DATABASE MANAGEMENT SYSTEM PROJECT

ZOO MANAGEMENT SYSTEM

ASSIGNMENT – 4

By:

Manasa R- PES2UG19CS215

Namitha Nayak: PES2UG19CS247

Nidhi Bharatiya: PES2UG19CS254

1) Write a paragraph suggesting the dependencies installed for the database connectivity(front end to back end).

For connecting front end to backend: python.

For GUI: tkinter

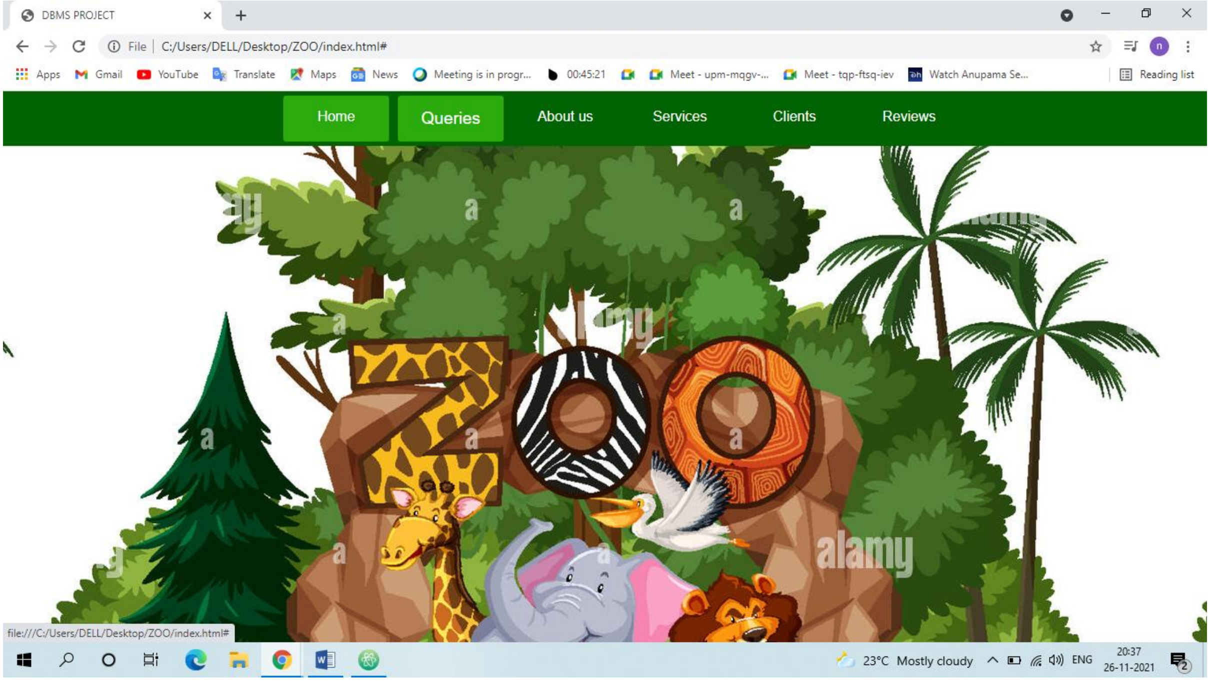
Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. Tkinter is included with standard GNU/Linux, Microsoft Windows and macOS installs of Python.

