Самостоятельное задание по теме «Этапы проектирования БД. Модели данных»

ИСР 2.1

Для каждой модели и подходов к организации данных предложить соответствующею предметную область и описать взаимоотношение ее объектов:

1. **Иерархическая модель данных:** Иерархическая модель опирается на теорию графов. В основе иерархической модели данных лежит один главный элемент (главный узел), с которого все и начинается, такой элемент называет корневым элементом, в теории графов это называется корнем дерева. Каждый узел на более низком уровне связан только с одним узлом, находящимся на более высоком уровне. Иерархическое дерево имеет только одну вершину, не подчиненную никакой другой вершине и находящуюся на самом верхнем - первом уровне. Зависимые (подчиненные) узлы находятся на втором, третьем и т. д. уровнях. Количество деревьев в базе данных определяется числом корневых записей. К каждой записи базы данных существует только один иерархический путь от корневой записи.

Имеется спортивный клуб, где у каждого спортсмена свой тренер. У тренера может быть несколько спортсменов. Спортсмены участвуют в соревнованиях. Каждый спортсмен может участвовать во многих соревнованиях. Для автоматизации учета в спортивном клубе потребуются следующие записи:

* Спортсмен (id спортсмена, разряд, ФИО)
* Тренер (id тренера, разряд, ФИО)
* Соревнование (id соревнования, дата, название)

Отношения между записями соответствует связям между объектами реального мира. Например, отношение между объектом «тренер» и объектом «спортсмен» моделируется связью типа «один-ко-многим». Поэтому в записи «спортсмен» являются дочерними по отношению к записи «тренер». А вот между объектами «спортсмен» и «соревнование» в реальной жизни присутствует связь «много-ко-многим», потому что спортсмен может участвовать во многих соревнованиях, а в одном соревновании участвует много спортсменов. Единственный способ смоделировать его – это дублирование информации путем создания дополнительного дерева.

1. **Реляционная модель:** объекты и связи между ними представляет в виде таблиц, при этом связи тоже рассматриваются как объекты. Все строки, составляющие таблицу в реляционной базе данных, должны иметь первичный ключ. Все современные средства СУБД поддерживают реляционную модель данных.

В качестве примера рассмотрим две таблицы, которые небольшое предприятие использует для обработки заказов продукции. Первая таблица содержит информацию о заказчиках: каждая запись в ней включает в себя имя и адрес заказчика, платежные данные и информацию о доставке, номер телефона и т. д. Каждый элемент информации (атрибут) помещен в отдельный столбец базы данных, которому назначен уникальный идентификатор (ключ) для каждой строки. Во второй таблице—(с информацией о заказе) каждая—запись содержит идентификатор заказчика, совершившего заказ, название заказанного продукта, его количество, размер или цвет и т. д. Записи в этой таблице не содержат таких данных, как имя заказчика или его контактные данные.

У обеих таблиц есть только один общий элемент — идентификатор столбца (ключ). Благодаря наличию этого общего столбца реляционные базы данных могут устанавливать взаимосвязи между двумя таблицами. Когда приложение для обработки заказов передает заказ в базу данных, база данных обращается к таблице со сведениями о заказах, извлекает сведения о продукции и использует идентификатор заказчика из этой таблицы, чтобы найти сведения об оплате и доставке в таблице с информацией о нем. Затем на складе подбирают нужный продукт, заказчик своевременно получает свой заказ и производит оплату.

1. **Сетевая модель данных:** Разница между [иерархической моделью данных](https://ru.wikipedia.org/wiki/%D0%98%D0%B5%D1%80%D0%B0%D1%80%D1%85%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) и сетевой состоит в том, что в иерархических структурах запись-потомок должна иметь в точности одного предка, а в сетевой структуре данных у потомка может иметься любое число предков.

Сетевая БД состоит из набора экземпляров определенного типа записи и набора экземпляров определенного типа связей между этими записями.

Тип связи определяется для двух типов записи: предка и потомка. Экземпляр типа связи состоит из одного экземпляра типа записи предка и упорядоченного набора экземпляров типа записи потомка.

Примером может служить *Служба поиска информации*, которой пользуются члены парламента, где могут быть вызваны документы, относящиеся к какому-либо делу или имеющие определенную ссылку. Существует функция ключевого слова, позволяющая «помечать» некоторые слова в тексте, как ключевые. Операция вызова выведет названия тех документов, в которых присутствуют эти слова.

