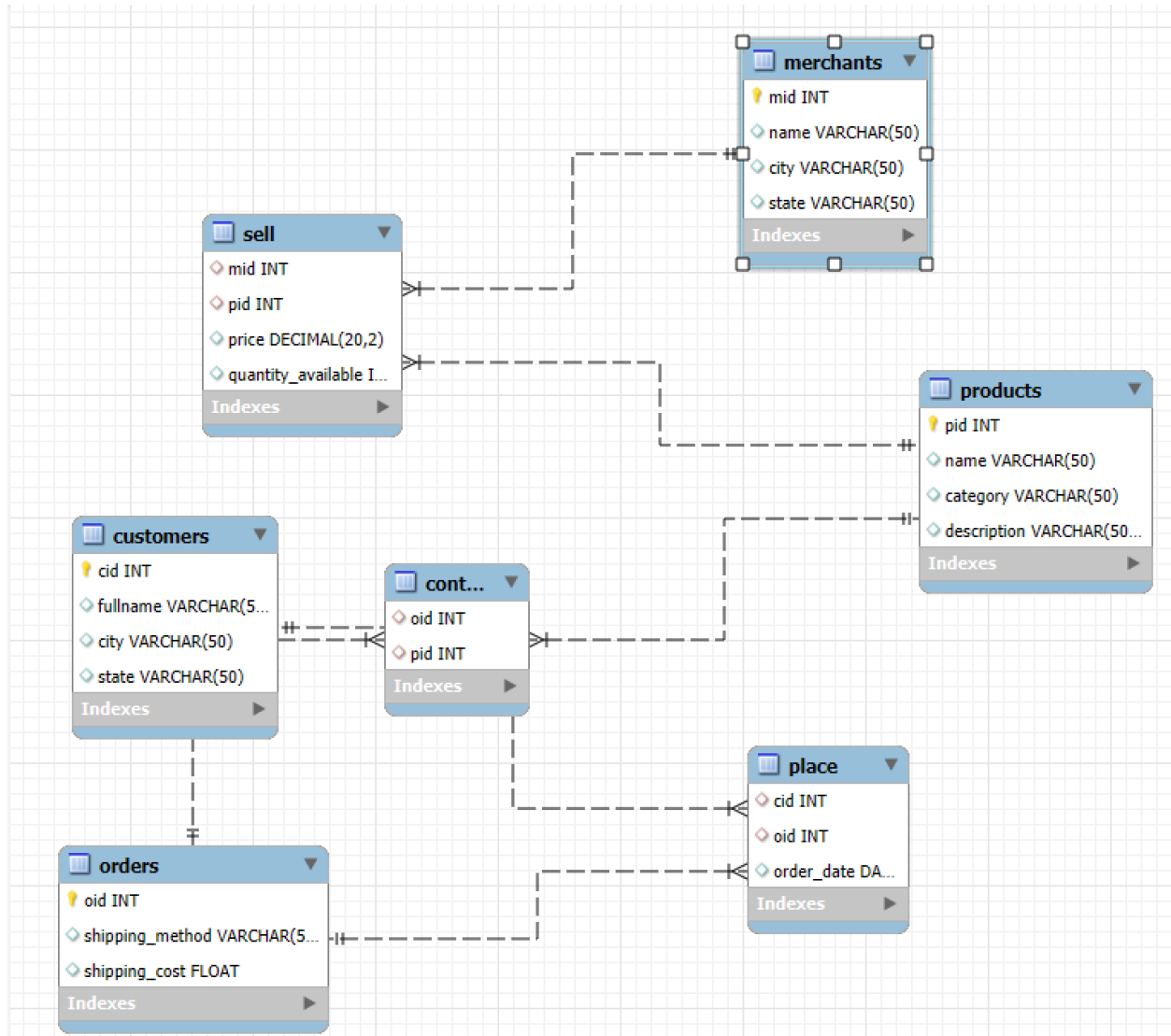


Title: DB Assignment 3
Name: Daniela Cruz Perez
Date: 10/21/2025

ER Diagram:



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

Explanation: Query cross joins tables merchants, sell, and products and retrieve where quantity_available is equal to 0.

