ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 2](#_Toc73104236)

[1. Теоретическая часть – проектирование базы данных 3](#_Toc73104237)

[1.1 Анализ предметной области 3](#_Toc73104238)

[1.2 Построение логической структуры базы данных; 3](#_Toc73104239)

[1.3 Преобразование ER–диаграммы в схему базы данных 4](#_Toc73104240)

[1.4 Нормализация отношений 5](#_Toc73104241)

[1.5 Определение ограничений целостности 12](#_Toc73104242)

[2 Практическая часть – реализация базы данных в выбранной СУБД 14](#_Toc73104243)

[2.1 Реализация базы данных 14](#_Toc73104244)

[2.2 Создание запросов 18](#_Toc73104245)

[Задание а. 18](#_Toc73104246)

[Задание b 19](#_Toc73104247)

[Задание c. 20](#_Toc73104248)

[Задание d. 21](#_Toc73104249)

[Задание e. 23](#_Toc73104250)

[2.3 Создание представлений 24](#_Toc73104251)

[Задание a 24](#_Toc73104252)

[Задание b 24](#_Toc73104253)

[Задание c 25](#_Toc73104254)

[Задание d 25](#_Toc73104255)

[Задание e 26](#_Toc73104256)

[2.4. Создание индексов 26](#_Toc73104257)

[ЗАКЛЮЧЕНИЕ 28](#_Toc73104258)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 29](#_Toc73104259)

# 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 – применение на практике знаний, полученных в процессе изучения дисциплины "Базы данных", и получение практических навыков создания автоматизированных информационных систем (АИС), основанных на базах 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).

# 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

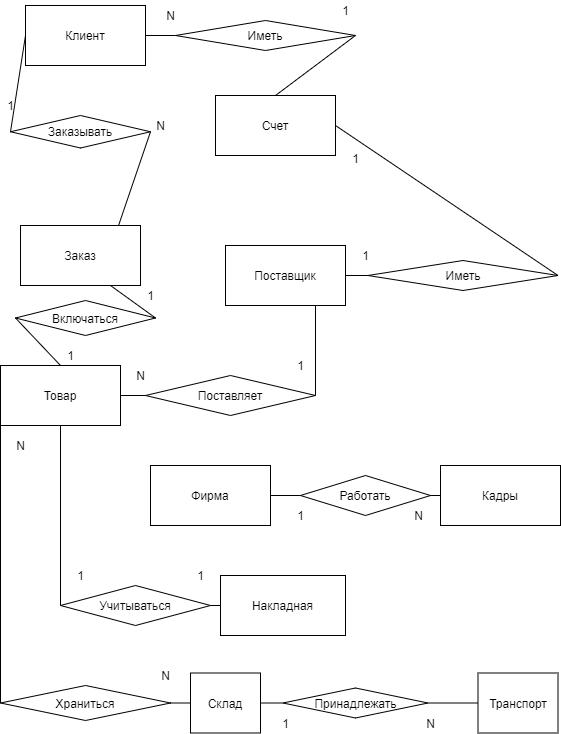
10.3аказ (№заказа, дата заказа, Ф.И.О. клиента, №счета клиента, наименование товара, количество, цена товара)

## 1.2 Построение логической структуры базы данных;

Исходя из выявленных сущностей, построим ER–диаграмму (рисунок 1)

Рисунок 1

ER-диаграмма фирмы.

