Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Волгоградский государственный технический университет»

Факультет Электроники и вычислительной техники\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Направление (специальность) \_Информатика и вычислительная техника\_\_\_\_  
Кафедра \_\_\_\_\_\_\_\_\_САПР и ПК\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дисциплина \_\_\_\_\_\_\_Базы данных\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  | Утверждаю  Зав. кафедрой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | «\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20 \_\_\_ г. |

**ЗАДАНИЕ**

**на курсовую работу (проект)**

Студент\_\_\_\_Иванова Валерия Александровна\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

Группа\_\_\_ИВТ-263\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Тема: \_Проектирование и разработка базы данных сети магазинов и автосервисов и приложения для взаимодействия с ней

Утверждена приказом от «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г. № \_\_\_\_\_\_\_\_\_

2. Срок представления работы (проекта) к защите «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ г.

3. Содержание расчетно-пояснительной записки: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Перечень графического материала: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Дата выдачи задания «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20 \_\_\_ г.

Руководитель работы (проекта)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_Соколов А.А.\_\_\_

подпись, дата инициалы и фамилия

Задание принял к исполнению  Иванова В.А.

подпись, дата инициалы и фамилия

Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Волгоградский государственный технический университет»

Факультет Электроники и вычислительной техники\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_САПР и ПК\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе (проекту)**

по дисциплине \_\_\_\_\_\_\_Базы данных\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

на тему Проектирование и разработка базы данных сети магазинов и автосервисов и приложения для взаимодействия с ней

Студент\_\_\_ Иванова В.А.

(фамилия, имя, отчество)

Группа\_\_\_ИВТ-263\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель работы (проекта) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Соколов А.А.\_

(подпись и дата подписания) (инициалы и фамилия)

Члены комиссии:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и дата подписания) (инициалы и фамилия)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и дата подписания) (инициалы и фамилия)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и дата подписания) (инициалы и фамилия)

Нормоконтролер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата подписания) (инициалы и фамилия)

Волгоград 2021 г.

Содержание

[Концептуальное проектирование базы данных 4](#_Toc74524237)

[1. Предметная область 4](#_Toc74524238)

[3. ER-диаграмма 11](#_Toc74524239)

[Логическое проектирование БД 12](#_Toc74524240)

[1. Реляционная модель: 12](#_Toc74524241)

[2. Набор данных 12](#_Toc74524242)

[Физическое проектирование БД 16](#_Toc74524243)

[1. Реляционная схема в MySQL 16](#_Toc74524244)

[2. Реляционная схема (из второй лабораторной работы) 17](#_Toc74524245)

[3. Таблицы 17](#_Toc74524247)

[Реализация SQL-запросов базы данных 23](#_Toc74524248)

[Индексы, хранимые процедуры, представления 39](#_Toc74524249)

[Взаимодействие с БД из приложения 59](#_Toc74524250)

[Ссылка на репозиторий: 69](#_Toc74524251)

[Список литературы 69](#_Toc74524252)

# Концептуальное проектирование базы данных

# Предметная область

Создание базы данных магазинашин и автосервиса. БД будут использовать филиалы магазинов в городе. Данная БД поможет упростить работу за счет оптимизации хранения информации о товаре и времени записи в автосервис.

1. **Описание предметной области**

Магазин шин – это организация, занимающаяся продажей шин, дисков, а также провести диагностику автомобиля на автосервисе, и, если будет найдена проблема – ее устранить.

Процесс покупки можно разделить на следующий этапы:

1. Консультация клиента с сотрудником. На данном этапе определяются требования и пожелание клиента.

2. Выбор товара. На основании требований и пожеланий клиента сотрудник подбирает подходящий товар.

3. Оформление заказа. После определения товара между магазином и клиентом заключается договор, в котором указываются данные о двух сторонах, вид и стоимость товара.

Процесс записи на диагностику с последующим сопровождением:

1. Консультация с клиентом. Менеджер должен узнать клиент хочет пройти диагностику или записаться на определенную услугу автосервиса.
2. Выбор времени. Менеджер должен узнать нужное время и дату, подходящую для клиента, после чего посмотреть наличие свободной записи с учетом предпочтений клиента.
3. Заключение договора о предоставленной услуге. В день оказания услуги заключается договор, в котором указывается данные двух сторон, стоимость и вид услуги.
4. Функциональные и нефункциональные требования

##### Транзакционные (задачи учёта):

1. Изменение цены товара
2. Запись на сервис
3. Оформление заказа
4. Добавить новые товары
5. Убрать неактуальные товары

**Справочные (оперативные запросы):**

1. Предоставление информации о количестве имеющихся единиц товаров в определенном магазине
2. Предоставление информации в каком магазине товар в наличии
3. Предоставление информации о конкретном товаре (описание, страна производителя, марка)
4. Показать свободное время в автосервисе
5. Показать отзывы о товаре

**Справочные расчётные (аналитические запросы):**

1. Показать самые популярные товары, которые чаще всего покупались
2. Показать данные о продажах за определенный период(какого товара и в каком количестве продали)
3. Предоставить общие доходы за определенный период
4. Показать товары с высоким рейтингом
5. Показать товары с низким отзывом

**Плановые (расчётные задачи):**

1. Составление плана закупок - сколько и какого товара надо закупить на след. месяц
2. Составление расписания для записи в автосервис
3. Расчет дохода за месяц
4. Расчет сколько товаров было куплено за определенный период
5. Составление прайс-листа стоимости доставки из расчета удаленности до клиента (учесть километраж и цену топлива)
6. Составление прайс-листа о стоимости товаров с учетом НДС
7. Составление прайс-листа о стоимости услуг в автосервисе
8. Типы сущностей

Тип «Покупатель»

|  |  |  |
| --- | --- | --- |
| **Название атрибута** | **Тип данных** | **Формат** |
| ФИО | Строка | - |
| Номер телефона | Число | +7 (900) 000-00-00 |

Тип «Заказ»

|  |  |  |
| --- | --- | --- |
| **Название атрибута** | **Тип данных** | **Формат** |
| ID Заказа | Число | - |
| ID Товара | Число | - |
| Дата заказа | Дата | ДД.ММ.ГГГГ |

Тип «Товар»

|  |  |  |
| --- | --- | --- |
| **Название атрибута** | **Тип данных** | **Формат** |
| Наименование | Строка | - |

Тип «Шины»

|  |  |  |
| --- | --- | --- |
| **Название атрибута** | **Тип данных** | **Формат** |
| ID Товара | Число | - |

Тип «Тип резины»

|  |  |  |
| --- | --- | --- |
| **Название атрибута** | **Тип данных** | **Формат** |
| Шипы | логический | да/нет |
| Липучка | логический | да/нет |
| Зима | логический | да/нет |
| Лето | логический | да/нет |

Тип «Диски»

|  |  |  |
| --- | --- | --- |
| **Название атрибута** | **Тип данных** | **Формат** |
| ID Товара | Число | - |

Тип «Размеры шин»

|  |  |  |
| --- | --- | --- |
| **Название атрибута** | **Тип данных** | **Формат** |
| Высота | Число | - |
| Ширина | Число | - |
| Диаметр | Число | - |

Тип «Размеры дисков»

|  |  |  |
| --- | --- | --- |
| **Название атрибута** | **Тип данных** | **Формат** |
| Ширина обода | Число | - |
| Диаметр обода | Число | - |

Тип «Производитель дисков»

|  |  |  |
| --- | --- | --- |
| **Название атрибута** | **Тип данных** | **Формат** |
| ID Производителя | Число | - |
| Наименование | Строка | - |
| Страна | Строка | - |

Тип «Производитель шин»

|  |  |  |
| --- | --- | --- |
| **Название атрибута** | **Тип данных** | **Формат** |
| ID Производителя | Число | - |
| Наименование | Строка | - |
| Страна | Строка | - |

Тип «Услуги автосервиса»

|  |  |  |
| --- | --- | --- |
| **Название атрибута** | **Тип данных** | **Формат** |
| ID Услуги | Число | - |
| Наименование | Строка | - |
| Стоимость | Число | - |

Тип «Запись»

|  |  |  |
| --- | --- | --- |
| **Название атрибута** | **Тип данных** | **Формат** |
| ID Записи | Число | - |
| Дата записи | Дата | ДД.ММ.ГГГГ |

Тип «Магазин»

|  |  |  |
| --- | --- | --- |
| **Название атрибута** | **Тип данных** | **Формат** |
| Адрес | Строка | - |

Тип «Сотрудник»

|  |  |  |
| --- | --- | --- |
| **Название атрибута** | **Тип данных** | **Формат** |
| ID Сотрудника | Число | - |
| ФИО | Строка | - |
| Должность | Строка | - |
| Стаж работы в магазине | Число | - |

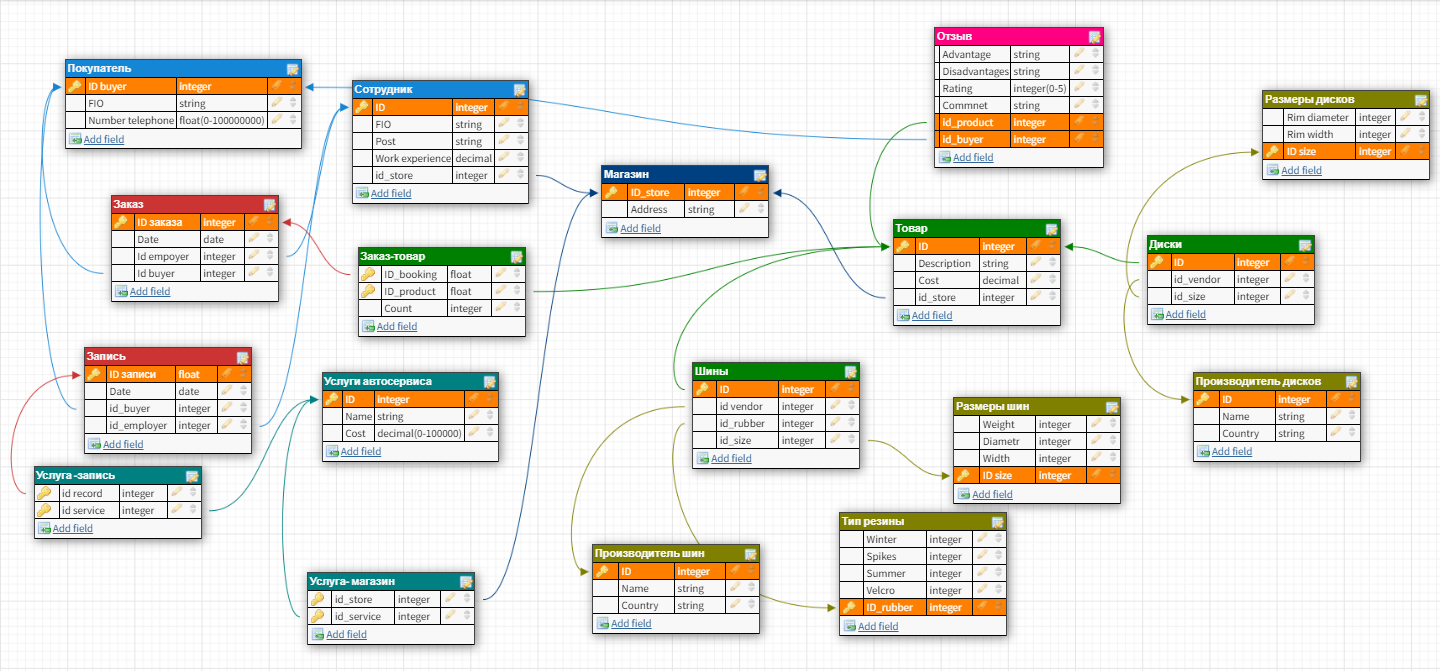
Тип «Отзыв»

|  |  |  |
| --- | --- | --- |
| **Название атрибута** | **Тип данных** | **Формат** |
| Оценка товара | Число | - |
| Достоинства | Строка | - |
| Недостатки | Строка | - |
| Комментарий | Число | - |

