Rohit Varma Follow

Feb 13 · 4 min read · Listen

Learn SQL using ChatGPT

And How can we utilize this fantastic tool to improve our skills in SQL

Structured Query Language (SQL) is a programming language used to manage and manipulate data in relational databases. It is a standardized language used by most database management systems and is a crucial tool for data analysis.

Learning SQL using ChatGPT can be a useful and interactive way to enhance your skills in data analysis.

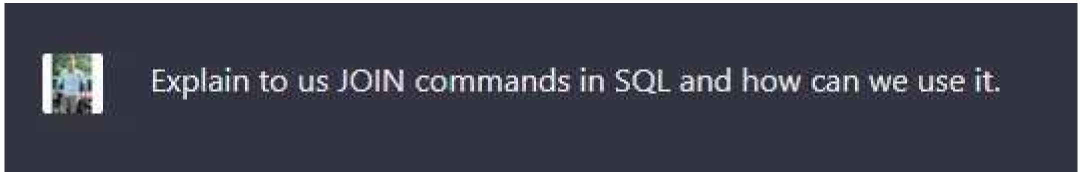
Below are the 5 ways to improve our SQL skills by using ChatGPT.

1. Explaining any concept or c QL

63

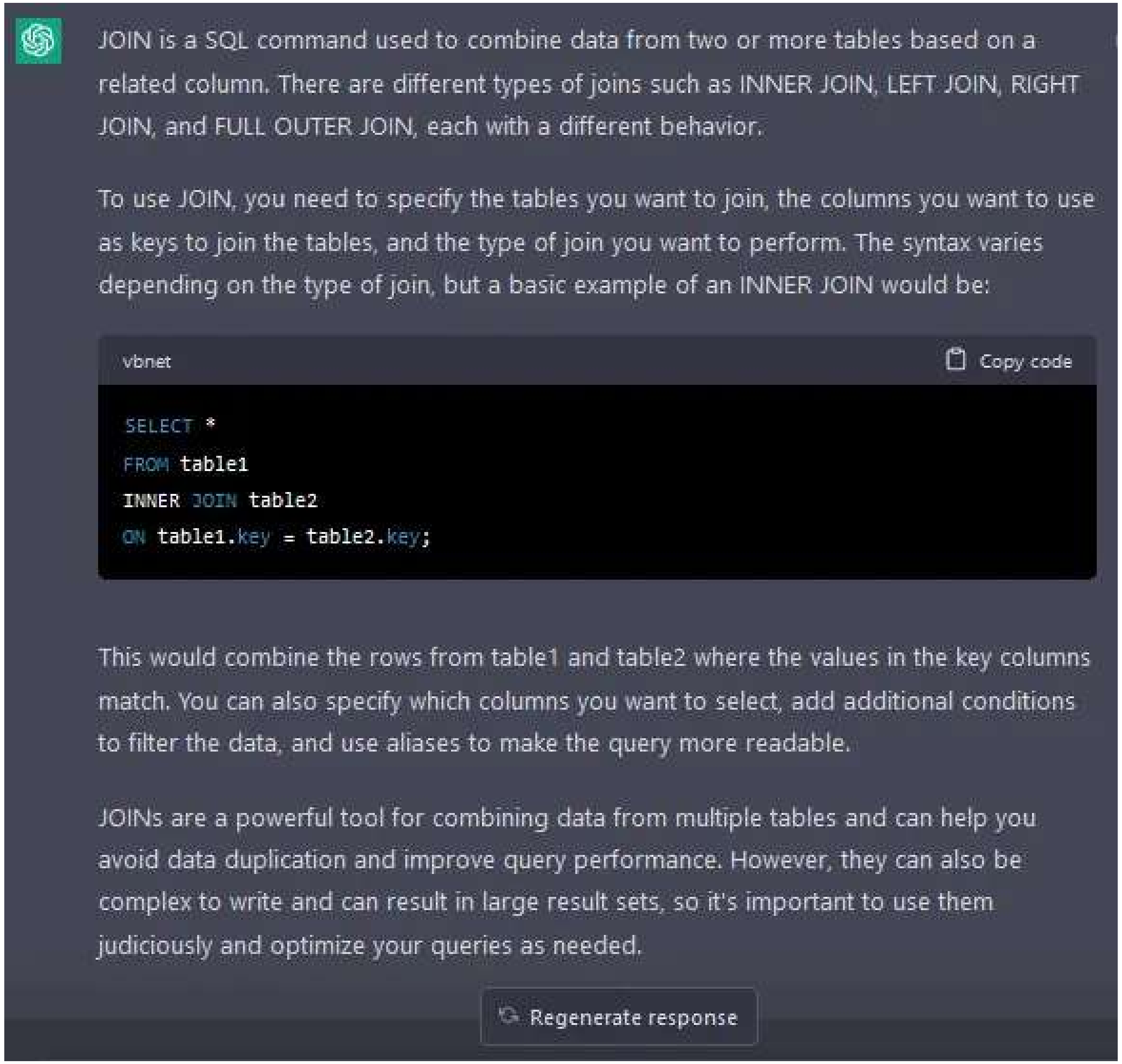
When learning a new programming language like SQL, it's common to feel confused and have doubts about different commands. ChatGPT can provide easy-to-understand explanations of SQL commands and provide multiple examples to help learners grasp their usage.

