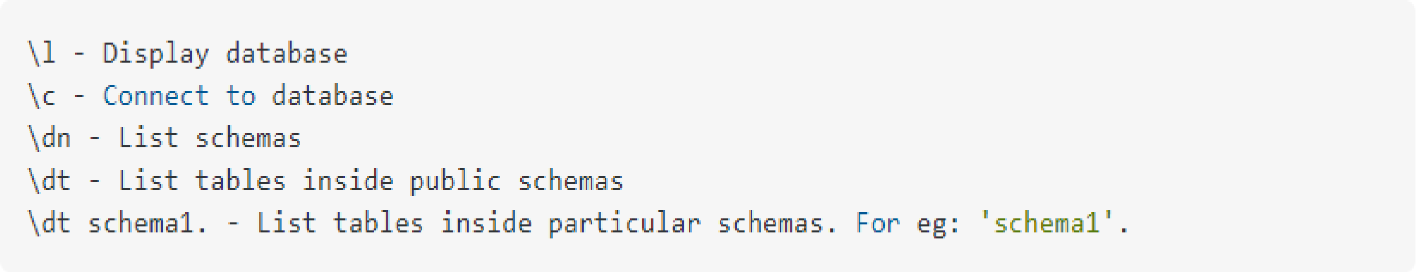
POSTGRESQL DAY2

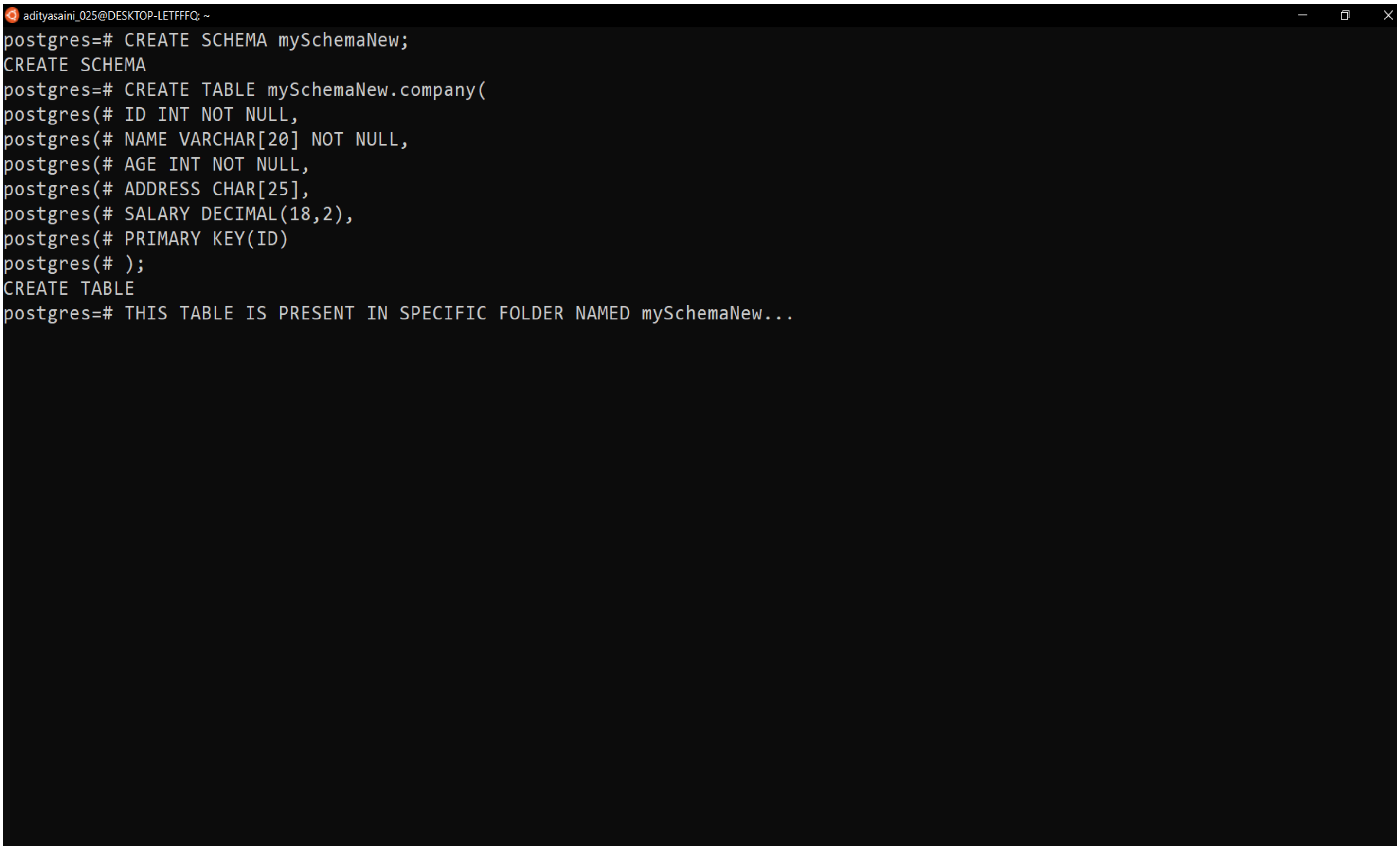
Schema (folders inside drive. Drive is our database)-: **As we create a database, we create it in drive. But, as in our system, we store files in folders similarly in PostgreSQL we create folders to store our database and create files inside the folder. This is known as Schema. Benefits of creating a schema if we have to give access to some file to second person, instead of giving access to complete drive, we can give access of one folder, and he/she can only read and write in that folder only. Another benefit of creating a schema is that our files are stored in a systematically organized way.**

SOME IMPORTANT COMMANDS TO DISPLAY ITEMS.



```
\l - Display database
\c - Connect to database
\dn - List schemas
\dt - List tables inside public schemas
\dt schema1. - List tables inside particular schemas. For eg: 'schema1'.
```

