SQL

Queries with two or more tables

MySQL or MariaDB

Teacher: Alonso Rebolo

Academic Course 2019-2020

Cartesian Product

Shirt Table

Shirt-ID	Shirt	Weight-GR
1	White flax	210
2	Orange cotton	290
3	Black silk	260

Trouser Table

Trouser-ID	Trouser	Weight-GR
1	Navy fabric	470
2	Brown corduroy	730

Select *
from Shirt, Trouser;

Inadvisable

Shirt-ID	Shirt	Weight-GR	Trouser-ID	Trouser	Weight-GR
1	White flax	210	1	Navy fabric	470
1	White flax	210	2	Brown corduroy	730
2	Orange cotton	290	1	Navy fabric	470
2	Orange cotton	290	2	Brown corduroy	730
3	Black silk	260	1	Navy fabric	470
3	Black silk	260	2	Brown corduroy	730

Queries with two tables

To use two, or more, tables is necessary to join them. There are three possibilities:

1.- INNER JOIN

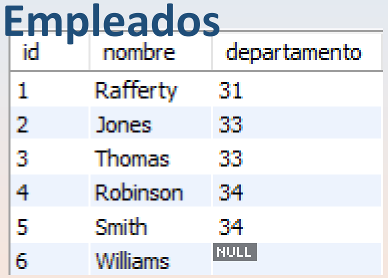
2.- LEFT OUTER JOIN

3.- RIGHT OUTER JOIN

Words: INNER and OUTER are not compulsories

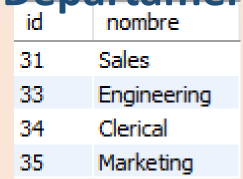
INNER JOIN

Empleados

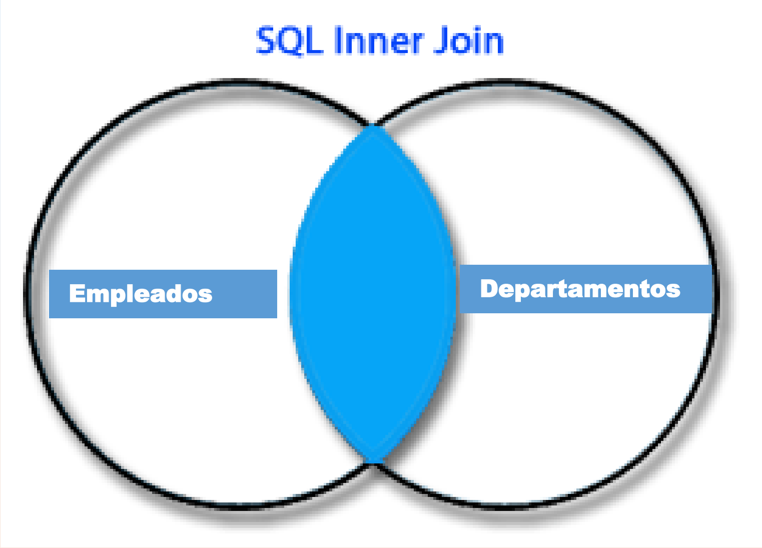


id	nombre	departamento
1	Rafferty	31
2	Jones	33
3	Thomas	33
4	Robinson	34
5	Smith	34
6	Williams	NULL

Departamentos

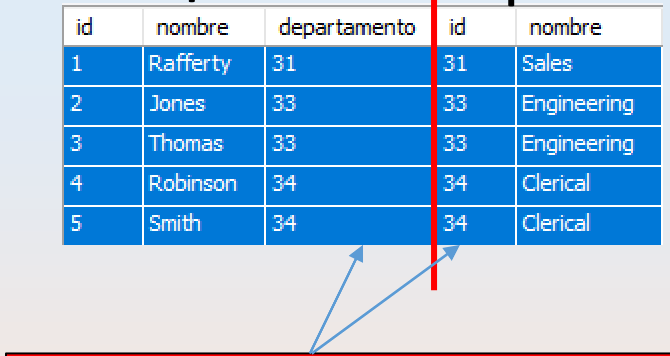


id	nombre
31	Sales
33	Engineering
34	Clerical
35	Marketing



```
select *
from Empleados e
inner join Departamentos d
on e.departamento = d.id;
```

