INF4401 - SQL 2

Julien Romero - Télécom SudParis

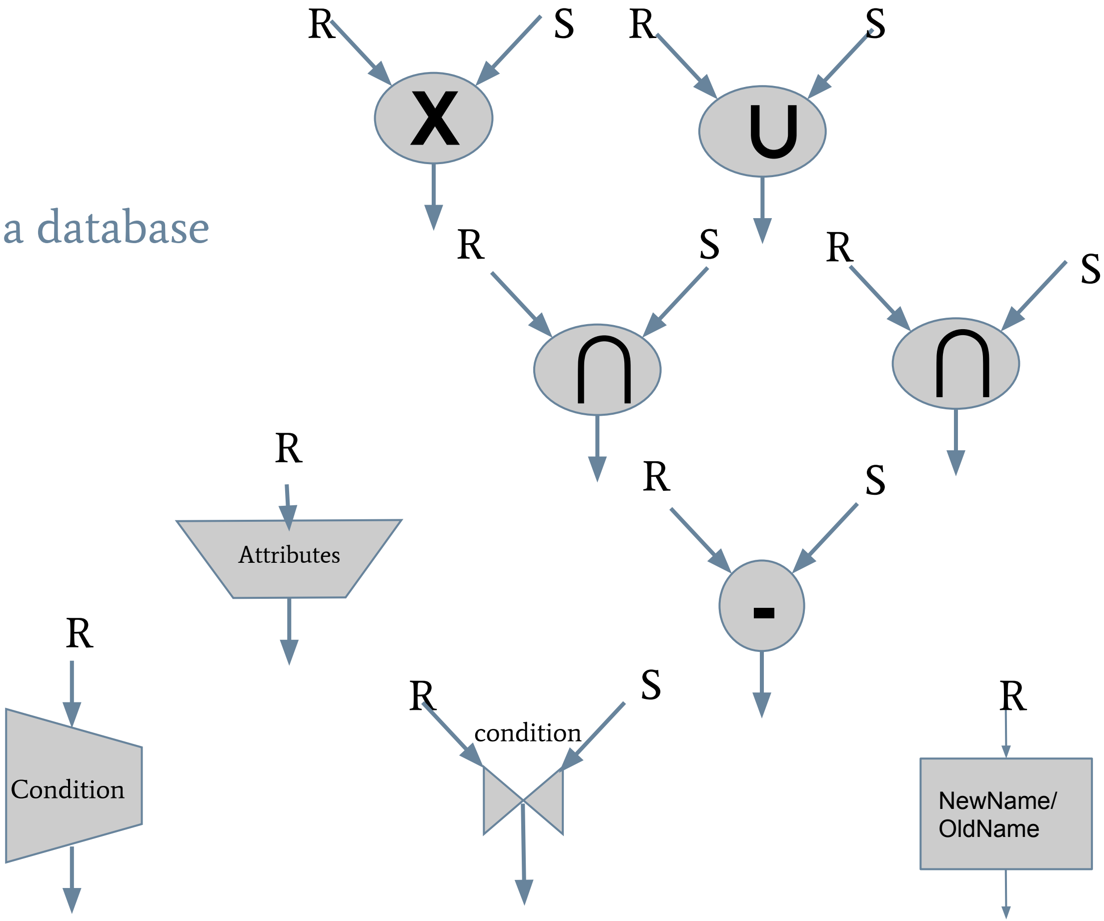
Previously...

The Relational Model

- We organize data into **tables** composed of **columns** (attributes) and **rows** (tuples)
- Each column has a name and a **domain**
- The content of a table is called a **relation**
- A **key** identifies uniquely a row
- A **relation schema** is a description of a table
- A **database schema** is the description of all the tables in the database

