

10 Days of SQL with MJ!

Check out this in-depth recap of MJ's thrilling 10-day SQL series! Dive into the excitement as we break down each day's lessons, exploring the ins and outs of SQL in a way that's both engaging and informative. Get ready for an adventure through databases like never before – MJ's SQL series is one you won't want to miss!

Day 1: Understanding the Basics

On [Day 1](#), we started our SQL journey with a fundamental query:

```
Unset
SELECT amount
FROM order_sheet
WHERE amount > 20;
```

This query served as a foundation, introducing key concepts like:

- SELECT: Picking out specific data.
- FROM: Identifying the data source.
- WHERE: Establishing conditions for refined data extraction.

These fundamental elements laid the groundwork for our exploration of more advanced SQL topics in the days that followed...

Day 2: Order of Execution

On [Day 2](#), our focus shifted to unravelling the sequence of execution in SQL queries. The order, as we discussed, went like this:

```
FROM
WHERE
GROUP BY
HAVING
SELECT
ORDER BY
LIMIT
```

To make this sequence memorable, we came up with a fun sentence: "Frogs wear green hats, singing 'Oh, let's leap!'" Crafting this with my colleague, Guled Egal, added a touch of humour, making it easier to recall. Feel free to create your own quirky phrase to solidify this order in your memory!

Day 3: Practical Exercises

Stepping beyond the theoretical realm, [Day 3](#) was all about putting theory into practice! We ventured into writing our own queries using a nifty website: [SQL Playground](#).

We kicked off with a straightforward query:

```
Unset
SELECT *
FROM products
```

This served as a starting point, allowing us to explore the data and gradually refine our queries by specifying the columns we wanted to extract. The key to learning? Embrace trial and error—so go ahead, sink your teeth into it, and explore the vast possibilities of these example datasets!

Day 4: Top 3 SQL Terms

On Saturday, we decided to take it easy for [Day 4](#) while still delving into the top three most-used SQL terms.

- **SQL:** SQL stands for "Structured Query Language" - the magic language for seamlessly interacting with databases.
- **Database:** Think of a database as a digital warehouse, meticulously storing all your data - a sort of folder containing your Excel files.
- **Tables:** And then there are tables - like superpowered spreadsheets - where data is neatly organised into rows and columns.

Day 5: SQL Data Types

On a relaxed Sunday, we continued our journey with a similar vibe as Saturday. [Day 5](#) was all about delving into different SQL data types - an essential key to comprehending how data is stored.

We kicked off with:

- **INT (Integer):** The go-to for whole numbers. Fun fact: being Dutch, I discovered there's no direct translation for "Integer" in Dutch. Integers are those straightforward whole numbers like 1, 2, 3, 4 - no decimals involved.

- **VARCHAR (Variable Character):** for flexible text containers. A Variable Character can hold a string of letters - perfect for names, email addresses, and cities.
- **DATE:** This one's a breeze - it holds dates. However, a little heads-up, in most databases, dates are stored as UNIX timestamps - a bunch of numbers telling the database what the date is. But fear not, you can convert these using `FROM_UNIXTIME('Your column')` to make them readable for us humans.

Day 6: Data Aggregations - COUNT, SUM, and AVG

As we crossed the halfway mark on [Day 6](#), we dove into some SQL functions that hold significant weight - COUNT, SUM, and AVG. These functions are the power players in data analysis, allowing us to manipulate data within the database.

COUNT: A game-changer for grasping the size of a dataset. For instance, if you use `COUNT(company_id)`, it tallies all the `company_id` values in the `company_id` column—an incredibly handy function frequently used in my queries.

SUM: Stepping into the realm of quantitative data, SUM takes the spotlight. We're all familiar with SUM - it operates similarly to Excel or Google Sheets. For instance, `SUM(order_total)` gives you the sum of all order totals.

AVG: Short for Average, this function provides the mean value of a set of numerical data. Want to know the average age of people in your database? Simply use `AVG(age)`, and voila, you're off to the races.

These aggregation functions are indispensable tools in the world of data analysis. Feel free to apply them to the test datasets you explored on Day 3!

Day 7: Exploring JOINS

JOINS! What a topic to kick [Day 7](#) off with! JOINS, these magical connectors, merge tables based on a common key (like `user_id`) and understanding how they function is key to effectively combining data from different tables.

Each type of JOIN warrants its paragraph:

INNER JOIN: This brings together rows from two tables based on a common key, revealing only the rows with matches. This often results in the smallest amount of data.

