INDIVIDUAL REPORT

BUAN 6320
Database Foundations for Business Analytics

Vikram Arikath
VXA170003

Table of Contents

Introduction

This project has helped me understand the real-world application of a database and how it can be used to fetch specific results and use those results in prediction models to obtain the required output. It was an amazing experience working with the Customer Sales Data of the Large Co and to build it from the base in My SQL. The project helped me understand and implement the concepts studied throughout the course this semester. Not only did the project helped me increase my experience with SQL, but also made me work with R to obtain a more detailed outlook of the data. Implementation of statistical concepts were also pivotal in deciphering and acquiring the required output. The primary objectives of the project were:

- Normalizing the data.
- Cleaning the data.
- Creating a Logical and Physical model of the data.
- Fetching the required results with the specific and efficient queries.
- Implementing regression using R and arriving at a suitable forecast model.
- Analyzing the outputs as needed.
- Documenting the results and graphs obtained.

The values, outputs and graphs related to all the questions in the problem set are addressed in this report.

Problem Statement:

The data is primarily composed of 10 tables; Brand, Customer, Department, Employee, Invoice, Line, Salary History, Product, Vendor and Suppliers. It is the Customer Sales Data of Large Co. Using SQL, the desired results are obtained by implementing the specific queries with proper conditions and constraints. Information such as the prices, invoice values, most selling product are all obtained by proper and efficient querying. The data is further analyzed using R to forecast the sales of next year using linear regression.

Software/Tools used:

Microsoft Excel is the base tool used to normalize and store the data. The data is stored as tables in each spreadsheet for easy referencing and performing the cleaning process. Several redundant values, spaces, data types and other values are corrected before the data is taken for the next step. MySQl is the database system where the tables are exported from the Excel files by using a specific script. MySQL is where the major querying process is done to produce the results required for the problems given. The queries are made as efficient as possible to get fast and reliable results. Descriptive statistical analysis is also performed here analyze the data. Then, for the final problem R studio is used to run linear regression analysis to produce the required forecast from the data that is normalized and cleaned to process. For graphical representation of the Entity Relationship diagram ErWin is used to observe and understand how the tables are interconnected with one another.

Data Pre-Processing

Normalizing:

The initial data was provided as .txt files and with the help of the data dictionary given the data was split into 10 tables such as Brand, Customer, Department, Employee, Invoice, Line, Salary History, Product, Vendor and Supplies. The Brand table comprises of the brand-_id, brand_name and brand_type. The Customer table is made up of cust_code, cust_fname, cust_lname, cust_street, cust_city, cust_state and cust_zip. The Department table comprised of dept_num, dept_name, dept_mail_box, dept_phone and emp_num. The Employee table is made up of emp_num, emp_fname, emp_lname, emp_email, emp_phone, emp_hiredate and emp_title. The Invoice table comprises of inv_num, inv_date, cust_code, inv_total and employee_id. The Line table consists of inv_num, line_num, prod_sku, line_qty and line_price. The Product table comprises of prod_sku, prod_descript, prod_type, prod_base, prod_category, prod_price and prod_qoh. The Salary table is made up of emp_num, sal_from, sal_end and sal_amount. The Supplies table consists of prod_sku and vend_id. The final Vendors table is made up of vend_id, vend_name, vend_street, vend_city, vend_state and vend_zip. All these tables were tabulated and kept using Microsoft Excel.

```
BRAND:           BRAND_ID, BRAND_NAME, RAND_TYPE
CUSTOMER :       CUST_CODE, CUST_FNAME, CUST_LNAME, CUST_STREET, CUST_CITY, CUST_STATE, CUST_ZIP, CUST_BALANCE
DEPARTMENT:      DEPT_NUM, DEPT_NAME, DEPT_MAIL_BOX,    DEPT_PHONE, EMP_NUM
EMPLOYEE:        EMP_NUM, EMP_FNAME, EMP_LNAME, EMP_EMAIL, EMP_PHONE, EMP_HIREDATE, EMP_TITLE, EMP_COMM, DEPT_NUM
INVOICE:         INV_NUM, INV_DATE, CUST_CODE, INV_TOTAL, EMPLOYEE_ID
LINE:            INV_NUM, LINE_NUM, PROD_SKU, LINE_QTY, LINE_PRICE
PRODUCT:         PROD_SKU, PROD_DESCRIPT, PROD_TYPE, PROD_BASE, PROD_CATEGORY, PROD_PRICE, PROD_QOH, PROD_MIN, BRAND_ID
SALARY_HISTORY:  EMP_NUM, SAL_FROM, SAL_END, SAL_AMOUNT
SUPPLIER:        PROD_SKU, VEND_ID
VENDOR:          VEND_ID, VEND_NAME, VEND_STREET, VEND_CITY, VEND_STATE, VEND_ZIP
```

