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-251: Ивченко Я.А.

подпись, дата

Воронеж 2025

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

**Ход выполнения работы:**

В ходе лабораторной работы была спроектирована реляционная модель базы данных для автоматизации работы туристического агентства. Модель предназначена для хранения информации о клиентах, турах, отелях, бронированиях и платежах.

Диаграмма базы данных построена с помощью инструмента draw.io представлена на рисунке 1.

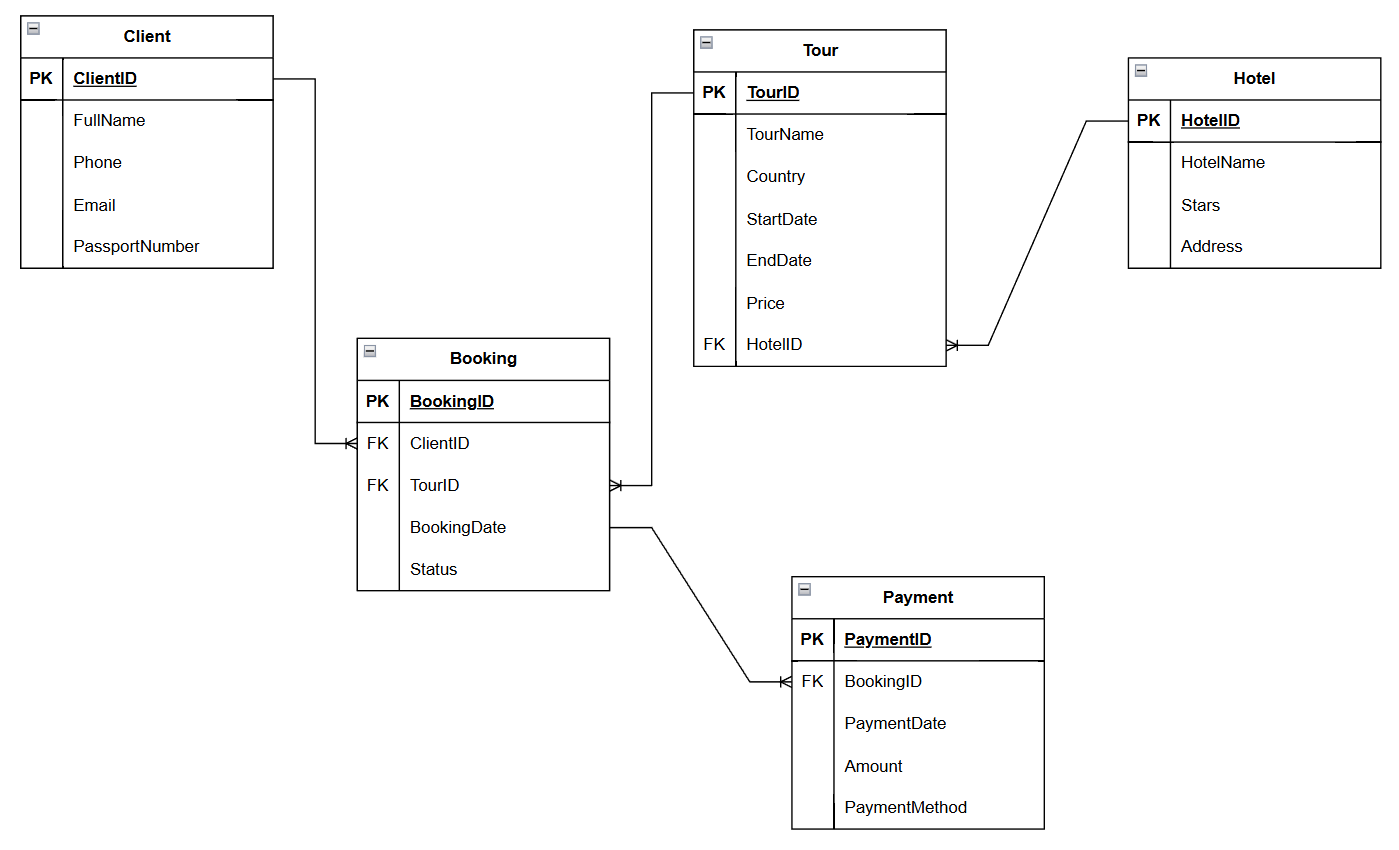


Рисунок 1 – Диаграмма базы данных

Описание сущностей и атрибутов:

1. Клиент – хранит информацию о туристах.

2. Тур – описывает предлагаемые агентством туры.

3. Отель – справочник отелей, связанных с турами.

4. Бронирование – фиксирует заказы клиентов на туры.

