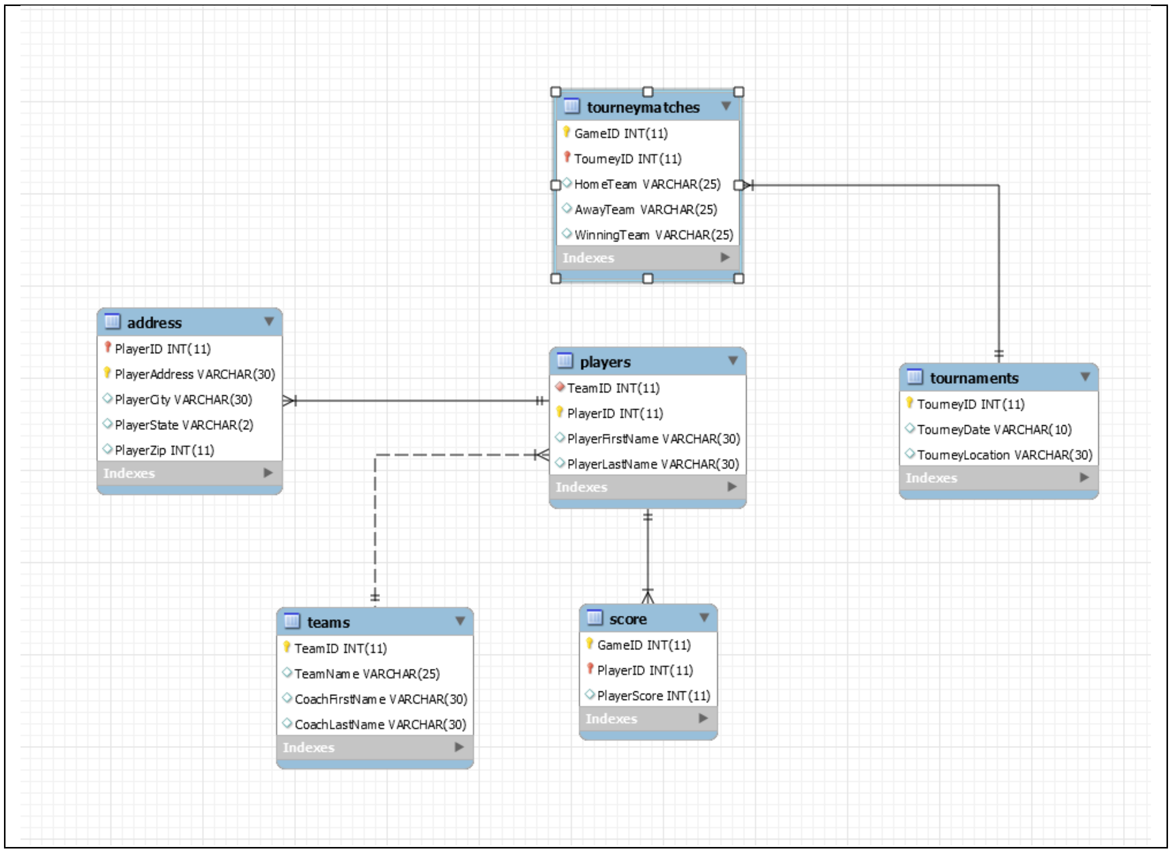
MSDS 7330
File Organization and Database Management

Midterm

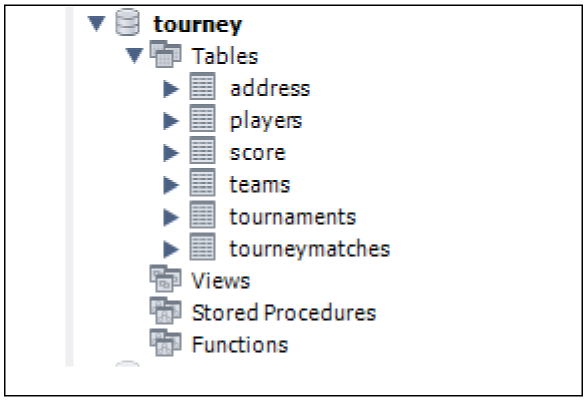
Evangelos Giakoumakis

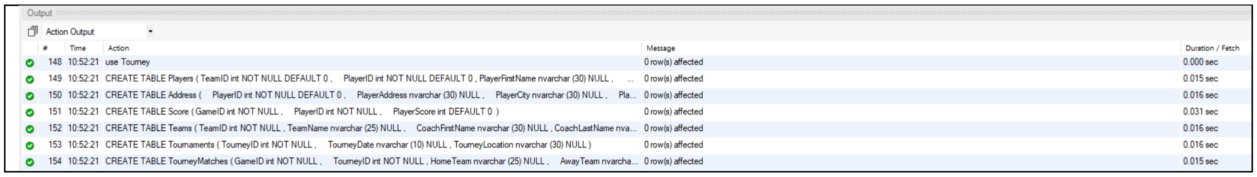
1. Develop a normalized schema for the database using MySQL Workbench

Normalization level was not specified, so did a first normal form (1NF) splitting addresses from players information.

