



Spring-2024 DataBase Systems (CS-103)

Lab Task # 08 (SQL-Sub Queries)

Instructors: Mr. Hamza Javed

Date: 8 June 2024

Important Instruction

1. No plagiarism allowed.
2. You will required to submit only soft copy via google classroom
3. Late submission not accepted
4. Rename submission file DB24-Name-RegNo. **like: DB24-hamza-1234**
5. You will required to submit single **pdf**

Submission Example

Question .

Write query to select all data from table:

Solution

SELECT * FROM db.detail;

The screenshot shows a database query editor window titled 'Query 1' with a 'detail' tab. The query entered is 'SELECT * FROM sectionb.detail;'. Below the query, the 'Result Grid' is displayed, showing three rows of data. The columns are 'id', 'i_name', 'phone', and 'email'. The data is as follows:

	id	i_name	phone	email
1	1	Aqeel	0323434	assdfgf
1	1	Aqeel	0323434	assdfgf
1	1	Aqeel	0323434	assdfgf

Question 1

Orchestras

In this exercises, the orchestras database will be used that contains following three tables. and you will required to answer the question based on given table by using sql sub-queries.

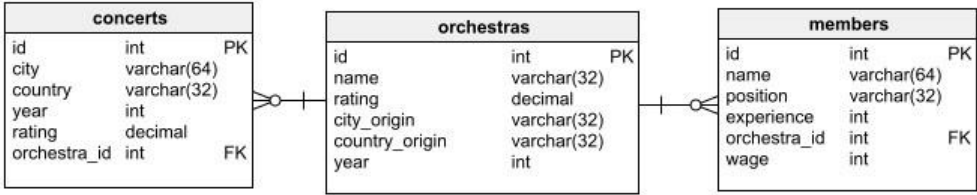


Table 1: Orchestras

id	name	rating	city_origin	country_origin	year
1	Berlin Philharmonic	9.8	Berlin	Germany	1882
2	Vienna Philharmonic	9.7	Vienna	Austria	1842
3	London Symphony	9.5	London	UK	1904
4	New York Philharmonic	9.6	New York	USA	1842
5	Chicago Symphony	9.4	Chicago	USA	1891
6	Los Angeles Philharmonic	9.3	Los Angeles	USA	1919
7	Royal Concertgebouw	9.5	Amsterdam	Netherlands	1888

Table 2: Concerts

id	city	country	year	rating	orchestra_id
1	Berlin	Germany	2020	9.8	1
2	Vienna	Austria	2021	9.7	2
3	London	UK	2019	9.5	3
4	New York	USA	2022	9.6	4
5	Chicago	USA	2021	9.4	5
6	Los Angeles	USA	2018	9.3	6
7	Amsterdam	Netherlands	2023	9.5	7

Table 3: Member

id	name	position	wage	experience	orchestra_id
1	John Smith	Violinist	60000	10	1
2	Anne Brown	Cellist	65000	12	2
3	Robert Johnson	Conductor	120000	15	3
4	Emily Davis	Flutist	55000	8	4
5	Michael Wilson	Percussionist	50000	7	5
6	Sarah Miller	Harpist	70000	10	6
7	David Martinez	Trombonist	58000	9	7
8	Laura Hernandez	Violinist	62000	11	1
9	James White	Trumpeter	59000	9	2
10	Linda Walker	Pianist	60000	8	3

Question 1 (a)

Select the names of all orchestras that have the same city of origin as any city in which any orchestra performed in 2021.

```
SELECT DISTINCT o.name
```

```
FROM orchestras o
```

```
WHERE o.city_origin IN (
```

```
    SELECT DISTINCT c.city
```

```
    FROM concerts c
```

```
    WHERE c.year = 2021
```

```
);
```

(b)

Select the names and positions (i.e. instrument played) of all orchestra members that have above 10 years of experience and do not belong to orchestras with a rating below 9.4

```
SELECT m.name, m.position
```

```
FROM members m
```

```
JOIN orchestras o ON m.orchestras_id = o.id
```

```
WHERE m.experience > 10
```

```
AND o.rating >= 9.4;
```

(c)

Show the name and position of orchestra members who earn more than the average wage of all violinists.

```
SELECT m.name, m.position
```

```
FROM members m
```

