HW 3

Title: DB Assignment 3

Name: Brennen Cramp

Date: 10/11/2024

Problem 1:

First, I wanted to combine the 3 tables (merchants, sell, products) with inner joins to be able to get which products are sold by which merchants. A projection is then used to grab the name of the merchant and the name of the product they sell where the product is no longer available (quantity is equal to 0).

```
97      -- Problem 1:
98      -- List names and sellers of products that are no longer available (quantity=0)
99
100 •   SELECT m.name AS sellerName, p.name AS productName            -- Grabs the merch
101     FROM (merchants AS m INNER JOIN sell AS s ON m.mid = s.mid    -- Combines mercha
102             INNER JOIN products AS p ON p.pid = s.pid)            -- Resulting previ
103     WHERE quantity_available = 0;                                 -- Finds where the
104
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

sellerName	productName
Acer	Router
Acer	Network Card
Apple	Printer
Apple	Router
HP	Router
HP	Super Drive
HP	Laptop
Dell	Router
Lenovo	Ethernet Adapter

Problem 2:

Starting off, I wanted to combine the 2 tables (products and sell) with a left join to be able to get which products are sold via a corresponding merchant ID. A projection is then used to retrieve the name of the product and its description where merchant is not matched to a product thus the merchant ID is null (i.e. meaning that corresponding product is not sold).

```
87      -- Problem 2:
88      -- List names and descriptions of products that are not sold
89
90      SELECT p.name AS productName, description                    --
91      FROM (products AS p LEFT JOIN sell AS s ON p.pid = s.pid)    --
92      WHERE s.mid IS NULL;                                         --
93
94
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	productName	description
▶	Super Drive	External CD/DVD/RW
	Super Drive	UInternal CD/DVD/RW

Problem 3:

I combined the 5 tables (customers, place, orders, contain, products) with inner joins to be able to get what orders the customers placed and the products that were bought. A projection is then used to grab the count of the customers where they have bought SATA drives and not products with the name “Router.”

```
94      -- Problem 3:
95      -- How many customers bought SATA drives but not any routers
96
97  •   SELECT COUNT(cust.cid) AS totalCustomers
98      FROM (customers AS cust INNER JOIN place AS pl ON cust.cid = pl.cid
99              INNER JOIN orders AS o ON o.oid = pl.oid
100             INNER JOIN contain AS cn ON o.oid = cn.oid
101             INNER JOIN products AS pr ON pr.pid = cn.pid)
102     WHERE description LIKE '%SATA%' AND pr.name NOT LIKE '%Router%';
103
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

totalCustomers
96

Problem 4:

First, I wanted to combine 3 tables (merchants, sell, products) with inner joins to be able to get which products are sold by which merchants. Next, a projection was used to grab the name of the product, quantity available, description of the product, the original price of the product, and I calculated the 20% sale price of the product. Finally, I filter on the networking category of products sold by the HP company.

```
140     -- Problem 4:
141     -- HP has a 20% sale on all its Networking products
142
143 •   SELECT p.name, quantity_available, description, price AS origPrice,     -- Grabs
144             (price - (price * 0.2)) AS salePrice                            -- Final
145     FROM (merchants AS m INNER JOIN sell AS s ON m.mid = s.mid              -- Combi
146             INNER JOIN products AS p ON p.pid = s.pid)                      -- Resul
147     WHERE m.name = "HP" AND category = "Networking";                        -- Filte
148
149
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	quantity_available	description	origPrice	salePrice
Router	3	Gigabit Router with USB for Windows and Linux	1034.46	827.568
Network Card	2	Wireless a/b/g/n	1154.68	923.744
Network Card	3	100Mbps Full/Low Profile Ethernet Multi Mode S...	345.01	276.008
Network Card	15	Livewire Powerline AV Kit	262.2	209.76
Ethernet Adapter	2	High Performance Wireless-N Adapter	1260.45	1008.36
Router	0	Wireless-G Broadband Router with VPN	205.56	164.448
Router	11	Wireless Dual Band Gigabit Router	1474.87	1179.896
Router	9	Wireless N HD Media Router 1000 DIR-657	552.02	441.616
Router	9	54 Mbps 4-port Wireless 802.11g DSL Router	100.95	80.76
Network Card	6	MegaPlug AV 200 Mbs	1179.01	943.208

Problem 5:

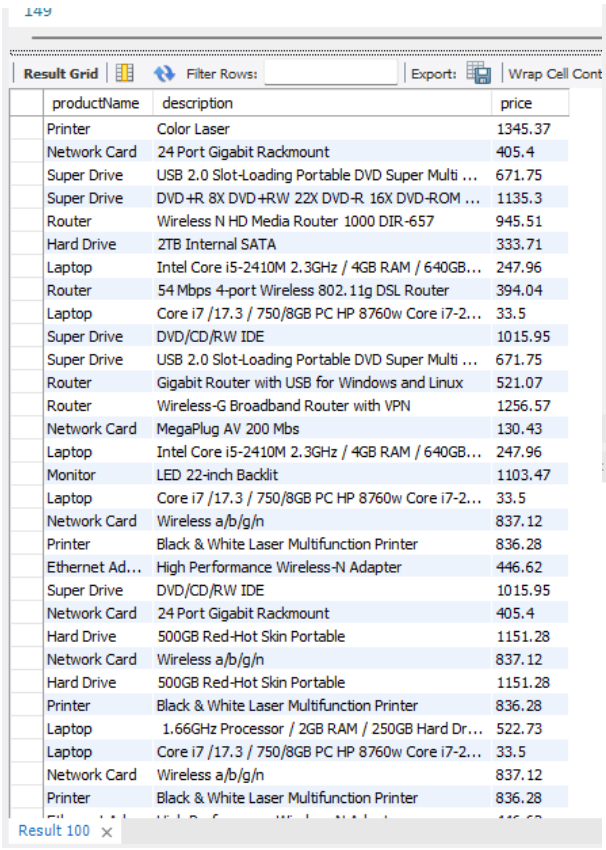
I first combined 7 tables (customers, place, orders, contain, products, sell, merchants) with inner joins to be able to get all the companies and the customers who have bought products, along with the price of the products bought. Next, I used a projection to grab the name of the product, the product's description, and the price of the product where I then sort based on the customer's full name (Uriel Whitney) and the merchants name (Acer) to show me all the products that Uriel Whitney has bought from Acer.

```
151     -- Problem 5:
152     -- What did Uriel Whitney order from Acer (make sure to at least retrieve product names and prices)
153
154 •   SELECT pr.name AS productName, description, price                    -- Grabs the product name, 
155     FROM (customers AS cust INNER JOIN place AS pl ON cust.cid = pl.cid  -- Combines customers and p
156             INNER JOIN orders AS o ON o.oid = pl.oid                     -- Resulting previous table
157             INNER JOIN contain AS cn ON o.oid = cn.oid                   -- Resulting previous table
158             INNER JOIN products AS pr ON pr.pid = cn.pid                 -- Resulting previous table
159             INNER JOIN sell AS s ON pr.pid = s.pid                       -- Resulting previous table
160             INNER JOIN merchants AS m ON m.mid = s.mid)                  -- Resulting previous table
161     WHERE cust.fullname = "Uriel Whitney" AND m.name = "Acer";           -- Filters on just the prod
162
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

productName	description	price
Monitor	27-inch LED	1435.38
Router	Gigabit Router with USB for Windows and Linux	521.07
Router	Wireless-G Broadband Router with VPN	1256.57
Monitor	LED 22-inch Backlit	1103.47
Super Drive	12x Internal Blu-ray Disc/DVD/CD Writer	356.13
Printer	Color Laser	1345.37
Super Drive	USB 2.0 Slot-Loading Portable DVD Super Multi ...	671.75
Super Drive	DVD+R 8X DVD+RW 22X DVD-R 16X DVD-ROM ...	1135.3
Super Drive	12x Internal Blu-ray Disc/DVD/CD Writer	356.13
Super Drive	DVD/CD/RW IDE	1015.95
Network Card	24 Port Gigabit Rackmount	405.4
Hard Drive	640GB USB 2.0 Portable	836.99
Super Drive	External CD/DVD R/RW USB	1124.26
Network Card	Livewire Powerline AV Kit	609.2

Result 100



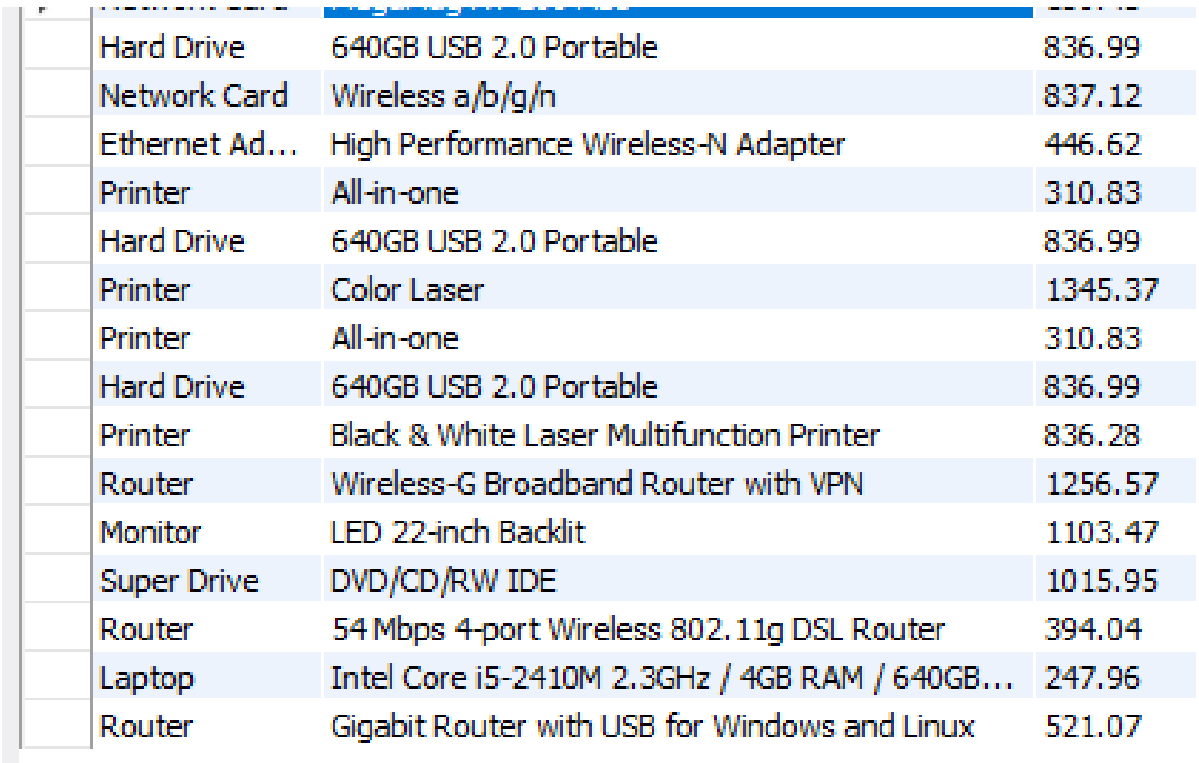
Result Grid | Filter Rows: | Export: | Wrap Cell Cont

productName	description	price
Printer	Color Laser	1345.37
Network Card	24 Port Gigabit Rackmount	405.4
Super Drive	USB 2.0 Slot-Loading Portable DVD Super Multi ...	671.75
Super Drive	DVD+R 8X DVD+RW 22X DVD-R 16X DVD-ROM ...	1135.3
Router	Wireless N HD Media Router 1000 DIR-657	945.51
Hard Drive	2TB Internal SATA	333.71
Laptop	Intel Core i5-2410M 2.3GHz / 4GB RAM / 640GB...	247.96
Router	54 Mbps 4-port Wireless 802.11g DSL Router	394.04
Laptop	Core i7 /17.3 / 750/8GB PC HP 8760w Core i7-2...	33.5
Super Drive	DVD/CD/RW IDE	1015.95
Super Drive	USB 2.0 Slot-Loading Portable DVD Super Multi ...	671.75
Router	Gigabit Router with USB for Windows and Linux	521.07
Router	Wireless-G Broadband Router with VPN	1256.57
Network Card	MegaPlug AV 200 Mbs	130.43
Laptop	Intel Core i5-2410M 2.3GHz / 4GB RAM / 640GB...	247.96
Monitor	LED 22-inch Backlit	1103.47
Laptop	Core i7 /17.3 / 750/8GB PC HP 8760w Core i7-2...	33.5
Network Card	Wireless a/b/g/n	837.12
