A Brief Tutorial on Database Queries, Data Mining, and OLAP

Lutz Hamel

Department of Computer Science and Statistics

University of Rhode Island

Tyler Hall

Kingston, RI 02881

Tel: (401) 480-9499

Fax: (401) 874-4617

Email: hamel@cs.uri.edu

A Brief Tutorial on Database Queries, Data Mining, and OLAP

Lutz Hamel, University of Rhode Island, USA

INTRODUCTION

Modern, commercially available relational database systems now routinely include a cadre of data retrieval and analysis tools. Here we shed some light on the interrelationships between the most common tools and components included in today's database systems: query language engines, data mining components, and on-line analytical processing (OLAP) tools. We do so by pair-wise juxtaposition which will underscore their differences and highlight their complementary value.

BACKGROUND

Today's commercially available relational database systems now routinely include tools such as SQL database query engines, data mining components, and OLAP (Craig, Vivona, & Bercovitch, 1999; Oracle, 2001; Scalzo, 2003; Seidman, 2001). These tools allow developers to construct high powered business intelligence (BI) applications which are not only able to retrieve records efficiently but also support sophisticated analyses such as customer classification and market segmentation. However, with powerful tools so tightly integrated with the database technology understanding the differences between these tools and their comparative advantages and disadvantages becomes critical for

effective application development. From the practitioner's point of view questions like the following often arise:

- Is running database queries against large tables considered data mining?
- Can data mining and OLAP be considered synonymous?
- Is OLAP simply a way to speed up certain SQL queries?

The issue is being complicated even further by the fact that data analysis tools are often implemented in terms of data retrieval functionality. Consider the data mining models in the Microsoft SQL server which are implemented through extensions to the SQL database query language (e.g. predict join) (Seidman, 2001) or the proposed SQL extensions to enable decision tree classifiers (Sattler & Dunemann, 2001). OLAP cube definition is routinely accomplished via the data definition language (DDL) facilities of SQL by specifying either a star or snowflake schema (Kimball, 1996).

MAIN THRUST OF THE CHAPTER

The following sections contain the pair wise comparisons between the tools and components considered in this chapter.

Database Queries vs. Data Mining

Virtually all modern, commercial database systems are based on the relational model formalized by Codd in the 60s and 70s (Codd, 1970) and the SQL language (Date, 2000) which allows the user to efficiently and effectively manipulate a database. In this model a database table is a representation of a mathematical relation, that is, a set of items that share certain characteristics or attributes. Here, each table column represents an

attribute of the relation and each record in the table represents a member of this relation. In relational databases the tables are usually named after the kind of relation they represent. Figure 1 is an example of a table that represents the set or relation of all the customers of a particular store. In this case the store tracks the total amount of money spent by its customers.

Figure 1: A relational database table representing customers of a store.

Id	*Name*	*ZIP*	*Sex*	*Age*	*Income*	*Children*	*Car*	*Total Spent*
5	Peter	05566	M	35	$40,000	2	Mini Van	$250.00
…	…	…	…	…	…	…	…	…
22	Maureen	04477	F	26	$55,000	0	Coupe	$50.00

Relational databases do not only allow for the creation of tables but also for the manipulation of the tables and the data within them. The most fundamental operation on a database is the query. This operation enables the user to retrieve data from database tables by asserting that the retrieved data needs to fulfill certain criteria. As an example, consider the fact that the store owner might be interested in finding out which customers spent more than $100 at the store. The following query returns all the customers from the above customer table that spent more than $100:

```
SELECT * FROM CUSTOMER_TABLE WHERE TOTAL_SPENT > $100;
```

This query returns a list of all instances in the table where the value of the attribute *Total Spent* is larger than $100. As this example highlights, queries act as filters that allow the user to select instances from a table based on certain attribute values. It does not matter how large or small the database table is, a query will simply return all the instances from a table that satisfy the attribute value constraints given in the query. This straightforward approach to retrieving data from a database has also a drawback. Assume for a moment that our example store is a large store with tens of thousands of customers (perhaps an online store). Firing the above query against the customer table in the database will most likely produce a result set containing a very large number of customers and not much can be learned from this query except for the fact that a large number of customers spent more than $100 at the store. Our innate analytical capabilities are quickly overwhelmed by large volumes of data.

