Essential SQL

APPLYING SQL TO REAL-WORLD PROBLEMS

Dmitriy (Dima) Gorenshteyn
Lead Data Scientist, Memorial Sloan Kettering Cancer Center

datacamp

What you will learn!

- **Chapter 1:** Use Real World SQL
- **Chapter 2:** Find Your Data
- **Chapter 3:** Manage Your Data
- **Chapter 4:** Best Practices for Writing SQL

The database

- DVD Rental Store
- Based off the pagila database [1]

Tables	
actor	film_actor
address	inventory
category	language
customer	payment
film	rental

[1] https://github.com/devrimgunduz/pagila

Essential SQL commands

```
SELECT title, length
FROM film AS f
INNER JOIN category AS c
  ON f.film_id = c.film_id
WHERE c.category = 'Documentary'
  AND f.rating IN ('G', 'PG')
ORDER BY length DESC;
```

SELECT, FROM

```
SELECT title, length
FROM film




;
```

SELECT

```
SELECT <column1>, <column2>
-- OR
SELECT *
```

FROM

```
FROM <table_name>
```

INNER JOIN

```
SELECT title, length
FROM film AS f
INNER JOIN category AS c
  ON f.film_id = c.film_id


;
```

AS

```
<column_name> AS <alias1>
```

INNER JOIN

```
INNER JOIN <table_name> AS <alias2>
```

ON

```
ON <alias1>.<column> = <alias2>.<column>
```

WHERE, AND, IN

```
SELECT title, length
FROM film AS f
INNER JOIN category AS c
  ON f.film_id = c.film_id
WHERE c.category = 'Documentary'
  AND f.rating IN ('G', 'PG')
;
```

WHERE

```
WHERE <condition>
```

AND

```
AND <condition>
```

IN

```
IN ('<item1>', '<item2>', ...)
```

ORDER BY, DESC

```
SELECT title, length
FROM film AS f
INNER JOIN category AS c
  ON f.film_id = c.film_id
WHERE c.category = 'Documentary'
  AND f.rating IN ('G', 'PG')
ORDER BY length DESC;
```

ORDER BY

```
ORDER BY <column_name>
```

DESC

```
ORDER BY <column_name> DESC
```

Practice the essentials!

APPLYING SQL TO REAL-WORLD PROBLEMS

Transforming your results

APPLYING SQL TO REAL-WORLD PROBLEMS

Dmitriy (Dima) Gorenshteyn
Lead Data Scientist, Memorial Sloan Kettering Cancer Center

datacamp

Transforming strings

`UPPER(<column>)` & `LOWER(<column>)`

```
SELECT city,
       UPPER(city) AS upper_city,
       LOWER(city) AS lower_city
FROM address;
```

```
city           lower_city     upper_city
Lethbridge     lethbridge     LETHBRIDGE
Woodridge      woodridge      WOODRIDGE
Lethbridge     lethbridge     LETHBRIDGE
Woodridge      woodridge      WOODRIDGE
```

Transforming numbers

Operators: add (`+`), subtract (`-`), divide (`/`), multiply(`*`)

```
SELECT replacement_cost,
       replacement_cost + 2 AS updated_cost,
       replacement_cost / length AS cost_per_minute
FROM film;
```

```
replacement_cost     updated_cost     cost_per_minute
20.99                22.99            0.24406977
12.99                14.99            0.27062500
18.99                20.99            0.37980000
```

Transforming dates

```
EXTRACT(<part> FROM <date_column>)
```

```
SELECT rental_date,
       EXTRACT(YEAR FROM rental_date) AS rental_year,
       EXTRACT(HOUR FROM rental_date) AS rental_hour
FROM rental;
```

```
rental_date             rental_year    rental_hour
2005-05-30 23:54:19     2005            23
2005-05-30 23:55:36     2005            23
2005-05-31 00:06:02     2005            0
```

Time to transform!

APPLYING SQL TO REAL-WORLD PROBLEMS

Working with aggregate functions

APPLYING SQL TO REAL-WORLD PROBLEMS

Dmitriy (Dima) Gorenshteyn
Lead Data Scientist, Memorial Sloan Kettering Cancer Center

datacamp

GROUP BY

```
SELECT rating,
       AVG(replacement_cost)
FROM film
GROUP BY rating
```

