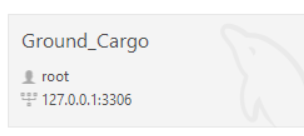
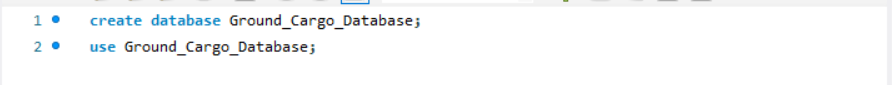
ЭТАПЫ ВЫПОЛНЕНИЯ

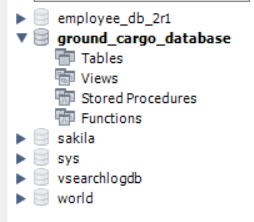
**Этапы выполнения:**

1. Создание подключения к серверу;



1. Создание файла БД;

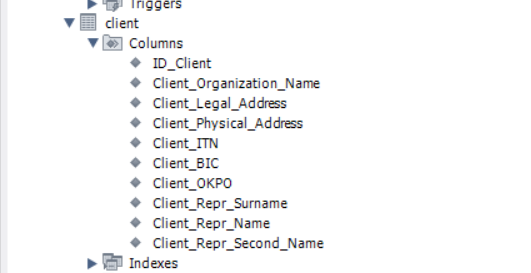
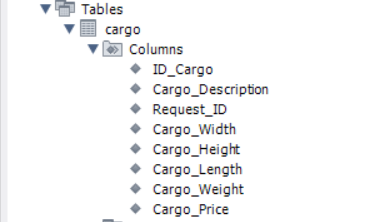


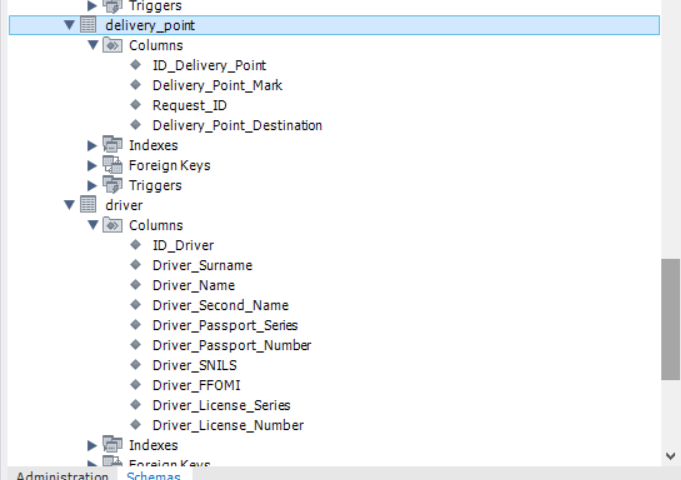
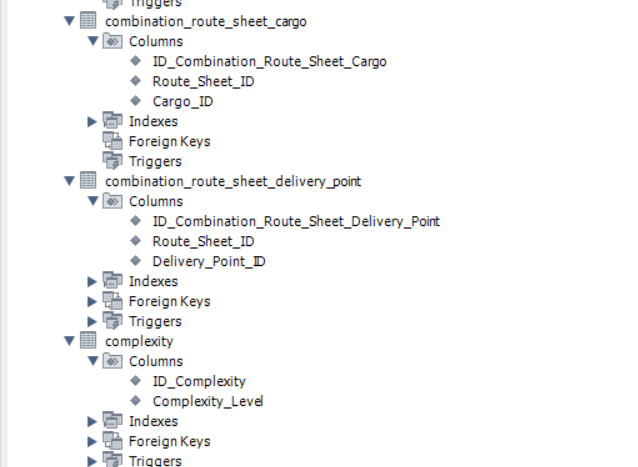


