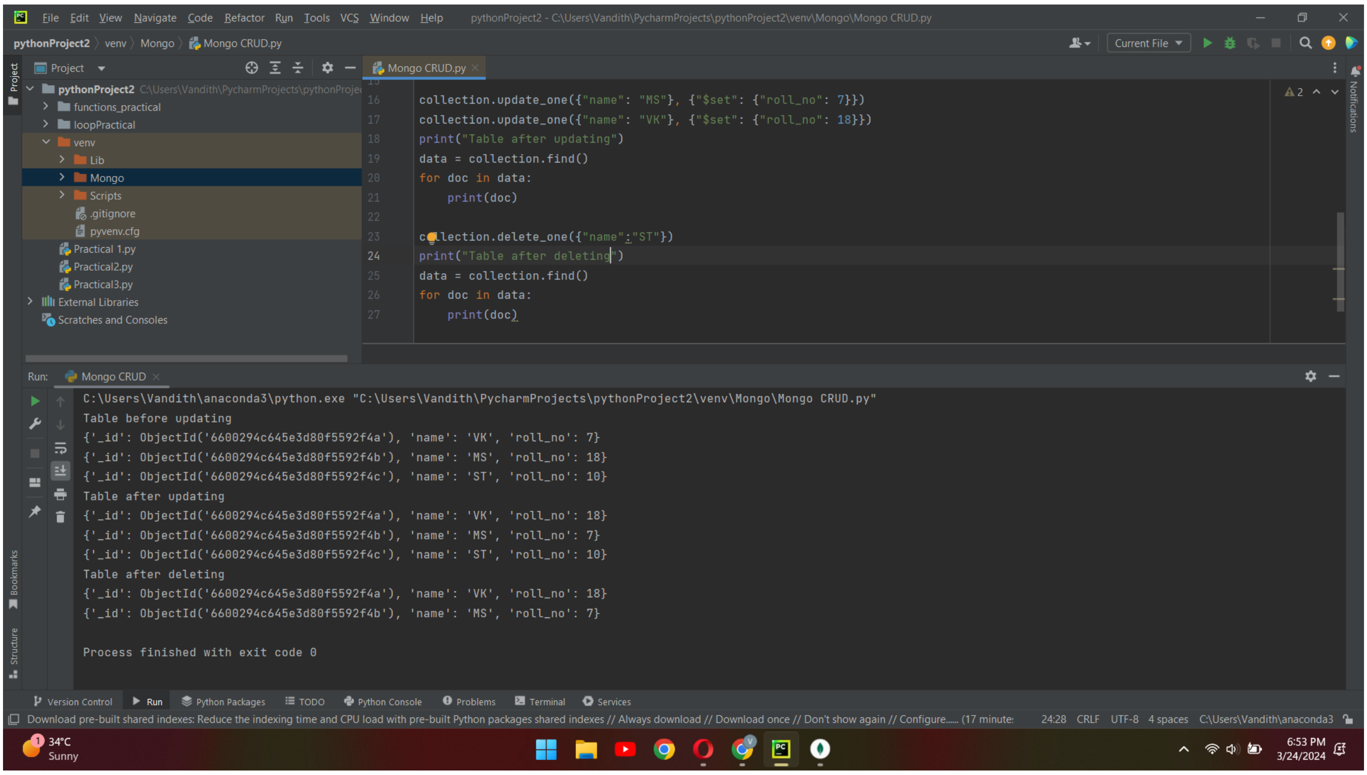
1. Write a program to perform CRUD operations using mongoDB shell and pycharm IDE

C-Create, R-Read, U-Update, D-Delete

```python
import pymongo

client = pymongo.MongoClient("mongodb://localhost:27017")
db = client["Python"]
collection = db["pythonCRUD"]

collection.insert_one({"name": "VK", "roll_no": 7})
collection.insert_one({"name": "MS", "roll_no": 18})
collection.insert_one({"name": "ST", "roll_no": 10})

print("Table before updating")
data = collection.find()
for doc in data:
    print(doc)

collection.update_one({"name": "MS"}, {"$set": {"roll_no": 7}})
collection.update_one({"name": "VK"}, {"$set": {"roll_no": 18}})
print("Table after updating")
data = collection.find()
for doc in data:
    print(doc)

collection.delete_one({"name":"ST"})
print("Table after deleting")
data = collection.find()
for doc in data:
    print(doc)
```



File Edit View Navigate Code Refactor Run Tools VCS Window Help
pythonProject2 - C:\Users\Vandith\PycharmProjects\pythonProject2\venv\Mongo\Mongo CRUD.py
pythonProject2 › venv › Mongo › Mongo CRUD.py
Project
pythonProject2 C:\Users\Vandith\PycharmProjects\pythonProje
functions_practical
loopPractical
venv
Lib
Mongo
Scripts
.gitignore
pyvenv.cfg
Practical 1.py
Practical2.py
Practical3.py
External Libraries
Scratches and Consoles
Mongo CRUD.py
Run: Mongo CRUD
Version Control
Run
Python Packages
TODO
Python Console
Problems
Terminal
Services
24:28 CRLF UTF-8 4 spaces C:\Users\Vandith\anaconda3
34°C
Sunny
6:53 PM
3/24/2024

```python
collection.update_one({"name": "MS"}, {"$set": {"roll_no": 7}})
collection.update_one({"name": "VK"}, {"$set": {"roll_no": 18}})
print("Table after updating")
data = collection.find()
for doc in data:
    print(doc)

collection.delete_one({"name":"ST"})
print("Table after deleting")
data = collection.find()
for doc in data:
    print(doc)
```

```
C:\Users\Vandith\anaconda3\python.exe "C:\Users\Vandith\PycharmProjects\pythonProject2\venv\Mongo\Mongo CRUD.py"
Table before updating
{'_id': ObjectId('6600294c645e3d80f5592f4a'), 'name': 'VK', 'roll_no': 7}
{'_id': ObjectId('6600294c645e3d80f5592f4b'), 'name': 'MS', 'roll_no': 18}
{'_id': ObjectId('6600294c645e3d80f5592f4c'), 'name': 'ST', 'roll_no': 10}
Table after updating
{'_id': ObjectId('6600294c645e3d80f5592f4a'), 'name': 'VK', 'roll_no': 18}
{'_id': ObjectId('6600294c645e3d80f5592f4b'), 'name': 'MS', 'roll_no': 7}
{'_id': ObjectId('6600294c645e3d80f5592f4c'), 'name': 'ST', 'roll_no': 10}
Table after deleting
{'_id': ObjectId('6600294c645e3d80f5592f4a'), 'name': 'VK', 'roll_no': 18}
{'_id': ObjectId('6600294c645e3d80f5592f4b'), 'name': 'MS', 'roll_no': 7}

Process finished with exit code 0
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download once // Don't show again // Configure... (17 minute:

2. Create a pandas dataFrame using mtcars.csv CSV file and perform a following

a) Display column names

import pandas as pd

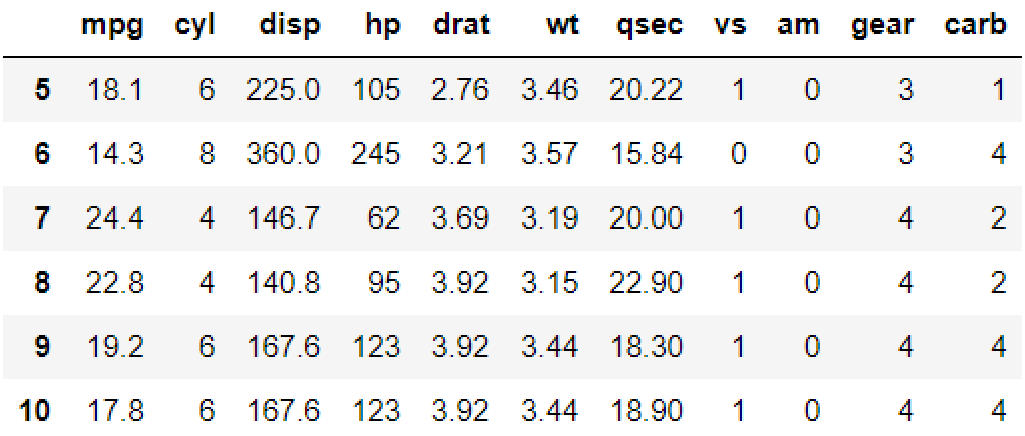
cars_data=pd.read_csv('mtcars.csv')

cars_data.columns

