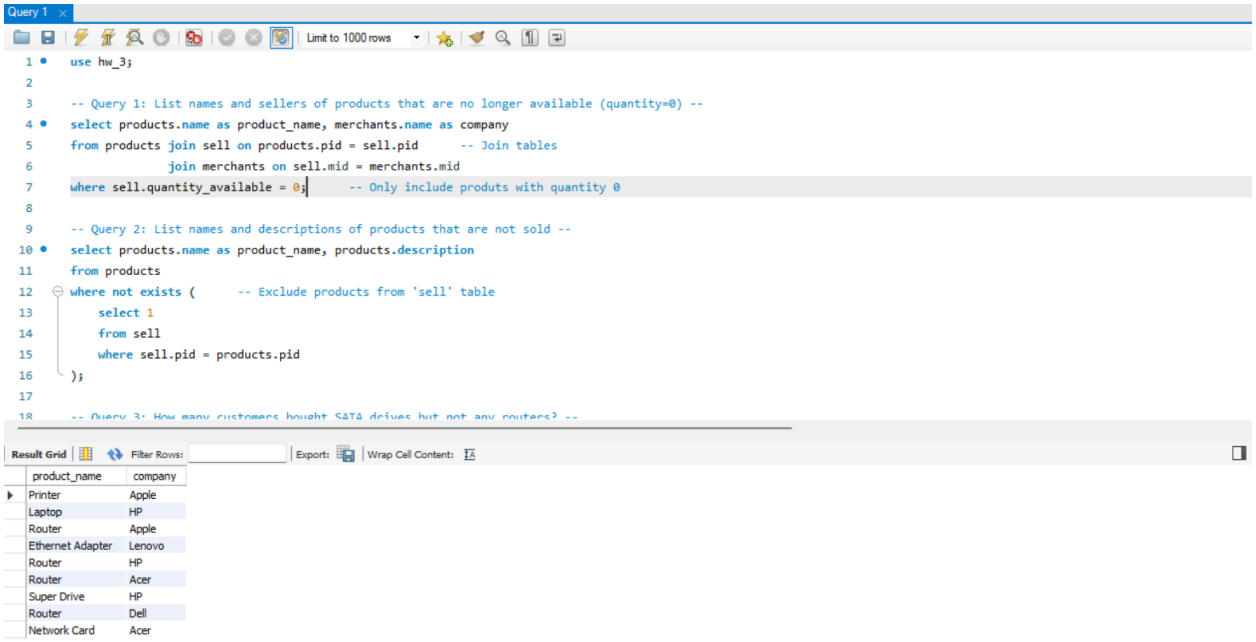
DB Assignment 3
Graeme Glavin
10/21/2025

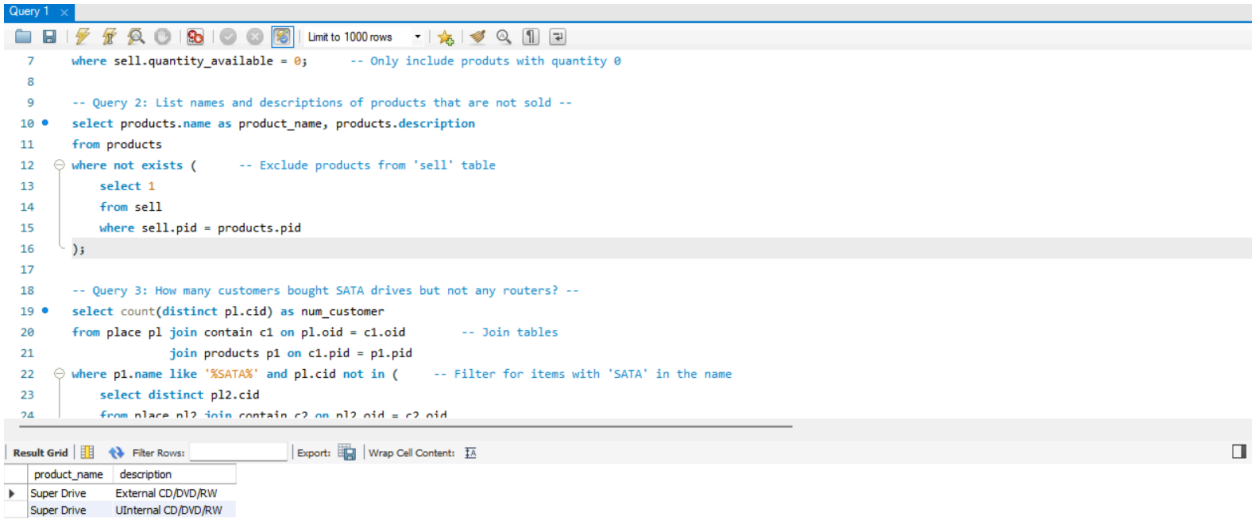
Query 1: List names and sellers of products that are no longer available (quantity=0)



select products.name as product_name, merchants.name as company
from products join sell on products.pid = sell.pid -- Join tables
join merchants on sell.mid = merchants.mid
where sell.quantity_available = 0; -- Only include products with quantity 0

Finds all products that are currently out of stock (quantity = 0) and lists their names along with the merchants who sell them.

