

---

Package

**com.psychofree**

## com.psychofree Class PSFreeConnection

java.lang.Object

└─com.psychofree.PSFreeConnection

### All Implemented Interfaces:

com.jpbc.PConnection

```
public class PSFreeConnection
extends java.lang.Object
implements com.jpbc.PConnection
```

This class implements the JPBC interface PConnection.

## Constructor Summary

public	<a href="#">PSFreeConnection</a> (java.lang.String url, java.lang.String user, java.lang.String pwd)
--------	--

## Method Summary

void	<a href="#">close</a> ()
com.jpbc.PBStatement	<a href="#">createPBStatement</a> (java.lang.String command)
com.jpbc.PdlStatement	<a href="#">createPDLStatement</a> (java.lang.String command)
com.jpbc.PmlStatement	<a href="#">createPMLStatement</a> (java.lang.String command)
com.jpbc.PqlStatement	<a href="#">createPQLStatement</a> (java.lang.String command)
java.sql.Connection	<a href="#">getPBConnection</a> ()

### Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

### Methods inherited from interface com.jpbc.PConnection

`close, createPBStatement, createPDLStatement, createPMLStatement, createPQLStatement`

## Constructors

(continued on next page)

(continued from last page)

## PSFreeConnection

```
public PSFreeConnection(java.lang.String url,  
                        java.lang.String user,  
                        java.lang.String pwd)
```

## Methods

### createPBStatement

```
public com.jpbc.PBStatement createPBStatement(java.lang.String command)  
    throws com.jpbc.PException
```

### createPDLStatement

```
public com.jpbc.PdlStatement createPDLStatement(java.lang.String command)  
    throws com.jpbc.PException
```

### createPMLStatement

```
public com.jpbc.PmlStatement createPMLStatement(java.lang.String command)  
    throws com.jpbc.PException
```

### createPQLStatement

```
public com.jpbc.PqlStatement createPQLStatement(java.lang.String command)  
    throws com.jpbc.PException
```

### close

```
public void close()  
    throws com.jpbc.PException
```

### getPBConnection

```
public java.sql.Connection getPBConnection()
```

## com.psychofree Class PSFreeDriver

java.lang.Object

└─com.psychofree.PSFreeDriver

### All Implemented Interfaces:

com.jpbc.PDriver

```
public class PSFreeDriver
extends java.lang.Object
implements com.jpbc.PDriver
```

This class implements the JPBC interface PDriver.

## Constructor Summary

public	<a href="#">PSFreeDriver()</a>
--------	--------------------------------

## Method Summary

com.jpbc.PConnection	<a href="#">connect</a> (java.lang.String url, java.lang.String user, java.lang.String pwd)
int	<a href="#">PBMSCompliant</a> ()

### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Methods inherited from interface com.jpbc.PDriver

connect, PBMSCompliant

## Constructors

### PSFreeDriver

```
public PSFreeDriver()
```

## Methods

### connect

```
public com.jpbc.PConnection connect(java.lang.String url,
    java.lang.String user,
    java.lang.String pwd)
throws com.jpbc.PException
```

(continued from last page)

---

## **PBMSCompliant**

```
public int PBMSCompliant()
```

## com.psychofree Class PSFreeResultSet

java.lang.Object

└─com.psychofree.PSFreeResultSet

### All Implemented Interfaces:

com.jpbc.PResultSet

```
public class PSFreeResultSet
extends java.lang.Object
implements com.jpbc.PResultSet
```

This class implements the JPBC interface PResultSet.

## Constructor Summary

public	<a href="#">PSFreeResultSet</a> (java.sql.ResultSet rs)
--------	---

## Method Summary

void	<a href="#">close</a> ()
java.io.InputStream	<a href="#">getAsciiStream</a> (int columnIndex)
java.io.InputStream	<a href="#">getAsciiStream</a> (java.lang.String columnName)
java.io.InputStream	<a href="#">getBinaryStream</a> (int columnIndex)
java.io.InputStream	<a href="#">getBinaryStream</a> (java.lang.String columnName)
boolean	<a href="#">getBoolean</a> (int columnIndex)
boolean	<a href="#">getBoolean</a> (java.lang.String columnName)
byte	<a href="#">getBytes</a> (int columnIndex)
byte	<a href="#">getBytes</a> (java.lang.String columnName)
byte[]	<a href="#">getBytes</a> (int columnIndex)
byte[]	<a href="#">getBytes</a> (java.lang.String columnName)
java.sql.Date	<a href="#">getDate</a> (int columnIndex)
java.sql.Date	<a href="#">getDate</a> (java.lang.String columnName)

double	<a href="#">getDouble</a> (int columnIndex)
double	<a href="#">getDouble</a> (java.lang.String columnName)
float	<a href="#">getFloat</a> (int columnIndex)
float	<a href="#">getFloat</a> (java.lang.String columnName)
int	<a href="#">getInt</a> (int columnIndex)
int	<a href="#">getInt</a> (java.lang.String columnName)
long	<a href="#">getLong</a> (int columnIndex)
long	<a href="#">getLong</a> (java.lang.String columnName)
java.lang.Object	<a href="#">getObject</a> (int columnIndex)
short	<a href="#">getShort</a> (int columnIndex)
short	<a href="#">getShort</a> (java.lang.String columnName)
java.lang.String	<a href="#">getString</a> (int columnIndex)
java.lang.String	<a href="#">getString</a> (java.lang.String columnName)
java.sql.Time	<a href="#">getTime</a> (int columnIndex)
java.sql.Time	<a href="#">getTime</a> (java.lang.String columnName)
java.sql.Timestamp	<a href="#">getTimestamp</a> (int columnIndex)
java.sql.Timestamp	<a href="#">getTimestamp</a> (java.lang.String columnName)
boolean	<a href="#">next</a> ()

#### Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

#### Methods inherited from interface com.jpbc.PResultSet

`close`, `getAsciiStream`, `getAsciiStream`, `getBinaryStream`, `getBinaryStream`, `getBoolean`, `getBoolean`, `getByte`, `getByte`, `getBytes`, `getBytes`, `getDate`, `getDate`, `getDouble`, `getDouble`, `getFloat`, `getFloat`, `getInt`, `getInt`, `getLong`, `getLong`, `getObject`, `getShort`, `getShort`, `getString`, `getString`, `getTime`, `getTime`, `getTimestamp`, `getTimestamp`, `next`

## Constructors

(continued from last page)

## PSFreeResultSet

```
public PSFreeResultSet(java.sql.ResultSet rs)
```

## Methods

### next

```
public boolean next()  
    throws com.jpbc.PException
```

### close

```
public void close()  
    throws com.jpbc.PException
```

### getString

```
public java.lang.String getString(int columnIndex)  
    throws com.jpbc.PException
```

### getBoolean

```
public boolean getBoolean(int columnIndex)  
    throws com.jpbc.PException
```

### getBytes

```
public byte getBytes(int columnIndex)  
    throws com.jpbc.PException
```

### getShort

```
public short getShort(int columnIndex)  
    throws com.jpbc.PException
```

### getInt

```
public int getInt(int columnIndex)  
    throws com.jpbc.PException
```

(continued from last page)

---

## getLong

```
public long getLong(int columnIndex)
    throws com.jpbc.PException
```

---

## getFloat

```
public float getFloat(int columnIndex)
    throws com.jpbc.PException
```

---

## getDouble

```
public double getDouble(int columnIndex)
    throws com.jpbc.PException
```

---

## getBytes

```
public byte[] getBytes(int columnIndex)
    throws com.jpbc.PException
```

---

## getDate

```
public java.sql.Date getDate(int columnIndex)
    throws com.jpbc.PException
```

---

## getTime

```
public java.sql.Time getTime(int columnIndex)
    throws com.jpbc.PException
```

---

## getTimestamp

```
public java.sql.Timestamp getTimestamp(int columnIndex)
    throws com.jpbc.PException
```

---

## getAsciiStream

```
public java.io.InputStream getAsciiStream(int columnIndex)
    throws com.jpbc.PException
```

---

(continued from last page)

---

## getBinaryStream

```
public java.io.InputStream getBinaryStream(int columnIndex)
    throws com.jpbc.PException
```

---

## getString

```
public java.lang.String getString(java.lang.String columnName)
    throws com.jpbc.PException
```

---

## getBoolean

```
public boolean getBoolean(java.lang.String columnName)
    throws com.jpbc.PException
```

---

## getBytes

```
public byte getBytes(java.lang.String columnName)
    throws com.jpbc.PException
```

---

## getShort

```
public short getShort(java.lang.String columnName)
    throws com.jpbc.PException
```

---

## getInt

```
public int getInt(java.lang.String columnName)
    throws com.jpbc.PException
```

---

## getLong

```
public long getLong(java.lang.String columnName)
    throws com.jpbc.PException
```

---

## getFloat

```
public float getFloat(java.lang.String columnName)
    throws com.jpbc.PException
```

---

(continued from last page)

---

## getDouble

```
public double getDouble(java.lang.String columnName)
    throws com.jpbc.PException
```

---

## getBytes

```
public byte[] getBytes(java.lang.String columnName)
    throws com.jpbc.PException
```

---

## getDate

```
public java.sql.Date getDate(java.lang.String columnName)
    throws com.jpbc.PException
```

---

## getTime

```
public java.sql.Time getTime(java.lang.String columnName)
    throws com.jpbc.PException
```

---

## getTimestamp

```
public java.sql.Timestamp getTimestamp(java.lang.String columnName)
    throws com.jpbc.PException
```

---

## getAsciiStream

```
public java.io.InputStream getAsciiStream(java.lang.String columnName)
    throws com.jpbc.PException
```

---

## getBinaryStream

```
public java.io.InputStream getBinaryStream(java.lang.String columnName)
    throws com.jpbc.PException
```

---

## getObject

```
public java.lang.Object getObject(int columnIndex)
    throws com.jpbc.PException
```

---

## com.psychofree Class PSFreeStruct

java.lang.Object

└─ com.psychofree.PSFreeStruct

### All Implemented Interfaces:

com.jpbc.PStruct

```
public class PSFreeStruct
extends java.lang.Object
implements com.jpbc.PStruct
```

This class implements the JPBC interface PStruct.

