

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение  
высшего образования «Санкт-Петербургский политехнический  
университет Петра Великого»

институт компьютерных наук и кибербезопасности

высшая школа технологий искусственного интеллекта

Направление: 02.03.01 Математика и компьютерные науки

Теоретические основы баз данных  
*Сеть частных автозаправочных станций*

Студент,

группы 5130201/20001

\_\_\_\_\_ Якунин Д. Д.

Преподаватель,

к.т.н, доц., доц.

\_\_\_\_\_ Попов С. Г.

«\_\_\_\_\_» \_\_\_\_\_ 2024г.

Санкт-Петербург, 2024

# Содержание

<b>1</b>	<b>Аналитика</b>	<b>3</b>
1.1	Описание предметной области . . . . .	3
1.2	Функционирование предметной области . . . . .	3
1.3	ER-диаграмма . . . . .	4
<b>2</b>	<b>Проектированные базы данных</b>	<b>6</b>
2.1	Цели функционирования системы . . . . .	6
2.2	Описание ролей и субъектов . . . . .	6
2.3	Описание сущностей . . . . .	6
2.4	Схема объектов . . . . .	7
2.5	Схема базы данных . . . . .	7
2.6	Атрибуты таблиц базы данных . . . . .	10
2.7	Заполнение базы данных . . . . .	13
<b>3</b>	<b>Составление запросов</b>	<b>14</b>
3.1	Запрос №1 . . . . .	14
3.1.1	Запрос 1.1 . . . . .	14
3.1.2	Запрос 1.2 . . . . .	15
3.2	Запрос №2 . . . . .	17
3.3	Запрос №3 . . . . .	19
3.3.1	Запрос 3.1 . . . . .	19
3.3.2	Запрос 3.2 . . . . .	21
3.4	Запрос №4 . . . . .	23
3.4.1	Запрос 4.1 . . . . .	23
3.4.2	Запрос 4.2 . . . . .	27
3.5	Запрос №5 . . . . .	30
3.6	Запрос №6 . . . . .	33
3.7	Запрос №7 . . . . .	37
3.8	Запрос №8 . . . . .	40
	<b>Заключение</b>	<b>48</b>
	<b>Приложение А. Код создания базы данных</b>	<b>49</b>
	<b>Приложение В. Скрипт заполнения базы данных</b>	<b>54</b>

# 1 Аналитика

## 1.1 Описание предметной области

Рассматривается сеть частных автозаправочных станций (АЗС) коммерческой компании, предоставляющая комплекс услуг для водителей. АЗС оборудованы всем необходимым для оперативной заправки различными видами топлива: бензином, дизелем и сжиженным газом. Для обеспечения бесперебойной работы станций компания поддерживает стратегические запасы топлива и регулярно обновляет их через постоянные поставки, гарантируя непрерывность доступа к топливу для клиентов.

АЗС оборудованы современными системами учёта топлива, позволяющими мониторить уровни запасов в реальном времени и автоматически инициировать заказ топлива при достижении определённого критического уровня. Для доставки топлива в компании имеется собственный автопарк бензовозов.

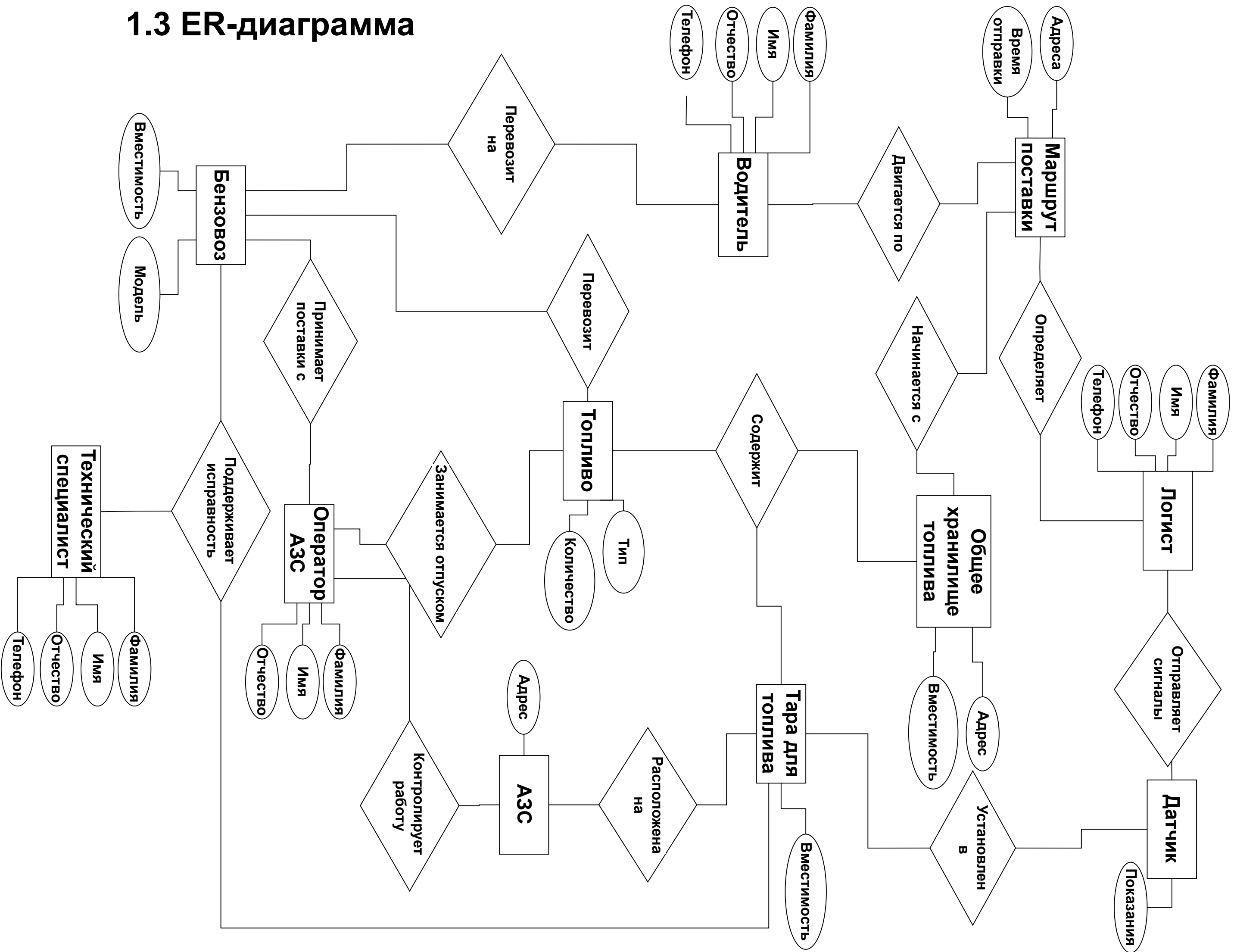
Автозаправочные комплексы функционируют круглосуточно, а их расположение стратегически продумано для обеспечения максимальной доступности для водителей.

## 1.2 Функционирование предметной области

На АЗС топливо хранится в тарах, оснащенных датчиками, автоматически отслеживающих уровень топлива и при достижении низкого уровня отправляют сигнал о том, что необходимо пополнить запасы. Эти сигналы и уровни топлива на всех АЗС отслеживают логисты и оптимизируют маршруты поставок.

По выведенным маршрутам и расписанию поставок топливо на АЗС доставляют водители с помощью бензовозов. На самих станциях операторы АЗС принимают поставки и осуществляют непосредственную заправку топливом, соблюдая при этом технику безопасности и контролируя правильную работу оборудования.

В компании регулярно проводятся проверки технического состояния всех частей системы: оборудования на АЗС, хранилищ для топлива, транспортных средств для доставки топлива. В случае каких-либо технических неисправностей на АЗС приезжает технический специалист и проводит ремонт пришедшего в негодность оборудования.



### 1.3 ER-диаграмма

## Чтение ER-диаграммы

1. Датчик установлен в тару для топлива.
2. Датчик отправляет сигналы логисту.
3. Логист определяет маршрут поставки.
4. Маршрут поставки начинается с общего хранилища топлива.
5. Водитель двигается по маршруту поставки.
6. Водитель перевозит на бензовозе.
7. Бензовоз перевозит топливо.
8. Технический специалист поддерживает исправность бензовозов и тар для топлива.
9. Оператор АЗС принимает поставки с бензовоза.
10. Оператор АЗС занимается отпуском топлива.
11. Оператор АЗС контролирует работу АЗС.
12. Общее хранилище топлива и тары для топлива содержат топливо.
13. Тары для топлива расположены на АЗС.

## 2 Проектированные базы данных

### 2.1 Цели функционирования системы

1. Мониторинг остатков топлива: отслеживание остатков топлива в резервуарах АЗС для своевременных заказов и пополнения запасов, а также для предотвращения простоев из-за отсутствия топлива.
2. Техническое обслуживание оборудования: ведение журнала регулярных проверок и технического обслуживания оборудования АЗС для предотвращения неисправностей и сбоев в работе.
3. Отслеживание истории продаж топлива.
4. Отчет по фактическим датам доставки топлива.

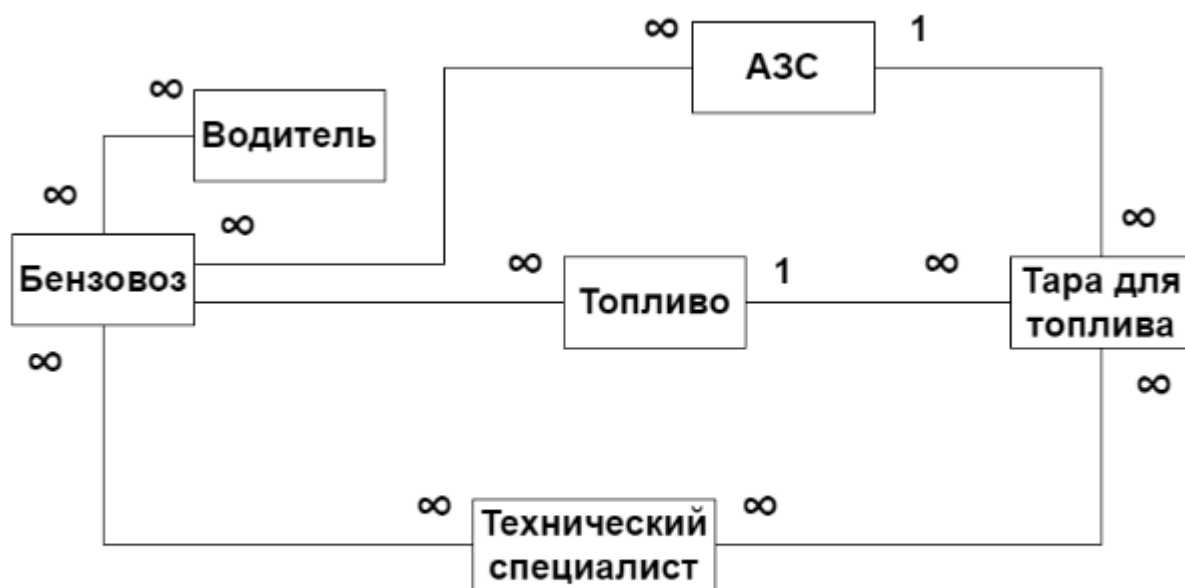
### 2.2 Описание ролей и субъектов

1. Операторы АЗС: находятся непосредственно на АЗС, осуществляют заправку топливом, контролируют правильное функционирование оборудования и соблюдение правил безопасности.
2. Водители бензовозов: люди, ответственные за доставку топлива на АЗС из общей нефтебазы.
3. Технические специалисты: сотрудники, занимающиеся обслуживанием и ремонтом оборудования.
4. Логисты: специалисты, отвечающие за планирование маршрутов доставки топлива, оптимизацию расписания поставок и управление запасами топлива.

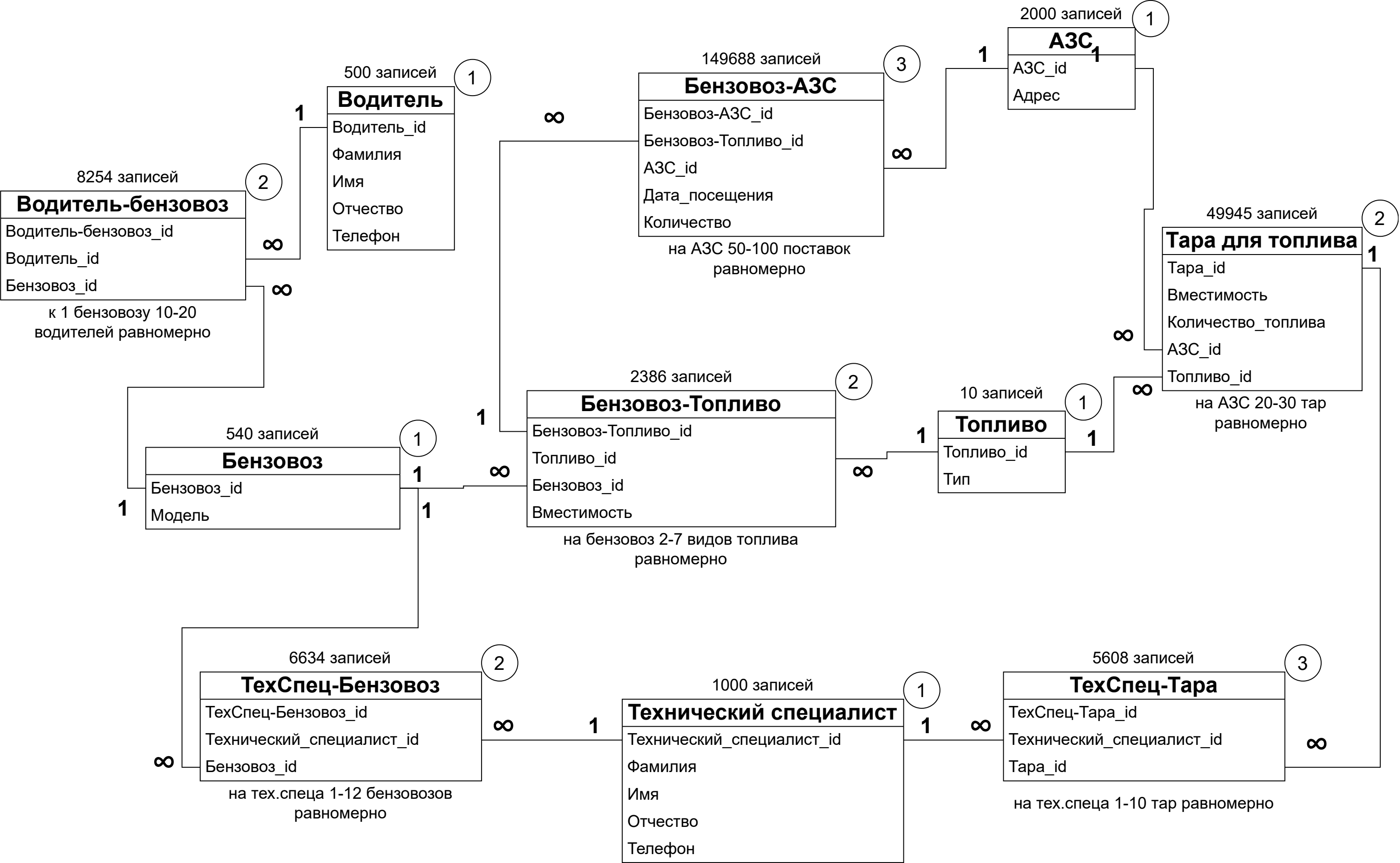
### 2.3 Описание сущностей

1. Датчики уровня топлива: показания.
2. Тары для топлива: вместимость.
3. Бензовозы: модель, вместимость.
4. Топливо: тип.
5. АЗС: адрес.
6. Общее хранилище топлива: адрес, вместимость.
7. Маршрут поставки: адреса маршрута, дата и время отправления.

