Data Café: A Dining Car Approach to Educational Research Data Management and Distribution

Allan Knight, Kevin C. Almeroth, and Hangjin Zhang
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, California, USA
{aknight,almeroth,hangjin}@cs.ucsb.edu

Rich Mayer, and Krista DeLeeuw
Department of Psychology
University of California, Santa Barbara
Santa Barbara, California, USA
{mayer,deleeuw}@psych.ucsb.edu

Researchers are investigating the effect of technology-supported instructional methods on the learning process. One team investigating this effect is the Technology in Education (TIE) group at the University of California, Santa Barbara. A problem these researchers faced, however, was how to collect and distribute the data needed for the wide variety of analyses. The team designed the Data Café statistical database application to provide access and an intuitive interface. Furthermore, the application also provides an up-to-date online "codebook" shared amongst the entire research team. This paper describes this application and shows that it is scalable and flexible. The Data Café provides a diverse group of researchers with a simple to use interface that allows them to download exactly the data they need in exactly the format they need it.

Introduction

The term "technology in the classroom" has become cliché. The question that researchers are now investigating is the impact of technology-supported instructional methods on the academic performance of students in and out of the classroom. The Technology in Education (TIE) Group at the University of California, Santa Barbara is a diverse group of researchers investigating the efficacy of a variety of instructional technologies in college classrooms. This group, made up of a computer scientist, a political scientist, a linguist, a psychologist, an education researcher, and graduate students from various disciplines, has begun to collect and analyze data. Our goal is to determine if technology-supported instructional methods can bridge the gap between under-performers and over-achievers, as well as bridge the gap between under-represented groups and their counterparts.

The basic research methodology of the TIE group is to find willing instructors that will allow TIE to observe their courses in at least two iterations: one that uses little or no technology, called the control; and a second instance that uses a set of technologies that support instructional methods intended to foster deep learning. The group collects information about the students' socio-economic background, their feelings about technology, and their academic performance. TIE collects and analyzes this information to find correlations between performance, technology, and a variety of other factors, *e.g.*, socio-economic status. This research has led to several publications (Bulger, Mayer & Almeroth, 2006; Mayer, Almeroth, Bimber, Chun, Knight & Campbell, 2006; Mayer, Stull, Campbell, Almeroth, Bimber, Chun & Knight, 2006). The problem is that each of the members of the diverse research group has a unique perspective and different needs for which parts of the data set are used in their analyses.

The process of maintaining and disseminating the resulting data is a difficult problem. Each researcher requires a different set of information. Each researcher requires storing the information in different formats. Furthermore, researchers must have consistently up-to-date information as quickly and accurately as possible. Also, each of these requirements needs to be addressed without burdening any individual too much. In other words, we would very much like to not have to appoint a single person as the "database/data manager". The question the TIE group faced was how to manage the data in such a way that all of the researchers could have access to the latest information for analysis, use a heterogeneous set of analysis tools, and not rely too heavily on one particular individual.

Before beginning an implementation of a solution, we explored and discussed several alternate approaches. The "typical" approach, and the approach the TIE group started with, was to store all the data in Microsoft Excel files. The data were organized across multiple Excel files, one for each of the different instances of the courses. Each file contained multiple worksheets for each of the sources of information: data from the campus registrar, pre-questionnaire, post-questionnaire, and course grades. The problem with this strategy was that gathering the data necessary for any particular analysis was time consuming and repetitive, and it was difficult to match data from the same learner across multiple spreadsheets. Also, this strategy relied too heavily on one graduate student and did not allow for researchers to access the data easily in the necessary application formats. We also considered other approaches such as storing the data in flat text files or a database. An approach was needed that addressed all of the needs of each researcher.

Our approach, called the Data Café, has several unique features. First, while we do use a database to actually store the data, we designed a unique schema to achieve flexibility and performance. Rather than store data from each source in individual tables, with each row representing a student, each data points is stored in its own row. Secondly, in order to provide researchers with data, anytime, anywhere, in multiple formats, the data is made available through a simple-to-use web interface. Our goal was that instead of a single graduate student maintaining the data and re-distributing it after the data is modified, researchers could select only the data they need for a particular round of analysis and choose the application format in which to download the data. By storing the data in a database, we overcame the three disadvantages mentioned above. First, the data is available anytime and anywhere from any web browser. Second, the database does not dictate any specific application format; therefore, a researcher can request in one of several formats. Finally, flexibility in the types of data stored and high performance data access is maintained for even large amounts of data with our unique schema that keeps the number of tables static regardless of the number of data sources.

There are many broad areas of research related to the Data Café. The three most relevant areas are: decomposition storage models, data warehousing, and statistical databases. The decomposition storage model research asks which is a better format to store data: column or row stores. Our design of the database resembles a column store, but our design works at the schema level. Data warehousing is a broad topic that looks to find ways to query and integrate data from disparate sources. Our work is similar, but we do not need to integrate data from several database sources. Instead we collect binary sources and integrate them into one database. Our schema may, however, offer insight into how best to design data stores to avoid the issues associated with integrating data in a data warehouse (Kolaitis, 2005). Finally, our work is a large statistical database. The research in this area, however, does not cover the best practices for managing metadata and data in an efficient and flexible manner.

