**Problem Statement:**

ABC Fashion is a leading retailer with a vast customer base and a team of dedicated sales representatives. They have a Sales Order Processing System that helps manage customer orders and interactions.

**Dataset:**

Find the below information and the script for the table creation and record insertion.

https://docs.google.com/document/d/1ngN7Q0Mpo8j5BXidNHGRHmgbMSuG5XcFYnp\_gD3woL

A/edit?usp=sharing

--Salesman table creation

CREATE TABLE Salesman (SalesmanId INT not null primary key, Name VARCHAR(255), Commission DECIMAL(10, 2), City VARCHAR(255), Age INT );

--Salesman table record insertion

INSERT INTO Salesman (SalesmanId, Name, Commission, City, Age)

VALUES (101, 'Joe', 50, 'California', 17),

(102, 'Simon', 75, 'Texas', 25),

(103, 'Jessie', 105, 'Florida', 35),

(104, 'Danny', 100, 'Texas', 22),

(105, 'Lia', 65, 'New Jersey', 30);

select\* from Salesman

-- Customer table creation

CREATE TABLE Customer( SalesmanId INT not null, CustomerId INT,CustomerName VARCHAR(255) not null,PurchaseAmount INT );

--Customer table record insertion

INSERT INTO Customer (SalesmanId, CustomerId, CustomerName, PurchaseAmount)

VALUES (101, 2345, 'Andrew', 550),

(103, 1575, 'Lucky', 4500),

(104, 2345, 'Andrew', 4000),

(107, 3747, 'Remona', 2700),

(110, 4004, 'Julia', 4545);

select\* from Customer

-- Orders table Creation

CREATE TABLE Orders (OrderId int , CustomerId int, SalesmanId int, Orderdate Date, Amount money)

--Orders table record insertion

INSERT INTO Orders Values (5001,2345,101,'2021-07-01',550),

(5003,1234,105,'2022-02-15',1500)

select\* from Orders

-- 1. Insert a new record in your Orders table.

INSERT INTO Orders (OrderId, CustomerId, SalesmanId, Orderdate, Amount)

Values (5002,2345,101,'2023-07-01',555)

select\* from Orders

/\*2) Add Primary key constraint for SalesmanId column in Salesman table. Add default constraint for City column in Salesman table. Add Foreign key constraint for SalesmanId column in Customer table. Add not null constraint in Customer\_name column for the Customer table.\*/

/\* a. primary key constaint in salesman table (already added while inserting data)

ALTER TABLE Salesman

ADD CONSTRAINT PK\_Salesman PRIMARY KEY (SalesmanId);

b. not null constraint in Customer table (already added while inserting data )

ALTER TABLE Customer

ALTER COLUMN Customername VARCHAR(29) NOT NULL; \*/

--c. Add default constraint for City column in Salesman table.

select \*From Salesman

alter table salesman

add constraint DF\_salesman\_city

default 'nw' for city

--d. Add Foreign key constraint for SalesmanId column in Customer table

/\* 3. Fetch the data where the Customer’s name is ending with ‘N’ also get the purchase amount value greater than 500.\*/

select\* from Customer

select \* from customer where CustomerName like '%N' and PurchaseAmount > 500

/\*4. Using SET operators, retrieve the first result with unique SalesmanId values from two tables,

and the other result containing SalesmanId with duplicates from two tables\*/

select salesmanid from Customer - -without duplicate

union

select SalesmanId from Salesman

select salesmanid from Customer -- with duplicate

union all

select SalesmanId from Salesman

select \* from Customer

select \* from Salesman

select\* from Orders

select \* from Customer

/\* 5. Display the below columns which has the matching data. Orderdate, Salesman Name, Customer Name, Commission, and City which has the range of Purchase Amount between 500 to 1500\*/

-- joining 3 table and finding out the common Purchase Amount between 500 to 1500

select c.customername, c.purchaseamount, b.salesmanid, b.Orderdate, a.Name, a.Commission, a.City from Salesman a

inner join Orders b on a.SalesmanId=b.SalesmanId

inner join Customer c on b.SalesmanId = c.SalesmanId

where c.PurchaseAmount between 500 and 1500

order by SalesmanId

SELECT o.OrderDate, s.Name AS SalesmanName, c.CustomerName, s.Commission, s.City

FROM Orders o

INNER JOIN Customer c ON o.CustomerId = c.CustomerId

INNER JOIN Salesman s ON o.SalesmanId = s.SalesmanId

WHERE c.PurchaseAmount BETWEEN 500 AND 1500

--joining salesman and orders table

select b.salesmanid, b.Orderdate, a.Name, a.Commission, a.City

from Salesman a

inner join Orders b

on a.SalesmanId=b.SalesmanId

/\*6. Using right join fetch all the results from Salesman and Orders table. \*/

select \* from Salesman

select\* from Orders

-- right joining only specific column of the table

select b.OrderId,b.CustomerId,b.Amount,b.salesmanid, b.Orderdate, a.Name, a.Commission, a.City from Salesman a

right join Orders b

on a.SalesmanId=b.SalesmanId

-- right joining orders table

select \* from Salesman a

right join Orders b

on a.SalesmanId=b.SalesmanId

-- right joing salesman table

select \* from orders a

right join Salesman b

on a.SalesmanId=b.SalesmanId