MSSQL --> Microsoft SQL server 2

SSMS---> SQL Server Management Studio:

SQL---> structure query language which help us to interact with our database.

Database---> database is a kind of container which provide us space which we can use to store our data.

COMMAND CATEGORY:

a)-DDL COMMAND (Data Definitions Language)

b)-DQL COMMAND (Data Query Language)

c)-DML COMMAND (Data Manipulation Language)

d)-DCL COMMAND (Data Control Language)

a)-DDL COMMAND (Data definitions Language)

* CREATE
* ALTER
* DROP

1.CREATE : 🡪

CREATE DATABASE (DATABASE\_NAME)

CREATE TABLE (TABLE\_NAME)

Question:🡪

1. How we can create our database?

CREATE DATABASE SQL\_Class

1. How we can create Tables?

CREATE TABLE EmployeeDetails\_18thMay (EmployeeID INT, EmployeeName VARCHAR (50), Salary INTEGER)

1. ALTER :🡪

ALTER TABLE (TABLE NAME) ADD (TABLE NAME)

Question:🡪

1. Write a query to add city column in EmployeeDetails\_18thMay?

ALTER TABLE EmployeeDetails\_18thMay ADD City VARCHAR (50)

ALTER TABLE EmployeeDetails\_18thMay ADD Country VARCHAR (50),   
State VARCHAR (50)

1. DROP ---> this command we can use to remove any object from our database or database itself.

DROP DATABASE DATABASE\_NAME.

DROP TABLE TABLE\_NAME.

Question:🡪

How we can drop a database?

DROP DATABASE SQL\_Claass

Write a query to remove EmployeeName column from EmployeeDetails\_18thMay?

ALTER TABLE EmployeeDetails\_18thMay DROP COLUMN EmployeeName

b)-DQL COMMAND (Data Query Language)

a)-SELECT---> This command we can use to display the table structure along with data

SELECT \* FROM (TABEL\_NAME)

SELECT \* FROM EmployeeDetails\_18thMay

SELECT EmployeeID FROM EmployeeDetails\_18thMay

SELECT EmployeeID,EmployeeName FROM EmployeeDetails\_18thMay

c)-DML COMMAND (Data Manipulation Language)

a)-INSERT

b)-UPDATE

c)-DELETE

a)-INSERT

SELECT \* FROM (TABLENAME)

INSERT INTO TABLE NAME (TABLE NAME1, TABLE NAME2, TABLE NAME3)

VALUES (1,2,3)

SELECT \* FROM EmployeeDetails\_18thMay

INSERT INTO EmployeeDetails\_18thMay (Employee ID, Employee Name, Salary)

VALUES (1,'Sahil Jaiswal',70000)

If we don’t want to add values in specific column

INSERT INTO EmployeeDetails\_18thMay (Employee, Salary)

VALUES (2,10000)

Multiple values in single line to add “, (Commma)” for separator.

INSERT INTO EmployeeDetails\_18thMay (Employee ID, Employee Name, Salary)

VALUES(3,'Miraj',90000), (4,'Meenakshi',89000), (5,'Vishal',67000)

UPDATE---> this command we are going to use update the existing data.

SELECT \* FROM EmployeeDetails\_18thMay

UPDATE EmployeeDetails\_18thMay SET Salary=95000

Write a query to update salary=65000 for employeeID 3?

UPDATE EmployeeDetails\_18thMay SET Salary=65000 WHERE EmployeeID=3

UPDATE EmployeeDetails\_18thMay SET EmployeeName='abdul',Salary=70000 WHERE EmployeeID=1

DELETE-----> this command we can use to remove the data from our table.

SELECT \* FROM EmployeeDetails\_18thMay

DELETE FROM EmployeeDetails\_18thMay

Write a query to delete a data for employeeId 5?

