  
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**«Дальневосточный федеральный университет»ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ  
Департамент программной инженерии и искусственного интеллекта**

**ОТЧЁТ**по лабораторной работе №2.3  
«Проектирование и разработка базы данных (MySQL)   
в выбранной предметной области»  
по дисциплине «Проектирование и разработка баз данных»  
Специальность 09.03.04 «Программная инженерия»

Выполнили:  
студенты гр. Б9119-09.03.04прогин  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Сазонтова М.Д.

Проверил:  
старший преподаватель  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Краморенко Н.В.

г. Владивосток  
2022

**Оглавление**

[1. Описание предметной области «Каталог запчастей автомобиля» 3](#_Toc106075869)

[2. Построение концептуальной модели 4](#_Toc106075870)

[3. Проектирование и разработка БД 6](#_Toc106075871)

[3.1. Проектирование БД 6](#_Toc106075872)

[3.2. Перечень реализованных функций 6](#_Toc106075873)

[3.3. Выбор среды разработки 8](#_Toc106075874)

[Приложение 1. Прямое проектирование 9](#_Toc106075875)

[Приложение 2. Реализованные функции 15](#_Toc106075876)

# Описание предметной области «Каталог запчастей автомобиля»

Необходимо учесть следующие зависимости между объектами предметной области и их характеристиками:

* В стране может быть несколько марок авто,
* у марки несколько моделей авто,
* у автомобиля множество деталей,
* агрегат (совокупность механизмов машины) состоит из множества узлов, узлы же состоят из деталей,
* некоторые детали подходят автомобилям разных марках,
* деталь может подходить не всем моделям одной марки в зависимости от типа двигателя авто,
* у автомобиля существует три типа двигателя,
* в стране может быть множество фирм-изготовителей делателей,
* фирма-изготовитель может производить несколько деталей,
* авто содержит информацию о необходимой детали и деталь содержит информацию об авто, которым подходит.

Информация будет использоваться сотрудниками автосервиса, поставщиками, работниками магазина автозапчастей.

При работе с системой **сотрудник автосервиса и сотрудник магазина автозапчастей** должен иметь возможность решать следующие задачи:

* Зарегистрироваться в системе,
* Авторизоваться с правами доступа «Сотрудник»,
* редактировать список автомобилей;
* просматривать список всех запчастей, с указанием агрегата
* просматривать список всех запчастей с указанием моделей авто, которым они подходят,
* просматривать список запчастей, соответствующих фильтру: номер кузова, название или номер детали, узел авто, производитель, диапазон цены,
* вносить запчасть в список избранного,

При работе с системой **поставщик** должен иметь возможность решать следующие задачи:

* Зарегистрироваться в системе,
* авторизоваться с правами доступа «Поставщик»,
* просматривать список автомобилей;
* просматривать список всех запчастей, с указанием агрегата
* добавлять запчасть,
* просматривать список всех запчастей с указанием моделей авто, которым они подходят,
* добавлять деталь в избранное,
* просматривать список запчастей, соответствующих фильтру: номер кузова, название или номер детали, узел авто, производитель, диапазон цены,
* просматривать полную или частичную информацию о пользователях каталога.

# Построение концептуальной модели

Необходимо построить такую базу данных, в которой хранится информация о странах, фирмах-изготовителях, брендах и марках автомобиля; агрегатах, узлах, деталях. Пользователи: сотрудники автосервисов, поставщики.

Таким образом, можно выделить следующие сущности с их атрибутами:

* Пользователь (ID, ФИО, номер телефона, mail, адрес, логин, пароль),
* список избранного (ID пользователя, ID детали),
* страна (название),
* фирма-изготовитель (название, страна),
* марка (модель автобуса, номер автобуса),
* модель (расписание маршрута, автобус, дата отправления, цена),
* машина (ID, марка, модель, тип двигателя, год выпуска, тип трансмиссии, привод),
* тип двигателя (ID, мощность, тип топлива, объем),
* деталь для машины (ID машины, ID детали),
* деталь (ID, номер детали, название, цена, изготовитель, ID узла),
* узел (ID, название, агрегат),
* агрегат (название).

