



Islamic University of Technology (IUT)

## Report on Lab 05

Submitted By

Shanta Maria (ID:200042172)

CSE 4308 Database Management Systems Lab

Submitted To

Md. Bakhtiar Hasan

Lecturer, Department of CSE

Zannatun Naim Srsity

Lecturer, Department of CSE

October 9, 2022

# Introduction

In the lab class, we were given four tasks to solve using SQL command line to understand advanced data definition and data manipulation using views and roles.

## 1 Task 1

Create a view named `Advisor_Selection` that shows the ID, name and department name of instructors.

### 1.1 Solution

```
---task01---  
  
create or replace view Advisor_Selection  
as  
  
select id, name, dept_name  
from instructor;
```

### 1.2 Analysis and Explanation

This task was completed easily by following the instructions in the PDF document.

### 1.3 Difficulties

I did not face any difficulties when doing this task and no mentionable issues were encountered.

```
SQL> ---task01---
SQL> create or replace view Advisor_Selection
2  as
3  select id, name, dept_name
4  from instructor;

View created.
```

Figure 1: Task 1

## 2 Task 2

Create another view named `Student_Count` using `Advisor_Selection` and `advisor` to show the name of the instructor and the number students assigned under them.

### 2.1 Solution

```
---task02---

create or replace view Student_Count
as
select max(Advisor_Selection.name) as advisor_name,
count(advisor.s_id) as student_count
from Advisor_Selection
left join advisor on Advisor_Selection.id=advisor.i_id
group by Advisor_Selection.id;
```

### 2.2 Analysis and Explanation

This task was also completed easily by following the instructions in the PDF document.

## 2.3 Difficulties

I did face some difficulties when using group by clause since we are grouping by `Advisor_Selection.id` here but it is not present in the select clause. So I used an aggregate function like `max()` for the attributes remaining in the select clause which will not alter the value of the attribute.

```
SQL> ---task02---
SQL> create or replace view Student_Count
 2 as
 3 select max(Advisor_Selection.name) as advisor_name, count(advisor.s_id) as student_count
 4 from Advisor_Selection
 5 left join advisor on Advisor_Selection.id=advisor.i_id
 6 group by Advisor_Selection.id;

View created.
```

Figure 2: Task 2

## 3 Task 3

Four categories of users have been identified in the database:

- (a) Students should be able to view information regarding advisors and courses.
- (b) Course teachers should be able to view information about the students and courses.
- (c) Head of the Departments should have all the privileges that a course teacher has. Additionally, s/he should be able to add new instructors.
- (d) Administrator should be able to see information about department and instructors. They should also be able to update the department budget.

Create roles for these categories and grant them appropriate privileges.

### 3.1 Solution

```
---task03---
```

```
---task03(a)---
```

```
drop role student_role;
```

```
create role student_role;
```

```

grant create session to student_role;
grant select on course to student_role;
grant select on Advisor_Selection to student_role;

---task03(b)---
drop role course_teacher;
create role course_teacher;
grant create session to course_teacher;
grant select on course to course_teacher;
grant select on student to course_teacher;

---task03(c)---
drop role head_dept;
create role head_dept;
grant create session to head_dept;
grant course_teacher to head_dept;
grant select on instructor to head_dept;
grant insert on instructor to head_dept;

---task03(d)---
drop role administrator;
create role administrator;
grant create session to administrator;
grant select on department to administrator;
grant select on instructor to administrator;
grant update(budget) on department to administrator;

```

## 3.2 Analysis and Explanation

This problem was easy to solve by following the instructions in the PDF document.

For (a) and (b), I granted the `create session` and `select` commands to the roles for connecting to the user identified by that role and viewing tables. The `select` command was granted for the required tables only.

For(c), I granted the same commands as the ones granted to the `course_teacher` role by granting that role to the `head_dept` role. I also granted the `insert` role only for the `instructor` table to this role. The `select` command was granted for the required tables only.

For(d), I granted the `create session` and `select` commands to the role for connecting to the user identified by that role and viewing tables. The `select` command was granted for the required tables only. I also granted the `update` command but only for the `budget` attribute in the `department` table.

## 3.3 Difficulties

I did not face any difficulties when doing this task and no mentionable issues were encountered.

```
SQL> ---task03---
SQL> ---task03(a)---
SQL> drop role student_role;

Role dropped.

SQL> create role student_role;

Role created.

SQL> grant create session to student_role;

Grant succeeded.

SQL> grant select on course to student_role;

Grant succeeded.

SQL> grant select on Advisor_Selection to student_role;

Grant succeeded.
```

Figure 3: Task 3(a)

```
SQL>
SQL> ---task03(b)---
SQL> drop role course_teacher;

Role dropped.

SQL> create role course_teacher;

Role created.

SQL> grant create session to course_teacher;

Grant succeeded.

SQL> grant select on course to course_teacher;

Grant succeeded.

SQL> grant select on student to course_teacher;

Grant succeeded.
```

Figure 4: Task 3(b)

```
SQL>
SQL> ---task03(c)---
SQL> drop role head_dept;

Role dropped.

SQL> create role head_dept;

Role created.

SQL> grant create session to head_dept;

Grant succeeded.

SQL> grant course_teacher to head_dept;

Grant succeeded.

SQL> grant select on instructor to head_dept;

Grant succeeded.

SQL> grant insert on instructor to head_dept;

Grant succeeded.
```

Figure 5: Task 3(c)

```
SQL>
SQL> ---task03(d)---
SQL> drop role administrator;

Role dropped.

SQL> create role administrator;

Role created.

SQL> grant create session to administrator;

Grant succeeded.

SQL> grant select on department to administrator;

Grant succeeded.

SQL> grant select on instructor to administrator;

Grant succeeded.

SQL> grant update(budget) on department to administrator;

Grant succeeded.
```

Figure 6: Task 3(d)



## 4 Task 4

Create users under these roles and write relevant SQL queries to demonstrate that the imposed access control works.

