Введение в язык SQL

2.Построение запросов к нескольким таблицам

к.т.н. Герасимов Н.А., Магин Б.Е.

Занятие 2. Построение запросов к нескольким таблицам

- **Тема занятия 2**: использование в запросах нескольких таблиц,
- Для демонстрации примеров SQL-запросов будем использовать учебную базу данных Avto.mdb, описание которой дано в приложении 1.

к.т.н. Герасимов Н.А., Магин Б.Е.

3.1. Объединение таблиц в запросах и выполнение операции JOIN (объединение)

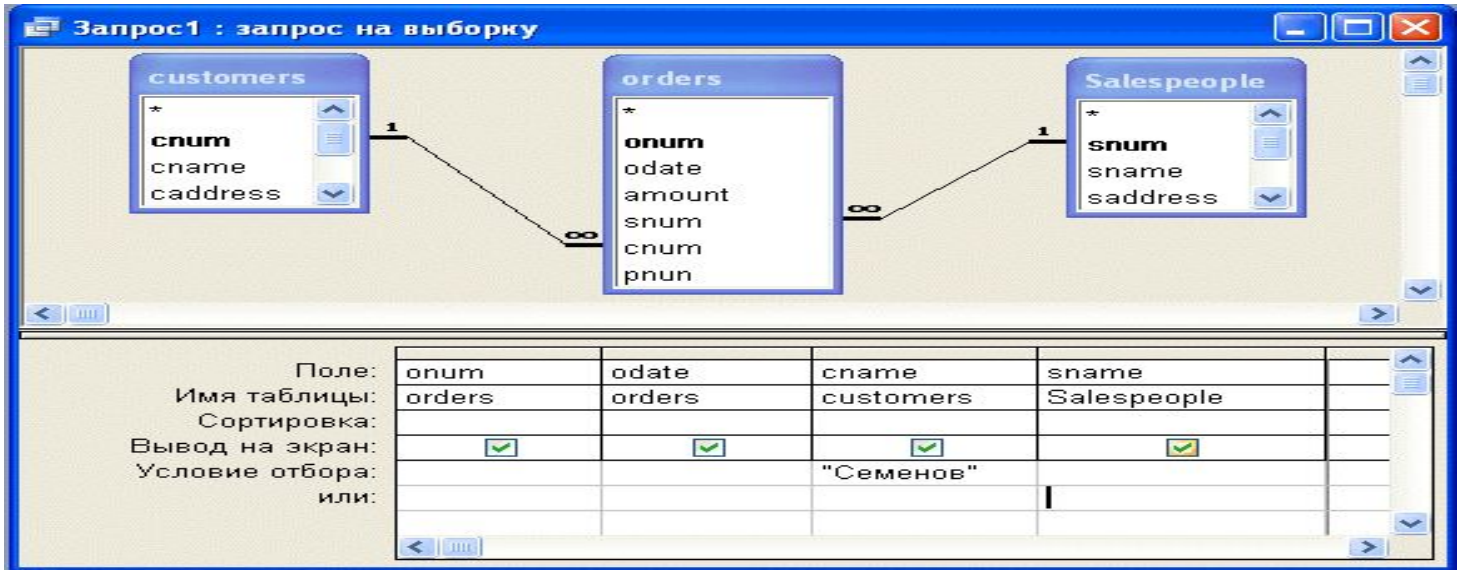
- ***Предположим, нам надо получить ответ на вопрос: сколько заказов сделал покупатель с фамилией Семенов, причем в результате мы хотим получить таблицу, которая содержит только 4-ре следующих столбца: номер заказа, дата заказа, фамилия покупателя и фамилия продавца.***
  Понятно, что из одной таблицы Orders (Заказы) мы всю необходимую информацию по данному запросу не получим, т.к. таблица orders не содержит полей sname и cname. Поэтому для построения этого запроса необходимо привлечь таблицы Customers и Salespeople.

Построение запроса к нескольким таблицам с помощью QBE

- Для построения запросов над несколькими таблицами в Access удобно пользоваться режимом Конструктор (Запросы→Конструктор), который открывает окно построения запроса QBE, состоящее из двух частей: зона таблиц (сверху) и зона для макета запроса (снизу) (более подробно о QBE см. Занятие 1).
- В верхнюю зону поместим три таблицы Customers, Orders и Salespeople, в нижнюю часть (зона запроса) выберем столбцы: onum и odate из таблицы orders, cname из таблицы Customers и sname из таблицы Salespeople, как показано на рис. ниже.

