Министерство образования Республики Беларусь

Учреждение образования

«Могилевский государственный университет имени А.А.Кулешова»

«ИНФОРМАЦИОННАЯ СИСТЕМА «фИРМА 12»»

Курсовая работа

по дисциплине «Базы данных»

cтудента 1 курса специальности 1-40 01 01

«Программное обеспечение информационных технологий»

заочной формы получения образования, группы 5242201

Алина Андреевны Парахневич

Научный руководитель

доцент кафедры программного обеспечения

информационных технологий,

кандидат физико-математических наук

Иван Николаевич Сидоренко

Могилев 2021

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 3](#_Toc73909388)

[ВВЕДЕНИЕ 4](#_Toc73909389)

[1 ПОСТАНОВКА ЗАДАЧИ 5](#_Toc73909390)

[Анализ предметной области 5](#_Toc73909391)

[Постановка задачи 5](#_Toc73909392)

[2 ТЕХНИЧЕСКИЙ ПРОЕКТ ИНФОРМАЦИОННОЙ СИСТЕМЫ 6](#_Toc73909393)

[2.1 Модель данных 6](#_Toc73909394)

[2.2 Проектирование БД методом нормальных форм 7](#_Toc73909395)

[2.3 Нормализация таблиц в базе дынных 8](#_Toc73909396)

[2.4 ER-диаграмма в нотации Мартина 13](#_Toc73909397)

[2.5 Создание представлений и готовых запросов 15](#_Toc73909398)

[Задание а. 15](#_Toc73909399)

[Задание b 16](#_Toc73909400)

[Задание c. 17](#_Toc73909401)

[Задание d. 18](#_Toc73909402)

[Задание e. 19](#_Toc73909403)

[ЗАКЛЮЧЕНИЕ 21](#_Toc73909404)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 22](#_Toc73909405)

# ВВЕДЕНИЕ

Тема курсовой работы: Информационная система «Фирма 12»

В качестве предметной области проектирования БД выбрана торговая фирма, которая осуществляет выполнение заказов, поступающих от клиентов.

Цель курсового проектирования – применить на практике знания, полученные в процессе изучения дисциплины "Базы данных», получить теоретические знания и практические навыки реализации баз данных. Осуществить анализ предметной области. Освоить концептуальное проектирование и научиться определять сущности и атрибуты БД. Научиться разрабатывать инфологическую модель БД в виде ER-диаграмм. Получить теоретические знания и практические навыки при физическом проектировании баз данных (БД). Научиться создавать даталогическую модель БД.

В качестве исходных данных предоставлены объекты и атрибуты описывающие торговую фирму, клиентов, поставщиков и т.д.MySQL — это система управления реляционными базами данных (СУРБД). Где «реляционные» значит, что данные хранятся в виде таблиц.   
 SQL — это язык структурированных запросов, который позволяет взаимодействовать между вами и базой данных. Есть множество других СУРБД, которые также используют язык SQL.

В целом SQL — один самых популярных языков, использующихся в базах данных. Так как с его помощью можно все манипуляции с данными проходят эффективно. Это важно для скорости работы и стабильности базы. Многие системы, в том числе и MySQL, работают по принципу «клиент-сервер». То есть коммуникация происходит по SQL-запросу.

# ГЛАВА 1 ПОСТАНОВКА ЗАДАЧИ

## Анализ предметной области

Предметная область информационной системы – это материальная система или система, характеризующая элементы материального мира, информация которая храниться и обрабатывается. Предметная область рассматривается, как некоторая совокупность реальных объектов и связей между ними.

Предметная область данной курсовой работы заключается в создании информационной системы «Фирма 12». Преимущества данной системы заключается в просмотре, хранении и добавлении данных.

Проведение разбора предметной области в интересах последующего проектирования базы данных является задачей, формирующей целостный взгляд на сведения, которые в предметной области обрабатываются, учитывая не только их структуры, но и правила хранения и обработки, что отражается в выделяемых функциях и задачах.

Прежде чем начать работу с информационной системой, были спроектированы нужные база дынных, составлен перечень нужных данных для работы.

## Постановка задачи

В данной курсовой работе требуется создать информационную систему «Фирма 12», которая обеспечивает формирование и вывод информации. База данных должна обеспечивать формирование данных сформированной с помощью Microsoft SQL сведенья о товарах, заказах, клиентах. Так же необходимо создать запросы в базе данных.

В отличии от работы проводимый вручную, автоматизированная система дает ряд преимуществ:

* сокращение ручного труда (сведение к минимуму);
* экономия временного ресурса;
* быстрый поиск информации.

Информационная система должна осуществлять ввод полной информации, ее редактирование, удаление и сохранение. Входные данные будут вводиться пользователем с клавиатуры, результаты работы сохраняться в базу данных.

# ГЛАВА2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ СИСТЕМЫ

## 2.1 Модель данных

Хранимые в базе данные имеют определенную логическую структуру, то есть представлены некоторой моделью, поддерживаемой СУБД. К числу важнейших относятся следующие модели данных:

−иерархическая;

−сетевая;

