ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc74133744)

[ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ 4](#_Toc74133745)

[1.1 Техническое задание 4](#_Toc74133746)

[1.2 Анализ предметной области 4](#_Toc74133747)

[ГЛАВА 2 ПРОЕКТИРОВАНИЕ СИСТЕМЫ 6](#_Toc74133748)

[2.1 Построение логической структуры 6](#_Toc74133749)

[2.2 Преобразование ER-диаграммы в схему базы данных 7](#_Toc74133750)

[2.3 Нормализация отношений. 12](#_Toc74133751)

[ГЛАВА 3 РЕАЛИЗАЦИЯ СИСТЕМЫ В СУБД 20](#_Toc74133752)

[3.1 Создание объектов БД 20](#_Toc74133753)

[3.2. Создание индексов 23](#_Toc74133754)

[3.3 Ограничения целостности 24](#_Toc74133755)

[3.4 Создание триггеров 28](#_Toc74133756)

[3.5 Создание запросов 31](#_Toc74133757)

[Задание а 31](#_Toc74133758)

[Задание b 33](#_Toc74133759)

[Задание c 33](#_Toc74133760)

[Задание d 34](#_Toc74133761)

[Задание e 35](#_Toc74133762)

[3.6 Создание представлений 36](#_Toc74133763)

[Задание a 36](#_Toc74133764)

[Задание b 37](#_Toc74133765)

[Задание c 37](#_Toc74133766)

[Задание d 37](#_Toc74133767)

[Задание e 37](#_Toc74133768)

[ЗАКЛЮЧЕНИЕ 39](#_Toc74133769)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 40](#_Toc74133770)

# ВВЕДЕНИЕ

Актуальность работы определяется тем, что с ростом вычислительных мощностей, растет и объём производимой человечеством информации. Для обработки таких больших объемов были разработаны специализированные программные средства, ускоряющие и упрощающие работу. Умение пользоваться этими средствами в настоящее время является необходимостью, а не прихотью. Именно поэтому умение систематизировать и обрабатывать большие массивы данных востребовано в современном мире.

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

Практическая значимость курсового проекта заключается в том, что данная информационная система позволит структурировать большие объемы информации, и тем самым существенно облегчит работу и доступ к данным сотрудникам фирмы.

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

Задачи курсового проекта:

- изучить и проанализировать соответствующую литературу по теме работы;

- провести анализ предметной области;

- разработать логической структуры базы данных, и базу данных на физическом уровне;

- составить запросы и представления;

- протестировать выполнение запросов, выявить все допущенные ошибки и устранить их.

Объект данной работы – автоматизированная информационная система.

Предмет проектирования – информационная система управления торговой фирмой.

При реализации курсового проекта будут использованы:

- язык структурированных запросов – SQL, PL/SQL;

- Oracle 11.2;

# ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ

## 1.1 Техническое задание

Перечень документов, на основании которых создаётся информационный ресурс:

* задание на курсовую работу;
* правила оформления курсовой работы;
* методическое пособие к курсовому проектированию;
* нормативно правовые акты (в рамках законодательства РБ).

Информационный ресурс представляет собой функционирующего комплекс программных средств. Вместе с программным продуктом предоставляется в печатной форме сопроводительная документация.

Систему предполагается создать для улучшения качества обслуживания клиентов торговой фирмы, позволит ускорить процессы обработки информации, повысить надежность её хранения.

## 1.2 Анализ предметной области

В качестве предметной области проектирования БД выбрана торговая фирма, которая осуществляет выполнение заказов, поступающих от клиентов. Целью работы является построение логической структуры базы данных на уровне взаимосвязей между объектами и на уровне взаимосвязей между атрибутами разных объектов, предварительно определив эти взаимосвязи; подтвердить правильность полученной структуры на экземплярах данных; выполнить нормализацию отношений: ввести ограничения, определить функциональные зависимости и ключи; привести каждое отношение к третьей нормальной форме; сформулировать данные запросы в терминах реляционной алгебры и показать правильность их обработки на примерах.

Основным назначением информационных ресурсов является оперативное обеспечение пользователя информацией путем реализации взаимодействия пользователя с системой, через запросы и ответы на них со стороны информационной системы. Информация представляется в информационном ресурсе в форме данных. Совокупность этих выделенных для информационных ресурсов данных, связей между ними и операций над ними образует информационную и функциональную модели предметной области, описывающие ее состояние с определенной точностью.

В результате проделанной работы были изучены нормативно-правовые акты, перечень необходимых для проектирования документов, произведен анализ предметной области, оценка актуальности поставленной задачи, определена и уточнена цель поставленной задачи.

# ГЛАВА 2 ПРОЕКТИРОВАНИЕ СИСТЕМЫ

## 2.1 Построение логической структуры

Для построения БД использованы следующие объекты и их атрибуты:

Фирма (название фирмы, юр. адрес, телефон фирмы, №счета в банке, наименование товара, цена товара);

Склад (№склада, адрес склада, №заказа, наименование товара, единицы измерения, кол-во товара, цена товара, отметки об оплате);

Кадры (Название фирмы, Ф.И.О. сотрудника, должность сотрудника, образование сотрудника, оклад, дата принятия на работу);

Поставщик (№поставщика, Ф.И.О. поставщика, юр. адрес поставщика, №счета поставщика, наименование товара, цена товара);

Транспорт (№склада, №автомобиля, марка автомобиля, грузоподъемность, год выпуска, Ф.И.О. владельца, дата перевозки);

Накладная (№накладной, наименование товара, единицы измерения, кол-во, цена, сумма);

Товар (Наименование товара, единицы измерения, сорт, цена, поставщик, адрес поставщика);

Клиент (№клиента, Ф.И.О. клиента, юр. адрес клиента, №счета клиента, №заказа, наименование товара, цена товара); Форма 3

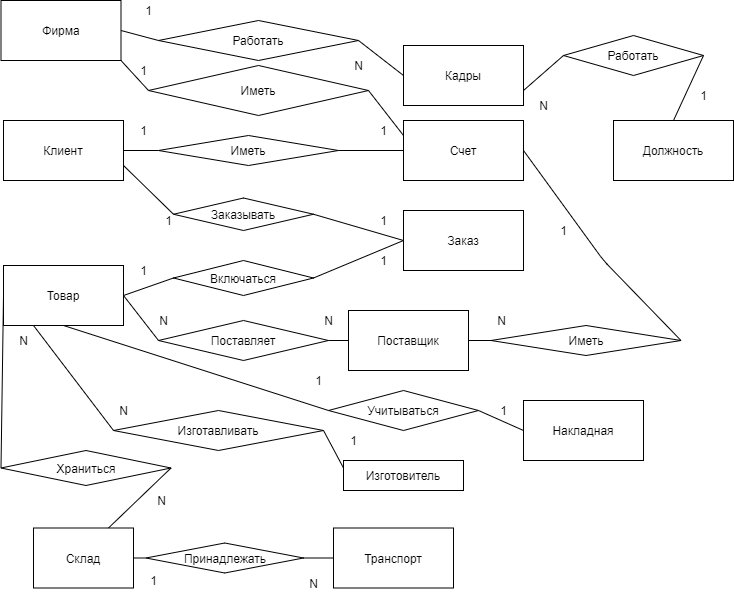
Счет (Наименование банка, №счета, Ф.И.О. владельца счета, состояние счета, дата, наличие кредита, сумма кредита); Форма 3

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

Исходя из выявленных сущностей, построим ER–диаграмму (рисунок 2.1.1)

Рисунок 2.1.1

ER-диаграмма фирмы.



## 2.2 Преобразование ER-диаграммы в схему базы данных

Рассмотрим атрибуты данных нам объектов.

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, отношение представлено в таблице 2.2.1.

Таблица 2.2.1

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, отношение представлено в таблице 2.2.2.

Таблица 2.2.2

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, отношение представлено в таблице 2.2.3.

Таблица 2.2.3

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, отношение представлено в таблице 2.2.4.

Таблица 2.2.4

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 |

Таблица 2.2.4 продолжение

|  |  |  |  |
| --- | --- | --- | --- |
| 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, отношение представлено в таблице 2.2.5.

Таблица 2.2.5

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, отношение представлено в таблице 2.2.6.

Таблица 2.2.6

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, отношение представлено в таблице 2.2.7.

Таблица 2.2.7

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 |

Таблица 2.2.7 продолжение

|  |  |  |  |
| --- | --- | --- | --- |
| 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, отношение представлено в таблице 2.2.8.

