

**Database II - CSEN 604**  
**Project – Part II**  
**Team 8**

**Part A:**

**A.Q1**

[('Seq Scan on played\_in (cost=0.00..1113.00 rows=8313 width=22)'), (' Filter: ("position" = 1)'),)]

**A.Q2**

cost=0.00..1113.00

**A.Q3**

After posIndex

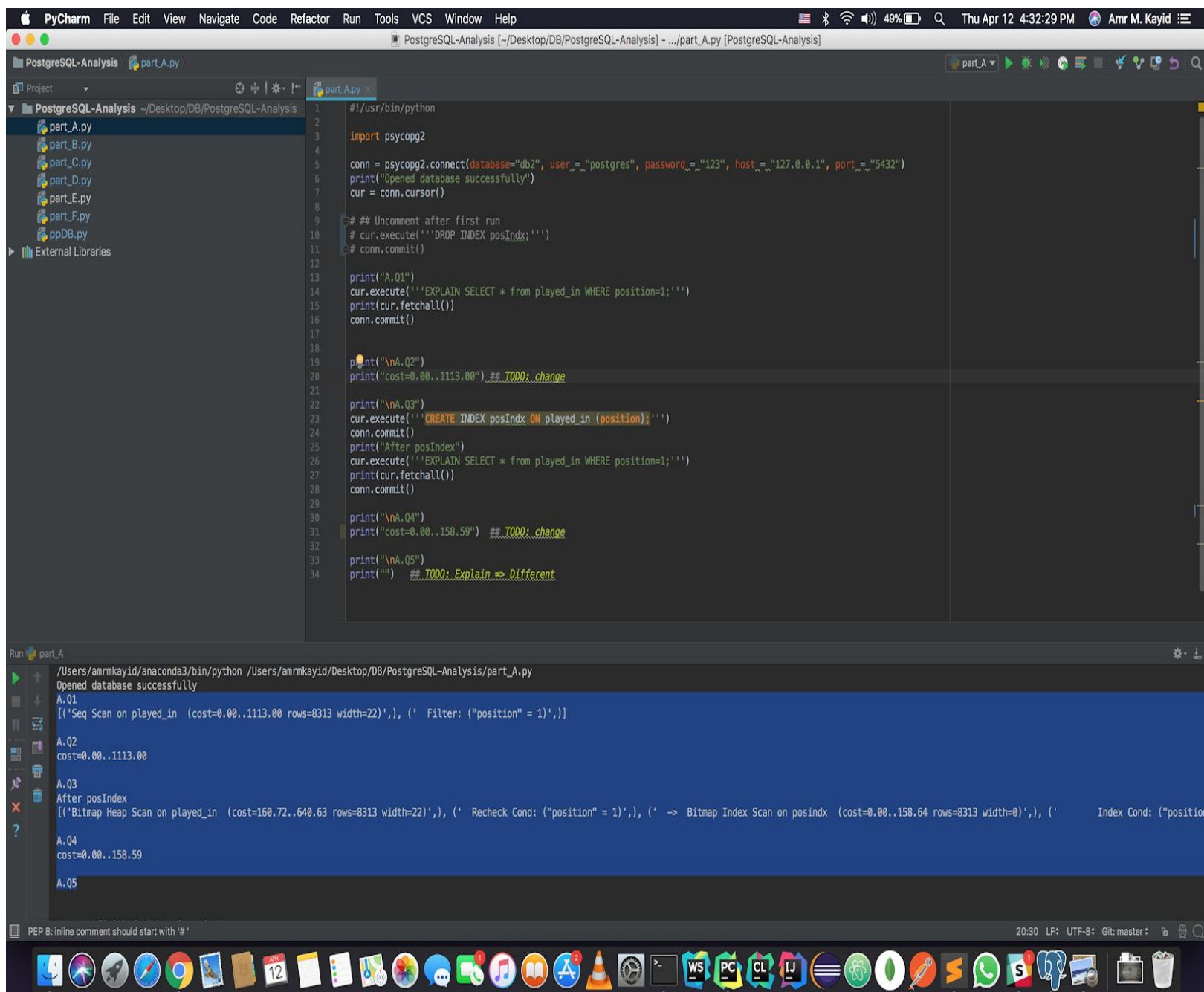
[('Bitmap Heap Scan on played\_in (cost=160.72..640.63 rows=8313 width=22)'), (' Recheck Cond: ("position" = 1)'), (' -> Bitmap Index Scan on posindx (cost=0.00..158.64 rows=8313 width=0)'), (' Index Cond: ("position" = 1)'),)]

**A.Q4**

cost=0.00..158.64

**A.Q5**

The planner uses the created index, and thus the retrieval from the database is faster.



## Part B:

### B.Q1

[('Seq Scan on played\_in (cost=0.00..1113.00 rows=6 width=22)',), (" Filter: (name ~~ '%pele%':::text)",,)]

### **B.Q2**

cost=0.00..1113.00

### **B.Q3**

After nameIndx

[('Seq Scan on played\_in (cost=0.00..1113.00 rows=6 width=22)',), (" Filter: (name ~~ '%pele%':::text)",,)]

### **B.Q4**

cost=0.00..1113.00

### **B.Q5**

The index is not on the field that the select is done, so it does not expedite the query execution.

