



# Базы данных и SQL

Семинар 6.



Вопросы?

Вопросы?



Вопросы?



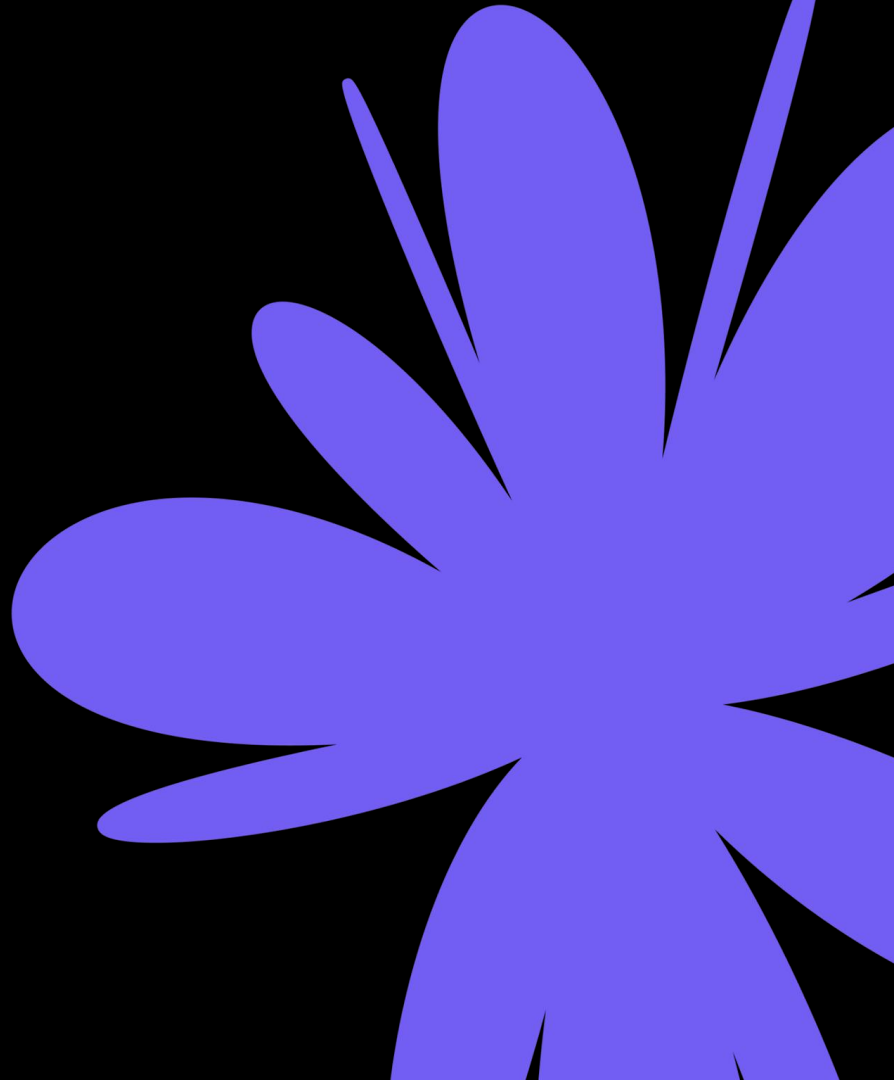
# План на сегодня:

- Викторина
- Транзакции
- Хранимые процедуры
- Перерыв
- Функции
- Циклы
- Домашнее задание





# Викторина



## Выберите верные утверждения:

1. функции вызываются с помощью SELECT, а процедуры CALL
2. процедура используется исключительно для выполнения вычисления и возвращения результата
3. в процедурах можно указать тип переменной IN, OUT
4. в функциях можно использовать переменные типа OUT
5. процедуру можно указать в качестве параметра другой процедуры



## Выберите верные утверждения:

1. функции вызываются с помощью SELECT, а процедуры CALL
2. процедура используется исключительно для выполнения вычисления и возвращения результата
3. в процедурах можно указать тип переменной IN, OUT
4. в функциях можно использовать переменные типа OUT
5. процедуру можно указать в качестве параметра другой процедуры



## **Команда ... создает хранимую процедуру;**

1. ALTER PROCEDURE
2. CREATE PROCEDURE
3. ADD PROCEDURE |
4. DROP PROCEDURE



## Команда ... создает хранимую процедуру;

1. ALTER PROCEDURE
2. CREATE PROCEDURE
3. ADD PROCEDURE
4. DROP PROCEDURE





## Выберите ложное утверждение:

1. В MYSQL реализовать автоматическое выполнение COMMIT или ROLLBACK транзакции можно в рамках хранимой процедуры.
2. Для начала транзакции нужно написать START TRANSACTION.
3. При завершении транзакции изменения, сделанные от начала транзакции и ранее невидимые для других транзакций действия, всегда фиксируются в базе данных.



