**NAMA : FITRIA WIDYA ASTARI**

**NIM : L200150027**

**KELAS : A**

Note: Semua pertanyaan terkait dengan database ‘CLASSICMODELS’ seperti yang sudah disepakati di awal pertemuan kuliah.

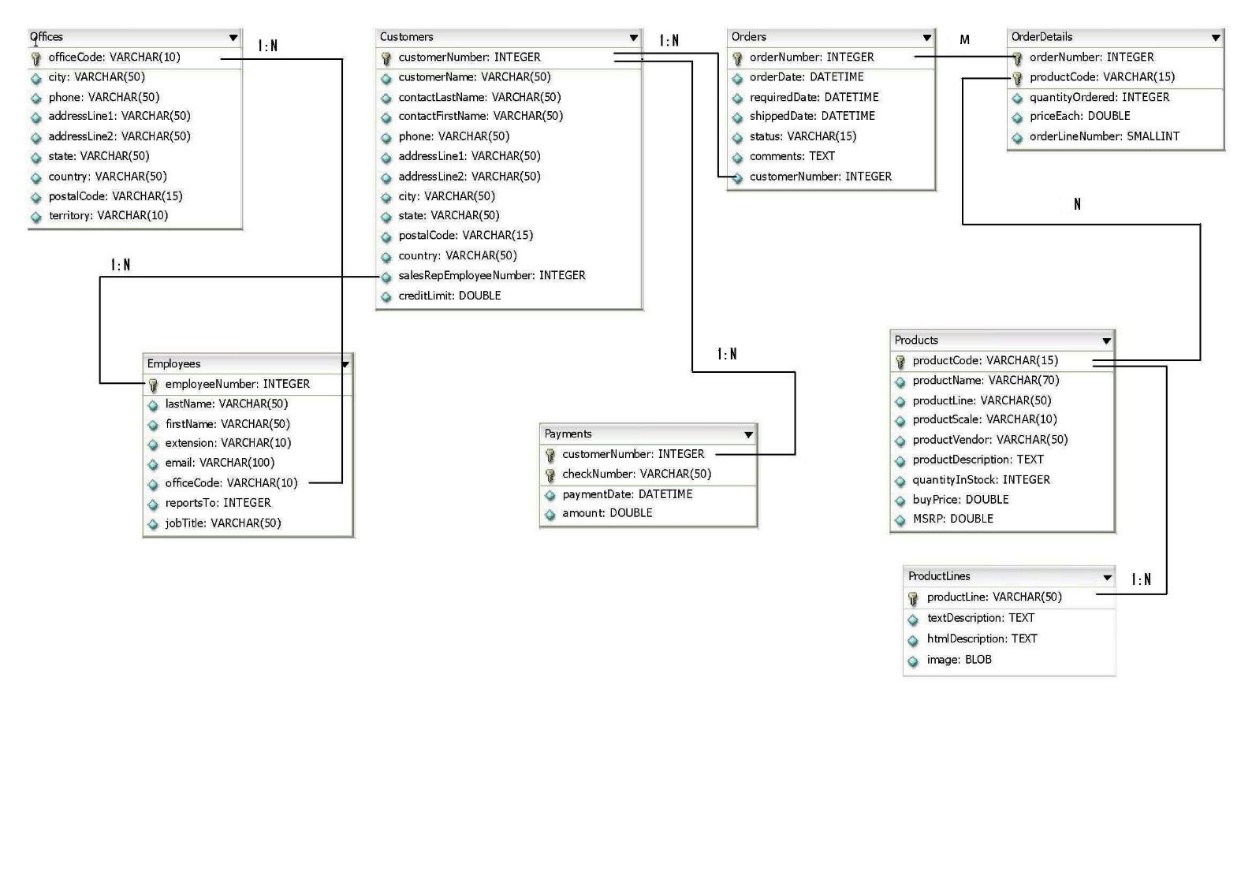
1. Lihat gambar-1 di halaman-2, gambar tersebut menampilkan semua tabel yang terdapat dalam database ‘classicmodels’. (a) Buatlah semua relasi antar tabel, dan terangkan jenis relasinya (20%). (b) Jelaskan fungsi masing-masing tabel (digunakan untuk menyimpan data apa?) (10%).
2. Buatlah query untuk menampilkan daftar barang yang terjual pada tanggal tertentu! Kolom output minimal terdiri atas kode, nama barang, dan jumlah barang yang terjual. (20%)
3. Sama dengan soal kedua tetapi dibuat dalam bentuk store procedure dengan tanggal sebagai input, sebagai output adalah nilai uang yang diperoleh di tanggal tersebut, dan table daftar barang. (30%)
4. Buatlah store procedure untuk menampilkan daftar customer yang melakukan pembelian (order) selama satu minggu terakhir, sebagai output adalah jumlah pelanggan dan daftar pelanggan. (25%)



Gambar-1.RancanganTabel dalam database Classicmodels

**Jawab :**

1. **(a)**



Jenis relasi pada table classicmodels ada 3 yaitu one-to-one(1:1), one-to-many(1:n), dan many-to-many(m:n).

