МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра цифровой трансформации (ЦТ)

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ

по дисциплине «Разработка баз данных»

Практическое занятие №1

Студенты группы *ИКБО-66-23 Смирнов А.Ю.* _____
(подпись)

Ассистент *Копылова Я.А*

_____
(подпись)

Отчет представлен «___»_____2025 г.

Москва 2025 г.

ПРАКТИЧЕСКАЯ РАБОТА №1. ОСНОВЫ DDL И ЗАПРОСЫ НА ВЫБОРКУ ДАННЫХ В POSTGRES PRO

Постановка задачи: основываясь на логической модели данных, спроектированной в рамках курса «Проектирование баз данных» в предыдущем семестре, выполните следующие шаги:
1. Письменно опишите не менее 5 различных бизнес-правил и не менее 3 ограничений целостности для таблиц. Выбор бизнес-правил и ограничений целостности производится на усмотрение студента. Результаты представить в виде таблицы.
2. С использованием DDL-оператора CREATE TABLE создать все необходимые таблицы (согласно созданной в прошлом семестре логической модели данных) в СУБД Postgres Pro, корректно реализовав все описанные ограничения целостности.
3. Заполнить созданные таблицы согласованными тестовыми данными (не менее 5-7 записей на таблицу, где это применимо) с помощью DML-оператора INSERT INTO.
4. Составить и выполнить не менее 6 SQL-запросов к таблицам, иллюстрирующих использование различных элементов списка выборки и условных выражений WHERE, согласно перечню, указанному в задании (см. Ход выполнения работы). В запросах должны быть использованы все приведённые операторы (4 для SELECT и 5 для WHERE).
5. Составить и выполнить по два SQL-запроса к таблицам для демонстрации работы ORDER BY, GROUP BY и HAVING.
6. Каждый SQL-запрос сопроводить комментарием, объясняющим его назначение и логику работы.

Таблица 1. Описание ограничений таблицы Payment_Method

Название столбца	**Тип данных**	**Ограничение**	**Обоснование (Бизнесправило)**
id_payment_method	VARCHAR(5)	PRIMARY KEY	Уникальный идентификатор способа оплаты.
reference	VARCHAR(30)	NOT NULL	Название способа оплаты (например, наличные, карта). Не может быть пустым.

Таблица 2. Описание ограничений таблицы Payment

Название столбца	**Тип данных**	**Ограничение**	**Обоснование (Бизнесправило)**
id_payment	VARCHAR(20)	PRIMARY KEY	Уникальный идентификатор платежа.
id_invoice	VARCHAR(15)	NOT NULL, FOREIGN KEY (Invoice)	Ссылка на счёт, к которому относится оплата.
id_payment_method	VARCHAR(5)	NOT NULL, FOREIGN KEY (Payment_Method)	Оплата должна быть привязана к способу оплаты.
amount	VARCHAR(2500)	NOT NULL	Сумма платежа обязательна.
payment_date	VARCHAR(24)	NOT NULL	Дата проведения платежа обязательна.
method	VARCHAR(10)	NOT NULL	Краткое обозначение способа оплаты (например, Cash, Card).

Таблица 3. Описание ограничений таблицы Employee

Название столбца	**Тип данных**	**Ограничение**	**Обоснование (Бизнесправило)**
id_employee	VARCHAR(15)	PRIMARY KEY	Уникальный идентификатор сотрудника.
first_name	VARCHAR(20)	NOT NULL	Имя сотрудника обязательно.обязательным.

last_name	VARCHAR(20)	NOT NULL	Фамилия сотрудника обязательна.
middle_name	VARCHAR(20)	NOT NULL	Отчество сотрудника обязательно.
contact	VARCHAR(30)	NOT NULL	Контактные данные обязательны.
position	VARCHAR(5)	NOT NULL	Должность обязательна.

Таблица 4. Описание ограничений таблицы Car_Type

Название столбца	**Тип данных**	**Ограничение**	**Обоснование (Бизнесправило)**
id_car_type	VARCHAR(5)	PRIMARY KEY	Уникальный идентификатор типа авто.
reference	BIGINT	NOT NULL	Код типа автомобиля обязателен.

Таблица 5. Описание ограничений таблицы Car

Название столбца	**Тип данных**	**Ограничение**	**Обоснование (Бизнесправило)**
Car_id	VARCHAR(30)	PRIMARY KEY	Уникальный идентификатор автомобиля.
id_car_type	VAR	NOT NULL,	Автомобиль

	CHAR(5)	FOREIGN KEY (Car_Type	должен относиться к определённому типу.
brand	VARCHAR(5)	NOT NULL	Марка автомобиля обязательна.
model	VARCHAR(5)	NOT NULL	Модель автомобиля обязательна.
license_plate	VARCHAR(30)	NOT NULL	Номерной знак обязателен.
status	VARCHAR(50)	NOT NULL	Статус автомобиля обязателен (например, Свободен, Забронирован).
production_year	DATE	NOT NULL	Год выпуска обязателен.

Таблица 6. Описание ограничений таблицы Booking_Status

Название столбца	**Тип данных**	**Ограничение**	**Обоснование (Бизнесправило)**
id_booking_status	VARCHAR(30)	PRIMARY KEY	Уникальный идентификатор статуса брони.
status	VARCHAR(50)	NOT NULL	Название статуса обязательно (Активна,

			Отменена, Завершена). обязательным параметром.

Таблица 7. Описание ограничений Invoice

Название столбца	**Тип данных**	**Ограничение**	**Обоснование (Бизнесправило)**
id_invoice	VARCHAR(20)	PRIMARY KEY	Уникальный идентификатор счёта.
cost	VARCHAR(2500)	NOT NULL	Стоимость обязательна.

Таблица 8. Описание ограничений таблицы Client

Название столбца	**Тип данных**	**Ограничение**	**Обоснование (Бизнесправило)**
Client_id	VARCHAR(15)	PRIMARY KEY	Уникальный идентификатор клиента.
first_name	VARCHAR(20)	NOT NULL,	Имя клиента обязательно.
last_name	VARCHAR(20)	NULL	Отчество клиента отсутствовать.
driver_license	VARCHAR(15)	NOT NULL	Водительское удостоверение обязательно. находиться в допустимом

			диапазоне (например, от 1 до 10).
contact	VARCHAR(50)	NO T NULL	Контактные данные обязательны.

