Title: DB Assignment 3

Your Name: Cooper Simon

Date: 10/21/25

```
/* ********************************************************************************************
   QUERY SECTION (Question 1)
   ********************************************************************************************
  List names and sellers of products that are no longer available (quantity=0)
********************************************************************************************** */

select products.name as products_name,
            merchants.name as merchant_name
from products inner join sell using(pid)
                              inner join merchants using(mid)
where sell.quantity_available = 0;
```

products_name	merchant_name
Router	Acer
Network Card	Acer
Printer	Apple
Router	Apple
Laptop	HP
Router	HP
Super Drive	HP
Router	Dell
Ethernet Adapter	Lenovo

Q1 Explanation: This query joins the three tables, merchant, sell, and product, in order to list out the products and merchants that are no longer available. It does this by using a where clause, which is looking at the attribute quantity_available on the sell table.

```
/* ********************************************************************************************
   QUERY SECTION (Question 2)
   ********************************************************************************************
```

```
  List names and descriptions of products that are not sold.
********************************************************************************************** */

select products.name as product_name,
            products.description as descriptions
from products left outer join sell
           on products.pid = sell.pid
where sell.pid is null;
```

product_name	descriptions
Super Drive	External CD/DVD/RW
Super Drive	UInternal CD/DVD/RW

Q2 Explanation: This query lists all products that were not sold by comparing two tables; product and sell. A left join ensures all products(left side) are included, and filtering with sell.pid is null isolates those with no sales. The final output shows the names and descriptions of unsold products.

```
/* ********************************************************************************************
   QUERY SECTION (Question 3)
   ********************************************************************************************
  How many customers bought SATA drives but not any routers?
********************************************************************************************** */
select count(*) as total_count
from (
    select distinct cid
    from customers inner join place using (cid)
    inner join contains using (oid)
    inner join products using (pid)
    where products.description like '%SATA%'

    except

    select distinct cid
    from customers inner join place using (cid)
    inner join contains using (oid)
    inner join products using (pid)
    where products.description like '%Router%'
) as total_count;
```

total_count
0

Q3 Explanation: This query first identifies all the customers who use Sata products. However we also want to count those same customers which also did not buy any routers as well. We do this using an except command. Although in our data set there are no customers who buy a sata product without buying a router.

```
/* ********************************************************************************************
   QUERY SECTION (Question 4)
   ********************************************************************************************
HP has a 20% sale on all its Networking products.
******************************************************************************************** */

select merchants.name as merchant,
       products.name as product,
       sell.price as current_price,
       sell.price * 0.8 as discounted_price
from sell inner join merchants using (mid)
                    inner join products using (pid)
where merchants.name = 'HP' and products.category = 'Networking';
```

merchant	product	current_price	discounted_price
HP	Router	1034.46	827.5680000000001
HP	Network Card	1154.68	923.7440000000001
HP	Network Card	345.00999999999993	276.008
HP	Network Card	262.2	209.76
HP	Ethernet Adapter	1260.45	1008.3600000000001
HP	Router	205.56	164.448
HP	Router	1474.87	1179.896
HP	Router	552.02	441.616
HP	Router	100.95	80.76
HP	Network Card	1179.0100000000002	943.2080000000002

Q4 Explanation: In this query we do not actually update our database current price, instead we create a new discounted price variable that displays what the discounted price would be.

```
/* ********************************************************************************************
   QUERY SECTION (Question 5)
```

```
  ********************************************************************************************
What did Uriel Whitney order
******************************************************************************************** */
select distinct products.name as Product_Name,
      avg(sell.price) as Price
from customers inner join place using (cid)
                              inner join contains using (oid)
            inner join products using (pid)
            inner join sell using (pid)
where customers.fullname = 'Uriel Whitney'
group by products.name;
```

Product_Name	Price
Super Drive	857.1565789473675
Network Card	751.5183146067418
Hard Drive	860.9584210526315
Printer	858.4893333333332
Monitor	851.9311111111112
Router	800.0658208955223
Laptop	626.5613953488372
Ethernet Adapter	645.778
Desktop	936.768

Q5 Explanation : In this question it wants us too list the products which the customer Uriel Whitney has purchased, along with the price of those products. Our database does not allow us to access the price of the exact product which Uriel purchased however we can use an average price of what that product typically would cost.

```
/* ********************************************************************************************
  QUERY SECTION (Question 6)
  ********************************************************************************************
List the annual total sales for each company (sort the results along the company and the year attributes).
******************************************************************************************** */
select merchants.name as Merchant,
            year(place.order_date) as year,
      sum(sell.price * sell.quantity_available) as Total_Sales
from merchants inner join sell using(mid)
                              inner join products using(pid)
```

```
        inner join contains using(pid)
        inner join place using (oid)
group by Merchant, year
order by Merchant, year;
```

Merchant	year	Total_Sales
Acer	2011	828677.08
Acer	2016	307909.83
Acer	2017	1100206.85
Acer	2018	1592886.5799999966
Acer	2019	1180216.6999999993
Acer	2020	1062622.2999999996
Apple	2011	972240.9199999986
Apple	2016	409402.38000000006
Apple	2017	1071712.9300000004
Apple	2018	1664629.770000003
Apple	2019	1311417.5699999994
Apple	2020	1213964.9599999995
Dell	2011	1542228.9899999995
Dell	2016	625684.1399999997
Dell	2017	1522794.2799999984
Dell	2018	2601060.960000003
Dell	2019	1796684.029999999
Dell	2020	1736811.86
HP	2011	873547.1000000001
HP	2016	375547.44999999995
HP	2017	938168.0300000004
HP	2018	1281764.9500000016
HP	2019	1111063.4600000004
HP	2020	1164518.2700000005
Lenovo	2011	1235551.84
Lenovo	2016	483906.55999999994
Lenovo	2017	1329707.7699999998
Lenovo	2018	2090330.1000000031
Lenovo	2019	1573616.3800000018
Lenovo	2020	1306860.8599999992

