ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 2](#_Toc72954723)

[1. Теоретическая часть – проектирование базы данных 3](#_Toc72954724)

[1.1 Анализ предметной области 3](#_Toc72954725)

[1.2 Построение логической структуры базы данных; 3](#_Toc72954726)

[1.3 Преобразование ER–диаграммы в схему базы данных 4](#_Toc72954727)

[1.4 Нормализация отношений 5](#_Toc72954728)

[2 Практическая часть – реализация базы данных в выбранной СУБД 13](#_Toc72954729)

[3 13](#_Toc72954730)

[ЗАКЛЮЧЕНИЕ 14](#_Toc72954731)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 15](#_Toc72954732)

# 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.

# 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ – ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

## 1.1 Анализ предметной области

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

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

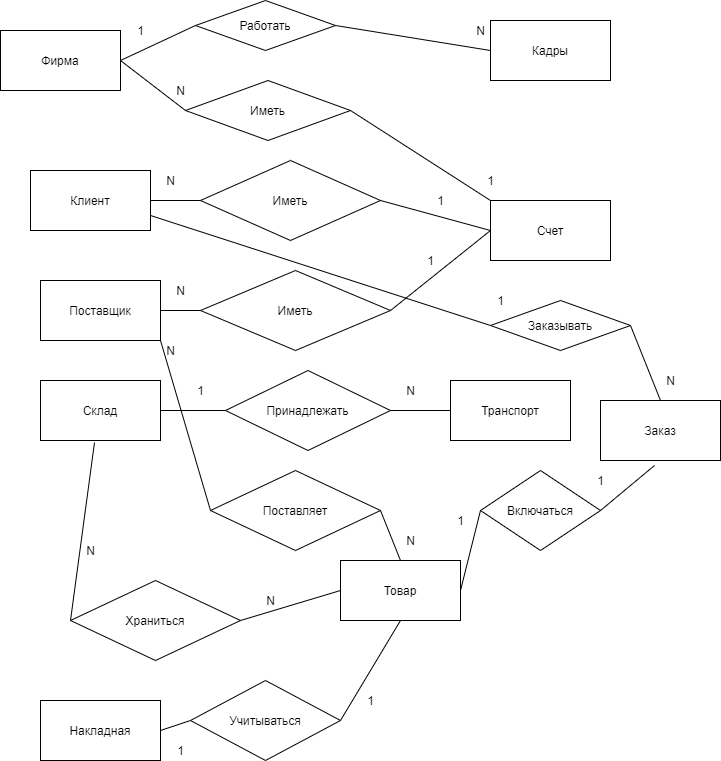
10.3аказ (№заказа, дата заказа, Ф.И.О. клиента, №счета клиента, наименование товара, количество, цена товара)

## 1.2 Построение логической структуры базы данных;

Исходя из выявленных сущностей, построим ER–диаграмму (рисунок 1.2.1)

Рисунок 1.2.1

ER-диаграмма.