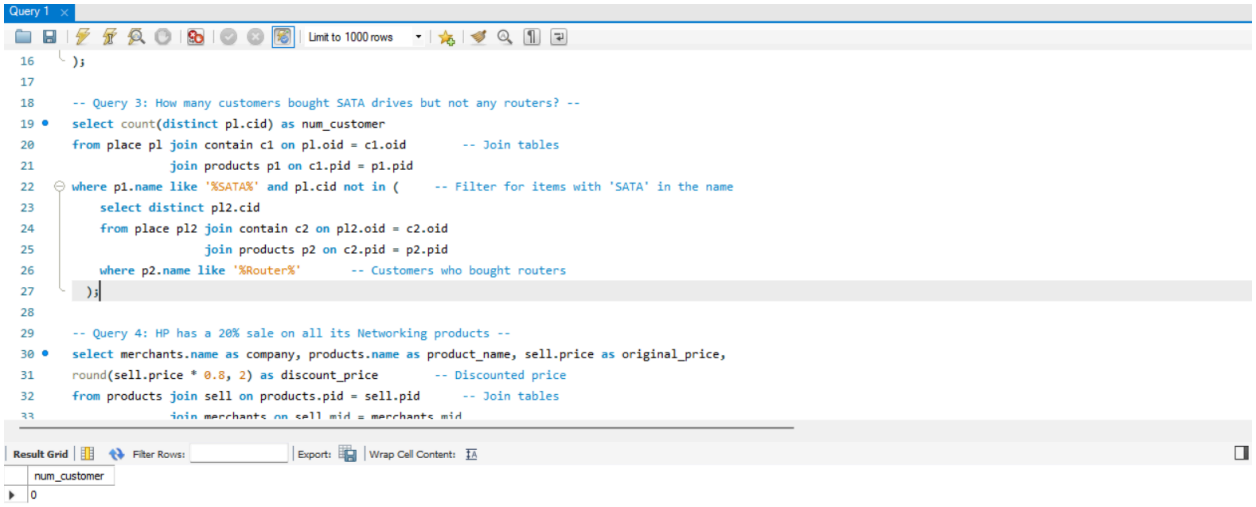
Query 2: List names and descriptions of products that are not sold




```
select products.name as product_name, products.description
from products
where not exists (                  -- Exclude products from 'sell' table
    select 1
    from sell
    where sell.pid = products.pid
);
```

Lists all products that no merchant sells, or products that exist in the products table but don't appear in the sell table.

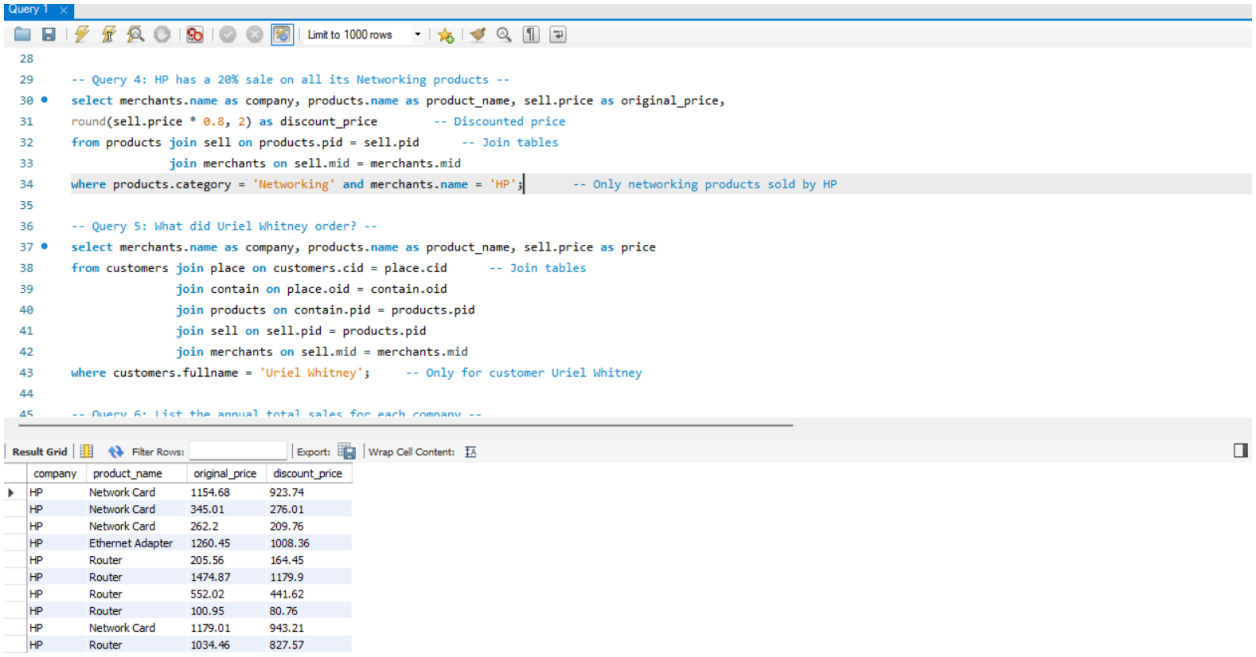
Query 3: How many customers bought SATA drives but not any routers?



select count(distinct p1.cid) as num_customer
from place p1 join contain c1 on p1.oid = c1.oid -- Join tables
join products p1 on c1.pid = p1.pid
where p1.name like '%SATA%' and p1.cid not in (-- Filter for items with 'SATA' in the name
select distinct p12.cid
from place p12 join contain c2 on p12.oid = c2.oid
join products p2 on c2.pid = p2.pid
where p2.name like '%Router%' -- Customers who bought routers
);

Counts how many customers bought at least one SATA product but never purchased any routers.

