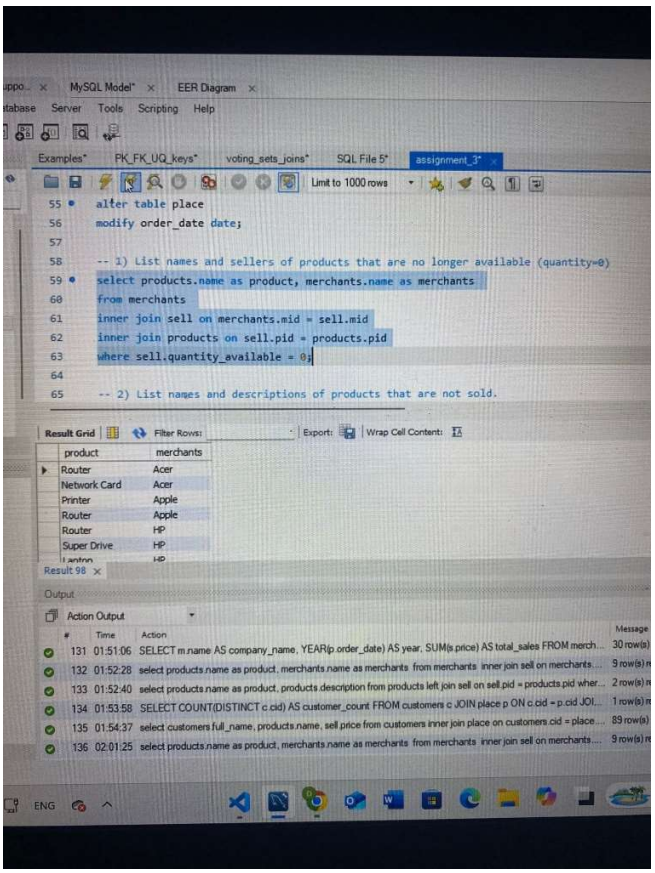
Data Base management homework 3

10.10.2024

David Schwartzman

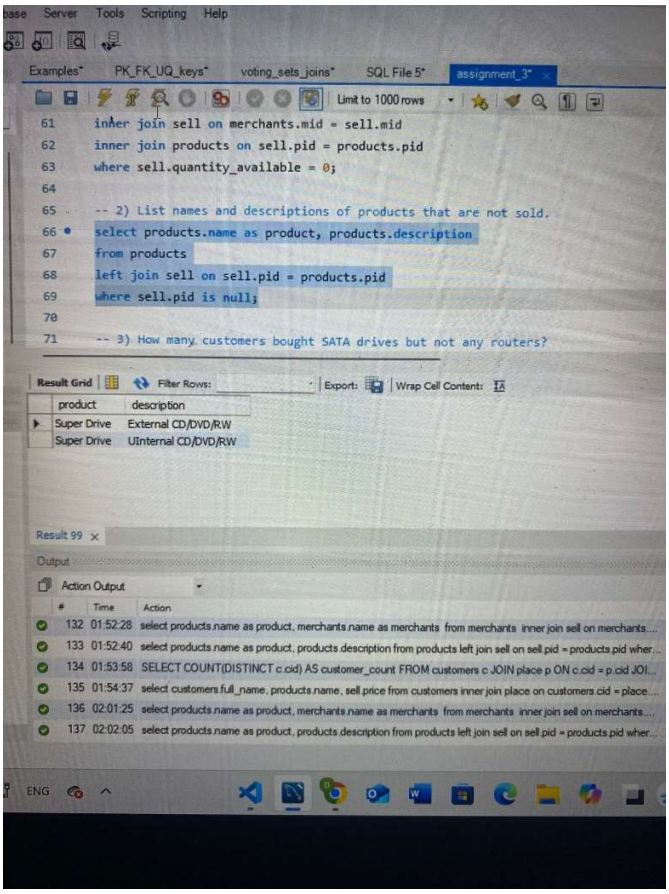
1) **List names and sellers of products that are no longer available (quantity=0)**:

This query selects product names and the corresponding merchants for products where the quantity available is zero by inner joining merchants with sell and products.