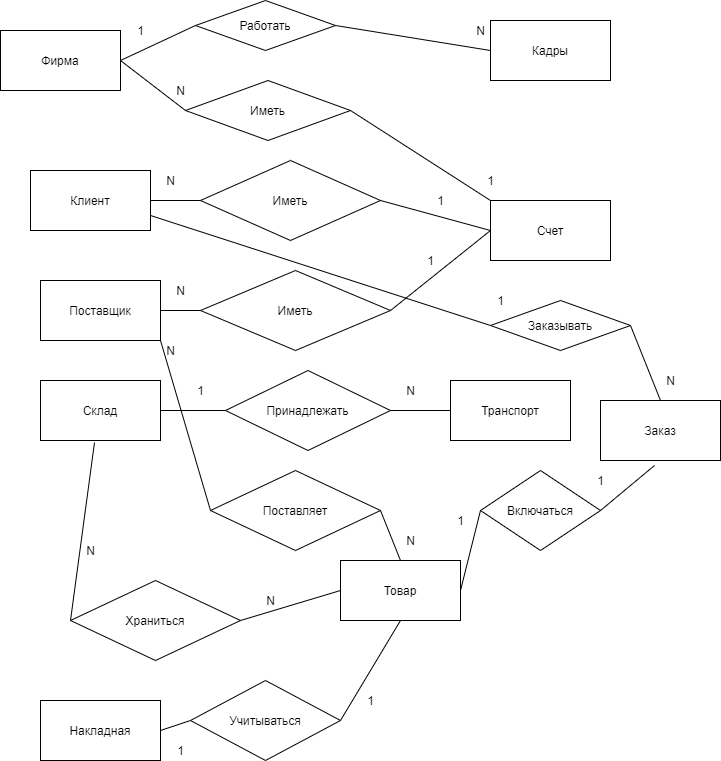
−реляционная;

−объектно-ориентированная.

Исходя из анализа предметной области, набора требований и возможностей реляционной модели данных разработана модель реляционной базы данных данной информационной системы. Её структура представлена на рисунке 2.1.1.

Рисунок 2.1.1

Модель данных «Информационной системы «Фирма 12»»



На диаграмме «Модель данных» представлены следующие отношения:

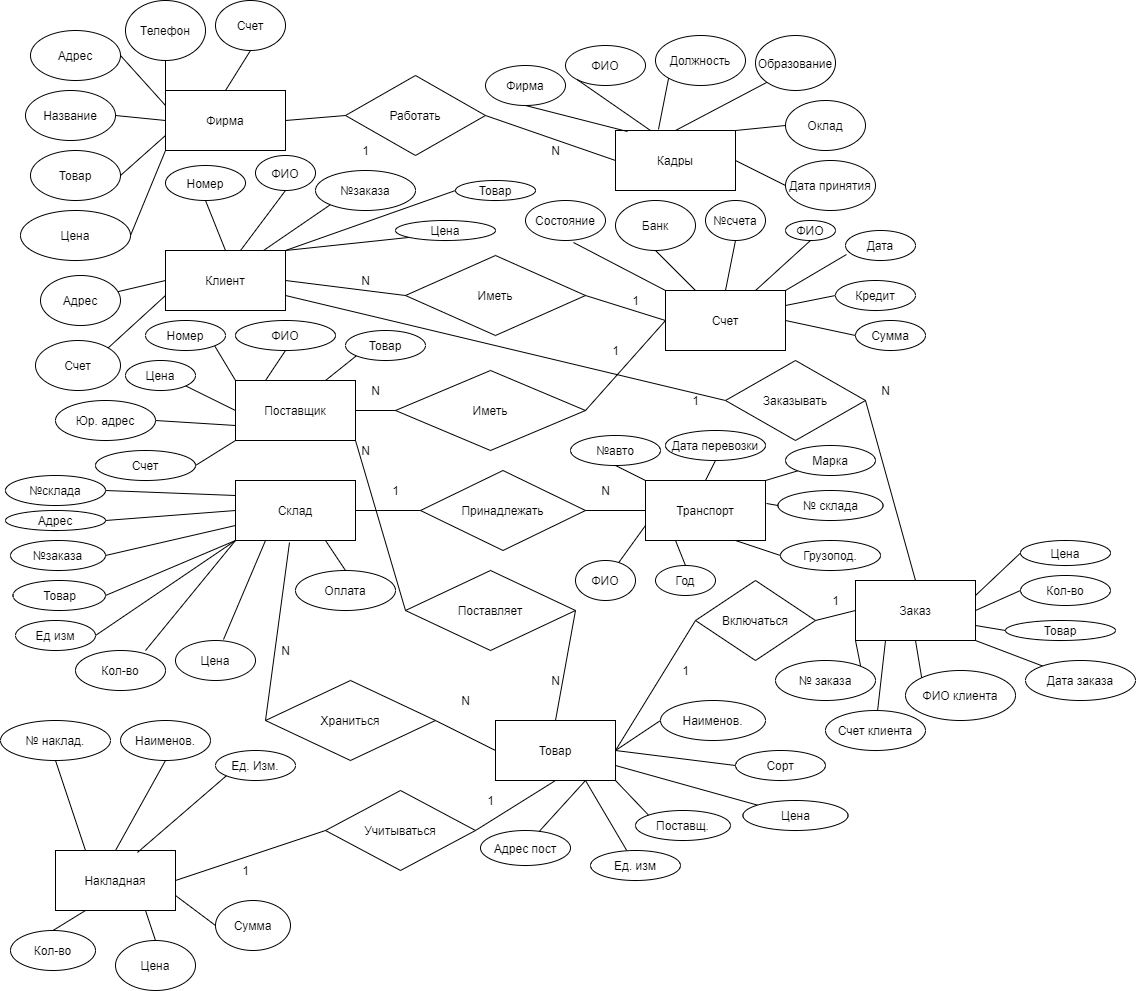
* + - 1. Фирма (название фирмы, юр. адрес, телефон фирмы, №счета в банке, наименование товара, цена товара);
      2. Склад (№склада, адрес склада, №заказа, наименование товара, единицы измерения, кол-во товара, цена товара, отметки об оплате);
      3. Кадры (Название фирмы, Ф.И.О. сотрудника, должность 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)

Рассмотрим расширенную диаграмму с атрибутами отношений (рисунок 2.1.2)

Рисунок 2.1.1

Расширенная модель данных «Информационной системы «Фирма 12»»



## 2.2 Проектирование БД методом нормальных форм

Проектирование базы данных является одним из этапов жизненного цикла информационной системы. Основной задачей, решаемой в процессе проектирования базы данных, является задача нормализацией ее отношений.