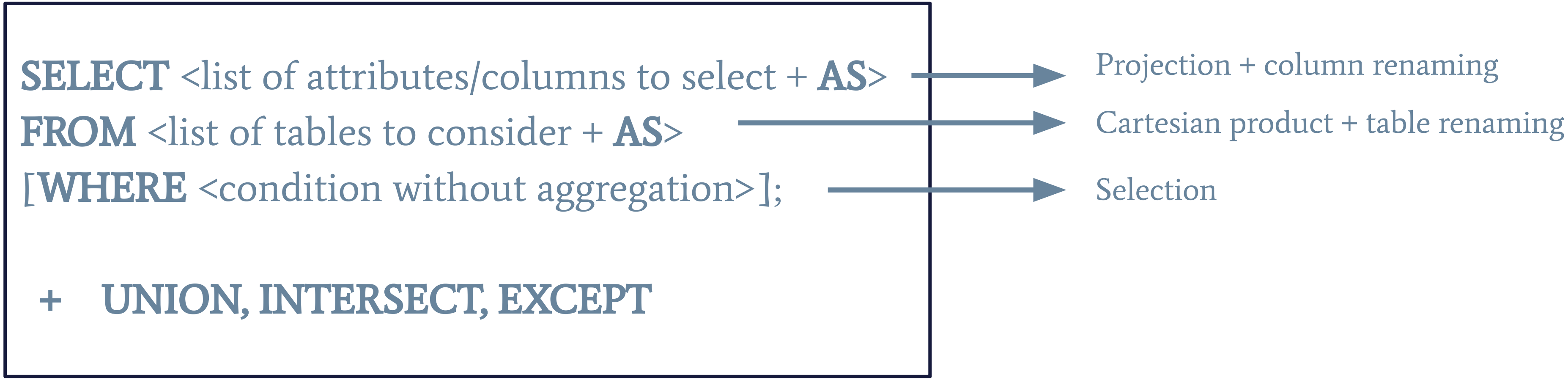
Relational Algebra

- A formalism to query data in a database
- Operations:
  - Project
  - Select
  - Product
  - Union, Intersection, Difference
  - Renaming
  - Join
- Join = Product + Select



SQL

- SQL is a programming language that implement relational algebra (but does more, see today lecture)



SELECT <list of attributes/columns to select + **AS**>
FROM <list of tables to consider + **AS**>
[**WHERE** <condition without aggregation>];

+ **UNION, INTERSECT, EXCEPT**

Projection + column renaming

Cartesian product + table renaming

Selection

SQL JOIN

- We saw that a join is a cartesian product followed by a selection. We were able to reproduce it with a FROM and a WHERE.
  - SELECT * FROM T1, ..., Tn WHERE <join condition>;
- This is hard to read and understand, especially when we have several tables, several join conditions, and normal conditions.
- SQL gives us two easy keywords for the join:
  - **JOIN** <table to join> **ON** <join condition> between **FROM** and **WHERE**

SQL JOIN

```
SELECT *
FROM Table1, Table2, Table3
WHERE join_condition1 AND join_condition2
  AND normal conditions
```

Is the same as

```
SELECT *
FROM Table1
JOIN Table2
ON join_condition1
JOIN Table3
ON join_condition2
WHERE normal conditions
```

SQL Query So Far

```
SELECT <list of attributes/columns to select>
FROM <list of tables to consider>
[JOIN <table>
 [ON <join condition>]]*
[WHERE <condition without aggregation>];
```

([ ] means the content is optional, * means we can repeat several time the content)

SQL JOIN - Example - Old Solution

- Get all movies titles from Damien Chazelle and their notes.

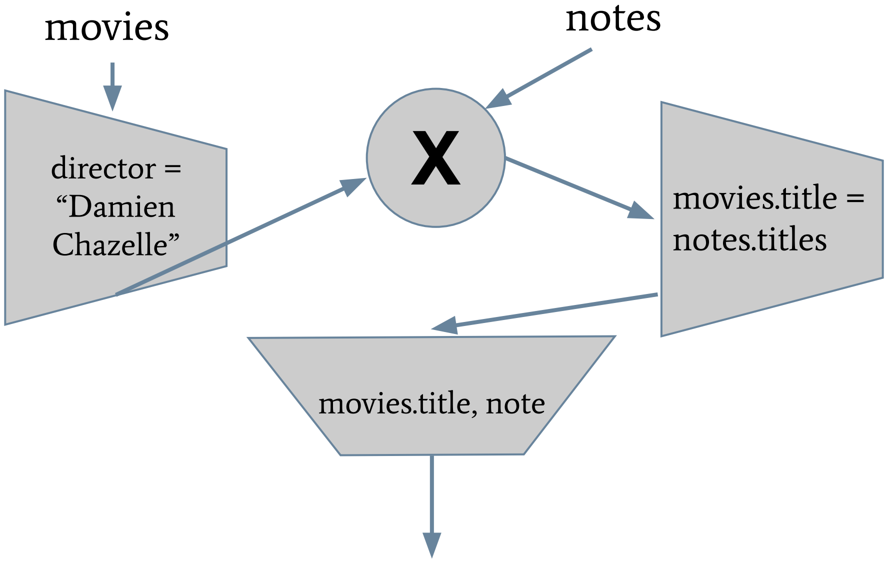


Table movies

title	director	year
Avatar	James Cameron	2009
The Grand Budapest Hotel	Wes Anderson	2014
Whiplash	Damien Chazelle	2014
Gran Torino	Clint Eastwood	2008
The Godfather	Francis Ford Coppola	1972
Moonrise Kingdom	Wes Anderson	2012

