Web Analytics Data Warehouse for Information Systems

Application for Fatal Accident Reporting System

A Thesis Presented to

The Faculty of the Computer Science Program

California State University Channel Islands

In (Partial) Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Deepa Santhanam

December 2013

APPROVED FOR THE COMPUTER SCIENCE PROGRAM

12/16/2013

Advisor: Dr Richard Wasniowski Date

12/16/2013

Dr. Andrzej Bieszczad Date

12/16/2013

Dr Peter Smith Date

APPROVED FOR THE UNIVERSITY

12-16-13

Dr Gary A. Berg Date

Non-Exclusive Distribution License

In order for California State University Channel Islands (CSUCI) to reproduce, translate and distribute your submission worldwide through the CSUCI Institutional Repository, your agreement to the following terms is necessary. The author(s) retain any copyright currently on the item as well as the ability to submit the item to publishers or other repositories.

By signing and submitting this license, you (the author(s) or copyright owner) grants to CSUCI the nonexclusive right to reproduce, translate (as defined below), and/or distribute your submission (including the abstract) worldwide in print and electronic format and in any medium, including but not limited to audio or video.

You agree that CSUCI may, without changing the content, translate the submission to any medium or format for the purpose of preservation.

You also agree that CSUCI may keep more than one copy of this submission for purposes of security, backup and preservation.

You represent that the submission is your original work, and that you have the right to grant the rights contained in this license. You also represent that your submission does not, to the best of your knowledge, infringe upon anyone's copyright. You also represent and warrant that the submission contains no libelous or other unlawful matter and makes no improper invasion of the privacy of any other person.

If the submission contains material for which you do not hold copyright, you represent that you have obtained the unrestricted permission of the copyright owner to grant CSUCI the rights required by this license, and that such third party owned material is clearly identified and acknowledged within the text or content of the submission. You take full responsibility to obtain permission to use any material that is not your own. This permission must be granted to you before you sign this form.

IF THE SUBMISSION IS BASED UPON WORK THAT HAS BEEN SPONSORED OR SUPPORTED BY AN AGENCY OR ORGANIZATION OTHER THAN CSUCI, YOU REPRESENT THAT YOU HAVE FULFILLED ANY RIGHT OF REVIEW OR OTHER OBLIGATIONS REQUIRED BY SUCH CONTRACT OR AGREEMENT.

The CSUCI Institutional Repository will clearly identify your name(s) as the author(s) or owner(s) of the submission, and will not make any alteration, other than as allowed by this license, to your submission.

Web Analytics Data Warehouse for Information Systems

Title of Item

Information Systems, Data Warehousing, BI tools, Star schema, ETL

3 to 5 keywords or phrases to describe the item

DEEPA SANTHANAM

Author(s) Name (Print)

01. 14. 2014

Author(s) Signature Date

This is a permitted, modified version of the Non-exclusive Distribution
License from MIT Libraries and the University of Kansas

Web Analytics Data Warehouse for Information Systems

Application for Fatal Accident Reporting System

by

Deepa Santhanam

Computer Science Program

California State University Channel Islands

Abstract

Collecting, organizing, analyzing, and presenting data for consumption is well known for information systems. There are numerous commercial tools and vast sums of money dedicated to making this process efficient. Be it data warehousing, or map reduce framework, the core idea is to deduce useful information from disorganized and often disparate data. The objective of this thesis is to apply an organized approach to analyze public information systems so as to provide a simplified analytics interface that can help users and the public in general to make better use of the information. As one of exemplar of this approach, this thesis focuses on one specific source of publicly available data—data about fatal accidents. The methodology adopted is to apply a data warehousing approach to the fatal accident data, so as to transform the data into useful trend information. Another aspect of the methodology is to apply a commercially used approach, namely, business intelligence reporting, to serve public information in a simplified, easy to understand format. Thus, this thesis seeks to answer whether an industrial approach is applicable to public data. The work performed convincingly shows that indeed such an approach is useful. In fact, application of the data warehousing and business intelligence reporting approach results in a system which provides meaningful information of public importance. This thesis establishes the viability and simplicity of data warehousing and business intelligence reporting to help organize complex data and make them more accessible.

Acknowledgements

I would like to thank Dr Richard Wasniowski for his guidance and support to finish my thesis in time. I would also like to thank Dr Andrzej Bieszczad and Dr Peter Smith for evaluating my thesis. I also thank my husband, Srinivasan for his endless support and guidance that helped me complete my thesis. Finally, I would like to thank my mom and my two kids for supporting and encouraging me to do my best.

TABLE OF CONTENT

TABLE OF FIGURES

Chapter 1: Introduction

An information system is an integration of components for collecting, storing, and processing of data for delivering information. [1] There are numerous commercial tools and vast sums of money dedicated to development of efficient information systems. The objective of this thesis is to study the effectiveness of a data warehousing approach and leverage business intelligence reporting tools to process data in public information systems so as to provide a simplified analytics interface. The methodology adopted is to apply a data warehousing approach to fatal accident data in the National Highway Traffic Safety Administration (NHTSA) Fatality Analysis Reporting System (FARS), so as to transform the data into useful trend information.

The National Highway Traffic Safety Administration (NHTSA) maintains a Fatality Analysis Reporting System (FARS). According to the FARS website, "FARS is a nationwide census providing NHTSA, Congress and the American public yearly data regarding fatal injuries suffered in motor vehicle traffic crashes." [2 The FARS database contains data on every single accident on a public roadway in the nation that involves a fatality. The associated FARS website provides both raw data related to the fatalities and an interface to write advanced queries. [3, 4]

However, the interface is complex and the reporting features limited. Generally speaking, the system appears to be designed for advanced users and researchers. Moreover, the nature of the data is such that it has evolved in complex ways over time. So much so, the FARS web site does not allow for queries spanning multiple years: "Due to the complexities within the FARS data, users cannot query across multiple years." [5] Instead, the web site provides a limited number of "trend reports" that identify trends across years. For example, the "Occupants" trends report presents details of age distribution of the occupants in the fatal crashes over different years. Further, the report can be drilled down by state. [6]

With respect to the query interface, because there is no limitation to types of research queries that may be posed by researchers, the interface is generic and provides greater flexibility in the types of queries. The trade-off is in terms of user friendliness and support for data visualization. For instance, the trends report on the FARS web site is presented as a table of numbers. This presents a complicated structure for the user to comprehend.

The general features, and the complexities in the FARS data and interface is typical in a complex information system. The FARS data therefore provides a good case study to address the objective of this thesis, namely, to study the effectiveness of a data warehousing approach to organize and present meaningful information from a public information system.

1.1 Objective

The inherent complexity of the underlying data is a motivation for this thesis. In this thesis, we implement a web based data analytics systems for the FARS data that is focused on the layperson audience who may be interested in simple questions. This is a classic data analytics and business intelligence issue where the "view" is fixed at design time to answer specific questions. The objective is to apply this approach to develop a simple, easy-to-use website to obtain fatality information. Such a website will make this valuable information available to a broad audience and empower them.

The viability of simplifying the complex, multi-dimensional FARS information rests on data warehouse technology. A data warehouse is a repository that consolidates data from one or more sources of information, often in different formats. It is generally used for data analysis and reporting. Moreover, the information from the data warehouse can be presented in various forms like reports, graphs etc. Data warehouses typically reside on dedicated servers and run on a database management system such as Oracle, Microsoft etc. Having a data warehouse model in place, gives the user an opportunity to discover variety of information, helps them in making decisions faster by providing access to data derived from multiple data sources

1.2 Architecture of a Data Warehousing system

An illustration of the basic architecture of a data warehousing system is shown in Figure 1.

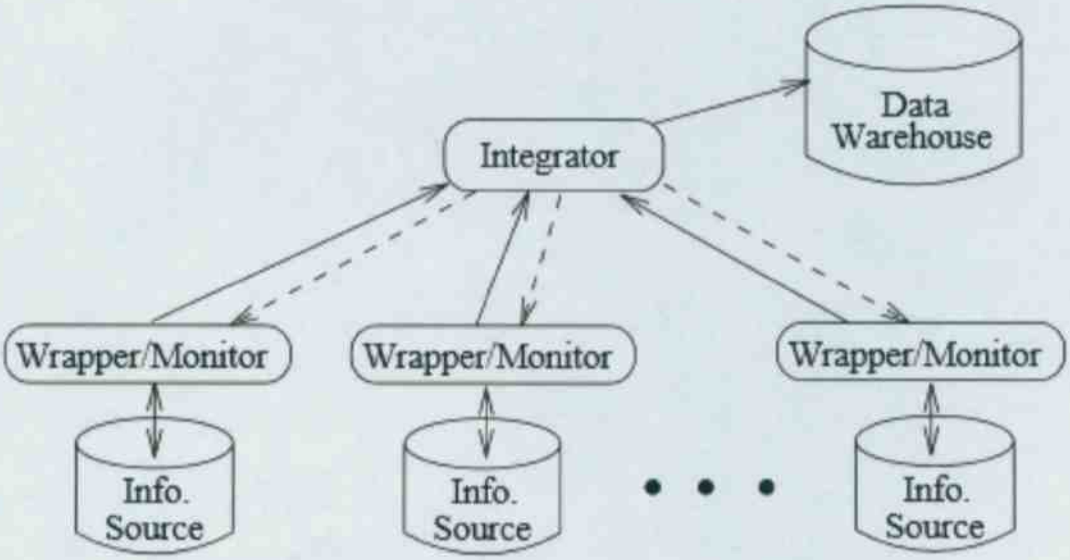


Figure 1. An illustration of a data warehousing system. [7]

Figure 1 includes information sources, wrapper/monitors and an integrator. Information sources could be anything like a HTML document, flat files, newswires, RDBMS etc. Connecting each information source is the Wrapper/Monitor. Conceptually, there is a wrapper/monitor associated with each information source. The wrapper helps in gathering data from the associated information sources, performs data cleansing and translation of the available information from a native format to a generic format required by the data warehouse. The monitor helps in periodically updating the information in the warehouse. The integrator filters, summarizes and merges the information from the wrapper and helps installing this information into the data warehouse. The information in the data warehouse conforms to that of a data model.

For more complex applications, a combination of a data warehouse and a set of data marts can be used. A data mart is a subset of a data warehouse dedicated to a specific part of a business or organization. A data mart can be created by manipulating and extracting data from the data warehouse layer and placing it in separate tables or by specifying database views based on the specific tables in the data warehouse layer. [9] Figure 2 depicts a data warehouse and data marts architecture.

