CSL3050: DBMS

# Assignment 04

By Shashank Shekhar Asthana (B21CS093)

---

For the question given to us, we have to do a variety of things in the problem. First, we have to write the university schema, then populate the tables, and then write the queries.

Below are the displayed screenshots of the tables, which are populated with the six dummy value columns

### TABLE CLASSROOM

| | BUILDING | ROOM_NUMBER | CAPACITY |
|---|---|---|---|
| 1 | Building A | 101 | 30 |
| 2 | Building B | 201 | 40 |
| 3 | Building C | 301 | 25 |
| 4 | Building A | 102 | 35 |
| 5 | Building B | 202 | 45 |
| 6 | Building C | 302 | 20 |

### TABLE DEPARTMENT

| | DEPT_NAME | BUILDING | BUDGET |
|---|---|---|---|
| 1 | Computer Science | Building X | 120000 |
| 2 | Mathematics | Building Y | 96000 |
| 3 | Physics | Building Z | 70000 |
| 4 | Chemistry | Building Z | 60000 |
| 5 | Biology | Building W | 75000 |
| 6 | Economics | Building V | 90000 |

**TABLE INSTRUCTOR**

| | ID | NAME | DEPT_NAME | SALARY |
|---|---|---|---|---|
| 1 | 101 | Prof. Suchetana Chakraborty | Computer Science | 200000 |
| 2 | 102 | Prof. Gaurav | Mathematics | 250000 |
| 3 | 103 | Prof. Kumar | Physics | 180000 |
| 4 | 104 | Prof. Banerjee | Biology | 320000 |
| 5 | 105 | Prof. Narayan | Chemistry | 170000 |
| 6 | 106 | Prof. Philip | Economics | 100000 |

**TABLE COURSE**

| | COURSE_ID | TITLE | DEPT_NAME | CREDIT |
|---|---|---|---|---|
| 1 | CS101 | Introduction to Computer Science | Computer Science | 3 |
| 2 | MATH101 | Calculus I | Mathematics | 4 |
| 3 | PHY101 | Physics I | Physics | 4 |
| 4 | BIO101 | Biology I | Biology | 3 |
| 5 | CHEM101 | Chemistry I | Chemistry | 3 |
| 6 | HIST101 | Indian Economics | Economics | 3 |

**TABLE SECTION**

| | COURSE_ID | SEC_ID | SEMESTER | YEAR | BUILDING | ROOM_NUMBER | TIME_SLOT_ID |
|---|---|---|---|---|---|---|---|
| 1 | CS101 | 1 | Fall | 2023 | Building A | 101 | 1 |
| 2 | MATH101 | 1 | Fall | 2023 | Building B | 201 | 2 |
| 3 | PHY101 | 1 | Fall | 2023 | Building C | 301 | 3 |
| 4 | BIO101 | 1 | Fall | 2023 | Building A | 102 | 4 |
| 5 | CHEM101 | 1 | Fall | 2023 | Building B | 202 | 5 |
| 6 | HIST101 | 1 | Fall | 2023 | Building C | 302 | 6 |

**TABLE TEACHES**

| | ID | COURSE_ID | SEC_ID | SEMESTER | YEAR |
|---|---|---|---|---|---|
| 1 | 101 | CS101 | 1 | Fall | 2023 |
| 2 | 102 | MATH101 | 1 | Fall | 2023 |
| 3 | 103 | PHY101 | 1 | Fall | 2023 |
| 4 | 104 | BIO101 | 1 | Fall | 2023 |
| 5 | 105 | CHEM101 | 1 | Fall | 2023 |
| 6 | 106 | HIST101 | 1 | Fall | 2023 |

**TABLE STUDENT**

| | ID | NAME | DEPT_NAME | TOTAL_CREDIT |
|---|---|---|---|---|
| 1 | 201 | Samuel | Computer Science | 0 |
| 2 | 202 | Shashank Asthana | Mathematics | 0 |
| 3 | 203 | SVS | Physics | 0 |
| 4 | 204 | Manan Choti | Biology | 0 |
| 5 | 205 | SJ | Chemistry | 0 |
| 6 | 206 | Yash | Economics | 0 |

**TABLE TAKES**

| | ID | COURSE_ID | SEC_ID | SEMESTER | YEAR | GRADE |
|---|---|---|---|---|---|---|
| 1 | 201 | CS101 | 1 | Fall | 2023 | A |
| 2 | 202 | MATH101 | 1 | Fall | 2023 | C |
| 3 | 203 | PHY101 | 1 | Fall | 2023 | C |
| 4 | 204 | BIO101 | 1 | Fall | 2023 | A |
| 5 | 205 | CHEM101 | 1 | Fall | 2023 | B |
| 6 | 206 | HIST101 | 1 | Fall | 2023 | C |

**TABLE ADVISOR**

| | S_ID | I_ID |
|---|---|---|
| 1 | 201 | 101 |
| 2 | 202 | 102 |
| 3 | 203 | 103 |
| 4 | 204 | 104 |
| 5 | 205 | 105 |
| 6 | 206 | 106 |

