

Запросы.

Обработка данных

Выборка данных

- **SELECT** – определяет какие таблицы и столбцы источника данных необходимо использовать
- **FROM** – указывает таблицу
- **WHERE** – критерий отбора
- **ORDER BY** – порядок сортировки

Выборка данных

```
Select * from Goods
```

```
Select * from Goods where vid='конфеты'
```

```
select * from goods where price between 60 and 100
```

```
select nameG , price from goods where vid='Шоколад'
```

```
Select * from Goods where vid='конфеты' order by price
```

```
Select * from Goods where vid='конфеты' order by price desc
```

```
select top 3 * from goods order by price desc
```

Запросы с трафаретными символами и конструкцией LIKE

| Трафаретные символы | Значения |
|---------------------|--|
| - | Любой одиночный символ |
| % | Любая последовательность из нуля и более символов |
| [] | Любой одиночный символ в границах указанного диапазона ([a-c]) или набора ([abc]) |
| [^] | Любой одиночный символ не в границах указанного диапазона ([a-c]) или набора ([abc]) |

Запросы с трафаретными символами и конструкцией LIKE

- **LIKE ‘_ob’** – осуществляется поиск слова, состоящего из трех букв и заканчивающегося на ob , например Bob или Rob
- **LIKE ‘Иван%’** - осуществляется поиск фамилий, начинающихся с Иван, например Иванов или Иванова или Иванченко
- **LIKE ‘%ов’** - осуществляется поиск фамилий, заканчивающихся на ов, например Иванов или Петров
- **LIKE ‘%bi%’** - осуществляется поиск всех слов, содержащих bi, например binary или cabinet
- **LIKE ‘[CS]ery1’** - осуществляется поиск Chery1 или Shery1
- **LIKE ‘U[^n]’** - осуществляется поиск , начинающихся с буквы U и не содержащих букву n во второй позиции

Запросы с трафаретными символами и конструкцией LIKE

```
select * from client where Firma LIKE 'A%'
```

```
select * from client where Firma not LIKE 'A%'
```

```
select * from client where phone like '9[1-2]%'
```

```
select * from client where phone like '9__3%'
```

Задание псевдонимов для полей

```
select firma as [наименование фирмы] , city as [город]  
from client
```

```
select firma as [наименование фирмы] , SurnameC + ' ' + nameC as [имя клиента]  
from client
```



Объединение
строк

Стандартные функции

- Sum
- Count
- Min
- Max
- Avg
- Round(число, длина)

Стандартные функции

```
select month(datesell) as [месяц],  
year(datesell) as [год],  
num as [количество]  
from sell  
where year(datesell)=2005
```

```
select month(datesell) as [месяц],  
num as [количество]  
from sell  
where year(datesell)=2005  
order by month(datesell)|
```

Группировка данных

```
select Count(*) as [количество] ,  
       AVG (price) as [среднее значение] ,  
       max (price) as [максимальная цена] ,  
       min (price) as [минимальная цена]  
from goods
```

```
select month(datesell) as [месяц] ,  
       sum(num) as [количество]  
from sell  
where year(datesell)=2005  
group by month(datesell)
```

Группировка данных

```
select month(datesell) as месяц, sum(num) as количество,  
round(avg(num),0) as [среднее кол] from sell  
group by month(datesell)
```

```
select top 5 month(datesell) as месяц, sum(num) as количество,  
round(avg(num),0) as [среднее кол] from sell  
group by month(datesell)  
order by sum(num) desc|
```

```
select city from client
group by city
```

```
select * from goods
compute avg(store)
```

```
select * from goods
order by vid
compute avg(store) by vid
```

```
select * from goods
order by vid
compute avg(store), sum(store) | by vid
```

Добавление записей

- **INSERT INTO** имя_приемника
[(поле_1[, поле_2[, ...]])]
- **VALUES** (значение_1[, значение_2[, ...])

```
insert into goods  
(nameg, price, vid, store)  
values ('мечта', 120, 'конфеты', 50)
```

Удаление записей

- **DELETE** [таблица.*]
- **FROM** таблица
- **WHERE** УсловиеОтбора

```
delete from goods  
where vid='торты'
```

Обновление (корректировка) данных

- **UPDATE** таблица
- **SET** НовоеЗначение
- **WHERE** УсловиеОтбора

```
select * from goods
```

```
update goods
```

```
set price=price*10
```

```
where vid='конфеты'
```

```
select * from goods|
```

Операция соединения по двум отношениям (таблицам)

Соединение - это процесс, когда две или более таблицы объединяются в одну.

```
FROM имя_таблицы_1 {INNER | LEFT |  
RIGHT} JOIN имя_таблицы_2  
ON условие_соединения
```

Внутреннее объединение (INNER JOINT)

```
Select sell.datesell, goods.nameg, sell.num  
from sell  
inner join  
Goods ON Sell.idgoods = Goods.IDGoods  
|
```

Внешнее соединение

Иногда необходимо отображать и сохранять все строки основной таблицы независимо от того, есть ли соответствующие данные в другой таблице

Внешнее соединение

- Левое (LEFT OUTER JOIN) – ссылается на все строка левой таблицы
- Правое (RIGHT OUTER JOIN) ссылается на все строка правой таблицы
- Полное (FULL OUTER JOIN) – возвращает все строки из обеих таблиц

```
Select goods.nameeg, sell.num  
from sell  
right outer join  
goods ON sell.idgoods = goods.IDGoods
```

Подзапрос

- Часто невозможно решить поставленную задачу путем одного запроса.
- Например, когда при использовании условия поиска в предложении WHERE значение, с которым надо сравнивать, заранее не определено и должно быть вычислено в момент выполнения оператора SELECT.
- Внутренний подзапрос представляет собой также оператор SELECT

Подзапрос

- **Подзапрос** – это инструмент создания временной таблицы, содержимое которой извлекается и обрабатывается внешним оператором. Текст *подзапроса* должен быть заключен в скобки.
- фраза **ORDER BY** не используется, хотя и может присутствовать во внешнем *подзапросе*;
- список в предложении **SELECT** состоит из имен отдельных столбцов или составленных из них выражений
- по умолчанию имена столбцов в *подзапросе* относятся к таблице, имя которой указано в предложении **FROM**

Подзапрос

- Существует два типа *подзапросов*:
 - **Скалярный подзапрос** возвращает единственное значение..
 - **Табличный подзапрос** возвращает множество значений, т.е. значения одного или нескольких столбцов таблицы, размещенные в более чем одной строке.

Подзапрос

- Определить дату продажи максимальной партии товара.

```
SELECT Datesell, num  
FROM sell  
WHERE num= (SELECT Max (num) FROM sell)
```

Подзапрос

- Определить даты сделок, превысивших по количеству товара среднее значение и указать для этих сделок превышение над средним уровнем.

```
num- (SELECT Avg (num)
      FROM sell) AS Превышение
FROM sell
WHERE num >
      (SELECT Avg (num)
      FROM sell)
```

Подзапрос

- Определить клиентов, совершивших сделки с максимальным количеством товара.

```
SELECT Client.Firma
FROM Client INNER JOIN Sell
ON Client.Idclient=sell.idclient
WHERE sell.num=
      (SELECT Max(sell.num)
FROM sell)
```

Подзапрос

- Определить клиентов, в сделках которых количество товара отличается от максимального не более чем на 10%.

```
SELECT Client.Firma
FROM Client INNER JOIN Sell
ON Client.Idclient=sell.idclient
WHERE sell.num>=0.9*
      (SELECT Max(sell.num)
FROM sell)
```

- Определить даты, когда среднее количество проданного за день товара оказалось больше 40 единиц

```
SELECT Month(Sell.DateSell), Avg(Sell.Num) AS  
    Среднее_за_месяц  
FROM Sell  
GROUP BY Month(Sell.DateSell)  
HAVING Avg(Sell.Num) >40|
```

Табличный подзапрос

- Определить проданные товары

```
SELECT NameG
FROM Goods
WHERE IDGoods In
      (SELECT Idgoods FROM sell)
```

Табличный подзапрос

- Определить товары, не пользующиеся спросом

```
SELECT NameG
FROM Goods
WHERE IDGoods Not In
      (SELECT Idgoods FROM sell)
```

|

Табличный *подзапрос*

- Ключевые слова ANY и ALL могут использоваться с *подзапросами*, возвращающими один столбец чисел
- Если *подзапросу* будет предшествовать ключевое слово ALL, условие сравнения считается выполненным, только когда оно выполняется для всех значений в результирующем столбце *подзапроса*.
- Если записи *подзапроса* предшествует ключевое слово ANY, то условие сравнения считается выполненным, когда оно выполняется хотя бы для одного из значений в результирующем столбце *подзапроса*.

- Определить клиентов, совершивших сделки с максимальным количеством товара

(Если в результате выполнения *подзапроса* получено пустое значение, то для ключевого слова ALL условие сравнения будет считаться выполненным , а для ключевого слова ANY – невыполненным.)

```
SELECT Client.Firma, sell.num
FROM client INNER JOIN sell
    ON client.idclient=sell.idclient
WHERE sell.num>=ALL(SELECT num
    FROM sell)
```

```
SELECT Client.Firma, sell.num
FROM client INNER JOIN sell
      ON client.idclient=sell.idclient
WHERE sell.num>=Any(SELECT num
      FROM sell)
```

Представления

- Представление – это виртуальная таблица, которая сама по себе не содержит никакой информации. Она содержит только запрос, составляемый во время создания представления.

Представления

CREATE VIEW имя представления
AS

Команда SELECT

```
create view test as  
select * from goods  
where vid='конфеты'
```

Выборка данных из представления

Обращение к *представлению* осуществляется с помощью оператора **SELECT** как к обычной таблице.

SELECT * FROM имя_представления

Представление можно использовать в запросе так же, как и любую другую таблицу

Ограничения представлений

- Нельзя применять конструкции
 - Order by
 - Compute
 - Compute by
- Нельзя применять ключевое слово Into
- Нельзя ссылаться на временную таблицу