

Министерство науки и высшего образования РФ  
Российский химико-технологический университет имени  
Д.И. Менделеева

Кафедра информационных компьютерных технологий

Отчёт по курсовой работе по дисциплине «Базы данных»  
на тему:  
«Прокат автомобилей внутренний учет.»

Исполнитель: гр. КС-34  
Ф.И.О. студента: Мохов М.Г.  
Ф.И.О. руководителя: Семёнов Г.Н.

Москва, 2024

ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ ПО  
ДИСЦИПЛИНЕ «БАЗЫ ДАННЫХ»

Студенту Мохов Марк Геннадьевич группы КС-34

Тема курсовой работы

*«Прокат автомобилей внутренний учет.»*

Этапы разработки работы

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_
4. \_\_\_\_\_
5. \_\_\_\_\_

| Основные разделы | Удельный вес раздела работы | Срок выполнения |
|------------------|-----------------------------|-----------------|
| Этап 1           |                             |                 |
| Этап 2           |                             |                 |
| Этап 3           |                             |                 |
| Этап 4           |                             |                 |
| Этап 5           |                             |                 |

Дата выдачи задания \_\_\_\_\_ 20\_ г. \_\_\_\_

Дата сдачи работы \_\_\_\_\_ 20\_ г. \_\_\_\_

Дата защиты \_\_\_\_\_ 20\_ г. \_\_\_\_

Руководитель работы \_\_\_\_\_

# СОДЕРЖАНИЕ

|  |    |
|--|----|
| Введение .....   | 2  |
| 1. Техническое задание .....   | 3  |
| 2. Инфологическая модель .....   | 3  |
| 3. Даталогическая модель .....   | 4  |
| 3.1. Схема БД .....  | 5  |
| 3.2. SQL код для создания таблиц .....   | 5  |
| 3.3. Наполнение таблиц .....   | 6  |
| 3.4. Таблицы .....   | 7  |
| 4. Результаты запросов .....   | 9  |
| 4.1. История заказов .....   | 9  |
| 4.2. Компания поставщик автомобиля .....   | 10 |
| 4.3. Запросы для определения, какие автомобили находятся на заказе, а какие стоят в гараже. .... | 12 |
| 4.4. Среднее количество выданных автомобилей за месяц .....                                      | 13 |
| 4.5. Функции и запросы для получения заказов клиента. ....                                       | 15 |
| 4.6. Запросы выдающие все автомобили взятые клиентом .....                                       | 16 |
| 5. Выводы и заключение .....   | 19 |
| 6. Список литературы .....   | 20 |

## **ВВЕДЕНИЕ**

Базы данных активно применяются в организациях самого разного вида и назначения (от ИТ организаций, до никак не связанных с информационными технологиями организаций для хранения бюрократии различного рода). Базы данных являются основным методом организации, хранения и управления информации, а также автоматизации бизнес-процессов. Процесс разработки базы данных включает в себя анализ предметной области, проектирования структуры данных, реализации связи между ними, а также создание инструментов для из сборки.

Целью этой курсовой работы является создание базы данных для организации по прокату автомобилей, а также описание процесса создания этой базы данных. По большей части в этой работе будет рассмотрен процесс создания базы данных для системы обработки заказов, которая позволит хранить историю заказов со всей необходимой информацией о заказах.

В процессе выполнения работы будет выполнено проектирование инфологической и даталогической моделей базы данных, реализация схемы базы в СУБД PostgreSQL, наполнение базы осмысленными данными, а также разработка SQL-запросов, триггеров и хранимых процедур. Результаты будут представлены в виде отчета, включающего графические и текстовые материалы.

Работа демонстрирует практическое применение знаний по проектированию баз данных и написанию SQL-запросов, а также способствует развитию навыков работы с современными инструментами для управления данными.

# 1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Тема — Прокат автомобилей внутренний учет. В задании также были представлены ключевые слова, описывающие предметную область:

**Автомобиль:** идентификационный номер, марка, компания-поставщик, число штук данного типа общее, в гараже, выдано в настоящее время, выдано всего, выдано в среднем за месяц, общий пробег; модель, длительность, мощность, год выпуска и производитель.

**Клиенты:** ФИО, паспортные данные, адрес, телефон; заказы, взятые автомобили сейчас и в прошлом: номер, дата выдачи, дата возвращения, общая стоимость заказа.

## 2. ИНФОЛОГИЧЕСКАЯ МОДЕЛЬ

Из технического задания я выделил четыре основных сущностей: Заказ, клиент, автомобиль, компания поставщик (Рисунок 2.1). Заказ является основной дочерней сущностей, так как клиент может иметь различные заказы, автомобиль может быть заказ много раз, также поставщики могут поставлять автомобили для разных заказов. Заказ имеет информацию о клиенте (**ID** клиента), об автомобиле (**VIN Номер** автомобиля), об компании поставщике (**ID** компании поставщика), также заказ имеет стоимость за день, дату выдачи и дату возвращения.

