The Relational Model

Data and Its Structure

- Data is actually stored as bits, but it is difficult to work with data at this level.
- It is convenient to view data at different *levels of abstraction*.
- ***Schema***: Description of data at some abstraction level. Each level has its own schema.
- We will be concerned with three schemas: *physical*, *conceptual*, and *external*.

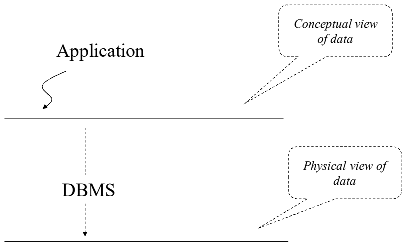
Physical Data Level

- *Physical schema* describes details of how data is stored: tracks, cylinders, indices etc.
- Early applications worked at this level – explicitly dealt with details.
- **Problem:** Routines were hard-coded to deal with physical representation.
  - Changes to data structure difficult to make.
  - Application code becomes complex since it must deal with details.
  - Rapid implementation of new features impossible.

Conceptual Data Level

- Hides details.
  - In the relational model, the conceptual schema presents data as a set of tables.
- DBMS maps from conceptual to physical schema automatically.
- Physical schema can be changed without changing application:
  - DBMS would change mapping from conceptual to physical transparently
  - This property is referred to as ***physical data independence***

Conceptual Data Level (con't)



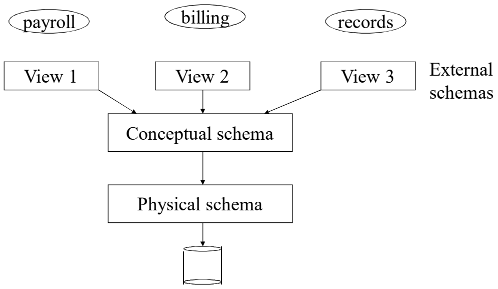
External Data Level

- In the relational model, the *external schema* also presents data as a set of relations.
- An external schema specifies a *view* of the data in terms of the conceptual level. It is tailored to the needs of a particular category of users.
  - Portions of stored data should not be seen by some users.
    - Students should not see their files in full.
    - Faculty should not see billing data.
  - Information that can be derived from stored data might be viewed as if it were stored.
    - GPA not stored, but calculated when needed.

External Data Level (con't)

- ➢ Application is written in terms of an external schema.
- ➢ A view is computed when accessed (not stored).
- ➢ Different external schemas can be provided to different categories of users.
- ➢ Translation from external to conceptual done automatically by DBMS at run time.
- ➢ Conceptual schema can be changed without changing application:
  - Mapping from external to conceptual must be changed.
- ➢ Referred to as ***conceptual data independence.***

Levels of Abstraction



Data Model

- ➢ **Schema**: description of data at some level (*e.g.,* tables, attributes, constraints, domains)
- ➢ **Model**: tools and language for describing:
  - Conceptual and external schema
    - *Data definition language* (DDL)
  - Integrity constraints, domains (DDL)
  - Operations on data
    - *Data manipulation language* (DML)
  - Directives that influence the physical schema (affects performance, not semantics)
    - *Storage definition language* (SDL)

Relational Model

- ➢ A particular way of structuring data (using relations)
- ➢ Simple
- ➢ Mathematically based
  - Expressions (≡ *queries*) can be analyzed by DBMS
  - Queries are transformed to equivalent expressions automatically (*query optimization*)
    - Optimizers have limits (=> programmer needs to know how queries are evaluated and optimized)

Relation Instance

- ➢ Relation is a set of tuples
  - Tuple ordering immaterial
  - No duplicates
  - *Cardinality* of relation = number of tuples
- ➢ All tuples in a relation have the same structure; constructed from the same set of attributes
  - Attributes are named (ordering is immaterial)
  - Value of an attribute is drawn from the attribute's *domain*
    - There is also a special value **null** (value unknown or undefined), which belongs to no domain
  - *Arity* of relation = number of attributes

Relation Instance (Example)

Id	*Name*	*Address*	*Status*
1111111	John	123 Main	freshman
2345678	Mary	456 Cedar	sophmore
4433322	Art	77 So. 3rd	senior
7654321	Pat	88 No. 4th	sophmore

Student

Relation Schema

- Relation name
- Attribute names & domains
- Integrity constraints like
  - The values of a particular attribute in all tuples are unique
  - The values of a particular attribute in all tuples are greater than 0
- Default values

Relational Database

- Finite set of relations
- Each relation consists of a schema and an instance
- *Database schema* = set of relation schemas constraints among relations (*inter-relational* constraints)
- *Database instance* = set of (corresponding) relation instances

Database Schema (Example)

Student (*Id*: INT, *Name*: STRING, *Address*: STRING,
Status: STRING)
Professor (*Id*: INT, *Name*: STRING, *DeptId*: DEPTS)
Course (*DeptId*: DEPTS, *CrsName*: STRING,
CrsCode: COURSES)
Transcript (*CrsCode*: COURSES, *StudId*: INT,
Grade: GRADES, *Semester*: SEMESTERS)
Department(*DeptId*: DEPTS, *Name*: STRING)

The Relational Model Exercise

Design a schema for airline information about passengers, planes, flights, and passenger reservations. There are different types of planes and there are a maximum number of passengers that will fit on a plane. Each flight consists of a plane that flies from one location to another at a specific time and date.