The screenshot displays the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The title bar shows the project name "PostgreSQL-Analysis" and the file path. The left sidebar shows the project structure with files part\_A.py, part\_B.py, part\_C.py, part\_D.py, part\_E.py, part\_F.py, and ppDB.py. The main editor window shows the code for part\_B.py, which is a Python script using psycopg2 to connect to a PostgreSQL database and execute SQL queries. The script includes comments and prints the results of the queries. The bottom panel shows the output of the script execution, which includes the following text:

```
Run part_B
/Users/ammkayid/anaconda3/bin/python /Users/ammkayid/Desktop/DB/PostgreSQL-Analysis/part_B.py
Opened database successfully
B.Q1
[('Seq Scan on played_in (cost=0.00..1113.00 rows=6 width=22)',), (' Filter: (name ~ '%pele%':text)',)]
B.Q2
cost=0.00..1113.00
A.Q3
After nameIndx
[('Seq Scan on played_in (cost=0.00..1113.00 rows=6 width=22)',), (' Filter: (name ~ '%pele%':text)',)]
B.Q4
cost=0.00..1113.00
B.Q5
```

## Part C:

### **C.Q1**

```
[('Seq Scan on cup_matches (cost=0.00..58.20 rows=893  
width=24)'), (" Filter: ((rating * '3'::double precision) >  
'20'::double precision)" ,)]
```

### **C.Q2**

cost=0.00..58.20

### **C.Q3**

After posIndex

```
[('Seq Scan on cup_matches (cost=0.00..58.20 rows=893  
width=24)'), (" Filter: ((rating * '3'::double precision) >  
'20'::double precision)" ,)]
```

### **C.Q4**

cost=0.00..58.20

### **C.Q5**

The index is not on the field that the select is done, so it does not expedite the query execution.

PyCharm File Edit View Navigate Code Refactor Run Tools VCS Window Help

PostgreSQL-Analysis [~/Desktop/DB/PostgreSQL-Analysis] - .../part\_C.py [PostgreSQL-Analysis]

Project PostgreSQL-Analysis ~/Desktop/DB/PostgreSQL-Analysis

- part\_A.py
- part\_B.py
- part\_C.py
- part\_D.py
- part\_E.py
- part\_F.py
- ppDB.py
- External Libraries

```
1 #!/usr/bin/python
2
3 import psycopg2
4
5 conn = psycopg2.connect(database="db2", user="postgres", password="123", host="127.0.0.1", port="5432")
6 print("Opened database successfully")
7 cur = conn.cursor()
8
9 ## Uncomment after first run
10 # cur.execute('DROP INDEX ratingIndex;')
11 # conn.commit()
12
13 print("C.01")
14 cur.execute('EXPLAIN SELECT * from cup_matches WHERE rating*3 > 20;')
15 print(cur.fetchall())
16 conn.commit()
17
18
19
20 print("\nC.02")
21 print("cost=0.00..58.20") ## TODO: change
22
23
24 print("\nC.03")
25 cur.execute('CREATE INDEX ratingIndex ON cup_matches (rating);')
26 conn.commit()
27 print("After posIndex")
28 cur.execute('EXPLAIN SELECT * from cup_matches WHERE rating*3 > 20;')
29 print(cur.fetchall())
30 conn.commit()
31
32 print("\nC.04")
33 print("cost=0.00..58.20") ## TODO: change
34
35 print("\nC.05")
36 print("") ## TODO: Explain => Same!!
```

Run part\_C

/Users/amrmkayid/anaconda3/bin/python /Users/amrmkayid/Desktop/DB/PostgreSQL-Analysis/part\_C.py

Opened database successfully

C.01

[('Seq Scan on cup\_matches (cost=0.00..58.20 rows=893 width=24)',), (' Filter: ((rating \* '3'::double precision) > '20'::double precision)',)]

C.02

cost=0.00..58.20

C.03

After posIndex

[('Seq Scan on cup\_matches (cost=0.00..58.20 rows=893 width=24)',), (' Filter: ((rating \* '3'::double precision) > '20'::double precision)',)]

C.04

cost=0.00..58.20

C.05

2:29 LF UTF-8 Git: master

## **Part D:**

### **D.Q1**

```
[('Hash Join (cost=78.30..5195.97 rows=304657
width=46)'), (' Hash Cond: (played_in.year =
cup_matches.year)'), (' -> Seq Scan on played_in
(cost=0.00..965.60 rows=58960 width=22)'), (' -> Hash
(cost=44.80..44.80 rows=2680 width=24)'), (' -> Seq
Scan on cup_matches (cost=0.00..44.80 rows=2680
width=24)'),)]
```

### **D.Q2**

After cup\_yearIndx

```
[('Hash Join (cost=78.30..5195.97 rows=304657
width=46)'), (' Hash Cond: (played_in.year =
cup_matches.year)'), (' -> Seq Scan on played_in
(cost=0.00..965.60 rows=58960 width=22)'), (' -> Hash
(cost=44.80..44.80 rows=2680 width=24)'), (' -> Seq
Scan on cup_matches (cost=0.00..44.80 rows=2680
width=24)'),)]
```

### **D.Q3**

The index is not on the field that the select is done, so it does not expedite the query execution.

