

# SPRINT #2

## RELATIONAL DATABASES AND INTRODUCTION TO SQL

student:  
Leo Kalugin



Date: 19/09/2024

### SUMMARY

In this sprint, I will review the basic concepts of working with relational databases. I will begin hands-on experience with a database containing information from a company dedicated to selling products online. In this activity, I will focus on data related to completed transactions and the corporate information of the companies involved.

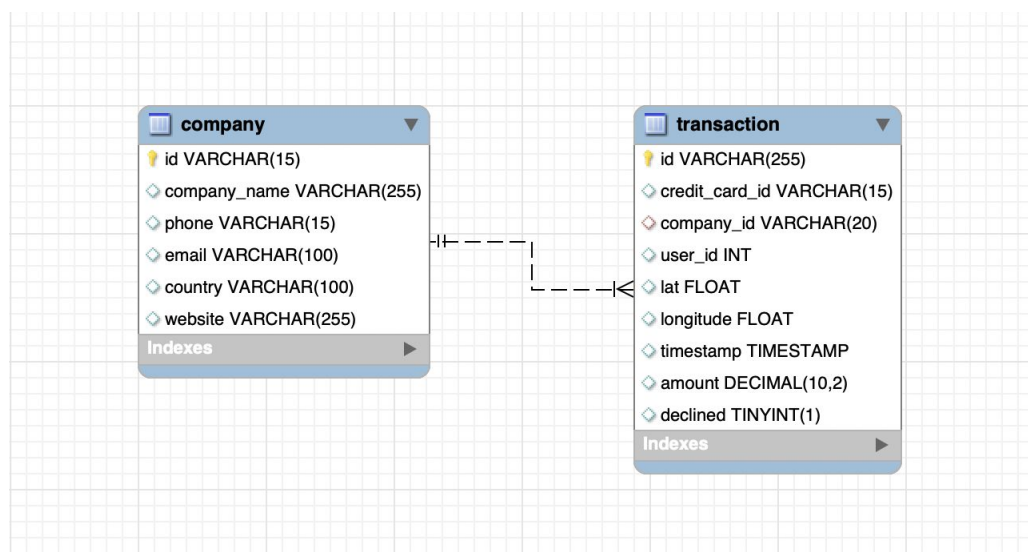
### RESULT

In this folder on the GitHub repository, you will find: the file **S2\_01.sql**, which contains all the scripts; **transactions\_schema.pdf**, which provides the database schema (also shown in this presentation); and this file, **Sprint\_2.pdf**, which includes screenshots of the work environment showing the queries I executed and the results obtained for each exercise:

[https://github.com/leocareer/DA\\_specialization/tree/main/Sprint%20%232](https://github.com/leocareer/DA_specialization/tree/main/Sprint%20%232)

### LEVEL 1 EXERCISE 1

From the attached documents (data\_structure.sql and input\_data.sql), import the two tables. It shows the main features of the created schema and explains the different tables and variables that exist. Be sure to include a diagram that illustrates the relationship between the different tables and variables.



## LEVEL 1 EXERCISE 2

Using JOIN you will perform the following queries:

- List of countries that are shopping;
- From how many countries the purchases are made;
- Identify the company with the highest average sales.

The screenshot shows a SQL query editor with the following query:

```
-- List of countries that are shopping
SELECT DISTINCT country
FROM transactions.company
ORDER BY country;
```

The result grid displays 15 countries:

country
Australia
Belgium
Canada
China
France
Germany
Ireland
Italy
Netherlands
New Zealand
Norway
Spain
Sweden
United Kingdom
United States

The Action Output shows: 219 12:02:48 S 15 row(s) returned, Duration / Fetch Time: 0.0026 sec / 0.00001...

The screenshot shows a SQL query editor with the following query:

```
-- From how many countries the purchases are made
SELECT count(DISTINCT country)
FROM transactions.company;
```

The result grid displays the count:

count(DISTINCT country)
15

The Action Output shows: 221 12:12:54 S 1 row(s) returned, Duration / Fetch Time: 0.00081 sec / 0.0000...

