка вывода	5
3. Выборка данных	7
4. Операции соединения	9
5. Множественные операции	11
Задания на лабораторную работу	13
Контрольные вопросы	24
Литература	24
Приложение	26

ФОРМИРОВАНИЕ ПРОСТЫХ ЗАПРОСОВ SELECT

Цель работы:

1. обучение использованию оператора SELECT и некоторых функций для выполнения простых запросов к БД в режиме диалогового (интерактивного) SQL.
2. приобретение практических навыков форматирования списка вывода SELECT запросов.
3. изучение специальных условий в запросах: IN, BETWEEN, LIKE, IS NULL.
4. приобретение практических навыков написания запросов с использованием возможностей специальных условий:
5. приобретение практических навыков написания многотабличных запросов с использованием различных вариантов соединения таблиц:
 - внутреннее соединение
 - соединение таблицы с самой собой
 - внешнее соединение таблиц

Теоретические сведения

1. Оператор SELECT

Одним из наиболее часто используемых SQL операторов является оператор SELECT. Он применяется для выборки информации из таблиц.

Синтаксис команды SELECT

SELECT список вывода
FROM {таблица, представление}
[**WHERE** условие]
[**GROUP BY** список полей группировки]
[**HAVING** условие]
[**ORDER BY** список полей сортировки]

Примеры

1. Выборка отдельных столбцов
SELECT ProdName, Available FROM Product
2. Выборка всех столбцов
SELECT * FROM Product

2. Форматирование списка вывода

Сортировка данных

По умолчанию данные в запросах на выборку выводятся в том порядке, в котором они находятся в таблице.

Для сортировки данных при выводе используется предложение ORDER BY оператора SELECT. В нем указывается имя одного или нескольких столбцов, по которым необходимо отсортировать результаты. Рассмотрим следующий пример.

```
SELECT ProdID, ProdName, Available FROM Product ORDER BY Available
```

Предложение ORDER BY в примере позволяет отсортировать данные по возрастанию значений столбца Available.

Для сортировки по нескольким столбцам, нужно перечислить их названия через запятую. В следующем запросе выбираются три столбца, а результат сортируется по двум из них - сначала по доступному количеству, а потом по названию.

```
SELECT ProdID, ProdName, Available FROM Product  
ORDER BY Available, ProdName
```

В случае сортировки по нескольким столбцам порядок сортировки будет соответствовать порядку перечисления столбцов. Т.е., в вышеприведенном примере продукция сортируется по столбцу Available, а строки с одинаковым значением поля Available сортируются еще и по столбцу ProdName.

По умолчанию в предложении ORDER BY используется порядок сортировки по возрастанию. Если нужно получить данные в обратном порядке следует указать ключевое слово DESC непосредственно после имени сортируемого столбца. В следующем примере сортировка осуществляется по количеству продукта в убывающем порядке и по названию продукта (по возрастанию).

```
SELECT ProdID, ProdName, Available FROM Product  
ORDER BY Available DESC, ProdName
```

Вычисляемые поля

В списке вывода оператора SELECT помимо имен столбцов таблиц может добавлять и так называемые вычисляемые поля. Вычисляемые поля создаются SQL-оператором SELECT непосредственно при выполнении команды. Значения таких полей нигде не сохраняются. Рассмотрим следующий пример:

```
SELECT CustID AS 'Номер', FirstName + ' ' + LastName AS 'Фамилия и имя  
клиента'  
FROM Customer
```

Здесь в качестве вычисляемого поля используется выражение FirstName + ' ' + LastName. Значением такого выражения является конкатенация значений

полей таблицы (фамилия и имя клиента) и символа пробела. При выводе данное вычисляемое поле получит псевдоним 'Фамилия и имя клиента' с помощью ключевого слова AS. Псевдонимы можно назначать для любых столбцов списка вывода, в том числе и для обычных полей таблицы. Так, в примере выше, полю CustID назначен псевдоним 'Номер'. Вычисляемые поля могут содержать выражения различных типов, в том числе и математические операции. Рассмотрим *пример*:

```
SELECT ProdID, Qty, Price, Qty * Price AS 'Итоговая стоимость'  
FROM OrdItem WHERE OrdID = 1
```

Здесь выражение Qty * Price (произведение количества товара на цену) определяет итоговую стоимость каждого товара в заказе с кодом 1.

Устранение дубликатов записей

Устранение дубликатов записей может потребоваться если результат вывода запроса содержит одинаковые строки данных. Для исключения повторов используется ключевое слово DISTINCT. Оно указывается непосредственно перед списком вывода сразу после SELECT. Например, требуется получить список фамилий клиентов (без повторов).

```
SELECT DISTINCT LastName FROM Customer
```

3. Фильтрация данных

Запросы, извлекающие всю информацию из таблиц, встречаются на практике довольно редко. Чаще запросы осуществляют поиск информации (например, данные для отчетов, сводок и т.д.). Для заданий условий отбора данных в команде SELECT используется предложение WHERE.

```
SELECT ProdID, ProdName, Available FROM Product WHERE Available = 0
```

Данный запрос извлекает товары, которых не осталось на складе (Available = 0).

Если в одном запросе присутствует и часть WHERE и ORDER BY, то часть ORDER BY всегда указывается последней (т.к. сортируется уже подготовленный вывод).

В качестве условных операций в запросах можно использовать следующие:

- Операции сравнения:
 - <>, != (неравно)
 - =, <, <=, >, >= (равно, меньше, меньше равно, больше, больше равно)
 - !>, !< (не больше, не меньше)
- Специальные условия

IS NULL – проверка на значение NULL

BETWEEN min AND max – проверка на принадлежность диапазону

IN (список) – проверка на вхождение в список значений (значения записываются через запятую)

LIKE “шаблон” – проверка строкового значения на соответствие шаблону

Также можно использовать логические операторы NOT, AND и OR. Предложения WHERE могут содержать любое количество логических операторов AND и OR. Комбинируя их можно создавать сложные фильтры. Однако при комбинировании ключевых слов AND и OR необходимо учитывать, что оператор AND выполняется раньше оператора OR, т.е. имеет более высокий приоритет. Изменить приоритет можно с помощью круглых скобок.

Пример – IS NULL: Найти всех клиентов, у которых контактный телефон неизвестен

```
SELECT FirstName, LastName, Phone FROM Customer WHERE PHONE IS NULL
```

Пример – BETWEEN: Найти товары, количество которых на складе находится в диапазоне [5 .. 10]:

```
SELECT ProdID, ProdName, Available FROM Product  
WHERE Available BETWEEN 5 AND 10
```

Пример – IN: Найти клиентов с фамилиями Иванов или Петров, у которых не указан контактный телефон:

```
SELECT FirstName, LastName, Phone FROM Customer  
WHERE LastName IN ('Иванов','Петров') AND PHONE IS NULL
```

Обращаем внимание на то, что строковые константы заключаются в одинарные кавычки.

