Using AI with SQL

Table of contents

Harnessing the power of ChatGPT in our programming adventure

Advantages of ChatGPT in programming

Areas of caution when using ChatGPT

Making the most of ChatGPT on our journey

Harnessing the power of ChatGPT in our programming adventure

Written in collaboration with ChatGPT.

Embarking on the journey of learning programming is an exciting venture in today's digital age. The ability to create websites, apps, games, or conduct data analysis is empowering. Yet, it can also be a road paved with challenges, from deciphering syntax errors to navigating logic puzzles. But what if we told you there's a co-pilot ready to help?

Enter ChatGPT, a digital assistant trained on vast amounts of technical text, including the nuances of numerous programming languages. But remember, while ChatGPT is a helpful resource, it's essential for us not to neglect the development of critical thinking and problem-solving abilities.

Large Language Models (LLMs) like ChatGPT are essentially sophisticated predictors, trained to anticipate the next word in a sequence based on the context of the previous words. They don't inherently "know" things, but rather draw from vast amounts of data they've been trained on. For instance, they've been finely honed on programming content from countless sources, including platforms like StackOverflow, which has amassed a treasure trove of coding knowledge over the decades. This gives LLMs a unique ability to generate coherent, contextually relevant, and technically accurate responses, especially in domains like programming.

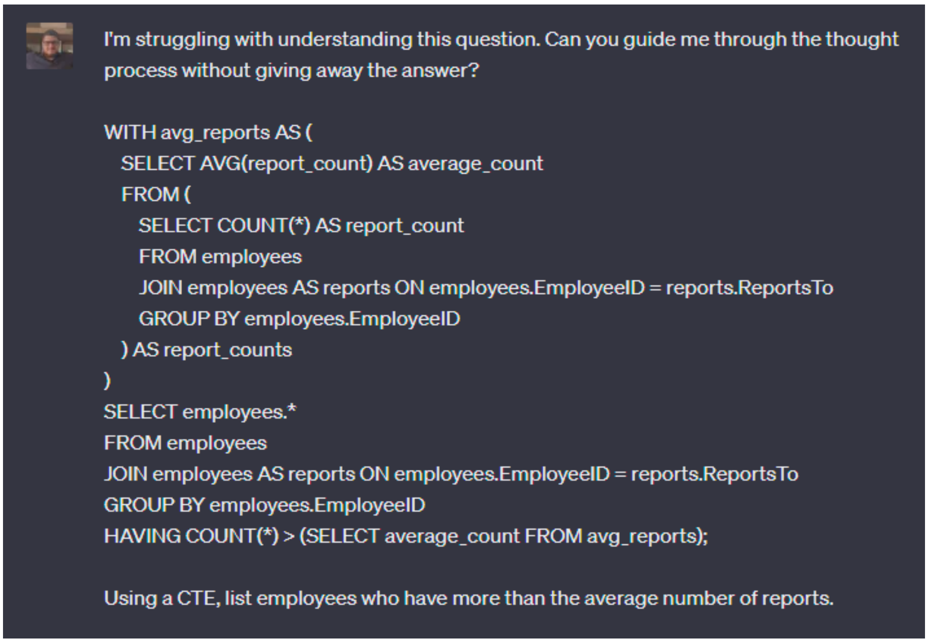
We stand at the cusp of a new era in our civilisation. Artificial Intelligence, represented here by ChatGPT, is set to redefine the way we interact with technology, learn, and even think. While the full breadth of AI's impact remains shrouded in the mists of the future, one thing is certain: just as in past technological revolutions, this era will present its challenges. Take, for instance, the Industrial Revolution. It ushered in unprecedented levels of mechanisation and productivity but also came with social upheavals, displacement of certain jobs, and an initial phase of adjustment. However, as history would have it, the opportunities it unlocked â€“ from creating new industries to enhancing global trade â€“ far outweighed its challenges.

As students and emerging technologists, it's vital for us to remember that, while AI and tools like ChatGPT will shape the landscape, our adaptability, creativity, and thirst for knowledge will determine our place within it.

