The University of Texas at Dallas

Database Design – Prof. Nurcan, Yuruk

**Assignment 02**

**Part1 - 1.** Creating Library Database:

**a. creating the tables:**

--DROP TABLE if exists BOOKS;

CREATE TABLE BOOKS (

book\_id varchar(10),

title varchar(20),

publ\_name varchar(20),

primary key(book\_id));

CREATE TABLE BOOK\_AUTHORS (

book\_id varchar(10),

author\_name varchar(20),

primary key(book\_id, author\_name));

CREATE TABLE PUBLISHER (

pname varchar(20),

address varchar(50),

phone varchar(20),

primary key(pname));

CREATE TABLE BOOK\_COPIES (

book\_id varchar(10),

branch\_id varchar(10),

no\_of\_copies integer,

primary key(book\_id, branch\_id));

CREATE TABLE BOOK\_LOANS (

book\_id varchar(10),

branch\_id varchar(10),

card\_no varchar(10),

date\_out date,

due\_date date,

return\_date date default null,

primary key(book\_id, branch\_id, card\_no));

CREATE TABLE LIBRARY\_BRANCH (

branch\_id varchar(10),

branch\_name varchar(20),

address varchar(50),

primary key(branch\_id));

CREATE TABLE BORROWER (

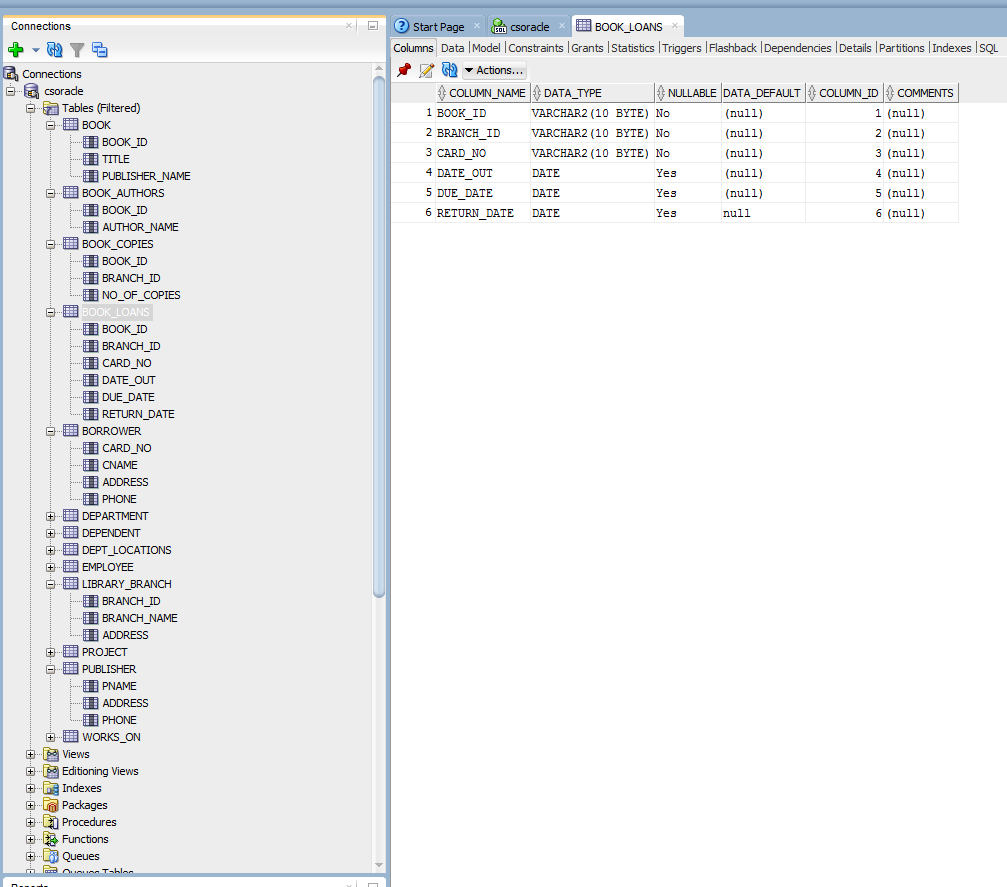
card\_no varchar(10),

cname varchar(20),

address varchar(50),

phone varchar(20),

primary key(card\_no));

****

***\*\* The default value for RETURN\_DATE is set to be NULL.***

**b. triggered actions that will be attached to each foreign key constraint:**

ALTER TABLE BOOK ADD CONSTRAINT fk1 FOREIGN KEY(publisher\_name) REFERENCES publisher(pname) ON DELETE CASCADE;

