Chapter-1

INTRODUCTION

Data

- Data is Raw, unorganized facts that need to be processed.
- Data itself is meaningless and worthless.
- A meaningful information is only possible after processing such a raw data.

Information

- When data is processed, organized, structured or presented in a given context so as to make useful, is called information.
- It is output of data processing operation.
- For example, marks obtained by students and their roll numbers form data where their mark sheet is the information.

Database

- A Database is a collection of inter-related data organized in a way that data can be easily accessed, managed and updated.
- For example: The college Database organizes the data about the admin, staff, students and faculty etc.
- Using the database, you can easily retrieve, insert, and delete the information.

Database Management system

- Database management system (DBMS) is a collection of programs that allows to create and maintain databases.
- DBMS provides an interface to the user so that the operations on database can be performed using the interface.

Some examples of DBMS are:

- MySQL
- Microsoft SQL-Server
- Oracle
- IBM DB2
- PostgreSQL etc.

Operations that allow by DBMS

- **Data Definition:** Creating, modifying, and deleting the structure of the database, including tables, fields, and relationships.
- **Data Modification:** It is used for the insertion, modification, and deletion of the actual data in the database.
- **Data Retrieval:** DBMS allows users to fetch data from the database.
- **Data Security**: Ensuring the security of the database and its contents, such as controlling access to the data, protecting against unauthorized access, and ensuring the integrity and consistency of the data.
- **Data Administration:** Managing the overall performance and efficiency of the database system, including monitoring, and tuning the system, optimizing queries, and managing resources such as disk space and memory.

Objective of DBMS

- Providing mass storage of relevant data.
- Make easy access to data for authorized users.
- Eliminating data redundancy.
- Providing data integrity.
- Providing security with user access privilege.
- Allow multiple users to be active at one time.
- Serving different types of users.
- Provide prompt response to user requests for data.
- Combining interrelated data to generate reports.
- Protect the data from physical harm and un-authorized systems.
- Allowing the growth of the database system.
- Make the latest modifications visible to all the database available immediately.

Components of DBMS

A DBMS consists of several components and each component has a specific function in the database management system environment. Although DBMS consists of many components, Hardware, Software, Data, Database Access Language, and Procedures are the five major components of DBMS.

Hardware

Hardware refers to the set of physical electronic devices such as computers and their components, which include I/O devices for data in or out, storage devices, and other physical components. Hardware, with the help of software, helps establish an interface between a computer and a real-world system to help users create, access, or maintain databases.

One cannot implement or use DBMS without hardware. Databases are created, accessed, and/or managed using a range of machines from microcomputers to mainframes. When we run any database management system, such as Oracle, MySQL, etc. on our PC, then the computer parts like the mouse, keyboard, RAM, ROM, hard disk all become part of DBMS hardware.

Software

Software is one of the main components of a database management system. It refers to the set of multiple programs used to control and manage the entire database and its structure.

The software establishes an easy-to-use interface for users to control the hardware and to create, store, access, and/or update data in the database. All requests made by users for database management are handled and processed by the DBMS software. DBMS software usually understands the 'Database Access Language' and converts it into database commands or instructions to be executed/run on the respective database.

Data

Data is another important component in a database management system environment as DBMS mainly exists to collect, store, access, and process data. Using a DBMS, the database is first created, constructed, or defined. After that, the desired data is stored, accessed, or updated to/from the created databases.

The primary reason behind the introduction of a database management system was to store and maintain data within the database. A typical database stores both actual (or operational) data and metadata (information or details about the actual data for better understanding).

When we store specific data (for example, a person's name) in the database, the DBMS also stores additional information such as when and where the data was stored, the size of the data, whether the data is relative or independent, etc. All this additional information about the actual data (i.e. the name) is collectively called metadata.

Procedures

Procedures refer to the rules and instructions that help to design and use a database management system. This typically includes procedures such as assisting in setting up and installing a database management system (DBMS), logging in or out of the DBMS software, changing or modifying the database structure, managing databases, generating reports, backing up databases, etc.