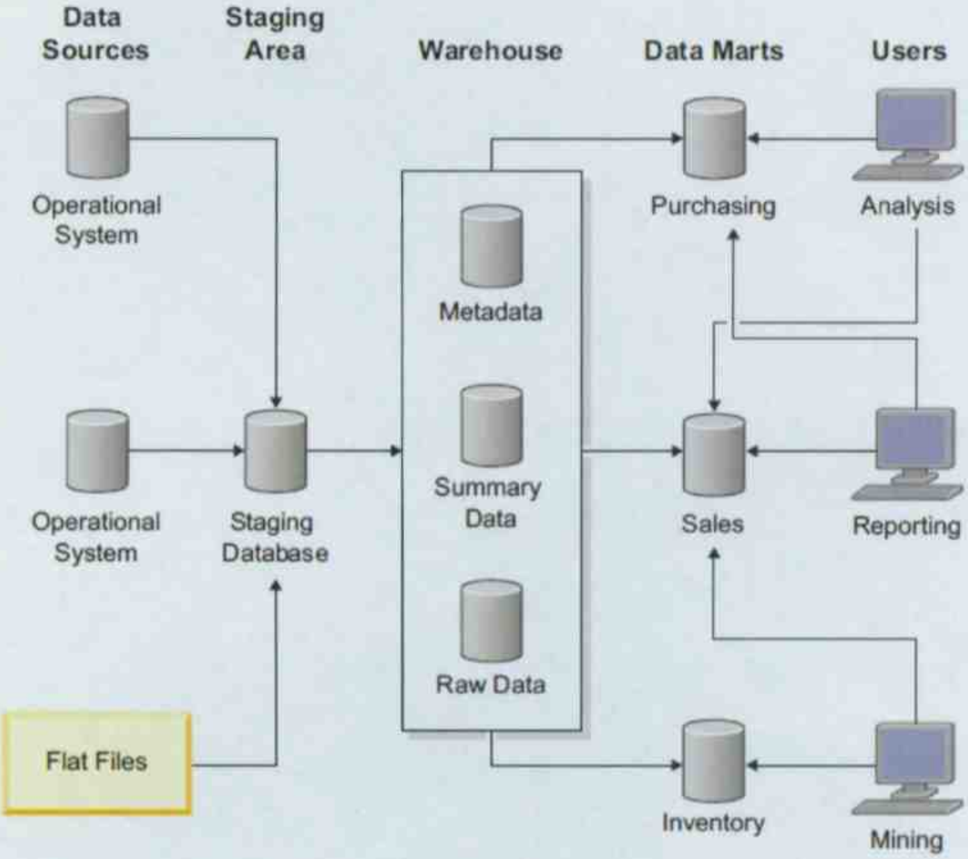


Figure 2. An example of a practical implementation of a data warehouse that includes data marts in addition to data warehouse. [9]

1.3 Data Modeling

To create a data warehouse, the raw data has to be logically analyzed. In order to do so, data models needs to be constructed. "A data model defines the structure and meaning of data." [10] Constructing a data model helps the user to understand the business needs and requirements and provides a foundation for implementing the database. Data modeling involves identifying the entities in the system and their relationships. Often, E-R diagrams are constructed to visually represent the entities and relationships. In data warehousing, the entity relationships often take the form of a star schema. Another example of a schema used in data warehousing is the snow flake schema. [11]

Once data modeling is accomplished, the schema can be implemented as a database design, and the database can be populated using a process called Extraction, Transform and Load, also called ETL. Details of the ETL process are described in Chapter Chapter 2:.

1.4 Related Work

The field of information systems combines research work from a number of areas in computer science, primarily from database systems, and more broadly from distributed systems, web architectures, and networking.

Fundamentally the term "information systems" refers to an integrated set of components related to collection, storage, and processing of data for the purpose data delivery and presentation. [1] One of the early formulations of data warehouses is by Widom. [7] Widom's architecture of a data warehouse, illustrated in Figure 1 above, consists of a variety of information source feeding information through a wrapper, and an integrator acquiring this information for organization and storage in the data warehouse. Widom presents an overview of research problems in the area of data warehousing, which is one particular way of implementing an information system. Sen and Sinha present an overview of data warehousing and a comparison of different warehousing methodologies. [8]

Luo [20, 21] describe work related to improving the operation of the databases in data warehouse systems. Efficiencies are achieved, for example, by reordering transactions so as to take advantage of data already fetched into memory, and by "pre-aggregating" to reduce the number of certain type of SQL statements related to load transactions. Luo's work focuses on real time data warehousing involving continuous updates to the warehouse and multiple simultaneous transactions.

An alternative vision of collecting, storing, and presenting information is represented by a search engine such as Google. This type of data analysis is often called "big data." Big data is a technology that uses massive amount of information being collected through various sources on the internet, and performs analysis to draw conclusions from it. Google implements a massive data collection enterprise, and organizes the data in readily accessible form so as to respond to web queries. Likewise, the presentation of the data is also customized for a web searching context. Thus, a search engine implementation is also an information system. In order to deal with enormous amounts of data and to support their storage needs, Google created two technologies named "Google File System" (GFS) and "Map Reduce". Google also developed a distributed storage system named "Bigtable" for managing structured data. [22, 23, 24]

Liu et al. describes the use of Map Reduce framework to perform the ETL process of loading data into a data warehouse. [25, 25] Thusoo et al. also describes a Hadoop based implementation of ETL processes at Facebook. [27, 28] Hadoop is an implementation of the Map Reduce framework. [29]

With respect to organization of data within a data warehouse, a number of researchers and practitioners have investigated data models such as star schema and snow flake. [10, 11] Generally speaking, a star schema has more data redundancy than a snow flake schema. The cost of greater redundancy provides better query time performance because fewer database join operations are needed. [10]

Another area of focus has been effective query and presentation tools. XML and related web technologies have been an important area of work in this regard. XML technologies, in particular, are versatile enough to be deployed not only at the front end, but also for source data integration, and data storage. [12] Business intelligence reporting tools have made it easy to create effective presentation interfaces for complex information that has been aggregated and organized in data warehouses and data marts. Examples of business intelligence reporting tools include BIRT [13], Spago [14], Pentaho [15], Jaspersoft [16].

1.5 Summary

As discussed, the main objective of this thesis is to apply a data warehousing approach to develop an easy-to-query website for obtaining trend information from public data. The NHTSA FARS data is a case study exemplar to apply this approach. Using the NHTSA FARS as an example, we present a simplified approach to design and develop a data warehouse starting from a set of user queries. We implement a web interface to the data warehouse using a business intelligence reporting tool.

In Chapter 2, we provide an overview of star schema and the process of extraction transformation, and load to populate and update a data warehouse.

In Chapter 3, we discuss about the structure and evolution of FARS data. The Fatality Analysis Reporting System has data on the fatal crashes within 50 States, the District Columbia and Puerto Rico. The FARS database is designed and developed by NHTSA's National Center for Statistics and Analysis. It has data collected from 1975 till present day. The user manual for FARS describes 17 data files as of 2011. Some of the data files include Accident, Vehicle, Parkwork, Person, Distract, Factor, and Vision. Of these, Vehicle, Person and Accident tables were used in this thesis to answer questions related to fatality crashes.

In chapter 4, we discuss about a number of questions related to FARS data for which users would like to find answers. A simple web-based interface is created for the user to enter the required parameters for the query for which they seek answers from the data warehouse model.

We then discuss how this information is being used to find some interesting statistics about the trends in fatality crashes.

In chapter 5, we discuss about the tools, platform choices and database that is used to implement the data warehouse model. We then discuss the extract, transform and load process for loading the data warehouse. In the extract process, data is extracted from FARS raw data. Although raw data is available in multiple formats like SAS, DBF and SEQL, the DBF format is chosen. It is then converted to CSV format. In the transform phase, the extracted data is converted to a form which supports the user requirements. In load phase, we discuss about how the data is written to the database.

In Chapter 6, we will discuss some of the interesting conclusions drawn using the data warehouse and reporting tools about accident fatalities that happen in the United States.

1.6 Key Terms

Information systems, Data warehousing, Business intelligence tools, Star schema, Extraction Transformation Load (ETL).

Chapter 2: Overview of Star Schema and Extraction, Transformation and Load (ETL)

2.1 Star schema

A star schema is a dimensional design for a relational database, and is widely used in data warehouses where data is stored in a manner that is easier to retrieve information from the database. A star schema comprises of two major components—dimension tables and fact tables. Dimensions are stored in dimension table and Facts are stored in the facts table. The fact tables include foreign keys representing specific values for the dimensions. The entity relationship diagram resembles a star in which the center of the star schema consists of one or more fact tables and the points of a star have the dimension tables. Figure 3 below illustrates a star schema.

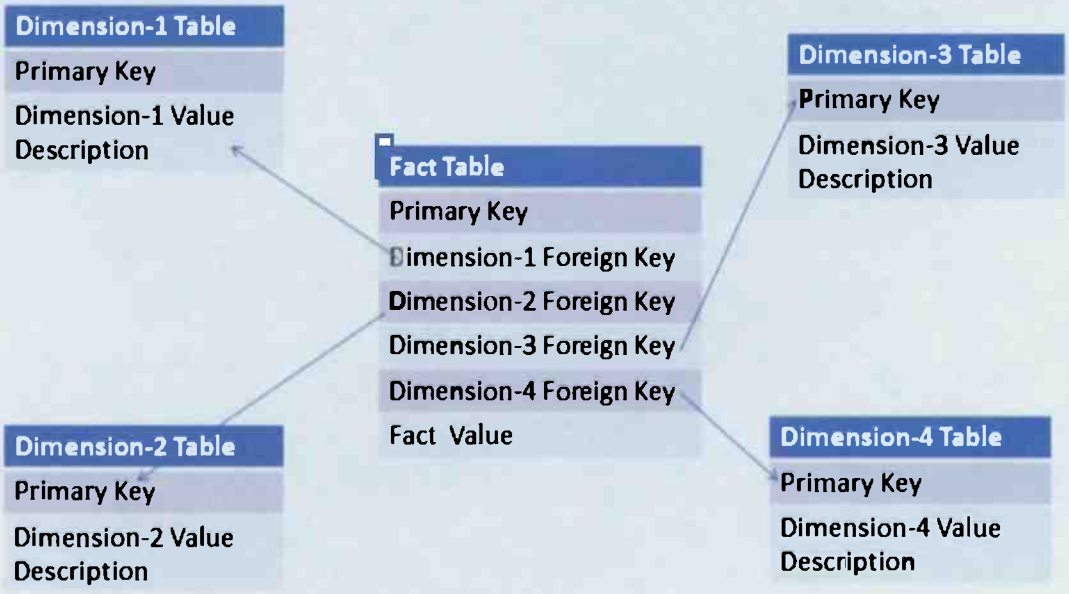


Figure 3. An illustration of star schema.

A fact table contains data corresponding to a particular business such as sales or profit. Fact tables generally have numeric data that are calculated during the transformation step of the ETL process and stored in the database during the loading step. The rows in a fact table may store an aggregate value (*e.g.*, "total" sales) and are called as aggregate fact table or summary table. Within each row of a fact table, the set of dimensional values for which the fact value is calculated are represented in the form of foreign keys that are linked to specific rows in the dimension tables. The dimension tables store descriptions of the characteristics of a business. Contrary to fact table, dimension tables generally contain descriptive textual values for the dimensional value. Several distinct dimensions can be combined with facts to produce some meaningful information. As one example, the total sales in each month of a year are facts stored in a fact table. The textual month descriptions (*i.e.*, "January," "February," *etc*) arc dimension values stored in a month dimension table. Each row in the fact table will include a foreign key referencing a row in the month dimension table for which the sales fact has been calculated.

A more complex example is illustrated in Figure 4, in the form of a sales fact table with product, period and store as dimensions. Using such a schema, one can query sales facts filtered by product, period and store, implemented in the form of a join query that combines information from the fact table and the dimension tables.

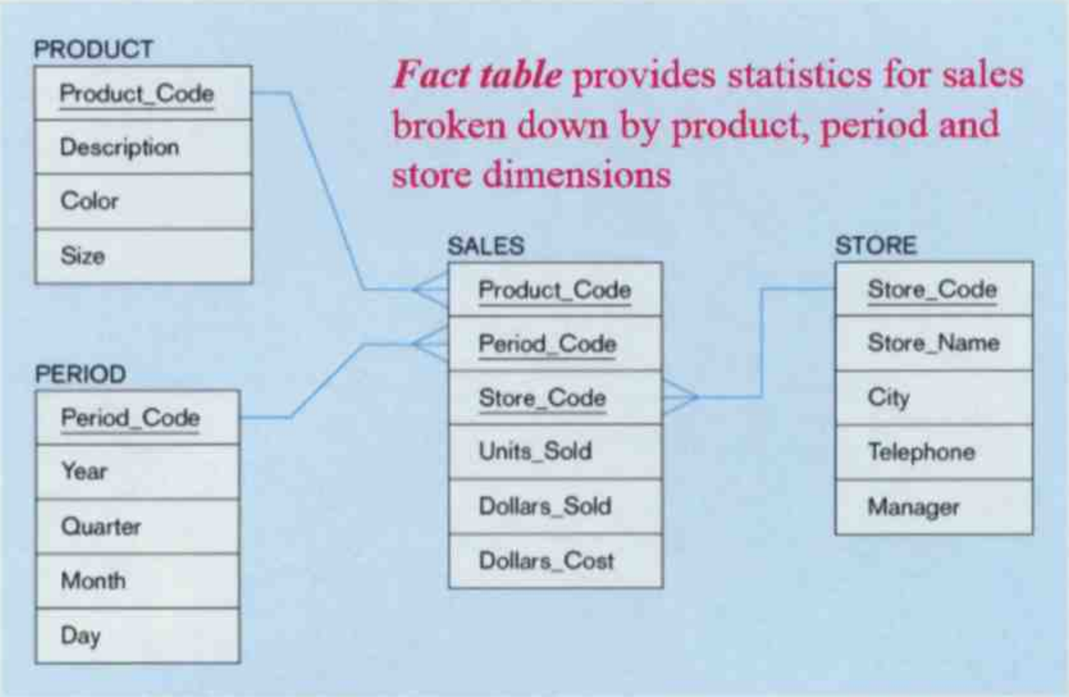


Figure 4. A star schema example organizing sales data by product, period, and store. [30]

"Sales" is the fact table connected to various dimensions like Product, Period and Store. In the Product dimension table, the primary key is the "Product_Code". The product code in the sales fact table is a foreign key drawn from one of the primary keys in the product dimension table. The "Sales" fact table contains Product_Code, Period_Code, Store_Code as the detail level facts and Units_Sold, Dollars_Sold and Dollars_Cost as the aggregate facts. The above example shows that the data can be retrieved and aggregated across multiple dimensions. For example, in the Units_Sold field in the fact table, "Sales" can either be combined for a single dimension independently to find out the Units_Sold during a particular period or Units_Sold for a particular store or to find the Units_Sold for a particular product or can be calculated across all the dimensions as the total sales across all products, units and periods.

