|  |  |
| --- | --- |
|  | MODULE: 5 (Database)  **Topics Covered Basics of Database** |
| **1. What do you understand By Database** | **🡪A database is an organized collection of data, stored and accessed electronically. Databases are used to store and manage large amounts of**[**structured and unstructured data**](https://www.simplilearn.com/structured-vs-unstructured-data-article)**, and they can be used to support a wide range of activities, including data storage, data analysis, and data management.** |
| **2. What is Normalization?** | **🡪Normalization is a methodological method used in the design of databases to create a neat, structured, and structured table in which each table relates to just one subject or one-to-one correspondence. The objective is to extensively reduce data redundancy and dependency.** |
| **3. What is Difference between DBMS and RDBMS?** | **🡪 RDBMS: - Relation database management system.**  **Data Stored is in table format.**  **Multiple data element is accessible together.**  **Data in the form of a table are linked together.**  **Support distributed database.**  **Data is Stored in large amount.**  **RDBMS supports multiple users.**  **The software and hardware requirement are higher.**  **Example: - Oracle, SQL, Server.**  **🡪DBMS: - Data stored is in file format**  **Individual access of data element**  **No connection between data**  **No support for distributed database**  **Data stored is a small quantity**  **DBMS support a single user**  **The software and hardware requirements are low**  **Example: - XML, Microsoft Assess.** |
| **4. What is MF Cod Rule of RDBMS Systems?** | **🡪**[**The MF Cod Rule of RDBMS Systems states that for a system to qualify as an RDBMS, it must be able to manage database entirely through the relational capabilities**](https://www.bing.com/ck/a?!&&p=7681d4cee16c9fa1JmltdHM9MTcxNzk3NzYwMCZpZ3VpZD0zOGNlYzg3OS04MDMyLTY3ODUtM2RjNy1kYjFlODFmMTY2YjcmaW5zaWQ9NTc0OQ&ptn=3&ver=2&hsh=3&fclid=38cec879-8032-6785-3dc7-db1e81f166b7&psq=4.+What+is+MF+Cod+Rule+of+RDBMS+Systems%3f++&u=a1aHR0cHM6Ly93d3cuc3R1ZHl0b25pZ2h0LmNvbS9kYm1zL2NvZGQtcnVsZS5waHA&ntb=1) **.**[**Rule 0 of the MF Cod Rules states that the system must qualify as relational, as a database, and as a management system. For a system to qualify as an RDBMS, that system must use its relational facilities exclusively to manage the database**](https://www.bing.com/ck/a?!&&p=82c96f9cd357cdbaJmltdHM9MTcxNzk3NzYwMCZpZ3VpZD0zOGNlYzg3OS04MDMyLTY3ODUtM2RjNy1kYjFlODFmMTY2YjcmaW5zaWQ9NTc1Mg&ptn=3&ver=2&hsh=3&fclid=38cec879-8032-6785-3dc7-db1e81f166b7&psq=4.+What+is+MF+Cod+Rule+of+RDBMS+Systems%3f++&u=a1aHR0cDovL3d3dy5ta2ljcy5pbi9Ob3Rlcy9TWUJDQS9TRU0zL1JEQk1TL0NPREVTLUxBVy1SREJNUy5wZGY&ntb=1)**.** |
