|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | |
|  | Report on Lab-06  DATABASE MANAGEMENT SYSTEMS LAB | | | | | |  | |
|  |  | | | | | | |  | | |
| Submitted by  Adid-Al-Mahamud Shazid  Student Id: 210042172  Department: CSE  Programme: SWE  Course Title: CSE 4308  Submitted to  Zannatun Naim Sristy  Lecturer, Department of CSE  September 21, 2023 | |  | | | | |
|  |
|  | | |  |
|  | |  | | | |  | | | |

**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.

**Task 1**

Create a view named Advisor\_Selection that shows the ID, name and department name of instructors.

* 1. **Solution**

create or replace view advisor\_Selection as

select ID, name, dept\_name

from instructor;

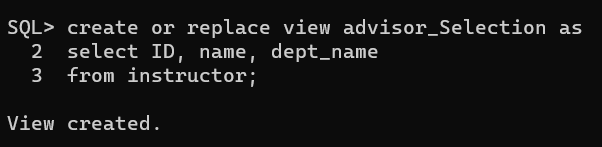
* 1. **Analysis and Explanation**

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

* 1. **Difficulties**

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

* 1. **Output**

****

**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.

* 1. **Solution**

create or replace view student\_Selection 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;

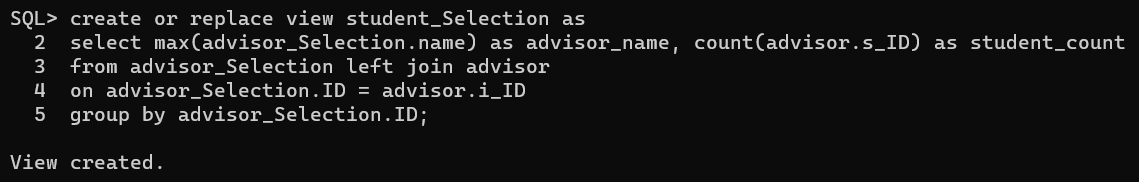
* 1. **Analysis and Explanation**

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

* 1. **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.

* 1. **Output**

****

**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.

* 1. **Solution**

--a--

drop role student\_role;

create role student\_role;

grant create session, resource, create tablespace to student\_role;

grant select on course to student\_role;

grant select on advisor to student\_role;

--b--

drop role course\_teacher;

create role course\_teacher;

grant create session, resource, create tablespace to course\_teacher;

grant select on course to course\_teacher;

grant select on student to course\_teacher;

--c--

drop role head\_dept;

create role head\_dept;

grant create session, resource, create tablespace to head\_dept;

grant course\_teacher to head\_dept;

grant select on instructor to head\_dept;

grant insert on instructor to head\_dept;

--d--

drop role administrator;

create role administrator;

grant create session, resource, create tablespace to administrator;

grant select on department to administrator;

grant select on instructor to administrator;

grant update(budget) on department to administrator;

