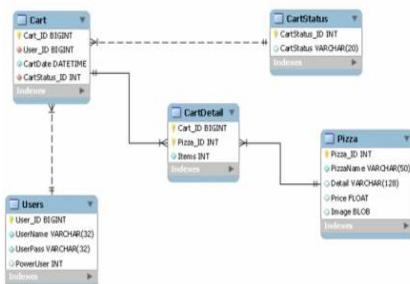


МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РФ

ФГБОУ ВО БРЯНСКИЙ ГАУ

КАФЕДРА ИНФОРМАТИКИ,  
ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

*Лысенкова С.Н.*



***Распределенные  
базы данных  
Основы языка SQL***

*Методическое пособие  
для студентов направления подготовки  
09.04.03. Прикладная информатика*

Брянская область  
2022

УДК 004.75 (07)

ББК 32.972

Л 88

Лысенкова, С. Н. «Распределенные базы данных». Основы языка SQL: методическое пособие для студентов направления подготовки 09.04.03 Прикладная информатика / С. Н. Лысенкова. - Брянск: Изд-во Брянский ГАУ, 2022. - 36 с.

Издание окажет помощь магистрантам при выполнении лабораторно-практических работ по созданию SQL-запросов. Издание предназначено для студентов направления подготовки: 09.04.03 Прикладная информатика.

Рецензенты: к.п.н., доцент Петракова Н.В., зам. генерального директора по развитию ООО НПО «ГКМП» Добровольский Г.И.

Рекомендовано к изданию решением учебно-методической комиссии института энергетики и природопользования Брянского ГАУ, протокол № 3 от 28 октября 2022 г.

© Брянский ГАУ, 2022

© С.Н. Лысенкова, 2022

## Содержание

Введение	4
1. Элементы языка SQL	5
2. Структура SQL	7
3. Типы данных	13
4. Простые запросы	14
5. Использование реляционных и логических операторов	18
6. Использование специальных операторов в условиях поиска	22
7. Использование обобщающих функций языка SQL	25
8. Многотабличные запросы	29
9. Изменение содержимого базы данных	31
Литература	35

## **Введение**

Распределенная база данных представляет собой группу баз данных, которая выглядит для пользователей и приложений как одна база данных. В большинстве случаев базы данных, составляющие распределенную базу данных, расположены на отдельных компьютерах, взаимодействующих по сети. Распределенные БД определяют сегодня развитие технологий реляционных баз данных и языка SQL.

Задачи SQL в распределенных БД: удаленный запрос, удаленная транзакция, распределенные транзакции, распределенные запросы.

## 1. Элементы языка SQL

**SQL** (язык структурированных запросов») – декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

Инструкция SQL – это строка SQL, определяющая команду

*Пример:*

*SELECT Наименование, Склад*

*FROM Комплекующие;*

Инструкция SQL состоит из зарезервированных слов, а также из слов, определяемых пользователем. **Зарезервированные** слова являются постоянной частью языка SQL и имеют фиксированное значение. Слова, **определяемые** пользователем, задаются самим и представляют собой имена различных объектов базы данных. Хотя в стандарте это не указано, многие инструкции языка SQL требуют задания в конце оператора некоторого символа, обозначающего окончание его текста (как правило, с этой целью используется символ точки с запятой (;)).

Большинство компонентов SQL не чувствительно к регистру. Рекомендуется придерживаться следующих правил.

- каждая фраза в инструкции должна начинаться с новой строки.
- начало каждой фразы должно быть выровнено с началом остальных инструкций.
- если инструкция имеет несколько частей, каждая из них должна начинаться с новой строки с некоторым отступом относительно начала фразы, что будет указывать на их подчиненность.

## **Язык SQL представляет собой совокупность**

К основным элементам SQL можно отнести: операторы (команды), типы данных, значения, функции, выражения, предложения и предикаты.

Язык SQL состоит из ограниченного числа **команд**, специально предназначенных для управления данными. Одни из этих команд служат для определения данных, другие – для их обработки, остальные – для администрирования.

**Тип данных SQL** – это совокупность значений, способных представлять данные. В SQL признаны только шесть предопределенных общих типов: числовой, строковый, логический, дата-время, интервальный и XML.

**Значения** можно получать несколькими способами - непосредственно или с помощью функций или выражений. **Функция** принимает данные и на их основе вычисляет значение. **Выражение** является комбинацией элементов данных, из которой SQL в результате вычислений получает одно значение. В SQL существует несколько видов значений: значения типа записи; литеральные значения; переменные; специальные переменные; ссылки на столбцы.

SQL позволяет использовать некоторые характеристики самих данных для определения, представляет ли конкретная строка интерес для вас. Операторы SELECT, DELETE и UPDATE сообщают ядру базы данных (т.е. той части СУБД, которая непосредственно взаимодействует с данными), какие именно строки нужно отбирать, удалять или обновлять. Для отсеивания нужных строк в инструкции SELECT, DELETE и UPDATE добавляются **уточняющие предложения**.

**Предикаты** - это используемые в SQL эквиваленты логических высказываний.

## 2. Структура SQL

SQL не является языком программирования общего назначения, но обладает рядом достаточно мощных средств. Все необходимые действия по созданию, изменению, поддержке базы данных и обеспечению ее безопасности выполняются с помощью входящих в состав SQL языков.

**1. Язык определения данных (DDL).** Это та часть SQL, которая используется для создания (полного определения) базы данных, изменения ее структуры и удаления базы после того, как она становится ненужной (CREATE, DROP, ALTER).

**2. Язык манипулирования данными (DML).** Он предназначен для поддержки базы данных. С помощью этого мощного инструмента можно точно указать, что именно нужно сделать с данными, находящимися в базе, — добавить, изменить или извлечь.

**3. Язык запросов DQL** наиболее известен пользователям реляционной базы данных, несмотря на то, что он включает всего одну команду SELECT. Эта команда вместе со своими многочисленными опциями и предложениями используется для формирования запросов к реляционной базе данных.

**4. Команды администрирования данных.** С помощью команд администрирования данных пользователь осуществляет контроль за выполняемыми действиями и анализирует операции базы данных; они также могут оказаться полезными при анализе производительности системы. Не следует путать администрирование данных с администрированием базы данных, которое представляет собой общее управление базой данных и подразумевает использование команд всех уровней.