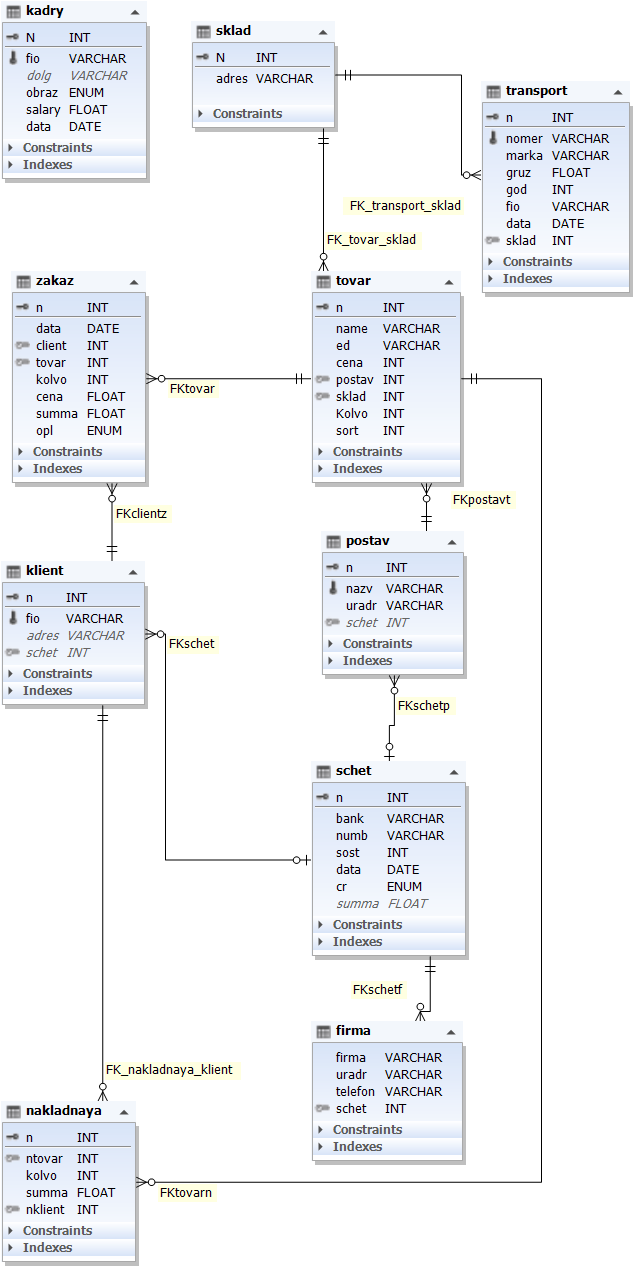


## 1.3 Преобразование ER–диаграммы в схему базы данных

База данных создается на основании схемы базы данных. Для преобразования ER–диаграммы в схему БД приведем уточненную ER–диаграмму, содержащую атрибуты сущностей (рисунок 2).

Рисунок 2

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, поскольку название в дальнейшей работе фигурировать не будет, и фирма у нас всего одна, оно и будет ключевым полем.

Таблица 1.1 Отношение ФИРМА

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

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

Таблица 1.2 Отношение СКЛАД

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

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

Таблица 1.3 Схема отношения КАДРЫ

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

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

Таблица 1.4 Схема отношения ПОСТАВЩИК

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

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

Таблица 1.5 Схема отношения ТРАНСПОРТ

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

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

Таблица 1.6 Схема отношения НАКЛАДНАЯ

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

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

Таблица 1.7 Схема отношения ТОВАР

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Наименование | Naimen\_tovara | C(500) | Обязательное поле |
| Ед. изм. | Ed\_izm | C(50) | Обязательное поле |
| Сорт | sort | N | Обязательное поле |
| Цена | cena | N | Обязательное поле |
| Поставщик | postav | C(500) | Обязательное поле |
| Адрес поставщика | adr | C(500) | Обязательное поле |
| Идентификатор | id | N | Первичный ключ |

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

Таблица 1.8 Схема отношения КЛИЕНТ

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

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

Таблица 1.9 Схема отношения СЧЕТ

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

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

Таблица 1.10 Схема отношения ЗАКАЗ

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

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

Таблица 1.11 Схема отношения СЧЕТ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Идентификатор | idschet | N | Первичный ключ |
| № счета | schet | C(100) | Обязательное поле |
| Наименов. банка | bank | C(100) | 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 1.12 Схема отношения ПОСТАВЩИК

