DB Assignment 3

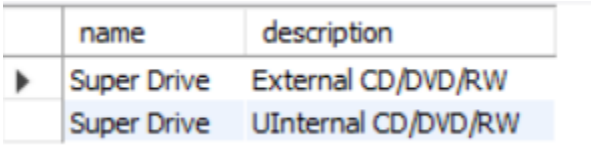
Brennan Duff

October 10, 2024

1. List names and sellers of products that are no longer available (quantity=0)
    a. Find the names and sellers of products that are out of stock
    b. Query:
        i. SELECT p.name AS product_name, m.name AS seller_name
        ii. FROM products p
        iii. JOIN sell s ON p.pid = s.pid
        iv. JOIN merchants m ON s.mid = m.mid
        v. WHERE s.quantity_available = 0;
    c. Screenshot:
        i.

product_name	seller_name
Printer	Apple
Laptop	HP
Router	Apple
Ethernet Adapter	Lenovo
Router	HP
Router	Acer
Super Drive	HP
Router	Dell
Network Card	Acer

    d. Explanation:
        i. This query retrieves the names of out of stock products and their respective sellers. It joins products with sell to match products with their availability, and then joins sell with merchants to get the seller information.

2. List names and descriptions of products that are not sold.
    a. Find the names and descriptions of products not being sold.
    b. Query:
        i. SELECT p.name, p.description
        ii. FROM products p
        iii. LEFT JOIN sell s ON p.pid = s.pid
        iv. WHERE s.pid IS NULL;
    c. Screenshot:
        i. 

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

    d. Explanation:
        i. The query selects product names and descriptions for all products that are not being sold. It uses left join between products and sell to find products without any match in sell (i.e., those that are not sold). Products with no corresponding record in the table are returned.

3. How many customers bought SATA drives but not any routers?
    a. Find the customers that bought SATA drives but not routers.
    b. Query:

```
i.    SELECT COUNT(DISTINCT c.cid) AS customer_count
ii.   FROM customers c
iii.  JOIN place pl ON c.cid = pl.cid
iv.   JOIN contain co ON pl.oid = co.oid
v.    JOIN products p1 ON co.pid = p1.pid
vi.   WHERE p1.description LIKE '%SATA%'
vii.  AND NOT EXISTS (
viii.       SELECT 1
ix.         FROM contain co2
x.          JOIN products p2 ON co2.pid = p2.pid
xi.         WHERE co2.oid = pl.oid AND p2.name = 'Router'
xii.  );
```

    c. Screenshot:
        i.

	customer_count
▶	19

    d. Explanation:
        i. The query counts the number of customers who bought SATA drives but did not buy routers. It joins customers, place, contains and products to find customers who purchased SATA drives and uses a not exists clause to exclude those who also bought routers.

4. HP has a 20% sale on all its Networking products
    a. Modify the price of hp networking products
    b. Query:
        i. UPDATE sell
        ii. SET price = price * 0.8
        iii. WHERE pid IN (
        iv. SELECT subquery.pid
        v. FROM (
        vi. SELECT DISTINCT s.pid
        vii. FROM sell s
        viii. JOIN products p ON s.pid = p.pid
        ix. JOIN merchants m ON s.mid = m.mid
        x. WHERE m.name = 'HP' AND p.category = 'Networking'
        xi.) AS subquery
        xii.);
    c. Screenshot:
        i. n/a
    d. Explanation:
        i. This query updates the prices of HP's Networking products by applying a 20% discount, that is, if it was working. It stumped me. It is currently affecting 47 rows, while it should only be affecting 10.