## 5. Команды управления транзакциями (TCL).

Транзакцией называется последовательность операций, производимых над базой данных и переводящих базу данных из одного непротиворечивого (согласованного) состояния в другое непротиворечивое (согласованное) состояние. Транзакция рассматривается как некоторое неделимое действие над базой данных, осмысленное с точки зрения пользователя. Существуют следующие команды, позволяющие управлять транзакциями базы данных: COMMIT, ROLLBACK, SAVEPOINT, SET TRANSACTION.

6. Программный SQL предназначен для того, чтобы встраивать SQL-запросы в прикладную программу, написанную на одном из языков программирования.

*Таблица 1. Операторы определения данных DDL*

Оператор	Смысл	Действие
CREATE <i>TABLE</i>	Создать таб-лицу	Создает новую таблицу в БД
DROP <i>TABLE</i>	Удалить таб-лицу	Удаляет таблицу из БД
ALTER <i>TABLE</i>	Изменить таб-лицу	Изменяет структуру существующей таблицы или ограничения целостности, задаваемые для данной таблицы
CREATE <i>INDEX</i>	Создать индекс	Создает индекс для некоторой таблицы для обеспечения быстрого доступа по атрибутам, входящим в индекс
DROP <i>INDEX</i>	Удалить индекс	Удаляет ранее созданный индекс
GRANT	Предоставляет пользователю	Используется для назначения разрешений на выполнения

	(группе) разрешения на определенные операции с объектом	операций с объектом базы данных определенному пользователю или группе пользователей.
REVOKE	Отзывает ранее выданные разрешения	Используется для снятия разрешений на выполнения операций с объектом базы данных определенному пользователю или группе пользователей
DENY	Задаёт запрет, имеющий приоритет над разрешением	Налагает явный запрет на предоставление учетным именам некоторых разрешений, а также на наследование этих разрешений посредством членства в группе или роли

**Таблица 2. Операторы манипулирования данными Data Manipulation Language (DML)**

<b>Оператор</b>	<b>Смысл</b>	<b>Действие</b>
DELETE	Удалить строки	Удаляет одну или несколько строк, соответствующих условиям фильтрации, из базовой таблицы. Применение оператора согласуется с принципами поддержки целостности, поэтому этот оператор не всегда может быть выполнен корректно, даже если синтаксически он записан правильно

INSERT	Вставить строку	Вставляет одну строку в базовую таблицу.
UPDATE	Обновить строку	Обновляет значения одного или нескольких столбцов в одной или нескольких строках, соответствующих условиям фильтрации

*Таблица 3. Язык запросов Data Query Language (DQL)*

SELECT	Выбрать строки	Оператор, заменяющий все операторы реляционной алгебры и позволяющий сформировать результирующее отношение, соответствующее запросу
--------	----------------	---

*Таблица 4. Средства управления транзакциями (TCL)*

<b>Оператор</b>	<b>Смысл</b>	<b>Действие</b>
COMMIT	Завершить транзакцию	Завершить комплексную взаимосвязанную обработку информации, объединенную в транзакцию
ROLLBACK	Откатить транзакцию	Отменить изменения, проведенные в ходе выполнения транзакции
SAVEPOINT	Сохранить промежуточную точку выполнения транзакции	Сохранить промежуточное состояние БД, пометить его для того, чтобы можно было в дальнейшем к нему вернуться

*Таблица 5. Средства администрирования данных*

<b>Оператор</b>	<b>Смысл</b>	<b>Действие</b>
ALTER DATABASE	Изменить БД	Изменить набор основных объектов в базе данных, ограничений, касающихся всей базы данных
ALTER DBAREA	Изменить область хранения БД	Изменить ранее созданную область хранения
ALTER PASSWORD	Изменить пароль	Изменить пароль для всей базы данных
CREATE DATABASE	Создать БД	Создать новую базу данных, определив основные параметры для нее
CREATE DBAREA	Создать область хранения	Создать новую область хранения и сделать ее доступной для размещения данных
DROP DATABASE	Удалить БД	Удалить существующую базу данных (только в том случае, когда вы имеете право выполнить это действие)
DROP DBAREA	Удалить область хранения БД	Удалить существующую область хранения (если в ней на настоящий момент не располагаются активные данные)

*Таблица 6. Программный SQL*

<b>Оператор</b>	<b>Смысл</b>	<b>Действие</b>
DECLARE	Определяет курсор для запроса	Задает некоторое имя и определяет связанный с ним запрос к БД, который соответствует виртуальному набору данных
OPEN	Открыть курсор	Формирует виртуальный набор данных, соответствующий описанию указанного курсора и текущему состоянию БД
FETCH	Считать строку из множества строк, определенных курсором	Считывает очередную строку, заданную параметром команды из виртуального набора данных, соответствующего открытому курсору
CLOSE	Закрывает курсор	Прекращает доступ к виртуальному набору данных, соответствующему указанному курсору
PREPARE	Подготовить оператор SQL к динамическому выполнению	Сгенерировать план выполнения запроса, соответствующего заданному оператору SQL
EXECUTE	Выполнить оператор SQL, ранее подготовленный к динамическому выполнению	Выполняет ранее подготовленный план запроса

В коммерческих СУБД набор основных операторов расширен. В большинство СУБД включены операторы определения и запуска хранимых процедур и операторы определения триггеров.

### 3. Типы данных

Данные, хранящиеся в столбцах таблиц SQL-ориентированной базы данных, являются типизированными, т. е. представляют собой значения одного из типов данных. Для этого при определении таблицы каждому ее столбцу назначается некоторый тип данных (или домен), и в дальнейшем СУБД должна следить, чтобы в каждом столбце каждой строки каждой таблицы присутствовали только допустимые значения.

*Таблица 7. Типы данных в языке SQL/89*

Типы данных	Описание
INT	Целое число.
REAL	Число с плавающей запятой. Этот тип допускает больший диапазон значений, чем INT, но не обладает его точностью.
TEXT(length)	Символьная величина переменной длины
DATE	Стандартное значение даты.
TIME	Стандартное значение времени. Этот тип используется для хранения времени дня безотносительно какой-либо даты. При использовании вместе с датой позволяет хранить конкретную дату и время. В SQL есть дополнительный тип DATETIME для совместного хранения даты и времени в одном поле.