При использовании логического оператора **NOT** важно помнить, что он вставляется не перед названием столбца, а после.

Пример – NOT: Найти всех клиентов, за исключением Ивановых и Петровых, у которых известен контактный телефон:

```
SELECT FirstName, LastName, Phone FROM Customer  
WHERE NOT LastName IN ('Иванов','Петров') AND NOT PHONE IS NULL
```

Условие LIKE

Условие LIKE используется для проверки соответствия строки заданному шаблону. В шаблоне можно указывать как обычные символы, так и символы-шаблоны. Шаблон может включать в себя следующие символы-шаблоны:

% - символ заменяет любую последовательность символов

_ - символ заменяет любой одиночный символ

[] – любой одиночный символ из указанных в квадратных скобках (в наборе, например, [абвАБВ] или в диапазоне, например, [а-мА-М])

[^] – любой одиночный символ кроме указанных в квадратных скобках (в наборе, например, [^абвАБВ] или в диапазоне, например, [^а-мА-М])

Пример: Найти всех клиентов с именами на букву 'И'

```
SELECT * FROM Customer WHERE FirstName LIKE 'И%'
```

Пример: Найти всех клиентов с фамилиями, состоящими из шести букв и заканчивающихся на сочетание 'етров' (Петров, Ветров и т.п.).

```
SELECT * FROM Customer WHERE LastName LIKE '_етров'
```

Пример: Найти всех клиентов фамилии, которых заканчиваются на 'омов' и начинаются на любую букву в промежутке от 'Л' до 'С', например Ломов, Ромов, Сомов и т.п.

```
SELECT * FROM Customer WHERE LastName LIKE '[Л-С]омов'
```

Пример: Найти всех клиентов фамилии, которых начинаются на 'ив' и третья буква любая, кроме 'а'

```
SELECT * FROM Customer WHERE LastName LIKE 'ив[^а]%'
```

4. Многотабличные запросы.

Соединение таблиц – это операция, которая приводит к соединению всех пар строк из двух и более таблиц, для которых выполняется заданное условие.

Соединение таблиц используется, когда необходимо вывести:

- значения столбцов разных таблиц;
- значения столбцов одной таблицы, отвечающих условию, заданному в другой таблице.

Операция соединения может быть записана двумя способами:

- неявно, как выборка из декартова произведения таблиц – в этом случае условие соединения записывается в части WHERE запроса;
- явно – условие соединения указывается непосредственно в части FROM

Декартово произведение двух таблиц может получено средствами языка SQL двумя способами:

- 1) FROM таблица1, таблица2
- 2) FROM таблица1 CROSS JOIN таблица2

Тогда соединение по предикату можно записать по схеме:

FROM таблица1 CROSS JOIN таблица2 WHERE предикат

Если соединяемые таблицы содержат одноименные поля, в запросе требуется использовать уточняющие имена: «таблица.поле». На практике для задания более лаконичных имен часто используют короткие псевдонимы таблиц, по которым можно сослаться на них в любых других местах запроса. Псевдоним указывается непосредственно после имени таблицы через пробел в предложении FROM.

Пример. Получить список клиентов с указанием названия города проживания:

```
SELECT k.FirstName, k.LastName, City.CityName  
FROM Customer k, City WHERE k.CityID = City.CityID
```

Помимо условий соединения в многотабличных запросах могут присутствовать и обычные условия отбора.

Пример: Найти всех клиентов из Москвы с фамилией Петров:

```
SELECT K.CustID, k.FirstName FROM Customer k, City  
WHERE ck.CityID = c.CityID AND k.LastName = 'Петров' AND  
c.CityName = 'Москва'
```

Соединения с использованием фразы FROM

На практике рекомендуется использовать явные соединения таблиц. В этом случае условия соединения указываются непосредственно при перечислении имен таблиц в части FROM в соответствии с приведенной ниже схемой:

таблица [INNER | {FULL | LEFT | RIGHT} [OUTER]] JOIN таблица {ON
условие}

Соединение может быть либо внутренним (**INNER**), либо одним из внешних (**OUTER**). Служебные слова **INNER** и **OUTER** можно опускать, поскольку внешнее соединение однозначно определяется его типом — **LEFT** (левое), **RIGHT** (правое) или **FULL** (полное), а просто JOIN будет означать внутреннее соединение.

Внутреннее соединение (JOIN – по умолчанию) означает, что в результирующий набор попадут только те соединения строк двух таблиц, для которых значение предиката равно TRUE. Условие соединения указывается после ключевого слова ON, а дополнительные условия для наглядности помещают в часть WHERE.

Пример: Получить список всех клиентов из Казани с фамилией Иванов:

Способ 1: дополнительные условия в части WHERE

```
SELECT K.CustID, k.FirstName  
FROM Customer k INNER JOIN City c ON k.CityID = c.CityID  
WHERE k.LastName = 'Иванов' AND c.CityName = 'Казань'
```

Способ 2: дополнительные условия в части ON

```
SELECT K.CustID, k.FirstName  
FROM Customer k INNER JOIN City c ON k.CityID = c.CityID  
AND k.LastName = 'Иванов' AND c.CityName = 'Казань'
```

Внешнее соединение означает, что в результирующий набор попадают также соединения строк таблиц, для которых условие соединения не выполняются. Внешние соединения отличаются направлением:

- **LEFT/RIGHT JOIN** (левое/правое внешнее соединение) – включает в результирующий запрос данные внутреннего соединения таблиц, а также те строки левой (правой) таблицы, для которых условие соединения не выполняется. Недостающие данные полей из правой (левой) таблицы заполняются значением NULL;
- **FULL JOIN** (полное внешнее соединение) – оставляет в запросе данные внутреннего соединения обеих таблиц, а также не включенные в него строки и первой и второй таблиц (дополненные значениями NULL для отсутствующих столбцов).

Пример: Вывести полный список городов и фамилии клиентов, которые в них проживают. Если в городе нет ни одного клиента, в соответствующем столбце вывести значение NULL.

```
SELECT c.CityName, k.FirstName  
FROM City c LEFT JOIN Customer k ON c.CityID = k.CityID  
ORDER BY c.CityName
```

5. Операторы работы с наборами

Выделяют три оператора работы с наборами: объединение – **UNION**, пересечение – **INTERSECT**, разность – **EXCEPT**. Аргументами этих

операций служат SELECT-запросы. В отличие от операции соединения, данные операторы не изменяют схему исходного отношения.

