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-241: Тогушов В.А.

подпись, дата

Принял: Короленко В.В.

подпись, дата

Воронеж 2024

Цель лабораторной работы:

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

Основные задачи:

* определение сущности для проекта в соответствии с индивидуальным заданием и их атрибуты;
* выделение ключевых атрибутов;
* определение связей между сущностями и типов связей;
* построение диаграммы сущность-связь для отображения логической структуры базы данных.

Учебная задача:

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.

Индивидуальное задание:

Выполнить все пункты раздела «Учебная задача», оформить отчёт, представить результаты выполнения лабораторной работы к защите.

Вариант: база данных «Учет сделок с недвижимостью»

Предлагаемый набор базовых таблиц:

1. КВАРТИРЫ

2. РИЭЛТОРЫ

3. СДЕЛКИ

Минимальный набор полей базовых таблиц

1. Название улицы

2. Номер дома

3. Номер квартиры

4. Площадь квартиры

5. Количество комнат

6. Дата сделки

7. Цена квартиры

8. ФИО риэлтора

9. Процент вознаграждения

Описание предметной области

Фирма занимается оформлением сделок с объектами жилой недвижимости. При оформлении сделки фиксируется информация о продаваемой квартире, о риэлторе, оформляющем сделку купли-продажи, о дате оформления сделки. Риэлтор, оформивший сделку купли-продажи, получает комиссионное вознаграждение, которое вычисляется как

*Цена квартиры х Процент вознаграждения.*

Процент вознаграждения является индивидуальным и фиксированным для каждого конкретного риэлтора.

Базы данных — это упорядоченные наборы данных, предназначенные для хранения, управления и обработки информации. Основные модели данных:

* Иерархическая – данные представляются в виде дерева.
* Сетевая – каждая запись может иметь несколько связанных записей.
* Реляционная – данные организованы в таблицы, где строки — записи, а столбцы – атрибуты.
* Объектно-ориентированная – данные представлены в виде объектов с методами и свойствами.

Реляционная модель наиболее распространена благодаря своей простоте, нормализации данных и поддержке запросов SQL.

Draw.io – это бесплатный онлайн-сервис, который позволяет создавать и совместно редактировать диаграммы, различные схемы и прочие визуальные представления данных.

Основные возможности сервиса:

* Моделирование бизнес-процессов. Сервис поддерживает различные нотации, в том числе BPMN (Business Process Model and Notation).
* Создание UML-диаграмм. UML (Unified Modeling Language) — это графический язык, в котором каждому символу и фигуре присвоены конкретные значения.
* Разработка прототипов. Сервис позволяет создавать макеты приложений и сайтов.
* Создание инфографики. Поскольку в сервисе много встроенных графических элементов, создавать там графики для презентаций легко и удобно.
* Разработка архитектуры приложений и сайтов. Архитектура описывает, как организованы отдельные компоненты внутри системы, как они взаимодействуют друг с другом, какие свойства имеют.
* Проектирование помещений. Дизайнеры по интерьерам и архитекторы иногда используют Draw.io для создания планов квартир или общественных пространств.

На основе описания предметной области можно выделить основные сущности:

1) Квартиры – описывает объект недвижимости, который является центральной частью сделки (адрес, площадь, комнаты).

2) Риэлторы – необходимы для фиксации информации о сотрудниках, их результативности и расчёта вознаграждения (ФИО, процент вознаграждения).

3) Сделки – описывают каждую операцию купли-продажи, объединяя квартиру, риэлтора и клиентов (дата, цена, риэлтор, квартира).

4) Клиенты – Позволяют учитывать покупателей и продавцов, фиксировать их контакты.

Каждая сущность отражает конкретный объект, необходимый для учета операций фирмы.

Определение атрибутов сущностей:

Квартиры:

- Улица;

- Номер дома;

- Номер квартиры;

- Площадь;

- Количество комнат;

- PK: Улица, Номер дома, Номер квартиры

Риэлторы:

- ID риэлтора (PK);

- ФИО;

- Процент вознаграждения.

