3005 Final Full Notes

William Findlay

April 17, 2018

Contents

1 Definitions

1.1 Database Terms

- **Database**
  - a collection of related data stored on a computer
- **Data**
  - a value which represents known facts with an implicit meaning
- **Mini world**
  - some part of the real world which is represented by the data stored in the database
- **Database management system** (DBMS)
  - software to facilitate creation and maintenance of a database
- **Database system**
  - database and...
  - the application programs developed on top of the DBMS

1.2 Actors

1.2.1 Behind the Scenes

- **System designer**
  - design and implement DBMS modules
- **Tool developer**
  - design and implement tools
    - modeling
    - designing
    - performance monitoring
    - prototyping
    - test data generation
    - UI creation
    - simulation
- **Operator and maintenance personnel**
  - tunnel rats
  - manage the running and maintenance of the DB

1.2.2 On the Scene

- **DBA** (database administrator)
  - acquire software and hardware resources
  - control the use of those resources
  - monitor efficiency
  - monitor use of DB
  - authorize access to DB
- **DB designer**
  - define the following aspects of a DB:
    - structure
    - constraints
    - content
    - transactions
  - must understand end users' needs
- **System analyst**
  - design applications and canned transactions for a DB

- **Application developer**
  - implement the specifications developed by analysts
- **End user**
  - use DB day-to-day
  - don't know or care how DB is structured
  - two categories:
    - naïve users
    - business analysts

1.3 Data Models

- **Data model**
  - way of representing data in a meaningful way
  - how data is *structured* and *operated*
  - three parts:
    - concepts to describe structure
    - operations for manipulating structures
    - constraints which must be obeyed
  - **entity relationship model**
    - entities connected by relationships
  - **hierarchical model**
    - tree-like structure
    - group by records and links
    - navigational and procedural operations
  - **network model**
    - network structure
    - grouped by records and links
    - navigational and procedural operations
  - **relational model**
    - tables
    - tuples in relations
    - declarative operations
- **Constructs**
  - a data model concept which defines the structure of the DB
  - elements and their types
  - groups of elements
  - relationships between such groups
- **Operations**
  - basic model operations
    - `insert`
    - `delete`
    - `update`
    - `query`
  - user-defined operations
    - `compute_gpa`
    - `update_inventory`
- **Constraints**
  - specify restrictions on the data
- **Physical data model**
  - low level
  - describe how data is stored physically
- **Conceptual data model**
  - high level

  - how the user will perceive data
  - how the user will access/modify data
- **Implementation data model**
  - somewhere between physical and conceptual
  - the sum of those two parts
- **Self-describing data model**
  - description of the data is combined with its values
- **Database schema**
  - description of data at some abstraction level
  - just the relations and attribute names
  - also called **intension**
- **Database instance**
  - a snapshot of the data at a given point in time
  - relations, attribute names, tuples
  - also called **extension**

1.4 Database Languages

- **DDL** (data definition language)
  - add or remove data
- **DML** (data manipulation language)
  - change data
- **QL** (query language)
  - query data

1.5 Relational Database Definitions

- **Schema of a relation**
  - denoted by $R(A_1, A_2, ..., A_n)$
  - R is the **name**
  - $A_1, A_2, ..., A_n$ are the **attributes**
- **Tuple**
  - ordered set of values
  - written : $< V_1, V_2, ..., V_n >$
    - each value V_n is derived from an appropriate domain
  - an *n-tuple* is a tuple with n values
- **Domain**
  - three parts:
    - name
    - data type
    - set of **atomic** values (indivisible values)
- **Attribute**
  - attribute name designates a role played by a domain in a relation
  - can be the same as a domain name
    - e.g., a user-defined type `Name` which is the domain of an attribute also called `Name`
- **Cartestian product**
  - let $D_1, D_2, ..., D_n$ be a set of n domains
  - cartesian product on $D_1, D_2, ..., D_n$ is
    - $\{< d_1, d_2 > | \ d_1$ in D_1, d_2 in $D_2\}$

2 Intro

2.1 Types of Database

- We are only concerned with **traditional applications**
- Business Data Processing (Numeric and Textual)

