ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 2](#_Toc73737390)

[ГЛАВА 1 4](#_Toc73737391)

[1.1 Анализ предметной области 4](#_Toc73737392)

[1.2 Построение логической структуры базы данных; 4](#_Toc73737393)

[1.3 Преобразование ER–диаграммы в схему базы данных 5](#_Toc73737394)

[1.4 Нормализация отношений 7](#_Toc73737395)

[1.5 Определение ограничений целостности 14](#_Toc73737396)

[2 Практическая часть – реализация базы данных в выбранной СУБД 16](#_Toc73737397)

[2.1 Реализация базы данных 16](#_Toc73737398)

[2.2 Создание запросов 19](#_Toc73737399)

[Задание а. 19](#_Toc73737400)

[Задание b 21](#_Toc73737401)

[Задание c. 24](#_Toc73737402)

[Задание d. 25](#_Toc73737403)

[Задание e. 28](#_Toc73737404)

[2.3 Создание представлений 29](#_Toc73737405)

[Задание a 30](#_Toc73737406)

[Задание b 30](#_Toc73737407)

[Задание c 31](#_Toc73737408)

[Задание d 31](#_Toc73737409)

[Задание e 32](#_Toc73737410)

[2.4. Создание индексов 32](#_Toc73737411)

[ЗАКЛЮЧЕНИЕ 33](#_Toc73737412)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 34](#_Toc73737413)

# ВВЕДЕНИЕ

В текущее время активно развивается отрасль – информационная индустрия, связанная с производством технических средств, методов и технологий для решения разного рода задач. Эта индустрия тесно связана с развитием компьютерных технологий. Возрастает спрос на новые технологии, от человека требуется способность к интеллектуальному труду и творчеству. В результате появились противоречия между ограниченными возможностями человека по восприятию и переработке информации и существующими массивами хранящейся и передаваемой информации.

Для решения подобных проблем применяются автоматизированные базы данных. Они стали неотъемлемой частью практически всех компьютерных систем – от отрасти до отдельного предприятия.

Базы данных (БД) — это совокупность структур, предназначенных для хранения больших объемов информации и программных модулей, осуществляющих управление данными, их выборку, сортировку и другие подобные действия.

СУБД — это программное обеспечение или служба, используемая для создания и управления базами данных на основе реляционной модели. Когда им нужно получить доступ к данным, они подключаются к серверу СУБД. SQL Server является одним из многих вариантов программного обеспечения СУБД.

Тема курсовой работы: Информационная система «Фирма VII»  
В качестве предметной области проектирования БД выбрана торговая фирма, которая осуществляет выполнение заказов, поступающих от клиентов.

Создание базы данных информационная система «Фирма VII», которая осуществляет выполнение заказов, поступающих от клиентов, существенно ускорит процесс выполнения заказов.

Преимущество базы данных информационная система «Фирма VII»:

* быстрое внесение данных о заказе поступающего от клиента ускоряет процесс выдачи заказа;
* поиск по базе данных в разы быстрее поиска информации о заказе в бумажной документации.
* готовая база данных используется более быстрее для анализа и процесса приема заказов.  
   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. Произвести 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).

# ГЛАВА 1 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

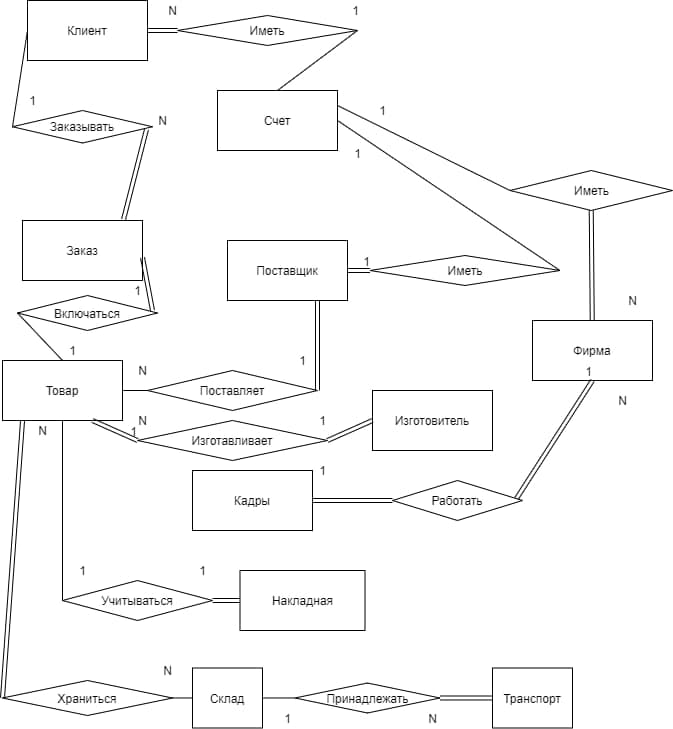
10.3аказ (№заказа, дата заказа, Ф.И.О. клиента, №счета клиента, наименование товара, количество, цена товара)

## 1.2 Построение логической структуры базы данных

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

Рисунок 1.1

ER-диаграмма фирмы.



## 1.3 Преобразование ER–диаграммы в схему базы данных

На основании схемы базы данных уточним ER-диаграмму и уже на основе уточненной диаграммы создадим базу данных.

Преобразование ER–диаграммы в схему БД выполним 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.

Введём первичный ключ отношения ФИРМА им будет атрибут Название, в дальнейшем он нам не понадобится, так как фирма у нас всего одна.

Таблица 1.1 Отношение ФИРМА

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Название | firma | C(300) | Обязательное поле, первичный ключ |
| Юр. адрес | uradr | C(300) | Обязательное поле |
| Телефон | telefon | C(30) | Необязательное поле |
| № счета банка | schet | C(50) | Обязательное поле |
| Наименование товара | tovar | C(100) | Необязательное поле |
| Цена товара | cena | N | Необязательное поле |

Хорошим ключом отношения СКЛАД является атрибут Номер, он уникален, поэтому мы и выбираем его в качестве первичного ключа.

Таблица 1.2 Отношение СКЛАД

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| №склада | n | N | Первичный ключ |
| Адрес склада | adres | 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, длина этой текстовой строки довольно велика, введем суррогатный ключ N и выбираем его в качестве первичного ключа.

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, перечисляемое |
| Оклад | salary | N | Обязательное поле |
| Дата принятия | data | D | Обязательное поле |
| Идентификатор | n | N | Первичный ключ суррогатный |

Потенциальным ключом отношения ПОСТАВЩИК является атрибут Номер поставщика, у нас он будет уникальным, поэтому мы выбираем его в качестве первичного ключа.

Таблица 1.4 Схема отношения ПОСТАВЩИК

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| №поставщика | N | N | Первичный ключ, уникальное |
| 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.

Таблица 1.5 Схема отношения ТРАНСПОРТ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| №склада | sklad | 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 | fio | C(500) | Обязательное поле |
| Дата перев. | data | D | Обязательное поле |
| Идентификатор | n | N | Первичный ключ, уникальное |

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

Таблица 1.6 Схема отношения НАКЛАДНАЯ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| №накладной | n | 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 |
| Клиент | nklient | N | Обязательное поле |
| Сумма | summa | N | Обязательное поле |

Ключом отношения ТОВАР можно выбрать наименование, однако это длинная строка, поэтому мы вводим суррогатный ключ N и выбираем его в качестве первичного ключа.

