

Title: DB Assignment 3

Name: Aidan Elm

Date: 2024-10-10

Creation of Tables and ER Diagram

-- create merchants table

```
create table merchants (  
    mid int primary key,  
    name varchar(100) not null,  
    city varchar(100),  
    state varchar(100)  
);
```

-- create products table

```
create table products (  
    pid int primary key,  
    name varchar(100) not null,  
    category varchar(100),  
    description varchar(1000),  
    constraint check_product_name check (name in ('printer', 'ethernet adapter', 'desktop',  
'hard drive', 'laptop', 'router', 'network card', 'super drive', 'monitor')),  
    constraint check_product_category check (category in ('peripheral', 'networking',  
'computer'))  
);
```

-- create sell table

```
create table sell (  
    mid int,  
    pid int,  
    price decimal(10, 2) not null,  
    quantity_available int not null,  
    primary key (mid, pid),  
    foreign key (mid) references merchants(mid),  
    foreign key (pid) references products(pid),
```

```
constraint check_price check (price between 0 and 100000),  
constraint check_quantity check (quantity_available between 0 and 1000)  
);
```

```
-- create orders table  
create table orders (  
    oid int primary key,  
    shipping_method varchar(100),  
    shipping_cost decimal(10, 2),  
    constraint check_shipping_method check (shipping_method in ('ups', 'fedex', 'usps')),  
    constraint check_shipping_cost check (shipping_cost between 0 and 500)  
);
```

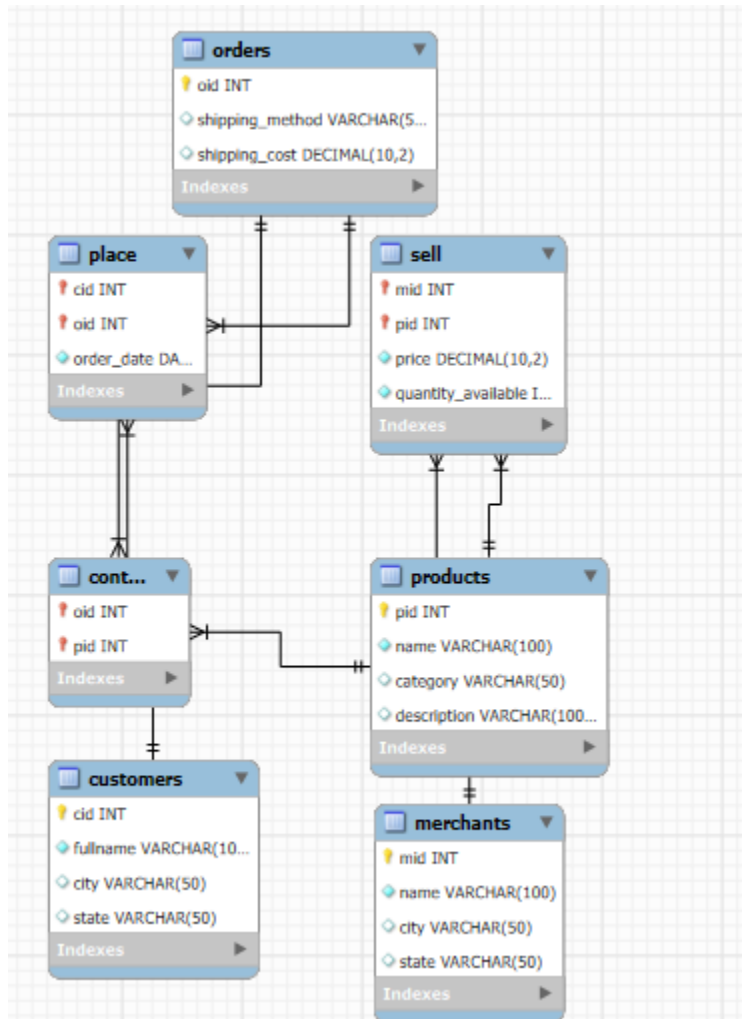
```
-- create contain table  
create table contain (  
    oid int,  
    pid int,  
    primary key (oid, pid),  
    foreign key (oid) references orders(oid),  
    foreign key (pid) references products(pid)  
);
```

```
-- create customers table  
create table customers (  
    cid int primary key,  
    fullname varchar(100) not null,  
    city varchar(100),  
    state varchar(100)  
);
```

```

-- create place table
create table place (
  cid int,
  oid int,
  order_date date not null,
  primary key (cid, oid),
  foreign key (cid) references customers(cid),
  foreign key (oid) references orders(oid),
  constraint check_order_date check (order_date = date(order_date))
);

