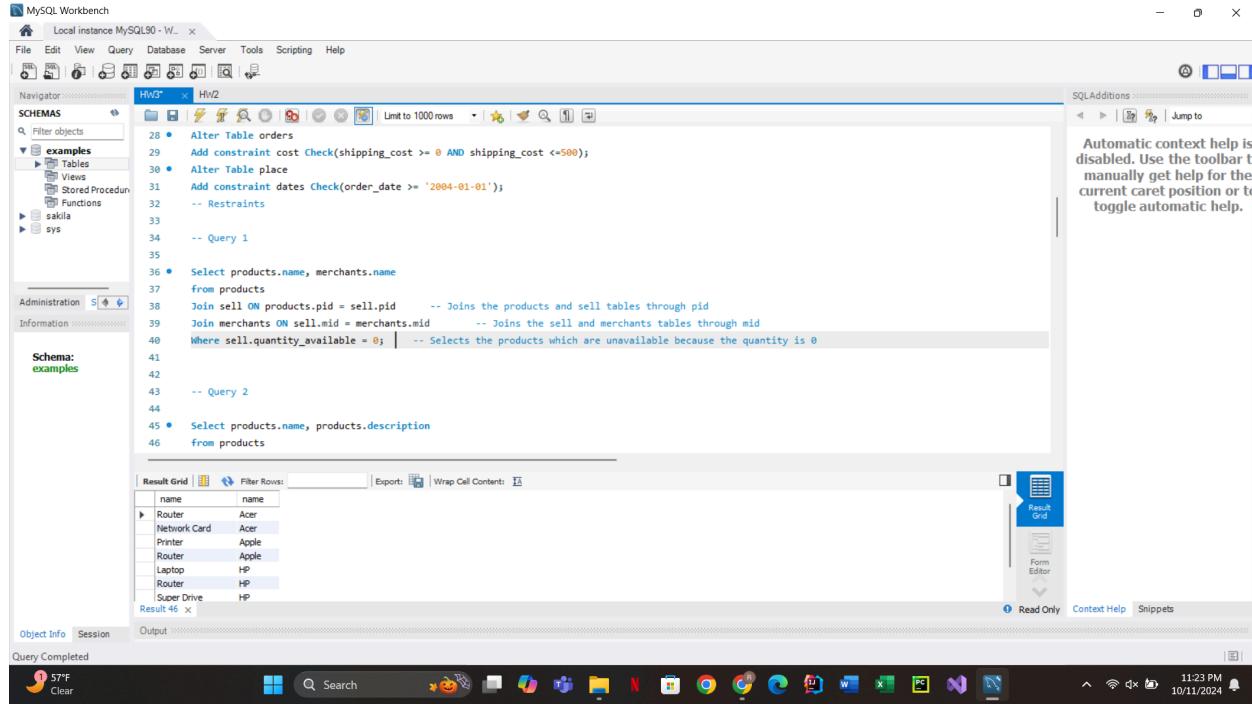


Title: DB Assignment 3

Your Name: Ryan Smith

Date: 10/11/2024

Github link: <https://github.com/rsmith1388/Databases.git>



The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar has sections for Schemas (examples), Administration, and Information. The main area contains two tabs: 'Hv3' and 'Hv2'. The 'Hv3' tab displays a SQL script with the following content:

```
28 • Alter Table orders
29   Add constraint cost Check(shipping_cost >= 0 AND shipping_cost <=500);
30 • Alter Table place
31   Add constraint dates Check(order_date >= '2004-01-01');
32   -- Restraints
33
34   -- Query 1
35
36 • Select products.name, merchants.name
37   from products
38   Join sell ON products.pid = sell.pid      -- Joins the products and sell tables through pid
39   Join merchants ON sell.mid = merchants.mid    -- Joins the sell and merchants tables through mid
40   Where sell.quantity_available = 0; | -- Selects the products which are unavailable because the quantity is 0
41
42
43   -- Query 2
44
45 • Select products.name, products.description
46   from products
```

The 'Hv2' tab shows a results grid with the following data:

name	name
Router	Acer
Network Card	Acer
Processor	Apple
Router	Apple
Laptop	HP
Router	HP
Super Drive	HP

The status bar at the bottom indicates 'Query Completed'.

