The Relational Data Model

Chapter 4

1

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 level. Each level has its own schema.
- We will be concerned with three schemas: physical, conceptual, and external.

2

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 hardcoded 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.

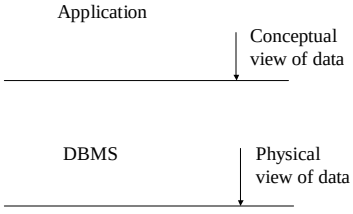
3

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 must change mapping from conceptual to physical.
- Referred to as *physical data independence*.

4

Conceptual Data Level (con't)



5

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 faculty salaries.
    - 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, calculated when needed.

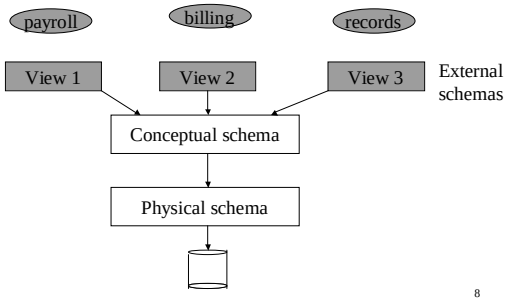
6

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*.

7

Levels of Abstraction



8

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)

9

Relational Model

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

10

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 named (=> ordering immaterial)
  - Value of an attribute drawn from the attribute's domain
  - Arity = number of attributes

11

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

12

Relation Schema

- Relation name
- Attribute names and domains
- Integrity constraints - e.g.,:
  - 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

13

Relational Database

- Finite set of relations
- Each relation consists of a schema and an instance
- Database schema = set of relation schemas (and other things)
- Database instance = set of (corresponding) relation instances

14

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)

15