CHAR(length)	Символьная величина фиксированной длины. Поля типа CHAR не могут содержать строки длины большей, чем указанное значение. Поля меньшей длины дополняются пробелами. Вероятно, это наиболее употребительный тип в любой реализации SQL.
--------------	---

В стандарте SQL92 добавлены следующие типы данных:

- VARCHAR(n) – строки символов переменной длины.
- NCHAR(N) – строки локализованных символов постоянной длины.
- NCHAR VARYING(n) – строки локализованных символов переменной длины.
- BIT(n) – строка битов постоянной длины.
- BIT VARYING(n) – строка битов переменной длины.
- DATE – календарная дата.
- TIMESTAMP(точность) – дата и время.
- INTERVAL – временной интервал.

#### 4. Простые запросы

Запрос - команда к программе базы данных, которая сообщает ей чтобы она вывела определенную информацию из таблиц в память. Эта информация обычно посылается непосредственно на экран компьютера которым вы пользуетесь, хотя, в большинстве случаев, ее можно также послать принтеру, сохранить в файле, или представить как вводную информацию для другой команды или процесса.

Запросы обычно рассматриваются как часть языка

DML. Запрос не меняет информацию в таблицах, а просто показывает ее пользователю.

Все запросы в SQL состоят из одиночной команды - SELECT (ВЫБОР). Операция SELECT является закрытой - результат запроса к таблице представляет собой другую таблицу.

В общем случае, команда SELECT начинается с ключевого слова SELECT, сопровождаемого пробелом, после этого должен следовать список имен столбцов которые вы хотите видеть, отделяемые запятыми. Если необходимо вывести все столбцы таблицы, можно заменить этот список звездочкой (\*). Ключевое слово FROM следующее далее, сопровождается пробелом и именем таблицы запрос к которой делается. В заключение, точка с запятой (;) должна использоваться чтобы закончить запрос и указать что команда готова к выполнению.

**Задание:** Создать и заполнить таблицы используя язык SQL.

**Порядок выполнения:**

1. Откройте базу данных *СКЛАД*, а затем выберите из списка Объекты пункт **Запросы**.
2. В области задач, расположенной в правой части окна, выберите вариант **Создание запроса в режиме конструктора**. Откроется диалоговое окно **Добавить таблицу**, закрыть его.
3. В главном меню Access выберите команду **Режим/Режим SQL**. Откроется окно редактора с начальным фрагментом инструкции языка SQL SELECT.
4. Удалите эту инструкцию, а затем введите нужную.

Выполните запрос нажав

  
Выполнить

5. Закончив работу, щелкните на пиктограмме **Сохранить**. Access предложит ввести имя для только что созданного запроса.

6. Введите имя запроса и щелкните на кнопке **ОК**.

### Использование простых запросов SQL

#### Запрос 1.

Выбрать все столбцы из таблицы Поставщик

#### Инструкция

**SELECT \***

**FROM Поставщик;**

Код_пост.	Фирма_поставщик	Реквизиты	Адрес	Телефон	E-mail	Представитель
3000	ОАО Брянск мясо	45429410908040100035	Ул. Литейная, д. 68, кв. 51	(0832) 54-64-56	brm@rambler.ru	Павлов Геннадий Юрьевич
3001	ОАО Мяспрод	76473547635676400063	Ул. Емлютина, д. 44, кв. 78	(0832) 75-46-33	myasprod@bk.ru	Загладин Никитий Вадимович
3002	ОАО Главпродукт	7575946564345300022	Ул. Новозыбковская, д. 14, кв. 69	(0832) 34-06-56	glprod@mail.ru	Ткачук Юрий Константинович
3003	ООО Мясо	43264733232364500053	Ул. Вокзальная, д. 103, кв. 273	(095) 88-34-356	myas@rambler.ru	Андреев Сергей Егорович
3004	ОАО Мясопродукты	38839686433243200064	Ул. Пушкина, д. 21, кв. 165	(095) 77-46-112	mprod@yandex.ru	Репин Константин Андреевич
3005	ООО Престиж	72039623602650900072	Ул. Петровская, д. 12, кв. 65	(0832) 75-26-62	prestizh@list.ru	Ухов Василий Петрович
3006	ОАО Главмясо	69862862897206900063	Ул. Пионерская, д. 66, кв. 8	(095) 12-67-135	glavmyaso@mail.ru	Васильцов Иван Григорьевич
3007	ООО Мяспром	26256234623698600013	Ул. Советская, д. 43, кв. 82	(095) 24-77-177	myaspromo@mail.ru	Прошкин Степан Иванович
3008	ОАО Регионмясо	89072092809236700052	Ул. Лермонтова, д. 14, кв. 14	(095) 24-33-333	regionmyaso@rambl	Егорьев Николай Александр
3009	ООО Гормясо	52163458971698752062	Ул. Советская, д. 13, кв. 62	(0832) 51-51-61	gormyaso@yandex.r	Прохоров Иван Михайлович
3010	ОАО Брянскпродукт	84325634856918369238	Ул. Ивановская, д. 56, кв. 3	(095) 32-86-34	Bryanskprod@bk.ru	Пирогов Михаил Иванович

#### Запрос 2.

Выбрать поля Фирма\_поставщик, Телефон из таблицы Поставщик

#### Инструкция

**SELECT Фирма\_поставщик, Телефон**

**FROM Поставщик;**

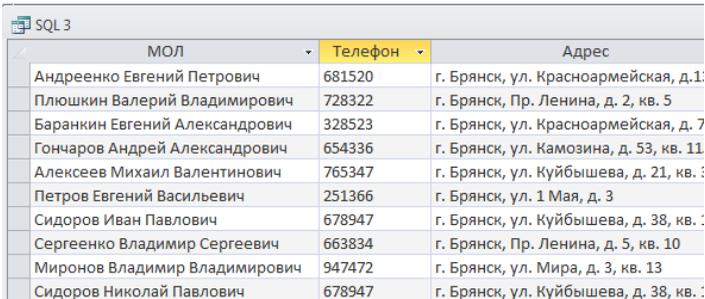
Фирма_поставщик	Телефон
ОАО Брянск мясо	(0832) 54-64-56
ОАО Мяспрод	(0832) 75-46-33
ОАО Главпродукт	(0832) 34-06-56
ООО Мясо	(095) 88-34-356
ОАО Мясопродукты	(095) 77-46-112
ООО Престиж	(0832) 75-26-62
ОАО Главмясо	(095) 12-67-135
ООО Мяспром	(095) 24-77-177
ОАО Регионмясо	(095) 24-33-333
ООО Гормясо	(0832) 51-51-61
ОАО Брянскпродукт	(095) 32-86-34

