МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГТУ», ВГТУ)

Факультет информационных технологий и компьютерной безопасности

Кафедра компьютерных интеллектуальных технологий проектирования

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

По дисциплине: «Системы хранения и обработки данных»

Тема: «Разработка логической структуры базы данных»

Выполнил работу студент группы мИИВТ-231: Никулин В.С.

подпись, дата

Принял: Короленко В.В.

подпись, дата

Воронеж 2023

**Цель работы:** изучить основы логического проектирования базы данных, освоить процесс разработки логической структуры базы данных и построения диаграммы «сущность-связь».

**Задачи работы:** необходимо выполнить все пункты учебной задачи:

1) Изучить основные понятия теории баз данных, основные модели данных.

2) Изучить средство для разработки диаграмм draw.io

3) Определить основные сущности для разрабатываемой базы данных  
(не менее 4) Обосновать выбор.

4) Определить атрибуты сущностей.

5) Выделить ключевые атрибуты сущностей (первичные и внешние ключи).  
Пояснить свой выбор.

6) Определить связи между сущностями. Как минимум одна связь должна быть «многие ко многим».

7) Пояснить выбор типов связей.

8) Избавиться от связей «многие ко многим» с помощью введения дополнительных сущностей.

9) Построить диаграмму сущность-связь для отображения логической структуры проектируемой базы данных с использованием средства для разработки диаграмм draw.io.

10) На диаграмме выделить ключевые атрибуты (PK, FK), обозначить связи (для каждой связи вставить соответствующий текст, в котором указано, что это за связь).

11) Описать диаграмму (текстом ниже диаграммы). Пояснить выбор сущностей, атрибутов, связей, ключей.

12) Подготовить отчёт о проделанной работе.

13) Отчётные материалы загрузить в репозиторий Git и отправить ссылку на ваш репозиторий на платформе github на почту преподавателю. Репозиторий должен быть публичным.

**Ход работы:**

Для выполнения лабораторной работы была выбрана тема №3 – «Учет оптовых продаж магазина». Далее, был изучен предлагаемый в задании набор основных сущностей для базы данных: Товары, Продажи и Продавцы. После этого начался процесс построения логической схемы БД в программе для построения диаграмм «draw.io».

Для начала, была рассмотрена сущность «Товары». Для данной сущности был определён первичный ключ «ID товара». Далее, были определены основные атрибуты данной сущности: «Наименование товара», Цена закупки, «Количество закупленного товара», «Дата закупки», «Наименование единицы измерения», «Наименование единицы измерения». При непосредственном построении схемы, атрибут «Наименование единицы измерения» данной сущности был вынесен в отдельную таблицу  
«Единицы измерения» в качестве первичного ключа, а в основной таблице данный атрибут представлен внешнем ключом «Наименование единицы измерения». Сделано это было для того, чтобы пользователь имел возможность заранее определить набор единиц измерения (которых не так уж и много) и ему не приходилось каждый раз при заполнении таблицы «Товары» вводить один и тот же текст единицы измерения. Он, например, смог бы выбрать единицу измерения из списка готовых, которые хранились бы в таблице «Единицы измерения». Также, между данными таблицами была определена связь «Один ко многим», так как один товар может обладать лишь одной единицей измерения (например, килограмм, а не сантиметр), в то время, как единицы измерения могут относится к десятками и сотням разных товаров. Может быть много товаров с единицей измерения сантиметр, например несколько разных видов скотча, а килограммы могут описывать, например несколько разных сортов мяса. По итогу были получены первые две сущности в логической схеме (рисунок 1).

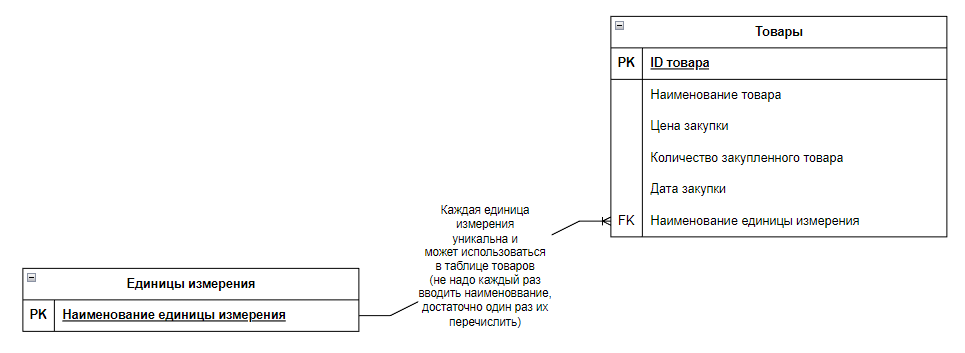


Рисунок 1 - Таблицы «Товары» и «Единицы измерения»  
и связь «Один ко многим» между ними

Далее, была без особых проблем спроектирована таблица «Продавцы». Первичным ключом выступает «ID продавца». А основные атрибуты продавца – это фамилия, имя, отчество, процент его комиссионных, телефон, email и его адрес.  
Собственно, рисунок 2 демонстрирует эту таблицу на диаграмме.

