Apriori Algorithm Web Application

Gabrielle Mack

CS 634

11 MAR 2017

Contents

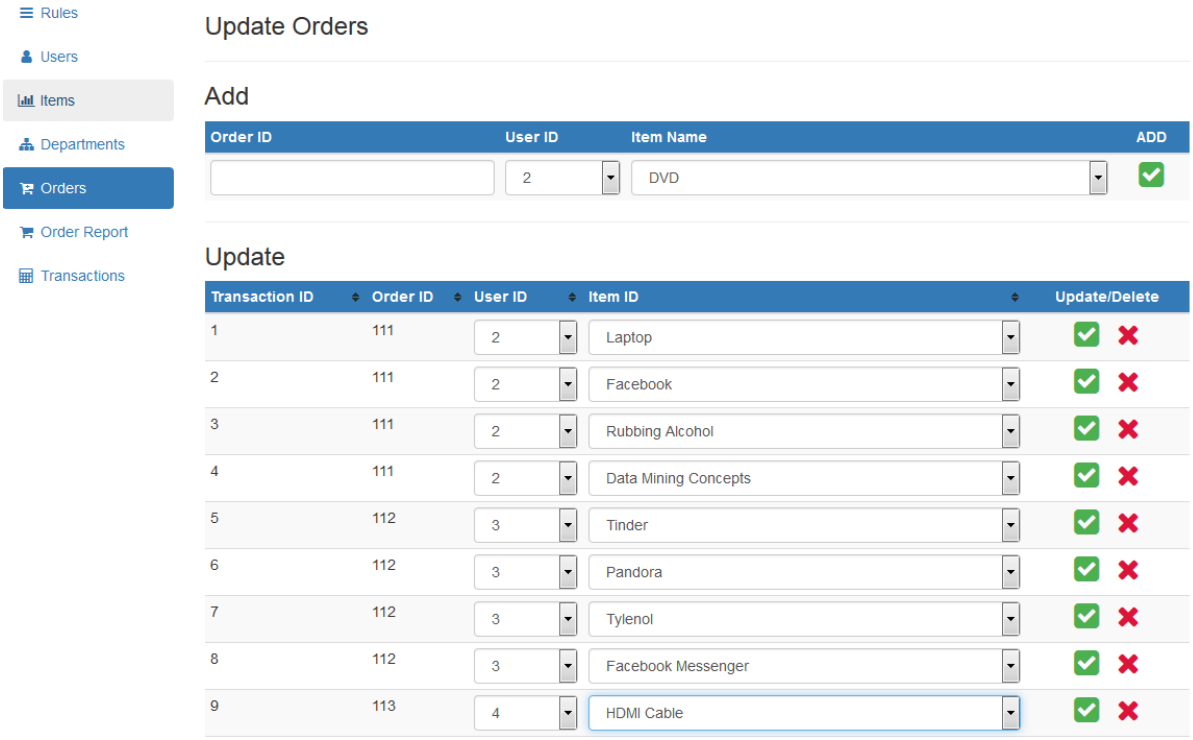
Abstract 3
Application Overview 3
Association Rules Overview 5
The Apriori Algorithm 7
Database Design Overview 7
Stored Procedure Algorithm Processing 9
System Development Environment: 10
Installation Instructions 10
References 11
Appendix 1: SQL Code Reference 12
Appendix 2 : Web Code Reference 13

Abstract

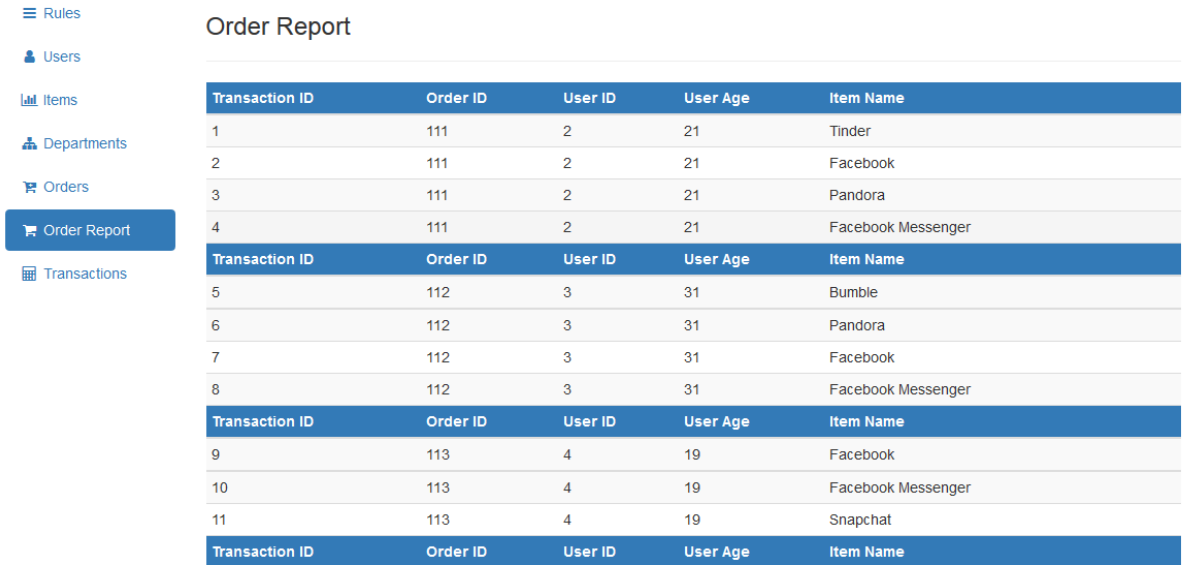
The Apriori algorithm is one of the most prevalent methods of pattern recognition in machine learning. It allows for the generation of association rules from large datasets, providing a means for which stakeholders can analyze valuable criterion in user interaction. To illustrate the usefulness of the algorithm, a small database was created emulating an online storefront. The storefront accepts user input for confidence and support values and then generates association rules using the Apriori algorithm.

Application Overview

The storefront contains a simple interface that allows users to add, update, and delete orders via drop down menus:

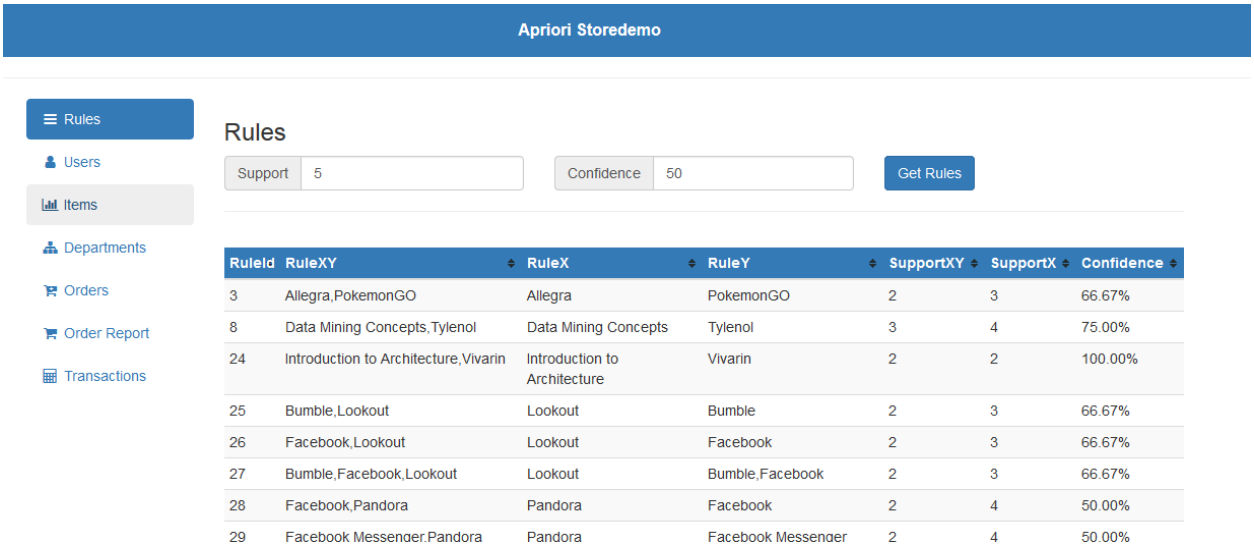


Order transactions are stored in the database grouped by orderID:



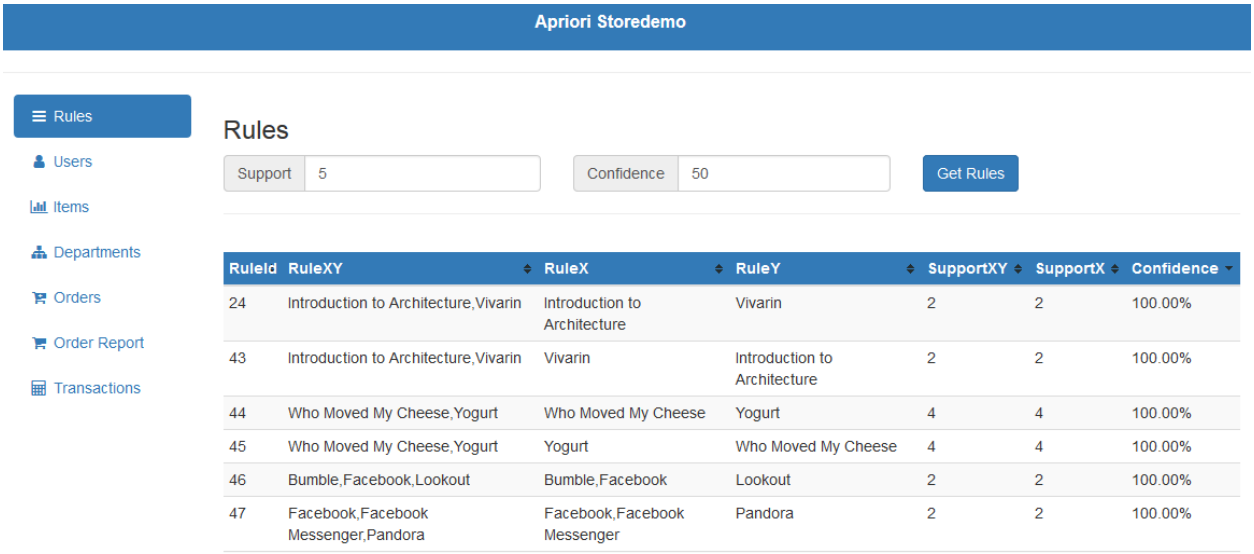
Transaction ID	Order ID	User ID	User Age	Item Name
1	111	2	21	Tinder
2	111	2	21	Facebook
3	111	2	21	Pandora
4	111	2	21	Facebook Messenger
Transaction ID	**Order ID**	**User ID**	**User Age**	**Item Name**
5	112	3	31	Bumble
6	112	3	31	Pandora
7	112	3	31	Facebook
8	112	3	31	Facebook Messenger
Transaction ID	**Order ID**	**User ID**	**User Age**	**Item Name**
9	113	4	19	Facebook
10	113	4	19	Facebook Messenger
11	113	4	19	Snapchat
Transaction ID	**Order ID**	**User ID**	**User Age**	**Item Name**

After a sufficient amount of orders are inputted, the user can then input a given support and confidence level for which the application will calculate and return all association rules and confidence values:



RuleId	RuleXY	RuleX	RuleY	SupportXY	SupportX	Confidence
3	Allegra,PokemonGO	Allegra	PokemonGO	2	3	66.67%
8	Data Mining Concepts,Tylenol	Data Mining Concepts	Tylenol	3	4	75.00%
24	Introduction to Architecture,Vivarin	Introduction to Architecture	Vivarin	2	2	100.00%
25	Bumble,Lookout	Lookout	Bumble	2	3	66.67%
26	Facebook,Lookout	Lookout	Facebook	2	3	66.67%
27	Bumble,Facebook,Lookout	Lookout	Bumble,Facebook	2	3	66.67%
28	Facebook,Pandora	Pandora	Facebook	2	4	50.00%
29	Facebook Messenger,Pandora	Pandora	Facebook Messenger	2	4	50.00%

Users can also sort by Confidence and Support Values:



RuleId	RuleXY	RuleX	RuleY	SupportXY	SupportX	Confidence
24	Introduction to Architecture,Vivarin	Introduction to Architecture	Vivarin	2	2	100.00%
43	Introduction to Architecture,Vivarin	Vivarin	Introduction to Architecture	2	2	100.00%
44	Who Moved My Cheese,Yogurt	Who Moved My Cheese	Yogurt	4	4	100.00%
45	Who Moved My Cheese,Yogurt	Yogurt	Who Moved My Cheese	4	4	100.00%
46	Bumble,Facebook,Lookout	Bumble,Facebook	Lookout	2	2	100.00%
47	Facebook,Facebook Messenger,Pandora	Facebook,Facebook Messenger	Pandora	2	2	100.00%

Association Rules Overview

Association rule analysis and generation provides a means for which patterns of behavior can be recognized within a dataset. If a dataset comprises of X amount of items, how are these items related? What patterns can we distinguish from these items? Is there any type of correlation between one item and another?

Item sets are comprised of multiple elements. For example, a customer can shop at a store that sells thousands of distinct elements, but the subset of elements they choose to buy are contained in one item set. Patterns of distinction can be discovered within these item sets and analyzed.

These patterns are known as association rules and can dictate with a certain level of support and confidence, how likely the rule is to have an outcome.

Association rules are generally presented in the following format:

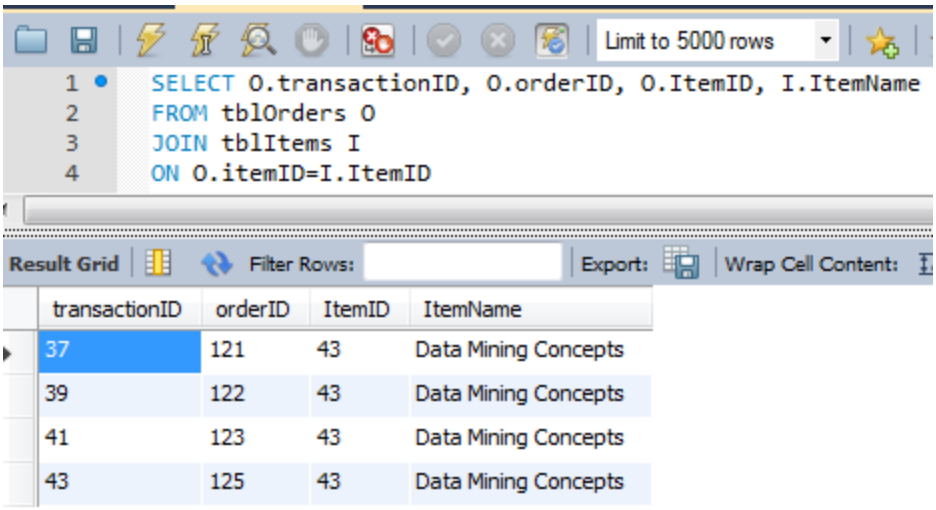
Data Mining Concepts -> Tylenol [4%, 75%]

The above rule is a relationship between the book “Data Mining Concepts” and the medication “Tylenol”, followed by a 4% support and 75% confidence. Association rules are read as simple if-then statements where the first item (if) is known as the antecedent and the second item (then) is known as a consequent.

Antecendent -> Consequent [Support, Confidence]

Support for the antecedent is calculated by the frequency in which the items appear in the record set.

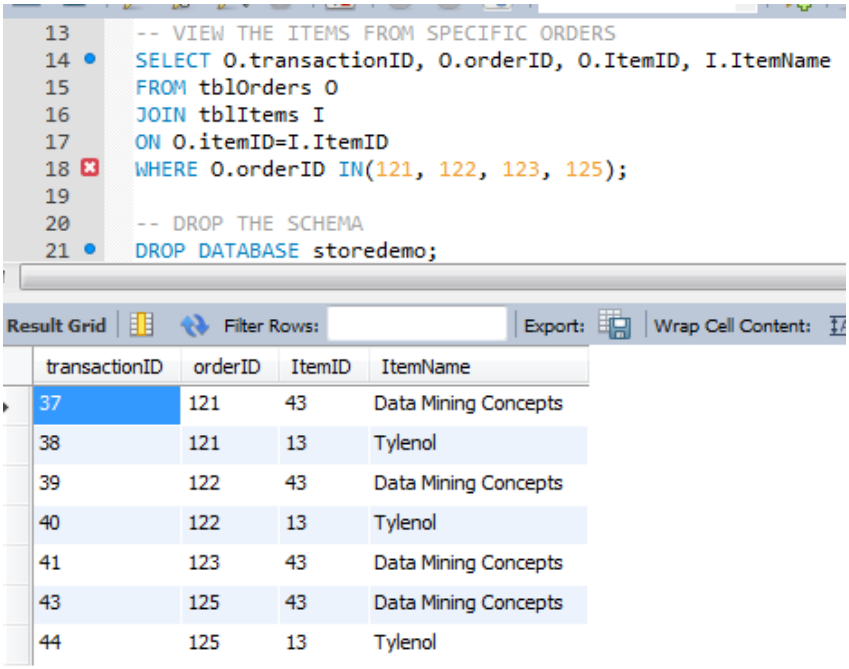
In this case the book “Data Mining Concepts” appears in the database in 4 orders:



Confidence is calculated by multiplying the support of both the antecedent and consequent and dividing it by the support for the antecedent:

$$\boldsymbol{Confidence}\ \{\boldsymbol{X},\boldsymbol{Y}\} = \frac{\textbf{Support}\ \{\mathbf{X},\mathbf{Y}\}}{\textbf{Support X}}$$

Using the above example of the orders with the antecedent "Data Mining Concepts", it is observed that although there are 4 orders with that item, 3 have Tylenol associated with them. This would suggest that although this book is not purchased frequently, it is almost always purchased with a bottle of Tylenol. One can draw conclusions from this observation. Perhaps suggesting that this book gives its readers headaches?

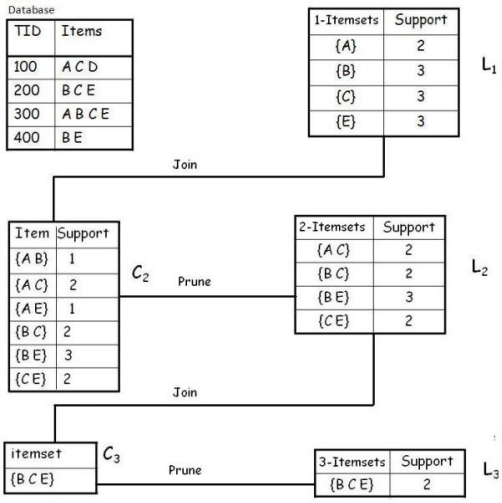


Similarly, people who purchased the book “Introduction to Architecture” had a need for Vivarin, a caffeine supplement. 66% of users who downloaded the outdoor mobile game PokemonGo, also purchased allergy medication.