### Задание 3.

Вывести на экран поля из таблицы МОЛ в следующем порядке МОЛ, Телефон, Адрес

#### Инструкция

```
SELECT МОЛ, Телефон, Адрес  
FROM МОЛ;
```



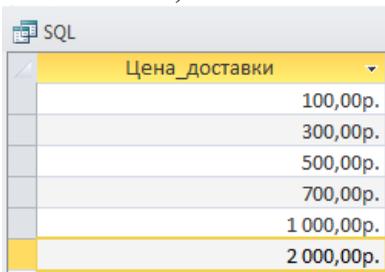
МОЛ	Телефон	Адрес
Андрееенко Евгений Петрович	681520	г. Брянск, ул. Красноармейская, д.1.
Плюшкин Валерий Владимирович	728322	г. Брянск, Пр. Ленина, д. 2, кв. 5
Баранкин Евгений Александрович	328523	г. Брянск, ул. Красноармейская, д. 7
Гончаров Андрей Александрович	654336	г. Брянск, ул. Камозина, д. 53, кв. 11
Алексеев Михаил Валентинович	765347	г. Брянск, ул. Куйбышева, д. 21, кв. 3
Петров Евгений Васильевич	251366	г. Брянск, ул. 1 Мая, д. 3
Сидоров Иван Павлович	678947	г. Брянск, ул. Куйбышева, д. 38, кв. 10
Сергеенко Владимир Сергеевич	663834	г. Брянск, Пр. Ленина, д. 5, кв. 10
Миرونор Владимир Владимирович	947472	г. Брянск, ул. Мира, д. 3, кв. 13
Сидоров Николай Павлович	678947	г. Брянск, ул. Куйбышева, д. 38, кв. 10

### Задание 4

Выбрать из таблицы Заказы цену доставки без повторения

#### Инструкция

```
SELECT DISTINCT Цена_доставки  
FROM Заказ;
```



Цена_доставки
100,00р.
300,00р.
500,00р.
700,00р.
1 000,00р.
2 000,00р.

### Задание 5.1

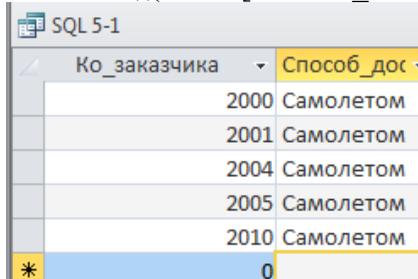
Из таблицы Заказы выбрать коды всех заказчиков доставки товаров которым осуществляется с самолетом

#### Инструкция

```
SELECT Заказ.Ко_заказчика, Заказ.Способ_доставки
```

FROM Заказ

WHERE (((Заказ.[Способ\_доставки])="Самолетом"));



Ко_заказчика	Способ_дос
2000	Самолетом
2001	Самолетом
2004	Самолетом
2005	Самолетом
2010	Самолетом
*	0

### Задание 5.2

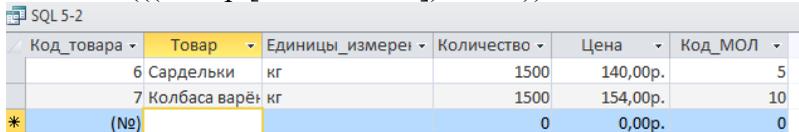
Для числового поля выбрать товар с количеством 1500 единиц

#### Инструкция

SELECT \*

FROM Товар

WHERE (((Товар.[Количество])=1500));



Код_товара	Товар	Единицы_измерен	Количество	Цена	Код_МОЛ
6	Сардельки	кг	1500	140,00р.	5
7	Колбаса варёная	кг	1500	154,00р.	10
*	(№)		0	0,00р.	0

## 5. Использование реляционных и логических операторов

Реляционный оператор - математический символ, который указывает на определенный тип сравнения между двумя значениями. В языке SQL можно использовать следующие операторы сравнения:

- = Равный
- > Больше чем
- < Меньше чем
- >= Больше чем или равно

<= Менше чем или равно

0 Не равно

Более сложные предикаты могут быть построены с помощью логических операторов AND, OR или NOT, а также с помощью скобок, используемых для определения порядка вычисления выражения.

- AND берет два подвыражения (в форме A AND B) как аргументы и оценивает их по отношению к истине, верны ли они оба.
- OR берет два подвыражения (в форме A OR B) как аргументы и оценивает на правильность, верен ли один из них.
- NOT берет одиночное подвыражения (в форме NOT A) как аргумент и заменяет его значение с неверного на верное или верное на неверное.

Вычисление выражений в условиях выполняется по следующим правилам:

- выражение вычисляется слева направо;
- первыми вычисляются подвыражения в скобках;
- операторы NOT выполняются до выполнения операторов AND и OR;
- операторы AND выполняются до выполнения операторов OR.

Для устранения любой возможной неоднозначности рекомендуется использовать скобки.

В базе данных Склад выполнить следующие запросы:

### **Задание 1**

Отобразить заказы с ценой доставки более 900 руб.

#### Инструкция

```
SELECT *
```

```
FROM Заказ
```

```
WHERE Цена_доставки >900;
```

2 SQL 1	Ко_заказчика	Код_товара	Дата_заказ:	Способ_дос	Цена_доставки	Об
	2001	3	04.10.2011	Самолетом	1 000,00р.	
	2003	2	14.05.2011	Машиной	2 000,00р.	
	2004	3	07.12.2011	Самолетом	1 000,00р.	
	2009	9	22.11.2011	Пароходом	1 000,00р.	
*	0	0			0,00р.	

## Задание 2

Выбрать способ доставки заказа Самолетом с ценой доставки менее 500 руб.

### Инструкция

SELECT \*

FROM Заказ

WHERE Способ\_доставки ="Самолетом" AND Цена\_доставки <500;

2 SQL 2	Ко_заказчика	Код_товара	Дата_заказ:	Способ_дос	Цена_доставки
	2005	6	25.11.2011	Самолетом	300,00р.
	2010	5	05.07.2011	Самолетом	300,00р.
*	0	0			0,00р.