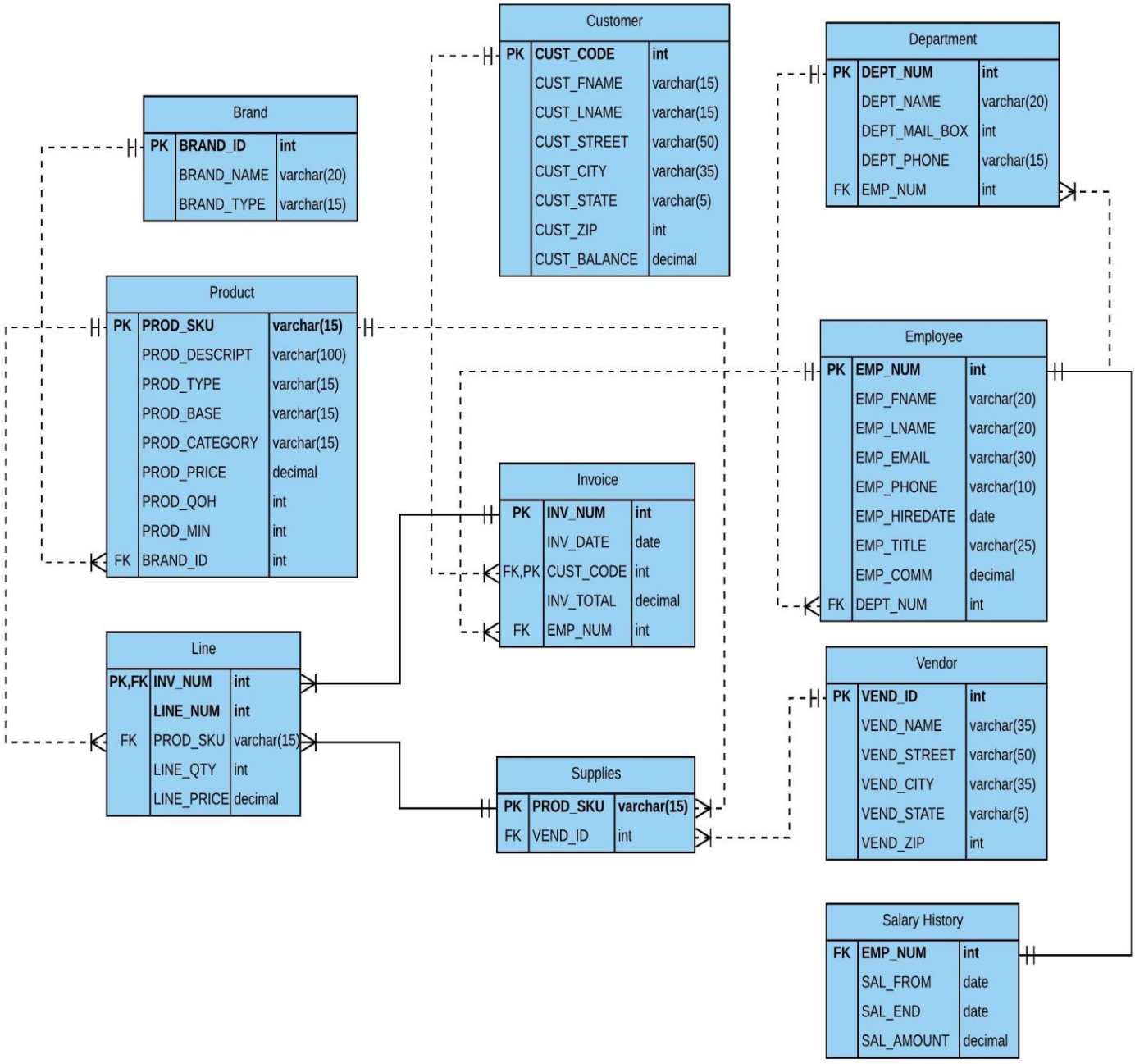
Cleaning Data:

