

# Basic Queries

<https://github.com/leocareer>

Limit to 5000 rows

```
4 -- List of countries that are shopping
5 • SELECT DISTINCT country 'Countries that are shopping'
6 FROM company AS t1
7 JOIN transaction AS t2
8 ON t1.id = t2.company_id
9 ORDER BY country;
```

100% 29:11

**Result Grid** Filter Rows: Search Export:

Countries that are shopping
Australia
Belgium
Canada
China
France
Germany
Ireland
Italy
Netherlands
New Zealand
Norway
Spain
Sweden
United Kingdom
United States

Result 3 Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
27...	11:39:42	SELECT DISTINCT country '...	15 row(s) returned	0.0015 sec / 0.00001...

Limit to 5000 rows

```
12 • SELECT count(DISTINCT country) 'How many countries are shopping'
13 FROM company AS t1
14 JOIN transaction AS t2
15 ON t1.id = t2.company_id;
```

100% 21:18

**Result Grid** Filter Rows: Search Export:

How many countries are shopping
15

Result 5 Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
27...	11:43:44	SELECT count(DISTINCT co...	1 row(s) returned	0.0025 sec / 0.00000...

```
17 -- Identify the company with the highest average sales (solution with 'limit')
18 • SELECT company_name 'Company', ROUND(AVG(amount), 2) AS Average_sales
19 FROM transaction AS t1
20 JOIN company AS t2
21 ON t2.id = t1.company_id
22 GROUP BY t1.company_id
23 ORDER BY Average_sales DESC
24 LIMIT 1;
25
```

Company	Average_sales
Eget Ipsum Ltd	473.08

Time	Action	Response	Duration / Fetch Time
27...	12:05:49	SELECT company_name 'Co... 1 row(s) returned	0.0060 sec / 0.0000...

Using LIMIT in SQL isn't always a good idea because:

1. Performance issues: With large datasets, the query often selects all rows, sorts them, and then trims the result to the specified limit. This can slow down execution for big tables.
2. Inaccurate results: LIMIT can hide important rows, especially without proper sorting, leading to random or non-representative samples.
3. Unpredictable behavior: Without an explicit ORDER BY clause, rows may be returned in a random order, producing unpredictable results.

While LIMIT is useful, it should be used carefully, considering these limitations.

# Basic Queries

<https://github.com/leocareer>

26 -- Identify the company with the highest average sales (solution without 'limit')

27 • **SELECT** company\_name, Average\_sales

28 **FROM** (

29     **SELECT** company\_id, ROUND(AVG(amount), 2) **AS** Average\_sales

30     **FROM** transaction

31     **GROUP BY** company\_id

32 ) **AS** t2

33 **JOIN** company **AS** t1 **ON** t1.id = t2.company\_id

34 **WHERE** Average\_sales = (

35     **SELECT** max(avg\_amount\_t3)

36     **FROM** (

37         **SELECT** ROUND(AVG(amount), 2) **AS** avg\_amount\_t3 **FROM** transaction

38         **GROUP BY** company\_id

39     ) **AS** t3

40 );

41

100% 28:23

**Result Grid** Filter Rows: Search Export:

company_name	Average_sales
Eget Ipsum Ltd	473.08

Result 35 Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
✓	27...	12:24:58	SELECT company_name, Av...	1 row(s) returned 0.0035 sec / 0.00000...

45 -- Show all transactions made by companies in Germany

46 • **SELECT** \*

47 **FROM** transaction

48 **WHERE** company\_id **IN** (

49     **SELECT** id **FROM** company

50     **WHERE** country = 'Germany'

51 );

100% 26:55

**Result Grid** Filter Rows: Search Edit: Export/Import:

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
108B1D1D-5B23-A76C-55...	CcU-2938	b-2222	275	83.7839	-178.86	2021-07-07 17:43:16	293.57	0
EA2C3281-C9C1-A387-44...	CcU-2938	b-2222	275	20.2004	-116.84	2021-05-09 10:25:08	119.36	1
0DD2E608-5C9E-D1B3-4...	CcU-2959	b-2234	275	9.68811	130.282	2021-04-17 05:30:17	252.47	1
AB069F53-965E-A2A8-CE...	CcU-2959	b-2234	275	1.64819	-158.007	2021-04-15 13:37:18	60.99	0
0466A42E-47CF-8D24-FD...	CcU-4219	b-2302	170	-43.9695	-117.525	2021-07-26 07:29:18	49.53	0

transaction 37 Apply

Action Output

	Time	Action	Response	Duration / Fetch Time
✓	27...	12:27:51	SELECT * FROM transactio...	118 row(s) returned 0.0012 sec / 0.00005...

