DBMS Assignment-4

Team_ID: H18

Name: SRINJAY MAITRA	SRN: PES1UG19CS506
Name: Sreesha	SRN: PES1UG19CS503
Name: Suheb Papa	SRN: PES1UG19CS515

DBMS for RTO Database

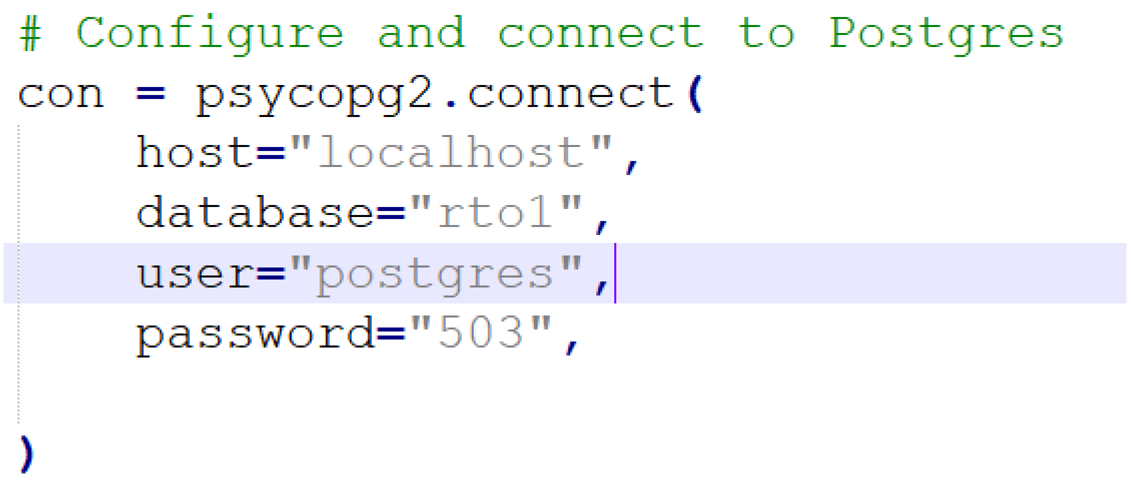
Task 1: Simple User interface design for front end

Language used: Python (Using Tkinter GUI library)

Tkinter is used for frontend user interface

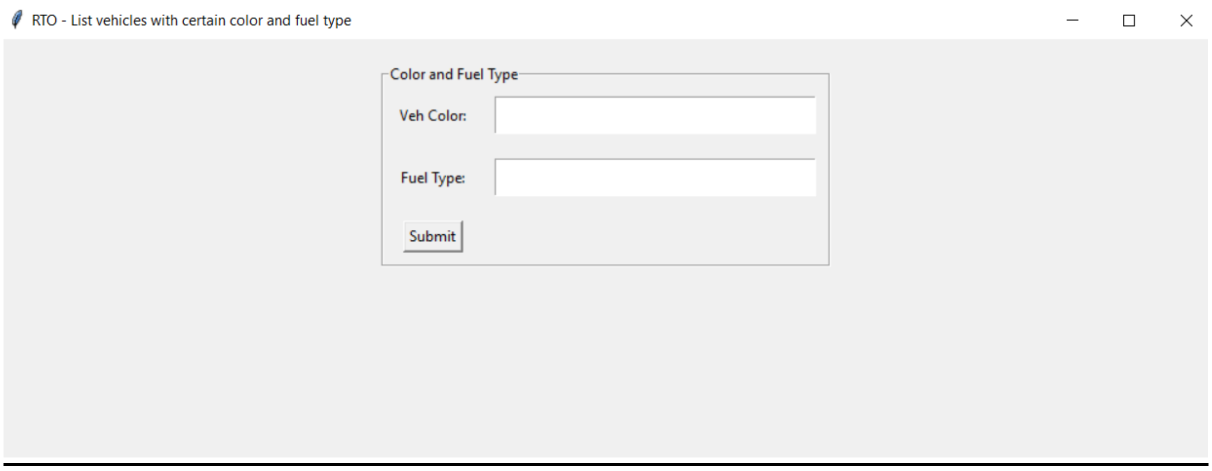
Database connectivity(front end to back end)

Psycopg2 is used. It is the most popular PostgreSQL database adapter for the Python programming language.