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

Таблица 1.13 Схема отношения КЛИЕНТ

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

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

Таблица 1.14 Схема отношения ТОВАР

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

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

Таблица 1.15 Схема отношения ЗАКАЗ

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

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

Таблица 1.16 Схема отношения НАКЛАДНАЯ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| №накладной | idnakladnaya | N | Первичный ключ |
| Идентификатор товара | idtovar | N | Обязательное поле, внешний ключ к ТОВАР |
| 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 | nsklad | N | Первичный ключ |
| Адрес склада | adr | C(500) | Обязательное поле |

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

Таблица 1.18 Схема отношения ФИРМА

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

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

Таблица 1.19 Схема отношения КАДРЫ

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

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

Таблица 1.20 Схема отношения ТРАНСПОРТ

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

## 1.5 Определение ограничений целостности

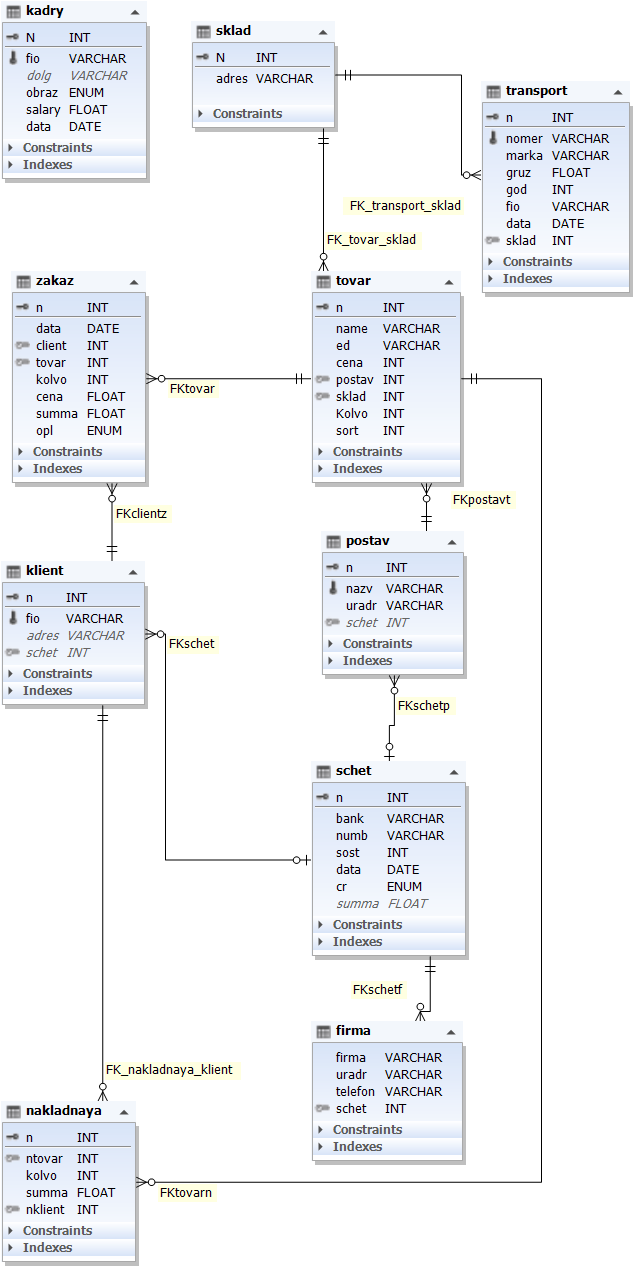
Атрибут Образование отношения КАДРЫ может принимать одно из следующих значений: ‘Высшее’.’Среднеспециальное’,’Среднее’,’Базовое’.

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’

Рисунок 3

Окончательная схема БД



# Практическая часть – реализация базы данных в выбранной СУБД

## 2.1 Реализация базы данных

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)

)

## 2.2 Создание запросов

### Задание а.

Кто из сотрудников был принят на работу на указанную дату?