# ER-диаграмма

# Логическое проектирование БД

# Реляционная модель:



# Набор данных

1. Размеры дисков

|  |  |  |  |
| --- | --- | --- | --- |
| Диаметр | 15 | 16 | 17 |
| Ширина | 200 | 205 | 215 |
| ID размеров | 1 | 2 | 3 |

1. Производитель дисков

|  |  |  |  |
| --- | --- | --- | --- |
| ID | 21 | 22 | 23 |
| Страна | Китай | США | Германия |
| Наименование | Alcasta | 5-zigen | AEZ |

1. Диски

|  |  |  |  |
| --- | --- | --- | --- |
| ID | 31 | 32 | 33 |
| Id производителя | 21 | 22 | 23 |
| Id размеров | 1 | 2 | 3 |

1. Тип резины

|  |  |  |  |
| --- | --- | --- | --- |
| Лето | да | Нет | Нет |
| Зима | нет | Да | Да |
| Шипы | Нет | Да | Нет |
| Липучка | нет | нет | Да |
| ID резины | 41 | 42 | 43 |

1. Шины

|  |  |  |  |
| --- | --- | --- | --- |
| ID | 51 | 52 | 53 |
| Id производителя | 71 | 72 | 73 |
| Id размеров | 61 | 62 | 63 |
| Id резины | 41 | 42 | 43 |

1. Параметры шин

|  |  |  |  |
| --- | --- | --- | --- |
| Диаметр | 15 | 16 | 17 |
| Ширина | 200 | 205 | 215 |
| Высота | 50 | 55 | 60 |
| ID размеров | 61 | 62 | 63 |

1. Производитель шин

|  |  |  |  |
| --- | --- | --- | --- |
| ID | 71 | 72 | 73 |
| Страна | Япония | США | Германия |
| Наименование | Yokohama | Cooper | Toyo |

1. Товар

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Описание | Хорошие шины для зимы | Литые диски для машин | Идеальные шины для бездорожья | Идеальные шины для бездорожья | Литые диски для машин |
| Стоимость (руб) | 15000,00 | 20000,00 | 40000,00 | 40000,00 | 20000,00 |
| ID | 1 | 2 | 3 | 4 | 5 |
| Id\_магазина | 1 | 2 | 3 | 4 | 4 |

1. Услуги автосервиса

|  |  |  |  |
| --- | --- | --- | --- |
| ID | 91 | 92 | 93 |
| Наименование | Шиномонтаж | Замена масла | Диагностика |
| Стоимость | 1500,00 | 5000,00 | 4200,00 |

1. Услуга- магазин

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id магазина | 1 | 2 | 3 | 4 |
| Id услуги | 91 | 93 | 92 | 91 |

1. Услуга-запись

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID услуги | 91 | 93 | 92 | 91 |
| ID записи | 171 | 172 | 173 | 174 |

1. Заказ-товар

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID\_товара | 51 | 32 | 53 | 53 | 32 |
| ID\_заказа | 161 | 162 | 163 | 164 | 164 |
| Количество товара | 2 | 4 | 1 | 4 | 4 |

1. Магазин

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес | Ул. Козловская, 55 | Ул. Шекснинская ,170 | Ул. 8-й воздушной армии, 14 | Ул. Ленина, 55 |
| ID\_магазина | 1 | 2 | 3 | 4 |

1. Сотрудник

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 141 | 142 | 143 | 144 | 145 |
| ФИО | Калмыков И.А. | Василин Е.Г. | Мурмская О.А. | Кирин Ф.И. | Фомина И.В. |
| Должность | Менеджер | Автослесарь | Кассир | Шиномонтажник | Менеджер |
| Стаж(лет) | 1 | 5 | 3 | 1 | 3 |
| Id магазин | 1 | 3 | 2 | 4 | 4 |

1. Отзыв

|  |  |  |  |
| --- | --- | --- | --- |
| Достоинства | Хватило на 5+ лет, проходимые | Цена | Красивые на литье |
| Недостатки | Нет | Развалились за год | Нет |
| Оценка | 5 | 1 | 5 |
| Комментарий | Классные шины, рекомендую | Не рекомендую покупать данные шины | Для ценителей красивых дисков |
| ID\_товара | 53 | 51 | 32 |
| ID\_покупателя | 181 | 183 | 183 |

1. Заказ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID заказа | 161 | 162 | 163 | 164 |
| ID покупателя | 181 | 183 | 183 | 184 |
| Дата | 26.04.2016 | 24.11.2018 | 31.07.2020 | 14.03.2021 |
| ID сотрудника | 141 | 144 | 141 | 145 |

1. Запись

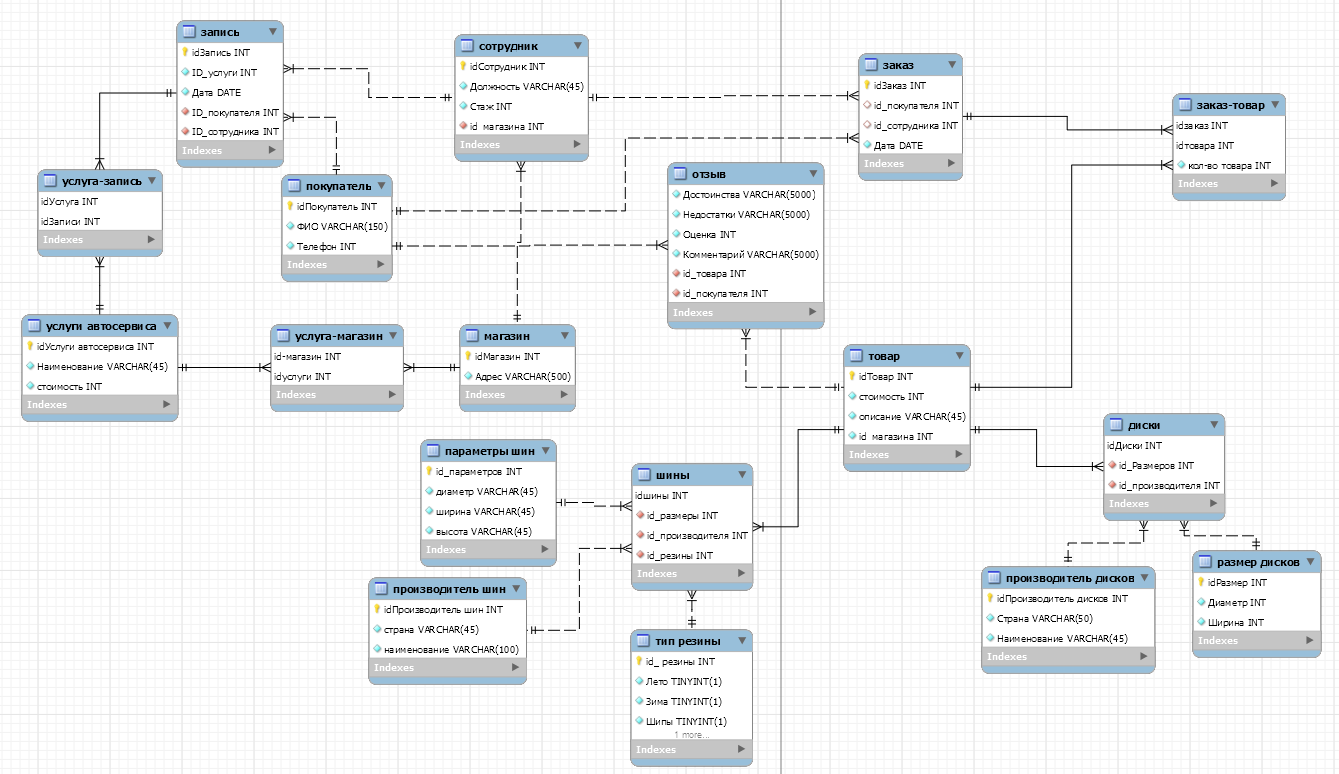
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID записи | 171 | 172 | 173 | 174 |
| ID услуги | 91 | 91 | 92 | 91 |
| Дата | 15.03.2021 | 26.05.2019 | 07.06.2020 | 14.03.2021 |
| ID покупателя | 181 | 183 | 182 | 184 |
| ID сотрудника | 142 | 142 | 142 | 144 |

1. Покупатель

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | 181 | 182 | 183 | 184 |
| ФИО | Климов А.В. | Белов И.К. | Альпина В.А. | Иванов И.И. |
| Телефон | +79044556336 | +79176478899 | +7(961)5436545 | +7(988)4516323 |

# Физическое проектирование БД

# Реляционная схема в MySQL



# Реляционная схема (из второй лабораторной работы)

# 

# 3. Таблицы

**Для дисков:**

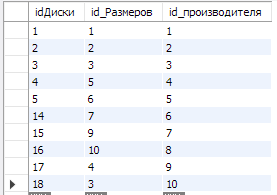
1. Размеры дисков



1. Производитель дисков

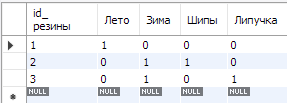


1. Диски

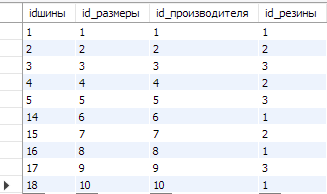


**Для шин:**

1. Тип резины



1. Шины



1. Параметры шин



1. Производитель шин

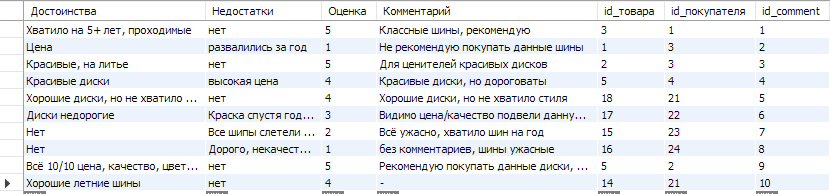


Товар:

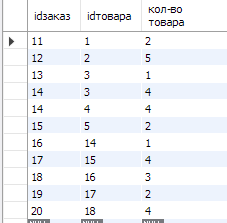
1. Товар



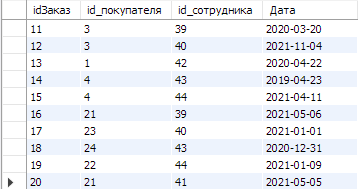
1. Отзыв



1. Заказ-товар



1. Заказ

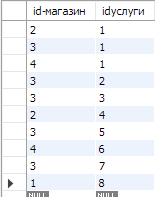


Автосервис:

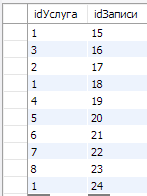
1. Услуги автосервиса



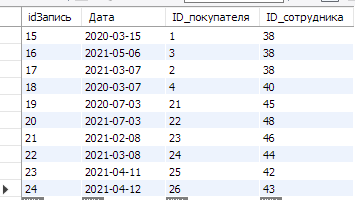
1. Услуга- магазин



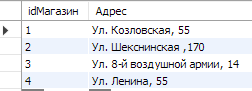
1. Услуга-запись



1. Запись



1. Магазин



1. Сотрудник

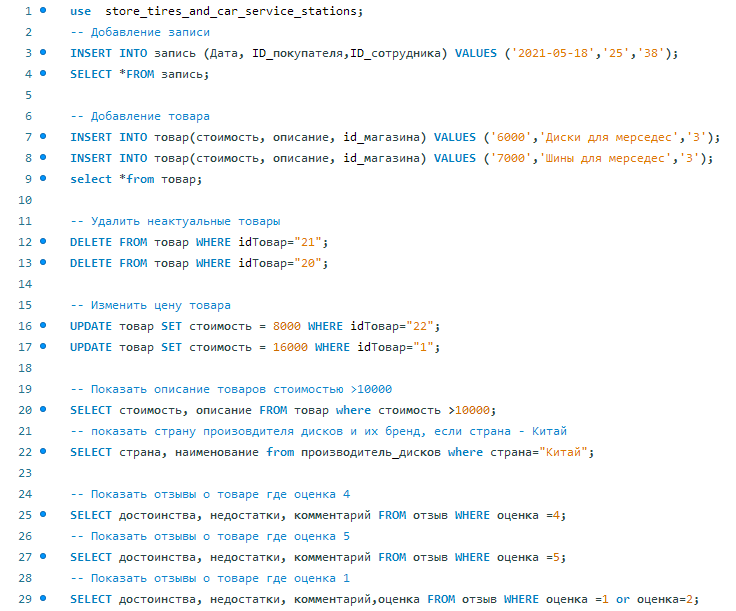


1. Покупатель

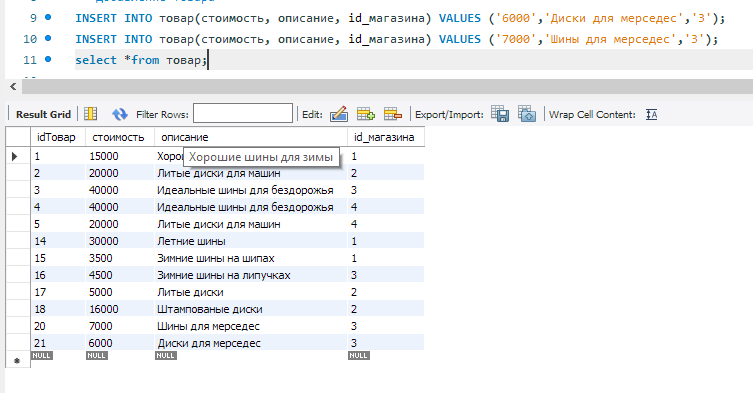


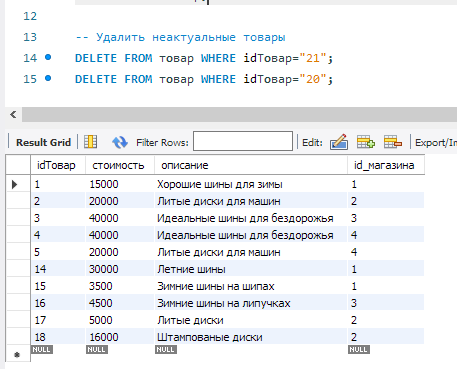
# Реализация SQL-запросов базы данных

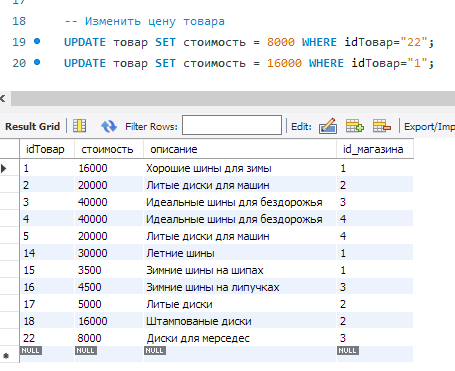
Функциональные требования

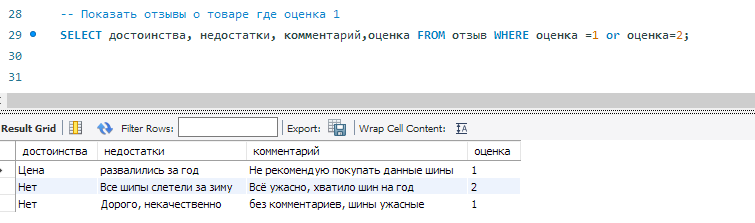


Пример работы программы:

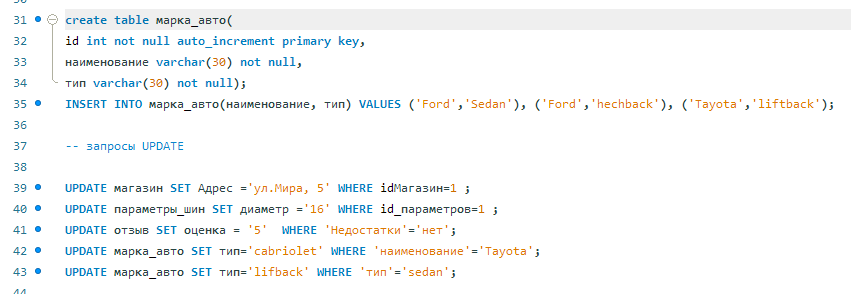


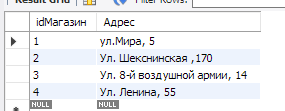


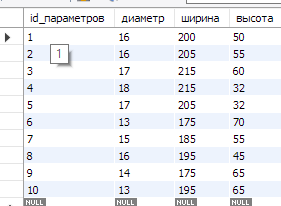


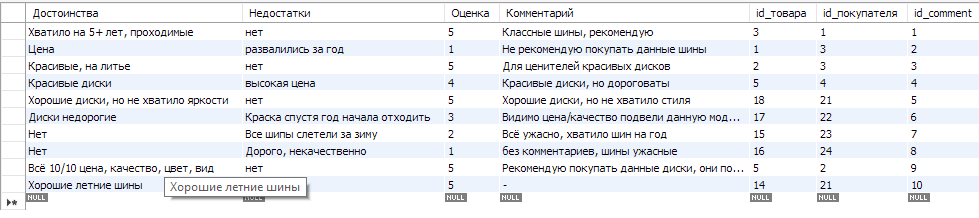


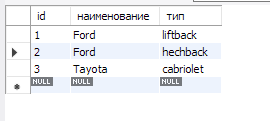
UPDATE – обновление параметров



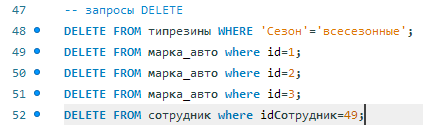








DELETE – удаление объектов







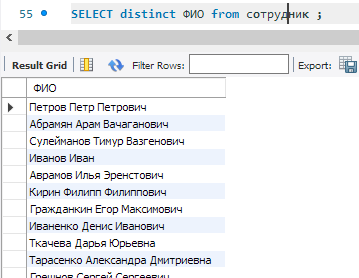


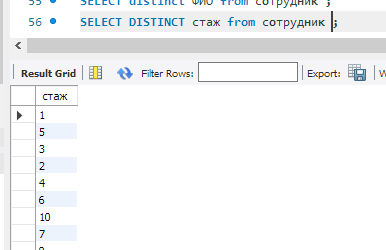
**SELECT, DISTINCT, WHERE, AND/OR/NOT, IN, BETWEEN**, различная работа с **датами**и **числами**, преобразование данных, **IS NULL**, **AS**

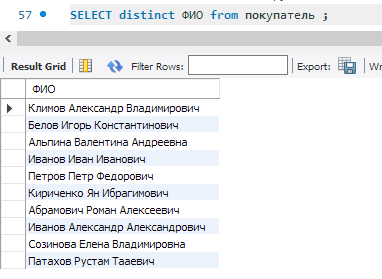
**-- запросы SELECT**

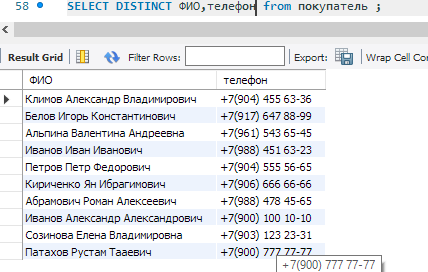
**-- DISTINCT – показать неповторяющиеся данные**

1. SELECT DISTINCT ФИО FROM сотрудник ;
2. SELECT DISTINCT стаж FROM сотрудник ;
3. SELECT DISTINCT ФИО FROM покупатель ;
4. SELECT DISTINCT ФИО,телефон FROM покупатель ;







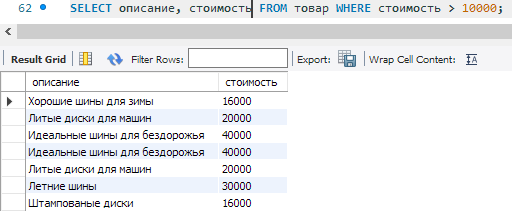


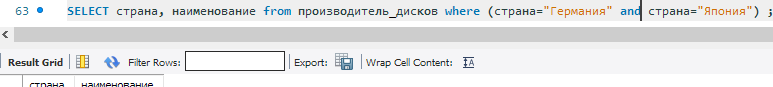
-- WHERE условие показа таблиц

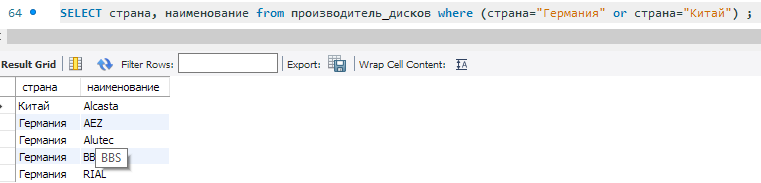
1. SELECT описание, стоимость FROM товар WHERE стоимость > 10000;
2. SELECT страна, наименование from производитель\_дисков where (страна="Германия" and страна="Япония") ;
3. SELECT страна, наименование from производитель\_дисков where (страна="Германия" or страна="Китай") ;
4. SELECT наименование, сезон FROM типрезины WHERE сезон="лето" ;
5. SELECT наименование, стоимость FROM услуги\_автосервиса WHERE стоимость BETWEEN 2000 AND 4000;
6. SELECT диаметр,ширина FROM `размер дисков` WHERE ширина BETWEEN 200 AND 220;
7. SELECT диаметр,ширина,высота FROM параметры\_шин WHERE ширина IN (200,215);
8. SELECT наименование, стоимость FROM услуги\_автосервиса WHERE стоимость IN (1000,3000);
9. SELECT наименование, стоимость FROM услуги\_автосервиса WHERE стоимость NOT IN (1000,3000);
10. SELECT описание, стоимость FROM товар WHERE NOT (стоимость=6000 or стоимость=7000) ;
11. SELECT \*FROM товар WHERE стоимость IS NOT NULL;
12. SELECT \*FROM товар WHERE стоимость IS NULL;

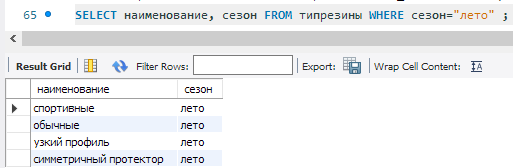
-- AS

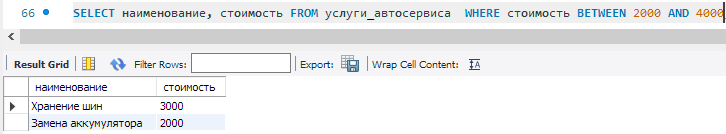
1. SELECT idТовар AS ID, стоимость AS цена FROM товар;
2. SELECT недостатки AS минусы, достоинства AS плюсы FROM отзыв;
3. SELECT idЗапись AS id FROM запись;
4. SELECT idПокупатель AS id FROM покупатель;

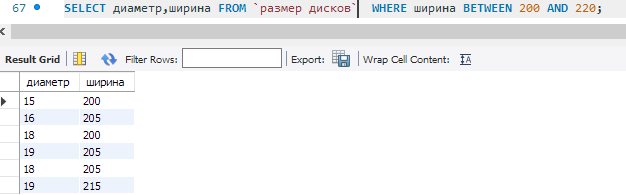


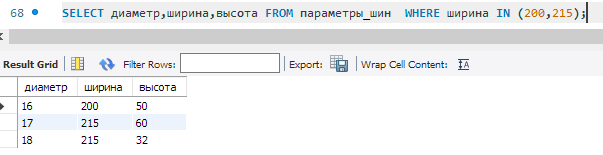


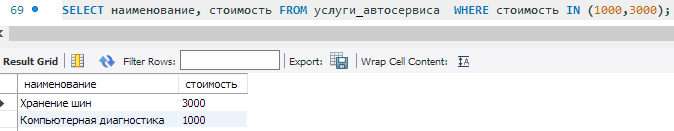






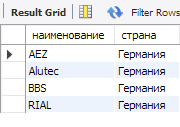






**-- LIKE**

1. SELECT наименование, страна FROM производитель\_дисков WHERE страна LIKE 'Германия';
2. SELECT наименование, страна FROM `производитель шин` WHERE страна LIKE 'Япония';
3. SELECT Адрес FROM магазин WHERE Адрес LIKE 'Ул. Ленина, 55';
4. SELECT телефон, ФИО FROM покупатель WHERE ФИО LIKE 'Иванов Иван Иванович';
5. SELECT наименование, страна FROM производитель\_дисков WHERE страна LIKE 'Г%';
6. SELECT наименование, страна FROM производитель\_дисков WHERE страна LIKE '%я';