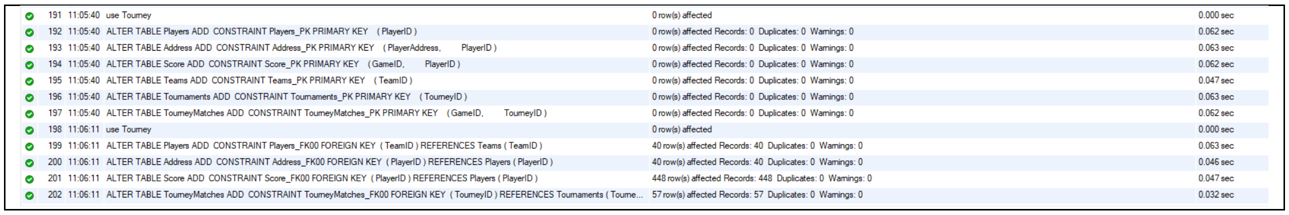


2. Create the tourney DB using the model you just created (forward engineering)





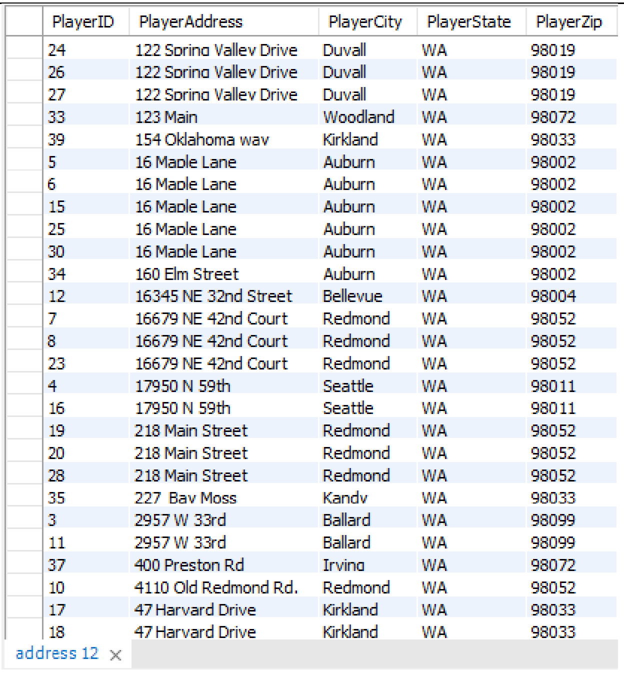
```
Output
Action Output
 #    Time      Action                                                                                                               Message             Duration / Fetch
 148  10:52:21  use Tourney                                                                                                          0 row(s) affected   0.000 sec
 149  10:52:21  CREATE TABLE Players ( TeamID int NOT NULL DEFAULT 0 ,   PlayerID int NOT NULL DEFAULT 0 , PlayerFirstName nvarchar (30) NULL ,   ...   0 row(s) affected   0.015 sec
 150  10:52:21  CREATE TABLE Address (   PlayerID int NOT NULL DEFAULT 0 ,   PlayerAddress nvarchar (30) NULL ,   PlayerCity nvarchar (30) NULL ,   Pla...   0 row(s) affected   0.016 sec
 151  10:52:21  CREATE TABLE Score ( GameID int NOT NULL ,   PlayerID int NOT NULL ,   PlayerScore int DEFAULT 0 )                 0 row(s) affected   0.031 sec
 152  10:52:21  CREATE TABLE Teams ( TeamID int NOT NULL , TeamName nvarchar (25) NULL ,   CoachFirstName nvarchar (30) NULL , CoachLastName nva...   0 row(s) affected   0.016 sec
 153  10:52:21  CREATE TABLE Tournaments ( TourneyID int NOT NULL ,   TourneyDate nvarchar (10) NULL , TourneyLocation nvarchar (30) NULL )   0 row(s) affected   0.016 sec
 154  10:52:21  CREATE TABLE TourneyMatches ( GameID int NOT NULL ,   TourneyID int NOT NULL , HomeTeam nvarchar (25) NULL ,   AwayTeam nvarcha...   0 row(s) affected   0.015 sec
```



```
 191  11:05:40  use Tourney                                                                                     0 row(s) affected                                          0.000 sec
 192  11:05:40  ALTER TABLE Players ADD  CONSTRAINT Players_PK PRIMARY KEY   ( PlayerID )                        0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0   0.062 sec
 193  11:05:40  ALTER TABLE Address ADD  CONSTRAINT Address_PK PRIMARY KEY   ( PlayerAddress,     PlayerID )   0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0   0.063 sec
 194  11:05:40  ALTER TABLE Score ADD  CONSTRAINT Score_PK PRIMARY KEY   ( GameID,     PlayerID )               0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0   0.062 sec
 195  11:05:40  ALTER TABLE Teams ADD  CONSTRAINT Teams_PK PRIMARY KEY   ( TeamID )                            0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0   0.047 sec
 196  11:05:40  ALTER TABLE Tournaments ADD  CONSTRAINT Tournaments_PK PRIMARY KEY   ( TourneyID )             0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0   0.063 sec
 197  11:05:40  ALTER TABLE TourneyMatches ADD  CONSTRAINT TourneyMatches_PK PRIMARY KEY   ( GameID,     TourneyID )   0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0   0.062 sec
 198  11:06:11  use Tourney                                                                                     0 row(s) affected                                          0.000 sec
 199  11:06:11  ALTER TABLE Players ADD  CONSTRAINT Players_FK00 FOREIGN KEY  ( TeamID ) REFERENCES Teams ( TeamID )   40 row(s) affected Records: 40  Duplicates: 0  Warnings: 0   0.063 sec
 200  11:06:11  ALTER TABLE Address ADD  CONSTRAINT Address_FK00 FOREIGN KEY  ( PlayerID ) REFERENCES Players ( PlayerID )   40 row(s) affected Records: 40  Duplicates: 0  Warnings: 0   0.046 sec
 201  11:06:11  ALTER TABLE Score ADD  CONSTRAINT Score_FK00 FOREIGN KEY  ( PlayerID ) REFERENCES Players ( PlayerID )   448 row(s) affected Records: 448  Duplicates: 0  Warnings: 0   0.047 sec
 202  11:06:11  ALTER TABLE TourneyMatches ADD  CONSTRAINT TourneyMatches_FK00 FOREIGN KEY  ( TourneyID ) REFERENCES Tournaments ( Tourne...   57 row(s) affected Records: 57  Duplicates: 0  Warnings: 0   0.032 sec
```

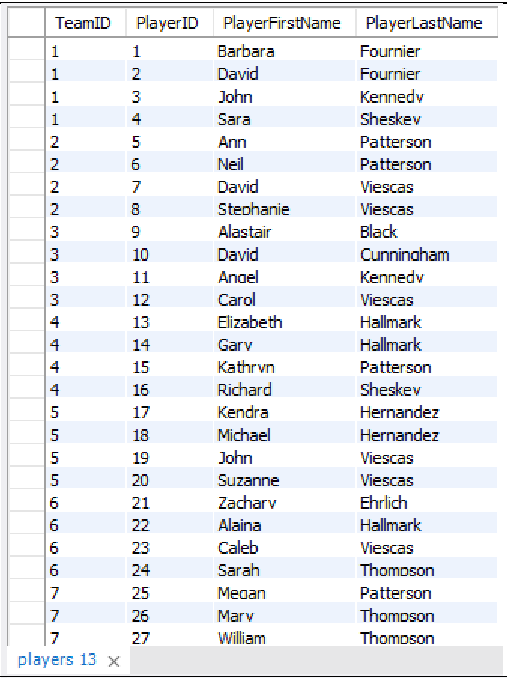
3. Insert values into the Database from the csv files provided.

Inserted data using provided .csv files and mysql data import wizard.
Below are all the elements inside all Database tables.



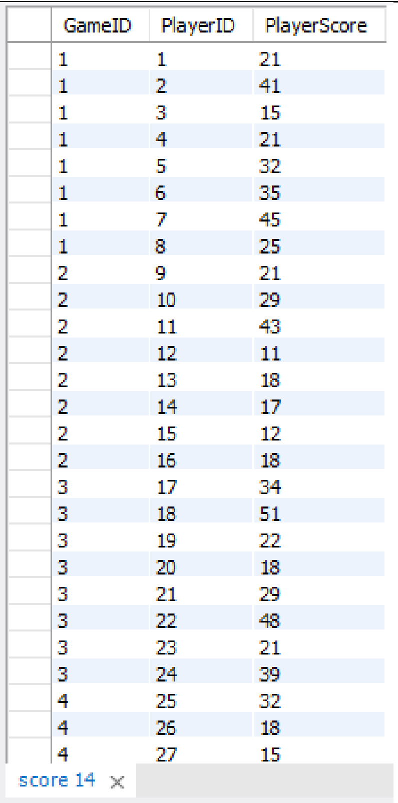
PlayerID	PlayerAddress	PlayerCity	PlayerState	PlayerZip
24	122 Spring Valley Drive	Duvall	WA	98019
26	122 Spring Valley Drive	Duvall	WA	98019
27	122 Spring Valley Drive	Duvall	WA	98019
33	123 Main	Woodland	WA	98072
39	154 Oklahoma way	Kirkland	WA	98033
5	16 Maple Lane	Auburn	WA	98002
6	16 Maple Lane	Auburn	WA	98002
15	16 Maple Lane	Auburn	WA	98002
25	16 Maple Lane	Auburn	WA	98002
30	16 Maple Lane	Auburn	WA	98002
34	160 Elm Street	Auburn	WA	98002
12	16345 NE 32nd Street	Bellevue	WA	98004
7	16679 NE 42nd Court	Redmond	WA	98052
8	16679 NE 42nd Court	Redmond	WA	98052
23	16679 NE 42nd Court	Redmond	WA	98052
4	17950 N 59th	Seattle	WA	98011
16	17950 N 59th	Seattle	WA	98011
19	218 Main Street	Redmond	WA	98052
20	218 Main Street	Redmond	WA	98052
28	218 Main Street	Redmond	WA	98052
35	227 Bay Moss	Kandy	WA	98033
3	2957 W 33rd	Ballard	WA	98099
11	2957 W 33rd	Ballard	WA	98099
37	400 Preston Rd	Irving	WA	98072
10	4110 Old Redmond Rd.	Redmond	WA	98052
17	47 Harvard Drive	Kirkland	WA	98033
18	47 Harvard Drive	Kirkland	WA	98033

