

Assignment 4 SQL

In []:

```
%load_ext sql
import pandas as pd
%sql sqlite://
```

In [1]:

```
pip install ipython-sql
```

Requirement already satisfied: ipython-sql in c:\programdata\anaconda3\lib\site-packages (0.4.0)
Requirement already satisfied: prettytable<1 in c:\programdata\anaconda3\lib\site-packages (from ipython-sql) (0.7.2)
Requirement already satisfied: ipython>=1.0 in c:\programdata\anaconda3\lib\site-packages (from ipython-sql) (7.16.1)
Requirement already satisfied: sqlparse in c:\programdata\anaconda3\lib\site-packages (from ipython-sql) (0.4.1)
Requirement already satisfied: sqlalchemy>=0.6.7 in c:\programdata\anaconda3\lib\site-packages (from ipython-sql) (1.3.18)
Requirement already satisfied: six in c:\programdata\anaconda3\lib\site-packages (from ipython-sql) (1.15.0)
Requirement already satisfied: ipython-genutils>=0.1.0 in c:\programdata\anaconda3\lib\site-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: setuptools>=18.5 in c:\programdata\anaconda3\lib\site-packages (from ipython>=1.0->ipython-sql) (49.2.0.post20200714)
Requirement already satisfied: colorama; sys_platform == "win32" in c:\programdata\anaconda3\lib\site-packages (from ipython>=1.0->ipython-sql) (0.4.3)
Requirement already satisfied: jedi>=0.10 in c:\programdata\anaconda3\lib\site-packages (from ipython>=1.0->ipython-sql) (0.17.1)
Requirement already satisfied: pygments in c:\programdata\anaconda3\lib\site-packages (from ipython>=1.0->ipython-sql) (2.6.1)
Requirement already satisfied: prompt-toolkit!=3.0.0,!<3.0.1,<3.1.0,>=2.0.0 in c:\programdata\anaconda3\lib\site-packages (from ipython>=1.0->ipython-sql) (3.0.5)
Requirement already satisfied: traitlets>=4.2 in c:\programdata\anaconda3\lib\site-packages (from ipython>=1.0->ipython-sql) (4.3.3)
Requirement already satisfied: pickleshare in c:\programdata\anaconda3\lib\site-packages (from ipython>=1.0->ipython-sql) (0.7.5)
Requirement already satisfied: backcall in c:\programdata\anaconda3\lib\site-packages (from ipython>=1.0->ipython-sql) (0.2.0)
Requirement already satisfied: decorator in c:\programdata\anaconda3\lib\site-packages (from ipython>=1.0->ipython-sql) (4.4.2)
Requirement already satisfied: parso<0.8.0,>=0.7.0 in c:\programdata\anaconda3\lib\site-packages (from jedi>=0.10->ipython>=1.0->ipython-sql) (0.7.0)
Requirement already satisfied: wcwidth in c:\programdata\anaconda3\lib\site-packages (from prompt-toolkit!=3.0.0,!<3.0.1,<3.1.0,>=2.0.0->ipython>=1.0->ipython-sql) (0.2.5)
Note: you may need to restart the kernel to use updated packages.

World Cup Data

In [2]:

```
# Load tables from db files
# Players
%load_ext sql
%sql sqlite:///playersExt.db
import matplotlib.pyplot as plt
plt.style.use('bmh')
```

World Cup Basic Queries

1) What player on a team with "ie" in the team name played less than 200 minutes and made more than 100

1) What player on a team with 'ia' in the team name played less than 200 minutes and made more than 100 passes? Return the player surname. Note: To check if attribute A contains string S use "A like '%S%'"

In [3]:

```
%%sql
select surname from playersExt where team like '%ia%' and minutes < 200 and passes > 100

* sqlite:///playersExt.db
Done.
```

Out[3]:

surname
Kuzmanovic

2) Find all players who made more than 10 shots. Return all information about the players in descending order of shots made.

In [5]:

```
%%sql
select * from PlayersExt where shots > 10 order by shots desc

* sqlite:///playersExt.db
Done.
```

