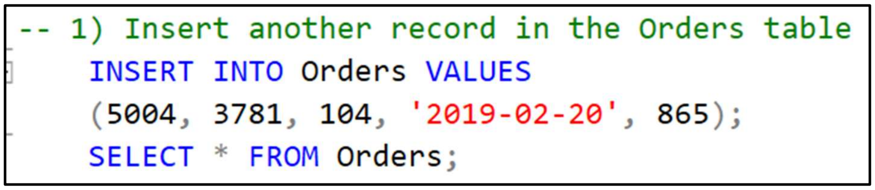
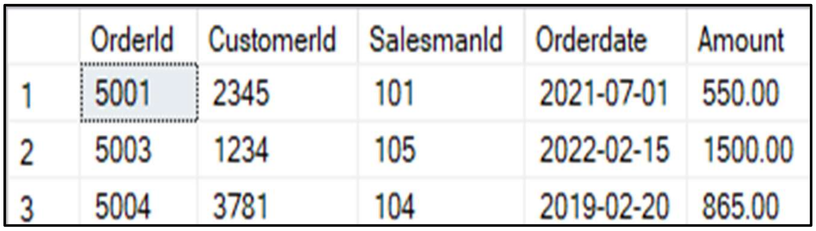
ANIKET CHAKRABORTY MSSQL ASSIGNMENT - 1 ANSWER COPY

Task – 1 MSSQL Query



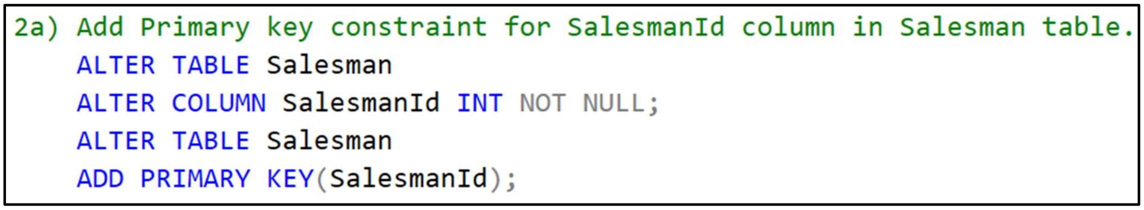
```
-- 1) Insert another record in the Orders table
    INSERT INTO Orders VALUES
    (5004, 3781, 104, '2019-02-20', 865);
    SELECT * FROM Orders;
```

Task – 1 Question Answer

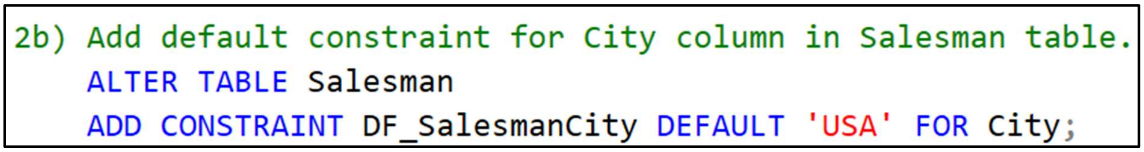


	OrderId	CustomerId	SalesmanId	Orderdate	Amount
1	5001	2345	101	2021-07-01	550.00
2	5003	1234	105	2022-02-15	1500.00
3	5004	3781	104	2019-02-20	865.00