address 12 ×



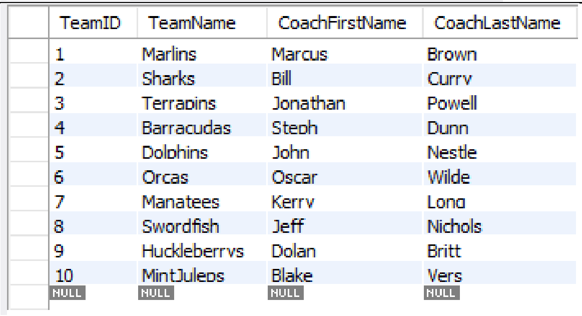
TeamID	PlayerID	PlayerFirstName	PlayerLastName
1	1	Barbara	Fournier
1	2	David	Fournier
1	3	John	Kennedy
1	4	Sara	Sheskey
2	5	Ann	Patterson
2	6	Neil	Patterson
2	7	David	Viescas
2	8	Stephanie	Viescas
3	9	Alastair	Black
3	10	David	Cunningham
3	11	Angel	Kennedy
3	12	Carol	Viescas
4	13	Elizabeth	Hallmark
4	14	Gary	Hallmark
4	15	Kathryn	Patterson
4	16	Richard	Sheskey
5	17	Kendra	Hernandez
5	18	Michael	Hernandez
5	19	John	Viescas
5	20	Suzanne	Viescas
6	21	Zachary	Ehrlich
6	22	Alaina	Hallmark
6	23	Caleb	Viescas
6	24	Sarah	Thompson
7	25	Megan	Patterson
7	26	Mary	Thompson
7	27	William	Thompson

players 13 ×

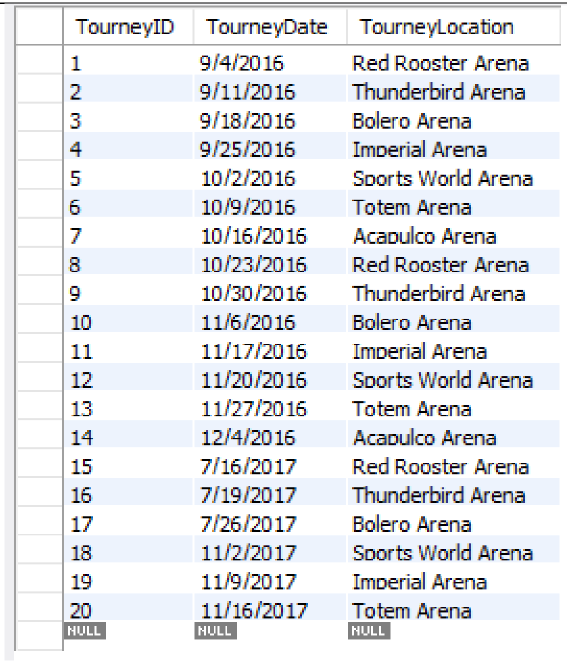


GameID	PlayerID	PlayerScore
1	1	21
1	2	41
1	3	15
1	4	21
1	5	32
1	6	35
1	7	45
1	8	25
2	9	21
2	10	29
2	11	43
2	12	11
2	13	18
2	14	17
2	15	12
2	16	18
3	17	34
3	18	51
3	19	22
3	20	18
3	21	29
3	22	48
3	23	21
3	24	39
4	25	32
4	26	18
4	27	15

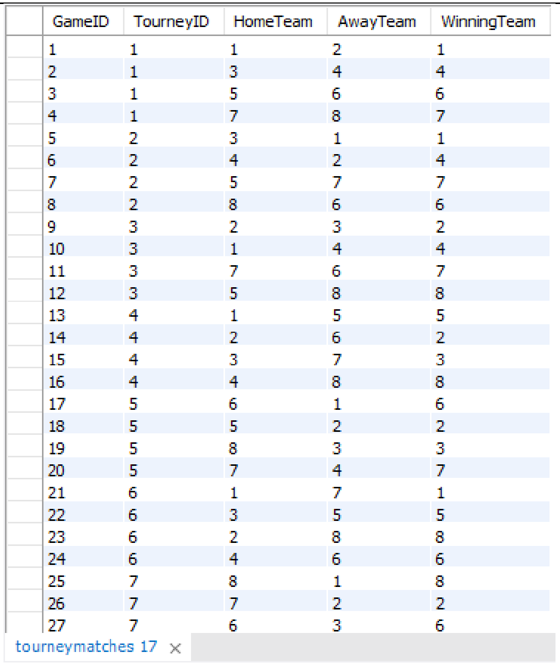
score 14 ×



TeamID	TeamName	CoachFirstName	CoachLastName
1	Marlins	Marcus	Brown
2	Sharks	Bill	Curry
3	Terrapins	Jonathan	Powell
4	Barracudas	Steph	Dunn
5	Dolphins	John	Nestle
6	Orcas	Oscar	Wilde
7	Manatees	Kerry	Long
8	Swordfish	Jeff	Nichols
9	Huckleberrys	Dolan	Britt
10	MintJuleps	Blake	Vers
NULL	NULL	NULL	NULL



TourneyID	TourneyDate	TourneyLocation
1	9/4/2016	Red Rooster Arena
2	9/11/2016	Thunderbird Arena
3	9/18/2016	Bolero Arena
4	9/25/2016	Imperial Arena
5	10/2/2016	Sports World Arena
6	10/9/2016	Totem Arena
7	10/16/2016	Acapulco Arena
8	10/23/2016	Red Rooster Arena
9	10/30/2016	Thunderbird Arena
10	11/6/2016	Bolero Arena
11	11/17/2016	Imperial Arena
12	11/20/2016	Sports World Arena
13	11/27/2016	Totem Arena
14	12/4/2016	Acapulco Arena
15	7/16/2017	Red Rooster Arena
16	7/19/2017	Thunderbird Arena
17	7/26/2017	Bolero Arena
18	11/2/2017	Sports World Arena
19	11/9/2017	Imperial Arena
20	11/16/2017	Totem Arena
NULL	NULL	NULL