Рассматриваемый ниже метод нормальных форм является классическим методом проектирования реляционных базы данных. Этот метод основан на фундаментальном в теории реляционных баз данных понятия зависимости между атрибутами отношений.

Процесс проектирования базы данных с использованием метода нормальных форм является, итерационным и заключается в последовательном переводе отношений из первой нормальной формы в нормальные формы более высокого порядка по определенным правилам.

Каждая следующая нормальная форма ограничивает определенный тип функциональных зависимостей, устраняет соответствующие аномалии при выполнении операций над отношениями базы данных и сохраняет свойства предшествующих нормальных форм.

Выделяют следующую последовательность нормальных форм:

− первая нормальная форма (1НФ);

− вторая нормальная форма(2НФ);

− третья нормальная форма(3НФ);

− усиленная третья нормальная форма, или форма Бойса-Кодда;

− четвертая нормальная форма(4НФ);

− пятая нормальная форма(5НФ).

Рассмотрим данные нам отношения и связи между ними.

Потенциальным ключом отношения ФИРМА является атрибут Название, однако название может быть достаточно длинным, поэтому мы введём суррогатный первичный ключ ID числового типа (Таблица 2.3.1).

Таблица 2.3.1 Схема отношения ФИРМА(firma)

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |
| № счета банка | nschet | C(500) | Обязательное поле |
| Наименование товара | tovar | C(500) | Обязательное поле |
| Цена товара | cena | N | Обязательное поле |
| ID | id | N | Первичный ключ суррогатный |

Потенциальным ключом отношения СКЛАД является атрибут Номер, поэтому мы выбираем его в качестве первичного ключа ( Таблица 2.3.2).

Таблица 2.3.2 Схема отношения СКЛАД(sklad)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| №склада | nsklad | N | Первичный ключ |
| Адрес склада | adress | C(500) | Обязательное поле |
| №заказа | nzakaz | N | Обязательное поле |
| № счета банка | nschet | 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, но это длинная текстовая строка, поэтому мы введем суррогатный ключ ID и выбираем его в качестве первичного ключа (Таблица 2.3.3).

Таблица 2.3.3 Схема отношения КАДРЫ(kadry)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Название фирмы | nazvan | N | Обязательное поле |
| ФИО сотр. | fio | C(500) | Обязательное поле |
| Должность | dolgznost | C(500) | Обязательное поле |
| Образование | obrazovan | C(500) | Обязательное поле |
| Оклад | oklad | N | Обязательное поле |
| Дата принятия | data | D | Обязательное поле |
| Идентификатор | id | N | Первичный ключ суррогатный |

Потенциальным ключом отношения ПОСТАВЩИК является атрибут Номер поставщика, он уникален, поэтому мы выбираем его в качестве первичного ключа (Таблица 2.3.4).

Таблица 2.3.4 Схема отношения ПОСТАВЩИК(postav)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| №поставщика | id | N | Первичный ключ |
| ФИО пост. | fio | C(500) | Обязательное поле |
| Юр. адр. | uradr | C(500) | Обязательное поле |
| №счета | nschet | C(500) | Обязательное поле |
| Наимен. товара | naimentov | 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 2.3.5).

Таблица 2.3.5 Схема отношения ТРАНСПОРТ(transport)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| №склада | idsklad | N | Обязательное поле |
| №авто | nauto | C(500) | Обязательное поле |
| Марка авто | marka | C(500) | Обязательное поле |
| Грузоподъемность | gruzopod | N | Обязательное поле |
| Год выпуска | god | N | Обязательное поле |
| ФИО владельца | fio | 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 (Таблица 2.3.6).

Таблица 2.3.6 Схема отношения НАКЛАДНАЯ(nakladnaya)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| №накладной | id | N | Первичный ключ |
| Наименование товара | naimen | C(500) | Обязательное поле |
| Ед. изм. | edizm | C(500) | 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 (Таблица 2.3.7).

Таблица 2.3.7 Схема отношения ТОВАР(tovar)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Наименование | naimen | C(500) | Обязательное поле |
| Ед. изм. | edizm | C(50) | Обязательное поле |
| Сорт | sort | N | Обязательное поле |
| Цена | cena | N | Обязательное поле |
| Поставщик | postav | C(500) | Обязательное поле |
| Адрес поставщика | adr\_postav | 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

(Таблица 2.3.8).

Таблица 2.3.8 Схема отношения КЛИЕНТ(klient)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| № клиента | id | N | Первичный ключ |
| ФИО клиента | fio | C(500) | Обязательное поле |
| Юр. адр. | uradr | C(500) | Обязательное поле |
| №счета | nschet | C(500) | Обязательное поле |
| № заказа | nzakaz | 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 (Таблица 2.3.9).

Таблица 2.3.9 Схема отношения СЧЕТ(schet)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| № счета | nschet | C(500) | 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 | kredit | N | Обязательное поле |
| Сумма кредита | summa | N | Обязательное поле |
| Идентификатор | id | N | Первичный ключ |

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

(Таблица 2.3.10).

