СОДЕРЖАНИЕ

[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 9»

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

Целью курсового проекта является систематизация знаний и накопление первоначального опыта в области проектирования баз данных и автоматизированных рабочих мест в составе корпоративных информационных систем.

В процессе работы осуществляется постановка задачи, выбор методики проектирования, обследование и анализ требований предметной области, проектирование структуры базы данных и разработка полной АИС, для автоматизации функций пользователей.

При выполнении проекта отрабатываются навыки работы с базами данных, проводится работа со специальной технической и справочной литературой, самостоятельно принимаются решения и оценивается их качество. Осваивается концептуальное проектирование и разработка инфологической модели БД в виде ER-диаграмм. Развиваются практические навыки при физическом проектировании баз данных.

# 1. Проектирование базы данных

## 1.1 Анализ предметной области

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

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

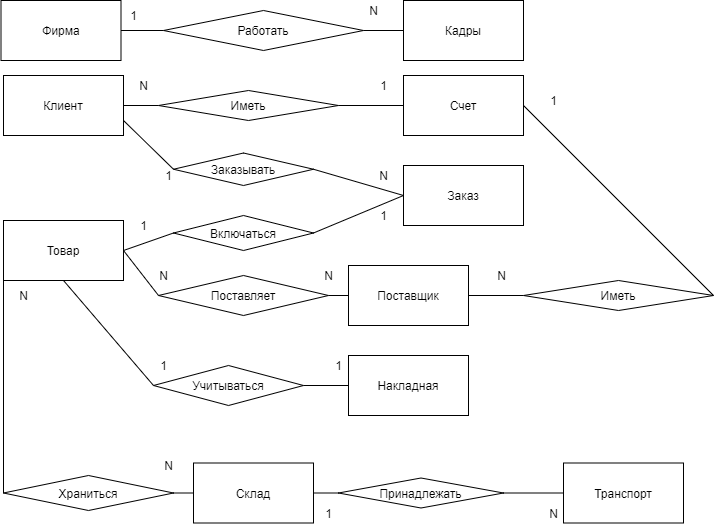
10.3аказ (№заказа, дата заказа, Ф.И.О. клиента, №счета клиента, наименование товара, количество, цена товара)

## 1.2 Построение логической структуры базы данных;

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

Рисунок 1.1

ER-диаграмма фирмы.

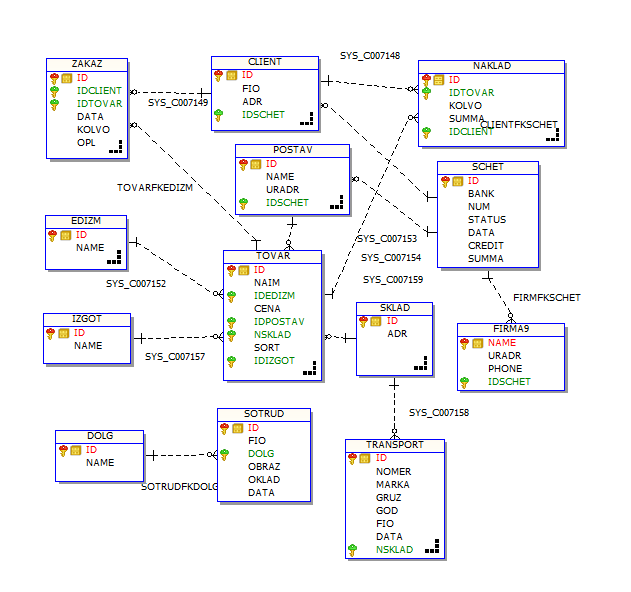


## 1.3 Преобразование ER–диаграммы в схему базы данных

Для преобразования ER–диаграммы в схему БД уточним ER–диаграмму, дополнив её атрибутами сущностей (рисунок 1.2).

Рисунок 1.2

Уточненная ER-диаграмма фирмы.