Клиент — родительская сущность, хранящая всю персональную информацию об клиенте такую, как ФИО, паспортные данные и номер телефона. В качестве ключа используется поле **ID**.

Автомобиль — родительская сущность, описывающая конкретный автомобиль. Сущность автомобиля хранит в себе информацию об пробеге автомобиля, **VIN номер** автомобиля, а также марку. Для идентификации используется поле **VIN номер** так, как он является уникальным для каждого автомобиля.

Компания поставщик — родительская сущность, описывающая компанию поставляющую автомобили. В качестве ключа используется поле **ID**.



Рисунок 2.1 – Инфологическая модель базы данных проката автомобилей.

### 3. ДАТАЛОГИЧЕСКАЯ МОДЕЛЬ

Даталогическая модель составлялась в соответствии с инфологической моделью. Сущность «Orders» соответствуют сущности «Заказ» из инфологической модели. «Orders» имеют следующие атрибуты:

1. id — идентификационное поле. Числовой тип.
2. car\_VIN — VIN номер автомобиля. Связь один ко многим. Поле является массивом символов длиной 17, в соответствии с международным стандартом.
3. issue\_date — дата выдачи заказа. Является значением типа Date (в postgresSQL этот тип отвечает за хранение информации об дате и времени).
4. return\_date — дата возвращения заказа. Так же, как и issue\_date, значение является типа Date.
5. supplier\_id — ID компании поставщика. Связь один ко многим. Является численным типом INT.
6. client\_id — ID клиента. Связь один ко многим. Является численным типом INT.

Сущность «Clients» соответствуют сущности «Клиент» из инфологической модели. «Clients» имеют следующий атрибуты:

1. id — идентификационное поле.
2. full\_name — ФИО клиента. Значение является строкой.
3. id\_number — Номер и серия паспорта.
4. id\_issue\_date — Дата выдачи паспорта.

5. `id_institution` — Код подразделения паспорта.
6. `address` — Адрес.
7. `phone` — Номер телефона клиента.

Сущность «Cars» соответствует сущности «Машина» из инфологической модели. «Cars» имеет следующие атрибуты:

1. `VIN` — VIN номер автомобиля.
2. `model` — Название модели и марки автомобиля.
3. `milage_km` — Пробег автомобиля в километрах.

Сущность «Suppliers» соответствует сущности «Компания Поставщик» из инфологической модели. «Suppliers» имеет следующие атрибуты:

1. `id` — идентификационное поле.
2. `name` — Название компании.

### 3.1. Схема БД

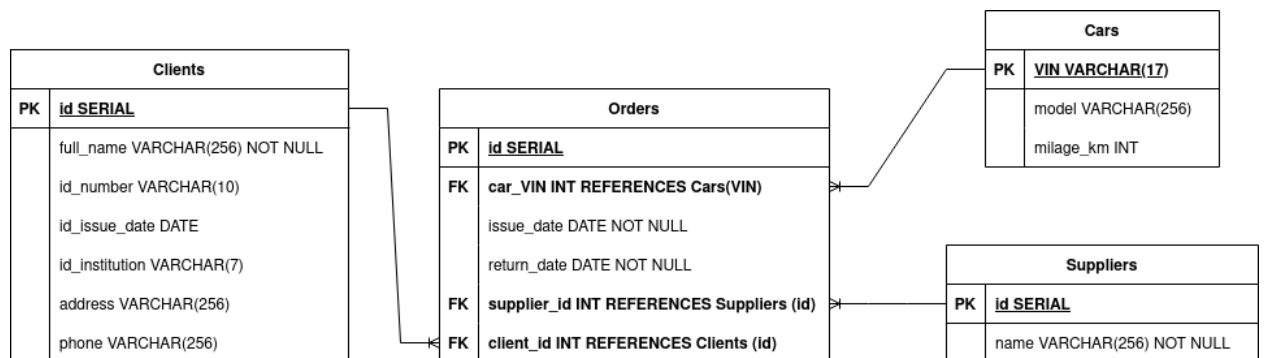


Рисунок 3.1 – Даталогическая модель.

