

## 03-01-2026 Weekly Assignment - 1

---

### Create Sales Database

```
CREATE DATABASE Sales;
USE Sales;
```

### Create Tables

```
CREATE TABLE Client (
    ClientNo VARCHAR(6) PRIMARY KEY,
    Name VARCHAR(20) NOT NULL,
    Address1 VARCHAR(30),
    Address2 VARCHAR(30),
    City VARCHAR(15),
    PinCode NUMERIC(8),
    State VARCHAR(15),
    BalDue DECIMAL(10,2),
    CHECK (ClientNo LIKE 'C%')
);
```

```
CREATE TABLE Product (
    ProductNo VARCHAR(6) PRIMARY KEY,
    Description VARCHAR(15) NOT NULL,
    ProfitPerc DECIMAL(4,2) NOT NULL,
    UnitMeasure VARCHAR(10) NOT NULL,
    QtyOnHand NUMERIC(8) NOT NULL,
    ReorderLvl NUMERIC(8) NOT NULL,
    SellPrice DECIMAL(8,2) NOT NULL,
    CostPrice DECIMAL(8,2) NOT NULL,
    CHECK (ProductNo LIKE 'P%'),
```

```

        CHECK  (SellPrice <> 0),
        CHECK  (CostPrice <> 0)
);

CREATE TABLE Salesman (
    SalesmanNo VARCHAR(6) PRIMARY KEY,
    SalesmanName VARCHAR(20) NOT NULL,
    Address1 VARCHAR(30) NOT NULL,
    Address2 VARCHAR(30),
    City VARCHAR(20),
    PinCode NUMERIC(8),
    State VARCHAR(20),
    SalAmt DECIMAL(8,2) NOT NULL,
    TgtToGet DECIMAL(6,2) NOT NULL,
    YtdSales DECIMAL(6,2) NOT NULL,
    Remarks VARCHAR(60),
    CHECK  (SalesmanNo LIKE 'S%'),
    CHECK  (SalAmt <> 0)
);

```

```

CREATE TABLE SalesOrder (
    OrderNo VARCHAR(6) PRIMARY KEY,
    ClientNo VARCHAR(6),
    OrderDate DATE,
    DelyAddr VARCHAR(25),
    SalesmanNo VARCHAR(6),
    DelyType CHAR(1),
    BilledYN CHAR(1),
    DelyDate DATE,
    OrderStatus VARCHAR(20),
    CHECK  (OrderNo LIKE 'O%'),
    CHECK  (DelyType IN ('P','F'))
);

```

```

        CHECK (BilledYN IN ('Y', 'N')) ,
        CHECK (OrderStatus IN ('In
Process', 'Fulfilled', 'Backorder', 'Cancelled')) ,
        FOREIGN KEY (ClientNo) REFERENCES Client(ClientNo),
        FOREIGN KEY (SalesmanNo) REFERENCES Salesman(SalesmanNo)
);

CREATE TABLE SalesOrderDetails (
    OrderNo VARCHAR(6),
    ProductNo VARCHAR(6),
    QtyOrdered NUMERIC(8),
    QtyDisp NUMERIC(8),
    ProductRate DECIMAL(10,2),
    PRIMARY KEY (OrderNo, ProductNo),
    FOREIGN KEY (OrderNo) REFERENCES SalesOrder(OrderNo),
    FOREIGN KEY (ProductNo) REFERENCES Product(ProductNo)
);

```

## **Insert data into tables**

```

INSERT INTO Client VALUES
('C00001', 'Ivan Bayross', 'Addr1', 'Addr2', 'Mumbai', 400054,
'Maharashtra', 15000),
('C00002', 'Ravi Kumar', 'Addr1', 'Addr2', 'Delhi', 110001,
'Delhi', 20000),
('C00003', 'Anita Sharma', 'Addr1', 'Addr2', 'Bangalore',
560001, 'Karnataka', 12000);

```

```

INSERT INTO Product VALUES
('P00001', 'T-Shirts', 5.00, 'Piece', 200, 50, 350, 250),

```

```
('P00002', 'Jeans', 6.50, 'Piece', 150, 40, 1200, 900),  
('P00003', 'Shoes', 7.00, 'Pair', 100, 30, 2500, 2000);  
  
INSERT INTO Salesman VALUES  
('S00001', 'Aman', 'A/14', 'Worli', 'Mumbai', 400002,  
'Maharashtra', 3000, 100, 50, 'Good'),  
('S00002', 'Rahul', 'B/21', 'Karol Bagh', 'Delhi', 110005,  
'Delhi', 3500, 150, 75, 'Excellent'),  
('S00003', 'Sneha', 'C/10', 'Indiranagar', 'Bangalore', 560038,  
'Karnataka', 3200, 120, 60, 'Average');
```

```
INSERT INTO SalesOrder VALUES  
('O19001', 'C00001', '2002-06-12', 'Kochi', 'S00001', 'F', 'N',  
'2002-07-20', 'In Process'),  
('O19002', 'C00002', '2002-07-10', 'Hyderabad', 'S00002', 'P',  
'Y', '2002-07-25', 'Fulfilled'),  
('O19003', 'C00003', '2002-08-05', 'Mumbai', 'S00003', 'F', 'N',  
'2002-08-30', 'Backorder');
```

```
INSERT INTO SalesOrderDetails VALUES  
('O19001', 'P00001', 4, 4, 525),  
('O19002', 'P00002', 2, 2, 1300),  
('O19003', 'P00003', 1, 0, 2700);
```

-- Answer following queries with the help of above schema:

1. Display the names of all the clients.

```
select Name from Client;
```

Name	
1	Ivan Bayross
2	Ravi Kumar
3	Anita Sharma

2. Display all the clients who are located in Mumbai.

```
select * from Client  
where City='Mumbai';
```

ClientNo	Name	Address1	Address2	City	PinCode	State	BalDue
C00001	Ivan Bayross	Addr1	Addr2	Mumbai	400054	Maharashtra	15000.00

3. Display all the products whose selling price is > 2000 and < 5000.

```
select * from Product  
where SellPrice>2000 and SellPrice<5000;
```

ProductNo	Description	ProfitPerc	UnitMeasure	QtyOnHand	ReorderLvl	SellPrice	CostPrice
P00003	Shoes	7.00	Pair	100	30	2500.00	2000.00

4. Display Name, City and State of Clients not in the state of Maharashtra.

```
select Name,City,State from Client  
where State!='Maharashtra';
```

	Name	City	State
1	Ravi Kumar	Delhi	Delhi
2	Anita Sharma	Bangalore	Karnataka

5. Display all the information of client\_no C0001 and C0002.

```
select * from Client  
where ClientNo='C0001' or ClientNo='C0002';
```

ClientNo	Name	Address1	Address2	City	PinCode	State	BalDue	phone_no

6. Change the selling price of '1.44 drive' to Rs. 1150.50.

```
update Product  
set SellPrice=1150.50  
where Description='1.44 drive';
```

## Messages

17:10:52      Started executing query at Line 125  
(0 rows affected)  
Total execution time: 00:00:00.033

7. Delete the record of client\_no C0005.

```
delete from Client  
where ClientNo='C0005';
```

### Messages

17:11:11      Started executing query at Line 129  
(0 rows affected)  
Total execution time: 00:00:00.025

8. Display the clients who stay in a city whose second letter is 'a'.

```
select * from Client  
where City LIKE '_a%';
```

Results										Messages	
	ClientNo	Name	Address1	Address2	City	PinCode	State	BalDue			
1	C00003	Anita Sharma	Addr1	Addr2	Bangalore	560001	Karnataka	12000.00			

9. Count the number of products having price greater than or equal to 1500.

```
select count(*) from Product  
where SellPrice>=1500;
```

Results		Messages	
	(No column name)		
1	1		

```
10. Display qtyordered, qtydisp and balancedqty (not in table).  
select sod.QtyOrdered,sod.QtyDisp,p.QtyOnHand as balanceQty from  
SalesOrderDetails as sod  
join product as p  
on sod.ProductNo=p.ProductNo;
```

	QtyOrdered	QtyDisp	balanceQty
1	4	4	200
2	2	2	150
3	1	0	100

-- Write commands to the following

1. Make Client\_no as primary key in client\_master.

```
ALTER TABLE Client  
ADD CONSTRAINT pk_client PRIMARY KEY(ClientNo);
```

2. Add a new column phone\_no in the client\_master table.

```
ALTER TABLE Client  
ADD phone_no VARCHAR(15);
```

3. Add the not null constraint in the product\_master table with the column description, profit percent, sell price and cost price.

```
ALTER TABLE Product  
ALTER COLUMN Description VARCHAR(15) NOT NULL;
```

```
ALTER TABLE Product  
ALTER COLUMN ProfitPerc DECIMAL(4,2) NOT NULL;
```

```
ALTER TABLE Product  
ALTER COLUMN SellPrice DECIMAL(8,2) NOT NULL;
```

```
ALTER TABLE Product  
ALTER COLUMN CostPrice DECIMAL(8,2) NOT NULL;
```

4. Change size of name column to 60 in client\_master table.

```
ALTER TABLE Client  
ALTER COLUMN Name VARCHAR(60);
```

5. Remove pincode column from table.

```
ALTER TABLE Client  
DROP COLUMN PinCode;
```

-- Define in 1 or 2 lines and give one example also

1. Recursive Relationship.

Relationship where a table is related to itself is called recursive relationship.

2. Composite key.

A composite key is a primary key made up of two or more columns used together to uniquely identify a record.

3. The 'LIKE' operator with pattern matching.

The LIKE operator is used to search for a specified pattern in a column using wildcard characters % and \_.

#### 4. Drop Table command.

The DROP TABLE command permanently removes a table and all its data from the database schema.