Query 1: Finds products which are out of stock. The query joins tables to access products with a quantity of 0.

MySQL Workbench - Local instance MySQL90 - HW2

Query 2:

```

37   from products
38   Join sell ON products.pid = sell.pid      -- Joins the products and sell tables through pid
39   Join merchants ON sell.mid = merchants.mid    -- Joins the sell and merchants tables through mid
40   Where sell.quantity_available = 0;        -- Selects the products which are unavailable because the quantity is 0
41
42
43   -- Query 2
44
45 •   Select products.name, products.description
46   from products
47   Left Join sell ON products.pid = sell.pid      -- Joins the products and sells tables through pid and includes all products
48   Where sell.pid is null; | -- Selects the products which are not sold
49
50
51   -- Query 3
52
53 •   Select COUNT(distinct place.cid)      -- Calculates the number of customers who bought SATA drives but not routers
54   from place
55   Join contain ON place.oid = contain.oid       -- Joins the contain and place tables through oid

```

Result Grid:

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

Object Info Session Output

Query Completed

11:23 PM 10/11/2024

Query 2: Finds products which are not sold. Left join compares all products to products which are sold.

MySQL Workbench - Local instance MySQL90 - HW2

Query 3:

```

49   Execute the statement under the keyboard cursor
50
51   -- Query 3
52
53 •   Select COUNT(distinct place.cid)      -- Calculates the number of customers who bought SATA drives but not routers
54   from place
55   Join contain ON place.oid = contain.oid       -- Joins the contain and place tables through oid
56   Join products ON contain.pid = products.pid    -- Joins the products and contain tables through pid
57   Where products.name like 'SATA'
58   AND cid NOT IN (                           -- Selects the customers who bought SATA drives and checks if they bought a router
59     Select cid
60     from place
61     Join contain ON place.oid = contain.oid      -- Joins the contain and place tables through cid
62     Join products ON contain.pid = products.pid    -- Joins the products and contain tables through pid
63     Where products.name like 'Router'           -- Selects customers who bought a router
64   )
65
66
67   -- Query 4

```

Result Grid:

COUNT(distinct place.cid)
0

Object Info Session Output

Query Completed

11:23 PM 10/11/2024

Query 3: Counts how many customers bought SATA drives but not routers. The query uses joins to check if the customer bought a SATA drive and checks if they did not buy a router.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigators: Schemas Examples Tables Views Stored Procedures Functions

SQL Editor: Hw2

```

58 AND cid NOT IN ( -- Selects the customers who bought SATA drives and checks if they bought a router
59     Select Execute the statement under the keyboard cursor
60     from place
61     Join contain ON place.cid = contain.cid -- Joins the contain and place tables through cid
62     Join products ON contain.pid = products.pid -- Joins the products and contain tables through pid
63     Where products.name like 'Router' -- Selects customers who bought a router
64 );
65
66 -- Query 4
67
68 • Select products.name, sell.price * 0.8 AS sale -- Calculates the sale price after the 20% discount
69     from products
70     Join sell ON products.pid = sell.pid -- Joins the products and sell tables through pid
71     Join merchants ON sell.mid = merchants.mid -- Joins the merchants and sell tables through mid
72     Where merchants.name = 'HP' AND products.category = 'Networking'; -- Selects the products that are made by HP and fall under the Networking category
73
74 -- Query 5
75
76 • Select products.name, sell.price

```

Result Grid:

name	sale
Router	827.568000000001
Network Card	923.740000000001
Network Card	276.008
Network Card	209.76
Ethernet Adapter	1008.360000000001
Router	164.48
Router	1179.896

Object Info Session Output

Query Completed

11:24 PM 10/11/2024

Query 4: Applies a 20% discount on all HP networking products.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigators: Schemas Examples Tables Views Stored Procedures Functions