Table notes

title	note
Avatar	9
Whiplash	10
Whiplash	8

SQL JOIN - Example - Old Solution

- Get all movies titles from Damien Chazelle and their notes.

SELECT movies.title, note
FROM movies, notes
WHERE director = “Damien Chazelle” **AND** movies.title = notes.title;

Table movies

title	director	year
Avatar	James Cameron	2009
The Grand Budapest Hotel	Wes Anderson	2014
Whiplash	Damien Chazelle	2014
Gran Torino	Clint Eastwood	2008
The Godfather	Francis Ford Coppola	1972
Moonrise Kingdom	Wes Anderson	2012

Table notes

title	note
Avatar	9
Whiplash	10
Whiplash	8

SQL JOIN - Example - New Solution

- Get all movies titles from Damien Chazelle and their notes.

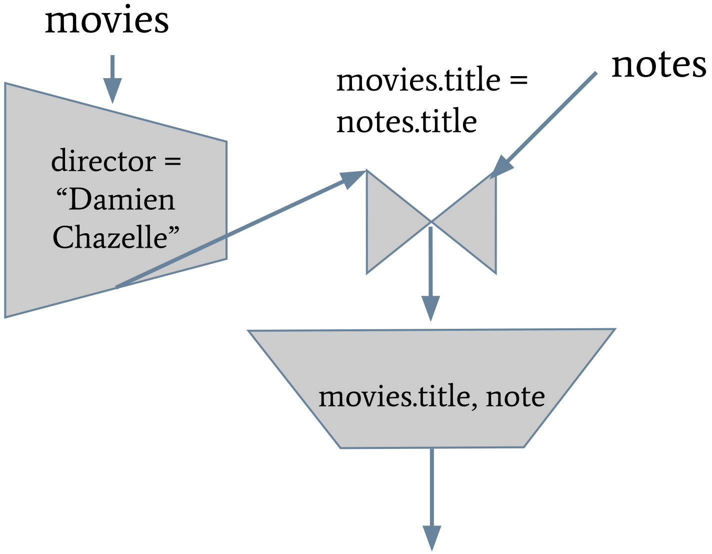


Table movies

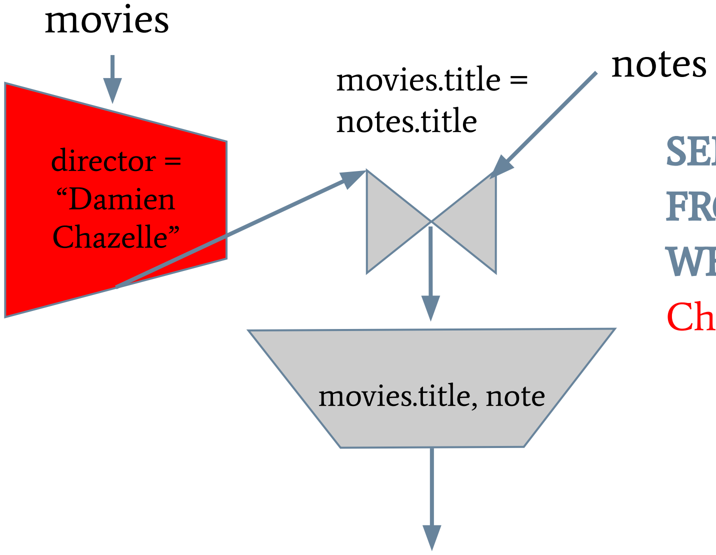
title	director	year
Avatar	James Cameron	2009
The Grand Budapest Hotel	Wes Anderson	2014
Whiplash	Damien Chazelle	2014
Gran Torino	Clint Eastwood	2008
The Godfather	Francis Ford Coppola	1972
Moonrise Kingdom	Wes Anderson	2012

