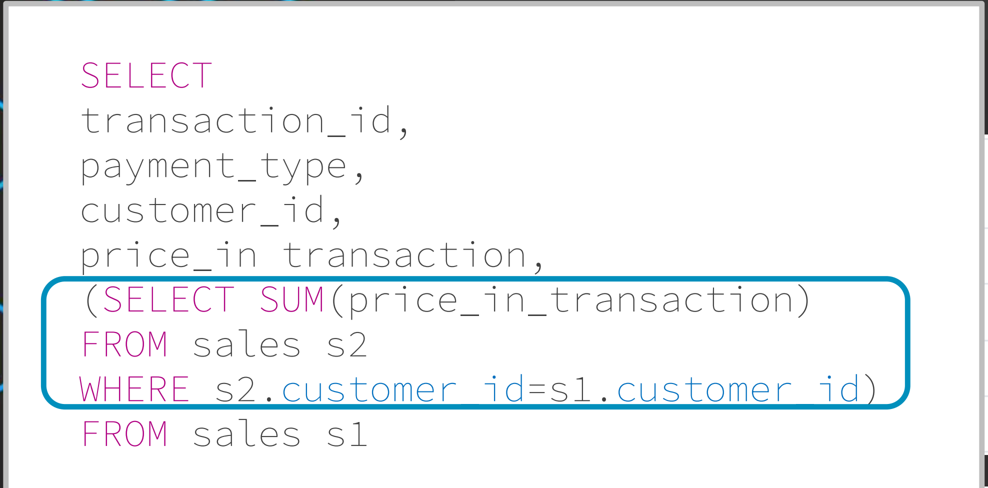
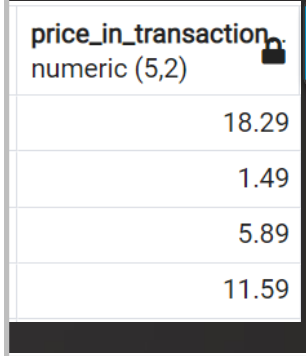
Window functions



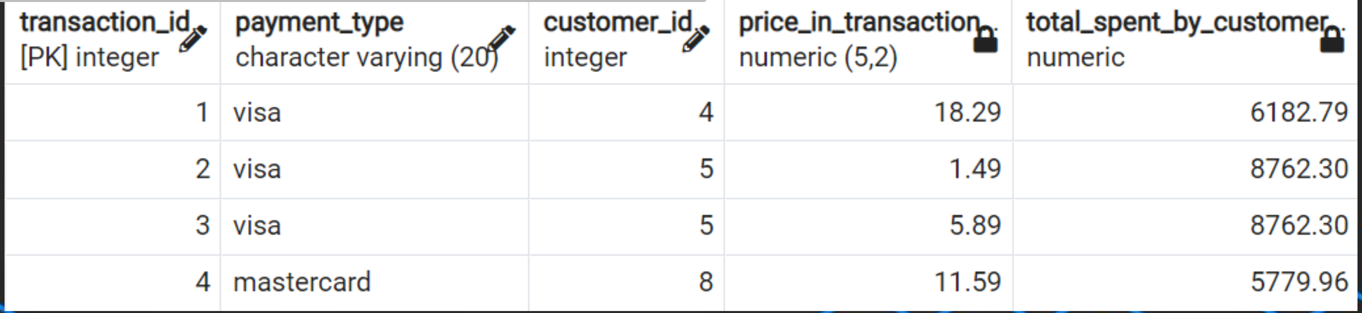
```
SELECT
transaction_id,
payment_type,
customer_id,
price_in_transaction,
(SELECT SUM(price_in_transaction)
FROM sales s2
WHERE s2.customer_id=s1.customer_id)
FROM sales s1
```