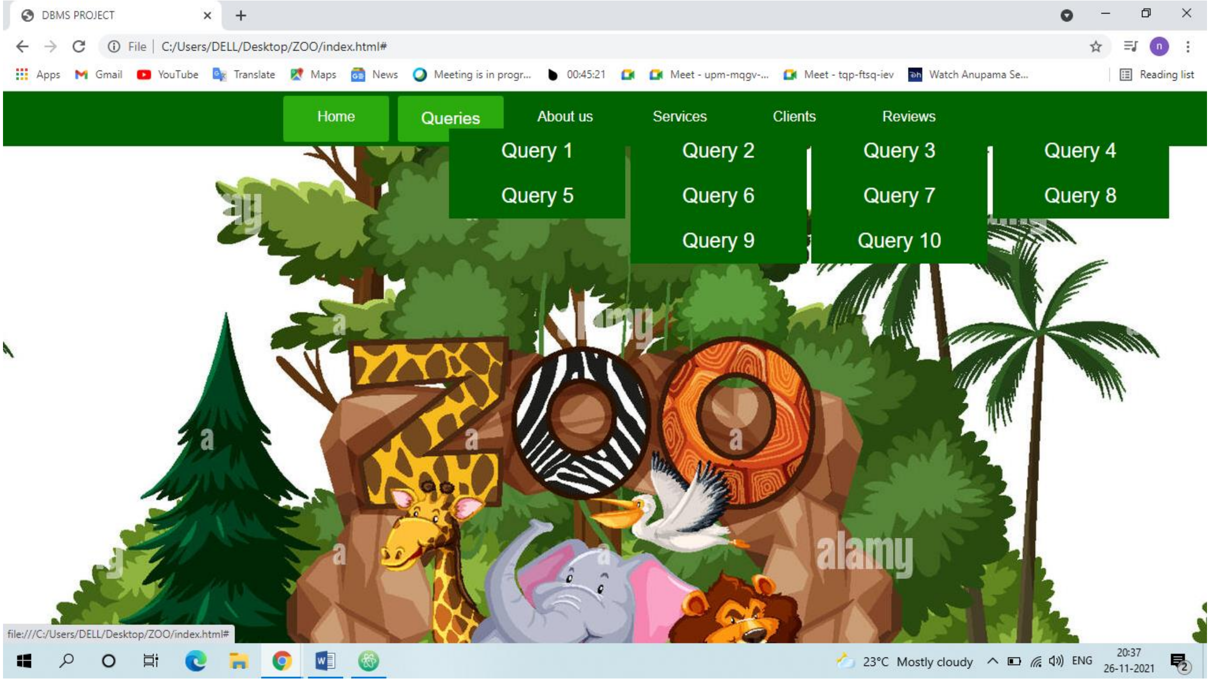
Backend: postgresql

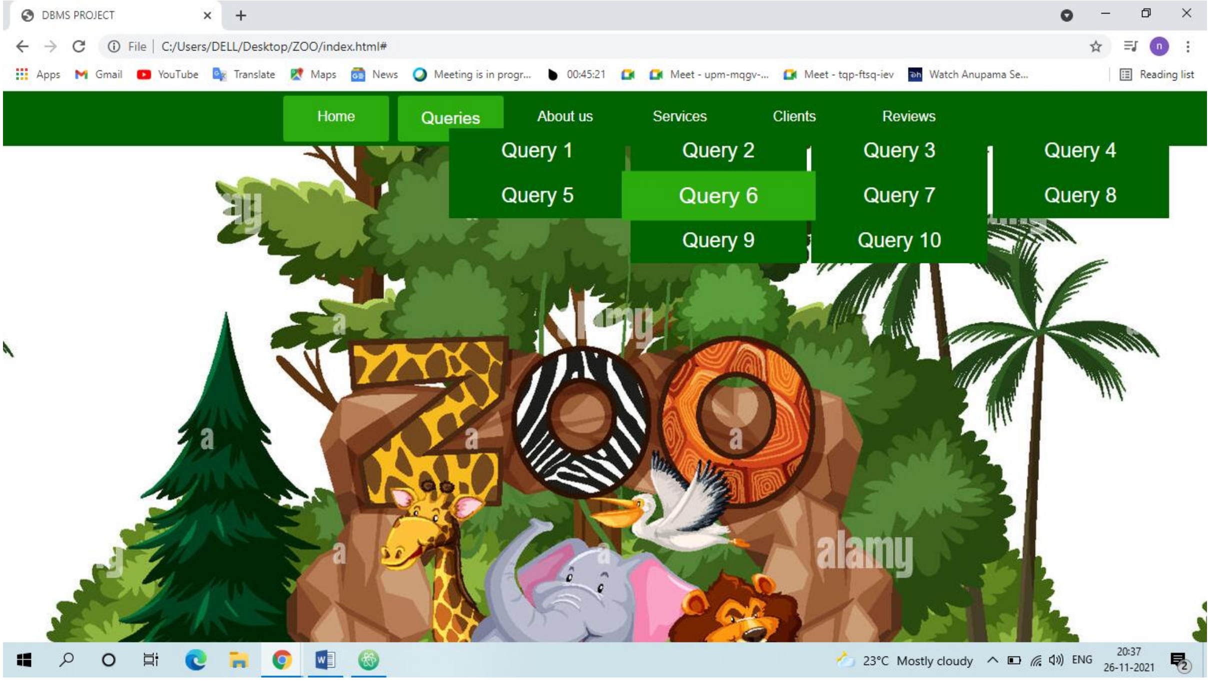
PostgreSQL, also known as Postgres, is a free and open-source relational database management system emphasizing extensibility and SQL compliance. It was originally named POSTGRES, referring to its origins as a successor to the Ingres database developed at the University of California, Berkeley.

2)Screenshots for the statements executed from the front end.

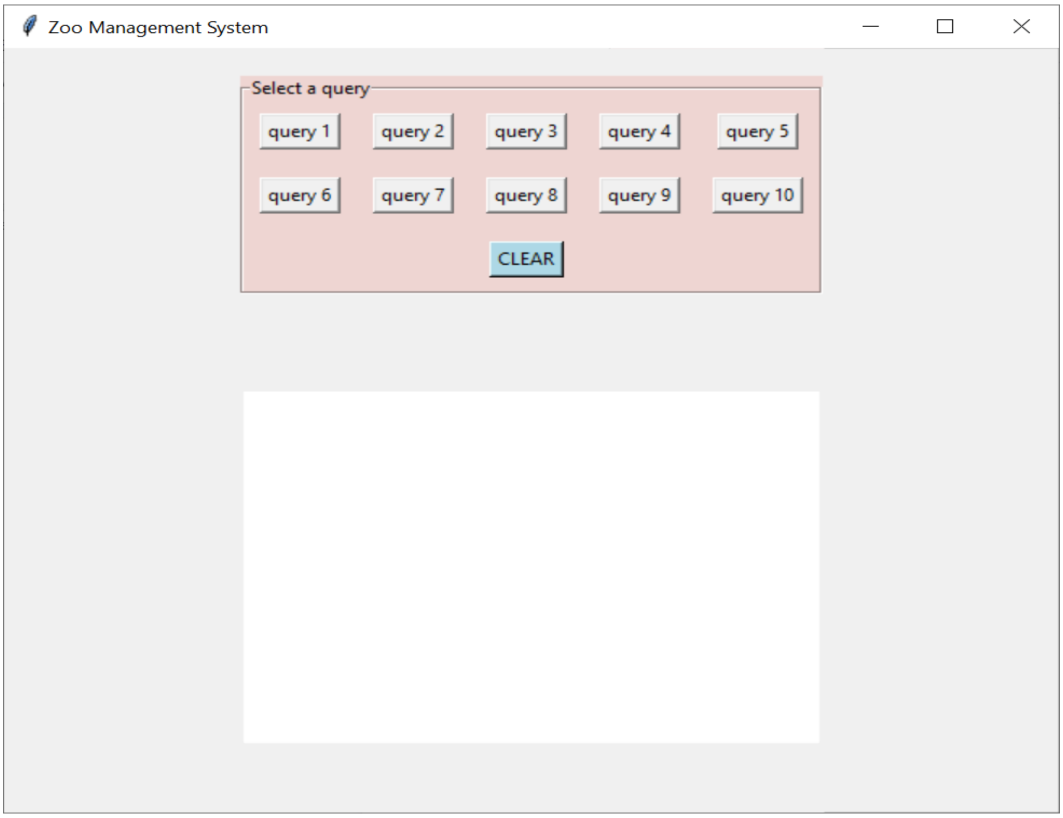
Front end html:

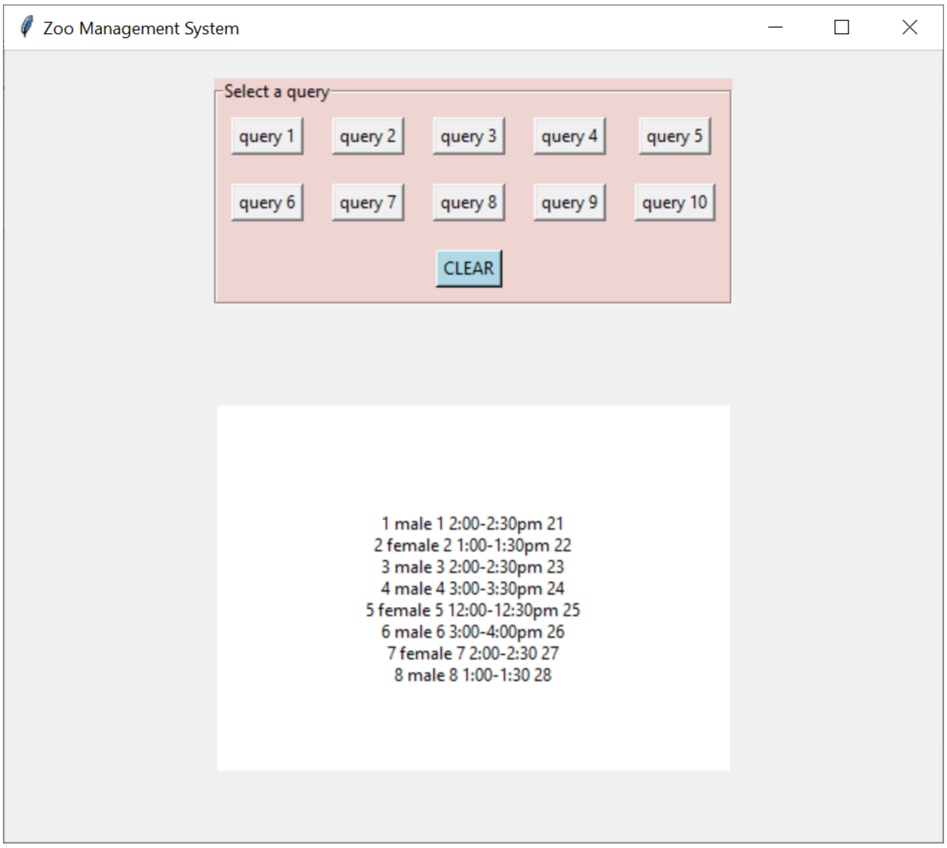

DBMS PROJECT
C:/Users/DELL/Desktop/ZOO/index.html#
Home
Queries
About us
Services
Clients
Reviews
file:///C:/Users/DELL/Desktop/ZOO/index.html#
23°C Mostly cloudy
20:37
26-11-2021


DBMS PROJECT
C:/Users/DELL/Desktop/ZOO/index.html#
Home
Queries
About us
Services
Clients
Reviews
Query 1
Query 2
Query 3
Query 4
Query 5
Query 6
Query 7
Query 8
Query 9
Query 10
file:///C:/Users/DELL/Desktop/ZOO/index.html#
23°C Mostly cloudy
20:37
26-11-2021

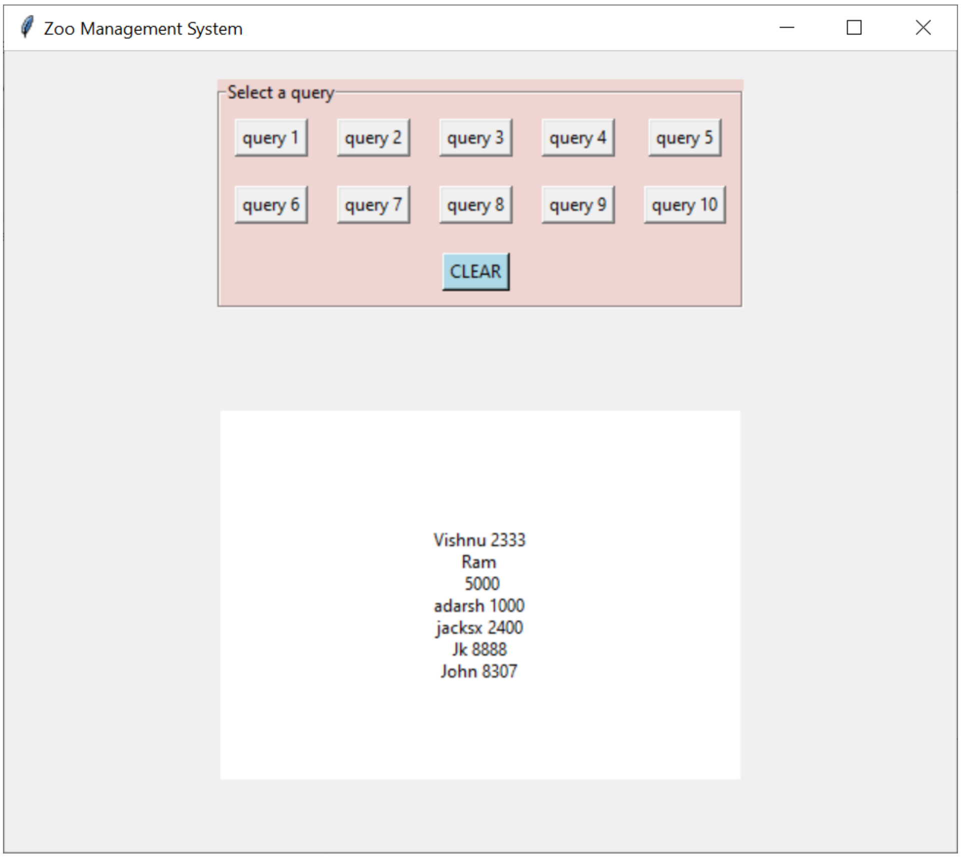


Front end tkinter:

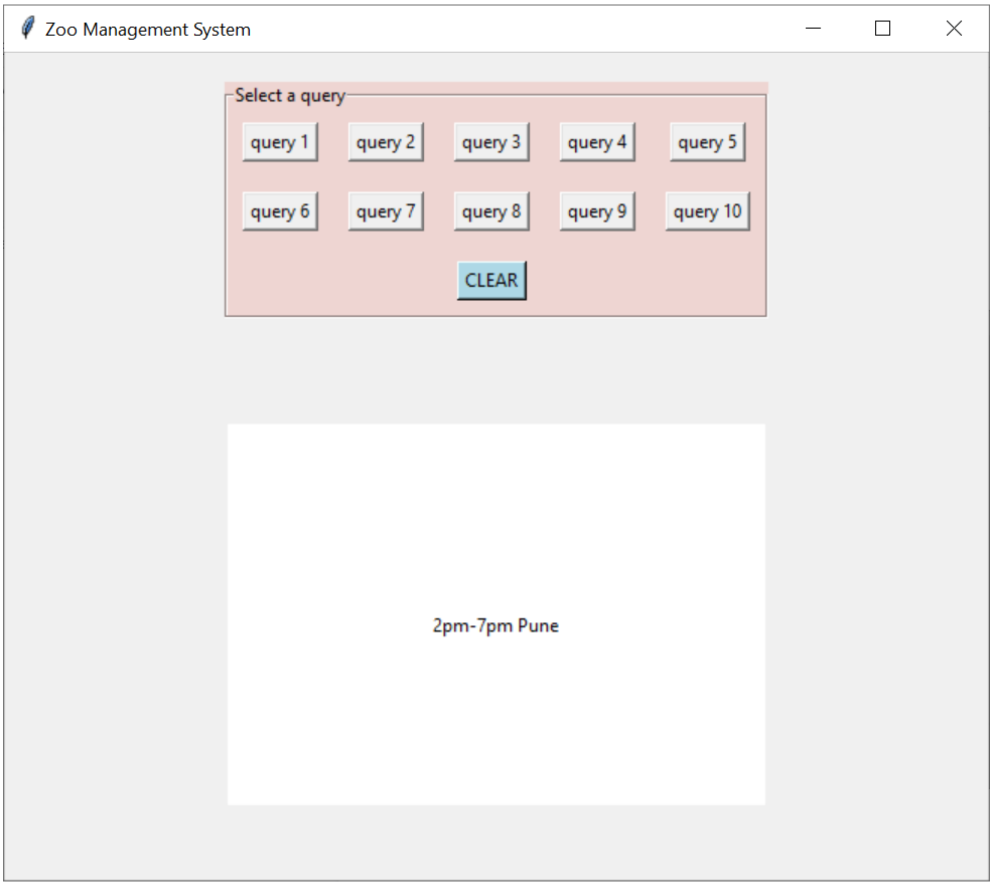




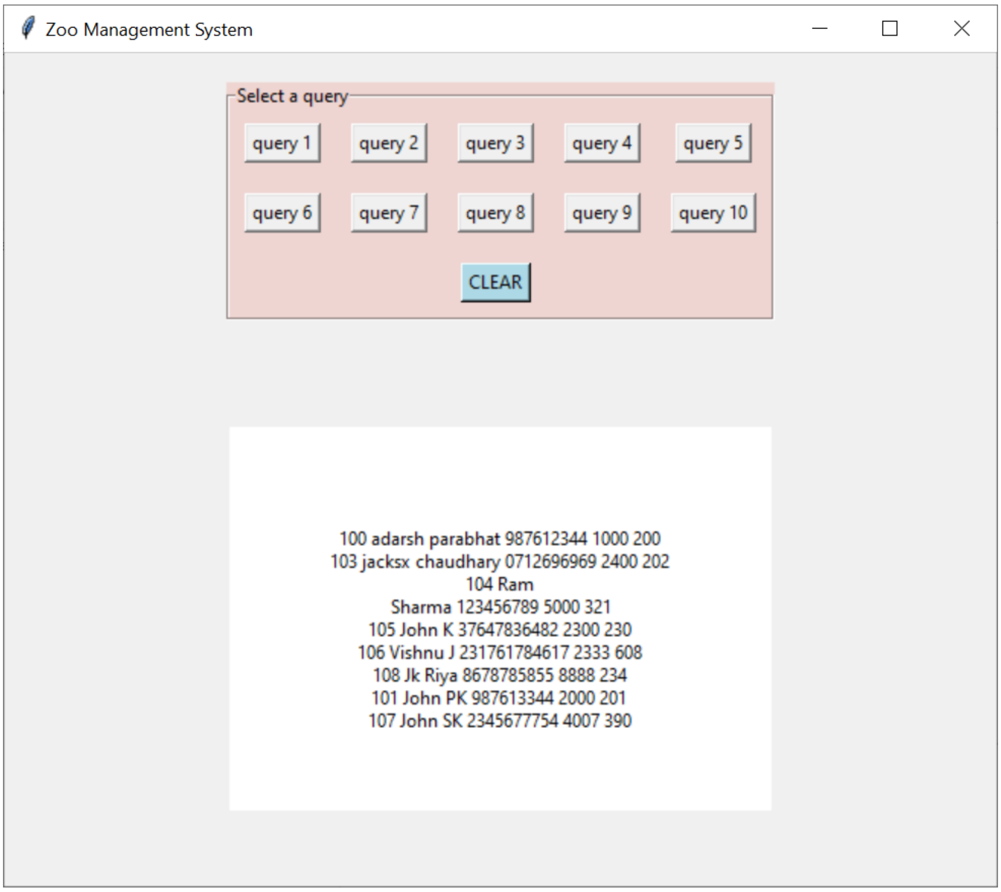
```
1 male 1 2:00-2:30pm 21
2 female 2 1:00-1:30pm 22
3 male 3 2:00-2:30pm 23
4 male 4 3:00-3:30pm 24
5 female 5 12:00-12:30pm 25
6 male 6 3:00-4:00pm 26
7 female 7 2:00-2:30 27
8 male 8 1:00-1:30 28
```



```
Vishnu 2333
Ram
5000
adarsh 1000
jacksx 2400
Jk 8888
John 8307
```