### **D.Q4**

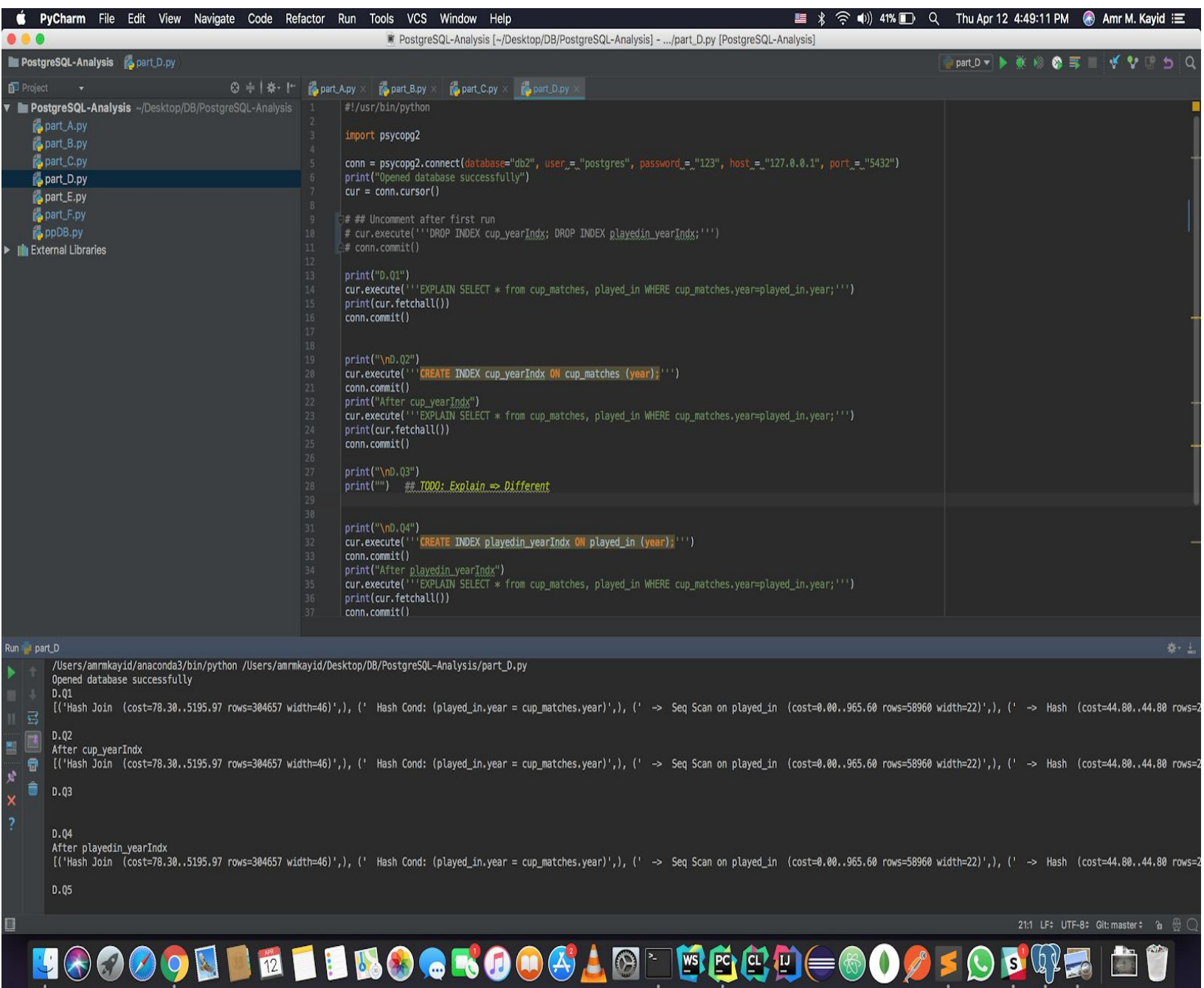


After playedin\_yearIndx

```
[('Hash Join (cost=78.30..5195.97 rows=304657
width=46)'), (' Hash Cond: (played_in.year =
cup_matches.year)'), (' -> Seq Scan on played_in
(cost=0.00..965.60 rows=58960 width=22)'), (' -> Hash
(cost=44.80..44.80 rows=2680 width=24)'), (' -> Seq
Scan on cup_matches (cost=0.00..44.80 rows=2680
width=24)'),)]
```

### **D.Q5**

The index is not on the field that the select is done, so it does not expedite the query execution.



## **Part E:**

### **E.Q1**

[('Hash Join' (cost=78.30..1788.07 rows=58960 width=46)'), (' Hash Cond: (played\_in.mid = cup\_matches.mid)'), (' -> Seq Scan on played\_in' (cost=0.00..965.60 rows=58960 width=22)'), (' -> Hash' (cost=44.80..44.80 rows=2680 width=24)'), (' -> Seq Scan on cup\_matches' (cost=0.00..44.80 rows=2680 width=24)'),)]

### **E.Q2**

[('Hash Join' (cost=78.30..1788.07 rows=58960 width=46)'), (' Hash Cond: (played\_in.mid = cup\_matches.mid)'), (' -> Seq Scan on played\_in' (cost=0.00..965.60 rows=58960 width=22)'), (' -> Hash' (cost=44.80..44.80 rows=2680 width=24)'), (' -> Seq Scan on cup\_matches' (cost=0.00..44.80 rows=2680 width=24)'),)]

## E.Q3

[('Nested Loop (cost=0.41..5466.40 rows=58960 width=46)'), (' -> Seq Scan on cup\_matches (cost=0.00..44.80 rows=2680 width=24)'), (' -> Index Scan using played\_in\_pkey on played\_in (cost=0.41..1.80 rows=22 width=22)'), (' Index Cond: (mid = cup\_matches.mid)')]