Таблица 1.7 Схема отношения ТОВАР

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Наименование | name | C(500) | Обязательное поле |
| Ед. изм. | ed | C(50) | Обязательное поле |
| Сорт | sort | N | Обязательное поле |
| Цена | cena | N | Обязательное поле |
| Поставщик | postav | C(500) | Обязательное поле |
| Адрес поставщика | adr | C(500) | Обязательное поле |
| Изготовитель | izgot | N | Обязательное поле |
| Идентификатор | n | N | Первичный ключ, уникальное |

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

Таблица 1.8 Схема отношения КЛИЕНТ

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

Таблица 1. Продолжение

|  |  |  |  |
| --- | --- | --- | --- |
| 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 N и выбираем его в качестве первичного ключа.

Таблица 1.9 Схема отношения СЧЕТ

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

Ключом отношения ЗАКАЗ является атрибут Номер заказа, он уникален, поэтому мы выбираем его в качестве первичного ключа.

Таблица 1.10 Схема отношения ЗАКАЗ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| № заказа | n | 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 |

## 1.4 Нормализация отношений

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.

Таблица 1.11 Схема отношения СЧЕТ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Идентификатор | n | N | Первичный ключ |
| № счета | numb | C(100) | Обязательное поле |
| Наименов. банка | bank | C(100) | Обязательное поле |
| Состояние счета | sost | N | Обязательное поле |
| Дата | data | D | Обязательное поле |
| Наличие кредита | cr | 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.

Таблица 1.12 Схема отношения ПОСТАВЩИК

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| № поставщика | n | N | Первичный ключ, уникальное |
| ФИО пост. | nazv | C(500) | Обязательное поле |
| Юр. адр. | uradr | C(500) | Обязательное поле |
| №счета | schet | N | Обязательное поле, уникальное, внешний ключ к СЧЕТ |

Таблица 1.13 Схема отношения КЛИЕНТ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| № клиента | n | N | Первичный ключ, уникальное |
| ФИО клиента | fio | C(100) | Обязательное поле |
| Юр. адр. | adres | C(300) | Обязательное поле |
| №счета | schet | 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 sklad и с поставщиком IZGOT. И необходимо установить связь с отношением ИЗГОТОВИТЕЛЬ.

Таблица 1.14 Схема отношения ТОВАР

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Идентификатор | idtovar | N | Первичный ключ |
| Наименование | name | C(500) | Обязательное поле |

Таблица 1.14 продолжение

|  |  |  |  |
| --- | --- | --- | --- |
| Ед. изм. | ed | C(50) | Обязательное поле |
| Сорт | sort | N | Обязательное поле |
| Цена | cena | N | Обязательное поле |
| Идентификатор склад | sklad | N | Обязательное поле, внешний ключ к СКЛАД |
| Идентификатор изготовителя | izgot | N | Обязательное поле, внешний ключ к ИЗГОТОВИТЕЛЬ |
| Количество | kolvo | N | Обязательное поле, больше нуля |
| Идентификатор поставщика | postav | 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. Цена товара может быть получена транзитивно.

Таблица 1.15 Схема отношения ЗАКАЗ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| № заказа | n | N | Первичный ключ |
| Дата заказа | data | D | Обязательное поле |
| Идентификатор клиента | client | N | Обязательное поле, внешний ключ к КЛИЕНТ |
| Идентификатор товара | tovar | 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.

Таблица 1.16 Схема отношения НАКЛАДНАЯ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| №накладной | n | N | Первичный ключ |
| Идентификатор товара | ntovar | N | Обязательное поле, внешний ключ к ТОВАР |

Таблица 1.16 продолжение

|  |  |  |  |
| --- | --- | --- | --- |
| Клиент | nklient | N | Обязательное поле, внешний ключ к КЛИЕНТ |
| Кол-во | kolvo | N | Обязательное поле |
| Сумма | summa | N | Обязательное поле |

№ 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.

Таблица 1.17 Схема отношения СКЛАД

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| №склада | n | 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. Телефон условимся считать необязательным полем

Таблица 1.18 Схема отношения ФИРМА

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Название | firma | C(500) | Обязательное поле |
| Юр. адрес | uradr | C(500) | Обязательное поле |
| Телефон | telefon | C(50) | Необязательное поле |
| Идентификатор счета банка | schet | 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 N.

Таблица 1.19 Схема отношения КАДРЫ

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

Таблица1.19 продолжение

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

Первичный ключ отношения ТРАНСПОРТ атрибут N. № склада заменим связью с отношением СКЛАД.

Таблица 1.20 Схема отношения ТРАНСПОРТ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Идентификатор | N | N | Первичный ключ |
| №склада | sklad | 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 |

Создадим справочную таблицу ИЗГОТОВИТЕЛЬ

Таблица 1.21 Схема отношения ИЗГОТОВИТЕЛЬ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Идентификатор | N | N | Первичный ключ |
| Название | name | C(200) | Обязательное поле |

## 1.5 Определение ограничений целостности

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

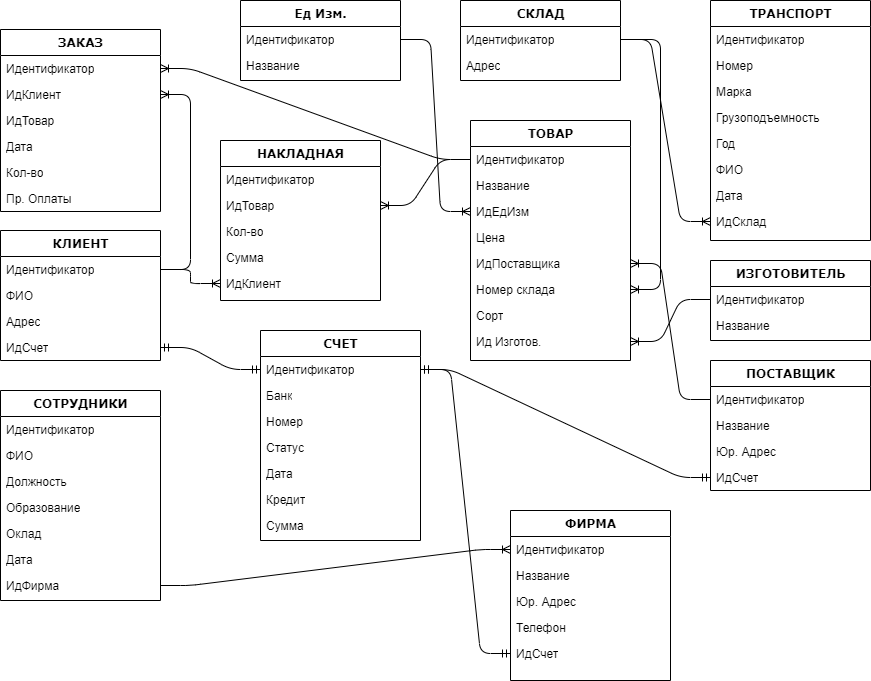
Атрибуты Зарплата отношения КАДРЫ, Количество и Сумма отношения НАКЛАДНАЯ, Количество и Сумма отношения СЧЕТ, Цена и Количество отношения ТОВАР, Год и Тоннаж отношения ТРАНСПОРТ, Количество и Оплачено отношения ЗАКАЗ могут принимать только неотрицательные значения.