The cleaning process was a bit complex due to the presence of dirty characters, different data types and redundant data. The initial .txt files dataset 2, dataset 3 and dataset 4 was changed format wise. The format of tab separated value was converted as comma separated value. The Primary and Foreign keys for each table were identified and proper relationship model was built. Then the repeating data records were eliminated to ensure greater accuracy while fetching the data for the required problems. Data formatting was primarily done by removing the extra spaces in the records. Decimal corrections were done to make sure there was uniformity. Date formats were changed to ensure that data could be processed without hassle. There were empty fields in vend_id which was removed. The cust_zip data was converted from 4 digits to 5 digits to make it uniform throughout and further converted from a string to numerical format. The date format had to be converted to YYYY-MM-DD format to ensure proper processing of data to fetch the results in My SQL and R. In the Salary History table columns with the value "-", were changed to NULL to make it into an acceptable form to input into the database. To shift the data from Microsoft Excel to My SQL the following script was used :

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/viki.csv'
INTO TABLE product
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

To use the data in R the data was split into training and test sets for processing the linear regression model and the results were interpreted accordingly providing us a better insight.

Entity Relationship Diagram



Questions and Answers

1. **Write a query to display the current salary for each employee in department 300. Assume that only current employees are kept in the system, and therefore the most current salary for each employee is the entry in the salary history with a NULL end date. Sort the output in descending order by salary amount.**

QUERY:

```
SELECT  emp_num, emp_fname, emp_lname, sal_amount
FROM Employee JOIN Salary_history
USING (emp_num)
WHERE sal_end IS NULL AND dept_num = 300
ORDER BY sal_amount DESC;
```

OUTPUT:

emp_num	emp_fname	emp_lname	sal_amount
83746	SEAN	RANKIN	95550.00
84328	FERN	CARPENTER	94090.00
83716	HENRY	RIVERA	85920.00
84432	MERLE	JAMISON	85360.00
83902	ROCKY	VARGAS	79540.00
83695	CARROLL	MENDEZ	79200.00
84500	CHRISTINE	WESTON	78690.00
84594	ODELL	TIDWELL	77400.00
83910	LAUREN	AVERY	76110.00
83359	MERLE	WATTS	72240.00
83790	LAVINA	ACEVEDO	72000.00

2. **Write a query to display the starting salary for each employee. The starting salary would be the entry in the salary history with the oldest salary start date for each employee. Sort the output by employee number.**

QUERY:

```
SELECT e.emp_num, emp_lname, emp_fname, sal_amount
FROM Employee e JOIN Salary_history s
ON e.emp_num = s.emp_num
WHERE sal_from = (SELECT Min(sal_from)
FROM Salary_history s2
WHERE e.emp_num = s2.emp_num)
ORDER BY e.emp_num;
```

OUTPUT:

emp_num	emp_lname	emp_fname	sal_amount
83304	MCDONALD	TAMARA	19770.00
83308	LOVE	CONNIE	11230.00
83312	BAKER	ROSALBA	39260.00
83314	DAVID	CHAROLETTE	15150.00
83318	PECK	DARCIE	22330.00
83321	FARMER	ANGELINA	18250.00
83332	LONG	WILLARD	23380.00
83341	CORTEZ	CHRISTINE	14510.00
83347	WINN	QUINTIN	17010.00
83349	SINGH	JENNIFFER	21220.00
83359	WATTS	MERLE	25370.00
83366	BLEDSOE	PHOEBE	23200.00