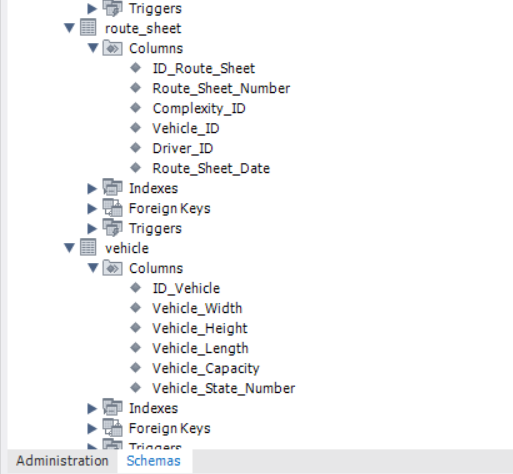
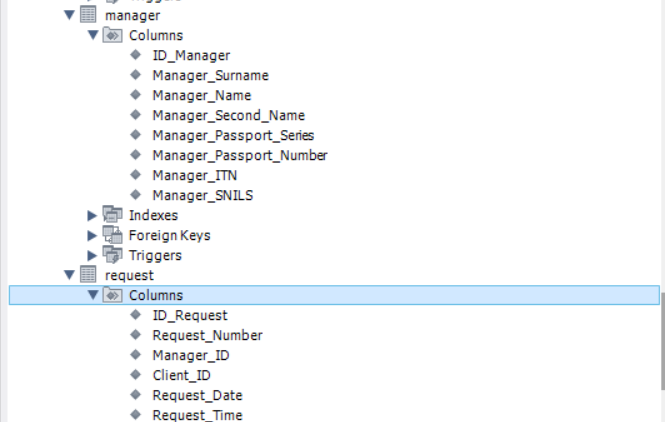
1. Реализация таблиц в соответствии со словарём данных;

