

Topic 2

Data Models, Schema, Instances

Ts. Chng Chern Wei

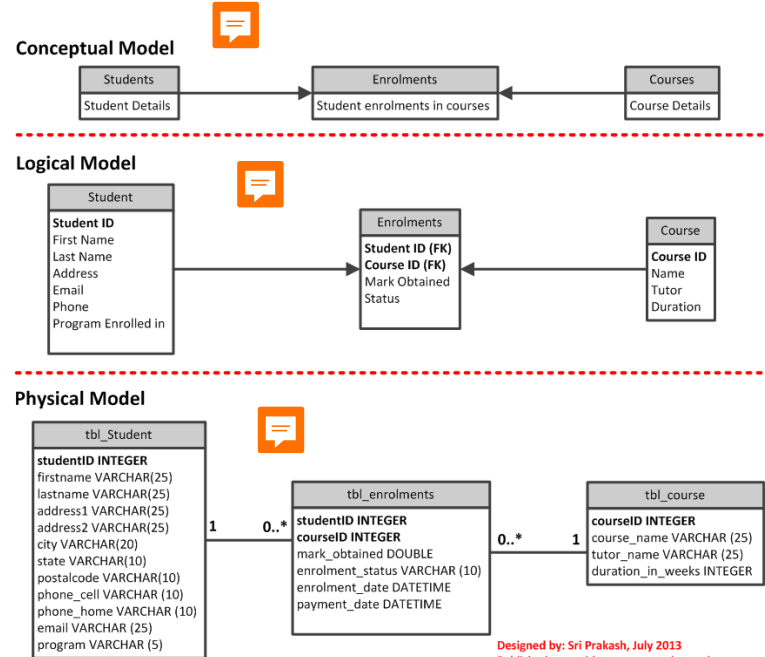


Data Model

Describe structure of the database by providing the **definition** and **format** of data and standardizing how they relate to one another and to the properties of real-world entities.

Classification:

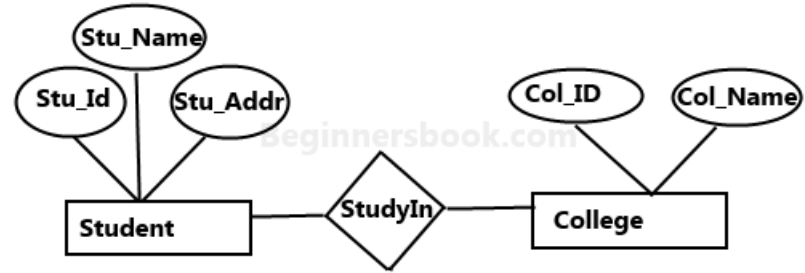
- A. High-level/Conceptual Model
- B. Representation/Logical Model
- C. Low-level/Physical Model



Designed by: Sri Prakash, July 2013
Published on my blog: ecomcanada.wordpress.com

A. High-Level Model

- provide concepts for presenting data in ways that are close to the way people perceive data



Sample E-R Diagram

- **Entity:** Any object, exists physically or conceptually
- **Attribute:** Property or characteristic of entity
- **Relationship:** Association or link between two/more entities
- These 3 terms make **Entity-Relationship Model** (ER Model)

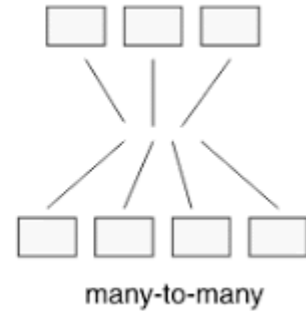
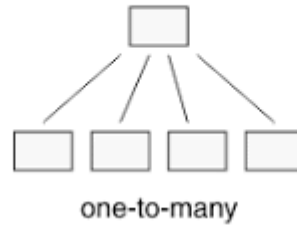
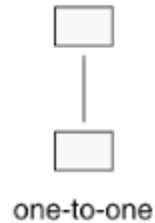
B. Representation Model



- Representation of data stored inside a database.
- Describes the physical structure of the database.
- It uses the concepts which are close to the end-users.
- **Classification:**
 1. Hierarchical
 2. Network
 3. Relational

Relationship between Data

- One to One
- One to Many
- Many to Many



Relationship between Data

- **One to One
bidirectional**

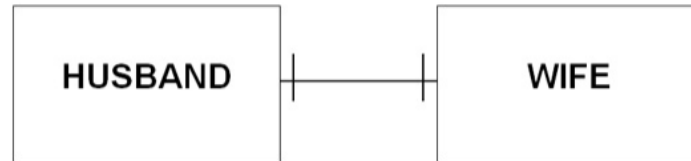
relationships. each row in one database table is linked to one and only one other row in another table.