### 4.1 Solution

```
---task04---  
---demonstrating task03(a)---  
drop user s1;  
create user s1 identified by ps1;  
grant student_role to s1;  
conn s1/ps1;  
select * from swe200042172.Advisor_Selection;  
select * from swe200042172.course;  
drop table swe200042172.course;  
--will give insufficient privileges error--  
conn swe200042172/cse4308  
  
---demonstrating task03(b)---  
drop user c1;  
create user c1 identified by ps2;  
grant course_teacher to c1;  
conn c1/ps2;  
select * from swe200042172.student;  
select * from swe200042172.course;  
drop table swe200042172.course;  
--will give insufficient privileges error--
```

```

conn swe200042172/cse4308

---demonstrating task03(c)---

drop user h1;

create user h1 identified by ps3;

grant head_dept to h1;

conn h1/ps3;

select * from swe200042172.student;

select * from swe200042172.course;

insert into swe200042172.instructor values
('24172', 'Maria', 'Music', '456700');

drop table swe200042172.course;

--will give insufficient privileges error--

conn swe200042172/cse4308

---demonstrating task03(d)---

drop user a1;

create user a1 identified by ps4;

grant administrator to a1;

conn a1/ps4;

select * from swe200042172.department;

select * from swe200042172.instructor;

update swe200042172.department set
budget='150000' where dept_name='Music';

drop table swe200042172.department;

--will give insufficient privileges error--

conn swe200042172/cse4308

```

## 4.2 Analysis and Explanation

For this task, I created 4 new users then granted each user to a one of the four roles from task 3 and tested the privileges granted to each role using SQL queries. To check for permissions not granted to the roles, I used drop to try and drop the tables.

## 4.3 Difficulties

I faced some difficulty when using the insert command. The structure of the inserting values had to match the attribute constrictions of the table and the dept\_name had to be an existing one from department table which I did not understand at first and faced errors.

```
SQL> ---task04---
SQL> ---demonstrating task03(a)---
SQL> drop user s1;

User dropped.

SQL> create user s1 identified by ps1;

User created.

SQL> grant student_role to s1;

Grant succeeded.

SQL> conn s1/ps1;

Connected.
```

Figure 7: Task 4(a)-1

```
SQL> select * from swe200042172.Advisor_Selection;
```

ID	NAME	DEPT_NAME
10101	Srinivasan	Comp. Sci.
12121	Wu	Finance
15151	Mozart	Music
22222	Einstein	Physics
32343	El Said	History
33456	Gold	Physics
45565	Katz	Comp. Sci.
58583	Califieri	History
76543	Singh	Finance
76766	Crick	Biology
83821	Brandt	Comp. Sci.
-----		
ID	NAME	DEPT_NAME
98345	Kim	Elec. Eng.

12 rows selected.

Figure 8: Task 4(a)-2

```
SQL> drop table swe200042172.course;
drop table swe200042172.course
      *
ERROR at line 1:
ORA-01031: insufficient privileges

SQL> --will give insufficient privileges error--
SQL> conn swe200042172/cse4308
Connected.
```

Figure 9: Task 4(a)-3

```
SQL>
SQL> ---demonstrating task03(b)---
SQL> drop user c1;

User dropped.

SQL> create user c1 identified by ps2;

User created.

SQL> grant course_teacher to c1;

Grant succeeded.

SQL> conn c1/ps2;
Connected.
```

Figure 10: Task 4(b)-1

```
SQL> select * from swe200042172.student;
```

ID	NAME	DEPT_NAME	TOT_CRED
00128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp. Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp. Sci.	58
76653	Aoi	Elec. Eng.	60
-----			
ID	NAME	DEPT_NAME	TOT_CRED
98765	Bourikas	Elec. Eng.	98
98988	Tanaka	Biology	120

13 rows selected.

Figure 11: Task 4(b)-2

```
SQL> drop table swe200042172.course;
drop table swe200042172.course
      *
ERROR at line 1:
ORA-01031: insufficient privileges

SQL> --will give insufficient privileges error--
SQL> conn swe200042172/cse4308
Connected.
```

Figure 12: Task 4(b)-3

```

SQL>
SQL> ---demonstrating task03(c)---
SQL> drop user h1;

User dropped.

SQL> create user h1 identified by ps3;

User created.

SQL> grant head_dept to h1;

Grant succeeded.

SQL> conn h1/ps3;
Connected.

```

Figure 13: Task 4(c)-1

```

SQL> insert into swe200042172.instructor values ('24172', 'Maria', 'Music', '456780');
1 row created.

```

Figure 14: Task 4(c)-2

```

SQL> drop table swe200042172.course;
drop table swe200042172.course
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL> --will give insufficient privileges error--
SQL> conn swe200042172/cse4308
Connected.

```

Figure 15: Task 4(c)-3

## Conclusion

As shown in the report, I have solved and tested the solutions for all four tasks given in the lab. All the commands used were written in notepad which was then saved with .sql extension. The .sql file was then run through the SQL command line to execute all the commands after executing the given DDL+drop.sql and smallRelationsInsertFile.sql files.