**Introduction to Joins**

**Please be aware that MS Access ‘Date’ outputs are formatted slightly differently ‘01/02/2020’ compared to Oracle ’01-FEB-20’**

A join is used when a SQL query requires data from more than one table on the database. Rows in one table may be joined to rows in another table according to common values existing in corresponding columns.

There are two main types of join condition:

* + Equi-join
  + Non-equi-join

**Equi-Join**

In order to work out manually, which department any employee is in, we would compare the value in the employee's DEPTNO column with the same DEPTNO values in the DEPT table. The relationship between the EMP and DEPT table is an *equi-join*, in that values in the DEPTNO column on both tables are equal.

A join condition is specified in the WHERE clause:

SELECT *column (s)*

FROM *tables*

WHERE *join condition is...*

To join the tables EMP and DEPT, enter:

SELECT ENAME, JOB, DNAME

FROM EMP, DEPT

WHERE EMP. DEPTNO = DEPT.DEPTNO;

**ENAME JOB DNAME**

**----- ------ -----------**

**CLARK MANAGER ACCOUNTING**

**KING PRESIDENT ACCOUNTING**

**MILLER CLERK ACCOUNTING**

**JONES MANAGER RESEARCH**

**FORD ANALYST RESEARCH**

**ADAMS CLERK RESEARCH**

**SMITH CLERK RESEARCH**

**SCOTT ANALYST RESEARCH**

**WARD SALESMAN SALES**

**TURNER SALESMAN SALES**

**ALLEN SALESMAN SALES**

**JAMES CLERK SALES**

**BLAKE MANAGER SALES**

**MARTIN SALESMAN SALES**

**Equi-Join cont….**

Notice that every employee now has his respective department name displayed. Note also that the join condition specifies table names prior to the column name. This is a requirement when column names are the same in both tables (this requirement is also essential when referring to those columns which may be ambiguous in a SELECT or ORDER BY clause.

Again, to distinguish between the DEPTNO column in EMP and the one in DEPT, for example, enter:

SELECT ENAME, DEPT.DEPTNO, DNAME

FROM EMP, DEPT

WHERE EMP. DEPTNO = DEPT. DEPTNO;

**Using Table Aliases**

*Temporary labels* (or aliases) can be used in the FROM clause. These temporary names are valid only for the current statement.

Table aliases should also be specified in the SELECT clause, as this more specific form speeds up the query:

SELECT E. ENAME, D. DEPTNO, D. DNAME

FROM EMP E, DEPT D

WHERE E. DEPTNO = D. DEPTNO

ORDER BY D. DEPTNO;

**MS Access**

**Inner Joins is also an equi-join**

This join is used to retrieve rows from two or more tables by matching a field value that is common between the tables. The fields you join on must have similar data types, and you cannot join on MEMO or OLEOBJECT data types.

To build an **INNER JOIN** statement, use the **INNER JOIN** keywords in the FROM clause of a SELECT statement.

This example uses the **INNER JOIN** to build a result set of all employees, their departments and department name, sorted by department number.

SELECT E.ENAME, D.DEPTNO, D.DAME

FROM EMP INNER JOIN DEPT

ON EMP.DEPTNO=DEPT.DEPTNO

ORDER BY D.DEPTNO;

Be aware that the table names are divided by the **INNER JOIN** keywords and that the relational comparison is after the **ON** keyword. For the relational comparisons, you can also use the <, >, <=, >=, or <> operators, and you can also use the **BETWEEN** keyword. Also note that the fields (DeptNo) from both tables are used only in the relational comparison; they are not part of the final result set.

**Outer Joins**

If a row does not satisfy a join condition, then the row will not appear in the query result. In fact, in the equi-join condition of EMP and DEPT, department 40 does not appear. This is because there are no employees in department 40!

Such rows can be returned if an *outer join* operator is used in the join condition.

The operator is a plus sign enclosed in brackets (+), and **is placed on the side of the join (table) which is deficient in information.**

The operator has the effect of creating one or more NULL rows, to which one or more rows from the non-deficient table can be joined. One NULL row is created for every additional row in the non-deficient table.

SELECT E. ENAME, D. DEPTNO, D. DNAME

FROM EMP E, DEPT D

WHERE E. DEPTNO **(+)** = D. DEPTNO

AND D. DEPTNO IN (30, 40);

**ENAME DEPTNO DNAME**

**------ ------ ------------**

**ALLEN 30 SALES**

**WARD 30 SALES**

**MARTIN 30 SALES**

**BLAKE 30 SALES**

**TURNER 30 SALES**

**JAMES 30 SALES**

**40 OPERATIONS**

## MS Access OUTER JOINs

An OUTER join is used to retrieve records from multiple tables while preserving records from one of the tables, even if there is no matching record in the other table. MS Access has two types of **OUTER JOINs the LEFT OUTER JOIN** and the **RIGHT OUTER JOIN**.

Think of two tables that are beside each other, a table on the left and a table on the right. The **LEFT OUTER JOIN** selects all rows in the right table that match the relational comparison criteria. It also selects all rows from the left table, even if no match exists in the right table. The **RIGHT OUTER JOIN** is simply the reverse of the **LEFT OUTER JOIN**; all rows in the right table are preserved instead.

**OUTER JOINs** can be nested inside **INNER JOINs** in a multi-table join, but **INNER JOINs** cannot be nested inside **OUTER JOINs**.

**Products / Cartesian products**

When a join condition is invalid or omitted completely, the result is a PRODUCT, and all possible combinations of rows in all tables will be displayed. An example would be

SELECT \*

FROM EMP, DEPT;

In the absence of a WHERE condition, each employee record would be selected and matched/linked to each department record.

14 employees x 4 department records = 56 rows outputted!

**EMPNO ENAME JOB MGR HIREDATE DEPTNO**

7900 JAMES CLERK 7698 23-JUL-04 30

**DEPTNO DNAME LOC**

**10 ACCOUNTING NEW YORK**

**20 RESEARCH DALLAS**

**30 SALES CHICAGO**

**40 OPERATIONS BOSTON**

Not a good thing, especially with tables that contain hundreds or thousands of rows.

Always qualify your joins.

**Non-Equi-Join**

The relationship between the EMP and SALGRADE tables is a *non-equi-join*, in that no column in EMP is exactly the same as a column in SALGRADE. The relationship is obtained *using an operator other than equal (=)*. To evaluate an employee's grade, her salary must be *between* any one of the low and high salary ranges.

The BETWEEN operator is used to construct the condition, enter:

SELECT E. ENAME, E. SAL, S. GRADE

FROM EMP E, SALGRADE S

WHERE E.SAL BETWEEN S. LOSAL AND S.HISAL;

**ENAME SAL GRADE**

**------ ---- ------**

**SMITH 800 1**

**ADAMS 1100 1**

**JAMES 950 1**

**WARD 1250 2**

**MARTIN 1250 2**

**MILLER 1300 2**

**ALLEN 1600 3**

**TURNER 1500 3**

**JONES 2975 4**

**BLAKE 2850 4**

**CLARK 2450 4**

**SCOTT 3000 4**

**FORD 3000 4**

**KING 5000 5**

Other operators such as <= and >= can be used, but BETWEEN is the simplest. Remember, the low value first, high value last when using BETWEEN. Again, table aliases have been specified for performance reasons.

**Rules for Joining Tables**

In order to join all three tables, it is necessary to construct two join conditions. To join four tables, a minimum of three join conditions would be needed. Thus:

**The number of tables minus one = minimum number of join conditions**

This rule may not apply if your table has a concatenated primary key that uniquely identifies each row (see later for details of primary keys).

**Review of Syntax (including *Joins)***

**SELECT** [ DISTINCT ] { [table]. \* /expr [alias],...}

**FROM** table [alias],...

**WHERE** [ join condition ] ...

**AND** [ row condition ] ...

**OR** [ another row condition ] ...

**GROUP BY** { expr / column }

**HAVING** { group condition }

**ORDER BY** { expr / column }[ASC / DESC]

**Introduction to Joins Worksheet**

**Please be aware that MS Access ‘Date’ outputs are formatted slightly differently ‘01/02/2020’ compared to Oracle ’01-FEB-20’**

1. Display all employee names and their department name, in department name order.

**ENAME DNAME**

**----- ----------**

**MILLER ACCOUNTING**

**KING ACCOUNTING**

**CLARK ACCOUNTING**

**ADAMS RESEARCH**

**FORD RESEARCH**

**JONES RESEARCH**

**SMITH RESEARCH**

**SCOTT RESEARCH**

**WARD SALES**

**TURNER SALES**

**ALLEN SALES**

**JAMES SALES**

**BLAKE SALES**

**MARTIN SALES**

**14 records selected.**

2. Display all employee names, department number and name.

**ENAME DNAME DEPTNO**

**----- ---------- ------**

**CLARK ACCOUNTING 10**

**KING ACCOUNTING 10**

**MILLER ACCOUNTING 10**

**JONES RESEARCH 20**

**FORD RESEARCH 20**

**ADAMS RESEARCH 20**

**SMITH RESEARCH 20**

**SCOTT RESEARCH 20**

**WARD SALES 30**

**TURNER SALES 30**

**ALLEN SALES 30**

**JAMES SALES 30**

**BLAKE SALES 30**

**MARTIN SALES 30**

3. Display the department that has no employees. Same question as last week where you used a subquery. This week use an Outer Join (+).

**DEPTNO DNAME**

**----------- ----------**

**40 OPERATIONS**

4. Display the name, location and department of employees whose salary is more than 1500 a month.

**ENAME LOCATION DNAME**

**----- --------- -------**

**CLARK NEW YORK ACCOUNTING**

**KING NEW YORK ACCOUNTING**

**FORD DALLAS RESEARCH**

**SCOTT DALLAS RESEARCH**

**JONES DALLAS RESEARCH**

**BLAKE CHICAGO SALES**

**ALLEN CHICAGO SALES**

**7 records selected.**

5. Produce a list showing employee's salary grades.

**ENAME JOB SAL GRADE**

**------ -------- ---- -----**

**SMITH CLERK 800 1**

**JAMES CLERK 950 1**

**ADAMS CLERK 1100 1**

**WARD SALESMAN 1250 2**

**MARTIN SALESMAN 1250 2**

**MILLER CLERK 1300 2**

**TURNER SALESMAN 1500 3**

**ALLEN SALESMAN 1600 3**

**CLARK MANAGER 2450 4**

**BLAKE MANAGER 2850 4**

**JONES MANAGER 2975 4**

**SCOTT ANALYST 3000 4**

**FORD ANALYST 3000 4**

**KING PRESIDENT 5000 5**

6. Show only employees on grade 3.

**ENAME JOB SAL GRADE**

**------ -------- ---- -----**

**ALLEN SALESMAN 1600 3**

**TURNER SALESMAN 1500 3**

7. Show all employees in Dallas.

**ENAME SAL LOCATION**

**----- ---- --------**

**JONES 2975 DALLAS**

**FORD 3000 DALLAS**

**ADAMS 1100 DALLAS**

**SMITH 800 DALLAS**

**SCOTT 3000 DALLAS**

8. List the employee name, job, salary, grade and department name for everyone in the company except clerks. Sort on descending salary.

**ENAME JOB SAL GRADE DNAME**

**----- --- ---- ----- ----------------**

**KING PRESIDENT 5000 5 ACCOUNTING**

**FORD ANALYST 3000 4 RESEARCH**

**SCOTT ANALYST 3000 4 RESEARCH**

**JONES MANAGER 2975 4 RESEARCH**

**BLAKE MANAGER 2850 4 SALES**

**CLARK MANAGER 2450 4 ACCOUNTING**

**ALLEN SALESMAN 1600 3 SALES**

**TURNER SALESMAN 1500 3 SALES**

**MARTIN SALESMAN 1250 2 SALES**

**WARD SALESMAN 1250 2 SALES**

**10 records selected.**

9. (Harder). List the following details for employees who earn 36000 a year or who are clerks.

**ENAME JOB ANNUAL\_SAL DEPTNO DNAME GRADE**

**----- ---- ---------- ------ ---------- -----**

**FORD ANALYST 36000 20 RESEARCH 4**

**SCOTT ANALYST 36000 20 RESEARCH 4**

**JAMES CLERK 11400 30 SALES 1**

**ADAMS CLERK 13200 20 RESEARCH 1**

**SMITH CLERK 9600 20 RESEARCH 1**

**MILLER CLERK 15600 10 ACCOUNTING 2**

**6 records selected.**