

Лекція 14-2020_2021

- На попередній лекції
- Підзапити в операторах модифікації
- Перехресні запити (кростаби)
- Оператор LOAD DATA
- Завдання на співбесіді

ХНУ ім.В.Н.Каразіна, ФКН,
Лазурик В.М.

1

Что было в предыдущей лекции

4. Составили план проведения запроса по проведению акции по стимуляции потребительского спроса. Рассмотрели реализацию запроса на SQL.

ХНУ ім.В.Н.Каразіна, ФКН,
Лазурик В.М.

3

Что было в предыдущей лекции

1. Рассмотрели алгоритм построения иерархического запроса и реализацию на SQL запросов типа «Поиск родственников», имея 2 таблицы – люди и взимосвязь между людьми.
2. **Соотнесенным** (коррелированным, зависимым) называется такой подчиненный запрос, внутри которого используются значения из внешнего запроса. Такой подчиненный запрос содержит ссылку на поля главного запроса. Соотнесенный подзапрос выполняется неоднократно, по одному разу для каждой строки таблицы основного запроса.
3. Подзапросы или представления используются всегда, когда нужно применить последовательно две агрегатные функции

ХНУ ім.В.Н.Каразіна, ФКН,
Лазурик В.М.

2

Подзапросы в операторах модификации

Продавцы оптовых складов обслуживают фирмы-покупатели. Покупатели делают заказы. Для обслуживания за покупателем закреплен продавец.

Продавцы: **CREATE TABLE SalPeople**

```
( snum integer NOT NULL,  
  sname char (10) NOT NULL, city char (10),  
  comm decimal);
```

Покупатели: **CREATE TABLE Customers**

```
( custnum integer NOT NULL  
  AUTO_INCREMENT PRIMARY KEY ,  
  firm char(10),city char(10),  
  sale integer, rating integer,  
  FOREIGN KEY (sale)
```

```
REFERENCES SalPeople ( snum );
```

ХНУ ім.В.Н.Каразіна, ФКН,
Лазурик В.М.

4

Подзапросы в операторах модификации

Заказы:

```
CREATE TABLE Orders
( onum integer NOT NULL,
  sum integer, dat date,
  cust integer, sal integer ,
  FOREIGN KEY (cust)
      REFERENCES Customers (custnum ) ,
  FOREIGN KEY (sal)
      REFERENCES SalPeople (snum ) ;
```

Подзапросы в операторах модификации

Пример 1. Продавец Иванов уволился. Нужно переназначить его покупателей Петрову.

```
UPDATE Customers
SET sale = (Select snum FROM SalPeople
            WHERE sname = "Петров")
WHERE sale = (Select snum FROM SalPeople
            WHERE sname = "Иванов") ;
```

Подзапросы в операторах модификации

Пример 2. В нашей сети оптовых складов закрыли склад в Сумах, необходимо удалить всех заказчиков, назначенных к продавцам в Сумах.

```
DELETE FROM Customers
WHERE sale = ANY
( SELECT snum
  FROM Salespeople
  WHERE city = 'Сумы' );
```

Подзапросы в операторах модификации

Пример 3. Из БД удалить записи о всех продавцах, у которых рейтинг = 0.

Вариант 1.

```
DELETE FROM Salespeople
WHERE EXISTS
( SELECT * FROM Customers WHERE rating = 0
  AND Salespeople.snum = Customers.sale );
```

Вариант 1.

```
DELETE FROM Salespeople WHERE 0 IN
( SELECT rating FROM Customers
  WHERE Salespeople.snum = Customers.sale);
```

Подзапросы в операторах модификации

Пример 4. Необходимо создать таблицу **Multi** с такой же структурой как **Salespeople**. Заполнить ее сведениями о тех продавцах, кто имеет более одного заказчика.

```
MySQL: CREATE TEMPORARY TABLE Multi
        Like SalPeople;
INSERT INTO Multi
    SELECT * FROM Salespeople
    WHERE 1 < (SELECT COUNT (*) FROM Customers
        WHERE Customers.sale = Salespeople.snum);
```

Подзапросы в операторах модификации

Пример 5. Увеличить комиссионные тем продавцам, которые имеют хотя бы 2 заказчика.

```
UPDATE Salespeople
    SET comm = comm + .01
    WHERE 2 < =
        ( SELECT COUNT (sale)
          FROM Customers
          WHERE Customers.sale =
            Salespeople.snum );
```

Подзапросы в операторах модификации

Пример 4. Access :

```
SELECT S.* INTO Multi
FROM (SELECT * FROM Salespeople
    WHERE 1 <
        (SELECT COUNT (*) FROM Customers
            WHERE Customers.sale =
                Salespeople.snum) as S;
```

Подзапросы в операторах модификации

Основной запрос и подзапрос не могут в качестве источника записей иметь одну и ту же таблицу.

Пример 6. Пусть таблица **SalTown** имеет такую же структуру, что и **Salespeople**. Заполним ее данными о продавцах, которые имеют заказчиков в Днепре.

```
INSERT INTO SalTown
    SELECT * FROM Salespeople
    WHERE snum = ANY
        ( SELECT sale FROM Customers
          WHERE city = 'Днепр' );
```


Подзапросы в операторах модификации

Пример 7. Пусть есть таблица **Bonus**, которая содержит поле **snum** продавца, поле **dat** сделки и поле **sum** стоимости заказа. Вводим информацию о продавцах, которые имеют максимальный заказ каждый день. Здесь основной запрос работает с таблицей **Bonus**, а **SELECT** с таблицей **Orders**. Это правильно.

```
INSERT INTO Bonus
SELECT sal, dat, sum
FROM Orders a
WHERE sum =
( SELECT MAX (sum )
FROM Orders b
WHERE a.dat = b.dat );
```

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

13

MySQL оператор LOAD DATA

LOAD DATA – загрузка данных в таблицу из файла.
LOAD DATA [LOCAL]

```
INFILE 'file_name.txt'
[REPLACE | IGNORE]
INTO TABLE tbl_name
[LINES TERMINATED BY '\n']
```

1. Задано **LOCAL** – файл читается с клиентского хоста, иначе с сервера.
2. Незаполненные поля имеют значение **NULL**. В текстовом файле это значение – символами **\N**.
3. По умолчанию разделитель между значениями столбцов – символ табуляции. (**\t**)
4. По умолчанию маркер конца строки – символ перевода строки (**\n**)

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

15

Подзапросы в операторах модификации

Пример 8. Удалить всех продавцов с рейтингом, ниже среднего.

Так нельзя: **DELETE FROM Customers**
WHERE rating < (SELECT AVG (rating)
FROM Customers);

MySQL:

```
SELECT @Sr := AVG (rating) FROM Customers;
DELETE FROM Customers WHERE rating < @Sr;
```

Access:

```
SELECT T.* INTO Temp FROM
(SELECT AVG (rating) as Sr
FROM Customers) as T;
DELETE FROM Customers
WHERE rating < (Select Sr From Temp);
```

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

14

MySQL оператор LOAD DATA

5. Указано ключевое слово **REPLACE** – новые строки заменят существующие с таким же уникальным ключом.
6. Указано **IGNORE** – входные строки, имеющие тот же уникальный ключ, что и существующие, будут пропускаться.
7. Если не указано ни **REPLACE**, ни **IGNORE** – при обнаружении дублирующегося значения ключа возникает ошибка и оставшаяся часть текстового файла игнорируется.

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

16

MySQL оператор LOAD DATA

Пример 8. Загрузить данные из файла на диске клиента

```
LOAD DATA LOCAL INFILE  
'c:/input/aaa.txt' INTO TABLE tab;
```

Пример 9. Загрузить с клиентского хоста текстовые данные, экспортированные из Access.

В текстовых файлах Windows каждая строка заканчивается парой символов “ перевод строки” + “возврат каретки”.

```
LOAD DATA LOCAL INFILE  
'c:/dd.txt' INTO TABLE tab  
LINES TERMINATED BY '\r\n';
```

LOAD DATA с русскоязычными БД

LOAD DATA при загрузке использует кодировку, принятую для БД по умолчанию.

Если работаем с версиями MySQL < 5.5 при необходимости загрузки текста в русскоязычную БД с использованием **LOAD DATA** необходимо кодировку по умолчанию для БД поменять на Cp1251 или UTF8.

```
ALTER DATABASE db_name  
DEAULT CHARACTER SET Cp1251  
DEAULT COLLATE Cp1251_general_ci
```

MySQL оператор LOAD DATA

Пример 10. Из файла, расположенного на сервере, загрузить данные в некоторые столбцы таблицы:

```
LOAD DATA INFILE 'd:/data.txt'  
INTO TABLE person (col1,col3,...);
```

Перекрестный запрос

Перекрестный запрос – это разновидность запроса на выборку. Результаты выполнения перекрестного запроса выводятся в виде таблицы, структура которой отличается от обычных таблиц.

