PB_05_Kushagra Suryawanshi Batch B1
(Music Collection)

DBMS LAB - 4.

Title : SQL queries using group by functions, join and nested queries.

Aim : Design and develop SQL queries using group by functions, join and nested queries.

Objectives : To study join and sub-queries.

Theory :

Q1 SQL Join types :

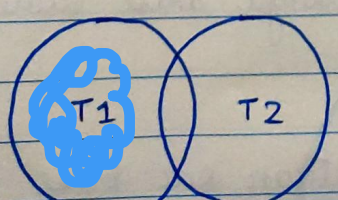
A i.] Inner join : Returns records that has matching values in both the tables.

ii.] Left (Outer) Join : Returns all records from left table and the matched record from right table.

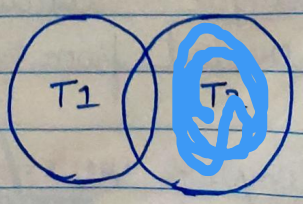
iii.] Right (Outer) Join : Returns all records from right table and the matched records from the left table.

iv.] Full (Outer) join : Returns all records when there is a match in left or right table.

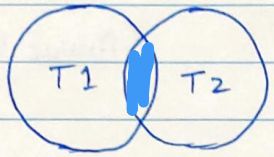
Ex.



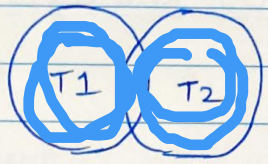
LEFT JOIN



RIGHT JOIN



INNER JOIN



FULL OUTER JOIN

Q2. Subqueries and use:

→ In SQL a subquery can be defined as a query within another query. In other words we can say that a subquery is a query that is embedded in WHERE clause of another SQL query.

Ex. Subquery with select statement:

SELECT column_name FROM table_name WHERE column_name expression operator ;

Q3. Views and types:

A Views are kind of virtual tables that contains data from one or multiple tables. It does not exist physically in the database.

Types of views:

i.] Simple: contains only one single base table or is created from only one table.

ii.] Complex view: contains more than one base tables

or is created from more than one tables.

* Input : Flight Management System Database.

* Output : DDL & DCL commands.

* Platform : MySQL Workbench.

* Conclusions : Thus, we have learned joins & subqueries commands thoroughly.

* FAQs.

Q1) How to get groupwise data from table? What is use of having clause?

→ Inorder to get groupwise data from table, GROUP BY clause is used. This clause divideds the rows in a table into smaller groups. The GROUP BY clause is used with the SQL SELECT statement. The grouping can happen after the retrieves the rows from a table.

* Syntax:

```
SELECT   <column_list>
FROM     <table_name>
WHERE    <condition> GROUP BY <columns>
[Having] <condition>;
```

A Having clause in SQL specifies that an SQL

SELECT statement must be only return rows where aggregate values meet the specified conditions.

Q2.) How to display data from view? Are the views updatable? Explain.

→ To display data from view are not updatable. So, UPDATE command it is not applicable to all views. An updatable view is one which allows performing a UPDATE command on itself without affecting any other table.

Q3.) When to use self joint? How it differes other join?