### 3.2. SQL код для создания таблиц

```

1 DROP TABLE IF EXISTS Orders CASCADE;
2 DROP TABLE IF EXISTS Clients CASCADE;
3 DROP TABLE IF EXISTS Cars CASCADE;
4 DROP TABLE IF EXISTS Suppliers CASCADE;
5
6 CREATE TABLE Clients (
7     id SERIAL PRIMARY KEY,
8     full_name VARCHAR(256) NOT NULL,
9
10    id_number VARCHAR(10),
11    id_issue_date DATE,
12    id_institution VARCHAR(7),
13

```

sql

```

14     address VARCHAR(256),
15     phone VARCHAR(256)
16 );
17
18 CREATE TABLE Cars (
19     VIN VARCHAR(17) PRIMARY KEY,
20     model VARCHAR(256),
21     milage_km INT
22 );
23
24 CREATE TABLE Suppliers (
25     id SERIAL PRIMARY KEY,
26     name VARCHAR(256) NOT NULL
27 );
28
29 CREATE TABLE Orders (
30     id SERIAL PRIMARY KEY,
31     car_VIN VARCHAR(17) REFERENCES Cars(VIN),
32     cost INT NOT NULL,
33     issue_date DATE NOT NULL,
34     return_date DATE NOT NULL,
35     supplier_id INT REFERENCES Suppliers (id),
36     client_id INT REFERENCES Clients (id)
37 );
38
39

```

### 3.3. Наполнение таблиц

```

1  INSERT INTO Clients (full_name, id_number, id_issue_date,
2                          id_institution, address, phone)
3  VALUES ('Краснов Дмитрий Александрович', '4379123456', '2012-2-1', '723-235',
4          'г. Москва, ул. Кривая д. 23', '81321232354'),
5
6          ('Зеленов Влад Георгиевич', '4379543456', '2010-5-1', '703-235',
7          'г. Москва, ул. Кривая д. 24', '81323432354'),
8
9          ('Пятков Алексей Маркович', '4379543700', '2008-5-1', '703-235',
10         'г. Москва, ул. Ленина, д. 1', '89033432354'),
11
12         ('Титова Влада Петрова', '4379543700', '2008-5-1', '703-235',
13         'г. Москва, ул. Ленина, д. 1', '89033432354'),
14

```



```

15      ('Каролёва Лена Георгиевна', '4379543700', '2008-5-1', '703-235',
16      'г. Москва, ул. Ленина, д. 1', '89033432354');
17
18
19 INSERT INTO Cars (VIN, model, milage_km)
20 VALUES ('15AD1135931ADB325', 'Ford Focus RS', 40000),
21          ('15AD1135ABASDSAK5', 'Ford Focus RS', 401232),
22
23          ('23AD11235DAB1B145', 'Porsche 911', 232124),
24          ('2AS1293BADKAS1235', 'Porsche 911', 220242),
25          ('23AD1121ASBLAS4AB', 'Porsche 911', 208402),
26
27          ('15AD1135931ADAB49', 'KamAZ-6560', 659493),
28          ('12351135931ADAB49', 'KamAZ-6560', 593912);
29
30
31 INSERT INTO Suppliers (name)
32 VALUES ('000 Машины'), ('000 КАПСФОРЭВРИВАН'), ('000 Большие машины');
33
34
35 INSERT INTO Orders (car_VIN, cost, issue_date, return_date,
36                    supplier_id, client_id)
37 VALUES ('15AD1135931ADB325', 10000, '2022-11-2', '2022-12-5', 1, 1),
38
39          ('23AD11235DAB1B145', 50000, '2022-5-2', '2022-12-5', 2, 2),
40          ('2AS1293BADKAS1235', 50500, '2023-1-2', '2023-1-10', 2, 2),
41          ('23AD11235DAB1B145', 50100, '2023-2-25', '2023-3-2', 2, 2),
42
43          ('15AD1135931ADAB49', 101000, '2021-2-25', '2021-3-2', 3, 3),
44          ('15AD1135931ADAB49', 105000, '2022-2-25', '2022-3-2', 3, 1),
45          ('12351135931ADAB49', 105100, '2023-2-25', '2022-3-2', 3, 5),
46
47          ('12351135931ADAB49', 105100, '2023-1-2', '2023-1-10', 3, 4),
48
49          ('15AD1135931ADAB49', 105000, '2020-6-21', '2020-6-25', 2, 5);
50

```

### 3.4. Таблицы

Таблица 1 – Таблица клиентов.

| id | full_name | id_number | id_issue_date | id_institution | address | phone |
|----|-----------|-----------|---------------|----------------|---------|-------|
|----|-----------|-----------|---------------|----------------|---------|-------|

|   |                                     |            |            |         |                                   |             |
|---|-------------------------------------|------------|------------|---------|-----------------------------------|-------------|
| 1 | Краснов<br>Дмитрий<br>Александрович | 4379123456 | 2012-02-01 | 723-235 | г. Москва,<br>ул. Кривая<br>д. 23 | 81321232354 |
| 2 | Зеленов Влад<br>Георгиевич          | 4379543456 | 2010-05-01 | 703-235 | г. Москва,<br>ул. Кривая<br>д. 24 | 81323432354 |
| 3 | Пятков Алексей<br>Маркович          | 4379543700 | 2008-05-01 | 703-235 | г. Москва,<br>ул. Ленина,<br>д. 1 | 89033432354 |
| 4 | Титова Влада<br>Петрова             | 4379543700 | 2008-05-01 | 703-235 | г. Москва,<br>ул. Ленина,<br>д. 1 | 89033432354 |
| 5 | Каролёва Лена<br>Георгиевна         | 4379543700 | 2008-05-01 | 703-235 | г. Москва,<br>ул. Ленина,<br>д. 1 | 89033432354 |