Q6 Explanation : This query wants us to specifically state the total sales of each company depending on the year.

```
/* ********************************************************************************************
   QUERY SECTION (Question 7)
   ********************************************************************************************
Which company had the highest annual revenue and in what year?
******************************************************************************************** */
select merchants.name as Merchant,
            year(place.order_date) as year,
       sum(sell.price * sell.quantity_available) as Total_Sales
from merchants inner join sell using(mid)
                              inner join products using(pid)
            inner join contains using(pid)
            inner join place using (oid)
group by Merchant, year
order by year limit 1;
```

Merchant	year	Total_Sales
Acer	2011	828677.08

Q7 Explanation: Similar to Query 6; however, this time we need to limit the number of merchants that can be listed. Which is why we use the limit 1 keyword.

```
/* ********************************************************************************************
   QUERY SECTION (Question 8)
   ********************************************************************************************
On average, what was the cheapest shipping method used ever?
******************************************************************************************** */
select orders.shipping_method as Shipping_Method,
            round(avg(orders.shipping_cost), 2) as AVG_Cost
from orders
group by shipping_method
order by AVG_Cost asc limit 1;
```

Shipping_Meth...	AVG_Cost
USPS	7.46

Q8 Explanation : This query discovers the cheapest average shipping method .

```
/* ********************************************************************************************
   QUERY SECTION (Question 9)
   ********************************************************************************************
```

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

```
select
    merchants.name as Merchant,
    products.category as Category,
    round(sum(sell.price * sell.quantity_available), 2)as Total_Sales
from sell
     inner join merchants using (mid)
     inner join products  using (pid)
group by merchants.name, products.category

having sum(sell.price * sell.quantity_available) >= all
(
    /* for this same merchant, compute the totals across its categories and
       keep only those categories whose total is >= every other category total */
    select
        sum(s2.price * s2.quantity_available)
    from sell s2
         inner join merchants m2 using (mid)
         inner join products  p2 using (pid)
    where m2.name = merchants.name              -- correlate on the SAME merchant
    group by p2.category
)
order by Merchant;
```

Merchant	Category	Total_Sales
Acer	Peripheral	78136.53
Apple	Peripheral	63974.74
Dell	Peripheral	100753.96
HP	Peripheral	51133.47
Lenovo	Peripheral	83479.83

Query 9 Explanation : This query finds the best-selling product category for each company by comparing total sales across all categories. It calculates each company's total revenue per category and then uses a correlated subquery with HAVING >= ALL to keep only the highest totals. The final result lists each merchant alongside their most profitable category and the total dollar amount.

```
/* ********************************************************************************************
   QUERY SECTION (Question 10)
```

```
  ********************************************************************************************
For each company find out which customers have spent the most and the least amounts
******************************************************************************************** */
select sales.Merchant,
          Group_Concat(distinct case
                    when sales.Total_Sales = maxmin.max_total then sales.Customer
        end SEPARATOR ', ') as Highest_Spender,
        round(maxmin.max_total, 2) as Highest_Spent,
          Group_Concat(distinct case
                    when sales.Total_Sales = maxmin.min_total then sales.Customer
        end SEPARATOR ', ') as Lowest_Spender,
        round(maxmin.min_total, 2) as Lowest_Spent

from (
        select
                merchants.name as Merchant,
                customers.fullname AS Customer,
                sum(sell.price * sell.quantity_available)as Total_Sales
        from customers
        inner join place using(cid)
   inner join contains using(oid)
   inner join products using(pid)
   inner join sell using(pid)
   inner join merchants using(mid)
   group by Merchant, Customer
) as sales
 inner join (
        select
                Merchant,
      MAX(Total_Sales) as max_total,
      MIN(Total_Sales) as min_total
        from (
                select
                merchants.name as Merchant,
                customers.fullname AS Customer,
                sum(sell.price * sell.quantity_available)as Total_Sales
                from customers inner join place using(cid)
                                        inner join contains using(oid)
               inner join products using(pid)
               inner join sell using(pid)
               inner join merchants using(mid)
               group by Merchant, Customer
```

```
    ) as sub
    group by Merchant
) as maxmin
on sales.Merchant = maxmin.Merchant
group by sales.Merchant
order by sales.Merchant;
```

Merchant	Highest_Spender	Highest_Spent	Lowest_Spender	Lowest_Spent
Acer	Dean Heath	443713.32	Inez Long	190191.56
Apple	Clementine Travis	497858.48	Wynne Mckinney	193504.63
Dell	Clementine Travis	741615.84	Inez Long	259552.37
HP	Clementine Travis	412323.26	Wynne Mckinney	168651.54
Lenovo	Haviva Stewart	536047.37	Inez Long	243477.23

Q10 Explanation: This query determines which customers have spent the most and the least amounts for each company.
It first calculates each customer's total spending per merchant, then identifies the maximum and minimum totals for each merchant.
Using GROUP_CONCAT, it displays the highest- and lowest-spending customers alongside their respective spending totals in a single summarized table.
The result allows the company to easily identify its most and least valuable customers.