1-5 SQL Quires & Scripts

Q1. Given a Teams table with columns TeamID (integer) and Members (comma-separated string of names), write a query to calculate and display the total number of members in each team.

```
CREATE TABLE Teams (
   TeamID INT PRIMARY KEY,
   Members TEXT
);

-- Insert Data
INSERT INTO Teams (TeamID, Members) VALUES
(1, 'Chris, Evan, Marty, Eva'),
(2, 'Jake, Olivia'),
(3, 'Sophia, Liam, Noah, Emma'),
(4, 'Ava, Lucas, Mia, Ethan, Amelia'),
(5, 'Benjamin, Charlotte'),
(6, 'Harper, Henry, Evelyn, Daniel, Ella'),
(7, 'Michael, Emily, Alexander'),
(8, 'James, Abigail, William, Isabella, Jack, Grace'),
(9, 'Sebastian, Chloe'),
(10, 'David, Lily, Samuel, Madison');

select * from Teams
```

Method: 1

```
select
TeamID,
Members,
len (Members) -len (REPLACE(Members,',','')) + 1 as Total_Members
from Teams
```

Method: 2

```
select
TeamID,
Members,
count (*) as Total_Members
from Teams
cross apply string_split (Members,',')
group by TeamID, Members
```

Q2. Write a SQL query to retrieve the most frequently ordered item(s) for each date from a given orders table. If multiple items have the highest order count on a particular date, include all such items in the result.

```
-- Creating the Table
CREATE TABLE orders (
   order_date DATE,
   item VARCHAR(255)
);

-- Inserting Data
INSERT INTO orders (order_date, item) VALUES
('2024-03-01', 'Apple'),
('2024-03-01', 'Banana'),
('2024-03-01', 'Apple'),
('2024-03-02', 'Orange'),
('2024-03-02', 'Orange'),
('2024-03-02', 'Mango'),
('2024-03-03', 'Banana'),
('2024-03-03', 'Banana'),
('2024-03-03', 'Mango'),
('2024-03-03', 'Mango');
```

```
select * from orders

with cte as (
select
order_date, item,
COUNT (*) as Item_Count,
RANK () over (partition by order_date order by count (*) desc) as Ranked_item
from orders
group by order_date, item)
select * from cte
where Ranked_item = 1
```

Q3. Return number of survival and not survival count of passengers for each class.

```
create table passengers (
    passengerid int primary key,
    survived tinyint,
    pclass int,
    name varchar(255),
    sex varchar(10),
    age float,
    sibsp int,
    parch int,
    ticket varchar(50),
    fare float, f
    cabin varchar (50),
    embarked char(1)
);

--Insert statement:
-----------------------------

insert into passengers values
(1, 0, 3, 'Reddy, Mr. Rajesh Kumar', 'male', 28, 0, 0, 'IND/101', 15.75, NULL, 'S'),
(2, 1, 2, 'Sharma, Mrs. Priya Devi', 'female', 32, 1, 1, 'IND/102', 22.50, 'B52', 'C'),
(3, 0, 1, 'Singh, Mr. Arjun Pratap', 'male', 45, 0, 0, 'IND/103', 76.25, 'A19', 'S'),
(4, 1, 3, 'Patel, Miss. Nisha', 'female', 24, 0, 0, 'IND/104', 12.75, NULL, 'Q'),
(5, 0, 2, 'Nair, Mr. Suresh', 'male', 37, 1, 0, 'IND/105', 19.50, 'C41', 'S'),
(6, 1, 1, 'Gupta, Mrs. Anjali Ramesh', 'female', 40, 1, 2, 'IND/106', 82.10, 'D10', 'C'),
(7, 0, 3, 'Khan, Mr. Imran Ali', 'male', 30, 0, 0, 'IND/107', 7.50, NULL, 'S'),
(8, 1, 2, 'Chopra, Miss. Aarti', 'female', 26, 0, 0, 'IND/108', 16.85, NULL, 'Q'),
(9, 0, 3, 'Joshi, Mr. Devendra', 'male', 29, 0, 0, 'IND/109', 8.60, NULL, 'S'),
(10, 1, 1, 'Das, Dr. Sunita Reddy', 'female', 50, 1, 0, 'IND/110', 90.30, 'B18', 'C'),
(11, 0, 2, 'Verma, Mr. Pradeep Kumar', 'male', 38, 0, 0, 'IND/111', 20.25, 'D22', 'S'),
(12, 1, 3, 'Iyer, Miss. Radhika', 'female', 21, 0, 0, 'IND/112', 7.75, NULL, 'Q'),
(13, 0, 1, 'Bose, Mr. Anirudh', 'male', 42, 1, 0, 'IND/113', 67.50, 'A33', 'S'),
(14, 1, 2, 'Sinha, Mrs. Meera Dev', 'female', 34, 1, 1, 'IND/114', 29.40, 'C27', 'C'),
(15, 0, 3, 'Mishra, Mr. Sandeep', 'male', 31, 0, 0, 'IND/115', 8.25, NULL, 'S'),
(16, 1, 1, 'Bhatia, Mrs. Kavita Anil', 'female', 39, 0, 0, 'IND/116', 75.00, 'B35', 'C'),
(17, 0, 2, 'Kapoor, Mr. Aditya', 'male', 27, 0, 0, 'IND/117', 18.80, NULL, 'S'),
(18, 1, 3, 'Pandey, Miss. Sneha', 'female', 25, 0, 0, 'IND/118', 7.85, NULL, 'Q'),
(19, 0, 2, 'Rao, Mr. Ankit Ramesh', 'male', 33, 0, 0, 'IND/119', 21.15, 'C12', 'S'),
(20, 1, 1, 'Malhotra, Mrs. Neeta Rakesh', 'female', 36, 1, 1, 'IND/120', 65.90, 'A21', 'C')

select * from passengers

select
pclass,
COUNT (case when survived = 1 then passengerid else null end) as sur_cnt,
COUNT (case when survived = 0 then passengerid else null end) as nonsur_cnt
from passengers
group by pclass
```

