LARGECO PRODUCT SALES ANALYSIS

BUAN 6320 INDIVIDUAL PROJECT

AUTHOR : SUJITH NAIR

NET ID : SXN180033

DATE : 12TH DECEMBER 2018

Contents

[ABSTRACT 3](#_Toc532326997)

[DATA NORMALIZATION 3](#_Toc532326998)

[DATA IMPORT 4](#_Toc532326999)

[Data Import Issue 5](#_Toc532327000)

[DATA MODEL 6](#_Toc532327001)

[QUESTION SET (QUERYING) 7](#_Toc532327002)

[Question 1: Current Salary 7](#_Toc532327003)

[Question 2: Starting Salary 8](#_Toc532327004)

[Question 3: Same Brand On Same Invoice 9](#_Toc532327005)

[Question 4: High performer 10](#_Toc532327006)

[Recommendation : 10](#_Toc532327007)

[Question 5: Same customer with invoices from 2 specific employees. 11](#_Toc532327008)

[Question 6: Largest purchase in Alabama 12](#_Toc532327009)

[Question 7: Impact of product prices 14](#_Toc532327010)

[Question 8: Expensive non-premium brands 14](#_Toc532327011)

[Question 9: SQL descriptive statistics 15](#_Toc532327012)

[Question 10 : Regression Analysis 16](#_Toc532327013)

[Steps to convert data to perform regression 17](#_Toc532327014)

[Regression in Excel 18](#_Toc532327015)

[Summary of Regression Results 19](#_Toc532327016)

[Forecast using Excel’s built in Forecast Sheet 20](#_Toc532327017)

[Forecast from Regression Results 21](#_Toc532327018)

[APPENDIX 22](#_Toc532327019)

[Source datasets : 22](#_Toc532327020)

[Data Dictionary : 22](#_Toc532327021)

[Normalized data tables : 22](#_Toc532327022)

[DDL Script : 23](#_Toc532327023)

[DML Insert Scripts : 23](#_Toc532327024)

[Invoice table duplicate rows : 23](#_Toc532327025)

[Physical Data Model : 23](#_Toc532327026)

[Question 2 - Full Result Set : 24](#_Toc532327027)

# ABSTRACT

This report presents a data analysis on the product sales for LargeCo during the years 2013 and 2014. Three source datasets describing the customers, products and employees was the source data. The following steps describe the process followed :

1. The three datasets were normalized into 10 tables using Data Dictionary provided.
2. Data was cleaned by handling missing values and bad characters.
3. A logical and physical model was created using DBeaver.
4. Data was implemented in MySQL. Cleaned data was imported into DBeaver.
5. A set of questions were answered using SQL querying.
6. A linear regression in excel was performed and the forecast results were presented.

# DATA NORMALIZATION

Three datasets were provided as source data. They are denormalized datasets that provide information on Customers, Products and Employees of the company. Find source datasets [here](#_Source_datasets_:) .

A data dictionary for the normalized table names and indexes was also provided. Find it [here](#_Data_Dictionary_:).

Following steps were done in the data normalization process :

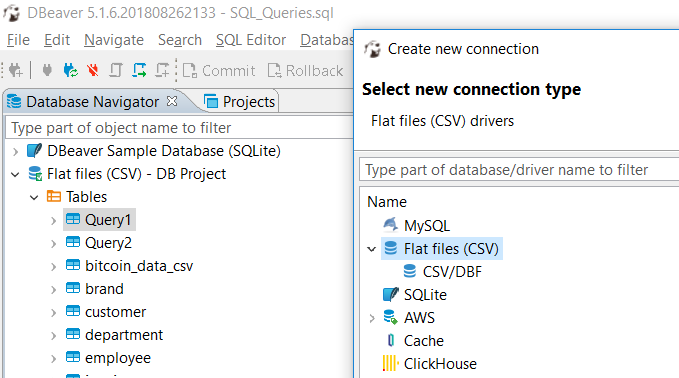
1. Source datasets which was in text format was copied to csv files.
2. Data was normalized into new csv files with names and data as per the data dictionary.
3. Normalized data files in csv can be viewed [here](#_Normalized_data_tables).
4. The following data munging steps were taken :
   1. Bad characters were identified using data validation in Excel – by checking for alphanumeric characters only in all columns. Exception like ‘@’ in EMP\_EMAIL and ‘-‘ in PROD\_SKU,PROD\_DESCRIPT,BRAND\_NAME and EMP\_PHONE were allowed.
   2. Characters like Æ,?, ñ, Ä were removed individually from the column PROD\_QOH.
   3. Additional bad characters in PROD\_SKU were removed by applying Text-to-Column method and fixing its width to 8 characters in Excel.
   4. SAL\_END was updated to empty columns. This was later updated to [NULL] in MySQL.