**TABLE TIME_SLOT**

| | TIME_SLOT_ID | DAY | START_TIME | END_TIME |
|---|---|---|---|---|
| 1 | 1 | Monday | 01-SEP-23 | 01-SEP-23 |
| 2 | 2 | Tuesday | 01-SEP-23 | 01-SEP-23 |
| 3 | 3 | Wednesday | 01-SEP-23 | 01-SEP-23 |
| 4 | 4 | Thursday | 01-SEP-23 | 01-SEP-23 |
| 5 | 5 | Friday | 01-SEP-23 | 01-SEP-23 |
| 6 | 6 | Saturday | 01-SEP-23 | 01-SEP-23 |

**TABLE PREREQ**

| | COURSE_ID | PREREQ_ID |
|---|---|---|
| 1 | CS101 | MATH101 |
| 2 | PHY101 | MATH101 |
| 3 | BIO101 | CHEM101 |
| 4 | CHEM101 | PHY101 |
| 5 | HIST101 | BIO101 |
| 6 | MATH101 | HIST101 |

Now we can move on to solving the queries given in the question

**A. Use FOR loop, to insert a Fibonacci sequence into a column in the output table.**

| | FIBONACCI_NUM |
|---|---|
| 1 | 0 |
| 2 | 1 |
| 3 | 1 |
| 4 | 2 |
| 5 | 3 |
| 6 | 5 |

**B. Use CURSOR, FOR loop, and IF statements to insert the (course_id,prereq_id) of prereq table, whose prereq_id is among [1011, 1032, 2310].**

*The screenshot below shown is appeared to not to return any rows because while populating the tables the prere_id is not among [1011,1032,2310]


```
CREATE TABLE output_table2 (
   course_id VARCHAR2(255),
   prereq_id VARCHAR2(255)
);

DECLARE
   v_prereq_id VARCHAR2(255);
   v_course_id VARCHAR2(255);

   CURSOR prereq_cursor IS
      SELECT course_id, prereq_id FROM prereq
      WHERE prereq_id IN ('1011', '1032', '2310');
BEGIN
   FOR prereq_rec IN prereq_cursor LOOP
      v_course_id := prereq_rec.course_id;
      v_prereq_id := prereq_rec.prereq_id;

      -- Insert into the output table
      INSERT INTO output_table2 (course_id, prereq_id)
      VALUES (v_course_id, v_prereq_id);
   END LOOP;
END;
/
```

**The above code displays the query.**

**C. Use WHILE loop/s to display number of instructors in each department. (Output department names and respective number).**

| | DEPT_NAME | INSTRUCTOR_COUNT |
|---|---|---|
| 1 | Mathematics | 1 |
| 2 | Chemistry | 1 |
| 3 | Economics | 1 |
| 4 | Biology | 1 |
| 5 | Computer Science | 1 |
| 6 | Physics | 1 |

**D. Use DBMS_OUTPUT to display the average salary of instructors in each dept, ordered in ascending order (average salary).**

| | DEPARTMENT_NAME | AVERAGE_SALARY |
|---|---|---|
| 1 | Mathematics | 250000 |
| 2 | Chemistry | 170000 |
| 3 | Economics | 100000 |
| 4 | Biology | 320000 |
| 5 | Computer Science | 200000 |
| 6 | Physics | 180000 |

**E. Use UPDATE to increase the grade of students who got an 'F' in the course 'Database Systems/MATH101' to 'C' grade.**

| | ID | COURSE_ID | SEC_ID | SEMESTER | YEAR | GRADE |
|---|---|---|---|---|---|---|
| 1 | 201 | CS101 | 1 | Fall | 2023 | A |
| 2 | 202 | MATH101 | 1 | Fall | 2023 | C |
| 3 | 203 | PHY101 | 1 | Fall | 2023 | C |
| 4 | 204 | BIO101 | 1 | Fall | 2023 | A |
| 5 | 205 | CHEM101 | 1 | Fall | 2023 | B |
| 6 | 206 | HIST101 | 1 | Fall | 2023 | C |

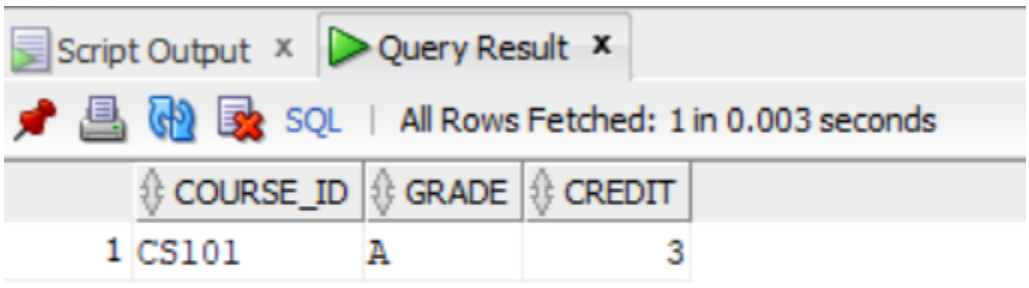
**F. Use UPDATE to increase the department budget by 20% for CS and Maths department.**

| | DEPT_NAME | BUILDING | BUDGET |
|---|---|---|---|
| 1 | Computer Science | Building X | 144000 |
| 2 | Mathematics | Building Y | 115200 |
| 3 | Physics | Building Z | 70000 |
| 4 | Chemistry | Building Z | 60000 |
| 5 | Biology | Building W | 75000 |
| 6 | Economics | Building V | 90000 |

