|  |
| --- |
| 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 задание № 2**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент группы ИНБО-01-17 | ИКБО-20-21 Сидоров С.Д. | (подпись) | |
| Преподаватель | Благовещенский В.Г | (подпись) | |
| Отчет представлен | «30» сентября 2023 г. | |  | |

Москва 2023 г.

**Отчёт**

**Цель**: изучение и создание выборки, сортировки данных. Изучение и применение операторов для изменения данных.

**Результат работы:**

Создание выборки всех колонок без фильтрации и сортировки таблицы “user\_level” с помощью команды SELECT представлено на рисунке 1.

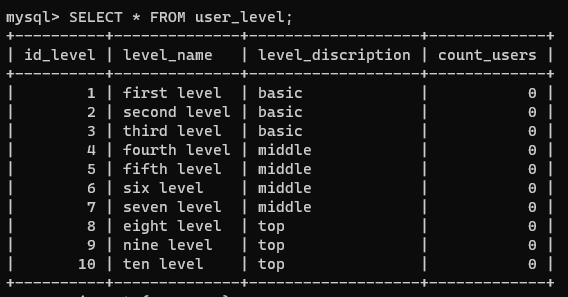


Рисунок 1 –Создание выборки

Создание выборки всех колонок с фильтром на колонку description таблицы “user\_level” с помощью команды SELECT и параметра WHERE представлено на рисунке 2.

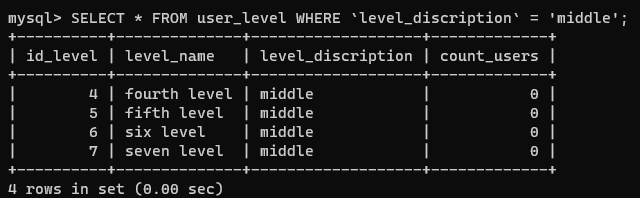


Рисунок 2 – Создание выборки с фильтрацией

Создание выборки всех колонок с фильтром на колонку description и на колонку id таблицы “user\_level” с помощью команды SELECT и параметра WHERE представлено на рисунке 3.

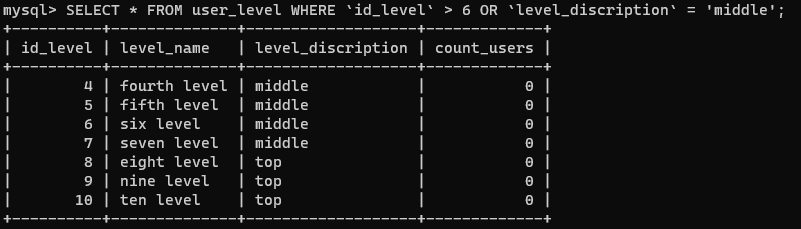


Рисунок 3 – Создание выборки с множественной фильтрацией

Создание выборки части колонок таблицы “user\_level” представлено на рисунке 4.

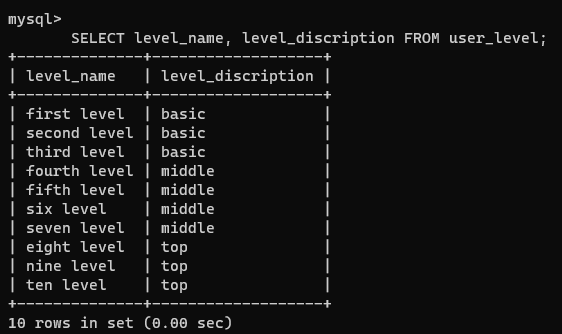


Рисунок 4 – Создание выборки с фильтрацией колонок

Создание выборки с объединением данных нескольких таблиц с помощью параметра JOIN представлено на рисунке 5.

