**Часть 1. Создание БД.**

* 1. **Создания EDR-модели**

В процессе проектирования ERD модели БД были выделены следующие сущности:

foremens, destinations, brigadiers, brigades, masters, landchiefs,

service, transport, driver, transport\_type.

Описание сущностей:

foremens – хранит информацию о начальниках цеха.

destinations – хранит информацию о маршрутах.

brigadiers – содержит информацию о бригадирах.

brigades – содержит информацию о бригадах.

masters – содержит информацию о мастерах.

landchiefs - содержит информацию о начальниках участков.

service - содержит информацию о работниках сервиса.

transport - содержит информацию о транспорте.

driver - содержит информацию о водителях.

transport\_type - содержит информацию о типах транспорта.

Описание атрибутов сущностей:

|  |  |  |  |
| --- | --- | --- | --- |
| **brigades** | | | |
| Поле | Тип | Атрибуты | Описание |
| brigade\_ID | INT | PK NN | Уникальный номер бригады |
| brigade\_Name | VARCHAR(45) | NN | Название бригады |

|  |  |  |  |
| --- | --- | --- | --- |
| **brigadiers** | | | |
| Поле | Тип | Атрибуты | Описание |
| brigadier\_ID | INT | PK NN | Уникальный номер бригадира |
| brigadier\_Name | VARCHAR(45) | NN | Название бригады |
| brigades\_brigade\_ID | INT | PK NN | Уникальный номер бригады |

|  |  |  |  |
| --- | --- | --- | --- |
| **destinations** | | | |
| Поле | Тип | Атрибуты | Описание |
| destination\_ID | INT | PK NN | Уникальный номер маршрута |
| destination\_departure | VARCHAR(45) | NN | Отправная точка маршрута |
| destination\_terminal | VARCHAR(45) | NN | Конечная точка маршрута |
| destination\_code | INT | NN | Код маршрута |

|  |  |  |  |
| --- | --- | --- | --- |
| **driver** | | | |
| Поле | Тип | Атрибуты | Описание |
| driver\_ID | INT | PK NN | Уникальный номер водителя |
| driver\_Name | VARCHAR(45) | NN | Имя водителя |
| Brigades\_brigade\_ID | INT | PK NN | Уникальный номер бригады |

|  |  |  |  |
| --- | --- | --- | --- |
| **foremens** | | | |
| Поле | Тип | Атрибуты | Описание |
| foremen\_ID | INT | PK NN | Уникальный номер начальника цеха |
| foremen\_Name | VARCHAR(45) | NN | Имя начальника цеха |
| masters\_master\_ID | INT | PK NN | Уникальный номер мастера |

|  |  |  |  |
| --- | --- | --- | --- |
| **landchiefs** | | | |
| Поле | Тип | Атрибуты | Описание |
| landchief\_ID | INT | PK NN | Уникальный номер начальника участка |
| landchief\_Name | VARCHAR(45) | NN | Имя начальника участка |
| foremens\_foremen\_ID | INT | PK NN | Уникальный номер начальника цеха |

|  |  |  |  |
| --- | --- | --- | --- |
| **masters** | | | |
| Поле | Тип | Атрибуты | Описание |
| master\_ID | INT | PK NN | Уникальный номер мастера |
| master\_Name | VARCHAR(45) | NN | Имя мастера |
| brigadiers\_brigadier\_ID | INT | PK NN | Уникальный номер бригадира |

|  |  |  |  |
| --- | --- | --- | --- |
| **service** | | | |
| Поле | Тип | Атрибуты | Описание |
| service\_ID | INT | PK NN | Уникальный номер работника сервиса |
| service\_Name | VARCHAR(45) | NN | Имя работника сервиса |
| brigades\_brigade\_ID | INT | PK NN | Уникальный номер бригады |

|  |  |  |  |
| --- | --- | --- | --- |
| **transport** | | | |
| Поле | Тип | Атрибуты | Описание |
| transport\_ID | INT | PK NN | Уникальный номер транспорта |
| transport\_probeg | INT | PK NN | Пробег транспорта |
| transport\_date\_of\_receipt | DATE |  | Дата получения |
| transport\_date\_of\_deduction | DATE |  | Дата списания |
| transport\_brand | VARCHAR(45) | NN | Марка транспорта |
| driver\_driver\_ID | INT | PK NN | Уникальный номер водителя |
| transport\_type\_transport\_typeID | INT | PK NN | Уникальный номер типа транспорта |
| transport\_destination\_ID | INT | PK NN | Уникальный номер маршрута |

|  |  |  |  |
| --- | --- | --- | --- |
| transport\_type | | | |
| Поле | Тип | Атрибуты | Описание |
| transport\_typeID | INT | PK NN | Уникальный номер типа транспорта |
| transport\_typeName | VARCHAR(45) | NN | Название типа транспорта |
| transport\_capacity | INT | NN | Вместимость типа транспорта |