ALTER TABLE BOOK\_AUTHORS ADD CONSTRAINT fk2 FOREIGN KEY(book\_id) REFERENCES BOOK(book\_id) ON DELETE CASCADE;

ALTER TABLE BOOK\_COPIES ADD CONSTRAINT fk3 FOREIGN KEY(book\_id) REFERENCES BOOK(book\_id) ON DELETE CASCADE;

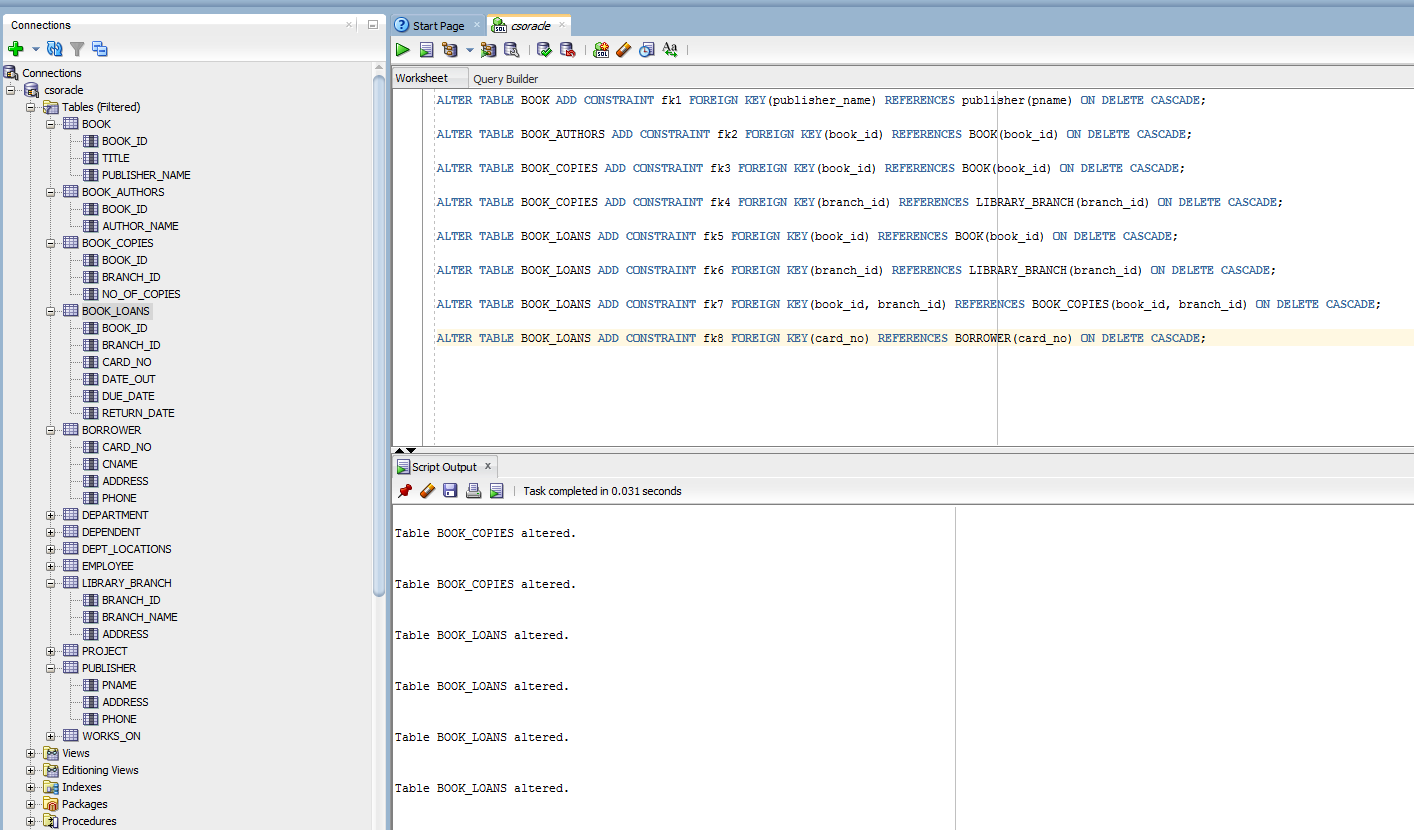
ALTER TABLE BOOK\_COPIES ADD CONSTRAINT fk4 FOREIGN KEY(branch\_id) REFERENCES LIBRARY\_BRANCH(branch\_id) ON DELETE CASCADE;