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

Для преобразования ER–диаграммы в схему БД приведем уточненную ER–диаграмму, содержащую атрибуты сущностей (рисунок 1.2).

Рисунок 1.2

Уточненная ER-диаграмма фирмы.



# ГЛАВА 2 Реализация базы данных в выбранной СУБД

## 2.1 Реализация базы данных на языке DDL

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

Отношение СЧЕТ

CREATE TABLE schet (

n int NOT NULL AUTO\_INCREMENT,

bank varchar(50) NOT NULL DEFAULT 'Не указано',

numb varchar(50) NOT NULL,

sost int NOT NULL DEFAULT 0,

data date NOT NULL,

cr enum ('Y', 'N') NOT NULL,

summa float(12, 0) UNSIGNED DEFAULT NULL,

PRIMARY KEY (n),

UNIQUE INDEX idschet (n)

)

Отношение ПОСТАВЩИК

CREATE TABLE postav (

n int NOT NULL AUTO\_INCREMENT,

nazv varchar(500) NOT NULL DEFAULT 'Не указано',

uradr varchar(500) NOT NULL DEFAULT 'Не указано',

schet int DEFAULT NULL,

PRIMARY KEY (n),

UNIQUE INDEX idpostav (n)

)

Отношение КЛИЕНТ

CREATE TABLE klient (

n int NOT NULL AUTO\_INCREMENT,

fio varchar(250) NOT NULL DEFAULT 'Не указано',

adres varchar(500) DEFAULT NULL,

schet int DEFAULT NULL,

PRIMARY KEY (n),

UNIQUE INDEX idclient (n)

)

Отношение ТОВАР

CREATE TABLE tovar (

n int NOT NULL AUTO\_INCREMENT,

name varchar(500) NOT NULL DEFAULT 'Не указано',

ed varchar(50) NOT NULL DEFAULT 'Не указано',

cena int UNSIGNED NOT NULL,

postav int NOT NULL,

sklad int UNSIGNED NOT NULL,

Kolvo int UNSIGNED NOT NULL,

sort int UNSIGNED NOT NULL,

PRIMARY KEY (n),

UNIQUE INDEX idtovar (n)

)

Отношение ЗАКАЗ

CREATE TABLE zakaz (

n int NOT NULL AUTO\_INCREMENT,

data date NOT NULL,

client int NOT NULL,

tovar int NOT NULL,

kolvo int NOT NULL DEFAULT 0,

cena float(12, 0) NOT NULL,

summa float(12, 0) NOT NULL,

opl enum ('Y', 'N') NOT NULL,

PRIMARY KEY (n),

UNIQUE INDEX id\_zakaz (n)

)

Отношение НАКЛАДНАЯ

CREATE TABLE nakladnaya (

n int NOT NULL AUTO\_INCREMENT,

ntovar int NOT NULL,

kolvo int UNSIGNED NOT NULL,

summa float(12, 0) UNSIGNED NOT NULL,

nklient int NOT NULL,

PRIMARY KEY (n),

UNIQUE INDEX idnaklad (n)

)

Отношение СКЛАД

CREATE TABLE sklad (

N int UNSIGNED NOT NULL AUTO\_INCREMENT,

adres varchar(5000) NOT NULL,

PRIMARY KEY (N)

)

Отношение ФИРМА

CREATE TABLE firma (

firma varchar(500) NOT NULL DEFAULT '',

uradr varchar(500) NOT NULL DEFAULT '',

telefon varchar(50) NOT NULL DEFAULT '',

schet int NOT NULL

)

Отношение КАДРЫ

CREATE TABLE kadry (

N int NOT NULL AUTO\_INCREMENT,

fio varchar(250) NOT NULL,

dolg varchar(500) DEFAULT NULL,

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

salary float(12, 0) UNSIGNED NOT NULL,

data date NOT NULL,

PRIMARY KEY (N),

UNIQUE INDEX idkadr (N)

)

Отношение ТРАНСПОРТ

CREATE TABLE transport (

n int NOT NULL AUTO\_INCREMENT,

nomer varchar(45) NOT NULL,

marka varchar(45) NOT NULL,

gruz float(12, 0) NOT NULL,

god int NOT NULL,

fio varchar(150) NOT NULL,

data date NOT NULL,

sklad int UNSIGNED NOT NULL,

PRIMARY KEY (n)

)

Отношение ИЗГОТОВИТЕЛЬ

CREATE TABLE izgot (

`n` INT(10) NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(200) NOT NULL COLLATE 'utf8mb4\_0900\_ai\_ci',

PRIMARY KEY (`n`) USING BTREE

)

## 2.2 Создание запросов

### Задание а.

Какие предприятия изготавливают данный товар не выше данной цены?

Целью задания является получение списка изготовителей, у которых цена указанного товара ниже заданной.

Запишем запрос на языке DML:

SELECT i.name

FROM tovar t, izgot i

WHERE

i.n=t.izgot

AND t.name='Лампы'

and t.cena<30

Результат выполнения запроса представлен на рисунках 2.1 - 2.3.

Рисунок 2.1

Таблица товаров

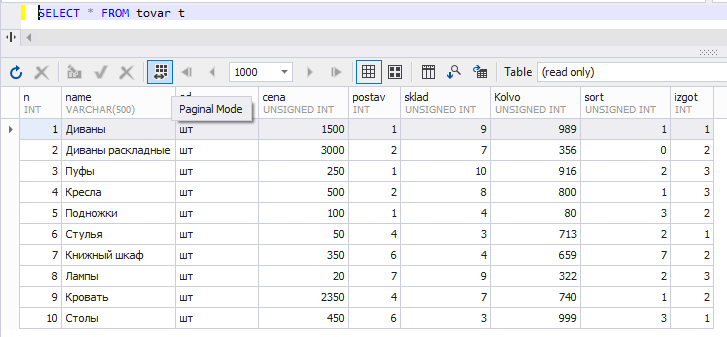


Рисунок 2.2

Таблица изготовителей

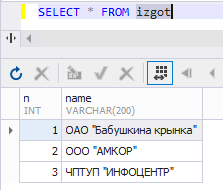
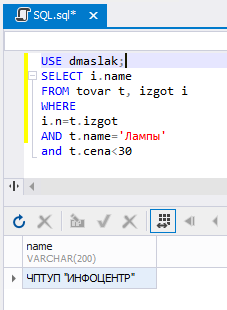


Рисунок 2.3

Результат выполнения запроса



### Задание b

Кто из владельцев счетов данного банка имеет кредиты на данную дату?

Входные данные задания: банк. Для получения необходимой информации нам необходимо несколько таблиц: таблица поставщиков, клиентов и нашей фирмы. Выберем все записи из трех таблиц у которых по ссылке на счет данные счета совпадают со входными даннями.

