आईएफटीएम विश्वविद्यालय, मुरादाबाद, उत्तर प्रदेश

IFTM University, Moradabad, Uttar Pradesh

NAAC ACCREDITED

E-Content

IFTM University, Moradabad

More Create Blog Sign In

Information Technology

SATURDAY, FEBRUARY 27, 2021

Data Base Management System

Database Management System (DBMS) (MBA IT-02)

UNIT-I

Data: Data is a collection of information. In other word we can say that the facts that can be recorded and which have implicit meaning known as 'data'.

Ex: Customer -

1. customer_name
2. customer _no.
3. customer _city.

Database: Collection of that interrelated data. These data can be stored in tables form. Ex: Customer database consists the fields as c_name, c_no, and c_city

customer_name	**customer _no**	**customer _city**
Ram	**45662**	**Moradabad**
Shyam	**78598**	**Delhi**

-

DBMS DBMS stands for **Database Management System**. We can break it like this

DBMS = Database + Management System.

As we discuss that by data we mean useful information. In other ways Data as a general concept refers to the fact that have some accessible information or knowledge is represented or coded in some form suitable for better usage or processing.

Database is a collection of that data and Management System is a set of programs to store and

Blog Archive

▼ 2021 (4)

▼ February (4)

Design and Analysis of Algorithm

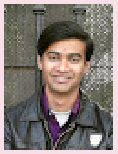
Data Base Management System

Computer Application in Management

► 2020 (1)

► 2009 (1)

About Me



manish r pandey

View my complete profile

retrieve those data. In view of this we can characterize DBMS like this:

Database management system (DBMS) as a collection of interconnected data and set of programs to update, store & access that data in a simple and effective way.

“DBMS is general purpose application software which create delete and update data from database.”

Need of DBMS

Database systems are fundamentally developed for large amount of data. When managing enormous amount of information, there are two things that require improvement:

The way we store the data and **the way we access data**

Storage of Data: According to the philosophy of database systems, the data must stored in such a way that it acquires lot less space as the redundant data (duplicate data) has been removed earlier than storage. Let’s take an example to clear this: In a financial framework, assume a client is having two accounts, one is current account and another is pay account. Suppose bank stores current record information at one spot and pay account information at somewhere else, all things considered if the client data, for example, client name, address and so forth are put away at the two places then this is only a wastage of capacity (excess/duplication of information), to compose the information in a superior way the data ought to be put away at one spot and both the records ought to be connected to that data some way or another. Something very similar we accomplish in DBMS.

Fast Retrieval of data: Along with storing the data in an optimized and systematic manner, it is also important that we retrieve the data quickly when needed. Database systems ensure that the data is retrieved as quickly as possible.

Purpose of Database Systems

The main purpose of database systems is to **deal with the data**. Lets take an example of a college which has data of students, faculty, dept, courses, books etc. To manage this data we need to store this data somewhere where we can add new data, delete unused data, update outdated data, retrieve data, to perform these task on data we need a system called Database management system that allows us to store the data in such a way so that all these tasks can be performed on the data efficiently.

Applications ar

eas of DBMS

Applications areas where Database Management Systems are used:

In the field of telecommunication: There is an information database to monitors the data with respect to calls made, network utilization, client subtleties and so on. Without the information database frameworks it is difficult to keep up that enormous measure of information that continues refreshing each millisecond.

In the field of organization: Where it is an manufacturing organization, stockroom or circulation focus, everyone needs an information base to keep the records of intricate details. For instance conveyance focus should monitor the item units that provided into the middle just as the items that got conveyed out from the appropriation community on every day; this is the place DBMS comes into picture.

In the field of Banking System: For putting client information, following everyday credit and charge exchanges, producing bank proclamations and so forth. This work has been finished with the assistance of Database frameworks.

In the field of Sales: To store client data, creation data and receipt subtleties and sales data we need database system.

In the field of Airlines: To travel though airlines, we reserve early, this booking data alongside flight plan is put away in information database.

In the field of Education: Information database frameworks are oftentimes utilized in schools and universities to store and recover the information with respect to understudy subtleties, staff subtleties, course subtleties, test subtleties, finance information, student subtleties, expenses subtleties and so forth. There is a enormous related information that should be put away and recovered in an effective way.