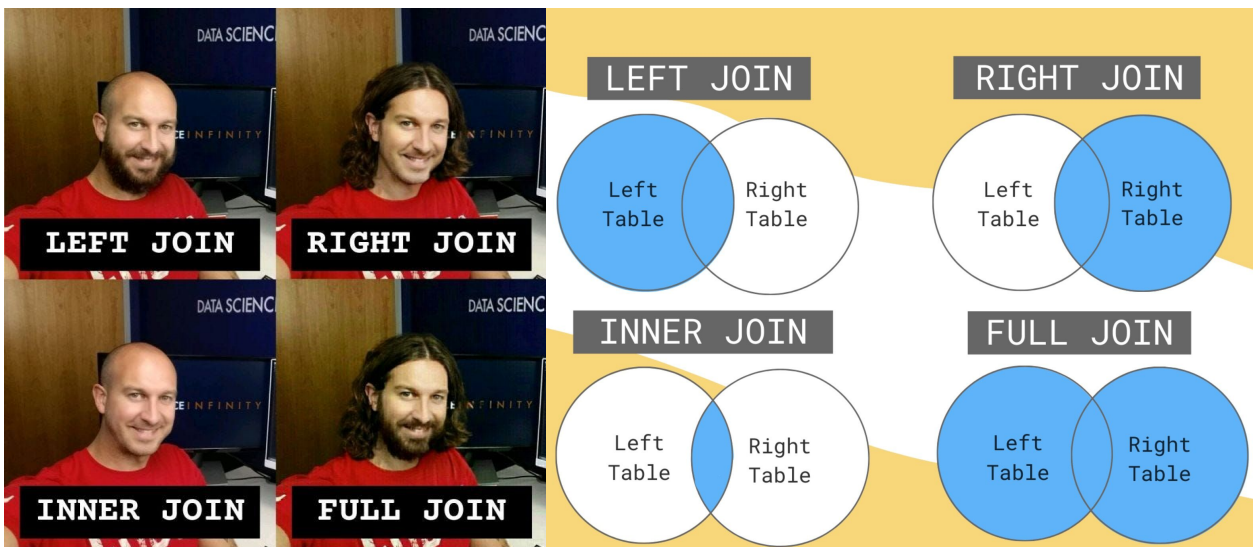
LEFT JOIN: Picture this as returning all rows from the left table and the matched rows from the right table. It brings all the left table's data and adds matched data from the right.

RIGHT JOIN: The opposite of the LEFT JOIN, it returns all rows from the right table and the matched rows from the left table.

FULL JOIN: It returns all rows when there's a match in either the left or right table, blending data from both tables.

Understanding these JOINS is like having a backstage pass to merge and mingle your data effectively.

But as the saying goes, a picture is worth a thousand words, so let this visual aid illustrate the concept even more effectively:



Day 8: Crafting Clean Queries

[Day 8:](#) One of my pet peeves: poorly structured SQL queries. Did you observe that in the examples we provided, we meticulously gave each element its proper place? SELECT, FROM, WHERE, and all those other significant statements each had their dedicated line.

As a data team expands, and you start sharing code with team members, the importance of crafting clean and clear code becomes evident.

Here are examples of messy and clean code:

Messy Code:

Unset

```
select user_id, amount, product, quantity, total from orders where quantity >= 10
and amount != 0 order by amount
```

Clean Code:

Unset

```
SELECT
user_id, -- Customer identifier
amount, -- Total amount spent on the product
product, -- Name of the ordered product
quantity, -- Quantity of the product ordered
total -- Total cost for the ordered quantity
FROM
orders -- Accessing order data
WHERE
quantity >= 10 -- Considering orders with quantity greater than or equal to 10
AND amount != 0 -- Excluding orders with a zero amount
ORDER BY
amount; -- Sorting results by amount in ascending order
```

By the way, that "--" is like a secret handshake in SQL - it's a comment, letting SQL know this part is just for humans. Sometimes we throw in a "#" for variety. Adding these comments and keeping our queries tidy not only enhances code quality but also fosters seamless collaboration.

Day 9: SQL Learning Resources

As we neared the end of our 10-day SQL series, I realised there's only so much I can cover. So, on [Day 9](#), I happily shared some valuable learning resources that have been my companions on this data journey. Here's a peek:

YouTube: Dive into YouTube tutorials, where channels like Alex Freberg dish out massive help in SQL, share data analytics skills, and offer fantastic advice on tackling common issues faced by data analysts.

Codecademy: Let me tell you, this isn't a sponsored post, but Codecademy has been one of the best courses I've ever taken. The beauty of Codecademy lies in learning a concept and immediately putting it to the test in a little exercise. I loved every bit of it and can't recommend it enough.

Data Community: Join vibrant communities, like Reddit's r/SQL, where you can ask questions, share your knowledge, and connect with fellow SQL enthusiasts. Consider

attending meetups or online classes to connect with other data enthusiasts. Personally, I found the Summer school from Carruthers and Jackson to be transformative - right up there with Codecademy - in shaping my day-to-day work life.

These resources have been my trusty companions, and I hope they guide you just as well on your data journey.

Day 10: Final Recap

On the final day of #10DaysOfSQL we had a quick recap on the last few days. Let's do a quick run down:

Day 1: We took our first steps into the world of data basics with our inaugural query.

Day 2: Explored the intriguing order of execution in SQL queries, using the memorable mantra, "Frogs Wearing Green Hats, Singing 'Oh, Let's leap!'"

Day 3: Penned our first query - putting theory into practice and getting our hands dirty.

Day 4: Unravelling the mystery of the top three most-used SQL terms, keeping it short and sweet on a Saturday.

Day 5: Explored different SQL data types on a laid-back Sunday.

Day 6: Went beyond the basics, diving into the world of COUNT(), SUM(), and AVG() functions.

Day 7: Joined the JOINS party, learning how to effectively combine data from different tables.

Day 8: Mastered the art of crafting clean and organised SQL queries.

Day 9: I shared some valuable learning resources to fuel your ongoing data and SQL journey.

And here we are, Day 10! It's been a blast creating this series, and I hope you enjoyed it too. Feel free to connect with [me](#) and [Signable](#) on LinkedIn, and here's to wishing you a fantastic 2024 filled with more SQL adventures!

Written by Marijn Quartel, Business Data Analyst Team Lead at Signable.