SELECT \* FROM EmployeeDetails\_18thMay

DELETE FROM EmployeeDetails\_18thMay WHERE EmployeeID=5

d)-DCL COMMAND (Data Control Language)

a)-GRANT

b)-REVOKE

a)-GRANT

GRANT SELECT ON EmployeeDetails\_18thMay TO USER1

b)-REVOKE

REVOKE SELECT, INSERT, DELETE ON EmployeeDetails\_18thMay FROM User1

**DATATYPE**

1. NUMERIC
2. INTEGER
3. STRING
4. DATE

1)-NUMERIC

a)-TINYINT---------------->0 to 255

b)-SMALLINT---------------> -32768 to 32767

c)-INTEGER/INT

d)-BIGINT

e)-DECIMAL

CREATE TABLE TinyInt\_Test\_18thMay (ID TINYINT)

INSERT INTO TinyInt\_Test\_18thMay (ID) VALUES (0)

INSERT INTO TinyInt\_Test\_18thMay (ID)VALUES (-100) ---Error

INSERT INTO TinyInt\_Test\_18thMay (ID) VALUES (500) ----Error

INSERT INTO TinyInt\_Test\_18thMay (ID) VALUES (255)

INSERT INTO TinyInt\_Test\_18thMay (ID) VALUES (256)

CREATE TABLE SmallInt\_Test\_18thMay (ID SMALLINT)

INSERT INTO SmallInt\_Test\_18thMay (ID) VALUES (-40000) ---- Error

INSERT INTO SmallInt\_Test\_18thMay (ID) VALUES (40000) ---Error

INSERT INTO SmallInt\_Test\_18thMay (ID) VALUES (32767)

INSERT INTO SmallInt\_Test\_18thMay (ID) VALUES (32768) –Error

10.12

100.2375

DECIMAL(p,s)

p-->Precision---> total number of digit

s-->Scale--->Number of digit that can be stored after decimal

DECIMAL(4,2)

10.12

total no of digit--->4

After Decimal Place-->2

How many digit it will accept before decimal place-->4-2=2

CREATE TABLE Decimal Test(Price Decimal (5,2))

Total Digit-->5

After Decimal-->2

Before Decimal--->5-2=3

INSERT INTO Decimal Test (Price) VALUES (100.12)

INSERT INTO Decimal Test (Price) VALUES (1001.1) --->Error

INSERT INTO Decimal Test (Price) VALUES (1.1)

INSERT INTO Decimal Test (Price) VALUES (1.137)

INSERT INTO Decimal Test (Price) VALUES (1124.134)

SELECT \* FROM Decimal Test

STRING:

a)-CHAR

b)-VARCHAR

a)-CHAR

CREATE TABLE LearnerInfo (CandidateName CHAR(10))

INSERT INTO LearnerInfo(CandidateName)VALUES('Sahil')

INSERT INTO LearnerInfo(CandidateName)VALUES('Raj')

INSERT INTO LearnerInfo(CandidateName)VALUES('SahilJAISW')

INSERT INTO LearnerInfo(CandidateName)VALUES('SahilJAISWSDS')

SELECT \* FROM LearnerInfo

'Sahil '-->10-5=5

'Raj '-->10-3=7

'SahilJAISW'

--MAXIMUM WE CAN DEFINE 8000

b)-VARCHAR

CREATE TABLE Employee\_Data\_24thMay(EmployeeName VARCHAR(10))

INSERT INTO Employee\_Data\_24thMay(EmployeeName)VALUES('Sahil')

INSERT INTO Employee\_Data\_24thMay(EmployeeName)VALUES('Raj')

SELECT \* FROM Employee\_Data\_24thMay 75

'Sahil'

'Raj'

CREATE TABLE Employee\_Data\_24thMay(EmployeeName VARCHAR(10))

CREATE TABLE Employee\_Data\_24thMay(EmployeeName VARCHAR(MAX)---> this can accept upto 2 gb character)

a)-DATE

b)-TIME

c)-DATETIME

'yyyy-mm-dd hh:min:sec.ms'

CREATE TABLE SaleData\_24thMay(OrderDate DATE)

INSERT INTO SaleData\_24thMay(OrderDate)VALUES('2025-05-01')

INSERT INTO SaleData\_24thMay(OrderDate)VALUES('2025-05-02 09:10:40.123')

SELECT \* FROM SaleData\_24thMay

--2025-05-24 15:06:48.173

CREATE TABLE SalesTime\_24thMay(OrderTime TIME)

