

ru - week 8

Лабораторная работа

Тема: Работа с базами данных. SQL-запросы SELECT, агрегатные функции, JOIN

Цель работы:

Научиться использовать SQL-запросы для выборки и анализа данных из таблиц реального проекта — сервиса аренды автомобилей.

Студенты должны уметь:

- выполнять простые SELECT-запросы,
- применять группировку и агрегатные функции,
- использовать JOIN для объединения таблиц.

База данных: Автопрокат

Перед началом работы проверьте, что у вас загружены и выполнены SQL-скрипты создания и наполнения таблиц из [github репы](#) . Если что — обращайтесь к преподавателю, он уже 3 раза его исправлял :)

Раздел 1. Простые SELECT-запросы

Задание 1.1

Вывести имена и телефоны всех клиентов

```
SELECT name, phone  
FROM customers;
```

✦ *Пояснение:* простой выбор данных из таблицы `customers` . Никаких фильтров, просто познакомимся с содержимым таблицы.

Задание 1.2

Получить все бронирования с указанием даты начала и окончания аренды

✦ *Пояснение:* можно визуально прикинуть, на какие даты приходятся аренды.

Задание 1.3

Вывести все автомобили со статусом "available"

✦ *Пояснение:* используется фильтрация с `WHERE` . Выводим только свободные машины — как будто хотим взять авто напрокат.

Задание 1.4

Найти все бронирования клиента с ID = 5

Раздел 2. Агрегация + GROUP BY

Задание 2.1

Посчитать количество машин каждого бренда

```
SELECT brand, COUNT(*) AS car_count
FROM cars
GROUP BY brand;
```

✦ *Пояснение:* используем `GROUP BY` по бренду и агрегатную функцию `COUNT(*)` — получаем, сколько у нас KIA, Toyota и т.д.

Задание 2.2

Общая сумма всех платежей

✦ *Пояснение:* агрегатная функция `SUM()` — суммируем все оплаты.

Задание 2.3

Средняя стоимость бронирования

```
SELECT AVG(total_cost) AS avg_reservation_cost
FROM reservations;
```

✦ *Пояснение:* находим среднюю цену одной аренды по всем бронированиям.

Задание 2.4

Средняя цена аренды по марке автомобиля

✦ *Пояснение:* узнаем, какие машины дорогие в аренде, а какие — бюджетные.

Задание 2.5

Количество бронирований по каждому клиенту

Раздел 3. JOIN + анализ

Задание 3.1

Вывести бронирования с данными о клиенте (имя, телефон)

```
SELECT r.id, c.name, c.phone, r.start_date, r.end_date
FROM reservations r
JOIN customers c ON r.customer_id = c.id;
```

✦ *Пояснение:* объединяем `reservations` и `customers`, чтобы получить полную информацию по каждому заказу.

Задание 3.2

Показать все платежи с данными о бронировании (даты и клиент)

✦ *Пояснение:* 2 JOIN — сначала от платежей к бронированиям, потом к клиентам.

Задание 3.3

Найти клиентов, арендовавших Chevrolet Cobalt

✦ Пояснение: тут важно `DISTINCT`, чтобы избежать дублирования клиентов.

Задание 3.4

Найти бронирования длительностью более 5 дней

✦ Пояснение: функция `DATEDIFF()` считает количество дней аренды.

Задание 3.5

Вывести количество повреждений и общую стоимость ремонта по каждому автомобилю

```
SELECT car_id, COUNT(*) AS damage_count, SUM(repair_cost) AS
total_repair_cost
FROM damage_reports
GROUP BY car_id;
```

Задание 3.6

Средняя зарплата по каждой должности

Заключение

В этой лабораторной вы научились:

- извлекать данные из таблиц,
- использовать группировку и фильтры,
- комбинировать таблицы с помощью JOIN.

Если всё понятно — поздравляю, вы на шаг ближе к званию *SQL-мастер 80-го уровня*. Если что-то не ясно — не стесняйтесь спрашивать, я тут не просто так