We'll start with JOIN SQL command.



Explain to us JOIN commands in SQL and how can we use it.

ChatGPT answers a detailed short answer explaining the `JOIN` command and gives the most basic of JOIN, it's types and how can we use all joins.



JOIN is a SQL command used to combine data from two or more tables based on a related column. There are different types of joins such as INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN, each with a different behavior.

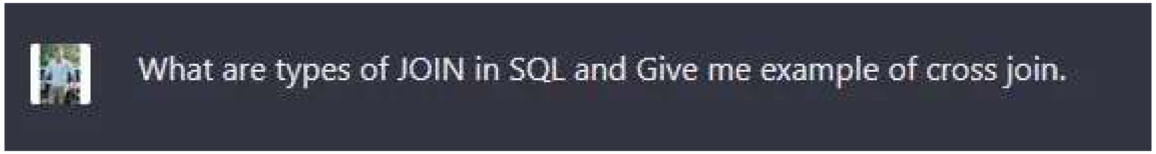
To use JOIN, you need to specify the tables you want to join, the columns you want to use as keys to join the tables, and the type of join you want to perform. The syntax varies depending on the type of join, but a basic example of an INNER JOIN would be:

```vbnet
SELECT *
FROM table1
INNER JOIN table2
ON table1.key = table2.key;
```

This would combine the rows from table1 and table2 where the values in the key columns match. You can also specify which columns you want to select, add additional conditions to filter the data, and use aliases to make the query more readable.

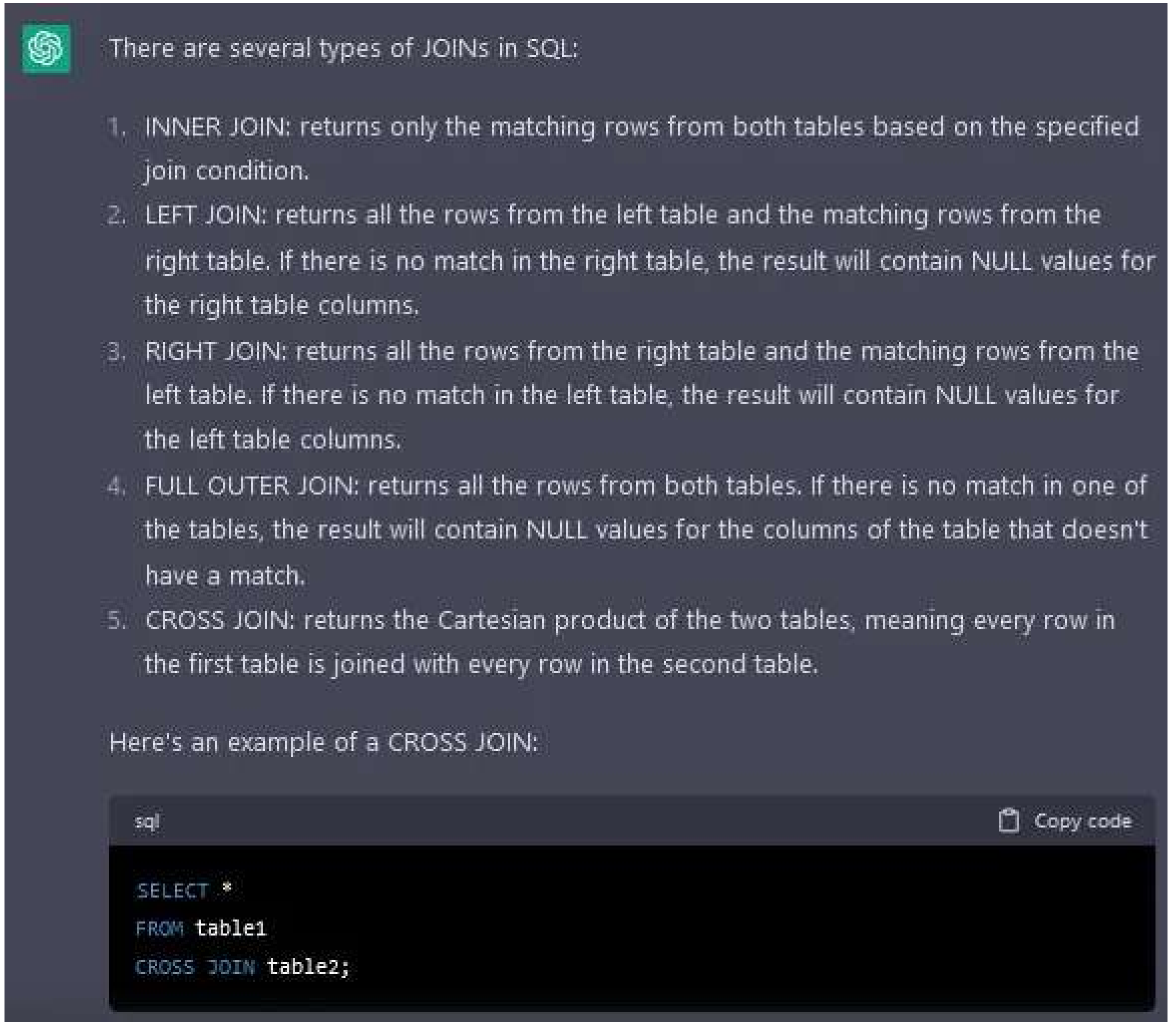
JOINs are a powerful tool for combining data from multiple tables and can help you avoid data duplication and improve query performance. However, they can also be complex to write and can result in large result sets, so it's important to use them judiciously and optimize your queries as needed.