Для соединения таблиц использовались следующие типы связей:

 - один ко многим вида: **identifying relationship**

**Данный вид связи свидетельствует о том, что данные подчинённой таблицы не могут существовать без связи с главной таблицей.**

**- один к одному вида: identifying relationship**

**После того как были установлены все отношения между сущностями получили следующую модель данных: см. Приложение 1.**

На основе полученной модели была создана БД средствами мастера. По итогу работы мастера был получен следующий SQL код: см. Приложение 2.

**1.2 Заполнение БД**

Следующим шагом необходимо заполнить БД:

Таблица brigades

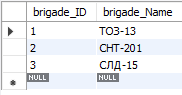


Таблица brigadiers

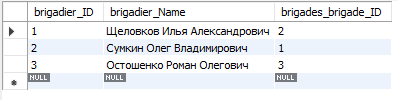


Таблица destinations

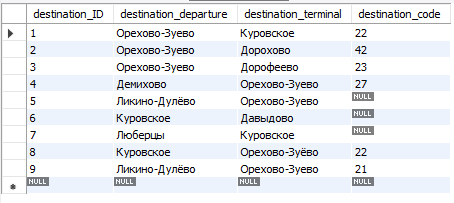


Таблица driver

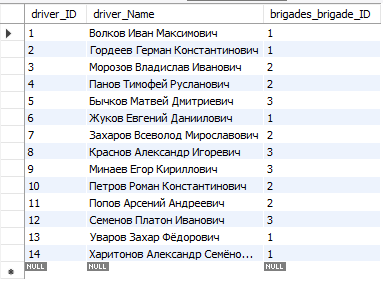


Таблица foremens

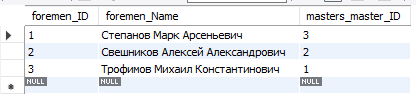


Таблица landchiefs

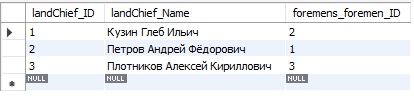


Таблица masters

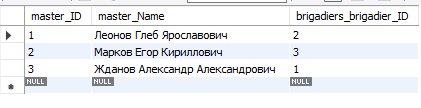


Таблица service

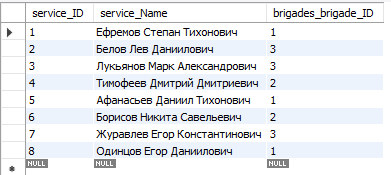


Таблица transport

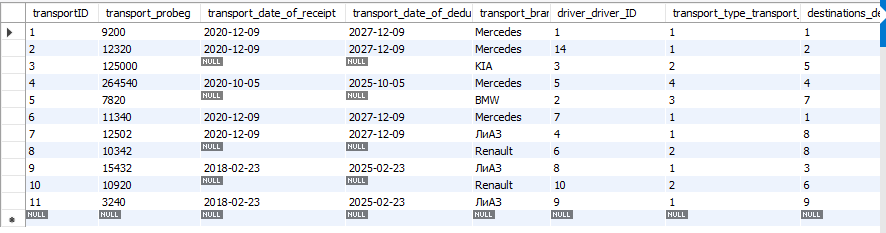
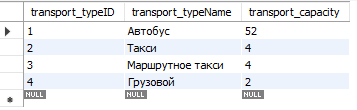


Таблица transport\_type



При заполнении таблиц очень важна последовательность: сперва заполняются главные таблицы, затем подчиненные.

**Часть 2. Выборка и операции с данными.**

В качестве примера работы с данными были написаны запросы различного вида.

