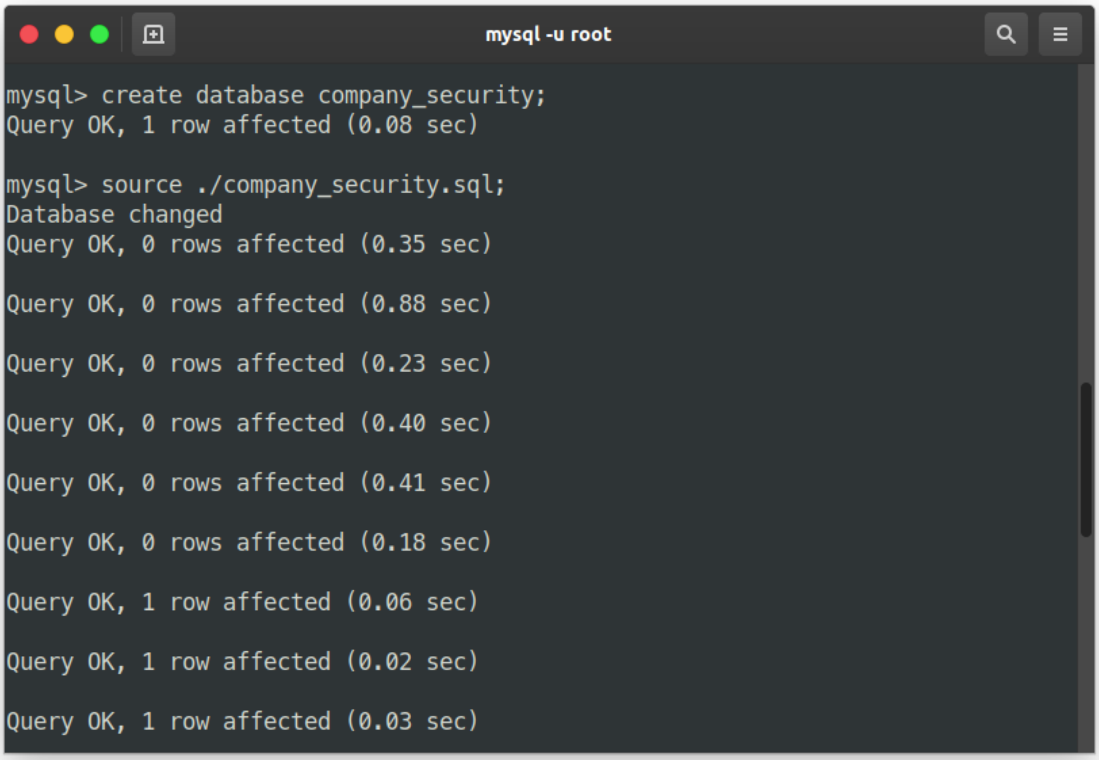
CO527 Advanced Database Systems

Lab Number: 05
Topic: Database Security
Name: M.M.M Irfan
Reg No: E/15/138

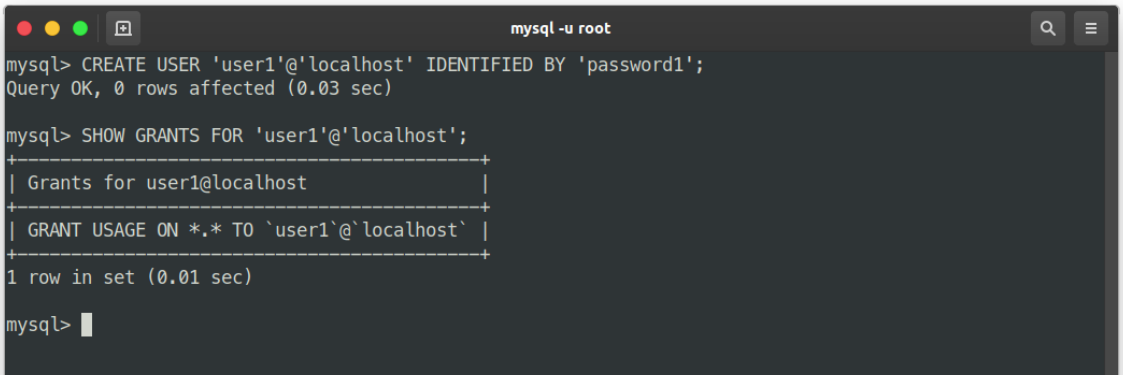
4. In-Class Exercise

1. Create database company security.
2. Load the given company security.sql le to the company security database.



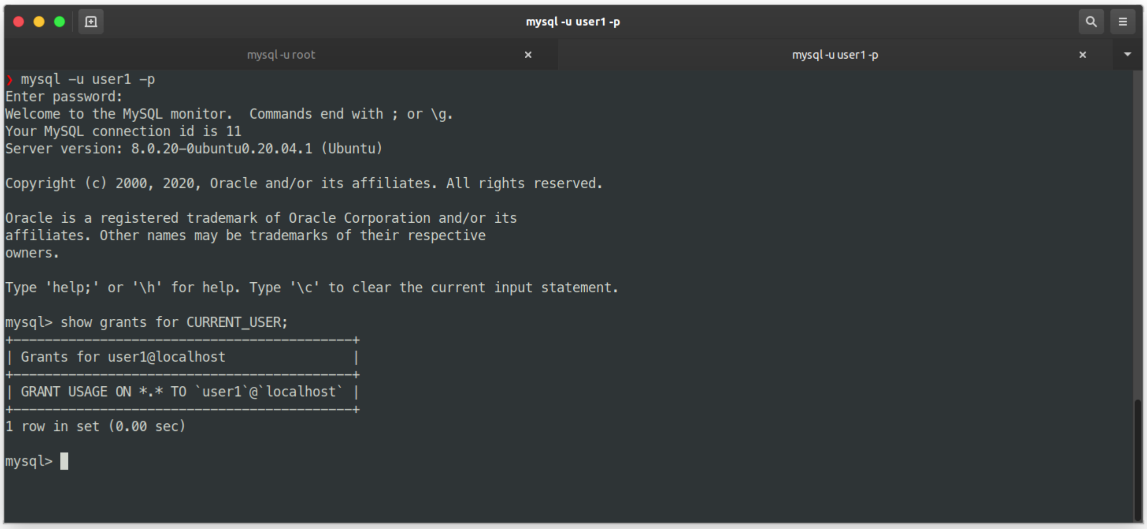
```
mysql> create database company_security;
Query OK, 1 row affected (0.08 sec)

mysql> source ./company_security.sql;
Database changed
Query OK, 0 rows affected (0.35 sec)

Query OK, 0 rows affected (0.88 sec)

Query OK, 0 rows affected (0.23 sec)

Query OK, 0 rows affected (0.40 sec)

Query OK, 0 rows affected (0.41 sec)

