

Fundamentos de Bases de Datos

Práctica 1

Miguel Lozano Alonso

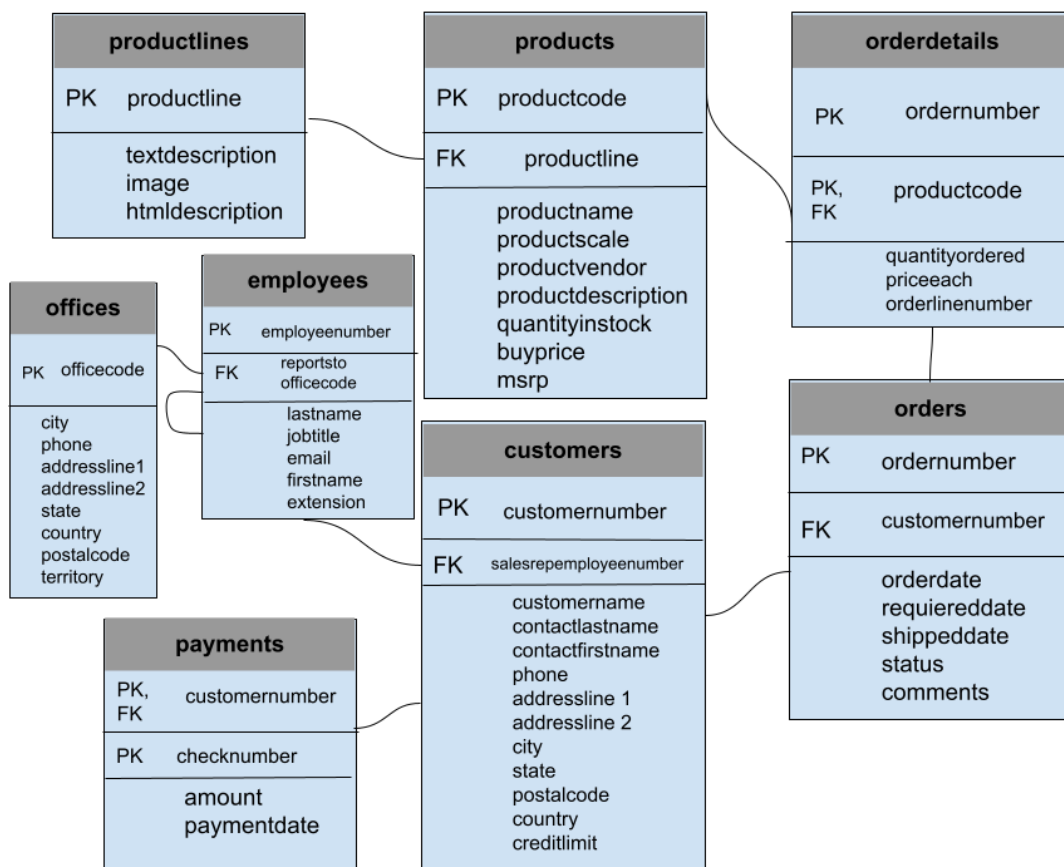
Cristina Rodríguez de los Ríos Medina

Esquema de la base de datos	3
Diagrama Relacional de la base de datos	3
Querys:	4
1. Query 1	4
2. Query 2	4
3. Query 3	5
4. Query 4	5
5. Query 5	6
6. Query 6	7

Esquema de la base de datos

- productlines(**productline**, textdescription, htmldescription, image)
- products(**productcode**, productname, productline -> productline.productline, productscale, productvendor, productdescription, quantityinstock, buyprice, msrp)
- orderdetails(**ordernumber** -> **orders.ordernumber**, **productcode** -> **products.productcode**, quantityordered, priceeach, orderlinenumber)
- orders(**ordernumber**, orderdate, requireddate, shippeddate, status, comments, customernumber -> customers.customernumber)
- customers (**customernumber**, salesrepemployeenumber -> employees.employeenumber, customername, contactfirstname, phone, addressline1, addressline2, city, state, postalcode, country, creditlimit)
- employees (**employeenumber**, reportsto -> employees.employeenumber, officecode -> office.officecode, lastname, jobtitle, email, firstname, extension)
- offices (**officecode**, city, phone, adressline1, adessline2, state, country, postalcode, territory)
- payments (**customernumber** -> customers.customernumber, **checknumber**, amount, paymentdate)

Diagrama Relacional de la base de datos



Querys:

1. Query 1

Dinero total abonado por los clientes que han adquirido el “1940 Ford Pickup Truck”

```
SELECT c.customernumber,  
       c.customername,  
       Sum(p.amount)  
FROM   customers c  
       natural JOIN payments p  
WHERE  c.customernumber IN (SELECT c.customernumber  
                           FROM   customers c  
                           natural JOIN orders o  
                           natural JOIN orderdetails od  
                           natural JOIN products p  
                           WHERE  p.productname = '1940 Ford Pickup Truck')  
GROUP BY c.customernumber  
ORDER BY Sum(p.amount) DESC;
```

Para la implementación de esta consulta se han hecho dos select anidados. En el interno se seleccionan los clientes que han adquirido el producto “1940 Ford Pickup Truck”, y en el externo, se seleccionan esos clientes y la cantidad de dinero que han gastado en cualquier compra de la tabla *payments* que coincida con su *customernumber*.

