Week 4, Homework 2 – SQL II Solutions

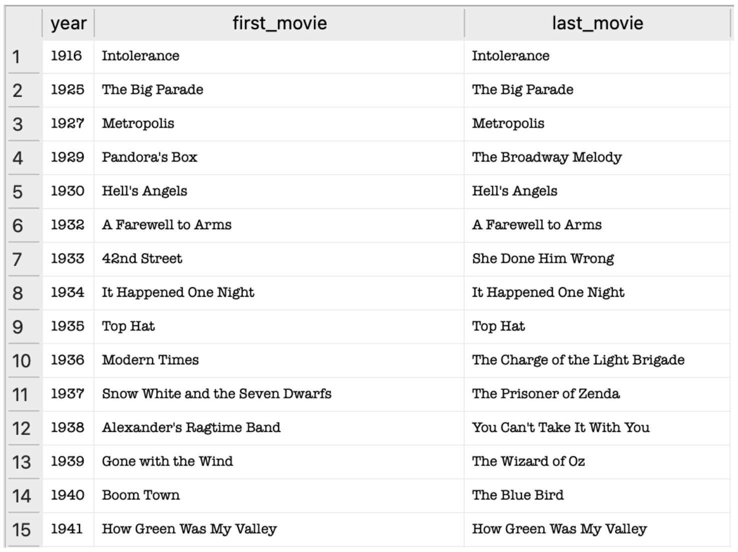
Mariah N Cornelio, 1002053287

#1. Find the first and last movie released each year

- Code

```sql
SELECT strftime("%Y", release_date) as year, min(title) as first_movie, max(title) as last_movie
FROM movies
GROUP BY year;
```

- Output



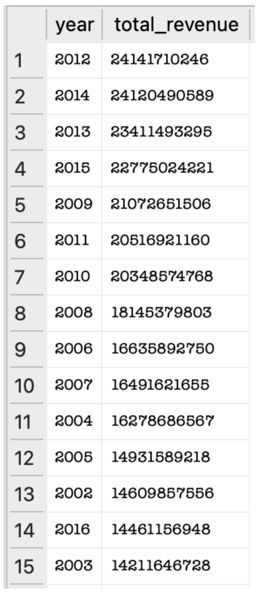
	year	first_movie	last_movie
1	1916	Intolerance	Intolerance
2	1925	The Big Parade	The Big Parade
3	1927	Metropolis	Metropolis
4	1929	Pandora's Box	The Broadway Melody
5	1930	Hell's Angels	Hell's Angels
6	1932	A Farewell to Arms	A Farewell to Arms
7	1933	42nd Street	She Done Him Wrong
8	1934	It Happened One Night	It Happened One Night
9	1935	Top Hat	Top Hat
10	1936	Modern Times	The Charge of the Light Brigade
11	1937	Snow White and the Seven Dwarfs	The Prisoner of Zenda
12	1938	Alexander's Ragtime Band	You Can't Take It With You
13	1939	Gone with the Wind	The Wizard of Oz
14	1940	Boom Town	The Blue Bird
15	1941	How Green Was My Valley	How Green Was My Valley

#2. Total revenue generated each year

- Code

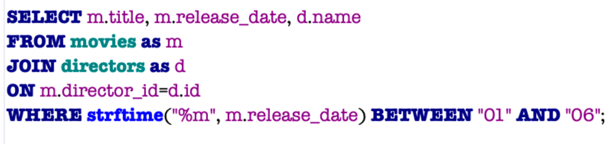
```sql
SELECT strftime("%Y", release_date) as year, SUM(revenue) as total_revenue
FROM movies
GROUP BY year
ORDER BY total_revenue DESC;
```