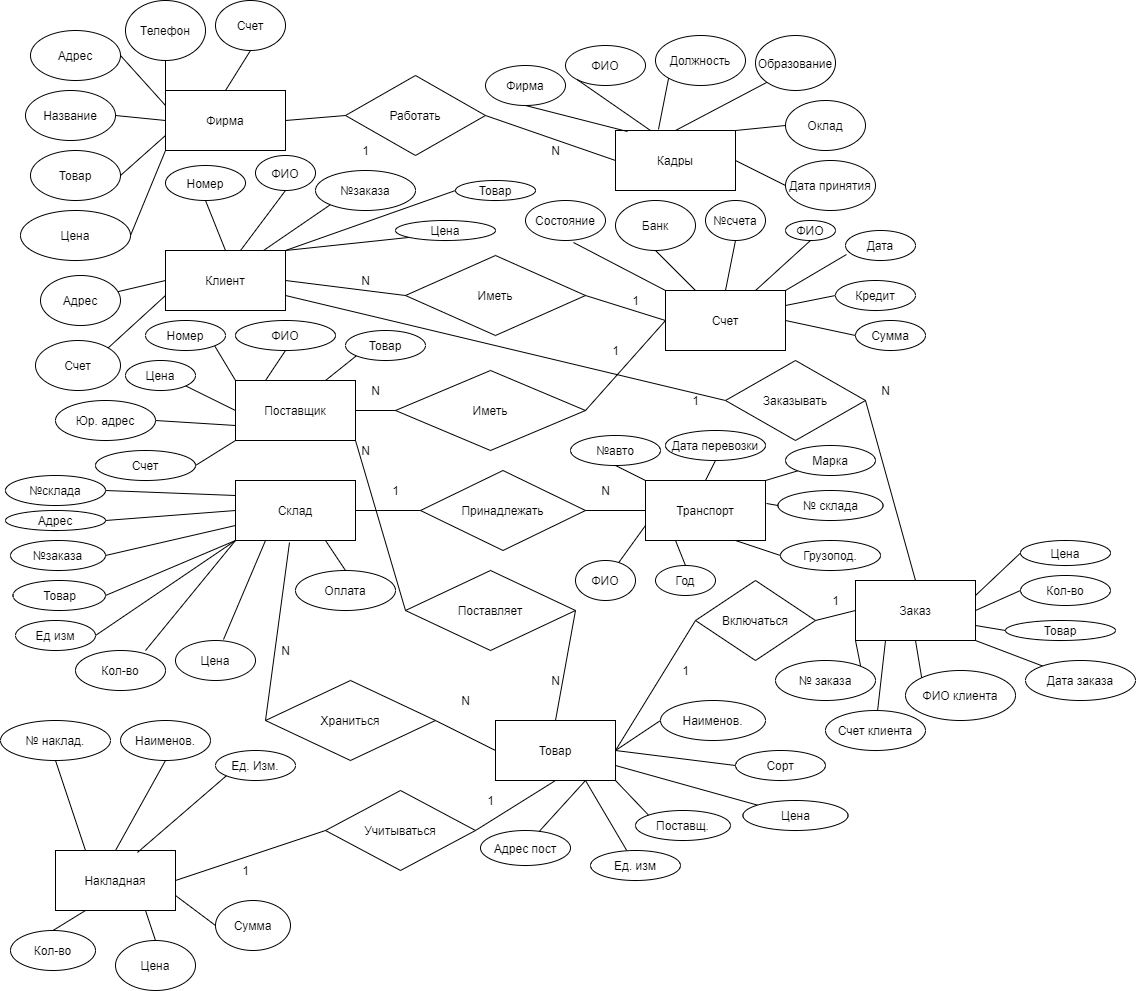


## 1.3 Преобразование ER–диаграммы в схему базы данных

База данных создается на основании схемы базы данных. Для преобразования ER–диаграммы в схему БД приведем уточненную ER–диаграмму, содержащую атрибуты сущностей (рисунок 1.3.1)

Рисунок 1.3.1

Уточненная ER-диаграмма фирмы.



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

## 1.4 Нормализация отношений

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

Таблица 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. 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 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 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 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 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 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 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 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 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 |

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 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 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 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 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 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 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 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 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 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 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 |

