

Базы данных и информационные системы (ИНФ)

ОСНОВЫ **Structured Query Language (SQL)**. Язык DQL

Простые запросы (запросы к одной таблице)

Лекция 5

План занятия

Общая информация (введение)

- 1. Стандарты**
 - 2. Типы команд SQL**
 - 3. Запись SQL - операторов**
 - 4. Запросы на выборку данных (DML/DQL)**
 - 5. Общий формат оператора SELECT**
 - 6. Выборка всех строк**
 - 7. Выборка строк, удовлетворяющих условию (WHERE)**
 - сравнение;
 - диапазон;
 - принадлежность множеству;
 - соответствие шаблону;
 - проверка на неопределенное значение;
 - 8. Сортировка строк (ORDER BY)**
 - 9. Получение итоговых значений (агрегатные функции)**
 - 10. Группирование результатов (GROUP BY)**
- Заключение**

Стандарты SQL

Типы команд SQL

Согласно общепринятому стилю программирования, операторы (и другие зарезервированные слова) в SQL всегда следует писать прописными буквами.

Операторы SQL делятся на:

- операторы определения данных (*Data Definition Language*, DDL)
 - CREATE создает объект БД (саму базу, таблицу, представление, пользователя и т. д.)
 - ALTER изменяет объект
 - DROP удаляет объект
- операторы манипуляции данными (*Data Manipulation Language*, DML)
 - SELECT считывает данные, удовлетворяющие заданным условиям (DQL)
 - INSERT добавляет новые данные
 - UPDATE изменяет существующие данные
 - DELETE удаляет данные
- операторы определения доступа к данным (*Data Control Language*, DCL)
 - GRANT предоставляет пользователю (группе) разрешения на определенные операции с объектом
 - REVOKE отзывает ранее выданные разрешения
 - DENY задает запрет, имеющий приоритет над разрешением
- операторы управления транзакциями (*Transaction Control Language*, TCL)
 - COMMIT применяет транзакцию.
 - ROLLBACK откатывает все изменения, сделанные в контексте текущей транзакции.
 - SAVEPOINT делит транзакцию на более мелкие участки.

Запись SQL - операторов

- Оператор SQL состоит из **зарезервированных слов** и **пользовательских названий**.
- **Зарезервированные слова** являются постоянной частью языка SQL и имеют фиксированное значение. Их следует записывать в точности так, как это установлено, нельзя разбивать на части для переноса с одной строки на другую.
- **Пользовательские названия** - слова, определяемые пользователем, задаются пользователем в соответствии с синтаксическими правилами и представляют собой *идентификаторы* или имена различных объектов базы данных.
- **Синтаксические правила для пользовательских названий:**
Пользовательские названия представляют собой набор символов, который:
 - включает строчные и прописные буквы латинского алфавита (A-Z, a-z), цифры (0-9) и символ подчеркивания (_),
 - название может иметь длину до 128 символов,
 - начинаться с буквы,
 - не может содержать пробелы.
- Большинство компонентов языка не чувствительны к регистру (исключение - символьная информация в БД).
- Поскольку у языка SQL свободный формат, отдельные SQL-операторы и их последовательности будут иметь более читаемый вид при использовании отступов и выравнивания.

Запись SQL - операторов

- Язык, в терминах которого дается описание языка SQL, называется *метаязыком*. Синтаксические определения обычно задают с помощью специальной металингвистической символики, называемой *Бэкуса-Наура формулами* (БНФ).
- Прописные буквы используются для записи *зарезервированных слов*.
- Строчные буквы употребляются для записи слов, определяемых пользователем.
- Применяемые в нотации *БНФ* символы и их обозначения показаны в *таблице*:

Символ	Обозначение
::=	Равно по определению
	Необходимость выбора одного из нескольких приведенных значений
{...}	Обязательный выбор некоторой конструкции из списка
[...]	Необязательный выбор некоторой конструкции из списка
[...n]	Необязательная возможность повторения конструкции от нуля до нескольких раз

Запросы на выборку данных (DML/DQL).

Оператор SELECT

- Предназначен для выборки и отображении данных одной или более таблиц базы данных (ключевое/ зарезервированное слово).
- В пределах одной команды SELECT выполняются действия, эквивалентные операторам реляционной алгебры:
 - выборки;
 - проекции;
 - декартового произведения;
 - соединения;
- Оператор SELECT является чаще всего используемой командой языка SQL
- Общий формат оператора SELECT имеет следующий вид:

```
SELECT [ALL | DISTINCT ] {*[имя_столбца или выражение [AS новое_имя]]} [,...n]  
FROM имя_таблицы [[AS] псевдоним] [,...n]  
[WHERE <критерии выбора кортежей>]  
[GROUP BY имя_столбца [,...n]]  
[HAVING <критерии выбора групп>]  
[ORDER BY имя_столбца [,...n]]
```

Запросы на выборку данных (DML/DQL).

Оператор SELECT

Последовательность обработки элементов оператора SELECT:

1. **FROM** - определяются имена используемой таблицы или нескольких таблиц;
 2. **WHERE** - выполняется фильтрация кортежей (строк) объекта в соответствии с заданными условиями;
 3. **GROUP BY** - образуются группы строк, имеющих одно и то же значение в указанном столбце;
 4. **HAVING** - фильтруются группы строк объекта в соответствии с указанным условием;
- **SELECT** - устанавливается, какие столбцы должны присутствовать в выходных данных
 - **ORDER BY** - определяется упорядоченность результатов выполнения оператора.

