PRÁCTICA 1 DE BASES DE DATOS

14/10/2022

PABLO ANTÓN ALAMILLO

EDUARDO JUNOY ORTEGA

LABORATORIO 08b

2.FAMILIARIZARSE CON LA TABLA.

products(**productCode**, productName, productLine→productlines.productline, productScale, productVendor, productDescription, quantityInStock, buyPrice, MSRP)

productlines(**productLines**, textDescription, htmlDescription, image)

orderdetails(**orderNumber**→orders.orderNumber, **productCode**→products.prooductCode, quantityOrdered, priceEach, orderLineNumber)

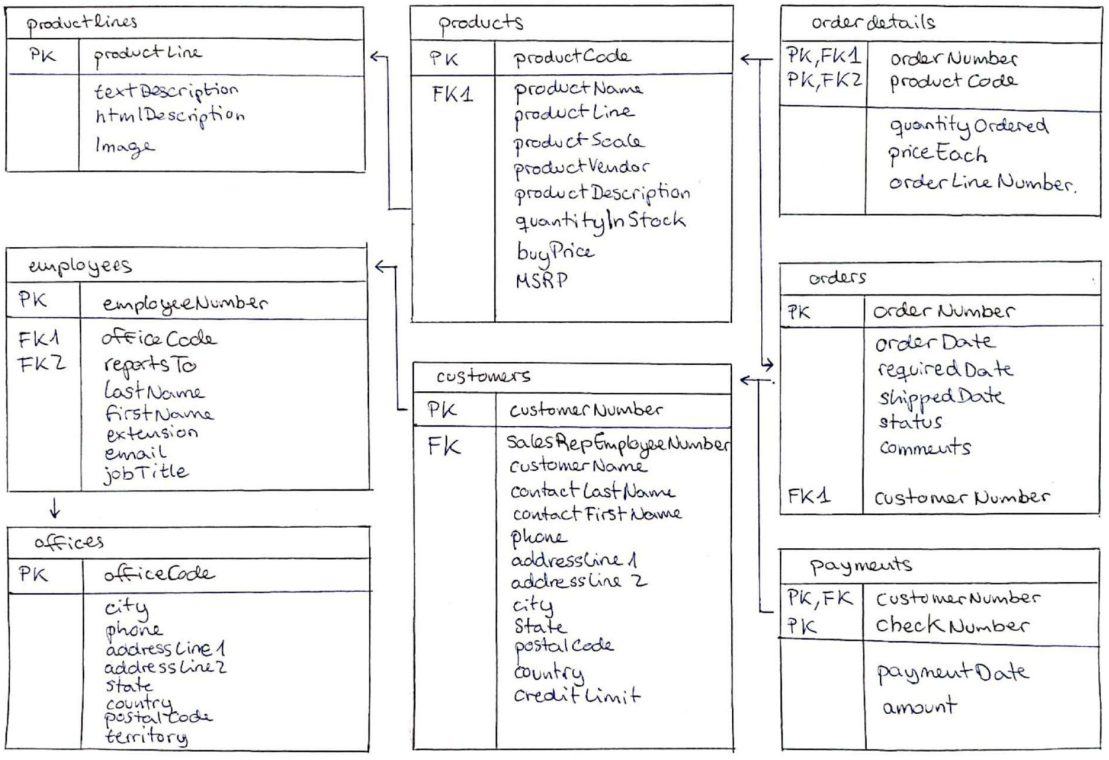
orders(**orderNumber,** orderDate, requiredDate, shipedDate, status, comments, customerNumber→customers.customerNumber)

customers(**customerNumber**, customerName, contactLastName, contactFirstName, phone, adressLine1, adressline2, city, state, postalCode, country, salesRepEmpliyeeNumber→employees.employeeNumber, creditLimit)

payments(**customerNumber**→customers.customerNumber**, checkNumber**, paymentDate, ammount)

employees(**employeeNumber**, lastName, firstName, extension, email, officeCode→offices.officeCode, reportsTo→employees.employeeNumber, jobTitle)

offices(**officeCode**, city, phone, adressLine1, adressLine2, state, country, postalCode, territory)



3.CUESTIONES

1. **Consulta 1:**

```
SELECT s1.customernumber,
       s1.customername,
       Sum(pm.amount)
FROM   (SELECT DISTINCT c.customernumber,
                        c.customername
        FROM   customers c
               JOIN orders o
                 ON c.customernumber = o.customernumber
               JOIN orderdetails od
                 ON o.ordernumber = od.ordernumber
               JOIN products p
                 ON od.productcode = p.productcode
        WHERE  p.productname = '1940 Ford Pickup Truck') AS s1
       JOIN payments pm
         ON s1.customernumber = pm.customernumber
GROUP  BY s1.customernumber,
          s1.customername
ORDER  BY Sum(pm.amount) DESC;
```

Unimos todas las tablas necesarias de forma anidada y las ordenamos según la

2. **Consulta 2:**

```
SELECT productline,
       Avg(shippeddate - orderdate) AS avg_time
FROM   productlines
       natural JOIN products
       natural JOIN orderdetails
       natural JOIN orders
GROUP  BY productline;
```

3. **Consulta 3:**

```
WITH boss
     AS (SELECT employeenumber
         FROM   employees
         WHERE  reportsto IS NULL),
     employee1
     AS (SELECT e.employeenumber
         FROM   employees e,
                boss b
         WHERE  e.reportsto = b.employeenumber)
SELECT e.employeenumber,
       e.lastname
FROM   employee1 e1,
       employees e
WHERE  e.reportsto = e1.employeenumber;
```

4. **Consulta 4:**

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

5. **Consulta 5:**

```
WITH yearorders
     AS (SELECT ordernumber
         FROM   orders
         WHERE  orders.orderdate >= '2003-01-01'
                AND orders.orderdate <= '2003-12-31'
        ),
     noorders
     AS (SELECT officecode,
                country
         FROM   offices OF
         WHERE  NOT EXISTS(SELECT ordernumber
                           FROM   yearorders
                                  natural JOIN customers
                                  JOIN employees
                                    ON employeenumber = salesrepemployeenumber
                           WHERE  officecode = OF.officecode))
SELECT country,
       Count(*) AS number_offices
FROM   noorders
GROUP  BY country
HAVING Count(*) > 0
ORDER  BY Count(*) DESC;
```

6. **Consulta 6:**

```
SELECT p1.productcode AS p1,
       p2.productcode AS p2,
       Count(p1.ordernumber)
FROM   orderdetails p1
       JOIN orderdetails p2
         ON p1.ordernumber = p2.ordernumber
WHERE  p1.productcode < p2.productcode
GROUP  BY p1.productcode,
          p2.productcode
HAVING Count(p1.ordernumber) > 1
ORDER  BY p1.productcode,
          p2.productcode;
```

4.REDISEÑO DE LA BASE DE DATOS