Вопрос 4: Ранние подходы к организации БД.

Подходы исторически предшествовали реляционным. На основе методов, использованных в них организованы и реляционные. Знания о них полезны для теории.

**Общие характеристики:**

* **Использовались задолго до реляционных;**
* **БД накапливались громадные;**
* **Их надо использовать совместно с реляционными;**
* **Не строились на абстрактных моделях;**
* **В них был интерактивный доступ на уровне записей, что требовало писать прикладные программы со своим интерфейсом.**

1. Системы, основанные на инвертированных списках

Дата появления – конец 1960-х

**Представители: Datacom/Db(1972 г.), Adabas(1971 г.)**

Организация доступа к данным применяется в современных БД на основе индексов(инвертированных списокв).

**Структура данных похожа на реляционную, но таблицы и пути доступа к ним видны пользователям. Строки таблиц упорядочены в некой физической последовательности. Эта упорядоченность может определяться и для всей БД. Может быть определено количество ключей поиска**

**Индексы автоматически поддерживаются и видны пользователям**

**Манипулирование данными реализовано двумя классами операторов - операторы, устанавливающие адрес записи (прямые поисковые операторы, относительные поисковые операторы), и операторы работы над записями.**

**Правил ограничения целостности тогда не было, СУБД это не делала, этим занимался программист**

1. Иерархические СУБД

Дата появления – 1968 г.

**Представители: IMS (Information Management System, 1968 г.), Caché(1997 г.)**

**Такие СУБД состоят из упорядоченного набора деревьев. Дерево состоит из одной корневой записи и упорядоченного набора поддеревьев. Каждое дерево - иерархически организованный набор записей. Запись-потомок должна иметь одного родителя. Все записи с общим родителем называются близнецами. Использовалось понятие сегмент вместо записи. Запись базы данных - всё дерево сегментов**

**Манипуляции данными:**

**1. Найти указанное дерево**

**2. Перейти от одного дерева к другому**

**3. Перейти от одной записи к другой внутри дерева**

**4. Вставить запись**

**5. Удалить запись**

**Задача управлением целостностью решалась наблюдением за записями между предками и потомками - никакой потомок не может существовать без родителя**

Общую схему иерархической модели можно увидеть на рис. ?

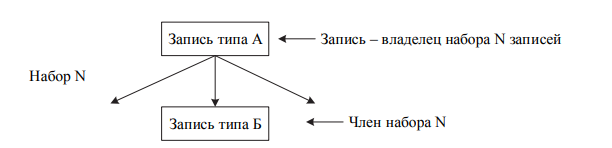


Рис. ?. Общая схема иерархической модели.

1. Сетевые

Появление – 1971 г.

**Представители: CODASYL(1971 г.), DMS-90(1979 г.)**

**Структура данных - расширяет иерархический: потомок может иметь любое число предков**

**Сетевая БД состоит из предка и потомка и записей между ними. Для записей предка с потомком выполняются условия:**

**1. Каждый предок является таковым только в одной связи**

**2. Каждый потомок является таковым не более, чем в одной**

**Манипуляции данными:**

**1. Найти конкретную запись**

**2. Перейти от предка к первому потомку по некоторой связи**

**3. Перейти к следующему потомку**

**4. Включить связь**

**5. Исключить из связи**

Пример схемы БД можно увидеть на рис. ?.

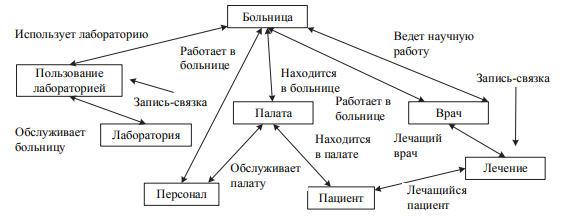


Рис. ?. Схема сетевой базы данных

**Плюсы:**

1. **Возможность построения в ручную эффективных программ**
2. **Возможность экономии памяти засчет разделения объектов хранения.**
3. **Имеют развитые способы управления данными в ПЗУ на низком уровне**

**Минусы:**

**1. Слишком сложно пользоваться**

**2. Необходимо знание физической организации**

**3. От физ-организации зависят прикладные программы**

**4. Логика программы перегружена логикой доступа к БД**

Вопрос 5: Реляционный подход к организации БД. Базовые понятия реляционных баз данных.