RuleId	RuleXY	RuleX	RuleY	SupportXY	SupportX	Confidence
3	Allegra,PokemonGO	Allegra	PokemonGO	2	3	66.67%
8	Data Mining Concepts,Tylenol	Data Mining Concepts	Tylenol	3	4	75.00%
24	Introduction to Architecture,Vivarin	Introduction to Architecture	Vivarin	2	2	100.00%

The Apriori Algorithm

The Apriori algorithm generates the given support and confidence of an association rule through a series of Cartesian product calculations. Given a minimum support and confidence, the algorithm prunes the dataset and combines all possible combinations of the item sets thus generating strong association rules.



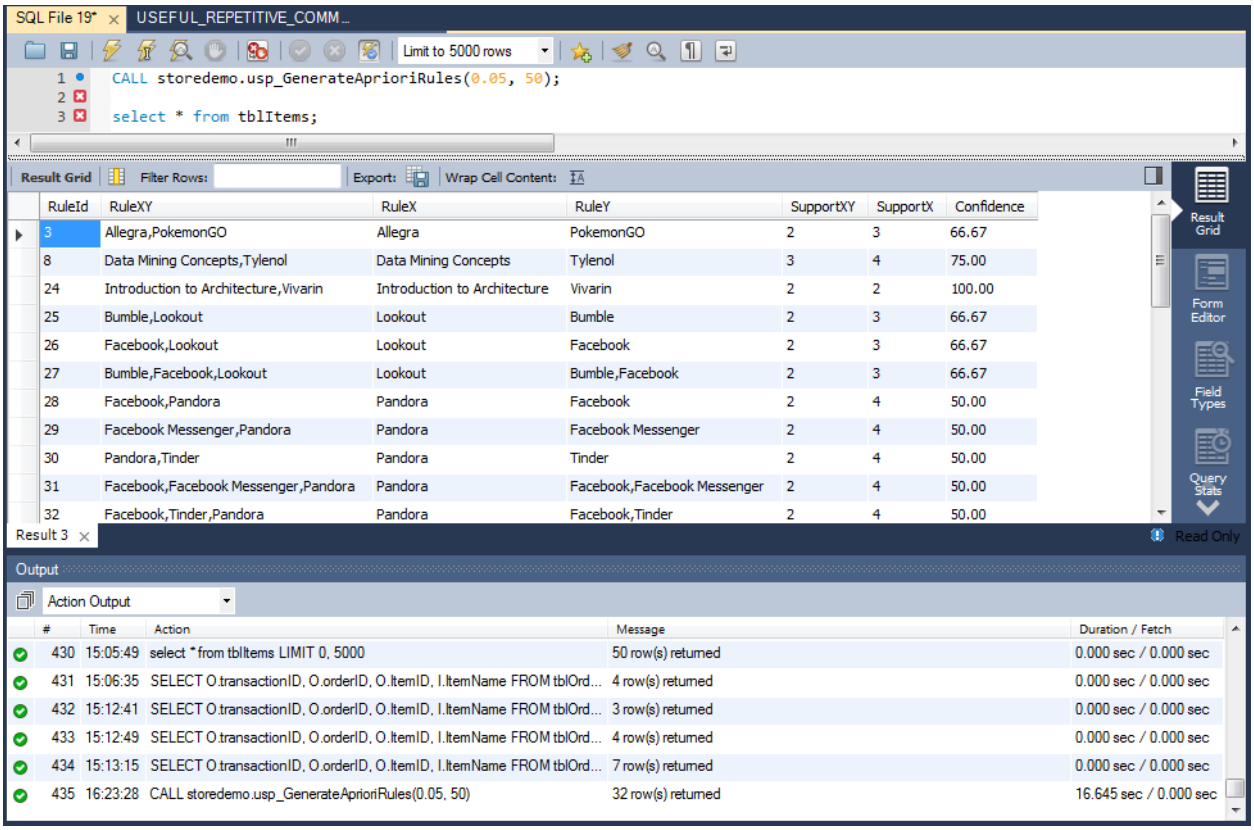
Rules	Support (X Y)	Support(X)	Confidence
{A} -> {C}	2	2	100
{B} -> {C}	2	3	66.66666667
{B} -> {E}	3	3	100
{C} -> {E}	2	3	66.66666667
{B} -> {C E}	2	3	66.66666667
{C} -> {B E}	2	3	66.66666667
{E} -> {B C}	2	3	66.66666667
{C} -> {A}	2	3	66.66666667
{C} -> {B}	2	3	66.66666667
{E} -> {B}	3	3	100
{E} -> {C}	2	3	66.66666667
{C E} -> {B}	2	2	100
{B E} -> {C}	2	3	66.66666667
{B C} -> {E}	2	2	100

(Reference: codeproject.com)

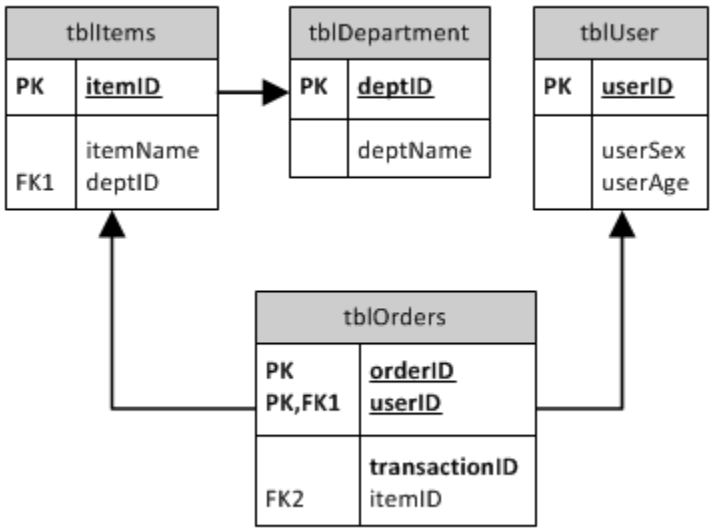
Database Design Overview

To illustrate the use of the algorithm, a rudimentary database was designed to hold transactions in relational form. The calculations were then completed through a series of stored procedures in the database.

NOTE: This approach is purely for academic purposes. Although it fulfills the function of the Apriori algorithm and pattern recognition, it is not suitable for large scale production environments. Refer to run time calculation of 16 seconds:

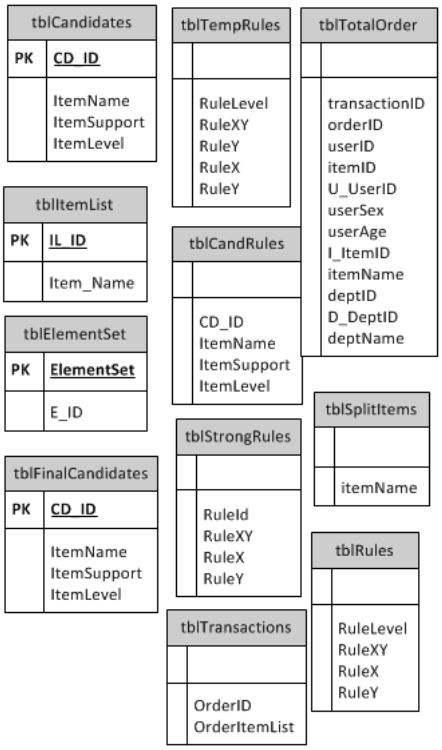


Run time for this procedure is 16.645 seconds for a total of 74 rows



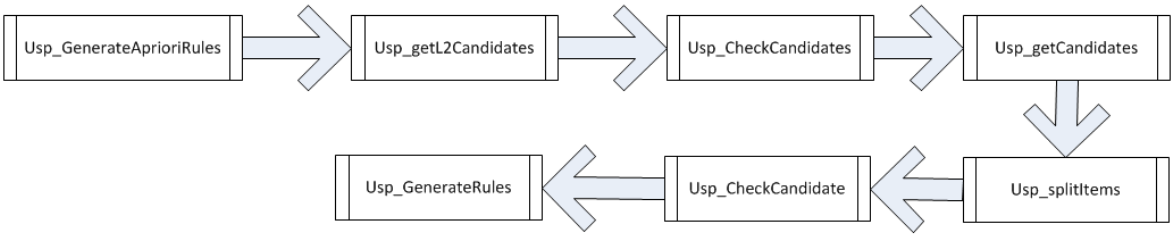
Rudimentary Order Database

The database stores all orders in a combined order table referencing both the item and user tables. A series of transactional tables were created in or to accomplish the rule generation. A series of cross joins was utilized to generate the different product combinations:



Transactional Tables

Stored Procedure Algorithm Processing



Stored Procedure Flow Diagram

The process begins with the stored procedure usp_GenerateAprioriRules which accepts parameters minSupport and confidence. These values are passed in from the php input boxes. Data distributed across the relational tables 'tblorders', 'tbluser', 'tblitems', 'tbldepartment' is generated in to a single non-relational, denormalized table 'tblTotalOrder' to consolidate all data into one table for preprocessing. All distinct items from orders are respectively generated in the table 'tblItemList'.

Items are inherently stored in the table row wise per order. i.e for a single orderID, items are placed in multiple rows. To facilitate easier access, all the items for an orderID are concatenated in to a single comma separated value string and this data is stored in the table 'tblTransactions'.

