Constantine Macris (252850) & Stefanos Georgiou (201841)

Maira Kotsovoulou

ITC 3160: Fundamentals of RDBMS

November 8th, 2021

SQL coursework

Group 1

GitHub URL: <https://github.com/Ogstef/RDBMSmidtermProject>

**1. Display the names of all workers in the accounting department. Use the ‘Accounting’ as a condition and not the department id.**

 SELECT FIRSTNAME||' '||LASTNAME "Name"

   FROM WORKER

   JOIN DEPT

  USING (DEPARTMENTID)

  WHERE DEPARTMENTNAME = 'Accounting';

Graphical user interface, text, application

Description automatically generated

**2. Find the details of any project with the word “urn” anywhere in its name.**

 SELECT projno,

        projname,

        projmgrid,

        budget,

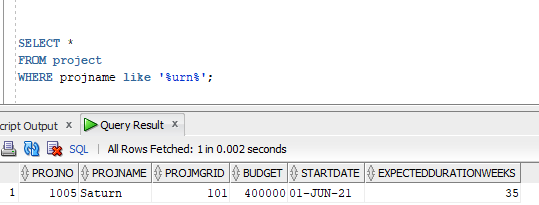
        startdate,

        expecteddurationweeks

   FROM project

  WHERE projname

   LIKE '%urn%';



**3. Assuming that salary contains annual salary, find each worker’s ID, name, and monthly salary.**

 SELECT empid,

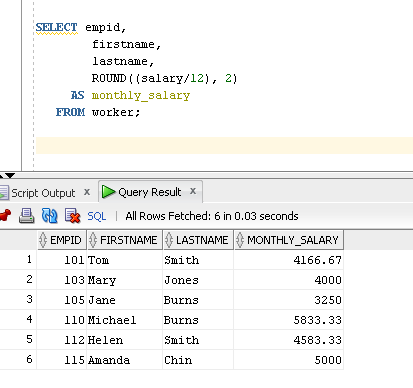
        firstname,

        lastname,

        ROUND((salary/12), 2)

     AS monthly\_salary

   FROM worker;



Graphical user interface, text, application

Description automatically generated**4. Display an alphabetical list of names of all workers assigned to project 1001, sorted by last name.**

     SELECT firstname,

            lastname

       FROM worker

       JOIN assign USING (empid)

       JOIN project USING (projno)

      WHERE projno = '1001'

   ORDER BY worker.lastname ASC;

**5. Display the name of the employee in the research department who has the lowest salary.**

 SELECT firstname,

        lastname,

        salary,

        departmentid

   FROM worker

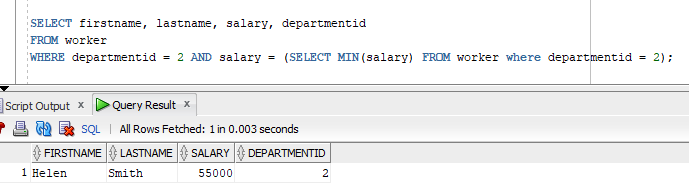
  WHERE departmentid = 2

    AND salary = (

        SELECT MIN(salary)

          FROM worker

         WHERE departmentid = 2);



**6. Display details of the project with the highest budget.**

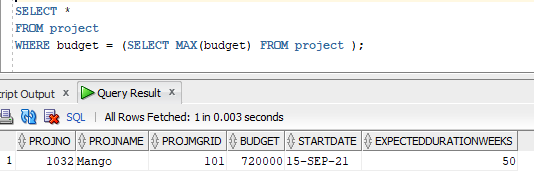
 SELECT \*

   FROM project

  WHERE budget = (

      SELECT MAX(budget)

        FROM project );



**7. Display the names and departments of all workers on project 1019.**

 SELECT firstname,

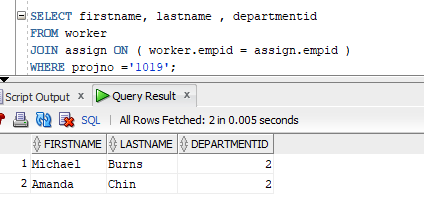
        lastname,

        departmentid

   FROM worker

   JOIN assign ON ( worker.empid = assign.empid )

  WHERE projno = '1019';



**8. Display an alphabetical list of names and corresponding ratings of all workers on any project that is managed by Michael Burns. Use ‘Michael and ‘Burns’ as conditions.**

SELECT firstname, lastname, rating

  FROM worker JOIN assign ON (worker.empid = assign.empid)

 WHERE assign.projno IN(

       SELECT projno

       FROM project

       WHERE projmgrid IN(

                SELECT empid

                FROM worker

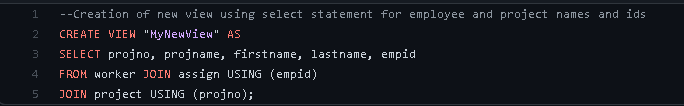
                WHERE lastname = 'Burns'