Замечания!

1. Порядок конструкций в операторе SELECT *не может* быть изменен.
2. ? Только две конструкции оператора - SELECT и FROM - являются обязательными, все остальные конструкции могут быть опущены.
3. Операция выборки с помощью оператора SELECT является *замкнутой*, в том смысле, что результат запроса к таблице также представляет собой таблицу

Исходная схема данных БД «Торговля»

Клиент (КодКлиента, Фамилия, Имя, Отчество, Фирма, ГородКлиента, Телефон)

Товар (КодТовара, Название, Тип, Сорт, Цена, Остаток, ГородТовара)

Сделка (КодСделки, Кол_во, Дата, КодТовара, КодКлиента)

Клиент						
КодКлиента	Фамилия	Имя	Отчество	Фирма	ГородКлиента	Телефон
1	Иванов	Иван	Иванович	ООО Буд	Харьков	050-789 45 56
2	Петров	Петр	Петрович	ООО Ух	Киев	067- 786 34-87
3	Сидоров	Сидор	Сидорович	ООО Буд	Харьков	050-711 65 88
4	Климов	Кузьма	Васильевич	ООО Буд	Киев	098-777 45 22
5	Абрамов	Алексей	Федорович	ООО Ух	Харьков	050-232 11 45
6	Семенов	Василий	Степанович	ООО Уют	Харьков	098-34522 65
7	Бобырь	Алексей	Иванович	ООО Уют	Киев	050-555 22 44



Товар						
Код Товара	Название	Тип	Сорт	Цена	Остаток	ГородТовара
1	Стул	мебель	высший	400,00р.	10	Харьков
2	Стол	мебель	первый	200,00р.	20	Киев
3	Стул	мебель	высший	400,00р.	1	Киев
4	Диван	мебель	второй	4 000,00р.	3	Харьков
5	Диван	мебель	высший	8 000,00р.	1	Киев
6	Стол	мебель	второй	400,00р.	2	Москва
7	Рамка для фото	интерьер	высший	150,00р.	10	Москва
8	Подсвечник	интерьер	первый	40,00р.	10	Харьков
9	Шкаф	мебель	высший	10 000,00р.	2	Киев

Сделка				
КодСделки	КодТовара	КодКлиента	Кол_во	Дата
1	1	1	10	11.10.2010
2	2	1	2	13.10.2009
3	1	2	1	13.10.2009
4	2	2	1	14.10.2009
5	1	1	2	15.10.2009
6	3	4	5	15.10.2009
7	4	3	1	15.10.2009
8	5	5	2	16.10.2009

Запросы на выборку данных (DML/DQL). Оператор SELECT

Выборка всех строк

Запрос1. Составить список сведений о всех клиентах.

```
SELECT КодКлиента, Фамилия, Имя, Отчество, Фирма, ГородКлиента, Телефон FROM Клиент;
```

Упрощенный вариант записи

```
SELECT * FROM Клиент;
```

Выборка конкретных столбцов

Запрос 2. Составить список всех фирм.

```
SELECT Фирма FROM Клиент;
```

```
SELECT Клиент.Фирма FROM Клиент;
```

(результат с повторами)

Предикат ALL задает включение в выходной набор всех дубликатов

(значение действует по умолчанию)

```
SELECT ALL Фирма FROM Клиент;
```

Запрос2
Фирма
ООО Буд
ООО Ух
ООО Буд
ООО Буд
ООО Ух
ООО Уют
ООО Уют

Запросы на выборку данных (DML/DQL).

Оператор SELECT

Использование DISTINCT

- Ключевое слово DISTINCT позволяет отбросить блоки данных, содержащие *дублирующие записи* в выбранных полях.
- Причиной ограничения в применении DISTINCT является то обстоятельство, что его использование может резко замедлить выполнение запросов

Запрос 3. Составить список всех фирм (без повторений).

```
SELECT DISTINCT Фирма FROM Клиент;
```

Запрос 4. Составить список всех фирм и их месторасположения (без повторений).

```
SELECT DISTINCT Фирма, ГородКлиента FROM Клиент;
```

Запрос4	
Фирма	ГородКлиента
ООО Буд	Киев
ООО Буд	Харьков
ООО Ух	Киев
ООО Ух	Харьков
ООО Уют	Киев
ООО Уют	Харьков

Запросы на выборку данных (DML/DQL).

Оператор SELECT

Замена имени поля

Запрос 5.

SELECT КодКлиента as Номер, Фамилия FROM Клиент;

Вставка литералов

Запрос 6. (", '')

SELECT КодКлиента, 'проживает в', ГородКлиента
FROM Клиент;

Запрос 7.

SELECT КодКлиента AS Клиент, 'проживает в ' AS проживает, ГородКлиента AS Город FROM
Клиент;

Номер	Фамилия
1	Иванов
2	Петров
3	Сидоров
4	Климов
5	Абрамов
6	Семенов
7	Бобырь

КодКлиента	Ехрг1001	ГородКлиента
1	проживает в	Харьков
2	проживает в	Киев
3	проживает в	Харьков
4	проживает в	Киев
5	проживает в	Харьков
6	проживает в	Харьков
7	проживает в	Киев

Клиент	проживает	Город
1	проживает в	Харьков
2	проживает в	Киев
3	проживает в	Харьков
4	проживает в	Киев
5	проживает в	Харьков
6	проживает в	Харьков
7	проживает в	Киев

Запросы на выборку данных (DML/DQL).

Оператор SELECT

Вычисляемые поля

- **Вычисляемое поле:** некоторое выражение языка SQL, указанное в списке SELECT
- В этих выражениях применяются:
 - арифметические операции сложения, вычитания, умножения и деления;
 - встроенные функции языка SQL.
- В этих выражениях можно указать имя любого столбца (поля) таблицы или запроса, которые указаны в списке предложения FROM
- Стандарты SQL позволяют явным образом задавать имена столбцов результирующей таблицы, для чего применяется фраза AS.

Запрос 8. Уменьшить цену товаров в половину.

Вывести код товара, старую и измененную цену.

```
SELECT КодТовара, Цена, Цена*0.5 AS Уценка  
FROM Товар;
```

КодТовара	Цена	Уценка
1	400,00р.	200
2	200,00р.	100
3	400,00р.	200
4	4 000,00р.	2000
5	8 000,00р.	4000
6	400,00р.	200
7	150,00р.	75
8	40,00р.	20
9	10 000,00р.	5000

Запросы на выборку данных (DML/DQL). Оператор SELECT

Вычисляемые поля

Запрос9. Получить список товаров с указанием года и месяца продажи.

```
SELECT КодТовара, Year(Дата) AS Год,  
Month(Дата) AS Месяц FROM Сделка
```

Функции `Year` и `Month` выделяют год и месяц из даты соответственно.

Запрос9		
КодТовара	Год	Месяц
1	2010	10
2	2009	10
1	2009	10
2	2009	10
1	2009	10
3	2009	10
4	2009	10
5	2009	10
6	2009	10
8	2009	10
5	2009	10

Конкатенация

Запрос10. Получить список фирм с указанием фамилии и инициалов клиентов

```
SELECT Фирма, Фамилия+' '+Left(Имя,1)+' '+Left(Отчество,1)+' ' AS ФИО FROM Клиент;
```

& - оператор конкатенации строк в Access

Функция `Left` вырезает в текстовой переменной один символ слева.

```
SELECT Фирма,  
Фамилия & ' ' & Left(Имя,1) & ' ' & Left(Отчество,1) & ' ' AS ФИО  
FROM Клиент;
```

Запрос_10	
Фирма	ФИО
ООО Буд	Иванов И.И.
ООО Ух	Петров П.П.
ООО Буд	Сидоров С.С.
ООО Буд	Климов К.В.
ООО Ух	Абрамов А.Ф.
ООО Уют	Семенов В.С.
ООО Уют	Бобырь А.И.

Запросы на выборку данных (DML/DQL). Оператор SELECT

Выборка строк (конструкция WHERE)

- За ключевым словом WHERE следует перечень *условий поиска*, определяющих те строки (кортежи), которые должны быть выбраны при выполнении запроса.
- Существует пять основных типов *условий поиска* (или предикатов):

Сравнение - сравниваются результаты вычисления одного выражения с результатами вычисления другого.

Диапазон - проверяется, попадает ли результат вычисления выражения в заданный *диапазон* значений.

Принадлежность множеству - проверяется, принадлежит ли результат вычислений выражения заданному множеству значений.

Соответствие шаблону - проверяется, отвечает ли некоторое строковое значение заданному шаблону.

Значение NULL - проверяется, содержит ли данный столбец NULL (неопределенное значение).

Запросы на выборку данных (DML/DQL). Оператор SELECT

Выборка строк (конструкция WHERE)

Сравнение

Операторы *сравнения*:

- = – равенство;
- < – меньше;
- > – больше;
- <= – меньше или равно;
- >= – больше или равно;
- <> – не равно (!=).

Запрос 11. Показать все операции отпуска товаров (сделки) объемом больше 20.

```
SELECT * FROM Сделка WHERE Количество>20
```

Для сложных предикатов используются логических операторов AND, OR, NOT, скобки.

Вычисление выражения 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.

Запросы на выборку данных (DML/DQL). Оператор SELECT

Выборка строк (конструкция WHERE)

Сравнение

Запрос 12. Вывести список товаров, цена которых больше или равна 100 и меньше или равна 150.

Запрос_12	
Название	Цена
Рамка для фото	150,00р.

Запрос 13. Вывести список клиентов из Харькова или из Москвы.

```
SELECT Фамилия, ГородКлиента
```

```
FROM Клиент
```

```
WHERE ГородКлиента='Харьков' OR ГородКлиента='Москва';
```

```
SELECT Фамилия, ГородКлиента
```

```
FROM Клиент
```

```
WHERE ГородКлиента="Харьков" OR ГородКлиента="Москва";
```

Запрос_13	
Фамилия	ГородКлиента
Иванов	Харьков
Сидоров	Харьков
Абрамов	Харьков
Семенов	Харьков

Запросы на выборку данных (DML/DQL). Оператор SELECT

Выборка строк (конструкция WHERE)

Диапазон (BETWEEN / NOT BETWEEN)

BETWEEN - поиск значений внутри некоторого интервала с включением крайних значений

Запрос 14. Вывести список товаров, цена которых лежит в диапазоне от 100 до 150 (запрос эквивалентен **запросу 12**).

```
SELECT Название, Цена
```

```
FROM Товар
```

```
WHERE Цена Between 100 And 150;
```

NOT BETWEEN - поиск значений вне границ заданного *диапазона*.