Empleados Departamentos



id	nombre	departamento	id	nombre
1	Rafferty	31	31	Sales
2	Jones	33	33	Engineering
3	Thomas	33	33	Engineering
4	Robinson	34	34	Clerical
5	Smith	34	34	Clerical

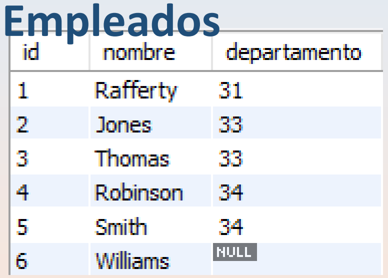
Only records which e.departamento is equal than d.id

Empleados(id int not null auto_increment, **nombre** varchar(30), **departamento** int);

Departamentos(id int, **nombre** varchar(30));

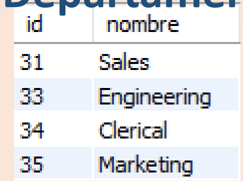
LEFT OUTER JOIN

Empleados

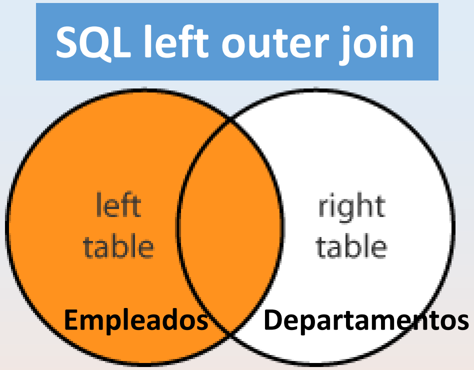


id	nombre	departamento
1	Rafferty	31
2	Jones	33
3	Thomas	33
4	Robinson	34
5	Smith	34
6	Williams	NULL

Departamentos

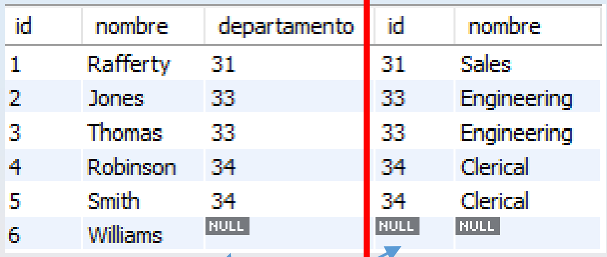


id	nombre
31	Sales
33	Engineering
34	Clerical
35	Marketing



SQL left outer join

Empleados Departamentos



id	nombre	departamento	id	nombre
1	Rafferty	31	31	Sales
2	Jones	33	33	Engineering
3	Thomas	33	33	Engineering
4	Robinson	34	34	Clerical
5	Smith	34	34	Clerical
6	Williams	NULL	NULL	NULL

Every records of the left table and on the right one records that match the id and if there is any match NULL is inserted

```
select *
from Empleados e
left outer join Departamentos d
on e.departamento = d.id;
```

Empleados(id int not null auto_increment, **nombre** varchar(30), **departamento** int);

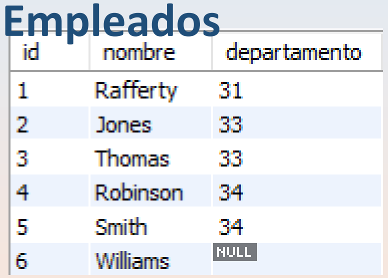
Departamentos(id int, **nombre** varchar(30));