Table notes

title	note
Avatar	9
Whiplash	10
Whiplash	8

SQL JOIN - Example - New Solution

- Get all movies titles from Damien Chazelle and their notes.



```
SELECT *
FROM movies
WHERE director = “Damien
Chazelle”;
```

Table movies

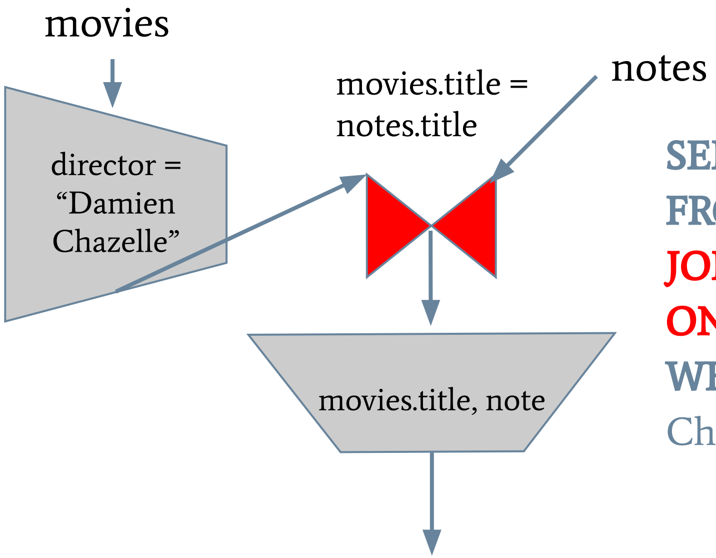
title	director	year
Avatar	James Cameron	2009
The Grand Budapest Hotel	Wes Anderson	2014
Whiplash	Damien Chazelle	2014
Gran Torino	Clint Eastwood	2008
The Godfather	Francis Ford Coppola	1972
Moonrise Kingdom	Wes Anderson	2012

Table notes

title	note
Avatar	9
Whiplash	10
Whiplash	8

SQL JOIN - Example - New Solution

- Get all movies titles from Damien Chazelle and their notes.



movies

director = “Damien Chazelle”

movies.title = notes.title

notes

movies.title, note

```
SELECT *
FROM movies
JOIN notes
ON movies.title = notes.title
WHERE director = “Damien
Chazelle”;
```

Table movies

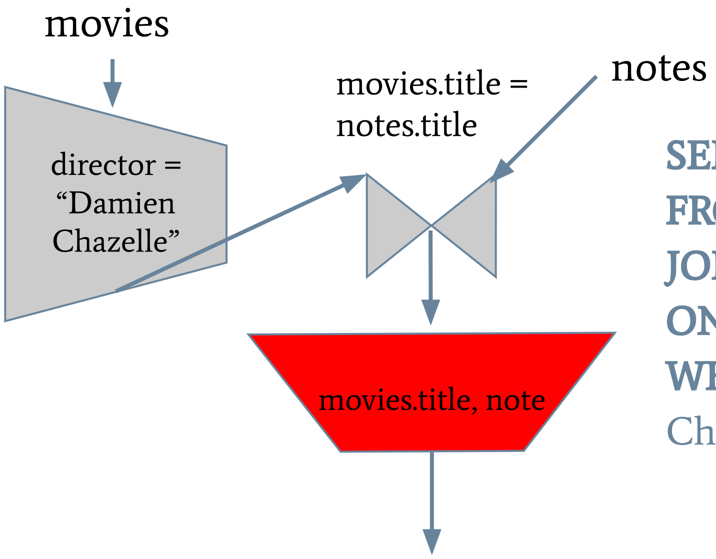
title	director	year
Avatar	James Cameron	2009
The Grand Budapest Hotel	Wes Anderson	2014
Whiplash	Damien Chazelle	2014
Gran Torino	Clint Eastwood	2008
The Godfather	Francis Ford Coppola	1972
Moonrise Kingdom	Wes Anderson	2012

Table notes

title	note
Avatar	9
Whiplash	10
Whiplash	8

SQL JOIN - Example - New Solution

- Get all movies titles from Damien Chazelle and their notes.



```
SELECT movies.title, note
FROM movies
JOIN notes
ON movies.title = notes.title
WHERE director = “Damien
Chazelle”;
```

Table movies

title	director	year
Avatar	James Cameron	2009
The Grand Budapest Hotel	Wes Anderson	2014
Whiplash	Damien Chazelle	2014
Gran Torino	Clint Eastwood	2008
The Godfather	Francis Ford Coppola	1972
Moonrise Kingdom	Wes Anderson	2012