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

Сформулируем запрос в терминах реляционной алгебры:

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

Запишем запрос на языке DML:

SELECT FIO, DATA

FROM matvey.kadry

WHERE DATA<'2021-01-01'

Результат выполнения запроса:

Рисунок 2.1

Таблица товаров



Рисунок 2.2

Таблица складов

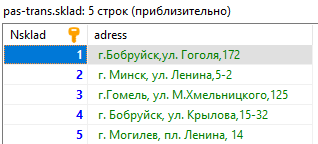
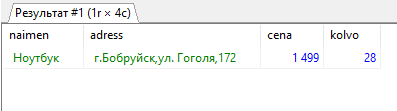


Рисунок 2.3

Результат выполнения запроса



### Задание b

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

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

Сформулируем запрос в терминах реляционной алгебры:

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

Запишем запрос на языке DML:

SELECT marka, gruz, god, DATA

FROM matvey.transport

WHERE

DATA<'2015-01-01'

AND gruz=1000

Рисунок 2.4

Таблица поставщиков



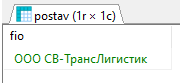
Рисунок 2.5

Таблица товаров



Рисунок 2.6

Результат выполнения запроса



### Задание c.

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

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

Сформулируем запрос в терминах реляционной алгебры:

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

Запишем запрос на языке DML:

SELECT p.fio, p.uradr

FROM matvey.tovar t, matvey.postav p

WHERE

p.idpostav=t.idpostav

AND t.sort=1

AND t.cena=500

AND t.naimen\_tovara='Телефон'

Рисунок 2.7

Таблица накладных

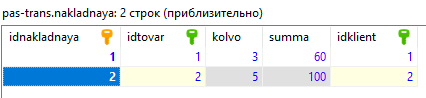


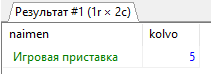
Рисунок 2.8

Таблица товаров



Рисунок 2.9

Результат выполнения запроса



### Задание d.

В каком банке открыл счет данный поставщик.

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

Сформулируем запрос в терминах реляционной алгебры:

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

Запишем запрос на языке DML:

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

matvey.schet s, matvey.postav p

WHERE

p.fio= 'Название'

AND p.idschet=s.idschet

Рисунок 2.10

Таблица поставщиков



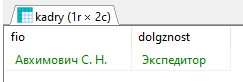
Рисунок 2.11

Таблица сотрудников



Рисунок 2.12

Результат выполнения запроса



### Задание e.

Является ли владелец автомобиля с данной фамилией работником данной фирмы.

В задании необходимо найти записи у которых совпадает ФИО в таблице владельцев автомобилей и в таблице сотрудников. Необходимо задействовать 2 таблицы – транспорта и кадров. Из таблицы

Сформулируем запрос в терминах реляционной алгебры:

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

Запишем запрос на языке DML:

SELECT fio, dolg

FROM matvey.kadry

INNER JOIN matvey.transport

ON kadry.fio=transport.vladelec

Рисунок 2.13

Таблица банковских счетов

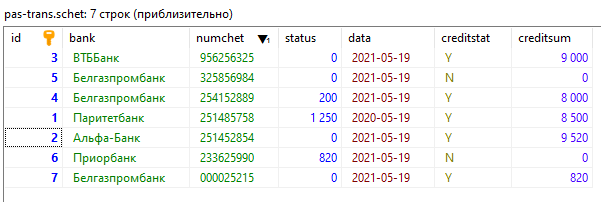


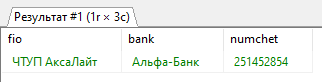
Рисунок 2.14

Таблица поставщиков



Рисунок 2.15

Результат выполнения запроса

****

## 2.3 Создание представлений

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

### Задание a

CREATE OR REPLACE DEFINER = `root` @`localhost` SQL SECURITY DEFINER VIEW matvey.a

(

fio,

data

)

AS

SELECT `matvey`.`kadry`.`fio` AS `fio`, `matvey`.`kadry`.`data` AS `data`

FROM `matvey`.`kadry`

WHERE (`matvey`.`kadry`.`data` < '2021-01-01')

### Задание b

CREATE OR REPLACE DEFINER = `root` @`localhost` SQL SECURITY DEFINER VIEW matvey.b