Query OK, 0 rows affected (0.18 sec)

Query OK, 1 row affected (0.06 sec)

Query OK, 1 row affected (0.02 sec)

Query OK, 1 row affected (0.03 sec)
```

3. Create a new user ‘user1’ within the MySQL shell.



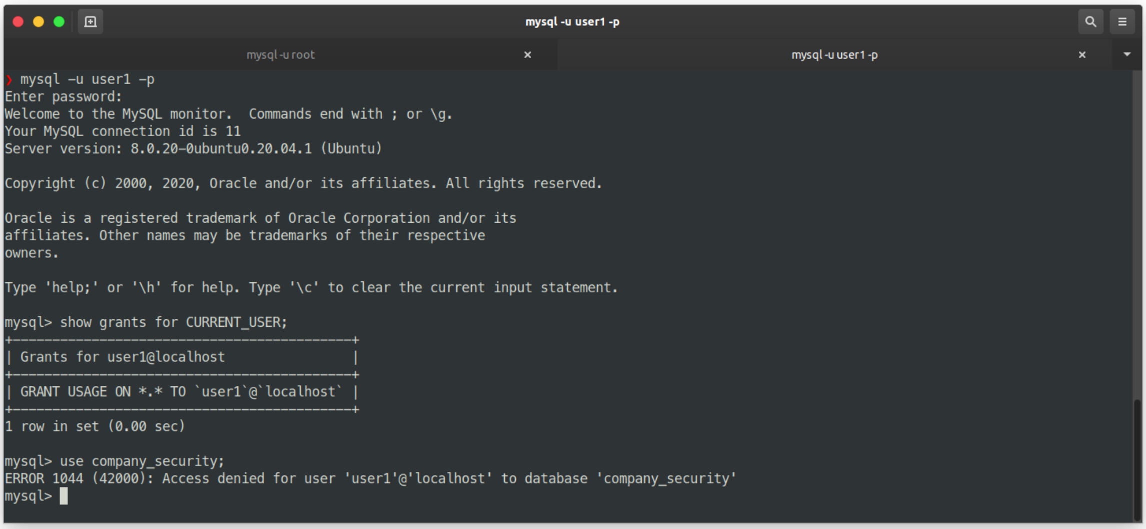
```
mysql> CREATE USER 'user1'@'localhost' IDENTIFIED BY 'password1';
Query OK, 0 rows affected (0.03 sec)

mysql> SHOW GRANTS FOR 'user1'@'localhost';
+------------------------------------------+
| Grants for user1@localhost               |
+------------------------------------------+
| GRANT USAGE ON *.* TO `user1`@`localhost` |
+------------------------------------------+
1 row in set (0.01 sec)

mysql> 
```

