

Лекция 3. Часть 3.

**Логическое проектирование БД.  
Типовые модели данных.**

## Второй этап проектирования БД

- **Сущности** концептуальной модели описываются как записи, состоящие из полей.
- Каждый **атрибут** описывается как поле с типом и характеристиками, возможными в выбранной СУБД.
- **Связи** концептуальной модели описываются в понятиях, соответствующих выбранной СУБД.
- Определяется порядок реализации запросов пользователей к базе данных с помощью типовых операций СУБД и т.д.

## Модель данных СУБД

- Определены типы и характеристики логических структур данных (полей, записей, файлов).
- Заданы правила составления структур более общего типа из структур более простых типов (например, записей из полей, файлов из записей и т.д.).
- Определен способ представления связей (отношений) между файлами и записями:
  - основные элементарные операции над данными;
  - обобщенные операции (хранимые процедуры);
  - средства контроля условий корректности изменения данных (ограничения на основе триггеров).

Основные элементарные операции: поиск записи с заданным значением ключа, чтение нужной записи, добавление записи, корректировка, удаление (**CRUD** – Create, Read, Update, Delete).

# Эволюция моделей данных

**Связь** - ключевое слово, отличающее базу данных от простого файла или набора файлов. Как представить эту связь в базе данных? В разное время применялись различные подходы (модели данных).

<b>Годы</b>	<b>Используемые модели данных</b>
<b>1960-е</b>	Сетевая и иерархическая ( <i>навигационный подход, императивное программирование</i> )
<b>1970-е</b>	1. Сетевая и иерархическая 2. Реляционная ( <i>математическая база, декларативное программирование</i> )
<b>1980-е</b>	1. Реляционная 2. Сетевая и иерархическая
<b>1990-е</b>	Реляционная
<b>2000-е</b>	1. Реляционная 2. NoSQL

## Первые СУБД



- Начало 1960-х годов
- Чарльз Бахман (работал в General Electric)
- СУБД Integrated Data Store (затем IDMS, работавшая на мейнфреймах IBM).

Получил за нее премию Тьюринга в 1973 году.

Впервые доступ к данным осуществлялся не непосредственно, как в файловой системе, а через описание данных в виде иерархической системы «указателей», помогающих осуществить доступ к конкретной ячейке памяти.

## Первые СУБД



СУБД Бахмана строились с использованием иерархических и сетевых моделей и были названы им **навигационными**, поскольку для перемещения по записям использовались «указатели» или «пути».

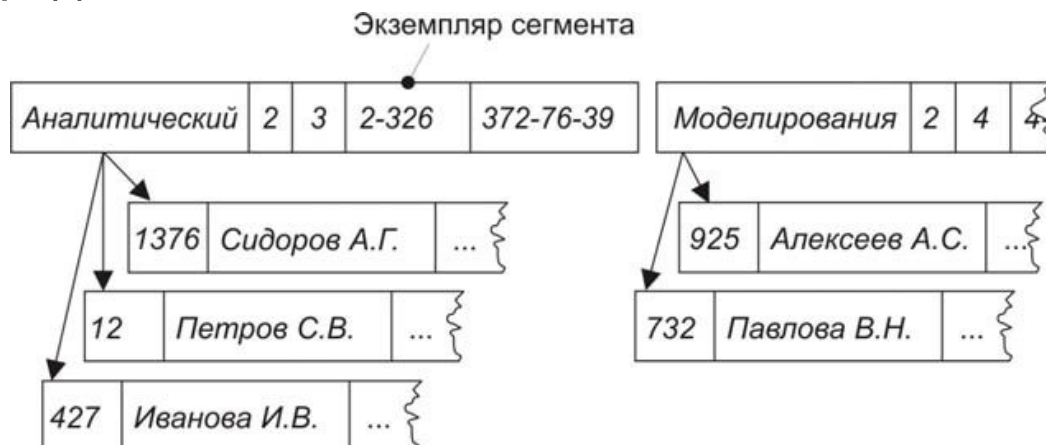
Около 10 лет все программы, работающие с данными, писали в основном с использованием навигационного подхода на языке COBOL, в который были интегрированы команды для работы с БД.

# Иерархическая модель данных

**Сегмент** – именованная совокупность полей.



**Экземпляр сегмента** – логическая запись, описывающая конкретный объект предметной области.



**Тип сегмента** (или просто **сегмент**) – это структурная схема такой записи, в которой указаны имена составляющих ее полей

# Схема иерархической базы данных

Дерево, вершинами которого являются типы сегментов.



**Иерархическая база данных** – это совокупность деревьев, корнями которых являются экземпляры корневого типа сегмента.

Каждое такое дерево называется **набором** и образует **физическую запись** иерархической базы данных.

От записи к записи можно перемещаться, только если они объединены набором.



## Типичные операторы для манипулирования данными

- Найти указанное дерево БД (например, аналитический отдел);
- 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.

