|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

**Институт информационных технологий (ИИТ)**

**Кафедра практической и прикладной информатики (ППИ)**

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

по дисциплине «Разработка баз данных»

**Практическое занятие № 3**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент группы ИНБО-01-17 | ИКБО-20-21 Сидоров С.Д. | (подпись) | |
| Преподаватель | Благовещенский В.Г. | (подпись) | |
| Отчет представлен | «\_\_\_»\_\_\_\_\_\_\_\_2023 г. | |  | |

Москва 2023 г.

**Цель занятия:**

Написание триггеров, функций и процедур различной степени сложности.

**Постановка задачи:**

Изучить и создать несколько (по 3) триггеров, функций и процедур.

Результат работы в виде отчета должен содержать:

- снимок экрана (скриншоты) триггеров, функций и процедур

- снимок экрана (скриншоты) результата работы триггеров, функций и процедур;

- снимки экрана (скриншоты)

**Результат работы:**

**Написание триггеров**

Триггер, который обновляет статус машины если появилась аренда с ней.

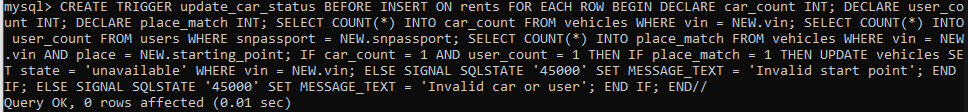


Рисунок 1 – Триггер update\_car\_status

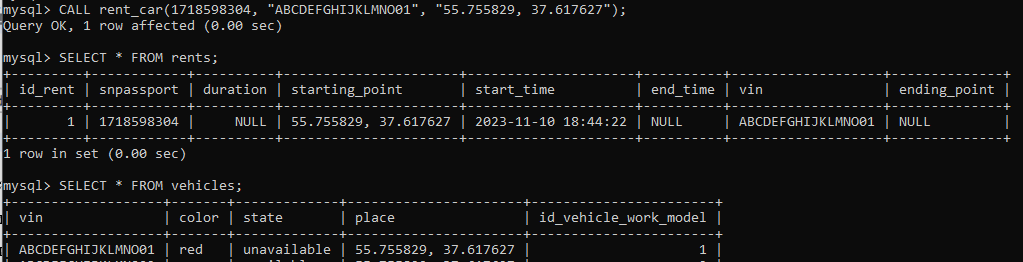


Рисунок 2 – Результат работы триггера

Триггер, который проверяет правильность данных и устанавливает базовые значения для place и state при создании новой машины представлен на рисунке 7.

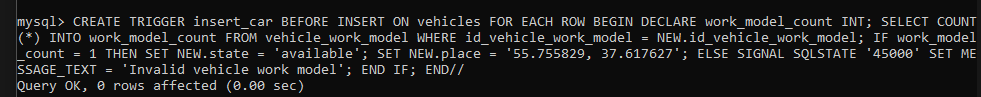


Рисунок 7 – Триггер insert\_car

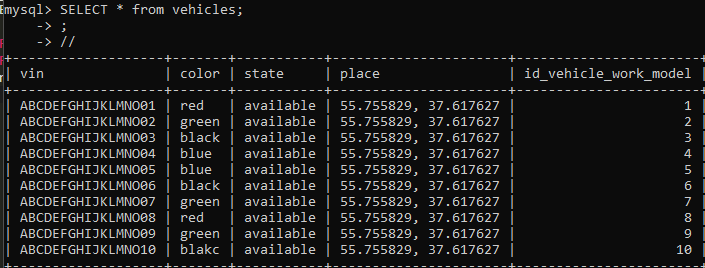


Рисунок 3 – Результат работы триггера при добавлении машины

Триггер, который после закрытия аренды устанавливает статус и место для машины, которая была задействована в аренде представлен на рисунке 11.