2) **List names and descriptions of products that are not sold**:

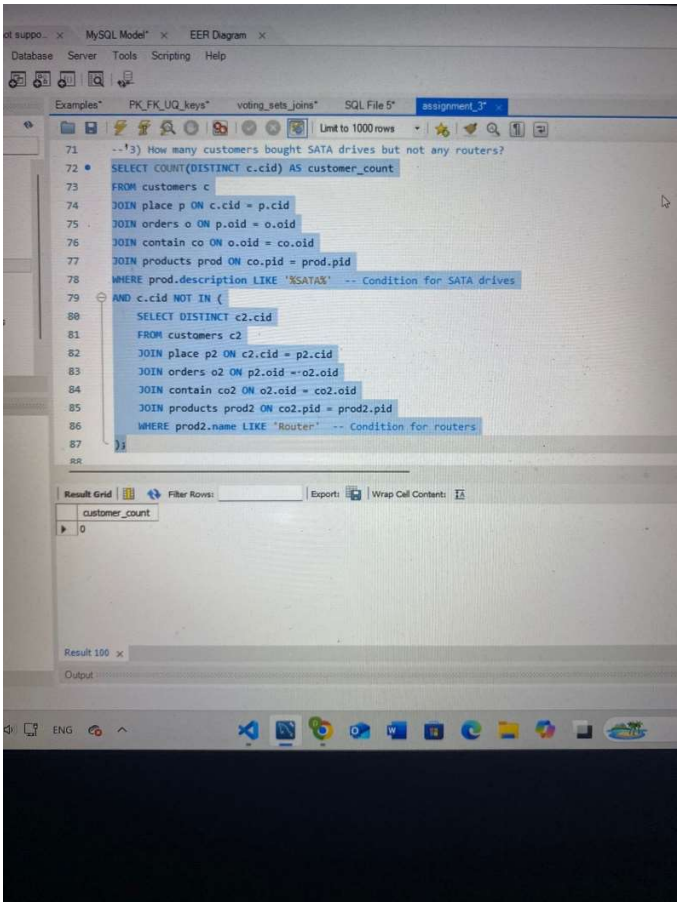
This query retrieves the names and descriptions of products that do not have any associated sales records. I am checking the items that exit but not on the sell table.



3) **How many customers bought SATA drives but not any routers?**:

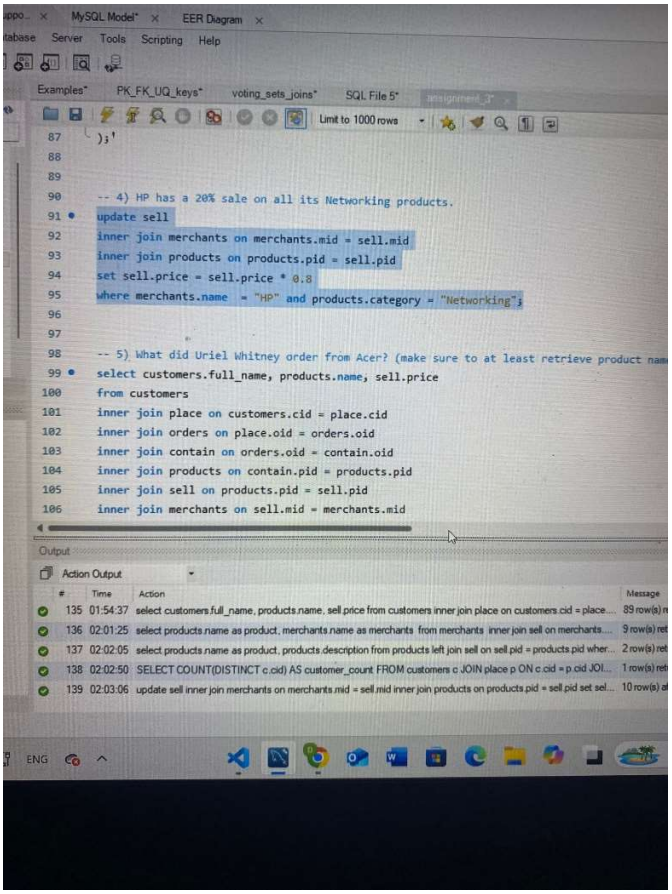
Assumptions: I assume that there is no product that it's description is "SATA" but every product that has SATA as a substring in it's description is a SATA drive.

