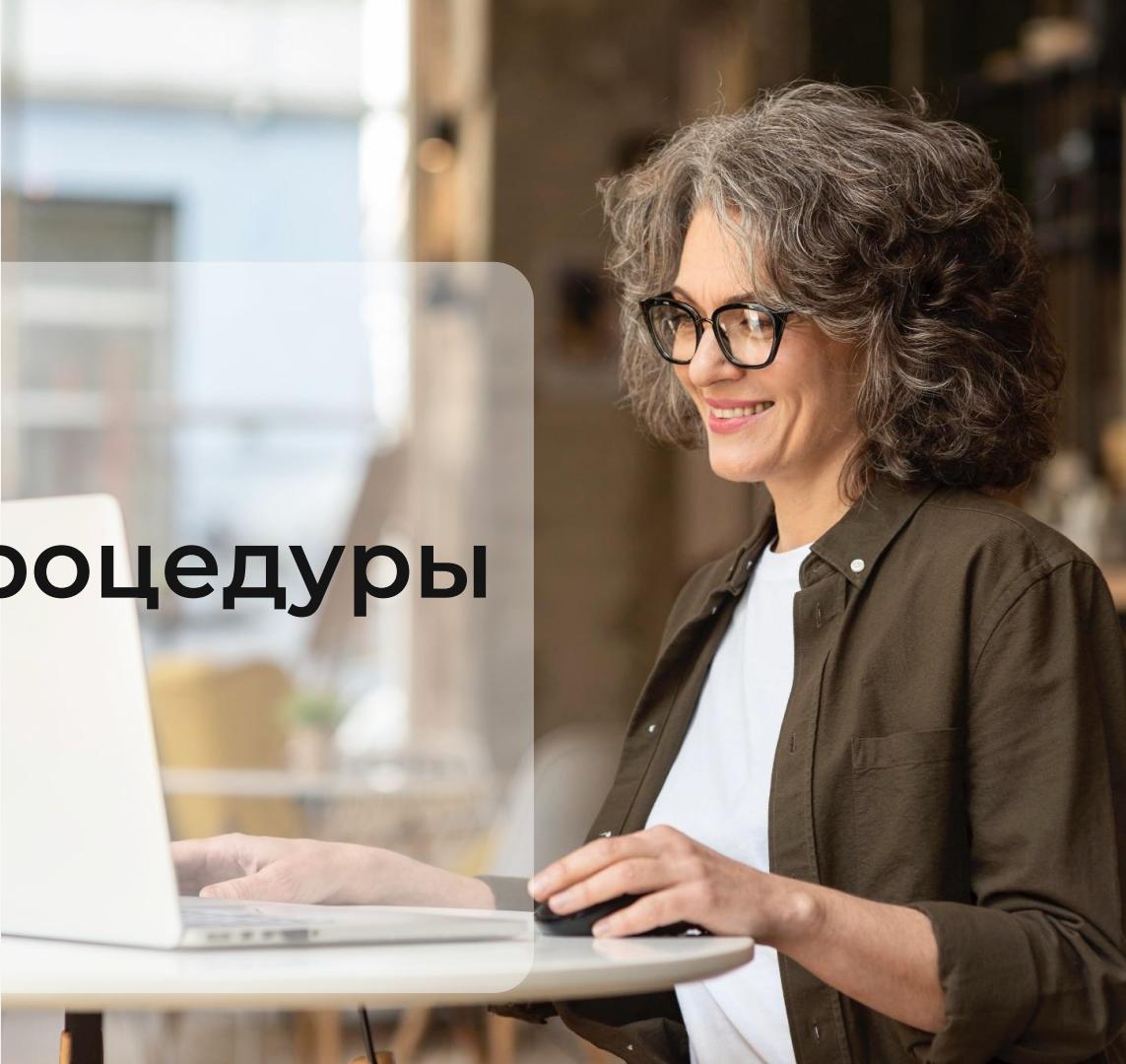


Node.JS

Хранимые процедуры



Преподаватель

Портрет

Имя Фамилия

Текущая должность

Количество лет опыта

Какой у Вас опыт - ключевые кейсы

Самые яркие проекты

Дополнительная информация по вашему усмотрению

Корпоративный e-mail

Социальные сети (по желанию)

