Лекція 12-2020_2021

- На попередній лекції
- Змінні в в MySQL
- Підлеглі запити

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

Что было в предыдущей лекции

Access: INSERT ... VALUES. Поле счетчика **HE** 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.

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).

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 'Tepaneвт'); 

Select Surname, Standing 

From Spec Inner Join Doc 

On Spec.CodeS = Doc.CodeS 

Where ((Spec.Spec like 'Tepaneвт') And 

(Standing = @MaSt));
```

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)

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

ОМ 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

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));

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 ВУ Спец

HAVING (Avg(CTax)>=

5

7

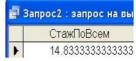
(SELECT Avg(CTax) as NoBcem FROM Bpau ct));

靈	СредСтажВра	чей : запрос на
	Спец	СтажПоСпец
-	хирург	19

Проверка правильности выполнения запроса.

1.Средний стаж по всем

SELECT Avg(CTam) as ПоВсем FROM Врач СТ



Подзапрос возвращает скалярную величину

2. Средний стаж по каждой специализации SELECT Спец, Avg (Стаж) AS ПоСпец FROM Врач ст GROUP BY Спец;



Средний стаж по всем был

₫ За	прос2 : запрос на вы
	СтажПоВсем
-	14.83333333333333

Результат правильный

запроса

ē	📴 СредСтажВрачей ; запрос на і				
	Спец	СтажПоСпец			
-	хирург	19			

Подзапрос возвращает один столбец

<u>Запрос</u>. Вывести названия и ЦенуПриема тех товаров, у которых ЦенаПродажи со скидкой >= 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 >=10);

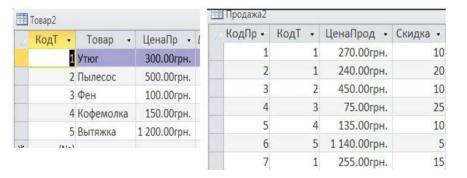
Результат

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

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:



Подзапрос возвращает один столбец

Вариант 2. (c ANY – не правильный)

SELECT * FROM Tomap

WHERE (ЦенаПр >

ANY (SELECT ЦенаПрод FROM Продажа

WHERE Скидка>=10));

Результат

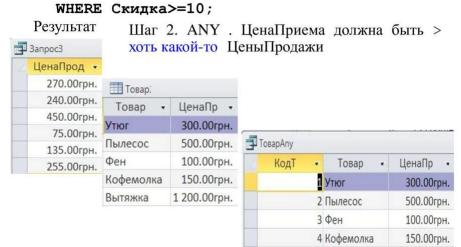
11

КодТ	*	Товар •	ЦенаПр •
	1	Утюг	300.00грн
	2	Пылесос	500.00грн
	3	Фен	100.00грн
	4	Кофемолка	150.00грн
	5	Вытяжка	1 200.00грн

Почему? Ведь вытяжку продали со скидкой 5%!

Подзапрос возвращает один столбец

Вариант 2. Проверка правильности выполнения запроса ANY. Шаг 1. Подзапрос SELECT ЦенаПрод FROM Продажа



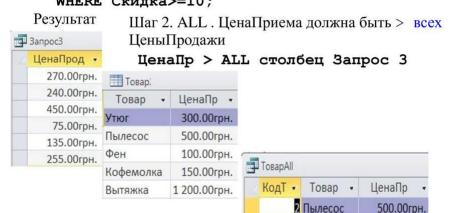
Подзапрос возвращает один столбец

5 Вытяжка

2 Пылесос

5 Вытяжка

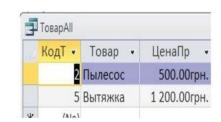
Вариант 3. Проверка правильности выполнения запроса ALL. Шаг 1. Подзапрос SELECT ЦенаПрод FROM Продажа WHERE Скидка>=10;



Подзапрос возвращает один столбец

Вариант 3. (с ALL – не правильный) SELECT * FROM Tobap WHERE (ЦенаПр > FROM Продажа ALL (SELECT ЦенаПрод WHERE Скидка>=10));

Результат



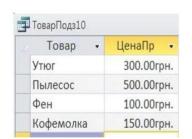
Почему? Есть еще товар, проданный со скидкой >=10%!

14

Подзапрос возвращает множество значений

Вариант 4. SELECT Товар, ЦенаПр FROM Товар WHERE (KogT in (SELECT distinct КодТ FROM Продажа WHERE (Скидка>=10)));

Результат правильный



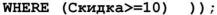
15

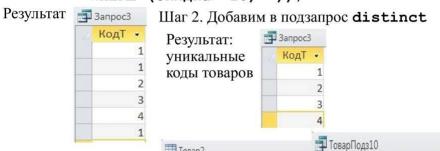
1 200.00грн.

1 200.00грн.

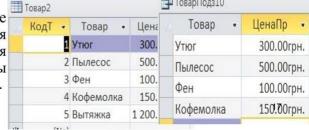
Подзапрос возвращает множество значений

Вариант 4. Проверка правильности выполнения запроса с in Шаг 1. SELECT КодТ FROM Продажа





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



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 Проекте 2.

	Пр1 : та	блица		=	Пр2 : таб	лица	
	Номер	Имя	Отд		Номер	Имя	Отд
	11	Иванов	310		11	Иванов	310
۲	12	Петров	310	-	12	Петров	310
	13	Сидоров	313		16	Сидоренко	313
	14	Федоров	310		17	Федоренко	310
	15	Иванова	315		18	Иваненко	315
	19	Ивкин	315		19	Ивкин	315

Простой запрос

Можно ли исправить запрос из варианта 1?

Товар. Товар, SELECT Товар. ЦенаПр, Продажа. ЦенаПрод, Продажа. Скидка

FROM Продажа INNER JOIN Товар ОN 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 которые отдела, участвуют в двух проектах, то вывести их номера и имена.

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

Результат

SELECT Np1. Homep, Np1. Mmg, Пр1.Отд

FROM Np1, Np2

((Пр1.Отд=310) проверка отдела WHERE

and

 $(\Pi p1.Homep=\Pi p2.Homep)$

наличие в двух проектах

and exists

(SELECT Пр2. Homep Пр2

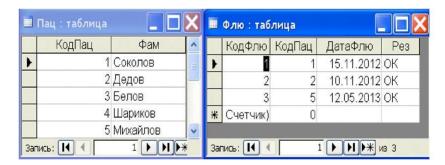
FROM если во втором проекте вообще есть люди из указанного отдела

(Пр2.Отд=310))); WHERE

Пример решения задачи

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

ПУСТЬ СЕГОДНЯ 12.11.2013



21

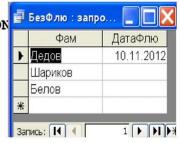
23

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



Пример решения задачи

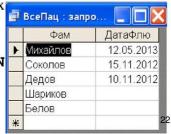
Алгоритм решения задачи:

- 1.Определить коды благополучных пациентов (разница < 365).
- 2. Определить ВСЕХ пациентов.
- 3. Из всех вычесть благополучных <u>Шаг первый</u>: Определение благополучных пациентов

SELECT ΚομΠαμ FROM Φπю WHERE (int(NOW()) - int(ДатаФπю) <= 365);

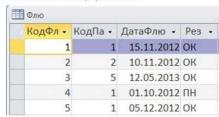


Шаг второй: Определение всех пациентов и ДатыФлю SELECT Пац.Фам, Флю.ДатаФлю FROM Пац LEFT JOIN Флю ON Пац.КодПац = Флю.КодПац ORDER BY Флю.ДатаФлю DESC;



Пример решения задачи

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



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



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.

Пример решения задачи

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

Дополним таблицу ФЛЮ еще одной записью.

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

	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
ē	БезФлю Фам	-	на выб 🔃 [атаФлю	
•	-	и Д		
<u>-</u>	Фам	и Д	атаФлю	
<u>-</u>	Фам Дедов	л <u>Д</u>	атаФлю 10.11.2012	
→	Фам Дедов Дедов	л <u>Д</u>	атаФлю 10.11.2012	

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

12.11.2013

Пример решения задачи

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

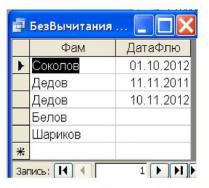
🗐 БезВычитания 🔲 🔲 🔀			
	Фам	ДатаФлю	
•	Дедов	10.11.2012	
	Соколов	01.10.2012	
	Дедов	11.11.2011	
*			
Заг	пись: 🔣 🕕	1	

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));



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.

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
```