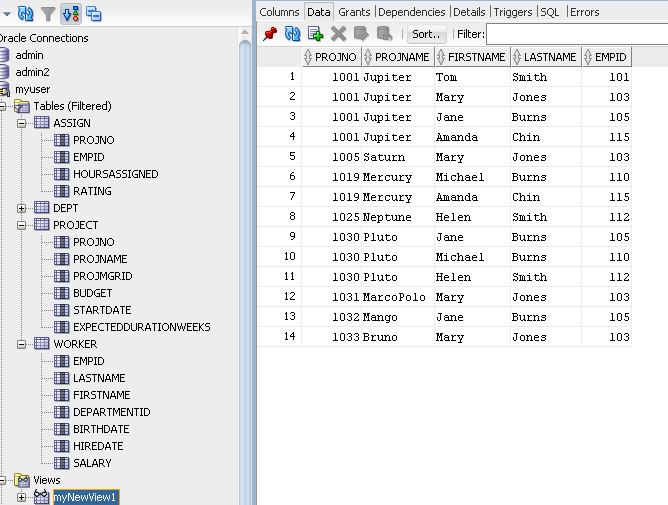
Graphical user interface, text, application

Description automatically generatedTable

Description automatically generated               AND firstname = 'Michael'));

**9. Create a view that has project number and name of each project, along with the IDs and names of all workers assigned to it.**

****

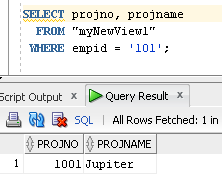


**10. Using the view created in Exercise 7, find the project number and project name of all projects to which employee 1001 is assigned.**

SELECT projno, projname

FROM "myNewView"

WHERE empid = '101';

Graphical user interface, text, application

Description automatically generated

**11. Add new workers, named with your names and your IDs to the research department. Assign yourselves into 2 projects each. Select everything from the view you created in Exercise 7.**

INSERT INTO worker(empid, lastname, firstname,

            departmentid, birthdate, hiredate, salary)

     VALUES (201841, 'Georgiou', 'Stefanos', 2,

            '13-APR-1999', '15-APR-2021', 75000);

INSERT INTO assign (projno , empid, hoursassigned, rating)

     VALUES ('1001', '201841', '15', '3');

INSERT INTO assign (projno , empid, hoursassigned, rating)

     VALUES ('1030', '201841', '15', '3');

INSERT INTO worker(empid, lastname, firstname,

            departmentid, birthdate, hiredate, salary)

     VALUES (201869, 'Macris', 'Constantine', 2 ,

            '03-AUG-2001', '27-MAR-2021', 100000);

INSERT INTO assign (projno , empid, hoursassigned, rating)

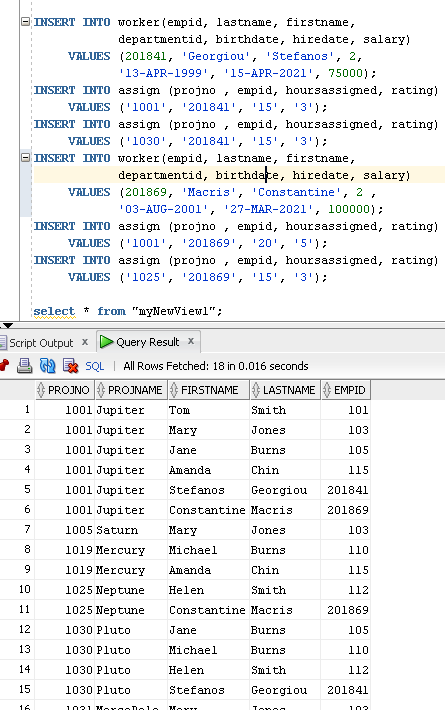
     VALUES ('1001', '201869', '20', '5');

INSERT INTO assign (projno , empid, hoursassigned, rating)

     VALUES ('1025', '201869', '15', '3');

     SELECT \*

       FROM "myNewView";

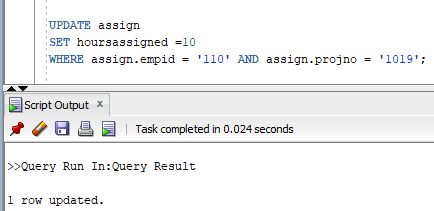
A picture containing graphical user interface

Description automatically generated

**12. Change the hours, which employee 110 is assigned to project 1019, from 20 to 10.**

UPDATE assign

SET hoursassigned =10

Table

Description automatically generated with medium confidenceWHERE assign.empid = '110' AND assign.projno = '1019';

**13. For all projects starting after Sep 1, 2021, find the project number and the IDs and names of all workers assigned to them.**

 SELECT PROJNO,

        EMPID,

        firstname || ' ' || lastname

     AS fullname

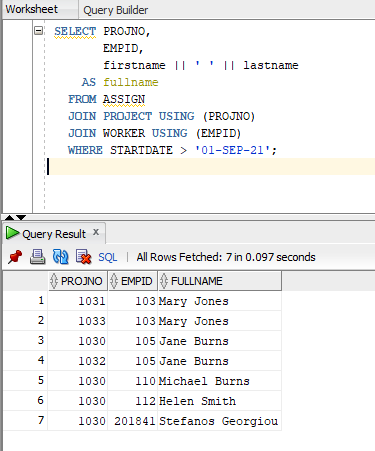
   FROM ASSIGN

   JOIN PROJECT USING (PROJNO)

   JOIN WORKER USING (EMPID)

  WHERE STARTDATE > '01-SEP-21';

**Graphical user interface, application

Description automatically generated**

**14. For each project, list the project number, how many workers are assigned to it and how many hours they are assigned for.**

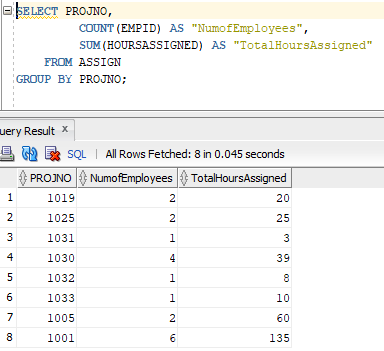
  SELECT PROJNO,

         COUNT(EMPID) AS "NumofEmployees",

         SUM(HOURSASSIGNED) AS "TotalHoursAssigned"

    FROM ASSIGN

GROUP BY PROJNO;

****

**15. Find the employee names and department manager names of all workers who are not assigned to any project.**

--EDIT: We were given useful info via a wonderful professor and inserted a worker

   SELECT firstname||' '||lastname "Name",

          PROJNO,

          EMPID

     FROM WORKER

LEFT JOIN ASSIGN

    USING (EMPID)

    WHERE PROJNO IS NULL;

**Table, calendar

Description automatically generated**

**16. Display a list of project numbers and names and starting dates of all projects that have the same starting date.**

SELECT PROJNAME, PROJNO, STARTDATE

FROM PROJECT

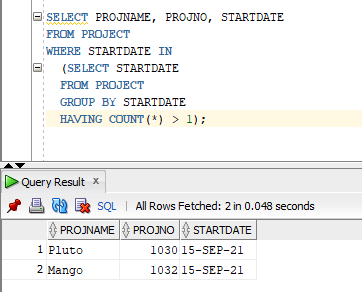
WHERE STARTDATE IN

  (SELECT STARTDATE

  FROM PROJECT

  GROUP BY STARTDATE

  HAVING COUNT(\*) > 1);

****

**17. Add a field called status to the Project table (Sample values for this field are active, completed, planned, cancelled). Update the Projects table and make some of them active, one completed and one cancelled. Display a list of all ‘active’ projects.**

ALTER TABLE PROJECT

ADD (STATUS VARCHAR(20));

--Arbitrary assignment of active, cancelled, and completed project statuses

    UPDATE PROJECT

SET STATUS = 'ACTIVE'

     WHERE MOD(PROJNO,2) = 1;

    UPDATE PROJECT

SET STATUS = 'CANCELLED'

     WHERE MOD(PROJNO,2) = 0;

    UPDATE PROJECT

SET STATUS = 'COMPLETED'

     WHERE PROJNO = 1032;

--Displays all active projects from projects table

SELECT PROJNO,

        PROJNAME,

        PROJMGRID,

        BUDGET,

        STARTDATE,

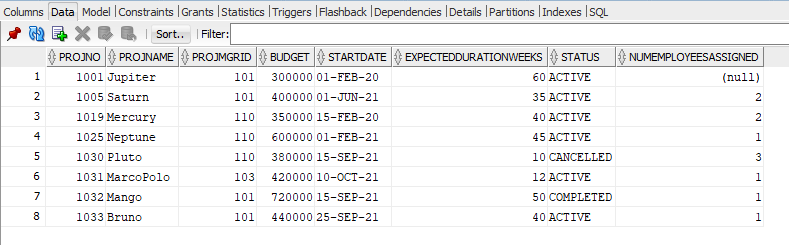
        EXPECTEDDURATIONWEEKS,

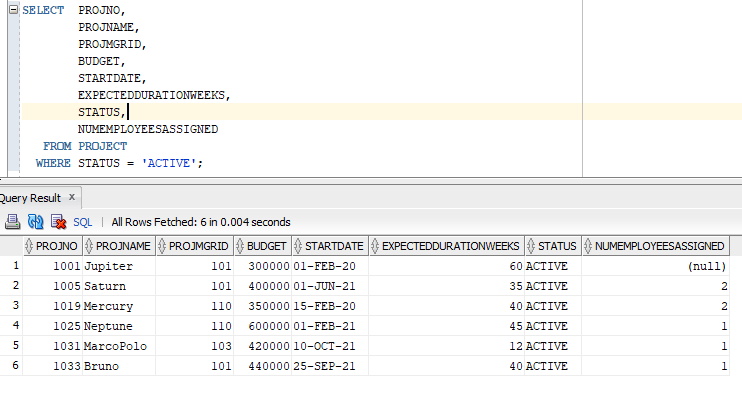
        STATUS,

        NUMEMPLOYEESASSIGNED

   FROM PROJECT

  WHERE STATUS = 'ACTIVE';

****

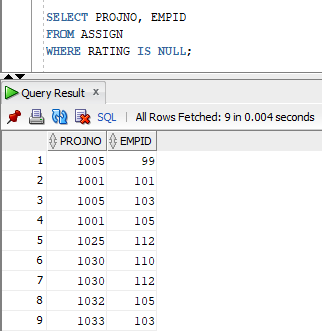
****

**18. Display the employee ID and project number of all employees who have no ratings on that project.**

SELECT PROJNO, EMPID

  FROM ASSIGN

 WHERE RATING IS NULL;

****

**19. Add a field called numEmployeesAssigned to the Project table. Use the UPDATE command to insert values into the field to correspond to the current information in the Assign table.**

--Adding field for number of employees

ALTER TABLE PROJECT ADD(

numEmployeesAssigned NUMBER(7,0));

--Using DML to calculate number of employees assigned per project

  SELECT PROJNO,

         COUNT(EMPID)

    FROM ASSIGN

GROUP BY PROJNO;

--Updating project table field "numEmployeesassigned" accordingly

UPDATE PROJECT

   SET numEmployeesassigned = 4

 WHERE PROJNO = 1001;

UPDATE PROJECT

   SET numEmployeesassigned = 3

 WHERE PROJNO = 1030;

UPDATE PROJECT

   SET numEmployeesassigned = 2

 WHERE PROJNO = 1019;

UPDATE PROJECT

   SET numEmployeesassigned = 1

 WHERE PROJNO = 1005;

UPDATE PROJECT

   SET numEmployeesassigned = 1

 WHERE PROJNO = 1025;

UPDATE PROJECT

   SET numEmployeesassigned = 1

 WHERE PROJNO BETWEEN 1031 AND 1033;

**Table

Description automatically generated**

**20. Write a trigger that will update the numEmployeesAssigned field correctly whenever an assignment is made, dropped, or updated.**

--Creating/modifying trigger to synchronize project table and assign table

CREATE OR REPLACE TRIGGER update\_Employees\_assigned

        AFTER INSERT OR DELETE OR UPDATE ON ASSIGN

    FOR EACH ROW

BEGIN

    IF INSERTING THEN

        UPDATE PROJECT SET numEmployeesassigned = numEmployeesassigned + 1

            WHERE PROJNO = :NEW.PROJNO;

    END IF;

    IF DELETING THEN

        UPDATE PROJECT SET numEmployeesassigned = numEmployeesassigned - 1

            WHERE PROJNO = :OLD.PROJNO;

    END IF;

    IF UPDATING THEN

        UPDATE PROJECT SET numEmployeesassigned = numEmployeesassigned + 1

            WHERE PROJNO = :NEW.PROJNO;

        UPDATE PROJECT SET numEmployeesassigned = numEmployeesassigned - 1

            WHERE PROJNO = :OLD.PROJNO;

    END IF;

END;

--Testing to see if trigger works properly

INSERT INTO WORKER (

            EMPID,

            LASTNAME,

            FIRSTNAME,

            DEPARTMENTID,

            BIRTHDATE,

            HIREDATE,

            SALARY)

     VALUES (99,

            'Gaye',

            'Marvin', 1,

            '05-APR-90',

            '20-SEP-95',

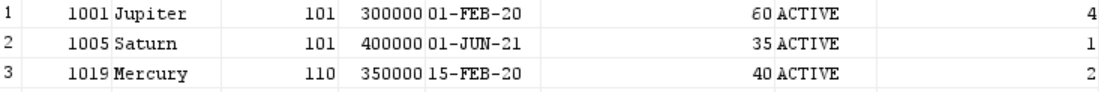
            999999);

INSERT INTO ASSIGN (PROJNO, EMPID, HOURSASSIGNED)

     VALUES (1005, 99, 40);

DELETE FROM ASSIGN

      WHERE EMPID = 99 AND PROJNO = 1005;

--IT WORKS HALELLUJAH

Text

Description automatically generatedText

Description automatically generated with medium confidence

