Functions and Grouping

Functions

Rentals Database

The rentals database include include the following tables:

Primary key of each relation is underlined.

accommodations (id , title, location, price, room, rating, type)

> - id - accommodation id
> - title - accommodation title
> - location - accommodation location
> - price - accommodation price
> - rating - accommodation rating (by company)
> - type - accommodation type
> - **id is a primary key**

ratings (userId, accoId, rating)

> - userId - user id
> - accoId - accommodation id
> - rating - rating of an accommodation by user
> - **userId and accoId - primary key**

In [1]:

```
%load_ext sql
%sql sqlite:///rentals.db
```

--

`%sql` is used for single line SQL commands:

`%%sql` is used for multiple lines SQL commands:

In [17]:

```
%%sql

-- LIMIT restricts how many rows are displayed

SELECT *
FROM accommodations
LIMIT 5;
```

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

Out[17]:

id	title	location	price	rooms	rating	type
1	Comfy Quiet Chalet	Vancouver	50	3	3.1	cottage
2	Cozy Calm Hut	London	65	2	4.1	cottage
3	Agreable Calm Place	London	65	4	4.8	house
4	Colossal Quiet Chateau	Paris	3400	16	2.7	castle
5	Homy Quiet Shack	Paris	50	1	1.1	cottage

In [18]:

```
%%sql
SELECT *
FROM ratings
LIMIT 5;
```

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

Out[18]:

userId	accoId	rating
10	1	1
18	1	2
13	1	1
7	2	2
4	2	2

Functions

In [4]:

```
%%sql

-- Count the rows in the accommodations table
```

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

Out[4]:

count(*)
99

In [5]:

```
%%sql

-- How many accommodations are in Paris?
```

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

Out[5]:

count(*)
9

In [6]:

```
%%sql

-- How many castles have a rating over 3.0?
-- Display the result using column heading "# castles"
```

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

Out[6]:

castles
14

In [21]:

```
%%sql

-- Display rating min, max, avg, and sum
```

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

Out[21]:

Rating Min	Max	Avg	Sum
1.0	4.9	2.9686868686868686	293.9

In [8]:
```
%%sql

-- The round function will restrict the number of digits after the decimal place.
-- round(7.456789, 2)  will produce 7.45.
-- Use the round function to restrict the average to 2 digits after decimal place
```

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

Out[8]:

Rating Min	Max	Avg	Sum
1.0	4.9	2.97	293.9

GROUP BY

In [22]:
```
%%sql

-- Display row count per type
-- This requires the GROUP BY clause
```

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

Out[22]:

type	count
castle	33
cottage	28
house	20
mansion	18

In [23]:

```
%%sql

-- THIS IS INCORRECT!  Missing a GROUP BY clause.
-- Displays the type found in last row along with the total row count
```

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

Out[23]:

type	count
house	99

In [24]:

```
%%sql

-- Display row count per type and location
-- GROUP BY must match non-function columns from SELECT clause
```

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

Out[24]:

type	location	count
castle	Auckland	1
castle	Berlin	3
castle	Buenos Aires	2
castle	London	2
castle	Melbourne	6
castle	NYC	2
castle	Paris	5
castle	San Francisco	2
castle	Seattle	5
castle	Tokyo	2
castle	Vancouver	3
cottage	Auckland	3
cottage	Berlin	1
cottage	Buenos Aires	2
cottage	London	3
cottage	Melbourne	3
cottage	NYC	3
cottage	Paris	2
cottage	San Francisco	6
cottage	Seattle	4
cottage	Vancouver	1
house	Auckland	1
house	Berlin	3
house	Dublin	1
house	London	3
house	Melbourne	1
house	NYC	3

house	Paris	1
house	San Francisco	1
house	Seattle	2
house	Tokyo	3
house	Vancouver	1
mansion	Auckland	2
mansion	Buenos Aires	1
mansion	Dublin	1
mansion	London	2
mansion	Melbourne	2
mansion	NYC	1
mansion	Paris	1
mansion	San Francisco	3
mansion	Seattle	3
mansion	Tokyo	1
mansion	Vancouver	1

In [12]:

```
%%sql

-- Display row count per type and location
```

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

Out[12]:

location	type	count
Auckland	castle	1
Auckland	cottage	3
Auckland	house	1
Auckland	mansion	2
Berlin	castle	3
Berlin	cottage	1
Berlin	house	3
Buenos Aires	castle	2
Buenos Aires	cottage	2
Buenos Aires	mansion	1
Dublin	house	1
Dublin	mansion	1
London	castle	2
London	cottage	3
London	house	3
London	mansion	2
Melbourne	castle	6
Melbourne	cottage	3
Melbourne	house	1
Melbourne	mansion	2
NYC	castle	2
NYC	cottage	3
NYC	house	3
NYC	mansion	1
Paris	castle	5
Paris	cottage	2
Paris	house	1

Paris	mansion	1
San Francisco	castle	2
San Francisco	cottage	6
San Francisco	house	1
San Francisco	mansion	3
Seattle	castle	5
Seattle	cottage	4
Seattle	house	2
Seattle	mansion	3
Tokyo	castle	2
Tokyo	house	3
Tokyo	mansion	1
Vancouver	castle	3
Vancouver	cottage	1
Vancouver	house	1
Vancouver	mansion	1

In [27]:

```
%%sql

-- Display average price per location. Sort results by average price
```

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

Out[27]:

location	average price
London	494.5
Dublin	592.5
Auckland	657.1428571428571
San Francisco	667.0833333333334
Tokyo	699.1666666666666
NYC	885.5555555555555
Buenos Aires	911.0
Berlin	1032.857142857143
Seattle	1216.0714285714287
Melbourne	1505.4166666666667
Paris	1583.3333333333333
Vancouver	1625.0

In [28]:

```
%%sql

-- Same query, round average price to whole dollars
```

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

Out[28]:

location	average price
London	495.0
Dublin	593.0
Auckland	657.0
San Francisco	667.0
Tokyo	699.0
NYC	886.0
Buenos Aires	911.0
Berlin	1033.0
Seattle	1216.0
Melbourne	1505.0
Paris	1583.0
Vancouver	1625.0

HAVING - SFWGHO

In [14]:
```
%%sql

-- Display location and counts above 10
-- Use HAVING clause to restrict result based on function
```

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

Out[14]:

location	count(*)
Melbourne	12
San Francisco	12
Seattle	14

In [15]:
```
%%sql

-- INCORRECT!! Use HAVING to test function value
-- Using a multi-line comment
```

```
 * sqlite:///rentals.db
0 rows affected.
```

Out[15]:
```
[]
```

In [16]:
```
%%sql

-- Display location, avg, min and max rating for castles.
-- Restrict the results to locations with average rating above 3.5.
-- Sort in descending order of average rating
```

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

Out[16]:

location	avg(rating)	min(rating)	max(rating)
Buenos Aires	3.9	3.3	4.5
San Francisco	3.9	3.7	4.1
Vancouver	3.6999999999999997	3.2	4.0

In [38]:

```
%%sql

-- Assume you only have $40.  Which locations can you visit?
-- Show minimum price for each location, label the resulting column "cheapest".
-- Do not show location with a min price above $40.
```

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

Out[38]:

location	cheapest
Berlin	30
Buenos Aires	40
Dublin	35
San Francisco	40
Seattle	35
Tokyo	30

In [ ]:

```
```