PES UNIVERSITY

B.TECH. (CSE)

V SEMESTER

UE19CS301 – DATABASE MANAGEMENT SYSTEM

ASSIGNMENT-3

SUBMITTED BY

TEAM ID-3

NAME	SRN
T. LOHITH SRINIVAS	PES2UG19CS203
LALITHA SRAVANTI DASU	PES2UG19CS201
MEENAKSHI SURESH	PES2UG19CS228

AUGUST – DECEMBER 2021

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ELECTRONIC CITY CAMPUS,

BENGALURU – 560100, KARNATAKA, INDIA

SIMPLE QUERIES

1) Display employee details of all the employees born in or after 2000.

SELECT * FROM employee WHERE b_date >= '2000-01-01';

```
employeemanagement=# SELECT * FROM employee WHERE b_date >= '2000-01-01';
 ssn |   b_date   |  status   |  sex   |      s_name      |  l_name  |            address             | grade
-----+------------+-----------+--------+------------------+----------+--------------------------------+-------
   1 | 2000-11-03 | permanent | female | Lalitha Sravanti | Dasu     | HSR Layout                     |     5
   2 | 2001-04-10 | permanent | female | Meenakshi        | Suresh   | sector-5 HSR Layout, Bangalore |     5
   3 | 2001-12-26 | permanent | male   | Lohith           | Srinivas | Electronic City, Bangalore     |     4
   5 | 2001-09-03 | temporary | female | Satya            | Rajan    | Koramangala, Bangalore         |     4
   9 | 2000-09-07 | temporary | male   | Vinod            | Narayan  | Bommanahalli, Bangalore        |     4
(5 rows)

employeemanagement=#
```

2) Display the name and relationship of the dependents of the employee whose first name is Deep.

SELECT d.Name, d.relationship FROM dependents d, employee e WHERE e.SSN = d.ssn and e.s_name = 'Deep';

```
employeemanagement=# SELECT d.Name, d.relationship FROM dependents d, employee e WHERE e.SSN = d.ssn and e.s_name = 'Deep';
     name      | relationship
---------------+--------------
 Anika Sen     | mother
 Riya Mehta    | daughter
 Sunitha Mehta | wife
(3 rows)
```

3) What are the leave types availed by the employee whose essn is 6?

SELECT l.Type FROM leave_type l, employee_leave e WHERE e.ssn=6 AND e.lid=l.lid;

```
employeemanagement=# SELECT l.Type FROM leave_type l, employee_leave e WHERE e.ssn=6 AND e.lid=l.lid;
      type
-----------------
 Sick Leave
 Maternity Leave
(2 rows)

employeemanagement=#
```

4) Display each department name along with its manager.

SELECT d.name , e.s_name as "Manager first name", e.l_name as "Manager last name" FROM employee e, department d WHERE e.SSN = d.Mssn;

```
employeemanagement=# SELECT d.name , e.s_name as "Manager first name", e.l_name as "Manager last name" FROM employee e, department d WHERE e.S
SN = d.Mssn;
        name         | Manager first name | Manager last name
---------------------+--------------------+-------------------
 Product Development | Sunaina            | Agrawal
 Human Resources     | Deep               | Mehta
 Technical Support   | Rahul              | Mittal
 Sales Support       | Sanjay             | Dutt
 Product Research    | Neel               | Roy
(5 rows)
```

5) Display all the departments (IDs) in the Verizon Corporation Limited branch (building).

SELECT d.name as "Department Name" FROM department d,department_building h, building b WHERE (b.name='Verizon Corporation Limited' AND b.bid=h.bid AND d.did=h.did);

```
employeemanagement=# SELECT d.name as "Department Name" FROM department d,department_building h, building b WHERE (b.name='Verizon Corporation
 Limited' AND b.bid=h.bid AND d.did=h.did);
   Department Name
--------------------
 Product Development
 Human Resources
 Technical Support
 Sales Support
(4 rows)
```

COMPLEX QUERIES

1) Find manager details of employee with id 3.

SELECT s_name,l_name FROM employee WHERE ssn= (SELECT mssn FROM ((employee e NATURAL JOIN employee_department d) p NATURAL JOIN department t) WHERE ssn=3);

```
employeemanagement=# SELECT s_name,l_name FROM employee WHERE ssn= ( SELECT mssn FROM ((employee e NATURAL JOIN employee_department d) p NATUR
AL JOIN department t) WHERE ssn=3);
 s_name | l_name
--------+--------
 Deep   | Mehta
(1 row)
```

2) Find all employees who do not have dependents.

SELECT * FROM employee e WHERE NOT EXISTS (SELECT * FROM dependents d WHERE e.ssn=d.ssn);

```
employeemanagement=# SELECT * FROM employee e WHERE NOT EXISTS ( SELECT * FROM dependents d WHERE e.ssn=d.ssn);
 ssn |   b_date   |  status   |  sex   |      s_name      |  l_name  |          address           | grade
-----+------------+-----------+--------+------------------+----------+----------------------------+-------
   1 | 2000-11-03 | permanent | female | Lalitha Sravanti | Dasu     | HSR Layout                 |     5
   3 | 2001-12-26 | permanent | male   | Lohith           | Srinivas | Electronic City, Bangalore |     4
   5 | 2001-09-03 | temporary | female | Satya            | Rajan    | Koramangala, Bangalore     |     4
   7 | 1991-03-10 | permanent | male   | Rahul            | Mittal   | Banashankari, Bangalore    |     6
   8 | 1997-07-12 | permanent | male   | Sanjay           | Dutt     | Whitefield, Bangalore      |     6
   9 | 2000-09-07 | temporary | male   | Vinod            | Narayan  | Bommanahalli, Bangalore    |     4
(6 rows)
```

3) Display skills of all employees working on fingerprint voting system project.

select s.name from Skill s where s.sid in (select es.sid from Employee_Skill es where es.ssn in (select ep.ssn from Employee_Project ep where ep.pid in (select p.pid from Project p where p.name='Fingerprint Voting System')));

```
employeemanagement=# select s.name from Skill s where s.sid in (select es.sid from Employee_Skill es where es.ssn in (select ep.ssn from Emplo
yee_Project ep where ep.pid in (select p.pid from Project p where p.name='Fingerprint Voting System')));
              name
---------------------------------
 Big Data Analysis
 Project Planning
 Quality Testing
 Database Systems
 Market Intelligence
 Integrated Circuit Design
 Object Oriented Design
 Programming
 Written and Verbal Communication
 Mentoring
(10 rows)
```

4) Display all employees working in the department whose manager is Sunaina.

SELECT e.s_name,e.l_name FROM employee e WHERE e.ssn in (SELECT dep.ssn FROM employee_department dep WHERE dep.did =(SELECT d.did FROM department d WHERE d.mssn=(SELECT emp.ssn from employee emp WHERE emp.s_name='Sunaina'))) AND e.s_name NOT IN ('Sunaina');

```
employeemanagement=# SELECT e.s_name,e.l_name FROM employee e WHERE e.ssn in (SELECT dep.ssn FROM employee_department dep WHERE dep.did =(SELE
CT d.did FROM department d WHERE d.mssn=(SELECT emp.ssn from employee emp WHERE emp.s_name='Sunaina'))) AND e.s_name NOT IN ('Sunaina');
  s_name   | l_name
-----------+--------
 Meenakshi | Suresh
(1 row)
```

5) Display the names of the builidings where the department in which Meenakshi Suresh works operate.

select b.name from Building b where b.bid in (select db.bid from Department_Building db where db.did in (select ed.did from Employee_Department ed where ed.ssn in (select e.ssn from Employee e where e.s_name='Meenakshi' and e.l_name='Suresh')));

```
employeemanagement=# select b.name from Building b where b.bid in (select db.bid from Department_Building db where db.did in (select ed.did fr
om Employee_Department ed where ed.ssn in (select e.ssn from Employee e where e.s_name='Meenakshi' and e.l_name='Suresh')));
            name
----------------------------
 Verizon Corporation Limited
 Verizon Research Lab
(2 rows)
```

PERFORMANCE ANALYSIS

1) Performance analysis for displaying skills of all employees working on fingerprint voting system project.

