

INGENIERÍA INFORMÁTICA
Escuela Politécnica Superior
Universidad Autónoma De Madrid

Estructura de Datos

Memoria práctica 1

Miguel Angel Martinez Parra
Sergio Sánchez Alcalde

20/10/2020

CLAVES PRIMARIAS Y CLAVES EXTERNAS

public.customers(**customernumber**, customername, contactlastname, contactfirstname, phone, addressline1, addressline2, city, state, postalcode, country, salesrepemployeenumber → public.customers(employeeenumber), creditlimit)

public.employees(**employeeenumber**, lastname, firstname, extension, email, officecode → public.offices(officecode), reportsto → public.employees(employeeenumber), jobtitle)

public.offices (**officecode**, city, phone, addressline1, addressline2, state, country, postalcode, territory)

public.orderdetails (**ordernumber** → public.orders(ordernumber), **productcode** → public.products(productcode), quantityordered, priceeach, orderlinenumber)

public.orders (**ordernumber**, orderdate, requireddate, shippeddate, status, comments, customernumber → public.customers(customernumber))

public.payments (**customernumber** → public.customers(customernumber), **checknumber**, paymentdate, amount)

public.productlines (**productline**, textdescription, htmldescription, image)

public.products (**productcode**, productname, productline → public.productlines(productline), productscale, productvendor, productdescription, quantityinstock, buyprice, msrp)

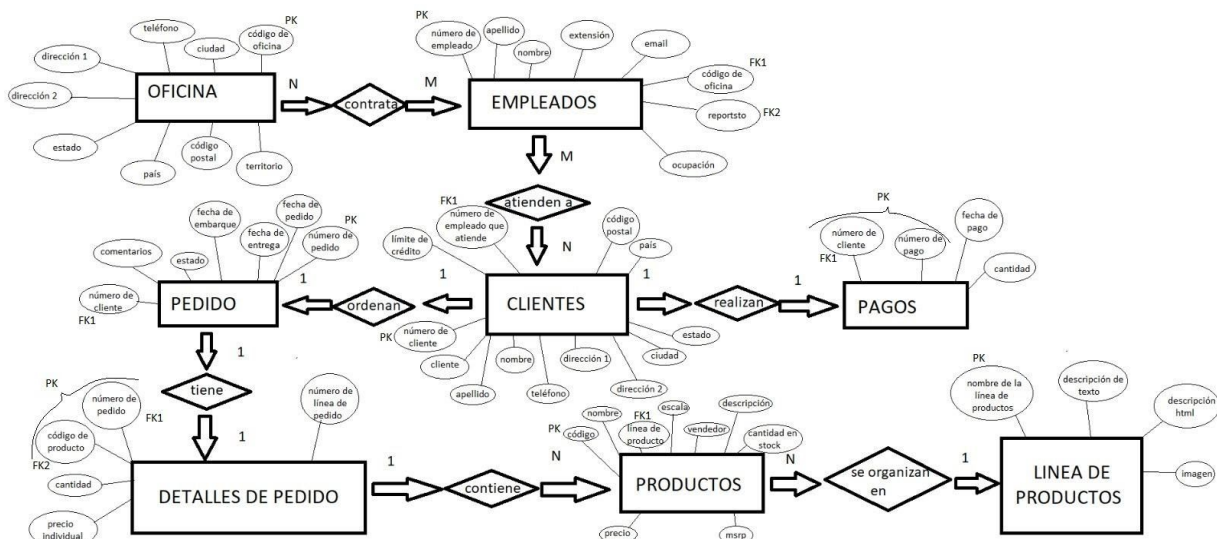
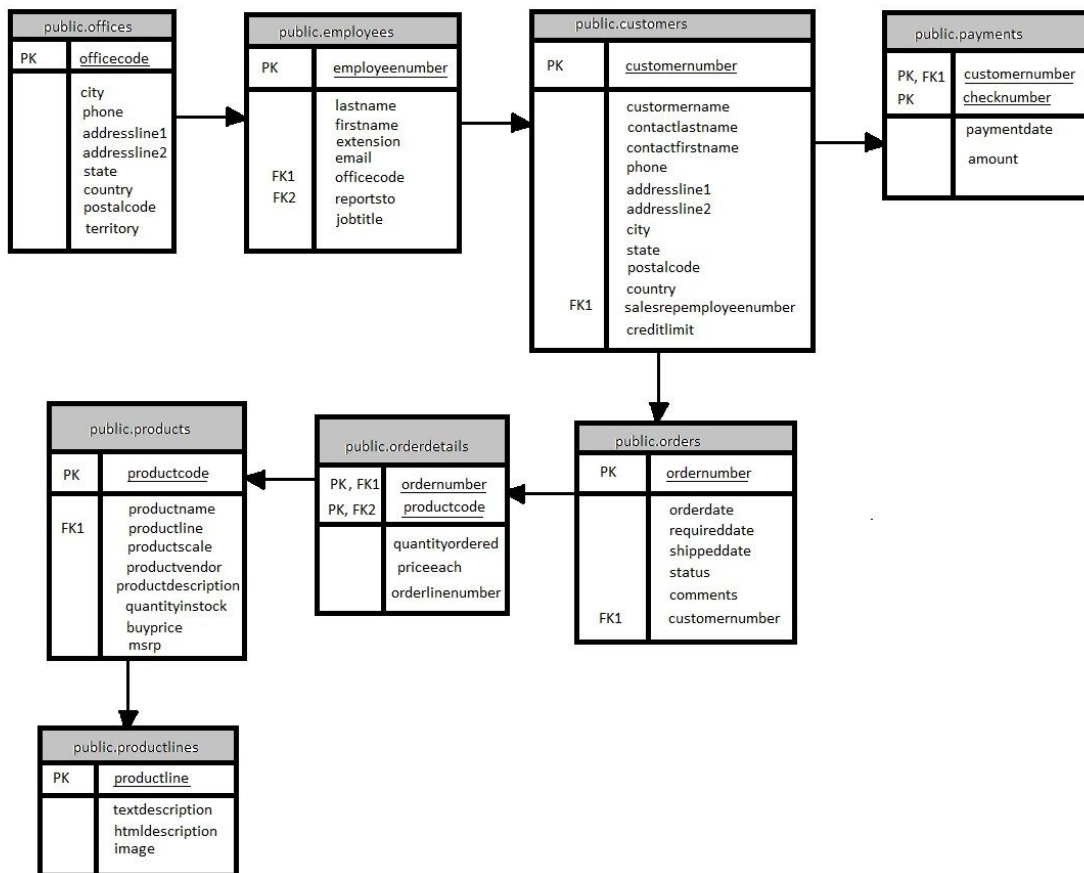