Запрос 15. Вывести список товаров, цена которых не лежит в диапазоне от 100 до 150.

```
SELECT Название, Цена
```

```
FROM Товар
```

```
WHERE Цена NOT Between 100 And 150;
```

или (что эквивалентно)

```
SELECT Название, Цена
```

```
FROM Товар WHERE (Цена<100) OR (Цена>150);
```

Запрос_15	
Название	Цена
Стул	400,00р.
Стол	200,00р.
Стул	400,00р.
Диван	4 000,00р.
Диван	8 000,00р.
Стол	400,00р.
Подсвечный	40,00р.
Шкаф	10 000,00р.

Запросы на выборку данных (DML/DQL). Оператор SELECT

Выборка строк (конструкция WHERE)

Принадлежность множеству (IN/NOT IN)

Оператор **IN** проверяет соответствует ли результат вычисления выражения одному из значений в предоставленном списке.

При помощи оператора **IN** может быть достигнут тот же результат, что и в случае применения оператора **OR**.

Запрос 16. Вывести список клиентов из Харькова или из Москвы (запрос эквивалентен **запросу 13**)

```
SELECT Фамилия, ГородКлиента  
FROM Клиент  
WHERE ГородКлиента IN ('Харьков','Москва');
```

(что эквивалентно)

```
SELECT Фамилия, ГородКлиента  
FROM Клиент  
WHERE ГородКлиента='Харьков' OR ГородКлиента='Москва');
```

Запросы на выборку данных (DML/DQL).

Оператор SELECT

Выборка строк (конструкция WHERE)

Принадлежность множеству (IN/NOT IN)

NOT IN позволяет отобрать любые значения, кроме тех, которые указаны в представленном списке.

Запрос 17. Вывести список клиентов, проживающих не в Харькове и не в Москве.

```
SELECT Фамилия, ГородКлиента FROM Клиент  
WHERE ГородКлиента NOT IN ('Харьков','Москва');
```

NOT можно использовать после where

```
SELECT Фамилия, ГородКлиента FROM Клиент  
WHERE NOT ГородКлиента IN ('Харьков','Москва');
```

Запрос_17	
Фамилия	ГородКлиента
Петров	Киев
Климов	Киев
Бобырь	Киев

Запросы на выборку данных (DML/DQL).

Оператор SELECT

Выборка строк (конструкция WHERE)

Соответствие шаблону (LIKE/NOT LIKE)

Оператор **LIKE** выполняет *сравнение* выражения с заданным шаблоном, в котором допускается использование символов-заменителей:

Стандарт ANSI

% - любое количество произвольных символов;

_ - заменяет один символ строки.

Платформа **MS SQL Server** поддерживает дополнительно:

[список] – вместо символа строки будет подставлен один из возможных символов, указанный в этих ограничителях;

[^ список] – вместо соответствующего символа строки будут подставлены все символы, кроме указанных в ограничителях.

Символы – заменителя для **СУБД Access**

? - один символ;

* - любое количество символов;

- любая цифра (0-9);

[список] - любой символ из списка;

[!список] - любой символ не из списка.

Запросы на выборку данных (DML/DQL). Оператор SELECT

Выборка строк (конструкция WHERE)

Соответствие шаблону (LIKE\NOT LIKE)

Если искомая строка содержит символ – заменитель, то следует задать управляющий символ в предложении ESCAPE.

Пример 1. Необходимо найти строки, содержащие символ «_».

Шаблон '%_%' – вернет все записи;

Шаблон '%#_%' ESCAPE '#' – вернет строки, содержащие символ «_»

Пример 2. Необходимо найти строки, содержащие значение “25%”.

Шаблон '25|%' ESCAPE '|' – вернет все строки, содержащие значение “25%”.

Запрос 18. Найти клиентов, у которых в номере телефона вторая цифра – 5.

MS SQL Server:

```
SELECT Фамилия, Телефон
```

```
FROM Клиент
```

```
WHERE Телефон Like '_5%';
```

Access: SELECT Фамилия, Телефон

```
FROM Клиент
```

```
WHERE Телефон Like '?5*';
```

Запрос_17	
Фамилия	Телефон
Иванов	050-789 45 56
Сидоров	050-711 65 88
Абрамов	050-232 11 45
Бобырь	050-555 22 44

Запросы на выборку данных (DML/DQL). Оператор SELECT

Выборка строк (конструкция WHERE)

Соответствие шаблону (LIKE/NOT LIKE)

Запрос 19. Найти клиентов, у которых в номере телефона вторая цифра – 5 или 6.

MS SQL Server:

```
SELECT Фамилия, Телефон
```

```
FROM Клиент
```

```
WHERE Телефон Like '_[56]%';
```

Access: SELECT Фамилия, Телефон

```
FROM Клиент
```

```
WHERE Телефон Like '?[56]*';
```

Запрос19	
Фамилия	Телефон
Иванов	050-789 45 56
Петров	067- 786 34-87
Сидоров	050-711 65 88
Абрамов	050-232 11 45
Бобырь	050-555 22 44

Запрос 20. Найти клиентов, у которых в номере телефона вторая цифра не 5 и не 6.

MS SQL Server:

```
SELECT Фамилия, Телефон
```

```
FROM Клиент
```

```
WHERE Телефон Like '_[^56]%';
```

Access: SELECT Фамилия, Телефон

```
FROM Клиент
```

```
WHERE Телефон Like '?[!56]*';
```

Запросы на выборку данных (DML/DQL). Оператор SELECT

Выборка строк (конструкция WHERE)

Соответствие шаблону (LIKE\NOT LIKE)

Запрос 21. Найти клиентов, у которых в номере телефона вторая цифра 5 или 7,8,9.