SQL Editor: Hw2

```

70 Join -- Joins the products and sell tables through pid
71 Join merchants ON sell.mid = merchants.mid -- Joins the merchants and sell tables through mid
72 Where merchants.name = 'HP' AND products.category = 'Networking'; -- Selects the products that are made by HP and fall under the Networking category
73
74 -- Query 5
75
76 • Select products.name, sell.price
77     from customers
78     Join place ON customers.cid = place.cid -- Joins the place and customers tables through cid
79     Join contain ON place.cid = contain.cid -- Joins the place and contain tables through cid
80     Join products ON contain.pid = products.pid -- Joins the products and contain tables through pid
81     Join sell ON products.pid = sell.pid -- Joins the products and sell tables through pid
82     Join merchants ON sell.mid = merchants.mid -- Joins the merchants and sell tables through mid
83     Where customers.fullname = 'Uriel Whitney' AND merchants.name = 'Acer'; -- Selects the products purchased by Uriel Whitney from Acer
84
85 -- Query 6
86
87 • Select merchants.name, SUM(sell.price) AS revenue, year(place.order_date) AS year -- Calculates annual total sales for each merchant
88     from merchants

```

Result Grid:

name	price
Super Drive	356.13
Network Card	130.43
Hard Drive	836.99
Printer	310.83
Printer	1345.37
Super Drive	356.13
Super Drive	1015.95

Object Info Session Output

Query Completed

11:24 PM 10/11/2024

Query 5: Checks what Uriel Whitney ordered from Acer. The query uses joins to access Uriel's order history with Acer.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas Examples Tables Views Stored Procedure Functions Sakila sys

SQLAdditions Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Query 6:

```

79 Join Execute the statement under the keyword cursor; -- Joins the place and contain tables through oid
80 Join products ON contain.pid = products.pid -- Joins the products and contain tables through pid
81 Join sell ON products.pid = sell.pid -- Joins the products and sell tables through pid
82 Join merchants ON sell.mid = merchants.mid -- Joins the merchants and sell tables through mid
83 Where customers.fullname = 'Uriel Whitney' AND merchants.name = 'Acer'; -- Selects the products purchased by Uriel Whitney from Acer
84
85 -- Query 6
86
87 • Select merchants.name, SUM(sell.price) AS revenue, year(place.order_date) AS year -- Calculates annual total sales for each merchant
88 From merchants
89 Join sell ON merchants.mid = sell.mid -- Joins the merchants and sell tables through mid
90 Join contain ON sell.pid = contain.pid -- Joins the sell and contain tables through pid
91 Join place ON contain.oid = place.oid -- Joins the place and contain tables through oid
92 Group by merchants.name, year; -- Groups the output by the company's name and year
93
94 -- Query 7
95
96 • Select merchants.name, SUM(sell.price) AS revenue, year(place.order_date) AS year -- Calculates annual total sales for each merchant
97 From merchants

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result 51 x

	name	revenue	year
▶	Acer	20815.7999999998	2019
▶	Acer	152986.2999999993	2011
▶	Acer	176722.77	2017
▶	Acer	262059.2899999998	2018
▶	Acer	182311.1499999999	2020
▶	Acer	60291.14	2016
▶	Apple	231573.17	2019

Object Info Session Output

Query Completed

11:24 PM 10/11/2024

Query 6: Calculates the annual total sales of each company and lists them by company name and year.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas Examples Tables Views Stored Procedure Functions Sakila sys

SQLAdditions Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Query 7:

```

91 Join place ON contain.oid = place.oid -- Joins the place and contain tables through oid
92 Group by merchants.name, year; -- Groups the output by the company's name and year
93
94 -- Query 7
95
96 • Select merchants.name, SUM(sell.price) AS revenue, year(place.order_date) AS year -- Calculates annual total sales for each merchant
97 From merchants
98 Join sell ON merchants.mid = sell.mid -- Joins the merchants and sell tables through mid
99 Join contain ON sell.pid = contain.pid -- Joins the contain and sell tables through pid
100 Join place ON contain.oid = place.oid -- Joins the place and contain tables through oid
101 Group by merchants.name, year -- Groups the output by the company's name and year
102 Order by revenue desc -- Orders the output in descending order based on revenue
103 limit 1; -- Limits the output to the company with the highest sales
104
105 -- Query 8
106
107 • Select shipping_method
108 From orders
109 Group by shipping_method

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result 52 x

	name	revenue	year
▶	Lenovo	324291.5900000067	2018

Object Info Session Output

Query Completed

11:24 PM 10/11/2024

Query 7: Finds which company had the highest revenue and in which year. The query orders the data in descending order to locate the company with the highest revenue.

