WeeklyAssignment – 1

Database and Table Creation:

```
CREATE DATABASE Sales;

USE Sales;


CREATE TABLE CLIENT_MASTER (
   CLIENTNO    VARCHAR(6) PRIMARY KEY,
   NAME       VARCHAR(20) NOT NULL,
   ADDRESS1    VARCHAR(30),
   ADDRESS2    VARCHAR(30),
   CITY       VARCHAR(15),
   PINCODE     INT,
   STATE      VARCHAR(15),
   BALDUE     DECIMAL(10,2)
);


CREATE TABLE PRODUCT_MASTER (
   PRODUCTNO     VARCHAR(6) PRIMARY KEY,
   DESCRIPTION    VARCHAR(15) NOT NULL,
   PROFITPERCENT  DECIMAL(4,2) NOT NULL,
   UNITMEASURE    VARCHAR(10) NOT NULL,
   QTYONHAND     INT NOT NULL,
   REORDERLVL     INT NOT NULL,
   SELLPRICE     DECIMAL(8,2) NOT NULL CHECK (SELLPRICE > 0),
   COSTPRICE     DECIMAL(8,2) NOT NULL CHECK (COSTPRICE > 0)
```

```
);

CREATE TABLE SALESMAN_MASTER (
   SALESMANNO  VARCHAR(6) PRIMARY KEY,
   SALESMANNAME VARCHAR(20) NOT NULL,
   ADDRESS1    VARCHAR(30),
   ADDRESS2    VARCHAR(30),
   CITY       VARCHAR(20),
   PINCODE     INT,
   STATE      VARCHAR(20),
   SALAMT     DECIMAL(8,2) NOT NULL CHECK (SALAMT >= 0),
   TGTTOGET    DECIMAL(6,2) NOT NULL,
   YTDSALES    DECIMAL(6,2) NOT NULL,
   REMARKS     VARCHAR(60)
);

CREATE TABLE SALES_ORDER (
   ORDERNO     VARCHAR(6) PRIMARY KEY,
   CLIENTNO    VARCHAR(6),
   ORDERDATE   DATE,
   DELYADDR    VARCHAR(25),
   SALESMANNO  VARCHAR(6),
   DELYTYPE    CHAR(1) CHECK (DELYTYPE IN ('P','F')),
   BILLEDYN    CHAR(1) CHECK (BILLEDYN IN ('Y','N')),
   DELYDATE    DATE,
   ORDERSTATUS VARCHAR(15)
```

```
    CHECK (ORDERSTATUS IN ('In Process','Fulfilled','Backorder','Cancelled')),
  FOREIGN KEY (CLIENTNO) REFERENCES CLIENT_MASTER(CLIENTNO),
  FOREIGN KEY (SALESMANNO) REFERENCES SALESMAN_MASTER(SALESMANNO)
);

CREATE TABLE SALES_ORDER_DETAILS (
  ORDERNO    VARCHAR(6),
  PRODUCTNO   VARCHAR(6),
  QTYORDERED  INT,
  QTYDISP    INT,
  PRODUCTRATE DECIMAL(10,2),
  PRIMARY KEY (ORDERNO, PRODUCTNO),
  FOREIGN KEY (ORDERNO) REFERENCES SALES_ORDER(ORDERNO),
  FOREIGN KEY (PRODUCTNO) REFERENCES PRODUCT_MASTER(PRODUCTNO)
);
```

Sample Data Insertion:

```
INSERT INTO CLIENT_MASTER VALUES
('C00001','Ivan Bayross','Mumbai','Maharashtra','Mumbai',400054,'Maharashtra',15000),
('C00002','Mamta Muzumdar','Madras','Tamil Nadu','Chennai',780001,'Tamil Nadu',0),
('C00003','Chhaya Bankar','Mumbai','Maharashtra','Mumbai',400057,'Maharashtra',5000),
('C00004','Ashwini Joshi','Bangalore','Karnataka','Bangalore',560001,'Karnataka',0),
('C00005','Hansel Colaco','Mumbai','Maharashtra','Mumbai',400060,'Maharashtra',2000);

INSERT INTO PRODUCT_MASTER VALUES
('P00001','T-Shirts',5,'Piece',200,50,350,250),
```

('P00002','Shirts',6,'Piece',150,50,500,350),

('P00003','Jeans',5,'Piece',100,30,700,500),

('P00004','Pull Overs',8,'Piece',80,20,1200,900),

('P00005','Denim Shirts',4,'Piece',60,20,750,550);