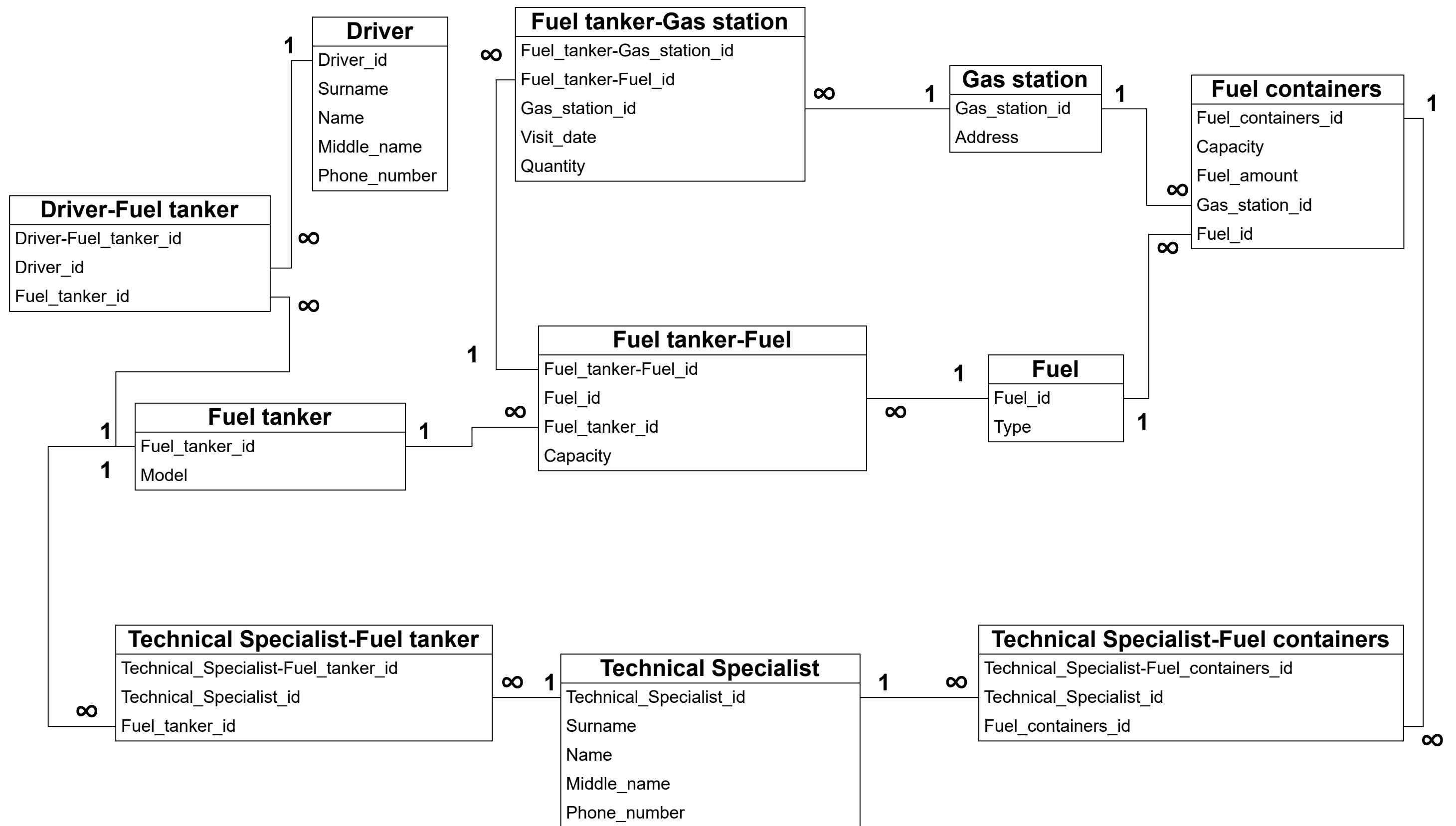
## 2.4 Схема объектов



# 2.5 Схема базы данных







## 2.6 Атрибуты таблиц базы данных

В Таблице 1 - Таблице 11 представлены атрибуты каждой таблицы базы данных. Перечислены тип переменной, размер переменной, ограничения атрибута и тип ключа, если поле является ключом, а также на какую колонку ссылается данная колонка, если она является внешним ключом.

PK - первичный ключ, FK - внешний ключ.

Водитель - Driver					
Key	Название EN	Название RU	Тип атрибута	NULL/NOT NULL	Ссылка
PK	Driver_id	Водитель_id	INT	NOT NULL	-
-	Surname	Фамилия	VARCHAR [50]*	NOT NULL	-
-	Name	Имя	VARCHAR [50]*	NOT NULL	-
-	Middle_name	Отчество	VARCHAR [50]*	NOT NULL	-
-	Phone_number	Телефон	VARCHAR [20]**	NOT NULL	-

Таблица 1. Водитель - Driver

\* Ограничение количества символов для фамилии, имени и отчества равно 50, потому что максимальная длина имени и фамилии пользователя на Facebook в настоящее время составляет 50 символов.

\*\* Ограничение количества символов для номера телефона равно 20, так как самый длинный номер телефона имеет длину 17 цифр, к ним добавляем символы '(', ')' и '+'.

Бензовоз-Fuel tanker					
Key	Название EN	Название RU	Тип атрибута	NULL/NOT NULL	Ссылка
PK	Fuel_tanker_id	Бензовоз_id	INT	NOT NULL	-
-	Model	Модель	VARCHAR [45]	NOT NULL	-

Таблица 2. Бензовоз-Fuel tanker

Топливо - Fuel					
Key	Название EN	Название RU	Тип атрибута	NULL/NOT NULL	Ссылка
PK	Fuel_id	Топливо_id	INT	NOT NULL	-
-	Type	Тип	VARCHAR [20]	NOT NULL	-

Таблица 3. Топливо - Fuel

АЗС - Gas station					
Key	Название EN	Название RU	Тип атрибута	NULL/NOT NULL	Ссылка
PK	Gas_station_id	АЗС_id	INT	NOT NULL	-
-	Address	Адрес	VARCHAR [200]*	NOT NULL	-

Таблица 4. АЗС - Gas station

\* Ограничение количества символов для адреса равно 200, потому что максимальная длина улицы в России составляет 138 символов. К названию улицы же ещё нужно добавить знаки препинания и номер дома.

Тара для топлива - Fuel container					
Key	Название EN	Название RU	Тип атрибута	NULL/NOT NULL	Ссылка
PK	Fuel_containers_id	Тара_id	INT	NOT NULL	-
-	Capacity	Вместимость	DECIMAL(7, 2)*	NOT NULL	-
-	Fuel_amount	Количество_топлива	DECIMAL(7, 2)*	NOT NULL	-
FK	Gas_station_id	АЗС_id	INT	NOT NULL	Gas station (Gas_station_id)
FK	Fuel_id	Топливо_id	INT	NOT NULL	Fuel (Fuel_id)

Таблица 5. Тара для топлива - Fuel containers

\* Ограничение длины в decimal(7,2) связано с тем, что бензовозов, перевозящих 1 000 000 и больше литров не существует, а точности в 2 знака после запятой достаточно, так как дальнейшие числа уже на уровне погрешности.

Технический специалист - Technical Specialist					
Key	Название EN	Название RU	Тип атрибута	NULL/NOT NULL	Ссылка
PK	Technical_Specialist_id	Технический_специалист_id	INT	NOT NULL	-
-	Surname	Фамилия	VARCHAR [50]*	NOT NULL	-
-	Name	Имя	VARCHAR [50]*	NOT NULL	-
-	Middle_name	Отчество	VARCHAR [50]*	NOT NULL	-
-	Phone_number	Телефон	VARCHAR [20]*	NOT NULL	-

Таблица 6. Технический специалист - Technical Specialist

\* Причину таких ограничений длин смотри под таблицей 1.

Водитель-бензовоз - Driver-Fuel tanker					
Key	Название EN	Название RU	Тип атрибута	NULL/NOT NULL	Ссылка
PK	Driver-Fuel_tanker_id	Водитель-бензовоз_id	INT	NOT NULL	-
FK	Driver_id	Водитель_id	INT	NOT NULL	Driver (Driver_id)
FK	Fuel_tanker_id	Бензовоз_id	INT	NOT NULL	Fuel tanker (Fuel_tanker_id)

Таблица 7. Водитель-бензовоз - Driver-Fuel tanker

Бензовоз-Топливо - Fuel tanker-Fuel					
Key	Название EN	Название RU	Тип атрибута	NULL/NOT NULL	Ссылка
PK	Fuel_tanker-Fuel_id	Бензовоз-Топливо_id	INT	NOT NULL	-
FK	Fuel_id	Топливо_id	INT	NOT NULL	Fuel (Fuel_id)
FK	Fuel_tanker_id	Бензовоз_id	INT	NOT NULL	Fuel tanker (Fuel_tanker_id)
-	Capacity	Вместимость	DECIMAL(7, 2)*	NOT NULL	-

Таблица 8. Бензовоз-Топливо - Fuel tanker-Fuel

\* Ограничение в DECIMAL(7, 2) связано с тем, что по ГОСТу на АЗС недопустимы тары больше, чем в 60 000 литров. Точности в 2 знака после запятой достаточно, так как дальнейшие числа уже на уровне погрешности.

Бензовоз-АЗС - Fuel tanker-Gas station					
Key	Название EN	Название RU	Тип атрибута	NULL/NOT NULL	Ссылка
PK	Fuel_tanker-Gas_station_id	Бензовоз-АЗС_id	INT	NOT NULL	-
FK	Fuel_tanker_id	Бензовоз_id	INT	NOT NULL	Fuel tanker (Fuel_tanker_id)
FK	Gas_station_id	АЗС_id	INT	NOT NULL	Gas station (Gas_station_id)
-	Visit_date	Дата посещения	DATE	NOT NULL	-
FK	Fuel_id	Топливо_id	INT	NOT NULL	Fuel (Fuel_id)
-	Quantity	Количество	DECIMAL(7, 2)*	NOT NULL	-

Таблица 9. Бензовоз-АЗС - Fuel tanker-Gas station

\* Выбран DECIMAL(7, 2), так как бензовоз не может привезти на АЗС топлива больше, чем в нем есть, а у бензовоза ограничение вместимости в DECIMAL(7, 2).

ТехСпец-Бензовоз - Technical Specialist-Fuel tanker					
Key	Название EN	Название RU	Тип атрибута	NULL/NOT NULL	Ссылка
PK	Technical_Specialist-Fuel_tanker_id	ТехСпец-Бензовоз_id	INT	NOT NULL	-
FK	Technical_Specialist_id	Технический_специалист_id	INT	NOT NULL	Technical Specialist (Technical_Specialist_id)
FK	Fuel_tanker_id	Бензовоз_id	INT	NOT NULL	Fuel tanker (Fuel_tanker_id)

Таблица 10. ТехСпец-Бензовоз - Technical Specialist-Fuel tanker

ТехСпец-Тара - Technical Specialist-Fuel containers					
Key	Название EN	Название RU	Тип атрибута	NULL/NOT NULL	Ссылка
PK	Technical_Specialist-Fuel_containers_id	ТехСпец-Тара_id	INT	NOT NULL	-
FK	Technical_Specialist_id	Технический_специалист_id	INT	NOT NULL	Technical Specialist (Technical_Specialist_id)
FK	Fuel_containers_id	Тара_id	INT	NOT NULL	Fuel containers (Fuel_containers_id)

Таблица 11. ТехСпец-Тара - Technical Specialist-Fuel containers

Код создания базы данных по приведенной схеме можно посмотреть в [Приложении А: Код создания базы данных](#).

## 2.7 Заполнение базы данных

Для генерации записей был использован язык Python, API MySQL Connector и библиотека Faker, генерирующая, например, номера телефонов и ФИО.

При генерации используется команда REPLACE при заполнении таблиц первого уровня, чтобы количество записей увеличивалось после регенерации и не дублировалось. Таблицы же уровней выше перед заполнением очищаются от всех данных.

Всего было сгенерировано 226565 записей. Код заполнения можно посмотреть в [Приложении Б: Скрипт заполнения базы данных](#).

## 3 Составление запросов

### 3.1 Запрос №1

#### 3.1.1 Запрос 1.1

Найти все АЗС, на которые привозил топливо А (type="АИ-92") водитель Б (id=3) и который обслуживал технический специалист С (id=3).

#### Код запроса

```
1 select
2     `gas station`.Address,
3     fuel.`Type` as "Тип топлива",
4     driver.Surname as "Фамилия водителя",
5     driver.Name as "Имя водителя",
6     driver.Middle_name as "Отчество водителя",
7     `technical specialist`.Surname as "Фамилия специалиста",
8     `technical specialist`.Name as "Имя специалиста",
9     `technical specialist`.Middle_name as "Отчество специалиста"
10 from
11     `gas station`
12 inner join `fuel tanker-gas station`
13     on `fuel tanker-gas station`.Gas_station_id=`gas station`.Gas_station_id
14 inner join `fuel tanker-fuel`
15     on `fuel tanker-gas station`.`Fuel_tanker-Fuel_id`=`fuel
16     ⇨ tanker-fuel`.`Fuel_tanker-Fuel_id`
17 inner join fuel
18     on fuel.Fuel_id=`fuel tanker-fuel`.Fuel_id
19 inner join `fuel tanker`
20     on `fuel tanker-fuel`.Fuel_tanker_id=`fuel tanker`.Fuel_tanker_id
21 inner join `driver-fuel tanker`
22     on `driver-fuel tanker`.Fuel_tanker_id=`fuel tanker`.Fuel_tanker_id
23 inner join driver
24     on driver.driver_id=`driver-fuel tanker`.Driver_id
25 inner join `technical specialist-fuel tanker`
26     on `technical specialist-fuel tanker`.Fuel_tanker_id=`fuel
27     ⇨ tanker`.Fuel_tanker_id
28 inner join `technical specialist`
29     on `technical specialist-fuel tanker`.`Technical_Specialist_id`=`technical
30     ⇨ specialist`.`Technical_Specialist_id`
31 where fuel.`Type`="АИ-92" and driver.driver_id=1 and `technical
32     ⇨ specialist`.Technical_Specialist_id=18
33 order by `gas station`.Address
```

## Результат запроса

Запрос выполнен за 110 миллисекунд. В результирующей таблице получилось 52 строки.

	Address	Тип топлива	Фамилия водителя	Имя водителя	Отчество водителя	Фамилия специалиста	Имя специалиста	Отчество специалиста
▶	г. Гремячинск (Бурят.), пер. Енисейский, д. ...	АИ-92	Кондратьев	Аполлинарий	Вилорович	Воробьев	Ираклий	Гаврилович
	г. Моршанск, алл. Металлистов, д. 321 стр. ...	АИ-92	Кондратьев	Аполлинарий	Вилорович	Воробьев	Ираклий	Гаврилович
	г. Плещеек, ш. Мостовое, д. 4 к. 9, 427584	АИ-92	Кондратьев	Аполлинарий	Вилорович	Воробьев	Ираклий	Гаврилович
	г. Рыбинск, пр. Новгородский, д. 84 к. 7/9, ...	АИ-92	Кондратьев	Аполлинарий	Вилорович	Воробьев	Ираклий	Гаврилович
	г. Славянск-на-Кубани, пр. Загородный, д. ...	АИ-92	Кондратьев	Аполлинарий	Вилорович	Воробьев	Ираклий	Гаврилович
	г. Солнечногорск, ш. Спартака, д. 72, 124870	АИ-92	Кондратьев	Аполлинарий	Вилорович	Воробьев	Ираклий	Гаврилович

Рис. 1. Часть результата запроса 1.1

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	driver	NULL	const	PRIMARY	PRIMARY	4	const	1	100.00	Using temporary; Using filesort
	1	SIMPLE	technical specialist	NULL	const	PRIMARY	PRIMARY	4	const	1	100.00	NULL
	1	SIMPLE	technical specialist-f...	NULL	ref	Fuel_tanker_id2_idx,Technical_Specialist_id_idx	Technical_Specialist_id_idx	4	const	3	100.00	NULL
	1	SIMPLE	driver-fuel tanker	NULL	ref	driver_id1_idx,fuel_tanker_id1_idx	fuel_tanker_id1_idx	4	fueldb.technical specialist-fuel tanker.Fuel_tank...	15	0.33	Using where; Using index
	1	SIMPLE	fuel tanker	NULL	eq_ref	PRIMARY	PRIMARY	4	fueldb.technical specialist-fuel tanker.Fuel_tank...	1	100.00	Using index
	1	SIMPLE	fuel tanker-fuel	NULL	ref	PRIMARY,Fuel_tanker_id2_idx,Fuel_id_idx	Fuel_tanker_id2_idx	4	fueldb.technical specialist-fuel tanker.Fuel_tank...	4	100.00	NULL
	1	SIMPLE	fuel	NULL	eq_ref	PRIMARY	PRIMARY	4	fueldb.fuel tanker-fuel.Fuel_id	1	10.00	Using where
	1	SIMPLE	fuel tanker-gas station	NULL	ref	Fuel_tanker-Fuel_id_idx,Gas_station_id_idx	Fuel_tanker-Fuel_id_idx	4	fueldb.fuel tanker-fuel.Fuel_tanker-Fuel_id	61	100.00	NULL
	1	SIMPLE	gas station	NULL	eq_ref	PRIMARY	PRIMARY	4	fueldb.fuel tanker-gas station.Gas_station_id	1	100.00	NULL

Рис. 2. EXPLAIN запроса 1.1

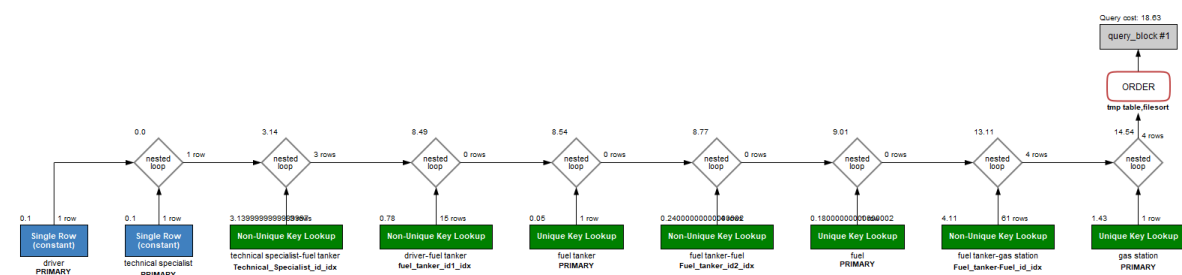


Рис. 3. Execution Plan запроса 1.1

## Объяснение запроса

В запросе объединяются таблицы 'gas station', 'fuel tanker-gas station', 'fuel tanker-fuel', 'fuel', 'fuel tanker', 'driver-fuel tanker', 'driver', 'technical specialist'. Выбираются только строки с определенным значением driver.driver\_id, fuel.Type, 'technical specialist'.Technical\_Specialist\_id.

### 3.1.2 Запрос 1.2

Найти все АЗС, на которые привозил топливо А (type="АИ-92") водитель Б (id=3) не менее 4 раз.

## Код запроса

```
1 select
2     `gas station`.Address,
3     fuel.`Type` as "Тип топлива",
4     driver.Name as "Имя водителя",
5     driver.Surname as "Фамилия водителя",
6     driver.Middle_name as "Отчество водителя",
7     COUNT(*) AS "Подсчет"
8 from
9     `gas station`
10 inner join `fuel tanker-gas station` on `fuel tanker-gas station`.Gas_station_id=`gas
    ↳ station`.Gas_station_id
11 inner join `fuel tanker-fuel` on `fuel tanker-gas
    ↳ station`.`Fuel_tanker-Fuel_id`=`fuel tanker-fuel`.`Fuel_tanker-Fuel_id`
12 inner join fuel on fuel.Fuel_id=`fuel tanker-fuel`.Fuel_id
13 inner join `fuel tanker` on `fuel tanker-fuel`.Fuel_tanker_id=`fuel
    ↳ tanker`.Fuel_tanker_id
14 inner join `driver-fuel tanker` on `driver-fuel tanker`.Fuel_tanker_id=`fuel
    ↳ tanker`.Fuel_tanker_id
15 inner join driver on driver.driver_id=`driver-fuel tanker`.Driver_id
16 where fuel.`Type`="АИ-92" and driver.driver_id=7
17 GROUP BY `gas station`.Address
18 HAVING COUNT(*) >= 4
19 order by `gas station`.Address
```

## Результат запроса

Запрос выполнен за 140 миллисекунд. В результирующей таблице получилась 1 строка.