Limit to 5000 rows

```
13 -- Identify the company with the highest average sales (solution with 'limit')
14 • SELECT company_name, AVG(amount) AS avg_amount FROM transactions.transaction AS t1
15 JOIN transactions.company AS t2
16 ON t2.id = t1.company_id
17 GROUP BY t1.company_id
18 ORDER BY avg_amount DESC
19 LIMIT 1;
20
```

100% 22:11

Result Grid Filter Rows: Search Export: Fetch rows:

company_name	avg_amount
Eget Ipsum Ltd	473.075000

Result 136 Read Only

Action Output

	Time	Response	Duration / Fetch Time
222	12:21:04	S 1 row(s) returned	0.0078 sec / 0.00001...

Limit to 5000 rows

```
21 -- Identify the company with the highest average sales (solution without 'limit')
22 • SELECT company_name, avg_amount_t2
23 FROM (
24     SELECT company_id, AVG(amount) AS avg_amount_t2
25     FROM transactions.transaction
26     GROUP BY company_id
27 ) AS t2
28 JOIN company AS t1 ON t1.id = t2.company_id
29 WHERE avg_amount_t2 = (
30     SELECT max(avg_amount_t3)
31     FROM (
32         SELECT AVG(amount) AS avg_amount_t3 FROM transactions.transaction
33         GROUP BY company_id
34     ) AS t3
35 );
36
```

100% 19:18

Result Grid Filter Rows: Search Export:

company_name	avg_amount_...
Eget Ipsum Ltd	473.075000

Result 137 Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
223	12:25:55	SELECT company_name,...	1 row(s) returned	0.0067 sec / 0.00000...

## LEVEL 1 EXERCISE 3

Using only subqueries (without using JOIN):

- Show all transactions made by companies in Germany;
- List the companies that have made transactions for an amount higher than the average of all transactions;
- Companies that do not have registered transactions will be removed from the system, provide the list of these companies.

Limit to 5000 rows

```

40 -- Show all transactions made by companies in Germany
41 • SELECT *
42 FROM transactions.transaction
43 WHERE company_id IN (
44     SELECT id FROM transactions.company
45     WHERE country = 'Germany'
46 );
47

```

100% 14:51

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

id	credit_card...	company_id	user_id	lat	longitude	timestamp	amount	declined
108B1D1D-5B23-A76C-55EF-C568E49A05DD	CcU-2938	b-2222	275	83.7839	-178.86	2021-07-07 17:43:16	293.57	0
EA2C3281-C9C1-A387-44F8-729FB4B51C76	CcU-2938	b-2222	275	20.2004	-116.84	2021-05-09 10:25:08	119.36	1
0DD2E608-5C9E-D1B3-4999-B99F43AD735A	CcU-2959	b-2234	275	9.68811	130.282	2021-04-17 05:30:17	252.47	1
AB069F53-965E-A2A8-CE06-CA8C4FD92501	CcU-2959	b-2234	275	1.64819	-158.007	2021-04-15 13:37:18	60.99	0
0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	170	-43.9695	-117.525	2021-07-26 07:29:18	49.53	0
0A476ED9-0C13-1962-F87B-D3563924B539	CcU-4359	b-2302	221	-56.4901	114.801	2022-02-26 20:33:54	430.49	0
122DC333-E19F-D629-DCD8-9C54CF1EBB9A	CcU-4366	b-2302	221	29.6372	-166.173	2021-06-09 06:04:14	172.01	0
135267BA-2E7D-957C-C42C-6450A2B3ED54	CcU-4520	b-2302	210	20.6724	14.9732	2021-12-29 20:38:23	17.97	0
14CAE5B5-8FB1-3E4A-4C85-0EA167534F4	CcU-4849	b-2302	189	-53.6202	93.0533	2021-12-31 00:29:42	388.04	0
158A3ACB-541C-DBCC-65BD-6373CC67BF1C	CcU-4849	b-2302	183	42.5424	-170.347	2022-03-08 05:02:19	240.29	0
162C7E78-2B6B-7971-A1E4-D2124E732451	CcU-4527	b-2302	210	-69.1381	58.0017	2021-04-11 05:59:18	231.26	0
1717FD6B-ADAD-7082-A748-9112BE892CCC	CcU-4219	b-2302	172	69.4892	-138.411	2021-12-29 16:18:54	249.91	0
1753A288-9FC1-52E6-5C39-A1FFB97B0D3A	CcU-4345	b-2302	222	57.9422	-114.729	2021-08-17 05:32:08	497.84	0

transaction 138 Apply

**Action Output**

	Time	Action	Response	Duration / Fetch Time
224	12:31:47	SELECT * FROM transact...	118 row(s) returned	0.0019 sec / 0.00003...