price_in_transaction numeric (5,2)
18.29
1.49
5.89
11.59

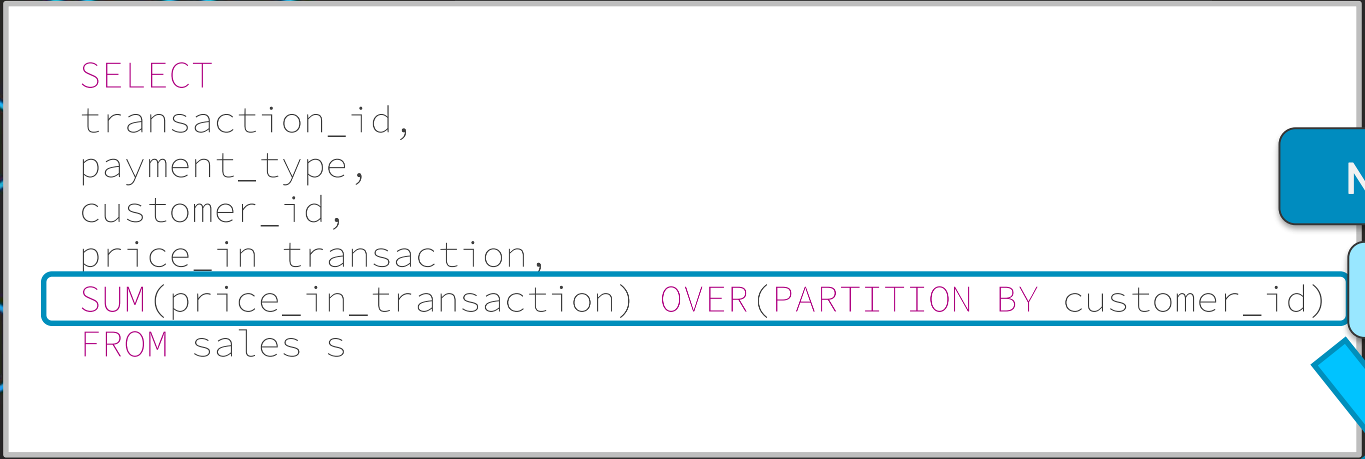
No. of rows not affected

Aggregated by



transaction_id [PK] integer	payment_type character varying (20)	customer_id integer	price_in_transaction numeric (5,2)	total_spent_by_customer numeric
1	visa	4	18.29	6182.79
2	visa	5	1.49	8762.30
3	visa	5	5.89	8762.30
4	mastercard	8	11.59	5779.96

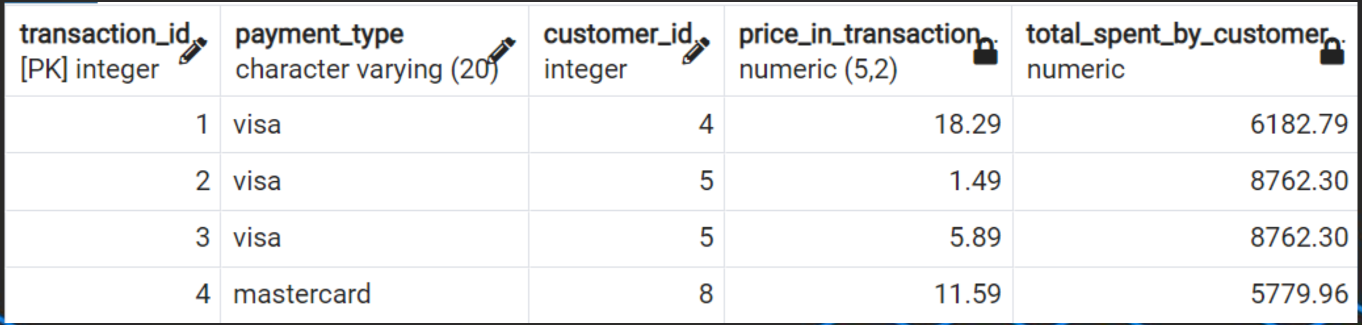
Window functions



```
SELECT
transaction_id,
payment_type,
customer_id,
price_in_transaction,
SUM(price_in_transaction) OVER(PARTITION BY customer_id)
FROM sales s
```