## Выберите ложное утверждение:

1. В MYSQL реализовать автоматическое выполнение COMMIT или ROLLBACK транзакции можно в рамках хранимой процедуры.
2. Для начала транзакции нужно написать START TRANSACTION.
3. При завершении транзакции изменения, сделанные от начала транзакции и ранее невидимые для других транзакций действия, всегда фиксируются в базе данных.



# Циклы в MySQL:

1. Циклы реализуются в процедурах и функциях.
2. Данный функционал не реализован.
3. Для реализации цикла используются операторы WHILE, FOR, LOOP



# Циклы в MySQL:

1. Циклы реализуются в процедурах и функциях.
2. Данный функционал не реализован.
3. Для реализации цикла используются операторы WHILE, FOR, LOOP



# Транзакции

```
START TRANSACTION;
```

```
INSERT INTO users (firstname, lastname, email)  
VALUES ('Дмитрий', 'Дмитриев', 'dima@mail.ru');
```

```
SET @last_user_id = last_insert_id();
```

```
INSERT INTO profiles (user_id, hometown, birthday, photo_id)  
VALUES (@last_user_id, 'Moscow', '1999-10-10', NULL);
```

```
COMMIT; -- применить изменения
```

```
-- ROLLBACK; -- отменить изменения
```



# Создание и вызов процедуры в MySQL

-- Создание процедуры:

```
DROP PROCEDURE IF EXISTS procedure_name;  
DELIMITER //  
CREATE PROCEDURE procedure_name(parameter_list)  
BEGIN  
    statements;  
    statements;  
    statements;  
END //  
DELIMITER ;
```

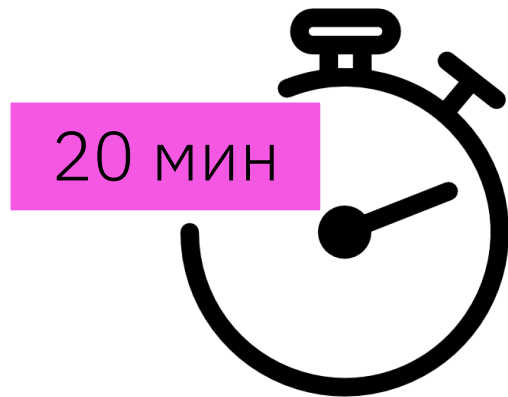
-- Вызов процедуры:

```
CALL procedure_name(argument_list);
```



**Задача 1. Создайте процедуру, которая выберет для одного пользователя 5 пользователей в случайной комбинации, которые удовлетворяют хотя бы одному критерию:**

- 1) из одного города**
- 2) состоят в одной группе**
- 3) друзья друзей**



**Решение 1.** Создайте процедуру, которая выберет для одного пользователя 5 пользователей в случайной комбинации, которые удовлетворяют хотя бы одному критерию: 1) из одного города; 2) состоят в одной группе; 3) друзья друзей.

```
DROP PROCEDURE IF EXISTS sp_friendship_offers;  
DELIMITER //  
CREATE PROCEDURE sp_friendship_offers(for_user_id BIGINT)  
BEGIN  
    -- друзья  
    WITH friends AS (  
        SELECT initiator_user_id AS id  
            FROM friend_requests  
            WHERE status = 'approved' AND target_user_id = for_user_id  
        UNION  
        SELECT target_user_id  
            FROM friend_requests  
            WHERE status = 'approved' AND initiator_user_id = for_user_id  
    )  
    ...
```