Вид запроса в QBE

- Пример построения запроса с тремя таблицами в QBE



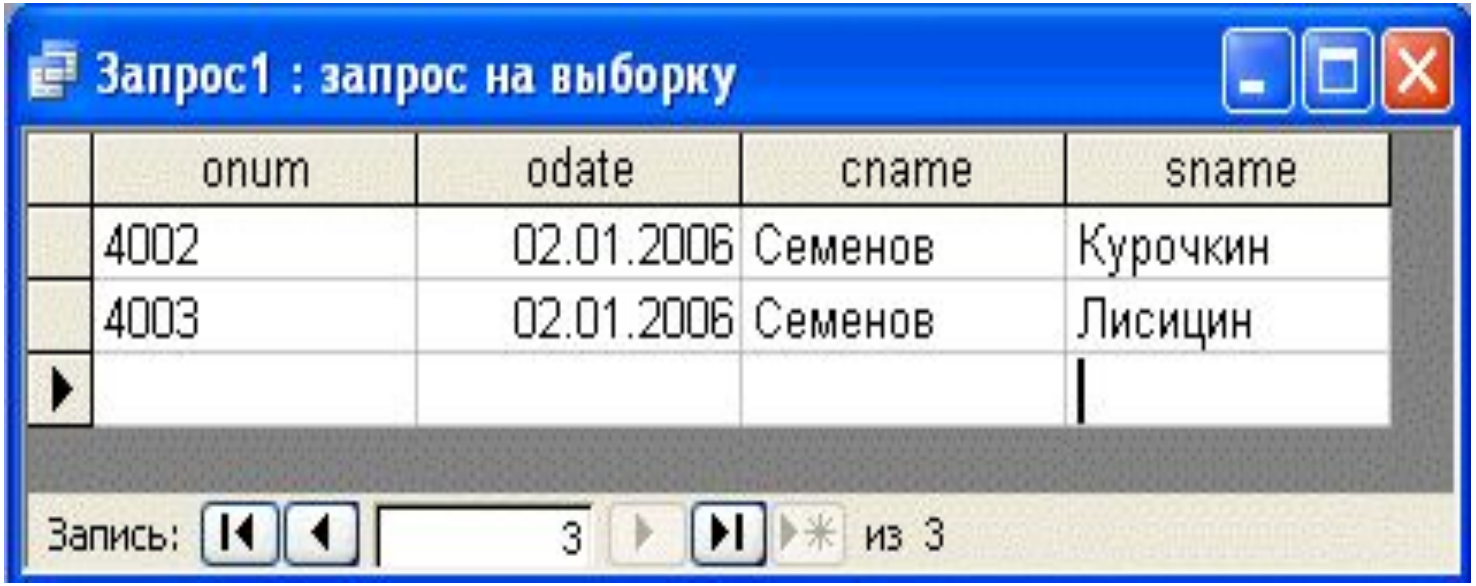
к.т.н. Герасимов Н.А., Магин Б.Е.

Пояснение:

- В столбце с именем покупателя (cname.customers) запишем условие отбора записей (“Семенов”) (см.рис выше).
- Выполнение запроса даст следующий результат, который показан на рис. ниже, т. е. будут отобраны только строки с заказами, которые обслужили покупателя с фамилией «Семенов».

Результат запроса к нескольким таблицам

- Результат запроса к БД «Авто.mdb»



к.т.н. Герасимов Н.А., Магин Б.Е.

- Посмотрим, как выглядит это запрос в режиме SQL, для чего войдем в редактор языка SQL через соответствующую пиктограмму или выполнив команду Вид→Запрос на SQL. Открывается окно редактора SQL и теперь мы можем рассмотреть текст запроса на языке SQL:

```
 SELECT orders.onum, orders.odate,
customers.cname, Salespeople.sname
 FROM Salespeople INNER JOIN (customers
INNER JOIN orders ON customers.cnum =
orders.cnum) ON      Salespeople.snum =
orders.snum
WHERE (((customers.cname)="Семенов"));
```

- Отметим, что имена столбцов в строке с ключевым словом SELECT записаны в полной синтаксической структуре: **<имя таблицы>.<имя столбца>.**
  Это необходимо, чтобы каждый столбец был четко приписан к соответствующей таблице.
- В строке с ключевым словом **FROM** используются все три таблицы **Orders, Salespeple и Customers**, которые объединены с помощью оператора **INNER JOIN** и с указанием соответствующих условий связи между ними (в нашем случае **customers.cnum = orders.cnum** и **salespeople.snum = orders.snum**).

