**Week 6**

**Saritha Pothugunty**

**Database System Design and Management**

**Professor: Dr. Esteban Deleon**

**Date: 17-11-2024**

**Three Types of Outer Joins**

Left outer joins, right outer joins, and full outer joins are the three primary kinds of outer joins. Depending on the orientation of the join, these joins can include rows that do not match in the results.

**Left Outer Join:**

Gets the whole set of records from the first table as well as the set of records that match from the second table. In the event that no match is found, the right table columns will be filled with NULL values.

**Example:**

Anyone can see all customers, even those who haven't ordered, when a "Customers" table is left-joined with a "Orders" table.

A screenshot of a computer program

Description automatically generated

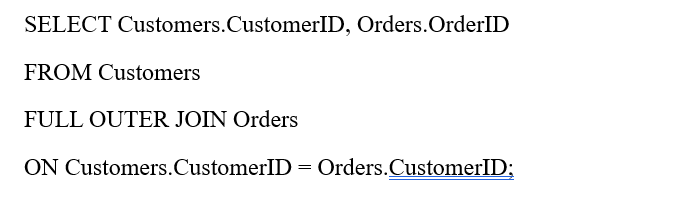
**Right Outer Join:**

It works on the same lines as above but gets all records from the right table and matching rows from the left. , NULL values complete for the unmatched rows of the left table.

**Full Outer Join:**

More complex than the previous two presenting all rows of both tables and filling in missing values of unmatched records with NULL. In case there are unmatched rows on the two sides, one row, or column will have NULLs for its corresponding fields.

**SQL Example:**



**Cross Join**

In Cross Join, all rows of one table will be matched with all the rows of another table, therefore also known as Cartesian Product. It does not impose joining condition criteria and may produce a number of rows if tables are extensive in their size.

Example: Cross join if a “Products” table includes 3 rows and a “Suppliers” table includes 2 rows results in 3\*2=6 rows.

A close-up of a product

Description automatically generatedCartesian product joins are employed where two data sets have no relation to one other or merely to create combinations as per the possibility of array pair or generation of list of items from two arrays.

**Recursive Join**

Recursive Join is employed on hierarchical or self-referential DSTs in which a table contains an attribute that points to the table’s own name e.g. employment routes within an organization or family lines. Data relationships can also be queried and traversed using recursive joins to look for parent/child relationship or ancestor/descendant relationship.

**Example:**

Suppose that there is an "Employees" table which contains multiple records, consisting of "EmployeeID" and "ManagerID” column. A recursive join can discover all the employees in an organization that reports to a particular manager.

A screenshot of a computer

Description automatically generated

Recursive joins are very useful in SQL for traversal and processing of a hierarchical database.

**Reference**

GeeksforGeeks. (2024h, November 4). *SQL Joins (Inner, Left, Right and Full Join)*. GeeksforGeeks. <https://www.geeksforgeeks.org/sql-join-set-1-inner-left-right-and-full-joins/>