In the field of E-Commerce: You should know about the web based shopping sites, for example, Snapdeal, Amazon, Flipkart and many more. These websites store the item data, our addresses and inclinations, credit subtleties and give us the significant rundown of items dependent on our question. This includes a Database the board framework.

Database System Vs File System

In traditional approach, before to computer, all information and data were stored in papers. At that time when we need data, we used to look through the papers. After the invention of computer all data and information were stored in files.

File System

File processing system was an early endeavor to modernize the manual documenting system. A file system is a strategy for putting and arranging PC files and the information they contain to make it simple to discover and get to them. File system may utilize a capacity gadget, for example, a hard disk.

So a file system is a process that oversees how and where information on a storage device, normally a hard disk drive (HDD. It is an intelligent disk that deals with a disk's inward activities as it identifies with a PC and is conceptual to a human client.

A file system commonly oversees activities, for example, storage management, naming of file, directories and folders, metadata, retrieval rules and privileges.

There are various types of file system. Every type has distinctive structure and foundation, properties of speed, adaptability, security, size etc. Some file system has been intended to be utilized for explicit applications. For instance, the ISO-9660 file system is planned explicitly for optical discs.

Advantage of DBMS over file system

There are several advantages of Database management system over file system. Few of them are as follows:

Now we will describe in detail how DBMS is different from traditional file system and what its limitations were.

DBMS Architecture

Database management systems architecture will assist us with understanding the segments of database and the connection among them.

The architecture of DBMS relies upon the PC framework on which it runs. For instance, in a client-server DBMS architecture, the database system frameworks at server machine can run a few requests made by client machine.

Types of DBMS Architecture

Types of DBMS architecture are as follows:

1. Single tier architectu

re

2. Two tier architecture

3. Three tier architecture

1. Singl

e tier architecture

In this kind of architecture, the database is rapidly accessible on the user-machine, any request made by user doesn't need a connection tp play out the activity on the database.

For instance, lets state you need to get the records of worker from the database and the database is accessible on your PC framework, so the srequest to get representative subtleties will be finished by your PC and the records will be brought from the database by your PC also. This type of framework is known as local database system.

2. Two tier architecture

In two-tier architecture, the Database system is there at the server machine and the DBMS application is there at the client machine, these two machines are linked with each other by a reliable network as shown in the two-tier diagram. When user machine makes a request to get to the database present at server utilizing an inquiry language like sql, the server play out the request on the data set and returns the back to the user.

For the connection between server and client application connection interface such as JDBC, ODBC are used.

3. Three tier architecture

In three-level architecture, another layer is available between the user machine and server machine.

In this design, the user application doesn't discuss straight forwardly with the database system present at the server machine, rather the user application connect with server application and the server application inside connect with the database system present at the server machine.

DBMS – Three Leve

l Architecture

This architecture consists of three levels as follows:

1. External level
2. Conceptual level
3. Internal level

1. External level

It is likewise called see level. The explanation this level is classified "see" is on the grounds that few clients can see their ideal information from this level which is inside brought from database with the assistance of conceptual and internal level connectivity.
External level is also called "high level" of the Three Level DBMS Architecture.

2. Conceptual level

It is also called logical level. The entire plan of the database, for example, relationship among data, types of data, description of data and many more things are described in this level.

Database requirements/constraints and security are additionally actualized in this level of architecture.

DB

MS Three Level Architecture

3.Internal level

This level is also known as physical level. This level defines how the data is stored actually in the storage devices .allocating space to the data is also key responsibility of this level. This is the lowest level of the architecture.

Databases

ystem concepts

DATA Model

Logical structure of Database is Data model. Data models depict how data is related to one another and how they are handled and put inside the system.

Types of Data Models

There are different types of data models in DBMS.

Object based logical Models-Describe data at the conceptual and view levels.

1. E-R Model

2. Object oriented Model

Record based logical Models

1. Relational Model
2. Hierarchical Model
3. Network Model

Physical Data Models

<u>Data Abstraction</u> **-** Abstraction is one of the main principle of database system. Hiding unimportant details from client and giving dynamic perspective on data to clients, helps in simple and effective client-database association. This cycle of hiding unessential details from client is called data abstraction.

Example: Let's say we are saving user information in a user table. At **physical level** these data can be described as blocks of storage (bytes, gigabytes, terabytes etc.) in memory. These details are generally hidden from the programmers.