Table notes

title	note
Avatar	9
Whiplash	10
Whiplash	8

SQL JOIN - Example - New Solution

- Get all movies titles from Damien Chazelle and their notes.

```
SELECT movies.title, note
FROM movies
JOIN notes
ON movies.title = notes.title
WHERE director = “Damien Chazelle”;
```

Table movies

title	director	year
Avatar	James Cameron	2009
The Grand Budapest Hotel	Wes Anderson	2014
Whiplash	Damien Chazelle	2014
Gran Torino	Clint Eastwood	2008
The Godfather	Francis Ford Coppola	1972
Moonrise Kingdom	Wes Anderson	2012

Table notes

title	note
Avatar	9
Whiplash	10
Whiplash	8

SQL - Aggregation Functions

The aggregation functions transform a list of values into a single one. SQL contains **AVG** (average), **COUNT** (number of lines, often combined with **DISTINCT**), **MAX** (maximum), **MIN** (minimum), **SUM**

These operations are not supported by relational algebra!

SQL - Aggregation Functions - COUNT

- How many movies do we have?

```
SELECT COUNT(title)
FROM movies;
```

Or

```
SELECT COUNT(*)
FROM movies;
```

COUNT(title)
6

Table movies

title	director	year
Avatar	James Cameron	2009
The Grand Budapest Hotel	Wes Anderson	2014
Whiplash	Damien Chazelle	2014
Gran Torino	Clint Eastwood	2008
The Godfather	Francis Ford Coppola	1972
Moonrise Kingdom	Wes Anderson	2012

SQL - Aggregation Functions - COUNT

- How many directors do we have?

```
SELECT COUNT(director)
FROM movies;
```

COUNT(title)
6

Wes Anderson is counted twice!

Table movies

title	director	year
Avatar	James Cameron	2009
The Grand Budapest Hotel	Wes Anderson	2014
Whiplash	Damien Chazelle	2014
Gran Torino	Clint Eastwood	2008
The Godfather	Francis Ford Coppola	1972
Moonrise Kingdom	Wes Anderson	2012

SQL - Aggregation Functions - COUNT

- How many directors do we have?

```
SELECT COUNT(DISTINCT director)
FROM movies;
```

COUNT(title)
5

Table movies

title	director	year
Avatar	James Cameron	2009
The Grand Budapest Hotel	Wes Anderson	2014
Whiplash	Damien Chazelle	2014
Gran Torino	Clint Eastwood	2008
The Godfather	Francis Ford Coppola	1972
Moonrise Kingdom	Wes Anderson	2012

SQL - Aggregation Functions - AVG

Table notes

title	note
Avatar	9
Whiplash	10
Whiplash	8

What is the average note of a movie?

```
SELECT AVG(note)
FROM notes;
```

AVG(note)
9

SQL - Aggregation Functions - MAX/MIN

What is the minimum/maximum note of a movie?

```
SELECT MIN(note)
FROM notes;
```

MIN(note)
8

```
SELECT MAX(note)
FROM notes;
```

MAX(note)
10

Table notes

title	note
Avatar	9
Whiplash	10
Whiplash	8

SQL - Aggregation Functions - SUM

- How much money does the film industry generate?

```
SELECT SUM(income)
FROM cinemas;
```

SUM(income)
510

Table cinema

cinemaID	title	income
1	Avatar	100
2	The Grand Budapest Hotel	50
3	Whiplash	80
4	Whiplash	70
1	Whiplash	60
2	Avatar	150

SQL - Aggregation Functions - Remarks

- It is often useful to rename an aggregation to give it a meaningful name.

```
SELECT SUM(income) AS total_income
FROM cinemas;
```

total_income
510

- In a simple SFW, it is **not possible** to mix a normal column with an aggregation!

```
SELECT cinemaID, SUM(income)
FROM cinemas;
```

6 rows

1 row

cinemaID	SUM(income)
???	510

Grouping

We still would like to have both non-aggregate and aggregate attributes together. For example, to answer the query:

- How much money did each movie generate?

What we have to do is group the line together if they are from the same movie, and then apply an aggregation function for each group.