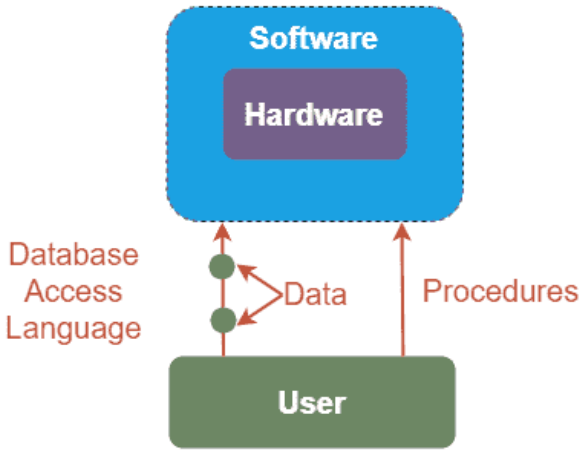
The basic purpose of procedures is to help guide users to the operation and management of database systems.

Database Access Language

Database Access Language is a simple language that allows users to write commands to perform desired operations on the data stored in the database. Everything like creating a new database, creating tables, inserting data, accessing, updating, or deleting stored data can be done using database access language.

The user has to write a set of appropriate commands in the database access language and submit it to the DBMS for further execution. The DBMS translates the given commands, processes the data accordingly, and generates or displays the result in a readable form to the user.

The following diagram is a structural visualization to explain how all the components of a DBMS fit together for users interacting with the database.



Application of DBMS

- **Telecom:** For keeping records of calls made, generating monthly bills, maintaining balances on prepaid calling cards, and storing information about the communication networks.
- **Manufacturing**: For management of product manufactured by companies such as supply chain and tracking production , inventories of items in stores, and orders for items.
- **Banking System**: For storing customer info, tracking day to day credit and debit transactions, generating bank statements etc.
- **Sales:** To store customer information, production information and invoice as well to track, manage and generate historical data to analyze the sales data.
- **Airlines:** For reservations, and schedule Information.
- **Education sector:** Used in schools and colleges to store and retrieve the data regarding student details, staff details, course details, exam details, attendance details, fees details etc.
- **Human Resource Management:** For information of employees such as salaries, performance reviews, job history etc.

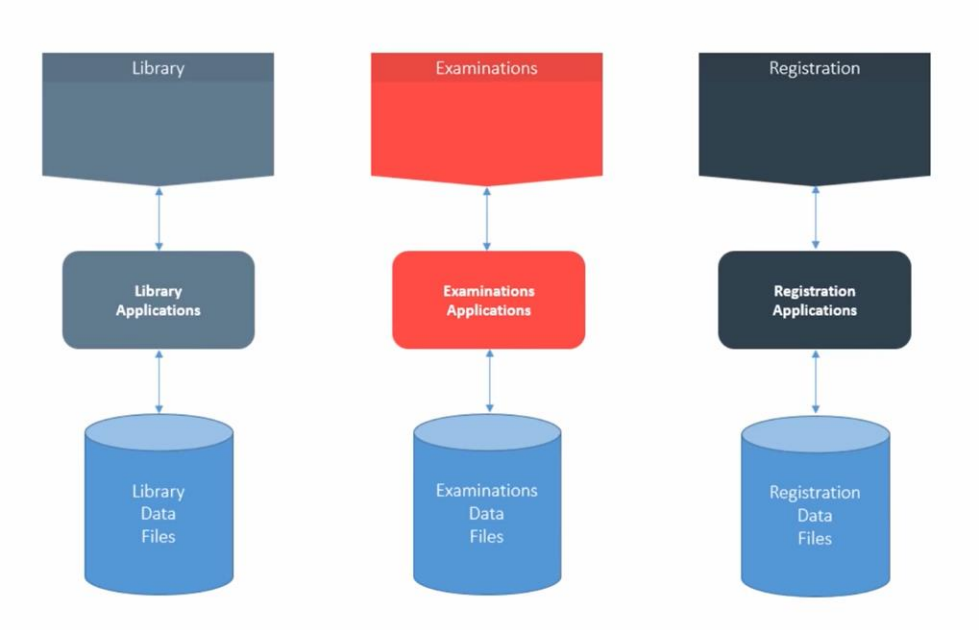
File processing system

In earlier days, data was stored manually, using pen and paper.

