The major difference between a scalar subquery and a single-row subquery lies in their expected results and usage within SQL queries:

Scalar Subquery

* **Definition**: A scalar subquery is a subquery that returns a single value (one column and one row).
* **Usage**: It is often used in the SELECT clause, WHERE clause, or HAVING clause to compare or assign a single value.
* **Example**: Find the average rating of a specific product.

SELECT ProductName,   
       (SELECT AVG(Rating) FROM Reviews WHERE ProductID = p.ProductID) AS AverageRating  
FROM Products p;

Single-Row Subquery

* **Definition**: A single-row subquery is a subquery that is expected to return a single row, but it can contain multiple columns.
* **Usage**: It is typically used in the WHERE clause to compare multiple columns in a single row against another row.
* **Example**: Find the product with the highest price.

SELECT \*  
FROM Products  
WHERE Price = (SELECT MAX(Price) FROM Products);

Key Differences

1. **Return Value**:
   * **Scalar Subquery**: Returns a single value (one column and one row).
   * **Single-Row Subquery**: Returns a single row but can have multiple columns.
2. **Placement**:
   * **Scalar Subquery**: Can be used in places where a single value is expected, such as in the SELECT list, WHERE clause, or as part of an expression.
   * **Single-Row Subquery**: Typically used in the WHERE clause to compare against a single row but may involve multiple columns.
3. **Example of Scalar Subquery**:

SELECT ProductName,   
       (SELECT AVG(Rating) FROM Reviews WHERE ProductID = p.ProductID) AS AverageRating  
FROM Products p;

1. **Example of Single-Row Subquery**:

SELECT \*  
FROM Products  
WHERE (ProductID, Price) = (SELECT ProductID, MAX(Price) FROM Products);

Practical Use Cases

* **Scalar Subquery**: Use when you need a single value to compare or compute within a query.
* **Single-Row Subquery**: Use when you need to compare or select rows based on multiple columns from a single row of a subquery.

Understanding these differences helps in writing more efficient and accurate SQL queries based on the required data extraction or comparison tasks.

questions

SELECT \* FROM Products WHERE Price = (SELECT MAX(Price) FROM Products);

but here the sub query also return only value that is max then why we cannot say this is scaler

================================

You are correct that in the given example, the subquery returns a single value, which can indeed be considered a scalar subquery. The distinction between a scalar subquery and a single-row subquery is sometimes subtle and context-dependent. Here’s a more refined explanation:

Scalar Subquery

* **Definition**: A subquery that returns a single value (one column and one row).
* **Usage**: Can be used in the SELECT clause, WHERE clause, or other places where a single value is needed.
* **Example**:

SELECT ProductName,   
       (SELECT MAX(Price) FROM Products) AS MaxPrice  
FROM Products;

Single-Row Subquery

* **Definition**: A subquery that is expected to return a single row, which may contain one or more columns.
* **Usage**: Typically used in the WHERE clause to compare against a single row of values.
* **Example**:

SELECT \*  
FROM Products  
WHERE (ProductID, Price) = (SELECT ProductID, MAX(Price) FROM Products);

Your Example Revisited

Your query:

SELECT \*  
FROM Products  
WHERE Price = (SELECT MAX(Price) FROM Products);

This query indeed uses a scalar subquery because the subquery (SELECT MAX(Price) FROM Products) returns a single value (the maximum price). Therefore, it fits the definition of a scalar subquery.

Why This Example Can Be Considered Scalar:

* The subquery returns a single value, which is the maximum price.
* It is used in the WHERE clause to compare against the Price column.

Clarification on Terminology

* **Scalar Subquery**: Returns a single value and is used in a context where a single value is expected.
* **Single-Row Subquery**: Expected to return a single row, which can contain one or more columns. If it returns a single column, it behaves like a scalar subquery.

Conclusion

In your example, the subquery indeed behaves as a scalar subquery because it returns a single value. The term "single-row subquery" is more general and can refer to subqueries that return a single row but potentially multiple columns. Scalar subqueries are a specific type of single-row subquery that return exactly one value.