## Задание 3

Выбрать способ доставки заказа Самолетом или с ценой доставки менее 500 руб.

### Инструкция

SELECT \*

FROM Заказ

WHERE Способ\_доставки ="Самолетом" OR Цена\_доставки <500;

2 SQL 3	Ко_заказчика	Код_товара	Дата_заказ:	Способ_дос	Цена_доставки
	2000	1	25.12.2011	Самолетом	500,00р.
	2000	4	18.09.2011	Пароходом	100,00р.
	2001	3	04.10.2011	Самолетом	1 000,00р.
	2004	3	07.12.2011	Самолетом	1 000,00р.
	2005	6	25.11.2011	Самолетом	300,00р.
	2005	10	14.12.2011	Машиной	100,00р.
	2010	5	05.07.2011	Самолетом	300,00р.
*	0	0			0,00р.

#### Задание 4

Выбрать любой способ доставки заказа, кроме Самолетом или с ценой доставки менее 500 руб.

#### Инструкция

SELECT \*

FROM Заказ

WHERE NOT Способ\_доставки ="Самолетом" OR Цена\_доставки <500;

Ко_заказчика	Код_товара	Дата_заказ	Способ_дос	Цена_доставки
2000	4	18.09.2011	Пароходом	100,00р.
2003	2	14.05.2011	Машиной	2 000,00р.
2005	6	25.11.2011	Самолетом	300,00р.
2005	10	14.12.2011	Машиной	100,00р.
2007	7	06.09.2011	Машиной	700,00р.
2009	9	22.11.2011	Пароходом	1 000,00р.
2010	5	05.07.2011	Самолетом	300,00р.
*	0	0		0,00р.

#### Задание 5

Выбрать любой способ доставки заказа, кроме Самолетом и с ценой доставки не менее 500 руб.

#### Инструкция

SELECT \*

FROM Заказ

WHERE NOT Способ\_доставки ="Самолетом" AND Цена\_доставки >500;

Ко_заказчика	Код_товара	Дата_заказ	Способ_дос	Цена_доставки
2000	1	25.12.2011	Самолетом	500,00р.
2000	4	18.09.2011	Пароходом	100,00р.
2003	2	14.05.2011	Машиной	2 000,00р.
2005	6	25.11.2011	Самолетом	300,00р.
2005	10	14.12.2011	Машиной	100,00р.
2007	7	06.09.2011	Машиной	700,00р.
2009	9	22.11.2011	Пароходом	1 000,00р.
2010	5	05.07.2011	Самолетом	300,00р.
*	0	0		0,00р.

## 6. Использование специальных операторов в условиях поиска

В дополнении к реляционным и логическим операторам SQL использует специальные операторы IN, BETWEEN, LIKE.

Оператор **IN** определяет набор значений в который данное значение может или не может быть включено. **IN** определяет набор значений с помощью имен членов набора заключенных в круглые скобки и отделенных запятыми. Он затем проверяет различные значения указанного поля пытаясь найти совпадение со значениями из набора.

Оператор **BETWEEN** определяет диапазон, значения которого должны уменьшаться, что делает предикат верным. Вы должны ввести ключевое слово **BETWEEN** с начальным значением, ключевое **AND** и конечное значение. В отличие от **IN**, **BETWEEN** чувствителен к порядку, и первое значение в предложении должно быть первым по алфавитному или числовому порядку.

Оператор **LIKE** применяется для организации поиска с помощью шаблонов.

В языке SQL существует два специальных символа шаблона, используемых при проверке символьных значений:

- символ подчеркивания (  ) замещает любой одиночный символ. Например, "b\_t" будет соответствовать словам "bat" или "bit", но не будет соответствовать "brat". Для Ms Access - знак вопроса (?);
- знак процента (%) замещает последовательность любого числа символов (включая символы нуля). Например "%p%t" будет соответствовать словам "put", "posit", или "opt", но не "spite". Ms Access - звездочка (\*);

*Условия поиска с проверкой вхождения во множество.*

### **Задание 1**

Найти все заказы товаров, доставляемых пароходом или самолетом.

#### Инструкция

SELECT \*

FROM Заказ

WHERE Способ\_доставки IN ("Пароходом", "Самолетом");

Ко_заказчика	Код_товара	Дата_заказ	Способ_дос	Цена_доставки
2000	1	25.12.2011	Самолетом	500,00р.
2000	4	18.09.2011	Пароходом	100,00р.
2001	3	04.10.2011	Самолетом	1 000,00р.
2004	3	07.12.2011	Самолетом	1 000,00р.
2005	6	25.11.2011	Самолетом	300,00р.
2009	9	22.11.2011	Пароходом	1 000,00р.
2010	5	05.07.2011	Самолетом	300,00р.
*	0	0		0,00р.

### **Задание 2**

Найти все товары, поставляемые заказчиками с кодом 2000, 2005, 2001.

#### Инструкция

SELECT \*

FROM Заказ

WHERE Ко\_заказчика IN (2000, 2005, 2001);

Ко_заказчика	Код_товара	Дата_заказ	Способ_дос	Цена_доставки
2000	1	25.12.2011	Самолетом	500,00р.
2000	4	18.09.2011	Пароходом	100,00р.
2001	3	04.10.2011	Самолетом	1 000,00р.
2005	6	25.11.2011	Самолетом	300,00р.
2005	10	14.12.2011	Машиной	100,00р.
*	0	0		0,00р.

*Использование диапазонов в условиях поиска.*

### **Задание 3**

Найти все заказы с ценой доставки от 300 до 1100 руб.

#### Инструкция

SELECT \*

FROM Заказ  
WHERE Цена\_доставки BETWEEN 300 AND 1100;

SQL 3	Ко_заказчика	Код_товара	Дата_заказ	Способ_дос	Цена_доставки
	2000	1	25.12.2011	Самолетом	500,00р.
	2001	3	04.10.2011	Самолетом	1 000,00р.
	2004	3	07.12.2011	Самолетом	1 000,00р.
	2005	6	25.11.2011	Самолетом	300,00р.
	2007	7	06.09.2011	Машиной	700,00р.
	2009	9	22.11.2011	Пароходом	1 000,00р.
	2010	5	05.07.2011	Самолетом	300,00р.
*	0	0			0,00р.

*Условия поиска с указанием шаблонов.*

#### Задание 4

Найти поставщиков с организационно-правовой формой собственности – «ООО».

#### Инструкция

SELECT \*  
FROM Поставщик  
WHERE Фирма\_поставщик LIKE "ООО\*";

SQL 4	Код_постав	Фирма_постав	Реквизиты	Адрес
	3003	ООО Мясо	43264733232364500053	Ул. Вокзальная, д. 103, кв. 273 (095
	3005	ООО Престиж	72039623602650900072	Ул. Петровская, д. 12, кв. 65 (083
	3007	ООО Мяспром	26256234623698600013	Ул. Советская, д. 43, кв. 82 (095
	3009	ООО Гормясо	52163458971698752062	Ул. Советская, д. 13, кв. 62 (083
*				

*Сортировка результатов.*

#### Задание 5

Отсортировать товары по цене в порядке убывания.

#### Инструкция

SELECT \*  
FROM Товар  
ORDER BY Цена DESC;

Код_товара	Товар	Единицы_измере	Количество	Цена	Кс
5	Ветчина	кг	7000	348,00р.	
12	Грудинка	кг	1000	320,00р.	
4	Балык	кг	5000	235,00р.	
8	Котлеты	кг	3000	230,00р.	
11	Фарш	кг	400	210,00р.	
9	Пельмени	кг	1000	170,00р.	
7	Колбаса варён	кг	1500	154,00р.	
1	Колбаса копчё	кг	10000	145,00р.	
6	Сардельки	кг	1500	140,00р.	
2	Сосиски	кг	5000	120,00р.	
3	Окорочка	кг	2000	50,00р.	
10	Хинкали	кг	900	40,00р.	
*	(№)		0	0,00р.	

## 7. Использование обобщающих функций языка SQL

Стандарт ISO содержит определение следующих пяти обобщающих (агрегатных) функций:

**COUNT** - возвращает количество значений в указанном столбце;

**SUM** - возвращает сумму значений в указанном столбце;

**AVG** - возвращает усредненное значение в указанном столбце;

**MIN** - возвращает минимальное значение в указанном столбце;

**MAX** - возвращает максимальное значение в указанном столбце.

Все функции оперируют со значениями в единственном столбце таблицы и возвращают единственное значение. Функции **COUNT**, **MIN** и **MAX** применимы как к числовым, так и к нечисловым полям, тогда как функ-

ции SUM и AVG могут использоваться только в случае числовых полей.

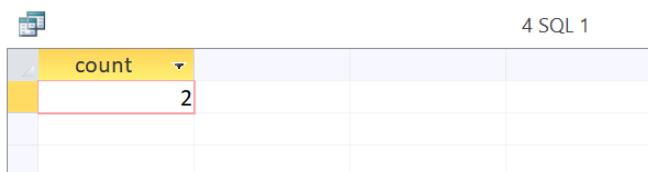
### Обобщающие функции

#### Запрос 1

Использование функции COUNT. Посчитать количество МОЛ, чьи фамилии начинаются на «А»

#### Инструкция

```
SELECT COUNT (МОЛ) AS [count]
FROM МОЛ
WHERE МОЛ LIKE "А*";
```



4 SQL 1

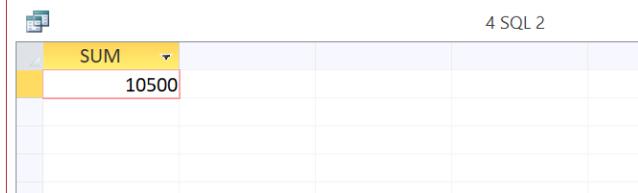
count			
2			

#### Запрос 2.

Использование функции SUM. Найти общий объем заказа

#### Инструкция

```
SELECT SUM (Объем_заказа) AS [SUM]
FROM Заказ;
```



4 SQL 2

SUM			
10500			

### Функции MIN, MAX, AVG

#### Запрос 3.

Найти минимальный Код заказчика, максимальный Объем заказа и среднюю цену доставки

#### Инструкция

```
SELECT Min(Ко_заказчика) AS [MIN],
       Max(Объем_заказа) AS [MAX], Avg(Цена_доставки)
       AS [AVG]
FROM Заказ;
```

4 SQL 3

MIN	MAX	AVG
2000	4000	700,00 Р

Запись: 1 из 1 Нет фильтра

*Группировка результатов*

#### **Запрос 4.**

##### Инструкция

Выбрать максимальное количество товара, числящегося за МОЛ

```
SELECT Код_МОЛ, Товар, MAX (Количество) AS
       [MAX]
```

```
FROM Товар
```

```
GROUP BY Код_МОЛ, Товар;
```

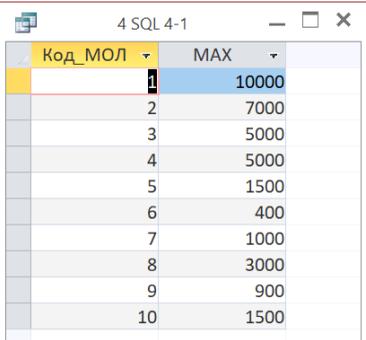
4 SQL 4

Код_МОЛ	Товар	MAX
1	Колбаса копчёная	10000
1	Окорочка	2000
2	Ветчина	7000
3	Балык	5000
4	Сосиски	5000
5	Сардельки	1500
6	Фарш	400
7	Пельмени	1000
8	Котлеты	3000
9	Хинкали	900
10	Грудинка	1000
10	Колбаса варёная	1500

ИЛИ

Инструкция

```
SELECT Код_МОЛ, MAX (Количество) AS [MAX]  
FROM Товар  
GROUP BY Код_МОЛ;
```



Код_МОЛ	MAX
1	10000
2	7000
3	5000
4	5000
5	1500
6	400
7	1000
8	3000
9	900
10	1500

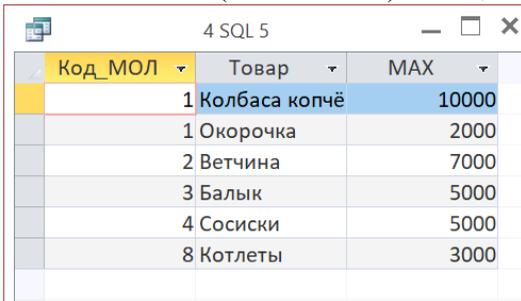
*Ограничения на выполнение группировки*

**Запрос 5**

Выбрать максимальное количество товара, числящегося за МОЛ, значение которого превышает 1500