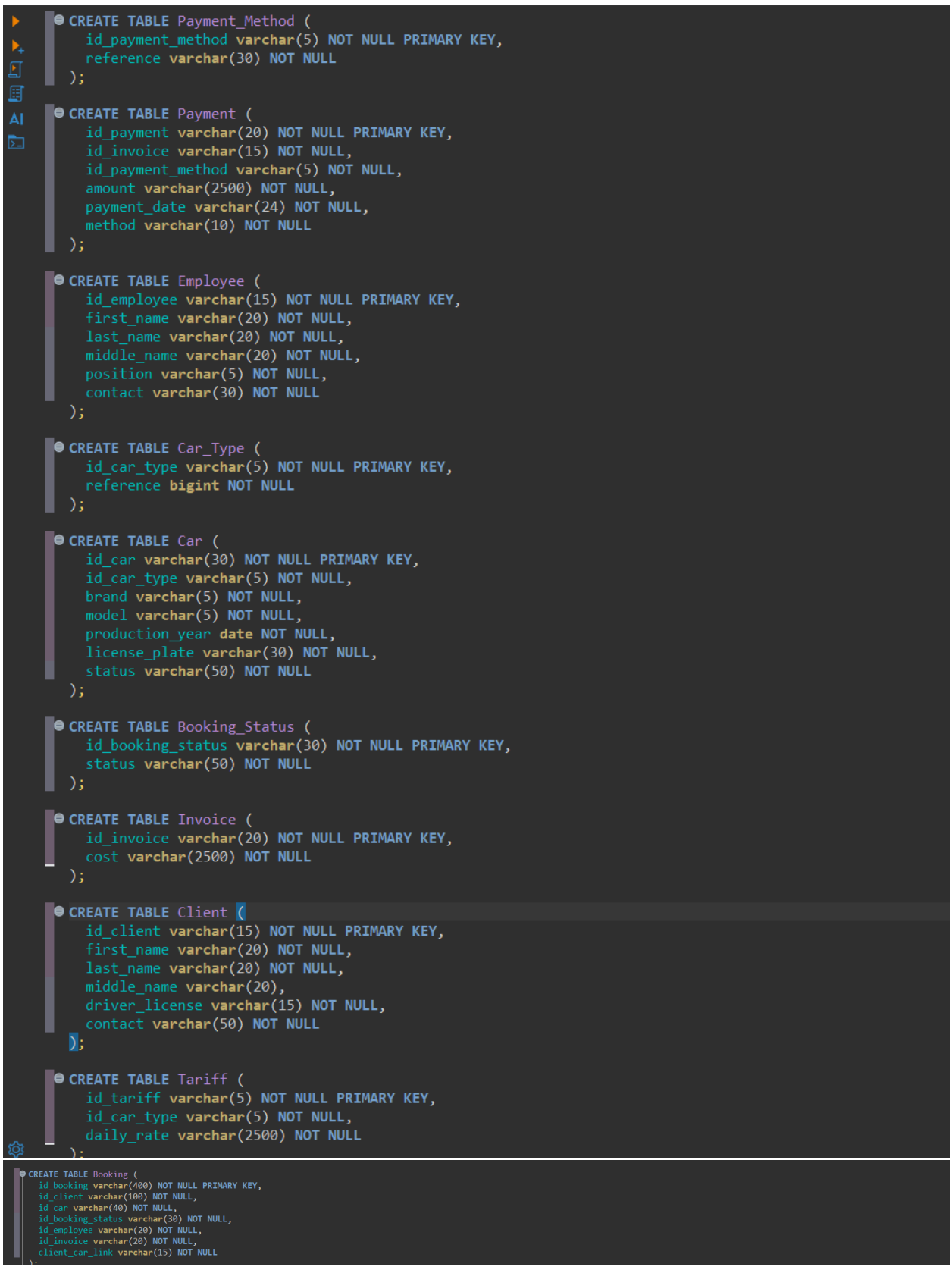
*Таблица 9. Описание ограничений таблицы **Tariff***

Название столбца	**Тип данных**	**Ограничение**	**Обоснование (Бизнесправило)**
genre_id	VARCHAR(5)	PRIMARY KEY	Уникальный идентификатор жанра, генерируется автоматически.
id_car_type	VARCHAR(5)	NOT NULL, FOREIGN KEY (Car_Type)	Тариф должен относиться к типу автомобиля.
daily_rate	VARCHAR(2500)	NOT NULL	Стоимость аренды в сутки обязательна.

Таблица 10. Описание ограничений таблицы movie_genre_link

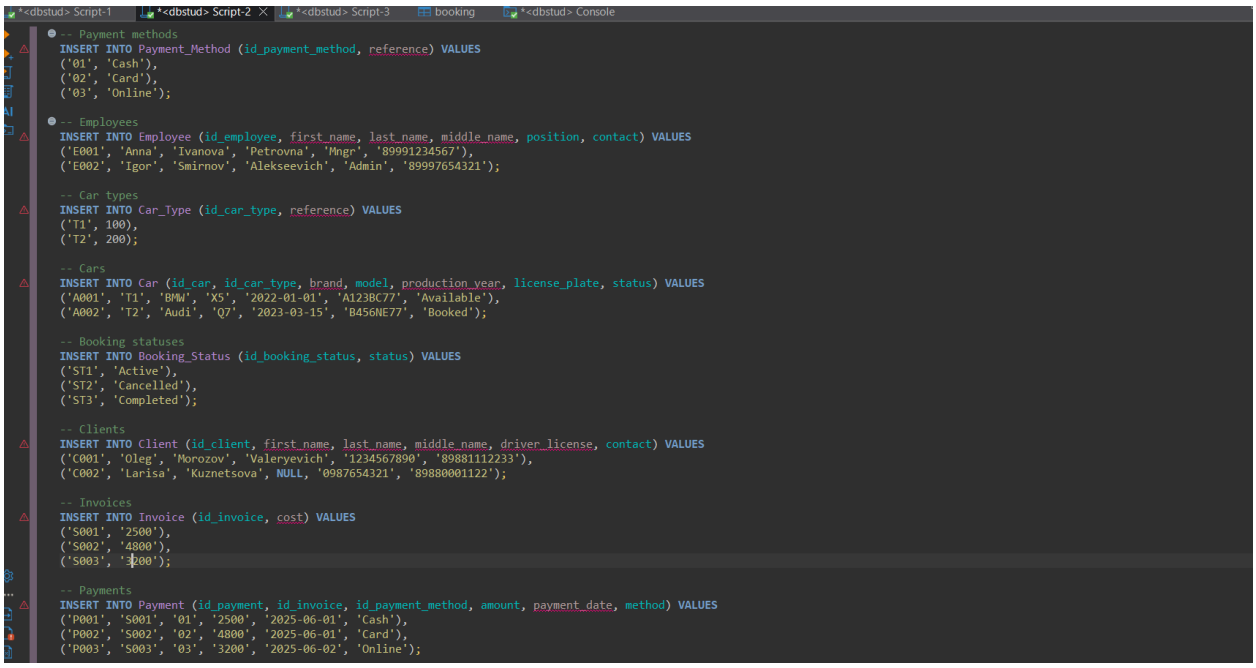
Название столбца	**Тип данных**	**Ограничение**	**Обоснование (Бизнесправило)**
id_booking	VARCHAR(400)	PRIMARY KEY	Уникальный идентификатор связи,

