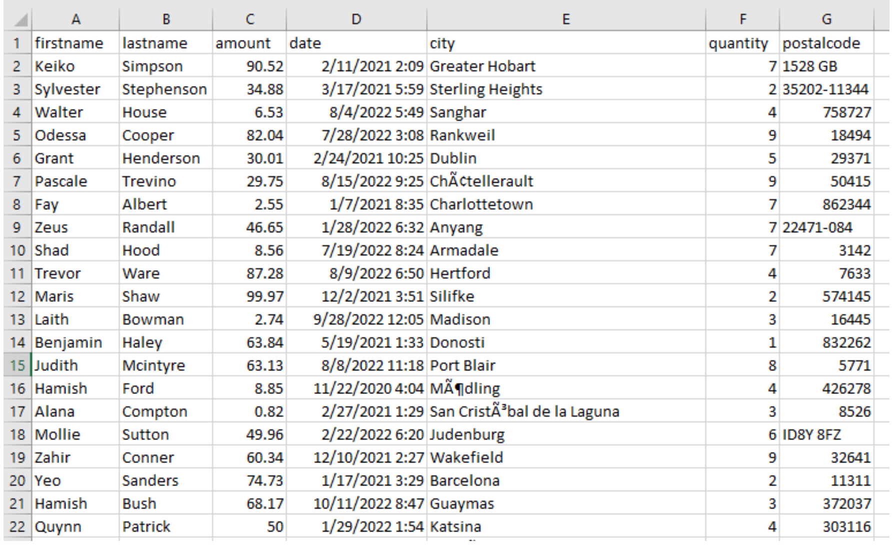
https://www.analyticsvidhya.com/blog/2020/07/sql-functions-for-data-analysis-tasks/



	A	B	C	D	E	F	G
1	firstname	lastname	amount	date	city	quantity	postalcode
2	Keiko	Simpson	90.52	2/11/2021 2:09	Greater Hobart	7	1528 GB
3	Sylvester	Stephenson	34.88	3/17/2021 5:59	Sterling Heights	2	35202-11344
4	Walter	House	6.53	8/4/2022 5:49	Sanghar	4	758727
5	Odessa	Cooper	82.04	7/28/2022 3:08	Rankweil	9	18494
6	Grant	Henderson	30.01	2/24/2021 10:25	Dublin	5	29371
7	Pascale	Trevino	29.75	8/15/2022 9:25	ChÃ¢tellerault	9	50415
8	Fay	Albert	2.55	1/7/2021 8:35	Charlottetown	7	862344
9	Zeus	Randall	46.65	1/28/2022 6:32	Anyang	7	22471-084
10	Shad	Hood	8.56	7/19/2022 8:24	Armadale	7	3142
11	Trevor	Ware	87.28	8/9/2022 6:50	Hertford	4	7633
12	Maris	Shaw	99.97	12/2/2021 3:51	Silifke	2	574145
13	Laith	Bowman	2.74	9/28/2022 12:05	Madison	3	16445
14	Benjamin	Haley	63.84	5/19/2021 1:33	Donosti	1	832262
15	Judith	Mcintyre	63.13	8/8/2022 11:18	Port Blair	8	5771
16	Hamish	Ford	8.85	11/22/2020 4:04	MÃ¶dling	4	426278
17	Alana	Compton	0.82	2/27/2021 1:29	San CristÃ³bal de la Laguna	3	8526
18	Mollie	Sutton	49.96	2/22/2022 6:20	Judenburg	6	ID8Y 8FZ
19	Zahir	Conner	60.34	12/10/2021 2:27	Wakefield	9	32641
20	Yeo	Sanders	74.73	1/17/2021 3:29	Barcelona	2	11311
21	Hamish	Bush	68.17	10/11/2022 8:47	Guaymas	3	372037
22	Quynn	Patrick	50	1/29/2022 1:54	Katsina	4	303116

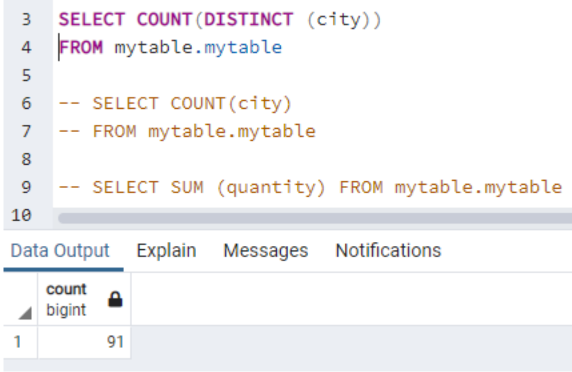
These are 100 records

Aggregate functions

- **Count**

get the number of cities where the order came from

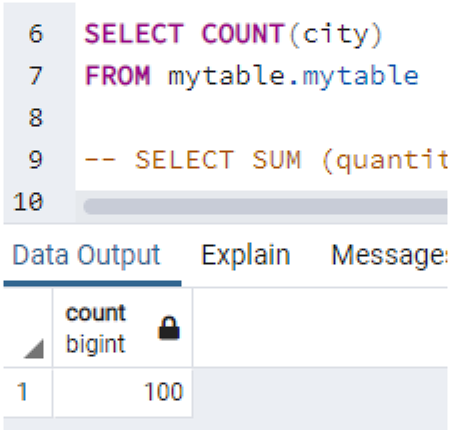
SELECT COUNT(DISTINCT city)
FROM mytable.mytable



```
SELECT COUNT(DISTINCT (city))
FROM mytable.mytable

-- SELECT COUNT(city)
-- FROM mytable.mytable

-- SELECT SUM (quantity) FROM mytable.mytable

```

```
   count
   bigint
1      91
```

SELECT COUNT(city)
FROM mytable.mytable



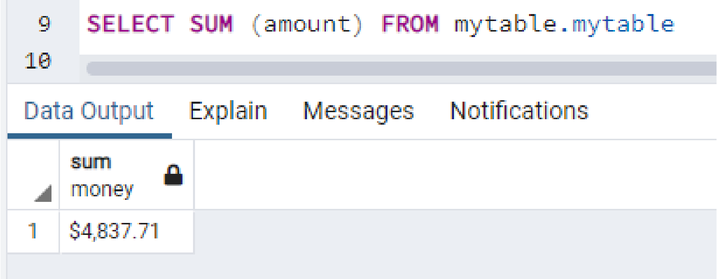
```
SELECT COUNT(city)
FROM mytable.mytable

-- SELECT SUM (quantit
```

```
   count
   bigint
1      100
```