**Выборка и выборка с условием**

1. Выбрать все записи из таблицы Водители:

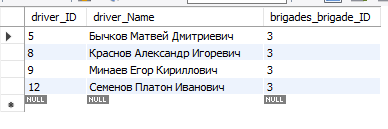
SELECT \* FROM driver;



1. выбрать данные из одной таблицы по условию WHERE

Выбрать все записи водителей, которые находятся в бригаде №3

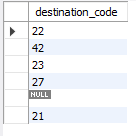
SELECT \* FROM driver WHERE brigades\_brigade\_ID = 3;



1. выбрать данные с применением команды DISTINCT

Выбрать уникальные не повторяющиеся коды маршрутов.

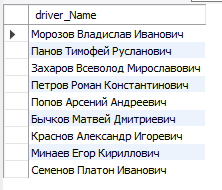
SELECT DISTINCT destination\_code FROM destinations;



1. выбрать данные из таблицы по условию WHERE с применением оператора вхождения в коллекцию – IN

Вывести имена водителей, состоящих в бригаде №2 и №3.

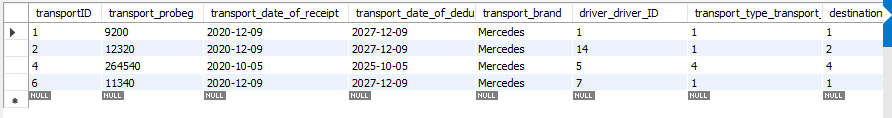
SELECT driver\_Name FROM driver WHERE brigades\_brigade\_ID IN (2, 3);



1. выбрать данные из таблицы по условию WHERE с использованием оператора Like

Вывести информацию о транспорте, марка которого Mercedes

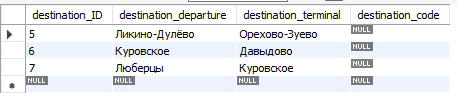
select \* from transport where transport\_brand like 'Mercedes';



1. выберите данные из таблицы с применением проверки данных на NULL (IS NULL, IS NOT NULL)

Вывести информацию о маршрутах у которых нет кода маршрута.

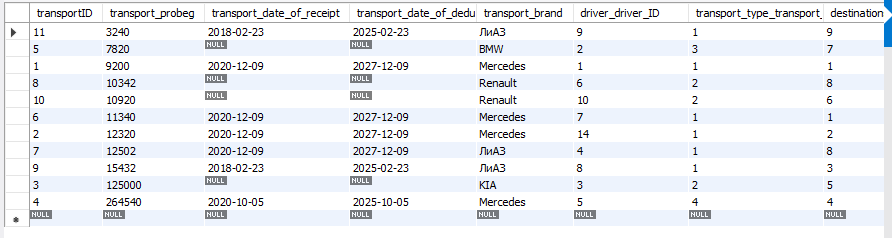
select \* from destinations where destination\_code is null;



1. выберите данные из таблицы с применением сортировки по выбранным полям

Вывести информацию о транспорте с сортировкой по пробегу.

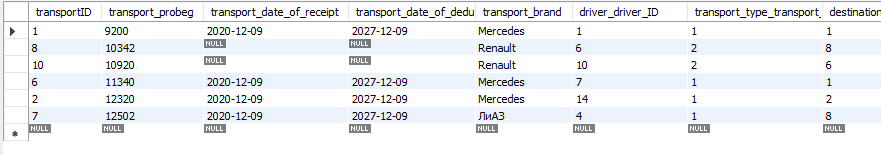
select \* from transport order by transport\_probeg;



1. выбрать данные с применением оператора BETWEEN

Вывести информацию о транспорте с пробегом между 9000 и 13000, отсортировав по пробегу

select \* from transport where transport\_probeg between 9000 and 13000 order by transport\_probeg;



**Агрегатные функции**

Напишите пример использования агрегатных функций, если это возможно для вашей структуры данных:

1. Avg

Вывести среднее арифметическое значение пробега между всем транспортом.

select avg(transport\_probeg) from transport;



1. Sum

Вывести сумму пробега всего транспорта.

select sum(transport\_probeg) from transport;



1. Min

Вывести минимальное значение пробега из всего транспорта

select min(transport\_probeg) from transport;



1. Max

Вывести максимальное значение пробега из всего транспорта

select max(transport\_probeg) from transport;



1. Count

Вывести количество транспорта, которое было получено 2018-02-23

select count(transport\_date\_of\_receipt) from transport where transport\_date\_of\_receipt = '2018-02-23';

