

CS 7720-01

DATA MINING

FALL 2015

PROFESSOR: ERIC SAUNDERS

PROJECT: PROMOTIONS FOR BUSINESS DEVELOPMENT

Name: RAJASHEKAR SUGANDAM UID: U00750675

“I have neither given nor received aid on this project, nor have I observed any violation of the Honor code”

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: 12/17/2015 Report due date: 12/17/2015

**TABLE OF CONTENTS**

**DESCRPTION OF PROJECT………………………………………………………..3**

**DATABASE SCHEMA ……………………………………………………………….3**

**DATA REQUIREMENTS…………………………………………………………….3**

**DESIGN CONSIDERATION…………………………………………………………3**

**TABLE STRUCTURES……………………………………………………………….4**

**CONTENTS OF DATA TABLES…………………………………………………….5**

**TRANSACTION RESULTS…………………………………………………………..7**

**OVERVIEW OF MINING PROCESS……………………………………………….12**

**PROJECT REQUIREMENTS………………………………………………………..13**

**APPENDIX……………………………………………………………………………..15**

**REGERENCES………………………………………………………………………...15**

**BACKGROUND FOR STUDY:**

**Description of Project:**

Present days promotions are an important factor for a business development. So the process of providing promotions to the customers should be in such a way that promotions should make the customers visit store regularly. The main objective of this project is “Promotions for Business Development” where a particular business analyst keeps track of all its business transactions i.e. (Historic Data), so that when he wants to improve his business profits he will analyze his history data thereby mining an interesting knowledge which helps in business improvement.

In this project I made a system where a new customer will be registered into the system so all transactions which he/she is done will be stored into the database along with the time, date, and items purchased. So in this project I want to mine frequent itemsets from the transaction history data of all customers. So that using these frequent itemsets the business analyst can provide some deals on these items which makes the customers come back. In order to find these frequent itemsets I used association rule mining (Apriori Mining) in this project.

**Database Schema:**

**Problem Statement:** In this database design when a new customer is registered he will be stored in a customer table with all his personal information. So using his login credentials customer will be logged in, which directs to his home page where his personal information will be displayed. On home page there will be a button called shop when clicked on that it takes to an items page where all the items in store will be displayed you can select the items you need then upon clicking checkout the whole transaction will be stored in transaction table along with time. So this table will be used for mining association between the items.

**Data Requirements:**

The software should capture the following information:

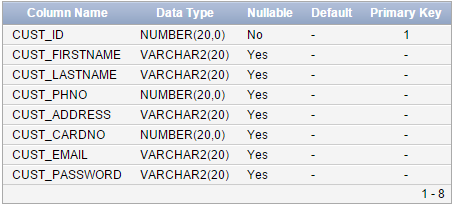
1. Each customer who is newly registered should be stored in customers table with all the information entered in registration form.
2. Each transaction done by each customer should be stored in transaction table along with Customer Id, Transaction Id, Time, and Total sale.
3. Upon storing each transaction in transaction table this each transaction need to be updated in Trans\_spec table which is represented by Transaction\_ID, Item\_ID.
4. So the resultant Trans\_spec table will be the input for mining algorithm which finally generates the frequent itemsets.

**Design Consideration:**

After looking at the project requirements, I decided Relational Database Management System (RDBMS) would be the best solution for my implementation. ORACLE 11g APEX has been chosen to develop the backend database system as it is cost effective and is most widely used RDBMS. And my front end implementation is done using Java Programming Language (JSP, SERVLETS). I have chosen some transaction to be executed using front end.