## 1.5 Определение ограничений целостности

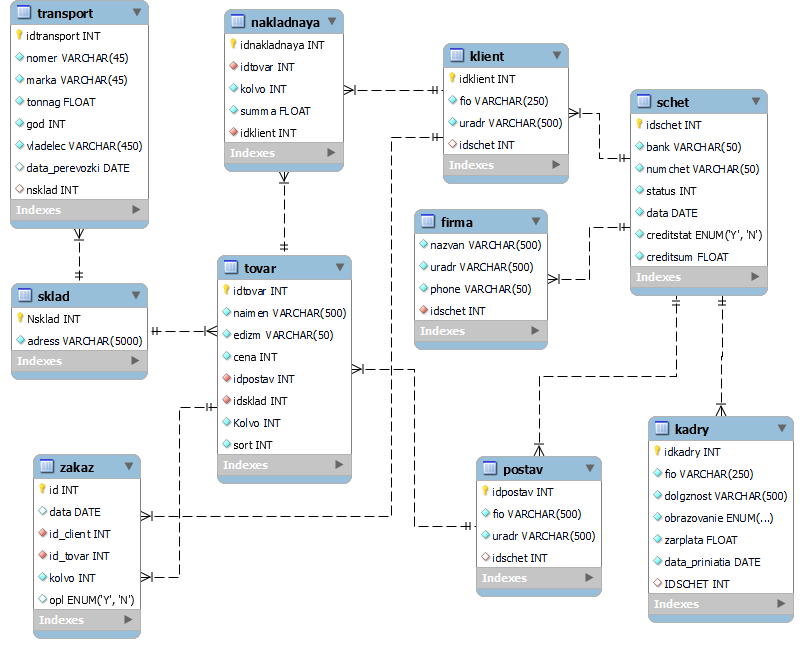
Атрибут Образование отношения КАДРЫ может принимать одно из следующих значений: «Среднее», «Среднее-специальное», «Высшее».

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». Окончательная схема БД (рисунок 1.5.1)

Рисунок 1.5.1

Окончательная схема БД



# 2 ПРАКТИЧЕСКАЯ ЧАСТЬ – РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ В ВЫБРАННОЙ СУБД

## 2.1 Реализация базы данных. Создание таблиц

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

1.Отношение schet (счет):

CREATE TABLE `schet` (

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

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

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

`status` INT(10) NOT NULL DEFAULT '0',

`data` DATE NOT NULL,

`creditstat` ENUM('Y','N') NOT NULL COLLATE 'utf8mb4\_0900\_ai\_ci',

`creditsum` FLOAT UNSIGNED NOT NULL,

PRIMARY KEY (`idschet`) USING BTREE,

UNIQUE INDEX `idschet` (`idschet`) USING BTREE )

COLLATE='utf8mb4\_0900\_ai\_ci'

ENGINE=InnoDB

2.Отношение postav (поставщик):

CREATE TABLE `postav` (

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

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

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

`idschet` INT(10) NULL DEFAULT NULL,

PRIMARY KEY (`idpostav`) USING BTREE,

UNIQUE INDEX `idpostav` (`idpostav`) USING BTREE,

INDEX `FKschetp` (`idschet`) USING BTREE,

CONSTRAINT `FKschetp` FOREIGN KEY (`idschet`) REFERENCES `pas-trans`.`schet` (`idschet`) ON UPDATE NO ACTION ON DELETE NO ACTION

)

COMMENT='Поставщики'

COLLATE='utf8mb4\_0900\_ai\_ci'

ENGINE=InnoDB

3.Отношение klient (клиент):

CREATE TABLE `klient` (

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

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

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

`idschet` INT(10) NULL DEFAULT NULL,

PRIMARY KEY (`idklient`) USING BTREE,

UNIQUE INDEX `idclient` (`idklient`) USING BTREE,

INDEX `FKschet` (`idschet`) USING BTREE,

CONSTRAINT `FKschet` FOREIGN KEY (`idschet`) REFERENCES `pas-trans`.`schet` (`idschet`) ON UPDATE NO ACTION ON DELETE NO ACTION)

COLLATE='utf8mb4\_0900\_ai\_ci'

ENGINE=InnoDB

4.Отношение tovar (товар):

CREATE TABLE `tovar` (

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

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

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

`cena` INT(10) UNSIGNED NOT NULL,

`idpostav` INT(10) NOT NULL,

`idsklad` INT(10) UNSIGNED NOT NULL,

`Kolvo` INT(10) UNSIGNED NOT NULL,

`sort` INT(10) NOT NULL,

PRIMARY KEY (`idtovar`) USING BTREE,

UNIQUE INDEX `idtovar` (`idtovar`) USING BTREE,

INDEX `FKpostavt` (`idpostav`) USING BTREE,

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

CONSTRAINT `FKpostavt` FOREIGN KEY (`idpostav`) REFERENCES `pas-trans`.`postav` (`idpostav`) ON UPDATE NO ACTION ON DELETE NO ACTION,