Операция *объединение* помещает в результирующую таблицу данные двух запросов. Операция *пересечения* помещает в результирующую таблицу строки, имеющиеся как в первом, так и во втором запросах. Операция *разность* помещает в результирующую таблицу строки первого запроса, которых нет во втором запросе.

В стандарте SQL операторы работы с наборами имеют следующий синтаксис:

```
оператор SELECT
  {UNION | INTERSECT | EXCEPT} [DISTINCT | ALL]
оператор SELECT
```

Основное требование для применения данных операторов – является совместимость по объединению. Отношения совместимы по объединению если:

- они имеют одинаковое число полей
- соответствующие пары столбцов являются значениями одинаковых или совместимых типов

Ключевые слова ALL и DISTINCT (по умолчанию) указывают, допускаются ли в результирующей таблице повторяющиеся строки или нет.

Пример. Получить список товаров, с указанием статуса «В наличии», «Заканчивается» (если не больше 10) или «Нет на складе» в зависимости от значения поля Available.

```
SELECT ProdID, ProdName, «В наличии» FROM Product
WHERE Available > 10
UNION
SELECT ProdID, ProdName, «Заканчивается» FROM Product
WHERE Available BETWEEN 1 AND 10
UNION
SELECT ProdID, ProdName, «Нет на складе» FROM Product
WHERE Available = 10
```

Пример. Получить имена, фамилии и адреса клиентов, у которых были заказы на товары как стоимостью 1000 руб., так и стоимостью больше 100000 руб.

```
SELECT c.CustID, FirstName, LastName, Address
FROM Customers c JOIN Order o ON c.CustID=o.CustID
      JOIN OrdItem oi ON o.OrdID=oi.OrdID WHERE Price =1000
INTERSECT
SELECT c.CustID, FirstName, LastName, Address
FROM Customers c JOIN Order o ON c.CustID=o.CustID
      JOIN OrdItem oi ON o.OrdID=oi.OrdID WHERE Price =100000
```

Пример. Получить имена, фамилии и телефоны клиентов, у которых были заказы только товаров стоимостью больше 100000 руб.

```
SELECT c.CustID, FirstName, LastName, Phone
FROM Customers c JOIN Order o ON c.CustID=o.CustID
      JOIN OrdItem oi ON o.OrdID=oi.OrdID WHERE Price > 100000
EXCEPT
SELECT c.CustID, FirstName, LastName, Phone
FROM Customers c JOIN Order o ON c.CustID=o.CustID
      JOIN OrdItem oi ON o.OrdID=oi.OrdID WHERE Price !> 100000
```

В случае, если в запросе используются несколько операторов работы с наборами, они выполняются в порядке следования. Порядок выполнения можно изменить с помощью скобок.

Пример. Получить имена, фамилии и телефоны клиентов, у которых были заказы только товаров стоимостью больше 100000 руб. В результирующем наборе не выводить клиентов из Москвы

```
( SELECT c.CustID, FirstName, LastName, Phone
  FROM Customers c JOIN City ct ON c.CityID=ct.CityID
  WHERE CityName = 'Москва'
)
INTERSECT
( SELECT c.CustID, FirstName, LastName, Phone
  FROM Customers c JOIN Order o ON c.CustID=o.CustID
      JOIN OrdItem oi ON o.OrdID=oi.OrdID WHERE Price > 100000
EXCEPT
  SELECT c.CustID, FirstName, LastName, Phone
  FROM Customers c JOIN Order o ON c.CustID=o.CustID
      JOIN OrdItem oi ON o.OrdID=oi.OrdID WHERE Price !> 100000
)
```

Задания на лабораторную работу

Составить SQL-запросы по их заданному содержательному описанию, в соответствии с вариантом. Во всех нижеперечисленных заданиях используется БД Университет (см. Приложение).

Варианты лабораторных заданий

Номер варианта	№№ Задания						
	1	2	3	4	5	6	7
1	3	6	9	12	15	4	7

2	4	7	10	13	1	5	8
3	5	8	11	14	2	6	9
4	6	9	12	15	3	7	10
5	7	10	13	1	4	8	11
6	8	11	14	2	5	9	12
7	9	12	15	3	6	10	13
8	10	13	1	4	7	11	14
9	11	14	2	5	8	12	15
10	12	15	3	6	9	13	1
11	13	1	4	7	10	14	2
12	14	2	5	8	11	15	3
13	15	3	6	9	12	1	4
14	1	4	7	10	13	2	5
15	2	5	8	11	14	3	6

Задание № 1. Выражения во фразе SELECT

В заданиях ниже *зарплата* – это сумма (salary + commission), salary – ставка (оклад), commission – надбавка.

1	По каждому преподавателю выведите его имя и сумму hiredate + salary (столбец с именем «Дата»). Сформулируйте назначение столбца «Дата»
2	По каждому преподавателю выведите его имя и разность между датой приема на работу и деления ставки на надбавку (столбец с именем «Выражение»). Что получается в результате вычисления hiredate - (Salary/Commission)?
3	Получить двумя способами информацию о факультетах в следующем виде: Факультет <название> расположен в корпусе <корпус> имеет фонд финансирования <фонд> 1. все поля в виде одного столбца «Информация о факультетах» 2. все данные выводятся в 6 столбцах: Константа1, <Название>, Константа2, <Корпус>, Константа3, <Фонд>
4	Получить двумя способами информацию о кафедрах в следующем виде: Кафедра <название кафедры> с первичным ключом <номер первичного ключа> имеет удвоенный фонд финансирования <фонд финансирования, умноженный на 2> 1. все поля в виде одного столбца с названием «Информация о кафедрах» 2. все данные в шести столбцах с именами: «Константа1», «Кафедра», «Константа2», «Номер первичного ключа», «Константа3», «Финансирование»
5	Получить двумя способами информацию о преподавателях в следующем виде: <должность преподавателя> <имя преподавателя> принят на работу <дата приема> и имеет зарплату <сумма ставки и надбавки>