# 

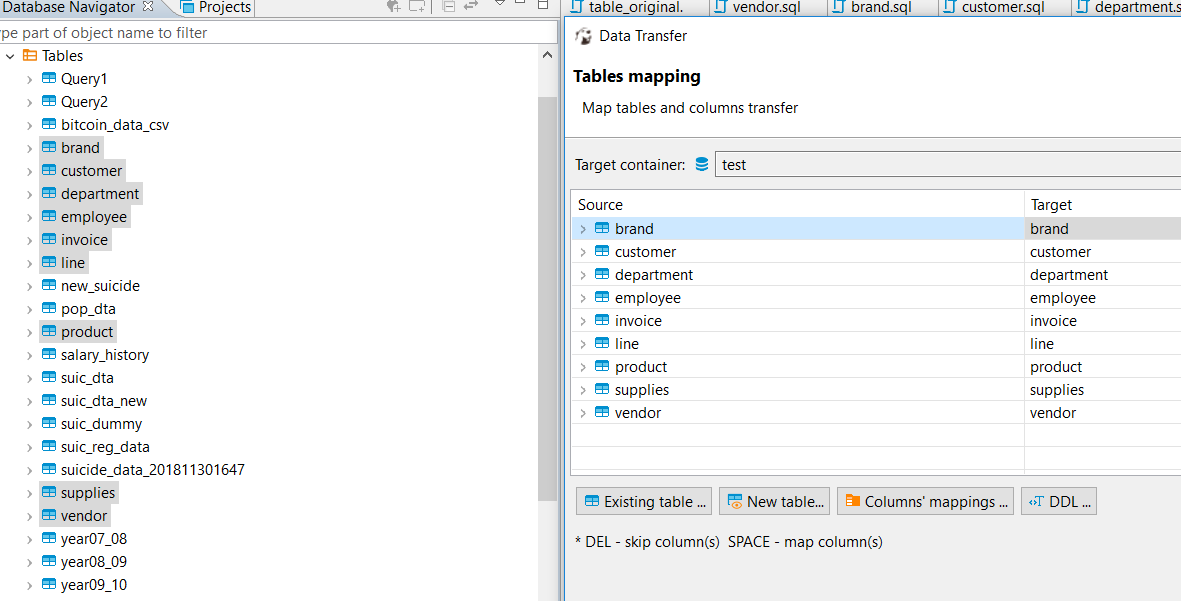
# DATA IMPORT

Dbeaver MySQL used for data import from CSV files to database. Below were the steps :