5. Оплата – хранит данные о внесённых платежах.

Первичные ключи (PK) уникально идентифицируют записи в таблицах.

Внешние ключи (FK) обеспечивают связи между таблицами:

- ClientID в Бронирование ссылается на Клиент;

- TourID в Бронирование ссылается на Тур;

- HotelID в Тур ссылается на Отель;

- BookingID в Оплата ссылается на Бронирование.

Связи

1. Клиент – Бронирование: один клиент может иметь много бронирований (1:M).

2. Тур – Бронирование: один тур может быть забронирован многими клиентами (1:M).

3. Бронирование – Оплата: одно бронирование может иметь несколько оплат (1:M).

4. Тур – Отель: одному отелю могут соответствовать несколько туров (M:1).

Связь «многие ко многим» между Клиентами и Турами реализована через промежуточную сущность Бронирование.

**Вывод:** в результате лабораторной работы была успешно спроектирована реляционная база данных для туристического агентства. Полученная модель является логически целостной, непротиворечивой и готова к физической реализации в СУБД.

**Ответы на контрольные вопросы**

1. Данные – это совокупность дискретных, объективных фактов (цифр, символов, слов, кодов), не несущих смысловой нагрузки без учета контекста. Это "сырье" для получения информации.

База данных (БД) – это организованная, структурированная совокупность данных, постоянного хранящихся в памяти вычислительной системы, отражающая состояние предметной области и предназначенная для удовлетворения информационных потребностей пользователей.

Система управления базами данных (СУБД) – это программное обеспечение, которое служит для создания, ведения и совместного использования БД многими пользователями. СУБД является посредником между БД и прикладными программами/пользователями.

Ведение базы данных (Администрирование БД) – это комплекс мероприятий по обеспечению корректной, надежной и эффективной работы БД. Включает в себя: проектирование, наполнение, обеспечение безопасности, резервное копирование, восстановление после сбоев, мониторинг производительности и т.д.

2. Данные — это сырые факты и цифры.

Информация — это обработанные данные, имеющие смысл в конкретной ситуации.

Простой пример:

Данные: 25, Иванов, 27.05.2024

Информация: «Клиент Иванов совершил покупку на 25 рублей 27 мая 2024 года»

3. База данных (БД) — это сами данные, хранящиеся в определенном порядке.

СУБД — это программное обеспечение для работы с этими данными.

Банк данных (БнД) — это более широкое понятие. Это информационная система, которая включает в себя:

Простая аналогия: Библиотека (Банк данных) содержит книги (База данных) и библиотекаря (СУБД), который выдает книги по запросу.

4. База данных: Хранилище структурированных данных.

СУБД: Обеспечивает создание, доступ, модификацию и управление данными.

Языковые средства:

Язык определения данных (DDL) — для описания структуры БД.

Язык манипулирования данными (DML) — для работы с данными (запросы, обновления).

Методические средства: Инструкции, руководства, регламенты по проектированию и использованию БД.

Административные средства и персонал (АБД): Обеспечивают функционирование всей системы, безопасность, целостность и производительность.

5. Фактографические системы: Хранят строго структурированные данные в виде записей и полей (например, кадровые системы, системы учета заказов).

Документальные системы: Хранят слабоструктурированные или неструктурированные документы (тексты, письма, законы). Поиск ведется по метаданным или содержимому.

Геоинформационные системы (ГИС): Хранят и обрабатывают пространственные данные (карты, планы, привязка объектов к местности).

Мультимедийные системы: Хранят аудио, видео, изображения.

6. Это концепция, разделяющая описание БД на три уровня абстракции для обеспечения независимости данных.

Внешний уровень (External Level): Уровень представления данных для конкретных пользователей или приложений. Определяет различные "внешние схемы" (представления, View), скрывая ненужные детали.

Концептуальный уровень (Conceptual Level): Общее, логическое представление всей БД в целом. Описывает какие данные хранятся, и отношения между ними, без привязки к физическому хранению.

Внутренний уровень (Internal Level): Описывает физическое хранение данных на носителе (структуры файлов, индексы, типы записей).

Цель: Изменения на одном уровне (например, физическое хранение) не требуют изменений на других уровнях (логическая структура или приложения).

7. Внешняя схема БД: Это описание части БД, необходимой конкретному пользователю или приложению. По сути, это "окно" в данные. Может быть несколько внешних схем для одной БД.

Концептуальная схема БД: Это полное логическое описание всей БД: сущности, атрибуты, связи, ограничения целостности. Она едина для всей БД и не зависит от приложений.