ALTER TABLE BOOK\_LOANS ADD CONSTRAINT fk5 FOREIGN KEY(book\_id) REFERENCES BOOK(book\_id) ON DELETE CASCADE;

ALTER TABLE BOOK\_LOANS ADD CONSTRAINT fk6 FOREIGN KEY(branch\_id) REFERENCES LIBRARY\_BRANCH(branch\_id) ON DELETE CASCADE;

ALTER TABLE BOOK\_LOANS ADD CONSTRAINT fk7 FOREIGN KEY(book\_id, branch\_id) REFERENCES BOOK\_COPIES(book\_id, branch\_id) ON DELETE CASCADE;

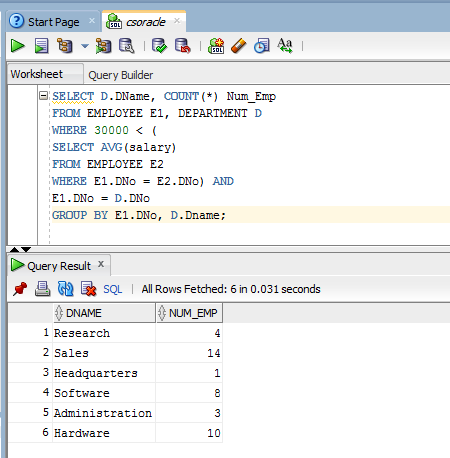
ALTER TABLE BOOK\_LOANS ADD CONSTRAINT fk8 FOREIGN KEY(card\_no) REFERENCES BORROWER(card\_no) ON DELETE CASCADE;



**Part 2**

**1. Queries on COMPANY database.**

**a.** *for each dept with AVG\_EMP\_SALARY > 30000, deptName and number of employees in it.*

****

SELECT D.DName, COUNT(\*) Num\_Emp

FROM EMPLOYEE E1, DEPARTMENT D

WHERE 30000 < (

SELECT AVG(salary)

FROM EMPLOYEE E2

WHERE E1.DNo = E2.DNo) AND

E1.DNo = D.DNo

GROUP BY E1.DNo, D.Dname;

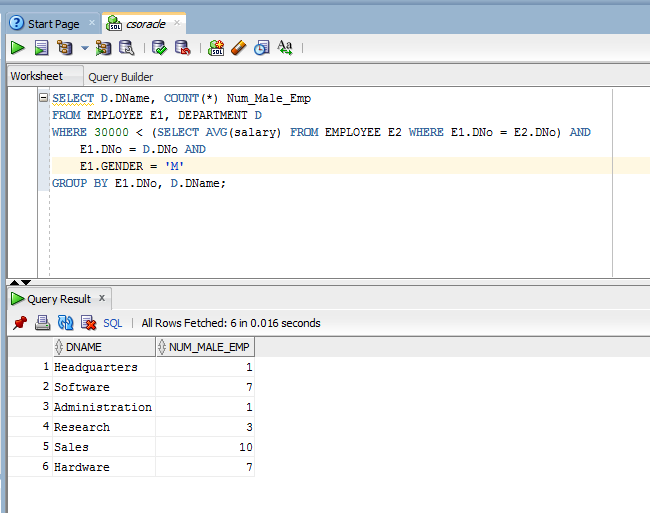
***\*\* There can be differences in query outputs,***

***because of different databases data values.***

***So, I have outputs based on the employee***

***database that I have.***

**b.** *for each dept with AVG\_EMP\_SALARY > 30000, deptName and number of MALE employees in it.*



SELECT D.DName,

COUNT(\*) Num\_Male\_Emp

FROM EMPLOYEE E1, DEPARTMENT D

WHERE 30000 < (

SELECT AVG(salary)

FROM EMPLOYEE E2

WHERE E1.DNo = E2.DNo) AND

E1.DNo = D.DNo AND

E1.GENDER = 'M'

GROUP BY E1.DNo, D.DName;