Таблица 2 – Таблица машин.

| vin               | model         | milage_km |
|-------------------|---------------|-----------|
| 15AD1135931ADB325 | Ford Focus RS | 40000     |
| 15AD1135ABASDSA5  | Ford Focus RS | 401232    |
| 23AD11235DAB1B145 | Porsche 911   | 232124    |
| 2AS1293BADKAS1235 | Porsche 911   | 220242    |
| 23AD1121ASBLAS4AB | Porsche 911   | 208402    |
| 15AD1135931ADAB49 | KamAZ-6560    | 659493    |
| 12351135931ADAB49 | KamAZ-6560    | 593912    |

Таблица 3 – Таблица компаний поставщиков.

| id | name               |
|----|--------------------|
| 1  | ООО Машины         |
| 2  | ООО КАРСФОРЭВРИВАН |
| 3  | ООО Большие машины |

Таблица 4 – Таблица заказов.

| id | car_vin           | cost   | issue_date | return_date | supplier_id | client_id |
|----|-------------------|--------|------------|-------------|-------------|-----------|
| 1  | 15AD1135931ADB325 | 10000  | 2022-11-02 | 2022-12-05  | 1           | 1         |
| 2  | 23AD11235DAB1B145 | 50000  | 2022-05-02 | 2022-12-05  | 2           | 2         |
| 3  | 2AS1293BADKAS1235 | 50500  | 2023-01-02 | 2023-01-10  | 2           | 2         |
| 4  | 23AD11235DAB1B145 | 50100  | 2023-02-25 | 2023-03-02  | 2           | 2         |
| 5  | 15AD1135931ADAB49 | 101000 | 2021-02-25 | 2021-03-02  | 3           | 3         |

|   |                   |        |            |            |   |   |
|---|-------------------|--------|------------|------------|---|---|
| 6 | 15AD1135931ADAB49 | 105000 | 2022-02-25 | 2022-03-02 | 3 | 1 |
| 7 | 12351135931ADAB49 | 105100 | 2023-02-25 | 2022-03-02 | 3 | 5 |
| 8 | 12351135931ADAB49 | 105100 | 2023-01-02 | 2023-01-10 | 3 | 4 |
| 9 | 15AD1135931ADAB49 | 105000 | 2020-06-21 | 2020-06-25 | 2 | 5 |

## 4. РЕЗУЛЬТАТЫ ЗАПРОСОВ

### 4.1. История заказов

Запрос отображающий все заказы с подключением родительских таблиц.

```

1  SELECT
2      o.id AS "ID",
3      o.cost AS "Стм. за день.",
4      o.issue_date AS "Дата выдачи",
5      o.return_date AS "Дата возврата",
6      s.name AS "Компания поставщик",
7      c.model AS "Модель машины",
8      cl.full_name AS "ФИО клента"
9  FROM orders AS o
10 LEFT JOIN suppliers AS s  ON s.id = o.supplier_id
11 LEFT JOIN cars      AS c  ON c.VIN = o.car_VIN
12 LEFT JOIN clients   AS cl ON cl.id = o.client_id;
13

```

sql

Таблица 5 – Результат выполнения запроса 1.

| ID | Стм.<br>за<br>день. | Дата<br>выдачи | Дата<br>возврата | Компания поставщик | Модель<br>машины | ФИО клента                       |
|----|---------------------|----------------|------------------|--------------------|------------------|----------------------------------|
| 1  | 10000               | 2022-11-02     | 2022-12-05       | ООО Машины         | Ford Focus RS    | Краснов Дмитрий<br>Александрович |
| 2  | 50000               | 2022-05-02     | 2022-12-05       | ООО КАРСФОРЭВРИВАН | Porsche 911      | Зеленов Влад<br>Георгиевич       |
| 3  | 50500               | 2023-01-02     | 2023-01-10       | ООО КАРСФОРЭВРИВАН | Porsche 911      | Зеленов Влад<br>Георгиевич       |
| 4  | 50100               | 2023-02-25     | 2023-03-02       | ООО КАРСФОРЭВРИВАН | Porsche 911      | Зеленов Влад<br>Георгиевич       |
| 5  | 101000              | 2021-02-25     | 2021-03-02       | ООО Большие машины | KamAZ-6560       | Пятков Алексей<br>Маркович       |

