

Overview

For this project, we created our own database using pgAdmin 4. With pgAdmin4 we are able to import our CSV file with the data using SQL queries. Once we confirm our database is complete and in BCNF form, we can use our Python to create a CLI interface using scripts so we can allow users to modify the database however they want.

pgAdmin 4

With pgAdmin 4 we can import any CSV data to your database. To create tables, you can choose to use Queries or use the interface provided by pgAdmin. Using the interface, you can create a table by clicking on the left side, if you click “Databases” it will open a table containing your database, click on your database and click on “schema” the schema will have a drop down menu one of them will be “table”. If you right click “table” it will give you an option to create your table with the column and the type of each column.

To create a table using SQL queries, you can create a table using SQL commands.

An example is: `CREATE TABLE tutorials (id int, tutorial_name text);`

To Import you CSV file to your table, you can use the command:

```
COPY table_name(Columns_in_table)
FROM 'Path to File'
DELIMITER ','
CSV HEADER;
```

In the projects case, an example is:

```
COPY drivers
FROM 'C:\Users\Public\Downloads\archive\drivers.csv'
DELIMITER ','
CSV HEADER;
```

If the data you imported and the data type you set up are incompatible, you can go to the properties of the table to change the type or clear out the contents of the data with this command:

```
TRUNCATE TABLE Table_name  
RESTART IDENTITY;
```

In the projects case, an example is:

```
TRUNCATE TABLE drivers  
RESTART IDENTITY;
```

Python Script:

The main script prompts the user to enter a value corresponding to the SQL command they want to do and then calls the imported functions to execute the scripts within those functions.

insert.py:

This file contains many functions that ask for the users to enter the data to the corresponding tables since when you insert into a table you would need to enter all information about that data. It prompts the user to enter what table they would like to access to insert data into with `choice()` and then depending on the input the corresponding table would ask the user to input the data to insert into the table with the SQL query `INSERT`.

delete.py:

Similar to `Insert.py` we prompt the user to enter what table they would like to access to with `choice()` and then depending on the input the corresponding table would ask the user to input the data of the column name and the value to match so we can use it to execute the SQL query `DELETE` to delete data.

update.py:

We ask the user to enter what table they would like to access to with choice() and then depending on the input the corresponding table would ask the user to input the data of the column name and the value to match so we can use the information to execute the SQL query UPDATE the data for the table.

select 1.py:

We ask the user to enter what table they would like to access to with choice() and then depending on the input the corresponding table would ask the user to input the data of the column name and the value to match so we can use the information to execute the SQL query SELECT the data for the table. This allows the user to search for data have the CLI display the information.

aggregate.py:

This script is a little different, in addition to the choice function, we also have the aggregate function aggregate(). The aggregate function lets the user choose what they want to do with data: add, find average, count, find the minimum and find the maximum. We first tell the user to make a choice on what table they want to modify and then ask the user how they modify the data. Depending on how the user wants to aggregate data, we would perform the corresponding SQL queries for sum(), avg(), count(), min() and max().

sort.py

This script has a special function as well, the user has a choice function to choose what function along with a choice if they want to sort sqlsort() the data by ascending or descending. Depending on what the user chooses to sort by we run SQL query command corresponding to what the user wanted with the SELECT * with desc/asc command.

join.py

This script doesn't have a choice function, instead we get it immediately from the user's input. We select the table name using SQL queries SELECT and join it with a second table we get from the user input with INNER JOIN with the matching columns.

group.py

This script uses the choice function to allow the users to choose a table. Then allow the users to type in the two columns they want to group the columns within the table. We can group it using the SQL query GROUP BY.

sub.py

This script is used for subqueries, we give the users a choice to input what table they wanted to subquery first. Once the table is chosen, we ask the user to enter the column name from the table and another column table from the second table. We use the SQL query command SELECT *from IN (SELECT * from)

transaction.py

This script doesn't have a choice function, instead we get it immediately from the user's input. We allow the user to select what they want to do for a transaction() they have the option to update or rollback. If the user selects update we select the table name with user input using SQL queries UPDATE the data with a new value for each transaction. If we want to rollback we give them the option to rollback with connection.rollback()

How to use the CLI:

On start up the CLI will prompt you to choose on which command you with a number corresponding to the SQL command, enter the number that matches the command

```
Please Select an Option:  
1.Insert Data  
2.Delete Data  
3.Update Data  
4.Search Data  
5.Aggregate Functions  
6.Sorting  
7.Joins  
8.Grouping  
9.Subqueries  
10.Transactions  
11.Exit  
Enter your choice (1-11): 
```

Everyone of these options except join and transaction will ask you to choose a table you would like to work with

After each operation if there is an error, it will print the error to the user, otherwise it will print that the operation is successful.

