JOINS

We have covered Natural JOINS and Cartesian products – but there are a couple of other types of JOINS we should cover as well.

Natural JOIN

A Natural JOIN creates a new table, combining the common fields. Look at the difference of the two below:

```
mysql> SELECT * FROM Students S, Enrolled E
    -> WHERE S.SID = E.SID;
+-----+------------------+------------+-----+---------+-------+---------+
| SID | SName            | NumCredits | SID | ClassID | Grade | Credits |
+-----+------------------+------------+-----+---------+-------+---------+
|   1 | Elmer Fudd       |         48 |   1 | CS314   | NULL  |    NULL |
|   1 | Elmer Fudd       |         48 |   1 | CS575   | NULL  |       4 |
|   2 | Roger Ramjet     |         20 |   2 | CS430   | NULL  |       4 |
|   2 | Roger Ramjet     |         20 |   2 | CS575   | NULL  |       4 |
|   3 | Alice Wonderland |         72 |   3 | CS314   | NULL  |    NULL |
|   3 | Alice Wonderland |         72 |   3 | CT320   | NULL  |       4 |
+-----+------------------+------------+-----+---------+-------+---------+
6 rows in set (0.05 sec)
```

NATURAL JOINS are specified in the FROM clause

```
mysql> SELECT * FROM Students NATURAL JOIN Enrolled;
+-----+------------------+------------+---------+-------+---------+
| SID | SName            | NumCredits | ClassID | Grade | Credits |
+-----+------------------+------------+---------+-------+---------+
|   1 | Elmer Fudd       |         48 | CS314   | NULL  |    NULL |
|   1 | Elmer Fudd       |         48 | CS575   | NULL  |       4 |
|   2 | Roger Ramjet     |         20 | CS430   | NULL  |       4 |
|   2 | Roger Ramjet     |         20 | CS575   | NULL  |       4 |
|   3 | Alice Wonderland |         72 | CS314   | NULL  |    NULL |
|   3 | Alice Wonderland |         72 | CT320   | NULL  |       4 |
+-----+------------------+------------+---------+-------+---------+
6 rows in set (0.00 sec)
```

Cartesian Product

Cartesian Product is done by not specifying any criteria.

```
mysql> SELECT * FROM Students, Enrolled;
+-----+------------------+------------+-----+---------+-------+---------+
| SID | SName            | NumCredits | SID | ClassID | Grade | Credits |
+-----+------------------+------------+-----+---------+-------+---------+
|   1 | Elmer Fudd       |         48 |   1 | CS314   | NULL  |    NULL |
|   2 | Roger Ramjet     |         20 |   1 | CS314   | NULL  |    NULL |
|   3 | Alice Wonderland |         72 |   1 | CS314   | NULL  |    NULL |
|   1 | Elmer Fudd       |         48 |   1 | CS575   | NULL  |       4 |
|   2 | Roger Ramjet     |         20 |   1 | CS575   | NULL  |       4 |
|   3 | Alice Wonderland |         72 |   1 | CS575   | NULL  |       4 |
|   1 | Elmer Fudd       |         48 |   2 | CS430   | NULL  |       4 |
|   2 | Roger Ramjet     |         20 |   2 | CS430   | NULL  |       4 |
|   3 | Alice Wonderland |         72 |   2 | CS430   | NULL  |       4 |
|   1 | Elmer Fudd       |         48 |   2 | CS575   | NULL  |       4 |
|   2 | Roger Ramjet     |         20 |   2 | CS575   | NULL  |       4 |
|   3 | Alice Wonderland |         72 |   2 | CS575   | NULL  |       4 |
|   1 | Elmer Fudd       |         48 |   3 | CS314   | NULL  |    NULL |
|   2 | Roger Ramjet     |         20 |   3 | CS314   | NULL  |    NULL |
|   3 | Alice Wonderland |         72 |   3 | CS314   | NULL  |    NULL |
|   1 | Elmer Fudd       |         48 |   3 | CT320   | NULL  |       4 |
|   2 | Roger Ramjet     |         20 |   3 | CT320   | NULL  |       4 |
|   3 | Alice Wonderland |         72 |   3 | CT320   | NULL  |       4 |
+-----+------------------+------------+-----+---------+-------+---------+
18 rows in set (0.02 sec)
```

INNER JOIN

A Cartesian Product is the basic form of an INNER JOIN. Predicate information can be added as well – but an INNER JOIN is the one of the most common mechanisms used today.

```
mysql> SELECT * from Students INNER JOIN Enrolled;
+-----+------------------+------------+-----+---------+-------+---------+
| SID | SName            | NumCredits | SID | ClassID | Grade | Credits |
+-----+------------------+------------+-----+---------+-------+---------+
|   1 | Elmer Fudd       |         48 |   1 | CS314   | NULL  |    NULL |
|   2 | Roger Ramjet     |         20 |   1 | CS314   | NULL  |    NULL |
|   3 | Alice Wonderland |         72 |   1 | CS314   | NULL  |    NULL |
|   1 | Elmer Fudd       |         48 |   1 | CS575   | NULL  |       4 |
|   2 | Roger Ramjet     |         20 |   1 | CS575   | NULL  |       4 |
|   3 | Alice Wonderland |         72 |   1 | CS575   | NULL  |       4 |
|   1 | Elmer Fudd       |         48 |   2 | CS430   | NULL  |       4 |
|   2 | Roger Ramjet     |         20 |   2 | CS430   | NULL  |       4 |
|   3 | Alice Wonderland |         72 |   2 | CS430   | NULL  |       4 |
|   1 | Elmer Fudd       |         48 |   2 | CS575   | NULL  |       4 |
|   2 | Roger Ramjet     |         20 |   2 | CS575   | NULL  |       4 |
|   3 | Alice Wonderland |         72 |   2 | CS575   | NULL  |       4 |
|   1 | Elmer Fudd       |         48 |   3 | CS314   | NULL  |    NULL |
|   2 | Roger Ramjet     |         20 |   3 | CS314   | NULL  |    NULL |
|   3 | Alice Wonderland |         72 |   3 | CS314   | NULL  |    NULL |
|   1 | Elmer Fudd       |         48 |   3 | CT320   | NULL  |       4 |
|   2 | Roger Ramjet     |         20 |   3 | CT320   | NULL  |       4 |
|   3 | Alice Wonderland |         72 |   3 | CT320   | NULL  |       4 |
+-----+------------------+------------+-----+---------+-------+---------+
18 rows in set (0.00 sec)

mysql> SELECT * from Students INNER JOIN Enrolled ON
    -> Students.SID = Enrolled.SID;
+-----+------------------+------------+-----+---------+-------+---------+
| SID | SName            | NumCredits | SID | ClassID | Grade | Credits |
+-----+------------------+------------+-----+---------+-------+---------+
|   1 | Elmer Fudd       |         48 |   1 | CS314   | NULL  |    NULL |
|   1 | Elmer Fudd       |         48 |   1 | CS575   | NULL  |       4 |
|   2 | Roger Ramjet     |         20 |   2 | CS430   | NULL  |       4 |
|   2 | Roger Ramjet     |         20 |   2 | CS575   | NULL  |       4 |
|   3 | Alice Wonderland |         72 |   3 | CS314   | NULL  |    NULL |
|   3 | Alice Wonderland |         72 |   3 | CT320   | NULL  |       4 |
+-----+------------------+------------+-----+---------+-------+---------+
6 rows in set (0.00 sec)
```

OUTER JOIN

An OUTER JOIN does not require each record in the two joined tables to have a matching record. The joined table retains each record – even if no other matching record exists.

There are three variations of an OUTER JOIN – a LEFT OUTER JOIN, a RIGHT OUTER JOIN, and a FULL OUTER JOIN. A LEFT OUTER JOIN adds all the unmatched records from the first relation, a RIGHT OUTER JOIN adds all the unmatched records from the second relation, and a FULL OUTER JOIN adds them both.

```
mysql> SELECT * FROM Employee;
+-----+------+
| Eid | Did  |
+-----+------+
|   1 |    1 |
|   2 |    2 |
|   3 | NULL |
+-----+------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM DEPARTMENT;
+-----+-------------+
| Did | DName       |
+-----+-------------+
|   1 | Marketing   |
|   2 | Engineering |
|   3 | Management  |
+-----+-------------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM Employee E
    ->   LEFT OUTER JOIN Department D on  E.Did = D.Did;
+-----+------+------+-------------+
| Eid | Did  | Did  | DName       |
+-----+------+------+-------------+
|   1 |    1 |    1 | Marketing   |
|   2 |    2 |    2 | Engineering |
|   3 | NULL | NULL | NULL        |
+-----+------+------+-------------+
3 rows in set (0.03 sec)

mysql> SELECT * FROM Employee E
    ->   RIGHT OUTER JOIN Department D on  E.Did = D.Did;
+------+------+-----+-------------+
| Eid  | Did  | Did | DName       |
+------+------+-----+-------------+
|    1 |    1 |   1 | Marketing   |
|    2 |    2 |   2 | Engineering |
| NULL | NULL |   3 | Management  |
+------+------+-----+-------------+
3 rows in set (0.00 sec)
```

Unfortunately, MySQL does not support a FULL OUTER JOIN – however you can simulate one by doing the following:

```
mysql> SELECT * FROM Employee E
    -> LEFT OUTER JOIN Department D on  E.Did = D.Did
    -> UNION
    -> SELECT * FROM Employee E
    -> RIGHT OUTER JOIN Department D on  E.Did = D.Did;

+------+------+------+-------------+
| Eid  | Did  | Did  | DName       |
+------+------+------+-------------+
|    1 |    1 |    1 | Marketing   |
|    2 |    2 |    2 | Engineering |
|    3 | NULL | NULL | NULL        |
| NULL | NULL |    3 | Management  |
+------+------+------+-------------+
4 rows in set (0.00 sec
```