* 1. **Analysis and Explanation**

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

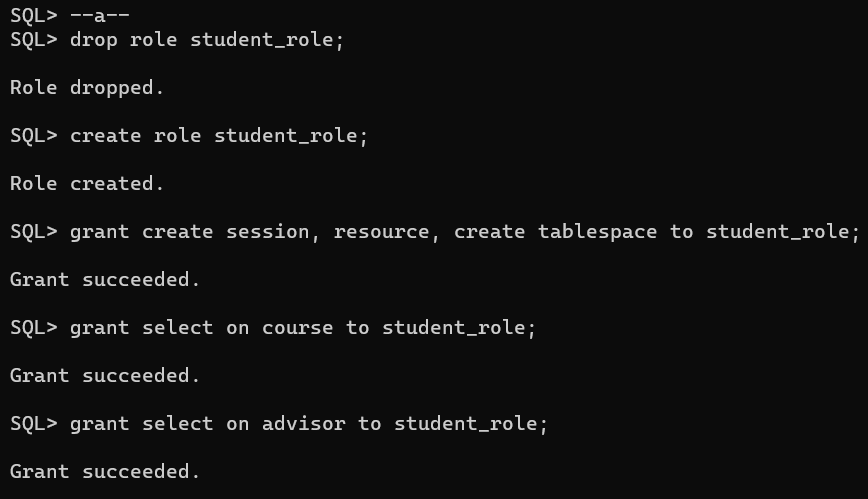
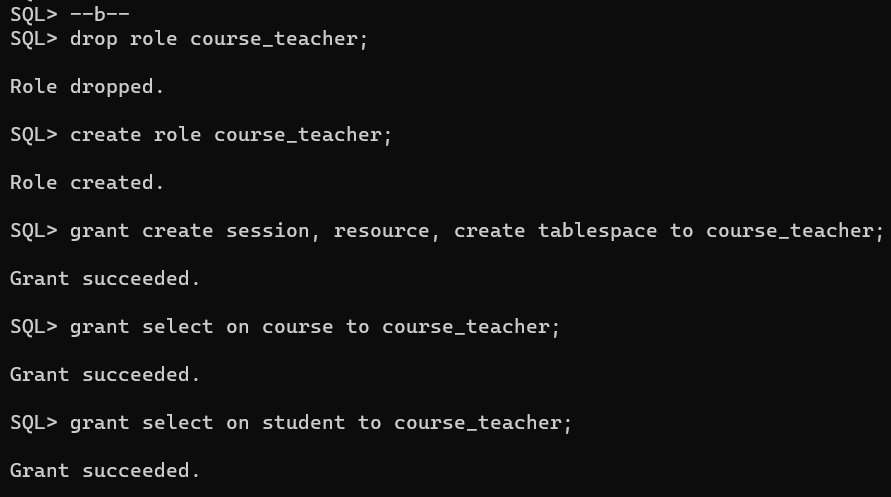
For (a) and (b), I granted the create session and select commands to the roles for connecting to the user identified by that role and viewing tables. The select command was granted for the required tables only.

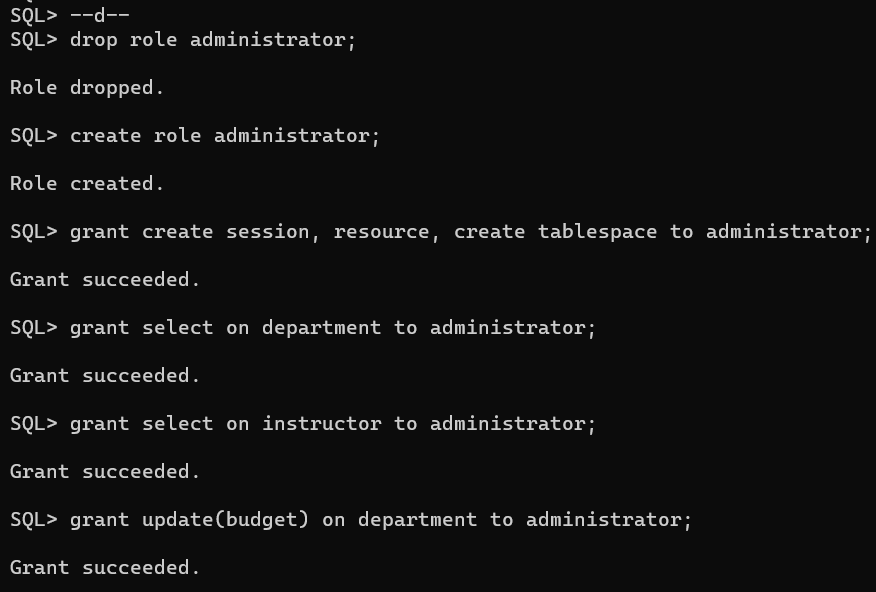
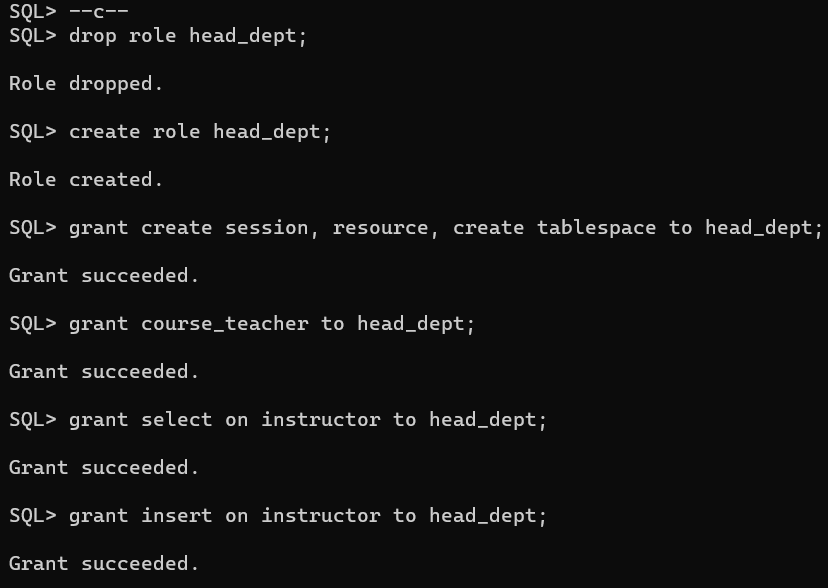
For(c), I granted the same commands as the ones granted to the course\_teacher role by granting that role to the head\_dept role. I also granted the insert role only for the instructor table to this role. The select command was granted for the required tables only.