|   |        |            |            |                    |            |                                  |
|---|--------|------------|------------|--------------------|------------|----------------------------------|
| 6 | 105000 | 2022-02-25 | 2022-03-02 | ООО Большие машины | KamAZ-6560 | Краснов Дмитрий<br>Александрович |
| 7 | 105100 | 2023-02-25 | 2022-03-02 | ООО Большие машины | KamAZ-6560 | Каролёва Лена<br>Георгиевна      |
| 8 | 105100 | 2023-01-02 | 2023-01-10 | ООО Большие машины | KamAZ-6560 | Титова Влада<br>Петрова          |
| 9 | 105000 | 2020-06-21 | 2020-06-25 | ООО КАРСФОРЭВРИВАН | KamAZ-6560 | Каролёва Лена<br>Георгиевна      |

## 4.2. Компания поставщик автомобиля

Представление `last_orders_of_cars` показывает для каждого автомобиля последний заказ (автомобили, которые ещё не заказывали не перечисляются).

```

1 CREATE OR REPLACE VIEW last_orders_of_cars AS
2 SELECT *
3 FROM orders AS o
4 WHERE (o.car_VIN, o.return_date) IN
5       (SELECT _o.car_VIN, MAX(_o.return_date)
6        FROM orders as _o GROUP BY _o.car_VIN);
7

```

sql

Информация о последних заказах для каждого автомобиля.

```

1 SELECT
2     c.model AS "Модель",
3     o.issue_date AS "Дата выдачи",
4     o.return_date AS "Дата выдачи",
5     o.cost AS "Цена за день",
6     o.cost * (o.return_date - o.issue_date) AS "Общая цена"
7 FROM last_orders_of_cars as o
8 LEFT JOIN cars AS c ON o.car_VIN = c.VIN;
9

```

lang

Таблица 6 – Последние заказы всех автомобилей.

| Модель        | Дата выдачи | Дата выдачи | Цена за день | Общая цена |
|---------------|-------------|-------------|--------------|------------|
| Ford Focus RS | 2022-11-02  | 2022-12-05  | 10000        | 330000     |
| Porsche 911   | 2023-01-02  | 2023-01-10  | 50500        | 404000     |
| Porsche 911   | 2023-02-25  | 2023-03-02  | 50100        | 250500     |
| KamAZ-6560    | 2022-02-25  | 2022-03-02  | 105000       | 525000     |
| KamAZ-6560    | 2023-01-02  | 2023-01-10  | 105100       | 840800     |

Функция `supplier_of_car` определяет поставщика по VIN номеру автомобиля. Функция принимает VIN номер автомобиля как входной параметр и возвращает ID поставщика, который последний раз предоставлял этот автомобиль для брони.

```

1 DROP FUNCTION IF EXISTS supplier_of_car(VARCHAR(17));
2
3 CREATE OR REPLACE FUNCTION supplier_of_car(VIN VARCHAR(17))
4 RETURNS INT
5 AS
6 $$
7 DECLARE id INT;
8 BEGIN
9
10     SELECT s.id
11         FROM last_orders_of_cars as o
12         LEFT JOIN suppliers AS s ON s.id = o.supplier_id
13         WHERE o.car_VIN = VIN
14     INTO id;
15
16     return id;
17 END
18 $$ language plpgsql;
19

```

sql

Запрос выводящий все автомобили и соответствующих им поставщиков.

```

1 SELECT
2     c.VIN    AS "VIN",
3     c.model AS "Модель",
4     s.name  AS "Компания поставщик"
5 FROM cars AS c
6 LEFT JOIN suppliers AS s ON s.id = supplier_of_car(c.VIN);
7

```

sql

Таблица 7 – Поставщики всех автомобилей.

| VIN               | Модель        | Компания поставщик |
|-------------------|---------------|--------------------|
| 15AD1135931ADB325 | Ford Focus RS | ООО Машины         |
| 15AD1135ABASDSA5  | Ford Focus RS |                    |
| 23AD11235DAB1B145 | Porsche 911   | ООО КАРСФОРЭВРИВАН |
| 2AS1293BADKAS1235 | Porsche 911   | ООО КАРСФОРЭВРИВАН |
| 23AD1121ASBLAS4AB | Porsche 911   |                    |
| 15AD1135931ADAB49 | KamAZ-6560    | ООО Большие машины |
| 12351135931ADAB49 | KamAZ-6560    | ООО Большие машины |

### 4.3. Запросы для определения, какие автомобили находятся на заказе, а какие стоят в гараже.

Функция `ordered_cars` принимает дату, как аргумент, и возвращает автомобили, которые находились на заказе в эту дату.

Функция `cars_on_parking` принимает дату, как аргумент, и возвращает автомобили, которые находились в гараже в эту дату.