Teacher: Alonso Rebolo

Academic Course 2019-2020

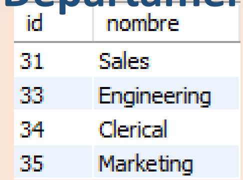
RIGHT OUTER JOIN

Empleados



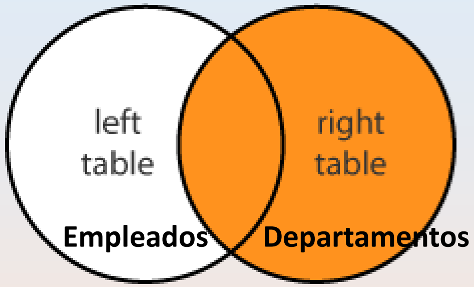
id	nombre	departamento
1	Rafferty	31
2	Jones	33
3	Thomas	33
4	Robinson	34
5	Smith	34
6	Williams	NULL

Departamentos

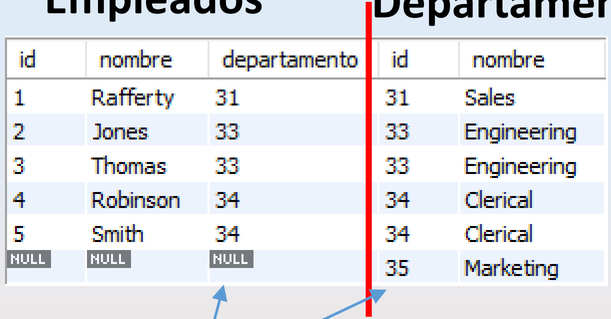


id	nombre
31	Sales
33	Engineering
34	Clerical
35	Marketing

SQL right outer join



Empleados Departamentos



id	nombre	departamento	id	nombre
1	Rafferty	31	31	Sales
2	Jones	33	33	Engineering
3	Thomas	33	33	Engineering
4	Robinson	34	34	Clerical
5	Smith	34	34	Clerical
NULL	NULL	NULL	35	Marketing

Every records of the right table and on the left one records that match the id and if there is any match NULL is inserted

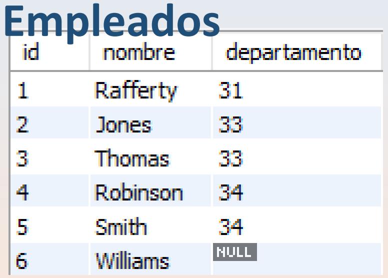
```
select *
from Empleados e
right outer join Departamentos d
on e.departamento = d.id;
```

Empleados(id int not null auto_increment, **nombre** varchar(30), **departamento** int);

Departamentos(id int, **nombre** varchar(30));

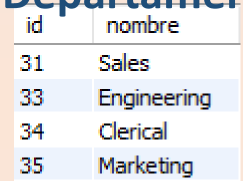
WHERE CLAUSE (INNER JOIN)

Empleados



id	nombre	departamento
1	Rafferty	31
2	Jones	33
3	Thomas	33
4	Robinson	34
5	Smith	34
6	Williams	NULL

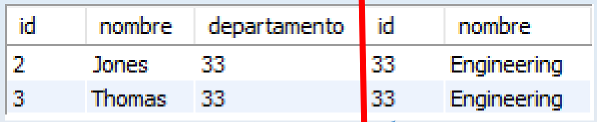
Departamentos



id	nombre
31	Sales
33	Engineering
34	Clerical
35	Marketing

```
select *
from Empleados e
inner join Departamentos d
on e.departamento = d.id
where e.departamento > 31 and e.departamento <34;
```

Empleados Departamentos



id	nombre	departamento	id	nombre
2	Jones	33	33	Engineering
3	Thomas	33	33	Engineering

Only records which e.departamento is equal than d.id and e.departamento Is greater tan 31 and less tan 34

Empleados(id int not null auto_increment, **nombre** varchar(30), **departamento** int);

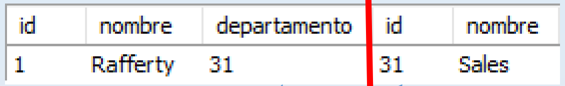
Departamentos(id int, **nombre** varchar(30));

Teacher: Alonso Rebolo

Academic Course 2019-2020

WHERE CLAUSE (LEFT OUTER JOIN)

