

# 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.

Выполнил следующей командой:

```
START TRANSACTION;  
INSERT into sample.users (id, name) select id, name from shop.users s where s.id = 2;  
DELETE from shop.users s where s.id = 2;  
COMMIT;
```

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

```
CREATE VIEW prod_catalog AS  
SELECT  
p.name as prod_name,  
c.name as cat_name  
FROM products p  
left join catalogs c on c.id = p.catalog_id;
```

3. по желанию) Пусть имеется таблица с календарным полем created\_at. В ней размещены разряженные 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 august as  
WITH RECURSIVE dates AS (  
    SELECT date('2018-08-01') AS level  
    UNION ALL  
    SELECT level + INTERVAL 1 day AS value  
    FROM dates WHERE dates.level < date('2018-08-31')  
)  
SELECT level  
FROM dates;  
  
select  
a.level,  
CASE  
    when p.name is NULL then 0 else 1  
END  
from august a  
left join posts p on p.created_at = a.level;
```

4. (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.

```
SELECT @dell := created_at FROM posts  
order by created_at desc  
limit 5, 1;
```

```
DELETE from posts  
where created_at <= @dell;
```

## 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;
SET GLOBAL log_bin_trust_function_creators = 1;
DELIMITER //

CREATE FUNCTION hello(num INT)
RETURNS VARCHAR(255) DETERMINISTIC
BEGIN
    IF num >= 6 AND num <= 11 THEN RETURN "Доброе утро";
    ELSEIF num >= 12 AND num <= 17 THEN RETURN "Добрый день";
    ELSEIF num >= 18 AND num <= 24 THEN RETURN "Добрый вечер";
    ELSEIF num >= 0 AND num <= 5 THEN RETURN "Доброй ночи";
    ELSE RETURN "Нет такого времени";
END IF;
END//

DELIMITER ;

SELECT hello(2);
```

2. 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.

```
DROP TRIGGER IF EXISTS check_name_descr_update;
DELIMITER //
CREATE trigger check_name_descr_update before UPDATE on products
FOR EACH ROW
BEGIN
    IF
        NEW.name IS NULL AND NEW.description IS NULL THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Empty values are not
allowed';
    END IF;
END//
DELIMITER ;

DROP TRIGGER IF EXISTS check_name_descr_insert;
DELIMITER //
CREATE trigger check_name_descr_insert before INSERT on products
FOR EACH ROW
BEGIN
    IF
        NEW.name IS NULL AND NEW.description IS NULL THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Empty values are not
allowed';
    END IF;
END//
DELIMITER ;
```

3. (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.

0	1	2	3	4	5	6	7	8	9	10
0	1	1	2	3	5	8	13	21	34	55

```
DROP FUNCTION IF EXISTS fibon;
```

```
DELIMITER //
```

```
CREATE FUNCTION fibon(num INT)
```

```
RETURNS INT DETERMINISTIC
```

```
BEGIN
```

```
    RETURN (POW((1 + SQRT(5)) / 2.0, num) - POW((1 - SQRT(5)) / 2.0, num)) / SQRT(5);
```

```
END//
```

```
DELIMITER ;
```

```
SELECT fibon(10);
```