Schemas: The description of a database is known as the database schema, which is specified when we design the database and is not expected to change generally. Most data models have sure conventions for displaying schemas as diagrams. A displayed schema is called a schema diagram. Some examples of schema are as follows:

fig-schema

DBM

S Instance

The data stored in database at a particular moment of time is known as instance of database. The real data in an data base may change every now and again.

For example, lets say we have a table employee in the database, today the table has 100 employee, so today the instance of the database has 100 records. Lets say we are going to add another 100 employees in this table by tomorrow so the instance of database tomorrow will be

200 records in table. So we can say that at a particular moment the data stored in database is called the instance, that changes over time when we add /delete data from the database.

DBMS languages

In DBMS we need to update/access/read/store data. We can perform these operations with the help of DBMS languages. Database languages are utilized to read, update and store information in a data base. There are several different DBMS languages that can be used for this motive; one well-known language of them is SQL (Structured Query Language).

Types of

DBMS languages:

DBMS Languages

Data Definition Language (DDL)

Data Manipulation Language (DML)

Data Control language (DCL)

Transaction Control Language (TCL)

Data Definition Language (DDL)

We use DDL (Data Definition Language) to specifying the database schema. It can perform very functions like table creation, defining schema, defining indexes, applying constraints and many more in database. Let's see the operations that we can perform on database using DDL:

CREATE- is used to create the database instance.

ALTER-is used to alter the structure of database .

DROP- is used to drop database instances.

TRUNCATE- is used to delete tables in a database instance.

RENAME- is used to rename database instances.

DROP- is used to drop objects from database such as tables.

Comment- is used to Co

mment.

```
CREATE TABLE table_name (
column1 datatype,
column2 datatype,
column3 datatype,   ....
);
CREATE TABLE Student (
   Roll_No int,
   Name varchar(255),
   F_Name
```

```
    varchar(255),
    Address varchar(255),
    City varchar(255)
  );
```

For updating and defining design of database we use these commands so that they are come under

Data Definition Language.

Data Manipulation Language (DML)

For accessing data and manipulating data in a database we use DML (Data Manipulation Language).Under DML following operations are peformed:

SELECT- is used to read records from table(s).

INSERT- is used to insert record(s) into the table(s).

UPDATE-is used to update the data in table(s).

DELETE – is used to delete all the records from the table.

INSERT **INTO *table_name* (*column1*, *column2*, *column3*, ...)**

VALUES **(*value1*, *value2*, *value3*, ...);**

INSERT **INTO *student* (*stu_name*, *stu_rollno*, *stu_course* ...)**

VALUES **(*ram*, *19005221*, *BCA*, ...);**

SELECT *column1*, *column2*, ...

FROM *table_name*;

```
SELECT
stu_name,
stu_rollno
FROM
student;
```

Data Control Language (DCL)

For granting and revoking user access on a database we use DCL (Data Control Language)

- GRANT – is used to grant access to user.
- REVOKE – is used to revoke access from user.

Transaction Control Language (TCL)

Using TCL (Transaction Control Language) we can rollbacked or performed the changes in the database that we made using DML commands.

COMMIT- is used to persist the changes made by DML commands in database.

ROLLBACK- is used to rollback the changes made to the database.

Entity Relationship Diagram – ER Diagram in DBMS

An **Entity–relationship model (ER model)** definess the structure of a database that is schema with the help of a diagram, which is called **Entity Relationship Diagram (ER Diagram)**. An ER model is a plan or outline of a database that can later be actualized as a database.
There are two main components of E-R model that are: entity set and relationship set.

ER Diagram example:

Here we are showing a simple ER-Diagram. In the diagram we are have two entities named

Student and College and both entity are linked with each other by a relationship. Attributes of student entity are Stu_Id, Stu_Name & Stu_Addr and attributes of college entity are Col_Id, Col_Name.

Geometric shapes and their meaning in an E-R Diagram are as follows.

Rectangle: By rectangle shape we represent an entity sets. **Ellipses**: By ellipses or oval shape we represents an Attributes. **Diamonds**: By diamond shape we represent a relationship Set

Lines: By lines we link entity sets to relationship set and attributes to entity set.

Double Ellipses: By double ellipses we represent multivalued Attributes. **Dashed Ellipses**: By double ellipses we represent derived Attributes.

Double Rectangles: By double rectangles we represent weak Entity Sets.

Double Lines: By double lines we represent total participation of an entity in a relationship set.