Таблица 2.3.10 Схема отношения ЗАКАЗ(zakaz)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| № заказа | nschet | N | Первичный ключ |
| Дата заказа | data | D | Обязательное поле |
| ФИО клиента | fio | C(500) | Обязательное поле |
| № счета | schet | C(500) | Обязательное поле |
| Наименование товара | naimenov | C(500) | Обязательное поле |
| Количество | kolvo | N | Обязательное поле |
| Цена товара | cena | N | Обязательное поле |

## 2.3 Нормализация таблиц в базе дынных

Процесс проектирования БД с использование метода нормальной формы является итерационным и заключается в последовательном переводе отношения из первой нормальной формы в нормальную форму более высокого порядка по определенным правилам. Каждая следующая нормальная форма ограничивается определенным типом функциональных зависимостей и устранением соответствующих аномалий при выполнении операций над отношениями базы данных, а также сохранении свойств предшествующих нормальных форм.

Рассмотрим приведение таблиц базы данных информационной системы «Фирма 12» к третьей нормальной форме, представленные в таблицах 2.3.11 -2.3.20:

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 2.3.11).

Таблица 2.3.11 Схема отношения СЧЕТ(schet)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Идентификатор | id | N | Первичный ключ |
| № счета | nschet | C(500) | Обязательное поле |
| Наименов. банка | bank | C(500) | Обязательное поле |
| Состояние счета | status | N | Обязательное поле |
| Дата | data | D | Обязательное поле |
| Наличие кредита | kredit | 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 2.3.12).

Таблица 2.3.12 Схема отношения ПОСТАВЩИК(postav)

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

Таблица 2.3.13 Схема отношения КЛИЕНТ(klient)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| № клиента | id | N | Первичный ключ |
| ФИО клиента | fio | C(500) | Обязательное поле |
| Юр. адр. | uradr | C(500) | 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 (Таблица 2.3.14).

Таблица 2.3.14 Схема отношения ТОВАР(tovar)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Идентификатор | id | N | Первичный ключ |
| Наименование | naimen | C(500) | Обязательное поле |
| Ед. изм. | edizm | C(50) | Обязательное поле |
| Сорт | sort | N | Обязательное поле |
| Цена | cena | N | Обязательное поле |
| Идентификатор склад | idsklad | 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 (Таблица 2.3.15).

Таблица 2.3.15 Схема отношения ЗАКАЗ(zakaz)

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

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 2.3.16).

Таблица 2.3.16 Схема отношения НАКЛАДНАЯ(nakladnaya)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| №накладной | id | N | Первичный ключ |
| Идентификатор товара | idtovar | N | Обязательное поле, внешний ключ к ТОВАР |
| Кол-во | kolvo | N | Обязательное поле |
| Сумма | summa | N | Обязательное поле |

Номер 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 2.3.17).

Таблица 2.3.17 Схема отношения СКЛАД(sklad)

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

Атрибуты Наименование товара, Цена товара могут быть получены из отношения ТОВАР, удалим их. В качестве первичного ключа у нас выступает id

(Таблица 2.3.18).

Таблица 2.3.18 Схема отношения ФИРМА(firma)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Идентификатор | id | N | Первичный ключ суррогатный |
| 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 заменим атрибут Наименование фирмы связью с отношением ФИРМА, первичный ключ id (Таблица 2.3.19).

Таблица 2.3.19 Схема отношения КАДРЫ(kadry)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Идентификатор | id | N | Первичный ключ суррогатный |
| Идентификатор фирмы | idfirma | N | Обязательное поле, внешний ключ к ФИРМА |
| ФИО сотр. | fio | C(500) | Обязательное поле |
| Должность | dolgznost | C(500) | Обязательное поле |
| Образование | obrazovan | C(500) | Обязательное поле, перечисляемое |
| Оклад | oklad | N | Обязательное поле |
| Дата принятия | data | D | Обязательное поле |

Первичный ключ отношения ТРАНСПОРТ атрибут id. № склада заменим связью с отношением СКЛАД (Таблица 2.3.20).

Таблица 2.3.20 Схема отношения ТРАНСПОРТ(transport)

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

## 2.4 ER-диаграмма в нотации Мартина

Нотация Мартина является одной из наиболее известных в разработке баз данных, отражающей уровень логического представления базы данных с обозначением некоторых компонентов модели базы данных в графическом виде, облегчая, тем самым, отображение диаграммы в рабочем пространстве.

Для построения логической модели базы данных в этой нотации разработчик исходит из терминов логического представления базы данных, где элементами хранения данных являются сущности, а не объекты, как в концептуальном представлении. По сути, при последовательном движении в процессе разработки базы данных сущности логической модели базы данных будут иметь соответствие с объектами предметной области, но в процессе моделирования и нормализации моделей возможно появление вспомогательных сущностей, которые не имеют непосредственных представителей в предметной области, но объективно необходимы для эффективного представления и обработки данных.

