MODULE: 5 (Database)

• What do you understand By Database

Ans. A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a [database management system (DBMS)](https://www.oracle.com/database/what-is-database/#WhatIsDBMS).

• What is Normalization?

Ans. Normalization is the process of organizing a database to reduce redundancy and improve [data integrity](https://database.guide/what-is-data-integrity/).

• What is Difference between DBMS and RDBMS?

Ans. **Database Management System (DBMS)** is a software that is used to define, create and maintain a database and provides controlled access to the data.

**Relational Database Management System (RDBMS)** is an advanced version of a DBMS.

DBMS:

1.[DBMS](https://www.geeksforgeeks.org/introduction-of-dbms-database-management-system-set-1/) stores data as file.

2. Data elements need to access individually.

3. No relationship between data.

4. Normalization is not present.

5. DBMS does not support distributed database.

RDBMS:

1. [RDBMS](https://www.geeksforgeeks.org/rdbms-architecture/) stores data in tabular form.

2. Multiple data elements can be accessed at the same time.

3. Data is stored in the form of tables which are related to each other.

4. Normalization is present.

5. RDBMS supports distributed database.

• What is MF Cod Rule of RDBMS Systems?

Ans. Database Management System or DBMS essentially consists of a comprehensive set of application programs that can be leveraged to access, manage and update the data, provided the data is interrelated and profoundly persistent.

* Data Modeling − It is all about defining the structures for information storage.
* Provision of Mechanisms − To manipulate processed data and modify file and system structures, it is important to provide query processing mechanisms.
* Crash Recovery and Security − To avoid any discrepancies and ensure that the data is secure, crash recovery and security mechanisms are must.
* Concurrency Control − If the system is shared by multiple users, concurrency control is the need of the hour.

• What do you understand By Data Redundancy?

Ans . Data redundancy occurs when the same piece of data is stored in two or more separate places and is a common occurrence in many businesses. As more companies are moving  away from siloed data to using a central repository to store information, they are finding that their database is filled with inconsistent duplicates of the same entry.

• What is DDL Interpreter?

Ans . It interprets DDL statements and record them in tables containing metadata.

Data Definition Language (DDL) is used to create and modify the structure of [objects](https://www.techtarget.com/searchapparchitecture/definition/object) in a [database](https://www.techtarget.com/searchdatamanagement/definition/database) using predefined [commands](https://www.techtarget.com/searchwindowsserver/definition/command) and a specific [syntax](https://www.techtarget.com/whatis/definition/syntax). These database objects include [tables](https://www.techtarget.com/whatis/definition/table), sequences, locations, [aliases](https://www.techtarget.com/whatis/definition/alias), [schemas](https://www.techtarget.com/searchdatamanagement/definition/schema) and indexes.

* [CREATE](https://www.w3schools.in/mysql/php-mysql-create/) - to create a database and its objects like (table, index, views, store procedure, function, and triggers)
* ALTER - alters the structure of the existing database
* DROP - delete objects from the database
* TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed
* COMMENT - add comments to the data dictionary
* RENAME - rename an object

• What is DML Compiler in SQL?

* Ans . DML Compiler: It processes the DML statements into low level instruction (machine language), so that they can be executed.

DML is short name of Data Manipulation Language which deals with data manipulation and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE, etc., and it is used to store, modify, retrieve, delete and update data in a database.

* + [SELECT](https://www.w3schools.in/mysql/php-mysql-select/) - retrieve data from a database
  + [INSERT](https://www.w3schools.in/mysql/php-mysql-insert/) - insert data into a table
  + [UPDATE](https://www.w3schools.in/mysql/php-mysql-update/) - updates existing data within a table
  + [DELETE](https://www.w3schools.in/mysql/php-mysql-delete/) - Delete all records from a database table
  + MERGE - UPSERT operation (insert or update)
  + CALL - call a PL/SQL or Java subprogram
  + EXPLAIN PLAN - interpretation of the data access path
  + LOCK TABLE - concurrency Control

• What is SQL Key Constraints writing an Example of SQL Key Constraints

Ans . Constraints can be specified when the table is created with the CREATE TABLE statement, or after the table is created with the ALTER TABLE statement.

SQL constraints are used to specify rules for the data in a table.

Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

The following constraints are commonly used in SQL:

* [NOT NULL](https://www.w3schools.com/sql/sql_notnull.asp) - Ensures that a column cannot have a NULL value
* [UNIQUE](https://www.w3schools.com/sql/sql_unique.asp) - Ensures that all values in a column are different
* [PRIMARY KEY](https://www.w3schools.com/sql/sql_primarykey.asp) - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
* [FOREIGN KEY](https://www.w3schools.com/sql/sql_foreignkey.asp) - Prevents actions that would destroy links between tables
* [CHECK](https://www.w3schools.com/sql/sql_check.asp) - Ensures that the values in a column satisfies a specific condition
* [DEFAULT](https://www.w3schools.com/sql/sql_default.asp) - Sets a default value for a column if no value is specified
* [CREATE INDEX](https://www.w3schools.com/sql/sql_create_index.asp) - Used to create and retrieve data from the database very quickly

• What is save Point? How to create a save Point write a Query?