When Data Café was complete, we evaluated it. Our goal was to compare the performance of a typical schema design with our unique schema design. Additionally, our goal was to show that Data Café is scalable even as the amount of data and the number of sources increase. This result is especially important to preserve flexibility as the needs of researchers and the sources of data grow. We feel that this analysis shows that the Data Café design and implementation are well suited, not only for educational research, but for other types of research as well.

The remainder of this paper is organized as follows. Section 2 discusses the work related to ours. Section 3 discusses who we believe are the target users of the Data Café. Section 4 discusses different methods of storing research data. Section 5 discusses different methods for distributing research data. Section 6 provides a quantitative analysis of the performance of the Data Café. Finally, Section 7 provides a summary of the paper and mentions future work for the Data Café project.

Related Work

The work related to the Data Café falls into three main categories (1) statistical databases, (2) data warehousing, and (3) database storage models. The following section discusses the pertinent works in these three areas and identifies where the work on the Data Café fits within these areas and what it contributes beyond the research previously done.

Statistical databases are databases tailored specifically for storing numerical data for statistical analysis. Some of the seminal work in this area is by Shoshani (Shoshani, 1982) and McCarthy (McCarthy, 1982). The contribution of these papers is to define the area and begin discussing the problems associated with managing such databases. Shoshani details problems with the user interface, logical modeling, data management, and security. The Data Café must address all of these issues as well. McCarthy specifically addresses the issue of metadata management for statistical databases and provides a hierarchical organizational scheme for metadata. This paper, however, does not address the issue of logical organization of the data and how it relates to the metadata itself. The metadata referred to in this paper relates to that stored physically in the database table descriptions, not information as it would be seen by the user. Most databases do not allow the association of textual descriptions of fields at that level. The Data

Café allows the user to define a much greater amount of metadata for each field type and allows for the definition of enumeration types used by modern statistical analysis packages such as SPSS (SPSS, 2007) and SAS (SAS, 2007).

Data warehousing is a very broad topic. The main idea behind data warehousing is to provide a single point for users to access data from disparate sources, usually other databases (Greif, & Sarin, 1986). The main problems associated with data warehousing are integrating and exchanging data between these sources (Widom, 1995). The difference between integrating and exchanging is mostly superficial as identified by Kolaitis (Kolaitis, 2005). Integration refers to displaying a virtual live view of the data from multiple sources whereas exchanging refers to copying data from multiple sources into a new target database. There are a multitude of papers that deal with these two topics (Aydin, Altay, Aktas, Necati, Aysan, Fox, Ikibas, Kim, Kaplan, Topcu, Pierce, Yildiz & Balsoy, 2003; Kent & Schuerhoff, 1997; Müller, Stöhr & Rahm, 1999). We believe, at least for smaller data stores, our schema avoids most of the problems of integration and exchange by decomposing the data into rows within the database. Incidentally, this body of work also provides a classification of two metadata types identified by Stöhr (Müller, Stöhr, & Rahm, 1999). Technical metadata deals with the physical implementation of the data store, in other words, data used by the programmers. Semantic data refers to the relationship and meaning of data used by users in searching and in the integration and exchange process. The metadata used in the Data Café can be classified as technical data; however, it also has utility for statistical analysis.

Finally, there exists work that relates to the logical design of the database itself. The Data Café uses a decomposition model similar to the one first expressed in the work of Copeland (Copeland & Khoshafian, 1985). Khoshafian also mentions this approach and creates a query process for decomposition storage model databases (Khoshafian, Copeland, Jagodis, Boral & Valduriez, 1987). They decompose each column of the data into its individual pieces, which is stored with a surrogate, now called an index identifier. This approach, however, leads to performance issues when integrating data across tables. With this type of design, joining each table will severely degrade performance when querying the data. Data Café avoids this performance degradation by maintaining a fixed number of tables and decomposing each column into a row of a single table and also associating a field identifier with each data point. A further difference is that the work related to data decomposition does so at the database level, the Data Café does so at the table level. This fact means that our design need not reside on a database specifically designed for the decomposition storage model. There are several other works regarding column store databases (Stonebreaker, Abadi, Batkin, Chen, Cherniack, Ferreira, Lau, Lin, Madden, O'Neil, O'Neil, Rasin, Tran & Zdonik, 2005; Abadi, 2007). These types of databases store their data grouped by columns rather than grouped by rows, as is done by traditional database applications, such as Microsoft's SQL Server (Microsoft, 2007) and PostgreSQL (PostgreSQL, 2007). The theory being that column stores more efficiently store and retrieve data for either wide and/or sparse databases (Harizopoulos, Liang, Abadi, & Madden, 2006). We have achieved a similar performance advantage on a row store database by logically decomposing tables into column tables. Again, column stores, like decomposition storage models, work at the database level, not the table level. Ironically, if the data from Harizopoulos is correct (Harizopoulos, Liang, Abadi & Madden, 2006), our table decomposition technique should achieve better performance on a row store database than on a similar column store database.