3. **Write a query to display the invoice number, line numbers, product SKUs, product descriptions, and brand ID for sales of sealer and top coat products of the same brand on the same invoice.**

QUERY:

```
SELECT l.inv_num, l.line_num, p.prod_sku, p.prod_descript, l2.line_num,
p2.prod_sku, p2.prod_descript, p.brand_id
FROM (line l JOIN Product p ON l.prod_sku = p.prod_sku)
JOIN (line l2 JOIN Product p2 ON l2.prod_sku = p2.prod_sku)
ON l.inv_num = l2.inv_num
WHERE p.brand_id = p2.brand_id
AND p.prod_category = 'Sealer'
AND p2.prod_category = 'Top Coat'
ORDER BY l.inv_num, l.line_num;
```

OUTPUT:

There is no observable output.

inv_num	line_num	prod_sku	prod_descript	line_num	prod_sku	prod_descript	brand_id

4. **The Binder Prime Company wants to recognize the employee who sold the most of their products during a specified period. Write a query to display the employee number, employee first name, employee last name, e-mail address, and total units sold for the employee who sold the most Binder Prime brand products between November 1, 2015, and December 5, 2015. If there is a tie for most units sold, sort the output by employee last name.**

   QUERY:

```
SELECT emp.emp_num, emp_fname, emp_lname, emp_email, total
FROM Employee emp
JOIN (SELECT employee_id, SUM (line_qty) AS total
FROM Invoice i JOIN line l USING (inv_num)
JOIN Product p USING (prod_sku) JOIN Brand b
USING (brand_id) WHERE brand_name = 'BINDER PRIME'
AND INV_DATE BETWEEN '2011-11-01' AND '2011-12-06'
GROUP BY employee_id) sub
ON emp.emp_num = sub.employee_id
WHERE total = (SELECT Max(total)
FROM (SELECT employee_id, Sum(line_qty) AS total
FROM Invoice i JOIN Line l USING (inv_num)
JOIN Product p USING (prod_sku) JOIN Brand b
USING (brand_id)
WHERE brand_name = 'BINDER PRIME' AND INV_DATE
BETWEEN '2011-11-01' AND '2011-12-06'
GROUP BY employee_id) AS T);
```

   OUTPUT:

   There is no observable output.

	emp_num	emp_fname	emp_lname	emp_email	total

5. **Write a query to display the customer code, first name, and last name of all customers who have had at least one invoice completed by employee 83649 and at least one invoice completed by employee 83677. Sort the output by customer last name and then first name.**

QUERY:

```
SELECT cust_code, cust_fname, cust_lname
FROM Customer JOIN Invoice USING (cust_code)
WHERE employee_id = 83649
AND EXISTS (SELECT cust_code, cust_fname, cust_lname
FROM Customer JOIN Invoice USING (cust_code)
WHERE employee_id = 83677)
ORDER BY cust_lname, cust_fname;
```

OUTPUT:

There is no observable output.

cust_code	cust_fname	cust_lname

6. **LargeCo is planning a new promotion in Alabama (AL) and wants to know about the largest purchases made by customers in that state. Write a query to display the customer code, customer first name, last name, full address, invoice date, and invoice total of the largest purchase made by each customer in Alabama. Be certain to include any customers in Alabama who have never made a purchase (their invoice dates should be NULL and the invoice totals should display as 0).**

QUERY:

```
SELECT c.cust_code, cust_fname, cust_lname, cust_street, cust_city,
cust_state, cust_zip,inv_date, inv_total AS "Largest Invoice"
FROM Customer c JOIN Invoice i
ON c.cust_code = i.cust_code
WHERE cust_state = 'AL'
AND inv_total = (SELECT MAX(inv_total)
FROM Invoice i2 WHERE i2.cust_code = c.cust_code)
UNION SELECT cust_code, cust_fname, cust_lname, cust_street, cust_city,
cust_state, cust_zip, NULL, 0
FROM Customer
WHERE cust_state = 'AL' AND cust_code
NOT IN (SELECT cust_code FROM Invoice)
ORDER BY cust_lname, cust_fname;
```