ER Diagram components

1. Entity

An entity is a real life object or concept. In an ER diagram we represent entity as rectangle. For example: In the above ER diagram two entities are there one is Student and another is College which is represented by a rectangle.

Weak Entity:

Weak entity is an entity can’t be uniquely identified with the help if its own attributes and depends on the relationship with other entity for this purpose. We represent weak entity by a by a double rectangle box. For example – if we want uniquely identify a bank by only bank account. This cannot be possible cannot be without having the bank in which the account exists, so in this case bank account is a weak-entity.

2. <u>Attribute</u>

An attribute of an entity describes the property of that’s entity. We represent an attribute by Oval in an ER diagram. Four types of attributes are there in ER-Model as

1. Key attribute

2. Composite att

ribute

3.Multivalued attribute

4.Derived attribute

1.Key att

ribute:

A key attribute of an entity has the whole sole ability to uniquely identify that entity from an entity set. Lets take an example, we can uniquely identify a student by its attribute roll no from a set of students. We represent Key attribute by an oval same as other attributes however we underlined the text of key attribute.

2. Composite attribute:

Composite attribute is a combination of other attributes. For example as we can see in below diagram that in student entity address is a composite attribute because address is collection of other attributes such as pin code, state, country in student entity.

3. Multival

ued attribute:

Multivalued attribute is the attribute that can keep multiple. We represented a multivalued attribute with **double ovals** in an ER Diagram. For example as we can see in the diagram below

– A student can have more than one phone numbers so the phone number attribute is multivalued.

4. Derived attribute:

A derived attribute is the attribute whose value is not stable over time i.e. dynamic and which can derived from another attribute. We represent it by **dashed oval** in an ER Diagram. For example see in below diagram – student age is a derived attribute because it changes over time and can be derived from another attribute i.e (Date of birth) of that student.

E-R diagram with key, composite, multivalued and derived attributes**:**

3. Relationship

A relationship set describes the relationship between entities that they hold. We represent it by diamond shape in ER diagram; it shows the relationship among entities. There are four types of relationships in ER-Model:

1. One to One
2. One to Many
3. Many to One
4. Many to Many

1. One to One Relationship

When a single instance of an entity is connected to a single instance of another entity then this type of relationship is called one to one relationship. Let’s take an example, a person can have only one passport and a passport is issued to one person.

Example: One student can have one roll no. And one roll no can be assigned to one student.

2. One to Many Relationship

When a single instance of an entity is connected to more than one instances of another entity then this type of relationship is called one to many relationship. For example – a customer can

purchase many orders but a order cannot be placed by many customers.

For example, one class is consisting of multiple students and multiple students may have in a class.

3. Many to One Relat

ionship

When more than one instances of an entity is connected to a single instance of another entity then this type of relationship is called many to one relationship. For example – many students studies in a single college but a student cannot study in many colleges at the same time.

For example, many students belong to the same class.

4. Many to Many Relationship

When more than one instances of an entity is connected to more than one instances of another entity then this type of relationship is called many to many relationship. For example, a student can be allotted to many task or projects and a task or project can be issued to many students.

For example, group of students are connected to multiple faculty members, and a faculty members can be associated with multiple students.

Example of an ER diagram for College Database

Example of an ER diagram for Company Database

Mapping Constraints

- A mapping constraint is a data constraint that describes the number of entities to which another entity can be connected by a relationship set.
- It is very important in describing the relationship sets that engage more than two entity sets.
- For relationship set R on an entity set R and S, there are four probable mapping cardinalities which are as follows:
  - One to one (1:1)
  - One to many (1:M)
  - Many to one (M:1)
  - Many to many (M:M)

- One to One: An entity of entity-set R can be connected with at most one entity of entity- set S and an entity in entity-set S can be connected with at most one entity of entity-set R.
- One to Many: An entity of entity-set R can be connected with any number of entities of entity-set S and an entity in entity-set S can be connected with at most one entity of entity-set R.
- Many to One: An entity of entity-set R can be connected with at most one entity of entity-set S and an entity in entity-set S can be connected with any number of entities of entity-set R.
- Many to Many: An entity of entity-set R can be connected with any number of entities of entity-set S and an entity in entity-set S can be connected with any number of entities of entity-set R.

Posted by manish r pandey at 11:48 PM

No comments:

Post a Comment

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)