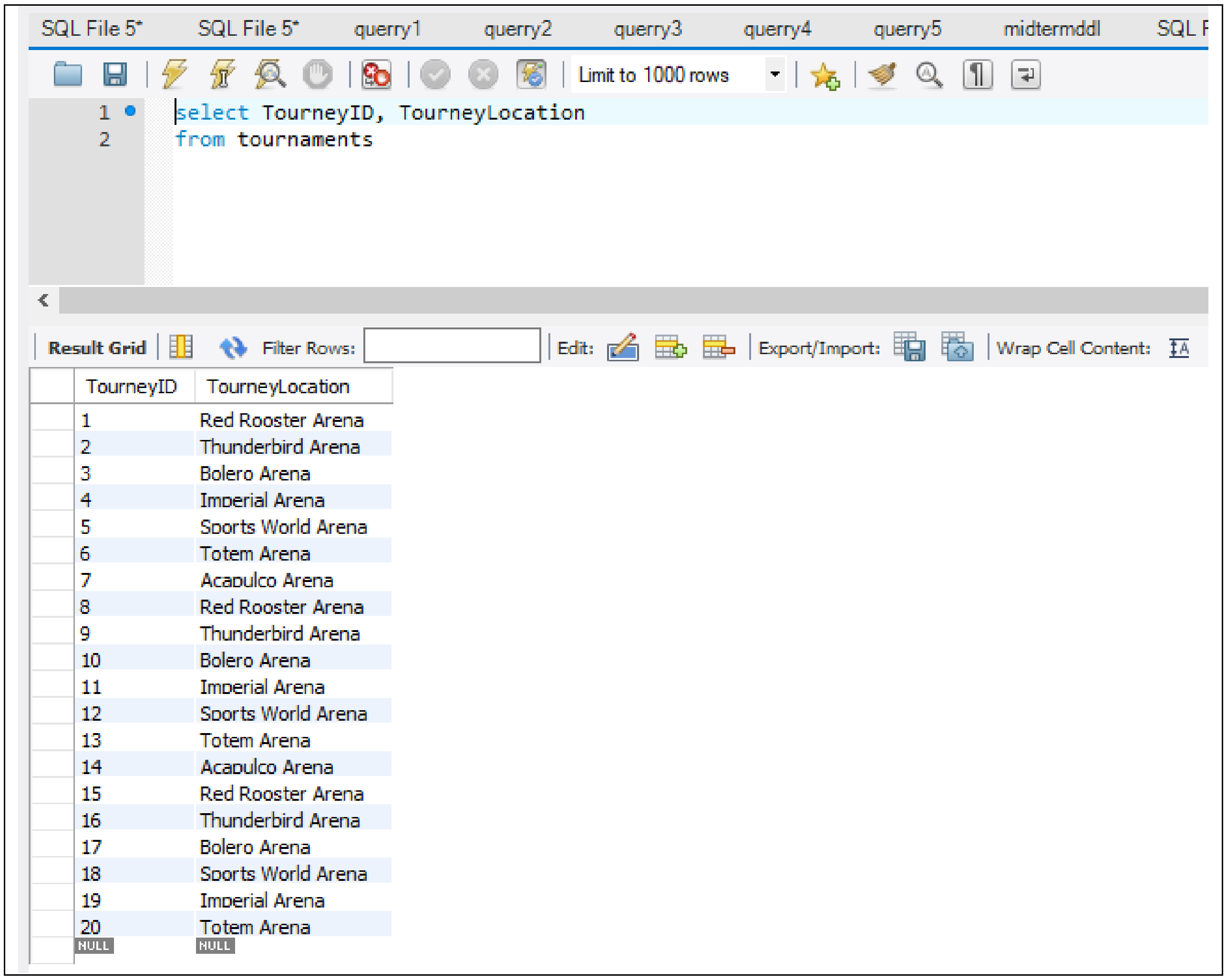


GameID	TourneyID	HomeTeam	AwayTeam	WinningTeam
1	1	1	2	1
2	1	3	4	4
3	1	5	6	6
4	1	7	8	7
5	2	3	1	1
6	2	4	2	4
7	2	5	7	7
8	2	8	6	6
9	3	2	3	2
10	3	1	4	4
11	3	7	6	7
12	3	5	8	8
13	4	1	5	5
14	4	2	6	2
15	4	3	7	3
16	4	4	8	8
17	5	6	1	6
18	5	5	2	2
19	5	8	3	3
20	5	7	4	7
21	6	1	7	1
22	6	3	5	5
23	6	2	8	8
24	6	4	6	6
25	7	8	1	8
26	7	7	2	2
27	7	6	3	6

tourneymatches 17 ×

4. Perform following queries:

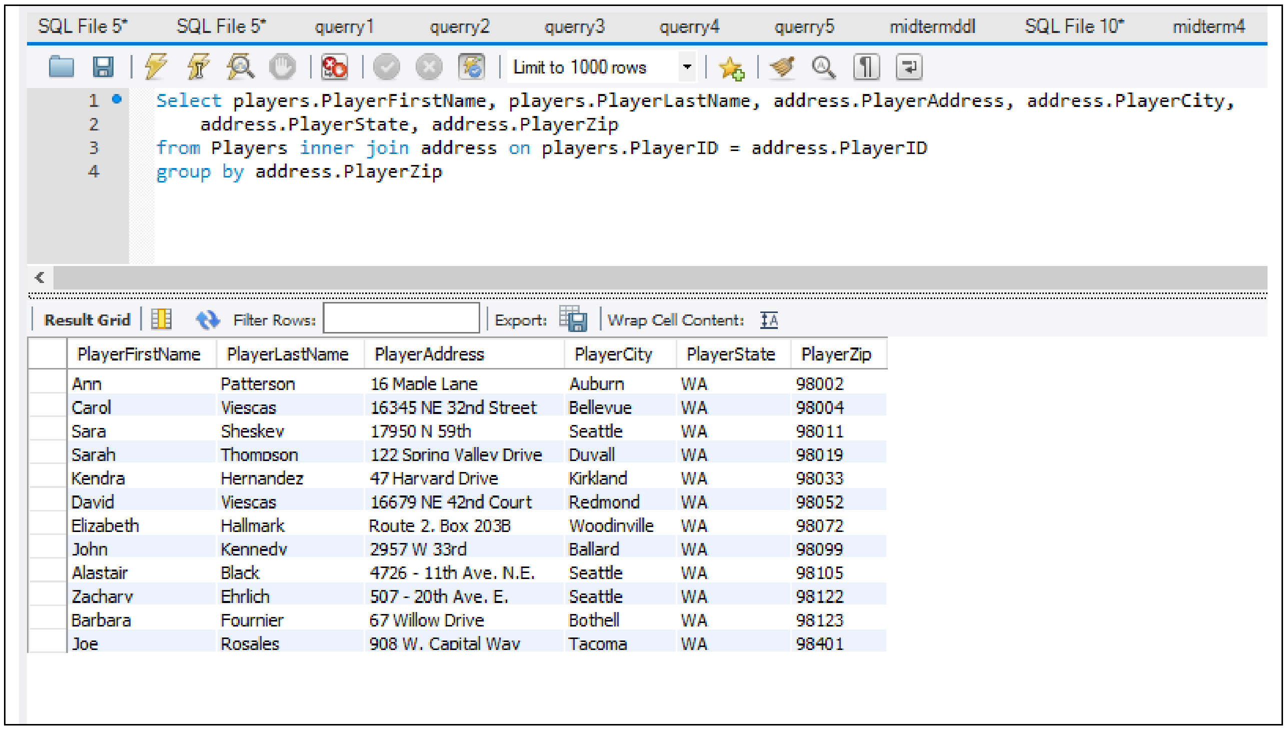
a) Provide locations where association is holding tournaments



```
select TourneyID, TourneyLocation
from tournaments
```

TourneyID	TourneyLocation
1	Red Rooster Arena
2	Thunderbird Arena
3	Bolero Arena
4	Imperial Arena
5	Sports World Arena
6	Totem Arena
7	Acapulco Arena
8	Red Rooster Arena
9	Thunderbird Arena
10	Bolero Arena
11	Imperial Arena
12	Sports World Arena
13	Totem Arena
14	Acapulco Arena
15	Red Rooster Arena
16	Thunderbird Arena
17	Bolero Arena
18	Sports World Arena
19	Imperial Arena
20	Totem Arena
NULL	NULL

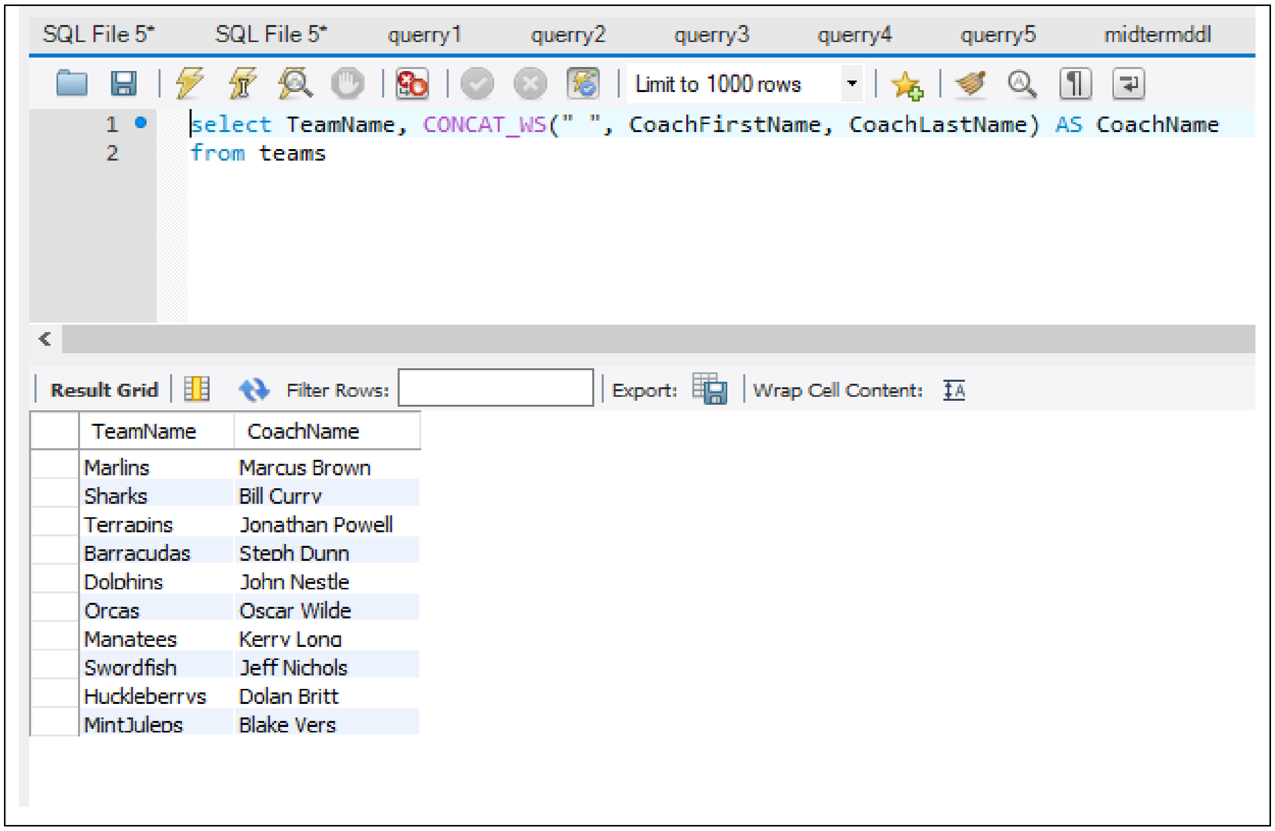
b) Display all players and their address formatted suitably for a mailing list, sorted by zip code.



```
Select players.PlayerFirstName, players.PlayerLastName, address.PlayerAddress, address.PlayerCity,
    address.PlayerState, address.PlayerZip
from Players inner join address on players.PlayerID = address.PlayerID
group by address.PlayerZip
```

PlayerFirstName	PlayerLastName	PlayerAddress	PlayerCity	PlayerState	PlayerZip
Ann	Patterson	16 Maple Lane	Auburn	WA	98002
Carol	Viescas	16345 NE 32nd Street	Bellevue	WA	98004
Sara	Sheskey	17950 N 59th	Seattle	WA	98011
Sarah	Thompson	122 Spring Valley Drive	Duvall	WA	98019
Kendra	Hernandez	47 Harvard Drive	Kirkland	WA	98033
David	Viescas	16679 NE 42nd Court	Redmond	WA	98052
Elizabeth	Hallmark	Route 2, Box 203B	Woodinville	WA	98072
John	Kennedy	2957 W 33rd	Ballard	WA	98099
Alastair	Black	4726 - 11th Ave. N.E.	Seattle	WA	98105
Zachary	Ehrlich	507 - 20th Ave. E.	Seattle	WA	98122
Barbara	Fournier	67 Willow Drive	Bothell	WA	98123
Joe	Rosales	908 W. Capital Way	Tacoma	WA	98401

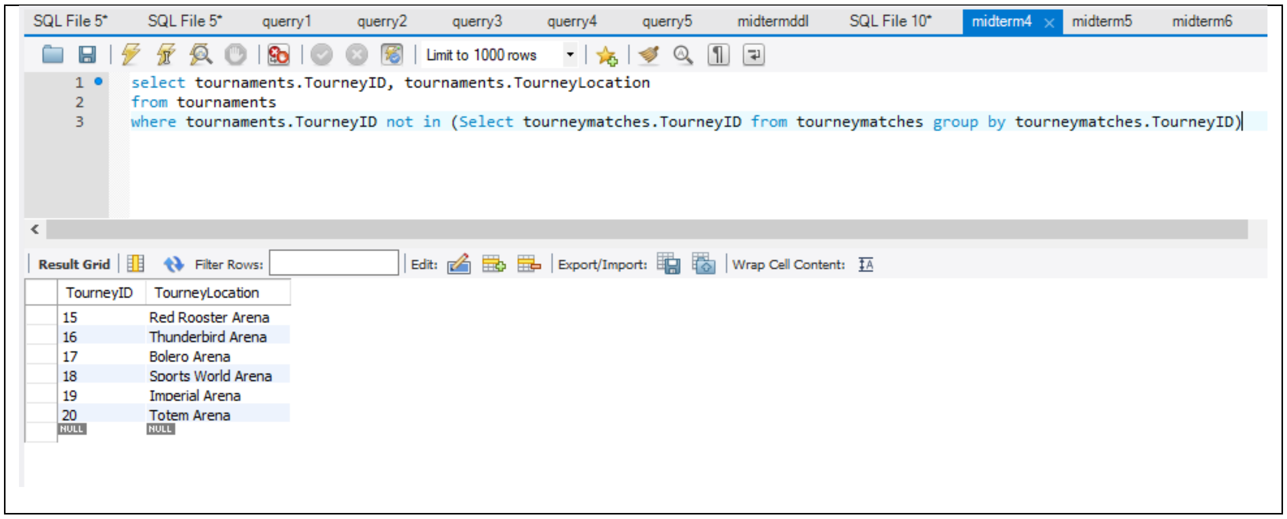
c) Display teams and name of their head coach.



```sql
select TeamName, CONCAT_WS(" ", CoachFirstName, CoachLastName) AS CoachName
from teams
```

TeamName	CoachName
Marlins	Marcus Brown
Sharks	Bill Curry
Terrapins	Jonathan Powell
Barracudas	Steph Dunn
Dolphins	John Nestle
Orcas	Oscar Wilde
Manatees	Kerry Long
Swordfish	Jeff Nichols
Huckleberrys	Dolan Britt
MintJuleps	Blake Vers

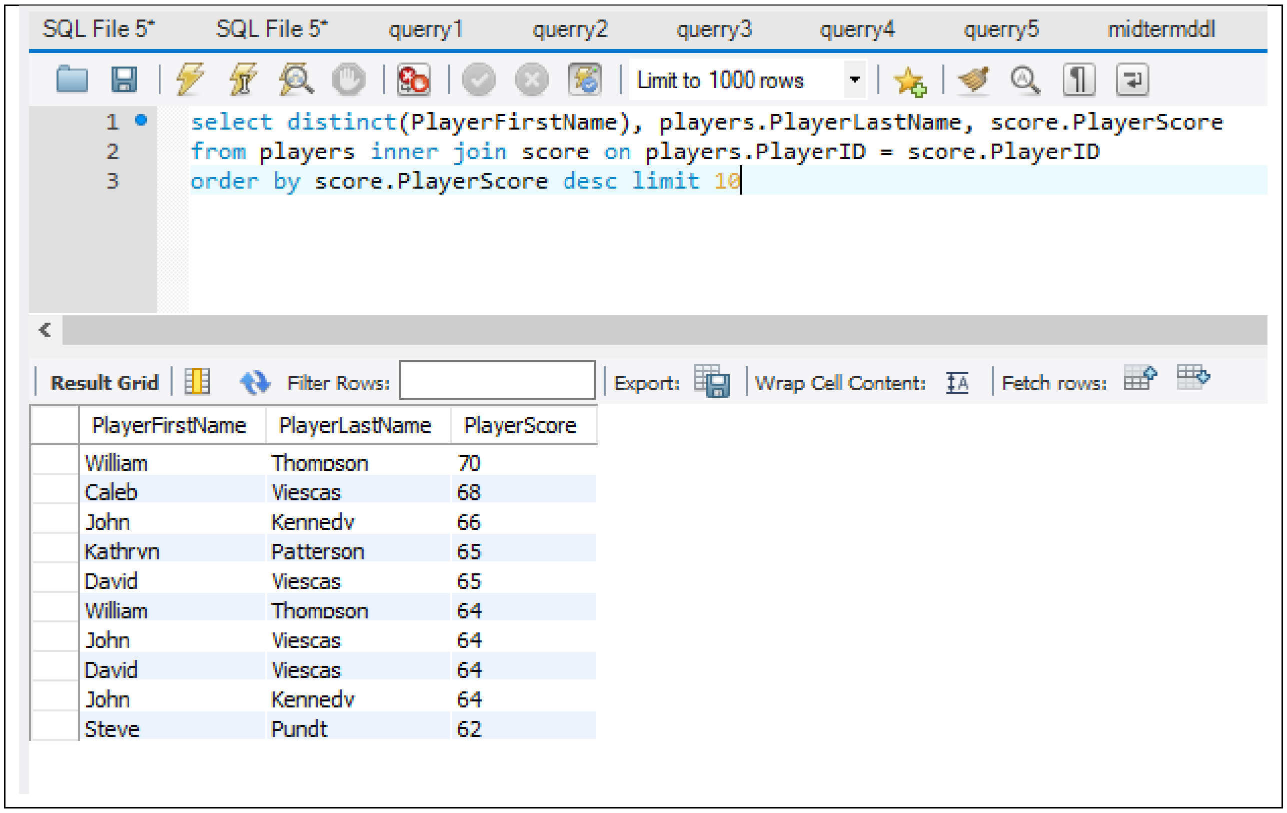
d) Show tournaments that have not been played yet.



```sql
select tournaments.TourneyID, tournaments.TourneyLocation
from tournaments
where tournaments.TourneyID not in (Select tourneymatches.TourneyID from tourneymatches group by tourneymatches.TourneyID)
```