Результатом построения концептуальной модели является ER-диаграмма, приведенная на рис. 1.

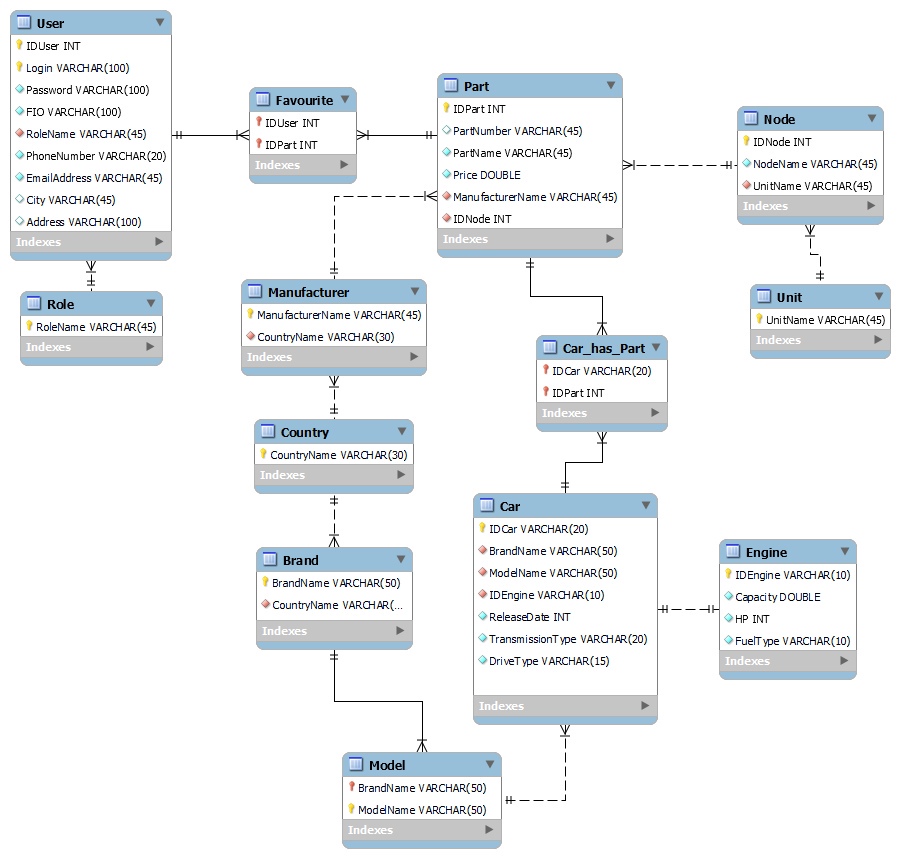


Рис. 1. ER-диаграмма

# Проектирование и разработка БД

# Проектирование БД

Для проектирования базы данных был использован MySQL Workbench и инструмент Forward Engineer, предоставляющий возможность сгенерировать скрипт создания базы данных из ER-диаграммы. Полученный скрипт, сгенерированный на основе созданной схемы (рис. 1), можно найти в приложении 1.

# Перечень реализованных функций

Некоторое описанные далее запросы можно найти в приложении 2.

Запрос 1. Заполнение справочника стран первичными данными.

Запрос 2. Заполнение таблицы моделей первичными данными.

Запрос 3. Заполнение таблицы марок первичными данными.

Запрос 4. Заполнение таблицы изготовителей первичными данными.

Запрос 5. Заполнение справочника агрегатов первичными данными.

Запрос 6. Заполнение таблицы узлов первичными данными.

Запрос 7. Заполнение справочника ролей первичными данными.

Запрос 8. Заполнение таблицы пользователей первичными данными.

Запрос 9. Заполнение таблицы тип двигателя первичными данными.

Запрос 10. Заполнение таблицы машина первичными данными.

Запрос 11. Заполнение таблицы деталь первичными данными.

Запрос 12. Заполнение таблицы деталей машины первичными данными.

