Kamran Shaikh
Lab3
CSE 581

1)

Code

```
SELECT TOP 10 VendorID, SUM(PaymentTotal) AS PaymentSum FROM Invoices GROUP BY
VendorID ORDER BY PaymentSum DESC
```

Comments)

Gets Top 10 Vendors who had the least PaymentSum

	VendorID	PaymentSum
1	110	86069.2200
2	122	23177.9600
3	72	21842.0000
4	104	7125.3400
5	99	6940.2500
6	121	6740.2500
7	119	4901.2600
8	123	4167.1300
9	86	2433.0000
10	100	2184.5000

2)

Code

```
SELECT VendorName, COUNT(InvoiceTotal) AS InvoiceCount, AVG(InvoiceTotal) AS
InvoiceAverage FROM Vendors JOIN Invoices ON Vendors.VendorID = Invoices.VendorID
GROUP BY VendorName ORDER BY InvoiceCount
```

Comments)

Count the amount of Invoice Totals, Calculate Average of the Invoice Total. Sort by the Invoice Count

	VendorName	InvoiceCount	InvoiceAverage
1	Abbey Office Furnishings	1	17.5000
2	Bertelsmann Industry Svcs…	1	6940.2500
3	Cahners Publishing Company	1	2184.5000
4	Coffee Break Service	1	41.8000
5	Computerworld	1	2433.0000
6	Ford Motor Credit Company	1	503.2000
7	Franchise Tax Board	1	1600.0000
8	Fresno County Tax Collect…	1	856.9200
9	Gostanian General Building	1	450.0000
10	Dean Witter Reynolds	1	1367.5000
11	Digital Dreamworks	1	7125.3400
12	Dristas Groom & McCormick	1	220.0000

3)

```
SELECT AccountDescription, COUNT(*) AS LineItemCount, SUM(InvoiceLineItemAmount) AS LineItemSum FROM InvoiceLineItems JOIN GLAccounts ON InvoiceLineItems.AccountNo = GLAccounts.AccountNo GROUP BY AccountDescription HAVING (COUNT(*) > 1) ORDER BY LineItemSum ASC
```

Comments)

Count the amount of Entries. Sum the InvoiceLineItemAmounts and check if the LineItemCount is more than 1. Sort the table by increasing LineItemSum

	AccountDescription	LineItemCount	LineItemSum
1	Office Supplies	3	175.8000
2	Telephone	7	266.0100
3	Group Insurance	3	564.0000
4	Computer Equipment	3	2137.0500
5	Direct Mail Advertising	6	3900.7700
6	Books, Dues, and Subscrip…	6	5207.3200
7	Book Production Costs	8	6175.1200
8	Outside Services	3	13394.1000
9	Freight	60	27599.6500
10	Book Printing Costs	8	148759.9700

4)

Code)

```
SELECT AccountNo, SUM(InvoiceLineItemAmount) AS LineItemSum FROM InvoicelineItems
GROUP BY AccountNo WITH ROLLUP
```

Comments)

Determines total amount for each invoice account using ROLLUP function

	AccountNo	LineItemSum
1	150	17.5000
2	160	2137.0500
3	170	356.4800
4	400	148759.9700
5	403	6175.1200
6	507	1600.0000
7	510	564.0000
8	520	1750.0000
9	521	16.6200
10	522	266.0100
11	523	450.0000
12	540	3900.7700

5)

Code)

```
SELECT VendorName, COUNT(DISTINCT AccountNo) AS TotalNumber FROM Vendors JOIN Invoices
ON Vendors.VendorID = Invoices.VendorID JOIN InvoiceLineItems ON Invoices.InvoiceID =
InvoiceLineItems.InvoiceID GROUP BY VendorName HAVING COUNT(DISTINCT AccountNo) > 2
```

Comments)

Returns Unique VendorName, and Count of Account Numbers that occur more than twice.

	VendorName	TotalNumber
1	Wells Fargo Bank	3

6)

Code)

```
SELECT VendorName, InvoiceNumber, InvoiceTotal FROM Vendors JOIN Invoices ON
Vendors.VendorID = Invoices.VendorID WHERE PaymentTotal > (SELECT AVG(PaymentTotal)
FROM Invoices);
```

Comments)

Returns VendorName, InvoiceNumber and InvoiceTotal whose PaymentTotal is greater than the average PaymentTotal

	VendorName	InvoiceNumber	InvoiceTotal
1	United Parcel Service	989319-457	3813.3300
2	Yesmed, Inc	10843	4901.2600
3	Pollstar	77290	1750.0000
4	Digital Dreamworks	P02-3772	7125.3400
5	Franchise Tax Board	RTR-72-3662-X	1600.0000
6	Malloy Lithographing Inc	0-2058	37966.1900
7	Data Reproductions Corp	40318	21842.0000
8	United Parcel Service	989319-437	2765.3600
9	Ingram	31359783	1575.0000
10	United Parcel Service	989319-477	2184.1100

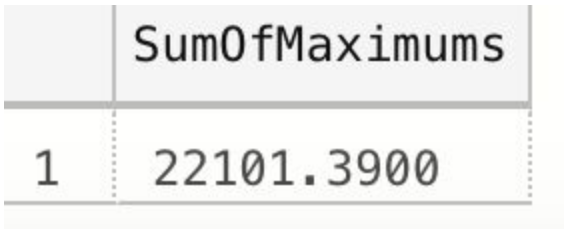
7)

Code

```
SELECT SUM(LargestInvoice) AS SumOfMaximums FROM (SELECT VendorID, MAX(InvoiceTotal)
AS LargestInvoice FROM Invoices WHERE InvoiceTotal - CreditTotal - PaymentTotal > 0
GROUP BY VendorID) AS MaxInvoice
```

Comments)

Returns the total sum of the Largest Invoice Total for each row where BalanceDue is greater than 0.



	SumOfMaximums
1	22101.3900

8)

Code)

```
SELECT Vendor1.VendorID, Vendor1.VendorName, Vendor1.VendorCity, Vendor1.VendorState
FROM Vendors AS Vendor1 LEFT JOIN Vendors AS Vendor2 ON Vendor1.VendorCity =
Vendor2.VendorCity AND Vendor1.VendorState = Vendor2.VendorState AND Vendor1.VendorID
<> Vendor2.VendorID WHERE Vendor2.VendorID IS NULL ORDER BY VendorState DESC, VendorID
ASC
```

Comments

Checks if VendorCity and VendorState occurs more than once. If it does, it will not return in the table, else return VendorID, VendorName, VendorCity, and VendorState. Sort by decreasing VendorState and Increasing VendorID

	VendorID	VendorName	VendorCity	VendorState
1	1	US Postal Service	Madison	WI
2	31	Enterprise Communications…	Mclean	VA
3	83	Ingram	Dallas	TX
4	123	Federal Express Corporati…	Memphis	TN
5	84	Boucher Communications Inc	Fort Washington	PA
6	40	Nat Assoc of College Stor…	Oberlin	OH
7	50	Publishers Weekly	Marion	OH
8	88	Edward Data Services	Cleves	OH
9	38	Venture Communications In…	New York	NY
10	60	The Mailers Guide Co	New Rochelle	NY
11	61	American Booksellers Assoc	Tarrytown	NY
12	100	Cahners Publishing Company	The Lake	NY

Remarks)

Overall, I thought this lab had a mix of easy and difficult problems. The first 4 were quite simple, whereas the last 4 were difficult. This lab however did teach me more about the usage of subqueries and helped me get used to the JOIN function. I definitely now feel more comfortable with the basics of SQL, and can’t wait to see what the next lab has in store for us.