#### 5. Full Outer Join.

A full outer join returns all records from both tables, matching rows where possible and NULLs where no match exists.

-- Write queries for following descriptions (JOINS)

#### 1. Find out the products which have been sold to 'Ivan Bayross'.

```
select p.Description from SalesOrderDetails as sod
join SalesOrder as so
on sod.OrderNo=so.OrderNo
join Product as p
on p.ProductNo=sod.ProductNo
join Client as c
on c.ClientNo=so.ClientNo
where c.Name='Ivan Bayross';
```

Results		Messages
		Description
1		T-Shirts

#### 2. Finding out the products and their quantities that will have to be delivered in the current month.

```

select p.Description, (QtyOrdered-QtyDisp) as
Quan_to_be_delivered from SalesOrderDetails as sod
join SalesOrder as so
on sod.OrderNo=so.OrderNo
join Product as p
on p.ProductNo=sod.ProductNo
where DATENAME(month,DelyDate)=DATENAME(month,GETDATE());

```

Results grid		Messages
Description	Quan_to_be_del...	

3. Listing the ProductNo and description of constantly sold (i.e. rapidly moving) products.

```

SELECT
    p.ProductNo,
    p.Description
FROM Product p
JOIN SalesOrderDetails s
ON p.ProductNo = s.ProductNo
GROUP BY p.ProductNo, p.Description
HAVING COUNT(s.ProductNo) > 1;

```

Results		Messages
ProductNo	Description	
<b>Results grid</b>		

4. Finding the names of clients who have purchased 'Trousers'.

```
select DISTINCT c.Name from SalesOrderDetails as sod
join SalesOrder as so
on sod.OrderNo=so.OrderNo
join Product as p
on p.ProductNo=sod.ProductNo
join Client as c
on c.ClientNo=so.ClientNo
where p.Description='Trousers';
```

Results		Messages
Name		

5. Listing the products and orders from customers who have ordered less than 5 units of 'Pull Overs'.

```
select so.OrderNo, p.ProductNo, p.Description, sod.QtyOrdered
from SalesOrderDetails as sod
join SalesOrder as so
```

```

on sod.OrderNo=so.OrderNo
join Product as p
on p.ProductNo=sod.ProductNo
where sod.QtyOrdered < 5 and p.Description='Pull Overs';

```

SELECT Name			
Results	Messages		
Results grid	ProductNo	Description	QtyOrdered

-- Write queries for following descriptions (sub queries)

1. Finding the non-moving products i.e. products not being sold.

```

select * from Product
where ProductNo NOT IN (select ProductNo from
SalesOrderDetails);

```

SELECT Name							
ProductNo	Description	ProfitPerc	UnitMeasure	QtyOnHand	ReorderLvl	SellPrice	CostPrice
Results grid							

2. Finding the name and complete address for the customer who has placed Order number '019001'.

```

SELECT Name,
       Address1 + ', ' + Address2 + ', ' + City + ', ' + State AS
Complete_Address
FROM Client
WHERE ClientNo = (
    SELECT ClientNo
    FROM SalesOrder
    WHERE OrderNo = '019001'
)

```

) ;

Results		Messages
Results grid		
	Name	Complete_Address
1	Ivan Bayross	Addr1, Addr2, Mumbai, Maharashtra

3. Finding the clients who have placed orders before the month of May'02.

```
SELECT Name  
FROM Client  
WHERE ClientNo IN (  
    SELECT ClientNo  
    FROM SalesOrder  
    WHERE OrderDate < '2002-05-01'  
) ;
```

Results		Messages
Results grid		
	Name	

-- Write commands to do following

1. Display system date as Saturday, February 11, 2012.

```
SELECT FORMAT(CONVERT(date, GETDATE()), 'dddd, MMMM dd, yyyy') AS System_Date;
```

Results		Messages
	System_Date	▼
1	Saturday, January 03, 2026 Saturday, January 03, 2026	

2. Display balance Due from client master as \$99,999.99

```
SELECT 99999.99 AS "Balance Due"  
FROM Client;
```

3. Display message as

```
'Salesman Aman sold goods of 50 while given target was 100.'  
SELECT  
'Salesman ' + SalesmanName +  
' sold goods of ' + CAST(YtdSales AS VARCHAR) +  
' while given target was ' + CAST(TgtToGet AS VARCHAR)  
AS Message  
FROM Salesman  
WHERE SalesmanName = 'Aman';
```

Results		Messages
	Message	▼
1	Salesman Aman sold goods of 50.00 while given target was 100.00	

4. Display your Age in Years.

```
select  
case  
    when DATEADD(YEAR, DATEDIFF(YEAR, '2005-02-09', GETDATE()),  
'2005-02-09') > GETDATE()  
        then DATEDIFF(YEAR, '2005-02-09', GETDATE()) - 1  
    else DATEDIFF(YEAR, '2005-02-09', GETDATE())  
end as Age_In_Years;
```

Results		Messages
Age_In_Years		▼
1	20	