EXPLAIN ANALYZE (select s.name from Skill s where s.sid in (select es.sid from Employee_Skill es where es.ssn in (select ep.ssn from Employee_Project ep where ep.pid in (select p.pid from Project p where p.name='Fingerprint Voting System'))));

```
                                                                    QUERY PLAN
----------------------------------------------------------------------------------------------------------------------------------------------
----------------
 Hash Semi Join  (cost=57.67..72.99 rows=116 width=218) (actual time=0.355..0.381 rows=10 loops=1)
   Hash Cond: (s.sid = es.sid)
   ->  Seq Scan on skill s  (cost=0.00..11.70 rows=170 width=222) (actual time=0.028..0.034 rows=10 loops=1)
   ->  Hash  (cost=55.97..55.97 rows=136 width=4) (actual time=0.246..0.253 rows=12 loops=1)
         Buckets: 1024  Batches: 1  Memory Usage: 9kB
         ->  Nested Loop  (cost=47.63..55.97 rows=136 width=4) (actual time=0.163..0.237 rows=12 loops=1)
               ->  HashAggregate  (cost=47.48..47.60 rows=12 width=4) (actual time=0.119..0.129 rows=5 loops=1)
                     Group Key: ep.ssn
                     ->  Hash Join  (cost=14.03..47.45 rows=12 width=4) (actual time=0.075..0.105 rows=5 loops=1)
                           Hash Cond: (ep.pid = p.pid)
                           ->  Seq Scan on employee_project ep  (cost=0.00..28.50 rows=1850 width=8) (actual time=0.019..0.028 rows=18 loops=1
)
                           ->  Hash  (cost=14.00..14.00 rows=2 width=4) (actual time=0.037..0.039 rows=1 loops=1)
                                 Buckets: 1024  Batches: 1  Memory Usage: 9kB
                                 ->  Seq Scan on project p  (cost=0.00..14.00 rows=2 width=4) (actual time=0.023..0.029 rows=1 loops=1)
                                       Filter: ((name)::text = 'Fingerprint Voting System'::text)
                                       Rows Removed by Filter: 4
               ->  Index Only Scan using employee_skill_pkey on employee_skill es  (cost=0.15..0.59 rows=11 width=8) (actual time=0.012..0.017
 rows=2 loops=5)
                     Index Cond: (ssn = ep.ssn)
                     Heap Fetches: 12
 Planning Time: 1.374 ms
 Execution Time: 0.553 ms
(21 rows)

(END)
```

2)Performance analysis for displaying all employees working in the department whose manager is Sunaina.

EXPLAIN ANALYZE (SELECT e.s_name,e.l_name FROM employee e WHERE e.ssn in (SELECT dep.ssn FROM employee_department dep WHERE dep.did =(SELECT d.did FROM department d WHERE d.mssn=(SELECT emp.ssn from employee emp WHERE emp.s_name='Sunaina'))) AND e.s_name NOT IN ('Sunaina'));

```
                                                              QUERY PLAN
----------------------------------------------------------------------------------------------------------------------------------------------
-
 Hash Join  (cost=60.67..71.57 rows=11 width=436) (actual time=0.299..0.316 rows=1 loops=1)
   Hash Cond: (e.ssn = dep.ssn)
   InitPlan 2 (returns $1)
     ->  Seq Scan on department d  (cost=10.75..24.75 rows=2 width=4) (actual time=0.047..0.051 rows=1 loops=1)
           Filter: (mssn = $0)
           Rows Removed by Filter: 4
           InitPlan 1 (returns $0)
             ->  Seq Scan on employee emp  (cost=0.00..10.75 rows=1 width=4) (actual time=0.015..0.020 rows=1 loops=1)
                   Filter: ((s_name)::text = 'Sunaina'::text)
                   Rows Removed by Filter: 9
   ->  Seq Scan on employee e  (cost=0.00..10.75 rows=59 width=440) (actual time=0.032..0.040 rows=9 loops=1)
         Filter: ((s_name)::text <> 'Sunaina'::text)
         Rows Removed by Filter: 1
   ->  Hash  (cost=35.78..35.78 rows=11 width=4) (actual time=0.113..0.115 rows=2 loops=1)
         Buckets: 1024  Batches: 1  Memory Usage: 9kB
         ->  Bitmap Heap Scan on employee_department dep  (cost=25.11..35.78 rows=11 width=4) (actual time=0.076..0.080 rows=2 loops=1)
               Recheck Cond: (did = $1)
               Heap Blocks: exact=1
               ->  Bitmap Index Scan on employee_department_pkey  (cost=0.00..25.11 rows=11 width=0) (actual time=0.067..0.067 rows=2 loops=1)
                     Index Cond: (did = $1)
 Planning Time: 2.794 ms
 Execution Time: 0.672 ms
(22 rows)

(END)
```

3) Perfomance analysis for displaying the names of the builidings where the department in which Meenakshi Suresh works operate.