INSERT INTO SalesTime\_24thMay(OrderTime)VALUES('2025-05-02 09:10:40.1234')

INSERT INTO SalesTime\_24thMay(OrderTime)VALUES('12:11:51')

SELECT \* FROM SalesTime\_24thMay

Write a query to create a studentInfo table to store Student information, below is the data point

a)-StudentID

b)-StudentName

c)-Age

d)-DOB

and also insert 5 record in this table

CREATE TABLE StudentInfo(StudentID TINYINT,StudentName VARCHAR(50),Age TINYINT,DOB DATE)

INSERT INTO StudentInfo(StudentID,StudentName,Age,DOB)

VALUES(1,'Sahil Jaiswal',30,'1999-01-01')

SELECT \* FROM StudentInfo

CREATE TABLE Sales\_OrderData ( OrderDateTime DATETIME)

INSERT INTO Sales\_OrderData(OrderDateTime)VALUES('2025-05-02 09:10:40.123')

INSERT INTO Sales\_OrderData(OrderDateTime)VALUES('2025-05-03') 149

INSERT INTO Sales\_OrderData(OrderDateTime)VALUES('25-05-03') 151

INSERT INTO Sales\_OrderData(OrderDateTime)VALUES('09:10:40.123')

SELECT \* FROM Sales\_OrderData

Insert into Studentinfo(StudentID,StudentName,Age,DOB)

Values(1,'Ishaan',33,'1991-09-10')

CONSTRAINT:

TYPE OF CONSTRAINT:

a)-NOT NULL---> if we define any column as not null then that column will not accept any null value.

b)-UNIQUE-----> if we apply unique constraint then that column will not allow to store any duplicate value.

c)-CHECK

d)-DEFAULT

e)-PRIMARY

f)-FOREIGN

a)-NOT NULL

CREATE TABLE ManageInfo (ManagerID INT NOT NULL, ManagerName VARCHAR(50))

INSERT INTO ManageInfo(ManagerID,ManagerName)VALUES(1,'Paul') 178

INSERT INTO ManageInfo(ManagerID,ManagerName)VALUES(NULL,'Namrata') 180

INSERT INTO ManageInfo(ManagerName)VALUES('Paul') 182

SELECT \* FROM ManageInfo

b)-UNIQUE

CREATE TABLE OrderSale\_24thMay (OrderID INT UNIQUE, OrderAmount INT)

INSERT INTO OrderSale\_24thMay(OrderID,OrderAmount)VALUES(1,100) 195

INSERT INTO OrderSale\_24thMay(OrderID,OrderAmount)VALUES(2,50)

INSERT INTO OrderSale\_24thMay(OrderID,OrderAmount)VALUES(1,510) 198

SELECT \* FROM OrderSale\_24thMay

c)-CHECK

CREATE TABLE VoterListInfo (VoterId INT, VoterName VARCHAR(50), VoterAge TINYINT CHECK(VoterAge>=18))

INSERT INTO VoterListInfo(VoterId,VoterName,VoterAge) VALUES(1,'Sahil Jaiswal',17)

INSERT INTO VoterListInfo(VoterId,VoterName,VoterAge) VALUES(1,'Sahil Jaiswal',25)

SELECT \* FROM VoterListInfo

d)-DEFAULT

DEFAULT---> if user is not passing any value for country column then we want sql should automatically added India as a value.

CREATE TABLE EmployeeData\_24thMay

(EmployeeID INT,EmployeeName VARCHAR(100),Country VARCHAR(100) DEFAULT('India'))

INSERT INTO EmployeeData\_24thMay(EmployeeID,EmployeeName,Country)

VALUES(1,'Sahil','India')

INSERT INTO EmployeeData\_24thMay(EmployeeID,EmployeeName,Country)

VALUES(2,'Anish','USA')

SELECT \* FROM EmployeeData\_24thMay

INSERT INTO EmployeeData\_24thMay(EmployeeID,EmployeeName)

VALUES(3,'Abdul')

INSERT INTO EmployeeData\_24thMay(EmployeeID,EmployeeName,Country)

VALUES(4,'Sonam',NULL)

