

Самостоятельная работа №2 по БД

Задание № 2.1 Визуализация моделей и подходов к организации данных

1. Реляционная модель

Предметная область: Библиотека

Описание взаимоотношений:

- Книги: Хранят информацию о названии, авторе, жанре и ISBN.
- Авторы: Хранят данные о фамилии, имени и дате рождения.
- Читатели: Хранят информацию о фамилии, имени, дате рождения и номере телефона.
- Выдачи: Связывают читателей и книги, содержат данные о дате выдачи и сроке возврата.

Взаимоотношения:

- Каждая книга может иметь одного или нескольких авторов (Отношение "многие ко многим").
- Читатель может взять несколько книг (Отношение "один ко многим").
- Каждая выдача связывает одну книгу и одного читателя.

2. Иерархическая модель

Предметная область: Организационная структура компании

Описание взаимоотношений:

- Компания: Корневая сущность, представляющая всю организацию.
- Отделы: Подразделения компании, например, HR, IT, Продажи.
- Сотрудники: Лица, работающие в каждом отделе.

Взаимоотношения:

- Каждый отдел подчинен компании (один ко многим).
- Каждый сотрудник подчинен определенному отделу (один ко многим).
- Структура может иметь многоуровневую иерархию, где отделы могут содержать подотделы.

3. Сетевая модель

Предметная область: Автомобильные маршруты

Описание взаимоотношений:

- Города: Узлы сети, представляющие точки на карте.

- Дороги: Ребра, соединяющие города.

Взаимоотношения:

- Каждый город может быть связан с несколькими другими городами (отношение "многие ко многим").
- Каждая дорога может иметь различные характеристики, такие как длина и тип (асфальт, гравий и т.д.).

4. Объектно-ориентированная модель

Предметная область: Онлайн-магазин

Описание взаимоотношений:

- Товар: Объект с атрибутами, таким как название, цена и описание.
- Категория: Объект, который группирует товары, может содержать подкатегории.
- Пользователь: Объект, представляющий клиента магазина с атрибутами, такими как имя и адрес.
- Заказ: Объект, связывающий пользователя с товарами, которые он приобрел.

Взаимоотношения:

- Каждый товар принадлежит одной или нескольким категориям (отношение "многие ко многим").
- Один пользователь может сделать несколько заказов (отношение "один ко многим").
- Каждый заказ может включать несколько товаров (отношение "многие ко многим").

5. Документоориентированная модель

Предметная область: Управление проектами

Описание взаимоотношений:

- Проекты: Документы, содержащие информацию о проекте, такие как название, описание, дата начала и окончания.
- Задачи: Документы, связанные с проектами, содержащие информацию о назначении, статусе и дедлайне.
- Участники: Документы, представляющие участников проекта с их ролями и контактной информацией.

Взаимоотношения:

- Каждый проект может содержать несколько задач (отношение "один ко многим").
- Задача может быть назначена нескольким участникам (отношение "многие ко многим").

Задание № 2.2 "Преимущества и недостатки моделей данных"

№	Модель данных	Преимущества	Недостатки
1	Реляционная модель	Высокая гибкость в организации данных	Невозможность обработки сложных структур данных
		Использование стандартного языка запросов	Потребление ресурсов на нормализацию данных и выполнение сложных запросов
		Сильная теория и поддержка со стороны многих СУБД	Неэффективна для распределенных систем и больших объемов данных
2	Иерархическая модель	Простота в понимании и реализации.	Ограниченная гибкость в организации данных, трудности при изменении иерархии.
		Быстрый доступ к данным в иерархической структуре.	Невозможность представления сложных взаимосвязей (многие ко многим).
3	Сетевая модель	Гибкость в представлении сложных взаимосвязей между объектами.	Сложность в реализации и управлении структурой данных.
		Эффективное использование ресурсов, если данные имеют многоуровневую структуру.	Сложности в поддержке и обновлении из-за большого числа связей.
4	Объектно-ориентированная модель	Естественное моделирование реальных объектов и их взаимодействий.	Более сложная архитектура по сравнению с реляционными базами данных.
		Поддержка наследования и инкапсуляции, что делает код более используемым.	Нехватка стандартизированного языка запросов.
5	Документоориентированная модель	Гибкость в структуре данных, возможность хранить данные в их исходном формате (JSON, XML и др.).	Меньшая возможность выполнения сложных запросов по сравнению с реляционными СУБД.

		Хорошо подходит для хранения нестандартных и изменяющихся данных.	Отсутствие целостности данных, требуется ручное управление связями.
--	--	---	---