



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 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.

```
mysql> SELECT * FROM user_level;
```

id_level	level_name	level_discription	count_users
1	first level	basic	0
2	second level	basic	0
3	third level	basic	0
4	fourth level	middle	0
5	fifth level	middle	0
6	six level	middle	0
7	seven level	middle	0
8	eight level	top	0
9	nine level	top	0
10	ten level	top	0

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.

```
mysql> SELECT * FROM user_level WHERE 'level_discription' = 'middle';
```

id_level	level_name	level_discription	count_users
4	fourth level	middle	0
5	fifth level	middle	0
6	six level	middle	0
7	seven level	middle	0

4 rows in set (0.00 sec)

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.

```
mysql> SELECT * FROM user_level WHERE 'id_level' > 6 OR 'level_discription' = 'middle';
```

id_level	level_name	level_discription	count_users
4	fourth level	middle	0
5	fifth level	middle	0
6	six level	middle	0
7	seven level	middle	0
8	eight level	top	0
9	nine level	top	0
10	ten level	top	0

Рисунок 3 – Создание выборки с множественной фильтрацией

Создание выборки части колонок таблицы “user_level” представлено на рисунке 4.

```
mysql> SELECT level_name, level_discription FROM user_level;
```

level_name	level_discription
first level	basic
second level	basic
third level	basic
fourth level	middle
fifth level	middle
six level	middle
seven level	middle
eight level	top
nine level	top
ten level	top

10 rows in set (0.00 sec)

Рисунок 4 – Создание выборки с фильтрацией колонок

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

```
mysql> 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;
```

id_vehicle_name	id_model	brand_name
1	1	Toyota
2	2	Mercedes
3	3	Honda
4	4	Hundai
5	5	Mitsubishi
6	6	Daewoo
7	7	Cherry
8	8	BMW
9	9	Wolkswagen
10	10	Lada

10 rows in set (0.00 sec)

Рисунок 5 – Создание выборки с объединением

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

```
mysql> 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, vvm.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 vn.id_brand = vb.id_brand
```

```
JOIN vehicle_model AS vm ON vwm.id_model = vm.id_model
```

```
JOIN vehicle_group AS vg ON vwm.id_group = vg.id_group;
```

id_rent	duration	starting_point	start_time	snpassport	full_name	level_name	vin	color	group_name	brand_name	model_name	price_per_hour
1	0	56,08,00;40,25,00	2023-09-19 09:00:00	1718598304	Stas	first level	ABCDEFHIJKLMNO01	red	first group	Toyota	Mark II	20
2	0	56,08,00;40,25,00	2023-09-19 09:00:00	1718598305	Roma	second level	ABCDEFHIJKLMNO02	green	second group	Toyota	Mark II	25.5
3	0	56,08,00;40,25,00	2023-09-19 09:00:00	1718598306	Vlad	third level	ABCDEFHIJKLMNO03	black	third group	Toyota	Mark II	30
4	0	56,08,00;40,25,00	2023-09-19 09:00:00	1718598307	Anton	fourth level	ABCDEFHIJKLMNO04	blue	fourth group	Toyota	Mark II	35.5
5	0	56,08,00;40,25,00	2023-09-19 09:00:00	1718598308	Petya	fifth level	ABCDEFHIJKLMNO05	blue	fifth group	Toyota	Mark II	40
6	0	56,08,00;40,25,00	2023-09-19 09:00:00	1718598309	Dina	six level	ABCDEFHIJKLMNO06	black	six group	Toyota	Mark II	45.5
7	0	56,08,00;40,25,00	2023-09-19 09:00:00	1718598310	Viktor	seven level	ABCDEFHIJKLMNO07	green	seven group	Toyota	Mark II	50
8	0	56,08,00;40,25,00	2023-09-19 09:00:00	1718598311	Egor	eight level	ABCDEFHIJKLMNO08	red	eight group	Toyota	Mark II	55.5
9	0	56,08,00;40,25,00	2023-09-19 09:00:00	1718598312	Nikita	nine level	ABCDEFHIJKLMNO09	green	nine group	Toyota	Mark II	60
10	0	56,08,00;40,25,00	2023-09-19 09:00:00	1718598313	Lexa	ten level	ABCDEFHIJKLMNO10	black	ten group	Toyota	Mark II	65.5

10 rows in set (0.00 sec)

Рисунок 6 – Создание выборки с множественным объединением

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

```
mysql> SELECT * FROM users ORDER BY full_name, id_level DESC;
```

snpassport	full_name	date_of_birth	id_level
1718598309	Anton	2014-04-03	6
1718598307	Anton	1999-06-13	4
1718598311	Egor	2002-03-03	8
1718598313	Lexa	2003-06-03	10
1718598312	Nikita	2003-05-03	9
1718598308	Petya	2002-04-13	5
1718598305	Roma	1998-08-05	2
1718598304	Stas	2004-06-03	1
1718598310	Viktor	2001-06-01	7
1718598306	Vlad	1995-05-26	3

Рисунок 7 – Создание выборки с множественной сортировкой

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

```
mysql> UPDATE users SET `full_name` = 'Anton' WHERE `snpassport` = '1718598309';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Рисунок 8 – Обновление данных в базе данных

Для удаления строки из таблицы “permissions” использовалась команда DELETE, результат выполнения представлен на рисунке 9.

```
mysql> DELETE FROM permissions WHERE `id_group` = 10;
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM permissions;
+-----+-----+
| id_group | id_level |
+-----+-----+
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| 6 | 6 |
| 7 | 7 |
| 8 | 8 |
| 9 | 9 |
+-----+-----+
9 rows in set (0.00 sec)
```

Рисунок 9 - Удаление данных из базы

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

```
mysql> SELECT * FROM users WHERE snpassport LIKE "%30%";
```

snpassport	full_name	date_of_birth	id_level
1718598304	Stas	2004-06-03	1
1718598305	Roma	1998-08-05	2
1718598306	Vlad	1995-05-26	3
1718598307	Anton	1999-06-13	4
1718598308	Petya	2002-04-13	5
1718598309	Anton	2014-04-03	6

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.