But after the computer was discovered, the same task could be done by using files. There are various formats in which data can be stored. e.g. Text files can be stored in .txt format.

Files are used to store various documents. All files are grouped based on their categories. The file names are very related to each other and arranged properly to easily access the files. In file processing system, if one needs to insert, delete, modify, store or update data, one must know the entire hierarchy of the files.

Let us consider the following example

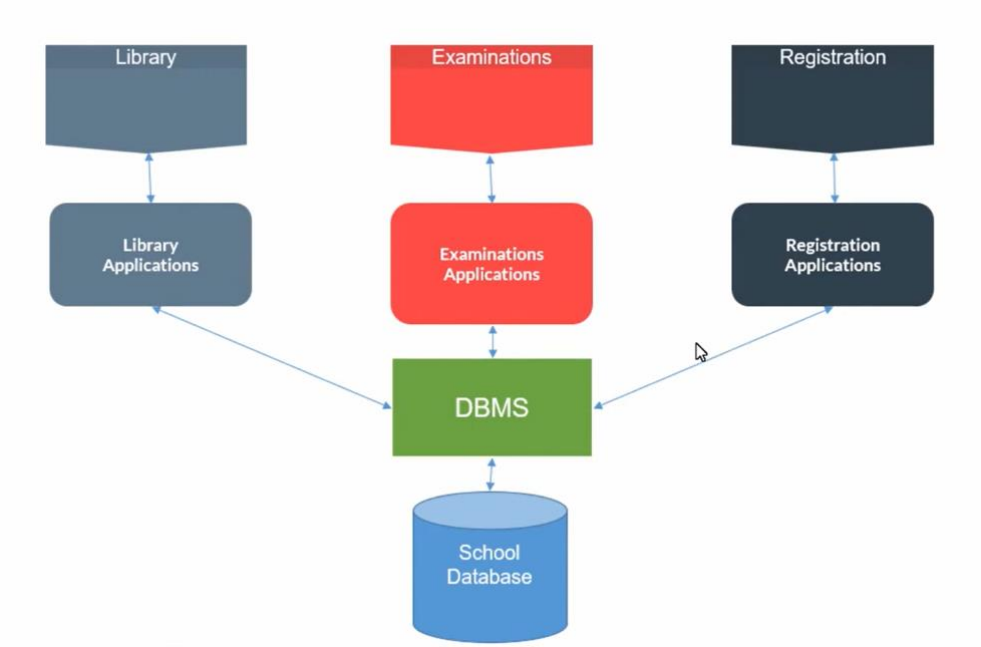


Examination File
1 Nadeem Ahmad city B+ totalMarks, 3 Sara khan city B+ totalMarks, 2 Muhammad Saim city B+ totalMarks,

Registration File
1 Nadeem Ahmad city B+ Admission date, 3 Sara khan city B+ Admission date, 2 Muhammad Saim city B+ Admission date,

- In the above figure we can see that each application contains data files.
- The Examination file will contain information regarding the student and their marks. Similarly, we have a registration file containing information regarding the student and the admission date. Student information is duplicated here, which leads to data redundancy.
- Another major drawback of traditional file processing is the non-sharing of data. It means, if different systems of organization are using some common data rather than storing it once and sharing it, each system stores separate files. This creates problems of redundancy and storage.

DBMS System



Database is shared collection of logically interrelated data designed to meet multiple users of an organization.

