

Лекція 12-2020_2021

- На попередній лекції
- Змінні в MySQL
- Підлеглі запити

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

1

Что было в предыдущей лекции

Access: INSERT ...VALUES. Поле счетчика **НЕ** включают в запрос, чтобы осуществилось автоматическое изменение счетчика.

MySQL: INSERT ...VALUES. Для автоинкрементного столбика (объявлен с атрибутом **Not Null**) вставка значения **Null** вызывает автоматическое наращивание автоинкрементного ключа.

3

Что было в предыдущей лекции

Команда **ALTER TABLE** – изменение структуры таблицы (добавление/удаление столбцов таблицы, изменение размеров столбцов, изменение типа данных).

RENAME TABLE – переименование, **DESCRIBE** – информация о структуре таблицы, **USE db_name** – установка текущей БД для последующих операторов.

UPDATE – обновление таблицы, **DELETE** – удаление данных таблицы, **DROP** – удаление таблицы, процедуры, представления, индекса из базы данных.

INSERT – добавление записей в таблицу. **INSERT ... VALUES** – вставка одной строки в таблицу, **INSERT ... SELECT** – внесение нескольких строк в таблицу, полученных в результате выполнения оператора **SELECT** (ограничение: целевая таблица команды **INSERT** не должна появляться в утверждении **FROM** части **SELECT**).

2

Работа с переменными в MySQL

Пользовательские переменные в MySQL.

: = определение переменной, @ первый символ имени.
Нельзя определять и использовать переменную в одном запросе.

Запрос: Вывести фамилию терапевта с максимальным стажем.

```
Select @MaSt := Max (Doc.Standing)
From Spec Inner Join Doc
On Spec.CodeS = Doc.CodeS
Where ( Spec.Spec like 'Терапевт');
```

```
Select Surname, Standing
From Spec Inner Join Doc
On Spec.CodeS = Doc.CodeS
Where (( Spec.Spec like 'Терапевт') And
        (Standing = @MaSt));
```

4

Подчиненные запросы

Подчиненный запрос — инструкция SELECT, помещенная в другую, внешнюю инструкцию или в другой подчиненный запрос. Подчиненный запрос может использоваться вместо выражения в списке полей инструкции **SELECT**, **FROM**, **WHERE**, **HAVING**, **SELECT...INTO**, **INSERT**, **DELETE**, **UPDATE**

Подзапрос может возвращать **скалярную величину**, **одну строку**, **один столбец** или **таблицу**.

Синтаксис — три формы:

1. Сравнение [ANY | ALL | SOME] (инструкция_sql)
2. [NOT] EXISTS (инструкция_sql)
3. выражение [NOT] IN (инструкция_sql)

5

Подзапрос возвращает скалярную величину

Пример 2. Есть упрощенный вариант двух таблиц.

Спец { КодС, Спец }

Врач { КодВ, КодС, Фам, Стаж }

Запрос на объединение двух таблиц

```
SELECT Спец.Спец, Врачи.Фам, Врачи.Стаж
FROM Спец INNER JOIN Врачи
ON Спец.КодС = Врачи.КодС;
```

Результат сохраним,
как Врач_ст

Спец	Фамилия	Стаж
Хирург	Попов	20
Терапевт	Котова	16
Хирург	Петров	18
Терапевт	Петрова	10
Невропатолог	Сидоров	10
Невропатолог	Иванова	15

7

Подзапрос возвращает скалярную величину

<скалярная форма> <оператор> <подзапрос>,

<подзапрос> <оператор> <скалярное выражение> — **ошибка**

Пример 1. Фамилия, имя и адрес самого **старого** пациента:

```
Select Фам & " " & Left (Имя,1) & "." As ФИО,
        Адрес, ДатаРож           From Пац
Where (ДатаРож =
        ( Select Min (ДатаРож) From Пац ));
```

6

Подзапрос возвращает скалярную величину

Задача. Указать специализации и средний стаж для тех специализаций, у которых средний стаж больше, чем средний стаж по всем специализациям

```
SELECT Спец, Avg (Стаж) as ПоСпец
FROM Врач_ст GROUP BY Спец
HAVING (Avg (Стаж) >=
        (SELECT Avg (Стаж) as ПоВсем FROM Врач_ст ));
```

Спец	СтажПоСпец
хирург	19

Проверка правильности выполнения запроса.