1. To create a schema and make a table inside it we use the following commands-:

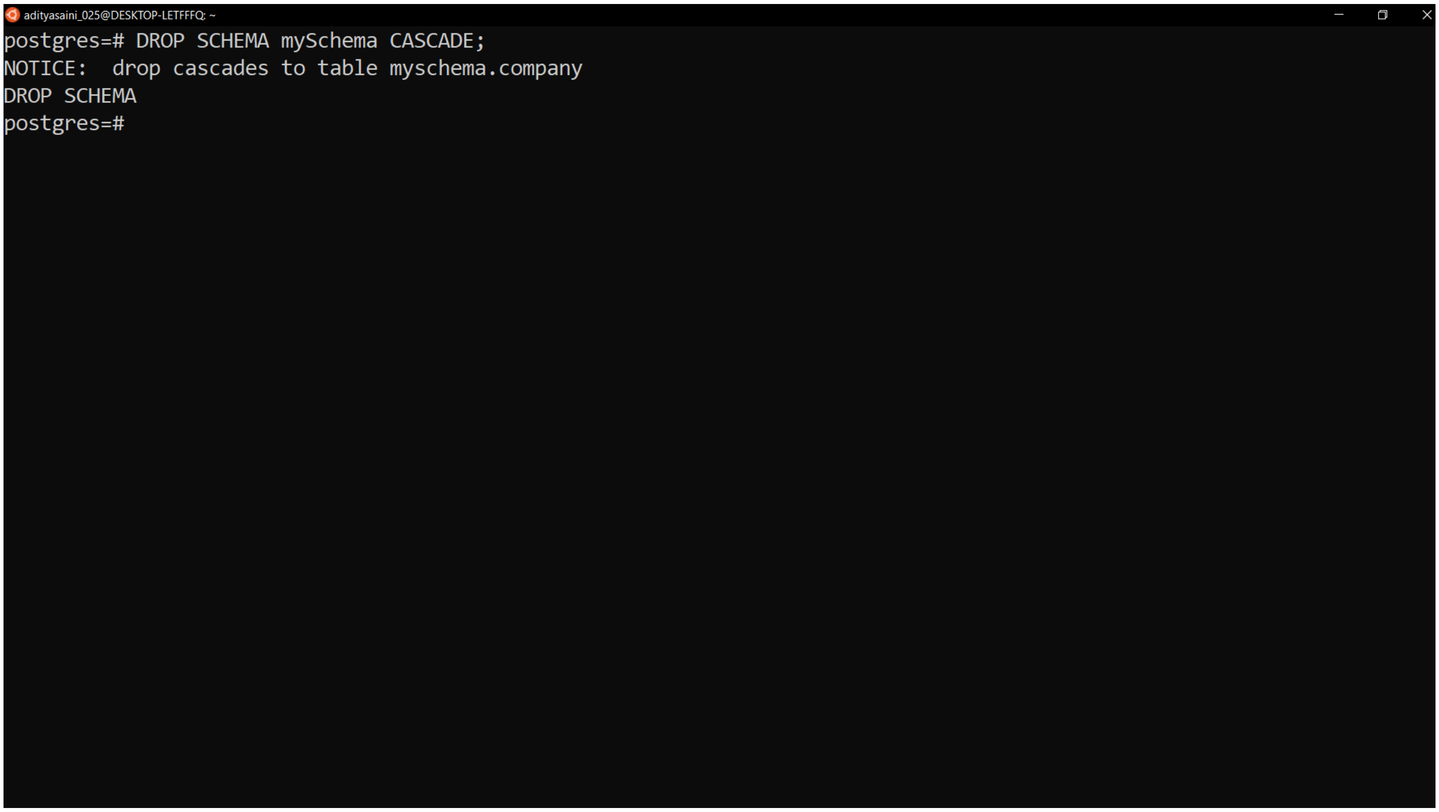


```
postgres=# CREATE SCHEMA mySchemaNew;
CREATE SCHEMA
postgres=# CREATE TABLE mySchemaNew.company(
postgres(# ID INT NOT NULL,
postgres(# NAME VARCHAR[20] NOT NULL,
postgres(# AGE INT NOT NULL,
postgres(# ADDRESS CHAR[25],
postgres(# SALARY DECIMAL(18,2),
postgres(# PRIMARY KEY(ID)
postgres(# );
CREATE TABLE
postgres=# THIS TABLE IS PRESENT IN SPECIFIC FOLDER NAMED mySchemaNew...
```

2. To drop the Schema, we need to write the following command-:

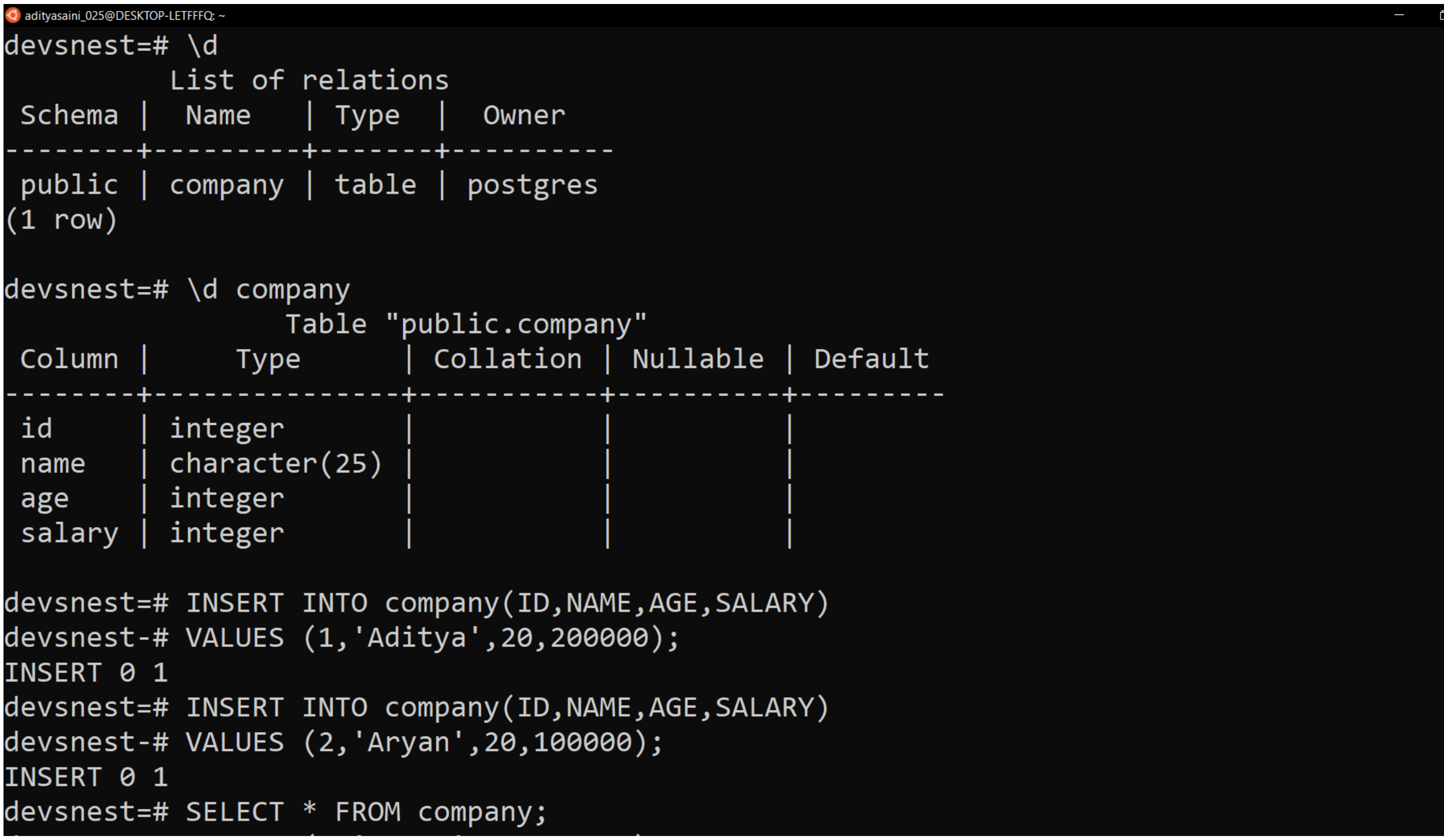
DROP SCHEMA mySchema (name of the schema) CASCADE;

CASCADE helps in deleting the complete schema folder.



```
postgres=# DROP SCHEMA mySchema CASCADE;
NOTICE:  drop cascades to table myschema.company
DROP SCHEMA
postgres=#
```

