**TUGAS SISTEM MANAJEMEN BASIS DATA**

**TUGAS 1**

Disusun guna memenuhi tugas mata kuliah Sistem Manajemen Basis Data

Kelas B

Pengampu: Bana Handaga



Oleh:

Sigit Rizqi Ramadan

L200150103

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS KOMUNIKASI DAN INFORMATIKA**

**UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**2017**

**MANAJEMEN BASISDATA**

**Rancangan Database**

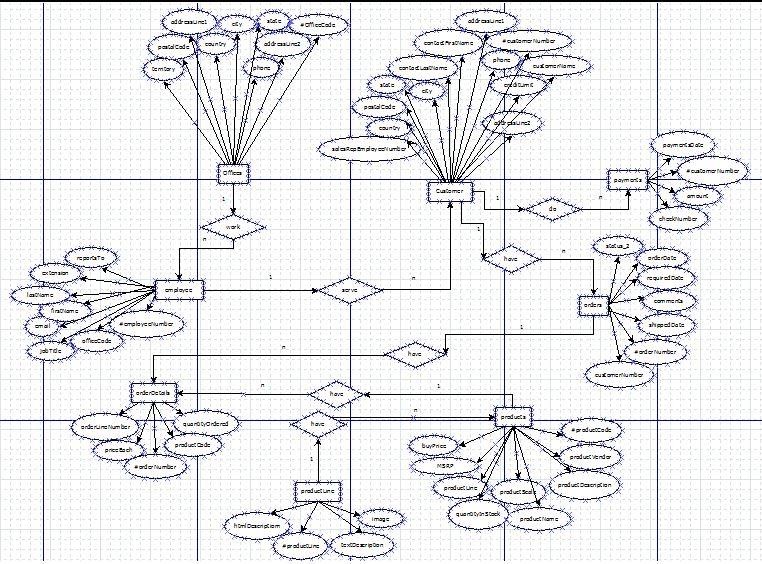
**RETAIL**

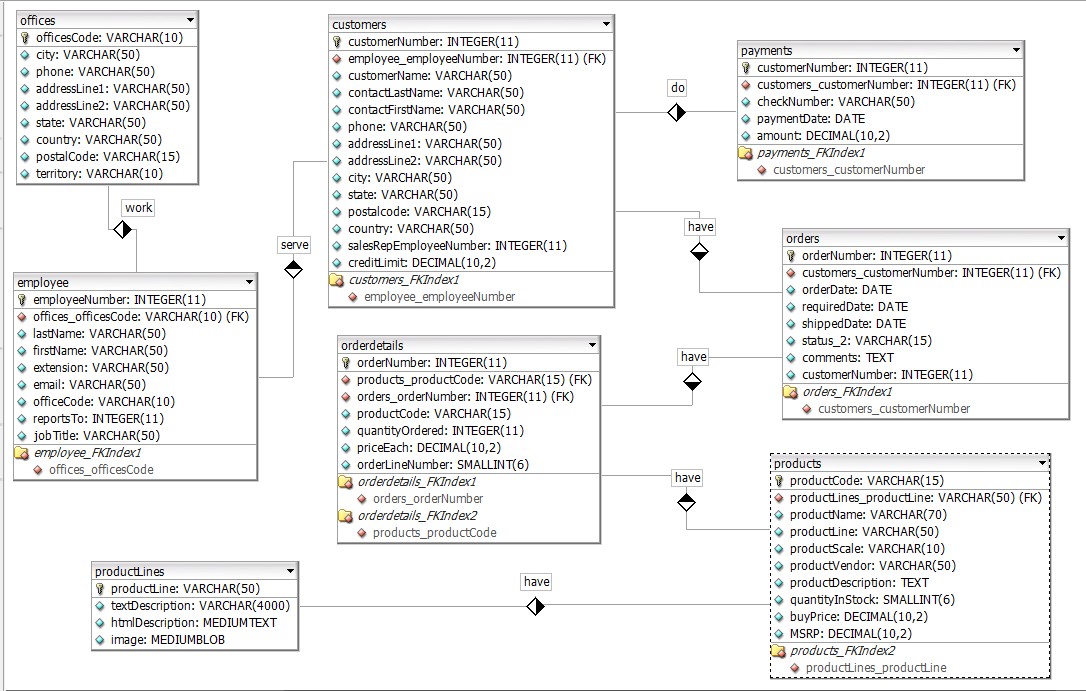
Pengantar :