(

marka,

gruz,

god,

`DATA`

)

AS

SELECT `matvey`.`transport`.`marka` AS `marka`,

`matvey`.`transport`.`gruz` AS `gruz`,

`matvey`.`transport`.`god` AS `god`,

`matvey`.`transport`.`data` AS `DATA`

FROM `matvey`.`transport`

WHERE ( (`matvey`.`transport`.`data` < '2015-01-01')

AND (`matvey`.`transport`.`gruz` = 1000))

### Задание c

CREATE OR REPLACE DEFINER = `root` @`localhost` SQL SECURITY DEFINER VIEW matvey.c

(

fio,

uradr

)

AS

SELECT `p`.`fio` AS `fio`, `p`.`uradr` AS `uradr`

FROM `matvey`.`tovar` `t` JOIN `matvey`.`postav` `p`

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

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

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

AND (`t`.`naimen\_tovara` = 'Телефон'))

### Задание d

CREATE OR REPLACE DEFINER = `root` @`localhost` SQL SECURITY DEFINER VIEW matvey.d

(

fio,

bank,

numchet

)

AS

SELECT `p`.`fio` AS `fio`,

`s`.`bank` AS `bank`,

`s`.`numchet` AS `numchet`

FROM `matvey`.`schet` `s` JOIN `matvey`.`postav` `p`

WHERE ( (`p`.`fio` = 'Название')

AND (`p`.`idschet` = `s`.`idschet`))

### Задание e

CREATE OR REPLACE DEFINER = `root` @`localhost` SQL SECURITY DEFINER VIEW matvey.e

(

fio,

dolg

)

AS

SELECT `matvey`.`kadry`.`fio` AS `fio`, `matvey`.`kadry`.`dolg` AS `dolg`

FROM (

`matvey`.`kadry`

JOIN `matvey`.`transport`

ON ((`matvey`.`kadry`.`fio` = `matvey`.`transport`.`vladelec`)))

## 2.4. Создание индексов

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

ALTER TABLE postav

ADD UNIQUE INDEX i4 (nazv);

ALTER TABLE postav

ADD CONSTRAINT FKschetp FOREIGN KEY (schet)

REFERENCES schet (n);

ALTER TABLE klient

ADD UNIQUE INDEX i3 (fio);

ALTER TABLE klient

ADD CONSTRAINT FKschet FOREIGN KEY (schet)

REFERENCES schet (n);

ALTER TABLE firma

ADD INDEX I1 (firma);

ALTER TABLE firma

ADD CONSTRAINT FKschetf FOREIGN KEY (schet)

REFERENCES schet (n);

ALTER TABLE transport

ADD UNIQUE INDEX i5 (nomer);

ALTER TABLE transport

ADD CONSTRAINT FK\_transport\_sklad FOREIGN KEY (sklad)

REFERENCES sklad (N);

ALTER TABLE tovar

ADD INDEX i5 (name);

ALTER TABLE tovar

ADD CONSTRAINT FK\_tovar\_sklad FOREIGN KEY (sklad)

REFERENCES sklad (N);

ALTER TABLE tovar

ADD CONSTRAINT FKpostavt FOREIGN KEY (postav)

REFERENCES postav (n);

ALTER TABLE zakaz

ADD CONSTRAINT FKclientz FOREIGN KEY (client)

REFERENCES klient (n);

ALTER TABLE zakaz

ADD CONSTRAINT FKtovar FOREIGN KEY (tovar)

REFERENCES tovar (n);

ALTER TABLE nakladnaya

ADD CONSTRAINT FK\_nakladnaya\_klient FOREIGN KEY (nklient)

REFERENCES klient (n);

ALTER TABLE nakladnaya

ADD CONSTRAINT FKtovarn FOREIGN KEY (ntovar)

REFERENCES tovar (n);

ALTER TABLE kadry

ADD UNIQUE INDEX i2 (fio);

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

# ЗАКЛЮЧЕНИЕ

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

В ходе курсовой работы была разработана и создана база данных "Фирма", 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.

# 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