

ГУАП

КАФЕДРА № 41

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

Старший преподаватель

должность, уч. степень, звание

подпись, дата

Б.К. Акопян

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5

ХРАНИМЫЕ ПРОЦЕДУРЫ И ПОЛЬЗОВАТЕЛЬСКИЕ ФУНКЦИИ

по курсу: БАЗЫ ДАННЫХ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4217

подпись, дата

Д.М. Никитин

инициалы, фамилия

Санкт-Петербург 2024

1. **Цель работы:** программная реализация хранимых процедур на сервере.
2. **Вариант:** 24 – база данных железнодорожной станции («Новосибирск-главный»).
3. **Схема таблиц:**

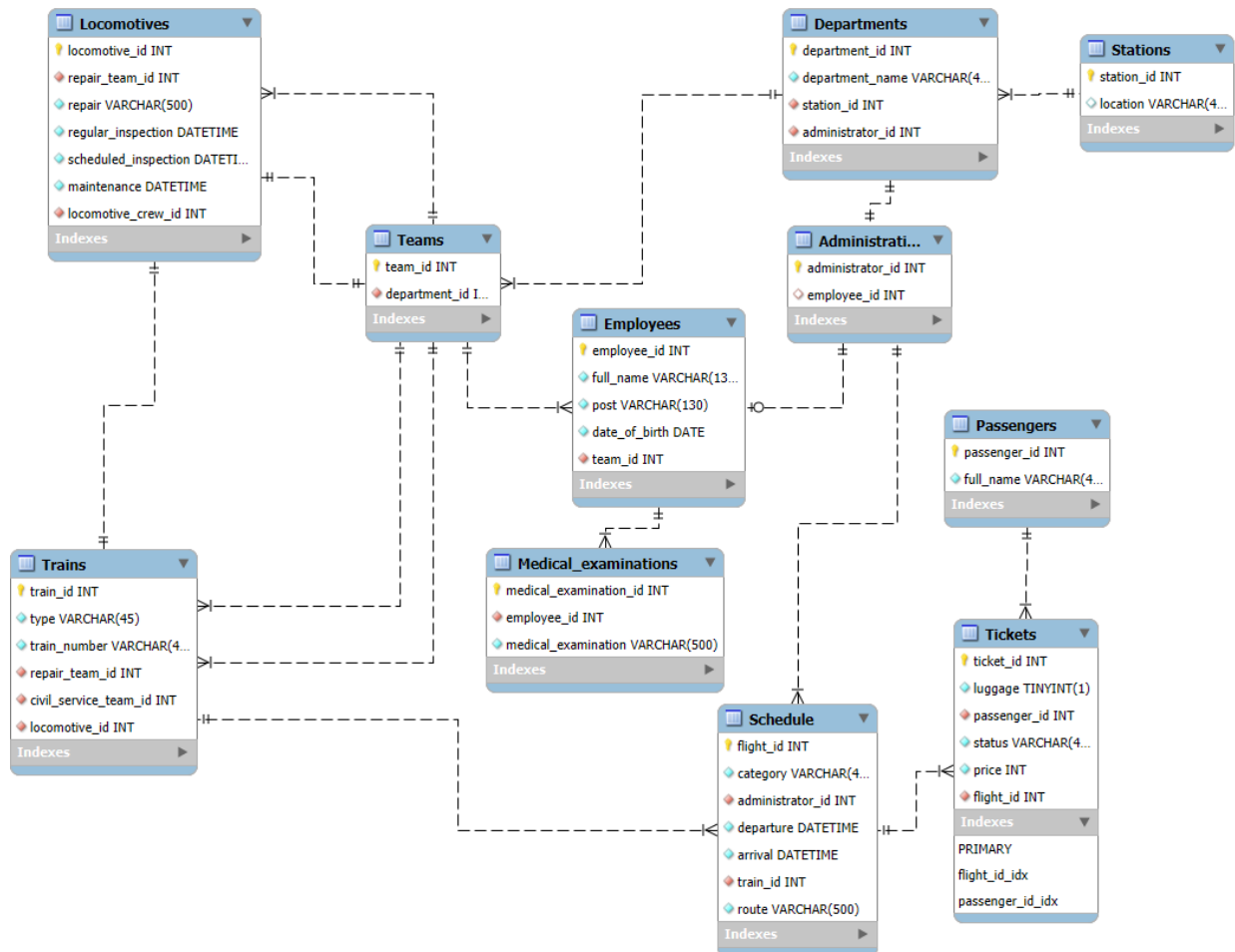


Рисунок 1 – Схема таблиц

4. **Скриншот списка хранимых процедур:**

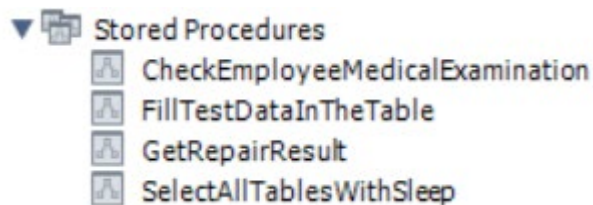


Рисунок 2 – Список хранимых процедур

5. **Текстовое и алгоритмическое описание процедур, входных параметров, листинг созданных процедур с комментариями:**

Процедура CheckEmployeeMedicalExamination:

DELIMITER //

```

CREATE PROCEDURE CheckEmployeeMedicalExamination()
BEGIN
    -- Удаляем временную таблицу после использования
    DROP          TEMPORARY          TABLE          IF          EXISTS
`Train_Station`.`medical_examination_result`;

    -- Создаем временную таблицу medical_examination_result
    CREATE          TEMPORARY          TABLE          IF          NOT          EXISTS
`Train_Station`.`medical_examination_result` (
        recommendation_id INT AUTO_INCREMENT PRIMARY KEY,
        employee_id INT
    );

    -- Вставляем данные в временную таблицу
    medical_examination_result
    INSERT INTO `Train_Station`.`medical_examination_result`
(employee_id)
    SELECT employee_id
    FROM `Train_Station`.`Medical_examinations`
    WHERE medical_examination LIKE '%Предстоит пройти%';

    -- Проверяем, есть ли сотрудники, требующие медицинского
    осмотра
    IF (SELECT COUNT(*) FROM
`Train_Station`.`medical_examination_result`) > 0 THEN
        -- Выводим содержимое временной таблицы
        medical_examination_result
        SELECT * FROM
`Train_Station`.`medical_examination_result`;
    ELSE
        -- Выводим сообщение, что все сотрудники в порядке
        SELECT 'Все сотрудники в порядке' AS message;
    END IF;

    -- Удаляем временную таблицу после использования