This query counts the number of unique customers who have purchased SATA drives while ensuring they did not purchase routers. I am counting the number of customers that purchased a SATA drive bot not a router by checking who bought a SATA driver but is not part of the subquery (people who also bought a router).



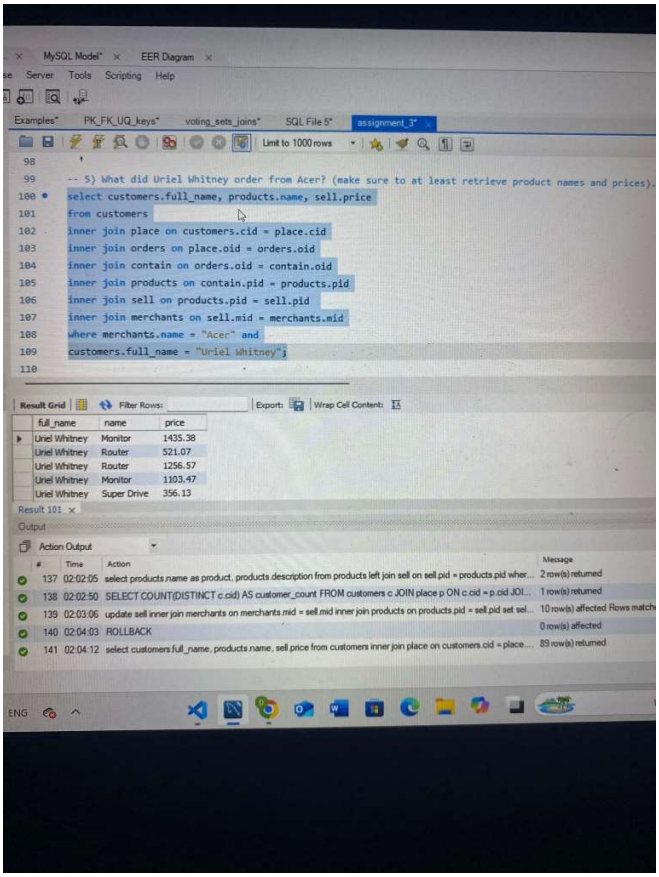
4) **HP has a 20% sale on all its Networking products**:

This query updates the prices of HP's Networking products to apply a 20% discount. It answers the question by reflecting the promotional change in pricing for specific products.



5) **What did Uriel Whitney order from Acer?**:

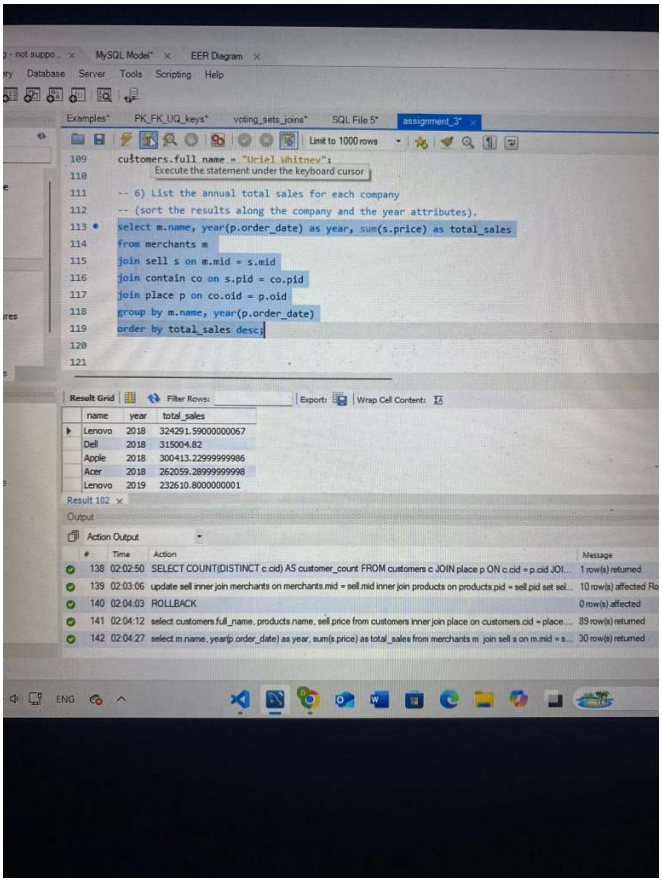
This query retrieves the names and prices of products ordered by Uriel Whitney from Acer. I am checking for all tables where the purchase made by Uriel and the merchant was Acer.



6) **List the annual total sales for each company (sort the results along the company and the year attributes)**:

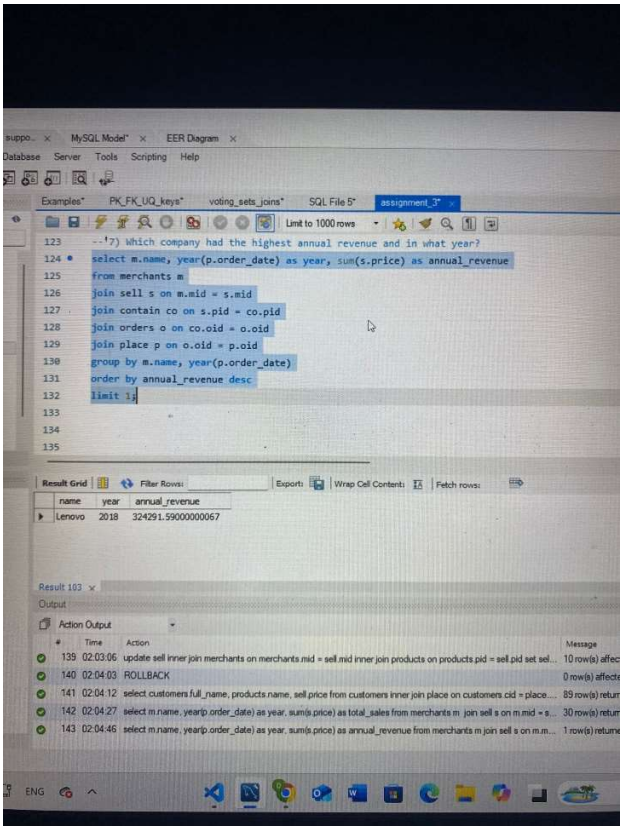
Assumptions: it doesn't matter if the output is sorted in descending orderor ascending order.

This query calculates and lists total sales for each company, grouped by year and merchant name. I then sort the list by discoing order,



