**Laboratory Exercise 8.1 - Exploring Oracle’s Authorization System**

1. If you have not already done so, create the Worker, Dept, Project and Assign tables in the database, as shown in Figure 5.3 in the textbook. You will find the DDL in the text file called *Fig\_5.3\_DDL&InsertsforWorkerDeptProjectAssign.txt* stored in this directory.

**\*\*\* I used my user RMART to start the assignment. That’s why it appears in some queries as the schema for tables.**

CREATE TABLE Worker (

empId NUMBER(6) PRIMARY KEY,

lastName VARCHAR2(20) NOT NULL,

firstName VARCHAR2(15) NOT NULL,

deptName VARCHAR2(15),

birthDate DATE,

hireDate DATE,

salary NUMBER(8,2));

CREATE TABLE Dept(

deptName VARCHAR2(15),

mgrId NUMBER(6),

CONSTRAINT Dept\_deptName\_pk PRIMARY KEY (deptName),

CONSTRAINT Dept\_mgrId\_fk FOREIGN KEY (mgrId) REFERENCES Worker(empId) ON DELETE SET NULL);

ALTER TABLE Worker ADD CONSTRAINT Worker\_deptName\_fk FOREIGN KEY (deptName) REFERENCES Dept(deptName) ON DELETE SET NULL;

CREATE TABLE Project (

projNo NUMBER(6),

projName VARCHAR2(20),

projMgrId NUMBER(6),

budget NUMBER (8,2),

startDate DATE,

expectedDurationWeeks NUMBER(4),

CONSTRAINT Project\_projNo\_pk PRIMARY KEY (projNo),

CONSTRAINT Project\_projMgrId\_fk FOREIGN KEY(projMgrId) REFERENCES WORKER(empId) ON DELETE SET NULL);

CREATE TABLE Assign (

projNo NUMBER(6),

empId NUMBER(6),

hoursAssigned NUMBER(3),

rating NUMBER(1),

CONSTRAINT Assign\_projNo\_empId\_pk PRIMARY KEY (projNo, empId),

CONSTRAINT Assign\_projNo\_fk FOREIGN KEY(projNo) REFERENCES Project(projNo) ON DELETE CASCADE,

CONSTRAINT Assign\_empId\_fk FOREIGN KEY(empId) REFERENCES Worker(empId) ON DELETE CASCADE);

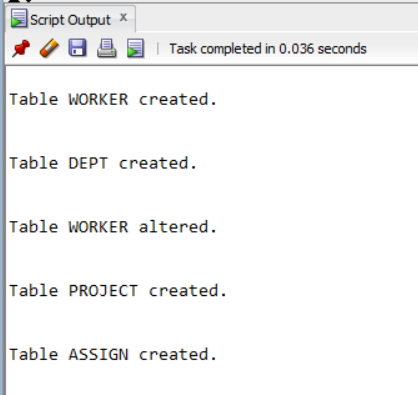


Figure 1

INSERT INTO Dept VALUES ('Accounting',null);

INSERT INTO Dept VALUES ('Research',null);

INSERT INTO Worker VALUES(101,'Smith','Tom', 'Accounting', '01-Feb-1970', '06-Jun-1993 ',50000);

INSERT INTO Worker VALUES(103,'Jones','Mary','Accounting', '15-Jun-1975', '20-Sep-2005',48000);

INSERT INTO Worker VALUES(105,'Burns','Jane','Accounting', '21-Sep-1980', '12-Jun-2000',39000);

INSERT INTO Worker VALUES(110,'Burns','Michael', 'Research', '05-Apr-1977', '10-Sep-2010',70000);

INSERT INTO Worker VALUES(115,'Chin','Amanda', 'Research', '22-Sep-1980', '19-Jun-2014',60000);

UPDATE Dept SET mgrId = 101 WHERE deptName = 'Accounting';

UPDATE Dept SET mgrId = 110 WHERE deptName = 'Research';

INSERT INTO Project VALUES (1001, 'Jupiter', 101, 300000, '01-Feb-2014', 50);

INSERT INTO Project VALUES (1005, 'Saturn', 101, 400000, '01-Jun-2014', 35);

INSERT INTO Project VALUES (1019, 'Mercury', 110, 350000, '15-Feb-2014', 40);