## Constructor Summary

public	<a href="#">PSFreeStruct</a> (java.lang.String value)
public	<a href="#">PSFreeStruct</a> (java.lang.String value, java.lang.String typeName)

## Method Summary

java.lang.Object[]	<a href="#">getAttributes</a> () Parse the string representing the structured type in order to separate its elements.
java.lang.String	<a href="#">getTypeName</a> ()

### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Methods inherited from interface com.jpbc.PStruct

getAttributes, getTypeName

## Constructors

### PSFreeStruct

```
public PSFreeStruct(java.lang.String value)
```

### PSFreeStruct

```
public PSFreeStruct(java.lang.String value,
                    java.lang.String typeName)
```

## Methods

### **getTypeName**

```
public java.lang.String getTypeName()
```

---

### **getAttributes**

```
public java.lang.Object[] getAttributes()  
    throws com.jpbc.PException
```

Parse the string representing the structured type in order to separate its elements.

---

Package

**com.psychofree.mining**

## com.psychofree.mining Class Apriori

java.lang.Object

└─com.psychofree.mining.Apriori

```
public class Apriori
extends java.lang.Object
```

This class implements the Apriori algorithm for finding large itemsets. (see "Fast Algorithms for Mining Association Rules" by Rakesh Agrawal and Ramakrishnan Srikant from IBM Almaden Research Center 1994) This file is a part of the ARMiner project. (P)1999-2000 by ARMiner Server Team: Dana Cristofor Laurentiu Cristofor

### Constructor Summary

public	<a href="#">Apriori()</a>
--------	---------------------------

### Method Summary

java.util.Vector	<a href="#">findLargeItemsets</a> (java.sql.ResultSet resultset, float minSupport) Find the frequent itemsets in a database
int	<a href="#">getPass_num</a> ()
void	<a href="#">setPass_num</a> (int pass_num)

### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Constructors

### Apriori

```
public Apriori()
```

## Methods

### findLargeItemsets

```
public java.util.Vector findLargeItemsets(java.sql.ResultSet resultset,
float minSupport)
```

Find the frequent itemsets in a database

#### Parameters:

`resultset` - The object used to read from the database  
`minSupport` - The minimum support

(continued from last page)

**Returns:**

The large itemsets found

---

**getPass\_num**

```
public int getPass_num()
```

---

**setPass\_num**

```
public void setPass_num(int pass_num)
```

## com.psychofree.mining Class AprioriRules

```
java.lang.Object
  |
  +- com.psychofree.mining.MiningFunction
      |
      +- com.psychofree.mining.AprioriRules
```

```
public class AprioriRules
extends MiningFunction
```

This class runs the Apriori algorithm for Association Rules using the interface supplied by com.psychofree.mining.Apriori class

### Constructor Summary

public	<a href="#">AprioriRules</a> ()
--------	---------------------------------

### Method Summary

java.util.Vector	<a href="#">findAssociations</a> (java.util.Vector generatedItemsets, float minSupport, float minConfidence) Find association rules in a database, given the set of frequent itemsets.
static void	<a href="#">main</a> (java.lang.String[] args) Used for testing an instance of the class outside the PBMS Engine
java.lang.String	<a href="#">minePatterns</a> (java.lang.String[] args, java.sql.Connection conn) Start the process of rule extraction and store the result in the Pattern Base

#### Methods inherited from class [com.psychofree.mining.MiningFunction](#)

[createOutputTable](#), [minePatterns](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Constructors

#### AprioriRules

```
public AprioriRules()
```

### Methods

(continued from last page)

## minePatterns

```
public java.lang.String minePatterns(java.lang.String[] args,  
    java.sql.Connection conn)  
throws java.sql.SQLException
```

Start the process of rule extraction and store the result in the Pattern Base

**Parameters:**

args - The array containing the parameters for the mining function, which are: minimum value for support and confidence, name of the data source and the pattern type name  
conn - The existing database connection to be used

**Returns:**

The temporary table name storing the results

---

## findAssociations

```
public java.util.Vector findAssociations(java.util.Vector generatedItemsets,  
    float minSupport,  
    float minConfidence)
```

Find association rules in a database, given the set of frequent itemsets.

**Parameters:**

minSupport - the minimum support  
minConfidence - the minimum confidence

**Returns:**

a Vector containing all association rules found

---

## main

```
public static void main(java.lang.String[] args)
```

Used for testing an instance of the class outside the PBMS Engine

---

## com.psychofree.mining Class AprioriRulesWeka

```
java.lang.Object
  |
  +- com.psychofree.mining.MiningFunction
      |
      +- com.psychofree.mining.AprioriRulesWeka
```

```
public class AprioriRulesWeka
extends MiningFunction
```

This class runs the Apriori algorithm for Association Rules supplied by WEKA (weka.associations.Apriori)

### Constructor Summary

public	<a href="#">AprioriRulesWeka()</a>
--------	------------------------------------

