Inner Join vs. Outer Join: A Brief Note

Both Inner and Outer Joins are fundamental operations in SQL used to combine rows from two or more tables based on a related column. However, they differ significantly in how they handle unmatched rows.

Inner Join:

- **Purpose:** Retrieves rows from both tables only when the join condition is met.[1] It returns only the matching rows between the tables.[2]
- **Result:** The result set contains only the rows where the values in the joined columns satisfy the specified condition.[3] Rows with no match in the other table are excluded.[4]
- **Pros:**
  - Provides a clear and concise result set containing only related data.[5]
  - Often performs efficiently as it only deals with matching records.[6]
- **Cons:**
  - May lead to loss of information if you need to see all records from one or both tables, regardless of a match.
- **Use Case:** When you need to retrieve information that exists in both tables based on a common key (e.g., getting customer names and their corresponding order details).
- **Example:**
  SQL
  ```
  SELECT Customers.CustomerID, Customers.CustomerName, Orders.OrderID
  FROM Customers
  INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
  ```
  This query[7] will only return rows where a CustomerID exists in both the Customers and Orders tables. Customers without orders and orders without a corresponding customer will not be included.

Outer Join:

- **Purpose:** Retrieves all rows from one table (left, right, or both) and the matching rows from the other table. If there is no match, NULL values are substituted for the columns of the non-matching table.[8]
- **Types:**
  - **LEFT OUTER JOIN (or LEFT JOIN):** Returns all rows from the left table and the matching rows from the right table. If no match in the right table, NULLs are used for the right table's columns.
  - **RIGHT OUTER JOIN (or RIGHT JOIN):** Returns all rows from the right table and the matching rows from the left table. If no match in the left table, NULLs

are used for the left table's columns.
    - **FULL OUTER JOIN (or FULL JOIN):** Returns all rows from both tables.[9] If there is no match in either table, NULLs are used for the columns of the non-matching table.
- **Pros:**
    - Ensures that all records from the specified "outer" table are included in the result, even if there are no matching records in the other table.
    - Useful for identifying records that have no corresponding entries in another related table.
- **Cons:**
    - The result set can contain NULL values, which might require additional handling in applications or further queries.[10]
    - FULL OUTER JOIN might not be supported by all database systems.
- **Use Case:**
    - **LEFT JOIN:** To get all customers and their orders (if any exist). Customers without orders will still be listed with NULL order details.[11]
    - **RIGHT JOIN:** To get all orders and the corresponding customer details (if available). Orders without a customer might be listed with NULL customer details.
    - **FULL JOIN:** To get a complete list of all customers and all orders, showing matches where they exist and NULLs where they don't.
- **Example (LEFT JOIN):**

```
SQL
SELECT Customers.CustomerID, Customers.CustomerName, Orders.OrderID
FROM Customers
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

  This query[12] will return all customers, and for those who have placed orders, their OrderID will be displayed. Customers with no orders will have NULL in the OrderID column.

In summary, **Inner Join** focuses on the intersection of related data, while **Outer Joins** prioritize including all data from at least one of the tables, highlighting both matching and non-matching records. The choice between them depends on the specific information requirements of the query.

Sources

1. https://www.projectpro.io/article/amazon-data-engineer-interview-guide/965
2. https://sqlskillz.com/blog/mastering-sql-how-to-join-multiple-tables-effectively
3. https://jyoondev.tistory.com/91
4. https://medium.com/feed/@AhlemMustapha