| **5. What do you understand By Data Redundancy?** | **🡪**[**Data redundancy refers to the situation where the same pieces of data are stored in multiple places within a database or data storage system**](https://www.bing.com/ck/a?!&&p=55e0be9278185267JmltdHM9MTcxNzk3NzYwMCZpZ3VpZD0zOGNlYzg3OS04MDMyLTY3ODUtM2RjNy1kYjFlODFmMTY2YjcmaW5zaWQ9NTc3MQ&ptn=3&ver=2&hsh=3&fclid=38cec879-8032-6785-3dc7-db1e81f166b7&psq=5.+What+do+you+understand+By+Data+Redundancy%3f++&u=a1aHR0cHM6Ly93d3cubWluaXRvb2wuY29tL2xpYi9kYXRhLXJlZHVuZGFuY3kuaHRtbA&ntb=1)**.**[**This can happen intentionally or accidentally. Redundancy can be useful for data recovery in case of corruption or loss**](https://www.bing.com/ck/a?!&&p=1713dccd36b64a4bJmltdHM9MTcxNzk3NzYwMCZpZ3VpZD0zOGNlYzg3OS04MDMyLTY3ODUtM2RjNy1kYjFlODFmMTY2YjcmaW5zaWQ9NTc3NQ&ptn=3&ver=2&hsh=3&fclid=38cec879-8032-6785-3dc7-db1e81f166b7&psq=5.+What+do+you+understand+By+Data+Redundancy%3f++&u=a1aHR0cHM6Ly9kYXRhY29ub215LmNvbS8yMDIyLzAzLzA5L3doYXQtaXMtZGF0YS1yZWR1bmRhbmN5Lw&ntb=1)**.**[**In computer memory and storage, data redundancy allows for error correction**](https://www.bing.com/ck/a?!&&p=17aadad700ccb733JmltdHM9MTcxNzk3NzYwMCZpZ3VpZD0zOGNlYzg3OS04MDMyLTY3ODUtM2RjNy1kYjFlODFmMTY2YjcmaW5zaWQ9NTc3Nw&ptn=3&ver=2&hsh=3&fclid=38cec879-8032-6785-3dc7-db1e81f166b7&psq=5.+What+do+you+understand+By+Data+Redundancy%3f++&u=a1aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvRGF0YV9yZWR1bmRhbmN5&ntb=1) |
| **6. What is DDL Interpreter?** | **🡪DML Compiler: It processes the DML statements into low level instruction (machine language), so that they can be executed. DDL Interpreter: It processes the DDL statements into a set of tables containing meta data (data about data).** |
| **7. What is DML Compiler in SQL?** | **🡪The Data Manipulation Language, or DML for short, is the group of commands responsible for manipulating data in a database; this generally entails inserting, editing, or deleting rows in SQL tables.** |
| **8. What is SQL Key Constraints writing an Example of SQL Key Constraints** | **🡪Constraints are the rules that we can apply on the type of data in a table. That is, we can specify the limit on the type of data that can be stored in a particular column in a table using constraints.**  **The available constraints in SQL are:**   * **NOT NULL: This constraint tells that we cannot store a null value in a column. That is, if a column is specified as NOT NULL then we will not be able to store null in this particular column any more.** * **UNIQUE: This constraint when specified with a column, tells that all the values in the column must be unique. That is, the values in any row of a column must not be repeated.** * **PRIMARY KEY: A primary key is a field which can uniquely identify each row in a table. And this constraint is used to specify a field in a table as primary key.** * **FOREIGN KEY: A Foreign key is a field which can uniquely identify each row in another table. And this constraint is used to specify a field as foreign key.** * **CHECK: This constraint helps to validate the values of a column to meet a particular condition. That is, it helps to ensure that the value stored in a column meets a specific condition.** * **DEFAULT: This constraint specifies a default value for the column when no value is specified by the user.** |