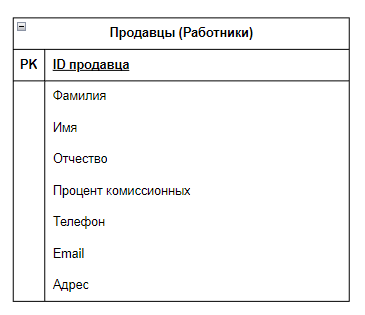


Рисунок 2 - Таблица «Продавцы»

Далее, между сущностями «Товары» и «Продавцы» необходимо было определить связь, и она здесь – «Многие ко многим», так как один продавец может продавать десятки разных товаров, в том время как один конкретный товар, может продаваться у нескольких разных продавцов по разной цене, на которую, в том числе, влияет и процент комиссионных конкретного продавца. Чтобы изобразить такую связь на логической схеме, было принято решение добавить дополнительную сущность  
(таблицу) «Товары\_Продавцы». В данной таблице будут храниться соответствия между товарами и продавцами, чтобы знать какой продавец какой товар продаёт, для этого необходимо определить в таблице атрибуты «ID товара» и «ID продавца», которые являются внешними ключами и связаны со своими таблицами связью «Один ко многим». Также, был определён первичный ключ – «Товары\_Продавцы», чтобы иметь возможность отличать каждое соответствие друг от друга. На рисунке 3, показаны таблицы «Товары», «Продавцы» и «Товары\_Продавцы» и взаимоотношения между ними.

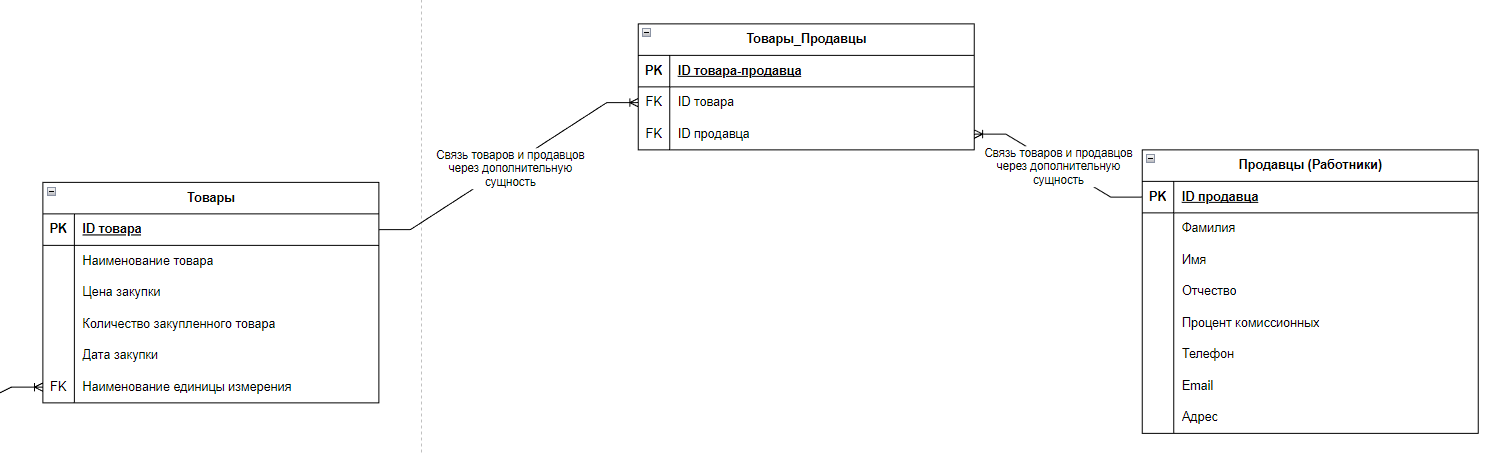


Рисунок 3 - Таблицы «Товары» и «Продавцы» с реализованной связью  
«Многие ко многим», с помощью дополнительной  
таблицы (сущности) «Товары\_Продавцы»