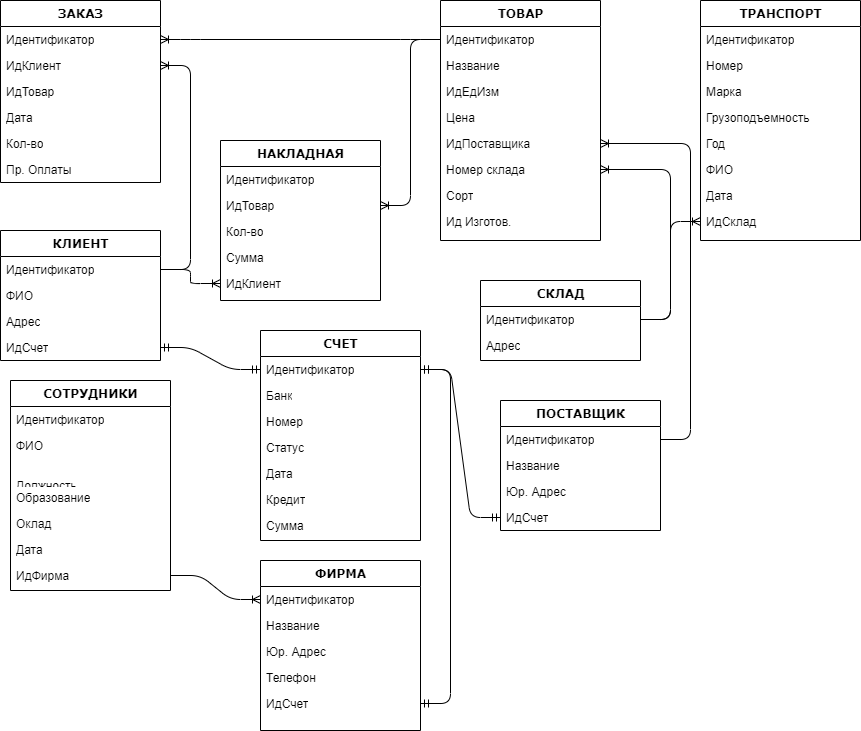
Основу всей модели базы данных в нотации Мартина составляют элементы «Сущность», представляемые прямоугольником с указанием существительного в качестве названия сущности. В некоторых случаях допускается использовать словосочетания, обозначающие особенности данных, которые представляются описываемой сущностью.

Устанавливая связи между сущностями, в нотации Мартина ее смысловое наполнение можно обозначать единственной глагольной формой, имеющей смысл связи от «левой» сущности к «правой» сущности, представляя в качестве «левой» сущности ту, у которой кардинальность связи в верхней границе равна «1». В случае установления связи многие — ко — многим (№М) «левой» сущностью является та, которая по логике модели является более значимой.

Следовательно, исходя из знаний о ER-диаграмме в нотации Мартина, набора требований и возможностей разработана модель базы данных данной информационной системы. Её структура представлена на рисунке 2.4.1.

Рисунок 2.4.1

ER-диаграмма в нотации Мартина



## 2.5 Создание представлений и готовых запросов

## Задание а.

В каком количестве и по какой цене имеется данный товар на данном складе?

В задании есть входные данные: наименование товара и номер склада, и имеются данные, которые нам необходимо выяснить, это количество товара и его цена. Для получения необходимой информации нам необходимо задействовать две таблицы, таблицу товаров (Рисунок 2.5.1), из которой выбрать записи необходимого товара, и таблицу складов (Рисунок 2.5.2), из которых необходимо выбрать указанный склад. Из таблицы товаров по номеру склада хранения товара мы получим искомый склад, если на нем есть товар.

SELECT t.naimen, s.adress, t.cena, t.kolvo

from tovar t, sklad s

WHERE

t.idsklad=s.nsklad

AND t.naimen='Ноутбук'

AND s.nsklad=1

Рисунок 2.5.1

Таблица товаров



Рисунок 2.5.2

Таблица складов

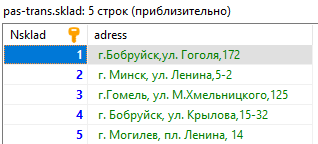
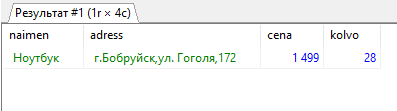


Рисунок 2.5.3

Результат выполнения запроса



## Задание b

Кто из поставщиков не поставляет данный товар по данной цене.

В задании есть входные данные: наименование товара и его цена, и имеются данные, которые нам необходимо выяснить, это поставщик , который не поставляет указанный товар по этой цене. Для получения необходимой информации нам необходимо задействовать две таблицы: таблицу товаров (Рисунок 2.5.5) и таблицу поставщиков (Рисунок 2.5.4). Из таблицы товаров мы выберем указанный товар с ценой не равной данной, и по этим записям выберем поставщиков.

SELECT p.fio FROM

postav p, tovar t

WHERE

t.idpostav=p.idpostav

AND t.naimen\_tovara='Мышка'

AND t.cena<>10

Рисунок 2.5.4

Таблица поставщиков



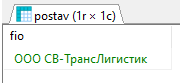
Рисунок 2.5.5

Таблица товаров



Рисунок 2.5.6

Результат выполнения запроса



## Задание c.

Какой товар и в каком количестве был отпущен на данную сумму по данной накладной.

В задании есть входные данные: номер накладной, и имеются данные, которые нам необходимо выяснить, это товар, его количество и сумма, который был отпущен. Для получения необходимой информации нам необходимо задействовать две таблицы: таблицу товаров (Рисунок 2.5.8) и таблицу накладных (Рисунок 2.5.7). Из таблицы накладных мы выберем указанную накладную по её номеру, и по этой записи выберем товар из таблицы товаров.

SELECT t.naimen\_tovara, n.kolvo FROM

nakladnaya n, tovar t

WHERE

n.idtovar=t.idtovar

AND n.idnakladnaya=2

AND n.summa=100

Рисунок 2.5.7

Таблица накладных

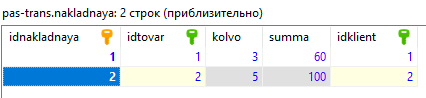


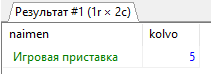
Рисунок 2.5.8

Таблица товаров



Рисунок 2.5.9

Результат выполнения запроса



## Задание d.

Кто из сотрудников данной фирмы является поставщиком товаров.

В задании необходимо найти записи у которых совпадает ФИО в таблице поставщиков (Рисунок 2.5.10) и в таблице сотрудников (Рисунок 2.5.11).

