Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет: «Информационных технологий и прикладной математики»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Базы данных»

**Курсовой проект**

**Тема: «Каталог автомобилей»**

|  |  |
| --- | --- |
| Cтуденты: | Почечура А.А.  Речинская А.Ю. |
| Группа: | М8О-306Б-20 |
| Преподаватель: | Сеченых П.А. |
| Дата: |  |
| Оценка: |  |

Москва 2022

Оглавление

[**1.** **Таблицы и схема данных** 3](#_Toc120745579)

[**2.** **Схема в ErWin** 13](#_Toc120745580)

[**3.** **Запросы** 14](#_Toc120745581)

[**4.** **Разработка процедур на PL/SQL** 21](#_Toc120745582)

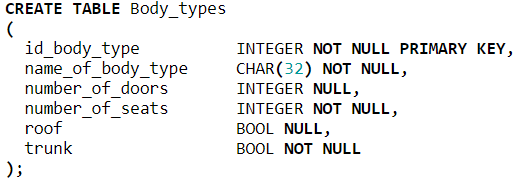
[**5.** **Список использованных источников** 26](#_Toc120745583)

# **Таблицы и схема данных**

В данной работе БД состоит из шести таблиц: Body\_types, Models, Salons\_of\_cars, Owners, List, Engines и Install. Таблицы Models и Salons\_of\_cars связаны друг с другом связью “многие ко многим” с помощью таблицы List, таблицы Models и Engines – связью “многие ко многим” с помощью таблицы Install, таблицы Models и Body\_types – связью “один к одному”, таблицы Owners и List – связью “один ко многим”. Опишем каждую из таблиц.

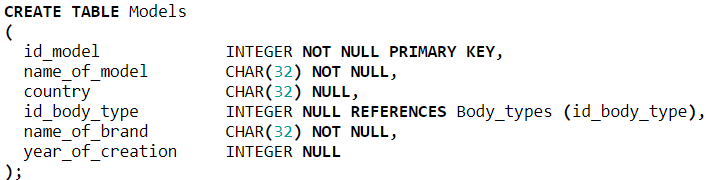
1. **Таблица Body\_types**

Содержит 6 полей, одно из которых является первичным ключом (id\_body\_type: в ней нумеруются типы кузовов автомобилей). Поле name\_of\_body\_type содержит название типа кузова, number\_of\_doors – количество дверей у данного типа, number\_of\_seats – количество мест, roof – наличие крыши, trunk – наличие багажника. Поля id\_body\_type, number\_of\_doors и number\_of\_seats имеют тип INTEGER, поле name\_of\_body\_type - тип CHAR (32), поля roof и trunk – тип BOOL. Нулевыми МОГУТ быть поля number\_of\_doors и roof.

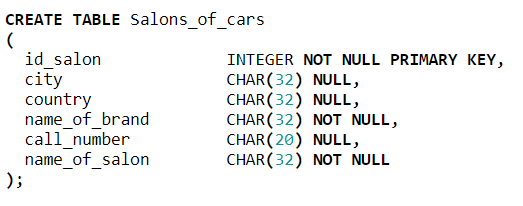


1. **Таблица Models**

Содержит 6 полей, одно из которых является первичным ключом (id\_model: в ней нумеруются модели), и ещё одно – ключом связи (id\_body\_type из таблицы Body\_types). Поле name\_of\_model содержит название модели, country – страну производителя, name\_of\_brand – марку модели, year\_of\_production – год создания модели. Все поля, кроме id\_model, id\_body\_type и year\_of\_production имеют тип CHAR (32) (id\_model, id\_body\_type и year\_of\_production имеют тип INTEGER). Нулевыми НЕ могут быть поля id\_model, name\_of\_model и name\_of\_brand.

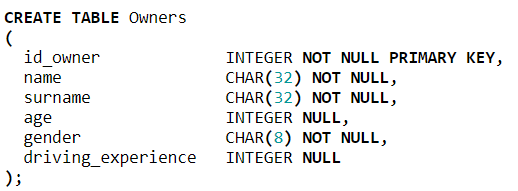


1. **Таблица Salons\_of\_cars**

Содержит 6 полей, одно из которых является первичным ключом (id\_salon: в ней нумеруются салоны). Поле city содержит город, в котором находятся салон, country – страну, в котором находится салон, name\_of\_brand – бренд салона, call\_number – телефонный номер салона, name\_of\_salon – название салона. Все поля, кроме id\_salon и call\_number имеют тип CHAR (32) (id\_salon имеет тип INTEGER, call\_number - тип CHAR (20)). Нулевыми НЕ могут быть поля id\_salon, name\_of\_brand и name\_of\_salon.  
  