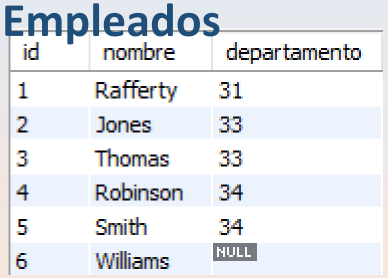
Empleados Departamentos



id	nombre	departamento	id	nombre
1	Rafferty	31	31	Sales

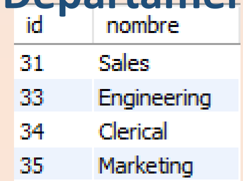
Every records on the left table which e.Departamento is not 33 or 34 and the Correspond records on the right table

Empleados



id	nombre	departamento
1	Rafferty	31
2	Jones	33
3	Thomas	33
4	Robinson	34
5	Smith	34
6	Williams	NULL

Departamentos



id	nombre
31	Sales
33	Engineering
34	Clerical
35	Marketing

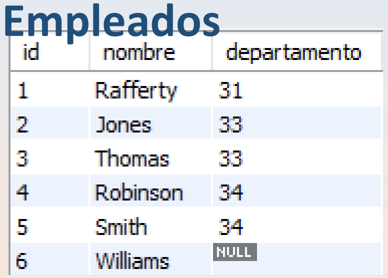
```
select *
from Empleados e
left outer join Departamentos d
on e.departamento = d.id
where e.departamento not in (33,34);
```

Empleados(id int not null auto_increment, **nombre** varchar(30), **departamento** int);

Departamentos(id int, **nombre** varchar(30));

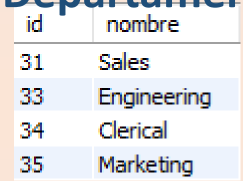
WHERE CLAUSE (RIGHT OUTER JOIN)

Empleados



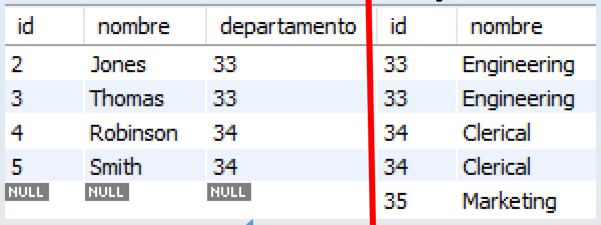
id	nombre	departamento
1	Rafferty	31
2	Jones	33
3	Thomas	33
4	Robinson	34
5	Smith	34
6	Williams	NULL

Departamentos



id	nombre
31	Sales
33	Engineering
34	Clerical
35	Marketing

Empleados Departamentos



id	nombre	departamento	id	nombre
2	Jones	33	33	Engineering
3	Thomas	33	33	Engineering
4	Robinson	34	34	Clerical
5	Smith	34	34	Clerical
NULL	NULL	NULL	35	Marketing

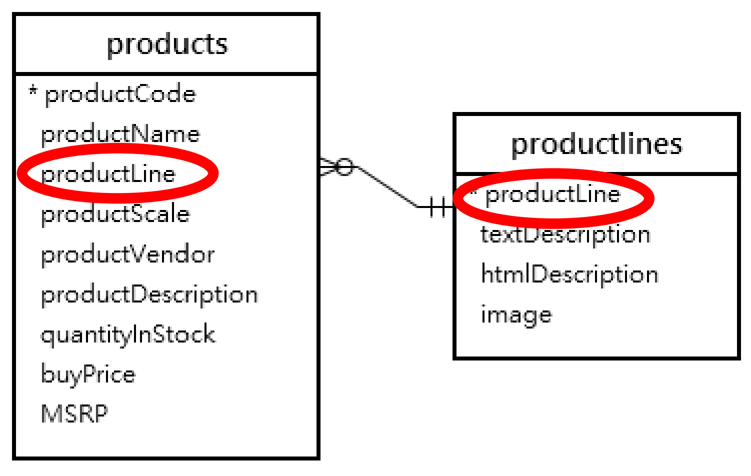
Every records on the right table which
d.is is not 31 and the corresponds records on the right table

```
select *
from Empleados e
right outer join Departamentos d
on e.departamento = d.id
where d.id != 31;
```

