Name: CBoisen
Date: November 14,2023
Course: IT FDN 130 A

Assignment 5- Joins

Introduction

When using relational databases, data is often organized into a number of individual tables connected through unique identities. While there are many different ways to view data that is stored in a single table, simultaneously viewing data present in multiple tables often requires the use of a function called JOIN.

Joins

Joins serve to combine data from two or more tables and are useful when extracting information from tables with shared column identifiers. There are a number types of joins that retrieve and combine data from multiple tables with different results including INNER JOIN, OUTER JOIN (LEFT and RIGHT), CROSS JOIN and SELF JOIN. The type of join used depends entirely on what data needs to be retrieved and where it is located in the database.

Inner

Inner joins are the most common type of join. By using a shared key (such as identity), inner joins are used to combine two tables and return the rows from both that have matching values. This specific operation requires (1) the names of the tables to be combined, (2) the name of the fields that are joined (the shared key) and (3) a potential relational comparison operator. When added to the ON clause, the comparison operator can create a conditional join statement for the rows returned (instead of joining all the rows that match THEN filtering the results).

Outer

Outer join is a category of join that contains three different subtypes: FULL, LEFT, and RIGHT. Similar to an inner join, outer joins use a shared key to extract and return a specific set of data from multiple tables. Unlike inner joins, outer joins return matched and unmatched rows from both tables. Using a full outer join will return all rows from both the tables specific in the function with unmatched rows displaying a NULL value (use of the WHERE clause can specify whether to display only the matching or not matching outputs as well). The output of left outer joins includes all matching records from the "left" specified table (table 1) and only matching records from the "right" table (table 2). In contrast, the output of right outer joins includes all matching records from the "right" specified table (table 2) and only matching records from the "left" table (table 1). If there are no matching records in the join, zero records will be returned from the second table.

Cross

Cross joins, also known as cartesian joins, are used to create paired combinations of rows from two tables and unlike the previous joins, does not require a joining condition. In this join, each row from the designated first table is combined with each row from a designated second table returning what is known as a Cartesian product set of joined rows. This query generates and returns all possible paired combinations of rows from the two tables.

Self

While most joins link two or more tables together, the self joins function links a table to itself. To match with itself, the table must contain a column with a primary key and column that can be matched up with the values in the primary key column. The rows in both columns do not have to match by row and can be null.

	EmployeeID	EmployeeFirstName	EmployeeLastName	ManagerID
1	1	Nancy	Davolio	2
2	2	Andrew	Fuller	2
3	3	Janet	Leverling	2
4	4	Margaret	Peacock	2
5	5	Steven	Buchanan	2
6	6	Michael	Suyama	5
7	7	Robert	King	5
8	8	Laura	Callahan	2
9	9	Anne	Dodsworth	5

Figure 1. A self join can be utilized within the table by matching values from the EmployeeID row with values from the ManagerID row to identify which employee works under which manager.

An example of matching column values within a table can be seen in Figure 1; EmployeeID can be matched with ManagerID to identify and display the name of the manager that each employee works under.

Conclusion

Joins are used in database management and data processing to extract and display data from two or more tables. There are many different types of joins that can be used with SQL to retrieve and combine data in different ways. Some of the most common types of joins include inner, outer (full, left and right), cross, and self. By using functions to match and return data stored in several tables, tabular access within a database can be modified in ways that most suit the needs and desires of the database operator.