Advantages of ChatGPT in programming

Let's break down what this digital helper can offer us:

- **Setup and use of tools**: Think of programming tools as magic boxes â€“ IDEs, compilers, libraries. ChatGPT can help us open these boxes, guiding us on how to set them up and use them effectively.
- **Help with designing and explaining code**: Sometimes, the hardest part is understanding why a piece of code does what it does. ChatGPT can help explain these complex ideas in simpler terms.
- **Assistance with code**: If we're stuck on how to start writing code, ChatGPT can help by turning our questions into beginner-friendly code snippets. It's like having a friend explain the basics to us.
- **Question:**

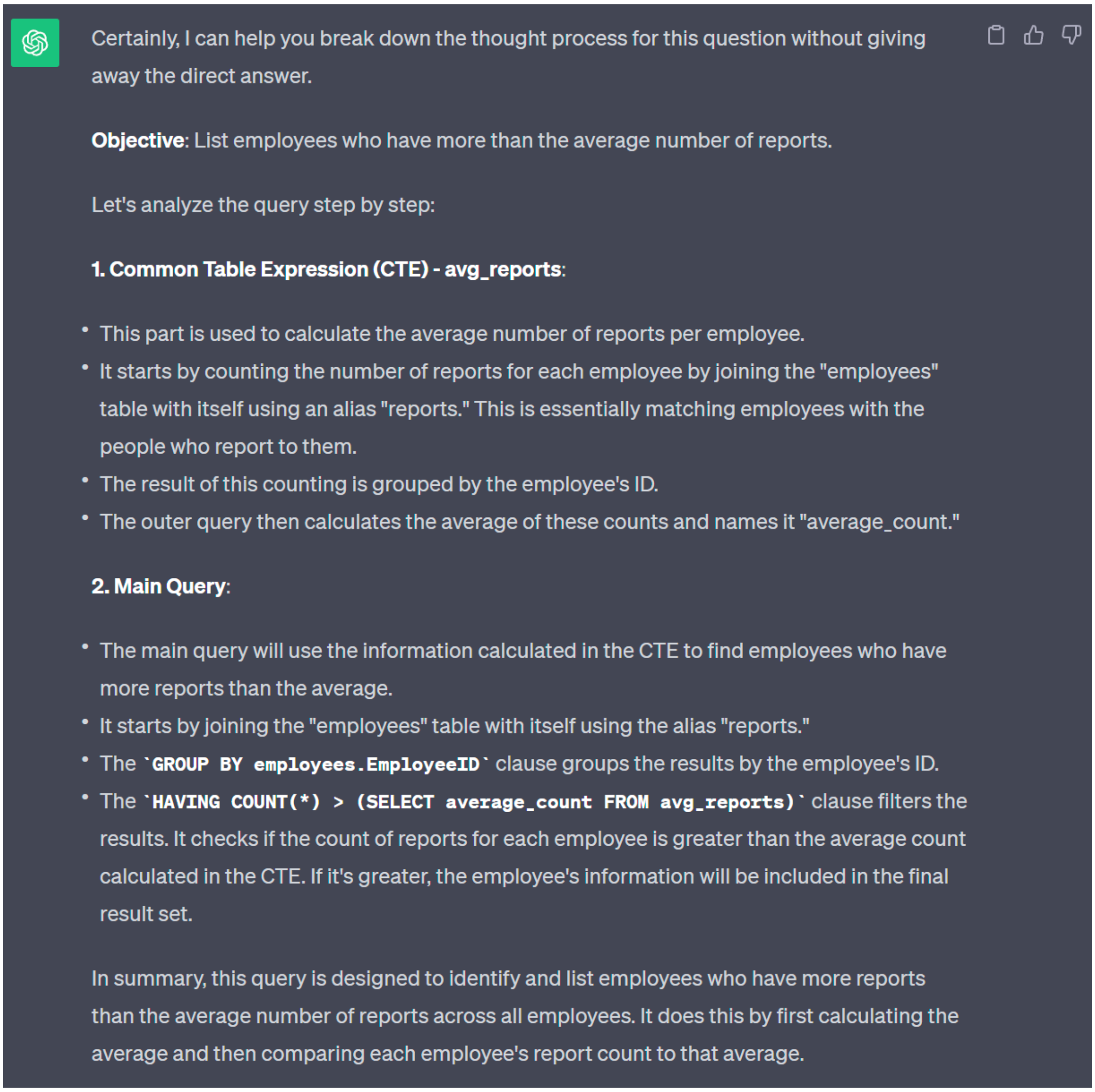


I'm struggling with understanding this question. Can you guide me through the thought process without giving away the answer?

```
WITH avg_reports AS (
  SELECT AVG(report_count) AS average_count
  FROM (
    SELECT COUNT(*) AS report_count
    FROM employees
    JOIN employees AS reports ON employees.EmployeeID = reports.ReportsTo
    GROUP BY employees.EmployeeID
  ) AS report_counts
)
SELECT employees.*
FROM employees
JOIN employees AS reports ON employees.EmployeeID = reports.ReportsTo
GROUP BY employees.EmployeeID
HAVING COUNT(*) > (SELECT average_count FROM avg_reports);
```

Using a CTE, list employees who have more than the average number of reports.

ChatGPT response:



Certainly, I can help you break down the thought process for this question without giving away the direct answer.

Objective: List employees who have more than the average number of reports.

Let's analyze the query step by step:

1. Common Table Expression (CTE) - avg_reports:

- This part is used to calculate the average number of reports per employee.
- It starts by counting the number of reports for each employee by joining the "employees" table with itself using an alias "reports." This is essentially matching employees with the people who report to them.
- The result of this counting is grouped by the employee's ID.
- The outer query then calculates the average of these counts and names it "average_count."

2. Main Query:

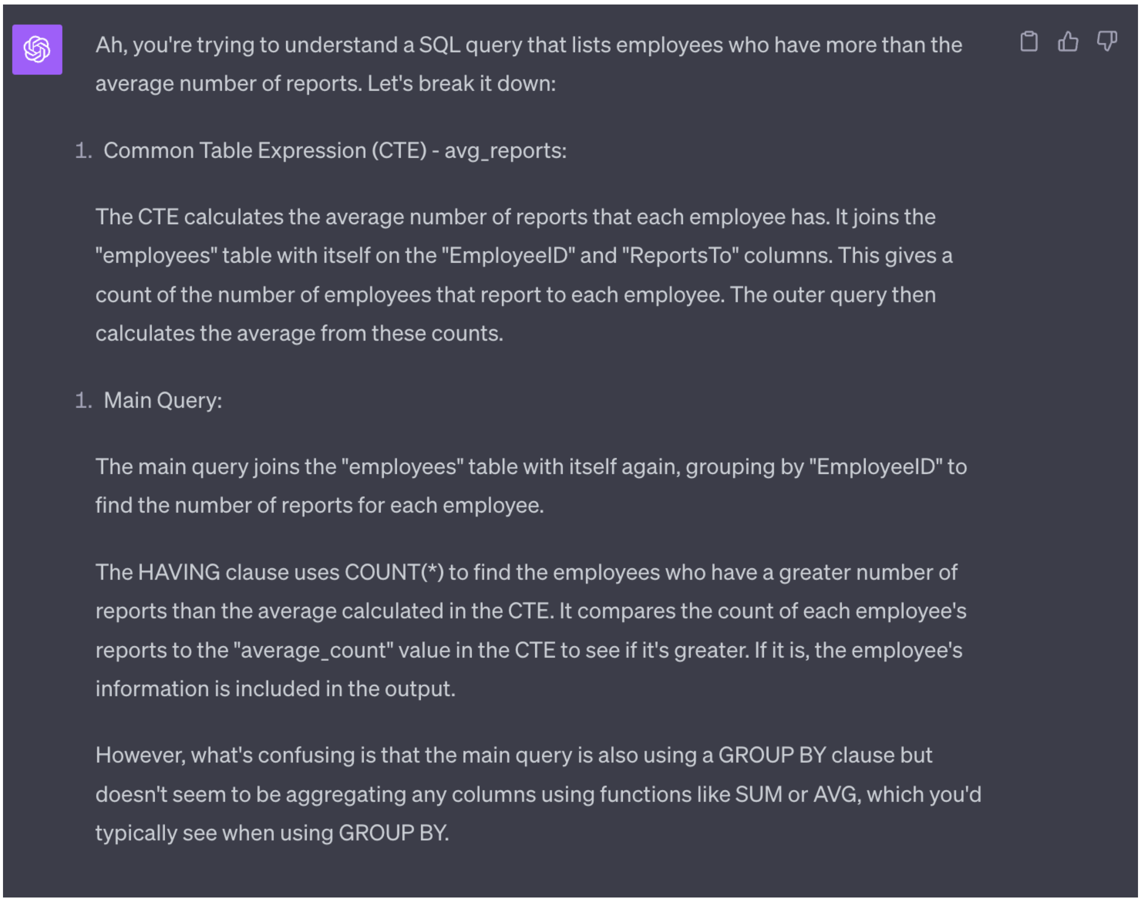
- The main query will use the information calculated in the CTE to find employees who have more reports than the average.
- It starts by joining the "employees" table with itself using the alias "reports."
- The `` `GROUP BY employees.EmployeeID` `` clause groups the results by the employee's ID.
- The `` `HAVING COUNT(*) > (SELECT average_count FROM avg_reports)` `` clause filters the results. It checks if the count of reports for each employee is greater than the average count calculated in the CTE. If it's greater, the employee's information will be included in the final result set.

In summary, this query is designed to identify and list employees who have more reports than the average number of reports across all employees. It does this by first calculating the average and then comparing each employee's report count to that average.

- **Fixing those pesky mistakes**: Everyone makes errors, especially when starting. ChatGPT can help point out where we might have gone wrong and suggest ways to fix it.