| Ключ | Наименование | Тип данных | Примечание | Скрипт MySQL |
| --- | --- | --- | --- | --- |
| Client | | | | |
| PK | ID\_Client | Int | Первичный ключ таблицы Заказчик | CREATE TABLE Client(      ID\_Client int not null auto\_increment PRIMARY KEY,      Client\_Organization\_Name varchar(100) not null,      Client\_Legal\_Address varchar(100) null DEFAULT '-',      Client\_Physical\_Address varchar(100) not null,      Client\_ITN varchar(11) not null UNIQUE constraint CH\_Client\_ITN           CHECK(regexp\_like(Client\_ITN, '[0-9]{10}')),      Client\_BIC varchar(10) not null UNIQUE constraint CH\_Client\_BIC          CHECK(regexp\_like(Client\_BIC, '[0-9]{9}')) ,      Client\_OKPO varchar(11) not null UNIQUE constraint CH\_Client\_OKPO          CHECK(regexp\_like(Client\_OKPO, '[0-9]{8}|[0-9]{10}')) ,      Client\_Repr\_Surname varchar(30) not null,      Client\_Repr\_Name varchar(30) not null,      Client\_Repr\_Second\_Name varchar(30) null DEFAULT '-'  ); |
|  | Client\_Organization\_Name | Varchar(100) |  |
|  | Client\_Legal\_Address | Varchar(100) | Null |
|  | Client\_Physical\_Address | Varchar(100) |  |
|  | Client\_ITN | Varchar(10) | Уникальное поле  0000000000 |
|  | Client\_BIC | Varchar(9) | Уникальное поле 000000000 |
|  | Client\_OKPO | Varchar(10) | Уникальное поле 00000000  Или  0000000000 |
|  | Client\_Repr\_Surname | Varchar(30) |  |
|  | Client\_Repr\_Name | Varchar(30) |  |
|  | Client\_Repr\_Second\_Name | Varchar(30) | Null |
| Manager | | | | |
| PK | ID\_Manager | Int | Первичный ключ таблицы менеджер по обработке заявок | CREATE TABLE Manager(      ID\_Manager int not null auto\_increment PRIMARY KEY,      Manager\_Surname varchar(30) not null,      Manager\_Name varchar(30) not null,      Manager\_Second\_Name varchar(30) null DEFAULT '-',      Manager\_Passport\_Series varchar(4) not null constraint CH\_Manager\_Passport\_Series          CHECK(regexp\_like(Manager\_Passport\_Series, '[0-9]{4}')),      Manager\_Passport\_Number varchar(6) not null constraint CH\_Manager\_Passport\_Number          CHECK(regexp\_like(Manager\_Passport\_Number, '[0-9]{6}')),      Manager\_ITN varchar(10) not null UNIQUE constraint CH\_Manager\_ITN          CHECK(regexp\_like(Manager\_ITN, '[0-9]{10}')),      Manager\_SNILS varchar(14) not null UNIQUE constraint CH\_Manager\_SNILS      CHECK(regexp\_like(Manager\_SNILS, '[0-9]{3}\\-[0-9]{3}\\-[0-9]{3} [0-9][0-9]'))  ); |
|  | Manager\_Surname | Varchar(30) |  |
|  | Manager\_Name | Varchar(30) |  |
|  | Manager\_Second\_Name | Varchar(30) | Null |
|  | Manager\_Passport\_Series | Varchar(4) | 0000 |
|  | Manager\_Passport\_Number | Varchar(6) | 000000 |
|  | Manager\_ITN | Varchar(10) | Уникальное поле  0000000000 |
|  | Manager\_SNILS | Varchar(14) | Уникальное поле  000-000-000 00 |
| Driver | | | | |
| PK | ID\_Driver | Int | Первичный ключ таблицы водитель | CREATE TABLE Driver(      ID\_Driver int not null auto\_increment PRIMARY KEY,      Driver\_Surname varchar(30) not null,      Driver\_Name varchar(30) not null,      Driver\_Second\_Name varchar(30) null DEFAULT '-',      Driver\_Passport\_Series varchar(4) not null constraint CH\_Driver\_Passport\_Series          CHECK(regexp\_like(Driver\_Passport\_Series, '[0-9]{4}')),      Driver\_Passport\_Number varchar(6) not null constraint CH\_Driver\_Passport\_Number      CHECK(regexp\_like(Driver\_Passport\_Number, '[0-9]{6}')),      Driver\_SNILS varchar(14) not null UNIQUE constraint CH\_Driver\_SNILS      CHECK(regexp\_like(Driver\_SNILS, '[0-9]{3}\\-[0-9]{3}\\-[0-9]{3} [0-9][0-9]')),      Driver\_FFOMI varchar(16) not null UNIQUE constraint CH\_Driver\_FFOMI      CHECK(regexp\_like(Driver\_FFOMI, '[0-9]{16}')),      Driver\_License\_Series varchar(4) not null constraint CH\_Driver\_License\_Series      CHECK(regexp\_like(Driver\_License\_Series, '[0-9]{4}')),      Driver\_License\_Number varchar(6) not null constraint CH\_Driver\_License\_Number      CHECK(regexp\_like(Driver\_License\_Number, '[0-9]{6}'))  ); |
|  | Driver\_Surname | Varchar(30) |  |
|  | Driver\_Name | Varchar(30) |  |
|  | Driver\_Second\_Name | Varchar(30) | NULL |
|  | Driver\_Passport\_Series | Varchar(4) | 0000 |
|  | Driver\_Passport\_Number | Varchar(6) | 000000 |
|  | Driver\_SNILS | Varchar(14) | Уникальное поле  000-000-000 00 |
|  | Driver\_FFOMI | Varchar(16) | Уникальное поле  0000000000000000 |
|  | Driver\_License\_Series | Varchar(4) | 0000 |
|  | Driver\_License\_Number | Varchar(6) | 000000 |
| Vehicle | | | | |