	<p>1. все данные в виде одного столбца с названием «Информация о преподавателях»</p> <p>2. все данные в шести столбцах с именами: «Должность», «Преподаватель», «Константа1», «Дата», «Константа2», «Зарплата».</p>
6	<p>Получить двумя способами информацию об аудиториях в следующем виде:</p> <p>Аудитория <номер аудитории> находится на <номер этажа> этаже корпуса <номер корпуса> и имеет <количество мест> мест</p> <p>1. все данные в виде одного столбца с названием «Информация об аудиториях»</p> <p>2. все данные в девяти столбцах с именами: «Литерал1», «Аудитория», «Литерал2», «Этаж», «Литерал3», «Корпус», «Литерал4», «К-во мест», «Литерал5»</p>
7	<p>Получить двумя способами информацию о лекциях в следующем виде:</p> <p>На <неделя> неделе в <название дня недели> на <номер пары> проводится <тип занятия></p> <p>1. все данные в виде одного столбца с именем «Информация о лекциях»</p> <p>2. все данные в восьми столбцах с именами: «Литерал1», «Неделя», «Литерал2», «День», «Литерал3», «Пара», «Литерал4», «Тип».</p>
8	<p>Получить двумя способами информацию о дисциплинах в следующем формате:</p> <p>Название дисциплины в БД: <название дисциплины> Название дисциплины прописными буквами: <название дисциплины прописными буквами> Название дисциплины строчными буквами: <название дисциплины строчными буквами></p> <p>1. в виде одного столбца с именем «Информация о дисциплинах»</p> <p>2. в шести столбцах с именами: «Константа1», «Дисциплина1», «Константа2», «Дисциплина2», «Константа3», «Дисциплина3»</p>
9	<p>Получить двумя способами информацию о группах в следующем виде:</p> <p>Группа <номер группы> курса <номер курса> с количеством студентов <количество студентов в группе> имеет рейтинг <рейтинг группы></p> <p>1. в виде одного столбца с именем «Сведения о группах»</p> <p>2. в восьми столбцах с именами: «Константа1», «Группа», «Константа2», «Курс», «Константа3», «К-во студентов», «Константа4», «Рейтинг»</p>
10	<p>Получить двумя способами информацию о группах в следующем виде:</p> <p>На курсе <номер курса> имеется группа <номер группы> у которой рейтинг <рейтинг></p> <p>1. в виде одного столбца с названием «Информация о группах»</p> <p>2. в шести столбцах с именами: «Константа1», «Курс», «Константа2», «Группа», «Константа3», «Рейтинг»</p>

11	<p>Получить двумя способами информацию о преподавателях в следующем виде:</p> <p>Принятый на работу <дата приема на работу> <должность преподавателя> <имя преподавателя>, имеет ставку <ставка> и надбавку <надбавка> Его телефон <номер телефона></p> <p>1. в виде одного столбца с названием «Информация о преподавателях»</p> <p>2. в девяти столбцах с именами: «Константа1», «Дата», «Должность», «Преподаватель», «Константа2», «Ставка», «Константа3», «Надбавка», «Константа4», «Телефон»</p>
12	<p>Получить двумя способами информацию о преподавателях в следующем виде:</p> <p><дата приема на работу> был принят на работу <имя преподавателя> на должность <должность преподавателя> и с зарплатой <зарплата преподавателя></p> <p>1. в виде одного столбца с именем «Сведения о преподавателях»</p> <p>2. в семи столбцах с именами: «Дата», «Константа1», «Преподаватель», «Константа2», «Должность», «Константа3», «Зарплата»</p>
13	<p>Получить двумя способами информацию об университетах в следующем виде:</p> <p>Университет <Название> расположен в городе <Город> имеет рейтинг <Рейтинг></p> <p>1. в виде одного столбца с именем «Сведения об университетах»</p> <p>2. в шести столбцах с именами: «Константа1», «Название_университета», «Константа2», «Город», «Константа3», «Рейтинг»</p>
14	<p>Получить двумя способами информацию о студентах в следующем виде:</p> <p>Студент <Фамилия> <Имя> родился <Дата рождения> в городе <Город></p> <p>1. в виде одного столбца с именем «Сведения о студентах»</p> <p>2. в семи столбцах с именами: «Константа1», «Фамилия студента», «Имя студента», «Константа2», «Дата рождения», «Константа3», «Город».</p>
15	<p>Получить двумя способами информацию о преподавателях в следующем виде:</p> <p>Преподаватель <Имя> имеет зарплату <выражение1> процент ставки по отношению к зарплате <выражение2> процент надбавки по отношению к зарплате <выражение3></p> <p>1. в виде одного столбца с именем «Данные о зарплате преподавателей»</p> <p>2. в восьми столбцах с именами: «Константа1», «Фамилия преподавателя», «Константа2», «Зарплата», «Константа3», «Результат_выражения_2», «Константа4», «Результат_выражения_3».</p>

Задание № 2. Использование сложных логических условий

1	Вывести преподавателей факультета информационных технологий со ставкой больше 12000 ИЛИ преподавателей факультета электронной техники со ставкой больше 15000
2	Вывести преподавателей, которые НЕ удовлетворяют следующему условию: преподаватель работает на факультете, расположенном в корпусе 4, и имеет должность ассистента ИЛИ работает на факультете, расположенном в корпусе 6, и имеет должность доцента
3	Вывести данные по группам факультета с фондом финансирования в диапазоне 100000-200000 и рейтингом больше 20 ИЛИ группы факультета из корпуса 9 и с рейтингом меньше 50 ИЛИ группы факультета из корпуса 7 и с рейтингом больше 60
4	Вывести рейтинги и номера групп, для которых НЕ выполняется следующее условие: они относятся к факультету электронной техники и курс не равен 5 ИЛИ они находятся на факультете, фонд финансирования которого в диапазоне 150000-300000, а рейтинг больше 40 ИЛИ они находятся на факультете с фондом финансирования больше 200000 и имеют рейтинг в диапазоне 20-50
5	Вывести данные по преподавателям, для которых выполняется следующее условие: эти преподаватели – кураторы на первом курсе или групп с рейтингом больше 15 И они имеют должность профессора или доцента И у них зарплата меньше 10000 или больше 20000.
6	Вывести данные по преподавателям, которые НЕ удовлетворяют следующему условию: они являются кураторами групп кафедры с фондом финансирования меньше 200000 или больше 300000 И они являются кураторами групп с рейтингом больше 15 или групп курса 5 И они имеют зарплату между 10000-15000 или между 20000-25000
7	Вывести данные по преподавателям, удовлетворяющих следующему условию: их зарплата больше 1000 или они были приняты на работу после 01.01.2001 И заведующие их кафедр имеют зарплату больше 3000 или меньше 2500 И деканы их факультетов являются профессорами или доцентами.
8	Вывести данные по преподавателям, НЕ удовлетворяющих следующему условию: они были приняты на работу в диапазоне 01.01.2000 – 31.12.2005 или они являются доцентами или ассистентами И деканы их факультетов получают зарплату либо между 1200-1500 либо между 1700-2000 И они являются кураторами групп с рейтингом либо меньше 40 либо