    DROP          TEMPORARY          TABLE          IF          EXISTS
`Train_Station`.`medical_examination_result`;
END //

```

DELIMITER ;

Процедура FillTestDataInTheTable:

```

DELIMITER //
CREATE PROCEDURE FillTestDataInTheTable()
BEGIN
    DECLARE i INT DEFAULT 1; -- Для обновления в конце
    -- Вставка данных в таблицу Stations
    INSERT INTO `Train_Station`.`Stations` (`location`)
VALUES
    ('Новосибирск-Главный'),
    ('Вердск'),
    ('Обь');

```

```

INSERT INTO `Train_Station`.`Passengers` (`full_name`)
VALUES
    ('Пассажир А'),
    ('Пассажир Б'),
    ('Пассажир В');

INSERT INTO `Train_Station`.`Administration`
(`employee_id`) VALUES
    (NULL), (NULL), (NULL), (NULL), (NULL), (NULL), (NULL);

INSERT INTO `Train_Station`.`Departments`
(`department_name`, `station_id`, `administrator_id`) VALUES
    ('Водители', 1, 1),
    ('Диспетчеры', 1, 2),
    ('Ремонтники подвижного состава', 1, 3),
    ('Служба ремонта путей', 1, 4),
    ('Кассиры', 1, 5),
    ('Служба подготовки составов', 1, 6),
    ('Справочная служба', 1, 7);

INSERT INTO `Train_Station`.`Teams` (`department_id`)
VALUES
    (1), (2), (3), (4), (5), (6), (7);

INSERT INTO `Train_Station`.`Locomotives`
(`repair_team_id`, `repair`, `regular_inspection`,
`scheduled_inspection`, `maintenance`, `locomotive_crew_id`)
VALUES
    (3, 'Мелкий ремонт', '2024-10-01 08:00:00', '2024-10-15
08:00:00', '2024-10-30 08:00:00', 1),
    (3, 'Крупный ремонт', '2024-10-02 08:00:00', '2024-10-
16 08:00:00', '2024-10-31 08:00:00', 2);

INSERT INTO `Train_Station`.`Employees` (`full_name`,
`post`, `date_of_birth`, `team_id`) VALUES
    ('Иван Иванов', 'Водитель', '1980-01-01', 1),
    ('Анна Петрова', 'Диспетчер', '1985-02-15', 2),
    ('Алексей Смирнов', 'Ремонтник', '1990-03-20', 3),
    ('Дмитрий Кузнецов', 'Путеец', '1995-04-25', 4),
    ('Екатерина Соколова', 'Кассир', '2000-05-30', 5),
    ('Мария Федорова', 'Служба подготовки составов', '2001-
06-05', 6),
    ('Ольга Морозова', 'Справочная служба', '2000-07-10',
7);

INSERT INTO `Train_Station`.`Medical_examinations`
(`employee_id`, `medical_examination`) VALUES
    (1, 'Пройден'),
    (2, 'Пройден'),
    (3, 'Пройден'),
    (4, 'Пройден'),
    (5, 'Пройден'),

