

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 №6

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПОЛЬЗОВАТЕЛЬСКИХ ФУНКЦИЙ

по курсу: БАЗЫ ДАННЫХ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4217

подпись, дата

Д.М. Никитин

инициалы, фамилия

Санкт-Петербург 2024

1. **Цель работы:** программная реализация пользовательских функций на сервере.

2. **Вариант и задание:** 24 – база данных железнодорожной станции («Новосибирск-главный»).

1. Создайте хранимую функцию, используя операторы ветвления с параметром (или параметрами).

2. Реализуйте хранимую функцию, используя циклы (для вашей предметной области).

3. Протестируйте каждую функцию на методами позитивного и негативного тестирования: при вводе значений из допустимого диапазона значение и при использовании значений вне этого диапазона. Сделайте скриншоты.

4. Покажите список хранимых функций.

5. Выполните отчет в соответствии с требованиями ГОСТ 7.32-2017 и ГОСТ 2.105-2019: по оформлению отчетов (<https://guap.ru/standart/doc>).

3. **Схема данных из MySQL Workbench:**

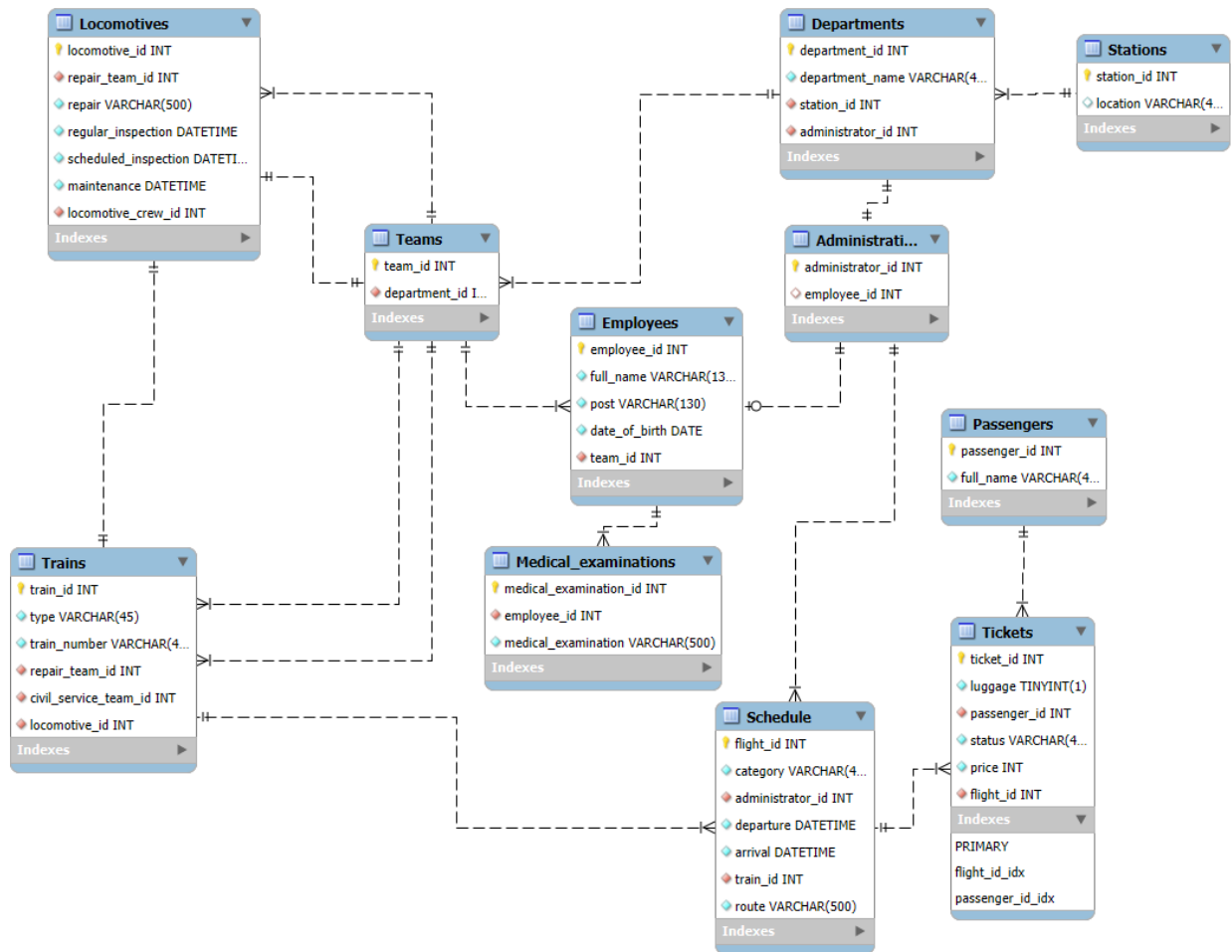


Рисунок 1 – Схема данных, на которой выполняется вариант

4. Скриншот списка пользовательских функций:

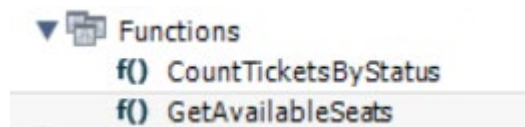


Рисунок 2 – Скриншот из списка пользовательских функций

5. Текстовое описание назначения функций, входных параметров, листинг созданных функций с комментариями. Скриншоты выполнения функций:

Функция GetAvailableSeats:

DELIMITER //

```

CREATE FUNCTION GetAvailableSeats(p_route VARCHAR(500),
p_departure DATETIME) RETURNS INT
READS SQL DATA
BEGIN
    DECLARE total_seats INT;
    DECLARE blocked_seats INT;
    DECLARE flight_id_funk INT;
  