	больше 60.
9	Вывести данные по преподавателям, которые удовлетворяют следующему условию: дата их приема на работу находится в диапазоне 02.03.1999–31.12.2007 или они являются профессорами И их непосредственный руководитель имеет зарплату в диапазоне 1200-1500 или он был принят на работу после 01.12.1998 И их непосредственный подчиненный является ассистентом или преподаватель или доцент
10	Вывести данные по преподавателям, которые НЕ удовлетворяют следующему условию: их ставка больше 10000 или их надбавка больше одной четвертой части ставки И их непосредственный руководитель был принят на работу либо в диапазоне дат 01.01.2005-31.12.2006 либо в диапазоне дат 01.01.2008-07.08.2009 И их непосредственный подчиненный либо является ассистентом либо имеет зарплату в диапазоне 20000-30000
11	Вывести дисциплины, которые удовлетворяют следующему условию: дисциплина читается профессорами, доцентами или ассистентами И дисциплина читается на 1 или 2 курсе И дисциплина читается в понедельник, четверг или пятницу
12	Вывести дисциплины, которые НЕ удовлетворяют следующему условию: дисциплина читается преподавателями кафедры 'АОИ' или кафедры 'программирование' И дисциплина читается в аудитории 313 корпуса 6 или в аудитории 202 корпуса 5 И дисциплина читается в понедельник первой недели или вторник второй недели
13	Вывести номера аудиторий и их корпуса, которые удовлетворяют следующему условию: они имеют вместимость в диапазоне 20-30 или в диапазоне 50-70 И в них проводятся занятия в группах факультета информационных технологий или электронной техники И в них проводят занятия преподаватели доценты или ассистенты
14	Вывести номера аудиторий и их корпуса, которые НЕ удовлетворяют следующему условию: они находятся на любом из этих этажей: '1', '3', '4', '5', '7', '9', '10' И в них проводят занятия преподаватели из кафедр 'АОИ' или 'программирование' И в них проводят занятия для студентов кафедр 'АОИ' или 'ИС'
15	Вывести преподавателей и названия читаемых ими дисциплин, которые удовлетворяют следующему условию: преподаватель принят на работу в диапазоне 01.01.1996-31.12.1998 или 01.01.2000-07-08-2005 И дисциплины читаются в аудитории 309 корпуса 6 или в аудитории 202 корпуса 5 И занятия по этим дисциплинам проводятся в понедельник первой недели или четверг второй недели И эти дисциплины читаются группам кафедр,

	расположенных в корпусах 5, 6 или 7
--	-------------------------------------

Задание № 3. Использование предиката BETWEEN

1	Вывести кафедры, которые относятся к факультету информационных технологий или к факультету с фондом финансирования от 500000 до 1000000
2	Вывести преподавателей факультета информационных технологий с зарплатой с 10000 до 15000
3	Вывести кафедры, на которых непосредственные подчиненные заведующих кафедр имеют зарплату от 20000 до 30000
4	Вывести кафедры факультета информационных технологий, непосредственные подчиненные заведующих которых (кафедр) приняты на работу с 01.10.2001 по 31.12.2001
5	Вывести группы факультета информационных технологий, кураторы которых (групп) были приняты на работу в одном из следующих диапазон дат: 01.06.2005 – 01.09.2006, 01.02.2003- 01.04.2004
6	Вывести кафедры, в группах которых проводят занятия типа 'лекция' преподаватели, принятые на работу с 01.01.2001 по 01.01.2002
7	Вывести факультеты, преподаватели-профессора которых имеют занятия не в корпусах с 3 по 5
8	Вывести группы факультета информационных технологий с количеством студентов от 27 до 32 или с рейтингом от 25 до 45
9	Вывести преподавателей, являющихся кураторами групп кафедры ИПО, с зарплатой от 9000 до 17000
10	Вывести группы факультета информационных технологий, кураторы которых имеют зарплату от 9000 до 17000
11	Вывести факультеты, на которых работают преподаватели профессора с зарплатой от 10000 до 15000
12	Вывести кафедры факультета информационных технологий, на которых работают преподаватели-доценты с зарплатой от 10000 до 15000
13	Вывести преподавателей кафедры АОИ, являющихся кураторами групп с рейтингом от 20 до 30
14	Вывести группы и преподавателей, которые проводят занятия в этих группах в аудиториях, расположенных в корпусе 6 на этажах с 3 по 7
15	Вывести преподавателей, ведущих занятия в группах, с количеством студентов в диапазоне 15-35 и в аудиториях с количеством мест в диапазоне 25-50

Задание № 4. Использование предикатов IN и LIKE

1	Вывести преподавателей-профессоров, которые работают на факультетах, расположенных в корпусах 2, 4, 7,10, а фамилии которых
---	---

	не содержат буквы Ф, Н, Р
2	Вывести данные по кафедрам факультета информационных технологий, на которых имеются группы с рейтингами 50, 55, 69, 73, 84, в названии которых вторая буква А О или П
3	Вывести дисциплины, которые ведут преподаватели-профессора кафедр, расположенных в корпусах 1, 4, 7, 9, 11, в названии которых вторая буква не А, Б или В
4	Вывести названия факультетов, на которых 1) имеются преподаватели, принятые на работу в одну из следующих дат: 01.09.2006, 01.05.2003, 01.08.2005, 01.01.2009, 2) или имеющих размер ставки от 5000 до 8000. Выводить только те факультеты, название которых содержит не более 10 букв, а вторая и предпоследняя буквы из набора О, Е, В
5	Вывести данные по непосредственным подчиненным заведующего кафедры АОИ, которые были приняты на работу в одну из следующих дат: 02.04.2000, 15.02.2002 или 10.12.2003, а их имена не содержат латинских букв
6	Вывести данные по кафедрам факультета, деканом которого является Абаков, непосредственные подчиненные заведующих этих кафедр имеют одну из следующих ставок: 10000, 15000, 20000, 25000, 30000. Выводимые кафедры должны удовлетворять шаблону: не более 15 букв, не содержит буквы М, Н, Л
7	Вывести фамилии преподавателей-доцентов, которые проводят занятия типа 'лабораторная работа' в одной из следующих аудиторий корпуса 4: 509, 511, 513, 515, 527. Имена выводимых преподавателей должны удовлетворять шаблону: не более 7 букв, начинаются на гласную, заканчиваются не на М, Н, Л
8	Вывести аудиторий и их корпуса, в которых преподаватели-ассистенты проводят занятия типа 'лабораторная' в один из следующих дней: понедельник, среда, четверг, суббота. Выводить данные только тех преподавателей, в именах которых нет букв С, О, У, а вторая буква И
9	Вывести информацию по преподавателям-профессорам, которые проводят занятия типа 'лекция' по дисциплинам, отличающимся от следующих: Программирование, С, JAVA, ООП. Имена выводимых преподавателей должны удовлетворять шаблону: не более 6 букв, начинаются на согласную, содержат буквы А, Я, М
10	Вывести информацию по факультетам, на которых читаются дисциплины, отличающиеся от следующих: «Вычислительная математика», «Технологии программирования», «Дискретная математика». Выводить данные только тех факультетов, в названии которых нет букв П, В, О, а третья и четвертая буквы ЕР
11	Вывести информацию по преподавателям-доцентам, читающим лекции по дисциплине «Базы данных» и работающим на факультете, в названии которого имеется подстрока 'ые на', но первая буква не из А, Р, Е