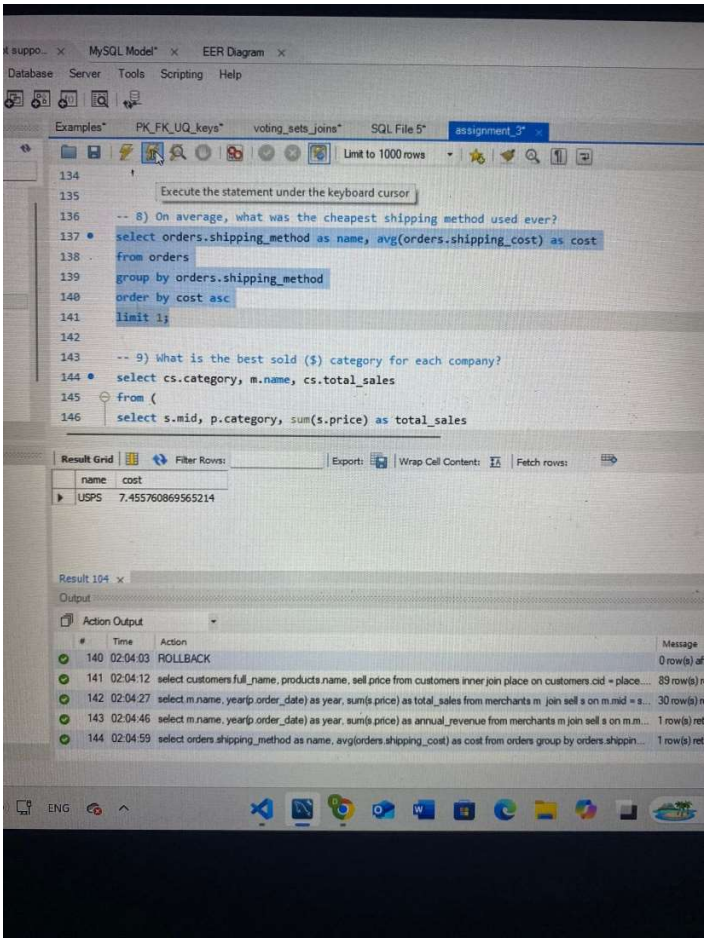
7) **Which company had the highest annual revenue and in what year?**:

This query identifies the company with the highest revenue in a particular year. I find the revenue of each company by summing up the sell price and then group it by merchant name and year. then I sort it in descending order limit 1 to find out the max that was made in a year.



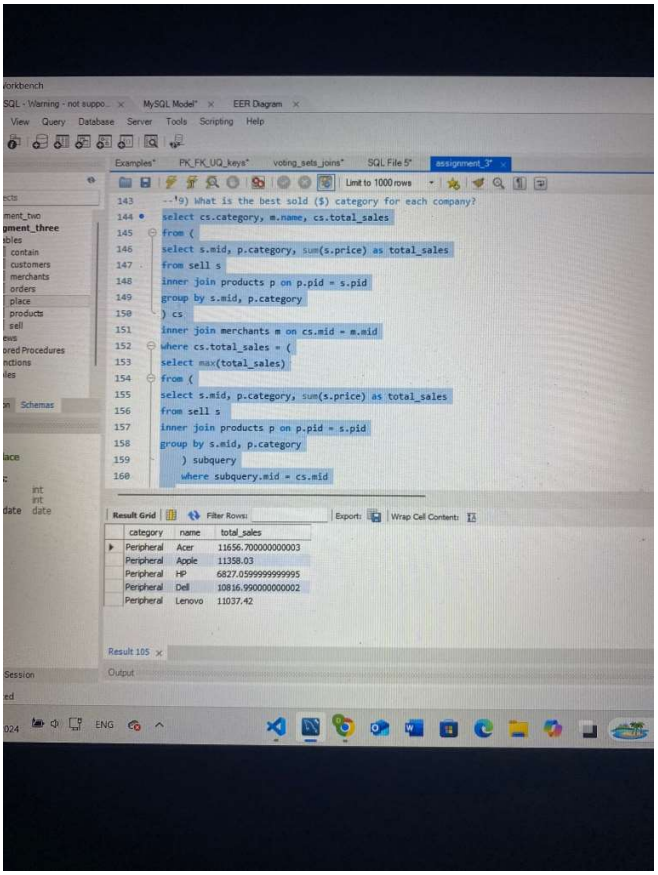
8) **On average, what was the cheapest shipping method used ever?**:

This query calculates the average shipping cost for each method to identify the cheapest one overall. It answers the question by providing insights into shipping cost efficiency across methods.