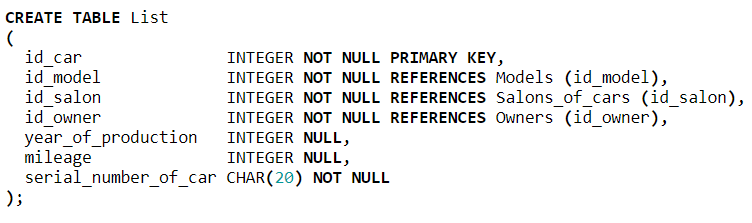
1. **Таблица Owners**

Содержит 6 полей, одно из которых является первичным ключом (id\_owner: в ней нумеруются владельцы автомобилей). Поле name содержит имя владельца, surname – фамилию, age – возраст, gender – пол, driving\_experience – опыт вождения. Все поля, кроме name, surname и gender имеют тип INTEGER (name и surname имеют тип CHAR (20), gender – тип CHAR(8)). Нулевыми МОГУТ быть поля age и driving\_experience.



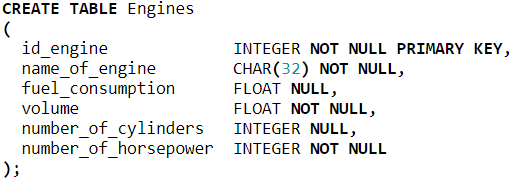
1. **Таблица List**

Содержит 7 полей, одно из которых является первичным ключом (id\_car: в ней нумеруются конкретные машины), а три – ключами связи (id\_model из таблицы Models, id\_salon из таблицы Salons\_of\_cars и id\_owners из таблицы Owners). Поле year\_of\_production содержит год производства машины, mileage – пробег машины, serial\_number\_of\_car – серийный номер автомобиля. Все поля, кроме serial\_number\_of\_car имеют тип INTEGER (serial\_number\_of\_car имеет тип CHAR (20)). Нулевыми МОГУТ быть поля year\_of\_production и mileage.



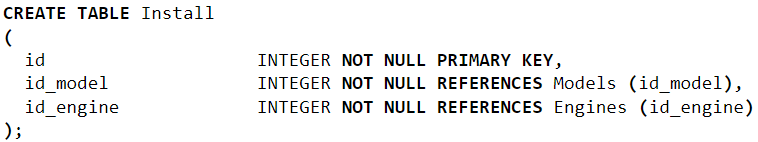
1. **Таблица Engines**

Содержит 6 полей, одно из которых является первичным ключом (id\_engine: в ней нумеруются двигатели). Поле name\_of\_engine содержит название двигателя, fuel\_consumption – средний расход топлива в литрах на 100 км, volume – объём топлива, number\_of\_cylinders – количество цилиндров, number\_of\_horsepower – количество лошадиных сил. Все поля, кроме name\_of\_engine, fuel\_consumption и volume имеют тип INTEGER (name\_of\_engine и fuel\_consumption имеют тип FLOAT, name\_of\_engine – тип CHAR(32)). Нулевыми МОГУТ быть поля fuel\_consumption и number\_of\_horsepower.



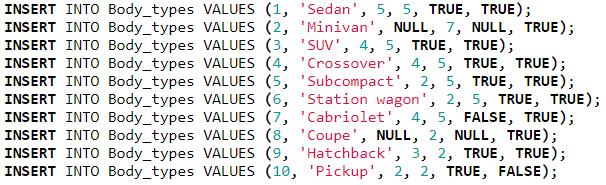
1. **Таблица Install**

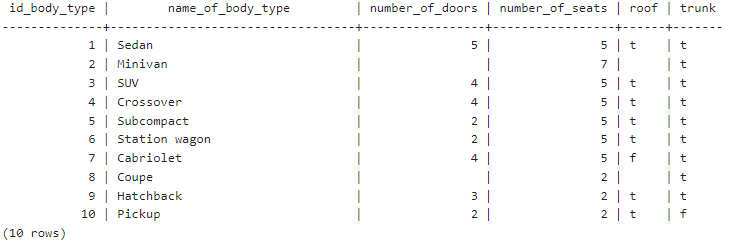
Содержит 3 поля, одно из которых является первичным ключом (id: в ней нумеруются соответствия “модель – двигатель”), а два – ключами связи (id\_model из таблицы Models и id\_engine из таблицы Engines). Все поля имеют тип INTEGER. Нулевым не может быть ни одно поле.