The screenshot shows the MySQL Workbench interface. The main window displays a SQL query editor with several tabs open. The current tab, 'Hv1', contains a complex multi-table query involving 'place', 'contain', 'orders', 'shipping_method', 'merchants', 'sell', and 'products' tables. The query calculates revenue by category and shipping method. Other tabs visible include 'Hv2', 'orders 53', and 'USPS'. The left sidebar shows the database schema with 'examples' selected. The bottom navigation bar includes tabs for 'Object Info', 'Session', 'Output', 'Read Only', 'Context Help', and 'Snippets'. A system tray at the bottom shows various application icons.

```
100 Join place ON contain.oid = place.id      -- Joins the place and contain tables through oid
101 Group Execute the statement under the keyboard cursor Jups the output by the company's name and year
102 Order by revenue desc                  -- Orders the output in descending order based on revenue
103 limit 1;                             -- Limits the output to the company with the highest sales
104
105 -- Query 8
106
107 • Select shipping_method
108   from orders
109   Group by shipping_method
110  Order by AVG(shipping_cost)    -- Orders the output by average shipping cost
111  limit 1; | -- Limits the output to the cheapest shipping method
112
113 -- Query 9
114
115 • Select merchants.name, products.category, SUM(sell.price) AS total    -- Calculates the revenue for each category in each company
116   from merchants
117   Join sell ON merchants.mid = sell.mid        -- Joins the merchants and sell tables through mid
118   Join products ON sell.pid = products.pid     -- Joins the products and sell tables through pid
```

Query 8: Finds which shipping method has been the cheapest on average.

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (examples, sakila, sys), Tables, Views, Stored Procedures, Functions.
- Query Editor:** Shows two tabs: Hw1* and Hw2. The Hw2 tab contains the following SQL code:

```
109 Group by shipping_method
110 Order Execute the statement under the keyboard cursor output by average shipping cost
111 limit 1; -- Limits the output to the cheapest shipping method
112
113 -- Query 9
114
115 • Select merchants.name, products.category, SUM(sell.price) AS total -- Calculates the revenue for each category in each company
from merchants
116 Join sell ON merchants.mid = sell.mid -- Joins the merchants and sell tables through mid
117 Join products ON sell.pid = products.pid -- Joins the products and sell tables through pid
118 Join contain ON products.pid = contain.pid -- Joins the contain and products tables through pid
119 Group by merchants.name, products.category -- Groups by merchant name and the category of the product
120 Order by total desc; -- Orders the output in descending order based on revenue from each category
121
122
123
124 -- Query 10
125
126 • Select merchants.name, SUM(sell.price) AS total, customers.fullname -- Calculates the total spent by each customer for each company
from merchants
```
- Result Grid:** Shows the results of the last query (Query 10).

name	category	total
Acer	Peripheral	751705.6600000017
Apple	Peripheral	725401.4000000114
Lenovo	Peripheral	702701.9400000081
Dell	Peripheral	690326.4899999971
Lenovo	Networking	530399.1700000013
Apple	Networking	464218.9299999987
HP	Networking	446802.8700000033
- Toolbar:** SQLAdditions, Jump to, Automatic context help is disabled.
- Bottom Bar:** Object Info, Session, Output, Read Only, Context Help, Snippets, 57°F Clear, Search, and various system icons.

Query 9: Calculate the highest selling category for each company. The query uses joins to access each company's sales information.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigators

SCHEMAS examples

```

116  from merchants
117  Join $ Execute the statement under the keyboard cursor -- Joins the merchants and sell tables through mid
118  Join products ON sell.pid = products.pid -- Joins the products and sell tables through pid
119  Join contain ON products.pid = contain.pid -- Joins the contain and products tables through pid
120  Group by merchants.name, products.category -- Groups by merchant name and the category of the product
121  Order by total desc -- Orders the output in descending order based on revenue from each category
122
123
124  -- Query 10
125
126 • Select merchants.name, SUM(sell.price) AS total, customers.fullname -- Calculates the total spent by each customer for each company
127  From merchants
128  Join sell ON merchants.mid = sell.mid -- Joins the merchants and sell tables through mid
129  Join contain ON sell.pid = contain.pid -- Joins the contain and sell tables through pid
130  Join place ON contain.oid = place.oid -- Joins the place and contain tables through oid
131  Join customers ON place.cid = customers.cid -- Joins the customers and place tables through cid
132  Group by merchants.name, customers.fullname -- Groups by merchants and customers name
133  Order by merchants.name, total desc -- Orders the output in descending order based on total spent by each customer
134