OUTPUT:

cust_code	cust_fname	cust_lname	cust_street	cust_city	cust_state	cust_zip	inv_date	Largest Invoice
903	ROBIN	ADDISON	323 LORETTA PLACE	Mobile	AL	36693	2013-10-24	128.71
643	NINA	ALLEN	680 RED TALON DRIVE	Robertsdale	AL	36574	2013-06-21	11.99
295	DORTHY	AUSTIN	829 BIG BEND LOOP	Diamond Shamrock	AL	36614	2013-11-01	86.54
393	FOSTER	BERNAL	1299 EAST 3RD AVENUE	Birmingham	AL	35280	2013-09-15	158.95
853	GAYLORD	BOLTON	1069 LUGENE LANE	Montgomery	AL	36131	2013-11-25	372.68
925	ALANA	BOOKER	1874 I STREET	Mccullough	AL	36502	2013-12-12	208.85
1248	LISA	BRADY	491 LOWLAND AVENUE	Daphne	AL	36577	2014-01-08	371.62
538	CHIQUITA	CALDWELL	1501 BRIGGS COURT	Normal	AL	35762	2013-05-26	143.90
89	MONICA	CANTRELL	697 ADAK CIRCLE	Loachapoka	AL	36865	2014-01-12	314.25
1233	NATHALIE	CHURCH	1802 SNOWY OWL CIRCLE	Napier Field	AL	36303	2013-11-24	160.96

7. **One of the purchasing managers is interested in the impact of product prices on the sale of products of each brand. Write a query to display the brand name, brand type, average price of products of each brand, and total units sold of products of each brand. Even if a product has been sold more than once, its price should only be included once in the calculation of the average price. However, you must be careful because multiple products of the same brand can have the same price, and each of those products must be included in the calculation of the brand's average price.**

QUERY:

```
SELECT brand_name, brand_type, ROUND(avgprice,2) AS "Average
Price", Units_Sold AS "Units Sold"
FROM Brand b
JOIN (SELECT brand_id, AVG(prod_price) AS avgprice
FROM Product GROUP BY brand_id) sub1
ON b.brand_id = sub1.brand_id
JOIN (SELECT brand_id, SUM(line_qty) AS "Units_Sold"
FROM Product p JOIN Line l ON p.prod_sku = l.prod_sku
GROUP BY brand_id) sub2
ON b.brand_id = sub2.brand_id ORDER BY brand_name;
```

OUTPUT:

brand_name	brand_type	Average Price	Units Sold
BINDER PRIME	PREMIUM	16.12	413
BUSTERS	VALUE	22.59	479
FORESTERS BEST	VALUE	20.94	221
HOME COMFORT	CONTRACTOR	21.80	466
LE MODE	PREMIUM	19.22	561
LONG HAUL	CONTRACTOR	20.12	665
OLDE TYME QUALITY	CONTRACTOR	18.33	430
STUTTENFURST	CONTRACTOR	16.47	401
VALU-MATTE	VALUE	16.84	312

8. **The purchasing manager is still concerned about the impact of price on sales. Write a query to display the brand name, brand type, product SKU, product description, and price of any products that are not a premium brand, but that cost more than the most expensive premium brand products.**

QUERY:

```
SELECT brand_name, brand_type, prod_sku, prod_descript, prod_price
FROM Product p JOIN Brand b USING(brand_id)
WHERE  brand_type <> "PREMIUM"
AND prod_price > (SELECT MAX(prod_price)
FROM Product WHERE brand_type = "PREMIUM");
```

OUTPUT:

brand_name	brand_type	prod_sku	prod_descript	prod_price
LONG HAUL	CONTRACTOR	1964-OUT	Fire Resistant Top Coat, for Interior Wood	78.49

9. **Using SQL descriptive statistics functions calculate the value of the following items:**
   a) **What are the products that have a price greater than $50?**
   b) **What is total value of our entire inventory on hand?**
   c) **How many customers do we presently have and what is the total of all customer balances?**
   d) **What are to top three states that buy the most product in dollars from the company?**