Q4. Return Customer IDs Who did the highest transaction amount difference from their first and last transaction amounts?

```
create table transactions (
 transaction_id int,
    customer_id int,
```

```
    transaction_amount float,
    transaction_date date
)
insert into transactions values
(100, 1, 8000, '2024-09-01'),
(101, 2, 25000, '2024-09-02'),
(102, 3, 10000, '2024-09-02'),
(103, 4, 3000, '2024-09-05'),
(104, 1, 9000, '2024-09-06'),
(105, 2, 22000, '2024-09-07'),
(106, 3, 12000, '2024-09-07'),
(107, 4, 4000, '2024-09-07'),
(108, 1, 15000, '2024-09-10'),
(109, 2, 27000, '2024-09-11'),
(110, 3, 5500, '2024-09-11'),
(111, 1, 14000, '2024-09-13'),
(112, 4, 7500, '2024-09-16'),
(113, 3, 4500, '2024-09-16'),
(114, 2, 24000, '2024-09-16'),
(115, 1, 11000, '2024-09-16'),
(116, 4, 9500, '2024-09-17'),
(117, 2, 18000, '2024-09-20'),
(118, 3, 6000, '2024-09-21'),
(119, 2, 25000, '2024-09-02')

select * from transactions
with cte as (
select *,
FIRST_VALUE (transaction_amount) over (partition by customer_id order by transaction_date) as FV,
LAST_VALUE (transaction_amount) over (partition by customer_id order by transaction_date
                                      rows between unbounded preceding and unbounded following) as LV
from transactions
),
cte2 as (
select distinct customer_id, FV, LV
from cte),
cte3 as (
select customer_id, ABS(FV-LV) as Diff_Amt,
       DENSE_RANK () over (order by ABS(FV-LV) desc) as rn
from cte2
)
select customer_id
from cte3
where rn = 1
```

Method 2

```
WITH FirstLast AS (
    SELECT
        t1.customer_id,
        t1.transaction_amount AS FV,
        t2.transaction_amount AS LV
    FROM
        (SELECT customer_id, transaction_amount
         FROM transactions
         WHERE transaction_date = (SELECT MIN (transaction_date)
                                   FROM transactions t
                                   WHERE t.customer_id = transactions.customer_id)) t1
    JOIN
        (SELECT customer_id, transaction_amount
         FROM transactions
         WHERE transaction_date = (SELECT MAX (transaction_date)
                                   FROM transactions t
                                   WHERE t.customer_id = transactions.customer_id)) t2
    ON t1.customer_id = t2.customer_id
)
SELECT distinct customer_id
FROM FirstLast
WHERE ABS (FV - LV) = (SELECT MAX (ABS (FV - LV)) FROM FirstLast
```

Q5. Find the duplicate Records and Remove them.

```
create table employees_details (
   emp_id int,
   emp_name varchar (50),
   mgr_id int,
   salary int
);
insert into employees_details (emp_id, emp_name, mgr_id, salary) values
(1, 'yogesh', 2, 32000),
(2, 'ravi', 3, 50000),
(3, 'harsha', 7, 63000),
(4, 'yogesh', 1, 28000),
(5, 'harsha', 3, 75000),
(1, 'yogesh', 2, 32000),
(7, 'basha', 1, 80000),
(8, 'ravi', 3, 50000)

select * from employees_details

with cte as (
select
emp_id,
emp_name,
ROW_NUMBER () over (partition by emp_id, emp_name order by (select null)) as rn
from employees_details)
Delete from cte
where rn>1
```