1. Offices - Employees

Diibaratkan seperti satu kantor memiliki(*have*) banyak pegawai. Sesuai dengan atribut berupa primary key pada Offices berada pada entitas Employees. Yang menunjukkan bahwa Employees merupakan *many*(**n**) dan Offices merupakan one(**1**). Sehingga relasi antara Offices dan Employees adalah 1:n (one-to-many).

1. Employees - Customers

Diibaratkan seperti satu pekerja melayani(*service*) banyak pelanggan. Sesuai dengan atribut berupa primary key pada Employees berada pada entitas Customers. Yang menunjukkan bahwa Customers merupakan *many*(**n**) dan Employees merupakan one(**1**). Sehingga relasi antara Employees dan Customers adalah 1:n (one-to-many).

1. Payments - Customers

Diibaratkan seperti banyak pembayaran dilakukan/dibuat (*made*) oleh satu pelanggan. Sesuai dengan atribut berupa primary key pada Customers berada pada entitas Payments. Yang menunjukkan bahwa Payments merupakan *many*(**n**) dan Customers merupakan one(**1**). Sehingga relasi antara Payments dan Customers adalah n:1 (many-to-one) atau (one-to-many).

1. Customers - Orders

Diibaratkan seperti satu Customers melakukan/menempatkan (*places*) banyak Orders. Sesuai dengan atribut berupa primary key pada Customers berada pada entitas Orders. Yang menunjukkan bahwa Orders merupakan *many*(**n**) dan Customers merupakan one(**1**). Sehingga relasi antara Customers dan Orders adalah 1:n (one-to-many).

1. Orders - Products

Diibaratkan seperti banyak Orders memiliki (*have*) banyak Products. Sesuai dengan atribut berupa primary key pada Orders dan Product berada pada table OrderDetails. Yang menunjukkan bahwa OrderDetails merupakan *table* baru yang terbentuk dari relasi pada Products dan Orders. Sehingga relasi antara Product dan Orders adalah m:n (many-to-many).

1. ProductLines – Products

Diibaratkan seperti satu ProductLines menunjukkan (*belongs to*) banyak Products. Sesuai dengan atribut berupa primary key pada ProductLines berada pada entitas Products. Yang menunjukkan bahwa Products merupakan *many*(**n**) dan ProductLines merupakan one(**1**). Sehingga relasi antara ProductLines dan Products adalah 1:n (one-to-many).

1. Orders - orderDetails

Diibaratkan seperti satu order hanya memiliki satu orderDetails. Yang menunjukkan bahwa Orders merupakan one(**1**) dan orderDetails merupakan one(**1**). Sehingga relasi antara Orders dan orderDetails adalah 1:1 (one-to-one)

