SQL Mandatory Assignment 1

Tables & Records for SQL Mandatory Assignment 1 & 2

Salesman Table

SalesmanId	SalesmanName	Commission	City	Age
101	Joe	50	California	17
102	Simon	75	Texas	25
103	Jessie	105	Florida	35
104	Danny	100	Texas	22
105	Lia	65	New Jersy	30

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);
```

Customer Table

SalesmanId	CustomerId	CustomerName	PurchaseAmount
101	2345	Andrew	550
103	1575	Lucky	4500
104	2345	Andrew	4000
107	3747	Remona	2700
110	4004	Julia	4545

Customer table creation

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

Orders Table

OrderId	CustomerId	SalesmanId	OrderDate	Amount
5001	2345	101	04-07-2021	550
5003	1234	105	15-02-2022	1500

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)
```

Tasks to be Performed:

1. Insert a new record in your Orders table.
    - A new record is inserted into orders table

```
insert into orders values(5006,4322,111,'2023-12-16',3000);
```

```
108 %
Messages

(1 row affected)

Completion time: 2024-02-02T15:01:10.5565943+05:30
```

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.

    - Added Primary key constraint for SalesmanId column in Salesman table

```
alter table salesman alter column salesmanid int not null;
alter table salesman add constraint PK_constraint primary key (salesmanId);
```

```
108 %
Messages
Commands completed successfully.

Completion time: 2024-02-02T15:13:53.8164586+05:30
```

    - Added default constraint for City column in Salesman table

```
alter table salesman add constraint  Default_city default 'Hyderabad' for city;
```

```
108 %
Messages
Commands completed successfully.

Completion time: 2024-02-02T15:06:47.4534127+05:30
```

    - Added Foreign key constraint for SalesmanId column in Customer table

```
alter table Customer
alter column PurchaseAmount int not null;

ALTER TABLE Customer
add CONSTRAINT pk_purchaseamount Primary Key (PurchaseAmount);

ALTER TABLE Customer with nocheck
ADD FOREIGN KEY (SalesmanId) REFERENCES Salesman(SalesmanId);
```

```
108 %
Messages
Commands completed successfully.

Completion time: 2024-02-02T19:43:20.0307016+05:30
```

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

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

108 %

Messages

```
Commands completed successfully.

Completion time: 2024-02-02T19:45:19.0628691+05:30
```

3. Fetch the data where the Customer's name is ending with 'N' also get the purchase amount value greater than 500.
   - Got empty result as no record have a Customer's name ending with 'N'

```
select CustomerName, PurchaseAmount from Customer where CustomerName like '%N';
```

108 %

Results Messages

CustomerName	PurchaseAmount

   - got the purchase amount values greater than 500.

```
select CustomerName, PurchaseAmount from Customer where PurchaseAmount>500;
```

108 %

Results Messages

	CustomerName	PurchaseAmount
1	Andrew	550
2	Remona	2700
3	Andrew	4000
4	Lucky	4500
5	Julia	4545

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.
    - Unique values from both tables

```
select SalesmanId from Salesman union select salesmanId from Customer;
```

108 %

Results | Messages

	SalesmanId
1	101
2	102
3	103
4	104
5	105
6	107
7	110

    - with duplicates from two table

```
select SalesmanId from Salesman union all select salesmanId from Customer;
```

108 %

Results | Messages

	SalesmanId
1	101
2	102
3	103
4	104
5	105
6	101
7	107
8	104
9	103
10	110

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.

```
select
orderdate, Name, CustomerName, Commission, City
from
salesman as s
inner join
customer as c
on s.salesmanId=c.salesmanid
inner join Orders
on s.salesmanid=Orders.salesmanid
where PurchaseAmount between 500 and 1500;
```

108 %

Results | Messages

	orderdate	Name	CustomerName	Commission	City
1	2021-07-01	Joe	Andrew	50.00	California

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

```
select *
from Orders
right join
Salesman
on Salesman.SalesmanId=Orders.SalesmanId;
```

108 %

Results | Messages

	OrderId	CustomerId	SalesmanId	Orderdate	Amount	SalesmanId	Name	Commission	City	Age
1	5001	2345	101	2021-07-01	550.00	101	Joe	50.00	California	17
2	NULL	NULL	NULL	NULL	NULL	102	Simon	75.00	Texas	25
3	NULL	NULL	NULL	NULL	NULL	103	Jessie	105.00	Florida	35
4	NULL	NULL	NULL	NULL	NULL	104	Danny	100.00	Texas	22
5	5003	1234	105	2022-02-15	1500.00	105	Lia	65.00	New Jersey	30