SQL

```
select merchants.name as seller, products.name as product, quantity_available
from    merchants, sell, products
where   (merchants.mid = sell.mid) and (sell.pid = products.pid) and
quantity_available = 0;
```

	seller	product	quantity_available
	Acer	Router	0
	Acer	Network Card	0
	Apple	Printer	0
	Apple	Router	0
	HP	Router	0
	HP	Super Drive	0
	HP	Laptop	0
	Dell	Router	0
	Lenovo	Ethernet Adapter	0

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

Explanation: The query left outer joins tables products and contains and retrieves the products whose ids are null in contain. Results show only two products were not sold.

SQL

```
select products.name, description
from products left outer join contain
      using (pid)
where contain.pid is null;
```

name	description
Super Drive	External CD/DVD/RW
Super Drive	UInternal CD/DVD/RW

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

Explanation: The query selects the names from 2 subqueries, each subquery inner join tables customers, place, orders, contain, and products. The first selects the names of customers who purchased a product that contain the string bit 'SATA' and the second retrieves the customers who purchase a router. The main select retrieves the number of customers who are in the first subquery and excluded in the second. There are no customers who bought a SATA drive but not any routers. Results show there were no customers.

```
SQL
select count(distinct fullname) as bought_SATA_no_router
from (
select customers.fullname
from customers join place on customers.cid = place.cid
            join orders on place.oid = orders.oid
            join contain on orders.oid = contain.oid
            join products on contain.pid = products.pid
where products.description like '%SATA%'
except
select customers.fullname
from customers join place on customers.cid = place.cid
            join orders on place.oid = orders.oid
            join contain on orders.oid = contain.oid
            join products on contain.pid = products.pid
where products.name = 'Router'
) as number_of_customers_that_have_SATA_but_no_routers;
```

	bought_SATA_no_router
▶	0

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

Explanation: The query inner joins tables merchants, sell, and products. The query retrieves data where the product category is Networking and the merchant company is HP. To apply, a 20% discount, the price is multiplied by .8, and displays the merchant, product id, product name, original price, and the discounted price.

```
SQL
select merchants.name,products.name as product_name, products.category,
sell.price as original_price, sell.price*.8 as discounted_price
from merchants join sell on merchants.mid = sell.mid
            join products on products.pid = sell.pid
where products.category = 'Networking' and merchants.name = "HP";
```

	name	product_name	category	original_price	discounted_price
	HP	Router	Networking	1034.46	827.568
	HP	Network Card	Networking	1154.68	923.744
	HP	Network Card	Networking	345.01	276.008
	HP	Network Card	Networking	262.20	209.760
	HP	Ethernet Adapter	Networking	1260.45	1008.360
	HP	Router	Networking	205.56	164.448
	HP	Router	Networking	1474.87	1179.896
	HP	Router	Networking	552.02	441.616
	HP	Router	Networking	100.95	80.760
	HP	Network Card	Networking	1179.01	943.208

Problem 5: What did Uriel Whitney order? (make sure to at least retrieve product names and prices)

Explanation: The query inner joins all tables and selects the product information and merchant information where the consumer is Uriel Whitney. Due to the limitations of the data as it does not track which company the consumer purchased from, the query displays the duplicated product and the merchants Uriel could have potentially purchased the product from. Note, too large to display all