INSERT INTO Project VALUES (1025, 'Neptune', 110, 600000, '01-Feb-2015', 45);

INSERT INTO Project VALUES (1030, 'Pluto', 110, 380000, '15-Sep-2014', 50);

INSERT INTO Assign VALUES(1001, 101, 30,null);

INSERT INTO Assign VALUES(1001, 103, 20,5);

INSERT INTO Assign VALUES(1005, 103, 20,null);

INSERT INTO Assign VALUES(1001, 105, 30,null);

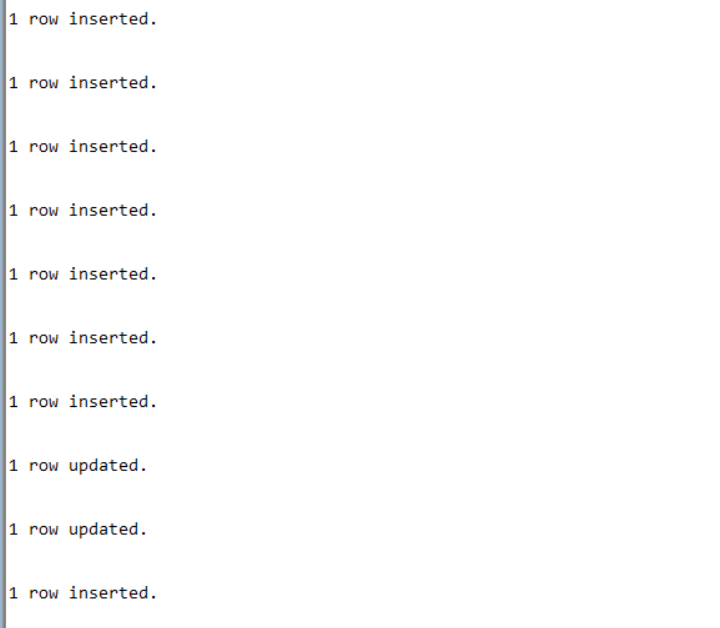
INSERT INTO Assign VALUES(1001, 115, 20,4);

INSERT INTO Assign VALUES(1019, 110, 20,5);

INSERT INTO Assign VALUES(1019, 115, 10,4);

INSERT INTO Assign VALUES(1025, 110, 10,null);

INSERT INTO Assign VALUES(1030, 110, 10,null);



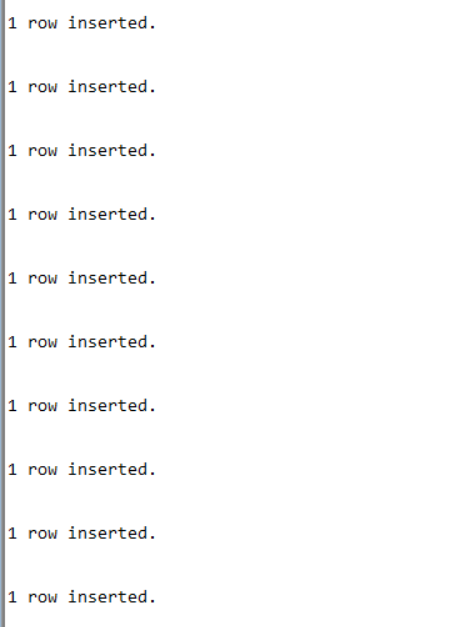


Figure 2

2. As you do each step, draw and modify an authorization graph by hand or by using a drawing tool, showing the privileges given.

a. Create five users: U100, U101, U102, U103, and U104.

**CREATE USER U100 IDENTIFIED BY 123456;**

**CREATE USER U101 IDENTIFIED BY 123456;**

**CREATE USER U102 IDENTIFIED BY 123456;**

**CREATE USER U103 IDENTIFIED BY 123456;**

**CREATE USER U104 IDENTIFIED BY 123456;**

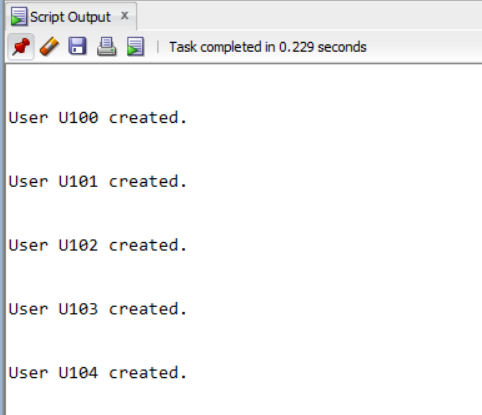


Figure 3



Figure 4

NOTE: In the figure 3, after the statements for creating users, it shows the confirmation of their creation. **In the figure 4 it shows the graph with the creation of users, but they haven’t received any privileges yet, that is why I didn’t include the line with arrow. They are connected to the database.**

