История развития СУБД

Области использования ВТ

- Численные расчеты
- Характерной особенностью данной области применения вычислительной техники является
- 1. наличие сложных алгоритмов обработки,
- 2. простые по структуре данные,
- 3. объем данных сравнительно невелик.

2-ая область применения

- Автоматизированные информационные системы
- Особенности :
- Большие объемы информации,
- Сложную структура данных

- История развития СУБД насчитывает более 30 лет.
- В 1968 году была введена в эксплуатацию первая промышленная СУБД система IMS фирмы IBM.
- В 1975 году появился первый стандарт ассоциации по языкам систем обработки данных - Conference of Data System Languages (CODASYL),

Три основных этапа развития

- Начальный этап был связан с созданием первого поколения СУБД, опиравшихся на иерархическую и сетевую модели данных
- Создание реляционной модели данных
- Третье поколение СУБД – распределенные, объектно-ориентированные СУБД

Начальный этап

- К сожалению, СУБД первого поколения были в подавляющем большинстве закрытыми системами:
- отсутствовал стандарт внешних интерфейсов,
- не обеспечивалась переносимость прикладных программ,
- не обладали средствами автоматизации программирования
- они были очень дороги.
- Функции управления распределением ресурсов в основном осуществляются операционной системой (ОС),
- Поддерживаются языки низкого уровня манипулирования данными, ориентированные на навигационные методы доступа к данным.
- Значительная роль отводится администрированию данных.

Создание реляционной модели данных

- Простота и гибкость модели привлекли к ней внимание разработчиков и снискали ей множество сторонников.

- Второй этап характеризовали две основные особенности –
- реляционная модель данных
- язык запросов SQL.

Эпоха персональных компьютеров

- Все СУБД были рассчитаны на создание БД в основном с монопольным доступом.
- Большинство СУБД имели развитый и удобный пользовательский интерфейс.
- Во всех настольных СУБД поддерживался только внешний уровень представления реляционной модели, то есть только внешний, табличный вид структур данных.
- При наличии высокоуровневых языков манипулирования данными т~ипа реляционной алгебры и SQL в настольных СУБД поддерживались низкоуровневые языки манипулирования данными на уровне отдельных строк таблиц.
- В настольных СУБД отсутствовали средства поддержки ссылочной и структурной целостности базы данных.

Распределенные базы данных

- Практически все современные СУБД обеспечивают поддержку полной реляционной модели, а именно:
- структурной целостности — допустимыми являются только данные, представленные в виде отношений реляционной модели;
- языковой целостности, то есть языков манипулирования данными высокого уровня (в основном SQL);
- ссылочной целостности, контроля за, соблюдением ссылочной целостности в течение всего времени функционирования системы, и гарантий не возможности со стороны СУБД нарушить эти ограничения,
- Большинство современных СУБД рассчитаны на многоплатформенную архитектуру.

Системы, основанные на инвертированных списках, иерархические и сетевые СУБД.

Сильные места и недостатки ранних систем

Общие характеристики

- Эти системы активно использовались в течение многих лет, дольше, чемиспользуется какая-либо из реляционных СУБД.
- Все ранние системы не основывались на каких-либо абстрактных моделях.
- В ранних системах доступ к БД производился на уровне записей. Пользователи этих систем осуществляли явную навигацию в БД, используя языки программирования, расширенные функциями СУБД.
- Интерактивный доступ к БД поддерживался только путем создания соответствующих прикладных программ с собственным интерфейсом.
- После появления реляционных систем большинство ранних систем было оснащено "реляционными" интерфейсами.

Основные особенности систем, основанных на инвертированных списках

Структуры данных

- Строки таблиц упорядочены системой в некоторой физической последовательности.
- Физическая упорядоченность строк всех таблиц может определяться и для всей БД так делается, например, в Datacom/DB).
- Для каждой таблицы можно определить произвольное число ключей поиска, для которых строятся индексы.
- Эти индексы автоматически поддерживаются системой, но явно видны пользователям.

Манипулирование данными

- Операторы, устанавливающие адрес записи, среди которых:
- прямые поисковые операторы (например, найти первую запись таблицы по некоторому пути доступа);
- операторы, находящие запись в терминах относительной позиции от предыдущей записи по некоторому пути доступа.
- операторы над адресуемыми записями

Типичный набор операторов:

- LOCATE FIRST - найти первую запись таблицы T в физическом порядке; возвращает адрес записи;
- LOCATE FIRST WITH SEARCH KEY EQUAL - найти первую запись таблицы T с заданным значением ключа поиска K; возвращает адрес записи;
- LOCATE NEXT - найти первую запись, следующую за записью с заданным адресом в заданном пути доступа; возвращает адрес записи;
- RETRIVE - выбрать запись с указанным адресом;
- UPDATE - обновить запись с указанным адресом;
- DELETE - удалить запись с указанным адресом;
- STORE - включить запись в указанную таблицу; операция генерирует адрес записи.