Out[5]:

surname	team	ranking	games	wins	draws	losses	goalsFor	goalsAgainst	yellowCards	redCards	position	mini
Gyan	Ghana	32	5	2	2	1	5	4	11	0	forward	
Villa	Spain	2	6	5	0	1	7	2	3	0	forward	
Messi	Argentina	7	5	4	0	1	10	6	7	0	forward	
Suarez	Uruguay	16	6	3	2	1	9	5	8	2	forward	
Ronaldo	Portugal	3	4	1	2	1	7	1	8	1	forward	
Forlan	Uruguay	16	6	3	2	1	9	5	8	2	forward	
Podolski	Germany	6	6	4	0	2	13	3	8	1	forward	
Sneijder	Netherlands	4	6	6	0	0	12	5	15	0	midfielder	
Dempsey	USA	14	4	1	2	1	5	5	9	0	midfielder	
van Persie	Netherlands	4	6	6	0	0	12	5	15	0	forward	
Park Chu-Young	South Korea	47	4	1	1	2	6	8	6	0	forward	
Higuain	Argentina	7	5	4	0	1	10	6	7	0	forward	
Jong Tae-Se	North Korea	105	3	0	0	3	1	12	2	0	forward	
Boateng	Ghana	32	5	2	2	1	5	4	11	0	midfielder	
Daniel Alves	Brazil	1	5	3	1	1	9	4	7	2	defender	
Lampard	England	8	4	1	2	1	3	5	6	0	midfielder	
Klose	Germany	6	6	4	0	2	13	3	8	1	forward	
Asamoah	Ghana	32	5	2	2	1	5	4	11	0	midfielder	
Kuyt	Netherlands	4	6	6	0	0	12	5	15	0	forward	

3) Find the goalkeepers of teams that played more than four games. List the surname of the goalkeeper, the team, and the number of minutes the goalkeeper played.

In [6]:

```
%%sql
select surname, team, minutes from PlayersExt where position = 'goalkeeper' and games > 4

* sqlite:///playersExt.db
Done.
```

Out[6]:

surname	team	minutes
Romero	Argentina	450
Julio Cesar	Brazil	450
Neuer	Germany	540
Kingson	Ghana	510
Stekelenburg	Netherlands	540
Villar	Paraguay	480
Casillas	Spain	540
Muslera	Uruguay	570

4) How many players who play on a team with ranking <10 played more than 350 minutes? Return one number in a column named 'superstar'.

In [7]:

```
%%sql
select count() as superstar from PlayersExt where ranking < 10 and minutes > 350

* sqlite:///playersExt.db
Done.
```

Out[7]:

superstar
54

5) What is the average number of passes made by forwards? By midfielders? Write one query that gives both values with the corresponding position.

In [8]:

```
%%sql
select position, avg(passes) from PlayersExt
where position = "forward" or position = "midfielder" group by position

* sqlite:///playersExt.db
Done.
```

Out[8]:

position	avg(passes)
forward	50.82517482517483
midfielder	95.2719298245614

6) Which team has the highest average number of passes per minute played? Return the team and average passes per minute. Hints: (1) You can compute a team's average number of passes per minute played by dividing the total number of passes by the total number of minutes. To force floating point division, multiply one operand by 1.0. (2) Consider using Limit.

In [9]:

```
* sqlite:///playersExt.db
Done.
```

team	1.0*sum(passes)/sum(minutes)
Spain	0.6230639730639731

1) Find all pairs of teams who have the same number of goalsFor as each other and the same number of goalsAgainst as each other. Return the teams and numbers of goalsFor and goalsAgainst. Make sure to return each pair only once.

```
%%sql
select T1.team, T2.team, T1.goalsFor, T1.goalsAgainst from playersExt T1, playersExt T2
where T1.goalsFor = T2.goalsFor and T1.goalsAgainst = T2.goalsAgainst and T1.team < T2.t
eam
```

```
* sqlite:///playersExt.db
Done.
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

Chile	England	3	5
Chile	England	3	5
Chile	England	3	5
Chile	England	3	5
Chile	England	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	England	3	5
Chile	England	3	5
Chile	England	3	5

[illegible]

[illegible]

[illegible]

Chile	England	3	5
Chile	England	3	5
Chile	England	3	5
Chile	England	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	England	3	5
Chile	England	3	5
Chile	England	3	5
Chile	England	3	5

[illegible]

[illegible]

[illegible]

Chile	England	3	5
Chile	England	3	5
Chile	England	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	Nigeria	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	South Africa	3	5
Chile	England	3	5
Chile	England	3	5
Chile	England	3	5
Chile	England	3	5
Chile	England	3	5

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

2) Find all teams with ranking <30 where no player made more than 150 passes. Return the team and ranking.

In [13]:

```
%%sql
select team, ranking from playersExt T where ranking < 30 and
not exists (select * from PlayersExt P where P.team = T.team and P.passes > 150)
```

```
* sqlite:///playersExt.db
Done.
```

Out [13] :

team	ranking
France	9
France	9
France	9
France	9

[illegible]

Switzerland	24
Switzerland	24
Switzerland	24
Switzerland	24
Switzerland	24
Switzerland	24

3) Which team has the highest ratio of goalsFor to goalsAgainst?

In [14]:

```
%%sql
select team from playersExt
where goalsFor/goalsAgainst = (select max(goalsFor/goalsAgainst) from playersExt)
```