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



```
28
29    -- Query 4: HP has a 20% sale on all its Networking products --
30    select merchants.name as company, products.name as product_name, sell.price as original_price,
31    round(sell.price * 0.8, 2) as discount_price        -- Discounted price
32    from products join sell on products.pid = sell.pid      -- Join tables
33                  join merchants on sell.mid = merchants.mid
34    where products.category = 'Networking' and merchants.name = 'HP';       -- Only networking products sold by HP
35
36    -- Query 5: What did Uriel Whitney order? --
37    select merchants.name as company, products.name as product_name, sell.price as price
38    from customers join place on customers.cid = place.cid      -- Join tables
39                   join contain on place.oid = contain.oid
40                   join products on contain.pid = products.pid
41                   join sell on sell.pid = products.pid
42                   join merchants on sell.mid = merchants.mid
43    where customers.fullname = 'Uriel Whitney';      -- Only for customer Uriel Whitney
44
45    -- Query 6: List the annual total sales for each company --
```

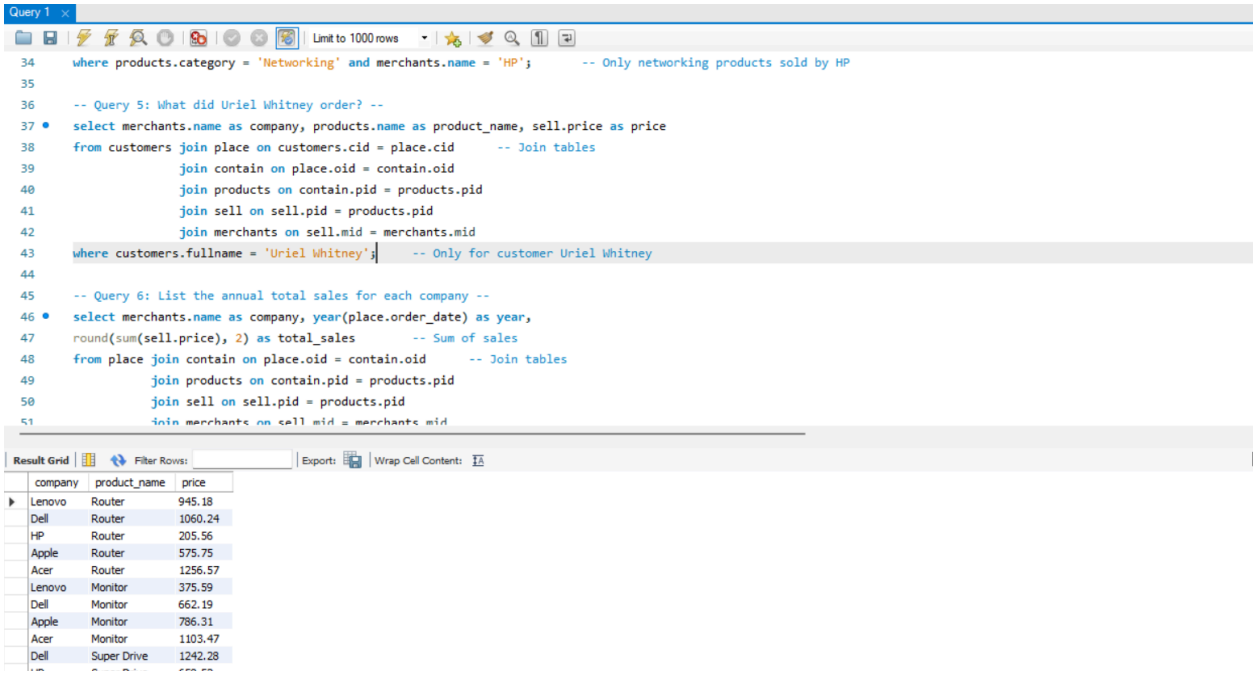
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

company	product_name	original_price	discount_price
HP	Network Card	1154.68	923.74
HP	Network Card	345.01	276.01
HP	Network Card	262.2	209.76
HP	Ethernet Adapter	1260.45	1008.36
HP	Router	205.56	164.45
HP	Router	1474.87	1179.9
HP	Router	552.02	441.62
HP	Router	100.95	80.76
HP	Network Card	1179.01	943.21
HP	Router	1034.46	827.57

select merchants.name as company, products.name as product_name, sell.price as original_price,
*round(sell.price * 0.8, 2) as discount_price -- Discounted price*
from products join sell on products.pid = sell.pid -- Join tables
join merchants on sell.mid = merchants.mid
where products.category = 'Networking' and merchants.name = 'HP'; -- Only networking products sold by HP

Applies a 20% discount to all HP's Networking products and shows the items and their new prices.