No. of rows not affected

Aggregated by



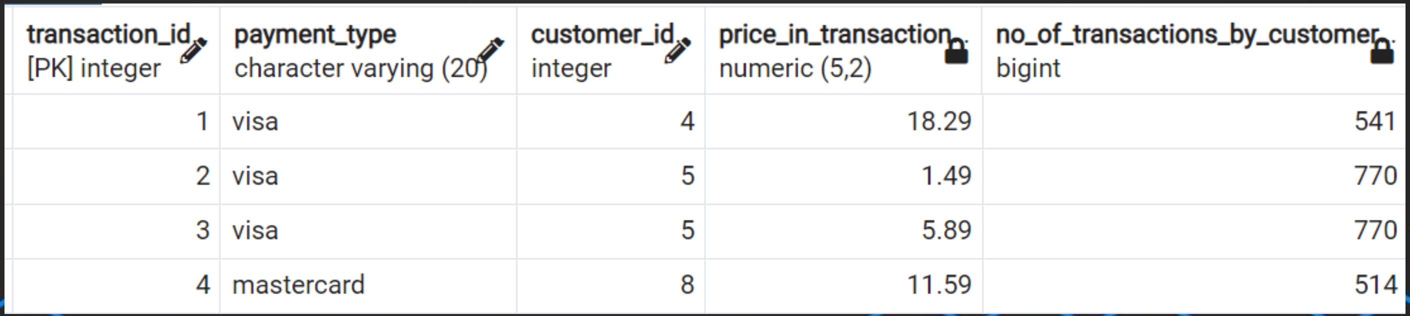
transaction_id [PK] integer	payment_type character varying (20)	customer_id integer	price_in_transaction numeric (5,2)	total_spent_by_customer numeric
1	visa	4	18.29	6182.79
2	visa	5	1.49	8762.30
3	visa	5	5.89	8762.30
4	mastercard	8	11.59	5779.96

Window functions

```
SELECT
transaction_id,
payment_type,
customer_id,
price_in_transaction,
COUNT(*) OVER(PARTITION BY customer_id)
FROM sales s
```

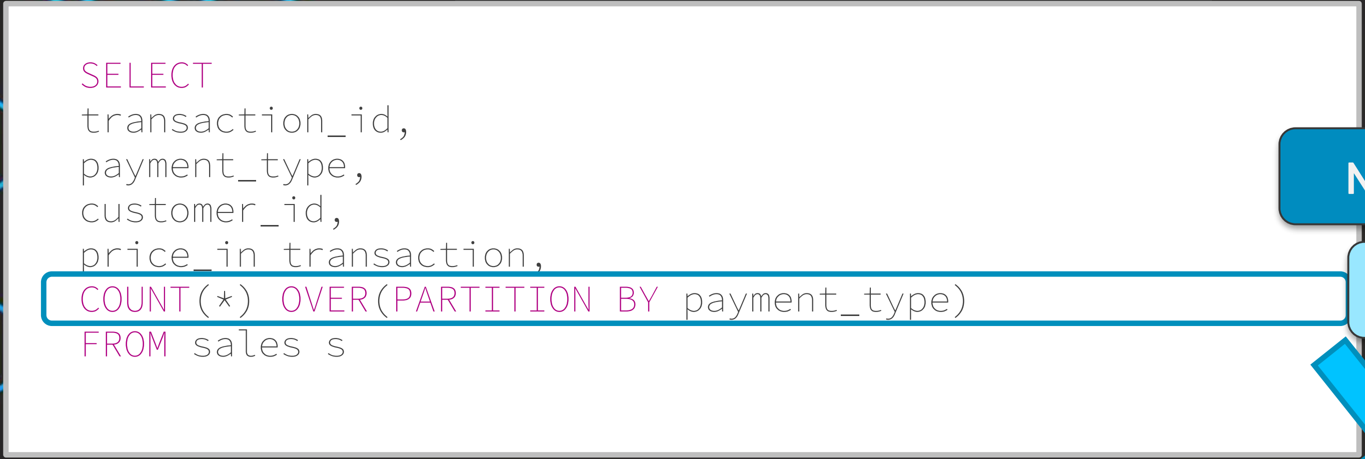
No. of rows not affected

Aggregated by



transaction_id [PK] integer	payment_type character varying (20)	customer_id integer	price_in_transaction numeric (5,2)	no_of_transactions_by_customer bigint
1	visa	4	18.29	541
2	visa	5	1.49	770
3	visa	5	5.89	770
4	mastercard	8	11.59	514