Рисунок 4 – Триггер close\_the\_rent

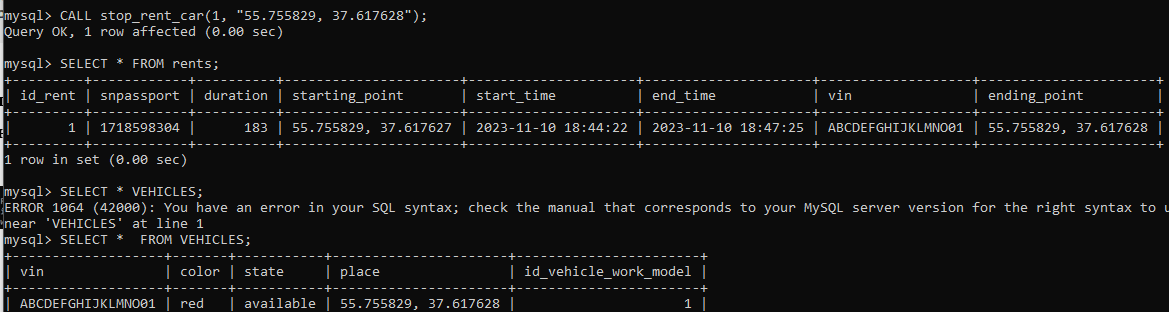


Рисунок 5 – Результат работы триггера

**Написание функций**

Функция, которая рассчитывает длительность аренды в зависимости от времени начала и конца.

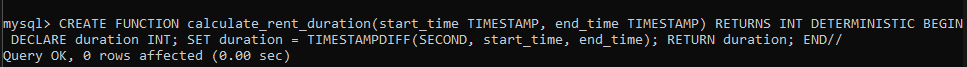


Рисунок 6 – Функция calculate\_rent\_duration

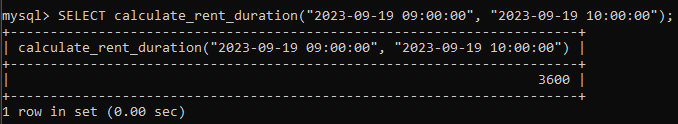


Рисунок 7 – Вызов функции

Функция, которая возвращает количество свободных машин;



Рисунок 8 – Функция get\_availeable\_vehicle\_count

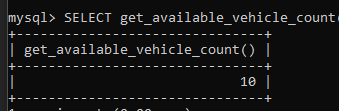


Рисунок 9 – Вызов функции

Функция, которая рассчитывает стоимость всех поездок пользователя.

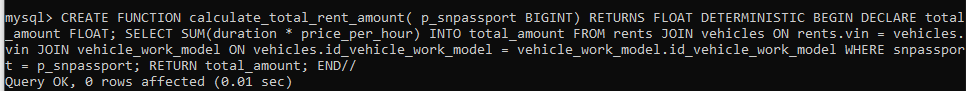


Рисунок 10 – Функция calculate\_total\_rent\_amount

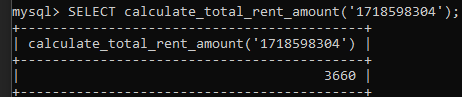


Рисунок 11 – Вызов функции

**Написание процедур**

Процедура, которая завершает аренду.

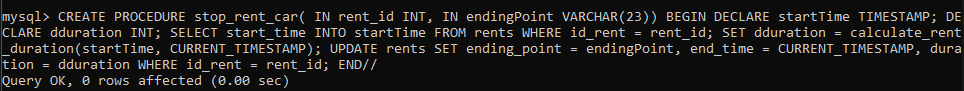


Рисунок 12 – Процедура stop\_rent\_car

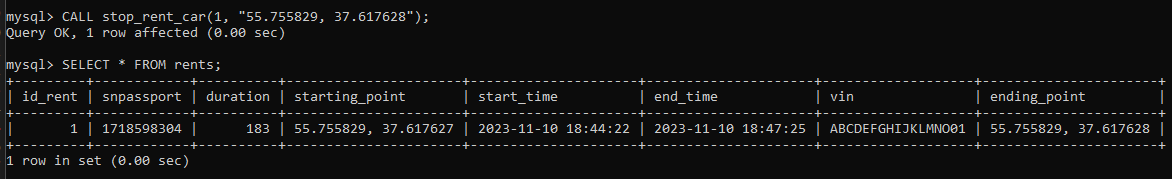


Рисунок 13 – Результат работы процедуры

Процедура, которая начинает аренду.

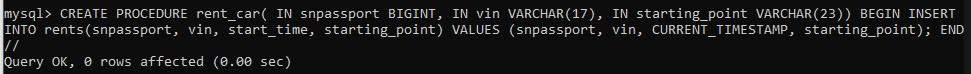


Рисунок 14 – Процедура rent\_car

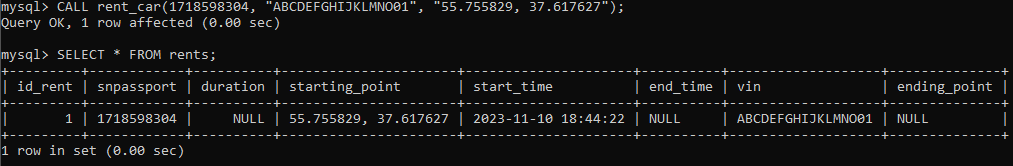


Рисунок 14 – результат вызова процедуры

Процедура, которая удаляет пользователя со всеми связанными данными.



