BUAN 6320 - Individual Project

Final Report

Sai Rahul Dhulipalla
Net ID: SXD170033

12/12/2018

Table of Contents

Objective:

This project is done as part of BUAN6320 Course. The purpose of this project is to Clean the Provided Datasets, Implement Database as per given Data Dictionary, insert data to Database, write SQL Queries to produce the results in answering the given questions and finally performing a linear regression to arrive at a suitable forecast.

1. Introduction:

In this Report, the following things have been discussed.

- Steps taken to clean the datasets
- Steps taken to Implement Database in Microsoft SQL Server and import data to Tables
- SQL Queries written to produce the results in answering the given questions.
- Linear regression performed on existing sales data to arrive at monthly forecast of sales for February 2014 to January 2015 time period.

2. Datasets Cleaning:

Data cleaning process started by uploading the provided 3 flat files into Microsoft SQL Server and naming them as

1) DataSet2
2) DataSet3
3) DataSet4

After this, I glanced through all datasets and found problems with data in Dataset2 & Dataset3. The problems faced, and the fixes done are mentioned below.

Problems and Fixes in DataSet2:

Problem1: In DataSet2 for 15 INV_NUM’s there are two records with two different Customers when each INV_NUM must have only one record with one Customer.
Fix: I fixed this problem by taking anyone record of the repeating INV_NUM’s and uploading them into a temporary backup table for further business confirmation and deleted those records from DataSet2. The Queries for Deleting these records is mentioned below.

Queries:

```
delete from DataSet2 where INV_NUM =3370 and CUST_CODE = 929;

delete from DataSet2 where INV_NUM =3364 and CUST_CODE = 1411;

delete from DataSet2 where INV_NUM =2921 and CUST_CODE = 888;

delete from DataSet2 where INV_NUM =3135 and CUST_CODE = 1307;

delete from DataSet2 where INV_NUM =3347 and CUST_CODE = 1401;

delete from DataSet2 where INV_NUM =3024 and CUST_CODE = 1208;

delete from DataSet2 where INV_NUM =2644 and CUST_CODE = 988;

delete from DataSet2 where INV_NUM =2362 and CUST_CODE = 1014;

delete from DataSet2 where INV_NUM =2364 and CUST_CODE = 878;

delete from DataSet2 where INV_NUM =2315 and CUST_CODE = 993;
delete from DataSet2 where INV_NUM =2885 and CUST_CODE = 1219;

delete from DataSet2 where INV_NUM =2577 and CUST_CODE = 730;

delete from DataSet2 where INV_NUM =2275 and CUST_CODE = 393;

delete from DataSet2 where INV_NUM =2124 and CUST_CODE = 917;

delete from DataSet2 where INV_NUM =1978 and CUST_CODE = 851;
```

Problems and Fixes in DataSet3:

Problem1: In DataSet3 few records of Column PROD_QOH has some special characters.

Fix: I fixed this problem by removing those Special Characters from PROD_QOH Column. The Queries of these fixes are mentioned below

Queries:

```
UPDATE DataSet3

SET PROD_QOH = right(PROD_QOH,2)

WHERE PROD_QOH in (SELECT PROD_QOH FROM DataSet3 where PROD_QOH like 'Ä%')


UPDATE DataSet3

SET PROD_QOH = right(PROD_QOH,2)

WHERE PROD_QOH in (SELECT PROD_QOH FROM DataSet3 where PROD_QOH like 'ñ%')


UPDATE DataSet3

SET PROD_QOH = right(PROD_QOH,2)

WHERE PROD_QOH in (SELECT PROD_QOH FROM DataSet3 where PROD_QOH like 'Ã%')


UPDATE DataSet3

SET PROD_QOH = right(PROD_QOH,3)

WHERE PROD_QOH in (SELECT PROD_QOH FROM DataSet3 where PROD_QOH like '?%')


UPDATE DataSet3

SET PROD_QOH = right(PROD_QOH,2)

WHERE PROD_QOH in (SELECT PROD_QOH FROM DataSet3 where PROD_QOH like '?%')
```

```
UPDATE DataSet3

SET PROD_QOH = right(PROD_QOH,2)

WHERE PROD_QOH in (SELECT PROD_QOH FROM DataSet3 where PROD_QOH like 'Æ%')
```

Problem2: In DataSet3 few records of Column PROD_SKU has some special characters.
Fix: I fixed this problem by removing those Special Characters from PROD_SKU Column. The Queries of these fixes are mentioned below

Queries:

```
UPDATE DataSet3

SET PROD_SKU = left(PROD_SKU,8)

WHERE PROD_SKU in (SELECT PROD_SKU FROM DataSet3 where PROD_SKU like '%?')


UPDATE DataSet3

SET PROD_SKU = left(PROD_SKU,8)

WHERE PROD_SKU in (SELECT PROD_SKU FROM DataSet3 where PROD_SKU like '%Â§')



UPDATE DataSet3

SET PROD_SKU = left(PROD_SKU,8)

WHERE PROD_SKU in (SELECT PROD_SKU FROM DataSet3 where PROD_SKU like '%§')


UPDATE DataSet3

SET PROD_SKU = left(PROD_SKU,8)

WHERE PROD_SKU in (SELECT PROD_SKU FROM DataSet3 where PROD_SKU like '%Ã»')


UPDATE DataSet3

SET PROD_SKU = left(PROD_SKU,8)

WHERE PROD_SKU in (SELECT PROD_SKU FROM DataSet3 where PROD_SKU like '%û')
```

Problem3: In DataSet3 few records had VEND_NAME and VEND_STREET fields as blanks.

Fix: When I closely looked at these records all these records are of Vendor with VENDOR_ID as 15. I fixed this problem by updating VEND_NAME and VEND_STREET field of these records with VENDOR_ID= 15 details. The Queries of these fixes are mentioned below

Queries:

```
UPDATE DataSet3

SET VEND_NAME = 'Unlimited Wholesale of Ohio'

WHERE VEND_NAME =''


UPDATE DataSet3

SET VEND_STREET = '454 WINDJAMMER CIRCLE'

WHERE VEND_STREET =''
```

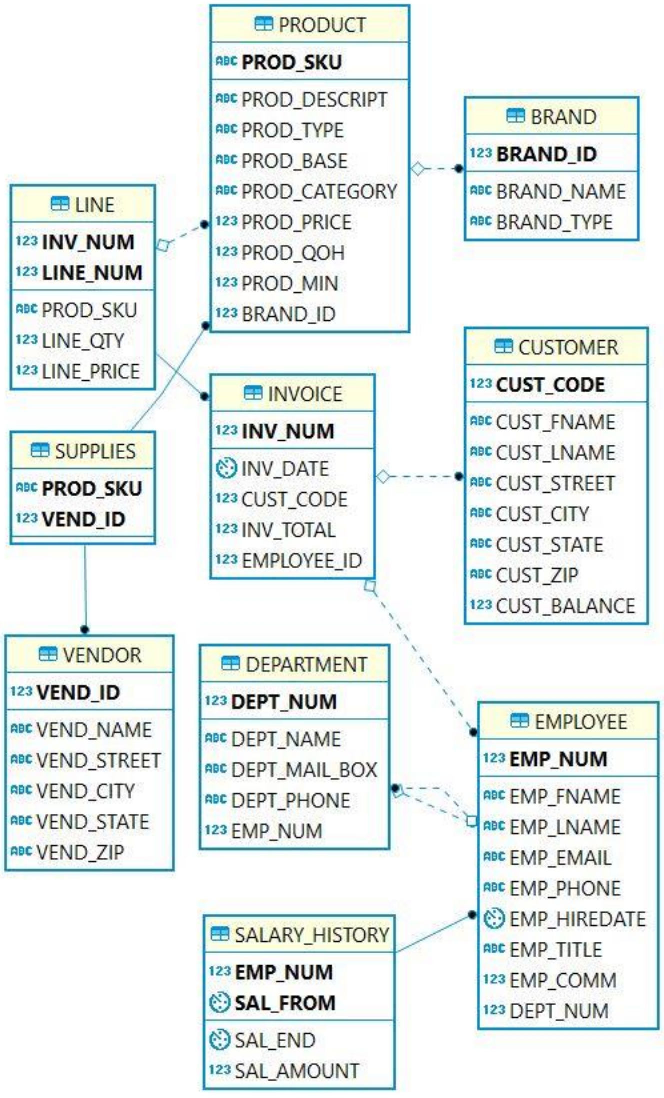
3. Database Implementation and Data import to tables:

For implementing the Database as per the Data Dictionary provided, I choose Microsoft SQL Server. I defined the tables with appropriate Primary Keys, Foreign keys and uploaded data into these tables from, initially uploaded and cleaned Datasets from Flat files.

