CS 3100

Homework 7

General Directions:

Please answer each question in a Word document as follows: 1) Copy your SQL query and 2) Attach a screen shot of the output. Please submit both the Word document and SQL script for grading. Each question is worth 20 points.

1. Assuming a sales commission of 5% on every order, calculate the sales commission due for all employees. List the employee name and the sales commission each person is due.

select

concat(e.firstName, e.lastName) as 'Employee Name',

concat('$', format(sum(od.quantityOrdered\*od.priceEach)\*0.05, 2)) as SalesCommission

from employees e

inner join customers c

on e.employeeNumber = c.salesRepEmployeeNumber

inner join orders o

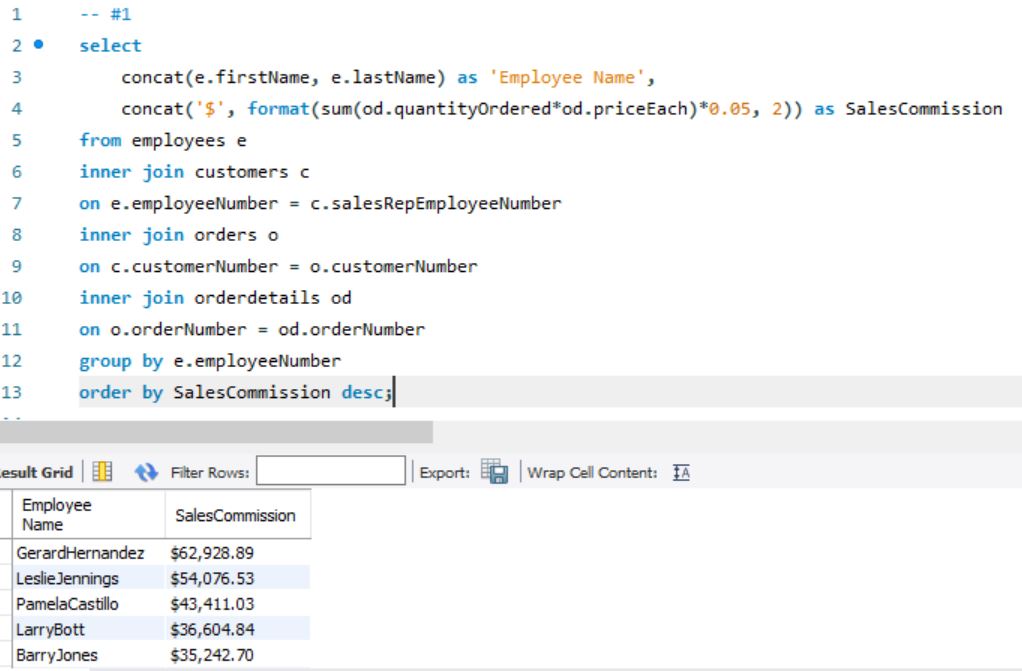
on c.customerNumber = o.customerNumber

inner join orderdetails od

on o.orderNumber = od.orderNumber

group by e.employeeNumber

order by SalesCommission desc;



1. Create a list of customers and the amount they currently owe us. List the customer name and the amount due. Create views to track the total amount ordered and the total amount paid. Use these views to create your final query. Important – do not format interim numeric results. If you need to round numbers use the round function. Don’t format your numbers until your final query. Having imbedded commas in numeric fields can cause math problems.

create view OrderAmount as

select

o.customerNumber,

od.orderNumber,

sum(od.quantityOrdered \* od.priceEach) as 'Amount'

from orderdetails od

left join orders o

on od.orderNumber = o.orderNumber

group by o.customerNumber

order by o.customerNumber;

create view PaymentAmount as

select

p.customerNumber,

c.customerName,

p.amount

from payments p

left join customers c

on p.customerNumber = c.customerNumber

group by p.customerNumber;