2.2 Extraction Transformation Load (ETL)

Loading data into data warehouse is one of the critical processes in business analysis. Data in the data warehouse gets updated periodically (monthly, daily or yearly basis). Hence, ETL is not a onetime process, but an ongoing part of the data warehouse. In order to load the data from the source system to the data warehouse, three important operations are to be performed: Extraction, Transformation and Loading. These operations separate the "raw" data from the information sources from the "transformed" information in the data warehouse.

2.3 Extract

Extraction process is about getting the information from the information source (e.g., source database) and making it accessible for further processing. In simple words, extraction process is about copying the data from the source destination to the target destination. Data is extracted from information sources can be in various formats: text files, spreadsheets, relational and non-relational databases. These extractions can be as partial or extracted as a whole. Partial extraction is when a portion of the data has changed over a period of time. Full extraction is where the entire data is extracted from the information source.

2.4 Transform

This process is about transforming or modifying the data to suit the business needs. Before the data it is loaded into the data warehouse database, the extracted data is modified in order and further processed (e.g., aggregated) to conform to the data warehouse schema. But there can be some data which does not require or requires very little transformation. Some of the typical modifications can include: sorting, cleaning, filtering, aggregating, *etc.*

2.5 Load

This is a data storage phase. This process is about loading the data into the appropriate table in the target database. Depending on the extraction technique, the loading phase can be either loading the entire data into the database or overwriting on the existing data.

2.6 ETL Tools

ETL implementation can either be done using ETL tools like (Oracle Warehouse Builder, BusinessObjects Data Integrator) or can be hand coded. ETL tools can be fairly expensive and hence this could be one of the reasons why ETL implementations can be hand coded by developers. An additional advantage of hand-coding is that ETL operations can be customized. A significant disadvantage of hand coding is that it can be a complex time consuming exercise.

Chapter 3: Structure and Evolution of FARS Data

3.1 FARS

According to National Highway Traffic Safety Administration (NHTSA), "Fatality Analysis Reporting System (FARS) is a nationwide census providing NHTSA, Federal agencies, State and local governments, Congress and the American public yearly data regarding fatal injuries suffered in motor vehicle traffic crashes." [1] NHTSA presents statistics about the accidents that happen in a public roadway that results in loss of human life from its primary data system, FARS. According to NHTSA, "FARS was created in the United States to provide an overall measure of highway safety, to help suggest solutions and to help provide an objective basis to evaluate the effectiveness of motor vehicle safety and highway safety programs." [17]

FARS was designed and developed by National Center for Statistics and Analysis (NCSA) of NHTSA in 1975. FARS data has information on fatal crashes of the 50 states of USA, the District of Columbia and Puerto Rico. In order for the data to be included in FARS database, a crash must involve a motor vehicle on a traffic way that is open to public and which results in loss of a person's life within 30 days of the crash. FARS data is available for every year since 1975. Although FARS has public information about the fatal crashes and types of vehicle that are involved in the crash, it does not include any personal information like names, address, SSN.

These are valuable information that have been collected over the years and are available for research, business needs or personal use. The FARS data can be downloaded from http://www.nhtsa.gov/FARS . This website also provides access to a query interface for answering questions about fatalities in any given year. The entire data is available for download in DBF, SAS and Seql formats. Once the data is downloaded, users can process the information according to their specific needs.

3.2 How is the data collected?

NHTSA and NCSA have an agreement with various States to provide information on fatal crashes. Types of information include Death Certificates, Police Accident Reports, State Vehicle Registration Files, State Driver Licensing Files, Hospital medical reports, Emergency Medical Service Reports, etc. Once the fatal crash information is available, analysts load the related data into a local computer and transmit the data to NHTSA's central FARS computer on a daily basis. Consistency checks are done to ensure that the data are consistent. The data is then loaded appropriately into different data files. As of 2011, there are 17 data files that are available. Some of the data files include Accident, Person, Vehicle, Parkwork, Person, Maneuver, and Vision. [18]

3.3 Data Files

There are a lot of changes that have been made over the years to the data collected and the way the data is presented in the data files. For example, some elements have been dropped and new ones have been added. Additionally, whereas the data was organized as a set of 3 files in 1975, presently the data are organized into 17 files. An important aspect of a data warehouse model design is to select what to keep and what to discard. For the purposes of this thesis, we focused on the specific set of user queries described in Chapter **Error! Reference source not**

found.. As a result, data from only three of the files was used. This allowed for analysis through the entire period for which FARS data is available. The following are the data files that were processed for populating the data warehouse.

3.3.1 Accident data file

An Accident file is available for each of the years from 1975 to current. The Accident file has information about crash details like: time of the accident, day of the week, number of fatal, drunk drivers involved in the accident and the manner of collision of the vehicle. It also has information about the environmental conditions at the time of the crash, when, where and at what time of the day did the crash happen. The Accident file includes one record per crash.

3.3.2 Vehicle data file

A Vehicle file is available for each of the years from 1975 to current. This file contains information related to type of vehicle like: make, model, VIN number, model year and also has information on the damage of vehicles. The Vehicle file includes one record per in-transport motor vehicle involved in a fatal crash.

3.3.3 Person data file

A Person file is available for each of the years from 1975 to current. This file contains information related to motorists and non-motorists involved in crash. The data also includes person related information such as age, gender and also other information related to injury severity. Person file includes one record per person. Notably, this file also includes information on persons who did not die within the 30 day period after the crash.

Chapter 4: User Questions on FARS Data

In order to construct a data warehouse model, we need to identify and collect the requirements of the problem the data warehouse is attempting to address. Since this thesis focuses on answering queries related to fatality information, a list of potential user queries was collected. The following questions were identified as the possible user queries that the data warehouse is designed to address.

1. Find the total number of accidents that resulted in fatalities between any two years.
2. Find the total number of accidents that resulted in fatalities for every month between any two years.
3. Find the total number of accidents that resulted in fatalities by day of the week between any two years.
4. Find the total number of accidents that resulted in fatalities for all the states between any two years and find out which state had maximum number of fatalities.
5. Find the total number of accidents that resulted in fatalities based on the weather conditions (Rain, snow, fog, smoke, sand, dust) starting between any two years.
6. Find the total number of accidents that resulted in fatalities based on the manner of collision between any two years.
7. Find out which age group (15-20, 21-30, 31-40, 41-50, 51-60, 61-70, 71-80, 81-90) had more number of fatalities between any two years.

8. Find the number of accidents that resulted in fatalities gender wise, between any two years.
9. Find the total accidents between any two years that resulted in fatalities crashes related to drivers consuming alcohol.
10. Find out which make/model of the car had many number of fatality crashes between any two years?
11. For a given year, find the number of fatalities that occurred for State 1 and State 2 and find out which state had more number of fatalities.
12. Find the total number of accidents for years between any two years.
13. Find which month had more number of accidents between any two years.

Based on the above possible user queries, a data warehouse model was designed as described in Chapter **Error! Reference source not found.** to addresses the questions above, and get meaningful trend formation about crashes involving fatalities.

5.Design and Implementation

A number of platform and design choices were made in implementing the data analytics system. Platform level decisions include choice of database system, web-server, and user interface technology. Design level decisions include choices related to database schema, the procedure for extracting, transforming, and loading data, and the presentation of the user interface.

4.1 Platform Choices

Since the ultimate objective is to develop a system where users can access useful information related to fatalities, deciding to implement a system with a web-interface was easy. Additionally, the availability of a number of free and open source web technologies made the decision practical. MySQL is also well known as web-development database. Web servers such as Apache and Apache TomCat are similarly well known. In fact, LAMP and WAMP are well known web development platforms. LAMP is an abbreviation of Linux-Apache-MySQL-Perl/PHP and WAMP an abbreviation of Windows-Apache-MySQL-Perl/PHP. As an easy starting point, WAMP was chosen. However, as discussed below, this decision was not appropriate because of platform dependencies of the user-interface technology selected.

With respect to the user-interface technology, a primary objective of the data analytics system is to provide a user friendly interface that is easy to access and understand. Therefore, the ability to chart or graph a trend is an important consideration. Additionally, data analytics on the FARS data is conceptually similar to data analytics on business data. A number of business intelligence tools have been implemented. These include open source free products like BIRT, SpagoBI, etc , and open source commercial products like Jaspersoft, Pentaho etc. Based on the rich set of tools available, a decision was made to utilize a business intelligence reporting component as part of the user interface.

Reporting tools helps users organize their data visually for their business needs. Business users can use these tools to view, run and save reports as and when they require. For the data analytics system described in this thesis, we need a reporting tool which is web based, easy to use, able to print reports in various formats, and providing rich variety of reporting features. BIRT (Business Intelligence and Reporting Tools) is one such open source reporting system developed as part of Eclipse can be used for Java J2EE based web applications. [13] Hence, BIRT was chosen for the business intelligence reporting component of the thesis.

Once BIRT was chosen, it became clear that TomCat or similar server was required to be installed. Therefore, a WAMP stack no longer made sense. Using PHP requires a Java Bridge to be installed. Since there is no pre-existing PHP codebase to support, the user interface could be designed using JSP technology on Apache TomCat.

The FARS raw data is available in multiple formats: SAS, DBF, and SEQL. Free tools exist to convert DBF files to comma-separated-value (CSV) files. [19] Therefore DBF format raw data was used. The next challenge was to identify appropriate tools and platforms to read the raw data and load it in the database. As discussed in Chapter 2, the structure of the data is fairly complex with a number of variations over time. Since a data warehousing approach makes most sense to provide simple user queries, the raw data has to be extracted and transformed, prior to loading in the database. A number of tools exist to read CSV data and load them in a MySQL database[1]. However, these are not useful in the present context because they assume the columns in the CSV file correspond to the database table schema. On the other hand, for a data warehouse, fields from multiple input records are aggregated and transformed to fields of database tables in the data warehouse. Though a number of ETL tools exist, it made more sense to write a custom tool to process the FARS data. Java packages to process CSV files, and support for MySQL are freely available. Therefore, a decision was made to write a Java based tool for the ETL process.

To summarize, the following platform choices were made:

1. TomCat web server, with JSP pages for user interface
2. BIRT reporting tool
3. MySQL database
4. Custom Java software for ETL

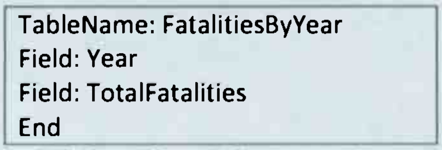
4.2 Database Design and ETL

The database design and the ETL software design was an iterative process. In particular, a number of missteps occurred in the early phases in designing the data warehouse. A particular problem encountered was in determining what data to include in the data warehouse and what to exclude. This is a consequential decision because once this determination is made, it is difficult if not impossible to recover data that has been discarded. To grapple with the inherent tension of simplifying the database while at the same time preserving as much detail as possible, the initial database schema iterations focused on maintaining identities of individual accidents, while dropping accident related fields that were deemed less important or unrelated to the user queries defined in Chapter 4 above.