The Data Definition and Data insertion queries used are mentioned below:

```
CREATE TABLE BRAND(
BRAND_ID INTEGER NOT NULL UNIQUE,
BRAND_NAME VARCHAR(100) NOT NULL,
BRAND_TYPE VARCHAR(20) NOT NULL,
PRIMARY KEY (BRAND_ID));

INSERT INTO BRAND (BRAND_ID, BRAND_NAME, BRAND_TYPE)
SELECT distinct BRAND_ID, BRAND_NAME, BRAND_TYPE FROM DataSet3;

CREATE TABLE CUSTOMER(
CUST_CODE INTEGER NOT NULL UNIQUE,
CUST_FNAME VARCHAR(20),
CUST_LNAME VARCHAR(20),
CUST_STREET VARCHAR(70),
CUST_CITY VARCHAR(50),
CUST_STATE VARCHAR(2),
CUST_ZIP VARCHAR(5),
CUST_BALANCE numeric(16,4)
PRIMARY KEY (CUST_CODE));

INSERT INTO CUSTOMER (CUST_CODE, CUST_FNAME, CUST_LNAME, CUST_STREET, CUST_CITY,
CUST_STATE, CUST_ZIP, CUST_BALANCE)
SELECT distinct CUST_CODE, CUST_FNAME, CUST_LNAME, CUST_STREET, CUST_CITY, CUST_STATE,
CUST_ZIP, CUST_BALANCE FROM DataSet2;

CREATE TABLE DEPARTMENT(
DEPT_NUM INTEGER NOT NULL UNIQUE,
DEPT_NAME VARCHAR(50),
DEPT_MAIL_BOX VARCHAR(3),
DEPT_PHONE VARCHAR(9),
EMP_NUM INTEGER
PRIMARY KEY (DEPT_NUM));

INSERT INTO DEPARTMENT (DEPT_NUM, DEPT_NAME, DEPT_MAIL_BOX, DEPT_PHONE, EMP_NUM)
SELECT distinct DEPT_NUM, DEPT_NAME, DEPT_MAIL_BOX, DEPT_PHONE, SUPV_EMP_NUM FROM
DataSet4;

CREATE TABLE EMPLOYEE(
EMP_NUM INTEGER NOT NULL UNIQUE,
EMP_FNAME VARCHAR(20),
EMP_LNAME VARCHAR(25),
EMP_EMAIL VARCHAR(25),
EMP_PHONE VARCHAR(20),
EMP_HIREDATE datetime,
EMP_TITLE VARCHAR(45),
EMP_COMM numeric(16,4),
DEPT_NUM INTEGER
PRIMARY KEY (EMP_NUM),
FOREIGN KEY (DEPT_NUM) REFERENCES DEPARTMENT );

CREATE INDEX EMPLOYEE_IDX
on EMPLOYEE (DEPT_NUM, EMP_NUM);

INSERT INTO EMPLOYEE (EMP_NUM, EMP_FNAME, EMP_LNAME, EMP_EMAIL, EMP_PHONE, EMP_HIREDATE,
EMP_TITLE, EMP_COMM, DEPT_NUM)
SELECT distinct EMP_NUM, EMP_FNAME, EMP_LNAME, EMP_EMAIL, EMP_PHONE,
cast(cast(EMP_HIREDATE as int) as datetime), EMP_TITLE, EMP_COMM, DEPT_NUM FROM DataSet4;

ALTER TABLE DEPARTMENT
ADD FOREIGN KEY (EMP_NUM) REFERENCES EMPLOYEE;

CREATE TABLE INVOICE(
INV_NUM INTEGER NOT NULL UNIQUE,
INV_DATE datetime,
CUST_CODE INTEGER NOT NULL,
INV_TOTAL numeric(16,4),
EMPLOYEE_ID INTEGER
PRIMARY KEY (INV_NUM),
FOREIGN KEY (CUST_CODE) REFERENCES CUSTOMER,
FOREIGN KEY (EMPLOYEE_ID) REFERENCES EMPLOYEE(EMP_NUM));

CREATE INDEX INVOICE_IDX
on INVOICE (CUST_CODE, EMPLOYEE_ID, INV_NUM);

INSERT INTO INVOICE (INV_NUM, INV_DATE, CUST_CODE, INV_TOTAL, EMPLOYEE_ID)
SELECT distinct INV_NUM, cast(cast(INV_DATE as int) as datetime), CUST_CODE, INV_TOTAL,
EMPLOYEE_ID FROM DataSet2;

CREATE TABLE PRODUCT(
PROD_SKU VARCHAR(15) NOT NULL UNIQUE,
PROD_DESCRIPT VARCHAR(255),
PROD_TYPE VARCHAR(255),
PROD_BASE VARCHAR(255),
PROD_CATEGORY VARCHAR(255),
PROD_PRICE numeric(16,4),
PROD_QOH numeric(16,4),
PROD_MIN numeric(16,4),
BRAND_ID INTEGER
PRIMARY KEY (PROD_SKU),
FOREIGN KEY (BRAND_ID) REFERENCES BRAND);

INSERT INTO PRODUCT (PROD_SKU, PROD_DESCRIPT, PROD_TYPE, PROD_BASE, PROD_CATEGORY,
PROD_PRICE, PROD_QOH, PROD_MIN, BRAND_ID)
SELECT distinct PROD_SKU, PROD_DESCRIPT, PROD_TYPE, PROD_BASE, PROD_CATEGORY, PROD_PRICE,
PROD_QOH, PROD_MIN, BRAND_ID FROM DataSet3;

CREATE TABLE LINE(
INV_NUM INTEGER NOT NULL,
LINE_NUM INTEGER NOT NULL,
PROD_SKU VARCHAR(15),
LINE_QTY BIGINT,
LINE_PRICE numeric(16,4),
PRIMARY KEY (INV_NUM, LINE_NUM),
FOREIGN KEY (INV_NUM)  REFERENCES INVOICE,
FOREIGN KEY (PROD_SKU) REFERENCES PRODUCT,
CONSTRAINT LINE_UPK UNIQUE(INV_NUM, LINE_NUM));

INSERT INTO LINE (INV_NUM, LINE_NUM, PROD_SKU, LINE_QTY, LINE_PRICE)
SELECT distinct INV_NUM, LINE_NUM, PROD_SKU, LINE_QTY, LINE_PRICE FROM DataSet2;

CREATE TABLE  SALARY_HISTORY(
EMP_NUM INTEGER NOT NULL,
SAL_FROM datetime NOT NULL,
SAL_END datetime,
SAL_AMOUNT numeric(16,4),
PRIMARY KEY (EMP_NUM, SAL_FROM),
FOREIGN KEY (EMP_NUM) REFERENCES EMPLOYEE,
CONSTRAINT SALARY_HISTORY_UPK UNIQUE(EMP_NUM, SAL_FROM));

CREATE INDEX SALARY_HISTORY_IDX
on SALARY_HISTORY (EMP_NUM);

INSERT INTO SALARY_HISTORY (EMP_NUM, SAL_FROM, SAL_END, SAL_AMOUNT)
SELECT distinct EMP_NUM, cast(cast(SAL_FROM as int) as datetime), cast(cast(SAL_END as
int) as datetime), SAL_AMOUNT FROM DataSet4;

UPDATE SALARY_HISTORY
SET SAL_END = NULL
WHERE SAL_END = '1900-01-01 00:00:00.000'

CREATE TABLE VENDOR(
VEND_ID INTEGER NOT NULL UNIQUE,
VEND_NAME VARCHAR(255),
VEND_STREET VARCHAR(50),
VEND_CITY VARCHAR(50),
VEND_STATE VARCHAR(2),
VEND_ZIP VARCHAR(5)
PRIMARY KEY (VEND_ID));

INSERT INTO VENDOR (VEND_ID, VEND_NAME, VEND_STREET, VEND_CITY, VEND_STATE, VEND_ZIP)
SELECT distinct VEND_ID, VEND_NAME, VEND_STREET, VEND_CITY, VEND_STATE, VEND_ZIP FROM
DataSet3;

CREATE TABLE  SUPPLIES(
PROD_SKU VARCHAR(15) NOT NULL,
VEND_ID INTEGER NOT NULL,
PRIMARY KEY (PROD_SKU, VEND_ID),
FOREIGN KEY (PROD_SKU) REFERENCES PRODUCT,
FOREIGN KEY (VEND_ID) REFERENCES VENDOR,
CONSTRAINT SUPPLIES_UPK UNIQUE(PROD_SKU, VEND_ID));

INSERT INTO SUPPLIES (PROD_SKU, VEND_ID)
SELECT distinct PROD_SKU, VEND_ID FROM DataSet3;
```

The Schema of the Database can be found below:



4. Answering the given questions with SQL Queries:

The answers for all the 10 questions were provided below.