Сделки

- ID сделки (PK);

- Дата сделки;

- Цена квартиры;

- ID квартиры (FK);

- ID риэлтора (FK).

Клиенты:

- ID клиента (PK);

- ФИО;

- Контактная информация.

Ключевые атрибуты:

Первичные ключи (PK):

- ID риэлтора, ID сделки, ID клиента;

- Для квартир первичным ключом будет составной ключ из атрибутов: Улица + Номер дома + Номер квартиры.

Внешние ключи (FK):

- В таблице сделок: ID риэлтора, ID квартиры.

Связи между сущностями:

Квартиры и Сделки: Один к одному (1:1), так как каждая квартира участвует только в одной сделке, так как сделка завершает процесс продажи или покупки.

Риэлторы и Сделки: Один ко многим (1:N), так как один риэлтор может оформить множество сделок, но каждая сделка оформляется только одним риэлтором.

Клиенты и Сделки: Многие ко многим (M:N), так как один клиент может участвовать в нескольких сделках, а в одной сделке могут быть задействованы несколько клиентов.

Проблема связи многие ко многим:

Одна сделка может включать нескольких клиентов:

- Покупатель

- Продавец или клиент может участвовать в нескольких сделках:

- Один клиент может купить несколько квартир.

- Один клиент может быть продавцом в одной сделке и покупателем в другой.

Теперь избавимся от связи многие ко многим, введением дополнительной сущности (таблицы) «Сделки-клиенты»:

- ID клиента (FK);

- ID сделки (FK);

- Роль клиента в сделке (например, "покупатель" или "продавец");

- PK: ID сделки\_клиента.

Эта таблица связывает каждую сделку с одним или несколькими клиентами и фиксирует их роли в сделке.

Теперь связь между "Клиенты" и "Сделки" заменяется на две связи через промежуточную таблицу "Сделки-Клиенты":

Клиенты и Сделки-Клиенты: Один ко многим (1:N), так как один клиент может участвовать в нескольких сделках, но в каждой сделке он фигурирует один раз.

Сделки и Сделки-Клиенты: Один ко многим (1:N), так как каждая сделка может включать нескольких клиентов, например, покупателя и продавца.

Теперь приступим к созданию диаграммы в формате draw.io (Рисунок 1).

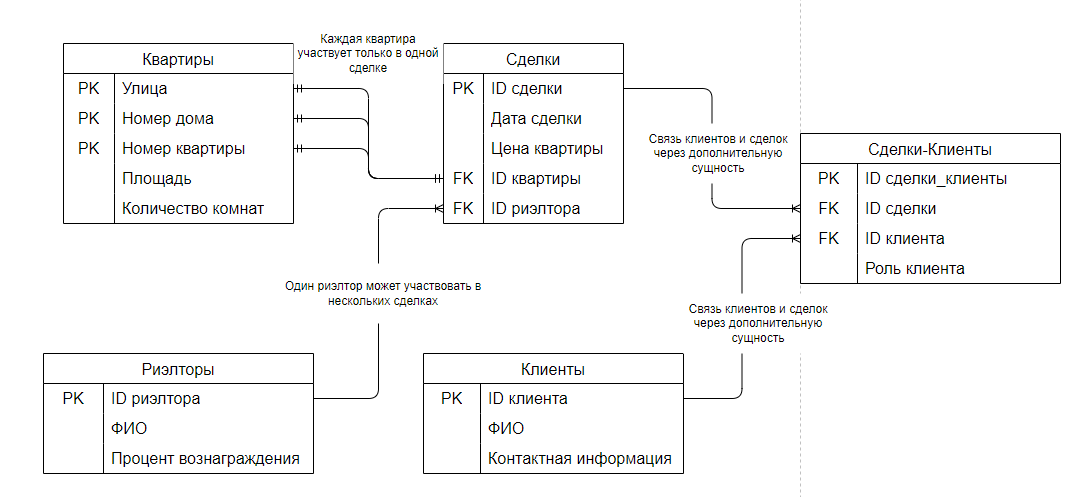


Рисунок 1 – Диаграмма логической структуры базы данных

Контрольные вопросы:

1) Дайте определения следующим понятиям: данные, база данных, СУБД, ведение базы данных.