Targeted Users

Before describing the Data Café application, we first discuss the target audience for this application. The key word in this discussion is "diversity". The Data Café is best suited for research teams composed of researchers from diverse research fields with support for a diverse set of applications. One might think that the amount of data could also be an important factor. We believe, however, that even if there are as few as several hundred data points, Data Café is an appropriate application for distributing and managing research data. The following discussion elaborates further on these three points.

First, diversity of researchers and diversity of application support are very important when deciding how to distribute research data. They are also directly related. Researchers within a common community tend to use the same applications for statistical analysis. On our own research team, we find that education and psychology researchers tend to use *SPSS*. On the other hand, computer scientists tend to use applications such as *gnuplot* or *R* for statistical analysis. This variety of applications requires that any distribution of research data account for the differing file formats of the data itself. Any application used for the distribution of data must support multiple data file formats, and Data Café currently supports several of these data file formats. Updating the Data Café to support further formats is straightforward.

As for the amount of data, we believe that it is not a significant factor in making the decision to use a system like the Data Café. The diversity of researchers and application support greatly outweighs data size considerations. Even if the number of data points is quite low, the need to support multiple application formats still exists. If a particular research group has no diversity of researchers, and is sharing a small amount of data, then Data Café may not provide any advantage over sharing a single data file. Conversely, if there is a large amount of frequently updated data, even in a homogenous research group, Data Café may help in better managing the research data.

Storing Research Data

The first question to answer when managing data for a research project is how to physically store the data. We discussed several possibilities. The first possibility was to store the data as it was collected: in a Microsoft Excel file. The graduate student in charge of collecting the data was already using this application file format. Another possibility was to store the data in a flat text file. This format would offer greater flexibility to the researchers for statistical analysis and was easy to implement since Excel allows for the exporting of files directly to flat text. The third possibility considered was to store the data as XML. We explored this possibility because of the popularity of XML, but immediately rejected it because of the extra work needed to create the XML schema and to produce an application capable of reading the data. The final idea was to store the data using a Relational Database Management System (RDBMS). This approach was appealing because it would allow remote access to the data. The only remaining consideration was the interface to be used to access the data. Therefore, we decided to use a RDBMS.

The decision to use the RDBMS was based on the disadvantages of the previously discussed approaches. First, as previously stated, the XML file format was not feasible. Using these types of files requires several laborious steps. After building an XML schema, or organization of the data representation, there would need to be a way to access and export the data to the applications the researchers use for statistical analysis. We deemed implementing such an application as wasteful and impractical, as we believed that storing data in a RDBMS would be better suited for such an application. After all, the application we would write would need to store and retrieve data from the XML file. The bottom line was that this new application would perform the same functionality provided by an RDBMS.

For flat text files we export the data from Excel into tab delimited or comma separated values text. We rejected simple flat text files for exactly the same reasons as the XML file format. Finally, we also decided against using Excel files to store the data. The reason for this rejection was twofold. First, some researchers wished to use SPSS for statistical analysis and exporting the data with the proper metadata was too time consuming. Second, we believed that, in general, distributing data by sharing a set of files is more complicated than using an RDBMS. The next section provides additional details on this point.

We finally decided that storing the data using an RDBMS was the best solution. We came to this decision for several reasons. First, because any application we could write to search and aggregate data from the text files would be a reimplementation of the functionality of an RDBMS. Second, we could more easily keep track of the metadata associated with the data fields using an RDBMS. Some statistical analysis packages allow the usage of metadata for processing and displaying results of the analysis. For example, SPSS allows for enumerations, so if the value 1 is stored for the gender of a student. SPSS can translate this value to the more explicit textual representation, “male”. These translations are necessary since the data we received from the university registrar was in the form of numbers along with their textual representations.

Once we decided to use an RDBMS the next question was how to organize the data within it. The *naïve*, or typical, approach would be to store the data in tables for each data source. After all, this is how researchers would likely view the data. There are, however, two problems with this design. First, associating the metadata with each column would be difficult. The challenge would be how to associate metadata information with a column in a table. Second, as the sources of data increase, the number of tables increases. This fact starts to cause performance issues as researchers retrieve data from an increasing number of sources. A researcher requesting data from multiple sources requires accessing and aggregating multiple tables when retrieving the data. Such retrievals necessitate a join operation in the RDBMS.

Student ID	Age	Gender	Birth Date
1	19	F	032388
2	20	M	052087
3	18	M	041489
4	22	F	012685
5	20	F	101087

Table 1. Typical table design for single data source.

Student ID	Value	Field ID
1	19	1
1	F	2
1	032388	3
2	20	1
2	M	2
2	052087	3
3	18	1
3	M	2
3	041489	3
4	22	1
4	F	2
4	012685	3
5	20	1
5	F	2
5	101087	3

Table 2.Decomposed table design for multiple data sources.