| **9. What is save Point? How to create a save Point write a Query?** | **🡪**[**A save point in SQL is a logical rollback point within a transaction**](https://www.bing.com/ck/a?!&&p=a1f26c8649e17103JmltdHM9MTcxNzk3NzYwMCZpZ3VpZD0zOGNlYzg3OS04MDMyLTY3ODUtM2RjNy1kYjFlODFmMTY2YjcmaW5zaWQ9NTgxMQ&ptn=3&ver=2&hsh=3&fclid=38cec879-8032-6785-3dc7-db1e81f166b7&psq=9.+What+is+save+Point%3f+How+to+create+a+save+Point+write+a++Query%3f&u=a1aHR0cHM6Ly9yZWludGVjaC5pby9ibG9nL2EtZGV0YWlsZWQtZ3VpZGUtb24tc3FsLXNhdmVwb2ludC1zdGF0ZW1lbnQ&ntb=1)**.**[**It allows you to specify a point in a transaction that you can roll back to without affecting the entire transaction. To create a, save point, use the following syntax: `SAVEPOINT savepoint\_name`**](https://www.bing.com/ck/a?!&&p=f3db28d8b83b0221JmltdHM9MTcxNzk3NzYwMCZpZ3VpZD0zOGNlYzg3OS04MDMyLTY3ODUtM2RjNy1kYjFlODFmMTY2YjcmaW5zaWQ9NTgxNA&ptn=3&ver=2&hsh=3&fclid=38cec879-8032-6785-3dc7-db1e81f166b7&psq=9.+What+is+save+Point%3f+How+to+create+a+save+Point+write+a++Query%3f&u=a1aHR0cHM6Ly93d3cuc3FsdHV0b3JpYWwubmV0L3NhdmVwb2ludC5odG1s&ntb=1)**.**[**You can then perform various SQL operations within the transaction. To roll back to a specific save point, use `ROLLBACK TO save\_point\_name`**](https://www.bing.com/ck/a?!&&p=2337a085fcb82b82JmltdHM9MTcxNzk3NzYwMCZpZ3VpZD0zOGNlYzg3OS04MDMyLTY3ODUtM2RjNy1kYjFlODFmMTY2YjcmaW5zaWQ9NTgxNg&ptn=3&ver=2&hsh=3&fclid=38cec879-8032-6785-3dc7-db1e81f166b7&psq=9.+What+is+save+Point%3f+How+to+create+a+save+Point+write+a++Query%3f&u=a1aHR0cHM6Ly9jb2RlYnVyc3QuaW8vc2F2ZXBvaW50LWluLW15c3FsLXdpdGgtZXhhbXBsZXMtMTk3NjkyODk5Mjdh&ntb=1) |
| **10.What is trigger and how to create a Trigger in SQL?** | **🡪Trigger is a statement that a system executes automatically when there is any modification to the database. In a trigger, we first specify when the trigger is to be executed and then the action to be performed when the trigger executes. Triggers are used to specify certain integrity constraints and referential constraints that cannot be specified using the constraint mechanism of SQL.**  **Example –  Suppose, we are adding a tuple to the ‘Donors’ table that is some person has donated blood. So, we can design a trigger that will automatically add the value of donated blood to the ‘Blood\_record’ table.**  **Types of Triggers –  We can define 6 types of triggers for each table:**   1. **AFTER INSERT activated after data is inserted into the table.** 2. **AFTER UPDATE: activated after data in the table is modified.** 3. **AFTER DELETE: activated after data is deleted/removed from the table.** 4. **BEFORE INSERT: activated before data is inserted into the table.** 5. **BEFORE UPDATE: activated before data in the table is modified.** 6. **BEFORE DELETE: activated before data is deleted/removed from the table.** |