1. Средний стаж по всем

```
SELECT Avg (Стаж) as ПоВсем
FROM Врач_ст
```

СтажПоВсем
14.8333333333333

8

Подзапрос возвращает скалярную величину

2. Средний стаж по каждой специализации

```
SELECT Спец, Avg (Стаж) AS ПоСпец
FROM Врач_ст GROUP BY Спец;
```

Запрос2 : запрос на выборку

Спец	СтажПоСпец
невропатолог	12.5
терапевт	13
хирург	19

Средний стаж по всем был

Запрос2 : запрос на вы

СтажПоВсем
14.8333333333333

Результат
правильный

запроса

СредСтажВрачей : запрос на в

Спец	СтажПоСпец
хирург	19

9

Подзапрос возвращает один столбец

С помощью предикатов ANY или SOME, которые являются синонимами, извлекаются записи в главном запросе, удовлетворяющие сравнению с любыми записями, извлеченными в подчиненном запросе.

Задача. Есть две таблицы:

Товар2

КодТ	Товар	ЦенаПр
1	Утюг	300.00грн.
2	Пылесос	500.00грн.
3	Фен	100.00грн.
4	Кофемолка	150.00грн.
5	Вытяжка	1200.00грн.

Продажа2

КодПр	КодТ	ЦенаПрод	Скидка
1	1	270.00грн.	10
2	1	240.00грн.	20
3	2	450.00грн.	10
4	3	75.00грн.	25
5	4	135.00грн.	10
6	5	1140.00грн.	5
7	1	255.00грн.	15

10

Подзапрос возвращает один столбец

Запрос. Вывести названия и ЦенуПриема тех товаров, у которых ЦенаПродажи со скидкой $\geq 10\%$.

Вариант 1.(плохой)

```
SELECT Товар.Товар, Товар.ЦенаПр,
Продажа.ЦенаПрод, Продажа.Скидка
FROM Продажа INNER JOIN Товар ON
Продажа.КодТ = Товар.КодТ
WHERE (Продажа.Скидка  $\geq 10$ );
```

Результат

Товары10

Товар	ЦенаПр	ЦенаПрод	Скидка
Утюг	300.00грн.	270.00грн.	10
Утюг	300.00грн.	240.00грн.	20
Пылесос	500.00грн.	450.00грн.	10
Фен	100.00грн.	75.00грн.	25
Кофемолка	150.00грн.	135.00грн.	10
Утюг	300.00грн.	255.00грн.	15

11

Подзапрос возвращает один столбец

Вариант 2. (с ANY – не правильный)

```
SELECT * FROM Товар
WHERE ( ЦенаПр >
ANY (SELECT ЦенаПрод FROM Продажа
WHERE Скидка $\geq 10$  ) );
```

Результат

ТоварAny

КодТ	Товар	ЦенаПр
1	Утюг	300.00грн.
2	Пылесос	500.00грн.
3	Фен	100.00грн.
4	Кофемолка	150.00грн.
5	Вытяжка	1200.00грн.

Почему? Ведь вытяжку продали со скидкой 5% !

12

Подзапрос возвращает один столбец

Вариант 2. Проверка правильности выполнения запроса ANY.

Шаг 1. Подзапрос **SELECT ЦенаПрод FROM Продажа**

WHERE Скидка>=10;

Результат Шаг 2. ANY . ЦенаПриема должна быть > **хоть какой-то** ЦеныПродажи

Запрос3	ЦенаПрод	Товар	ЦенаПр
	270.00грн.	Утюг	300.00грн.
	240.00грн.	Пылесос	500.00грн.
	450.00грн.	Фен	100.00грн.
	75.00грн.	Кофемолка	150.00грн.
	135.00грн.	Вытяжка	1 200.00грн.
	255.00грн.		

Подзапрос возвращает один столбец

Вариант 3. (с ALL – не правильный)

SELECT * FROM Товар

WHERE (ЦенаПр >

ALL (SELECT ЦенаПрод FROM Продажа

WHERE Скидка>=10));

Результат

ТоварAll	КодТ	Товар	ЦенаПр
	2	Пылесос	500.00грн.
	5	Вытяжка	1 200.00грн.

Почему? Есть еще товар, проданный со скидкой >=10% !

14

Подзапрос возвращает один столбец

Вариант 3. Проверка правильности выполнения запроса ALL.

