

Урок 6. SQL – Транзакции. Временные таблицы, управляющие конструкции, циклы

Для решения задач используйте базу данных lesson_6

```
source C:\Users\User\Downloads\lesson_6.sql;
```

```
mysql> show tables;
```

```
+-----+
| Tables_in_lesson_4 |
+-----+
| communities        |
| friend_requests    |
| likes               |
| media               |
| media_types         |
| messages            |
| profiles            |
| users               |
| users_communities  |
+-----+
```

1. Создайте таблицу users_old, аналогичную таблице users.

Создайте процедуру, с помощью которой можно переместить любого (одного) пользователя из таблицы users в таблицу users_old. (использование транзакции с выбором commit или rollback – обязательно).

```
DROP TABLE IF EXISTS users_old;
```

```
CREATE TABLE users_old (
    id SERIAL PRIMARY KEY,
    firstname VARCHAR(50),
    lastname VARCHAR(50) COMMENT 'Фамилия',
    email VARCHAR(120) UNIQUE
);
```

```
USE lesson_4;
```

```
START TRANSACTION;
```

```
    INSERT INTO users_old SELECT * FROM users WHERE id = 2;
```

```
    DELETE FROM users WHERE id = 2
```

```
    LIMIT 1;
```

```
COMMIT;
```

2. 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 PROCEDURE IF EXISTS hello;
```

```
delimiter //
```

```
CREATE PROCEDURE hello()
```

```
BEGIN
```

```
    SET @time = HOUR(NOW());
```

```
    CASE
```

```

        WHEN (@time > 0 AND @time < 6) THEN
            SELECT 'Доброй ночи';
        WHEN (@time > 6 AND @time < 12) THEN
            SELECT 'Доброе утро!';
        WHEN (@time > 12 AND @time < 18) THEN
            SELECT 'Добрый день!';
        WHEN (@time > 18 AND @time < 24) THEN
            SELECT 'Добрый вечер';
    END CASE;
END//
CALL hello()
delimiter ;

CALL hello();

```

3. (по желанию)* Создайте таблицу logs типа Archive. Пусть при каждом создании записи в таблицах users, communities и messages в таблицу logs помещается время и дата создания записи, название таблицы, идентификатор первичного ключа.

```

-- заполнить текущей датой и временем
UPDATE logs SET created_at = NOW(), updated_at = NOW();

DROP TABLE IF EXISTS logs;
CREATE TABLE logs (
    table_name VARCHAR(20) NOT NULL,
    prkey_id INT UNSIGNED NOT NULL,
    time_rec /*created_at*/ DATETIME DEFAULT NOW()
) ENGINE = ARCHIVE;

DROP TRIGGER IF EXISTS users_log;
CREATE TRIGGER users_log AFTER INSERT ON users FOR EACH ROW
    INSERT INTO logs
        SET
            table_name = 'users',
            prkey_id = NEW.id;

DROP TRIGGER IF EXISTS communities_log;
CREATE TRIGGER communities_log AFTER INSERT ON communities FOR EACH ROW
    INSERT INTO logs
        SET
            table_name = 'communities',
            pk_id = NEW.id;

DROP TRIGGER IF EXISTS messages_log;
CREATE TRIGGER messages_log AFTER INSERT ON messages FOR EACH ROW
    INSERT INTO logs
        SET
            table_name = 'messages',
            pk_id = NEW.id;

```