Using Union to simulate a full join

This is **optional** material. This is a fairly lengthy discussion of a technique we do not often use.

In unit 06, we were discussing joins and mentioned that MySQL does not support the full join. Most of the time, we use inner and outer joins.

These are the tables we used when discussing joins in unit 06. We have employees with no projects and a department with no employees and employees with no department.

z_em_dept

D_ID	D_Name
100	Manufacturing
150	Accounting
200	Marketing
250	Research

z_em_emp

E_ID	E_Name	D_ID
1	Jones	150
2	Martin	150
3	Gates	250
4	Anders	100
5	Bossy	
6	Perkins	

z_em_empproj

P_ID	E_ID
ORDB-10	3
ORDB-10	5
Q4-SALES	2
Q4-SALES	4
ORDB-10	2
Q4-SALES	5

A full join between the z_emp_dept and the z_em_emp table would return all departments and the matching employees and all employees and the matching departments.

This is the standard syntax for a full join. This does not work in MySQL

```
Select z_em_dept.d_id, d_name, e_id, e_name
From z_em_dept
full outer join z_em_emp on z_em_dept.d_id = z_em_emp.d_id;
```

And the result set

D_ID	D_Name	E_ID	E_Name
100	Manufacturing	4	Anders
150	Accounting	1	Jones
150	Accounting	2	Martin
200	Marketing	NULL	NULL
250	Research	3	Gates
NULL	NULL	5	Bossy
NULL	NULL	6	Perkins

Suppose we did two outer joins.

Demo 01: A left and a right join between dept and emp

```
Select z_em_dept.d_id, d_name, e_id, e_name
From z_em_dept
left join z_em_emp on z_em_dept.d_id = z_em_emp.d_id;
+------+---------------+------+--------+
| D_ID | D_Name        | E_ID | E_Name |
+------+---------------+------+--------+
|  100 | Manufacturing |    4 | Anders |
|  150 | Accounting    |    1 | Jones  |
|  150 | Accounting    |    2 | Martin |
|  200 | Marketing     | NULL | NULL   |
|  250 | Research      |    3 | Gates  |
+------+---------------+------+--------+
```

```
Select z_em_dept.d_id, d_name, e_id, e_name
From z_em_dept
right join z_em_emp on z_em_dept.d_id = z_em_emp.d_id;
+------+---------------+------+---------+
| D_ID | D_Name        | E_ID | E_Name  |
+------+---------------+------+---------+
|  150 | Accounting    |    1 | Jones   |
|  150 | Accounting    |    2 | Martin  |
|  250 | Research      |    3 | Gates   |
|  100 | Manufacturing |    4 | Anders  |
| NULL | NULL          |    5 | Bossy   |
| NULL | NULL          |    6 | Perkins |
+------+---------------+------+---------+
```

Demo 02: We can union these two queries. This is a union and it removes duplicates and gives us the same result set as the full join shown above.

```
Select z_em_dept.d_id, d_name, e_id, e_name
From z_em_dept
left join z_em_emp on z_em_dept.d_id = z_em_emp.d_id
union
Select z_em_dept.d_id, d_name, e_id, e_name
From z_em_dept
right join z_em_emp on z_em_dept.d_id = z_em_emp.d_id;
+------+---------------+------+---------+
| D_ID | D_Name        | E_ID | E_Name  |
+------+---------------+------+---------+
|  100 | Manufacturing |    4 | Anders  |
|  150 | Accounting    |    1 | Jones   |
|  150 | Accounting    |    2 | Martin  |
|  200 | Marketing     | NULL | NULL    |
|  250 | Research      |    3 | Gates   |
| NULL | NULL          |    5 | Bossy   |
| NULL | NULL          |    6 | Perkins |
+------+---------------+------+---------+
```