Далее, была определена последняя сущность «Продажи». Для неё был определён следующий набор атрибутов: «ID продажи» (первичный ключ), «цена продажи», «количество проданного товара», «дата продажи», и «ID товара-продавца» (внешний ключ). Таблица «Продажи» связана с таблицей «Товары\_Продавцы» посредством внешнего ключа «ID товара-продавца» связью «Один ко многим». Одна конкретная продажа может быть выполнена лишь у одного конкретного продавца по одному конкретному товару, в то время как соответствие «товар-продавец» может быть продано бесчисленное множество раз (сегодня продал Вася морковку, завтра и послезавтра). По итогу, получилась готовая логическая схема для темы №3 «Учет оптовых продаж магазина» (рисунок 4).

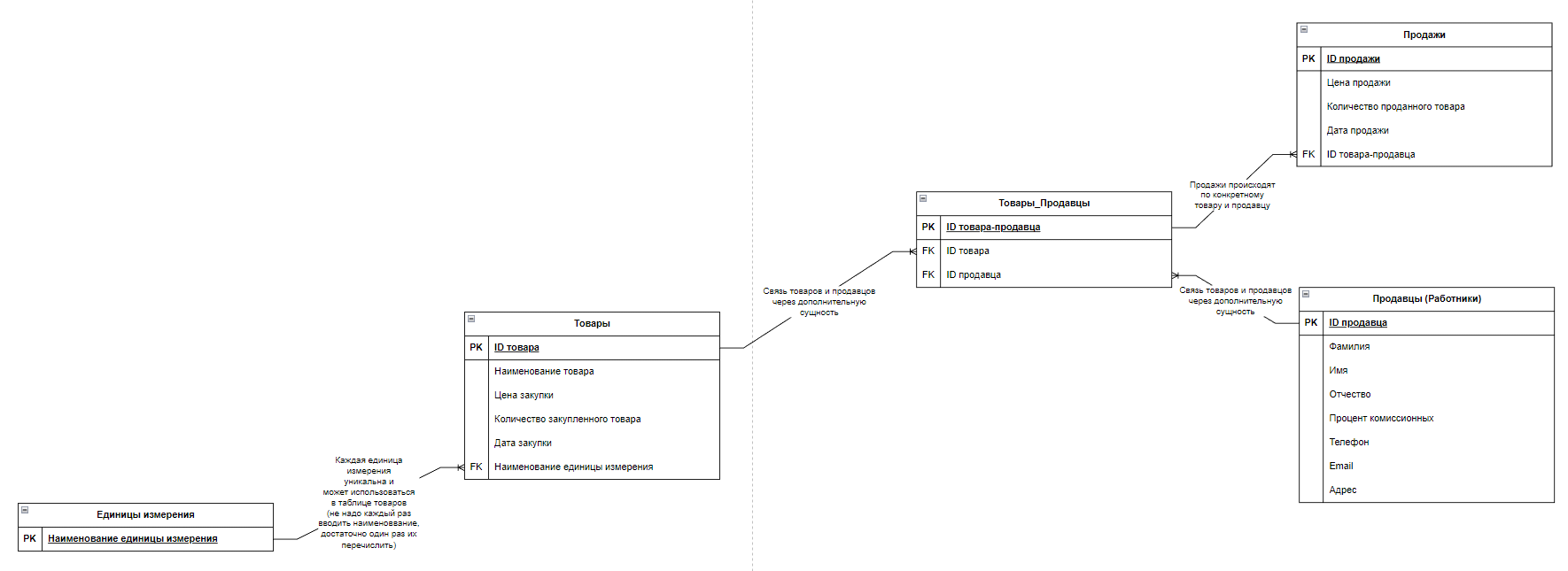


Рисунок 4 - Итоговый вариант логической схемы БД

**Ответы на контрольные вопросы:**

Вопрос 1: дайте определения следующим понятиям: данные, база данных, СУБД, ведение базы данных.

Ответ:

1) Данные – представление объектов реального мира и их свойств в формализованном виде, пригодном для хранения, передачи, интерпретации или обработки. В случае использования данных для уменьшения неопределенности знаний о каком-либо объекте данные превращаются в информацию.

2) База данных (БД) – совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, независимая от прикладных программ.

3) Система управления базами данных (СУБД) – совокупность программ и языковых средств, предназначенных для управления данными в базе данных, ведения базы данных и обеспечения взаимодействия ее с прикладными программами.

4) Ведение базы данных – деятельность по обновлению, восстановлению и перестройке структуры базы данных с целью обеспечения ее целостности, сохранности и эффективности использования.