The screenshot shows the PyCharm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The main editor window displays a Python script named `part_E.py` with the following content:

```
#!/usr/bin/python
import psycopg2
conn = psycopg2.connect(database="db2", user="postgres", password="123", host="127.0.0.1", port="5432")
print("Opened database successfully")
cur = conn.cursor()

print("E.Q1")
cur.execute('EXPLAIN SELECT * FROM cup_matches, played_in WHERE cup_matches.mid=played_in.mid;')
print(cur.fetchall())
conn.commit()

print("\nE.Q2")
cur.execute('SET enable_mergejoin = FALSE;')
cur.execute('EXPLAIN SELECT * FROM cup_matches, played_in WHERE cup_matches.mid=played_in.mid;')
print(cur.fetchall())
conn.commit()

print("\nE.Q3")
cur.execute('SET enable_hashjoin = FALSE;')
cur.execute('EXPLAIN SELECT * FROM cup_matches, played_in WHERE cup_matches.mid=played_in.mid;')
print(cur.fetchall())
conn.commit()
```

The bottom panel shows the Run output for `part_E`. The output indicates that the database was opened successfully and displays the execution plans for three queries (E.Q1, E.Q2, and E.Q3). The output for E.Q3 is:

```
E.Q3
[('Nested Loop (cost=0.41..5466.40 rows=58960 width=46)'), (' -> Seq Scan on cup_matches (cost=0.00..44.80 rows=2680 width=24)'), (' -> Index Scan using played_in_pkey on played_in (cost=0.41..1.80 rows=22 width=22)'), (' Index Cond: (mid = cup_matches.mid)')]
```

The process finished with exit code 0. The bottom status bar shows the file encoding as UTF-8 and the Git branch as master.

# **Part F**

## **Part A:**

### **A.Q1**

[('Seq Scan on played\_in (cost=0.00..197976.64 rows=1462688 width=24)'), (' Filter: ("position" = 1)'),)]

### **A.Q2**

cost=0.00..197976.64

### **A.Q3**

After posIndex

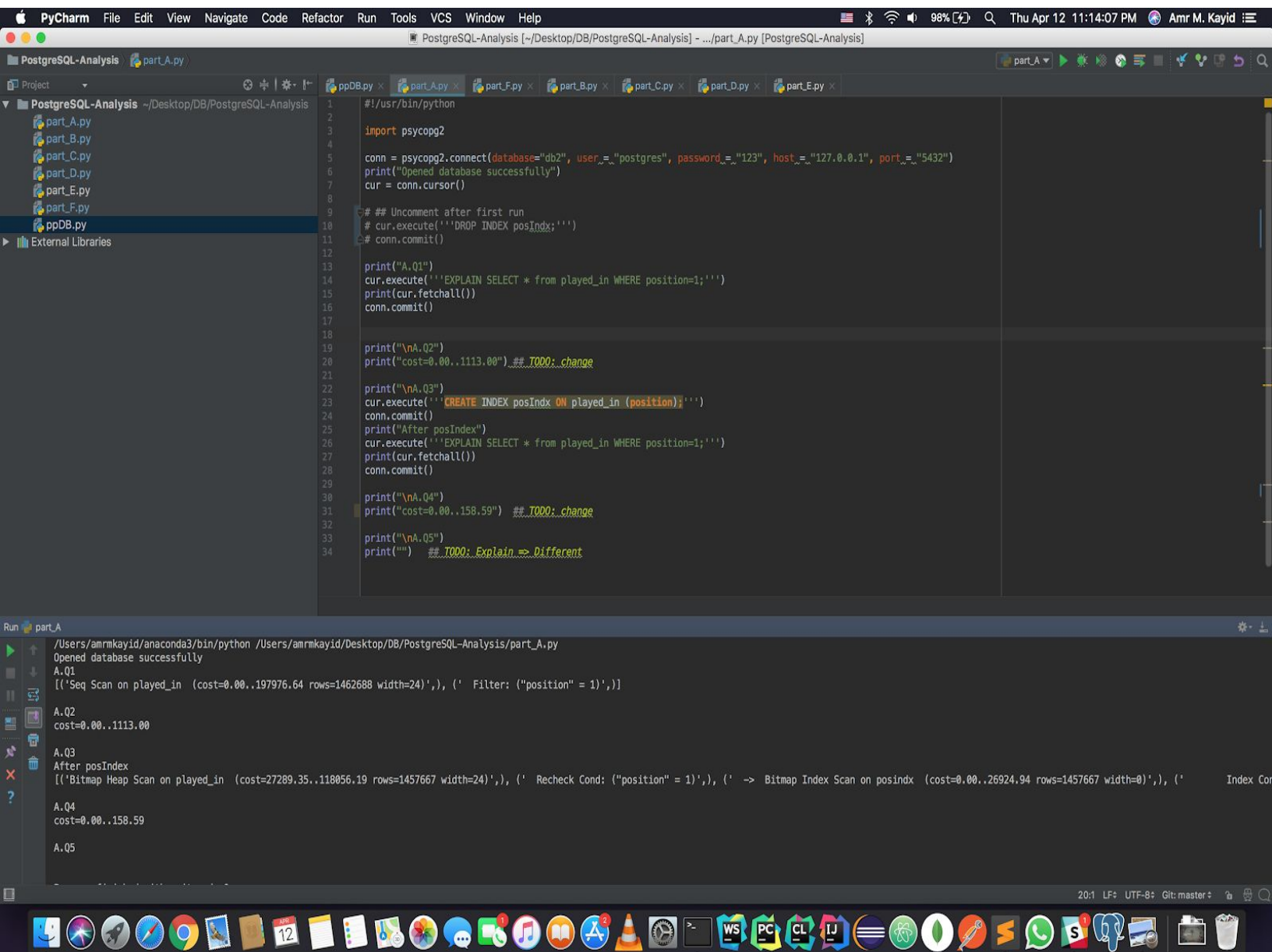
[('Bitmap Heap Scan on played\_in (cost=27289.35..118056.19 rows=1457667 width=24)'), (' Recheck Cond: ("position" = 1)'), (' -> Bitmap Index Scan on posindx (cost=0.00..26924.94 rows=1457667 width=0)'), (' Index Cond: ("position" = 1)'),)]