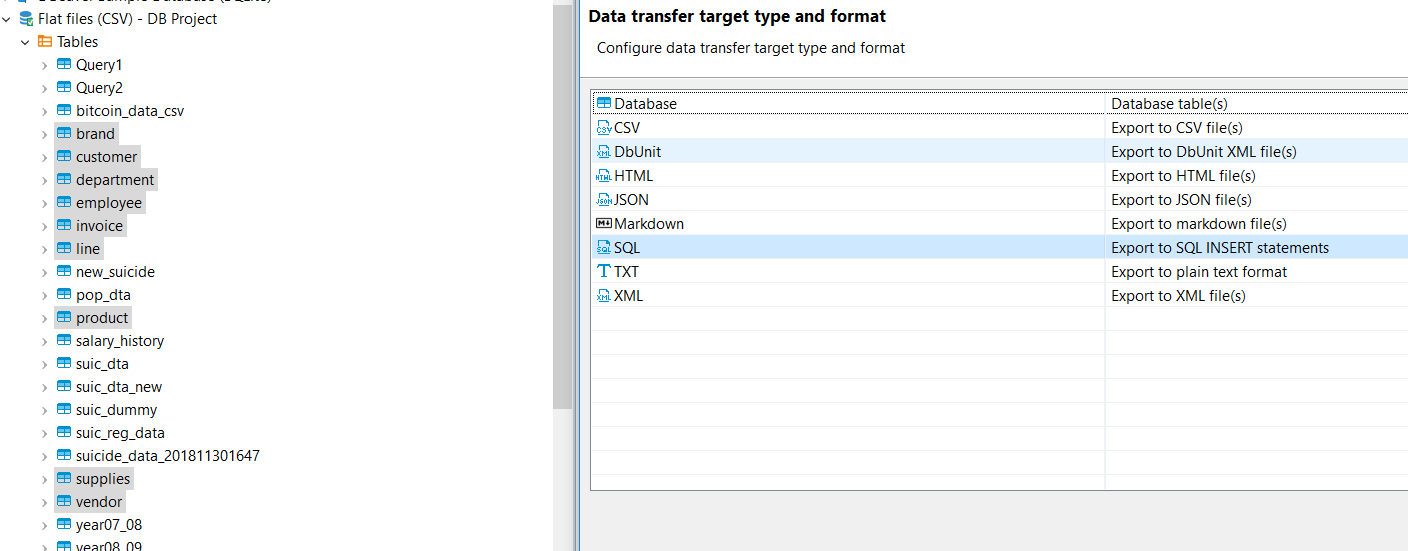
1. A new connection was created in DBeaver to load flat files(csv) and connected to a file location in the drive with all normalized CSV files. All tables appeared in the Dbeaver Flat Files connection.



1. All flat files were exported into the a new MySQL Database target.



1. The Columns mappings in above step was changed to match data dictionary. Obtained table creation DDL in this step.
2. Again, the flat files were selected to be exported, now using Export to SQL INSERT statements option as shown below.



1. The DDL script was modified to add primary key, foreign key and indexes. The DDL script can be found [here](#_DDL_Script_:). The statement were executed and tables were formed in the MySQL database.
2. The INSERT SQL statements from Step 4 were executed in order given in Data Dictionary to ensure correct data loading. DML script to insert records for each table can be found [here](#_DML_Insert_Scripts).
3. All table except INVOICE were loaded without error. Details on the INVOICE table data import issue are provided below.

## Data Import Issue

The invoice table data has the below issue :

Two customers with same invoice number. 15 such cases exist. Below is one of the, as a sample :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CUST\_CODE | INV\_DATE | INV\_TOTAL | EMPLOYEE\_ID | INV\_NUM | LINE\_NUM | PROD\_SKU | LINE\_QTY | LINE\_PRICE |
| 392 | 2013-09-15 | 158.95 | 84106 | 2275 | 2 | 5413-TTF | 2 | 29.99 |
| 393 | 2013-09-15 | 158.95 | 84106 | 2275 | 2 | 5413-TTF | 2 | 29.99 |

Here, CUST\_CODE 392,393 have the same invoice number – 2275. Above column values are also same. Hence, this seems like an entry error. Since INV\_NUM is the primary key in the INVOICE table, the 15 records that appeared later in the dataset were not inserted.