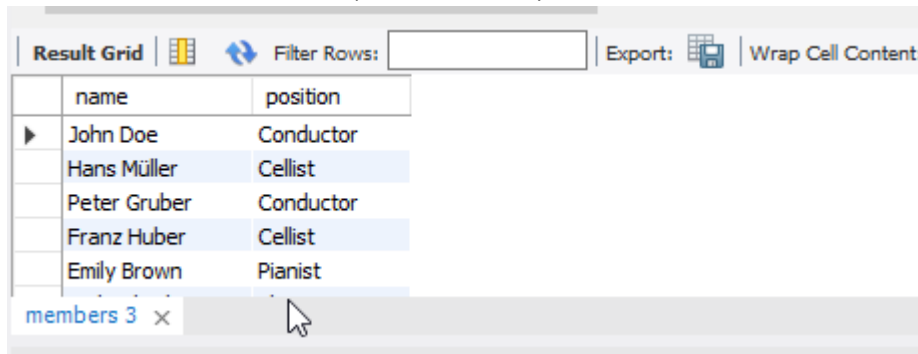
```
WHERE m.wage > (
```

```
    SELECT AVG(wage)
```

```
    FROM members
```

```
    WHERE position = 'Violinist'
```

```
);
```



	name	position
▶	John Doe	Conductor
	Hans Müller	Cellist
	Peter Gruber	Conductor
	Franz Huber	Cellist
	Emily Brown	Pianist

members 3 x

(d)

Show the names of orchestras that were created after the 'Chamber Orchestra' and have a rating greater than 9.5

-- Find the year the 'Chamber Orchestra' was created

SELECT name

FROM orchestras

WHERE year > (

SELECT year

FROM orchestras

WHERE name = 'Chamber Orchestra'

)

AND rating > 9.5;

(e)

Show the name and number of members for each orchestra that has more members than the average membership of all orchestras in the table.

-- Find the year the 'Chamber Orchestra' was created

SELECT name

FROM orchestras

WHERE year > (

SELECT year

FROM orchestras

WHERE name = 'Chamber Orchestra'

```
)  
AND rating > 9.5;
```

-- Step 1: Calculate the number of members for each orchestra

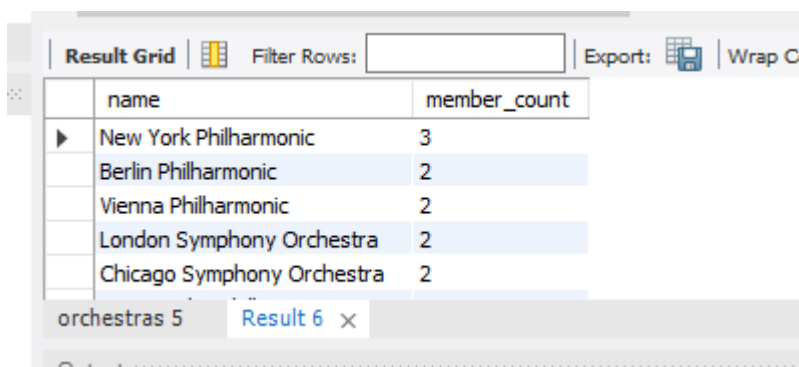
```
WITH OrchestraMemberCount AS (  
    SELECT o.id, o.name, COUNT(m.id) AS member_count  
    FROM orchestras o  
    JOIN members m ON o.id = m.orchestras_id  
    GROUP BY o.id, o.name  
)
```

-- Step 2: Calculate the average number of members across all orchestras

```
AverageMembership AS (  
    SELECT AVG(member_count) AS avg_membership  
    FROM OrchestraMemberCount  
)
```

-- Step 3: Filter orchestras that have more members than the average

```
SELECT omc.name, omc.member_count  
FROM OrchestraMemberCount omc  
JOIN AverageMembership am ON omc.member_count > am.avg_membership;
```



	name	member_count
▶	New York Philharmonic	3
	Berlin Philharmonic	2
	Vienna Philharmonic	2
	London Symphony Orchestra	2
	Chicago Symphony Orchestra	2

orchestras 5 Result 6 ×

Question 2

University

In this exercises, the university database will be used that contains following five tables. and you will required to answer the question based on given table by using sql sub-queries.

