ACCOMPLISHMENTS & BENEFITS TO ENTERPRISE CLASS LINUX – Groups of people in many diverse fields face the challenge of characterizing or mining information from a large data set. In science, the data set might contain astronomy or human genome data. In business, the data set might contain customer purchase or supply chain data. In sociology, the data set may contain census or demographic data. Companies like FedEx, UPS and Wal-Mart and scientific projects like the NASA space missions and the Human Genome Project all have terabytes of data being examined by hundreds of people. Wherever large databases exist, regardless of their exact contents, there are groups of people seeking to understand that data. Often, this is done with individual SQL queries that summarize certain aspects of the data with little support for collaboration between people. People may work together on forming a query to submit to the system, but the system offers no direct support to help identify and exploit possible collaboration. This lack of collaboration has three main drawbacks. (1) Running a query over a large data set can take hours or even days and users often run many of the same basic SQL queries to summarize basic aspects of the data set. With support for caching, users could benefit from answers obtained by other users. (2) As users move beyond basic queries, it can often take several tries to write a query that accomplishes the intended purpose. With support for collaboration, users could learn from each other's mistakes by modifying previously run successful queries. It is easier to modify a working query than to write a completely new query from scratch. (3) Users may be characterizing a similar aspect of the data, but would be unable to help one another modify their queries. Without support for collaboration, they would be wasting resources by running similar queries and would be unable to build on each other's ideas. To address these problems, we present the design of a generic system for collaborative exploration of large data sets. Our system has the following characteristics: (1) It caches the results of commonly executed queries to avoid repeating expensive, long-running queries. (2) It automatically caches the results of basic data characterization queries like a histogram of values for each row element. (3) It allows new users to build on the successful queries of others, making the system easier for new users. This could reduce the number of expensive, long-running queries that are caused by user error and return undesired results. (4) It allows expert users to export polished query sets that answer important questions and benefit others. (5) It allows users to export collaborative query sets that invite others to refine queries in real time. (6) It provides a generic interface that will work across any data set that can be represented in a relational

database. In other words, the system need not be customized based on the category of data being explored. **COMPLETION** – For this project, we began to develop an instance of the Generic Database Explorer, described above. This domain-specific implementation, called the BDAP Explorer, is an interface to a large collection of Border Gateway Protocol (BGP) data. The purpose of this project is to share the information that has been collected and analyzed, so that other researchers and anyone else interested in the study of BGP data can query, view, comment, and understand the data and its implications. We created a collection of tools to help the user more easily explore and characterize a large data set regardless of their experience in the field. **Browse Schema** – To explore the data set, you should first browse the database schema. This schema browser lists the available databases used in the project. When a database is selected, the list of tables in that database is displayed. Then, when a table is selected from that list, the fields of that table are displayed. Finally, once a field of a table is selected, sample rows or a histogram of the values for that field are displayed. Descriptions are either automatically generated by the system or provided by an administrator of the data set. **Cached Queries –** To show you correctly-written queries that have been run and to also give you ideas for queries that you may want to run on the data set, there are a list of Cached Queries available. Cached Queries are queries that have already been run and the results have been stored. For large data sets, like the ones we are dealing with, returning a cached result can save hours or days of intense computation. Also, once you have run a new successful query, you will be given the option to cache the results. The system determines which queries should be cached based on five primary factors. **Most Recent Queries –** To show you the N most recent queries that other users have submitted and to help build a sense of community when exploring and characterizing the data set, a list of the Most Recent Queries is provided. The number of queries listed (N) can be configured by the data set administrator. These queries can allow you to collaborate with other users that are working on the same or similar problem as you. The most recent queries are not necessarily cached nor are they the queries that were intended to solve the problem, but they give a sense of the current problems that are being worked on by others using the system. **Basic Summary Queries –** To help you become familiar with the data set, we have developed some basic summary queries to get you started. Basic Summary Queries provide a summary of the database and table information to help you characterize the data set that is being explored. These queries are commonly run by users that are new to the data set in order to get a better

understanding of its basic characteristics. The results of these queries are typically small, but the run times can be long. These results are cached and are available for your convenience. **Polished Query Sets –** Once you feel that you have a set of queries that answer an important question or questions that could benefit others, you can export these queries as a Polished Query Set. There is a list of Polished Query Sets available that will help you see the problems that other users have already solved or have started to solve. You can then browse the contents of the query sets to learn from them and also modify and run them for use in your own query sets. From these Polished Query Sets, users can begin to get ideas for problems to work together on. **Query Builder Tool –** To help you build simple queries by choosing the database, tables and fields that you want to use in the query, we provide a Query Builder Tool. Once you select the database, tables and fields that you want to use, the query is placed in a textarea so that you can manually edit it to create more advanced queries. **FUTURE WORK** – We have been keeping an extensive list of bugs from beta and usability testing. We are also keeping a list of features and recommendations to be added in the near future. Both lists can be found in the project documentation. **TECHNICAL & PERSONAL LESSONS –** We have learned techniques for better coding and have learned more about open source software development. We learned more about PHP structure and syntax and learned more about web development using HTML, CSS, and JavaScript. In particular, we learned the importance of validating HTML pages using W3C's online validator or other validator built into the browser. Finally, we learned how the presentation of the information on a webpage makes a huge difference to its usability and overall effectiveness. **EXTERNAL TOOLS USED & WHY** – Since users need a way to quickly and easily become familiar with the structure and contents of the databases, we first build a Generic Database Explorer. This database explorer reads and displays the descriptions and structure from specific infrastructure tables (these tables are not the domain-specific tables that contain BGP data used in our BDAP Explorer, but instead, are the administrative tables that are used in it). Our Generic Database Explorer is intended to be an easy-to-use interface to view the contents of any SQL database. It provides basic information, such as the number of tables, fields, and rows. It can also provide customizable levels of descriptions for the database and its contents. There is existing software that can provide some level of database exploration, but this software would need to be modified to use our infrastructure tables. Also, often this software has too much extra functionality, which adds complexity, gives too much power, and reduces usability.