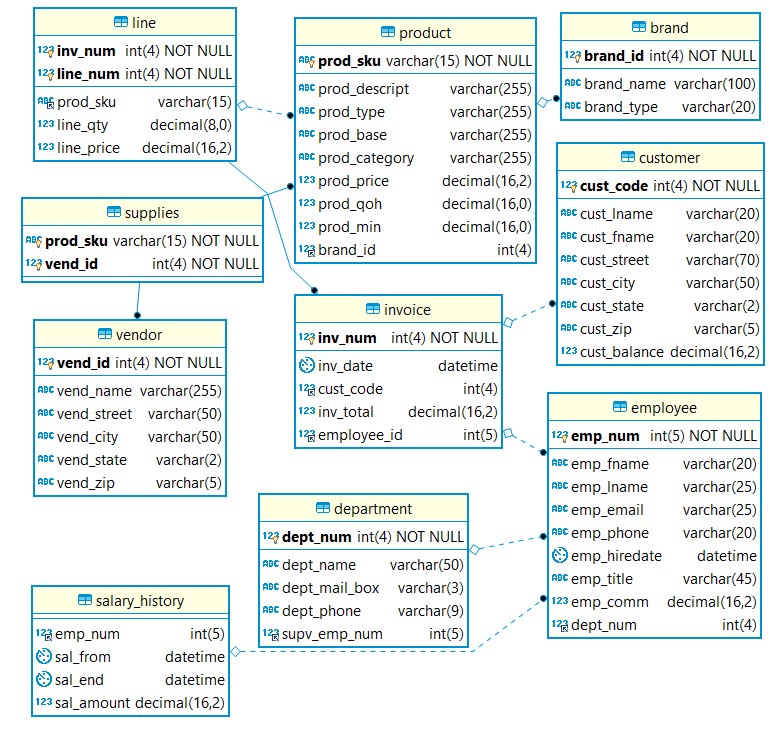
The following are list of all such records :



Find the csv sheet with list of duplicate invoice for same customer records in the appendix or click [here](#_Invoice_table_duplicate).

# DATA MODEL

Once the data tables were created and constraints applied, the below Entity Relationship Diagram was created in Dbeaver.



A .png and .pdf version of the above data model can be found [here](#_Physical_Data_Model).

# QUESTION SET (QUERYING)

Question 1: Current Salary

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.

**select** e.emp\_num,e.emp\_fname,e.emp\_lname,e.dept\_num,s.sal\_amount **as** current\_salary

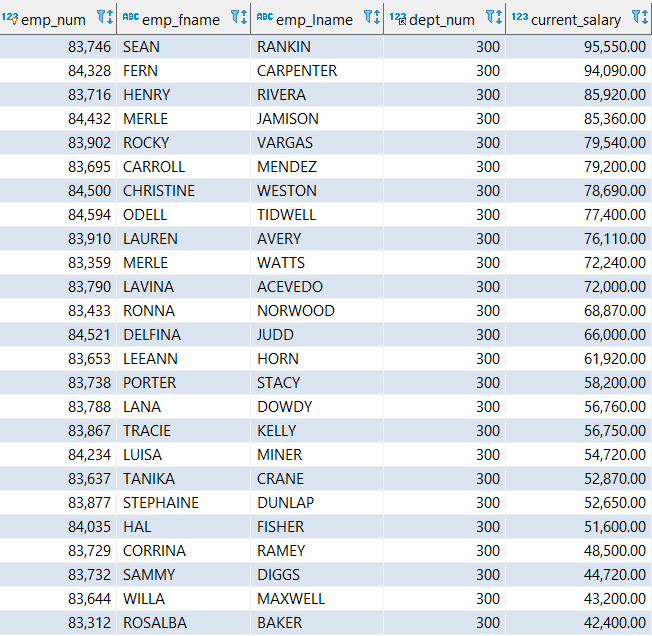
**from** employee e,salary\_history s

**where** e.dept\_num = '300'

**and** e.emp\_num = s.emp\_num

**and** s.sal\_end **is** **null**

**order** **by** s.sal\_amount **desc**;



## Question 2: Starting Salary

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.