INSERT INTO SALESMAN_MASTER VALUES

('S00001','Aman','Mumbai','', 'Mumbai',400002,'Maharashtra',3000,100,50,'Good'),

('S00002','Ravi','Delhi','', 'Delhi',110001,'Delhi',3000,200,100,'Good'),

('S00003','Kiran','Chennai','', 'Chennai',600001,'Tamil Nadu',3000,300,150,'Excellent');

INSERT INTO SALES_ORDER VALUES

('O19001','C00001','2022-06-12','Mumbai','S00001','F','N','2022-07-20','In Process'),

('O19002','C00002','2022-06-25','Chennai','S00002','P','Y','2022-06-30','Fulfilled'),

('O19003','C00001','2022-07-02','Mumbai','S00001','F','Y','2022-07-10','Fulfilled'),

('O19004','C00003','2022-07-10','Mumbai','S00003','P','N','2022-07-20','Backorder');

INSERT INTO SALES_ORDER_DETAILS VALUES

('O19001','P00001',4,4,350),

('O19001','P00002',2,2,500),

('O19002','P00003',3,3,700),

('O19003','P00004',1,1,1200),

('O19004','P00005',2,0,750);

a. Queries:

1. **Display the names of all the clients**
   SELECT NAME

FROM CLIENT_MASTER;

Results Messages

	CLIENTNO	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BALDUE
1	C00001	Ivan Bayross	Mumbai	Maharashtra	Mumbai	400054	Maharashtra	15000.00
2	C00002	Mamta Muzumdar	Madras	Tamil Nadu	Chennai	780001	Tamil Nadu	0.00
3	C00003	Chhaya Bankar	Mumbai	Maharashtra	Mumbai	400057	Maharashtra	5000.00
4	C00004	Ashwini Joshi	Bangalore	Karnataka	Bangalore	560001	Karnataka	0.00
5	C00005	Hansel Colaco	Mumbai	Maharashtra	Mumbai	400060	Maharashtra	2000.00

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

```
SELECT *
FROM CLIENT_MASTER
WHERE CITY = 'Mumbai';
```

Results Messages

	CLIENTNO	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BALDUE
1	C00001	Ivan Bayross	Mumbai	Maharashtra	Mumbai	400054	Maharashtra	15000.00
2	C00003	Chhaya Bankar	Mumbai	Maharashtra	Mumbai	400057	Maharashtra	5000.00
3	C00005	Hansel Colaco	Mumbai	Maharashtra	Mumbai	400060	Maharashtra	2000.00

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

```
SELECT *
FROM PRODUCT_MASTER
WHERE SELLPRICE > 2000 AND SELLPRICE < 5000;
```

Results Messages

	PRODUCTNO	DESCRIPTION	PROFITPERCENT	UNITMEASURE	QTYONHAND	REORDERLVL	SELLPRICE	COSTPRICE

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

```
SELECT NAME, CITY, STATE
FROM CLIENT_MASTER
WHERE STATE <> 'Maharashtra';
```

Results Messages

	NAME	CITY	STATE
1	Mamta Muzumdar	Chennai	Tamil Nadu
2	Ashwini Joshi	Bangalore	Karnataka

5. **Display all the information of client no C00001 and C00002**

```
SELECT *
FROM CLIENT_MASTER
WHERE CLIENTNO IN ('C00001','C00002');
```

Results | Messages

	CLIENTNO	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BALDUE
1	C00001	Ivan Bayross	Mumbai	Maharashtra	Mumbai	400054	Maharashtra	15000.00
2	C00002	Mamta Muzumdar	Madras	Tamil Nadu	Chennai	780001	Tamil Nadu	0.00

6. **Change the selling price of product 'T-Shirts' to Rs. 1150.50**

```
UPDATE PRODUCT_MASTER
SET SELLPRICE = 1150.50
WHERE DESCRIPTION = 'T-Shirts';
```

7. **Delete the record of client no C00005**

```
DELETE FROM CLIENT_MASTER
WHERE CLIENTNO = 'C00005';
```

8. **Display the clients whose city name starts with letter 'a'**

```
SELECT *
FROM CLIENT_MASTER
WHERE CITY LIKE 'A%';
```

Results | Messages

	CLIENTNO	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BALDUE

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

```
SELECT COUNT(*) AS Product_Count
FROM PRODUCT_MASTER
WHERE SELLPRICE >= 1500;
```

Results | Messages

	Product_Count
1	0

10. **Display qtyordered, qtydisp and balanceqty (not in table)**

```
SELECT
  ORDERNO,
  PRODUCTNO,
  QTYORDERED,
  QTYDISP,
  (QTYORDERED - QTYDISP) AS BALANCEQTY
FROM SALES_ORDER_DETAILS;
```

