VIDYA BHAWAN BALIKA VIDYAPITH ,LAKHISARAI

INFORMATION TECHNOLOGY FOR CLASS 12

(Study materials)

RAUSHAN DEEP DATE:-11.04.2020 (SATURDAY)

TYPES OF USER DATABASE MANAGEMENT SYSTEM

DBMS is used by many types of users depending on their requirements and interaction with the DBMS. There are mainly four types of users:

1. End Users: Users who use the database for querying, modifying and generating reports as per their needs. They are not concerned about the working and designing of the database.They simply use the DBMS to get their task done.

2. Database Administrator (DBA): As the name implies, the DBA administers the database and the DBMS. The DBA is responsible for authoring access, monitoring its use, providing technical support, acquiring software and hardware resources.

3. Application Programmers: Application programmes write application programs to interact with the database. These programs are written in high level languages and SQL to interact with the database.

4. System Analyst: System analyst determines the requirements of the end users and then develops specifications to meet these requirements. A system analyst plays a major role in the database design and all the technical, economic and feasibility aspects. The need of DBMS itself explains the advantages of using a DBMS. Following are the advantages of using a DBMS:

1. Reduction in Redundancy: Data in a DBMS is more concise because of the central repository of data. All the data is stored at one place. There is no repetition of the same data. This also reduces the cost of storing data on hard disks or other memory devices.

2. Improved Consistency: The chances of data inconsistencies in a database are also reduced as there is a single copy of data that is accessed or updated by all the users.

3. Improved Availability: Same information is made available to different users. This helps sharing of information by various users of the database.

4. Improved Security: Though there is improvement in the availability of information to users, it may also be required to restrict the access to confidential information. By making use of passwords and controlling users' database access rights, the DBA can provide security to the database.

5. User Friendly: Using a DBMS, it becomes very easy to access, modify and delete data. It reduces the dependency of users on computer specialists to perform various data related operations in a DBMS because of its user friendly interface.

Limitations of using DBMS Approach

The two main disadvantages of using a DBMS:

1. High Cost: The cost of implementing a DBMS system is very high. It is also a very timeconsuming process which involves analyzing user requirements, designing the database specifications, writing application programs and then also providing training.

2. Security and Recovery Overheads: Unauthorized access to a database can lead to threat to the individual or organization depending on the data stored. Also the data must be regularly backed up to prevent its loss due to fire, earthquakes, etc. Hence the DBMS approach is usually not preferred when the database is small, well defined, less frequently changed and used by few users. Various types of databases

have been developed. One of them was relational database developed by E.F Codd at IBM in 1970. It is used to organize collection of data as a collection of relations where each relation corresponds to a table of values. Each row in the table corresponds to a unique instance of data and each column name is used to interpret the meaning of that data in each row. For example, consider EMPLOYEE table in Figure 1.5(a).
Each row in this table represents facts about a particular employee. The column names – Name, Employee_ID, Gender, Salary and Date_of_Birth specify how to interpret the data in each row.

NAME	EMPLOYEE-ID	GENDER	SALARY	DATE OF BIRTH
NEHA BAHRTI	2142	FEMALE	32000	25.01.1990
PRASHANT KUMAR	2148	MALE	28000	22.08.1991
VISHWJEET VERMA	2140	MALE	27000	18.07.1990
SHIKHA SINGH	2138	FEMALE	34000	23.07.1993

Employee Table

In relational model,

- A row is called a **Tuple**.
- A column is called an **Attribute**.
- A table is called as a **Relation**.
- The data type of values in each column is called the **Domain**.
- The number of attributes in a relation is called the **Degree** of a relation.
- The number of rows in a relation is called the **Cardinality** of a relation.
- **Relation Schema** R is denoted by R (A , A , A …, A) where R is the relation name $_{1\,2\,3,\,n}$ and A , A , A ,….A is the list of attributes. $_{1\,2\,3\,n}$
- **Relation State** is the set of tuples in the relation at a point in time. A relation state r of relation schema R (A , A , ..., A), denoted r(R) is a set of n-tuples r = {t , $_{1\,2\,n\,1}$ t ,...., t }, where each **n-tuple** is an ordered list of values t = <v , v , ...,v >, where $v_{2\,m\,1\,2\,n\,i}$ is in domain of A or is NULL. Here n is the degree of the relation and m is the cardinality of the relation.

Hence in Figure

- EMPLOYEE table is a relation.
- There are three tuples in EMPLOYEE relation.
- Name, Employee_ID, Gender, Salary, Date_of_Birth are attributes.
- The domain is a set of atomic (or indivisible) values. The domain of a database attribute is the set of all the possible values that attribute may contain. In order to specify a domain, we specify the data type of that attribute. Following are the domain of attributes of the EMPLOYEE relation:
  (a) Name – Set of character strings representing names of persons.
  (b) Employee_ID–Set of 4-digit numbers
  (c) Gender – male or female
  (d) Salary – Number
  (e) Date_of_Birth – Should have a valid date, month and year. The birth year of The employee must be greater than 1985. Also the format should be dd-mm-yyyy.
- The degree of the EMPLOYEE relation is 5 as there are five attributes in this relation.
- The cardinality of the EMPLOYEE relation is 3 as there are three tuples in this relation.
- Relation Schema – EMPLOYEE (Name, Employee_ID, Gender, Salary, Date_of_Birth)

- Relation State –{<vishwjeet verma, 2140 ,Male,27000,18-07-1990>,<Prshant kumar , 2148, Male, 28000, 22-08-1991>,<Neha bharti, 2142, Female, 32000, 25-01-1990>,<Shikha singh, 2138, Male, 34000,23-.7-1993>}

RELATIONAL MODEL CONSTRAINTS

Constraints, are restrictions on the values, stored in a database based on the requirements. For example, in the relation EMPLOYEE, the Employee_ID must be a 4-digit number,the Date_of_Birth must be such that the birth year > 1985. We describe below various types of constraints in Relational model:

1. Domain Constraint: It specifies that the value of every attribute in each tuple must be from the domain of that attribute. For example, the Employee_ID must be a 4-digit number. Hence a value such as "12321" or "A234" violates the domain constraint as the former is not 4-digit long and the latter contains an alphabet.

2. Key Constraint: Before we can explain this constraint, we need to describe the terms superkey, key, candidate key and primary key.

(i) *Superkey* is a set of attributes in a relation, for which no two tuples in a relation state have the same combination of values. Every relation must have at least one superkey which is the combination of all attributes in a relation. Thus for the EMPLOYEE relation, following are some of the superkeys:

A. {Name, Employee_ID, Gender, Salary, Date_of_birth} - default superkey consisting of all attributes.
B. {Name, Employee_ID, Date_of_Birth}
C. {Employee_ID, Gender, Salary}
D. {Name, Employee_ID,Gender}
E. {Employee_id}

********** ********** **********

RAUSHAN DEEP
PGT (IT)
11.04.20XX