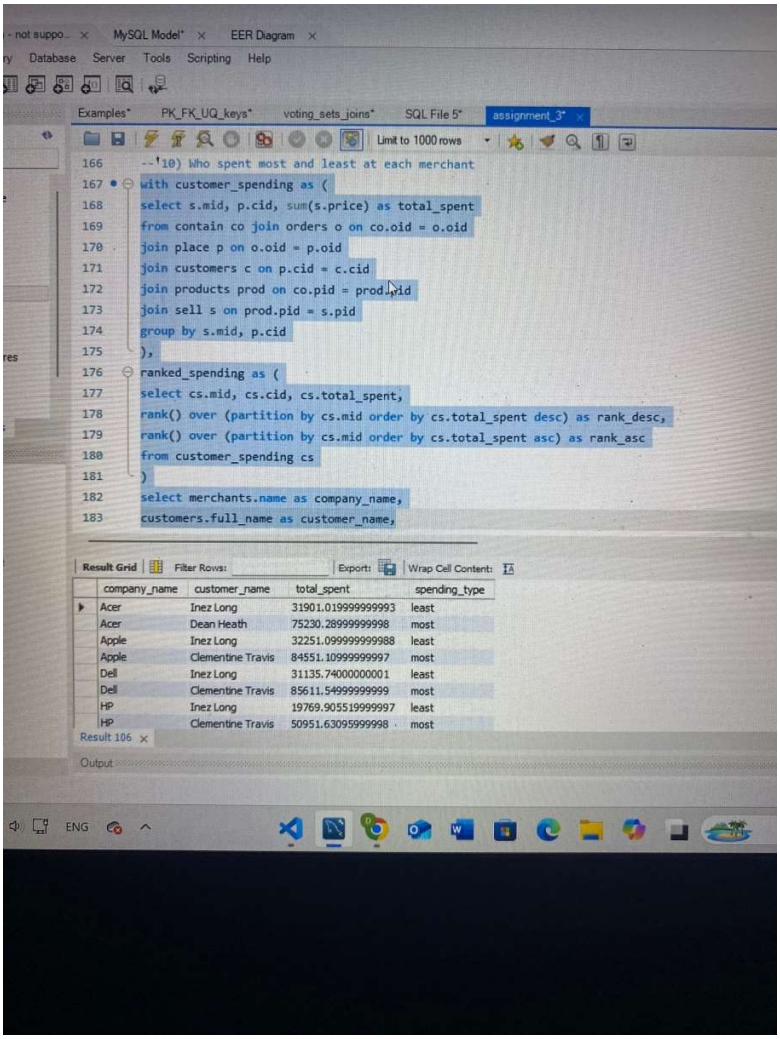
9) **What is the best sold ($) category for each company?**:

This SQL query identifies the category, name, and total sales of the merchant(s) with the highest sales for each product category. It first calculates the total sales for each merchant and product category by joining the sell and products tables. Then, it joins these results with the merchants table to get the merchant names. Finally, it filters the results to include only those merchants whose total sales match the maximum sales for that merchant, ensuring that only the top sellers in each category are displayed.