- **Sum**

<u>the total sum of Amount</u>
ALTER TABLE mytable.mytable ALTER COLUMN amount TYPE money using amount::text::money *--to change amount column to currency*

SELECT SUM (amount) FROM mytable.mytable

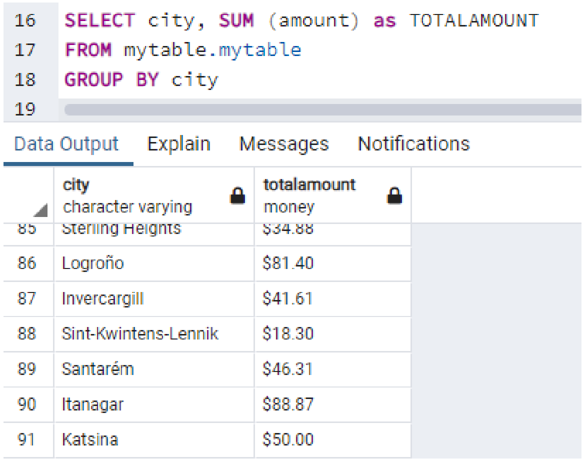


```
SELECT SUM (amount) FROM mytable.mytable
```

```
   sum
   money
1  $4,837.71
```

<u>total *amount* for every city</u>

SELECT city, SUM (amount) as TOTALAMOUNT
FROM mytable.mytable
GROUP BY city

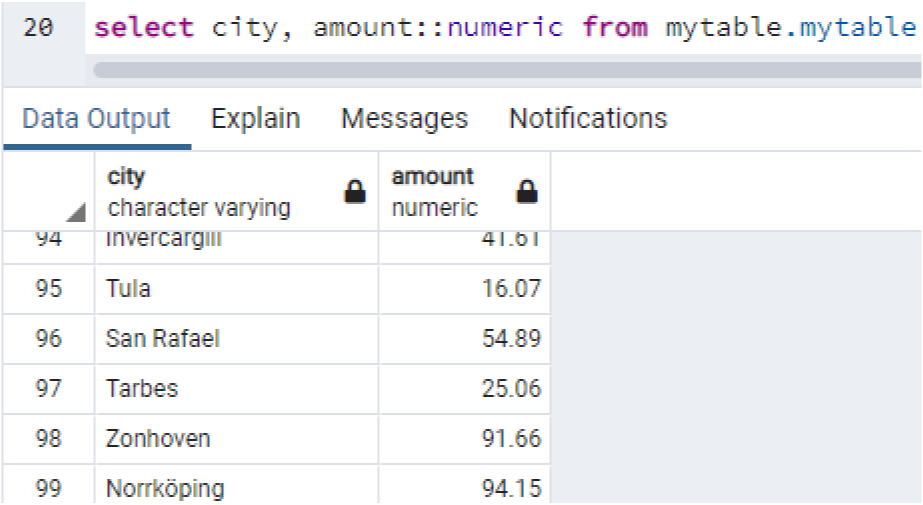


```
SELECT city, SUM (amount) as TOTALAMOUNT
FROM mytable.mytable
GROUP BY city
```

	city character varying	totalamount money
85	Sterling Heights	$34.88
86	Logroño	$81.40
87	Invercargill	$41.61
88	Sint-Kwintens-Lennik	$18.30
89	Santarém	$46.31
90	Itanagar	$88.87
91	Katsina	$50.00

- ## Average

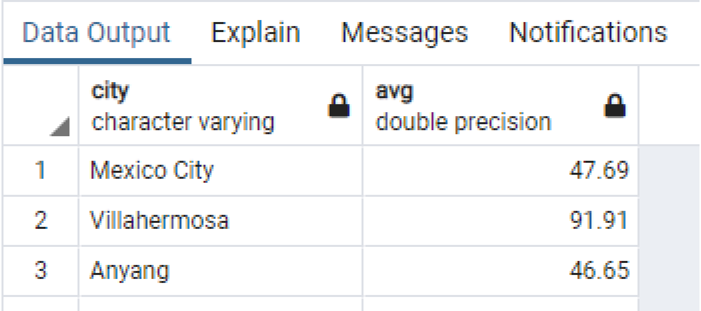
  <u>multiple orders from the same city - calculate the average amount rather than the total sum.</u>



```
select city, amount::numeric from mytable.mytable
```

	city character varying	amount numeric
94	Invercargill	41.61
95	Tula	16.07
96	San Rafael	54.89
97	Tarbes	25.06
98	Zonhoven	91.66
99	Norrköping	94.15

SELECT city, AVG(amount::money::numeric::float8) FROM mytable.mytable GROUP BY city

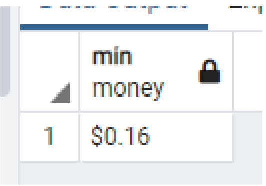


	city character varying	avg double precision
1	Mexico City	47.69
2	Villahermosa	91.91
3	Anyang	46.65

- ## Min and Max

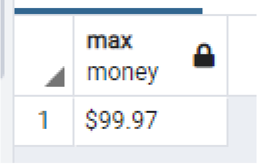
computing the min and max values.

```
SELECT min(amount) FROM mytable.mytable
```



	min money
1	$0.16

```
SELECT max(amount) FROM mytable.mytable
```



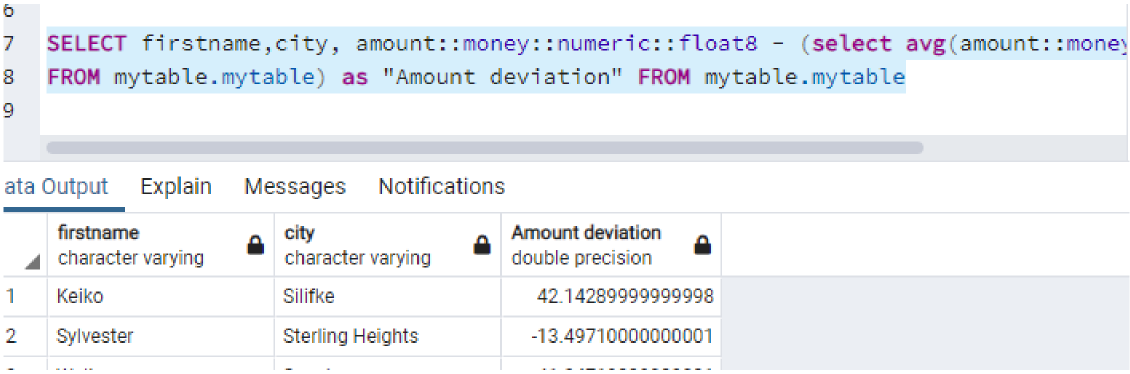
	max money
1	$99.97

Mathematical functions

- ## Absolute

the deviation of the amount for every record from the average

```
SELECT firstname,city, amount::money::numeric::float8 - (select

avg(amount::money::numeric::float8)

FROM mytable.mytable) as "Amount deviation" FROM mytable.mytable
```