**SQL Queries**

|  |  |
| --- | --- |
| Create Table Name: Student |  |
| 1 | CREAT TABLE Student  (  Rollno int PRIMARY KEY,  Name Varchar(30),  Branch Text  ); |
| 3 | INSERT INTO student VALUES (1,'Jay','Computer Science');  INSERT INTO student VALUES (2,'Suhani','Electronic and Com');  INSERT INTO student VALUES (2,'Kriti','Electronic and Com'); |
| 4 Create Table Name: Exam |  |
| 5 | CREATE TABLE Exam  (  Rollno int,  S\_code text,  Marks int,  P\_code text,  FOREIGN KEY(Rollno) REFERENCES student(Rollno)  ); |
| 6 | INSERT INTO exam VALUES(1,'CS11',50,'CS');  INSERT INTO exam VALUES(1,'CS12',60,'CS');  INSERT INTO exam VALUES(2,'EC101',66,'EC');  INSERT INTO exam VALUES(2,'EC102',70,'EC');  INSERT INTO exam VALUES(3,'EC101',45,'EC');  INSERT INTO exam VALUES(3,'EC102',50,'EC'); |

|  |  |  |  |
| --- | --- | --- | --- |
| 2.Create table given below: Employee |  |  |  |
| 1 | CREATE TABLE Employee  (  Employee\_id int,  First\_name varchar(50),  Last\_name varchar(50),  Salary int,  Joining\_date datetime,  Department text  ); |  |  |
| 3 | INSERT INTO employee VALUES(1,'John','Abraham',1000000,'13-01-01 12.00.00 AM','Banking');  INSERT INTO employee VALUES(2,'Michael','Clarke',800000,'13-01-01 12.00.00 AM','Insurance');  INSERT INTO employee VALUES(3,'Roy','Thomas',700000,'13-02-01 12.00.00 AM','Banking');  INSERT INTO employee VALUES(4,'Tom','Jose',600000,'13-02-01 12.00.00 AM','Insurance');  INSERT INTO employee VALUES(5,'Jerry','Pinto',650000,'13-02-01 12.00.00 AM','Insurance');  INSERT INTO employee VALUES(6,'Philip','Mathew',750000,'13-01-01 12.00.00 AM','Service');  INSERT INTO employee VALUES(7,'TestName1','123',650000,'13-01-01 12.00.00 AM','Service');  INSERT INTO employee VALUES(8,'TestName2','Lname%',600000,'13-02-01 12.00.00 AM','Insurance'); |  |  |
| Creat table give below: Incentive | CREATE TABLE Incentive  (  Employee\_ref\_id int,  Incentive\_date date,  Incentive\_amount int  ); |  |  |
| 2 | INSERT INTO incentive VALUES(1,'13-02-01',5000);  INSERT INTO incentive VALUES(2,'13-02-01',3000);  INSERT INTO incentive VALUES(3,'13-02-01',4000);  INSERT INTO incentive VALUES(1,'13-01-01',4500);  INSERT INTO incentive VALUES(2,'13-01-01',3500); |  |  |
| 3. Get First\_Name from employee table using Tom name “Employee Name”. | 🡪SELECT \* FROM employee WHERE First\_name='Tom'; |  |  |
| 4. Get FIRST\_NAME, Joining Date, and Salary from employee table. | 🡪SELECT First\_name,Joining\_date,Salary FROM employee; |  |  |
| 5. Get all employee details from the employee table order by First\_Name Ascending and Salary descending? | 🡪SELECT \* FROM employee ORDER BY First\_name ASC,Salary DESC; |  |  |
| 6. Get employee details from employee table whose first name contains ‘J’. | 🡪SELECT \* FROM employee WHERE First\_name LIKE'j%'; |  |  |
| 7. Get department wise maximum salary from employee table order by | 🡪SELECT MAX(Salary) AS Salary FROM employee; |  |  |
| 8. salaryascending? | 🡪SELECT \* FROM employee ORDER BY Salary ASC; |  |  |
| 9. Select first\_name, incentive amount from employee and incentives table forthose employees who have incentives and incentive amount greater than 3000 | Output: |  |  |
| 10. Create After Insert trigger on Employee table which insert records in viewtable | 🡪 |  |  |

|  |  |
| --- | --- |
| 11. Create table given below: Salesperson |  |
| 1 | CREATE TABLE Salesperson  (  PK\_SNo int,  SNAME varchar(30),  City varchar(30),  Comm text  ); |
| 2 | INSERT INTO salesperson VALUES(1001,'Peel','London',.12);  INSERT INTO salesperson VALUES(1002,'Serres','San Jose',.13);  INSERT INTO salesperson VALUES(1004,'Motika','London',.11);  INSERT INTO salesperson VALUES(1007,'Rafkin','Barcelona',.15);  INSERT INTO salesperson VALUES(1003,'Axelrod','New York',.1); |
| 11. Create table given below: Customer |  |
| 3 | CREATE TABLE Customer  (  PK\_CNM int,  CNAME varchar(30),  City varchar(30),  Rating int,  FK\_SNo int  ); |
| 4 | INSERT INTO customer VALUES(201,'Hoffman','London',100,1001);  INSERT INTO customer VALUES(202,'Giovanne','Roe',200,1003);  INSERT INTO customer VALUES(203,'Liu','San Jose',300,1002);  INSERT INTO customer VALUES(204,'Grass','Barcelona',100,1002);  INSERT INTO customer VALUES(206,'Clemens','London',300,1007);  INSERT INTO customer VALUES(207,'Pereira','Roe',100,1004); |
| Retrieve the below data from above table | All query is performed below. |
| 13.All orders for more than $1000.  Output: |  |
| 14.Names and cities of all salespeople in London with commission above 0.12  Output: |  |
| 15.All salespeople either in Barcelona or in London  Output: | 🡪 |
| 16. All salespeople with commission between 0.10 and 0.12. (Boundary valuesshould be excluded). | Output: |
| 17. All customers excluding those with rating <= 100 unless they are located inRome | Output: |