b. Give user U100 the SELECT,INSERT,DELETE,UPDATE privileges on all four tables, with grant option.

**GRANT SELECT, INSERT, DELETE, UPDATE ON ASSIGN TO U100 WITH GRANT OPTION;**

**GRANT SELECT, INSERT, DELETE, UPDATE ON WORKER TO U100 WITH GRANT OPTION;**

**GRANT SELECT, INSERT, DELETE, UPDATE ON PROJECT TO U100 WITH GRANT OPTION;**

**GRANT SELECT, INSERT, DELETE, UPDATE ON DEPT TO U100 WITH GRANT OPTION;**

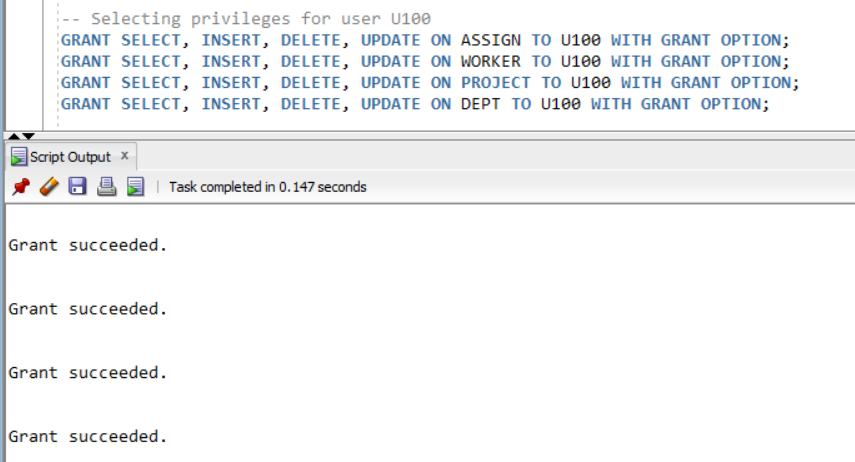


Figure 5



Figure 6

NOTE: In the figure 5, after statement for granting privileges for U100, it shows the confirmation after running the statement. **In the figure 6, it shows the authorization graph where the DBA gives privileges with grant option (doubled arrow) to U100 in all tables, which result in four nodes of U100 (one for each table) with solid line which means they have received grant option.**

c. Connect as U100 and pass the SELECT privilege on all four tables to U101 and U102, with grant option.

