ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 2](#_Toc73687788)

[1. Теоретическая часть – проектирование базы данных 3](#_Toc73687789)

[1.1 Анализ предметной области 3](#_Toc73687790)

[1.2 Построение логической структуры базы данных; 3](#_Toc73687791)

[1.3 Нормализация отношений 4](#_Toc73687792)

[1.4 Определение ограничений целостности 11](#_Toc73687793)

[1.5 Преобразование ER–диаграммы в схему базы данных 11](#_Toc73687794)

[2 Практическая часть – реализация базы данных в выбранной СУБД 13](#_Toc73687795)

[2.1 Реализация базы данных 13](#_Toc73687796)

[2.2 Создание запросов 17](#_Toc73687797)

[Задание а. 17](#_Toc73687798)

[Задание b 18](#_Toc73687799)

[Задание c. 18](#_Toc73687800)

[Задание d. 20](#_Toc73687801)

[Задание e. 21](#_Toc73687802)

[2.3 Создание представлений 22](#_Toc73687803)

[Задание a 22](#_Toc73687804)

[Задание b 23](#_Toc73687805)

[Задание c 23](#_Toc73687806)

[Задание d 23](#_Toc73687807)

[Задание e 23](#_Toc73687808)

[2.4. Создание индексов 24](#_Toc73687809)

[ЗАКЛЮЧЕНИЕ 25](#_Toc73687810)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 26](#_Toc73687811)

[Приложение А 27](#_Toc73687812)

# ВВЕДЕНИЕ

Тема курсовой работы: Информационная система «Фирма 13»

В качестве предметной области проектирования БД взята торговая фирма, которая исолняет реализацию заказов, исходящих со стороны клиентов.

Основной задачей курсового проекта является применение на практике знаний, приобретенных в ходе изучения дисциплины "Базы данных", и применение практических навыков создания автоматизированных информационных систем (АИС), основанных на базах данных. Прозвести анализ предметной области. Овладеть концептуальным проектированием и научиться определять сущности и атрибуты БД. Обучиться создавать инфологическую модель БД в виде ER-диаграмм. Получить практические навыки и теоретические знания в ходе физического проектирования баз данных (БД).

# 1. Теоретическая часть – проектирование базы данных

## 1.1 Анализ предметной области

В качестве предметной области проектирования БД выбрана торговая фирма, которая осуществляет выполнение заказов, поступающих от клиентов. Для построения БД использованы следующие объекты и их атрибуты:

* + - 1. Фирма (название фирмы, юр. адрес, телефон фирмы, №счета в банке, наименование товара, цена товара);
      2. Склад (№склада, адрес склада, №заказа, наименование товара, единицы измерения, кол-во товара, цена товара, отметки об оплате);
      3. Кадры (Название фирмы, Ф.И.О. сотрудника, должность сотрудника, образование сотрудника, оклад, дата принятия на работу);
      4. Поставщик (№поставщика, Ф.И.О. поставщика, юр. адрес поставщика, №счета поставщика, наименование товара, цена товара);
      5. Транспорт (№склада, №автомобиля, марка автомобиля, грузоподъемность, год выпуска, Ф.И.О. владельца, дата перевозки);
      6. Накладная (№накладной, наименование товара, единицы измерения, кол-во, цена, сумма);
      7. Товар (Наименование товара, единицы измерения, сорт, цена, поставщик, адрес поставщика);
      8. Клиент (№клиента, Ф.И.О. клиента, юр. адрес клиента, №счета клиента, №заказа, наименование товара, цена товара); Форма 3
      9. Счет (Наименование банка, №счета, Ф.И.О. владельца счета, состояние счета, дата, наличие кредита, сумма кредита); Форма 3

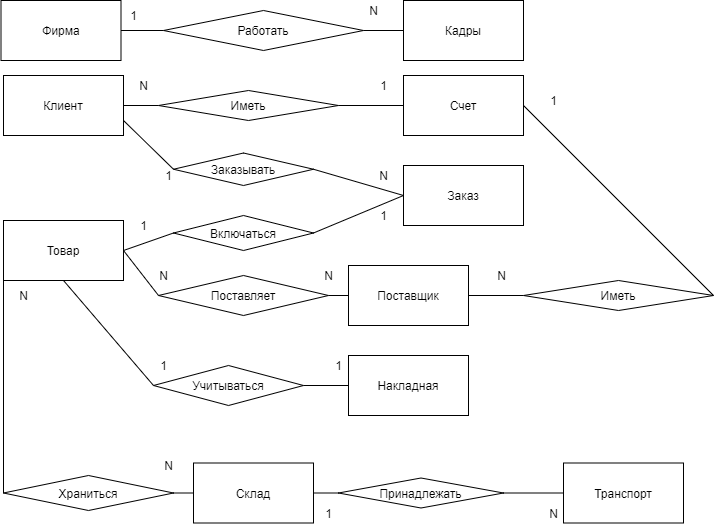
10.3аказ (№заказа, дата заказа, Ф.И.О. клиента, №счета клиента, наименование товара, количество, цена товара)

## 1.2 Построение логической структуры базы данных;

Исходя из выявленных сущностей, построим ER–диаграмму (рисунок 1.1)

Рисунок 1.1

ER-диаграмма фирмы.



Преобразование ER–диаграммы в схему БД выполняется путем сопоставления каждой сущности и каждой связи, имеющей атрибуты, отношения (таблицы) БД. Связь типа 1:N (один-ко-многим) между отношениями реализуется через внешний ключ. Ключ вводится для того отношения, к которому осуществляется множественная связь.

## 1.3 Нормализация отношений

Потенциальным ключом отношения ФИРМА является атрибут Название, поскольку название в дальнейшей работе фигурировать не будет, и фирма у нас всего одна, оно и будет ключевым полем. Отношение представлено в таблице 1.1.

Таблица 1.1 Отношение ФИРМА

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Название | naz | C(500) | Обязательное поле, первичный ключ |
| Юр. адрес | adr | C(500) | Обязательное поле |
| Телефон | phon | C(50) | Необязательное поле |
| № счета банка | schet | C(50) | Обязательное поле |
| Наименование товара | tovar | C(100) | Необязательное поле |
| Цена товара | cena | N | Необязательное поле |

Потенциальным ключом отношения СКЛАД является атрибут Номер, он уникален, поэтому мы выбираем его в качестве первичного ключа. Отношение представлено в таблице 1.2.

Таблица 1.2 Отношение СКЛАД

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| №склада | idsklad | N | Первичный ключ |
| Адрес склада | adr | C(500) | Обязательное поле |
| №заказа | zakaz | N | Обязательное поле |
| № счета банка | schet | C(500) | Обязательное поле |
| Наименование товара | tovar | C(500) | Обязательное поле |
| Цена товара | cena | N | Обязательное поле |
| Единицы измерения | edizm | N | Обязательное поле |
| Кол-во товара | kolvo | N | Обязательное поле |
| Отметки об оплате | opl | N | Обязательное поле |

Потенциальным ключом отношения КАДРЫ является атрибут ФИО, к тому же теоретически, хоть редко, но ФИО может и совпадать, так же это длинная текстовая строка, поэтому мы введем суррогатный ключ ID и выбираем его в качестве первичного ключа. Отношение представлено в таблице 1.3.

Таблица 1.3 Схема отношения КАДРЫ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Название фирмы | naz | N | Обязательное поле |
| ФИО сотр. | fio | C(150) | Обязательное поле |
| Должность | dolg | C(150) | Обязательное поле |
| Образование | obraz | C(100) | Обязательное поле |
| Оклад | oklad | N | Обязательное поле |
| Дата принятия | data | D | Обязательное поле |
| Идентификатор | idkadr | N | Первичный ключ суррогатный |

Потенциальным ключом отношения ПОСТАВЩИК является атрибут Номер поставщика, он уникален, поэтому мы выбираем его в качестве первичного ключа. Отношение представлено в таблице 1.4.

Таблица 1.4 Схема отношения ПОСТАВЩИК

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| №поставщика | idpostav | N | Первичный ключ |
| ФИО пост. | fio | C(150) | Обязательное поле |
| Юр. адр. | adr | C(150) | Обязательное поле |
| №счета | schet | C(100) | Обязательное поле |
| Наимен. товара | naimen | N | Обязательное поле |
| Цена товара | cena | N | Обязательное поле |

Потенциальным ключом отношения ТРАНСПОРТ является атрибут Номер Автомобиля, номера автомобилей уникальны, поэтому можно его выбрать в качестве первичного ключа, однако введение суррогатного ключа позволит сократить занимаемое базой данных место и ускорить обработку. Отношение представлено в таблице 1.5.

