

## SCHOOL OF INFORMATION COMPUTER SCIENCE AND ENGINEERING

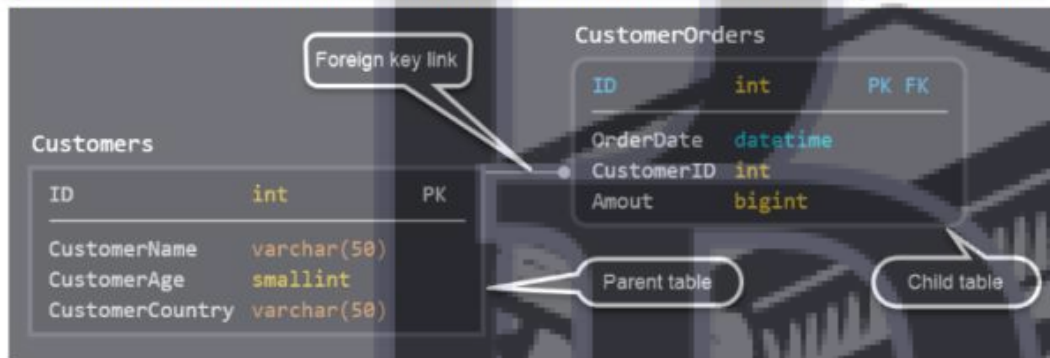
LAB: CYCLE SHEET – I(SQL) - FALL SEMESTER 2023-2024

Programme Name & Branch: B.Tech

Course Name: Database Systems LAB

Course Code: BCSE302L

The Customer and their order information about the Sales Data is given as below



- Create tables** identifying the primary keys and foreign keys, Insert necessary tuples into the tables. (Min 5 rows)

PAJAMA PADHAI

```
SQL> CREATE TABLE Customers (  
  2   ID NUMBER PRIMARY KEY,  
  3   CustomerName VARCHAR2(50),  
  4   CustomerAge NUMBER,  
  5   CustomerCountry VARCHAR2(50)  
  6 );
```

Table created.

```
SQL> CREATE TABLE CustomerOrders (  
  2   ID NUMBER PRIMARY KEY,  
  3   OrderDate DATE,  
  4   CustomerID NUMBER,  
  5   Amount NUMBER,  
  6   CONSTRAINT fk_CustomerID FOREIGN KEY (CustomerID) REFERENCES Customers(ID)  
  7 );
```

Table created.

SQL>

```
SQL> INSERT INTO Customers (ID, CustomerName, CustomerAge, CustomerCountry)  
  2 VALUES (1, 'John Doe', 30, 'USA');
```

1 row created.

```
SQL>  
SQL> INSERT INTO Customers (ID, CustomerName, CustomerAge, CustomerCountry)  
  2 VALUES (2, 'Jane Smith', 25, 'Canada');
```

1 row created.

```
SQL>  
SQL> INSERT INTO Customers (ID, CustomerName, CustomerAge, CustomerCountry)  
  2 VALUES (3, 'Michael Johnson', 35, 'UK');
```

1 row created.

```
SQL>  
SQL> INSERT INTO Customers (ID, CustomerName, CustomerAge, CustomerCountry)  
  2 VALUES (4, 'Emily Davis', 28, 'Australia');
```

1 row created.

```
SQL>  
SQL> INSERT INTO Customers (ID, CustomerName, CustomerAge, CustomerCountry)  
  2 VALUES (5, 'David Lee', 40, 'Singapore');
```

1 row created.

```
SQL> INSERT INTO CustomerOrders (ID, OrderDate, CustomerID, Amount)
  2 VALUES (1, TO_DATE('2023-05-20', 'YYYY-MM-DD'), 1, 100);
```

1 row created.

SQL>

```
SQL> INSERT INTO CustomerOrders (ID, OrderDate, CustomerID, Amount)
  2 VALUES (2, TO_DATE('2023-05-19', 'YYYY-MM-DD'), 3, 200);
```

1 row created.

SQL>

```
SQL> INSERT INTO CustomerOrders (ID, OrderDate, CustomerID, Amount)
  2 VALUES (3, TO_DATE('2023-05-18', 'YYYY-MM-DD'), 2, 150);
```

1 row created.