[1] See, *e.g.*, http://www.convertcsvtomysql.com/

Though there is no formal rule precluding such a data warehouse design, it shared fewer similarities with a conventional data warehouse that usually relies on aggregating data so as to reduce the computational effort in serving rich reporting data. Over time, it became clear that a star schema with aggregated fields was simple to implement, and effectively tracked the user queries defined in Chapter 4. In fact, each of the user queries directly translates to a particular organization of the database schema. This approach logically leads to a star schema or something very similar.

As one example, consider the user query, "How many fatalities have occurred between years A and B?" This suggests that there must be a count of number of fatalities for each possible time period. A simple solution is to compute the total number of fatalities in each year, and maintain that in the data warehouse. This is reflected in the following simplified schema.



```
TableName: FatalitiesByYear
Field: Year
Field: TotalFatalities
End
```

Figure 5. A simplified schema for fact tables in the fatality information warehouse.

Using this general approach, each user query was translated into a table. These correspond to a fact table in the terminology of star schema. Supporting data are stored in a dimension table. For instance, months are numbered 1 to 12 and stored as part of month-related accident facts. That the number 1 corresponds to January, 2 to February, and so on, is stored in a month dimension table. The entity relationship between the monthly fatalities fact table and the month dimension table is illustrated in Figure 6.

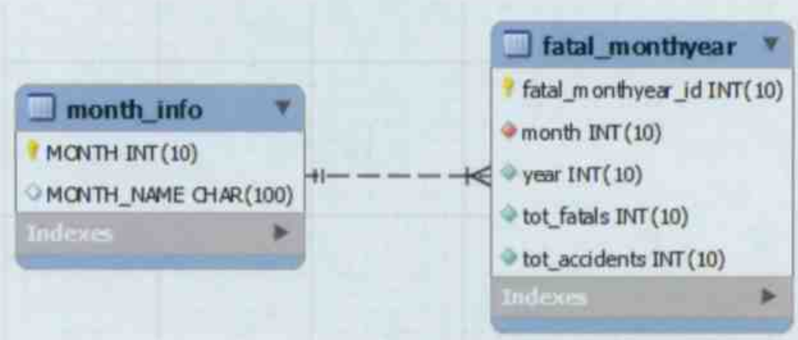


Figure 6. Entity relationships between fact table named fatal_monthyear and dimension table month_info.

Having decided on this approach to design the database, the exercise of defining the tables became a mechanical process of examining the user query and identifying the associated fact and dimension tables. Once design choice that was made intentionally is to include a year field in every fact table. The purpose was to allow a user to obtain answers for a query for a given year or over multiple years. This provides a level of flexibility that is absent in the present FARS query interface available at http://www-fars.nhtsa.dot.gov/QueryTool/QuerySection/SelectYear.aspx which does not allow the user to query across multiple years. At the same time, because this is a data warehouse, a number of complexities in the structure are discarded and only limited data is made available. Some examples of fact and dimension tables in the data warehouse are illustrated in Figure 7, Figure 8, and Figure 10.

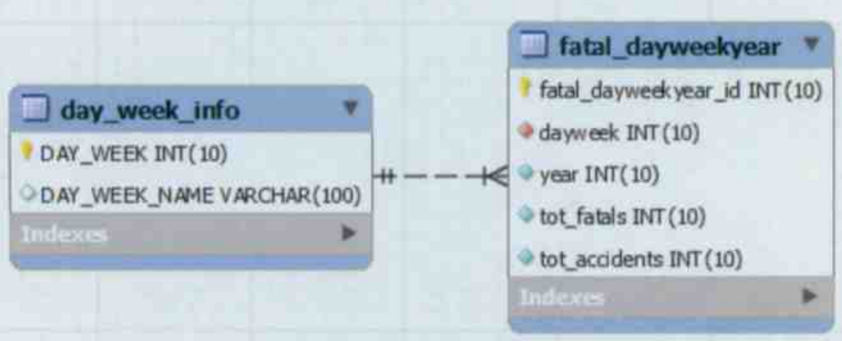


Figure 7. Entity relationships between fact table named fatal_dayweekyear and dimension table day_week_info.

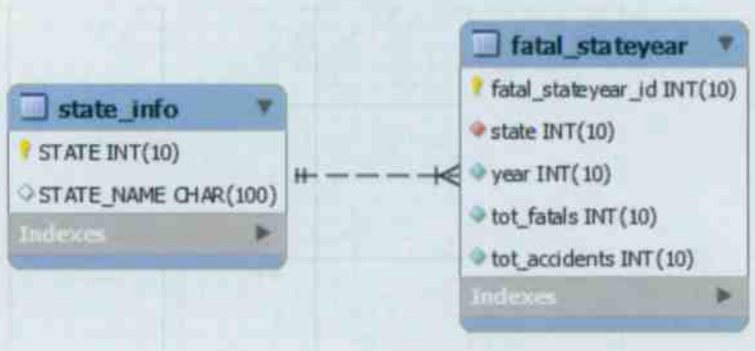


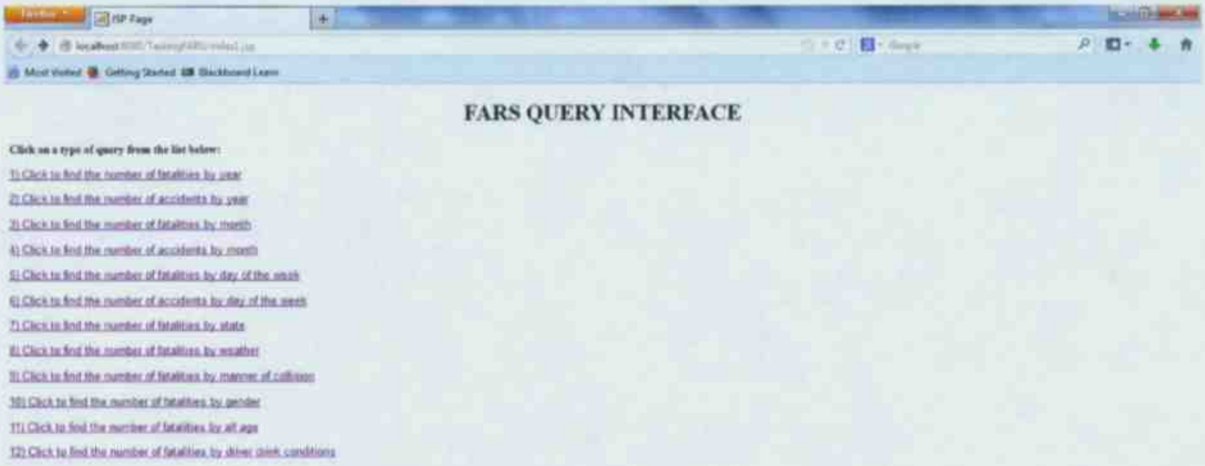
Figure 8. Entity relationships between fact table named fatal_stateyear and dimension table state_info.

One of the dimension tables that proved most complex to handle was the make and model information of cars. There are simply too many combinations of makes and models available to manually populate the make and model information in corresponding dimension tables. Additionally, because the aggregation operations for transforming the data requires a bucket for each valid row in the dimension table, writing code to create as many buckets to aggregate the data was not easy, especially because it is not even easy to count the number of valid make and model combinations manually. To deal with this, a semi-automated approach was adopted. The raw FARS data was first processed to identify all valid combinations of makes and model numbers and stored in the dimension table of the database. The rows in the dimension table were then output to a file. The fields in this file were then processed using regular expressions to convert it to a Java code segment that could be used during ETL. More particularly, the particular make and model combinations were manually populated into Java code to construct a hash table. When processing the FARS data, if a particular make-model combination was seen, the corresponding hash table entry was used to identify the particular bucket to increment. Even with these semi-automated procedures, the process of entering model names manually into the dimension table could not be avoided.

Chapter 5: Results

Using the approach discussed above, a data warehouse was implemented. Additionally, the BIRT reporting tool was used to develop parameterized reports associated with each of the user queries identified in Chapter 4. Java Server Page scripts were written to request the user to provide inputs, such as the specific query to run, and to provide parameters for the query. Java Server Page scripts to invoke the BIRT runtime environment to generate the associated report and to embed the generated BIRT report in an output web page were also developed. As discussed in Chapter 5, the JSP scripts are executed by Apache TomCat server.

Using the developed query interface, a number of queries were run, and interesting results obtained. Figure 9 shows the screenshot of the query interface.



FARS QUERY INTERFACE

Click on a type of query from the list below:

1) Click to find the number of fatalities by year

2) Click to find the number of accidents by year

3) Click to find the number of fatalities by month

4) Click to find the number of accidents by month

5) Click to find the number of fatalities by day of the week

6) Click to find the number of accidents by day of the week

7) Click to find the number of fatalities by state

8) Click to find the number of fatalities by weather

9) Click to find the number of fatalities by manner of collision

10) Click to find the number of fatalities by gender

11) Click to find the number of fatalities by all age

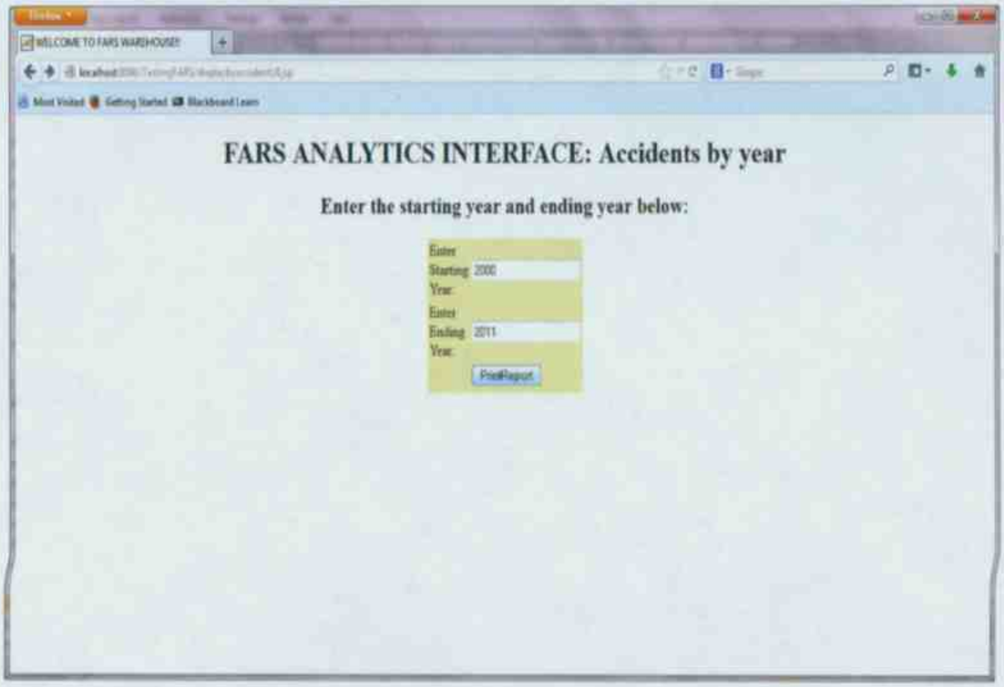
12) Click to find the number of fatalities by driver drink conditions

Figure 9. Screen shot of query interface

The following are some interesting conclusions drawn using the data warehouse and reporting tools.

5.1 Accident fatalities have decreased since 2005

Figure 10 and Figure 11 show screenshots of the user query interface to view the number of accidents and number of fatalities between two years. Figure 12 and Figure 13 show the responses generated by the BIRT reporting tool upon querying the data warehouse. The interesting trend observable in the result screenshots is that the number of accidents involving fatalities, and the number of fatalities, both, have declined since 2005. Moreover, the declines have been substantial since 2007.