4. Login to MySQL with a new user account and password and see if the new user has any authorities or privileges to the database.



```
❯ mysql -u user1 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.20-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show grants for CURRENT_USER;
+-------------------------------------------+
| Grants for user1@localhost                |
+-------------------------------------------+
| GRANT USAGE ON *.* TO `user1`@`localhost` |
+-------------------------------------------+
1 row in set (0.00 sec)

mysql> 
```

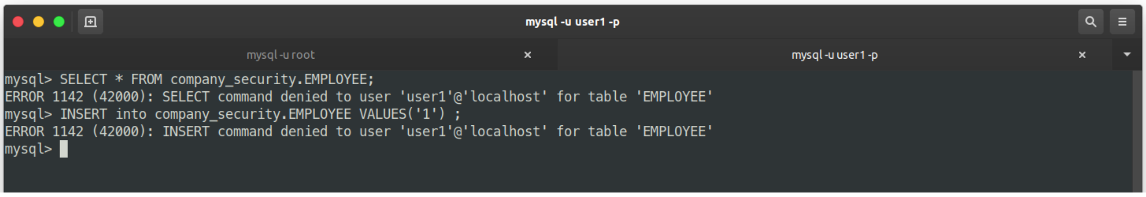
5. Make sure the new user has only read only permission to ‘Employee’ table.



```
❯ mysql -u user1 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.20-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show grants for CURRENT_USER;
+-------------------------------------------+
| Grants for user1@localhost                |
+-------------------------------------------+
| GRANT USAGE ON *.* TO `user1`@`localhost` |
+-------------------------------------------+
1 row in set (0.00 sec)

mysql> use company_security;
ERROR 1044 (42000): Access denied for user 'user1'@'localhost' to database 'company_security'
mysql> 
```

The user 1 has no permission to do anything on the ***company_security*** database

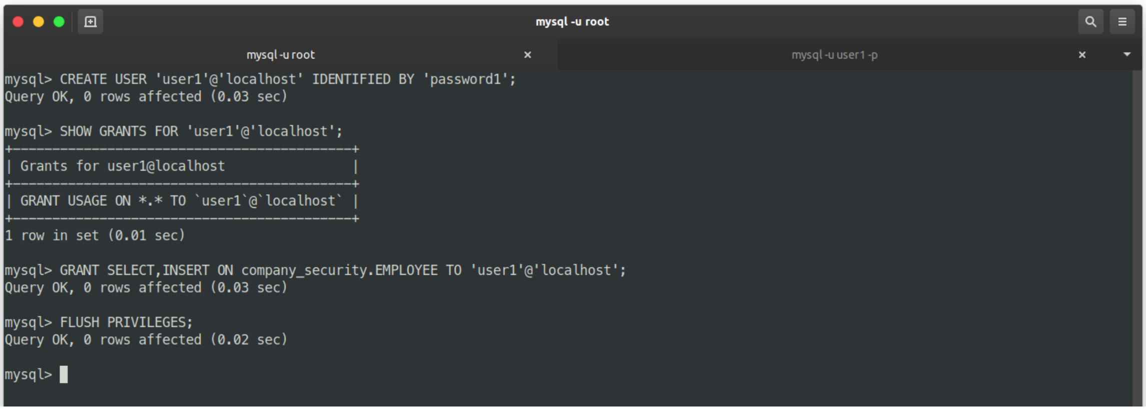
6. Now allow ‘user1’ to query the followings: SELECT * FROM Employee; INSERT into Employee(...)VALUES(...). What happens? Fix the problem.



```
mysql> SELECT * FROM company_security.EMPLOYEE;
ERROR 1142 (42000): SELECT command denied to user 'user1'@'localhost' for table 'EMPLOYEE'
mysql> INSERT into company_security.EMPLOYEE VALUES('1') ;
ERROR 1142 (42000): INSERT command denied to user 'user1'@'localhost' for table 'EMPLOYEE'
mysql> 
```

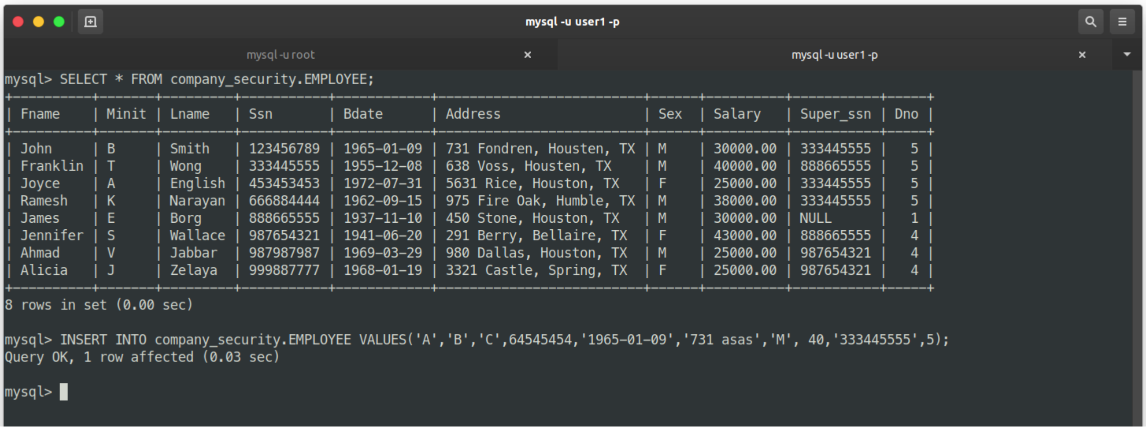
It can be seen that the user has no permission to execute select or insert to employee table. So it throws the above error

To fix the problem we need to grant access to the ***user1 SELECT and INSERT*** privileges on the employee table



```
mysql> CREATE USER 'user1'@'localhost' IDENTIFIED BY 'password1';
Query OK, 0 rows affected (0.03 sec)

mysql> SHOW GRANTS FOR 'user1'@'localhost';
+-------------------------------------------+
| Grants for user1@localhost                |
+-------------------------------------------+
| GRANT USAGE ON *.* TO `user1`@`localhost` |
+-------------------------------------------+
1 row in set (0.01 sec)

mysql> GRANT SELECT,INSERT ON company_security.EMPLOYEE TO 'user1'@'localhost';
Query OK, 0 rows affected (0.03 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.02 sec)

mysql> 
```

