Lecture-5
(Introduction to Databases)

©Dr. Nikhil Tripathi
https://nikhiltripathi.in

Learning Objectives

- Self Assessment
- DB User and DB Administrator (contd.)
- Data Models
- Database design

Self Assessment

- DDL and DML
- Relational Databases
  - Tables -> ?, Columns -> ?, Rows -> ?
- Providing authorization is a part of DDL or DML?
- Inserting data into database is a part of ---?
- How to access database using application programs?
- DB user vs DB administrator

DB User & DB Administrator

- People who work with a database.
- Types of DB users:
  - Naïve users: Unsophisticated users who interact with the database by invoking application programs that have been written previously
    - E.g. A university clerk adds details of a new faculty (name, department, back account no., etc.) using a simple web based form.
  - Application programmers: Computer professionals who write application programs to develop user interfaces.
  - Sophisticated Users: They interact with the database using a database query language or by using tools such as data analysis software.
  - Specialized users: Responsible for writing specialized database applications such as computer-aided design systems, environment-modeling systems, etc.

DB User & DB Administrator (contd.)

- It is necessary to have a central control of both the data and the programs that access those data.
- The person having such central control over the system is called a database administrator.
- Functions of a DB Administrator:
  - Schema Definition: DBA creates the original database schema by executing a set of data definition statements in the DDL.
  - Access-method definition
  - Schema modification
  - Granting of authorization for data access
  - Routine maintenance
    - Periodically backing up the database
    - Ensuring that enough free disk space is available
    - Monitoring jobs running on the database and ensuring that the performance is not degraded due to expensive tasks submitted by some users.

Data Models

- A collection of "conceptual" tools to describe:
  - Data
  - Data relationships
  - Data semantics
  - Consistency constraints
- Provides a way to describe the design of the database at all the three levels.
- Data models can be classified into four different categories:
  - Relational Model
  - Entity-Relationship Model
  - Object-Based Data Model
  - Semistructured Data Model
  - Network Data Model and Hierarchical Data Model (obsolete so rarely used)

Relational Model

- Relational model uses a collection of tables to represent:
  - data and
  - the relationships among those data
    - E.g. student and his/her address, phone number, etc.
- Each table has multiple columns and each column has a unique name
  - Tables are also known as relations
- Relational model is an example of record-based model.
  - DB is structured in fixed-format records of several types
  - Each table contains records of particular type
  - Each record has fixed number of fields/attributes/columns
- It is the most widely used data model
- Most of the current DBMSs are based on it

Entity-Relationship (ER) Model

- This model uses a collection of basic objects, called *entities,* and *relationships* among these objects.
  - E.g. Instructor X is a member of department Y. Here, X and Y are two different entities while ‘member’ is the relationship
- Thus, an entity is a thing or object in the real world that is indistinguishable from other objects
- Widely used in database design.

Object-Based Data Model

- Motivated from the Object-oriented programming
- It is seen as the extension of E-R model with notions of encapsulation, methods (functions), and object identity.
- Object-relational data model combines features of the object-oriented data model and relational data model.
- Why is it required at the first place itself?
  - Reason 1: Limited type system supported by the relational model and complex application domains require complex data types
    - Nested record structures (e.g. Address)
    - Multivalued attributes
    - Inheritance
  - Reason 2: Difficulty in accessing database data from C, C++ and Java.
    - type system of the database and the type system of the programming language make data storage and retrieval more complicated
    - Programmers' life become harder
- It can be used for complex applications such as computer-aided design, geographical information systems, etc.

Semistructured Data Model

- Semistructured data model permits the specification of data where individual data items of the same type may have different sets of attributes.
  - Car C1 has attributes ‘brand’, ‘color’, ‘Battery capacity’
  - Car C2 has attributes ‘brand’, ‘color’, ‘Fuel tank’
  - Student S1 may have an attribute ‘aadhar card no.’ while student S2 (non-Indian) may have an attribute ‘Passport No.’.
- In contrast, earlier data models do not permit the data items of the same type to have different sets of attributes.
- XML is a popular example of this data model.

Other Data Models

- Network Data model
- Hierarchical Data model
- They preceded the relational data model.
- These models were tied closely to the underlying implementation, and complicated the task of modeling data.
- Thus, they are rarely used now except in old database code.

Database Design

- Involves the design of database schema
- Since the database contains data essential for an enterprise's efficient working, it requires a large set of issues to be considered.
- The database designer needs to interact extensively with domain experts and users to carry out this task.
- Outcome: Specification of user requirements.
- What's next?

Database Design (contd.)

- Designer chooses a data model
- By applying the concepts of the chosen data model, the designer translates these requirements into a conceptual schema of database.
- This phase is known as conceptual-design phase.
- This schema provides a detailed overview of the enterprise.
- The designer reviews the schema to confirm that all data requirements are satisfied.
- The focus at this point is on describing the data and their relationships
- In terms of relational model, this phase involves decisions on:
  - *what* attributes we want to capture in the database and *how to group* these attributes to form various tables.
  - The *what* part is a business decision so we don't care. However, the *how* part is a computer science problem.
- Two ways to tackle this problem:
  - Use Entity-relationship model
  - Use *normalization* (takes as input all attributes and generate a set of tables)

Database Design (contd.)

- Now the abstract data model is ready (i.e. tables to be included are known and the attributes of these tables are known)
- What's next?
- Implement this abstract data model into a database.
- This proceeds in two final design phases.
  - Logical-design phase
  - Physical-design phase
- Logical-design phase:
  - Designer maps the high-level conceptual schema onto the implementation data model of the DBMS.
- Physical-design phase:
  - Designer specifies the physical features of the database (file organization, internal storage structures).

Database Design: An Example

- Designing a database for a university
- Initial specification of user requirements:
  - Interviews with the DB users (accounts, academic section, etc.)
- Major characteristics of a university:
  - University is organized into departments. Each department has a name, located in a particular building, and an annual budget.
  - Each department has a list of courses it offers. Each course has associated with it a course id, title, dept name, and credits, and may also have associated prerequisites.
  - Instructors are identified by their unique ID. Each instructor has name, associated department (dept name), and salary.
  - Students are identified by their unique ID. Each student has a name, an associated major department (dept name), and tot cred (total credit hours the student earned thus far).
  - The university maintains a list of classrooms, specifying the name of the building, room number, and room capacity.
- That's it (?)
  - No, university database is much more complex

Lecture Summary

- DB User and DB Administrator
- Data Models
- Database Design and illustration with an example.

References

- Silberschatz, Korth and Sudharsan