Database Management Systems

Module 04: Introduction to DBMS/1

Partha Pratim Das

Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

ppd@cse.iitkgp.ac.in

Module Recap

- Comparison of data management using Python & files and DBMS
- Efficacy and Efficient DBMS highlighted

Module Objectives

- To familiarize with the basic notions and terminology of database management systems
- To understand the role of data models and languages
- To understand the approaches to database design

Module Outline

- Levels of Abstraction
- Schema & Instance
- Data Models
  - Relational Databases
- DDL & DML
- SQL
- Database Design

Levels of Abstraction

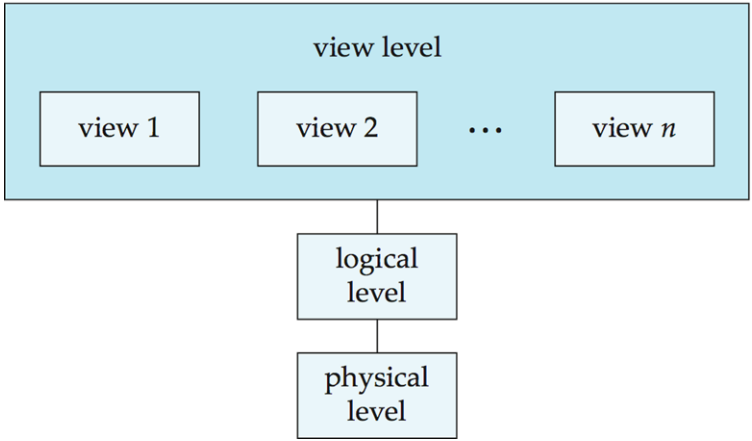
Levels of Abstraction

- **Physical level:** describes how a record (for example, instructor) is stored
- **Logical level:** describes data stored in database, and the relationships among the data fields

  type *instructor* = **record**
  ID : string;
  name : string;
  dept_name : string;
  salary : integer;
  end;
- **View level:** application programs hide details of data types
  - Views can also hide information (such as an employee's salary) for security purposes

View of Data

An architecture for a database system



Schema and Instance

- Similar to type of a variable and value of the variable at run-time in programming languages
- **Schema**
  - **Logical Schema** – the overall logical structure of the database
    - Analogous to type information of a variable in a program
    - Example: The database consists of information about a set of customers and accounts in a bank and the relationship between them
    - Customer Schema

Name	Customer ID	Account #	Aadhaar ID	Mobile #

    - Account Schema

Account #	Account Type	Interest Rate	Min. Bal.	Balance

  - **Physical Schema**– the overall physical structure of the database

- **Instance**
  - The actual content of the database at a particular point in time
  - Analogous to the value of a variable
  - Customer Instance

Name	Customer ID	Account #	Aadhaar ID	Mobile #
Pavan Laha	6728	917322	182719289372	9830100291
Lata Kala	8912	827183	918291204829	7189203928
Nand Prabhu	6617	372912	127837291021	8892021892

  - Account Instance

Account #	Account Type	Interest Rate	Min. Bal.	Balance
917322	Savings	4.0%	5000	7812
372912	Current	0.0%	0	291820
827183	Term Deposit	6.75%	10000	100000

Schema and Instances

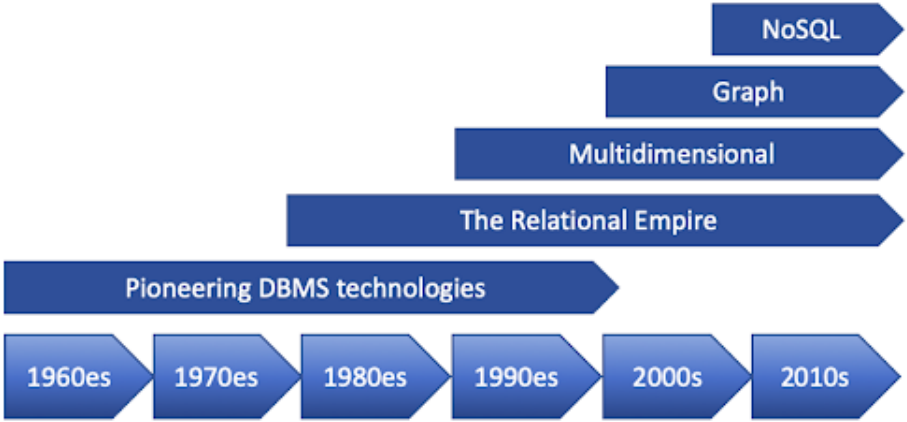
- **Physical Data Independence** – the ability to modify the physical schema without changing the logical schema
  - Analogous to independence of *Interface* and *Implementation* in Object-Oriented Systems
  - Applications depend on the logical schema
  - In general, the interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others.

