Shahjalal University Of Science And Technology, Sylhet
Software Engineering, IICT

Database management system

ASSIGNMENT ON DBMS

Prepared By:

Md. Mehedi Hasan
2017831023

Supervisor:

Ahsan Habib
Associate Professor, IICT, SUST

October 2020

Contents

1. Difference between a file-processing system and a DBMS

File-processing system	DBMS
1. File system is a software that manages and organizes the files in a storage within a computer	DBMS is a software for managing the database
2. Redundant data can be present in the file system.For example: if a student has a double major (say music and mathematics)his data may appear in a file that consists of data of the music department and also in a file that consists of data of the mathematics department.	In DBMS there is no redundant data.
3. It is very difficult to retrieve data from file system.	Data retrieval is very easy in DBMS.
4. There is no efficient way to query data. If it is needed to query a specific student then an application program should be created for this purpose. But this program can't process another query so different application programs should be created for different purposes which is very inefficient.	There is an efficient way for query processing.
5. The file system faces integrity problems. Suppose in a varsity database there is a balance for every department and it can't be less than zero. It is difficult to ensure in file processing system	There is no integrity problem in DBMS.
6. File systems provide less security in comparison to DBMS.	DBMS has more security mechanisms as compared to file system.
7. The file processing system faces the atomicity problem.	Atomicity problems can be handled in an efficient way in DBMS.
8. File processing system is less expensive compared to DBMS	DBMS is more expensive compared to file-processing system

2. Define DDL(Data definition language) and DML(Data manipulation language)

A database system provides a data-definition language to specify the database schema and a data-manipulation language to express database queries and updates. In practice, DDL and DML are not two separate languages instead, they simply form parts of a single database language, such as the SQL language.

2.1. DDL(Data definition language)

:A database schema is specified by a set of definitions expressed by a special language called a data-definition language.DDL is also used to specify additional properties of the data. The data values stored in a database must satisfy certain consistency and constraints. For example, suppose the university requires that the account balance of a department must never be negative. The DDL provides facilities to specify such constraints. The database system checks these constraints every time the database is updated. Database systems implement only those integrity constraints that can be tested with minimal overhead:

- **Domain Constraints**: A domain of possible values must be associated with constraint(for example, integer types, character types, date/time types). Declaring an attribute to be a particular domain acts as a constraint on the values that it can take. Domain constraints are the most elementary form of integrity constraint. They are tested easily by the system whenever a new data item is entered into the database.

- **Referential Integrity**: There are cases where we wish to ensure that a value that appears in one relation for a given set of attributes also appears in a certain set of attributes in another relation(referential integrity). For example, the dept name value in a course record must appear in the dept name attribute of some record of the department relation. Database modifications can cause violations of referential integrity. When a referential-integrity constraint is violated, the normal procedure is to reject the action that caused the violation.

- **Authorization**: We may want to differentiate among the users as far as the type of access they are permitted on various data values in the database. These differentiations are expressed in terms of authorization, the most common being: read authorization, which allows reading, but not modification, of data; insert authorization, which allows insertion of new data, but not modification of existing data; update authorization, which allows modification, but not deletion, of data; and delete authorization, which allows deletion of data. We may assign the user all, none, or a combination of these types of authorization.

Example: SQL provides a rich DDL that allows one to define tables with data types and integrity constrains. For instance, the following SQL DDL statement defines the department table:

Create table department(
dept_name char(20),
‘ building char(15),
budget numeric(12,2));

Execution of the preceding DDL creates the department table with three columns: dept_name, building, budget, each of which has a specific data types associated with it.

2.2. DML(Data manipulation language)

: A data-manipulation language is a language that enables users to access or manipulate data as organized by the appropriate data model. The types of access are:

- Retrieval of information stored in the database.
- Insertion of new information into the database.
- Deletion of information from the database.
- Modification of information from the database.

There are basically two types of data manipulation language:

- **Procedural DMLs**:Procedural DMLS require a user to specify what data are needed and how to get those data.
- **Declarative DMLs**: Declarative DMLs require a user to specify what data are needed without specifying how to get those data.

Declarative DML s are usually easier to learn and use than are procedural DML s. However, since a user does not have to specify how to get the data, the database system has to figure out an efficient means of accessing data.

Example: The SQL query language is a nonprocedural. Here is an example of SQL query that finds the names of all instructors in the History department:

Select instructor.name
from instructor
Where instructor.dept_name == ’History’

The result of executing this query is a table with a single column labeled name and a set of rows, each of which contains the name of an instructor whose dept name is History

3. Question 3: Define Instances and Schemas

Schema: The overall design of the database is called the database schema. **Instance**: The collection of information stored in the database at a particular moment is called an instance of the database.

The concept of database schema and instance can be understood by analogy to a program written in a programming language.**A database schema corresponds to the variable declarations (along with associated type definitions) in a program**. Each variable has a particular value at a given instant. The values of the variables in a program at a point in time correspond to an **instance** of a database schema.

Database systems have several schemas, partitioned according to the levels of abstraction. The physical schema describes the database design at the physical level, while the logical schema describes the database design at the logical level. A database may also have several schemas at the view level, sometimes called subschemas, that describe different views of the database.

The logical schema is by far the most important in terms of its effect on application programs, since programmers, construct applications by using the logical schema. The physical schema is hidden beneath the logical schema and can usually be changed easily without affecting application programs. Application programs are said to exhibit physical data independence if they do not depend on the physical schema and thus need not be rewritten if the physical schema changes.