Ans .

* Savepoint is a command in SQL that is used with the rollback command.
* It is a command in Transaction Control Language that is used to mark the transaction in a table.
* Consider you are making a very long table, and you want to roll back only to a certain position in a table then; this can be achieved using the savepoint.
* If you made a transaction in a table, you could mark the transaction as a certain name, and later on, if you want to roll back to that point, you can do it easily by using the transaction's name.
* Savepoint is helpful when we want to roll back only a small part of a table and not the whole table. In simple words, we can say savepoint is a bookmark in SQL.

• What is trigger and how to create a Trigger in SQL?

Ans .

Creates a DML, DDL, or logon trigger. A trigger is a special type of stored procedure that automatically runs when an event occurs in the database server. DML triggers run when a user tries to modify data through a data manipulation language (DML) event. DML events are INSERT, UPDATE, or DELETE statements on a table or view. These triggers fire when any valid event fires, whether table rows are affected or not. For more information, see [DML Triggers](https://learn.microsoft.com/en-us/sql/relational-databases/triggers/dml-triggers?view=sql-server-ver16).

DDL triggers run in response to a variety of data definition language (DDL) events. These events primarily correspond to Transact-SQL CREATE, ALTER, and DROP statements, and certain system stored procedures that perform DDL-like operations.

Task:

1 . Create Table Name : Student And Exam

Create a Database Assignment:

Query:

Create a Student: primary key

CREATE TABLE student (rid int NOT null AUTO\_INCREMENT PRIMARY KEY,

name varchar (50),branch varchar(5));

INSERT INTO student(rid,name,branch) VALUES(1,"Jainik","IT"),(2,"Jay","CS"),(3,"suhani","EC");

create a exam: foreign key

CREATE TABLE exam (eid int NOT null AUTO\_INCREMENT PRIMARY KEY, S\_code varchar (50),marks int (10), P\_code varchar(5), exid int NOT null, FOREIGN KEY(exid) REFERENCES student(rid));

INSERT INTO exam(eid,S\_code,marks,p\_code,exid) VALUES(1,"IT1",65,"IT",2),(2,"CS2",75,"CS",1),(3,"EC3",55,"EC",2);

Task-2:

Create table given below: Employee and IncentiveTable:

Create a Employee:

CREATE TABLE employee (eid int NOT null AUTO\_INCREMENT PRIMARY KEY, first\_name varchar (50),last\_name varchar(50),salary int (10),joining\_date varchar(50), department varchar (50));

INSERT INTO employee (eid,first\_name,last\_name,salary,Subcribed\_on,department)

VALUES(1,"John","Abraham",10000,"09-MAY-23","Banking"),

(2,"Michael","Clarke",80000,"10-MAY-23","Insurance"),

(3,"Roy","Thomas",700000,"11-MAY-23","Banking"),

(4,"Tom","Jose",600000,"12-MAY-23","Service");

Create a Incentive:

CREATE TABLE incentive(erid int NOT null AUTO\_INCREMENT PRIMARY KEY,incentive\_date varchar(50),incentive\_amount int NOT null);

INSERT INTO incentive(erid,incentive\_date,incentive\_amount)

VALUES (4,"01-FEB-13",5000),(2,"01-FEB-13",3000),(3,"01-FEB-13",5500),(1,"01-FEB-13",4000);

Ascending:

SELECT first\_name,last\_name,subcribed\_on,salary,Department FROM `employee` order by first\_name asc;

descending

SELECT first\_name,last\_name,subcribed\_on,salary,Department FROM `employee` order by first\_name DESC;

SalaryAscending:

SELECT first\_name,last\_name,subcribed\_on,salary,Department FROM `employee` order by salary ASC;

Trigger:

1. INSERT

CREATE TRIGGER AFTER ANOTHERTABLE ON employee AFTER INSERT AS SELECT \* from employee;

INSERT INTO employee VALUES (5,"Jainik","Rami",50000,"13-May-23","IT");

2. UPADTE

CREATE TRIGGER AFTER ANOTHERTABLE ON employee AFTER UPADTE AS SELECT \* from employee;

UPDATE employee SET first\_name="Jai-nik WHERE eid-5;

3. DELETE

CREATE TRIGGER AFTER ANOTHERTABLE ON employee AFTER DELETE AS SELECT \* from employee WHERE eid-5;

DELETE FROM employee WHERE eid-5;