	Address	Тип топлива	Имя водителя	Фамилия водителя	Отчество водителя	Подсчет
▶	с. Владикавказ, ш. Гоголя, д. 542 к. 2/6, 24...	АИ-92	Милий	Рыбаков	Марсович	4

Рис. 4. Результат запроса 1.2

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	driver	NULL	const	PRIMARY	PRIMARY	4	const	1	100.00	Using temporary; Using filesort
1	SIMPLE	fuel	NULL	ALL	PRIMARY	NULL	NULL	NULL	10	10.00	Using where
1	SIMPLE	driver-fuel tanker	NULL	ref	driver_id_idx,fuel_tanker_id1_idx	driver_id1_idx	4	const	18	100.00	NULL
1	SIMPLE	fuel tanker-fuel	NULL	ref	PRIMARY,Fuel_tanker_id2_idx,Fuel_id_idx	Fuel_tanker_id2_idx	4	fueldb.driver-fuel tanker.Fuel_tanker_id	4	10.00	Using where
1	SIMPLE	fuel tanker	NULL	eq_ref	PRIMARY	PRIMARY	4	fueldb.driver-fuel tanker.Fuel_tanker_id	1	100.00	Using index
1	SIMPLE	fuel tanker-gas station	NULL	ref	Fuel_tanker-Fuel_id_idx,Gas_station_id_idx	Fuel_tanker-Fuel_id_idx	4	fueldb.fuel tanker-fuel.Fuel_tanker-Fuel_id	61	100.00	NULL
1	SIMPLE	gas station	NULL	eq_ref	PRIMARY	PRIMARY	4	fueldb.fuel tanker-gas station.Gas_station_id	1	100.00	NULL

Рис. 5. EXPLAIN запроса 1.2



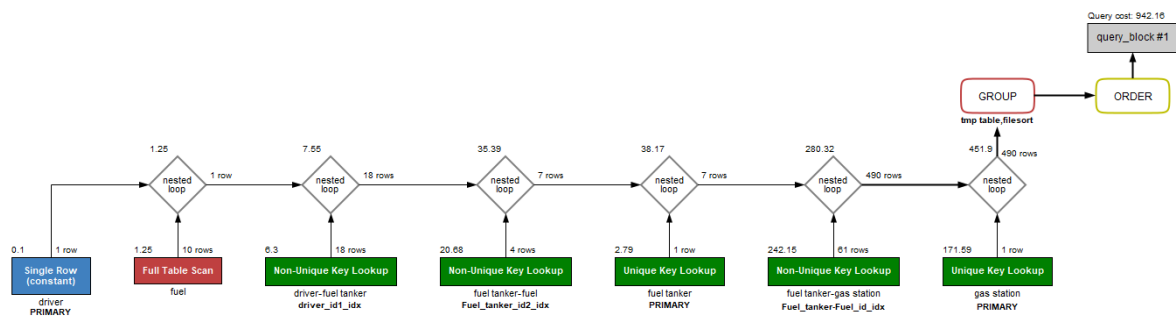


Рис. 6. Execution Plan запроса 1.2

## Объяснение запроса

В запросе объединяются таблицы 'gas station', 'fuel tanker-gas station', 'fuel tanker-fuel', 'fuel', 'fuel tanker', 'driver-fuel tanker', 'driver'. Выбираются только строки с определенным значением driver.driver\_id и fuel.Type. Далее идет группировка по 'gas station'.Address и идет подсчет одинаковых адресов. Далее идет отбор тех значений, где количество больше 4.

## 3.2 Запрос №2

Посчитать число поставок бензовозом А (id=1) топлива Б (Type="АИ-95").

### Первый вариант

#### Код запроса

```

1 select
2     `fuel in tanker`.Model "Модель бензовоза",
3     `fuel in tanker`.`Type` as "Тип топлива",
4     COUNT(*) AS "Число поставок"
5 from (
6     select
7         fuel.`Type`,
8         `fuel tanker`.Model,
9         `fuel tanker-fuel`.`Fuel_tanker-Fuel_id`
10    from
11        `fuel tanker-fuel`
12    inner join `fuel tanker` on `fuel tanker-fuel`.Fuel_tanker_id=`fuel
    ⇐ tanker`.Fuel_tanker_id

```

```

13         inner join `fuel` on fuel.Fuel_id=`fuel tanker-fuel`.`Fuel_tanker-Fuel_id`
14         where `fuel tanker`.Fuel_tanker_id=1 and fuel.`Type`="АИ-95"
15     ) as `fuel in tanker`
16     inner join `fuel tanker-gas station` on
17         `fuel in tanker`.`Fuel_tanker-Fuel_id`=`fuel tanker-gas
        ↪ station`.`Fuel_tanker-Fuel_id`

```

## Результат запроса

Запрос выполнен за 0 миллисекунд. В результирующей таблице получилась 1 строка.

	Модель бензов о́за	Тип топл ива	Число поста вок
►	Татра 815-11	АИ-95	55

Рис. 7. Результат первого варианта запроса 2

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
►	1	SIMPLE	fuel tanker	<small>NULL</small>	const	PRIMARY	PRIMARY	4	const	1	100.00	<small>NULL</small>
	1	SIMPLE	fuel	<small>NULL</small>	ALL	PRIMARY	<small>NULL</small>	<small>NULL</small>	<small>NULL</small>	10	10.00	Using where
	1	SIMPLE	fuel tanker-fuel	<small>NULL</small>	eq_ref	PRIMARY,Fuel_tanker_id2_idx	PRIMARY	4	fueldb.fuel.Fuel_id	1	5.00	Using where
	1	SIMPLE	fuel tanker-gas station	<small>NULL</small>	ref	Fuel_tanker-Fuel_id_idx	Fuel_tanker-Fuel_id_idx	4	fueldb.fuel.Fuel_id	61	100.00	Using index

Рис. 8. EXPLAIN первого варианта запроса 2

## Второй вариант

### Код запроса

```

1 select
2     fuel.`Type` as "Тип топлива",
3     `fuel tanker`.Model as "Модель бензовоза",
4     COUNT(*) AS "Число поставок"
5 from
6     `fuel tanker-fuel`
7 inner join `fuel tanker`
8     on `fuel tanker-fuel`.Fuel_tanker_id=`fuel tanker`.Fuel_tanker_id
9 inner join `fuel`
10    on fuel.Fuel_id=`fuel tanker-fuel`.`Fuel_tanker-Fuel_id`
11 inner join `fuel tanker-gas station`
12    on `fuel tanker-fuel`.`Fuel_tanker-Fuel_id`=`fuel tanker-gas
    ↪ station`.`Fuel_tanker-Fuel_id`
13 where `fuel tanker`.Fuel_tanker_id=1 and fuel.`Type`="АИ-95"

```

## Результат запроса

Запрос выполнен за 0 миллисекунд. В результирующей таблице получилась 1 строка.

	Модель бензов о́за	Тип топл ива	Число поста вок
►	Татра 815-11	АИ-95	55

Рис. 9. Результат второго варианта запроса 2

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	fuel tanker	NULL	const	PRIMARY	PRIMARY	4	const	1	100.00	NULL
1	SIMPLE	fuel	NULL	ALL	PRIMARY	NULL	NULL	NULL	10	10.00	Using where
1	SIMPLE	fuel tanker-fuel	NULL	eq_ref	PRIMARY_Fuel_tanker_id2_idx	PRIMARY	4	fueldb.fuel.Fuel_id	1	5.00	Using where
1	SIMPLE	fuel tanker-gas station	NULL	ref	Fuel_tanker-Fuel_id_idx	Fuel_tanker-Fuel_id_idx	4	fueldb.fuel.Fuel_id	61	100.00	Using index

Рис. 10. EXPLAIN второго варианта запроса 2

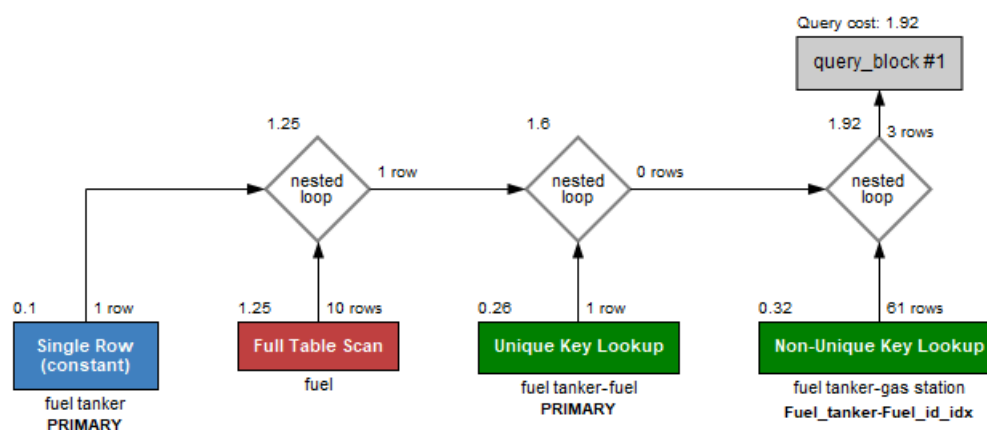


Рис. 11. Execution Plan второго варианта запроса 2

## Объяснение запроса

Соединяем таблицы 'fuel tanker-fuel', 'fuel tanker', 'fuel', 'fuel tanker-gas station'. Далее фильтруем по 'fuel tanker'.Fuel\_tanker\_id и fuel.Type и делаем подсчет строк.

## 3.3 Запрос №3

### 3.3.1 Запрос 3.1

Для каждого вида топлива посчитать число доставок на АЗС.

## Код запроса

```
1 select
2     fuel.`Type`,
3     COUNT(*) AS "Число поставок"
4 from
5     fuel
6 inner join `fuel tanker-fuel` on `fuel tanker-fuel`.Fuel_id=fuel.Fuel_id
7 inner join `fuel tanker-gas station`
8     on `fuel tanker-gas station`.`Fuel_tanker-Fuel_id`=`fuel
9     ⇨ tanker-fuel`.`Fuel_tanker-Fuel_id`
10 group by fuel.`Type`
```

## Результат запроса

Запрос выполнен за 375 миллисекунд. В результирующей таблице получилось 10 строк.

	Type	Число поставок
▶	АИ-80	15573
	АИ-92	14517
	АИ-95	14616
	АИ-98	14846
	АИ-100	14404
	АИ-101	14855
	АИ-102	15838
	Дизельное топливо	15182
	Биодизельное топливо	14576
	Керосин	15281

Рис. 12. Результат запроса 3.1

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	fuel	NULL	ALL	PRIMARY	NULL	NULL	NULL	10	100.00	Using temporary
	1	SIMPLE	fuel tanker-fuel	NULL	ref	PRIMARY,Fuel_id_idx	Fuel_id_idx	4	fueldb.fuel.Fuel_id	238	100.00	Using index
	1	SIMPLE	fuel tanker-gas station	NULL	ref	Fuel_tanker-Fuel_id_idx	Fuel_tanker-Fuel_id_idx	4	fueldb.fuel tanker-fuel.Fuel_tanker-Fuel_id	61	100.00	Using index

Рис. 13. EXPLAIN запроса 3.1

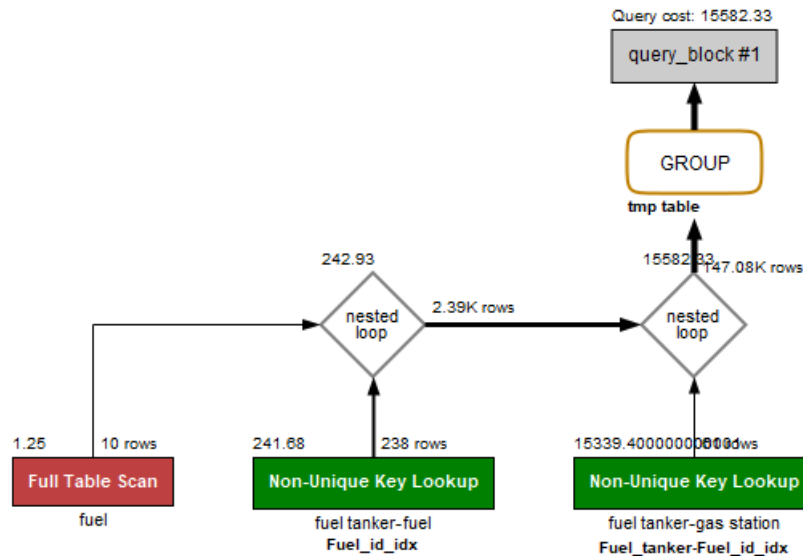


Рис. 14. Execution Plan запроса 3.1

## Объяснение запроса

В запросе объединяются таблицы ‘fuel tanker-fuel’, ‘fuel’, ‘fuel tanker-gas station’. Далее идет группировка по типу топлива и подсчет одинаковых адресов.

### 3.3.2 Запрос 3.2

Для каждого вида топлива посчитать число доставок на АЗС, считая по бензовозам.

## Код запроса

```

1 select
2     fuel.`Type`,
3     COUNT(distinct `fuel tanker-fuel`.Fuel_tanker_id) AS "Число поставок"
4 from
5     fuel
6 inner join `fuel tanker-fuel` on `fuel tanker-fuel`.Fuel_id=fuel.Fuel_id
7 inner join `fuel tanker-gas station` on `fuel tanker-gas
8     ↪ station`.`Fuel_tanker-Fuel_id`=`fuel tanker-fuel`.`Fuel_tanker-Fuel_id`
9 group by fuel.`Type`
  
```

## Результат запроса

Запрос выполнен за 79 миллисекунд. В результирующей таблице получилось 10 строк.

	Тип	Число поставок
▶	АИ-100	231
	АИ-101	238
	АИ-102	251
	АИ-80	249
	АИ-92	235
	АИ-95	233
	АИ-98	235
	Биодизельное топливо	231
	Дизельное топливо	240
	Керосин	243

Рис. 15. Результат запроса 3.2

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	fuel	NULL	ALL	PRIMARY	NULL	NULL	NULL	10	100.00	Using filesort
	1	SIMPLE	fuel tanker-fuel	NULL	ref	PRIMARY,Fuel_id_idx	Fuel_id_idx	4	fueldb.fuel.Fuel_id	238	100.00	Using index
	1	SIMPLE	fuel tanker-gas station	NULL	ref	Fuel_tanker-Fuel_id_idx	Fuel_tanker-Fuel_id_idx	4	fueldb.fuel tanker-gas station.Fuel_tanker-Fuel_id	61	100.00	Using index

Рис. 16. EXPLAIN запроса 3.2

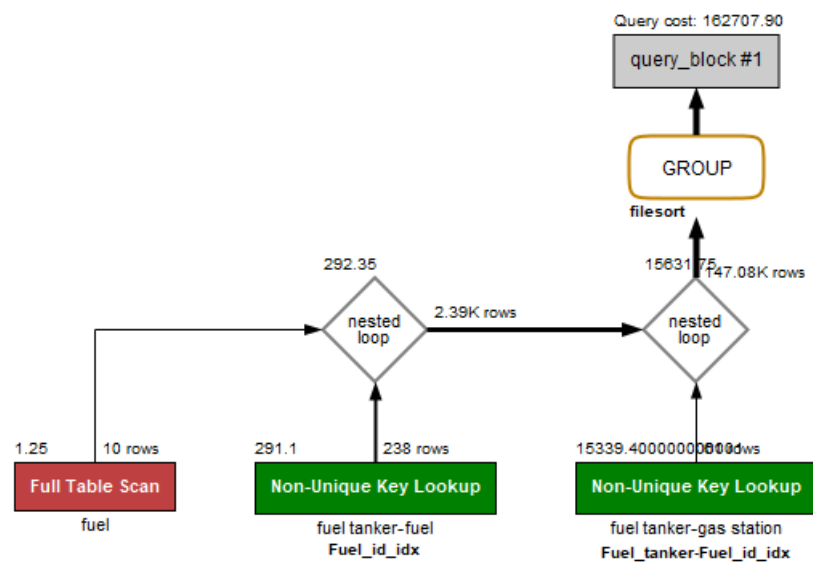


Рис. 17. Execution Plan запроса 3.2

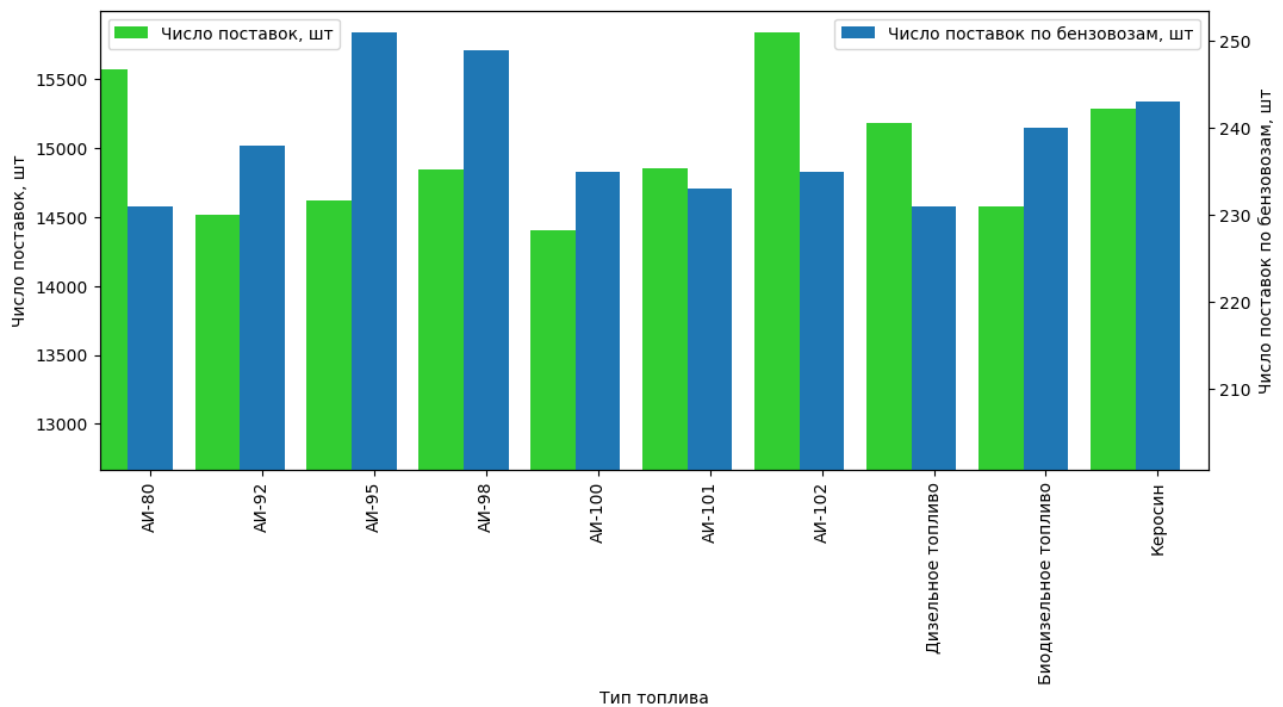


Рис. 18. Гистограмма результата третьего запроса

## Объяснение запроса

В запросе объединяются таблицы ‘fuel tanker-fuel’, ‘fuel’, ‘fuel tanker-gas station’. Далее идет группировка по типу топлива и подсчет по уникальным бензовозам.