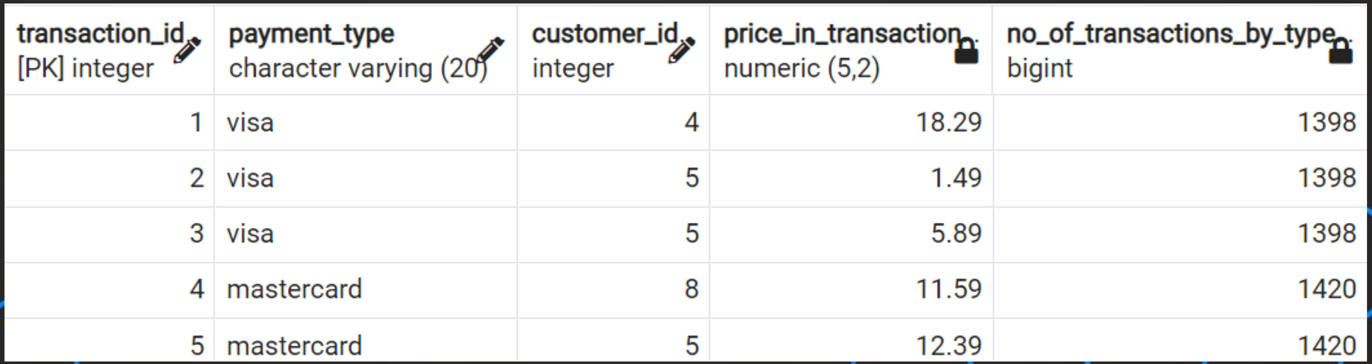
Window functions



```
SELECT
transaction_id,
payment_type,
customer_id,
price_in_transaction,
COUNT(*) OVER(PARTITION BY payment_type)
FROM sales s
```

No. of rows not affected

Aggregated by

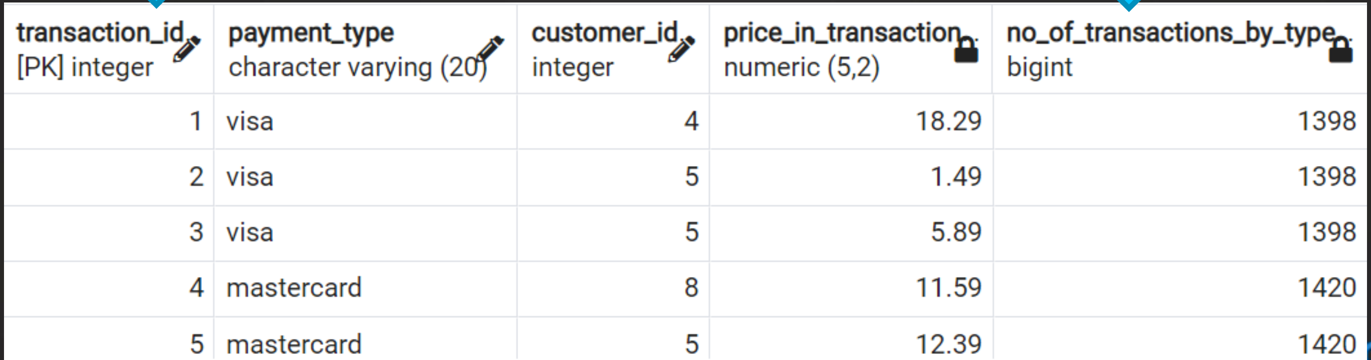


transaction_id [PK] integer	payment_type character varying (20)	customer_id integer	price_in_transaction numeric (5,2)	no_of_transactions_by_type bigint
1	visa	4	18.29	1398
2	visa	5	1.49	1398
3	visa	5	5.89	1398
4	mastercard	8	11.59	1420
5	mastercard	5	12.39	1420