Инструкция

```
SELECT Код_МОЛ, Товар, MAX (Количество) AS  
[MAX]  
FROM Товар  
GROUP BY Код_МОЛ, Товар  
HAVING MAX (Количество)>1500;
```



Код_МОЛ	Товар	MAX
1	Колбаса копчё	10000
1	Окорочка	2000
2	Ветчина	7000
3	Балык	5000
4	Сосиски	5000
8	Котлеты	3000

## 8. Многотабличные запросы

Одна из наиболее важных особенностей запросов SQL - это их способность определять связи между многочисленными таблицами и выводить информацию из них в терминах этих связей, всю внутри одной команды. Этот вид операции называется - соединением, которое является одним из видов операций в реляционных базах данных. В языке SQL операция соединения используется для объединения информации из двух таблиц посредством образования пар связанных строк, выбранных из каждой таблицы. Помещаемые в объединенную таблицу пары строк составляются по равенству входящих в них значений указанных столбцов.

Для выполнения соединения достаточно в предложении FROM указать имена двух или более таблиц, разделив их запятыми, после чего включить в запрос предложение WHERE с определением столбцов, используемых для соединения указанных таблиц. В предложении SELECT перечисляются все столбцы, которые должны быть помещены результирующую таблицу запроса.

При написании многотабличного запроса - используется полное имя столбца таблицы, состоящее из имени таблицы, сопровождаемого точкой и затем именем столбца.

Например:

Заказ.Код\_товара

*Простое соединение*

### **Запрос 1**

Вывести имена покупателей и продавцов, расположенных на ул. Советской

Инструкция

```

SELECT Заказчик.Фирма_заказчик, Заказчик.Адрес, По-
    ставщик.Фирма_поставщик, Поставщик.Адрес
FROM Заказчик, Поставщик
WHERE Заказчик.Адрес LIKE "Ул. Советская*" AND
    Поставщик.Адрес LIKE "Ул. Советская*";

```

Фирма_заказч	Заказчик.Адрес	Фирма_поставщик	Поставщик.Адрес
ООО Мяслик	Ул. Советская, д. 34, кв. 43	ООО Мяспром	Ул. Советская, д. 43, кв. 82
ОАО Поляна	Ул. Советская, д.43, кв. 51	ООО Мяспром	Ул. Советская, д. 43, кв. 82
ООО Мяслик	Ул. Советская, д. 34, кв. 43	ООО Гормясо	Ул. Советская, д. 13, кв. 62
ОАО Поляна	Ул. Советская, д.43, кв. 51	ООО Гормясо	Ул. Советская, д. 13, кв. 62

*Соединение таблиц через ссылочную целостность*

## Запрос 2

Вывести коды заказчиков и поставщиков товара

### Инструкция

```

SELECT Товар.Товар, Заказ.Ко_заказчика, Постав-
    ка.Код_поставщика
FROM Товар, Заказ, Поставка
WHERE Заказ.Код_товара=Поставка.Код_товара;

```

Товар	Ко_заказчика	Код_поставщика
Колбаса копчён	2000	3000
Колбаса копчён	2000	3004
Колбаса копчён	2003	3000
Колбаса копчён	2001	3001
Колбаса копчён	2004	3001
Колбаса копчён	2000	3002
Колбаса копчён	2005	3005
Колбаса копчён	2007	3008
Колбаса копчён	2007	3009
Колбаса копчён	2009	3007
Колбаса копчён	2005	3006
Сосиски	2000	3000
Сосиски	2000	3004
Сосиски	2003	3000

Запись: 19 из 132 | Нет фильтра | Поиск

## 9. Изменение содержимого базы данных

Язык SQL является полнофункциональным языком манипулирования данными, который может использоваться не только для выборки данных из базы, но и для модификации ее содержимого. Значения могут быть изменены, помещены и удалены из полей, тремя командами языка DML:

- `INSERT` — вставка данных в таблицу;
- `UPDATE` — обновление (изменение) данных в таблице;
- `DELETE` — удаление данных из таблицы.

Для добавления новых данных используется оператор **INSERT**, имеющий следующий синтаксис:

```
INSERT INTO table_name [(column_list)]
VALUES (data_value_list);
```

Здесь параметр `table_name` (Имя таблицы) представляет имя таблицы базы данных.

Параметр `column_list` (Список столбцов) представляет собой список столбцов, состоящий из имен одного или нескольких столбцов, разделенных запятыми и не является обязательным. Если он опущен, то предполагается использование списка из имен всех столбцов таблицы.

Параметр `data_value_list` (Список значений данных) должен следующим образом соответствовать параметру `column_list`:

- количество элементов в обоих списках должно быть одинаковым;
- должно существовать прямое соответствие между позицией одного и того же элемента в обоих списках;
- типы данных элементов списка `data_value_list` должны быть совместимы с типом данных соответствующих столбцов таблицы.

Оператор DELETE позволяет удалять строки данных из указанной таблицы. Этот оператор имеет следующий формат:

```
DELETE FROM table_name  
[WHERE search_condition];
```

Как и в случае оператора INSERT, параметр `table_name` представляет собой имя таблицы. Параметр `search_condition` является необязательным, если он опущен, из таблицы будут удалены все существующие в ней строки. Если предложение WHERE присутствует, из таблицы будут удалены только те строки, которые удовлетворяют условию отбора, заданному параметром `search_condition`.

Оператор UPDATE позволяет изменять содержимое уже существующих строк указанной таблицы. Этот оператор имеет следующий формат:

```
UPDATE table_name  
SET column_name1=data_value1, column_name2  
= data_value2  
[WHERE search_condition];
```

Параметр `table_name` представляет собой имя таблицы. В предложении SET указываются имена одного или более столбцов, данные в которых необходимо изменить. Предложение WHERE является необязательным. Если оно опущено, значения указанных столбцов будут изменены во всех строках таблицы. Если предложение WHERE присутствует, то будут обновлены только те строки, которые удовлетворяют условию поиска, заданному в параметре `search_condition`. Параметры `data_value` представляют новые значения соответствующих столбцов и должны быть совместимы с ними по типу данных.

