

Подписано электронной подписью:  
Вержицкий Данил Григорьевич  
Должность: Директор КГПИ КемГУ  
Дата и время: 2025-04-23 00:00:00  
471086fad29a3b30e244c728abc3661ab35c9d50210dcf0e75e03a5b6fdf6436  
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Кемеровский государственный университет»  
Кузбасский гуманитарно-педагогический институт

Факультет информатики, математики и экономики  
Кафедра математики, физики и математического моделирования

Ю.С. Гаврилова

## **СОВРЕМЕННЫЕ ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ SQL**

### **Сложные запросы на языке SQL**

*Методические рекомендации по изучению дисциплины  
для обучающихся по направлениям подготовки  
01.03.02 Прикладная математика и информатика,  
02.03.03 Математическое обеспечение и администрирование информационных систем*

Новокузнецк

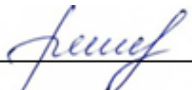
2022

УДК [378.147.88: 004.432.1](072)  
ББК 74.484(2Рос-4Кем)я73+32.973-018.6я73  
Г 12

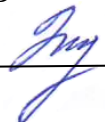
**Гаврилова Ю.С.**

Современные технологии программирования SQL. Сложные запросы на языке SQL: методические рекомендации по изучению дисциплины для студентов факультета информатики, математики и экономики, обучающихся по направлениям подготовки 01.03.02 Прикладная математика и информатика и 02.03.03 Математическое обеспечение и администрирование информационных систем / Ю.С. Гаврилова. – Новокузнецк : КГПИ ФГБОУ ВО «КемГУ», 2022. – 39 с.

Рекомендовано на заседании  
кафедры математики, физики и  
математического моделирования  
Протокол № 5 от 15.12.2022  
Заведующий каф. МФММ

 / Е.В. Решетникова

Утверждено методической комиссией  
факультета информатики, математики и  
экономики  
Протокол № 5 от 15.12.2022  
Председатель методической комиссии  
ФИМЭ

 / И.А. Жибинова

УДК [378.147.88: 004.432.1](072)  
ББК 74.484(2Рос-4Кем)я73+32.973-018.6я73  
Г 12

© Гаврилова Юлия Сергеевна  
© Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Кемеровский государственный университет»,  
Кузбасский гуманитарно-педагогический  
институт, 2022  
**текст представлен в авторской редакции**

## Содержание

ПРЕДИСЛОВИЕ.....	4
1 Теоретические сведения .....	5
1.1 Синтаксис языка запросов.....	5
1.2 Многотабличные запросы .....	11
1.3 Запросы с вложенными подзапросами.....	13
1.4 Запросы с коррелированными подзапросами .....	15
2 Лабораторные работы .....	18
2.1 Лабораторная работа «Построение многотабличных запросов» .....	18
2.1.1 Задания .....	18
2.1.2 Требования к отчету по лабораторной работе .....	24
2.1.3 Контрольные вопросы .....	24
2.2 Лабораторная работа «Построение запросов с вложенными подзапросами».....	25
2.1.1 Задания .....	25
2.1.2 Требования к отчету по лабораторной работе .....	29
2.1.3 Контрольные вопросы .....	30
3 Образец тестовых заданий.....	31
4 Рекомендуемая литература.....	38

## ПРЕДИСЛОВИЕ

Настоящие методические рекомендации адресованы студентам, получающим квалификацию бакалавр по направлениям подготовки: 01.03.02 Прикладная математика и информатика и 02.03.03 Математическое обеспечение и администрирование информационных систем и направлены на оказание помощи студентам в подготовке к выполнению лабораторных работ и индивидуальных заданий и прохождению тестирования по теме «Сложные запросы на языке SQL».

Язык SQL – один из базовых языков декларативного программирования, изучение которого необходимо для студентов вуза, поскольку именно с помощью данного языка можно создавать базы данных и их объекты: таблицы, запросы, представления, курсоры, хранимые процедуры и триггеры.

Данные методические материалы позволят студенту закрепить теоретический материал; подготовиться к лабораторным занятиям, прохождению тестирования и выполнению индивидуальных заданий по соответствующей теме.

Методические рекомендации могут оказаться полезными при выполнении проектных работ, прохождении учебных и производственных практик, написании курсовых и выпускных квалификационных работ.

## 1 Теоретические сведения

### 1.1 Синтаксис языка запросов

Для того чтобы приступить к созданию сложных запросов на языке SQL, необходимо изучить синтаксис и основные правила построения запросов.

Синтаксис запросов на выборку данных выглядит следующим образом:

**SELECT** <что выбираем? (название столбца)>

**FROM** <откуда выбираем? (название таблицы)>

**WHERE** <условие>.

Следует отметить, что условие (часть WHERE) не является обязательной частью запроса.

Если необходимо выбрать несколько столбцов, их названия перечисляются друг за другом через запятую.

*Пример 1:* На рисунке 1 представлена даталогическая модель данных. Необходимо получить название вида животного и его среднюю продолжительность жизни.

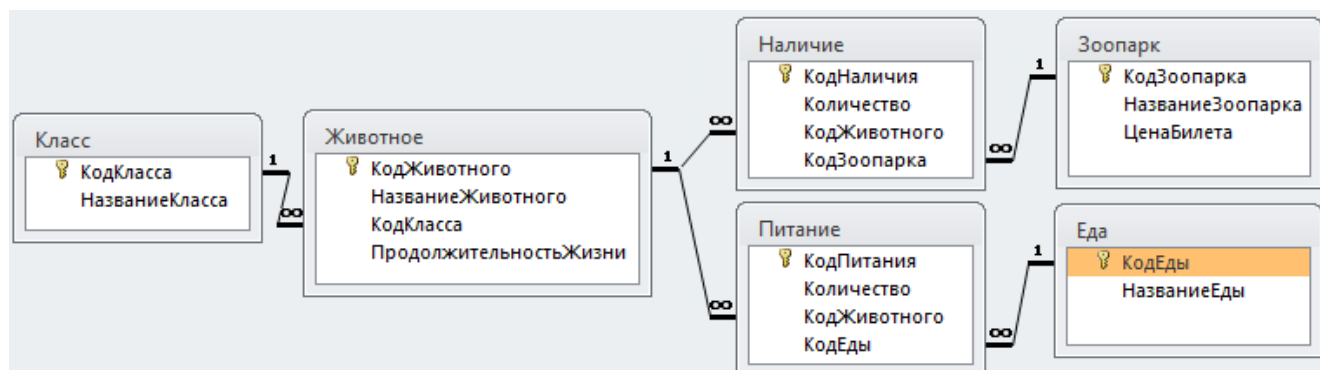


Рисунок 1 – Даталогическая модель данных «Зоопарк»

Так как в задании не поставлено условие отбора, запрос будет содержать только 2 фразы: SELECT и FROM. Тогда запрос на выборку данных будет выглядеть следующим образом:

**SELECT** НазваниеЖивотного, ПродолжительностьЖизни

**FROM** Животное

Если необходимо выбрать названия всех столбцов, можно заменить их перечисление символом \*.

Конструкции LIKE (похож на ...) и NOT LIKE (не похож на ...)

применяются для сравнения значения в ячейке с каким-либо текстом.

*Пример 2:* Необходимо получить список названий видов животных, начинающихся с буквы К. Тогда запрос на выборку данных будет выглядеть следующим образом:

```
SELECT НазваниеЖивотного  
FROM Животное  
WHERE НазваниеЖивотного LIKE 'К*'
```

Данный пример выполняется в СУБД Microsoft Office Access. Если нужно составить подобный запрос в СУБД PostgreSQL, то символ \* необходимо заменить на символ %.

*Пример 3:* Необходимо получить список названий видов животных, начинающихся с любых букв кроме буквы К. Тогда запрос на выборку данных будет выглядеть следующим образом:

```
SELECT НазваниеЖивотного  
FROM Животное  
WHERE НазваниеЖивотного NOT LIKE 'К*'
```

Ключевое слово IN применяется в том случае если определено множество, которому принадлежит значение. Если определено множество, которому не может принадлежать значение, то используется NOT IN.

*Пример 4:* Необходимо получить список животных, у которых продолжительность жизни составляет 20, 30, 35 или 40 лет.

```
SELECT НазваниеЖивотного  
FROM Животное  
WHERE ПродолжительностьЖизни IN (20,30,35,40)
```

В случае если в запросе необходимо учитывать несколько условий, их отделяют друг от друга ключевыми словами AND или OR.