# Basic Queries

<https://github.com/leocareer>

53 -- List the companies that have made transactions for an amount higher than the average of all transactions

54 • **SELECT** company\_name 'Companies'

55 **FROM** company

56 **WHERE** id **IN** (  
57     **SELECT** company\_id  
58     **FROM** transaction  
59     **WHERE** amount > (  
60         **SELECT** AVG(amount) **FROM** transaction  
61     )  
62 )

63 **ORDER BY** company\_name;

100% 53:60

**Result Grid** Filter Rows: Search Export:

Companies
A Institute
Ac Fermentum Incorporated
Ac Industries
Aliquam PC
Aliquam PC

company 53 Read Only

**Action Output**

	Time	Action	Response	Duration / Fetch Time
✓	27...	12:58:01	SELECT company_name 'Co... 70 row(s) returned	0.0027 sec / 0.00002...

66 -- Companies that do not have registered transactions will be removed from the system, provide the list of these companies

67 • **SELECT** company\_name 'Companies'

68 **FROM** company

69 **WHERE** NOT EXISTS (  
70     **SELECT** company\_id  
71     **FROM** transaction  
72     **WHERE** transaction.company\_id = company\_id  
73 );

100% 30:79

**Result Grid** Filter Rows: Search Export:

Companies
-----------

company 62 Read Only

**Action Output**

	Time	Action	Response	Duration / Fetch Time
✓	27...	13:07:13	SELECT company_name 'Co... 0 row(s) returned	0.00063 sec / 0.000...

The five days that generated the largest amount of revenue for the company from sales. It shows the date of each transaction along with the sales total.

The screenshot shows a SQL IDE interface. The query editor contains the following SQL code:

```
-- with 'limit'
SELECT sum(amount) 'Sales total', DATE(timestamp) 'Date'
FROM transaction
WHERE declined = 0
GROUP BY DATE(timestamp)
ORDER BY sum(amount) DESC
LIMIT 5;
```

The interface includes a toolbar with icons for file operations, a 'Limit to 5000 rows' dropdown, and a 'Filter Rows' search bar. Below the query editor is the 'Result Grid' showing the following data:

Sales total	Date
1532.36	2021-12-20
1397.96	2021-04-22
1344.37	2021-05-09
1337.62	2022-02-26
1325.12	2021-03-29

Below the result grid is the 'Action Output' table:

	Time	Action	Response	Duration / Fetch Time
27...	13:10:02	SELECT sum(amount) 'Sales...	5 row(s) returned	0.0026 sec / 0.00000...

The screenshot shows a SQL IDE interface. The query editor contains the following SQL code:

```
-- the second implementation option with window function without 'limit'
SELECT Sales_total, Sale_date
FROM (
    SELECT sum(amount) AS Sales_total, DATE(timestamp) AS Sale_date,
    ROW_NUMBER() OVER(ORDER BY sum(amount) DESC) AS ind_amount
    FROM transaction
    WHERE declined = 0
    GROUP BY Sale_date
) AS t
WHERE ind_amount <= 5
ORDER BY Sales_total;
```

The interface includes a toolbar with icons for file operations, a 'Limit to 5000 rows' dropdown, and a 'Filter Rows' search bar. Below the query editor is the 'Result Grid' showing the following data:

Sales_total	Sale_date
1325.12	2021-03-29
1337.62	2022-02-26
1344.37	2021-05-09
1397.96	2021-04-22
1532.36	2021-12-20

Below the result grid is the 'Action Output' table:

	Time	Action	Response	Duration / Fetch Time
27...	13:18:30	SELECT Sales_total, Sale_d...	5 row(s) returned	0.0035 sec / 0.00000...

