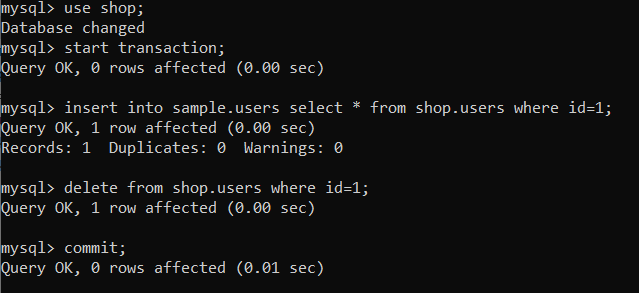
Практическое задание по теме “Транзакции, переменные, представления”

1. В базе данных shop и sample присутствуют одни и те же таблицы, учебной базы данных. Переместите запись id = 1 из таблицы shop.users в таблицу sample.users. Используйте транзакции.



1. Создайте представление, которое выводит название name товарной позиции из таблицы products и соответствующее название каталога name из таблицы catalogs.

**create** **view** v\_products **as**

**select**

p.name **as** product\_name,

c.name **as** catalog\_name

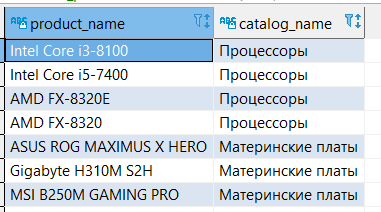
**from**

products p

**join** catalogs c **on** c.id = p.catalog\_id

;

**select** \* **from** v\_products;

****

1. по желанию) Пусть имеется таблица с календарным полем created\_at. В ней размещены разряженые календарные записи за август 2018 года '2018-08-01', '2016-08-04', '2018-08-16' и 2018-08-17. Составьте запрос, который выводит полный список дат за август, выставляя в соседнем поле значение 1, если дата присутствует в исходном таблице и 0, если она отсутствует.

**create** **temporary** **table** temp\_dates(

created\_at **varchar**(10) **not** **null** -- дата в символьном виде

);

**insert** **into** temp\_dates(created\_at)

**values**

('2018-08-01'),

('2018-08-04'),

('2018-08-16'),

('2018-08-17')

;

**select**

august.august\_date,

(**case** **when** temp\_dates.created\_at **is** **null** **then** 0 **else** 1 **end**) is\_exists

**from** (

**select** **adddate**('2018-08-01', @num:=@num+1) august\_date

**from**

messages, -- таблица должна иметь более 31 строк

(**select** @num:=-1) num

**limit** 31

) august

**left** **join** temp\_dates **on** str\_to\_date(created\_at, '%Y-%m-%d') = august.august\_date

;

1. (по желанию) Пусть имеется любая таблица с календарным полем created\_at. Создайте запрос, который удаляет устаревшие записи из таблицы, оставляя только 5 самых свежих записей.

-- испольузуем многотабличный DELETE

**delete** smth\_with\_date

**from** smth\_with\_date

**left** **join** (

**select**

created\_at

**from**

smth\_with\_date

**order** **by** created\_at **desc**

**limit** 5

) t2

**on** smth\_with\_date.created\_at = t2.created\_at

**where** t2.created\_at **is** **null**

;

Практическое задание по теме “Администрирование MySQL” (эта тема изучается по вашему желанию)

1. Создайте двух пользователей которые имеют доступ к базе данных shop. Первому пользователю shop\_read должны быть доступны только запросы на чтение данных, второму пользователю shop — любые операции в пределах базы данных shop.

**CREATE** **USER** shop\_read identified **WITH** sha256\_password **BY** 'pass';

**CREATE** **USER** shop identified **WITH** sha256\_password **BY** 'pass';

**COMMIT**;

**GRANT** **SELECT** **ON** shop.\* **TO** 'shop\_read'@'%';

**GRANT** **ALL** **ON** shop.\* **TO** 'shop'@'%';

1. (по желанию) Пусть имеется таблица accounts содержащая три столбца id, name, password, содержащие первичный ключ, имя пользователя и его пароль. Создайте представление username таблицы accounts, предоставляющий доступ к столбца id и name. Создайте пользователя user\_read, который бы не имел доступа к таблице accounts, однако, мог бы извлекать записи из представления username.