**Tables Structure:**

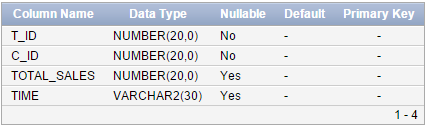
**Customers:**

****

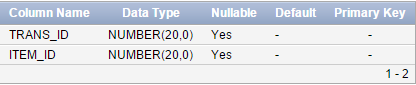
**Items:**

****

**Transactions:**

****

**Trans\_spec:**

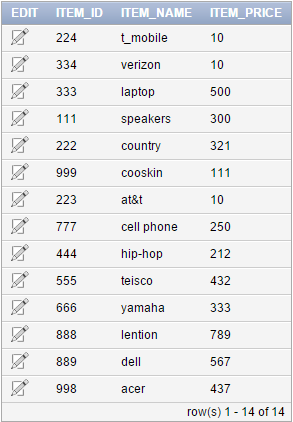
****

**Contents of Data Tables:**

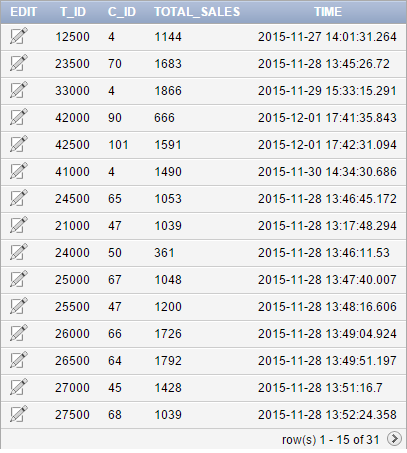
**Customers:** (Only few records have been displayed)

****

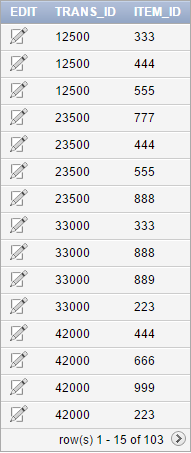
**Items:** (Only few records have been displayed)