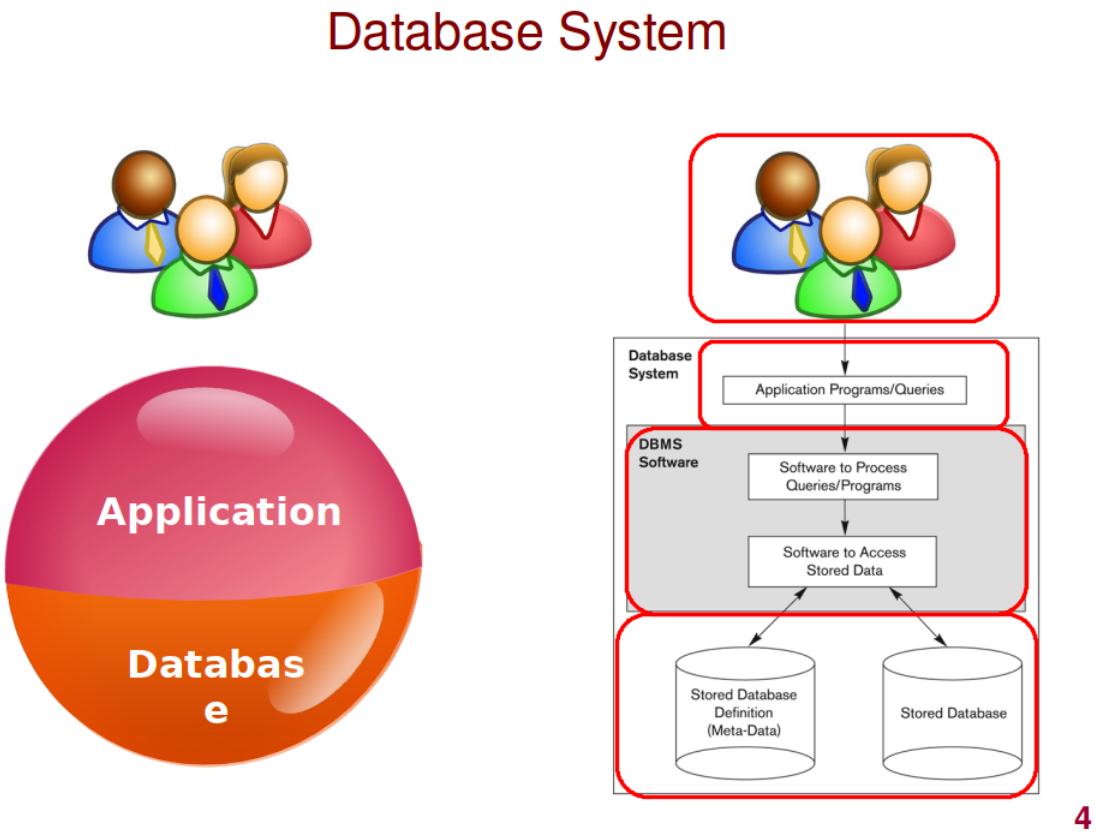

Figure 1: Database system diagram from Mengchi's slides.

2.2 DBMS Functionality

- **Load** initial database contents on a secondary storage medium
- **Define** a database in terms of:
  - data types
  - data structures
  - constraints
- **Manipulate** the database
  - retrieve
    - query
    - generate reports
  - modify
    - insert
    - delete
    - update

    - access
        * through web applications which provide a graphical front end
- **Handle concurrency** from multiple users
- **Security measures** to restrict unauthorized access
- **Presentation and visualization** of data
- **Maintenance** of database and application programs

2.3 Application/Database Interaction

- **Queries**
    - access data according to specifications and return a result
- **Transactions**
    - read data and update
    - store new data
- **No unauthorized access**
- Keep up with changing user requirements

2.4 Characteristics of the Database Approach

- **Self-Describing**
    - **catalog** stores descriptions of a database
        * data structures
        * data types
        * constraints
    - the description is called **meta-data**
    - allows the DBMS to work with many different applications
- **Insulation**
    - we can change the way the data is structured and organized without changing the application programs
- **Abstraction**
    - a **data model** is used to hide details
        * presents users with a *conceptual view* of the database
        * programmers refer to model constructs and not the nitty-gritty details
- **Multiple views**
    - each user can see a different view
    - **only see the data they care about**
- **Sharing data and multi-user transactions**
    - allow **concurrent** retrieval and modification of database
    - *concurrency control* guarantees either:
        * correct execution of a transaction OR
        * abortion of a transaction
    - *recovery* subsystem ensures each transaction's effect is correctly recorded
    - **OLTP** (online transaction processing) allows hundreds of concurrent transactions per second

2.5 Types of Database User

2.5.1 Actors Behind the Scenes

Those who design and develop DBMS software. Those who operate the computer systems.

