Lecture-4
(Introduction to Databases)

©Dr. Nikhil Tripathi

https://nikhiltripathi.in

Learning Objectives

- Self Assessment
- Database Languages
  - Data-Manipulation Language (DML)
  - Data-Definition Language (DDL)
- Relational Databases
  - Introduction
- DB User and DB Administrator

Self Assessment

- What is DBMS?
    - Collection of interrelated data and set of programs to access/modify the data.
- Three level data abstraction architecture
    - Physical level, logical level and view level
- Instances and schema:
    - Instance – Collection of information stored in the database at a particular point of time
    - Schema – Overall design of the database
- Physical data independence
    - Property of application program

Data-Manipulation Language (DML)

- Enables a DB user to access or manipulate data in DB.
    - Retrieval of information stored in DB
    - Insertion of new information into DB
    - Deletion of information from DB
    - Modification of information stored in DB
- Procedural DMLs
    - User specifies what data are needed and how to get those data.
- Declarative DMLs (aka nonprocedural DMLs)
    - User specifies only what data are needed.
    - Easier to learn
- What is a Query?
    - A statement requesting the retrieval of information.
    - Language that we use to write the query is called query language
        - E.g. SQL
    - Query processor component of DBMS translates DML queries into low-level sequence of actions at physical level.

Data-Definition Language (DDL)

- DDL is used to specify a database schema (design) by a set of definitions.
- DDL is also used to specify additional data properties.
- Data values stored in database must satisfy certain consistency constraints
  - E.g. Age of a student in a university database can never be negative.
  - DDL provides facilities to specify such constraints.
  - DB system checks these constraints every time the database is updated

Data-Definition Language (DDL)

- Different integrity constraints implemented by DB systems:
  - Domain Constraints: Domain of possible values must be associated with every attribute.
    - E.g. Integer types, character types
  - Referential Integrity: There are cases where we wish to ensure that a value that appears in one table for a given set of attributes also appears in a certain set of attributes in another table.
    - E.g. student_name value in a course record in 'courses registered' table must appear in the student_name attribute of some record of the 'student' table.
  - Assertions: An assertion is any condition that the database must always satisfy. Domain constraints and referential integrity are special types of assertions.
    - E.g. Every department must have at least five courses offered every semester.
    - When an assertion is created, the system tests it for validity. If the assertion is valid, then any future modification to the database is allowed only if it does not cause that assertion to be violated.

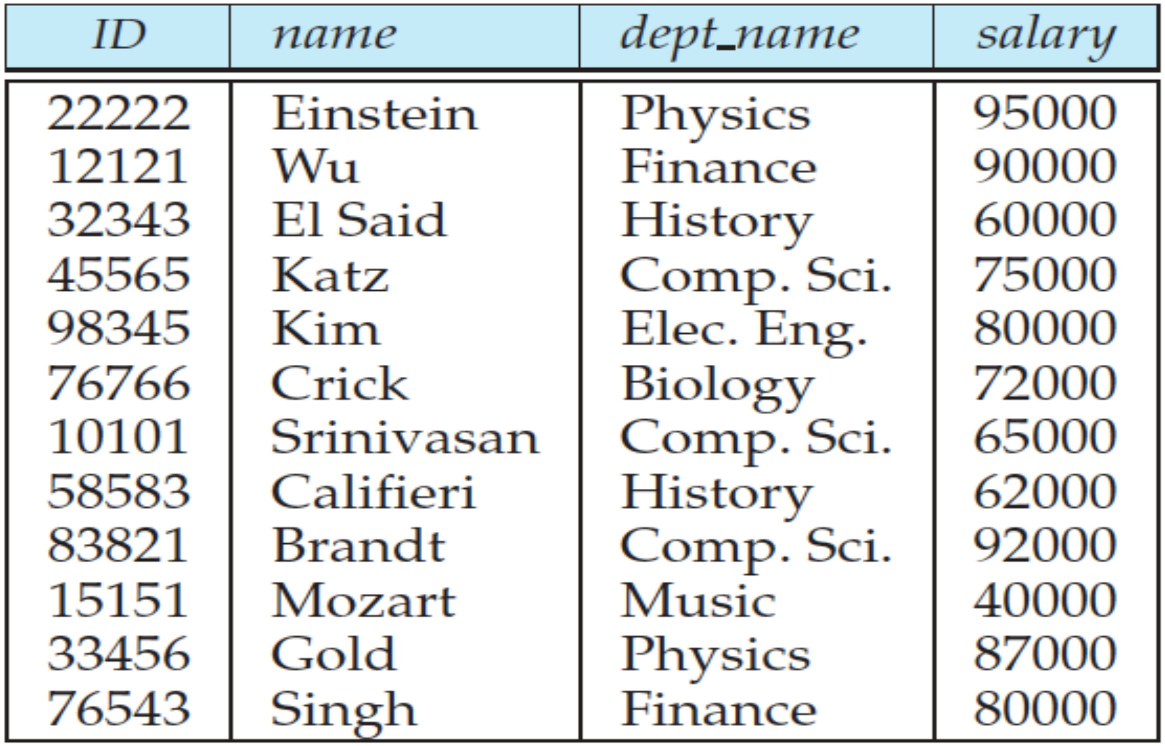
Data-Definition Language (DDL)

- Authorization
  - Differentiating users based on the type of access they are given:
    - Read authorization, Insert authorization, Update authorization, delete authorization
    - We may assign a user all of them or just a combination of these authorizations.

