[301] Database 3

Tyler Caraza-Harter

Learning Objectives Today

GROUP BY

- how to break data into buckets
- combination of GROUP BY with ORDER BY
- WHERE vs. HAVING

Aggregates

- SUM, COUNT, MAX, MIN, SUM
- Aliases with AS

Outline

Aggregation Queries

Grouping with Python

Grouping with SQL

Top results

WHERE vs. HAVING

Example: Movies Database

In [32]:
```
c = sqlite3.connect('movies.db')
df = pd.read_sql("select * from movies", c)
c.close()
df
```

Out[32]:

	Title	Director	Year	Runtime	Rating	Revenue
0	Guardians of the Galaxy	James Gunn	2014	121	8.1	333.13
1	Prometheus	Ridley Scott	2012	124	7.0	126.46
2	Split	M. Night Shyamalan	2016	117	7.3	138.12
3	Sing	Christophe Lourdelet	2016	108	7.2	270.32
4	Suicide Squad	David Ayer	2016	123	6.2	325.02
5	The Great Wall	Yimou Zhang	2016	103	6.1	45.13
6	La La Land	Damien Chazelle	2016	128	8.3	151.06

Title	Director	Year	Runtime	Rating	Revenue
Guardians of the Galaxy	James Gunn	2014	121	8.1	333.13
Prometheus	Ridley Scott	2012	124	7.0	126.46
Split	M. Night Shyamalan	2016	117	7.3	138.12
Sing	Christophe Lourdelet	2016	108	7.2	270.32
Suicide Squad	David Ayer	2016	123	6.2	325.02
The Great Wall	Yimou Zhang	2016	103	6.1	45.13
La La Land	Damien Chazelle	2016	128	8.3	151.06

Question: which movie has the highest rating?

SQL Query:

```
SELECT Title FROM Movies
ORDER BY Rating DESC
LIMIT 1
```

Title	Director	Year	Runtime	Rating	Revenue
Guardians of the Galaxy	James Gunn	2014	121	8.1	333.13
Prometheus	Ridley Scott	2012	124	7.0	126.46
Split	M. Night Shyamalan	2016	117	7.3	138.12
Sing	Christophe Lourdelet	2016	108	7.2	270.32
Suicide Squad	David Ayer	2016	123	6.2	325.02
The Great Wall	Yimou Zhang	2016	103	6.1	45.13
La La Land	Damien Chazelle	2016	128	8.3	151.06

Question: which director made the shortest movie?

SQL Query:

```
SELECT Director FROM Movies
ORDER BY Runtime ASC
LIMIT 1
```

Title	Director	Year	Runtime	Rating	Revenue
Guardians of the Galaxy	James Gunn	2014	121	8.1	333.13
Prometheus	Ridley Scott	2012	124	7.0	126.46
Split	M. Night Shyamalan	2016	117	7.3	138.12
Sing	Christophe Lourdelet	2016	108	7.2	270.32
Suicide Squad	David Ayer	2016	123	6.2	325.02
The Great Wall	Yimou Zhang	2016	103	6.1	45.13
La La Land	Damien Chazelle	2016	128	8.3	151.06

Question: which director made the highest-revenue movie?

SQL Query:

```
SELECT Director FROM Movies
ORDER BY Revenue DESC
LIMIT 1
```

Title	Director	Year	Runtime	Rating	Revenue
Guardians of the Galaxy	James Gunn	2014	121	8.1	333.13
Prometheus	Ridley Scott	2012	124	7.0	126.46
Split	M. Night Shyamalan	2016	117	7.3	138.12
Sing	Christophe Lourdelet	2016	108	7.2	270.32
Suicide Squad	David Ayer	2016	123	6.2	325.02
The Great Wall	Yimou Zhang	2016	103	6.1	45.13
La La Land	Damien Chazelle	2016	128	8.3	151.06

Question: which movie had the highest revenue in 2016?

SQL Query:

```
SELECT Director FROM Movies
WHERE Year = 2016
ORDER BY Revenue DESC
LIMIT 1
```

Title	Director	Year	Runtime	Rating	Revenue
Guardians of the Galaxy	James Gunn	2014	121	8.1	333.13
Prometheus	Ridley Scott	2012	124	7.0	126.46
Split	M. Night Shyamalan	2016	117	7.3	138.12
Sing	Christophe Lourdelet	2016	108	7.2	270.32
Suicide Squad	David Ayer	2016	123	6.2	325.02
The Great Wall	Yimou Zhang	2016	103	6.1	45.13
La La Land	Damien Chazelle	2016	128	8.3	151.06

Question: which 3 movies had the highest revenues in 2016?

SQL Query:

```
SELECT Director FROM Movies
WHERE Year = 2016
ORDER BY Revenue DESC
LIMIT 3
```

Data Questions

which movie has the highest rating?

which director made the shortest movie?

which director made the highest-revenue movie?

which movie had the highest revenue in 2016?

which 3 movies had the highest revenues in 2016?

