1)

```
SELECT VendorContactFName + LEFT(VendorContactLName,1) AS FirstL, RIGHT(VendorPhone,8)
AS PhoneNumber FROM Vendors WHERE LEFT(VendorPhone,3) != '916' ORDER BY
VendorContactFName,VendorContactLName
```

Comments)

Gets two columns, one prints out the FirstName with the Last Name Initial. The second prints out the phone number without the area code. These columns will not include names and phone numbers of area code 916.

	FirstL	PhoneNumber
1	AbeR	555-6113
2	AlexandroA	555-2993
3	AndersR	555-5570
4	AngelL	555-0700
5	AniaI	555-8950
6	AnneB	555-7900
7	AntonM	555-6670
8	BillL	555-9375
9	BrianG	555-9000
10	BrittneeB	555-3500
11	CaitlinJ	555-2420
12	CarleeR	555-5561
13	CesarA	555-3700
14	CharlieB	555-4091
15	ClarenceM	555-8700
16	DanielB	555-7222
17	DeangeloS	555-6621

2)

```
SELECT InvoiceNumber AS Invoice#, InvoiceTotal - PaymentTotal - CreditTotal AS Balance
FROM Invoices WHERE (InvoiceTotal-PaymentTotal-CreditTotal > 0) AND InvoiceDueDate <
GETDATE() + 15
```

Comments)

Gets two columns, Invoice Number and Balance when there is not a null balance and the InvoiceDueDate is due less than 15 days of today's date.

Results Messages

	Invoice#	Balance
1	39104	85.3100
2	963253264	52.2500
3	31361833	579.4200
4	263253268	59.9700
5	263253270	67.9200
6	263253273	30.7500
7	P-0608	19351.1800
8	9982771	503.2000
9	134116	90.3600
10	0-2436	10976.0600
11	547480102	224.0000

3)

```
SELECT InvoiceNumber AS Invoice#, InvoiceTotal - PaymentTotal - CreditTotal AS Balance
FROM Invoices WHERE (InvoiceTotal-PaymentTotal-CreditTotal > 0) AND InvoiceDueDate <
EOMONTH(GETDATE())
```

Comments)

Similar to task 2. Prints out Invoice Number and Balance Due when there is not a null Balance and the invoice due date is less than the end of this months day.

	Invoice#	Balance
1	39104	85.3100
2	963253264	52.2500
3	31361833	579.4200
4	263253268	59.9700
5	263253270	67.9200
6	263253273	30.7500
7	P-0608	19351.1800
8	9982771	503.2000
9	134116	90.3600
10	0-2436	10976.0600
11	547480102	224.0000

4)

```
SELECT InvoiceNumber AS Invoice#, InvoiceTotal - PaymentTotal - CreditTotal AS
Balance, RANK() OVER (ORDER BY InvoiceTotal-PaymentTotal-CreditTotal ASC) AS
BalanceRank FROM Invoices WHERE (InvoiceTotal-PaymentTotal-CreditTotal > 0) AND
InvoiceDueDate < GETDATE() + 15
```

Comments)
Similar to task 2, however we rank the new columns based on the Balance due.

	Invoice#	Balance	BalanceRank
1	263253273	30.7500	1
2	963253264	52.2500	2
3	263253268	59.9700	3
4	263253270	67.9200	4
5	39104	85.3100	5
6	134116	90.3600	6
7	547480102	224.0000	7
8	9982771	503.2000	8
9	31361833	579.4200	9
10	0-2436	10976.0600	10
11	P-0608	19351.1800	11

Remarks:

Overall I thought this lab was quite easy. It gave me the chance to use the functions taught in class and get a better understanding of how they are used. I feel like this was a great way to end the first part of the course before we begin designing databases.