SELECT \* FROM EmployeeData\_24thMay

CREATE TABLE NOT\_NULL\_CHECK (ID INT NOT NULL,FirstName VARCHAR(50) NOT NULL)

CREATE TABLE NOT\_NULL\_CHECK(ID INT NOT NULL UNIQUE,FirstName VARCHAR(50) NOT NULL)

PRIMARY KEY(UNIQUE + NOT NULL):

CREATE TABLE EmployeeDepartmentInfo (EmployeeID INT PRIMARY KEY,

EmployeeName VARCHAR(50), DepartmentID INT)

INSERT INTO EmployeeDepartmentInfo(EmployeeID, EmployeeName, DepartmentID)

VALUES(NULL,'Swapnil',1)

INSERT INTO EmployeeDepartmentInfo(EmployeeID, EmployeeName, DepartmentID)

VALUES(1,'Swapnil',1)

INSERT INTO EmployeeDepartmentInfo(EmployeeID,EmployeeName,DepartmentID)

VALUES(1,'Nimesh',1)

SELECT \* FROM EmployeeDepartmentInfo 290

Write a query to display employeeinfo for employeeid 2? 294

SELECT \* FROM EmployeeDetails WHERE EmployeeID=2 EmployeeID 2,3

Operator

IN-🡪when we have to compare more than one value for same column then we can use IN operator.

SELECT \* FROM EmployeeDetails WHERE EmployeeID IN (1,2)

NOT EQUAL(<>,!=)

Write a query to extract all employee info except employee ID 3?

SELECT \* FROM EmployeeDetails WHERE EmployeeID <>3

Write a query to display all employe except EmployeeID 1,5?

SELECT \* FROM EmployeeDetails WHERE EmployeeID NOT IN (1,5)

BETWEEN

10000 to 60000

SELECT \* FROM EmployeeDetails WHERE Salary BETWEEN 10000 AND 60000

--->=<=

SELECT \* FROM EmployeeDetails WHERE Salary>=10000 AND Salary<=600

LIKE :

SELECT \* FROM EmployeeDEtails WHERE FirstName='Vishal'

Write a query to display those employee whose firstname is start with 's'?

Sahil

Suman

Suraj

Syed

SELECT \* FROM EmployeeDetails WHERE FirstName LIKE 's%'

Write a query to display those employee whose firstname is end with 'l'?

SELECT \* FROM Employeedetails WHERE FirstName LIKE '%L'

Write a query to return those employee whose firstname is having 'a' character?

SELECT \* FROM EmployeeDetails WHERE FirstName LIKE '%a%'

Write a query to display those employees whose firstname is having 'a' character in second position?

\_-->

SELECT \* FROM EmployeeDetails WHERE FirstName LIKE '\_a%' 363

Write a query to displauy those employee whose firstname second last cjaracter is 'i'?

SELECT \* FROM EmployeeDetails WHERE FirstName LIKE '%i\_' 368

Write a query to display those employee whose firstname is having 's' as third letter?

SELECT \* FROM EmployeeDetails WHERE FirstName LIKE ' s%'

AND OPERATOR:

Whenever we want to apply a multiple condition and if all the condition is fulfilled on that scenario we

want to returned the data then we can use AND.

Write a query to display those employee who is working with departmentID 1 and Clientid 1?

SELECT \* FROM EmployeeDetails WHERE DepartmentID=1 AND ClientID=1

OR--->

if any one of the condition is true then it will return the data.

Write a query to display those employee who is working with departmentID 1 OR clientid 1?

SELECT \* FROM EmployeeDetails WHERE DepartmentID=1 OR ClientID=1

**JOIN:**

SELECT \* FROM EmployeeDetails

SELECT \* FROM Department

EmployeeID, FirstName, LastName, DepartmentName

TYPE OF JOIN:

a)-INNER JOIN

b)-LEFT JOIN

c)-RIGHT JOIN

d)-FULL OUTER JOIN

e)-CROSS JOIN

INNER JOIN---> this join will return those data which is matching between two table.