Важно



Камера должна быть включена на протяжении всего занятия



В течение занятия вопросы задавать в чате или когда преподаватель спрашивает, есть ли у Вас вопросы



Вести себя уважительно и этично по отношению к остальным участникам занятия



Организационные вопросы по обучению решаются с кураторами, а не на тематических занятиях



Во время занятия будут интерактивные задания, будьте готовы включить камеру или демонстрацию экрана по просьбе преподавателя

Повторение



Определение пользовательской функции



Примеры пользовательских функций



Использование пользовательских функций

План занятия

- Хранимые процедуры
- Создание хранимой процедуры
- Пример создания простой хранимой процедуры
- Вызов хранимой процедуры

База данных для работы на занятии

База данных с доступом на запись

hostname: ich-edit.edu.itcareerhub.de

MYSQL_USER: ich1

MYSQL_PASSWORD: ich1_password_ilovedbs

ОСНОВНОЙ БЛОК



Хранимые процедуры



Хранимые процедуры (stored procedures)

Это блоки SQL-кода, которые могут быть сохранены в базе данных для многократного использования.

Хранимые процедуры





Компилирование SQL-запросов

Это процесс, при котором сервер базы данных обрабатывает запрос перед его выполнением, разбивая его на этапы для более эффективного выполнения.

Основные особенности хранимых процедур

Улучшение производительности

Подходят для сложной бизнес-логики и выполняемых на сервере процессов

Безопасность

Возможность повторного использования

ВОПРОСЫ



**Создание
процедуры**

хранимой

Хранимая процедура в MySQL



Основной синтаксис

```
DELIMITER $$  
CREATE      PROCEDURE      имя_процедуры  
([параметры])  
BEGIN  
    -- тело процедуры  
END $$  
DELIMITER ;
```

Пояснения

- DELIMITER \$\$ — используется для изменения разделителя команд, так как внутри процедуры могут быть точки с запятой.
- CREATE PROCEDURE — команда для создания процедуры.
- [параметры] — могут быть входными (IN), выходными (OUT) и двунаправленными (INOUT) параметрами.
- BEGIN ... END — блок кода, который выполняет логику процедуры.

ВОПРОСЫ



Пример
простой
процедуры

создания
хранимой

Хранимая процедура для добавления новой записи в таблицу



Код

```
DELIMITER $$  
CREATE PROCEDURE add_employee(IN  
emp_name VARCHAR(100), IN emp_age INT)  
BEGIN  
    INSERT INTO employees (name, age)  
VALUES (emp_name, emp_age);  
END $$  
DELIMITER ;
```

Пояснения

- Параметры `emp_name` и `emp_age` передаются как входные параметры (`IN`).
- В теле процедуры происходит добавление новой записи в таблицу `employees`.

Подготовка к работе с хранимой процедурой



1

- Перед вызовом хранимой процедуры необходимо создать пустую таблицу в вашей персональной базе данных:

```
CREATE TABLE employees ( id INT PRIMARY KEY AUTO_INCREMENT, name VARCHAR(100), age INT, salary INT, department_id INT );
```

2

- Заполнить таблицу 5 любыми записями



**Вызов
процедуры**

хранимой

Вызов хранимой процедуры



Синтаксис

```
CALL имя_процедуры(аргументы);
```

Пример

```
CALL add_employee('John Doe', 30);
```

ВОПРОСЫ

ПРАКТИЧЕСКАЯ РАБОТА

Оценка зарплаты

Создайте хранимую процедуру, которая принимает в качестве входного параметра IN employee_id и возвращает в качестве выходного параметра 1 или 0. Если зарплата сотрудника выше средней зарплаты по всем департаментам – 1, в противном случае – 0.

ВОПРОСЫ

Заключение