### **A.Q4**

cost=0.00..26924.94

## A.Q5

The planner uses the created index, and thus the retrieval from the database is faster.



```
1 #!/usr/bin/python
2
3 import psycopg2
4
5 conn = psycopg2.connect(database="db2", user="postgres", password="123", host="127.0.0.1", port="5432")
6 print("Opened database successfully")
7 cur = conn.cursor()
8
9 ## Uncomment after first run
10 # cur.execute('DROP INDEX posIndex;')
11 # conn.commit()
12
13 print("A.Q1")
14 cur.execute('EXPLAIN SELECT * from played_in WHERE position=1;')
15 print(cur.fetchall())
16 conn.commit()
17
18
19 print("\nA.Q2")
20 print("cost=0.00..1113.00") ## T000: change
21
22 print("\nA.Q3")
23 cur.execute('CREATE INDEX posIndex ON played_in (position);')
24 conn.commit()
25 print("After posIndex")
26 cur.execute('EXPLAIN SELECT * from played_in WHERE position=1;')
27 print(cur.fetchall())
28 conn.commit()
29
30 print("\nA.Q4")
31 print("cost=0.00..158.59") ## T000: change
32
33 print("\nA.Q5")
34 print("") ## T000: Explain => Different
```

Run part\_A

```
/Users/amrmkayid/anaconda3/bin/python /Users/amrmkayid/Desktop/DB/PostgreSQL-Analysis/part_A.py
Opened database successfully
A.Q1
[('Seq Scan on played_in (cost=0.00..197976.64 rows=1462688 width=24)'), (' Filter: ("position" = 1)',)]

A.Q2
cost=0.00..1113.00

A.Q3
After posIndex
[('Bitmap Heap Scan on played_in (cost=27289.35..118056.19 rows=1457667 width=24)'), (' Recheck Cond: ("position" = 1)', (' -> Bitmap Index Scan on posIndex (cost=0.00..26924.94 rows=1457667 width=0)'), (' Index Cond: ("position" = 1)',)]

A.Q4
cost=0.00..158.59

A.Q5
```

## **Part B:**

### **B.Q1**

[('Gather (cost=1000.00..125729.33 rows=1000 width=24)'), (' Workers Planned: 2'), (' -> Parallel Seq Scan on played\_in (cost=0.00..124629.33 rows=417 width=24)'), (' Filter: (name ~~ '%pele%':::text)'),]

### **B.Q2**

cost=0.00..124629.33

### **A.Q3**

After nameIndx

[('Gather (cost=1000.00..125729.33 rows=1000 width=24)'), (' Workers Planned: 2'), (' -> Parallel Seq Scan on played\_in (cost=0.00..124629.33 rows=417 width=24)'), (' Filter: (name ~~ '%pele%':::text)'),]

### **B.Q4**

cost=0.00..124629.33

### **B.Q5**

The index is not on the field that the select is done, so it does not expedite the query execution.



PyCharm File Edit View Navigate Code Refactor Run Tools VCS Window Help

PostgreSQL-Analysis [~/Desktop/DB/PostgreSQL-Analysis] - .../part\_B.py [PostgreSQL-Analysis]

Project PostgreSQL-Analysis ~/Desktop/DB/PostgreSQL-Analysis

part\_A.py part\_B.py part\_C.py part\_D.py part\_E.py part\_F.py ppDB.py

External Libraries

```
1 #!/usr/bin/python
2
3 import psycopg2
4
5 conn = psycopg2.connect(database="db2", user="postgres", password="123", host="127.0.0.1", port="5432")
6 print("Opened database successfully")
7 cur = conn.cursor()
8
9 ## Uncomment after first run
10 # cur.execute('DROP INDEX nameIndx;')
11 # conn.commit()
12
13 print("B.Q1")
14 cur.execute('EXPLAIN SELECT * from played_in WHERE name like \'apele%\';')
15 print(cur.fetchall())
16 conn.commit()
17
18 print("\nB.Q2")
19 print("cost=0.00..1113.00") ## T000: change
20
21
22 print("\nA.Q3")
23 cur.execute('CREATE INDEX nameIndx ON played_in (name);')
24 conn.commit()
25 print("After nameIndx")
26 cur.execute('EXPLAIN SELECT * from played_in WHERE name like \'apele%\';')
27 print(cur.fetchall())
28 conn.commit()
29
30 print("\nB.Q4")
31 print("cost=0.00..1113.00") ## T000: change
32
33 print("\nB.Q5")
34 print("") ## T000: Explain => Same
```

Run part\_B

/Users/ammkayid/anaconda3/bin/python /Users/ammkayid/Desktop/DB/PostgreSQL-Analysis/part\_B.py

Opened database successfully

B.Q1

[('Gather (cost=1000.00..125729.33 rows=1000 width=24)'), (' Workers Planned: 2'), (' -> Parallel Seq Scan on played\_in (cost=0.00..124629.33 rows=417 width=24)'), (' Filter: (name ~ \'apele%\'::text)'), ('