EXPLAIN ANALYZE (select b.name from Building b where b.bid in (select db.bid from Department_Building db where db.did in (select ed.did from Employee_Department ed where ed.ssn in (select e.ssn from Employee e where e.s_name='Meenakshi' and e.l_name='Suresh'))));

```
                                                                                       QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------------------------
------------------------------
 Hash Join  (cost=75.75..91.57 rows=160 width=218) (actual time=0.085..0.089 rows=2 loops=1)
   Hash Cond: (b.bid = db.bid)
   ->  Seq Scan on building b  (cost=0.00..13.20 rows=320 width=222) (actual time=0.004..0.004 rows=2 loops=1)
   ->  Hash  (cost=73.25..73.25 rows=200 width=4) (actual time=0.062..0.064 rows=2 loops=1)
         Buckets: 1024  Batches: 1  Memory Usage: 9kB
         ->  HashAggregate  (cost=71.25..73.25 rows=200 width=4) (actual time=0.057..0.059 rows=2 loops=1)
               Group Key: db.bid
               ->  Hash Join  (cost=26.87..70.18 rows=429 width=4) (actual time=0.051..0.055 rows=2 loops=1)
                     Hash Cond: (db.did = ed.did)
                     ->  Seq Scan on department_building db  (cost=0.00..32.60 rows=2260 width=8) (actual time=0.002..0.003 rows=7 loops=1)
                     ->  Hash  (cost=26.40..26.40 rows=38 width=4) (actual time=0.031..0.032 rows=1 loops=1)
                           Buckets: 1024  Batches: 1  Memory Usage: 9kB
                           ->  HashAggregate  (cost=26.02..26.40 rows=38 width=4) (actual time=0.027..0.028 rows=1 loops=1)
                                 Group Key: ed.did
                                 ->  Nested Loop  (cost=4.24..25.92 rows=38 width=4) (actual time=0.016..0.019 rows=1 loops=1)
                                       ->  Seq Scan on employee e  (cost=0.00..10.90 rows=1 width=4) (actual time=0.007..0.009 rows=1 loops=1)
                                             Filter: (((s_name)::text = 'Meenakshi'::text) AND ((l_name)::text = 'Suresh'::text))
                                             Rows Removed by Filter: 9
                                       ->  Bitmap Heap Scan on employee_department ed  (cost=4.24..14.91 rows=11 width=8) (actual time=0.006..
0.007 rows=1 loops=1)
                                             Recheck Cond: (ssn = e.ssn)
                                             Heap Blocks: exact=1
                                             ->  Bitmap Index Scan on employee_department_pkey  (cost=0.00..4.24 rows=11 width=0) (actual time
=0.003..0.003 rows=1 loops=1)
                                                   Index Cond: (ssn = e.ssn)
 Planning Time: 0.899 ms
 Execution Time: 0.254 ms
(25 rows)

(END)
```

Creation of multiple users

List of users:

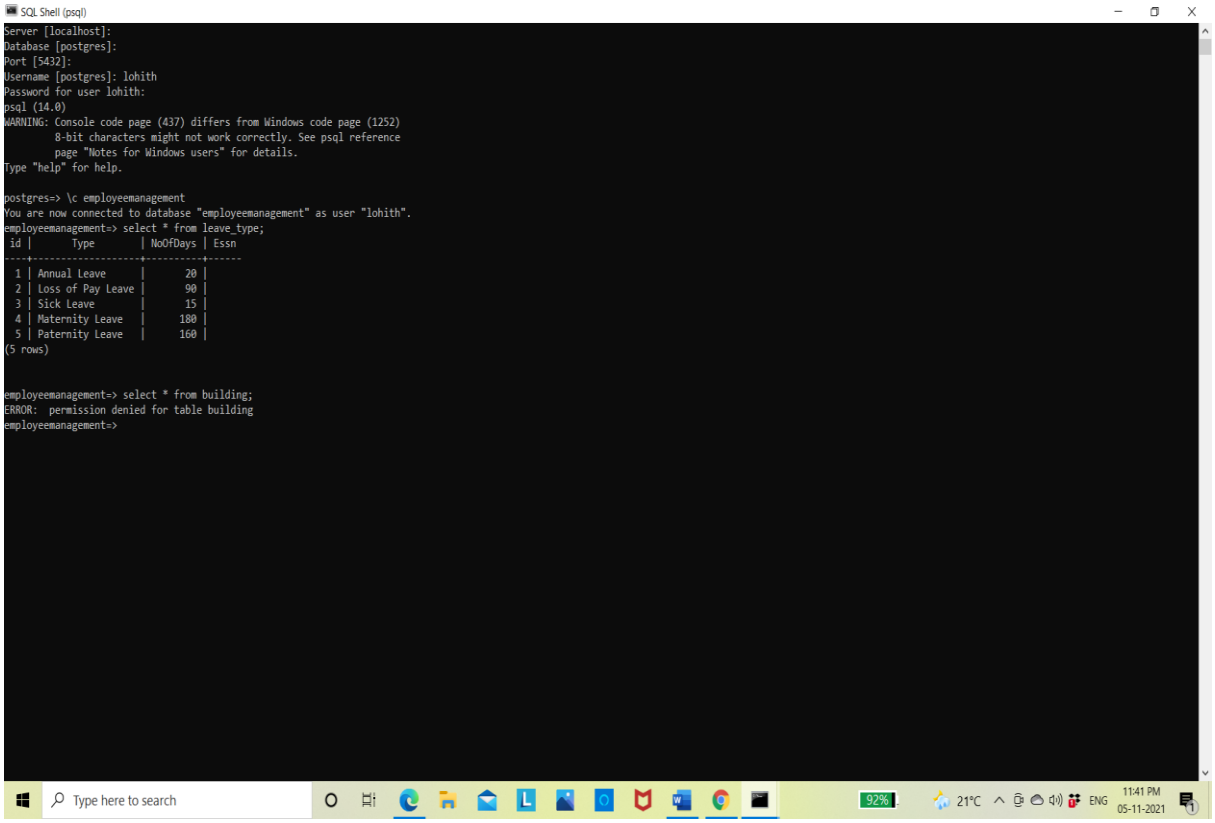
```
postgres=# \du
                                   List of roles
 Role name |                         Attributes                         | Member of
-----------+------------------------------------------------------------+-----------
 lalitha   |                                                            | {}
 lohith    |                                                            | {}
 meenakshi |                                                            | {}
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
 rahul     |                                                            | {}
 riya      |                                                            | {}


postgres=# \d
Did not find any relations.
postgres=# \l
                                                List of databases
       Name        |  Owner   | Encoding |      Collate      |       Ctype       |   Access privileges
-------------------+----------+----------+-------------------+-------------------+-----------------------
 company           | postgres | UTF8     | English_India.1252 | English_India.1252 |
 employeemanagement | postgres | UTF8     | English_India.1252 | English_India.1252 |
 postgres          | postgres | UTF8     | English_India.1252 | English_India.1252 |
 template0         | postgres | UTF8     | English_India.1252 | English_India.1252 | =c/postgres          +
                   |          |          |                   |                   | postgres=CTc/postgres
 template1         | postgres | UTF8     | English_India.1252 | English_India.1252 | =c/postgres          +
                   |          |          |                   |                   | postgres=CTc/postgres
(5 rows)
```