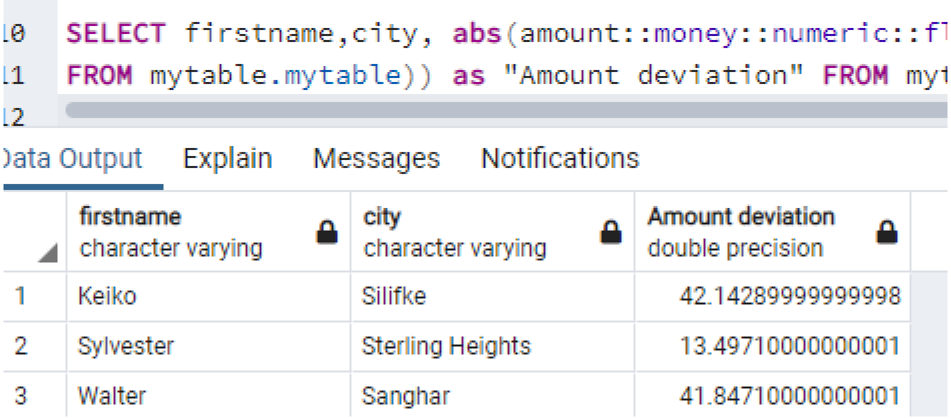
```
SELECT firstname,city, amount::money::numeric::float8 - (select avg(amount::mone
FROM mytable.mytable) as "Amount deviation" FROM mytable.mytable
```

	firstname character varying	city character varying	Amount deviation double precision
1	Keiko	Silifke	42.1428999999998
2	Sylvester	Sterling Heights	-13.4971000000001

negative values converted to positives

SELECT firstname,city, abs(amount::money::numeric::float8 - (select avg(amount::money::numeric::float8)

FROM mytable.mytable)) as "Amount deviation" FROM mytable.mytable



```
SELECT firstname,city, abs(amount::money::numeric::fl
FROM mytable.mytable)) as "Amount deviation" FROM myt
```

	firstname character varying	city character varying	Amount deviation double precision
1	Keiko	Silifke	42.1428999999998
2	Sylvester	Sterling Heights	13.4971000000001
3	Walter	Sanghar	41.8471000000001

- ## Ceil and Floor

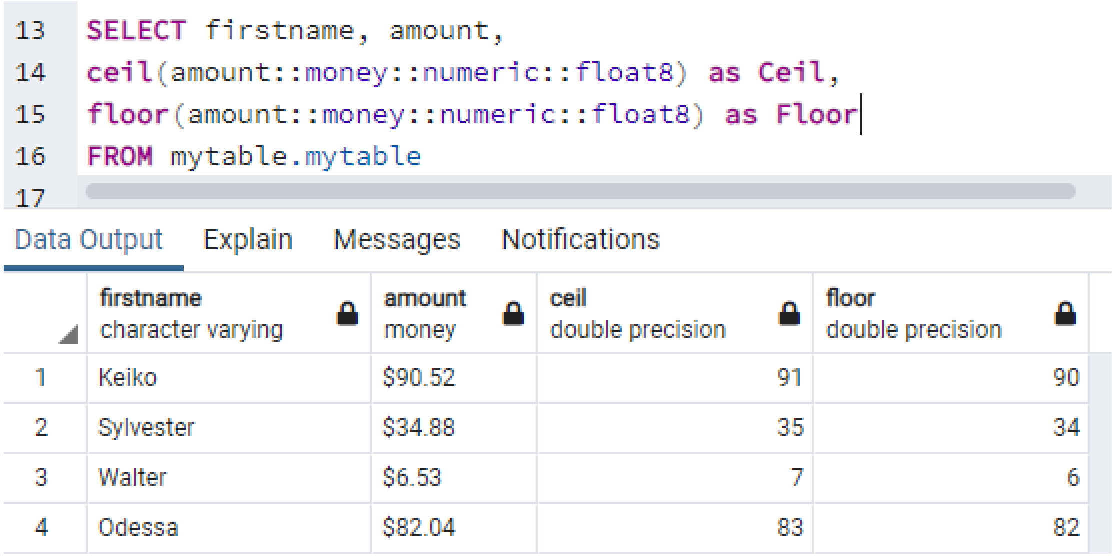
  the *Amount* column has lots of decimal values - convert them to integers using ceil() or the floor() function

SELECT firstname, amount,

ceil(amount::money::numeric::float8) as Ceil,

floor(amount::money::numeric::float8) as Floor

FROM mytable.mytable



```
SELECT firstname, amount,
ceil(amount::money::numeric::float8) as Ceil,
floor(amount::money::numeric::float8) as Floor
FROM mytable.mytable

```

	firstname character varying	amount money	ceil double precision	floor double precision
1	Keiko	$90.52	91	90
2	Sylvester	$34.88	35	34
3	Walter	$6.53	7	6
4	Odessa	$82.04	83	82

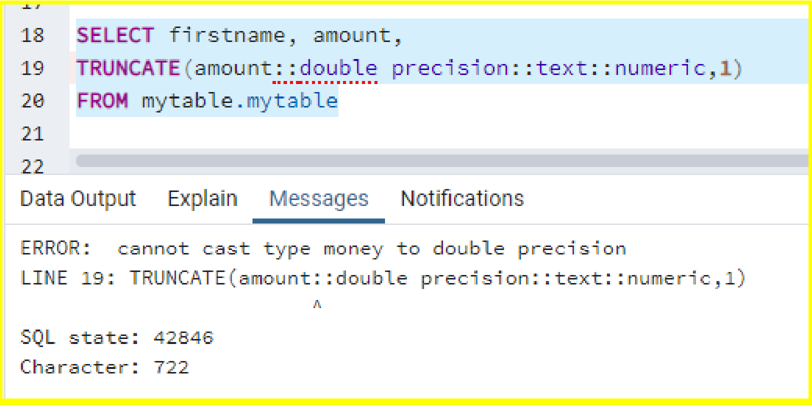
- **Truncate**

truncate the number of decimal places in the number.