2. Query 2

Tiempo medio entre la orderdate y la shippeddate de una pedido agrupado por tipo de producto

```
SELECT p.productline,  
       Round(Avg(o.shippeddate - o.orderdate), 2) average_time  
FROM   products p  
       natural JOIN orderdetails od  
       natural JOIN orders o  
GROUP BY p.productline  
ORDER BY average_time DESC;
```

Para obtener la diferencia entre la fecha en la que se realiza el pedido y la fecha en la que se envía, se hace directamente con la operación de resta de esos dos campos. Después se obtiene la media haciendo uso de la función avg() de SQL.

3. Query 3

Empleados que reportan a otros empleados que reportan al director

```
SELECT e3.employeenumber,  
       e3.lastname  
FROM   employees e3  
WHERE  e3.reportsto IN (SELECT e2.employeenumber  
                        FROM   employees e2  
                        WHERE  e2.reportsto IN (SELECT e.employeenumber  
                                                FROM   employees e  
                                                WHERE  e.reportsto IS NULL));
```

Para realizar la consulta, primero, seleccionamos el employeenumber del jefe. Después, seleccionamos el employeenumber de los empleados que reportan al jefe. Y por último, seleccionamos definitivamente a los empleados que reportan a los empleados que reportan al jefe.

4. Query 4

Oficina que ha vendido el mayor número de objetos

```
SELECT o1.officecode,  
       Sum(o3.quantityordered) AS number_of_product_sales  
FROM   offices o1  
       NATURAL JOIN employees e  
       JOIN customers c  
         ON e.employeenumber = c.salesrepemployeenumber  
       NATURAL JOIN orders o2  
       NATURAL JOIN orderdetails o3  
GROUP  BY o1.officecode  
ORDER  BY number_of_product_sales DESC  
LIMIT 1;
```

El funcionamiento de la consulta es el siguiente: Buscamos de cada oficina cuántos elementos ha vendido cada empleado. Después, agrupamos las ventas de cada oficina y las sumamos. Por último,

ordenamos de mayores a menores ventas cada oficina y seleccionamos el primer resultado junto con sus ventas totales.

5. Query 5

Países con al menos una oficina sin ventas en 2003

```
SELECT consulta.country,
       Count(*) num_offices
FROM   (SELECT o.country country,
              o.officecode
        FROM   offices o
        EXCEPT
        (SELECT op.country,
              op.oc
        FROM   (SELECT o.country          country,
                      o.officecode        oc,
                      Count(p.checknumber) pagos
                  FROM   offices o
                        natural JOIN employees e
                        JOIN customers c
                              ON e.employeenumber = c.salesrepemployeenumber
                        natural JOIN payments p
                  WHERE  Extract(year FROM p.paymentdate) = '2003'
                  GROUP BY o.country,
                          o.officecode) AS op)) AS consulta
GROUP BY consulta.country
```

Para realizar esta consulta se han hecho tres select anidados, extrayendo los pagos de cada oficina, ya que un producto se considera vendido una vez que se ha realizado el pago.

El select más interno, denominado op, extrae los pagos realizados a cada oficina con su correspondiente país en 2003.

El segundo select más interno, denominado consulta, extrae las oficinas con su correspondiente país que no obtuvieron pagos en 2003, haciendo uso del operador EXCEPT de SQL.

El select más externo, cuenta el número de oficinas de “consulta” por país, obteniendo por tanto los países que tienen al menos una oficina que no realizó ninguna venta en 2003.

6. Query 6

Listado de las parejas de productos que aparezcan en más de un carro de compra

```
SELECT p1.productcode,  
       p2.productcode,  
       Count(o2) AS carros  
FROM   products p1  
       natural JOIN orderdetails o1  
       natural JOIN orders o2,  
       products p2  
       natural JOIN orderdetails o  
       natural JOIN orders o3  
WHERE  o1.ordernumber = o.ordernumber  
       AND o2.ordernumber = o3.ordernumber  
       AND p1.productcode < p2.productcode  
GROUP BY p1.productcode,  
         p2.productcode  
ORDER BY carros DESC;
```

El proceso que realiza la consulta es el siguiente: Primero, relacionamos mediante joins las tablas: products, orders y orderdetails. Después, agrupamos cada productcode con el productcode de otro producto que está en la misma cesta. Por último, seleccionamos las parejas de productos que están en la misma cesta, el número de cestas en las que coinciden y las ordenamos por el número de cestas en las que coinciden de mayor a menor.