- One to Many
- Many to Many



A Person can have only one Passport and that a Passport can belong to only one Person.

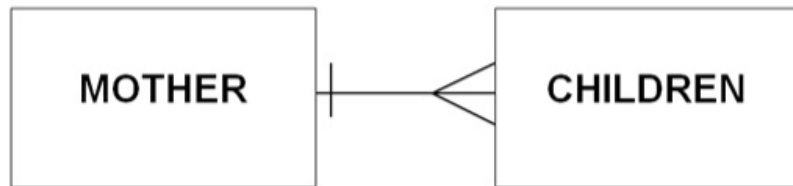
One-to-one relationship



Relationship between Data

- One to One
- **One to Many**
Single directional relationships. each row in one database table is linked to many rows in another table.
- Many to Many

One-to-many (or many-to-one) relationships

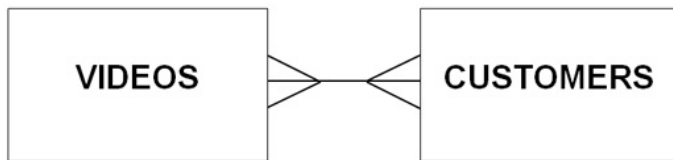


Relationship between Data

- One to One
- One to Many
- **Many to Many**

Bi-directional relationships. when multiple records in a table are associated with multiple records in another table.

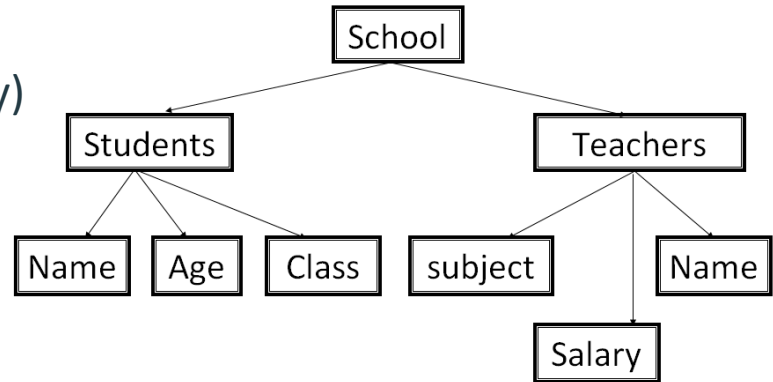
Many-to-many relationships



B1. Hierarchical Database Model

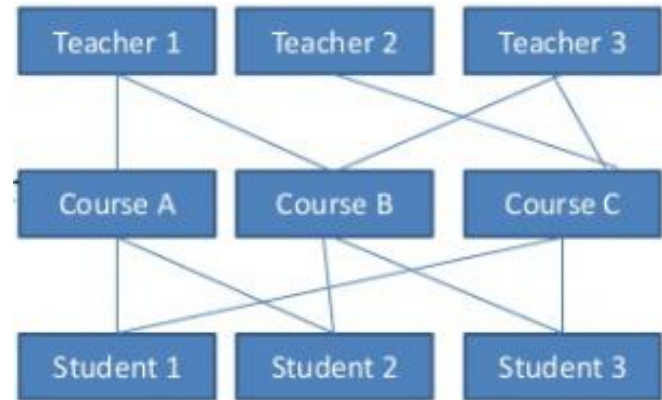
- Developed by IBM, the oldest database model
- Represented using a tree diagram
(Parent-child relationship / one to many)
- Each box is called a “**Node**”
- The nodes represent a record type.
- A line connecting nodes represents the link.

mainly for storing **file systems** and geographic information.



B2. Network Database Model

- Represented using a Data-Structure Diagram
- Boxes represents the records & lines the links
- Based on '**owner-member relationship**'
- Members of an owner may be many but for many member owner is one.
- Can represent **one-to-one** and **Many-to-many** as well.



B3. Relational Database Model

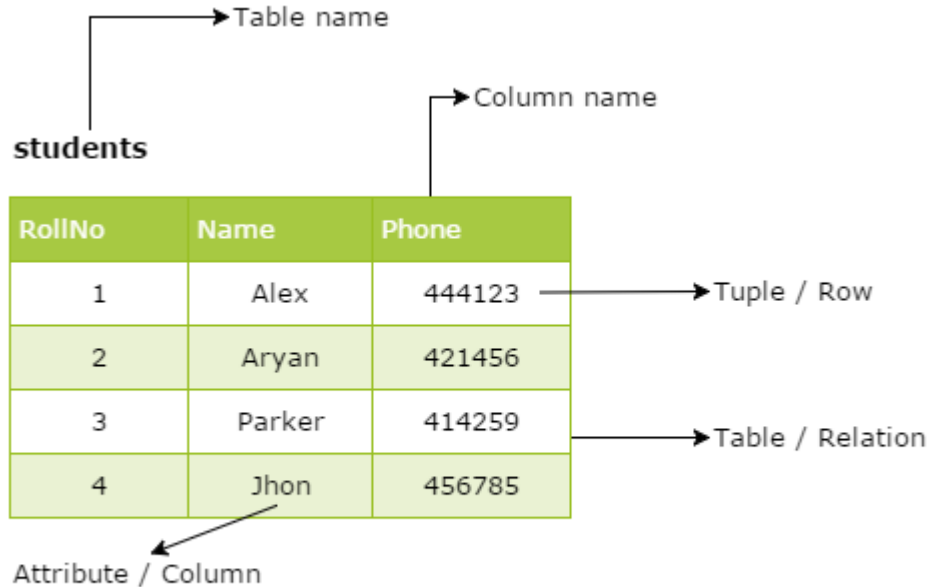
- Simplest and the most common model
- Data elements are stored in different tables made up of rows and columns.

Roll No	Name	Surname	Section
1001	Rajkumar	Tomar	D
1002	Rajkumar	Singh	D

B3. Relational Database Model

- Terminologies / Terms
 - Data Values
 - Columns / Attribute
 - Rows / Tuple
 - Table / Relation
 - Key

Roll No	Name	Surname	Section
1001	Rajkumar	Tomar	D
1002	Rajkumar	Singh	D



Relational Model Terms

Relation:

A relation is a table with columns and rows.

Attribute:

An attribute is a named column of a relation.

Domain:

A domain is the set of allowable values for one or more attributes.

Tuple:

A tuple is a row of a relation.

C. Low Level Database Model

- provides concepts that describe details (access path, record format, record ordering) of how data is stored in computer model. Low-level data model is only for Computer specialists not for end-user.

Data Schema

- A database schema is the **skeleton structure** that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.
- Doesn't show the data in database
- Classification:
 - Physical
 - Conceptual
 - External

STUDENT

Name	Student_number	Class	Major
------	----------------	-------	-------

COURSE

Course_name	Course_number	Credit_hours	Department
-------------	---------------	--------------	------------

PREREQUISITE

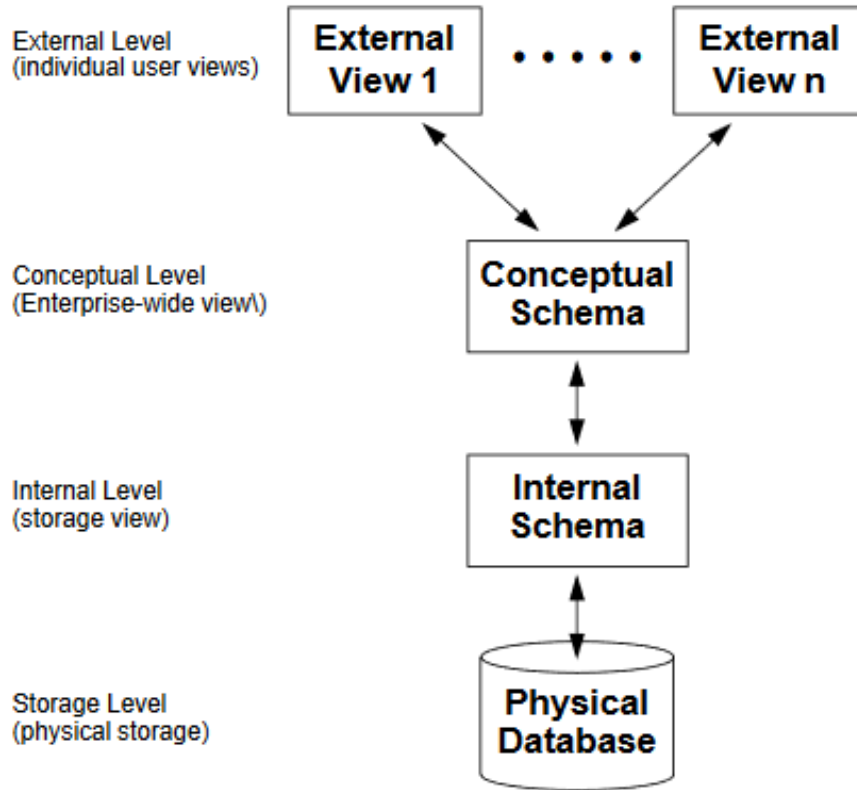
Course_number	Prerequisite_number
---------------	---------------------