В условии можно использовать операции сравнения (<,>,<=,>=) для работы с числами и числовыми полями.

*Пример 5:* Если необходимо получить названия животных, чья продолжительность жизни от 30 до 40 лет, то запрос на выборку данных можно

будет построить следующим образом:

```
SELECT  НазваниеЖивотного
FROM    Животное
WHERE   ПродолжительностьЖизни>=30 AND
          ПродолжительностьЖизни<=40
```

или

```
SELECT  НазваниеЖивотного
FROM    Животное
WHERE   ПродолжительностьЖизни BETWEEN 30 AND 40
```

Синтаксис запросов на выборку данных, предусматривающих сортировку и выбор N первых строк, выглядит следующим образом:

```
SELECT TOP N <что выбираем? (название столбца)>
FROM         <откуда выбираем? (название таблицы)>
WHERE        <условие>
ORDER BY    <название столбца> <порядок сортировки>
```

Для определения порядка сортировки необходимо указать ключевое слово:

**ASC** – сортировка по возрастанию;

**DESC** – сортировка по убыванию.

Если порядок сортировки в явном виде не указан, то будет применен заданный по умолчанию **ASC**. Вместо названия столбца во фразе **ORDER BY** можно указать его номер, например, **ORDER BY 1**. Такая конструкция в запросе говорит о том, что отобранные строки будут отсортированы по возрастанию значений первого столбца.

Конструкция **TOP N** применяется в Microsoft Office Access для выбора N первых значений, где N – любое целое число. Если вам необходимо сделать то же самое в PostgreSQL, необходимо воспользоваться конструкцией **LIMIT N**. Общий синтаксис запроса тогда выглядит следующим образом:

```
SELECT      <что выбираем? (название столбца)>
FROM       <откуда выбираем? (название таблицы)>
WHERE      <условие>
```

**ORDER BY** <название столбца> <порядок сортировки>

**LIMIT** N

*Пример 6:* Необходимо получить список десяти животных-долгожителей (10 животных с максимальной продолжительностью жизни).

Т.е. необходимо отсортировать список животных по столбцу «ПродолжительностьЖизни» с порядком сортировки «по убыванию» и выбрать 10 первых. Тогда запрос на выборку данных будет выглядеть следующим образом:

```
SELECT TOP 10   НазваниеЖивотного  
FROM           Животное  
ORDER BY       ПродолжительностьЖизни DESC
```

К столбцам, имеющим числовой тип данных, можно применять различные математические операции, причем использовать их как во фразе **WHERE**, так и во фразе **SELECT**.

Кроме того, в запросах можно использовать следующие математические функции:

- **ABS** – модуль числа;
- **Cos** – косинус;
- **Exp** – экспонента;
- **Log** – натуральный логарифм;
- **Rnd** – случайное число;
- **Round** – округление;
- **Sgn** – знак числа;
- **Sin** – синус;
- **Sqr** – квадратный корень;
- **Tan** – тангенс.

*Пример 7:* В зоопарке проходит акция «Билеты за треть от цены». В таком случае, если мы поделим на 3 некоторые числа, можем получить такую стоимость билетов: 33,(3). Воспользуемся функцией **round** для того чтобы округлить полученное значение: конструкция **Round(Стоимостьбилета/3,1)** округляет



значения столбца «Стоимостьбилета» до 1 знака после запятой. Запрос на выборку данных выглядит следующим образом:

```
SELECT  НазваниеЗоопарка, Round(Стоимостьбилета/3,1)
FROM    Зоопарк
```

В предыдущем примере мы рассмотрели результат выполнения запроса с вычисляемым полем, где заголовок таблицы в итоге имеет вид «Expr1001». Для дальнейшего использования в работе такой заголовок неудобен (если возникает необходимость к нему обращаться), поэтому рассмотрим синтаксис переименования столбца в запросе:

```
SELECT  <название столбца> AS <новое название>
FROM    <откуда выбираем? (название таблицы)>
```

*Пример 8:* Исправим название поля в запросе из примера 6 на название «Цена». Тогда запрос на выборку данных будет выглядеть следующим образом:

```
SELECT  НазваниеЗоопарка, Стоимостьбилета/2 AS Цена
FROM    Зоопарк
```

Функции для работы с датой в Microsoft Office Access:

- Date() – возвращает текущую дату;
- Year() – извлекает год из даты;
- Month() – извлекает месяц из даты;
- Day() – извлекает день из даты.

Данные функции можно использовать не только во фразе WHERE, но и во фразе SELECT.

*Пример 9:* Если необходимо получить список зоопарков, созданных в 2000 году, то запрос на выборку данных будет выглядеть следующим образом:

```
SELECT  НазваниеЗоопарка
FROM    Зоопарк
WHERE   Year(ДатаСоздания)=2000
```

В PostgreSQL для работы с датой применяется функция NOW(), которая выведет текущую дату вплоть до миллисекунд. Приписка ::DATE к команде NOW сузит ответ только до даты, а ::TIME – до времени.

Команда EXTRACT позволит получить нужный тип времени или даты, как из результата выполнения функции, так и из поля типа дата. Например, команда EXTRACT (MONTH FROM birthday) позволит выбрать только месяц из столбца с датой рождения.

Агрегатные функции предназначены для того, чтобы вычислять некоторое скалярное значение для заданного множества строк.

Чаще всего можно столкнуться с необходимостью использования следующих агрегатных функций:

- Count() – количество непустых строк в заданном диапазоне;
- Min() – минимальное значение;
- Max() – максимальное значение;
- Sum() – сумма;
- AVG() – среднее.

Выражения в функциях AVG и SUM должно представлять числовое значение (например, столбец, который хранит числовые значения). Выражение в функциях MIN, MAX и COUNT может представлять числовое значение, строковое значение или дату.

Все агрегатные функции за исключением COUNT(\*) игнорируют NULL (пустые ячейки).

*Пример 10:* Необходимо получить количество зоопарков, которые есть в системе. Тогда запрос на выборку данных будет выглядеть следующим образом:

```
SELECT Count(НазваниеЗоопарка)
FROM Зоопарк
```

В случае если нужно получить 2 столбца, один из которых является частью агрегатной функции, а другой – нет, необходимо группировать результаты по столбцу, не являющемуся частью агрегатной функции с помощью ключевого слова **GROUP BY**.

*Пример 11:* Необходимо получить столбец со средней продолжительностью жизни и количество видов животных, для которых она характерна. Тогда запрос на выборку данных будет выглядеть следующим образом:

**SELECT** ПродолжительностьЖизни, Count(НазваниеЖивотного)  
**FROM** Животное  
**GROUP BY** ПродолжительностьЖизни

Если нам нужно применить условие к результату агрегатной функции, то следует воспользоваться конструкцией **HAVING**.

*Пример 12:* Допустим, из результатов предыдущего запроса нам нужны только те значения продолжительности жизни, которым соответствуют 5 и более видов животных:

**SELECT** ПродолжительностьЖизни, Count(НазваниеЖивотного)  
**FROM** Животное  
**GROUP BY** ПродолжительностьЖизни  
**HAVING** Count(НазваниеЖивотного)>=5

## 1.2 Многотабличные запросы

Если необходимо получить данные из нескольких таблиц, то их можно соединить во фразе **FROM** запроса. Соединить таблицы можно следующим образом:

➤ **INNER JOIN** – внутреннее соединение, при котором результирующая таблица включает те строки, которые имеют соответствующие друг другу значения связанных полей в двух таблицах.

**SELECT** НазваниеЖивотного, НазваниеКласса  
**FROM** Животное **INNER JOIN** Класс  
**ON** Животное.КодКласса=Класс.КодКласса

➤ **LEFT JOIN** – левое внешнее соединение, при котором в результирующую таблицу попадут все строки из левой таблицы и только те строки из правой таблицы, для которой в левой таблице есть соответствующие значения связанных полей.