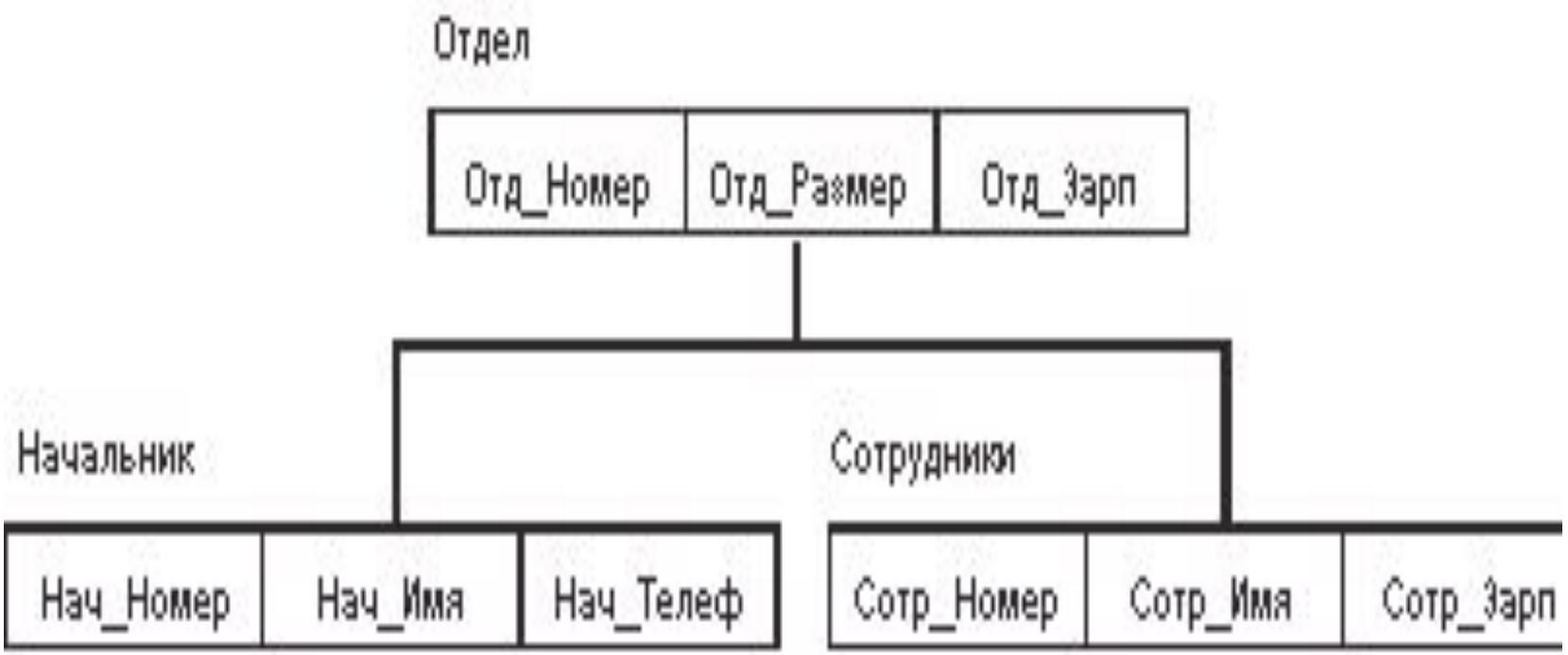
Ограничения целостности

- Общие правила определения целостности БД отсутствуют. В некоторых системах поддерживаются ограничения уникальности значений некоторых полей, но в основном все возлагается на прикладную программу.