This is where differences between querying a database and mining a database surface. In contrast to a query which simply returns the data that fulfills certain constraints, data mining constructs models of the data in question. The models can be viewed as high level summaries of the underlying data and are in most cases more useful than the raw data, since in a business sense they usually represent understandable and actionable items (Berry & Linoff, 2004). Depending on the questions of interest, data mining models can take on very different forms. They include decision trees and decision rules for classification tasks, association rules for market basket analysis, as well as clustering for market segmentation among many other possible models. Good overviews of current data mining techniques and models can be found in (Berry & Linoff, 2004;

Han & Kamber, 2001; Hand, Mannila, & Smyth, 2001; Hastie, Tibshirani, & Friedman, 2001).

To continue our store example, in contrast to a query, a data mining algorithm that constructs decision rules might return the following set of rules for customers that spent more than $100 from the store database:

```
IF AGE > 35 AND CAR = MINIVAN THEN TOTAL SPENT > $100

                        OR

IF SEX = M AND ZIP = 05566 THEN TOTAL SPENT > $100
```

These rules are understandable because they summarize hundreds, possibly thousands, of records in the customer database and it would be difficult to glean this information off the query result. The rules are also actionable. Consider that the first rule tells the store owner that adults over the age of 35 that own a mini van are likely to spend more than $100. Having access to this information allows the store owner to adjust the inventory to cater to this segment of the population, assuming that this represents a desirable cross-section of the customer base. Similar with the second rule, male customers that reside in a certain ZIP code are likely to spend more than $100. Looking at census information for this particular ZIP code the store owner could again adjust the store inventory to also cater to this population segment presumably increasing the attractiveness of the store and thereby increasing sales.

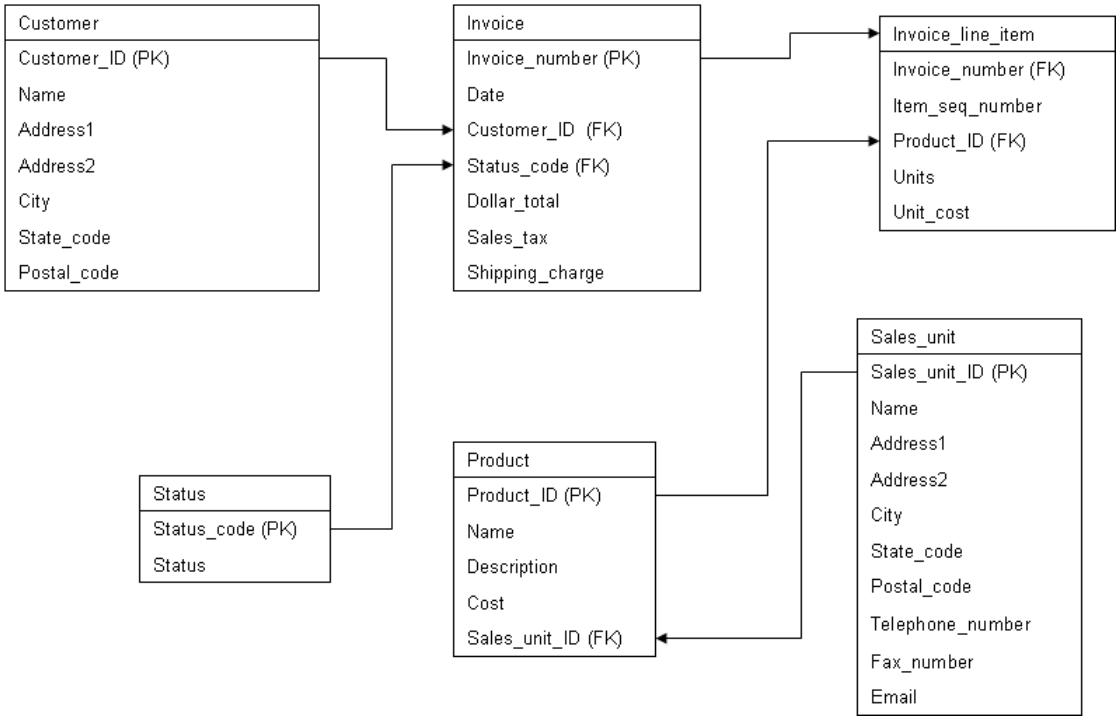
As we have shown, the fundamental difference between database queries and data mining is the fact that in contrast to queries data mining does not return raw data that satisfies certain constraints, but returns models of the data in question. These models are

attractive because in general they represent understandable and actionable items. Since no such modeling ever occurs in database queries we do not consider running queries against database tables as data mining, it does not matter how large the tables are.

Database Queries vs. OLAP

In a typical relational database queries are posed against a set of normalized database tables in order to retrieve instances that fulfill certain constraints on their attribute values (Date, 2000). The normalized tables are usually associated with each other via primary/foreign keys. For example, a normalized database of our store with multiple store locations or sales units might look something like the database given in Figure 2. Here, PK and FK indicate primary and foreign keys, respectively.