**SELECT** НазваниеЖивотного, НазваниеКласса  
**FROM** Животное **LEFT JOIN** Класс  
**ON** Животное.КодКласса=Класс.КодКласса

➤ **RIGHT JOIN** – правое внешнее соединение, при котором в результирующую таблицу попадут все строки из правой таблицы и только те строки из левой таблицы, для которой в правой таблице есть соответствующие значения связанных полей.

```
SELECT   НазваниеЖивотного, НазваниеКласса  
FROM     Животное RIGHT JOIN Класс  
          ON Животное.КодКласса=Класс.КодКласса
```

➤ Декартово произведение. Операция декартова произведения осуществляется путем сцепления каждой записи одной таблицы с каждой записью другой таблицы. Данную операцию используют вместе с выборкой по условию (селекцией), причем условие – это равенство соответствующих значений столбцов.

```
SELECT   НазваниеЖивотного, НазваниеКласса  
FROM     Животное, Класс  
WHERE    Животное.КодКласса=Класс.КодКласса
```

Использование синонимов в запросах позволяет обращаться к таблице, или соединению таблиц, а также сократить длину запроса. Синоним вводится в ключевой фразе **FROM** через пробел после названия таблицы, для которой он создается. Тогда во всех ссылках на названия таблиц, для которых введены синонимы, необходимо использовать синонимы (даже если ссылки расположены во фразе **SELECT**).

*Пример 13:* Выяснить, представители каких классов обитают в Московском зоопарке:

```
SELECT   НазваниеКласса  
FROM     (((Зоопарк З INNER JOIN Наличие Н  
          ON З.КодЗоопарка= Н.КодЗоопарка) INNER JOIN Животное Ж  
          ON Ж.КодЖивотного=Н.КодЖивотного) INNER JOIN Класс К  
          ON Ж.КодКласса=К.КодКласса)  
WHERE    НазваниеЗоопарка="Московский"
```

Результирующая таблица содержит повторяющиеся строки,

соответствующие разным животным, но принадлежащим одному и тому же классу. Для того чтобы исключить повторяющиеся строки из выборки, применяют ключевое слово DISTINCT.

### 1.3 Запросы с вложенными подзапросами

Запрос с вложенным подзапросом – такой запрос, внутри которого есть еще один или несколько запросов на выборку данных. Выполнение запроса с вложенным подзапросом происходит снизу вверх, т.е. сначала выполняется вложенный запрос, а затем – внешний. Подзапросы могут быть вложены во фразы SELECT, FROM и WHERE.

Подзапросы, вложенные во фразу WHERE

*Пример 14:* Получить список животных, продолжительность жизни которых в два раза меньше средней продолжительности жизни. Тогда запрос на выборку данных будет выглядеть следующим образом:

```
SELECT  НазваниеЖивотного
FROM    Животное
WHERE   ПродолжительностьЖизни=(
SELECT  AVG(ПродолжительностьЖизни)/2
FROM    Животное)
```

В качестве операторов сравнения могут использоваться =, >=, <=, <>. Если запрос построен с использованием одного из данных операторов сравнения, то необходимо, чтобы вложенный запрос возвращал ТОЛЬКО одно единственное значение (во фразе SELECT только одно поле и условие сформулировано таким образом, что ему удовлетворяет лишь одна строка). Например, агрегатная функция во вложенном подзапросе дает гарантию того, что будет возвращено только одно единственное значение. Если же заранее точно нельзя сказать, какое количество записей вернет вложенный подзапрос, необходимо использовать вместо операторов сравнения ключевое слово IN (NOT IN).

*Пример 15:* Получить список животных, продолжительность жизни которых совпадает с половиной продолжительности жизни зайца, медведя или дельфина.

Тогда запрос на выборку данных будет выглядеть следующим образом:

```
SELECT  НазваниеЖивотного
FROM    Животное
WHERE   ПродолжительностьЖизни IN (
SELECT   ПродолжительностьЖизни/2
FROM     Животное
WHERE    НазваниеЖивотного IN
          ('Заяц', 'Медведь', 'Дельфин'))
```

Подзапросы, вложенные во фразу FROM

Вложенные во фразу FROM подзапросы могут возвращать любое количество записей.

*Пример 16:* Получить список животных, которые относятся к классу «Млекопитающие».

Данный запрос можно реализовать, применив сначала операцию соединения таблиц «Животное» и «Класс», а затем – операцию выборки по условию НазваниеКласса='Млекопитающие'. Такая структура запроса не будет оптимальной в том случае, например, если у нас в зоопарках обитает 1000 видов животных, из которых млекопитающих всего 5. Гораздо лучше сначала выбрать из таблицы «Класс» одну строку, а потом соединять ее с таблицей «Животное»:

```
SELECT  НазваниеЖивотного
FROM    Животное INNER JOIN (SELECT * FROM Класс
WHERE   НазваниеКласса='Млекопитающие') К
ON      Животное.КодКласса=К.КодКласса
```

Подзапросы, вложенные во фразу SELECT

Вложенные во фразу SELECT подзапросы также могут возвращать любое количество записей.

*Пример 17:* Получить список зоопарков и долю, которую количество животных в данном зоопарке занимает от суммарного количества всех животных.

```
SELECT   НазваниеЗоопарка, SUM(Количество)/
          (SELECT SUM(Количество) FROM Наличие) AS Доля
```

```

FROM      Зоопарк INNER JOIN Наличие
ON        Зоопарк.КодЗоопарка=Наличие.КодЗоопарка
GROUP BY  НазваниеЗоопарка

```

### 1.4 Запросы с коррелированными подзапросами

Запрос с коррелированным подзапросом – такой запрос, внутри которого есть еще один запрос на выборку данных, который зависит от внешнего запроса.

Выполнение запроса с коррелированным подзапросом происходит сверху вниз – коррелированный подзапрос выполняется для каждой строки внешнего запроса, а результат выполнения коррелированного подзапроса попадает в условие внешнего запроса.

На рисунке 2 представлена даталогическая модель данных для предметной области «Университет».

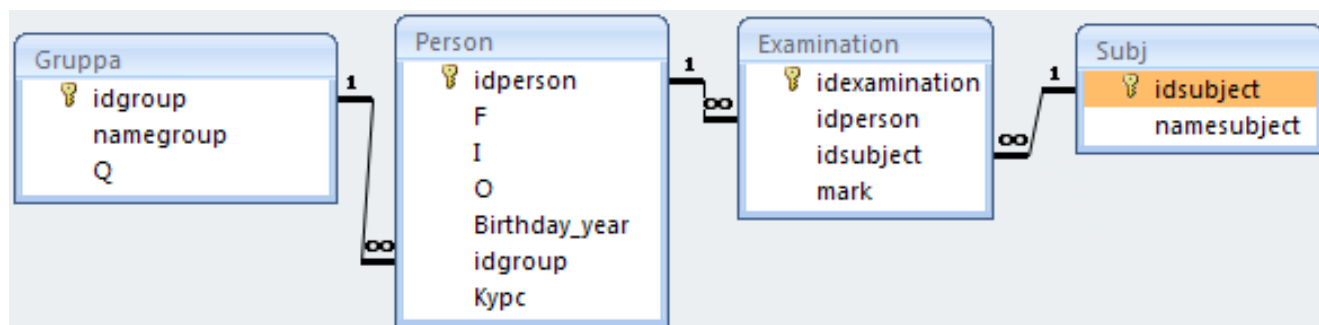


Рисунок 2 – Даталогическая модель данных «Университет»

*Пример 18:* Получить список студентов, у которых есть однофамилец.

Однофамилец – это человек с такой же фамилией, но с другим кодом человека. Внешний запрос в таком случае должен получать список студентов. Затем для каждой строки внешнего запроса выполняется коррелированный. Он ищет людей с теми же фамилиями, которые есть в списке, а условие, указанное во фразе **WHERE** позволяет исключить из выборки самих этих людей.

```

SELECT F,I,O
FROM   Person P
WHERE  F=(SELECT F
           FROM Person P1
           WHERE P.idperson<>P1.idperson)

```

*Пример 19:* Получить список студентов, у которых средний балл за экзамены больше среднего балла по их группе.

```
SELECT F,I,O
FROM (SELECT F,I,O,P.idgroup, AVG(mark) AS Среднее
FROM (Person P INNER JOIN Examination E
ON P.idperson=E.idperson) INNER JOIN Gruppya G
ON P.idgroup=G.idgroup
GROUP BY F,I,O,P.idgroup) AS Svodnoe
WHERE Среднее>(SELECT AVG(mark)
FROM Person P1 INNER JOIN Examination E1
ON P1.idperson=E1.idperson
WHERE Svodnoe.idgroup=P1.idgroup)
```

Первый подзапрос – вложенный, он позволяет найти среднюю оценку каждого студента и код группы, в которой он учится.

Второй подзапрос – коррелированный, который берет каждую строку из внешнего запроса (список студентов) и для каждой строки ищет среднюю оценку его группы.