**GRANT SELECT**

**ON RMART.DEPT**

**TO U101, U102**

**WITH GRANT OPTION;**

**GRANT SELECT**

**ON RMART.WORKER**

**TO U101, U102**

**WITH GRANT OPTION;**

**GRANT SELECT**

**ON RMART.ASSIGN**

**TO U101, U102**

**WITH GRANT OPTION;**

**GRANT SELECT**

**ON RMART.PROJECT**

**TO U101, U102**

**WITH GRANT OPTION;**

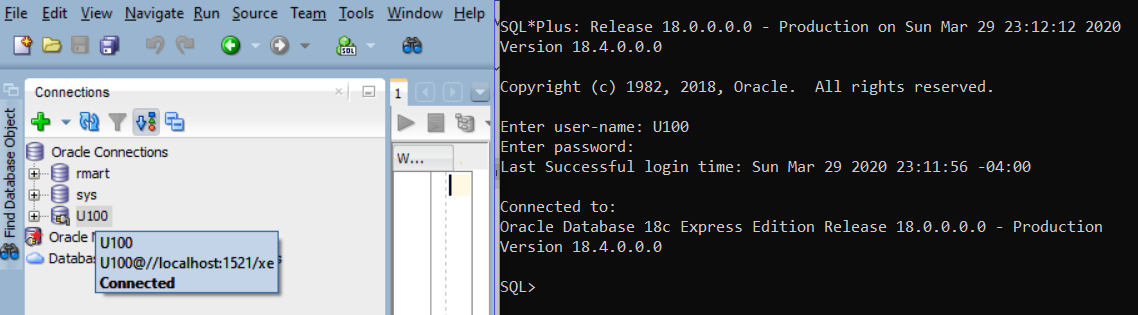


Figure 7

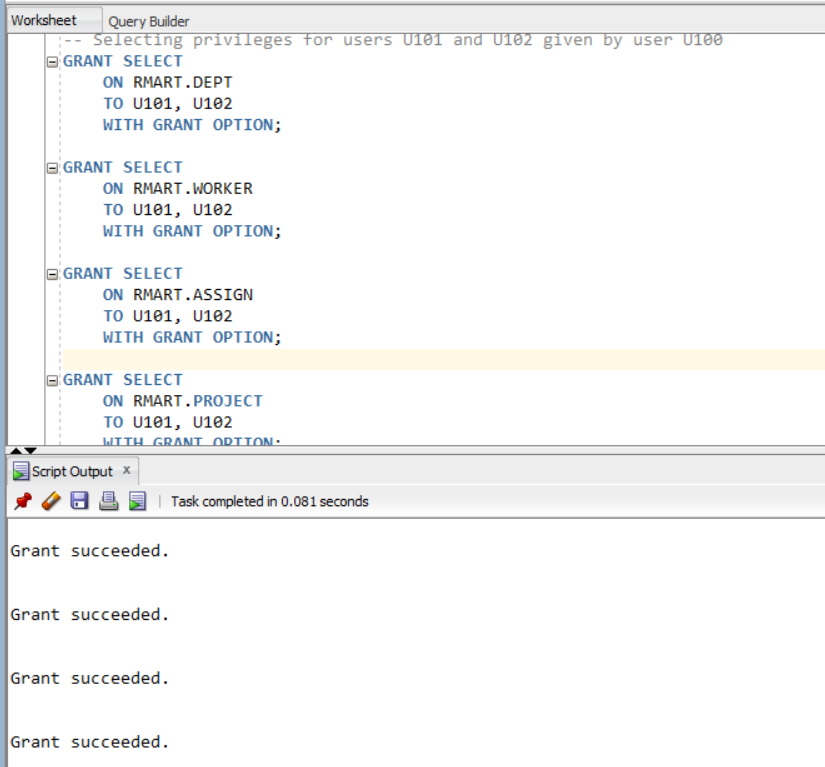


Figure 8



Figure 9

NOTE: **I first used prompt command to connect to U100 then I went back to oracle sql developer.** In figure 7 and 8, after statements of granting privileges to U101 and U102 given by U100, it shows the connection of U100 and the granting statement given by U100 to U101 and U102. **In the figure 9, it shows the authorization graph where the DBA gives privileges with grant option to U100 as shown before in figure 7, and the U100 gives privilege select on all four tables with grant option (doubled arrow) to U101 and U102, which result in four nodes of U101 and U102 (one for each table) that appear in solid line which means they have received grant option.**

d. Still acting as U100, pass the SELECT privilege on Worker to U103 and U104, without the grant option.

**GRANT SELECT**

**ON RMART.WORKER**

**TO U103, U104;**

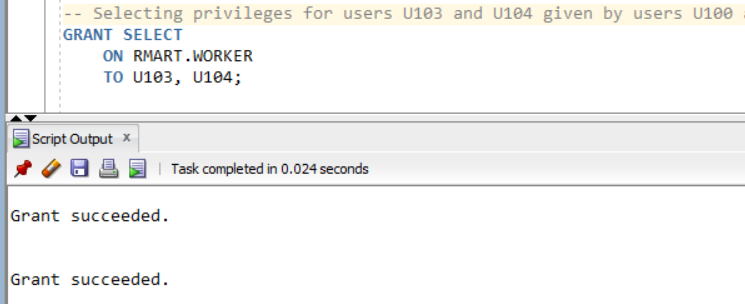


Figure 10



Figure 11

NOTE: In figure 10, it shows the granting statements given by U100 to U103 and U104. In the figure 11 it shows the authorization graph where the DBA gives privileges with grant option to U100 as shown before in figure 7, the U100 gives privileges with grant option to U101 and U102 as shown in figure 9, and **now it shows also that the U100 gives privilege select on worker without grant option (single arrow) to U103 and U104, which result in one node of U103 and U104 (only table worker) that appear in dashed line which means they have not received grant option.**

e. Connect as U101 and pass the SELECT privilege on Worker to U103 and U104, without the grant option.