Таблица 1.5 Схема отношения ТРАНСПОРТ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| №склада | nsklad | N | Обязательное поле |
| №авто | nomer | C(500) | Обязательное поле |
| Марка авто | marka | C(500) | Обязательное поле |
| Грузоподъемность | gruz | N | Обязательное поле |
| Год выпуска | god | N | Обязательное поле |
| ФИО владельца | vladelec | C(500) | Обязательное поле |
| Дата перев. | data | D | Обязательное поле |
| Идентификатор | id | N | Первичный ключ |

Потенциальным ключом отношения НАКЛАДНАЯ является атрибут Номер накладной, он уникален, поэтому мы выбираем его в качестве первичного ключа. Отношение представлено в таблице 1.6.

Таблица 1.6 Схема отношения НАКЛАДНАЯ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| №накладной | idnak | N | Первичный ключ |
| Наименование товара | naimen | C(150) | Обязательное поле |
| Ед. изм. | edizm | C(50) | Обязательное поле |
| Кол-во | kolvo | N | Обязательное поле |
| Цена | cena | N | Обязательное поле |
| Сумма | summa | N | Обязательное поле |

Потенциальным ключом отношения ТОВАР можно выбрать наименование, однако это длинная строка, поэтому мы вводим суррогатный ключ ID и выбираем его в качестве первичного ключа. Отношение представлено в таблице 1.7.

Таблица 1.7 Схема отношения ТОВАР

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Наименование | Naimen\_tovara | C(500) | Обязательное поле |
| Ед. изм. | Ed\_izm | C(50) | Обязательное поле |
| Сорт | sort | N | Обязательное поле |

Таблица 1.7 продолжение.

|  |  |  |  |
| --- | --- | --- | --- |
| Цена | cena | N | Обязательное поле |
| Поставщик | postav | C(500) | Обязательное поле |
| Адрес поставщика | adr | C(500) | Обязательное поле |
| Идентификатор | id | N | Первичный ключ |

Потенциальным ключом отношения КЛИЕНТ является атрибут Номер клиента, он уникален, поэтому мы выбираем его в качестве первичного ключа. Отношение представлено в таблице 1.8.

Таблица 1.8 Схема отношения КЛИЕНТ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| № клиента | idklient | N | Первичный ключ |
| ФИО клиента | fio | C(500) | Обязательное поле |
| Юр. адр. | adr | C(500) | Обязательное поле |
| №счета | idschet | C(500) | Обязательное поле |
| № заказа | zakaz | N | Обязательное поле |
| Цена товара | cena | N | Обязательное поле |
| Наименов. товара | naimen | C(500) | Обязательное поле |

Потенциальным ключом отношения СЧЕТ является атрибут Номер счета, однако в различных банках номер счета может совпадать, можно сделать составной ключ и добавить наименование банка, однако это существенно увеличит его размер, поэтому мы вводим суррогатный идентификатор ID и выбираем его в качестве первичного ключа. Отношение представлено в таблице 1.9.

Таблица 1.9 Схема отношения СЧЕТ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| № счета | numschet | C(100) | Обязательное поле |
| Наименов. банка | bank | C(500) | Обязательное поле |
| ФИО владельца | fio | C(500) | Обязательное поле |
| Состояние счета | status | N | Обязательное поле |
| Дата | data | D | Обязательное поле |
| Наличие кредита | kreditstat | N | Обязательное поле |
| Сумма кредита | summa | N | Обязательное поле |
| Идентификатор | idschet | N | Первичный ключ |

Потенциальным ключом отношения ЗАКАЗ является атрибут Номер заказа, он уникален, поэтому мы выбираем его в качестве первичного ключа. Отношение представлено в таблице 1.10.

Таблица 1.10 Схема отношения ЗАКАЗ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |

Таблица 1.10 продолжение.

|  |  |  |  |
| --- | --- | --- | --- |
| № заказа | schet | N | Первичный ключ |
| Дата заказа | data | D | Обязательное поле |
| ФИО клиента | fio | C(500) | Обязательное поле |
| № счета | schet | C(500) | Обязательное поле |
| Наименование товара | naimen | C(500) | Обязательное поле |
| Количество | kolvo | N | Обязательное поле |
| Цена товара | cena | N | Обязательное поле |

Рассмотрим отношения ФИРМА, ПОСТАВЩИК, КЛИЕНТ, в каждом из них есть атрибут Счет банка, заменим этот атрибут на идентификатор счета отношения СЧЕТ, а из отношения СЧЕТ удалим атрибут ФИО, таким образом образуем связь 1:1, наложив ограничение уникальности на атрибуты idschet таблиц ФИРМА, ПОСТАВЩИК, КЛИЕНТ. Отношение представлено в таблице 1.11.

Таблица 1.11 Схема отношения СЧЕТ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Идентификатор | idschet | N | Первичный ключ |
| № счета | numschet | C(50) | Обязательное поле |
| Наименов. банка | bank | C(50) | Обязательное поле |
| Состояние счета | status | N | Обязательное поле |
| Дата | data | D | Обязательное поле |
| Наличие кредита | kreditstat | N | Обязательное поле |
| Сумма кредита | summa | N | Обязательное поле |

В отношениях ПОСТАВЩИК и КЛИЕНТ исключим транзитивные атрибуты Наименование товара и Цена товара, а так же №Заказа. Отношение представлено в таблице 1.12 и 1.13.

Таблица 1.12 Схема отношения ПОСТАВЩИК

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| № поставщика | idpostav | N | Первичный ключ |
| ФИО пост. | fio | C(500) | Обязательное поле |
| Юр. адр. | adr | C(500) | Обязательное поле |
| №счета | idschet | N | Обязательное поле, уникальное, внешний ключ к СЧЕТ |

Таблица 1.13 Схема отношения КЛИЕНТ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| № клиента | idklient | N | Первичный ключ |
| ФИО клиента | fio | C(100) | Обязательное поле |
| Юр. адр. | adr | C(300) | Обязательное поле |
| №счета | idschet | N | Обязательное поле, уникальное, внешний ключ к СЧЕТ |

А в отношении ТОВАР установим связь с ПОСТАВЩИК по №Поставщик, исключив так же повторяющийся атрибут Адрес поставщика, так как он присущ поставщику и может быть получен из отношения ПОСТАВЩИК. С другой стороны необходимо установить связь с отношением СКЛАД введением атрибута idsklad. Отношение представлено в таблице 1.14.

Таблица 1.14 Схема отношения ТОВАР

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Идентификатор | idtovar | N | Первичный ключ |
| Наименование | Naimen | C(500) | Обязательное поле |
| Ед. изм. | Ed\_izm | C(50) | Обязательное поле |
| Сорт | sort | N | Обязательное поле |
| Цена | cena | N | Обязательное поле |
| Идентификатор склад | sklad | N | Обязательное поле, внешний ключ к СКЛАД |
| Поставщик | idpostav | N | Обязательное поле, внешний ключ к ПОСТАВЩИК |

Рассмотрим отношение ЗАКАЗ, атрибуты Цена товара, Наименование товара, №счета, ФИО клиента и номер счета могут быть транзитивно получены из других связанных отношений. Отношение представлено в таблице 1.15.

Таблица 1.15 Схема отношения ЗАКАЗ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| № заказа | idzakaz | N | Первичный ключ |
| Дата заказа | data | D | Обязательное поле |
| Идентификатор клиента | idclient | N | Обязательное поле, внешний ключ к КЛИЕНТ |
| Идентификатор товара | idtovar | N | Обязательное поле, внешний ключ к ТОВАР |
| Количество | kolvo | N | Обязательное поле |
| Отметки об оплате | opl | N | Обязательное поле, перечисление |

Рассмотрим отношение НАКЛАДНАЯ, атрибуты № накладной полностью удовлетворяет требования к ключу отношения. Наименование товара, единицы измерения товара, его цена могут быть транзитивно получены из отношения ТОВАР, заменим их связью. Отношение представлено в таблице 1.16.

Таблица 1.16 Схема отношения НАКЛАДНАЯ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| №накладной | idnakladnaya | N | Первичный ключ |

Таблица 1.17 продолжение.

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор товара | idtovar | N | Обязательное поле, внешний ключ к ТОВАР |
| Кол-во | kolvo | N | Обязательное поле |
| Сумма | summa | N | Обязательное поле |
| Идентификатор клиента | idclient | N | Обязательное поле, внешний ключ к КЛИЕНТ |

№ склада будет являться первичным ключом, так как он удовлетворяет всем требованиям к первичному ключу отношения. Атрибуты Наименование товара, Цена товара, Единицы измерения, Количество товара могут быть транзитивно получены из отношения ТОВАР и удалим их, так как у нас уже есть связь со складом в отношении ТОВАР, так же № заказа и его состояние его оплаты могут быть получены через транзитивную связь. Отношение представлено в таблице 1.17.