SELECT EmployeeID, FirstName, LastName, DepartmentName FROM EmployeeDetails INNER JOIN Department ON EmployeeDetails.DepartmentID=Department.DepartmentID

SELECT \* FROM Department

**Primary key** column only used to reference by another table

if we did not create a foreign on Employee table then we can face one problem, by mistake we can

insert a data in my employee details with department which is not present in my department table.

ex--> in my department table we have following record:

DepartmentID

1

2

3

4

5

and by mistake someone inserted a data in employee details table with departmentID 12 which is totally wrong, now if we want to apply such rule that before performing a insert statement in my employee details. SQL first validate the data for department and check that value is present in department table or not, if not present then it will throw error if value is present then it will successfully execute the statement and insert the data in our employee details table.

Note--> column which we are going to refer from another table that must be create as a primary key constraint.

CREATE TABLE DepartmentData\_25thMay

(DepartmentID INT PRIMARY KEy ,DepartmentName VARCHAR(50))

CREATE TABLE EmployeeData\_25thMay (EmployeeID INT, EmployeeName VARCHAR (50),

DepartmentID INT---->After creating a foreign column this column will allow only two type of value

1. -value which will be present in department table or it will allow null

CONSTRAINT FK\_DepartmentID FOREIGN KEY(DepartmentID) REFERENCES

DepartmentData\_25thMay(DepartmentID))

INSERT INTO EmployeeData\_25thMay(EmployeeID, EmployeeName, DepartmentID)

VALUES(1,'Sahil Jaiswal',1)

INSERT INTO EmployeeData\_25thMay(EmployeeID, EmployeeName, DepartmentID)

VALUES(2,'Raj',NULL)

INSERT INTO EmployeeData\_25thMay(EmployeeID, EmployeeName, DepartmentID)

VALUES(2,'Meenakshi',3)

SELECT \* FROM DepartmentData\_25thMay

INSERT INTO DepartmentData\_25thMay(DepartmentID,DepartmentName)

VALUES(1,'IT')

INSERT INTO DepartmentData\_25thMay(DepartmentID,DepartmentName)

VALUES(3,'Mechanical')

SELECT \* FROM DepartmentData\_25thMay

SELECT EmployeeID, EmployeeName, DepartmentName FROM EmployeeData\_25thMay

INNER JOIN DepartmentData\_25thMay ON EmployeeData\_25thMay.DepartmentID=DepartmentData\_25thMay.DepartmentID

SELECT EmployeeID, EmployeeName, DepartmentName FROM EmployeeData\_25thMay AS E

INNER JOIN DepartmentData\_25thMay AS D ON E.DepartmentID=D.DepartmentID

LEFT JOIN---> it will return all the data from left table and matching record from right table.

SELECT EmployeeID, FirstName, LastName, DepartmentName FROM EmployeeDetails -🡪LEFT TABLE

LEFT JOIN Department ON Department.DepartmentID=EmployeeDetails.DepartmentID-->RIGHT TABLE

SELECT EmployeeID,FirstName AS FName,LastName AS LName,DepartmentName FROM

EmployeeDetails AS ED ------------------->LEFT TABLE

LEFT JOIN Department AS D ON ED.DepartmentID=D.DepartmentID--->RIGHT

RIGHT JOIN-----> it will return all the data from right table and matching record from left table

SELECT EmployeeID, FirstName AS FName, LastName AS LName, DepartmentName FROM

EmployeeDetails AS ED ------------------->LEFT TABLE

RIGHT JOIN Department AS D ON ED.DepartmentID=D.DepartmentID--->RIGHT

FULL OUTER JOIN (LEFT JOIN + RIGHT JOIN) --> it will return all the data from left table and all the data from right table will be consider as our output

SELECT EmployeeID, FirstName AS FName, LastName AS LName, DepartmentName FROM

EmployeeDetails AS ED ------------------->LEFT TABLE

FULL OUTER JOIN Department AS D ON ED.DepartmentID=D.DepartmentID--->RIGHT

CROSS JOIN:

SELECT EmployeeID, FirstName, LastName, DepartmentName FROM EmployeeDetails CROSS JOIN Department

SELECT EmployeeID, FirstName, LastName, Date FROM DimDate CROSS JOIN EmployeeDetails

WHERE Date>='2021-01-04' AND Date<='2021-01-08'

FUNCTION--->Function we can use to perform some specific task

Write a query to display current datetime?