### Method Summary

static weka.core.Instances	<a href="#">initMarketBasketDataSet</a> (weka.core.Instances instances, java.lang.String strFalse) Initialize a dataset for market basket analysis, all 'false' values have to be replaced with '?'
static java.lang.String	<a href="#">itemsetToString</a> (weka.associations.ItemSet set, weka.core.Instances instances) Returns a String representation of a given weka.associations.ItemSet
static void	<a href="#">main</a> (java.lang.String[] args) Used for testing an instance of the class outside the PBMS Engine
java.lang.String	<a href="#">minePatterns</a> (java.lang.String[] args, java.sql.Connection conn) Start the process of rule extraction and store the result in the Pattern Base

#### Methods inherited from class [com.psychofree.mining.MiningFunction](#)

[createOutputTable](#), [minePatterns](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Constructors

#### AprioriRulesWeka

```
public AprioriRulesWeka()
```

### Methods

---

(continued from last page)

## minePatterns

```
public java.lang.String minePatterns(java.lang.String[] args,  
    java.sql.Connection conn)  
    throws java.sql.SQLException,  
    java.lang.Exception
```

Start the process of rule extraction and store the result in the Pattern Base

### Parameters:

`args` - The String array containing the required mining function parameters, which are: minimum value for support and confidence, name of the data source, the number of rule to extract, a flag indicating if we are performing market basket analysis, the data source name and the pattern type name  
`conn` - The existing connection to be used

### Returns:

The temporary table name storing the results

---

## initMarketBasketDataSet

```
public static weka.core.Instances initMarketBasketDataSet(weka.core.Instances  
instances,  
    java.lang.String strFalse)
```

Initialize a dataset for market basket analysis, all 'false' values have to be replaced with '?' as they are missing values in this case, in order to get only associations between true items

### Parameters:

`instances` - the instances to be initialized  
`strFalse` - the string (nominal value) that represents the false value

### Returns:

The update instances of the dataset

---

## itemsetToString

```
public static java.lang.String itemsetToString(weka.associations.ItemSet set,  
    weka.core.Instances instances)
```

Returns a String representation of a given weka.associations.ItemSet

### Parameters:

`set` - The weka.associations.ItemSet  
`instances` - The dataset of the itemset

---

## main

```
public static void main(java.lang.String[] args)
```

Used for testing an instance of the class outside the PBMS Engine

---

## com.psychofree.mining Class AssociationRule

java.lang.Object

└-com.psychofree.mining.AssociationRule

### All Implemented Interfaces:

java.io.Serializable

```
public class AssociationRule
extends java.lang.Object
implements java.io.Serializable
```

AssociationRule.java An association rule has two parts: the antecedent of the rule and the consequent of the rule, both of which are sets of items. Associated with these are a support and a confidence. The support tells how many rows of a database support this rule, the confidence tells what percentage of the rows that contain the antecedent also contain the consequent.

Field Summary	
public static final	<a href="#">ANTECEDENT_SIZE</a> Value: 1
public static final	<a href="#">CONFIDENCE</a> Value: 4
public static final	<a href="#">CONSEQUENT_SIZE</a> Value: 2
public static final	<a href="#">SUPPORT</a> Value: 3

Constructor Summary	
public	<a href="#">AssociationRule</a> ( <a href="#">Itemset</a> antecedent, <a href="#">Itemset</a> consequent, float support, float confidence) Creates a new association rule.

Method Summary	
int	<a href="#">antecedentSize</a> () Return size of antecedent.
int	<a href="#">compareTo</a> ( <a href="#">AssociationRule</a> ar, int criteria) Compare two AssociationRule objects on one of several criteria.
int	<a href="#">consequentSize</a> () Return size of consequent.
boolean	<a href="#">equals</a> (java.lang.Object obj) Compare two AssociationRule objects on one of several criteria.

int	<a href="#">getAntecedentItem</a> (int i) Return i-th item in antecedent.
float	<a href="#">getConfidence</a> () Return confidence of association rule.
int	<a href="#">getConsequentItem</a> (int i) Return i-th item in consequent.
float	<a href="#">getSupport</a> () Return support of association rule.
java.lang.String	<a href="#">toString</a> () Return a java.lang.String representation of the AssociationRule.

#### Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Fields

### ANTECEDENT\_SIZE

```
public static final int ANTECEDENT_SIZE
```

Constant value: **1**

### CONSEQUENT\_SIZE

```
public static final int CONSEQUENT_SIZE
```

Constant value: **2**

### SUPPORT

```
public static final int SUPPORT
```

Constant value: **3**

### CONFIDENCE

```
public static final int CONFIDENCE
```

Constant value: **4**

## Constructors

(continued from last page)

## AssociationRule

```
public AssociationRule(Itemset antecedent,  
                     Itemset consequent,  
                     float support,  
                     float confidence)
```

Creates a new association rule.

### Parameters:

antecedent - the antecedent of the association rule  
consequent - the consequent of the association rule  
support - the support of the association rule  
confidence - the confidence of the association rule

### Throws:

IllegalArgumentException - antecedent or consequent are null or support or confidence are not between 0 and 1

## Methods

### antecedentSize

```
public int antecedentSize()
```

Return size of antecedent.