SELECT firstname, amount,

TRUNCATE(amount::double precision::text::numeric,1)

FROM mytable.mytable



```sql
SELECT firstname, amount,
TRUNCATE(amount::double precision::text::numeric,1)
FROM mytable.mytable
```

```
ERROR:  cannot cast type money to double precision
LINE 19: TRUNCATE(amount::double precision::text::numeric,1)
                         ^
SQL state: 42846
Character: 722
```

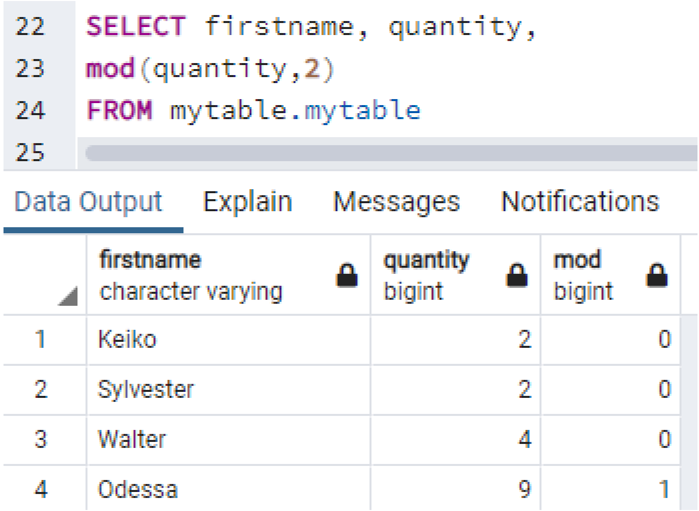
- ## Modulo

  <u>find those records which had an odd number of quantities.</u>

  SELECT firstname, quantity,

  mod(quantity,2)

  FROM mytable.mytable



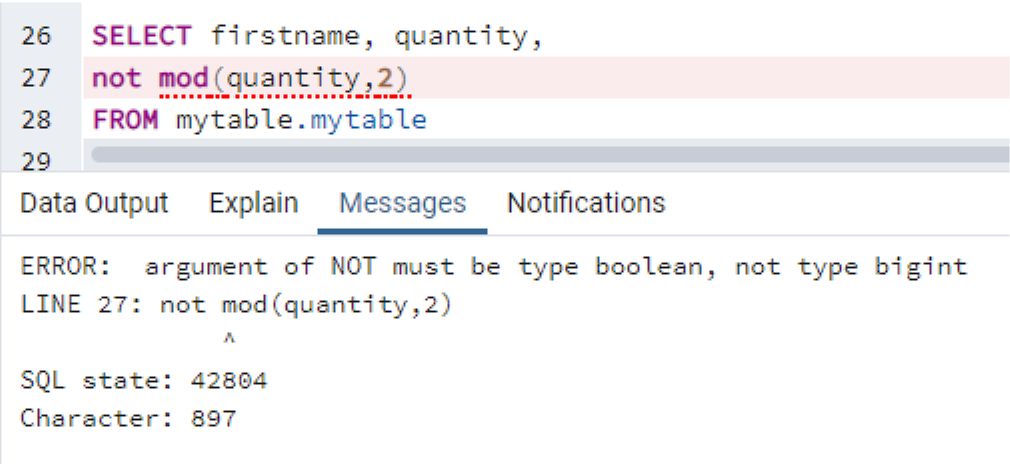
```sql
SELECT firstname, quantity,
mod(quantity,2)
FROM mytable.mytable
```

	firstname character varying	quantity bigint	mod bigint
1	Keiko	2	0
2	Sylvester	2	0
3	Walter	4	0
4	Odessa	9	1

SELECT firstname, quantity,

not mod(quantity,2)

FROM mytable.mytable



```sql
SELECT firstname, quantity,
not mod(quantity,2)
FROM mytable.mytable
```

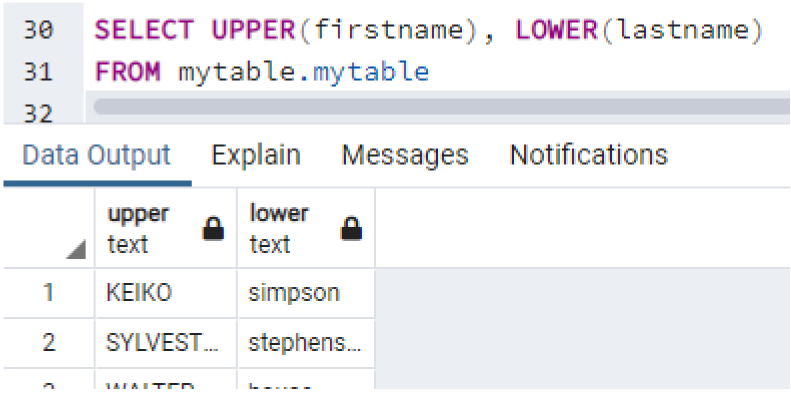
```
ERROR:  argument of NOT must be type boolean, not type bigint
LINE 27: not mod(quantity,2)
             ^
SQL state: 42804
Character: 897
```

String functions

- ### Lower and Upper

SELECT UPPER(firstname), LOWER(lastname)

FROM mytable.mytable

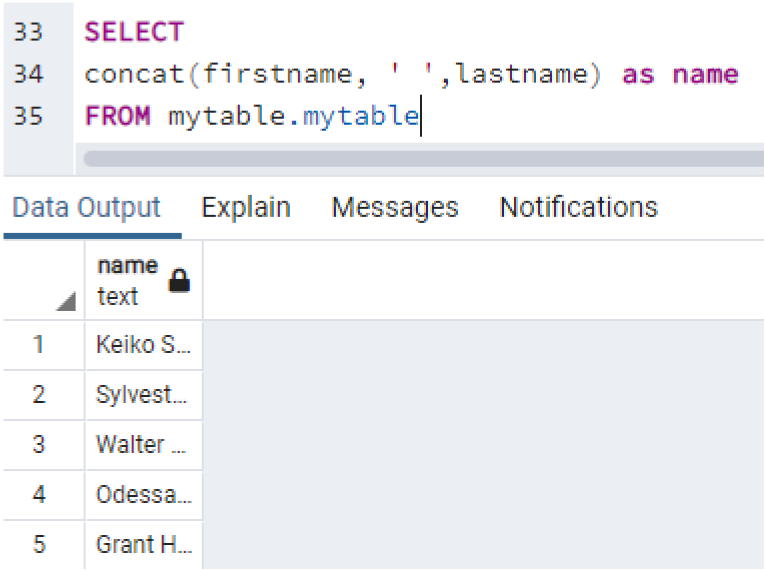


```sql
SELECT UPPER(firstname), LOWER(lastname)
FROM mytable.mytable
```

	upper text	lower text
1	KEIKO	simpson
2	SYLVEST...	stephens...
3	WALTER	hayes

- ### Concat

SELECT
concat(firstname, ' ',lastname) as name
FROM mytable.mytable



```
SELECT
concat(firstname, ' ',lastname) as name
FROM mytable.mytable
```

	name text
1	Keiko S...
2	Sylvest...
3	Walter ...
4	Odessa...
5	Grant H...