Insert:

If you choose Insert as the option, it will prompt you to choose a table you would like to insert to:

```
Enter your choice (1-11): 1
Choose a table: circuits(1), constructor_results(2), constructor_standings(3), constructors(4), driver_standings(5), drivers(6), lap_times(7), pit_stops(8), q
ualifying(9), races(10), results(11), seasons(12), sprint_results(13), status(14):
```

```
Enter your choice (1-11): 1
Choose a table: circuits(1), constructor_results(2), constructor_standings(3), constructors(4), driver_standings(5), drivers(6), lap_times(7), pit_stops(8), q
ualifying(9), races(10), results(11), seasons(12), sprint_results(13), status(14):
```

When you enter your choice, it will require you to enter the information you will like to insert into the table

```
circuitId: 102
circuitRef: test1
name: test2
location: test3
country: test4
1 Record inserted successfully into table
PostgreSQL connection is closed
PS C:\Users\dxie1>
```

The script will insert the data into the table for you but if there is an error, it will print the error for you.

```
Enter your choice (1-11): 1
Choose a table: circuits(1), constructor_results(2), constructor_standings(3), constructors(4), driver_standings(5), drivers(6), lap_times(7), pit_stops(8), q
ualifying(9), races(10), results(11), seasons(12), sprint_results(13), status(14): 1
circuitId: incorrect
circuitRef: test
name: test
location: test
country: test
Failed to insert record into table: invalid input syntax for type integer: "incorrect"
LINE 1: ...rcuitRef", "name", "location", "country") VALUES ('incorrect...
                                     ^
```

Delete:

Delete will again ask the user for the table they would like to access and then ask them for a more specific column the value in the column to match in order to perform the SQL command.

```
3.Update Data
4.Search Data
5.Aggregate Functions
6.Sorting
7.Joins
8.Grouping
9.Subqueries
10.Transactions
11.Exit
Enter your choice (1-11): 2
Choose a table: circuits(1), constructor_results(2), constructor_standings(3), constructors(4), driver_standings(5), drivers(6), lap_times(7), pit_stops(8), qualifying(9), races(10), results(11), seasons(12), sprint_results(13), status(14): 1
Enter Column name: circuitId
Enter a id: 102
Record deleted successfully
PostgreSQL connection is closed
PS C:\Users\dxiei>
```

Update:

After getting the user for the table they would like to modify, the script will then prompt you to input the specific column and row the user would like to update.

```
Enter your choice (1-11): 3
Choose a table: circuits(1), constructor_results(2), constructor_standings(3), constructors(4), driver_standings(5), drivers(6), lap_times(7), pit_stops(8), qualifying(9), races(10), results(11), seasons(12), sprint_results(13), status(14): 1
tableid: circuitId
Enter ID number: 101
Enter Table name: circuitRef
New Table value: u
1 Record Updated successfully
PostgreSQL connection is closed
```

Search:

After getting the user for the table they would like to modify, the script will ask you which column and a value from the row where it can use to search the table, so it can display the row you searched for.

```
Enter your choice (1-11): 4
Choose a table: circuits(1), constructor_results(2), constructor_standings(3), constructors(4), driver_standings(5), drivers(6), lap_times(7), pit_stops(8), qualifying(9), races(10), results(11), seasons(12), sprint_results(13), status(14): 1
Enter Column name: circuitId
Enter a value: 101
(101, 'u', 'test2', 'test3', 'test4')
PostgreSQL connection is closed
PS C:\Users\dxiei\Downloads\part 3>
```

Aggregate:

Once you choose a table to modify from the options, you will be presented with a choice to get the sum, average, count, minimum and maximum. When you decide on an option you will be prompted to enter information to find the column you would like to get the information from.

- Sum

```
Please Select an Option:
1.Insert Data
2.Delete Data
3.Update Data
4.Search Data
5.Aggregate Functions
6.Sorting
7.Joins
8.Grouping
9.Subqueries
10.Transactions
11.Exit
Enter your choice (1-11): 5
Choose a table: circuits(1), constructor_results(2), constructor_standings(3), constructors(4), driver_standings(5), seasons(12), sprint_results(13), status(14): 1
What would you like to do?: sum(1), average(2), count(3), min(4), max(5): 1
Enter Column name: circuitId
(3172,)
```

- Average

```
11.Exit
Enter your choice (1-11): 5
Choose a table: circuits(1), constructor_results(2), constructor_standings(3), driver_standings(5), qualifying(9), races(10), results(11), seasons(12), sprint_results(13), status(14): 1
What would you like to do?: sum(1), average(2), count(3), min(4), max(5): 2
Enter Column name: circuitId
(Decimal('40.666666666666667'),)
PostgreSQL connection is closed
```

- Count

```
Choose a table: circuits(1), constructor_results(2), constructor_standings(3), driver_standings(5), qualifying(9), races(10), results(11), seasons(12), sprint_results(13), status(14): 1
What would you like to do?: sum(1), average(2), count(3), min(4), max(5): 3
Enter Column name: circuitId
(78,)
PostgreSQL connection is closed
```


- Minimum

```
Please Select an Option:
1.Insert Data
2.Delete Data
3.Update Data
4.Search Data
5.Aggregate Functions
6.Sorting
7.Joins
8.Grouping
9.Subqueries
10.Transactions
11.Exit
Enter your choice (1-11): 5
Choose a table: circuits(1), constructor_results(2), constructor_standings(3), constructors
), seasons(12), sprint_results(13), status(14): 1
What would you like to do?: sum(1), average(2), count(3), min(4), max(5): 4
Enter Column name: circuitId
(1,)
PostgreSQL connection is closed
PS C:\Users\dxie1>
```

- Maximum

```
Please Select an Option:
1.Insert Data
2.Delete Data
3.Update Data
4.Search Data
5.Aggregate Functions
6.Sorting
7.Joins
8.Grouping
9.Subqueries
10.Transactions
11.Exit
Enter your choice (1-11): 5
Choose a table: circuits(1), constructor_results(2), constructor_standings(3), c
), seasons(12), sprint_results(13), status(14): 1
What would you like to do?: sum(1), average(2), count(3), min(4), max(5): 5
Enter Column name: circuitId
(101,)
PostgreSQL connection is closed
PS C:\Users\dxie1>
```

Sorting:

Like the aggregate script, after you choose a table to modify from the options, you will be presented with a choice to sort ascending or descending, when you decide on an option you will be prompted to enter information to find the column you would like to get the information from.