Шаг 1. Подзапрос **SELECT ЦенаПрод FROM Продажа**

WHERE Скидка>=10;

Результат Шаг 2. ALL . ЦенаПриема должна быть > **всех** ЦеныПродажи

ЦенаПр > ALL столбец Запрос 3

Запрос3	ЦенаПрод	Товар	ЦенаПр
	270.00грн.	Утюг	300.00грн.
	240.00грн.	Пылесос	500.00грн.
	450.00грн.	Фен	100.00грн.
	75.00грн.	Кофемолка	150.00грн.
	135.00грн.	Вытяжка	1 200.00грн.
	255.00грн.		

Подзапрос возвращает множество значений

Вариант 4. **SELECT Товар, ЦенаПр FROM Товар**

WHERE (КодТ in

(SELECT distinct КодТ FROM Продажа

WHERE (Скидка>=10)));

Результат
правильный

ТоварПодз10	Товар	ЦенаПр
	Утюг	300.00грн.
	Пылесос	500.00грн.
	Фен	100.00грн.
	Кофемолка	150.00грн.

15

16

Подзапрос возвращает множество значений

Вариант 4. Проверка правильности выполнения запроса с in

Шаг 1. **SELECT КодТ FROM Продажа**
WHERE (Скидка>=10));

Результат

КодТ
1
1
2
3
4
1

Шаг 2. Добавим в подзапрос **distinct**

Результат:
уникальные
коды товаров

КодТ
1
2
3
4

Шаг 3. В таблице Товар найти Названия и ЦенуПродажи для тех товаров, коды которых в подзапросе.

КодТ	Товар	ЦенаПр
1	Утюг	300.
2	Пылесос	500.
3	Фен	100.
4	Кофемолка	150.
5	Вытяжка	1200.

Предложение EXISTS

Предложение EXISTS (с необязательным зарезервированным словом NOT) используется в сравнениях для определения существования записей, возвращаемых в подчиненном запросе. Предложение Exists позволяет выбрать из одного столбика множество строк и возвращает True, если записи существуют и false если не существует записей.

Задача. Таблицы Пр1 и Пр2 с информацией о сотрудниках, участвующих в Проекте 1 и в Проекте2.

Пр1 : таблица			
Номер	Имя	Отд	
11	Иванов	310	
12	Петров	310	
13	Сидоров	313	
14	Федоров	310	
15	Иванова	315	
19	Ивкин	315	

Пр2 : таблица			
Номер	Имя	Отд	
11	Иванов	310	
12	Петров	310	
16	Сидоренко	313	
17	Федоренко	310	
18	Иваненко	315	
19	Ивкин	315	

19

Простой запрос

Можно ли исправить запрос из варианта 1?

SELECT Товар.Товар, Товар.ЦенаПр,
Продажа.ЦенаПрод, Продажа.Скидка
FROM Продажа INNER JOIN Товар ON Продажа.КодТ
= Товар.КодТ
WHERE (Продажа.Скидка >=10);

Добавим **distinct.**
 – результат такой же

Товар	ЦенаПр	ЦенаПрод	Скидка
Кофемолка	150.00грн.	135.00грн.	10
Пылесос	500.00грн.	450.00грн.	10
Утюг	300.00грн.	240.00грн.	20
Утюг	300.00грн.	255.00грн.	15
Утюг	300.00грн.	270.00грн.	10
Фен	100.00грн.	75.00грн.	25

18

Предложение EXISTS

Запрос. Если есть сотрудники указанного отдела, которые участвуют в двух проектах, то вывести их номера и имена.

SELECT Пр1.Номер, Пр1.Имя,
Пр1.Отд

FROM Пр1,Пр2

WHERE ((Пр1.Отд=310) проверка отдела

and

(Пр1.Номер= Пр2.Номер)

наличие в двух проектах

and exists

(SELECT Пр2.Номер FROM если во втором проекте
Пр2 вообще есть люди из

WHERE (Пр2.Отд=310)));

указанного отдела

Результат

ууу : запрос на выборку			
Номер	Имя	Отд	
11	Иванов	310	
12	Петров	310	

20

Пример решения задачи

Задача. Выдать сведения о пациентах, у кого уже прошел год и более с даты последнего ФЛЮ

