Ex. No: 04	JOINS AND SET OPERATIONS
Date	06-02-24

Objective:

To execute the given queries using set operators and joins.

Description:

Set Operators

The three *set operators* union, intersect and minus allow to serially combine more than one select statements. Although more than one select statement will then be present, only *one* result set is then returned. The following list briefly describes the three set operations supported by Oracle SQL:

1) UNION

union all is very similar to union, however, it dismisses duplicate rows found across different select statements:

```
select col_1, col_2, col_3 from table_1 union
select col_1, col_2, col_3 from table_2;
```

2) INTERSECT

intersect only returns the rows that are found in all select statements:

```
select col_1, col_2, col_3 from table_1 intersect
select col_1, col_2, col_3 from table_2;
```

3) MINUS

minus returns all rows from the first select statements except those who are duplicated in a following select statement:

```
select col_1, col_2, col_3 from table_1 minus
select col_1, col_2, col_3 from table_2;
```

SQL JOIN

The JOIN keyword is used in an SQL statement to query data from two or more tables, based on a relationship between certain columns in these tables.

Tables in a database are often related to each other with keys.

Different SQL JOINs

Before we continue with examples, we will list the types of JOIN you can use, and the differences between them.

JOIN: Return rows when there is at least one match in both tables

LEFT JOIN: Return all rows from the left table, even if there are no matches in the right table

RIGHT JOIN: Return all rows from the right table, even if there are no matches in the left table

FULL JOIN: Return rows when there is a match in one of the tables

SQL INNER JOIN Keyword

The INNER JOIN keyword return rows when there is at least one match in both tables.

Syntax

```
SELEC column_name(s)
FROM table_name1
INNER JOIN table_name2
ON table_name1.column_name=table_name2.column_name
```

PS: INNER JOIN is the same as JOIN.

SQL LEFT JOIN Keyword

The LEFT JOIN keyword returns all rows from the left table (table_name1), even if there are no matches in the right table (table_name2).

Syntax

```
SELECT column_name(s)
FROM table_name1
LEFT OUTER JOIN table_name2
ON table_name1.column_name=table_name2.column_name
```

SQL RIGHT JOIN Keyword

The RIGHT JOIN keyword Return all rows from the right table (table_name2), even if there are no matches in the left table (table_name1).

Syntax

```
SELECT column_name(s)
FROM table_name1
RIGHT OUTER JOIN table_name2
ON table_name1.column_name=table_name2.column_name
```

SQL FULL JOIN Keyword

The FULL JOIN keyword return rows when there is a match in one of the tables.

Syntax

```
SELECT column_name(s)
FROM table_name1
```

FULL OUTER JOIN table_name2

ON table_name1.column_name=table_name2.column_name

The JOIN keyword is used in an SQL statement to query data from two or more tables, based on a relationship between certain columns in these tables. Whenever a query is written which refers more than one table that needs the help of joins.

Questions:

1. Retrieve the names of users who have registered for the "Concert in Park" event:

```
SQL> SELECT u.Name
  2  FROM User_URK22AI1048 u
  3  JOIN Ticket_URK22AI1048 t ON u.UserID = t.UserID
  4  JOIN Event_URK22AI1048 e ON t.EventID = e.EventID
  5  WHERE e.Name = 'Concert in Park';

NAME
--------------------------------------------------------------------------------
John Smith
Jane Doe
```

2. Find the details of events (name, date, and time) that Sarah Adams has registered for.

```
SQL> SELECT e.Name, e.Description
  2  FROM Event_URK22AI1048 e
  3  LEFT JOIN Ticket_URK22AI1048 t ON e.EventID = t.EventID
  4  WHERE t.TicketID IS NULL;

NAME
--------------------------------------------------------------------------------
DESCRIPTION
--------------------------------------------------------------------------------
Food Festival
A celebration of diverse cuisines.
```

3. List the events (name and description) that do not have any registered participants.

```
SQL> SELECT e.Name, e.Description
  2  FROM Event_URK22AI1048 e
  3  LEFT JOIN Ticket_URK22AI1048 t ON e.EventID = t.EventID
  4  WHERE t.TicketID IS NULL;

NAME
--------------------------------------------------------------------------------
DESCRIPTION
--------------------------------------------------------------------------------
Food Festival
A celebration of diverse cuisines.
```

4. Retrieve the names of users and the events they have registered for, along with the event dates.

```
SQL>  SELECT u.Name, e.Name, e.EventDate
  2  FROM User_URK22AI1048 u
  3  JOIN Ticket_URK22AI1048 t ON u.UserID = t.UserID
  4  JOIN Event_URK22AI1048 e ON t.EventID = e.EventID;

NAME
                                                  NAME
                                                                                                    EVENTDATE
--------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------- ------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------- --------
Jane Doe
                                                  Concert in Park
                                                                                                    15-AUG-23
John Smith
                                                  Concert in Park
```