**c.** *for the department with highest emp salary, giving names of employees in it.*

SELECT E1.FName || ' ' || E1.LName Emp\_Name

FROM EMPLOYEE E1

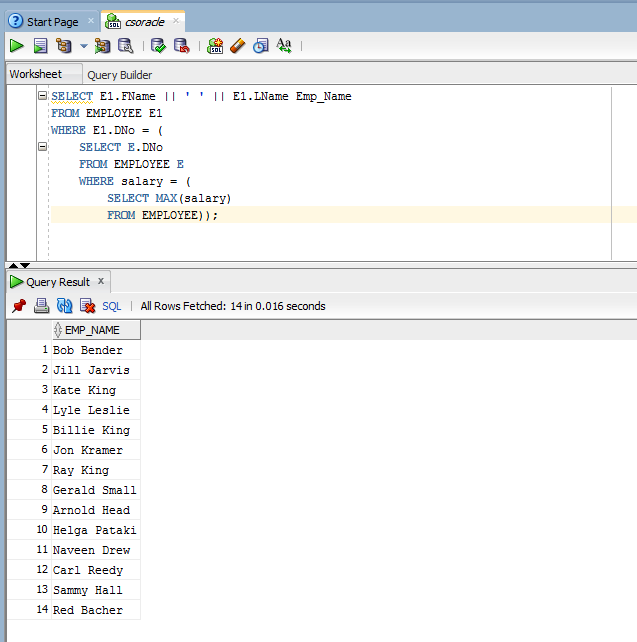
WHERE E1.DNo = (

SELECT E.DNo

FROM EMPLOYEE E

WHERE salary = ( SELECT MAX(salary)

FROM EMPLOYEE));

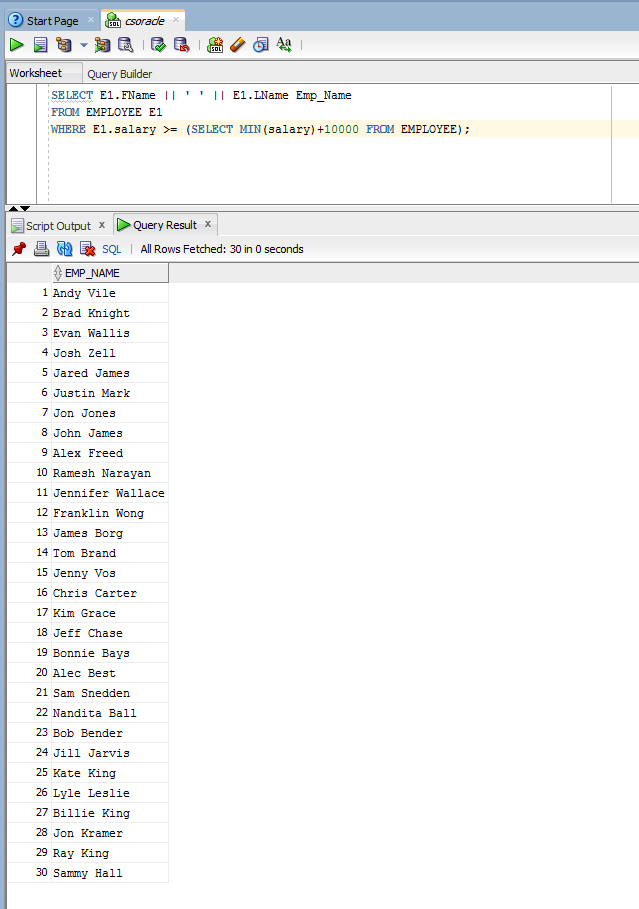


***\*\* There can be different total no of tuples from other's databases.***

***For e.g., I have 14 employees in Dno=8,***

***others may have 15, or so.***

***Kindly take a note of this.***

**d.** *employees making atleast 10000 more than the least salaried employee in the company.*

SELECT E1.FName || ' ' || E1.LName

Emp\_Name

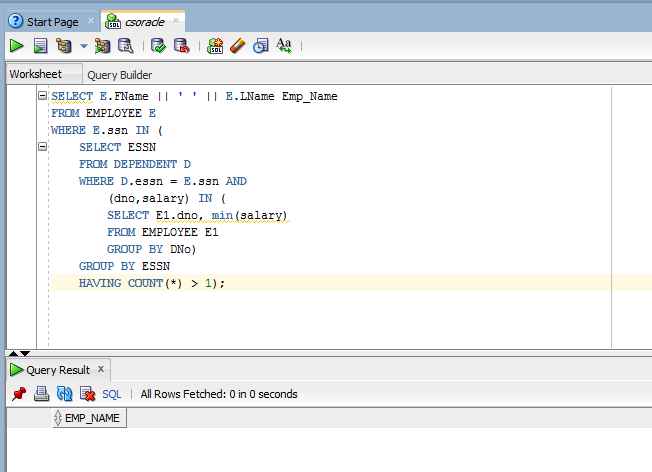
FROM EMPLOYEE E1

WHERE E1.salary >= (

SELECT MIN(salary)+10000

FROM EMPLOYEE);

**e.** *employees making least in their dept, with >1 dependent*



SELECT E.FName || ' ' || E.LName Emp\_Name

FROM EMPLOYEE E

WHERE E.ssn IN (

SELECT ESSN

FROM DEPENDENT D

WHERE D.essn = E.ssn AND

(dno,salary) IN (

SELECT E1.dno, min(salary)