			генерируется автоматически.
id_client	VARCHAR(100)	NOT NULL, FOREIGN KEY (Client)	Каждая бронь связана с клиентом.
id_car	VARCHAR(40)	NOT NULL, FOREIGN KEY (Car)	Бронирование обязательно связано с автомобилем.
id_invoice	VARCHAR(20)	NOT NULL, FOREIGN KEY (Invoice)	Каждое бронирование связано со счётом.
id_employee	VARCHAR(20)	NOT NULL, FOREIGN KEY (Employee)	Бронь оформляет сотрудник.
client_car_link	VARCHAR(15)	NOT NULL	Служебное поле для связи клиента и авто.
id_booking_status	VARCHAR(30)	NOT NULL, FOREIGN KEY (Booking_Status)	Каждая бронь имеет статус.



```sql
CREATE TABLE Payment_Method (
    id_payment_method varchar(5) NOT NULL PRIMARY KEY,
    reference varchar(30) NOT NULL
);

CREATE TABLE Payment (
    id_payment varchar(20) NOT NULL PRIMARY KEY,
    id_invoice varchar(15) NOT NULL,
    id_payment_method varchar(5) NOT NULL,
    amount varchar(2500) NOT NULL,
    payment_date varchar(24) NOT NULL,
    method varchar(10) NOT NULL
);

CREATE TABLE Employee (
    id_employee varchar(15) NOT NULL PRIMARY KEY,
    first_name varchar(20) NOT NULL,
    last_name varchar(20) NOT NULL,
    middle_name varchar(20) NOT NULL,
    position varchar(5) NOT NULL,
    contact varchar(30) NOT NULL
);

CREATE TABLE Car_Type (
    id_car_type varchar(5) NOT NULL PRIMARY KEY,
    reference bigint NOT NULL
);

CREATE TABLE Car (
    id_car varchar(30) NOT NULL PRIMARY KEY,
    id_car_type varchar(5) NOT NULL,
    brand varchar(5) NOT NULL,
    model varchar(5) NOT NULL,
    production_year date NOT NULL,
    license_plate varchar(30) NOT NULL,
    status varchar(50) NOT NULL
);

CREATE TABLE Booking_Status (
    id_booking_status varchar(30) NOT NULL PRIMARY KEY,
    status varchar(50) NOT NULL
);

CREATE TABLE Invoice (
    id_invoice varchar(20) NOT NULL PRIMARY KEY,
    cost varchar(2500) NOT NULL
);

CREATE TABLE Client (
    id_client varchar(15) NOT NULL PRIMARY KEY,
    first_name varchar(20) NOT NULL,
    last_name varchar(20) NOT NULL,
    middle_name varchar(20),
    driver_license varchar(15) NOT NULL,
    contact varchar(50) NOT NULL
);

CREATE TABLE Tariff (
    id_tariff varchar(5) NOT NULL PRIMARY KEY,
    id_car_type varchar(5) NOT NULL,
    daily_rate varchar(2500) NOT NULL
);

CREATE TABLE Booking (
    id_booking varchar(400) NOT NULL PRIMARY KEY,
    id_client varchar(100) NOT NULL,
    id_car varchar(40) NOT NULL,
    id_booking_status varchar(30) NOT NULL,
    id_employee varchar(20) NOT NULL,
    id_invoice varchar(20) NOT NULL,
    client_car_link varchar(15) NOT NULL
);
```

Рис.1-2 Скрипт создания таблиц



```sql
-- Payment methods
INSERT INTO Payment_Method (id_payment_method, reference) VALUES
('01', 'Cash'),
('02', 'Card'),
('03', 'Online');

-- Employees
INSERT INTO Employee (id_employee, first_name, last_name, middle_name, position, contact) VALUES
('E001', 'Anna', 'Ivanova', 'Petrovna', 'Mngr', '89991234567'),
('E002', 'Igor', 'Smirnov', 'Alekseevich', 'Admin', '89997654321');

-- Car types
INSERT INTO Car_Type (id_car_type, reference) VALUES
('T1', 100),
('T2', 200);

-- Cars
INSERT INTO Car (id_car, id_car_type, brand, model, production_year, license_plate, status) VALUES
('A001', 'T1', 'BMW', 'X5', '2022-01-01', 'A123BC77', 'Available'),
('A002', 'T2', 'Audi', 'Q7', '2023-03-15', 'B456NE77', 'Booked');

-- Booking statuses
INSERT INTO Booking_Status (id_booking_status, status) VALUES
('ST1', 'Active'),
('ST2', 'Cancelled'),
('ST3', 'Completed');

-- Clients
INSERT INTO Client (id_client, first_name, last_name, middle_name, driver_license, contact) VALUES
('C001', 'Oleg', 'Morozov', 'Valeryevich', '1234567890', '89881112233'),
('C002', 'Larisa', 'Kuznetsova', NULL, '0987654321', '89880001122');

-- Invoices
INSERT INTO Invoice (id_invoice, cost) VALUES
('S001', '2500'),
('S002', '4800'),
('S003', '3200');

-- Payments
INSERT INTO Payment (id_payment, id_invoice, id_payment_method, amount, payment_date, method) VALUES
('P001', 'S001', '01', '2500', '2025-06-01', 'Cash'),
('P002', 'S002', '02', '4800', '2025-06-01', 'Card'),
('P003', 'S003', '03', '3200', '2025-06-02', 'Online');
```

