

# *Модели данных в информационных системах*

Иван Вадимович Саинский  
доц. каф. ИНИТ

Информатика, 2009



# Темы лекции

- Основные модели хранения данных
- Достоинства и недостатки моделей
- Особенности использования моделей при решении задач



# Основные модели хранения данных

**Модель данных** – совокупность структур данных и операций их обработки.

**1. Модель на инвертированных списках**

**2. Иерархическая модель данных**

**3. Сетевая модель данных**



# Системы на инвертированных списках

**Примеры систем:** *Datacom/DB* компании *Applied Data Research, Inc. (ADR)*, *Adabas* компании *Software AG*.

## Структуры данных

- *БД - совокупность таблиц с данными и индексный файл – инвертированные списки. Таблицы и пути доступа к ним видны пользователям. При этом:*
- *Строки таблиц упорядочены системой в некоторой физической последовательности.*
- *Для каждой таблицы можно определить произвольное число ключей поиска, для которых строятся индексы. Эти индексы автоматически поддерживаются системой, но явно видны пользователям.*



# Системы на инвертированных списках

## Манипулирование данными

### Поддерживаются два класса операторов:

- Операторы, устанавливающие адрес записи: прямые и относительные поисковые операторы
- Операторы над адресуемыми записями

### 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.



# Системы на инвертированных списках

## 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;
- LOCATE NEXT WITH SEARCH KEY EQUAL**;
- LOCATE FIRST WITH SEARCH KEY GREATER**;
- 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.



# Иерархическая модель данных

**Примеры систем:** Information Management System (IMS) фирмы IBM. Первая версия появилась в 1968 г.

## Структура данных



# Иерархическая модель данных

## Типичный набор операторов

- 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. Поддержание целостности по ссылкам между записями, не входящими в одну иерархию не поддерживается.





# Сетевые системы

**Примеры систем:** *Integrated Database Management System (IDMS)* компании *Cullinet Software, Inc.* .

## Структуры данных



**Ограничения целостности:** *Поддержание не требуется, но иногда требуют целостности по ссылкам (как в иерархической модели).*



# Сетевые системы

## Типичный набор операторов

- **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.**