```
* sqlite:///playersExt.db
Done.
```

Out[14]:

team
Portugal
Portugal
Portugal
Portugal
Portugal
Portugal
Portugal
Portugal
Portugal
Portugal
Portugal
Portugal
Portugal
Portugal
Portugal
Portugal
Portugal
Portugal
Portugal

4) Find all team-position pairs where the average number of passes made by players in that position on that team is greater than 150. Return the team-position pairs.

In [15]:

```
%%sql
select team, position from PlayersExt group by team, position having avg(passes) > 150

* sqlite:///playersExt.db
Done.
```

Out[15]:

team	position
------	----------

Argentina	midfielder
Brazil	defender
Germany	defender
Germany	midfielder
Ghana	midfielder
Mexico	defender
Netherlands	defender
Netherlands	midfielder
Spain	defender
Spain	midfielder

5) Find all teams whose defenders averaged more than 150 passes. Return the team and average number of passes by defenders, in descending order of average passes.

In [16]:

```
%%sql
select team, avg(passes) from PlayersExt where position = "defender" group by team
having avg(passes) > 150 order by avg(passes) desc
```

* sqlite:///playersExt.db
Done.

Out[16]:

team	avg(passes)
Spain	213.0
Brazil	190.0
Germany	189.83333333333334
Netherlands	182.5
Mexico	152.14285714285714

Titanic Data

In [17]:

```
# Load tables from db files
# Players
%load_ext sql
%sql sqlite:///Titanic.db
import matplotlib.pyplot as plt
plt.style.use('bmh')
```

The sql extension is already loaded. To reload it, use:
%reload_ext sql

Titanic Basic Queries

1) How many married women over age 50 embarked in Cherbourg? Note: To check if attribute A begins with string S use "A like 'S%'"

In [18]:

```
%%sql
select count() from Titanic where first like 'Mrs.%' and age > 50 and embarked = 'Cherbourg'

* sqlite:///Titanic.db
```

```
* sqlite:///Titanic.db
sqlite:///playersExt.db
Done.
```

Out[18]:

count()

4

2) List the average fare paid by passengers in each of the embarkation cities (along with the city), in descending order of average fare.

In [19]:

```
%%sql
select embarked, avg(fare) from Titanic
group by embarked
order by avg(fare) desc
```

```
* sqlite:///Titanic.db
sqlite:///playersExt.db
Done.
```

Out[19]:

embarked	avg(fare)
Cherbourg	59.95414404761905
Southampton	27.243651393188795
Queenstown	13.276029870129872

3) What's the most common last name among passengers?

In [21]:

```
%%sql
select last from Titanic
group by last
order by count() desc
limit 1
```

```
* sqlite:///Titanic.db
sqlite:///playersExt.db
Done.
```

Out[21]:

last

Andersson

4) Write three queries: the total number of passengers; the number of passengers under 30; the number of passengers 30 or older. Why do the second and third numbers not add up to the first?

In [27]:

```
%%sql
select count() from Titanic
```

```
* sqlite:///Titanic.db
sqlite:///playersExt.db
Done.
```

Out[27]:

count()

891

In [26]:

```
%%sql
select count() from Titanic where age < 30
```

```
* sqlite:///Titanic.db
  sqlite:///playersExt.db
Done.
```

Out[26]:

```
count()
352
```

In [25]:

```
%%sql
select count() from Titanic where age >= 30
```

```
* sqlite:///Titanic.db
  sqlite:///playersExt.db
Done.
```

Out[25]:

```
count()
362
```

5) Blanks in SQL tables are given a special value called 'null', and conditions 'A is null' and 'A is not null' can be used in Where clauses to check whether attribute A has the 'null' value. How many passengers don't have a value for age? Now do your numbers add up?

In [28]:

```
%%sql
select count() from Titanic where age is null
```

```
* sqlite:///Titanic.db
  sqlite:///playersExt.db
Done.
```

Out[28]:

```
count()
177
```

6) For male survivors, female survivors, male non-survivors, and female non-survivors, how many passengers were in each of those four categories and what was their average fare paid?

In [29]:

```
%%sql
select gender, survived, count(), avg(fare)
from Titanic
group by survived, gender
```

```
* sqlite:///Titanic.db
  sqlite:///playersExt.db
Done.
```

Out[29]:

gender	survived	count()	avg(fare)
F	no	81	23.02438518518519
M	no	468	21.960992948717944
F	yes	233	51.93857339055791
M	yes	109	40.82148440366974

Titanic Advanced Queries

1) Are there any pairs of passengers with the same last name where one is in first class and the other is in third class? If so, return the last name and the two first names. Label the first-name columns 'first' for the passenger in first class and 'third' for the passenger in third class.

In [30]:

```
%%sql
select T1.last, T1.first, T2.first as 'third' from Titanic T1, Titanic T2
where T1.last = T2.last and T1.class = 1 and T2.class = 3
```

```
* sqlite:///Titanic.db
sqlite:///playersExt.db
Done.
```

Out[30]:

last	first	third
Allen	Miss Elisabeth Walton	Mr. William Henry
Smith	Mr. James Clinch	Mr. Thomas
Smith	Mr. Richard William	Mr. Thomas
Williams	Mr. Charles Duane	Mr. Howard Hugh "Harry"
Flynn	Mr. John Irwin ("Irving")	Mr. James
Daly	Mr. Peter Denis	Mr. Eugene Patrick
Williams	Mr. Charles Duane	Mr. Leslie
Carlsson	Mr. Frans Olof	Mr. August Sigfrid
Flynn	Mr. John Irwin ("Irving")	Mr. John

2) Which embarkation cities have more than 40 passengers whose age is missing? Reminder: Blanks in SQL tables are given a special value called 'null', and conditions 'A is null' and 'A is not null' can be used in Where clauses to check whether attribute A has the 'null' value.

In [31]:

```
%%sql
select distinct embarked from Titanic T1
where 40 < (select count() from Titanic T2 where T2.embarked = T1.embarked and age is null)
```

```
* sqlite:///Titanic.db
sqlite:///playersExt.db
Done.
```

Out[31]:

embarked
Southampton
Queenstown

3) Find all classes where the average fare paid by passengers in that class was more than twice the overall average or less than half the overall average.

In [32]:

```
%%sql
select class from Titanic group by class
having avg(fare)>2*(select avg(fare) from Titanic) or avg(fare)<0.5*(select avg(fare) from Titanic)
```

```
* sqlite:///Titanic.db
  sqlite:///playersExt.db
Done.
```

Out[32]:

class
1
3

4) EXTRA CREDIT CHALLENGE: List each class and its survival rate, i.e., the fraction of passengers in that class who survived.

In [33]:

```
%%sql
select yes.class, 1.0*Scount / Tcount as survival
from
(select class, count() as Scount
from Titanic
where survived = 'yes'
group by class)yes,
(select class, count() as Tcount
from Titanic
group by class)total
where yes.class = total.class
```

```
* sqlite:///Titanic.db
  sqlite:///playersExt.db
Done.
```

Out[33]:

class	survival
1	0.6296296296296297
2	0.47282608695652173
3	0.24236252545824846

Titanic Data Modification

1) Subtract 5 from the fare paid by any passenger under the age of 10. Then compute the new average fare. NOTE: You can put two SQL statements in one cell separated by a semicolon.

In [35]:

```
%%sql
update Titanic set fare = fare-5 where age < 10;
select avg(fare) from Titanic

* sqlite:///Titanic.db
  sqlite:///playersExt.db
14 rows affected.
Done.
```

Out[35]:

avg(fare)
32.125644556677855

2) Create a new table called Survivors, containing the last and first names of all passengers who survived. Then count the number of tuples in the new table.

In [36]:

```
%%sql
drop table if exists Survivors;
create table Survivors(last, first);
insert into Survivors
select last, first from Titanic
where survived = 'yes';
select count() from Survivors
```

```
* sqlite:///Titanic.db
  sqlite:///playersExt.db
Done.
Done.
342 rows affected.
Done.
```

Out[36]:

```
count()
```

```
342
```

3) In the Titanic table delete all but the highest-paying passengers.

In [37]:

```
%%sql
delete from Titanic where fare < (select max(fare) from Titanic)
```

```
* sqlite:///Titanic.db
  sqlite:///playersExt.db
888 rows affected.
```

Out[37]:

```
[]
```

4) In what's left of the table after (3), insert a new tuple for yourself. You can decide your class, fare, where you embarked, and whether you survived. Then show the whole table.

In [39]:

```
%%sql
insert into Titanic values ('Widom', 'Jennifer', 'F', 57, 2, 10, 'Stanford', 'yes');
select * from Titanic
```

```
* sqlite:///Titanic.db
  sqlite:///playersExt.db
1 rows affected.
Done.
```

Out[39]:

last	first	gender	age	class	fare	embarked	survived
Ward	Miss Anna	F	35	1	512.3292	Cherbourg	yes
Cardeza	Mr. Thomas Drake Martinez	M	36	1	512.3292	Cherbourg	yes
Lesurer	Mr. Gustave J	M	35	1	512.3292	Cherbourg	yes
Widom	Jennifer	F	57	2	10.0	Stanford	yes
Widom	Jennifer	F	57	2	10.0	Stanford	yes

END