The typical database environment is as shown in the above figure. The figure shows different subsystems over application in an educational institution like library system, examination system and registration system. There are separate different application programs for every application over subsystem. However, data for all applications is stored in the same place in the database and all application program relevant data and users are managed by Database Management System (DBMS).

This introduces the major benefit of data sharing. That is data that is common among different applications need not to be stored repeatedly as was the case in the file processing environment.

Now from the above discussion we can analyze and conclude problem associated with file processing system and advantages of DBMS over file processing system.

Problems associated with file processing system

1) Data Redundancy

- Data redundancy refers to the duplication of data.
- Lets say we are managing the data of new admitted student. The same student details in such case will be stored in account, library and exam section etc. which will take more storage than needed.

2) Data inconsistency

- Data redundancy leads to data inconsistency.
- lets same example that we have taken above, where we store the information of student in multiple places.
- Now lets say student requests to change his city. If the city is changed at one place and not on all the records then this can lead to data inconsistency.

3) Data isolation

- Because data are scattered in various files, and files may be in different formats, writing new application programs to retrieve the appropriate data is difficult.

4) **Security problem**

- Not every user of the database system should be able to access all the data. for example a student in a college should not be able to see the payroll details of the teachers, such kind of security constraints are difficult to apply in file processing systems.

5) **Integrity problem**

- The data values stored in the data must satisfy certain types of consistency constraints. For example, CGPA of students cannot be greater than 4 in Pokhara university grading system.

6) **Lack of atomicity**

- Operations performed in the database must be atomic i.e. either the operation takes place as a whole or does not take place at all.

7) **Problem in concurrent Access**

- When a number of users operates on a common data in database at the same time then anomalies arise, due to lack of concurrency control.

8**) Slow access time**

- Direct access of files is very difficult and one needs to know the entire hierarchy of folders to get to a specific file. This involves a lot of time.

Advantages of DBMS over File Processing system/Advantages of DBMS

1) **Remove data redundancy:** Redundancy removed by data normalization. Removal of duplication of data saves storage and improves access time.

2) **Data security:** It is easier to apply access constraints in database systems so that only an authorized user can access the data. Each user has a different set of access thus data is secured from issues such as identity theft, data leaks and misuse of data.

3) **Data Integrity:** DBMS ensures data integrity by enforcing data validation rules, such as data type, range, and format, which helps to prevent data entry errors and ensure data accuracy.

4) **Data sharing:** In DBMS, the authorized users of an organization can share the data among multiple users.

5) **Scalability and flexibility:** DBMS systems are scalable, the database size can be increased and decreased based on the amount of storage required. It also allows addition of additional tables as well as removal of existing tables without disturbing the consistency of data.

6) **Data Backup and Recovery:** DBMS provides backup and recovery mechanisms to protect against data loss or corruption. This helps to ensure the availability and reliability of the data, even in the event of hardware or software failures.

7) **Fast response:** Database systems manages data in such a way so that the data is easily accessible with fast response times.

Disadvantages of DBMS

- It occupies a large space of disks and large memory to run them efficiently.
- Except MYSQL(which is open source), licensed DBMS are costly.
- DBMS can be complex and difficult to learn, especially for non-technical users.
- Can be vulnerable to security breaches if not properly configured and managed. This can lead to data loss or theft.