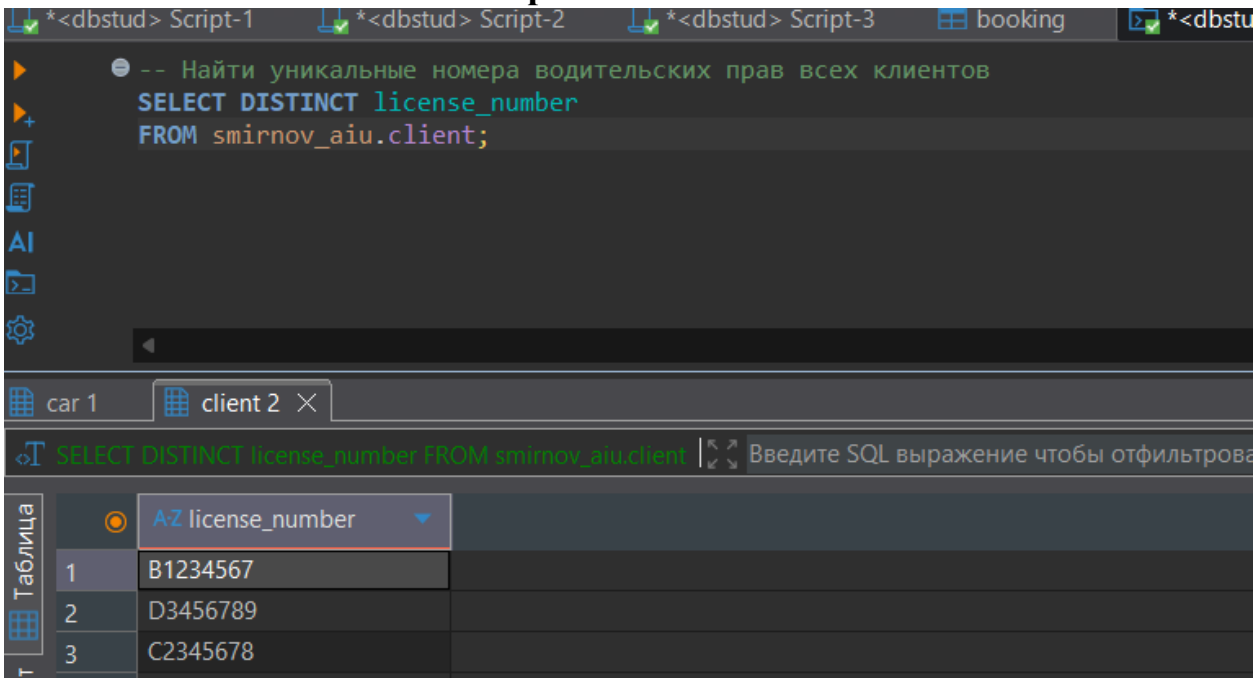
Рис 3 — Скрипт заполнения таблиц

```sql
ALTER TABLE Car
    ADD CONSTRAINT Car_id_car_type_fk FOREIGN KEY (id_car_type) REFERENCES Car_Type (id_car_type);

ALTER TABLE Booking
    ADD CONSTRAINT Booking_id_car_fk FOREIGN KEY (id_car) REFERENCES Car (id_car);

ALTER TABLE Booking
    ADD CONSTRAINT Booking_id_client_fk FOREIGN KEY (id_client) REFERENCES Client (id_client);

ALTER TABLE Booking
    ADD CONSTRAINT Booking_id_employee_fk FOREIGN KEY (id_employee) REFERENCES Employee (id_employee);

ALTER TABLE Booking
    ADD CONSTRAINT Booking_id_booking_status_fk FOREIGN KEY (id_booking_status) REFERENCES Booking_Status (id_booking_status);

ALTER TABLE Booking
    ADD CONSTRAINT Booking_id_invoice_fk FOREIGN KEY (id_invoice) REFERENCES Invoice (id_invoice);

ALTER TABLE Payment
    ADD CONSTRAINT Payment_id_payment_method_fk FOREIGN KEY (id_payment_method) REFERENCES Payment_Method (id_payment_method);

ALTER TABLE Payment
    ADD CONSTRAINT Payment_id_invoice_fk FOREIGN KEY (id_invoice) REFERENCES Invoice (id_invoice);

ALTER TABLE Tariff
    ADD CONSTRAINT Tariff_id_car_type_fk FOREIGN KEY (id_car_type) REFERENCES Car_Type (id_car_type);
```

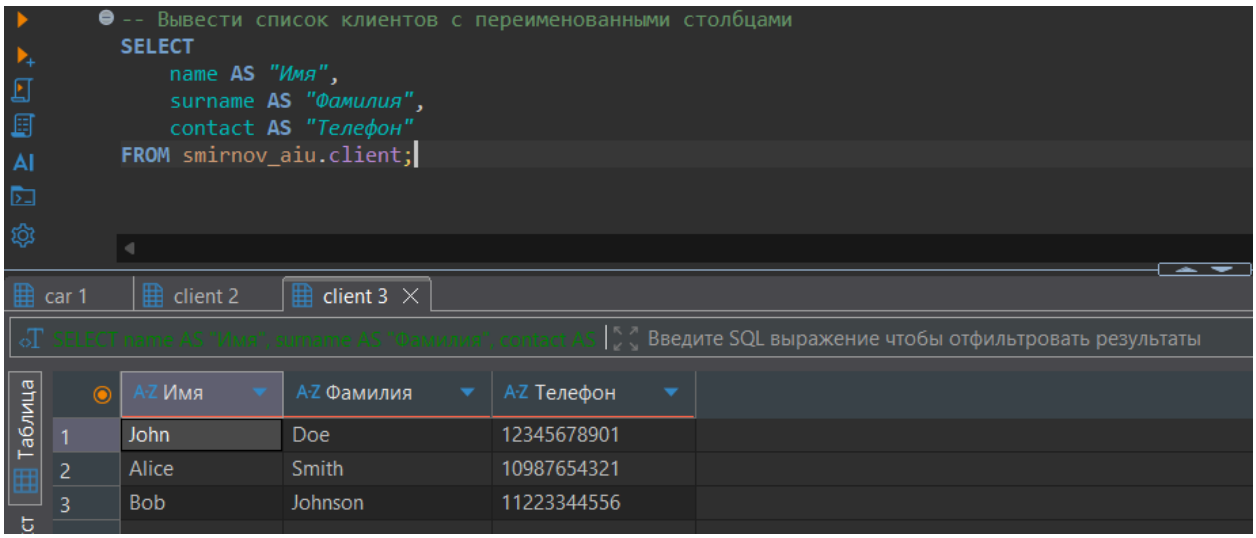
Рис 4 — Скрипт реализующий ограничения целостности

Выполнение пунктов 4-6:
Составить и выполнить не менее 6 SQL-запросов к таблицам, иллюстрирующих использование различных элементов списка выборки и условных выражений WHERE, согласно перечню, указанному в задании (см. Ход выполнения работы). В запросах должны быть использованы все приведённые операторы (4 для SELECT и 5 для WHERE).