Data Models

Data Models

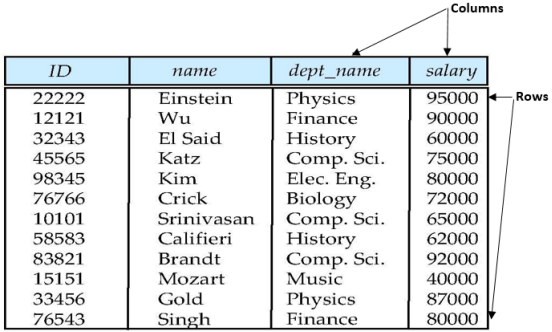
- A collection of tools for describing
  - Data
  - Data relationships
  - Data semantics
  - Data constraints
- **Relational model** (we focus in this course)
- Entity-Relationship data model (mainly for database design)
- Object-based data models (Object-oriented and Object-relational)
- Other older models
  - Network model
  - Hierarchical model
- Recent models for Semi-structured or Unstructured data
  - Converted to easily manageable formats
  - Content Addressable Storage (CAS) with metadata descriptors
  - XML format.
  - RDBMS which supports BLOBs

Data Models (2)



Relational Model

- All the data is stored in various tables
- Example of tabular data in the relational model



ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

A Sample Relational Database

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

DDL and DML

Data Definition Language (DDL)

- Specification notation for defining the database schema
  - Example:
    create table *instructor* (
    ID **char**(5),
    name **varchar**(20),
    dept_name **varchar**(20),
    salary **numeric**(8,2))
- DDL compiler generates a set of table templates stored in a *data dictionary*
- Data dictionary contains metadata (that is, data about data)
  - Database schema
  - Integrity constraints
    - Primary key (ID uniquely identifies instructors)
  - Authorization
    - Who can access what

Data Manipulation Language (DML)

- Language for accessing and manipulating the data organized by the appropriate data model
  - DML: also known as **Query Language**
- Two classes of languages
  - **Pure** – used for proving properties about computational power and for optimization
    - **Relational Algebra** (we focus in this course)
    - Tuple relational calculus
    - Domain relational calculus
  - **Commercial** – used in commercial systems
    - SQL is the most widely used commercial language

SQL

SQL

- The most widely used commercial language
- *SQL is NOT a Turing Machine equivalent language*
  - Cannot be used to solve all problems that a C program, for example, can solve
- To be able to compute complex functions, SQL is usually embedded in some higher-level language
- Application programs generally access databases through one of
  - Language extensions to allow embedded SQL
  - Application Programming Interface or API (for example, ODBC/JDBC) which allow SQL queries to be sent to a database

Database Design

Database Design

The process of designing the general structure of the database:

- **Logical Design** – Deciding on the database schema. Database design requires that we find a **good** collection of relation schema
  - Business decision
    - What attributes should we record in the database?
  - Computer Science decision
    - What relation schemas should we have and how should the attributes be distributed among the various relation schemas?
- **Physical Design** – Deciding on the physical layout of the database

Database Design (2)

- Is there any problem with this relation?

ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

Module Summary

- Familiarized with the basic notions and terminology of database management systems
- Introduced the role of data models and languages
- Introduced the approaches to database design

Slides used in this presentation are borrowed from http://db-book.com/ with kind permission of the authors.
Edited and new slides are marked with "PPD".