- **Trim**

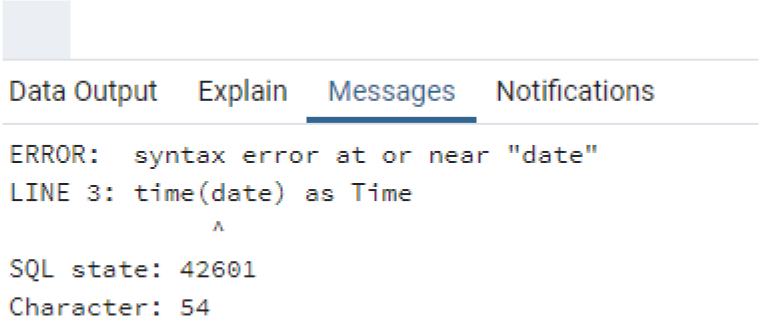
SELECT
concat(firstname, ' ',trim(lastname)) as name
FROM mytable.mytable

Date and time functions

- **Date and Time**

  common column for date and time - extract the respective values.

SELECT firstname, lastname,
date(date) as Date,
time(date) as Time
FROM mytable.mytable



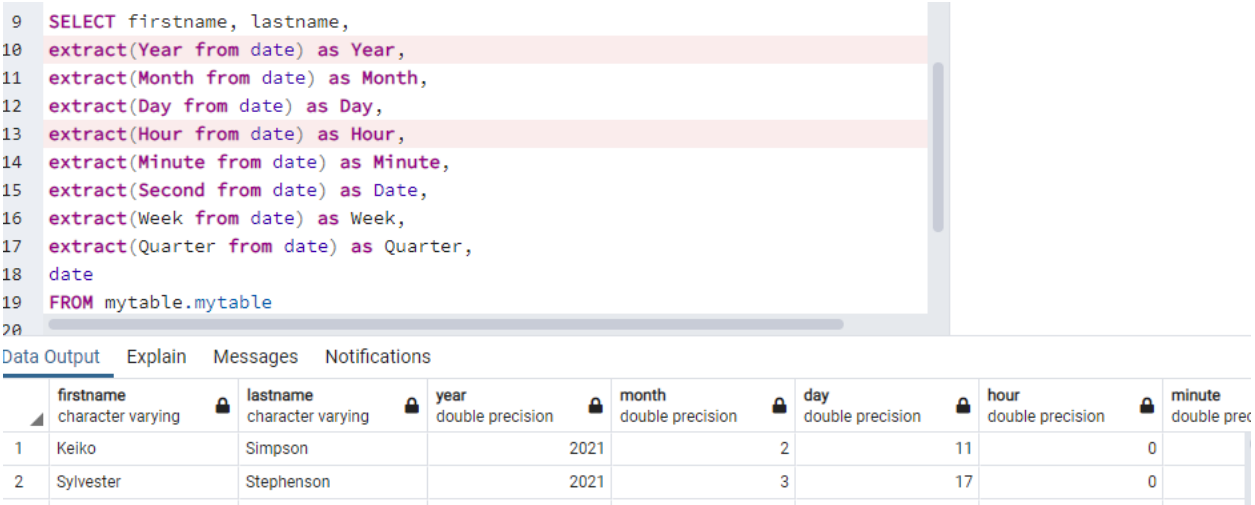
```
ERROR:  syntax error at or near "date"
LINE 3: time(date) as Time
             ^
SQL state: 42601
Character: 54
```

- **Extract**

- □ analyze how many of the orders were placed on a particular day of the week or month, or maybe the time of the day.
- □ extract the week of the year or the quarter of the year.

ALTER TABLE mytable.mytable ALTER COLUMN "date" type date using ("date"::text::date);

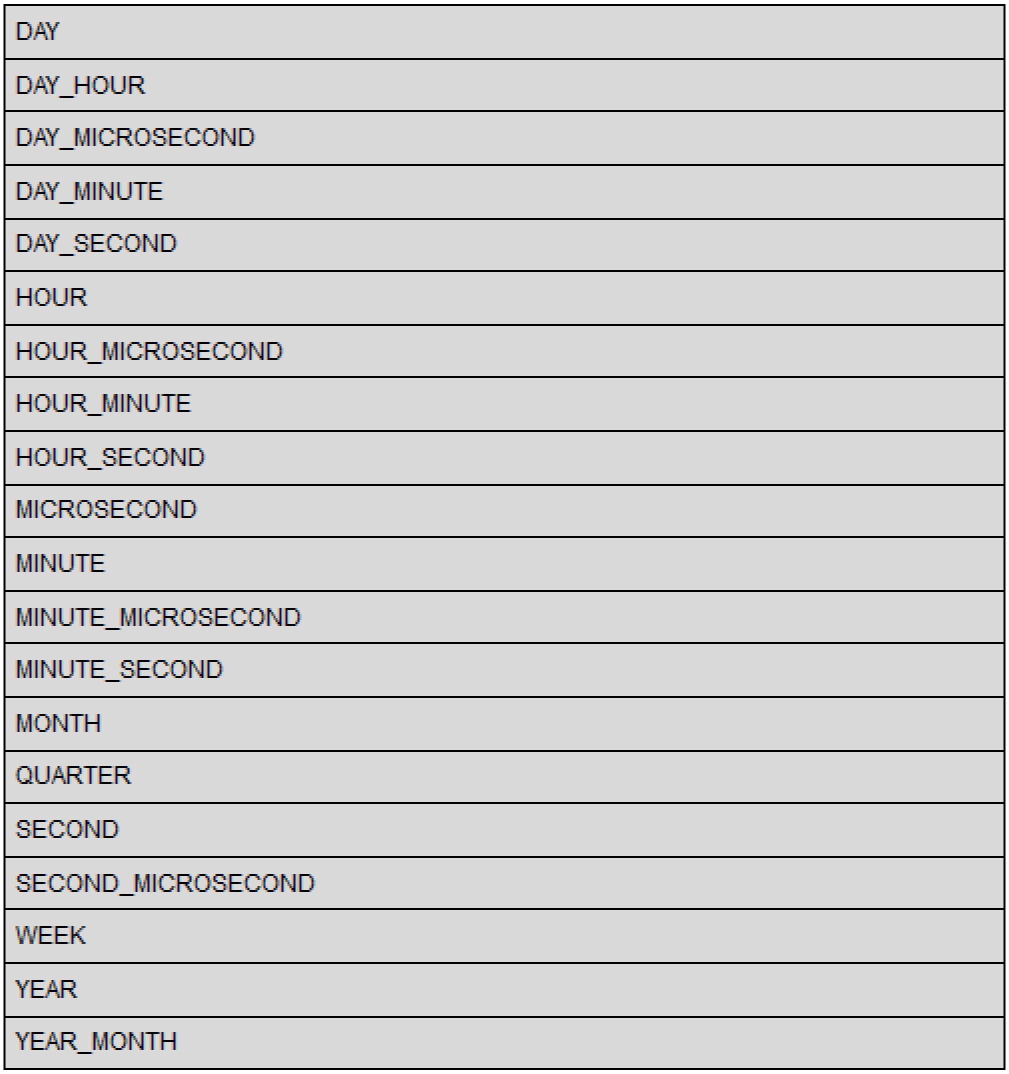
SELECT firstname, lastname,
extract(Year from date) as Year,
extract(Month from date) as Month,
extract(Day from date) as Day,
extract(Hour from date) as Hour,
extract(Minute from date) as Minute,
extract(Second from date) as Date,
extract(Week from date) as Week,
extract(Quarter from date) as Quarter,
date
FROM mytable.mytable



```
SELECT firstname, lastname,
extract(Year from date) as Year,
extract(Month from date) as Month,
extract(Day from date) as Day,
extract(Hour from date) as Hour,
extract(Minute from date) as Minute,
extract(Second from date) as Date,
extract(Week from date) as Week,
extract(Quarter from date) as Quarter,
date
FROM mytable.mytable
```

