

Sorting results

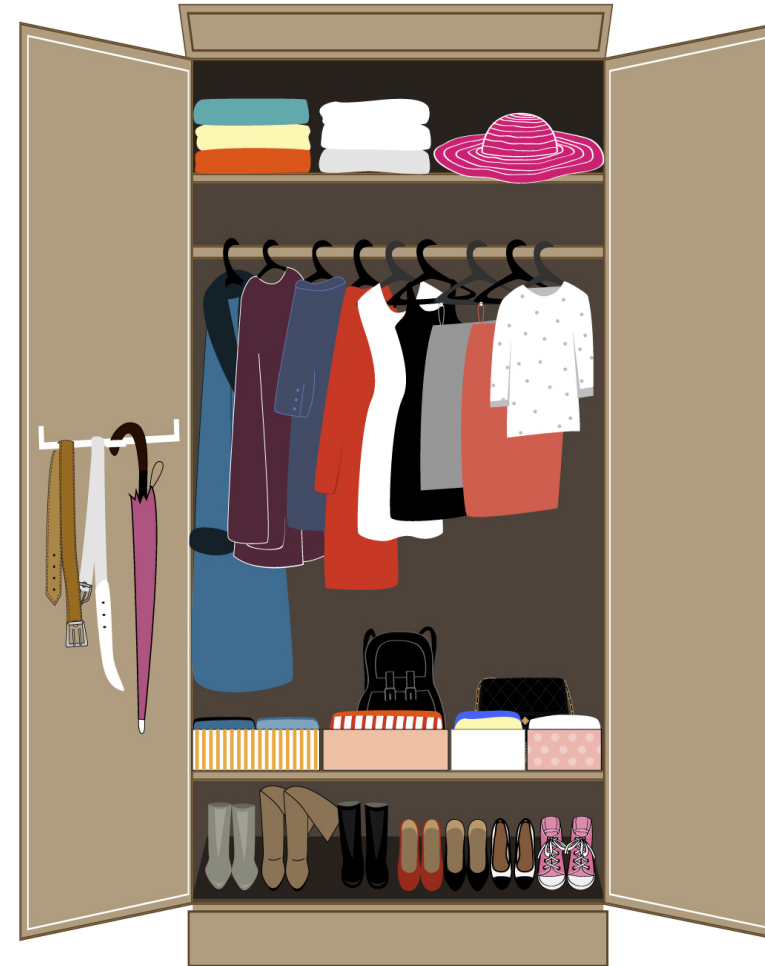
INTERMEDIATE SQL



Jasmin Ludolf

Data Science Content Developer,
DataCamp

Sorting results



ORDER BY

```
SELECT title, budget
FROM films
ORDER BY budget;
```

title	budget
-----	-----
Tarnation	218
My Date with Drew	1100
A Plague So Pleasant	1400
The Mongol King	3250
...	

```
SELECT title, budget
FROM films
ORDER BY title;
```

title	budget
-----	-----
#Horror	1500000
10 Cloverfield Lane	15000000
10 Days in a Madhouse	12000000
10 Things I Hate About You	16000000
...	

ASCending

```
SELECT title, budget
FROM films
ORDER BY budget ASC;
```

```
|title                |budget|
|-----|-----|
|Tarnation            |218   |
|My Date with Drew    |1100  |
|A Plague So Pleasant|1400  |
|The Mongol King      |3250  |
... 
```

DESCending

```
SELECT title, budget
FROM films
ORDER BY budget DESC;
```

```
|title                                     |budget|
|-----|-----|
|Love and Death on Long Island |null  |
|The Chambermaid on the Titanic|null  |
|51 Birch Street                |null  |
...
```

```
SELECT title, budget
FROM films
WHERE budget IS NOT NULL
ORDER BY budget DESC;
```

```
|title                                     |budget      |
|-----|-----|
|The Host                                |12215500000 |
|Lady Vengeance                         |42000000000 |
...
```