Single item sets are then generated according to the support and confidence value selected by the user. This data is stored in the table 'tblCandidates'. Using this table, level two candidates are generated using the stored candidates in Level 1. Each record in Level 2 candidates are checked in the transaction table to verify if the candidate is subset of at least one order and the support is calculated using the stored procedure 'usp_CheckCandidate'.

Item sets with more than 2 items (i.e. 3, 4) are generated using the 'usp_getCandidates' in a similar way using the Level 2 Items and Level 1 Items from the table 'tblCandidates'. Candidates from each level are checked in the transaction table to verify if the candidate is subset of at least one order and the support is calculated. This process continues until either there are no candidates generated at a level or all the levels are completed.

Once Candidate Item sets are generated, duplicates are removed and stored in the table 'tblFinalCandidates'. Only item sets with at least two items are used for Rule Generation. Candidates with a minimum of two items in the list are moved in to the table 'tblCandRules'. This table is used to generate initial rules using the Stored Procedure 'usp_GenerateRules'.

Each record in the table 'tblCandRules' is split into column level data and is stored in the table 'tempSplitCands'. This is used to self join the same table to generate second level rules and is stored in the table 'tbltempRules'. At the end of rule generation for each record set, data is inserted in to the table 'tblTempRules'. Distinct rules are taken from these tables are moved in to the table 'tblRules'.

The rules table (tblRules) is joined with the final candidates table (tblFinalCandidates) to get the support values and derive Confidence value for the rule.

System Development Environment:

Intel i5-4310M Dual Core Processor (2.7GHz x 2)
Windows 7 Enterprise
MySql WorkBench version 6.3
WAMP Server version 3.06 (MySQL version 5.7.14, Apache version 2.4.23, PHP 4.6.4)
PHP Bootstrap API

Installation Instructions

Project is available in a single zip file which contains both the PHP and SQL. Once WAMP is installed, the SQL scripts are to be run in their sequential order. Additional installation instructions available upon request.

References

Apriori Algorithm
Omar Salem - https://www.codeproject.com/Articles/70371/Apriori-Algorithm

Frequent Item sets | Mining of Massive Datasets | Stanford University
https://www.youtube.com/watch?v=O9QnC5WJJ90&index=8&list=PL-EtckxMpdUhgdQBGsWXtqhT7R-h3U22o

KDnuggets – Apriori Algorithm Tutorial
http://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html

Oracle Data Mining Concepts
https://docs.oracle.com/database/122/DMCON/apriori.htm#DMCON061

Oracle Advanced Analytics Data Mining Algorithms and Functions SQL API
http://www.oracle.com/technetwork/database/options/advanced-analytics/odm/odm-techniques-algorithms-097163.html

The Apriori Algorithm ... How The Apriori Algorithm Works
DrNoureddinSadawi - https://www.youtube.com/watch?v=Hk1zFOMLTrw

GROUP_CONCAT Function Use:
http://www.w3resource.com/mysql/aggregate-functions-and-grouping/aggregate-functions-and-grouping-group_concat.php

Server Side PREPARE Statements:
https://dev.mysql.com/doc/refman/5.7/en/sql-syntax-prepared-statements.html

MySQL Find_IN_SET():
http://www.w3resource.com/mysql/string-functions/mysql-find_in_set-function.php

Market Basket Analysis
Experfy - https://www.youtube.com/watch?v=WxDV9WEYqPw&index=1&list=PL-EtckxMpdUhgdQBGsWXtqhT7R-h3U22o

A Priori Algorithm | Mining of Massive Datasets | Stanford University
https://www.youtube.com/watch?v=tY1JE6XFjCY&index=7&list=PL-EtckxMpdUhgdQBGsWXtqhT7R-h3U22o

Appendix 1: SQL Code Reference

```
-- phpMyAdmin SQL Dump
-- version 4.6.4
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Mar 03, 2017 at 08:39 PM
-- Server version: 5.7.14
-- PHP Version: 5.6.25

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `storedemo`
--
CREATE DATABASE IF NOT EXISTS `storedemo` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci;
USE `storedemo`;

-- --------------------------------------------------------

--
-- Table structure for table `tbldepartment`
--
-- Creation: Mar 02, 2017 at 08:06 PM
-- Last update: Mar 02, 2017 at 08:13 PM
--

DROP TABLE IF EXISTS `tbldepartment`;
CREATE TABLE `tbldepartment` (
  `deptID` int(2) NOT NULL,
  `deptName` varchar(25) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `tbldepartment`
--

INSERT INTO `tbldepartment` (`deptID`, `deptName`) VALUES(1, 'Electronics');
INSERT INTO `tbldepartment` (`deptID`, `deptName`) VALUES(2, 'Pharmacy');
INSERT INTO `tbldepartment` (`deptID`, `deptName`) VALUES(3, 'App Store');
INSERT INTO `tbldepartment` (`deptID`, `deptName`) VALUES(4, 'Grocery');
INSERT INTO `tbldepartment` (`deptID`, `deptName`) VALUES(5, 'Books');

-- --------------------------------------------------------

--
-- Table structure for table `tblitems`
--
-- Creation: Mar 02, 2017 at 08:11 PM
-- Last update: Mar 02, 2017 at 08:34 PM
--

DROP TABLE IF EXISTS `tblitems`;
CREATE TABLE `tblitems` (
  `itemID` int(2) NOT NULL,
  `itemName` varchar(50) NOT NULL,
  `deptID` int(2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
-- Dumping data for table `tblitems`
--

INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(2, 'DVD', 1);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(3, 'Video Game', 1);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(4, 'Laptop', 1);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(5, 'Tablet', 1);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(6, 'USB Cable', 1);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(7, 'HDMI Cable', 1);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(8, 'Television', 1);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(9, 'Router', 1);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(10, 'Monitor', 1);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(11, 'Batteries', 1);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(12, 'Vivarin', 2);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(13, 'Tylenol', 2);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(14, 'Ibruprofen', 2);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(15, 'Rubbing Alcohol', 2);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(16, 'Peroxide', 2);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(17, 'Alka Seltzer', 2);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(18, 'Allegra', 2);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(19, 'Nyquil', 2);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(20, 'Multi-Vitamin', 2);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(21, 'Zantac', 2);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(22, 'Tinder', 3);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(23, 'Facebook', 3);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(24, 'Bumble', 3);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(25, 'Pandora', 3);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(26, 'Lookout', 3);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(27, 'PokemonGO', 3);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(28, 'Minecraft', 3);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(29, 'Snapchat', 3);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(30, 'CandyCrush', 3);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(31, 'Facebook Messenger', 3);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(32, 'Yogurt', 4);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(33, 'Bananas', 4);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(34, 'Coca Cola', 4);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(35, 'Brocolli', 4);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(36, 'Tabasco Sauce', 4);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(37, 'Frozen Meals', 4);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(38, 'Ramen Noodles', 4);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(39, 'Soy Milk', 4);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(40, 'Bread', 4);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(41, 'Organic Bananas', 4);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(42, '50 Shades of Grey', 5);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(43, 'Data Mining Concepts', 5);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(44, 'Database Administration for
Dummies', 5);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(45, 'Introduction to
Architecture', 5);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(46, 'Hunger Games', 5);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(47, 'Who Moved My Cheese', 5);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(48, 'Harry Potter', 5);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(49, 'The Martian', 5);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(50, 'Advanced Analytics', 5);
INSERT INTO `tblitems` (`itemID`, `itemName`, `deptID`) VALUES(51, 'Dr. Seuss', 5);

-- --------------------------------------------------------

--
-- Table structure for table `tblorders`
--
-- Creation: Mar 02, 2017 at 09:46 PM
```

```
-- Last update: Mar 03, 2017 at 08:29 PM
--

DROP TABLE IF EXISTS `tblorders`;
CREATE TABLE `tblorders` (
  `transactionID` int(5) NOT NULL,
  `orderID` int(4) NOT NULL,
  `userID` int(4) NOT NULL,
  `itemID` int(2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `tblorders`
--

INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(1, 111, 2, 22);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(2, 111, 2, 23);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(3, 111, 2, 25);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(4, 111, 2, 31);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(5, 112, 3, 24);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(6, 112, 3, 25);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(7, 112, 3, 23);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(8, 112, 3, 31);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(9, 113, 4, 23);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(10, 113, 4, 31);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(11, 113, 4, 29);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(12, 114, 5, 28);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(13, 114, 5, 30);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(14, 114, 5, 23);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(15, 115, 6, 22);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(16, 115, 6, 23);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(17, 115, 6, 31);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(18, 115, 6, 25);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(19, 116, 7, 26);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(20, 116, 7, 24);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(21, 116, 7, 23);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(22, 117, 8, 24);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(23, 117, 8, 23);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(24, 117, 8, 26);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(25, 118, 9, 22);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(26, 118, 9, 24);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(27, 118, 9, 25);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(28, 118, 9, 23);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(29, 118, 9, 31);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(30, 119, 10, 27);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(31, 119, 10, 29);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(32, 119, 10, 23);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(33, 119, 10, 31);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(34, 120, 11, 30);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(35, 120, 11, 28);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(36, 120, 11, 27);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(37, 121, 6, 43);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(38, 121, 6, 13);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(39, 122, 7, 43);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(40, 122, 7, 13);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(41, 123, 3, 43);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(42, 124, 3, 13);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(43, 125, 8, 43);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(44, 125, 8, 13);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(45, 126, 4, 45);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(46, 126, 4, 12);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(47, 127, 4, 45);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(48, 127, 4, 12);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(49, 128, 8, 50);
```

```
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(50, 128, 8, 17);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(51, 129, 3, 50);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(52, 129, 3, 17);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(53, 128, 7, 50);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(54, 128, 7, 17);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(55, 129, 7, 42);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(56, 129, 7, 24);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(57, 130, 6, 42);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(58, 130, 7, 22);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(59, 131, 3, 42);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(60, 131, 3, 24);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(61, 132, 8, 42);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(62, 132, 8, 24);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(63, 133, 9, 42);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(64, 133, 9, 22);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(65, 134, 5, 47);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(66, 134, 5, 32);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(67, 135, 11, 47);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(68, 135, 11, 32);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(69, 136, 8, 47);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(70, 136, 8, 32);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(71, 137, 3, 47);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(72, 137, 3, 32);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(75, 139, 4, 26);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(76, 139, 4, 18);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(77, 140, 2, 27);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(78, 140, 2, 18);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(79, 138, 9, 27);
INSERT INTO `tblorders` (`transactionID`, `orderID`, `userID`, `itemID`) VALUES(80, 138, 9, 18);

-- --------------------------------------------------------

--
-- Table structure for table `tbluser`
--
-- Creation: Mar 02, 2017 at 09:07 PM
-- Last update: Mar 02, 2017 at 08:41 PM
--

DROP TABLE IF EXISTS `tbluser`;
CREATE TABLE `tbluser` (
  `userID` int(4) NOT NULL,
  `userSex` varchar(1) NOT NULL,
  `userAge` int(2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `tbluser`
--

INSERT INTO `tbluser` (`userID`, `userSex`, `userAge`) VALUES(2, 'F', 21);
INSERT INTO `tbluser` (`userID`, `userSex`, `userAge`) VALUES(3, 'F', 31);
INSERT INTO `tbluser` (`userID`, `userSex`, `userAge`) VALUES(4, 'F', 19);
INSERT INTO `tbluser` (`userID`, `userSex`, `userAge`) VALUES(5, 'F', 45);
INSERT INTO `tbluser` (`userID`, `userSex`, `userAge`) VALUES(6, 'F', 25);
INSERT INTO `tbluser` (`userID`, `userSex`, `userAge`) VALUES(7, 'M', 30);
INSERT INTO `tbluser` (`userID`, `userSex`, `userAge`) VALUES(8, 'M', 35);
INSERT INTO `tbluser` (`userID`, `userSex`, `userAge`) VALUES(9, 'M', 22);
INSERT INTO `tbluser` (`userID`, `userSex`, `userAge`) VALUES(10, 'M', 18);
INSERT INTO `tbluser` (`userID`, `userSex`, `userAge`) VALUES(11, 'M', 55);

--
-- Indexes for dumped tables
--
```

```
--
-- Indexes for table `tbldepartment`
--
ALTER TABLE `tbldepartment`
  ADD KEY `deptID` (`deptID`);

--
-- Indexes for table `tblitems`
--
ALTER TABLE `tblitems`
  ADD PRIMARY KEY (`itemID`),
  ADD KEY `deptID` (`deptID`);

--
-- Indexes for table `tblorders`
--
ALTER TABLE `tblorders`
  ADD PRIMARY KEY (`transactionID`);

--
-- Indexes for table `tbluser`
--
ALTER TABLE `tbluser`
  ADD PRIMARY KEY (`userID`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `tblitems`
--
ALTER TABLE `tblitems`
  MODIFY `itemID` int(2) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=52;
--
-- AUTO_INCREMENT for table `tblorders`
--
ALTER TABLE `tblorders`
  MODIFY `transactionID` int(5) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=118;
--
-- AUTO_INCREMENT for table `tbluser`
--
ALTER TABLE `tbluser`
  MODIFY `userID` int(4) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=12;
--
-- Constraints for dumped tables
--

--
-- Constraints for table `tblitems`
--
ALTER TABLE `tblitems`
  ADD CONSTRAINT `fk_deptID` FOREIGN KEY (`deptID`) REFERENCES `tbldepartment` (`deptID`) ON
  UPDATE CASCADE;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

```
CREATE TABLE `tblcandidates` (
  `CD_ID` int(11) NOT NULL AUTO_INCREMENT,
  `ItemName` varchar(60000) DEFAULT NULL,
  `ItemSupport` int(11) DEFAULT NULL,
  `ItemLevel` int(11) DEFAULT NULL,
  PRIMARY KEY (`CD_ID`)
) ENGINE=InnoDB AUTO_INCREMENT=63 DEFAULT CHARSET=latin1;

CREATE TABLE `tblcandrules` (
  `CD_ID` int(11) NOT NULL AUTO_INCREMENT,
  `ItemName` varchar(60000) DEFAULT NULL,
  `ItemSupport` int(11) DEFAULT NULL,
  `ItemLevel` int(11) DEFAULT NULL,
  PRIMARY KEY (`CD_ID`)
) ENGINE=InnoDB AUTO_INCREMENT=32 DEFAULT CHARSET=latin1;


CREATE TABLE `tblelementset` (
  `E_ID` int(11) NOT NULL AUTO_INCREMENT,
  `ElementSet` varchar(60000) DEFAULT NULL,
  PRIMARY KEY (`E_ID`)
) ENGINE=InnoDB AUTO_INCREMENT=128 DEFAULT CHARSET=latin1;

CREATE TABLE `tblfinalcandidates` (
  `CD_ID` int(11) NOT NULL AUTO_INCREMENT,
  `ItemName` varchar(60000) DEFAULT NULL,
  `ItemSupport` int(11) DEFAULT NULL,
  `ItemLevel` int(11) DEFAULT NULL,
  PRIMARY KEY (`CD_ID`)
) ENGINE=InnoDB AUTO_INCREMENT=64 DEFAULT CHARSET=latin1;

CREATE TABLE `tblitemlist` (
  `IL_ID` int(11) NOT NULL AUTO_INCREMENT,
  `ItemName` varchar(500) DEFAULT NULL,
  PRIMARY KEY (`IL_ID`)
) ENGINE=InnoDB AUTO_INCREMENT=64 DEFAULT CHARSET=latin1;

CREATE TABLE `tblsplititems` (
  `itemName` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `tbltemprules` (
  `RuleLevel` int(11) DEFAULT NULL,
  `RuleXY` varchar(20000) DEFAULT NULL,
  `RuleX` varchar(20000) DEFAULT NULL,
  `RuleY` varchar(20000) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `tblstrongrules` (
  `RuleId` int(11) NOT NULL AUTO_INCREMENT,
  `RuleXY` varchar(20000) DEFAULT NULL,
  `RuleX` varchar(20000) DEFAULT NULL,
  `RuleY` varchar(20000) DEFAULT NULL,
  `SupportXY` int(11) DEFAULT NULL,
  `SupportX` int(11) DEFAULT NULL,
  `Confidence` decimal(5,2) DEFAULT NULL,
  PRIMARY KEY (`RuleId`)
) ENGINE=InnoDB AUTO_INCREMENT=128 DEFAULT CHARSET=latin1;

CREATE TABLE `tblrules` (
  `RuleLevel` int(11) DEFAULT NULL,
  `RuleXY` varchar(20000) DEFAULT NULL,
  `RuleX` varchar(20000) DEFAULT NULL,
  `RuleY` varchar(20000) DEFAULT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `tbltotalorder` (
  `transactionID` int(5) NOT NULL DEFAULT '0',
  `orderID` int(4) NOT NULL,
  `userID` int(4) NOT NULL,
  `itemID` int(2) NOT NULL,
  `U_UserID` int(4) NOT NULL DEFAULT '0',
  `userSex` varchar(1) NOT NULL,
  `userAge` int(2) NOT NULL,
  `I_itemID` int(2) NOT NULL DEFAULT '0',
  `itemName` varchar(50) NOT NULL,
  `deptID` int(2) NOT NULL,
  `D_deptID` int(2) NOT NULL,
  `deptName` varchar(25) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `tbltransactions` (
  `OrderID` int(11) DEFAULT NULL,
  `OrderItemList` varchar(60000) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` FUNCTION `SPLIT_STRING`( s VARCHAR(1024) , del CHAR(1) , i
INT) RETURNS varchar(1024) CHARSET latin1
    DETERMINISTIC
BEGIN

        DECLARE n INT ;

        -- get max number of items
        SET n = LENGTH(s) - LENGTH(REPLACE(s, del, '')) + 1;

        IF i > n THEN
            RETURN NULL ;
        ELSE
            RETURN SUBSTRING_INDEX(SUBSTRING_INDEX(s, del, i) , del , -1 ) ;
        END IF;