A RDBMS join operation entails collecting each row from one table and associating it with each row in a second table. The result produces n ´ m rows where n is the number of rows from the first table and m the number of rows from the second. These joined rows are then filtered based on keys from each table. For example, if Table A has 100 rows and Table B has 1000 rows, a join query involving Tables A and B would result in filtering 100,000 rows before the final join table is created. If more than two tables need to be joined, the problem becomes even greater. The resulting join table would contain n_1 ´ n_2 ´ … ´ n_i rows. The implication is that as the number of potentially joined tables increases, the performance decreases rapidly. The challenge is then how to keep the number of tables low, or better yet, constant, for a set of data that does not have a predetermined number of sources,

We developed a database schema that addresses this issue and keeps the number of tables constant. The key is to vertically partition or decompose the tables about the columns. But unlike vertical partitioning and decomposition, our schema keeps all the columns in one table. By adding a column to each row that identifies which field the data being stored is associated with, we avoid the usual performance issues associated with vertical partitioning or decomposition. As an added benefit, the identifier also allows for simpler maintenance of metadata pertaining to each field type in the database.

For example, Table 1 illustrates a typical table design for a student database consisting of rows of students with each column holding specific registrar data points such as age, gender, and birth date. Table 2, on the other hand, represents our decomposition of the columns into rows with the column headers replaced with field identifiers, which point to field identifiers stored in a separate table. Both the following section and the evaluation section of this paper demonstrate the advantages of this type of schema. In the next section we outline our strategy for data distribution, which outlines our metadata management as well. Also, the evaluation section demonstrates how our schema can result in better performance over the typical table designs, even for the same amount of data points.

One final alternative still exists. Instead of increasing the number of tables as the number of sources increase, we could create a single table with each data field represented by a column. We ultimately did not use this design because it requires expanding the table as we add more fields. The effect would be that the table would most likely be sparsely populated for each student, as not all data sources may include data about each student. Furthermore, this schema assumes that there is one set of data values for each student, and is therefore not considered flexible enough to handle all data needs. For example, web log data for each student does not work well in this type of schema. In a web log, multiple events are associated with each student, which would result in multiple rows in the database for a single student for each logged event. Our design, however, could easily accommodate such a situation.

Distributing Research Data

The final problem is how to allow researchers access to the information stored in the RDBMS. Programmers and tech savvy individuals understand the process of retrieving data from an RDBMS using a Structured Query Language (SQL). But we could not expect the same ability in non-programmers and non-tech-savvy researchers from diverse fields. We, therefore, needed a way to easily provide access to the data to all of the researchers.

We debated several approaches to distributing the data we collected. Again we had the *de facto* approach that we were already using: whenever the graduate student in charge received new data he distributed it to all the interested parties. Another technique discussed was to save the data on a centralized network file server. Finally, we also discussed saving the data onto some portable media such as CD-ROM distributed to the researchers as needed. We immediately rejected this last approach, as it either required one medium distributed amongst all the researchers, or multiple copies of the medium. The former meant that researchers would be waiting on whoever currently held the medium, and the latter meant devising a technique to know when the current medium became out of date. We eventually decided on distributing the data form a centralized server, not on a file server, but in an RDBMS.

We decided to not distribute a single file through electronic means such as e-mail amongst all the researchers for several reasons. First, this technique is similar to distributing the data on some storage medium and has the same disadvantages. Second, there exists a single, highly unreliable, point of failure: the graduate student. Not that our graduate student was unreliable, but more in general, people are unreliable. Graduate students get sick. Graduate students move on to other projects, as was the case in our project. We therefore deemed it a risk to leave such important data to the responsibility of a single, or even multiple pair of hands (not to mention the fact that the data could wind up in the hands of those not meant to view the data). Finally, we needed to decide the application file format to store the data. The first approach was to use one format and allow the researchers to convert it as they needed. We rejected this approach since each of the researchers would need to convert all the data they needed each time they received updated data. The second approach was to have the graduate student save the data in all the necessary formats. This approach seemed untenable since it required the graduate student to convert the data to many formats every time the data was distributed.

We decided to use a web interface to allow the researchers to tailor the data to their exact needs and to remove the single point of failure. First, by using a web interface, the researcher can download only the exact data they need, not all the data as the other techniques would require. Second, using a web interface allowed for greater security. Access can be restricted only to authorized users. While the user must still protect the received data, this solution remains safer as unintended distribution is less likely to occur than a disk or other media accidentally left in some random machine. Finally, an unforeseen advantage to both the web interface and metadata storage as outlined above was the automatic creation of a codebook for the entire research project. This codebook allows not only online viewing of the codebook, but online modification resulting in an always up-to-date codebook available to all researchers involved.