```



Question 1

```
select p.name, m.name
from products p
join sell s on p.pid = s.pid
join merchants m on s.mid = m.mid
where s.quantity_available = 0;
```

In this query, we join sellers to products and then merchants to sellers. We select the names of products and merchants, only where the quantity available is zero.

```
--
78  -- ** Problem 1 **
79  • select p.name, m.name
80  from products p
81  join sell s on p.pid = s.pid
82  join merchants m on s.mid = m.mid
83  where s.quantity_available = 0;
84
85  -- ** Problem 2 **
```

Result Grid		Filter Rows:	Export:	Wrap Cell Center
	name	name		
▶	Router	Acer		
	Network Card	Acer		
	Printer	Apple		
	Router	Apple		
	Laptop	HP		
	Router	HP		
	Super Drive	HP		
	Router	Dell		

Result 5 ×

Question 2

```
select p.name, p.description
from products p
left join sell s on p.pid = s.pid
where s.pid is null;
```

The same as above, except we are not joining merchants and are instead selecting product descriptions. Because we are looking for null values, we must use a left join instead of a regular join.

```
84
85 -- ** Problem 2 **
86 • select p.name, p.description
87 from products p
88 left join sell s on p.pid = s.pid
89 where s.pid is null;
90
91 -- ** Problem 3 **
```

Result Grid		Filter Rows:	Export:	Wrap
	name	description		
▶	Super Drive	External CD/DVD/RW		
	Super Drive	UInternal CD/DVD/RW		

Question 3

```
select count(distinct c.cid)
from customers c
join place pl on c.cid = pl.cid
join orders o on pl.oid = o.oid
join contain ct on o.oid = ct.oid
join products p on ct.pid = p.pid
where p.description like '%sata%'
and c.cid not in (
    select c.cid
    from customers c
    join place pl on c.cid = pl.cid
    join orders o on pl.oid = o.oid
```

```

join contain ct on o.oid = ct.oid
join products p on ct.pid = p.pid
where p.description like '%router%'
);

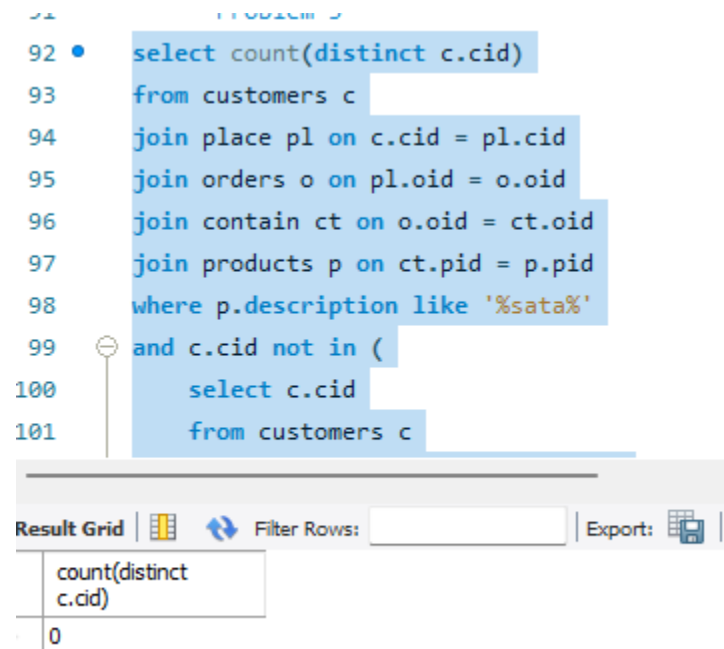
```