Window functions

No. of rows not affected

Aggregated by

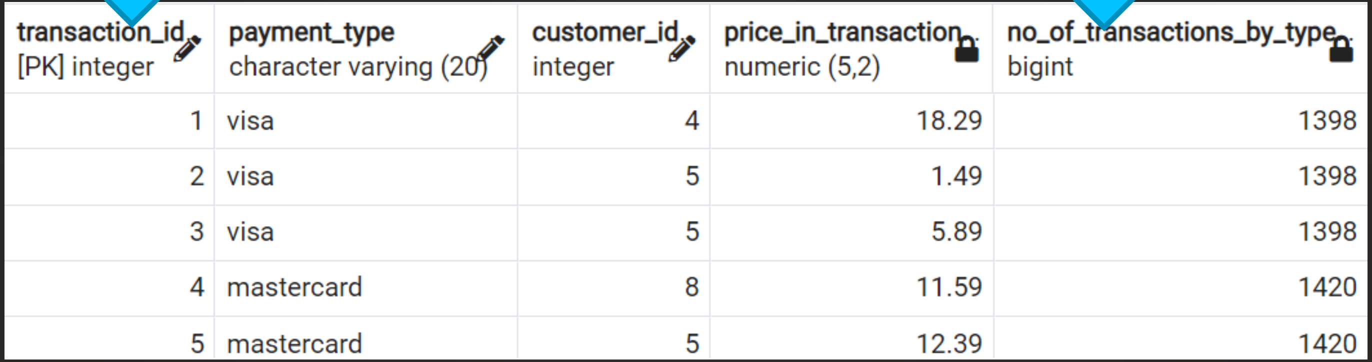


transaction_id [PK] integer	payment_type character varying (20)	customer_id integer	price_in_transaction numeric (5,2)	no_of_transactions_by_type bigint
1	visa	4	18.29	1398
2	visa	5	1.49	1398
3	visa	5	5.89	1398
4	mastercard	8	11.59	1420
5	mastercard	5	12.39	1420

OVER()

No. of rows not affected

Aggregated by



transaction_id [PK] integer	payment_type character varying (20)	customer_id integer	price_in_transaction numeric (5,2)	no_of_transactions_by_type bigint
1	visa	4	18.29	1398
2	visa	5	1.49	1398
3	visa	5	5.89	1398
4	mastercard	8	11.59	1420
5	mastercard	5	12.39	1420

SUM()

COUNT()

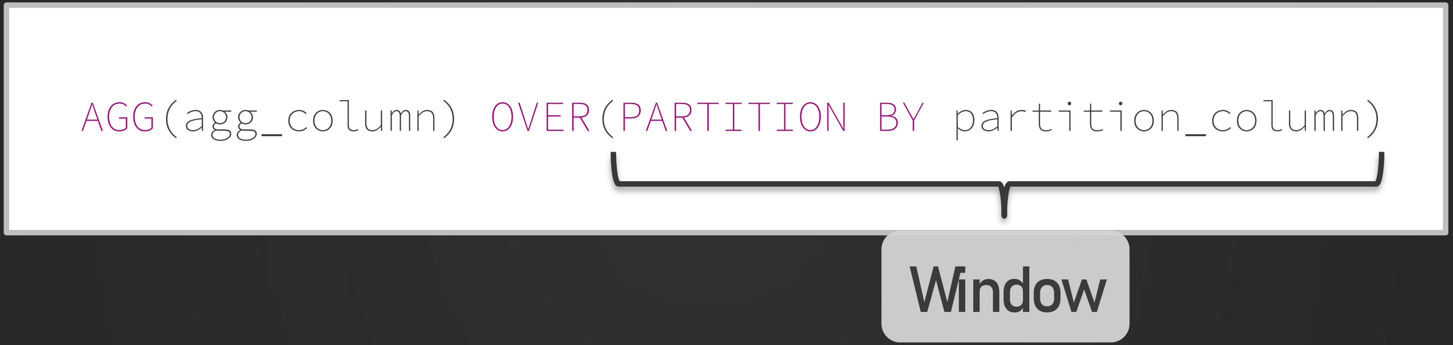
RANK()