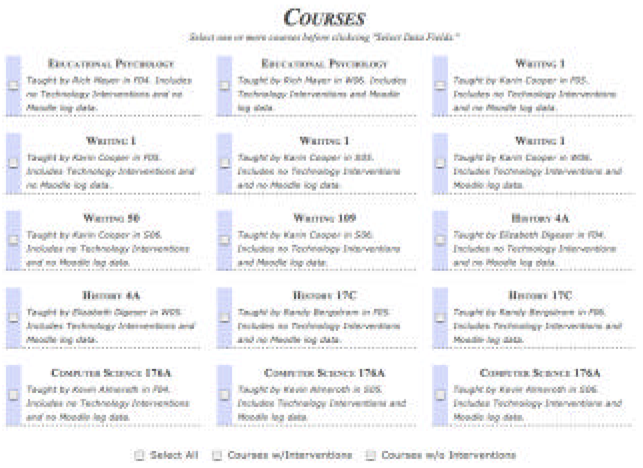
The web interface we designed led to the eventual name of the distribution software: the Data Café. The application uses a menu abstraction for data selection, similar menus found in a café or other eating establishment. Upon requesting data, the user sees a menu of courses for which data is available. Figure 1 has a screenshot of this menu. Once the user selects a set of courses to download data from, they select which fields of data to download. These sources of data are anything from specific registrar data fields, to answers to questions from questionnaires devised for the research project. Once the user chooses the appropriate data fields, the user can preview their data selection and see if it seems appropriate for their needs. Figure 2 shows this preview. After the user verifies that the data is acceptable, they can download the data into an appropriate application file format. Currently, the Data Café supports the following file formats:

1) *Microsoft Excel* – The data is in a tab-delimited format, which is directly importable into the application.
2) *SPSS Text File* – This format is not the binary version directly supported by SPSS, but rather a command file that imports the data as well as field and value labels.
3) *CSV* – The comma separated values format is supported by a wide range of applications included Excel, SPSS, and most RDBMS systems as well.

Evaluation

The purpose of the evaluation is twofold: (1) to determine under what conditions our schema design outperforms a typical design, and (2) to show that our schema design scales better than the typical database design for a statistical database. What follows outlines the evaluation setup, the evaluation process, the results of the evaluation, and a brief discussion of the implications of the results.

Setup



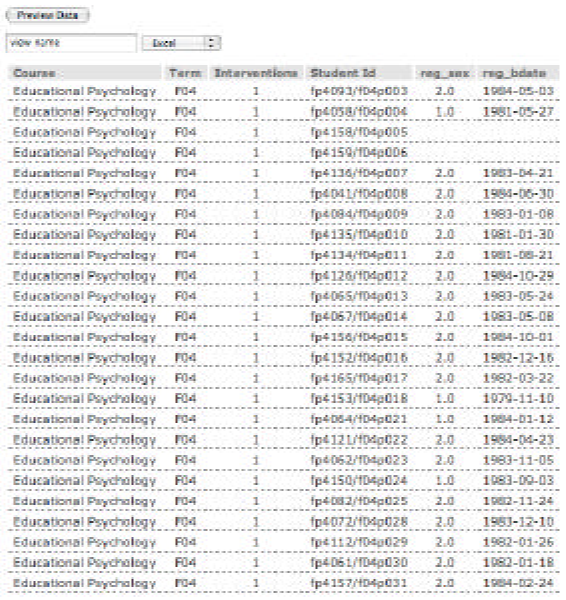
COURSES

Select one or more courses before clicking "Select Data Fields."

Educational Psychology Taught by Rich Mayer in F04. Includes no Technology Interventions and no Moodle log data.	**Educational Psychology** Taught by Rich Mayer in W06. Includes Technology Interventions and Moodle log data.	**Writing 1** Taught by Karin Cooper in F05. Includes no Technology Interventions and no Moodle log data.
Writing 1 Taught by Karin Cooper in F05. Includes Technology Interventions and no Moodle log data.	**Writing 1** Taught by Karin Cooper in S05. Includes no Technology Interventions and no Moodle log data.	**Writing 1** Taught by Karin Cooper in W06. Includes Technology Interventions and Moodle log data.
Writing 50 Taught by Karin Cooper in S06. Includes no Technology Interventions and no Moodle log data.	**Writing 109** Taught by Karin Cooper in S06. Includes no Technology Interventions and Moodle log data.	**History 4A** Taught by Elizabeth Digeser in F04. Includes no Technology Interventions and no Moodle log data.
History 4A Taught by Elizabeth Digeser in W05. Includes Technology Interventions and Moodle log data.	**History 17C** Taught by Randy Bergstrom in F05. Includes no Technology Interventions and no Moodle log data.	**History 17C** Taught by Randy Bergstrom in F06. Includes Technology Interventions and Moodle log data.
Computer Science 176A Taught by Kevin Almeroth in F04. Includes no Technology Interventions and no Moodle log data.	**Computer Science 176A** Taught by Kevin Almeroth in S05. Includes Technology Interventions and Moodle log data.	**Computer Science 176A** Taught by Kevin Almeroth in S06. Includes Technology Interventions and Moodle log data.

☐ Select All ☐ Courses w/Interventions ☐ Courses w/o Interventions

Figure 1: Screenshot of the Data Café Course Menu.