ПУСТЬ СЕГОДНЯ 12.11.2013

Пац : таблица		Флю : таблица			
КодПац	Фам	КодФлю	КодПац	ДатаФлю	Рез
1	Соколов	1	1	15.11.2012	ОК
2	Дедов	2	2	10.11.2012	ОК
3	Белов	3	5	12.05.2013	ОК
4	Шариков				
5	Михайлов				
Запись: 1		Запись: 1 из 3			

21

Пример решения задачи

Алгоритм решения задачи:

1. Определить коды благополучных пациентов (разница < 365).
2. Определить ВСЕХ пациентов.
3. Из всех вычесть благополучных

Шаг первый: Определение благополучных пациентов

```
SELECT КодПац FROM флю
WHERE (int(NOW()) -
int(ДатаФлю) <= 365);
```

Есть Флю...	
КодПац	
1	
5	
*	0

Шаг второй: Определение всех пациентов и Дат Флю

```
SELECT Пац.фам, флю.ДатаФлю
FROM Пац LEFT JOIN флю ON
Пац.КодПац = флю.КодПац
ORDER BY флю.ДатаФлю DESC;
```

Все Пац : запро...	
Фам	ДатаФлю
Михайлов	12.05.2013
Соколов	15.11.2012
Дедов	10.11.2012
Шариков	
Белов	
*	

22

Пример решения задачи

Шаг третий: Вычитание. Нужно учесть и тех, кто ни делал ФЛЮ вообще никогда.

```
SELECT Пац.фам, флю.ДатаФлю
FROM Пац LEFT JOIN флю ON
Пац.КодПац = флю.КодПац
WHERE ( (флю.КодПац
NOT IN
(SELECT флю.КодПац FROM флю
WHERE (int(NOW()) -
int(ДатаФлю) <= 365) ) )
OR (флю.КодПац IS NULL)
)
```

ORDER BY флю.ДатаФлю DESC;

12.11.2013

Без Флю : запро...	
Фам	ДатаФлю
Дедов	10.11.2012
Шариков	
Белов	
*	
Запись: 1	

23

Пример решения задачи

Предположим Соколову (КодПац=1) не повезло, он заболел и делал ФЛЮ несколько раз. Дополним таблицу ФЛЮ еще двумя записями.

КодФл	КодПа	ДатаФлю	Рез
1	1	15.11.2012	ОК
2	2	10.11.2012	ОК
3	5	12.05.2013	ОК
4	1	01.10.2012	ПН
5	1	05.12.2012	ОК

Запрос на поиск благополучных пациентов дает результат (Соколов с КодПац =1 и Михайлов с КодПац=5)

Есть Флю...	
КодПац	
1	
5	
1	
*	0

Окончательный запрос использует вычитание. Ищем тех, коды которых не находятся во множестве {1,5,1}. Повторение одного и того же кода 2 раза или более не повлияет на правильность итогового запроса.

12.11.2013

24

Пример решения задачи

Предположим Дедов (КодПац=2) делал ФЛЮ несколько раз.

Дополним таблицу ФЛЮ еще одной записью.

КодФл	КодПа	ДатаФлю	Рез
1	1	15.11.2012	ОК
2	2	10.11.2012	ОК
3	5	12.05.2013	ОК
4	1	01.10.2012	ПН
5	1	05.12.2012	ОК
6	2	11.11.2011	ОК

По каждой из дат Дедов попадает в неблагоприятные пациенты. Запрос на вычитание в этом случае выглядит так:

Фам	ДатаФлю
Дедов	10.11.2012
Дедов	11.11.2011
Шариков	
Белов	
*	

Такой результат НЕ КОРРЕКТЕН. На большой выборке получаем засоренный, не очевидный результат. Неизвестна дата последней ФЛЮ.

12.11.2013

25

Пример решения задачи

Если в запросе убрать **LEFT JOIN**, заменив его на **Inner JOIN** в результирующей выборке будут отсутствовать Белов и Шариков.

Фам	ДатаФлю
Дедов	10.11.2012
Соколов	01.10.2012
Дедов	11.11.2011
*	

27

Пример решения задачи

Варианты решения.