SELECT GETDATE()

TWO TYPES OF FUNCTION:

1)-BUILT-IN FUNCTION--(SYSTEM DEFINED FUNCTION) ----> those function which is already available within our system that is called system, defined.

2)-USER DEFINED FUNCTION---> those function which is developed by the user and then we can use those function with in our code that is called user defined function.

CATEGORY WITH IN SYSTEM DEFINED FUNCTION:

a)-STRING CATEGORY

a)-UPPER ()

b)-LOWER ()

c)-LTRIM ()

d)-RTRIM ()

e)-TRIM ()

f)-LEN ()

g)-REPLACE ()

h)-REVERSE ()

i)-SUBSTRING ()

UPPER ()---> this function is used to transform the data into upper case.

SELECT UPPER('sahil')

SELECT FirstName, UPPER(FirstName) AS New\_FirstName FROM EmployeeDetails

SELECT \* FROM EmployeeDetails

LOWER()--------> this function is used to convert the value into lower case?

SELECT LOWER('SAHIL')

SELECT FirstName, LOWER(FirstName) FROM EmployeeDetails

LTRIM()---> this function is used to remove the blank space from left side/ Leading space.

SELECT LTRIM (' Bharat')

'Bharat'

SELECT FirstName, LTRIM(FirstName) FROM EmployeeDetails

UPDATE EmployeeDetails SET FirstName=LTRIM(FirstName)

' Vishal'

'Vishal'

RTRIM () -----> this function is used to remove the Space from Right side/ Trailing Space.

SELECT RTRIM ('Sahil ')

'Sahil'

SELECT FirstName, RTRIM(FirstName) FROM EmployeeDetails

'Sahil '

'Sahil'

UPDATE EmployeeDetails SET FirstName=' Sahil ' WHERE EmployeeID=1

NESTED FUNCTION----> Function with in a another function

SELECT FirstName, RTRIM(LTRIM(FirstName)), FROM EmployeeDetails

'Sahil '

' Sahil '

'Sahil'

SELECT FirstName, LTRIM(RTRIM(FirstName)), FROM EmployeeDetails

'Sahil '

' Sahil '

'Sahil'

SELECT 2+3

SELECT 3+2

TRIM()---> this function is used to remove the space from both side.

SELECT FirstName, TRIM(FirstName) FROM EmployeeDetails

'Sahil'

REVERSE () ----> this function will change the sequence of data.

SELECT REVERSE('Sahil’) --->'lihas'

'lihaS'

Write a query to display the FirstName value in reverse order?

Write a query to add a new column (RFirstName) in EmployeeDetails table and update the Reverse value from firstname column?

ALTER TABLE EmployeeDetails ADD RFirstName VARCHAR(50)

SELECT \* FROM EmployeeDetails UPDATE EmployeeDetails SET RFirstName=REVERSE(FirstName)

DESC

LEN()----> it will count the no of character is present in our data.

SELECT LEN (TRIM (' Sahil'))

SELECT ' Sahil'

SELECT 'Sahil '

SELECT LEN ('Sahil ')

REPLACE()---> this function is used to replace the exisiting character with new one.

This function will accept below parameter

a)-ColumnName/String Data

b)-Character/String Data which we want to change

c)-New Character/String

Write a query to replace 'a' with 'b' from 'Sahil'?

SELECT REPLACE ('Sahil', 'a', 'b')

-->Sbhil

Write a query to replace 's' with 'l' from firstname column?

SELECT REPLACE(FirstName,'s','l') FROM EmployeeDetails

SELECT \* FROM EmployeeDetails

SUBSTRING () ---> this function we can used to extract the sub part from the entire string value then we can use substring.

This function can accept three parameters

a)-String Data/ColumnName

b)-Starting Position No

c)-Len-->How many characters we want to extract

'Sahil'---

12345

Rajesh

Write a query a display first three letter from 'sahil' value?

SELECT SUBSTRING('sahil',1,3)

Write a query to display a third and forth letter from 'sahil'?