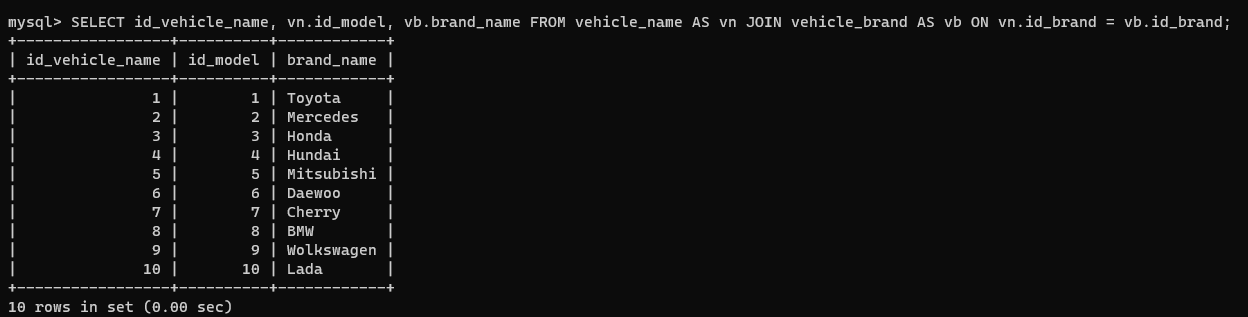


Рисунок 5 – Создание выборки с объединением

Создание выборки с множественным объединением данных нескольких таблиц с помощью параметра JOIN представлено на рисунке 6.

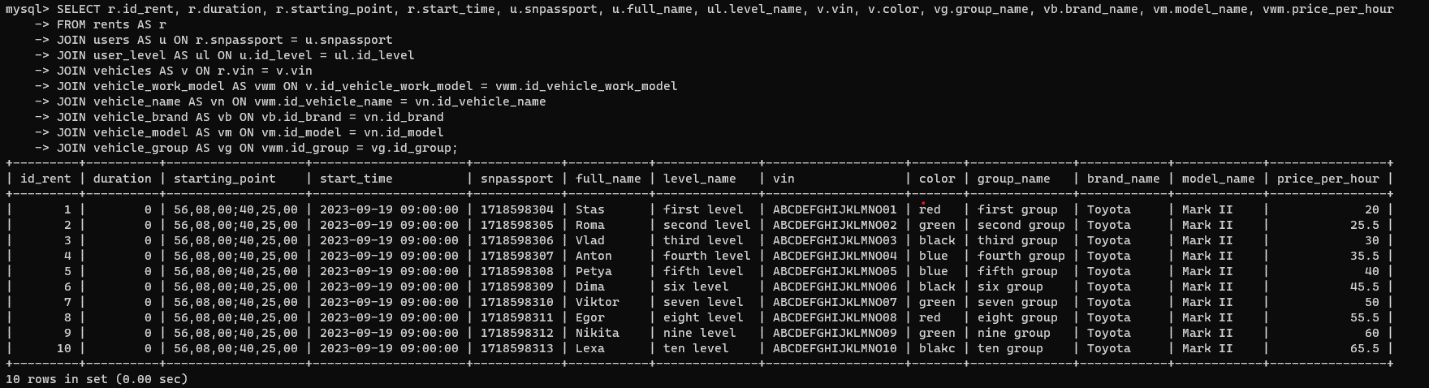


Рисунок 6 – Создание выборки с множественным объединением

Создание выборки из таблицы с использованием сортировки с помощью ORDER BY представлены на рисунке 7.



Рисунок 7 – Создание выборки с множественной сортировкой

Для обновления базы данных использовалась команда UPDATE результат выполнения представлен на рисунке 8.

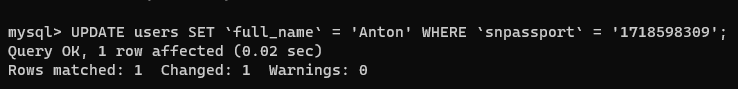


Рисунок 8 – Обновление данных в базе данных

Для удаления строки из таблицы “permissions” использовалась команда

DELETE, результат выполнения представлен на рисунке 9.