**G. Create an output table that contains the list of courses which are taken by students from atleast 3 different departments.**

| | COURSE_ID |
|---|---|
| 1 | MATH101 |
| 2 | PHY101 |
| 3 | CS101 |
| 4 | CHEM101 |
| 5 | HIST101 |
| 6 | BIO101 |

**H. Use RECORD to create an output table that displays the score of student “Samuel” in a tuple form (grade, credit) for each course taken by him in the last 2 semesters.**

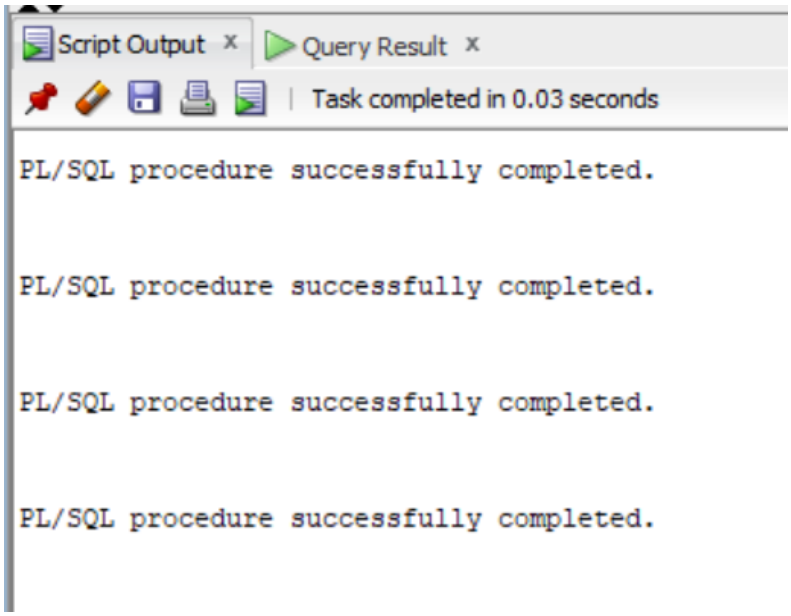



| | COURSE_ID | GRADE | CREDIT |
|---|---|---|---|
| 1 | CS101 | A | 3 |

**I. Use Exception Handling to handle an error occurred by searching for a instructor named “Ram” that is not present in the database.**

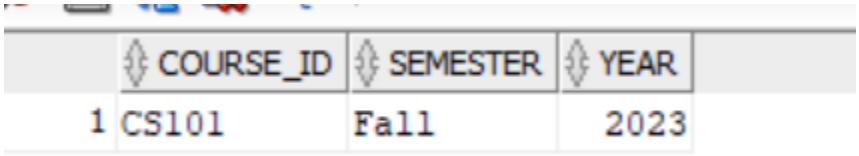
```
DECLARE
   v_instructor_name VARCHAR2(255) := 'Ram';
   v_instructor_id NUMBER(5);
BEGIN
   -- Attempt to find the instructor
   SELECT ID INTO v_instructor_id
   FROM instructor
   WHERE name = v_instructor_name;

   -- If instructor is found, display the ID
   DBMS_OUTPUT.PUT_LINE('Instructor ID: ' || v_instructor_id);
EXCEPTION
   WHEN NO_DATA_FOUND THEN
      -- Handle the exception if instructor is not found
      DBMS_OUTPUT.PUT_LINE('Instructor not found in the database.');
END;
/
```



**J. Create an output table that lists the courses taken by ‘Prof. Suchetana Chakraborty’ within the last 3 years and has more than 5% students getting an A grade.**

```
CREATE TABLE output_table AS
SELECT t.course_id, t.semester, t.year
FROM takes t
JOIN teaches ON teaches.course_id=t.course_id
JOIN instructor i ON t.ID = teaches.ID
WHERE i.name = 'Prof. Suchetana Chakraborty'
   AND t.year >= EXTRACT(YEAR FROM SYSDATE) -3
GROUP BY t.course_id, t.semester, t.year
HAVING SUM(CASE WHEN t.grade = 'B' THEN 1 ELSE 0 END) / COUNT(*) > 0.05;
```



| | COURSE_ID | SEMESTER | YEAR |
|---|---|---|---|
| 1 | CS101 | Fall | 2023 |