```
Index(['mpg', 'cyl', 'disp', 'hp', 'drat', 'wt', 'qsec', 'vs', 'am', 'g
ear','carb'], dtype='object')
```

b) Display 5th to 10th rows

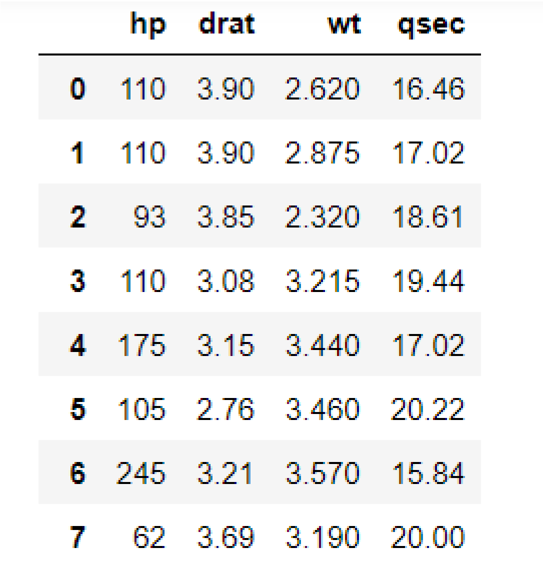
cars_data.iloc[5:11]



	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
5	18.1	6	225.0	105	2.76	3.46	20.22	1	0	3	1
6	14.3	8	360.0	245	3.21	3.57	15.84	0	0	3	4
7	24.4	4	146.7	62	3.69	3.19	20.00	1	0	4	2
8	22.8	4	140.8	95	3.92	3.15	22.90	1	0	4	2
9	19.2	6	167.6	123	3.92	3.44	18.30	1	0	4	4
10	17.8	6	167.6	123	3.92	3.44	18.90	1	0	4	4

c) Display 4th to 7th columns

cars_data.iloc[:,3:7]



	hp	drat	wt	qsec
0	110	3.90	2.620	16.46
1	110	3.90	2.875	17.02
2	93	3.85	2.320	18.61
3	110	3.08	3.215	19.44
4	175	3.15	3.440	17.02
5	105	2.76	3.460	20.22
6	245	3.21	3.570	15.84
7	62	3.69	3.190	20.00

d) Display no of rows and no of columns

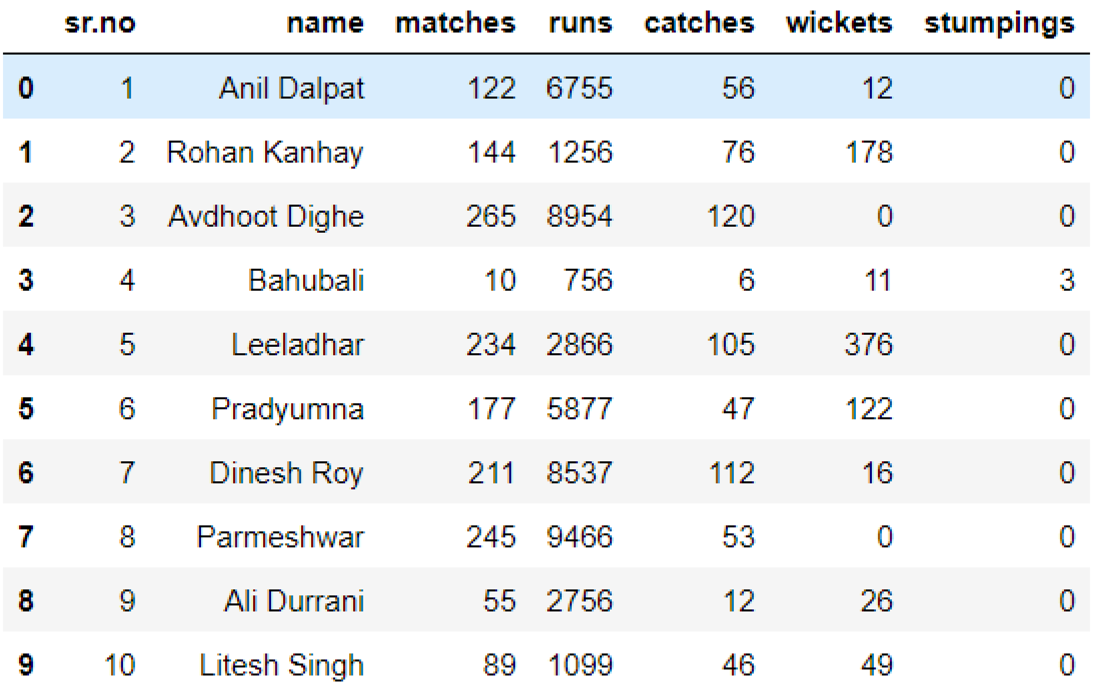
cars_data.shape