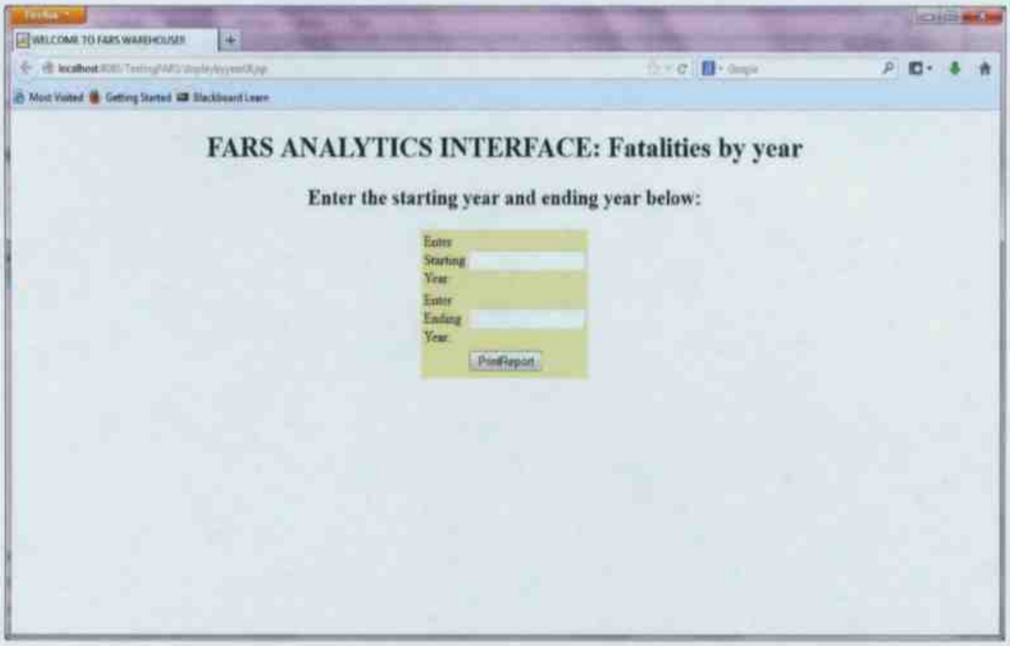
FARS ANALYTICS INTERFACE: Accidents by year

Enter the starting year and ending year below:

Enter Starting Year: 2000

Enter Ending Year: 2011

Figure 10. Query interface for obtaining trend information on accidents involving fatalities by year.



FARS ANALYTICS INTERFACE: Fatalities by year

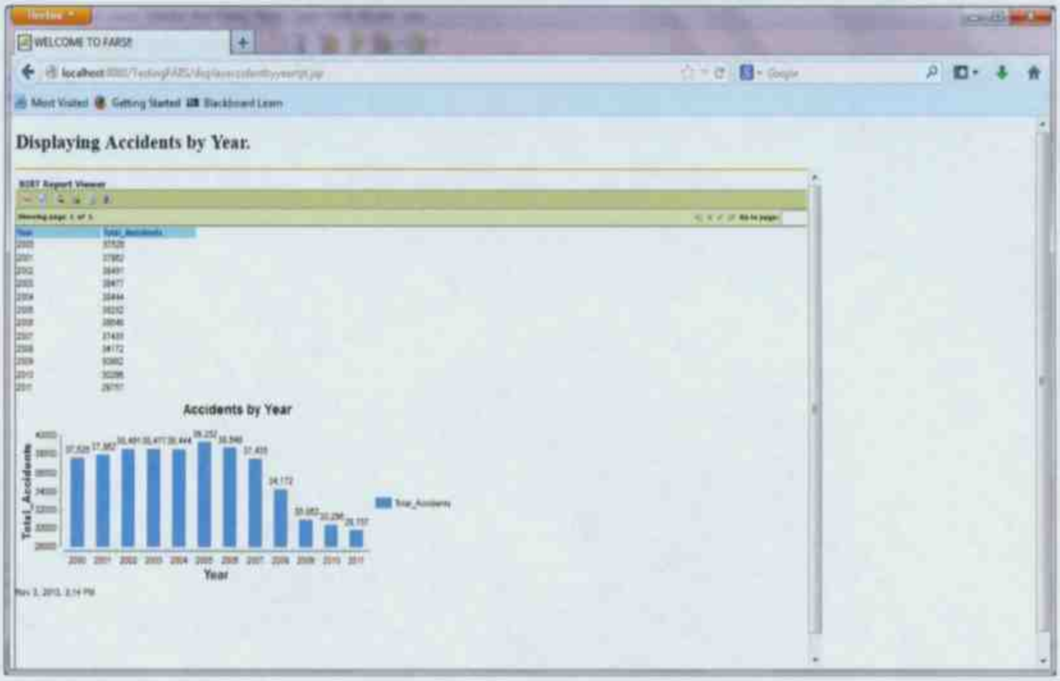
Enter the starting year and ending year below:

Enter Starting Year

Enter Ending Year

PrintReport

Figure 11. Query interface for obtaining trend information on fatalities in vehicle accidents by year.

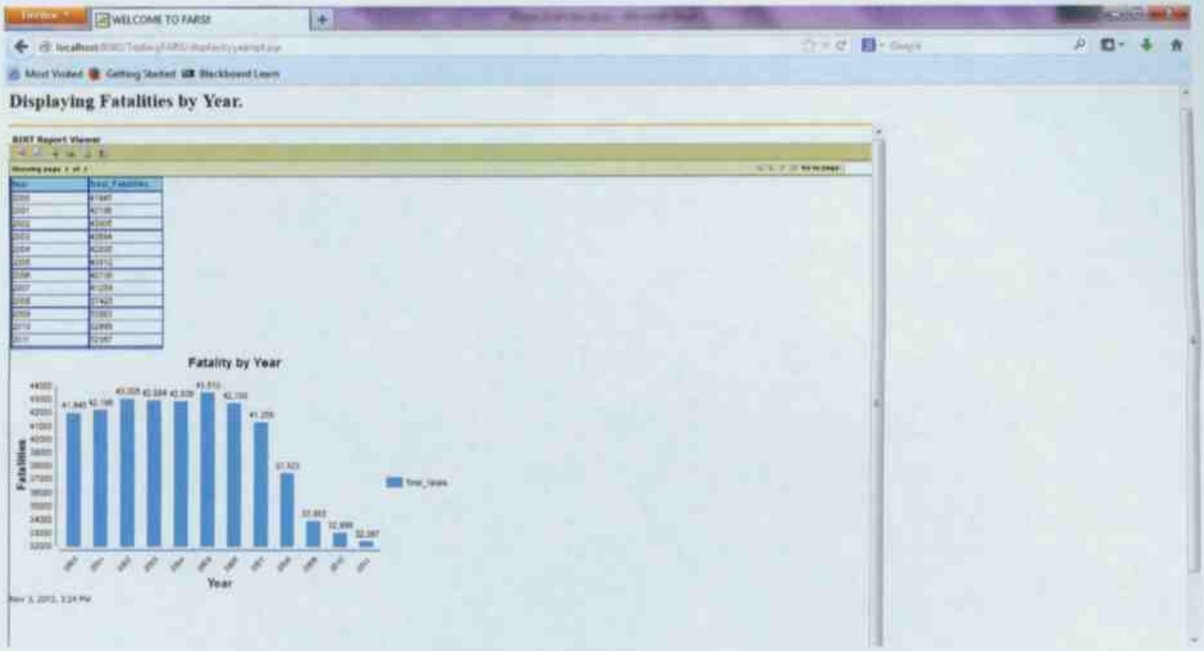


Displaying Accidents by Year.

Year	Total_Accidents
2000	37526
2001	37862
2002	38491
2003	38477
2004	38444
2005	39252
2006	38648
2007	37435
2008	34172
2009	30862
2010	30296
2011	29757

Nov 3, 2013, 3:14 PM

Figure 12. Results displayed for fatal accidents by year.



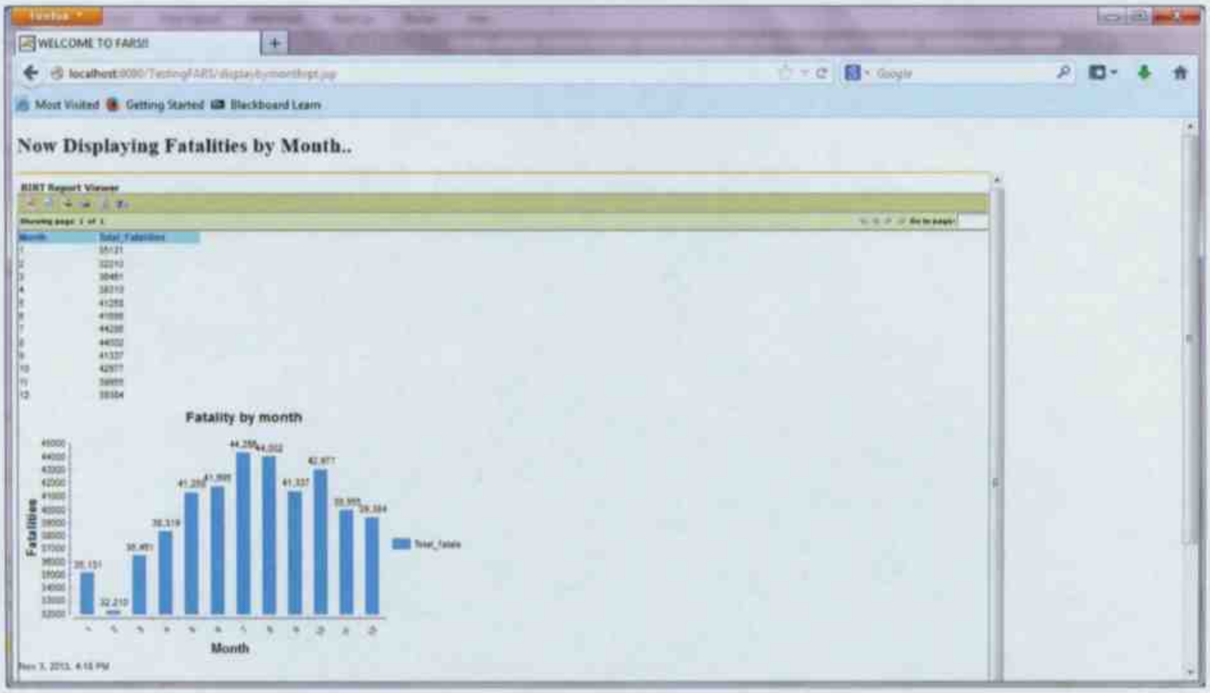
Displaying Fatalities by Year.

Year	Total_Fatalities
2000	41945
2001	42196
2002	43005
2003	42884
2004	42836
2005	43510
2006	42708
2007	41259
2008	37423
2009	33883
2010	32999
2011	32367

Figure 13. Results displayed for fatalities in vehicle accidents by year.

5.2 Accidents by month

Figure 14 and Figure 15 show the responses generated by the BIRT reporting tool upon querying the data warehouse to view the monthly number of accidents and number of fatalities between two years. The interesting trend observable in the result screenshots is that the number of accidents involving fatalities, and the number of fatalities, both, are high in the months of July, August, and October. The month of February has the least number of accidents involving fatalities, and number of fatalities, across all months.



Now Displaying Fatalities by Month..