12	Вывести данные по кафедрам, преподаватели которых являются кураторами групп, в названии прикрепленных кафедр которых имеется буква 'п', начало не из списка У К Р
13	Вывести информацию по деканам факультетов, имена заведующих кафедр которых не содержат подстроки 'петр', но эти буквы по отдельности в названии есть
14	Вывести данные по кафедрам, группам которых на 3 курсе читаются дисциплины, в названии которых отсутствует подстрока 'ого_обесп', а последний символ совпадает с Е или Я
15	Вывести данные по кафедрам, группам которых на 3 курсе преподаются дисциплины, в названии которых имеется следующая подстрока: буквы 'с' и 'д' разделены любыми двумя символами, а после буквы 'д' может стоять А, О или Е .

Задание № 5. Соединение таблиц

1	Вывести дисциплины, которые читаются преподавателями-доцентами, с указанием названия факультетов, на которых они работают
2	Вывести дисциплины, которые читаются преподавателями-доцентами кураторами 1 курса, с указанием названия факультетов, на которых они работают
3	Для каждой группы 3 курса вывести дисциплины, занятия по которым проводятся в 5 корпусе
4	Вывести аудитории, в которых проходят лекции по вторникам второй недели, с указанием кафедр преподавателей, ведущих занятия
5	Вывести аудитории корпуса 4, в которых проводятся занятия у первого курса, с указанием названия кафедр преподавателей, ведущих эти занятия.
6	Вывести данные по группам 3-го курса, которым проводят лабораторные работы преподаватели факультета информационных технологий
7	Вывести данные по преподавателям-доцентам, которые читают лекции студентам факультета электронной техники (из списка исключить преподавателей данного факультета)
8	Вывести расписание занятий преподавателей, которое содержит: имя преподавателя, название дисциплины, аудиторию, корпус, название группы, курс, неделя (1 или 2), день недели, номер пары, тип занятия.
9	Вывести загруженность аудитории 6-313. Вывод содержит следующие данные: Неделя, день недели, номер пары, тип занятия, группа, курс, имя преподавателя
10	Вывести дисциплины, которые читаются преподавателями-профессорами факультета информационных технологий студентам 3 курса, с указанием места проведения занятий

11	Вывести аудитории, в которых проводятся лабораторные занятия для студентов факультета информационных технологий, с указанием с дисциплин и имен преподавателей
12	Вывести группы, аудитории с корпусами и имена преподавателей кафедры АОИ, которые читают этим группам лекции по дисциплине 'Базы данных'.
13	Вывести данные по преподавателям-доцентам и дисциплины, которые они ведут студентам 3-го курса факультета электронной техники
14	Вывести данные по преподавателям-доцентам (кураторам групп 3-го курса) факультета электронной техники, читающих лекции по дисциплине 'Базы данных'.
15	Вывести номера групп и их курсов кафедры АОИ, которые имеют кураторов-ассистентов, ведущих практические занятия в корпусе 1

Задание № 6. Многократное соединение одной и той же таблицы

1	Вывести данные по подчиненным каждого преподавателя, который в свою очередь является подчиненным Петрова
2	Вывести информацию по преподавателю, который в свою очередь является руководителем непосредственного руководителя преподавателя Иванова
3	Вывести данные по кафедрам, преподаватели которых ведут занятия у студентов кафедры АОИ
4	Вывести данные по факультетам, преподаватели которых ведут занятия у студентов кафедры АОИ
5	Вывести названия кафедр, у студентов которых ведут занятия преподаватели кафедры АОИ
6	Вывести информацию по факультетам, студентам которых ведут занятия преподаватели кафедры АОИ
7	Вывести данные по факультетам, группы которых имеют кураторов с факультета информационных технологий
8	Вывести данные по факультетам, преподаватели которых являются кураторами групп факультета информационных технологий
9	Вывести данные по преподавателям кафедры, заведующим которой является профессор Брюлов
10	Вывести заведующего кафедрой, к которой относится преподаватель ассистент Шишкин
11	Вывести имена преподавателей факультета, непосредственным начальником декана которого является профессор Сидоров
12	Вывести декана факультета, в котором работает непосредственный руководитель преподавателя доцента Василенко
13	Получить данные по преподавателям, которые ведут занятия у студентов группы, куратором которой является Трушин
14	Для каждой из групп, которым преподает доцент Васильев, выдать данные по кураторам

15	Вывести информацию по группам, у которых ведет преподаватель, являющийся куратором группы с идентификатором 5600
----	--