Вопрос 2: в чем отличие данных от информации?

Ответ: данные – это фактические сведения или значения, которые могут быть в виде чисел, слов или символов. Информация – это данные, организованные и интерпретированные таким образом, чтобы они имели смысл и были полезными для принятия решений.

Вопрос 3: в чем отличие базы данных от банка данных и СУБД?

Ответ: база данных – это организованная коллекция данных, обычно хранящаяся в электронной форме. Банк данных – автоматизированная информационная  
система централизованного хранения и коллективного использования данных.  
В состав банка данных входят одна или несколько баз данных, справочник баз данных, СУБД, а также библиотеки запросов и прикладных программ. Система управления базами данных (СУБД) – это программное обеспечение, предназначенное для создания, управления и обработки баз данных, обеспечивая эффективное управление информацией.

Вопрос 4: назовите основные компоненты банка данных и их назначение.

Ответ:

1) База данных (БД): хранит сами данные в организованной структуре для эффективного доступа и управления информацией.

2) Система управления базами данных (СУБД): обеспечивает управление и координацию доступа к данным, управление транзакциями и обеспечение целостности базы данных.

3) Справочник баз данных: содержит метаданные, описывающие структуру и связи в базе данных, что облегчает понимание и использование данных.

4) Библиотеки запросов и прикладных программ: обеспечивают средства для создания и выполнения запросов к данным, а также интеграцию базы данных с прикладными программами для обработки информации.

Вопрос 5: классифицируйте АИС по типу хранимых данных.

Ответ: информационные системы можно классифицировать по ряду признаков:

1. по типу хранимых данных:
   1. документальные информационно-поисковые системы (ДИПС) предназначены для хранения и обработки документальных данных – адресов хранения документов, наименований, описаний и рефератов, а также текстов документов;
   2. Фактографические информационно-поисковые системы (ФИПС) хранят и обрабатывают фактографическую информацию – структурированные данные в виде чисел и текстов.
2. по характеру обработки данных:
   1. информационно-справочные системы (ИСС), выполняют поиск и вывод информации баз ее обработки;
   2. автоматизированные информационные системы обработки данных (ИСОД), сочетают в себе информационно-справочную систему с системой обработки данных.
3. по степени интеграции данных и автоматизации управления:
   1. АИС на автономных файлах (АИС АФ) – такие системы отличаются простой архитектурой и ограниченным кругом возможностей. Состоят такие системы из набора автономных файлов и комплекса прикладных программ, предназначенных для обработки этих файлов и выдачи документов;
   2. банки данных (БнД). Это системы с высокой степенью интеграции данных и автоматизации управления ими. Они ориентированы на коллективное использование и в основном лишены недостатков, присущих АИС АФ. В БнД хранимая информация сосредоточена в едином информационном массиве – базе данных (БД), а процесс манипулирования данными автоматизирован.
4. по степени распределенности:
   1. локальная система размещается на одной ЭВМ;
   2. распределенная система функционирует в среде вычислительной сети и распределена по ее узлам (серверам и рабочим станциям).

Вопрос 6: что понимается под трехуровневой архитектурой ANSI/SPARC?

Ответ: архитектура ANSI/SPARC представляет собой стандартную организацию базы данных, разделенную на три уровня:

1. В основе архитектуры ANSI/SPARC лежит концептуальный уровень. В современных СУБД он может быть реализован при помощи представления. Концептуальный уровень описывает данные и их взаимосвязи с наиболее общей точки зрения, — концепции архитекторов базы, используя реляционную или другую модель.
2. Внутренний уровень – позволяет скрыть подробности физического хранения данных (носители, файлы, таблицы, триггеры ...) от концептуального уровня. Отделение внутреннего уровня от концептуального обеспечивает так называемую физическую независимость данных.
3. Внешний уровень – описывает различные подмножества элементов концептуального уровня для представлений данных различным пользовательским программам. Каждый пользователь получает в свое распоряжение часть представлений о данных, но полная концепция скрыта. Отделение внешнего уровня от концептуального обеспечивает логическую независимость данных.

Вопрос 7: дайте определения внешней схеме БД, концептуальной схеме БД, внутренней схеме БД.

Ответ:

1. Внешняя схема БД: это описание структуры данных, которое видно и используется конкретным пользователем или приложением. Внешняя схема определяет, какие данные из концептуальной схемы видны и доступны для конкретного пользователя, обеспечивая уровень абстракции для удобства использования.
2. Концептуальная схема БД: это абстрактное представление о структуре данных в базе данных, независимое от конкретных приложений. Концептуальная схема определяет сущности, их атрибуты и отношения между ними, предоставляя общую модель данных для всей базы данных.
3. Внутренняя схема БД: это описание того, как данные фактически хранятся и обрабатываются на физическом уровне. Внутренняя схема заботится о деталях, таких как структура файлов, индексы и оптимизация запросов, предоставляя эффективное физическое хранение данных.