For this problem, to join products to customers, we have to go through contain, order, and place. We then search the description to match “SATA” and use a subquery to exclude descriptions with “router.”

```

92 • select count(distinct c.cid)
93     from customers c
94     join place pl on c.cid = pl.cid
95     join orders o on pl.oid = o.oid
96     join contain ct on o.oid = ct.oid
97     join products p on ct.pid = p.pid
98     where p.description like '%sata%'
99     and c.cid not in (
100         select c.cid
101         from customers c

```



The screenshot shows a SQL query editor with a query that counts distinct customer IDs. The query joins customers, place, orders, contain, and products tables, filtering for 'sata' in the product description and excluding customers who have ordered 'router' products. Below the editor is a 'Result Grid' showing the query result.

count(distinct c.cid)
0

Question 4

```

select p.name as product_name, s.price * 0.8
from products p
join sell s on p.pid = s.pid
join merchants m on s.mid = m.mid
where m.name like '%hp%' and p.category = 'networking';



```

To get this result, we join sell to products and then products to sell. We discount the price by 20% and include only products from HP in the networking category.

```

108
109 -- ** Problem 4 **
110 • select p.name as product_name, s.price * 0.8
111 from products p
112 join sell s on p.pid = s.pid
113 join merchants m on s.mid = m.mid
114 where m.name like '%hp%' and p.category = 'networking';

```

Result Grid		
Filter Rows: <input type="text"/>		
Export:  Wrap Cell Content: 		
product_name	s.price * 0.8	
Router	827.568	
Network Card	923.744	
Network Card	276.008	
Network Card	209.760	
Ethernet Adapter	1008.360	
Router	164.448	
Router	1179.896	

Question 5

```

select p.name as product_name, s.price
from customers c
join place pl on c.cid = pl.cid
join orders o on pl.oid = o.oid
join contain ct on o.oid = ct.oid
join products p on ct.pid = p.pid
join sell s on p.pid = s.pid
join merchants m on s.mid = m.mid
where c.fullname = 'uriel whitney' and m.name like '%acer%';

```

Similar to above, just with a couple of extra joins. We select only the rows where the full name of the customer is Uriel Whitne and the merchant name contains “Acer.”


```

116 -- ** Problem 5 **
117 • select p.name as product_name, s.price
118 from customers c
119 join place pl on c.cid = pl.cid
120 join orders o on pl.oid = o.oid
121 join contain ct on o.oid = ct.oid
122 join products p on ct.pid = p.pid

```

Result Grid		Filter Rows:	Export:	Wra
	product_name	price		
▶	Super Drive	356.13		
	Network Card	130.43		
	Hard Drive	836.99		
	Printer	310.83		
	Printer	1345.37		

Result 17 x

Question 6

```

select merchants.name, year(place.order_date), count(distinct orders.oid)
from merchants
join sell on merchants.mid = sell.mid
join contain on sell.pid = contain.pid
join orders on contain.oid = orders.oid
join place on orders.oid = place.oid
group by merchants.name, year(place.order_date)

```

For this one, we join place -> orders -> contain -> sell -> merchants. We select the name of the merchants, the year from the order_date in place, and the count of distinct orders for that given year.