Запрос 13. Заполнение таблицы избранное первичными данными.

Запрос14. Выборка всех авто из соответствующей таблицы. Используется для работы с таблицей моделей авто в пользовательском интерфейсе.

Запрос 15. Добавление авто в соответствующую таблицу. Значения полей задаются пользователем в форме.

Запрос 16. Изменение данных авто в соответствующей таблице. Значения полей задаются пользователем в форме, изменяется выбранная им запись.

Запрос 17. Выборка всех запчастей из соответствующей таблицы. Используется для работы с таблицей запчастей в пользовательском интерфейсе.

Запрос 18. Добавление запчасти в соответствующую таблицу. Значения полей задаются пользователем в форме.

Запрос 19. Изменение запчасти в соответствующей таблице. Значения полей задаются пользователем в форме, изменяется выбранная им запись.

Запрос 20. Выборка всех запчастей с привязкой к модели авто из соответствующей таблицы. Используется для работы с таблицей запчастей в пользовательском интерфейсе.

Запрос 21. Выборка запчастей по номеру кузова. Значение поля задается пользователем в форме.

Запрос 16. Выборка запчастей по названию или номеру детали. Значение поля задается пользователем в форме.

Запрос 17. Выборка запчастей по узлу авто. Значение поля задается пользователем в форме.

Запрос 18. Выборка запчастей по производителю. Значение поля задается пользователем в форме.

Запрос 19. Выборка запчастей по диапазону цены. Значение поля задается пользователем в форме.

Запрос 20. Добавление запчасти в таблицу избранное.

Запрос 21. Выборка всех избранных запчастей с соответствующей таблицы. Используется для работы с таблицей избранное в пользовательском интерфейсе.

Запрос 22. Выборка избранных запчастей для каждого пользователя с соответствующей таблицы. Используется для работы с таблицей избранное в пользовательском интерфейсе.

Запрос 23. Удаление запчасти из таблицы избранное.

Запрос 24. Выборка всех пользователей из соответствующей таблицы. Используется для работы с таблицей пользователи в пользовательском интерфейсе.

Запрос 25. Добавление пользователя в соответствующую таблицу. Используется для работы с регистрацией в пользовательском интерфейсе.

# Выбор среды разработки

Интерфейс для работы с базой данный реализован в виде веб-приложения. Такой выбор обосновывается следующими факторами:

* конечный продукт легкодоступен пользователю: в отличие от классического оконного приложения веб-приложение не требует установки,
* кроссплатформенность: для использования приложения нужен только браузер, поэтому оно становится доступно на любой операционной системе, настольной или мобильной.

Система состоит из нескольких частей.

1. База данных MySQL.
2. Frontend (клиент) – клиентская подсистема, предоставляющая браузерный пользовательский интерфейс. Реализован в виде Single Page Application - одностраничный веб-сайт, динамически генерирующий контент, используя устройство пользователя. Для получения и отображения данных выполняет http-запросы на сервер.
3. Backend (сервер) – серверная подсистема, являющаяся прослойкой между базой данных и интерфейсом пользователя. Реализация аутентификации пользователя и передача чувствительных данных сессии в Cookies. Работа с базой данных MySQL организована с помощью PDO.

# Приложение 1. Прямое проектирование

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

DROP SCHEMA IF EXISTS `mydb`;

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;

USE `mydb` ;

-- -----------------------------------------------------

-- Table `mydb`.`Role`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`Role` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Role` (

`RoleName` VARCHAR(45) NOT NULL,

PRIMARY KEY (`RoleName`),

UNIQUE INDEX `RoleName\_UNIQUE` (`RoleName` ASC) VISIBLE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `mydb`.`User`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`User` ;