5. What did Uriel Whitney order from Acer?
    a. Find the prices and their products that cid 1 bought
    b. Query:
        i. SELECT DISTINCT p.name AS product_name, s.price
        ii. FROM customers c
        iii. JOIN place pl ON c.cid = pl.cid
        iv. JOIN contain co ON pl.oid = co.oid
        v. JOIN products p ON co.pid = p.pid
        vi. JOIN sell s ON p.pid = s.pid
        vii. JOIN merchants m ON s.mid = m.mid
        viii. WHERE c.fullname = 'Uriel Whitney' AND m.name = 'Acer';
    c. Screenshot:

i.

	product_name	price
	Router	394.04
	Laptop	33.5
	Network Card	130.43
	Laptop	247.96
	Router	521.07
	Hard Drive	836.99
	Super Drive	1124.26
	Network Card	609.2
	Network Card	405.4
	Super Drive	1015.95
	Network Card	837.12
	Hard Drive	1151.28
	Printer	836.28
	Router	945.51
	Hard Drive	333.71
	Ethernet Ada...	446.62
	Laptop	522.73
	Desktop	311.06
	Printer	310.83
	Router	780.65
	Monitor	1435.38

d. Explanation:

   i. This query retrieves the product names and prices for items that Uriel Whitney ordered from Acer. It joins customers, place, contain, products, sell, and merchants to link orders to both the customer and the seller, filtering the results to include only orders made by Uriel Whitney from Acer.

6. List the annual total sales for each company (sort the results along the company and the year attributes).
    a. Find annual total sales and sort by company and year
    b. Query:
        i. SELECT m.name AS company,
        ii. EXTRACT(YEAR FROM pl.order_date) AS year,
        iii. SUM(s.price) AS total_sales
        iv. FROM merchants m
        v. JOIN sell s ON m.mid = s.mid
        vi. JOIN contain co ON s.pid = co.pid
        vii. JOIN place pl ON co.oid = pl.oid
        viii. GROUP BY m.name, EXTRACT(YEAR FROM pl.order_date)
        ix. ORDER BY m.name, year;
    c. Screenshot:
        i.

company	year	total_sales
Acer	2011	152986.3
Acer	2016	60291.14
Acer	2017	176722.76999999987
Acer	2018	262059.28999999986
Acer	2019	208815.80000000005
Acer	2020	182311.15000000002
Apple	2011	166822.9100000001
Apple	2016	64748.45999999994
Apple	2017	179560.77999999997
Apple	2018	300413.22999999986
Apple	2019	231573.17000000007
Apple	2020	216461.05999999997

ii.

	Dell	2011	181730.34999999992
	Dell	2016	71462.86999999997
	Dell	2017	182288.6099999999
	Dell	2018	315004.8199999998
	Dell	2019	221391.83000000005
	Dell	2020	208063.07999999993
	HP	2011	141030.14999999985
	HP	2016	56986.119999999995
	HP	2017	136092.42999999993
	HP	2018	222707.07999999987
	HP	2019	173334.00999999978
	HP	2020	180775.18
	Lenovo	2011	184939.4100000001
	Lenovo	2016	70131.56999999998
	Lenovo	2017	197980.33000000022
	Lenovo	2018	324291.5900000007
	Lenovo	2019	232610.80000000005
	Lenovo	2020	214154.2500000003

d. Explanation:

   i. This query retrieves the annual total sales for each company. It joins merchants, sell, contain, and place to compute total sales based on product prices and quantities, grouping the results by the company name and the year of the order.

7. Which company had the highest annual revenue and in what year?
    a. Find highest annual revenue
    b. Query:
        i. SELECT m.name AS company,
        ii. EXTRACT(YEAR FROM pl.order_date) AS year,
        iii. SUM(s.price) AS total_sales
        iv. FROM merchants m
        v. JOIN sell s ON m.mid = s.mid
        vi. JOIN contain co ON s.pid = co.pid
        vii. JOIN place pl ON co.oid = pl.oid
        viii. GROUP BY m.name, EXTRACT(YEAR FROM pl.order_date)
        ix. ORDER BY total_sales DESC;
    c. Screenshot:
        i.

company	year	total_sales
Lenovo	2018	315271.455968
Dell	2018	306980.7124959997
Apple	2018	292986.98678399954
Acer	2018	256153.0683359999
Lenovo	2019	226015.56228799978
Apple	2019	225315.53836799992
Dell	2019	215167.67124799988
HP	2018	214601.15190399974
Apple	2020	210092.41239999983
Lenovo	2020	207338.25199999986
Acer	2019	203713.66049599997
Dell	2020	201938.20387199984
Lenovo	2017	192512.35806399994
Lenovo	2011	179110.7776799999
Acer	2020	177477.68337599997
Dell	2017	177466.5811519998

ii.