Sorting fields

```
SELECT title
FROM films
ORDER BY release_year;
```

```
|title|
|-----|
|Intolerance: Love's Struggle Throu...|
|Over the Hill to the Poorhouse|
|The Big Parade|
|Metropolis|
|...|
```

```
SELECT title, release_year
FROM films
ORDER BY release_year;
```

```
|title|release_year|
|-----|-----|
|Intolerance: Love's S...|1916|
|Over the Hill to the ...|1920|
|The Big Parade|1925|
|Metropolis|1927|
|...|
```

ORDER BY multiple fields

- `ORDER BY field_one, field_two`

```
SELECT title, wins
FROM best_movies
ORDER BY wins DESC;
```

title	wins
Lord of the Rings:Return of t...	11
Titanic	11
Ben-Hur	11

- Think of `field_two` as a tie-breaker

```
SELECT title, wins, imdb_score
FROM best_movies
ORDER BY wins DESC, imdb_score DESC;
```

title	wins	imdb_score
Lord of the Rings:...	11	9
Ben-Hur	11	8.1
Titanic	11	7.9

Different orders

```
SELECT birthdate, name
FROM people
ORDER BY birthdate, name DESC;
```

```
|birthdate |name      |
|-----|-----|
|1990-01-01|Robert Brown|
|1990-02-02|Anne Smith  |
|1991-05-14|Amy Miller  |
|1991-11-22|Adam Waters |
...

```


Order of execution

```
-- Written code:  
SELECT item  
FROM coats  
WHERE color = `yellow`  
ORDER BY length  
LIMIT 3;
```

```
-- Order of execution:  
SELECT item  
FROM coats  
WHERE color = `yellow`  
ORDER BY length  
LIMIT 3;
```