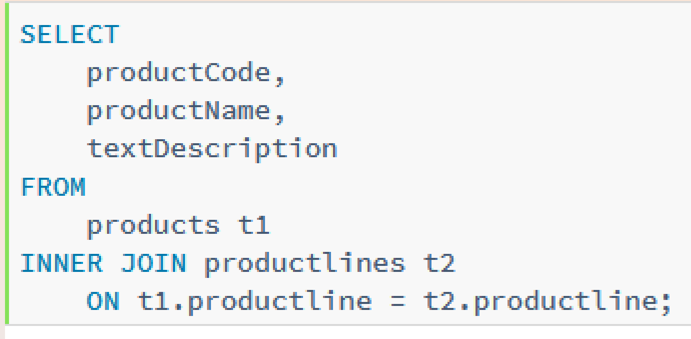
Empleados(id int not null auto_increment, **nombre** varchar(30), **departamento** int);

Departamentos(id int, **nombre** varchar(30));

USING sintax

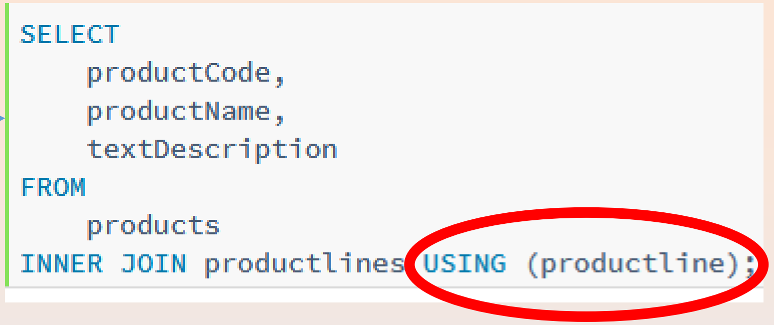


Field names are the same in both tables



```
SELECT
    productCode,
    productName,
    textDescription
FROM
    products t1
INNER JOIN productlines t2
    ON t1.productline = t2.productline;
```

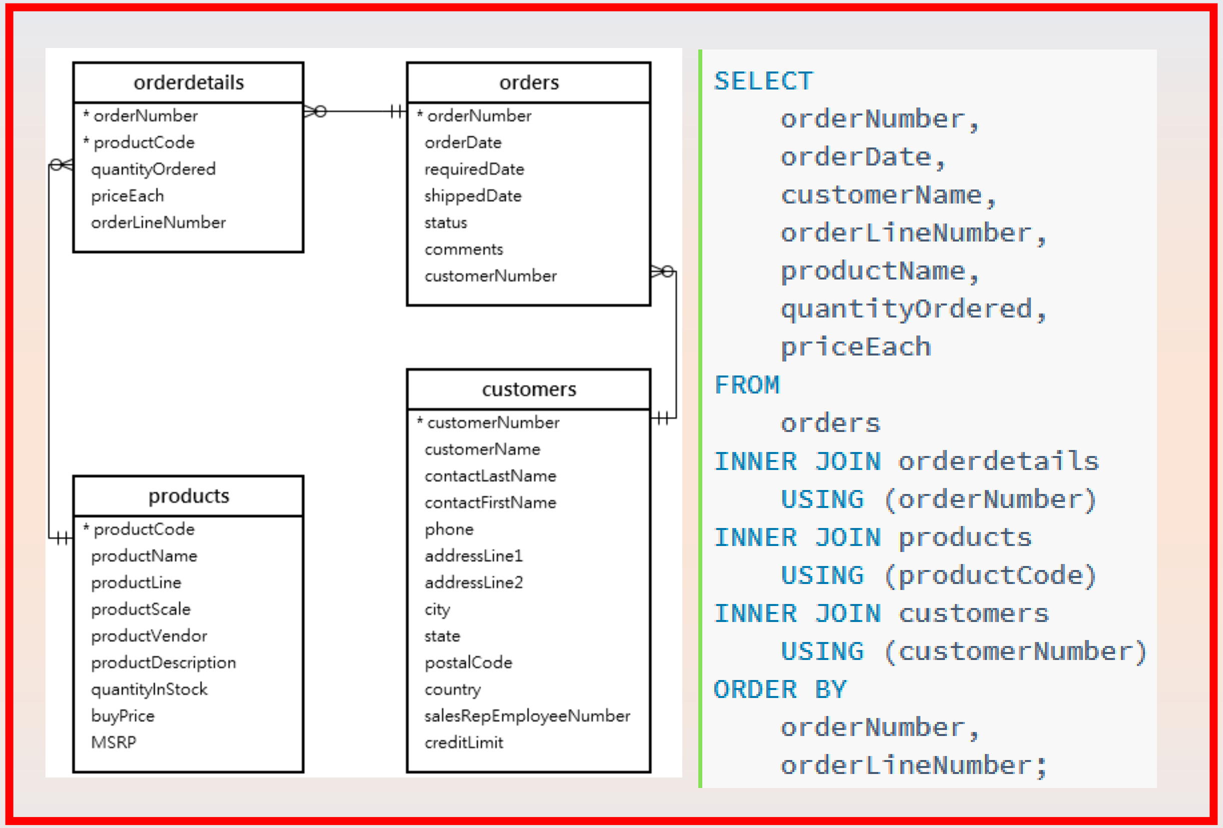
← **Synonym** →



```
SELECT
    productCode,
    productName,
    textDescription
FROM
    products
INNER JOIN productlines USING (productline);
```