****

**Transactions:** (Only few records have been displayed)

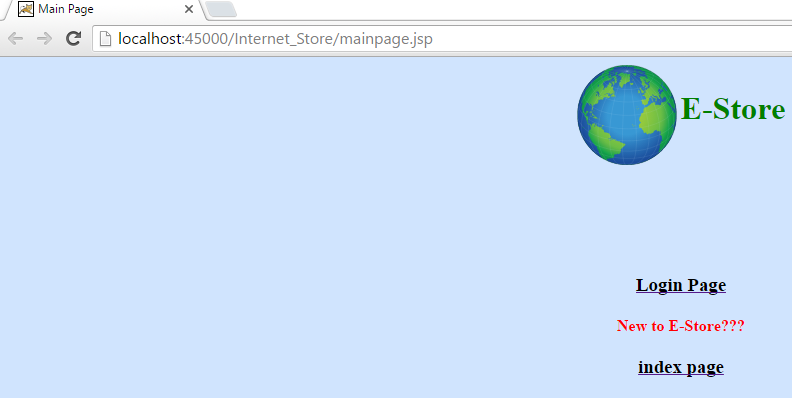
****

**Trans\_spec:** (Only few records have been displayed)

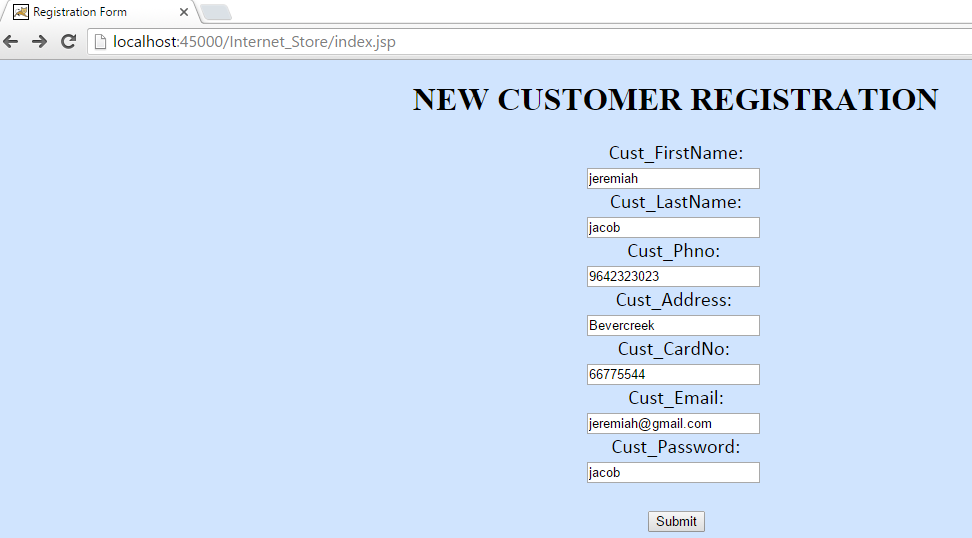
****

**Transaction Results:**

When we run the complete project we will be getting this following main page,



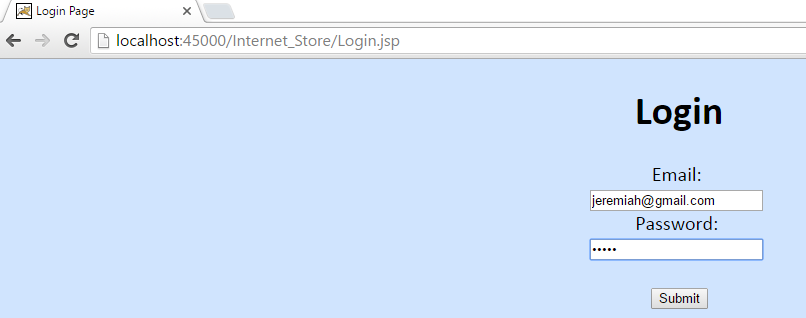
So, when a customer is new to the store he/she need to be registered. For new registration click index page it show as follows,

****

Upon filling all the fields then click submit so that a new customer will be registered. Let’s check the new customer in customers table,



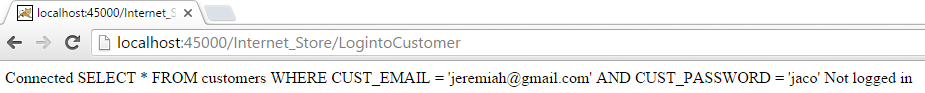
Now the newly registered he can login using his/her email, and respective password, let’s look at how it works,



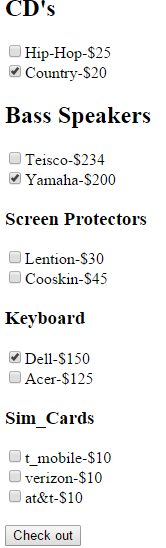
If the user credentials are correct then it take to the home page of customer which is as follows,



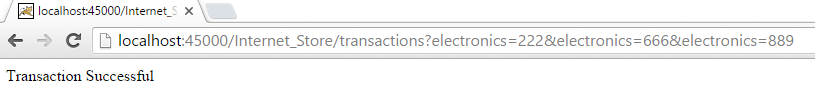
If the user entered credentials are invalid then it displays as follows,



Now if the customer want to shop some items in the store then click shop button. It redirects to the items page which is like this, (Only few items have been displayed)

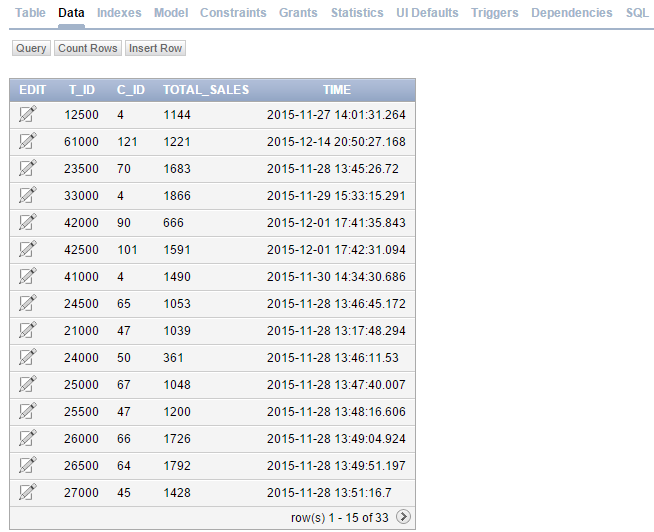


Now he/she selected some items and when clicked checkout following message will be displayed.