```

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

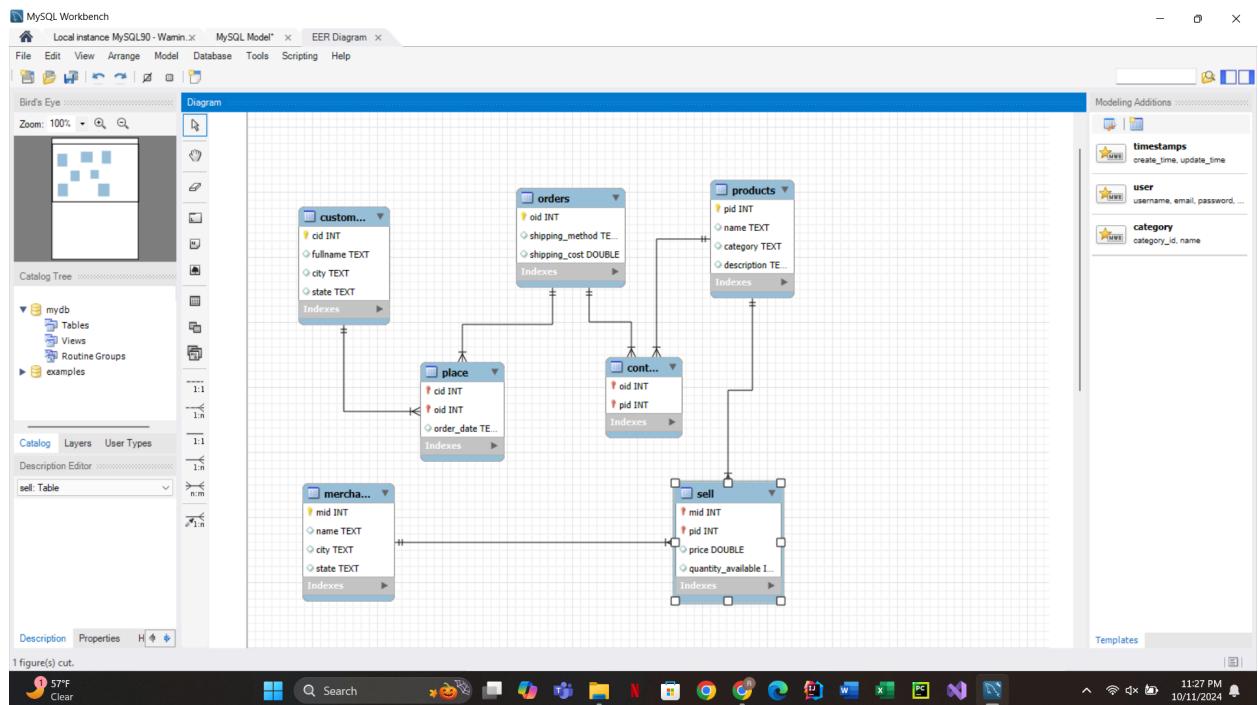
name	total	fullname
Acer	75230.29	Dean Heath
Acer	67210.0300000001	Clementine Travis
Acer	66628.59	Havva Stewart
Acer	66076.6	Uma Holland
Acer	65417.7400000001	Nissim Rosa
Acer	63868.96	Caryn Merritt
Acer	60092.6100000001	Urnel Whitney