GROUP BY

```
SELECT rating,
       AVG(replacement_cost)
FROM film
GROUP BY rating
```

rating	title	length	release_year	replacement_cost
G	ACE GOLDFINGER	48	2010.0	13.00
G	AFFAIR PREJUDICE	117	2009	27.00
G	AFRICAN EGG	130	2008.0	23.00
PG	ACADEMY DINOSAUR	86.0	2010.0	21.00
PG	AGENT TRUMAN	169.0	2007.0	18.0
PG	ALASKA PHANTOM	136.0	2009.0	23.00
PG-13	AIRPLANE SIERRA	62.0	2009.0	29.00
PG-13	ALABAMA DEVIL	114	2008.0	22.00
PG-13	ALTER VICTORY	57.0	2007.0	28.00
R	AIRPORT POLLOCK	54.0	2004.0	16.0
R	DATE SPEED	104	2010.0	20.00
R	ALONE TRIP	82.0	2004.0	15.0

GROUP BY

```
SELECT rating,
       AVG(replacement_cost)
FROM film
GROUP BY rating
```

rating	title	length	release_year	replacement_cost
G	ACE GOLDFINGER	48	2010.0	13.00
G	AFFAIR PREJUDICE	117	2009	27.00
G	AFRICAN EGG	130	2008.0	23.00
PG	ACADEMY DINOSAUR	86.0	2010.0	21.00
PG	AGENT TRUMAN	169.0	2007.0	18.0
PG	ALASKA PHANTOM	136.0	2009.0	23.00
PG-13	AIRPLANE SIERRA	62.0	2009.0	29.00
PG-13	ALABAMA DEVIL	114	2008.0	22.00
PG-13	ALTER VICTORY	57.0	2007.0	28.00
R	AIRPORT POLLOCK	54.0	2004.0	16.0
R	DATE SPEED	104	2010.0	20.00
R	ALONE TRIP	82.0	2004.0	15.0

GROUP BY

```
SELECT rating,
       AVG(replacement_cost)
FROM film
GROUP BY rating
```

rating	title	length	release_year	replacement_cost
G	ACE GOLDFINGER	48	2010.0	13.00
G	AFFAIR PREJUDICE	117	2009	27.00
G	AFRICAN EGG	130	2008.0	23.00
PG	ACADEMY DINOSAUR	86.0	2010.0	21.00
PG	AGENT TRUMAN	169.0	2007.0	18.0
PG	ALASKA PHANTOM	136.0	2009.0	23.00
PG-13	AIRPLANE SIERRA	62.0	2009.0	29.00
PG-13	ALABAMA DEVIL	114	2008.0	22.00
PG-13	ALTER VICTORY	57.0	2007.0	28.00
R	AIRPORT POLLOCK	54.0	2004.0	16.0
R	DATE SPEED	104	2010.0	20.00
R	ALONE TRIP	82.0	2004.0	15.0

GROUP BY

```
SELECT rating,
       AVG(replacement_cost)
FROM film
GROUP BY rating
```

rating	title	length	release_year	replacement_cost
G	ACE GOLDFINGER	48	2010.0	13.00
G	AFFAIR PREJUDICE	117	2009	27.00
G	AFRICAN EGG	130	2008.0	23.00
PG	ACADEMY DINOSAUR	86.0	2010.0	21.00
PG	AGENT TRUMAN	169.0	2007.0	18.0
PG	ALASKA PHANTOM	136.0	2009.0	23.00
PG-13	AIRPLANE SIERRA	62.0	2009.0	29.00
PG-13	ALABAMA DEVIL	114	2008.0	22.00
PG-13	ALTER VICTORY	57.0	2007.0	28.00
R	AIRPORT POLLOCK	54.0	2004.0	16.0
R	DATE SPEED	104	2010.0	20.00
R	ALONE TRIP	82.0	2004.0	15.0

Rating	AVG(replacement_cost)
G	20.1
PG	19.0
PG-13	20.4
R	20.2

Numeric aggregate functions

```
SELECT rating,
       AVG(replacement_cost) AS avg_cost,
       COUNT(rating) AS number_elements,
       SUM(replacement_cost) AS total_cost
FROM film
GROUP BY rating;
```

```
rating    avg_cost    number_elements    total_cost
PG-13     20.40256    223                4549.77
R         20.23103    195                3945.05
G         20.12483    178                3582.22
PG        18.95907    194                3678.06
```

String aggregate functions

```
STRING_AGG(<column>, '<separator>')
```

```
SELECT rating,
       STRING_AGG(title, ',') as films
FROM film
GROUP BY rating;
```

```
rating       films
PG-13        AIRPLANE SIERRA,ALABAMA DEVIL,...
R            AIRPORT POLLOCK,DATE SPEED,...
G            ACE GOLDFINGER,AFFAIR PREJUDICE,...
```

Time to aggregate!

APPLYING SQL TO REAL-WORLD PROBLEMS