**-- INSERT**

create table новая\_таблица(

id int not null auto\_increment primary key,

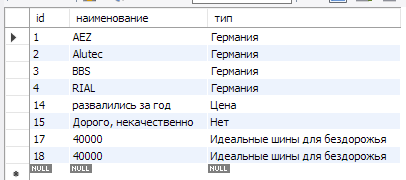
наименование varchar(30) not null,

тип varchar(30) not null);

INSERT INTO марка\_авто(наименование, тип) VALUES ('Ford','Sedan'), ('Ford','hechback'), ('Tayota','liftback');

-- INSERT – перенос данных из одной таблицы в другую

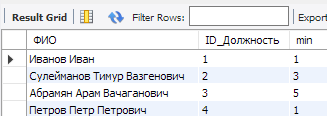
1. INSERT INTO новая\_таблица (наименование,тип) SELECT наименование, страна FROM производитель\_дисков WHERE страна="Германия";
2. INSERT INTO новая\_таблица (наименование,тип) SELECT недостатки, достоинства FROM отзыв WHERE оценка=1;
3. INSERT INTO новая\_таблица (наименование,тип) SELECT стоимость, описание FROM товар WHERE стоимость=40000;

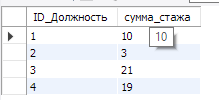


**GROUP BY – группировка по определенному признаку**

-- GROUP BY

1. SELECT COUNT(id), страна FROM `производитель шин` GROUP BY страна;
2. SELECT COUNT(idЗаказ), id\_покупателя FROM заказ GROUP BY id\_покупателя;
3. SELECT COUNT(id), страна FROM производитель\_дисков GROUP BY страна;
4. SELECT магазин.idМагазин, сотрудник.ФИО FROM магазин, сотрудник GROUP BY idМагазин;
5. SELECT товар.стоимость, `производитель шин`.страна FROM товар, `производитель шин` GROUP BY стоимость;
6. SELECT \* FROM услуги\_автосервиса WHERE id>0 LIMIT 3;
7. SELECT \* FROM услуги\_автосервиса LIMIT 5;
8. SELECT \* FROM услуги\_автосервиса WHERE id>0 ORDER BY id LIMIT 3;
9. SELECT наименование, SUM(стоимость) as sum FROM услуги\_автосервиса GROUP BY стоимость;
10. SELECT ID\_Должность, SUM(Стаж) as сумма\_стажа FROM сотрудник GROUP BY ID\_Должность;
11. SELECT страна, count(\*) as фирмы\_в\_стране FROM производитель\_дисков GROUP BY страна;
12. SELECT idтовара, SUM(количество) as кол\_во\_товара FROM `заказ-товар` GROUP BY idтовара;
13. SELECT COUNT(idтовара), количество FROM `заказ-товар` GROUP BY количество HAVING COUNT(idтовара)>1;
14. SELECT ФИО,ID\_Должность, MAX(стаж) as max FROM сотрудник GROUP BY ID\_Должность;
15. SELECT ФИО,ID\_Должность, MIN(стаж) as min FROM сотрудник GROUP BY ID\_Должность;
16. SELECT стоимость, описание FROM товар WHERE idТовар>0 ORDER BY стоимость;
17. SELECT стоимость, наименование FROM услуги\_автосервиса WHERE id>0 ORDER BY стоимость DESC;
18. SELECT ID\_Должность, AVG(стаж) as средний\_стаж FROM сотрудник WHERE ID\_Должность=4 ;
19. SELECT AVG(стоимость), idТовар FROM товар GROUP BY idТовар HAVING AVG(стоимость) > 10000;
20. SELECT idТовар, SUM(стоимость) FROM товар GROUP BY стоимость;
21. SELECT ID\_Должность, AVG(стаж) as средний\_стаж FROM сотрудник GROUP BY ID\_Должность HAVING AVG(стаж) < 7 ;

****

****

**JOIN - выборка данных из двух таблиц и включение этих данных в один результирующий набор**

-- JOIN

1. SELECT сотрудник.idСотрудник, сотрудник.ФИО `Имя сотрудника`, должность.id `id\_должности`, должность.имя `Должность`

FROM `сотрудник` сотрудник INNER JOIN `должность` должность ON должность.id = сотрудник.ID\_Должность;

1. SELECT сотрудник.idСотрудник, сотрудник.ФИО `Имя сотрудника`, магазин.idМагазин `id\_магазина`, магазин.адрес `Адрес магазина`

FROM `сотрудник` сотрудник INNER JOIN `Магазин` магазин ON магазин.idМагазин = сотрудник.id\_магазина;

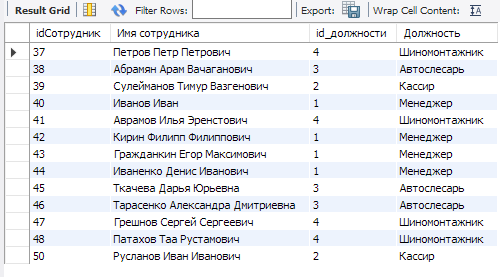
1. SELECT товар.idТовар, товар.стоимость, диски.idДиски, диски.id\_производителя, производитель\_дисков.Наименование

FROM товар INNER JOIN диски ON диски.idДиски = товар.idТовар INNER JOIN производитель\_дисков ON производитель\_дисков.id = диски.id\_производителя;

1. SELECT сотрудник.idСотрудник, сотрудник.ФИО `Имя сотрудника`, должность.id `id\_должности`, должность.имя `Должность`

FROM сотрудник RIGHT OUTER JOIN должность ON должность.id = сотрудник.ID\_Должность;

1. SELECT типрезины.сезон, шины.idшины FROM типрезины LEFT JOIN шины USING(idРезины);
2. SELECT типрезины.сезон, шины.idшины FROM типрезины RIGHT JOIN шины USING(idРезины);
3. SELECT товар.idТовар,товар.описание, отзыв.комментарий FROM товар RIGHT JOIN отзыв USING(idтовар);
4. SELECT сотрудник.ФИО `Сотрудник`, магазин.адрес`место работы` FROM сотрудник LEFT JOIN магазин USING(idМагазин);
5. SELECT \* FROM типрезины CROSS JOIN шины;
6. SELECT \* FROM отзыв CROSS JOIN покупатель;
7. SELECT \* FROM отзыв CROSS JOIN магазин;
8. SELECT \* FROM услуги\_автосервиса, магазин FULL JOIN `услуга-магазин`;
9. SELECT стоимость, описание, адрес FROM товар CROSS JOIN магазин ;
10. SELECT ФИО, адрес "Место работы" FROM сотрудник CROSS JOIN магазин;
11. SELECT сезон, наименование, idшины FROM типрезины CROSS JOIN шины;
12. SELECT \*FROM покупатель NATURAL JOIN отзыв;
13. SELECT IDПокупатель, ФИО, Дата FROM покупатель NATURAL JOIN запись;
14. SELECT \*FROM должность NATURAL JOIN сотрудник;
15. SELECT IDПокупатель, ФИО, адрес FROM покупатель NATURAL JOIN магазин;
16. SELECT \* FROM производитель\_дисков CROSS JOIN диски ;

****

****

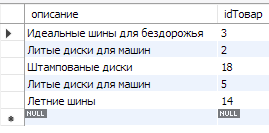
**-- UNION, EXCEPT, INTERSECT**

-- Объединение цен на товары и услуги

1. SELECT `Стоимость` FROM товар UNION SELECT `Стоимость` FROM услуги\_автосервиса ORDER BY `Стоимость`;
2. SELECT \* FROM производитель\_шин WHERE EXISTS(SELECT страна FROM производитель\_шин WHERE id>5);
3. SELECT `страна` FROM производитель\_шин UNION SELECT `Страна` FROM производитель\_дисков ORDER BY `страна`;

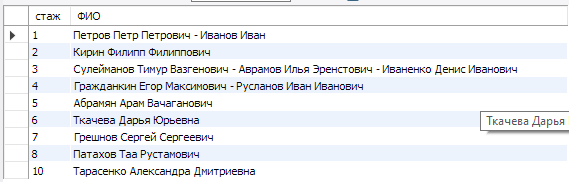
**-- ALL, ANY, EXISTS**

1. SELECT ALL \*FROM производитель\_дисков WHERE страна LIKE 'Г%';
2. SELECT ALL \*FROM покупатель WHERE ФИО LIKE '%а';
3. SELECT \* FROM товар WHERE EXISTS(SELECT стоимость FROM товар WHERE стоимость > 15000);
4. SELECT описание, idТовар FROM товар WHERE idТовар = ANY (SELECT idТовар FROM отзыв WHERE оценка = 5);



**-- GROUP\_CONCAT**

1. SELECT GROUP\_CONCAT(страна) as Страны FROM производитель\_шин;
2. SELECT GROUP\_CONCAT(стоимость) as Цены FROM товар;
3. SELECT GROUP\_CONCAT(наименование) as Страны FROM производитель\_шин;
4. SELECT GROUP\_CONCAT(адрес) as Страны FROM магазин;
5. SELECT стаж, GROUP\_CONCAT(ФИО SEPARATOR ' - ') as ФИО FROM сотрудник GROUP BY стаж;



-- сложные запросы

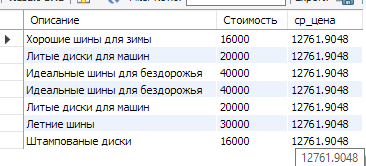
1. SELECT товар.idТовар, товар.стоимость, диски.idДиски, диски.id\_производителя, производитель\_дисков.Наименование

FROM товар INNER JOIN диски ON диски.idДиски = товар.idТовар

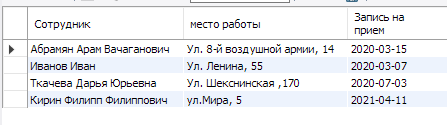
INNER JOIN производитель\_дисков ON производитель\_дисков.id = диски.id\_производителя;



1. SELECT Описание, Стоимость, (SELECT AVG(Стоимость) FROM товар) AS ср\_цена FROM товар WHERE стоимость > (SELECT AVG(стоимость) FROM товар);



1. SELECT сотрудник.ФИО `Сотрудник`, магазин.адрес`место работы`, запись.Дата as `Запись на прием` FROM сотрудник Inner JOIN магазин USING(idМагазин) Inner JOIN запись ON запись.IDСотрудника= сотрудник.idСотрудник GROUP BY адрес;



# Индексы, хранимые процедуры, представления

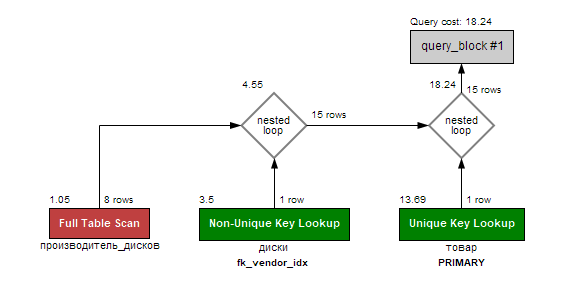
1. -- выводит всю информацию по дискам

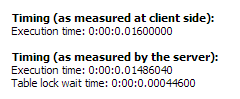
SELECT товар.idТовар, товар.стоимость, диски.idДиски, диски.idvendor, производитель\_дисков.Наименование