Granting access privileges on different parts of the database to different users:

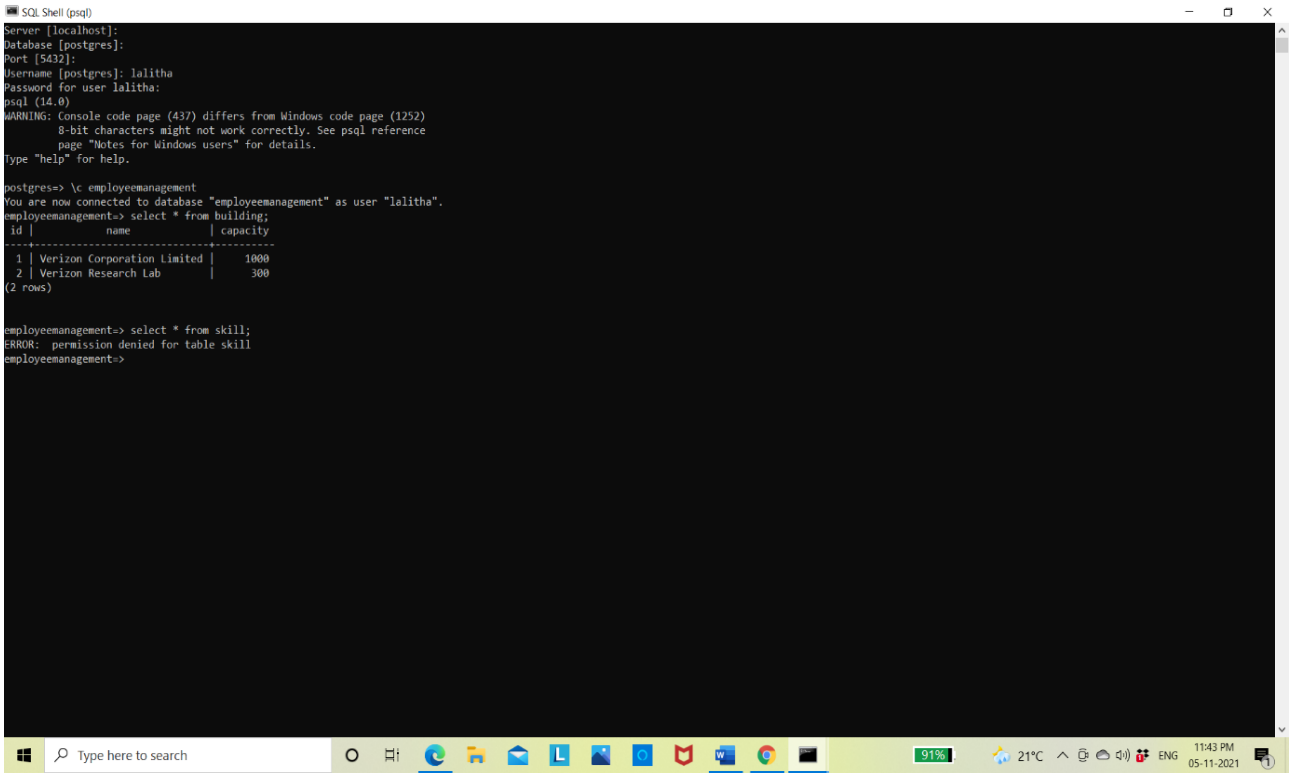
```
postgres=# \c employeemanagement
You are now connected to database "employeemanagement" as user "postgres".
employeemanagement=# \d
           List of relations
 Schema |     Name     | Type  |  Owner
--------+--------------+-------+----------
 public | assigned_to  | table | postgres
 public | avails_leave | table | postgres
 public | building     | table | postgres
 public | department   | table | postgres
 public | dependents   | table | postgres
 public | earns        | table | postgres
 public | employee     | table | postgres
 public | has          | table | postgres
 public | houses       | table | postgres
 public | leave_type   | table | postgres
 public | project      | table | postgres
 public | salary_range | table | postgres
 public | skill        | table | postgres
 public | works_in     | table | postgres
 public | works_on     | table | postgres
(15 rows)


employeemanagement=# GRANT SELECT,INSERT,DELETE,UPDATE,TRUNCATE,REFERENCES ON leave_type,skill to lohith
employeemanagement-# ;
GRANT
employeemanagement=# GRANT SELECT,INSERT,DELETE,UPDATE,TRUNCATE,REFERENCES ON leave_type,building to lalitha;
GRANT
employeemanagement=# GRANT SELECT,INSERT,DELETE,UPDATE,TRUNCATE,REFERENCES ON skill,building to meenakshi;
GRANT
employeemanagement=# GRANT SELECT,INSERT,DELETE,UPDATE,TRUNCATE,REFERENCES ON employee,department,dependents,salary_range to rahul;
GRANT
employeemanagement=# GRANT SELECT,INSERT,DELETE,UPDATE,TRUNCATE,REFERENCES ON employee,department,dependents,salary_range,project to riya;
GRANT
```