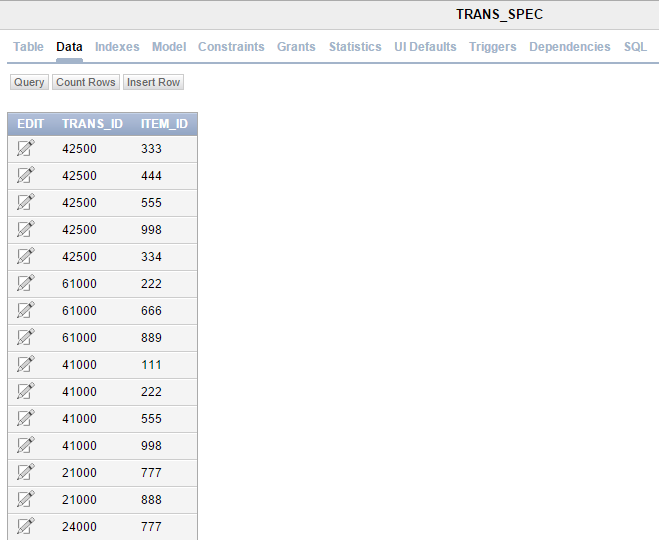
These completed transactions are saved in Transaction table, thereby updating the Trans\_spec table with their respective Transaction\_ID and Item\_ID.

**Transaction Table:**

****

In our example the customer with ID=121 has been logged in. That customer has done a transaction with T\_ID=61000 with the total\_sale=1221 at that particular time which we can see in the above table at 2nd row.

**Trans\_spec Table:**



From the above table we can see the items purchased under Trans\_ID=121 i.e. Item\_ID=222,666,889.

According to above process the whole business will be running such as new customer registering, logging in into his/her account, purchasing items, and storing the transactions performed by every customer along with the Transaction\_ID and Item\_ID.

So now the resultant Trans\_spec table will be the input for finding associations between the items which will give the output of frequent itemsets for a given min\_support.

**FINDINGS:**

**Overview of Mining Process:**

The data which I have collected for this project is a test data (Poor Quality) so the data is static and do not consider about missing or redundant data. Using this static data we are mining frequent itemsets using association rules (Apriori) of all transactions in the relational database. This frequent itemsets will be helpful for the business owner to improve his business by providing deals on these frequent items. As this data is stored in relational database so it can be imported in any database.

**Mining Results:**

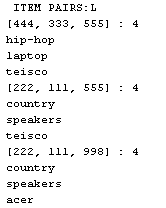
Using the table (Trans\_spec) as an input we implement the Apriori algorithm (apriori.java).

* Initially the whole Trans\_spec table which has (T\_ID, ITEM\_ID) is stored in a hash set.
* From that both Transaction number and Items in transactions are stored in an array list

d [][] and we request user for minimum support (as an integer value):.

* Using that list count of each item in all transactions are stored (count(s)), where s is the hash set.
* Using the min\_support we check for items which satisfies the support count (t.support >= min\_support) there by pruning prune () the rest. This we say it as candidate set.
* We generate frequent items sets using this method obtaining 1-candiatate\_set, 2-candiate\_set .., until we reach frequent itemsets that last satisfy the min\_support.
* Finally we obtain Item\_ID: t.itemset i.e. frequent itemsets along with item\_names.

Now for given minimum support=4 the final result of frequent itemsets are listed below. Due to space problem I’m not displaying all the transactions, candidate\_sets. But the text file (Freqlist.txt) holds the complete information.