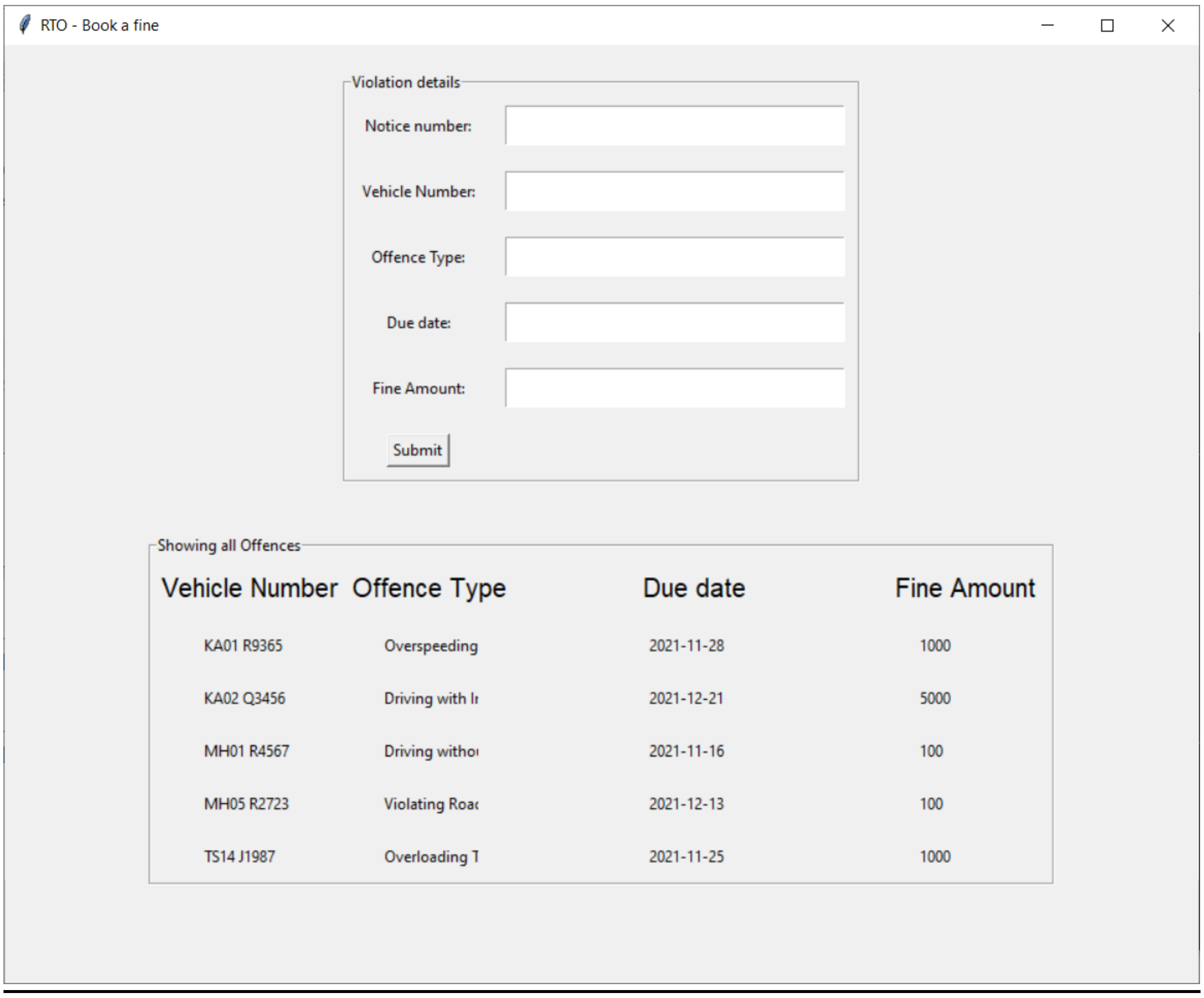
```python
# Configure and connect to Postgres
con = psycopg2.connect(
    host="localhost",
    database="rto1",
    user="postgres",
    password="503",

)
```


