

# 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 online analytical processing (OLAP) tools. We do so by pairwise 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

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 1960s and 1970s (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.

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 storeowner might be interested in finding out which customers spent more than \$100 at the store. The following query

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

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

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), and 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 storeowner that adults over the age of 35 that own a minivan are likely to spend more than \$100. Having access to this information allows the storeowner 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 storeowner 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.

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

3 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: [www.igi-global.com/chapter/database-queries-data-mining-olap/10619](http://www.igi-global.com/chapter/database-queries-data-mining-olap/10619)

## Related Content

---

### Group Pattern Discovery Systems for Multiple Data Sources

Shichao Zhang and Chengqi Zhang (2005). *Encyclopedia of Data Warehousing and Mining* (pp. 546-549).  
[www.irma-international.org/chapter/group-pattern-discovery-systems-multiple/10657](http://www.irma-international.org/chapter/group-pattern-discovery-systems-multiple/10657)

### Closed-Itemset Incremental-Mining Problem

Luminita Dumitriu (2005). *Encyclopedia of Data Warehousing and Mining* (pp. 150-153).  
[www.irma-international.org/chapter/closed-itemset-incremental-mining-problem/10583](http://www.irma-international.org/chapter/closed-itemset-incremental-mining-problem/10583)

### Conceptual and Systematic Design Approach for XML Document Warehouses

Vicky Nassis, R. Rajagopalapillai, Tharam S. Dillon and Wenny Rahayu (2008). *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications* (pp. 485-508).  
[www.irma-international.org/chapter/conceptual-systematic-design-approach-xml/7661](http://www.irma-international.org/chapter/conceptual-systematic-design-approach-xml/7661)

### A Porter Framework for Understanding the Strategic Potential of Data Mining for the Australian Banking Industry

Kate A. Smith and Mark S. Dale (2008). *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications* (pp. 2772-2791).  
[www.irma-international.org/chapter/porter-framework-understanding-strategic-potential/7799](http://www.irma-international.org/chapter/porter-framework-understanding-strategic-potential/7799)

### Data Warehousing and the Organization of Governmental Databases

Franklin Maxwell Haper (2008). *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications* (pp. 2427-2437).  
[www.irma-international.org/chapter/data-warehousing-organization-governmental-databases/7772](http://www.irma-international.org/chapter/data-warehousing-organization-governmental-databases/7772)