Другое название перекрестного запроса – кросстаб. Благодаря своеобразной структуре перекрестного запроса его легче читать, чем простой запрос на выборку, возвращающий те же данные.

①

Фамилия	Имя категории	Сумма итогов
Кротов	Напитки	46 302,09 р.
Кротов	Приправы	16 789,35 р.
Кротов	Сладости	36 182,13 р.
Крылова	Напитки	111 047,76 р.
Крылова	Приправы	49 566,21 р.
Крылова	Сладости	80 005,35 р.

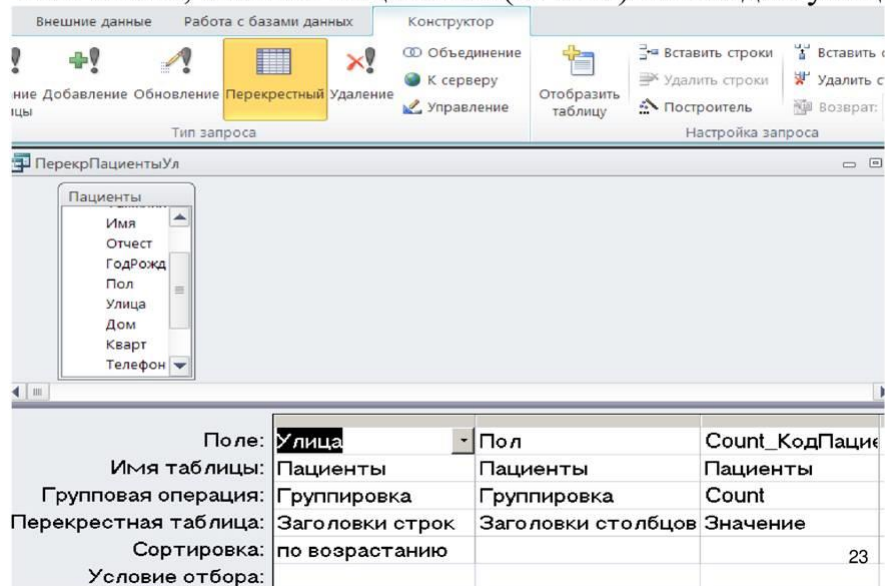
2

Фамилия	Напитки	Приправы	Сладости
Кротов	46 302,09 р.	16 789,35 р.	36 182,13 р.
Крылов а	111 047,76 р.	49 566,21 р.	80 005,35 р.

Необходимо указать:

1. Поля для заголовков строк.
2. Поля для заголовков столбцов.
3. Поля для значений, по которым вычисляются сводные данные. При задании заголовков столбцов и суммируемых значений можно использовать только одно поле. При задании заголовков строк можно использовать до трех полей

Посчитать, сколько пациентов (м. и ж.) на каждой улице.



1. Посчитать, сколько пациентов (мужчин и женщин) проживает на каждой улице.

Улица	Ж	М
Базарная		1
Гоголя	1	1
Ленина		1
Пушкинская	3	1
Светлая		2
Сумская		5
Широнинцев		1
Якира		1

TRANSFORM

Count (Пациенты.КодПациент)
AS Количество

```
SELECT Пациенты.Улица
FROM Пациенты
GROUP BY Пациенты.Улица
ORDER BY Пациенты.Улица
PIVOT Пациенты.Пол ;
```

Создание перекрестных запросов

1. Посчитать, сколько пациентов (мужчин и женщин) родилось в каждом году и каждом месяце года.