SQL

```
select merchants.name, customers.fullname, products.pid, products.name as
product_name, sell.price
from merchants join sell on merchants.mid = sell.mid
      join products on products.pid = sell.pid
      join contain on contain.pid = products.pid
      join orders on orders.oid = contain.oid
      join place on place.oid = orders.oid
      join customers on place.cid = customers.cid
where customers.fullname = 'Uriel Whitney'
group by merchants.name, customers.fullname, products.pid,product_name,
sell.price;
```

	name	fullname	pid	product_name	price
▶	Acer	Uriel Whitney	1	Hard Drive	836.99
	Apple	Uriel Whitney	1	Hard Drive	1328.19
	Dell	Uriel Whitney	1	Hard Drive	970.45
	Lenovo	Uriel Whitney	1	Hard Drive	903.48
	Acer	Uriel Whitney	2	Monitor	1103.47
	Apple	Uriel Whitney	2	Monitor	786.31
	Dell	Uriel Whitney	2	Monitor	662.19
	Lenovo	Uriel Whitney	2	Monitor	375.59
	Acer	Uriel Whitney	3	Printer	310.83
	Apple	Uriel Whitney	3	Printer	994.35
	HP	Uriel Whitney	3	Printer	1408.80
	Dell	Uriel Whitney	3	Printer	1294.84
	Lenovo	Uriel Whitney	3	Printer	866.69
	Apple	Uriel Whitnev	4	Super Drive	540.56

However, if we are only interested in knowing the products Uriel purchased: we select the product where the customer is Uriel Whitney

SQL

```
select customers.fullname, products.name as product_name
  from merchants join sell on merchants.mid = sell.mid
                join products on products.pid = sell.pid
                join contain on contain.pid = products.pid
                join orders on orders.oid = contain.oid
                join place on place.oid = orders.oid
                join customers on place.cid = customers.cid
 where customers.fullname = 'Uriel Whitney'
 group by customers.fullname,product_name;
```

	fullname	product_name
▶	Uriel Whitney	Monitor
	Uriel Whitney	Router
	Uriel Whitney	Super Drive
	Uriel Whitney	Printer
	Uriel Whitney	Network Card
	Uriel Whitney	Hard Drive
	Uriel Whitney	Laptop
	Uriel Whitney	Ethernet Adapter
	Uriel Whitney	Desktop

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

Explanation: Due to the limitations of the dataset as it does not contain what each company has sold and thus the product quantity sold, we assume revenue is the potential potential revenue as the cost of a product and its quantity. This is calculated as the sum of price and quantity and rounds the output to two decimal places. The query inner joins all the tables and filters the date of the order date to the year and groups the revenue by the merchant and the year the order was placed.

SQL

```
select merchants.name, format(sum(sell.price * sell.quantity_available),2) as
total_price, year(place.order_date)
from merchants join sell on merchants.mid = sell.mid
      join products on products.pid = sell.pid
      join contain on contain.pid = products.pid
      join orders on orders.oid = contain.oid
      join place on place.oid = orders.oid
      join customers on place.cid = customers.cid
group by merchants.name, year(place.order_date)
order by merchants.name,year(place.order_date);
```

	name	total_price	year(place.order_date)
	Acer	2,486,031.24	2011
	Acer	923,729.49	2016
	Acer	3,300,620.55	2017
	Acer	4,778,659.74	2018
	Acer	3,540,650.10	2019
▶	Acer	3,187,866.90	2020
	Apple	2,916,722.76	2011
	Apple	1,228,207.14	2016
	Apple	3,215,138.79	2017
	Apple	4,993,889.31	2018
	Apple	3,934,252.71	2019
	Apple	3,641,894.88	2020
	Dell	4,626,686.97	2011
	Dell	1,877,052.42	2016
	Dell	4,568,382.84	2017
	Dell	7,803,182.88	2018
	Dell	5,390,052.09	2019
	Dell	5,210,435.58	2020
	HP	2,620,641.30	2011
	HP	1,126,642.35	2016
	HP	2,814,504.09	2017
	HP	3,845,294.85	2018
	HP	3,333,190.38	2019
	HP	3,493,554.81	2020
	Lenovo	3,706,655.52	2011
	Lenovo	1,451,710.68	2016

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

Explanation: Note, we assume the same assumptions for revenue. The query is derived from the query from problem 6, in addition to ordering the revenue in descending order to retrieve the top value as the revenue and not formatting the price as it converts it to a string .