- Output

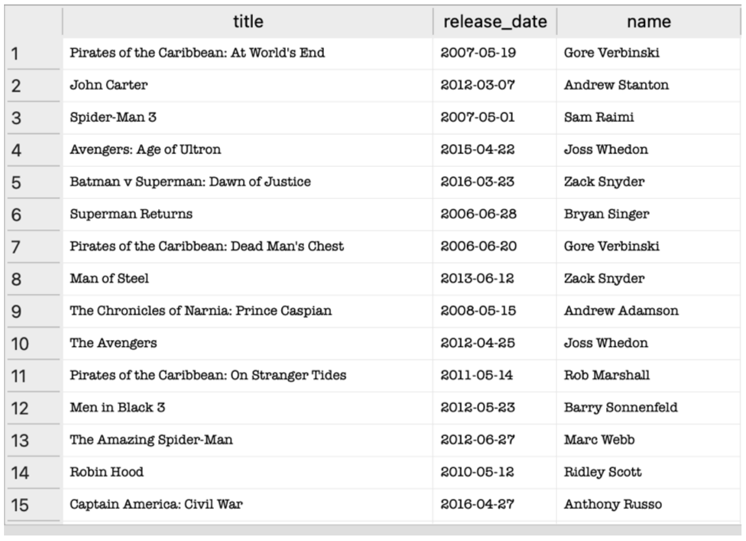
- 

	year	total_revenue
1	2012	24141710246
2	2014	24120490589
3	2013	23411493295
4	2015	22775024221
5	2009	21072651506
6	2011	20516921160
7	2010	20348574768
8	2008	18145379803
9	2006	16635892750
10	2007	16491621655
11	2004	16278686567
12	2005	14931589218
13	2002	14609857556
14	2016	14461156948
15	2003	14211646728

#3. Movies released in the first half of the year

- Code
  - 

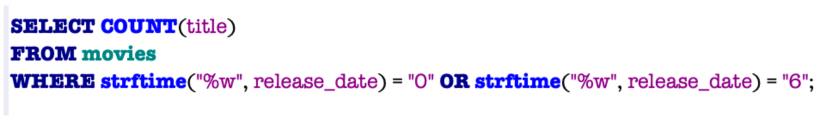
```
SELECT m.title, m.release_date, d.name
FROM movies as m
JOIN directors as d
ON m.director_id=d.id
WHERE strftime("%m", m.release_date) BETWEEN "01" AND "06";
```

- Output
  - 

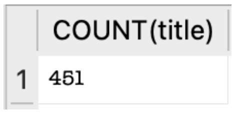
	title	release_date	name
1	Pirates of the Caribbean: At World's End	2007-05-19	Gore Verbinski
2	John Carter	2012-03-07	Andrew Stanton
3	Spider-Man 3	2007-05-01	Sam Raimi
4	Avengers: Age of Ultron	2015-04-22	Joss Whedon
5	Batman v Superman: Dawn of Justice	2016-03-23	Zack Snyder
6	Superman Returns	2006-06-28	Bryan Singer
7	Pirates of the Caribbean: Dead Man's Chest	2006-06-20	Gore Verbinski
8	Man of Steel	2013-06-12	Zack Snyder
9	The Chronicles of Narnia: Prince Caspian	2008-05-15	Andrew Adamson
10	The Avengers	2012-04-25	Joss Whedon
11	Pirates of the Caribbean: On Stranger Tides	2011-05-14	Rob Marshall
12	Men in Black 3	2012-05-23	Barry Sonnenfeld
13	The Amazing Spider-Man	2012-06-27	Marc Webb
14	Robin Hood	2010-05-12	Ridley Scott
15	Captain America: Civil War	2016-04-27	Anthony Russo

#4. Number of movies released on weekends

- Code

- 

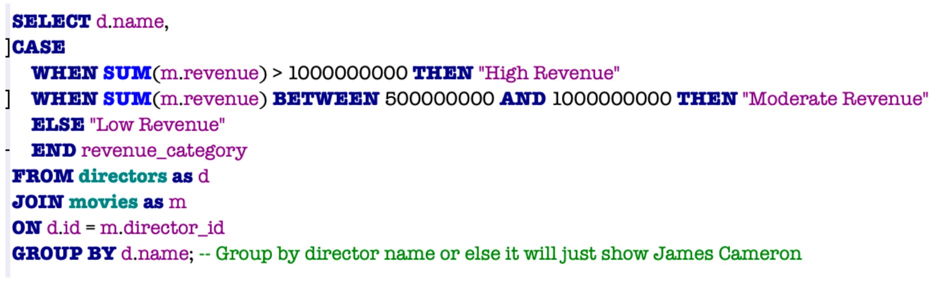
```sql
SELECT COUNT(title)
FROM movies
WHERE strftime("%w", release_date) = "0" OR strftime("%w", release_date) = "6";
```