Рисунок 15 – Процедура delete\_user

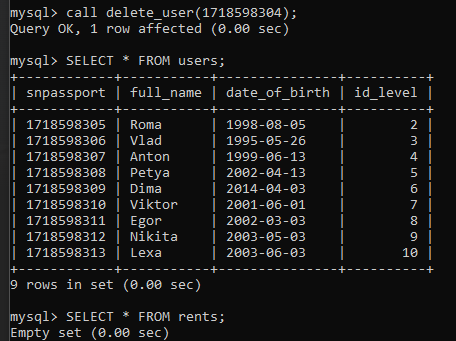


Рисунок 16 – Результат вызова процедуры

Для переноса базы данных использовался функционал MySQL Workbench – функция Export data. Результат представлен на рисунке 25.

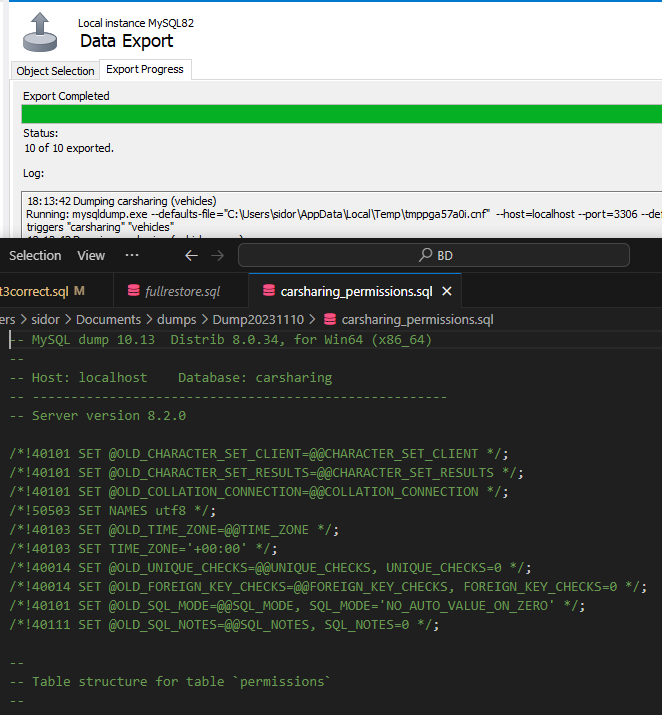


Рисунок 17 – Результат экспорта БД

Листинг 1 – Триггер 1

|  |
| --- |
| DELIMITER //  CREATE TRIGGER update\_car\_status  BEFORE INSERT ON rents  FOR EACH ROW  BEGIN  DECLARE car\_count INT;  DECLARE user\_count INT;  DECLARE place\_match INT;    SELECT COUNT(\*) INTO car\_count FROM vehicles WHERE vin = NEW.vin;  SELECT COUNT(\*) INTO user\_count FROM users WHERE snpassport = NEW.snpassport;  SELECT COUNT(\*) INTO place\_match FROM vehicles WHERE vin = NEW.vin AND place = NEW.starting\_point;    IF car\_count = 1 AND user\_count = 1 THEN  IF place\_match = 1 THEN  UPDATE vehicles SET state = 'unavailable' WHERE vin = NEW.vin;  ELSE  SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Invalid start point';  END IF;  ELSE  SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Invalid car or user';  END IF;  END//  DELIMITER ; |

Листинг 1 – Триггер 2

|  |
| --- |
| DELIMITER //  CREATE TRIGGER insert\_car  BEFORE INSERT ON vehicles  FOR EACH ROW  BEGIN  DECLARE work\_model\_count INT;  SELECT COUNT(\*) INTO work\_model\_count FROM vehicle\_work\_model WHERE id\_vehicle\_work\_model = NEW.id\_vehicle\_work\_model;    IF work\_model\_count = 1 THEN  SET NEW.state = 'available';  SET NEW.place = '55.755829, 37.617627';  ELSE  SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Invalid vehicle work model';  END IF;  END//  DELIMITER ; |