FROM EMPLOYEE E1

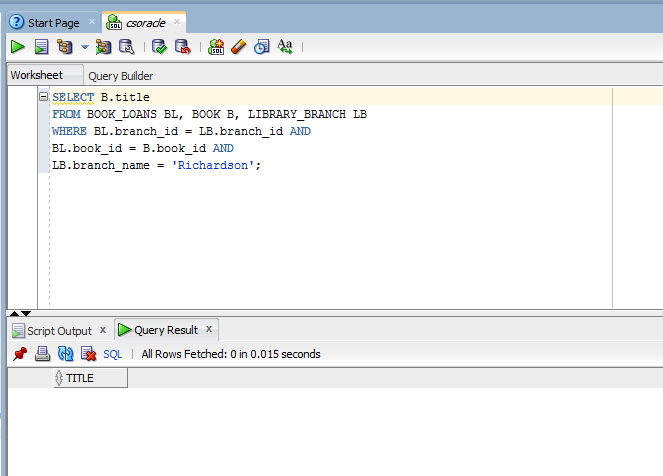
GROUP BY DNo)

GROUP BY ESSN

HAVING COUNT(\*) > 1);

**2. Queries on LIBRARY database.**

**a.** *books borrowed form 'Richardson' library*

****

SELECT B.title

FROM

BOOK\_LOANS BL, BOOK B,

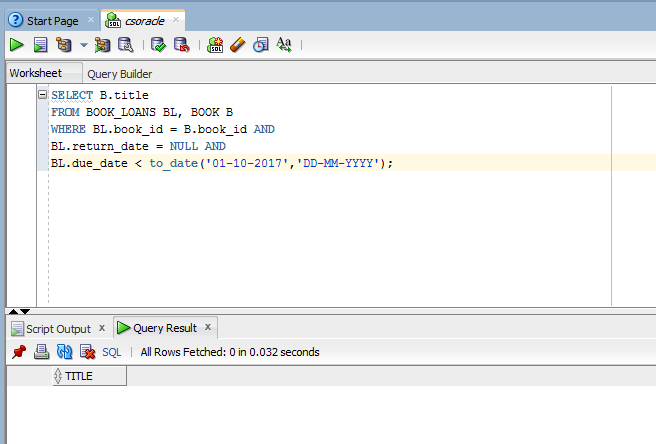
LIBRARY\_BRANCH LB

WHERE

BL.branch\_id = LB.branch\_id AND

BL.book\_id = B.book\_id AND

LB.branch\_name = 'Richardson';

**b.** *books that are overdue* 

SELECT B.title

FROM BOOK\_LOANS BL, BOOK B

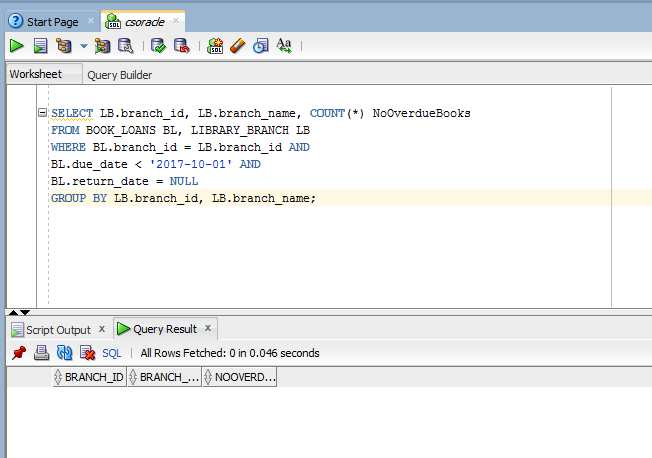
WHERE BL.book\_id = B.book\_id AND

BL.return\_date = NULL AND

BL.due\_date < to\_date(

'01-10-2017','DD-MM-YYYY');