These questions all have something in common:

identify certain rows, then **just extract**
specific columns from those rows to answer

Data Questions

which movie has the highest rating?

which director made the shortest movie?

which director made the highest-revenue movie?

which movie had the highest revenue in 2016?

which 3 movies had the highest revenues in 2016?

These questions all have something in common:

identify certain rows, then **just extract**
specific columns from those rows to answer

Sometimes we want a summary over multiple rows

called an **"aggregate"**

Extract data:

which movie has the highest rating?

which director made the shortest movie?

which director made the highest-revenue movie?

which movie had the highest revenue in 2016?

which 3 movies had the highest revenues in 2016?

Summarize data:

what is the **average** rating across all movies?

what is the **average** runtime for a James Gunn movie?

what is the **average** revenue for a Ridley Scott movie?

how many movies were there in 2016?

what was the **total** revenue of all movies in 2016?

Extract data:

which movie has the highest rating?

which director made the shortest movie?

which director made the highest-revenue movie?

which movie had the highest revenue in 2016?

which 3 movies had the highest revenues in 2016?

today

Summarize data:

what is the **average** rating across all movies?

what is the **average** runtime for a James Gunn movie?

what is the **average** revenue for a Ridley Scott movie?

how many movies were there in 2016?

what was the **total** revenue of all movies in 2016?

Title	Director	Year	Runtime	Rating	Revenue
Guardians of the Galaxy	James Gunn	2014	121	8.1	333.13
Prometheus	Ridley Scott	2012	124	7.0	126.46
Split	M. Night Shyamalan	2016	117	7.3	138.12
Sing	Christophe Lourdelet	2016	108	7.2	270.32
Suicide Squad	David Ayer	2016	123	6.2	325.02
The Great Wall	Yimou Zhang	2016	103	6.1	45.13
La La Land	Damien Chazelle	2016	128	8.3	151.06

Question: what is the total revenue of all the movies?

SQL Query:

```
SELECT * FROM Movies
```

	Revenue
0	333.13
1	126.46
2	138.12
3	270.32
4	325.02
5	45.13
6	151.06
7	0.00
8	8.01

Question: what is the total revenue of all the movies?

SQL Query:

```
SELECT Revenue FROM Movies
```

	SUM(Revenue)
0	72215.45

Question: what is the total revenue of all the movies?

SQL Query:

```
SELECT SUM(Revenue) FROM Movies
```

	SUM(Revenue)
0	72215.45

Question: what is the total revenue of all the movies?

SQL Query:

```
SELECT SUM(Revenue) FROM Movies
```

SUM is an aggregate function

	SUM(Revenue)
0	72215.45

Question: what is the total revenue of all the movies?

SQL Query:

```
SELECT SUM(Revenue) FROM Movies
```

SUM is an aggregate function

aggregates functions: SUM, AVG, COUNT, MIN, MAX

	SUM(Revenue)	COUNT()
0	72215.45	998

Question: what is the total revenue of all the movies?
and how many movies are there?

SQL Query:

```
SELECT SUM(Revenue), COUNT() FROM Movies
```

COUNT doesn't need an argument

aggregates functions: SUM, AVG, COUNT, MIN, MAX

	SUM(Revenue) / COUNT()
0	72.36017

Question: what is the average revenue of all the movies?

SQL Query:

```
SELECT SUM(Revenue) / COUNT() FROM Movies
```

you can combine aggregates

aggregates functions: SUM, AVG, COUNT, MIN, MAX

	AVG(Revenue)
0	72.36017

Question: what is the average revenue of all the movies?

SQL Query:

```
SELECT AVG(Revenue) FROM Movies
```

SUM divided by COUNT

aggregates functions: SUM, AVG, COUNT, MIN, MAX

	AVG(Revenue)	AVG(Runtime)
0	72.36017	113.170341

Question: what is the average revenue of all the movies?
what is the average runtime of all the movies?

SQL Query:

```
SELECT AVG(Revenue), AVG(Runtime) FROM Movies
```

aggregates functions: SUM, AVG, COUNT, MIN, MAX

Question: what percentage of the total revenue came from the highest-revenue movie?

SQL Query: `???`

aggregates functions: SUM, AVG, COUNT, MIN, MAX

	MAX(revenue) / SUM(revenue)
0	0.01297

Question: what percentage of the total revenue came from the highest-revenue movie?

SQL Query:

```
SELECT MAX(Revenue)/SUM(Revenue) FROM Movies
```

aggregates functions: SUM, AVG, COUNT, MIN, MAX

	MAX(revenue) * 100.0 / SUM(revenue)
0	1.296994

Question: what percentage of the total revenue came from the highest-revenue movie?