*Пример 20:* Получить список студентов, которые сдали больше одного экзамена.

```
SELECT DISTINCT F,I,O
FROM Person P INNER JOIN Examination E ON P.idperson=E.idperson
WHERE P.idperson IN (
SELECT P1.idperson
FROM Person P1 INNER JOIN Examination E1
ON P1.idperson=E1.idperson
WHERE E.idsubject<>E1.idsubject)
```

Если студент сдал больше одного экзамена, то должно быть как минимум 2 записи в соединении таблиц Person и Examination с одинаковым кодом студента (idperson), но с разными кодами предмета (idsubject).

*Пример 21:* Получить список студентов, которые сдали как минимум 2



экзамена на одинаковые оценки.

```
SELECT DISTINCT F,I,O  
FROM Person P INNER JOIN Examination E ON P.idperson=E.idperson  
WHERE P.idperson IN(  
    SELECT P1.idperson  
    FROM Person P1 INNER JOIN Examination E1  
    ON P1.idperson=E1.idperson  
    WHERE P.idsubject<>P1.idsubject AND E.mark=E1.mark)
```

## 2 Лабораторные работы

### 2.1 Лабораторная работа «Построение многотабличных запросов»

#### 2.1.1 Задания

Лабораторная работа состоит из двух заданий. Первое задание выполняется в учебной базе данных «Зоопарк» (СУБД Microsoft Office Access), второе задание выполняется в базе данных, разработанной для курсовой работы.

**Задание 1.** В соответствии с вариантом составить на языке SQL многотабличные запросы на выборку данных, результат выполнения которых соответствует описанию (описания по вариантам приведены в таблице 1).

**Задание 2.** Составить 10 произвольных многотабличных запросов на выборку данных (к своей базе данных).

Таблица 1 – Индивидуальные задания

№ варианта	Задание
1.	<ol style="list-style-type: none"><li>1. Животное, которое ест больше всех в Новокузнецком зоопарке.</li><li>2. Список классов животных, которые есть в Московском зоопарке.</li><li>3. Минимальная стоимость билета, за которую можно посмотреть на жирафа, а также зоопарк, где данная стоимость билета установлена.</li><li>4. Количество животных, принадлежащих к классу «Земноводные» и обитающих в Томском зоопарке.</li><li>5. Еда, которую класс «Пресмыкающиеся» в среднем съедает в сутки.</li><li>6. Количество мяса, которое необходимо закупить Новосибирскому зоопарку на неделю.</li><li>7. Зоопарк, в котором живет больше всего животных.</li><li>8. Количество зоопарков, где обитают животные, чье название начинается с буквы К.</li></ol>

	<p>9. Животное с минимальной средней продолжительностью жизни среди обитателей Томского зоопарка.</p> <p>10. Названия животных, обитающих в Новосибирском зоопарке и название класса, к которому каждое из них относится.</p>
2.	<p>1. Животное, которое ест больше всех.</p> <p>2. Средняя продолжительность жизни животных Московского зоопарка.</p> <p>3. Название зоопарка, в котором есть утки.</p> <p>4. Максимальная стоимость билета, за которую можно посмотреть на жирафа, а также зоопарк, где данная стоимость билета установлена.</p> <p>5. Количество животных, принадлежащих к классу «Пресмыкающиеся» и обитающих в Московском зоопарке.</p> <p>6. Количество еды, которое класс «Земноводные» в среднем съедает в сутки.</p> <p>7. Количество фруктов и овощей, которое необходимо закупить Новосибирскому зоопарку на месяц.</p> <p>8. Зоопарк, в котором обитает больше всего видов животных.</p> <p>9. Количество особей, употребляющих в пищу мясо.</p> <p>10. Названия животных, которые за неделю съедают больше 8 кг еды среди обитателей Новосибирского зоопарка.</p>
3.	<p>1. Животное, которое обладает самой большой средней продолжительностью жизни среди обитателей Московского зоопарка.</p> <p>2. Средняя продолжительность жизни животных, питающихся злаками.</p> <p>3. Список классов животных, которых не кормят мясом.</p> <p>4. Еда, которой питаются животные, живущие в среднем 30 лет.</p> <p>5. Зоопарк, в котором обитает больше всего видов животных.</p> <p>6. Животные, которых в Томском зоопарке больше 5 особей.</p>

	<p>7. Название еды, которая пользуется наибольшей популярностью у обитателей Новосибирского зоопарка.</p> <p>8. Зоопарки, в которых обитает больше 200 особей животных.</p> <p>9. Зоопарки, в которых за неделю съедают больше 40кг мяса.</p> <p>10. Зоопарк, в котором живет больше всего земноводных.</p>
4.	<p>1. Список животных, обитающих в Томском зоопарке.</p> <p>2. Еда животных, у которых в названии есть буква Л.</p> <p>3. Средняя стоимость билета (по всем зоопаркам), по которому можно посмотреть на носорога.</p> <p>4. Список животных, которых можно увидеть, приобретя билет за максимальную стоимость (по всем зоопаркам).</p> <p>5. Еда, которую ест меньше всего животных.</p> <p>6. Первые 5 долгожителей среди обитателей Московского зоопарка.</p> <p>7. Зоопарки, в которых живут животные с буквой Л в названии.</p> <p>8. Зоопарки, в которых количество белых медведей составляет от 5 до 10.</p> <p>9. Зоопарк, обитатели которого за сутки съедают больше всего еды (в килограммах).</p> <p>10. Список животных, которых кормят фруктами.</p>
5.	<p>1. Животное, которое ест меньше всех в Новокузнецком зоопарке.</p> <p>2. Список классов животных, которые есть в Томском зоопарке.</p> <p>3. Максимальная стоимость билета, за которую можно посмотреть на скунса, а также зоопарк, где данная стоимость билета установлена.</p> <p>4. Количество животных, принадлежащих к классу «Птицы» и обитающих в Томском зоопарке.</p> <p>5. Количество граммов еды, которое класс «Паукообразные» в среднем съедает в сутки.</p> <p>6. Количество овощей, которое необходимо закупить</p>

	<p>Новосибирскому зоопарку на неделю.</p> <ol style="list-style-type: none"> <li>7. Зоопарк, в котором живет меньше всего животных.</li> <li>8. Количество зоопарков, где обитают животные, чье название начинается с буквы А.</li> <li>9. Животное с максимальной средней продолжительностью жизни среди обитателей Томского зоопарка.</li> <li>10. Названия животных, обитающих в Томском зоопарке и название класса, к которому они относятся.</li> </ol>
6.	<ol style="list-style-type: none"> <li>1. Животное, которое ест меньше всех.</li> <li>2. Средняя продолжительность жизни животных Новосибирского зоопарка.</li> <li>3. Название зоопарка, в котором уток кормят фруктами.</li> <li>4. Максимальная стоимость билета, за которую можно посмотреть на слона, а также зоопарк, где данная стоимость билета установлена.</li> <li>5. Количество животных, принадлежащих к классу «Млекопитающие» и обитающих в Московском зоопарке.</li> <li>6. Количество граммов еды, которое класс «Птицы» в среднем съедает в сутки.</li> <li>7. Количество граммов отрубей, которое необходимо закупить Новосибирскому зоопарку на месяц.</li> <li>8. Зоопарк, в котором обитает меньше всего видов животных.</li> <li>9. Количество особей, употребляющих в пищу мясо.</li> <li>10. Названия животных, которые за неделю съедают меньше 8 кг еды среди обитателей Новосибирского зоопарка.</li> </ol>
7.	<ol style="list-style-type: none"> <li>1. Животное, которое обладает самой большой средней продолжительностью жизни среди обитателей Томского зоопарка.</li> <li>2. Средняя продолжительность жизни животных, питающихся злаками.</li> <li>3. Список классов животных, которых не кормят злаками.</li> </ol>