**select** e.emp\_num,e.emp\_fname,e.emp\_lname,s.sal\_amount **as** starting\_salary

**from** employee e,salary\_history s

**where** e.emp\_num = s.emp\_num

**and** (e.emp\_num,s.sal\_from) **in**

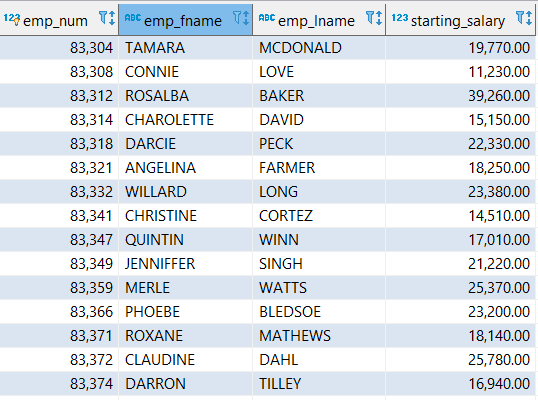
(**select** sh.emp\_num,**min**(sh.sal\_from)

**from** salary\_history sh

**group** **by** sh.emp\_num

);

The result set has 363 records. Below are the top 15 records. Find the full result [here](#_QUESTION_2_FULL).



## Question 3: Same Brand On Same Invoice

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.

**select** sealer.inv\_num,sealer.line\_num,sealer.prod\_sku,sealer.prod\_descript,sealer.brand\_id

**from**

(**select** l1.inv\_num,l1.line\_num,l1.prod\_sku,p1.prod\_descript,p1.brand\_id

**from** line l1,product p1

**where** l1.prod\_sku=p1.prod\_sku

**and** p1.prod\_category = 'Sealer'

) sealer,

(**select** l2.inv\_num,l2.line\_num,l2.prod\_sku,p2.prod\_descript,p2.brand\_id

**from** line l2,product p2

**where** l2.prod\_sku=p2.prod\_sku

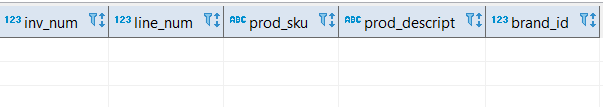
**and** p2.prod\_category = 'Top Coat'

) top\_coat

**where** sealer.inv\_num = top\_coat.inv\_num

**and** sealer.brand\_id = top\_coat.brand\_id

;



No records found.

## Question 4: High performer

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.

**select** e1.emp\_num,e1.emp\_fname,e1.emp\_lname,e1.emp\_email,**sum**(l1.line\_qty) **as** Quantity\_Sold

**from** employee e1,invoice i1,line l1,product p1,brand b1

**where** e1.emp\_num = i1.employee\_id

**and** i1.inv\_num = l1.inv\_num

**and** p1.prod\_sku = l1.prod\_sku

**and** p1.brand\_id = b1.brand\_id

**and** b1.brand\_name = 'BINDER PRIME'

**and** i1.inv\_date **between** '2015-11-01' **and** '2015-12-05'

**group** **by** e1.emp\_num,e1.emp\_fname,e1.emp\_lname,e1.emp\_email

**having** **sum**(l1.line\_qty) =

(

**select** **max**(a.Quantity\_Sold) **from**

(**select** e.emp\_num,**sum**(l.line\_qty) **as** Quantity\_Sold

**from** employee e,invoice i,line l,product p,brand b

**where** e.emp\_num = i.employee\_id

**and** i.inv\_num = l.inv\_num

**and** p.prod\_sku = l.prod\_sku

**and** p.brand\_id = b.brand\_id

**and** b.brand\_name = 'BINDER PRIME'

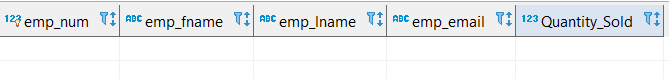
**and** i.inv\_date **between** '2015-11-01' **and** '2015-12-05'

**group** **by** e.emp\_num

) a

)

**order** **by** e1.emp\_lname;



No records found.