Object Info Session Output

Query Completed

57°F Clear Search 11:24 PM 10/11/2024

Query 10: Finds out which have spent the most and least amount with each company. The query orders the data in descending order with the most spent by customers at the top of the list.



```

MySQL Workbench - Local instance MySQL80 - HW2

File Edit View Query Database Server Tools Scripting Help
Navigator Schemas Examples Tables Views Stored Procedures Functions
Administration Information Schema: Examples
Object Info Session Output
Query Completed
57% Clear
Search
11:23 PM 10/11/2024

7 • Alter Table contain Add Primary key (oid,pid);
8 • Alter Table place Add Primary key (cid,oid);
9 • Alter Table place
10 Add constraint place_cid foreign key (cid) references customers(cid),
11 Add constraint place_oid foreign key (oid) references orders(oid);
12 • Alter Table contain
13 Add constraint contain_oid foreign key (oid) references orders(oid),
14 Add constraint contain_pid foreign key (pid) references products(pid);
15 • Alter Table sell
16 Add constraint sell_mid foreign key (mid) references merchants(mid),
17 Add constraint sell_pid foreign key (pid) references products(pid);
18 • Alter Table products
19 Add constraint product_names Check(name IN ('Printer','Ethernet Adapter','Desktop','Hard Drive','Laptop','Router','Network Card','Super Drive','Monitor'));
20 • Alter Table products
21 Add constraint product_category Check(category IN ('Peripheral','Networking','Computer'));
22 • Alter Table sell
23 Add constraint sell_price Check (price >=0 AND price <=100000);
24 • Alter Table sell
25 Add constraint sell_available Check (quantity_available >=0 AND quantity_available <=1000);
26 • Alter Table orders
27 Add constraint shipping Check(shipping_method IN ('UPS','FedEx','USPS'));
28 • Alter Table orders
29 Add constraint cost Check(shipping_cost >= 0 AND shipping_cost <=500);
30 • Alter Table place
31 Add constraint dates Check(order_date >= '2004-01-01');
32 -- Restraints
33
34 -- Query 1

```

-- Restraints

Alter Table merchants Add Primary key (mid);

Alter Table products Add Primary key (pid);

Alter Table orders Add Primary key (oid);

Alter Table customers Add Primary key (cid);

Alter Table sell Add Primary key (mid,pid);

Alter Table contain Add Primary key (oid,pid);

Alter Table place Add Primary key (cid,oid);

Alter Table place

Add constraint place_cid foreign key (cid) references customers(cid),

Add constraint place_oid foreign key (oid) references orders(oid);

Alter Table contain

Add constraint contain_oid foreign key (oid) references orders(oid),

Add constraint contain_pid foreign key (pid) references products(pid);

Alter Table sell

Add constraint sell_mid foreign key (mid) references merchants(mid),

Add constraint sell_pid foreign key (pid) references products(pid);

Alter Table products

```
Add constraint product_names Check(name IN ('Printer','Ethernet  
Adapter','Desktop','Hard Drive','Laptop','Router','Network Card','Super Drive','Monitor'));  
Alter Table products  
Add constraint product_category Check(category IN  
('Peripheral','Networking','Computer'));  
Alter Table sell  
Add constraint sell_price Check (price >=0 AND price <=100000);  
Alter Table sell  
Add constraint sell_available Check (quantity_available >=0 AND quantity_available  
<=1000);  
Alter Table orders  
Add constraint shipping Check(shipping_method IN ('UPS','FedEx','USPS'));  
Alter Table orders  
Add constraint cost Check(shipping_cost >= 0 AND shipping_cost <=500);  
Alter Table place  
Add constraint dates Check(order_date >= '2004-01-01');  
-- Restraints  
  
-- Query 1
```

```
Select products.name, merchants.name  
from products  
Join sell ON products.pid = sell.pid      -- Joins the products and sell tables through pid  
Join merchants ON sell.mid = merchants.mid      -- Joins the sell and merchants tables  
through mid  
Where sell.quantity_available = 0;      -- Selects the products which are unavailable  
because the quantity is 0
```

```
-- Query 2
```

```
Select products.name, products.description
```

from products
Left Join sell ON products.pid = sell.pid -- Joins the products and sells tables through pid and includes all products
Where sell.pid is null; -- Selects the products which are not sold

-- Query 3

Select COUNT(distinct place.cid) -- Calculates the number of customers who bought SATA drives but not routers
from place
Join contain ON place.oid = contain.oid -- Joins the contain and place tables through oid
Join products ON contain.pid = products.pid -- Joins the products and contain tables through pid
Where products.name like 'SATA'
AND cid NOT IN (-- Selects the customers who bought SATA drives and checks if they bought a router
Select cid
from place
Join contain ON place.oid = contain.oid -- Joins the contain and place tables through oid
Join products ON contain.pid = products.pid -- Joins the products and contain tables through pid
Where products.name like 'Router' -- Selects customers who bought a router
);

