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.

**start transaction;**

**DELETE IGNORE FROM sample.users WHERE id=1;**

**insert into sample.users select \* from shop.users where shop.users.id = 1;**

**DELETE FROM shop.users WHERE id=1**

**commit**

**--на всякий случай удаляем запись из sample, если она вдруг существует**

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

**create or replace view cat as**

**select**

**p.name as Product,**

**c.name as Catalog**

**from**

**products as p,**

**catalogs as c**

**where**

**p.catalog\_id = c.id**

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 or replace view august as**

**SELECT DATE\_ADD('2018-08-01', INTERVAL Days DAY) dayA**

**FROM (select (t1\*10+t2) as days from**

**(SELECT 0 t1 UNION SELECT 1 UNION SELECT 2 UNION SELECT 3 UNION SELECT 4 UNION SELECT 5 UNION SELECT 6 UNION SELECT 7 UNION SELECT 8 UNION SELECT 9) t1,**

**(SELECT 0 t2 UNION SELECT 1 UNION SELECT 2 UNION SELECT 3 UNION SELECT 4 UNION SELECT 5 UNION SELECT 6 UNION SELECT 7 UNION SELECT 8 UNION SELECT 9) t2) as f**

**WHERE DATE\_ADD('2018-08-01', INTERVAL days DAY) < '2018-09-01' order by dayA;**

**select august.dayA, if (august.dayA in (select created\_at from test), '1', '0') as flag**

**from august**

**-- сборная солянка из различных решений из Интернета, которые адаптировал под себя**

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.

**delete from test**

**where created\_at not in**

**(select \* from**

**(select created\_at from test order by created\_at desc limit 5) as sec**

**)**

**--есть подозрение, что надо было использовать переменную, но как-то и без неё вроде нормально**

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'@'localhost' IDENTIFIED WITH sha256\_password BY 'shop\_read';**

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

**CREATE USER 'shop'@'localhost' IDENTIFIED WITH sha256\_password BY 'shop';**

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

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 table accounts (**

**id int unsigned auto\_increment primary key,**

**name varchar(255) not null,**

**password varchar(255)**

**);**

**insert into accounts values**

**(default, 'First','First'),**

**(default, 'Second','Second'),**

**(default, 'Third','Third'),**

**(default, 'Fourth','Fourth'),**

**(default, 'Fifth','Fifth');**

**create view username as select id, name from accounts**

**CREATE USER 'user\_read'@'localhost' IDENTIFIED WITH sha256\_password BY 'user\_read';**

**GRANT SELECT ON shop.username TO 'user\_read'@'localhost';**

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".

**DROP FUNCTION IF EXISTS hello;**

**delimiter !!**

**Create function hello()**

**returns varchar(20) deterministic**

**begin**

**set @T = time(now());**

**case**

**when (@T between '00:00:00' and '05:59:59') then set @answer='Доброй ночи';**

**when (@T between '06:00:00' and '11:59:59') then set @answer='Доброе утро';**

**when (@T between '12:00:00' and '17:59:59') then set @answer='Добрый день';**

**when (@T between '18:00:00' and '23:59:59') then set @answer='Добрый вечер';**

**else set @answer='Error';**

**end case;**

**return @answer;**

**end; !!**

**delimiter ;**

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 !!**

**DROP TRIGGER IF EXISTS check\_products !!**

**CREATE TRIGGER check\_products BEFORE INSERT ON products**

**FOR EACH ROW**

**BEGIN**

**DECLARE checkIN varchar(255);**

**if (new.name IS null) and (new.description is NULL) then**

**SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'INSERT canceled';**

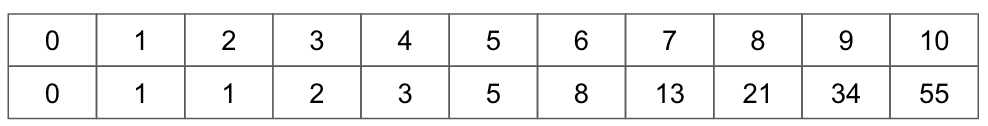
**end if;**

**END !!**

**delimiter ;**

**--аналогичное для UPDATE отличается только названием триггера и BEFORE UPDATE**

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.



**DROP FUNCTION IF EXISTS FIBONACCI;**

**delimiter !!**

**Create function FIBONACCI(num INT)**

**returns bigint deterministic**

**begin**

**declare result,res1,res2 bigint default 0;**

**declare sec int default 0;**

**while sec < num do**

**if sec=1 then**

**set res2=1;**

**end if;**

**set result = res2+res1;**

**set res1 = res2;**

**set res2 = result;**

**set sec = sec+1;**

**end while;**

**return result;**

**end; !!**

**delimiter ;**