CREATE TABLE IF NOT EXISTS `mydb`.`User` (

`IDUser` INT NOT NULL,

`FIO` VARCHAR(100) NOT NULL,

`RoleName` VARCHAR(45) NOT NULL,

`PhoneNumber` VARCHAR(20) NOT NULL,

`EmailAddress` VARCHAR(45) NOT NULL,

`City` VARCHAR(45) NULL,

`Address` VARCHAR(100) NULL,

`Login` VARCHAR(20) NULL,

`Password` VARCHAR(20) NULL,

PRIMARY KEY (`IDUser`),

UNIQUE INDEX `IDUser\_UNIQUE` (`IDUser` ASC) VISIBLE,

INDEX `fk\_User\_Role1\_idx` (`RoleName` ASC) VISIBLE,

CONSTRAINT `fk\_User\_Role1`

FOREIGN KEY (`RoleName`)

REFERENCES `mydb`.`Role` (`RoleName`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `mydb`.`Country`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`Country` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Country` (

`CountryName` VARCHAR(30) NOT NULL,

UNIQUE INDEX `CountryName\_UNIQUE` (`CountryName` ASC) VISIBLE,

PRIMARY KEY (`CountryName`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `mydb`.`Manufacturer`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`Manufacturer` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Manufacturer` (

`ManufacturerName` VARCHAR(45) NOT NULL,

`CountryName` VARCHAR(30) NOT NULL,

UNIQUE INDEX `ManufacturerName\_UNIQUE` (`ManufacturerName` ASC) VISIBLE,

PRIMARY KEY (`ManufacturerName`),

INDEX `fk\_Manufacturer\_Country1\_idx` (`CountryName` ASC) VISIBLE,

CONSTRAINT `fk\_Manufacturer\_Country1`

FOREIGN KEY (`CountryName`)

REFERENCES `mydb`.`Country` (`CountryName`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `mydb`.`Unit`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`Unit` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Unit` (

`UnitName` VARCHAR(45) NOT NULL,

UNIQUE INDEX `UnitName\_UNIQUE` (`UnitName` ASC) VISIBLE,

PRIMARY KEY (`UnitName`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `mydb`.`Node`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`Node` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Node` (

`IDNode` INT NOT NULL,

`NodeName` VARCHAR(45) NOT NULL,

`UnitName` VARCHAR(45) NOT NULL,

PRIMARY KEY (`IDNode`),

UNIQUE INDEX `IDNode\_UNIQUE` (`IDNode` ASC) VISIBLE,

INDEX `fk\_Node\_Unit1\_idx` (`UnitName` ASC) VISIBLE,

CONSTRAINT `fk\_Node\_Unit1`

FOREIGN KEY (`UnitName`)

REFERENCES `mydb`.`Unit` (`UnitName`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `mydb`.`Part`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`Part` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Part` (

`IDPart` INT NOT NULL COMMENT 'номер детали',

`PartNumber` VARCHAR(45) NULL,

`PartName` VARCHAR(45) NOT NULL,

`Price` DOUBLE NOT NULL,

`ManufacturerName` VARCHAR(45) NOT NULL,

`IDNode` INT NOT NULL,

PRIMARY KEY (`IDPart`),

UNIQUE INDEX `IDPart\_UNIQUE` (`IDPart` ASC) VISIBLE,

INDEX `fk\_Part\_Manufacturer1\_idx` (`ManufacturerName` ASC) VISIBLE,

INDEX `fk\_Part\_Node1\_idx` (`IDNode` ASC) VISIBLE,

CONSTRAINT `fk\_Part\_Manufacturer1`

FOREIGN KEY (`ManufacturerName`)

REFERENCES `mydb`.`Manufacturer` (`ManufacturerName`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Part\_Node1`

FOREIGN KEY (`IDNode`)

REFERENCES `mydb`.`Node` (`IDNode`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `mydb`.`Engine`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`Engine` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Engine` (

`IDEngine` VARCHAR(10) NOT NULL COMMENT 'название двигателя',

`Capacity` DOUBLE NOT NULL COMMENT 'объем',

`HP` INT NOT NULL COMMENT 'мощность',

`FuelType` VARCHAR(10) NOT NULL,

PRIMARY KEY (`IDEngine`),

UNIQUE INDEX `IDEngine\_UNIQUE` (`IDEngine` ASC) VISIBLE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `mydb`.`Brand`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`Brand` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Brand` (