Таблица 2.2.8

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, отношение представлено в таблице 2.2.9.

Таблица 2.2.9

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, отношение представлено в таблице 2.2.10.

Таблица 2.2.10

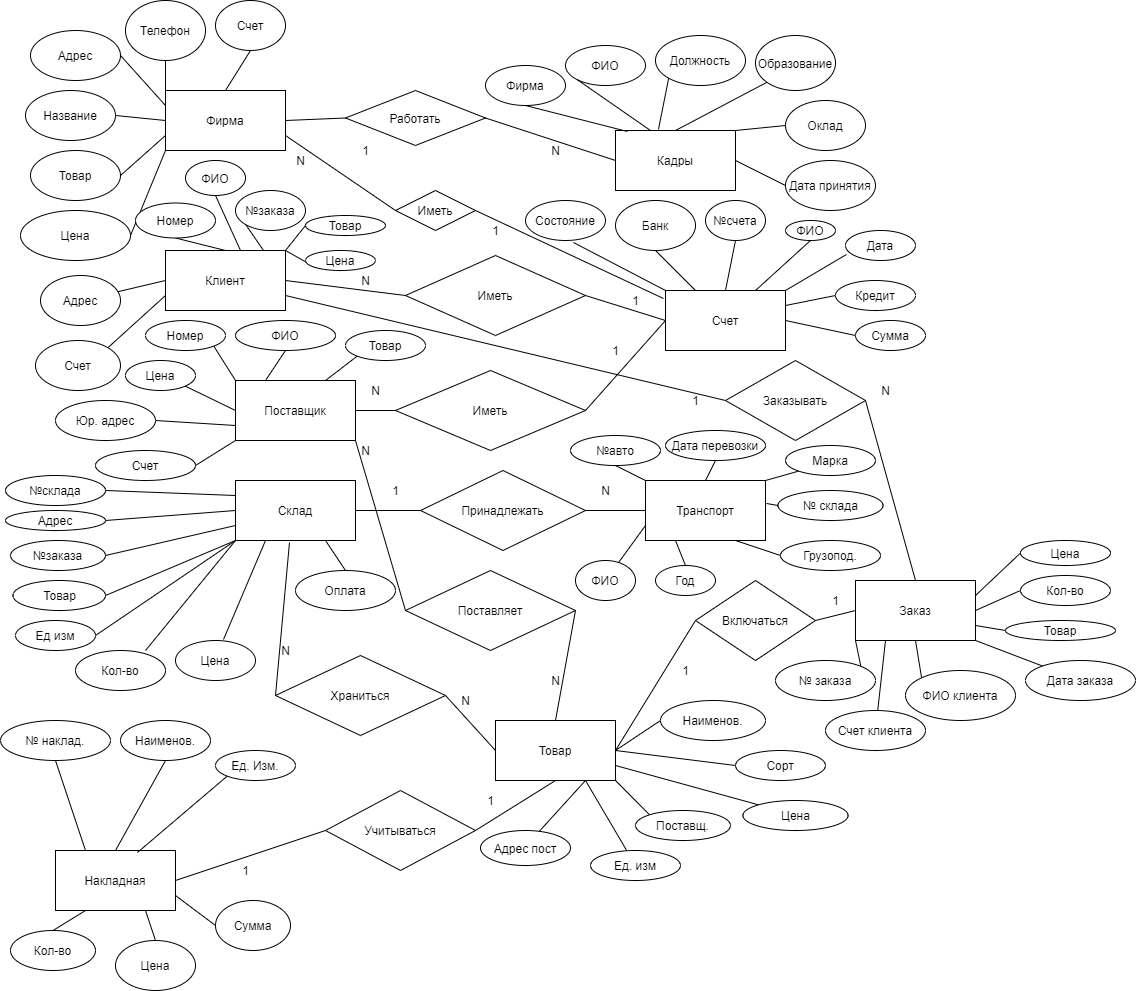
Схема отношения ЗАКАЗ

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |

Исходя из полученных данных построим уточненную ER-диаграмму с указанием всех данных по условию полей (рисунок 2.2.1).

Рисунок 2.2.1

Уточненная ER-диаграмма фирмы.