SELECT k.fio, k.dolgznost FROM

kadry k, postav p

WHERE

k.fio=p.fio

Рисунок 2.5.10

Таблица поставщиков



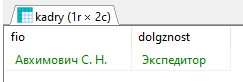
Рисунок 2.5.11

Таблица сотрудников



Рисунок 2.5.12

Результат выполнения запроса



## Задание e.

В каком банке открыл счет данный поставщик.

Входными данными для нас является название поставщика (Рисунок 2.5.14), выходными – банк и номер счета в банке (Рисунок 2.5.13).

SELECT p.fio, s.bank, s.numchet FROM

schet s, postav p

WHERE

p.fio= 'ЧТУП АксаЛайт'

AND p.idschet=s.id

Рисунок 2.5.13

Таблица банковских счетов

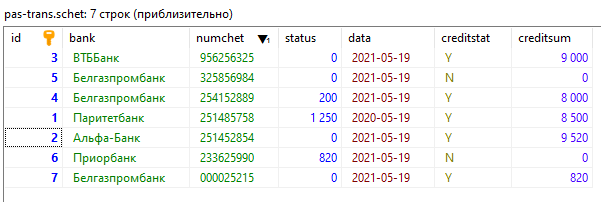


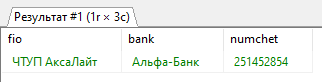
Рисунок 2.5.14

Таблица поставщиков



Рисунок 2.5.15

Результат выполнения запроса

****

## 2.6 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 SQL SECURITY DEFINER VIEW `Zd\_A` AS SELECT t.naimen, s.adress, t.cena, t.kolvo

from tovar t, sklad s

WHERE

t.idsklad=s.nsklad

AND t.naimen='Ноутбук'

AND s.nsklad=1

### Задание b

CREATE ALGORITHM = UNDEFINED SQL SECURITY DEFINER VIEW `Zd\_B` AS select `p`.`fio` AS `fio` from (`postav` `p` join `tovar` `t`) where ((`t`.`idpostav` = `p`.`idpostav`) and (`t`.`naimen` = 'Мышка') and (`t`.`cena` <> 10)) ;

### Задание c

CREATE ALGORITHM = UNDEFINED SQL SECURITY DEFINER VIEW `Zd\_C` AS SELECT t.naimen, n.kolvo FROM

nakladnaya n, tovar t

WHERE

n.idtovar=t.idtovar

AND n.idnakladnaya=2

AND n.summa=100

### Задание d

CREATE ALGORITHM = UNDEFINED SQL SECURITY DEFINER VIEW `Zd\_D` AS SELECT k.fio, k.dolgznost FROM

kadry k, postav p

WHERE

k.fio=p.fio

### Задание e

CREATE ALGORITHM = UNDEFINED SQL SECURITY DEFINER VIEW `Zd\_E` AS SELECT p.fio, s.bank, s.numchet FROM

schet s, postav p

WHERE

p.fio= 'ЧТУП АксаЛайт'

AND p.idschet=s.id

## 2.7. 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:

ALTER TABLE `firma`

ADD INDEX `Индекс 2` (`nazvan`);

ALTER TABLE `kadry`

ADD INDEX `Индекс 4` (`fio`);

ALTER TABLE `klient`

ADD INDEX `Индекс 4` (`fio`);

ALTER TABLE `postav`

ADD INDEX `Индекс 4` (`fio`);

ALTER TABLE `tovar`

ADD FULLTEXT INDEX `naimen` (`naimen`);

ALTER TABLE `transport`

ADD UNIQUE INDEX `nomer` (`nomer`);

# ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта была разработана база данных с заданными темами. В процессе разработки были использованы различные инструменты программирования выбранной среды разработки.

Разработанная база данных отличается простотой, краткостью и понятностью как управленческой, так и информационной части. Поскольку созданная база данных не подразумевает специализацию на деятельности конкретной организации, она может стать универсальным средством для применения в учебной деятельности. База данных может быть использована на предприятиях различных масштабов.

В ходе курсового проектирования была достигнута цель в заданной предметной области и выполнены следующие задачи:

* построение логической структуру базы данных на уровне взаимосвязей между объектами и на уровне взаимосвязей между атрибутами разных объектов, предварительно определив эти взаимосвязи;
* подтверждена правильность полученной структуры на экземплярах данных;
* выполнена нормализация отношений, определены функциональные зависимости и ключи;
* приведено каждое отношение к третьей нормальной форме;
* сформулированы запросы в терминах реляционной алгебры и показана правильность их обработки на примерах

База данных подвергалась многочисленным тестам для устранения большинства недостатков.

Была собрана необходимая информация, продумана структура проекта, разработана необходимая документация, отлажен и протестировано созданное приложение.

База данных протестирована и готова для демонстрации и практического использования. Пояснительная записка написана в соответствие с ГОСТами и требованиями, предъявляемыми к технической документации.