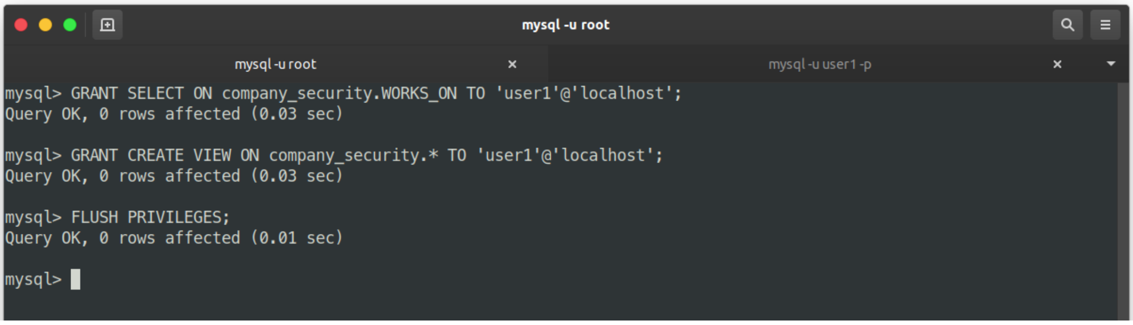
So now ***user1*** can execute those queries



```
mysql> SELECT * FROM company_security.EMPLOYEE;
+----------+-------+---------+-----------+------------+--------------------------+------+----------+-----------+-----+
| Fname    | Minit | Lname   | Ssn       | Bdate      | Address                  | Sex  | Salary   | Super_ssn | Dno |
+----------+-------+---------+-----------+------------+--------------------------+------+----------+-----------+-----+
| John     | B     | Smith   | 123456789 | 1965-01-09 | 731 Fondren, Housten, TX | M    | 30000.00 | 333445555 |   5 |
| Franklin | T     | Wong    | 333445555 | 1955-12-08 | 638 Voss, Housten, TX    | M    | 40000.00 | 888665555 |   5 |
| Joyce    | A     | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX   | F    | 25000.00 | 333445555 |   5 |
| Ramesh   | K     | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M    | 38000.00 | 333445555 |   5 |
| James    | E     | Borg    | 888665555 | 1937-11-10 | 450 Stone, Houston, TX   | M    | 30000.00 | NULL      |   1 |
| Jennifer | S     | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX  | F    | 43000.00 | 888665555 |   4 |
| Ahmad    | V     | Jabbar  | 987987987 | 1969-03-29 | 980 Dallas, Houston, TX  | M    | 25000.00 | 987654321 |   4 |
| Alicia   | J     | Zelaya  | 999887777 | 1968-01-19 | 3321 Castle, Spring, TX  | F    | 25000.00 | 987654321 |   4 |
+----------+-------+---------+-----------+------------+--------------------------+------+----------+-----------+-----+
8 rows in set (0.00 sec)

mysql> INSERT INTO company_security.EMPLOYEE VALUES('A','B','C',64545454,'1965-01-09','731 asas','M', 40,'333445555',5);
Query OK, 1 row affected (0.03 sec)

mysql> 
```

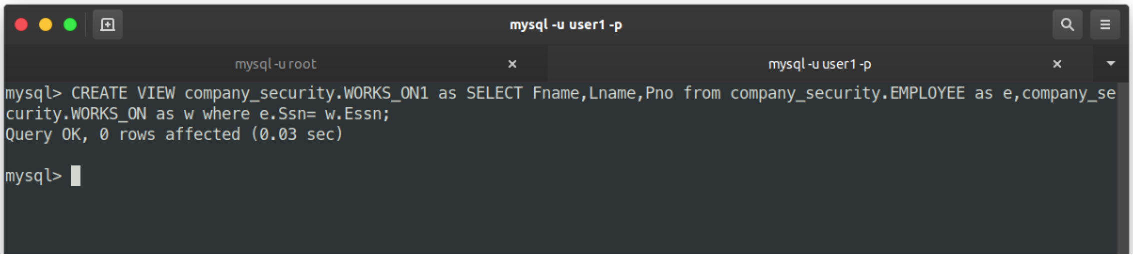
7. From user1 create a view WORKS ON1(Fname,Lname,Pno) on EMPLOYEE and WORKS ON. (Note: You will have to give permission to user1 on CREATE VIEW). Give another user ‘user2’ permission to select tuples from WORKS ON1(Note: user2 will not be able to see WORKS ON or EMPLOYEE).

   Grant ***user1*** to create views in the ***company_security*** database
   To use the ***WORKS_ON*** table to create the view we need to give ***SELECT*** privileges to ***user1.*** The user already has ***SELECT*** privilege on ***EMPLOYEE*** table



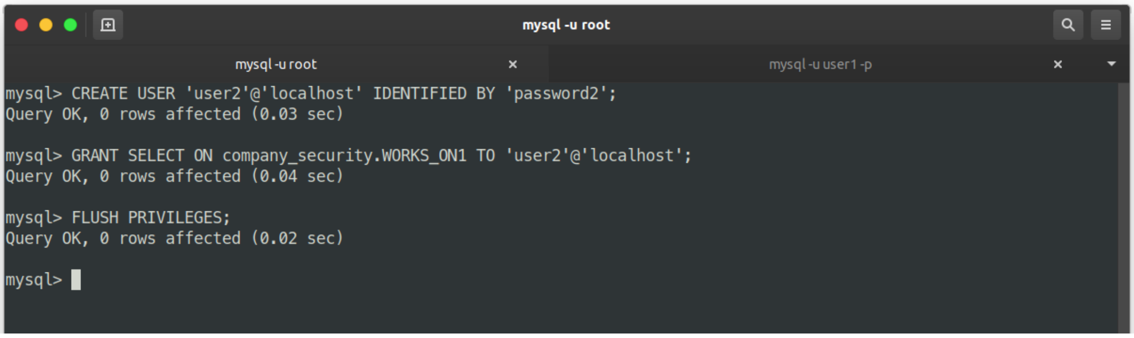
```
mysql> GRANT SELECT ON company_security.WORKS_ON TO 'user1'@'localhost';
Query OK, 0 rows affected (0.03 sec)

mysql> GRANT CREATE VIEW ON company_security.* TO 'user1'@'localhost';
Query OK, 0 rows affected (0.03 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql> 
```