## 3.4 Запрос №4

### 3.4.1 Запрос 4.1

Найти технических специалистов с максимальным числом обслуживаемых тар АЗС.

## Первый вариант

### Код запроса

```

1 select
2     `technical specialist`.Surname,
3     `technical specialist`.`Name`,
4     `technical specialist`.Middle_name,
5     MaxContainerSpecialists.`Число обслуживаемых тар`
6 from

```

```

7         `technical specialist`
8 inner join (
9     SELECT
10         Technical_Specialist_id,
11         COUNT(*) AS "Число обслуживаемых тар"
12 FROM `technical specialist-fuel containers`
13 GROUP BY Technical_Specialist_id
14 HAVING COUNT(Fuel_containers_id) = (
15     select max(`Число обслуживаемых тар`)
16 from (
17     select
18         `technical specialist-fuel
19         ↪ containers`.Technical_Specialist_id,
20         COUNT(*) AS "Число обслуживаемых тар"
21 from
22         `technical specialist-fuel containers`
23 group by `technical specialist-fuel
24         ↪ containers`.Technical_Specialist_id
25 ) as maxResult
26 ) as MaxContainerSpecialists
27 on `technical specialist`.Technical_Specialist_id=
    MaxContainerSpecialists.Technical_Specialist_id

```

## Результат запроса

Запрос выполнен за 16 миллисекунд. В результирующей таблице получилось 102 строки.

	Surname	Name	Middle_name	Число обслуживаемых тар
►	Власов	Фирс	Владленович	10
	Евдокимов	Федот	Ефстафьевич	10
	Евдокимов	Рюрик	Гаврилович	10
	Уваров	Матвей	Демьянович	10
	Поляков	Евстигней	Харлампьевич	10
	Шубин	Евстафий	Геннадиевич	10
	Евдокимов	Харлампий	Артёмович	10
	Жуков	Аристарх	Иосифович	10
	Михеев	Милован	Вячеславович	10
	Семенов	Всеволод	Владленович	10
	Ковалев	Сократ	Валентинович	10
	Лобанов	Касьян	Анисимович	10

Рис. 19. Часть результата первого варианта запроса 4.1



	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
	1	PRIMARY	<derived2>	NULL	ALL	NULL	NULL	NULL	NULL	5608	100.00	NULL
	1	PRIMARY	technical specialist	NULL	eq_ref	PRIMARY	PRIMARY	4	MaxContainerSpecialists.Technical_Specialist_id	1	100.00	NULL
	2	DERIVED	technical specialist-f...	NULL	index	Technical_Specialist_id2_idx	Technical_Specialist_id2_idx	4	NULL	5608	100.00	NULL
	3	SUBQUERY	<derived4>	NULL	ALL	NULL	NULL	NULL	NULL	5608	100.00	NULL
	4	DERIVED	technical specialist-f...	NULL	index	Technical_Specialist_id2_idx	Technical_Specialist_id2_idx	4	NULL	5608	100.00	Using index

Рис. 20. EXPLAIN первого варианта запроса 4.1

## Второй вариант

### Код запроса

```

1      select
2      `technical specialist`.Surname,
3      `technical specialist`.`Name`,
4      `technical specialist`.Middle_name,
5      count(*) as `Число обслуживаемых тар`
6  from
7      `technical specialist`
8  inner join `technical specialist-fuel containers`
9  on `technical specialist`.Technical_Specialist_id=`technical specialist-fuel
   ↪ containers`.Technical_Specialist_id
10 group by `technical specialist`.Technical_Specialist_id
11 having `Число обслуживаемых тар` = (
12  select
13  count(*)
14  from
15  `technical specialist-fuel containers`
16  group by `technical specialist-fuel containers`.Technical_Specialist_id
17  order by count(*) desc
18  limit 1
19  )

```

### Результат запроса

Запрос выполнен за 78 миллисекунд. В результирующей таблице получилось 102 строки.

	Surname	Name	Middle_name	Число обслуживаемых тар
▶	Власов	Фирс	Владленович	10
	Евдокимов	Федот	Ефстафьевич	10
	Евдокимов	Рюрик	Гаврилович	10
	Уваров	Матвей	Демьянович	10
	Поляков	Евстигней	Харлампьевич	10
	Шубин	Евстафий	Геннадиевич	10
	Евдокимов	Харлампий	Артёмович	10
	Жуков	Аристарх	Иосифович	10
	Михеев	Милован	Вячеславович	10
	Семенов	Всеволод	Владленович	10
	Ковалев	Сократ	Валентинович	10
	Лобанов	Касьян	Анисимович	10

Рис. 21. Часть результата второго варианта запроса 4.1

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	PRIMARY	technical specialist	index	PRIMARY	PRIMARY	PRIMARY	4	index	1000	100.00	index
	1	PRIMARY	technical specialist-f...	ref	Technical_Specialist_id2_idx	Technical_Specialist_id2_idx	Technical_Specialist_id2_idx	4	fueldb.technical specialist.Technical_Specialist_id	5	100.00	Using index
	2	SUBQUERY	technical specialist-f...	index	Technical_Specialist_id2_idx	Technical_Specialist_id2_idx	Technical_Specialist_id2_idx	4	index	5608	100.00	Using index; Using temporary; Using filesort

Рис. 22. EXPLAIN второго варианта запроса 4.1

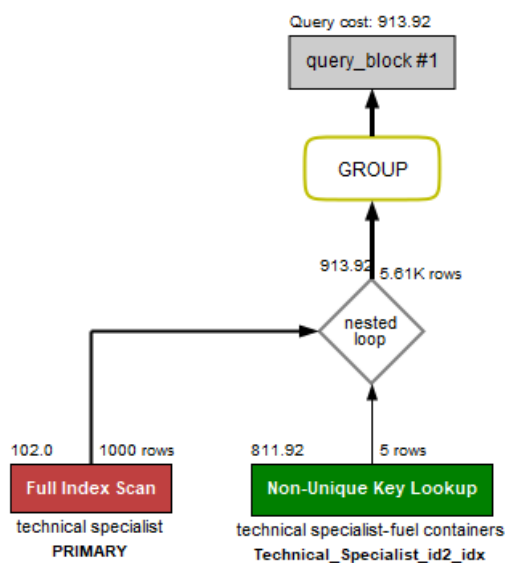


Рис. 23. Execution Plan второго варианта запроса 4.1

## Объяснение запроса

Сначала считаем количество обслуживаемых тар для каждого специалиста, сортируем по убыванию и оставляем число обслуживаемых тар

из первой строки. После этого снова считаем количество обслуживаемых тар для каждого специалиста и отбираем тех, у кого количество равно максимальному.

### 3.4.2 Запрос 4.2

Найти технических специалистов с минимальным числом обслуживаемых тар АЗС.

#### Первый вариант

#### Код запроса

```
1 select
2     `technical specialist`.Surname,
3     `technical specialist`.`Name`,
4     `technical specialist`.Middle_name,
5     MinContainerSpecialists.`Число обслуживаемых тар`
6 from
7     `technical specialist`
8 inner join (
9     SELECT
10         Technical_Specialist_id,
11         COUNT(*) AS "Число обслуживаемых тар"
12 FROM `technical specialist-fuel containers`
13 GROUP BY Technical_Specialist_id
14 HAVING COUNT(Fuel_containers_id) = (
15     select
16         min(`Число обслуживаемых тар`)
17     from (
18         select
19             `technical specialist-fuel
20             ↪ containers`.Technical_Specialist_id,
21             COUNT(*) AS "Число обслуживаемых тар"
22         from
23             `technical specialist-fuel containers`
24         group by `technical specialist-fuel
25             ↪ containers`.Technical_Specialist_id
26         ) as minResult
27     )
28 ) as MinContainerSpecialists
29 on `technical specialist`.Technical_Specialist_id=
30 MinContainerSpecialists.Technical_Specialist_id
```

## Результат запроса

Запрос выполнен за 16 миллисекунд. В результирующей таблице получилось 71 строка.

	Surname	Name	Middle_name	Число обслуживаемых тар
►	Шаров	Самуил	Демьянович	1
	Кондратьев	Ладислав	Арсенович	1
	Гурьев	Нифонт	Венедиктович	1
	Фомичев	Александр	Антонович	1
	Александров	Радован	Ефимьевич	1
	Буров	Валерий	Германович	1
	Кулаков	Кондрат	Афанасьевич	1
	Емельянов	Арсений	Валерианович	1
	Орехов	Пимен	Ильясович	1
	Лобанов	Боян	Захарьевич	1

Рис. 24. Часть результата первого варианта запроса 4.2

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
► 1	PRIMARY	<derived2>	NULL	ALL	NULL	NULL	NULL	NULL	5608	100.00	NULL
1	PRIMARY	technical specialist	NULL	eq_ref	PRIMARY	PRIMARY	4	MinContainerSpecialists.Technical_Specialist_id	1	100.00	NULL
2	DERIVED	technical specialist-f...	NULL	index	Technical_Specialist_id2_idx	Technical_Specialist_id2_idx	4	NULL	5608	100.00	NULL
3	SUBQUERY	<derived4>	NULL	ALL	NULL	NULL	NULL	NULL	5608	100.00	NULL
4	DERIVED	technical specialist-f...	NULL	index	Technical_Specialist_id2_idx	Technical_Specialist_id2_idx	4	NULL	5608	100.00	Using index

Рис. 25. EXPLAIN первого варианта запроса 4.2

## Второй вариант

### Код запроса

```
1 select
2     `technical specialist`.Surname,
3     `technical specialist`.`Name`,
4     `technical specialist`.Middle_name,
5     count(*) as `Число обслуживаемых тар`
6 from
7     `technical specialist`
8 inner join `technical specialist-fuel containers`
9     on `technical specialist`.Technical_Specialist_id=`technical specialist-fuel
    ↪ containers`.Technical_Specialist_id
10 group by `technical specialist`.Technical_Specialist_id
11 having `Число обслуживаемых тар` = (
12     select
13         count(*)
14     from
15         `technical specialist-fuel containers`
```

```

16      group by `technical specialist-fuel containers`.Technical_Specialist_id
17      order by count(*) asc
18      limit 1
19  )

```

## Результат запроса

Запрос выполнен за 16 миллисекунд. В результирующей таблице получилась 71 строка.

	Surname	Name	Middle_name	Число обслуживаемых тар
►	Шаров	Самуил	Демьянович	1
	Кондратьев	Ладислав	Арсенович	1
	Гурьев	Нифонт	Венедиктович	1
	Фомичев	Александр	Антонович	1
	Александров	Радован	Ефимьевич	1
	Буров	Валерий	Германович	1
	Кулаков	Кондрат	Афанасьевич	1
	Емельянов	Арсений	Валерианович	1
	Орехов	Пимен	Ильясович	1
	Лобанов	Боян	Захарьевич	1

Рис. 26. Часть результата второго варианта запроса 4.2

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
►	1	PRIMARY	technical specialist	NULL	index	PRIMARY	PRIMARY	4	NULL	1000	100.00	NULL
	1	PRIMARY	technical specialist-f...	NULL	ref	Technical_Specialist_id2_idx	Technical_Specialist_id2_idx	4	fueldb.technical specialist.Technical_Specialist_id	5	100.00	Using index
	2	SUBQUERY	technical specialist-f...	NULL	index	Technical_Specialist_id2_idx	Technical_Specialist_id2_idx	4	NULL	5608	100.00	Using index; Using temporary; Using filesort

Рис. 27. EXPLAIN второго варианта запроса 4.2

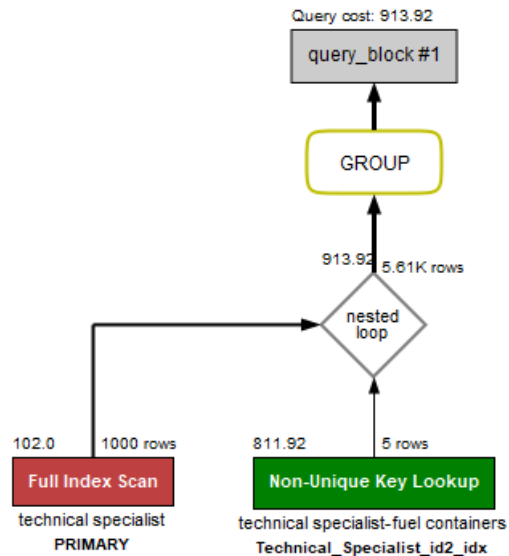


Рис. 28. Execution Plan второго варианта запроса 4.2

## Объяснение запроса

Сначала считаем количество обслуживаемых тар для каждого специалиста, сортируем по убыванию и оставляем число обслуживаемых тар из первой строки. После этого снова считаем количество обслуживаемых тар для каждого специалиста и отбираем тех, у кого количество равно максимальному.

## 3.5 Запрос №5

Посчитать число бензовозов с одинаковым числом доставок.

### Код запроса

```

1 select
2     count(*) as "Число бензовозов",
3     deliveryCount. `Число доставок`
4 from (
5     select
6         `fuel tanker-fuel`. `Fuel_tanker_id`,
7         count(*) as "Число доставок"
8     from
9         `fuel tanker-fuel`
10    inner join `fuel tanker-gas station`
  
```

```

11         on `fuel tanker-gas station`.`Fuel_tanker-Fuel_id`=
12         `fuel tanker-fuel`.`Fuel_tanker-Fuel_id`
13     group by `fuel tanker-fuel`.`Fuel_tanker_id`
14 ) as deliveryCount
15 group by deliveryCount.`Число доставок`
16 order by deliveryCount.`Число доставок`

```

## Результат запроса

Запрос выполнен за 62 миллисекунды. В результирующей таблице получилось 271 строка.