## 2.3 Нормализация отношений.

Нормальная форма — требование, предъявляемое к структуре таблиц в теории реляционных баз данных для устранения из базы избыточных функциональных зависимостей между атрибутами (полями таблиц) [2].

Метод нормальных форм (НФ) состоит в сборе информации о объектах решения задачи в рамках одного отношения и последующей декомпозиции этого отношения на несколько взаимосвязанных отношений на основе процедур нормализации отношений.

Цель нормализации: исключить избыточное дублирование данных, которое является причиной аномалий, возникших при добавлении, редактировании и удалении кортежей (строк таблицы). Аномалией называется такая ситуация в таблице БД, которая приводит к противоречию в БД либо существенно усложняет обработку БД. Причиной является излишнее дублирование данных в таблице, которое вызывается наличием функциональных зависимостей от не ключевых атрибутов. Всего выделяют три типа аномалий.

**Аномалии-модификации** проявляются в том, что изменение одних данных может повлечь просмотр всей таблицы и соответствующее изменение некоторых записей таблицы.

**Аномалии-удаления** — при удалении какого-либо кортежа из таблицы может пропасть информация, которая не связана напрямую с удаляемой записью.

**Аномалии-добавления** возникают, когда информацию в таблицу нельзя поместить, пока она не полная, либо вставка записи требует дополнительного просмотра таблицы.

Переменная отношения находится в первой нормальной форме (1НФ) тогда и только тогда, когда в любом допустимом значении отношения каждый его кортеж содержит только одно значение для каждого из атрибутов.

В реляционной модели отношение всегда находится в первой нормальной форме по определению понятия отношение. Что же касается различных таблиц, то они могут не быть правильными представлениями отношений и, соответственно, могут не находиться в 1НФ.

Переменная отношения находится во второй нормальной форме тогда и только тогда, когда она находится в первой нормальной форме и каждый неключевой атрибут неприводимо (функционально полно) зависит от её потенциального ключа. Функционально полная зависимость означает, что если потенциальный ключ является составным, то атрибут зависит от всего ключа и не зависит от его частей.

Переменная отношения находится в третьей нормальной форме тогда и только тогда, когда она находится во второй нормальной форме, и отсутствуют транзитивные функциональные зависимости неключевых атрибутов от ключевых атрибутов [6].

Рассмотрим наши отношения и связи между ними с точки зрения нормализации.

ФИРМА, ПОСТАВЩИК, КЛИЕНТ имеют счета в банке, если, например, поставщик решит изменить свои реквизиты (номер счета или банк) то возникнет аномалия модификации – нам придётся просматривать таблицу в поисках этого поставщика и менять во всех вхождениях его счет. Аналогично для клиентов и самой фирмы. В этих случаях имеет смысл выделить счет в отдельное отношение и связать его с первичной таблицей, тогда в случае модификации счета его необходимо будет изменить только в одном месте. В отношении счет есть атрибут ФИО, если его оставить то аномалия останется в ином виде: ФИО будет храниться и в отношении СЧЕТ и в отношениях КЛИЕНТ (ПОСТАВЩИК, ФИРМА), получается кольцевая связь, чтобы избавиться от неё достаточно отказаться от атрибута ФИО в отношении СЧЕТ, при связи 1:1 оно будет однозначно идентифицироваться [7].

Модернизированное отношение представлено в таблице 2.3.1.

Таблица 2.3.1

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 также №Заказа, отношение представлено в таблицах 2.3.2 и 2.3.3.

Таблица 2.3.2

Схема отношения ПОСТАВЩИК

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

Таблица 2.3.2. продолжение

|  |  |  |  |
| --- | --- | --- | --- |
| 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 2.3.3

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, отношение представлено в таблице 2.3.4.

Таблица 2.3.4

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, отношение представлено в таблице 2.3.5.

Таблица 2.3.5

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, отношение представлено в таблице 2.3.6.

Таблица 2.3.6

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, отношение представлено в таблице 2.3.7.

Таблица 2.3.7

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 же удалим реквизиты счета, заменим его связью со СЧЕТ, отношение представлено в таблице 2.3.8.

Таблица 2.3.8

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, отношение представлено в таблице 2.3.9.

Таблица 2.3.9

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, отношение представлено в таблице 2.3.10.