Данные – это информация, зафиксированная в определённой форме, пригодной для последующей обработки, передачи и хранения.

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

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

Ведение базы данных – деятельность по обновлению, восстановлению и изменению структуры базы данных с целью обеспечения её целостности, сохранности и эффективности использования.

2) В чем отличие данных от информации?

Отличие данных от информации заключается в том, что данные – это фиксированные сведения о событиях и явлениях, которые хранятся на определённых носителях, а информация появляется в результате обработки данных при решении конкретных задач.

3) В чем отличие базы данных от банка данных и СУБД?

Отличие базы данных от банка данных и системы управления базами данных (СУБД) заключается в следующем:

1. База данных — это совокупность взаимосвязанных данных из определённой предметной области, организованных специальным образом и хранимых во внешней памяти (файлах базы данных).

2. Банк данных — это система специально организованных данных, а также технических, программных, языковых и организационно-методических средств, предназначенных для централизованного накопления и коллективного многоцелевого использования данных. В состав банка данных входят одна или несколько баз данных, справочник баз данных, СУБД, а также библиотеки запросов и прикладных программ.

3. Система управления базами данных (СУБД) — это совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями. СУБД — комплекс программ, позволяющих создать базу данных и манипулировать данными (вставлять, обновлять, удалять и выбирать). Система обеспечивает безопасность, надёжность хранения и целостность данных, а также предоставляет средства для администрирования БД.

4) Назовите основные компоненты банка данных и их назначение.

В структуре банка данных выделяют следующие компоненты:

- Информационная база;

- Лингвистические средства;

- Программные средства;

- Технические средства;

- Организационно-административные подсистемы и нормативно-методическое обеспечение.

К основным функциям банка данных относятся: хранение данных и их защита, изменение (обновление, добавление и удаление) хранимых данных, поиск и отбор данных по запросам пользователей, обработка данных и вывод результатов.

5) Классифицируйте АИС по типу хранимых данных.

АИС можно классифицировать по типу хранимых данных на следующие виды:

1. Документальные информационно-поисковые системы (ДИСП). Предназначены для хранения и обработки документальных данных: адресов хранения документов, наименований, описаний, а также текстов документов, графических изображений (например, географических карт), звуковой информации (мелодии) и т. д. Такие данные представляются в неструктурированном виде.

2. Фактографические информационно-поисковые системы (ФИПС). Хранят и обрабатывают структурированные данные в виде чисел и текстов. Над такими данными можно выполнять различные операции (нахождение суммы, минимума, максимума и т. п.).

6) Что понимается под трехуровневой архитектурой ANSI/SPARC?

Трехуровневая архитектура ANSI-SPARC – это принципы, согласно которым рекомендуется строить системы управления базами данных (СУБД). Проект архитектуры был выдвинут в 1975 году подкомитетом SPARC ANSI.

Выделяется три уровня системы:

1. Внешний (пользовательский) уровень. Описывает часть базы данных, которая имеет отношение к конкретному пользователю. Он исключает нерелевантные данные, а также данные, доступ к которым пользователю не разрешён.

2. Концептуальный уровень. Это способ описания того, какие данные хранятся во всей базе данных и как они взаимосвязаны. Концептуальный уровень не определяет, как данные хранятся физически.

3. Внутренний (физический) уровень. Включает в себя то, как база данных физически представлена в компьютерной системе. Он описывает, как данные фактически хранятся в базе данных и на компьютерном оборудовании.

7) Дайте определения внешней схеме БД, концептуальной схеме БД, внутренней схеме БД.

1. Внешняя схема базы данных – схема базы данных, поддерживаемая системой управления базы данных для приложений.