98 -- What is the average sales per country? It presents the results sorted from highest to lowest average.  
99 • **SELECT** country 'Countries', ROUND(AVG(amount), 2) **AS** Average\_sales  
100 **FROM** transactions.company **AS** t1  
101 **JOIN** transactions.transaction **AS** t2  
102 **ON** t1.id = t2.company\_id  
103 **WHERE** declined = 0  
104 **GROUP BY** country  
105 **ORDER BY** Average\_sales **DESC**;

100% 22:107

**Result Grid** Filter Rows: Search Export:

Countries	Average_sales
United States	287.53
Ireland	285.83
Sweden	276.67
United Kingdom	271.77
Canada	261.94
Belgium	255.22

Result 93 Read Only

Action Output

Time	Action	Response	Duration / Fetch Time
28... 13:53:47	SELECT country 'Countries',...	15 row(s) returned	0.0045 sec / 0.00001...

The list of all transactions carried out by companies that are located in the same country as 'Non Institute' company:

110 -- with join  
111 • **SELECT** \*  
112 **FROM** transaction **AS** t1  
113 **JOIN** company **AS** t2  
114 **ON** t2.id = t1.company\_id  
115 **WHERE** country = (  
116 **SELECT** country **FROM** company  
117 **WHERE** company\_name **LIKE** 'Non Institute'  
118 )  
119 **AND** company\_name <> 'Non Institute'  
120 **ORDER BY** company\_name;

100% 18:115

**Result Grid** Filter Rows: Search Export:

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined	id	company_n
2A5A3001-104F-1D1F-78...	CcU-3169	b-2354	272	-67.2525	-142.557	2022-02-27 18:35:15	30.76	0	b-2354	Ac Libero In
9679E769-32DC-2591-B8...	CcU-3169	b-2354	272	47.6643	130.202	2021-06-28 11:22:10	186.34	1	b-2354	Ac Libero In
6D69D98A-F18A-99BD-B...	CcU-3204	b-2374	272	-52.3724	70.1522	2021-05-26 02:33:06	144.33	1	b-2374	Amet Faucit
E5078B1B-9591-E204-CC...	CcU-3204	b-2374	272	69.5434	98.8783	2021-09-06 01:29:42	220.85	0	b-2374	Amet Faucit
1479B3D2-B7BA-C7BB-4...	CcU-2994	b-2326	133	66.2672	172.399	2021-08-09 00:58:07	309.45	0	b-2326	Enim Condi
152598C2-029D-D684-4B...	CcU-2994	b-2326	126	-67.0189	-141.672	2021-07-05 03:10:00	395.43	0	b-2326	Enim Condi

Result 79 Read Only

Action Output

Time	Action	Response	Duration / Fetch Time
27... 13:28:08	SELECT * FROM transaction...	70 row(s) returned	0.0021 sec / 0.00005...



# Basic Queries

<https://github.com/leocareer>

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 5000 rows' dropdown. The SQL editor contains a query starting with a comment '-- with only subqueries'. The query selects all columns from the 'transaction' table, filtering by 'company\_id' in a subquery that selects 'id' from 'company' where 'country' is from a subquery selecting 'country' from 'company' where 'company\_name' is like 'Non Institute', and 'company\_name' is not equal to 'Non Institute'. The results grid shows columns: id, credit\_card\_id, company\_id, user\_id, lat, longitude, timestamp, amount, declined. The first row shows a transaction with id '2B928E1C-EC14-A760-0A...' and amount '383.73'. The action output shows the query executed at 13:39:10, returning 70 rows in 0.0014 seconds.

```
122 -- with only subqueries
123 • SELECT *
124 FROM transaction
125 WHERE company_id IN (
126     SELECT id FROM company
127     WHERE country = (
128         SELECT country FROM company
129         WHERE company_name LIKE 'Non Institute'
130     )
131     AND company_name <> 'Non Institute'
132 );
```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
2B928E1C-EC14-A760-0A...	CcU-2980	b-2246	275	-41.0496	161.685	2021-08-10 08:14:49	383.73	0
ACD2011A-A2B1-C365-41...	CcU-2980	b-2246	275	-54.4792	-82.7974	2022-03-05 20:41:20	60.07	1
4334349E-CEB0-3D68-A4...	CcU-3092	b-2310	275	-20.4859	150.87	2021-05-03 22:37:23	458.74	0
BC1B9A28-77B4-28CD-1...	CcU-2980	b-2246	275	-78.0305	18.5205	2021-10-18 07:07:05	477.05	1

transaction 85

Action Output

Time	Action	Response	Duration / Fetch Time
27... 13:39:10	SELECT * FROM transaction...	70 row(s) returned	0.0014 sec / 0.00007...