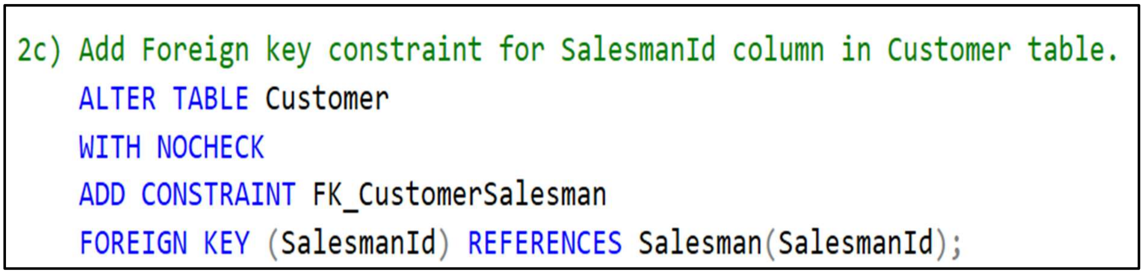
Task – 2 MSSQL Query



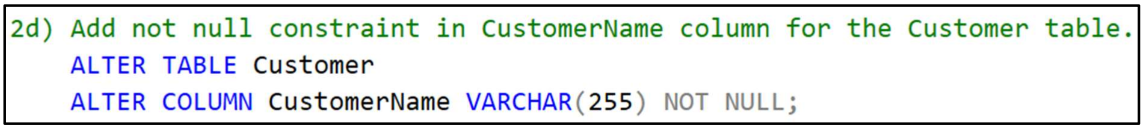
```
2a) Add Primary key constraint for SalesmanId column in Salesman table.
    ALTER TABLE Salesman
    ALTER COLUMN SalesmanId INT NOT NULL;
    ALTER TABLE Salesman
    ADD PRIMARY KEY(SalesmanId);
```



```
2b) Add default constraint for City column in Salesman table.
    ALTER TABLE Salesman
    ADD CONSTRAINT DF_SalesmanCity DEFAULT 'USA' FOR City;
```

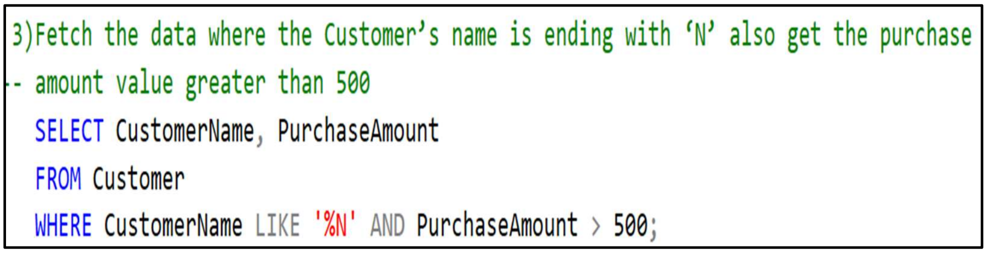


```
2c) Add Foreign key constraint for SalesmanId column in Customer table.
    ALTER TABLE Customer
    WITH NOCHECK
    ADD CONSTRAINT FK_CustomerSalesman
    FOREIGN KEY (SalesmanId) REFERENCES Salesman(SalesmanId);
```



```
2d) Add not null constraint in CustomerName column for the Customer table.
    ALTER TABLE Customer
    ALTER COLUMN CustomerName VARCHAR(255) NOT NULL;
```

Task – 3 MSSQL Query

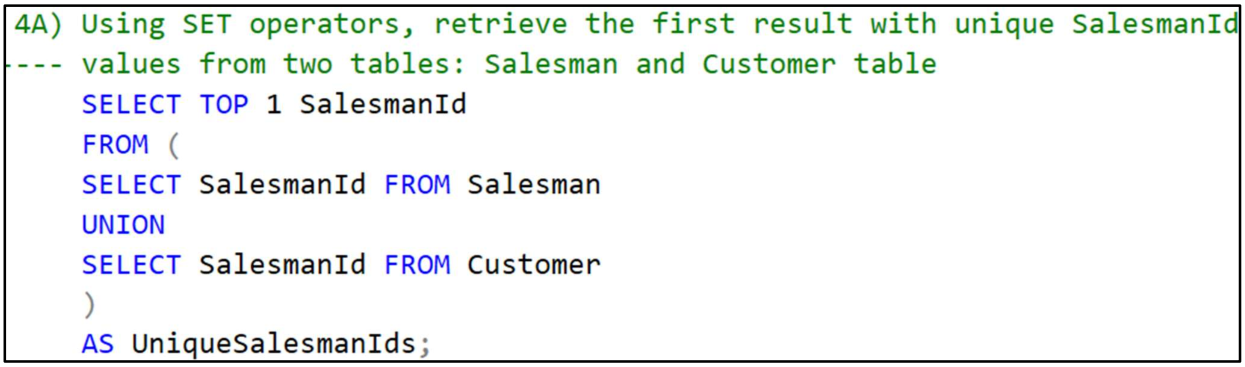


```
3)Fetch the data where the Customer’s name is ending with ‘N’ also get the purchase
- amount value greater than 500
  SELECT CustomerName, PurchaseAmount
  FROM Customer
  WHERE CustomerName LIKE '%N' AND PurchaseAmount > 500;
```