QUERY:

a) SELECT prod_sku, prod_descript FROM Product WHERE prod_price > 50;
b) SELECT SUM (prod_qoh*prod_price) as "total value" FROM Product;
c) SELECT COUNT(*) AS "number of customers", SUM(cust_balance) AS "total balance" FROM Customer;
d) SELECT cust.cust_state, SUM(invo.inv_total) AS inv_tot FROM Customer cust INNER JOIN Invoice invo ON cust.cust_code=invo.cust_code WHERE invo.inv_num NOT LIKE '-%' GROUP BY cust.cust_state ORDER BY inv_tot DESC LIMIT 3;

OUTPUT:

a)

PROD_SKU	PROD_DESCRIPT
1021-MTI	Elastomeric, Exterior, Industrial Grade, Water B...
1964-OUT	Fire Resistant Top Coat, for Interior Wood
3694-XFJ	Epoxy-Modified Latex, Interior, Semi-Gloss (MPI...

b)

	total value
▶	360307.79

c)

number of customers	total balance
1362	787201.15

d)

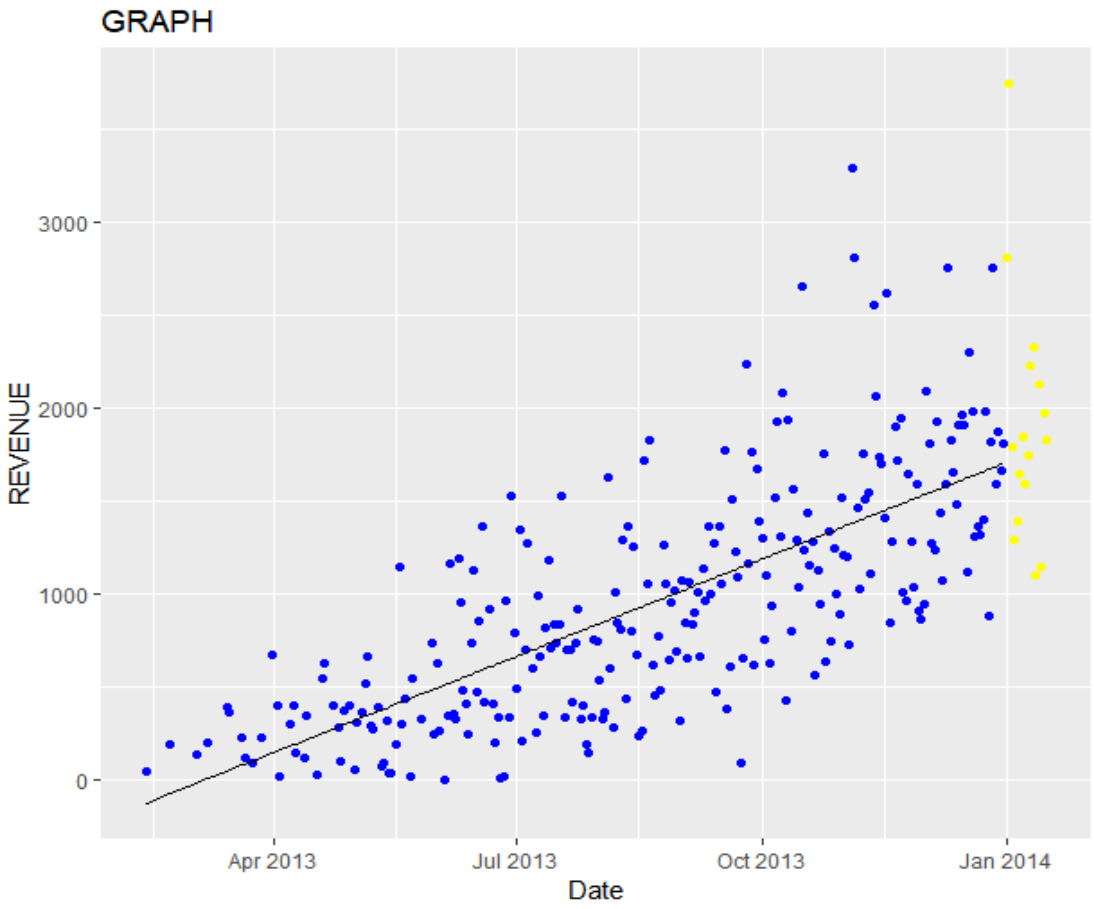
CUST_STATE	inv_tot
PA	38618.12
NY	32242.93
NC	19611.39