Запишем запрос на языке DML:

SELECT p.nazv AS naz

FROM postav p, schet s

WHERE

p.schet=s.n

AND s.bank='Bank BNP Paribas'

AND s.data='2015-11-18' AND s.cr='Y'

UNION

SELECT p.fio AS naz

FROM klient p, schet s

WHERE

p.schet=s.n

AND s.bank='Bank BNP Paribas'

AND s.data='2015-11-18' AND s.cr='Y'

union

SELECT p.firma AS naz

FROM firma p, schet s

WHERE

p.schet=s.n

AND s.bank='Bank BNP Paribas'

AND s.data='2015-11-18' AND s.cr='Y'

Результат выполнения запроса представлен на рисунках 2.4 - 2.7.

Рисунок 2.4

Таблица клиентов

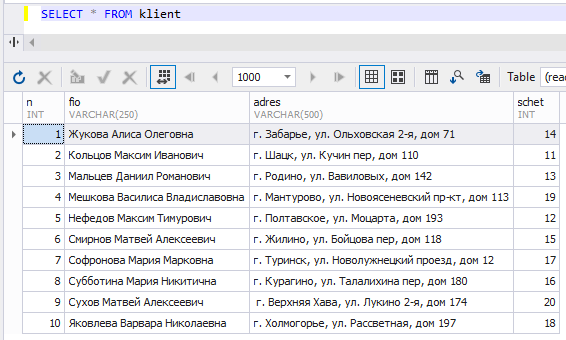


Рисунок 2.5

Таблица счетов

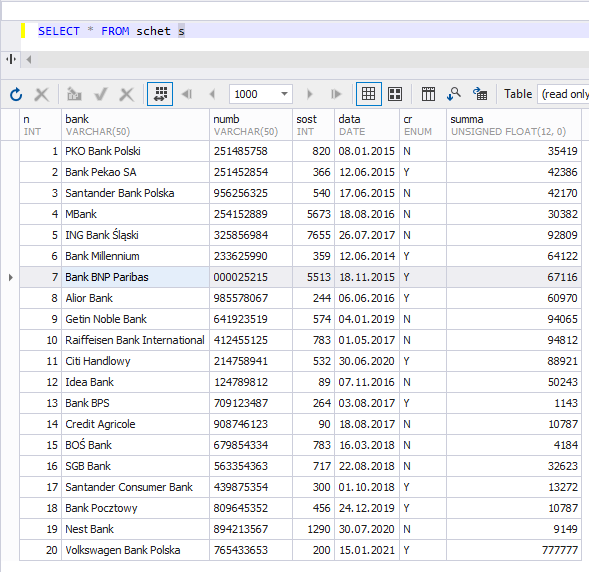


Рисунок 2.6

Таблица поставщиков

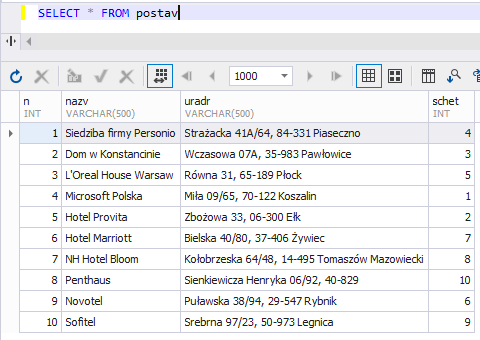
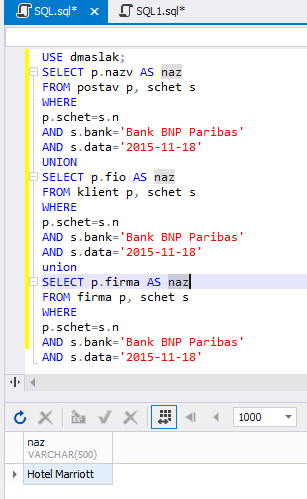


Рисунок 2.7

Результат выполнения запроса



### Задание c.

Определить юридический адрес и номера телефонов данной фирмы.

В задании есть входные данные: название фирмы, и имеются данные, которые нам необходимо выяснить, это адрес юридический и телефоны. Для получения необходимой информации нам необходимо задействовать таблицу firma.

Запишем запрос на языке DML:

Select uradr,telefon

From firma f

WHERE f.firma='com40'

Результат выполнения запроса представлен на рисунках 2.8-2.9.

Рисунок 2.8

Таблица фирма

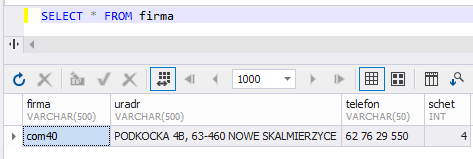
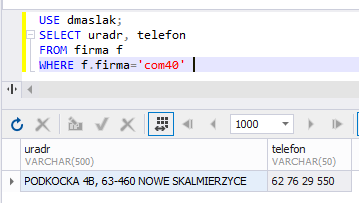


Рисунок 2.9

Результат выполнения запроса



### Задание d.

Перечислить номера складов на которых имеется данный товар данного изготовителя.

Входными данными для нас является название изготовителя и товара, выходными – номера складов.

Запишем запрос на языке DML:

SELECT s.n

FROM sklad s, izgot i, tovar t

WHERE

t.izgot=i.n

AND t.sklad=s.n

AND t.name='Кресла'

AND i.name='ЧПТУП "ИНФОЦЕНТР"'

Результат выполнения запроса представлен на рисунках 2.10-2.14.

Рисунок 2.10

Таблица складов

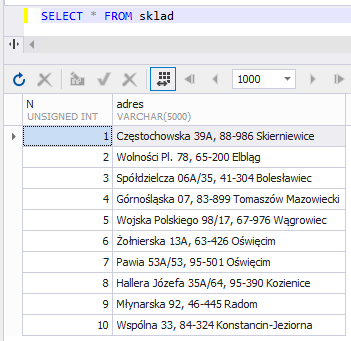


Рисунок 2.11

Таблица изготовителей

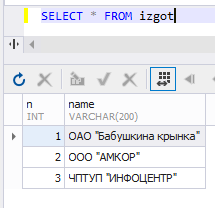


Рисунок 2.12

Таблица складов

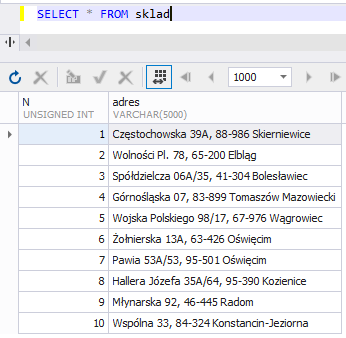
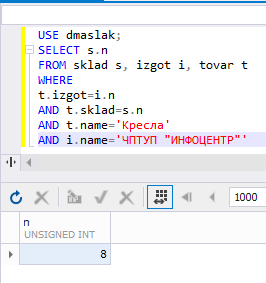


Рисунок 2.14

Результат выполнения запроса



### Задание e.

В каком банке имеет счет клиент с данным номером?

В задании необходимо найти записи, у которых совпадает Номер клиента в таблице клиентов и по связи его с таблицей СЧЕТ выбрать из неё название банка. Необходимо задействовать 2.

Запишем запрос на языке DML:

SELECT s.bank

FROM klient k, schet s

WHERE

k.schet=s.n

AND k.n=5

Результат выполнения запроса представлен на рисунках 2.15-2.17.