Grouping - Example

cinemaID	title	income
1	Avatar	100
2	The Grand Budapest Hotel	50
3	Whiplash	80
4	Whiplash	70
1	Whiplash	60
2	Avatar	150

Group the row by title

cinemaID	title	income
1	Avatar	100
2		50
3	Whiplash	80
4		70
1		60
2	The Grand Budapest Hotel	150

Group "Avatar"

Group "Whiplash"

Group "The Grand Budapest Hotel"

Grouping - Example

cinemaID	title	income
1	Avatar	100
2		50
3	Whiplash	80
4		70
1		60
2	The Grand Budapest Hotel	150

Aggregation for each group

title	income
Avatar	150
Whiplash	210
The Grand Budapest Hotel	150

We cannot keep the column cinemaID here!

SQL GROUP BY

- In SQL, we can perform the grouping by using the keyword **GROUP BY** after the **WHERE**.
- If our query contains a **GROUP BY**, then we work with groups and not single rows!
    - We can only print columns mentioned in the group by or aggregation of columns
- New SQL query:

```
SELECT <list of attributes/columns to select>
FROM <list of tables to consider>
[JOIN <table>
 [ON <join condition>]]*
[WHERE <condition without aggregation>]
[GROUP BY <list of columns used for grouping>];
```

SQL GROUP BY - Example 1

- How much money did each movie generate?

```
SELECT title, SUM(income)
FROM cinemas
GROUP BY title;
```

SQL GROUP BY - Example 2

- How many movies did each director directed?

```
SELECT director, COUNT(title)
FROM movies
GROUP BY director;
```

Table movies

title	director
Cry Macho	Clint Eastwood
The Grand Budapest Hotel	Wes Anderson
Whiplash	Damien Chazelle
Gran Torino	Clint Eastwood
The Godfather	Francis Ford Coppola
Moonrise Kingdom	Wes Anderson

SQL GROUP BY - Example 2

title	director
Cry Macho	Clint Eastwood
The Grand Budapest Hotel	Wes Anderson
Whiplash	Damien Chazelle
Gran Torino	Clint Eastwood
The Godfather	Francis Ford Coppola
Moonrise Kingdom	Wes Anderson

title	director
Cry Macho	Clint Eastwood
Gran Torino	
Whiplash	Damien Chazelle
The Godfather	Francis Ford Coppola
The Grand Budapest Hotel	Wes Anderson
Moonrise Kingdom	

COUNT(title)	director
2	Clint Eastwood
1	Damien Chazelle
1	Francis Ford Coppola
2	Wes Anderson

SQL - Conditions On Groups

- We might need to filter the groups just like we previously filtered rows
  - E.g.: Give the directors who directed more than 2 movies.
- These condition use aggregation functions:
  - “Give the directors who directed more than 2 movies.” -> COUNT(title) >= 2
- Conditions with aggregations must come after the keyword **HAVING**
- **HAVING** comes after **GROUP BY**
- **WHERE** and **HAVING** have the same meaning but:
  - **WHERE** is for conditions **without** aggregation
  - **HAVING** is for conditions **with** aggregations

SQL HAVING

```
SELECT <list of attributes/columns to select>
FROM <list of tables to consider>
[JOIN <table>
 [ON <join condition>]]*
[WHERE <condition without aggregation>]
[GROUP BY <list of columns used for grouping>
[HAVING <condition with aggregation>]];
```

Careful! The order of the commands is important! In the correct order, we have **SELECT**, **FROM**, **WHERE**, **GROUP BY**, **HAVING** (**HAVING** always preceded by **GROUP BY**)

SQL HAVING - Example 1

- Give the directors who directed more than 2 movies.

```
SELECT director
FROM movies
GROUP BY director
HAVING COUNT(title) >= 2;
```

director
Clint Eastwood
Wes Anderson

Table movies

title	director
Cry Macho	Clint Eastwood
The Grand Budapest Hotel	Wes Anderson
Whiplash	Damien Chazelle
Gran Torino	Clint Eastwood
The Godfather	Francis Ford Coppola
Moonrise Kingdom	Wes Anderson

SQL HAVING - Example 2

Give the directors who directed more than 2 movies before 2015.

```
SELECT director
FROM movies
WHERE year < 2015
GROUP BY director
HAVING COUNT(title) >= 2;
```

Table movies

title	director	year
Cry Macho	Clint Eastwood	2021
The Grand Budapest Hotel	Wes Anderson	2014
Whiplash	Damien Chazelle	2014
Gran Torino	Clint Eastwood	2008
The Godfather	Francis Ford Coppola	1972
Moonrise Kingdom	Wes Anderson	2012