B.Q2

cost=0.00..1113.00

A.Q3

After nameIndx

[('Gather (cost=1000.00..125729.33 rows=1000 width=24)'), (' Workers Planned: 2'), (' -> Parallel Seq Scan on played\_in (cost=0.00..124629.33 rows=417 width=24)'), (' Filter: (name ~ \'apele%\'::text)'), ('

B.Q4

cost=0.00..1113.00

B.Q5

4:74 LF UTF-8 Git: master



## **Part C:**

### **C.Q1**

```
[('Seq Scan on cup_matches (cost=0.00..10685.00  
rows=166667 width=24)'), (" Filter: ((rating * '3'::double  
precision) > '20'::double precision)",)]
```

### **C.Q2**

```
cost=0.00..10685.00
```

### **C.Q3**

After posIndex

```
[('Seq Scan on cup_matches (cost=0.00..10685.00  
rows=166667 width=24)'), (" Filter: ((rating * '3'::double  
precision) > '20'::double precision)",)]
```

### **C.Q4**

```
cost=0.00..10685.00
```

### **C.Q5**

The index is not on the field that the select is done, so it does not expedite the query execution.

PyCharm File Edit View Navigate Code Refactor Run Tools VCS Window Help

PostgreSQL-Analysis [~/Desktop/DB/PostgreSQL-Analysis] - .../part\_C.py [PostgreSQL-Analysis]

Project PostgreSQL-Analysis ~/Desktop/DB/PostgreSQL-Analysis

- part\_A.py
- part\_B.py
- part\_C.py
- part\_D.py
- part\_E.py
- part\_F.py
- ppDB.py
- External Libraries

```
1 #!/usr/bin/python
2
3 import psycopg2
4
5 conn = psycopg2.connect(database="db2", user="postgres", password="123", host="127.0.0.1", port="5432")
6 print("Opened database successfully")
7 cur = conn.cursor()
8
9 ## Uncomment after first run
10 # cur.execute('DROP INDEX ratingIdx;')
11 # conn.commit()
12
13 print("C.01")
14 cur.execute('EXPLAIN SELECT * from cup_matches WHERE rating*3 > 20;')
15 print(cur.fetchall())
16 conn.commit()
17
18
19
20 print("\nC.02")
21 print("cost=0.00..58.20") ## TODO: change
22
23
24 print("\nC.03")
25 cur.execute('CREATE INDEX ratingIdx ON cup_matches (rating);')
26 conn.commit()
27 print("After posIndex")
28 cur.execute('EXPLAIN SELECT * from cup_matches WHERE rating*3 > 20;')
29 print(cur.fetchall())
30 conn.commit()
31
32 print("\nC.04")
33 print("cost=0.00..58.20") ## TODO: change
34
35 print("\nC.05")
36 print("") ## TODO: Explain => Same!!
```

Run part\_C

/Users/ammkayid/anaconda3/bin/python /Users/ammkayid/Desktop/DB/PostgreSQL-Analysis/part\_C.py

Opened database successfully

C.01

[[('Seq Scan on cup\_matches (cost=0.00..10685.00 rows=166667 width=24)'), (' Filter: ((rating \* '3'::double precision) > '20'::double precision)'),]]

C.02

cost=0.00..58.20

C.03

After posIndex

[[('Seq Scan on cup\_matches (cost=0.00..10685.00 rows=166667 width=24)'), (' Filter: ((rating \* '3'::double precision) > '20'::double precision)'),]]

C.04

cost=0.00..58.20

C.05

20:1 LF+ UTF-8+ Git: master +

## **Part D:**

### **D.Q1**

```
[('Hash Join (cost=17365.00..113725592.29
rows=9651556329 width=48)'), (' Hash Cond:
(played_in.year = cup_matches.year)'), (' -> Seq Scan
on played_in (cost=0.00..172546.00 rows=10000000
width=24)'), (' -> Hash (cost=8185.00..8185.00
rows=500000 width=24)'), ('      -> Seq Scan on
cup_matches (cost=0.00..8185.00 rows=500000
width=24)'),)]
```

### **D.Q2**

After cup\_yearIdx

```
[('Hash Join (cost=17365.00..113725592.29
rows=9651556329 width=48)'), (' Hash Cond:
(played_in.year = cup_matches.year)'), (' -> Seq Scan
on played_in (cost=0.00..172546.00 rows=10000000
width=24)'), (' -> Hash (cost=8185.00..8185.00
rows=500000 width=24)'), ('      -> Seq Scan on
cup_matches (cost=0.00..8185.00 rows=500000
width=24)'),)]
```

### **D.Q3**

The index is not on the field that the select is done, so it does not expedite the query execution.

#### **D.Q4**

After playedin\_yearIndx

```
[('Hash Join (cost=17365.00..113725592.29
rows=9651556329 width=48)'), (' Hash Cond:
(played_in.year = cup_matches.year)'), (' -> Seq Scan
on played_in (cost=0.00..172546.00 rows=10000000
width=24)'), (' -> Hash (cost=8185.00..8185.00
rows=500000 width=24)'), (' -> Seq Scan on
cup_matches (cost=0.00..8185.00 rows=500000
width=24)'),)]
```

#### **D.Q5**

The index is not on the field that the select is done, so it does not expedite the query execution.