SELECT SUBSTRING('Sahil',3,2)

Write a query to display a third and fifth letter from 'sahil'?

SELECT SUBSTRING('Sahil',3,1),SUBSTRING('Sahil',5,1)

SELECT 'Sahil'+' '+'Jaiswal'

Write a query to Concat FirstName and LastName Column and display as a EmployeeName output?

SELECT TRIM(FirstName) +' ' + TRIM(LastName) As EmployeeName FROM EmployeeDetails

CONCAT ()

SELECT CONCAT (FirstName,' ', LastName) FROM EmployeeDetails

SELECT 'podila' + ' ' +'meghana'

Sahil Jaiswal

SELECT TRIM (FirstName +' ' + LastName) As EmployeeName FROM EmployeeDetails

SELECT TRIM(FirstName) +' ' + TRIM(LastName) As EmployeeName FROM EmployeeDetails

DATE CATEGORY: ---> those function which will work on date/datetime

a)-GETDATE ()

b)-DAY ()

c)-MONTH ()

d)-YEAR ()

e)-DATEADD ()

f)-DATEDIFF ()

g)-DATENAME ()

GETDATE () ---> This function will display current Datetime

'yyyy-mm-dd hh:mm:sec.ms'

SELECT GETDATE()

'2025-05-31 16:32:26.170'

DAY () ---> this function will return the Day no from Date value.

SELECT DAY ('2025-05-31')

SELECT DOB, DAY(DOB) FROM EmployeeDetails

MONTH () -----> this function will return the month no from Date/Datetime value.

SELECT

MONTH (GETDATE ())

YEAR () ----> this function will return the year no.

SELECT YEAR ('1999-01-01')

SELECT YEAR (GETDATE ())

'2025-05-31'

DATEADD () ---> this function is used to add no of days/no of month/no of year in any date value.

This function will accept below parameter:

a)-INTERVAL

a)-dd---->Day

b)-mm---->Month

c)-yy---->Year

b)-How many Days/Month/ year we want to add

c)-DATE VALUE/ColummName

'2025-05-01'

Write a query to add 5 Days in 1st may 2025?

SELECT DATEADD (dd,5,'2025-05-01')

Write a query to add 3 months in current date?

SELECT DATEADD (mm,3, GETDATE ())

Write a query to add 5 year in DOB column?

SELECT DOB, DATEADD (yy,5, DOB) FROM EmployeeDetails

DATEDIFF () ---> if we want to calculate the difference between two date then we can use DateDiff function?

Below is the parameter which we have to pass while using DateDiff function:

a)-INTERVAL

a)-dd-->Day

b)-mm-->month

c)-yy-->year

b)-Startdate

c)-EndDate

StartDate-->'2025-05-10'

EndDate---->'2025-05-17'

--Day difference

SELECT DATEDIFF (dd,'2025-05-10','2025-05-17')

SELECT DATEDIFF (mm,'2025-01-05','2026-02-01')

Write a query to Find out the Age in term of year for every employee which is present in my employeedetails?

SELECT \*, DATEDIFF (yy, DOB, GETDATE ()) FROM EmployeeDetails

DATENAME ()

SELECT DATENAME (mm, GETDATE ())

SELECT DATENAME (mm,'1999-10-01')

SELECT DATENAME (WeekDay, GETDATE ())

SELECT DATENAME(WeekDay,'2025-06-01')

SELECT DATENAME(WeekDay,'2025-06-01')

MATHMATICAL FUNCTION:

a)-POWER ()

b)-SQRT ()

c)-ABS ()

d)-FLOOR ()

e)-CEILING ()

2^3--->2\*2\*2

SELECT POWER (2,3)

25

ABS () -->it will convert all negative value into positive.

SELECT ABS (-100)

SELECT ABS (10)

FLOOR () ---> this function will work on the decimal value; it will return nearest smallest integer value.

SELECT FLOOR (10.45)

10 11

SELECT FLOOR (99.99)

99 100

CEILING () ----> this function will return nearest largest integer value.

SELECT CEILING (99.99)

99 100

SELECT CEILING (-75.32)

-75 -76