Вопрос 8: каковы особенности иерархической модели организации данных?

Ответ: иерархическая модель данных (ИМД) – это модель, в которой абстрактные понятия находятся в отношении предшествования таким образом, что каждому понятию соответствует только один предшественник (родитель).

Достоинство ИМД: данные, отражающие общие свойства совокупности конкретных данных не дублируются.

Недостаток ИМД связан с дублированием данных в случае их однотипности.

Вопрос 9: каковы особенности сетевой модели организации данных?

Ответ: в сетевых структурах данных потомок может иметь любое число предков. Связи между записями в сетевой МД выполняются в виде указателей, то есть каждая запись хранит ссылку на другую однотипную запись (или признак конца списка) и ссылки на списки подчиненных записей.

Достоинство сетевых МД: данные имеют четкую структуру.

Недостатком сетевых МД является большое количество дополнительной информации о связях.

Вопрос 10: каковы особенности многомерной модели организации данных?

Ответ: информация в многомерной модели представляется в виде многомерных массивов, называемых гиперкубами. В одной базе данных, построенной на многомерной модели, может храниться множество таких кубов, на основе которых можно проводить совместный анализ показателей. Конечный пользователь в качестве внешней модели данных получает для анализа определенные срезы или проекции кубов, представляемые в виде обычных двумерных таблиц или графиков.

Достоинства: быстрый аналитический доступ, гибкость в агрегации данных и поддержка бизнес-аналитики обеспечивают эффективный анализ и принятие решений.

Недостатки: ограниченная структура данных, сложность реализации и ограниченная масштабируемость могут быть препятствиями при обработке неструктурированных данных и работе с большими объемами информации.

Вопрос 11: каковы особенности постреляционной модели организации данных?

Ответ: постреляционная модель данных в общем случае представляет собой расширенную реляционную модель, снимающую ограничение неделимости значений полей. То есть, допускаются многозначные поля, значения которых состоят из подзначений. Набор значений многозначных полей считается самостоятельной таблицей, встроенной в основную.

Достоинство постреляционной модели данных: возможность представления связанных реляционных таблиц одной постреляционной таблицей.

Недостаток постреляционной модели: сложность в обеспечении целостности данных.

Вопрос 12: что относится к неструктурированным данным?

Ответ: неструктурированные данные представляют собой информацию, которая не имеет четкой организации или формы. Это может включать в себя текстовые документы, аудио- и видеозаписи, изображения, электронные письма, социальные медиа-публикации и другие формы данных, которые не подчиняются строгой таблице или схеме, как структурированные данные в базах данных.

Вопрос 13: в чем преимущество использования колоночной СУБД по  
сравнению с реляционной?

Ответ: преимущество использования колоночной СУБД по сравнению с реляционной заключается в эффективности хранения и обработки данных. В колоночных базах данных данные хранятся по столбцам, что обеспечивает более компактное использование памяти и ускоренный доступ к конкретным полям данных при выполнении аналитических запросов. Это особенно полезно при работе с большими объемами данных, где производительность и эффективность запросов играют ключевую роль.

Вопрос 14: Каким образом осуществляется связь между таблицами в  
реляционной СУБД?

Ответ: Связь между таблицами в реляционной СУБД осуществляется с использованием ключей. Есть два основных типа ключей:

1. Первичный ключ (Primary Key): это уникальный идентификатор записи в таблице. Каждая таблица имеет первичный ключ, который однозначно идентифицирует каждую строку.
2. Внешний ключ (Foreign Key): это поле в таблице, которое является первичным ключом в другой таблице. Внешний ключ создает связь между двумя таблицами, позволяя одной таблице ссылаться на данные в другой.

Вопрос 15: Каким образом на этапе проектирования решается проблема  
дублирующих записей в таблице?

Ответ: на этапе проектирования реляционной базы данных проблема дублирующих записей решается путем использования нормализации данных (приведение к нормальной форме, первая, вторая и т.д.). Нормализация – это процесс организации данных так, чтобы избежать избыточности и дублирования информации.

**Вывод:** в ходе лабораторной работы были изучены основы логического проектирования базы данных. Также, был освоен процесс разработки логической структуры базы данных, с помощью построения диаграммы (логической схемы)  
«сущность-связь» в программе «draw.io».