	<p>4. Еда, которой питаются животные, живущие в среднем от 15 до 20 лет.</p> <p>5. Зоопарк, в котором обитает меньше всего видов животных.</p> <p>6. Животные, которых в Томском зоопарке больше 15 особей.</p> <p>7. Название еды, которая пользуется наибольшей популярностью у обитателей Новосибирского зоопарка.</p> <p>8. Зоопарки, в которых обитает от 100 до 300 особей животных.</p> <p>9. Зоопарки, в которых есть животные, за неделю съедающие больше 40кг мяса.</p> <p>10. Зоопарк, в котором живет больше всего птиц (среди всех зоопарков).</p>
8.	<p>1. Список животных, обитающих в Московском зоопарке.</p> <p>2. Еда животных, у которых в названии есть буква У.</p> <p>3. Максимальная стоимость билета, по которому можно посмотреть на носорога.</p> <p>4. Список животных, которых можно увидеть, приобретя билет за максимальную стоимость.</p> <p>5. Еда, которую ест меньше всего животных.</p> <p>6. Первые 5 долгожителей среди обитателей Новосибирского зоопарка.</p> <p>7. Зоопарки, в которых живут животные с буквой О в названии.</p> <p>8. Зоопарки, в которых количество белых медведей составляет от 5 до 10.</p> <p>9. Зоопарк, обитатели которого за сутки съедают больше всего еды (в килограммах).</p> <p>10. Список животных, которых кормят фруктами.</p>
9.	<p>1. Животное, которое живет в среднем меньше всех в Новокузнецком зоопарке.</p> <p>2. Список еды, которую едят в Томском зоопарке.</p> <p>3. Максимальная стоимость билета, за которую можно</p>

	<p>посмотреть на пуму, а также зоопарк, где данная стоимость билета установлена.</p> <ol style="list-style-type: none"> <li>4. Количество животных, принадлежащих к классу «Ракообразные» и обитающих в Томском зоопарке.</li> <li>5. Количество граммов еды, которое класс «Ракообразные» в среднем съедает в сутки.</li> <li>6. Количество отрубей, которое необходимо закупить Московскому зоопарку на неделю.</li> <li>7. Зоопарк, в котором живет меньше всего видов животных.</li> <li>8. Количество зоопарков, где обитают животные, чье название начинается с буквы Ш.</li> <li>9. Животное с максимальной средней продолжительностью жизни среди обитателей Томского зоопарка.</li> <li>10. Названия животных, обитающих в Томском зоопарке и названия классов, к которым они относятся.</li> </ol>
10.	<ol style="list-style-type: none"> <li>1. Животное, которое ест больше всех злаков.</li> <li>2. Средняя продолжительность жизни животных Новосибирского зоопарка, питающихся мясом.</li> <li>3. Название зоопарка, в котором шиншиллы кормят фруктами.</li> <li>4. Максимальная стоимость билета, за которую можно посмотреть на питона, а также зоопарк, где данная стоимость билета установлена.</li> <li>5. Количество животных, принадлежащих к классу «Земноводные» и обитающих в Московском зоопарке.</li> <li>6. Количество граммов еды, которое класс «Млекопитающие» в среднем съедает в сутки в Московском зоопарке.</li> <li>7. Количество граммов фруктов, которое необходимо закупить Новосибирскому зоопарку на месяц.</li> <li>8. Зоопарк, в котором обитает меньше всего видов животных.</li> <li>9. Количество особей, употребляющих в пищу мясо в Томском</li> </ol>

зоопарке.

10. Названия животных, которые за неделю съедают от 3 до 5 кг фруктов среди обитателей Новосибирского зоопарка.

### **2.1.2 Требования к отчету по лабораторной работе**

В отчете по лабораторной работе «Построение многотабличных запросов» для каждого запроса необходимо представить:

1. текстовое описание ожидаемого результата выполнения запроса (из таблицы 1 для задания 1, для задания 2 – составленное самостоятельно);
2. текст запроса на языке SQL;
3. результат выполнения запроса (снимок окна СУБД с таблицей–результатом, выведенным на экран после выполнения запроса).

Отчет по лабораторной работе оформляется в соответствии с требованиями, принятыми в КГПИ ФГБОУ ВО «КемГУ», предоставляется преподавателю в электронном виде.

### **2.1.3 Контрольные вопросы**

1. Назовите способы соединения таблиц в многотабличном запросе.
2. Запишите синтаксис для каждого способа соединения таблиц в многотабличном запросе.
3. В каких случаях для соединения таблиц рекомендуется применять RIGHT JOIN, LEFT JOIN? Приведите примеры.
4. Какое максимальное количество таблиц можно объединить с помощью операции «декартово произведение»?
5. Опишите процесс выполнения операции «декартово произведение». Почему декартово произведение применяют только вместе с операцией выборки по условию?
6. Если в одной таблице 100 строк, а в другой – 50, то сколько строк получим в результате выполнения над этими таблицами операции декартова произведения?



## 2.2 Лабораторная работа «Построение запросов с вложенными подзапросами»

### 2.1.1 Задания

Лабораторная работа состоит из двух заданий. Первое задание выполняется в учебной базе данных «Студенты» (СУБД Microsoft Office Access), второе задание выполняется в базе данных, разработанной для курсовой работы.

**Задание 1.** В соответствии с вариантом составить на языке SQL запросы с вложенными подзапросами, результат выполнения которых соответствует описанию (описания по вариантам приведены в таблице 2).

**Задание 2.** Составить 10 произвольных запросов с вложенными подзапросами на выборку данных (к своей базе данных).

Таблица 2 – Индивидуальные задания

№ варианта	Задание
1.	<ol style="list-style-type: none"><li>1. Список студентов, которые за экзамен по дисциплине «Современные технологии программирования SQL» получили оценку выше средней по тому же предмету.</li><li>2. Список студентов, которые сдали экзамен по дисциплине «Современные технологии программирования SQL» в один день с Ивановым Сергеем Петровичем.</li><li>3. Список студентов, сдававших дисциплину с самой высокой средней оценкой.</li><li>4. Список дисциплин, экзамены по которым сдало больше половины всех учащихся.</li><li>5. Номер курса, на котором больше всего академических групп.</li><li>6. Номер курса, на котором сдают больше всего предметов.</li><li>7. Получить список однокурсников Иванова Сергея Петровича.</li></ol>
2.	<ol style="list-style-type: none"><li>1. Список студентов, которые за экзамен по дисциплине «Современные технологии программирования SQL» получили</li></ol>

	<p>оценку выше средней оценки по всем дисциплинам за этот год.</p> <ol style="list-style-type: none"> <li>2. Список студентов, которые учатся в группе с самой высокой средней оценкой.</li> <li>3. Список студентов, которые учатся в одной группе с Ивановым Сергеем Петровичем.</li> <li>4. Список дисциплин, экзамен по которым сдавали все учащиеся.</li> <li>5. Номер курса, на котором больше всего академических групп.</li> <li>6. Номер курса, на котором сдают меньше всего предметов.</li> <li>7. Список студентов, у которых есть однофамильцы.</li> </ol>
3.	<ol style="list-style-type: none"> <li>1. Список студентов, которые за экзамен по дисциплине «Современные технологии программирования SQL» получили оценку ниже средней по тому же предмету.</li> <li>2. Список студентов, которые получили оценку за экзамен по дисциплине «Современные технологии программирования SQL» выше средней по своей группе.</li> <li>3. Список студентов, родившихся в один год с Ивановым Сергеем Петровичем.</li> <li>4. Список дисциплин, экзамены по которым сдало больше половины всех учащихся.</li> <li>5. Номер курса, на котором меньше всего академических групп.</li> <li>6. Номер курса, на котором учится больше всего студентов, родившихся в 2003 году.</li> <li>7. Список студентов, у которых есть тезки.</li> </ol>
4.	<ol style="list-style-type: none"> <li>1. Список студентов, которые сдали экзамен по дисциплине «Современные технологии программирования SQL» в один день с Петровым Иваном Васильевичем.</li> <li>2. Список студентов, которые учатся в самой многочисленной группе.</li> <li>3. Список студентов, которые получили оценку за экзамен по дисциплине «Современные технологии программирования SQL»</li> </ol>

	<p>выше средней по своей группе.</p> <ol style="list-style-type: none"> <li>4. Список дисциплин, экзамен по которым сдавали все учащиеся.</li> <li>5. Номер курса, на котором меньше всего академических групп.</li> <li>6. Номер курса, на котором учится меньше всего студентов, родившихся в 2003 году.</li> <li>7. Самого младшего студента.</li> </ol>
5.	<ol style="list-style-type: none"> <li>1. Список студентов, которые учатся в самой малочисленной группе.</li> <li>2. Список студентов, родившихся в один год с Петровым Иваном Васильевичем.</li> <li>3. Список студентов, которые учатся в одной группе со студентом, имеющим самую высокую среднюю оценку.</li> <li>4. Список дисциплин, экзамены по которым сдало меньше половины всех учащихся.</li> <li>5. Группу, где больше всего студентов, родившихся в 2003 году.</li> <li>6. Список студентов, которые учатся на одном курсе с Петровым Иваном Васильевичем.</li> <li>7. Получить список однокурсников Иванова Сергея Петровича.</li> </ol>
6.	<ol style="list-style-type: none"> <li>1. Список студентов, которые учатся в группе с самой высокой средней оценкой.</li> <li>2. Список студентов, которые сдавали один набор экзаменов с Петровым Иваном Васильевичем.</li> <li>3. Список студентов, которые за экзамен по дисциплине «Современные технологии программирования SQL» получили оценку выше средней по тому же предмету.</li> <li>4. Список дисциплин, экзамены по которым сдало меньше половины всех учащихся.</li> <li>5. Группы, где нет студентов, родившихся в 2003 году.</li> <li>6. Список студентов, которые учатся на одном курсе с Петровым Иваном Васильевичем.</li> </ol>