Relational Databases

- Relational databases are based on the relational model (will be discussed later) and use a collection of tables to represent both data and the relationships among those data.
  - Relational databases also use DML and DDL (particularly SQL).
- Tables:
  - Each table has multiple columns; each column has unique name

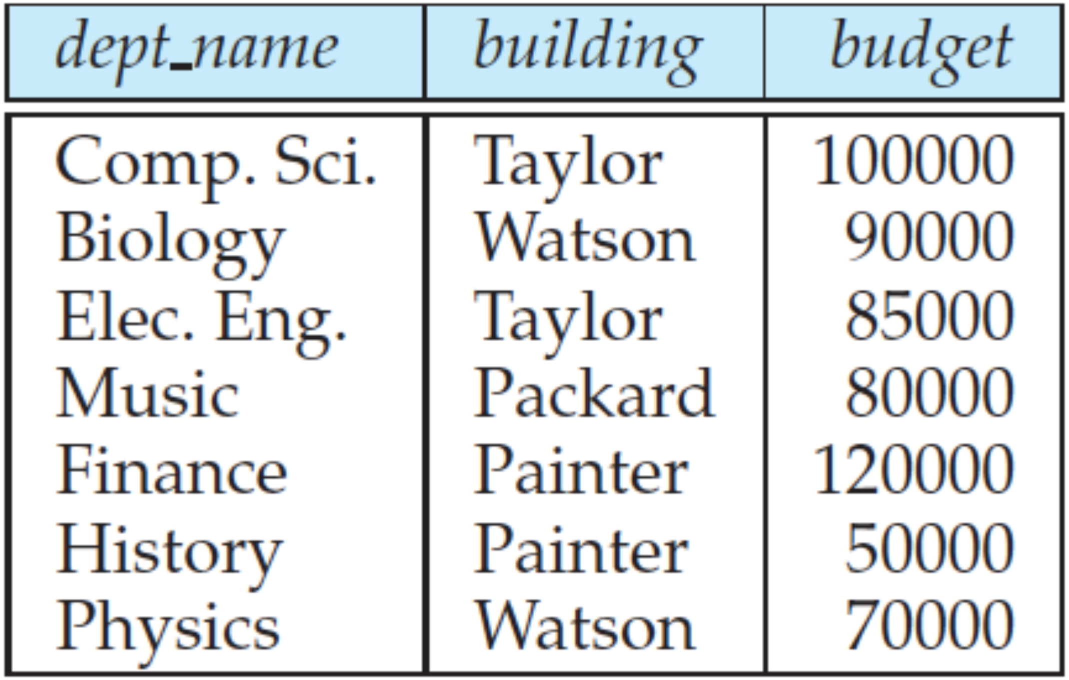
Relational Databases



ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

The *instructor* table

Relational Databases



dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

The *department* table

Relational Databases

- Relational model is an example of record-based model
  - Row -> record, Column -> attribute
- How are tables stored in files?
  - Attribute1, attribute2 … attributeN\n
  - However, the relational model hides such low-level implementation details from database developers and users.

Relational Databases

- Data-Manipulation Language
  - Example of an SQL query that finds the name of all instructors in the History department:

```
select instructor.name
from instructor
where instructor.dept_name = ‘History’;
```

  - Queries may involve information from more than one table.

```
select instructor.ID, department.dept_name
from instructor, department
where instructor.dept_name = department.dept_name and
department.budget > 95000;
```

Relational Databases

- Data-Definition Language
  - SQL provides a rich DDL to define tables, integrity constraints, assertions, etc.
  - Example of an SQL query that defines the *department* table:

```
create table department
(dept_name char (20),
building char (15),
budget numeric (12));
```

Database Access from Application Programs

- SQL can only be used to interact with the database but it can not be used to take input from users, output to displays or communication over the network.
- Such actions must be written in languages such as C, C++, or Java, with embedded SQL queries that access the data in the database. Such programs are called application programs.
- To access the database, DML statements are executed from the host language. The most efficient way to do this is:
  - Using an API that can be used to send DML and DDL statements to the database and retrieve the results.
    - E.g. ODBC for C and JDBC for Java

DB User vs DB Administrator

- DB user typically uses DML to access and/or modify data in the databse
- DB administrator is responsible for the database schema and thus, uses DDL.
- DBA is also responsible for differentiating DB users based on the authorization given to different users.
- E.g. IT Administrator can be a DBA while students, faculties and other university staff can be database users.

References

- Silberschatz, Korth and Sudharsan