Таблица 2.3.10

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, отношение представлено в таблице 2.3.11.

Таблица 2.3.11

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, отношение представлено в таблице 2.3.12.

Таблица 2.3.12

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, отношение представлено в таблице 2.3.13.

Таблица 2.3.13

Схема отношения ЕДИНИЦЫ

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

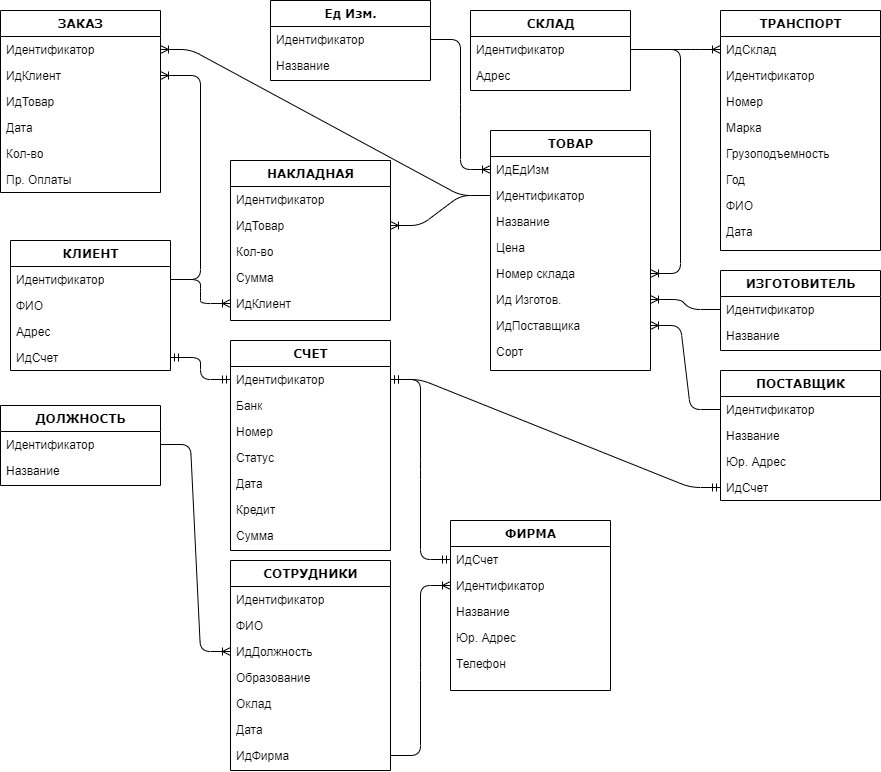
Атрибут Образование отношения КАДРЫ может принимать одно из следующих значений: ‘Высшее’, ’Среднее специальное’, ’Среднее’, ’Базовое’.

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 2.3.1) Составим ER-диаграмму в нотации Мартина, или, как её ещё называют «Вороньи лапки». В нотации Мартина: сущность изображается прямоугольником, внутри которого указано ее имя жирным шрифтом и список ее атрибутов, а связь – линией, название которой располагается над ней и ее вид в месте соединения с сущностью определяет кардинальность связи.