Let's practice!
INTERMEDIATE SQL

Grouping data

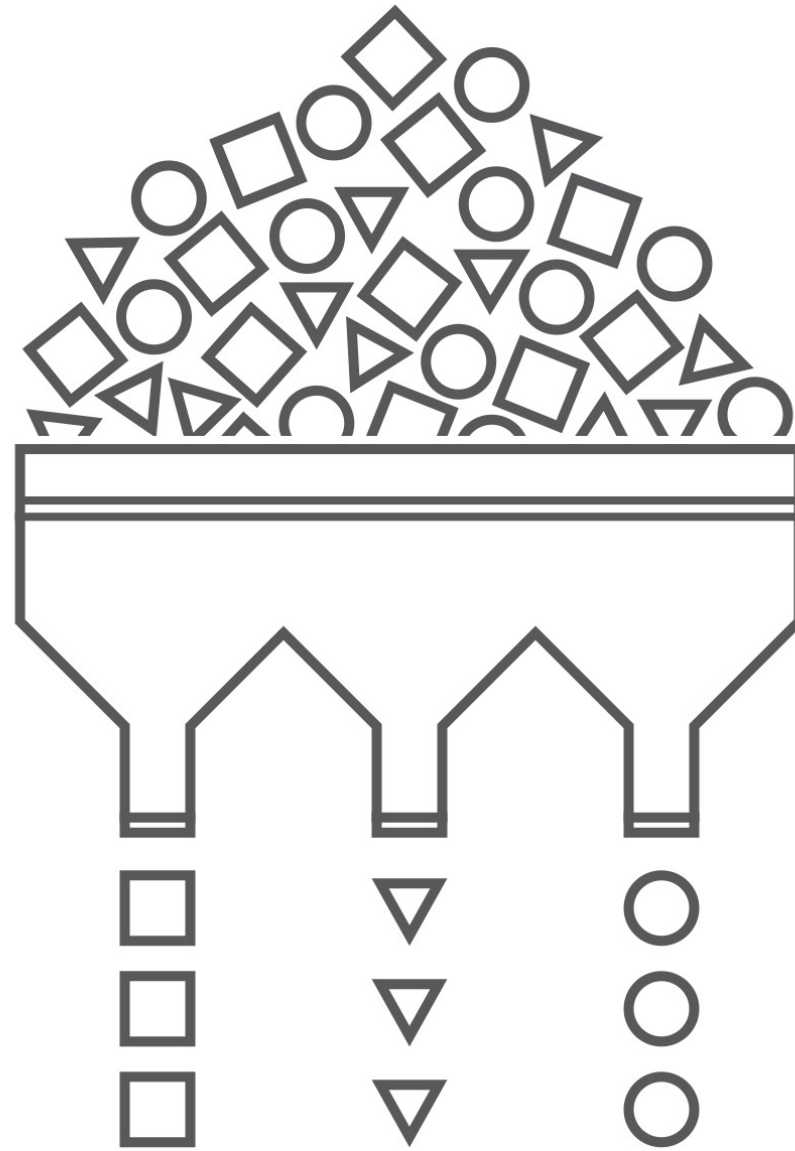
INTERMEDIATE SQL



Jasmin Ludolf

Data Science Content Developer,
DataCamp

Grouping data



GROUP BY single fields

```
SELECT certification, COUNT(title) AS title_count
FROM films
GROUP BY certification;
```

```
|certification|title_count|
|-----|-----|
|Unrated     |62         |
|M           |5          |
|G           |112        |
|NC-17       |7          |
... 
```

Error handling

```
SELECT certification, title
FROM films
GROUP BY certification;
```

column "films.title" must appear in the
GROUP BY clause or be used in an
aggregate function

```
LINE 1: SELECT certification, title
                        ^
```

```
SELECT
    certification,
    COUNT(title) AS count_title
FROM films
GROUP BY certification;
```

certification	count_title
Unrated	62
M	5
G	112
...	

GROUP BY multiple fields

```
SELECT certification, language, COUNT(title) AS title_count
FROM films
GROUP BY certification, language;
```

```
|certification|language |title_count|
|-----|-----|-----|
|null        |null     |5         |
|Unrated     |Japanese |2         |
|R           |Norwegian|2         |
... 
```

GROUP BY with ORDER BY

```
SELECT
    certification,
    COUNT(title) AS title_count
FROM films
GROUP BY certification;
```

```
|certification|title_count|
|-----|-----|
|Unrated      |62         |
|M            |5          |
|G            |112        |
|...          |           |
```

```
SELECT
    certification,
    COUNT(title) AS title_count
FROM films
GROUP BY certification
ORDER BY title_count DESC;
```

```
|certification|title_count|
|-----|-----|
|R            |2118       |
|PG-13       |1462       |
|...         |           |
```


Order of execution

-- Written code:

```
SELECT
    certification,
    COUNT(title) AS title_count
FROM films
GROUP BY certification
ORDER BY title_count DESC
LIMIT 3;
```

-- Order of execution:

```
SELECT
    certification,
    COUNT(title) AS title_count
FROM films
GROUP BY certification
ORDER BY title_count DESC
LIMIT 3;
```

Let's practice!
INTERMEDIATE SQL

Filtering grouped data

INTERMEDIATE SQL



Jasmin Ludolf

Data Science Content Developer,
DataCamp

HAVING

```
SELECT
    release_year,
    COUNT(title) AS title_count
FROM films
GROUP BY release_year
WHERE COUNT(title) > 10;
```

```
syntax error at or near "WHERE"
LINE 4: WHERE COUNT(title) > 10;
      ^
```

```
SELECT
    release_year,
    COUNT(title) AS title_count
FROM films
GROUP BY release_year
HAVING COUNT(title) > 10;
```

release_year	title_count
1988	31
null	42
2008	225
...	

Order of execution

-- Written code:

```
SELECT
    certification,
    COUNT(title) AS title_count
FROM films
WHERE certification
    IN ('G', 'PG', 'PG-13')
GROUP BY certification
HAVING COUNT(title) > 500
ORDER BY title_count DESC
LIMIT 3;
```

-- Order of execution:

```
SELECT
    certification,
    COUNT(title) AS title_count
FROM films
WHERE certification
    IN ('G', 'PG', 'PG-13')
GROUP BY certification
HAVING COUNT(title) > 500
ORDER BY title_count DESC
LIMIT 3;
```

HAVING vs WHERE

- WHERE filters individual records, HAVING filters grouped records
- What films were released in the year 2000?

```
SELECT title
FROM films
WHERE release_year = 2000;
```

```
|title          |
|-----|
|102 Dalmatians|
|28 Days       |
|...           |
```

- In what years was the average film duration over two hours?