	Число бензовозов	Число доставок
▶ 1	103	
2	107	
1	108	
1	109	
1	110	
2	112	
2	114	
1	115	
5	116	
3	117	
1	118	
4	119	

Рис. 29. Часть результата запроса 5

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	PRIMARY	<derived2>	NULL	ALL	NULL	NULL	NULL	NULL	147076	100.00	Using temporary; Using filesort
	2	DERIVED	fuel tanker-fuel	NULL	index	PRIMARY,Fuel_tanker_id2_idx	Fuel_tanker_id2_idx	4	NULL	2386	100.00	Using index
	2	DERIVED	fuel tanker-gas station	NULL	ref	Fuel_tanker-Fuel_id_idx	Fuel_tanker-Fuel_id_idx	4	fueldb.fuel tanker-fuel.Fuel_tanker-Fuel_id	61	100.00	Using index

Рис. 30. EXPLAIN запроса 5

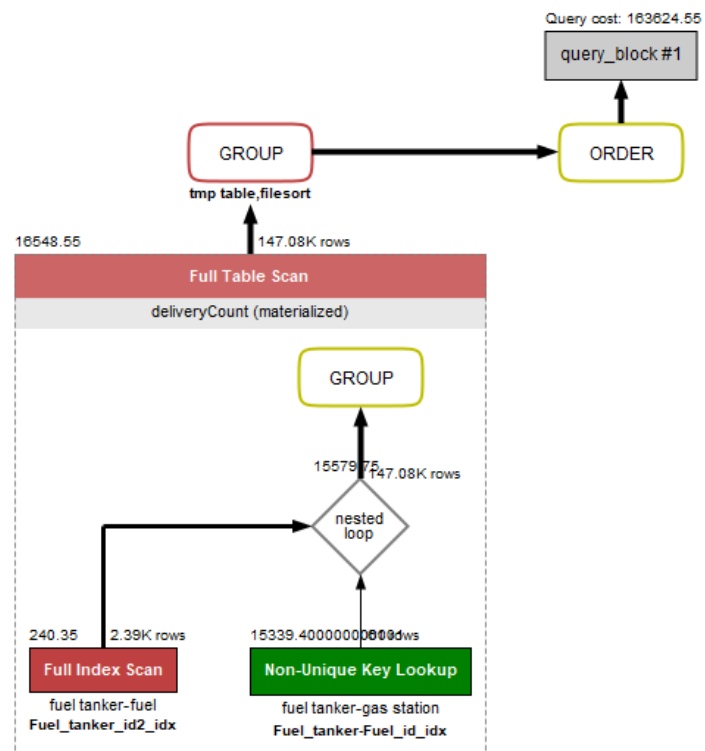


Рис. 31. Execution Plan запроса 5



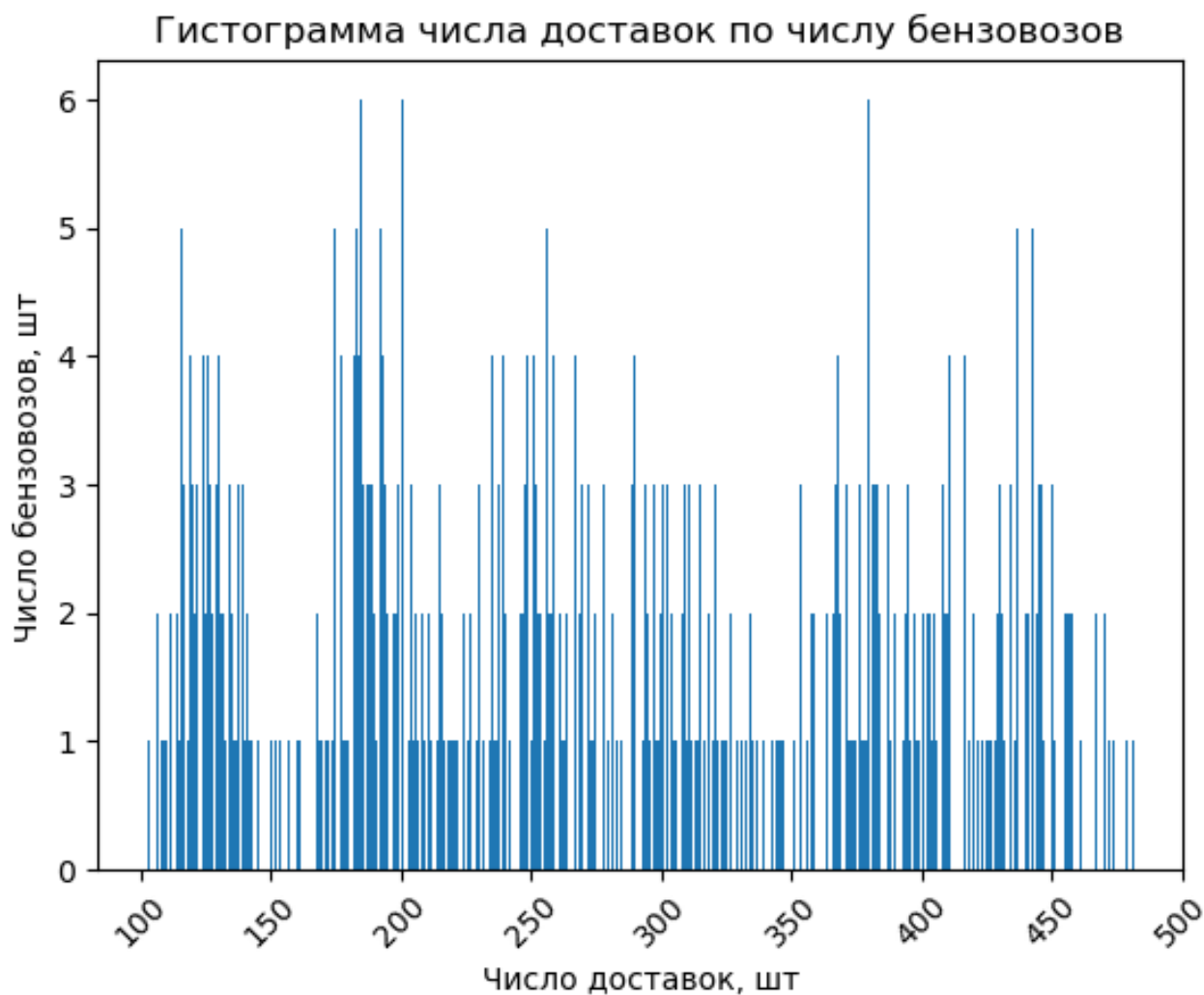


Рис. 32. Гистограмма результата пятого запроса

## Объяснение запроса

Сначала в подзапросе считается количество доставок для каждого бензовоза. Далее все записи группируются по числу доставок и считается количество бензовозов для каждого числа.

## 3.6 Запрос №6

Найти АЗС, на которые топливо типа А ('Type'="АИ-95") поставляли больше, чем на АЗС Б (id=1) топлива С ('Type'="АИ-92").

## Первый вариант

### Код запроса

```
1 WITH `ПопорТабл` AS (  
2     select  
3         firstStation.Address as `АЗС 1`,  
4         firstStation.`Type` as `Тип 1`,  
5         count(*) as `Попор`  
6     from (  
7         select  
8             `gas station`.Address,  
9             fuel.`Type`  
10        from  
11            `gas station`  
12        inner join `fuel tanker-gas station`  
13            on `fuel tanker-gas station`.Gas_station_id=`gas  
14                ↳ station`.Gas_station_id  
15        inner join `fuel tanker-fuel`  
16            on `fuel tanker-fuel`.`Fuel_tanker-Fuel_id`=  
17                `fuel tanker-gas station`.`Fuel_tanker-Fuel_id`  
18        inner join fuel on fuel.Fuel_id=`fuel tanker-fuel`.Fuel_id  
19        where fuel.`Type`="АИ-92" and `gas station`.Gas_station_id=1  
20    ) as firstStation  
21 )  
22 select  
23     `gas station`.Address as `АЗС 2`,  
24     fuel.`Type` as `Тип 2`,  
25     count(*) as `Число доставок на 2 АЗС`,  
26     (select `Попор` from `ПопорТабл`) as `Попор`,  
27     (select `АЗС 1` from `ПопорТабл`) as `АЗС 1`,  
28     (select `Тип 1` from `ПопорТабл`) as `Тип 1`  
29 from  
30     `gas station`  
31 inner join `fuel tanker-gas station`  
32     on `fuel tanker-gas station`.Gas_station_id=`gas station`.Gas_station_id  
33 inner join `fuel tanker-fuel`  
34     on `fuel tanker-fuel`.`Fuel_tanker-Fuel_id`=`fuel tanker-gas  
35         ↳ station`.`Fuel_tanker-Fuel_id`  
36 inner join fuel on fuel.Fuel_id=`fuel tanker-fuel`.Fuel_id  
37 where fuel.`Type`="АИ-95"  
38 group by `gas station`.Gas_station_id  
39 having count(*) > (select `Попор` from `ПопорТабл`)
```

## Результат запроса

Запрос выполнен за 125 миллисекунд. В результирующей таблице получилось 268 строк.

	АЗС 2	Тип 2	Число доставок на 2 АЗС	Порог	АЗС 1	Тип 1
▶	клх Яхрома, ш. Большевикское, д. 8 стр. 2,...	АИ-95	14	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	д. Углегорск, алл. Курская, д. 3/3 стр. 5/6, ...	АИ-95	12	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	п. Вытегра, алл. Беляева, д. 972 к. 684, 436...	АИ-95	13	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	ст. Новомосковск, пр. Ильича, д. 2 стр. 2, 4...	АИ-95	12	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	ст. Кондопога, бул. Школьный, д. 5/3, 576757	АИ-95	12	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	к. Тымовское, ул. Микрорайон, д. 3 к. 47, 10...	АИ-95	14	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	клх Нальчик, алл. Красина, д. 63 стр. 6/7, 5...	АИ-95	11	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	к. Нарьян-Мар, пр. Пушкина, д. 9/8 к. 993, 3...	АИ-95	12	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	с. Ногинск (Моск.), пер. Калинина, д. 464, 7...	АИ-95	13	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	д. Егорьевск, ш. Крайнее, д. 434, 844897	АИ-95	12	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	п. Арнавири, пр. Лазурный, д. 622, 644145	АИ-95	12	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	д. Горячинск, наб. Коминтерна, д. 7/4, 067202	АИ-95	14	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	п. Усть-Кут, бул. Братский, д. 46 стр. 7, 264...	АИ-95	14	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	к. Усть-Ишим, пр. Аэродромный, д. 8/8, 225...	АИ-95	11	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	ст. Котельнич, пр. Девосский, д. 448 к. 2, 5...	АИ-95	11	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	клх Кропоткин (Краснод.), алл. Локомотивн...	АИ-95	13	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92

Рис. 33. Часть результата первого варианта запроса 6

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	fuel	NONE	ALL	PRIMARY	NONE	NONE	NONE	10	10.00	Using temporary
1	PRIMARY	fuel tankers-fuel	NONE	ref	PRIMARY_Fuel_id_idx	Fuel_id_idx	4	fueldb.fuel.Fuel_id	238	100.00	Using index
1	PRIMARY	fuel tankers-gas station	NONE	ref	Fuel_tanker_Fuel_id_idx, Gas_station_id_idx	Fuel_tanker_Fuel_id_idx	4	fueldb.fuel tankers-fuel.Fuel_tanker_Fuel_id	61	100.00	NONE
1	PRIMARY	gas station	NONE	eq_ref	PRIMARY	PRIMARY	4	fueldb.fuel tankers-gas station.Gas_station_id	1	100.00	NONE
11	SUBQUERY	<derived3>	NONE	system	NONE	NONE	NONE	NONE	1	100.00	NONE
8	SUBQUERY	<derived3>	NONE	system	NONE	NONE	NONE	NONE	1	100.00	NONE
5	SUBQUERY	<derived3>	NONE	system	NONE	NONE	NONE	NONE	1	100.00	NONE
2	SUBQUERY	<derived3>	NONE	system	NONE	NONE	NONE	NONE	1	100.00	NONE
3	DERIVED	gas station	NONE	const	PRIMARY	PRIMARY	4	const	1	100.00	NONE
3	DERIVED	fuel	NONE	ALL	PRIMARY	NONE	NONE	NONE	10	10.00	Using where
3	DERIVED	fuel tankers-gas station	NONE	ref	Fuel_tanker_Fuel_id_idx, Gas_station_id_idx	Gas_station_id_idx	4	const	77	100.00	NONE
3	DERIVED	fuel tankers-fuel	NONE	eq_ref	PRIMARY_Fuel_id_idx	PRIMARY	4	fueldb.fuel tankers-gas station.Fuel_tanker_F...	1	10.00	Using where

Рис. 34. EXPLAIN первого варианта запроса 6

## Второй вариант

### Код запроса

```
1 select
2     `gas station`.Address as `Адрес`,
3     fuel.`Type` as `Тип А`,
4     count(*) as `Число доставок на АЗС`
5 from
6     `gas station`
7 inner join `fuel tankers-gas station`
8     on `fuel tankers-gas station`.Gas_station_id=`gas station`.Gas_station_id
9 inner join `fuel tankers-fuel`
10    on `fuel tankers-fuel`.Fuel_tanker_Fuel_id=`fuel tankers-gas
    ↳ station`.Fuel_tanker_Fuel_id`
11 inner join fuel
12    on fuel.Fuel_id=fuel tankers-fuel.Fuel_id
```

```

13 where fuel.`Type`="АИ-95"
14 group by `gas station`.Gas_station_id
15 having count(*) > (
16     select
17         count(*)
18     from
19         `gas station`
20     inner join `fuel tanker-gas station`
21         on `fuel tanker-gas station`.Gas_station_id=`gas
        ↳ station`.Gas_station_id
22     inner join `fuel tanker-fuel`
23         on `fuel tanker-fuel`.`Fuel_tanker-Fuel_id`=`fuel tanker-gas
        ↳ station`.`Fuel_tanker-Fuel_id`
24     inner join fuel
25         on fuel.Fuel_id=`fuel tanker-fuel`.Fuel_id
26     where fuel.`Type`="АИ-92" and `gas station`.Gas_station_id=1
27 )

```

## Результат запроса

Запрос выполнен за 219 миллисекунд. В результирующей таблице получилось 268 строк.

	АЗС 2	Тип 2	Число доставок на 2 АЗС	Порог	АЗС 1	Тип 1
▶	клх Яхрома, ш. Большевикское, д. 8 стр. 2,...	АИ-95	14	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	д. Углегорск, алл. Курская, д. 3/3 стр. 5/6, ...	АИ-95	12	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	п. Вытегра, алл. Беляева, д. 972 к. 684, 436...	АИ-95	13	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	ст. Новомосковск, пр. Ильича, д. 2 стр. 2, 4...	АИ-95	12	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	ст. Кондопога, бул. Школьный, д. 5/3, 576757	АИ-95	12	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	к. Тымовское, ул. Микрорайон, д. 3 к. 47, 10...	АИ-95	14	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	клх Нальчик, алл. Красина, д. 63 стр. 6/7, 5...	АИ-95	11	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	к. Нарьян-Мар, пр. Пушкина, д. 9/8 к. 993, 3...	АИ-95	12	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	с. Ногинск (Моск.), пер. Калинина, д. 464, 7...	АИ-95	13	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	д. Егорьевск, ш. Крайнее, д. 434, 844897	АИ-95	12	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	п. Арнавири, пр. Лазурный, д. 622, 644145	АИ-95	12	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	д. Горькинск, наб. Коминтерна, д. 7/4, 067202	АИ-95	14	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	п. Усть-Кут, бул. Братский, д. 46 стр. 7, 264...	АИ-95	14	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	к. Усть-Ишим, пр. Аэродромный, д. 8/8, 225...	АИ-95	11	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	ст. Котельнич, пр. Деповский, д. 448 к. 2, 5...	АИ-95	11	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92
	клх Кропоткин (Краснод.), алл. Локомотивн...	АИ-95	13	10	г. Тура, бул. Садовый, д. 47 к. 61, 295030	АИ-92

Рис. 35. Часть результата второго варианта запроса 6

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	PRIMARY	fuel	NULL	ALL	PRIMARY	NULL	NULL	NULL	10	10.00	Using where; Using temporary
	1	PRIMARY	fuel tanker-fuel	NULL	ref	PRIMARY_Fuel_id_idx	Fuel_id_idx	4	fueldb.fuel.Fuel_id	238	100.00	Using index
	1	PRIMARY	fuel tanker-gas station	NULL	ref	Fuel_tanker-Fuel_id_idx, Gas_station_id_idx	Fuel_tanker-Fuel_id_idx	4	fueldb.fuel tanker-fuel.Fuel_tanker-Fuel_id	61	100.00	NULL
	1	PRIMARY	gas station	NULL	eq_ref	PRIMARY	PRIMARY	4	fueldb.fuel tanker-gas station.Gas_station_id	1	100.00	NULL
	2	SUBQUERY	fuel	NULL	const	PRIMARY	PRIMARY	4	const	1	100.00	Using index
	2	SUBQUERY	fuel tanker-gas station	NULL	ALL	PRIMARY	PRIMARY	4	const	10	10.00	Using where
	2	SUBQUERY	fuel tanker-gas station	NULL	ref	Fuel_tanker-Fuel_id_idx, Gas_station_id_idx	Gas_station_id_idx	4	const	77	100.00	NULL
	2	SUBQUERY	fuel tanker-fuel	NULL	eq_ref	PRIMARY_Fuel_id_idx	PRIMARY	4	fueldb.fuel tanker-gas station.Fuel_tanker-Fuel_id	1	10.00	Using where

Рис. 36. EXPLAIN второго варианта запроса 6

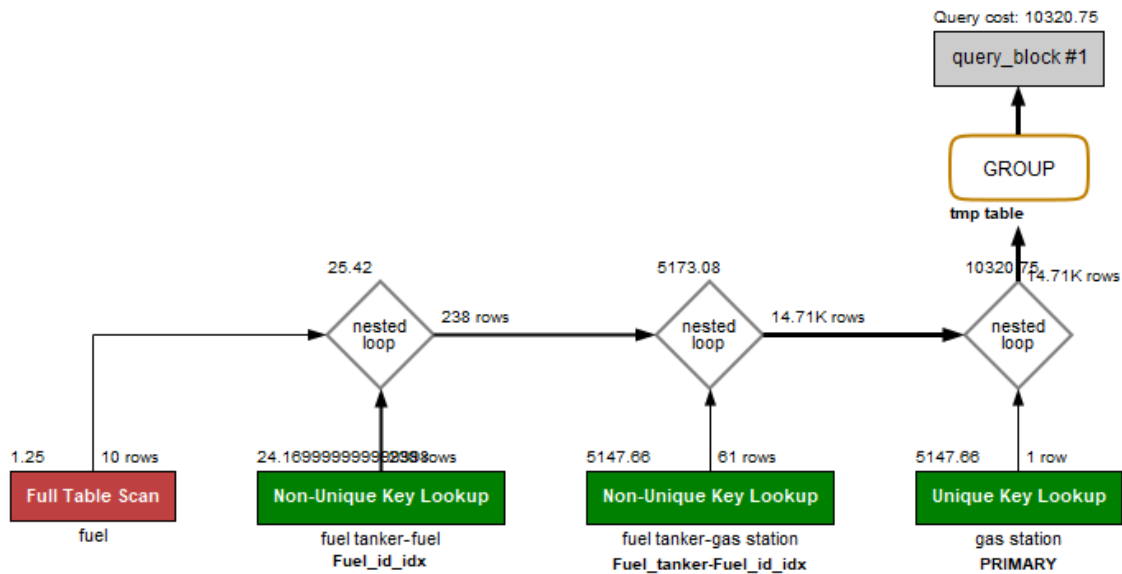


Рис. 37. Execution Plan второго варианта запроса 6

## Объяснение запроса

Сначала в подзапросе считаем, сколько было доставок топлива С на АЗС Б. Далее считаем с помощью группировки по АЗС и фильтрации по типу топлива А доставки этого топлива на каждую АЗС и через HAVING отбираем те, у которых значение больше порогового.

