# SQL-Mandatory-Assignment-1

### Salesman table creation

- Create Database Intellipaat_Assignments
Use Intellipaat_Assignments

- 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);

- Select * from Salesman

Results Messages

| | SalesmanId | Name | Commission | City | Age |
|---|---|---|---|---|---|
| 1 | 101 | Joe | 50.00 | California | 17 |
| 2 | 102 | Simon | 75.00 | Texas | 25 |
| 3 | 103 | Jessie | 105.00 | Florida | 35 |
| 4 | 104 | Danny | 100.00 | Texas | 22 |
| 5 | 105 | Lia | 65.00 | New Jersey | 30 |

## 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);

- Select * from Customer

Results | Messages

| | SalesmanId | CustomerId | CustomerName | PurchaseAmount |
|---|---|---|---|---|
| 1 | 101 | 2345 | Andrew | 550 |
| 2 | 103 | 1575 | Lucky | 4500 |
| 3 | 104 | 2345 | Andrew | 4000 |
| 4 | 107 | 3747 | Remona | 2700 |
| 5 | 110 | 4004 | Julia | 4545 |

## 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

Results | Messages

| | OrderId | CustomerId | SalesmanId | Orderdate | Amount |
|---|---|---|---|---|---|
| 1 | 5001 | 2345 | 101 | 2021-07-01 | 550.00 |
| 2 | 5003 | 1234 | 105 | 2022-02-15 | 1500.00 |

# Tasks to be Performed:

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

```
Insert Into Orders Values(5004,4004,104,'2022-02-19',3000)
```

| | OrderId | CustomerId | SalesmanId | Orderdate | Amount |
|---|---|---|---|---|---|
| 1 | 5001 | 2345 | 101 | 2021-07-01 | 550.00 |
| 2 | 5003 | 1234 | 105 | 2022-02-15 | 1500.00 |
| 3 | 5004 | 4004 | 104 | 2022-02-19 | 3000.00 |

### 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_SalesmanId Primary Key(SalesmanId)
```

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

```
Begin Transaction
/*Filtering and Deleting Not Matching Records From SalesmanId Column From the Customer
With Respective to SalesmanId Column From the Salesman Table */

Select * From Customer Where SalesmanId not in (Select SalesmanId from Salesman)

Delete From Customer Where SalesmanId not in (Select SalesmanId From Salesman)

Alter Table Customer Add Constraint fk_SalesmanId Foreign Key(SalesmanId)
References Salesman(SalesmanId)


Rollback
```

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

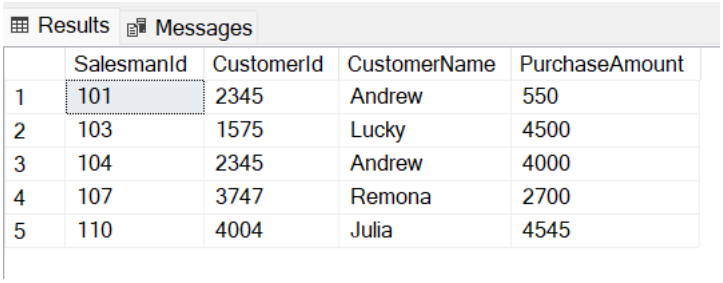
```
Alter Table Customer Alter Column CustomerName varchar(30) Not Null
```

### 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 Where CustomerName Like '%N'
  ```

  Ans: Found No Matching Records
  ```

- Select * From Customer Where PurchaseAmount > 500

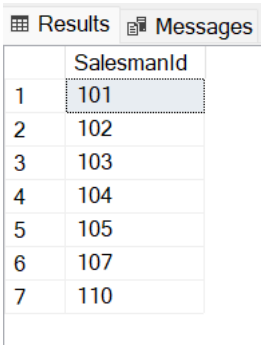


Results | Messages

| | SalesmanId | CustomerId | CustomerName | PurchaseAmount |
|---|---|---|---|---|
| 1 | 101 | 2345 | Andrew | 550 |
| 2 | 103 | 1575 | Lucky | 4500 |
| 3 | 104 | 2345 | Andrew | 4000 |
| 4 | 107 | 3747 | Remona | 2700 |
| 5 | 110 | 4004 | 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.**

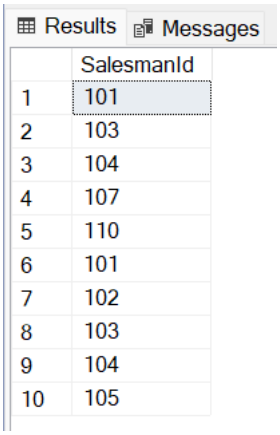
- Select Distinct(SalesmanId) From Customer
  Union
  Select Distinct(SalesmanId) From Salesman



Results | Messages

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

- Select Distinct(SalesmanId) From Customer
  Union All
  Select Distinct(SalesmanId) From Salesman



Results | Messages

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

**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 O.Orderdate,S.Name,C.CustomerName,S.Commission,S.City
  From Salesman As S
  Join Customer as C
  On S.SalesmanId = C.SalesmanId
  Join Orders as O
  On O.CustomerId = C.CustomerId
  Where C.PurchaseAmount Between 500 And 1500
  ```

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 Salesman As S
  Right Join Orders As O
  On S.SalesmanId = O.SalesmanId
  ```

| | SalesmanId | Name | Commission | City | Age | OrderId | CustomerId | SalesmanId | Orderdate | Amount |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 101 | Joe | 50.00 | California | 17 | 5001 | 2345 | 101 | 2021-07-01 | 550.00 |
| 2 | 105 | Lia | 65.00 | New Jersey | 30 | 5003 | 1234 | 105 | 2022-02-15 | 1500.00 |
| 3 | 104 | Danny | 100.00 | Texas | 22 | 5004 | 4004 | 104 | 2022-02-19 | 3000.00 |