- Output
  - 

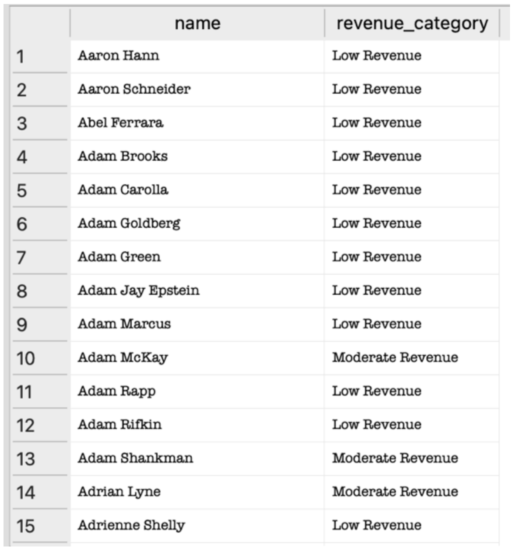
	COUNT(title)
1	451

CASE STATEMENT

#5. Categorize directors based on total revenue

- Code
  - 

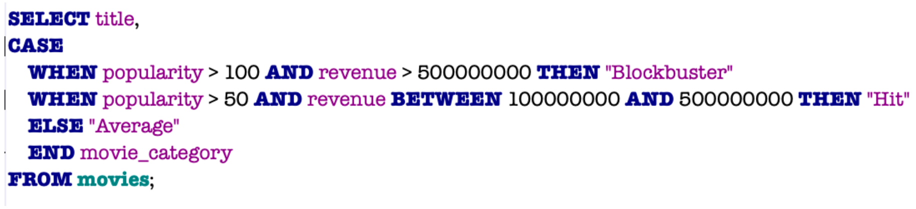
```sql
SELECT d.name,
CASE
  WHEN SUM(m.revenue) > 1000000000 THEN "High Revenue"
  WHEN SUM(m.revenue) BETWEEN 500000000 AND 1000000000 THEN "Moderate Revenue"
  ELSE "Low Revenue"
  END revenue_category
FROM directors as d
JOIN movies as m
ON d.id = m.director_id
GROUP BY d.name; -- Group by director name or else it will just show James Cameron
```

- Output
  - 

	name	revenue_category
1	Aaron Hann	Low Revenue
2	Aaron Schneider	Low Revenue
3	Abel Ferrara	Low Revenue
4	Adam Brooks	Low Revenue
5	Adam Carolla	Low Revenue
6	Adam Goldberg	Low Revenue
7	Adam Green	Low Revenue
8	Adam Jay Epstein	Low Revenue
9	Adam Marcus	Low Revenue
10	Adam McKay	Moderate Revenue
11	Adam Rapp	Low Revenue
12	Adam Rifkin	Low Revenue
13	Adam Shankman	Moderate Revenue
14	Adrian Lyne	Moderate Revenue
15	Adrienne Shelly	Low Revenue

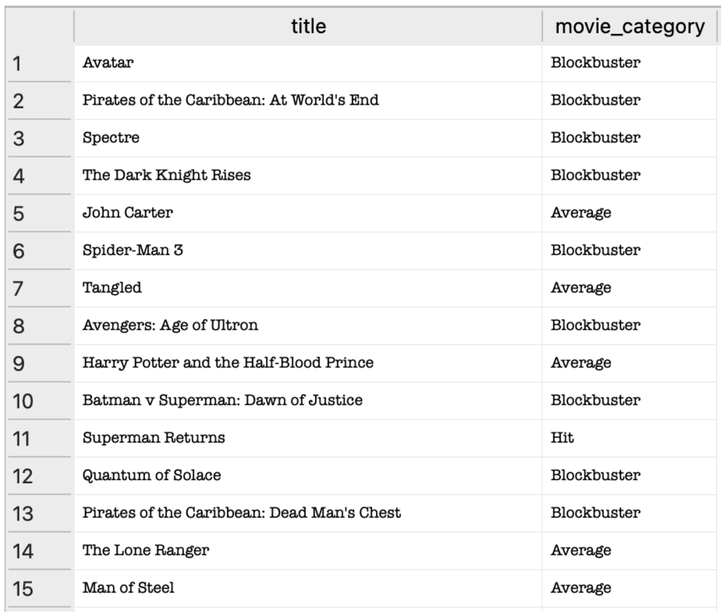
#6. Movies based on popularity and revenue

- Code



```
SELECT title,
CASE
   WHEN popularity > 100 AND revenue > 500000000 THEN "Blockbuster"
   WHEN popularity > 50 AND revenue BETWEEN 100000000 AND 500000000 THEN "Hit"
   ELSE "Average"
   END movie_category
FROM movies;
```

