



WITHOUT DATA, YOU'RE JUST ANOTHER PERSON WITH AN OPINION

Introduction to the common table expression (CTE)

A **Common Table Expression (CTE)** is a named temporary result set that you can reference within a `SELECT`, `INSERT`, `UPDATE`, or `DELETE` query. It is especially useful for breaking down complex queries into more manageable parts and for recursive queries. CTEs are created using the `WITH` keyword, followed by the name of the CTE and the query that defines it.

Similar to a derived table, a common table expression (CTE) is not stored as an object and lasts only during the query execution.

Unlike a derived table, a common table expression (CTE) can be self-referencing (in the case of a recursive CTE) or referenced multiple times within the same query. Moreover, a CTE offers enhanced readability and performance compared to a derived table.

MYSQL CTE syntax:

The structure of a CTE includes the name, an optional column list, and a query that defines the CTE. After you define a CTE, you can use like a view in the `SELECT`, `INSERT`, `UPDATE`, `DELETE`, or `CREATE VIEW` statement.

The following illustrates the basic syntax of a CTE:

```
WITH cte_name (column_list) AS  
(  
    query  
)
```

```
SELECT * FROM cte_name;
```

Code language: SQL (Structured Query Language) (sql)

In this syntax:

- `WITH cte_name (column_list) AS`: define a CTE with the name `cte_name` and a list of columns (`column_list`) that the CTE will have. The `column_list` is optional if you don't specify `column_list`, the CTE will inherit the column names from the result of the query.
- `query`: This is the query that defines the CTE. MySQL will store the result of the query in the CTE.
- `SELECT * FROM cte_name`: This is an example of how you can use the CTE. In this case, it is a simple `SELECT` statement that retrieves all columns from the CTE.



WITHOUT DATA, YOU'RE JUST ANOTHER PERSON WITH AN OPINION

Benefits of Using CTEs:

1. **Improved Readability:** Makes complex queries easier to read and maintain by breaking them into smaller logical parts.
2. **Reusable Result Sets:** You can reuse the CTE multiple times within the main query, avoiding the need to write the same logic repeatedly.
3. **Recursive Queries:** Enables recursive operations (e.g., working with hierarchical data like organizational structures or trees.)
4. **Temporary Nature:** CTEs exist only for the duration of the query, keeping your database clean.
5. **Better Organization:** Allows separation of logic into manageable components.

MySQL CTE examples:

Let's explore some examples of using MySQL CTE.

1) Basic MySQL CTE example

We'll use the customers table for demonstration:

customers
*customerNomer
customerName
contactLastName
contactFirstName
phone
addressLine1
addressLine2
city
state
postalCode
country
salesRepEmployeeNumber
creditLimit

The following example illustrates how to use a CTE for querying data from the customers table.

ANKKIT KUMAR GUPPTA

DATA ANALYST | PROMPT ENGINEER



WITHOUT DATA, YOU'RE JUST ANOTHER PERSON WITH AN OPINION

```
WITH customers_in_usa AS (SELECT customerName, state
FROM customers
WHERE country = 'USA'
)
SELECT customerName
FROM customers_in_usa
WHERE state = 'CA'
ORDER BY customerName;
```

```
+-----+
| customerName |
+-----+
| Boards & Toys Co. |
| Collectable Mini Designs Co. |
| Corporate Gift Ideas Co. |
| Men 'R' US Retailers, Ltd. |
| Mini Gifts Distributors Ltd. |
| Mini Wheels Co. |
| Signal Collectibles Ltd. |
| Technics Stores Inc. |
| The Sharp Gifts Warehouse |
| Toys4GrownUps.com |
| West Coast Collectables Co. |
+-----+
11 rows in set (0.00 sec)
```

How it works:

- First, define a CTE with the name customers_in_usa that stores the customer name and state of customers in the USA. The defining query retrieves data from the customers table.
- Second, select the customers located in California from the CTE.

2) Getting top sales using a CTE

We'll use the orders, orderdetails, and employees from the database:

The following example uses a CTE to retrieve the top 5 sales representatives based on their total sales in the year 2003:

```
WITH topsales2003 AS (SELECT salesRepEmployeeNumber employeeNumber,
SUM(quantityOrdered * priceEach) sales
FROM orders
INNER JOIN orderdetails USING (orderNumber)
INNER JOIN customers USING (customerNumber)
WHERE YEAR(shippedDate) = 2003 AND status = 'Shipped'
GROUP BY salesRepEmployeeNumber
ORDER BY sales DESC LIMIT 5 )

SELECT employeeNumber, firstName, lastName, sales
FROM employees
JOIN topsales2003 USING (employeeNumber);
```



WITHOUT DATA, YOU'RE JUST ANOTHER PERSON WITH AN OPINION

How it works:

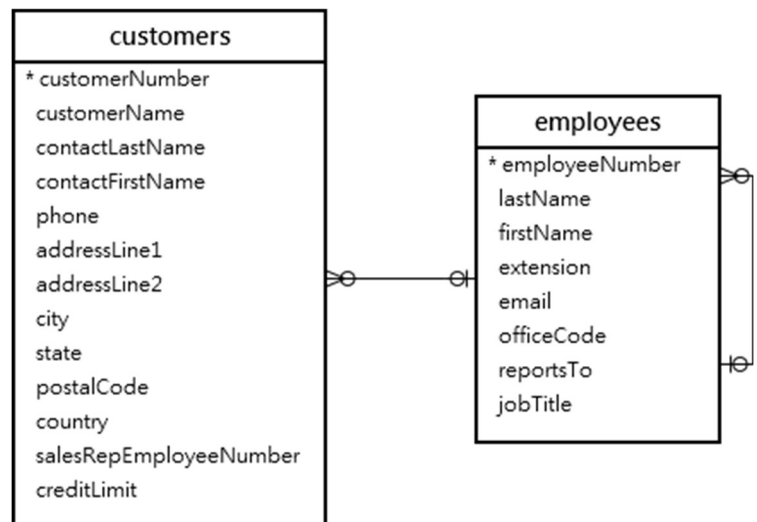
- First, define a CTE that retrieves the top 5 employees with their total sales in 2003.
- Second, join the CTE with the employees table to include the first and last names of the sales representatives.

employeeNumber	firstName	lastName	sales
1165	Leslie	Jennings	413219.85
1370	Gerard	Hernandez	295246.44
1401	Pamela	Castillo	289982.88
1621	Mami	Nishi	267249.40
1501	Larry	Bott	261536.95

5 rows in set (0.02 sec)

3) Using multiple CTEs:

We'll use the customers and employee's tables:



The following example uses multiple CTEs to map the customers with their respective sales representatives:

```
WITH salesrep AS (SELECT employeeNumber, CONCAT(firstName, ' ', lastName) AS salesrepName
FROM employees
WHERE jobTitle = 'Sales Rep'),
```

```
customer_salesrep AS (SELECT customerName, salesrepName
FROM customers
INNER JOIN salesrep ON employeeNumber = salesrepEmployeeNumber )
```

```
SELECT *
FROM customer_salesrep
ORDER BY customerName;
```

ANKKIT KUMAR GUPPTA

DATA ANALYST | PROMPT ENGINEER



WITHOUT DATA, YOU'RE JUST ANOTHER PERSON WITH AN OPINION

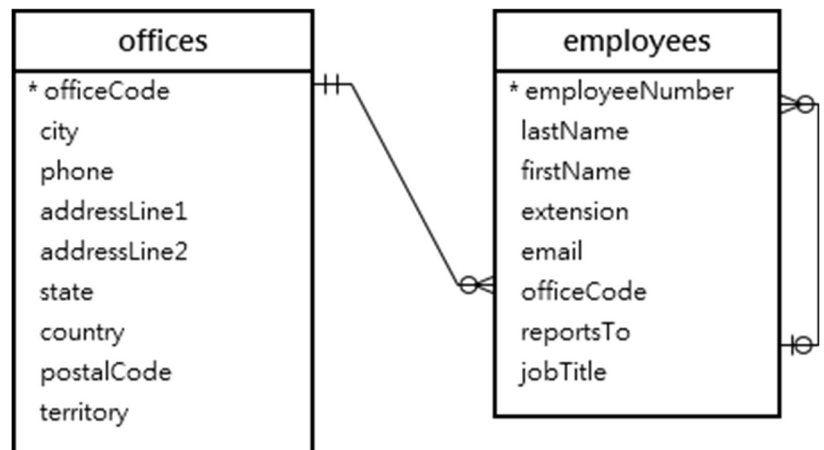
How it works:

- **CTE salesrep:** Select employeeNumber & concatenate the firstName & lastName columns to create a column named salesrepName, & include only employees with the job title 'Sales Rep'.
- **CTE customer_salesrep:** selects customerName & salesrepName by joining the customers table with the salesrep CTE based on the common column employeeNumber.
- **Main query:** Select all columns from the customer_salesrep CTE.

customerName	salesrepName
Alpha Cognac	Gerard Hernandez
American Souvenirs Inc	Foon Yue Tseng
Amica Models & Co.	Pamela Castillo
Anna's Decorations, Ltd	Andy Fixter
Atelier graphique	Gerard Hernandez
Australian Collectables, Ltd	Andy Fixter
Australian Collectors, Co.	Andy Fixter
Australian Gift Network, Co	Andy Fixter
...	

4) Joining two CTEs example

We'll use the offices and employee's tables



The following example is creating two CTEs and joining them to get the Sales Representatives located in the USA, including their office information:

```
WITH e AS (SELECT *
FROM employees
WHERE jobTitle = 'Sales Rep'), o AS (SELECT *
FROM offices
WHERE country = 'USA')
SELECT firstName,lastName,city,state,postalCode
FROM e INNER JOIN o USING (officeCode);
```


ANKKIT KUMAR GUPPTA

DATA ANALYST | PROMPT ENGINEER



WITHOUT DATA, YOU'RE JUST ANOTHER PERSON WITH AN OPINION

How it works:

- CTE e: Retrieve employees whose job title is Sales Rep.
- CTE o: Retrieve offices located in the USA.
- Main query: Joins the CTE e and o using the officeCode column.

```
+-----+-----+-----+-----+-----+
| firstName | lastName | city          | state | postalCode |
+-----+-----+-----+-----+-----+
| Leslie    | Jennings | San Francisco | CA    | 94080      |
| Leslie    | Thompson  | San Francisco | CA    | 94080      |
| Julie     | Firrelli  | Boston        | MA    | 02107      |
| Steve     | Patterson | Boston        | MA    | 02107      |
| Foon Yue  | Tseng     | NYC           | NY    | 10022      |
| George    | Vanauf    | NYC           | NY    | 10022      |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Summary

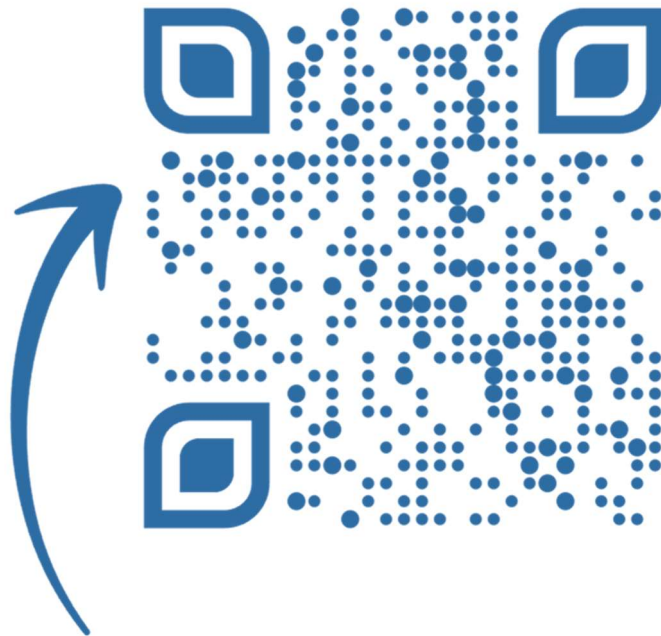
- Use MySQL CTEs to break down complex queries into simpler, more manageable queries. Each CTE represents a temporary result set that can be referenced within the main query.

ANKKIT KUMAR GUPPTA

DATA ANALYST | PROMPT ENGINEER



WITHOUT DATA, YOU'RE JUST ANOTHER PERSON WITH AN OPINION



If you would like to learn more & stay updated, please follow me on [LinkedIn](#)