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.

* 1. **Difficulties**

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

* 1. **Output**

****

**Task 4**

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

* 1. **Solution**

--a--

create user student1 identified by ps1;

grant student\_role to student1;

connect student1/ps1;

select \* from System.advisor;

select \* from System.course;

drop table System.course; --Won't work

--b--

create user teacher1 identified by ps1;

grant course\_teacher to teacher1;

connect teacher1/ps1;

select \* from System.student;

select \* from System.course;

drop table System.course; --Won't work

--c--

create user head1 identified by ps1;

grant head\_dept to head1;

connect head1/ps1;

select \* from System.student;

select \* from System.course;

insert into System.instructor values ('21172', 'Adid', 'Math', '456700');

drop table System.course; --Won't work

--d--

create user admin1 identified by ps1;

grant administrator to admin1;

connect admin1/ps1;

select \* from System.department;

select \* from System.instructor;

update System.department set budget='150000' where dept\_name='Music';

drop table System.department; --Won't work

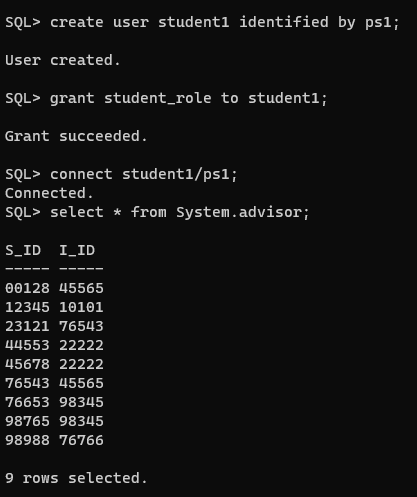
* 1. **Analysis and Explanation**

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.