Рисунок 2.3.1

Окончательная схема БД

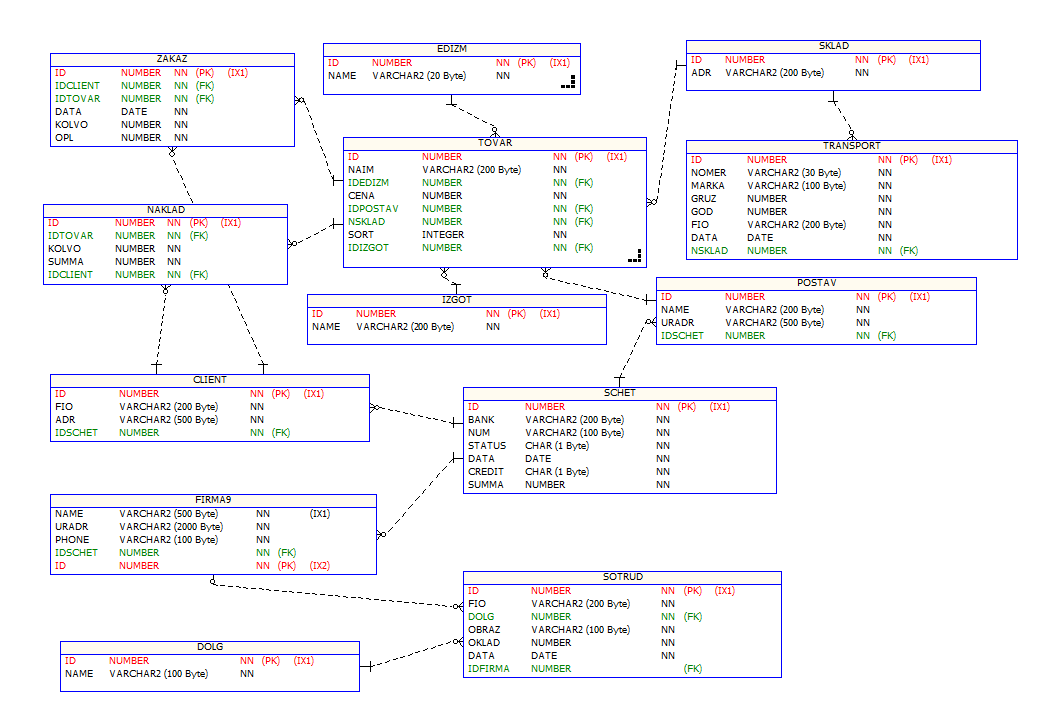


Атрибуты всех наших отношений являются простыми, и все используемые кортежи содержат только скалярные значения, следовательно, это 1 нормальная форма. Каждый неключевой атрибут наших отношений неприводимо зависит от Первичного Ключа (ПК), следовательно, отношения находится во 2НФ. Отношения находятся в 3НФ, так как каждый не ключевой атрибут нетранзитивно зависит от первичного ключа, потому что мы вынесли все неключевой поля в отдельные таблицы.

После того как мы привели таблицы базы данных к третьей нормальной форме, мы можем переходить к созданию таблиц в физической базе данных. Диаграмма, физически представляющая базу данных Oracle, представлена на рисунке 2.3.4

Рисунок 2.3.4

ER-диаграмма физической структуры базы данных



В процессе проектирования системы была построена логическая модель базы данных, составлены диаграммы отражающие отношения и связи между ними, проведена нормализация до 3 нормальной формы. Итогом проделанной работы является подготовленная ER-диаграмма для создания базы данных на физическом уровне, определены будущие таблицы, их поля, тип полей, связи между таблицами.

# ГЛАВА 3 РЕАЛИЗАЦИЯ СИСТЕМЫ В СУБД

## 3.1 Создание объектов БД

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.

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 OT 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

##### )

## 3.2. 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)

## 3.3 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);

## 3.4 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. Некоторые триггеры в схеме выполняют и дополнительные функции для валидации или модификации вносимых данных, примером такого триггера может служить триггер на таблице накладных. 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 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 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 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 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;

В триггере таблицы СЧЕТ, сделаем обработку суммы, если она равна нулю то поле наличие кредита будет принимать значение ‘N’, а если сумма больше нуля, то ‘Y’.

CREATE OR REPLACE TRIGGER FIRMA.SCHET\_TRG

BEFORE INSERT

ON FIRMA.SCHET

REFERENCING NEW AS NEW OLD AS OLD

FOR EACH ROW

BEGIN

-- For Toad: Highlight column ID

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

if :new.summa>0

then

:new.credit:='Y';

else

:new.credit:='N';

end if;

END SCHET\_TRG;

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 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 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 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 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;

В триггере для накладных организуем заполнение поля СУММА как произведение количества на цену соответствующего товара. В этом случае мы исключим возможность ошибки или опечатки.

CREATE OR REPLACE TRIGGER NAKLAD\_TRG

BEFORE INSERT

ON NAKLAD

REFERENCING NEW AS NEW OLD AS OLD

FOR EACH ROW

declare

z number;

BEGIN

-- For Toad: Highlight column ID

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

select cena into z from tovar where id=:new.idtovar;

:new.summa:=z\*:new.kolvo;

END NAKLAD\_TRG;

## 3.5 Создание запросов

### Задание а

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

Для того чтобы выяснить необходимую информацию необходимо обработать таблицы поставщиков, клиентов и самой фирмы, которые имеют ссылки на таблицу счетов и уже из таблицы счетов выяснить кто из них имеет кредит [1, c.122].