- Ascending

```
Enter your choice (1-11): 6
Choose a table: circuits(1), constructor_results(2), constructor_standings(3), constructors(4), driver_standings(5), drivers(6), lap_times(7), pit_stops(8), qualifying(9), races(10), results(11), seasons(12), sprint_results(13), status(14): 1
How would you like to sort?: ascent(1), descent(2): 1
Enter Column name: circuitid
[[1, 'albert park', 'Albert Park Grand Prix Circuit', 'Melbourne', 'Australia'), (2, 'sepang', 'Sepang International Circuit', 'Kuala Lumpur', 'Malaysia'), (3, 'bahrain', 'Bahrain International Circuit', 'Sakhir', 'Bahrain'), (4, 'catalunya', 'Circuit de Barcelona-Catalunya', 'Montmel', 'Spain'), (5, 'istanbul', 'Istanbul Park', 'Istanbul', 'Turkey'), (6, 'monaco', 'Circuit de Monaco', 'Monte-Carlo', 'Monaco'), (7, 'villeneuve', 'Circuit Gilles Villeneuve', 'Montreal', 'Canada'), (8, 'magny cours', 'Circuit de Nevers Magny-Cours', 'Magny Cours', 'France'), (9, 'silverstone', 'Silverstone Circuit', 'Silverstone', 'UK'), (10, 'hockenheimring', 'Hockenheimring', 'Hockenheim', 'Germany'), (11, 'hungaroring', 'Hungaroring', 'Budapest', 'Hungary'), (12, 'valencia', 'Valencia Street Circuit', 'Valencia', 'Spain'), (13, 'spa', 'Circuit de Spa-Francorchamps', 'Spa', 'Belgium'), (14, 'monza', 'Autodromo Nazionale di Monza', 'Monza', 'Italy'), (15, 'marina bay street circuit', 'Marina Bay Street Circuit', 'Marina Bay', 'Singapore'), (16, 'fuji', 'Fuji Speedway', 'Oyama', 'Japan'), (17, 'shanghai', 'Shanghai International Circuit', 'Shanghai', 'China'), (18, 'interlagos', 'Autodromo Jose Carlos Pace', 'Sao Paulo', 'Brazil'), (19, 'indianapolis', 'Indianapolis Motor Speedway', 'Indianapolis', 'USA'), (20, 'nurburgring', 'Nurburgring', 'Nurburg', 'Germany'), (21, 'imola', 'Autodromo Enzo e Dino Ferrari', 'Imola', 'Italy'), (22, 'suzuka', 'Suzuka Circuit', 'Suzuka', 'Japan'), (24, 'yas marina', 'Yas Marina Circuit', 'Abu Dhabi', 'UAE'), (25, 'galvez', 'Autodromo Juan y Oscar Gálvez', 'Buenos Aires', 'Argentina'), (26, 'jerez', 'Circuito de Jerez', 'Jerez de la Frontera', 'Spain'), (27, 'estoril', 'AutAdromo do Estoril', 'Estoril', 'Portugal'), (28, 'okayama', 'Okayama International Circuit', 'Okayama', 'Japan'), (29, 'adelaide', 'Adelaide Street Circuit', 'Adelaide', 'Australia'), (30, 'kyalami', 'Kyalami', 'Midrand', 'South Africa'), (31, 'donington', 'Donington Park', 'Castle Donington', 'UK'), (32, 'rodriguez', 'Autodromo Hermanos Rodriguez', 'Mexico City', 'Mexico'), (33, 'phoenix', 'Phoenix street circuit', 'Phoenix', 'USA'), (34, 'ricard', 'Circuit Paul Ricard', 'Le Castellet', 'France'), (35, 'yeongam', 'Korean International Circuit', 'Yeongam County', 'Korea'), (36, 'jacarepagua', 'AutAdromo Internacional Nelson Piquet', 'Rio de Janeiro', 'Brazil'), (37, 'detroit', 'Detroit Street Circuit', 'Detroit', 'USA'), (38, 'brands hatch', 'Brands Hatch', 'Kent', 'UK'), (39, 'zandvoort', 'Circuit Park Zandvoort', 'Zandvoort', 'Netherlands'), (40, 'zolder', 'Zolder', 'Heusden-Zolder', 'Belgium'), (41, 'dijon', 'Dijon-Prenois', 'Dijon', 'France'), (42, 'dallas', 'Fair Park', 'Dallas', 'USA'), (43, 'long beach', 'Long Beach', 'California', 'USA'), (44, 'las vegas', 'Las Vegas Street Circuit', 'Nevada', 'USA'), (45, 'jarama', 'Jarama', 'Madrid', 'Spain'), (46, 'watkins glen', 'Watkins Glen', 'New York State', 'USA'), (47, 'anderstorp', 'Scandinavian Raceway', 'Anderstorp', 'Sweden'), (48, 'mosport', 'Mosport International Raceway', 'Ontario', 'Canada'), (49, 'montjuic', 'Montjuic', 'Barcelona', 'Spain'), (50, 'niveles', 'Niveles-Baulers', 'Brussels', 'Belgium'), (51, 'charade', 'Charade Circuit', 'Clermont-Ferrand', 'France'), (52, 'tremblant', 'Circuit Mont-Tremblant', 'Quebec', 'Canada'), (53, 'essarts', 'Rouen-Les-Essarts', 'Rouen', 'France'), (54, 'lemans', 'Le Mans', 'Le Mans', 'France'), (55, 'reims', 'Reims-Gueux', 'Reims', 'France'), (56, 'george', 'Prince George Circuit', 'Eastern Cape Province', 'South Africa'), (57, 'zeltweg', 'Zeltweg', 'Styria', 'Austria'), (58, 'aintree', 'Aintree', 'Liverpool', 'UK'), (59, 'boavista', 'Circuito da Boavista', 'Oporto', 'Portugal'), (60, 'riverside', 'Riverside International Raceway', 'California', 'USA'), (61, 'avus', 'AVUS', 'Berlin', 'Germany'), (62, 'monsanto', 'Monsanto Park Circuit', 'Lisbon', 'Portugal'), (63, 'sebring', 'Sebring International Raceway', 'Florida', 'USA'), (64, 'ain-diab', 'Ain Diab', 'Casablanca', 'Morocco'), (65, 'pescara', 'Pescara Circuit', 'Pescara', 'Italy'), (66, 'bremgarten', 'Circuit Bremgarten', 'Bern', 'Switzerland'), (67, 'pedralbes', 'Circuit de Pedralbes', 'Barcelona', 'Spain'), (68, 'buddh', 'Buddh International Circuit', 'Uttar Pradesh', 'India'), (69, 'americas', 'Circuit of the Americas', 'Austin', 'USA'), (70, 'red bull ring', 'Red Bull Ring', 'Spielberg', 'Austria'), (71, 'sochi', 'Sochi Autodrom', 'Sochi', 'Russia'), (72, 'mugello', 'Mugello', 'Italy'), (73, 'jeddah', 'Jeddah Corniche Circuit', 'Jeddah', 'Saudi Arabia'), (74, 'losail', 'Losail International Circuit', 'Al Daayen', 'Qatar'), (75, 'miami', 'Miami International Autodrome', 'Miami', 'USA'), (76, 'vegas', 'Las Vegas Strip Street Circuit', 'Las Vegas', 'United States'), (77, 'test1', 'test2', 'test3', 'test4')]]
PostgreSQL connection is closed
```

