Simple Queries:

Simple Queries have 2 tables joined.

Premise 01: What products have a special offer?

Detailed explanation of the problem: Return all product IDs that have a special offer.

Query:

```
USE adventureworks2014;

SELECT  sales.specialofferproduct.productid
FROM    sales.specialoffer
        INNER JOIN sales.specialofferproduct
                ON sales.specialoffer.specialofferid =
                   sales.specialofferproduct.specialofferid;
```

Sample Output:

	ProductID
1	680
2	706
3	707
4	707
5	707
6	707
7	707
8	708
9	708
10	708
11	708
12	708
13	709
14	709
15	709
16	709
17	710
18	711
19	711
20	711

Premise 02: What phone numbers are the email-addresses associated with?

Detailed explanation of the problem: Return all Email Addresses and Phone Numbers that have the same BusinessEntityID.

Query:

```sql
USE adventureworks2014;

SELECT E.emailaddress,
       P.phonenumber
FROM   person.personphone AS P
       INNER JOIN person.emailaddress AS E
              ON P.businessentityid = E.businessentityid;
```

Sample Output:

	EmailAddress	PhoneNumber
1	a0@adventure-works.com	645-555-0193
2	a1@adventure-works.com	675-555-0100
3	aaron0@adventure-works.com	652-555-0115
4	aaron1@adventure-works.com	908-555-0100
5	aaron10@adventure-works.com	622-555-0113
6	aaron11@adventure-works.com	831-555-0184
7	aaron12@adventure-works.com	695-555-0117
8	aaron13@adventure-works.com	466-555-0180
9	aaron14@adventure-works.com	143-555-0139
10	aaron15@adventure-works.com	477-555-0181
11	aaron16@adventure-works.com	647-555-0136
12	aaron17@adventure-works.com	754-555-0137
13	aaron18@adventure-works.com	548-555-0129
14	aaron19@adventure-works.com	1 (11) 500 555-0142
15	aaron2@adventure-works.com	160-555-0100
16	aaron20@adventure-works.com	306-555-0143
17	aaron21@adventure-works.com	1 (11) 500 555-0130
18	aaron22@adventure-works.com	193-555-0161
19	aaron23@adventure-works.com	273-555-0153
20	aaron24@adventure-works.com	648 555 0158

Premise 03: Provide the order information for the credit cards.

Detailed explanation of the problem: Return credit card numbers and the account numbers, order IDs, order dates, and order statuses associated with them.

Query:
```sql
USE adventureworks2014;

SELECT sales.creditcard.cardnumber,
       sales.salesorderheader.accountnumber,
       sales.salesorderheader.salesorderid,
       sales.salesorderheader.orderdate,
       sales.salesorderheader.status
FROM   sales.creditcard
       INNER JOIN sales.salesorderheader
               ON sales.creditcard.creditcardid =
                  sales.salesorderheader.creditcardid;
```

Sample Output:

	CardNumber	AccountNumber	SalesOrderID	OrderDate	Status
1	77777462752259	10-4020-000676	43659	2011-05-31 00:00:00.000	5
2	11118446131713	10-4020-000117	43660	2011-05-31 00:00:00.000	5
3	55552021265206	10-4020-000442	43661	2011-05-31 00:00:00.000	5
4	77772950097421	10-4020-000227	43662	2011-05-31 00:00:00.000	5
5	11118755169665	10-4020-000510	43663	2011-05-31 00:00:00.000	5
6	77779337333220	10-4020-000397	43664	2011-05-31 00:00:00.000	5
7	11119090332357	10-4020-000146	43665	2011-05-31 00:00:00.000	5
8	11119334752893	10-4020-000511	43666	2011-05-31 00:00:00.000	5
9	33339376059746	10-4020-000646	43667	2011-05-31 00:00:00.000	5
10	77776957027030	10-4020-000514	43668	2011-05-31 00:00:00.000	5
11	11113874249349	10-4020-000578	43669	2011-05-31 00:00:00.000	5
12	11119277394350	10-4020-000504	43670	2011-05-31 00:00:00.000	5
13	11119759315644	10-4020-000200	43671	2011-05-31 00:00:00.000	5
14	11114254842698	10-4020-000119	43672	2011-05-31 00:00:00.000	5
15	11112905389252	10-4020-000618	43673	2011-05-31 00:00:00.000	5
16	33337125636756	10-4020-000083	43674	2011-05-31 00:00:00.000	5
17	11111724608350	10-4020-000670	43675	2011-05-31 00:00:00.000	5
18	77771339258802	10-4020-000017	43676	2011-05-31 00:00:00.000	5
19	33331483306664	10-4020-000679	43677	2011-05-31 00:00:00.000	5
20	11119338613020	10-4020-000203	43678	2011-05-31 00:00:00.000	5

Premise 04: Provide the ship date, due date, and city associated with the order number? Detailed explanation of the problem.

Detailed explanation of the problem: Return Order Numbers and the cities, ship dates, due dates associated with them.

Query:

```sql
USE adventureworks2014;

SELECT sales.salesorderheader.salesordernumber,
       person.address.city,
       sales.salesorderheader.shipdate,
       sales.salesorderheader.duedate
FROM   sales.salesorderheader
       INNER JOIN person.address
               ON sales.salesorderheader.billtoaddressid =
                  person.address.addressid
                  AND sales.salesorderheader.shiptoaddressid =
                      person.address.addressid;
```

Sample Output:

	SalesOrderNumber	City	ShipDate	DueDate
1	SO44802	Miami	2011-11-07 00:00:00.000	2011-11-12 00:00:00.000
2	SO45578	Miami	2012-02-05 00:00:00.000	2012-02-10 00:00:00.000
3	SO46388	Miami	2012-05-07 00:00:00.000	2012-05-12 00:00:00.000
4	SO67382	Paris	2014-03-08 00:00:00.000	2014-03-13 00:00:00.000
5	SO52952	Roncq	2013-07-28 00:00:00.000	2013-08-02 00:00:00.000
6	SO67993	Lille	2014-03-17 00:00:00.000	2014-03-22 00:00:00.000
7	SO60871	Versailles	2013-12-02 00:00:00.000	2013-12-07 00:00:00.000
8	SO74173	Chatou	2014-06-07 00:00:00.000	2014-06-12 00:00:00.000
9	SO56404	Roncq	2013-09-25 00:00:00.000	2013-09-30 00:00:00.000
10	SO51316	Les Ulis	2013-06-14 00:00:00.000	2013-06-19 00:00:00.000
11	SO48868	Les Ulis	2012-12-16 00:00:00.000	2012-12-21 00:00:00.000
12	SO47275	Roubaix	2012-07-30 00:00:00.000	2012-08-04 00:00:00.000
13	SO68279	Roubaix	2014-03-22 00:00:00.000	2014-03-27 00:00:00.000
14	SO71511	Morangis	2014-05-04 00:00:00.000	2014-05-09 00:00:00.000
15	SO49302	Morangis	2013-01-20 00:00:00.000	2013-01-25 00:00:00.000
16	SO71532	Paris	2014-05-04 00:00:00.000	2014-05-09 00:00:00.000
17	SO56224	Versailles	2013-09-21 00:00:00.000	2013-09-26 00:00:00.000
18	SO72031	Paris	2014-05-09 00:00:00.000	2014-05-14 00:00:00.000
19	SO52691	Lebanon	2013-07-23 00:00:00.000	2013-07-28 00:00:00.000
20	SO58509	Poissy en Prie	2013-10-30 00:00:00.000	2013-11-04 00:00:00.000

Premise 05: Provide information on orders with vendors with good credit rating.

Detailed explanation of the problem: Return Order IDs Employee IDs, Vendor IDs, total due, and the credit rating for orders that have a vendor with a credit rating greater than 3.

Query:

```sql
USE adventureworks2014;

SELECT  purchasing.purchaseorderheader.purchaseorderid,
        purchasing.purchaseorderheader.employeeid,
        purchasing.purchaseorderheader.vendorid,
        purchasing.purchaseorderheader.totaldue,
        purchasing.vendor.creditrating
FROM    purchasing.vendor
        INNER JOIN purchasing.purchaseorderheader
                ON purchasing.vendor.businessentityid =
                   purchasing.purchaseorderheader.vendorid
WHERE   creditrating > 3
ORDER   BY purchaseorderid;
```

Sample Output:

	PurchaseOrderID	EmployeeID	VendorID	TotalDue	CreditRating
1	7	255	1678	64847.5328	4
2	74	261	1652	49276.9778	5
3	86	253	1678	64847.5328	4
4	153	252	1652	7727.8451	5
5	165	251	1678	64847.5328	4
6	232	254	1652	49276.9778	5
7	244	261	1678	36852.4406	4
8	311	258	1652	7727.8451	5
9	323	257	1678	27995.0921	4
10	390	260	1652	53246.193	5
11	402	254	1678	36852.4406	4
12	420	250	1652	3758.6299	5
13	432	261	1678	27995.0921	4
14	499	258	1652	57004.8229	5
15	511	257	1678	64847.5328	4
16	582	260	1652	49276.9778	5
17	594	254	1678	36852.4406	4
18	669	259	1652	7727.8451	5
19	681	258	1678	27995.0921	4
20	748	256	1652	57004.8229	5