select

o.customerNumber,

p.customerName,

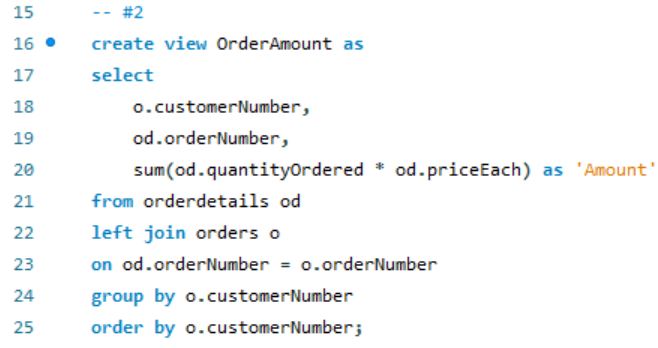
concat('$', format(sum(o.Amount - p.amount), 2)) as 'AmountDue'

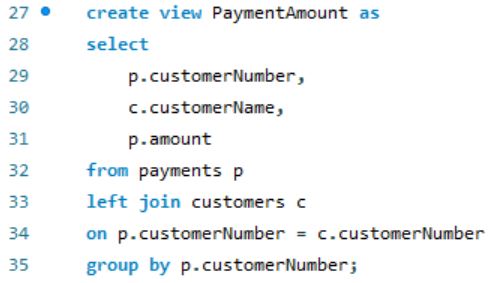
from OrderAmount o

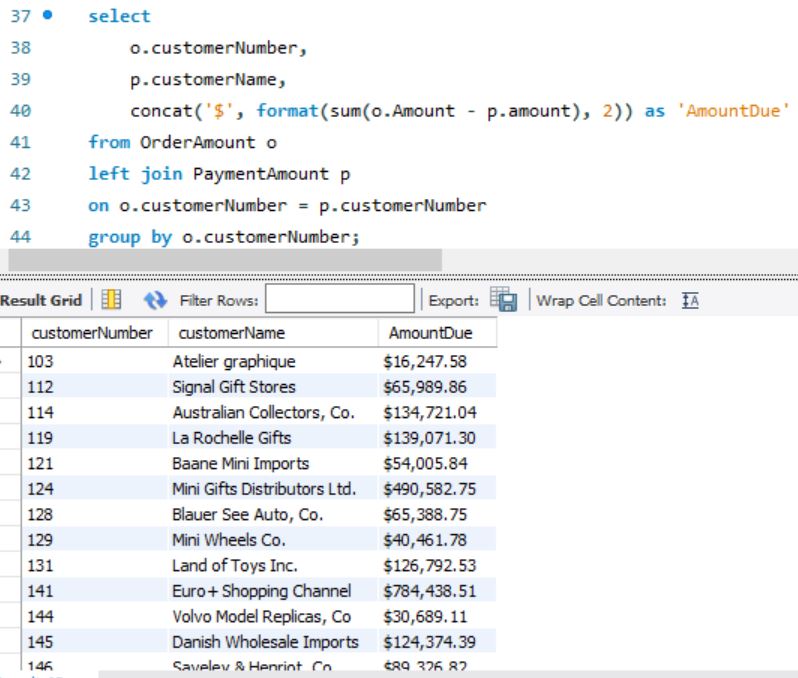
left join PaymentAmount p

on o.customerNumber = p.customerNumber

group by o.customerNumber;







1. Using the classicmodels schema, find any product that has never been purchased. Create a new table that show the productCode, productName, productLine, and buyPrice for any product that has never been purchased. Update the table by reducing the buyPrice by 10%.

create table NeverPurchased (

ProductCode varchar(15) not null,

ProductName varchar(70),

ProductLine varchar(50),

BuyPrice decimal(10,2)

);

insert into NeverPurchased

(ProductCode, ProductName, ProductLine, BuyPrice)

select

p.ProductCode,

p.ProductName,

p.ProductLine,

p.BuyPrice

from products p

where not exists

(select null

from orderdetails od

where od.productCode = p.productCode);

select \*

from NeverPurchased;

set SQL\_SAFE\_UPDATES = 0;

update NeverPurchased

set buyPrice = buyPrice\*0.9;

set SQL\_SAFE\_UPDATES = 1;