```

```

DECLARE train_id_funk INT;
DECLARE type_funk VARCHAR(45);

-- Получение train_id по маршруту и дате отправления
SELECT train_id INTO train_id_funk
FROM schedule
WHERE route = p_route AND departure = p_departure
LIMIT 1;

-- Получение типа поезда по train_id
SELECT type INTO type_funk
FROM trains
WHERE train_id = train_id_funk
LIMIT 1;

-- Установка total_seats в зависимости от типа поезда
IF type_funk = 'Скорый' THEN
    SET total_seats = 180;
ELSEIF type_funk = 'Пассажирский' THEN
    SET total_seats = 150;
ELSE
    SET total_seats = 0; -- Значение по умолчанию, если тип
неизвестен
END IF;

-- Получение flight_id по маршруту и дате отправления
SELECT flight_id INTO flight_id_funk
FROM schedule
WHERE route = p_route AND departure = p_departure
LIMIT 1;

-- Получение количества забронированных мест по flight_id
SELECT COUNT(*) INTO blocked_seats
FROM tickets
WHERE flight_id = flight_id_funk AND status =
'Забронирован';

-- Возврат количества доступных мест
RETURN total_seats - blocked_seats;
END //

DELIMITER ;

```

На вход функция получает 2 значения: p_route (VARCHAR(500)) – 2 основные точки следования поезда и p_departure (DATETIME) – время отправления поезда, выводит на экран количество свободных мест на поезде (INT). Обусловимся, что на скором поезде в стандартной комплектации – 180 мест, а на пассажирском поезде в стандартной комплектации – 150 мест. На остальных поездах мест нет. Таким образом, используя эту функцию можно

получить количество мест на заданном поезде.

Для вызова данной функции используется запрос SQL.

```
SELECT GetAvailableSeats('Новосибирск-Главный - Бердск', '2024-10-10 08:00:00') AS Скорый;  
SELECT SLEEP(2);  
SELECT GetAvailableSeats('Новосибирск-Главный - Обь', '2024-10-11 09:00:00') AS Пассажирский;
```

Результатом работы функции является количество мест.

	Скорый
▶	179

Рисунок 3 – Результат первого вызова функции GetAvailableSeats

	Пассажирский
▶	149

Рисунок 4 – Результат второго вызова функции GetAvailableSeats

Функция CountTicketsByStatus:

```
DELIMITER //  
  
CREATE FUNCTION CountTicketsByStatus() RETURNS VARCHAR(500)  
READS SQL DATA  
BEGIN  
    DECLARE current_status VARCHAR(45);  
    DECLARE ticket_count INT;  
    DECLARE result VARCHAR(500) DEFAULT '';  
    DECLARE finished INT DEFAULT 0;  
  
    -- Курсор для перебора уникальных статусов  
    DECLARE status_cursor CURSOR FOR  
    SELECT DISTINCT status  
    FROM `Train_Station`.`Tickets`;  
  
    -- Обработка окончания курсора  
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET finished = 1;  
  
    -- Открываем курсор  
    OPEN status_cursor;  
  
    status_loop: LOOP  
        FETCH status_cursor INTO current_status;  
        IF finished THEN  
            LEAVE status_loop;  
        END IF;  
  
        -- Считаем количество билетов для текущего статуса  
        SELECT COUNT(*) INTO ticket_count
```

```

        FROM `Train_Station`.`Tickets`
        WHERE status = current_status;

        -- Формируем строку результата
        SET result = CONCAT(result, current_status, ': ',
ticket_count, '; ');
    END LOOP;

    -- Закрываем курсор
    CLOSE status_cursor;

    -- Возвращаем результат
    RETURN result;
END //

DELIMITER ;

```

Функция CountTicketsByStatus предназначена для подсчёта количества билетов по каждому уникальному статусу в таблице Train_Station.Tickets. Результатом её работы является строка, содержащая перечисление всех статусов и количество билетов для каждого из них в формате: <статус_1>: <количество_1>; <статус_2>: <количество_2>;

Задачи функции:

1. Перебрать все уникальные статусы билетов.
2. Для каждого статуса вычислить количество записей (билетов) с этим статусом.
3. Сформировать текстовую строку с итоговыми данными.

Функция также вызывается с помощью SQL-запроса
SELECT CountTicketsByStatus() AS TicketCounts;.

	TicketCounts
►	Забронирован: 2; Отменен: 1;

Рисунок 5 – Вызов функции CountTicketsByStatus

6. Вывод:

В этой работе была поставлена цель реализовать и протестировать хранимые функции для базы данных железнодорожной станции «Новосибирск-главный». Задача заключалась в создании двух пользовательских функций с использованием операторов ветвления и циклов, а также в их тестировании.

Первая функция `GetAvailableSeats` рассчитывает количество доступных мест на поезде по маршруту и времени отправления. В зависимости от типа поезда (скорый или пассажирский), она определяет количество мест и вычитает забронированные, возвращая итоговое количество доступных мест.

Вторая функция `CountTicketsByStatus` использует курсор для подсчёта количества билетов для каждого уникального статуса. Результатом работы является строка, содержащая статусы и количество билетов для каждого из них.

После разработки функций было проведено тестирование с позитивными и негативными случаями. Позитивное тестирование подтвердило корректность работы функций с валидными данными, а негативное проверило обработку ошибок при некорректных входных данных.

Обе функции продемонстрировали свою эффективность. Функция подсчёта доступных мест на поезде корректно обрабатывает различные типы поездов, а функция подсчёта билетов по статусам правильно вычисляет количество записей в базе данных.

В целом, выполнение работы показало значимость хранимых функций для обработки данных и их высокую эффективность в реальных сценариях работы с базой данных. Тестирование подтвердило правильность реализации, а предложенный функционал улучшает работу с базой и облегчает выполнение аналитических задач.