Преобразование ER–диаграммы в схему БД выполняется путем сопоставления каждой сущности и каждой связи, имеющей атрибуты, отношения (таблицы) БД. В первую очередь необходимо определить нужные атрибуты, так как в задании они даны избыточно, то есть произвести нормализацию, затем определить типы полей будущей базы данных, первичные ключи, индексные поля, связи между таблицами, ограничения целостности БД. Реализовать связь типа 1:N (один-ко-многим) между отношениями через внешний ключ. Ключ вводится для того отношения, к которому осуществляется множественная связь.

## 1.4 Нормализация отношений

Ключом отношения ФИРМА можно принять 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 | name | C(300) | Обязательное поле, первичный ключ |
| Юр. адрес | uradr | C(300) | Обязательное поле |
| Телефон | phone | C(30) | Необязательное поле |
| № счета банка | idschet | 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 | id | 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 |

Ключом отношения КАДРЫ можно было бы выбрать атрибут ФИО, однако 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 | id | 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 | id | 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 | id | 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 | Naim | 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 | id | 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 | id | 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 | id | 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 | id | N | Первичный ключ |
| № счета | num | C(100) | Обязательное поле |
| Наименов. банка | bank | C(100) | Обязательное поле |
| Состояние счета | status | N | Обязательное поле |
| Дата | data | D | Обязательное поле |
| Наличие кредита | credit | 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. | name | C(500) | Обязательное поле |
| Юр. адр. | uradr | 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 | id | 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 №Поставщика, 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 nsklad. Так же создадим таблицы ИЗГОТОВИТЕЛЬ и ЕДИНИЦЫ, куда вынесем информацию об изготовителе и единицах измерения товара соответственно.

Таблица 1.14 Схема отношения ТОВАР

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Идентификатор | idtovar | N | Первичный ключ |
| Наименование | naim | C(500) | Обязательное поле |
| Ед. изм. | idedizm | C(50) | Обязательное поле, внешний ключ к ЕДИНИЦЫ |
| Сорт | sort | N | Обязательное поле |
| Цена | cena | N | Обязательное поле |
| Идентификатор склад | nsklad | N | Обязательное поле, внешний ключ к СКЛАД |
| Поставщик | idpostav | N | Обязательное поле, внешний ключ к ПОСТАВЩИК |
| Изготовитель | idizgot | N | Обязательное поле, внешний ключ к ИЗГОТОВИТЕЛЬ |

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 | id | N | Первичный ключ |
| Дата заказа | data | D | Обязательное поле |
| Идентификатор клиента | idсlient | 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 | 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 |
| Клиент | idclient | N | Обязательное поле, внешний ключ к КЛИЕНТ |

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

Таблица 1.17 Схема отношения СКЛАД

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| №склада | т | 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 | name | 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. Так же вынесем название должности в отдельную таблицу ДОЛЖНОСТЬ и установим с ней связь.

Таблица 1.19 Схема отношения КАДРЫ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Идентификатор | id | N | Первичный ключ суррогатный |
| ФИО сотр. | fio | C(150) | Обязательное поле |
| Должность | dolg | N | Обязательное поле, внешний ключ к ДОЛЖНОСТЬ |
| 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 | id | 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 |

Отношение ДОЛЖНОСТЬ, первичный ключ ID, суррогатный.

Таблица 1.21 Схема отношения ДОЛЖНОСТЬ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Идентификатор | id | N | Первичный ключ |
| Должность | name | N | Обязательное поле |

Отношение ИЗГОТОВИТЕЛЬ, первичный ключ ID, суррогатный.

Таблица 1.22 Схема отношения ИЗГОТОВИТЕЛЬ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Идентификатор | id | N | Первичный ключ |
| Изготовитель | name | N | Обязательное поле |

Отношение ЕДИНИЦЫ, первичный ключ ID, суррогатный.