SQL HAVING - Example 2

title	director	year
Cry Macho	Clint Eastwood	2021
The Grand Budapest Hotel	Wes Anderson	2014
Whiplash	Damien Chazelle	2014
Gran Torino	Clint Eastwood	2008
The Godfather	Francis Ford Coppola	1972
Moonrise Kingdom	Wes Anderson	2012

WHERE year < 2015 →

title	director	year
The Grand Budapest Hotel	Wes Anderson	2014
Whiplash	Damien Chazelle	2014
Gran Torino	Clint Eastwood	2008
The Godfather	Francis Ford Coppola	1972
Moonrise Kingdom	Wes Anderson	2012

SQL HAVING - Example 2

title	director	year
The Grand Budapest Hotel	Wes Anderson	2014
Whiplash	Damien Chazelle	2014
Gran Torino	Clint Eastwood	2008
The Godfather	Francis Ford Coppola	1972
Moonrise Kingdom	Wes Anderson	2012

GROUP BY director

title	director	year
The Grand Budapest Hotel	Wes Anderson	2014
Moonrise Kingdom		2012
Whiplash	Damien Chazelle	2014
Gran Torino	Clint Eastwood	2008
The Godfather	Francis Ford Coppola	1972

SQL HAVING - Example 2

title	director	year
The Grand Budapest Hotel	Wes Anderson	2014
Moonrise Kingdom	Wes Anderson	2012
Whiplash	Damien Chazelle	2014
Gran Torino	Clint Eastwood	2008
The Godfather	Francis Ford Coppola	1972

HAVING COUNT(title) >= 2 →

title	director	year
The Grand Budapest Hotel	Wes Anderson	2014
Moonrise Kingdom		2012

SQL HAVING - Example 2

title	director	year
The Grand Budapest Hotel	Wes Anderson	2014
Moonrise Kingdom		2012

SELECT director

director
Wes Anderson

SQL - Ordering The Results

It is often convenient to sort the results according to some columns. We can use the keyword **ORDER BY** at the end of our query.
ASC = ascending order, **DESC** = descending order

SELECT A1, ..., An
FROM T
ORDER BY Ai DESC/ASC, Aj DESC/ASC

title	note
Whiplash	10
Avatar	9
Lalaland	8
Whiplash	8

- We can ask to order according to several column, in case there is an equality.
  - If there is an equality with the first column mentioned, we use the second column
- Example:
  - **SELECT** title, note **FROM** notes **ORDER BY** note DESC, title ASC;

SQL - Limiting The Results

- We can limit the number of rows returned by using the keyword **LIMIT** followed by the number of rows we want. **LIMIT** comes at the end of the query.
- E.g.: Get the oldest movie in the database.

```
SELECT title
FROM movies
ORDER BY year ASC
LIMIT 1
```

SQL - Limiting The Results

title	director	year
Cry Macho	Clint Eastwood	2021
The Grand Budapest Hotel	Wes Anderson	2014
Whiplash	Damien Chazelle	2014
Gran Torino	Clint Eastwood	2008
The Godfather	Francis Ford Coppola	1972
Moonrise Kingdom	Wes Anderson	2012

ORDER BY year **ASC** →

title	director	year
The Godfather	Francis Ford Coppola	1972
Gran Torino	Clint Eastwood	2008
Moonrise Kingdom	Wes Anderson	2012
The Grand Budapest Hotel	Wes Anderson	2014
Whiplash	Damien Chazelle	2014
Cry Macho	Clint Eastwood	2021

SQL - Limiting The Results

title	director	year
The Godfather	Francis Ford Coppola	1972
Gran Torino	Clint Eastwood	2008
Moonrise Kingdom	Wes Anderson	2012
The Grand Budapest Hotel	Wes Anderson	2014
Whiplash	Damien Chazelle	2014
Cry Macho	Clint Eastwood	2021

LIMIT 1

title	director	year
The Godfather	Francis Ford Coppola	1972

Summary

```
SELECT <list of attributes/columns to select>
FROM <list of tables to consider>
[JOIN <table>
 [ON <join condition>]]*
[WHERE <condition without aggregation>]
[GROUP BY <list of columns used for grouping>
[HAVING <condition with aggregation>]]
[ORDER BY <list of column + ASC or DESC>]
[LIMIT <number of rows>];
```