## 3.7 Запрос №7

Найти водителей, которые не доставляли топливо А ('Type'="АИ-92") ни на одну АЗС.

### Первый вариант

#### Код запроса

```

1 with `ВодителиДоставка` as (
2     select
3         `driver-fuel tanker`.Driver_id,
4         fuel.`Type`
5     from
6         `fuel tanker-gas station`
7     inner join `fuel tanker-fuel`

```

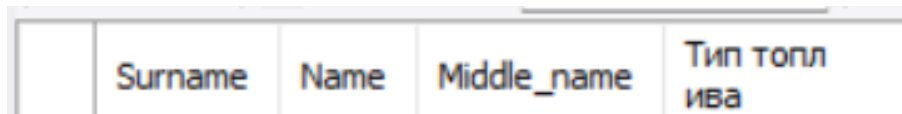
```

8         on `fuel tanker-gas station`.`Fuel_tanker-Fuel_id`
9         =`fuel tanker-fuel`.`Fuel_tanker-Fuel_id`
10    inner join fuel on fuel.Fuel_id=`fuel tanker-fuel`.Fuel_id
11    inner join `fuel tanker`
12         on `fuel tanker`.Fuel_tanker_id=`fuel tanker-fuel`.Fuel_tanker_id
13    inner join `driver-fuel tanker`
14         on `fuel tanker`.Fuel_tanker_id=`driver-fuel tanker`.Fuel_tanker_id
15    where fuel.`Type`="АИ-92"
16 )
17
18 select
19     driver.Surname,
20     driver.`Name`,
21     driver.Middle_name,
22     (select `Type` from `ВодителиДоставка`) as `Тип топлива`
23 from
24     driver
25 where driver.driver_id not in (select `Driver_id` from `ВодителиДоставка`)

```

## Результат запроса

Запрос выполнялся за 110 миллисекунд. В результирующей таблице получилось 0 строк.



Surname	Name	Middle_name	Тип топлива
---------	------	-------------	-------------

Рис. 38. Часть результата первого варианта запроса 7

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	driver	INDEX	ALL	INDEX	INDEX	INDEX	INDEX	500	100.00	INDEX
1	PRIMARY	driver-fuel tanker	INDEX	ref	driver_id1_idx,fuel_tanker_id1_idx	driver_id1_idx	4	fueldb.driver.driver_id	16	100.00	Using where; Not exists
1	PRIMARY	fuel tanker	INDEX	eq_ref	PRIMARY	PRIMARY	4	fueldb.driver-fuel tanker.Fuel_tanker_id	1	100.00	Using index
1	PRIMARY	fuel tanker-fuel	INDEX	ref	PRIMARY,Fuel_tanker_id2_idx,Fuel_id_idx	Fuel_tanker_id2_idx	4	fueldb.fuel tanker.Fuel_tanker_id	4	100.00	Using where
1	PRIMARY	fuel	INDEX	eq_ref	PRIMARY	PRIMARY	4	fueldb.fuel tanker-fuel.Fuel_id	1	100.00	Using where
1	PRIMARY	fuel tanker-gas station	INDEX	ref	Fuel_tanker-Fuel_id_idx	Fuel_tanker-Fuel_id_idx	4	fueldb.fuel tanker-fuel.Fuel_tanker-Fuel_id	61	100.00	Using index
2	SUBQUERY	fuel	INDEX	ALL	PRIMARY	INDEX	INDEX	INDEX	10	10.00	Using where
2	SUBQUERY	fuel tanker-fuel	INDEX	ref	PRIMARY,Fuel_tanker_id2_idx,Fuel_id_idx	Fuel_id_idx	4	fueldb.fuel.Fuel_id	238	100.00	INDEX
2	SUBQUERY	fuel tanker	INDEX	eq_ref	PRIMARY	PRIMARY	4	fueldb.fuel tanker-fuel.Fuel_tanker_id	1	100.00	Using index
2	SUBQUERY	driver-fuel tanker	INDEX	ref	fuel_tanker_id1_idx	fuel_tanker_id1_idx	4	fueldb.fuel tanker-fuel.Fuel_tanker_id	15	100.00	Using index
2	SUBQUERY	fuel tanker-gas station	INDEX	ref	Fuel_tanker-Fuel_id_idx	Fuel_tanker-Fuel_id_idx	4	fueldb.fuel tanker-fuel.Fuel_tanker-Fuel_id	61	100.00	Using index

Рис. 39. EXPLAIN первого варианта запроса 7

## Второй вариант

### Код запроса

```

1 select
2     driver.Surname,
3     driver.`Name`,
4     driver.Middle_name
5 from
6     driver
7 where driver.driver_id not in (
8     select
9         distinct `driver-fuel tanker`.Driver_id
10    from
11        `fuel tanker-gas station`
12    inner join `fuel tanker-fuel`
13        on `fuel tanker-gas station`.`Fuel_tanker-Fuel_id`=`fuel
14        ↪ tanker-fuel`.`Fuel_tanker-Fuel_id`
15    inner join fuel
16        on fuel.Fuel_id=`fuel tanker-fuel`.Fuel_id
17    inner join `fuel tanker`
18        on `fuel tanker`.Fuel_tanker_id=`fuel tanker-fuel`.Fuel_tanker_id
19    inner join `driver-fuel tanker`
20        on `fuel tanker`.Fuel_tanker_id=`driver-fuel tanker`.Fuel_tanker_id
21 where fuel.`Type`="Тестовое топливо"
22 )

```

## Результат запроса

Запрос выполнялся за 62 миллисекунды. В результирующей таблице получилось 0 строк.

Surname	Name	Middle_name	Тип топлива
---------	------	-------------	-------------

Рис. 40. Часть результата второго варианта запроса 7

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	driver	INDEX	ALL	INDEX	INDEX	INDEX	INDEX	500	100.00	INDEX
1	SIMPLE	driver-fuel tanker	INDEX	ref	driver_id1_idx,fuel_tanker_id1_idx	driver_id1_idx	4	fueldb.driver.driver_id	16	100.00	Using where; Not exists
1	SIMPLE	fuel tanker	INDEX	eq_ref	PRIMARY	PRIMARY	4	fueldb.driver-fuel tanker.Fuel_tanker_id	1	100.00	Using index
1	SIMPLE	fuel tanker-fuel	INDEX	ref	PRIMARY,Fuel_tanker_id2_idx,Fuel_id_idx	Fuel_tanker_id2_idx	4	fueldb.fuel tanker.Fuel_tanker_id	4	100.00	Using where
1	SIMPLE	fuel	INDEX	eq_ref	PRIMARY	PRIMARY	4	fueldb.fuel tanker-fuel.Fuel_id	1	100.00	Using where
1	SIMPLE	fuel tanker-gas station	INDEX	ref	Fuel_tanker-Fuel_id_idx	Fuel_tanker-Fuel_id_idx	4	fueldb.fuel tanker-fuel.Fuel_tanker-Fuel_id	61	100.00	Using index

Рис. 41. EXPLAIN второго варианта запроса 7

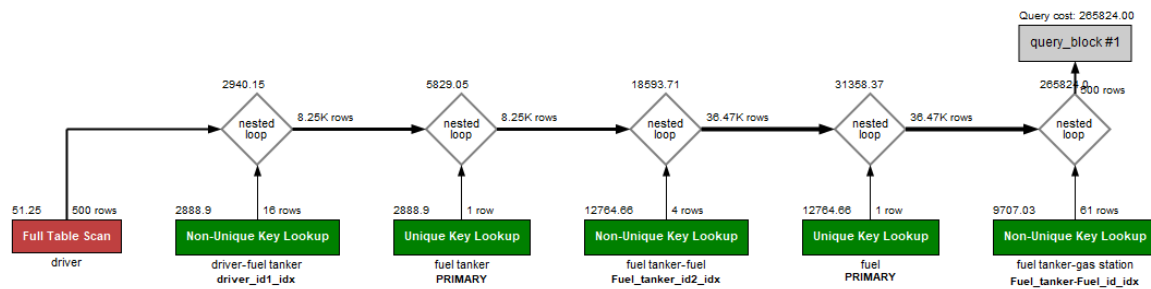


Рис. 42. Execution Plan первого варианта запроса 7

## Объяснение запроса

Сначала через подзапрос получаем список id всех водителей, доставлявших вид топлива А хотя бы 1 раз. После этого получаем тех водителей, которых в этом списке нет.

## 3.8 Запрос №8

Для каждого водителя и каждого вида топлива посчитать число доставок.

### Первый вариант

#### Код запроса

```

1  select
2      driver.Surname,
3      driver.`Name`,
4      driver.Middle_name,
5      `Type`,
6      `Количество` `доставок`
7  from
8      driver
9  inner join (
10     select
11         `driver-fuel tanker`.Driver_id,
12         fuel.`Type`,
13         count(*) as `Количество` `доставок`
14     from
15         `fuel tanker-gas station`
16     inner join `fuel tanker-fuel`
17     on `fuel tanker-gas station`.`Fuel_tanker-Fuel_id`=

```



```

18         `fuel tanker-fuel`.`Fuel_tanker-Fuel_id`
19     inner join fuel on fuel.Fuel_id=`fuel tanker-fuel`.Fuel_id
20     inner join `fuel tanker`
21         on `fuel tanker`.Fuel_tanker_id=
22         `fuel tanker-fuel`.Fuel_tanker_id
23     inner join `driver-fuel tanker`
24         on `fuel tanker`.Fuel_tanker_id=`driver-fuel tanker`.Fuel_tanker_id
25     group by Driver_id, `Type`
26 ) as deliveriesNum on deliveriesNum.Driver_id=driver.driver_id

```

## Результат запроса

Запрос выполнен за 7265 миллисекунд. В результирующей таблице получилось 4997 строк.

	Surname	Name	Middle_name	Type	Количество доста вок
►	Кондратьев	Аполлинарий	Вилорович	АИ-100	398
	Кондратьев	Аполлинарий	Вилорович	АИ-101	190
	Кондратьев	Аполлинарий	Вилорович	АИ-102	510
	Кондратьев	Аполлинарий	Вилорович	АИ-80	753
	Кондратьев	Аполлинарий	Вилорович	АИ-92	493
	Кондратьев	Аполлинарий	Вилорович	АИ-95	358
	Кондратьев	Аполлинарий	Вилорович	АИ-98	174
	Кондратьев	Аполлинарий	Вилорович	Биодизельное топливо	128
	Кондратьев	Аполлинарий	Вилорович	Дизельное топливо	234
	Кондратьев	Аполлинарий	Вилорович	Керосин	472
	Гурьев	Прокл	Бенедикто...	АИ-100	262
	Гурьев	Прокл	Бенедикто...	АИ-101	253
	Гурьев	Прокл	Бенедикто...	АИ-102	445
	Гурьев	Прокл	Бенедикто...	АИ-80	253
	Гурьев	Прокл	Бенедикто...	АИ-92	435
	Гурьев	Прокл	Бенедикто...	АИ-95	318

Рис. 43. Часть результата первого варианта запроса 8

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
► 1	PRIMARY	driver	NULL	ALL	PRIMARY	NULL	NULL	NULL	500	100.00	NULL
1	PRIMARY	<derived2>	NULL	ref	<auto_key0>	<auto_key0>	4	fueldb.driver.driver_id	4496	100.00	NULL
2	DERIVED	fuel	NULL	ALL	PRIMARY	NULL	NULL	NULL	10	100.00	Using temporary
2	DERIVED	fuel tanker-fuel	NULL	ref	PRIMARY,Fuel_tanker_id2_idx,Fuel_id_idx	Fuel_id_idx	4	fueldb.fuel.Fuel_id	238	100.00	Using index
2	DERIVED	fuel tanker	NULL	eq_ref	PRIMARY	PRIMARY	4	fueldb.fuel tanker-fuel.Fuel_tanker_id	1	100.00	Using index
2	DERIVED	driver-fuel tanker	NULL	ref	Fuel_tanker_id1_idx	Fuel_tanker_id1_idx	4	fueldb.fuel tanker-fuel.Fuel_tanker_id	15	100.00	Using index
2	DERIVED	fuel tanker-gas station	NULL	ref	Fuel_tanker-Fuel_id_idx	Fuel_tanker-Fuel_id_idx	4	fueldb.fuel tanker-fuel.Fuel_tanker-Fuel_id	61	100.00	Using index

Рис. 44. EXPLAIN первого варианта запроса 8

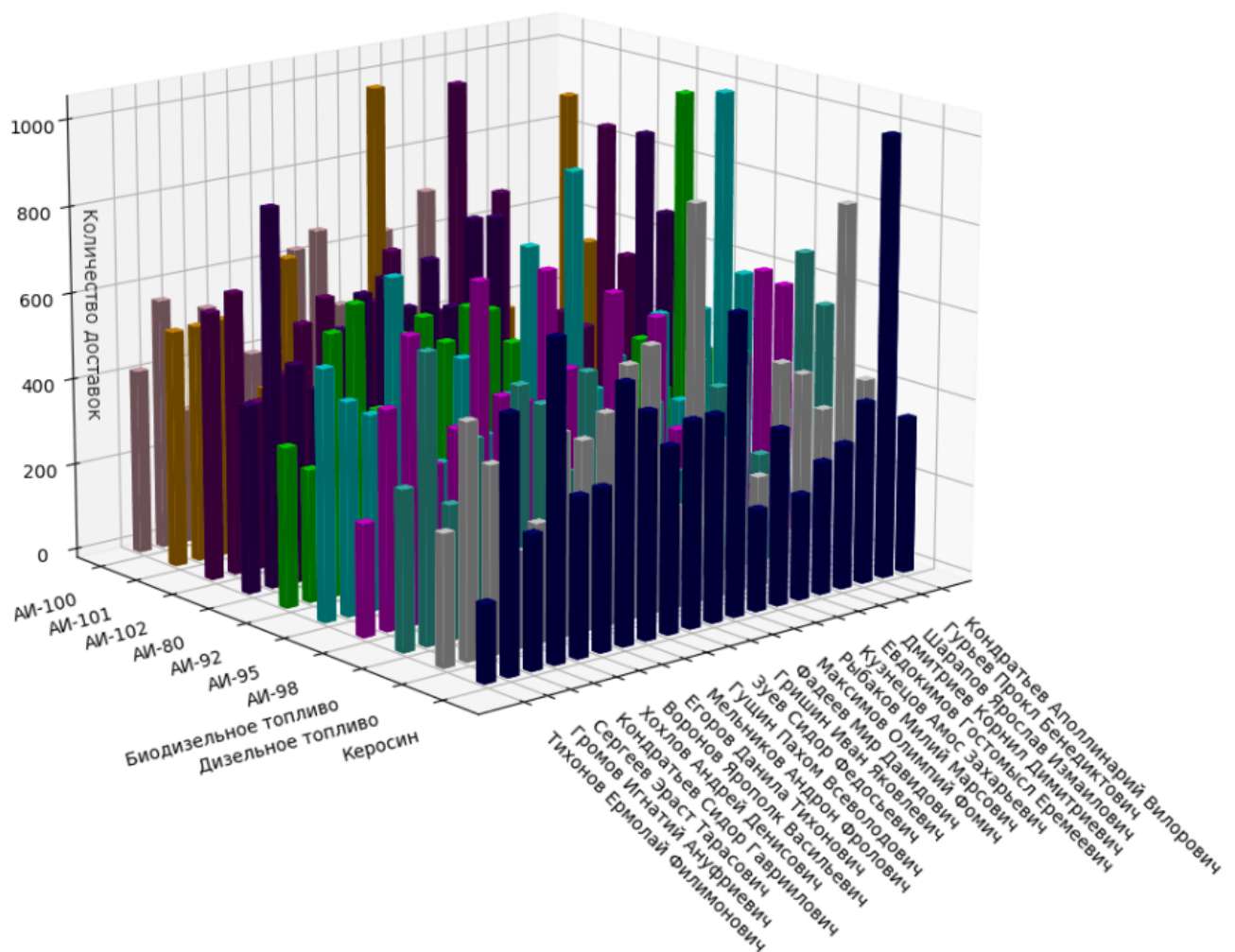


Рис. 45. Гистограмма результата первого варианта запроса 8

## Объяснение запроса

Сначала выполняем подзапрос: считаем для каждого водителя доставку каждого вида топлива, делая группировку по id водителя и типу топлива. Далее делаем join к таблице водителей и получаем ФИО водителей.

## Второй вариант

### Код запроса

```
1 select
2     driver.Surname,
3     driver.`Name`,
```

```

4         driver.Middle_name,
5         deliveriesNum.`Type`,
6         COALESCE (deliveriesNum.`Количество доставок`, 0) as 'Количество доставок'
7 from
8     driver
9 cross join fuel
10 left join (
11     select
12         `driver-fuel tanker`.Driver_id,
13         fuel.`Type`,
14         count(*) as `Количество доставок`
15     from
16         `fuel tanker-gas station`
17     inner join `fuel tanker-fuel`
18         on `fuel tanker-gas station`.`Fuel_tanker-Fuel_id`
19           = `fuel tanker-fuel`.`Fuel_tanker-Fuel_id`
20     inner join fuel
21         on fuel.Fuel_id=`fuel tanker-fuel`.Fuel_id
22     inner join `fuel tanker`
23         on `fuel tanker`.Fuel_tanker_id=`fuel tanker-fuel`.Fuel_tanker_id
24     inner join `driver-fuel tanker`
25         on `fuel tanker`.Fuel_tanker_id=`driver-fuel tanker`.Fuel_tanker_id
26     group by Driver_id, `Type`
27 ) as deliveriesNum
28     on deliveriesNum.Driver_id=driver.driver_id and
    ↪ deliveriesNum.`Type`=fuel.`Type`

```

## Результат запроса

Запрос выполнен за 7500 миллисекунд. В результирующей таблице получилось 5000 строк.

