

10 CTEs

CTEs Questions

Common Table Expressions (CTEs) in MySQL

Introduction to Common Table Expressions (CTEs)

What is a CTE?

- A Common Table Expression (CTE) is a temporary result set defined within the execution scope of a single **SELECT**, **INSERT**, **UPDATE**, or **DELETE** statement.
- CTEs make the query easier to read and maintain.

Types of CTEs

1. **Non-Recursive CTEs:** Used for simple queries that can be defined without recursion.
2. **Recursive CTEs:** Used for hierarchical data and self-referencing queries, where recursion is needed.

Benefits of Using CTEs

- Improved readability and maintenance of complex queries.
- Modularize query components.
- Simplify recursive queries.

Syntax

Basic CTE Syntax

```
WITH cte_name AS (  
    SELECT ...  
)  
SELECT * FROM cte_name;
```

SQL

Recursive CTE Syntax

```
WITH RECURSIVE cte_name AS (  
    SELECT ...  
    UNION ALL  
    SELECT ...  
)  
SELECT * FROM cte_name;
```

SQL

Examples Using BigBasket_Products Table

1. Non-Recursive CTE Example

Retrieve Top 5 Highest Rated Products

```
WITH TopRatedProducts AS (  
    SELECT product, rating  
    FROM BigBasket_Products  
    ORDER BY rating DESC  
    LIMIT 5  
)  
SELECT * FROM TopRatedProducts;
```

SQL

2. Recursive CTE Example

Hierarchical Categories and Sub-Categories

Assume we have a simplified hierarchical structure within the `category` and `sub_category` columns.

```
WITH RECURSIVE CategoryHierarchy AS (  
    SELECT category AS parent, sub_category AS child, 1 AS level  
    FROM BigBasket_Products  
    WHERE category = 'Fruits & Vegetables' -- Starting point  
  
    UNION ALL  
  
    SELECT bb.category AS parent, bb.sub_category AS child, ch.level + 1  
    FROM BigBasket_Products bb  
    INNER JOIN CategoryHierarchy ch ON bb.category = ch.child  
)  
SELECT * FROM CategoryHierarchy;
```

SQL

3. Using CTEs for Aggregation

Calculate Total Sales by Category

```
WITH SalesByCategory AS (  
    SELECT category, SUM(sale_price) AS total_sales  
    FROM BigBasket_Products  
    GROUP BY category  
)  
SELECT * FROM SalesByCategory;
```

SQL

4. Combining Multiple CTEs

Calculate Discount Percentage and List Products with a High Discount

SQL

```
WITH PriceDetails AS (
    SELECT product, sale_price, market_price,
           ((market_price - sale_price) / market_price) * 100 AS
discount_percentage
    FROM bb_products
), HighDiscountProducts AS (
    SELECT product, discount_percentage
    FROM PriceDetails
    WHERE discount_percentage > 20 -- Threshold for high discount
)
SELECT * FROM HighDiscountProducts;
```

```
mysql> WITH PriceDetails AS (
→   SELECT product, sale_price, market_price,
→   ((market_price - sale_price) / market_price) * 100 AS discount_percentage
→   FROM bb_products
→ ), HighDiscountProducts AS (
→   SELECT product, discount_percentage
→   FROM PriceDetails
→   WHERE discount_percentage > 20 -- Threshold for high discount
→ )
→ SELECT * FROM HighDiscountProducts;
```

product	discount_percentage
Brass Angle Deep - Plain, No.2	52.4000
Ceramic Barrel Brush - Colour May Vary	25.0000
Insulated Hot Fresh Casserole For Roti/Chapati - White	26.3687
Pet Solitaire Container Set - Silver	48.5036
Bodylicious Deodorant Spray - Mate (For Men)	29.7436
Sport Deo Spray - Fresh, for Men	44.8780
Storage/Lunch Steel Container with PP Lid - Red	63.5452
Dog Supplement - Absolute Skin + Coat Tablet	29.9197
Peanut Butter - Creamy, Super	40.1146
Aqua Halo Rejuvenating Conditioner	24.8889
Battery Power Kids Toothbrush - Barbie	25.0501
Sesame Seed Oil	30.0000
Disposable Bamboo Wood Skewer/Picker Sticks- For Barbeque, 12 Inch	54.8495
Toothbrush Superhero Assorted, Batman/Spiderman	37.9747
Stainless Steel Storage Lunch Container - Flat, Carry Snack Pack	48.1928
Oceans Deodorant for Men - Long Lasting & Fresh Aquatic Fragrance	40.1338
Frankincense Oil	24.9275
2 Fold Umbrella - Auto-Open & Manual Close, Dark Violet	61.4152
Sandwich Toaster - Non-Stick, Gas	20.9524

5. Using CTEs for Complex Joins

Example with Joining and Filtering

SQL

```
WITH FilteredProducts AS (  
    SELECT product, category, brand, sale_price, rating  
    FROM BigBasket_Products  
    WHERE sale_price < 100 AND rating > 4.0 -- Filtering criteria  
) , BrandAveragePrice AS (  
    SELECT brand, AVG(sale_price) AS avg_price  
    FROM FilteredProducts  
    GROUP BY brand  
)  
SELECT fp.product, fp.category, fp.brand, fp.sale_price, fp.rating, bap.avg_price  
FROM FilteredProducts fp  
INNER JOIN BrandAveragePrice bap ON fp.brand = bap.brand;
```

Summary

- CTEs provide a powerful way to simplify and modularize SQL queries.
- Non-recursive CTEs are used for simple query definitions.
- Recursive CTEs are useful for hierarchical and self-referencing data.
- CTEs enhance query readability, maintainability, and can be used for complex query scenarios.