

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 from sample.user where id=1;
insert into sample.user (id, user)
select id, name from shop.users where id=1;
delete from shop.users where id=1;
COMMIT;
```

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

```
CREATE VIEW shop.v_product AS
Select p.id id, p.name product, c.name catalog
from shop.products p left join shop.catalogs c on p.catalog_id=c.id;
```

shop

- Таблицы
- Представления
 - v_product**
 - 123 id (bigint)
 - ABC product (varchar(255))
 - ABC catalog (varchar(255))
- Индексы

```
mysql> Select * from shop.v_product;
```

id	product	catalog
1	Intel Core i3-8100	Процессоры
2	Intel Core i5-7400	Процессоры
3	AMD FX-8320E	Процессоры
4	AMD FX-8320	Процессоры
5	ASUS ROG MAXIMUS X HERO	Материнские платы
6	Gigabyte H310M S2H	Материнские платы
7	MSI B250M GAMING PRO	Материнские платы

7 rows in set (0,00 sec)

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, выставя в соседнем поле значение 1, если дата присутствует в исходном таблице и 0, если она отсутствует.

date	_check
2018-08-01	1
2018-08-02	0
2018-08-03	0
2018-08-04	0
2018-08-05	0
2018-08-06	0
2018-08-07	0
2018-08-08	0
2018-08-09	0
2018-08-10	0
2018-08-11	0
2018-08-12	0
2018-08-13	0
2018-08-14	0
2018-08-15	0
2018-08-16	1
2018-08-17	1

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.

```
delete FROM vk.USERS WHERE id NOT IN (
  SELECT id
  FROM (
    SELECT id
    FROM vk.USERS
    ORDER BY created_at
    LIMIT 5
  ) x
);
```

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.

```
mysql> CREATE USER 'user_select'@'localhost' IDENTIFIED BY '123';
Query OK, 0 rows affected (0,00 sec)

mysql> GRANT SELECT ON shop . * TO 'user_select'@'localhost';
Query OK, 0 rows affected (0,00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,00 sec)

mysql>
mysql> CREATE USER 'user_all_PRIVILEGES'@'localhost' IDENTIFIED BY '123';
Query OK, 0 rows affected (0,00 sec)

mysql> GRANT ALL PRIVILEGES ON shop . * TO 'user_all_PRIVILEGES'@'localhost';
Query OK, 0 rows affected (0,01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,00 sec)
```

2. (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.

```
mysql> CREATE VIEW shop.v_username as SELECT id , name from shop.accounts;
Query OK, 0 rows affected (0,01 sec)

mysql>
mysql> CREATE USER 'user_2_2'@'localhost' IDENTIFIED BY '123';
Query OK, 0 rows affected (0,00 sec)

mysql> GRANT SELECT ON shop.v_username TO 'user_2_2'@'localhost';
Query OK, 0 rows affected (0,01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,00 sec)
```

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;

CREATE FUNCTION hello ()
RETURNS VARCHAR(255) DETERMINISTIC
BEGIN
    RETURN (Select case when DATE_FORMAT(NOW(), '%H') BETWEEN 6 and 11 THEN 'Доброе утро'
        when DATE_FORMAT(NOW(), '%H') BETWEEN 12 and 17 THEN 'Добрый день'
        when DATE_FORMAT(NOW(), '%H') BETWEEN 18 and 23 THEN 'Добрый вечер'
        when DATE_FORMAT(NOW(), '%H') BETWEEN 0 and 5 THEN 'Добрый вечер'
    END);
END;

Select hello();

```

lt

ct hello() | Введите SQL выражение чтобы отфильтровать результаты

ABC hello()

Доброе утро

- 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.

Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed

mysql> INSERT INTO shop.products

-> (name, description, price, catalog_id, created_at, updated_at)

-> VALUES(null, 'test', 0, 0, CURRENT_TIMESTAMP, CURRENT_TIMESTAMP);

Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO shop.products

-> (name, description, price, catalog_id, created_at, updated_at)

-> VALUES(null, null, 0, 0, CURRENT_TIMESTAMP, CURRENT_TIMESTAMP);

ERROR 1643 (02000): Внимание: Не заполнены текстовые поля "name" и "description"

ERROR 1643 (02000): Внимание: Не заполнены текстовые поля "name" и "description"

- (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

123 f
0
1
1
2
3
5
8
13
21
34
55

