**IF2240 Basis Data**

**Semester II 2020/2021**

Praktikum II : Advanced SQL

Dipersiapkan oleh:

Asisten Laboratorium Basis Data

Pelaksanaan:

Waktu Mulai: Jumat, 19 Februari 2021 – 16.00 WIB

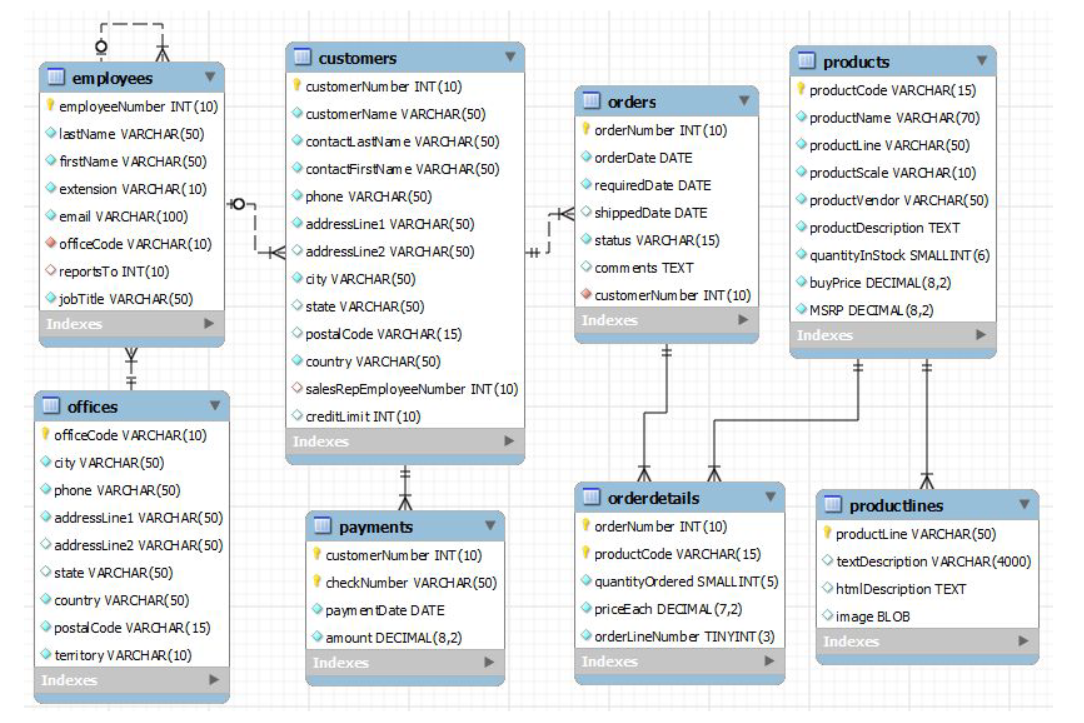
Waktu Selesai: Jumat, 19 Februari 2021 – 18.00 WIB

**I. Deskripsi Sistem**

Sebuah perusahaan mobil melakukan pencatatan terhadap proses bisnis perusahaannya. Perusahaan ini memiliki beberapa kantor cabang. Setiap pegawai akan ditempatkan di salah satu kantor cabang. Seluruh data produk yang dijual oleh perusahaan dicatat. Perusahaan ini juga mencatat seluruh pembeli beserta data lengkapnya. Seorang pembeli dapat dilayani oleh seorang pegawai. Perusahaan mencatat seluruh transaksi yang dilakukan oleh pembeli.

**II. Skema Basis Data**

Berikut skema relasional basis data berdasarkan deskripsi sistem yang telah dijabarkan pada bagian sebelumnya.



|  |
| --- |
| customers = (customerNumber, customerName, contactLastName, contactFirstName, phone, addressLine1, addressLine2, city, state, postalCode, country, salesRepEmployeeNumber, creditLimit)  employees = (employeeNumber, lastName, firstName, extension, email, officeCode, reportsTo, jobTitle)  offices = (officeCode, city, phone, addressLine1, addressLine2, state, country, postalCode, territory)  Payments = (customerNumber, checkNumber, paymentDate, amount)  orders = (orderNumber, orderDate, requiredDate, shippedDate, status, comments, customerNumber)  orderdetails = (orderNumber, productCode, quantityOrdered, priceEach, orderLineNumber) products =(productCode, productName, productLine, productScale, productVendor, productDescription, quantityInStock, buyPrice, MSRP)  productlines = (productLine, textDescription, htmlDescription, image) |

**III. Permasalahan**

Buatlah sebuah basis data lalu masukkan (*import*) script SQL Praktikum2.sql yang telah diberikan bersama deskripsi tugas ini.