Limit to 5000 rows

```
48 -- List the companies that have made transactions for an amount higher than the average of all transactions
49 • SELECT company_name
50   FROM transactions.company
51  WHERE id IN (
52      SELECT company_id
53      FROM transactions.transaction
54      GROUP BY company_id
55      HAVING sum(amount) > (
56          SELECT AVG(comp_amount)
57          FROM (
58              SELECT sum(amount) AS comp_amount FROM transactions.transaction
59              GROUP BY company_id
60          ) AS t1
61      )
62  )
```

100% 16:45

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

company_name
Arcu LLP
Nunc Interdum Incorporated
Enim Condimentum Ltd
Ut Semper Foundation
Lorem Eu Incorporated
Malesuada PC
Non Institute

company 139 Read Only

**Action Output**

	Time	Action	Response	Duration / Fetch Time
✓ 225	12:33:14	SELECT company_name...	7 row(s) returned	0.0069 sec / 0.0000...

Limit to 5000 rows

```
64 -- Companies that do not have registered transactions will be removed from the system, provide the list of
65 • SELECT company_name
66   FROM transactions.company
67  WHERE id NOT IN (
68      SELECT company_id FROM transactions.transaction
69  );
70
```

100% 5:60

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

company_name
--------------

company 140 Read Only

**Action Output**

	Time	Action	Response	Duration / Fetch Time
✓ 226	12:36:25	SELECT company_name...	0 row(s) returned	0.0017 sec / 0.0000...

## LEVEL 2 EXERCISE 1

Identify 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 with a query editor and a result grid. The query is as follows:

```
-- with 'limit'
77 • SELECT sum(amount), DATE(timestamp)
78 FROM transactions.transaction
79 GROUP BY DATE(timestamp)
80 ORDER BY sum(amount) DESC
81 LIMIT 5;
82
```

The result grid displays the following data:

sum(amount)	DATE(timestamp)
1564.87	2021-03-29
1532.36	2021-12-20
1469.90	2021-06-15
1463.73	2021-05-09
1443.11	2021-06-21

The interface also shows an Action Output table with the following entry:

	Time	Action	Response	Duration / Fetch Time
227	12:40:14	SELECT sum(amount), DA...	5 row(s) returned	0.0034 sec / 0.00000...

The screenshot shows a SQL IDE interface with a query editor and a result grid. The query is as follows:

```
-- with window function
84 • SELECT sum_amount, date_timestamp
85 FROM (
86     SELECT sum(amount) AS sum_amount, DATE(timestamp) AS date_timestamp,
87     ROW_NUMBER() OVER(ORDER BY sum(amount) DESC) AS ind_amount
88     FROM transactions.transaction
89     GROUP BY date_timestamp
90 ) AS t
91 WHERE ind_amount <= 5
92 ORDER BY date_timestamp;
93
```

The result grid displays the following data:

sum_amount	date_timestamp
1564.87	2021-03-29
1463.73	2021-05-09
1469.90	2021-06-15
1443.11	2021-06-21
1532.36	2021-12-20

The interface also shows an Action Output table with the following entry:

	Time	Action	Response	Duration / Fetch Time
229	12:42:49	SELECT sum_amount, dat...	5 row(s) returned	0.0034 sec / 0.00000...



## LEVEL 2 EXERCISE 2

What is the average sales per country? It presents the results sorted from highest to lowest average.

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and settings, along with a 'Limit to 5000 rows' dropdown. The SQL editor contains the following query:

```
95 -- What is the average sales per country? It presents the results sorted from highest to lowest average.
96 • SELECT AVG(amount) AS country_amount, country
97 FROM transactions.company AS t1
98 JOIN transactions.transaction AS t2
99 ON t1.id = t2.company_id
100 GROUP BY country
101 ORDER BY country_amount;
102
```

Below the editor is a 'Result Grid' section with a search bar and an 'Export' button. It displays a table with two columns: 'country\_amou...' and 'country'. The table contains 15 rows of data, sorted by average amount in descending order.

country_amou...	country
53.400000	Spain
179.198333	France
201.590000	Italy
222.277273	New Zealand
227.556667	China
228.147692	Belgium
232.052500	Australia
244.203220	Germany
253.017778	Netherlands
254.216324	Norway
260.615063	Sweden
269.647869	Canada
270.731700	United Kingdom
277.308387	Ireland
309.179412	United States

Below the result grid is an 'Action Output' section with a 'Read Only' button. It displays a table with four columns: 'Time', 'Action', 'Response', and 'Duration / Fetch Time'. The table contains one row of data.

