|  |  |
| --- | --- |
| **LOGO_KARTU_KECIL** | Universitas Muhammadiyah Surakarta  Jl. A. Yani Tromol Pos I Pabelan Kartasura Telp (0271)717417, 719483 Fax. (0271)715448 Surakarta 57102 |

**UJIAN TENGAH SEMESTER GANJIL 2017/2018**

***MID SEMESTER EXAM OF ODD SEMESTER***

|  |  |  |  |
| --- | --- | --- | --- |
| **FAKULTAS/FACULTY : KOMUNIKASI DAN INFORMATIKA/COMMUNICATION AND INFORMATICS** | | | |
| **JURUSAN/DEPARTMENT : INFORMATIKA/INFORMATICS** | | | |
| Mata Uji – *Course* | Sistem Management Basisdata | Hari / Tanggal – *Day/Date* |  |
| Smt./Klas - *Class* | 5/A, B | Jam ke - *Session* |  |
| Penguji - *Examiner* | Dr. Ir. Bana Handaga, MT | Waktu - *Duration* | 90 menit |

Open Cheatsheet

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 tabel daftar barang. (30%)
4. Buatlah store procedure untuk menampilkan daftar customer yang melakukan pembelian (order) selama satu minggu terakhir, sebagai output adalam jumlah pelanggan dan daftar pelanggan. (25%)



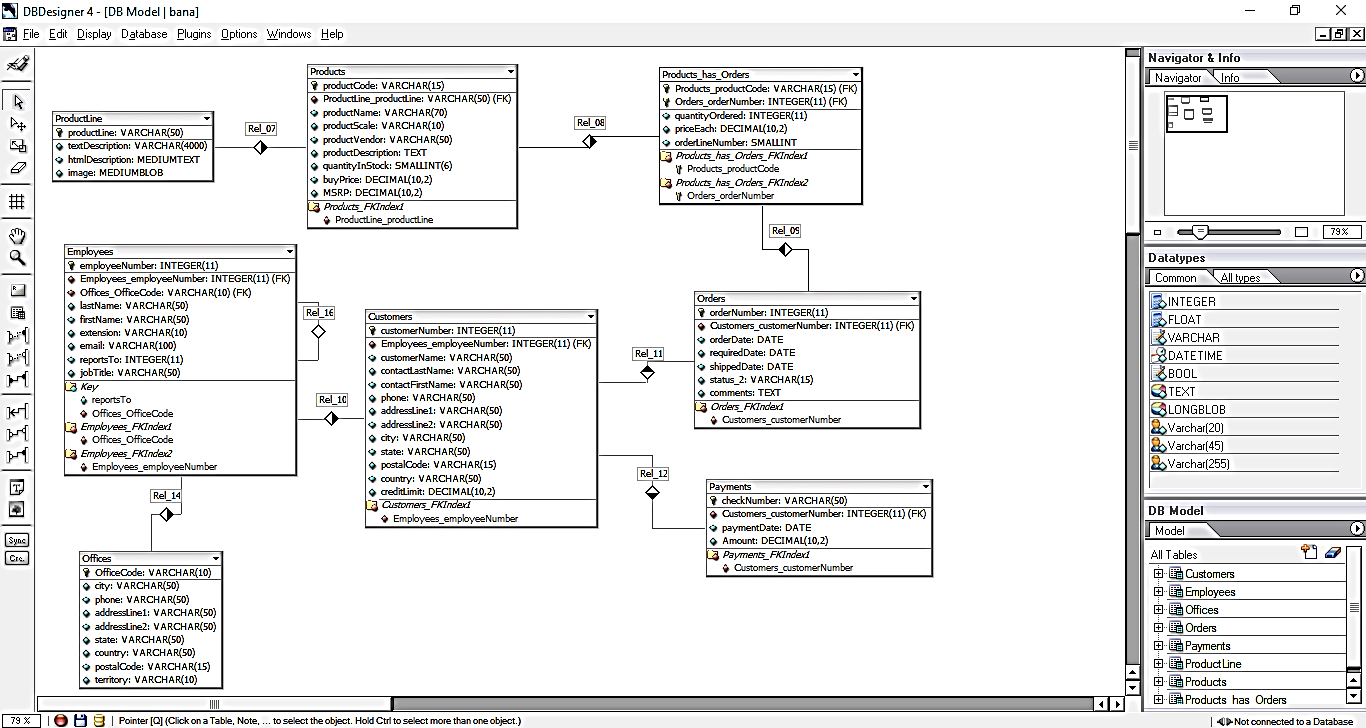
Gambar-1. Rancangan Tabel dalam database Classicmodels

Nama : Wahyu Tri Saputro

NIM : L200140149

Kelas : A

JAWAB:

1. a
2. Menentukan hubungan
3. Customer membayar Payments

Hubungan : One to Many

* Atribut penghubung : CustomerNumber (Integer (11) FK) berada di table Payments.

