Islamic University of Technology

Lab 05

CSE 4308 - DBMS Lab

Submitted To :

Zannatun Naim Sristy
Lecturer, CSE Department
Islamic University of Technology

Submitted By :

Rhidwan Rashid
ID : 200042149
Prog. : SWE
Dept. : CSE

Queries:

(1) Analysis: The problem requires creating a view named Advisor_Selection that shows the ID, name and department name of the instructors.

Working code:

```
CREATE OR REPLACE VIEW ADVISOR_SELECTION
AS
SELECT ID, NAME, DEPT_NAME
FROM INSTRUCTOR;
```

Explanation: With "create or replace view" we can create a view, after the keyword the view name is given, then after "as" keyword we select attributes of the view from another table.

(2) Analysis: The problem requires creating another view named Student_Count using Advisor_Selection(the view we created before) to show the name of instructors and number of students assigned under them

Working code:

```
CREATE OR REPLACE VIEW STUDENT_COUNT
AS
SELECT NAME,COUNT(S_ID) AS STUDENT_COUNT
FROM ADVISOR_SELECTION NATURAL JOIN ADVISOR
GROUP BY ADVISOR_SELECTION.NAME;
```

Explanation: Here we created a view same as task 1. The differences are an aggregate function(Count) was used to count number of students, natural join of two tables occurred and "group by" was used.

(3)Analysis: This problem requires creating roles and granting them appropriate privileges.

(a) Working code for Student role:

```
CREATE ROLE C##STUDENT1;
GRANT SELECT ON ADVISOR TO C##STUDENT1;
GRANT SELECT ON COURSE TO C##STUDENT1;
```

(b) Working code for course teacher role:

```
CREATE ROLE C##COURSE_TEACHER;
GRANT SELECT ON STUDENT TO C##COURSE_TEACHER;
GRANT SELECT ON COURSE TO C##COURSE_TEACHER;
```

(c) Working code for head of departments role:

```
CREATE ROLE C##HEAD_OF_DEPT;
GRANT COURSE_TEACHER TO C##HEAD_OF_DEPT;
GRANT INSERT ON INSTRUCTOR TO C##HEAD_OF_DEPT;
```

(d) Working code for administrator:

```
CREATE ROLE C##ADMINISTRATOR;
GRANT SELECT ON DEPARTMENT TO C##ADMINISTRATOR;
GRANT SELECT ON INSTRUCTOR TO C##ADMINISTRATOR;
GRANT UPDATE(BUDGET) ON DEPARTMENT TO C##ADMINISTRATOR;
```

Explanation: To create role we have to use "Create role" keyword, then role name. Then use "grant" keyword and privilege name(select, update, delete) on "table name" to the role we created. (a), (b) and (d) can be done using this method. We can also grant a role to another newly created role to give same privileges as role1. (c) was done like this.

(4) Analysis: Now we have to test the roles we have created using SQL queries.

Working code:

```
CREATE USER C##STDNT IDENTIFIED BY 1234;
CREATE USER C##TEACHER IDENTIFIED BY 1234;
CREATE USER C##HEAD IDENTIFIED BY 1234;
CREATE USER C##MINISTER IDENTIFIED BY 1234;

GRANT C##STUDENT1 TO C##STDNT;
GRANT C##COURSE_TEACHER TO C##TEACHER;
GRANT C##HEAD_OF_DEPT TO C##HEAD;
GRANT C##ADMINISTRATOR TO C##MINISTER;

GRANT CREATE SESSION TO C##STDNT;
GRANT CREATE SESSION TO C##TEACHER;
GRANT CREATE SESSION TO C##HEAD;
GRANT CREATE SESSION TO C##MINISTER;

-->STUDENT USER<--
CONN C##STDNT/1234;

SELECT * FROM SYSTEM.COURSE;

SELECT * FROM SYSTEM.DEPARTMENT;

-->COURSE TEACHER<--

CONN C##TEACHER/1234;

SELECT * FROM SYSTEM.COURSE;

SELECT * FROM SYSTEM.DEPARTMENT;

-->DEPARTMENT HEAD<--

CONN C##HEAD/1234
```

```
INSERT INTO INSTRUCTOR VALUE('1234', 'NONE', 'CSE', '200');

insert into prereq values ('EF-181', 'PHY-104');

-->ADMINISTRATOR<--
CONN C##MINISTER

UPDATE SYSTEM.DEPARTMENT
SET BUDGET = 12000
WHERE DEPT_NAME = 'History';

SELECT * FROM TEACHES;
```

Explanation: Created users are granted roles. They can do certain operations on tables based on their roles. For example, a user who have student role assigned to him/her can only view courses and advisors table. If they try to view other like departments the query will give an error. Same goes for other users who are granted different roles.