Построение запроса с объединением таблиц

- Разобраться в структуре такого SQL сразу достаточно трудно, поэтому попытаемся создать аналогичный запрос вручную с помощью редактора SQL-запросов, путем построения сначала более простого запроса на объединение двух таблиц, а затем добавим в него необходимые ограничения с помощью условий.
- Запрос сформулируем следующим образом **отобрать все заказы для продавцов, которые живут в городе «Тула» (в Туле живет один продавец) и показать только три столбца (код заказа, дата заказа и фамилию продавца). Очевидно, что в построении запроса должны участвовать две таблицы: orders и Salespeople**

- Сначала составим запрос без условия на отбор строк используя строки с ключевыми словами SELECT и FROM:

SELECT orders.onum, orders.odate, Salespeople.sname
FROM Orders, Salespeople

Результатом запроса будет объединение строк таблиц Orders и Salespeople по полям orders.onum и salespeople.sname и мы получим таблицу из 30 записей как показано на рис. 3.3. ниже, которая отражает всевозможные комбинации указанных столбцов.

Объединение двух таблиц

- Результат объединения двух таблиц Orders и Salespeople

Запрос1 : запрос на выборку

onum	odate	sname
4001	01.01.2006	Курочкин
4002	02.01.2006	Курочкин
4003	02.01.2006	Курочкин
4004	03.01.2006	Курочкин
4005	03.01.2006	Курочкин
4001	01.01.2006	Лисицин
4002	02.01.2006	Лисицин
4003	02.01.2006	Лисицин
4004	03.01.2006	Лисицин
4005	03.01.2006	Лисицин
4001	01.01.2006	Волков
4002	02.01.2006	Волков
4003	02.01.2006	Волков
4004	03.01.2006	Волков
4005	03.01.2006	Волков
4001	01.01.2006	Змеев
4002	02.01.2006	Змеев
4003	02.01.2006	Змеев
4004	03.01.2006	Змеев
4005	03.01.2006	Змеев
4001	01.01.2006	Веселов
4002	02.01.2006	Веселов
4003	02.01.2006	Веселов
4004	03.01.2006	Веселов
4005	03.01.2006	Веселов
4001	01.01.2006	Трофимов
4002	02.01.2006	Трофимов
4003	02.01.2006	Трофимов
4004	03.01.2006	Трофимов
4005	03.01.2006	Трофимов

Запись: 30 из 30

к.т.н. Герасимов Н.А., Магин Б.Е.

- Часть строк этого результата являются лишними (не реальные), т.к. не отвечают нашему условию и не содержатся в таблице orders. Поэтому чтобы отобрать только «реальные строки» необходимо добавить условие, отражающее межсущностную связь (межу таблицами Salespeople и orders): Salespeople.snum = orders.snum,
которое требует отобрать только те строки, в которых номер продавца в двух таблицах совпадает. SQL-запрос с дополнительным условием будет выглядеть так:

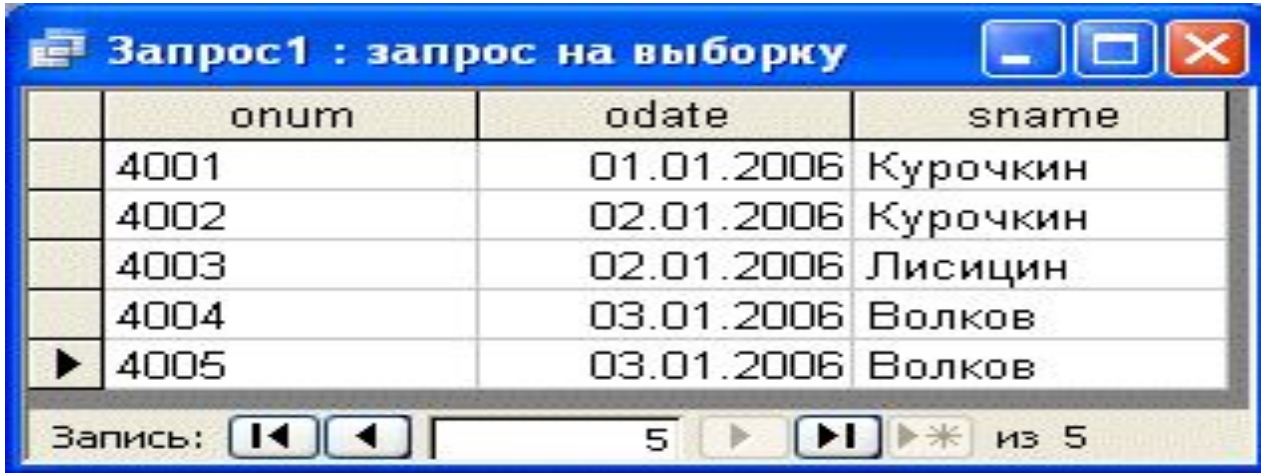
Запрос с дополнительным условием

- SELECT orders.onum, orders.odate, Salespeople.sname
  FROM Orders, Salespeople
  WHERE Salespeople.snum = orders.snum

- Теперь получим таблицу с «реальными» строками, которые отражают все строки в таблице orders и фамилии продавцов.

Запрос к двум таблицам

- Запрос над двумя таблицами с дополнительным условием (Salespeople.snum = orders.snum)




```
SELECT orders.onum, orders.odate, Salespeople.sname
    FROM Orders, Salespeople
    WHERE Salespeople.snum = orders.snum
```

Добавление условия на селекцию строк

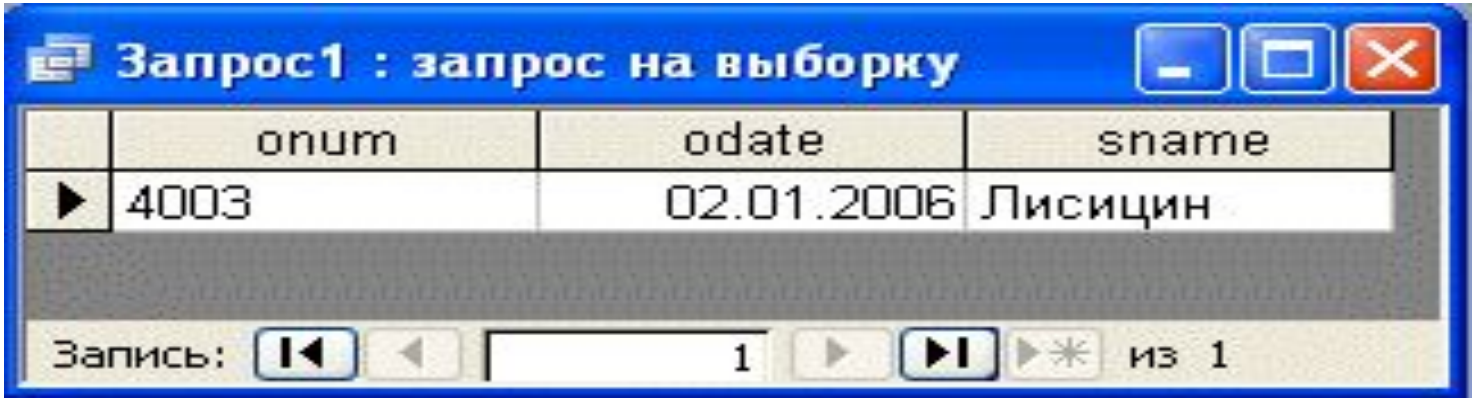
- Теперь в текст SQL-запроса добавим еще одно условие отбора строк: живущие в городе «Тула» (или Salespeople.saddress ='Тула') и соединим его при помощи логического оператора AND с условием в строке с ключевым словом WHERE. Полный запрос, отвечающий на посталенный вопрос показан ниже:

```
SELECT orders.onum, orders.odate, Salespeople.sname
FROM Orders, Salespeople
WHERE (Salespeople.snum = orders.snum) AND
    (Salespeople.saddress ='Тула');
```

- В результате получим искомую таблицу с одной строкой, отвечающую на поставленный вопрос (см. рис.3.5).