PyCharm File Edit View Navigate Code Refactor Run Tools VCS Window Help

PostgreSQL-Analysis [~/Desktop/DB/PostgreSQL-Analysis] - .../part\_D.py [PostgreSQL-Analysis]

Project PostgreSQL-Analysis ~/Desktop/DB/PostgreSQL-Analysis

part\_A.py part\_B.py part\_C.py part\_D.py part\_E.py part\_F.py ppDB.py

External Libraries

```
1 #!/usr/bin/python
2
3 import psycopg2
4
5 conn = psycopg2.connect(database="db2", user="postgres", password="123", host="127.0.0.1", port="5432")
6 print("Opened database successfully")
7 cur = conn.cursor()
8
9 # ## Uncomment after first run
10 # cur.execute('DROP INDEX cup_yearIndex; DROP INDEX playedin_yearIndex;')
11 # conn.commit()
12
13 print("\nD.Q1")
14 cur.execute('EXPLAIN SELECT * from cup_matches, played_in WHERE cup_matches.year=played_in.year;')
15 print(cur.fetchall())
16 conn.commit()
17
18
19 print("\nD.Q2")
20 cur.execute('CREATE INDEX cup_yearIndex ON cup_matches (year);')
21 conn.commit()
22 print("After cup_yearIndex")
23 cur.execute('EXPLAIN SELECT * from cup_matches, played_in WHERE cup_matches.year=played_in.year;')
24 print(cur.fetchall())
25 conn.commit()
26
27 print("\nD.Q3")
28 print("") ## TODO: Explain => Different
29
30
31 print("\nD.Q4")
32 cur.execute('CREATE INDEX playedin_yearIndex ON played_in (year);')
33 conn.commit()
34 print("After playedin_yearIndex")
35 cur.execute('EXPLAIN SELECT * from cup_matches, played_in WHERE cup_matches.year=played_in.year;')
36 print(cur.fetchall())
37 conn.commit()
38
```

Run part\_D

/Users/amrmkayid/anaconda3/bin/python /Users/amrmkayid/Desktop/DB/PostgreSQL-Analysis/part\_D.py

Opened database successfully

D.Q1

[[('Hash Join (cost=17365.00..113725592.29 rows=9651556329 width=48)'), (' Hash Cond: (played\_in.year = cup\_matches.year)'), (' -> Seq Scan on played\_in (cost=0.00..172546.00 rows=10000000 width=24)'), (' -> Hash (cost=17365.00..113725592.29 rows=9651556329 width=48)')]]

D.Q2

After cup\_yearIndex

[[('Hash Join (cost=17365.00..113725592.29 rows=9651556329 width=48)'), (' Hash Cond: (played\_in.year = cup\_matches.year)'), (' -> Seq Scan on played\_in (cost=0.00..172546.00 rows=10000000 width=24)'), (' -> Hash (cost=17365.00..113725592.29 rows=9651556329 width=48)')]]

D.Q3

D.Q4

After playedin\_yearIndex

[[('Hash Join (cost=17365.00..113725592.29 rows=9651556329 width=48)'), (' Hash Cond: (played\_in.year = cup\_matches.year)'), (' -> Seq Scan on played\_in (cost=0.00..172546.00 rows=10000000 width=24)'), (' -> Hash (cost=17365.00..113725592.29 rows=9651556329 width=48)')]]

D.Q5

21:1 LF UTF-8 Git: master

## **Part E:**

### **E.Q1**

```
[('Hash Join (cost=17365.00..436278.82 rows=10000000 width=48)'), (' Hash Cond: (played_in.mid = cup_matches.mid)'), (' -> Seq Scan on played_in (cost=0.00..172546.00 rows=10000000 width=24)'), (' -> Hash (cost=8185.00..8185.00 rows=500000 width=24)'), (' -> Seq Scan on cup_matches (cost=0.00..8185.00 rows=500000 width=24)')]
```