Areas of caution when using ChatGPT

While ChatGPT is a wonderful ally, it's also essential to be aware of where it might not be perfect. Using the same example, ChatGPT confidently gives us an incorrect answer:



Ah, you're trying to understand a SQL query that lists employees who have more than the average number of reports. Let's break it down:

1. Common Table Expression (CTE) - avg_reports:

   The CTE calculates the average number of reports that each employee has. It joins the "employees" table with itself on the "EmployeeID" and "ReportsTo" columns. This gives a count of the number of employees that report to each employee. The outer query then calculates the average from these counts.

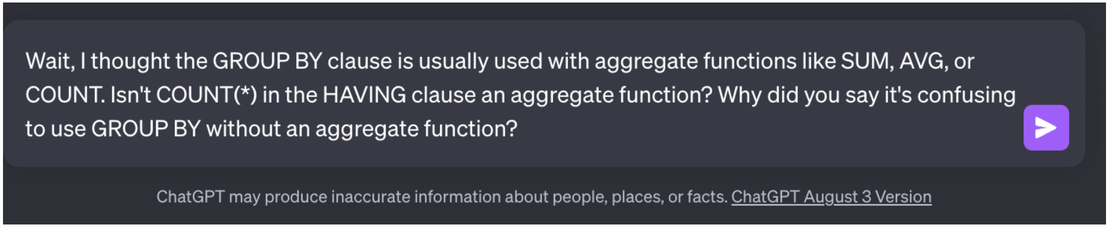
1. Main Query:

   The main query joins the "employees" table with itself again, grouping by "EmployeeID" to find the number of reports for each employee.

   The HAVING clause uses COUNT(*) to find the employees who have a greater number of reports than the average calculated in the CTE. It compares the count of each employee's reports to the "average_count" value in the CTE to see if it's greater. If it is, the employee's information is included in the output.

   However, what's confusing is that the main query is also using a GROUP BY clause but doesn't seem to be aggregating any columns using functions like SUM or AVG, which you'd typically see when using GROUP BY.

- **Not always spot-on**: Just like how sometimes even our smartest friend might make a mistake, ChatGPT can too. It's essential to double-check and understand what's being suggested.
  - Questioning the answers you get, arguing with it, or explaining your understanding to ChatGPT is a great way to improve the quality of the information you get.



Wait, I thought the GROUP BY clause is usually used with aggregate functions like SUM, AVG, or COUNT. Isn't COUNT(*) in the HAVING clause an aggregate function? Why did you say it's confusing to use GROUP BY without an aggregate function?

ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT August 3 Version

- **Be careful with personal information**: As we'd be cautious about what we share online, remember not to ask ChatGPT about personal or confidential project details.
- **Over-reliance can hamper growth**: Here's the thing â€“ while it's great to have help, if we lean on ChatGPT too much it might slow down our growth as programmers. We might miss out on developing critical thinking or the ability to troubleshoot problems on our own.

Understanding context: ChatGPT currently does not understand the structure or content of a database or the code you are using. It will confidently assist you but mistakingly provide incorrect advice. It is important to provide it with enough context to assist.

Making the most of ChatGPT on our journey

Navigating the labyrinth of programming might feel overwhelming at times, but with ChatGPT by our side, we have a valuable resource to tap into. Hereâ€™s how we can leverage this tool for a more enriching learning experience:

- **Seek clarity with simplicity**: Whenever a concept feels like it's over our heads, don't hesitate to ask ChatGPT. Remember, there's no shame in asking the bot to "explain things to us as beginners". The beauty of ChatGPT lies in its ability to distil complex ideas into more digestible information.
- **Dive deeper into mechanisms**: When a curiosity itch strikes and we want a comprehensive understanding, use the prompt:

  > "Can you break down how [specific concept] works in detail?"

  This will prime ChatGPT to elaborate, helping us get a firm grasp on the underlying principles.
- **Harness Socratic tutoring for problem solving**: There will be instances when we might feel stuck or fail to wrap our heads around a particular question, concept, or answer. In such scenarios, utilise the prompt:

  > "Iâ€™m struggling with understanding [specific concept]. Can you guide me through the thought process without giving away the answer?"

  ChatGPT, acting as our Socratic tutor, will guide us step by step, posing probing questions to not only help us arrive at the answer but also shed light on any misconceptions we might harbour.
- **Beyond just answers**: One pitfall students often fall into is the relentless pursuit of 'the right answer'. While answers are important, they're just the tip of the learning iceberg. Delve deeper. Ask 'why' and 'how'. Donâ€™t just settle for the surface-level explanation. Aim to understand the rationale, the logic, and the intricacies behind every solution. ChatGPT can be our companion in this exploration, but the onus of genuine understanding lies on us.

Remember, while ChatGPT is a formidable ally, the heart of our programming journey is our own curiosity, resilience, and the desire to understand the 'whys' behind the 'whats'. Keep questioning, keep learning!