### Returns:

size of antecedent

### consequentSize

```
public int consequentSize()
```

Return size of consequent.

### Returns:

size of consequent

### getSupport

```
public float getSupport()
```

Return support of association rule.

### getConfidence

```
public float getConfidence()
```

Return confidence of association rule.

### getAntecedentItem

```
public int getAntecedentItem(int i)
```

Return i-th item in antecedent.

### Parameters:

(continued from last page)

i - the index of the item to get

**Returns:**

the i-th item in antecedent

**Throws:**

`IndexOutOfBoundsException` - i is an invalid index

---

## getConsequentItem

```
public int getConsequentItem(int i)
```

Return i-th item in consequent.

**Parameters:**

i - the index of the item to get

**Returns:**

the i-th item in consequent

**Throws:**

`IndexOutOfBoundsException` - i is an invalid index

---

## compareTo

```
public int compareTo(AssociationRule ar,  
int criteria)
```

Compare two `AssociationRule` objects on one of several criteria.

**Parameters:**

ar - the `AssociationRule` object with which we want to compare this object

criteria - the criteria on which we want to compare, can be one of `ANTECEDENT_SIZE`, `CONSEQUENT_SIZE`, `SUPPORT` or `CONFIDENCE`.

**Returns:**

a negative value if this object is smaller than ar, 0 if they are equal, and a positive value if this object is greater.

**Throws:**

`IllegalArgumentException` - ar is null or criteria is invalid

---

## equals

```
public boolean equals(java.lang.Object obj)
```

Compare two `AssociationRule` objects on one of several criteria.

**Returns:**

true if the objects are equal in terms of antecedent and consequent items; false otherwise.

---

## toString

```
public java.lang.String toString()
```

Return a `java.lang.String` representation of the `AssociationRule`.

**Returns:**

`java.lang.String` representation of `AssociationRule`

---

## com.psychofree.mining Class AssociationRuleWeka

java.lang.Object

└─com.psychofree.mining.AssociationRuleWeka

### All Implemented Interfaces:

java.io.Serializable

public class **AssociationRuleWeka**  
 extends java.lang.Object  
 implements java.io.Serializable

Weka version of association rule

### Field Summary

public static final	<a href="#">ANTECEDENT_SIZE</a> Value: 1
public static final	<a href="#">CONFIDENCE</a> Value: 4
public static final	<a href="#">CONSEQUENT_SIZE</a> Value: 2
public static final	<a href="#">SUPPORT</a> Value: 3

### Constructor Summary

public	<a href="#">AssociationRuleWeka</a> (weka.associations.ItemSet antecedent, weka.associations.ItemSet consequent, double support, double confidence) Creates a new association rule.
--------	--

### Method Summary

int	<a href="#">antecedentSize</a> () Returns size of antecedent.
java.lang.String	<a href="#">antecedentToString</a> (weka.core.Instances instances) Return a java.lang.String representation of the AssociationRule
int	<a href="#">compareTo</a> ( <a href="#">AssociationRuleWeka</a> ar, int criteria) Compare two AssociationRule objects on one of several criteria.
int	<a href="#">consequentSize</a> () Returns size of consequent.
java.lang.String	<a href="#">consequentToString</a> (weka.core.Instances instances)

boolean	<a href="#">equals</a> (java.lang.Object obj)
weka.associations.ItemSet	<a href="#">getAntecedent</a> () Return the antecedent.
double	<a href="#">getConfidence</a> () Return confidence of association rule.
weka.associations.ItemSet	<a href="#">getConsequent</a> () Return the consequent.
double	<a href="#">getSupport</a> () Return support of association rule.

#### Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Fields

### ANTECEDENT\_SIZE

```
public static final int ANTECEDENT_SIZE
```

Constant value: 1

### CONSEQUENT\_SIZE

```
public static final int CONSEQUENT_SIZE
```

Constant value: 2

### SUPPORT

```
public static final int SUPPORT
```

Constant value: 3

### CONFIDENCE

```
public static final int CONFIDENCE
```

Constant value: 4

## Constructors

### AssociationRuleWeka

```
public AssociationRuleWeka(weka.associations.ItemSet antecedent,
    weka.associations.ItemSet consequent,
    double support,
    double confidence)
```

(continued from last page)

Creates a new association rule.

**Parameters:**

antecedent - the antecedent of the association rule  
consequent - the consequent of the association rule  
confidence - the confidence of the association rule

## Methods

### antecedentSize

```
public int antecedentSize()
```

Returns size of antecedent.

**Returns:**

size of antecedent

---

### consequentSize

```
public int consequentSize()
```

Returns size of consequent.

**Returns:**

size of consequent

---

### getSupport

```
public double getSupport()
```

Return support of association rule.

---

### getConfidence

```
public double getConfidence()
```

Return confidence of association rule.

---

### getAntecedent

```
public weka.associations.ItemSet getAntecedent()
```

Return the antecedent.

---

### getConsequent

```
public weka.associations.ItemSet getConsequent()
```

Return the consequent.

---

### compareTo

```
public int compareTo(AssociationRuleWeka ar,  
int criteria)  
throws java.lang.IllegalArgumentException
```

Compare two AssociationRule objects on one of several criteria.