Функция `cars_state` принимает дату, как аргумент, и возвращает таблицу автомобилей, с информацией о их состоянии (находится ли автомобиль на заказе или стоит в гараже).

```
1 DROP FUNCTION IF EXISTS cars_on_parking(moment Date);
2 DROP FUNCTION IF EXISTS ordered_cars(moment Date);
3 DROP FUNCTION IF EXISTS cars_state(moment Date);
4
5 CREATE OR REPLACE FUNCTION ordered_cars(moment Date)
6 RETURNS table(VIN VARCHAR(17), model VARCHAR(256), milage_km INT)
7 LANGUAGE plpgsql
8 AS $$
9 BEGIN
10     return query
11         SELECT c.VIN, c.model, c.milage_km
12         FROM cars AS c
13         RIGHT JOIN orders AS o ON o.car_VIN = c.VIN
14         WHERE o.issue_date <= moment AND o.return_date >= moment;
15 END
16 $$;
17
18 CREATE OR REPLACE FUNCTION cars_on_parking(moment Date)
19 RETURNS table(VIN VARCHAR(17), model VARCHAR(256), milage_km INT)
20 LANGUAGE plpgsql
21 AS $$
22 BEGIN
23     return query
24         SELECT *
25         FROM cars AS c
26         EXCEPT
27         SELECT * FROM ordered_cars(moment);
28 END
29 $$;
30
31
```

sql

```

32 CREATE OR REPLACE FUNCTION cars_state(moment Date)
33 RETURNS table(VIN VARCHAR(17), model VARCHAR(256), state TEXT)
34 LANGUAGE plpgsql
35 AS $$
36 BEGIN
37     return query
38         SELECT cp.VIN, cp.model, 'В гараже'
39         FROM cars_on_parking(moment) AS cp
40         UNION
41         SELECT oc.VIN, oc.model, 'На заказе'
42         FROM ordered_cars(moment) AS oc;
43 END
44 $$;
45

```

Запрос для получения состояния автомобилей на 5-е января 2023 года.

```

1 SELECT
2     cs.VIN AS "VIN",
3     cs.model AS "Модель",
4     cs.state AS "Состояние"
5 FROM cars_state('2023-1-5') as cs;
6

```

sql

Результат выполнения:

Таблица 8 – Состояние автомобилей на момент '2023-1-5'.

| VIN               | Модель        | Состояние |
|-------------------|---------------|-----------|
| 12351135931ADAB49 | KamAZ-6560    | На заказе |
| 23AD1121ASBLAS4AB | Porsche 911   | В гараже  |
| 23AD11235DAB1B145 | Porsche 911   | В гараже  |
| 15AD1135ABASDSAK5 | Ford Focus RS | В гараже  |
| 15AD1135931ADB325 | Ford Focus RS | В гараже  |
| 2AS1293BADKAS1235 | Porsche 911   | На заказе |
| 15AD1135931ADAB49 | KamAZ-6560    | В гараже  |

#### 4.4. Среднее количество выданных автомобилей за месяц

Представление ordered\_cars\_by\_monthes показывает сколько было выдано автомобилей в каждом месяце с первого заказа по последний.

```

1 CREATE OR REPLACE VIEW ordered_cars_by_monthes AS
2 SELECT
3     EXTRACT(YEAR FROM o.issue_date) AS year,

```

sql

```

4      EXTRACT(MONTH FROM o.issue_date) AS month,
5      COUNT(*) as "count"
6  FROM orders as o
7  GROUP BY year, month
8
9  UNION
10
11 SELECT
12      EXTRACT(YEAR FROM v::date) AS year,
13      EXTRACT(MONTH FROM v::date) AS month,
14      0 as "count"
15  FROM generate_series(
16      (SELECT MIN(_o.issue_date) FROM orders AS _o)::date,
17      (SELECT MAX(_o.issue_date) FROM orders AS _o)::date,
18      '1 month'::interval) v
19  WHERE NOT (EXTRACT(YEAR FROM v::date), EXTRACT(MONTH FROM v::date)) IN
20      (SELECT
21          EXTRACT(YEAR FROM __o.issue_date) AS _year,
22          EXTRACT(MONTH FROM __o.issue_date) AS _month
23      FROM orders as __o
24      GROUP BY _year, _month)
25
26  ORDER BY year, month;
27

```

Запрос, чтобы получить количество выданных автомобилей в каждом месяце.

```

1  SELECT
2      year AS "Год",
3      month AS "Месяц",
4      count AS "Кол."
5  FROM ordered_cars_by_monthes;
6

```

sql

Таблица 9 – Количество автомобилей в месяц.