More than two tables

Four Tables



orderdetails
* orderNumber
* productCode
quantityOrdered
priceEach
orderLineNumber

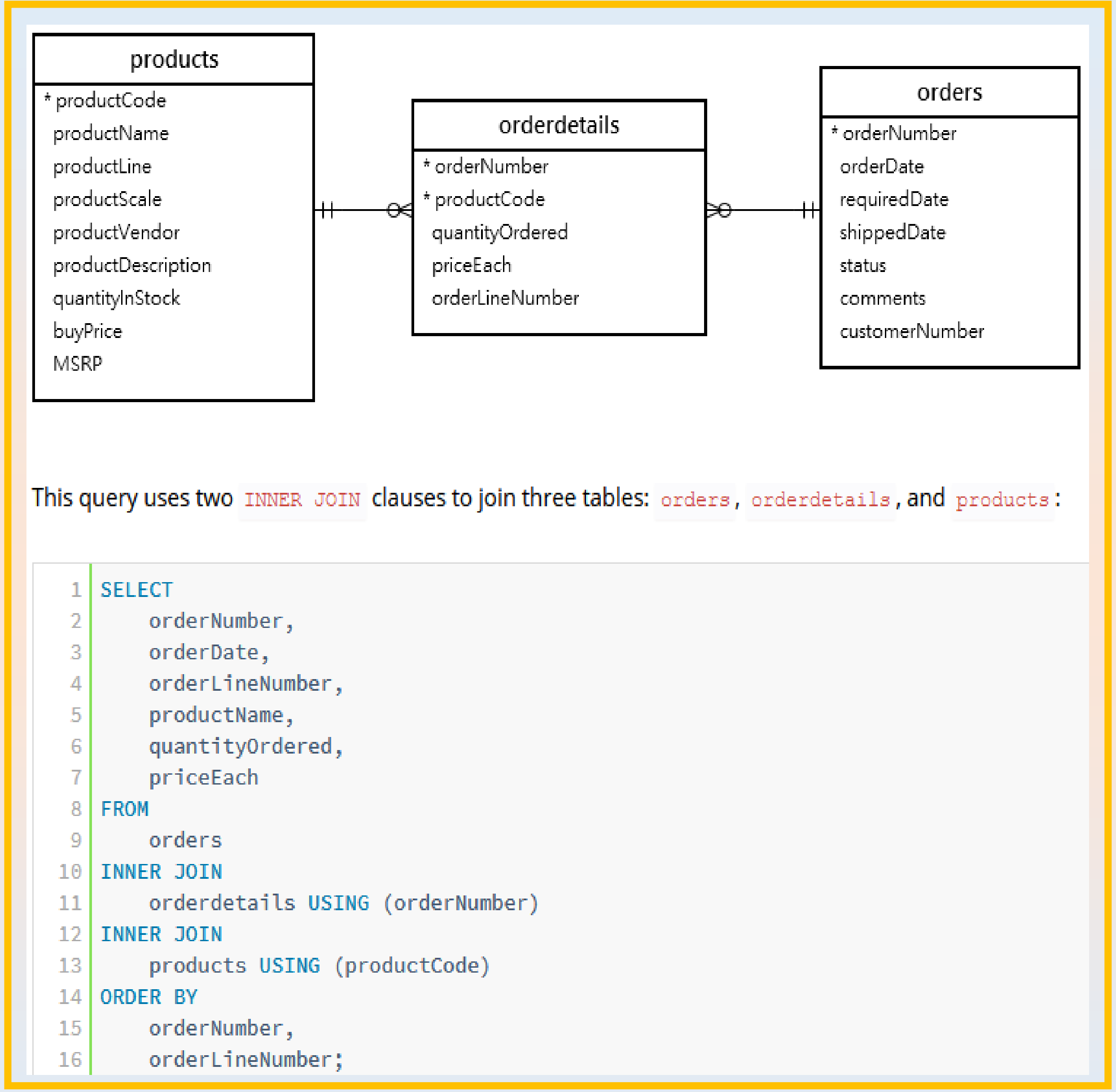
orders
* orderNumber
orderDate
requiredDate
shippedDate
status
comments
customerNumber