2. Концептуальная схема базы данных – это высший уровень абстракции, представляющий логическое представление всей базы данных, как её воспринимают конечные пользователи или разработчики приложений.

3. Внутренняя схема базы данных – схема базы данных, определяющая представление данных в среде хранения и пути доступа к ним.

8) Каковы особенности иерархической модели организации данных?

Особенности иерархической модели организации данных:

* Древовидная структура. Один родительский узел может иметь несколько дочерних узлов, но дочерний узел может иметь только один родительский узел.
* Отношения «родитель-потомок» представлены через родительские указатели или вложенные наборы.
* Оптимизация перехода от родительского узла к дочернему, а не наоборот.
* Поддержка отношений «один ко многим», когда родительский объект связан с несколькими дочерними объектами.

9) Каковы особенности сетевой модели организации данных?

Некоторые особенности сетевой модели организации данных:

* Представление взаимосвязей данных. Сетевая модель использует графовую структуру для представления взаимосвязей данных, допуская взаимосвязи «многие ко многим».
* Записи и наборы. Данные в сетевой модели организованы в записи и наборы.
* Связи владелец-участник. Сетевая модель определяет связи данных с использованием пар владелец-участник.
* Навигационный доступ. Сетевая модель поддерживает навигационный доступ к данным, при котором доступ к записям осуществляется по заранее определённым путям.
* Иерархические и неиерархические структуры. Сетевая модель может представлять как иерархические (древовидные), так и неиерархические (графоподобные) структуры, обеспечивая гибкость при моделировании данных.

10) Каковы особенности многомерной модели организации данных?

Особенности многомерной модели организации данных включают:

* Многомерное логическое представление структуры информации при описании и в операциях манипулирования данными.
* Использование измерений. Измерения выступают в роли индексов, которые служат для идентификации конкретных значений в ячейках гиперкуба.
* Применение специальных операций. К ним относятся формирование «среза», «вращение», агрегация и детализация.
* Использование иерархий. Иерархии — это способ организации измерений по уровням детализации.

11) Каковы особенности постреляционной модели организации данных?

Особенности постреляционной модели организации данных включают:

* Снятие ограничения неделимости данных. Допускаются многозначные поля, значения которых состоят из подзначений.
* Поддержка множественных групп. Совокупность объединённых множественных полей называется ассоциацией.
* Отсутствие требований на длину и количество полей в записях. Это делает структуру таблиц более наглядной.
* Использование расширенного SQL. Язык запросов позволяет извлекать сложные объекты из одной таблицы без операций соединения.

12) Что относится к неструктурированным данным?

К неструктурированным данным относятся любые данные, не имеющие заранее заданной структуры или организации.

Примеры неструктурированных данных: текстовые документы. изображения и видео, аудиофайлы, социальные медиа, показания датчиков.

13) В чем преимущество использования колоночной СУБД по сравнению с реляционной?

В отличие от реляционных систем управления базами данных, где при запросе поиск осуществляется по всей таблице, в колоночных он производится только по конкретным столбцам, что на порядок ускоряет получение нужных данных для анализа. Это делает колоночные СУБД незаменимыми для получения данных из таблиц, размер которых может достигать сотен ТБ, а возможность сильного сжатия данных экономит место на серверах.

14) Каким образом осуществляется связь между таблицами в реляционной СУБД?

Связь между таблицами в реляционной СУБД осуществляется через внешний ключ. Он связывает поле (значение) исходной таблицы с первичным ключом внешней таблицы.

Выделяют три разновидности связи между таблицами базы данных:

1. Отношение «один-ко-многим». Одной записи родительской таблицы может соответствовать несколько записей дочерней.

2. Отношение «один-к-одному». Одной записи в родительской таблице соответствует одна запись в дочерней.

3. Отношение «многие-ко-многим». Одной записи в родительской таблице соответствует более одной записи в дочерней, и наоборот.

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

Проблема дублирующих записей в таблице на этапе проектирования решается через процесс нормализации. Он основан на анализе функциональных зависимостей между атрибутами отношений и направлен на устранение избыточности данных.