Таблица 1.17 Схема отношения СКЛАД

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| №склада | nsklad | N | Первичный ключ |
| Адрес склада | adr | C(500) | Обязательное поле |

Атрибуты Наименование товара, Цена товара могут быть получены из отношения ТОВАР, удалим их. В качестве первичного ключа у нас выступает наименование фирмы, так же удалим реквизиты счета, заменим его связью со СЧЕТ. Отношение представлено в таблице 1.18.

Таблица 1.18 Схема отношения ФИРМА

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Название | nazvan | C(500) | Обязательное поле |
| Юр. адрес | uradr | C(500) | Обязательное поле |
| Телефон | phone | C(50) | Необязательное поле |
| Идентификатор счета банка | idschet | N | Обязательное поле, внешний ключ к СЧЕТ |

В отношении КАДРЫ удалим атрибут Наименование фирмы так как все наши сотрудники работают в нашей фирме, первичный ключ – идентификатор ID. Отношение представлено в таблице 1.19.

Таблица 1.19 Схема отношения КАДРЫ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Идентификатор | idkadr | N | Первичный ключ суррогатный |
| ФИО сотр. | fio | C(150) | Обязательное поле |
| Должность | dolg | C(500) | Обязательное поле |
| Образование | obraz | C(500) | Обязательное поле, перечисляемое |

Таблица 1.19 продолжение.

|  |  |  |  |
| --- | --- | --- | --- |
| Оклад | oklad | N | Обязательное поле |
| Дата принятия | data | D | Обязательное поле |

Первичный ключ отношения ТРАНСПОРТ атрибут id. № склада заменим связью с отношением СКЛАД. Отношение представлено в таблице 1.20.

Таблица 1.20 Схема отношения ТРАНСПОРТ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Идентификатор | idtransport | N | Первичный ключ |
| №склада | nsklad | N | Обязательное поле, внешний ключ к СКЛАД |
| №авто | nomer | C(50) | Обязательное поле |
| Марка авто | marka | C(500) | Обязательное поле |
| Грузоподъемность | gruz | N | Обязательное поле |
| Год выпуска | god | N | Обязательное поле |
| ФИО владельца | fio | C(500) | Обязательное поле |
| Дата перев. | data | D | Обязательное поле |

## 1.4 Определение ограничений целостности

Атрибут Образование отношения КАДРЫ может принимать одно из следующих значений: ‘Высшее’.’Среднеспециальное’,’Среднее’,’Базовое’.

Атрибуты Зарплата отношения КАДРЫ, Количество и Сумма отношения НАКЛАДНАЯ, Количество и Сумма отношения СЧЕТ, Цена и Количество отношения ТОВАР, Год и Тоннаж отношения ТРАНСПОРТ, Количество и Оплачено отношения ЗАКАЗ могут принимать только неотрицательные значения.

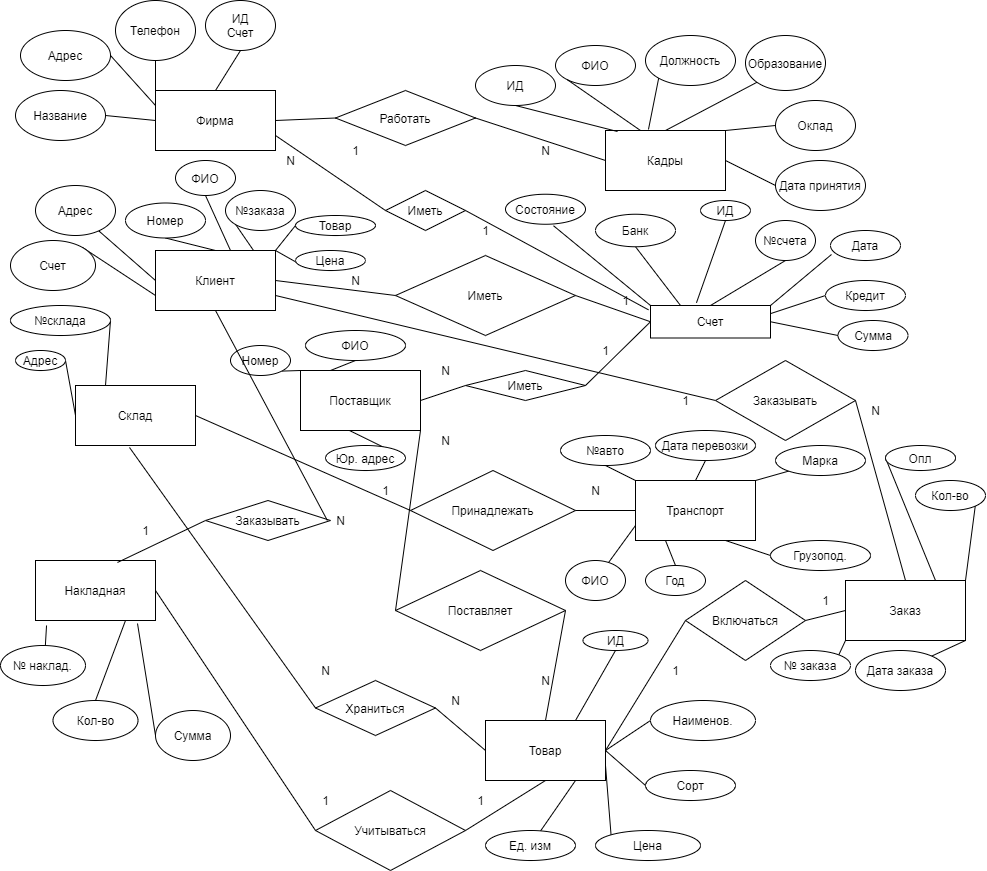
Атрибут Состояние кредита отношения СЧЕТ может принимать одно из следующих значений: ‘Y’, ‘N’

## 1.5 Преобразование ER–диаграммы в схему базы данных

База данных создается на основании схемы базы данных. Для преобразования ER–диаграммы в схему БД приведем уточненную ER–диаграмму, содержащую атрибуты сущностей (рисунок 1.2).

Рисунок 1.2

Окончательная ER-диаграмма фирмы.



Атрибуты всех наших отношений являются простыми, и все используемые домены содержат только скалярные значения, следовательно это 1 нормальная форма.К аждый не ключевой атрибут наших отношений неприводимо зависит от Первичного Ключа(ПК), следовательно отношения находится во 2НФ. Отношения находятся в 3НФ, так как каждый не ключевой атрибут нетранзитивно зависит от первичного ключа потому что мы вынесли все не ключевые поля в отдельные таблицы.

После создания ER-диаграммы можно приступать к созданию базы данных.

# Практическая часть – реализация базы данных в выбранной СУБД

## 2.1 Реализация базы данных

Для физического воплощения базы данных будет использоваться СУБД MySQL. Это легковесное, очень распространенное и быстрое ПО, применяемое, как правило, для работы в сети Internet. В этой СУБД можно создавать хранимые процедуры и триггеры, легко устанавливать связи, создавать представления. Возможность работы в многопользовательском режиме позволяет получить к ней доступ нескольким пользователям фирмы одновременно.

Приведем описание схемы БД на языке DDL с учетом наших связей и ограничений.

Отношение СЧЕТ

CREATE TABLE `schet` (

`idschet` int NOT NULL AUTO\_INCREMENT,

`bank` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL DEFAULT 'Не указано',

`numchet` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

`status` int NOT NULL DEFAULT '0',

`data` date NOT NULL,

`creditstat` enum('Y','N') NOT NULL,

`summa` float(12,0) DEFAULT NULL,

PRIMARY KEY (`idschet`),

UNIQUE KEY `idschet` (`idschet`)

) ENGINE=InnoDB AUTO\_INCREMENT=21 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

Отношение ПОСТАВЩИК

CREATE TABLE `postav` (

`idpostav` int NOT NULL AUTO\_INCREMENT,

`fio` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL DEFAULT 'Не указано',

`adr` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL DEFAULT 'Не указано',

`idschet` int DEFAULT NULL,

PRIMARY KEY (`idpostav`),

UNIQUE KEY `idpostav` (`idpostav`),

UNIQUE KEY `i4` (`fio`),

KEY `FKschetp` (`idschet`),

CONSTRAINT `FKschetp` FOREIGN KEY (`idschet`) REFERENCES `schet` (`idschet`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='Поставщики';

Отношение КЛИЕНТ

CREATE TABLE `client` (

`idklient` int NOT NULL AUTO\_INCREMENT,

`fio` varchar(250) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL DEFAULT 'Не указано',