# 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. Другой пример проектирования базы данных (MySQL). URL: https://pro-prof.com/forums/topic/db\_example
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. Куликов С.С. Реляционные базы данных в примерах : практическое пособие для программистов и тестировщи- ков / С. С. Куликов. — Минск: Четыре четверти, 2020. — 424 с.
7. Материалы для скачивания по SQL Server // Microsoft URL: https://www.microsoft.com/ru-ru/sql-server/sql-server-downloads (дата обращения: 05.01.2021).
8. Нормализация отношений. Шесть нормальных форм // Хабр URL: https://habr.com/ru/post/254773/ (дата обращения: 05.01.2021).
9. Технологии баз данных. Лекция 3. Модель «Сущность-связь». URL: https://docplayer.ru/27886777-Model-sushchnost-svyaz-tehnologii-baz-dannyh-lekciya-3.html (дата обращения: 02.01.2021).
10. Учимся проектированию Entity Relationship — диаграмм // Хабр URL: https://habr.com/ru/post/440556/ (дата обращения: 02.01.2021).
11. Шениг Г.Ю. PostgresSQL 11. Мастерство разработки / пер. с англ. А.А. Слинкина. – Москва: ДМК Пресс, 2019. – 352 с.
12. Entity Relationship Diagram. URL: https://plantuml.com/ru/ie-diagram (дата обращения: 03.01.2021).
13. 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
14. Transact-SQL Reference (Database Engine) // Microsoft Docs URL: https://docs.microsoft.com/ru-ru/sql/t-sql/language-reference?view=sql-server-ver15 (дата обращения: 05.01.2021).

# ПРИЛОЖЕНИЕ А

## Полная структура 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 pas-trans

DROP DATABASE IF EXISTS `pas-trans`;

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

USE `pas-trans`;

-- Дамп структуры для таблица pas-trans.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,

UNIQUE KEY `FKschetf` (`idschet`) USING BTREE,

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

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

-- Дамп данных таблицы pas-trans.firma: ~1 rows (приблизительно)

DELETE FROM `firma`;

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

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

('ПАС-Транс', 'г. Бобруйск, ул. Гоголя, д.166-В, оф. 301', '+375 29 620 84 00', 1);

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

-- Дамп структуры для таблица pas-trans.kadry

DROP TABLE IF EXISTS `kadry`;

CREATE TABLE IF NOT EXISTS `kadry` (

`idkadry` int NOT NULL AUTO\_INCREMENT,

`fio` varchar(250) NOT NULL,

`dolgznost` varchar(500) NOT NULL,

`obrazovanie` enum('Среднее','Среднее-специальное','Высшее') CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

`zarplata` float unsigned NOT NULL,

`data\_priniatia` date NOT NULL,

`IDSCHET` int DEFAULT NULL,

PRIMARY KEY (`idkadry`),

UNIQUE KEY `idkadr` (`idkadry`),

KEY `FK\_kadry\_schet` (`IDSCHET`),

CONSTRAINT `FK\_kadry\_schet` FOREIGN KEY (`IDSCHET`) REFERENCES `schet` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='Сотрудники фирмы';

-- Дамп данных таблицы pas-trans.kadry: ~6 rows (приблизительно)

DELETE FROM `kadry`;

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

INSERT INTO `kadry` (`idkadry`, `fio`, `dolgznost`, `obrazovanie`, `zarplata`, `data\_priniatia`, `IDSCHET`) VALUES

(1, 'Парахневич А.С.', 'Директор', 'Высшее', 945, '2018-02-20', NULL),

(2, 'Аверкова И.В.', 'Бухгалтер', 'Высшее', 750, '2018-02-20', NULL),

(3, 'Долбик К.В.', 'Водитель', 'Среднее-специальное', 550, '2020-02-20', NULL),

(4, 'Савицкий А.Н.', 'Менеджер ', 'Среднее-специальное', 600, '2020-02-20', NULL),

(5, 'Зарянина Т.Н.', 'Продавец', 'Среднее-специальное', 550, '2020-02-20', NULL),

(6, 'Авхимович С. Н.', 'Экспедитор', 'Среднее', 120, '2021-05-25', NULL);

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

-- Дамп структуры для таблица pas-trans.klient

DROP TABLE IF EXISTS `klient`;

CREATE TABLE IF NOT EXISTS `klient` (

`idklient` int NOT NULL AUTO\_INCREMENT,

`fio` varchar(250) 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',

`idschet` int DEFAULT NULL,

PRIMARY KEY (`idklient`),

UNIQUE KEY `idclient` (`idklient`),

KEY `FKschet` (`idschet`),

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

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

-- Дамп данных таблицы pas-trans.klient: ~5 rows (приблизительно)

DELETE FROM `klient`;

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

INSERT INTO `klient` (`idklient`, `fio`, `uradr`, `idschet`) VALUES

(1, 'Говрицов А.К.', 'г. Бобруйск, ул. Крылова, 152', 5),

(2, 'Иванов К.А.', 'г. Минск, ул. Радиальная, 11', 6),

(3, 'Максимова А.Г.', 'г. Жодино, ул. Неманская, 15-6', 7),

(4, 'Мавриков П.К.', 'г. Бобруйск, ул. Минская, 14-2', NULL),

(5, 'Парахневич С.А.', 'г. Рогачев, ул. Юбилейная, 23', NULL);

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

-- Дамп структуры для таблица pas-trans.nakladnaya

DROP TABLE IF EXISTS `nakladnaya`;

CREATE TABLE IF NOT EXISTS `nakladnaya` (

`idnakladnaya` int NOT NULL AUTO\_INCREMENT,

`idtovar` int NOT NULL,

`kolvo` int unsigned NOT NULL,

`summa` float 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 `klient` (`idklient`),

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

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='Накладные на товар';

-- Дамп данных таблицы pas-trans.nakladnaya: ~2 rows (приблизительно)

DELETE FROM `nakladnaya`;

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

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

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

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

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

-- Дамп структуры для таблица pas-trans.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',

`uradr` 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`),

KEY `FKschetp` (`idschet`),

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

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='Поставщики';

-- Дамп данных таблицы pas-trans.postav: ~6 rows (приблизительно)

DELETE FROM `postav`;