**IV. Soal**

Lakukan hal-hal berikut.

1. Buatlah query untuk menampilkan nama, negara, dan limit kredit dari 10 customer yang melakukan pembayaran produknya di atas harga pembayaran rata-rata. Urutkan dari customer yang paling banyak melakukan pembayaran produk

Query

|  |
| --- |
| WITH customer(customerName, country, creditLimit,checkNumber) as (SELECT customerName, country, creditLimit,checkNumber FROM customers NATURAL JOIN payments WHERE amount > (SELECT avg(amount) FROM payments))  SELECT customerName, country,creditLimit FROM customer GROUP BY customerName ORDER BY count(checkNumber) desc LIMIT 10; |

Hasil screenshot

|  |
| --- |
|  |

1. Buatlah query untuk menampilkan nama, jumlah penjualan (order), dan total keuntungan penjualan (profit) produk yang memiliki harga jual paling rendah   
   (HINT: profit dihitung berdasarkan total selisih harga jual dengan harga beli)

Query

|  |
| --- |
| SELECT productName, count(orderNumber) as jumlahJual, (priceEach-buyPrice)\*count(orderNumber) as profit FROM products NATURAL JOIN orderdetails WHERE priceEach = (SELECT min(priceEach) FROM orderdetails) GROUP BY productCode ; |

Hasil screenshot

|  |
| --- |
|  |

1. Buatlah query untuk menampilkan nama produk yang pernah dipesan dari kantor di Paris, tetapi TIDAK pernah dipesan dari kantor di NYC (Tampilkan juga nama kotanya dalam tabel), beserta jumlah barang yang telah dipesan tersebut

Query

|  |
| --- |
| WITH code(productCode) as(  (SELECT productCode FROM  offices INNER JOIN employees USING (officeCode)  INNER JOIN customers ON salesRepEmployeeNumber = employeeNumber  INNER JOIN orders USING (customerNumber)  INNER JOIN orderdetails USING (orderNumber)  INNER JOIN products USING(productCode)  WHERE offices.city = "Paris")  EXCEPT  (SELECT productCode FROM  offices INNER JOIN employees USING (officeCode)  INNER JOIN customers ON salesRepEmployeeNumber = employeeNumber  INNER JOIN orders USING (customerNumber)  INNER JOIN orderdetails USING (orderNumber)  INNER JOIN products USING(productCode)  WHERE offices.city = "NYC"))  SELECT productName,sum(quantityOrdered),offices.city FROM  offices INNER JOIN employees USING (officeCode)  INNER JOIN customers ON salesRepEmployeeNumber = employeeNumber  INNER JOIN orders USING (customerNumber)  INNER JOIN orderdetails USING (orderNumber)  INNER JOIN products USING(productCode) WHERE productCode in (SELECT productCode from code) GROUP BY productCode; |

Hasil screenshot

|  |
| --- |
|  |

1. Buatlah view bernama problematics untuk pesanan yang memiliki comment dan tidak berstatus "Shipped" maupun “On hold”. Dalam view tersebut, sertakan tanggal dan status pesanan, nama dan nomor telepon pelanggan, nama lengkap dan email sales rep untuk pelanggan, dan comment untuk pesanan tersebut. Setelah itu, tampilkan 5 baris pertama pada view.

Query

|  |
| --- |
| create view problematics as  SELECT orderDate,status,customerName,phone,CONCAT(firstName, " ", lastName) as fullname,email,comments  FROM employees NATURAL JOIN customers NATURAL JOIN orders WHERE comments IS NOT NULL and (status <> "Shipped" and status <>"On hold");  SELECT \* from problematics LIMIT 5; |

Hasil screenshot

|  |
| --- |
|  |

1. Buatlah query untuk membuat view dengan nama ‘planes\_customer\_spain’ yang menampilkan nama pelanggan (sebagai ‘nombre’) yang berasal dari Spanyol dan pernah memesan produk dengan kategori ‘Planes’. Tampilkan juga kota asalnya (city) sebagai ‘ciudad’ dan jumlah uang yang pelanggan tersebut habiskan untuk memesan produk dengan kategori ‘Planes’ tersebut sebagai ‘suma’!