Таблица 1.23 Схема отношения ЕДИНИЦЫ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Идентификатор | id | N | Первичный ключ |
| Единицы | name | N | Обязательное поле |

## 1.5 Определение ограничений целостности

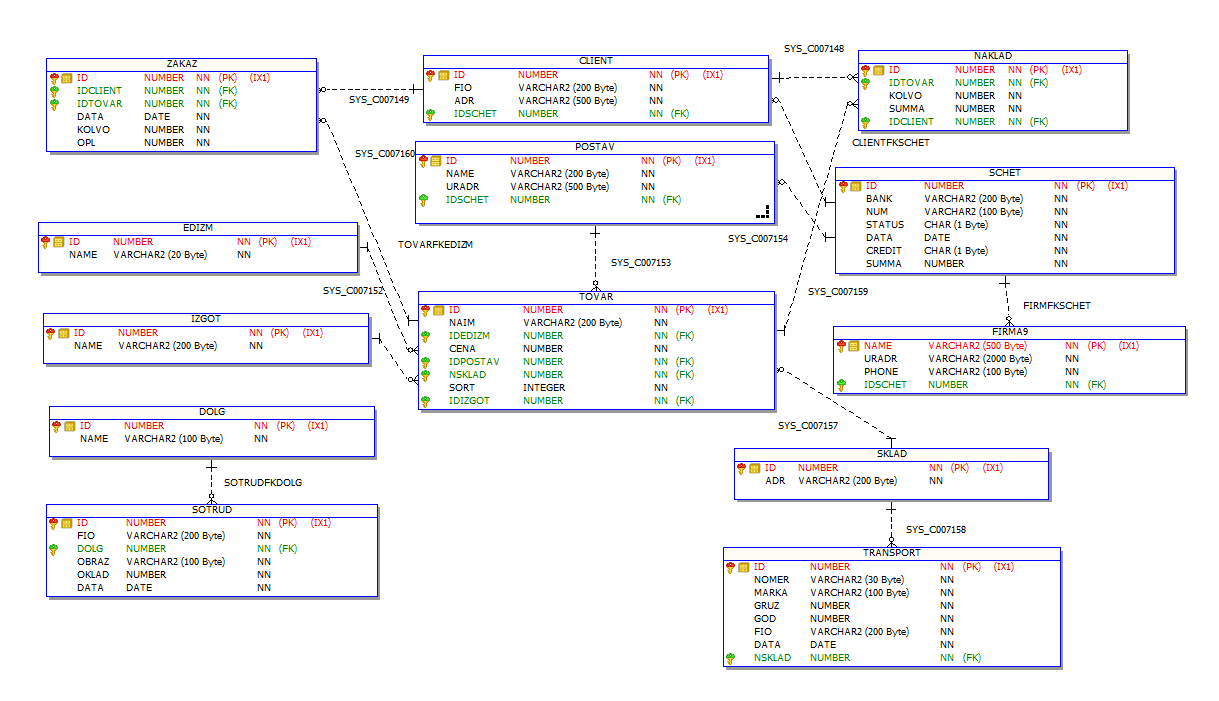
Атрибут Образование отношения КАДРЫ может принимать одно из следующих значений: ‘Высшее’.’Среднееспециальное’,’Среднее’,’Базовое’.

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’. Таким образом получаем окончательно следующую схему БД (рисунок 1.3).

Рисунок 1.3

Окончательная схема БД



# Реализация базы данных в выбранной СУБД

## 2.1 Реализация базы данных

Приведем описание схемы БД на языке DDL с учетом наших связей и ограничений. В качестве СУБД будет использоваться ORACLE. **Oracle Database** — это объектно-реляционная СУБД, обеспечивающая высокое быстродействие и максимально гибкую настройку одновременно с большими возможностями по её администрированию.

Приведем DDL запросы необходимые для создания нужных нам таблиц.

**Отношение СЧЕТ**

CREATE TABLE FIRMA.SCHET

(

ID NUMBER NOT NULL,

BANK VARCHAR2(200 BYTE) NOT NULL,

NUM VARCHAR2(100 BYTE) NOT NULL,

STATUS CHAR(1 BYTE) NOT NULL,

DATA DATE NOT NULL,

CREDIT CHAR(1 BYTE) NOT NULL,

SUMMA NUMBER NOT NULL

)

**Отношение ПОСТАВЩИК**

CREATE TABLE FIRMA.POSTAV

(

ID NUMBER NOT NULL,

NAME VARCHAR2(200 BYTE) NOT NULL,

URADR VARCHAR2(500 BYTE) NOT NULL,

IDSCHET NUMBER NOT NULL

)