## Запрос 1

Ввести новую строку в таблицу Товар.

### Инструкция

INSERT INTO Товар

VALUES (12, 'Грудинка', 'кг', 1000, 320.00, 10);

Код_товара	Товар	Единицы_измере-	Количество	Цена	Код_МОЛ
1	Колбаса копчё	кг	10000	145,00р.	1
2	Сосиски	кг	5000	120,00р.	4
3	Окорочка	кг	2000	50,00р.	1
4	Балык	кг	5000	235,00р.	3
5	Ветчина	кг	7000	348,00р.	2
6	Сардельки	кг	1500	140,00р.	5
7	Колбаса варён	кг	1500	154,00р.	10
8	Котлеты	кг	3000	230,00р.	8
9	Пельмени	кг	1000	170,00р.	7
10	Хинкали	кг	900	40,00р.	9
11	Фарш	кг	400	210,00р.	6
12	Грудинка	кг	1000	320,00р.	10
*	(№)		0	0,00р.	0

## Запрос 2

Удалить из таблицы Поставка строку с номером поставки 10

### Инструкция

DELETE \*

FROM Поставка

WHERE [№\_поставки]=10;

Код_поставщика	Код_товара	№_поставки
3000	1	1
3000	2	2
3001	3	3
3002	4	4
3004	1	5
3005	6	6
3006	10	7
3007	9	8
3008	7	9
*		(№)

## Модификация данных

### Запрос 3

Увеличить цену в таблице Товар до 210,00 у товара с кодом 11

#### Инструкция

```
UPDATE Товар SET Цена = 210
```

```
WHERE Код_товара=11;
```

Товар						
Код_товара	Товар	Единицы_измере	Количество	Цена	Код_МОЛ	Ц
1	Колбаса копчё	кг	10000	145,00р.	1	
2	Сосиски	кг	5000	120,00р.	4	
3	Окорочка	кг	2000	50,00р.	1	
4	Балык	кг	5000	235,00р.	3	
5	Ветчина	кг	7000	348,00р.	2	
6	Сардельки	кг	1500	140,00р.	5	
7	Колбаса варён	кг	1500	154,00р.	10	
8	Котлеты	кг	3000	230,00р.	8	
9	Пельмени	кг	1000	170,00р.	7	
10	Хинкали	кг	900	40,00р.	9	
11	Фарш	кг	400	210,00р.	6	
12	Грудинка	кг	1000	320,00р.	10	
*(No)			0	0,00р.	0	

### Запрос 4

МОЛ Иванов уволился присвоить его код новому сотруднику

#### Инструкция

```
UPDATE МОЛ SET МОЛ = 'Сидоров Иван Павлович',  
№_паспорта = 1501-6578556, Адрес = 'г. Брянск, ул.  
Куйбышева, д. 38, кв. 12', Телефон = 678947, Да-  
та_рождения = "30.01.1977"
```

```
WHERE Код_МОЛ = 7;
```

МОЛ						
Код_Мп	МОЛ	№_паспорт:	Адрес	Телефон	Дата_рож	Ще
1	Андрееко Евгений Петрови	1599-7862576	г. Брянск, ул. Красноармейская, д. 681520	728322	03.05.1965	
2	Плюшкин Валерий Владими	1599-7553228	г. Брянск, Пр. Ленина, д. 2, кв. 5	328523	12.08.1970	
3	Баранкин Евгений Алексан	1599-6523411	г. Брянск, ул. Красноармейская, д. 328523	1654336	25.01.1947	
4	Гончаров Андрей Алексан	1599-8832327	г. Брянск, ул. Камозина, д. 53, кв. 1	765347	03.08.1967	
5	Алексеев Михаил Валенти	1599-4596595	г. Брянск, ул. Куйбышева, д. 21, кв. 1	251366	28.11.1958	
6	Петров Евгений Васильеви	1598-4913475	г. Брянск, ул. 1 Мая, д. 3	678947	23.05.1971	
7	Сидоров Иван Павлович	-6577055	г. Брянск, ул. Куйбышева, д. 38, кв. 12	678947	30.01.1977	
8	Сергееко Владимир Серге	1588-1545176	г. Брянск, Пр. Ленина, д. 5, кв. 10	947472	15.01.1975	
9	Миронов Владимир Влади	1592-1546765	г. Брянск, ул. Мира, д. 3, кв. 13	678947	28.02.1979	
10	Сидоров Николай Павлови	1589-5874368	г. Брянск, ул. Куйбышева, д. 38, кв. 12	678947	01.08.1973	

## Литература

1. Багдуева А.В. SQL. Язык структурированных запросов. Улан-Удэ: Бурятский государственный университет, 2017. 122 с.
2. Ёсу М.Т., Вальдуриес П. Принципы организации распределенных баз данных / пер. с англ. А.А. Слинкина. М.: ДМК Пресс, 2021. 678 с.
3. Кондрашов, Ю.Н., Язык SQL. Сборник ситуационных задач по дисциплине «Базы данных»: учебно-практическое пособие. М.: Русайнс, 2022. 125 с.
4. Лысенкова С.Н. Основы проектирования баз данных: учебно-методическое пособие для студентов направления подготовки 09.03.03 Прикладная информатика. Брянск: Изд-во Брянский ГАУ, 2019. 66 с.
5. Остроух А.В., Помазанов А.В. Теория проектирования распределенных информационных систем: монография. СПб.: Лань, 2022. 96 с.
6. Токмаков Г.П. Базы данных: Модели и структуры данных, язык SQL, программирование баз данных: учебное пособие. Ульяновск: УлГТУ, 2021. 362 с.
7. Файли К. SQL. Руководство для использования с любыми SQL СУБД. М.: ДМК Пресс, 2012. 454 с.

Учебное издание

Лысенкова Светлана Николаевна

«Распределенные базы данных».  
ОСНОВЫ ЯЗЫКА SQL

методическое пособие  
для студентов направления подготовки  
09.04.03 Прикладная информатика

Редактор Осипова Е.Н.

---

Подписано к печати 01.11.2022 г. Формат 60x84<sup>1</sup>/<sub>16</sub>.  
Бумага офсетная. Усл. п. л. 2,09. Тираж 25 экз. Изд. № 7398.

---

Издательство Брянского государственного аграрного университета  
243365 Брянская обл., Выгоничский район, с. Кокино, Брянский ГАУ