SQL>

```
SQL> INSERT INTO CustomerOrders (ID, OrderDate, CustomerID, Amount)
  2 VALUES (4, TO_DATE('2023-05-17', 'YYYY-MM-DD'), 4, 300);
```

1 row created.

SQL>

```
SQL> INSERT INTO CustomerOrders (ID, OrderDate, CustomerID, Amount)
  2 VALUES (5, TO_DATE('2023-05-16', 'YYYY-MM-DD'), 5, 250);
```

1 row created.

```
SQL> set linesize 500;
```

```
SQL> SELECT * FROM Customers;
```

ID	CUSTOMERNAME	CUSTOMERAGE	CUSTOMERCOUNTRY
1	John Doe	30	USA
2	Jane Smith	25	Canada
3	Michael Johnson	35	UK
4	Emily Davis	28	Australia
5	David Lee	40	Singapore

```
SQL> SELECT * FROM CustomerOrders;
```

ID	ORDERDATE	CUSTOMERID	AMOUNT
1	20-MAY-23	1	100
2	19-MAY-23	3	200
3	18-MAY-23	2	150
4	17-MAY-23	4	300
5	16-MAY-23	5	250

## 2. Enforce the Constraints

- a). The year of Date of Order should be below 2023.
- b). Customer Age Should be <100 and >17.
- c). The Default value of country is 'India'
- d) Alter the table to add the mobile number of customer table and its length should be 10.
- e) Customer name should be a Null Value.
- f) Add a Column email should be unique and can be null
- g) Add a Column Gender which could be any one of{'Male','Female','Trans'}
- h) Enforce foreign key constraints as per the tables given with ON CASE CADE' feature.
- i) Rename the column named as 'amout' into 'Amount'
- j) add a column 'GST' into order table such that it store the percentage of GST tax (0.15 for 15%). The default % of GST is 8%.

a)

```
SQL> ALTER TABLE CustomerOrders
  2  ADD CONSTRAINT CHK_OrderDateYear CHECK (EXTRACT(YEAR FROM ORDERDATE) < 2023);

Table altered.
```

b)

```
SQL> ALTER TABLE Customers
  2  ADD CONSTRAINT chk_CustomerAge CHECK (CustomerAge < 100 AND CustomerAge > 17);

Table altered.
```

c)

```
SQL> ALTER TABLE Customers
  2  MODIFY CustomerCountry DEFAULT 'India';

Table altered.
```

d)

```
SQL> ALTER TABLE Customers
  2  ADD MobileNumber VARCHAR2(10);

Table altered.
```

e)

```
SQL> ALTER TABLE Customers
  2  ADD CUSTOMERNAME_TEMP VARCHAR2(255);
```

Table altered.

```
SQL> UPDATE Customers
  2  SET CUSTOMERNAME_TEMP = CUSTOMERNAME;
```

5 rows updated.

```
SQL> ALTER TABLE Customers
  2  DROP COLUMN CUSTOMERNAME;
```

Table altered.

```
SQL> ALTER TABLE Customers
  2  RENAME COLUMN CUSTOMERNAME_TEMP TO CUSTOMERNAME;
```

Table altered.

f)

```
SQL> ALTER TABLE Customers
  2  ADD email VARCHAR(255) UNIQUE;
```

Table altered.

g)

```
SQL> ALTER TABLE Customers
  2  ADD Gender VARCHAR2(10) CHECK (Gender IN ('Male', 'Female', 'Trans'));
```

Table altered.

h)

```
SQL> ALTER TABLE CustomerOrders
  2  ADD CONSTRAINT fk_customer
  3  FOREIGN KEY (CUSTOMERID)
  4  REFERENCES Customers(ID)
  5  ON DELETE CASCADE;
FOREIGN KEY (CUSTOMERID)
*
ERROR at line 3:
ORA-02275: such a referential constraint already exists in the table
```

i)

```
SQL> ALTER TABLE CustomerOrders
  2  RENAME COLUMN amout TO Amount;
ALTER TABLE CustomerOrders
*
ERROR at line 1:
ORA-00957: duplicate column name
```

j)

```
SQL> ALTER TABLE CustomerOrders
  2  ADD GST NUMBER(4, 2) DEFAULT 8;

Table altered.
```

### 3. Write SQL

a) List the Customers whose name starts with 'R'

```
SQL> SELECT *
  2  FROM Customers
  3  WHERE CUSTOMERNAME LIKE 'R%';

no rows selected
```

b) List the Customers whose age is >20 and <30.



```
SQL> SELECT *
  2  FROM Customers
  3  WHERE CustomerAge > 20 AND CustomerAge < 30;
```

