Государственное бюджетное профессиональное образовательное учреждение Республики Марий Эл

«ЙОШКАР-ОЛИНСКИЙ ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ»

|  |
| --- |
| Специальность: 09.02.07 Информационные системы и программирование  Группа: А-31  Дисциплина: Технология разработки |

ОТЧЕТ ПО

Практической работе № 2

«Проектирование реляционной схемы базы данных»

Руководитель: Пинешкин Ю.С

Выполнил(а): студентки А-31

Николаева С.В

Йошкар-Ола

2025

**Теоретический материал**

**Проектирование реляционной схемы базы данных**— это процесс разработки структуры данных с использованием реляционной модели данных. Цель — минимизировать дублирование данных, упростить процедуры их обработки и обновления, обеспечить целостность данных. **Этапы**

Проектирование реляционной схемы базы данных в общем случае включает **три этапа**:

1. **Концептуальное проектирование** — изучение и описание предметной области, выявление объектов и процессов, подлежащих загрузке в базу данных. Результат — информационно-логическая модель данных предметной области (концептуальная модель).
2. **Логическое проектирование** — выбор конкретной системы управления базами данных (СУБД) и отображение концептуальной модели в логическую модель, основанную на структурах, характерных для выбранной СУБД. Для реляционной базы данных — разработка структуры таблиц, определение их ключей, связей между ними, оптимизация модели (минимизация избыточности данных, устранение их дублирования).
3. **Физическое проектирование** — логическая модель расширяется характеристиками, необходимыми для определения способов физического хранения и использования базы данных, типа устройств для хранения, объёма памяти, правил сопровождения базы данных и т. п.

**Методы**

Для проектирования реляционной схемы базы данных используют, например:

* **Метод ER-диаграмм (семантическое моделирование)** — моделирование структуры данных, опираясь на смысл этих данных.
* **Метод пошаговой декомпозиции** — последовательное разбиение исходной и промежуточных схем отношений до тех пор, пока результирующие отношения не будут удовлетворять заданным свойствам.
* **Метод синтеза** — конструирование набора декомпозиционных подсхем, удовлетворяющих определённым свойствам, из заданного множества атрибутов выбранной предметной области на основе заданного множества функциональных зависимостей, связывающих эти атрибуты.

**Норма**

При проектировании реляционной схемы базы данных важно **нормализовать отношения** — устранять потенциальную противоречивость и избыточность данных. Некоторые правила нормализации:

* **Первая нормальная форма (1NF)** — в каждой ячейке таблицы должно быть только одно значение, каждая строка идентифицируется уникально по первичному ключу.
* **Вторая нормальная форма (2NF)** — если ключ в таблице состоит из нескольких полей (составной ключ), то каждое поле с данными (не входящее в ключ) должно зависеть от всех этих полей вместе, а не от части.
* **Третья нормальная форма (3NF)** — в таблице не должно быть зависимостей между столбцами, которые не являются ключами. Каждое поле в таблице должно напрямую зависеть только от главного идентификатора записи (первичного ключа), а не от других полей.

**Инструменты**

**Для проектирования реляционной схемы базы данных используют CASE-средства (Computer Aided Software Engineering).** Например:

* **ERwin (Logic Works) —** инструмент для создания концептуальных и логических схем баз данных.
* **S-Designor (SDP) —** графический CASE-инструмент для проектирования структуры реляционных баз данных, создаёт модели баз данных в два этапа — выстраивая концептуальную модель и затем преобразуя её в физическую.

**Проектирование системы кассовых операций**

Проектирование реляционной схемы базы данных для системы кассовых операций включает в себя несколько ключевых этапов.