	7. Список студентов, у которых есть тезки.
7.	<ol style="list-style-type: none"> <li>1. Список студентов, которые учатся в самой многочисленной группе.</li> <li>2. Список студентов, сдававших дисциплину с самой низкой средней оценкой.</li> <li>3. Список студентов, которые сдавали один набор экзаменов с Ивановым Сергеем Петровичем.</li> <li>4. Номер курса, на котором учится больше всего студентов.</li> <li>5. Группы, где больше всего студентов, родившихся в 2003 году.</li> <li>6. Список студентов, которые по дисциплине «Современные технологии программирования SQL» получили не такую оценку, как Иванов Сергей Петрович.</li> <li>7. Список студентов, у которых есть однофамильцы.</li> </ol>
8.	<ol style="list-style-type: none"> <li>1. Список студентов, которые учатся в самой малочисленной группе.</li> <li>2. Список студентов, которые учатся в одной группе со студентом, имеющим самую низкую среднюю оценку.</li> <li>3. Список студентов, которые за экзамен по дисциплине «Современные технологии программирования SQL» получили оценку выше средней оценки по всем дисциплинам за этот год.</li> <li>4. Номер курса, на котором учится меньше всего студентов.</li> <li>5. Группу, где нет студентов, родившихся в 2003 году.</li> <li>6. Список студентов, которые по дисциплине «Современные технологии программирования SQL» получили не такую оценку, как Иванов Сергей Петрович.</li> <li>7. Самого старшего студента.</li> </ol>
9.	<ol style="list-style-type: none"> <li>1. Список студентов, которые за экзамен по дисциплине «Современные технологии программирования SQL» получили оценку ниже средней оценки по всем дисциплинам за этот год.</li> <li>2. Список студентов, которые учатся на самом многочисленном</li> </ol>

	<p>курсе.</p> <ol style="list-style-type: none"> <li>3. Список студентов, которые за экзамен по дисциплине «Современные технологии программирования SQL» получили оценку выше средней оценки по всем дисциплинам за этот год.</li> <li>4. Номер курса, на котором учится больше всего студентов.</li> <li>5. Группу, где больше всего студентов, родившихся в 2003 году.</li> <li>6. Номер курса, на котором сдают больше всего предметов.</li> <li>7. Самого младшего студента.</li> </ol>
10.	<ol style="list-style-type: none"> <li>1. Список студентов, у которых средняя оценка больше, чем средняя оценка всех студентов по дисциплине «Современные технологии программирования SQL», но меньше, чем по дисциплине «Дискретная математика».</li> <li>2. Список студентов, которые учатся в одной группе с Петровым Иваном Васильевичем.</li> <li>3. Список студентов, которые учатся на самом малочисленном курсе.</li> <li>4. Номер курса, на котором учится меньше всего студентов.</li> <li>5. Группу, где меньше всего студентов, родившихся в 2003 году.</li> <li>6. Номер курса, на котором сдают меньше всего предметов.</li> <li>7. Самого старшего студента.</li> </ol>

### 2.1.2 Требования к отчету по лабораторной работе

В отчете по лабораторной работе «Построение запросов с вложенными подзапросами» для каждого запроса необходимо представить:

1. текстовое описание ожидаемого результата выполнения запроса (из таблицы 1 для задания 1, для задания 2 – составленное самостоятельно);
2. текст запроса на языке SQL;
3. результат выполнения запроса (снимок окна СУБД с таблицей–результатом, выведенным на экран после выполнения запроса).

Отчет по лабораторной работе оформляется в соответствии с требованиями, принятыми в КГПИ ФГБОУ ВО «КемГУ», предоставляется преподавателю в электронном виде.

### **2.1.3 Контрольные вопросы**

1. Назовите части запроса на выборку данных, в которые можно вставить вложенный подзапрос.
2. Приведите примеры вложенности подзапросов во фразу SELECT.
3. Приведите примеры вложенности подзапросов во фразу FROM.
4. Приведите примеры вложенности подзапросов во фразу WHERE.
5. В чем различие между вложенным и коррелированным подзапросом?
6. Опишите процесс выполнения запроса с вложенным подзапросом. Что выполняется раньше: запрос или подзапрос?
7. Опишите процесс выполнения запроса с коррелированным подзапросом. Что выполняется раньше: запрос или подзапрос?

### 3 Образец тестовых заданий

1. SQL-запрос выглядит следующим образом:

```
SELECT TOP 1    Фамилия, Имя
FROM            Сотрудник
WHERE           Month(ДатаРождения)='4'
ORDER BY       1
```

В таблице Сотрудник 2 записи:

Сотрудник							
	КодСотрудн	Фамилия	Имя	Отчество	ДатаРожде	КодДолжнс	КодОтдела
+	1	Иванов	Алексей	Петрович	05.04.1882	1	1
+	2	Сорокина	Людмила	Геннадьевна	18.04.1990	1	1

В результате выполнения запроса:

- Результирующая таблица будет содержать сведения про Сорокину.
- Появится диалоговое окно для уточнения параметров запроса.
- Результирующая таблица будет содержать сведения про Иванова.
- Произойдет ошибка.

2. SQL-запрос выглядит следующим образом:

```
SELECT DISTINCT (Имя)
FROM            Сотрудник
```

В таблице Сотрудник 2 записи:

Сотрудник							
	КодСотрудн	Фамилия	Имя	Отчество	ДатаРожде	КодДолжнс	КодОтдела
+	1	Иванов	Алексей	Петрович	05.04.1882	1	1
+	2	Сорокина	Людмила	Геннадьевна	18.04.1990	1	1

В результате выполнения запроса:

- Результирующая таблица будет содержать неповторяющиеся имена.
- Результирующая таблица будет содержать повторяющиеся имена.
- Произойдет ошибка.
- Появится диалоговое окно для уточнения параметров запроса.

3. SQL-запрос выглядит следующим образом:

**SELECT**      Фамилия, Имя  
**FROM**        Сотрудник  
**WHERE**        Имя **LIKE** 'Алексей' **AND** Имя **LIKE** 'Наталья'

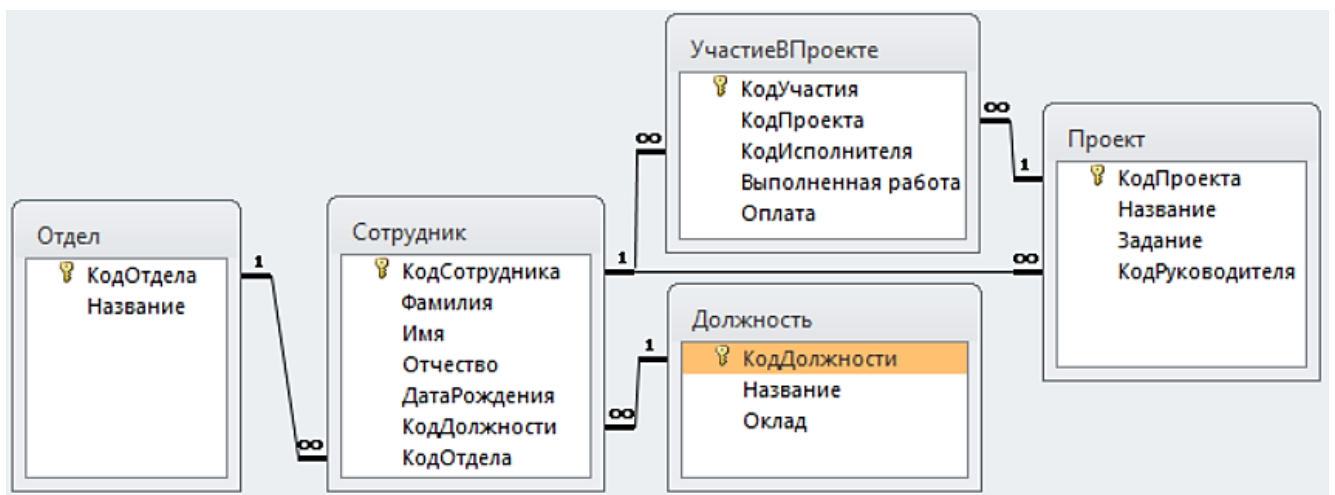
В таблице Сотрудник 2 записи:

Сотрудник							
	КодСотрудн	Фамилия	Имя	Отчество	ДатаРожде	КодДолжнс	КодОтдела
+	1	Иванов	Алексей	Петрович	05.04.1882	1	1
+	2	Сорокина	Людмила	Геннадьевна	18.04.1990	1	1

В результате выполнения запроса:

- Появится диалоговое окно для уточнения параметров запроса.
- Произойдет ошибка.
- Нет верного ответа.
- Результирующая таблица будет содержать строки, соответствующие людям с именами Алексей и Наталья.