CONSTRAINT `FK\_tovar\_sklad` FOREIGN KEY (`idsklad`) REFERENCES `pas-trans`.`sklad` (`Nsklad`) ON UPDATE NO ACTION ON DELETE NO ACTION)

COMMENT='Список товаров нормализованный'

COLLATE='utf8mb4\_0900\_ai\_ci'

ENGINE=InnoDB

5.Отношение zakaz (заказ):

CREATE TABLE `zakaz` (

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

`data` DATE NULL DEFAULT NULL,

`id\_client` INT(10) NOT NULL,

`id\_tovar` INT(10) NOT NULL,

`kolvo` INT(10) UNSIGNED NOT NULL DEFAULT '0',

`opl` ENUM('Y','N') NULL DEFAULT NULL COLLATE 'utf8mb4\_0900\_ai\_ci',

PRIMARY KEY (`id`) USING BTREE,

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

INDEX `FKtovar` (`id\_tovar`) USING BTREE,

INDEX `FKclientz` (`id\_client`) USING BTREE,

CONSTRAINT `FKclientz` FOREIGN KEY (`id\_client`) REFERENCES `pas-trans`.`klient` (`idklient`) ON UPDATE NO ACTION ON DELETE NO ACTION,

CONSTRAINT `FKtovar` FOREIGN KEY (`id\_tovar`) REFERENCES `pas-trans`.`tovar` (`idtovar`) ON UPDATE NO ACTION ON DELETE NO ACTION)

COMMENT='Список заказов п 10'

COLLATE='utf8mb4\_0900\_ai\_ci'

ENGINE=InnoDB

6.Отношение nakladnaya (накладная):

CREATE TABLE `nakladnaya` (

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

`idtovar` INT(10) NOT NULL,

`kolvo` INT(10) UNSIGNED NOT NULL,

`summa` FLOAT UNSIGNED NOT NULL,

`idklient` INT(10) NOT NULL,

PRIMARY KEY (`idnakladnaya`) USING BTREE,

UNIQUE INDEX `idnaklad` (`idnakladnaya`) USING BTREE,

INDEX `FKtovarn` (`idtovar`) USING BTREE,

INDEX `FK\_nakladnaya\_klient` (`idklient`) USING BTREE,

CONSTRAINT `FKtovarn` FOREIGN KEY (`idtovar`) REFERENCES `pas-trans`.`tovar` (`idtovar`) ON UPDATE NO ACTION ON DELETE NO ACTION,

CONSTRAINT `FK\_nakladnaya\_klient` FOREIGN KEY (`idklient`) REFERENCES `pas-trans`.`klient` (`idklient`) ON UPDATE NO ACTION ON DELETE NO ACTION)

COMMENT='Накладные на товар'

COLLATE='utf8mb4\_0900\_ai\_ci'

ENGINE=InnoDB

7.Отношение sklad (склад):

CREATE TABLE `sklad` (

`Nsklad` INT(10) UNSIGNED NOT NULL AUTO\_INCREMENT,

`adress` VARCHAR(5000) NOT NULL COLLATE 'utf8mb4\_0900\_ai\_ci',

PRIMARY KEY (`Nsklad`) USING BTREE)

COLLATE='utf8mb4\_0900\_ai\_ci'

ENGINE=InnoDB

8.Отношение firma (фирма):

CREATE TABLE `firma` (

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

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

`phone` VARCHAR(50) NOT NULL DEFAULT '' COLLATE 'utf8mb4\_0900\_ai\_ci',

`idschet` INT(10) NOT NULL,

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

CONSTRAINT `FKschetf` FOREIGN KEY (`idschet`) REFERENCES `pas-trans`.`schet` (`idschet`) ON UPDATE NO ACTION ON DELETE NO ACTION)