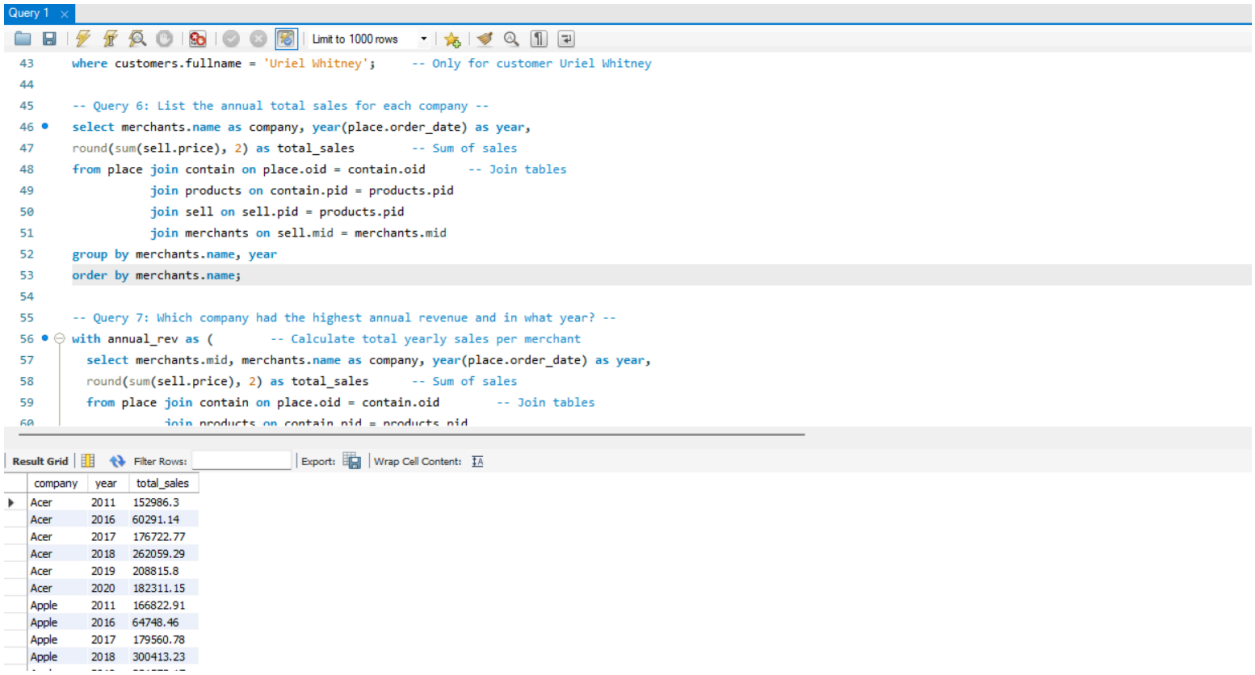
Query 5: What did Uriel Whitney order?



select merchants.name as company, products.name as product_name, sell.price as price
from customers join place on customers.cid = place.cid -- Join tables
join contain on place.oid = contain.oid
join products on contain.pid = products.pid
join sell on sell.pid = products.pid
join merchants on sell.mid = merchants.mid
where customers.fullname = 'Uriel Whitney'; -- Only for customer Uriel Whitney

Lists every product ordered by customer *Uriel Whitney*, including each product's name, price, and company.

Query 6: List the annual total sales for each company



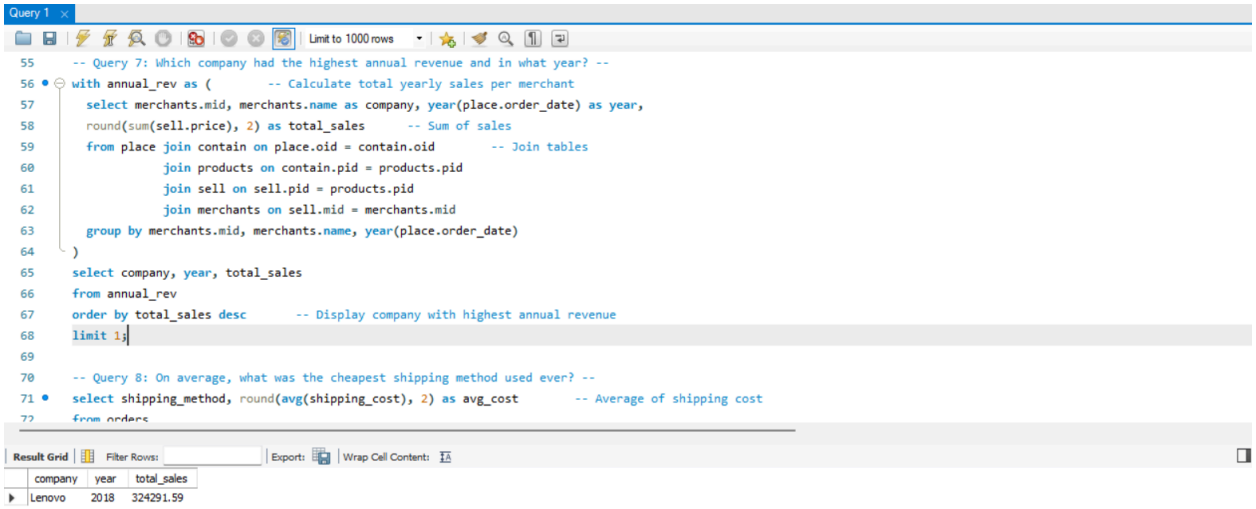
```
where customers.fullname = 'Uriel Whitney';     -- Only for customer Uriel Whitney

-- Query 6: List the annual total sales for each company --
select merchants.name as company, year(place.order_date) as year,
round(sum(sell.price), 2) as total_sales        -- Sum of sales
from place join contain on place.oid = contain.oid      -- Join tables
           join products on contain.pid = products.pid
           join sell on sell.pid = products.pid
           join merchants on sell.mid = merchants.mid
group by merchants.name, year
order by merchants.name;

-- Query 7: Which company had the highest annual revenue and in what year? --
with annual_rev as (        -- Calculate total yearly sales per merchant
  select merchants.mid, merchants.name as company, year(place.order_date) as year,
  round(sum(sell.price), 2) as total_sales      -- Sum of sales
  from place join contain on place.oid = contain.oid        -- Join tables
```