10. Using predictive statistics calculate what the predicted forecast of sales for the next year based on the INV_DATE (independent) and INV_TOTAL (dependent). Analyze your results from the linear regression, and provide the R2, model, coefficients, and the confidence interval for your analysis.

R CODE:

```
library("xlsx")
db <-read.xlsx("invoice.xlsx",1)

db=db[,c(2,4)]
db$INV_DATE=as.Date(db$INV_DATE,"%Y/%m/%d")
k <- aggregate(db$INV_TOTAL, by=list(Category=db$INV_DATE), FUN=sum)
names(k) <- c("Datemonthyear","total")

library(caTools)
training_set = k[1:259,]
test_set = k[260:275,]

vik <- lm(total~.,data=training_set)
y_pred = predict(vik, newdata = test_set)

ggplot()+geom_point(aes(x=training_set$Datemonthyear,y=training_set$total),
color='violet') +
geom_point(aes(x=test_set$Datemonthyear,y=test_set$total),color='blue') +
geom_line(aes(x=training_set$Datemonthyear ,
y=predict(reg,newdata=training_set)),color='yellow') +
ggtitle('GRAPH')  + labs(x="Date") +
labs(y="REVENUE")

summary(vik)
confint.lm(vik)
```

OUTPUT:




```
Call:
lm(formula = total ~ ., data = training_set)

Residuals:
     Min       1Q   Median       3Q      Max
-1047.82  -306.84   -39.37   248.12  1911.34

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   -89398.860   5371.606  -16.64   <2e-16 ***
Datemonthyear      5.669      0.337   16.82   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 438.3 on 257 degrees of freedom
Multiple R-squared:  0.524,     Adjusted R-squared:  0.5221
F-statistic: 282.9 on 1 and 257 DF,  p-value: < 2.2e-16
```

	2.5 %	97.5 %
(Intercept)	-99976.828087	-78820.890891
Datemonthyear	5.005343	6.332783

INTERPRETATION:

From the R-squared value we can interpret that 52.5% variation of the target variable is explained by the predictor variable. The low R squared value is due to the presence of only one independent variable.
The coefficient of Datemonthyear is 5.005343 which means for every increase in number of days the total sales is increased by $5.005343
The p-value is less than 0.05 informing us that predictor variable is statistically significant.
The confidence interval is between 5.005343 and 6.332783.

EXCEL OUTPUT:

Regression Statistics	
Multiple R	0.036842
R Square	0.001357
Adjusted R	0.000623
Standard E	128.6527
Observatic	1362

	Coefficients	*andard Err*	*t Stat*	*P-value*
Intercept	-2475.2	1969.937	-1.25649	0.209155
INV_DATE	0.064453	0.047406	1.359602	0.174181

INTERPRETATION:

When the uncleaned data is processed in Microsoft Excel an R squared value of 0.0013 is obtained suggesting that it is a very bad model and does not explain any variation. The p-value is higher than 0.05 also suggesting that the predictor variable is statistically not significant.

Conclusion

This project has helped get a much greater and wider understanding on the concepts of database foundations and helped me provide a hands-on experience with real-world data. The experience of normalizing and cleaning data and creating a database from scratch has helped me analyze and perceive the intricacies of data-preprocessing and manipulation. The challenge to derive insights from the processed data to solve the problem set has helped me further my skill-set in querying in MySQL. Taking a step further from the SQL environment, the problem set has tested my skill in R programming as well. The use of linear regression has helped to gain more detailed information of the trends in the dataset. Overall, the project has been a very positive experience is the testing the skills and concepts studied during my coursework.