- System designers and implementers
    - design and implement DBMS modules

- Tool developers
  - design and implement tools
    - modeling
    - designing
    - performance monitoring
    - prototyping
    - test data generation
    - UI creation
    - simulation
- Operators and maintenance personnel
  - tunnel rats
  - manage the running and maintenance of the DB

2.5.2 Actors on the Scene

Those who actually use and control the database content. Those who design, develop, and maintain database applications.

- DB administrators
  - acquire software and hardware resources
  - control the use of those resources
  - monitor efficiency
  - monitor use of DB
  - authorize access to DB
- DB designers
  - define the following aspects of a DB:
    - structure
    - constraints
    - content
    - transactions
  - must understand end users' needs
- System analysts
  - design applications and canned transactions for a DB
- Application developers
  - implement the specifications developed by analysts
- End users
  - use DB day-to-day
  - don't know or care how DB is structured
  - two categories:
    - naïve users
    - business analysts

3 Database System Concepts and Architecture

3.1 Data Representation

- We need to *abstract* the representation to make it meaningful

3.1.1 Hierarchical Model

- Tree-like structure

  - records
  - links
- Navigational and procedural operations

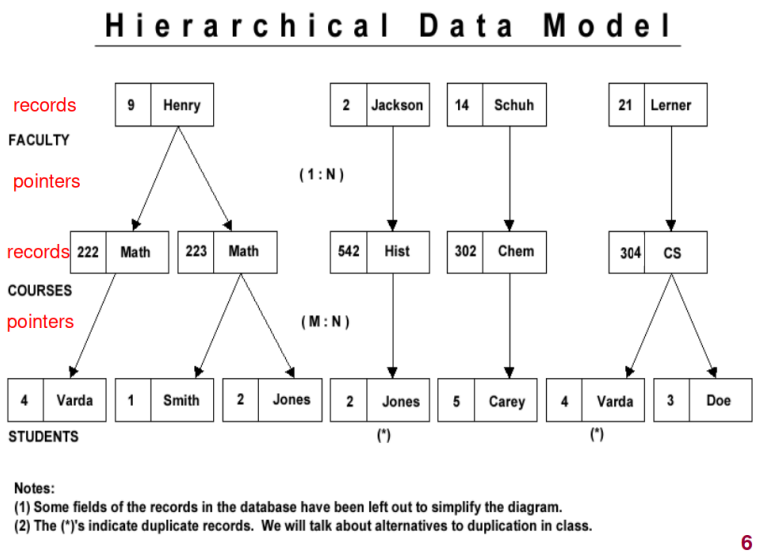

Figure 2: The hierarchical data model from Mengchi's slides.

3.1.2 Network Model

- Network structure
  - records
  - links
- Navigational and procedural operations

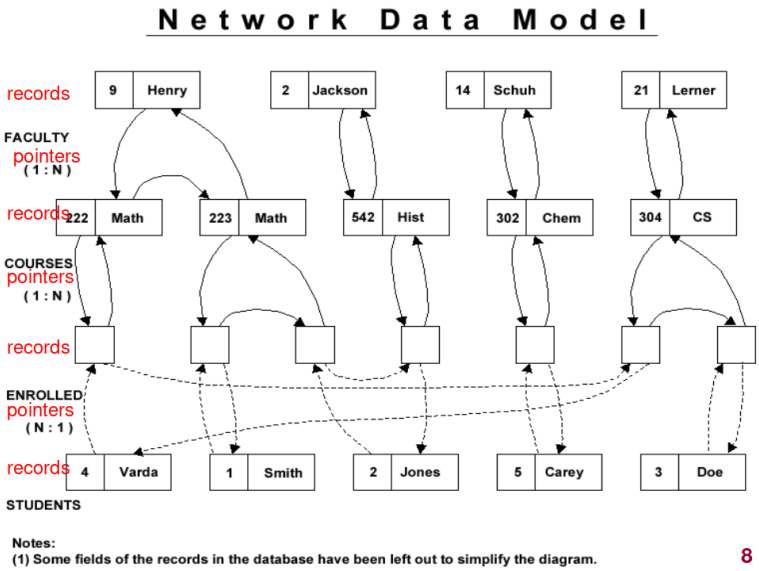

Figure 3: The network data model from Mengchi's slides.