TourneyID	TourneyLocation
15	Red Rooster Arena
16	Thunderbird Arena
17	Bolero Arena
18	Sports World Arena
19	Imperial Arena
20	Totem Arena
NULL	NULL

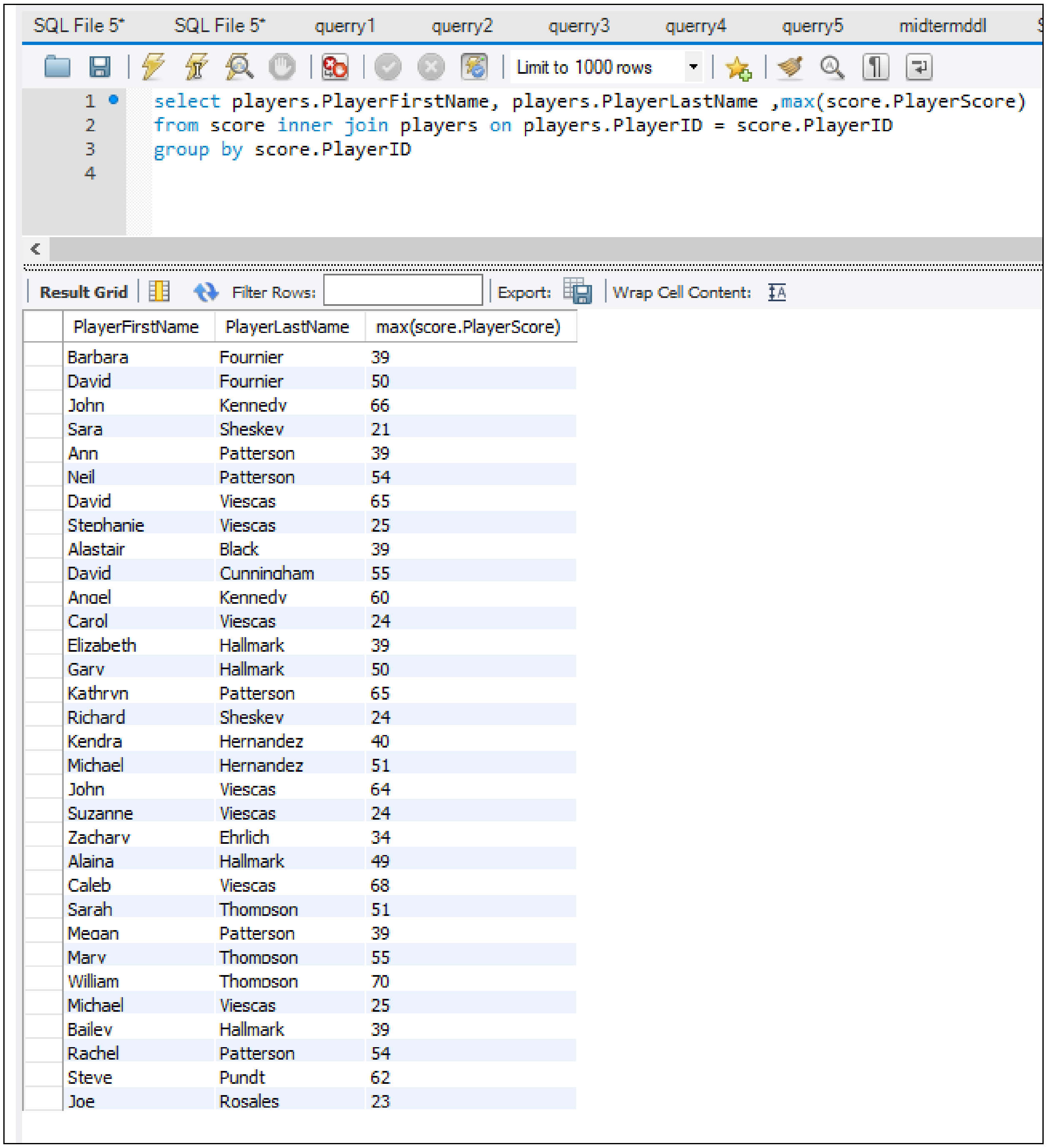
e) Display name of top 10 scorers (Players who scored highest) along with their score.



```
select distinct(PlayerFirstName), players.PlayerLastName, score.PlayerScore
from players inner join score on players.PlayerID = score.PlayerID
order by score.PlayerScore desc limit 10
```

PlayerFirstName	PlayerLastName	PlayerScore
William	Thompson	70
Caleb	Viescas	68
John	Kennedy	66
Kathryn	Patterson	65
David	Viescas	65
William	Thompson	64
John	Viescas	64
David	Viescas	64
John	Kennedy	64
Steve	Pundt	62

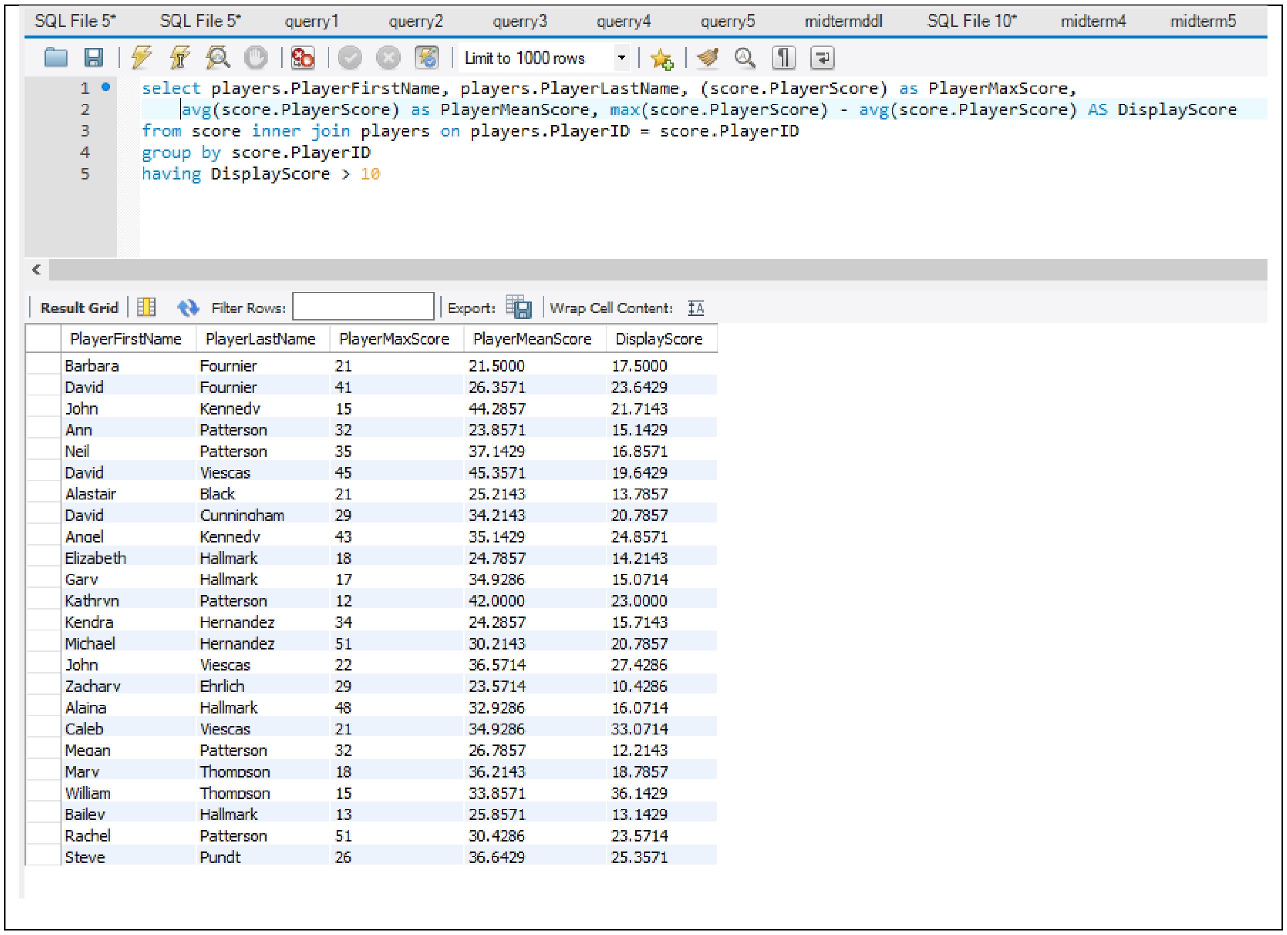
f) Display players' names along with their highest score.



```
select players.PlayerFirstName, players.PlayerLastName ,max(score.PlayerScore)
from score inner join players on players.PlayerID = score.PlayerID
group by score.PlayerID
```