However, I can further ask for the `JOIN` command.



What are types of JOIN in SQL and Give me example of cross join.

And as you can observe in the following image, ChatGPT explains us the types of JOIN in SQL and gives us some examples of cross join.

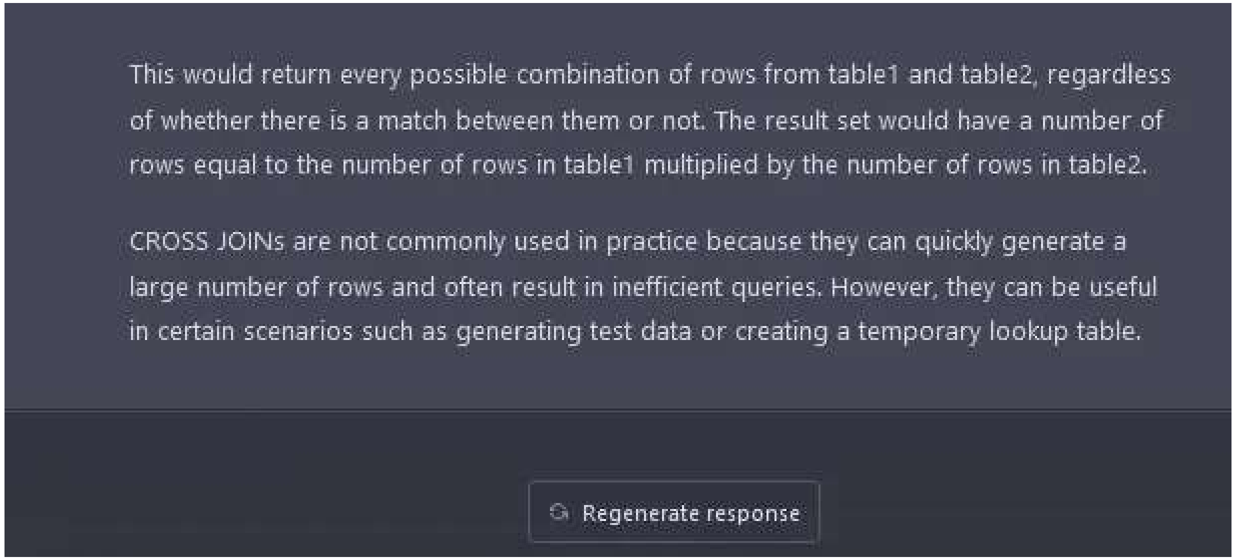


There are several types of JOINs in SQL:

1. INNER JOIN: returns only the matching rows from both tables based on the specified join condition.
2. LEFT JOIN: returns all the rows from the left table and the matching rows from the right table. If there is no match in the right table, the result will contain NULL values for the right table columns.
3. RIGHT JOIN: returns all the rows from the right table and the matching rows from the left table. If there is no match in the left table, the result will contain NULL values for the left table columns.
4. FULL OUTER JOIN: returns all the rows from both tables. If there is no match in one of the tables, the result will contain NULL values for the columns of the table that doesn't have a match.
5. CROSS JOIN: returns the Cartesian product of the two tables, meaning every row in the first table is joined with every row in the second table.

Here's an example of a CROSS JOIN:

```sql
SELECT *
FROM table1
CROSS JOIN table2;
```

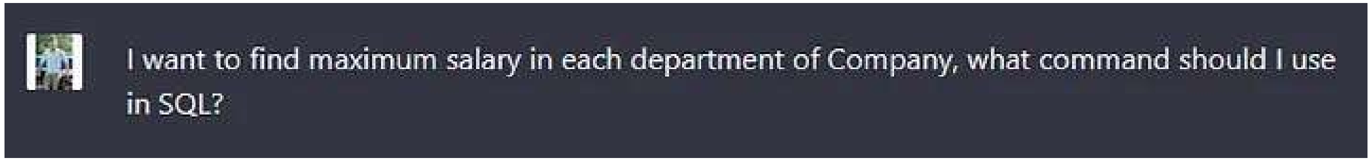


This would return every possible combination of rows from table1 and table2, regardless of whether there is a match between them or not. The result set would have a number of rows equal to the number of rows in table1 multiplied by the number of rows in table2.

CROSS JOINs are not commonly used in practice because they can quickly generate a large number of rows and often result in inefficient queries. However, they can be useful in certain scenarios such as generating test data or creating a temporary lookup table.