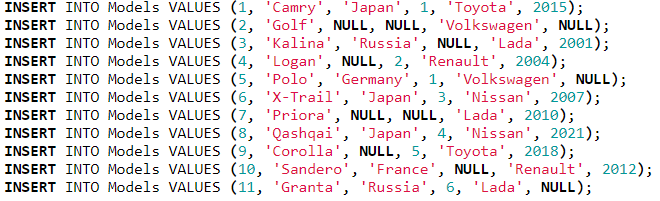
**Разработанная схема данных:**

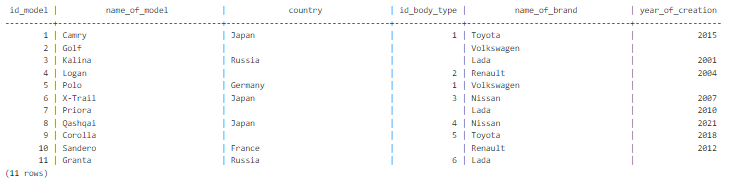
1. Таблица Body\_types





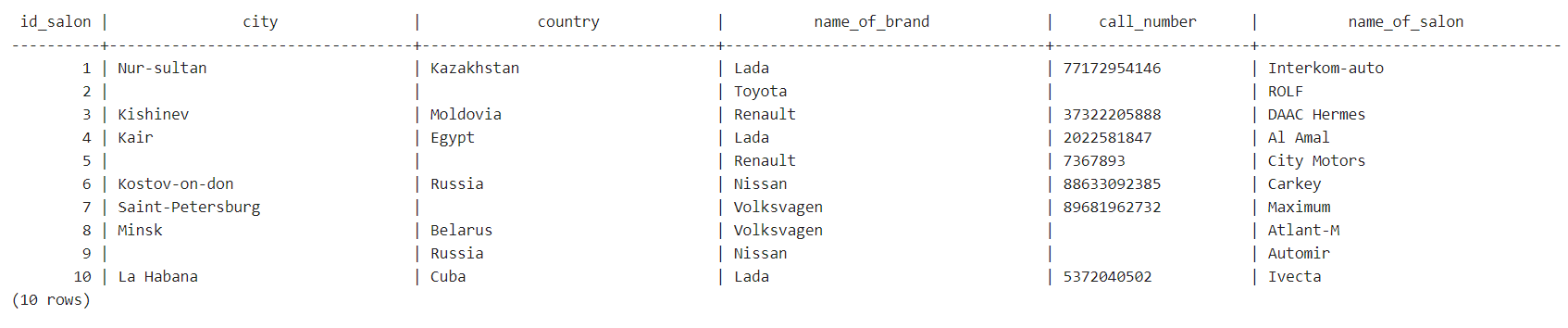
1. Таблица Models



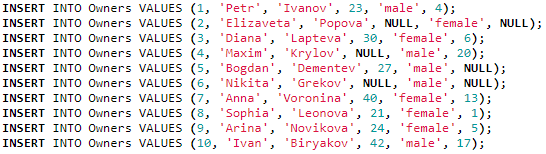


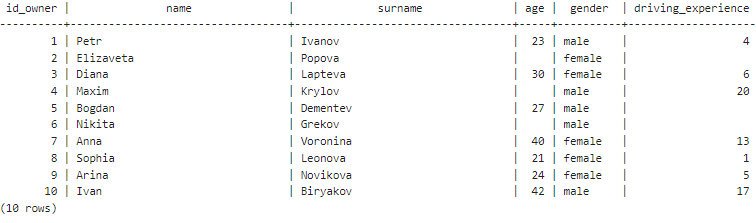
1. Таблица Salons\_of\_cars



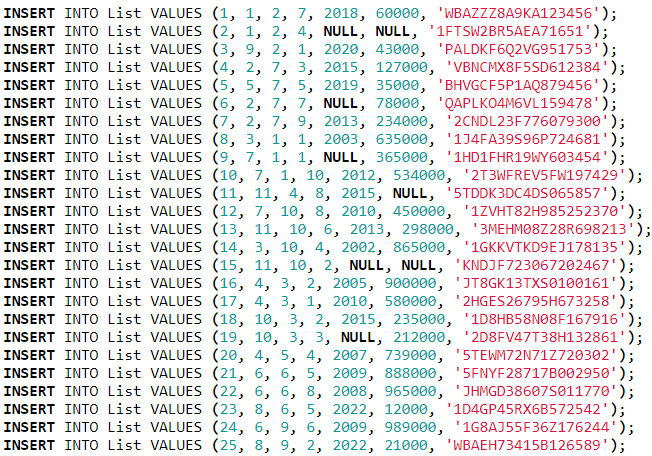


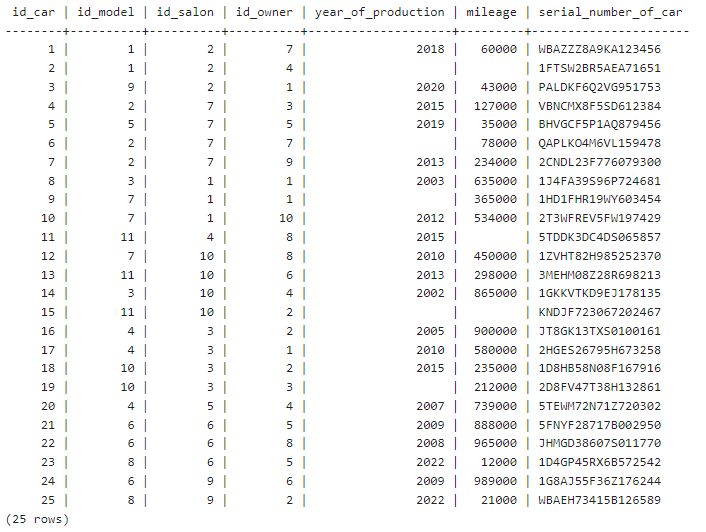
1. Таблица Owners



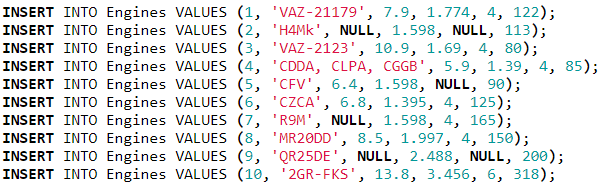


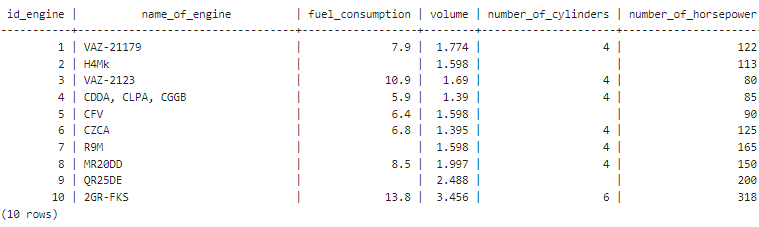