`BrandName` VARCHAR(50) NOT NULL,

`Country\_CountryName` VARCHAR(30) NOT NULL,

PRIMARY KEY (`BrandName`),

UNIQUE INDEX `BrandName\_UNIQUE` (`BrandName` ASC) VISIBLE,

INDEX `fk\_Brand\_Country1\_idx` (`Country\_CountryName` ASC) VISIBLE,

CONSTRAINT `fk\_Brand\_Country1`

FOREIGN KEY (`Country\_CountryName`)

REFERENCES `mydb`.`Country` (`CountryName`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `mydb`.`Model`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`Model` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Model` (

`BrandName` VARCHAR(50) NOT NULL,

`ModelName` VARCHAR(50) NOT NULL,

PRIMARY KEY (`BrandName`, `ModelName`),

INDEX `fk\_Model\_Brand1\_idx` (`BrandName` ASC) VISIBLE,

UNIQUE INDEX `ModelName\_UNIQUE` (`ModelName` ASC) VISIBLE,

CONSTRAINT `fk\_Model\_Brand1`

FOREIGN KEY (`BrandName`)

REFERENCES `mydb`.`Brand` (`BrandName`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `mydb`.`Car`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`Car` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Car` (

`IDCar` VARCHAR(20) NOT NULL COMMENT 'номер кузова или вин',

`BrandName` VARCHAR(50) NOT NULL,

`ModelName` VARCHAR(50) NOT NULL,

`IDEngine` VARCHAR(10) NOT NULL,

`ReleaseDate` INT NOT NULL COMMENT 'дата релиза',

`TransmissionType` VARCHAR(20) NOT NULL,

`DriveType` VARCHAR(15) NOT NULL,

PRIMARY KEY (`IDCar`),

INDEX `fk\_CarData\_Engine1\_idx` (`IDEngine` ASC) VISIBLE,

UNIQUE INDEX `IDCar\_UNIQUE` (`IDCar` ASC) VISIBLE,

INDEX `fk\_Car\_Model1\_idx` (`ModelName` ASC, `BrandName` ASC) VISIBLE,

CONSTRAINT `fk\_CarData\_Engine1`

FOREIGN KEY (`IDEngine`)

REFERENCES `mydb`.`Engine` (`IDEngine`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Car\_Model1`

FOREIGN KEY (`ModelName` , `BrandName`)

REFERENCES `mydb`.`Model` (`ModelName` , `BrandName`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `mydb`.`Favourite`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`Favourite` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Favourite` (

`IDUser` INT NOT NULL,

`IDPart` INT NOT NULL,

PRIMARY KEY (`IDUser`, `IDPart`),

INDEX `fk\_Favourite\_Part1\_idx` (`IDPart` ASC) VISIBLE,

CONSTRAINT `fk\_Favourite\_User1`

FOREIGN KEY (`IDUser`)

REFERENCES `mydb`.`User` (`IDUser`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Favourite\_Part1`

FOREIGN KEY (`IDPart`)

REFERENCES `mydb`.`Part` (`IDPart`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `mydb`.`Car\_has\_Part`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`Car\_has\_Part` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Car\_has\_Part` (

`IDCar` VARCHAR(20) NOT NULL,

`IDPart` INT NOT NULL,

PRIMARY KEY (`IDCar`, `IDPart`),

INDEX `fk\_Car\_has\_Part\_Part1\_idx` (`IDPart` ASC) VISIBLE,

INDEX `fk\_Car\_has\_Part\_Car1\_idx` (`IDCar` ASC) VISIBLE,

CONSTRAINT `fk\_Car\_has\_Part\_Car1`

FOREIGN KEY (`IDCar`)

REFERENCES `mydb`.`Car` (`IDCar`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Car\_has\_Part\_Part1`

FOREIGN KEY (`IDPart`)

REFERENCES `mydb`.`Part` (`IDPart`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

# Приложение 2. Реализованные функции

Примечание. В фигурных скобках указаны динамические данные, т.е. данные, получаемые в ходе использования системы пользователем.

Запрос 1.