```

```

        (6, 'Пройден'),
        (7, 'Предстоит пройти');

        INSERT INTO `Train_Station`.`Trains` (`type`,
`train_number`, `repair_team_id`, `civil_service_team_id`,
`locomotive_id`) VALUES
        ('Скорый', '101', 3, 6, 1),
        ('Пассажирский', '102', 3, 6, 2);

        INSERT INTO `Train_Station`.`Schedule` (`category`,
`administrator_id`, `departure`, `arrival`, `train_id`, `route`)
VALUES
        ('Внутренний', 1, '2024-10-10 08:00:00', '2024-10-10
12:00:00', 1, 'Новосибирск-Главный - Бердск'),
        ('Международный', 2, '2024-10-11 09:00:00', '2024-10-11
13:00:00', 2, 'Новосибирск-Главный - Обь');

        INSERT INTO `Train_Station`.`Tickets` (`luggage`,
`passenger_id`, `status`, `price`, `flight_id`) VALUES
        (1, 1, 'Забронирован', 50, 1),
        (0, 2, 'Забронирован', 30, 2),
        (1, 3, 'Отменен', 40, 1);
        -- цикл для обновления данных

        WHILE i <= 7 DO
        UPDATE `Train_Station`.`Administration`
        SET `employee_id` = i
        WHERE `administrator_id` = i;

        SET i = i + 1;
    END WHILE;
    END //
DELIMITER ;

```

Процедура GetRepairResult:

```

DELIMITER //

CREATE PROCEDURE GetRepairResult()
BEGIN
    DROP TEMPORARY TABLE IF EXISTS
`Train_Station`.`repair_result`;

    CREATE TEMPORARY TABLE IF NOT EXISTS
`Train_Station`.`repair_result` (
        recommendation_id INT AUTO_INCREMENT PRIMARY KEY,
        locomotive_id INT
    );

    -- Вставляем данные в временную таблицу repair_result
    INSERT INTO `Train_Station`.`repair_result` (locomotive_id)
    SELECT locomotive_id
    FROM `Train_Station`.`Locomotives`
    WHERE regular_inspection < NOW() - INTERVAL 24 HOUR;

```

```
SELECT * FROM `Train_Station`.`repair_result`;  
  
DROP          TEMPORARY          TABLE          IF          EXISTS  
`Train_Station`.`repair_result`;  
END //
```

```
DELIMITER ;
```

Процедура SelectAllTablesWithSleep:

```
DELIMITER //
```

```
CREATE PROCEDURE SelectAllTablesWithSleep(sleep_time INT)  
BEGIN
```

```
SELECT * FROM `Train_Station`.`Stations`;  
SELECT SLEEP(sleep_time);
```

```
SELECT * FROM `Train_Station`.`Passengers`;  
SELECT SLEEP(sleep_time);
```

```
SELECT * FROM `Train_Station`.`Administration`;  
SELECT SLEEP(sleep_time);
```

```
SELECT * FROM `Train_Station`.`Departments`;  
SELECT SLEEP(sleep_time);
```

```
SELECT * FROM `Train_Station`.`Teams`;  
SELECT SLEEP(sleep_time);
```

```
SELECT * FROM `Train_Station`.`Locomotives`;  
SELECT SLEEP(sleep_time);
```

```
SELECT * FROM `Train_Station`.`Employees`;  
SELECT SLEEP(sleep_time);
```

```
SELECT * FROM `Train_Station`.`Medical_examinations`;  
SELECT SLEEP(sleep_time);
```

```

SELECT * FROM `Train_Station`.`Trains`;
SELECT SLEEP(sleep_time);

SELECT * FROM `Train_Station`.`Tickets`;
SELECT SLEEP(sleep_time);

SELECT * FROM `Train_Station`.`Schedule`;
SELECT SLEEP(sleep_time);

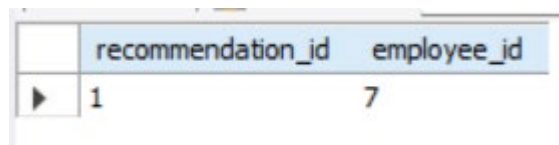
END //

DELIMITER ;