**GRANT SELECT**

**ON RMART.WORKER**

**TO U103, U104;**

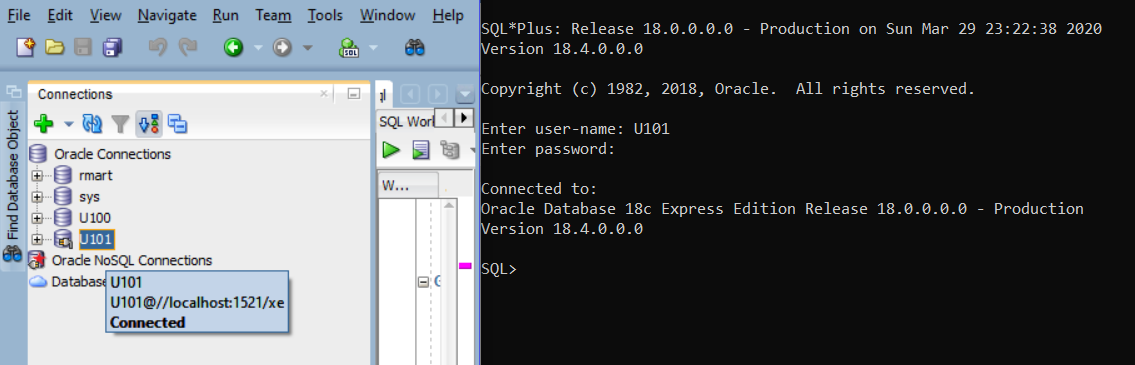


Figure 12

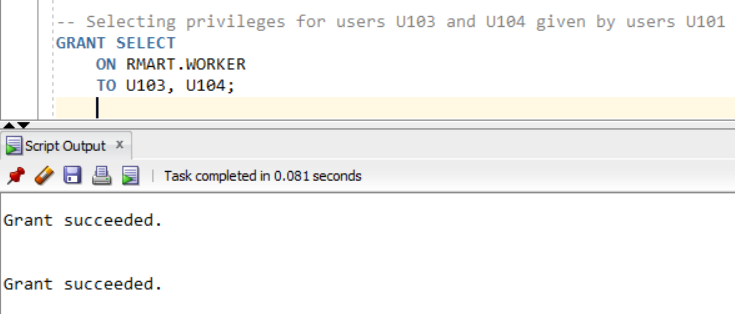


Figure 13



Figure 14

NOTE: In figure 12 and 13 it shows the connection of U101 and the granting statements given by U101 to U103 and U104. In the figure 14 it shows the authorization graph where the DBA gives privileges with grant option to U100 as shown before in figure 7, the U100 gives privileges to U101 and U102 with grant option as shown in figure 9 and to U103 and U104 without grant option as shown in figure 11; **now it shows also that the U101 gives privilege select on work table without grant option (single arrow) to U103 and U104 which result in one node of U103 and U104 (only table worker) that appear in dashed line which means they have not received grant option.**

f. Connect as U100 and revoke all privileges you granted to U101.