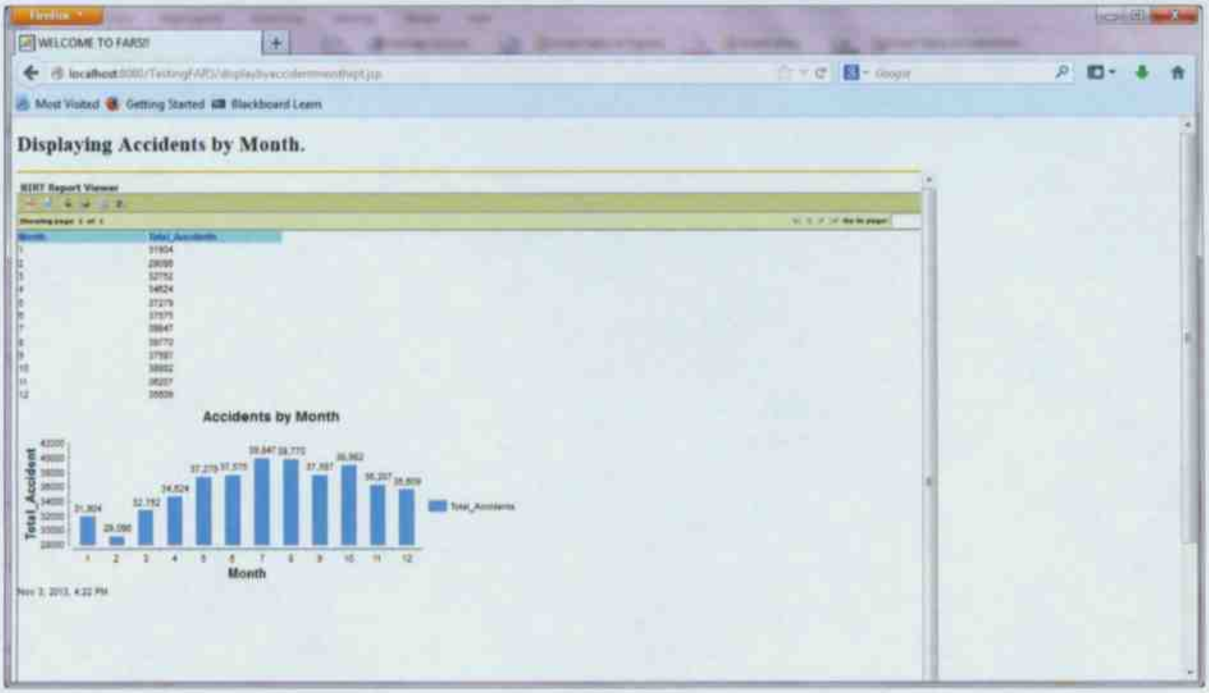
BIRT Report Viewer

Showing page: 1 of 1

Month	Total_Fatalities
1	35121
2	32210
3	36481
4	38319
5	41258
6	41686
7	44238
8	44002
9	41337
10	42977
11	39955
12	39384

Nov 3, 2013, 4:18 PM

Figure 14. Results displayed for total fatal accidents by month between years 2000 and 2011.



Displaying Accidents by Month.

BIRT Report Viewer

Showing page: 1 of 1

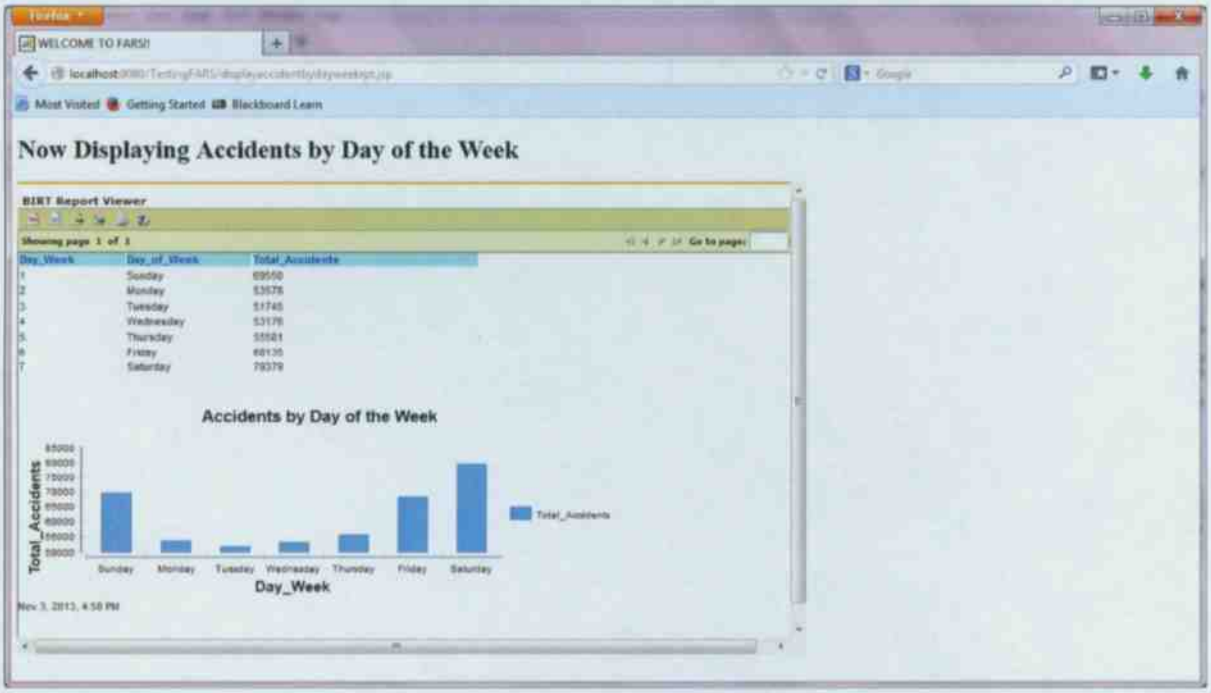
Month	Total_Accidents
1	31904
2	29096
3	32752
4	34624
5	37279
6	37575
7	39847
8	39770
9	37597
10	38962
11	36207
12	35809

Nov 3, 2013, 4:22 PM

Figure 15. Results displayed for total fatalities by month between years 2000 and 2011.

5.3 Accidents by day of the week

Figure 16 and Figure 17 show the responses generated by the BIRT reporting tool upon querying the data warehouse to view the number of accidents and number of fatalities between two years by day of the week. The interesting trend observable in the result screenshots is that number of accidents involving fatalities, and the number of fatalities are both high during Saturday, Sunday and Friday. Also, Tuesday has the least number of accidents involving fatalities, and number of fatalities, across all years.

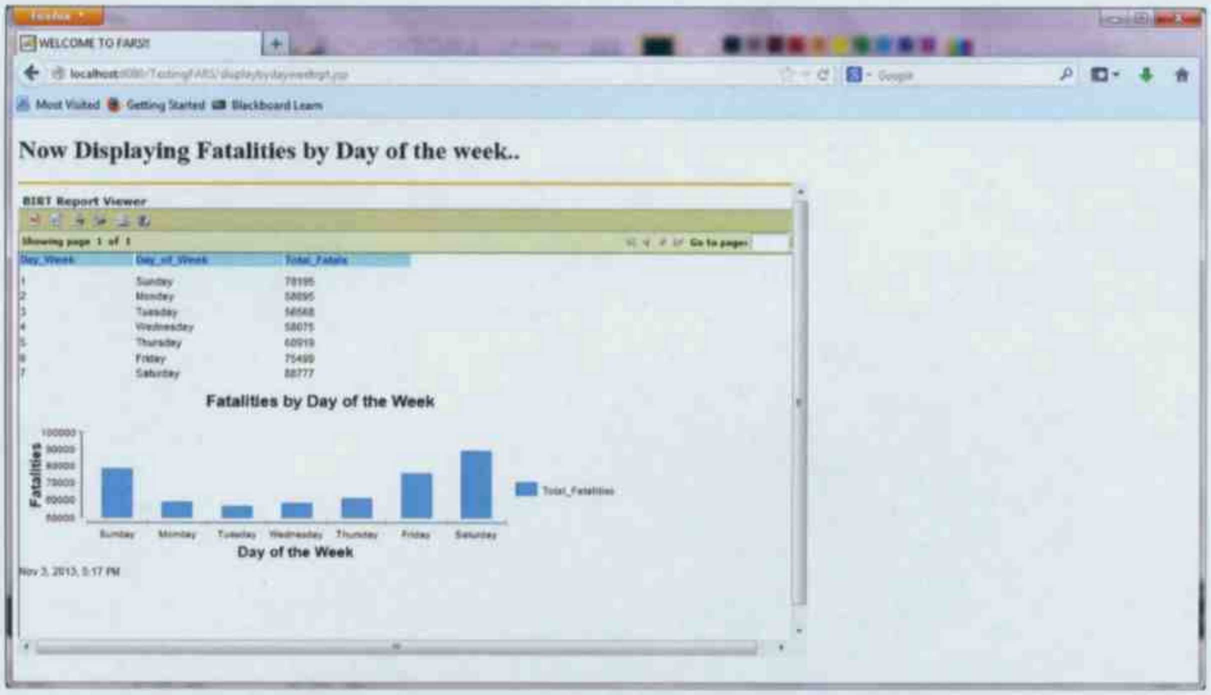


Now Displaying Accidents by Day of the Week

Day_Week	Day_of_Week	Total_Accidents
1	Sunday	69550
2	Monday	53578
3	Tuesday	51745
4	Wednesday	53178
5	Thursday	55581
6	Friday	68135
7	Saturday	79379

Nov 3, 2013, 4:58 PM

Figure 16. Results displayed for total fatal accidents by day of week between years 2000 and 2011.



Now Displaying Fatalities by Day of the week..

Showing page 1 of 1

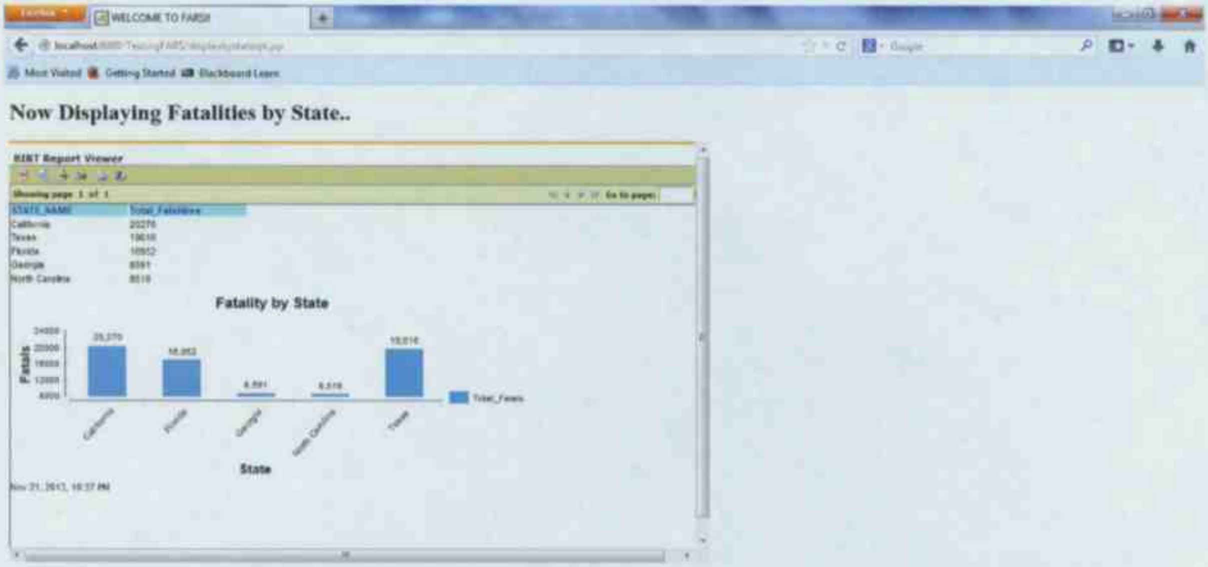
Day_Week	Day_of_Week	Total_Fatals
1	Sunday	78195
2	Monday	58895
3	Tuesday	56568
4	Wednesday	58075
5	Thursday	60919
6	Friday	75499
7	Saturday	88777

Nov 3, 2013, 5:17 PM

Figure 17. Results displayed for total accidents by day of week between years 2000 and 2011.

