JOINS

Imagine you have a customers table and orders table. We can use joins to link the data together.

`Customers`:

id	name
1	John Smith
2	Jane Smith
3	Joe Bloggs

`Orders`:

id	order_date	amount	customer_id
1	01/01/2020	100	1
2	01/02/2020	200	3
3	01/03/2020	400	4
4	01/04/2020	500	5

INNER JOIN

This will get a list of customers who placed an order:

```
select `name`, order_date, amount
from customers c
inner join orders o
on c.id = o.customer_id
```

name	order_date	amount
John Smith	01/01/2020	100
Joe Bloggs	01/02/2020	200

LEFT JOIN

This left join will append information about orders on the customers table, regardless of whether a customer placed an order or not.

```
select `name`, order_date, amount
from customers c
left join orders o
on c.id = o.customer_id
```

name	order_date	amount
John Smith	01/01/2020	100
Jane Smith	NULL	NULL
Joe Bloggs	01/02/2020	200

RIGHT JOIN

This is a mirror of the left join on the previous slide which will get all orders, appended with customer information.

```
select `name`, order_date, amount
from customers c
right join orders o
on c.id = o.customer_id
```

name	order_date	amount
John Smith	01/01/2020	100
Joe Bloggs	01/02/2020	200
NULL	01/02/2020	400
NULL	01/02/2020	500

FULL OUTER JOIN

A list of all records from both tables. MySQL has no concept of a full outer join so we have to play a trick here.

```
select `name`, order_date, amount
from
  orders o left join customers c
  on o.customer_id = c.id

union all -- don't remove duplicates

select `name`, order_date, amount
from
  orders o right join customers c
  on o.customer_id = c.id
```