company	year	total_sales
Acer	2011	152986.3
Acer	2016	60291.14
Acer	2017	176722.77
Acer	2018	262059.29
Acer	2019	208815.8
Acer	2020	182311.15
Apple	2011	166822.91
Apple	2016	64748.46
Apple	2017	179560.78
Apple	2018	300413.23

select merchants.name as company, year(place.order_date) as year,
round(sum(sell.price), 2) as total_sales -- Sum of sales
from place join contain on place.oid = contain.oid -- Join tables
join products on contain.pid = products.pid
join sell on sell.pid = products.pid
join merchants on sell.mid = merchants.mid
group by merchants.name, year
order by merchants.name;

Calculates the total yearly revenue for each merchant by summing product prices in all orders, grouped by year.

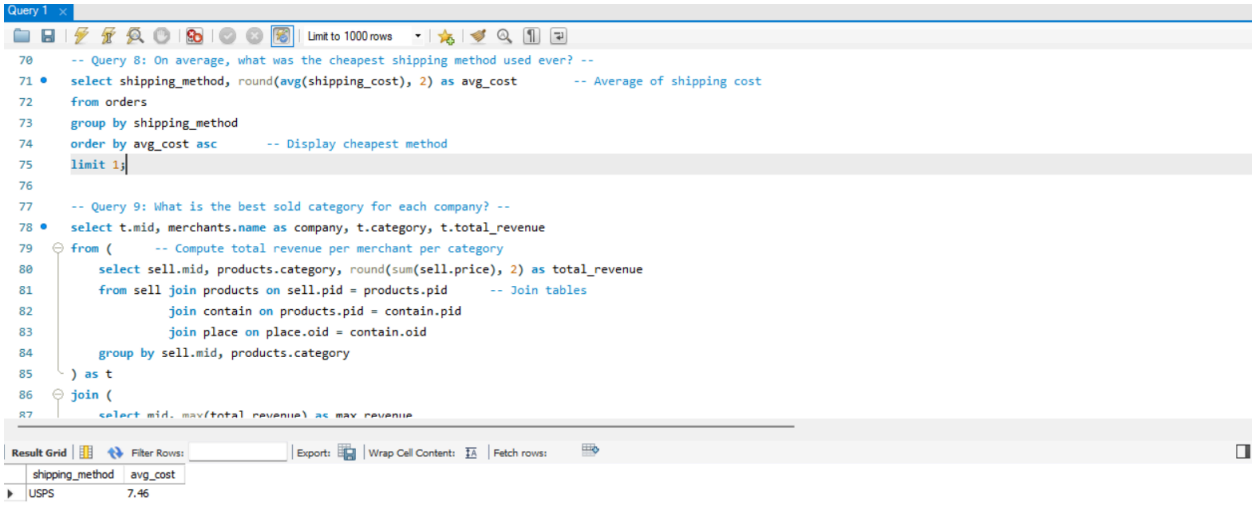
Query 7: Which company had the highest annual revenue and in what year?




```
with annual_rev as (            -- Calculate total yearly sales per merchant
  select merchants.mid, merchants.name as company, year(place.order_date) as year,
  round(sum(sell.price), 2) as total_sales              -- Sum of sales
  from place join contain on place.oid = contain.oid          -- Join tables
                          join products on contain.pid = products.pid
                          join sell on sell.pid = products.pid
                          join merchants on sell.mid = merchants.mid
  group by merchants.mid, merchants.name, year(place.order_date)
)
select company, year, total_sales
from annual_rev
order by total_sales desc               -- Display company with highest annual revenue
limit 1;
```

Finds which merchant had the highest annual revenue and in which year it occurred.