Course	Term	Interventions	Student Id	reg_sex	reg_bdate
Educational Psychology	F04	1	fp4093/f04p003	2.0	1984-05-03
Educational Psychology	F04	1	fp4058/f04p004	1.0	1981-05-27
Educational Psychology	F04	1	fp4158/f04p005		
Educational Psychology	F04	1	fp4159/f04p006		
Educational Psychology	F04	1	fp4136/f04p007	2.0	1983-04-21
Educational Psychology	F04	1	fp4041/f04p008	2.0	1984-06-30
Educational Psychology	F04	1	fp4084/f04p009	2.0	1983-01-08
Educational Psychology	F04	1	fp4135/f04p010	2.0	1981-01-30
Educational Psychology	F04	1	fp4134/f04p011	2.0	1981-08-21
Educational Psychology	F04	1	fp4126/f04p012	2.0	1984-10-29
Educational Psychology	F04	1	fp4065/f04p013	2.0	1983-05-24
Educational Psychology	F04	1	fp4067/f04p014	2.0	1983-05-08
Educational Psychology	F04	1	fp4156/f04p015	2.0	1984-10-01
Educational Psychology	F04	1	fp4152/f04p016	2.0	1982-12-16
Educational Psychology	F04	1	fp4165/f04p017	2.0	1982-03-22
Educational Psychology	F04	1	fp4153/f04p018	1.0	1979-11-10
Educational Psychology	F04	1	fp4064/f04p021	1.0	1984-01-12
Educational Psychology	F04	1	fp4121/f04p022	2.0	1984-04-23
Educational Psychology	F04	1	fp4062/f04p023	2.0	1983-11-05
Educational Psychology	F04	1	fp4150/f04p024	1.0	1983-09-03
Educational Psychology	F04	1	fp4082/f04p025	2.0	1982-11-24
Educational Psychology	F04	1	fp4072/f04p028	2.0	1983-12-10
Educational Psychology	F04	1	fp4112/f04p029	2.0	1982-01-26
Educational Psychology	F04	1	fp4061/f04p030	2.0	1982-01-18
Educational Psychology	F04	1	fp4157/f04p031	2.0	1984-02-24

Figure 2: Screenshot of data preview.

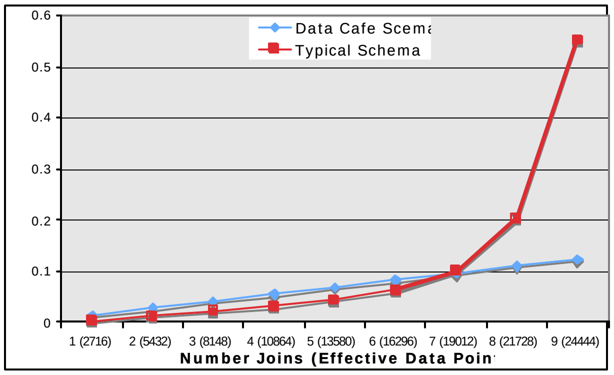
We performed the following evaluation on a Dell Power Edge 1800 Server with two 3.2 GHz Xeon processors. The test machine has 2 GB of RAM and 250 GB of hard drive space. The server is running version 2006.1 of the Gentoo Linux distribution with the 2.6.14-hardend-r1 version of the Linux Kernel compiled with support for hyper-threading, resulting in the kernel using four processors. The Data Café data is stored using version 8.0.8 of the PostgreSQL RDBMS. We ran all scripts using Ruby version 1.8.4 and the 0.7.1 Postgres extension for communication with the RDBMS (this extension is the C extension, not the pure Ruby version to hide as much as possible the overhead due to the scripting language).

Process

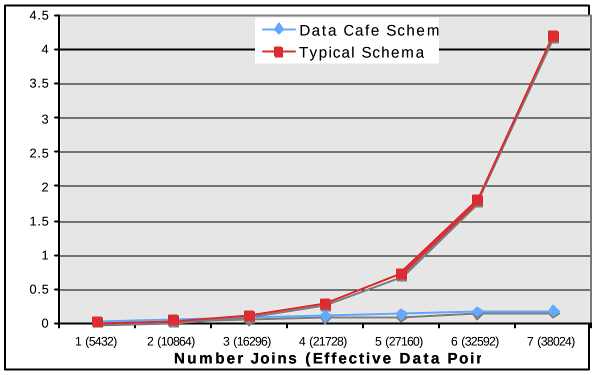
Since the goal is to measure how our design performs with respect to a typical design, we designed a very simple test to address the two goals of the evaluation. We chose to test the performance of simple database queries as we changed two separate dimensions of database growth. The first dimension was the number of data sources queried. These sources map to tables, and therefore, require a typical design to perform database join operations for each source included in a query. The second dimension is the number of students, or rows in the database for a typical database schema. Increasing this dimension also causes performance degradation in a typical schema that becomes more pronounced as the number of tables is increased. Because our schema has a fixed number of tables, increasing both the number of students and the number of sources should have less of an effect on query performance.