1. Таблица List



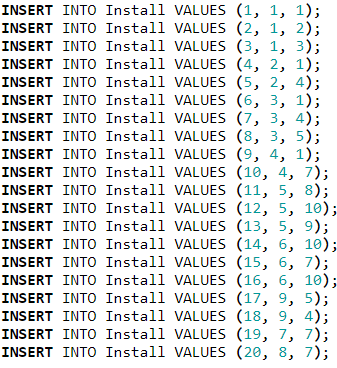


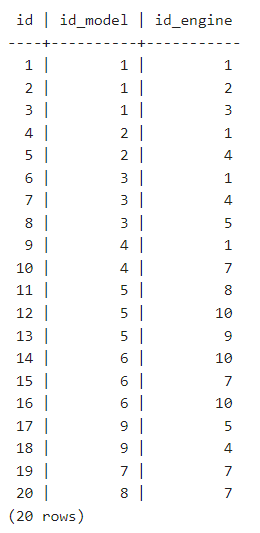
1. Таблица Engines





1. Таблица Install





**Проверка правильности ввода (на вводе с ошибками):**

1. Запрос:  
   

Вывод:



1. Запрос:  
   

Вывод: 

1. Запрос:  
   

Вывод:



1. Запрос:  
   

Вывод:



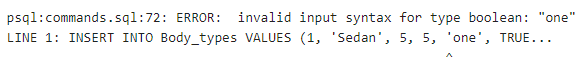
1. Запрос:  
   

Вывод:



1. Запрос:  
   

Вывод:



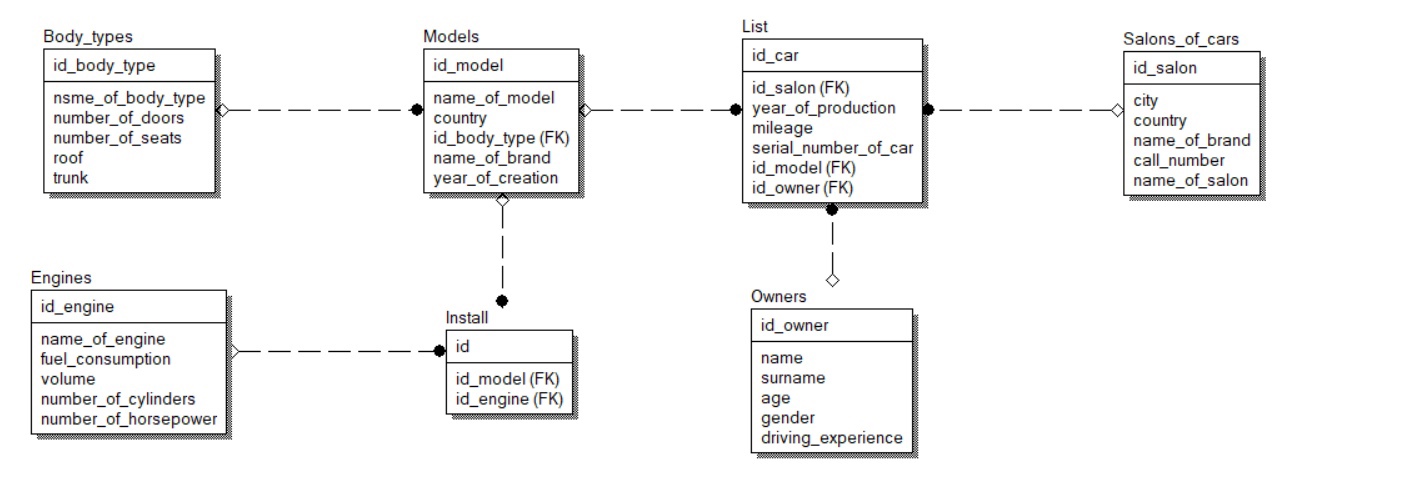
1. Запрос:  
   

Вывод:

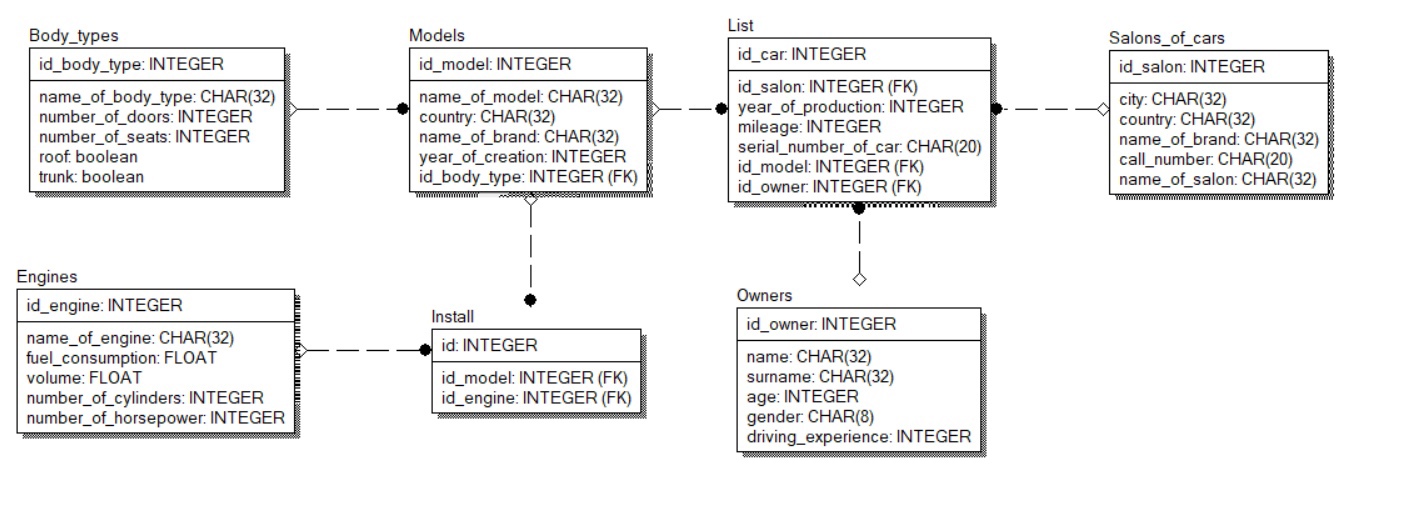


# **Схема в ErWin**

Логическая схема:



Физическая схема:



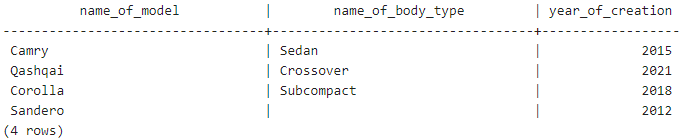
# **Запросы**

1. **Использование операторов JOIN (LEFT, RIGHT, CROSS, FULL)**
2. Вывести все модели и их тип кузова (при наличии), которые были выпущены позже 2012 года включительно.

Запрос:



Вывод:

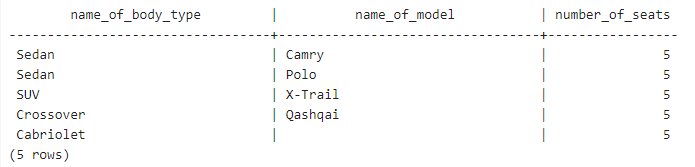


1. Вывести все типы кузовов и модели, соответствующие им из базы данных (при наличии), у которых больше трёх дверей.

Запрос:



Вывод:

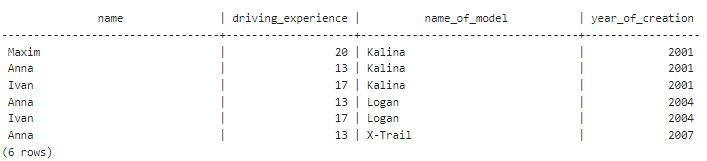


1. Вывести, на каких машинах из базы данных могли ездить владельцы со стажем больше 10 лет, как только они получили права.

Запрос:



Вывод:

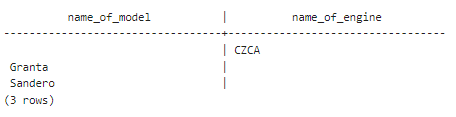


1. Вывести автомобили, на которые мы не сможем поставить двигатели из базы данных, и двигатели, которые мы не сможем поставить ни на одну модель из базы.

Запрос:

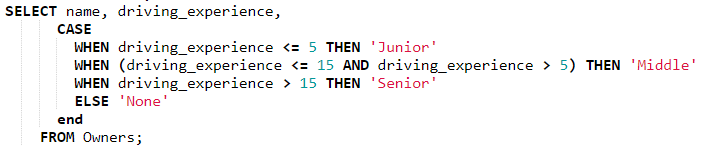


Вывод:

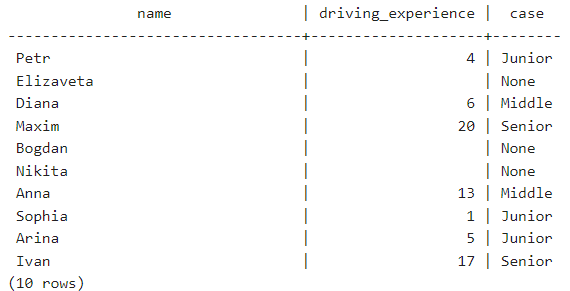


1. **Использование выражения CASE**
2. Вывести для каждого владельца его условный статус, зависящий от его водительского опыта (в случае отсутствия информации об опыте, вывести “None”).