Задание № 7. Внешнее соединение, соединение не по предикату равенства

1	Получить полный список кафедр университета, с указанием имен преподавателей, которые там работают. Если на кафедре нет преподавателей, в соответствующем поле выводить значение NULL
2	Получить полный список кафедр университета, с указанием номеров и курсов групп, которые к ним прикреплены. Если у кафедры нет групп, в соответствующих столбцах выводится значение 0. Результирующая таблица имеет следующие поля: <i>Кафедра, Группа, Курс</i>
3	Получить имена преподавателей, номера групп, в которых они являются кураторами, а также номера курсов этих групп. При этом в списке должны быть все преподаватели, независимо от того, являются они кураторами групп или нет. Если преподаватель не куратор, в соответствующих полях выводить значение 0.
4	Получить пары имен факультетов, таких что: декан факультета_1 является подчиненным декана факультета_2.
5	Получить пары номеров групп, для которых выполняется следующее условие: курс группы_1 старше, а ее рейтинг больше по сравнению с курсом и рейтингом группы_2
6	Получить пары номеров групп вместе с их курсами, для которых выполняется следующее условие: фонд финансирования факультета, к которому относится группа_1 превышает фонд финансирования факультета группы_2 больше, чем на 200000.
7	Получить пары номеров групп, таких что: декан факультета, к которому относится группа_1 имеет заработок больше, чем декан факультета группы_2
8	Получить пары имен преподавателей, для которых выполняется следующее условие: - преподаватель_1 – профессор, преподаватель_2 – доцент - зарплата преподавателя_1 больше зарплаты преподавателя_2 больше, чем в два раза
9	Получить пары имен преподавателей, для которых выполняется следующее условие: оба преподавателя являются кураторами групп, при этом рейтинг группы первого преподавателя больше, чем в два раза, рейтинга группы второго преподавателя (имеются в виду группы, у которых данные преподавателя являются кураторами)
10	Получить пары имен преподавателей, для которых выполняется следующее условие: преподаватель_1 ведет группе на курс выше, курса группы, у которой ведет занятия преподаватель_2.
11	Получить пары номеров аудиторий, для которых выполняется следующее условие: - аудитория_1 находится выше аудитории_2 и имеет количество мест

	как минимум на 20 больше
12	Получить пары названий кафедр, для которых выполняется следующее условие: они относятся к разным факультетам, и фонд финансирования факультета кафедры_1 превышает как минимум на 10000, фонд финансирования факультета кафедры_2
13	Получить пары названий кафедр, для которых выполняется следующее условие: кафедра_1 находится на том же этаже и в том же корпусе, что и кафедра_2, но они относятся к разным факультетам с разницей в фонде финансирования более 10000
14	Получить пары названий кафедр, для которых выполняется следующее условие: декан факультета кафедры_1 имеет зарплату, превышающую зарплату декана факультета кафедры_2 более чем на 5000
15	Получить пары названий кафедр, для которых выполняется следующее условие: куратор группы кафедры_1 имеет зарплату, превышающую зарплату куратора группы кафедры_2 более, чем на 1000

Контрольные вопросы

1. Структура команды SELECT?
2. Приоритеты логических операций в части WHERE?
3. Правила использования выражений в SELECT-запросах?
4. Устранение дубликатов с SELECT-запросах
5. Формирование списка вывода команды SELECT.
6. Можно ли менять порядок конструкций в операторе SELECT?
7. С помощью какой конструкции выполняется сортировка результата запроса?
8. Какое ключевое слово используется для выполнения сортировки по убыванию?
9. Какие типы условий поиска (предикаты) вы знаете? Какие ключевые слова используются для их задания?
10. Назначение предиката IN и правила его использования?
11. Назначение предиката BETWEEN и правила его использования?
12. Назначение предиката LIKE и правила его использования?
13. Соединение таблиц.
14. Виды внешних соединений. Приведите примеры.
15. Опишите назначение предложений UNION, INTERSECT, EXCEPT

Литература

1. Астахова И. Ф. Распределенные базы данных на примере ORACLE: Методические материалы к спецкурсу. Воронеж: Изд-во ВГУ, 2000. 40 с.
2. Остринская Л. И., Семенова И. И., Дороболук Т. Б. Теория и практика работы с современными базами и банками данных: Учебное пособие. Омск: Изд-во СибАДИ, 2005. 250 с.

4. *Семенова И. И.* Сборник упражнений по стандарту SQL. Омск: Изд-во СибАДИ, 2005. 43 с.
5. Интернет-институт информационных технологий. www.intuit.ra. Курс «Основы SQL».
6. *Мамаев Е. В.* Microsoft SQL Server 2000. СПб.: БХВ-Петербург, 2005. 1280 с.
7. *Семенова И. И.* SQL стандарт в СУБД MS SQL SERVER, ORACLE, VFP И ACCESS: манипулирование данными. Омск: Изд-во СибАДИ, 2008. 57 с.

Описание структуры БД Sales

БД Sales используется в запросах из примеров теоретической части данного пособия и состоит из пяти таблиц:

- Customer – информация о клиентах
- City– справочник городов
- Product - информация о товарах
- Order – подробная информация о заказах
- OrdItem – информация о составе заказа (перечень товаров, входящих в заказ).

ТАБЛИЦА CUSTOMER

Имя столбца	Тип данных	Разрешить null	Описание
CustID	int, identity	NOT NULL	Идентификатор клиента (Первичный ключ)
FirstName	nvarchar(20)	NOT NULL	Имя
LastName	nvarchar(20)	NOT NULL	Фамилия
CityID	Int	NOT NULL	Идентификатор города, внешний ключ к таблице City
Address	nvarchar(50)	NOT NULL	Адрес
Zip	nchar(5)	NOT NULL	Индекс
Phone	Phone	NULL	Телефон

ТАБЛИЦА CITY

CityID	int, identity	NOT NULL	Идентификатор города (Первичный ключ)
CityName	nvarchar(20)	NOT NULL	Название

ТАБЛИЦА PRODUCT

ProdID	int, identity	NOT NULL	Идентификатор товара (первичный ключ)
Description	nvarchar(100)	NOT NULL	описание товара
Available	Int	NOT NULL	Количество на складе

ТАБЛИЦА ORDER

OrdID	int, identity	NOT NULL	Идентификатор заказа (первичный ключ)
CustID	int	NOT NULL	Идентификатор клиента
OrdDate	smalldatetime	NOT NULL	Дата и время размещения заказа

ТАБЛИЦА ORDITEM

OrdID	Int	NOT NULL	Идентификатор заказа (внешний ключ)
ProdID	int	NOT NULL	Идентификатор товара (внешний ключ)
Qty	int	NOT NULL	Количество в заказе
Price	money	NOT NULL	Цена

Описание структуры таблиц БД «Университет»

БД «Университет» используется в заданиях на самостоятельную работу из данного пособия.

ТАБЛИЦА FACULTY

FacID	int	NOT NULL	Первичный ключ таблицы FACULTY
Name	varchar(20)	NULL	Название факультета; уникальное, не может быть NULL
DeanID	int	NULL	Декан факультета. Внешний ключ на TeacherID в TEACHER.
Fund	int	NULL	Фонд финансирования факультета.
Building	Char(2)	NULL	Корпус, где расположен деканат факультета
UnivID	int	NULL	Идентификатор университета, к которому относится факультет.

ТАБЛИЦА DEPARTMENT

DepID	int	NOT NULL	Первичный ключ таблицы
Name	varchar(20)	NOT NULL	Название кафедры. Не может быть NULL
Fund	int	NULL	Фонд финансирования кафедры.

