-- Slide 31 - Homework

--Q1: Show the number of frame sales in the database. Limit to the top 10 products.

SELECT p.name, SUM(od.orderqty) AS qtyordered

FROM sales.salesorderdetail AS od

LEFT JOIN production.product AS p

USING(productid)

WHERE p.productnumber ILIKE '%fr%'

GROUP BY p.name

ORDER BY qtyordered DESC

LIMIT 10;

--Q2: Generate a list of only the mountain bike frames sold. (26 rows)

SELECT p.name, SUM(od.orderqty) AS qtyordered

FROM sales.salesorderdetail AS od

LEFT JOIN production.product AS p

USING(productid)

WHERE p.productnumber ILIKE '%fr%'

AND p.name ILIKE '%mountain%'

GROUP BY p.name

ORDER BY qtyordered DESC;

--Q3: Which mountain bike frames were not sold? (2 rows)

SELECT p.name, SUM(od.orderqty) AS qtyordered

FROM sales.salesorderdetail AS od

RIGHT JOIN production.product AS p

USING(productid)

WHERE p.productnumber ILIKE '%fr%'

AND p.name ILIKE '%mountain%'

GROUP BY p.name

HAVING SUM(od.orderqty) IS NULL

ORDER BY qtyordered DESC;

--Q4: How many distinct stores are there? (Hint: use sales.store table)

SELECT COUNT(DISTINCT s.name)

FROM sales.store AS s;

--Q5: How much money has “Frugal Bike Shop” spent in its history with Adventure Works Cycles?

--Do not include tax or freight fees. (Hint: sales.salesorderheader table) (1 row - $7,259,567.88)

SELECT CAST(SUM(oh.totaldue) AS MONEY)

FROM sales.store AS s

LEFT JOIN sales.salesorderheader AS oh

USING(salespersonid)

WHERE s.name ILIKE 'Frugal Bike Shop';

--Q6: How many distinct / unique products has Frugal Bike Shop purchased in its history with Adventure Works Cycles? (1 row - 241)

--(Hint: start with the stores table and work toward product table. Your query will take a very very long time if you go the other way around)

SELECT COUNT(DISTINCT(od.productid))

FROM sales.store s

LEFT JOIN sales.salesorderheader AS oh

ON oh.salespersonid = s.salespersonid

LEFT JOIN sales.salesorderdetail AS od

ON od.salesorderid = oh.salesorderid

WHERE s.name ILIKE 'Frugal Bike Shop';

-- Q7: Write a query that generates a list of all products (id and name) purchased

--by Frugal Bike Shop, along with the total amount sold for each product. (241 rows)

SELECT p.productid, p.name, CAST(SUM(oh.totaldue) AS MONEY) AS totalsales

FROM production.product AS p

LEFT JOIN sales.salesorderdetail AS od

ON od.productid = p.productid

LEFT JOIN sales.salesorderheader AS oh

ON oh.salesorderid = od.salesorderid

LEFT JOIN sales.store AS s

ON s.salespersonid = oh.salespersonid

WHERE s.name ILIKE 'Frugal Bike Shop'

GROUP BY p.productid, p.name

ORDER BY totalsales DESC;

--Q8: Get a list of product ids that Frugal Bike Shop hasn’t purchased. (263 rows)

-- all products available

SELECT p.productid

FROM production.product AS p

EXCEPT

-- just the products that Frugal purchased

SELECT p.productid

FROM production.product AS p

LEFT JOIN sales.salesorderdetail AS od

ON od.productid = p.productid

LEFT JOIN sales.salesorderheader AS oh

ON oh.salesorderid = od.salesorderid

LEFT JOIN sales.store AS s

ON s.salespersonid = oh.salespersonid

WHERE s.name ILIKE 'Frugal Bike Shop';

-- Q9: Write a query to show total sales for all bike shops, by shop. Do not include tax and freight. (699 rows)

SELECT s.name, CAST(SUM(oh.totaldue) AS MONEY) AS totalsales

FROM sales.store AS s

LEFT JOIN sales.salesorderheader AS oh

USING(salespersonid). #BusinessEntityID

GROUP BY s.name

ORDER BY totalsales DESC;