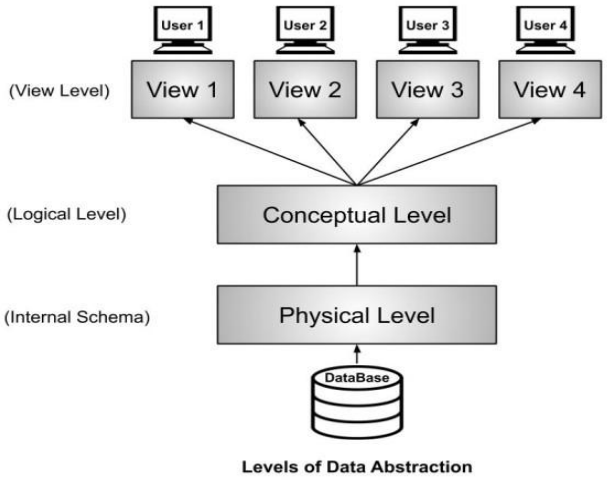
DBMS vs File processing system

DBMS	File processing system
A database management system is collection of programs that enables to create and maintain a database.	A file processing system is a collection of programs that store and manage files in computer hard-disk.
Data redundancy problem is less.	Data redundancy problem exist.
DBMS gives an abstract view of data that hides the details.	The file system provides the details of data representation and storage of data.
DBMS systems offers high security.	File systems provide less security in comparison to DBMS.
There is more data consistency because of process of normalization.	There is less data consistency in file processing system.

The problem of data isolation is not found in it.	Data are scattered in various files in different formats. Writing new program to retrieve appropriate data is difficult.
DBMS is more complex.	File processing system is less complex,
DBMS system provides backup and recovery of data even if it is lost.	It does not offer data recovery processes.
The data independence feature of DBMS allows separating data definition from program.	The data used for any file is dependent on program because file structure is defined in program code.

Data abstraction

- Data abstraction is a technique to hide the implementation details of a database that how a database is structured and how the data is stored physically.
- It allows database systems to provide abstract views to database users.
- It is a mechanism to hide the complexity of databases.
- Data abstraction simplifies users' interactions with the system.



Levels of Data Abstraction

There are three levels of abstraction.

1. Physical level
2. Logical level
3. View level

1. Physical Level

- It is the lowest level of abstraction, and it describes how a record are actually stored.
- At the physical level, complex level data structures are described in detail. (e.g. index, B-tree, hashing)

2. Logical Level

- This is the next highest level of abstraction.
- It describes what data are stored in databases and what relationship exists among them.

3. **View level**

- It is the highest level of abstraction. It describes only part of the entire database. It simplifies interaction with the system.
- It allows database system to provide many views for the same database. That is it allows each user/application to get a different perspective of the database.

Let's consider an example of a database system for managing employee information in a company.

Physical level: At the physical level, the database system might store the employee information in a specific file format optimized for fast access and efficient storage. It might also use disk blocks and indexing to manage the storage and retrieval of data.

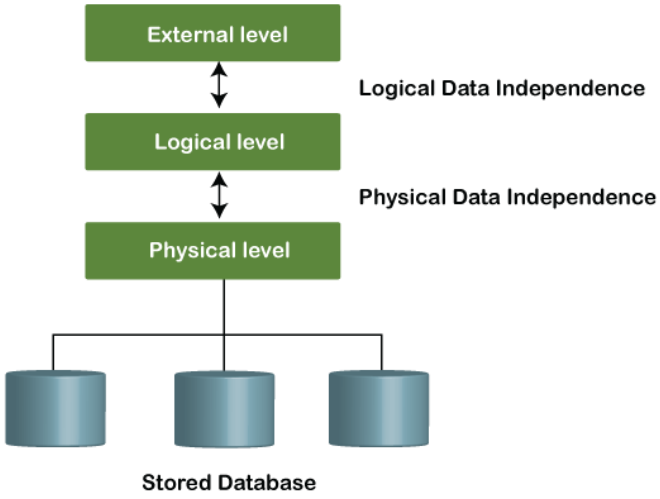
Logical level: At the logical level, the database system might define tables for storing employee information, with columns for employee ID, name, address, job title, salary, and other relevant data elements. Relationships might be defined between tables, such as a relationship between the employee table and a department table to track which department an employee belongs to.

View level: At the view level, different views of the data might be created for different users or applications. For example, HR department might have a view that shows employee names, job titles, and salaries for managing payroll, while the management team might have a view that shows employee performance metrics and career development plans for strategic planning.

Data independence

- Data independence can be explained using the three-schema architecture.
- Data independence is an ability to modify a schema definition in one level without affecting the schema definition in higher level.
- There are two types of data independence.