**c.** *for each library branch, no of overdue books*

****

SELECT LB.branch\_id, LB.branch\_name,

COUNT(\*) NoOverdueBooks

FROM

BOOK\_LOANS BL, LIBRARY\_BRANCH LB

WHERE

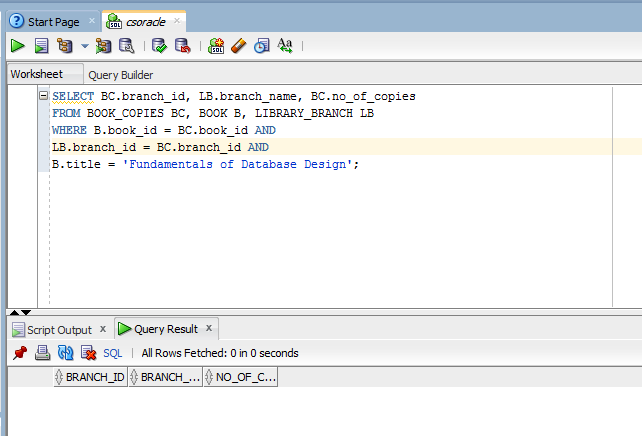
BL.branch\_id = LB.branch\_id AND

BL.due\_date < '2017-10-01' AND

BL.return\_date = NULL

GROUP BY LB.branch\_id, LB.branch\_name;

**d.** *no of copies of 'Fundamentals of Database Design' book, by each library branch*



SELECT BC.branch\_id,

LB.branch\_name, BC.no\_of\_copies

FROM BOOK\_COPIES BC, BOOK B,

LIBRARY\_BRANCH LB

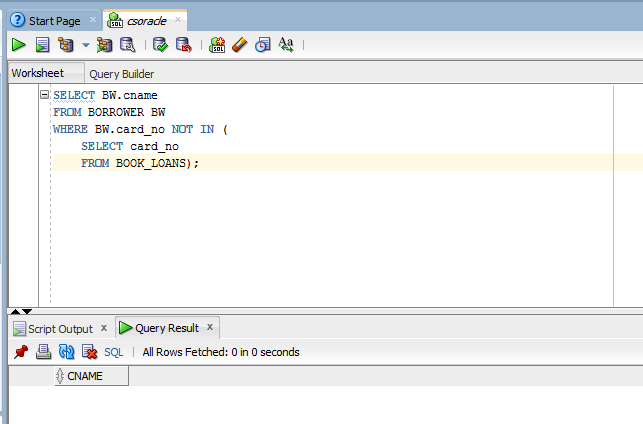
WHERE B.book\_id = BC.book\_id AND

LB.branch\_id = BC.branch\_id AND

B.title = 'Fundamentals of

Database Design';

**e.** *names of borrowers who hasn't checked out a book*



SELECT BW.cname

FROM BORROWER BW

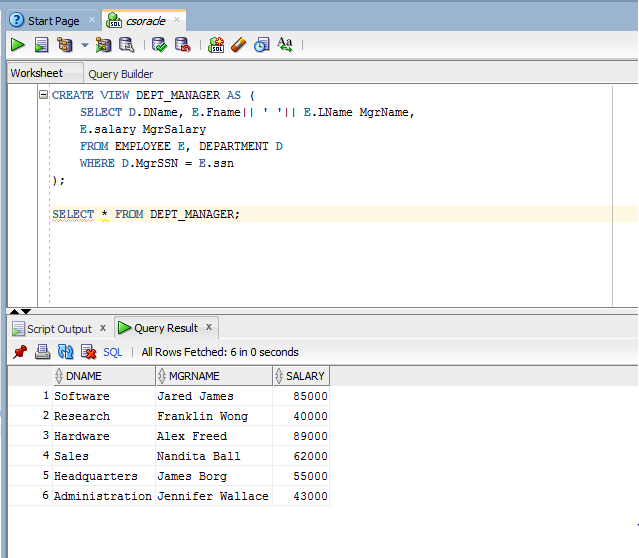
WHERE BW.card\_no NOT IN (

SELECT card\_no

FROM BOOK\_LOANS);

**3. Specifying VIEWS on COMPANY database using Correlated Nested Queries (except a.)**

**a.** *VIEW for each department, it's name, manager name and manager salary*

****

CREATE VIEW DEPT\_MANAGER AS (

SELECT D.DName,

E.Fname|| ' '|| E.LName MgrName,

E.salary MgrSalary

FROM

EMPLOYEE E, DEPARTMENT D

WHERE D.MgrSSN = E.ssn

);

SELECT \* FROM DEPT\_MANAGER;