**Отношение КЛИЕНТ**

CREATE TABLE FIRMA.CLIENT

(

ID NUMBER NOT NULL,

FIO VARCHAR2(200 BYTE) NOT NULL,

ADR VARCHAR2(500 BYTE) NOT NULL,

IDSCHET NUMBER NOT NULL

)

**Отношение ТОВАР**

CREATE TABLE FIRMA.TOVAR

(

ID NUMBER NOT NULL,

NAIM VARCHAR2(200 BYTE) NOT NULL,

IDEDIZM NUMBER NOT NULL,

CENA NUMBER NOT NULL,

IDPOSTAV NUMBER NOT NULL,

NSKLAD NUMBER NOT NULL,

SORT INTEGER NOT NULL,

IDIZGOT NUMBER NOT NULL

)

**Отношение ЗАКАЗ**

CREATE TABLE FIRMA.ZAKAZ

(

ID NUMBER NOT NULL,

IDCLIENT NUMBER NOT NULL,

IDTOVAR NUMBER NOT NULL,

DATA DATE NOT NULL,

KOLVO NUMBER NOT NULL,

OPL NUMBER NOT NULL

)

**Отношение НАКЛАДНАЯ**

CREATE TABLE FIRMA.NAKLAD

(

ID NUMBER NOT NULL,

IDTOVAR NUMBER NOT NULL,

KOLVO NUMBER NOT NULL,

SUMMA NUMBER NOT NULL,

IDCLIENT NUMBER NOT NULL

)

**Отношение СКЛАД**

CREATE TABLE FIRMA.SKLAD

(

ID NUMBER NOT NULL,

ADR VARCHAR2(200 BYTE) NOT NULL

)

**Отношение ФИРМА**

CREATE TABLE FIRMA.FIRMA9

(

NAME VARCHAR2(500 BYTE) NOT NULL,

URADR VARCHAR2(2000 BYTE) NOT NULL,

PHONE VARCHAR2(100 BYTE) NOT NULL,

IDSCHET NUMBER NOT NULL

)

**Отношение КАДРЫ**

CREATE TABLE FIRMA.SOTRUD

(

ID NUMBER NOT NULL,

FIO VARCHAR2(200 BYTE) NOT NULL,

DOLG NUMBER NOT NULL,

OBRAZ VARCHAR2(100 BYTE) NOT NULL,

OKLAD NUMBER NOT NULL,

DATA DATE NOT NULL

)

**Отношение ТРАНСПОРТ**

CREATE TABLE FIRMA.TRANSPORT

(

ID NUMBER NOT NULL,

NOMER VARCHAR2(30 BYTE) NOT NULL,

MARKA VARCHAR2(100 BYTE) NOT NULL,

GRUZ NUMBER NOT NULL,

GOD NUMBER NOT NULL,

FIO VARCHAR2(200 BYTE) NOT NULL,

DATA DATE NOT NULL,

NSKLAD NUMBER NOT NULL

)

**Отношение ДОЛЖНОСТЬ**

CREATE TABLE FIRMA.DOLG

(

ID NUMBER NOT NULL,

NAME VARCHAR2(100 BYTE) NOT NULL

)

**Отношение ЕДИНИЦЫ**

CREATE TABLE FIRMA.EDIZM

(

ID NUMBER NOT NULL,

NAME VARCHAR2(20 BYTE) NOT NULL

)

**Отношение ИЗГОТОВИТЕЛЬ**

CREATE TABLE FIRMA.IZGOT

(

ID NUMBER NOT NULL,

NAME VARCHAR2(200 BYTE) NOT NULL

)

## 2.2. Создание индексов

Для повышения эффективности работы с данными необходимо создать индексы для наиболее употребительных полей таблиц. Создадим индексы для наиболее используемых запросов:

CREATE UNIQUE INDEX FIRMA.DOLG\_PK ON FIRMA.DOLG

(ID)

CREATE UNIQUE INDEX FIRMA.EDIZM\_PK ON FIRMA.EDIZM

(ID)

CREATE UNIQUE INDEX FIRMA.FIRMA9\_PK ON FIRMA.FIRMA9

(NAME)