1. Logical data independence
2. Physical data independence



1. **Logical Data Independence**

- Logical data independence is the capacity to change the conceptual schema without having to change external schemas or application programs.
- Logical data independence is used to separate the external level from the conceptual view.
- If we do any changes in the conceptual view of the data, then the user view of the data would not be affected.
- E.g. Add/Modify/Delete a new attribute, entity or relationship is possible without a rewrite of existing application programs.

2. **Physical data independence**

- Physical data independence is the capacity to change the internal schema without having to change the conceptual schemas.
- With Physical independence, we can easily change the physical storage structures or devices with an effect on the conceptual schema.
- Due to Physical independence, any of the below changes will not affect the conceptual layer.
  - ✓ Using a new storage device
  - ✓ Switching to different data structures
  - ✓ Changing the location of database

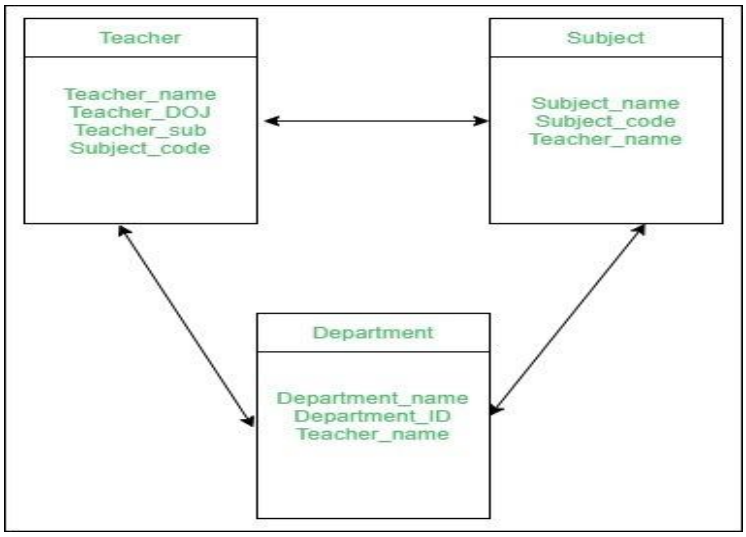
Assignment:

- ❖ *Differentiate between physical data independence and logical data independence.*

Schema and instances

Schema

- Schema is overall design of database.
- In other words, the basic structure of how the data will be stored in the database is called schema.
- In the following diagram, we have a schema that shows the relationship between three tables: Teacher, Subject and Department.
- The diagram only shows the design of the database, it doesn't show the data present in those tables.



There are three types of schema.

1. **Physical schema**: The design of a database at physical level is called physical schema, how the data stored in blocks of storage is described at this level and database administrator works at this level.
2. **Conceptual/Logical schema:** Design of database at logical level is called logical schema, programmers and database designers work at this level, at this level data can be described as certain types of data records gets stored in data structures, however the internal details such as implementation of data structure is hidden at this level (available at physical level).
3. **View schema**: Design of database at view level is called view schema. This generally describes end user interaction with database systems.

Instances

- The data stored in database at a particular moment of time is called instance of database.
- Database schema defines the attributes in tables that belong to a particular database.
- The value of these attributes at a moment of time is called the instance of that database.
- The instances may be changed by certain operations such as addition, deletion etc.
- It may be noted that any search query will not make any kind of change in the instances.

For better understanding of schema and instances Let us consider simple example

An employee table in database exists with the following attributes:

Eid	Name	Position	Salary

- This is the schema of the employee table.
- Schema defines the attributes of tables in the database.
- It shows the design of the database, it doesn't show the data present in those tables.

Instances

- we have seen the schema of table "employee".
- Let's look at the table with the data now.