	Dell	2011	176561.0350239999
	Apple	2017	175268.68027200003
	HP	2020	174232.308784
	Acer	2017	172602.8876319999
	HP	2019	166607.95404799996
	Apple	2011	162221.60851199998
	Acer	2011	148817.83863999994
	HP	2011	136003.97696000012
	HP	2017	131664.47169600005
	Dell	2016	69525.84511999998
	Lenovo	2016	67914.10361600001
	Apple	2016	63423.44903999995
	Acer	2016	58820.503391999984
	HP	2016	55097.21860799999

d. Explanation:

   i. This query does the same thing as the last, but sorts by the total sales instead.

8. On average, what was the cheapest shipping method used ever?
    a. Find lowest average shipping cost
    b. Query:
        i. SELECT shipping_method, AVG(shipping_cost) AS avg_shipping_cost
        ii. FROM orders
        iii. GROUP BY shipping_method
        iv. ORDER BY avg_shipping_cost ASC
        v. LIMIT 1;
    c. Screenshot:
        i.

	shipping_method	avg_shipping_cost
▶	USPS	7.455760869565214

    d. Explanation:
        i. This query takes the average of the shipping cost from the orders table.

9. What is the best sold ($) category for each company?
    a. Get total sales of products and group by category and company
    b. Query:
        i. SELECT m.name AS company,
        ii. p.category,
        iii. SUM(s.price) AS total_sales
        iv. FROM merchants m
        v. JOIN sell s ON m.mid = s.mid
        vi. JOIN contain co ON s.pid = co.pid
        vii. JOIN products p ON s.pid = p.pid
        viii. GROUP BY m.name, p.category
        ix. ORDER BY m.name, total_sales DESC;
    c. Screenshot:
        i.

company	category	total_sales
Acer	Peripheral	751705.6600000008
Acer	Networking	351617.9905919997
Acer	Computer	58136.399999999994
Apple	Peripheral	725401.4400000034
Apple	Networking	431960.3531039999
Apple	Computer	137254.73000000016
Dell	Peripheral	690326.4899999981
Dell	Networking	409805.57448000106
Dell	Computer	191426.1699999992
HP	Peripheral	416673.2899999985
HP	Networking	411773.52499200025
HP	Computer	182077.99999999974
Lenovo	Peripheral	702791.940000001
Lenovo	Networking	492336.44040000107
Lenovo	Computer	141047.69999999998

    d. Explanation:
        i. This query is similar to 6 and 7 in that it gets the total sales, but instead of the year, it is grouped by the category of product.

10. For each company find out which customers have spent the most and the least amounts.
    a. Get lowest and highest amount spent by a customer for each company
    b. Query:

```
WITH customer_spending AS (
        SELECT c.cid AS customer_id,
        m.name AS company,
        SUM(s.price) AS total_spent
        FROM customers c
        JOIN place pl ON c.cid = pl.cid
        JOIN contain co ON pl.oid = co.oid
        JOIN sell s ON co.pid = s.pid
        JOIN products p ON s.pid = p.pid
        JOIN merchants m ON s.mid = m.mid
        GROUP BY c.cid, m.name
)
SELECT company,
        MIN(total_spent) AS least_spent,
        MAX(total_spent) AS most_spent
FROM customer_spending
GROUP BY company;
```

    c. Screenshot:

i.

	company	least_spent	most_spent
▶	Acer	31125.240239999996	73298.82603199998
	Apple	31331.602847999995	81893.04032
	HP	25051.885936000002	64109.55172800003
	Dell	30127.103615999993	83038.63839999997
	Lenovo	32960.24836800001	80809.77443199999

d. Explanation:

   i. This query first calculates the amount each customer has spent at each company with a CTE. It then gets the min and max and groups by company.