MS SQL Server:

```
SELECT Фамилия, Телефон
```

```
FROM Клиент
```

```
WHERE Телефон Like '_[57-9]%';
```

Access: SELECT Фамилия, Телефон

```
FROM Клиент
```

```
WHERE Телефон Like '?[57-9]*';
```

Запрос_20	
Фамилия	Телефон
Иванов	050-789 45 56
Сидоров	050-711 65 88
Климов	098-777 45 22
Абрамов	050-232 11 45
Семенов	098-34522 65
Бобырь	050-555 22 44

Запросы на выборку данных (DML/DQL). Оператор SELECT

Выборка строк (конструкция WHERE)

Соответствие шаблону (LIKE/NOT LIKE)

Запрос 22. Найти клиентов, у которых в фамилии встречается слог “ов”.

MS SQL Server:

```
SELECT Фамилия  
FROM Клиент  
WHERE Фамилия Like '%ов%';
```

Access: SELECT Фамилия

```
FROM Клиент  
WHERE Фамилия Like '*ов*';
```

Запрос_22
Фамилия
Иванов
Петров
Сидоров
Климов
Абрамов
Семенов

Запрос 23. Найти клиентов, у которых фамилия заканчивается на слог “ов”, но не “мов”.

MS SQL Server:

Access:

Замечание!

При выполнении сравнения с помощью **like** значимыми являются все символы, включая начальные и конечные пробелы.

Запросы на выборку данных (DML/DQL). Оператор SELECT

Выборка строк (конструкция WHERE)

Значение NULL (IS NULL\ IS NOT NULL)

Оператор **IS NULL** используется для сравнения текущего значения с неопределенным значением NULL.

Запрос 24. Найти сотрудников, у которых нет телефонов.

```
SELECT Фамилия, Телефон  
FROM Клиент  
WHERE Телефон IS NULL;
```

Замечание! Не правильно:

```
WHERE Телефон = ' '      - ' ' – не является NULL - значением;  
WHERE Телефон = NULL    - вернет Unknown;  
(В SQL Server 2000   WHERE Телефон = NULL поддерживается);
```

IS NOT NULL используется для проверки присутствия значения в поле.

Запрос 25. Найти сотрудников, у которых есть телефон.

```
SELECT Фамилия, Телефон  
FROM Клиент  
WHERE Телефон IS NOT NULL;
```

Запросы на выборку данных (DML/DQL). Оператор SELECT

Сортировка строк (предложение ORDER BY)

ORDER BY – позволяет определить порядок возвращения записей

● Ключевые слова:

ASC – сортировка по возрастанию (по умолчанию);

DESC – сортировка по убыванию,

● Ключевые слова **ASC**, **DESC** относятся к одному столбцу.

● **NULL** – значения при сортировке собираются вместе (Access и SQL Server - вверху).

● Столбцы (ключи сортировки), определяющие порядок записей, могут указываться с помощью:

● **псевдонимов столбцов** (рекомендуется, Access не поддерживает);

● **имен столбцов**;

● **целый чисел, определяющих порядок столбцов в списке select** (не рекомендуется);

Запрос 26. Вывести фирмы, имена и фамилии клиентов,
список отсортировать по фирмам по возрастанию (от А до Я)

```
SELECT Фирма as Откуда, Имя, Фамилия
```

```
FROM Клиент
```

```
ORDER BY Откуда
```

```
SELECT Фирма, Имя, Фамилия
```

```
FROM Клиент
```

```
ORDER BY Фирма
```

```
SELECT Фирма, Имя, Фамилия
```

```
FROM Клиент
```

```
ORDER BY 1
```

Откуда	Имя	Фамилия
ООО Буд	Сидор	Сидоров
ООО Буд	Кузьма	Климов
ООО Буд	Иван	Иванов
ООО Ух	Петр	Петров
ООО Ух	Алексей	Абрамов
ООО Уют	Василий	Семенов
ООО Уют	Алексей	Бобырь

Запросы на выборку данных (DML/DQL). Оператор SELECT

Сортировка строк (предложение ORDER BY)

ORDER BY может проводить сортировку по нескольким ключам сортировки от главного ключа к второстепенному:

Шаг1. Сортировка по **первому** столбцу.

Шаг2. Строки с одинаковыми значениями **первого** столбца сортируются по **второму** столбцу.

Шаг3. Строки с одинаковыми значениями **первого** и **второго** столбца сортируются по **третьему** столбцу, и т.д.

Замечание! Многоключевая сортировка имеет смысл, если значения старшего ключа не уникальны!