- Descending

```
Enter your choice (1-11): 6
Choose a table: circuits(1), constructor_results(2), constructor_standings(3), constructors(4), driver_standings(5), drivers(6), lap_times(7), pit_stops(8), qualifying(9), races(10), results(11), seasons(12), sprint_results(13), status(14): 1
How would you like to sort?: ascent(1), descent(2): 2
Enter Column name: circuitid
[[101, 'test1', 'test2', 'test3', 'test4'), (80, 'vegas', 'Las Vegas Strip Street Circuit', 'Las Vegas', 'United States'), (79, 'miami', 'Miami International Autodrome', 'Miami', 'USA'), (78, 'losail', 'Losail International Circuit', 'Al Daayen', 'Qatar'), (77, 'jeddah', 'Jeddah Corniche Circuit', 'Jeddah', 'Saudi Arabia'), (76, 'mugello', 'Autodromo Internazionale del Mugello', 'Mugello', 'Italy'), (75, 'portimao', 'AutAdromo Internacional do Algarve', 'Portimao', 'Portugal'), (73, 'baku', 'Baku City Circuit', 'Baku', 'Azerbaijan'), (71, 'sochi', 'Sochi Autodrom', 'Sochi', 'Russia'), (70, 'red bull ring', 'Red Bull Ring', 'Spielberg', 'Austria'), (69, 'americas', 'Circuit of the Americas', 'Austin', 'USA'), (68, 'buddh', 'Buddh International Circuit', 'Uttar Pradesh', 'India'), (67, 'pedralbes', 'Circuit de Pedralbes', 'Barcelona', 'Spain'), (66, 'bremgarten', 'Circuit Bremgarten', 'Bern', 'Switzerland'), (65, 'pescara', 'Pescara Circuit', 'Pescara', 'Italy'), (64, 'ain-diab', 'Ain Diab', 'Casablanca', 'Morocco'), (63, 'sebring', 'Sebring International Raceway', 'Florida', 'USA'), (62, 'monsanto', 'Monsanto Park Circuit', 'Lisbon', 'Portugal'), (61, 'avus', 'AVUS', 'Berlin', 'Germany'), (60, 'riverside', 'Riverside International Raceway', 'California', 'USA'), (59, 'boavista', 'Circuito da Boavista', 'Oporto', 'Portugal'), (58, 'aintree', 'Aintree', 'Liverpool', 'UK'), (57, 'zeltweg', 'Zeltweg', 'Styria', 'Austria'), (56, 'george', 'Prince George Circuit', 'Eastern Cape Province', 'South Africa'), (55, 'reims', 'Reims-Gueux', 'Reims', 'France'), (54, 'lemans', 'Le Mans', 'Le Mans', 'France'), (53, 'essarts', 'Rouen-Les-Essarts', 'Rouen', 'France'), (52, 'tremblant', 'Circuit Mont-Tremblant', 'Quebec', 'Canada'), (51, 'charade', 'Charade Circuit', 'Clermont-Ferrand', 'France'), (50, 'niveles', 'Niveles-Baulers', 'Brussels', 'Belgium'), (49, 'montjuic', 'Montjuic', 'Barcelona', 'Spain'), (48, 'mosport', 'Mosport International Raceway', 'Ontario', 'Canada'), (47, 'anderstorp', 'Scandinavian Raceway', 'Anderstorp', 'Sweden'), (46, 'watkins glen', 'Watkins Glen', 'New York State', 'USA'), (45, 'jarama', 'Jarama', 'Madrid', 'Spain'), (44, 'las vegas', 'Las Vegas Street Circuit', 'Nevada', 'USA'), (43, 'long beach', 'Long Beach', 'California', 'USA'), (42, 'dallas', 'Fair Park', 'Dallas', 'USA'), (41, 