    END$$
DELIMITER ;
```

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `usp_CheckCandidate`(IN citlevel INT,IN EID INT,IN
minSupport DECIMAL(5,2), IN transactionCount INT)
BEGIN

DECLARE itemSum INT;
DECLARE elementString VARCHAR(60000);
SET itemSum = 0;

SET elementString = (SELECT GROUP_CONCAT(ElementSet) as Data FROM tblelementset
WHERE E_ID = EID);


DROP TABLE IF EXISTS tempSplitElements;
CREATE TEMPORARY TABLE tempSplitElements (itemName CHAR(255));
SET @S2 = CONCAT("INSERT INTO tempSplitElements (itemName) VALUES ('",REPLACE((SELECT
GROUP_CONCAT(ElementSet) as Data FROM tblelementset
WHERE E_ID = EID), ",", "'),('"),"');");

PREPARE stmt1 FROM @S2;
EXECUTE stmt1;

SET itemSum = (SELECT COUNT(*) FROM tempSplitElements);

INSERT INTO tblCandidates(ItemName,ItemSupport,ItemLevel)
SELECT ABCDE.* FROM
(SELECT  elementString AS itemName,
    COUNT(ABCD.orderID) AS itemSupport,
    citlevel  + 1 AS ItemLevel FROM
(SELECT OrderID,OrderItemList,SUM(ItemExists) AS SumItems FROM
(SELECT DISTINCT itemName,OrderID,OrderItemList,ItemExists FROM
(SELECT *, CASE WHEN find_in_set(A.itemName,B.OrderItemList) = 0 THEN 0 ELSE 1 END AS
ItemExists FROM tempSplitElements A
 CROSS JOIN tbltransactions B) AB) ABC
 GROUP BY OrderID,OrderItemList
HAVING SumItems = itemSum) ABCD
GROUP BY ABCD.OrderItemList
ORDER BY ABCD.OrderItemList) ABCDE
WHERE ABCDE.itemsupport/transactionCount >= minSupport;

 END$$
DELIMITER ;
```

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `usp_getCandidates`(IN minSupport DECIMAL(5,2), IN
transactionCount INT, IN CLevel INT)
BEGIN

DECLARE ctr INT;
declare cnt int;
declare itlevel  INT;
declare elementCount INT;
declare candCheckCtr INT;
SET ctr=1;
set cnt=0;
set elementCount =0;
set candCheckCtr =1;
set itlevel = CLevel;


myloop: WHILE (ctr < 100)
DO

 CALL storedemo.usp_splitItems(itlevel);

TRUNCATE TABLE tblElementSet;
INSERT INTO tblElementSet(ElementSet)
SELECT DISTINCT case when strcmp(TC.itemName,TS.itemName) = 1 then
concat(TS.itemName,',',TC.itemName)
ELSE concat(TC.itemName,',',TS.itemName) END AS itemName FROM tblCandidates TC
CROSS JOIN tblSplitItems TS
ON TC.ItemLevel = itlevel
AND find_in_set(TS.itemName,TC.itemName) = 0;


SET elementCount = (SELECT COUNT(*) FROM tblElementSet);

WHILE(candCheckCtr <= elementCount)
DO
CALL storedemo.usp_CheckCandidate(itlevel,candCheckCtr,minSupport,transactionCount);
set candCheckCtr = candCheckCtr + 1;
END WHILE;

set itlevel  = itlevel  + 1;
set ctr = ctr+1;

   SET cnt = (SELECT COUNT(*) FROM tblCandidates WHERE ItemLevel = itlevel);
   IF cnt = 0 THEN
      -- SELECT cnt;
      LEAVE myloop;
   END IF;
END WHILE;

 END$$
DELIMITER ;
```

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `usp_splitItems`(IN ILevel INT)
BEGIN

DROP TABLE IF EXISTS tempSplitItems;
CREATE TEMPORARY TABLE tempSplitItems (itemName CHAR(255));
SET @S3 = CONCAT("INSERT INTO tempSplitItems (itemName) VALUES ('",REPLACE((SELECT
GROUP_CONCAT(itemName) as Data FROM tblCandidates WHERE ItemLevel = ILevel), ",",
"'),('"),"');");
PREPARE stmt1 FROM @S3;
EXECUTE stmt1;

TRUNCATE TABLE tblSplitItems;
INSERT INTO tblSplitItems
SELECT DISTINCT(itemName) FROM tempSplitItems Order By itemName;

DROP TABLE tempSplitItems;

 END$$
DELIMITER ;
```

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `usp_getL2Candidates`(IN minSupport DECIMAL(5,2),
IN transactionCount INT, IN CLevel INT)
BEGIN
 DECLARE x INT;

declare elementCount INT;
declare candCheckCtr INT;
SET x = 1;
set elementCount =0;
set candCheckCtr =1;

TRUNCATE TABLE tblElementSet;

INSERT INTO tblElementSet(ElementSet)
SELECT
    CONCAT(B.ItemName, ',', A.ItemName) AS ItemName
FROM
    (SELECT
        *
    FROM
        tblCandidates
    WHERE
        ItemLevel = CLevel) A
        CROSS JOIN
    (SELECT
        *
    FROM
        tblCandidates
    WHERE
        ItemLevel = CLevel) B ON A.CD_ID > B.CD_ID;



SET elementCount = (SELECT COUNT(*) FROM tblElementSet);

WHILE(candCheckCtr <= elementCount)
DO
CALL storedemo.usp_CheckCandidate(CLevel,candCheckCtr,minSupport,transactionCount);
set candCheckCtr = candCheckCtr + 1;
END WHILE;

 END$$
DELIMITER ;
```

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `usp_GenerateRules`()
BEGIN

DECLARE CRCount INT;
DECLARE RCtr INT;
DECLARE LCtr INT;
DECLARE LCount INT;
DECLARE CRItemName VARCHAR(60000);

SET CRCount = (SELECT COUNT(1) FROM tblCandRules);
SET RCtr = 1;
SET LCtr = 1;

WHILE (RCtr <=CRCount)
DO

-- Get Candidate to Split
SET CRItemName = (SELECT itemName as iData FROM tblCandRules WHERE CD_ID = RCtr);

DROP TABLE IF EXISTS tempSplitCands;
CREATE TABLE tempSplitCands (itemName VARCHAR(255));
SET @S1 = CONCAT("INSERT INTO tempSplitCands (itemName) VALUES ('",REPLACE((SELECT
GROUP_CONCAT(CRItemName)), ",", "'),('"),"');");

PREPARE stmt1 FROM @s1;
EXECUTE stmt1;

TRUNCATE TABLE tbltempRules;
INSERT INTO tbltempRules
select 1,case when strcmp(a.itemName,b.itemName) = 1 then concat(b.itemName,',',a.itemName)
ELSE concat(a.itemName,',',b.itemName) END AS RuleXY, a.itemname AS RuleX,b.itemname AS RuleY
from tempSplitCands a
cross join tempSplitCands b
on a.itemname <>b.itemname;

SET LCount = (SELECT COUNT(*) FROM tempSplitCands);

WHILE (LCtr < LCount-1)
DO
INSERT INTO tbltempRules
SELECT DISTINCT ABC.RuleLevel, ABC.RuleXY, ABC.RuleX, ABC.RuleY
FROM
(SELECT 2 AS RuleLevel, case when strcmp(A.itemName,B.RuleXY) = 1 then
concat(B.RuleXY,',',A.itemName)
ELSE concat(A.itemName,',',B.RuleXY) END AS RuleXY, a.itemname AS RuleX,b.B.RuleXY AS RuleY
FROM tempSplitCands A
CROSS JOIN (SELECT * FROM tbltempRules
WHERE RuleLevel = LCtr) B
ON find_in_set(A.itemname,B.RuleXY) = 0) ABC;

SET LCtr = LCtr+1;
END WHILE;

INSERT INTO tbltempRules(RuleLevel,RuleXY,RuleX,RuleY)
SELECT RuleLevel,RuleXY,RuleY,RuleX
FROM tbltempRules
WHERE RuleLevel >1;

INSERT INTO tblRules
SELECT DISTINCT RuleLevel,RuleXY,RuleX,RuleY
FROM tbltempRules;

SET LCtr = 1;
```

```
SET RCtr = RCtr +1;
END WHILE;

END$$
DELIMITER ;
```

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `usp_GenerateAprioriRules`(IN GARminSupport
DECIMAL(5,2), IN GARconfidence DECIMAL(5,2))
BEGIN
DECLARE transactionCount INT;

TRUNCATE TABLE tblTotalOrder;
INSERT INTO tblTotalOrder
SELECT O.transactionID,O.orderID,O.userID,O.itemID,U.userID As U_UserID, U.userSex,
U.userAge, I.itemID AS I_itemID, I.itemName, I.deptID, D.deptID AS D_deptID, D.deptName
FROM tblorders O
JOIN tbluser U
ON O.userID = U.userID
JOIN tblitems I
ON O.itemID = I.itemID
JOIN tbldepartment D
ON D.deptID = I.deptID
ORDER BY O.orderID;

-- Get Distinct ItemList from all transactions.
TRUNCATE TABLE tblItemList;
INSERT INTO tblItemList(ItemName)
SELECT DISTINCT ItemName FROM tbltotalorder
ORDER BY itemName;

-- Get Transactions in form of comma separated values
TRUNCATE TABLE tblTransactions;
INSERT INTO tblTransactions(OrderID,OrderItemList)
select orderID, GROUP_CONCAT(DISTINCT itemname order by itemname) As OrderItems from
tbltotalorder
group by orderID
order by itemname;

-- Set MinSupport and Confidence
-- SELECT @minSupport:=0.05;
-- SELECT @confidence:=0.5;

-- Get total no. of transactions.
SET transactionCount = (SELECT COUNT(DISTINCT OrderID) FROM  tbltotalorder);

-- Insert Single element Candidates with support values.
TRUNCATE TABLE tblCandidates;
INSERT INTO  tblCandidates(ItemName,ItemSupport,ItemLevel)
SELECT C.* FROM
(Select itemName,count(distinct orderID) as itemsupport, 1 AS ItemLevel from tbltotalorder
group by itemName) C
WHERE C.itemsupport/transactionCount >= GARminSupport;