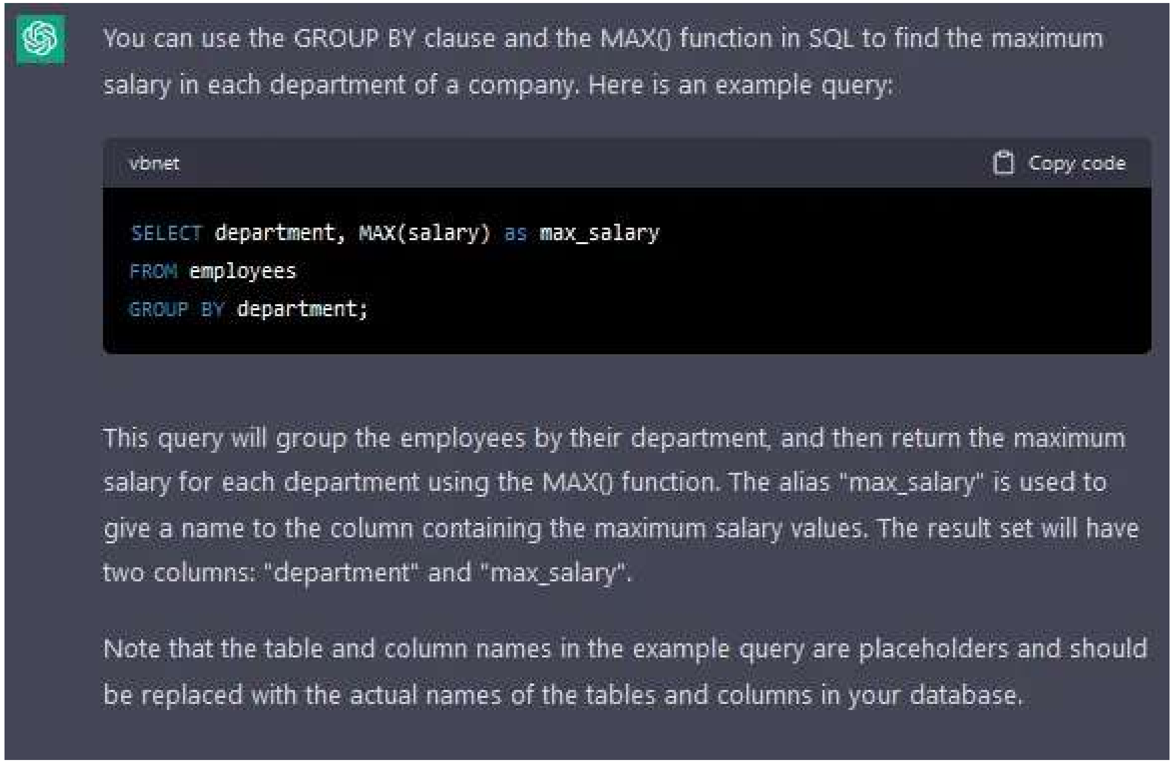
Try any other command you can think of, and it will answer right away.

2. You can ask how to do something in SQL and ChatGPT will let you know what command (or commands) to use.



I want to find maximum salary in each department of Company, what command should I use in SQL?

And ChatGPT answers me to use group by command, as you can observe in the following picture.



You can use the GROUP BY clause and the MAX() function in SQL to find the maximum salary in each department of a company. Here is an example query:

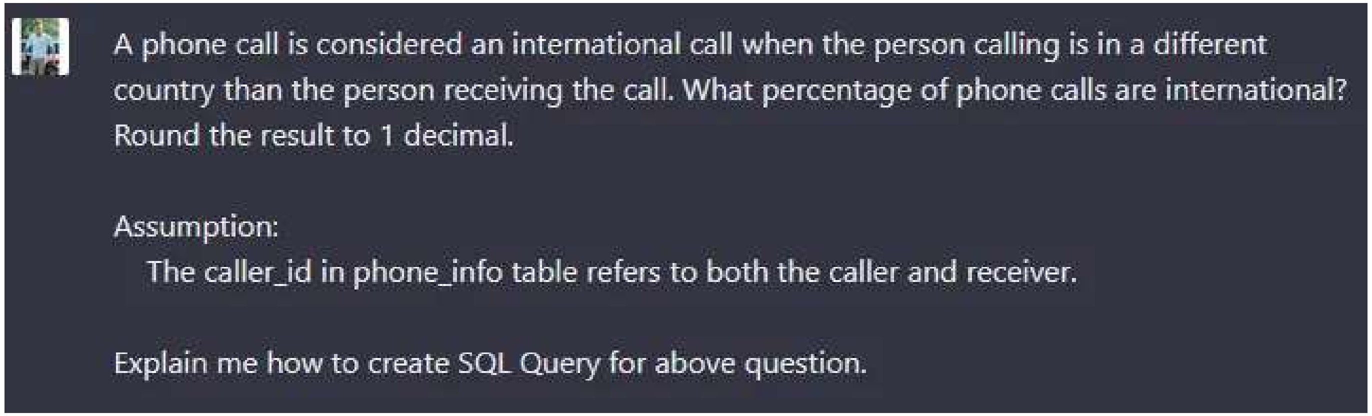
```
SELECT department, MAX(salary) as max_salary
FROM employees
GROUP BY department;
```

This query will group the employees by their department, and then return the maximum salary for each department using the MAX() function. The alias "max_salary" is used to give a name to the column containing the maximum salary values. The result set will have two columns: "department" and "max_salary".

Note that the table and column names in the example query are placeholders and should be replaced with the actual names of the tables and columns in your database.

3. You can ask ChatGPT to create a query using natural language

If you're unsure how to form a query but know the result you need, you can describe your intention to ChatGPT and it will provide a framework for you to follow. This will enable you to learn how to structure queries by observing the examples given by ChatGPT.