Printer	Black & White Laser Multifunction Printer	836.28
Ethernet Ad...	High Performance Wireless-N Adapter	446.62
Super Drive	DVD/CD/RW IDE	1015.95
Network Card	24 Port Gigabit Rackmount	405.4
Hard Drive	500GB Red-Hot Skin Portable	1151.28
Network Card	Wireless a/b/g/n	837.12
Hard Drive	500GB Red-Hot Skin Portable	1151.28
Printer	Black & White Laser Multifunction Printer	836.28
Laptop	1.66GHz Processor / 2GB RAM / 250GB Hard Dr...	522.73
Laptop	Core i7 /17.3 / 750/8GB PC HP 8760w Core i7-2...	33.5
Network Card	Wireless a/b/g/n	837.12
Printer	Black & White Laser Multifunction Printer	836.28

Result 100

Result Grid | Filter Rows: | Export: | Wrap Cell Co

productName	description	price
Ethernet Ad...	High Performance Wireless-N Adapter	446.62
Super Drive	USB 2.0 Slot-Loading Portable DVD Super Multi ...	671.75
Network Card	Wireless a/b/g/n	837.12
Network Card	Livewire Powerline AV Kit	609.2
Desktop	Intel Core i7-2630QM 2.00GHz / 8GB RAM / 2TB...	311.06
Monitor	LED 22-inch Backlit	1103.47
Super Drive	12x Internal Blu-ray Disc/DVD/CD Writer	356.13
Printer	All-in-one	310.83
Super Drive	12x Internal Blu-ray Disc/DVD/CD Writer	356.13
Network Card	MegaPlug AV 200 Mbs	130.43
Router	Wireless Dual Band Gigabit Router	780.65
Super Drive	12x Internal Blu-ray Disc/DVD/CD Writer	356.13
Router	54 Mbps 4-port Wireless 802.11g DSL Router	394.04
Laptop	Core i7 /17.3 / 750/8GB PC HP 8760w Core i7-2...	33.5
Printer	All-in-one	310.83
Hard Drive	2TB Internal SATA	333.71
Laptop	Intel Core i5-2410M 2.3GHz / 4GB RAM / 640GB...	247.96
Network Card	Livewire Powerline AV Kit	609.2
Router	Wireless Dual Band Gigabit Router	780.65
Hard Drive	500GB Red-Hot Skin Portable	1151.28
Network Card	Wireless a/b/g/n	837.12
Router	Wireless Dual Band Gigabit Router	780.65
Printer	Color Laser	1345.37
Network Card	MegaPlug AV 200 Mbs	130.43
Laptop	Intel Core i5-2410M 2.3GHz / 4GB RAM / 640GB...	247.96
Desktop	Intel Core i7-2630QM 2.00GHz / 8GB RAM / 2TB...	311.06
Router	Wireless-G Broadband Router with VPN	1256.57
Super Drive	12x Internal Blu-ray Disc/DVD/CD Writer	356.13
Super Drive	DVD/CD/RW IDE	1015.95
Network Card	MegaPlug AV 200 Mbs	130.43



Hard Drive	640GB USB 2.0 Portable	836.99
Network Card	Wireless a/b/g/n	837.12
Ethernet Ad...	High Performance Wireless-N Adapter	446.62
Printer	All-in-one	310.83
Hard Drive	640GB USB 2.0 Portable	836.99
Printer	Color Laser	1345.37
Printer	All-in-one	310.83
Hard Drive	640GB USB 2.0 Portable	836.99
Printer	Black & White Laser Multifunction Printer	836.28
Router	Wireless-G Broadband Router with VPN	1256.57
Monitor	LED 22-inch Backlit	1103.47
Super Drive	DVD/CD/RW IDE	1015.95
Router	54 Mbps 4-port Wireless 802.11g DSL Router	394.04
Laptop	Intel Core i5-2410M 2.3GHz / 4GB RAM / 640GB...	247.96
Router	Gigabit Router with USB for Windows and Linux	521.07

Problem 6:

I combined 5 tables (merchants, sell, products, contain, place) with inner joins to be able to get all the companies and the products that have been sold with their order dates. Next, I used a projection to retrieve the company name, year (using the year() function), and the sum of the price multiplied by the quantity to get the total sales. I also used group by so the sum aggregate can count the total sales by a given company and its corresponding year. Finally, I ordered the results alphanumerically by the company and year.

```
128     -- Problem 6:
129     -- List the annual total sales for each company (sort the results along the company and the year attributes)
130     -- Assumption: We can assume that the shipping cost is not factored into the sales.
131
132 •   SELECT m.name AS company, YEAR(order_date) AS year,          -- Grabs the company name, year of the order, a
133             SUM(price * quantity_available) AS totalSales       -- creates an aggregate of the totalSales with
134     FROM (merchants AS m INNER JOIN sell AS s ON m.mid = s.mid  -- Combines merchants and sell (via the same mi
135             INNER JOIN products AS pr ON pr.pid = s.pid         -- Resulting previous table combo is combined w
136             INNER JOIN contain AS cn ON cn.pid = pr.pid         -- Resulting previous table combo is combined w
137             INNER JOIN place AS pl ON pl.oid = cn.oid)          -- Resulting previous table combo is combined w
138     GROUP BY company, year                                      -- Groups the results by company name and the y
139     ORDER BY company, year;                                     -- In ascending order (i.e. alphanumeric), orde
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

company	year	totalSales
Acer	2020	1062622.300000001
Apple	2011	972240.9200000012
Apple	2016	409402.3799999999
Apple	2017	1071712.9300000023
Apple	2018	1664629.7700000035
Apple	2019	1311417.5700000043
Apple	2020	1213964.9600000044
Dell	2011	1542228.9899999993
Dell	2016	625684.1399999999
Dell	2017	1522794.2799999989
Dell	2018	2601060.96
Dell	2019	1796684.0299999958
Dell	2020	1736811.8599999968
HP	2011	873547.1000000003
HP	2016	375547.4499999999
HP	2017	938168.0300000005
HP	2018	1281764.9500000025
HP	2019	1111063.4600000023
HP	2020	1164518.2700000005
Lenovo	2011	1235551.840000001
Lenovo	2016	483906.5600000001
Lenovo	2017	1329707.7699999972
Lenovo	2018	2090330.0999999978
Lenovo	2019	1573616.3799999987
Lenovo	2020	1306860.859999998