Question1:

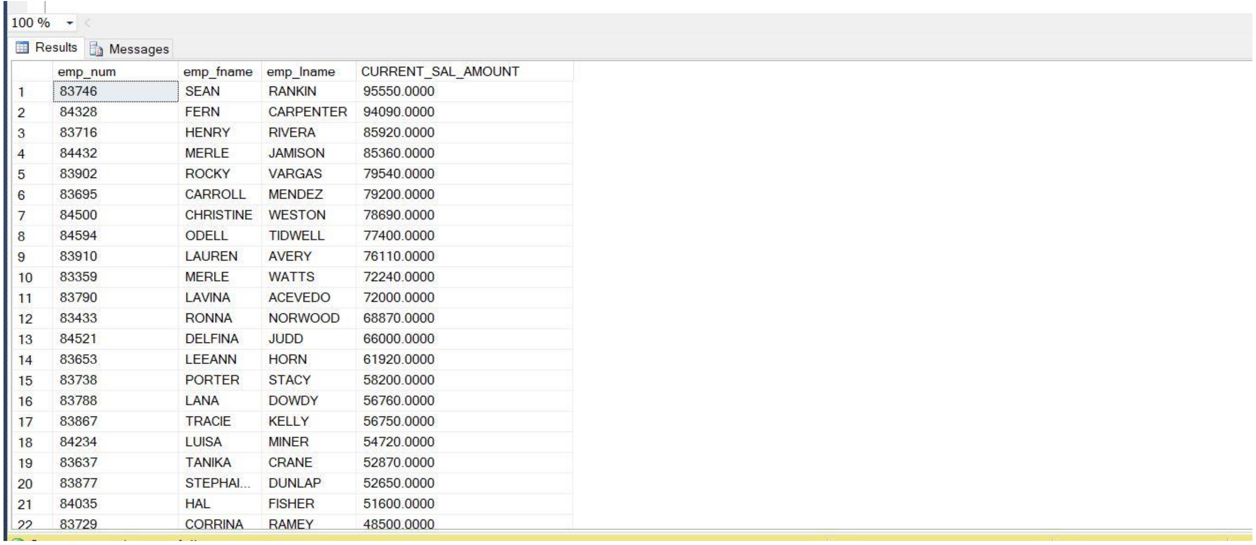
Query:

```
SELECT E.emp_num,
       E.emp_fname,
       E.emp_lname,
       SH.sal_amount AS CURRENT_SAL_AMOUNT
FROM   salary_history SH
       INNER JOIN employee E
               ON SH.emp_num = E.emp_num
WHERE  E.dept_num = '300'
       AND SH.sal_end IS NULL
ORDER  BY SH.sal_amount DESC
```

Output:

Question_1_Query_Output.xlsx

Sample Output:



100 %

Results Messages

	emp_num	emp_fname	emp_lname	CURRENT_SAL_AMOUNT
1	83746	SEAN	RANKIN	95550.0000
2	84328	FERN	CARPENTER	94090.0000
3	83716	HENRY	RIVERA	85920.0000
4	84432	MERLE	JAMISON	85360.0000
5	83902	ROCKY	VARGAS	79540.0000
6	83695	CARROLL	MENDEZ	79200.0000
7	84500	CHRISTINE	WESTON	78690.0000
8	84594	ODELL	TIDWELL	77400.0000
9	83910	LAUREN	AVERY	76110.0000
10	83359	MERLE	WATTS	72240.0000
11	83790	LAVINA	ACEVEDO	72000.0000
12	83433	RONNA	NORWOOD	68870.0000
13	84521	DELFINA	JUDD	66000.0000
14	83653	LEEANN	HORN	61920.0000
15	83738	PORTER	STACY	58200.0000
16	83788	LANA	DOWDY	56760.0000
17	83867	TRACIE	KELLY	56750.0000
18	84234	LUISA	MINER	54720.0000
19	83637	TANIKA	CRANE	52870.0000
20	83877	STEPHAI...	DUNLAP	52650.0000
21	84035	HAL	FISHER	51600.0000
22	83729	CORRINA	RAMEY	48500.0000

Question2:

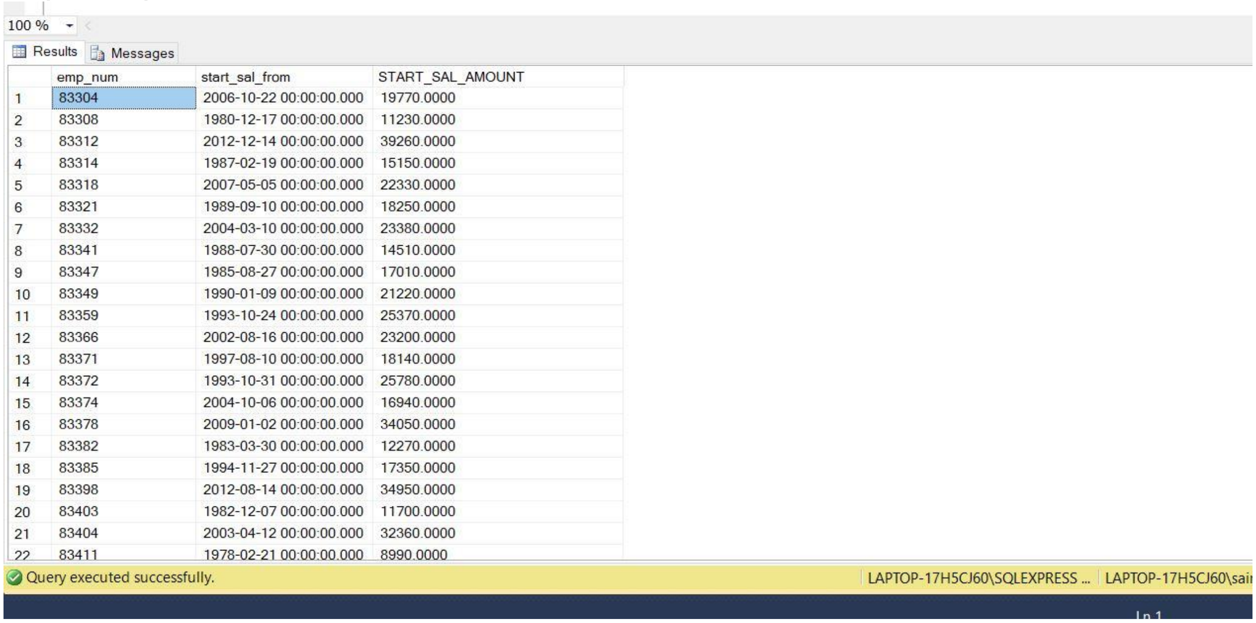
Query:

```
SELECT SH.emp_num,
       MSH.start_sal_from,
       SH.sal_amount AS START_SAL_AMOUNT
FROM   salary_history SH
       INNER JOIN (SELECT emp_num,
                          Min(sal_from) AS START_SAL_FROM
                   FROM   salary_history
                   GROUP  BY emp_num) MSH
               ON SH.emp_num = MSH.emp_num
                  AND SH.sal_from = MSH.start_sal_from
ORDER  BY SH.emp_num
```

Output:

Question_2_Query_Output.xlsx

Sample Output:



	emp_num	start_sal_from	START_SAL_AMOUNT
1	83304	2006-10-22 00:00:00.000	19770.0000
2	83308	1980-12-17 00:00:00.000	11230.0000
3	83312	2012-12-14 00:00:00.000	39260.0000
4	83314	1987-02-19 00:00:00.000	15150.0000
5	83318	2007-05-05 00:00:00.000	22330.0000
6	83321	1989-09-10 00:00:00.000	18250.0000
7	83332	2004-03-10 00:00:00.000	23380.0000
8	83341	1988-07-30 00:00:00.000	14510.0000
9	83347	1985-08-27 00:00:00.000	17010.0000
10	83349	1990-01-09 00:00:00.000	21220.0000
11	83359	1993-10-24 00:00:00.000	25370.0000
12	83366	2002-08-16 00:00:00.000	23200.0000
13	83371	1997-08-10 00:00:00.000	18140.0000
14	83372	1993-10-31 00:00:00.000	25780.0000
15	83374	2004-10-06 00:00:00.000	16940.0000
16	83378	2009-01-02 00:00:00.000	34050.0000
17	83382	1983-03-30 00:00:00.000	12270.0000
18	83385	1994-11-27 00:00:00.000	17350.0000
19	83398	2012-08-14 00:00:00.000	34950.0000
20	83403	1982-12-07 00:00:00.000	11700.0000
21	83404	2003-04-12 00:00:00.000	32360.0000
22	83411	1978-02-21 00:00:00.000	8990.0000