-- Call Stored Procedure to get Level2 element candidates.
CALL storedemo.usp_getL2Candidates(GARminSupport, transactionCount, 1);

-- Call Stored Procedure to get Level3 and above element candidates.
CALL storedemo.usp_getCandidates(GARminSupport, transactionCount, 2);

TRUNCATE TABLE tblFinalCandidates;
INSERT INTO tblFinalCandidates(ItemName,ItemSupport,ItemLevel)
select distinct ItemName,ItemSupport,ItemLevel from tblCandidates;

TRUNCATE TABLE tblCandRules;
INSERT INTO tblCandRules(ItemName,ItemSupport,ItemLevel)
SELECT ItemName,ItemSupport,ItemLevel FROM (
SELECT   *,LENGTH(itemName) - LENGTH(REPLACE(itemName, ',', ''))+1 As NoElements from
tblFinalCandidates) ABC
```

```
Where NoElements > 1;

TRUNCATE TABLE tblRules;
CALL storedemo.usp_GenerateRules();

TRUNCATE TABLE tblStrongRules;
INSERT INTO tblStrongRules(RuleXY,RuleX,RuleY,SupportXY,SupportX,Confidence)
SELECT A.RuleXY, A.RuleX, A.RuleY,B.ItemSupport AS SupportXY, C.ItemSupport AS SupportX,
CAST((B.ItemSupport/C.ItemSupport)*100 AS DECIMAL(5,2)) AS Confidence
FROM (SELECT DISTINCT RuleXY, RuleX,RuleY FROM tblRules) A
JOIN tblFinalCandidates B
ON A.RuleXY = B.ItemName
JOIN tblFinalCandidates C
ON A.RuleX = C.ItemName;


SELECT * FROM tblStrongRules
WHERE Confidence>= GARconfidence;

 END$$
DELIMITER ;
```

Appendix 2 : Web Code Reference

```
<h3>Update Departments</h3>
<hr>
<div id="notification"></div>
<h3>Add</h3>
<table class="table table-condensed table-hover table-striped bootgrid-table" cellspacing="0">
   <thead>
      <tr>
         <th>Department Id</th>
         <th>Department Name</th>
         <th style="text-align: center;">ADD</th>
      </tr>
   </thead>
   <tbody>
      <tr>
         <td ><input class="form-control" id="deptID" type="text" value=""></td>
         <td ><input class="form-control" id="deptName" type="text" value=""></td>
         <td style="text-align: center;"><i class="fa fa-check-square fa-fw fa-2x"
         aria-hidden="true" id="add_dept"></i></td>
      </tr>
   </tbody>
</table>
<hr>
<h3>Update</h3>

<table id="tbldepartment_sort" class="table table-condensed table-hover table-striped
bootgrid-table tablesorter" cellspacing="0">
   <thead>
      <tr>
         <th>Department ID</th>
         <th>Department Name</th>
         <th style="text-align: center;">Update/Delete</th>
      </tr>
   </thead>
   <tbody>
      <?php include("tbldepartments.php");?>
   </tbody>
</table>
<script src="js/departments.js"></script>
```

```php
<?php
//include connection file
include_once("../db/connection.php");
$sql = "SELECT * FROM `tbldepartment`";
$departments = mysqli_query($conn, $sql) or die("error to fetch employees data");
?>
<h3>Update Items</h3>
<hr>
<div id="notification"></div>
<h3>Add</h3>
<table class="table table-condensed table-hover table-striped bootgrid-table" cellspacing="0">
   <thead>
      <tr>
         <th>Item Name</th>
         <th>Department</th>
         <th style="text-align: center;">ADD</th>
      </tr>
   </thead>
   <tbody>
      <tr>
         <td ><input class="form-control" id="itemName" type="text" value=""></td>
         <td ><select class="form-control" id="deptID">
         <?php foreach($departments as $dep) :?>
            <option value="<?php echo $dep['deptID'];?>"><?php echo $dep['deptName'];?></option>
         <?php endforeach;?>
         </select></td>
         <td style="text-align: center;"><i class="fa fa-check-square fa-fw fa-2x"
         aria-hidden="true" id="add_item"></i></td>
      </tr>
   </tbody>
</table>
<hr>
<h3>Update</h3>
<table id="tblitems_sort" class="table table-condensed table-hover table-striped bootgrid-table
tablesorter" cellspacing="0">
   <thead>
      <tr>
         <th>Item ID</th>
         <th>Item Name</th>
         <th>Department</th>
         <th style="text-align: center;">Update/Delete</th>
      </tr>
   </thead>
   <tbody>
      <?php include("tblitems.php");?>
   </tbody>
</table>

<script src="js/items.js"></script>
```

```
<?php
//include connection file
include_once("../db/connection.php");
$sql = "SELECT * FROM `tbluser`";
$users = mysqli_query($conn, $sql) or die("error to fetch employees data");

$sql = "SELECT * FROM `tblitems`";
$items = mysqli_query($conn, $sql) or die("error to fetch employees data");
?>

<h3>Update Orders</h3>
<hr>
<div id="notification"></div>
<h3>Add</h3>
<table class="table table-condensed table-hover table-striped bootgrid-table" cellspacing="0">
   <thead>
      <tr>
         <th>Order ID</th>
         <th>User ID</th>
         <th>Item Name</th>
         <th style="text-align: center;">ADD</th>
      </tr>
   </thead>
   <tbody>
      <tr>
        <td ><input class="form-control" id="orderID" type="text" value=""></td>
         <td ><select class="form-control" id="userID">
                <?php foreach($users as $user) :?>
                  <option value="<?php echo $user['userID'];?>"><?php echo
                  $user['userID'];?></option>
               <?php endforeach;?>
         </select></td>
         <td ><select class="form-control" id="itemID">
               <?php foreach($items as $item) :?>
                  <option value="<?php echo $item['itemID'];?>"><?php echo
                  $item['itemName'];?></option>
               <?php endforeach;?>
         </select></td>
         <td style="text-align: center;"><i class="fa fa-check-square fa-fw fa-2x"
         aria-hidden="true" id="add_order"></i></td>
      </tr>
   </tbody>
</table>
<hr>
<h3>Update</h3>

<table id="tblorders_sort" class="table table-condensed table-hover table-striped
bootgrid-table tablesorter" cellspacing="0">
   <thead>
      <tr>
         <th>Transaction ID</th>
         <th>Order ID</th>
         <th>User ID</th>
         <th>Item ID</th>
         <th style="text-align: center;">Update/Delete</th>
      </tr>
   </thead>
   <tbody>
      <?php include("tblorders.php");?>
   </tbody>
</table>
<script src="js/orders.js"></script>
```

```php
<?php
//include connection file
include_once("../db/connection.php");
$sql = "SELECT o.*, u.userAge, i.itemName FROM `tblorders` o LEFT JOIN `tbluser` u ON (o.userID
= u.userID) LEFT JOIN `tblitems` i ON (o.itemID = i.itemID) ORDER BY o.`orderID`";
$queryRecords = mysqli_query($conn, $sql) or die("error to fetch employees data");
?>
<h3>Order Report</h3>
<hr>
<table id="tblorders_grid" class="table table-condensed table-hover table-striped
bootgrid-table" cellspacing="0">
   <thead>
      <tr>
         <th>Transaction ID</th>
         <th>Order ID</th>
         <th>User ID</th>
         <th>User Age</th>
         <th>Item Name</th>
      </tr>
   </thead>
   <tbody>
      <?php $orderId = "first";
      foreach($queryRecords as $res) {
         if(($orderId != "first") && ($orderId != $res['orderID'])){
   print("<thead>
      <tr>
         <th>Transaction ID</th>
         <th>Order ID</th>
         <th>User ID</th>
         <th>User Age</th>
         <th>Item Name</th>
      </tr>
   </thead>
   <tbody>");
         }
         print("<tr><td >".$res['transactionID']. "</td>");
         print("<td >".$res['orderID']. "</td>");
         print("<td >".$res['userID']. "</td>");
         print("<td >".$res['userAge']. "</td>");
         print("<td >".$res['itemName']. "</td></tr>");

      if(($orderId != "first") && ($orderId != $res['orderID'])){
   print("</tbody>");
         }
    $orderId = $res['orderID'];

    }?>

</table>
```

```
<h3>Rules</h3>
<!-- <p>This is a page that will require some data mining rules.</p>
<p>Please use pages/rules.php file to edit this pages.</p> -->

<div class="row">
    <div class="col-md-4">
        <div class="input-group">
           <span class="input-group-addon">Support</span>
           <input id="support" type="text" class="form-control" name="support"
           placeholder="integer value < 100">
        </div>
    </div>
    <div class="col-md-4">
        <div class="input-group">
           <span class="input-group-addon">Confidence</span>
           <input id="confidence" type="text" class="form-control" name="confidence"
           placeholder="integer value < 100">
        </div>
    </div>
    <div class="col-md-4">
        <button type="button" class="btn btn-primary" id="get_rules">Get Rules</button>
    </div>
</div>
<hr>
<div class="rules" style="text-align: center; padding-top: 15px;"></div>
<script src="js/rules.js"></script>
```

```
<?php
//include connection file
include_once("../db/connection.php");
$sql = "SELECT * FROM `tbldepartment`";
$queryRecords = mysqli_query($conn, $sql) or die("error to fetch employees data");
?>