	firstname character varying	lastname character varying	year double precision	month double precision	day double precision	hour double precision	minute double prec
1	Keiko	Simpson	2021	2	11	0	
2	Sylvester	Stephenson	2021	3	17	0	

A complete list of all the units that you can extract from the date is as follows:

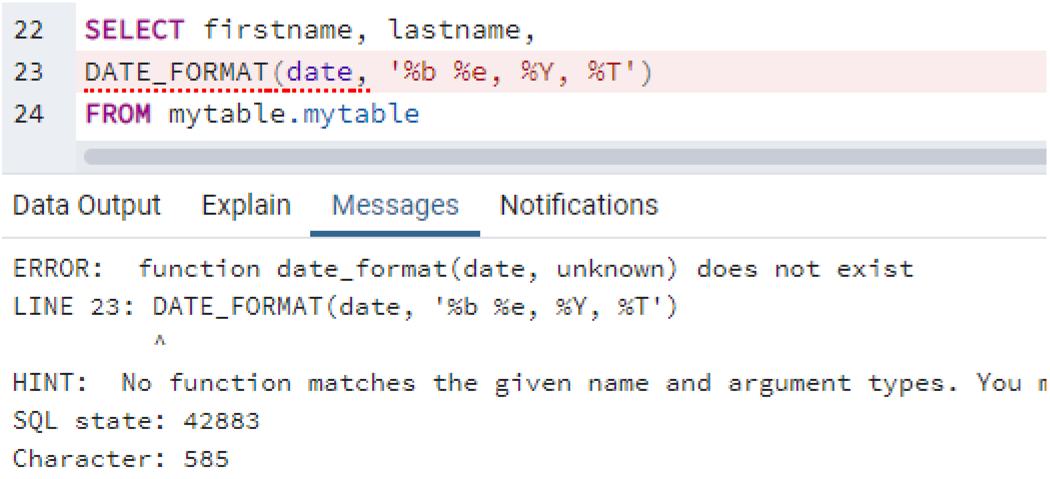


DAY
DAY_HOUR
DAY_MICROSECOND
DAY_MINUTE
DAY_SECOND
HOUR
HOUR_MICROSECOND
HOUR_MINUTE
HOUR_SECOND
MICROSECOND
MINUTE
MINUTE_MICROSECOND
MINUTE_SECOND
MONTH
QUARTER
SECOND
SECOND_MICROSECOND
WEEK
YEAR
YEAR_MONTH

- ## Date format

  output the dates in the day-month name-year format

  ```
  SELECT firstname, lastname,
  DATE_FORMAT(date, '%b %e, %Y, %T')
  FROM mytable.mytable
  ```



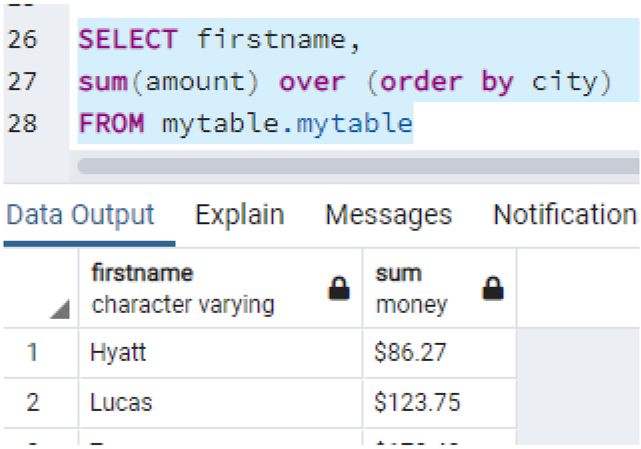
```
SELECT firstname, lastname,
DATE_FORMAT(date, '%b %e, %Y, %T')
FROM mytable.mytable
```

```
ERROR:  function date_format(date, unknown) does not exist
LINE 23: DATE_FORMAT(date, '%b %e, %Y, %T')
         ^
HINT:  No function matches the given name and argument types. You
SQL state: 42883
Character: 585
```

Other date formats https://dev.mysql.com/doc/refman/8.0/en/date-and-time-functions.html#function_date-format