```

6. Скриншоты выполнения хранимых процедур:

Процедура CheckEmployeeMedicalExamination:



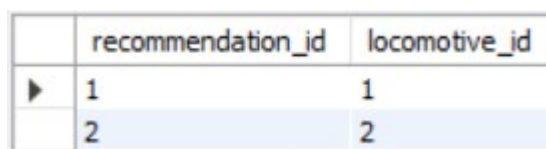
| | recommendation_id | employee_id |
|---|-------------------|-------------|
| ▶ | 1 | 7 |

Рисунок 3 – Работа процедуры CheckEmployeeMedicalExamination

Процедура FillTestDataInTheTable:

Результатом работы процедуры FillTestDataInTheTable является заполнение таблицы тестовыми данными.

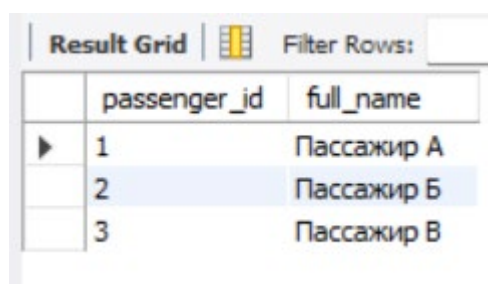
Процедура GetRepairResult:



| | recommendation_id | locomotive_id |
|---|-------------------|---------------|
| ▶ | 1 | 1 |
| | 2 | 2 |

Рисунок 4 – Работа процедуры GetRepairResult

Процедура SelectAllTablesWithSleep:



| | passenger_id | full_name |
|---|--------------|------------|
| ▶ | 1 | Пассажир А |
| | 2 | Пассажир Б |
| | 3 | Пассажир В |

Рисунок 5 – Работа процедуры SelectAllTablesWithSleep

7. Прикладное назначение процедур:

Процедура CheckEmployeeMedicalExamination:

Данная процедура выводит временную таблицу с айди работников, которым требуется пройти медосмотр. Может подойти для работников медблока.

Процедура FillTestDataInTheTable:

Процедура заполняет таблицу тестовыми данными.

Процедура GetRepairResult:

Процедура выводит на экран айди локомотивов, которым требуется регулярный осмотр. Может подойти для работников команды ремонта.

Процедура SelectAllTablesWithSleep:

Выводит все таблицы по очереди с задержкой.

8. Вывод:

В ходе выполнения лабораторной работы №5 по курсу "Базы данных" была проведена программная реализация хранимых процедур на сервере. Работа включала создание и тестирование нескольких хранимых процедур, которые выполняют различные задачи, связанные с управлением данными в базе данных железнодорожной станции "Новосибирск-главный".

Реализованные процедуры

CheckEmployeeMedicalExamination:

Процедура проверяет, есть ли сотрудники, которым требуется медицинский осмотр, на основе ключевого слова в столбце medical_examination.

Если такие сотрудники найдены, их идентификаторы выводятся в временную таблицу medical_examination_result.

Если нет, выводится сообщение о том, что все сотрудники в порядке.

FillTestDataInTheTable:

Процедура заполняет таблицы базы данных тестовыми данными.

Включает вставку данных в таблицы Stations, Passengers, Administration, Departments, Teams, Locomotives, Employees, Medical_examinations, Trains, Schedule, и Tickets.

GetRepairResult:

Процедура выводит информацию о локомотивах, которым требуется регулярный осмотр, если последний осмотр был более 24 часов назад.

Идентификаторы локомотивов выводятся в временную таблицу repair_result.

SelectAllTablesWithSleep:

Процедура последовательно выводит данные из всех таблиц базы данных с задержкой, заданной параметром sleep_time.