Figure 2: Normalized database schema for a store.



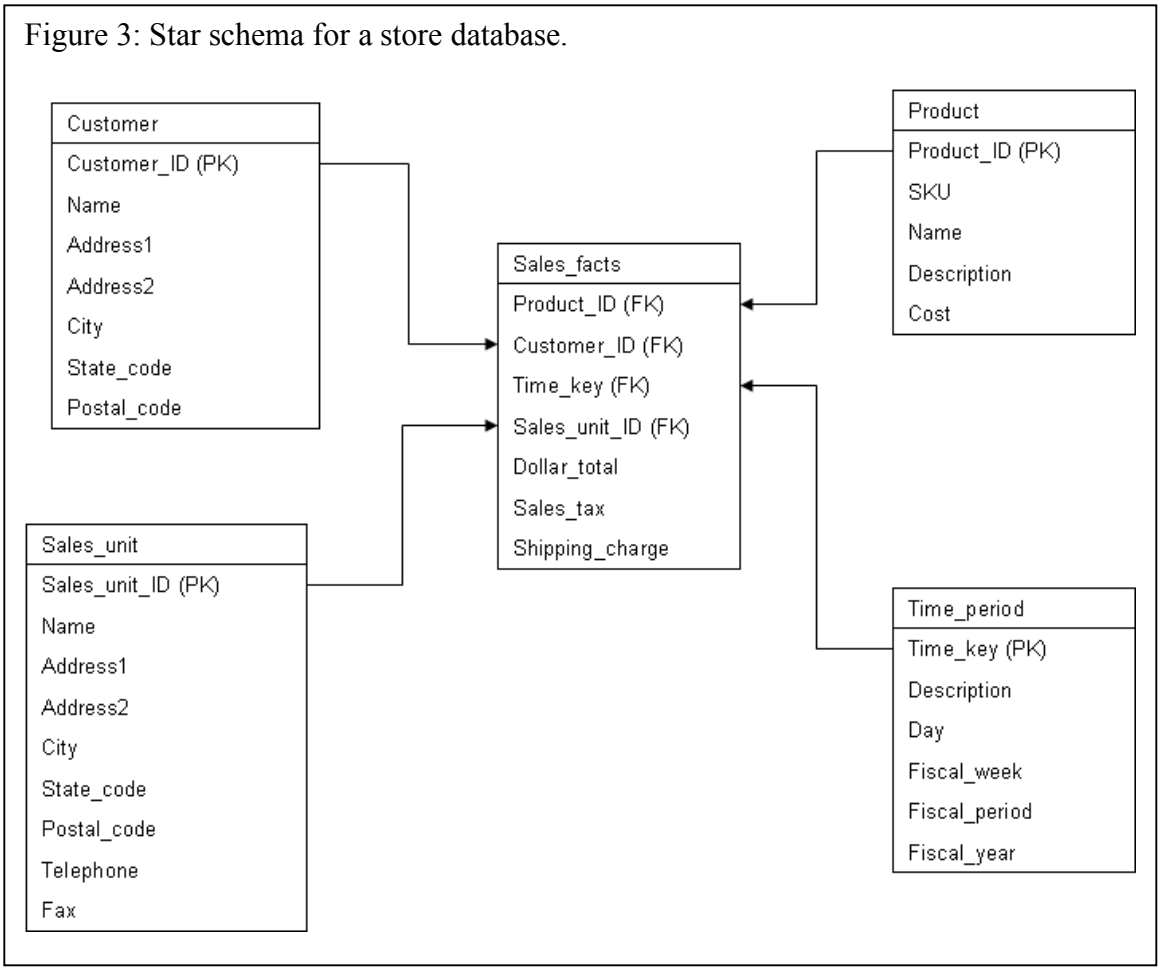
From a user perspective it might be interesting to ask some of the following questions:

- How much did sales unit A earn in January?
- How much did sales unit B earn in February?
- What was their combined sales amount for the first quarter?

Even though it is possible to extract this information with standard SQL queries from our database, the normalized nature of the database makes the formulation of the appropriate SQL queries very difficult. Furthermore, the query process is likely to be slow due to the fact that it must perform complex joins and multiple scans of entire database tables in order to compute the desired aggregates.