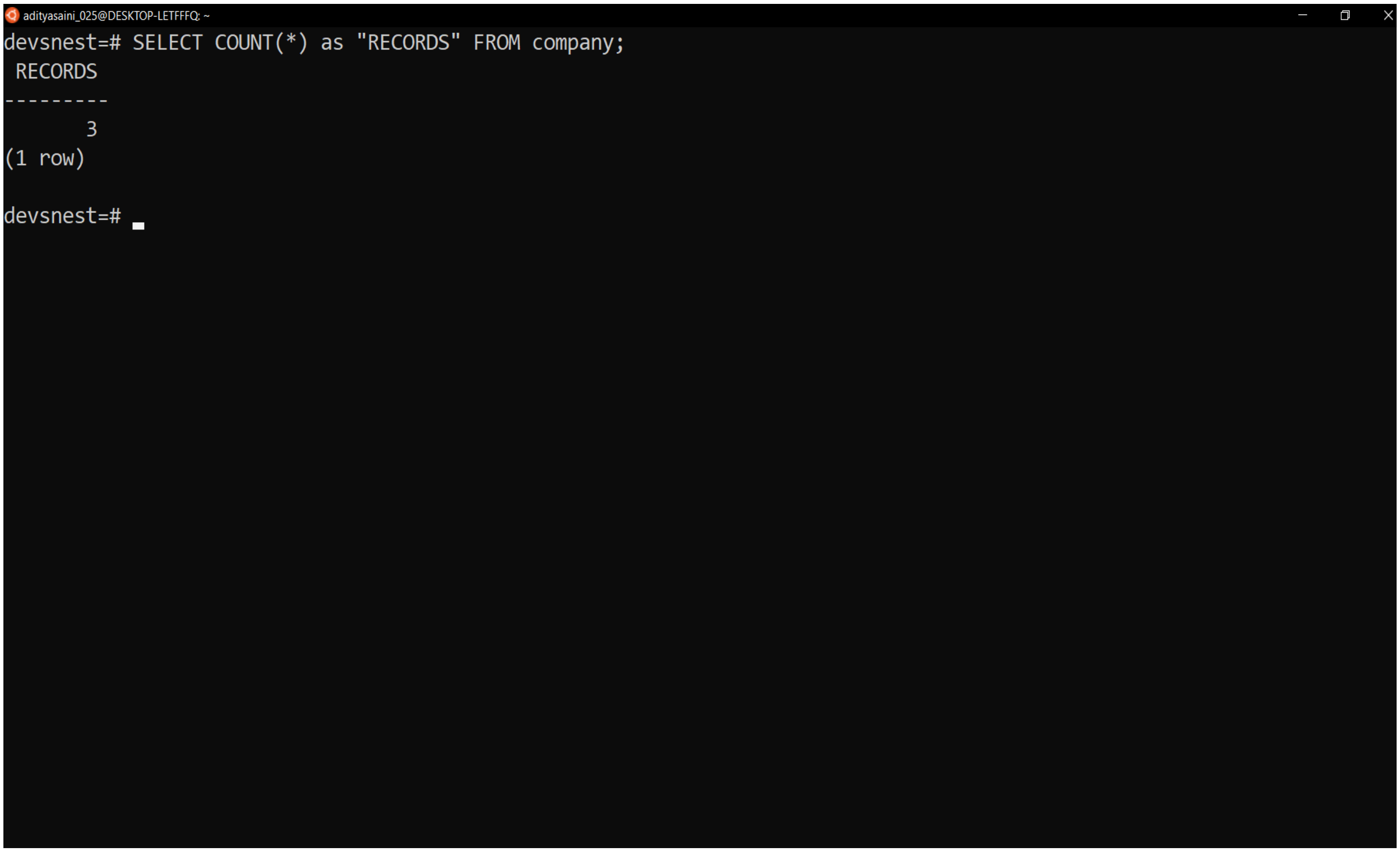
3. To insert items in a table we use the following commands-:



```
devsnest=# \d
          List of relations
 Schema |  Name   | Type  |  Owner
--------+---------+-------+----------
 public | company | table | postgres
(1 row)

devsnest=# \d company
                 Table "public.company"
 Column |     Type      | Collation | Nullable | Default
--------+---------------+-----------+----------+---------
 id     | integer       |           |          |
 name   | character(25) |           |          |
 age    | integer       |           |          |
 salary | integer       |           |          |

devsnest=# INSERT INTO company(ID,NAME,AGE,SALARY)
devsnest-# VALUES (1,'Aditya',20,200000);
INSERT 0 1
devsnest=# INSERT INTO company(ID,NAME,AGE,SALARY)
devsnest-# VALUES (2,'Aryan',20,100000);
INSERT 0 1
devsnest=# SELECT * FROM company;
```

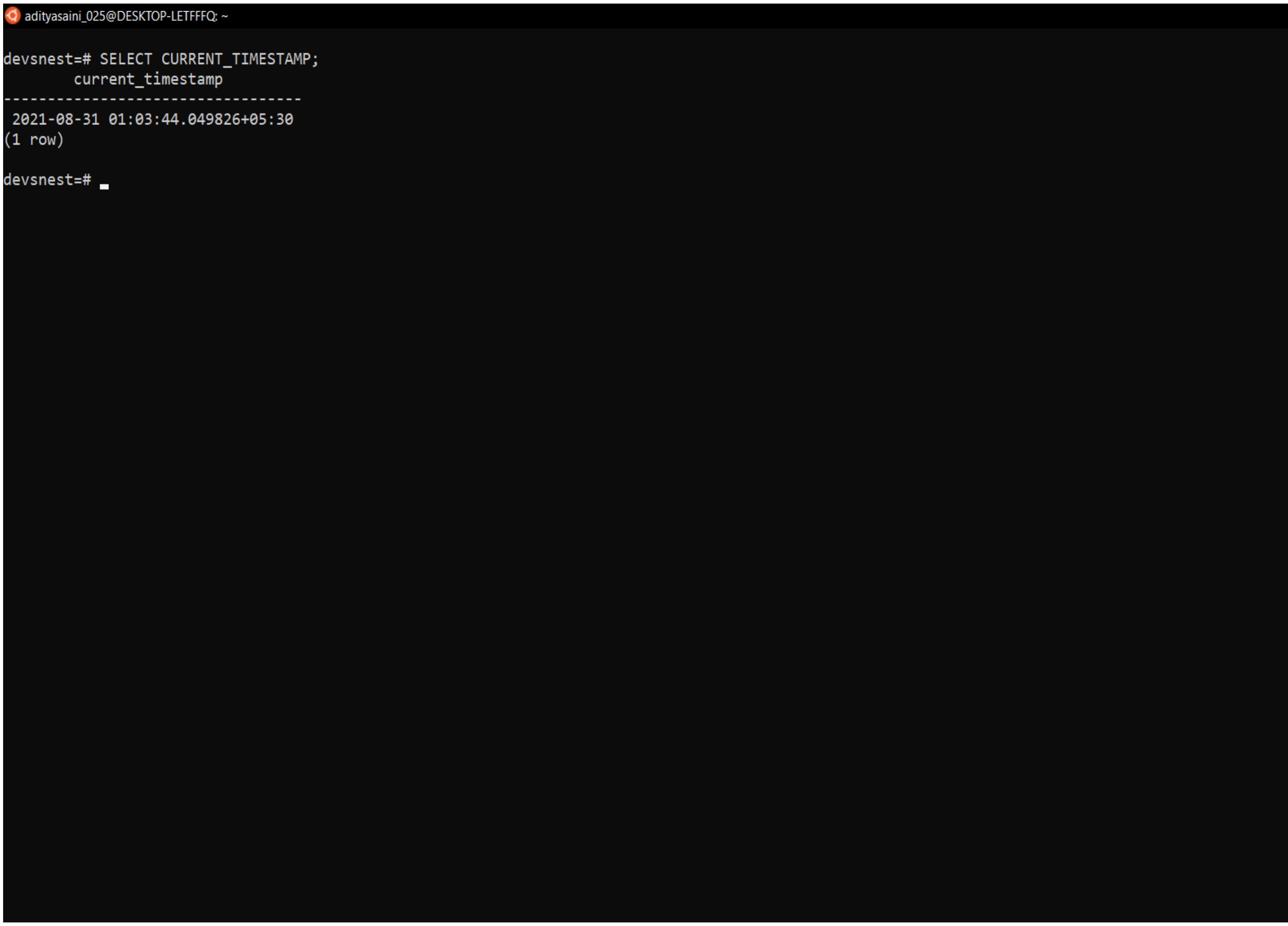
4. To delete the table from the database we write, DROP TABLE IF EXISTS company;
5. To see the data of the table we write, SELECT * FROM company;
6. We can also use SELECT keyword to perform calculations, SELECT (15+2) as addition; and so on.
7. We print total number of records present in our table, SELECT COUNT (*) as “RECORDS” FROM company;