3.1.3 Relational Model

- Tuples and relations
- Declarative operations specify what to get instead of how to get it

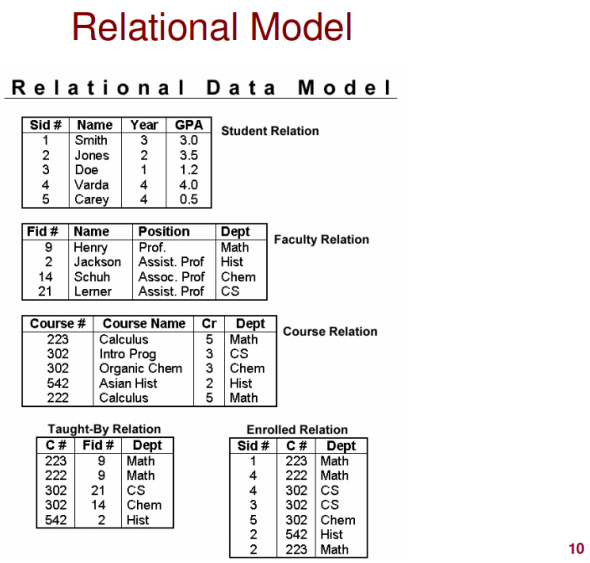

Figure 4: The relational data model from Mengchi's slides.

3.2 Schemas

- Description of the data at some abstraction level
- Three levels, each with its own schema:
  - internal (physical)
    - how the data is stored, physically
    - physical storage structures
    - access paths
  - conceptual
    - structure and constraints for the whole database
    - high-level or implementation data model
  - external
    - user views
    - typically same data model as conceptual schema
- Physical data independence
  - change internal schema without changing the conceptual schema
- Logical data independence
  - change conceptual schema without changing external schema
- See Figure 5 for a trick here (ICE PL)

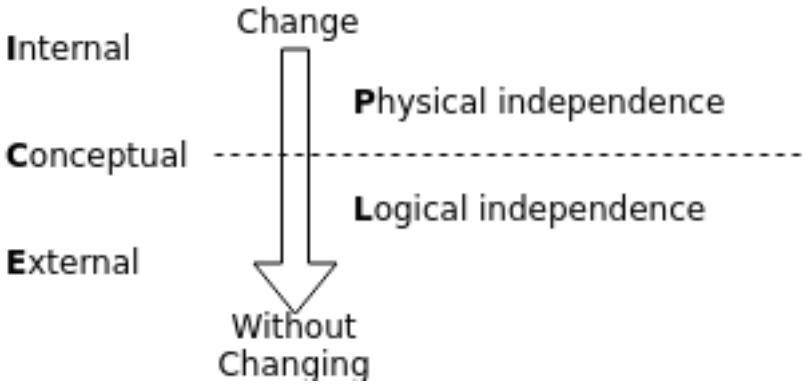

Figure 5: **ICE PL**, my trick for remembering schema types and which independence is which.

- Two important physical models
  - centralized
    - can still remote in but all processing is done centrally
  - client/server

3.3 Database Languages

- DDL (data definition language)
  - `insert`
  - `delete`
- DML (data manipulation language)
  - `update`
- QL (query language)
  - `get`
- SQL
  - combines all three

4 Relational Databases

4.1 Concepts

- Relation name
- Attributes (schema)
  - column headers
- Tuples (instance)
  - rows of entries in the table
- Domain
  - the set of all possible values of an attribute

4.2 Summary of Definitions

Informal Terms	Formal Terms
Table	Relation
Column Name	Attribute

Informal Terms	Formal Terms
All Possible Column Values	Domain
Row	Tuple
Table Definition	Schema of a Relation
Populated Table	Instance of a Relation