### Recommendation :

The above query does not give any output as invoice data ends on 16th January 2014. A more relevant query would be to find the best performing employee in the entire period for BINDER PRODUCTS. Find the query and result for the same below :

**select** e1.emp\_num,e1.emp\_fname,e1.emp\_lname,e1.emp\_email,**sum**(l1.line\_qty) **as** Quantity\_Sold

**from** employee e1,invoice i1,line l1,product p1,brand b1

**where** e1.emp\_num = i1.employee\_id

**and** i1.inv\_num = l1.inv\_num

**and** p1.prod\_sku = l1.prod\_sku

**and** p1.brand\_id = b1.brand\_id

**and** b1.brand\_name = 'BINDER PRIME'

**group** **by** e1.emp\_num,e1.emp\_fname,e1.emp\_lname,e1.emp\_email

**having** **sum**(l1.line\_qty) =

(

**select** **max**(a.Quantity\_Sold) **from**

(**select** e.emp\_num,**sum**(l.line\_qty) **as** Quantity\_Sold

**from** employee e,invoice i,line l,product p,brand b

**where** e.emp\_num = i.employee\_id

**and** i.inv\_num = l.inv\_num

**and** p.prod\_sku = l.prod\_sku

**and** p.brand\_id = b.brand\_id

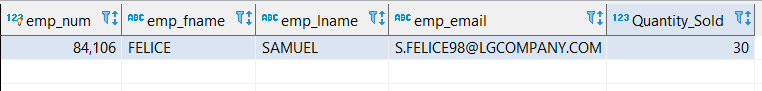
**and** b.brand\_name = 'BINDER PRIME'

**group** **by** e.emp\_num

) a

)

**order** **by** e1.emp\_lname;



Samuel Felice is the best performing employee for BINDER PRIME products for the entire period.

## Question 5: Same customer with invoices from 2 specific employees.

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.

**select** c1.cust\_code,c1.cust\_fname,c1.cust\_lname **from** customer c1,invoice i1

**where** i1.employee\_id=83649

**and** c1.cust\_code=i1.cust\_code

**and** **exists**

(**select** (1)

**from** customer c2,invoice i2

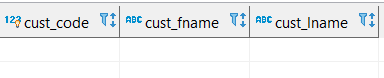
**where** i2.employee\_id=83677

**and** c2.cust\_code=i2.cust\_code

**and** c1.cust\_code=c2.cust\_code

)

**order** **by** c1.cust\_lname,c1.cust\_fname;



No records found

## Question 6: Largest purchase in Alabama

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).

**select** c.cust\_code,c.cust\_fname,c.cust\_lname,**concat**(c.cust\_street,' ',c.cust\_state,' ',cust\_zip) address,

i.inv\_date,IFNULL(i.inv\_total,0) largest\_purchase

**from** customer c **left** **outer** **join** invoice i

**on** c.cust\_code = i.cust\_code

**where** (c.cust\_code,i.inv\_total) **in**

(

**select** c1.cust\_code,**max**(i1.inv\_total)

**from** customer c1 **left** **outer** **join** invoice i1

**on** c1.cust\_code = i1.cust\_code

**where** c1.cust\_state = 'AL'

**group** **by** c1.cust\_code

)

**or** (i.inv\_total **is** **null** **and** c.cust\_state = 'AL');

## Question 7: Impact of product prices

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.

**select** qty.brand\_name,qty.brand\_type,price.avg\_price,qty.total\_units\_sold

**from**

(

**select** b.brand\_id,b.brand\_name,b.brand\_type,**sum**(l.line\_qty) total\_units\_sold **from** brand b,product p,line l

**where** b.brand\_id=p.brand\_id

**and** p.prod\_sku=l.prod\_sku

**group** **by** b.brand\_id

) qty,

(

**select** b.brand\_id,**avg**(p.prod\_price) avg\_price

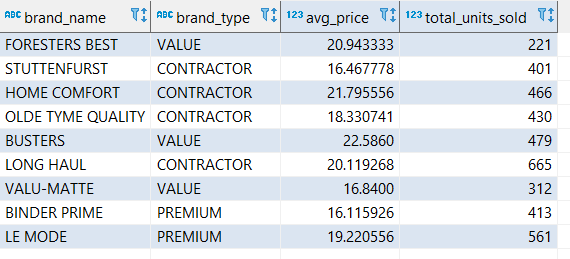
**from** brand b,product p

**where** b.brand\_id=p.brand\_id

**group** **by** b.brand\_id

) price

**where** qty.brand\_id = price.brand\_id;



## Question 8: Expensive non-premium brands

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.

**select** b.brand\_id,b.brand\_name,b.brand\_type,p.prod\_sku,p.prod\_descript,p.prod\_price

**from** brand b,product p

**where** b.brand\_id=p.brand\_id

**and** b.brand\_type <> 'PREMIUM'

**and** p.prod\_price > **any**

(

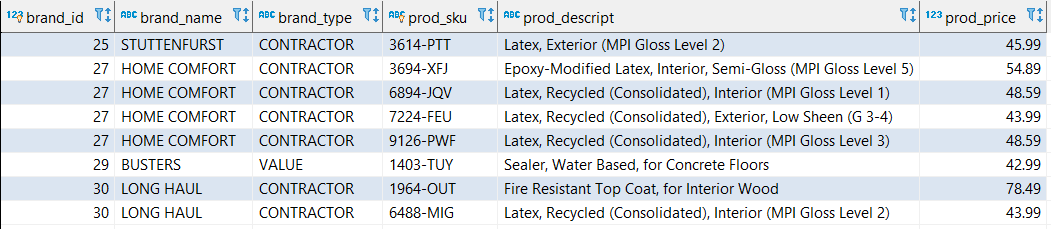
**select** **max**(p1.prod\_price)

**from** product p1,brand b1

**where** p1.brand\_id = b1.brand\_id

**and** b1.brand\_type = 'PREMIUM'

**group** **by** b1.brand\_id );



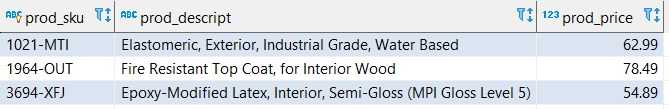
## Question 9: SQL descriptive statistics

Using SQL descriptive statistics functions calculate the value of the following items:

1. What are the products that have a price greater than $50?

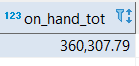
**select** p.prod\_sku,p.prod\_descript,p.prod\_price

**from** product p **where** p.prod\_price > 50;



1. What is total value of our entire inventory on hand?

**select** **sum**(p.prod\_price\*p.prod\_qoh) on\_hand\_tot **from** product p;



1. How many customers do we presently have and what is the total of all customer balances?

**select** **count**(c.cust\_code) cust\_tot,**sum**(c.cust\_balance) cust\_tot\_bal **from** customer c ;



d. What are to top three states that buy the most product in dollars from the company?

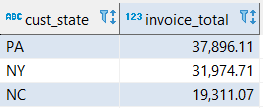
**select** c.cust\_state,**sum**(i.inv\_total) **from** invoice i,customer c

**where** c.cust\_code = i.cust\_code

**group** **by** c.cust\_state

**order** **by** 2 **desc**

**limit** 3;



# Question 10 : Regression Analysis

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.

To make a regression analysis on the invoice data, the following 2 data conversion steps were taken :

1. Group and aggregate per day sales.

* This is necessary because the data contains multiple sales on a single day. A regression performed on such data will not yield meaningful result regarding the growth of sales across time period.
* Aggregated data provides a better dataset to analyze the growth of the company over time and look at the causal effect of inv\_date on inv\_total.