5.4 Fatalities by State

Figure 18 show the responses generated by the BIRT reporting tool upon querying the data warehouse to view the number of fatalities by state between two years. Because the number of states is high, the report was designed to present top 5 states by fatality. The interesting trend observable in the result screenshots is that, out of all the states in the United States of America, the number of fatalities is high for California, followed by Texas, Florida, Georgia and North Carolina. This trend is logical, given the size and population of these states. Surprisingly, New York does not appear in the top 5 states by number of fatalities, even though New York is a big state with a sizeable population. A likely reason is that a majority of the population is distributed in the New York metro area, where use of mass transit, as opposed to motor vehicles, is more popular.



Now Displaying Fatalities by State..

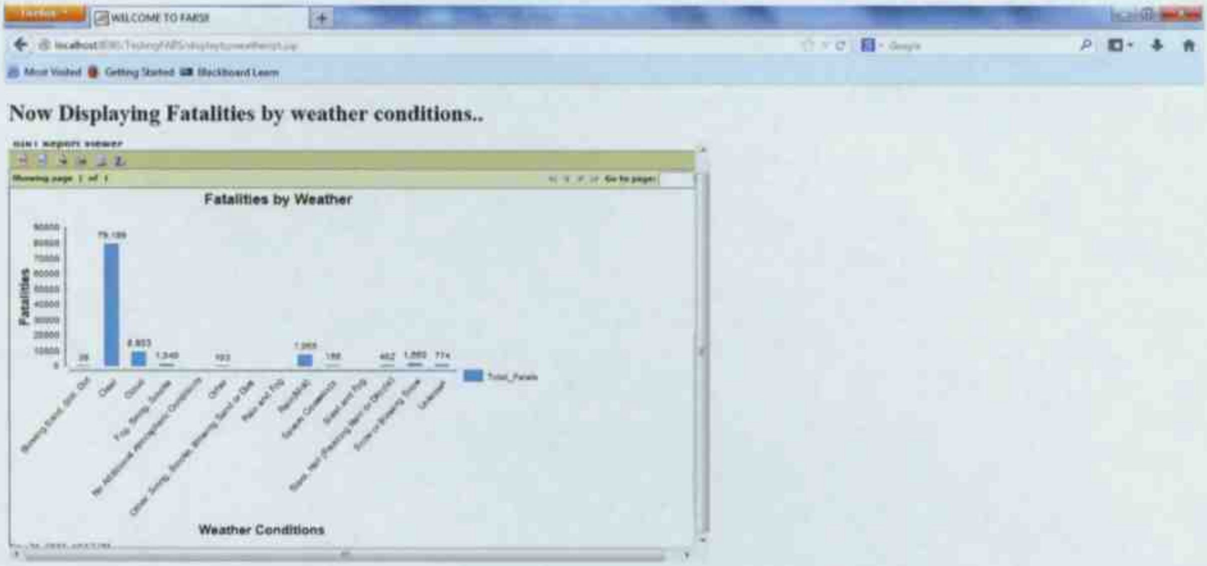
STATE_NAME	Total_Fatalities
California	20270
Texas	19618
Florida	16952
Georgia	8591
North Carolina	8519

Nov 21, 2013, 10:57 PM

Figure 18. Results displayed for total fatalities by State between 2006 to 2011

5.5 Fatalities by Weather

Figure 19 show the responses generated by the BIRT reporting tool upon querying the data warehouse to view the number of fatalities by weather conditions between 2009 and 2011. The interesting, but counter intuitive, trend observable in the result screenshots is that, more number of accidents that resulted in fatalities occurred during clear day. One possible reason is that the number of clear weather days in each year may vastly outnumber poor weather days. Additionally, it is possible that the amount of traffic may be less in days with really poor weather, leading to fewer fatalities.

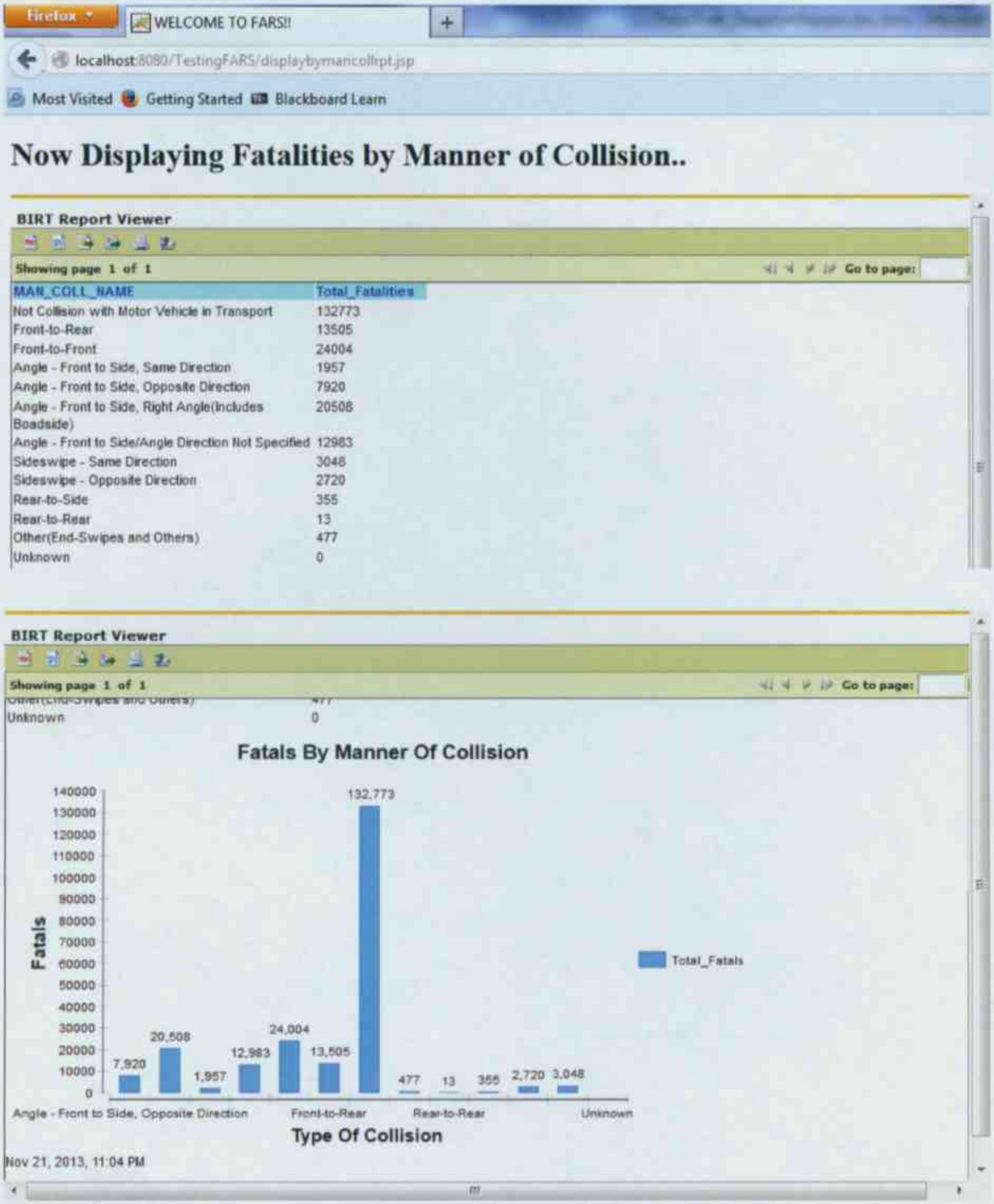


Now Displaying Fatalities by weather conditions..

Figure 19. Results displayed for total fatalities by weather between 2009 to 2011.

5.6 Fatalities by Manner of Collision

Figure 20 show the responses generated by the BIRT reporting tool upon querying the data warehouse to view the number of fatalities by manner of collisions between 2006 and 2011. The interesting trend observable in the result screenshots is that, more number of accidents that resulted in fatalities occurred when the collision did not involve two vehicles in transport, in other words, the collision involved one moving vehicle and another non-moving entity.



Now Displaying Fatalities by Manner of Collision..

BIRT Report Viewer

Showing page 1 of 1 Go to page:

MAN_COLL_NAME	Total_Fatalities
Not Collision with Motor Vehicle in Transport	132773
Front-to-Rear	13505
Front-to-Front	24004
Angle - Front to Side, Same Direction	1957
Angle - Front to Side, Opposite Direction	7920
Angle - Front to Side, Right Angle(Includes Boadside)	20508
Angle - Front to Side/Angle Direction Not Specified	12983
Sideswipe - Same Direction	3048
Sideswipe - Opposite Direction	2720
Rear-to-Side	355
Rear-to-Rear	13
Other(End-Swipes and Others)	477
Unknown	0

BIRT Report Viewer

Showing page 1 of 1 Go to page:

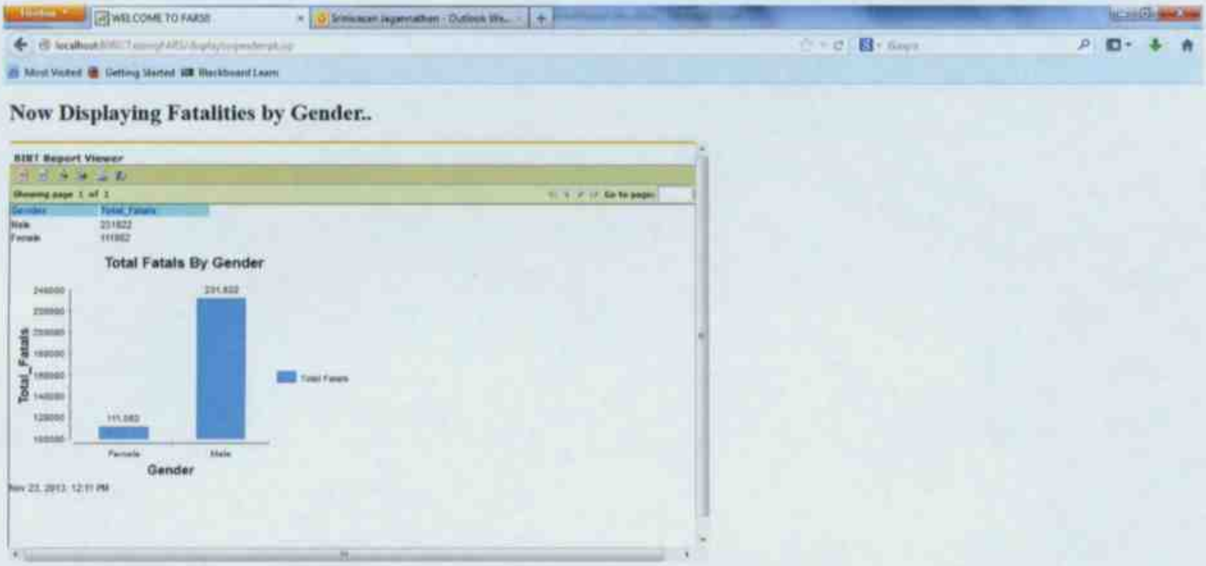
Other(End-Swipes and Others)	477
Unknown	0

Nov 21, 2013, 11:04 PM

Figure 20. Results displayed for total fatalities by manner of collision between 2006 to 2011

5.7 Fatalities by Gender

Figure 21 show the responses generated by the BIRT reporting tool upon querying the data warehouse to view the number of fatalities by gender between 2006 and 2011. The interesting trend observable in the result screenshots is that, male drivers were more likely to die in fatality related crashes than women.



Now Displaying Fatalities by Gender..