`adr` varchar(500) DEFAULT NULL,

`idschet` int DEFAULT NULL,

PRIMARY KEY (`idklient`),

UNIQUE KEY `idclient` (`idklient`),

UNIQUE KEY `i3` (`fio`),

KEY `FKschet` (`idschet`),

CONSTRAINT `FKschet` FOREIGN KEY (`idschet`) REFERENCES `schet` (`idschet`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

Отношение ТОВАР

CREATE TABLE `tovar` (

`idtovar` int NOT NULL AUTO\_INCREMENT,

`naimen` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL DEFAULT 'Не указано',

`ed\_izm` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL DEFAULT 'Не указано',

`cena` int NOT NULL,

`idpostav` int NOT NULL,

`sklad` int unsigned NOT NULL,

`sort` int NOT NULL,

PRIMARY KEY (`idtovar`),

UNIQUE KEY `idtovar` (`idtovar`),

KEY `FKpostavt` (`idpostav`),

KEY `FK\_tovar\_sklad` (`sklad`),

KEY `i5` (`naimen`) USING BTREE,

CONSTRAINT `FK\_tovar\_sklad` FOREIGN KEY (`sklad`) REFERENCES `sklad` (`Nsklad`),

CONSTRAINT `FKpostavt` FOREIGN KEY (`idpostav`) REFERENCES `postav` (`idpostav`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='Список товаров нормализованный';

Отношение ЗАКАЗ

CREATE TABLE `zakaz` (

`idzakaz` int NOT NULL AUTO\_INCREMENT,

`data` date DEFAULT NULL,

`idclient` int NOT NULL,

`idtovar` int NOT NULL,

`kolvo` int NOT NULL DEFAULT '0',

`opl` enum('Y','N') DEFAULT NULL,

PRIMARY KEY (`idzakaz`) USING BTREE,

UNIQUE KEY `id\_zakaz` (`idzakaz`) USING BTREE,

KEY `FKtovar` (`idtovar`) USING BTREE,

KEY `FKclientz` (`idclient`) USING BTREE,

CONSTRAINT `FKclientz` FOREIGN KEY (`idclient`) REFERENCES `client` (`idklient`),

CONSTRAINT `FKtovar` FOREIGN KEY (`idtovar`) REFERENCES `tovar` (`idtovar`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='Список заказов п 10';

Отношение НАКЛАДНАЯ

CREATE TABLE `naklad` (

`idnakladnaya` int NOT NULL AUTO\_INCREMENT,

`idtovar` int NOT NULL,

`kolvo` int unsigned NOT NULL,

`summa` float(12,0) unsigned NOT NULL,

`idklient` int NOT NULL,

PRIMARY KEY (`idnakladnaya`),

UNIQUE KEY `idnaklad` (`idnakladnaya`),

KEY `FKtovarn` (`idtovar`),

KEY `FK\_nakladnaya\_klient` (`idklient`),

CONSTRAINT `FK\_nakladnaya\_klient` FOREIGN KEY (`idklient`) REFERENCES `client` (`idklient`),

CONSTRAINT `FKtovarn` FOREIGN KEY (`idtovar`) REFERENCES `tovar` (`idtovar`)

) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='Накладные';

Отношение СКЛАД

CREATE TABLE `sklad` (

`Nsklad` int unsigned NOT NULL AUTO\_INCREMENT,

`adress` varchar(5000) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

PRIMARY KEY (`Nsklad`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

Отношение ФИРМА

CREATE TABLE `firma` (

`nazvan` varchar(500) NOT NULL DEFAULT '',

`uradr` varchar(500) NOT NULL DEFAULT '',

`phone` varchar(50) NOT NULL DEFAULT '',

`idschet` int NOT NULL,

KEY `FKschetf` (`idschet`),

KEY `I1` (`nazvan`),

CONSTRAINT `FKschetf` FOREIGN KEY (`idschet`) REFERENCES `schet` (`idschet`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='Описание фирмы';

Отношение КАДРЫ

CREATE TABLE `kadr` (

`idkadry` int NOT NULL AUTO\_INCREMENT,

`fio` varchar(250) NOT NULL,

`dolg` varchar(500) DEFAULT NULL,

`obraz` enum('Высшее','Среднеспециальное','Среднее','Базовое') NOT NULL,

`oklad` float(12,0) NOT NULL,

`data` date NOT NULL,

PRIMARY KEY (`idkadry`),

UNIQUE KEY `idkadr` (`idkadry`),

UNIQUE KEY `i2` (`fio`)

) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='Сотрудники фирмы';COLLATE 'utf8mb4\_0900\_ai\_ci'

ROW\_FORMAT DEFAULT

Отношение ТРАНСПОРТ

CREATE TABLE `transport` (

`idtransport` int NOT NULL AUTO\_INCREMENT,

`nomer` varchar(45) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

`marka` varchar(45) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

`gruz` float(12,0) NOT NULL,

`god` int NOT NULL,

`fio` varchar(450) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

`data` date DEFAULT NULL,

`nsklad` int unsigned DEFAULT NULL,

PRIMARY KEY (`idtransport`) USING BTREE,

UNIQUE KEY `i5` (`nomer`),

KEY `FK\_transport\_sklad` (`nsklad`),

CONSTRAINT `FK\_transport\_sklad` FOREIGN KEY (`nsklad`) REFERENCES `sklad` (`Nsklad`)

) ENGINE=InnoDB AUTO\_INCREMENT=8 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

## 2.2 Создание запросов

### Задание а.

Кто из сотрудников был принят на работу на указанную дату?

Целью задания является получение списка сотрудников у которых дата принятия на работу менее указанной.

Сформулируем запрос в терминах реляционной алгебры:

([Сотрудник.Дата<’01.01.2021’])[Сотрудник.ФИО, Сотрудник.Дата]

Запишем запрос на языке DML:

select `matvey`.`kadr`.`fio` AS `fio`,`matvey`.`kadr`.`data` AS `data` from `matvey`.`kadr`

where (`matvey`.`kadr`.`data` < '2021-01-01');

Результат выполнения запроса и входные данные представлены на рисунках 2.1 и 2.2.

Рисунок 2.1

Таблица сотрудников

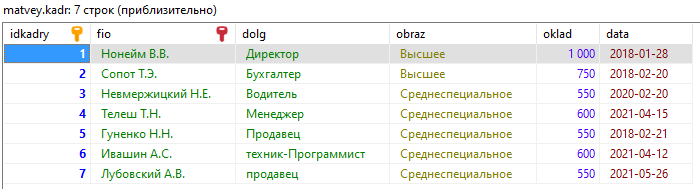
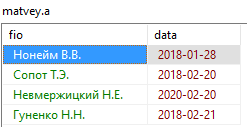


Рисунок 2.2

Результат выполнения запроса



### Задание b

Какие автомобили имеют данную грузоподъемность и выпущены не позже данной даты?

Входные данные задания: грузоподъемность автомобиля и его год выпуска. Необходимо получить список автомобилей этой грузоподъемности с годом выпуска менее указанного. Для получения необходимой информации нам достаточно одной таблицы: таблицы транспорта.

Сформулируем запрос в терминах реляционной алгебры:

([Транспорт.Грузоподъемность=1000&Транспорт.Год<’2015-01-01’])[Транспорт.Марка, Транспорт.Номер, Транспорт.Год]

Запишем запрос на языке DML:

select `matvey`.`transport`.`marka` AS `marka`,`matvey`.`transport`.`gruz` AS `gruz`,`matvey`.`transport`.`god` AS `god`,`matvey`.`transport`.`data` AS `DATA` from `matvey`.`transport` where ((`matvey`.`transport`.`data` < '2021-01-01') and (`matvey`.`transport`.`gruz` = 1000));

Результат выполнения запроса и входные данные представлены на рисунках 2.3 и 2.4.

Рисунок 2.3

Таблица транспорта

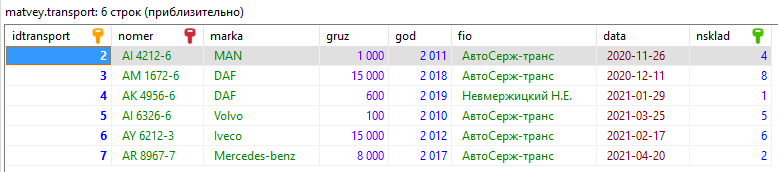


Рисунок 2.4

Результат выполнения запроса



### Задание c.

Найти адреса изготовителей данного товара с данной ценой и данным сортом.

В задании есть входные данные: наименование товара, его цена и сорт, и имеются данные, которые нам необходимо выяснить, это адрес поставщика данного товара. Для получения необходимой информации нам необходимо задействовать две таблицы: таблицу товаров и таблицу поставщиков. Из таблицы товаров мы выберем указанный товар, а из неё по номеру поставщика, выберем искомые данные из таблицы поставщиков.