DIAGRAMA DEL MODELO RELACIONAL



CONSULTAS

- Query 1:

Para este primer query hemos utilizado un dos select para sacar primero los datos que se nos pedían en el enunciado y otro para marcar la condición de que fuera ese producto. Obtenemos el total haciendo la suma de todos los pagos.

```

select

    c.customername,

    c.customernumber,

    sum(p.amount) as Total

from
    
```

```
customers c , payments p

where c.customernumber = p.customernumber and
c.customernumber in (

    select

        o.customernumber

    from

        orders o,

        orderdetails o2,

        products p2

    where

        o2.ordernumber = o.ordernumber

        and o2.productcode = p2.productcode

        and p2.productname = '1940 Ford Pickup Truck')

group by

    c.customername,

    c.customernumber

order by

    Total desc
```

- Query 2:

Para este query lo que hemos hecho ha sido seleccionar el productline para que nos lo muestre al ejecutar el query. También hemos utilizado un join para unir las tablas de product, order y orderdetails. Utilizamos el comando average para

sacar el tiempo medio que tarda desde que se realiza el pedido hasta que es enviado.

```
select

    p.productline,

    AVG(o1.shippeddate-o1.orderdate) as media

from

    orders o1 join orderdetails o2 on
o1.ordernumber=o2.ordernumber join products p on
p.productcode=o2.productcode

group by

    p.productline
```

- Query 3:

Para este query lo primero que hemos hecho ha sido buscar al director, que es lo último que aparece en el código. Luego hemos buscado los empleados que tenían contacto entre ellos y llegaron hasta el director.

```
select

    e3.employeenumber, e3.lastname

from

    employees e3

where

    e3.reportsto in (

select

    e2.employeenumber

from

    employees e2
```

```
where

    e2.reportsto=(

select

    e.employeenumber

from

    employees e

where

    e.reportsto is NULL))
```

- Query 4:

Para esta query verificamos el código usando la tabla de empleados de la oficina y el número de empleado de ventas usando la referencia de empleados. Luego tomamos el número de cliente usando la tabla del cliente y el número de pedido usando la tabla de pedidos. Finalmente sumamos la cantidad pedida en orden decreciente y tomamos la primera con la cláusula "límite".

```
select

    o.officecode,

    sum(o3.quantityordered) as ProductosVendidos

from

    employees e

    join offices o on e.officecode = o.officecode

    join customers c on c.salesrepemployeenumber =

e.employeenumber

    join orders o1 on o1.customernumber = c.customernumber

    join orderdetails o3 on o3.ordernumber = o1.ordernumber

group by

    o.officecode
```

```
order by

    ProductosVendidos DESC

LIMIT 1;
```

- Query 5:

Para este query hemos buscado las oficinas que no han vendido nada durante el año 2003. Para eso hacemos un join de las tablas orders, customers y employees. Después las agrupamos por países y las ordenamos por el número de oficinas. Llegamos a la conclusión de que no hay ninguna que no haya vendido nada.

```
select
    o.country as Pais,
    Count(*) as NumOficinas
from
    offices o
where not exists(
    select
        o2.ordernumber
    from
        orders o2
        join customers c on c.customernumber = o2.customernumber
        join employees e on c.salesrepemployeenumber = e.employeenumber
    where
        o2.orderdate >= '2003-01-01'
        and o2.orderdate <= '2003-12-31'
        and o2.status <> 'Cancelled'
        and e.officecode = o.officecode
)
group by
    Pais
order by
    NumOficinas DESC;
```

- Query 6:

Utilizamos la tabla orderdetails dos veces para encontrar las parejas y hacemos que no nos enseñe dos veces la misma pareja. Agrupamos las parejas y vemos cuántas veces aparecen.

```
select
    o.productcode as Producto1,
    o2.productcode as Producto2,
    Count(o.ordernumber) as Carro
from
    orderdetails o
    join orderdetails o2 on o.ordernumber = o2.ordernumber
    and o.productcode < o2.productcode
group by
    o.productcode,
    o2.productcode
having
    Count(o.ordernumber) > 1;
```