HERE WE CAN WRITE ANY SPECIFIC NAME OF THE COLOUM LIKE ID, NAME etc. AND ALSO RECORD HERE IS JUST A TEXT WHICH ONLY SHOWED UP ABOVE THE PRINTED VALUE.



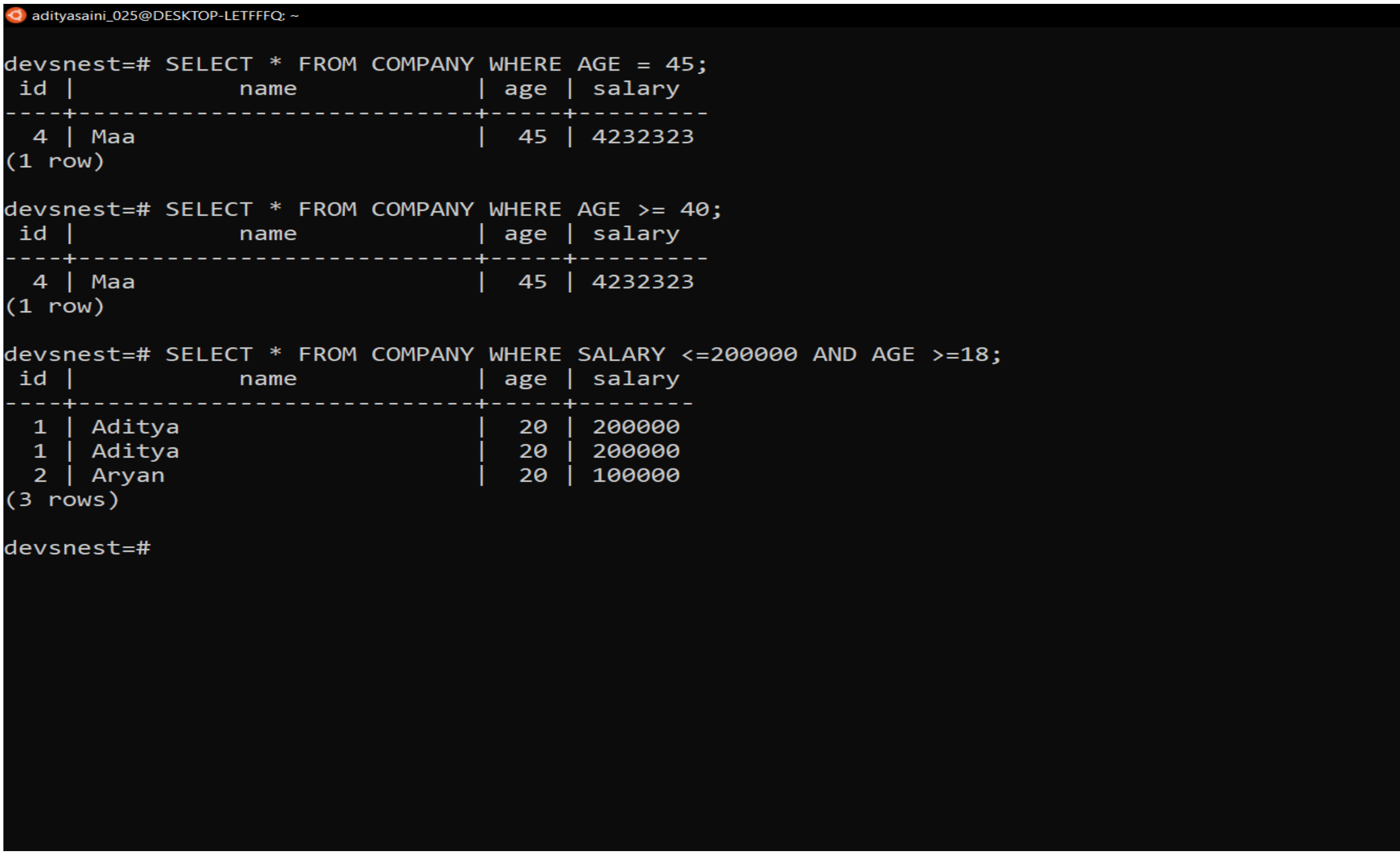
```
devsnest=# SELECT COUNT(*) as "RECORDS" FROM company;
 RECORDS
---------
       3
(1 row)

devsnest=# 
```

8. PostgreSQL is not only used to perform SQL commands but also can be used to perform C, C++, python commands. For example,

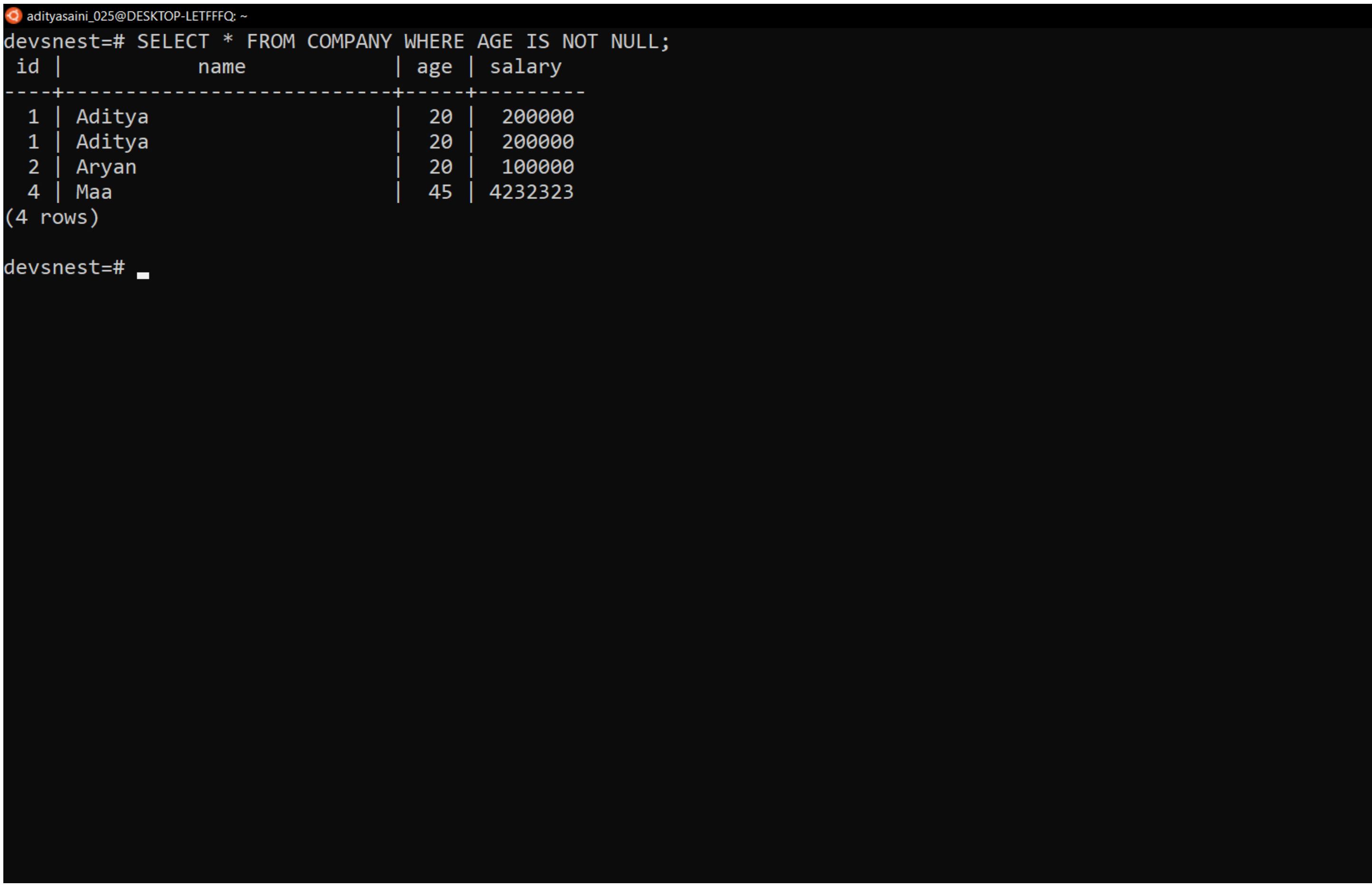


```
adityasaini_025@DESKTOP-LETFFFQ: ~

devsnest=# SELECT CURRENT_TIMESTAMP;
        current_timestamp
----------------------------------
 2021-08-31 01:03:44.049826+05:30
(1 row)

devsnest=#
```

9. Now we use if conditions in PostgreSQL by using keyword WHERE, for example-:

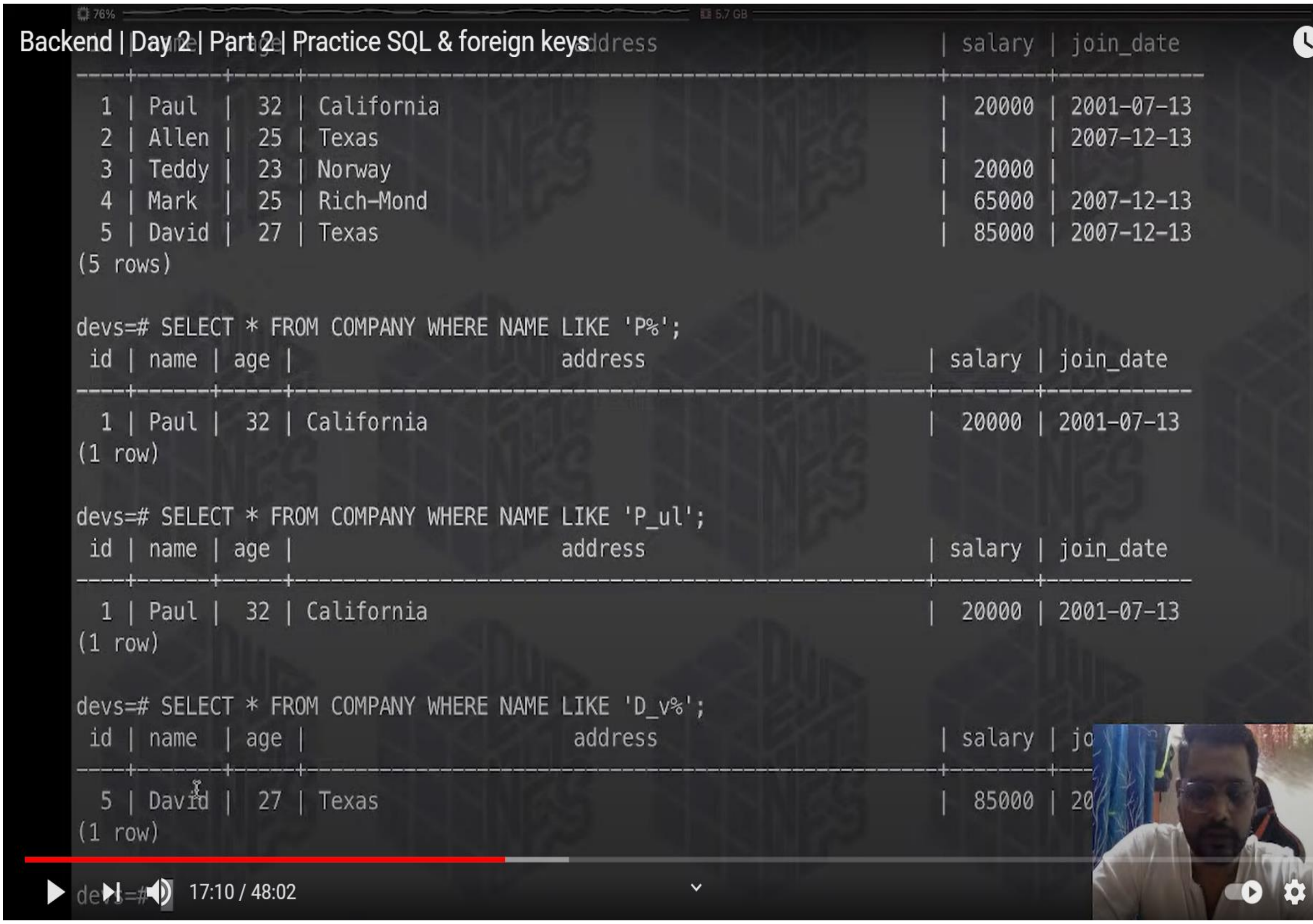


```
devsnest=# SELECT * FROM COMPANY WHERE AGE = 45;
 id |            name            | age | salary
----+----------------------------+-----+---------
  4 | Maa                        |  45 | 4232323
(1 row)

devsnest=# SELECT * FROM COMPANY WHERE AGE >= 40;
 id |            name            | age | salary
----+----------------------------+-----+---------
  4 | Maa                        |  45 | 4232323
(1 row)

devsnest=# SELECT * FROM COMPANY WHERE SALARY <=200000 AND AGE >=18;
 id |            name            | age | salary
----+----------------------------+-----+--------
  1 | Aditya                     |  20 | 200000
  1 | Aditya                     |  20 | 200000
  2 | Aryan                      |  20 | 100000
(3 rows)

devsnest=#
```



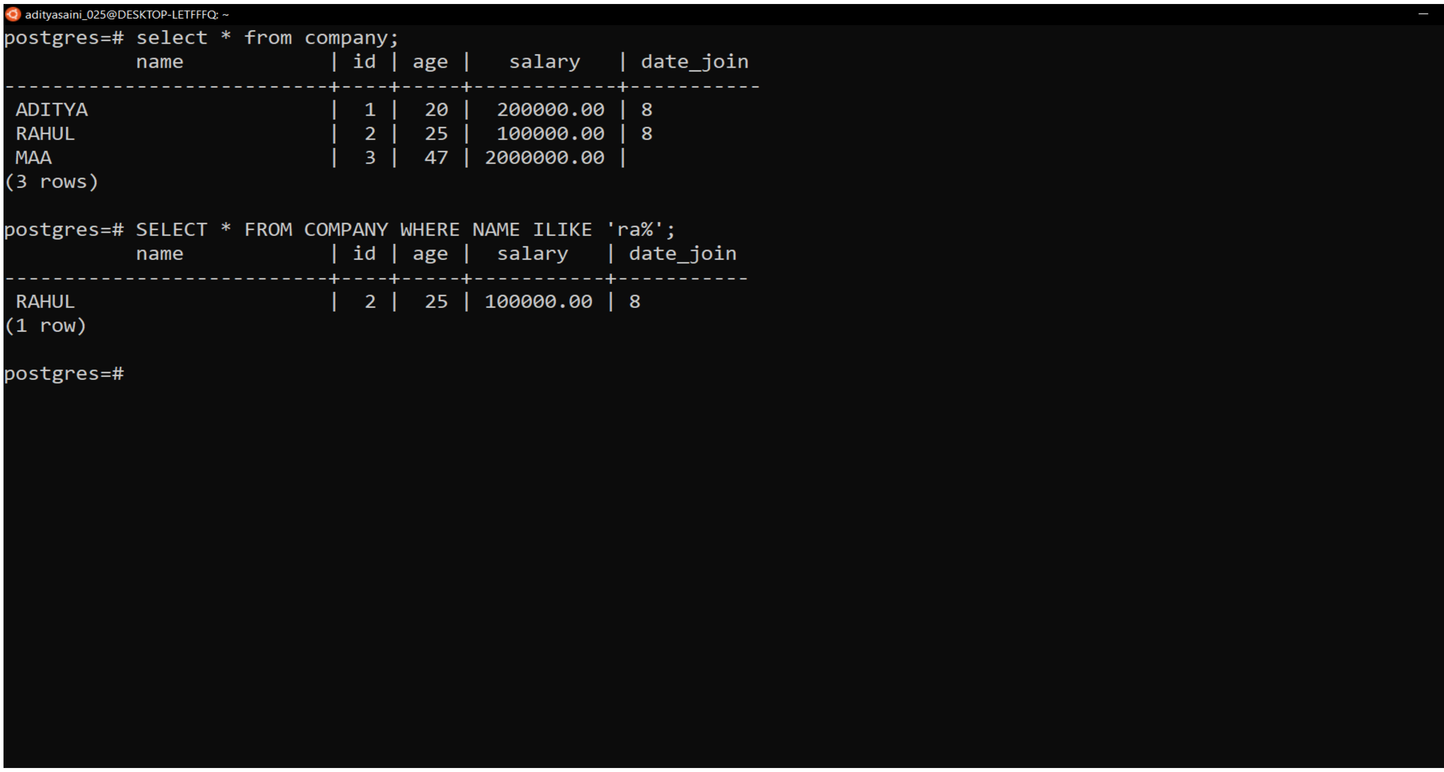
```
devsnest=# SELECT * FROM COMPANY WHERE AGE IS NOT NULL;
 id |           name            | age | salary
----+---------------------------+-----+---------
  1 | Aditya                    |  20 |  200000
  1 | Aditya                    |  20 |  200000
  2 | Aryan                     |  20 |  100000
  4 | Maa                       |  45 | 4232323
(4 rows)

devsnest=# _
```

10. Sometimes we have to search the name starting with some characters or sometimes we just need to find the names with first and any other character so we use, SELECT * FROM COMPANY WHERE NAME LIKE ‘A%’; or we use, SELECT * FROM COMPANY WHERE NAME LIKE ‘A_D%’;



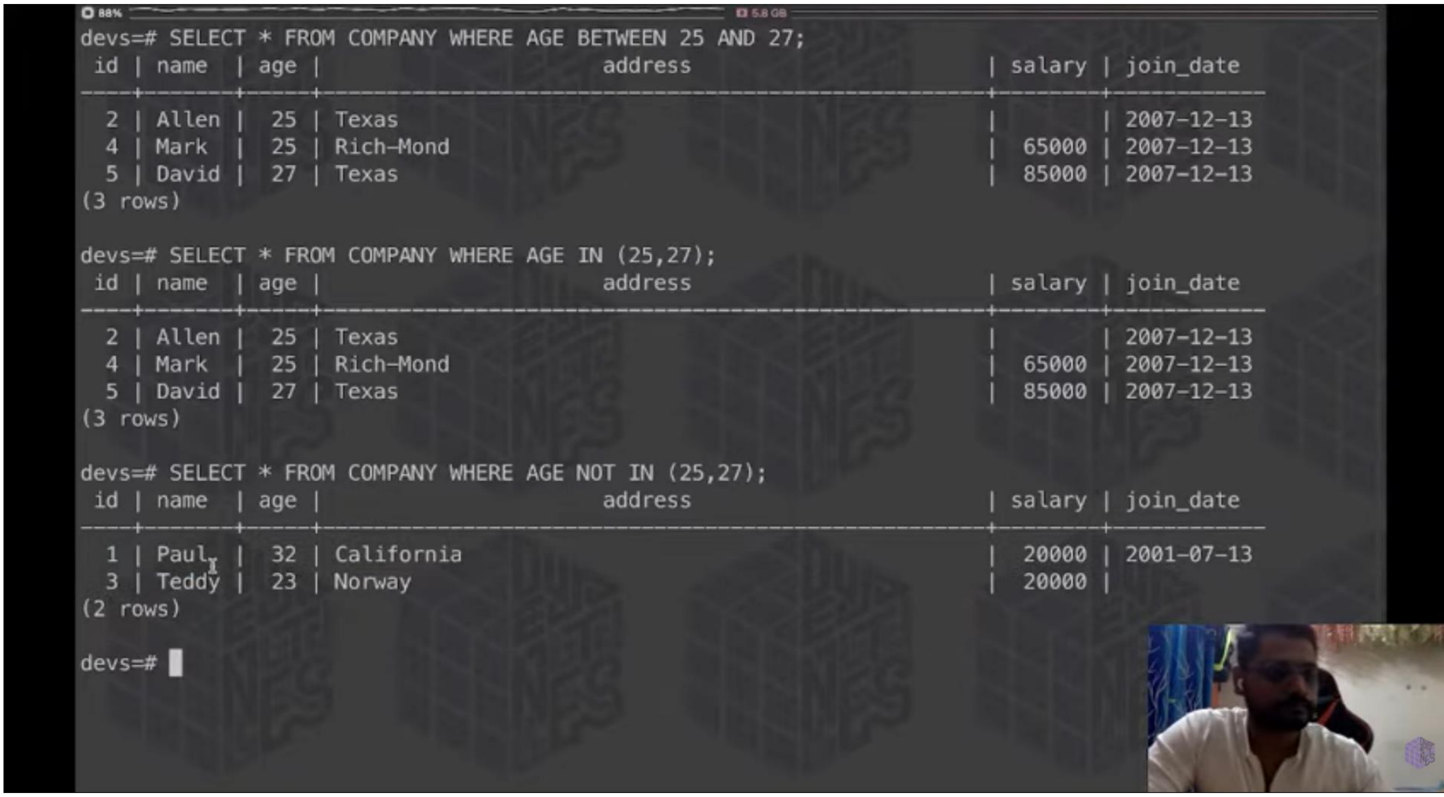
```
 id | name  | age |                     address                      | salary | join_date
----+-------+-----+--------------------------------------------------+--------+------------
  1 | Paul  |  32 | California                                       |  20000 | 2001-07-13
  2 | Allen |  25 | Texas                                            |        | 2007-12-13
  3 | Teddy |  23 | Norway                                           |  20000 |
  4 | Mark  |  25 | Rich-Mond                                        |  65000 | 2007-12-13
  5 | David |  27 | Texas                                            |  85000 | 2007-12-13
(5 rows)

devs=# SELECT * FROM COMPANY WHERE NAME LIKE 'P%';
 id | name | age |                     address                      | salary | join_date
----+------+-----+--------------------------------------------------+--------+------------
  1 | Paul |  32 | California                                       |  20000 | 2001-07-13
(1 row)

devs=# SELECT * FROM COMPANY WHERE NAME LIKE 'P_ul';
 id | name | age |                     address                      | salary | join_date
----+------+-----+--------------------------------------------------+--------+------------
  1 | Paul |  32 | California                                       |  20000 | 2001-07-13
(1 row)

devs=# SELECT * FROM COMPANY WHERE NAME LIKE 'D_v%';
 id | name  | age |                     address                      | salary | jo
----+-------+-----+--------------------------------------------------+--------+---
  5 | David |  27 | Texas                                            |  85000 | 20
(1 row)

devs=#
```