      <?php foreach($queryRecords as $res) :?>
      <tr dept_id="<?php echo $res['deptID'];?>">
         <td contenteditable="true" colname="deptID"><?php echo $res['deptID'];?></td>
         <td contenteditable="true" colname="deptName"><?php echo $res['deptName'];?></td>
         <td style="text-align: center;"><i class="fa fa-check-square fa-fw fa-2x"
         aria-hidden="true" onclick="updateDept(<?php echo $res['deptID'];?>)"></i> <i
         class="fa fa-close fa-fw fa-2x" aria-hidden="true" onclick="removeDept(<?php echo
         $res['deptID'];?>)"></i></td>
      </tr>
    <?php endforeach;?>
```

```php
<?php
//include connection file
include_once("../db/connection.php");
$sql = "SELECT * FROM `tblitems`";
$queryRecords = mysqli_query($conn, $sql) or die("error to fetch employees data");

$sql = "SELECT * FROM `tbldepartment`";
$departments = mysqli_query($conn, $sql) or die("error to fetch employees data");
?>


      <?php foreach($queryRecords as $res) :?>
      <tr item_id="<?php echo $res['itemID'];?>">
         <td colname="itemID" contenteditable="true"><?php echo $res['itemID'];?></td>
         <td colname="itemName" contenteditable="true"><?php echo $res['itemName'];?></td>
         <td colname="deptID"><select class="form-control">
            <?php foreach($departments as $dep) :?>
               <option <?php echo $dep['deptID'] == $res['deptID'] ? "selected" : "";?>
               value="<?php echo $dep['deptID'];?>"><?php echo $dep['deptName'];?></option>
            <?php endforeach;?>
         </select></td>
         <td style="text-align: center;"><i class="fa fa-check-square fa-fw fa-2x"
         aria-hidden="true" onclick="updateItem(<?php echo $res['itemID'];?>)"></i> <i
         class="fa fa-close fa-fw fa-2x" aria-hidden="true" onclick="removeItem(<?php echo
         $res['itemID'];?>)"></i></td>
      </tr>
    <?php endforeach;?>
```

```
<?php
//include connection file
include_once("../db/connection.php");
$sql = "SELECT * FROM `tbluser`";
$users = mysqli_query($conn, $sql) or die("error to fetch employees data");

$sql = "SELECT * FROM `tblorders`";
$queryRecords = mysqli_query($conn, $sql) or die("error to fetch employees data");

$sql = "SELECT * FROM `tblitems`";
$items = mysqli_query($conn, $sql) or die("error to fetch employees data");
?>

      <?php foreach($queryRecords as $res) :?>
      <tr order_id="<?php echo $res['transactionID'];?>">
         <td contenteditable="true" colname="transactionID"><?php echo
         $res['transactionID'];?></td>
         <td contenteditable="true" colname="orderID"><?php echo $res['orderID'];?></td>
         <td colname="userID"><select class="form-control">
            <?php foreach($users as $user) :?>
               <option <?php echo $user['userID'] == $res['userID'] ? "selected" : "";?>
               value="<?php echo $user['userID'];?>"><?php echo $user['userID'];?></option>
            <?php endforeach;?>
         </select></td>
         <td colname="itemID"><select class="form-control">
               <?php foreach($items as $item) :?>
                  <option <?php echo $item['itemID'] == $res['itemID'] ? "selected" : "";?>
                  value="<?php echo $item['itemID'];?>"><?php echo $item['itemName'];?></option>
               <?php endforeach;?>
         </select></td>
         <td style="text-align: center;"><i class="fa fa-check-square fa-fw fa-2x"
         aria-hidden="true" onclick="updateOrder(<?php echo
         $res['transactionID'];?>)"></i> <i class="fa fa-close fa-fw fa-2x"
         aria-hidden="true" onclick="removeOrder(<?php echo $res['transactionID'];?>)"></i></td>
      </tr>
    <?php endforeach;?>
```

```php
<?php
set_time_limit(400); //slow procedure

//include connection file
include_once("../db/connection.php");

$error = true;
$msg = '';

if(isset($_POST)){
    if(isset($_POST['support']) && !empty($_POST['support']) && ($_POST['support'] <= 100 )) {
      $support = (int)$_POST['support']/100;
      $error = false;
    } else {
      $error = true;
      $msg = " Incorrect parameter value!";
    }

    if(isset($_POST['confidence']) && !empty($_POST['confidence']) && ($_POST['confidence'] <=
    100) && !$error) {
      $confidence = (int)$_POST['confidence'];
      $error = false;
    } else {
      $error = true;
      $msg = " Incorrect parameter value!";
    }

   if(!$error) {
      $sql = "CALL usp_GenerateAprioriRules(".$support.", ".$confidence.")";
      // $result = mysqli_query($conn, $sql) or die('<div class="alert alert-warning">MySQL
      error to fetch rules data</div>');
      //$sql = "SELECT * FROM tblStrongRules";
      $rules = mysqli_query($conn, $sql) or die('<div class="alert alert-warning">MySQL error
      to fetch rules data</div>');

?>

<!-- <div class="alert alert-success"></div> -->

<table id="tblrules_sort" style="text-align: left;" class="table table-condensed table-hover
table-striped bootgrid-table tablesorter" cellspacing="0">
   <thead>
      <tr>
        <th>RuleId</th>
         <th>RuleXY</th>
         <th>RuleX</th>
         <th>RuleY</th>
         <th style="padding-right: 20px;">SupportXY</th>
         <th style="padding-right: 20px;">SupportX</th>
         <th style="padding-right: 20px;">Confidence</th>
      </tr>
   </thead>
   <tbody>

      <?php foreach($rules as $rule) :?>
      <tr>
         <td><?php echo $rule['RuleId'];?></td>
         <td><?php echo $rule['RuleXY'];?></td>
         <td><?php echo $rule['RuleX'];?></td>

         <td><?php echo $rule['RuleY'];?></td>
         <td><?php echo $rule['SupportXY'];?>%</td>
         <td><?php echo $rule['SupportX'];?>%</td>
         <td><?php echo $rule['Confidence'];?>%</td>
```

```
      </tr>
    <?php endforeach;?>
   </tbody>
</table>
<?php } else {
   echo '<div class="alert alert-warning">Error!' . $msg . '</div>';
}}?>
```

```
<?php
//include connection file
include_once("../db/connection.php");
$sql = "SELECT * FROM `tbluser`";
$queryRecords = mysqli_query($conn, $sql) or die("error to fetch employees data");
?>


      <?php foreach($queryRecords as $res) :?>
      <tr user_id="<?php echo $res['userID'];?>">
         <td colname="userID" contenteditable="true"><?php echo $res['userID'];?></td>
         <td colname="userSex"><select class="form-control">
            <option <?php echo $res['userSex'] == "F" ? "selected" : "";?> value="F">F</option>
            <option <?php echo $res['userSex'] == "M" ? "selected" : "";?> value="M">M</option>
         </select></td>
         <td class="editable-col" colname="userAge" contenteditable="true"><?php echo
         $res['userAge'];?></td>
         <td style="text-align: center;"><i class="fa fa-check-square fa-fw fa-2x"
         aria-hidden="true" onclick="updateUser(<?php echo $res['userID'];?>)"></i> <i
         class="fa fa-close fa-fw fa-2x" aria-hidden="true" onclick="removeUser(<?php echo
         $res['userID'];?>)"></i></td>
      </tr>
    <?php endforeach;?>
```

```php
<?php
//include connection file
include_once("../db/connection.php");
$sql = "SELECT * FROM `tblorders`";
$queryRecords = mysqli_query($conn, $sql) or die("error to fetch employees data");
?>
<h3>Transactions Report</h3>
<hr>
<table id="tblorders_sort" class="table table-condensed table-hover table-striped
bootgrid-table tablesorter" cellspacing="0">
   <thead>
      <tr>
         <th>Transaction ID</th>
         <th>Order ID</th>
         <th>User ID</th>
         <th>Item ID</th>
      </tr>
   </thead>

   <tbody>
      <?php foreach($queryRecords as $res) :?>
      <tr>
         <td ><?php echo $res['transactionID'];?></td>
         <td ><?php echo $res['orderID'];?></td>
         <td ><?php echo $res['userID'];?></td>
         <td ><?php echo $res['itemID'];?></td>
      </tr>
    <?php endforeach;?>
   </tbody>
</table>

<script src="js/transactions.js"></script>
```

```
<h3>Update Users</h3>
<hr>
<div id="notification"></div>
<h3>Add</h3>
<table class="table table-condensed table-hover table-striped bootgrid-table" cellspacing="0">
   <thead>
      <tr>
         <th>USERSEX</th>
         <th>USERAGE</th>
         <th style="text-align: center;">ADD</th>
      </tr>
   </thead>
   <tbody>
      <tr>
         <td ><select id="add_userSex" class="form-control">
            <option value="F">F</option>
            <option value="M">M</option>
         </select></td>
         <td ><input class="form-control" id="add_userAge" type="text" value=""></td>
         <td style="text-align: center;"><i class="fa fa-check-square fa-fw fa-2x"
         aria-hidden="true" id="add_user"></i></td>
      </tr>
   </tbody>
</table>
<hr>
<h3>Update</h3>

<table id="tblusers_sort" class="table table-condensed table-hover table-striped bootgrid-table
tablesorter" cellspacing="0">
   <thead>
      <tr>
         <th>USERID</th>
         <th>USERSEX</th>
         <th>USERAGE</th>
         <th style="text-align: center;">Update/Delete</th>
      </tr>
   </thead>
   <tbody>
      <?php include("tblusers.php");?>
   </tbody>
</table>

<script src="js/users.js"></script>
```