ID	CUSTOMERNAME	CUSTOMERAGE	CUSTOMERCOUNTRY
2	Jane Smith	25	Canada
4	Emily Davis	28	Australia

c) List the Foreign Customers and their email-ID

```
SQL> SELECT CustomerName
  2  FROM Customers
  3  WHERE CustomerCountry <> 'India';
```

CUSTOMERNAME

John Doe  
Jane Smith  
Michael Johnson  
Emily Davis  
David Lee

d) Test the string manipulation functions – UPPER, LOWER, INITCAP, LENGTH, LPAD, RPAD, LTRIM, RTRIM and TRIM and NVL using select queries on data present in the tables. Use one query each for demonstration of one function

UPPER

```
SQL> SELECT UPPER(CustomerName)
  2  FROM Customers;
```

UPPER(CUSTOMERNAME)

JOHN DOE  
JANE SMITH  
MICHAEL JOHNSON  
EMILY DAVIS  
DAVID LEE

LOWER

```
SQL> SELECT LOWER(CustomerName)
       2 FROM Customers;
```

```
LOWER(CUSTOMERNAME)
-----
```

```
john doe
jane smith
michael johnson
emily davis
david lee
```

INITCAP

```
SQL> SELECT INITCAP(CustomerName)
       2 FROM Customers;
```

```
INITCAP(CUSTOMERNAME)
-----
```

```
John Doe
Jane Smith
Michael Johnson
Emily Davis
David Lee
```

LENGTH

```
SQL> SELECT LENGTH(CustomerName)
       2 FROM Customers;
```

```
LENGTH(CUSTOMERNAME)
-----
```

```
8
10
15
11
9
```

LPAD



```
SQL> SELECT LPAD(CustomerName, 10, '*')
2 FROM Customers;
```

```
LPAD(CUSTOMERNAME,10,'*')
-----
```

```
**John Doe
Jane Smith
Michael Jo
Emily Davi
*David Lee
```

RPAD

```
SQL> SELECT RPAD(CustomerName, 10, '*')
2 FROM Customers;
```

```
RPAD(CUSTOMERNAME,10,'*')
-----
```

```
John Doe**
Jane Smith
Michael Jo
Emily Davi
David Lee*
```

LTRIM

```
SQL> SELECT LTRIM(CustomerName)
2 FROM Customers;
```

```
LTRIM(CUSTOMERNAME)
-----
```

```
John Doe
Jane Smith
Michael Johnson
Emily Davis
David Lee
```

RTRIM

```
SQL> SELECT RTRIM(CustomerName)
       2 FROM Customers;
```

```
RTRIM(CUSTOMERNAME)
-----
```

```
John Doe
Jane Smith
Michael Johnson
Emily Davis
David Lee
```

TRIM

```
SQL> SELECT TRIM(CustomerName)
       2 FROM Customers;
```

```
TRIM(CUSTOMERNAME)
-----
```

```
John Doe
Jane Smith
Michael Johnson
Emily Davis
David Lee
```

NVL

```
SQL> SELECT CustomerName, NVL(Email, 'N/A') AS Email
       2 FROM Customers;
```

```
CUSTOMERNAME
```

```
EMAIL
```

```
-----
John Doe      N/A
Jane Smith    N/A
Michael Johnson N/A
Emily Davis   N/A
David Lee     N/A
              john.doe@example.com
              jane.smith@example.com
              michael.johnson@example.com
              emily.davis@example.com
              david.lee@example.com
```

```
10 rows selected.
```

e) Sort the List of Customers alphabetically

```
SQL> SELECT *
  2  FROM Customers
  3  ORDER BY CustomerName ASC;
```