-- Query 4

Select products.name, sell.price * 0.8 AS sale -- Calculates the sale price after the 20% discount
from products
Join sell ON products.pid = sell.pid -- Joins the products and sell tables through pid

```
Join merchants ON sell.mid = merchants.mid      -- Joins the merchants and sell  
tables through mid  
  
Where merchants.name = 'HP' AND products.category = 'Networking'; -- Selects the  
products that are made by HP and fall under the Networking category
```

-- Query 5

```
Select products.name, sell.price  
from customers  
  
Join place ON customers.cid = place.cid      -- Joins the place and customers tables  
through cid  
  
Join contain ON place.oid = contain.oid      -- Joins the place and contain tables  
through oid  
  
Join products ON contain.pid = products.pid    -- Joins the products and contain  
tables through pid  
  
Join sell ON products.pid = sell.pid          -- Joins the products and sell tables through  
pid  
  
Join merchants ON sell.mid = merchants.mid    -- Joins the merchants and sell tables  
through mid  
  
Where customers.fullname = 'Uriel Whitney' AND merchants.name = 'Acer'; -- Selects  
the products purchased by Uriel Whitney from Acer
```

-- Query 6

```
Select merchants.name, SUM(sell.price) AS revenue, year(place.order_date) AS year  
-- Calculates annual total sales for each merchant  
  
from merchants  
  
Join sell ON merchants.mid = sell.mid          -- Joins the merchants and sell tables  
through mid  
  
Join contain ON sell.pid = contain.pid        -- Joins the sell and contain tables through  
pid  
  
Join place ON contain.oid = place.oid         -- Joins the place and contain tables through  
oid
```

```
Group by merchants.name, year;      -- Groups the output by the company's name and  
year
```

-- Query 7

```
Select merchants.name, SUM(sell.price) AS revenue, year(place.order_date) AS year  
-- Calculates annual total sales for each merchant
```

```
from merchants
```

```
Join sell ON merchants.mid = sell.mid      -- Joins the merchants and sell tables  
through mid
```

```
Join contain ON sell.pid = contain.pid    -- Joins the contain and sell tables through pid
```

```
Join place ON contain.oid = place.oid     -- Joins the place and contain tables through  
oid
```

```
Group by merchants.name, year      -- Groups the output by the company's name and  
year
```

```
Order by revenue desc      -- Orders the output in descending order based on  
revenue
```

```
limit 1;      -- Limits the output to the company with the highest sales
```

-- Query 8

```
Select shipping_method
```

```
from orders
```

```
Group by shipping_method
```

```
Order by AVG(shipping_cost)  -- Orders the output by average shipping cost
```

```
limit 1;      -- Limits the output to the cheapest shipping method
```

-- Query 9

```
Select merchants.name, products.category, SUM(sell.price) AS total      -- Calculates the  
revenue for each category in each company
```

```
from merchants
```

```
Join sell ON merchants.mid = sell.mid      -- Joins the merchants and sell tables  
through mid  
Join products ON sell.pid = products.pid   -- Joins the products and sell tables  
through pid  
Join contain ON products.pid = contain.pid -- Joins the contain and products tables  
through pid  
Group by merchants.name, products.category -- Groups by merchant name and the  
category of the product  
Order by total desc;      -- Orders the output in descending order based on revenue  
from each category
```

-- Query 10

```
Select merchants.name, SUM(sell.price) AS total, customers.fullname -- Calculates  
the total spent by each customer for each company  
from merchants  
Join sell ON merchants.mid = sell.mid      -- Joins the merchants and sell tables  
through mid  
Join contain ON sell.pid = contain.pid    -- Joins the contain and sell tables through pid  
Join place ON contain.oid = place.oid     -- Joins the place and contain tables through  
oid  
Join customers ON place.cid = customers.cid -- Joins the customers and place  
tables through cid  
Group by merchants.name, customers.fullname -- Groups by merchants and  
customers name  
Order by merchants.name, total desc;      -- Orders the output in descending order  
based on total spent by each customer
```