Запрос:

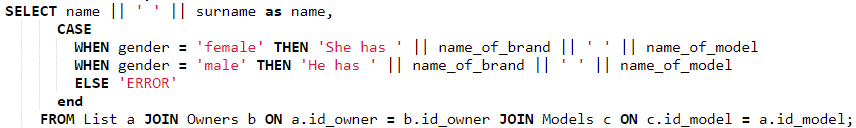


Вывод:

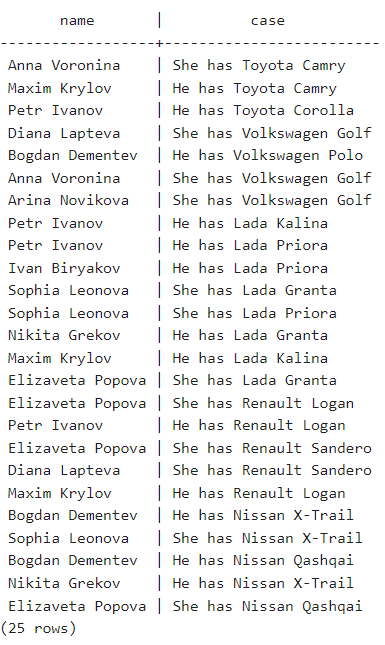
****

1. Вывести имена владельцев и те автомобили, которыми они владеют (склонить предложение, учитывая пол).

Запрос:

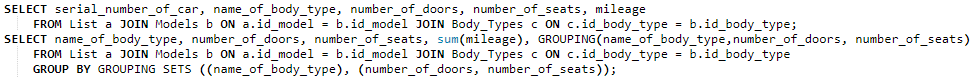


Вывод:

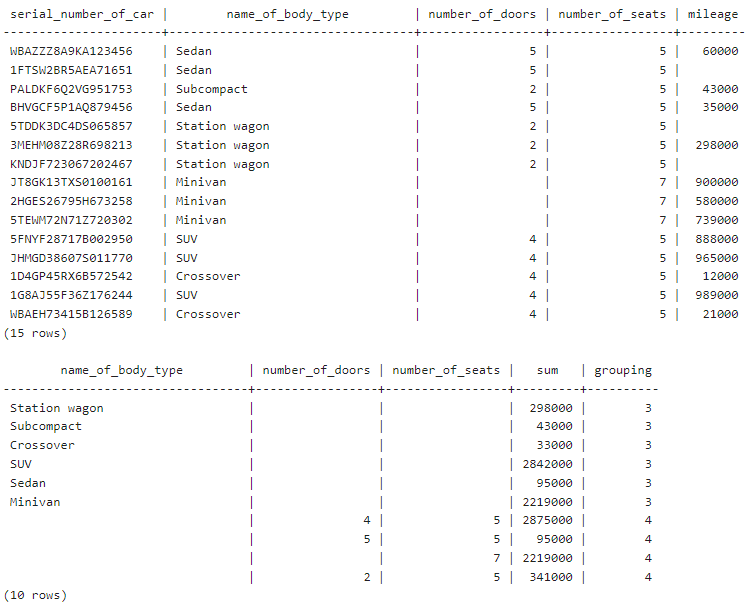
****

1. **Использование оператора GROUP BY с расширениями (ROLLUP, CUBE, GROUPING).**
2. Вывести суммы пробегов всех машин одного типа и всех машин, у количество дверей и сидений которых совпадает (столбец grouping содержит число, битовая маска которого означает следующее: если на позиции i стоит единица, то колонка с номером i из таблицы не участвует в группировке).

Запрос:

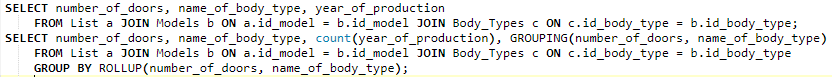


Вывод:

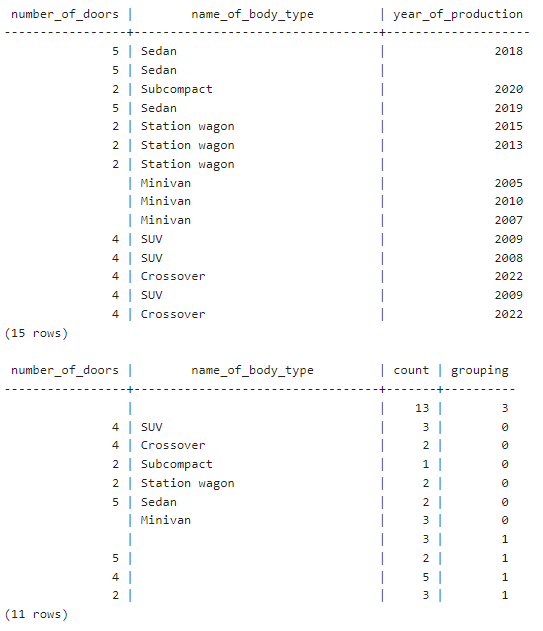
****

1. Вывести количество различных годов выпуска среди всех машин из базы данных, затем количество различных годов выпуска среди машин, сгруппированных по количеству дверей и типу кузова, а дальше среди машин, сгруппированных только по количеству дверей (учитывание “столбцов влияния” – в колонке grouping). (ROLLUP создаёт группы по префиксам из переданных ему столбцов).

