



Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering

Semester: (Fall,Year:2024),B.Sc.in CSE (Day)

LAB REPORT NO - 04

Course Title: Database System

Course Code: CSE210

Section:222-D3

Lab Experiment Name : Subquery Implementation with Aggregate Functions

Student Details

Name		ID
1.	MD.SHAJALAL	223002088

Lab Date : 13 - 11 - 2024

Submission Date : 22 – 11 – 2024

Course Teacher's Name : Farhana Akter Sunny

Lab Report Status

Marks:

Signature:.....

Comments:.....

Date:.....

1. TITLE OF THE LAB REPORT EXPERIMENT

Subquery Implementation with Aggregate Functions

2. OBJECTIVES

To demonstrate the use of subqueries in the provided database schema (concerts, orchestras, and members) by applying aggregate functions, GROUP BY clauses, IN, NOT IN, and comparison operators.

3 Database Schema:

The database contains three tables: **concerts**, **orchestras**, and **members**. Below are their details:

➤ **Concerts**

This table stores details about concerts performed by orchestras.

Columns: id, city, country, year, rating, orchestra_id.

➤ **orchestras**

This table stores details of orchestras.

Columns: id, name, rating, city_origin, country_origin, year.

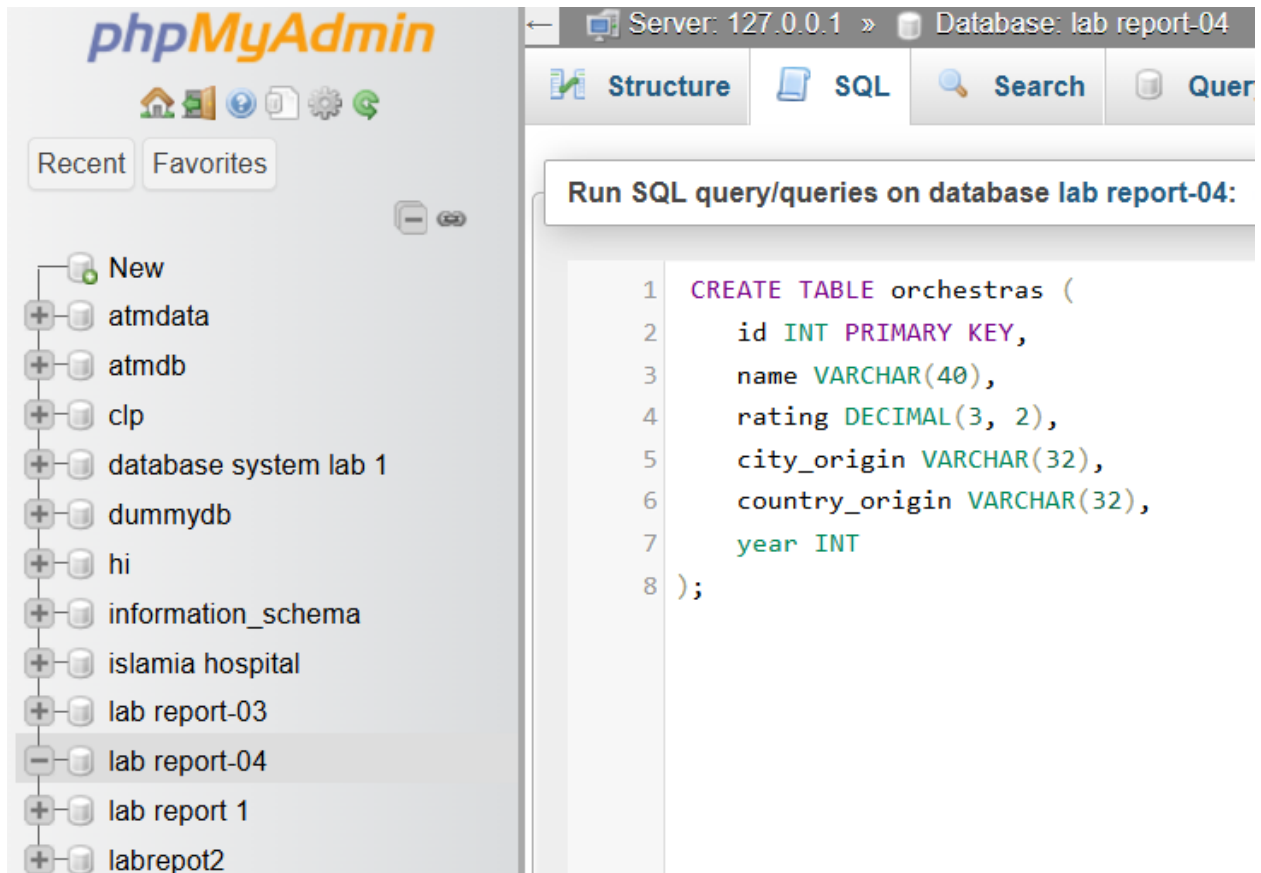
➤ **members**

This table contains details of members of orchestras.

Columns: id, name, position, experience, orchestra_id, wage.

