Intermediate Problems

20. Categories, and the total products in each category

For this problem, we’d like to see the total number of products in each category. Sort the results by the total number of products, in descending order.

select

c.category\_name,

count(p.product\_id) as nb\_products

from categories c

inner join products p on c.category\_id = p.category\_id

GROUP BY c.category\_name

ORDER BY nb\_products DESC;

21. Total customers per country/city

In the Customers table, show the total number of customers per Country and City.

select

country,

city,

count(city) AS total\_customer

from customers

group by country, city

order by total\_customer;

22. Products that need reordering

What products do we have in our inventory that should be reordered? For now, just use the fields Units In Stock and Reorder Level, where Units In Stock is less than the Reorder Level, ignoring the fields Units On Order and Discontinued. Order the results by Product ID.

select

product\_id,

product\_name,

units\_in\_stock,

reorder\_level

from products

where units\_in\_stock < reorder\_level

order by product\_id;

23. Products that need reordering, continued

Now we need to incorporate these fields—Units In Stock, Units On Order, Reorder Level, Discontinued—into our calculation. We’ll define “products that need reordering” with the following: Units In Stock plus Units On Order are less than or equal to Reorder Level The Discontinued flag is false (0).

select

product\_id,

product\_name,

units\_in\_stock,

units\_on\_order,

reorder\_level

from products

where (units\_in\_stock + units\_on\_order) <= reorder\_level

and discontinued = 0

order by product\_id;

24. Customer list by region

A salesperson for North wind is going on a business trip to visit

customers, and would like to see a list of all customers, sorted by region, alphabetically. However, he wants the customers with no region (null in the Region field) to be at the end, instead of at the top, where you’d normally find

the null values. Within the same region, companies should be sorted by CustomerID.

select

c.customer\_id,

c.company\_name,

c.region

from customers c

order by c.region,c.customer\_id;

25. High freight charges

Some of the countries we ship to have very high freight charges. We'dlike to investigate some more shipping options for our customers, to be able to offer them lower freight charges. Return the three ship countries with the highest average freight overall, in descending order by average freight.

select

ship\_country,

avg(freight) as mean\_freight

from orders

group by ship\_country

order by mean\_freight DESC

limit 3 ;

26. High freight charges - 2015

We're continuing on the question above on high freight charges. Now,

instead of using *all* the orders we have, we only want to see orders from

the year 2015.

select

ship\_country,

avg(freight) as mean\_freight

from orders

where order\_date('year', 'order\_date') = 2015

group by ship\_country

order by mean\_freight DESC

limit 3 ;

27. High freight charges with between

Another (incorrect) answer to the problem above is this:

Select Top 3

Ship Country

,Average Freight = avg(freight)

From Orders

Where

Order Date between '1/1/2015' and '12/31/2015'

Group By Ship Country

Order By Average Freight desc;

Notice when you run this, it gives Sweden as the Ship Country with the

third highest freight charges. However, this is wrong - it should be France. What is the Order ID of the order that the (incorrect) answer above is missing?

select

ship\_country,

avg(freight) as mean\_freight

from orders

where Order\_Date between '1997-01-01' and '1998-01-01'

group by ship\_country

order by mean\_freight DESC

limit 3 ;

28. High freight charges - last year

We're continuing to work on high freight charges. We now want to get the three ship countries with the highest average freight charges. But instead of filtering for a particular year, we want to use the last 12 months of order data, using as the end date the last Order Date in Orders.

select

ship\_country,

avg(freight) as mean\_freight

from orders

where order\_date >= (select max(order\_date) - interval '12 month' from orders)

group by ship\_country

order by mean\_freight DESC

limit 3 ;

29. Inventory list

We're doing inventory, and need to show information like the below, for all orders. Sort by Order ID and Product ID.

select

o.employee\_id,

e.last\_name,

o.order\_id,

p.product\_name,

od.quantity

from orders o

inner join

employees e on o.employee\_id = e.employee\_id

inner join

order\_details od on od.order\_id = o.order\_id

inner join

products p on p.product\_id = od.product\_id

order by order\_id,od.product\_id;

30. Customers with no orders

There are some customers who have never actually placed an order. Show these customers.

select

c.customer\_id,

c.company\_name

from customers c

left join

orders o on c.customer\_id =o.customer\_id

where o.order\_id is null;

31. Customers with no orders for Employee ID 4

One employee (Margaret Peacock, Employee ID 4) has placed the most orders. However, there are some customers who've never placed an order with her. Show only those customers who have never placed an order with her. select

c.customer\_id,

c.company\_name

from customers c

left join

orders o on c.customer\_id =o.customer\_id and o.employee\_id =4

where o.order\_id is null;