Запрос:

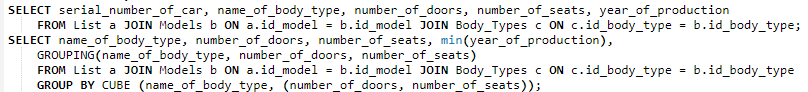


Вывод:

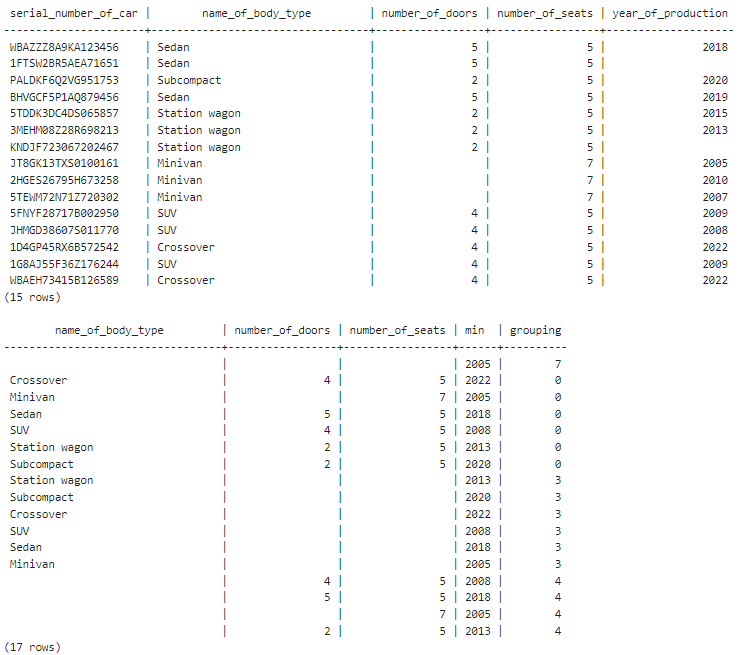


1. Вывести минимальный год производства среди машин, у которых указан в базе данных тип кузова, затем минимальный год производства среди машин, сгруппированных по типу кузова, количеству мест и дверей, и далее вывести минимальный год производства для машин, сгруппированных по количеству мест и дверей. (CUBE группирует по всем возможным комбинациям из переданных ему столбцов).

Запрос:



Вывод:

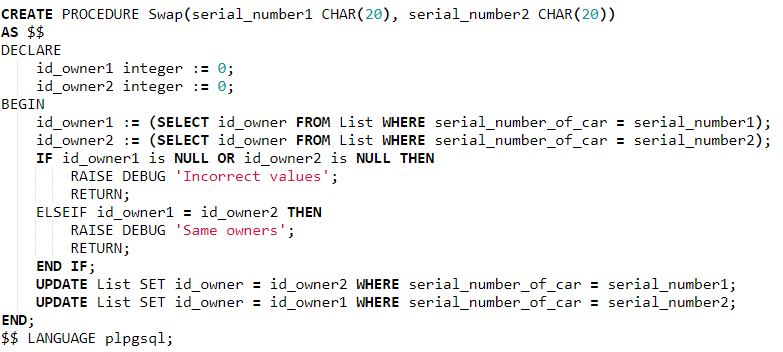


# **Разработка процедур на PL/SQL**

1. **Процедура**

Процедуре на вход подаётся два серийных номера машин. Процедура находит владельцев каждой из двух машин и производит обмен: владелец первой машины теперь (согласно базе данных) владеет второй машиной, а второй владелец – соответственно владеет первой. Если хотя бы один из указанных серийных номеров не был найден, то в раздел Messages выводится строка 'Incorrect values'. Если серийные номера одинаковые или владелец у двух машин один, то выводится строка 'Same owners'.