Database pada sebuah sorum motor, dimana sorum itu juga

mempunyai PT yang memproduksi.Rancangan dan detail produk disimpan pada data productilines.Setelah produk siap dipasarkan semua data terkait produk seperti kode produk,nama produuk,jumlah produk,dll disimpan pada entity products. Setelah produk dipesan,data pemesanan disimpan di order seperti nomor order,tanggal,waktu dan status dan detail dari pemesanan seperti nomor pemesanan,kode produk,jumlah pesanan disimpan di orderdetails. Data pelanggan yang melakukan pemesanan disimpan di table costumer termasuk nomor dari karyawan yang melayani. Data dari karyawan yang berada di kantor disimpan di table employees. Data mengenai informasi kantor disimpan di Offices. Data pembayaran yang dilakukan oleh costumer disimpan di Payments.

Gambar ER-Diagram





1. Entities yang diperlukan dalam database classicModels :
   1. Customer
   2. Employee
   3. Office
   4. Order
   5. Order Details
   6. Payments
   7. Product
   8. Product Line
2. Menentukan Atribut dari setiap Entities dalam database classicModels:
   1. Customer
      * customerNumber : INTEGER(11) Primary Key
      * employee\_employeeNumber : INTEGER(11) Foreign Key
      * customerName : VARCHAR(50)
      * contactLastName : VARCHAR(50)
      * contactFirstName : VARCHAR(50)
      * phone : VARCHAR(50)
      * addressLine1 : VARCHAR(50)
      * addressLine2 : VARCHAR(50)
      * city : VARCHAR(50)
      * state : VARCHAR(50)
      * postalCode : VARCHAR(15)
      * country : VARCHAR(50)
      * salesRepEmployee : INTEGER(11)
      * creditLimit : DECIMAL(10,2)
   2. Employee
      * employeeNumber : INTEGER(11) Primary Key
      * offices\_officesCode : VARCHAR(10) Foreign Key
      * lastName : VARCHAR(50)
      * firstName : VARCHAR(50)
      * extension : VARCHAR(50)
      * email : VARCHAR(50)
      * officeCode : VARCHAR(10)
      * reportsTo : INTEGER(11)
      * jobTitle : VARCHAR(50)
   3. Offices
      * officesCode : VARCHAR(10) Primary Key
      * city : VARCHAR(50)
      * phone : VARCHAR(50)
      * addressLine1 : VARCHAR(50)
      * addressLine2 : VARCHAR(50)
      * state : VARCHAR(50)
      * country : VARCHAR(50)
      * postalCode : VARCHAR(15)
      * territory : VARCHAR(50)
   4. Order
      * orderNumber : INTEGER(11) Primary Key
      * customer\_customerNumber : INTEGER(11) Foreign Key
      * orderDate : DATE
      * requiredDate : DATE
      * shippedDate : DATE
      * status\_2 : VARCHAR(15)
      * comments : TEXT
      * customerNumber : INTEGER(11)
   5. Order Details
      * orderNumber : INTEGER(11) Primary Key
      * products\_ productCode : VARCHAR(15) Foreign Key
      * orders\_orderNumber : INTEGER(11) Foreign Key
      * productCode : VARCHAR(15)
      * quantityOrdered : INTEGER(11)
      * priceEach : DECIMAL(10,2)
      * orderLineNumber : SMALLINT(6)
   6. Payments
      * customerNumber : INTEGER(11) Primary Key
      * customer\_customerNumber : INTEGER(11) Foreign Key
      * checkNumber : VARCHAR(50)
      * paymentDate : DATE
      * amount : DECIMAL(10,2)
   7. Product
      * productCode : VARCHAR(15) Primary Key
      * productLine\_productLine : VARCHAR(50) Foreign Key
      * productName : VARCHAR(70)
      * productLine : VARCHAR(50)
      * productScale : VARCHAR(10)
      * productVendor : VARCHAR(50)
      * productDescription : TEXT
      * quantityInStock : SMALLINT(6)
      * buyPrice : DECIMAL(10,2)
      * MSRP : DECIMAL(10,2)
   8. Product Line
      * productLine : VARCHAR(50) Primary Key
      * textDescription : VARCHAR(4000)
      * htmlDescription : MEDIUMTEXT
      * image : MEDIUMBLOB
3. Menentukan Hubungan/ Relasi antar Entities dalam database classicModels:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | customer | Offices | products | productLine | orders | Payment | employee | Orderdetails |
| Customer | - |  |  |  | 1:n | 1:n |  |  |
| Offices |  | - |  |  |  |  |  |  |
| Products |  |  | - | 1:n |  |  |  | 1:n |
| ProductLine |  |  |  | - |  |  |  |  |
| Orders |  |  |  |  | - |  |  | 1:n |
| Payments |  |  |  |  |  | - |  |  |
| Employee | 1:n | n:1 |  |  |  |  | - |  |
| OrdersDetails |  |  |  |  |  |  |  | - |