Запрос 27. Вывести фирмы, имена и фамилии клиентов,

список отсортировать по фирмам по убыванию,

по фамилиям по возрастанию

```
SELECT Фирма, Фамилия
```

```
FROM Клиент
```

```
ORDER BY Фирма DESC, Фамилия
```

Запрос 28. Вывести номера и фамилии клиентов,

список отсортировать по номерам по убыванию,

по фамилиям по возрастанию

```
SELECT КодКлиента, Фамилия
```

```
FROM Клиент
```

```
ORDER BY КодКлиента DESC, Фамилия
```

Запрос27	
Фирма	Фамилия
ООО Уют	Бобырь
ООО Уют	Семенов
ООО Ух	Абрамов
ООО Ух	Петров
ООО Буд	Иванов
ООО Буд	Климов
ООО Буд	Сидоров

Запрос28	
КодКлиента	Фамилия
7	Бобырь
6	Семенов
5	Абрамов
4	Климов
3	Сидоров
2	Петров
1	Иванов

Запросы на выборку данных (DML/DQL). Оператор SELECT

Самостоятельная работа

Запросы на выборку данных (DML/DQL). Оператор SELECT

Получение итоговых значений (агрегатные функции)

Функция	Описание
COUNT ([ALL DISTINCT] имя поля или выражение)	вычисляет количество значений в указанном поле
COUNT (*)	возвращает количество записей в наборе
MAX (имя поля или выражение)	возвращает максимальное значение из множества значений в поле
MIN (имя поля или выражение)	возвращает минимальное значение из множества значений в поле
AVG ([ALL DISTINCT] имя поля или выражение)	вычисляет среднее арифметическое значение в поле
SUM ([ALL DISTINCT] имя поля или выражение)	вычисляет сумму множества значений в поле

Общее описание:

- Оперируют со значением в одном поле или с выражением
- Возвращают единственное значение
- Функции SUM и AVG могут использоваться только в случае числовых полей
- Все функции (кроме COUNT(*)) не учитывают NULL-значения
- DISTINCT применяется для исключения дублирующих значений (**в Access в агрегатных функциях не применяется**)

Запросы на выборку данных (DML/DQL). Оператор SELECT

Получение итоговых значений (агрегатные функции)

Правила выполнения агрегатных функций:

- Аргументом агрегатной функции не может быть агрегатная функция
- Если при выполнении функции SUM происходит переполнение типа, возникает ошибка
- Не используется в строке **WHERE**

Запрос 29. Определить дату первой сделки

```
SELECT MIN(Дата)  
FROM Сделка;
```

Запрос_29
Expr1000
13.10.2009

Запрос 30. Определить количество сделок.

Вариант1

```
SELECT COUNT(*) AS Количество_сделок  
FROM Сделка;
```

Запрос_30
Количество_сделок
11

Вариант2

```
SELECT COUNT(КодСделки) AS Количество_сделок  
FROM Сделка;
```

Запросы на выборку данных (DML/DQL). Оператор SELECT

Получение итоговых значений (агрегатные функции)

Запрос 31. Определить максимальное, минимальное, среднее количество проданного товара за одну операцию (сделку), а также разность между максимальным и минимальным значениями.

```
SELECT MAX(Кол_во), MIN(Кол_во), AVG(Кол_во), (MAX(Кол_во)-MIN(Кол_во))  
FROM Сделка;
```

Запрос31			
Expr1000	Expr1001	Expr1002	Expr1003
10	1	3,27272727272727	9

● Функции ABS(), ROUND(), INT()

Запрос 32. Подсчитать количество клиентов, которые хотя бы один раз покупали товар

```
SELECT COUNT (DISTINCT КодКлиента)  
FROM Сделка;
```

(в Access не применяется)

Запрос 33. Подсчитать количество проданных товаров в 2009 году.

Запросы на выборку данных (DML/DQL). Оператор SELECT

Группирование результатов (предложение GROUP BY)

GROUP BY – используется при необходимости подсчета агрегатных значений для каждой группы записей

Предложение **GROUP BY** содержит список полей, для которых необходимо создать группы с последующим вычислением агрегатных значений, в результате чего для каждой группы формируется одна строка

При наличии в операторе SELECT предложения **GROUP BY** :

- Агрегатные значения подсчитываются для каждой группы, которая задается полями в предложении **GROUP BY**
- Каждая комбинация элементов списка в предложении **SELECT** должна иметь единственное значение для всей группы;
- Все имена полей, приведенные в списке **SELECT**, должны присутствовать и в предложении **GROUP BY**, за исключением случаев, когда имя столбца используется в агрегатной функции. Обратное правило не является справедливым;
- При использовании совместно с **WHERE** предложение **GROUP BY** обрабатывается после **WHERE**, т.е. группированию подвергаются только те строки, которые удовлетворили условию поиска;
- При группировании **NULL**- значения рассматриваются как равные и при идентичных значениях в остальных группируемых полях помещаются в одну группу

Замечание!

- Если запрос не содержит **GROUP BY** агрегатные функции применяются ко всем строкам (т.е. все строки – это одна группа)

Запросы на выборку данных (DML/DQL). Оператор SELECT

Группирование результатов (предложение GROUP BY)

Запрос 34. Подсчитать сколько продано каждого товара. Вывести номер товара и подсчитанное количество.

```
SELECT КодТовара, SUM(Кол_во) AS Количество  
FROM Сделка  
GROUP BY КодТовара;
```

Запрос_34	
КодТовара	Количество
1	13
2	3
3	5
4	1
5	7
6	3
8	4

Запрос 35. Сколько каждый день продавалось товара

```
SELECT Дата, SUM(Кол_во) AS Количество  
FROM Сделка  
GROUP BY Дата;
```

Запрос_35	
Дата	Количество
13.10.2009	3
14.10.2009	1
15.10.2009	8
16.10.2009	5
17.10.2009	4
18.10.2009	5
11.10.2010	10

Запрос 36. Сколько сделок осуществлялось каждый день

Запрос_36	
Дата	КоличествоСделок
13.10.2009	2
14.10.2009	1
15.10.2009	3
16.10.2009	2
17.10.2009	1
18.10.2009	1
11.10.2010	1