Query

|  |
| --- |
| create view planes\_customer\_spain as  WITH buyPlane(orderNumber) as (SELECT DISTINCT orderNumber FROM orders NATURAL JOIN orderdetails NATURAL JOIN products WHERE productLine = "Planes")  SELECT customerName as nombre, city as ciudad, priceEach\*quantityOrdered as suma, productLine FROM customers NATURAL JOIN orders NATURAL JOIN orderdetails NATURAL JOIN products WHERE orderNumber in(SELECT orderNumber from buyPlane) and country="spanyol"; |

Hasil screenshot

|  |
| --- |
| Gatau kenapa kok error hehe |

1. Sebuah order dapat dilakukan untuk membeli banyak jenis product berbeda sekaligus.

Buatlah query untuk menghapus semua payments oleh customer yang pernah membeli 100 jenis product berbeda.

Query

|  |
| --- |
| WITH cust100(customerNumber) as (SELECT customerNumber FROM customers NATURAL JOIN orders NATURAL JOIN orderdetails GROUP BY customerNumber HAVING count(productCode) >100)  DELETE from payments WHERE customerNumber in (SELECT customerNumber FROM cust100); |

Hasil screenshot

|  |
| --- |
| Harusnya bener gatau kenapa gabisa di run |

1. Buatlah query untuk menghapus product yang memiliki kategori product line “Classic Cars” dan product memiliki tahun keluaran 1960an dari orders yang memiliki status “On Hold”. Tunjukkan bahwa tidak ada lagi 1960s classic cars pada order berstatus on hold.

(HINT : tahun keluaran dapat dilihat pada nama product)

Query

|  |
| --- |
|  |

Hasil screenshot

|  |
| --- |
| Gasempettt ☺ |

1. Buatlah query untuk menambahkan record pegawai baru dengan nama depan ‘Forsythe’ yang merupakan anak dari pegawai lama dengan nomor pegawai ‘1504’. Forsythe akan ditempatkan di office di kota yang sama dengan ayahnya dengan jabatan ‘Sales Rep’, dan akan melapor kepada ayahnya sendiri. Nomor pegawainya adalah 1999 dan memiliki extension x100. Email diisi bebas. (HINT: Nama belakang ayah diturunkan menjadi nama belakang anak)

Query

|  |
| --- |
| insert into employees values (1999,"Jones","Forsythe","x100","haha@gmail.com","7","1504","Sales Rep"); |

Hasil screenshot

|  |
| --- |
|  |

1. Tambahkan satu product line dengan nama “Sports car” dengan deskripsi “The fastest car in the world”, kemudian tambahkan satu produk yang mengacu pada product line tersebut, dengan kode S100\_01, nama bugatti 2021, product scale 1:20, deskripsi “Must buy”, product vendor Super Sport, berjumlah 500, dengan harga 50, dan MSRP 50. Buktikan dengan menampilkan semua data product lines dan products yang baru ditambahkan.

Query

|  |
| --- |
| insert into productLines values("Sports car","The fastest car in the world",NULL,NULL);  insert into products values ("S100\_01","bugatti 2021","Sports car","1:20","Super Sport","Must buy",500,50,50); |

Hasil screenshot

|  |
| --- |
|  |

1. Vendor Unimax Art Galleries menaikkan harga beli (buyPrice) semua produknya sebanyak 10% dan vendor Red Start Diecast menurunkan harga beli semua produknya sebanyak 5%. Aplikasikan perubahan harga beli ini ke basis data. Buktikan perubahan dengan menampilkan nama produk, vendor, dan harga beli yang baru.

**Constraint**: hanya boleh menggunakan satu query update.

Query

|  |
| --- |
| update products  set buyPrice = case  when productVendor="Unimax Art Galleries" then buyPrice\*1.1  when productVendor="Red Start Diecast" then buyPrice\*0.95  else buyPrice  end; |

Hasil screenshot

|  |
| --- |
|  |

1. Buatlah query untuk melakukan update nomor telepon pada tabel offices, dimana untuk offices yang tidak berada di USA, nomor telepon disamakan menjadi nomor telepon office di London.

Query

|  |
| --- |
| update offices  set phone = case  when country <>"USA" then "+44 20 7877 2041"  else phone  end; |

Hasil screenshot

|  |
| --- |
|  |

1. Buatlah tabel baru bernama employeessalary yang berisi employeeNumber yang mereferensi kepada employees(employeeNumber) dan salary bertipe integer. Kedua atribut tidak boleh bernilai null. Apabila suatu record A pada tabel employees dihapus/diupdate, maka record B pada employeessalary yang berisi foreign key record A juga ikut terhapus/terupdate. Buktikan menggunakan describe.

Query

|  |
| --- |
| create table employeesalary (  employeeNumber int NOT NULL,  int salary NOT NULL,  foreign (employeNumber) references employees(employeeNumber) on delete cascade;  describe employeesalary; |

Hasil screenshot

|  |
| --- |
| Error kak maaf |