COMMENT='Описание фирмы'

COLLATE='utf8mb4\_0900\_ai\_ci'

ENGINE=InnoDB

9.Отношение kadry (КАДРЫ):

CREATE TABLE `kadry` (

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

`fio` VARCHAR(250) NOT NULL COLLATE 'utf8mb4\_0900\_ai\_ci',

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

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

`zarplata` FLOAT UNSIGNED NOT NULL,

`data\_priniatia` DATE NOT NULL,

`IDSCHET` INT(10) NULL DEFAULT NULL,

PRIMARY KEY (`idkadry`) USING BTREE,

UNIQUE INDEX `idkadr` (`idkadry`) USING BTREE,

INDEX `FK\_kadry\_schet` (`IDSCHET`) USING BTREE,

CONSTRAINT `FK\_kadry\_schet` FOREIGN KEY (`IDSCHET`) REFERENCES `pas-trans`.`schet` (`idschet`) ON UPDATE NO ACTION ON DELETE NO ACTION)

COMMENT='Сотрудники фирмы'

COLLATE='utf8mb4\_0900\_ai\_ci'

ENGINE=InnoDB

10.Отношение transport (ТРАНСПОРТ):

CREATE TABLE `transport` (

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

`nomer` VARCHAR(45) NOT NULL COLLATE 'utf8mb4\_0900\_ai\_ci',

`marka` VARCHAR(45) NOT NULL COLLATE 'utf8mb4\_0900\_ai\_ci',

`tonnag` FLOAT UNSIGNED NOT NULL,

`god` INT(10) UNSIGNED NOT NULL,

`vladelec` VARCHAR(450) NOT NULL COLLATE 'utf8mb4\_0900\_ai\_ci',

`data\_perevozki` DATE NULL DEFAULT NULL,

`nsklad` INT(10) UNSIGNED NULL DEFAULT NULL,

PRIMARY KEY (`idtransport`) USING BTREE,

INDEX `FK\_transport\_sklad` (`nsklad`) USING BTREE,

CONSTRAINT `FK\_transport\_sklad` FOREIGN KEY (`nsklad`) REFERENCES `pas-trans`.`sklad` (`Nsklad`) ON UPDATE NO ACTION ON DELETE NO ACTION

)

COLLATE='utf8mb4\_0900\_ai\_ci'

ENGINE=InnoDB

## 2.2 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, из которой выбрать записи необходимого товара, и таблицу складов, из которых необходимо выбрать указанный склад. Из таблицы товаров по номеру склада хранения товара мы получим искомый склад, если на нем есть товар.

([Товар.склад=Склад.номер&Товар.наименование=’Ноутбук’&СкладюНаименование=’ 'Бобр Гоголя’])[Наименование товара, адрес склада, цена товара, количество товара]

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 2.2.1

Таблица товаров



Рисунок 2.2.2

Таблица складов

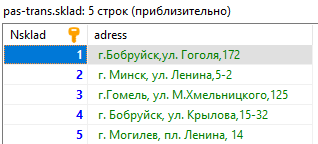
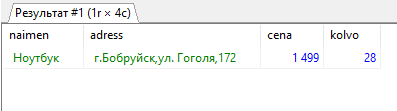


Рисунок 2.2.3

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 и таблицу поставщиков. Из таблицы товаров мы выберем указанный товар с ценой не равной данной, и по этим записям выберем поставщиков.

([Товар.ИДпоставщика=Поставщик.ИДпоставщика&Товар.наименование=’Мышка’&Товар.цена<>10=’])[Поставщик.ФИО]

SELECT p.fio FROM

postav p, tovar t

WHERE

t.idpostav=p.idpostav

AND t.naimen\_tovara='Мышка'

AND t.cena<>10

Рисунок 2.2.4

Таблица поставщиков



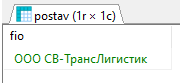
Рисунок 2.2.5

Таблица товаров



Рисунок 2.2.6

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 и таблицу накладных. Из таблицы накладных мы выберем указанную накладную по её номеру, и по этой записи выберем товар из таблицы товаров.