5. Find the names of users who have registered for events taking place on or after September 1, 2023.

```
SQL> SELECT u.Name
  2  FROM User_URK22AI1048 u
  3  JOIN Ticket_URK22AI1048 t ON u.UserID = t.UserID
  4  JOIN Event_URK22AI1048 e ON t.EventID = e.EventID
  5  WHERE e.EventDate >= TO_DATE('2023-09-01', 'YYYY-MM-DD');

NAME
----------------------------------------------------------------
----------------------------------------------------------
David Wang
Emily Chen
SQL>
```

6. Retrieve the names of users who have booked tickets for the "Movie Night" event.

```
SQL>  SELECT u.Name
  2  FROM User_URK22AI1048 u
  3  JOIN Ticket_URK22AI1048 t ON u.UserID = t.UserID
  4  JOIN Event_URK22AI1048 e ON t.EventID = e.EventID
  5  WHERE e.Name = 'Movie Night';

NAME
-------------------------------------------------------------
----------------------------------------------------------
Michael Lee
Sarah Adams
SQL>
```

7. List the event names, user names, and seat numbers for all booked tickets.

```
SQL> SELECT e.Name, u.Name, t.SeatNumber
  2  FROM User_URK22AI1048 u
  3  JOIN Ticket_URK22AI1048 t ON u.UserID = t.UserID
  4  JOIN Event_URK22AI1048 e ON t.EventID = e.EventID;

NAME
                                                  NAME
                                                                                                    SEATNUMBER
--------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------- ------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------- --------------------
Concert in Park
                                                  Jane Doe
                                                                                                    B2
```

8. Find the names of users who have not booked any tickets for any event.

```
SQL>  SELECT u.Name
  2  FROM User_URK22AI1048 u
  3  LEFT JOIN Ticket_URK22AI1048 t ON u.UserID = t.UserID
  4  WHERE t.TicketID IS NULL;
```

9. Perform Left Outer Join to Retrieve Event Details along with Venue Information.

```
SQL>  SELECT e.Name, e.EventDate, e.Time, v.Name AS VenueName, v.Address, v.City, v.State, v.Country
  2  FROM Event_URK22AI1048 e
  3  LEFT JOIN Venue_URK22AI1048 v ON e.VenueID = v.VenueID;

NAME
                                                                 EVENTDATE TIME
                                                                                                                                            VENUENAME
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
ADDRESS
                                                                 CITY
                                                                                                                                  STATE
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
```

10. Perform Full Outer Join to Retrieve Combined Event and Venue Information.

```
SQL> SELECT e.Name, e.EventDate, e.Time, v.Name AS VenueName, v.Address, v.City, v.State, v.Country
  2  FROM Event_URK22AI1048 e
  3  FULL OUTER JOIN Venue_URK22AI1048 v ON e.VenueID = v.VenueID;

NAME
                                                                 EVENTDATE TIME
                                                                                                                                            VENUENAME
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
ADDRESS
                                                                 CITY
                                                                                                                                  STATE
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
COUNTRY
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
```

11. Write a query to retrieve a list of unique email addresses from both the User and Event tables using set operations.

```
SQL> SELECT Email FROM User_URK22AI1048
  2  UNION
  3  SELECT NULL AS Email FROM Event_URK22AI1048;

EMAIL
--------------------------------------------------
--------------------------------------------------
alex.kim@example.com
```

12.Write a query to retrieve a list of unique cities where events are scheduled or venues are located.

```
SQL>  SELECT TIME FROM Event_URK22AI1048
  2  UNION
  3  SELECT City FROM Venue_URK22AI1048;

TIME
-----------------------------------------------------
-----------------------------------------------------
09:00
11:00
12:00
```

13.Write a query to display the details of Venue ID conducted in the same Venues.

```
SQL> SELECT v1.*
  2  FROM Venue_URK22AI1048 v1
  3  JOIN Venue_URK22AI1048 v2 ON v1.VenueID = v2.VenueID;

  VENUEID NAME
                                                            ADDRESS
                                                                                                    CITY
---------- --------------------------------------------------------------------------------
--------------------------------------------------------------------------------
STATE
                                                   COUNTRY
                                                                                  REMARKS
--------------------------------------------------------------------------------
--------------------
```

14. Write a query to display the details of User ID who are users and have registered for an event.

```
SQL> SELECT u.*
  2  FROM User_URK22AI1048 u
  3  JOIN Ticket_URK22AI1048 t ON u.UserID = t.UserID;

   USERID NAME
                                                            EMAIL
                                                                                                    PASSWORD

          PHONE
---------- --------------------------------------------------------------------------------
--------------------------------------------------------------------------------
----------
        1 John Smith
                                                            john.smith@example.com
                                                                                                    password1
```

15. Write a query to display the details of Event ID which are events but not booked by any one.

```
SQL> SELECT e.*
  2  FROM Event_URK22AI1048 e
  3  LEFT JOIN Ticket_URK22AI1048 t ON e.EventID = t.EventID
  4  WHERE t.TicketID IS NULL;

  EVENTID NAME
                                                            EVENTDATE TIME
                                                                                                    VENUEID
---------- --------------------------------------------------------------------------------
----------------------------------------------------- --------- ----------------------
----------------------------------------------------------------------------------- ---------
DESCRIPTION
```

Result:

The given queries executed by the set operations and joins successfully.