|  |  |
| --- | --- |
| 18. Write a SQL statement that displays all the information about all salespeople |  |
| 1 | CREATE TABLE salespeople  (  salesman\_id int,  name varchar(30),  city text,  commission text  ); |
| 2 | INSERT INTO salespeople VALUES(5001,'James Hoog','New York',0.15);  INSERT INTO salespeople VALUES(5002,'Nail Knite','paris',0.13);  INSERT INTO salespeople VALUES(5005,'Pit Alex','London',0.11);  INSERT INTO salespeople VALUES(5006,'Mc Lyon','paris',0.14);  INSERT INTO salespeople VALUES(5007,'Paul Adam','Rome',0.13);  INSERT INTO salespeople VALUES(5003,'Lauson Hen','San Jose',0.12); |

|  |  |
| --- | --- |
| 19. From the following table, write a SQL query to find orders that are delivered by a salesperson with ID. 5001. Return ord\_no, ord\_date, purch\_amt. |  |
| 1 | CREATE TABLE orders  (  ord\_no int,  purch\_amt text,  ord\_date date,  customer\_id int,  salesman\_id int  ); |
| 2 | INSERT INTO orders VALUES(70001,150.5,'2012-10-05',3005,5002);  INSERT INTO orders VALUES(70009,270.65,'2012-09-10',3001,5005);  INSERT INTO orders VALUES(70002,65.26,'2012-10-05',3002,5001);  INSERT INTO orders VALUES(70004,110.5,'2012-08-17',3009,5003);  INSERT INTO orders VALUES(70007,948.5,'2012-09-10',3005,5002);  INSERT INTO orders VALUES(70005,2400.6,'2012-07-27',3007,5001);  INSERT INTO orders VALUES(70008,5760,'2012-09-10',3002,5001);  INSERT INTO orders VALUES(70010,1983.43,'2012-10-10',3004,5006);  INSERT INTO orders VALUES(70003,2480.4,'2012-10-10',3009,5003);  INSERT INTO orders VALUES(70012,250.45,'2012-06-27',3008,5002);  INSERT INTO orders VALUES(70011,75.29,'2012-08-17',3003,5007);  INSERT INTO orders VALUES(70013,3045.6,'2012-04-25',3002,5001); |
| Query: |  |
| Output: |  |

|  |  |
| --- | --- |
| 20. From the following table, write a SQL query to select a range of products whose price is in the range Rs.200 to Rs.600. Begin and end values are included. Return pro\_id, pro\_name, pro\_price, and pro\_com. |  |
|  | CREATE TABLE item\_mast  (  pro\_id int,  pro\_name varchar(30),  pro\_price text,  pro\_com int  ); |
|  | INSERT INTO item\_mast VALUES(101,'Mother Board',3200.00,15);  INSERT INTO item\_mast VALUES(102,'Key Board',450.00,16);  INSERT INTO item\_mast VALUES(103,'ZIP Drive',250.00,14);  INSERT INTO item\_mast VALUES(104,'Speaker',550.00,16);  INSERT INTO item\_mast VALUES(105,'Monitor',5000.00,11);  INSERT INTO item\_mast VALUES(106,'DVD drive',900.00,12);  INSERT INTO item\_mast VALUES(107,'CD drive',800.00,12);  INSERT INTO item\_mast VALUES(108,'Printer',2600.00,13);  INSERT INTO item\_mast VALUES(109,'Refill catridge',350.00,13);  INSERT INTO item\_mast VALUES(110,'Mouse',250.00,12); |
| Query: |  |
| Output: |  |

