Using Subqueries and CTEs

Mihaela Danci
Data Analyst

linkedin.com/in/mihaela-danci/

Overview

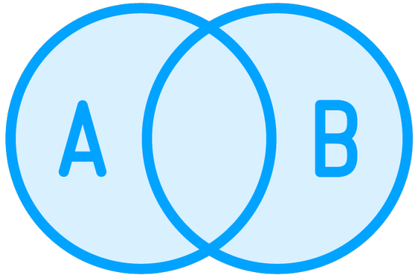
Subqueries

Common table expressions

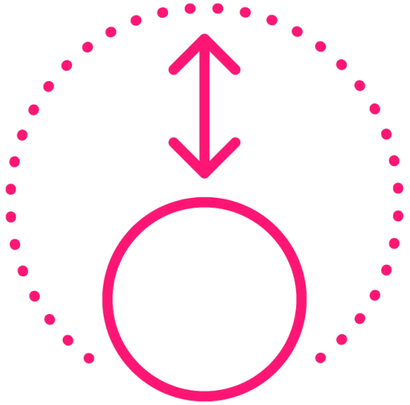
Pivoting and unpivoting data

Subqueries

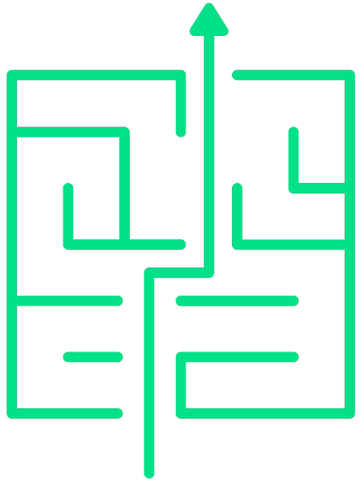
Subqueries



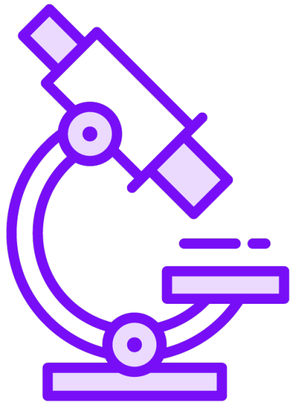
Combine data



Manipulate data



Simplify complex queries



Combine summarized with detailed data

Subqueries

```
SELECT CustomerKey, OrderKey, OrderValue
FROM Orders
WHERE CustomerKey IN
   (SELECT c.CustomerKey
    FROM Customer c
    INNER JOIN Georgraphy g
    ON c.CustomerKey = g.CustomerKey
    WHERE c.Country = ‘United States’);
```

Subqueries

```
SELECT Customerkey, OrderKey, OrderValue
FROM Orders
WHERE CustomerKey IN
   (SELECT c.CustomerKey
    FROM Customer c
    INNER JOIN Georgraphy g
    ON c.CustomerKey = g.CustomerKey
    WHERE c.Country = ‘United States’);
```