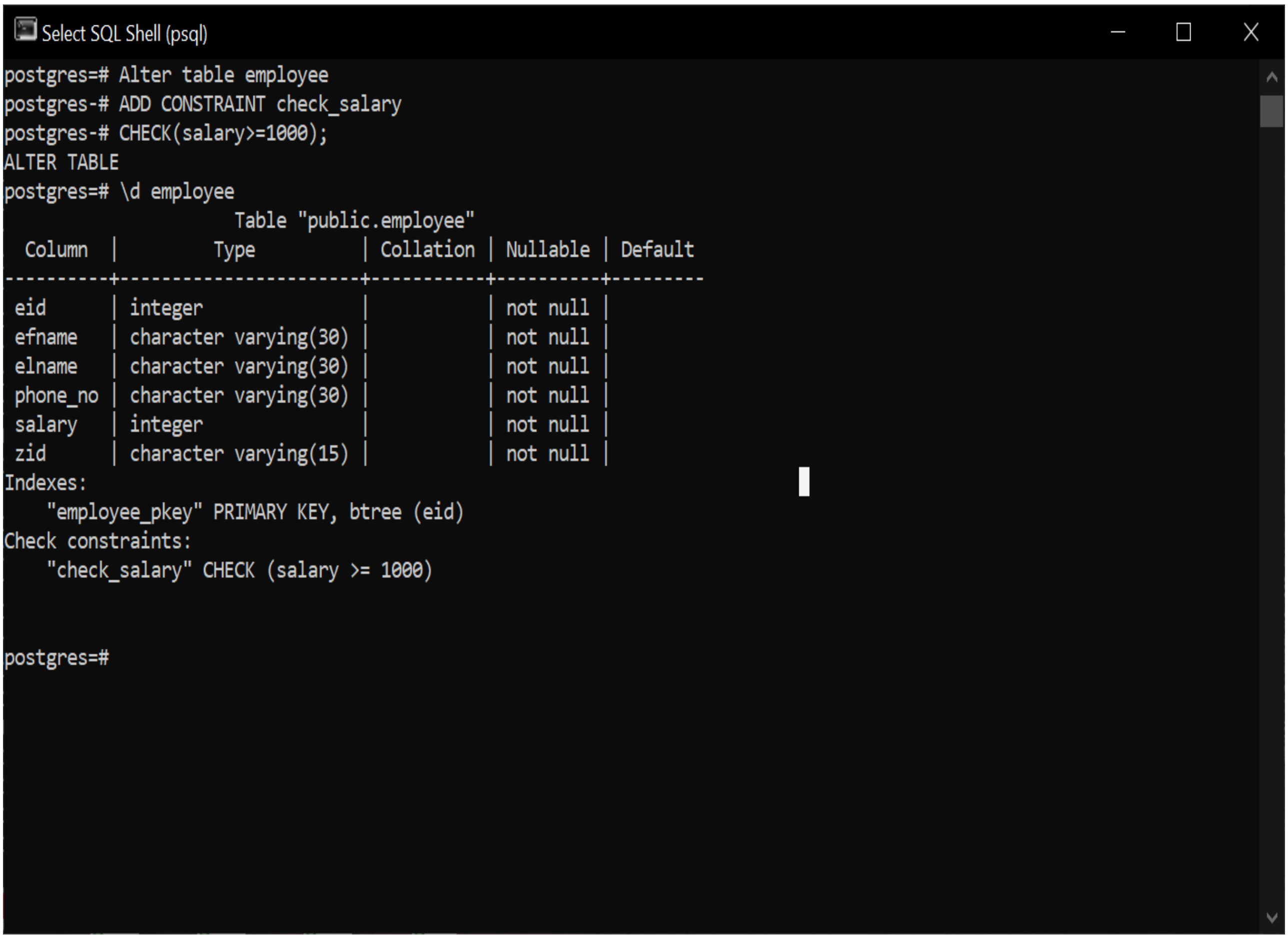


```
2pm-7pm Pune
```

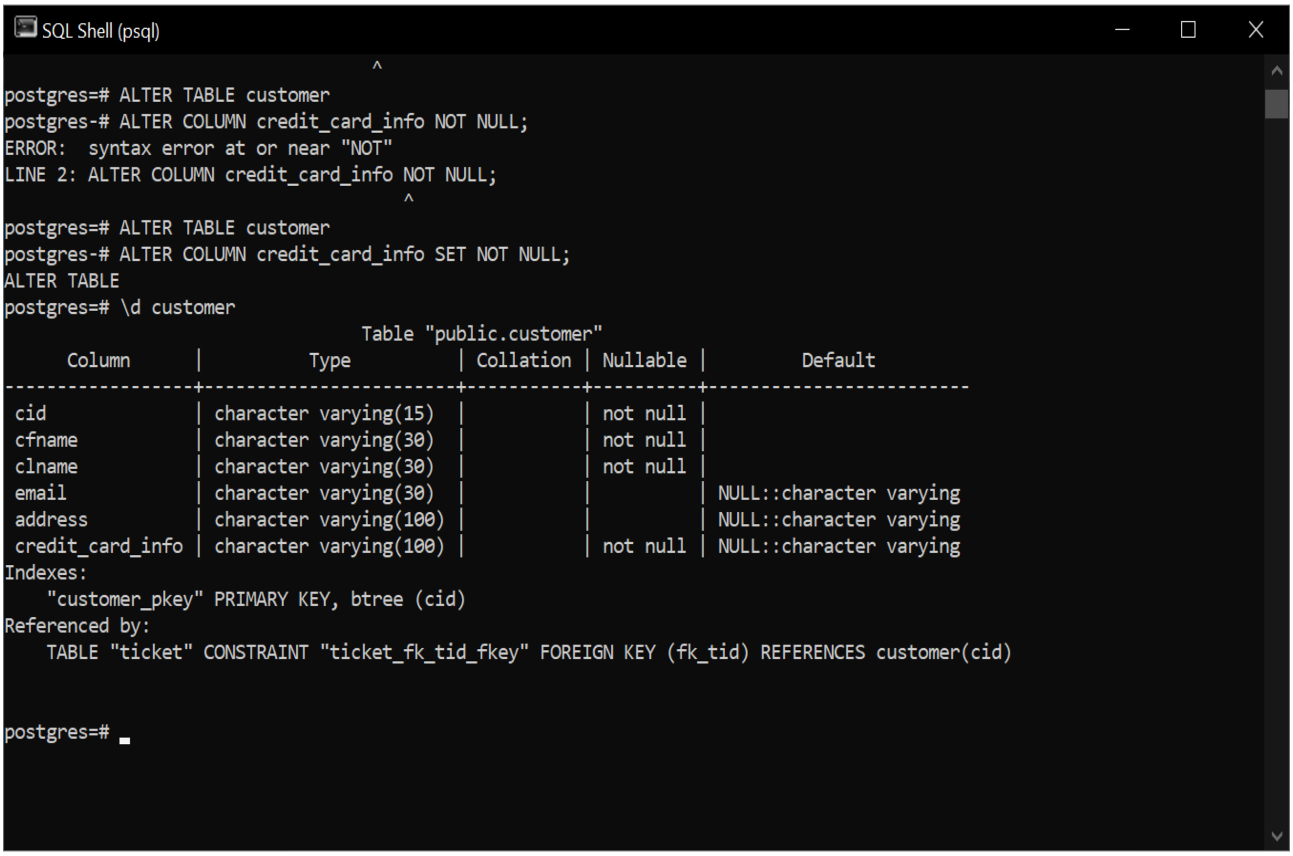


```
100 adarsh parabhat 987612344 1000 200
103 jacksx chaudhary 0712696969 2400 202
104 Ram
Sharma 123456789 5000 321
105 John K 37647836482 2300 230
106 Vishnu J 231761784617 2333 608
108 Jk Riya 8678785855 8888 234
101 John PK 987613344 2000 201
107 John SK 2345677754 4007 390
```