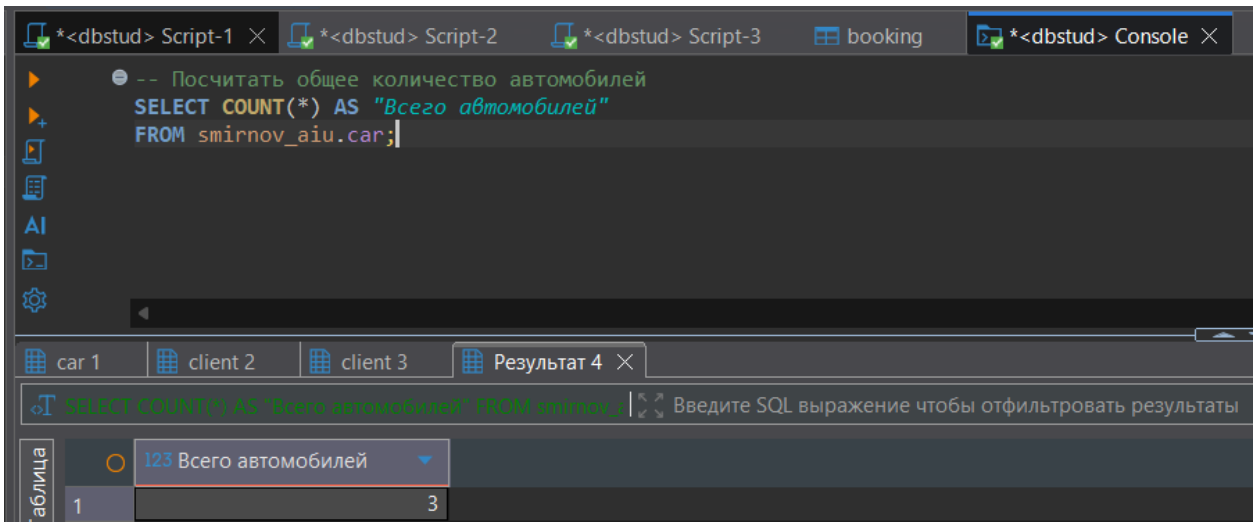
Запрос с DISTINCT



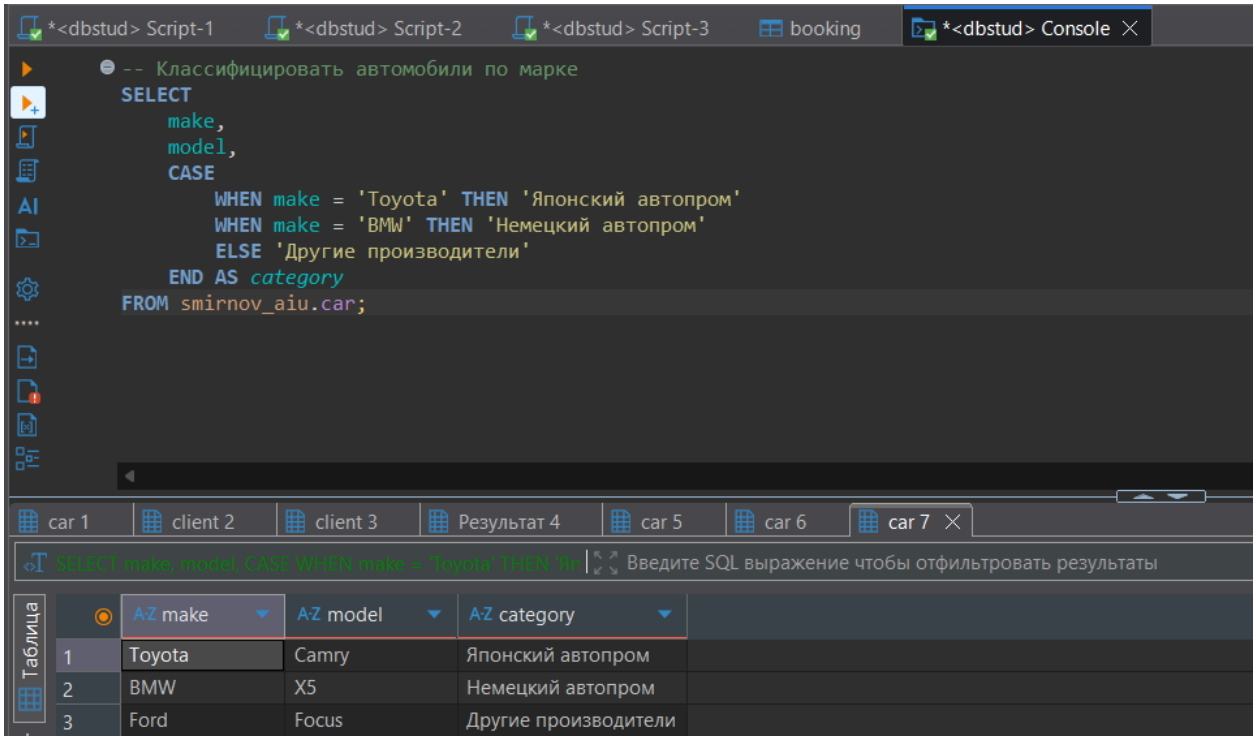
Показывает все уникальные номера водительских прав без повторений