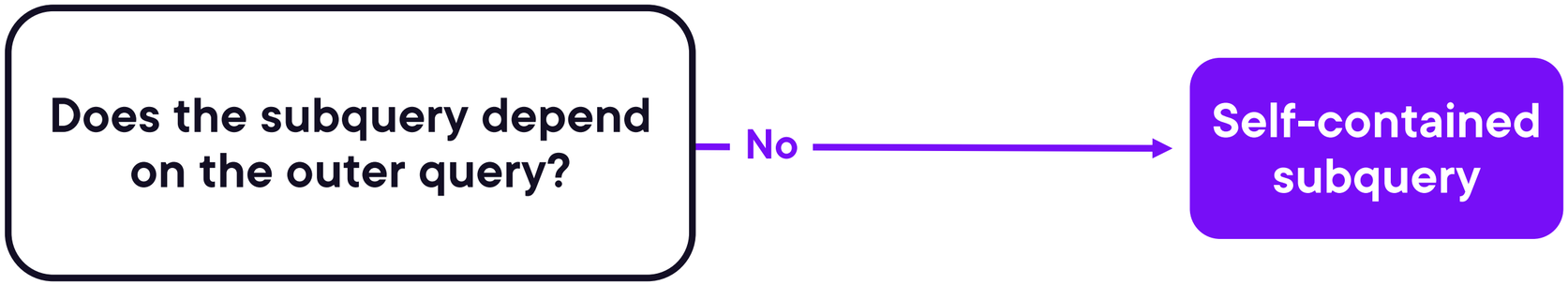
Outer query

Subqueries

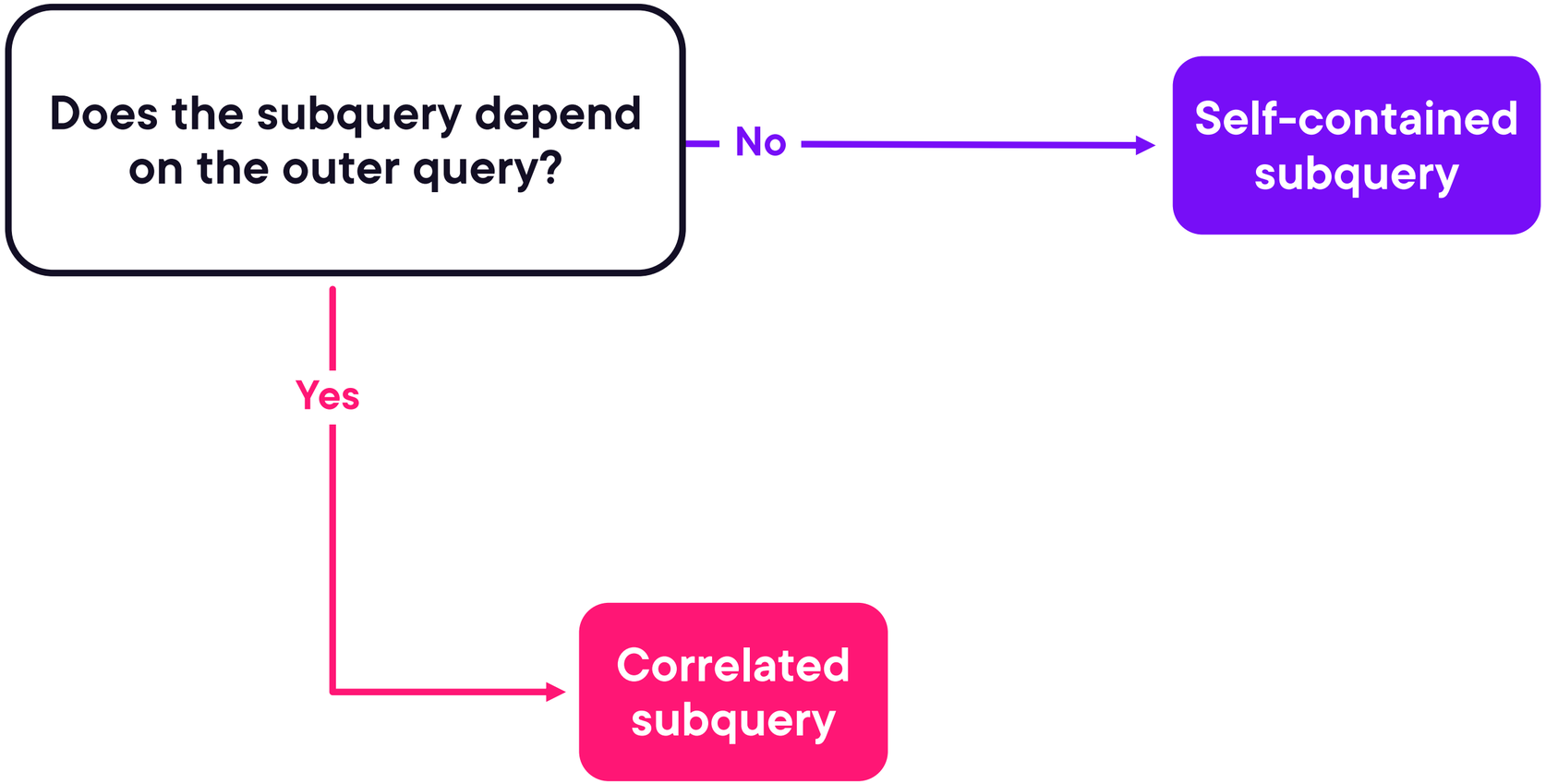
```
SELECT CustomerKey, OrderKey, OrderValue
FROM Orders
WHERE CustomerKey IN
   (SELECT c.CustomerKey
    FROM Customer c
    INNER JOIN Georgraphy g
    ON c.CustomerKey = g.CustomerKey
    WHERE c.Country = ‘United States’);
```