Запрос к нескольким таблицам с условием

- Запрос над двумя таблицами с условием Salespeople.saddress ='Тула'




```
SELECT orders.onum, orders.odate, Salespeople.sname
FROM Orders, Salespeople
WHERE (Salespeople.snum = orders.snum) AND
(Salespeople.saddress ='Тула');
```

Сравнение SQL запросов

- Аналогичный результат можно получить, используя оператор INNER JOIN для объединения таблиц с включением условия объединения в строку с ключевым словом FROM. В этом случае запрос на SQL будет выглядеть следующим образом:

```
SELECT Orders.onum, Orders.odate,
Salespeople.sname
FROM Salespeople INNER JOIN Orders ON
Salespeople.snum = Orders.snum
WHERE Salespeople.saddress ='Тула';
```

Замечание:

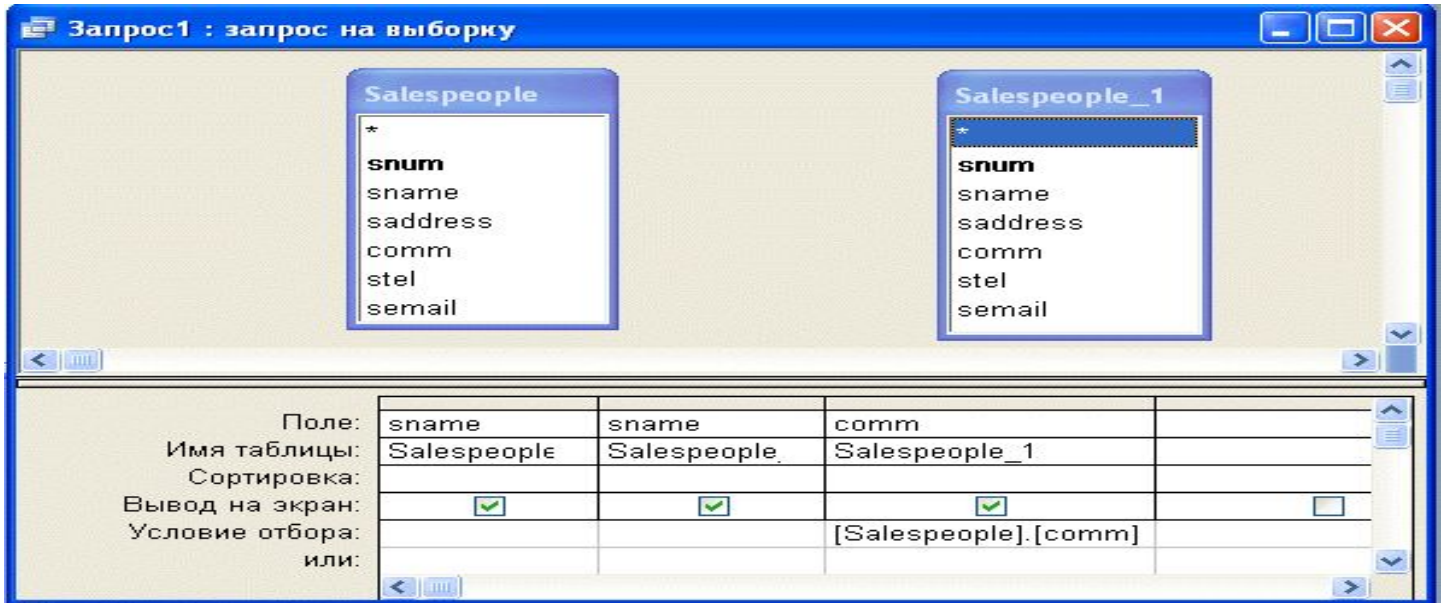
- Следует отметить, что поле snum в таблице Salespeople является первичным ключем (**PK**-primary key), а поле snum в таблице orders – внешним ключом (**FK** – foreign key), который используется для связи и организации проверки целостности. Более подробно о целостности в реляционных базах данных смотри в работе [1,2]. **Таким образом, ключевое слово INNER JOIN с условием на первичных и внешних ключах позволяет упростить структуру запроса над несколькими таблицами, оставляя в строке с ключевым словом WHERE только условия над обычными полями.**

3.2. Объединение копий таблиц

- ***В сложных запросах можно объединять не только разные таблицы, но и копии таблиц. Например, если необходимо найти все пары продавцов, которые имеют одинаковые комиссионные. Для этого с помощью Конструктора создадим запрос как показано на рис. 3.6. Здесь таблица Salespeople выбрана два раза, поэтому в зоне таблиц находится таблица Salespeople и ее копия Salespeople_1.***

Запрос над копиями одной таблицы

- Пример запроса над таблицей Salespeople и ее копией (Salespeople_1).



Построение SQL запроса над копиями одной таблицы

- Текст соответствующего SQL-запроса показан ниже:

```
SELECT Salespeople.sname, Salespeople_1.sname,
Salespeople_1.comm
  FROM Salespeople, Salespeople AS Salespeople_1
  WHERE (((Salespeople_1.comm)=[Salespeople].[comm])
```

- Имя второй таблицы и ее синоним (Salespeople AS Salespeople_1) содержится в строке с ключевым словом FROM.
- Результат этого запроса будет содержать все комбинации пар фамилий, которые имею одинаковый рейтинг, в том числе и строки с идентичными фамилиями (см. рис. ниже).

Запрос над копиями одной таблицы

- Результат запроса с двумя таблицами Salespeople. (***найти все пары продавцов, которые имеют одинаковые комиссионные***)

SELECT Salespeople.sname, Salespeople_1.sname, Salespeople_1.comm FROM **Salespeople**, Salespeople AS **Salespeople_1** WHERE (((Salespeople_1.comm)=[Salespeople].[comm])

Модификация запроса с копиями таблицы

- Исключить строки с одинаковыми значениями в первом и втором столбцах можно, добавив через логическую связку AND условие типа: Salespeople_1.snum < Salespeople.snum (т.е. «номер продавца в первой таблице должен быть строго меньше номера во второй таблице»). Это позволит исключить строки типа «Веселов=Веселов». В этом случае полный запрос будет выглядеть как показано ниже:

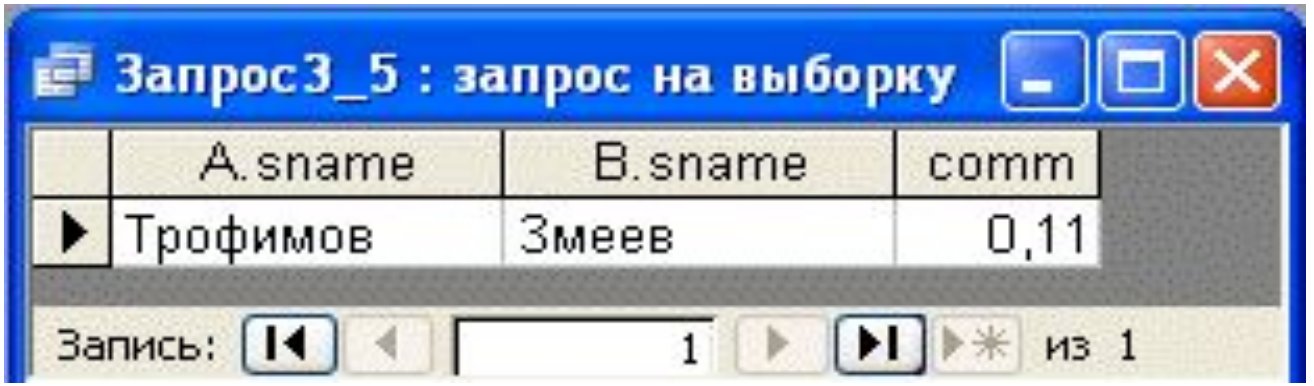
```
SELECT Salespeople.sname, Salespeople_1.sname,
Salespeople_1.comm
 FROM Salespeople, Salespeople AS Salespeople_1
 WHERE (((Salespeople_1.comm) = [Salespeople].[comm])
       AND ((Salespeople_1.snum) < [Salespeople].[snum]));
```

Пояснение:

- Результат запроса с дополнительным условием показан на рис.ниже. Здесь отобрана только одна строка, в которой продавец «Трофимов» имеет комиссионные такие же как и продавец «Змеев», что полностью соответствует поставленному вопросу.

Результат запроса

- Рис.3.8. Результат отбора строк с дополнительным условием (Salespeople_1.snum < Salespeople.snum);



Модификация запроса

- Следует отметить, что конструктор, строит синонимы таблиц по упрощенному алгоритму. В общем случает можно использовать любые синонимы для выбранных таблиц. Вот как будет выглядеть тот же запрос, если мы для первой таблицы Salespeople будем использовать букву A, а для второй таблицы Salespeople - ,букву B:

  SELECT A.sname, B.sname, B.comm

  FROM Salespeople AS A, Salespeople AS B

  WHERE B.comm = A.comm AND B.snum < A.snum;

  Результат показан на рис.ниже.

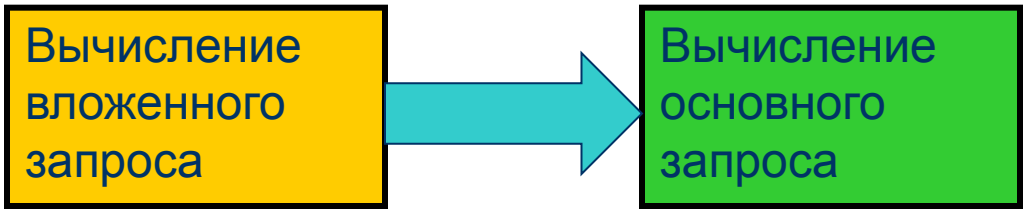
  Таким образом, вводя синонимы исходным таблицы базы данных, пользователь может существенно расширить возможности при построении сложных запросов над несколькими таблицами.

3.3.Вложенные запросы

- Язык SQL позволяет строить вложенные запросы, т.е. такие запросы, которые в условиях используют результаты работы другого запроса. Например, нам известна фамилия продавца (пусть это «Курочкин»), но не помним его номера, а нам необходимо знать все его заказы. Этот запрос можно выполнить в два этапа:
- - сначала найти по таблице Salespeople код продавца (SELECT snum FROM Salespeople WHERE sname = "Курочкин");
- - затем, по коду продавца (код продавца с фамилией «Курочкин» равен ‘0001’) выбрать из таблицы orders все его заказы (SELECT * FROM Orders WHERE snum=’0001’).

Логика вложенного запроса

- Такой способ получения результата не очень удобный, и его можно получить другим способом – построением вложенного запроса. Для этого в Конструкторе в строке «Условия запроса» введем запрос на поис кода продавца. На рис.ниже. показано как выглядит вложенный запрос в Конструкторе.



Построение вложенного запроса в QBE

- Построение вложенного запроса

к.т.н. Герасимов Н.А., Магин Б.Е.

SQL-запрос с вложенным запросом

- Текст SQL-запроса с вложенным запросом показан ниже:

```
SELECT *    FROM Orders
WHERE
snum=(SELECT snum FROM Salespeople
WHERE sname = "Курочкин")
```

Пояснение:

- Важным условием корректности исполнения вложенных запросов, является однозначность результата «внутреннего» (вложенного) запроса. Нашем случае мы имеем однозначный результат внутреннего запроса (код продавца равен ‘0001’).
- Результ выполнения вложенного запроса показан на рис. 9.10.

Выполнение запроса с вложенным подзапросом

- Результат SQL-запроса с вложенным запросом

SELECT * FROM Orders WHERE snum=(**SELECT snum FROM Salespeople WHERE sname = "Курочкин")**

Усложнение запроса

- Более сложный вложенный запрос, построенный на основании предыдущего запроса может выглядеть так:

```
SELECT orders.onum as НОМЕР, orders.odate as ДАТА,
Salespeople.sname as ФАМИЛИЯ,
        price.pname as МАРКА, orders.amount as ЦЕНА
FROM Salespeople INNER JOIN (price INNER JOIN orders ON
        price.pnum = orders.pnun) ON Salespeople.snum = orders.snum
WHERE Salespeople.snum =(SELECT snum FROM Salespeople
WHERE sname = "Курочкин");
```

- Результат этого запроса, который использует несколько таблиц (Orders, Salespeople и Price) и содержит встроенный запрос, который по фамилии ("Курочкин") определяет код продавца показан на рис. ниже.

Результат

- Результат сложного запроса с несколькими таблицами и встроенным запросом.