Situations of permission denied with different users:

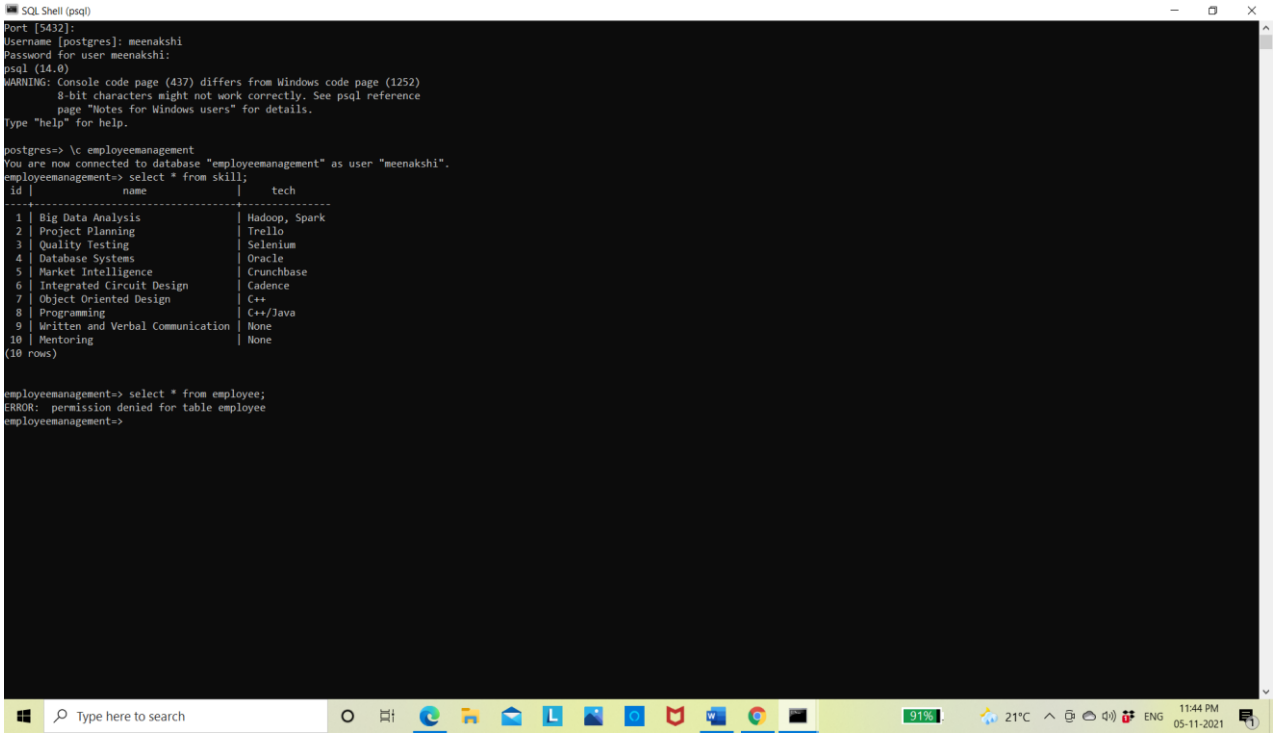
Lohith:



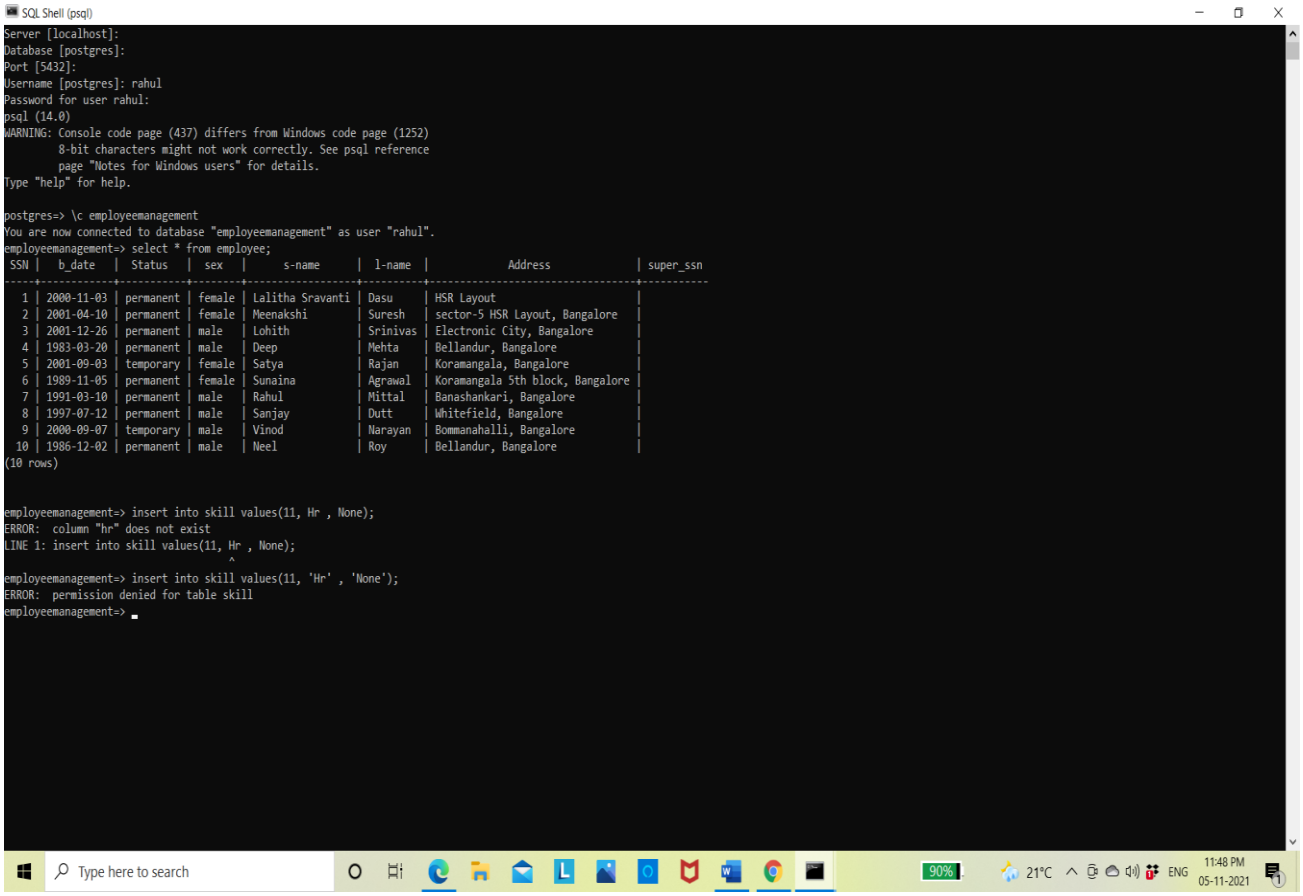
Lalitha:



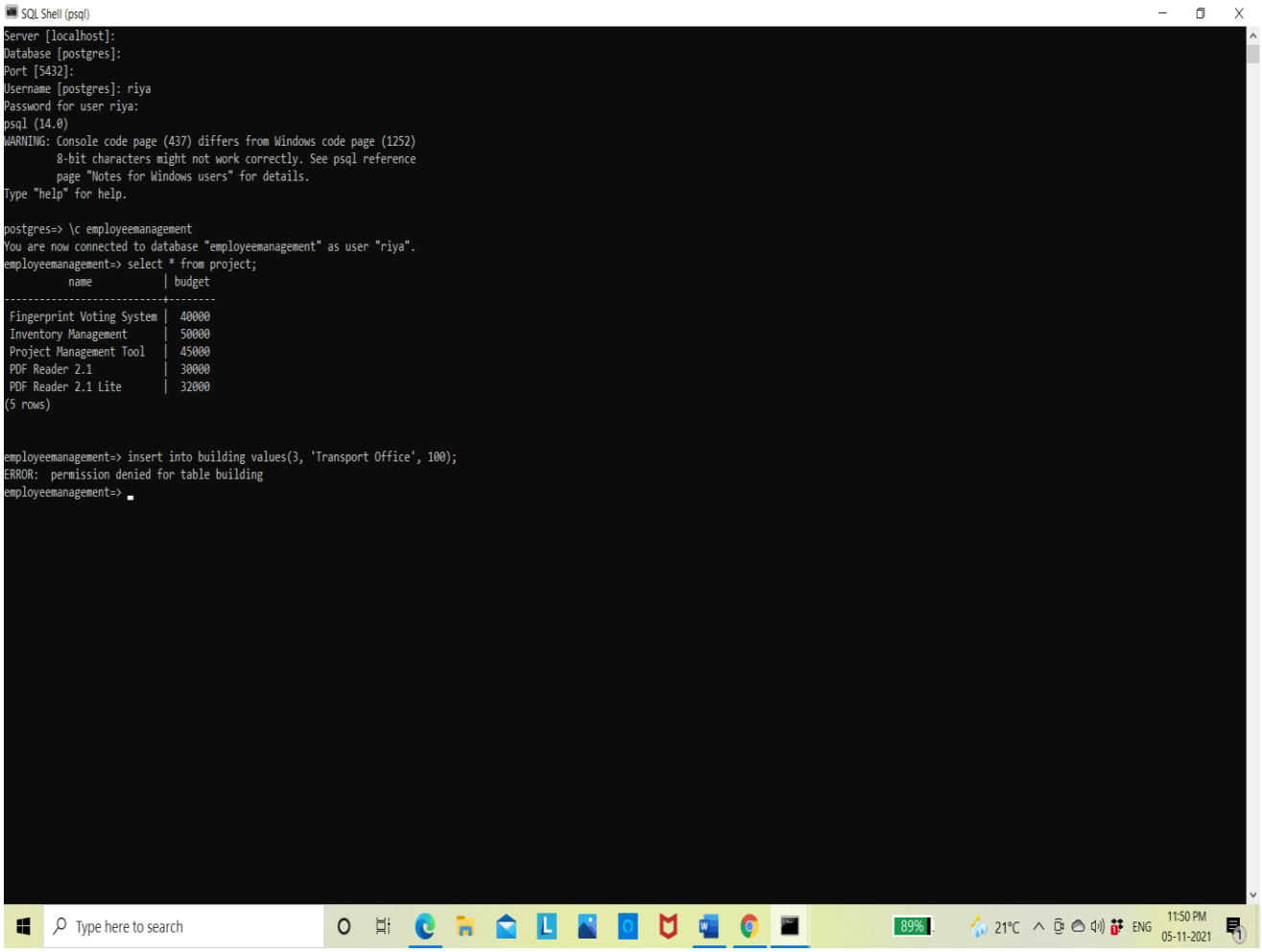
Meenakshi:



Rahul:



Riya:



While granting privileges major factors like referential integrity, dependence of relations, reduced complexity etc. was considered.

CONTRIBUTIONS

NAME	SRN	CONTRIBUTION	TIME SPENT
Lalitha Sravanti Dasu	PES2UG19CS201	Simple queries, complex queries, documentation	3 hours
Lohith Srinivas	PES2UG19CS204	User privileges **(Will be submitted with next assignment)**	3 hours
Meenakshi Suresh	PES2UG19CS220	Simple queries, complex queries, documentation	3 hours