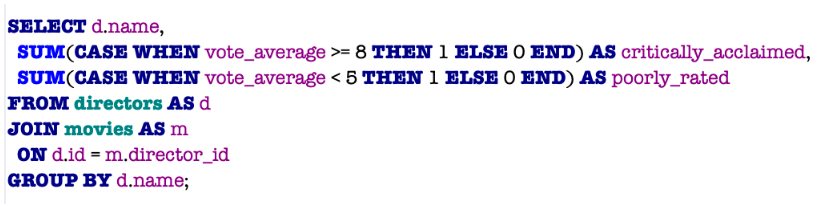
- Output



	title	movie_category
1	Avatar	Blockbuster
2	Pirates of the Caribbean: At World's End	Blockbuster
3	Spectre	Blockbuster
4	The Dark Knight Rises	Blockbuster
5	John Carter	Average
6	Spider-Man 3	Blockbuster
7	Tangled	Average
8	Avengers: Age of Ultron	Blockbuster
9	Harry Potter and the Half-Blood Prince	Average
10	Batman v Superman: Dawn of Justice	Blockbuster
11	Superman Returns	Hit
12	Quantum of Solace	Blockbuster
13	Pirates of the Caribbean: Dead Man's Chest	Blockbuster
14	The Lone Ranger	Average
15	Man of Steel	Average

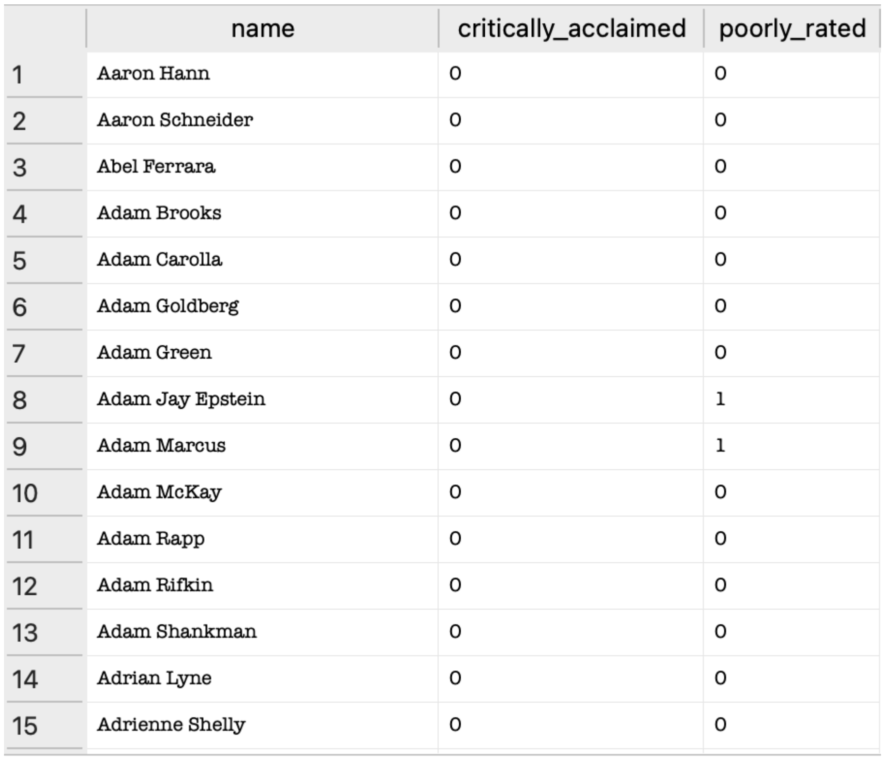
#7. Directors with critically acclaimed and poorly rated movies

- Code



```
SELECT d.name,
 SUM(CASE WHEN vote_average >= 8 THEN 1 ELSE 0 END) AS critically_acclaimed,
 SUM(CASE WHEN vote_average < 5 THEN 1 ELSE 0 END) AS poorly_rated
FROM directors AS d
JOIN movies AS m
 ON d.id = m.director_id
GROUP BY d.name;
```

- Output



	name	critically_acclaimed	poorly_rated
1	Aaron Hann	0	0
2	Aaron Schneider	0	0
3	Abel Ferrara	0	0
4	Adam Brooks	0	0
5	Adam Carolla	0	0
6	Adam Goldberg	0	0
7	Adam Green	0	0
8	Adam Jay Epstein	0	1
9	Adam Marcus	0	1
10	Adam McKay	0	0
11	Adam Rapp	0	0
12	Adam Rifkin	0	0
13	Adam Shankman	0	0
14	Adrian Lyne	0	0
15	Adrienne Shelly	0	0