```
(32, 11)
```

3. Use the given file named cricket.csv and perform the following operations:

1. Read the file in DataFrame

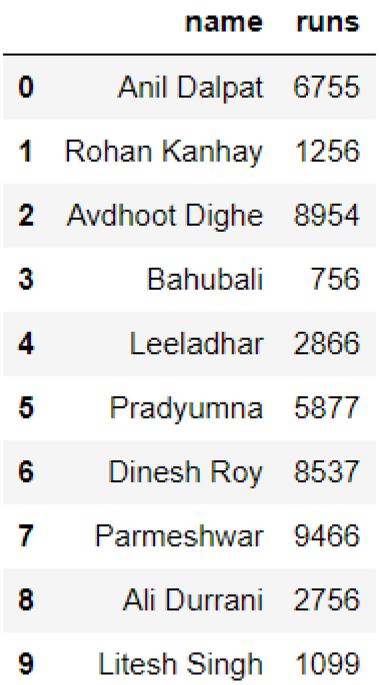
cricket=pd.read_csv("cricket.csv")
cricket



	sr.no	name	matches	runs	catches	wickets	stumpings
0	1	Anil Dalpat	122	6755	56	12	0
1	2	Rohan Kanhay	144	1256	76	178	0
2	3	Avdhoot Dighe	265	8954	120	0	0
3	4	Bahubali	10	756	6	11	3
4	5	Leeladhar	234	2866	105	376	0
5	6	Pradyumna	177	5877	47	122	0
6	7	Dinesh Roy	211	8537	112	16	0
7	8	Parmeshwar	245	9466	53	0	0
8	9	Ali Durrani	55	2756	12	26	0
9	10	Litesh Singh	89	1099	46	49	0

2. List the name of cricketer and their respective runs

cricket[['name','runs']]



	name	runs
0	Anil Dalpat	6755
1	Rohan Kanhay	1256
2	Avdhoot Dighe	8954
3	Bahubali	756
4	Leeladhar	2866
5	Pradyumna	5877
6	Dinesh Roy	8537
7	Parmeshwar	9466
8	Ali Durrani	2756
9	Litesh Singh	1099

3. Find total wickets taken by them

```
cricket['wickets'].sum()
```

```
790
```

4. Find average of catches taken

```
cricket['catches'].mean()
```

```
63.3
```

5. Find the name of wicketkeeper

```
wicket_keeper=cricket[cricket['stumpings']>0].name

print(wicketkeeper)
```

```
3    Bahubali
Name: name, dtype: object
```

6. Print the name of bowler who played highest number of matches

```
max_matches_bowler = cricket[cricket['matches'] == cricket['matches'].max()].name

print(max_matches_bowler)
```

```
2    Avdhoot Dighe
Name: name, dtype: object
```

7. Find average of all the bowlers

```
average_bowlers = cricket[cricket['wickets'] > 0]['wickets'].mean()

print(average_bowlers)
```

```
98.75
```

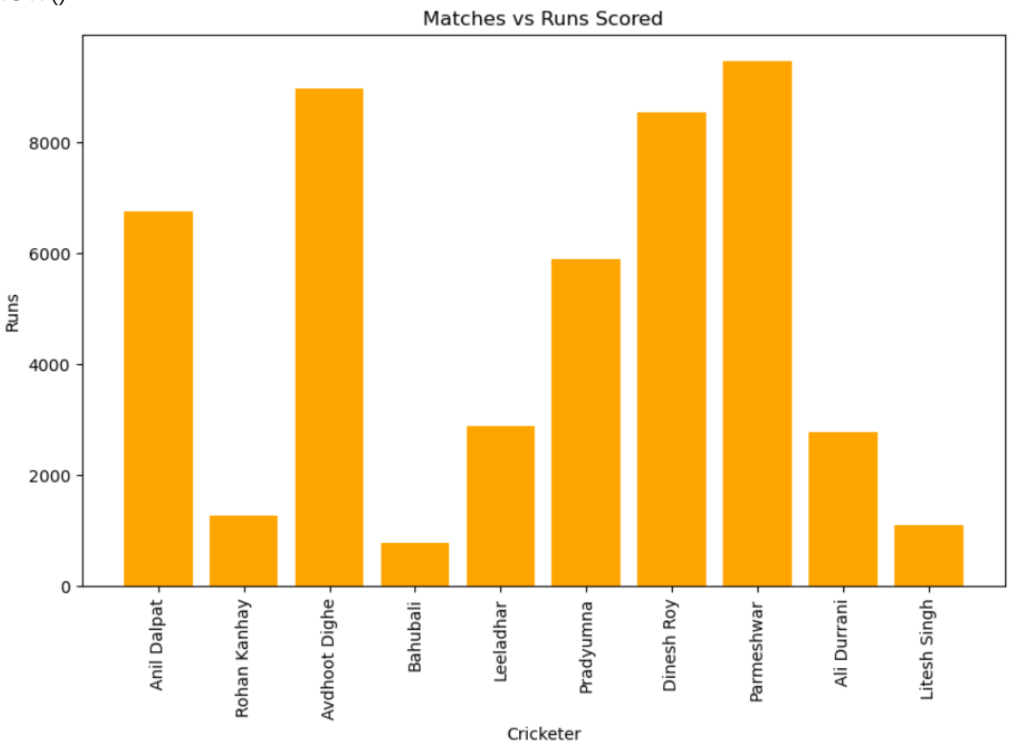
8. Find name of the bowler with least bowling average