Обойтись без вычитания, искать тех, у кого количество дней > 365 и тех, у кого поле IS NULL?

```
SELECT Фам, ДатаФлю
FROM Пац LEFT JOIN флю
ON
Пац. КодПац=флю. КодПац
WHERE
```

```
((int(NOW()) -
int(ДатаФлю) > 365)
Or
(флю.КодПац Is Null));
```

Фам	ДатаФлю
Соколов	01.10.2012
Дедов	11.11.2011
Дедов	10.11.2012
Белов	
Шариков	
*	

Результат НЕ ВЕРЕН. Соколов благополучный пациент. Он попал в итоговую выборку, т.к. делал ФЛЮ несколько раз.

26

Решение задачи

1. Вместо таблицы ФЛЮ использовать представление (сохраненный запрос) с максимальными датами ФЛЮ
2. Использовать соотнесенный запрос.

28

Представления

Представления – определение **виртуальной таблицы** в ANSI.

В Access это сохраненный запрос, образованный оператором SELECT. Представление реализуется при выполнении. В базовых таблицах данные сохраняются физически. В представлении **данные не сохраняются физически**, а только возвращаются. В запросах можно обратиться к представлению.

```
CREATE
    [OR REPLACE]
    VIEW view_name [(column_list)]
    AS select_statement
```

29

Выполнение раздела FROM с представлением

```
SELECT фам, МахДата
FROM Пац LEFT JOIN МахФлюДляВсех
ON Пац.КодПац = МахФлюДляВсех.КодПац
WHERE ((int(NOW())-int(МахДата)>365)
Or (МахФлюДляВсех.КодПац Is Null));
```

1. Определение источника записей в разделе **FROM**:

- поиск внешних ссылок, первая ссылка **Пац**, поиск среди базовых таблиц, ссылка найдена, создание виртуальной таблицы-копии **Пац**;
- вторая ссылка **МахФлюДляВсех**, поиск среди базовых таблиц, ссылка не найдена. Поиск среди представлений. Ссылка найдена. Выполнение запроса из **МахФлюДляВсех**, создание виртуальной таблицы-результата выполнения **МахФлюДляВсех**;
- создание виртуальной таблицы расширенного декартова произведения двух виртуальных таблиц **Пац** и **МахФлюДляВсех**;
- выполнение операции **LEFT JOIN**.

31

Пример решения задачи

Выдать сведения о пациентах, у кого уже прошел год и более с даты последнего ФЛЮ.

Вариант решения 1. Создать представление **МахФлюДляВсех**.

```
SELECT КодПац, max(ДатаФлю) as МахДата
FROM флю
GROUP BY КодПац;
```

В запросе с проверкой на >365 дней используем представление.

```
SELECT фам, МахДата
FROM Пац LEFT JOIN МахФлюДляВсех
ON Пац.КодПац = МахФлюДляВсех.КодПац
WHERE
((int(NOW())-int(МахДата)>365)
Or (МахФлюДляВсех.КодПац Is Null));
```

30

Пример решения задачи 1

Вариант решения 2. Пациенты, у кого прошел год и более с даты последнего ФЛЮ (использование подчиненного запроса)

```
SELECT фам, Запрос.МахДата
FROM Пац LEFT JOIN
(SELECT КодПац, max(ДатаФлю) as МахДата
FROM флю GROUP BY КодПац) as Запрос
ON Пац.КодПац = Запрос.КодПац
WHERE
((int(NOW())-int(Запрос.МахДата)>365)
Or (Запрос.КодПац Is Null));
```

32

Представления и запросы с подзапросами.

Когда применять сложные подчиненные, а когда лучше представления?

1. Лучше использовать сложные запросы, включающие в себя подзапросы, если уровень вложенности не больше 3, иначе запросы становятся нечитаемыми и трудно отлаживаются.
2. Если уровень вложенности больше 3 целесообразно использовать представления (сохраненные запросы в Access)
3. Отлаживать и проверять сложные запросы нужно на большой выборке по шагам.

33

Пример решения задачи

Алгоритм решения: определить макс. балл в каждой группе, потом по баллу – фамилию студента.