Now create the view as ***user1***



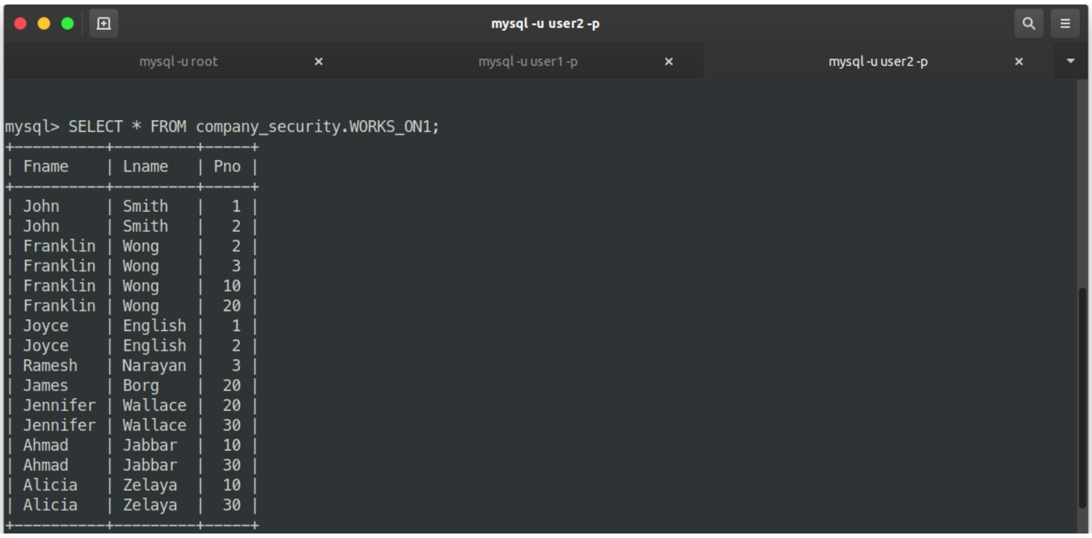
```
mysql> CREATE VIEW company_security.WORKS_ON1 as SELECT Fname,Lname,Pno from company_security.EMPLOYEE as e,company_se
curity.WORKS_ON as w where e.Ssn= w.Essn;
Query OK, 0 rows affected (0.03 sec)

mysql> 
```

8. Select tuples from user2 account. What happens?
   Create ***user2***, grant permission to ***SELECT*** in ***WORKS_ON1***



```
mysql> CREATE USER 'user2'@'localhost' IDENTIFIED BY 'password2';
Query OK, 0 rows affected (0.03 sec)

mysql> GRANT SELECT ON company_security.WORKS_ON1 TO 'user2'@'localhost';
Query OK, 0 rows affected (0.04 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.02 sec)

mysql> 
```