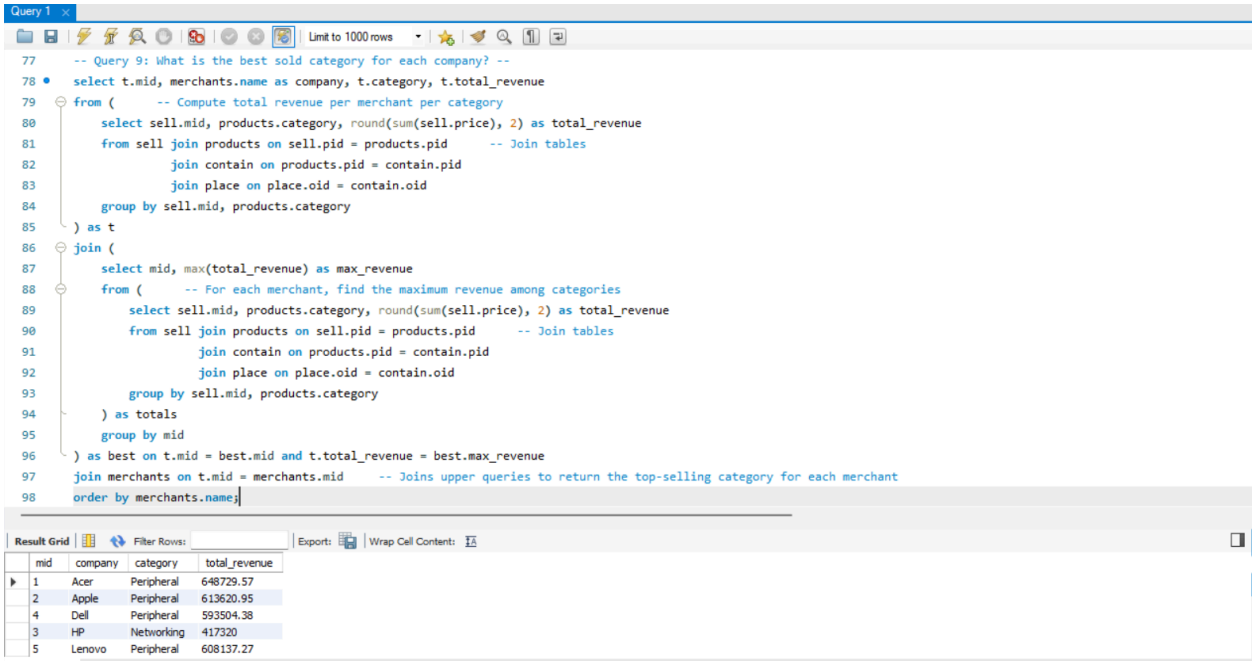
Query 8: On average, what was the cheapest shipping method used ever?



select shipping_method, round(avg(shipping_cost), 2) as avg_cost -- Average of shipping cost
from orders
group by shipping_method
order by avg_cost asc -- Display cheapest method
limit 1;

Calculates the average cost for each shipping method and returns the one with the lowest average.

Query 9: What is the best sold category for each company?



```
-- Query 9: What is the best sold category for each company? --
select t.mid, merchants.name as company, t.category, t.total_revenue
from (      -- Compute total revenue per merchant per category
    select sell.mid, products.category, round(sum(sell.price), 2) as total_revenue
    from sell join products on sell.pid = products.pid      -- Join tables
              join contain on products.pid = contain.pid
              join place on place.oid = contain.oid
    group by sell.mid, products.category
) as t
join (
    select mid, max(total_revenue) as max_revenue
    from (      -- For each merchant, find the maximum revenue among categories
        select sell.mid, products.category, round(sum(sell.price), 2) as total_revenue
        from sell join products on sell.pid = products.pid      -- Join tables
                  join contain on products.pid = contain.pid
                  join place on place.oid = contain.oid
        group by sell.mid, products.category
    ) as totals
    group by mid
) as best on t.mid = best.mid and t.total_revenue = best.max_revenue
join merchants on t.mid = merchants.mid     -- Joins upper queries to return the top-selling category for each merchant
order by merchants.name;
```

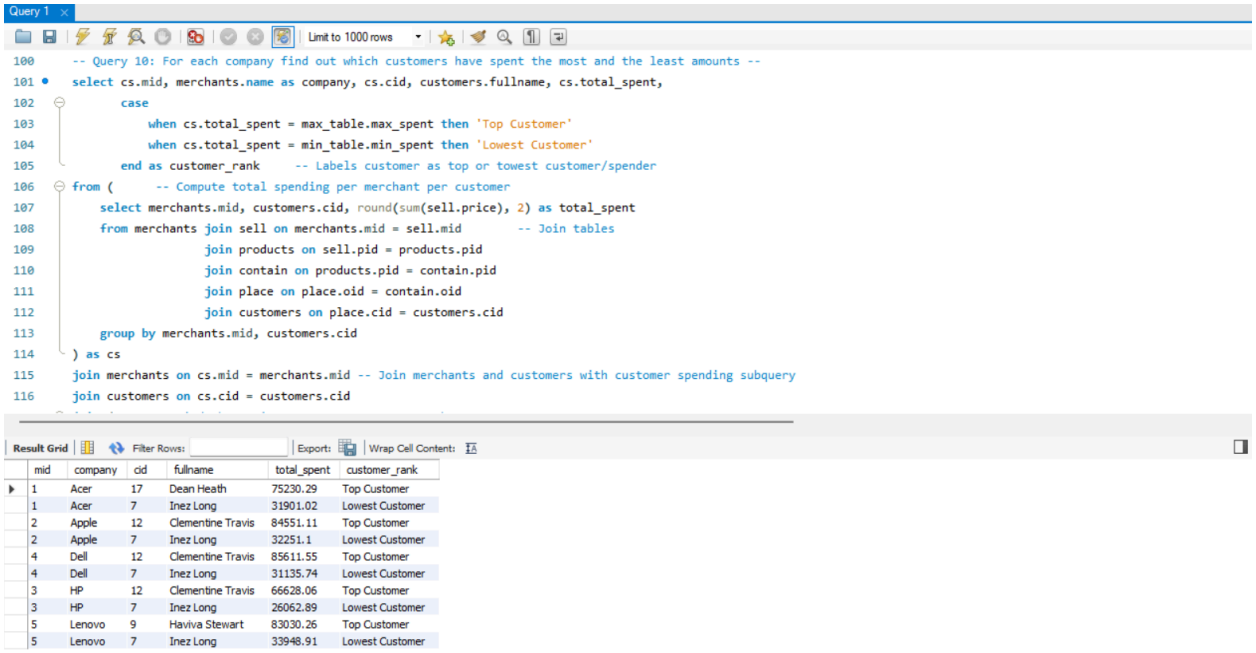
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

mid	company	category	total_revenue
1	Acer	Peripheral	648729.57
2	Apple	Peripheral	613620.95
4	Dell	Peripheral	593504.38
3	HP	Networking	417320
5	Lenovo	Peripheral	608137.27

select t.mid, merchants.name as company, t.category, t.total_revenue
from (-- Compute total revenue per merchant per category
select sell.mid, products.category, round(sum(sell.price), 2) as total_revenue
from sell join products on sell.pid = products.pid -- Join tables
join contain on products.pid = contain.pid
join place on place.oid = contain.oid
group by sell.mid, products.category
) as t
join (
select mid, max(total_revenue) as max_revenue
from (-- For each merchant, find the maximum revenue among categories
select sell.mid, products.category, round(sum(sell.price), 2) as total_revenue
from sell join products on sell.pid = products.pid -- Join tables
join contain on products.pid = contain.pid
join place on place.oid = contain.oid
group by sell.mid, products.category
) as totals
group by mid
) as best on t.mid = best.mid and t.total_revenue = best.max_revenue
join merchants on t.mid = merchants.mid -- Joins upper queries to return the top-selling category for each merchant
order by merchants.name;

Finds, for each merchant, the product category that generated the highest total revenue.

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



```
-- Query 10: For each company find out which customers have spent the most and the least amounts --
select cs.mid, merchants.name as company, cs.cid, customers.fullname, cs.total_spent,
      case
          when cs.total_spent = max_table.max_spent then 'Top Customer'
          when cs.total_spent = min_table.min_spent then 'Lowest Customer'
      end as customer_rank     -- Labels customer as top or towest customer/spender
from (      -- Compute total spending per merchant per customer
    select merchants.mid, customers.cid, round(sum(sell.price), 2) as total_spent
    from merchants join sell on merchants.mid = sell.mid        -- Join tables
                   join products on sell.pid = products.pid
                   join contain on products.pid = contain.pid
                   join place on place.oid = contain.oid
                   join customers on place.cid = customers.cid
    group by merchants.mid, customers.cid
) as cs
join merchants on cs.mid = merchants.mid -- Join merchants and customers with customer spending subquery
join customers on cs.cid = customers.cid
```

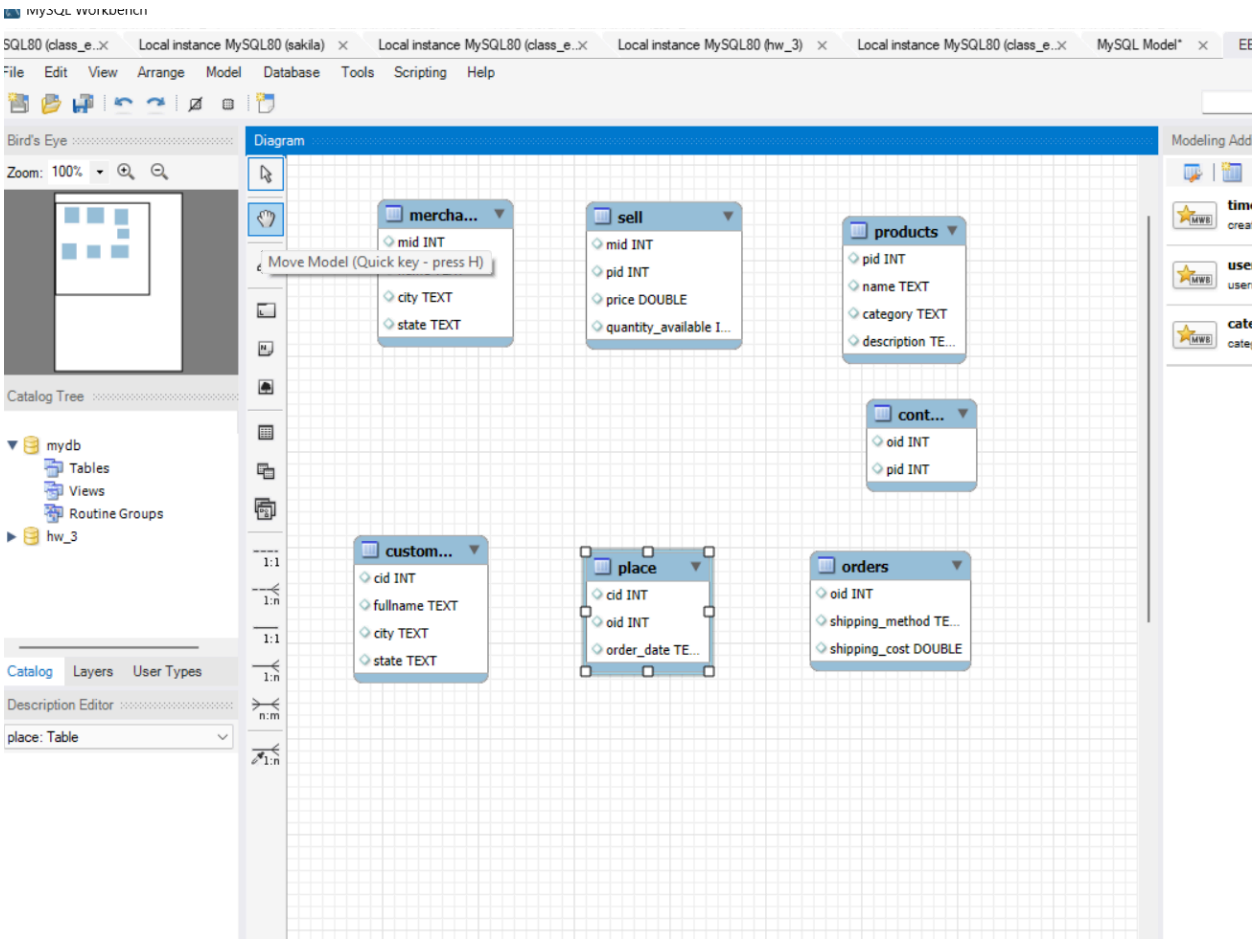
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

