

Chapter 2: Database System Concepts and Architecture

define: *data model* - set of concepts that can be used to describe the structure of a database
data types, relationships and constraints
set of basic operations - retrievals and updates
specify *behavior* - set of valid user-defined operations

categories:

- high-level (conceptual data model) - provides concepts the way a user perceives data
 - entity - real world object or concept to be represented in db
 - attribute - some property of the entity
 - relationship - represents and interaction among entities
- representational (implementation data model) - hide some details of how data is stored, but can be implemented directly
 - record-based models like relational are representational
- low-level (physical data model) - provides details of how data is stored
 - record formats
 - record orderings
 - access path (for efficient search)

schemas and instances:

- database *schema* - description of the data (meta-data)
 - defined at design time
 - each object in schema is a schema construct
 - EX: look at TOY example - top notation represents schema
 - schema constructs: cust ID; order #; etc.
- database *state* - the data in the database at any particular time - also called set of *instances*
 - an instance of data is filled when database is populated / updated
 - EX: cust name is a schema construct; George Grant is an instance of cust name
- difference between schema and state
 - at design time, schema is defined and state is the empty state
 - state changes each time data is inserted or updated, schema remains the same

Three-schema architecture

- goal: to separate the user applications and the physical database
- three levels:

(draw picture)

- 1) internal level (internal schema) - describes physical storage
 - uses a physical data model to specify details of data storage and access paths
 - 2) conceptual level (conceptual schema) - describes structure of db for all users
 - hides details of storage
 - describes entities, data types, relationships, user operations, constraints
 - uses high-level data model or representational data model
 - 3) external or view level (several external schemas or user views)
 - each describes part of db of interest to particular user group
 - hides the rest of the db
 - uses high-level data model or representational data model
- users make db requests at the level of their particular external view
- mapped to conceptual level and then to internal level
 - db response mapped back to user's external view

Data independence:

change schema at one level of db without affecting schemas at other levels

logical data independence: change the conceptual schema without having to change external schemas or application programs

physical data independence: change internal schema without having to change conceptual (or external) schemas.

mapping between levels changes so that levels can remain independent

- incurs an overhead during compilation or execution of queries or programs
- because of this, few dbms's have implemented the full three-schema architecture

DBMS languages:

data definition language (DDL) - specifies conceptual schema

storage definition language (SDL) - specifies internal schema

- some DBMSs have no distinction between these schemas and thus the DDL and the SDL are the same

- mappings between schemas are done in either of the two languages

view definition language (VDL) - specifies external schemas (in DBMSs where this level exists)

data manipulation language (DML) - means for users to manipulate data once db is defined

- includes retrieval, insertion, deletion and modification
- high-level (nonprocedural) DML
 - entered interactively directly to terminal (query language)
 - or embedded into programming language
 - can retrieve sets of data at a time
 - specifies what data to retrieve and not how to retrieve it
- low-level (procedural) DML - must be embedded in programming language
 - can only retrieve one data item at a time
 - must use prog lang constructs to manipulate groups of data (loops, etc)

- if stand-alone, called a query language
 - can be embedded into another language
- SQL relational db language represents a combination of all above languages
- can be stand-alone, or embedded into a language like C

Other Interfaces:

- menus - provide a list of options
- graphical interface - schema displayed graphically
 - form query by manipulating the graph
- natural language - request written in English - interpreted for db use - AI

Classification of DBMSs:

- data model - relational, network, hierarchical, OO, other
 - most often used way of classifying a dbms
- # users - single / multiple users
- # sites - centralized - data stored in one central computer site
 - distributed - databases and dbms distributed among several sites
 - homogeneous - same dbms software at each site
 - heterogeneous - own local dbms software - different sites have to be coordinated somehow