4. Дана схема базы данных.



SQL-запрос выглядит следующим образом:

**SELECT**      Фамилия, Имя, Отдел.Название, Должность.Название  
**FROM**        Сотрудник С, Должность Д, Отдел О  
**WHERE**        О.КодОтдела=С.КодОтдела  
                  **AND** С.КодДолжности=Д.КодДолжности.

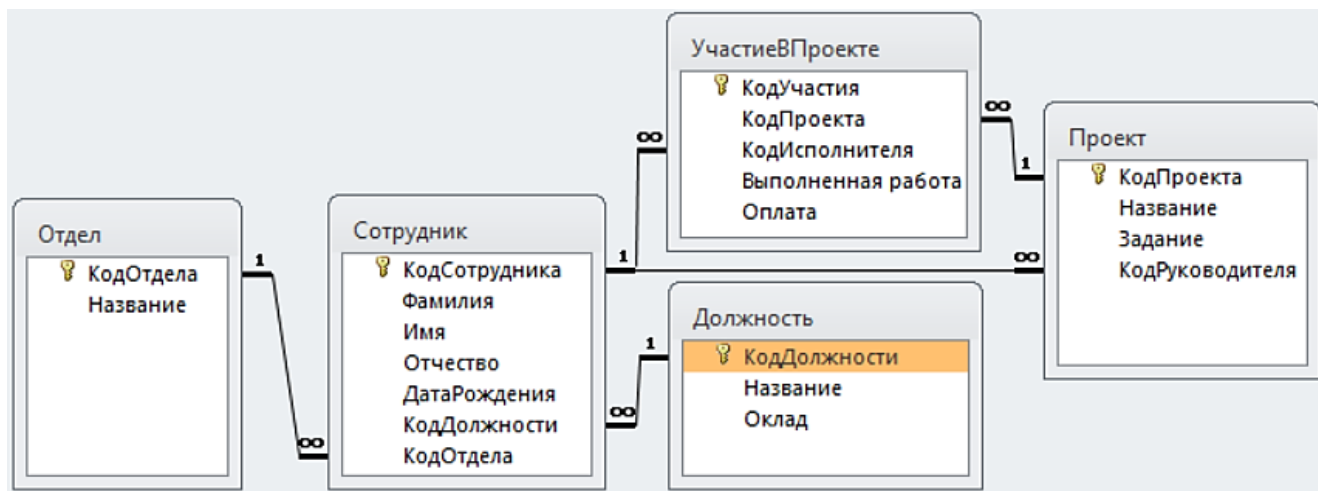
В результате выполнения запроса:

- Результирующая таблица будет пустой.



- b. Произойдет ошибка.
- c. Результирующая таблица будет содержать неверные данные.
- d. Появится диалоговое окно для уточнения параметров запроса.

5. Дана схема базы данных.



SQL-запрос выглядит следующим образом:

```

SELECT    Фамилия, Имя, Название
FROM      Сотрудник, Отдел
    
```

В результате выполнения запроса:

- a. Результирующая таблица будет содержать неверные данные.
- b. Произойдет ошибка.
- c. Появится диалоговое окно для уточнения параметров запроса.
- d. Результирующая таблица будет пустой.

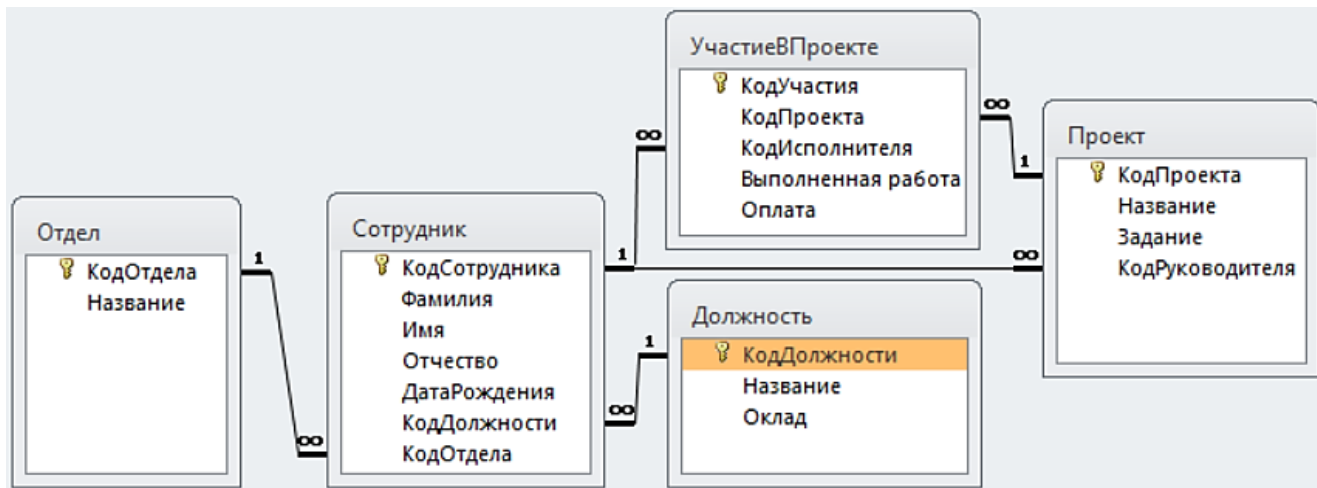
6. В какую часть запроса на выборку данных можно вложить подзапрос?

- a. WHERE.
- b. FROM.
- c. SELECT.
- d. HAVING.

7. Дана схема базы данных:

Необходимо посчитать, сколько человек работает в каждом отделе.

Правильный SQL-запрос будет выглядеть следующим образом:

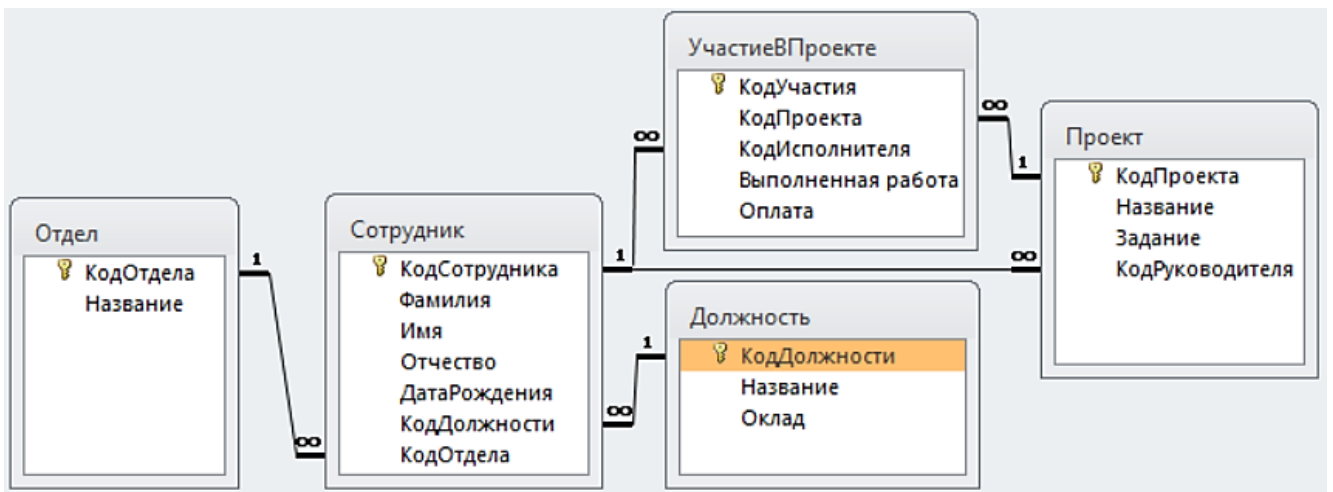


- a. **SELECT**            Название, Count(Фамилия, Имя)  
**FROM**                Сотрудник, Отдел  
**WHERE**                Отдел.КодОтдела=Сотрудник.КодСотрудника  
**GROUP BY**            Название
- b. **SELECT**            Название, Count(Фамилия, Имя)  
**FROM**                Сотрудник **LEFT JOIN** Отдел  
**ON** Отдел.КодОтдела=Сотрудник.КодСотрудника  
**GROUP BY**            Название
- c. Нет верного ответа
- d. **SELECT**            Название, Count(Фамилия)  
**FROM**                Сотрудник **LEFT JOIN** Отдел  
**WHERE**                Отдел.КодОтдела=Сотрудник.КодСотрудника  
**GROUP BY**            Название
- e. **SELECT**            Название, Count(Фамилия)  
**FROM**                Сотрудник, Отдел  
**WHERE**                Отдел.КодОтдела=Сотрудник.КодСотрудника  
**GROOP BY**            Название

8. Дана схема базы данных.

