ITC3160 – Fundamentals of RDBMS, I.Christou

Danai-Georgia Sakelliadou, ID:243439

Andreas Natsis Vertsiotis, ID: 253537

Github repo: <https://github.com/dgsakelliadou/ITC3160>

1. Display the names of all workers in the accounting department. Use the ‘Accounting’ as a condition and not the department id. (4points)

Select firstName, lastName

from worker

where departmentID = (select departmentID from dept where departmentName = 'Accounting')

Graphical user interface, text

Description automatically generated

1. Find the details of any project with the word “urn” anywhere in its name. (4points)

SELECT \* FROM project

WHERE projname LIKE '%urn%'

Graphical user interface, text, application

Description automatically generated

1. Assuming that salary contains annual salary, find each worker’s ID, name, and monthly salary. (4points)

select empId, firstName, lastName, round(salary/12,2) as monthly\_salary

from worker

Table

Description automatically generated with medium confidence

1. Display an alphabetical list of names of all workers assigned to project 1001, sorted by last name. (4points)

select lastname, firstname

from worker w, assign a

where w.empid = a.empid

and a.projno = '1001'

order by lastname

Graphical user interface, text, application

Description automatically generated

1. Display the name of the employee in the research department who has the lowest salary. (6points)

select firstname, lastname

from worker w

where w.salary = (select min(salary) from worker where departmentid = (select departmentid from dept where departmentname = 'Research'))

group by firstname, lastname, salary

Table

Description automatically generated

1. Display details of the project with the highest budget. (4points)

SELECT PROJNO ,

PROJNAME ,

PROJMGRID ,

BUDGET ,

STARTDATE ,

EXPECTEDDURATIONWEEKS

from project p

where p.budget = (select max(budget)from project)

Graphical user interface, text, application, email

Description automatically generated

1. Display the names and departments of all workers on project 1019. (4points)

SELECT firstname, lastname, departmentID

FROM worker w, assign a

WHERE w.empid = a.empid

AND a.projno = '1019'

Graphical user interface, text, application

Description automatically generated

1. 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. (4points)

SELECT firstname, lastname, rating

FROM assign a, worker w

WHERE a.projno IN (select projno from project p where p.projmgrid=(select empid from worker where firstname = 'Michael' and lastname = 'Burns'))

and w.empid = a.empid

order by lastname, firstname

Graphical user interface, text, application, email

Description automatically generated

1. Create a view that has project number and name of each project, along with the IDs and names of all workers assigned to it. (6points)

create view project\_list as select p.projno, p.projname, a.empid, w.lastname, w.firstname

from project p, assign a, worker w

where p.projno = a.projno and a.empid = w.empid

order by a.projno;

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

1. Using the view created in Exercise 7, find the project number and project name of all projects to which employee 1001 is assigned. (6points)

select projno, projname

from project\_list

where empid = '101';

Graphical user interface, text, application, email

Description automatically generated

1. Add new workers, named with your names and your IDs to the research department. Assign your selves into 2 projects each. Select everything from the view you created in Exercise 7. (6points)

insert into worker values(243439,'Sakelliadou', 'Danai Georgia', 2, '20-Dec-1998','5-Nov-2021', 50000);

insert into worker values(253537, 'Vervitsiotis Natsis','Andreas', 2, '12-Feb-2002', '5-Nov-2021', 50000);

INSERT INTO Assign VALUES (1001, 243439, 30,5);

INSERT INTO Assign VALUES (1005, 243439, 30,5);

INSERT INTO Assign VALUES (1001, 253537, 30,5);

INSERT INTO Assign VALUES (1005, 253537, 30,5);

select \* from project\_list

Table

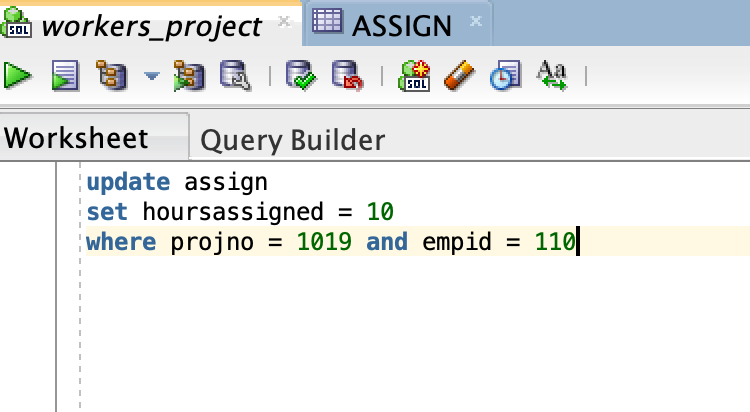
Description automatically generated with medium confidence