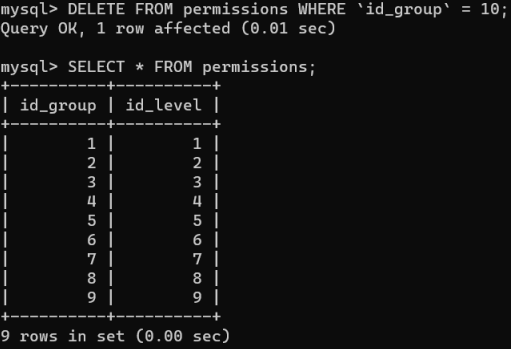


Рисунок 9 - Удаление данных из базы

Выборка внутри таблиц по определённому паттерну выполнялась с помощью параметра LIKE. (Рис. 10)

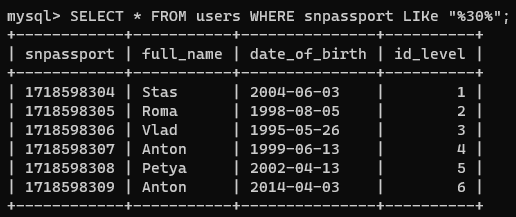


Рисунок 10 - Выборка с параметром LIKE

Содержимое команд выполнение которых было представлено выше представлено в листинге 1.

Листинг 1 – Содержимое выполненных команд.

SELECT \* FROM user\_level;

SELECT level\_name, level\_discription FROM user\_level;

SELECT \* FROM user\_level WHERE `level\_discription` = 'middle';

SELECT \* FROM user\_level WHERE `id` > 6 OR `level\_discription` = 'middle';

SELECT id\_vehicle\_name, vn.id\_model, vb.brand\_name FROM vehicle\_name AS vn JOIN vehicle\_brand AS vb ON vn.id\_brand = vb.id\_brand;

SELECT r.id\_rent, r.duration, r.starting\_point, r.start\_time, u.snpassport, u.full\_name, ul.level\_name, v.vin, v.color, vg.group\_name, vb.brand\_name, vm.model\_name, vwm.price\_per\_hour

FROM rents AS r

JOIN users AS u ON r.snpassport = u.snpassport

JOIN user\_level AS ul ON u.id\_level = ul.id\_level

JOIN vehicles AS v ON r.vin = v.vin

JOIN vehicle\_work\_model AS vwm ON v.id\_vehicle\_work\_model = vwm.id\_vehicle\_work\_model

JOIN vehicle\_name AS vn ON vwm.id\_vehicle\_name = vn.id\_vehicle\_name

JOIN vehicle\_brand AS vb ON vb.id\_brand = vn.id\_brand

JOIN vehicle\_model AS vm ON vm.id\_model = vn.id\_model

JOIN vehicle\_group AS vg ON vwm.id\_group = vg.id\_group;

Продолжение листинга 1

SELECT r.id\_rent, r.duration, r.starting\_point, r.start\_time, u.snpassport, u.full\_name, ul.level\_name, v.vin, v.color, vg.group\_name, vb.brand\_name, vm.model\_name, vwm.price\_per\_hour

FROM rents AS r

JOIN users AS u ON r.snpassport = u.snpassport

JOIN user\_level AS ul ON u.id\_level = ul.id\_level

JOIN vehicles AS v ON r.vin = v.vin

JOIN vehicle\_work\_model AS vwm ON v.id\_vehicle\_work\_model = vwm.id\_vehicle\_work\_model

JOIN vehicle\_name AS vn ON vwm.id\_vehicle\_name = vn.id\_vehicle\_name

JOIN vehicle\_brand AS vb ON vb.id\_brand = vn.id\_brand

JOIN vehicle\_model AS vm ON vm.id\_model = vn.id\_model

JOIN vehicle\_group AS vg ON vwm.id\_group = vg.id\_group

ORDER BY u.snpassport ASC;

SELECT \* FROM users ORDER BY full\_name, id\_level DESC;

SELECT \* FROM users WHERE snpassport LIKE "%30%";

UPDATE users SET `full\_name` = 'Anton' WHERE `snpassport` = '1718598309';

**Вывод:**

В результате данной практической работы была создана база данных, а также было произведено наполнение её тестовыми данными.