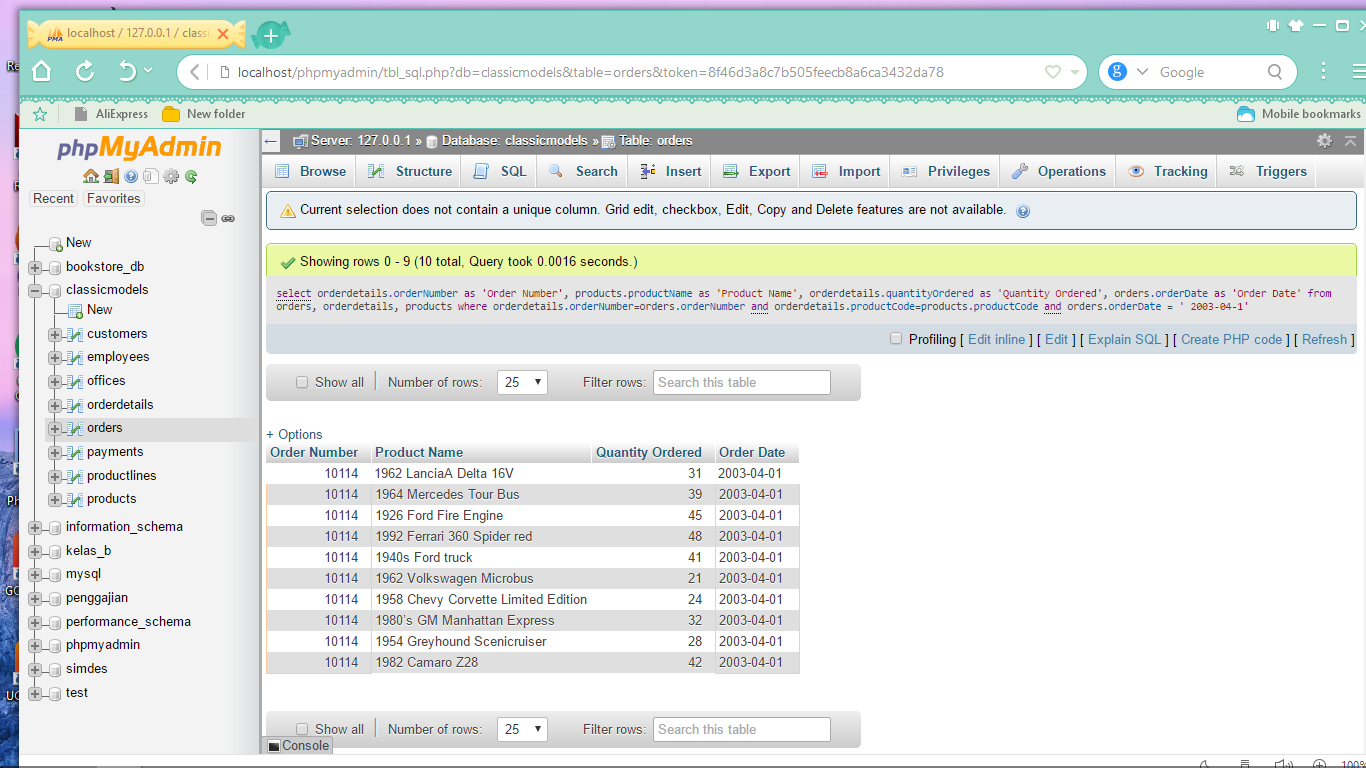
1. orderDetails - Products

Diibaratkan seperti satu Products memiliki banyak orderDeails. Sesuai dengan atribut berupa primary key pada Products berada pada entitas orderDetails. Yang menunjukkan bahwa orderDetails merupakan *many*(**n**) dan Products merupakan one(**1**). Sehingga relasi antara orderDetails dan Products adalah n:1 (many-to-one) atau (one-to-many).

**(b)**

|  |  |
| --- | --- |
| **Table** | **Fungsi** |
| Payments | Menyimpan data pembayaran product yang di beli oleh customers |
| Customers | Menyimpan data customers yang membeli product yang dijual di perusahaan tersebut |
| Offices | Menyimpan data cabang kantor perusahaan tersebut |
| Employees | Menyimpan data pegawai yang bekerja di perusahaan tersebut |
| Orders | Menyimpan data barang yang di pesan |
| orderDetails | Menyimpan data detail dari order berupa jumlah order dan juga harga satuan barang yang di-*order* yang merupakan table hasil relasi orders dan products |
| Products | Menyimpan data produk yang di jual beserta harga dan jumlah stock yang tersedia |
| productLines | Menyimpan data text dan html description dan juga image yang mana productLines memiliki relasi dengan table products |

select orderdetails.orderNumber as 'Order Number', products.productName as 'Product Name', orderdetails.quantityOrdered as 'Quantity Ordered', orders.orderDate as 'Order Date' from orders, orderdetails, products where orderdetails.orderNumber=orders.orderNumber and orderdetails.productCode=products.productCode and orders.orderDate = ' 2003-04-01';



DELIMITER $$

CREATE PROCEDURE coba(IN tanggal varchar(10), OUT total int)

BEGIN

SELECT SUM(orderdetails.quantityOrdered\*orderdetails.priceEach) INTO total

FROM orders, orderdetails, products

WHERE orders.orderNumber=orderdetails.orderNumber AND products.productCode = orderdetails.productCode AND orders.orderDate = tanggal ;

SELECT products.productCode, products.productName, orderdetails.quantityOrdered

FROM orders, orderdetails, products

WHERE orders.orderNumber=orderdetails.orderNumber AND products.productCode = orderdetails.productCode AND orders.orderDate = tanggal GROUP BY products.productCode;

END $$

DELIMITER;

DELIMITER $$

CREATE PROCEDURE cobaCustomer(IN bulan int, IN tahun int, OUT jumlah int)

BEGIN

SELECT COUNT(distict customerNumber) into jumlah

FROM orders

WHERE month(orderdate)=bulan AND

Year(orderdate)=tahun AND

Day(orderdate) > 24;

SELECT customers.customerNumber, customers.customerName

FROM customers, orders

WHERE customers.customerNumber = orders.customerNumber AND

Month(orders.orderDate) = bulan AND

Year(orders.orderDate) = tahun AND

Day(orders.orderDate) > 24

GROUP by customers.customerNumber;

END $$

DELIMITER ;