Запросы на выборку данных (DML/DQL). Оператор SELECT

Группирование результатов (предложение GROUP BY)

Запрос 37. Сколько сделок осуществлялось каждый день с товаром 1. Результат отсортировать по количеству по убыванию

```
SELECT Дата, COUNT(КодСделки) AS КоличествоСделок
FROM Сделка
WHERE КодТовара=1
GROUP BY Дата
ORDER BY COUNT(КодСделки) DESC;
```

Запрос_37	
Дата	КоличествоСделок
11.10.2010	1
15.10.2009	1
13.10.2009	1

Запрос 38. Какое количество каждого вида товара купил каждый клиент. Вывести номер клиента, номер товара и количество.

```
SELECT КодКлиента, КодТовара, SUM(Кол_во) AS Количество_товара
FROM Сделка
GROUP BY КодКлиента, КодТовара;
```

Запрос_38		
КодКлиента	КодТовара	Количество
1	1	12
1	2	2
2	1	1
2	2	1
3	4	1
4	3	5
5	5	7
5	6	3
6	8	4

Запросы на выборку данных (DML/DQL). Оператор SELECT

Отбор групп (предложение HAVING)

HAVING – позволяет отобрать группы, для которых выполняются определенные условия

- Формат соответствует формату предложения **WHERE**

Запрос 39а. Подсчитать сколько с каждым клиентом было совершено операций (сделок)

```
SELECT КодКлиента, COUNT(*) AS Кол_во_сделок  
FROM Сделка  
GROUP BY КодКлиента
```

Запрос39а	
КодКлиента	Кол_во_сделок
1	3
2	2
3	1
4	1
5	3
6	1

Запрос 39б. Вывести номера клиентов, с которыми совершено 3 операции (сделки).

```
SELECT КодКлиента, COUNT(*) AS Кол_во_сделок  
FROM Сделка  
GROUP BY КодКлиента  
HAVING COUNT(Кол_во)=3;
```

Запрос39б	
КодКлиента	Кол_во_сделок
1	3
5	3

Запросы на выборку данных (DML/DQL). Оператор SELECT

Отбор групп (предложение HAVING)

Запрос 40а. Подсчитать сколько с каждым клиентом было совершено операций (сделок) и сколько каждый клиент купил всего товара в результате этих операций. Вывести код клиента, количество операций, количество товара.

```
SELECT КодКлиента, COUNT(Кол_во) AS Кол_во_сделок, SUM(Кол_во) AS Кол_во_товара  
FROM Сделка  
GROUP BY КодКлиента
```

Запрос40а		
КодКлиента	Кол_во_сделок	Кол_во_товара
1	3	14
2	2	2
3	1	1
4	1	5
5	3	10
6	1	4

Запрос 40б. Подсчитать количество операций (сделок) с клиентами, для которых общее количество купленного товара находится в интервале [3;5]. Вывести код клиента, количество операций

Запрос40б	
КодКлиента	Кол_во_сделок
4	1
6	1

Запрос 40с. В запрос 40б добавить условие (в 2009 году) и отсортировать по коду клиента по убыванию.

Запрос40с	
КодКлиента	Кол_во_сделок
6	1
4	1
1	2

Запросы на выборку данных (DML/DQL). Оператор SELECT

Отбор групп (предложение HAVING)

Замечания!

- По стандарту условия предложения **HAVING** могут содержать:
 - агрегатные функции;
 - поля, по которым проводилась группировка (однако рекомендуется данную фильтрацию проводить с помощью WHERE).
- Не могут содержать: псевдонимы полей.

Запрос 41. Вывести номера товаров, объем продаж которых до 14.10.2009 году превысил 5 штук, причем информация о товарах с кодом 3 и 6 не интересует

Запрос_41	
КодТовара	
	1
	5

Предпочтительно в HAVING использовать только агрегатные функции, все остальные условия писать в WHERE :

Запросы на выборку данных (DML/DQL). Оператор SELECT

Темпоральный тип данных (дата/время)

MS Access

● Тип: Дата/Время;

При сравнении с помощью

операторов =, >, <, >=, <=, <>...

BETWEEN, IN

дата записывается:

● #ММ/ДД/ГГГГ# ,

● #ММ/ДД/ГГ#,

● #месяц/ДД/ГГГГ#

И используются функции: **Year(), Month(), Day(), Hour(), Minute(), Second()**

При сравнении с помощью LIKE

дата записывается как строка символов следующего формата: 'ДД.
ММ.ГГГГ Ч:ММ:СС'

Например, 09.11.2012 9:59:07,
09.11.2012 10:01:57

Пример использования LIKE

WHERE Дата LIKE '*2009'

WHERE Дата LIKE '*03.2009'

WHERE Дата LIKE '02*2009'

- март 2009г.

- 2 числа, неизвестного месяца,

2009г

Запросы на выборку данных (DML/DQL). Оператор SELECT

Темпоральный тип данных (дата/время)

MS SQL Server

Типы:

datetime; smalldatetime; date; time; datetimeoffset; datetime2; timestamp;

При сравнении с помощью

операторов =, >, <, >=, <=, <>...