Листинг 3 – Триггер 3

|  |
| --- |
| DELIMITER //  CREATE TRIGGER update\_car\_status  BEFORE INSERT ON rents  FOR EACH ROW  BEGIN  DECLARE car\_count INT;  DECLARE user\_count INT;  DECLARE place\_match INT;    SELECT COUNT(\*) INTO car\_count FROM vehicles WHERE vin = NEW.vin;  SELECT COUNT(\*) INTO user\_count FROM users WHERE snpassport = NEW.snpassport;  SELECT COUNT(\*) INTO place\_match FROM vehicles WHERE vin = NEW.vin AND place = NEW.starting\_point;    IF car\_count = 1 AND user\_count = 1 THEN  IF place\_match = 1 THEN  UPDATE vehicles SET state = 'unavailable' WHERE vin = NEW.vin;  ELSE  SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Invalid start point';  END IF;  ELSE  SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Invalid car or user';  END IF;  END//  DELIMITER ; |

Листинг 4 – Процедура 1

|  |
| --- |
| DELIMITER //  CREATE PROCEDURE delete\_user (IN p\_snpassport BIGINT)  BEGIN  DELETE FROM rents WHERE snpassport = p\_snpassport;  DELETE FROM users  WHERE snpassport = p\_snpassport;  END//  DELIMITER ; |

Листинг 5 – Процедура 2

|  |
| --- |
| DELIMITER //  CREATE PROCEDURE rent\_car(  IN snpassport BIGINT,  IN vin VARCHAR(17),  IN starting\_point VARCHAR(23)  )  BEGIN  INSERT INTO rents (snpassport, vin, start\_time, starting\_point)  VALUES (snpassport, vin, CURRENT\_TIMESTAMP, starting\_point);  END//  DELIMITER ; |

Листинг 6 – Процедура 3

|  |
| --- |
| DELIMITER //  CREATE PROCEDURE stop\_rent\_car(  IN rent\_id INT,  IN endingPoint VARCHAR(23)  )  BEGIN  DECLARE startTime TIMESTAMP;  DECLARE dduration INT;  SELECT start\_time INTO startTime  FROM rents  WHERE id\_rent = rent\_id;  SET dduration = calculate\_rent\_duration(startTime, CURRENT\_TIMESTAMP);  UPDATE rents  SET ending\_point = endingPoint,  end\_time = CURRENT\_TIMESTAMP,  duration = dduration  WHERE id\_rent = rent\_id;  END//  DELIMITER ; |

Листинг 7 – Функция 1

|  |
| --- |
| DELIMITER //  CREATE FUNCTION calculate\_total\_rent\_amount (  p\_snpassport BIGINT  )  RETURNS FLOAT  DETERMINISTIC  BEGIN  DECLARE total\_amount FLOAT;  SELECT SUM(duration \* price\_per\_hour) INTO total\_amount  FROM rents  JOIN vehicles ON rents.vin = vehicles.vin  JOIN vehicle\_work\_model ON vehicles.id\_vehicle\_work\_model = vehicle\_work\_model.id\_vehicle\_work\_model  WHERE snpassport = p\_snpassport;  RETURN total\_amount;  END//  DELIMITER ; |

Листинг 8 – Функция 2

|  |
| --- |
| DELIMITER //  CREATE FUNCTION get\_available\_vehicle\_count()  RETURNS INT  DETERMINISTIC  BEGIN  DECLARE count INT;  SELECT COUNT(\*) INTO count  FROM vehicles  WHERE state = 'available';  RETURN count;  END//  DELIMITER ; |

Листинг 9 – Функций 3

|  |
| --- |
| DELIMITER //  CREATE FUNCTION calculate\_rent\_duration(start\_time TIMESTAMP, end\_time TIMESTAMP) RETURNS INT  DETERMINISTIC  BEGIN  DECLARE duration INT;  SET duration = TIMESTAMPDIFF(SECOND, start\_time, end\_time);  RETURN duration;  END//  DELIMITER ; |

**Вывод:**

В результате данной практической работы были созданы три процедуры, три функции и три триггера. А также создан backup для базы данных.

**Список использованных источников и литературы:**

1. Методические указания к практической работе по дисциплине “Разработка баз данных” [Электронный ресурс]: Режим доступа: <https://online-edu.mirea.ru/mod/folder/view.php?id=263588> [Дата обращения : 13.09.2023]