1. Customer membeli / memesan Orders

Hubungan: One to Many

* Atribut penghubung : CustomerNumber (Integer (11) FK) berada di table Orders.

1. Employees melayani Customers

Hubungan: One to Many

* Atribut penghubung : SalesRepEmployeeNumber (Integer (11) FK) yang berada di table Customers.

1. Employees mengamati / memimpin Employees

Hubungan: One to Many

* Atribut penghubung : ReportsTo (Integer (11) FK) yang berada di table Employees itu sendiri.

1. Employees menempati Offices

Hubungan: Many to One

* Atribut penghubung : OfficeCode (Varchar (10) FK) yang berada di table Offices.

1. Orders memiliki Order Details

Hubungan: Many to many

* Atribut penghubung: Productcode (Varchar (15) FK), OrderNumber (Integer (11) FK) yang berada di table Order Details.

1. Product memiliki Order Details

Hubungan: Many to many

* Atribut penghubung: Productcode (Varchar (15) FK), OrderNumber (Integer (11) FK) yang berada di table Order Details.

1. Productlines mengklasifikasi Product

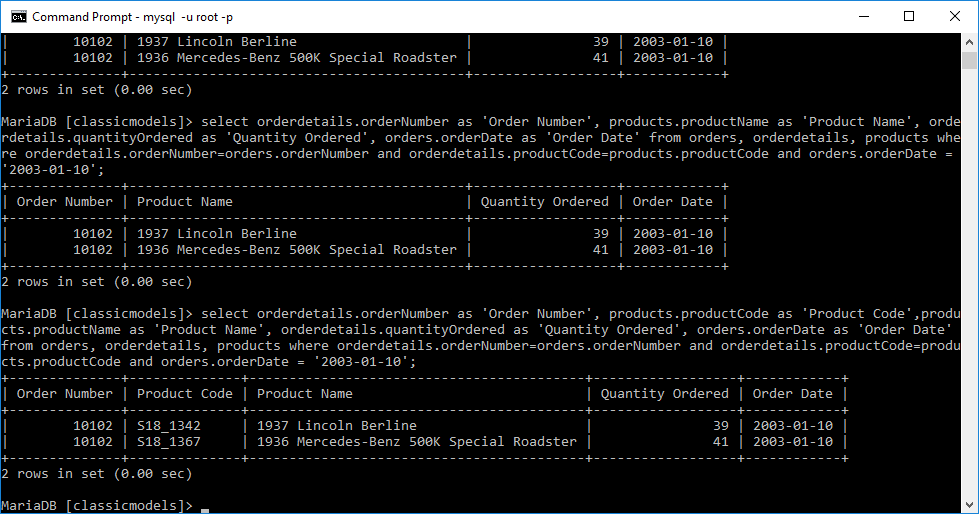
Hubungan: One to many

* Atribut penghubung : Productline (Varchar (50) FK) yang berada di table Product

b.

|  |  |
| --- | --- |
| Table | Fungsi |
| Payments | Mengetahui data pembayaran product yang di beli oleh customers |
| Customers | Mengetahui data customers yang membeli product yang dijual di perusahaan tersebut |
| Orders | Mengetahui data barang yang di pesan. |
| Employees | Mengetahui data pegawai yang bekerja di tempat tersebut |
| Office | Mengetahui data cabang kantor perusahaan tersebut |
| Products | Mengetauhi data product yang di jual beserta harga dan jumlah stock yang tersedia |
| Productlines | Merupakan table relasi dari Products dan Orders yang berfungsi untuk mengetahui jenis product yang dijual |

1. 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 = '2003-01-06';



1. DELIMITER //

CREATE PROCEDURE cekk(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 ;

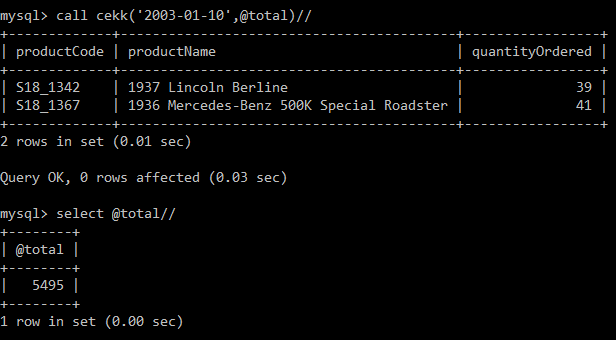
SELECT products.productCode, products.productName, orderdetails.quantityOrdered

FROM orders, orderdetails, products

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

END//

DELIMITER;



1. DELIMITER //

CREATE PROCEDURE cek(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 ;