SQL Query:

```
SELECT 100 * MAX(Revenue)/SUM(Revenue) FROM Movies
```

aggregates functions: SUM, AVG, COUNT, MIN, MAX

clunky column name for Pandas DataFrame

	MAX(revenue) * 100.0 / SUM(revenue)
0	1.296994

Question: what percentage of the total revenue came from the highest-revenue movie?

SQL Query:

```
SELECT 100 * MAX(Revenue)/SUM(Revenue) FROM Movies
```

aggregates functions: SUM, AVG, COUNT, MIN, MAX

	percent
0	1.296994

Question: what percentage of the total revenue came from the highest-revenue movie?

SQL Query:

```
SELECT 100 * MAX(Revenue)/SUM(Revenue) AS percent
FROM Movies
```

aggregates functions: SUM, AVG, COUNT, MIN, MAX

	percent
0	1.296994

Question: what percentage of the total revenue came from the highest-revenue movie?

SQL Query:

```
SELECT 100 * MAX(Revenue)/SUM(Revenue) AS percent
FROM Movies
```

you can use "AS" to call columns whatever you like…

aggregates functions: SUM, AVG, COUNT, MIN, MAX

	percent
0	1.296994

Question: what percentage of the total revenue came from the highest-revenue movie?

SQL Query:

```
SELECT 100 * MAX(Revenue)/SUM(Revenue) AS percent
FROM Movies
```

aggregates functions: SUM, AVG, COUNT, MIN, MAX

	percent
0	1.296994

what if we want to answer this question just for movies in 2016?

Question: what percentage of the total revenue came from the highest-revenue movie?

SQL Query:

```
SELECT 100 * MAX(Revenue)/SUM(Revenue) AS percent
FROM Movies
```

aggregates functions: SUM, AVG, COUNT, MIN, MAX

what if we want to
answer this question just
for movies in 2016?

Question: what percentage of the total revenue **in 2016** came from the highest-revenue movie?

SQL Query:

```
SELECT 100 * MAX(Revenue)/SUM(Revenue) AS percent
FROM Movies
```

aggregates functions: SUM, AVG, COUNT, MIN, MAX

Question: what percentage of the total revenue **in 2016** came from the highest-revenue movie?

SQL Query:

```
SELECT 100 * MAX(Revenue)/SUM(Revenue) AS percent
FROM Movies
WHERE year = 2016
```

you can combine WHERE with aggregates
(filtering is done before aggregation)

aggregates functions: SUM, AVG, COUNT, MIN, MAX

Year	percent
2014	333.13
2012	126.46
2016	138.12
2016	270.32
2016	325.02
2016	45.13
2016	151.06

in progress…

Question: what percentage of the total revenue **in 2016** came from the highest-revenue movie?

SQL Query:

```
SELECT 100 * MAX(Revenue)/SUM(Revenue) AS percent
FROM Movies
WHERE year = 2016
```

you can combine WHERE with aggregates
(filtering is done before aggregation)

aggregates functions: SUM, AVG, COUNT, MIN, MAX

max(revenue)	sum(revenue)
532.17	11211.65

in progress…

Question: what percentage of the total revenue **in 2016** came from the highest-revenue movie?

SQL Query:

```
SELECT 100 * MAX(Revenue)/SUM(Revenue) AS percent
FROM Movies
WHERE year = 2016
```

you can combine WHERE with aggregates
(filtering is done before aggregation)

aggregates functions: SUM, AVG, COUNT, MIN, MAX

	percent
0	4.746581

Question: what percentage of the total revenue **in 2016** came from the highest-revenue movie?

SQL Query:

```
SELECT 100 * MAX(Revenue)/SUM(Revenue) AS percent
FROM Movies
WHERE year = 2016
```

you can combine WHERE with aggregates
(filtering is done before aggregation)

aggregates functions: SUM, AVG, COUNT, MIN, MAX

Outline

Aggregation Queries

Grouping with Python

Grouping with SQL

Top results

WHERE vs. HAVING

Extract data:

which movie has the highest rating?

which director made the shortest movie?

previously which director made the highest-revenue movie?

which movie had the highest revenue in 2016?

which 3 movies had the highest revenues in 2016?

Summarize data:

what is the average rating across all movies?

what is the average runtime for a James Gunn movie?

just now what is the average revenue for a Ridley Scott movie?

how many movies were there in 2016?

what was the total revenue of all movies in 2016?

Summarize across groups:

what is the average rating for **each** director?

what is the average runtime for **each** director?

now… what is the average revenue for **each** year?

how many movies were there in **each** year?

what was the total revenue for **each** year?

Outline

Aggregation Queries

Grouping with Python

Grouping with SQL

Top results

WHERE vs. HAVING

Outline

Aggregation Queries

Grouping with Python

Grouping with SQL

Top results

WHERE vs. HAVING

Outline

Aggregation Queries

Grouping with Python

Grouping with SQL

Top results

WHERE vs. HAVING