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

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

(1, 'ООО СВ-ТрансЛигистик', 'Не указано', 4),

(2, 'ИП Тишкевич А.Т.', 'Не указано', 3),

(3, 'ООО СимулПлюс', 'Не указано', 2),

(4, 'ЧТУП Сом-Трак', 'Не указано', 3),

(5, 'ЧТУП АксаЛайт', 'Не указано', 2),

(6, 'Авхимович С. Н.', 'Не указано', 5);

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

-- Дамп структуры для таблица pas-trans.schet

DROP TABLE IF EXISTS `schet`;

CREATE TABLE IF NOT EXISTS `schet` (

`id` 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,

`creditsum` float unsigned NOT NULL,

PRIMARY KEY (`id`) USING BTREE,

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

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

-- Дамп данных таблицы pas-trans.schet: ~7 rows (приблизительно)

DELETE FROM `schet`;

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

INSERT INTO `schet` (`id`, `bank`, `numchet`, `status`, `data`, `creditstat`, `creditsum`) VALUES

(1, 'Паритетбанк', '251485758', 1250, '2020-05-19', 'Y', 8500),

(2, 'Альфа-Банк', '251452854', 0, '2021-05-19', 'Y', 9520),

(3, 'ВТББанк', '956256325', 0, '2021-05-19', 'Y', 9000),

(4, 'Белгазпромбанк', '254152889', 200, '2021-05-19', 'Y', 8000),

(5, 'Белгазпромбанк', '325856984', 0, '2021-05-19', 'N', 0),

(6, 'Приорбанк', '233625990', 820, '2021-05-19', 'N', 0),

(7, 'Белгазпромбанк', '000025215', 0, '2021-05-19', 'Y', 820);

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

-- Дамп структуры для таблица pas-trans.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=6 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

-- Дамп данных таблицы pas-trans.sklad: ~5 rows (приблизительно)

DELETE FROM `sklad`;

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

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

(1, 'г.Бобруйск,ул. Гоголя,172'),

(2, 'г. Минск, ул. Ленина,5-2'),

(3, 'г.Гомель, ул. М.Хмельницкого,125'),

(4, 'г. Бобруйск, ул. Крылова,15-32'),

(5, 'г. Могилев, пл. Ленина, 14');

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

-- Дамп структуры для таблица pas-trans.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',

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

`cena` int unsigned NOT NULL,

`idpostav` int NOT NULL,

`idsklad` int unsigned NOT NULL,

`Kolvo` int unsigned NOT NULL,

`sort` int NOT NULL,

PRIMARY KEY (`idtovar`),

UNIQUE KEY `idtovar` (`idtovar`),

KEY `FKpostavt` (`idpostav`),

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

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

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

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='Список товаров нормализованный';

-- Дамп данных таблицы pas-trans.tovar: ~6 rows (приблизительно)

DELETE FROM `tovar`;

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

INSERT INTO `tovar` (`idtovar`, `naimen`, `edizm`, `cena`, `idpostav`, `idsklad`, `Kolvo`, `sort`) VALUES

(1, 'Ноутбук', 'шт', 1499, 1, 1, 28, 0),

(2, 'Игровая приставка', 'шт', 2455, 2, 1, 21, 0),

(3, 'Мышка', 'шт', 15, 1, 2, 16, 0),

(4, 'Фен', 'шт', 59, 2, 2, 4, 0),

(5, 'Телвизор', 'шт', 4500, 1, 2, 5, 0),

(6, 'Мышка', 'шт', 10, 4, 1, 10, 0);

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

-- Дамп структуры для таблица pas-trans.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,

`tonnag` float unsigned NOT NULL,

`god` int unsigned NOT NULL,

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

`data\_perevozki` date DEFAULT NULL,

`nsklad` int unsigned DEFAULT NULL,

PRIMARY KEY (`idtransport`) USING BTREE,

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

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

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

-- Дамп данных таблицы pas-trans.transport: ~4 rows (приблизительно)

DELETE FROM `transport`;

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

INSERT INTO `transport` (`idtransport`, `nomer`, `marka`, `tonnag`, `god`, `vladelec`, `data\_perevozki`, `nsklad`) VALUES

(2, 'AI 5215-6', 'MAN', 24000, 2011, 'ПАС-Транс', NULL, NULL),

(3, 'AM 2532-6', 'DAF', 15000, 2018, 'ПАС-Транс', NULL, NULL),

(4, 'AK 1256-6', 'DAF', 24000, 2019, 'ПАС-Транс', NULL, NULL),

(5, 'AI 6326-6', 'Volvo', 10000, 2010, 'ПАС-Транс', NULL, NULL);

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

-- Дамп структуры для таблица pas-trans.zakaz

DROP TABLE IF EXISTS `zakaz`;

CREATE TABLE IF NOT EXISTS `zakaz` (

`id` int NOT NULL AUTO\_INCREMENT,

`data` date DEFAULT NULL,

`id\_client` int NOT NULL,

`id\_tovar` int NOT NULL,

`kolvo` int unsigned NOT NULL DEFAULT '0',

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

PRIMARY KEY (`id`) USING BTREE,

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

KEY `FKtovar` (`id\_tovar`),

KEY `FKclientz` (`id\_client`),

CONSTRAINT `FKclientz` FOREIGN KEY (`id\_client`) REFERENCES `klient` (`idklient`),

CONSTRAINT `FKtovar` FOREIGN KEY (`id\_tovar`) REFERENCES `tovar` (`idtovar`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='Список заказов п 10';

-- Дамп данных таблицы pas-trans.zakaz: ~3 rows (приблизительно)

DELETE FROM `zakaz`;

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

INSERT INTO `zakaz` (`id`, `data`, `id\_client`, `id\_tovar`, `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);

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

/\*!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) \*/;