| Год  | Месяц | Кол. | Год  | Месяц | Кол. | Год  | Месяц | Кол. | Год  | Месяц | Кол. |
|------|-------|------|------|-------|------|------|-------|------|------|-------|------|
| 2020 | 6     | 1    | 2020 | 7     | 0    | 2020 | 8     | 0    | 2020 | 9     | 0    |
| 2020 | 10    | 0    | 2020 | 11    | 0    | 2020 | 12    | 0    | 2021 | 1     | 0    |
| 2021 | 2     | 1    | 2021 | 3     | 0    | 2021 | 4     | 0    | 2021 | 5     | 0    |
| 2021 | 6     | 0    | 2021 | 7     | 0    | 2021 | 8     | 0    | 2021 | 9     | 0    |
| 2021 | 10    | 0    | 2021 | 11    | 0    | 2021 | 12    | 0    | 2022 | 1     | 0    |
| 2022 | 2     | 1    | 2022 | 3     | 0    | 2022 | 4     | 0    | 2022 | 5     | 1    |
| 2022 | 6     | 0    | 2022 | 7     | 0    | 2022 | 8     | 0    | 2022 | 9     | 0    |
| 2022 | 10    | 0    | 2022 | 11    | 1    | 2022 | 12    | 0    | 2023 | 1     | 2    |



|      |   |   |  |  |  |  |  |  |  |  |  |
|------|---|---|--|--|--|--|--|--|--|--|--|
| 2023 | 2 | 2 |  |  |  |  |  |  |  |  |  |
|------|---|---|--|--|--|--|--|--|--|--|--|

Для нахождения среднего количества автомобилей по месяцам, достаточно выполнить запрос:

```
1 SELECT AVG(o.count) AS ordered_cars_by_monthes;
```

sql

Результат будет: 0.2727

## 4.5. Функции и запросы для получения заказов клиента.

Функция `orders_of_client` принимает ID клиента и возвращает все заказы этого клиента:

```
1 DROP FUNCTION IF EXISTS orders_of_client(INT);
2
3 CREATE OR REPLACE FUNCTION orders_of_client(_client_id INT)
4 RETURNS table(
5     order_id INT,
6     car_VIN VARCHAR(17),
7     cost INT,
8     issue_date DATE,
9     return_date DATE,
10    supplier_id INT,
11    client_id INT
12 )
13 LANGUAGE plpgsql
14 AS $$
15 BEGIN
16     return query
17         SELECT * FROM orders
18         WHERE orders.client_id = _client_id;
19 END
20 $$;
21
```

sql

Пример использования, для нахождения всех заказов Краснова Дмитрия Александровича:

```
1 SELECT
2     c.full_name AS "ФИО",
3     cr.model AS "Модель",
4     o.cost AS "Цена за день",
5     o.issue_date AS "Дата выдачи",
6     o.return_date AS "Дата возврата"
7
```

sql

```

8 FROM orders_of_client(
9     (SELECT c.id
10        FROM clients as c
11        WHERE c.full_name = 'Краснов Дмитрий Александрович')) as o
12
13 LEFT JOIN clients AS c ON c.id = o.client_id
14 LEFT JOIN cars AS cr ON cr.VIN = o.car_VIN;
15

```

Результат выполнения:

Таблица 10 – Заказы клиента номер 1.

| ФИО                           | Модель        | Цена за день | Дата выдачи | Дата возврата |
|-------------------------------|---------------|--------------|-------------|---------------|
| Краснов Дмитрий Александрович | Ford Focus RS | 10000        | 2022-11-02  | 2022-12-05    |
| Краснов Дмитрий Александрович | KamAZ-6560    | 105000       | 2022-02-25  | 2022-03-02    |

#### 4.6. Запросы выдающие все автомобили взятые клиентом

Функция `cars_of_client` принимает ID клиента и возвращает все машины, которые клиент заказывал.

Функция `cars_of_client_by_range` принимает ID клиента, а также промежуток даты (начало и конец промежутка), и возвращает машины, которые клиент заказывал в этот промежуток.

Функция `cars_of_client_at_moment` принимает ID клиента, а также дату, и возвращает машины, которые арендовывал клиент в эту дату.

```

1 DROP FUNCTION IF EXISTS cars_of_client(INT);
2 DROP FUNCTION IF EXISTS cars_of_client_by_range(INT, DATE, DATE);
3 DROP FUNCTION IF EXISTS cars_of_client_at_moment(INT, DATE);
4
5
6 CREATE OR REPLACE FUNCTION cars_of_client(id INT)
7 RETURNS table(VIN VARCHAR(17), model VARCHAR(256))
8 LANGUAGE plpgsql
9 AS $$
10 BEGIN
11     return query
12         SELECT DISTINCT c.VIN, c.model
13         FROM cars AS c
14         WHERE c.VIN IN (SELECT o.car_VIN