ПерекрПациентыМес2

Поле:	Год: Year(Месяц: Month(Пол	Колич: Код
Имя таблицы:			Пациенты	Пациенты
Групповая операция:	Группировка	Группировка	Группировка	Count
Перекрестная таблица:	Заголовки	Заголовки стр	Заголовки столбцов	Значение
Сортировка:				
Условие отбора:				

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

25

Создание перекрестных запросов

ПерекрПациентыМес2

Год	Месяц	ж	м
1900	9		1
1930	7		1
1939	2		1
1940	5		1
1940	10		1
1948	5		1
1955	9		1
1960	4		1
1969	3		1
1970	5	1	
1970	8	1	1
1980	6	1	
1980	10	1	
1995	5		1
2000	1		1
2011	1		1

TRANSFORM

Count (КодПациент) AS
Колич

SELECT Year (Рожд) AS Год,
Month (Рожд) AS Месяц

FROM Пациенты

GROUP BY Year (Рожд) ,
Month (Рожд)

PIVOT Пол;

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

26

Создание перекрестных запросов

1. Access 2010 для перекрестных запросов не допускает ввод параметров. Запрос: Вывести в виде двумерного кросстаба количество пациентов, принятых каждым врачом в каждую дату. Если не требуется уточнять год и месяц, запрос выглядит:

TRANSFORM Count (Прием.КодПациент) AS Колич

SELECT [Врачи]![Фамилия] & " " &
Left ([Врачи]![Имя],1) & ". " &
Left ([Врачи]![Отчест],1) & ". " AS ФИО

FROM Врачи INNER JOIN Прием

ON Врачи.КодВрача = Прием.КодВрача

GROUP BY [Врачи]![Фамилия] & " " &
Left ([Врачи]![Имя],1) & ". " &
Left ([Врачи]![Отчест],1) & ". "

PIVOT Прием.Дата;

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

27

Создание перекрестных запросов

перекрДаты

ФИО врача	01_01_2015	02_01_2015	03_01_2015	04_01_2015
Борисов О. В.	1			
Дрозд В. П.	1			
Коваленко Т. Б.				1
Котов И. В.	2	1		
Николаев П. Н.			2	

1. Если требуется уточнить год и месяц, введение параметра вызывает ошибку ядра Access. Даже если перекрестный запрос построен на представлении, которое запрашивает год и/или месяц, возникает та же ошибка.



28

Создание перекрестных запросов

1. Задача. Вывести в виде перекрестного запроса количество пациентов, принятых каждым врачом в каждую дату указанного года и месяца.

Прием					
КодПриема	КодПс	КодВр	Дата	Жалобы	
1	1	2	01.01.2015	все плохо	
2	2	2	01.01.2015	все пропало	
3	3	2	02.01.2015	все болит	
4	1	5	03.02.2015	тревожно	
5	5	6	01.02.2015	болит голова	
6	5	5	03.01.2015	болит колено	
7	1	3	01.01.2015	болит живот	
8	2	1	04.03.2015	болит пятка	

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

29

Создание перекрестных запросов

Решение. Шаг 1. Создать временную таблицу.

```
SELECT Врачи!Фамилия & " " & Left(Врачи!Имя,1)
& "." & Left(Врачи!Отчест,1) & "." AS ФИО,
Прием.КодПациент, Прием.Дата INTO temp2
FROM Врачи INNER JOIN Прием
ON Врачи.КодВрача = Прием.КодВрача
WHERE (Year(Прием.Дата)=[Год] and
Month(Прием.Дата)=[Месяц]);
```

1. Год = 2015

2. Месяц = 1

temp2		
ФИО врача	КодПациент	Дата
Котов И. В.	1	01.01.2015
Котов И. В.	2	01.01.2015
Котов И. В.	3	02.01.2015
Дрозд В. П.	1	01.01.2015
Николаев П. Н.	5	03.01.2015

30

Создание перекрестных запросов

Решение. Шаг 2. Создать перекрестный запрос

```
TRANSFORM Count(temp2.КодПациент) AS Колич
SELECT temp2.[ФИО врача]
FROM temp2
GROUP BY temp2.[ФИО врача]
PIVOT temp2.Дата;
```

Запрос15			
ФИО врача	01_01_2015	02_01_2015	03_01_2015
Дрозд В. П.	1		
Котов И. В.	2	1	
Николаев П. Н.			1

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

31

Создание перекрестных запросов

Другой вид запроса. Не перекрестный.

```
SELECT [Врачи]![Фамилия] & " " &
Left([Врачи]![Имя],1) & "." &
Left([Врачи]![Отчест],1) & "." AS ФИО,
Прием.Дата, Count(Прием.КодПациент) AS Колич
FROM Врачи INNER JOIN Прием
ON Врачи.КодВрача = Прием.КодВрача
WHERE (Year(Прием.Дата)=[Год] and
Month(Прием.Дата)=[Месяц])
Group by [Врачи]![Фамилия] & " " &
Left([Врачи]![Имя],1) & "." &
Left([Врачи]![Отчест],1) & ".", Прием.Дата;
```

