**Unit 6**

## Data Mining Query Languages

How can we integrate data mining more closely with traditional database systems, particularly querying?” Three possible ways are there.

* DMQL: A Data Mining Query Language for Relational Databases
* Integrating Data Mining with SQL Databases: OLE DB for Data Mining
* MSQL: A Query Language for Database Mining

The Data Mining Query Language (DMQL) was proposed by Han, Fu, Wang, et al. for the DBMiner data mining system. The Data Mining Query Language is actually based on the Structured Query Language (SQL).Data Mining Query Languages can be designed to support ad hoc and interactive data mining. The DMQL can work with databases and data warehouses as well. DMQL can be used to define data mining tasks.

The language adopts an SQL-like syntax, so that it can easily be integrated with the relational query language SQL. The syntax of DMQL is defined in an extended BNF grammar, where “[ ]” represents 0 or one occurrence, “{ }” represents 0 or more occurrences, and words in sans serif font represent keywords.

**Syntax for Task-Relevant Data Specification**

Here is the syntax of DMQL for specifying task-relevant data −

*use database database\_name*

***or***

*use data warehouse data\_warehouse\_name*

*in relevance to att\_or\_dim\_list*

*from relation(s)/cube(s) [where condition]*

*order by order\_list*

*group by grouping\_list*

***Example***

*Use database ABCompany\_db*

*In relevance to I .name,I.price, C.income, C.age*

*From customer C, item I, purchases P, items\_sold S*

*Where I.item\_ID=S. item.JD and S.trans\_ID =P.trans\_ID and P .custJD=C.cust\_ID and C. country –“Sri Lanka”  Group by p.data*

**Syntax for Specifying the Kind of Knowledge**

Here we will discuss the syntax for Characterization, Discrimination, Association, Classification, and Prediction.

**Characterization**

The syntax for characterization is −

*mine characteristics [as pattern\_name]*

*analyze {measure(s) }*

The analyze clause, specifies aggregate measures, such as count, sum, or count%. For example − Description describing customer purchasing habits.

*mine characteristics as customerPurchasing*

*analyze count%*

**Discrimination**

The syntax for Discrimination is −

*mine comparison [as {pattern\_name]}*

*For {target\_class } where {target\_condition }*

*{versus {contrast\_class\_i }*

*where {contrast\_condition\_i}}*

*analyze {measure(s) }*

For example, a user may define big spenders as customers who purchase items that cost $100 or more on an average; and budget spenders as customers who purchase items at less than $100 on an average. The mining of discriminant descriptions for customers from each of these categories can be specified in the DMQL as −

*mine comparison as purchaseGroups*

*for bigSpenders where avg(I.price) ≥$100*

*versus budgetSpenders where avg(I.price)< $100*

*analyze count*

**Association**

The syntax for Association is−

*mine associations [ as {pattern\_name} ]*

*{matching {metapattern} }*

For Example

*mine associations as buyingHabits*

*matching P(X:customer,W) ^ Q(X,Y) ≥ buys(X,Z)*

Where X is key of customer relation; P and Q are predicate variables; and W, Y, and Z are object variables, Such as

age(X,”20-30”) *^ inclome(X,”40-50K”) ≥ buys (X, “Computer”)*

This rule states that customers in their thirties, with an annual income of between 40K and 50K, are likely to purchase a Computer.

**Classification**

The syntax for Classification is −

*mine classification [as pattern\_name]*

*analyze classifying\_attribute\_or\_dimension*

Example:

*mine classifications as classifyCustomerCreditRating*

*analyze credit\_rating*

For categorical attributes or dimensions, each value represents a class (such as low-risk, medium risk, high risk). For numeric attributes, each class defined by a range (such as 20-39, 40-59, 60-89 for age)

**Prediction**

The syntax for prediction is

*mine prediction [as pattern\_name]*

*analyze prediction\_attribute\_or\_dimension*

*{set {attribute\_or\_dimension\_i= value\_i}}*

**Syntax for Concept Hierarchy Specification**

To specify concept hierarchies, use the following syntax

*use hierarchy <hierarchy> for <attribute\_or\_dimension>*

We use different syntaxes to define different types of hierarchies such as−

- schema hierarchies

*define hierarchy time\_hierarchy on date as [date,month quarter,year]*

- set-grouping hierarchies

*define hierarchy age\_hierarchy for age on customer as*

*level1: {young, middle\_aged, senior} < level0: all*

*level2: {20, ..., 39} < level1: young*

*level3: {40, ..., 59} < level1: middle\_aged*

*level4: {60, ..., 89} < level1: senior*

- Operation-derived hierarchies

*define hierarchy age\_hierarchy for age on customer as*

*{age\_category(1), ..., age\_category(5)}*

*:= cluster(default, age, 5) < all(age)*

This statement says that hierarchy is generated by default clustering algorithm with 5 as *fan-out* value. Fan-out value defined levels in tree while generating concept hierarchy.

- Rule-based hierarchies

*define hierarchy profit\_margin\_hierarchy on item as*

*level\_1: low\_profit\_margin < level\_0: all*

*if (price - cost)< $50*

*level\_1: medium-profit\_margin < level\_0: all*

*if ((price - cost) > $50) and ((price - cost) ≤ $250))*

*level\_1: high\_profit\_margin < level\_0: all*

**Syntax for Interestingness Measures Specification**

- Interestingness measures and thresholds can be specified by the user with the statement

*with <interest\_measure\_name> threshold = threshold\_value*

For Example −

*with support threshold = 0.05*

*with confidence threshold = 0.7*

**Syntax for Pattern Presentation and Visualization Specification**

We have a syntax, which allows users to specify the display of discovered patterns in one or more forms.

*display as <result\_form>*

For Example −

*display as table*

**Full Specification of DMQL**

As a market manager of a company, you would like to characterize the buying habits of customers who can purchase items priced at no less than $100; with respect to the customer's age, type of item purchased, and the place where the item was purchased. You would like to know the percentage of customers having that characteristic. In particular, you are only interested in purchases made in Canada, and paid with an American Express credit card. You would like to view the resulting descriptions in the form of a table.

*use database AllElectronics\_db*

*use hierarchy location\_hierarchy for B.address*

*mine characteristics as customerPurchasing*

*analyze count%*

*in relevance to C.age,I.type,I.place\_made*

*from customer C, item I, purchase P, items\_sold S, branch B*

*where I.item\_ID = S.item\_ID and P.cust\_ID = C.cust\_ID and*

*P.method\_paid = "AmEx" and B.address = "Canada" and I.price ≥ 100*

*with noise threshold = 5%*

*display as table*

**Data Mining Languages Standardization**

Standardizing the Data Mining Languages will serve the following purposes −

* Helps systematic development of data mining solutions.
* Improves interoperability among multiple data mining systems and functions.
* Promotes education and rapid learning.
* Promotes the use of data mining systems in industry and society.