Модели БД

1. Концептуальная модель БД. Документация, входная, выходная. Мат аппарат. Пример, расчет зарплаты: информация работника, тарифы, табель, выходные: платежные ведомости. Мат аппарат – формулы или алгоритмы расчета начислений или удержаний.
2. Инфологическая модель БД. В этом случае строится модель без привязки к каким-либо моделям, каким-либо СУБД и тд. Одним из вариантов – построение ЕР диаграммы, сущность-связь. Выходные данные: модель или диаграмма типа сущность-связь.
3. Проектирование дата-логической модели. Входные: сущность-связь. Определяется модель данных. Иерархическая, сетевая и тд. Нормализация БД или теория нормальных форм.

Выход: Нормализованная БД

1. Выбор целевой СУБД. Строится физическая модель базы-данных. На этом этапе учитываются особенности выбранной СУБД. Учитывается размещение данных на носителях.

Структура подразделяется:

* Рандомизированная.
* Детерменированная.

3 типа логических моделей(в зависимости от связей между данными):

* Иерархическая
* Сетевая
* Реляционная

Основные операции в иерархической модели

* Добавить в БД новую запись. Для корневой записи обязательно формирование значение ключа.
* Изменение значения данных предварительно извлеченной записи или считанной. Ключевые данные не изменяются -> ключ единожды присвоенный единой записи не меняется до тех пор, пока не удалена вся БД
* Удаление записи и все подчиненные ей. Т.е.
* Извлечение корневой записи по ключу. Допускается так же последовательный просмотр корневых записей.
* Извлечение следующей записи. Следующая запись извлекается в порядке левостороннего обхода дерева.

Достоинства:

* Связь в таких структурах именовать не требуется.
* Удобный для отображения связи один ко многим предметной области.

Недостатки:

* Сложность операций добавления и удаления объектов в бд, особенно связей.
* Сложность отображения связи многие ко многим.
* Сложность навигации с более низкого на более высокий.

Иерархическая модель данных

Данная модель реализует древовидную структуру. Дерево – совокупность корневого элемента и множества подчиненных ему элементов. Отношение между элементами носят подчиненный вертикальный характер. Горизонтальные связи не допускаются. В модели имеется один корневой узел (root). Остальные узлы называются порожденными и связаны между собой следующим образом: каждый узел имеет исходный, находящийся на вышестоящем уровне. На следующем уровне каждый узел может иметь более одного узла – потомка или вообще не иметь. Между исходными и порожденными узлами существует связь типа один ко многим.

Иерархическая модель данных должна удовлетворять следующим условиям:

* Иерархия имеет исходный узел или корень из которого строится дерево. Каждое дерево имеет только один корень.
* Узел имеет непустое множество атрибутов, которые описывают объект, моделируемый в данном узле.
* Порожденные узлы могут добавляться в дерево как вертикально, так и в горизонтальном направлении.
* Доступ к порожденным узлам возможен только через исходный узел, поэтому существует только один путь доступа к каждому узлу.
* Возможно использование нескольких экземпляров каждого узла, каждого уровня, при этом каждый экземпляр исходного узла начинает логическую запись.

СУБД IMS (Imformation Management System)

Сетевая модель

Является модификацией иерархической модели. Возможна горизонтальная связь между элементами модели. Появилось определение типа группа.

Элемент данных – наименьшая именованная единица

Агрегат – именованный набор элементов данных внутри записи.

Вектор – одномерная последовательность элементов данных, имеющих идентичные характеристики, например, массив или последовательность значений

Повторяющиеся группы – набор данных с различными характеристиками, которые могут многократно повторяться внутри записи.

Запись – экземпляр поименованного набора данных, состоящего из ноля или нескольких элементов или агрегатов данных.

Появление понятия ключ БД. Ключ служит для идентификации каждой записи.

Появилось понятие текущей записи. Указатель текущей записи.

Достоинства:

* Можно использовать для более широкого круга задач
* Возможность реализовать связь многие ко многим

Недостатки:

* Каждую связь нужно именовать.
* Сложность для восприятия достаточно разветвленной структуры БД.

СУБД IDMS (Integrated Database Management System)

Реляционная модель данных

Предложена в 70-х годах компанией IBM. В настоящее время является наиболее популярной, фактический стандарт. Данная модель является более строгой по причине того, что при работе используется специальный математический аппарат. Называется он реляционная алгебра, вычисления отношений. Основан на теории множеств. Она основана на понятии отношений в понятий математики. Основные элементы – двумерные отношения.

**Мощность** – количество строк в таблице

**Арность** – количество столбцов в таблице

Несколько атрибутов одного отношения или атрибуты разных отношений могут быть определены на одном и том же домене. Понятие домена имеет семантическую нагрузку. Т.е. данные можно считать сравнивыми только тогда, когда они относятся к одному домену.

Атрибут значение которого однозначно идентифицирует кортежи называется ключевым (ключом). Отношение может содержать несколько ключей. Один из них – первичный, остальные – возможные. Активен может быть только один ключ. При определении ключа создается индексный файл. Значение ключа должно быть уникально и обязательным (NOT NULL).

Внешний ключ определяет значение целостности, в соответствии с которым множество значений внешнего ключа является подмножеством значений первичного или уникального ключа родительской таблицы.

Считается, что целостность должна проверятся при:

* При добавлении записи в дочернюю таблицу, СУБД проверяет, что в родительской таблице есть запись с таким же значением первичного ключа.
* При удалении записи из родительской таблицы, СУБД проверяет, что в подчиненной таблице нет записей с таким же значением внешнего ключа.

Основные свойства отношений:

* Отсутствие кортежей-дубликатов.
* Отсутствие упорядоченности кортежей
* Отсутствие упорядоченности атрибутов.
* Атомарность значений атрибутов. Атрибуты не могут содержать множество.

Достоинства:

* Простота заключается в использовании двумерных таблиц.
* Гибкость. Операции реляционной алгебры позволяют резать и склеивать отношения
* Точность.
* Независимость данных. Обычно структура БД должна допускать возможность ее росту. Т.е. добавление новых атрибутов и отношений.
* Простота языка манипулирования данными.

Недостатки:

* Низкая в ряде случаев эффективность использования ресурсов вычислительной системы.
* Зависимость произвольной раннее заданной структуры от требований доступа к БД.
* Необходимость глубокого рассмотрения зависимостей между данными в силу специфики структуры нормализованного отношения.
* Сложность специфического представления данных.

В реляционных СУБД используется 2 механизма манипулирования:

* Реляционная алгебра. Основывается на классической теории множеств. Операторы – реляционные отношения, результат – реляционные отношения. Существует 5 основных операций
  + **Проекция**
  + **Селекция**
  + **Декартово произведение**
  + **Разность**
  + **Объединение**
  + *Соединение*
  + *Пересечение*
  + *Деление*

Для представления каждой операции используется терминология введенная Коддом.

* Исчисление отношений. На классическом логическом аппарате исчисления предикатов первого порядка.