mid	company	cid	fullname	total_spent	customer_rank
1	Acer	17	Dean Heath	75230.29	Top Customer
1	Acer	7	Inez Long	31901.02	Lowest Customer
2	Apple	12	Clementine Travis	84551.11	Top Customer
2	Apple	7	Inez Long	32251.1	Lowest Customer
4	Dell	12	Clementine Travis	85611.55	Top Customer
4	Dell	7	Inez Long	31135.74	Lowest Customer
3	HP	12	Clementine Travis	66628.06	Top Customer
3	HP	7	Inez Long	26062.89	Lowest Customer
5	Lenovo	9	Haviva Stewart	83030.26	Top Customer
5	Lenovo	7	Inez Long	33948.91	Lowest Customer

```
select cs.mid, merchants.name as company, cs.cid, customers.fullname, cs.total_spent,
      case
          when cs.total_spent = max_table.max_spent then 'Top Customer'
          when cs.total_spent = min_table.min_spent then 'Lowest Customer'
      end as customer_rank              -- Labels customer as top or towest customer/spender
from (            -- Compute total spending per merchant per customer
    select merchants.mid, customers.cid, round(sum(sell.price), 2) as total_spent
    from merchants join sell on merchants.mid = sell.mid              -- Join tables
                                  join products on sell.pid = products.pid
                                  join contain on products.pid = contain.pid
                                  join place on place.oid = contain.oid
                                  join customers on place.cid = customers.cid
    group by merchants.mid, customers.cid
) as cs
join merchants on cs.mid = merchants.mid -- Join merchants and customers with customer
spending subquery
join customers on cs.cid = customers.cid
join (            -- Find the maximum amount spent per merchant
    select mid, max(total_spent) as max_spent
    from (
        select merchants.mid, customers.cid, round(sum(sell.price), 2) as total_spent
        from merchants join sell on merchants.mid = sell.mid          -- Join tables
                                      join products on sell.pid = products.pid
```

```
                                        join contain on products.pid = contain.pid
                                        join place on place.oid = contain.oid
                                        join customers on place.cid = customers.cid
        group by merchants.mid, customers.cid
    ) as totals
    group by mid
) as max_table on cs.mid = max_table.mid
join (              -- Find the minimum amount spent per merchant
    select mid, min(total_spent) as min_spent
    from (
        select merchants.mid, customers.cid, round(sum(sell.price), 2) as total_spent
        from merchants join sell on merchants.mid = sell.mid            -- Join tables
                                        join products on sell.pid = products.pid
                                        join contain on products.pid = contain.pid
                                        join place on place.oid = contain.oid
                                        join customers on place.cid = customers.cid
        group by merchants.mid, customers.cid
    ) as totals
    group by mid
) as min_table on cs.mid = min_table.mid
where cs.total_spent = max_table.max_spent or cs.total_spent = min_table.min_spent
order by merchants.name, cs.total_spent desc;
```

Determines, for each merchant, which customer spent the most money and which spent the least.


MySQL Workbench
Local instance MySQL80 (sakila)
Local instance MySQL80 (class_e...
Local instance MySQL80 (hw_3)
MySQL Model*
File Edit View Arrange Model Database Tools Scripting Help
Bird's Eye
Zoom: 100%
Catalog Tree
mydb
Tables
Views
Routine Groups
hw_3
Catalog Layers User Types
Description Editor
place: Table
Diagram
Move Model (Quick key - press H)
mercha...
mid INT
city TEXT
state TEXT
sell
mid INT
pid INT
price DOUBLE
quantity_available I...
products
pid INT
name TEXT
category TEXT
description TE...
cont...
oid INT
pid INT
custom...
cid INT
fullname TEXT
city TEXT
state TEXT
place
cid INT
oid INT
order_date TE...
orders
oid INT
shipping_method TE...
shipping_cost DOUBLE
Modeling Add