FROM товар INNER JOIN диски ON диски.idДиски = товар.idТовар

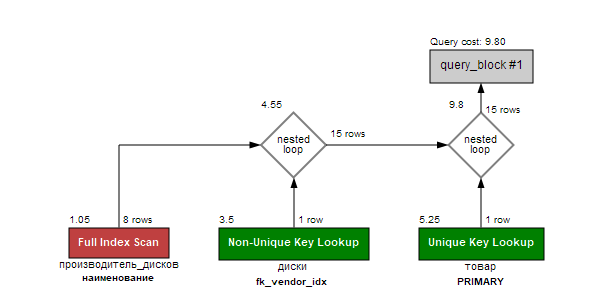
INNER JOIN производитель\_дисков ON производитель\_дисков.id = диски.idvendor;

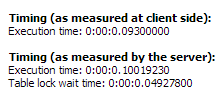
Без индексов:





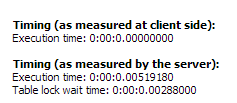
С индексами:

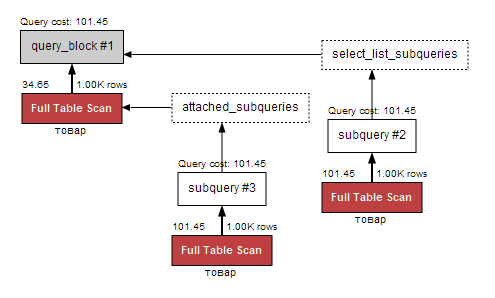




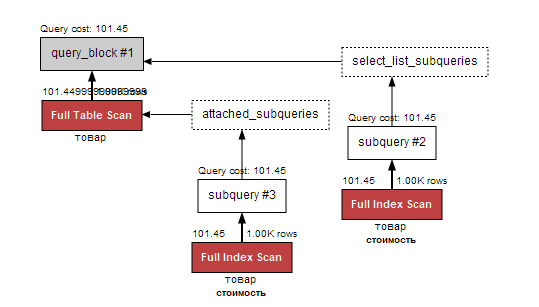
1. -- средняя цена на товар

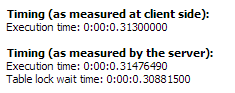
SELECT описание, стоимость, (SELECT AVG(стоимость) FROM товар) AS ср\_цена FROM товар WHERE стоимость > (SELECT AVG(стоимость) FROM товар);





С индексами:



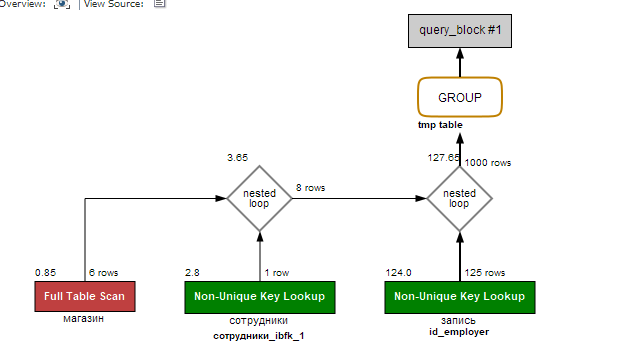


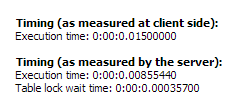
1. -- к кому больше всего записи и в каком магазине

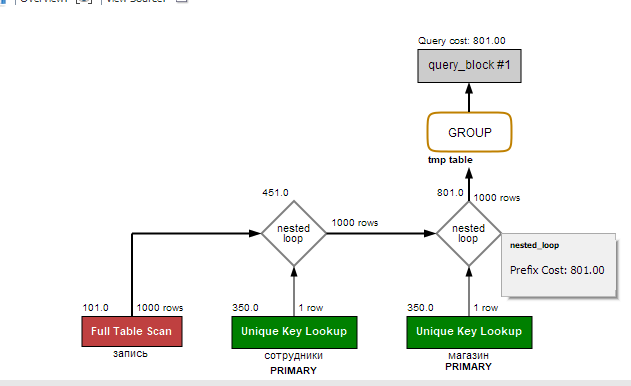
SELECT сотрудники.FIO `Сотрудник`, магазин.address`место работы`, запись.Date as `Запись на прием`

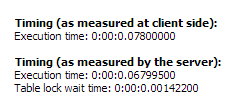
FROM сотрудники Inner JOIN магазин USING(idstore) Inner JOIN запись

ON запись.id\_employer= сотрудники.id\_employer GROUP BY address;





С индексами:



1. -- покупатель сделал заказ, указать сумму товаров заказа, на какую дату

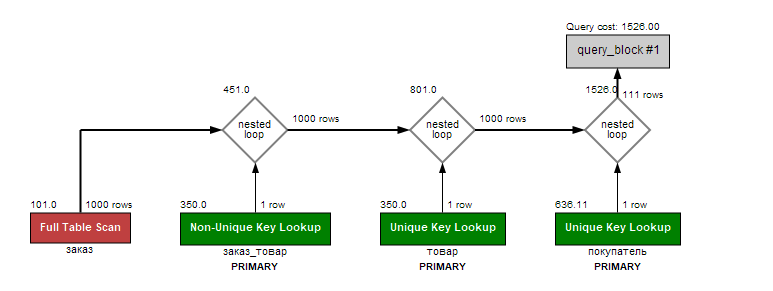
SELECT покупатель.FIO, SUM(стоимость) as "Общая стоимость товара", GROUP\_CONCAT(заказ.Date) AS "Дата заказов",GROUP\_CONCAT(товар.описание) as "Описание товара" -- производитель\_дисков.Наименование, производитель\_шин.наименование

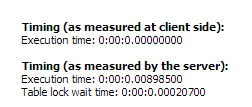
FROM покупатель JOIN заказ USING (id\_buyer)

JOIN `заказ\_товар` on `заказ\_товар`.id\_booking=заказ.id\_record

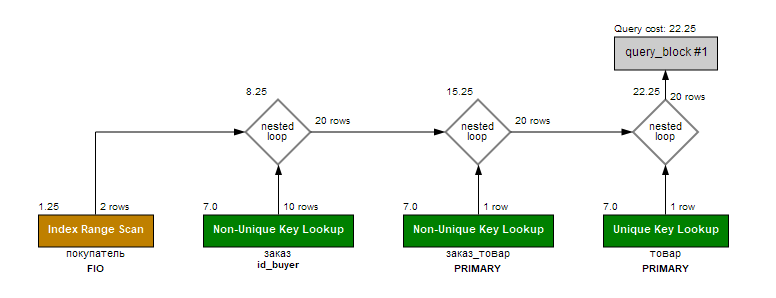
JOIN товар on товар.idТовар=заказ\_товар.id\_product

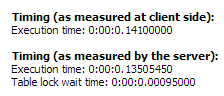
WHERE покупатель.FIO LIKE "Penn Klimko" ;





С индексами:





1. -- всю информацию по шинам через товар с описанием+ средняя оценка товара

SELECT distinct страна, производитель\_шин.Имя,

товар.стоимость , параметры\_шин.диаметр, параметры\_шин.высота, параметры\_шин.ширина, тип\_резины.Сезон, тип\_резины.Наименование,

AVG(Оценка) as средняя\_оценка

FROM шины

LEFT JOIN тип\_резины ON тип\_резины.idРезины= шины.id\_резины

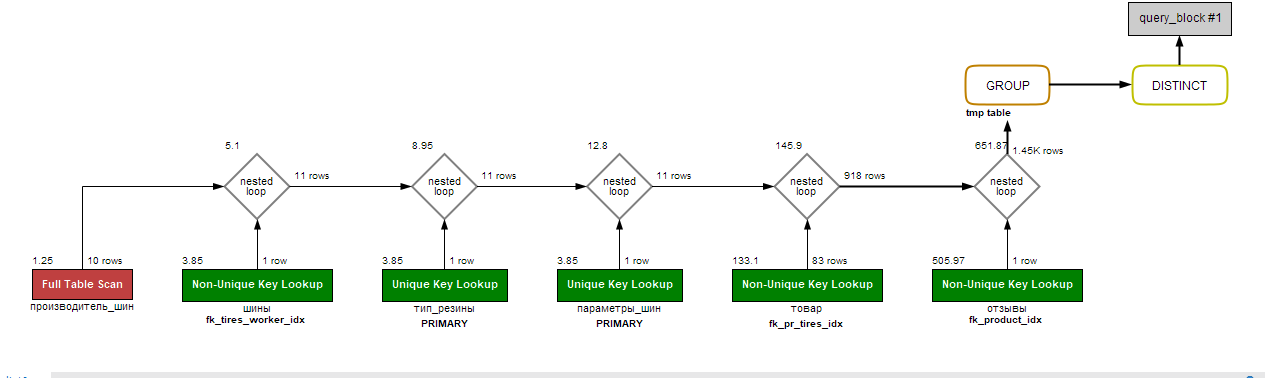
RIGHT JOIN товар ON товар.id\_шины=шины.idшины

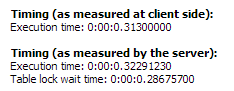
JOIN отзывы ON отзывы.id\_товара=товар.idТовар

JOIN производитель\_шин ON производитель\_шин.id\_vendor\_psh=шины.id\_производителя

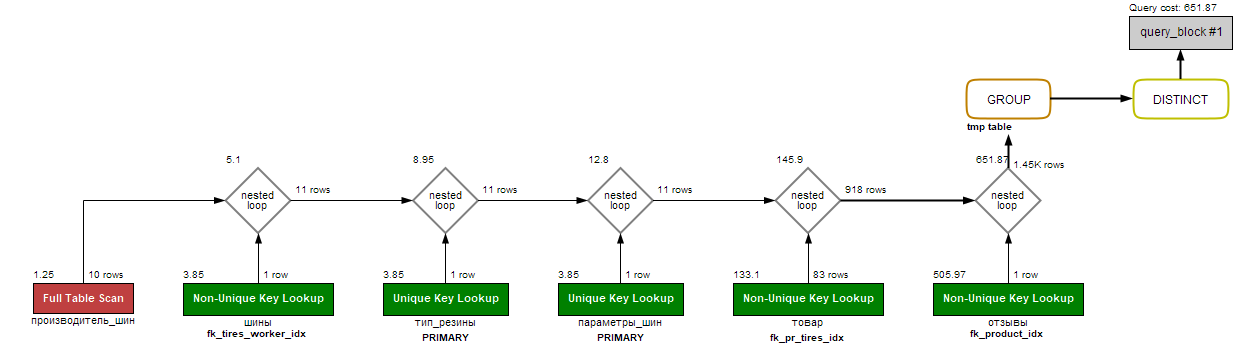
JOIN параметры\_шин ON параметры\_шин.id\_параметров=шины.id\_размеры

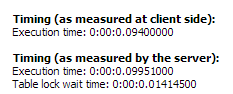
GROUP BY idшины;





С индексами:





1. -- топ 10 сотрудников определенного магазина за определенный период по записям

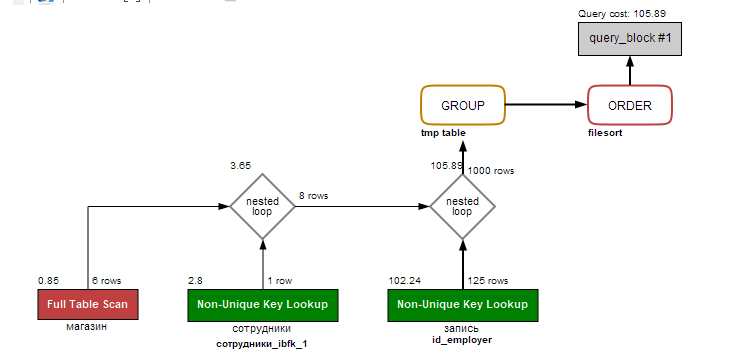
select COUNT(id\_record) as total, сотрудники.FIO, магазин.address

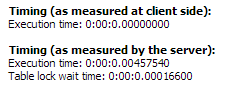
from сотрудники

join запись using(id\_employer)