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 5000 rows' dropdown. The SQL editor contains a query starting with a comment '-- It presents the name, telephone, country, date and amount of those companies that made transactions with a value between 100 and 200 euros and on any of these dates: April 29, 2021, July 20, 2021 and March 13, 2022. Sort the results from highest to lowest amount.' The query selects 'company\_name', 'phone', 'country', 'DATE(timestamp) 'date'', and 'amount' from 'company' as 't1' joined with 'transaction' as 't2' on 't1.id = t2.company\_id', where 'DATE(timestamp)' is in ('2021-04-29', '2021-07-20', '2022-03-13') and 'amount' is between 100 and 200, ordered by 'amount' descending. The results grid shows columns: company\_name, phone, country, date, amount. The first row shows a company named 'Interdum Feugiat Sed Associates' with phone '04 88 40 32 52' and amount '164.86'. The action output shows the query executed at 13:43:01, returning 5 rows in 0.0019 seconds.

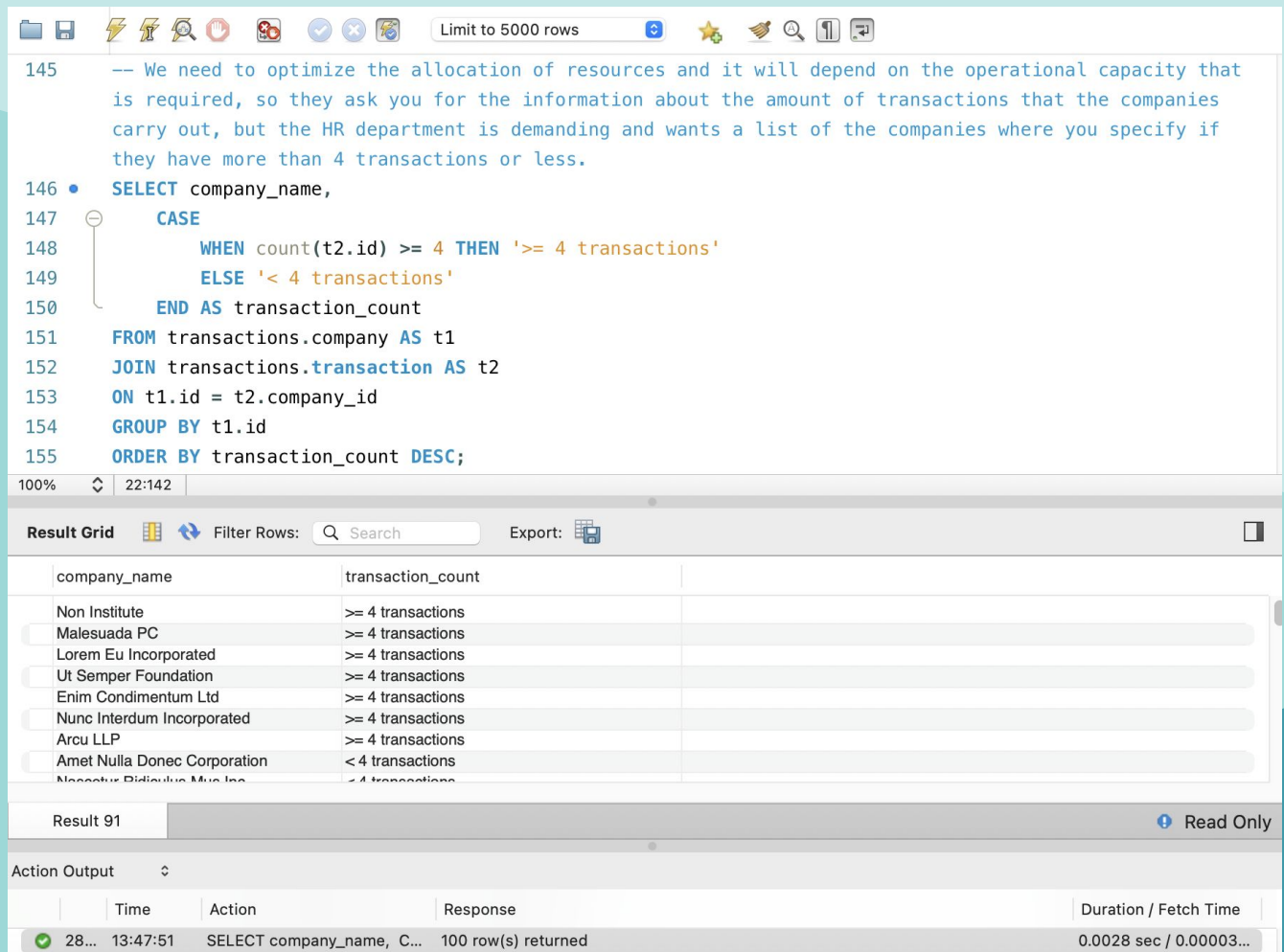
```
135 -- It presents the name, telephone, country, date and amount of those companies that made transactions
136 -- with a value between 100 and 200 euros and on any of these dates: April 29, 2021, July 20, 2021 and
137 -- March 13, 2022. Sort the results from highest to lowest amount.
138 • SELECT company_name, phone, country, DATE(timestamp) 'date', amount
139 FROM company AS t1
140 JOIN transaction AS t2
141 ON t1.id = t2.company_id
142 WHERE DATE(timestamp) IN ('2021-04-29', '2021-07-20', '2022-03-13')
143 AND amount BETWEEN 100 AND 200
144 ORDER BY amount DESC;
```

company_name	phone	country	date	amount
Interdum Feugiat Sed Associates	04 88 40 32 52	United Kingdom	2021-07-20	164.86
Nunc Interdum Incorporated	05 18 15 48 13	Germany	2022-03-13	164.32
Enim Condimentum Ltd	09 55 51 66 25	United Kingdom	2021-04-29	149.89
Lorem Eu Incorporated	01 83 66 62 07	Canada	2021-07-20	133.39
Nunc Interdum Incorporated	05 18 15 48 13	Germany	2021-04-29	111.51

Result 87

Action Output

Time	Action	Response	Duration / Fetch Time
27... 13:43:01	SELECT company_name, ph...	5 row(s) returned	0.0019 sec / 0.00001...



The screenshot shows a SQL IDE interface. At the top, there's a toolbar with various icons and a 'Limit to 5000 rows' dropdown. Below the toolbar, a SQL query is displayed with line numbers 145 to 155. The query is a SELECT statement that joins 'transactions.company' (t1) and 'transactions.transaction' (t2) on t1.id = t2.company\_id. It groups by t1.id and orders by transaction\_count in descending order. A CASE statement is used to categorize the transaction counts as '>= 4 transactions' or '< 4 transactions'.

```
145 -- We need to optimize the allocation of resources and it will depend on the operational capacity that
146 -- is required, so they ask you for the information about the amount of transactions that the companies
147 -- carry out, but the HR department is demanding and wants a list of the companies where you specify if
148 -- they have more than 4 transactions or less.
149
150 • SELECT company_name,
151       CASE
152         WHEN count(t2.id) >= 4 THEN '>= 4 transactions'
153         ELSE '< 4 transactions'
154       END AS transaction_count
155 FROM transactions.company AS t1
156 JOIN transactions.transaction AS t2
157 ON t1.id = t2.company_id
158 GROUP BY t1.id
159 ORDER BY transaction_count DESC;
```

Below the query editor, the 'Result Grid' is visible. It shows a table with two columns: 'company\_name' and 'transaction\_count'. The results are as follows:

company_name	transaction_count
Non Institute	>= 4 transactions
Malesuada PC	>= 4 transactions
Lorem Eu Incorporated	>= 4 transactions
Ut Semper Foundation	>= 4 transactions
Enim Condimentum Ltd	>= 4 transactions
Nunc Interdum Incorporated	>= 4 transactions
Arcu LLP	>= 4 transactions
Amet Nulla Donec Corporation	< 4 transactions
Nuncitur Biddulus Mus Inc	< 4 transactions

Below the result grid, the 'Action Output' section shows the execution details. It indicates that the query was executed at 13:47:51, returned 100 row(s), and took 0.0028 seconds to execute.

Time	Action	Response	Duration / Fetch Time
28... 13:47:51	SELECT company_name, C...	100 row(s) returned	0.0028 sec / 0.00003...