Рисунок 2.15

Таблица банковских счетов

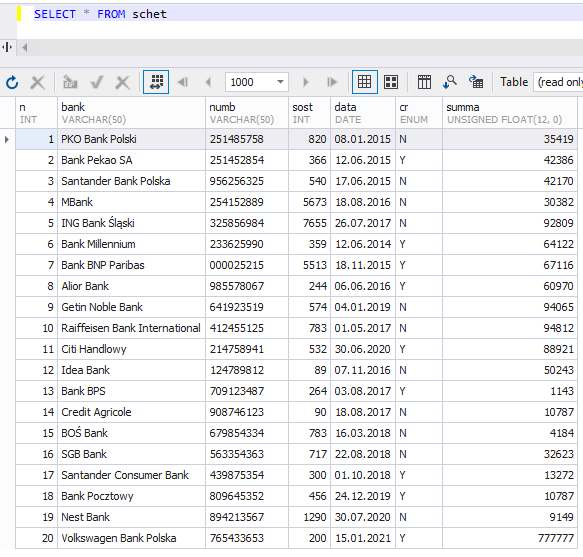


Рисунок 2.16

Таблица клиентов

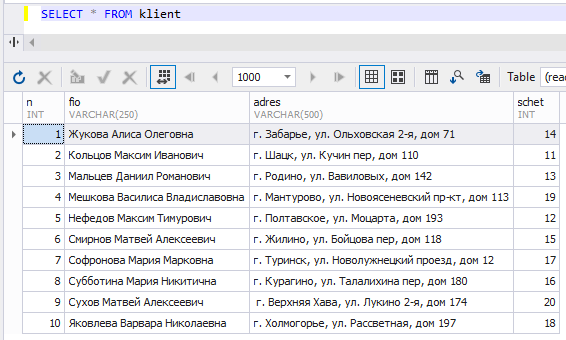
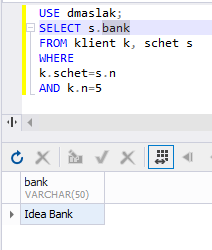


Рисунок 2.17

Результат выполнения запроса

****

## 2.3 Создание представлений

Представление (VIEW) - объект данных, который не содержит никаких данных его владельца. Это - тип таблицы, чье содержание выбирается из других таблиц с помощью выполнения запроса. Поскольку значения в этих таблицах меняются, то автоматически, их значения могут быть показаны представлением. Представления — это таблицы чье содержание выбирается или получается из других таблиц. Они работают в запросах и операторах DML точно также как и основные таблицы, но не содержат никаких собственных данных. Представления - подобны окнам, через которые можно просматривать информацию (как она есть, или в другой форме), которая фактически хранится в базовой таблице. Представление — это фактически запрос, который выполняется всякий раз, когда представление становится темой команды. Вывод запроса при этом в каждый момент становится содержанием представления.

### Задание a

CREATE

DEFINER = 'root'@'localhost'

VIEW dmaslak.viewa

AS

SELECT

i.name

FROM izgot i

INNER JOIN tovar t

ON i.n = t.izgot

WHERE t.name = 'Лампы'

AND t.cena < 30;

### Задание b

CREATE

DEFINER = 'root'@'localhost'

VIEW dmaslak.viewb

AS

SELECT

p.nazv AS naz

FROM postav p,

schet s

WHERE p.schet = s.n

AND s.bank = 'Bank BNP Paribas'

AND s.data = '2015-11-18'

AND s.cr = 'Y'

UNION

SELECT

p.fio AS naz

FROM klient p,

schet s

WHERE p.schet = s.n

AND s.bank = 'Bank BNP Paribas'

AND s.data = '2015-11-18'

AND s.cr = 'Y'

UNION

SELECT

p.firma AS naz

FROM firma p,

schet s

WHERE p.schet = s.n

AND s.bank = 'Bank BNP Paribas'

AND s.data = '2015-11-18'

AND s.cr = 'Y';

### Задание c

CREATE

DEFINER = 'root'@'localhost'

VIEW dmaslak.viewc

AS

SELECT

uradr,

telefon

FROM firma

WHERE firma.firma = 'com40';

### Задание d

CREATE

DEFINER = 'root'@'localhost'

VIEW dmaslak.viewd

AS

SELECT

s.n

FROM sklad s,

izgot i,

tovar t

WHERE t.izgot = i.n

AND t.sklad = s.n

AND t.name = 'Кресла'

AND i.name = 'ЧПТУП "ИНФОЦЕНТР"';

### Задание e

CREATE

DEFINER = 'root'@'localhost'

VIEW dmaslak.viewe

AS

SELECT

s.bank

FROM klient k,

schet s

WHERE k.schet = s.n

AND k.n = 5;

## 2.4. Создание индексов

Анализ готовых запросов показывает, что для повышения эффективности работы с данными необходимо создать индексы для наиболее употребительных полей таблиц. Индексы для наиболее часто употребительных запросов, таких как наименования поставщиков, клиентов, идентификаторы и суррогатные ключи созданы в коде создания таблиц.

# ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта были получены навыки работы с системой управления базы данных, проектирования и непосредственно разработки базы данных. БД позволяют собирать, хранить, обновлять и выводить информацию в понятной пользователю форме.

В результате курсового проектирования была разработана и создана база данных "Фирма VII", которая значительно упрощает работу с данными и во многом экономит время сотрудников. Созданная база данных позволяет вести учет товаров на складах, выданных накладных, управление персоналом фирмы, контроль заказов, в нее могут быть введены дополнительные средства формирования данных.

В процессе проектирования и разработки данного программного модуля были задействованы язык СУБД MySQL Server.

После проектирования c помощью SQL Server была создана полноценная база данных для осуществления выполнения заказов, поступающих от клиентов.

Кроме того, было проведено тестирование базы данных, которое подтвердило правильность, работоспособность и эффективность базы данных.

Проект разработан в полном объеме и полностью соответствует поставленной цели и готов к использованию.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Грофф, Дж. Р.SQL : Полное руководство / Дж. Р. Грофф, П. Н. Вайнберг ; под ред. В. Р. Гинзбург. – 2-е изд., перераб. и доп. – Киев : BHV: Ирина, 2001. – 816 с
2. Дейт, К. Введение в системы баз данных / К. Дейт. – 8-е изд. – Москва:Санкт-Петербург : Киев : Вильямс, 2017. – 1328 с.
3. Куликов С.С. Реляционные базы данных в примерах : практическое пособие для программистов и тестировщи- ков / С. С. Куликов. — Минск: Четыре четверти, 2020. — 424 с.
4. 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
5. Куликов С.С. Работа с MySQL, MS SQL Server и Oracle в примерах (ссылка для скачивания http://svyatoslav.biz/database\_book/)
6. 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

## Полный экспорт 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 dmaslak

DROP DATABASE IF EXISTS `dmaslak`;

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

USE `dmaslak`;

-- Дамп структуры для таблица dmaslak.firma

DROP TABLE IF EXISTS `firma`;

CREATE TABLE IF NOT EXISTS `firma` (

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

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

`telefon` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci DEFAULT '',

`schet` int NOT NULL,

`id` int NOT NULL AUTO\_INCREMENT,

PRIMARY KEY (`id`),

KEY `FKschetf` (`schet`) USING BTREE,

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

) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='Описание фирмы';