| PK | ID\_Vehicle | Int | Первичный ключ таблицы Автотранспорт | CREATE TABLE Vehicle  (              ID\_Vehicle int not null auto\_increment PRIMARY KEY,              Vehicle\_Width int not null,              Vehicle\_Height int not null,          Vehicle\_Length int not null,          Vehicle\_Capacity int not null,          Vehicle\_State\_Number varchar(10) not null constraint CH\_Vehicle\_State\_Number      CHECK(regexp\_like(Vehicle\_State\_Number, '[А-Я]{2}[0-9]{3}[А-Я] [0-9]{2,3}'))  ); |
|  | Vehicle\_Width | Int | Ширина в сантиметрах |
|  | Vehicle\_Height | Int | Высота в сантиметрах |
|  | Vehicle\_Length | Int | Длина в сантиметрах |
|  | Vehicle\_Capacity | Int | Грузоподъёмность в килограммах |
|  | Vechicle\_State\_Number | Varchar(10) | AA000A 999 |
| Complexity | | | | |
| PK | ID\_Complexity | Int | Первичный ключ таблицы сложность | CREATE TABLE Complexity  (              ID\_Complexity int not null auto\_increment PRIMARY KEY,              Complexity\_Level int not null UNIQUE  ); |
|  | Complexity\_Level | Int | Значения – 1,2,3 |
| Request | | | | |
| PK | ID\_Request | Int | Первичный ключ таблицы Заявка | CREATE TABLE Request(      ID\_Request int not null auto\_increment PRIMARY KEY,      Request\_Number int not null,      Manager\_ID int not null REFERENCES Manager (ID\_Manager),      Client\_ID int not null REFERENCES Client (ID\_Client),      Request\_Date date null DEFAULT(CURRENT\_DATE),      Request\_Time TIME null DEFAULT(CURRENT\_TIMESTAMP)  ); |
|  | Request\_Number | Int | Порядковый номер заявки |
| FK | Manager\_ID | Int |  |
| FK | Client\_ID | Int |  |
|  | Request\_Date | Date | Дата заявки |
|  | Request\_Time | Time | Время заявки |
| Cargo | | | | |
| PK | ID\_Cargo | Int | Первичный ключ таблицы Груз | CREATE TABLE Cargo(      ID\_Cargo int not null auto\_increment PRIMARY KEY,      Cargo\_Description varchar(200) not null,      Request\_ID int not null REFERENCES Request (ID\_Request),      Cargo\_Width int not null,      Cargo\_Height int not null,      Cargo\_Length int not null,      Cargo\_Weight int not null,      Cargo\_Price int null DEFAULT 0  ); |
|  | Cargo\_Description | Varchar(200) |  |
| FK | Request\_ID | Int |  |
|  | Cargo\_Width | Int | Ширина в сантиметрах |
|  | Cargo\_Height | Int | Высота в сантиметрах |
|  | Cargo\_Length | Int | Длина в сантиметрах |
|  | Cargo\_Weight | Int | Вес в кг |
|  | Cargo\_Price | Int | NULL  Стоимость в рублях |
| Delivery\_Point | | | | |
| PK | ID\_Delivery\_Point | Int | Первичный ключ таблицы Точка доставки | CREATE TABLE Delivery\_Point(      ID\_Delivery\_Point int not null auto\_increment PRIMARY KEY,      Delivery\_Point\_Mark varchar(1) null DEFAULT '-' constraint CH\_Delivery\_Point\_Mark          CHECK(regexp\_like(Delivery\_Point\_Mark, '[+]|[-]')),      Request\_ID int not null REFERENCES Request (ID\_Request),      Delivery\_Point\_Destination varchar(200) not null  ); |
|  | Delivery\_Point\_Mark | Varchar(1) | NULL  + - завершён  NULL – - незавершён |
| FK | Request\_ID | Int |  |
|  | Delivery\_Point\_Destination | Varchar(MAX) | Адрес |
| Route\_Sheet | | | | |
| PK | ID\_Route\_Sheet | Int | Первичный ключ таблицы Маршрутный лист | CREATE TABLE Route\_Sheet(      ID\_Route\_Sheet int not null auto\_increment PRIMARY KEY,      Route\_Sheet\_Number int not null,      Complexity\_ID int not null REFERENCES Complexity (ID\_Complexity),      Vehicle\_ID int not null REFERENCES Vehicle (ID\_Vehicle),      Driver\_ID int not null REFERENCES Driver (ID\_Driver),      Route\_Sheet\_Date Date null DEFAULT(CURRENT\_TIMESTAMP)  ); |
|  | Route\_Sheet\_Number | Int | Порядковый номер маршрутного листа |
| FK | Complexity\_ID | Int |  |
| FK | Vehicle\_ID | Int |  |
| FK | Driver\_ID | Int |  |
|  | Route\_Sheet\_Date | Date |  |
|  |  |  |  |  |
| Combination\_Route\_Sheet\_Delivery\_Point | | | | |
| PK | ID\_Combination\_Route\_Sheet\_Delivery\_Point | Int | Первичный ключ таблицы Совместительство маршрутный лист + точка доставки | CREATE TABLE Combination\_Route\_Sheet\_Delivery\_Point(      ID\_Combination\_Route\_Sheet\_Delivery\_Point int not null auto\_increment PRIMARY KEY,      Route\_Sheet\_ID int not null REFERENCES Route\_Sheet (ID\_Route\_Sheet),      Delivery\_Point\_ID int not null REFERENCES Delivery\_Point (ID\_Delivery\_Point)  ); |
| FK | Route\_Sheet\_ID | Int |  |
| FK | Delivery\_Point\_ID | Int |  |
| Combination\_Route\_Sheet\_Cargo | | | | |
| PK | ID\_Combination\_Route\_Sheet\_Cargo | Int | Первичный ключ таблицы Совместительство маршрутный лист + груз | CREATE TABLE Combination\_Route\_Sheet\_Cargo(      ID\_Combination\_Route\_Sheet\_Cargo int not null auto\_increment PRIMARY KEY,      Route\_Sheet\_ID int not null REFERENCES Route\_Sheet (ID\_Route\_Sheet),      Cargo\_ID int not null REFERENCES Cargo (ID\_Cargo)  ); |
| FK | Route\_Sheet\_ID | Int |  |
| FK | Cargo\_ID | Int |  |