BETWEEN, IN

дата записывается: **‘ ММ.ДД.ГГГГ’** или **‘ГГГГ-ММ-ДД’**

Например, **‘2010-10-11 00:00:00.000’, ‘09.20.2010’, ‘2010-10-09’**

И используются функции: **Year(), Month(), Day(), Hour(), Minute(), Second(),**

datepart(datepart, дата) - возвращает указанную первым аргументом часть даты в числовом формате

Например, **datepart(month, Дата)**

При сравнении с помощью **LIKE** лучше использовать функцию **CONVERT(varchar,поле)**

Пример,

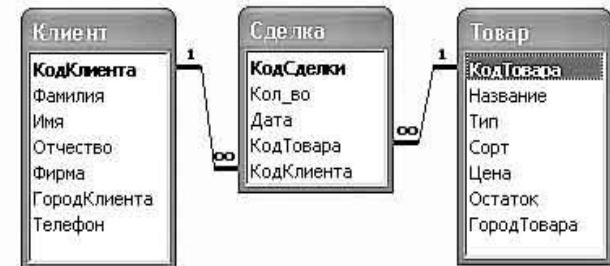
CONVERT (varchar,Дата)

Oct 13 2009 12:00AM

Пример использования **LIKE**

WHERE CONVERT (varchar,Дата) LIKE ‘*2009*’

Исходная схема данных БД «Торговля»



Клиент						
КодКлиента	Фамилия	Имя	Отчество	Фирма	ГородКлиента	Телефон
1	Иванов	Иван	Иванович	ООО Буд	Харьков	050-789 45 56
2	Петров	Петр	Петрович	ООО Ух	Киев	067- 786 34-87
3	Сидоров	Сидор	Сидорович	ООО Буд	Харьков	050-711 65 88
4	Климов	Кузьма	Васильевич	ООО Буд	Киев	098-777 45 22
5	Абрамов	Алексей	Федорович	ООО Ух	Харьков	050-232 11 45
6	Семенов	Василий	Степанович	ООО Уют	Харьков	098-34522 65
7	Бобырь	Алексей	Иванович	ООО Уют	Киев	050-555 22 44

Товар						
Код Товара	Название	Тип	Сорт	Цена	Остаток	ГородТовара
1	Стул	мебель	высший	400,00р.	10	Харьков
2	Стол	мебель	первый	200,00р.	20	Киев
3	Стул	мебель	высший	400,00р.	1	Киев
4	Диван	мебель	второй	4 000,00р.	3	Харьков
5	Диван	мебель	высший	8 000,00р.	1	Киев
6	Стол	мебель	второй	400,00р.	2	Москва
7	Рамка для фото	интерьер	высший	150,00р.	10	Москва
8	Подсвечник	интерьер	первый	40,00р.	10	Харьков
9	Шкаф	мебель	высший	10 000,00р.	2	Киев

Сделка				
КодСделки	КодТовара	КодКлиента	Кол_во	Дата
1	1	1	10	11.10.2010
2	2	1	2	13.10.2009
3	1	2	1	13.10.2009
4	2	2	1	14.10.2009
5	1	1	2	15.10.2009
6	3	4	5	15.10.2009
7	4	3	1	15.10.2009
8	5	5	2	16.10.2009
9	6	5	3	16.10.2009
10	8	6	4	17.10.2009
11	5	5	5	18.10.2009

Закрепление (подготовка в самостоятельной работе):

1. Вывести всю информацию о товарах, в названии которых есть слово «фото». Отсортировать результат по названию по возрастанию, по цене по убыванию
2. Вывести коды клиентов, которые совершали покупки в 2009г. (без повторов)
3. Сколько операций (сделок) было совершено с товарами с кодом 3,5,6.
4. Подсчитать в нашем ассортименте количество товаров каждого типа, товары типа «фурнитура» не интересуют (выводить тип и подсчитанное количество), отсортировать результат по типу по убыванию
5. Какова максимальная стоимость товаров каждого сорта, отсортировать результат по максимальной стоимости по убыванию.
6. Вывести названия фирм, в которых 2 и более клиентов, покупающих товары в нашей фирме
7. Подсчитать в нашем ассортименте количество товаров каждого типа, каждого сорта, стоимостью от 150 грн до 1000 грн включительно. В результирующем наборе выводите строки, где подсчитанное количество превышает 3. Отсортировать результат по типу по возрастанию, по сорту по убыванию.

УСПЕВАЕМОСТЬ

Группа (ИН_Гр, Деканат)

Студент(ИН_Студ, Фам, ИН_Группа, Адрес, Консультант)

Дисциплина (ИН_Дисц, Наименование)

Успеваемость(ИН_Студ, ИН_Дисц, Семестр, Оценка)

ГРУППА

ИН_Гр	Деканат
ПМ-08-1	ПММ
СА-08-1	ПММ
ИНФ-08-1	ПММ
ИНФ-08-2	ПММ
БИКС-08-1	КИУ
БИКС-08-2	КИУ

СТУДЕНТ

ИН_Студ	Фам	ИН_Гр	Адрес	Консультант
001	Иванов И.И.	ПМ-08-1	A1	004
002	Иванов И.И.	ПМ-08-1	A2	004
003	Петрова П.П.	СА-08-1	A3	002
004	Андреева А. А.	ИНФ-08-1	A4	Null

УСПЕВАЕМОСТЬ

ИН_Студ	ИН_Дисц	Семестр	Оценка
001	K1	6	4
001	K2	1	4
001	K2	2	3
002	K1	6	5
002	K3	3	4
004	K1	6	5
004	K2	1	5
004	K3	3	5
004	K2	2	5

КУРС

ИН_Дисц	Наименование
K1	Базы данных
K2	Высшая матем.
K3	Физика