Реляционное (relational) отношение. Являются наиболее распространёнными сегодня.

**Плюсы**

* **Наличие небольшого набора абстракций, на основе которых просто моделировать распространённые предметные области;**
* **Наличие простого, но мощного математического аппарата из теории множеств и мат. логики;**
* **Возможность манипулирования данными без необходимости знаний о конкретной физической организации данных во внешней памяти.**

**Минусы**

* **Ограниченность в нетрадиционных областях применения;**
* **Реляционная модель данных не позволяет определить набор операций, связанных с данными определенного типа, что часто является естественным требованием при моделировании данных со сложной структурой. Операции приходится задавать в конкретном приложении;**
* **Реляционная модель не позволяет рассматривать данные послойно, на различных уровнях абстракции, при необходимости отвлекаясь от ненужных деталей.**

**Применение**

**В системах организации, проектирования, машиностроения**

**Понятия**

**Тип данных - на ряду с традиционными (текстовые, символьные, числовые, дата), есть и специальные (домен), схожие с подтипами основных типов данных**

**Домен - допустимое потенциальное множество значений типа**

**Атрибут - Столбец в таблице характеризует одна из характеристик хранения**

**Схема отношения - именованое множество пар имя\_атрибута=имя\_типа**

**Арность (arity) схем отношения - мощность множества**

**Схема БД - набор именованных схем отношений**

**Кортеж - множество пар имя\_атрибута=значение, которое содержит одно вхождение каждого именти атрибута, принадлежащего схеме отношения**

**Отношение - множество кортежей, соответствующих одной схеме отношения. Различают отношение-схему и отношение-экземпляр**

**Реляционная БД - отношения, имена которых совпадают с именами схем отношений в схеме базы данных**

Вопрос 6: Фундаментальные свойства отношений. Реляционная модель данных.

**Фундаментальные свойства отношений;**

* **Значения атрибута должны быть атомарны;**
* **Так как домен это потенциальное множество значений простого типа данных, то среди значений домена не могут содержаться множества значений;**
* **Отсутствие кортежей дупликатов. Свойство следует из определения. В классической теории множеств каждое множество состоит из различных элементов. Это свойство требует наличия уникальных идентификаторов (первичных ключей) у каждого отношения, определяющих их кортеж, и альтернативных ключей;**
* **Отсутствие упорядоченности кортежей. Это следует из определения: "Отношение это множество кортежей";**
* **Отсутствие упорядоченности атрибутов.**

Общая характеристика реляционной модели данных

Кристофер Дейт - основатель модели

**Состоит из трёх частей**

1. **Структурная**

**Говорит о том, что единственной структурой данных, используемой в реляционных БД, является нормализованное n-арное отношение**

1. **Манипуляционная**

**Утверждает два фундаментальных механизма манипулирования данными - реляционная алгебра и реляционные исчисления (классический логический аппарат исчисления предикатов первого порядка).**

**Обеспечение меры реляционности любого конкретного языка реляционных баз данных.**

**Язык называется реляционным не меньшей выразительностью и мощностью, чем реляционная алгебра или реляционные исчисления**

**3. Целостная**

**Фиксируется два базовых требования целостности, которые должны поддерживаться в любой реляционной СУБД**

**1. Целостность сущностей - любой кортеж любого отношения отличим от любого другого кортежа этого отношения. Решается введением первичного ключа**

**2. Целостность по ссылкам - при соблюдении нормализованности отношений сложности сущности реального мира представляются в реляционной БД в виде нескольких кортежей нескольких отношений**

**Базисные средства манипулирования реляционными данными**

**Основные алгебраические операции**

**Теоретико-множественные операции (объединение множеств, объединение отношений, пересечение отношений, разность отношений, произведение отношений)**

**Специальные реляционные операции (ограничение отношения, проекция отношения, соединение отношения, деление отношений)**

Также в состав алгебры включены: операция присваивания, операция переименования аттрибутов