By rearranging the database tables in a slightly different manner and using a process called pre-aggregation or *computing cubes* the above questions can be answered with much less computational power enabling a real time analysis of aggregate attribute values – OLAP (Craig et al., 1999; Kimball, 1996; Scalzo, 2003). In order to enable OLAP, the database tables are usually arranged into a star schema where the inner-most table is called the fact table and the outer tables are called dimension tables. Figure 3 shows a star schema representation of our store organized along the main dimensions of the store business: customers, sales units, products, and time.

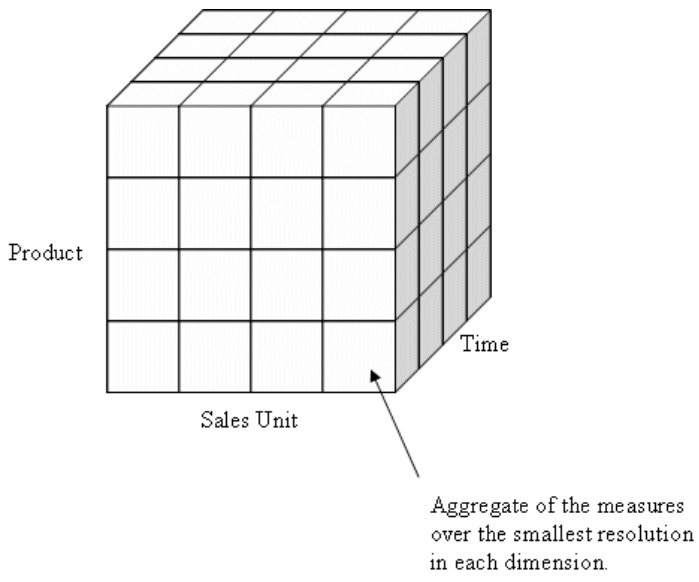
Figure 3: Star schema for a store database.



The dimension tables give rise to the dimensions in the pre-aggregated data cubes. The fact table relates the dimensions to each other and specifies the measures which are to be aggregated. Here the measures are "dollar_total", "sales_tax", and "shipping_charge". Figure 4 shows a three-dimensional data cube pre-aggregated from the star schema in Figure 3 (in this cube we ignored the customer dimension, since it is difficult to illustrate four-dimensional cubes). In the cube building process the measures are aggregated along the smallest unit in each dimension giving rise to small pre-aggregated segments in a cube.

Data cubes can be seen as a compact representation of pre-computed query results[i]. Essentially, each segment in a data cube represents a pre-computed query result to a particular query within a given star schema. The efficiency of cube querying allows the user to interactively move from one segment in the cube to another enabling the inspection of query results in real time. Cube querying also allows the user to group and ungroup segments, as well as project segments onto given dimensions. This corresponds to such OLAP operations as roll-ups, drill-downs, and slice-and-dice, respectively (Gray, Bosworth, Layman, & Pirahesh, 1997). These specialized operations in turn provide answers to the kind of questions mentioned above.

Figure 4: A three-dimensional data cube.



As we have seen, OLAP is enabled by organizing a relational database in a way that allows for the pre-aggregation of certain query results. The resulting data cubes hold the pre-aggregated results giving the user the ability to analyze these aggregated results in

real time using specialized OLAP operations. In a larger context we can view OLAP as a methodology for the organization of databases along the dimensions of a business making the database more comprehensible to the end user.

Data Mining vs. OLAP

Is OLAP data mining? As we have seen, OLAP is enabled by a change to the data definition of a relational database in such a way that it allows for the pre-computation of certain query results. OLAP itself is a way to look at these pre-aggregated query results in real time. However, OLAP itself is still simply a way to evaluate queries which is different from building models of the data as in data mining. Therefore, from a technical point of view we cannot consider OLAP to be data mining. Where data mining tools model data and return actionable rules, OLAP allows users to compare and contrast measures along business dimensions in real time.

It is interesting to note, that recently a tight integration of data mining and OLAP has occurred. For example, Microsoft SQL Server 2000 not only allows OLAP tools to access the data cubes but also enables its data mining tools to mine data cubes (Seidman, 2001).

FUTURE TRENDS

Perhaps the most important trend in the area of data mining and relational databases is the liberation of data mining tools from the “single table requirement”. This new breed of data mining algorithms is able to take advantage of the full relational structure of a relational database obviating the need of constructing a single table that

contains all the information to be used in the data mining task (Dézeroski & Lavraéc, 2001). This allows for data mining tasks to be represented naturally in terms of the actual database structures, e.g. (Yin, Han, Yang, & Yu, 2004), and also allows for a natural and tight integration of data mining tools with relational databases.