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

DELETE FROM `firma`;

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

INSERT INTO `firma` (`firma`, `uradr`, `telefon`, `schet`, `id`) VALUES

('com40', 'PODKOCKA 4B, 63-460 NOWE SKALMIERZYCE', '62 76 29 550', 4, 0);

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

-- Дамп структуры для таблица dmaslak.izgot

DROP TABLE IF EXISTS `izgot`;

CREATE TABLE IF NOT EXISTS `izgot` (

`n` int NOT NULL AUTO\_INCREMENT,

`name` varchar(200) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

PRIMARY KEY (`n`),

UNIQUE KEY `name` (`name`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

-- Дамп данных таблицы dmaslak.izgot: ~3 rows (приблизительно)

DELETE FROM `izgot`;

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

INSERT INTO `izgot` (`n`, `name`) VALUES

(1, 'ОАО "Бабушкина крынка"'),

(2, 'ООО "АМКОР"'),

(3, 'ЧПТУП "ИНФОЦЕНТР"');

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

-- Дамп структуры для таблица dmaslak.kadry

DROP TABLE IF EXISTS `kadry`;

CREATE TABLE IF NOT EXISTS `kadry` (

`N` int NOT NULL AUTO\_INCREMENT,

`fio` varchar(250) NOT NULL,

`dolg` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

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

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

`data` date NOT NULL,

`firma` int NOT NULL,

PRIMARY KEY (`N`) USING BTREE,

UNIQUE KEY `i2` (`fio`),

UNIQUE KEY `idkadr` (`N`) USING BTREE,

UNIQUE KEY `fio` (`fio`),

KEY `FK\_kadry\_firma` (`firma`),

CONSTRAINT `FK\_kadry\_firma` FOREIGN KEY (`firma`) REFERENCES `firma` (`id`)

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

-- Дамп данных таблицы dmaslak.kadry: ~8 rows (приблизительно)

DELETE FROM `kadry`;

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

INSERT INTO `kadry` (`N`, `fio`, `dolg`, `obraz`, `salary`, `data`, `firma`) VALUES

(1, 'Анатолий Николаевич Жерносенко', 'Генеральный директор (приемная)', 'Базовое', 1500, '2014-08-21', 0),

(2, 'Юрий Владимирович Главатский', 'Главный инженер', 'Высшее', 1450, '2014-08-21', 0),

(3, 'Андрей Николаевич Рожков', 'Зам. генерального директора по кадровой, идеологической работе и экономической безопасности', 'Высшее', 1350, '2014-08-22', 0),

(4, 'Сергей Борисович Юзвяк', 'Зам. генерального директора по материально-техническому снабжению и транспорту', 'Высшее', 1355, '2014-08-21', 0),

(5, 'Андрей Владимирович Пескун', 'Зам. генерального директора – начальник управления маркетинга, сбыта и фирменного обслуживания', 'Высшее', 1400, '2014-08-21', 0),

(6, 'Александр Николаевич Таланцев', 'Главный бухгалтер', 'Высшее', 1000, '2014-08-24', 0),

(7, 'Ольга Михайловна Старченко', 'Начальник ОТК', 'Среднеспециальное', 1000, '2014-08-25', 0),

(8, 'Елена Адамовна Рапеева', 'Главный технолог', 'Высшее', 1100, '2014-08-21', 0);

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

-- Дамп структуры для таблица dmaslak.klient

DROP TABLE IF EXISTS `klient`;

CREATE TABLE IF NOT EXISTS `klient` (

`n` int NOT NULL AUTO\_INCREMENT,

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

`adres` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci DEFAULT NULL,

`schet` int DEFAULT NULL,

PRIMARY KEY (`n`) USING BTREE,

UNIQUE KEY `i3` (`fio`),

UNIQUE KEY `idclient` (`n`) USING BTREE,

UNIQUE KEY `fio` (`fio`),

KEY `FKschet` (`schet`) USING BTREE,

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

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

-- Дамп данных таблицы dmaslak.klient: ~7 rows (приблизительно)

DELETE FROM `klient`;

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

INSERT INTO `klient` (`n`, `fio`, `adres`, `schet`) VALUES

(1, 'Жукова Алиса Олеговна', 'г. Забарье, ул. Ольховская 2-я, дом 71', 14),

(2, 'Кольцов Максим Иванович', 'г. Шацк, ул. Кучин пер, дом 110', 11),

(3, 'Мальцев Даниил Романович', 'г. Родино, ул. Вавиловых, дом 142', 13),

(4, 'Мешкова Василиса Владиславовна', 'г. Мантурово, ул. Новоясеневский пр-кт, дом 113', 19),

(5, 'Нефедов Максим Тимурович', 'г. Полтавское, ул. Моцарта, дом 193', 12),

(6, 'Смирнов Матвей Алексеевич', 'г. Жилино, ул. Бойцова пер, дом 118', 15),

(7, 'Софронова Мария Марковна', 'г. Туринск, ул. Новолужнецкий проезд, дом 12', 17),

(8, 'Субботина Мария Никитична', 'г. Курагино, ул. Талалихина пер, дом 180', 16),

(9, 'Сухов Матвей Алексеевич', ' г. Верхняя Хава, ул. Лукино 2-я, дом 174', 20),

(10, 'Яковлева Варвара Николаевна', 'г. Холмогорье, ул. Рассветная, дом 197', 18);

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

-- Дамп структуры для таблица dmaslak.nakladnaya

DROP TABLE IF EXISTS `nakladnaya`;

CREATE TABLE IF NOT EXISTS `nakladnaya` (

`n` int NOT NULL AUTO\_INCREMENT,

`ntovar` int NOT NULL,

`kolvo` int unsigned NOT NULL,

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

`nklient` int NOT NULL,

PRIMARY KEY (`n`) USING BTREE,

UNIQUE KEY `idnaklad` (`n`) USING BTREE,

KEY `FKtovarn` (`ntovar`) USING BTREE,

KEY `FK\_nakladnaya\_klient` (`nklient`) USING BTREE,

CONSTRAINT `FK\_nakladnaya\_klient` FOREIGN KEY (`nklient`) REFERENCES `klient` (`n`),

CONSTRAINT `FKtovarn` FOREIGN KEY (`ntovar`) REFERENCES `tovar` (`n`)

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

-- Дамп данных таблицы dmaslak.nakladnaya: ~5 rows (приблизительно)

DELETE FROM `nakladnaya`;

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

INSERT INTO `nakladnaya` (`n`, `ntovar`, `kolvo`, `summa`, `nklient`) 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 `nakladnaya` ENABLE KEYS \*/;

-- Дамп структуры для таблица dmaslak.postav

DROP TABLE IF EXISTS `postav`;

CREATE TABLE IF NOT EXISTS `postav` (

`n` int NOT NULL AUTO\_INCREMENT,

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

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

`schet` int DEFAULT NULL,

PRIMARY KEY (`n`) USING BTREE,

UNIQUE KEY `idpostav` (`n`) USING BTREE,

UNIQUE KEY `i4` (`nazv`) USING BTREE,

UNIQUE KEY `nazv` (`nazv`),

KEY `FKschetp` (`schet`) USING BTREE,

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

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

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

DELETE FROM `postav`;