- ## Window function

SELECT firstname,
sum(amount) over (order by city)
FROM mytable.mytable



```
SELECT firstname,
sum(amount) over (order by city)
FROM mytable.mytable
```

	firstname character varying	sum money
1	Hyatt	$86.27
2	Lucas	$123.75

- ## Rank

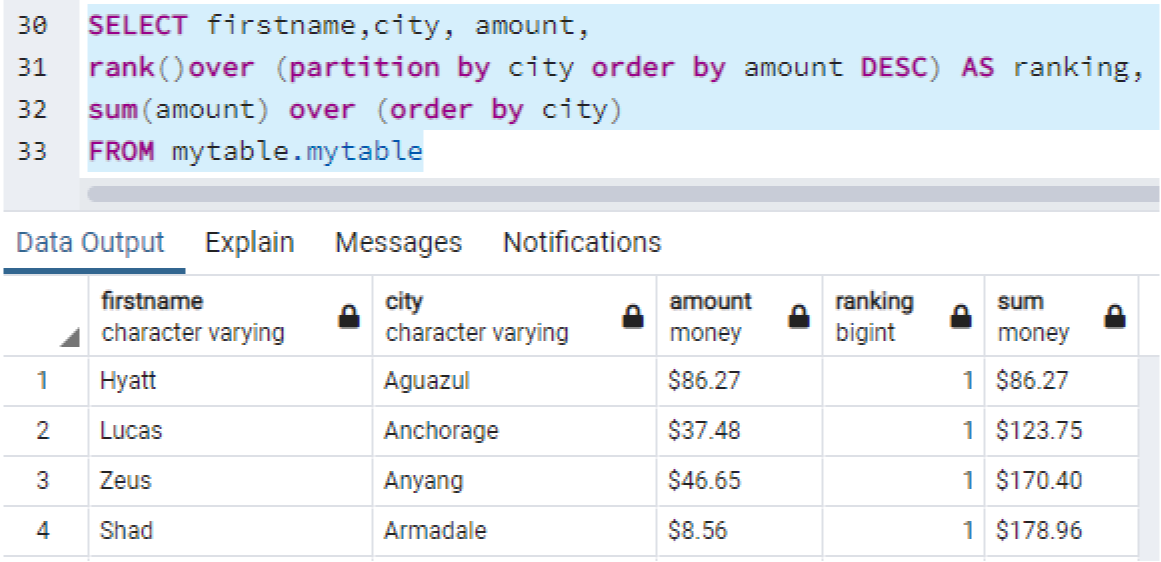
rank the rows in our table based on the amount of order within each city.

SELECT firstname,city, amount,

rank()over (partition by city order by amount DESC) AS ranking,

sum(amount) over (order by city)

FROM mytable.mytable



```sql
SELECT firstname,city, amount,
rank()over (partition by city order by amount DESC) AS ranking,
sum(amount) over (order by city)
FROM mytable.mytable
```

	firstname character varying	city character varying	amount money	ranking bigint	sum money
1	Hyatt	Aguazul	$86.27	1	$86.27
2	Lucas	Anchorage	$37.48	1	$123.75
3	Zeus	Anyang	$46.65	1	$170.40
4	Shad	Armadale	$8.56	1	$178.96

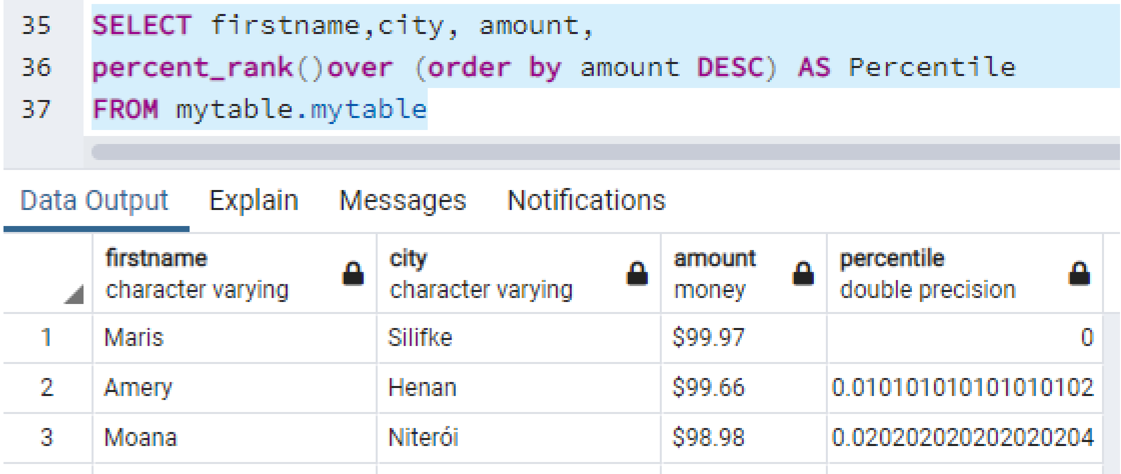
- ## Percent_value

the amount percentile of each customer in the table.

SELECT firstname,city, amount,

percent_rank()over (order by amount DESC) AS Percentile

FROM mytable.mytable



```sql
SELECT firstname,city, amount,
percent_rank()over (order by amount DESC) AS Percentile
FROM mytable.mytable
```

	firstname character varying	city character varying	amount money	percentile double precision
1	Maris	Silifke	$99.97	0
2	Amery	Henan	$99.66	0.010101010101010102
3	Moana	Niterói	$98.98	0.020202020202020204

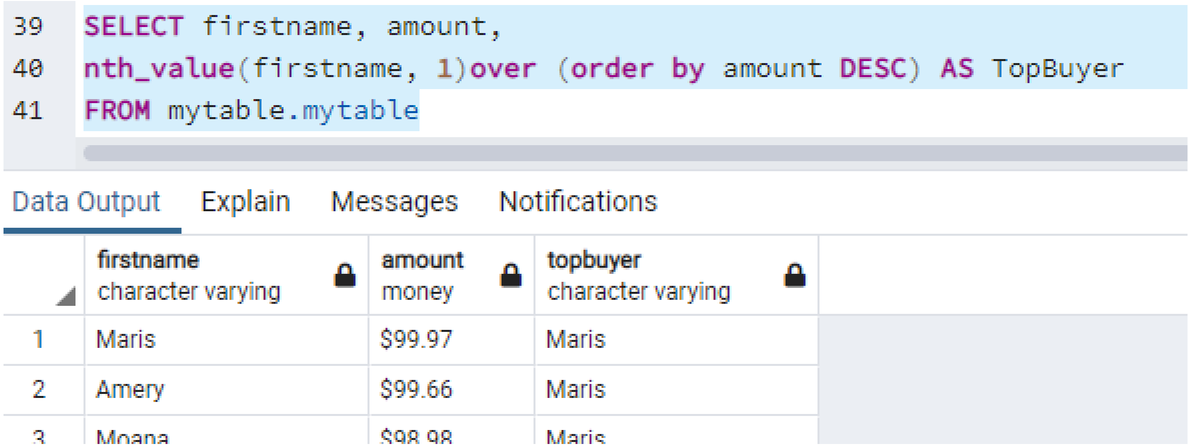
- Nth_value

find out who was the top buyer in the table.