* 1. Customer memiliki Orders Relasi : One – to – Many [1 – n]

Atribut yang menghubungkan Entities Customer dengan Entities Orders yaitu atribut customerNumber : INTEGER(11) yang berada pada Entities Customer, sehingga atribut customerNumber : INTEGER(11) menjadi Foreign Key pada Entities Orders.

* 1. Customer melakukan Payment Relasi : One – to – Many [1 – n]

Atribut yang menghubungkan Entities Customer dengan Entities Payment yaitu atribut customerNumber : INTEGER(11) yang berada pada Entities Customer, sehingga atribut customerNumber : INTEGER(11) menjadi Foreign Key pada Entities Payments.

* 1. Employee melayani Customer Relasi : One – to – Many [1 – n]

Atribut yang menghubungkan Entities Employee dengan Entities Customer yaitu atribut employeeNumber : INTEGER(11) yang berada pada Entities Employee, sehingga atribut employeeNumber : INTEGER(11) menjadi Foreign Key pada Entities Customer.

* 1. Offices memiliki Employee Relasi : One – to – Many [1 – n]

Atribut yang menghubungkan Entities Offices dengan Entities Employee yaitu atribut officesCode : VARCHAR(10) yang berada pada Entities Offices, sehingga atribut officesCode : VARCHAR(10) menjadi Foreign Key pada Entities Employee.

* 1. Orders memiliki OrderDetails Relasi : One – to – Many [1 – n]

Atribut yang menghubungkan Entities Orders dengan Entities OrdersDetails yaitu atribut orderNumber : INTEGER(11) yang berada pada Entities Orders, sehingga atribut orderNumber : INTEGER(11) menjadi Foreign Key pada Entities OrdersDetails.

* 1. Product memiliki OrderDetails Relasi : One – to – Many [1 – n]

Atribut yang menghubungkan Entities Product dengan Entities OrdersDetails yaitu atribut productCode : VARCHAR(15) yang berada pada Entities Product, sehingga atribut productCode : VARCHAR(15) menjadi Foreign Key pada Entities OrderDetails.

* 1. ProductLines memiliki Products Relasi : One – to – Many [1 – n]

Atribut yang menghubungkan Entities ProductLines dengan Entities Product yaitu atribut productLine : VARCHAR(50) yang berada pada Entities ProductLines, sehingga atribut customerNumber : INTEGER(11) menjadi Foreign Key pada Entities Product.