Сформулируем запрос в терминах реляционной алгебры:

([Товар.Наименование=’Телефон’&Товар.Сорт=1&Товар.Цена=500&Товар.ИДпоставщика=Поставщик.ИД])[Поставщик.наименование, Поставщик.ЮрАдрес]

Запишем запрос на языке DML:

select `p`.`fio` AS `fio`,`p`.`adr` AS `uradr`

from (`matvey`.`tovar` `t` join `matvey`.`postav` `p`)

where ((`p`.`idpostav` = `t`.`idpostav`)

and (`t`.`sort` = 1) and (`t`.`cena` = 500) and (`t`.`naimen` = 'Телефон'));

Результат выполнения запроса и входные данные представлены на рисунках 2.5, 2.6 и 2.7.

Рисунок 2.5

Таблица поставщиков

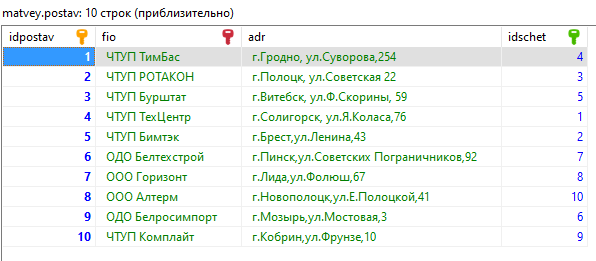


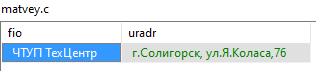
Рисунок 2.6

Таблица товаров



Рисунок 2.7

Результат выполнения запроса



### Задание d.

В каком банке открыл счет данный поставщик.

Входными данными для нас является название поставщика, выходными – банк и номер счета в банке.

Сформулируем запрос в терминах реляционной алгебры:

([Поставщик.ФИО=’Название поставщика’ &Поставщик.ИДСчет=Счета.ИДСчет])[Поставщик.Наименование, Счет.Банк, Счет.Номер]

Запишем запрос на языке DML:

select `p`.`fio` AS `fio`,`s`.`bank` AS `bank`,`s`.`numchet` AS `numchet`

from (`matvey`.`schet` `s` join `matvey`.`postav` `p`)

where ((`p`.`fio` = 'ЧТУП ТимБас')

and (`p`.`idschet` = `s`.`idschet`));

Результат выполнения запроса и входные данные представлены на рисунках 2.8, 2.9 и 2.10.

Рисунок 2.8

Таблица поставщиков

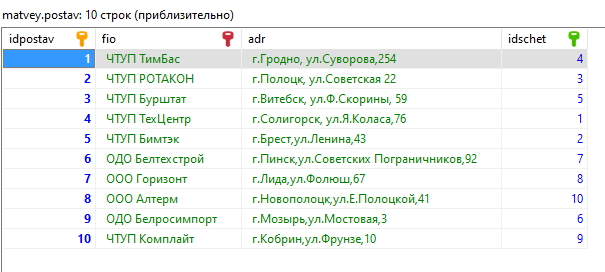


Рисунок 2.9

Таблица счетов

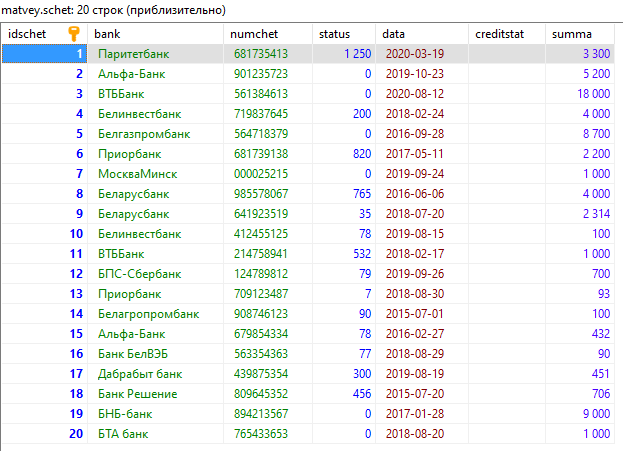
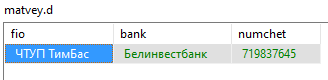


Рисунок 2.10

Результат выполнения запроса



### Задание e.

Является ли владелец автомобиля с данной фамилией работником данной фирмы.

В задании необходимо найти записи у которых совпадает ФИО в таблице владельцев автомобилей и в таблице сотрудников. Необходимо задействовать 2 таблицы – транспорта и кадров. Из таблицы кадров выберем фил сотрудников и проверим являются ли они владельцами автомобиля.

Сформулируем запрос в терминах реляционной алгебры:

([Транспорт.Владелец=Кадры.ФИО])[Кадры.ФИО, Кадры.Должность]

Запишем запрос на языке DML:

select `matvey`.`kadr`.`fio` AS `fio`,`matvey`.`kadr`.`dolg` AS `dolg` from (`matvey`.`kadr`

join `matvey`.`transport` on((`matvey`.`kadr`.`fio` = `matvey`.`transport`.`fio`)));

Результат выполнения запроса и входные данные представлены на рисунках 2.11, 2.12 и 2.13.

Рисунок 2.11

Таблица транспорта

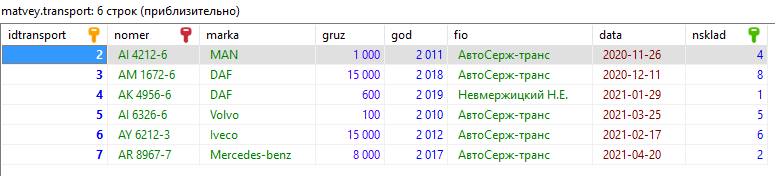


Рисунок 2.12

Таблица сотрудников

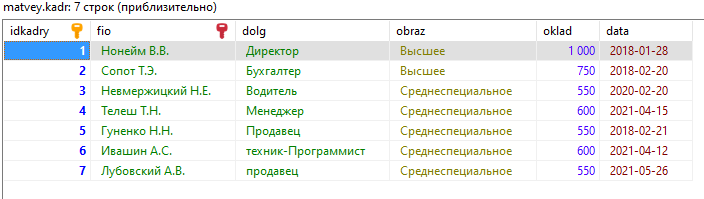
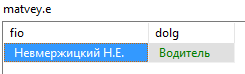


Рисунок 2.13

Результат выполнения запроса

****

## 2.3 Создание представлений

Представление — виртуальная (логическая) таблица, представляющая собой поименованный запрос (синоним к запросу), который будет подставлен как подзапрос при использовании представления. Представления используются для облегчения работы с базами данных, освобождая от частого написания громоздких запросов.

### Задание a

CREATE ALGORITHM=UNDEFINED

DEFINER=`root`@`localhost`

SQL SECURITY DEFINER

VIEW `matvey`.`a` AS

select `matvey`.`kadr`.`fio` AS `fio`,`matvey`.`kadr`.`data` AS `data` from `matvey`.`kadr`

where (`matvey`.`kadr`.`data` < '2021-01-01');

### Задание b

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost`

SQL SECURITY DEFINER

VIEW `matvey`.`b` (`marka`,`gruz`,`god`,`DATA`) AS select `matvey`.`transport`.`marka` AS `marka`,`matvey`.`transport`.`gruz` AS `gruz`,`matvey`.`transport`.`god` AS `god`,`matvey`.`transport`.`data` AS `DATA`

from `matvey`.`transport`

where ((`matvey`.`transport`.`data` < '2021-01-01') and (`matvey`.`transport`.`gruz` = 1000));

### Задание c

CREATE ALGORITHM=UNDEFINED

DEFINER=`root`@`localhost` SQL SECURITY DEFINER

VIEW `matvey`.`c` AS select `p`.`fio` AS `fio`,`p`.`adr` AS `uradr` from (`matvey`.`tovar` `t`

join `matvey`.`postav` `p`)

where ((`p`.`idpostav` = `t`.`idpostav`)

and (`t`.`sort` = 1)

and (`t`.`cena` = 500)

and (`t`.`naimen` = 'Телефон'));

### Задание d

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost`

SQL SECURITY DEFINER

VIEW `matvey`.`d` (`fio`,`bank`,`numchet`) AS select `p`.`fio` AS `fio`,`s`.`bank` AS `bank`,`s`.`numchet` AS `numchet` from (`matvey`.`schet` `s`

join `matvey`.`postav` `p`)

where ((`p`.`fio` = 'ЧТУП ТимБас')

and (`p`.`idschet` = `s`.`idschet`));

### Задание e

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost`

SQL SECURITY DEFINER

VIEW `matvey`.`e` AS

select `matvey`.`kadr`.`fio` AS `fio`,`matvey`.`kadr`.`dolg` AS `dolg` from (`matvey`.`kadr`

join `matvey`.`transport` on((`matvey`.`kadr`.`fio` = `matvey`.`transport`.`fio`)));

## 2.4. Создание индексов

Анализ готовых запросов показывает, что для повышения эффективности работы с данными необходимо создать индексы для наиболее употребительных полей таблиц. Создадим индексы для наиболее используемых запросов:

CREATE INDEX `I1`

ON matvey.firma(nazvan)

CREATE UNIQUE INDEX `i2`

ON matvey.kadry(fio)

CREATE UNIQUE INDEX `i3`

ON matvey.klient(fio)

CREATE UNIQUE INDEX `i4`

ON matvey.postav(fio)

CREATE INDEX `i5`

ON matvey.tovar(naimen\_tovara)

CREATE UNIQUE INDEX `i5`

ON matvey.transport(nomer)

Остальные таблицы уже имеют индексы по интересующим нас полям так как на них ссылаются по этим полям таблицы со внешним ключом.