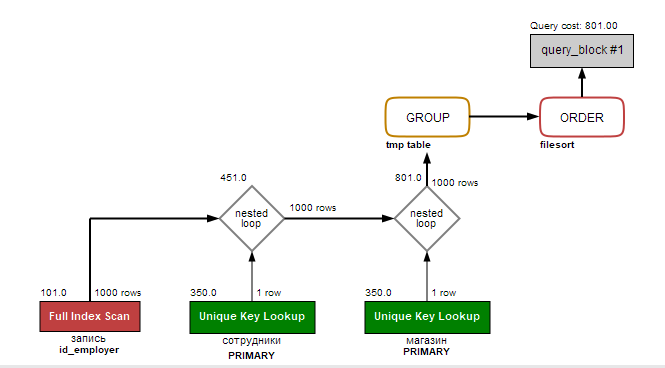
join магазин USING (idstore)

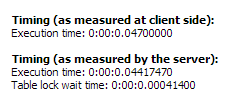
group by id\_employer order by total desc limit 10;



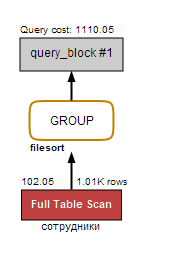


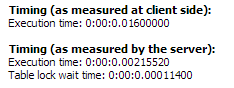
С индексами:



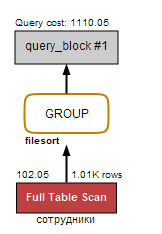


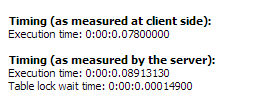
1. SELECT Work\_expirience, GROUP\_CONCAT(FIO SEPARATOR ' - ') as FIO FROM сотрудники GROUP BY Work\_expirience;





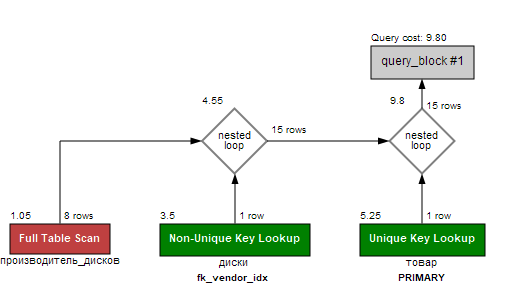
С индексами:

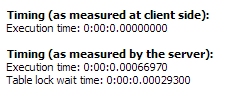




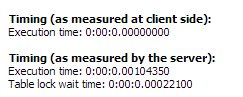
1. SELECT товар.idТовар, товар.стоимость, диски.idДиски, диски.idvendor, производитель\_дисков.Наименование

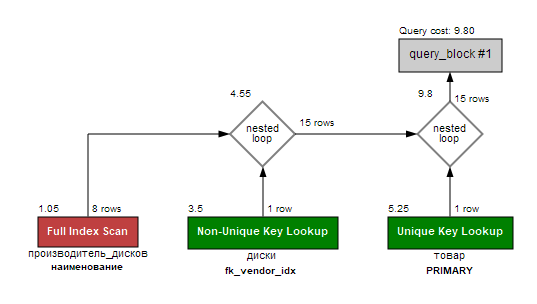
FROM товар INNER JOIN диски ON диски.idДиски = товар.idТовар INNER JOIN производитель\_дисков ON производитель\_дисков.id = диски.idvendor;





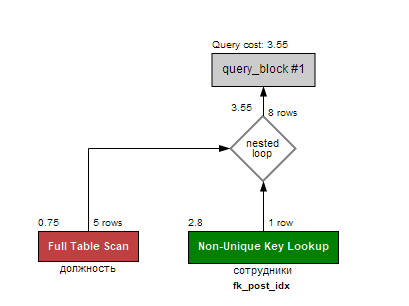
С индексами:

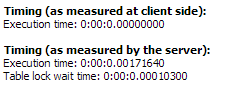




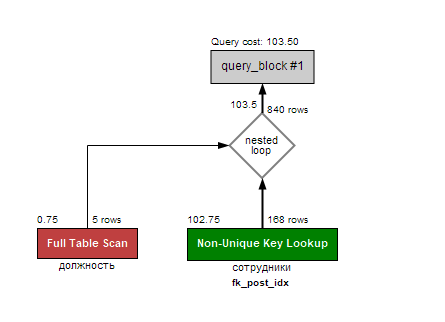
1. SELECT сотрудники.id\_employer, сотрудники.FIO `Имя сотрудника`, должность.id `id\_должности`, должность.Имя `Должность`

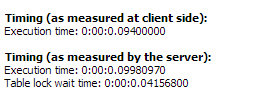
FROM `сотрудники` сотрудники INNER JOIN `должность` должность ON должность.id = сотрудники.id\_post;



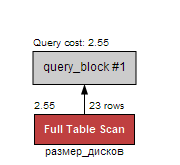


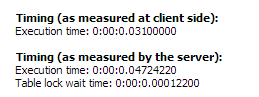
С индексами:



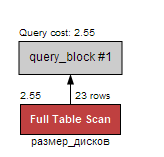


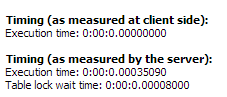
1. SELECT диаметр,ширина FROM `размер\_дисков` WHERE ширина BETWEEN 32 AND 55;



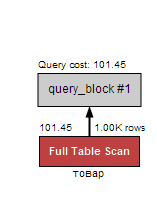


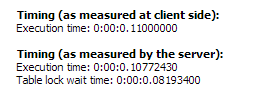
С индексами:



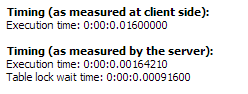


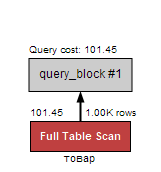
1. SELECT \*FROM товар WHERE стоимость IS NOT NULL;





С Индексами:





**Создание индексов:**

CREATE INDEX стоимость ON товар(стоимость);

CREATE INDEX id\_record on запись(id\_record);

CREATE INDEX FIO ON покупатель(FIO);

CREATE INDEX FIO ON сотрудники(FIO);

CREATE INDEX ширина ON размер\_дисков(ширина);

**Процедуры:**

-- Процедуры

drop procedure booking;

DELIMITER //

-- Процедура, которая показывает группы сотрудников с одинаковым стажем

CREATE PROCEDURE booking()

BEGIN

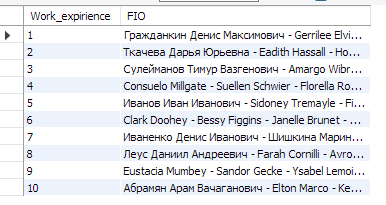
select Work\_expirience, GROUP\_CONCAT(FIO SEPARATOR ' - ') as FIO FROM сотрудники

GROUP BY Work\_expirience;

END //

DELIMITER ;

Call booking();



drop procedure product;

-- Процедура показывает всю информацию по дискам

DELIMITER //

CREATE PROCEDURE product()

BEGIN SELECT товар.idТовар,товар.id\_диски, товар.стоимость, диски.idДиски, диски.idvendor, производитель\_дисков.Наименование, IFNULL(Описание, 'нет описания')

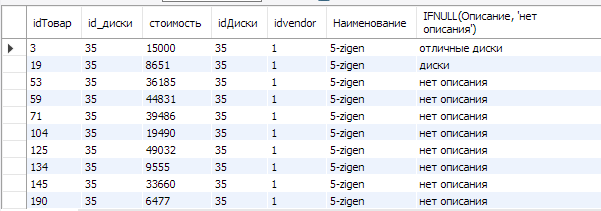
FROM товар INNER JOIN диски ON диски.idДиски = товар.id\_диски

INNER JOIN производитель\_дисков ON производитель\_дисков.id = диски.idvendor;

END //

DELIMITER ;

call product();



-- Дает нечеткую характеристику о товаре

drop procedure feedback;

DELIMITER //

CREATE PROCEDURE feedback()

BEGIN

SELECT оценка, Достоинства, недостатки,

CASE

WHEN оценка = 4 THEN 'хороший товар'

WHEN оценка = 5 THEN 'отличный товар'

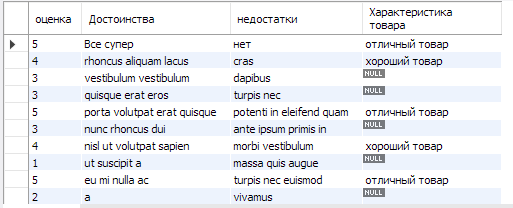
END AS 'Характеристика товара'

FROM отзывы;

END//

DELIMITER ;

call feedback();



**Функции:**

-- Функции

-- Расчет стоимости товара

DELIMITER //

CREATE FUNCTION inf\_product (стоимость int)

RETURNS INT

READS SQL DATA

DETERMINISTIC

BEGIN

DECLARE cost INT;

SET cost = 0;

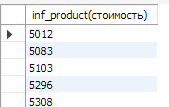
SELECT стоимость INTO cost FROM `товар` LIMIT 1;

RETURN cost;

END//

DELIMITER ;

SELECT inf\_product(стоимость) FROM товар;



-- Повышение рейтинга, переход от 5 балльной к 10 балльной системе оценки

drop function sr\_oc;

DELIMITER $$

CREATE FUNCTION Sr\_oc ( оценка INT)

RETURNS INT

DETERMINISTIC

BEGIN

IF оценка > 3 AND оценка = 5 THEN

SET оценка = Оценка/0.5;

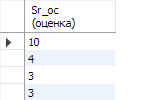
END IF;

RETURN (оценка);

END $$

DELIMITER ;

Select Sr\_oc (оценка) from отзывы;



-- Расчет стоимости услуг

DELIMITER //

CREATE FUNCTION inf\_serv (cost\_ int)

RETURNS INT

READS SQL DATA

DETERMINISTIC

BEGIN

DECLARE cost INT;

SET cost = 0;

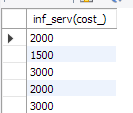
SELECT cost\_ INTO cost FROM `услуги\_автосервиса` LIMIT 1;

RETURN cost;

END//

DELIMITER ;

SELECT inf\_serv(cost\_) FROM услуги\_автосервиса;



**Представления:**

-- топ 20 сотрудников определенного магазина за определенный период по записям

Create VIEW Info AS select COUNT(id\_record) as total, сотрудники.FIO, магазин.address

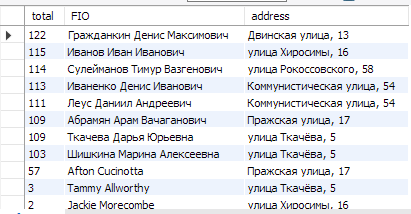
from сотрудники

join запись using(id\_employer)

join магазин USING (idstore)

group by id\_employer order by total desc limit 20;

SELECT \* FROM Info;

****

-- вся информация по шинам через товар с описанием+ средняя оценка товара

Create view Info2 as SELECT distinct страна, производитель\_шин.имя,

товар.стоимость , параметры\_шин.диаметр, параметры\_шин.высота, параметры\_шин.ширина, тип\_резины.сезон, тип\_резины.наименование,

AVG(оценка) as средняя\_оценка

FROM шины

LEFT JOIN тип\_резины ON тип\_резины.idРезины= шины.id\_резины

RIGHT JOIN товар ON товар.id\_шины=шины.idшины

JOIN отзывы ON отзывы.id\_товара=товар.idТовар

JOIN производитель\_шин ON производитель\_шин.id\_vendor\_psh=шины.id\_производителя

JOIN параметры\_шин ON параметры\_шин.id\_параметров=шины.id\_размеры

GROUP BY idшины;

SELECT \* FROM Info2;



-- покупатель сделал заказ, указать сумму товаров заказа, на какую дату

Create view Info3 as SELECT покупатель.FIO, SUM(стоимость) as "Общая стоимость товара", GROUP\_CONCAT(заказ.Date) AS "Дата заказов" -- ,GROUP\_CONCAT(товар.описание) as "Описание товара" -- производитель\_дисков.Наименование, производитель\_шин.наименование

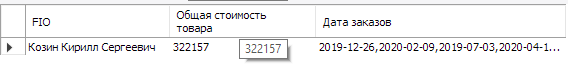
FROM покупатель JOIN заказ USING (id\_buyer)

JOIN `заказ\_товар` on `заказ\_товар`.id\_booking=заказ.id\_record

JOIN товар on товар.idТовар=заказ\_товар.id\_product

WHERE покупатель.FIO LIKE "Козин Кирилл Сергеевич" ;

SELECT \* FROM Info3;

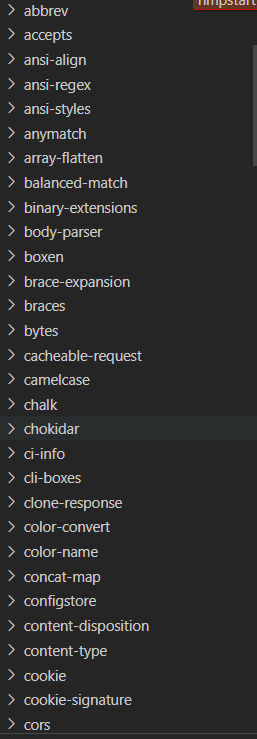


# Взаимодействие с БД из приложения

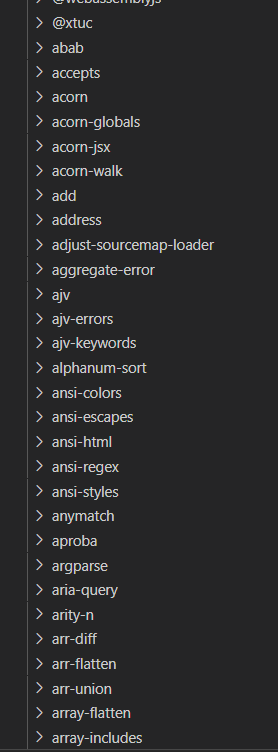
**1.** Для реализации лабораторной работы и в дальнейшем курсовой работы был выбран язык node js для сервера и javascript react.

**2.** Были изучены технологии (библиотеки) для работы с выбранной СУБД и реализован сайт для подключения к собственной БД в СУБД на выбранном языке программирования и с помощью выбранных библиотек для подключения:

Сервер:



Клиент:



**3.** В качестве интерфейса взаимодействия с пользователем на данной лабораторной работе был реализован сайт с товарами \_ шины и диски.

**4.** Исходный код программы был закоммичен в репозиторий, ссылка приложена в протокол.

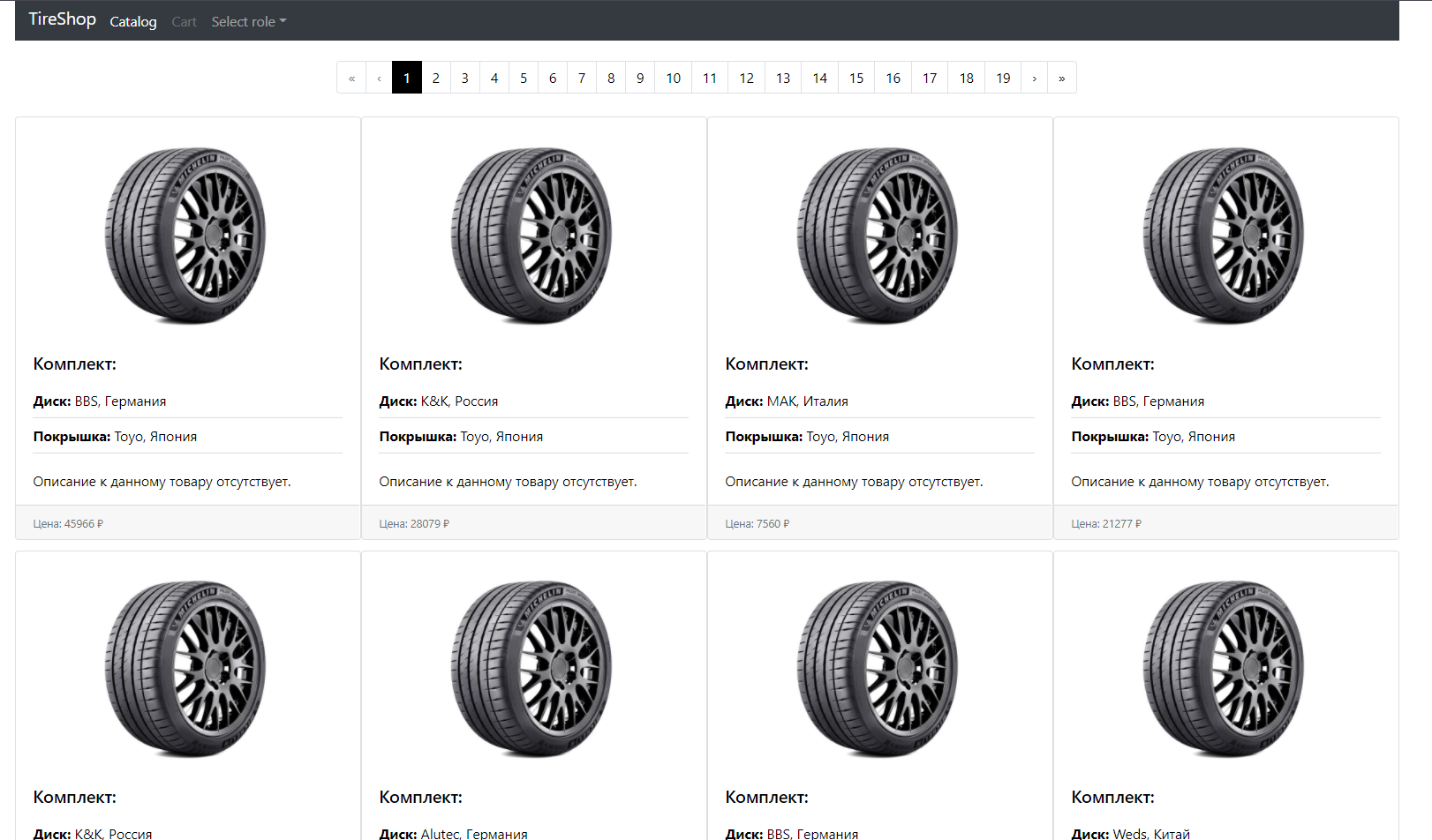
**5.** Была реализована CRUD (получение, добавление, редактирование и удаление данных) для таблиц – товар, диски, шины, производители шин и дисков, таблицы параметров и сезон, которые связаны внешними ключами.

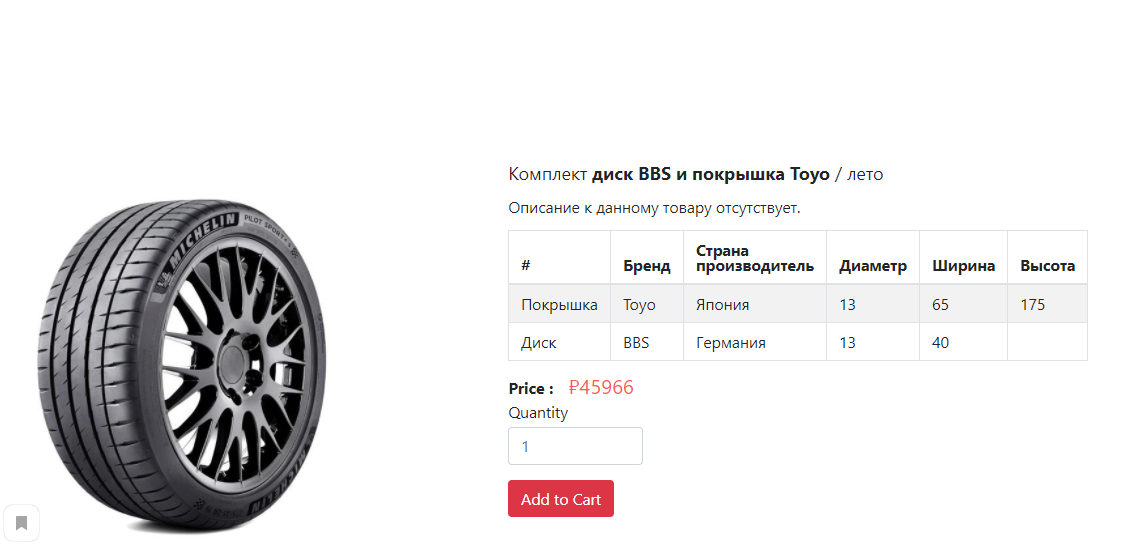
Присоединённые данные отображаются в виде атрибутов с данными из другой таблицы, а не как ID внешних ключей. Т.е. в запросах использованы JOIN’ы.

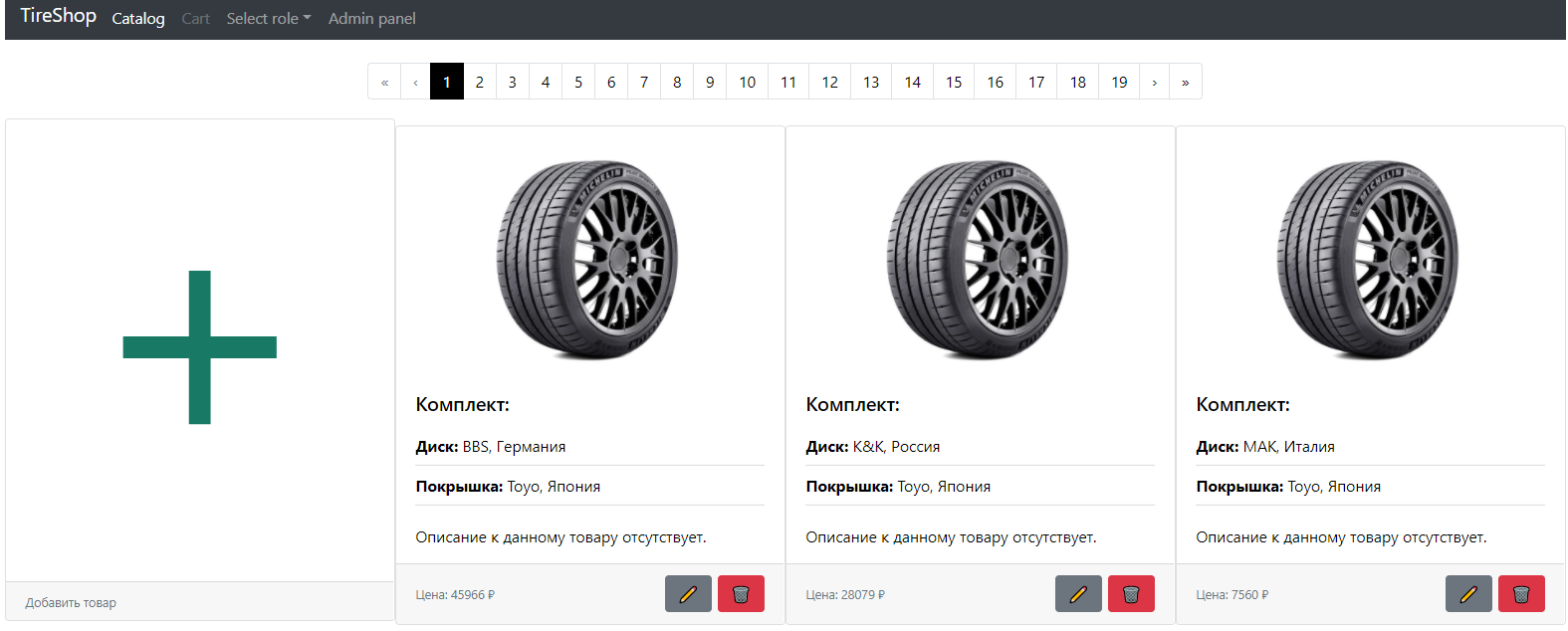
Удаление производится с помощью каскада, поэтому при удалении из одной таблицы, данные автоматически удаляются в связанных таблицах.