Result 189

Problem 7:

I combined 5 tables (merchants, sell, products, contain, place) with inner joins to be able to get all the companies and the products that have been sold with their order dates. Next, I used a projection to retrieve the company name, year (using the year() function), and the sum of the price multiplied by the quantity to get the total sales. I also used group by so the sum aggregate can count the total sales by a given company and its corresponding year. Finally, I ordered the total sales from highest to lowest and used limit 1 to grab the highest annual revenue.

```
142     -- Problem 7:
143     -- Which company had the highest annual revenue and in what year?
144
145     SELECT m.name AS company, YEAR(order_date) AS year,          -- G
146             SUM(price * quantity_available) AS totalSales        -- c
147     FROM (merchants AS m INNER JOIN sell AS s ON m.mid = s.mid   -- C
148             INNER JOIN products AS pr ON pr.pid = s.pid          -- R
149             INNER JOIN contain AS cn ON cn.pid = pr.pid          -- R
150             INNER JOIN place AS pl ON pl.oid = cn.oid)           -- R
151     GROUP BY company, year                                       -- G
152     ORDER BY totalSales DESC LIMIT 1;                            -- I
153
```

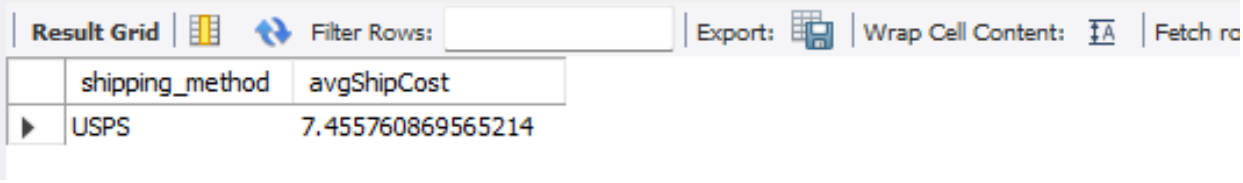
Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch row

company	year	totalSales
Dell	2018	2601060.96

Problem 8:

For this query, I can simply use the orders table where a projection is used to grab the method of shipping and an aggregation to calculate the average shipping cost. I also grouped by the shipping method for the aggregation to find the average shipping cost per the method of shipping and, in ascending order, grabbed the top result off the query to grab the cheapest shipping method (on average) used.

```
173     -- Problem 8:
174     -- On average, what was the cheapest shipping method used ever?
175
176 •   SELECT shipping_method, AVG(shipping_cost) AS avgShipCost     -- G
177     FROM orders                                                   -- U
178     GROUP BY shipping_method                                      -- G
179     ORDER BY avgShipCost LIMIT 1;                                 -- I
180
181
```



shipping_method	avgShipCost
USPS	7.455760869565214

Problem 9:

Using a CTE for the problem, I first combined 5 tables (orders, contain, products, sell, merchants) with inner joins to be able to get all the companies and the best category per company based on its total sold. Next, I used a projection to retrieve the company, category, and sum aggregate of the price multiplied by the quantity available to get the total sold, grouping by the company and category. With that query forming my CTE, my next query is retrieving the company and the highest total from the CTE table where I group the aggregate by company. However, I was stuck on how to grab the category with this query, meaning I am only displaying the company and the total sold.

```
163     -- Problem 9:
164     -- What is the best sold ($) category for each company?
165
166 ●   WITH bestSoldCatPerComp AS (
167         SELECT m.name AS company, category, SUM(price * quantity_available) AS total
168         FROM (orders AS o INNER JOIN contain AS cn ON o.oid = cn.oid
169                 INNER JOIN products AS pr ON pr.pid = cn.pid
170                 INNER JOIN sell AS s ON pr.pid = s.pid
171                 INNER JOIN merchants AS m ON m.mid = s.mid)
172         GROUP BY company, category
173     )
174     SELECT company, MAX(total) AS totalSold     -- Grabs the company and the max total
175     FROM bestSoldCatPerComp                     -- Uses the CTE formed above
176     GROUP BY company;                           -- Groups the aggregate on the company
177
178
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

company	totalSold
Acer	5281119.98999994
Apple	3938546.6100000143
Dell	6338444.07
Lenovo	5336522.410000019
HP	3055029.3099999996

Problem 10:

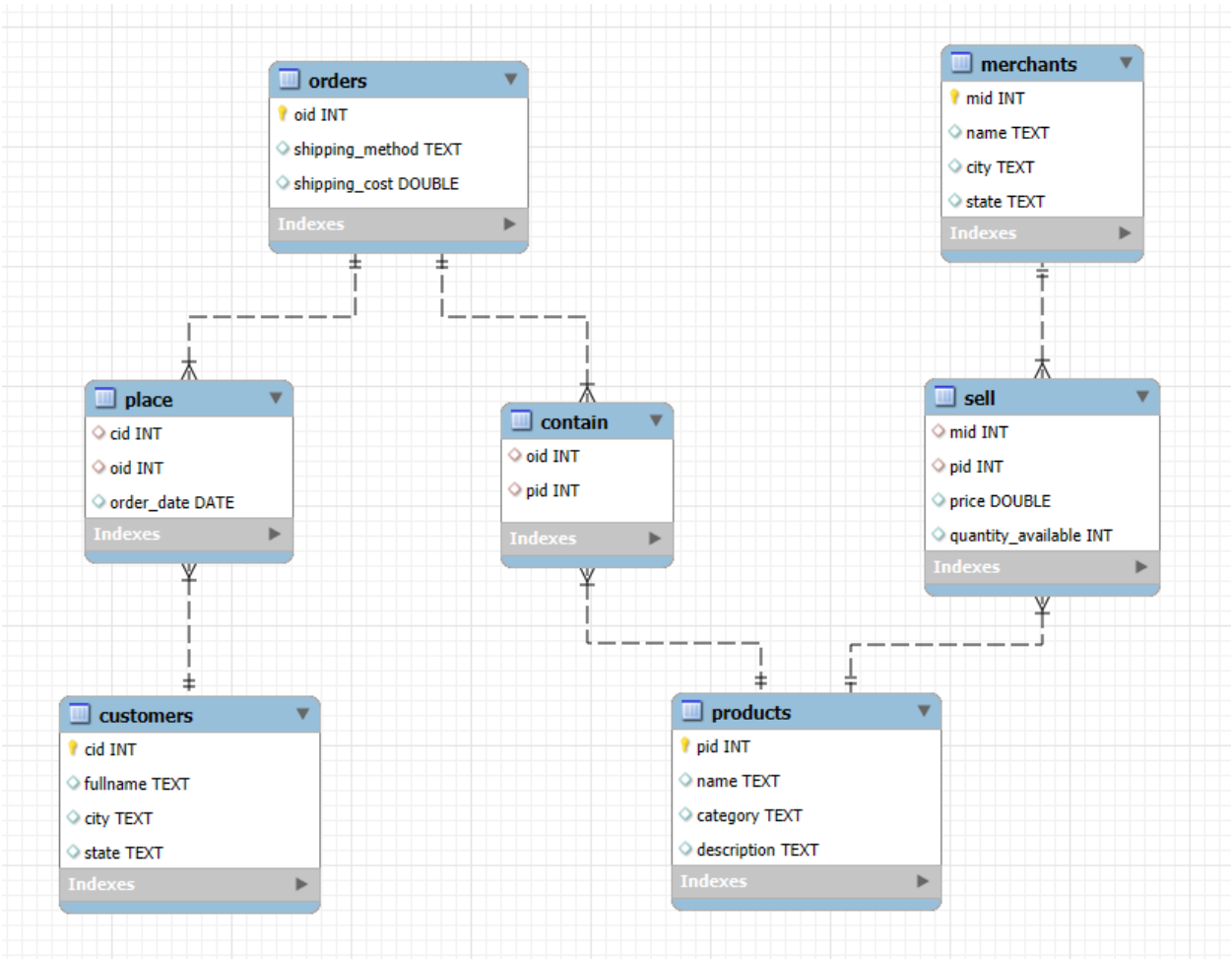
Using a CTE for the problem, I first combined 7 tables (customers, place, orders, contain, products, sell, merchants) with inner joins to be able to get all the companies and the customers who have bought products, along with the price of the products bought. Next, I used a projection to retrieve the company, customer name, and sum aggregate of the price to get the total the customer spent, grouping by the company and customer name. With that query forming my CTE, my next queries are retrieving the company and the highest total from the CTE table where I group the aggregate by company; this is then combined using "Union" with the lowest total found in the CTE table when grouping by the company (all done in company alphabetic order). However, like the previous problem, I was stuck on how to grab the customer's name with this query, meaning I am only displaying the company and the highest and lowest totals spent by the customer at that company. This query and results are shown on the next page:

```sql
-- Problem 10:
-- For each company find out which customers have spent the most and the least amounts.

WITH totalSpentByCustPerComp AS (                                   -- Creates a CT
    SELECT  m.name AS company, fullname, SUM(price) AS totalSpent   -- Grabs the co
    FROM (customers AS cust INNER JOIN place AS pl ON cust.cid = pl.cid  -- Combines cus
            INNER JOIN orders AS o ON o.oid = pl.oid                -- Resulting pr
            INNER JOIN contain AS cn ON o.oid = cn.oid              -- Resulting pr
            INNER JOIN products AS pr ON pr.pid = cn.pid            -- Resulting pr
            INNER JOIN sell AS s ON pr.pid = s.pid                  -- Resulting pr
            INNER JOIN merchants AS m ON m.mid = s.mid)             -- Resulting pr
    GROUP BY company, fullname                                      -- Groups the s
)
-- Query that gets the most spent
SELECT company, MAX(totalSpent) AS amountSpent      -- Grabs the company and the max total t
FROM totalSpentByCustPerComp                        -- Uses the CTE formed above
GROUP BY company                                    -- Groups the aggregate on the company

UNION                                               -- Uses a union to combine the highest a

-- Query that gets the least spent
SELECT company, MIN(totalSpent) AS amountSpent      -- Grabs the company and the min total t
FROM totalSpentByCustPerComp                        -- Uses the CTE formed above
GROUP BY company                                    -- Groups the aggregate on the company
ORDER BY company;                                   -- Orders by the name of the company alp

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

company	amountSpent
Acer	75230.28999999998
Acer	31901.019999999993
Apple	84551.10999999997
Apple	32251.099999999988
Dell	85611.54999999999
Dell	31135.74000000001
HP	66628.05999999995
HP	26062.89
Lenovo	83030.25999999997
Lenovo	33948.909999999996

ERD Diagram:


orders
oid INT
shipping_method TEXT
shipping_cost DOUBLE
Indexes
merchants
mid INT
name TEXT
city TEXT
state TEXT
Indexes
place
cid INT
oid INT
order_date DATE
Indexes
contain
oid INT
pid INT
Indexes
sell
mid INT
pid INT
price DOUBLE
quantity_available INT
Indexes
customers
cid INT
fullname TEXT
city TEXT
state TEXT
Indexes
products
pid INT
name TEXT
category TEXT
description TEXT
Indexes