1. **Определение сущностей**

Для системы кассовых операций можно выделить следующие основные сущности:

• Клиент (Customer) – ID\_клиента (CustomerID) – Имя (FirstName) – Фамилия (LastName) – Телефон (Phone) – Email (Email)

• Товар (Product) – ID\_товара (ProductID) – Название (ProductName) – Описание (Description) – Цена (Price) – Количество на складе (StockQuantity)

• Чек (Receipt) – ID\_чека (ReceiptID) – Дата и время (DateTime) – ID\_клиента (CustomerID) [внешний ключ] – Общая сумма (TotalAmount)

• Строка чека (ReceiptItem) – ID\_строки (ReceiptItemID) – ID\_чека (ReceiptID) [внешний ключ] – ID\_товара (ProductID) [внешний ключ] – Количество (Quantity) – Цена (Price) [может отличаться от текущей цены товара]

• Сотрудник (Employee) – ID\_сотрудника (EmployeeID) – Имя (FirstName) – Фамилия (LastName) – Должность (Position)

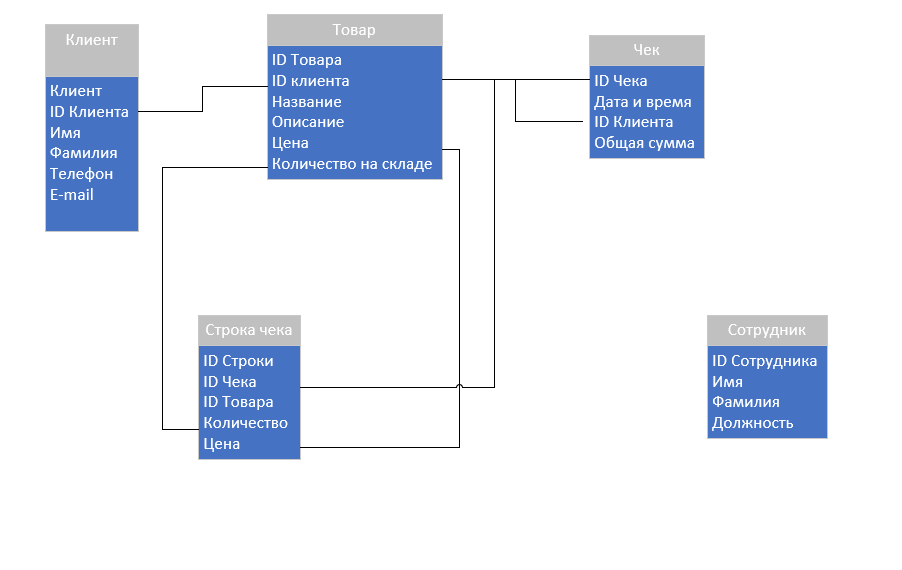


Рисунок 1 – пример Проектирование системы кассовых операций

1. **Определение связей между сущностями**

• Клиент может совершать несколько чеков: – Связь: Один ко многим (1:N) между Клиент и Чек.

• Чек может содержать несколько строк: – Связь: Один ко многим (1:N) между Чек и Строка чека.

• Товар может быть в нескольких строках чеков: – Связь: Один ко многим (1:N) между Товар и Строка чека.

• Сотрудник может обрабатывать несколько чеков: – Если необходимо, можно добавить поле EmployeeID в таблицу Чек, чтобы отслеживать, какой сотрудник обработал чек.

1. **Нормализация**

Важно также провести нормализацию базы данных, чтобы избежать избыточности данных и аномалий. Основные правила нормализации включают:

• Убедиться, что каждая таблица содержит только одну тему.

• Удалить повторяющиеся группы данных.

• Обеспечить, чтобы все атрибуты зависели только от первичного ключа.

1. **Дополнительные аспекты**

• История операций: возможно, потребуется добавить таблицу для хранения истории изменений цен на товары или истории транзакций.

• Отчеты: можно создать представления или дополнительные таблицы для формирования отчетов о продажах, анализе клиентов и т.д.

1. **Заключение**

Проектирование реляционной схемы базы данных для системы кассовых операций требует тщательного анализа требований и понимания бизнес-процессов. Указанные выше сущности и связи могут быть адаптированы в зависимости от конкретных потребностей вашего проекта.