My SQL
Joining Tables

Objectives

1. What is a join?
2. Inner Joins
3. Outer Joins
4. Natural Joins

1. What is a join?

Customers	
CustomerId	Name
1	Tom Willis
2	Terry Neils
3	Cindy Mason
4	Paul Novak
5	Jack Fonda

Reservations			
Id	CustomerId	Day	Status
1	4	2009-11-22	1
2	2	2009-11-28	1
3	2	2009-11-29	1
4	4	2009-11-29	1
5	5	2009-12-02	1
6	2	2009-12-03	2
7	3	2009-12-04	2

=> Want to know Cindy reserved for when?

1. What is a join?

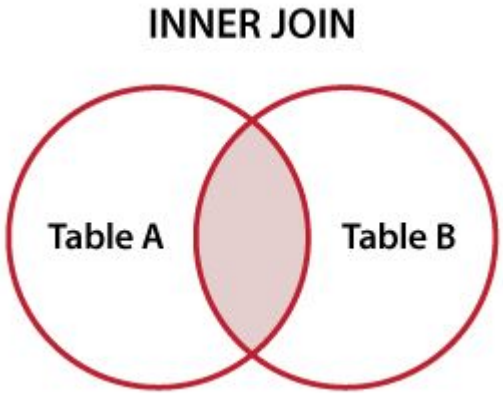
To combine rows from two or more tables, with some relationship between them, we using `JOIN` clause.

Here are the different types of the `JOIN`s in SQL:

- `(INNER) JOIN`
- `LEFT (OUTER) JOIN`
- `RIGHT (OUTER) JOIN`
- `FULL (OUTER) JOIN`

2. Inner Joins

The inner join is the most common type of joins. It is the default join also. The inner join selects only those records from database tables that have matching values. We have three types of *INNER JOINS*: `INNER JOIN`, `NATURAL INNER JOIN`, and `CROSS INNER JOIN`. The `INNER` keyword can be omitted.



2. Inner Joins

The `inner join` select only those records from database tables that have matching values.

```
mysql> SELECT Name, Day     FROM Customers AS C     JOIN Reservations AS R     ON C.CustomerId=R.CustomerId     WHERE R.Status=1;
```

```
+------------+-------------+
| Customers                |
+------------+-------------+
| CustomerId | Name        |
+------------+-------------+
| 1          | Tom Willis  |
| 2          | Terry Neils |
| 3          | Cindy Mason |
| 4          | Paul Novak  |
| 5          | Jack Fonda  |
+------------+-------------+
```

```
+-------------------------------------+
| Reservations                        |
+----+------------+------------+------+
| Id | CustomerId | Day        | Status |
+----+------------+------------+------+
| 1  | 4          | 2009-11-22 | 1    |
| 2  | 2          | 2009-11-28 | 1    |
| 3  | 2          | 2009-11-29 | 1    |
| 4  | 4          | 2009-11-29 | 1    |
| 5  | 5          | 2009-12-02 | 1    |
| 6  | 2          | 2009-12-03 | 2    |
| 7  | 3          | 2009-12-04 | 2    |
+----+------------+------------+------+
```

```
---\
---/
```

```
+-------------+------------+
| Result                   |
+-------------+------------+
| Name        | Day        |
+-------------+------------+
| Paul Novak  | 2009-11-22 |
| Terry Neils | 2009-11-28 |
| Terry Neils | 2009-11-29 |
| Paul Novak  | 2009-11-29 |
| Jack Fonda  | 2009-12-02 |
+-------------+------------+
```

```
5 rows in set (0.00 sec)
```

In this `SELECT` statement, we have selected all customers that have made some reservations. Paul Novak and Terry Neils made two reservations. Jack Fonda has made one. Tom Willis is missing, he has not yet made any reservations. Note that we have omitted the `INNER` keyword.

2. Inner Joins

```
mysql> SELECT Name, Day  FROM Customers AS C  JOIN Reservations AS R ON C.CustomerId=R.CustomerId WHERE R.Status=1;
```

The statement is equivalent to the following one:

```
mysql> SELECT Name, Day FROM Customers, Reservations WHERE Customers.CustomerId=Reservations.CustomerId AND Reservations.Status=1;
+-------------+------------+
| Name        | Day        |
+-------------+------------+
| Paul Novak  | 2009-11-22 |
| Terry Neils | 2009-11-28 |
| Terry Neils | 2009-11-29 |
| Paul Novak  | 2009-11-29 |
| Jack Fonda  | 2009-12-02 |
+-------------+------------+
```

We get the same data.