	Surname	Name	Middle_name	Type	Количество доставок
►	Кондратьев	Аполлинарий	Вилорович	Керосин	472
	Кондратьев	Аполлинарий	Вилорович	Биодизельное топливо	128
	Кондратьев	Аполлинарий	Вилорович	Дизельное топливо	234
	Кондратьев	Аполлинарий	Вилорович	АИ-102	510
	Кондратьев	Аполлинарий	Вилорович	АИ-101	190
	Кондратьев	Аполлинарий	Вилорович	АИ-100	398
	Кондратьев	Аполлинарий	Вилорович	АИ-98	174
	Кондратьев	Аполлинарий	Вилорович	АИ-95	358
	Кондратьев	Аполлинарий	Вилорович	АИ-92	493

Рис. 46. Часть результата второго варианта запроса 8

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
	1	PRIMARY	fuel	HULL	ALL	HULL	HULL	HULL	HULL	10	100.00	HULL
	1	PRIMARY	driver	HULL	ALL	HULL	HULL	HULL	HULL	500	100.00	Using join buffer (hash join)
	1	PRIMARY	<derived2>	HULL	ALL	<auto_key0>	<auto_key0>	66	fueldb.driver.driver_id,fueldb.fuel.Type	22480	100.00	HULL
	2	DERIVED	fuel	HULL	ALL	PRIMARY	HULL	HULL	HULL	10	100.00	Using temporary
	2	DERIVED	fuel tanker-fuel	HULL	ref	PRIMARY_Fuel_tanker_id2_idx,Fuel_id_idx	Fuel_id_idx	4	fueldb.fuel.Fuel_id	238	100.00	HULL
	2	DERIVED	fuel tanker	HULL	eq_ref	PRIMARY	PRIMARY	4	fueldb.fuel tanker-fuel.Fuel_tanker_id	1	100.00	Using index
	2	DERIVED	driver-fuel tanker	HULL	ref	fuel_tanker_id1_idx	fuel_tanker_id1_idx	4	fueldb.fuel tanker-fuel.Fuel_tanker_id	15	100.00	Using index
	2	DERIVED	fuel tanker-gas station	HULL	ref	Fuel_tanker-Fuel_id_idx	Fuel_tanker-Fuel_id_idx	4	fueldb.fuel tanker-fuel.Fuel_tanker-Fuel_id	61	100.00	Using index

Рис. 47. EXPLAIN второго варианта запроса 8

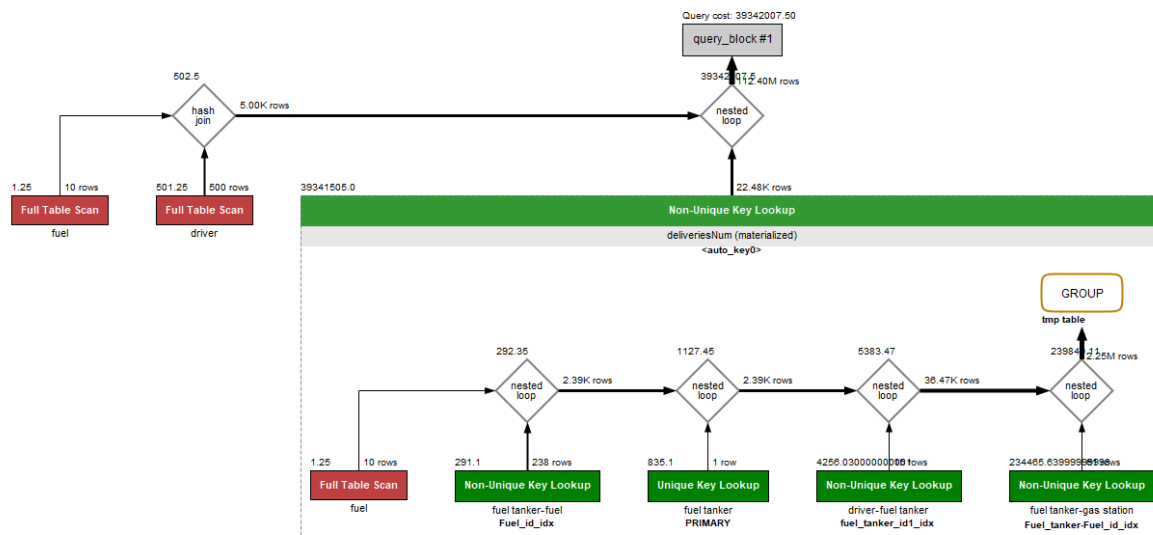


Рис. 48. Execution Plan второго варианта запроса 8

## Объяснение запроса

Сначала с помощью группировки получаем все ненулевые доставки водителями каждого типа топлива. Далее берем всех водителей, через cross join делаем к ним колонки с каждым видом топлива и к этой таблице через left join подключаем первую таблицу и выводим количество доставок каждым водилетом каждого вида топлива. Если 'Количество доставок' = null, то заменяем его на 0.

## Третий вариант

### Код запроса

```

1 select
2     driver.Surname,
3     driver.`Name`,
4     driver.Middle_name,
5     fuel.`Type`,

```

```

6      COALESCE((
7          select
8              count(*)
9          from
10             `fuel tanker-gas station`
11         inner join `fuel tanker-fuel`
12             on `fuel tanker-gas station`.`Fuel_tanker-Fuel_id`=`fuel
13                ↳ tanker-fuel`.`Fuel_tanker-Fuel_id`
14         inner join fuel as f
15             on fuel.Fuel_id=`fuel tanker-fuel`.Fuel_id
16         inner join `fuel tanker`
17             on `fuel tanker`.Fuel_tanker_id=`fuel
18                ↳ tanker-fuel`.Fuel_tanker_id
19         inner join `driver-fuel tanker`
20             on `fuel tanker`.Fuel_tanker_id=`driver-fuel
21                ↳ tanker`.Fuel_tanker_id
22         where `driver-fuel tanker`.Driver_id = driver.driver_id and f.`Type`
23            ↳ = fuel.`Type`
24         group by `driver-fuel tanker`.Driver_id, f.`Type`
25     ), 0) as 'Количество доставок'
26 from
27     driver
28 cross join fuel

```

## Результат запроса

Запрос выполнен за 3547 миллисекунд. В результирующей таблице получилось 5000 строк.

	Surname	Name	Middle_name	Type	Количество доста вок
►	Кондратьев	Аполлинарий	Вилорович	Керосин	472
	Кондратьев	Аполлинарий	Вилорович	Биодизельное топливо	128
	Кондратьев	Аполлинарий	Вилорович	Дизельное топливо	234
	Кондратьев	Аполлинарий	Вилорович	АИ-102	510
	Кондратьев	Аполлинарий	Вилорович	АИ-101	190
	Кондратьев	Аполлинарий	Вилорович	АИ-100	398
	Кондратьев	Аполлинарий	Вилорович	АИ-98	174
	Кондратьев	Аполлинарий	Вилорович	АИ-95	358
	Кондратьев	Аполлинарий	Вилорович	АИ-92	493

Рис. 49. Часть результата третьего варианта запроса 8

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	fuel	10000	ALL	PRIMARY	PRIMARY	4		10	100.00	Using index
1	PRIMARY	driver	10000	ALL	PRIMARY	PRIMARY	4		500	100.00	Using join buffer (hash join)
2	DEPEND...	f	10000	ALL	PRIMARY	PRIMARY	4		10	10.00	Using where; Using temporary
2	DEPEND...	driver-fuel tanker	10000	ref	driver_id1_idx,fuel_tanker_id1_idx	driver_id1_idx	4	fueldb.driver.driver_id	16	100.00	Using index
2	DEPEND...	fuel tanker-fuel	10000	ref	PRIMARY,Fuel_tanker_id2_idx,Fuel_id_idx	Fuel_tanker_id2_idx	4	fueldb.driver-fuel tanker.Fuel_tanker_id	4	10.00	Using where
2	DEPEND...	fuel tanker	10000	eq_ref	PRIMARY	PRIMARY	4	fueldb.driver-fuel tanker.Fuel_tanker_id	1	100.00	Using index
2	DEPEND...	fuel tanker-gas station	10000	ref	Fuel_tanker-Fuel_id_idx	Fuel_tanker-Fuel_id_idx	4	fueldb.fuel tanker-fuel.Fuel_tanker-Fuel_id	61	100.00	Using index

Рис. 50. EXPLAIN третьего варианта запроса 8

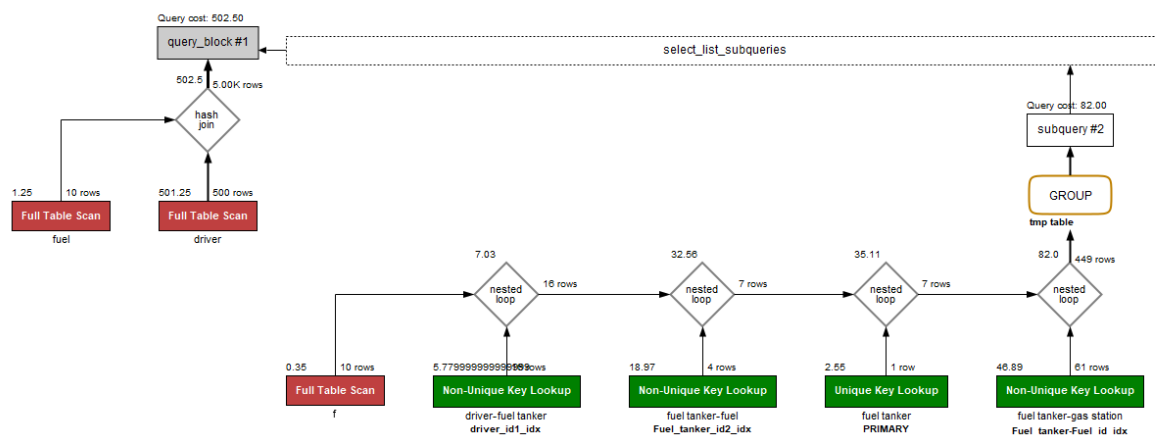


Рис. 51. Execution Plan третьего варианта запроса 8

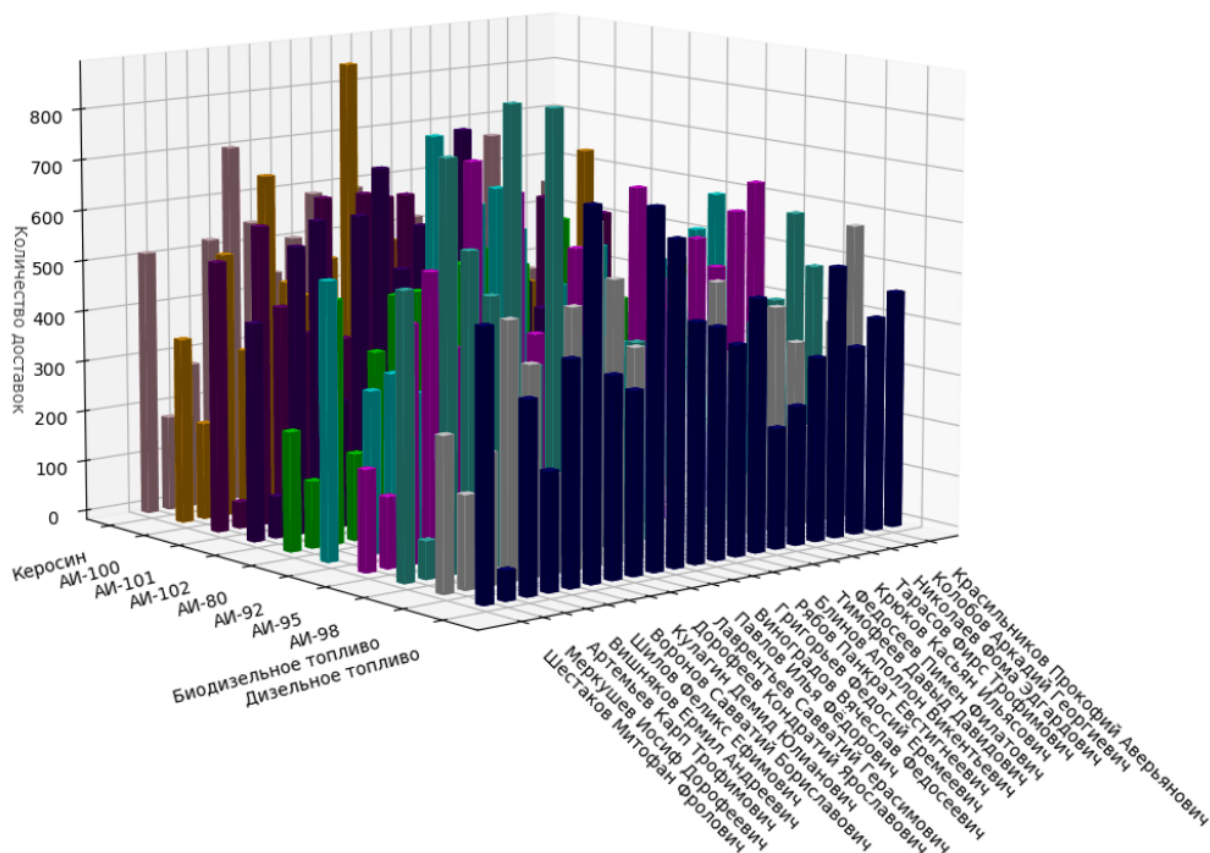


Рис. 52. Гистограмма результата восьмого запроса

## Объяснение запроса

Сначала берем всех водителей, через `cross join` создаем для каждого водителя колонки с каждым видом топлива. Далее в подзапросе будет вызван `count` для каждой комбинации водителя и типа топлива из внешнего `select`.

## Заключение

На первом этапе выполнения курсовой работы была выполнена аналитическая часть: выбрана и проанализирована предметная область «Сеть частных автозаправочных станций». Была построена ER-диаграмма, содержащая 11 сущностей и 26 атрибутов.

На втором этапе были выделены цели функционирования системы, описаны роли, субъекты и сущности, создана схема объектов с указанием типов связи. На основе схемы была создана база данных, состоящая из 11 таблиц, связанных через внешние ключи. Для каждой таблицы описаны атрибуты: тип ключа, название на русском и английском языках, тип атрибута, ссылка на атрибут другой таблицы, если атрибут является внешним ключом. Был написан скрипт на языке Python для заполнения базы данными. Всего в базе получилось 226565 записей.

На третьем этапе к полученной базе данных были написаны 8 запросов для получения данных. Все запросы были написаны на языке SQL, после чего проанализированы с помощью команды EXPLAIN и Visual EXPLAIN в MySQL Workbench.



# Приложение А. Код создания базы данных

```
1 CREATE SCHEMA IF NOT EXISTS `fueldb` DEFAULT CHARACTER SET utf8 ;
2 USE `fueldb` ;
3
4 -----
5 -- Table `fueldb`.`Driver`
6 -----
7 CREATE TABLE IF NOT EXISTS `fueldb`.`Driver` (
8     `driver_id` INT NOT NULL AUTO_INCREMENT,
9     `Surname` VARCHAR(50) NOT NULL,
10    `Name` VARCHAR(50) NOT NULL,
11    `Middle_name` VARCHAR(50) NOT NULL,
12    `Phone_number` VARCHAR(20) NOT NULL,
13    PRIMARY KEY (`driver_id`))
14 ENGINE = InnoDB;
15
16 -----
17 -- Table `fueldb`.`Fuel tanker`
18 -----
19
20 CREATE TABLE IF NOT EXISTS `fueldb`.`Fuel tanker` (
21     `Fuel_tanker_id` INT NOT NULL AUTO_INCREMENT,
22     `Model` VARCHAR(45) NOT NULL,
23     PRIMARY KEY (`Fuel_tanker_id`))
24 ENGINE = InnoDB;
25
26 -----
27 -- Table `fueldb`.`Fuel`
28 -----
29
30 CREATE TABLE IF NOT EXISTS `fueldb`.`Fuel` (
31     `Fuel_id` INT NOT NULL AUTO_INCREMENT,
32     `Type` VARCHAR(20) NOT NULL,
33     PRIMARY KEY (`Fuel_id`))
34 ENGINE = InnoDB;
35
36 -----
37 -- Table `fueldb`.`Gas station`
38 -----
39
40 CREATE TABLE IF NOT EXISTS `fueldb`.`Gas station` (
41     `Gas_station_id` INT NOT NULL AUTO_INCREMENT,
42     `Address` VARCHAR(200) NOT NULL,
43     PRIMARY KEY (`Gas_station_id`))
44 ENGINE = InnoDB;
```

```

45
46
47 -----
48 -- Table `fueldb`.`Fuel container`
49 -----
50 CREATE TABLE IF NOT EXISTS `fueldb`.`Fuel container` (
51     `Fuel_container_id` INT NOT NULL AUTO_INCREMENT,
52     `Capacity` DECIMAL(7,2) NOT NULL,
53     `Fuel_amount` DECIMAL(7,2) NOT NULL,
54     `Gas_station_id` INT NOT NULL,
55     `Fuel_id` INT NOT NULL,
56     PRIMARY KEY (`Fuel_container_id`),
57     INDEX `Gas_station_id2_idx` (`Gas_station_id` ASC) VISIBLE,
58     INDEX `Fuel_id2_idx` (`Fuel_id` ASC) VISIBLE,
59     CONSTRAINT `Gas_station_id2`
60         FOREIGN KEY (`Gas_station_id`)
61         REFERENCES `fueldb`.`Gas station` (`Gas_station_id`)
62         ON DELETE CASCADE
63         ON UPDATE CASCADE,
64     CONSTRAINT `Fuel_id2`
65         FOREIGN KEY (`Fuel_id`)
66         REFERENCES `fueldb`.`Fuel` (`Fuel_id`)
67         ON DELETE CASCADE
68         ON UPDATE CASCADE)
69 ENGINE = InnoDB;
70
71
72 -----
73 -- Table `fueldb`.`Technical Specialist`
74 -----
75 CREATE TABLE IF NOT EXISTS `fueldb`.`Technical Specialist` (
76     `Technical_Specialist_id` INT NOT NULL AUTO_INCREMENT,
77     `Surname` VARCHAR(50) NOT NULL,
78     `Name` VARCHAR(50) NOT NULL,
79     `Middle_name` VARCHAR(50) NOT NULL,
80     `Phone_number` VARCHAR(20) NOT NULL,
81     PRIMARY KEY (`Technical_Specialist_id`))
82 ENGINE = InnoDB;
83
84
85 -----
86 -- Table `fueldb`.`Driver-Fuel tanker`
87 -----
88 CREATE TABLE IF NOT EXISTS `fueldb`.`Driver-Fuel tanker` (
89     `Driver-Fuel_tanker_id` INT NOT NULL AUTO_INCREMENT,
90     `Driver_id` INT NOT NULL,