RTO - List vehicles with certain color and fuel type

Color and Fuel Type

Veh Color:	
Fuel Type:	


RTO - Book a fine

Violation details

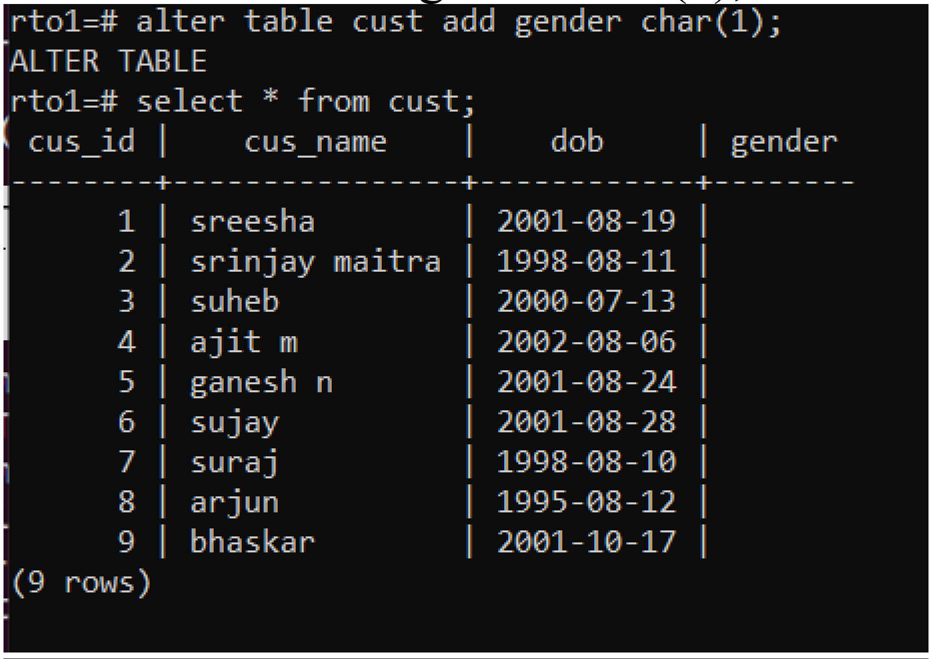
Notice number:	
Vehicle Number:	
Offence Type:	
Due date:	
Fine Amount:	

Showing all Offences

Vehicle Number	Offence Type	Due date	Fine Amount
KA01 R9365	Overspeeding	2021-11-28	1000
KA02 Q3456	Driving with Ir	2021-12-21	5000
MH01 R4567	Driving withou	2021-11-16	100
MH05 R2723	Violating Roac	2021-12-13	100
TS14 J1987	Overloading T	2021-11-25	1000

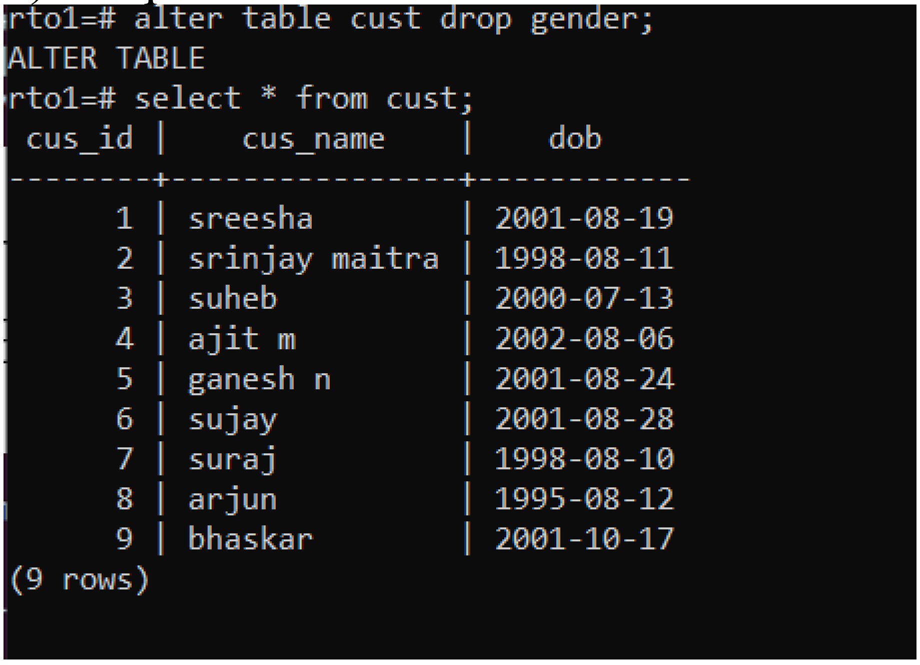
Task 2: Additional queries , Schema Changes