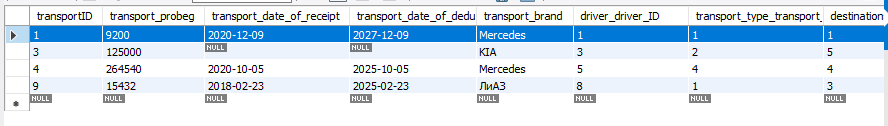


**Группировка**

1. Напишите пример запроса с применением группировки GROUP BY

Вывести данные о транспорте группируя по дате получения

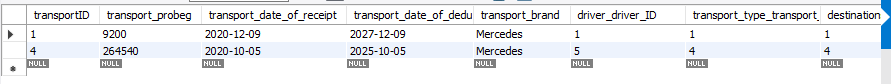
select \* from transport group by transport\_date\_of\_receipt;



1. Напишите пример запроса с применением группировки GROUP BY с использованием условия HAVING

Вывести данные о транспорте с маркой Mercedes и группируя его по дате получения

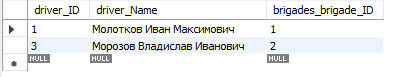
select \* from transport group by transport\_date\_of\_receipt having transport\_brand like 'Mercedes';



1. Напишите пример запроса с применением группировки GROUP BY с использованием условия HAVING

Сделать группировку водителей по номеру бригады, имя которых начинается с буквы М

select \* from driver group by brigades\_brigade\_ID having driver\_Name like 'М%';

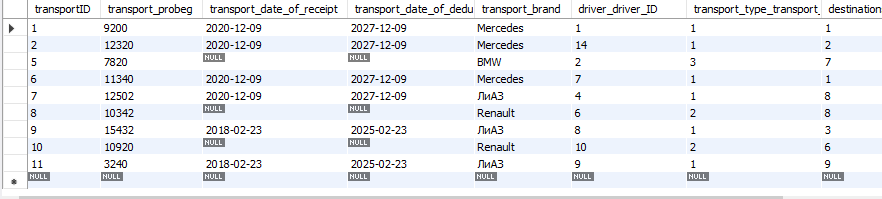


**Подзапросы**

1. Напишите пример использования подзапросов

Вывести транспорт пробег которого меньше среднего арифметического значения пробега всего транспорта

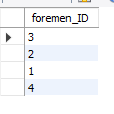
SELECT \* FROM transport WHERE transport\_probeg < (SELECT AVG(transport\_probeg) FROM transport)



**Соединение таблиц**

1. Напишите запрос с внутренним соединением таблиц (Inner Join)
2. Напишите запрос с внешним соединением таблиц (Left | Right Join)
3. Напишите запрос объединения таблиц (Union)

select foremen\_ID from foremens union select master\_ID from masters;



**Напишите запросы на работу с данными БД:**

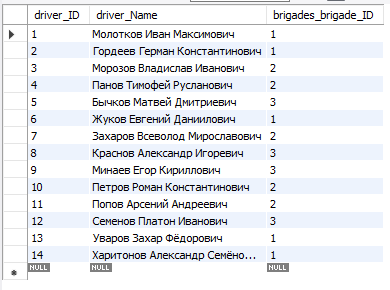
1. Запрос на добавление

insert into driver (driver\_ID, driver\_Name, brigades\_brigade\_ID) values (15, 'Созонов Алексей Юрьевич', 2);



1. Запрос на удаление

delete from driver where driver\_ID = 15;