Запишем запрос на языке DML:

SELECT name, data, summa

FROM schet s, firma9 f

WHERE f.idschet = s.id

AND s.data = '13/07/2006'

AND s.credit = 'Y'

UNION

SELECT name, data, summa

FROM schet s, postav f

WHERE f.idschet = s.id

AND s.data = '13/07/2006'

AND s.credit = 'Y'

UNION

SELECT fio name, data, summa

FROM schet s, client f

WHERE f.idschet = s.id

AND s.data = '13/07/2006'

AND s.credit = 'Y'

Результат выполнения запроса и содержимое таблиц представлены на рисунках 3.5.1-3.5.4.

Рисунок 3.5.1

Таблица клиентов



Рисунок 3.5.2

Таблица поставщиков



Рисунок 3.5.3

Таблица фирма



Рисунок 3.5.4

Результат выполнения запроса



### Задание b

Определить юридический адрес и номера телефонов данной фирмы.

Входные данные задания отсутствуют, так как фирма у нас всего одна. Для получения необходимой информации нам достаточно одной таблицы: таблицы фирмы.

Запишем запрос на языке DML:

select uradr, phone

from firma9

Результат выполнения запроса и содержимое таблиц представлены на рисунках 3.5.5-3.5.6.

Рисунок 3.5.5

Таблица Фирма



Рисунок 3.5.6

Результат выполнения запроса



### Задание c

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

В задании есть входные данные: дата приема сотрудника на работу, и данные, которые нам необходимо выяснить, это ФИО сотрудника. Для получения необходимой информации нам необходимо задействовать одну таблицу: таблицу сотрудников, в ней есть вся необходимая информация.

Запишем запрос на языке DML:

SELECT FIO

FROM sotrud

WHERE data = '21/07/2018'

Результат выполнения запроса и содержимое таблиц представлены на рисунках 3.5.7-3.5.8.

Рисунок 3.5.7

Таблица Сотрудников

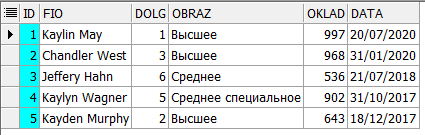


Рисунок 3.5.8

Результат выполнения запроса



### Задание d

Кто из поставщиков имеет кредит на сумму, превышающую заданную?

Входными данными для нас является заданная сумма кредита, выходными – поставщик, превысивший эту сумму по кредиту. Используются две таблицы -поставщиков и счетов.

Запишем запрос на языке DML:

SELECT p.name, s.summa

FROM postav p, schet s

WHERE s.id = p.idschet

AND s.Credit = 'Y'

AND s.summa > 379

Результат выполнения запроса и содержимое таблиц представлены на рисунках 3.5.9-3.5.11.

Рисунок 3.5.9

Таблица поставщиков



Рисунок 3.5.10

Таблица счетов

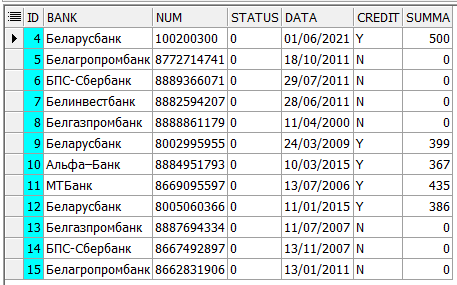
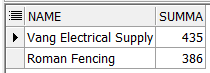


Рисунок 3.5.11

Результат выполнения запроса



### Задание e

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

В задании необходимо найти записи в таблице заказов, у которых совпадает с указанной дата заказа, а затем в таблице клиентов выяснить юридический адрес соответствующего клиента. Необходимо задействовать 2 таблицы – заказов и клиентов.

Запишем запрос на языке DML:

SELECT c.fio, c.adr

FROM client c, zakaz z

WHERE z.idclient = c.id AND z.data = '02/11/2020'

Результат выполнения запроса и содержимое таблиц представлены на рисунках 3.5.12-3.5.14.