Внутренняя схема БД: Это описание физической реализации БД: как данные хранятся на диске, какие используются структуры (файлы, индексы, методы доступа).

8. Иерархическая модель:

Данные организованы в виде дерева с одним корнем.

Связи: "родитель-потомок". У записи-потомка ровно один родитель.

Недостаток: Сложность отображения сложных связей "многие-ко-многим".

Сетевая модель:

Усовершенствование иерархической. Запись может иметь несколько родителей.

Позволяет напрямую моделировать сложные связи.

Недостаток: Высокая сложность структуры и программирования.

Реляционная модель:

Данные представлены в виде таблиц (отношений).

Связи устанавливаются на основе значений данных (ключей).

Преимущество: Простота понимания, мощный математический аппарат (реляционная алгебра).

Многомерная модель:

Используется для систем аналитической обработки (OLAP).

Данные организованы в виде "кубов", где измерения — это атрибуты анализа (время, регион, товар), а меры — числовые показатели (продажи, прибыль).

Преимущество: Высокая скорость выполнения сложных аналитических запросов.

Постреляционная модель:

Развитие реляционной модели, снимающее ограничение "все атрибуты должны быть атомарными".

Позволяет хранить в одном поле составные значения, массивы или даже другие таблицы.

Пример: СУБД UniData, UniVerse.

9. Это данные, которые не имеют предопределенной модели или не организованы в соответствии с набором правил.

Примеры: Текстовые документы (договоры, статьи), электронные письма, видео- и аудиофайлы, изображения, PDF-файлы, сообщения в социальных сетях.

10. Преимущество колоночной СУБД перед реляционной (строчной)

Реляционная (строчная) СУБД: Хранит данные построчно. Все поля одной записи лежат рядом на диске.

Колоночная СУБД: Хранит данные по столбцам. Все значения одного столбца лежат рядом на диске.

Преимущество колоночного хранения: При аналитических запросах (OLAP), которые часто агрегируют данные по нескольким столбцам (например, SUM(продажи) BY регион), системе не нужно загружать всю строку со всеми полями. Она считывает только нужные столбцы, что резко увеличивает скорость выполнения запросов.

11. Связь осуществляется через внешние ключи (Foreign Key, FK).

В таблице-потомке создается столбец (или несколько столбцов), который является внешним ключом.

Значения этого столбца должны соответствовать значениям первичного ключа (Primary Key, PK) или уникального ключа в таблице-родителе.

СУБД обеспечивает целостность ссылок (Referential Integrity), запрещая операции, которые нарушили бы эти связи (например, нельзя удалить запись в родительской таблице, если на нее есть ссылки).

12. Проблема решается путем нормализации базы данных и правильного назначения ключей.

Определение первичного ключа (Primary Key): Для каждой таблицы выбирается атрибут (или набор атрибутов), который уникально идентифицирует каждую запись. Это исключает полные дубликаты.

Нормализация (в основном 1NF и 2NF): Устранение повторяющихся групп данных путем вынесения их в отдельные таблицы. Например, вместо хранения нескольких одинаковых адресов для одного клиента в одной записи, создается отдельная таблица "Адреса" со связью "один-ко-многим".

Создание уникальных индексов: На атрибуты, которые по смыслу должны быть уникальными (например, email, паспортные данные), накладывается ограничение UNIQUE, чтобы СУБД сама блокировала вставку дубликатов.

13. Колоночные СУБД оптимизированы для аналитических запросов и обработки больших данных. Их главное преимущество — высокая скорость выполнения операций агрегации (SUM, AVG, COUNT) и обработки данных в отдельных столбцах. Это достигается за счет хранения данных по столбцам, а не по строкам. При аналитическом запросе система считывает только нужные столбцы, а не всю строку целиком. Дополнительное преимущество — эффективное сжатие данных, так как в одном столбце обычно хранятся однотипные значения.

14. Связь между таблицами в реляционной СУБД осуществляется через механизм внешних ключей. Внешний ключ — это поле в одной таблице, которое ссылается на первичный ключ в другой таблице. Этот механизм обеспечивает целостность данных: СУБД следит за тем, чтобы значения внешнего ключа всегда соответствовали существующим значениям в связанной таблице. Связи бывают трех типов: один-ко-многим, один-к-одному и многие-ко-многим.

15. Проблема дублирования записей решается путем нормализации базы данных и правильного проектирования структуры таблиц. Основные методы включают назначение первичных ключей для уникальной идентификации каждой записи, вынесение повторяющихся данных в отдельные таблицы с установлением связей между ними, а также использование ограничений уникальности для полей, которые должны содержать уникальные значения. Это обеспечивает целостность данных и исключает избыточность.