**CREATE** **VIEW** username **AS** **SELECT** id, name **FROM** accounts;

**CREATE** **USER** user\_read identified **WITH** sha256\_password **BY** 'pass';

**GRANT** **SELECT** **ON** username **TO** 'user\_read'@'%';

Практическое задание по теме “Хранимые процедуры и функции, триггеры"

1. Создайте хранимую функцию hello(), которая будет возвращать приветствие, в зависимости от текущего времени суток. С 6:00 до 12:00 функция должна возвращать фразу "Доброе утро", с 12:00 до 18:00 функция должна возвращать фразу "Добрый день", с 18:00 до 00:00 — "Добрый вечер", с 00:00 до 6:00 — "Доброй ночи".

**use** shop;

**DROP** **FUNCTION** **IF** **EXISTS** hello;

**DELIMITER //**

**CREATE** **FUNCTION** hello ()

**RETURNS** **TEXT** **NO** **SQL**

**begin**

**declare** v\_ret **text**;

**declare** v\_time **time** **DEFAULT** **CURTIME**();

**IF** v\_time **BETWEEN** '06:00:00' **AND** '12:00:00' **THEN**

**set** v\_ret = 'Доброе утро';

**ELSEIF** v\_time **BETWEEN** '12:00:00' **AND** '18:00:00' **THEN**

**set** v\_ret := 'Добрый день';

**ELSEIF** v\_time **BETWEEN** '18:00:00' **AND** '00:00:00' **THEN**

**set** v\_ret := 'Добрый вечер';

**ELSE**

**set** v\_ret := 'Доброй ночи';

**END** **IF**;

**return** v\_ret;

**END** //

**delimiter ;**

**select** hello();

1. В таблице products есть два текстовых поля: name с названием товара и description с его описанием. Допустимо присутствие обоих полей или одно из них. Ситуация, когда оба поля принимают неопределенное значение NULL неприемлема. Используя триггеры, добейтесь того, чтобы одно из этих полей или оба поля были заполнены. При попытке присвоить полям NULL-значение необходимо отменить операцию.

**delimiter //**

**CREATE** **TRIGGER** BI\_RejectNullsOnProducts **BEFORE** **INSERT** **ON** products

**FOR** **EACH** **ROW**

**BEGIN**

**IF**(**ISNULL**(**NEW**.name) **AND** **ISNULL**(**NEW**.description)) **THEN**

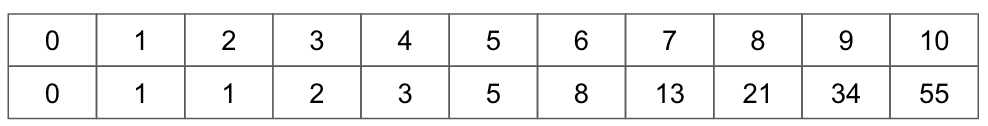
**SIGNAL** **SQLSTATE** '45000' **SET** MESSAGE\_TEXT = 'Попытка присвоить пустые значения и полю NAME и полю DESCRIPTION!';

**END** **IF**;

**END** //

**delimiter ;**

1. (по желанию) Напишите хранимую функцию для вычисления произвольного числа Фибоначчи. Числами Фибоначчи называется последовательность в которой число равно сумме двух предыдущих чисел. Вызов функции FIBONACCI(10) должен возвращать число 55.



**use** shop;

**DROP** **FUNCTION** **IF** **EXISTS** fibo;

**DELIMITER //**

**CREATE** **FUNCTION** fibo(p **int**)

**RETURNS** **int** **DETERMINISTIC**

**begin**

**declare** v\_ret **int**;

**declare** v1 **int** **default** 1;

**declare** v0 **int** **default** 0;

**declare** i **int** **default** 2;

**if** p = 0 **then**

**set** v\_ret = 0;

**elseif** p = 1 **then**

**set** v\_ret = 1;

**else**

**while** i <= p do

**set** v1 = v1 + v0;

**set** v0 = v1 - v0;

**set** i = i + 1;

**end** **while**;

**set** v\_ret = v1;

**end** **if**;

**return** v\_ret;

**END** //

**delimiter ;**

**select** fibo(10);