1. Change the hours, which employee 110 is assigned to project 1019, from 20 to 10. (4points)

update assign

set hoursassigned = 10

where projno = 1019 and empid = 110



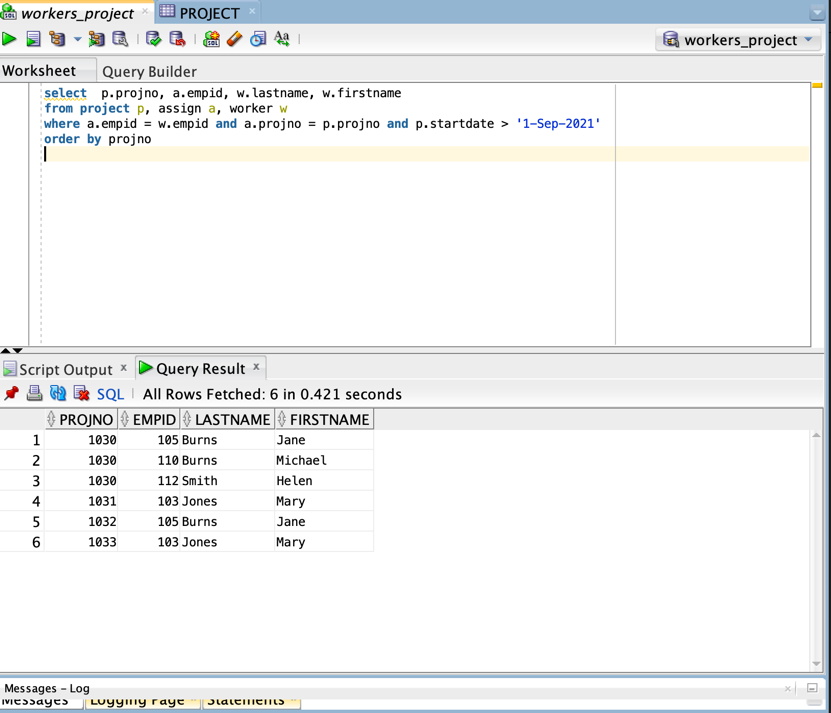
1. For all projects starting after Sep 1, 2021, find the project number and the IDs and names of all workers assigned to them. (4points)

select p.projno, a.empid, w.lastname, w.firstname

from project p, assign a, worker w

where a.empid = w.empid and a.projno = p.projno and p.startdate > '1-Sep-2021'

order by projno



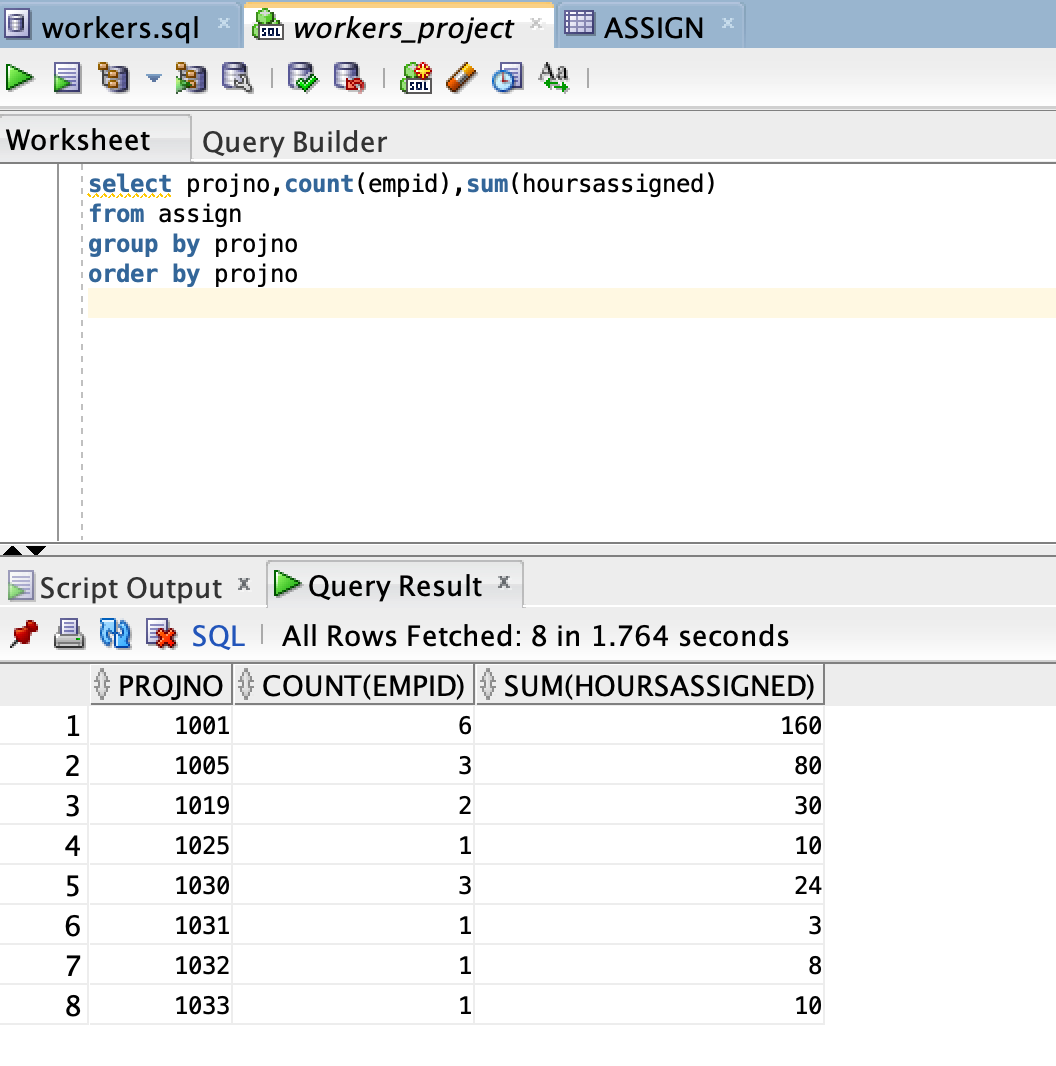
1. For each project, list the project number, how many workers are assigned to it and how many hours they are assigned for. (6points)

