**Related Tables**

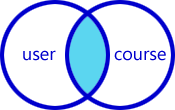
A very simple example would be users (students) and course enrollments:

**‘user’ table: ‘course’ table:**

|  |  |  |
| --- | --- | --- |
| **id** | **name** | **course** |
| 1 | Alice | 1 |
| 2 | Bob | 1 |
| 3 | Caroline | 2 |
| 4 | David | 5 |
| 5 | Emma | (NULL) |

|  |  |
| --- | --- |
| **id** | **name** |
| 1 | HTML5 |
| 2 | CSS3 |
| 3 | JavaScript |
| 4 | PHP |
| 5 | MySQL |

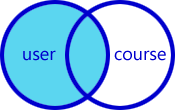
**INNER JOIN (or just JOIN)**

The most frequently used clause is INNER JOIN. This produces a set of records which match in both the user and course tables, i.e. all users who are enrolled on a course:

**Result:**

|  |  |
| --- | --- |
| **user.name** | **course.name** |
|  |  |
|  |  |
|  |  |
|  |  |

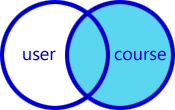
**LEFT JOIN**

What if we require a list of all students and their courses even if they’re not enrolled on one? A LEFT JOIN produces a set of records which matches every entry in the left table (user) regardless of any matching entry in the right table (course):

**Result:**

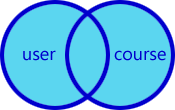
|  |  |
| --- | --- |
| **user.name** | **course.name** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**RIGHT JOIN**

Perhaps we require a list all courses and students even if no one has been enrolled? A RIGHT JOIN produces a set of records which matches every entry in the right table (course) regardless of any matching entry in the left table (user): **Result:**

|  |  |
| --- | --- |
| **user.name** | **course.name** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**OUTER JOIN (or FULL OUTER JOIN)**

Our last option is the OUTER JOIN which returns all records in both tables regardless of any match. Where no match exists, the missing side will contain NULL.

|  |  |
| --- | --- |
| **user.name** | **course.name** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |