ISO/IEC JTC 1/SC 32 N 0568

Date: 2000-10-09

REPLACES: --

ISO/IEC JTC 1/SC 32

Data Management and Interchange

Secretariat: United States of America (ANSI)
Administered by Pacific Northwest National Laboratory on behalf of ANSI

DOCUMENT TYPE	Other Document (Open)
TITLE	SQL/MM Part 6: Data Mining Presentation
SOURCE	Friedemann Schwenkreis
PROJECT NUMBER	1.32.04.01.06.00
STATUS	Presentation at the SC 32 Tutorial 2000-10-09
REFERENCES	
ACTION ID.	FYI
REQUESTED ACTION	
DUE DATE	
Number of Pages	36
LANGUAGE USED	English
DISTRIBUTION	P & L Members SC Chair WG Conveners and Secretaries

Douglas Mann, Secretariat, ISO/IEC JTC 1/SC 32

Pacific Northwest National Laboratory *, 901 D Street, SW., Suite 900, Washington, DC, 20024-2115, United States of America

Telephone: +1 703 575 2114; Facsimile; +1 703 671 9180; E-mail: MannD@battelle.org
available from the JTC 1/SC 32 WebSite http://www.jtc1sc32.org/

*Pacific Northwest National Laboratory (PNL) administers the ISO/IEC JTC 1/SC 32 Secretariat on behalf of ANSI

SQL/MM Part 6: Data Mining

Friedemann Schwenkreis

Introduction

ISO/IEC 13249 (SQL/MM)
SQL Multimedia and Applications

- **Part 1: Framework**
- **Part 2: Full Text**
- **Part 3: Spatial**
- **Part 5: Still Image**
- **Part 6: Data Mining**

Data Mining: Ideas

- **Hidden information: Patterns**
- **Guess a pattern and try to prove it: That is statistics!**
- **Describe the kind of patterns and let the algorithms find them: That is data mining!**

Kinds of patterns

- **Frequent combinations of values (group concept).**
- **Frequent / similar sequences of values (time concept).**
- **Groups of similar records.**
- **Patterns to predict the value of a specific attribute.**
- **Patterns to identify deviations.**

Concepts I

- **Provide routines rather than a schema.**
- **Support a warehouse scenario.**
- **Support all three phases of mining:**
  - **Training: Find the patterns (model)!**
  - **Test: Test the model!**
    Does not exist for all kinds of patterns!
  - **Application: Use the model!**
    Not necessarily in software.

Concepts II

- **Metadata for mining, input data, and results.**
- **Four major data mining techniques:**
  - **Association Rules**
  - **Clustering**
  - **Classification**
  - **Regression**

The warehouse scenario

- **“Data”and “Transformations” can be arbitrarily connected.**
  Usually data flow oriented.
- **All metadata for a computation is pre-defined.**
- **Computations are usually scheduled and can be executed multiple times.**

DM_MiningData

- **Metadata for training and test**
- **Points to a table (or view)**
- **Set of fields (abstract from columns)**
  - **Name: identifier**
  - **Alias: mapping**
  - **Type: numeric / categorical / …**
  - **…**

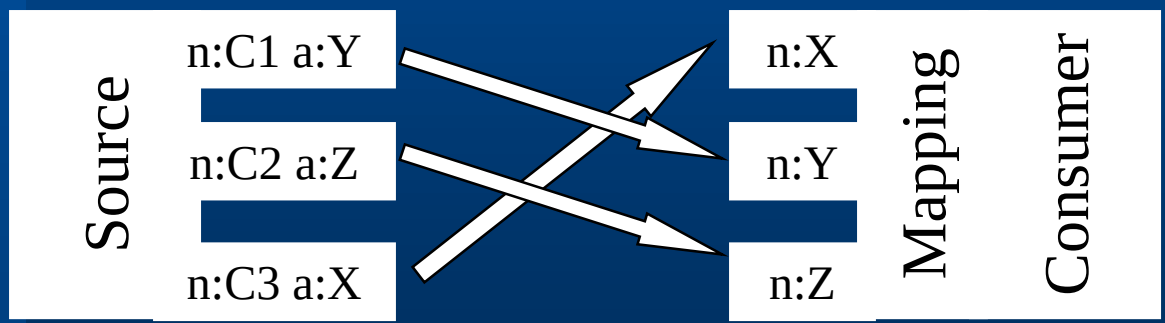
DM_MiningData: Example

```
Table: T (C1 VARCHAR(10),
          C2 INTEGER,
          C3 CHAR(5))

NEW DM_MiningData("T")
   References: table T
   Field 1: name: "C1", alias: "C1", type: CAT
   Field 2: name: "C2", alias: "C2", type: NUM
   Field 3: name: "C3", alias: "C3", type: CAT
```

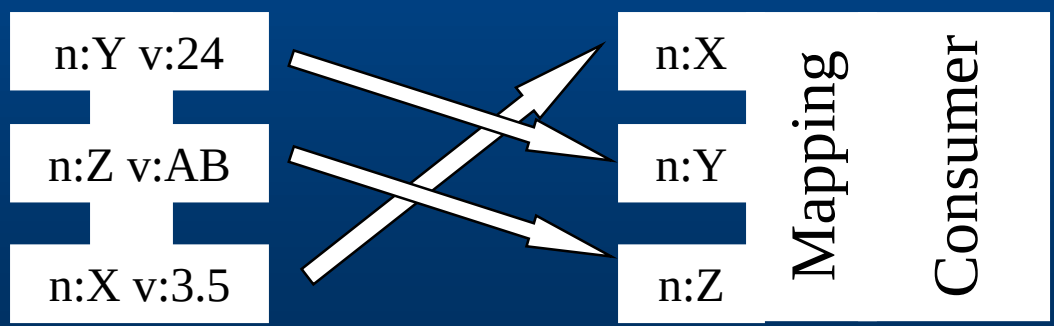
DM_MiningMapping

- **Map the fields of the data source onto fields of the consumer.**
- **Aliases of the data source fields must match the names of the consumer fields.**



DM_ApplicationData

- **Data used for application mode.**
- **Similar to a single row of data.**
- **Carries field names and values!**



Association Rules: Basics

Idea:

Find frequent combinations of values contained in groups of values.

Pattern / Model:

Sets of rules:

If value X appears in a group, then value Y appears in the same group.

Association Rules: Example

Example:

Given a set of purchases consisting of a set of items. Find the sets of items which are frequently bought together.

(Sub-)Pattern:

If a customer buys napkins, then the customer also buys orange juice.

Association Rules: Application

- **No test mode.**
  (result is not an approximation)
- **Currently, no application mode.**
  “Manual” application – future work.
- **Usually visualizers are used/needed.**

DM_RuleSettings

- **Contains a DM_MiningMapping,**
- **Grouping field specification,**
- **Minimum Support (what is frequent).**

```
((( NEW DM_RuleSettings())
       .DM_ruleUseMapping(DM_MiningMapping(…)))
       .DM_ruleSetGroup("X"))
       .DM_ruleSetMinSupport(0.05)
```

DM_RuleTask

- Represents all metadata necessary to compute association rules.
- DM_RuleSettings & DM_MiningData
- Provides a method to compute the rules.

```
( DM_defRuleTask ( DM_MiningData(…),
                   DM_RuleSettings(…)))
  .DM_buildRuleModel()
```

DM_RuleModel

- Represents the association rules.
- Provides methods to query the result.
- Provides methods to exchange rules with other tools/DBs.

```
DM_RuleModel(…).DM_getNORules()

DM_RuleModel(…).DM_expRuleModel()
```

Clustering: Basics

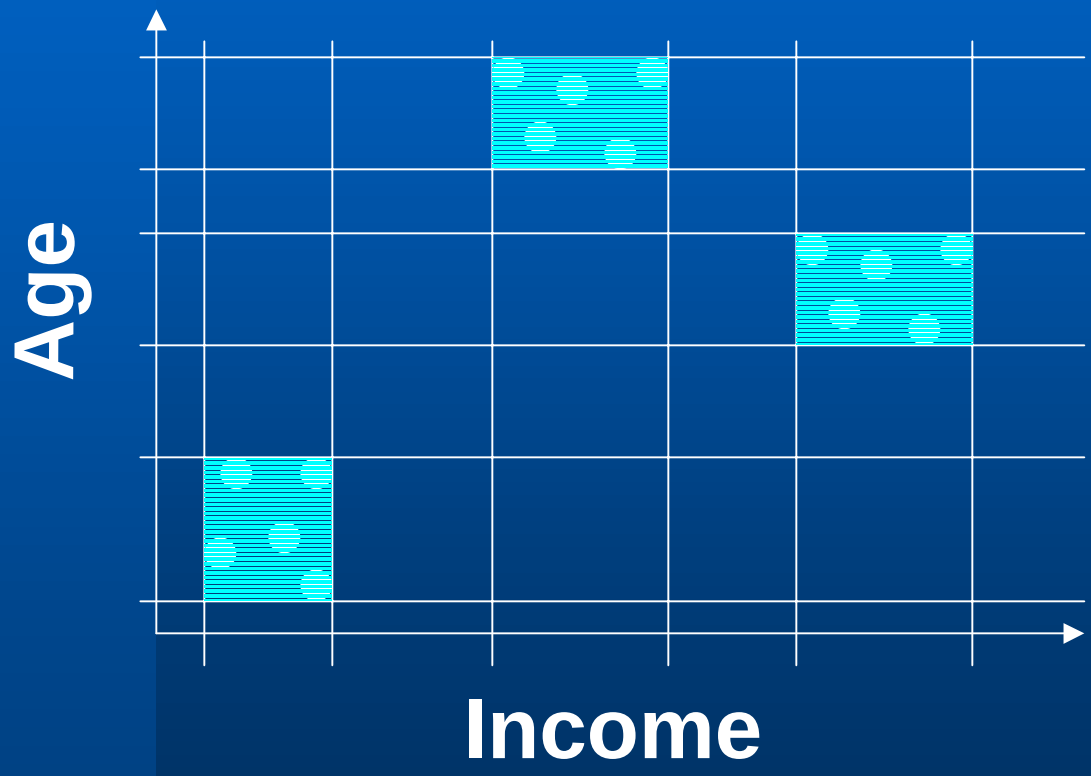
Idea:

Find groups of records which are similar / with common characteristics

Pattern / Model:

List of attribute values which characterize each group in the order of importance.

Clustering: Example



Clustering: Test/Application

- **No test mode.**
  (result is non-deterministic)
- **Application:**
  Which records belong to which group?
  Example:
  Target mailing: Which customer should get which mailing?

DM_ClusSettings

- Contains a DM_MiningMapping
- Defines the maximum number of clusters.
- …

```
(( NEW DM_ClusSettings())
       .DM_clusUseMapping(DM_MiningMapping(…)))
       .DM_ruleSetMaxNOClus(5)
```

DM_ClusTask

- Represents all metadata necessary to compute a segmentation.
- DM_ClusSettings & DM_MiningData
- Provides a method to compute the clustering model.

```
( DM_defClusTask ( DM_MiningData(…),
                   DM_ClusSettings(…)))
  .DM_buildClusModel()
```

DM_ClusModel

- Represents the clusters
- Provides methods to exchange and query the model.
- Provides a method to apply the model.

```
DM_ClusteringModel(…)
  .DM_applyClusModel(DM_ApplicationData(…))
```

DM_ClusResult

- **Result of an application of a clustering model.**
- **Abstracts from rather complex result.**
- **Provides methods to query aspects.**

```
( DM_ClusteringModel(…)
  .DM_applyClusModel(DM_ApplicationData(…)))
       .DM_getClusterID()
```

Prediction: Basics I

- **Predict the value of an attribute: Classification: type of predicted attribute is non-numeric. Regression: type of predicted attribute is numeric.**
- **Train a model based on data for which we know the values of the predicted attribute.**

Prediction: Basics II

- **Build the model such that a minimal number of values have to be provided to predict the value of the unknown attribute.**
- **The computed model can be tested with data for which we know the value of the predicted attribute.**

Prediction: Example

- **Insurance risk prediction.**
- **Train a prediction model using historical data.**
- **Reduce the number of questions we have to ask new customers.**
- **Predict the risk using the model and the customer information.**

Prediction: Test

- Test data has to be provided or derived from training data
- Test shows quality of prediction
- Detects whether the model is really applicable.

DM_ClasSettings

- Contains a DM_MiningMapping
- Defines the predicted field
- …

```
(( NEW DM_ClasSettings())
       .DM_clasUseMapping(DM_MiningMapping(…)))
       .DM_clasSetTarget("X")
```

DM_ClasTask

- **Represents all metadata necessary to compute a classification.**
- **DM_ClasSettings & DM_MiningData**
- **Provides a method to compute the classification model.**

```
( DM_defClasTask ( DM_MiningData(…),
                   DM_ClasSettings(…)))
  .DM_buildClasModel()
```

DM_ClasModel

- Represents the classification model.
- Provides methods to export and query the model.
- Provides methods to apply and test the model.

```
DM_ClasModel(…)
  .DM_testClasModel(DM_MiningData(…))
DM_ClasModel(…)
  .DM_applyClasModel(DM_ApplicationData(…))
```

DM_ClasTestResult

- **Result of a test of a classification model.**
- **Abstracts from complex test result.**
- **Provides methods to query aspects.**

```
(DM_ClasModel(…)
  .DM_testClasModel(DM_MiningData(…)))
       .DM_getClasError()
```

DM_ClasResult

- **Result of an application of a classification model.**
- **Abstracts from rather complex application result.**
- **Provides methods to query aspects.**

```
(DM_ClasModel(…)
  .DM_applyClasModel(DM_ApplicationData(…)))
       .DM_getPredClass()
```

Schedule

- **Vote on CD: 10/2000**
- **CD ballot: 01/2001 – 04/2001**
- **FCD ballot: 06/2001 – 10/2001**
- **FDIS ballot: 04/2002 – 06/2002**

Editor

Friedemann Schwenkreis
fschwenk@acm.org