ID	CUSTOMERNAME	CUSTOMERAGE	CUSTOMERCOUNTRY
5	David Lee	40	Singapore
4	Emily Davis	28	Australia
2	Jane Smith	25	Canada
1	John Doe	30	USA
3	Michael Johnson	35	UK

f) List out the customer ID who have ordered the items on '03-may-2023'

```
SQL> SELECT Customers.ID AS CustomerID
  2  FROM Customers
  3  JOIN CustomerOrders ON Customers.ID = CustomerOrders.CustomerID
  4  WHERE CustomerOrders.OrderDate = TO_DATE('2023-05-03', 'YYYY-MM-DD');

no rows selected
```

g) List the customer Id who ordered the item where as its item cost >2000

```
SQL> SELECT Customers.ID AS CustomerID
  2  FROM Customers
  3  JOIN CustomerOrders ON Customers.ID = CustomerOrders.CustomerID
  4  WHERE CustomerOrders.Amount > 2000;

no rows selected
```

h) Calculate the GST amount based on the GST % for all items.

```
SQL> SELECT ID, Amount, (Amount * 0.15) AS GSTAmount
  2  FROM CustomerOrders;
```

ID	AMOUNT	GSTAMOUNT
1	100	15
2	200	30
3	150	22.5
4	300	45
5	250	37.5

i)Delete the orders with order-id 101

```
SQL> DELETE FROM CustomerOrders
2 WHERE ID = 101;
```

0 rows deleted.

j) Check the referential integrity during insertion / updation / Deletion of records of the table

Insertion:

```
SQL> INSERT INTO CustomerOrders (ID, OrderDate, CustomerID, Amount)
2 VALUES (101, DATE '2023-05-22', 3, 500);
```

1 row created.

```
SQL> select * from CustomerOrders;
```

ID	ORDERDATE	CUSTOMERID	AMOUNT
1	20-MAY-23	1	100
2	19-MAY-23	3	200
3	18-MAY-23	2	150
4	17-MAY-23	1	300
5	16-MAY-23	4	250
101	22-MAY-23	3	500

6 rows selected.

Updating:

```
SQL> UPDATE Customers
2 SET CustomerAge = 40
3 WHERE ID = 2;
```

1 row updated.

```
SQL> select * from Customers;
```

ID	CUSTOMERNAME	CUSTOMERAGE	CUSTOMERCOUNTRY
1	John Doe	30	USA
2	Jane Smith	40	Canada
3	Michael Johnson	35	UK
4	Emily Davis	28	Australia
5	David Lee	40	Singapore

Deletion:

```
SQL> DELETE FROM Customers
2 WHERE ID = 3;
```

1 row deleted.

```
SQL> select * from Customers;
```

ID	CUSTOMERNAME	CUSTOMERAGE	CUSTOMERCOUNTRY
1	John Doe	30	USA
2	Jane Smith	40	Canada
4	Emily Davis	28	Australia
5	David Lee	40	Singapore

k) Update the Customer's Country column by new value by replacing the old value

```
SQL> UPDATE Customers
2 SET CustomerCountry = 'India'
3 WHERE CustomerCountry = 'USA';
```

1 row updated.

```
SQL> select * from Customers;
```

ID	CUSTOMERNAME	CUSTOMERAGE	CUSTOMERCOUNTRY
1	John Doe	30	India
2	Jane Smith	40	Canada
4	Emily Davis	28	Australia
5	David Lee	40	Singapore

l) Count and Display the number of customers.

```
SQL> SELECT COUNT(*) AS TotalCustomers
2 FROM Customers;
```

TOTALCUSTOMERS
9

1 customer entry was deleted

m) Display the Names of customers grouped by country

```
SQL> SELECT CustomerCountry, LISTAGG(CustomerName, ', ') WITHIN GROUP (ORDER BY CustomerName) AS CustomerNames
2 FROM Customers
3 GROUP BY CustomerCountry;
```

CUSTOMERCOUNTRY

CUSTOMERNAMES

Australia  
Emily Davis

Canada  
Jane Smith

India  
John Doe

CUSTOMERCOUNTRY

CUSTOMERNAMES

Singapore  
David Lee

n) Count the No of customers from each country in descending order of the count

```
SQL> SELECT CustomerCountry, COUNT(*) AS CustomerCount
2 FROM Customers
3 GROUP BY CustomerCountry
4 ORDER BY CustomerCount DESC;
```

CUSTOMERCOUNTRY

CUSTOMERCOUNT

Canada  
Singapore  
India  
Australia

5  
1  
1  
1  
1

o) Count the number of customers from 'India'



```
SQL> SELECT COUNT(*) AS CustomerCount
2 FROM Customers
3 WHERE CustomerCountry = 'India';
```

```
CUSTOMERCOUNT
```