In order to vary these two dimensions, we performed a series of queries against the two competing schemas for the same amount of effective data. That is, the query to each design should return the same number of data points, but not necessarily the same number of rows. We used data stored from our database to create a snapshot of the data separated into four sources, and therefore, tables containing data from the registrar, from pre- and post-questionnaires, and the final grades for each course. For this simple test we chose to query the typical database table containing final grades. We chose this data source because it contains two columns: final grade as a percentage and final letter grade. Also, the amount of effective data from a number of joins fits within the effective data we currently have from our database schema design. For the original 1358 students in our database, there are 2×1358, or 2716, pieces of effective data. An equivalent query on our database schema would request this number of rows.

We measured how much time each query took for equivalent effective data points in each of the database schema designs. The "effective data points" is defined as the number of joins times the number of columns times the number of rows. We then varied the number of joins for the typical design and increased the number of rows for our database schema design. For example, one test measured the time needed to join the final grades to itself. The

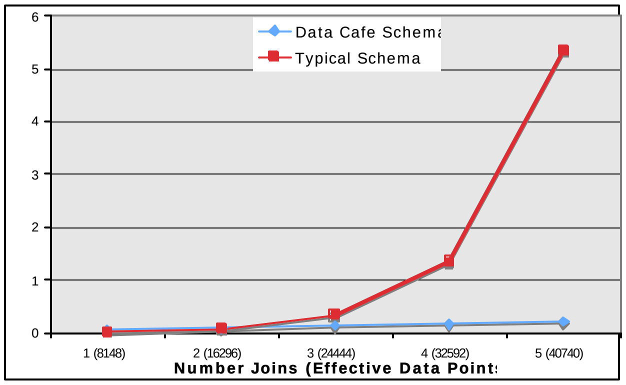


(a)



(b)

number of effective data points for this query is $2 \times 2 \times 1358$, or 5432, effective data points. We queried the same number of data points from our database schema design. We ran each of these tests 100 times and recorded the average for 1 to as many as 10 joins of the final grades table. We then doubled and tripled the number of students in the final grades table and ran the same tests again.



(c)

Figure 3. Results for (a) 1358, (b) 2716, and (b) 4074 students (rows) respectively.

Results

Figure 3 illustrates the final results of all the tests we ran. The goal was to test whether our decomposition table schema scales better than a typical table schema. This figure includes three graphs for each of the tests run. The y-axis for each graph represents the average time take over 100 iterations of each test, measured in seconds. The x-axis for each graph represents both the number of joins performed for the typical schema and the total number of effective data points in parentheses. Each graph contains two lines that illustrate the change in performance for each number of joins, and therefore, effective data points are varied. One line represents the times measured for our data schema and the other line for the typical schema. The three graphs show similar trends. At the leftmost position in the graph, where we have only one join and the fewest number of effective data points, the typical schema exhibits slightly better performance because there is no increase in the data processed caused by joins. As the number of joins and effective data points increases, the typical schema exhibits worse performance than our schema. The difference in performance continues to worsen as the number of joins increases.

The results show that as we increase the number of effective data points by increasing the numb er of sources, the typical database schema query time grows exponentially. For our database schema the query time grows linearly because the number of sources does not require a join operation. Furthermore, as the number of rows in the typical database schema grows, the performance problem is exacerbated. For example, in the graph for 1358 rows, the breakeven point for performance for either schema occurs at 7 sources. For the tests involving 2716, however, the breakeven point occurs at just 3 joins. Therefore, as the number of rows in the database increases, the breakeven point decreases.

Discussion

The implication of these results is that the typical database schema does not scale well. If there is an increase in either the number of sources (and therefore number of tables) or the number of students in any table increases, the performance of data queries does not scale well. We do not, however, observe this same effect with our schema design. There is only a linear effect on the performance based on the total number of data points. We therefore conclude that our design scales better as the number of students, or rows increase as well as the number of data sources.

Conclusions

In this paper we have presented the design and rationale for the Data Café research data management application. We have made several contributions in this work. First, we have designed a database schema that will scale and maintain performance even as data and data sources grow. The evaluation section shows that using a more typical source/table approach can lead to degradation in performance as the sources, and therefore table joins, increase. Second, we provided a novel interface using a menu paradigm that allows for the selection of the exact data needed by each researcher. Third, the data is downloadable in a variety of application file formats that allow a diverse group of researcher to use the tools they are comfortable using without having to convert data on their own. Fourth, we have created an automatic codebook that is maintainable and viewable on line in the most up-to-date fashion. These contributions have fostered research among a diverse team of researchers and allowed them to analyze the data as they see fit without worrying about the age of the data or how to convert it to and from a multitude of application formats.

In the future, we will continue to update the Data Café to include more features necessary to support ongoing research. First, more security features are necessary to allow the Data Café to scale to larger teams of researchers. For example, while we currently restrict access to the data using passwords. We do not, however, have the concept of roles built into the application. The usage of roles will allow for finer grain control over who can access the data and who can modify it. Second, we are finding it necessary to version the data itself. An example of this need relates to the usage of accumulative grade point averages from the registrar. The studies we have done so far look at this piece of information as it relates to the current time the courses are given. These averages, though, change over time. For each course, we may want the most up-to-date average at the time the student takes the course. To prevent overwriting previously recorded averages for the same student enrolled in multiple classes from the study, Data Café needs versioning. Implementing versions requires the addition of a version field in the database and some user interface additions that allow the user to specify which version they prefer.