```
min_average_bowler = cricket[cricket['wickets'] > 0].sort_values(by='wickets', ascending=True).iloc[0]['name']

print(min_average_bowler)
```

```
Bahubali
```

9. Draw the bar chart of matches against number of runs scored

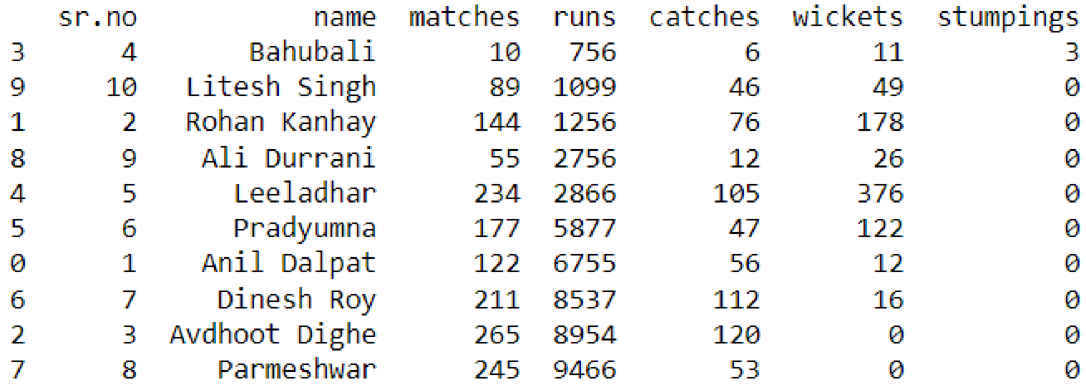
```
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))

plt.bar(cricket['name'], cricket['runs'], color='skyblue')

plt.xlabel('Cricketer')

plt.ylabel('Runs')

plt.title('Matches vs Runs Scored')

plt.xticks(rotation=90)

plt.show()
```



10. Sort and print information about players by ascending order of runs

sorted_cricket = cricket.sort_values(by='runs', ascending=True)

print(sorted_cricket)

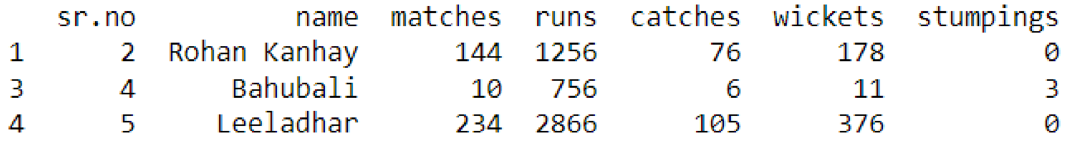


```
   sr.no           name  matches  runs  catches  wickets  stumpings
3      4       Bahubali       10   756        6       11          3
9     10   Litesh Singh       89  1099       46       49          0
1      2   Rohan Kanhay      144  1256       76      178          0
8      9    Ali Durrani       55  2756       12       26          0
4      5      Leeladhar      234  2866      105      376          0
5      6      Pradyumna      177  5877       47      122          0
0      1    Anil Dalpat      122  6755       56       12          0
6      7     Dinesh Roy      211  8537      112       16          0
2      3  Avdhoot Dighe      265  8954      120        0          0
7      8     Parmeshwar      245  9466       53        0          0
```

11. Print the names of players whose wickets are greater than matches

wickets_greater_than_matches = cricket[cricket['wickets']>cricket['matches']]

print(wickets_greater_than_matches)



```
   sr.no          name  matches  runs  catches  wickets  stumpings
1      2  Rohan Kanhay      144  1256       76      178          0
3      4      Bahubali       10   756        6       11          3
4      5     Leeladhar      234  2866      105      376          0
```