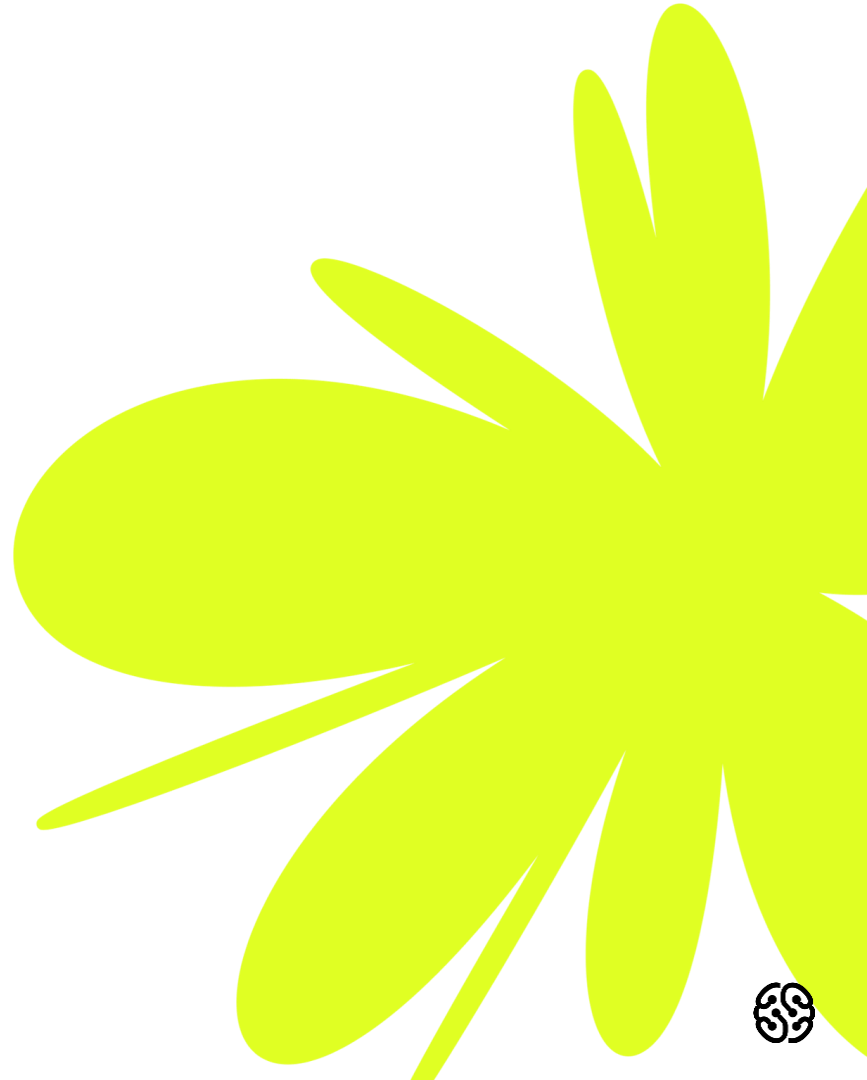


```
...
-- общий город
SELECT p2.user_id FROM profiles p1
JOIN profiles p2 ON p1.hometown = p2.hometown
WHERE p1.user_id = for_user_id AND p2.user_id <> for_user_id
-- состоят в одной группе
UNION SELECT uc2.user_id FROM users_communities uc1
JOIN users_communities uc2 ON uc1.community_id = uc2.community_id
WHERE uc1.user_id = for_user_id AND uc2.user_id <> for_user_id
-- друзья друзей
UNION SELECT fr.initiator_user_id FROM friends f
JOIN friend_requests fr ON fr.target_user_id = f.id
WHERE fr.initiator_user_id != for_user_id AND fr.status = 'approved'
UNION SELECT fr.target_user_id FROM friends f
JOIN friend_requests fr ON fr.initiator_user_id = f.id
WHERE fr.target_user_id != for_user_id AND status = 'approved'
ORDER BY rand()
LIMIT 5;
END//
DELIMITER ;
```



Ваши вопросы?

Перерыв



# Создание и вызов функции в MySQL

-- Создание функции:

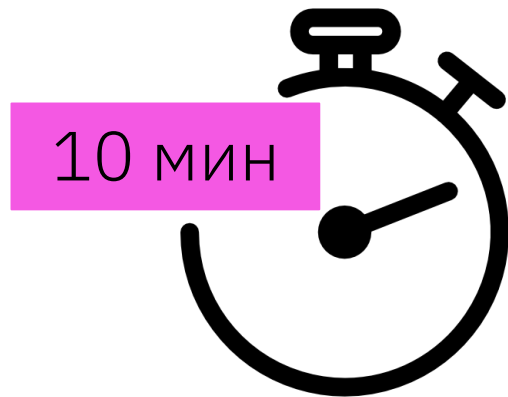
```
DROP FUNCTION IF EXISTS function_name;  
DELIMITER //  
CREATE FUNCTION function_name(parameter_list)  
RETURNS type [characteristic]  
BEGIN  
    statements;  
    statements;  
    statements;  
  
RETURN ... ;  
END//  
DELIMITER ;
```

-- Вызов функции:

```
SELECT function_name(argument_list);
```



**Задача 2. Создание функции, вычисляющей  
коэффициент популярности пользователя**  
(по заявкам на дружбу – таблица *friend\_requests*)



## Решение 2. Создание функции, вычисляющей коэффициент популярности пользователя *(по заявкам на дружбу – таблица friend\_requests)*

```
DROP FUNCTION IF EXISTS friendship_direction;
DELIMITER //
CREATE FUNCTION friendship_direction(check_user_id BIGINT)
RETURNS FLOAT READS SQL DATA
BEGIN
    DECLARE requests_to_user INT; -- заявок к пользователю
    DECLARE requests_from_user INT; -- заявок от пользователя

    SET requests_to_user = (SELECT count(*) FROM friend_requests
    WHERE target_user_id = check_user_id);

    SELECT count(*) INTO requests_from_user
    FROM friend_requests WHERE initiator_user_id = check_user_id;

    RETURN requests_to_user / requests_from_user;
END//
DELIMITER ;
```