## Иерархическая модель данных: резюме

- Данные в иерархической БД упорядочены: всегда есть первая запись и список следующих и предыдущих.
- В БД хранятся не только данные, но и связи между ними (наборы). Наборы – существенный и обязательный объект, т.к. от записи к записи можно перемещаться, только если они объединены набором.
- Навигационная БД предполагает исключительно процедурный стиль программирования – разработчик сам определяет алгоритм получения интересующих его данных в рамках существующих в БД связей; изменятся связи – перестанут работать программы.

# Иерархическая модель данных: плюсы и минусы

## Сильные места ранних СУБД:

- Развитые средства управления данными во внешней памяти на низком уровне;
- Возможность построения вручную эффективных прикладных систем;
- Возможность экономии памяти за счет разделения подобъектов (в сетевых системах).

# Иерархическая модель данных: плюсы и минусы

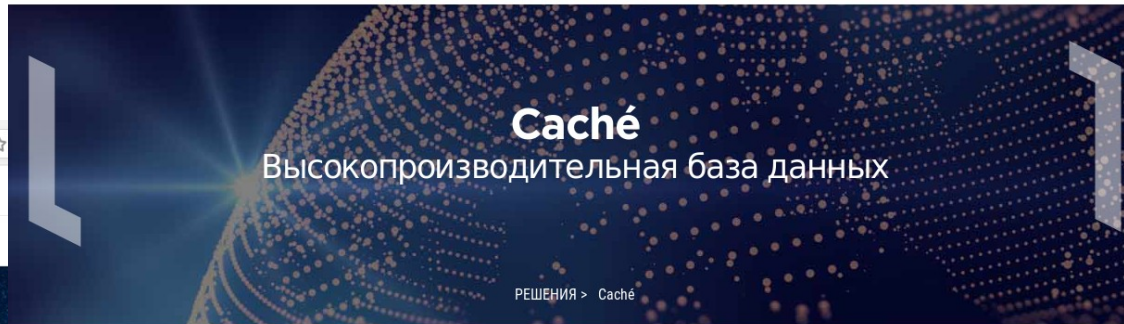
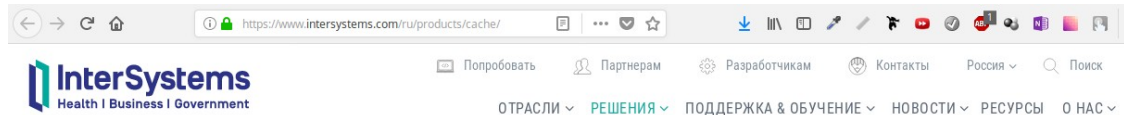
## Сильные места ранних СУБД:

- Развитые средства управления данными во внешней памяти на низком уровне;
- Возможность построения вручную эффективных прикладных систем;
- Возможность экономии памяти за счет разделения подобъектов (в сетевых системах).

## Недостатки:

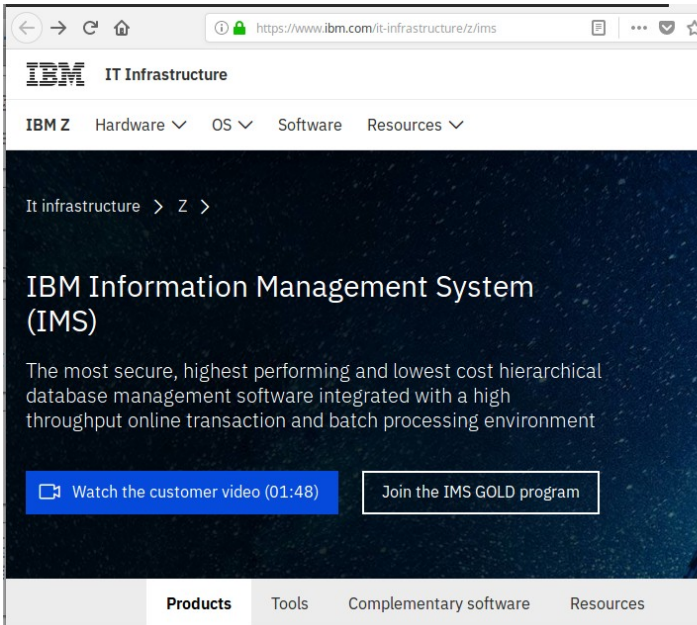
- Слишком сложно пользоваться (требуется написание большого количества кода, возрастает вероятность допустить ошибку);
- Фактически необходимы знания о физической организации данных;
- Прикладные системы зависят от этой организации, их логика перегружена деталями организации доступа к БД.

# Иерархические БД сегодня



База данных для критически важных приложений

InterSystems Cache® — это высокопроизводительная база данных, обеспечивающая поддержку транзакционных приложений по всему миру. Она используется для решения любых задач, будь то создание карты Млечного Пути с миллиардами звезд, обработка миллиардов трейдинговых операций ежедневно или управление интеллектуальными энергетическими сетями.



IBM Information Management System (IMS)™ and the IMS tools portfolio provide industrial strength capabilities for both managing and distributing data. IMS delivers the highest levels of availability, performance, security and scalability for OLTP in the industry.

IMS is used by many of the Fortune 1000 companies worldwide. Collectively these companies process more than **265 billion transactions per day** in IMS – securely.