2. Inner Joins

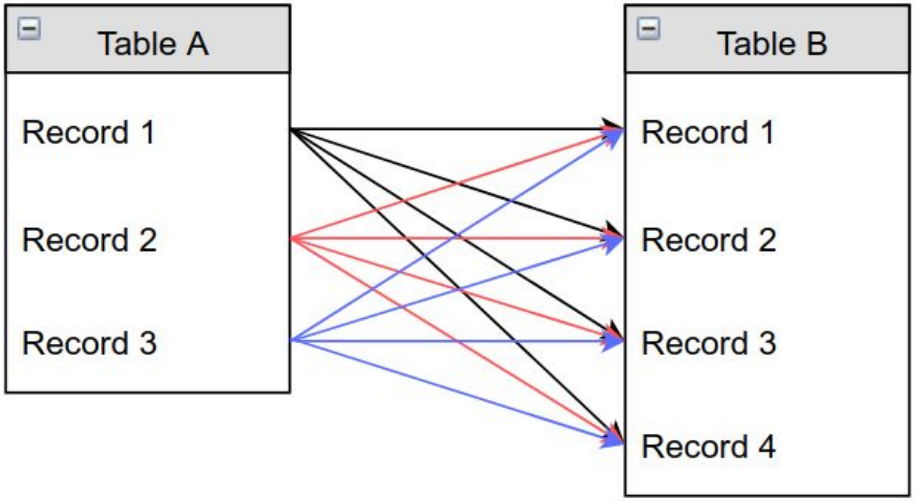
CROSS INNER JOIN

The `CROSS INNER JOIN` combines all records from one table with all records from another table. This type of join has little practical value. It is also called a Cartesian product of records.

```
mysql> SELECT Record FROM TableA CROSS JOIN TableB;
```

```
...
```

The same result can be achieved with the following SQL statement: `SELECT Record FROM TableA, TableB;`



=> 3x4 = 12 results !!!

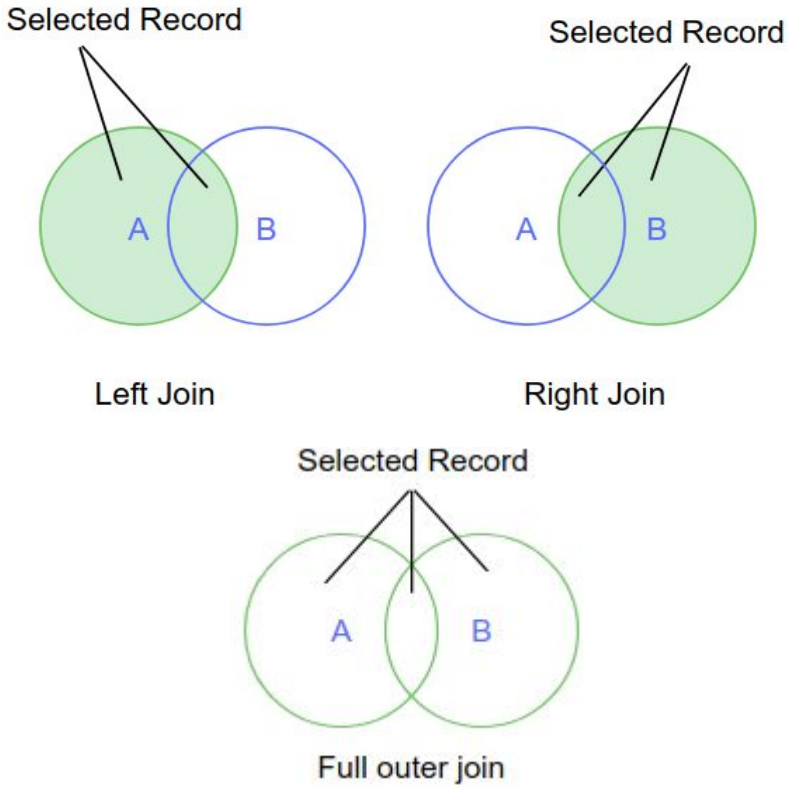
3. Outer Joins

An ***outer join*** does **not require** each record in the two joined tables to have **a matching record**. There are 3 types of outer joins:

- Left outer joins
- Right outer joins
- Full outer joins.

MySQL does not support full outer joins at the time of the tutorial creation.

As we have already stated above, the inner joins are the most common ones. Outer joins may be useful to find out orphaned records. Is a person a customer if he has not made any reservations? Is a reservation valid if we cannot match it with a customer?