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
--------------------	---------------	----------	------	------------

GRADE_REPORT

Student_number	Section_identifier	Grade
----------------	--------------------	-------



External Schema:

- Provides a user's view of data.
- Shows relevant info particular to user, hides rest of the info

Conceptual Schema:

- Describes structure of whole database
- Describes entities their relationships and constraints

Internal / Physical Schema:

- Describes physical storage of database

Data Model VS Data Schema

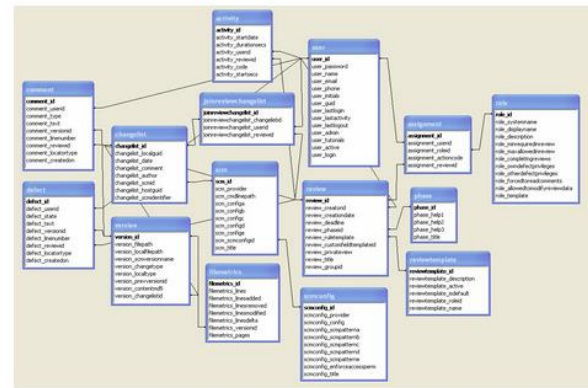
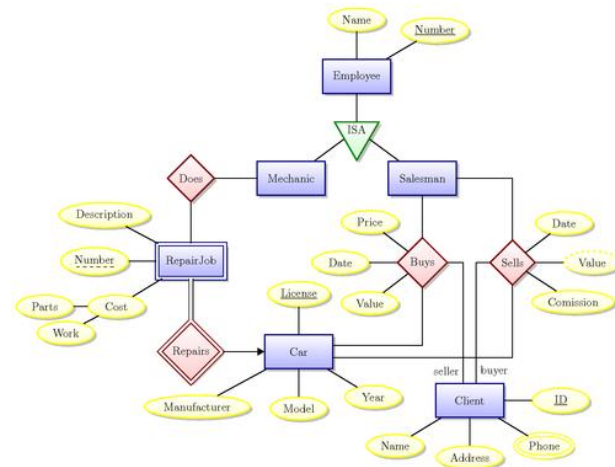
It is understandable why this would confuse:

- Both are not the data itself
- Both have data as a focus
- Both describe how data is stored
- Both require an understanding of special vocabularies, tools, and techniques to support development projects.
- Both do not have to be constrained to a single process or application, however they should have documented boundaries.

Data Model VS Data Schema

The difference is that a ***data model describes requirements*** for data definitions of descriptive information and relationships, while a ***database schema is the code that implements storage on a specific database platform.***

So more simply the ***data model is the definition***, the ***database schema the implementation***.



Database Instances

The environment of database is said to be instance. A database instance or an 'instance' is made up of the background processes needed by the database software. These processes usually include a process monitor, session monitor, lock monitor, etc. They will vary from database vendor to database vendor.

At a particular moment the data stored in database is called the instance, that changes over time when we add or delete data from the database.

Q&A?

Thank You!

Reference:

Abraham Silberschatz (2019), Database System Concepts. McGraw-Hill Education

Carlos Coronel et., al (2017) Database Systems: Design, Implementation, & Management, 12th Edn. Cengage Learning