PlayerFirstName	PlayerLastName	max(score.PlayerScore)
Barbara	Fournier	39
David	Fournier	50
John	Kennedy	66
Sara	Sheskey	21
Ann	Patterson	39
Neil	Patterson	54
David	Viescas	65
Stephanie	Viescas	25
Alastair	Black	39
David	Cunningham	55
Angel	Kennedy	60
Carol	Viescas	24
Elizabeth	Hallmark	39
Gary	Hallmark	50
Kathryn	Patterson	65
Richard	Sheskey	24
Kendra	Hernandez	40
Michael	Hernandez	51
John	Viescas	64
Suzanne	Viescas	24
Zachary	Ehrlich	34
Alaina	Hallmark	49
Caleb	Viescas	68
Sarah	Thompson	51
Megan	Patterson	39
Mary	Thompson	55
William	Thompson	70
Michael	Viescas	25
Bailey	Hallmark	39
Rachel	Patterson	54
Steve	Pundt	62
Joe	Rosales	23

g) List the player (names) whose highest score in a game is more than 10 points higher than their average.

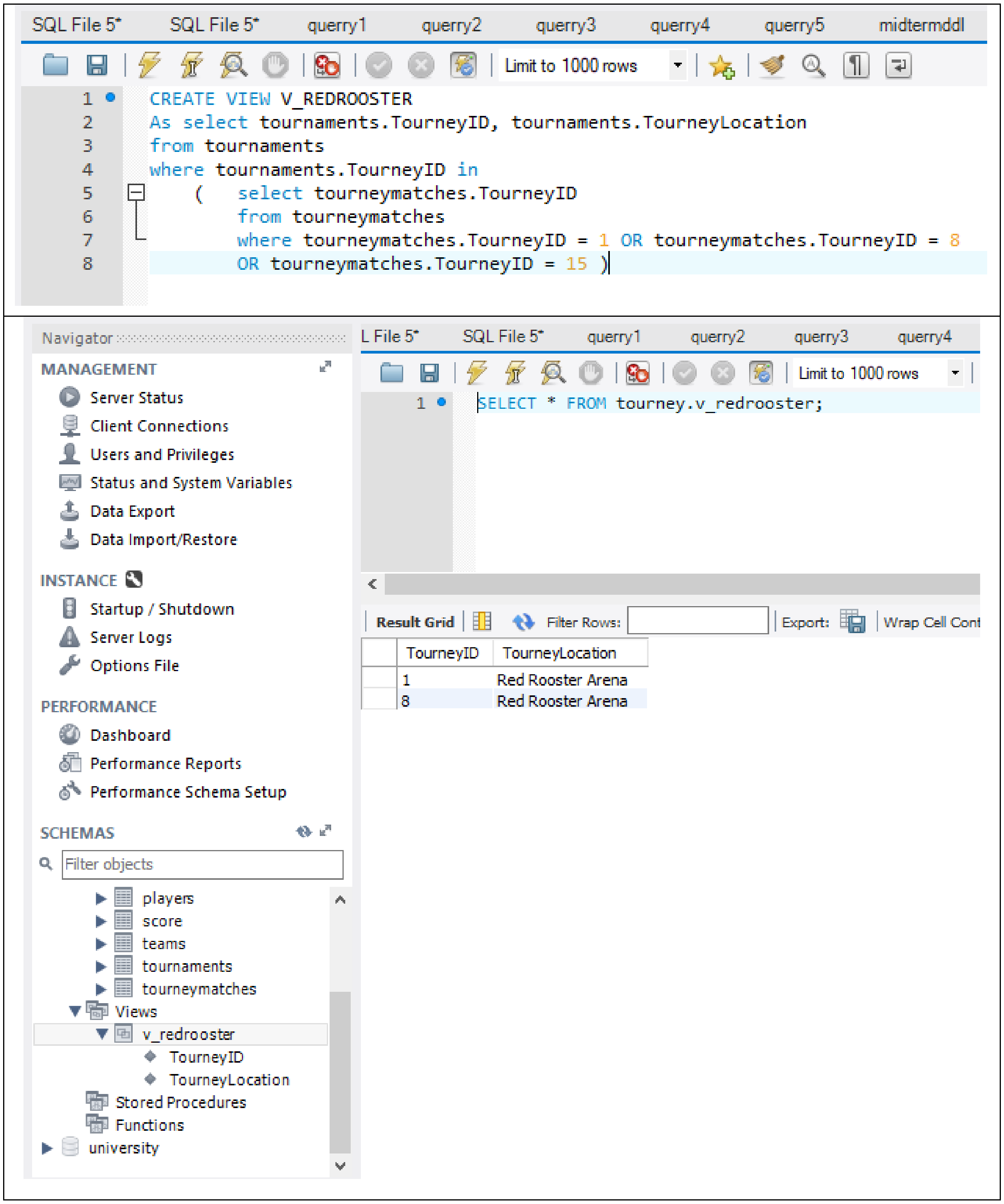


Limit to 1000 rows

```sql
select players.PlayerFirstName, players.PlayerLastName, (score.PlayerScore) as PlayerMaxScore,
    avg(score.PlayerScore) as PlayerMeanScore, max(score.PlayerScore) - avg(score.PlayerScore) AS DisplayScore
from score inner join players on players.PlayerID = score.PlayerID
group by score.PlayerID
having DisplayScore > 10
```

PlayerFirstName	PlayerLastName	PlayerMaxScore	PlayerMeanScore	DisplayScore
Barbara	Fournier	21	21.5000	17.5000
David	Fournier	41	26.3571	23.6429
John	Kennedy	15	44.2857	21.7143
Ann	Patterson	32	23.8571	15.1429
Neil	Patterson	35	37.1429	16.8571
David	Viescas	45	45.3571	19.6429
Alastair	Black	21	25.2143	13.7857
David	Cunningham	29	34.2143	20.7857
Angel	Kennedy	43	35.1429	24.8571
Elizabeth	Hallmark	18	24.7857	14.2143
Gary	Hallmark	17	34.9286	15.0714
Kathryn	Patterson	12	42.0000	23.0000
Kendra	Hernandez	34	24.2857	15.7143
Michael	Hernandez	51	30.2143	20.7857
John	Viescas	22	36.5714	27.4286
Zachary	Ehrlich	29	23.5714	10.4286
Alaina	Hallmark	48	32.9286	16.0714
Caleb	Viescas	21	34.9286	33.0714
Megan	Patterson	32	26.7857	12.2143
Mary	Thompson	18	36.2143	18.7857
William	Thompson	15	33.8571	36.1429
Bailey	Hallmark	13	25.8571	13.1429
Rachel	Patterson	51	30.4286	23.5714
Steve	Pundt	26	36.6429	25.3571
]