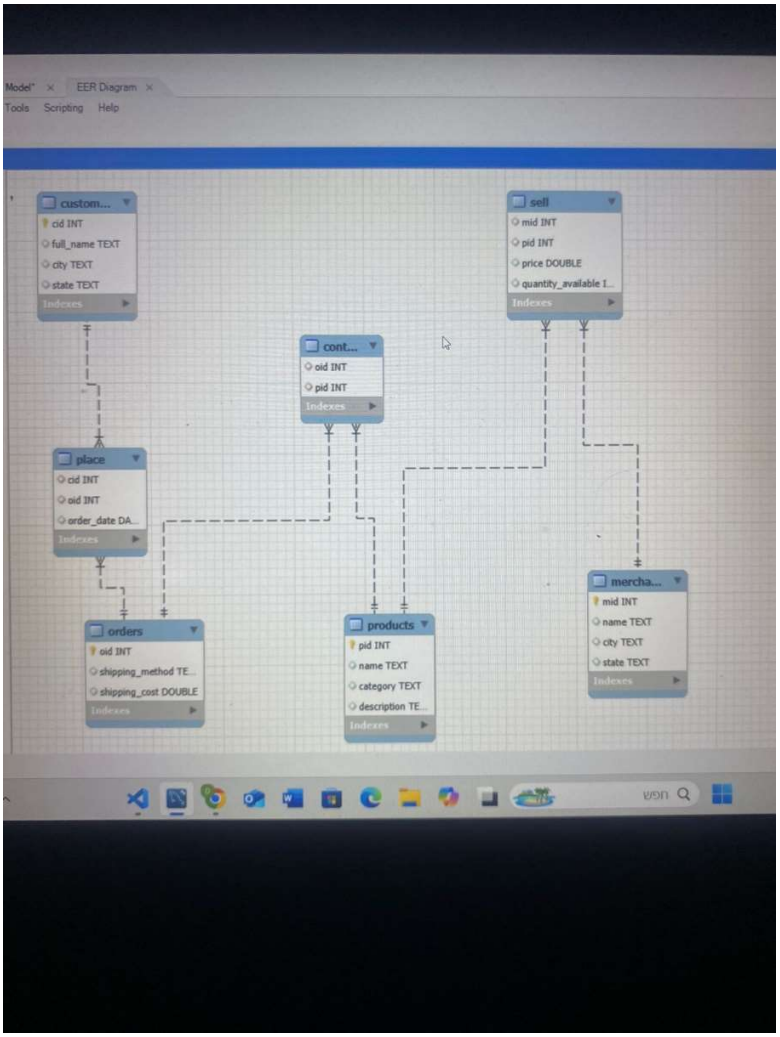


10) **For each company find out which customers have spent the most and the least amounts**:

The query calculates total spending for each customer per merchant using, then ranks them with to identify the highest and lowest spenders. It filters results to include only the top and bottom ranks for each merchant, joining the necessary tables to retrieve customer and merchant names. The final output clearly shows the highest and lowest spending customers for each company.


MySQL Model*
EER Diagram
Database
Server
Tools
Scripting
Help
Examples*
PK_FK_UQ_keys*
voting_sets_joins*
SQL File 5*
assignment_3*
Limit to 1000 rows
--'10) Who spent most and least at each merchant
with customer_spending as (
select s.mid, p.cid, sum(s.price) as total_spent
from contain co join orders o on co.oid = o.oid
join place p on o.oid = p.oid
join customers c on p.cid = c.cid
join products prod on co.pid = prod.pid
join sell s on prod.pid = s.pid
group by s.mid, p.cid
),
ranked_spending as (
select cs.mid, cs.cid, cs.total_spent,
rank() over (partition by cs.mid order by cs.total_spent desc) as rank_desc,
rank() over (partition by cs.mid order by cs.total_spent asc) as rank_asc
from customer_spending cs
)
select merchants.name as company_name,
customers.full_name as customer_name,
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
company_name customer_name total_spent spending_type
Acer Inez Long 31901.019999999993 least
Acer Dean Heath 75230.28999999998 most
Apple Inez Long 32251.099999999988 least
Apple Clementine Travis 84551.10999999997 most
Dell Inez Long 31135.74000000001 least
Dell Clementine Travis 85611.54999999999 most
HP Inez Long 19769.905519999997 least
HP Clementine Travis 50951.63095999998 most
Result 106
Output
ENG

ER model:


Model*
EER Diagram
Tools
Scripting
Help
custom...
cid INT
full_name TEXT
city TEXT
state TEXT
Indexes
sell
mid INT
pid INT
price DOUBLE
quantity_available I...
Indexes
cont...
oid INT
pid INT
Indexes
place
cid INT
oid INT
order_date DA...
Indexes
orders
oid INT
shipping_method TE...
shipping_cost DOUBLE
Indexes
products
pid INT
name TEXT
category TEXT
description TE...
Indexes
mercha...
mid INT
name TEXT
city TEXT
state TEXT
Indexes