---

(continued from last page)

**Parameters:**

`ar` - the AssociationRule object with which we want to compare this object  
`criteria` - the criteria on which we want to compare, can be one of ANTECEDENT\_SIZE, CONSEQUENT\_SIZE, SUPPORT or CONFIDENCE.

**Returns:**

a negative value if this object is smaller than `ar`, 0 if they are equal, and a positive value if this object is greater.

---

**equals**

```
public boolean equals(java.lang.Object obj)
```

**Returns:**

true if the objects are equal in terms of antecedent and consequent items, false otherwise.

---

**antecedentToString**

```
public java.lang.String antecedentToString(weka.core.Instances instances)
```

Return a java.lang.String representation of the AssociationRule

**Parameters:**

`instances` - dataset containing header informations for the antecedent

**Returns:**

java.lang.String representation of AssociationRule

---

**consequentToString**

```
public java.lang.String consequentToString(weka.core.Instances instances)
```

## com.psychofree.mining Class BasicKMeans

java.lang.Object

└-com.psychofree.mining.BasicKMeans

All Implemented Interfaces:

[KMeans](#)

public class **BasicKMeans**  
 extends java.lang.Object  
 implements [KMeans](#)

Basic implementation of K-means clustering. Since it's a Runnable, it's designed to be executed by a dedicated thread, but that thread does not create any other threads to divide up the work.

### Constructor Summary

public	<a href="#">BasicKMeans</a> (double[][][] coordinates, int k, int maxIterations, long randomSeed) Constructor
--------	--

### Method Summary

void	<a href="#">addKMeansListener</a> ( <a href="#">KMeansListener</a> l) Adds a KMeansListener to be notified of significant happenings.
static double[][][]	<a href="#">generateCoordinates</a> (int coordCount, int dimensions, int clusterCount, long randomSeed) Generates sample coordinates datasets.
<a href="#">Cluster</a> []	<a href="#">getClusters</a> () Get the clusters computed by the algorithm.
double[][][]	<a href="#">getDistances</a> ()
void	<a href="#">removeKMeansListener</a> ( <a href="#">KMeansListener</a> l) Removes a KMeansListener from the listener list.
void	<a href="#">run</a> () Run the clustering algorithm.

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface [com.psychofree.mining.KMeans](#)

[addKMeansListener](#), [getClusters](#), [removeKMeansListener](#)

#### Methods inherited from interface java.lang.Runnable

[run](#)

---

## Constructors

### BasicKMeans

```
public BasicKMeans(double[][] coordinates,  
                  int k,  
                  int maxIterations,  
                  long randomSeed)
```

Constructor

**Parameters:**

`coordinates` - two-dimensional array containing the coordinates to be clustered.  
`k` - the number of desired clusters.  
`maxIterations` - the maximum number of clustering iterations.  
`randomSeed` - seed used with the random number generator.

---

## Methods

### addKMeansListener

```
public void addKMeansListener(KMeansListener l)
```

Adds a `KMeansListener` to be notified of significant happenings.

**Parameters:**

`l` - the listener to be added.

---

### removeKMeansListener

```
public void removeKMeansListener(KMeansListener l)
```

Removes a `KMeansListener` from the listener list.

**Parameters:**

`l` - the listener to be removed.

---

### getClusters

```
public Cluster[] getClusters()
```

Get the clusters computed by the algorithm. This method should not be called until clustering has completed successfully.

**Returns:**

an array of `Cluster` objects.

---

### getDistances

```
public double[][] getDistances()
```

---

(continued from last page)

## generateCoordinates

```
public static double[][] generateCoordinates(int coordCount,  
        int dimensions,  
        int clusterCount,  
        long randomSeed)  
throws InsufficientMemoryException
```

Generates sample coordinates datasets.

### Parameters:

coordCount - the number of coordinates.  
dimensions - the length of the coordinates.  
clusterCount - the number of clusters in the distribution.  
randomSeed - the seed used by the random number generator.

### Returns:

sample coordinates generated

---

## run

```
public void run()
```

Run the clustering algorithm.

## com.psychofree.mining Class Cluster

java.lang.Object

↳ com.psychofree.mining.Cluster

```
public class Cluster
extends java.lang.Object
```

Class to represent a cluster of coordinates.

### Constructor Summary

public	<a href="#">Cluster</a> (int[] memberIndexes, double[] center) Constructor.
--------	--

### Method Summary

double[]	<a href="#">getCenter</a> () Get the cluster center.
int[]	<a href="#">getMemberIndexes</a> () Get the member indices.
java.lang.String	<a href="#">toString</a> () Returns a string representation of the cluster containing the centroid and the cluster dimension (the number of instance data contained)

### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### Cluster

```
public Cluster(int[] memberIndexes,
              double[] center)
```

Constructor.

#### Parameters:

memberIndexes - indices of the member coordinates.  
 center - the cluster center.

## Methods

### getMemberIndexes

```
public int[] getMemberIndexes()
```

Get the member indices.

(continued from last page)

**Returns:**

an array containing the indices of the member coordinates.

---

**getCenter**

```
public double[] getCenter()
```

Get the cluster center.