5. Create a view of all the tournaments that have been played at Red Rooster.

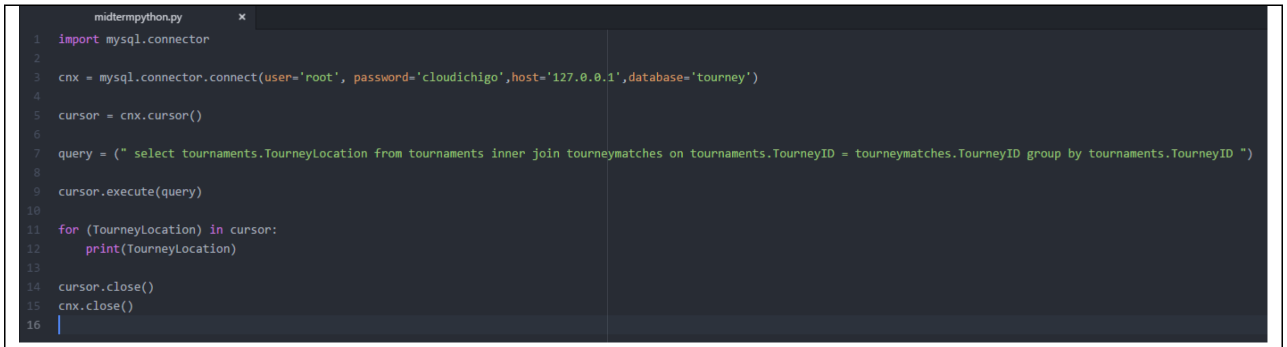


```
CREATE VIEW V_REDROOSTER
As select tournaments.TourneyID, tournaments.TourneyLocation
from tournaments
where tournaments.TourneyID in
    (   select tourneymatches.TourneyID
        from tourneymatches
        where tourneymatches.TourneyID = 1 OR tourneymatches.TourneyID = 8
        OR tourneymatches.TourneyID = 15 )
```

```
SELECT * FROM tourney.v_redrooster;
```

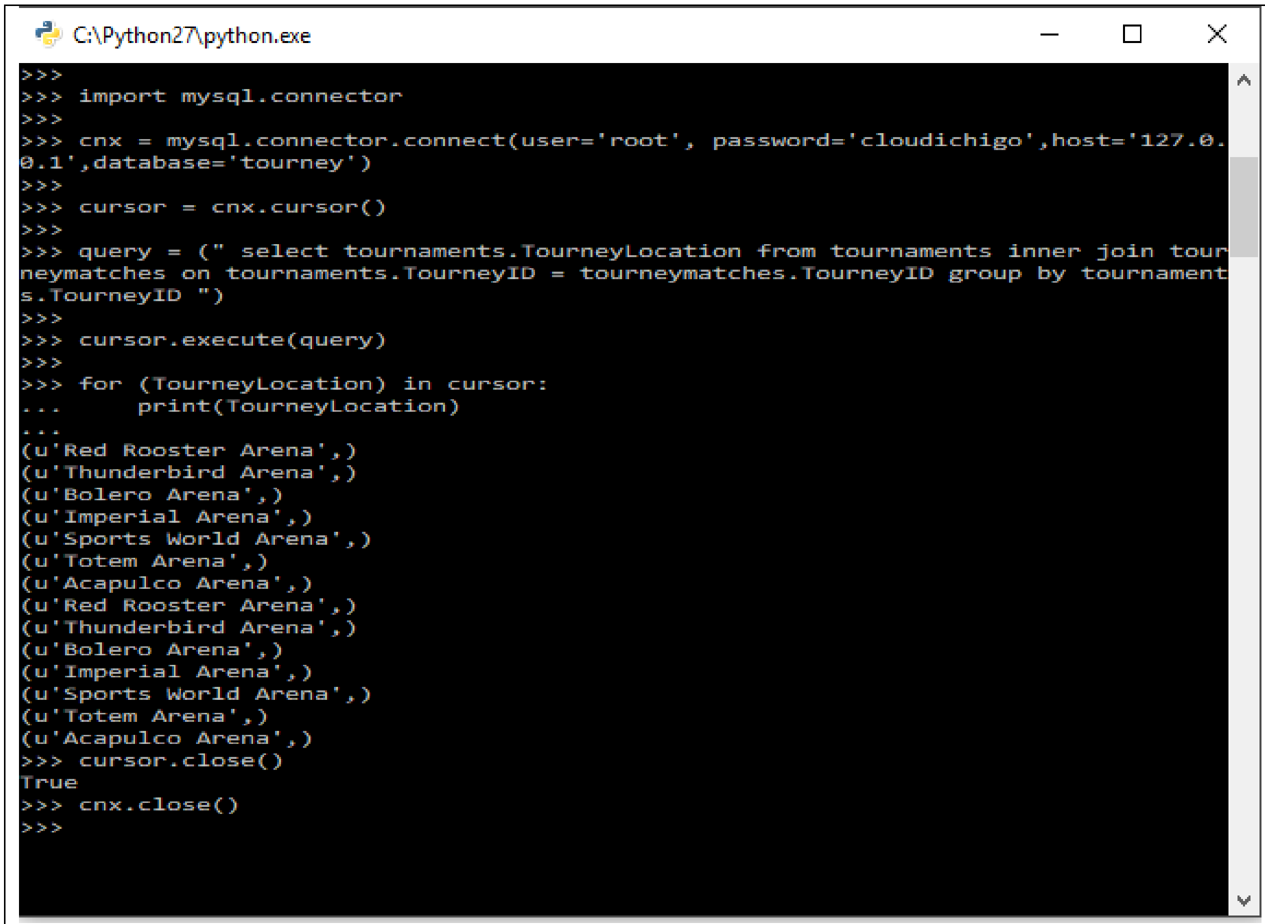
TourneyID	TourneyLocation
1	Red Rooster Arena
8	Red Rooster Arena

6. Connect to this database using Python


midtermpython.py

```python
import mysql.connector

cnx = mysql.connector.connect(user='root', password='cloudichigo',host='127.0.0.1',database='tourney')

cursor = cnx.cursor()

query = (" select tournaments.TourneyLocation from tournaments inner join tourneymatches on tournaments.TourneyID = tourneymatches.TourneyID group by tournaments.TourneyID ")

cursor.execute(query)

for (TourneyLocation) in cursor:
    print(TourneyLocation)

cursor.close()
cnx.close()
```

a) Display name of all the tournaments that were previously held.


C:\Python27\python.exe

```
>>>
>>> import mysql.connector
>>>
>>> cnx = mysql.connector.connect(user='root', password='cloudichigo',host='127.0.
0.1',database='tourney')
>>>
>>> cursor = cnx.cursor()
>>>
>>> query = (" select tournaments.TourneyLocation from tournaments inner join tour
neymatches on tournaments.TourneyID = tourneymatches.TourneyID group by tournament
s.TourneyID ")
>>>
>>> cursor.execute(query)
>>>
>>> for (TourneyLocation) in cursor:
...     print(TourneyLocation)
...
(u'Red Rooster Arena',)
(u'Thunderbird Arena',)
(u'Bolero Arena',)
(u'Imperial Arena',)
(u'Sports World Arena',)
(u'Totem Arena',)
(u'Acapulco Arena',)
(u'Red Rooster Arena',)
(u'Thunderbird Arena',)
(u'Bolero Arena',)
(u'Imperial Arena',)
(u'Sports World Arena',)
(u'Totem Arena',)
(u'Acapulco Arena',)
>>> cursor.close()
True
>>> cnx.close()
>>>
```