Day 3

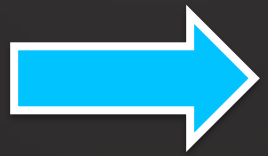
Aggregation functions

- ✓ **Aggregate values in multiple rows to one value**

Data Output | Explain

	amount numeric (5,2)
1	1.99
2	0.99
3	6.99
4	0.99
5	4.99
6	2.99

SUM



Data Output | Expl

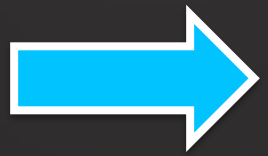
	sum numeric
1	67416.51

Aggregation functions

- ✓ **Aggregate values in multiple rows to one value**

Data Output | Explain

	amount numeric (5,2)
1	1.99
2	0.99
3	6.99
4	0.99
5	4.99
6	2.99



Data Output | Expla

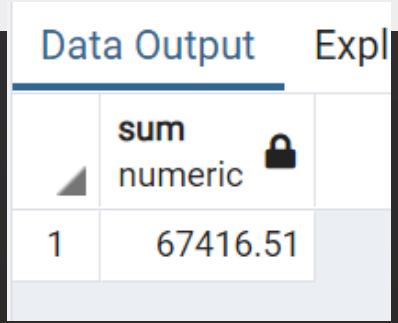
	avg numeric
1	4.20

Most common aggregation functions

SUM()
AVG()
MIN()
MAX()
COUNT()

SYNTAX

```
SELECT
SUM(amount)
FROM payment
```

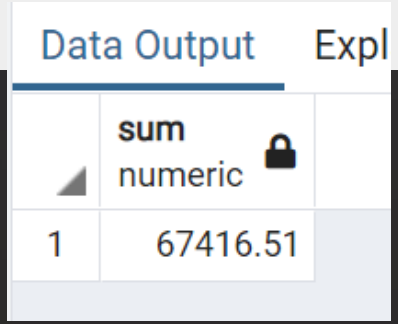


SYNTAX

```
SELECT
COUNT(*)
FROM payment
```

What we can't do...

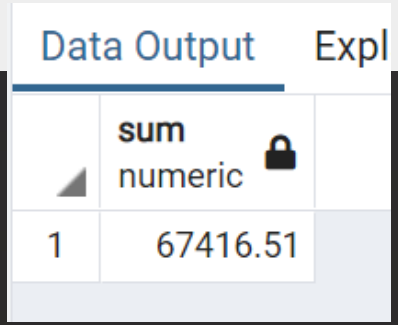
```
SELECT
SUM(amount)
payment_id
FROM payment
```



No mixing possible!

```
SELECT
SUM(amount)
payment_id
FROM payment
```

Only possible with grouping!



Multiple aggregations is possible!

```
SELECT
SUM(amount),
COUNT(*),
AVG(amount)
FROM payment
```

Data Output | Explain | Messages

	sum numeric	count bigint	avg numeric
1	67416.51	16049	4.20

Challenge

Challenge

Your manager wants to which of the two employees (staff_id) is responsible for more payments?

Which of the two is responsible for a higher overall payment amount?

How do these amounts change if we don't consider amounts equal to 0?

Write two SQL queries to get the answers!

Result

Data Output | Explain | Messages

	staff_id smallint	sum numeric	count bigint
1	2	33927.04	7992
2	1	33489.47	8057

Data Output | Explain | Messages

	staff_id smallint	sum numeric	count bigint
1	2	33927.04	7983
2	1	33489.47	8042

Solution

```
SELECT
MIN(replacement_cost),
MAX(replacement_cost),
ROUND(AVG(replacement_cost),2) AS AVG,
SUM(replacement_cost)
FROM film
```

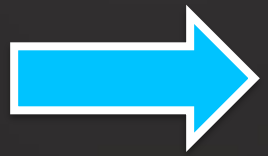
GROUP BY

✓ **Used to GROUP aggregations BY specific columns**

Data Output | Explain | Messages | No

	customer_id smallint	amount numeric (5,2)
1	269	1.99
2	269	0.99
3	269	6.99
4	269	0.99
5	269	4.99
6	269	2.99
7	270	1.99
8	270	4.99