Building	Char(2)	NULL	Корпус, где расположена кафедра.
FacID	int	NULL	Идентификатор факультета.
HeadID	int	NULL	Идентификатор заведующего кафедры.
Ограничение целостности:		Пара столбцов (FacID, name) уникальна	

ТАБЛИЦА TEACHER

TeacherID	int	NOT NULL	Идентификатор преподавателя
DepID	int	NULL	Идентификатор кафедры
Name	varchar(20)	NOT NULL	Имя преподавателя. Не может быть NULL
Post	varchar(20)	NOT NULL	Должность преподавателя: ассистент, преподаватель, доцент, профессор
Tel	varchar(20)	NULL	Рабочий телефон
Hiredate	datetime	NOT NULL	Дата приема на работу.
Salary	int	NOT NULL	Ставка. Должна быть больше 1000, не может быть NULL
Comission	int	NULL	Надбавка к ставке. Значение по умолчанию равно 0.
ChiefID	int	NULL	Идентификатор непосредственного руководителя преподавателя.
City	varchar(20)	NULL	Город
Surname	varchar(20)	NULL	Фамилия
Ограничение целостности:		а) Commission должно быть по крайней мере в два раза меньше, чем Salary б) Salary + Commission должен находиться в диапазоне 1000 - 3000 в) ChiefID не должен быть равен TeacherID (то есть преподаватель не может быть руководителем самого себя)	

ТАБЛИЦА SGROUP

GroupID	int	NOT NULL	Идентификатор группы
Course	int	NULL	Курс
Num	int	NULL	Номер группы
Quantity	int	NULL	Количество студентов в группе.
Rating	int	NULL	Рейтинг группы. Значение по умолчанию равно 0.

DepID	int	NULL	Идентификатор кафедры группы.
CuratorID	int	NULL	Идентификатор куратора группы (, ссылка на TEACHER).
Ограничение целостности таблицы		Пара значений столбцов (DepID, Num) является UNIQUE (то есть на одной кафедре не могут быть группы с одинаковыми номерами) Пара значений столбцов внешних ключей (DepID, Curator) является уникальной. То есть один и тот же преподаватель не может быть куратором более, чем одной группы на одной кафедре)	

ТАБЛИЦА SUBJECT

SubjID	int	NOT NULL	Идентификатор дисциплины
Name	varchar(50)	NOT NULL	Название дисциплины.

ТАБЛИЦА ROOM

RoomID	int	NOT NULL	Идентификатор аудитории
Num	int	NOT NULL	Номер аудитории. Не может принимать значение NULL
Seats	int	NULL	Количество место в аудитории
Floor	char	NULL	Этаж аудитории
Building	int	NOT NULL	Корпус аудитории
Ограничение целостности таблицы		Пара значений столбцов (Num, Building) является уникальной (то есть в одном корпусе на могут быть аудитории с одинаковыми номерами)	

ТАБЛИЦА RASPISANIE

RaspID	int	NOT NULL	Идентификатор позиции в расписании
UchPlanID	int	NOT NULL	Ссылка на конкретную позицию учебного плана. Внешний ключ к таблице UCHPLAN
WeekDay	varchar(20)	NOT NULL	День недели. Принимает значения из списка: пон, втр, срд, чет, пят, суб, вск. Не может быть NULL
WeekNum	int	NOT NULL	Неделя. Принимает значения 1 или 2. Не может быть NULL
Lesson	int	NOT NULL	Занятие (пара)

RoomID	int	NULL	Идентификатор аудитории
Ограничение целостности таблицы	Столбцы (UchPlanID, Day, Week, Lesson) в совокупности являются UNIQUE (то есть у одной группы не может быть два различных занятия на одной и той же паре одного и того же дня недели и одной и той же недели)		

ТАБЛИЦА UNIVERS

UnivID	int	NOT NULL	Идентификатор университета
UnivName	varchar(20)	NULL	Название
Reiting	int	NULL	Рейтинг
City	varchar(20)	NULL	Город

ТАБЛИЦА UCHPLAN

UchPlanID	int	NOT NULL	Первичный ключ таблицы
Semestr	int	NULL	Номер семестра дисциплины
GroupID	int	NULL	Идентификатор группы
TeacherID	int	NULL	Идентификатор преподавателя
SubjID	int	NULL	Идентификатор дисциплины плана.
TypeID	int	NULL	Идентификатор типа занятия.
DepID	int	NULL	Идентификатор кафедры дисциплины.
Hours	int	NULL	Количество часов по плану
Ограничение целостности таблицы	Столбцы (Semestr, GroupID, SubjID, TypeID) в совокупности являются UNIQUE (т.е. дисциплины в группе в семестре дублироваться не должны)		

ТАБЛИЦА SUBJTYPE

SubjTypeID	int	NOT NULL	Первичный ключ таблицы
SubjType	varchar(20)	NULL	Тип занятия: лек, лб, прк, сем, ...

ТАБЛИЦА STUDENT

StudID	int	NOT NULL	Первичный ключ таблицы
Surname	varchar(20)	NULL	Фамилия студента
Name	int	NULL	Имя студента
Stipend	int	NULL	Стипендия (если нет = 0, иначе размер стипендии)
City	varchar(20)	NULL	Город
Birthday	datetime	NULL	Дата рождения

GroupID	int	NULL	Группа студента. Внешний ключ к таблице SGROUP
---------	-----	------	--

ТАБЛИЦА VEDOMOSTI

VedID	int	NOT NULL	Первичный ключ таблицы
ExamDate	datetime	NULL	Дата экзамена
UchPlanID	int	NULL	Внешний ключ к таблице UCHPLAN
VedomNumber	varchar(20)	NULL	Номер ведомости в формате '17/18-00001'
TryNumber	int	NULL	Номер попытки
Ограничение целостности таблицы	<p>Столбцы (UchPlanID, ExamDate) в совокупности являются UNIQUE (то есть у одной группы не может быть в один день больше одного экзамена)</p> <p>Столбцы (VedomNumber, TryNumber) в совокупности являются UNIQUE (то есть нельзя сформировать две ведомости с одинаковым номером и попыткой)</p>		

ТАБЛИЦА EXAM MARKS

VedID	int	NOT NULL	Первичный ключ таблицы Ведомость Внешний ключ к таблице VEDOMOSTI
StudID	int	NOT NULL	Первичный ключ таблицы Студент. Внешний ключ к таблице STUDENTS
Mark	int	NULL	Оценка