11. Sometimes we do not know the name of the person is entered in capital letters or small letters, so we use one special keyword ILIKE, for example-:



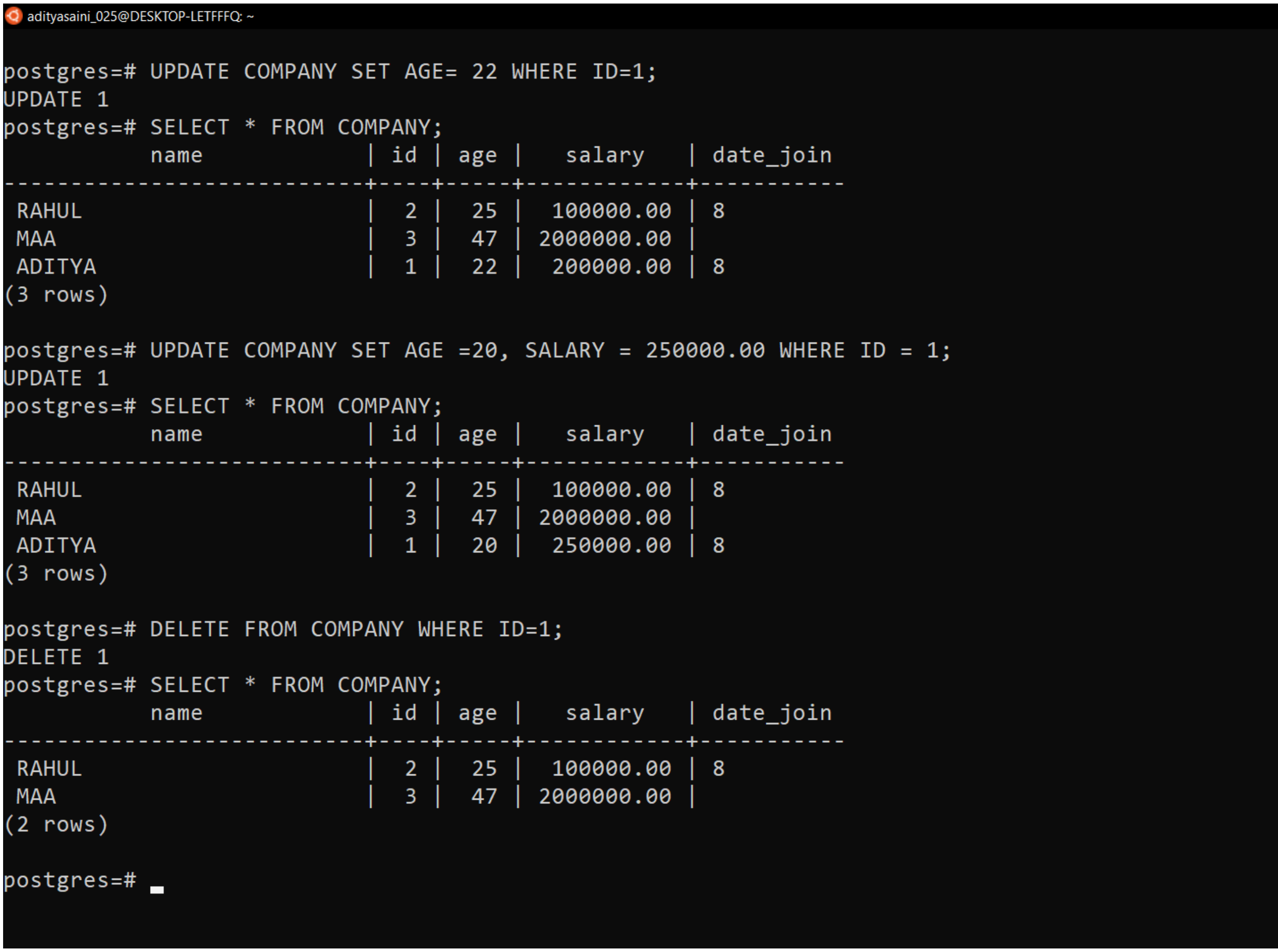
```
postgres=# select * from company;
          name            | id | age |   salary   | date_join
--------------------------+----+-----+------------+-----------
 ADITYA                   |  1 |  20 |  200000.00 | 8
 RAHUL                    |  2 |  25 |  100000.00 | 8
 MAA                      |  3 |  47 | 2000000.00 |
(3 rows)

postgres=# SELECT * FROM COMPANY WHERE NAME ILIKE 'ra%';
          name            | id | age |  salary   | date_join
--------------------------+----+-----+-----------+-----------
 RAHUL                    |  2 |  25 | 100000.00 | 8
(1 row)

postgres=#
```