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

INSERT INTO `postav` (`n`, `nazv`, `uradr`, `schet`) VALUES

(1, 'Siedziba firmy Personio', 'Strażacka 41A/64, 84-331 Piaseczno', 4),

(2, 'Dom w Konstancinie', 'Wczasowa 07A, 35-983 Pawłowice', 3),

(3, 'L\'Oreal House Warsaw', 'Równa 31, 65-189 Płock', 5),

(4, 'Microsoft Polska', 'Miła 09/65, 70-122 Koszalin', 1),

(5, 'Hotel Provita', 'Zbożowa 33, 06-300 Ełk', 2),

(6, 'Hotel Marriott', 'Bielska 40/80, 37-406 Żywiec', 7),

(7, 'NH Hotel Bloom', 'Kołobrzeska 64/48, 14-495 Tomaszów Mazowiecki', 8),

(8, 'Penthaus', 'Sienkiewicza Henryka 06/92, 40-829', 10),

(9, 'Novotel', 'Puławska 38/94, 29-547 Rybnik', 6),

(10, 'Sofitel', 'Srebrna 97/23, 50-973 Legnica', 9);

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

-- Дамп структуры для таблица dmaslak.schet

DROP TABLE IF EXISTS `schet`;

CREATE TABLE IF NOT EXISTS `schet` (

`n` int NOT NULL AUTO\_INCREMENT,

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

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

`sost` int NOT NULL DEFAULT '0',

`data` date NOT NULL,

`cr` enum('Y','N') CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

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

PRIMARY KEY (`n`) USING BTREE,

UNIQUE KEY `idschet` (`n`) USING BTREE

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

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

DELETE FROM `schet`;

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

INSERT INTO `schet` (`n`, `bank`, `numb`, `sost`, `data`, `cr`, `summa`) VALUES

(1, 'PKO Bank Polski', '251485758', 820, '2015-01-08', 'N', 35419),

(2, 'Bank Pekao SA', '251452854', 366, '2015-06-12', 'Y', 42386),

(3, 'Santander Bank Polska', '956256325', 540, '2015-06-17', 'N', 42170),

(4, 'MBank', '254152889', 5673, '2016-08-18', 'N', 30382),

(5, 'ING Bank Śląski', '325856984', 7655, '2017-07-26', 'N', 92809),

(6, 'Bank Millennium', '233625990', 359, '2014-06-12', 'Y', 64122),

(7, 'Bank BNP Paribas', '000025215', 5513, '2015-11-18', 'Y', 67116),

(8, 'Alior Bank', '985578067', 244, '2016-06-06', 'Y', 60970),

(9, 'Getin Noble Bank', '641923519', 574, '2019-01-04', 'N', 94065),

(10, 'Raiffeisen Bank International', '412455125', 783, '2017-05-01', 'N', 94812),

(11, 'Citi Handlowy', '214758941', 532, '2020-06-30', 'Y', 88921),

(12, 'Idea Bank', '124789812', 89, '2016-11-07', 'N', 50243),

(13, 'Bank BPS', '709123487', 264, '2017-08-03', 'Y', 1143),

(14, 'Credit Agricole', '908746123', 90, '2017-08-18', 'N', 10787),

(15, 'BOŚ Bank', '679854334', 783, '2018-03-16', 'N', 4184),

(16, 'SGB Bank', '563354363', 717, '2018-08-22', 'N', 32623),

(17, 'Santander Consumer Bank', '439875354', 300, '2018-10-01', 'Y', 13272),

(18, 'Bank Pocztowy', '809645352', 456, '2019-12-24', 'Y', 10787),

(19, 'Nest Bank', '894213567', 1290, '2020-07-30', 'N', 9149),

(20, 'Volkswagen Bank Polska', '765433653', 200, '2021-01-15', 'Y', 777777);

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

-- Дамп структуры для таблица dmaslak.sklad

DROP TABLE IF EXISTS `sklad`;

CREATE TABLE IF NOT EXISTS `sklad` (

`N` int unsigned NOT NULL AUTO\_INCREMENT,

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

PRIMARY KEY (`N`) USING BTREE

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

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

DELETE FROM `sklad`;

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

INSERT INTO `sklad` (`N`, `adres`) VALUES

(1, 'Częstochowska 39A, 88-986 Skierniewice'),

(2, 'Wolności Pl. 78, 65-200 Elbląg'),

(3, 'Spółdzielcza 06A/35, 41-304 Bolesławiec'),

(4, 'Górnośląska 07, 83-899 Tomaszów Mazowiecki'),

(5, 'Wojska Polskiego 98/17, 67-976 Wągrowiec'),

(6, 'Żołnierska 13A, 63-426 Oświęcim'),

(7, 'Pawia 53A/53, 95-501 Oświęcim'),

(8, 'Hallera Józefa 35A/64, 95-390 Kozienice'),

(9, 'Młynarska 92, 46-445 Radom'),

(10, 'Wspólna 33, 84-324 Konstancin-Jeziorna');

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

-- Дамп структуры для таблица dmaslak.tovar

DROP TABLE IF EXISTS `tovar`;

CREATE TABLE IF NOT EXISTS `tovar` (

`n` int NOT NULL AUTO\_INCREMENT,

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

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

`cena` int unsigned NOT NULL,

`postav` int NOT NULL,

`sklad` int unsigned NOT NULL,

`Kolvo` int unsigned NOT NULL,

`sort` int unsigned NOT NULL,

`izgot` int DEFAULT NULL,

PRIMARY KEY (`n`) USING BTREE,

UNIQUE KEY `idtovar` (`n`) USING BTREE,

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

KEY `FKpostavt` (`postav`) USING BTREE,

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

KEY `FK\_tovar\_izgot` (`izgot`),

FULLTEXT KEY `name` (`name`),

CONSTRAINT `FK\_tovar\_izgot` FOREIGN KEY (`izgot`) REFERENCES `izgot` (`n`),

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

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

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

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

DELETE FROM `tovar`;

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

INSERT INTO `tovar` (`n`, `name`, `ed`, `cena`, `postav`, `sklad`, `Kolvo`, `sort`, `izgot`) VALUES

(1, 'Диваны', 'шт', 1500, 1, 9, 989, 1, 1),

(2, 'Диваны раскладные', 'шт', 3000, 2, 7, 356, 0, 2),

(3, 'Пуфы', 'шт', 250, 1, 10, 916, 2, 3),

(4, 'Кресла', 'шт', 500, 2, 8, 800, 1, 3),

(5, 'Подножки', 'шт', 100, 1, 4, 80, 3, 2),

(6, 'Стулья', 'шт', 50, 4, 3, 713, 2, 1),

(7, 'Книжный шкаф', 'шт', 350, 6, 4, 659, 7, 2),

(8, 'Лампы', 'шт', 20, 7, 9, 322, 2, 3),

(9, 'Кровать', 'шт', 2350, 4, 7, 740, 1, 2),

(10, 'Столы', 'шт', 450, 6, 3, 999, 3, 1);

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

-- Дамп структуры для таблица dmaslak.transport

DROP TABLE IF EXISTS `transport`;