CONCLUSION

Modern, commercially available relational database systems now routinely include a cadre of data retrieval and analysis tools. Here, we briefly described and contrasted the most often bundled tools: SQL database query engines, data mining components, and OLAP tools. Contrary to many assertions in the literature and business press, performing queries on large tables or manipulating query data via OLAP tools is not considered data mining due to the fact that no data modeling occurs in these tools. On the other hand, these three tools complement each other and allow developers to pick the tool that is right for their application: queries allow ad hoc access to virtually any instance in a database; data mining tools can generate high-level, actionable summaries of data residing in database tables; and OLAP allows for real-time access to pre-aggregated measures along important business dimensions. In this light it does not seem surprising that all three tools are now routinely bundled.

REFERENCES

Berry, M. J. A., & Linoff, G. S. (2004). *Data Mining Techniques : For Marketing, Sales, and Customer Relationship Management* (2nd ed.): John Wiley & Sons.

Codd, E. F. (1970). A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM, 13*(6), 377-387.

Craig, R. S., Vivona, J. A., & Bercovitch, D. (1999). *Microsoft Data Warehousing*: John Wiley & Sons.

Date, C. J. (2000). *An introduction to database systems* (7th ed.). Reading, Mass.: Addison-Wesley.

Dézeroski, S., & Lavraéc, N. (2001). *Relational data mining*. Berlin ; New York: Springer.

Gray, J., Bosworth, A., Layman, A., & Pirahesh, H. (1997). Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. *Data Mining and Knowledge Discovery, 1*(1), 29-53.

Han, J., & Kamber, M. (2001). *Data mining : concepts and techniques*. San Francisco: Morgan Kaufmann Publishers.

Hand, D. J., Mannila, H., & Smyth, P. (2001). *Principles of data mining*. Cambridge, Mass.: MIT Press.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2001). *The elements of statistical learning : data mining, inference, and prediction*. New York: Springer.

Kimball, R. (1996). *The data warehouse toolkit : practical techniques for building dimensional data warehouses*. New York: John Wiley & Sons.

Oracle. (2001). Oracle 9i Data Mining, *Oracle White Paper*.

Pendse, N. (2001). *Multidimensional data structures*, from http://www.olapreport.com/MDStructures.htm

Sattler, K., & Dunemann, O. (2001, November 5-10). *SQL Database Primitives for Decision Tree Classifiers.* Paper presented at the 10th International Conference on Information and Knowledge Management, Atlanta, Georgia.

Scalzo, B. (2003). *Oracle DBA guide to data warehousing and star schemas*. Upper Saddle River, N.J.: Prentice Hall PTR.

Seidman, C. (2001). *Data Mining with Microsoft SQL Server 2000 Technical Reference*: Microsoft Press.

Yin, X., Han, J., Yang, J., & Yu, P. S. (2004). *CrossMine: Efficient Classification Across Multiple Database Relations.* Paper presented at the 20th International Conference on Data Engineering (ICDE 2004), Boston, MA, USA.

TERMS AND THEIR DEFINITION

SQL: Structured Query Language - SQL is a standardized programming language for defining, retrieving, and inserting data objects in relational databases.

OLAP: On-Line Analytical Processing - a category of applications and technologies for collecting, managing, processing and presenting multidimensional data for analysis and management purposes. (Source: http://www.olapreport.com/glossary.htm)

Star Schema: A database design that is based on a central detail fact table linked to surrounding dimension tables. Star schemas allow access to data using business terms and perspectives. (Source: http://www.ds.uillinois.edu/glossary.asp)

Normalized Database: A database design that arranges data in such a way that it is held at its lowest level avoiding redundant attributes, keys, and relationships.

(Source: http://www.oranz.co.uk/glossary_text.htm)

Query: This term generally refers to databases. A query is used to retrieve database records that match certain criteria.

(Source: http://usa.visa.com/business/merchants/online_trans_glossary.html)

Business Intelligence: Business intelligence (BI) is a broad category of technologies that allows for gathering, storing, accessing and analyzing data to help business users make better decisions. (Source: http://www.oranz.co.uk/glossary_text.htm)

Data Cubes: Also known as OLAP cubes. Data stored in a format that allows users to perform fast multi-dimensional analysis across different points of view. The data is often sourced from a data warehouse and relates to a particular business function.

(Source: http://www.oranz.co.uk/glossary_text.htm)

Figures 2 and 3 are based on Figures 3.2 and 3.3 from (Craig et al., 1999), respectively.

---

[i] Another interpretation of data cubes is as an effective representation of multidimensional data along the main business dimensions (Pendse, 2001).