Необходимо посчитать, сколько проектов выполнил каждый сотрудник.

SQL-запрос будет выглядеть следующим образом:



```

SELECT    Фамилия, Count(Проект.Название)
FROM      Сотрудник С, УчастиеВПроекте У, Проект П
WHERE     П.КодПроекта=У.КодПроекта
          AND С.КодСотрудника= У.КодСотрудника.

```

Что произойдет в результате выполнения запроса?

Выберите один ответ:

- Запрос не выполнится.
- Появится диалоговое окно для уточнения параметров запроса.
- Запрос выполнится правильно.
- Запрос выполнится, но выведет неверные значения.

9. SQL-запрос выглядит следующим образом:

```

SELECT    Фамилия, Название
FROM      Сотрудник, Должность.

```

В таблице Сотрудник 50 записей, а в таблице Должность – 10 записей.

Сколько записей будет в результирующей таблице?

- 500.
- 10.
- 5.
- 50.

10. Что называют сортировкой данных в БД?

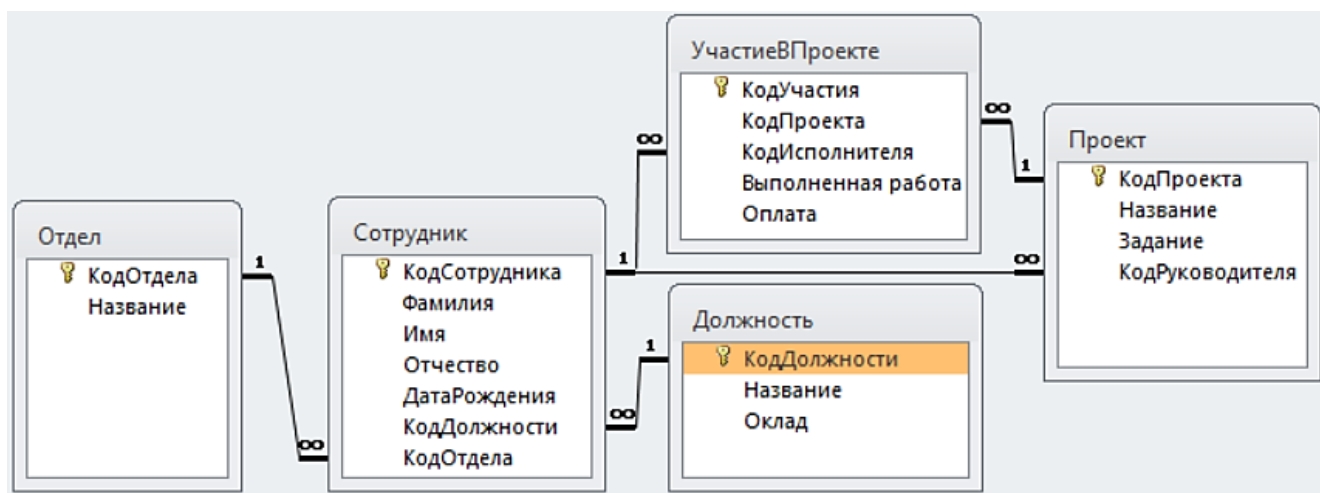
- Упорядочение записей по значениям одного из полей.

- b. Отбор записей, удовлетворяющих условиям поиска.
- c. Нет правильного ответа.
- d. Вывод на печать упорядоченных записей.

11. Какое поле нельзя считать уникальным?

- a. Поле, которое носит уникальное имя.
- b. Поле, значения которого имеют свойство наращивания.
- c. Ключевое поле.
- d. Поле, значения в котором не могут повторяться.

12. Дана схема базы данных. Необходимо найти сотрудника с максимальным окладом.



Необходимо найти сотрудника с максимальным окладом. SQL-запрос будет выглядеть следующим образом:

```

SELECT    Фамилия, MAX(Д.Оклад)
FROM      Сотрудник С, Должность Д
WHERE      С.КодДолжности= Д.КодДолжности
GROUP BY  Фамилия
  
```

Почему в результате выполнения запроса получится неверный результат?

Как бы вы исправили данный запрос, чтобы он выводил нужную информацию?

13. SQL-запрос выглядит следующим образом:

```

SELECT    Фамилия
  
```

**FROM** Сотрудник С1  
**WHERE** КодДолжности=(  
**SELECT** КодДолжности  
**FROM** Сотрудник С2  
**WHERE** Фамилия **LIKE** 'Иванов'  
**AND** С2.КодСотрудника<>С1.КодСотрудника).

Что делает этот запрос?

## **4 Рекомендуемая литература**

### **Основная учебная литература**

1. Маркин, А.В. Программирование на SQL. В 2 частях. Ч. 1: учебник и практикум для вузов / А.В. Маркин. – Москва : Изд-во Юрайт, 2022. – 403 с. – ISBN 978-5-534-12256-5. - URL: <https://urait.ru/viewer/programmirovanie-na-sql-v-2-ch-chast-1-491238#page/2> (дата обращения: 14.12.2022). – Текст : электронный.
2. Маркин, А.В. Программирование на SQL. В 2 частях. Ч. 2: учебник и практикум для вузов / А.В. Маркин. – Москва : Изд-во Юрайт, 2022. – 340 с. – ISBN 978-5-534-12258-9. - URL: <https://urait.ru/viewer/programmirovanie-na-sql-v-2-ch-chast-2-490104#page/2> (дата обращения: 14.12.2022). – Текст : электронный.

### **Дополнительная литература**

1. Стасышин, В.М. Базы данных: технологии доступа: учебное пособие для вузов / В.М. Стасышин, Т.Л. Стасышина. – Москва : Издательство «Диалог-МИФИ», 2014. – 164 с. – ISBN 978-5-86404-227-4. – URL: <https://urait.ru/viewer/bazy-dannyh-tehnologii-dostupa-514252#page/2>. - (дата обращения: 12.12.2022). – Текст : электронный.
2. Волк, В.К. Базы данных. Проектирование, программирование, управление и администрирование : учебник / В.К. Волк. – Санкт-Петербург : Лань, 2020. – 244 с. – ISBN 978-5-8114-4189-1. - URL: <https://e.lanbook.com/reader/book/126933> (дата обращения: 12.12.2022). – Текст : электронный.

### **Литература для оформления отчета по лабораторной работе**

Правила оформления учебных работ студентов : учебно-методическое пособие / И.А. Жибинова, А.Е. Аракелян, О.В. Соколова, Ю.Н. Соина-Кутищева. – Новокузнецк : НФИ КемГУ, 2018. – 124 с. – Текст : непосредственный

### **Современные профессиональные базы данных и справочные системы**

CITForum.ru : on-line библиотека свободно доступных материалов по информационным технологиям на русском языке : сайт. – 2001 – URL: <http://citforum.ru> (дата обращения: 22.03.2020). – Текст: электронный.

eLIBRARY.RU : научная электронная библиотека : сайт. – Москва, 2000 - . – URL: <http://www.elibrary.ru> (дата обращения: 22.03.2020). – Режим доступа: для зарегистрир. пользвателей. – Текст: электронный.

Единое окно доступа к образовательным ресурсам : сайт. – Москва, 2005 - . – URL: <http://window.edu.ru/> (дата обращения: 22.03.2020). –Текст: электронный.