'dijon', 'Dijon-Prenois', 'Dijon', 'France'), (40, 'zolder', 'Zolder', 'Heusden-Zolder', 'Belgium'), (39, 'zandvoort', 'Circuit Park Zandvoort', 'Zandvoort', 'Netherlands'), (38, 'brands hatch', 'Brands Hatch', 'Kent', 'UK'), (37, 'detroit', 'Detroit Street Circuit', 'Detroit', 'USA'), (36, 'jacarepagua', 'AutAdromo Internacional Nelson Piquet', 'Rio de Janeiro', 'Brazil'), (35, 'yeongam', 'Korean International Circuit', 'Yeongam County', 'Korea'), (34, 'ricard', 'Circuit Paul Ricard', 'Le Castellet', 'France'), (33, 'phoenix', 'Phoenix street circuit', 'Phoenix', 'USA'), (32, 'rodriguez', 'Autodromo Hermanos Rodriguez', 'Mexico City', 'Mexico'), (31, 'donington', 'Donington Park', 'Castle Donington', 'UK'), (30, 'kyalami', 'Kyalami', 'Midrand', 'South Africa'), (29, 'adelaide', 'Adelaide Street Circuit', 'Adelaide', 'Australia'), (28, 'okayama', 'Okayama International Circuit', 'Okayama', 'Japan'), (27, 'estoril', 'AutAdromo do Estoril', 'Estoril', 'Portugal'), (26, 'jerez', 'Circuito de Jerez', 'Jerez de la Frontera', 'Spain'), (25, 'galvez', 'Autodromo Juan y Oscar Gálvez', 'Buenos Aires', 'Argentina'), (24, 'yas marina', 'Yas Marina Circuit', 'Abu Dhabi', 'UAE'), (22, 'suzuka', 'Suzuka Circuit', 'Suzuka', 'Japan'), (21, 'imola', 'Autodromo Enzo e Dino Ferrari', 'Imola', 'Italy'), (20, 'nurburgring', 'Nurburgring', 'Nurburg', 'Germany'), (19, 'indianapolis', 'Indianapolis Motor Speedway', 'Indianapolis', 'USA'), (18, 'interlagos', 'Autodromo Jose Carlos Pace', 'Sao Paulo', 'Brazil'), (17, 'shanghai', 'Shanghai International Circuit', 'Shanghai', 'China'), (16, 'fuji', 'Fuji Speedway', 'Oyama', 'Japan'), (15, 'marina bay street circuit', 'Marina Bay Street Circuit', 'Marina Bay', 'Singapore'), (14, 'monza', 'Autodromo Nazionale di Monza', 'Monza', 'Italy'), (13, 'spa', 'Circuit de Spa-Francorchamps', 'Spa', 'Belgium'), (12, 'valencia', 'Valencia Street Circuit', 'Valencia', 'Spain'), (11, 'hungaroring', 'Hungaroring', 'Budapest', 'Hungary'), (10, 'hockenheimring', 'Hockenheimring', 'Hockenheim', 'Germany'), (9, 'silverstone', 'Silverstone Circuit', 'Silverstone', 'UK'), (8, 'magny cours', 'Circuit de Nevers Magny-Cours', 'Magny Cours', 'France'), (7, 'villeneuve', 'Circuit Gilles Villeneuve', 'Montreal', 'Canada'), (6, 'monaco', 'Circuit de Monaco', 'Monte-Carlo', 'Monaco'), (5, 'istanbul', 'Istanbul Park', 'Istanbul', 'Turkey'), (4, 'catalunya', 'Circuit de Barcelona-Catalunya', 'Montmel', 'Spain'), (3, 'bahrain', 'Bahrain International Circuit', 'Sakhir', 'Bahrain'), (2, 'sepang', 'Sepang International Circuit', 'Kuala Lumpur', 'Malaysia'), (1, 'albert park', 'Albert Park Grand Prix Circuit', 'Melbourne', 'Australia')]]
PostgreSQL connection is closed
```