CREATE UNIQUE INDEX FIRMA.IZGOT\_PK ON FIRMA.IZGOT

(ID)

CREATE UNIQUE INDEX FIRMA.POSTAV\_PK ON FIRMA.POSTAV

(ID)

CREATE UNIQUE INDEX FIRMA.SKLAD\_PK ON FIRMA.SKLAD

(ID)

CREATE UNIQUE INDEX FIRMA.SOTRUD\_PK ON FIRMA.SOTRUD

(ID)

CREATE UNIQUE INDEX FIRMA.TOVAR\_PK ON FIRMA.TOVAR

(ID)

CREATE UNIQUE INDEX FIRMA.TRANSPORT\_PK ON FIRMA.TRANSPORT

(ID)

CREATE UNIQUE INDEX FIRMA.ZAKAZ\_PK ON FIRMA.ZAKAZ

(ID)

CREATE UNIQUE INDEX FIRMA.CLIENT\_PK ON FIRMA.CLIENT

(ID)

CREATE UNIQUE INDEX FIRMA.NAKLAD\_PK ON FIRMA.NAKLAD

(ID)

## 2.3 Ограничения целостности

Для того чтобы наложить ограничения на определенные поля таблиц и установить связи между таблицами по внешнему ключу, модернизируем таблицы, следующим образом.

ALTER TABLE FIRMA.DOLG ADD (

CONSTRAINT DOLG\_PK

PRIMARY KEY

(ID)

USING INDEX FIRMA.DOLG\_PK

ENABLE VALIDATE);

ALTER TABLE FIRMA.EDIZM ADD (

CONSTRAINT EDIZM\_PK

PRIMARY KEY

(ID)

USING INDEX FIRMA.EDIZM\_PK

ENABLE VALIDATE);

ALTER TABLE FIRMA.FIRMA9 ADD (

CONSTRAINT FIRMA9\_PK

PRIMARY KEY

(NAME)

USING INDEX FIRMA.FIRMA9\_PK

ENABLE VALIDATE);

ALTER TABLE FIRMA.IZGOT ADD (

CONSTRAINT IZGOT\_PK

PRIMARY KEY

(ID)

USING INDEX FIRMA.IZGOT\_PK

ENABLE VALIDATE);

ALTER TABLE FIRMA.POSTAV ADD (

CONSTRAINT POSTAV\_PK

PRIMARY KEY

(ID)

USING INDEX FIRMA.POSTAV\_PK

ENABLE VALIDATE);

ALTER TABLE FIRMA.SCHET ADD (

PRIMARY KEY

(ID)

USING INDEX

ENABLE VALIDATE);

ALTER TABLE FIRMA.SKLAD ADD (

CONSTRAINT SKLAD\_PK

PRIMARY KEY

(ID)

USING INDEX FIRMA.SKLAD\_PK

ENABLE VALIDATE);

ALTER TABLE FIRMA.SOTRUD ADD (

CONSTRAINT SOTRUD\_PK

PRIMARY KEY

(ID)

USING INDEX FIRMA.SOTRUD\_PK

ENABLE VALIDATE);

ALTER TABLE FIRMA.TOVAR ADD (

CONSTRAINT CBN1

CHECK (cena>0)

ENABLE VALIDATE

, CONSTRAINT TOVAR\_PK

PRIMARY KEY

(ID)

USING INDEX FIRMA.TOVAR\_PK

ENABLE VALIDATE);

ALTER TABLE FIRMA.TRANSPORT ADD (

CONSTRAINT GBN1

CHECK (gruz>0)

ENABLE VALIDATE

, CONSTRAINT GBN2

CHECK (god>0)

ENABLE VALIDATE

, CONSTRAINT TRANSPORT\_PK

PRIMARY KEY

(ID)

USING INDEX FIRMA.TRANSPORT\_PK

ENABLE VALIDATE);

ALTER TABLE FIRMA.ZAKAZ ADD (

CONSTRAINT KBN2

CHECK (kolvo>0)

ENABLE VALIDATE

, CONSTRAINT ZAKAZ\_PK

PRIMARY KEY

(ID)

USING INDEX FIRMA.ZAKAZ\_PK

ENABLE VALIDATE);

ALTER TABLE FIRMA.CLIENT ADD (

CONSTRAINT CLIENT\_PK

PRIMARY KEY

(ID)

USING INDEX FIRMA.CLIENT\_PK

ENABLE VALIDATE);

ALTER TABLE FIRMA.NAKLAD ADD (

CONSTRAINT KBN1

CHECK (kolvo>0)

ENABLE VALIDATE

, CONSTRAINT SBN1

CHECK (summa>0)

ENABLE VALIDATE

, CONSTRAINT NAKLAD\_PK

PRIMARY KEY

(ID)

USING INDEX FIRMA.NAKLAD\_PK

ENABLE VALIDATE);

ALTER TABLE FIRMA.FIRMA9 ADD (

CONSTRAINT FIRMFKSCHET

FOREIGN KEY (IDSCHET)

REFERENCES FIRMA.SCHET (ID)

ENABLE VALIDATE);

ALTER TABLE FIRMA.POSTAV ADD (

FOREIGN KEY (IDSCHET)

REFERENCES FIRMA.SCHET (ID)

ENABLE VALIDATE);

ALTER TABLE FIRMA.SOTRUD ADD (

CONSTRAINT SOTRUDFKDOLG

FOREIGN KEY (DOLG)

REFERENCES FIRMA.DOLG (ID)

ENABLE VALIDATE);

ALTER TABLE FIRMA.TOVAR ADD (

FOREIGN KEY (IDIZGOT)

REFERENCES FIRMA.IZGOT (ID)

ENABLE VALIDATE

, FOREIGN KEY (IDPOSTAV)

REFERENCES FIRMA.POSTAV (ID)

ENABLE VALIDATE

, FOREIGN KEY (NSKLAD)

REFERENCES FIRMA.SKLAD (ID)

ENABLE VALIDATE

, CONSTRAINT TOVARFKEDIZM

FOREIGN KEY (IDEDIZM)

REFERENCES FIRMA.EDIZM (ID)

ENABLE VALIDATE);

ALTER TABLE FIRMA.TRANSPORT ADD (

FOREIGN KEY (NSKLAD)

REFERENCES FIRMA.SKLAD (ID)

ENABLE VALIDATE);

ALTER TABLE FIRMA.ZAKAZ ADD (

FOREIGN KEY (IDCLIENT)

REFERENCES FIRMA.CLIENT (ID)

ENABLE VALIDATE

, FOREIGN KEY (IDTOVAR)

REFERENCES FIRMA.TOVAR (ID)

ENABLE VALIDATE);

ALTER TABLE FIRMA.CLIENT ADD (

CONSTRAINT CLIENTFKSCHET

FOREIGN KEY (IDSCHET)

REFERENCES FIRMA.SCHET (ID)

ENABLE VALIDATE);

ALTER TABLE FIRMA.NAKLAD ADD (

FOREIGN KEY (IDCLIENT)

REFERENCES FIRMA.CLIENT (ID)

ENABLE VALIDATE

, FOREIGN KEY (IDTOVAR)

REFERENCES FIRMA.TOVAR (ID)

ENABLE VALIDATE);

## 2.4 Создание триггеров

Для обеспечения автоматической нумерации полей таблиц для обеспечения их уникальности, или, как например в таблице накладных, ведения нумерации необходимо использовать небольшие процедуры выполняющиеся при каждой операции вставки строки в таблицу – триггеры. Для обеспечения нумерации в БД предусмотрены последовательности, из которых триггер извлекает следующий номер для поля, поэтому они, как правило создаются парами – последовательность и триггер.