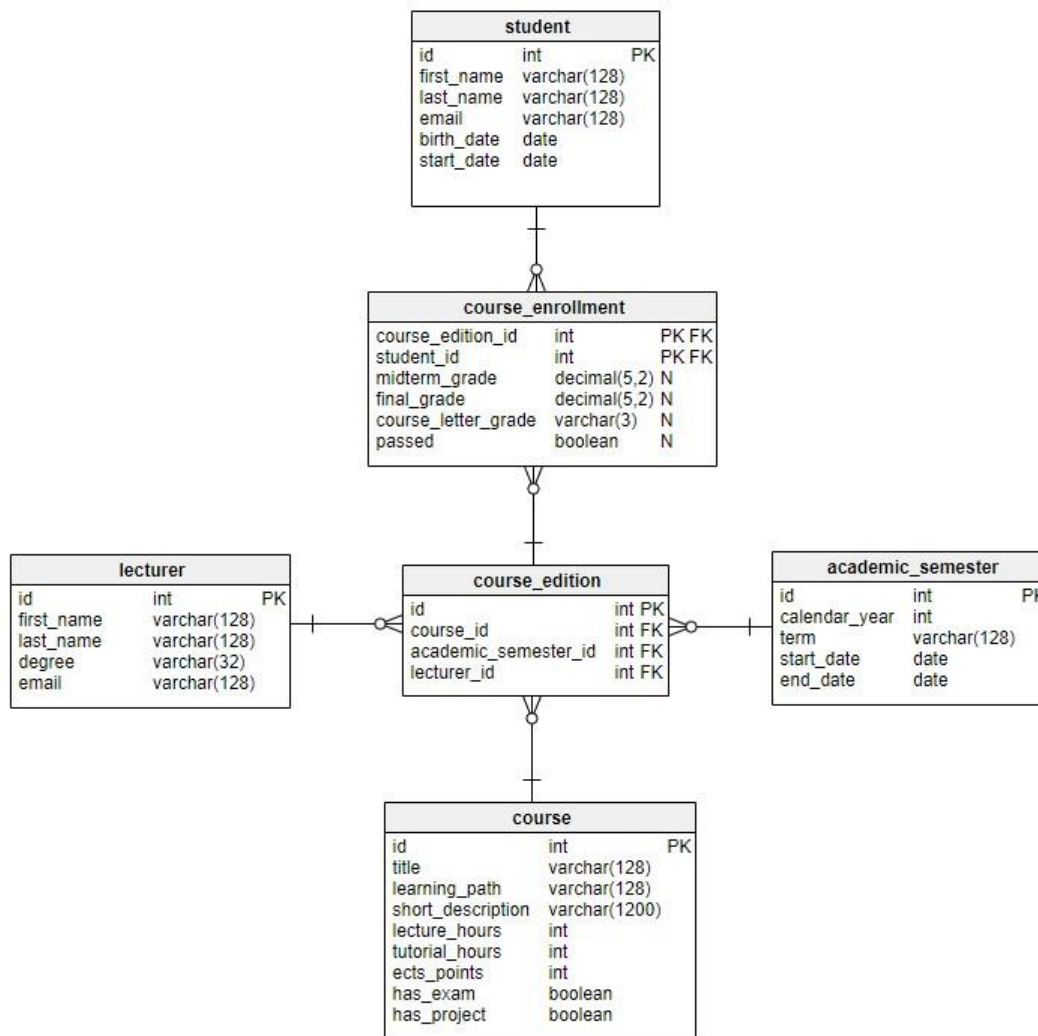


Table 1: Course

id	title	learning_path	short_description	lecture_hours	tutorial_hours	ects_points
1	Introduction to CS	Computer Science	Basics of computer science	30	15	5
2	Data Structures	Computer Science	Study of data structures	40	20	6
3	Algorithms	Computer Science	Introduction to algorithms	45	25	7
4	Databases	Computer Science	Database design and SQL	35	20	6
5	Software Engineering	Computer Science	Software development process	50	25	8
6	Operating Systems	Computer Science	OS principles and design	40	20	7
7	Computer Networks	Computer Science	Networking fundamentals	45	25	7

Table 2: Lecturer

id	first_name	last_name	degree	email
1	Alice	Johnson	PhD in CS	alice.johnson@example.com
2	Bob	Smith	PhD in CS	bob.smith@example.com
3	Carol	Williams	PhD in CS	carol.williams@example.com
4	David	Brown	PhD in CS	david.brown@example.com
5	Eve	Davis	PhD in CS	eve.davis@example.com
6	Frank	Miller	PhD in CS	frank.miller@example.com
7	Grace	Wilson	PhD in CS	grace.wilson@example.com