Join:

As explained above, this script prompts the user to enter the table instead of letting them choose, this allows more flexibility to join tables. Enter the required information and it will output the tables that have been join if there are no errors.

```

4.Search Data
5.Aggregate Functions
6.Sorting
7.Joins
8.Grouping
9.Subqueries
10.Transactions
11.Exit
Enter your choice (1-11): 7
Enter Table name: drivers
Enter Second Table name: constructors
Enter Column name: driverId
Enter Second Column name: constructorId
[[('1', 'hamilton', '44', 'HAM', 'Lewis', 'Hamilton', datetime.date(1985, 1, 7), 'British', 1, 'mclaren', 'McLaren', 'British'), (2, 'heidfeld', '\\N', 'HEI', 'Nick', 'Heidfeld', datetime.date(1977, 5, 10), 'German', 2, 'bmw_sauber', 'BMW Sauber', 'German'), (3, 'rosberg', '6', 'ROS', 'Nico', 'Rosberg', datetime.date(1985, 6, 27), 'German', 3, 'williams', 'Williams', 'British'), (4, 'alonso', '14', 'ALO', 'Fernando', 'Alonso', datetime.date(1981, 7, 29), 'Spanish', 4, 'renault', 'Renault', 'French'), (5, 'kovalainen', '\\N', 'KOV', 'Heikki', 'Kovalainen', datetime.date(1981, 10, 19), 'Finnish', 5, 'toro_rosso', 'Toro Rosso', 'Italian'), (6, 'nakajima', '\\N', 'NAK', 'Kazuki', 'Nakajima', datetime.date(1985, 1, 11), 'Japanese', 6, 'ferrari', 'Ferrari', 'Italian'), (7, 'bourdais', '\\N', 'BOU', 'Sebastien', 'Bourdais', datetime.date(1979, 2, 28), 'French', 7, 'toyota', 'Toyota', 'Japanese'), (8, 'raikkonen', '7', 'RAI', 'Kimi', 'Raikkonen', datetime.date(1979, 10, 17), 'Finnish', 8, 'super_aguri', 'Super Aguri', 'Japanese'), (9, 'kubica', '88', 'KUB', 'Robert', 'Kubica', datetime.date(1984, 12, 7), 'Polish', 9, 'red_bull', 'Red Bull', 'Austrian'), (10, 'glock', '\\N', 'GLO', 'Timo', 'Glock', datetime.date(1982, 3, 18), 'German', 10, 'force_india', 'Force India', 'Indian'), (11, 'sato', '\\N', 'SAT', 'Takuma', 'Sato', datetime.date(1977, 1, 28), 'Japanese', 11, 'honda', 'Honda', 'Japanese'), (12, 'piquet_jr', '\\N', 'PIQ', 'Nelson', 'Piquet Jr.', datetime.date(1985, 7, 25), 'Brazilian', 12, 'spyker', 'Spyker', 'Dutch'), (13, 'massa', '19', 'MAS', 'Felipe', 'Massa', datetime.date(1981, 4, 25), 'Brazilian', 13, 'mfl', 'MFL', 'Russian'), (14, 'coulthard', '\\N', 'COU', 'David', 'Coulthard', datetime.date(1971, 3, 27), 'British', 14, 'spyker_mfl', 'Spyker MFL', 'Dutch'), (15, 'trulli', '\\N', 'TRU', 'Jarno', 'Trulli', datetime.date(1974, 7, 13), 'Italian', 15, 'sauber', 'Sauber', 'Swiss'), (16, 'sutil', '99', 'SUT', 'Adrian', 'Sutil', datetime.date(1983, 1, 11), 'German', 16, 'bar', 'BAR', 'British'), (17, 'webber', '\\N', 'WEB', 'Mark', 'Webber', datetime.date(1976, 8, 27), 'Australian', 17, 'jordan', 'Jordan', 'Irish'), (18, 'button', '22', 'BUT', 'Jenson', 'Button', datetime.date(1980, 1, 19), 'British', 18, 'minardi', 'Minardi', 'Italian'), (19, 'davidson', '\\N', 'DAV', 'Anthony', 'Davidson', datetime.date(1979, 4, 18), 'British', 19, 'jaguar', 'Jaguar', 'British'), (20, 'vettel', '5', 'VET', 'Sebastian', 'Vettel', datetime.date(1987, 7, 3), 'German', 20, 'prost', 'Prost', 'French'), (21, 'fisichella', '\\N', 'FIS', 'Giancarlo', 'Fisichella', datetime.date(1973, 1, 14), 'Italian', 21, 'arrows', 'Arrows', 'British'), (22, 'barrichello', '\\N', 'BAR', 'Rubens', 'Barrichello', datetime.date(1972, 5, 23), 'Brazilian', 22, 'benetton', 'Benetton', 'Italian'), (23, 'ralf_schumacher', '\\N', 'SCH', 'Ralf', 'Schumacher', datetime.date(1975, 6, 30), 'German', 23, 'brawn', 'Brawn', 'British'), (24, 'liuzzi', '\\N', 'LIU', 'Vitantonio', 'Liuzzi', datetime.date(1980, 8, 6), 'Italian', 24, 'stewart', 'Stewart', 'British'), (25, 'wurz', '\\N', 'WUR', 'Alexander', 'Wurz', datetime.date(1974, 10, 16), 'Austrian', 25, 'mclaren', 'McLaren', 'British'), (26, 'marquez', '\\N', 'MAR', 'Marc', 'Marquez', datetime.date(1991, 7, 13), 'Spanish', 26, 'haas', 'Haas', 'German')]]

```

Grouping:

This script asks the user to choose from tables, then it will prompt you for two column for the database to join, it will then display the joined column for you.

```

Enter your choice (1-11): 8
Choose a table: circuits(1), constructor_results(2), constructor_standings(3), constructors(4), driver_standings(5), drivers(6), lap_times(7), pit_stops(8), qualifying(9), races(10), results(11), seasons(12), sprint_results(13), status(14): 6
Enter column name: driverRef
Enter second column name: driverId
(1, 'chaboud'), (1, 'hellings'), (1, 'riess'), (1, 'gasly'), (1, 'gilles villeneuve'), (1, 'loof'), (1, 'vergne'), (1, 'broeker'), (1, 'galica'), (1, 'elisian'), (1, 'jo schlesser'), (1, 'foit k'), (1, 'roos'), (1, 'puzey'), (1, 'pilette'), (1, 'gonzalez'), (1, 'seidel'), (1, 'cesaris'), (1, 'crossley'), (1, 'peterson'), (1, 'lawrence'), (1, 'seiffert'), (1, 'evans'), (1, 'pirocchi'), (1, 'londono'), (1, 'dolhem'), (1, 'button'), (1, 'galvez'), (1, 'scotti'), (1, 'ernesto brambilla'), (1, 'adams'), (1, 'capelli'), (1, 'gould'), (1, 'lees'), (1, 'villoreisi'), (1, 'bunduran'), (1, 'latifi'), (1, 'sala'), (1, 'lovely'), (1, 'hoffmann'), (1, 'raikkonen'), (1, 'mike taylor'), (1, 'francia'), (1, 'stevenson'), (1, 'garrett'), (1, 'bueno'), (1, 'gaze'), (1, 'jarier'), (1, 'sainz'), (1, 'parker'), (1, 'diniz'), (1, 'creus'), (1, 'ghinzani'), (1, 'ashdown'), (1, 'testut'), (1, 'mazepin'), (1, 'herman'), (1, 'wallard'), (1, 'zonta'), (1, 'ascari'), (1, 'boull on'), (1, 'tingelstad'), (1, 'badoer'), (1, 'la caze'), (1, 'montoya'), (1, 'arundell'), (1, 'amon'), (1, 'mazen'), (1, 'rigsby'), (1, 'balsa'), (1, 'gardner'), (1, 'hattori'), (1, 'rosa'), (1, 'klodwig'), (1, 'herrmann'), (1, 'campos'), (1, 'bruno senna'), (1, 'rodriguez'), (1, 'ball'), (1, 'simon'), (1, 'reed'), (1, 'larrauri'), (1, 'heyer'), (1, 'poele'), (1, 'flaherty'), (1, 'depailleur'), (1, 'weidler'), (1, 'adamich'), (1, 'filippis'), (1, 'drogo'), (1, 'gugelmin'), (1, 'musso'), (1, 'hulkenberg'), (1, 'hutchinson'), (1, 'ericsson'), (1, 'hartley'), (1, 'leoni'), (1, 'apicella'), (1, 'ratzenberger'), (1, 'bussiniello'), (1, 'hesnault'), (1, 'lehto'), (1, 'heidfeld'), (1, 'schell'), (1, 'larrousse'), (1, 'branca'), (1, 'gartner'), (1, 'patrese'), (1, 'louveau'), (1, 'john barber'), (1, 'marinon'), (1, 'monarch'), (1, 'kling'), (1, 'miles'), (1, 'massa'), (1, 'wurz'), (1, 'manzon'), (1, 'greene'), (1, 'piper'), (1, 'giovinazzi'), (1, 'berg'), (1, 'rosberg'), (1, 'ieckx'), (1, 'jackson'), (1, 'damon hill'), (1, 'blignaut'), (1, 'parsons'), (1, 'freeland'), (1, 'morgan'), (1, 'beauman'), (1, 'webber'), (1, 'grouillard'), (1, 'templeman'), (1, 'phil hill'), (1, 'klenk'), (1, 'cruz'), (1, 'blundell'), (1, 'kevin magnussen'), (1, 'richardson'), (1, 'aston'), (1, 'enge'), (1, 'emery'), (1, 'bianchi'), (1, 'maireisse'), (1, 'mansell'), (1, 'goldsmith'), (1, 'mags'), (1, 'spence'), (1, 'sospiri'), (1, 'nicholson'), (1, 'wehrlein'), (1, 'bob scott'), (1, 'roig'), (1, 'naspetti'), (1, 'markus winkelhoek'), (1, 'natili'), (1, 'prinoth'), (1, 'raby'), (1, 'pretorius'), (1, 'daly'), (1, 'mayer'), (1, 'scheckter'), (1, 'barber'), (1, 'edwards'), (1, 'moreno'), (1, 'burti'), (1, 'hans stuck'), (1, 'cevert'), (1, 'piotti'), (1, 'adolff

```

Subqueries:

Much like grouping script, once you choose a table, the user will be prompted to input a column from the table you chose and then another table and the column from that table.

```
Enter your choice (1-11): 9
Choose a table: circuits(1), constructor_results(2), constructor_standings(3), constructors(4), driver_standings(5), drivers(6), lap_times(7), pit_stops(8), qualifying(9), races(10), results(11), seasons(12), sprint_results(13), status(14): 1
Enter column name: circuitId
Enter second column name: raceId
Enter table name: races
[(1, 'albert_park', 'Albert Park Grand Prix Circuit', 'Melbourne', 'Australia'), (2, 'sepang', 'Sepang International Circuit', 'Kuala Lumpur', 'Malaysia'), (3, 'bahrain', 'Bahrain International Circuit', 'Sakhir', 'Bahrain'), (4, 'catalunya', 'Circuit de Barcelona-Catalunya', 'MontmelA', 'Spain'), (5, 'istanbul', 'Istanbul Park', 'Istanbul', 'Turkey'), (6, 'monaco', 'Circuit de Monaco', 'Monte-Carlo', 'Monaco'), (7, 'villeneuve', 'Circuit Gilles Villeneuve', 'Montreal', 'Canada'), (8, 'magny_cours', 'Circuit de Nevers Magny-Cours', 'Magny Cours', 'France'), (9, 'silverstone', 'Silverstone Circuit', 'Silverstone', 'UK'), (10, 'hockenheimring', 'Hockenheimring', 'Hockenheim', 'Germany'), (11, 'hungaroring', 'Hungaroring', 'Budapest', 'Hungary'), (12, 'valencia', 'Valencia Street Circuit', 'Valencia', 'Spain'), (13, 'spa', 'Circuit de Spa-Francorchamps', 'Spa', 'Belgium'), (14, 'monza', 'Autodromo Nazionale di Monza', 'Monza', 'Italy'), (15, 'marina_bay', 'Marina Bay Street Circuit', 'Marina Bay', 'Singapore'), (16, 'fuji', 'Fuji Speedway', 'Oyama', 'Japan'), (17, 'shanghai', 'Shanghai International Circuit', 'Shanghai', 'China'), (18, 'interlagos', 'Autodromo Jose Carlos Pace', 'Sao Paulo', 'Brazil'), (19, 'indianapolis', 'Indianapolis Motor Speedway', 'Indianapolis', 'USA'), (20, 'nurburgring', 'Nurburgring', 'Nurburg', 'Germany'), (21, 'imola', 'Autodromo Enzo e Dino Ferrari', 'Imola', 'Italy'), (22, 'suzuka', 'Suzuka Circuit', 'Suzuka', 'Japan'), (23, 'las_vegas', 'Las Vegas Strip Street Circuit', 'Las Vegas', 'United States'), (24, 'yas_marina', 'Yas Marina Circuit', 'Abu Dhabi', 'UAE'), (25, 'galvez', 'Autodromo Juan y Oscar Galvez', 'Buenos Aires', 'Argentina'), (26, 'jerez', 'Circuito de Jerez', 'Jerez de la Frontera', 'Spain'), (27, 'estoril', 'AutAdromo do Estoril', 'Estoril', 'Portugal'), (28, 'okayama', 'Okayama International Circuit', 'Okayama', 'Japan'), (29, 'adelaide', 'Adelaide Street Circuit', 'Adelaide', 'Australia'), (30, 'kyalami', 'Kyalami', 'Midrand', 'South Africa'), (31, 'donington', 'Donington Park', 'Castle Donington', 'UK'), (32, 'rodriguez', 'Autodromo Hermanos Rodriguez', 'Mexico City', 'Mexico'), (33, 'phoenix', 'Phoenix street circuit', 'Phoenix', 'USA'), (34, 'ricard', 'Circuit Paul Ricard', 'Le Castellet', 'France'), (35, 'yeongam', 'Korean International Circuit', 'Yeongam County', 'Korea'), (36, 'jacarepagua', 'AutAdromo Internacional Nelson Piquet', 'Rio de Janeiro', 'Brazil'), (37, 'detroit', 'Detroit Street Circuit', 'Detroit', 'USA'), (38, 'brands_hatch', 'Brands Hatch', 'Kent', 'UK'), (39, 'zandvoort', 'Circuit Park Zandvoort', 'Zandvoort', 'Netherlands'), (40, 'zolder', 'Zolder', 'Heusden-Zolder', 'Belgium'), (41, 'dijon', 'Dijon-Prenois', 'Dijon', 'France'), (42, 'dallas', 'Fair Park', 'Dallas', 'USA'), (43, 'long_beach', 'Long Beach', 'California', 'USA'), (44, 'las_vegas', 'Las Vegas Street Circuit', 'Nevada', 'USA'), (45, 'jarama', 'Jarama', 'Madrid', 'Spain'), (46, 'watkins_glen', 'Watkins Glen', 'New York State', 'USA'), (47, 'anderstorp', 'Scandinavian Raceway', 'Anderstorp', 'Sweden'), (48, 'mosport', 'Mosport International Raceway', 'Ontario', 'Canada'), (49, 'montjuic', 'Montjuic', 'Barcelona', 'Spain'), (50, 'niveles', 'Niveles-Baulers', 'Brussels', 'Belgium'), (51, 'charade', 'Charade Circuit', 'Clermont-Ferrand', 'France'), (52, 'tremblant', 'Circuit Mont-Tremblant', 'Quebec', 'Canada'), (53, 'essarts', 'Rouen-Les-Essarts', 'Rouen', 'France'), (54, 'lemans', 'Le Mans', 'France')]
```

Transaction:

In this script you are not prompted to choose from a table, instead you are typing the table as explained above. This is to provide more flexibility, you are however prompted to choose if you want to perform a transaction or a rollback, if you choose to do a transaction the SQL will update the table, if you choose to rollback the transaction will redo the most recent change.