3. Outer Joins

LEFT OUTER JOIN

The `LEFT OUTER JOIN` returns all values from the left table, even if there is no match with the right table. In such rows, there will be `NULL` values. In other words, left outer join returns all the values from the left table, plus matched values from the right table. Note that the `OUTER` keyword can be omitted.

```
SELECT Name, Day FROM Customers
LEFT JOIN Reservations ON Customers.CustomerId=Reservations.CustomerId;
```

Customers

CustomerId	Name
1	Tom Willis
2	Terry Neils
3	Cindy Mason
4	Paul Novak
5	Jack Fonda

Reservations

Id	CustomerId	Day	Status
1	4	2009-11-22	1
2	2	2009-11-28	1
3	2	2009-11-29	1
4	4	2009-11-29	1
5	5	2009-12-02	1
6	2	2009-12-03	2
7	3	2009-12-04	2

---\
---/

Result

Name	Day
Paul Novak	2009-11-22
Terry Neils	2009-11-28
Terry Neils	2009-11-29
Paul Novak	2009-11-29
Jack Fonda	2009-12-02
Terry Neils	2009-12-03
Cindy Mason	2009-12-04
Tom Willis	NULL

3. Outer Joins

We can use the `USING` keyword to achieve the same result. This is because the relationship column has the **same name** in both tables. The SQL statement will be less verbose.

```
SELECT Name, Day FROM Customers LEFT JOIN Reservations USING (CustomerId);
```

```
+-------------+------------+
| Name        | Day        |
+-------------+------------+
| Paul Novak  | 2009-11-22 |
| Terry Neils | 2009-11-28 |
| Terry Neils | 2009-11-29 |
| Paul Novak  | 2009-11-29 |
| Jack Fonda  | 2009-12-02 |
| Terry Neils | 2009-12-03 |
| Cindy Mason | 2009-12-04 |
| Tom Willis  | NULL       |
+-------------+------------+
```

Same result, with shorter SQL statement.

3. Outer Joins

RIGHT OUTER JOIN

`RIGHT OUTER JOIN` and `RIGHT JOIN` are the **same**. It gives all the records match in both tables and all possibilities of the right table. Orphaned right records show `NULL` on the left.

```
SELECT Name, Day FROM Customers RIGHT JOIN  Reservations USING (CustomerId);
```

```
+-------------+------------+
| Name        | Day        |
+-------------+------------+
| Paul Novak  | 2009-11-22 |
| Terry Neils | 2009-11-28 |
| Terry Neils | 2009-11-29 |
| Paul Novak  | 2009-11-29 |
| Jack Fonda  | 2009-12-02 |
| Terry Neils | 2009-12-03 |
| Cindy Mason | 2009-12-04 |
| Tom Willis  | NULL       |
+-------------+------------+
```

This is an output for the right join of two tables. All the records of the table on the right side (Reservations) have a matching record on the left side (Customers)

4. Natural Joins

A natural join links all columns in two tables with the **same name**. In our Customers and Reservations tables, we have a column named CustomerId.

R

A	B
1	2
4	5

S

B	C
2	3
6	7

R NATURAL JOIN S

A	B	C
1	2	3

4. Natural Joins

NATURAL INNER JOIN

The `NATURAL INNER JOIN` automatically uses all the **matching column names** for the join. In our tables, we have a column named *CustomerId* in both tables.

```
SELECT Name, Day FROM Customers NATURAL JOIN Reservations;
+-------------+------------+
| Name        | Day        |
+-------------+------------+
| Paul Novak  | 2009-11-22 |
| Terry Neils | 2009-11-28 |
| Terry Neils | 2009-11-29 |
| Paul Novak  | 2009-11-29 |
| Jack Fonda  | 2009-12-02 |
| Terry Neils | 2009-12-03 |
| Cindy Mason | 2009-12-04 |
+-------------+------------+
```

We get the same data. The SQL statement is less verbose.

4. Natural Joins

NATURAL LEFT OUTER JOIN

The `NATURAL LEFT OUTER JOIN` gives all the matching records from the tables and all other records on the left table. It automatically uses all the **matching column names** for the join.

```
SELECT Name, Day FROM Customers NATURAL LEFT JOIN Reservations;
+-------------+------------+
| Name        | Day        |
+-------------+------------+
| Paul Novak  | 2009-11-22 |
| Terry Neils | 2009-11-28 |
| Terry Neils | 2009-11-29 |
| Paul Novak  | 2009-11-29 |
| Jack Fonda  | 2009-12-02 |
| Terry Neils | 2009-12-03 |
| Cindy Mason | 2009-12-04 |
| Tom Willis  | NULL       |
+-------------+------------+
```

Same result, but with fewer key strokes.

4. Natural Joins

NATURAL RIGHT OUTER JOIN

The `NATURAL RIGHT OUTER JOIN` gives all the matching records from the tables and all other records on the right table. It automatically uses matching column names for the join.

```
SELECT Name, Day FROM Customers NATURAL RIGHT JOIN Reservations;
```

Name	Day
Terry Neils	2009-11-28
Terry Neils	2009-11-29
Terry Neils	2009-12-03
Cindy Mason	2009-12-04
Paul Novak	2009-11-22
Paul Novak	2009-11-29
Jack Fonda	2009-12-02