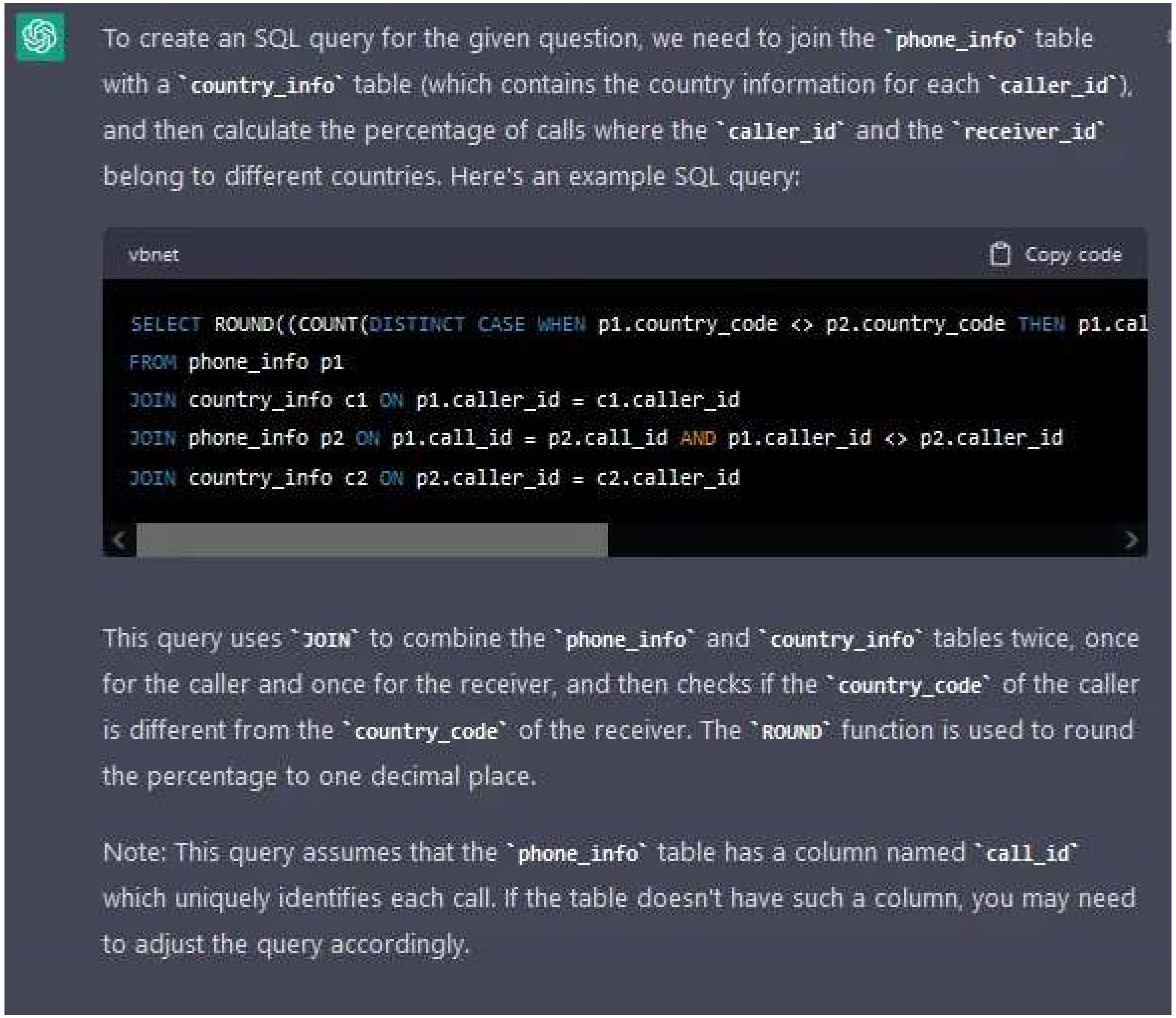


A phone call is considered an international call when the person calling is in a different country than the person receiving the call. What percentage of phone calls are international? Round the result to 1 decimal.

Assumption:
The caller_id in phone_info table refers to both the caller and receiver.

Explain me how to create SQL Query for above question.

ChatGPT answers me right away, as you can observe in the following image.



To create an SQL query for the given question, we need to join the `phone_info` table with a `country_info` table (which contains the country information for each `caller_id`), and then calculate the percentage of calls where the `caller_id` and the `receiver_id` belong to different countries. Here's an example SQL query:

```
SELECT ROUND((COUNT(DISTINCT CASE WHEN p1.country_code <> p2.country_code THEN p1.cal
FROM phone_info p1
JOIN country_info c1 ON p1.caller_id = c1.caller_id
JOIN phone_info p2 ON p1.call_id = p2.call_id AND p1.caller_id <> p2.caller_id
JOIN country_info c2 ON p2.caller_id = c2.caller_id
```