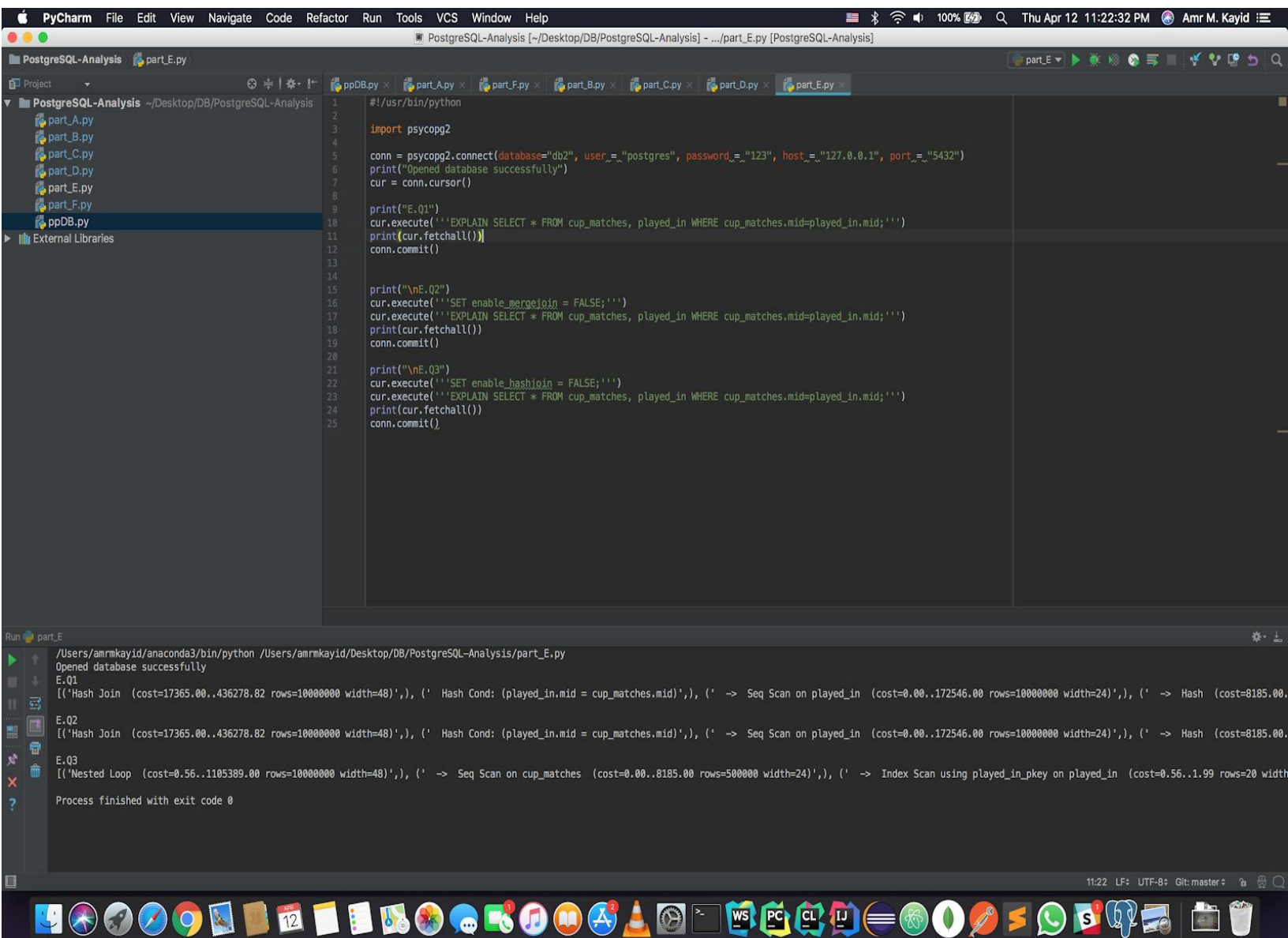
### **E.Q2**

```
[('Hash Join (cost=17365.00..436278.82 rows=10000000 width=48)'), (' Hash Cond: (played_in.mid = cup_matches.mid)'), (' -> Seq Scan on played_in (cost=0.00..172546.00 rows=10000000 width=24)'), (' -> Hash (cost=8185.00..8185.00 rows=500000 width=24)'), (' -> Seq Scan on cup_matches (cost=0.00..8185.00 rows=500000 width=24)')]
```

### **E.Q3**

```
[('Nested Loop (cost=0.56..1105389.00 rows=10000000 width=48)'), (' -> Seq Scan on cup_matches (cost=0.00..8185.00 rows=500000 width=24)'), (' -> Index Scan using played_in_pkey on played_in
```

(cost=0.56..1.99 rows=20 width=24)'), (' Index Cond:  
(mid = cup\_matches.mid)'),)]



The screenshot shows the PyCharm IDE interface. The top toolbar includes buttons for Run, Debug, and other development tools. The main editor window displays a Python script named `part_E.py` with the following content:

```
1 #!/usr/bin/python
2
3 import psycopg2
4
5 conn = psycopg2.connect(database="db2", user="postgres", password="123", host="127.0.0.1", port="5432")
6 print("Opened database successfully")
7 cur = conn.cursor()
8
9 print("\nE.Q1")
10 cur.execute('EXPLAIN SELECT * FROM cup_matches, played_in WHERE cup_matches.mid=played_in.mid;')
11 print(cur.fetchall())
12 conn.commit()
13
14 print("\nE.Q2")
15 cur.execute('SET enable_mergejoin = FALSE;')
16 cur.execute('EXPLAIN SELECT * FROM cup_matches, played_in WHERE cup_matches.mid=played_in.mid;')
17 print(cur.fetchall())
18 conn.commit()
19
20 print("\nE.Q3")
21 cur.execute('SET enable_hashjoin = FALSE;')
22 cur.execute('EXPLAIN SELECT * FROM cup_matches, played_in WHERE cup_matches.mid=played_in.mid;')
23 print(cur.fetchall())
24 conn.commit()
25
```

The bottom panel shows the output of the script execution:

```
Run part_E
/Users/amrmkayid/anaconda3/bin/python /Users/amrmkayid/Desktop/DB/PostgreSQL-Analysis/part_E.py
Opened database successfully
E.Q1
[('Hash Join (cost=17365.00..436278.82 rows=10000000 width=48)'), (' Hash Cond: (played_in.mid = cup_matches.mid)'), (' -> Seq Scan on played_in (cost=0.00..172546.00 rows=10000000 width=24)'), (' -> Hash (cost=8185.00.
E.Q2
[('Hash Join (cost=17365.00..436278.82 rows=10000000 width=48)'), (' Hash Cond: (played_in.mid = cup_matches.mid)'), (' -> Seq Scan on played_in (cost=0.00..172546.00 rows=10000000 width=24)'), (' -> Hash (cost=8185.00.
E.Q3
[('Nested Loop (cost=0.56..1185389.00 rows=10000000 width=48)'), (' -> Seq Scan on cup_matches (cost=0.00..8185.00 rows=500000 width=24)'), (' -> Index Scan using played_in_pkey on played_in (cost=0.56..1.99 rows=20 width=
Process finished with exit code 0
```

The bottom status bar shows the current file is `part_E.py`, the encoding is `UTF-8`, and the Git status is `master`.