1) To Add Gender attribute to customer entity
alter table cust add gender char(1);



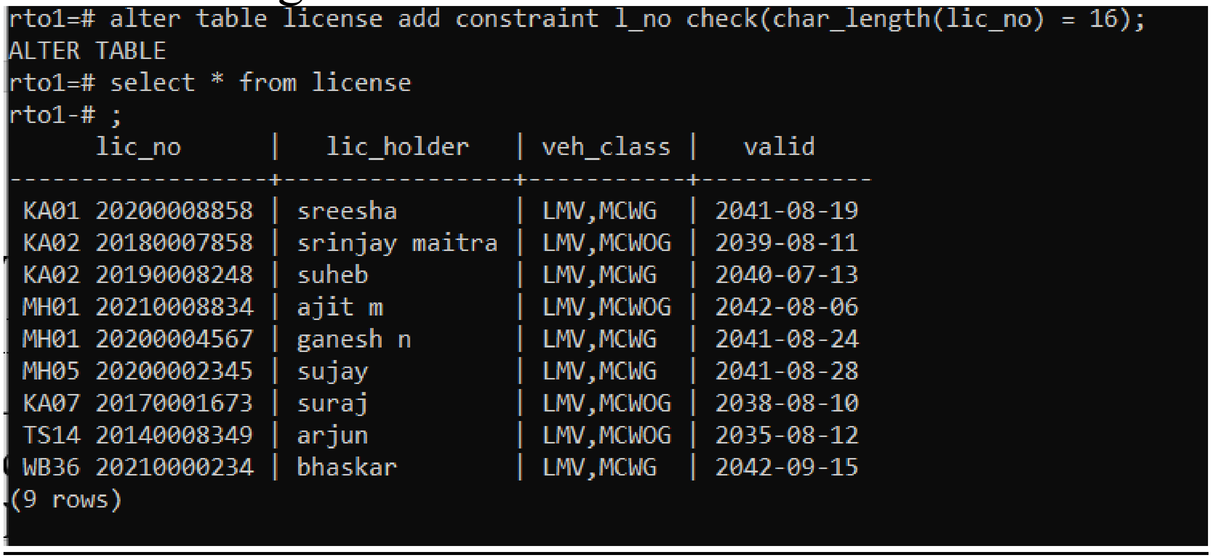
```
rto1=# alter table cust add gender char(1);
ALTER TABLE
rto1=# select * from cust;
 cus_id |    cus_name     |    dob     | gender
--------+----------------+------------+--------
      1 | sreesha        | 2001-08-19 |
      2 | srinjay maitra | 1998-08-11 |
      3 | suheb          | 2000-07-13 |
      4 | ajit m         | 2002-08-06 |
      5 | ganesh n       | 2001-08-24 |
      6 | sujay          | 2001-08-28 |
      7 | suraj          | 1998-08-10 |
      8 | arjun          | 1995-08-12 |
      9 | bhaskar        | 2001-10-17 |
(9 rows)
```