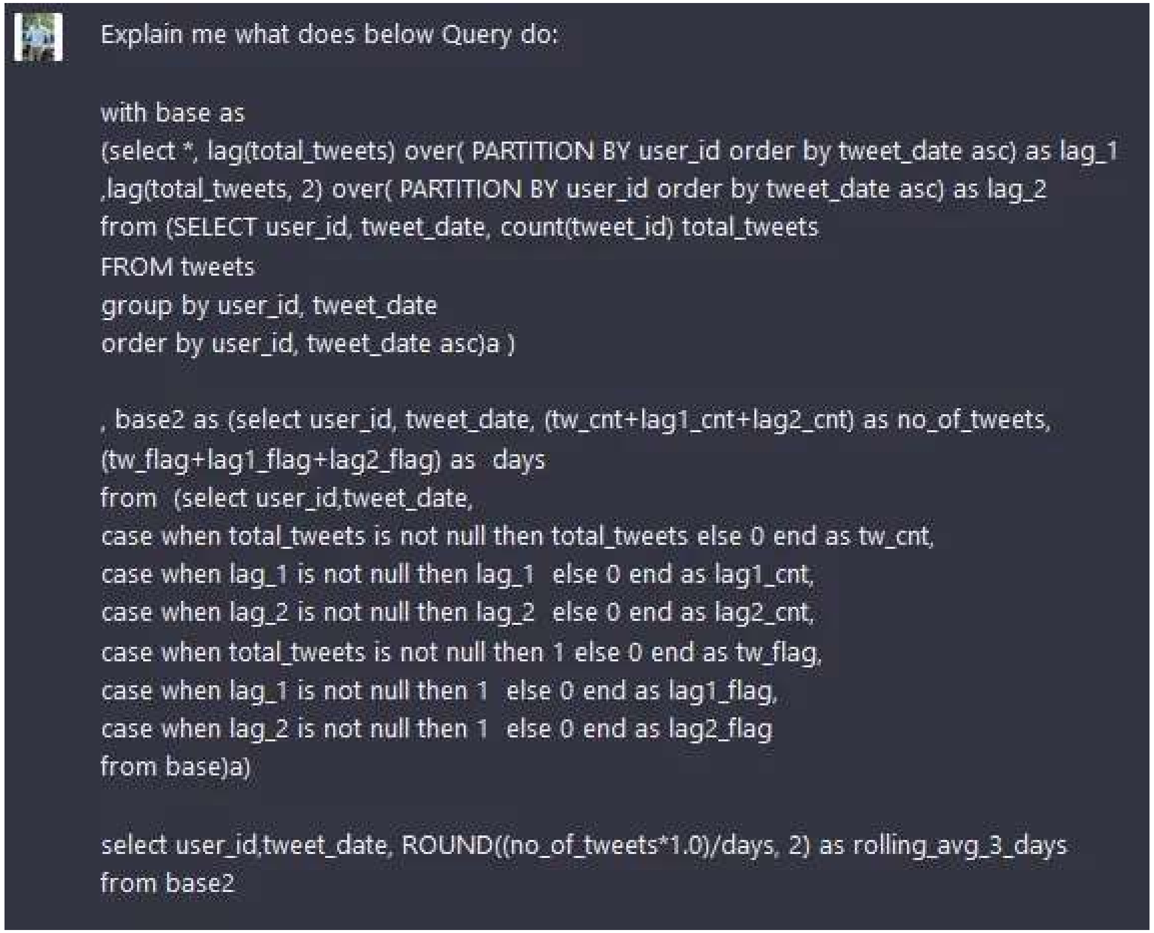
This query uses `JOIN` to combine the `phone_info` and `country_info` tables twice, once for the caller and once for the receiver, and then checks if the `country_code` of the caller is different from the `country_code` of the receiver. The `ROUND` function is used to round the percentage to one decimal place.

Note: This query assumes that the `phone_info` table has a column named `call_id` which uniquely identifies each call. If the table doesn't have such a column, you may need to adjust the query accordingly.

4. You can ask ChatGPT that explains you how a query works.

If we receive a query from a colleague that we don't understand, we can use ChatGPT to help explain it.

For example, if we have the following query and want to understand what it does:



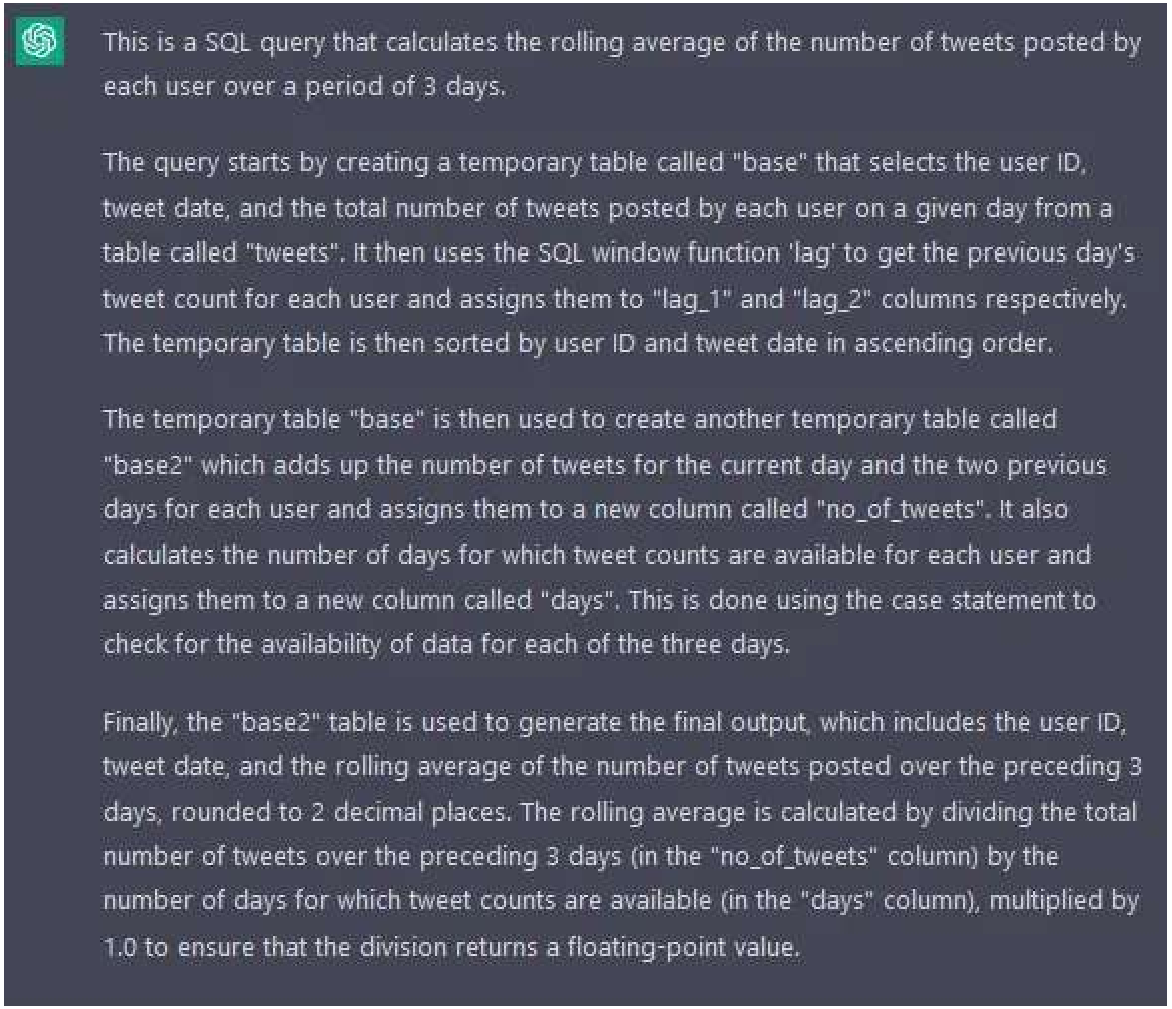
Explain me what does below Query do:

```
with base as
(select *, lag(total_tweets) over( PARTITION BY user_id order by tweet_date asc) as lag_1
,lag(total_tweets, 2) over( PARTITION BY user_id order by tweet_date asc) as lag_2
from (SELECT user_id, tweet_date, count(tweet_id) total_tweets
FROM tweets
group by user_id, tweet_date
order by user_id, tweet_date asc)a )

, base2 as (select user_id, tweet_date, (tw_cnt+lag1_cnt+lag2_cnt) as no_of_tweets,
(tw_flag+lag1_flag+lag2_flag) as  days
from  (select user_id,tweet_date,
case when total_tweets is not null then total_tweets else 0 end as tw_cnt,
case when lag_1 is not null then lag_1  else 0 end as lag1_cnt,
case when lag_2 is not null then lag_2  else 0 end as lag2_cnt,
case when total_tweets is not null then 1 else 0 end as tw_flag,
case when lag_1 is not null then 1  else 0 end as lag1_flag,
case when lag_2 is not null then 1  else 0 end as lag2_flag
from base)a)

select user_id,tweet_date, ROUND((no_of_tweets*1.0)/days, 2) as rolling_avg_3_days
from base2
```

ChatGPT just answers right away:



This is a SQL query that calculates the rolling average of the number of tweets posted by each user over a period of 3 days.

The query starts by creating a temporary table called "base" that selects the user ID, tweet date, and the total number of tweets posted by each user on a given day from a table called "tweets". It then uses the SQL window function 'lag' to get the previous day's tweet count for each user and assigns them to "lag_1" and "lag_2" columns respectively. The temporary table is then sorted by user ID and tweet date in ascending order.

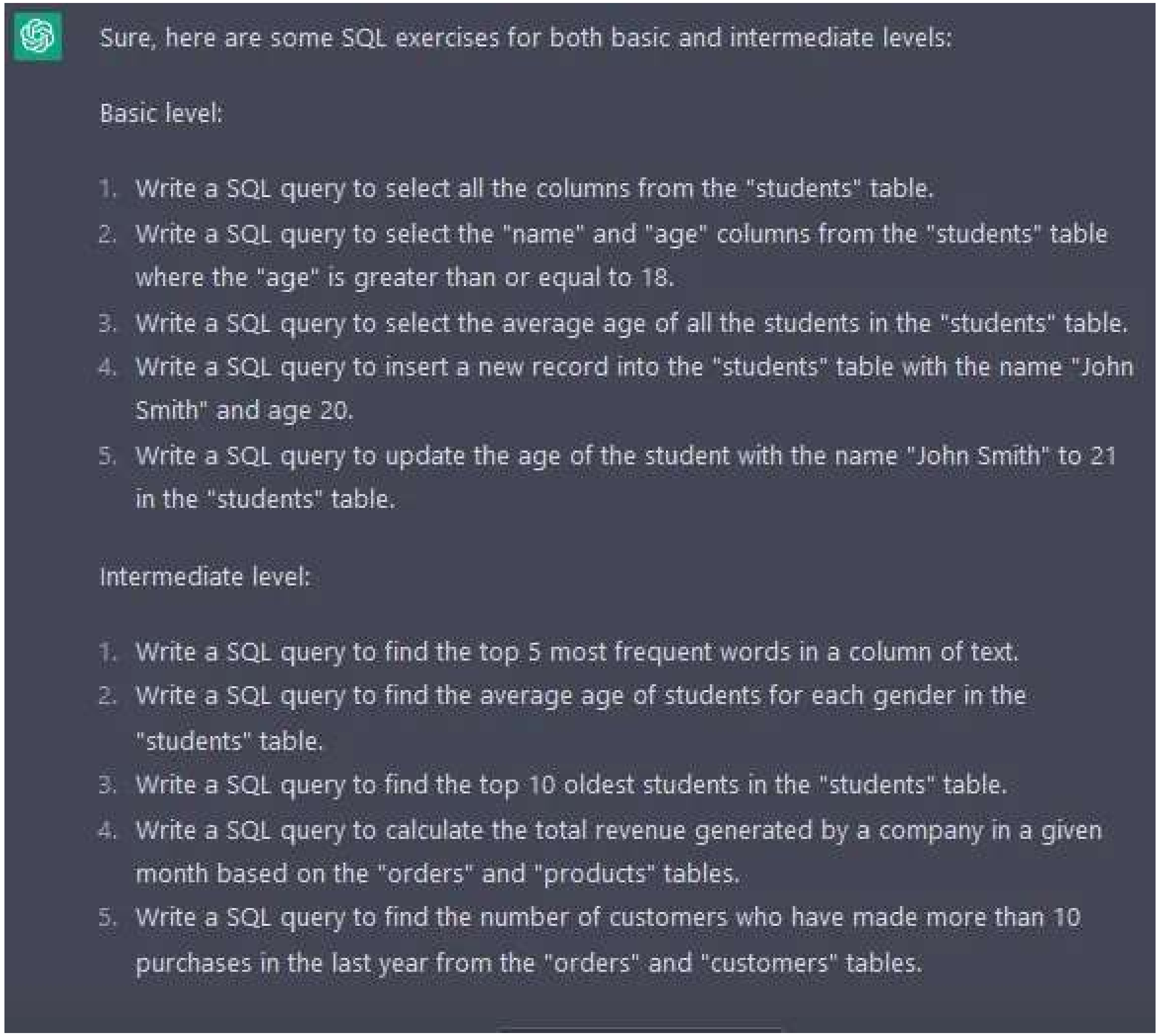
The temporary table "base" is then used to create another temporary table called "base2" which adds up the number of tweets for the current day and the two previous days for each user and assigns them to a new column called "no_of_tweets". It also calculates the number of days for which tweet counts are available for each user and assigns them to a new column called "days". This is done using the case statement to check for the availability of data for each of the three days.

Finally, the "base2" table is used to generate the final output, which includes the user ID, tweet date, and the rolling average of the number of tweets posted over the preceding 3 days, rounded to 2 decimal places. The rolling average is calculated by dividing the total number of tweets over the preceding 3 days (in the "no_of_tweets" column) by the number of days for which tweet counts are available (in the "days" column), multiplied by 1.0 to ensure that the division returns a floating-point value.

5. You can ask ChatGPT to challenge you with exercises.

Please give me basic and intermediate level exercise to practice SQL

ChatGPT suggested me questions on basic and intermediate level.



Sure, here are some SQL exercises for both basic and intermediate levels:

Basic level:

1. Write a SQL query to select all the columns from the "students" table.
2. Write a SQL query to select the "name" and "age" columns from the "students" table where the "age" is greater than or equal to 18.
3. Write a SQL query to select the average age of all the students in the "students" table.
4. Write a SQL query to insert a new record into the "students" table with the name "John Smith" and age 20.
5. Write a SQL query to update the age of the student with the name "John Smith" to 21 in the "students" table.

Intermediate level:

1. Write a SQL query to find the top 5 most frequent words in a column of text.
2. Write a SQL query to find the average age of students for each gender in the "students" table.
3. Write a SQL query to find the top 10 oldest students in the "students" table.
4. Write a SQL query to calculate the total revenue generated by a company in a given month based on the "orders" and "products" tables.
5. Write a SQL query to find the number of customers who have made more than 10 purchases in the last year from the "orders" and "customers" tables.

You can solve this questions and ask ChatGPT for evaluation.

Conclusion:

The use of SQL is highly beneficial in the current era of data-driven business. You can rely on ChatGPT to help you understand the fundamentals and refine your SQL skills. With consistent learning and practice, you can enhance your capabilities and take a significant leap forward in your data career with SQL.

Feel free to share any other noteworthy features you discover while interacting with ChatGPT in the comments section. Always remember to trust the data as it usually provides the best insights.

Give Claps if you found this helpful. Follow for more awesome content.

Sql Chatgpt Data Science Programming Data

Get an email whenever Rohit Varma publishes.

Your email

Subscribe

By signing up, you will create a Medium account if you don’t already have one. Review our Privacy Policy for more information about our privacy practices.

About Help Terms Privacy