Шаг 1. Определить максимальный балл в каждой группе.

```
SELECT Stud.Group, Max (Stud.Ball) AS MaxBall  
FROM Stud  
GROUP BY Stud.Group;
```

Результат выполнения
запроса

Group	MaxBall
КС-31	6,7
КС-32	8
КС-33	6,8

Пример решения задачи

Задача 1. По результатам компьютерного тестирования, проведенного среди студентов разных групп установить фамилии студентов, занявших первые места в своих группах:

Group	Fio	Ball
КС-31	Сорокин	6.7
КС-31	семенов	6.6
КС-32	Свистов	6.7
КС-32	Малышев	8
КС-33	Бендок	6.8
КС-33	Миронов	6.3

Пример решения задачи

Шаг 2. Определить фамилии тех студентов, кто имеет максимальный балл в группе:

```
SELECT Stud.Group, Stud.Fio,  
Stud.Ball FROM Stud  
WHERE (Stud.Ball in  
(SELECT Max(Stud.Ball) AS MaxBall  
FROM Stud GROUP BY Stud.Group ))  
ORDER BY Stud.Group;
```

Результат:

Запрос5 : запрос на выборку			
	Group	Fio	Ball
►	КС-31	Сорокин	6.7
	КС-32	Свистов	6.7
	КС-32	Малышев	8
	КС-33	Бендок	6.8
*			0

Пример решения задачи

Почему у нас не три, а больше человек?

Дело в том, что максимальный балл в 31 группе=6.7 имеет человек из 32 группы.

Каким образом составить запрос, который давал бы правильные результаты даже в этом, не стандартном и нетривиальном случае?

Решение. Использовать запрос с зависимым подзапросом.

Как? Указать два источника данных.

Первый – таблица **Stud**, второй – подзапрос по определению максимального балла в каждой группе. Связать их между собой по группе.

Пример решения задачи

```
SELECT Stud.Group, Stud.Fio, Stud.Ball
FROM Stud, (SELECT Stud.Group,
                  Max(Stud.Ball) AS MaxBall
             FROM Stud GROUP BY Stud.Group ) AS Zapr1
WHERE ((Stud.Group = Zapr1.Group) and
      (Stud.Ball = Zapr1.MaxBall))
ORDER BY Stud.Group;
```

Результат:



Group	Fio	Ball
к-31	Сорокин	6.7
к-32	Малышев	8
к-33	Бендок	6.8

Пример «ВрачиПациенты»

Задача. Используя таблицы БД «Поликлиника» вывести фамилию доктора (ов) , у которого минимальное количество пациентов, посетивших прием в указанную дату.

Шаг 1. Определить количество принятых пациентов каждым врачом в указанную дату (сохраненный запрос CodeFamCount)

```
SELECT врачи.кодВр as Cod, врачи.фам as Fam,
       count(Прием.код_пац) AS PacDoct
FROM врачи INNER JOIN Прием
ON врачи.кодВр = Прием.код_вр
WHERE (Прием.дата =[Dat])
GROUP BY врачи.кодВр, врачи.фам;
```

Особенность (фишка): т.к. **врачи.кодВр** и **врачи.фам** определяют одного и того же человека, группировка вначале по **врачи.кодВр**, а потом по **врачи.фам** не нарушает групп и позволяет вывести фамилию. Группировка в обратном порядке не верна. Почему?

Пример «ВрачиПациенты»

Шаг 2. На основании CodeFamCount определить минимальное количество принятых пациентов.

```
SELECT Min(CodeFamCount.PacDoct) AS MinPacDoct
FROM CodeFamCount;
```

Этот шаг используется для правильного построения конечного запроса и для проверки одновременно.

Пример «ВрачиПациенты»

Шаг 3. Вариант 1. Конечный запрос, реализованный через сохраненный запрос (представление) – Указать фамилию доктора, у которого минимальное количество принятых пациентов в указанную дату (докторов может быть несколько).

```
SELECT  Fam, PacDoct
FROM CodeFamCount
WHERE (PacDoct =
      (SELECT
        Min(CodeFamCount.PacDoct) AS MinPacDoct
      FROM CodeFamCount
      )
      );
```

42

Пример «ВрачиПациенты»

Вариант 2. Конечный запрос – один запрос без представлений.

```
SELECT Cod, Fam, PacDoct
FROM
  ( SELECT врачи.кодВр AS Cod,
        врачи.фам AS Fam,
        Count(Прием.код_пац) AS PacDoct
    FROM врачи INNER JOIN Прием
        ON врачи.кодВр = Прием.код_вр
    WHERE (Прием.дата=[Dat])
    GROUP BY врачи.кодВр, врачи.фам
  ) as CodeFamCount
WHERE (PacDoct =
      (SELECT Min(CodeFamCount.PacDoct) AS MinPacDoct
      FROM CodeFamCount
      )
      );
```

43