2) Drop Gender from Cust



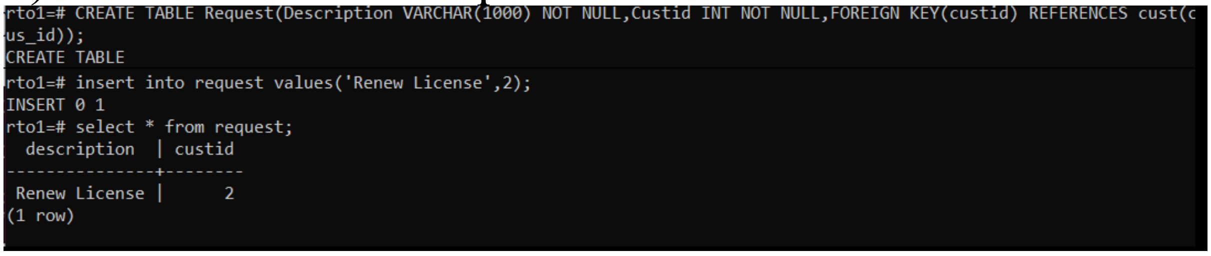
```
rto1=# alter table cust drop gender;
ALTER TABLE
rto1=# select * from cust;
 cus_id |    cus_name     |    dob
--------+----------------+------------
      1 | sreesha        | 2001-08-19
      2 | srinjay maitra | 1998-08-11
      3 | suheb          | 2000-07-13
      4 | ajit m         | 2002-08-06
      5 | ganesh n       | 2001-08-24
      6 | sujay          | 2001-08-28
      7 | suraj          | 1998-08-10
      8 | arjun          | 1995-08-12
      9 | bhaskar        | 2001-10-17
(9 rows)
```

3) Add constraint in license table to check lic_no character length



```
rto1=# alter table license add constraint l_no check(char_length(lic_no) = 16);
ALTER TABLE
rto1=# select * from license
rto1-# ;
      lic_no      |   lic_holder    | veh_class |   valid
------------------+-----------------+-----------+------------
 KA01 20200008858 | sreesha         | LMV,MCWG  | 2041-08-19
 KA02 20180007858 | srinjay maitra  | LMV,MCWOG | 2039-08-11
 KA02 20190008248 | suheb           | LMV,MCWG  | 2040-07-13
 MH01 20210008834 | ajit m          | LMV,MCWOG | 2042-08-06
 MH01 20200004567 | ganesh n        | LMV,MCWG  | 2041-08-24
 MH05 20200002345 | sujay           | LMV,MCWG  | 2041-08-28
 KA07 20170001673 | suraj           | LMV,MCWOG | 2038-08-10
 TS14 20140008349 | arjun           | LMV,MCWOG | 2035-08-12
 WB36 20210000234 | bhaskar         | LMV,MCWG  | 2042-09-15
(9 rows)
```

4) create new table request



```
rto1=# CREATE TABLE Request(Description VARCHAR(1000) NOT NULL,Custid INT NOT NULL,FOREIGN KEY(custid) REFERENCES cust(c
us_id));
CREATE TABLE
rto1=# insert into request values('Renew License',2);
INSERT 0 1
rto1=# select * from request;
  description  | custid
---------------+--------
 Renew License |      2
(1 row)
```

Data Migration

Due to storage of large amounts of data hence there is a need to for a storage system that can manage these data quickly and efficiently, query performance depends on data volume and transaction concurrency. Executing the same query on a table with millions of records requires more time that performing the same operation on the same table with only thousands of records. A lot of concurrent transactions can degrade SQL Server performance which can lead to CPU bottle necks and I/O bottlenecks, also RDBMS can be too restrictive, for unstructured or semi-structured data like images, text which comprise almost 90% of data a traditional RDBMS may not be suitable due to restrictive nature of it, that’s why organizations are migrating their databases from SQL to NoSQL because NoSQL provides good performance and scalability

Steps for migration from SQL to NoSQL

As the classified car database doesn't have complex data and mainly has operations like CRUD and a lot of unstructured data, a NoSQL database like a key-value model would be perfect as it would perfectly fit the operations, mongo DB can be used as it can be scaled quickly and schema changes can be done easily in case of Business changes or application changes.

MongoDB stores the data In form of JSON document hence the data should be converted to JSON.

>Prepare your application for connecting to MongoDB., MongoDB has support for all the major programming languages as well as many popular frameworks.

>Consider the schema changes that would be best for your data, while keeping in mind MongoDB schema best practices and avoiding anti-patterns.

>Export the data from your PostgreSQL databases by piping the result of an SQL query into a COPY command, outputting the result either as JSON or TSV.

>Restructure the data to fit your MongoDB schema by using mongo import.

Contribution:

Front-end design: Srinjay Maitra

Additional Queries and Database Connectivity: Sreesha I N

Database Migration Report Writeup: Suheb