---

```
1
```

p) List out the count of customers in each country whereas the country should have minimum 5 customers

```
SQL> SELECT CustomerCountry, COUNT(*) AS CustomerCount
2 FROM Customers
3 GROUP BY CustomerCountry
4 HAVING COUNT(*) >= 5;
```

```
CUSTOMERCOUNTRY
```

---

```
CUSTOMERCOUNT
```

---

```
5
```

q) List out the sum , avg, min max of all orders ordered

```
SQL> SELECT
2 SUM(Amount) AS TotalAmount,
3 AVG(Amount) AS AverageAmount,
4 MIN(Amount) AS MinimumAmount,
5 MAX(Amount) AS MaximumAmount
6 FROM CustomerOrders;
```

```
TOTALAMOUNT AVERAGEAMOUNT MINIMUMAMOUNT MAXIMUMAMOUNT
```

---

```
800
```

```
200
```

```
100
```

```
300
```

r) Find out the Total Amount of all orders for each month of the year 2023

PAJAMA PADHAI

```
SQL> SELECT EXTRACT(MONTH FROM OrderDate) AS Month,
2          SUM(Amount) AS TotalAmount
3 FROM CustomerOrders
4 WHERE EXTRACT(YEAR FROM OrderDate) = 2023
5 GROUP BY EXTRACT(MONTH FROM OrderDate);
```

MONTH	TOTALAMOUNT
5	800

s) .Find out the average of sales of orders for each customer.

```
SQL> SELECT CustomerID, AVG(Amount) AS AverageSales
2 FROM CustomerOrders
3 GROUP BY CustomerID;
```

CUSTOMERID	AVERAGESALES
1	200
2	150
4	250

t) Apply the multiple functions into a group by clause using any one of the table

```
SQL> SELECT CustomerID,
2          COUNT(*) AS TotalOrders,
3          SUM(Amount) AS TotalAmount,
4          AVG(Amount) AS AverageAmount,
5          MIN(Amount) AS MinimumAmount,
6          MAX(Amount) AS MaximumAmount
7 FROM CustomerOrders
8 GROUP BY CustomerID;
```

CUSTOMERID	TOTALORDERS	TOTALAMOUNT	AVERAGEAMOUNT	MINIMUMAMOUNT	MAXIMUMAMOUNT
1	2	400	200	100	300
2	1	150	150	150	150
4	1	250	250	250	250

u) Apply the Nesting Aggregate Functions

```
SQL> SELECT AVG(MaxOrderAmount) AS AverageMaxOrderAmount
2 FROM (
3     SELECT CustomerID, MAX(Amount) AS MaxOrderAmount
4     FROM CustomerOrders
5     GROUP BY CustomerID
6 ) Subquery;
```

```
AVERAGEMAXORDERAMOUNT
-----
                233.333333
```

v) List out the customer id who have placed the order for more than one time

```
SQL> SELECT CustomerID
2 FROM CustomerOrders
3 GROUP BY CustomerID
4 HAVING COUNT(*) > 1;
```

```
CUSTOMERID
-----
          1
```

w) Display the order-id sorted based on its amount

```
SQL> SELECT ID
2 FROM CustomerOrders
3 ORDER BY Amount ASC;
```

```
ID
-----
  1
  3
  5
  4
```

x) Find out the months difference between two orders by customer id 102

```
SQL> SELECT (MONTHS_BETWEEN(MAX(OrderDate), MIN(OrderDate))) AS MonthsDifference
2 FROM CustomerOrders
3 WHERE CustomerID = 102;
```

```
MONTHSDIFFERENCE
-----
```

y) Display the Order date of each order in terms of its sentences

```
SQL> SELECT ID, SUBSTR(OrderDate, 1, INSTR(OrderDate, ' ', 1) - 1) AS Sentence  
2 FROM CustomerOrders;
```

ID	SENTENCE
----	----------

1	
3	
4	
5	

z) Display the order date with YYYY-MM-DD HH24:MI:SS format

```
SQL> SELECT ID, TO_CHAR(OrderDate, 'YYYY-MM-DD HH24:MI:SS') AS FormattedOrderDate  
2 FROM CustomerOrders;
```

ID	FORMATTEDORDERDATE
----	--------------------

1	2023-05-20 00:00:00
3	2023-05-18 00:00:00
4	2023-05-17 00:00:00
5	2023-05-16 00:00:00

PAJAMA PADHAI