Results Messages

	ORDERNO	PRODUCTNO	QTYORDERED	QTYDISP	BALANCEQTY
1	O19001	P00001	4	4	0
2	O19001	P00002	2	2	0
3	O19002	P00003	3	3	0
4	O19003	P00004	1	1	0
5	O19004	P00005	2	0	2

b. Write commands to do following:

1. **Make CLIENTNO as primary key in client master**

```
ALTER TABLE CLIENT_MASTER
ADD CONSTRAINT PK_CLIENT_MASTER PRIMARY KEY (CLIENTNO);
```

2. **Add a new column PHONE_NO in the client master table**

```
ALTER TABLE CLIENT_MASTER
ADD PHONE_NO VARCHAR(15);
```

3. **Add the NOT NULL constraint in product master table**

```
ALTER TABLE PRODUCT_MASTER
ALTER COLUMN DESCRIPTION VARCHAR(15) NOT NULL;

ALTER TABLE PRODUCT_MASTER
ALTER COLUMN PROFITPERCENT DECIMAL(4,2) NOT NULL;

ALTER TABLE PRODUCT_MASTER
ALTER COLUMN SELLPRICE DECIMAL(8,2) NOT NULL;

ALTER TABLE PRODUCT_MASTER
ALTER COLUMN COSTPRICE DECIMAL(8,2) NOT NULL;
```

4. **Change size of NAME column to 60 in client master**
   ALTER TABLE CLIENT_MASTER
   ALTER COLUMN NAME VARCHAR(60);

5. **Remove PINCODE column from client master table**
   ALTER TABLE CLIENT_MASTER
   DROP COLUMN PINCODE;

c. Define in 1 or 2 lines and give one example:

1. **Recursive Relationship**
   A relationship where a table is related to itself, usually using a foreign key that references its own primary key.
   Example: An employee managing another employee in the same table.

2. **Composite Key**
   A primary key made using two or more columns to uniquely identify a record in a table.
   Example: (OrderNo, ProductNo) in Sales_Order_Details.

3. **LIKE Operator with Pattern Matching**
   Used to search for a specified pattern in a column using wildcards like % and _.
   Example: WHERE CITY LIKE 'M%'

4. **DROP TABLE Command**
   Used to permanently delete a table along with its structure and data from the database.
   Example: DROP TABLE CLIENT_MASTER;

5. **Full Outer Join**
   Returns all records from both tables, matching rows where possible and NULL where no match exists.
   Example: SELECT * FROM A FULL OUTER JOIN B ON A.id = B.id;

d. Joins Queries:

1. **Products sold to ‘Ivan Bayross’**

```
SELECT DISTINCT p.PRODUCTNO, p.DESCRIPTION
FROM CLIENT_MASTER c
JOIN SALES_ORDER s ON c.CLIENTNO = s.CLIENTNO
JOIN SALES_ORDER_DETAILS d ON s.ORDERNO = d.ORDERNO
JOIN PRODUCT_MASTER p ON d.PRODUCTNO = p.PRODUCTNO
WHERE c.NAME = 'Ivan Bayross';
```

Results Messages

	PRODUCTNO	DESCRIPTION
1	P00001	T-Shirts
2	P00002	Shirts
3	P00004	Pull Overs

2. **Products and quantities to be delivered in the current month**

```
SELECT p.DESCRIPTION, d.QTYORDERED
FROM SALES_ORDER s
JOIN SALES_ORDER_DETAILS d ON s.ORDERNO = d.ORDERNO
JOIN PRODUCT_MASTER p ON d.PRODUCTNO = p.PRODUCTNO
WHERE MONTH(s.DELYDATE) = MONTH(GETDATE())
AND YEAR(s.DELYDATE) = YEAR(GETDATE());
```

Results Messages

	DESCRIPTION	QTYORDERED

3. **Constantly sold (rapidly moving) products**

```
SELECT p.PRODUCTNO, p.DESCRIPTION
FROM PRODUCT_MASTER p
JOIN SALES_ORDER_DETAILS d ON p.PRODUCTNO = d.PRODUCTNO
GROUP BY p.PRODUCTNO, p.DESCRIPTION
HAVING SUM(d.QTYORDERED) > 5;
```