On a final note, while we originally designed the Data Café for educational researcher, we believe its utility goes beyond any specific field of study and can be modified for use with other areas of research. A more general-purpose version of the software is being planned.

References

Abadi, D. (2007). Column -Stores For Wide and Sparse Data. *Third Biennial Conference on Innovative Data Systems Research, 2007,* Morgan Kaufmann Publishers, Asilomar, CA. 292-297.

Aydin, G., Altay, H., Aktas, M. S., Necati Aysan, M. N., Fox, G., Ikibas, C., Kim, J., Kaplan, A., Topcu, A. E., Pierce, M., Yildiz, B., and Balsoy, O. (2003). Online Knowledge Center Tools for Metadata Management. *High Performance Computing Modernization Program Users Group Meeting, 2003,* Department of Defense, Seattle, WA.

Bulger, M., Mayer, R., and Almeroth, K. (2006). Engaged by Design: Using Simulations to Promote Active Learning. *World Conference on Educational Multimedia, Hypermedia and Telecommunications, 2006,* Chesapeake, VA. 1770-1777.

Copeland, G. P., and Khoshafian, S. N. (1985). A decomposition storage model. *International Conference on Management of Data, 1985*, ACM Press, New York, NY. 268-279

Greif, I. and Sarin, S. (1986). Data sharing in group work. *Conference on Computer-Supported Cooperative Work, 1986*, ACM Press, New York, NY. 175-183.

Harizopoulos, S., Liang, V., Abadi, D. J., and Madden, S. (2006). Performance tradeoffs in read-optimized databases, *International Conference on Very Large Data Bases, 2006*, VLDB Endowment, Seoul, Korea. 487-498.

Kent, J-J., and Schuerhoff, M. (1997). Some thoughts about a metadata management system, *International Conference on Scientific and Statistical Database Management, 1997,* IEEE Computer Society, Olympia, Washington. 174 – 185.

Khoshafian, S., Copeland, G. P., Jagodis, T., Boral, H., and Valduriez, P. (1987). A Query Processing Strategy for the Decomposed Storage Model, *International Conference on Data Engineering, 1987*, IEEE Computer Society, Washington, DC, 636-643.

Kolaitis, P. G. (2005). Schema mappings, data exchange, and metadata management. *Symposium on Principles of Database Systems, 2005,* ACM Press, Baltimore, MA. 61 – 75.

Mayer, R., Almeroth, K., Bimber, B., Chun, D., Knight, A. and Campbell, A. (2006). Technology Comes to College: Understanding the Cognitive Consequences of Infusing Technology in College Classrooms, *Educational Technology*, vol. 46, num. 2, 2006. 48-53.

Mayer, R., Stull, A., Campbell, J., Almeroth, K., Bimber, B., Chun, D., and Knight, A. (2006). Some Shortcomings of Soliciting Students' Self-Reported SAT Scores, *American Educational Research Association Annual Conference (AERA), 2006*, San Francisco, CA.

McCarthy, J. L. (1982). Metadata Management for Large Statistical Databases, *International Conference on Very Large Data Bases, 1982*, Morgan Kaufmann Publishers, San Francisco, CA, 234-243.

Microsoft Inc. (2007). http://www.microsoft.com/sql/

Müller, R., Stöhr, T., and Rahm, E. (1999). An Integrative and Uniform Model for Metadata Management in Data Warehousing Environments. *Workshop on Design and Management of Data Warehouses, 1999*, ACM Press, Heidelberg, Germany.

PostgreSQL, (2007). http://www.postgresql.org.

SAS, (2007). SAS Institute Inc. http://www.sas.com.

Shoshani, A. (1982). Statistical Databases: Characteristics, Problems, and some Solutions. *International Conference on Very Large Data Bases*, *1982*. Morgan Kaufmann Publishers, San Francisco, CA. 208-222.

SPSS Inc. (2007). http://www.spss.com.

Stonebraker, M., Abadi, D., Batkin, A., Chen, X., Cherniack, M., Ferreira, M., Lau, E., Lin, A., Madden, S., O'Neil, E., O'Neil, P., Rasin, A., Tran, N. and Zdonik, S. (2005). C-Store: A Column Oriented DBMS. *International Conference on Very Large Databases, 2005*, Morgan Kaufmann Publishers. 553-564.

Vaduva, A., and Dittrich, K. R. (2001). Metadata Management for Data Warehousing: Between Vision and Reality, *International Database Engineering & Applications Symposium, 2001*, IEEE Computer Society, Montreal, Canada. 129 – 135.

Widom, J. (1995). Research Problems in Data Warehousing. *International Conference on Information and Knowledge Management, 1995,* ACM Press, Kansas City, MO. 25 – 30.