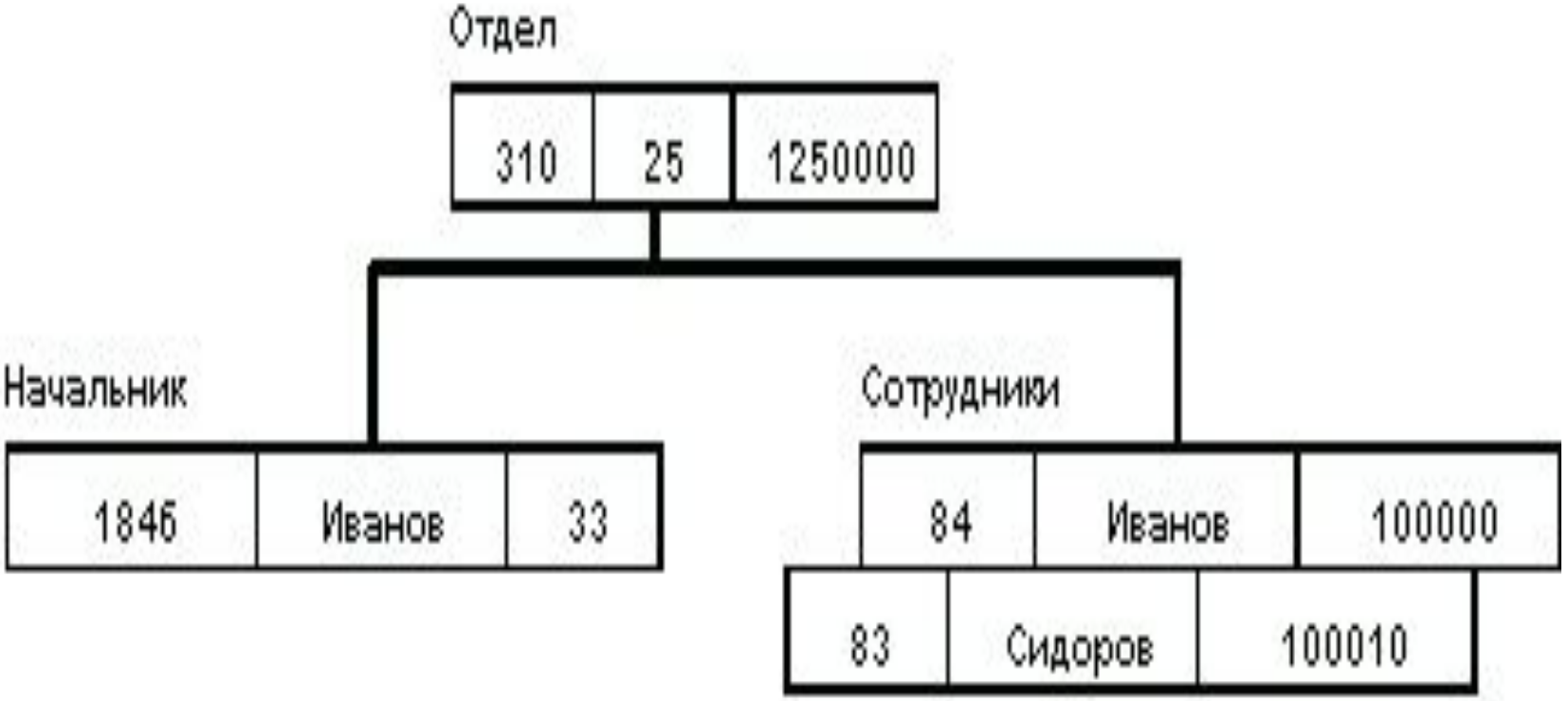
Иерархические модели данных

Иерархические структуры данных

- Иерархическая БД состоит из упорядоченного набора деревьев; более точно, из упорядоченного набора нескольких экземпляров одного типа дерева.



Один экземпляр дерева



Манипулирование данными

- Примерами типичных операторов манипулирования иерархически организованными данными могут быть следующие:
- Найти указанное дерево БД (например, отдел 310);
- Перейти от одного дерева к другому;
- Перейти от одной записи к другой внутри дерева (например, от отдела - к первому сотруднику);
- Перейти от одной записи к другой в порядке обхода иерархии;
- Вставить новую запись в указанную позицию;
- Удалить текущую запись.

Ограничения целостности

- Автоматически поддерживается целостность ссылок между предками и потомками.
- **Основное правило:** *никакой потомок не может существовать без своего родителя.*

Сетевые модели данных

- Структуры данных

Тип связи

- 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.
- Для данного типа связи L с типом записи предка P и типом записи потомка C должны выполняться следующие два условия:
- Каждый экземпляр типа P является предком только в одном экземпляре L;
- Каждый экземпляр C является потомком не более, чем в одном экземпляре L.

Пример сетевой схемы БД:

Манипулирование данными

- Найти конкретную запись в наборе однотипных записей (инженера Сидорова);
- Перейти от предка к первому потомку по некоторой связи (к первому сотруднику отдела 310);
- Перейти к следующему потомку в некоторой связи (от Сидорова к Иванову);
- Перейти от потомка к предку по некоторой связи (найти отдел Сидорова);
- Создать новую запись;
- Уничтожить запись;
- Модифицировать запись;
- Включить в связь;
- Исключить из связи;
- Переставить в другую связь и т.д.

Ограничения целостности

- В принципе их поддержание не требуется, но иногда требуют целостности по ссылкам
- (как в иерархической модели).