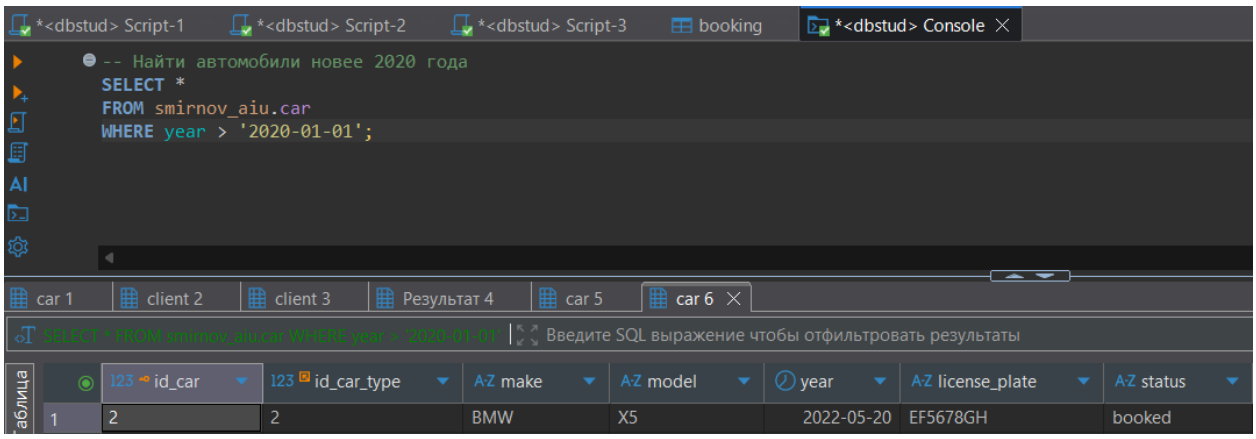
Выводит информацию о клиентах



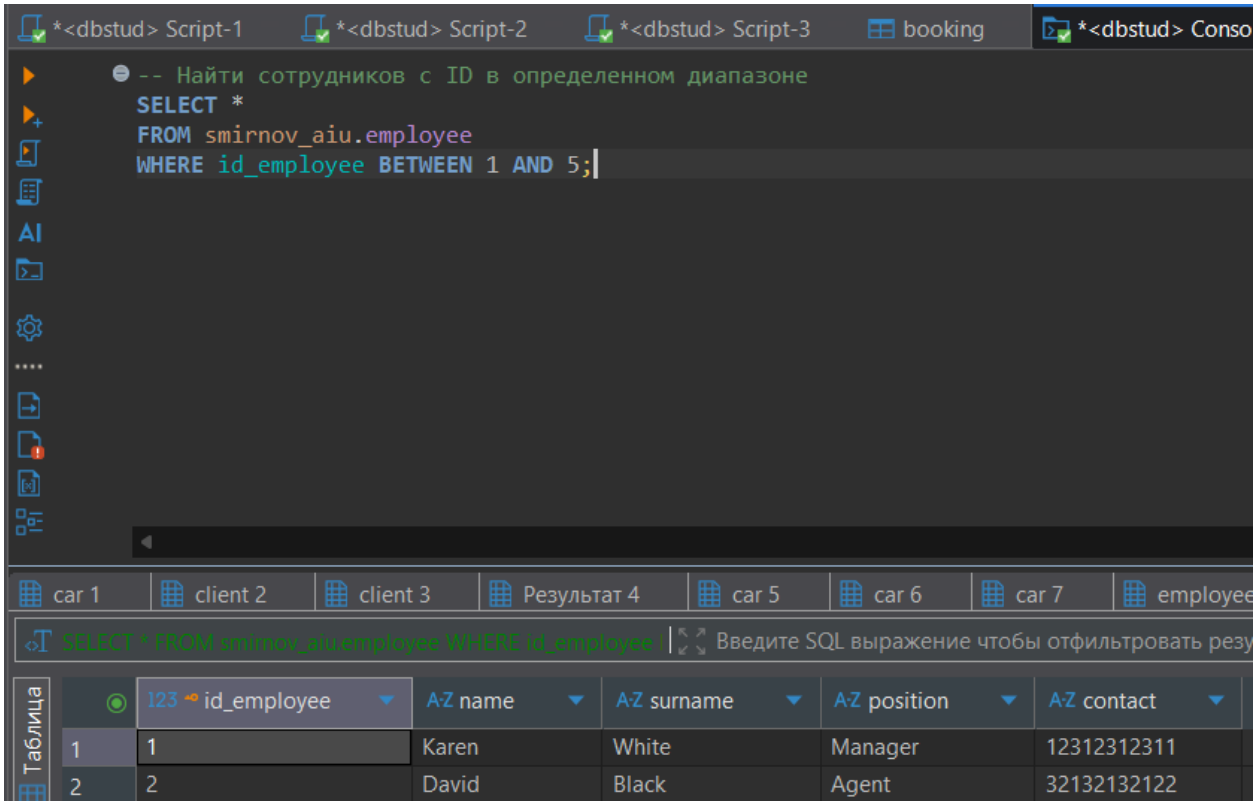
Подсчитывает общее количество автомобилей в базе данных.



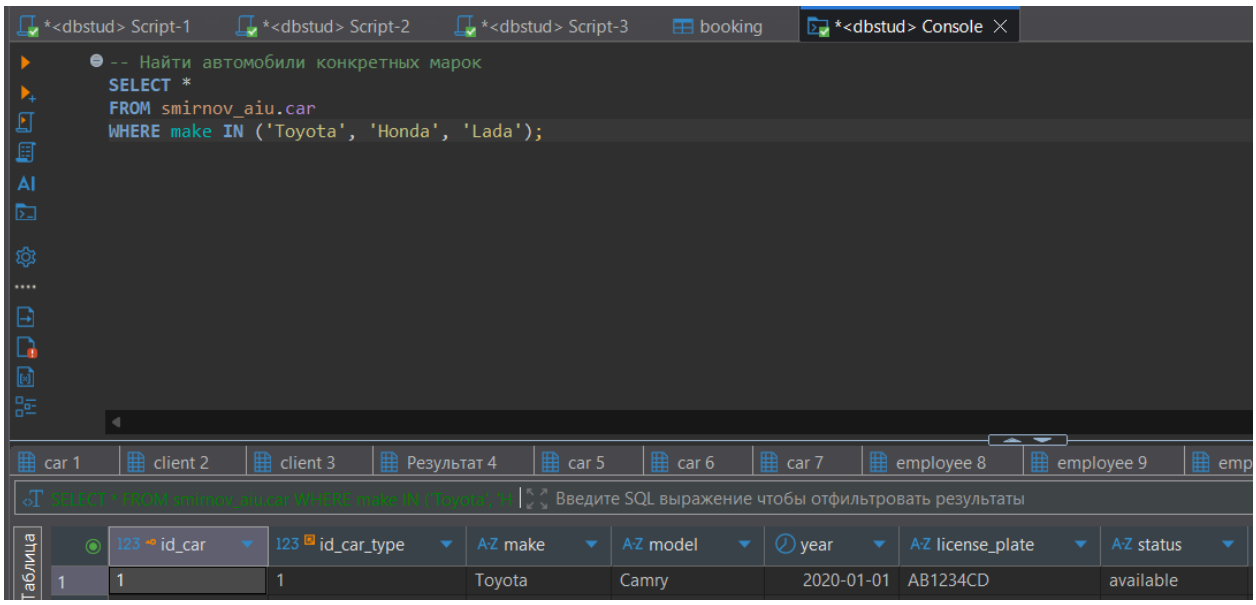
Разделяет автомобили на категории в зависимости от марки



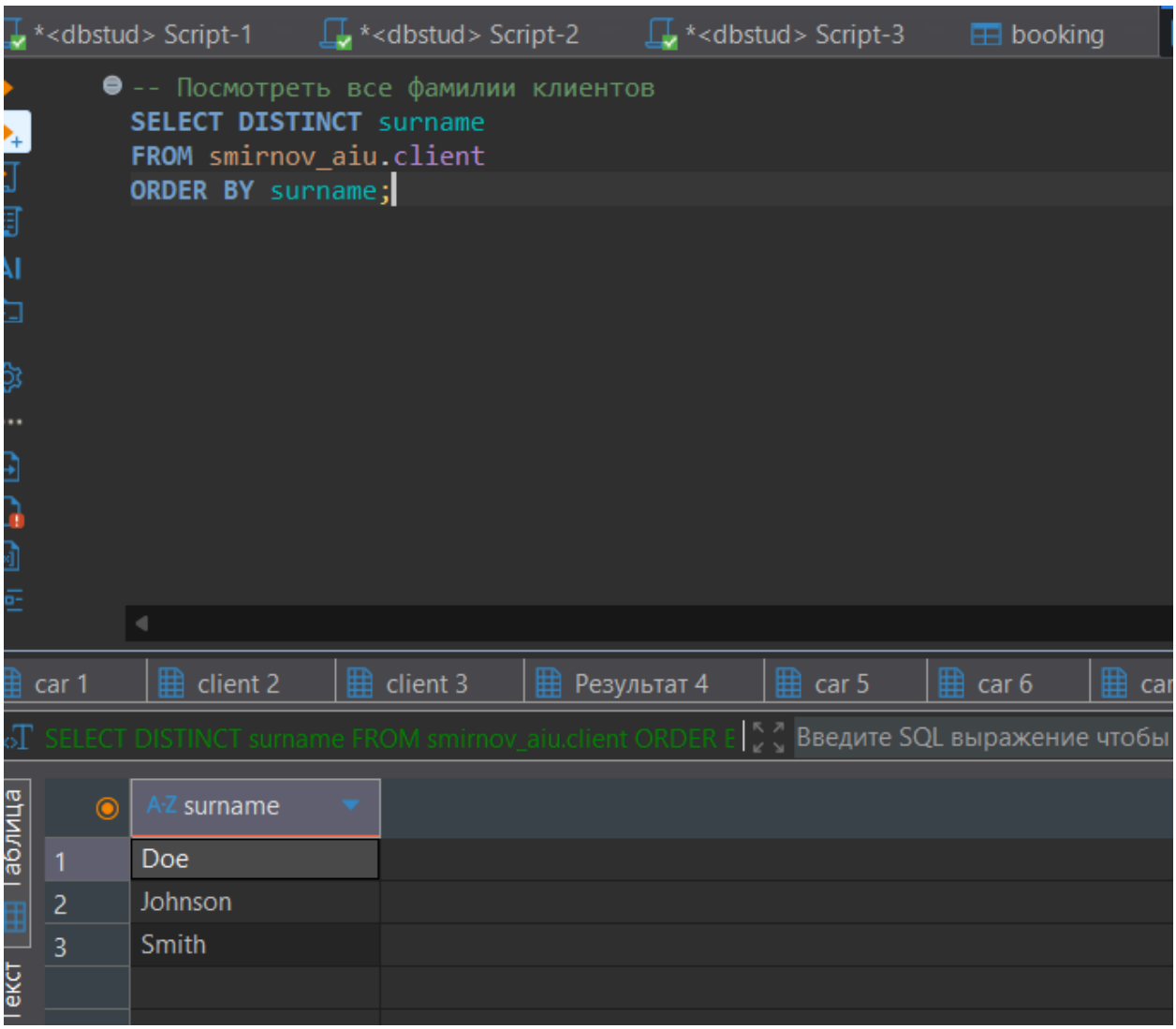
Выводит автомобили, выпущенные после 2020 года.



