Ex No: 3	**ADVANCED SQL COMMANDS**
Date	**23-1-24**

Aim:

To execute the given commands making use of aggregate functions, group by clause and order by clause.

Description:

Aggregate Functions:

SQL aggregate functions operate on the multiset of values of a column of a relation, and return a value

The various aggregate functions are:

- **Sum**: - returns the sum of the values.

   Eg: select sum(sal) from emp;

- **Avg**:- returns the average of the values.

   Eg: select avg(sal) from emp;

- **Count**:- returns the number of elements in the collection.

   Eg: select count(*) from emp;

- **Min**:- returns the minimum value in a collection.

   Eg: select min(sal) from emp;

- **Max**:- returns the maximum value in a collection.

   Eg: select max(sal) from emp;

The input to sum and average must be a collection of numbers, but the other operators can operate on collections of non-numeric data types, such as strings as well. The average function will return the average of the given tuple. The aggregation function count is used frequently to count the number of tuples in relation.

Distinct Keyword

To eliminate the duplicates, the keyword *distinct* is used in the aggregation expression. SQL does not allow the use of keyword distinct with count (*) to count the number of records in a relation. It is allowed to use distinct with max and minfunctions, even

though the result does not change.

Eg: Select count (distinct job);

GROUP BY Clause

To apply aggregate function to a group of sets of tuples. The attributes given in the group by clause are used to form groups. Tuples with some value on all attributes in the group by clause are placed in one group.

Eg: Select branch_name,avg(bal) from account group by branch_name;

ORDER BY Clause

This clause causes the tuples in the result of a query to appear in sorted order. We specify asc for ascending order and desc for descending order.

Eg: Select * from loan order by amount desc;

Advanced SQL Queries:

1. Find the average price of confirmed tickets

```
SQL> select avg(price) from ticket_urk22ai1048 where status ='Booked';

AVG(PRICE)
----------
    28.375
```

2. Display the starting date as "Start" of all the events from events table.

```
SQL> select eventdate as "start" from event_urk22ai1048;

start
---------
15-AUG-23
20-AUG-23
05-SEP-23
10-AUG-23
25-SEP-23
01-OCT-23
15-OCT-23
05-NOV-23

8 rows selected.
```

3. Find the Minimum date, maximum date of all the events, and the number of months in between the min and max dates

```
SQL> select min(eventdate),max(eventdate),months_between(max(eventdate),min(eventdate)) from event_urk22ai1048;

MIN(EVENT MAX(EVENT MONTHS_BETWEEN(MAX(EVENTDATE),MIN(EVENTDATE))
--------- --------- ----------------------------------------------
10-AUG-23 05-NOV-23                                     2.83870968
```

4. Find the total number of venues used to conduct the events

```
SQL> select count(venueid) from event_urk22ai1048;

COUNT(VENUEID)
--------------
             8
```

5. Find the number of users in User table

```
SQL> select count(*) from user_urk22ai1048;

  COUNT(*)
----------
         8
```

6. Find the length of password of User = p@ssw0rd from User table

```
SQL> select length(password) from user_urk22ai1048 where password='lisa123';

LENGTH(PASSWORD)
----------------
               7
```

7. Concatenate the data and time of Event table as "Date-Time" and display Date-Time and the name of the event if the 5th character of event name is 'e'

```
SQL> select concat(eventdate,time) as "DT" , name from event_urk22ai1048 where substr(name,5,1)='e';

DT
-------------------------------------------------------------------------------
NAME
-------------------------------------------------------------------------------
15-AUG-2318:00
Concert in Park

20-AUG-2320:30
Movie Night

05-NOV-2316:00
Dance Workshop
```

8. Find the user names whose name ends with "pez" from user table

```
SQL> select name from user_urk22ai1048 where name like '%pez';

NAME
--------------------------------------------------------------------------
Lisa Lopez
```

9. Left pad the seat number of Ticket table with "000"

```
SQL> select lpad(seatnumber,5,'0') from ticket_urk22ai1048;

LPAD(SEATNUMBER,5,'0
--------------------
000A1
000B2
000C3
000D4
000A2
000B3
000C4
000D5

8 rows selected.
```

10. Display the event details conducted at the same Venue_ID

```
SQL> select * from event_urk22ai1048 where venueid in(select venueid from event_urk22ai1048 group by venueid having count(*)>1);

no rows selected
```

11. Find out the number of venues in each country from Venue table

```
SQL> select country,count(venueid) from venue_urk22ai1048 group by country;

COUNTRY
--------------------------------------------------------------------------
COUNT(VENUEID)
--------------
USA
             8
```

12. Add a column named "Remarks" in Venue table. Fill the remarks column with "No Remarks" values using NVL2 command. Print VenueID, Country, and Remarks column.

```
SQL> alter table venue_urk22ai1048 add (remarks varchar(100));

Table altered.

SQL> update venue_urk22ai1048 set remarks = NVL2(remarks,'No remarks',remarks);

8 rows updated.
```

13. Use round and trunc functions to round off and truncate the value 25.235789 to 2 decimal positions using dual table.

```
SQL> select round(25.235789,2),trunc(25.235789,2) from dual;

ROUND(25.235789,2) TRUNC(25.235789,2)
------------------ ------------------
            25.24              25.23
```

14. Prefix price column with a value of 0 in ticket table to make the length of price =10 digits.

```
SQL> update ticket_urk22ai1048 set price = lpad(price,10,'0');

8 rows updated.
```

15. Retrieve all events ordered by date and time in ascending order.

```
SQL> select * from event_urk22ai1048 order by eventdate,time ASC;

  EVENTID NAME
                      EVENTDATE TIME
                                                VENUEID DESCRIPTION
--------- ------------------------------------------------------------------------------
------------------------------------ -------- --------------------------------------------
-------------------------------------------------------------- --------- -----------------
-------------------------------------------------------------------------------------------
        4 Art Exhibition
                      10-AUG-23 12:00
                                                    104 Explore various artworks by local artists.
        1 Concert in Park
                      15-AUG-23 18:00
                                                    101 Enjoy a live concert in the city park.
        2 Movie Night
                      20-AUG-23 20:30
                                                    102 Movie night under the stars.
        3 Sports Tournament
                      05-SEP-23 14:00
                                                    110 Join us for an exciting sports tournament.
        5 Food Festival
                      25-SEP-23 11:00
                                                    105 A celebration of diverse cuisines.
        6 Comedy Show
                      01-OCT-23 19:30
                                                    106 Laugh your heart out at our comedy show.
        7 Tech Conference
                      15-OCT-23 09:00
                                                    107 Join tech experts for informative sessions.
        8 Dance Workshop
                      05-NOV-23 16:00
                                                    108 Learn various dance styles in this workshop.
SQL>
```

Result:

The given commands making use of aggregate functions, group byclause and order by clause are executed successfully.