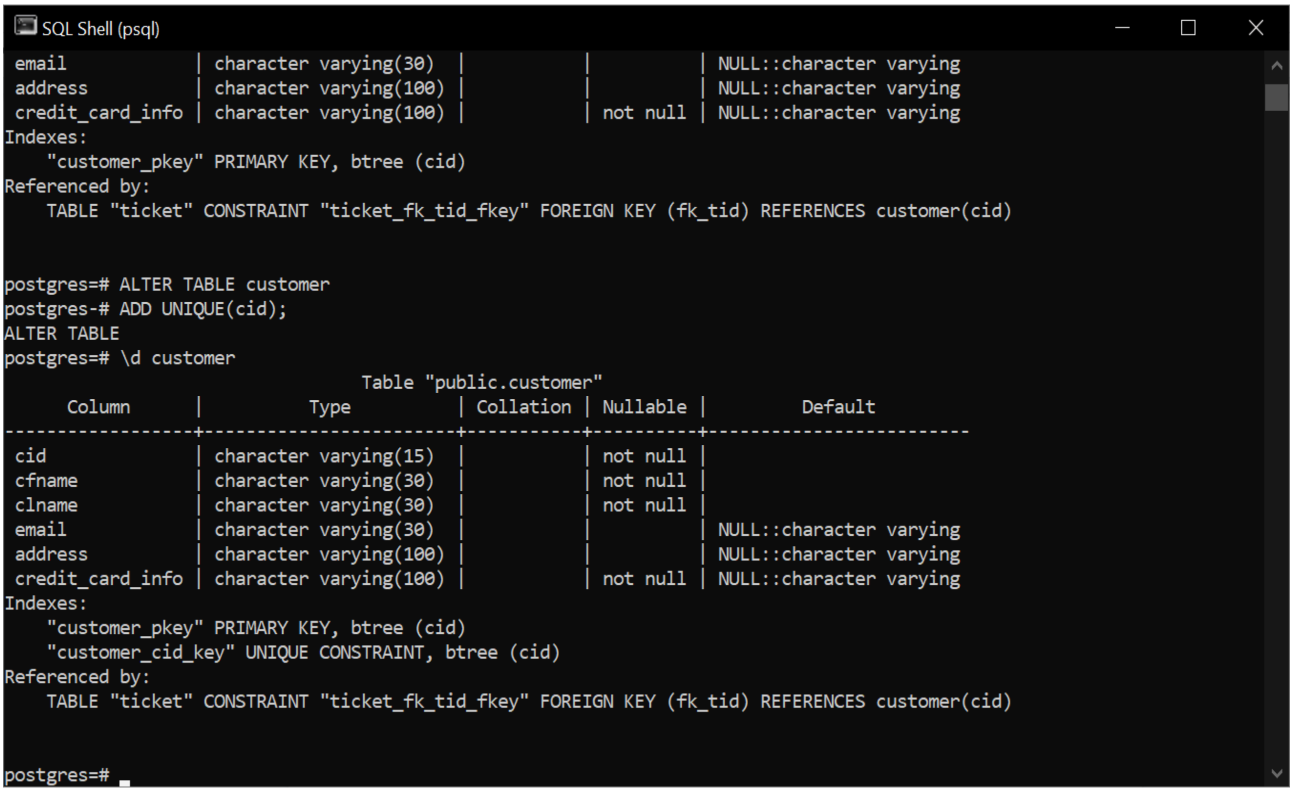
Additional queries:



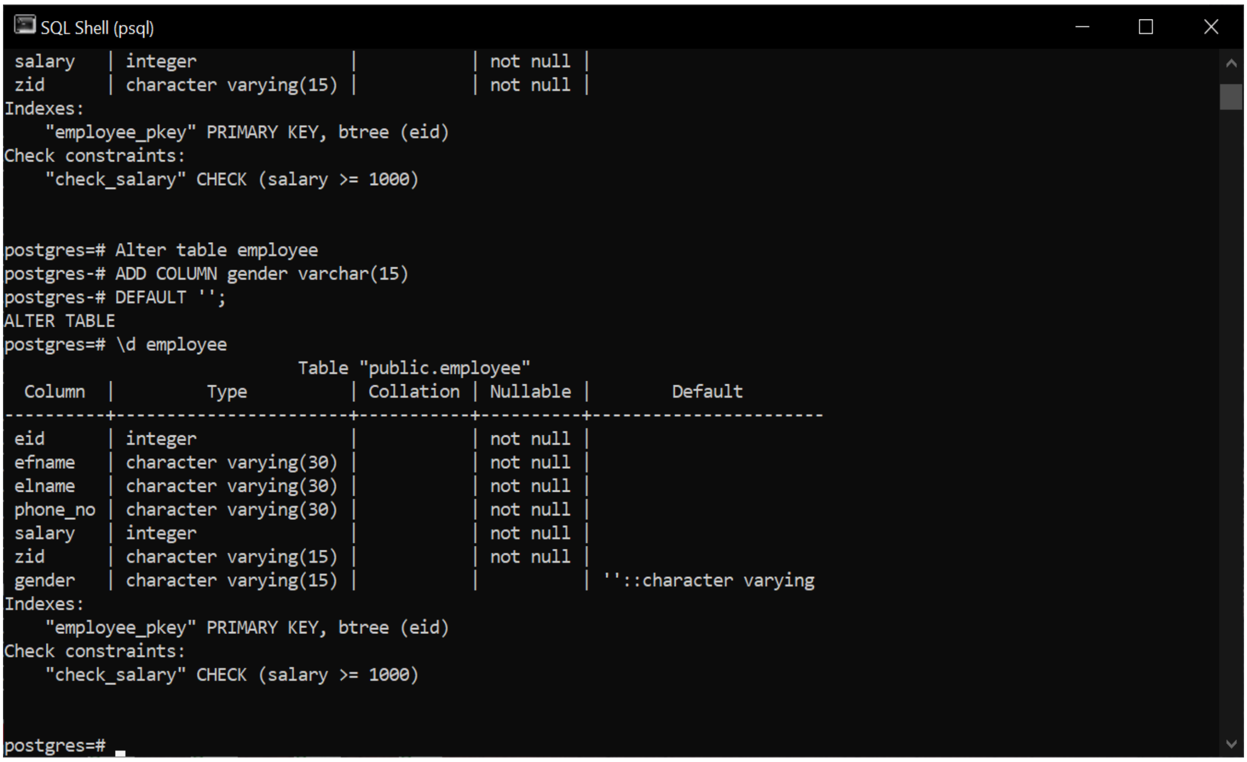
```
postgres=# Alter table employee
postgres-# ADD CONSTRAINT check_salary
postgres-# CHECK(salary>=1000);
ALTER TABLE
postgres=# \d employee
                      Table "public.employee"
  Column  |         Type          | Collation | Nullable | Default
----------+-----------------------+-----------+----------+---------
 eid      | integer               |           | not null |
 efname   | character varying(30) |           | not null |
 elname   | character varying(30) |           | not null |
 phone_no | character varying(30) |           | not null |
 salary   | integer               |           | not null |
 zid      | character varying(15) |           | not null |
Indexes:
    "employee_pkey" PRIMARY KEY, btree (eid)
Check constraints:
    "check_salary" CHECK (salary >= 1000)


postgres=#
```