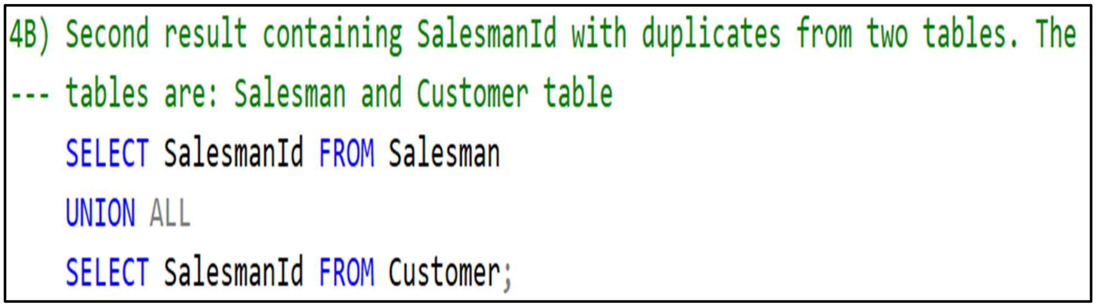
Task – 3 Question Answer

This returns an empty table having two columns ‘CustomerName’ & ‘PurchaseAmount’.

Task – 4 MSSQL Query



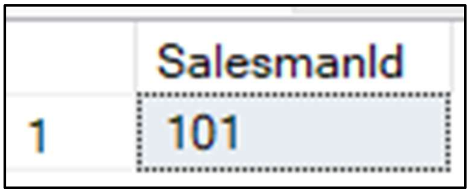
```
4A) Using SET operators, retrieve the first result with unique SalesmanId
--- values from two tables: Salesman and Customer table
    SELECT TOP 1 SalesmanId
    FROM (
    SELECT SalesmanId FROM Salesman
    UNION
    SELECT SalesmanId FROM Customer
    )
    AS UniqueSalesmanIds;
```



```
4B) Second result containing SalesmanId with duplicates from two tables. The
--- tables are: Salesman and Customer table
    SELECT SalesmanId FROM Salesman
    UNION ALL
    SELECT SalesmanId FROM Customer;
```

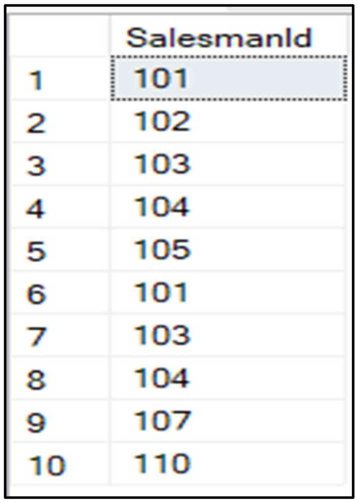
Task – 4 Question Answer

Answer for Query 4A)



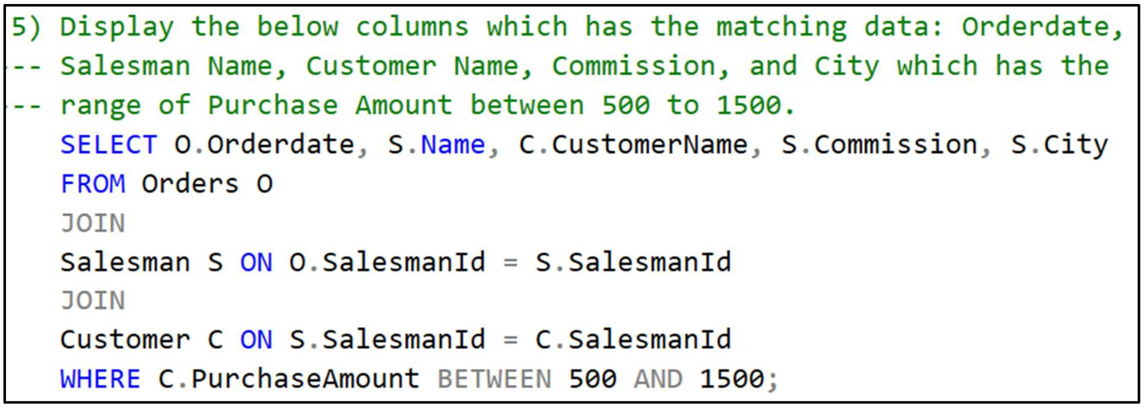
	SalesmanId
1	101

Answer for Query 4B)