([Накладная.ИДтовара=Товар.ИДтовара])[Товар.Наименование, Накладная.Количество]

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 2.2.7

Таблица накладных

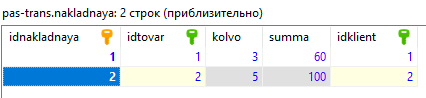


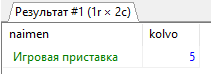
Рисунок 2.2.8

Таблица товаров



Рисунок 2.2.9

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 и в таблице сотрудников.

([Поставщики.ФИО=Кадры.ФИО])[Кадры.ФИО, Кадры.Должность]

SELECT k.fio, k.dolgznost FROM

kadry k, postav p

WHERE

k.fio=p.fio

Рисунок 2.2.10

Таблица поставщиков



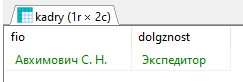
Рисунок 2.2.11

Таблица сотрудников



Рисунок 2.2.12

Результат выполнения запроса



## Задание e.

В каком банке открыл счет данный поставщик.

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

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

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

schet s, postav p

WHERE

p.fio= 'ЧТУП АксаЛайт'

AND p.idschet=s.id

Рисунок 2.2.13

Таблица банковских счетов

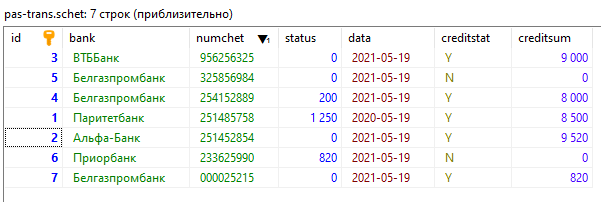


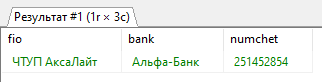
Рисунок 2.2.14

Таблица поставщиков



Рисунок 2.2.15

Результат выполнения запроса

****

# 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 покаана 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. Дейт, К. Введение в системы баз данных / К. Дейт. – 8-е изд. – Москва:Санкт-Петербург : Киев : Вильямс, 2017. – 1328 с.
2. Куликов С.С. Реляционные базы данных в примерах : практическое пособие для программистов и тестировщи- ков / С. С. Куликов. — Минск: Четыре четверти, 2020. — 424 с.
3. Коннолли Т., Бегг К. Базы данных: проектирование, реализация, сопровождение. Теория и практика. – 3-е изд.: Пер. с англ.: Уч. пос. – М.: Изд. дом "Вильямс", 2003. – 1440 с
4. Куликов С.С. Работа с MySQL, MS SQL Server и Oracle в примерах (ссылка для скачивания http://svyatoslav.biz/database\_book/)
5. 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
6. Шениг Г.Ю. PostgresSQL 11. Мастерство разработки / пер. с англ. А.А. Слинкина. – Москва: ДМК Пресс, 2019. – 352 с.
7. 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
8. Учимся проектированию Entity Relationship — диаграмм // Хабр URL: https://habr.com/ru/post/440556/ (дата обращения: 02.01.2021).
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. Entity Relationship Diagram. URL: https://plantuml.com/ru/ie-diagram (дата обращения: 03.01.2021).
11. 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).
12. Нормализация отношений. Шесть нормальных форм // Хабр URL: https://habr.com/ru/post/254773/ (дата обращения: 05.01.2021).
13. Материалы для скачивания по SQL Server // Microsoft URL: https://www.microsoft.com/ru-ru/sql-server/sql-server-downloads (дата обращения: 05.01.2021).
14. Другой пример проектирования базы данных (MySQL). URL: https://pro-prof.com/forums/topic/db\_example