**Returns:**

a reference to the cluster center array.

---

**toString**

```
public java.lang.String toString()
```

Returns a string representation of the cluster containing the centroid and the cluster dimension (the number of instance data contained)

**Returns:**

a String representing the cluster.

## com.psychofree.mining Class ExampleSimpleKMeans

```
java.lang.Object
  |
  +- com.psychofree.mining.MiningFunction
      |
      +- com.psychofree.mining.ExampleSimpleKMeans
```

```
public class ExampleSimpleKMeans
extends MiningFunction
```

This class runs the K-means algorithm for Clustering supplied by WEKA (weka.clusterers.SimpleKMeans) using a dataset containing numeric and nominal attributes

### Constructor Summary

public	<a href="#">ExampleSimpleKMeans()</a>
--------	---------------------------------------

### Method Summary

static void	<a href="#">main</a> (java.lang.String[] args) Used for testing an instance of the class outside the PBMS Engine
java.lang.String	<a href="#">minePatterns</a> (java.lang.String[] args, java.sql.Connection conn) Start the process of rule extraction and store the result in the Pattern Base

#### Methods inherited from class [com.psychofree.mining.MiningFunction](#)

[createOutputTable](#), [minePatterns](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Constructors

### ExampleSimpleKMeans

```
public ExampleSimpleKMeans()
```

## Methods

### minePatterns

```
public java.lang.String minePatterns(java.lang.String[] args,
    java.sql.Connection conn)
throws java.sql.SQLException,
    java.lang.Exception
```

Start the process of rule extraction and store the result in the Pattern Base

(continued from last page)

**Parameters:**

args - The String array containing the mining function parameters

conn - The connection

---

**main**

```
public static void main(java.lang.String[] args)
```

Used for testing an instance of the class outside the PBMS Engine

## com.psychofree.mining

### Class HashTree

java.lang.Object

└-com.psychofree.mining.HashTree

public class **HashTree**  
 extends java.lang.Object

A HashTree is a special data structure that is used to index a Vector of Itemset objects for more efficient processing.

### Constructor Summary

public	<a href="#">HashTree</a> (int listSize, int hashSize, java.util.Vector itemsets) Create a new HashTree.
public	<a href="#">HashTree</a> (java.util.Vector itemsets) Create a new HashTree.

### Method Summary

void	<a href="#">add</a> (int index) This method indexes in the HashTree the Itemset at index index from Vector itemsets which was passed to the constructor of this HashTree.
void	<a href="#">checkLargeness</a> ( <a href="#">Itemset</a> itemset) Verifies if any of the indexed Itemsets is not large by checking whether they're included in the frequent itemset itemset.
long	<a href="#">countFrequentSubsets</a> ( <a href="#">Itemset</a> itemset, long minWeight) Count how many frequent Itemsets (frequent = having weight greater than a specified minimum weight) are included in itemset
long	<a href="#">countSubsets</a> ( <a href="#">Itemset</a> itemset) Count how many Itemsets are included in itemset
void	<a href="#">prepareForDescent</a> () This method should be called before calling update() to gather all leaves of the HashTree for more efficient processing.
void	<a href="#">update</a> ( <a href="#">Itemset</a> row) Update the weights of all indexed Itemsets that are included in row
void	<a href="#">update</a> ( <a href="#">Itemset</a> row, long[][] counts) Update the weights of all indexed Itemsets that are included in row and also update the matrix counts

### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructors

(continued from last page)

## HashTree

```
public HashTree(int listSize,  
                int hashSize,  
                java.util.Vector itemsets)
```

Create a new HashTree. The `listSize` parameter determines after how many inserts in a ListNode we have to change it to a HashNode (i.e. perform a split). The `hashSize` parameter can be specified to improve the efficiency of the structure.

### Parameters:

`listSize` - the size of the internal lists in the list nodes  
`hashSize` - the size of the internal hashtables in the hash nodes  
`itemsets` - the Vector of Itemsets that we should index

### Throws:

IllegalArgumentException - `itemsets` is null or `listSize` <= 0 or `hashSize` <= 0

---

## HashTree

```
public HashTree(java.util.Vector itemsets)
```

Create a new HashTree. This initializes the HashTree with default parameters.

### Parameters:

`itemsets` - the Vector of Itemsets that we should index

### Throws:

IllegalArgumentException - `itemsets` is null

## Methods

### prepareForDescent

```
public void prepareForDescent()
```

This method should be called before calling `update()` to gather all leaves of the HashTree for more efficient processing.

---

### add

```
public void add(int index)
```

This method indexes in the HashTree the Itemset at index `index` from Vector `itemsets` which was passed to the constructor of this HashTree.

### Parameters:

`index` - the index of the Itemset that we need to index in this HashTree.

---

### update

```
public void update(Itemset row)
```

Update the weights of all indexed Itemsets that are included in `row`

### Parameters:

`row` - the Itemset (normally a database row) against which we test for inclusion

(continued from last page)

## update

```
public void update(Itemset row,  
                  long[][] counts)
```

Update the weights of all indexed Itemsets that are included in `row` and also update the matrix `counts`

### Parameters:

`row` - the Itemset (normally a database row) against which we test for inclusion

`counts` - a matrix used by some algorithms to speed up computations; its rows correspond to Itemsets and its columns correspond to items; each value in the matrix tells for how many times had an item appeared together with an itemset in the rows of the database.