CREATE SEQUENCE FIRMA.CLIENT\_SEQ

START WITH 1

MAXVALUE 999999999999999999999999999

MINVALUE 1

NOCYCLE

CACHE 20

NOORDER;

CREATE SEQUENCE FIRMA.DOLG\_SEQ

START WITH 1

MAXVALUE 999999999999999999999999999

MINVALUE 1

NOCYCLE

CACHE 20

NOORDER;

CREATE OR REPLACE TRIGGER FIRMA.DOLG\_TRG

BEFORE INSERT

ON FIRMA.DOLG

REFERENCING NEW AS New OLD AS Old

FOR EACH ROW

BEGIN

:new.ID := DOLG\_SEQ.nextval;

END DOLG\_TRG;

CREATE SEQUENCE FIRMA.EDIZM\_SEQ

START WITH 21

MAXVALUE 999999999999999999999999999

MINVALUE 1

NOCYCLE

CACHE 20

NOORDER;

CREATE OR REPLACE TRIGGER FIRMA.EDIZM\_TRG

BEFORE INSERT

ON FIRMA.EDIZM

REFERENCING NEW AS New OLD AS Old

FOR EACH ROW

BEGIN

:new.ID := EDIZM\_SEQ.nextval;

END EDIZM\_TRG;

CREATE SEQUENCE FIRMA.IZGOT\_SEQ

START WITH 1

MAXVALUE 999999999999999999999999999

MINVALUE 1

NOCYCLE

CACHE 20

NOORDER;

CREATE OR REPLACE TRIGGER FIRMA.IZGOT\_TRG

BEFORE INSERT

ON FIRMA.IZGOT

REFERENCING NEW AS New OLD AS Old

FOR EACH ROW

BEGIN

:new.ID := IZGOT\_SEQ.nextval;

END IZGOT\_TRG;

CREATE SEQUENCE FIRMA.NAKLAD\_SEQ

START WITH 21

MAXVALUE 999999999999999999999999999

MINVALUE 1

NOCYCLE

CACHE 20

NOORDER;

CREATE SEQUENCE FIRMA.POSTAV\_SEQ

START WITH 1

MAXVALUE 999999999999999999999999999

MINVALUE 1

NOCYCLE

CACHE 20

NOORDER;

CREATE OR REPLACE TRIGGER FIRMA.POSTAV\_TRG

BEFORE INSERT

ON FIRMA.POSTAV

REFERENCING NEW AS New OLD AS Old

FOR EACH ROW

BEGIN

:new.ID := POSTAV\_SEQ.nextval;

END POSTAV\_TRG;

CREATE SEQUENCE FIRMA.SCHET\_SEQ

START WITH 1

MAXVALUE 999999999999999999999999999

MINVALUE 1

NOCYCLE

CACHE 20

NOORDER;

CREATE OR REPLACE TRIGGER FIRMA.SCHET\_TRG

BEFORE INSERT

ON FIRMA.SCHET

REFERENCING NEW AS New OLD AS Old

FOR EACH ROW

BEGIN

:new.ID := SCHET\_SEQ.nextval;

END SCHET\_TRG;

CREATE SEQUENCE FIRMA.SKLAD\_SEQ

START WITH 21

MAXVALUE 999999999999999999999999999

MINVALUE 1

NOCYCLE

CACHE 20

NOORDER;

CREATE OR REPLACE TRIGGER FIRMA.SKLAD\_TRG

BEFORE INSERT

ON FIRMA.SKLAD

REFERENCING NEW AS New OLD AS Old

FOR EACH ROW

BEGIN

:new.ID := SKLAD\_SEQ.nextval;

END SKLAD\_TRG;

CREATE SEQUENCE FIRMA.SOTRUD\_SEQ

START WITH 1

MAXVALUE 999999999999999999999999999

MINVALUE 1

NOCYCLE

CACHE 20

NOORDER;

CREATE OR REPLACE TRIGGER FIRMA.SOTRUD\_TRG

BEFORE INSERT

ON FIRMA.SOTRUD

REFERENCING NEW AS New OLD AS Old

FOR EACH ROW

BEGIN

:new.ID := SOTRUD\_SEQ.nextval;

END SOTRUD\_TRG;

CREATE SEQUENCE FIRMA.TOVAR\_SEQ

START WITH 21

MAXVALUE 999999999999999999999999999

MINVALUE 1

NOCYCLE

CACHE 20

NOORDER;

CREATE OR REPLACE TRIGGER FIRMA.TOVAR\_TRG

BEFORE INSERT

ON FIRMA.TOVAR

REFERENCING NEW AS New OLD AS Old

FOR EACH ROW

BEGIN

:new.ID := TOVAR\_SEQ.nextval;