**b.** *VIEW for each dept, it's name, mngr name, no of empl, no of projects*

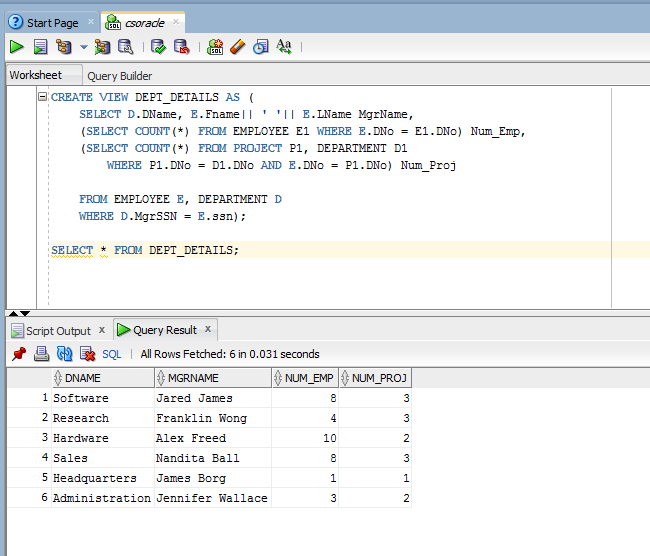
CREATE VIEW DEPT\_DETAILS AS (

SELECT D.DName, E.Fname|| ' '|| E.LName MgrName,

(SELECT COUNT(\*) FROM EMPLOYEE E1 WHERE E.DNo = E1.DNo) Num\_Emp,

(SELECT COUNT(\*) FROM PROJECT P1, DEPARTMENT D1

WHERE P1.DNo = D1.DNo AND E.DNo = P1.DNo) Num\_Proj



FROM

EMPLOYEE E, DEPARTMENT D

WHERE D.MgrSSN = E.ssn);

SELECT \* FROM DEPT\_DETAILS;

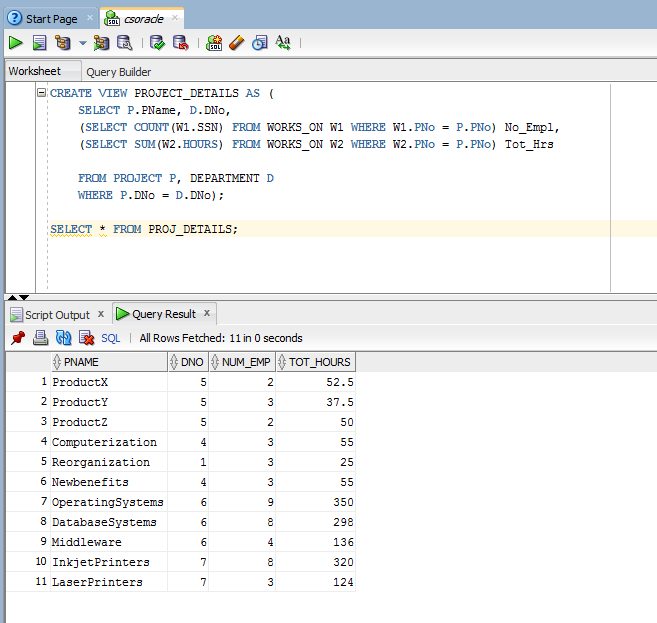
**c.** *VIEW for each project, its name, deptname, no of empl, total hrs per week*

CREATE VIEW PROJECT\_DETAILS AS (

SELECT P.PName, D.DNo,

(SELECT COUNT(W1.SSN) FROM WORKS\_ON W1 WHERE W1.PNo = P.PNo) No\_Empl,

(SELECT SUM(W2.HOURS) FROM WORKS\_ON W2 WHERE W2.PNo = P.PNo) Tot\_Hrs



FROM PROJECT P, DEPARTMENT D

WHERE P.DNo = D.DNo);

SELECT \* FROM PROJ\_DETAILS;

**d.** *VIEW for each project, its name, deptname, no of empl, total hrs per week, with > 1 empl*

CREATE VIEW PROJ\_DETAILS AS (

SELECT P.PName, D.DNo,

(SELECT COUNT(W1.SSN) FROM WORKS\_ON W1 WHERE W1.PNo = P.PNo) Num\_Emp,

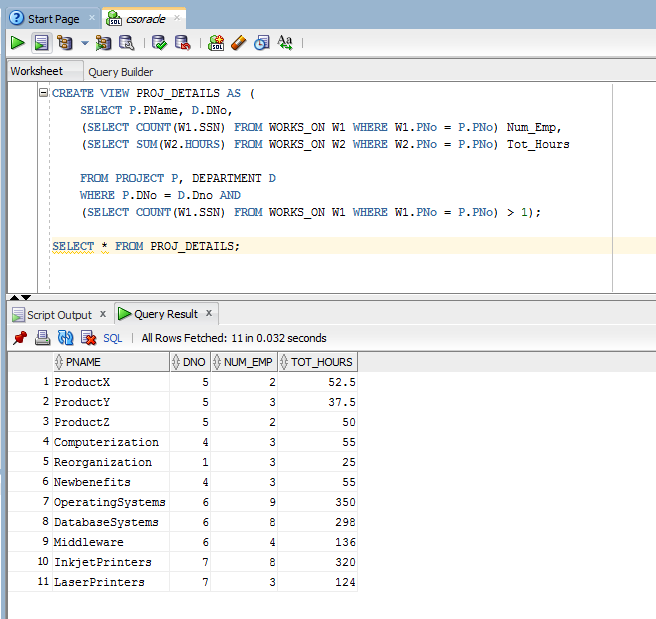
(SELECT SUM(W2.HOURS) FROM WORKS\_ON W2 WHERE W2.PNo = P.PNo) Tot\_Hours