Demo 03: With a union All we get duplicates- the duplicated rows are those that would be returned by an inner join

```
Select z_em_dept.d_id, d_name, e_id, e_name
From z_em_dept
left join z_em_emp on z_em_dept.d_id = z_em_emp.d_id
union all
Select z_em_dept.d_id, d_name, e_id, e_name
From z_em_dept
right join z_em_emp on z_em_dept.d_id = z_em_emp.d_id;
+------+---------------+------+---------+
| D_ID | D_Name        | E_ID | E_Name  |
+------+---------------+------+---------+
|  100 | Manufacturing |    4 | Anders  |
|  150 | Accounting    |    1 | Jones   |
|  150 | Accounting    |    2 | Martin  |
|  200 | Marketing     | NULL | NULL    |
|  250 | Research      |    3 | Gates   |
|  150 | Accounting    |    1 | Jones   |
|  150 | Accounting    |    2 | Martin  |
```

```
|  250 | Research      |    3 | Gates   |
|  100 | Manufacturing |    4 | Anders  |
| NULL | NULL          |    5 | Bossy   |
| NULL | NULL          |    6 | Perkins |
+------+---------------+------+---------+
```

```
Select z_em_dept.d_id, d_name, e_id, e_name
From z_em_dept
join z_em_emp on z_em_dept.d_id = z_em_emp.d_id;
```

```
+------+---------------+------+--------+
| D_ID | D_Name        | E_ID | E_Name |
+------+---------------+------+--------+
|  150 | Accounting    |    1 | Jones  |
|  150 | Accounting    |    2 | Martin |
|  250 | Research      |    3 | Gates  |
|  100 | Manufacturing |    4 | Anders |
+------+---------------+------+--------+
```

Most of the time, our tables will have primary keys, but we occasionally have tables with no primary keys and those tables can have duplicate rows. The tables we are working with here have no primary keys. So we can add a duplicate row to the dept table and a duplicate row to the Emp table.

```
insert into z_em_dept values (100, 'Manufacturing');
insert into z_em_emp values ( 3, 'Gates', 250);
```

If we do a full outer join, then we should get both copies of the duplicate rows.

But with the union join in the earlier demo we do not get the two copies of those rows.

```
Select z_em_dept.d_id, d_name, e_id, e_name
From z_em_dept
left join z_em_emp on z_em_dept.d_id = z_em_emp.d_id
union
Select z_em_dept.d_id, d_name, e_id, e_name
From z_em_dept
right join z_em_emp on z_em_dept.d_id = z_em_emp.d_id;
```

```
+------+---------------+------+---------+
| D_ID | D_Name        | E_ID | E_Name  |
+------+---------------+------+---------+
|  100 | Manufacturing |    4 | Anders  |
|  150 | Accounting    |    1 | Jones   |
|  150 | Accounting    |    2 | Martin  |
|  200 | Marketing     | NULL | NULL    |
|  250 | Research      |    3 | Gates   |
| NULL | NULL          |    5 | Bossy   |
| NULL | NULL          |    6 | Perkins |
+------+---------------+------+---------+
```

The following adds a where clause to the second part of the union join to eliminate rows from the second part that are already in the first part.

Demo 04:

```
Select z_em_dept.d_id, d_name, e_id, e_name
From z_em_dept
left join z_em_emp on z_em_dept.d_id = z_em_emp.d_id
union all
```

```
Select z_em_dept.d_id, d_name, e_id, e_name
From z_em_dept
right join z_em_emp on z_em_dept.d_id = z_em_emp.d_id
Where z_em_dept.d_id is null;
+------+---------------+------+---------+
| D_ID | D_Name        | E_ID | E_Name  |
+------+---------------+------+---------+
|  100 | Manufacturing |    4 | Anders  |
|  150 | Accounting    |    1 | Jones   |
|  150 | Accounting    |    2 | Martin  |
|  200 | Marketing     | NULL | NULL    |
|  250 | Research      |    3 | Gates   |
|  250 | Research      |    3 | Gates   |
|  100 | Manufacturing |    4 | Anders  |
| NULL | NULL          |    5 | Bossy   |
| NULL | NULL          |    6 | Perkins |
+------+---------------+------+---------+
```