→ We use self join when a table references data in itself. A self join allows you to join a table to itself. The differences in self join and other joins is that in self join a single table is listed as both the left and right table in the join.

```
Enter password: *************
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.21 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| company            |
| hotel_mgmt         |
| information_schema |
| music_library2     |
| music_system       |
| mysql              |
| performance_schema |
| sakila             |
| sys                |
| world              |
+--------------------+
10 rows in set (0.01 sec)

mysql> use music_library2;
Database changed
mysql> select * from album;
+----------+------------+---------------+-----------+---------------+
| Album_id | album_name | collection_id | artist_id | year_creation |
+----------+------------+---------------+-----------+---------------+
|      101 | Tokyo      |             1 |     10001 |          2015 |
|      102 | Ashiqui    |             2 |     10002 |          2002 |
|      103 | Rap God    |             3 |     10003 |          1995 |
|      104 | Football   |             4 |     10004 |          2020 |
|      105 | Brave      |             5 |     10005 |          2010 |
+----------+------------+---------------+-----------+---------------+
5 rows in set (0.04 sec)

mysql> select * from artist;
+-----------+--------------+
| Artist_id | Artist_name  |
+-----------+--------------+
|     10001 | Charlie Puth |
|     10002 | Arjit Singh  |
```

```
|     10003 | Eminem       |
|     10004 | Shakira      |
|     10005 | Billie       |
+-----------+--------------+
5 rows in set (0.06 sec)

mysql> select * from music_collection;
+---------------+-----------------+
| Collection_id | collection_type |
+---------------+-----------------+
|             1 | Jazz            |
|             2 | Hip Hop         |
|             3 | Rap             |
|             4 | Classical       |
|             5 | Contemporary    |
+---------------+-----------------+
5 rows in set (0.04 sec)

mysql> select * from tracks;
+----------+--------------+----------+-------------+-------------+
| Track_id | Track_name   | album_id | track_price | time_length |
+----------+--------------+----------+-------------+-------------+
|     1001 | Attention    |      101 |         100 |         300 |
|     1002 | Tum Hi Ho    |      102 |         200 |        2000 |
|     1003 | Mocking Bird |      103 |        1000 |        5000 |
|     1004 | Wakka Wakka  |      104 |         150 |       10000 |
|     1005 | Lovely       |      105 |         600 |       10000 |
+----------+--------------+----------+-------------+-------------+
5 rows in set (0.00 sec)
```

```
--QUERIES--
--1--
mysql> SELECT album_name, COUNT(track_id) AS no_of_tracks FROM tracks RIGHT OUTER
JOIN album on tracks.album_id = album.album_id GROUP BY album.album_id;
+------------+--------------+
| album_name | no_of_tracks |
+------------+--------------+
| Tokyo      |            1 |
| Ashiqui    |            1 |
| Rap God    |            1 |
| Football   |            1 |
| Brave      |            1 |
+------------+--------------+
5 rows in set (0.07 sec)
```

```
--2--
```

```
mysql> SELECT artist.Artist_name, COUNT(album.album_id) AS albums_compiled FROM
album RIGHT OUTER JOIN artist on artist.artist_id = album.artist_id GROUP BY
artist.artist_id HAVING albums_compiled > 1;
Empty set (0.06 sec)

mysql> SELECT artist.Artist_name, COUNT(album.album_id) AS albums_compiled FROM
album RIGHT OUTER JOIN artist on artist.artist_id = album.artist_id GROUP BY
artist.artist_id HAVING albums_compiled > 0;
+--------------+-----------------+
| Artist_name  | albums_compiled |
+--------------+-----------------+
| Charlie Puth |               1 |
| Arjit Singh  |               1 |
| Eminem       |               1 |
| Shakira      |               1 |
| Billie       |               1 |
+--------------+-----------------+
5 rows in set (0.00 sec)
```

```
--3--
mysql> SELECT artist_name FROM artist WHERE EXISTS(SELECT * FROM album WHERE
album_id IS NOT NULL AND artist.artist_id = album.artist_id);
+--------------+
| artist_name  |
+--------------+
| Charlie Puth |
| Arjit Singh  |
| Eminem       |
| Shakira      |
| Billie       |
+--------------+
5 rows in set (0.04 sec)
```

```
--4--
mysql> SELECT album_name FROM album, tracks ORDER BY time_length DESC LIMIT 1;
+------------+
| album_name |
+------------+
| Brave      |
+------------+
1 row in set (0.02 sec)
```

```
--5--
mysql> SELECT album_name AS Jazz_Album FROM album, music_collection where
```

```
album.collection_id = music_collection.collection_id AND
music_collection.collection_type = 'Jazz';
+------------+
| Jazz_Album |
+------------+
| Tokyo      |
+------------+
1 row in set (0.03 sec)
```

```
--7--
mysql> SELECT track_id, track_name FROM tracks, album, artist WHERE album.album_id
= tracks.album_id AND album.artist_id = artist.artist_id AND artist.artist_name
LIKE '%a%';
+----------+-------------+
| track_id | track_name  |
+----------+-------------+
|     1001 | Attention   |
|     1002 | Tum Hi Ho   |
|     1004 | Wakka Wakka |
+----------+-------------+
3 rows in set (0.07 sec)
```

```
--8--
mysql> SELECT album_id, track_name FROM tracks WHERE time_length > (SELECT
AVG(time_length) FROM tracks);
+----------+-------------+
| album_id | track_name  |
+----------+-------------+
|      104 | Wakka Wakka |
|      105 | Lovely      |
+----------+-------------+
2 rows in set (0.04 sec)
```

```
--9--
mysql> CREATE VIEW info AS SELECT artist_name, album_name, year_creation FROM
album, artist WHERE artist.artist_id = album.artist_id; SELECT * FROM info;
Query OK, 0 rows affected (0.24 sec)

+--------------+------------+---------------+
| artist_name  | album_name | year_creation |
+--------------+------------+---------------+
| Charlie Puth | Tokyo      |          2015 |
| Arjit Singh  | Ashiqui    |          2002 |
| Eminem       | Rap God    |          1995 |
```

```
| Shakira      | Football   |          2020 |
| Billie       | Brave      |          2010 |
+--------------+------------+---------------+
5 rows in set (0.03 sec)
```

--10--

```
mysql> ALTER TABLE tracks ADD COLUMN price DECIMAL(6,2);SELECT track_name FROM
tracks WHERE track_price > (SELECT MIN(tracks.track_price) FROM tracks);
ERROR 1060 (42S21): Duplicate column name 'price'
+--------------+
| track_name   |
+--------------+
| Tum Hi Ho    |
| Mocking Bird |
| Wakka Wakka  |
| Lovely       |
+--------------+
4 rows in set (0.00 sec)
```

--6--

```
mysql> SELECT artist_name FROM artist WHERE artist_id IN(SELECT a.artist_id FROM album a, album b WHERE a.artist_id = b.artist_id AND
b.year_creation <= (a.year_creation+2) AND a.album_id != b.album_id AND a.year_creation <= b.year_creation GROUP BY a.year_creation, a.artist_id HAVIN
(COUNT(a.year_creation)) = (SELECT (COUNT(a.year_creation)) AS albums_in_two FROM album a, album b WHERE a.artist_id = b.artist_id AND
b.year_creation <= (a.year_creation + 2) AND a.album_id != b.album_id AND a.year_creation <= b.year_creation GROUP BY a.year_creation, a.artist_id
ORDER BY albums_in_two LIMIT 1)+1);
Empty set (0.00 sec)
```