FROM PROJECT P, DEPARTMENT D

WHERE P.DNo = D.Dno AND

(SELECT COUNT(W1.SSN) FROM WORKS\_ON W1 WHERE W1.PNo = P.PNo) > 1);

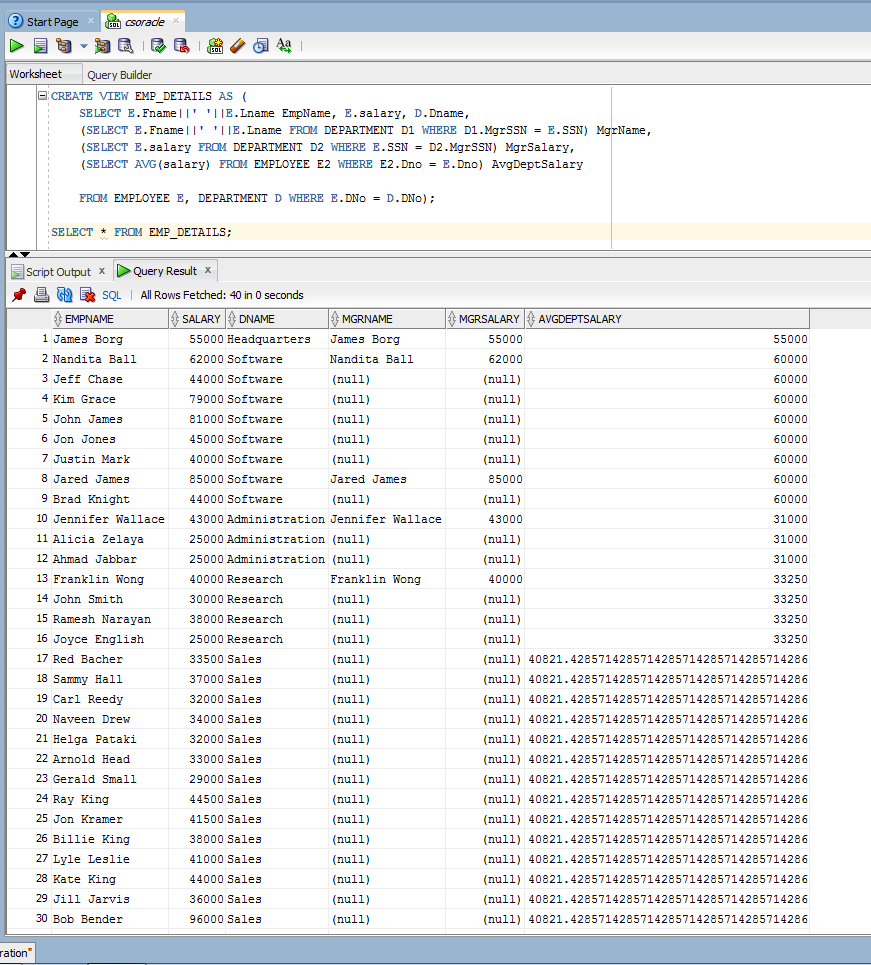
SELECT \* FROM PROJ\_DETAILS;



**e.** *VIEW for each employee, it's name, salary, dept, dept mngr, mngr salary and avg dept salary*

CREATE VIEW EMP\_DETAILS AS (

SELECT E.Fname||' '||E.Lname EmpName, E.salary, D.Dname,

(SELECT E.Fname||' '||E.Lname FROM DEPARTMENT D1 WHERE D1.MgrSSN = E.SSN) MgrName,

(SELECT E.salary

FROM DEPARTMENT D2

WHERE E.SSN = D2.MgrSSN) MgrSalary,

(SELECT AVG(salary)

FROM EMPLOYEE E2

WHERE E2.Dno = E.Dno) AvgDeptSalary

FROM

EMPLOYEE E, DEPARTMENT D

WHERE E.DNo = D.DNo);

SELECT \* FROM EMP\_DETAILS;