CREATE TABLE IF NOT EXISTS `transport` (

`n` 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(150) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

`data` date NOT NULL,

`sklad` int unsigned NOT NULL,

PRIMARY KEY (`n`) USING BTREE,

UNIQUE KEY `i5` (`nomer`),

KEY `FK\_transport\_sklad` (`sklad`) USING BTREE,

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

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

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

DELETE FROM `transport`;

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

INSERT INTO `transport` (`n`, `nomer`, `marka`, `gruz`, `god`, `fio`, `data`, `sklad`) VALUES

(2, 'Х 410 Х А', 'MD-11', 340000, 2010, 'Com-Avia', '2014-09-24', 3),

(3, 'В 189 У К', 'SCANIA', 70000, 2011, 'Com-Avto', '2014-10-10', 4),

(4, 'С 159 Н У', 'MAЗ', 24000, 2014, 'Com-Avto', '2014-11-27', 1),

(5, 'Р 115 С О', 'RENAULT', 3400, 2012, 'Com-Avto', '2014-08-28', 3),

(6, 'В 674 К Т', 'FOTON', 50000, 2012, 'Com-Avto', '2014-09-24', 5),

(7, 'Х 360 О Е', 'Ан-225', 520000, 2013, 'Com-Avia', '2015-01-07', 10);

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

-- Дамп структуры для представление dmaslak.viewa

DROP VIEW IF EXISTS `viewa`;

-- Создание временной таблицы для обработки ошибок зависимостей представлений

CREATE TABLE `viewa` (

`name` VARCHAR(200) NOT NULL COLLATE 'utf8mb4\_0900\_ai\_ci'

) ENGINE=MyISAM;

-- Дамп структуры для представление dmaslak.viewb

DROP VIEW IF EXISTS `viewb`;

-- Создание временной таблицы для обработки ошибок зависимостей представлений

CREATE TABLE `viewb` (

`naz` VARCHAR(500) NOT NULL COLLATE 'utf8mb4\_0900\_ai\_ci'

) ENGINE=MyISAM;

-- Дамп структуры для представление dmaslak.viewc

DROP VIEW IF EXISTS `viewc`;

-- Создание временной таблицы для обработки ошибок зависимостей представлений

CREATE TABLE `viewc` (

`uradr` VARCHAR(500) NOT NULL COLLATE 'utf8mb4\_0900\_ai\_ci',

`telefon` VARCHAR(50) NULL COLLATE 'utf8mb4\_0900\_ai\_ci'

) ENGINE=MyISAM;

-- Дамп структуры для представление dmaslak.viewd

DROP VIEW IF EXISTS `viewd`;

-- Создание временной таблицы для обработки ошибок зависимостей представлений

CREATE TABLE `viewd` (

`n` INT(10) UNSIGNED NOT NULL

) ENGINE=MyISAM;

-- Дамп структуры для представление dmaslak.viewe

DROP VIEW IF EXISTS `viewe`;

-- Создание временной таблицы для обработки ошибок зависимостей представлений

CREATE TABLE `viewe` (

`bank` VARCHAR(50) NOT NULL COLLATE 'utf8mb4\_0900\_ai\_ci'

) ENGINE=MyISAM;

-- Дамп структуры для таблица dmaslak.zakaz

DROP TABLE IF EXISTS `zakaz`;

CREATE TABLE IF NOT EXISTS `zakaz` (

`n` int NOT NULL AUTO\_INCREMENT,

`data` date NOT NULL,

`client` int NOT NULL,

`tovar` int NOT NULL,

`kolvo` int NOT NULL DEFAULT '0',

`opl` enum('Y','N') CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

PRIMARY KEY (`n`) USING BTREE,

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

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

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

CONSTRAINT `FKclientz` FOREIGN KEY (`client`) REFERENCES `klient` (`n`),

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

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

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

DELETE FROM `zakaz`;

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

INSERT INTO `zakaz` (`n`, `data`, `client`, `tovar`, `kolvo`, `opl`) VALUES

(1, '2021-03-24', 6, 8, 3, ''),

(2, '2021-05-05', 9, 10, 1, ''),

(3, '2021-02-28', 1, 10, 3, ''),

(4, '2021-01-13', 4, 4, 1, '');

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

-- Дамп структуры для представление dmaslak.viewa

DROP VIEW IF EXISTS `viewa`;

-- Удаление временной таблицы и создание окончательной структуры представления

DROP TABLE IF EXISTS `viewa`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY DEFINER VIEW `viewa` AS select `i`.`name` AS `name` from (`izgot` `i` join `tovar` `t` on((`i`.`n` = `t`.`izgot`))) where ((`t`.`name` = 'Лампы') and (`t`.`cena` < 30));

-- Дамп структуры для представление dmaslak.viewb

DROP VIEW IF EXISTS `viewb`;

-- Удаление временной таблицы и создание окончательной структуры представления

DROP TABLE IF EXISTS `viewb`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY DEFINER VIEW `viewb` AS select `p`.`nazv` AS `naz` from (`postav` `p` join `schet` `s`) where ((`p`.`schet` = `s`.`n`) and (`s`.`bank` = 'Bank BNP Paribas') and (`s`.`data` = '2015-11-18') and (`s`.`cr` = 'Y')) union select `p`.`fio` AS `naz` from (`klient` `p` join `schet` `s`) where ((`p`.`schet` = `s`.`n`) and (`s`.`bank` = 'Bank BNP Paribas') and (`s`.`data` = '2015-11-18') and (`s`.`cr` = 'Y')) union select `p`.`firma` AS `naz` from (`firma` `p` join `schet` `s`) where ((`p`.`schet` = `s`.`n`) and (`s`.`bank` = 'Bank BNP Paribas') and (`s`.`data` = '2015-11-18') and (`s`.`cr` = 'Y'));

-- Дамп структуры для представление dmaslak.viewc

DROP VIEW IF EXISTS `viewc`;

-- Удаление временной таблицы и создание окончательной структуры представления

DROP TABLE IF EXISTS `viewc`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY DEFINER VIEW `viewc` AS select `firma`.`uradr` AS `uradr`,`firma`.`telefon` AS `telefon` from `firma` where (`firma`.`firma` = 'com40');

-- Дамп структуры для представление dmaslak.viewd

DROP VIEW IF EXISTS `viewd`;

-- Удаление временной таблицы и создание окончательной структуры представления

DROP TABLE IF EXISTS `viewd`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY DEFINER VIEW `viewd` AS select `s`.`N` AS `n` from ((`sklad` `s` join `izgot` `i`) join `tovar` `t`) where ((`t`.`izgot` = `i`.`n`) and (`t`.`sklad` = `s`.`N`) and (`t`.`name` = 'Кресла') and (`i`.`name` = 'ЧПТУП "ИНФОЦЕНТР"'));

-- Дамп структуры для представление dmaslak.viewe

DROP VIEW IF EXISTS `viewe`;

-- Удаление временной таблицы и создание окончательной структуры представления

DROP TABLE IF EXISTS `viewe`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY DEFINER VIEW `viewe` AS select `s`.`bank` AS `bank` from (`klient` `k` join `schet` `s`) where ((`k`.`schet` = `s`.`n`) and (`k`.`n` = 5));

/\*!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) \*/;