```

```

91         `Fuel_tanker_id` INT NOT NULL,
92     PRIMARY KEY (`Driver-Fuel_tanker_id`, `Driver_id`),
93     INDEX `driver_id1_idx` (`Driver_id` ASC) VISIBLE,
94     INDEX `fuel_tanker_id1_idx` (`Fuel_tanker_id` ASC) VISIBLE,
95     CONSTRAINT `driver_id1`
96         FOREIGN KEY (`Driver_id`)
97         REFERENCES `fueldb`.`Driver` (`driver_id`)
98         ON DELETE CASCADE
99         ON UPDATE CASCADE,
100    CONSTRAINT `fuel_tanker_id1`
101        FOREIGN KEY (`Fuel_tanker_id`)
102        REFERENCES `fueldb`.`Fuel tanker` (`Fuel_tanker_id`)
103        ON DELETE CASCADE
104        ON UPDATE CASCADE)
105    ENGINE = InnoDB;
106
107
108    -----
109    -- Table `fueldb`.`Fuel tanker-Fuel`
110    -----
111    CREATE TABLE IF NOT EXISTS `fueldb`.`Fuel tanker-Fuel` (
112        `Fuel_tanker-Fuel_id` INT NOT NULL AUTO_INCREMENT,
113        `Fuel_id` INT NOT NULL,
114        `Fuel_tanker_id` INT NOT NULL,
115        `Capacity` DECIMAL(7,2) NOT NULL,
116        PRIMARY KEY (`Fuel_tanker-Fuel_id`),
117        INDEX `Fuel_tanker_id2_idx` (`Fuel_tanker_id` ASC) VISIBLE,
118        INDEX `Fuel_id_idx` (`Fuel_id` ASC) VISIBLE,
119        CONSTRAINT `Fuel_tanker_id2`
120            FOREIGN KEY (`Fuel_tanker_id`)
121            REFERENCES `fueldb`.`Fuel tanker` (`Fuel_tanker_id`)
122            ON DELETE CASCADE
123            ON UPDATE CASCADE,
124        CONSTRAINT `Fuel_id`
125            FOREIGN KEY (`Fuel_id`)
126            REFERENCES `fueldb`.`Fuel` (`Fuel_id`)
127            ON DELETE CASCADE
128            ON UPDATE CASCADE)
129    ENGINE = InnoDB;
130
131
132    -----
133    -- Table `fueldb`.`Technical Specialist-Fuel tanker`
134    -----
135    CREATE TABLE IF NOT EXISTS `fueldb`.`Technical Specialist-Fuel tanker` (
136        `Technical_Specialist-Fuel_tanker_id` INT NOT NULL AUTO_INCREMENT,

```

```

137     `Technical_Specialist_id` INT NOT NULL,
138     `Fuel_tanker_id` INT NOT NULL,
139     PRIMARY KEY (`Technical_Specialist-Fuel_tanker_id`),
140     INDEX `Fuel_tanker_id2_idx` (`Fuel_tanker_id` ASC) VISIBLE,
141     INDEX `Technical_Specialist_id_idx` (`Technical_Specialist_id` ASC) VISIBLE,
142     CONSTRAINT `Fuel_tanker_id3`
143         FOREIGN KEY (`Fuel_tanker_id`)
144         REFERENCES `fueldb`.`Fuel tanker` (`Fuel_tanker_id`)
145         ON DELETE CASCADE
146         ON UPDATE CASCADE,
147     CONSTRAINT `Technical_Specialist_id`
148         FOREIGN KEY (`Technical_Specialist_id`)
149         REFERENCES `fueldb`.`Technical Specialist`
150         ↪ (`Technical_Specialist_id`)
151         ON DELETE CASCADE
152         ON UPDATE CASCADE)
153
154 ENGINE = InnoDB;
155
156
157 -----
158 -- Table `fueldb`.`Technical Specialist-Fuel containers`
159 -----
160 CREATE TABLE IF NOT EXISTS `fueldb`.`Technical Specialist-Fuel containers` (
161     `Technical_Specialist-Fuel_containers_id` INT NOT NULL AUTO_INCREMENT,
162     `Technical_Specialist_id` INT NOT NULL,
163     `Fuel_containers_id` INT NOT NULL,
164     PRIMARY KEY (`Technical_Specialist-Fuel_containers_id`),
165     INDEX `Technical_Specialist_id2_idx` (`Technical_Specialist_id` ASC) VISIBLE,
166     INDEX `Fuel_containers_id_idx` (`Fuel_containers_id` ASC) VISIBLE,
167     CONSTRAINT `Technical_Specialist_id2`
168         FOREIGN KEY (`Technical_Specialist_id`)
169         REFERENCES `fueldb`.`Technical Specialist`
170         ↪ (`Technical_Specialist_id`)
171         ON DELETE CASCADE
172         ON UPDATE CASCADE,
173     CONSTRAINT `Fuel_containers_id`
174         FOREIGN KEY (`Fuel_containers_id`)
175         REFERENCES `fueldb`.`Fuel container` (`Fuel_container_id`)
176         ON DELETE CASCADE
177         ON UPDATE CASCADE)
178 ENGINE = InnoDB;
179
180
181 -----
182 -- Table `fueldb`.`Fuel tanker-Gas station`
183 -----

```

```

181 CREATE TABLE IF NOT EXISTS `fueldb`.`Fuel tanker-Gas station` (
182     `Fuel_tanker-Gas_station_id` INT NOT NULL AUTO_INCREMENT,
183     `Fuel_tanker-Fuel_id` INT NOT NULL,
184     `Gas_station_id` INT NOT NULL,
185     `Visit_date` DATE NOT NULL,
186     `Quantity` DECIMAL(7,2) NOT NULL,
187     PRIMARY KEY (`Fuel_tanker-Gas_station_id`),
188     INDEX `Fuel_tanker-Fuel_id_idx` (`Fuel_tanker-Fuel_id` ASC) VISIBLE,
189     INDEX `Gas_station_id_idx` (`Gas_station_id` ASC) VISIBLE,
190     CONSTRAINT `Fuel_tanker-Fuel_id`
191         FOREIGN KEY (`Fuel_tanker-Fuel_id`)
192         REFERENCES `fueldb`.`Fuel tanker-Fuel` (`Fuel_tanker-Fuel_id`)
193         ON DELETE CASCADE
194         ON UPDATE CASCADE,
195     CONSTRAINT `Gas_station_id`
196         FOREIGN KEY (`Gas_station_id`)
197         REFERENCES `fueldb`.`Gas station` (`Gas_station_id`)
198         ON DELETE CASCADE
199         ON UPDATE CASCADE)
200 ENGINE = InnoDB;

```

## Приложение В. Код заполнения базы данных

```
1 from faker import Faker
2 import random
3 import mysql.connector
4 import openpyxl
5 from mysql.connector import Error
6 from config import password
7
8
9 def export_to_sql():
10     fake = Faker('ru_RU')
11     try: connection = mysql.connector.connect(
12         host="localhost",
13         user="root",
14         password=password,
15         database="fueldb",
16     )
17
18     rows_count = 0
19     cursor = connection.cursor()
20
21     query = "DELETE FROM `Driver-Fuel tanker`"
22     cursor.execute(query)
23     connection.commit()
24     print(f"Все строки таблицы `Driver-Fuel tanker` были удалены.")
25
26     query = "DELETE FROM `Fuel tanker-Fuel`"
27     cursor.execute(query)
28     connection.commit()
29     print(f"Все строки таблицы `Fuel tanker-Fuel` были удалены.")
30
31     query = "DELETE FROM `Technical Specialist-Fuel tanker`"
32     cursor.execute(query)
33     connection.commit()
34     print(f"Все строки таблицы `Technical Specialist-Fuel tanker` были удалены.")
35
36     query = "DELETE FROM `Fuel container`"
37     cursor.execute(query)
38     connection.commit()
39     print(f"Все строки таблицы `Fuel container` были удалены.")
40
41     query = "DELETE FROM `Fuel tanker-Gas station`"
42     cursor.execute(query)
43     connection.commit()
44     print(f"Все строки таблицы `Fuel tanker-Gas station` были удалены.")
```

```

45
46 query = "DELETE FROM `Technical Specialist-Fuel containers`"
47 cursor.execute(query)
48 connection.commit()
49 print(f"Все строки таблицы `Technical Specialist-Fuel containers` были
↳ удалены.")
50
51 # Таблиц уровня 1
52 # заполнение таблицы Driver
53 with connection.cursor() as cursor:
54     for i in range(1, 501):
55         cursor.execute("REPLACE INTO Driver (driver_id, Surname,
↳ Name,"
56             "Middle_name, Phone_number) VALUES (%s, %s, %s, %s, %s)",
57             (i, fake.last_name_male(), fake.first_name_male(),
58             fake.middle_name_male(), fake.phone_number()))
59         connection.commit()
60         rows_count += 1
61
62 # заполнение таблицы Fuel tanker
63 with connection.cursor() as cursor:
64     file_to_read = openpyxl.load_workbook('data.xlsx', data_only=True)
65     sheet = file_to_read['Лист2']
66     for row in range(1, sheet.max_row + 1):
67         value = sheet.cell(row, 1).value
68         cursor.execute("REPLACE INTO `Fuel tanker` (Fuel_tanker_id,
↳ Model)"
69             "VALUES (%s, %s)", (row, value))
70         connection.commit()
71         rows_count += 1
72
73 # заполнение таблицы Gas station
74 with connection.cursor() as cursor:
75     for i in range(1, 2001):
76         cursor.execute("REPLACE INTO `Gas station` (Gas_station_id,
↳ Address) "
77             "VALUES (%s, %s)", (i, fake.address()))
78         connection.commit()
79         rows_count += 1
80
81 # заполнение Fuel
82 fuel_types = ['АИ-80', 'АИ-92', 'АИ-95', 'АИ-98', 'АИ-100', 'АИ-101',
83 'АИ-102', 'Дизельное топливо', 'Биодизельное топливо', 'Керосин']
84 with connection.cursor() as cursor:
85     cursor.executemany("REPLACE INTO Fuel (Fuel_id, Type) VALUES (%s,
↳ %s)",

```

```

86         [(id_num, fuel_type) for id_num, fuel_type in enumerate(fuel_types,
87             ↪ start=1)])
88         connection.commit(
89
90     # заполнение таблицы Technical Specialist
91     with connection.cursor() as cursor:
92         for i in range(1, 1001):
93             cursor.execute("REPLACE INTO `Technical Specialist` "
94                 ↪ "(Technical_Specialist_id, Surname, Name, Middle_name,
95                 ↪ Phone_number) "
96                 ↪ "VALUES (%s, %s, %s, %s, %s)", (i, fake.last_name_male(),
97                 ↪ fake.first_name_male(), fake.middle_name_male(),
98                 ↪ fake.phone_number()))
99             connection.commit()
100             rows_count += 1
101
102     # Таблиц уровня 2
103     # заполняем Driver-Fuel tanker
104     with connection.cursor() as cursor:
105         row_id = 1
106         for Fuel_tanker_id in range(1, 541): # для каждого бензовоза
107             driver_count = random.randint(10, 20) # получаем число
108             ↪ водителей
109             driver_ids = list(range(1, 501)) # заполняем список
110             ↪ айдишников водителей
111             for i in range(driver_count): # цикл по количеству водителей
112                 driver_id = driver_ids[random.randint(0,
113                     ↪ len(driver_ids)-1)]
114                 driver_ids.remove(driver_id)
115                 cursor.execute("REPLACE INTO `Driver-Fuel tanker` "
116                     ↪ "(`Driver-Fuel_tanker_id`, Driver_id, "
117                     ↪ "Fuel_tanker_id) VALUES (%s, %s, %s)", (row_id,
118                     ↪ driver_id,
119                     ↪ Fuel_tanker_id))
120                 row_id += 1
121                 rows_count += 1
122                 connection.commit()
123
124     # заполняем Fuel tanker-Fuel
125     with connection.cursor() as cursor:
126         row_id = 1
127         for Fuel_tanker_id in range(1, 541):
128             fuel_count = random.randint(2, 7)
129             fuel_ids = list(range(1, 11))
130             for i in range(fuel_count):

```



```

125         fuel_id = fuel_ids[random.randint(0, len(fuel_ids) -
126         ↪ 1)]
127     fuel_ids.remove(fuel_id)
128     cursor.execute("REPLACE INTO `Fuel tanker-Fuel` "
129     "(`Fuel_tanker-Fuel_id`, Fuel_id, "
130     "Fuel_tanker_id, Capacity) VALUES (%s, %s, %s, %s)",
131     ↪ (row_id, fuel_id, Fuel_tanker_id,
132     ↪ random.uniform(1200, 8000)))
133     row_id += 1
134     rows_count += 1
135     connection.commit()
136
137 # заполняем Technical Specialist-Fuel tanker
138 with connection.cursor() as cursor:
139     row_id = 1
140     for Technical_Specialist_id in range(1, 1001):
141         fuel_tanker_count = random.randint(1, 12)
142         fuel_tanker_ids = list(range(1, 541))
143         for i in range(fuel_tanker_count):
144             fuel_tanker_id = fuel_tanker_ids[random.randint(0,
145             ↪ len(fuel_tanker_ids)
146             ↪ - 1)]
147             fuel_tanker_ids.remove(fuel_tanker_id)
148             cursor.execute("REPLACE INTO `Technical
149             ↪ Specialist-Fuel tanker` "
150             "(`Technical_Specialist-Fuel_tanker_id`, "
151             "Technical_Specialist_id, Fuel_tanker_id) "
152             "VALUES (%s, %s, %s)", (row_id,
153             ↪ Technical_Specialist_id,
154             ↪ fuel_tanker_id))
155             row_id += 1
156             rows_count += 1
157         connection.commit()
158
159 # заполняем Fuel container
160 with connection.cursor() as cursor:
161     row_id = 1
162     for Gas_station_id in range(1, 2001):
163         fuel_containers_count = random.randint(20, 30)
164         for i in range(fuel_containers_count):
165             capacity = random.uniform(20000, 40000)
166             fuel_amount = random.uniform(0, capacity)
167             cursor.execute(
168             "REPLACE INTO `Fuel container` (`Fuel_container_id`,
169             ↪ Capacity, "
170             "Fuel_amount, Gas_station_id, Fuel_id) VALUES (%s,
171             ↪ %s, %s, %s, %s)",

```

```

165         (row_id, capacity, fuel_amount,
166         Gas_station_id, random.randint(1, 10)))
167         row_id += 1
168         rows_count += 1
169     connection.commit()
170
171     # Таблицы уровня 3
172     # заполняем Fuel tanker-Gas station
173     with connection.cursor() as cursor:
174         row_id = 1
175
176         cursor.execute('SELECT `Fuel_tanker-Fuel_id` FROM `Fuel
177         ↳ tanker-Fuel`')
178         delivery_values_list = [item[0] for item in cursor.fetchall()]
179
180         for Gas_station_id in range(1, 2001):
181             delivery_count = random.randint(50, 100)
182             delivery_ids = delivery_values_list.copy()
183             for i in range(delivery_count):
184                 delivery_id = delivery_ids[random.randint(0,
185                 len(delivery_ids) - 1)]
186                 delivery_ids.remove(delivery_id)
187                 cursor.execute("REPLACE INTO `Fuel tanker-Gas
188                 ↳ station` "
189                 "(`Fuel_tanker-Gas_station_id`, "
190                 "`Fuel_tanker-Fuel_id`, Gas_station_id, Visit_date,
191                 ↳ Quantity) "
192                 "VALUES (%s, %s, %s, %s, %s)",
193                 (row_id, delivery_id, Gas_station_id,
194                 fake.date_between('-2y', 'today'),
195                 ↳ random.uniform(400, 3000)))
196                 row_id += 1
197                 rows_count += 1
198             connection.commit()
199
200     # заполняем Technical Specialist-Fuel containers
201     with connection.cursor() as cursor:
202         row_id = 1
203
204         cursor.execute('SELECT `Fuel_tanker-Fuel_id` FROM `Fuel
205         ↳ tanker-Fuel`')
206         delivery_values_list = [item[0] for item in cursor.fetchall()]
207
208         for Technical_Specialist_id in range(1, 1001):
209             fuel_containers_count = random.randint(1, 10)
210             fuel_containers_ids = delivery_values_list.copy()

```

```

206         for i in range(fuel_containers_count):
207             fuel_containers_id =
208                 ↪ fuel_containers_ids[random.randint(0,
209                     len(fuel_containers_ids) - 1)]
210             fuel_containers_ids.remove(fuel_containers_id)
211             cursor.execute(
212                 "REPLACE INTO `Technical Specialist-Fuel containers`"
213                 "(`Technical_Specialist-Fuel_containers_id`, "
214                 "Technical_Specialist_id, Fuel_containers_id)"
215                 " VALUES (%s, %s, %s)",
216                 (row_id, Technical_Specialist_id,
217                 ↪ fuel_containers_id))
218             row_id += 1
219             rows_count += 1
220             connection.commit()
221
222             print("Количество записей в БД: " + str(rows_count))
223
224     except Error as e:
225         print(e)

```

---

```

225 export_to_sql()

```