SELECT firstname, amount,

nth_value(firstname, 1)over (order by amount DESC) AS TopBuyer

FROM mytable.mytable



```sql
SELECT firstname, amount,
nth_value(firstname, 1)over (order by amount DESC) AS TopBuyer
FROM mytable.mytable
```

	firstname character varying	amount money	topbuyer character varying
1	Maris	$99.97	Maris
2	Amery	$99.66	Maris
3	Moana	$98.98	Maris

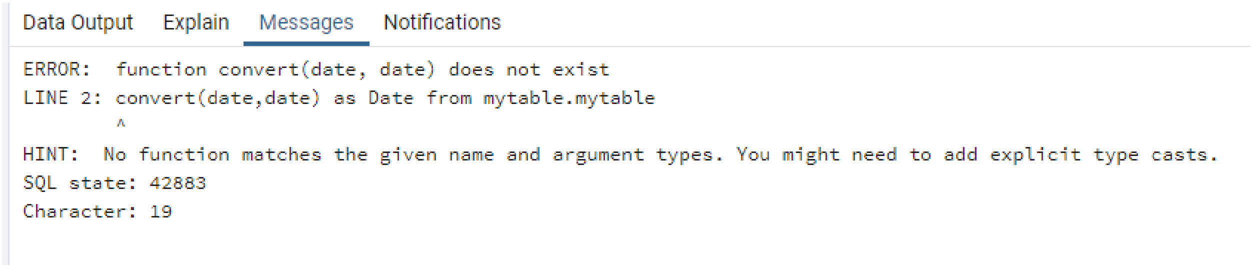
Miscellaneous functions

Convert

convert the data type of the date column before we print the value.

SELECT firstname,

convert(date,date) as Date from mytable.mytable

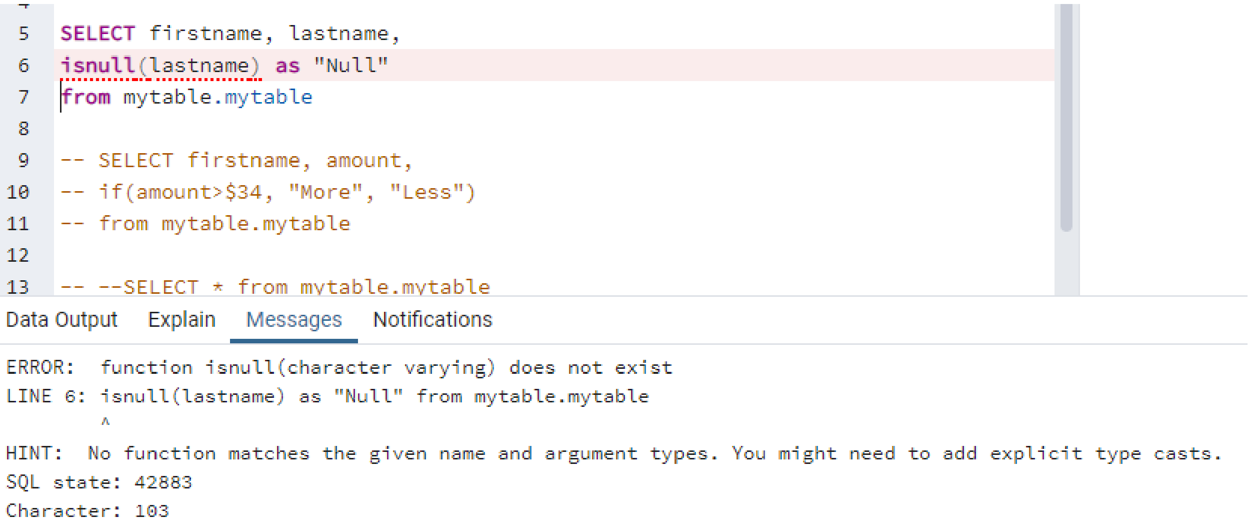


```
ERROR:  function convert(date, date) does not exist
LINE 2: convert(date,date) as Date from mytable.mytable
        ^
HINT:  No function matches the given name and argument types. You might need to add explicit type casts.
SQL state: 42883
Character: 19
```

Isnull

Generally, if you don’t specify the non-value for your attribute, chances are you will end up with some null values in the column. But you can easily deal with them using the isnull() function.

You just have to write the expression within the function. It will return 1 for a null and 0 otherwise.

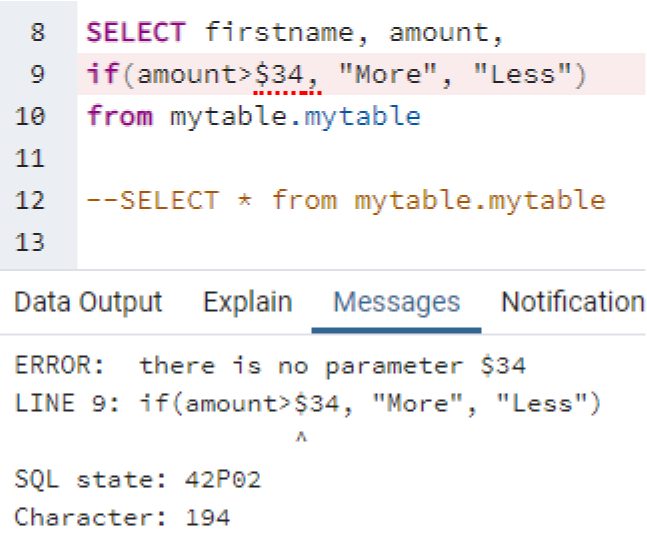


```
SELECT firstname, lastname,
isnull(lastname) as "Null"
from mytable.mytable

-- SELECT firstname, amount,
-- if(amount>$34, "More", "Less")
-- from mytable.mytable

-- --SELECT * from mytable.mytable
```

```
ERROR:  function isnull(character varying) does not exist
LINE 6: isnull(lastname) as "Null" from mytable.mytable
        ^
HINT:  No function matches the given name and argument types. You might need to add explicit type casts.
SQL state: 42883
Character: 103
```

If

find out which customer paid more than 34 amount for their order.



```
SELECT firstname, amount,
if(amount>$34, "More", "Less")
from mytable.mytable

--SELECT * from mytable.mytable

```

```
ERROR:  there is no parameter $34
LINE 9: if(amount>$34, "More", "Less")
                  ^
SQL state: 42P02
Character: 194
```