

Практическое задание по теме «Транзакции, переменные, представления»

Задание 1. Выполнено. Перенес запись с id = 1 из таблицы shop.users в таблицу sample.users, используя транзакции.

The screenshot shows a SQL script in a window titled '*<localhost> Script' with a tab for 'users'. The script contains the following SQL statements:

```
START TRANSACTION;  
INSERT INTO sample.users SELECT * FROM shop.users WHERE id = 1;  
DELETE FROM shop.users  
WHERE id = 1;  
COMMIT;  
  
SELECT * FROM sample.users;  
SELECT * FROM shop.users;
```

Below the script, two result grids are shown. The first grid, titled 'users 1', shows the result of the first SELECT statement:

	id	name
1	1	Andrey YABLOKOV

The second grid, titled 'users 1 (2)', shows the result of the second SELECT statement:

	id	name
1	2	Sergey IVANOV
2	3	Vladimir FEDOTOV

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

The screenshot shows a SQL script in a window titled '*<localhost> Script' with a tab for 'users'. The script contains the following SQL statements:

```
CREATE VIEW prod_cat AS  
SELECT p.name AS product_name, c.name AS catalog_name  
FROM  
    products AS p  
JOIN  
    catalogs AS c  
ON  
    p.catalog_id = c.id;  
  
SELECT * FROM prod_cat;
```

Below the script, a result grid titled 'prod_cat 1' shows the output of the SELECT statement:

	name_product	name_catalog
1	apple	fruits
2	banana	fruits
3	orange	fruits
4	chocolate	sweets
5	caramel	sweets
6	pork	meet
7	beef	meet

Задание 3. Выполнено. Составил запрос, который выводит полный список дат за август, выставя в соседнем поле значение 1, если дата присутствует в исходной таблице и 0, если она отсутствует.

*<localhost> Script

```
CREATE OR REPLACE VIEW august_day AS SELECT DISTINCT DAY(created_at) FROM messages WHERE MONTH(created_at)=8;
SET @num=0;
SELECT
    @num := @num + 1 AS `number`,
    IF(@num IN (SELECT * FROM august_day),1,0) AS presence
FROM messages
WHERE @num < 31;
```

Results 1 Statistics 1

SELECT @num := @num + 1 AS `number`, IF(Enter a SQL expression to filter results (use Ctrl+Space)

	123 number	123 presence
1	1	0
2	2	0
3	3	1
4	4	1
5	5	1
6	6	0
7	7	0
8	8	1
9	9	0
10	10	0
11	11	1
12	12	1
13	13	1
14	14	0
15	15	0
16	16	0
17	17	0
18	18	0
19	19	0

Задание 4. Выполнено. Составил запрос, который удаляет устаревшие записи из таблицы, оставляя только 5 самых свежих записей.

*<localhost> Script

```
CREATE OR REPLACE VIEW top_new_records AS SELECT id FROM messages ORDER BY created_at DESC LIMIT 5;
DELETE FROM messages
WHERE id NOT IN(SELECT * FROM top_new_records);
SELECT id, created_at FROM messages;
```

messages 1 Statistics 1

SELECT id, created_at FROM messages Enter a SQL expression to filter results (use Ctrl+Space)

	123 id	created_at
1	18	2021-02-13 00:03:48
2	20	2021-03-06 18:12:03
3	33	2020-07-29 19:14:20
4	89	2019-11-07 23:24:44
5	90	2019-12-18 11:55:55

Практическое задание по теме «Администрирование MySQL»

Задание 1. Выполнено. Создал 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 'shop_read'@'localhost' IDENTIFIED BY 'StrongPassword';
Query OK, 0 rows affected (0,01 sec)
```

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

```
mysql> CREATE USER 'shop'@'localhost' IDENTIFIED BY 'SuperPassword';
Query OK, 0 rows affected (0,01 sec)
```

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

Задание 2. Выполнено. Создал представление username таблицы accounts, предоставляющий доступ к столбцам id и name. Создал пользователя user_read, который не имеет доступа к таблице accounts, однако, может извлекать записи из представления username.