Subquery

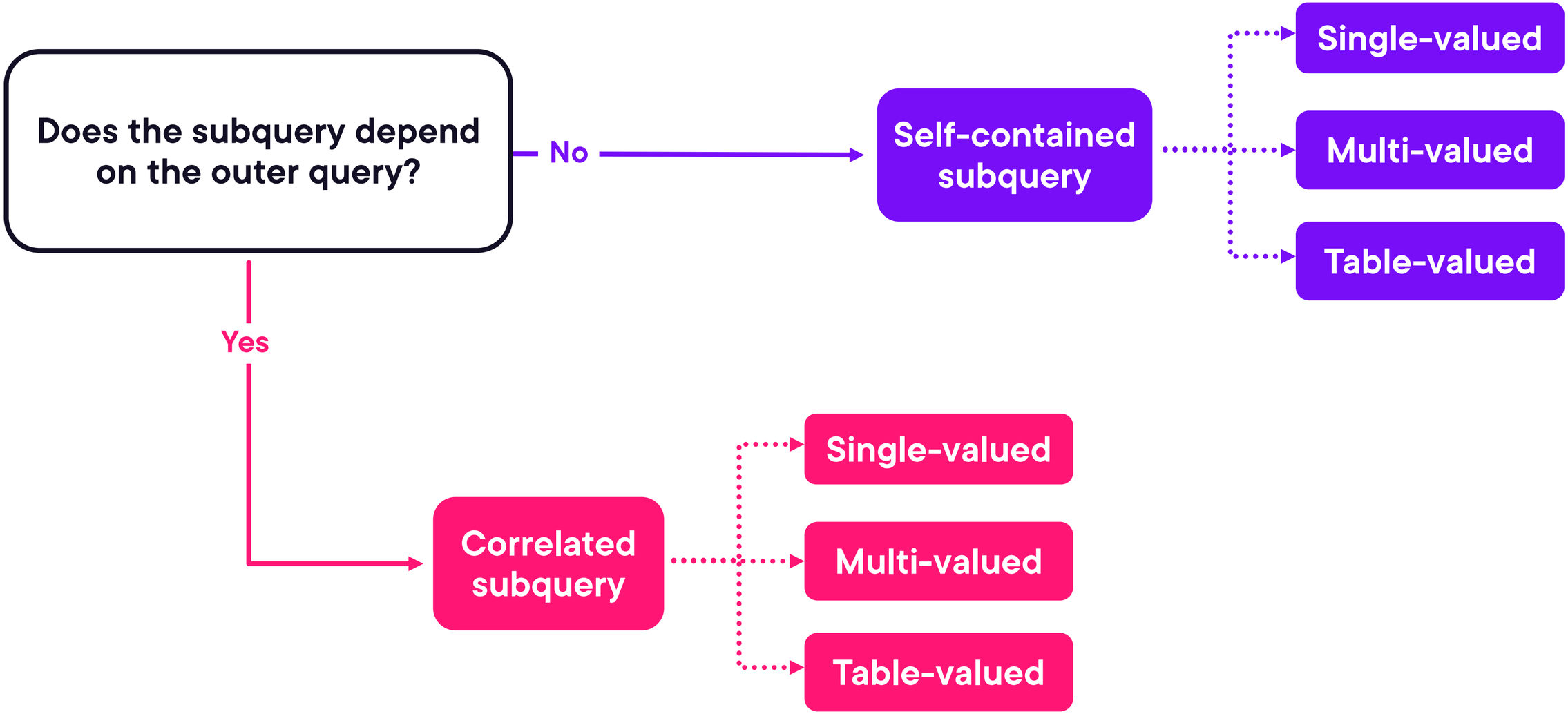
Subquery Types



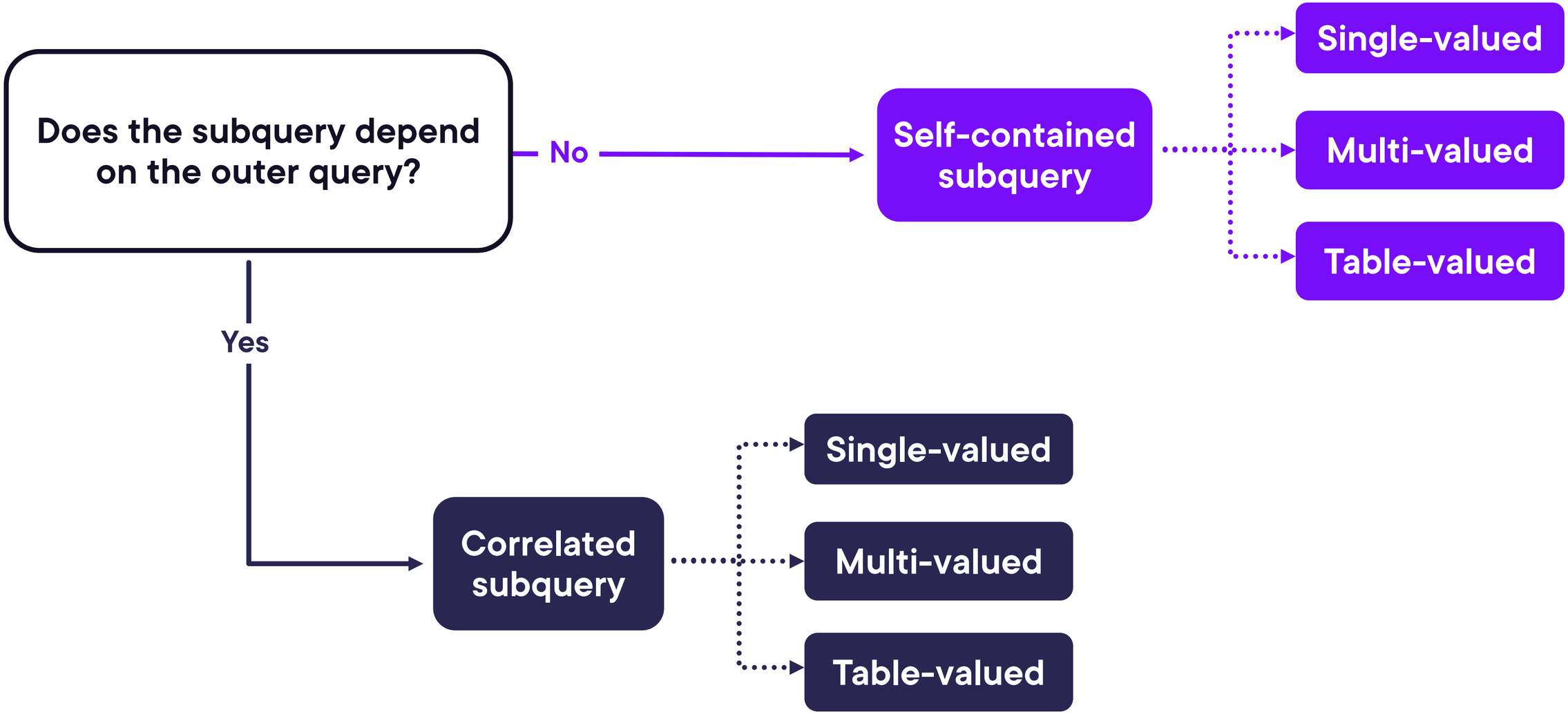
Subquery Types



Subquery Types

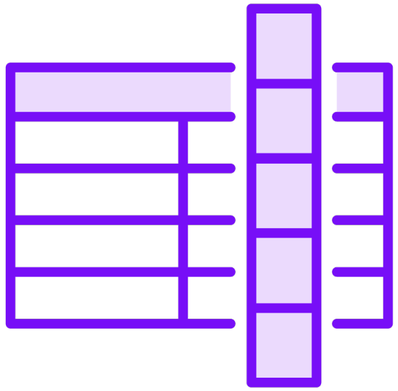


Subquery Types



Placement

Single value



Multivalued

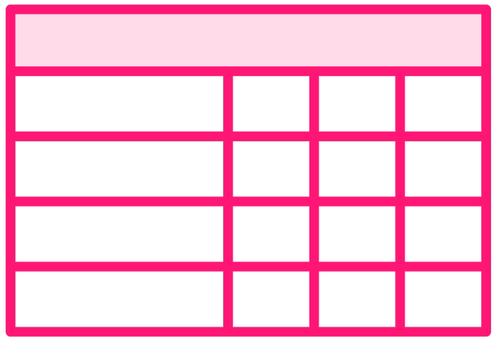


Table-valued

Demo

Subqueries

- Identify orders with quantities exceeding the average order value
- Compute running totals

Common Table Expressions

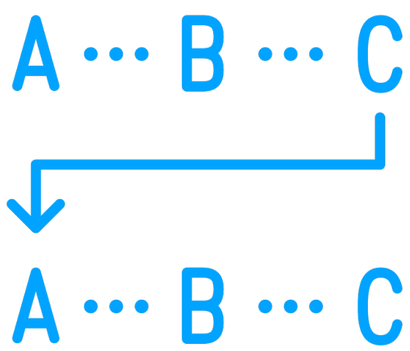
```sql
-- Subquery
SELECT CustomerKey, OrderKey, OrderValue
FROM Orders
WHERE CustomerKey IN
   (SELECT c.CustomerKey
    FROM Customer c