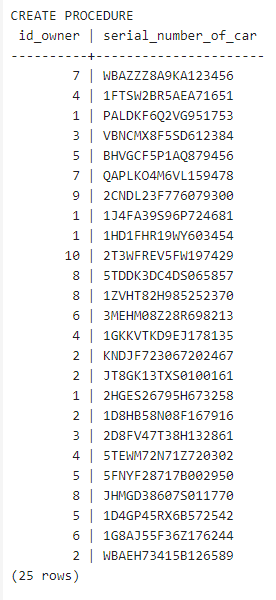
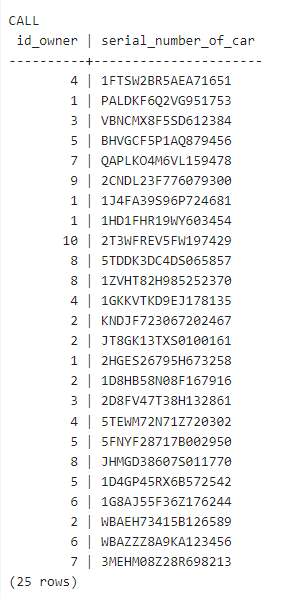
Процедура:



Запрос:



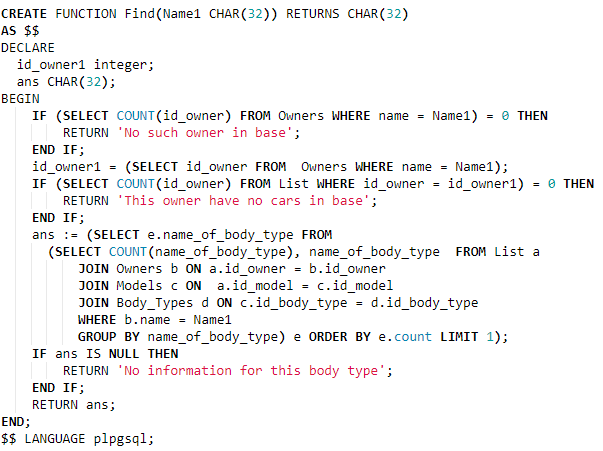
Вывод (владельцы указанных серийных номеров имеют id 7 и 6):

1. **Фунция**

Функция принимает на вход имя владельца и выводит самый частый тип кузова среди машин, которыми он владеет (согласно базе данных). Если указанное имя не найдено в базе, то функция возвращает строку “ No such owner in base”. Если нет информации о том, какие машины имеются у данного владельца, но в базе он нашёлся, то выводится строка “ This owner have no cars in base”. Строка “No information for this body type” выводится в случае, когда тип кузова определить не удаётся (недостаточно информации).

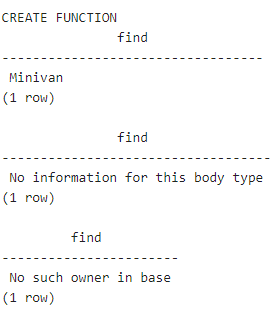
Функция:



Запросы:



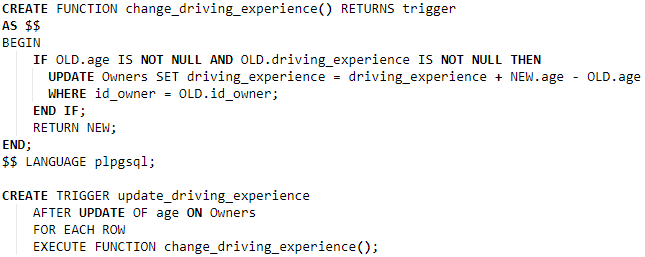
Вывод:



1. **Триггер**

Триггер срабатывает в момент обновления возрастов владельцев из таблицы Owners. После изменения возраста происходит обновление водительского опыта для каждого владельца ровно на ту величину, на которую был изменён его возраст. Обновление происходит только в тех строках, в которых имеется информация и о водительском опыте, и о возрасте владельца.

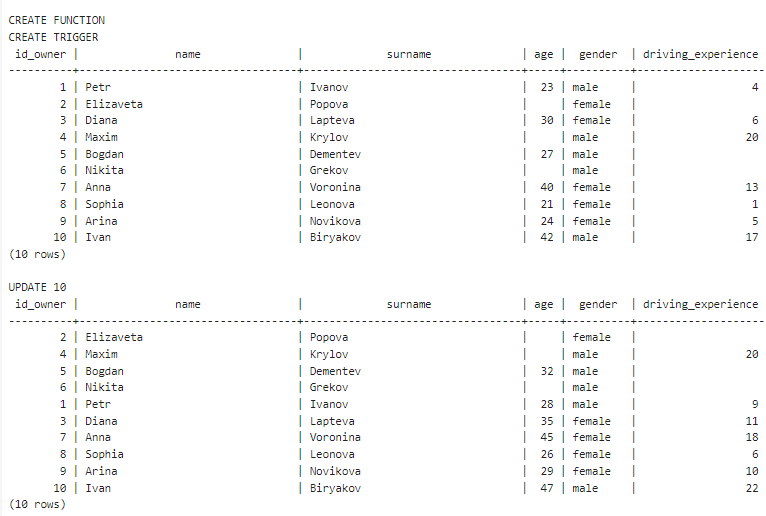
Триггер:



Запрос:



Вывод:



# **Список использованных источников**

1. <https://postgrespro.ru/docs/postgresql/> - PostgreSQL : Документация.

2. <https://timeweb.cloud/blog/postgresql-triggery-sozdanie-udalenie-primery?ysclid=lb43v0tmol735117323> - PostgreSQL-триггеры: создание, удаление, примеры.