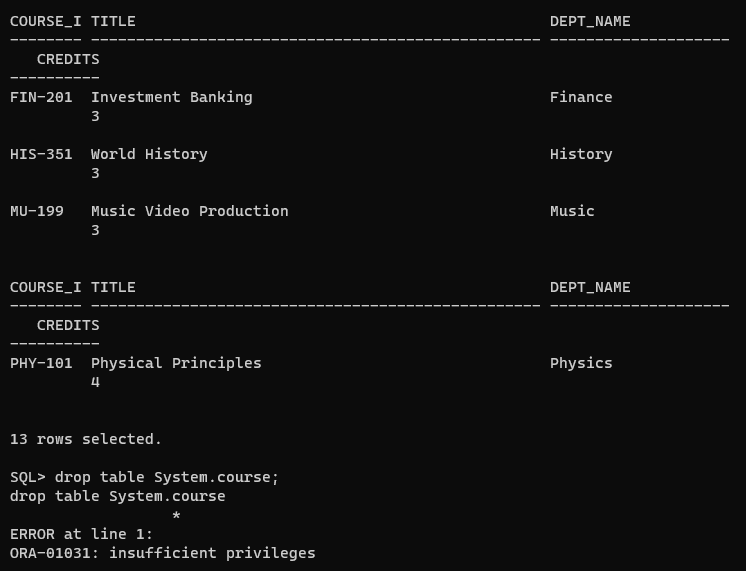
* 1. **Difficulties**

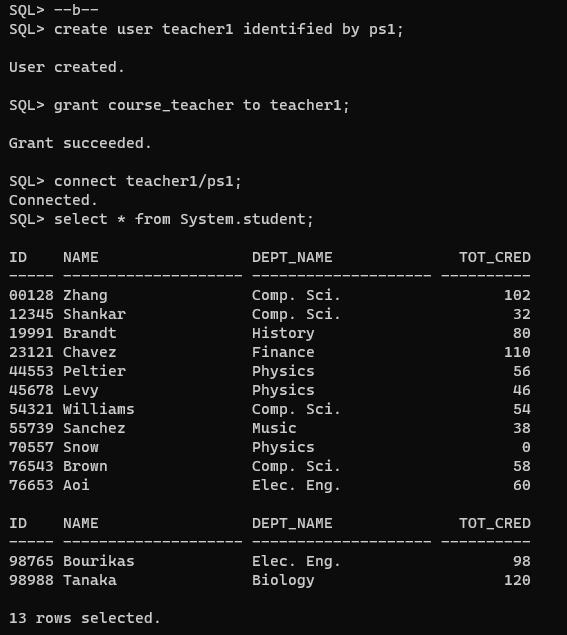
I faced some difficulty when using the insert command. The structure of the inserting values had to match the attribute constrictions of the table and the dept\_name had to be an existing one from department table which I did not understand at first and faced-errors.