Рисунок 3.5.12

Таблица заказов

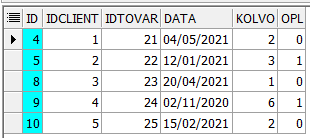


Рисунок 3.5.13

Таблица клиентов



Рисунок 3.5.14

Результат выполнения запроса

****

## 3.6 Создание представлений

Представление (VIEW) — объект базы данных, являющийся результатом выполнения запроса к базе данных, определенного с помощью оператора SELECT, в момент обращения к представлению.

Представления иногда называют «виртуальными таблицами». Такое название связано с тем, что представление доступно для пользователя как таблица, но само оно не содержит данных, а извлекает их из таблиц в момент обращения к нему. Если данные изменены в базовой таблице, то пользователь получит актуальные данные при обращении к представлению, использующему данную таблицу.

Представления могут основываться как на таблицах, так и на других представлениях, т.е. могут быть вложенными (до 32 уровней вложенности).

### Задание a

CREATE OR REPLACE VIEW FIRMA.V\_A

(N, D, S)

AS

SELECT name n, data d, summa s

FROM schet s, firma9 f

WHERE f.idschet = s.id AND s.data >= '13/07/2006' AND s.credit = 'Y'

UNION

SELECT name n, data d, summa s

FROM schet s, postav f

WHERE f.idschet = s.id AND s.data >= '13/07/2006' AND s.credit = 'Y'

UNION

SELECT fio n, data d, summa s

FROM schet s, client f

WHERE f.idschet = s.id AND s.data >= '13/07/2006' AND s.credit = 'Y';

### Задание b

CREATE OR REPLACE VIEW FIRMA.V\_B

(URADR, PHONE)

AS

SELECT uradr, phone FROM firma9;

### Задание c

CREATE OR REPLACE VIEW FIRMA.V\_C

(ID, FIO, DOLG, OBRAZ, OKLAD,

DATA)

AS

SELECT \*

FROM sotrud

WHERE data = '21/07/2018';

### Задание d

CREATE OR REPLACE VIEW FIRMA.V\_D

(NAME, SUMMA)

AS

SELECT p.name, s.summa

FROM postav p, schet s

WHERE s.id = p.idschet AND s.Credit = 'Y' AND s.summa > 379;

### Задание e

CREATE OR REPLACE FORCE VIEW FIRMA.V\_E

(

FIO,

ADR

)

AS

SELECT c.fio, c.adr

FROM client c, zakaz z

WHERE z.idclient = c.id AND z.data = '02/11/2020';

# ЗАКЛЮЧЕНИЕ

В результате проделанной работы была разработана и создана база данных "Фирма 9", которая значительно упрощает работу с данными и во многом экономит время сотрудников. Созданная база данных позволяет вести учет товаров на складах, выданных накладных, управление персоналом фирмы, контроль заказов, в нее могут быть введены дополнительные средства формирования данных.

В процессе работы была осуществлена постановка задачи, выбор методики проектирования, обследование и анализ требований предметной области, проектирование структуры базы данных и разработка полной АИС, для автоматизации работы пользователей. В ходе выполнения поставленной задачи отработаны навыки работы с базами данных, проведена работа со специальной технической и справочной литературой, самостоятельно приняты решения и оценено их качество. Освоено концептуальное проектирование и разработка инфологической модели БД в виде ER-диаграмм. Улучшены практические навыки проектирования баз данных на физическом уровне.

Достигнута цель курсового проекта, заключающаяся в систематизации знаний и накоплении первоначального опыта в области проектирования баз данных и автоматизированных рабочих мест в составе корпоративных информационных систем. Все необходимые задачи выполнены в полном объеме.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Грофф, Дж. Р.SQL : Полное руководство / Дж. Р. Грофф, П. Н. Вайнберг ; под ред. В. Р. Гинзбург. – 2-е изд., перераб. и доп. – Киев : BHV: Ирина, 2001. – 816 с
2. Другой пример проектирования базы данных (MySQL). URL: https://pro-prof.com/forums/topic/db\_example
3. Куликов С.С. Реляционные базы данных в примерах : практическое пособие для программистов и тестировщи- ков / С. С. Куликов. — Минск: Четыре четверти, 2020. — 424 с.
4. Шениг, Г.Ю. PostgresSQL 11. Мастерство разработки / пер. с англ. А.А. Слинкина. – Москва: ДМК Пресс, 2019. – 352 с.