2

Выполнение раздела FROM с представлением

SELECT Фам, МахДата
FROM Пац LEFT JOIN МахФлюДляВсех
ON Пац.КодПац = МахФлюДляВсех.КодПац
WHERE ((int(NOW())-int(MaxДата)>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**.

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 MaxДата 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(MaxДата)>365)

Or (МахФлюДляВсех.КодПац Is Null));

(

Пример решения задачи 1

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

```
SELECT Фам, Запрос.МажДата
FROM Пац LEFT JOIN
(SELECT КодПац, маж (Датафлю) аз МажДата
FROM Флю GROUP BY КодПац) аз Запрос
ON Пац.КодПац = Запрос.КодПац
WHERE
((int(NOW())-int(Запрос.МажДата)>365)
Or (Запрос.КодПац Is Null));
```

Представления и запросы с подзапросами.

Когда применять сложные подчиненные, а когда лучше представления?

- 1. Лучше использовать запросы, сложные включающие в себя подзапросы, если уровень вложенности не больше 3, иначе 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.

Пример решения задачи

Алгоритм решения: определить макс. балл в каждой группе, потом по баллу – фамилию студента. <u>Шаг 1</u>. 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

группа

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:

\overline 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 <u>больше</u> 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?

<u>Решение</u>. 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 «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 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:

Ē F	irstPlaces :	запрос на	
	Group	Fio	Ball
•	c-31	Сорокин	6.7
H	с-32	Малышев	8
H	кс-33	Бендок	6.8

Пример «ВрачиПациенты»

 $\underline{\text{Шаг 2}}$. 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»

<u>Шаг 3</u>. <u>Вариант 1</u>. 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
) );
```

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
```