    INNER JOIN Georgraphy g
    ON c.CustomerKey = g.CustomerKey
    WHERE c.Country = ‘United States’);
```

```sql
-- CTE
WITH CTE1 AS
(
    SELECT c.CustomerKey
    FROM Customer c
    INNER JOIN Georgraphy g
    ON c.CustomerKey = g.CustomerKey
    WHERE c.Country = ‘United States’
)
```

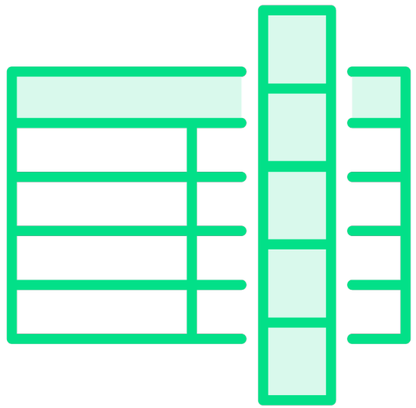
```sql
-- Subquery
SELECT CustomerKey, OrderKey, OrderValue
FROM Orders
WHERE CustomerKey IN
   (SELECT c.CustomerKey
    FROM Customer c
    INNER JOIN Georgraphy g
    ON c.CustomerKey = g.CustomerKey
    WHERE c.Country = ‘United States’);
```

```sql
-- CTE
WITH CTE1 AS
(
    SELECT c.CustomerKey
    FROM Customer c
    INNER JOIN Georgraphy g
    ON c.CustomerKey = g.CustomerKey
    WHERE c.Country = ‘United States’
)
SELECT CustomerKey, OrderKey, OrderValue
FROM Orders
WHERE CustomerKey IN CTE1;
```

Valid Common Table Expressions



Sorting results is not possible



Every column must have a name

Common Table Expressions

Nesting CTEs is not possible

```
WITH CTE1 AS(),
     CTE2 AS(),
     CTE3 AS()
```

Common Table Expressions

Nesting CTEs is not possible

Cross-reference each other

Common table expressions enhance the readability of complex queries.