# Циклы

```
DECLARE v1 INT DEFAULT 5;  
WHILE v1 > 0 DO  
  BEGIN  
    ...  
    SET v1 = v1 - 1;  
  END;  
END WHILE;
```



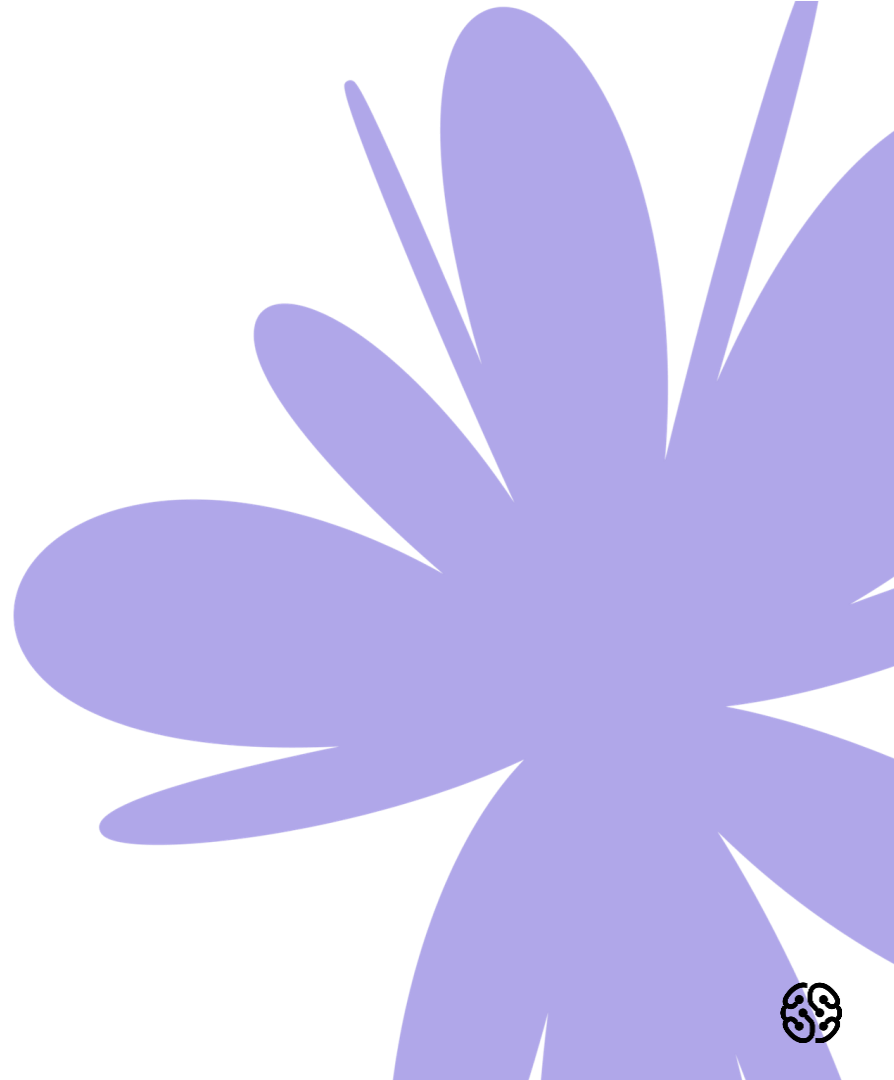
**Задача 3. Необходимо перебрать всех пользователей и тем пользователям, у которых дата рождения меньше определенной даты обновить дату рождения на сегодняшнюю дату. (реализация с помощью цикла)**

Тело процедуры:

```
DECLARE id_max_users INT;  
DECLARE count_find INT;  
SET id_max_users = (SELECT MAX(user_id) FROM profiles);  
WHILE (id_max_users > 0) DO  
    BEGIN  
        SET count_find = (SELECT COUNT(*) FROM profiles WHERE user_id = id_max_users AND  
birthday > start_date);  
        IF (count_find>0 )THEN UPDATE profiles SET birthday=NOW() WHERE user_id=id_max_users;  
        END IF;  
        SET id_max_users = id_max_users - 1;  
    END;  
END WHILE;
```



Ваши вопросы?





**Для решения задач используйте базу данных lesson\_4 (скрипт создания, прикреплен к 4 семинару).**

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).*
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".
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.



# Рефлексия



**Был урок полезен вам?**



**Узнали вы что-то новое?**




**Что было сложно?**





Спасибо  
за внимание

A yellow smiley face is drawn over the text. It has two vertical lines for eyes and a curved line for a mouth, positioned to the right of the word 'Спасибо' and below the word 'за'.