Table 3: Student

id	first_name	last_name	email	birth_date	start_date
1	John	Doe	john.doe@example.com	2000-01-01	2018-09-01
2	Jane	Smith	jane.smith@example.com	1999-05-15	2017-09-01
3	Michael	Johnson	michael.johnson@example.com	2001-07-20	2019-09-01
4	Emily	Davis	emily.davis@example.com	2000-12-05	2018-09-01
5	Daniel	Wilson	daniel.wilson@example.com	1998-11-25	2016-09-01
6	Sarah	Brown	sarah.brown@example.com	1999-08-14	2017-09-01
7	Matthew	Martinez	matthew.martinez@example.com	2001-03-30	2019-09-01

Table 4: Academic Semester

id	calendar_year	term	start_date	end_date
1	2018	Fall	2018-09-01	2018-12-31
2	2019	Spring	2019-01-15	2019-05-15
3	2019	Fall	2019-09-01	2019-12-31
4	2020	Spring	2020-01-15	2020-05-15
5	2020	Fall	2020-09-01	2020-12-31
6	2021	Spring	2021-01-15	2021-05-15
7	2021	Fall	2021-09-01	2021-12-31

Table 5: Course Edition

id	course_id	academic_semester_id	lecturer_id
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	6
7	7	7	7

Table 6: Course Enrollment

course_edition_id	student_id	midterm_grade	final_grade	course_letter_grade	passed
1	1	85	90	A	Yes
2	2	78	82	B	Yes
3	3	92	88	A	Yes
4	4	65	70	C	Yes
5	5	88	85	A	Yes
6	6	55	60	D	No
7	7	90	95	A+	Yes
1	2	77	79	C+	Yes
2	3	83	87	B+	Yes
3	4	89	91	A	Yes

1. **Display the IDs and titles of all courses that took place during any spring term.**

SELECT course_code, title

FROM courses

WHERE course_code IN (

 SELECT ce.course_id

 FROM course_edition ce

JOIN academic_semester a ON ce.academic_session_id = a.id

WHERE a.term = 'Spring'

);

course_code	title
MATH201	Calculus I
ENG101	English Literature
HIST101	World History
MATH202	Calculus II
NULL	NULL

courses 8

2. **Select the IDs and names of students who passed at least one course.**

SELECT id, CONCAT(f_name, ' ', l_name) AS full_name

FROM student

WHERE id IN (

SELECT DISTINCT student_id

FROM course_enrollement

WHERE passed = TRUE

);

id	full_name
1	John Doe
2	Jane Smith
3	Michael Brown
4	Emily Davis
5	Daniel Wilson

Result 9

3. **Find the lecturer(s) with the least number of courses taught. Display the lecturer's first and last name and the number of courses they teach (as no of courses).**

SELECT l.f_name, l.l_name, COUNT(ce_count.lecturer_id) AS num_of_courses

FROM lecturer l

JOIN (

SELECT lecturer_id, COUNT(*) AS num_of_courses

FROM course_edition

GROUP BY lecturer_id

) AS ce_count ON l.id = ce_count.lecturer_id

WHERE ce_count.num_of_courses = (

```
SELECT MIN(num_of_courses)
FROM (
    SELECT COUNT(*) AS num_of_courses
    FROM course_edition
    GROUP BY lecturer_id
) AS min_courses
)
GROUP BY l.id, l.f_name, l.l_name;
```

4. **Find the student(s) enrolled in the greatest number of course editions. Display the student's ID, first and last names, and the number of course editions they've been enrolled in (as no of course ed).**

```
SELECT s.id, s.f_name, s.l_name, COUNT(ce.student_id) AS num_of_course_editions
FROM student s
JOIN (
    SELECT student_id, COUNT(*) AS num_of_course_editions
    FROM course_enrollement
    GROUP BY student_id
) AS ce ON s.id = ce.student_id
WHERE ce.num_of_course_editions = (
    SELECT MAX(num_of_course_editions)
    FROM (
        SELECT COUNT(*) AS num_of_course_editions
        FROM course_enrollement
        GROUP BY student_id
    ) AS max_ce
)
GROUP BY s.id, s.f_name, s.l_name;
```

Result Grid			Filter Rows: <input type="text"/>	Export:	Wrap C
	id	f_name	l_name	num_of_course_editions	
▶	2	Jane	Smith	1	
	3	Michael	Brown	1	
	4	Emily	Davis	1	
	5	Daniel	Wilson	1	

Result 11 ×