customers
* customerNumber
customerName
contactLastName
contactFirstName
phone
addressLine1
addressLine2
city
state
postalCode
country
salesRepEmployeeNumber
creditLimit

products
* productCode
productName
productLine
productScale
productVendor
productDescription
quantityInStock
buyPrice
MSRP

```sql
SELECT
    orderNumber,
    orderDate,
    customerName,
    orderLineNumber,
    productName,
    quantityOrdered,
    priceEach
FROM
    orders
INNER JOIN orderdetails
    USING (orderNumber)
INNER JOIN products
    USING (productCode)
INNER JOIN customers
    USING (customerNumber)
ORDER BY
    orderNumber,
    orderLineNumber;
```

Three Tables



products
* productCode
productName
productLine
productScale
productVendor
productDescription
quantityInStock
buyPrice
MSRP

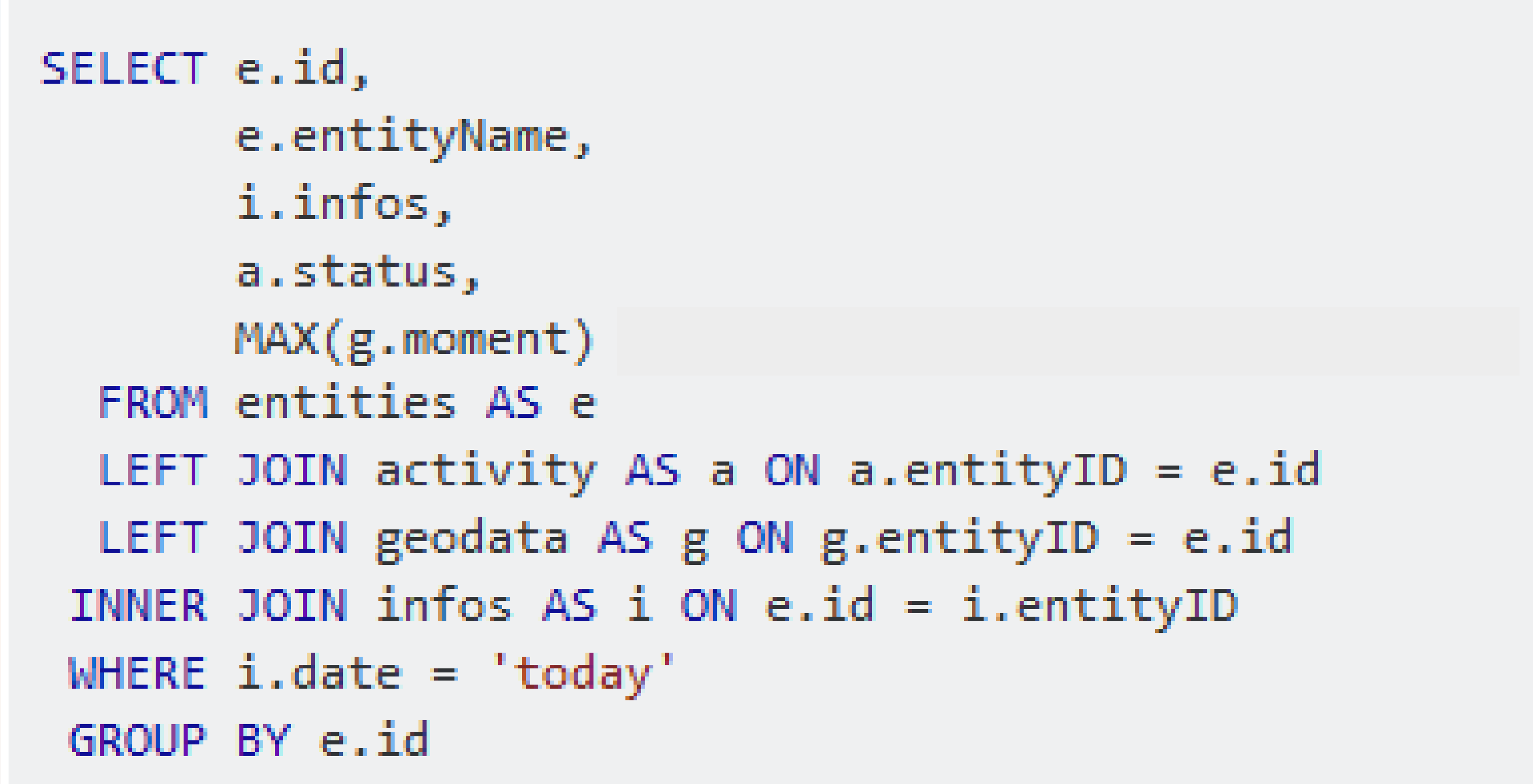
orderdetails
* orderNumber
* productCode
quantityOrdered
priceEach
orderLineNumber

orders
* orderNumber
orderDate
requiredDate
shippedDate
status
comments
customerNumber

This query uses two `INNER JOIN` clauses to join three tables: `orders`, `orderdetails`, and `products`:

```sql
SELECT
    orderNumber,
    orderDate,
    orderLineNumber,
    productName,
    quantityOrdered,
    priceEach
FROM
    orders
INNER JOIN
    orderdetails USING (orderNumber)
INNER JOIN
    products USING (productCode)
ORDER BY
    orderNumber,
    orderLineNumber;
```

More than two tables combining left and inner joing



```sql
SELECT e.id,
       e.entityName,
       i.infos,
       a.status,
       MAX(g.moment)
  FROM entities AS e
  LEFT JOIN activity AS a ON a.entityID = e.id
  LEFT JOIN geodata AS g ON g.entityID = e.id
 INNER JOIN infos AS i ON e.id = i.entityID
 WHERE i.date = 'today'
 GROUP BY e.id
```

Working explanation when are more tan two tables

```
SELECT t1.col, t3.col
FROM table1 t1
join table2 t2
ON t1.primarykey = t2.foreignkey
join table3 t3
ON t2.primarykey = t3.foreignkey;
```

We first join table1 and table2 which produce a temporary table with combined data from table1 and table2 which is then joined to table3.

This formula can be extended to more than 3 tables to N tables.

You just need to make sure that SQL query should have N-1 join statement in order to join N tables. For joining two tables we require 1 join statement and for joining 3 tables we need 2 join statement