1. Convert date to number of days after Jan 1 1900.

* To perform a regression analysis with date as the dependent variable, the date has to be in numeric form. There are two ways to achieve this :
  + Convert data to numbers, with the oldest date converted to 1 followed by numbers increasing in sequence.
  + Convert date to its numeric form as stored in excel, i.e. the number of days after Jan 1 1900.
* The latter format was chosen, because on many days there was no sales and this format will help later in forecasting, as date format can be reconverted from these numeric values.

## Steps to convert data to perform regression

1. Write the query to group data on date ; aggregate and order by invoice total.

**select** inv\_date,**sum**(inv\_total) inv\_total **from** invoice

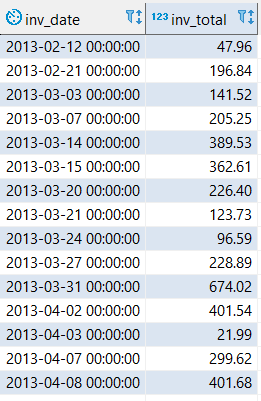
**group** **by** inv\_date

**order** **by** 1;

It gives 275 records for 275 days given in below sheet.



Below is a snapshot of top 15 records :



1. Export the output result set to excel for regression analysis.
2. Convert inv\_date to Date format in excel. This enables excel to perform regression on date values using its Julian numeric value(Number of days after 1 Jan 1900).
3. Add index column. The final regression data and a snapshot is given below :





## Regression in Excel

There are a number of advanced software products like R and Stata that can perform regression. However, since here the requirement is a linear regression model with 275 data points, excel was chosen as it provides all the necessary tools to perform the required modelling.

Excel’s Data Analysis Add-In provides the option for regression. Find below the complete regression result and forecasting result in excel.



### Summary of Regression Results



**Significance :** Anova F value is 325.5589. The overall regression result is highly significant.

**Goodness of Fit :** The model is a good fit to predict sales. R Square is 54.39%.

**Model :** inv\_total = β0  + β1.inv\_date + u

inv\_total = -238862.287 + 5.778\* inv\_date + u

**Coefficients :**

**Intercept :** -238862.287

It is significant with a t stat of -17.9681. The large negative value is expected due to the Julian date format(number of days after Jan 1 1900). It effectively means that it is the sales value on 1 Jan 1900 which is irrelevant and of no use for our purposes. Hence, the intercept can be ignored.

Its confidence interval is (-265033 , -212691)

**inv\_date :** 5.778

It is highly significant and positive with a t stat value of 18.04 >> 2. Hence, increasing dates have a positive correlation on the sales.

Its confidence interval is (5.147,6.408)

**Interpretation :** Each day the sales value is expected to increase by 5.78$ on average.

**Regression Plot :** Below is the regression plot for the given data :

### Forecast using Excel’s built in Forecast Sheet

To calculate the forecast, excel provides a built in Forecast Sheet that predicts the future sales based on the past year data and seasonality. It provides the below result with the grouped regression data.

The data for the same is provided in this sheet.



### Forecast from Regression Results

Using the regression model, we predict next year sales.

inv\_total = -238862.287 + 5.778\* inv\_date + u

The forecast for the dates 17th Jan 2014 to 18th October 2014 is provided in the sheet below :



Below is a snapshot :



# APPENDIX

## Source datasets :







[Click here to go back to DATA NORMALIZATION](#_DATA_NORMALIZATION)

## Data Dictionary :



[Click here to go back to DATA NORMALIZATION](#_DATA_NORMALIZATION)



## Normalized data tables :



[Click here to go back to DATA NORMALIZATION](#_DATA_NORMALIZATION)

## 

## DDL Script :



[Click here to go back to Data Import](#_DATA_IMPORT_:)

## DML Insert Scripts :



[Click here to go back to Data Import Issue](#_DATA_IMPORT_ISSUE)

## Invoice table duplicate rows :



[Click here to go back to Data Import Issue](#_DATA_IMPORT_ISSUE)

## Physical Data Model :

[Click here to go back to QUESTION SET.](#_QUESTION_SET_(QUERYING))

## Question 2 - Full Result Set :



[Click here to go back to Question 3](#_Question_3:_Same)