Limit to 1000 rows

```

126 -- ** Problem 6 **
127 • select merchants.name, year(place.order_date), count(distinct orders.oid)
128 from merchants
129 join sell on merchants.mid = sell.mid
130 join contain on sell.pid = contain.pid
131 join orders on contain.oid = orders.oid
132 join place on orders.oid = place.oid
133 group by merchants.name, year(place.order_date)
134 order by merchants.name, year(place.order_date);
135

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	name	year(place.order_date)	count(distinct orders.oid)
▶	Acer	2011	65
	Acer	2016	24
	Acer	2017	72
	Acer	2018	97
	Acer	2019	80
	Acer	2020	68

Result 1 x

Question 7

```

select merchants.name, year(place.order_date), count(distinct orders.oid)
from merchants
join sell on merchants.mid = sell.mid
join contain on sell.pid = contain.pid
join orders on contain.oid = orders.oid
join place on orders.oid = place.oid
group by merchants.name, year(place.order_date)
order by count(distinct orders.oid) desc
limit 1;

```

The same as above, except this time we are counting the number, ordering descending, and limiting to one.

The screenshot shows a SQL IDE window with a query editor and a result grid. The query editor contains the following SQL code:

```
137 • select merchants.name, year(place.order_date), count(distinct orders.oid)
138 from merchants
139 join sell on merchants.mid = sell.mid
140 join contain on sell.pid = contain.pid
141 join orders on contain.oid = orders.oid
142 join place on orders.oid = place.oid
```

The result grid shows the following data:

	name	year(place.order_date)	count(distinct orders.oid)
▶	Acer	2018	97



Question 8

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

For this problem, we only need to query one table. We can get the average of shipping cost by method, order ascending, and limit to one.

```
-- ** Problem 8 **
```

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

ult Grid |  Filter Rows: | Export:  | Wrap C

shipping_method	avg(shipping_cost)
JSPS	7.455761

Question 9



```
select merchants.name, products.category, count(distinct orders.oid)
from merchants
join sell on merchants.mid = sell.mid
join contain on sell.pid = contain.pid
join orders on contain.oid = orders.oid
join products on sell.pid = products.pid
group by merchants.name, products.category
order by merchants.name, count(distinct orders.oid) desc;
```

Similar to Question 7, except we are joining products to get the category and, from there, getting the categories with the most sales.

```

153
154 -- ** Problem 9 **
155 • select merchants.name, products.category, count(distinct orders.oid)
156 from merchants
157 join sell on merchants.mid = sell.mid
158 join contain on sell.pid = contain.pid
159 join orders on contain.oid = orders.oid

```

Result Grid			
Filter Rows: <input type="text"/>			
Export:  Wrap Cell Content: 			
	name	category	count(distinct orders.oid)
▶	Acer	Peripheral	414
	Acer	Networking	367
	Acer	Computer	191
	Apple	Peripheral	432
	Apple	Networking	355
	Apple	Computer	191
	Dell	Peripheral	429
	Dell	Networking	388

Result 6 x

Question 10

```

select merchants.name, customers.fullname, count(distinct orders.oid)
from merchants
join sell on merchants.mid = sell.mid
join contain on sell.pid = contain.pid
join orders on contain.oid = orders.oid
join place on orders.oid = place.oid
join customers on place.cid = customers.cid
group by merchants.name, customers.fullname
order by merchants.name, count(distinct orders.oid);

```

Similar to above, except this time we are joining customers to place to get the number of orders per merchant per customer.

Limit to 1000 rows


```

156 from merchants
157 join sell on merchants.mid = sell.mid
158 join contain on sell.pid = contain.pid
159 join orders on contain.oid = orders.oid
160 join products on sell.pid = products.pid
161 group by merchants.name, products.category
162 order by merchants.name, count(distinct orders.oid) desc;
163
164 -- ** Problem 10 **
165 • select merchants.name, customers.fullname, count(distinct orders.oid)

```

Result Grid

Filter Rows:

Export:  Wrap Cell Content: ☐

	name	fullname	count(distinct orders.oid)
▶	Acer	Inez Long	15
	Acer	Charde Mcgee	15
	Acer	Wynne Mckinney	15
	Acer	Jerry Roberts	17
	Acer	Gray Perkins	18
	Acer	Abraham Sears	19
	Acer	Demetrius Garcia	20
	Acer	Brynn Haney	21

Result 8 x