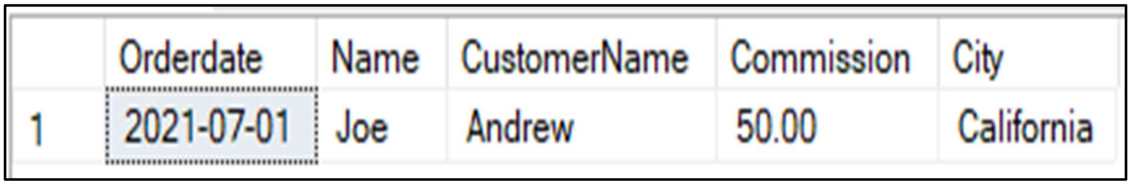
	SalesmanId
1	101
2	102
3	103
4	104
5	105
6	101
7	103
8	104
9	107
10	110

Task – 5 MSSQL Query



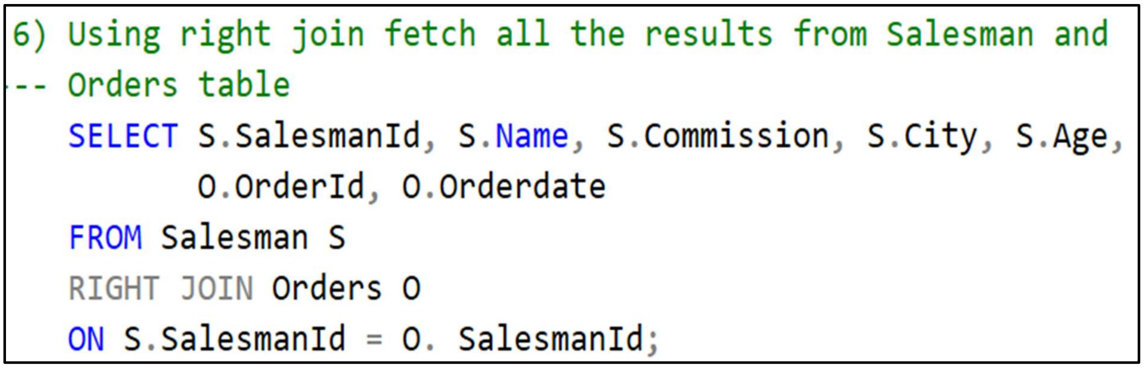
```
5) Display the below columns which has the matching data: Orderdate,
-- Salesman Name, Customer Name, Commission, and City which has the
-- range of Purchase Amount between 500 to 1500.
   SELECT O.Orderdate, S.Name, C.CustomerName, S.Commission, S.City
   FROM Orders O
   JOIN
   Salesman S ON O.SalesmanId = S.SalesmanId
   JOIN
   Customer C ON S.SalesmanId = C.SalesmanId
   WHERE C.PurchaseAmount BETWEEN 500 AND 1500;
```

Task – 5 Question Answer



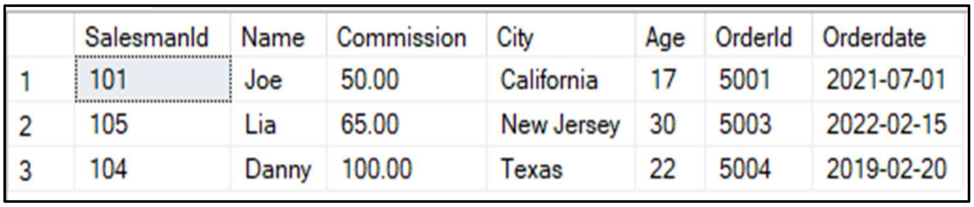
	Orderdate	Name	CustomerName	Commission	City
1	2021-07-01	Joe	Andrew	50.00	California

Task – 6 MSSQL Query



```
6) Using right join fetch all the results from Salesman and
-- Orders table
   SELECT S.SalesmanId, S.Name, S.Commission, S.City, S.Age,
          O.OrderId, O.Orderdate
   FROM Salesman S
   RIGHT JOIN Orders O
   ON S.SalesmanId = O. SalesmanId;
```

Task – 6 Question Answer



	SalesmanId	Name	Commission	City	Age	OrderId	Orderdate
1	101	Joe	50.00	California	17	5001	2021-07-01
2	105	Lia	65.00	New Jersey	30	5003	2022-02-15
3	104	Danny	100.00	Texas	22	5004	2019-02-20

Note:

Since, according to the question 1 of assignment 1, one new record having ‘SalesmanId’ equal to 104 is added in the orders table. So, while using the orders table in the last question, the last row appears in result 6. Otherwise, there will be 2 rows having ‘SalesmanId’ with 101 and 105.