```
mysql> SELECT * from accounts;
+----+-----+-----+
| id | name   | password_hash |
+----+-----+-----+
|  1 | Andrey | Password      |
|  2 | Sergey | SuperPassword |
|  3 | Vladimir | StrongPassword |
+----+-----+-----+
3 rows in set (0,00 sec)
```

```
mysql> CREATE VIEW username AS SELECT id, name FROM accounts;
Query OK, 0 rows affected (0,01 sec)
```

```
mysql> SELECT * FROM username;
+----+-----+
| id | name   |
+----+-----+
|  1 | Andrey |
|  2 | Sergey |
|  3 | Vladimir |
+----+-----+
3 rows in set (0,00 sec)
```

```
mysql> CREATE USER 'user_read'@'localhost';
Query OK, 0 rows affected (0,00 sec)
```

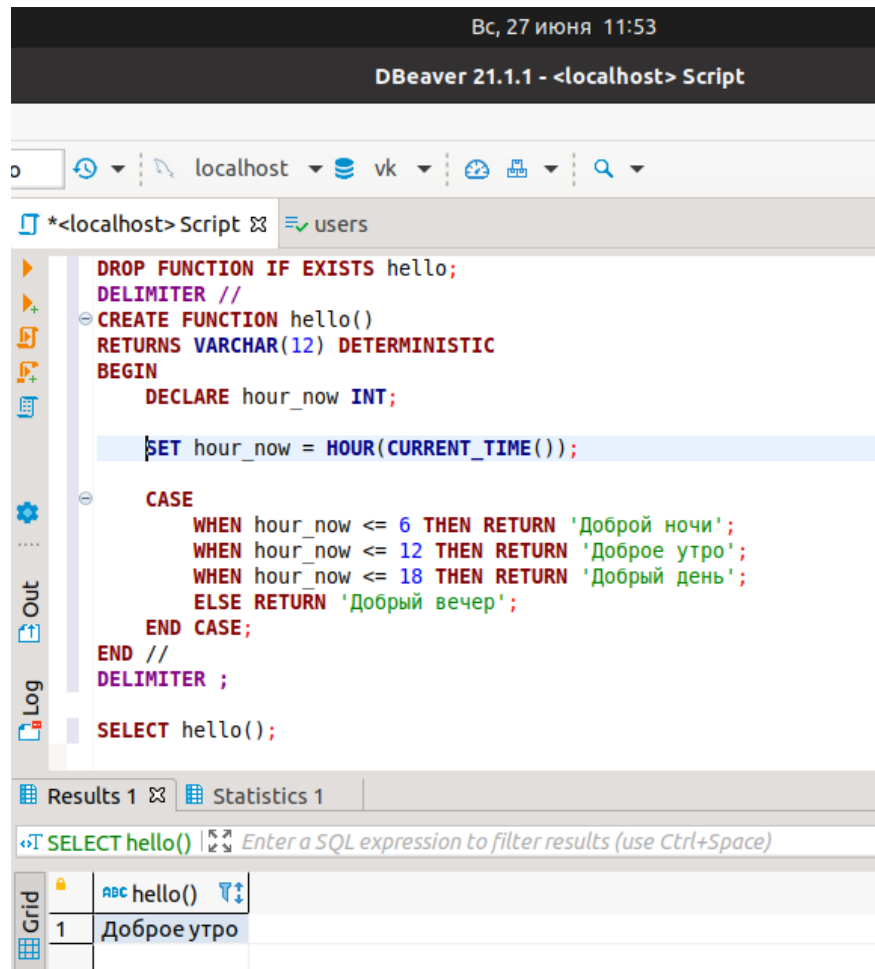
```
mysql> GRANT USAGE, SELECT ON shop.username TO 'user_read'@'localhost';
Query OK, 0 rows affected (0,00 sec)
```

```
mysql> SHOW GRANTS;
+-----+
| Grants for user_read@localhost |
+-----+
| GRANT USAGE ON *.* TO `user_read`@`localhost` |
| GRANT SELECT ON `shop`.`username` TO `user_read`@`localhost` |
+-----+
2 rows in set (0,00 sec)
```