**REVOKE SELECT**

**ON RMART.DEPT**

**FROM U101;**

**REVOKE SELECT**

**ON RMART.WORKER**

**FROM U101;**

**REVOKE SELECT**

**ON RMART.ASSIGN**

**FROM U101;**

**REVOKE SELECT**

**ON RMART.PROJECT**

**FROM U101;**

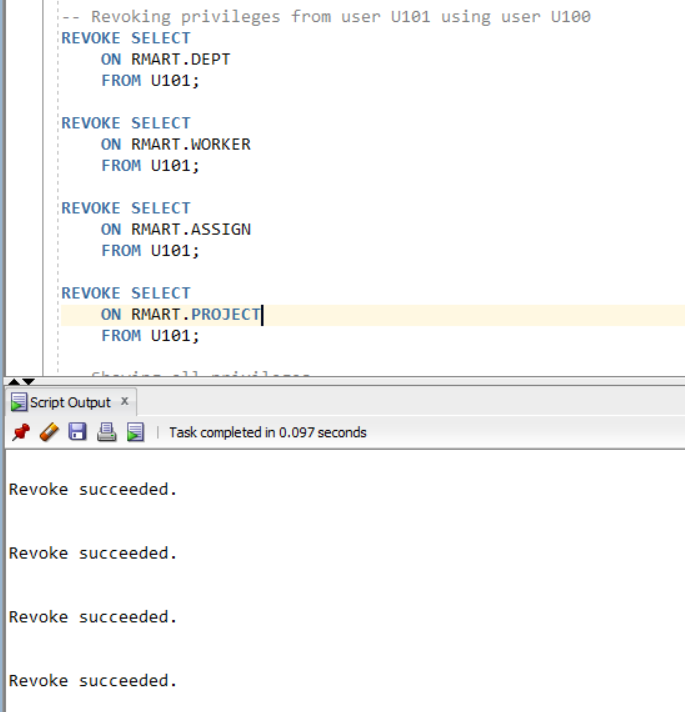


Figure 15



Figure 16

NOTE: In figure 15, it shows the connection of U100 and the revoke statement from U101. In the figure 16 it shows the authorization graph updated where the DBA gives privileges with grant option to U100 as shown before in figure 7, the U100 gives privileges to U101 and U102 with grant option as shown in figure 9 and to U103 and U104 without grant option as shown in figure 11; **now it shows the U101 nodes without any privileges since U100 revoked their privileges, and also it shows that U103 and U104 don’t have the privilege given by U101 anymore, since U101 don’t have any privileges to grant them after U100 revoke its privileges.**

g. From the authorization graph, determine what privileges, if any, the remaining users should still have.

As you can see in figure 17 bellow, each user has the following privileges:

- U100:

a) select, insert, delete, update ON **dept** with grant option

b) select, insert, delete, update ON **worker** with grant option

c) select, insert, delete, update ON **project** with grant option

d) select, insert, delete, update ON **assign** with grant option

- U101: **none**

- U102:

a) select ON **dept** with grant option

b) select ON **worker** with grant option

c) select ON **project** with grant option

d) select ON **assign** with grant option

- U103: select ON **worker** without grant option

- U104: select ON **worker** without grant option



Figure 17

h. Confirm these privileges by selecting the table privileges for each user from the sys.dba\_tab\_privs data dictionary view.

**SELECT**

**GRANTEE,**

**GRANTOR,**

**PRIVILEGE,**

**table\_name,**

**GRANTABLE**

**FROM DBA\_TAB\_PRIVS**

**WHERE grantee IN ('U100', 'U101', 'U102', 'U103', 'U104')**

**ORDER BY grantee;**

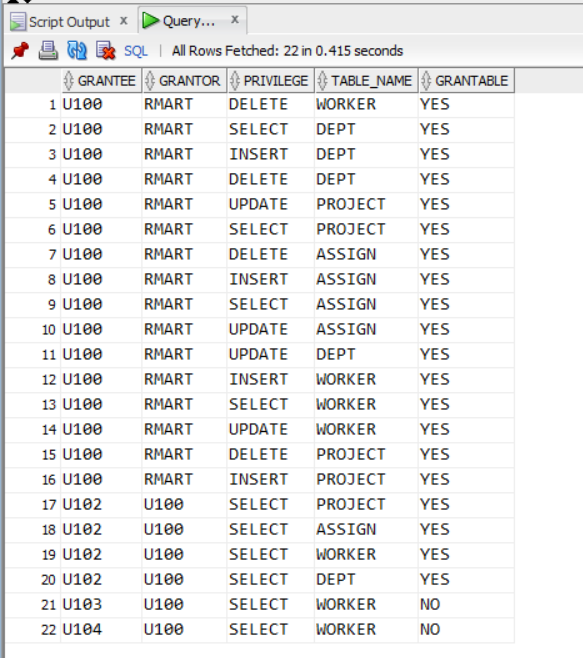


Figure 18