Запрос15		
ФИО	Дата	Колич
Дрозд В. П.	01.01.2015	1
Котов И. В.	01.01.2015	2
Котов И. В.	02.01.2015	1
Николаев П. Н.	03.01.2015	1

32

Задача 1 на собеседовании на разделение

таблиц

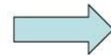
Есть
таблица

col1	col2
1	1
1	3
null	null
null	2

Нужно
получить

col1
1
null
1
3
null
2

```
SELECT tabl.col1
      as ColValue
FROM tabl
UNION
SELECT tabl.col2
      as ColValue
FROM tabl;
```



ColValue
1
2
3
null

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

33

Задача 1 на собеседовании

Решение Access

```
1. SELECT * INTO tabl2
   FROM
   (select col1 from tabl) AS s;
2. INSERT INTO
      tabl2 ( col1 )
SELECT col2 FROM tabl;
```



col1
1
1
null
null



col1
1
1
null
1
3
null
2

Решение MySQL

```
1.CREATE TABLE tabl2
   select col1 from tabl;
2.INSERT INTO tabl2 ( col1 )
SELECT col2 FROM tabl;
```

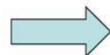
ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

34

Задача 1 на собеседовании

Простое решение и для Access и для MySQL

```
SELECT tabl.col1
      as ColValue
FROM tabl
UNION ALL
SELECT tabl.col2
      as ColValue
FROM tabl;
```



col1
1
1
null
null
1
3
null
2

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

35

Задача 2 на собеседовании

Как из таблицы, в которой есть 2 первичных ключа
удалить дубликаты записей.

Есть таблица tab

f	g
-----	-----
1	1
1	1
2	2
3	3
3	3
1	3

Должно быть после удаления

f	g
-----	-----
1	1
2	2
3	3
1	3

36

Задача 2 на собеседовании

Как из таблицы, у которой есть составной первичный ключ {f, g} удалить дубликаты записей.

Есть таблица tab

f g

1 1

1 1

2 2

3 3

3 3

1 3

Должно быть после удаления

f g

1 1

2 2

3 3

1 3

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

37

Задача 2 на собеседовании

Решение в Оракл, предлагаемое на форуме

`delete from tab a where rowid not in`

`(select min(rowid) from tab b`

`where a.f = b.f and a.g = b.g)`

Rowid – порядковый номер (индекс) строки в таблице

1 1 => rowid = 1, в подзапросе выбираем строки с одинаковыми значениями f и g.

Это строки rowid = 1 и rowid = 2. Минимальное значение = 1. В результат поступит только строка с номером = 2.

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

39

Задача 2 на собеседовании

Как из таблицы, у которой есть два поля {f, g} удалить дубликаты записей.

Есть таблица tab

f g

1 1

1 1

2 2

3 3

3 3

1 3

Должно быть после удаления

f g

1 1

2 2

3 3

1 3

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

38

Задача 2 на собеседовании

Проверить в Access совет из форума (rowid – нет)?

Шаг 1. Добавим в таблицу инкрементированный столбик, чтобы пронумеровать строки

`ALTER TABLE tab ADD COLUMN id counter;`

Шаг 2. `delete a.* from tab as a`
`where id not in`

`(select min(id) from tab as b`

`where`

`((a.f = b.f) and (a.g = b.g));`

f	g	id
1	1	1
1	1	2
2	2	3
3	3	4
3	3	5
1	3	6

f	g	id
1	1	2
3	3	5

40

Задача 2 на собеседовании

Возможное решение

1. Скопировать таблицу **tab**
2. Создание новой таблицы с уникальными строками

Access

```
SELECT T.f, T.g INTO tab_new  
FROM  
(SELECT DISTINCT tab.f, tab.g FROM tab) AS T;
```

MySQL

```
CREATE TABLE tab_new  
SELECT DISTINCT tab.f, tab.g FROM tab;
```

3. Уничтожить таблицу **tab**
4. Переименовать **tab_new** в **tab**