select projno,count(empid),sum(hoursassigned)

from assign

group by projno

order by projno



1. Find the employee names and department manager names of all workers who are not assigned to any project. (6points)

select w1.lastname , w1.firstname , w2.lastname as managerlastname, w2.firstname as managerfirstname

from dept d left join worker w1 on w1.empid not in (select empid from assign)

left join worker w2 on d.mgrid = w2.empid

where d.departmentid = w1.departmentid

Graphical user interface, text, application

Description automatically generated

1. Display a list of project numbers and names and starting dates of all projects that have the same starting date. (6points)

select projno, projname, startdate

from project

where startdate in (select startdate from project group by startdate having

count(\*) > 1)

Graphical user interface, text, application, email

Description automatically generated

1. 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. (6points)

alter table project

add status varchar2(20);

update project set status = 'cancelled' where projno = '1033';

update project set status = 'active' where projno = '1032';

update project set status = 'completed' where projno = '1031';

update project set status = 'active' where projno = '1030';

update project set status = 'active' where projno = '1025';

update project set status = 'active' where projno = '1019';

update project set status = 'active' where projno = '1005';

update project set status = 'active' where projno = '1001';

select \* from project where status = 'active'

Table

Description automatically generated

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

select empid, projno

from assign where rating is null

order by empid

Table

Description automatically generated

1. 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. (6points)

alter table project

add numEmployeesAssigned int;

update project set numEmployeesAssigned = '1' where projno = '1033';

update project set numEmployeesAssigned = '1' where projno = '1032';

update project set numEmployeesAssigned = '1' where projno = '1031';

update project set numEmployeesAssigned = '3' where projno = '1030';

update project set numEmployeesAssigned = '1' where projno = '1025';

update project set numEmployeesAssigned = '2' where projno = '1019';

update project set numEmployeesAssigned = '3' where projno = '1005';

update project set numEmployeesAssigned = '6' where projno = '1001';

A picture containing table

Description automatically generated

Table

Description automatically generated

1. Write a trigger that will update the numEmployeesAssigned field correctly whenever an assignment is made, dropped, or updated. (6points)

We wrote 3 triggers, each for each action.

1 . update

create or replace TRIGGER UPDATE\_ASSIGN

after update of empid, projno on assign

for each row

begin

update project

set numemployeesassigned = numemployeesassigned - 1

where projno = :old.projno;

update project

set numemployeesassigned = numemployeesassigned + 1

where projno = :new.projno;

end;

Graphical user interface, text, application, email

Description automatically generated

2. deletion

create or replace TRIGGER UPDATE\_ASSIGN\_delete

after delete on assign

for each row

begin

update project

set numemployeesassigned = numemployeesassigned - 1

where projno= :old.projno;

end;

Graphical user interface, text, application, email

Description automatically generated

3.insertion

create or replace TRIGGER UPDATE\_ASSIGN\_insert

after insert on assign

for each row

begin

update project

set numemployeesassigned = numemployeesassigned + 1

where projno= :new.projno;

end;

Graphical user interface, text, application, email

Description automatically generated

Finally, checking the UPDATE ASSIGN TRIGGER:

Graphical user interface, text, application, chat or text message

Description automatically generatedTable

Description automatically generated

Checking the UPDATE\_ASSIGN\_INSERT trigger:

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

Checking the UPDATE\_ASSIGN\_DELETE trigger:

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated