SQL Assignment 1

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_gD3woLA/edit?usp=sharing

```sql
-- Salesman Table Creation

CREATE TABLE Salesman(
    SalesmanId INT,
    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);

-- Display the records in Salesman table

select * from Salesman;

-- Customer table creation

CREATE TABLE Customer(
    SalesmanId INT,
    CustomerId INT,
    CustomerName VARCHAR(255),
    PurchaseAmount INT,
    );


-- Customer table record insertion
```

```sql
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);

-- Display the records in Customer table

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)

-- Display the records in Orders table

select * from Orders;

-- Tasks Performed:

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

INSERT INTO Orders Values
(5004,5687,108,'2020-05-15',1200)

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.*/


-- Add Primary key constraint for SalesmanId column in Salesman table.

ALTER TABLE Salesman ALTER COLUMN SalesmanId INT NOT NULL;

ALTER TABLE Salesman ADD CONSTRAINT PK_Salesman_SalesmanId PRIMARY KEY(SalesmanId);

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

ALTER TABLE Salesman ADD CONSTRAINT DF_Salesman_City DEFAULT('DefaultCity') FOR City;

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

ALTER TABLE Customer
ADD CONSTRAINT FK_Customer_SalesmanId
FOREIGN KEY(SalesmanId)
```

```
REFERENCES Salesman(SalesmanId);


UPDATE Customer
SET SalesmanId = NULL
WHERE SalesmanId IS NOT NULL
AND SalesmanId NOT IN (SELECT SalesmanId FROM Salesman);

ALTER TABLE Customer
ADD CONSTRAINT FK_Customer_SalesmanId
FOREIGN KEY (SalesmanId)
REFERENCES Salesman(SalesmanId);

-- Add not null constraint in Customer_name column for the Customer table.

ALTER TABLE Customer ALTER COLUMN CustomerName VARCHAR(255) NOT NULL;

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

select * from Customer;

SELECT *
FROM Customer
WHERE CustomerName LIKE '%N';

select * from Customer where PurchaseAmount > 500;


/* 4. Using SET operators, retrieve the first result with unique SalesmanId values from
two
tables, and the other result containing SalesmanId without duplicates from two tables. */

-- retrieve the first result with unique SalesmanId values from two tables

SELECT SalesmanId
FROM Salesman
UNION
SELECT SalesmanId
FROM Customer;


-- the other result containing SalesmanId without duplicates from two tables.

SELECT SalesmanId
FROM Salesman
UNION ALL
SELECT SalesmanId
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 1500 to 3000. */

select * from Salesman;

SELECT
    O.Orderdate,
```

```sql
    S.Name AS [Salesman Name],
    C.CustomerName AS [Customer Name],
    S.Commission,
    S.City
FROM
    Orders AS O
JOIN
    Salesman AS S ON O.SalesmanId = S.SalesmanId
JOIN
    Customer AS C ON O.CustomerId = C.CustomerId
WHERE
    C.PurchaseAmount BETWEEN 1500 AND 3000;



-- 6. Using right join fetch all the results from Salesman and Orders table.

SELECT
    S.SalesmanId,
    S.Name AS [Salesman Name],
    S.Commission,
    S.City,
    S.Age,
    O.OrderId,
    O.CustomerId,
    O.Orderdate,
    O.Amount
FROM
    Salesman AS S
RIGHT JOIN
    Orders AS O ON S.SalesmanId = O.SalesmanId;
```