



# THE RELATIONAL MODEL

# Model Data Representasional atau Implementasi

- ❑ Model Data Flat File (menggunakan format text)
- ❑ Model Data Hierarchical (data disusun secara berjenjang)
- ❑ Model Data Network (menggunakan directed graphs)
- ❑ Model Data Relasional (yang akan menjadi pembahasan utama).
- ❑ Dikenalkan Model Data Object.

# Relational Model

The central data description construct in this model is a relation, which can be thought of as a set of records.

A description of data in terms of a data model is called a schema.

In the relational model, the schema for a relation species its name, the name of each field (or attribute or column), and the type of each field.

Students(sid: string, name: string, login: string, age: integer, gpa: real)

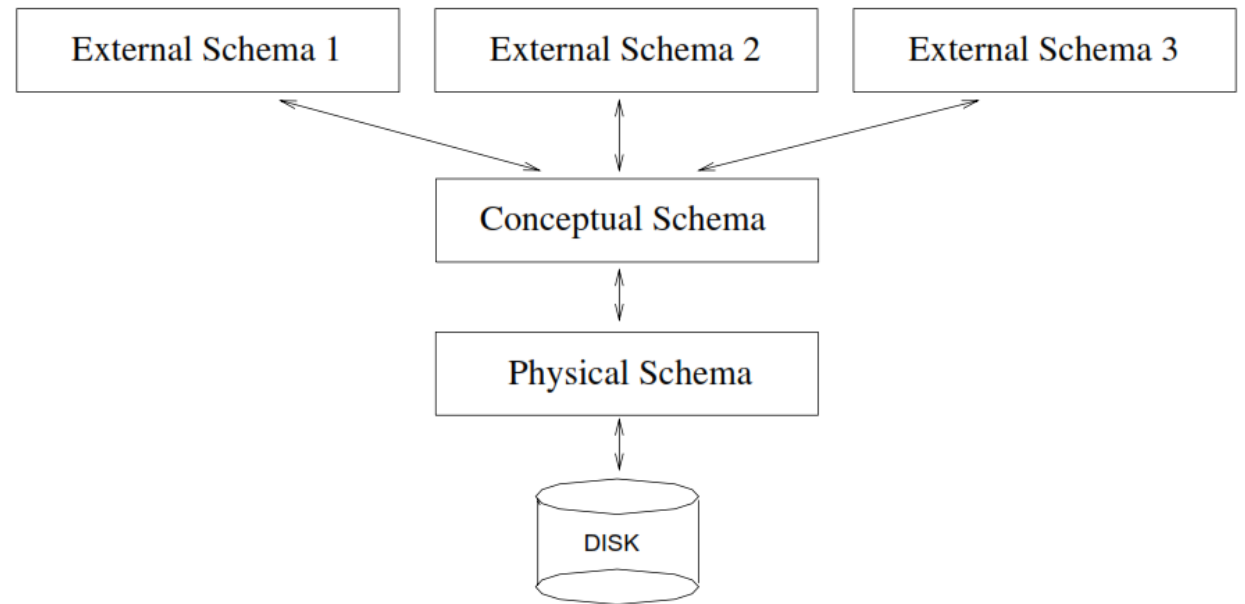
<i>sid</i>	<i>name</i>	<i>login</i>	<i>age</i>	<i>gpa</i>
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

# Levels of Abstraction in a DBMS

Conceptual schemas.

Physical schemas

External schemas.



*Information about the conceptual, external, and physical schemas is stored in the **system catalogs***

# Conceptual Schema (Logical Schema)

Describes :

- The stored data in terms of the data model of the DBMS
- All relations that are stored in the database

**Example:** university database

These relations contain information about entities, such as students and faculty, and about relationships, such as students' enrollment in courses.

Students(*sid*: string, *name*: string, *login*: string, *age*: integer, *gpa*: real)

Faculty(*fid*: string, *fname*: string, *sal*: real)

Courses(*cid*: string, *cname*: string, *credits*: integer)

Rooms(*rno*: integer, *address*: string, *capacity*: integer)

Enrolled(*sid*: string, *cid*: string, *grade*: string)

Teaches(*fid*: string, *cid*: string)

Meets\_In(*cid*: string, *rno*: integer, *time*: string)

# Physical Schema

- Specifies additional storage details
- Summarizes how the relations described in the conceptual schema are actually stored on secondary storage devices such as disks and tapes.
- What file organizations to use to store the relations, and create auxiliary data structures called **indexes** to speed up data retrieval operations.

**Example:** A physical schema for the university database:

- Store all relations as unsorted files of records. (A file in a DBMS is either a collection of records or a collection of pages, rather than a string of characters as in an operating system.)
- Create indexes on the first column of the Students, Faculty, and Courses relations, the *sal* column of Faculty, and the *capacity* column of Rooms.

# External Schema

- Allow data access to be customized (and authorized) at the level of individual users or groups of users.
- Any given database has exactly one conceptual schema and one physical schema because it has just one set of stored relations, but it may have several external schemas
- Each external schema consists of a collection of one or more views and relations from the conceptual schema.

- The external schema design is guided by end user requirements.
- For example, we might want to allow students to find out the names of faculty members teaching courses, as well as course enrollments, by defining the following view:

`Courseinfo(cid: string, fname: string, enrollment: integer)`