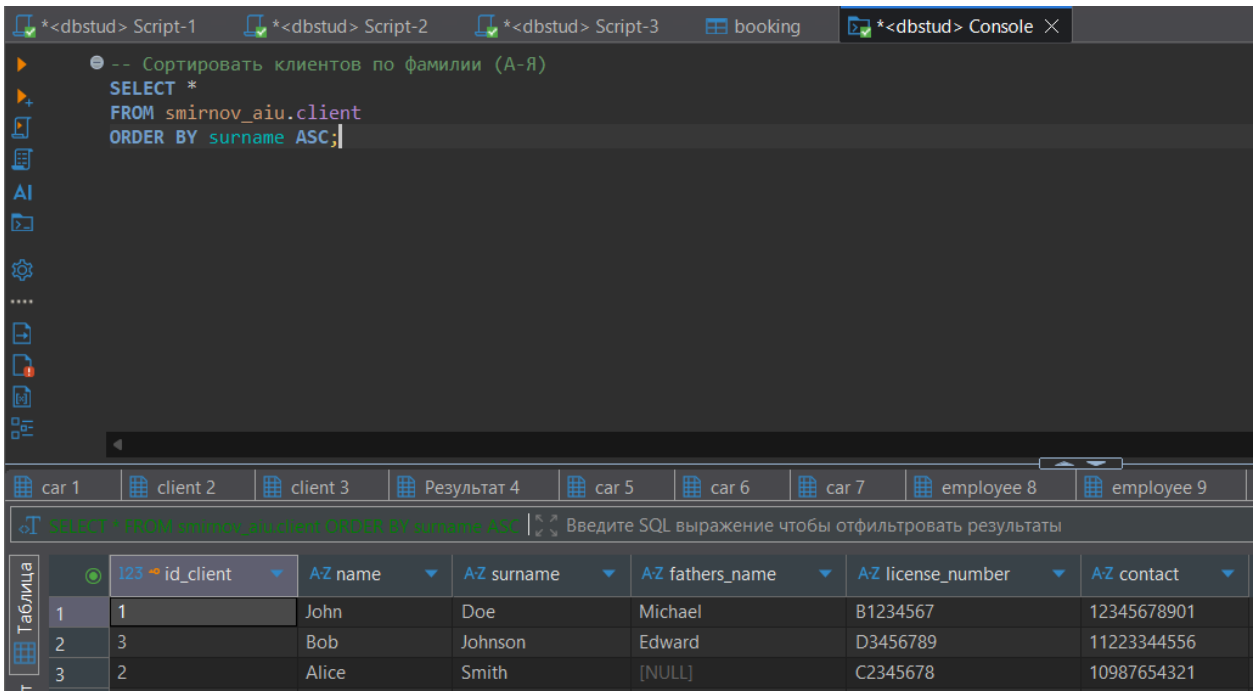
Выводит сотрудников с идентификаторами в указанном диапазоне.



Показывает автомобили только указанных марок.

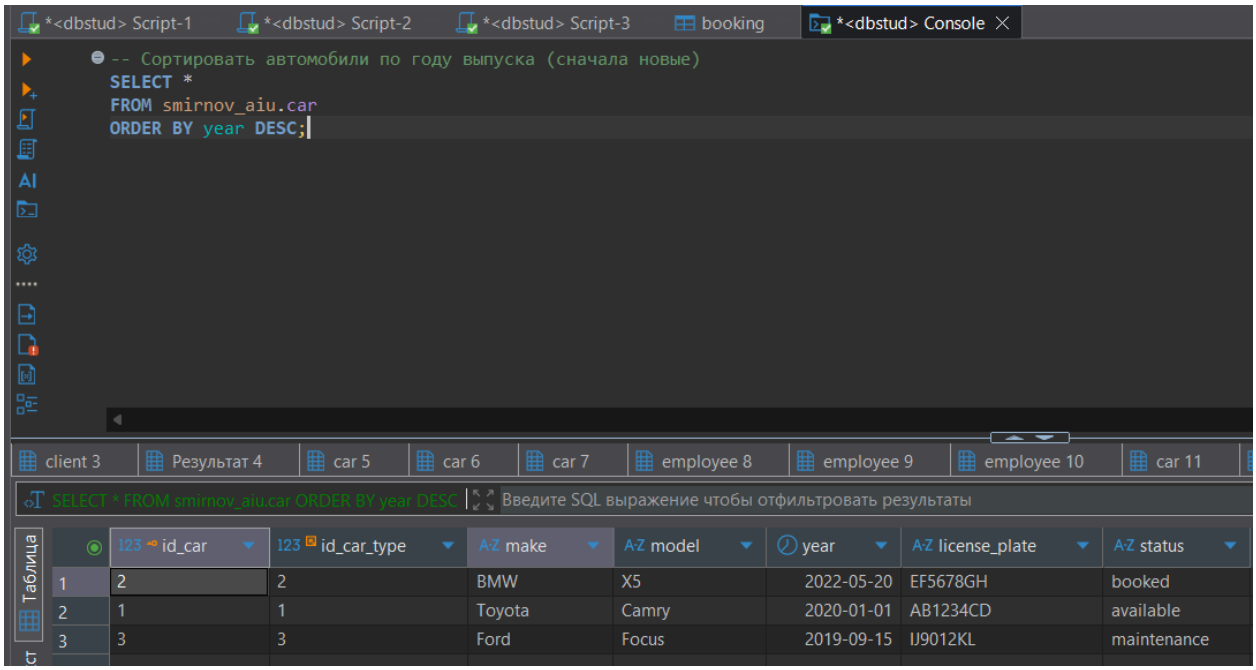


Сначала посмотреть какие фамилии вообще есть в базе.