FIRST_VALUE()

LEAD()

LAG()

Window functions



```
AGG(agg_column) OVER(PARTITION BY partition_column)
```

Window

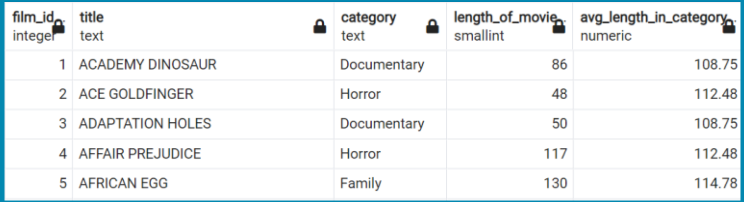
Challenge

Write a query that returns the list of movies including

- film_id,
- title,
- length,
- category,
- average length of movies in that category.

Order the results by film_id.

Result



film_id	title	category	length_of_movie	avg_length_in_category
integer	text	text	smallint	numeric
1	ACADEMY DINOSAUR	Documentary	86	108.75
2	ACE GOLDFINGER	Horror	48	112.48
3	ADAPTATION HOLES	Documentary	50	108.75
4	AFFAIR PREJUDICE	Horror	117	112.48
5	AFRICAN EGG	Family	130	114.78

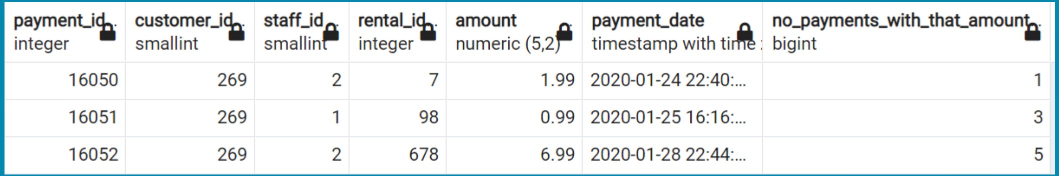
Challenge

Write a query that returns all payment details including

- the number of payments that were made by this customer and that amount

Order the results by payment_id.

Result



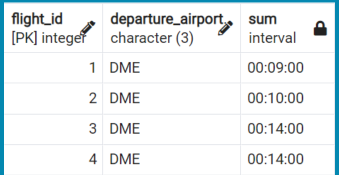
payment_id integer	customer_id smallint	staff_id smallint	rental_id integer	amount numeric (5,2)	payment_date timestamp with time	no_payments_with_that_amount bigint
16050	269	2	7	1.99	2020-01-24 22:40:...	1
16051	269	1	98	0.99	2020-01-25 16:16:...	3
16052	269	2	678	6.99	2020-01-28 22:44:...	5

Challenge

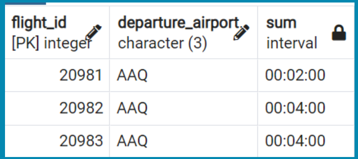
Write a query that returns the running total of how late the flights are (difference between actual_arrival and scheduled arrival) ordered by flight_id including the departure airport.

As a second query, calculate the same running total but partition also by the departure airport.

