Aggregation Functions

- Aggregate functions group multiple rows together to form a single summary value

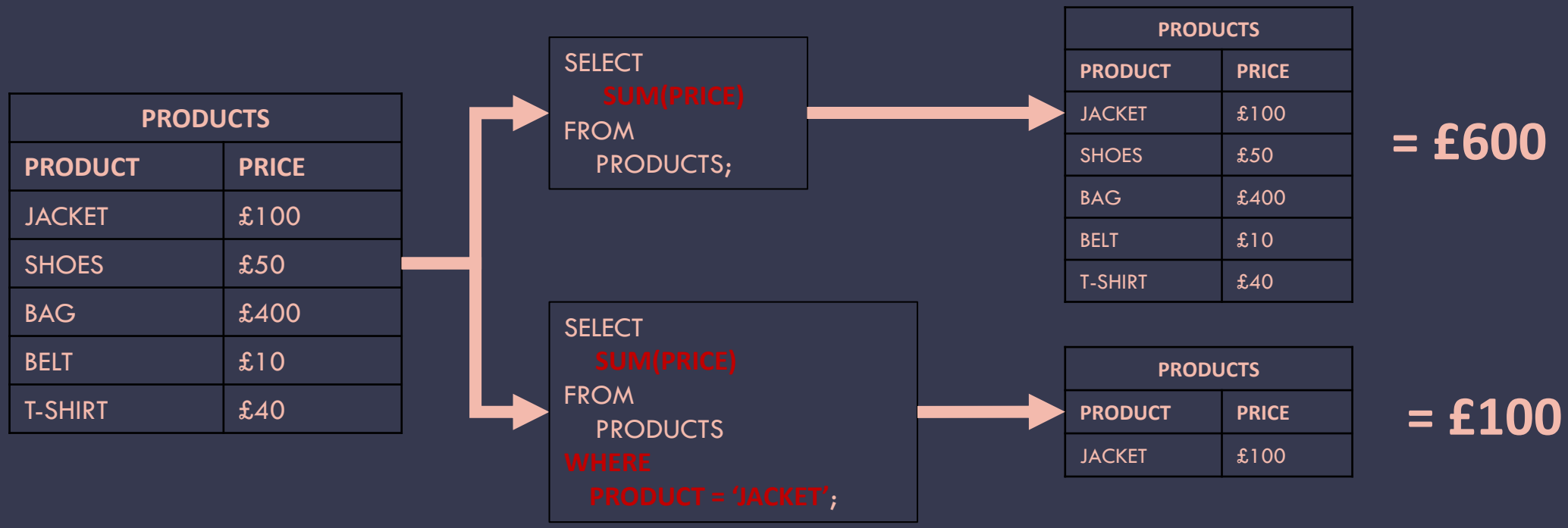
- Not the same as row level functions
  - Row level operations are calculated for each row of the data
  - Aggregate operations are first aggregated and then calculated

Aggregation Functions

FUNCTION	DESCRIPTION
SUM	Calculates the sum of the given attribute/expression in the group defined
COUNT	Counts the number of elements in the group defined
AVG	Calculates the average value of the given attribute/expression in the group defined.
MIN	Finds the minimum in the group defined.
MAX	Finds the maximum in the group defined.

Aggregation Functions & Where Clause

- The Where clause restricts the rows to which the aggregation function is applied



Aggregation Functions & Group By

- The **GROUP** BY statement **groups** rows that have the same values into summary rows
- The GROUP BY statement is used with aggregate functions
- Columns in the select statement that are not included in the group by clause should be aggregated

```
SELECT
        COLUMN NAME(S)
FROM
        TABLE_NAME
WHERE CONDITION
GROUP BY COLUMN_NAME(S);
```

Single Column Group By

IMDB_TOP10			
NAME	**CATEGORY**	**NETWORK**	**RATING**
Breaking Bad	Drama	AMC	9.5
Game of Thrones	Adventure	HBO	9.3
The Wire	Drama	HBO	9.3
Rick and Morty	Animation	Adult Swim	9.2
Avatar: The Last Airbender	Animation	Nickelodeon	9.2
The Sopranos	Drama	HBO	9.2
Sherlock	Drama	BBC	9.1
Fullmetal Alchemist: Brotherhood	Animation	JNN	9.1
Death Note	Animation	Nippon TV	9.0
Firefly	Sci-Fi	Fox	9.0

```
SELECT
        CATEGORY,
        AVG(RATING)
FROM
        IMDB_TOP10
GROUP BY CATEGORY;
```

NETWORK	AVG(RATING)
Drama	9.275
Adventure	9.300
Animation	9.125
Sci-Fi	9.000

Multiple Column Group By

IMDB_TOP10			
NAME	**CATEGORY**	**NETWORK**	**RATING**
Breaking Bad	Drama	AMC	9.5
Game of Thrones	Adventure	HBO	9.3
The Wire	Drama	HBO	9.3
Rick and Morty	Animation	Adult Swim	9.2
Avatar: The Last Airbender	Animation	Nickelodeon	9.2
The Sopranos	Drama	HBO	9.2
Sherlock	Drama	BBC	9.1
Fullmetal Alchemist: Brotherhood	Animation	JNN	9.1
Death Note	Animation	Nippon TV	9.0
Firefly	Sci-Fi	Fox	9.0

```
SELECT
        CATEGORY,
        NETWORK,
        AVG(RATING)
FROM
        IMBD_TOP10
GROUP BY CATEGORY, NETWORK;
```

CATEGORY	NETWORK	RATING
Adventure	HBO	9.3
Animation	Adult Swim	9.2
Animation	Nickelodeon	9.2
Animation	JNN	9.1
Animation	Nippon TV	9
Drama	AMC	9.5
Drama	HBO	9.25
Drama	BBC	9.1
Sci-Fi	Fox	9

Selecting Non-Aggregated Fields That Are Not In Group By Clause

IMDB_TOP10			
NAME	**CATEGORY**	**NETWORK**	**RATING**
Breaking Bad	Drama	AMC	9.5
Game of Thrones	Adventure	HBO	9.3
The Wire	Drama	HBO	9.3
Rick and Morty	Animation	Adult Swim	9.2
Avatar: The Last Airbender	Animation	Nickelodeon	9.2
The Sopranos	Drama	HBO	9.2
Sherlock	Drama	BBC	9.1
Fullmetal Alchemist: Brotherhood	Animation	JNN	9.1
Death Note	Animation	Nippon TV	9.0
Firefly	Sci-Fi	Fox	9.0

```
SELECT
        CATEGORY,
        NETWORK
        AVG(RATING)
FROM
        IMBD_TOP10
GROUP BY NETWORK;
```

- What to do with the category field?
- Do I aggregate, if so what aggregation do I apply?
- Should I group by category as well?

Group By On Numerical Fields

- Doesn't make sense to group by a numerical field such as salary as there will be too many distinct fields

- When applying a group by to such a numerical field it would make sense to classify the salaries into buckets i.e.
  - Salary <1000 is "LOW"
  - Salary between 1000 and 2000 is "MEDIUM"
  - Salary > 2000 is "HIGH

Group By & Having

- The where clause is applied to all rows of the data and the having clause is applied to aggregated data after the group by clause
- The having clause was introduced to SQL because the where clause can not be applied to aggregated tables

```
SELECT
        COLIUMN_NAME(S)
FROM
        TABLE_NAME
WHERE CONDITION
GROUP BY COLUMN_NAME(S)
HAVING CONDITION;
```