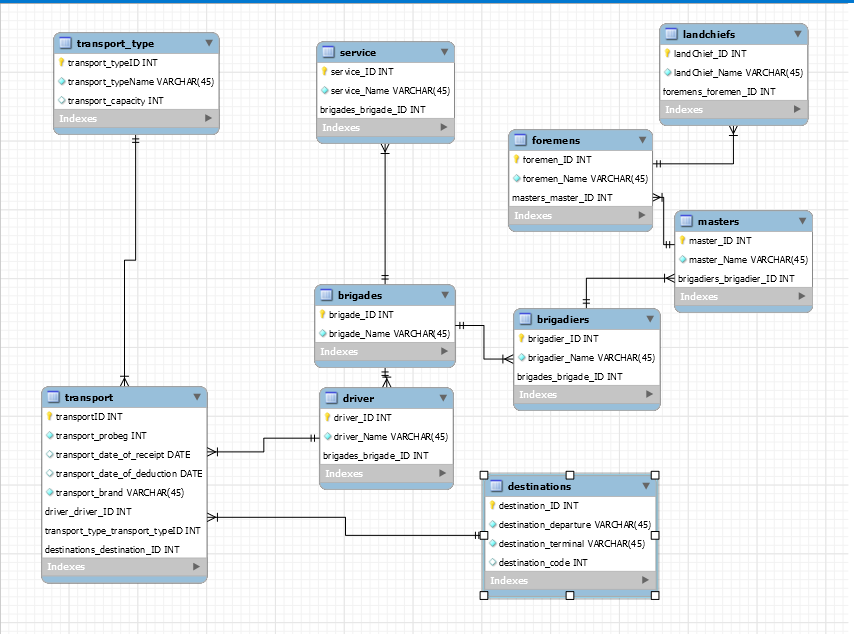


1. Запрос на изменение

update driver set driver\_Name = 'Минюскин Дмитрий Александрович', brigades\_brigade\_ID = '3' where driver\_ID = 15;



**Приложение 1.** Модель БД «Продажа»



**Приложение 2.** SQL код создания БД

**Приложение 2.** SQL код создания БД

-- 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 autoenterprise

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

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

-- Schema autoenterprise

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

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

USE `autoenterprise` ;

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

-- Table `autoenterprise`.`brigades`

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

CREATE TABLE IF NOT EXISTS `autoenterprise`.`brigades` (

`brigade\_ID` INT NOT NULL,

`brigade\_Name` VARCHAR(45) NOT NULL,

PRIMARY KEY (`brigade\_ID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

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

-- Table `autoenterprise`.`brigadiers`

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

CREATE TABLE IF NOT EXISTS `autoenterprise`.`brigadiers` (

`brigadier\_ID` INT NOT NULL,

`brigadier\_Name` VARCHAR(45) NOT NULL,

`brigades\_brigade\_ID` INT NOT NULL,

PRIMARY KEY (`brigadier\_ID`, `brigades\_brigade\_ID`),

INDEX `fk\_brigadiers\_brigades1\_idx` (`brigades\_brigade\_ID` ASC) VISIBLE,

CONSTRAINT `fk\_brigadiers\_brigades1`

FOREIGN KEY (`brigades\_brigade\_ID`)

REFERENCES `autoenterprise`.`brigades` (`brigade\_ID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

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

-- Table `autoenterprise`.`destinations`

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

CREATE TABLE IF NOT EXISTS `autoenterprise`.`destinations` (

`destination\_ID` INT NOT NULL,

`destination\_departure` VARCHAR(45) NOT NULL,

`destination\_terminal` VARCHAR(45) NOT NULL,

`destination\_code` INT NULL DEFAULT NULL,

PRIMARY KEY (`destination\_ID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

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

-- Table `autoenterprise`.`driver`

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

CREATE TABLE IF NOT EXISTS `autoenterprise`.`driver` (

`driver\_ID` INT NOT NULL,

`driver\_Name` VARCHAR(45) NOT NULL,

`brigades\_brigade\_ID` INT NOT NULL,

PRIMARY KEY (`driver\_ID`, `brigades\_brigade\_ID`),

INDEX `fk\_driver\_brigades1\_idx` (`brigades\_brigade\_ID` ASC) VISIBLE,

CONSTRAINT `fk\_driver\_brigades1`

FOREIGN KEY (`brigades\_brigade\_ID`)

REFERENCES `autoenterprise`.`brigades` (`brigade\_ID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

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

-- Table `autoenterprise`.`masters`

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

CREATE TABLE IF NOT EXISTS `autoenterprise`.`masters` (

`master\_ID` INT NOT NULL,

`master\_Name` VARCHAR(45) NOT NULL,

`brigadiers\_brigadier\_ID` INT NOT NULL,

PRIMARY KEY (`master\_ID`, `brigadiers\_brigadier\_ID`),

INDEX `fk\_masters\_brigadiers1\_idx` (`brigadiers\_brigadier\_ID` ASC) VISIBLE,

CONSTRAINT `fk\_masters\_brigadiers1`

FOREIGN KEY (`brigadiers\_brigadier\_ID`)

REFERENCES `autoenterprise`.`brigadiers` (`brigadier\_ID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