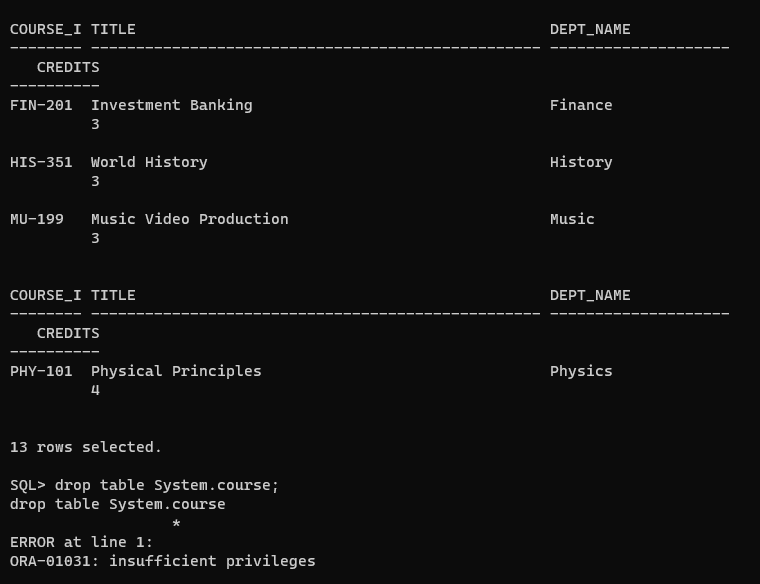
* 1. **Output**

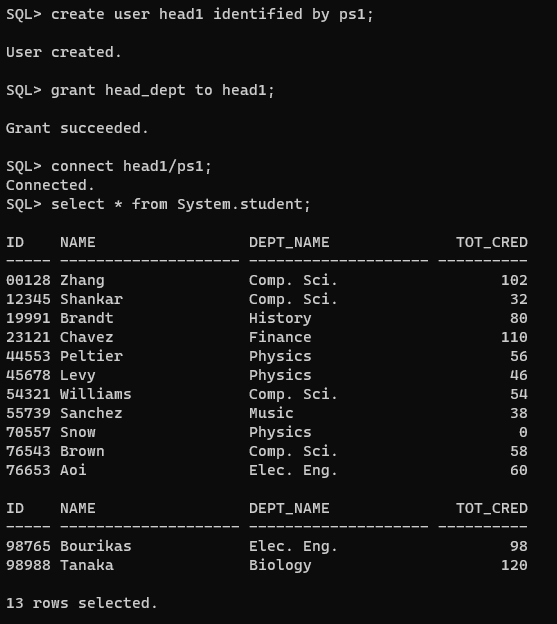
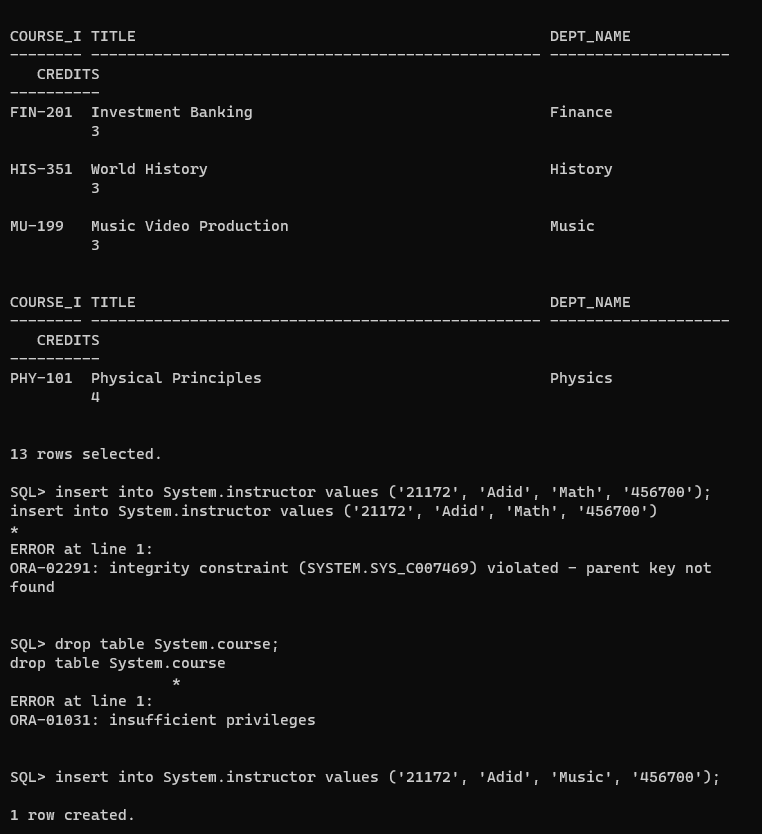
**(a)**

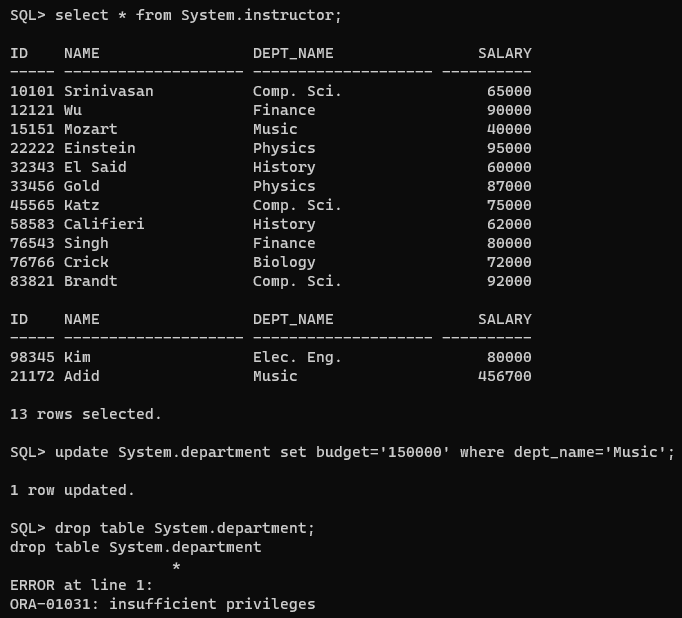
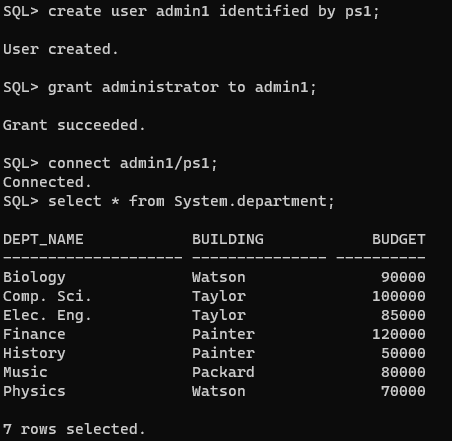
****

****

(b)



(c) 

(d)