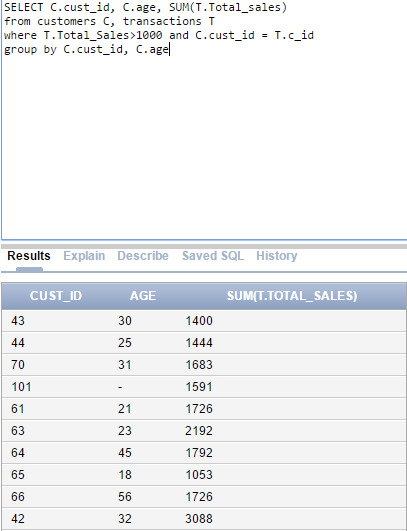


As there is no enough space to display the whole output in the report I have saved the output to a text file named (Freqlist.txt) which is provided in the documentation.

**APPENDIX:**

**Project Requirements:**

**Data Characterization:**



Data characterization is summarization of characteristics of a specified class. In this project summarization is done using OLAP operation on specific dimensions. Above is an example of OLAP query where we extracted the customer whose total sales are above 1000 based on their age.

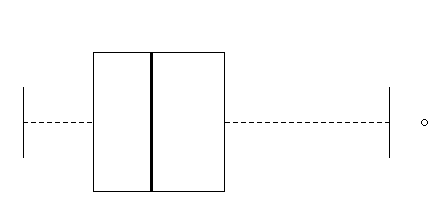
**Statistics:**



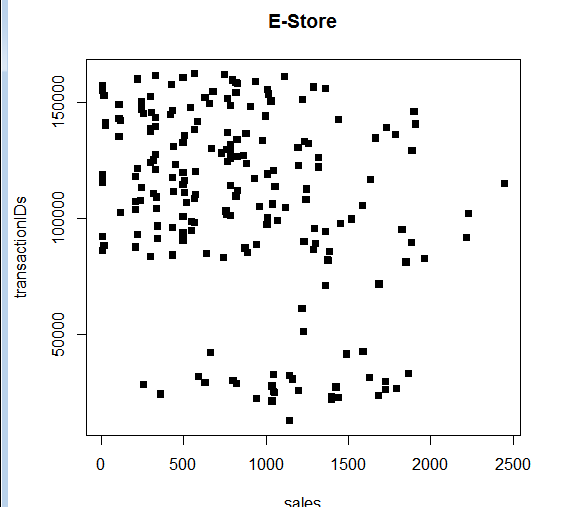
**Outliers:** Due to space issue only few values been displayed.

****

**Outlier image:**

****

**Boxplot:**

****

**Types of attributes:**

* Nominal attributes- Customer\_ID, Item\_ID, Transaction\_ID.
* Numeric attributes- Item\_price, Total\_sale.
* Interval attribute- Time.

**Hypothesis Questions:**

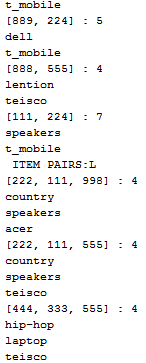
* When finding the frequent sets reducing the obtaining output set by allowing the possibility to express constraints on target results.
* Another drawback is to reduce the number of times scanning the database thereby generating efficient sets.

**Additional:**

* Total time taken to run the project i.e. total time: 25 seconds.
* Total time taken to run the algorithm i.e. total time: 10 seconds.

Apriori algorithm is something for mining frequent itemsets for Boolean association rules. The set of items which satisfy the minimum support are frequent itemsets. The subsets of frequent itemsets should also satisfy min\_support. So, the same procedure has been implemented in my project i.e. developing a transaction table (Trans\_spec) which consists of only Transaction\_ID, Item\_ID.

**Output:**



**REFERENCES:**

1. Han, Jiawei, Micheline Kamber, and Jian Pei. *Data mining: concepts and techniques: concepts and techniques*. Elsevier, 2011.
2. http://www.pracspedia.com/DWM/apriori.html.