# ЗАКЛЮЧЕНИЕ

Основной задачей создания любой БД является упрощение использования больших массивов информации. БД позволяют собирать, хранить, обновлять и выводить информацию в понятной пользователю форме.

Во время выполнения курсовой работы была разработана и реализована база данных "Фирма 13", которая в достаточной мере упрощает и ускоряет работу с данными, и тем самым сокращает время сотрудников, затраченное на поиск и обработку данных. Полученная в ходе создания курсового проекта база данных позволяет вести учет товаров на складах, выданных накладных, управление персоналом фирмы, контроль заказов, а в случаях особой необходимости, в нее могут быть внесены дополнительные средства формирования данных.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Дейт, К. Введение в системы баз данных / К. Дейт. – 8-е изд. – Москва:Санкт-Петербург : Киев : Вильямс, 2017. – 1328 с.
2. Куликов С.С. Реляционные базы данных в примерах : практическое пособие для программистов и тестировщи- ков / С. С. Куликов. — Минск: Четыре четверти, 2020. — 424 с.
3. Коннолли Т., Бегг К. Базы данных: проектирование, реализация, сопровождение. Теория и практика. – 3-е изд.: Пер. с англ.: Уч. пос. – М.: Изд. дом "Вильямс", 2003. – 1440 с
4. Куликов С.С. Работа с MySQL, MS SQL Server и Oracle в примерах (ссылка для скачивания http://svyatoslav.biz/database\_book/)
5. Материалы для скачивания по SQL Server // Microsoft URL: https://www.microsoft.com/ru-ru/sql-server/sql-server-downloads (дата обращения: 05.01.2021).
6. Нормализация отношений. Шесть нормальных форм // Хабр URL: https://habr.com/ru/post/254773/ (дата обращения: 05.01.2021).
7. Технологии баз данных. Лекция 3. Модель «Сущность-связь». URL: https://docplayer.ru/27886777-Model-sushchnost-svyaz-tehnologii-baz-dannyh-lekciya-3.html (дата обращения: 05.01.2021).
8. Шениг Г.Ю. PostgresSQL 11. Мастерство разработки / пер. с англ. А.А. Слинкина. – Москва: ДМК Пресс, 2019. – 352 с.
9. Entity Relationship Diagram. URL: https://plantuml.com/ru/ie-diagram (дата обращения: 03.01.2021).
10. MySQL 8 для больших данных / пер. с англ. А.В. Логунова / Чаллавала Ш., Лакхатария Дж., Мехта Ч., Патель К. - Москва: ДМК Пресс, 2018. – 226 с

# Приложение А

Дамп базы данных

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

-- Хост: 127.0.0.1

-- Версия сервера: 8.0.25 - MySQL Community Server - GPL

-- Операционная система: Win64

-- HeidiSQL Версия: 11.2.0.6213

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

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET NAMES utf8 \*/;

/\*!50503 SET NAMES utf8mb4 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

-- Дамп структуры базы данных matvey

DROP DATABASE IF EXISTS `matvey`;

CREATE DATABASE IF NOT EXISTS `matvey` /\*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;

USE `matvey`;

-- Дамп структуры для таблица matvey.client

DROP TABLE IF EXISTS `client`;

CREATE TABLE IF NOT EXISTS `client` (

`idklient` int NOT NULL AUTO\_INCREMENT,

`fio` varchar(250) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL DEFAULT 'Не указано',

`adr` varchar(500) DEFAULT NULL,

`idschet` int DEFAULT NULL,

PRIMARY KEY (`idklient`),

UNIQUE KEY `idclient` (`idklient`),

UNIQUE KEY `i3` (`fio`),

KEY `FKschet` (`idschet`),

CONSTRAINT `FKschet` FOREIGN KEY (`idschet`) REFERENCES `schet` (`idschet`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

-- Дамп данных таблицы matvey.client: ~10 rows (приблизительно)

DELETE FROM `client`;

/\*!40000 ALTER TABLE `client` DISABLE KEYS \*/;

INSERT INTO `client` (`idklient`, `fio`, `adr`, `idschet`) VALUES

(1, 'Ситник М.И.', 'г. Бобруйск, ул. Крылова, 152', 14),

(2, 'Микаилов В.В.', 'г. Минск, ул. Радиальная, 11', 11),

(3, 'Туля Е.О.', 'г. Жодино, ул. Неманская, 15-6', 13),

(4, 'Бочаров М.Н.', 'г. Бобруйск, ул. Минская, 14-2', 19),

(5, 'Плохотский И.М.', 'г. Рогачев, ул. Юбилейная, 23', 12),

(6, 'Бобр Ю.Н.', 'г. Мозырь, ул.Б.Юности, 68', 15),

(7, 'Кузьменко Е.В.', 'г.Гомель, ул.Б.Хмельницкого 72 ', 17),

(8, 'Стаховский В.В.', 'г.Калинковичи, ул.Гастело, 33', 16),

(9, 'Фещенко В.П.', 'г.Речица, ул.8 марта, 12', 20),

(10, 'Комисаров Т.В.', 'г.Житковичи, ул.Пушкина,69', 18);

/\*!40000 ALTER TABLE `client` ENABLE KEYS \*/;

-- Дамп структуры для таблица matvey.firma

DROP TABLE IF EXISTS `firma`;

CREATE TABLE IF NOT EXISTS `firma` (

`nazvan` varchar(500) NOT NULL DEFAULT '',

`uradr` varchar(500) NOT NULL DEFAULT '',

`phone` varchar(50) NOT NULL DEFAULT '',

`idschet` int NOT NULL,

KEY `FKschetf` (`idschet`),

KEY `I1` (`nazvan`),

CONSTRAINT `FKschetf` FOREIGN KEY (`idschet`) REFERENCES `schet` (`idschet`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='Описание фирмы';

-- Дамп данных таблицы matvey.firma: ~0 rows (приблизительно)

DELETE FROM `firma`;

/\*!40000 ALTER TABLE `firma` DISABLE KEYS \*/;

INSERT INTO `firma` (`nazvan`, `uradr`, `phone`, `idschet`) VALUES

('Инспектор гаджет', 'г.Мозырь ул.Чапаева 28', '+375336512162', 4);

/\*!40000 ALTER TABLE `firma` ENABLE KEYS \*/;

-- Дамп структуры для таблица matvey.kadr

DROP TABLE IF EXISTS `kadr`;

CREATE TABLE IF NOT EXISTS `kadr` (

`idkadry` int NOT NULL AUTO\_INCREMENT,

`fio` varchar(250) NOT NULL,

`dolg` varchar(500) DEFAULT NULL,

`obraz` enum('Высшее','Среднеспециальное','Среднее','Базовое') NOT NULL,

`oklad` float(12,0) NOT NULL,

`data` date NOT NULL,

PRIMARY KEY (`idkadry`),

UNIQUE KEY `idkadr` (`idkadry`),

UNIQUE KEY `i2` (`fio`)

) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='Сотрудники фирмы';

-- Дамп данных таблицы matvey.kadr: ~7 rows (приблизительно)

DELETE FROM `kadr`;

/\*!40000 ALTER TABLE `kadr` DISABLE KEYS \*/;