Result



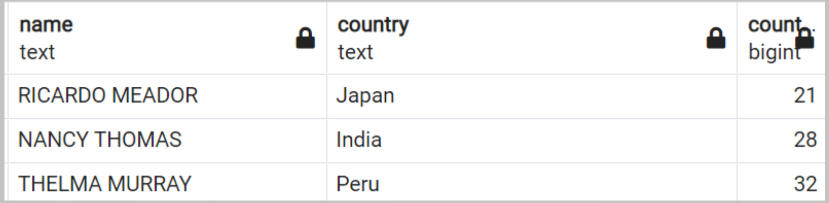
flight_id [PK] integer	departure_airport character (3)	sum interval
1	DME	00:09:00
2	DME	00:10:00
3	DME	00:14:00
4	DME	00:14:00



flight_id [PK] integer	departure_airport character (3)	sum interval
20981	AAQ	00:02:00
20982	AAQ	00:04:00
20983	AAQ	00:04:00

Challenge

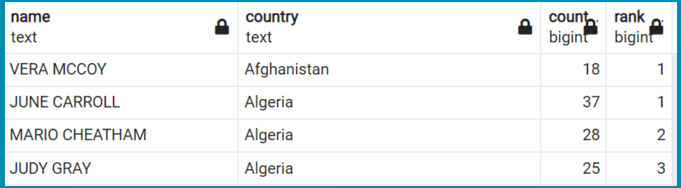
Write a query that returns the customers' name, the country and how many payments they have. For that use the existing view *customer_list.*



name text	country text	count bigint
RICARDO MEADOR	Japan	21
NANCY THOMAS	India	28
THELMA MURRAY	Peru	32

Afterwards create a ranking of the top customers with most sales for each country. Filter the results to only the top 3 customers per country.

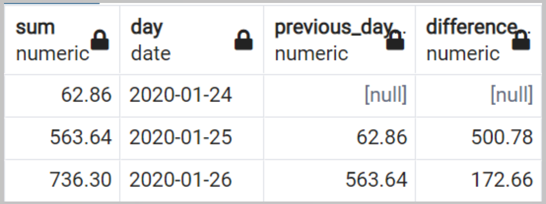
Result



name text	country text	count bigint	rank bigint
VERA MCCOY	Afghanistan	18	1
JUNE CARROLL	Algeria	37	1
MARIO CHEATHAM	Algeria	28	2
JUDY GRAY	Algeria	25	3

Challenge

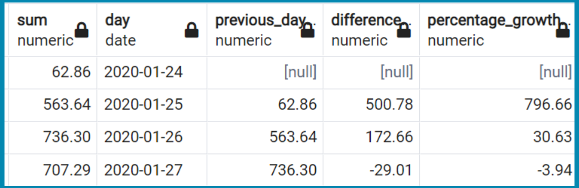
Write a query that returns the revenue of the day and the revenue of the previous day.



sum numeric	day date	previous_day.. numeric	difference.. numeric
62.86	2020-01-24	[null]	[null]
563.64	2020-01-25	62.86	500.78
736.30	2020-01-26	563.64	172.66

Afterwards calculate also the percentage growth compared to the previous day.

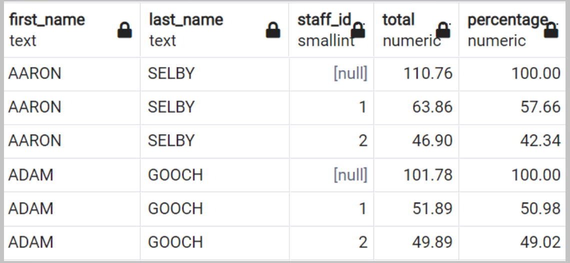
Result



sum numeric	day date	previous_day.. numeric	difference.. numeric	percentage_growth.. numeric
62.86	2020-01-24	[null]	[null]	[null]
563.64	2020-01-25	62.86	500.78	796.66
736.30	2020-01-26	563.64	172.66	30.63
707.29	2020-01-27	736.30	-29.01	-3.94

Challenge

Write a query that calculates now the share of revenue each staff_id makes per customer. The result should look like this:



first_name	last_name	staff_id	total	percentage
text	text	smallint	numeric	numeric
AARON	SELBY	[null]	110.76	100.00
AARON	SELBY	1	63.86	57.66
AARON	SELBY	2	46.90	42.34
ADAM	GOOCH	[null]	101.78	100.00
ADAM	GOOCH	1	51.89	50.98
ADAM	GOOCH	2	49.89	49.02

Hint

You need to use a subquery.