1. Демонстрация древа объектов БД.



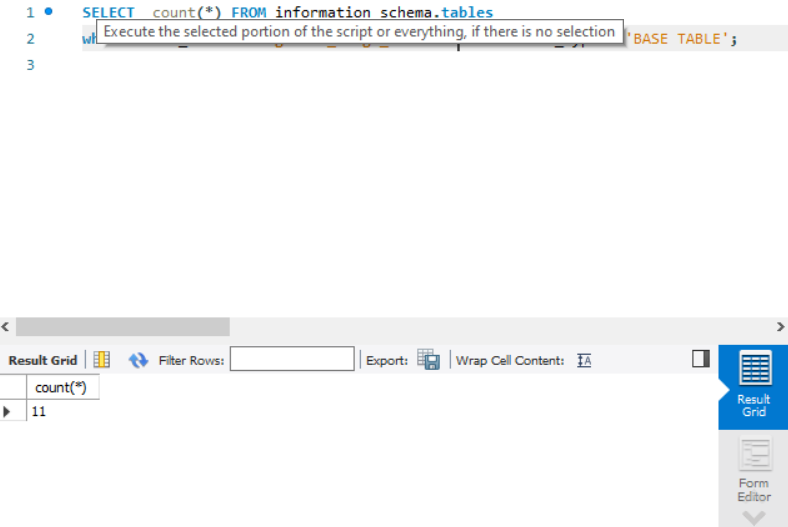




1. Отчёт о количестве таблиц;

SELECT count(\*) FROM information\_schema.tables

where table\_schema = 'sale\_product\_database' and table\_type = 'BASE TABLE';



1. Отчёт о количестве столбцов в таблицах.

SELECT table\_name, count(column\_Name) FROM information\_schema.columns

where table\_schema = 'sale\_product\_database'

group by table\_name;