SQL

```
select merchants.name, sum(sell.price * sell.quantity_available) as
total_price, year(place.order_date)
from merchants join sell on merchants.mid = sell.mid
                join products on products.pid = sell.pid
                join contain on contain.pid = products.pid
                join orders on orders.oid = contain.oid
                join place on place.oid = orders.oid
                join customers on place.cid = customers.cid
group by merchants.name, year(place.order_date)
order by total_price desc
limit 1;
```

	name	total_price	year(place.order_date)
▶	Dell	7803182.88	2018

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

Explanation: The query selects from table orders and calculates the rounded average shipping cost grouped by the shipping method. The query order by ascending order by default, and retrieves the top cost as the lowest cost

SQL

```
select orders.shipping_method, round(avg(orders.shipping_cost),2)
from orders
group by orders.shipping_method
order by avg(orders.shipping_cost)
limit 1;
```

	shipping_method	round(avg(orders.shipping_cost),2)
▶	USPS	7.46

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

Explanation: Note, since we do not know what each company sold, we instead consider the potential amount the company may have sold from the products that are in the contain table. The query has a subquery to find the max amount sold from each company and the main query selects the max amount spent for each company and the category.

SQL

```
With company_category as (
  select merchants.name as company, products.category as category,
  sum(sell.price * sell.quantity_available) as revenue
  from merchants join sell on merchants.mid = sell.mid
  join products on products.pid = sell.pid
  join contain on contain.pid = products.pid
  join orders on orders.oid = contain.oid
  join place on place.oid = orders.oid
  join customers on place.cid = customers.cid
group by merchants.name, products.category
),
Max_revenue as (
  select company ,max(revenue) as max_revenue
from company_category
group by company)
```



```

select cc.company, cc.category, cc.revenue
from company_category cc
JOIN Max_revenue mr on cc.company = mr.company and cc.revenue = mr.max_revenue
ORDER BY cc.revenue;

```

	company	category	revenue
▶	HP	Peripheral	7516556.10
	Apple	Peripheral	10283042.64
	Lenovo	Peripheral	13804501.89
	Acer	Peripheral	13906726.26
	Dell	Peripheral	15928539.42

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

Explanation: The query uses a CTE where it joins all the tables. There are two subqueries, highest and lowest, each having their max and lowest spender and price respectively. In each query, the CTE has their own inner subquery company_customers to find the max/min price and spender. Finally, the main query joins the two subquery using the common variable company.

```

SQL ▼
With company_customers as (
    select merchants.name as company, customers.fullname as customer,
    sum(sell.price) as amount_spent
    from merchants join sell on merchants.mid = sell.mid
        join products on products.pid = sell.pid
        join contain on contain.pid = products.pid
        join orders on orders.oid = contain.oid
        join place on place.oid = orders.oid
        join customers on place.cid = customers.cid
    group by merchants.name, customers.fullname
),
Highest as (
    select c1.company, amount_spent as Highest_max_price, c1.customer as h_customer
    from company_customers c1
    where amount_spent =
    (select max(amount_spent)

```

```

from company_customers c2
where c2.company = c1.company)
),
Lowest as (
select c1.company, amount_spent as lowest_min_price, c1.customer as l_customer
from company_customers c1
where amount_spent = (select min(amount_spent)
from company_customers c3
where c3.company = c1.company)
)
select * from Highest, Lowest
where Highest.company = Lowest.company;

```

	company	Highest_max_price	h_customer	company	lowest_min_price	l_customer
►	Lenovo	249090.78	Haviva Stewart	Lenovo	101846.73	Inez Long
	Apple	253653.33	Clementine Travis	Apple	96753.30	Inez Long
	HP	199884.18	Clementine Travis	HP	78188.67	Inez Long
	Dell	256834.65	Clementine Travis	Dell	93407.22	Inez Long
	Acer	225690.87	Dean Heath	Acer	95703.06	Inez Long