**EXPERIMENT 7**

CODE

import mysql.connector

# Establish connection to the MySQL database

conn = mysql.connector.connect(

    host="localhost",

    user="root",

    password="Guru@2099",

    database="exp6"

)

cursor = conn.cursor()

try:

    # 1. Order the tuples in the instructors relation as per their salary

    cursor.execute("SELECT \* FROM instructor ORDER BY salary")

    result1 = cursor.fetchall()

    print("Ordered Tuples by Salary:")

    for row in result1:

        print(row)

    # 2. Find courses that ran in Fall 2017 or in Spring 2018

    cursor.execute("SELECT \* FROM teaches WHERE semester IN ('Fall', 'Spring') AND year IN (2017, 2018)")

    result2 = cursor.fetchall()

    print("\nCourses in Fall 2017 or Spring 2018:")

    for row in result2:

        print(row)

    # 3. Find courses that ran in Fall 2017 and in Spring 2018

    cursor.execute("SELECT \* FROM teaches WHERE semester = 'Fall' AND year = 2017 INTERSECT SELECT \* FROM teaches WHERE semester = 'Spring' AND year = 2018")

    result3 = cursor.fetchall()

    print("\nCourses in Fall 2017 and Spring 2018:")

    for row in result3:

        print(row)

   # 4. Find courses that ran in Fall 2017 but not in Spring 2018

    cursor.execute("""

    SELECT \* FROM teaches

    WHERE semester = 'Fall' AND year = 2017

    AND (Course\_id, ID) NOT IN (

        SELECT Course\_id, ID FROM teaches WHERE semester = 'Spring' AND year = 2018

    )

    """)

    # 5. Insert additional tuples into the instructor table

    insert\_query = "INSERT INTO instructor (ID, name, dept\_name, salary) VALUES (%s, %s, %s, %s)"

    values = [('10211', 'Smith', 'Biology', 66000), ('10212', 'Tom', 'Biology', None)]

    cursor.executemany(insert\_query, values)

    conn.commit()

    # 6. Find all instructors whose salary is null

    cursor.execute("SELECT \* FROM instructor WHERE salary IS NULL")

    result6 = cursor.fetchall()

    print("\nInstructors with Null Salary:")

    for row in result6:

        print(row)

    # 7. Find the average salary of instructors in the Computer Science department

    cursor.execute("SELECT AVG(salary) FROM instructor WHERE dept\_name = 'Comp. Sci.'")

    result7 = cursor.fetchone()

    print("\nAverage Salary in Computer Science Department:", result7[0])

    # 8. Find the total number of instructors who teach a course in the Spring 2018 semester

    cursor.execute("SELECT COUNT(DISTINCT ID) FROM teaches WHERE semester = 'Spring' AND year = 2018")

    result8 = cursor.fetchone()

    print("\nTotal Instructors Teaching in Spring 2018:", result8[0])

    # 9. Find the number of tuples in the teaches relation

    cursor.execute("SELECT COUNT(\*) FROM teaches")

    result9 = cursor.fetchone()

    print("\nNumber of Tuples in Teaches Relation:", result9[0])

    # 10. Find the average salary of instructors in each department

    cursor.execute("SELECT dept\_name, AVG(salary) FROM instructor GROUP BY dept\_name")

    result10 = cursor.fetchall()

    print("\nAverage Salary in Each Department:")

    for row in result10:

        print(row)

    # 11. Find the names and average salaries of all departments whose average salary is greater than 42000

    cursor.execute("SELECT dept\_name, AVG(salary) AS avg\_salary FROM instructor GROUP BY dept\_name HAVING avg\_salary > 42000")

    result11 = cursor.fetchall()

    print("\nDepartments with Average Salary > 42000:")

    for row in result11:

        print(row)

    # 12. Name all instructors whose name is neither “Mozart” nor “Einstein”

    cursor.execute("SELECT \* FROM instructor WHERE name NOT IN ('Mozart', 'Einstein')")

    result12 = cursor.fetchall()

    print("\nInstructors with Names Not 'Mozart' or 'Einstein':")

    for row in result12:

        print(row)

    # 13. Find names of instructors with salary greater than that of some instructor in the Biology department

    cursor.execute("SELECT \* FROM instructor WHERE salary > ANY (SELECT salary FROM instructor WHERE dept\_name = 'Biology')")

    result13 = cursor.fetchall()

    print("\nInstructors with Salary Greater Than Some in Biology Department:")

    for row in result13:

        print(row)

    # 14. Find the names of all instructors whose salary is greater than the salary of all instructors in the Biology department

    cursor.execute("SELECT \* FROM instructor WHERE salary > ALL (SELECT salary FROM instructor WHERE dept\_name = 'Biology')")

    result14 = cursor.fetchall()

    print("\nInstructors with Salary Greater Than All in Biology Department:")

    for row in result14:

        print(row)

    # 15. Find the average instructors’ salaries of those departments where the average salary is greater than 42,000

    cursor.execute("SELECT dept\_name, AVG(salary) AS avg\_salary FROM instructor GROUP BY dept\_name HAVING avg\_salary > 42000")

    result15 = cursor.fetchall()

    print("\nAverage Salary in Departments with Average Salary > 42000:")

    for row in result15:

        print(row)

    # 16. Find all departments where the total salary is greater than the average of the total salary at all departments

    cursor.execute("SELECT dept\_name FROM instructor GROUP BY dept\_name HAVING SUM(salary) > (SELECT AVG(SUM(salary)) FROM instructor GROUP BY dept\_name)")

    result16 = cursor.fetchall()

    print("\nDepartments with Total Salary > Average Total Salary:")

    for row in result16:

        print(row)

    # 17. List the names of instructors along with the course ID of the courses that they taught

    cursor.execute("SELECT i.name, t.Course\_id FROM instructor i LEFT JOIN teaches t ON i.ID = t.ID")

    result17 = cursor.fetchall()

    print("\nInstructors with Course IDs:")

    for row in result17:

        print(row)

    # 18. List the names of instructors along with the course ID of the courses that they taught. In case, an instructor teaches no courses keep the course ID as null

    cursor.execute("SELECT i.name, IFNULL(t.Course\_id, 'null') FROM instructor i LEFT JOIN teaches t ON i.ID = t.ID")

    result18 = cursor.fetchall()

    print("\nInstructors with Course IDs (Null for No Courses):")

    for row in result18:

        print(row)

except mysql.connector.Error as err:

    print("Error:", err)

# Close cursor and connection

cursor.close()

conn.close()