```
mysql> SELECT * FROM shop.accounts;
ERROR 1142 (42000): SELECT command denied to user 'user_read'@'localhost' for table 'accounts'
mysql> SELECT * FROM shop.username;
+----+-----+
| id | name   |
+----+-----+
| 1  | Andrey |
| 2  | Sergey |
| 3  | Vladimir |
+----+-----+
3 rows in set (0,00 sec)
```

Практическое задание по теме «Хранимые процедуры и функции, триггеры»

Задание 1. Выполнено. Создал хранимую функцию `hello()`, которая будет возвращать приветствие, в зависимости от текущего времени суток. С 6:00 до 12:00 функция возвращает фразу "Доброе утро", с 12:00 до 18:00 функция возвращает 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(12) DETERMINISTIC
BEGIN
    DECLARE hour_now INT;

    SET hour_now = HOUR(CURRENT_TIME());

    CASE
        WHEN hour_now <= 6 THEN RETURN 'Доброй ночи';
        WHEN hour_now <= 12 THEN RETURN 'Доброе утро';
        WHEN hour_now <= 18 THEN RETURN 'Добрый день';
        ELSE RETURN 'Добрый вечер';
    END CASE;
END //
DELIMITER ;

SELECT hello();

```

Grid	abc hello()
1	Доброе утро

Задание 2. Выполнено. 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, добился того, чтобы одно из этих полей или оба поля были заполнены. При попытке присвоить обоим полям `NULL`-значение операция отменяется.

```
USE shop;
DELIMITER //
DROP TRIGGER IF EXISTS check_products_insert //
CREATE TRIGGER check_products_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 = 'INSERT canceled';
    END IF;
END//
DROP TRIGGER IF EXISTS check_products_update //
CREATE TRIGGER check_products_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 = 'UPDATE canceled';
    END IF;
END//
DELIMITER ;
```

```
mysql> INSERT INTO products VALUES(8, NULL, NULL, 1);
ERROR 1644 (45000): INSERT canceled
```


```
mysql> UPDATE products
-> SET name = NULL, description = NULL
-> WHERE id=1;
ERROR 1644 (45000): UPDATE canceled
```

```
mysql> INSERT INTO products VALUES(8, 'pear', NULL, 1);
Query OK, 1 row affected (0,00 sec)
```

```
mysql> SELECT * FROM products;
```

```
+-----+-----+-----+-----+
| id | name      | description | catalog_id |
+-----+-----+-----+-----+
| 1 | apple     | NULL       | 1          |
| 2 | banana    | NULL       | 1          |
| 3 | orange    | NULL       | 1          |
| 4 | chocolate | NULL       | 2          |
| 5 | caramel   | NULL       | 2          |
| 6 | pork      | NULL       | 3          |
| 7 | beef      | NULL       | 3          |
| 8 | pear      | NULL       | 1          |
+-----+-----+-----+-----+
8 rows in set (0,00 sec)
```

Задание 3. Выполнено. Написал 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.

*<localhost> Script  users

```

DROP FUNCTION IF EXISTS FIBONACCI;
DELIMITER //
CREATE FUNCTION FIBONACCI(num_count INT)
RETURNS BIGINT DETERMINISTIC
BEGIN
    DECLARE num1, num2, res BIGINT;
    DECLARE idx INT;

    SET num1 = 0;
    SET num2 = 1;
    SET idx = 2;

    IF num_count = 1 THEN RETURN num1;
    ELSEIF num_count = 2 THEN RETURN num2;
    ELSE
        WHILE idx <= num_count DO
            SET res = num1 + num2;
            SET num1 = num2;
            SET num2 = res;
            SET idx = idx + 1;
        END WHILE;
        RETURN res;
    END IF;
END //
DELIMITER ;

SELECT FIBONACCI(10);

```

Results 1 Statistics 1

 SELECT FIBONACCI(10) *Enter a SQL expression to filter results (use Ctrl+Space)*

Grid		123 FIBONACCI(10)
	1	55