Eid	Name	Position	Salary
1	Hari	Manager	55625
2	Ramesh	Helper	12550
3	Gita	CEO	95450
4	Sita	HOD	68325
5	Gopal	Secretary	25450

- At this moment the table contains 5 rows (records). This is the current instance of the table "employee" because this is the data that is stored in this table at this moment of time.
- Instances can be changed if we perform certain operations such as addition, deletion etc.

Concept of DDL, DML, DCL and TCL

DDL (Data Definition Language)

- It consists of SQL commands that can be used to define the database schema.
- It is used to create and modify the structure of database objects in database but not data.
- DDL is used by database designers and Database Administrators (DBA) but not user.

CREATE- is used to create the database or its objects (like table, function, views, store procedure etc).

Example:

1. CREATE DATABASE db_eemc; *//creates database named db_eemc*
2. CREATE TABLE employee_info(eid int, name varchar(50),position varchar(50),department varchar(50);

DROP-is used to delete objects from the database.

Example:

1. DROP DATABASE db_eemc; *//deletes database named db_eemc*
2. DROP TABLE employee_info; *//deletes table named employee_info from database*

ALTER- is used to Alter the structure of the database.

Eg.Add/Rename/Delete/Change columns in tables of database.

1. ALTER TABLE employee_info
   ADD salary decimal (10,2);
   //column named salary will be added in table named employee_info
2. ALTER TABLE employee_info
   DROP COLUMN salary;
   //column named salary will be deleted from table named employee_info

TRUNCATE–is used to remove all records from a table, including all spaces allocated for the records are removed.

Example:

TRUNCATE TABLE employee_info; *//All records will be deleted but not table*

Data Manipulation Language (DML)

It deals with the manipulation of data present in database.

INSERT – is used to insert data into a table.

Example:

INSERT INTO employee_info

VALUES(1,'Hari','manager', 'sales');

UPDATE – is used to update existing data within a table.

Example:

UPDATE employee_info

SET department='marketing'

WHERE eid=1;

//updates the department to marketing in employee_info table whose eid is 1

DELETE – is used to delete records from a database table.

Example:

DELETE FROM employee_info

WHERE eid=1;

// delete a row from table named employee_info which has eid=1

SELECT - is used to select (retrieve) data from a database table.

Example:

SELECT name,position

FROM employee_info

WHERE deparment='marketing';

//selects the name and position of employees whose department is marketing from employee_info table

Data Control Language (DCL)

- ✓ DCL includes commands such as GRANT and REVOKE which mainly deals with the rights, permissions, and other controls of the database system.
- ✓ It is used to give and withdraw specific privileges (as defined by query) to the user in a multi-user database.
- ✓ By setting up the permission, user can be prevented from unauthorized access to the database.

DCL commands are:

GRANT

- ✓ This is a SQL command which is used to provide privileges/permissions to modify and retrieve database objects like tables, views etc.
- ✓ It can be used to grant SELECT, INSERT, UPDATE, DELETE etc. privileges to a user.

Syntax:

GRANT <privilege list> on <relation or view> to <user>;

Example:

GRANT INSERT, SELECT on employee_info to ram;
GRANT SELECT, UPDATE,DELETE on employee_info to sita;

REVOKE

This command is used to withdraw user's access privileges given by using the GRANT command.

Syntax:

REVOKE <privilege list> on <relation or view> from <user>;

Example:

REVOKE INSERT on employee_info from ram;
REVOKE DELETE on employee_info from sita;

Transaction control Language (TCL)

- ✓ Transaction Control Language (TCL) is a set of special commands that deal with the transactions within the database.
- ✓ The changes made by DML commands are either committed or rolled back by TCL commands.
- ✓ There is another TCL command that can place a save point in the transactions which makes it possible to rollback all the transaction till the last save point.

COMMIT

Commit command make the changes made to the database permanent.