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

-- Table `autoenterprise`.`foremens`

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

CREATE TABLE IF NOT EXISTS `autoenterprise`.`foremens` (

`foremen\_ID` INT NOT NULL,

`foremen\_Name` VARCHAR(45) NOT NULL,

`masters\_master\_ID` INT NOT NULL,

PRIMARY KEY (`foremen\_ID`, `masters\_master\_ID`),

INDEX `fk\_foremens\_masters1\_idx` (`masters\_master\_ID` ASC) VISIBLE,

CONSTRAINT `fk\_foremens\_masters1`

FOREIGN KEY (`masters\_master\_ID`)

REFERENCES `autoenterprise`.`masters` (`master\_ID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

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

-- Table `autoenterprise`.`landchiefs`

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

CREATE TABLE IF NOT EXISTS `autoenterprise`.`landchiefs` (

`landChief\_ID` INT NOT NULL,

`landChief\_Name` VARCHAR(45) NOT NULL,

`foremens\_foremen\_ID` INT NOT NULL,

PRIMARY KEY (`landChief\_ID`, `foremens\_foremen\_ID`),

INDEX `fk\_landchiefs\_foremens1\_idx` (`foremens\_foremen\_ID` ASC) VISIBLE,

CONSTRAINT `fk\_landchiefs\_foremens1`

FOREIGN KEY (`foremens\_foremen\_ID`)

REFERENCES `autoenterprise`.`foremens` (`foremen\_ID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

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

-- Table `autoenterprise`.`service`

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

CREATE TABLE IF NOT EXISTS `autoenterprise`.`service` (

`service\_ID` INT NOT NULL,

`service\_Name` VARCHAR(45) NOT NULL,

`brigades\_brigade\_ID` INT NOT NULL,

PRIMARY KEY (`service\_ID`, `brigades\_brigade\_ID`),

INDEX `fk\_service\_brigades1\_idx` (`brigades\_brigade\_ID` ASC) VISIBLE,

CONSTRAINT `fk\_service\_brigades1`

FOREIGN KEY (`brigades\_brigade\_ID`)

REFERENCES `autoenterprise`.`brigades` (`brigade\_ID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

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

-- Table `autoenterprise`.`transport\_type`

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

CREATE TABLE IF NOT EXISTS `autoenterprise`.`transport\_type` (

`transport\_typeID` INT NOT NULL,

`transport\_typeName` VARCHAR(45) NOT NULL,

`transport\_capacity` INT NULL DEFAULT NULL,

PRIMARY KEY (`transport\_typeID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

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

-- Table `autoenterprise`.`transport`

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

CREATE TABLE IF NOT EXISTS `autoenterprise`.`transport` (

`transportID` INT NOT NULL,

`transport\_probeg` INT NOT NULL,

`transport\_date\_of\_receipt` DATE NULL DEFAULT NULL,

`transport\_date\_of\_deduction` DATE NULL DEFAULT NULL,

`transport\_brand` VARCHAR(45) NOT NULL,

`driver\_driver\_ID` INT NOT NULL,

`transport\_type\_transport\_typeID` INT NOT NULL,

`destinations\_destination\_ID` INT NOT NULL,

PRIMARY KEY (`transportID`, `driver\_driver\_ID`, `transport\_type\_transport\_typeID`, `destinations\_destination\_ID`),

INDEX `fk\_transport\_driver\_idx` (`driver\_driver\_ID` ASC) VISIBLE,

INDEX `fk\_transport\_transport\_type1\_idx` (`transport\_type\_transport\_typeID` ASC) VISIBLE,

INDEX `fk\_transport\_destinations1\_idx` (`destinations\_destination\_ID` ASC) VISIBLE,

CONSTRAINT `fk\_transport\_destinations1`

FOREIGN KEY (`destinations\_destination\_ID`)

REFERENCES `autoenterprise`.`destinations` (`destination\_ID`),

CONSTRAINT `fk\_transport\_driver`

FOREIGN KEY (`driver\_driver\_ID`)

REFERENCES `autoenterprise`.`driver` (`driver\_ID`),

CONSTRAINT `fk\_transport\_transport\_type1`

FOREIGN KEY (`transport\_type\_transport\_typeID`)

REFERENCES `autoenterprise`.`transport\_type` (`transport\_typeID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

USE `autoenterprise` ;

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

-- procedure allBrigades

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

DELIMITER $$

USE `autoenterprise`$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `allBrigades`()

BEGIN

select \* from brigades;

END$$

DELIMITER ;

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

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

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