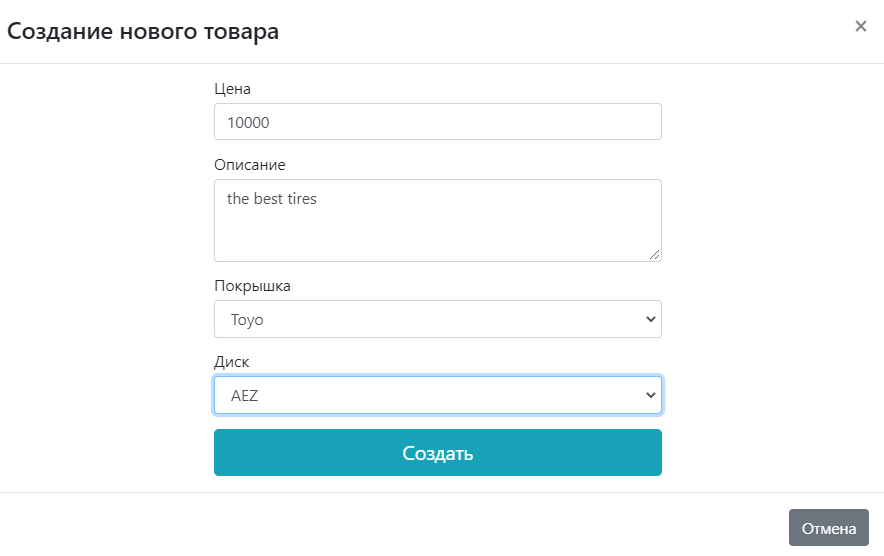
**6.** Была протестирована логика взаимодействия с БД, с помощью проверки данных в БД можно убедиться, что все функции приложения выполняются корректно. Методы, в которых содержатся запросы к БД, возвращают корректные данные, корректно производится запись, создаются корректные связи между сущностями БД.

Результаты работы сайта:

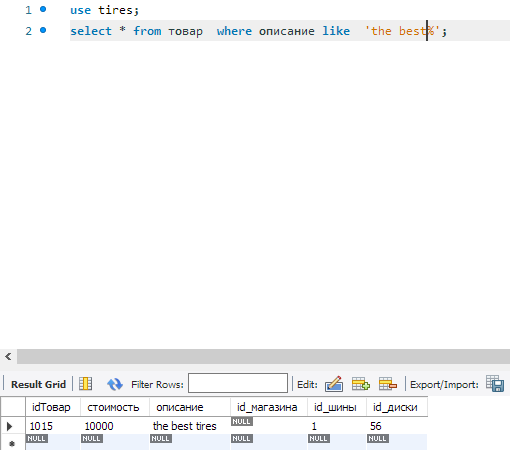


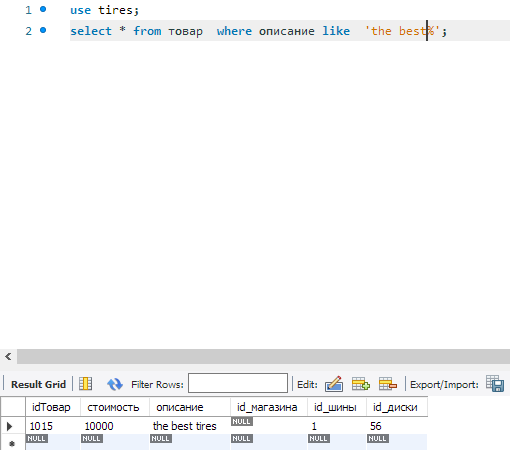




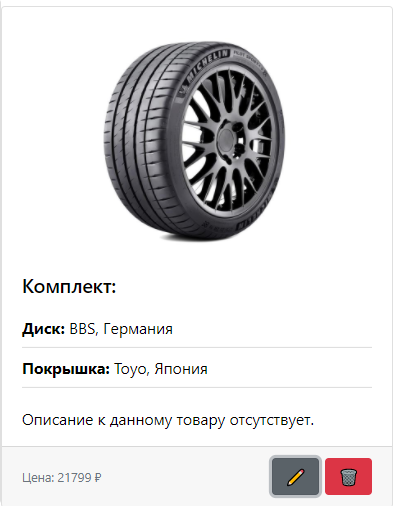
1. Кнопка Создать 

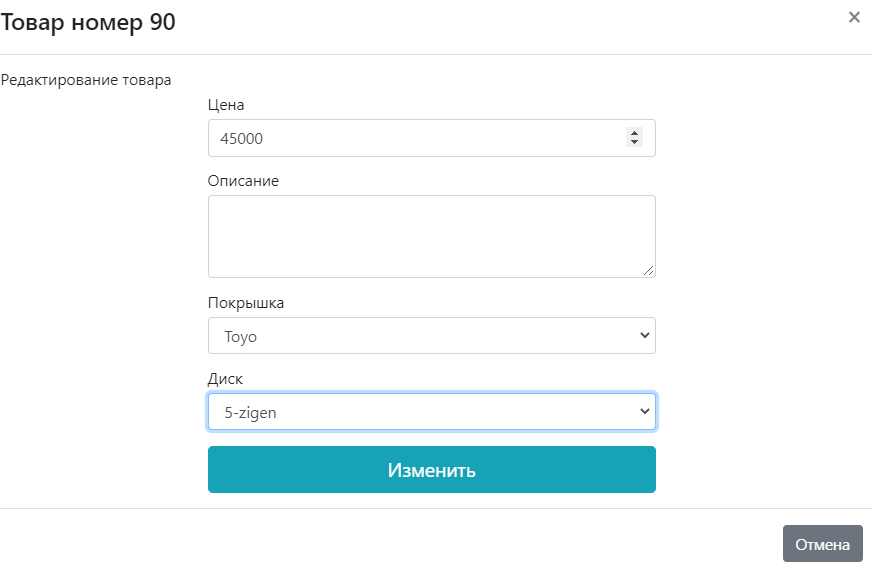
Проверим в бд создание товара

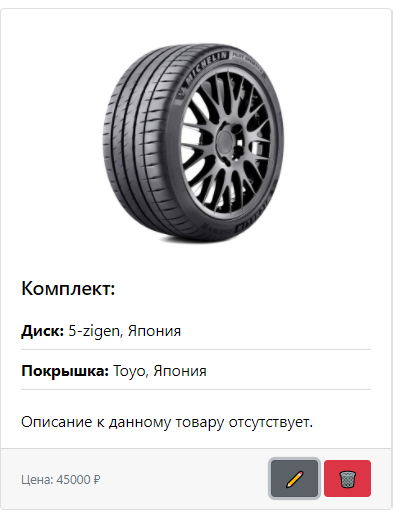




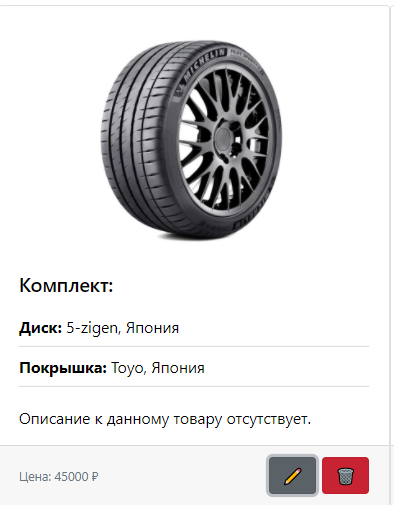
1. Кнопка Изменить







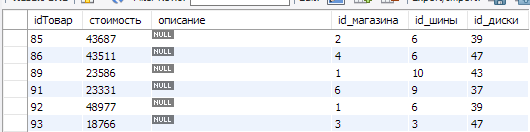
1. Кнопка удалить



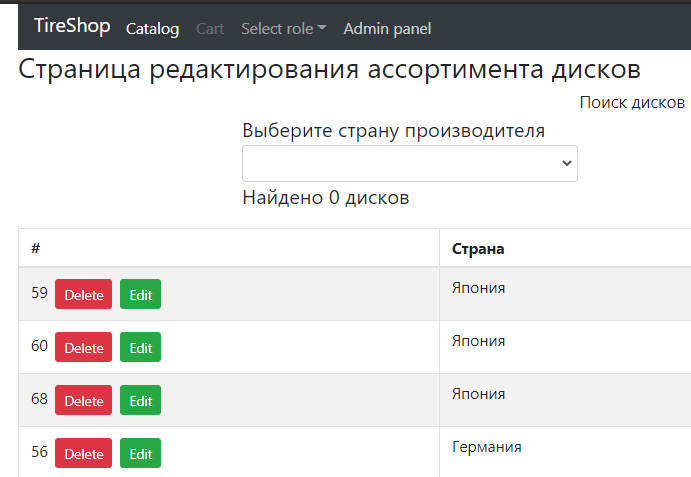
(изображена в виде корзины)

Также удалим товар №90

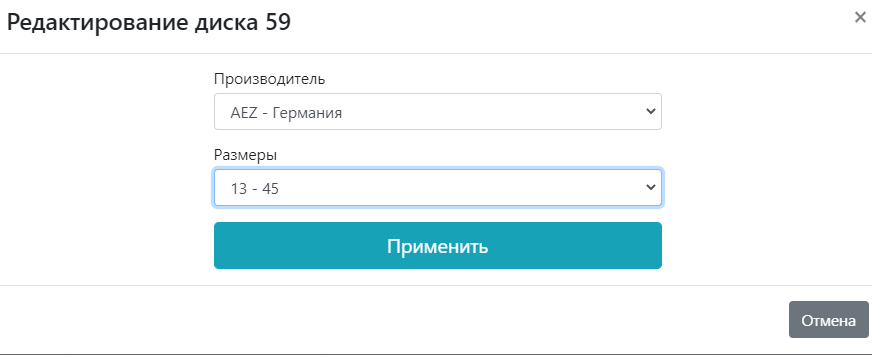
И посмотрим результат в бд



1. Кнопка edit

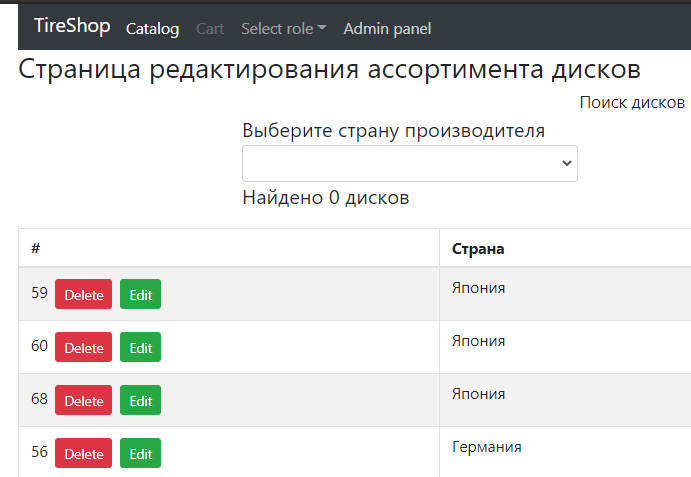






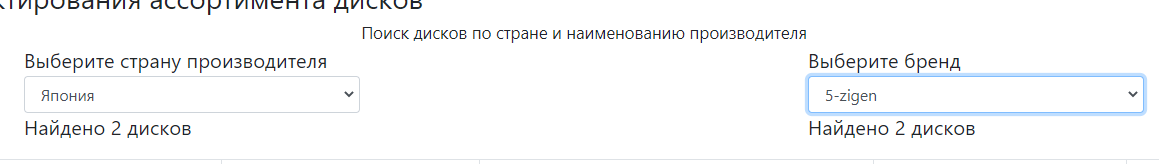


1. Кнопка delete



Удаляет выбранный диск

Аналитические запросы:



**Вывод:**

В ходе работы была разработана программа на node js+react для работы с базой данных на SQL server.

**Заключение:**

В ходе работы была разработана CУБД «Магазин шин и дисков» в среде разработки MySQL Workbench и программа на языке node js+react для непосредственной работы с ней.

# Ссылка на репозиторий:

<https://github.com/ValeriaIvanova/Service_/blob/master/README.md>

# Список литературы

1. Учебно-методическое пособие БД
2. Интернет-статья https://wiki.gentoo.org/wiki/MySQL/Startup\_Guide/ru
3. https://metanit.com/web/nodejs/