Syntax:

```
COMMIT;
```

Here's the syntax demonstrating the use of the COMMIT command with a transaction in MySQL:

```
START TRANSACTION;

{a set of SQL statements};

COMMIT;
```

The parameters used in the syntax are:

- ✓ START TRANSACTION: It is used for marking the beginning of changes or operations in a transaction.
- ✓ {a set of SQL statements}: It is used for mentioning the task that is supposed to be completed.
- ✓ COMMIT: It is used to save transactional changes made by SQL statement

Example:

```
START TRANSACTION;
DELETE FROM student_info
WHERE sid = 11;
COMMIT ;
```

ROLLBACK

- ✓ Rollback command is used to undo the changes that have been made to the database temporarily.
- ✓ The important point to note here is that the changes saved using COMMIT command cannot be undone using ROLLBACK command.

Example:

```
 UPDATE student_info SET location='Dharan'
WHERE name='ram';
ROLLBACK;
```

SAVEPOINT

It's used to roll back a transaction to a specific point rather than the complete transaction.

Syntax:

SAVEPOINT SavepointName;

- ✓ Among all transactions, this command is exclusively used to create SAVEPOINT.
- ✓ ROLLBACK is a command that is used to undo a set of transactions.

The syntax for rollback to savepoint command:

ROLLBACK TO SavepointName;

Example:

```
UPDATE student_info
SET program = 'BBA'
WHERE sid = 5;
SAVEPOINT A;

UPDATE student_info
SET name = 'ram'
WHERE location = 'pokhara;
SAVEPOINT B;
```

Now if we want to roll back the certain DML commands, we can do so by using Rollback like this:
This will rollback the transaction till savepoint A;
Rollback to A;

Database Administrator (DBA)

- Database administrator is a person who has central control of both the data and program accessing that data.
- Coordinates all the activities of the database system; the database administrator has a good understanding of the enterprise's information resources and needs.
- The responsibility of a database administrator varies and depends on the job description, corporate and Information Technology (IT) policies and the technical features and capabilities of the DBMS being administrated.

The functions of DBA

1. **Schema definition:** The DBA creates the original database schema by executing a set of data definition statements in the DDL.

2. **Storage structure and access method modification**

The DBA is responsible for selecting the storage device, defining storage structure and access methods of database.

3. **Schema and physical-organization modification**: The DBA carries out changes to the schema and physical organization of database to reflect the changing needs of the organization, or to alter the physical organization to improve performance.

4. **Granting the authorization for data access:** By granting different types of authorization, the database administrator can regulate which parts of the database system various users can access. The authorization information is kept in special system structure that the database system consults whenever someone attempts to access the data in the system.

5. **Routine maintenance**

Examples of the database administrator's routine maintenance activities are:

- Periodically backing up the database, either onto tapes or remote servers, to percent loss of data in case of disasters such as flooding.
- Ensuring that enough free disk space is available for normal operations and upgrading disk space as required.
- Monitoring jobs running on the database and ensuring that performance is not degraded by very expensive tasks submitted by some users.

Database users

- There are four different types of database system differentiated by the way they expect to interact with the system. Difference types of user interfaces have been designed for different types of users in the system.

1. **Naive users:**

- Naive users are the simple users who just uses the application that have been built previously.
- For example, a customer who uses ATM simply invokes the program which checks his identity and balance and finally allows to withdraw certain amount from his account.

2. **Application programmer:**

- These are the computer professionals who write application programs.
- They build user interfaces to interact with the database and hence it makes naïve users easy.
- Application programmers can choose from may tools to develop user interfaces.
- Rapid application development (RAD) tools are tools that enable an application programmer to construct forms and reports with minimal programming effort.

3. **Sophisticated users:**

- They interact with the system without writing programs.
- Instead, they form requests either using a database query language or by using tools such as data analysis software.
- Analysts who submit queries to explore data in database fall in this category.

4. **Specialized users:**

- Specialized users are sophisticated users.
- They write specialized database applications that do not fit in traditional processing framework.
- Among these applications are computer aided design systems, knowledge based and expert systems, system that store data with complex data types (for example, graphics data and audio data)
- Scientists and researchers fall into this category.