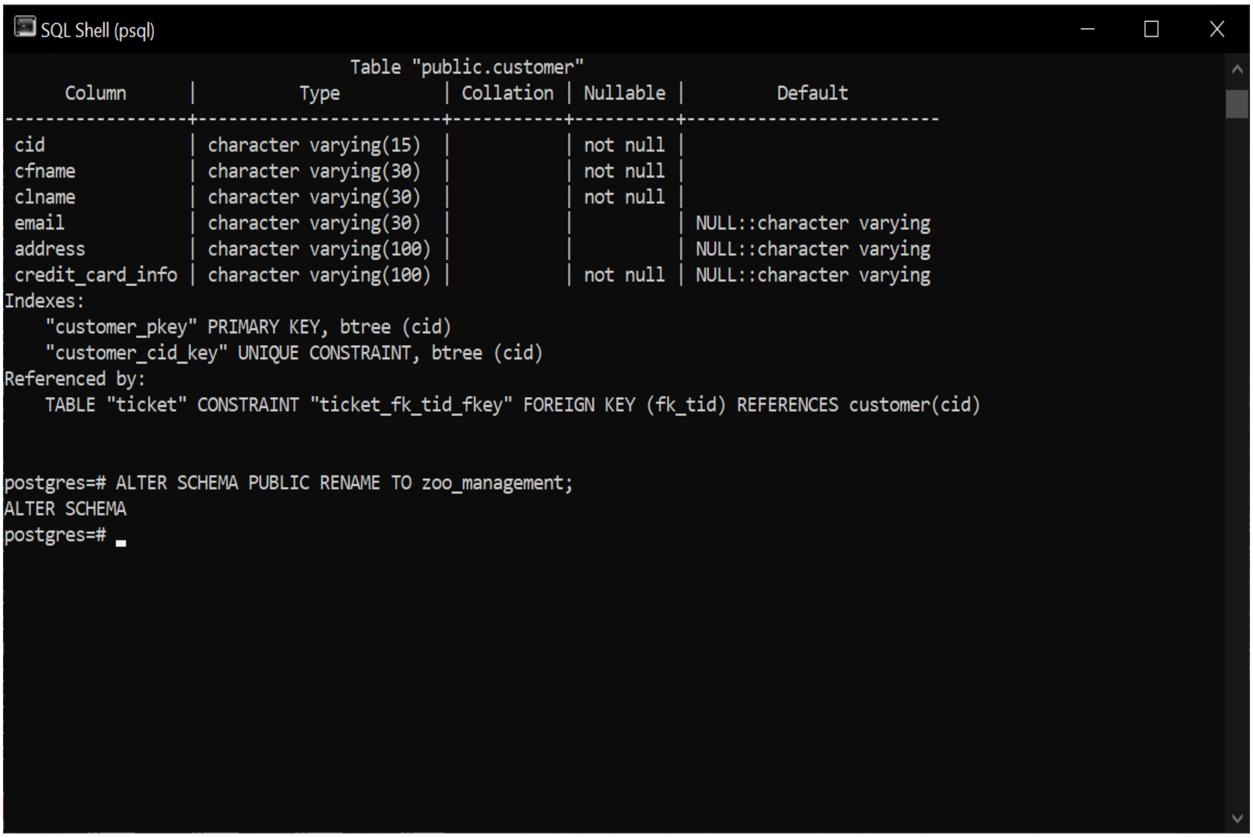
```
                                   ^
postgres=# ALTER TABLE customer
postgres-# ALTER COLUMN credit_card_info NOT NULL;
ERROR:  syntax error at or near "NOT"
LINE 2: ALTER COLUMN credit_card_info NOT NULL;
                                      ^
postgres=# ALTER TABLE customer
postgres-# ALTER COLUMN credit_card_info SET NOT NULL;
ALTER TABLE
postgres=# \d customer
                                  Table "public.customer"
      Column      |          Type          | Collation | Nullable |         Default
------------------+------------------------+-----------+----------+------------------------
 cid              | character varying(15)  |           | not null |
 cfname           | character varying(30)  |           | not null |
 clname           | character varying(30)  |           | not null |
 email            | character varying(30)  |           |          | NULL::character varying
 address          | character varying(100) |           |          | NULL::character varying
 credit_card_info | character varying(100) |           | not null | NULL::character varying
Indexes:
    "customer_pkey" PRIMARY KEY, btree (cid)
Referenced by:
    TABLE "ticket" CONSTRAINT "ticket_fk_tid_fkey" FOREIGN KEY (fk_tid) REFERENCES customer(cid)


postgres=#
```



```
 email            | character varying(30)  |           |          | NULL::character varying
 address          | character varying(100) |           |          | NULL::character varying
 credit_card_info | character varying(100) |           | not null | NULL::character varying
Indexes:
    "customer_pkey" PRIMARY KEY, btree (cid)
Referenced by:
    TABLE "ticket" CONSTRAINT "ticket_fk_tid_fkey" FOREIGN KEY (fk_tid) REFERENCES customer(cid)


postgres=# ALTER TABLE customer
postgres-# ADD UNIQUE(cid);
ALTER TABLE
postgres=# \d customer
                                  Table "public.customer"
      Column      |          Type          | Collation | Nullable |         Default
------------------+------------------------+-----------+----------+------------------------
 cid              | character varying(15)  |           | not null |
 cfname           | character varying(30)  |           | not null |
 clname           | character varying(30)  |           | not null |
 email            | character varying(30)  |           |          | NULL::character varying
 address          | character varying(100) |           |          | NULL::character varying
 credit_card_info | character varying(100) |           | not null | NULL::character varying
Indexes:
    "customer_pkey" PRIMARY KEY, btree (cid)
    "customer_cid_key" UNIQUE CONSTRAINT, btree (cid)
Referenced by:
    TABLE "ticket" CONSTRAINT "ticket_fk_tid_fkey" FOREIGN KEY (fk_tid) REFERENCES customer(cid)


postgres=#
```



```
 salary   | integer               |           | not null |
 zid      | character varying(15) |           | not null |
Indexes:
    "employee_pkey" PRIMARY KEY, btree (eid)
Check constraints:
    "check_salary" CHECK (salary >= 1000)


postgres=# Alter table employee
postgres-# ADD COLUMN gender varchar(15)
postgres-# DEFAULT '';
ALTER TABLE
postgres=# \d employee
                               Table "public.employee"
  Column  |         Type          | Collation | Nullable |        Default
----------+-----------------------+-----------+----------+-----------------------
 eid      | integer               |           | not null |
 efname   | character varying(30) |           | not null |
 elname   | character varying(30) |           | not null |
 phone_no | character varying(30) |           | not null |
 salary   | integer               |           | not null |
 zid      | character varying(15) |           | not null |
 gender   | character varying(15) |           |          | ''::character varying
Indexes:
    "employee_pkey" PRIMARY KEY, btree (eid)
Check constraints:
    "check_salary" CHECK (salary >= 1000)


postgres=#
```



```
                                  Table "public.customer"
      Column      |          Type          | Collation | Nullable |         Default
------------------+------------------------+-----------+----------+-------------------------
 cid              | character varying(15)  |           | not null |
 cfname           | character varying(30)  |           | not null |
 clname           | character varying(30)  |           | not null |
 email            | character varying(30)  |           |          | NULL::character varying
 address          | character varying(100) |           |          | NULL::character varying
 credit_card_info | character varying(100) |           | not null | NULL::character varying
Indexes:
    "customer_pkey" PRIMARY KEY, btree (cid)
    "customer_cid_key" UNIQUE CONSTRAINT, btree (cid)
Referenced by:
    TABLE "ticket" CONSTRAINT "ticket_fk_tid_fkey" FOREIGN KEY (fk_tid) REFERENCES customer(cid)


postgres=# ALTER SCHEMA PUBLIC RENAME TO zoo_management;
ALTER SCHEMA
postgres=#
```

3)Write up about the changes in Business/Application changes/expansion - that might lead to o schema changes o constraint changes o DBMS migration (from SQL based to No-SQL)

Schema changes:

A schema change is **an alteration made to a collection of logical structures (or schema objects) in a database**. Schema changes are generally made using structured query language (SQL) and are typically implemented during maintenance windows.

The schema is part of what helps turn data into useful information. When a schema is changed, **it creates a ripple through all the applications that depend on that schema**. With relational databases, a schema change can take weeks for developers to deal with while they adapt their code to the new model.

Constraint changes:

According to his theory, a business constraint is anything that **interferes with the profitability of a company or business endeavor**. Improving profitability requires the removal or reduction of business constraints. **Size of the market** – Businesses may operate in small or niche markets, meaning that the demand for their goods/services will be limited. As a result of this, there is little room for the business to expand and therefore there is less chance of them experiencing economies of scale.

4)With the existing design of your database, if you have to migrate to any No-SQL variety, then which one will be your choice? Why?

Postgresql is a RDBMS and in an RDBMS data is stored in tables(relations)

and it's preferable to define the schema on creation as later when we

alter tables we might need to alter the applications too and this might

cause errors.

MongoDB is scalable and flexible. It stores data in collections of Binary JSON documents and can manage structured and unstructured data allowing you to build your application without first defining the schema. We can add additional attributes to a record easily without altering the structure of the database.

CONTRIBUTIONS:

Manasa R: front-end tkinter, additional queries and execution

Namitha Nayak: front-end tkinter, queries and report 1 question

Nidhi Bharatiya: Front-end html, report and report compilation