Query executed successfully. LAPTOP-17H5CJ60\SQLEXPRESS ... LAPTOP-17H5CJ60\sai

Ln 1

Question3:

Query:

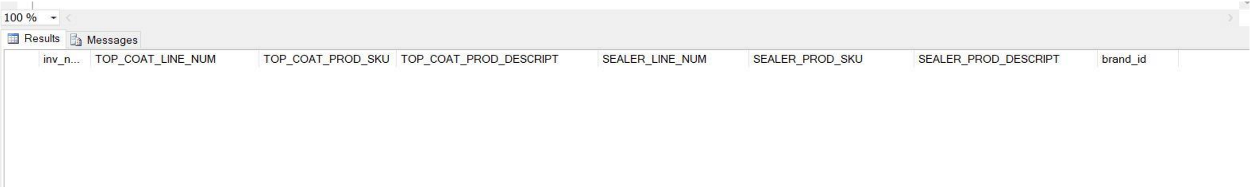
```
SELECT STC.inv_num,
       STC.line_num      AS TOP_COAT_LINE_NUM,
       STC.prod_sku      AS TOP_COAT_PROD_SKU,
       STC.prod_descript AS TOP_COAT_PROD_DESCRIPT,
       SS.line_num       AS SEALER_LINE_NUM,
       SS.prod_sku       AS SEALER_PROD_SKU,
       SS.prod_descript  AS SEALER_PROD_DESCRIPT,
       SS.brand_id
FROM   (SELECT L.inv_num,
               L.line_num,
               P.prod_sku,
               P.prod_descript,
               P.brand_id
        FROM   line L
               INNER JOIN product P
                       ON P.prod_sku = L.prod_sku
        WHERE  P.prod_category IN ( 'Top Coat' )) STC
       INNER JOIN (SELECT L.inv_num,
                          L.line_num,
                          P.prod_sku,
                          P.prod_descript,
                          P.brand_id
                   FROM   line L
                          INNER JOIN product P
                                  ON P.prod_sku = L.prod_sku
                   WHERE  P.prod_category IN ( 'Sealer' )) SS
               ON STC.inv_num = SS.inv_num
```

```
                  AND STC.brand_id = SS.brand_id
```

Output:

There is no output for this Query since there is no data as per question criteria

Sample Output:



inv_n...	TOP_COAT_LINE_NUM	TOP_COAT_PROD_SKU	TOP_COAT_PROD_DESCRIPT	SEALER_LINE_NUM	SEALER_PROD_SKU	SEALER_PROD_DESCRIPT	brand_id

Question4:

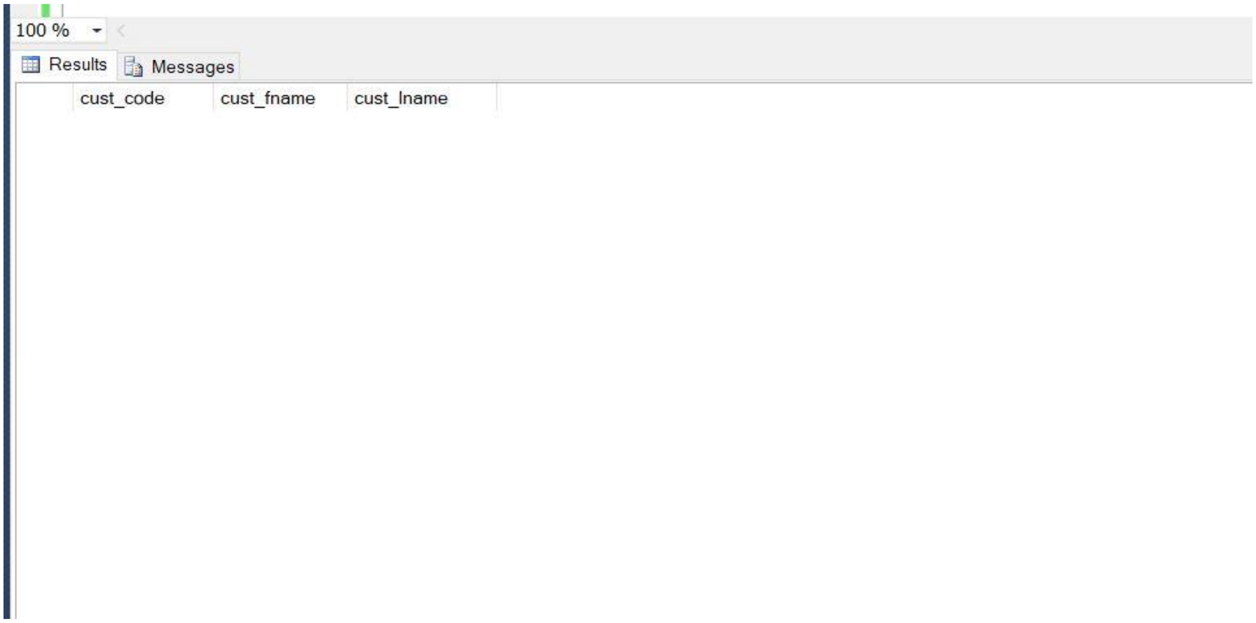
Query:

```
SELECT TOP 1 E.emp_num,
             E.emp_fname,
             E.emp_lname,
             E.emp_email,
             Sum(L.line_qty) AS TOTAL_UNITS_SOLD
FROM   employee E
       INNER JOIN invoice I
               ON E.emp_num = I.employee_id
       INNER JOIN line L
               ON I.inv_num = L.inv_num
       INNER JOIN product P
               ON L.prod_sku = P.prod_sku
       INNER JOIN brand B
               ON P.brand_id = B.brand_id
WHERE  brand_name = 'BINDER PRIME'
       AND I.inv_date BETWEEN '2015-11-01 00:00:00.000' AND
                              '2015-12-05 00:00:00.000'
GROUP  BY E.emp_num,
          E.emp_fname,
          E.emp_lname,
          E.emp_email
ORDER  BY total_units_sold DESC,
          emp_lname
```

Output:

There is no output for this Query since there is no data as per question criteria

Sample Output:



	cust_code	cust_fname	cust_lname

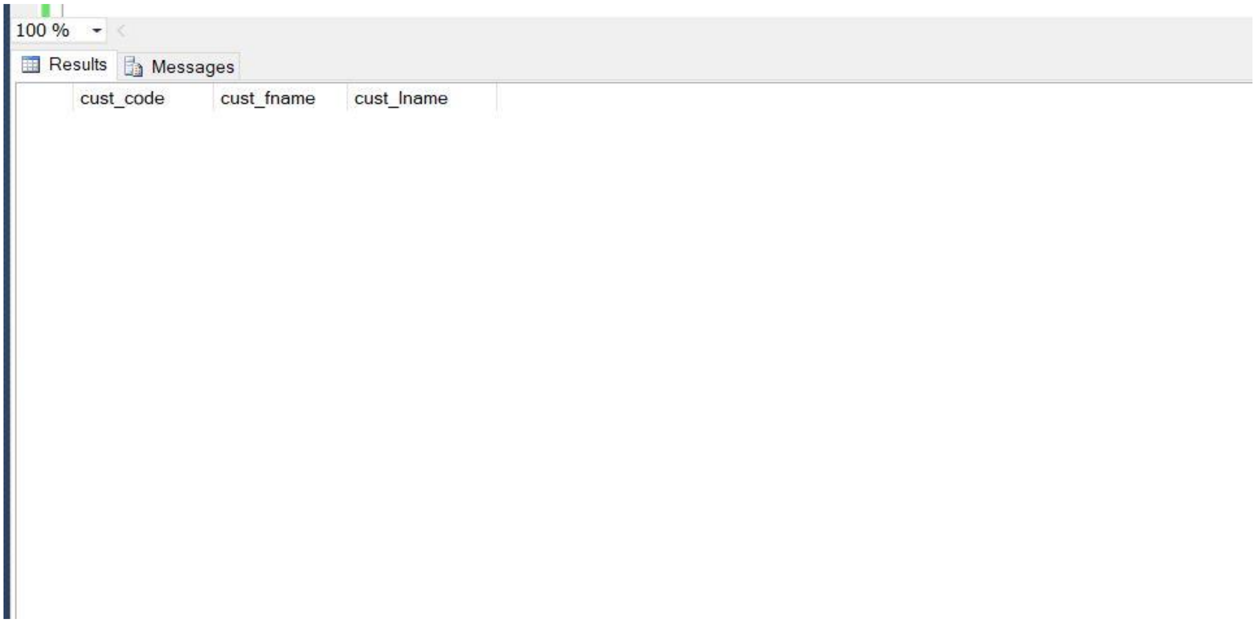
Question5:

Query:

```
SELECT C.cust_code,
       C.cust_fname,
       C.cust_lname
FROM   customer C
       INNER JOIN invoice I
               ON C.cust_code = I.cust_code
WHERE  I.cust_code IN (SELECT cust_code
                       FROM   invoice
                       WHERE  employee_id = '83649')
       AND I.employee_id = '83677'
ORDER  BY C.cust_lname,
          C.cust_fname
```

Output:
There is no output for this Query since there is no data as per question criteria

Sample Output:



cust_code	cust_fname	cust_lname

Question6:

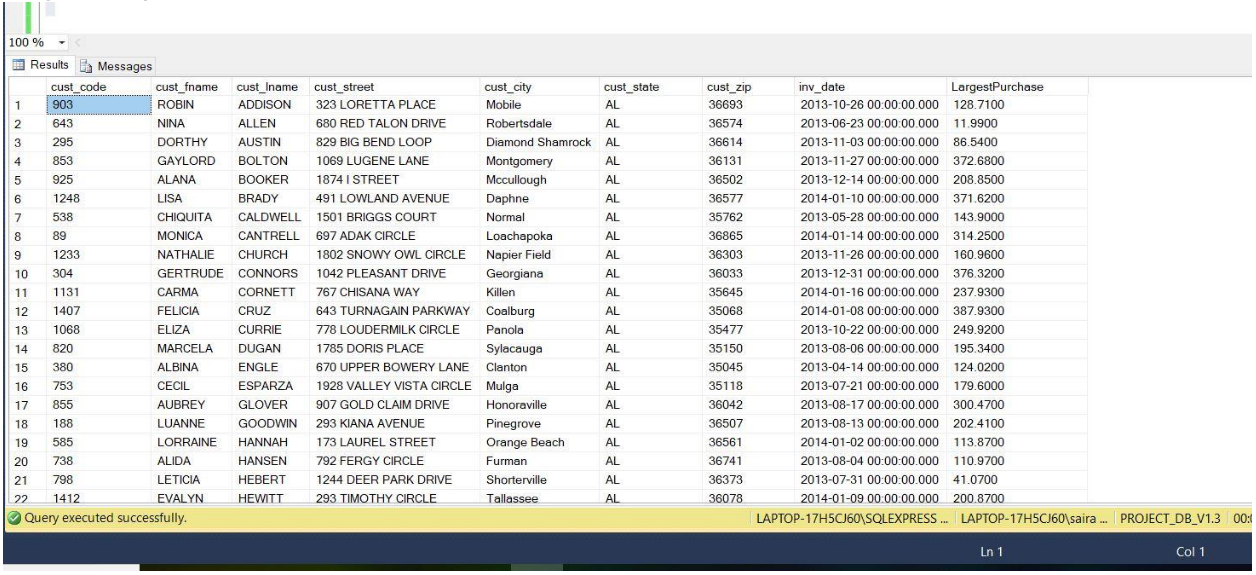
Query:

```
SELECT C.cust_code,
       C.cust_fname,
       C.cust_lname,
       C.cust_street,
       C.cust_city,
       C.cust_state,
       C.cust_zip,
       I.inv_date,
       Isnull(I.inv_total, 0) AS LargestPurchase
FROM   customer C
       INNER JOIN invoice I
               ON C.cust_code = I.cust_code
WHERE  C.cust_state = 'AL'
       AND I.inv_total = (SELECT Max(IM.inv_total)
                          FROM   invoice IM
                          WHERE  IM.cust_code = C.cust_code)
UNION ALL
SELECT C.cust_code,
       C.cust_fname,
       C.cust_lname,
       C.cust_street,
       C.cust_city,
       C.cust_state,
       C.cust_zip,
       NULL,
       0
FROM   customer c
WHERE  C.cust_state = 'AL'
       AND C.cust_code NOT IN (SELECT cust_code
                               FROM   invoice)
ORDER  BY C.cust_lname,
          C.cust_fname;
```

Output:

Question_6_Query_Output.xlsx

Sample Output:



	cust_code	cust_fname	cust_lname	cust_street	cust_city	cust_state	cust_zip	inv_date	LargestPurchase
1	903	ROBIN	ADDISON	323 LORETTA PLACE	Mobile	AL	36693	2013-10-26 00:00:00.000	128.7100
2	643	NINA	ALLEN	680 RED TALON DRIVE	Robertsdale	AL	36574	2013-06-23 00:00:00.000	11.9900
3	295	DORTHY	AUSTIN	829 BIG BEND LOOP	Diamond Shamrock	AL	36614	2013-11-03 00:00:00.000	86.5400
4	853	GAYLORD	BOLTON	1069 LUGENE LANE	Montgomery	AL	36131	2013-11-27 00:00:00.000	372.6800
5	925	ALANA	BOOKER	1874 I STREET	Mccullough	AL	36502	2013-12-14 00:00:00.000	208.8500
6	1248	LISA	BRADY	491 LOWLAND AVENUE	Daphne	AL	36577	2014-01-10 00:00:00.000	371.6200
7	538	CHIQUITA	CALDWELL	1501 BRIGGS COURT	Normal	AL	35762	2013-05-28 00:00:00.000	143.9000
8	89	MONICA	CANTRELL	697 ADAK CIRCLE	Loachapoka	AL	36865	2014-01-14 00:00:00.000	314.2500
9	1233	NATHALIE	CHURCH	1802 SNOWY OWL CIRCLE	Napier Field	AL	36303	2013-11-26 00:00:00.000	160.9600
10	304	GERTRUDE	CONNORS	1042 PLEASANT DRIVE	Georgiana	AL	36033	2013-12-31 00:00:00.000	376.3200
11	1131	CARMA	CORNETT	767 CHISANA WAY	Killen	AL	35645	2014-01-16 00:00:00.000	237.9300
12	1407	FELICIA	CRUZ	643 TURNAGAIN PARKWAY	Coalburg	AL	35068	2014-01-08 00:00:00.000	387.9300
13	1068	ELIZA	CURRIE	778 LOUDERMILK CIRCLE	Panola	AL	35477	2013-10-22 00:00:00.000	249.9200
14	820	MARCELA	DUGAN	1785 DORIS PLACE	Sylacauga	AL	35150	2013-08-06 00:00:00.000	195.3400
15	380	ALBINA	ENGLE	670 UPPER BOWERY LANE	Clanton	AL	35045	2013-04-14 00:00:00.000	124.0200
16	753	CECIL	ESPARZA	1928 VALLEY VISTA CIRCLE	Mulga	AL	35118	2013-07-21 00:00:00.000	179.6000
17	855	AUBREY	GLOVER	907 GOLD CLAIM DRIVE	Honoraville	AL	36042	2013-08-17 00:00:00.000	300.4700
18	188	LUANNE	GOODWIN	293 KIANA AVENUE	Pinegrove	AL	36507	2013-08-13 00:00:00.000	202.4100
19	585	LORRAINE	HANNAH	173 LAUREL STREET	Orange Beach	AL	36561	2014-01-02 00:00:00.000	113.8700
20	738	ALIDA	HANSEN	792 FERGY CIRCLE	Furman	AL	36741	2013-08-04 00:00:00.000	110.9700
21	798	LETICIA	HEBERT	1244 DEER PARK DRIVE	Shorterville	AL	36373	2013-07-31 00:00:00.000	41.0700
22	1412	EVALYN	HEWITT	293 TIMOTHY CIRCLE	Tallassee	AL	36078	2014-01-09 00:00:00.000	200.8700

```
Query executed successfully.    LAPTOP-17H5CJ60\SQLEXPRESS ...  LAPTOP-17H5CJ60\saira ...  PROJECT_DB_V1.3  00:
                                                                Ln 1                Col 1
```

Question7:

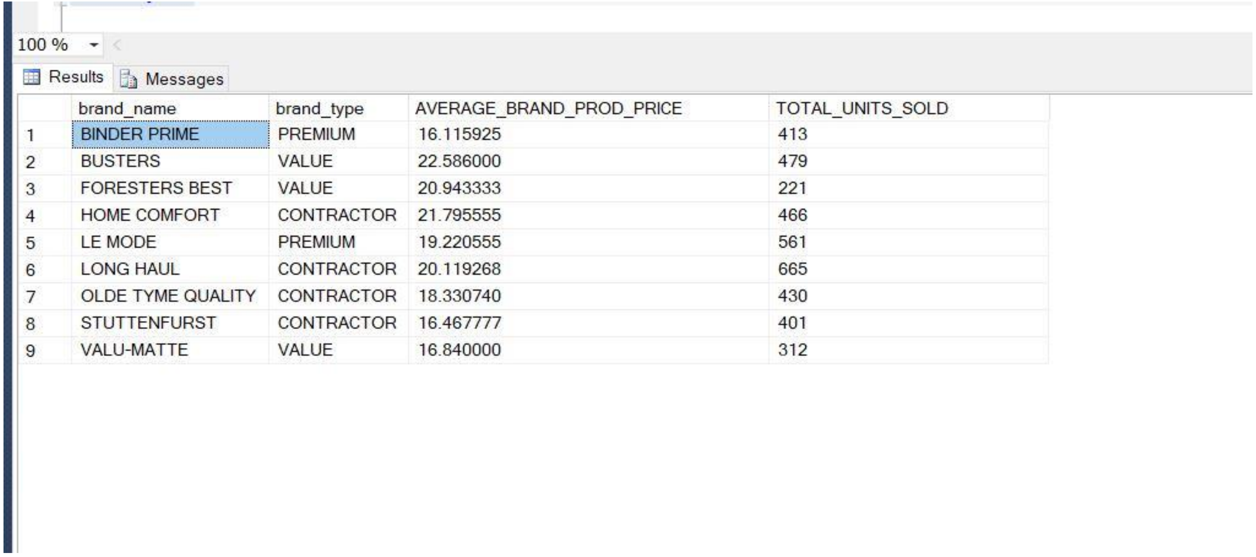
Query:

```
SELECT PA.brand_name,
       PA.brand_type,
       Avg(PA.average_prod_price) AS AVERAGE_BRAND_PROD_PRICE,
       Sum(PA.total_units_sold)   AS TOTAL_UNITS_SOLD
FROM   (SELECT B.brand_name,
               B.brand_type,
               P.prod_sku,
               Avg(P.prod_price) AS AVERAGE_PROD_PRICE,
               Sum(L.line_qty)   AS TOTAL_UNITS_SOLD
        FROM   brand B
               FULL JOIN product P
                       ON B.brand_id = P.brand_id
               FULL JOIN line L
                      ON L.prod_sku = P.prod_sku
        GROUP  BY B.brand_name,
                  B.brand_type,
                  P.prod_sku) PA
GROUP  BY PA.brand_name,
          PA.brand_type
order by 1
```

Output:

Question_7_Query_Output.xlsx

Sample Output:



	brand_name	brand_type	AVERAGE_BRAND_PROD_PRICE	TOTAL_UNITS_SOLD
1	BINDER PRIME	PREMIUM	16.115925	413
2	BUSTERS	VALUE	22.586000	479
3	FORESTERS BEST	VALUE	20.943333	221
4	HOME COMFORT	CONTRACTOR	21.795555	466
5	LE MODE	PREMIUM	19.220555	561
6	LONG HAUL	CONTRACTOR	20.119268	665
7	OLDE TYME QUALITY	CONTRACTOR	18.330740	430
8	STUTTENFURST	CONTRACTOR	16.467777	401
9	VALU-MATTE	VALUE	16.840000	312

Question8:

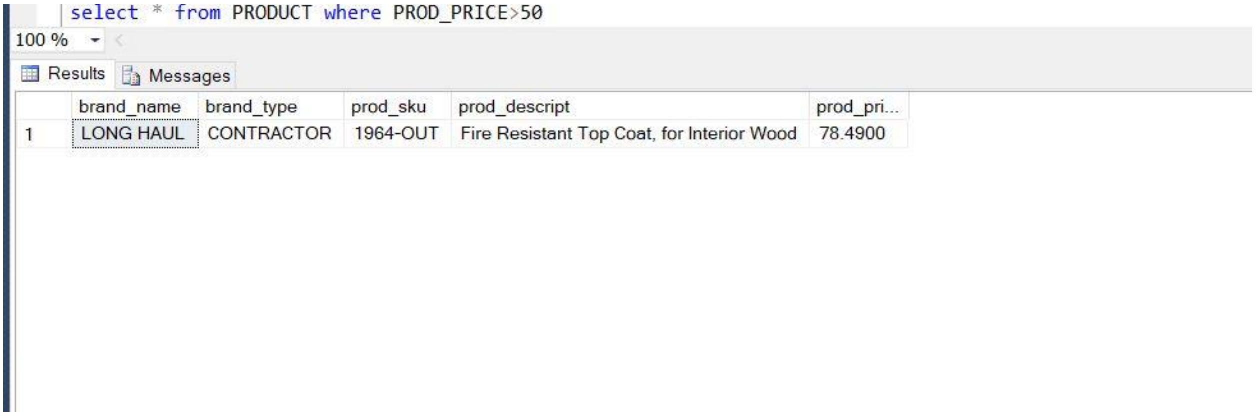
Query:

```sql
SELECT B.brand_name,
       B.brand_type,
       P.prod_sku,
       P.prod_descript,
       P.prod_price
FROM   brand B
       INNER JOIN product P
               ON B.brand_id = P.brand_id
WHERE  brand_type != 'PREMIUM'
       AND P.prod_price > (SELECT TOP 1 P.prod_price
                           FROM   brand B
                                  INNER JOIN product P
                                          ON B.brand_id = P.brand_id
                           WHERE  brand_type = 'PREMIUM'
                           ORDER  BY prod_price DESC)
ORDER  BY P.prod_price,
          B.brand_name,
          B.brand_type
```

Output:

Question_8_Query_Output.xlsx

Sample Output:



```
select * from PRODUCT where PROD_PRICE>50
```

100 %

Results | Messages

	brand_name	brand_type	prod_sku	prod_descript	prod_pri...
1	LONG HAUL	CONTRACTOR	1964-OUT	Fire Resistant Top Coat, for Interior Wood	78.4900

Question9:

Query:

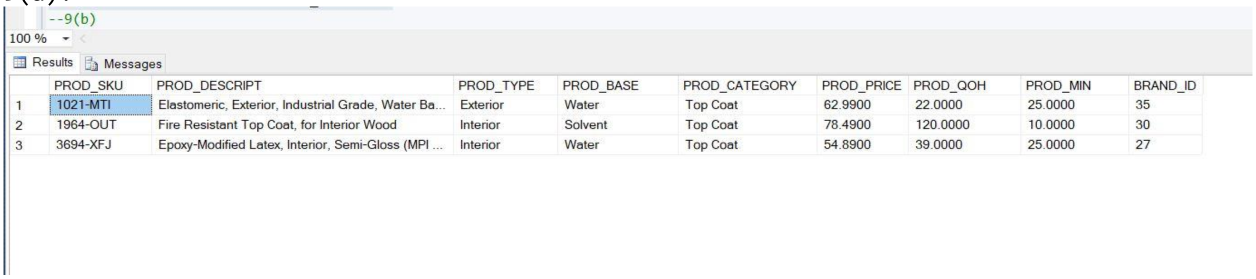
```
--9(a)
select * from PRODUCT where PROD_PRICE>50
--9(b)
select SUM(PROD_PRICE*PROD_QOH) as TOTAL_INVENTORY_VALUE from PRODUCT
--9(c)
select count(CUST_CODE) as NUM_OF_CUSTOMERS, sum(CUST_BALANCE) as TOTAL_CUST_BALANCE from
CUSTOMER
--9(d)
select top 3 CUST_STATE,SUM(INV_TOTAL) as TOTAL_SALES_Dollars from INVOICE I INNER JOIN
CUSTOMER C ON I.CUST_CODE=C.CUST_CODE group by CUST_STATE
order by SUM(INV_TOTAL) desc
```

Output:

Question_9_Query_Ou
tput.xlsx

Sample Output:

9(a):



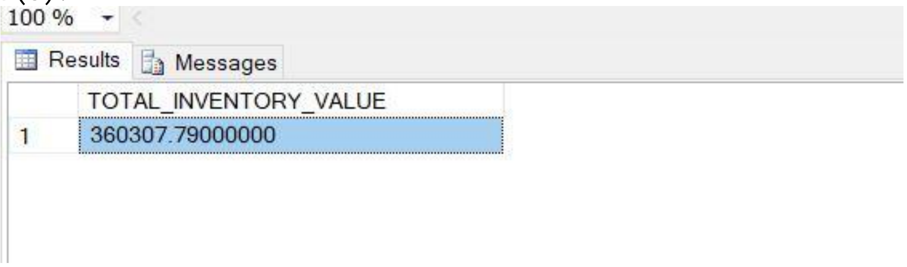
```
--9(b)
```

100 %

Results | Messages

	PROD_SKU	PROD_DESCRIPT	PROD_TYPE	PROD_BASE	PROD_CATEGORY	PROD_PRICE	PROD_QOH	PROD_MIN	BRAND_ID
1	1021-MTI	Elastomeric, Exterior, Industrial Grade, Water Ba...	Exterior	Water	Top Coat	62.9900	22.0000	25.0000	35
2	1964-OUT	Fire Resistant Top Coat, for Interior Wood	Interior	Solvent	Top Coat	78.4900	120.0000	10.0000	30
3	3694-XFJ	Epoxy-Modified Latex, Interior, Semi-Gloss (MPI ...	Interior	Water	Top Coat	54.8900	39.0000	25.0000	27

9(b):

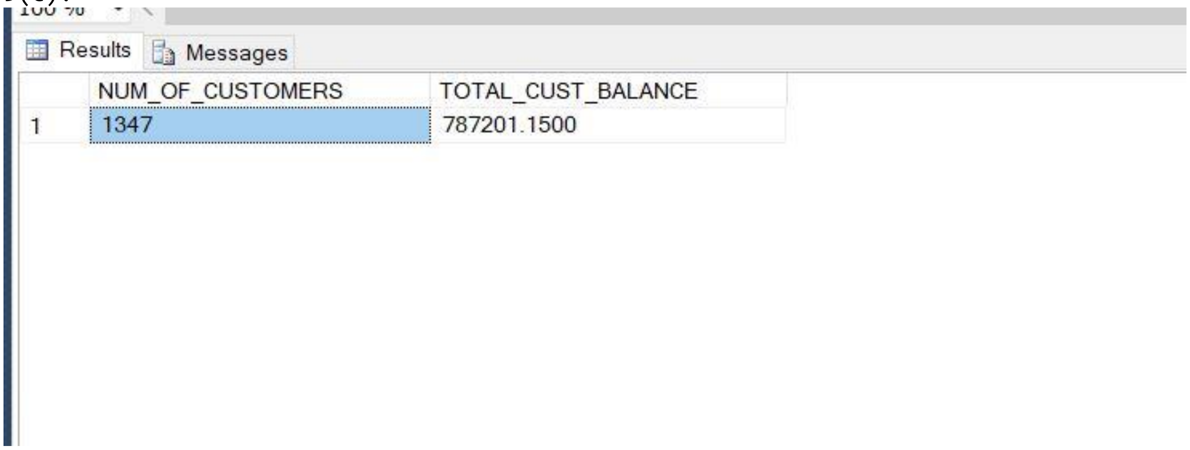


100 %

Results | Messages

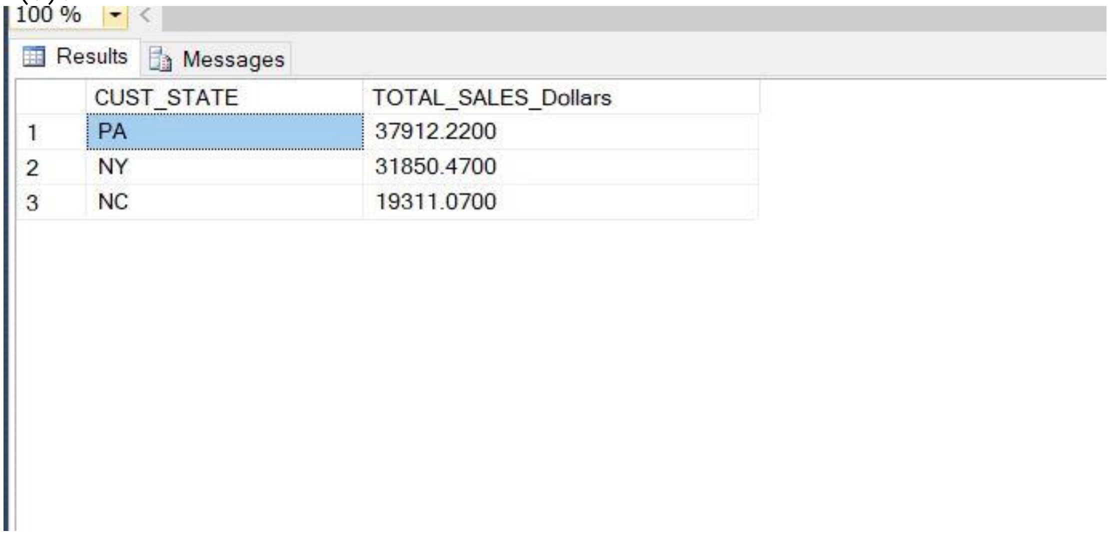
	TOTAL_INVENTORY_VALUE
1	360307.79000000

9(c):



	NUM_OF_CUSTOMERS	TOTAL_CUST_BALANCE
1	1347	787201.1500

9(d):



	CUST_STATE	TOTAL_SALES_Dollars
1	PA	37912.2200
2	NY	31850.4700
3	NC	19311.0700

Question10:

The Query to get data to perform Linear Regression is mentioned below. The interpretation of Regression and the forecasting results for next year are taken up in detail in section "Predicting forecast of sales".

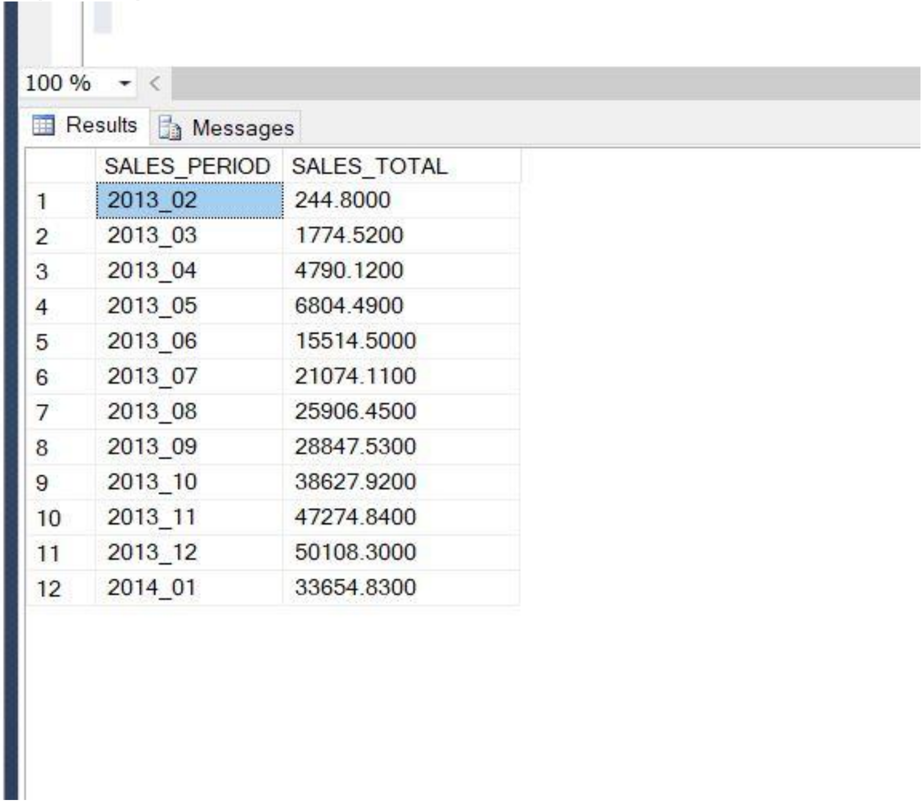
Query:

```
select FORMAT(INV_DATE, 'yyyy_MM') AS SALES_PERIOD,sum(INV_TOTAL) as SALES_TOTAL from
invoice
group by FORMAT(INV_DATE, 'yyyy_MM')
order by FORMAT(INV_DATE, 'yyyy_MM')
```

Output:

Question_10_Query_Output.xlsx

Sample Output:



	SALES_PERIOD	SALES_TOTAL
1	2013_02	244.8000
2	2013_03	1774.5200
3	2013_04	4790.1200
4	2013_05	6804.4900
5	2013_06	15514.5000
6	2013_07	21074.1100
7	2013_08	25906.4500
8	2013_09	28847.5300
9	2013_10	38627.9200
10	2013_11	47274.8400
11	2013_12	50108.3000
12	2014_01	33654.8300

5. Predicting forecast of sales:

For predicting the forecast of sales for the next year, I performed below steps

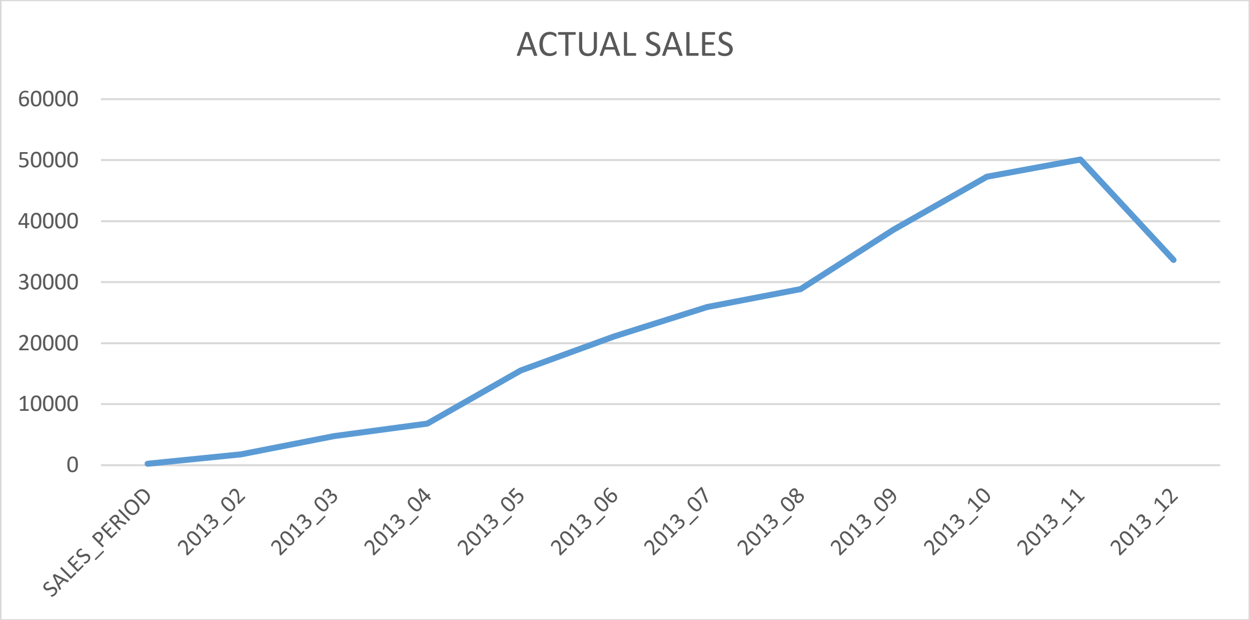
Step1:
I collected the previous monthly cumulative Sales data from the Invoice table by using the below query

```
select FORMAT(INV_DATE, 'yyyy_MM') AS SALES_PERIOD,sum(INV_TOTAL) as SALES_TOTAL from
invoice
group by FORMAT(INV_DATE, 'yyyy_MM')
order by FORMAT(INV_DATE, 'yyyy_MM')
```

The Output of the data is present in below file:

Question_10_Query_Output.xlsx

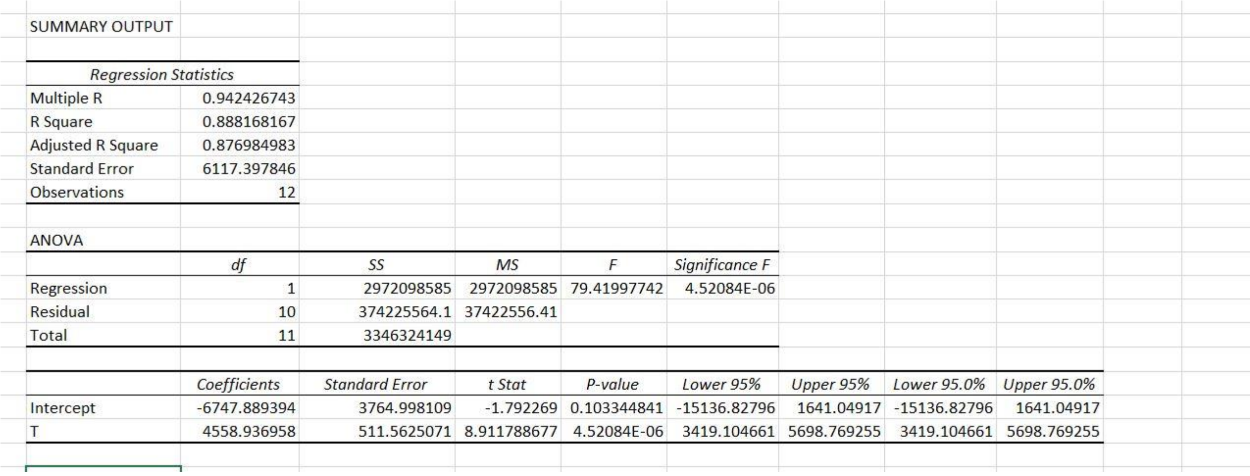
Step2:
I plotted a line graph for the collected data. And the graph can be seen below.



Step3:

I performed a Linear regression of sales(dependent) over the time period(Independent) using Excel.

The Result the linear regression is shown below:


SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.942426743
R Square	0.888168167
Adjusted R Square	0.876984983
Standard Error	6117.397846
Observations	12

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	2972098585	2972098585	79.41997742	4.52084E-06
Residual	10	374225564.1	37422556.41		
Total	11	3346324149			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	-6747.889394	3764.998109	-1.792269	0.103344841	-15136.82796	1641.04917	-15136.82796	1641.04917
T	4558.936958	511.5625071	8.911788677	4.52084E-06	3419.104661	5698.769255	3419.104661	5698.769255

Model:

*Sales = -6747.89 + 4558.94 * T*

R^2 *of the model is 0.88 which very good.*

Coefficients and confidence interval for the model can be found above in screenshot.

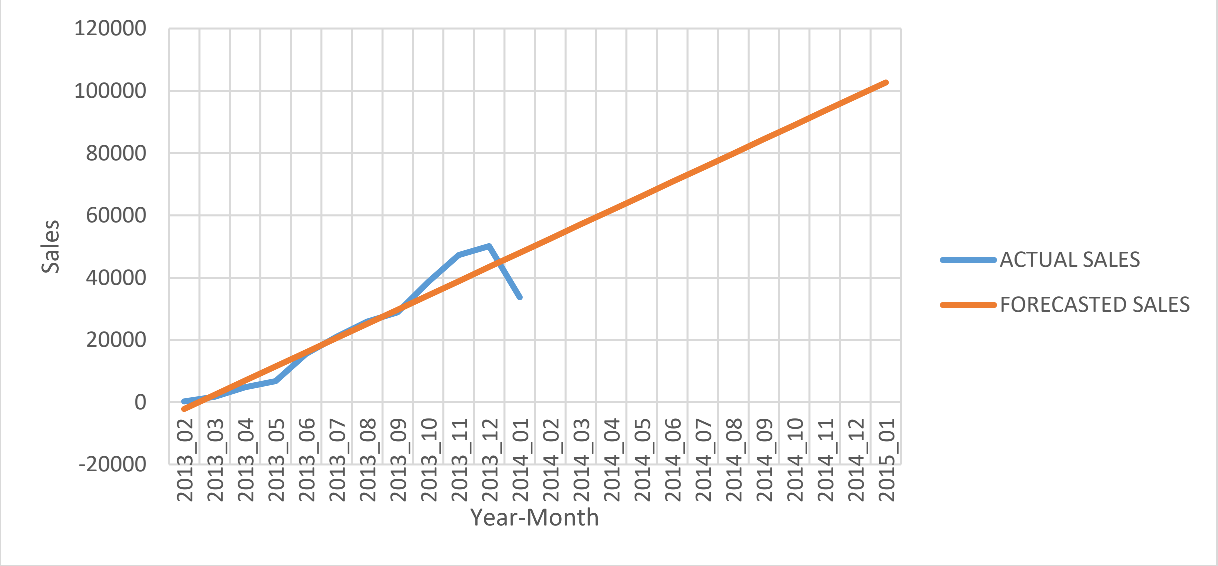
Step3:

Predicting forecast of sales for the next year using the model.

I calculated the forecast for next year and shown them below.

SALES_PERIOD	FORECASTED SALES
2014_02	52518.29106
2014_03	57077.22802
2014_04	61636.16498
2014_05	66195.10193
2014_06	70754.03889
2014_07	75312.97585
2014_08	79871.91281
2014_09	84430.84977
2014_10	88989.78672
2014_11	93548.72368
2014_12	98107.66064
2015_01	102666.5976
Total	**931109.332**

A Graph is plotted for Actual and forecasted sales below.


Sales
2013_02
2013_03
2013_04
2013_05
2013_06
2013_07
2013_08
2013_09
2013_10
2013_11
2013_12
2014_01
2014_02
2014_03
2014_04
2014_05
2014_06
2014_07
2014_08
2014_09
2014_10
2014_11
2014_12
2015_01
Year-Month
ACTUAL SALES
FORECASTED SALES