SQL Code to Create Tables:

Table Number:02



The screenshot shows the phpMyAdmin interface. On the left, a sidebar lists various databases, with 'lab report-04' selected. The main panel displays the 'SQL' tab, where a query is entered to create a table named 'orchestras'. The query is as follows:

```
1 CREATE TABLE orchestras (  
2     id INT PRIMARY KEY,  
3     name VARCHAR(40),  
4     rating DECIMAL(3, 2),  
5     city_origin VARCHAR(32),  
6     country_origin VARCHAR(32),  
7     year INT  
8 );
```

OutPut:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(11)			No	None			Change Drop More
<input type="checkbox"/>	2 name	varchar(40)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	3 rating	decimal(3,2)			Yes	NULL			Change Drop More
<input type="checkbox"/>	4 city_origin	varchar(32)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	5 country_origin	varchar(32)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	6 year	int(11)			Yes	NULL			Change Drop More

```
INSERT INTO orchestras (id, name, rating, city_origin, country_origin, year)
VALUES
```

```
(1, 'Md.Shajala', 9.5, 'Lakshmipur', 'Bangladesh', 1900),
```

```
(2, 'Md.Sojib', 9.0, 'Dhaka', 'Bangladesh', 1882),
```

```
(3, 'Mizanur Rhaman', 8.5, 'Noakhali', 'Bangladesh', 1904),
```

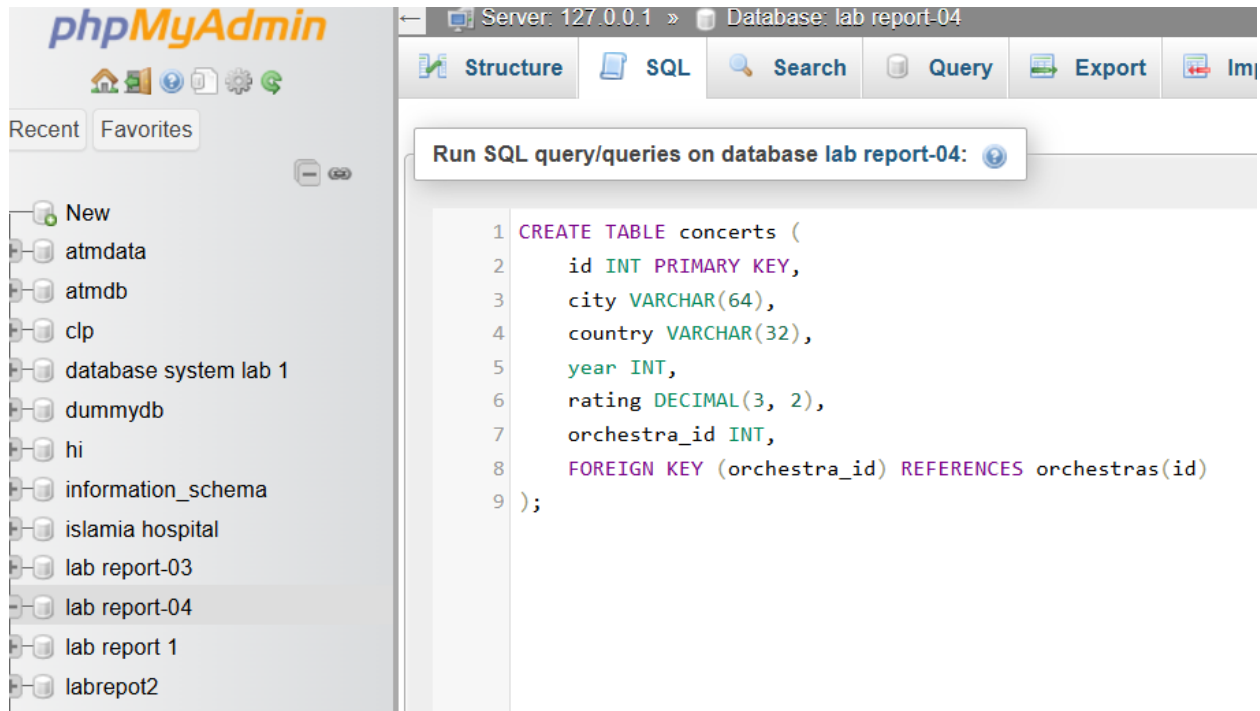
```
(4, 'Ashraful Hridoy', 8.0, 'Feni', 'Bangladesh', 1967),
```

```
(5, 'Abdullah Al Fuad', 9.2, 'Pabna', 'Bangladesh', 1842);
```

Output:

id	name	rating	city_origin	country_origin	year
1	Md.Shajala	9.50	Lakshmipur	Bangladesh	1900
2	Md.Sojib	9.00	Dhaka	Bangladesh	1882
3	Mizanur Rhaman	8.50	Noakhali	Bangladesh	1904
4	Ashraful Hridoy	8.00	Feni	Bangladesh	1967
5	Abdullah Al Fuad	9.20	Pabna	Bangladesh	1842