SUM



Data Output | Explain | Messa

	customer_id smallint	sum numeric
1	1	118.68
2	2	128.73
3	3	135.74
4	4	81.78
5	5	144.62

SYNTAX

```
SELECT
customer_id,
SUM(amount)
FROM payment
GROUP BY customer_id
```

Data Output | Explain | Messa

	customer_id smallint	sum numeric
1	1	118.68
2	2	128.73
3	3	135.74
4	4	81.78
5	5	144.62

SYNTAX

```
SELECT
customer_id,
SUM(amount)
FROM payment
WHERE customer_id >3
GROUP BY customer_id
```

Data Output | Explain | Messa

	customer_id smallint	sum numeric
1	1	118.68
2	2	128.73
3	3	135.74
4	4	81.78
5	5	144.62

SYNTAX

```
SELECT
customer_id,
SUM(amount)
FROM payment
WHERE customer_id >3
GROUP BY customer_id
ORDER BY customer_id
```

Data Output | Explain | Messa

	customer_id smallint	sum numeric
1	1	118.68
2	2	128.73
3	3	135.74
4	4	81.78
5	5	144.62

SYNTAX

```
SELECT
customer_id,
SUM(amount)
FROM payment
GROUP BY customer_id
```

Every column:
In GROUP BY or
in aggregate functions

SYNTAX

```
SELECT
customer_id,
SUM(amount)
FROM payment
GROUP BY customer_id
```

Every column:
In GROUP BY or
in aggregate functions

Challenge

Challenge

There are two competitions between the two employees.

Which employee had the highest sales amount in a single day?

Which employee had the most sales in a single day (not counting payments with amount = 0?

Write two SQL queries to get the answers!

Result

Data Output | Explain | Messages | Notificati

	date date	staff_id smallint	sum numeric	count bigint
1	2020-04-30	2	2866.42	658
2	2020-04-30	1	2736.75	625
3	2020-03-21	2	1505.52	348

Challenge

Challenge

Your manager wants to get a better understanding of the films.
That's why you are asked to write a query to see the

- Minimum
- Maximum
- Average (rounded)
- Sum

of the replacement cost of the films.

Write a SQL query to get the answers!

Result

Data Output | Explain | Messages | Notifications

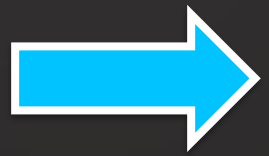
	min numeric	max numeric	avg numeric	sum numeric
1	9.99	29.99	19.98	19984.00

HAVING

✓ Used to FILTER Groupings BY aggregations

	staff_id smallint	date date	sum numeric	count bigint
1	2	2020-04-30	2866.42	658
2	1	2020-04-30	2736.75	625
3	2	2020-03-21	1505.52	348
4	1	2020-03-01	143	
5	2	2020-03-19	140	

HAVING COUNT(*)>400



	staff_id smallint	date date	sum numeric	count bigint
1	2	2020-04-30	2866.42	658
2	1	2020-04-30	2736.75	625

Note!

HAVING an only be used with GROUP BY!

SYNTAX

```
SELECT
customer_id,
SUM(amount)
FROM payment
GROUP BY customer_id
HAVING SUM(amount)>200
```

Data Output | Explain | Mes

	customer_id smallint	sum numeric
1	526	221.55
2	148	216.54

Solution

```
SELECT
MIN(replacement_cost),
MAX(replacement_cost),
ROUND(AVG(replacement_cost),2) AS AVG,
SUM(replacement_cost)
FROM film
```

Challenge

Challenge

In 2020, April 28, 29 and 30 were days with very high revenue. That's why we want to focus in this task only on these days (filter accordingly).

Find out what is the average payment amount grouped by customer and day – consider only the days/customers with more than 1 payment (per customer and day).
Order by the average amount in a descending order.

Write a SQL query to find out!

Result

Data Output | Explain | Messages | Notifications

	customer_id smallint	date date	avg_amount numeric	count bigint
1	459	2020-04-29	10.49	2
2	443	2020-04-28	9.49	2
3	510	2020-04-28	9.49	2