INSERT INTO `kadr` (`idkadry`, `fio`, `dolg`, `obraz`, `oklad`, `data`) VALUES

(1, 'Нонейм В.В.', 'Директор', 'Высшее', 1000, '2018-01-28'),

(2, 'Сопот Т.Э.', 'Бухгалтер', 'Высшее', 750, '2018-02-20'),

(3, 'Невмержицкий Н.Е.', 'Водитель', 'Среднеспециальное', 550, '2020-02-20'),

(4, 'Телеш Т.Н.', 'Менеджер ', 'Среднеспециальное', 600, '2021-04-15'),

(5, 'Гуненко Н.Н.', 'Продавец', 'Среднеспециальное', 550, '2018-02-21'),

(6, 'Ивашин А.С.', 'техник-Программист', 'Среднеспециальное', 600, '2021-04-12'),

(7, 'Лубовский А.В.', 'продавец', 'Среднеспециальное', 550, '2021-05-26');

/\*!40000 ALTER TABLE `kadr` ENABLE KEYS \*/;

-- Дамп структуры для таблица matvey.naklad

DROP TABLE IF EXISTS `naklad`;

CREATE TABLE IF NOT EXISTS `naklad` (

`idnakladnaya` int NOT NULL AUTO\_INCREMENT,

`idtovar` int NOT NULL,

`kolvo` int unsigned NOT NULL,

`summa` float(12,0) unsigned NOT NULL,

`idklient` int NOT NULL,

PRIMARY KEY (`idnakladnaya`),

UNIQUE KEY `idnaklad` (`idnakladnaya`),

KEY `FKtovarn` (`idtovar`),

KEY `FK\_nakladnaya\_klient` (`idklient`),

CONSTRAINT `FK\_nakladnaya\_klient` FOREIGN KEY (`idklient`) REFERENCES `client` (`idklient`),

CONSTRAINT `FKtovarn` FOREIGN KEY (`idtovar`) REFERENCES `tovar` (`idtovar`)

) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='Накладные на товар';

-- Дамп данных таблицы matvey.naklad: ~5 rows (приблизительно)

DELETE FROM `naklad`;

/\*!40000 ALTER TABLE `naklad` DISABLE KEYS \*/;

INSERT INTO `naklad` (`idnakladnaya`, `idtovar`, `kolvo`, `summa`, `idklient`) VALUES

(1, 1, 3, 60, 1),

(2, 2, 5, 100, 2),

(3, 6, 1, 55, 6),

(4, 7, 1, 20, 4),

(5, 10, 5, 55, 7);

/\*!40000 ALTER TABLE `naklad` ENABLE KEYS \*/;

-- Дамп структуры для таблица matvey.postav

DROP TABLE IF EXISTS `postav`;

CREATE TABLE IF NOT EXISTS `postav` (

`idpostav` int NOT NULL AUTO\_INCREMENT,

`fio` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL DEFAULT 'Не указано',

`adr` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL DEFAULT 'Не указано',

`idschet` int DEFAULT NULL,

PRIMARY KEY (`idpostav`),

UNIQUE KEY `idpostav` (`idpostav`),

UNIQUE KEY `i4` (`fio`),

KEY `FKschetp` (`idschet`),

CONSTRAINT `FKschetp` FOREIGN KEY (`idschet`) REFERENCES `schet` (`idschet`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='Поставщики';

-- Дамп данных таблицы matvey.postav: ~10 rows (приблизительно)

DELETE FROM `postav`;

/\*!40000 ALTER TABLE `postav` DISABLE KEYS \*/;

INSERT INTO `postav` (`idpostav`, `fio`, `adr`, `idschet`) VALUES

(1, 'ЧТУП ТимБас', 'г.Гродно, ул.Суворова,254', 4),

(2, 'ЧТУП РОТАКОН', 'г.Полоцк, ул.Советская 22', 3),

(3, 'ЧТУП Бурштат', 'г.Витебск, ул.Ф.Скорины, 59', 5),

(4, 'ЧТУП ТехЦентр', 'г.Солигорск, ул.Я.Коласа,76', 1),

(5, 'ЧТУП Бимтэк', 'г.Брест,ул.Ленина,43', 2),

(6, 'ОДО Белтехстрой', 'г.Пинск,ул.Советских Пограничников,92', 7),

(7, 'ООО Горизонт', 'г.Лида,ул.Фолюш,67', 8),

(8, 'ООО Алтерм', 'г.Новополоцк,ул.Е.Полоцкой,41', 10),

(9, 'ОДО Белросимпорт', 'г.Мозырь,ул.Мостовая,3', 6),

(10, 'ЧТУП Комплайт', 'г.Кобрин,ул.Фрунзе,10', 9);

/\*!40000 ALTER TABLE `postav` ENABLE KEYS \*/;

-- Дамп структуры для таблица matvey.schet

DROP TABLE IF EXISTS `schet`;

CREATE TABLE IF NOT EXISTS `schet` (

`idschet` int NOT NULL AUTO\_INCREMENT,

`bank` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL DEFAULT 'Не указано',

`numchet` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

`status` int NOT NULL DEFAULT '0',

`data` date NOT NULL,

`creditstat` enum('Y','N') NOT NULL,

`summa` float(12,0) DEFAULT NULL,

PRIMARY KEY (`idschet`),

UNIQUE KEY `idschet` (`idschet`)

) ENGINE=InnoDB AUTO\_INCREMENT=21 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

-- Дамп данных таблицы matvey.schet: ~20 rows (приблизительно)

DELETE FROM `schet`;

/\*!40000 ALTER TABLE `schet` DISABLE KEYS \*/;

INSERT INTO `schet` (`idschet`, `bank`, `numchet`, `status`, `data`, `creditstat`, `summa`) VALUES

(1, 'Паритетбанк', '681735413', 1250, '2020-03-19', '', 3300),

(2, 'Альфа-Банк', '901235723', 0, '2019-10-23', '', 5200),

(3, 'ВТББанк', '561384613', 0, '2020-08-12', '', 18000),

(4, 'Белинвестбанк', '719837645', 200, '2018-02-24', '', 4000),

(5, 'Белгазпромбанк', '564718379', 0, '2016-09-28', '', 8700),

(6, 'Приорбанк', '681739138', 820, '2017-05-11', '', 2200),

(7, 'МоскваМинск', '000025215', 0, '2019-09-24', '', 1000),

(8, 'Беларусбанк', '985578067', 765, '2016-06-06', '', 4000),

(9, 'Беларусбанк', '641923519', 35, '2018-07-20', '', 2314),

(10, 'Белинвестбанк', '412455125', 78, '2019-08-15', '', 100),

(11, 'ВТББанк', '214758941', 532, '2018-02-17', '', 1000),

(12, 'БПС-Сбербанк', '124789812', 79, '2019-09-26', '', 700),

(13, 'Приорбанк', '709123487', 7, '2018-08-30', '', 93),

(14, 'Белагропромбанк', '908746123', 90, '2015-07-01', '', 100),

(15, 'Альфа-Банк', '679854334', 78, '2016-02-27', '', 432),

(16, 'Банк БелВЭБ', '563354363', 77, '2018-08-29', '', 90),

(17, 'Дабрабыт банк', '439875354', 300, '2019-08-19', '', 451),

(18, 'Банк Решение', '809645352', 456, '2015-07-20', '', 706),

(19, 'БНБ-банк', '894213567', 0, '2017-01-28', '', 9000),

(20, 'БТА банк', '765433653', 0, '2018-08-20', '', 1000);

/\*!40000 ALTER TABLE `schet` ENABLE KEYS \*/;

-- Дамп структуры для таблица matvey.sklad

DROP TABLE IF EXISTS `sklad`;

CREATE TABLE IF NOT EXISTS `sklad` (

`Nsklad` int unsigned NOT NULL AUTO\_INCREMENT,

`adress` varchar(5000) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

PRIMARY KEY (`Nsklad`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

-- Дамп данных таблицы matvey.sklad: ~10 rows (приблизительно)

DELETE FROM `sklad`;

/\*!40000 ALTER TABLE `sklad` DISABLE KEYS \*/;

INSERT INTO `sklad` (`Nsklad`, `adress`) VALUES

(1, 'г.Орша,ул. Пушкина,13'),

(2, 'г. Минск, ул. Первомайская,44'),

(3, 'г.Гомель, ул. Б.Хмельницкого,13'),

(4, 'г. Бобруйск, ул. Холмогорская,74'),

(5, 'г. Могилев, ул.Б.Непокоренных, 19'),

(6, 'г.Мозырь, ул.Фрунзе, 2'),

(7, 'г.Новоплоцк, ул.Социалистическая 7'),

(8, 'г.Витебск, ул.50 лет СССР'),

(9, 'г.Солигорск, ул.Рябиновая 7'),

(10, 'г.Гродно, ул.Комунальная 16');

/\*!40000 ALTER TABLE `sklad` ENABLE KEYS \*/;

-- Дамп структуры для таблица matvey.tovar

DROP TABLE IF EXISTS `tovar`;

