Module 09 – Joins

Module Objectives

Write SQL statements to retrieve data from multiple tables

List the different types and forms of joins in SQL

Write SQL statements to retrieve data from multiple tables

When data from more than one table in the database is required, a join condition is used. Rows in one table can be joined with rows in another table according to common values existing in corresponding columns.

To display data from two or more tables, write a simple join condition in the WHERE clause.

When writing a SELECT statement that joins tables, precede the column name with the table name for clarity and to enhance database access.

If the same column name appears in more than one table, the column must be pre-fixed with the table name.

To joins n tables together, you need a minimum of n-1 joins conditions. Therefore, to join four tables, a minimum of three join conditions are required.

Cartesian Product

All rows in the first table are joined with all rows from the second table.

This is the results of a join condition being omitted or being invalid.

List the different types and forms of joins in SQL

Using WHERE to Join

Equijoin

Also called a simple join or an inner join.

```
SELECT p.productname, c.categoryname

FROM victor.products p, victor.categories c

WHERE p.categoryid = c.categoryid;
```

Self join

A join that combines a table with itself. This is accomplished by using the same table two times and creating a different alias for each of the two tables.

```
SELECT e.firstname, e.lastname,  ' reports to ', b.firstname, b.lastname

FROM victor.employees e, victor.employees b

WHERE e.reportsto = b.employeeid;
```

Outer join (right / left)

Use a left or right outer join when one table is deficient in information to match up with values in the other table.

In Oracle, you place the (+) on the table that is deficient in information

```
SELECT e.firstname, e.lastname,  ' reports to ', b.firstname, b.lastname

FROM victor.employees e, victor.employees b

WHERE e.reportsto = b.employeeid(+);
```

Using FROM to Join

Equijoin

Also called a simple join or an inner join.

```
SELECT p.productname, c.categoryname

FROM victor.products p JOIN victor.categories c

ON  p.categoryid = c.categoryid;
```

<u>In Oracle and MySQL</u>, if the column names are the same in the ON clause, you may use a USING clause to create the same results by only using the column name once.

```
SELECT productname, categoryname

FROM victor.products JOIN victor.categories

USING (categoryid);
```

Self join

A join that combines a table with itself. This is accomplished by using the same table two times and creating a different alias for each of the two tables.

```
SELECT e.firstname, e.lastname,  ' reports to ', b.firstname, b.lastname

FROM victor.employees e JOIN victor.employees b

ON e.reportsto = b.employeeid;
```

Left or Right Outer join

Use a left or right outer join when one table is deficient in information to match up with values in the other table.

Left

```
SELECT e.firstname, e.lastname,  ' reports to ', b.firstname, b.lastname

FROM victor.employees e LEFT OUTER JOIN victor.employees b

ON e.reportsto = b.employeeid;
```

Right

```
SELECT e.firstname, e.lastname,  ' reports to ', b.firstname, b.lastname

FROM victor.employees b RIGHT OUTER JOIN victor.employees e

ON b.employeeid = e.reportsto;
```

If you need to join three or more tables, you will need to include an additional JOIN ON clause for each additional table in the SELECT statement.

```
SELECT categoryname, companyname, productname

FROM victor.products  p

JOIN victor.categories c

ON p.categoryid = c.categoryid

JOIN victor.suppliers s

ON p.supplierid = s.supplierid;
```