|  |  |
| --- | --- |
| 21. From the following table, write a SQL query to calculate the average price for a manufacturer code of 16. Return avg. |  |
|  | CREATE TABLE item\_mast  (  pro\_id int,  pro\_name varchar(30),  pro\_price text,  pro\_com int  ); |
|  | INSERT INTO item\_mast VALUES(101,'Mother Board',3200.00,15);  INSERT INTO item\_mast VALUES(102,'Key Board',450.00,16);  INSERT INTO item\_mast VALUES(103,'ZIP Drive',250.00,14);  INSERT INTO item\_mast VALUES(104,'Speaker',550.00,16);  INSERT INTO item\_mast VALUES(105,'Monitor',5000.00,11);  INSERT INTO item\_mast VALUES(106,'DVD drive',900.00,12);  INSERT INTO item\_mast VALUES(107,'CD drive',800.00,12);  INSERT INTO item\_mast VALUES(108,'Printer',2600.00,13);  INSERT INTO item\_mast VALUES(109,'Refill catridge',350.00,13);  INSERT INTO item\_mast VALUES(110,'Mouse',250.00,12); |
| Query: |  |
| Output: |  |

|  |  |
| --- | --- |
| 22. From the following table, write a SQL query to display the pro\_name as 'Item Name' and pro\_priceas 'Price in Rs.' |  |
|  | CREATE TABLE item\_mast  (  pro\_id int,  pro\_name varchar(30),  pro\_price text,  pro\_com int  ); |
|  | INSERT INTO item\_mast VALUES(101,'Mother Board',3200.00,15);  INSERT INTO item\_mast VALUES(102,'Key Board',450.00,16);  INSERT INTO item\_mast VALUES(103,'ZIP Drive',250.00,14);  INSERT INTO item\_mast VALUES(104,'Speaker',550.00,16);  INSERT INTO item\_mast VALUES(105,'Monitor',5000.00,11);  INSERT INTO item\_mast VALUES(106,'DVD drive',900.00,12);  INSERT INTO item\_mast VALUES(107,'CD drive',800.00,12);  INSERT INTO item\_mast VALUES(108,'Printer',2600.00,13);  INSERT INTO item\_mast VALUES(109,'Refill catridge',350.00,13);  INSERT INTO item\_mast VALUES(110,'Mouse',250.00,12); |
| Query: |  |
| Output: |  |

|  |  |
| --- | --- |
| 23. From the following table, write a SQL query to find the items whose prices are higher than or equal to $250. Order the result by product price in descending, then product name in ascending. Return pro\_name and pro\_price. |  |
|  | CREATE TABLE item\_mast  (  pro\_id int,  pro\_name varchar(30),  pro\_price text,  pro\_com int  ); |
|  | INSERT INTO item\_mast VALUES(101,'Mother Board',3200.00,15);  INSERT INTO item\_mast VALUES(102,'Key Board',450.00,16);  INSERT INTO item\_mast VALUES(103,'ZIP Drive',250.00,14);  INSERT INTO item\_mast VALUES(104,'Speaker',550.00,16);  INSERT INTO item\_mast VALUES(105,'Monitor',5000.00,11);  INSERT INTO item\_mast VALUES(106,'DVD drive',900.00,12);  INSERT INTO item\_mast VALUES(107,'CD drive',800.00,12);  INSERT INTO item\_mast VALUES(108,'Printer',2600.00,13);  INSERT INTO item\_mast VALUES(109,'Refill catridge',350.00,13);  INSERT INTO item\_mast VALUES(110,'Mouse',250.00,12); |
| Query: |  |
| Output: |  |

|  |  |
| --- | --- |
| 24. From the following table, write a SQL query to calculate average price of the items for each company. Return average price and company code. |  |
|  | CREATE TABLE item\_mast  (  pro\_id int,  pro\_name varchar(30),  pro\_price text,  pro\_com int  ); |
|  | INSERT INTO item\_mast VALUES(101,'Mother Board',3200.00,15);  INSERT INTO item\_mast VALUES(102,'Key Board',450.00,16);  INSERT INTO item\_mast VALUES(103,'ZIP Drive',250.00,14);  INSERT INTO item\_mast VALUES(104,'Speaker',550.00,16);  INSERT INTO item\_mast VALUES(105,'Monitor',5000.00,11);  INSERT INTO item\_mast VALUES(106,'DVD drive',900.00,12);  INSERT INTO item\_mast VALUES(107,'CD drive',800.00,12);  INSERT INTO item\_mast VALUES(108,'Printer',2600.00,13);  INSERT INTO item\_mast VALUES(109,'Refill catridge',350.00,13);  INSERT INTO item\_mast VALUES(110,'Mouse',250.00,12); |
| Query: |  |
| Output: |  |