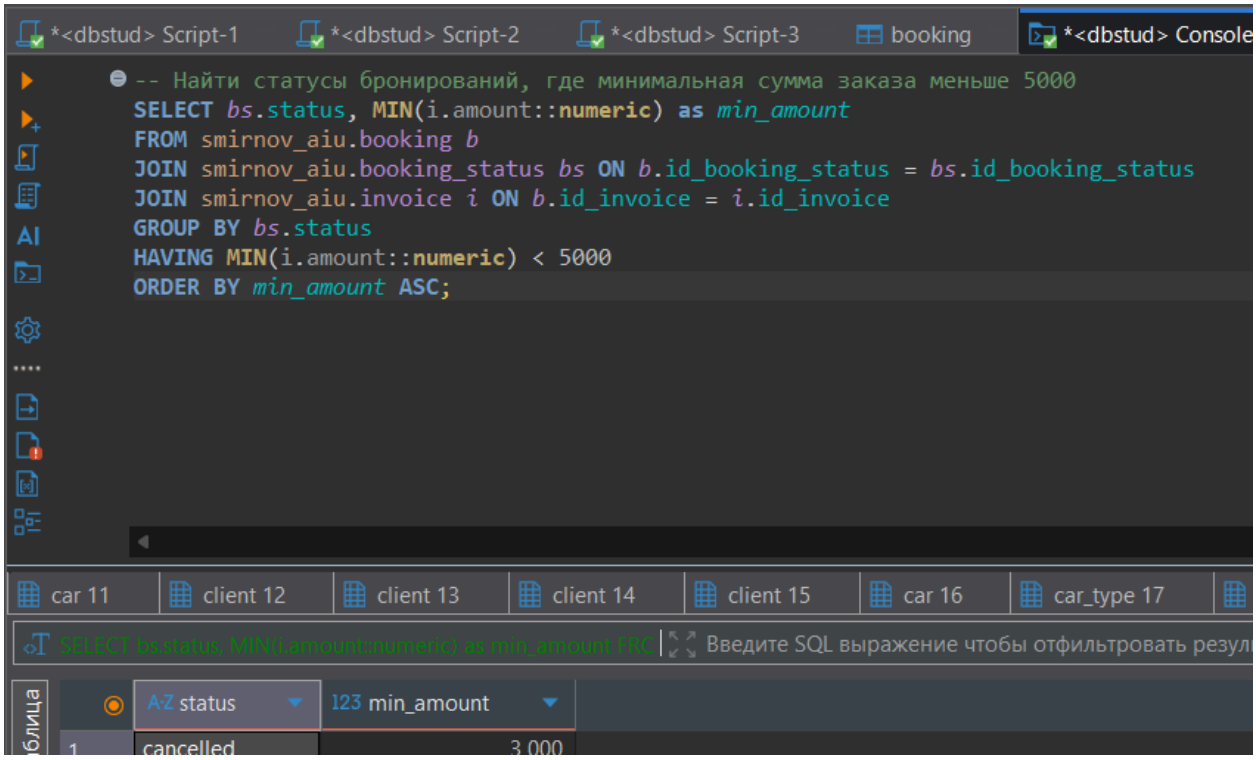


Сортировать клиентов по фамилии

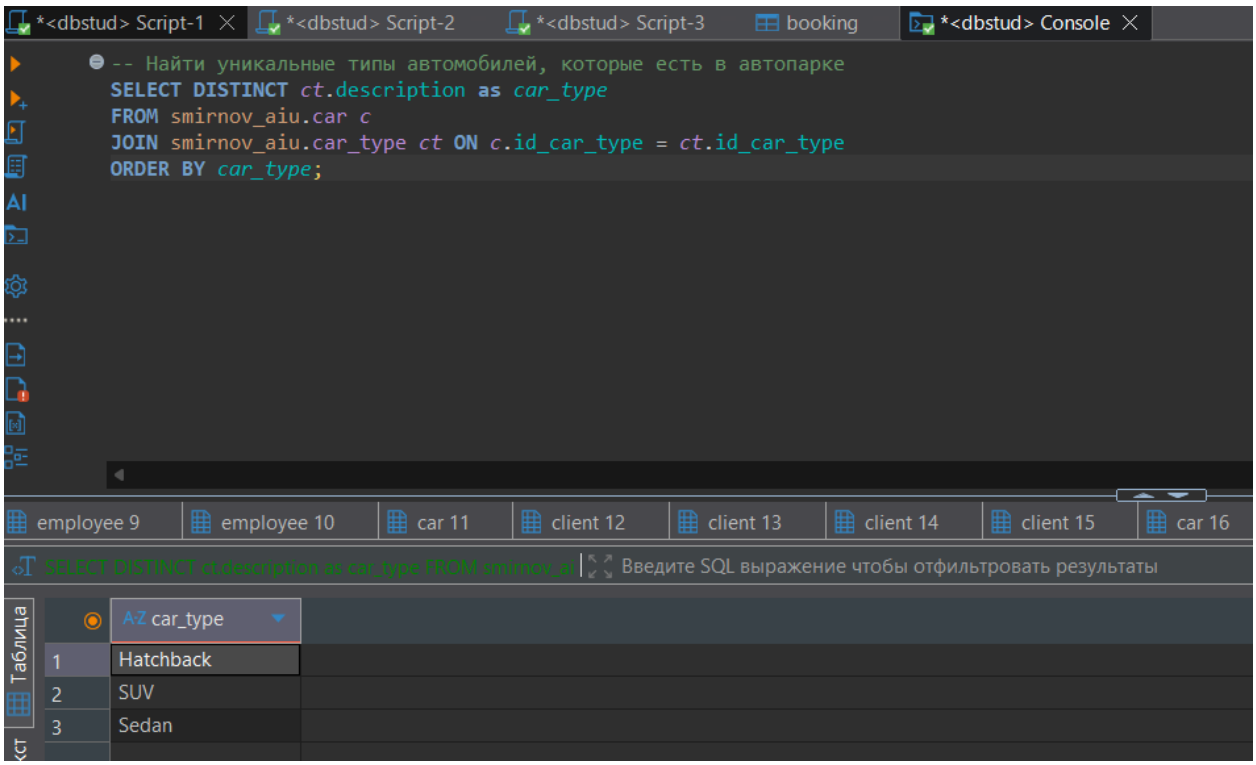
.



Сортировать автомобили по году выпуска



Найти статусы бронирований, где минимальная сумма заказа меньше 5000(HAVING)



Показывает сколько бронирований в каждом статусе, исключая статусы без бронирований.(Distinct)

Вывод: научился создавать ограничения в таблицах и использовать SQL-запросы для фильтрации данных