Results Messages

	PRODUCTNO	DESCRIPTION
1	P00001	T-Shirts

4. **Names of clients who purchased ‘Trousers’**

```
SELECT DISTINCT c.NAME
FROM CLIENT_MASTER c
JOIN SALES_ORDER s ON c.CLIENTNO = s.CLIENTNO
JOIN SALES_ORDER_DETAILS d ON s.ORDERNO = d.ORDERNO
```

```
JOIN PRODUCT_MASTER p ON d.PRODUCTNO = p.PRODUCTNO
WHERE p.DESCRIPTION = 'Trousers';
```

5. **Products and orders where less than 5 units of ‘Pull Overs’ were ordered**

```
SELECT s.ORDERNO, p.DESCRIPTION, d.QTYORDERED
FROM SALES_ORDER s
JOIN SALES_ORDER_DETAILS d ON s.ORDERNO = d.ORDERNO
JOIN PRODUCT_MASTER p ON d.PRODUCTNO = p.PRODUCTNO
WHERE p.DESCRIPTION = 'Pull Overs'
AND d.QTYORDERED < 5;
```

Results Messages

	ORDERNO	DESCRIPTION	QTYORDERED
1	O19003	Pull Overs	1

e. Subqueries:

1. **Find the non-moving products (products not being sold)**

```
SELECT *
FROM PRODUCT_MASTER
WHERE PRODUCTNO NOT IN (
  SELECT PRODUCTNO
  FROM SALES_ORDER_DETAILS
);
```

2. **Find the name and complete address of the customer who has placed Order number ‘019001’**

```
SELECT NAME, ADDRESS1, ADDRESS2, CITY, STATE
FROM CLIENT_MASTER
WHERE CLIENTNO = (
  SELECT CLIENTNO
  FROM SALES_ORDER
  WHERE ORDERNO = 'O19001'
);
```

Results Messages

	NAME	ADDRESS1	ADDRESS2	CITY	STATE
1	Ivan Bayross	Mumbai	Maharashtra	Mumbai	Maharashtra

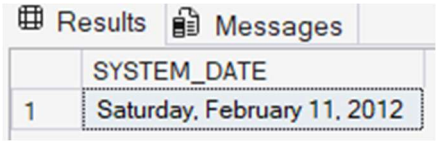
3. **Find the clients who have placed orders before the month of May'02**

```
SELECT *
FROM CLIENT_MASTER
WHERE CLIENTNO IN (
  SELECT CLIENTNO
  FROM SALES_ORDER
  WHERE ORDERDATE < '2002-05-01'
);
```

f. Write Commands to do following:

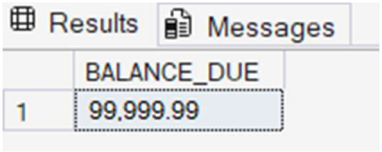
1. **Display system date as Saturday, February 11, 2012**

```
SELECT FORMAT(
  CAST('2012-02-11' AS DATE),
  'dddd, MMMM dd, yyyy'
) AS SYSTEM_DATE;
```



2. **Display balance due as 99,999.99**

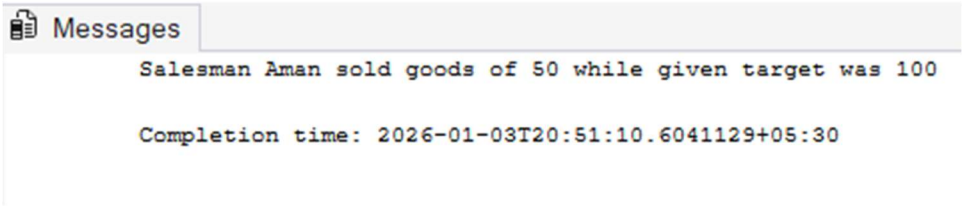
```
SELECT FORMAT(99999.99, 'N2') AS BALANCE_DUE;
```



3. **Display message: "Salesman Aman sold goods of 50 while given target was 100."**

```
DECLARE @name VARCHAR(20) = 'Aman';
DECLARE @sold INT = 50;
DECLARE @target INT = 100;

PRINT 'Salesman ' + @name +
    ' sold goods of ' + CAST(@sold AS VARCHAR) +
    ' while given target was ' + CAST(@target AS VARCHAR);
```



4. **Display your age in years**

```
DECLARE @dob DATE = '2005-02-03';  -- change DOB if needed

SELECT DATEDIFF(YEAR, @dob, GETDATE()) -
    CASE
      WHEN DATEADD(YEAR, DATEDIFF(YEAR, @dob, GETDATE()), @dob) >
GETDATE()
      THEN 1 ELSE 0
    END AS AGE;
```