Common Table Expressions

```
WITH CTE1 AS
(
     SELECT c.CustomerKey
     FROM Customer c
        INNER JOIN Georgraphy g
        ON c.CustomerKey = g.CustomerKey
     WHERE c.Country = ‘United States’
)

SELECT CustomerKey, OrderKey, OrderValue
FROM Orders
WHERE CustomerKey IN CTE1;
```

Combining Data

Joins

Subqueries

CTEs

It depends!

Technique Selection

Goal

Known method

Obtain the correct result

Demo

Each order stored information about

- Customers
- Employees

Common table expressions

Percentage of sales for each customer in relation to the total sales for each employee

- The total sales per employee
- The total sales for each combination of customer and employee
- Divide these 2 numbers

Pivoting and Unpivoting Data

Pivoting

Pivoting is a way to summarize data in a more organized manner, making it easier to compare data.

Transforming Data

Transforming Data

Calculate the quantity sold by year and region

Region	Year	Quantity
East	2022	100
East	2023	250
West	2022	280
West	2023	340
North	2022	180
North	2023	200

Pivoting Data

Year	East	West	North
2022	100	280	180
2023	250	340	200

PIVOT()

Pivoting Data

Year	East	West	North
2022	100	280	180
2023	250	340	200

Pivoting Data

Year	East	West	North
2022	100	280	180
2023	250	340	200

Syntax

```
SELECT …
FROM <input_table>
    PIVOT (<agg_function> (<aggregation_element>)
            FOR <spreading_element> IN (<list_of_target_columns>)) AS <result_table>
```

Syntax

```
SELECT …
FROM <input_table>
    PIVOT (<agg_function> (<aggregation_element>)
            FOR <spreading_element> IN (<list_of_target_columns>)) AS <result_table>
```

Syntax

```
SELECT …
FROM <input_table>
     PIVOT (<agg_function> (<aggregation_element>)
             FOR <spreading_element> IN (<list_of_target_columns>)) AS <result_table>
```

Syntax

```
SELECT …
FROM <input_table>
    PIVOT (<agg_function> (<aggregation_element>)
            FOR <spreading_element> IN (<list_of_target_columns>)) AS <result_table>
```

Syntax

```
SELECT …
FROM <input_table>
     PIVOT (<agg_function> (<aggregation_element>)
             FOR <spreading_element> IN (<list_of_target_columns>)) AS <result_table>
```

Syntax

```
SELECT …
FROM <input_table>
     PIVOT (<agg_function> (<aggregation_element>)
             FOR <spreading_element> IN (<list_of_target_columns>)) AS <result_table>
```

Syntax

```
SELECT …
FROM <input_table>
     PIVOT (<agg_function> (<aggregation_element>)
             FOR <spreading_element> IN (<list_of_target_columns>)) AS <result_table>
```

Syntax

```
SELECT …
FROM <input_table>
     PIVOT (<agg_function> (<aggregation_element>)
             FOR <spreading_element> IN (<list_of_target_columns>)) AS <result_table>
```

Unpivoting Data

Year	East	West	North
2022	100	280	180
2023	250	340	200

Region	Year	Quantity
East	2022	100
East	2023	250
West	2022	280
West	2023	340
North	2022	180
North	2023	200

Transforming Data

Pivoting data

PIVOT()

Unpivoting data

UNPIVOT()

Unpivoting Data

```
SELECT …
FROM <input_table>
     UNPIVOT (<values_column>
              FOR <column_names> IN (<source_columns>)) AS <result_table>
```

Unpivoting Data

```
SELECT …
FROM <input_table>
     UNPIVOT (<values_column>
              FOR <column_names> IN (<source_columns>)) AS <result_table>
```

Unpivoting Data

```
SELECT …
FROM <input_table>
     UNPIVOT (<values_column>
              FOR <column_names> IN (<source_columns>)) AS <result_table>
```

Unpivoting Data

```
SELECT …
FROM <input_table>
     UNPIVOT (<values_column>
              FOR <column_names> IN (<source_columns>)) AS <result_table>
```

Demo

PIVOT()

UNPIVOT()

Summary

Subqueries

Common table expressions

Pivoting and unpivoting data

Up Next:

Working with Window Functions