12. Now to search for the rows in a particular range we generally write following commands-:



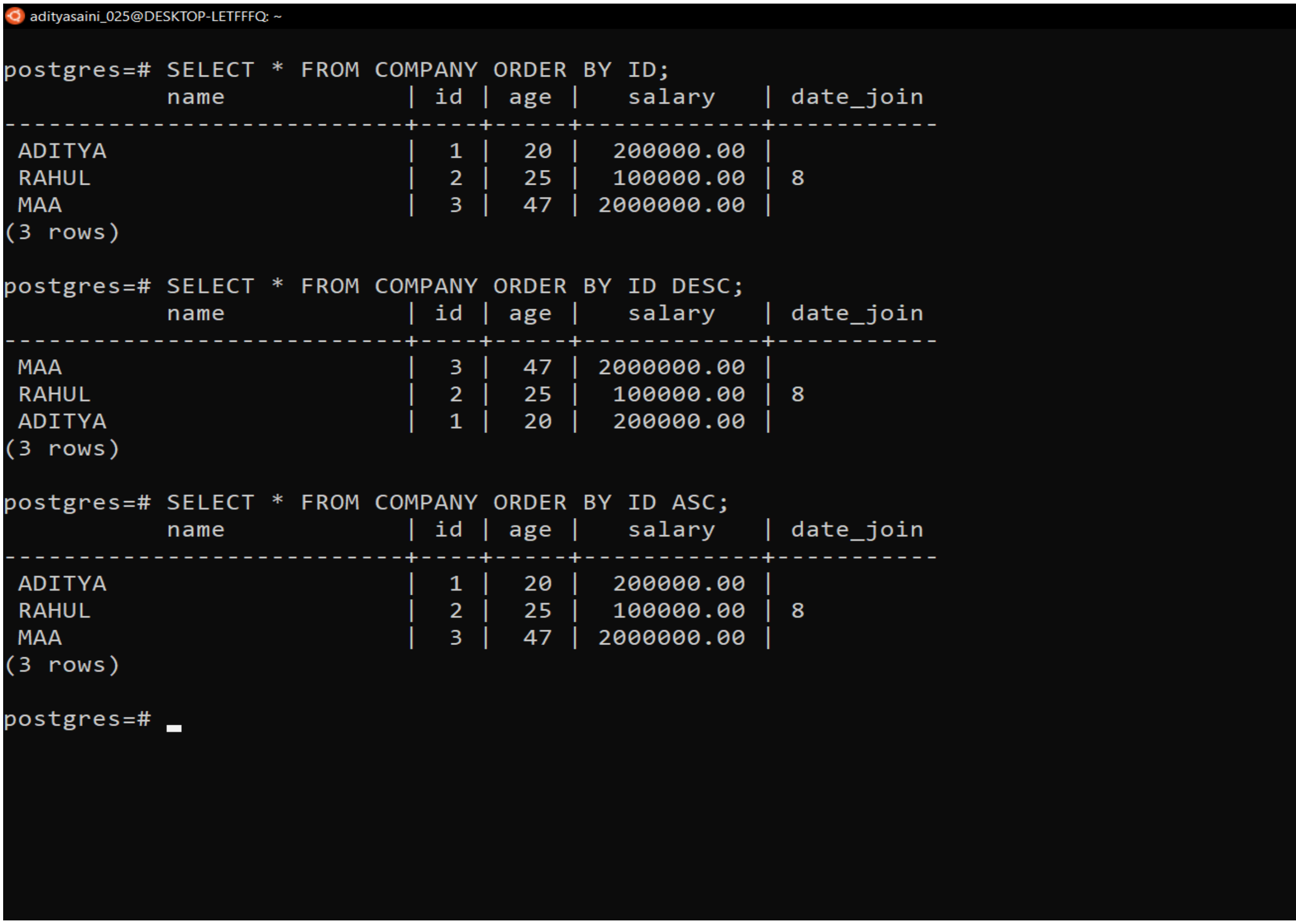
```
devs=# SELECT * FROM COMPANY WHERE AGE BETWEEN 25 AND 27;
 id | name  | age |                       address                        | salary | join_date
----+-------+-----+------------------------------------------------------+--------+------------
  2 | Allen |  25 | Texas                                                |        | 2007-12-13
  4 | Mark  |  25 | Rich-Mond                                            |  65000 | 2007-12-13
  5 | David |  27 | Texas                                                |  85000 | 2007-12-13
(3 rows)

devs=# SELECT * FROM COMPANY WHERE AGE IN (25,27);
 id | name  | age |                       address                        | salary | join_date
----+-------+-----+------------------------------------------------------+--------+------------
  2 | Allen |  25 | Texas                                                |        | 2007-12-13
  4 | Mark  |  25 | Rich-Mond                                            |  65000 | 2007-12-13
  5 | David |  27 | Texas                                                |  85000 | 2007-12-13
(3 rows)

devs=# SELECT * FROM COMPANY WHERE AGE NOT IN (25,27);
 id | name  | age |                       address                        | salary | join_date
----+-------+-----+------------------------------------------------------+--------+------------
  1 | Paul  |  32 | California                                           |  20000 | 2001-07-13
  3 | Teddy |  23 | Norway                                               |  20000 |
(2 rows)

devs=#
```

13. To update the table with some other values we use keyword UPDATE, for example:



```
postgres=# UPDATE COMPANY SET AGE= 22 WHERE ID=1;
UPDATE 1
postgres=# SELECT * FROM COMPANY;
           name            | id | age |   salary   | date_join
---------------------------+----+-----+------------+-----------
 RAHUL                     |  2 |  25 |  100000.00 | 8
 MAA                       |  3 |  47 | 2000000.00 |
 ADITYA                    |  1 |  22 |  200000.00 | 8
(3 rows)

postgres=# UPDATE COMPANY SET AGE =20, SALARY = 250000.00 WHERE ID = 1;
UPDATE 1
postgres=# SELECT * FROM COMPANY;
           name            | id | age |   salary   | date_join
---------------------------+----+-----+------------+-----------
 RAHUL                     |  2 |  25 |  100000.00 | 8
 MAA                       |  3 |  47 | 2000000.00 |
 ADITYA                    |  1 |  20 |  250000.00 | 8
(3 rows)

postgres=# DELETE FROM COMPANY WHERE ID=1;
DELETE 1
postgres=# SELECT * FROM COMPANY;
           name            | id | age |   salary   | date_join
---------------------------+----+-----+------------+-----------
 RAHUL                     |  2 |  25 |  100000.00 | 8
 MAA                       |  3 |  47 | 2000000.00 |
(2 rows)

postgres=# _
```

14. To make the table in a particular order we write following commands-:



```
postgres=# SELECT * FROM COMPANY ORDER BY ID;
           name            | id | age |   salary    | date_join
---------------------------+----+-----+-------------+-----------
 ADITYA                    |  1 |  20 |   200000.00 |
 RAHUL                     |  2 |  25 |   100000.00 | 8
 MAA                       |  3 |  47 |  2000000.00 |
(3 rows)

postgres=# SELECT * FROM COMPANY ORDER BY ID DESC;
           name            | id | age |   salary    | date_join
---------------------------+----+-----+-------------+-----------
 MAA                       |  3 |  47 |  2000000.00 |
 RAHUL                     |  2 |  25 |   100000.00 | 8
 ADITYA                    |  1 |  20 |   200000.00 |
(3 rows)

postgres=# SELECT * FROM COMPANY ORDER BY ID ASC;
           name            | id | age |   salary    | date_join
---------------------------+----+-----+-------------+-----------
 ADITYA                    |  1 |  20 |   200000.00 |
 RAHUL                     |  2 |  25 |   100000.00 | 8
 MAA                       |  3 |  47 |  2000000.00 |
(3 rows)

postgres=#
```

15. To set the primary key, we write-:

ID INT PRIMARY KEY

16. FOREIGN KEY-: Foreign key act as a connector between the two tables. Most of the time we use ID as a foreign key as we have to select the unique key to connect the two tables with each other and ID is the only key which is unique most of the time. We set ID as a primary key.

All the class notes if given in this gist and also, I have forked this gist.

https://gist.github.com/rachit1994/6a17079c2089a2dfd3cb6362016915d0