BIRT Report Viewer

Showing page 1 of 1

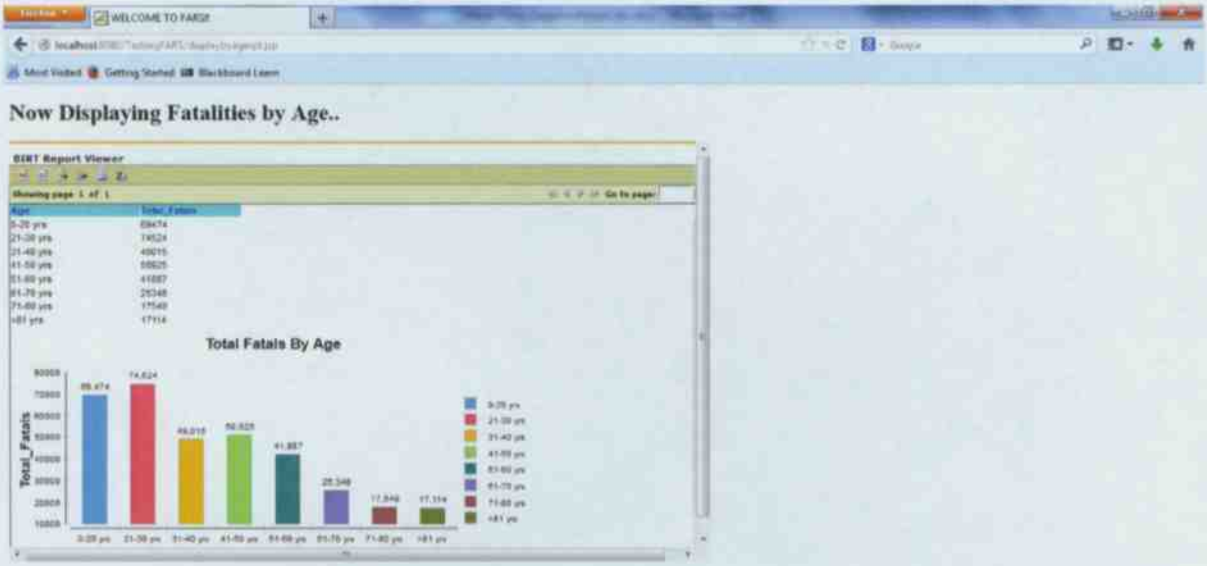
Gender	Total_Fatals
Male	231822
Female	111882

Nov 23, 2013 12:11 PM

Figure 21. Results displayed for total fatalities by gender between 2006 to 2011

5.8 Fatalities by Age

Figure 22 show the responses generated by the BIRT reporting tool upon querying the data warehouse to view the number of fatalities by age groups between 2006 and 2011. The interesting trend observable in the result screenshots is that, the number of fatal accidents is higher for drivers of age group 21-30. One possible reason could be that young drivers take more risks while driving.



WELCOME TO FARS!

Now Displaying Fatalities by Age..

BIRT Report Viewer

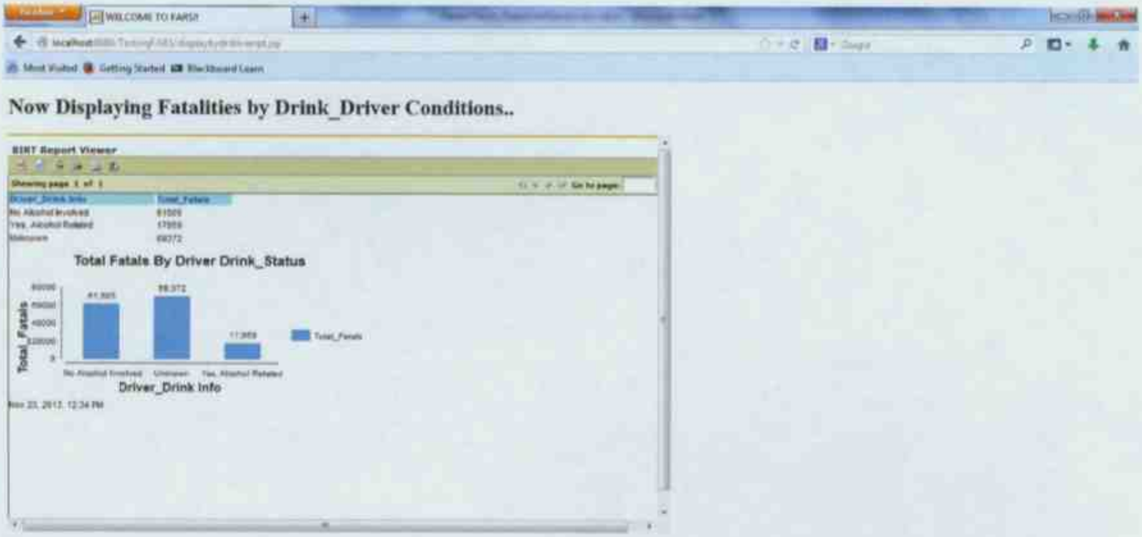
Showing page 1 of 1

Age	Total_Fatals
0-20 yrs	69474
21-30 yrs	74624
31-40 yrs	49015
41-50 yrs	50625
51-60 yrs	41887
61-70 yrs	25348
71-80 yrs	17549
>81 yrs	17114

Figure 22. Results displayed for total fatalities by age between 2006 to 2011

5.9 Fatalities by Drink_Driver

Figure 23 show the responses generated by the BIRT reporting tool upon querying the data warehouse to view the number of fatalities by whether alcohol was involved in the accident, between 2006 and 2011. An interesting trend observable in the result screenshots is that fatalities where alcohol involvement is known with certainty was very less compared to the group where alcohol involvement is not known.



Now Displaying Fatalities by Drink_Driver Conditions..

Driver_Drink Info	Total_Fatals
No Alcohol Involved	61505
Yes, Alcohol Related	17959
Unknown	68372

Nov 23, 2013, 12:34 PM

Figure 23. Results displayed for total fatalities by driver drink status between 2006 to 2011.

Chapter 6: Conclusions and Future Work

6.1 Conclusion

The aim of this thesis is to provide a simplified web based analytics system that helps the user to query the interface so as to get useful trend information. This thesis also explains on how a methodical approach can be applied to analyze data from a public information system so as to populate a data warehouse that is designed to answer trend information queries. This thesis combines the data warehousing approach with business intelligence reporting technology to provide trend information in a simplified format.

I have successfully designed and implemented a data warehouse and query interface to support user queries for fatal accident trend information. The data was collected from the publicly available NHTSA FARS database, and processed before loading into the data warehouse using a hand coded ETL process. Using the data warehouse and querying tools, I uncovered a number of interesting trends in accident fatalities in the United States.

6.2 Future Work

There are a number of areas of future work to build upon the approach discussed in this thesis. One area of further work is to apply the data warehousing methodology discussed in thesis to other public sources of information. For instance, weather data can be analyzed to form a massive data warehouse to gain better insights into global wanning. Another area of work is to use a Map Reduce and NoSQL framework to collect, analyze and store the data. This will be particularly useful for a global weather data warehouse where massive amounts of data is expected. A third area of research is to compare SQL based solutions against a NoSQL approach to understand the cost-performance benefits of each approach.

Chapter 7: REFERENCES

1. Information System, Encyclopedia Britannica, http://www.britannica.com/EBchecked/topic/287895/information-system.
2. Fatality Analysis Reporting System (FARS), http://www.nhtsa.gov/FARS.
3. NHTSA FARS Data FTP Site, ftp://ftp.nhtsa.dot.gov/fars/.
4. NCSA Data Resource Website, Fatality Analysis Reporting System (FARS) Encylopedia, http://www-fars.nhtsa.dot.gov/Main/index.aspx.
5. NHTSA FARS Query Selection Tool, http://www-fars.nhtsa.dot.gov/QueryTool/QuerySection/SelectYear.aspx.
6. NHTSA FARS Occupants Trends, http://www-fars.nhtsa.dot.gov/Trends/TrendsOccupants.aspx.
7. Research Problems in Data Warehousing, Jennifer Widom, Proceedings of the 4th International Conference on Information and Knowledge Management, November 1995, http://ilpubs.stanford.edu:8090/91/1/1995-24.pdf
8. A comparison of data warehousing methodologies, Arun Sen and Atish Sinha, Communications of the ACM - The disappearing computer, Volume 48 Issue 3, March 2005, pp. 79-84, http://student.bus.olemiss.edu/files/conlon/others/Others/Bus669_CompInfo/DataWH/A%20comparison%20of%20data%20warehousing%20methodologies%20-sen.pdf.
9. Best Practices Physical database design for data warehouse environments, by Maksym Petrenko, Amyris Rada, Garrett Fitzsimons, Enda McCallig, and Clisto Zuzarte, IBM, 2012, https://www.ibm.com/developerworks/community/wikis/form/anonymous/api/wiki/0fc2f498-7b3e-4285-8881-2b6c0490ceb9/page/2d6faf27-ee09-455f-b88f-9ac9b4a9c212/attachment/23ca37f4-53db-4e2e-b940-bafa2f3476a2/media/DB2BP_Warehouse_Design_0912.pdf
10. Developing High Quality Data Models, Matthew West and Julian Fowler, The European Process Industries STEP Technical Liaison Executive (EPISTLE), September 2003.
11. Star Schema, The Complete Reference, Christopher Adamson, 2010.
12. Toward XML-Based Data Warehouse Architecture, by Rami Rifaieh and Nabila Aïcha Benkat, Idea Group Publishing, 2003, http://www.irma-international.org/viewtitle/32072/.
13. BIRT Overview, http://www.eclipse.org/birt/phoenix/intro/.
14. Spago BI, http://www.spagobi.org/
15. Pentaho, http://www.pentaho.com/

16. Jaspersoft, http://www.jaspersoft.com/
17. Traffic Safety Facts, 2011, http://www-nrd.nhtsa.dot.gov/Pubs/811754AR.pdf.
18. Fatality Analysis Reporting System (FARS) Analytical Users Manual 1975-2011, http://www-nrd.nhtsa.dot.gov/Pubs/811693.pdf.
19. dbf2csv http://sourceforge.net/projects/dbf2csv/.
20. Transaction reordering and grouping for continuous data loading, by Gang Luo, Jeffrey F. Naughton, Curt J. Ellmann, Michael W. Watzke, Proceedings of the 1st international conference on Business intelligence for the real-time enterprises, 2006, http://pages.cs.wisc.edu/~gangluo/tpump_full.pdf.
21. Transaction reordering with application to synchronized scans, by Gang Luo, Jeffrey F. Naughton, Curt J. Ellmann, Michael W. Watzke, Proceedings of the ACM 11th international workshop on Data warehousing and OLAP, 2008, http://pages.cs.wisc.edu/~gangluo/syncscan.pdf.
22. "Google File System" by Sanjay Ghemawat, Howard Gobioff and Shun-Tak Leung, 19th ACM Symposium on Operating Systems Principles, 2003, http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/archive/gfs-sosp2003.pdf
23. "MapReduce: Simplified Data Processing for Large Clusters ", by Jeffrey Dean and Sanjay Ghemawat, 5th Symposium on Operating Systems Design and Implementation, 2004, http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/archive/mapreduce-osdi04.pdf.
24. "Bigtable: A Distributed Storage System for Structure Data," by Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber, 7th Symposium on Operating Systems Design and Implementation, 2006, http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/archive/bigtable-osdi06.pdf.
25. ETLMR: A Highly Scalable ETL Framework Based on MapReduce, by X. Liu, C. Thomsen and T. B. Pedersen, Proceedings of 13th International Conference on Data Warehousing and Knowledge Discovery, 2011, http://vbn.aau.dk/files/66687494/etlmr.pdf.
26. MapReduce-based Dimensional ETL Made Easy, by Xiufeng Liu, Christian Thomsen, Torben Bach Pedersen, Proceedings of 38th International Conference on Very Large Data Bases, 2012, http://vldb.org/pvldb/vol5/p1882_xiufengliu_vldb2012.pdf.
27. Hive - A Warehousing Solution Over a Map-Reduce Framework, by Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Suresh Anthony, Hao Liu, Pete Wyckoff and Raghotham Murthy, Proceedings of 35th International Conference on Very Large Databases, 2009, http://www.vldb.org/pvldb/2/vldb09-938.pdf.

28. Hive - A Petabyte Scale Data Warehouse using Hadoop, by Ashish Thusoo, http://www.facebook.com/note.php?note_id=89508453919.
29. Apache Hadoop, http://wiki.apache.org/hadoop.
30. Data Warehouse Star Schema Tutorial, Debbie Chu, http://www.fbe.hku.hk/~is/busi0092/Notes/t1_dataWarehouse_full_v3.pdf.