```

sql

```

15             FROM orders_of_client(id) AS o);
16 END
17 $$;
18
19 CREATE OR REPLACE FUNCTION cars_of_client_by_range(id INT,
20             r_begin DATE, r_end DATE)
21 RETURNS table(VIN VARCHAR(17), model VARCHAR(256))
22 LANGUAGE plpgsql
23 AS $$
24 BEGIN
25     return query
26         SELECT DISTINCT c.VIN, c.model
27         FROM cars AS c
28         WHERE c.VIN IN (SELECT o.car_VIN
29             FROM orders_of_client(id) AS o
30             WHERE o.issue_date BETWEEN r_begin AND r_end);
31 END
32 $$;
33
34
35 CREATE OR REPLACE FUNCTION cars_of_client_at_moment(id INT, moment DATE)
36 RETURNS table(VIN VARCHAR(17), model VARCHAR(256))
37 LANGUAGE plpgsql
38 AS $$
39 BEGIN
40     return query
41         SELECT DISTINCT c.VIN, c.model
42         FROM cars AS c
43         WHERE c.VIN IN (SELECT o.car_VIN
44             FROM orders_of_client(id) AS o
45             WHERE moment BETWEEN o.issue_date AND o.return_date);
46 END
47 $$;
48

```

Чтобы получить автомобили, заказанные Пятковым Алексеем Морковичем, можно выполнить простой запрос:

```

1 SELECT
2     c.VIN AS "VIN",
3     c.model AS "Модель"
4 FROM cars_of_client(
5     (SELECT c.id
6     FROM clients as c

```

sql

```

7         WHERE c.full_name = 'Пятков Алексей Маркович')) as c;
8

```

Результат выполнения:

Таблица 11 – Автомобили, заказанные Пятковым Алексеем Морковичем.

| VIN               | Модель     |
|-------------------|------------|
| 15AD1135931ADAB49 | KamAZ-6560 |

Чтобы получать взятые автомобили за определённый период можно выполнить следующие запросы:

```

1 SELECT
2     c.VIN AS "VIN",
3     c.model AS "Модель"
4 FROM cars_of_client_by_range(
5     (SELECT c.id
6      FROM clients as c
7      WHERE c.full_name = 'Краснов Дмитрий Александрович'),
8     '2022-1-1', '2023-1-1') as c;
9

```

sql

Таблица 12 – Автомобили, заказанные Красновым Дмитрием Александровичем с '2022-1-1' по '2023-1-1'.

| VIN               | Модель        |
|-------------------|---------------|
| 15AD1135931ADAB49 | KamAZ-6560    |
| 15AD1135931ADB325 | Ford Focus RS |

Запрос, чтобы получить автомобили, заказанные Красновым Дмитрием Александровичем 2022-11-02.

```

1 SELECT c.VIN AS "VIN", c.model AS "Модель"
2 FROM cars_of_client_at_moment(
3     (SELECT c.id FROM clients as c
4     WHERE c.full_name = 'Краснов Дмитрий Александрович'),
5     '2022-11-02') as c;
6

```

sql

Таблица 13 – Автомобили клиента 1 с '2022-11-02'.

| VIN               | Модель        |
|-------------------|---------------|
| 15AD1135931ADB325 | Ford Focus RS |

## 5. ВЫВОДЫ И ЗАКЛЮЧЕНИЕ

В ходе проекта была разработана и внедрена база данных для управления компанией по выдаче автомобилей на прокат. На различных этапах проекта основное внимание уделялось организации данных, их целостности и возможностям анализа.

### **Основные результаты проекта:**

1. Проектирование структуры базы данных: Построены инфологическая и даталогическая модели построены инфологическая и даталогическая модели базы данных, учитывающие все ключевые сущности и их взаимосвязи.
2. Проведена нормализация данных для устранения избыточности.
3. Реализация базы данных: Были созданы таблицы и реализованы связи между ними в PostgreSQL с помощью DDL-скриптов.
4. Наполнение таблиц: Были разработаны скрипты DML для добавления данных о клиентах, заказах и автомобилях, обеспечивая логическую непротиворечивость данных.

**Разработка аналитических инструментов:** Были созданы SQL-запросы для получения подробной информации об автомобилях их использования, модель позволяет проводить различные срезы состояния автомобилей, клиентов и заказов в определённые периоды времени.

Переведено с помощью DeepL.com (бесплатная версия)

## **6. СПИСОК ЛИТЕРАТУРЫ**

1. Туманов В. Е. Основы проектирования реляционных баз данных. – 2007.
2. Кириллов В. В. Основы проектирования реляционных баз данных. Учебное пособие
3. Кузнецов С. Д. Основы современных баз данных
4. Кириллов В. В. Введение в реляционные базы данных. – БХВ-Петербург, 2012.
5. Осипов Д. Технологии проектирования баз данных. – Litres, 2022.
6. Стружкин Н. П., Годин В. В. Базы данных: проектирование. – 2017.