HAVING vs WHERE

- In what years was the average film duration over two hours?

```
SELECT release_year
FROM films
GROUP BY release_year
HAVING AVG(duration) > 120;
```

```
|release_year|
|-----|
|1954        |
|1959        |
|...        |
```

Let's practice!
INTERMEDIATE SQL

Congratulations!

INTERMEDIATE SQL



Jasmin Ludolf

Data Science Content Developer,
DataCamp

What you've learned

- Chapter 1: Selecting with `COUNT()` , `LIMIT`
- Chapter 2: Filtering with `WHERE` , `BETWEEN` , `AND` , `OR` , `LIKE` , `NOT LIKE` , `IN` , `%` , `_` , `IS NULL` , `IS NOT NULL`
- Chapter 3: `ROUND()` and aggregate functions
- Chapter 4: Sorting and grouping with `ORDER BY` , `DESC` , `GROUP BY` , `HAVING`
- Comparison operators
- Arithmetic

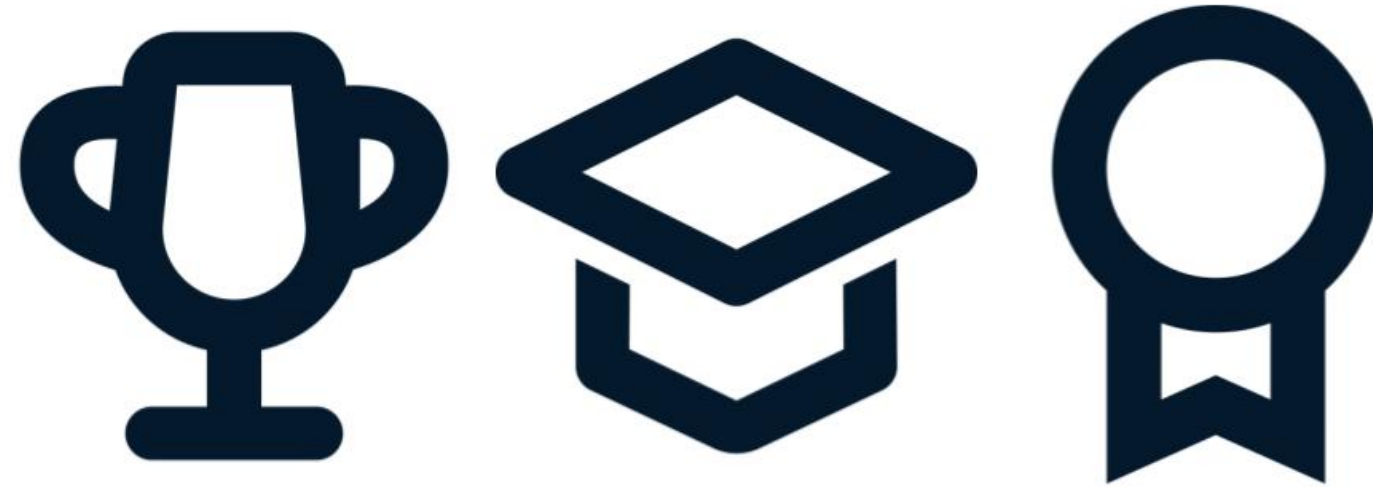
Skills

- Error handling
- Debugging
- Writing readable code
- Selecting data
- Querying data
- Filtering and summarizing data
- Sorting and grouping data



What's next?

- DataCamp SQL joining course
- DataCamp Project, Practice, Competitions
- Workspace



Congratulations!

INTERMEDIATE SQL