Table Number :01



INSERT INTO concerts (id, city, country, year, rating, orchestra_id) VALUES
(1, 'Noakhali', 'Bangladesh', 2021, 9.0, 1),
(2, 'Lakshmipur', 'Bangladesh', 2020, 8.5, 2),
(3, 'Feni', 'Bangladesh', '2019', 8.0, 3),
(4, 'Pabna', 'Bangladesh', 2022, 7.8, 4),
(5, 'Dhaka', 'Bangladesh', 2023, 9.2, 1);

Output:

id	city	country	year	rating	orchestra_id
1	Noakhali	Bangladesh	2021	9.00	1
2	Lakshmipur	Bangladesh	2020	8.50	2
3	Feni	Bangladesh	2019	8.00	3
4	Pabna	Bangladesh	2022	7.80	4
5	Dhaka	Banglasesh	2023	9.20	1

Table Number:03

Server: 127.0.0.1 » Database: lab report-04

Structure SQL Search Query Export In

Run SQL query/queries on database lab report-04:

```
1 CREATE TABLE members (  
2     id INT PRIMARY KEY,  
3     name VARCHAR(64),  
4     position VARCHAR(32),  
5     experience INT,  
6     orchestra_id INT,  
7     wage DECIMAL(8, 2),  
8     FOREIGN KEY (orchestra_id) REFERENCES orchestras(id)  
9 );  
10
```

```
INSERT INTO members (id, name, position, experience, orchestra_id, wage)
VALUES
```

```
(1, 'Md.Shajalal', 'A', 10, 1, 5500.00),
```

```
(2, 'Sojib', 'B', 15, 2, 7500.00),
```

```
(3, 'Sakib', 'AB', 8, 3, 5000.00),
```

```
(4, 'Saimon', 'A', 12, 4, 4800.00),
```

```
(5, 'Saima', 'C', 20, 5, 9000.00),
```

```
(6, 'Samiya', 'A', 5, 1, 4000.00);
```

Output:










id	name	position	experience	orchestra_id	wage
1	Md.Shajalal	A	10	1	5500.00
2	Sojib	B	15	2	7500.00
3	Sakib	AB	8	3	5000.00
4	Saimon	A	12	4	4800.00
5	Saima	C	20	5	9000.00
6	Samiya	A	5	1	4000.00

1: Subquery with Aggregate Function:

Find the orchestras with ratings above the average rating.

```
SELECT name, rating
FROM orchestras
WHERE rating > (SELECT AVG(rating) FROM orchestras);
```

Output:






					name	rating
<input type="checkbox"/>	 Edit	 Copy	 Delete	Md.Shajala	9.50	
<input type="checkbox"/>	 Edit	 Copy	 Delete	Md.Sojib	9.00	
<input type="checkbox"/>	 Edit	 Copy	 Delete	Abdullah Al Fuad	9.20	

2: Subquery with GROUP BY

Find countries with more than 2 orchestras.

```
SELECT country_origin
FROM orchestras
GROUP BY country_origin
HAVING COUNT(id) > 2;
```


Output:











		country_origin	
<input type="checkbox"/>	 Edit	 Copy	 Delete Bangladesh
	<input type="checkbox"/> Check all	With selected:  Edit	

3: Subquery Using IN

Retrieve members of orchestras that have participated in concerts rated above 8.

```
SELECT name
FROM members
WHERE orchestra_id IN (
    SELECT orchestra_id
    FROM concerts
    WHERE rating > 8.0
);
```

Output:





				name
<input type="checkbox"/>	 Edit	 Copy	 Delete	Md.Shajalal
<input type="checkbox"/>	 Edit	 Copy	 Delete	Sojib
<input type="checkbox"/>	 Edit	 Copy	 Delete	Samiya

4: Subquery Using NOT IN

Find members who are not part of any orchestra that has performed a concert.

```
SELECT name
FROM members
WHERE orchestra_id NOT IN (
    SELECT DISTINCT orchestra_id
    FROM concerts
);
```

Output:

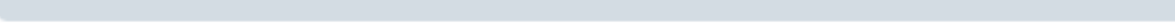
				name
<input type="checkbox"/>	 Edit	 Copy	 Delete	Saima

5: Subquery Using Comparison Operator

Retrieve orchestras that were founded after the oldest concert year.

```
SELECT name  
FROM orchestras  
WHERE year > (SELECT MIN(year) FROM concerts);
```

Output:











		name

6: Nested Subquery with Aggregate Function

Find cities where orchestras' average ratings are above the global average rating.

```
SELECT city_origin  
FROM orchestras  
GROUP BY city_origin  
HAVING AVG(rating) > (SELECT AVG(rating) FROM orchestras);
```

Output:

				city_origin
<input type="checkbox"/>	 Edit	 Copy	 Delete	Dhaka
<input type="checkbox"/>	 Edit	 Copy	 Delete	Lakshmipur
<input type="checkbox"/>	 Edit	 Copy	 Delete	Pabna

▲

Summary of the Database :

The database represents a classical music orchestra system, designed to manage data about orchestras, their members, and concerts. It consists of three interrelated tables: concerts, orchestras, and members.

Conclusion:

This lab exercise demonstrated the ability to extract meaningful information from a database using subqueries. By employing aggregate functions, GROUP BY, and logical operators like IN and NOT IN, the queries addressed real-world scenarios relevant to orchestra and concert management.