Retrieve entries from ***WORKS_ON1*** as ***user2***

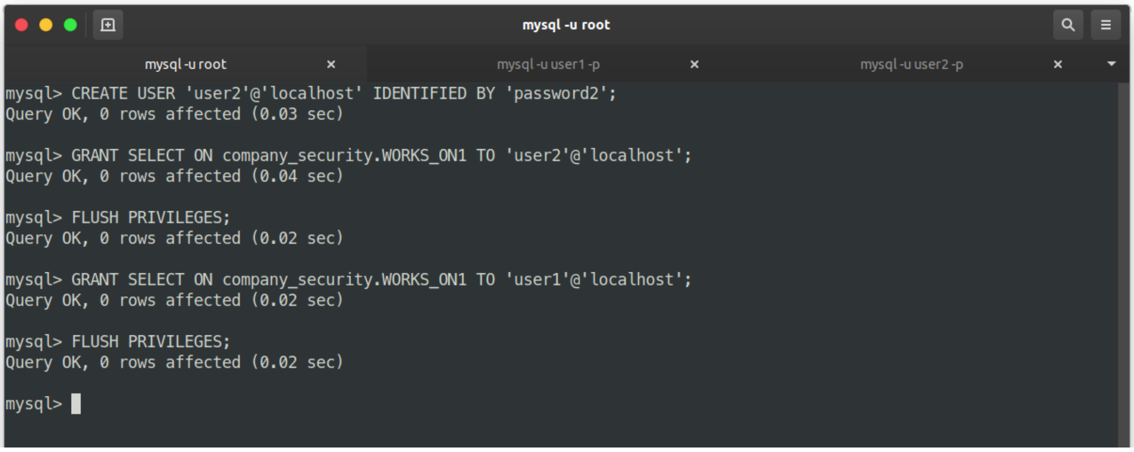


```
mysql> SELECT * FROM company_security.WORKS_ON1;
+----------+---------+-----+
| Fname    | Lname   | Pno |
+----------+---------+-----+
| John     | Smith   |   1 |
| John     | Smith   |   2 |
| Franklin | Wong    |   2 |
| Franklin | Wong    |   3 |
| Franklin | Wong    |  10 |
| Franklin | Wong    |  20 |
| Joyce    | English |   1 |
| Joyce    | English |   2 |
| Ramesh   | Narayan |   3 |
| James    | Borg    |  20 |
| Jennifer | Wallace |  20 |
| Jennifer | Wallace |  30 |
| Ahmad    | Jabbar  |  10 |
| Ahmad    | Jabbar  |  30 |
| Alicia   | Zelaya  |  10 |
| Alicia   | Zelaya  |  30 |
+----------+---------+-----+
```

9. Remove privileges of user1 on WORKS ON and EMPLOYEE. Can user1 still access WORKS ON1? What happened to WORKS ON1? Why?

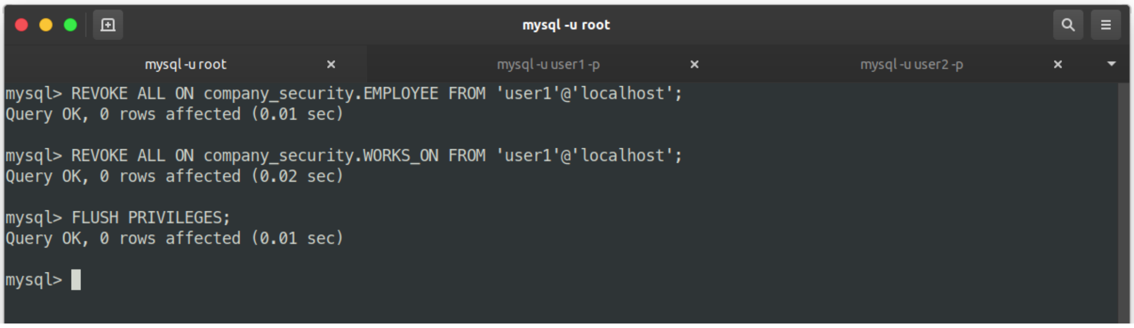
To perform this exercise, First of all, the ***user1*** has to have permission to retrieve entries from the ***WORKS_ON1*** view.

So first give permission to ***user1*** so that it can retrieve data from ***WORKS_ON1*** view



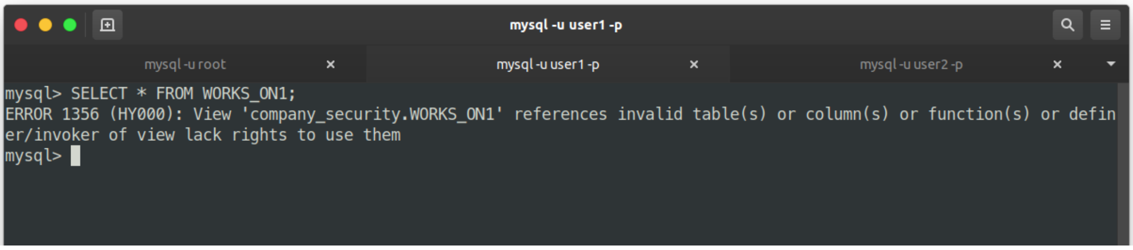
```
mysql> CREATE USER 'user2'@'localhost' IDENTIFIED BY 'password2';
Query OK, 0 rows affected (0.03 sec)

mysql> GRANT SELECT ON company_security.WORKS_ON1 TO 'user2'@'localhost';
Query OK, 0 rows affected (0.04 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT SELECT ON company_security.WORKS_ON1 TO 'user1'@'localhost';
Query OK, 0 rows affected (0.02 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.02 sec)

mysql> ▌
```