END TOVAR\_TRG;

CREATE SEQUENCE FIRMA.TRANSPORT\_SEQ

START WITH 1

MAXVALUE 999999999999999999999999999

MINVALUE 1

NOCYCLE

CACHE 20

NOORDER;

CREATE OR REPLACE TRIGGER FIRMA.TRANSPORT\_TRG

BEFORE INSERT

ON FIRMA.TRANSPORT

REFERENCING NEW AS New OLD AS Old

FOR EACH ROW

BEGIN

:new.ID := TRANSPORT\_SEQ.nextval;

END TRANSPORT\_TRG;

CREATE SEQUENCE FIRMA.ZAKAZ2\_SEQ

START WITH 1

MAXVALUE 999999999999999999999999999

MINVALUE 1

NOCYCLE

CACHE 20

NOORDER;

CREATE OR REPLACE TRIGGER FIRMA.ZAKAZ\_TRG

BEFORE INSERT

ON FIRMA.ZAKAZ

REFERENCING NEW AS New OLD AS Old

FOR EACH ROW

BEGIN

:new.ID := ZAKAZ2\_SEQ.nextval;

END ZAKAZ\_TRG;

CREATE OR REPLACE TRIGGER FIRMA.CLIENT\_TRG

BEFORE INSERT

ON FIRMA.CLIENT

REFERENCING NEW AS New OLD AS Old

FOR EACH ROW

BEGIN

:new.ID := CLIENT\_SEQ.nextval;

END CLIENT\_TRG;

CREATE OR REPLACE TRIGGER FIRMA.NAKLAD\_TRG

BEFORE INSERT

ON FIRMA.NAKLAD

REFERENCING NEW AS New OLD AS Old

FOR EACH ROW

BEGIN

:new.ID := NAKLAD\_SEQ.nextval;

END NAKLAD\_TRG;

## 2.5 Создание запросов

### Задание а.

Кто из владельцев счетов данного банка имеет кредиты на данную 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

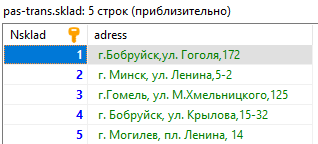
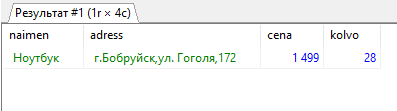


Рисунок 2.3

Результат выполнения запроса



### Задание b

Ь. Определить юридический адрес и номера телефонов данной фирмы.

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



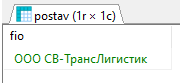
Рисунок 2.5

Таблица товаров



Рисунок 2.6

Результат выполнения запроса



### Задание c.

с. Кто из сотрудников данной фирмы был принят на работу на данную дату?

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

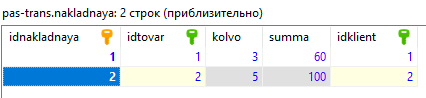


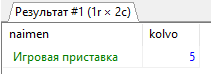
Рисунок 2.8

Таблица товаров



Рисунок 2.9

Результат выполнения запроса



### Задание d.

d. Кто из поставщиков имеет кредит на сумму превышающую заданную?

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



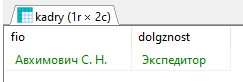
Рисунок 2.11

Таблица сотрудников



Рисунок 2.12

Результат выполнения запроса



### Задание e.

е. Найти юридические адреса клиентов, которые оформили заказы на данную дату.

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

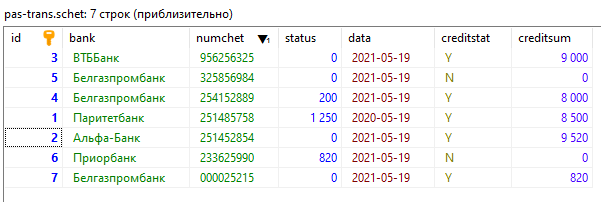


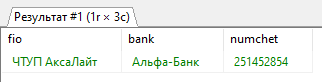
Рисунок 2.14

Таблица поставщиков



Рисунок 2.15

Результат выполнения запроса

****

## 2.6 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 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`)))

# 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