Time	Action	Response	Duration / Fetch Time
230 12:47:03	SELECT AVG(amount) A...	15 row(s) returned	0.0055 sec / 0.00001...

## LEVEL 2 EXERCISE 3

In your company, a new project is being considered to launch some advertising campaigns to compete with the 'Non Institute' company. For this, they ask you for the list of all transactions carried out by companies that are located in the same country as this company.

- Display the list by applying JOIN and subqueries;
- Display the listing by applying only subqueries.

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, a 'Limit to 5000 rows' dropdown, and other utility icons. The SQL editor contains the following query:

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

Below the editor is the 'Result Grid' section, which displays the query results in a table:

company_name
Ac Libero Inc.
Amet Faucibus Ut Foundation
Enim Condimentum Ltd
Interdum Feugiat Sed Associates
Non Magna LLC
Orci Adipiscing Limited
Sed Nunc Ltd
Viverra Donec Foundation

At the bottom, the 'Action Output' section shows a log of database actions:

	Time	Action	Response	Duration / Fetch Time
✓ 231	12:51:19	SELECT DISTINCT comp...	8 row(s) returned	0.0013 sec / 0.00000...



```
121  -- with only subqueries
122  •  SELECT company_name
123  FROM transactions.company AS t1
124  WHERE country IN (
125      SELECT country FROM transactions.company
126      WHERE company_name LIKE 'Non Institute'
127  )
128  AND company_name <> 'Non Institute'
129  AND EXISTS (
130      SELECT company_id FROM transactions.transaction AS t2
131      WHERE t1.id = t2.company_id
132  )
133  ORDER BY company_name;
134
```

100% 20:135

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

company_name
Ac Libero Inc.
Amet Faucibus Ut Foundation
Enim Condimentum Ltd
Interdum Feugiat Sed Associates
Non Magna LLC
Orci Adipiscing Limited
Sed Nunc Ltd
Viverra Donec Foundation

company 146 Read Only

**Action Output**

	Time	Action	Response	Duration / Fetch Time
✓ 232	12:52:40	SELECT company_name...	8 row(s) returned	0.0016 sec / 0.00000...

## LEVEL 3 EXERCISE 1

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 screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and search, along with a 'Limit to 5000 rows' dropdown. The SQL editor contains the following query:

```
136 -- It presents the name, telephone, country, date and amount of those companies that made transactions wit
137 -- euros and on any of these dates: April 29, 2021, July 20, 2021 and March 13, 2022. Sort the results fro
138 • SELECT company_name, phone, country, DATE(timestamp), amount
139 FROM transactions.company AS t1
140 JOIN transactions.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;
145
```

Below the editor, the 'Result Grid' shows the query results. It includes a search bar and an 'Export' button. The results are displayed in a table with 6 columns: company\_name, phone, country, DATE(timestamp), and amount. The table contains 5 rows of data.

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

Below the result grid, the 'Action Output' section shows the execution details. It includes a table with columns: Time, Action, Response, and Duration / Fetch Time. The output shows that the query was executed at 12:55:34, returned 5 rows, and took 0.0028 seconds to execute.

Time	Action	Response	Duration / Fetch Time
233 12:55:34	SELECT company_name,...	5 row(s) returned	0.0028 sec / 0.00001...

## LEVEL 3 EXERCISE 2

We need to optimize the allocation of resources and it will depend on the operational capacity that is required, so they ask you for the information about the amount of transactions that the companies carry out, but the HR department is demanding and wants a list of the companies where you specify if they have more than 4 transactions or less.

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 the following query:

```
147 -- We need to optimize the allocation of resources and it will depend on the operational capacity that is
148 -- so they ask you for the information about the amount of transactions that the companies carry out, but
149 -- is demanding and wants a list of the companies where you specify if they have more than 4 transactions
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 editor is a 'Result Grid' showing the query results. It has columns for '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
Nascetur Ridiculus Mus Inc.	<= 4 transactions

At the bottom, the 'Action Output' tab shows the execution details:

	Time	Action	Response	Duration / Fetch Time
234	12:58:21	SELECT company_name,...	100 row(s) returned	0.0026 sec / 0.00002...