Now revoke all permission on ***EMPLOYEE***, ***WORKS_ON1*** from ***user1***



```
mysql> REVOKE ALL ON company_security.EMPLOYEE FROM 'user1'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> REVOKE ALL ON company_security.WORKS_ON FROM 'user1'@'localhost';
Query OK, 0 rows affected (0.02 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql> ▌
```

So now ***user1*** cannot access the created view



```
mysql> SELECT * FROM WORKS_ON1;
ERROR 1356 (HY000): View 'company_security.WORKS_ON1' references invalid table(s) or column(s) or function(s) or defin
er/invoker of view lack rights to use them
mysql> ▌
```

The reason for this is, A view is nothing more than a SQL statement that is stored in the database with an associated name. A view is actually a composition of a table in the form of a predefined SQL query. So the user should have ***SELECT*** Privileges on the corresponding tables used in the ***CREATE VIEW*** command. So once we revoke the permission the user cannot execute the select queries in the view so it becomes a invalid table (view).

5. Assignment

Consider the relational database schema provided. Suppose that all the relations were created by (and hence are owned by) user X, who wants to grant the following privileges to user accounts A, B, C, D, and E:

I. Account A can retrieve or modify any relation except DEPENDENT and can grant any of these privileges to other users.

```
GRANT SELECT, UPDATE ON EMPLOYEE, DEPARTMENT, DEPT_LOCATIONS, PROJECT,
WORKS_ON TO AccountA WITH GRANT OPTION;
```

II. Account B can retrieve all the attributes of EMPLOYEE and DEPARTMENT except for Salary, Mgr ssn, and Mgr start date.

```
--create separate views with specific attributes on the corresponding
tables and grant select access
-- for EMPLOYEE table
CREATE VIEW EMPS AS SELECT Fname, Minit, Lname, Ssn, Bdate, Address,
Sex, Super_ssn, Dno FROM EMPLOYEE;
GRANT SELECT ON EMPS TO AccountB;

-- for DEPARTMENT TABLE
CREATE VIEW DEPTS AS SELECT Dname, Dnumber FROM DEPARTMENT;
GRANT SELECT ON DEPTS TO AccountB;
```

III. Account C can retrieve or modify WORKS ON but can only retrieve the Fname, Minit, Lname, and Ssn attributes of EMPLOYEE and the Pname and Pnumber attributes of PROJECT.

```
-- grant select and update for works on
GRANT SELECT, UPDATE ON WORKS_ON TO AccountC;
-- create a view for account c with specific attributes and grant select
CREATE VIEW EMP_C AS SELECT Fname, Minit, Lname, Ssn FROM EMPLOYEE;
GRANT SELECT ON EMP_C TO AccountC;

CREATE VIEW PROJ_C AS SELECT Pname, Pno FROM PROJECT;
GRANT SELECT ON PROJ_C TO AccountC;
```

IV. Account D can retrieve any attribute of EMPLOYEE or DEPENDENT and can modify DEPENDENT.

```
-- DEPENDENT.
GRANT SELECT ON EMPLOYEE, DEPENDENT TO AccountD;
GRANT UPDATE ON DEPENDENT TO AccountD;
```

V. Account E can retrieve any attribute of EMPLOYEE but only for EMPLOYEE tuples that have Dno = 3.

```
-- Account E can retrieve any attribute of EMPLOYEE but only for
EMPLOYEE tuples that
-- have Dno = 3.
CREATE VIEW DNO3_EMPLOYEES AS SELECT * FROM EMPLOYEE WHERE Dno = 3;
GRANT SELECT ON DNO3_EMPLOYEES TO AccountE;
```

Note: The Lab files (SQL scripts, screenshots, report) can be found at https://github.com/irfanm96/CO527/tree/master/lab05

REFERENCES

- https://www.tutorialspoint.com/sql/sql-using-views.htm#:~:text=To%20create%20a%20view%2C%20a,according%20to%20the%20specific%20implementation.&text=CREATE%20VIEW%20view_name%20AS%20SELECT,a%20normal%20SQL%20SELECT%20query.