CREATE TABLE IF NOT EXISTS `tovar` (

`idtovar` int NOT NULL AUTO\_INCREMENT,

`naimen` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL DEFAULT 'Не указано',

`ed\_izm` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL DEFAULT 'Не указано',

`cena` int NOT NULL,

`idpostav` int NOT NULL,

`sklad` int unsigned NOT NULL,

`sort` int NOT NULL,

PRIMARY KEY (`idtovar`),

UNIQUE KEY `idtovar` (`idtovar`),

KEY `FKpostavt` (`idpostav`),

KEY `FK\_tovar\_sklad` (`sklad`),

KEY `i5` (`naimen`) USING BTREE,

CONSTRAINT `FK\_tovar\_sklad` FOREIGN KEY (`sklad`) REFERENCES `sklad` (`Nsklad`),

CONSTRAINT `FKpostavt` FOREIGN KEY (`idpostav`) REFERENCES `postav` (`idpostav`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='Список товаров нормализованный';

-- Дамп данных таблицы matvey.tovar: ~10 rows (приблизительно)

DELETE FROM `tovar`;

/\*!40000 ALTER TABLE `tovar` DISABLE KEYS \*/;

INSERT INTO `tovar` (`idtovar`, `naimen`, `ed\_izm`, `cena`, `idpostav`, `sklad`, `sort`) VALUES

(1, 'Ноутбук', 'шт', 1499, 1, 1, 0),

(2, 'Игровая приставка', 'шт', 2455, 2, 1, 0),

(3, 'Мышка', 'шт', 15, 1, 2, 2),

(4, 'Фен', 'шт', 59, 2, 2, 1),

(5, 'Телвизор', 'шт', 4500, 1, 2, 3),

(6, 'Обогреватель', 'шт', 100, 4, 1, 2),

(7, 'Чайник', 'шт', 40, 6, 4, 3),

(8, 'Монитор', 'шт', 700, 7, 3, 2),

(9, 'Телефон', 'шт', 500, 4, 2, 1),

(10, 'Миксер', 'шт', 55, 6, 3, 2);

/\*!40000 ALTER TABLE `tovar` ENABLE KEYS \*/;

-- Дамп структуры для таблица matvey.transport

DROP TABLE IF EXISTS `transport`;

CREATE TABLE IF NOT EXISTS `transport` (

`idtransport` int NOT NULL AUTO\_INCREMENT,

`nomer` varchar(45) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

`marka` varchar(45) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

`gruz` float(12,0) NOT NULL,

`god` int NOT NULL,

`fio` varchar(450) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

`data` date DEFAULT NULL,

`nsklad` int unsigned DEFAULT NULL,

PRIMARY KEY (`idtransport`) USING BTREE,

UNIQUE KEY `i5` (`nomer`),

KEY `FK\_transport\_sklad` (`nsklad`),

CONSTRAINT `FK\_transport\_sklad` FOREIGN KEY (`nsklad`) REFERENCES `sklad` (`Nsklad`)

) ENGINE=InnoDB AUTO\_INCREMENT=8 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

-- Дамп данных таблицы matvey.transport: ~0 rows (приблизительно)

DELETE FROM `transport`;

/\*!40000 ALTER TABLE `transport` DISABLE KEYS \*/;

INSERT INTO `transport` (`idtransport`, `nomer`, `marka`, `gruz`, `god`, `fio`, `data`, `nsklad`) VALUES

(2, 'AI 4212-6', 'MAN', 1000, 2011, 'АвтоСерж-транс', '2020-11-26', 4),

(3, 'AM 1672-6', 'DAF', 15000, 2018, 'АвтоСерж-транс', '2020-12-11', 8),

(4, 'AK 4956-6', 'DAF', 600, 2019, 'Невмержицкий Н.Е.', '2021-01-29', 1),

(5, 'AI 6326-6', 'Volvo', 100, 2010, 'АвтоСерж-транс', '2021-03-25', 5),

(6, 'AY 6212-3', 'Iveco', 15000, 2012, 'АвтоСерж-транс', '2021-02-17', 6),

(7, 'AR 8967-7', 'Mercedes-benz', 8000, 2017, 'АвтоСерж-транс', '2021-04-20', 2);

/\*!40000 ALTER TABLE `transport` ENABLE KEYS \*/;

-- Дамп структуры для таблица matvey.zakaz

DROP TABLE IF EXISTS `zakaz`;

CREATE TABLE IF NOT EXISTS `zakaz` (

`idzakaz` int NOT NULL AUTO\_INCREMENT,

`data` date DEFAULT NULL,

`idclient` int NOT NULL,

`idtovar` int NOT NULL,

`kolvo` int NOT NULL DEFAULT '0',

`opl` enum('Y','N') DEFAULT NULL,

PRIMARY KEY (`idzakaz`) USING BTREE,

UNIQUE KEY `id\_zakaz` (`idzakaz`) USING BTREE,

KEY `FKtovar` (`idtovar`) USING BTREE,

KEY `FKclientz` (`idclient`) USING BTREE,

CONSTRAINT `FKclientz` FOREIGN KEY (`idclient`) REFERENCES `client` (`idklient`),

CONSTRAINT `FKtovar` FOREIGN KEY (`idtovar`) REFERENCES `tovar` (`idtovar`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='Список заказов п 10';

-- Дамп данных таблицы matvey.zakaz: ~0 rows (приблизительно)

DELETE FROM `zakaz`;

/\*!40000 ALTER TABLE `zakaz` DISABLE KEYS \*/;

INSERT INTO `zakaz` (`idzakaz`, `data`, `idclient`, `idtovar`, `kolvo`, `opl`) VALUES

(1, '2021-05-19', 2, 2, 3, NULL),

(2, '2021-05-19', 4, 3, 1, NULL),

(3, '2021-05-22', 5, 3, 1, NULL),

(4, '2019-03-16', 6, 6, 1, NULL);

/\*!40000 ALTER TABLE `zakaz` ENABLE KEYS \*/;

-- Дамп структуры для представление matvey.a

DROP VIEW IF EXISTS `a`;

-- Удаление временной таблицы и создание окончательной структуры представления

DROP TABLE IF EXISTS `a`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY DEFINER VIEW `a` AS select `kadr`.`fio` AS `fio`,`kadr`.`data` AS `data` from `kadr` where (`kadr`.`data` < '2021-01-01');

-- Дамп структуры для представление matvey.b

DROP VIEW IF EXISTS `b`;

-- Удаление временной таблицы и создание окончательной структуры представления

DROP TABLE IF EXISTS `b`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY DEFINER VIEW `b` (`marka`,`gruz`,`god`,`DATA`) AS select `transport`.`marka` AS `marka`,`transport`.`gruz` AS `gruz`,`transport`.`god` AS `god`,`transport`.`data` AS `DATA` from `transport` where ((`transport`.`data` < '2021-01-01') and (`transport`.`gruz` = 1000));

-- Дамп структуры для представление matvey.c

DROP VIEW IF EXISTS `c`;

-- Удаление временной таблицы и создание окончательной структуры представления

DROP TABLE IF EXISTS `c`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY DEFINER VIEW `c` AS select `p`.`fio` AS `fio`,`p`.`adr` AS `uradr` from (`tovar` `t` join `postav` `p`) where ((`p`.`idpostav` = `t`.`idpostav`) and (`t`.`sort` = 1) and (`t`.`cena` = 500) and (`t`.`naimen` = 'Телефон'));

-- Дамп структуры для представление matvey.d

DROP VIEW IF EXISTS `d`;

-- Удаление временной таблицы и создание окончательной структуры представления

DROP TABLE IF EXISTS `d`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY DEFINER VIEW `d` (`fio`,`bank`,`numchet`) AS select `p`.`fio` AS `fio`,`s`.`bank` AS `bank`,`s`.`numchet` AS `numchet` from (`schet` `s` join `postav` `p`) where ((`p`.`fio` = 'ЧТУП ТимБас') and (`p`.`idschet` = `s`.`idschet`));

-- Дамп структуры для представление matvey.e

DROP VIEW IF EXISTS `e`;

-- Удаление временной таблицы и создание окончательной структуры представления

DROP TABLE IF EXISTS `e`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY DEFINER VIEW `e` AS select `kadr`.`fio` AS `fio`,`kadr`.`dolg` AS `dolg` from (`kadr` join `transport` on((`kadr`.`fio` = `transport`.`fio`)));

/\*!40101 SET SQL\_MODE=IFNULL(@OLD\_SQL\_MODE, '') \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=IFNULL(@OLD\_FOREIGN\_KEY\_CHECKS, 1) \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40111 SET SQL\_NOTES=IFNULL(@OLD\_SQL\_NOTES, 1) \*/;