Department of Computer Science
MY University, Islamabad.

Implementation Phase (LAB)

Project No:	08
Course Title:	Database System
Student Name:	Taha Saddiqui
Registration No:	baib232005
Batch:	Fall2023
Program:	Ai
Semester:	6th
Lab Instructor:	Mr. Hamza Javed
Submission Date:	15/06/24

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.

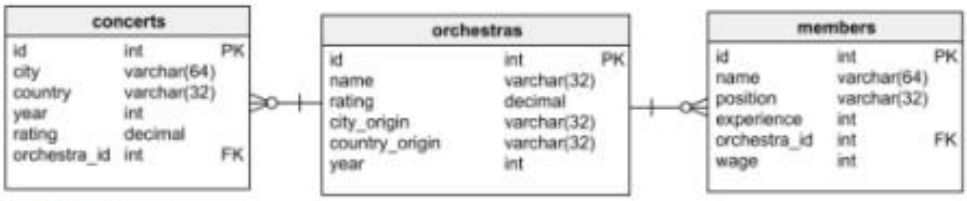

Table 1: Orchestras

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

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

SQL Query

```
SELECT name
FROM orchestras
WHERE city_origin IN (
    SELECT city
    FROM concerts
    WHERE year = 2021
);
```

OUTPUT

name
Vienna Philharmonic
Chicago Symphony

2. **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.**

SQL Query

```
SELECT name, position
FROM members
WHERE experience > 10
AND orchestra_id IN (
    SELECT id
    FROM orchestras
    WHERE rating >= 9.4
);
```

OUTPUT

name	position
Anne Brown	Cellist
Laura Hernandez	Violinist
Robert Johnson	Conductor

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

SQL Query

```
SELECT name, position
FROM members
WHERE wage > (
    SELECT AVG(wage)
    FROM members
    WHERE position = 'Violinist'
);
```

OUTPUT

name	position
Anne Brown	Cellist
Robert Johnson	Conductor
Sarah Miller	Harpist
Laura Hernandez	Violinist

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

There is no "Chamber Orchestra" in the provided table, so we created one.
'8', 'Chamber Orchestra', '9.4', 'New York', 'USA', '1872'

SQL Query

```
SELECT name
FROM orchestras
WHERE year > (
    SELECT year
    FROM orchestras
    WHERE name = 'Chamber Orchestra'
)
AND rating > 9.5;
```

OUTPUT

name
Berlin Philharmonic

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

SQL Query

```
SELECT o.name, COUNT(m.id) AS member_count
FROM orchestras o
JOIN members m ON o.id = m.orchestra_id
GROUP BY o.name
HAVING COUNT(m.id) > (
    SELECT AVG(member_count)
    FROM (
        SELECT COUNT(m.id) AS member_count
        FROM orchestras o
        JOIN members m ON o.id = m.orchestra_id
        GROUP BY o.id
    ) AS avg_member_count
);
```

OUTPUT

	name	member_count
▶	Berlin Philharmonic	2
	Vienna Philharmonic	2
	London Symphony	3

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.**

SQL Query

```
SELECT id, title
FROM course
WHERE id IN (
    SELECT course_id
    FROM course_edition
    WHERE academic_semester_id IN (
        SELECT id
        FROM academic_semester
        WHERE term = 'Spring'
    )
);
```

OUTPUT

id	title
2	Data Structures
4	Databases
6	Operating Systems

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

SQL Query

```
SELECT id, first_name, last_name
FROM student
WHERE id IN (
    SELECT student_id
    FROM course_enrollment
    WHERE passed = TRUE
);
```

OUTPUT

id	first_name	last_name
1	John	Doo
2	Jane	Smith
3	Michael	Johnson
4	Emily	Davis
5	Daniel	Wilson
6	Sarah	Brown
7	Matthew	Martinez

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).

Since all lecturers teach one course, the query sorts them by course count and shows only one result. Thus, the first lecturer in the list appears.

SQL Query

```
SELECT first_name, last_name, no_of_courses
FROM (
    SELECT lecturer.first_name, lecturer.last_name,
COUNT(course_edition.id) AS no_of_courses
    FROM lecturer
    JOIN course_edition ON lecturer.id =
course_edition.lecturer_id
    GROUP BY lecturer.id
) AS sub
ORDER BY no_of_courses ASC
LIMIT 1;
```

OUTPUT

	first_name	last_name	no_of_courses
▶	Alice	Johnson	1

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).

SQL Query

```
SELECT id, first_name, last_name, no_of_course_ed
FROM (
    SELECT student.id, student.first_name,
student.last_name,
COUNT(course_enrollment.course_edition_id) AS
no_of_course_ed
    FROM student
    JOIN course_enrollment ON student.id =
course_enrollment.student_id
    GROUP BY student.id
) AS sub
ORDER BY no_of_course_ed DESC
LIMIT 1;
```

OUTPUT

	id	first_name	last_name	no_of_course_ed
▶	2	Jane	Smith	2