---

## countFrequentSubsets

```
public long countFrequentSubsets(Itemset itemset,  
                                  long minWeight)
```

Count how many frequent Itemsets (frequent = having weight greater than a specified minimum weight) are included in `itemset`

### Parameters:

`itemset` - the Itemset for which we count the subsets

`minWeight` - the minimum weight

---

## countSubsets

```
public long countSubsets(Itemset itemset)
```

Count how many Itemsets are included in `itemset`

### Parameters:

`itemset` - the Itemset for which we count the subsets

---

## checkLargeness

```
public void checkLargeness(Itemset itemset)
```

Verifies if any of the indexed Itemsets is not large by checking whether they're included in the frequent itemset `itemset`. If an Itemset is not large then it will be marked.

### Parameters:

`itemset` - the Itemset we check

## com.psychofree.mining Class **InsufficientMemoryException**

```

java.lang.Object
  |
  +- java.lang.Throwable
      |
      +- java.lang.Exception
          |
          +- com.psychofree.mining.InsufficientMemoryException
  
```

### All Implemented Interfaces:

java.io.Serializable

```

public class InsufficientMemoryException
extends java.lang.Exception
  
```

Exception thrown when insufficient memory is available to perform an operation. Designed to be throw before doing something that would cause a `java.lang.OutOfMemoryError`.

## Constructor Summary

public	<a href="#">InsufficientMemoryException</a> (java.lang.String message) Constructor.
public	<a href="#">InsufficientMemoryException</a> () Default constructor.

### Methods inherited from class java.lang.Throwable

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### **InsufficientMemoryException**

```
public InsufficientMemoryException(java.lang.String message)
```

Constructor.

#### Parameters:

message - an explanatory message.

### **InsufficientMemoryException**

```
public InsufficientMemoryException()
```

Default constructor.

## com.psychofree.mining Class Itemset

java.lang.Object

↳ com.psychofree.mining.Itemset

### All Implemented Interfaces:

java.io.Serializable

```
public class Itemset
extends java.lang.Object
implements java.io.Serializable
```

An itemset is an ordered list of integers that identify items coupled with a float value representing the support of the itemset as a percentage.

### Constructor Summary

public	<a href="#">Itemset()</a> Creates a new empty itemset.
public	<a href="#">Itemset(int c)</a> Create a new empty itemset of specified capacity.
public	<a href="#">Itemset(Itemset itemset)</a> Create a new itemset by copying a given one.

### Method Summary

<a href="#">Itemset</a>	<a href="#">add(Itemset itemset)</a> Return a new Itemset that contains all those items that appear in this Itemset and in itemset.
boolean	<a href="#">addItem(int item)</a> Add a new item to the itemset.
boolean	<a href="#">canCombineWith(Itemset itemset)</a> Check whether two itemsets can be combined.
<a href="#">Itemset</a>	<a href="#">combineWith(Itemset itemset)</a> Combine two itemsets into a new one that will contain all the items in the first itemset plus the last item in the second itemset.
boolean	<a href="#">doesIntersect(Itemset itemset)</a> Return true if this itemset has items in common with itemset.
int	<a href="#">getFirstItem()</a> Return first item in set.
int	<a href="#">getItem(int i)</a> Return i-th item in set.
int	<a href="#">getNextItem()</a> Return next item in set.

float	<a href="#">getSupport()</a> Return support of itemset.
long	<a href="#">getWeight()</a> Return weight of itemset.
boolean	<a href="#">hasMoreItems()</a> Return true if there are more items in the itemset.
void	<a href="#">incrementWeight()</a> Increment the weight of the itemset.
boolean	<a href="#">isEqualTo(Itemset itemset)</a> Checks equality with a given itemset.
boolean	<a href="#">isIncludedIn(Itemset itemset)</a> Checks inclusion in a given itemset.
boolean	<a href="#">isMarked()</a> Return itemset mark.
boolean	<a href="#">mark()</a> Mark the itemset.
static void	<a href="#">pruneDuplicates(java.util.Vector v)</a> Remove all duplicate itemsets from the vector v
static void	<a href="#">pruneNonMaximal(java.util.Vector v)</a> Remove all non-maximal itemsets from the vector v
boolean	<a href="#">removeItem(int item)</a> Removes a given item from the itemset.
boolean	<a href="#">removeLastItem()</a> Removes last item (which has the greatest value) from the itemset.
void	<a href="#">setSupport(float newSupport)</a> Set the support of the itemset.
void	<a href="#">setWeight(long newWeight)</a> Set the weight of the itemset.
int	<a href="#">size()</a> Return size of itemset.
<a href="#">Itemset</a>	<a href="#">subtract(Itemset itemset)</a> Return a new Itemset that contains only those items that do not appear in itemset.
java.lang.String	<a href="#">toString()</a> Return a String representation of the Itemset.
boolean	<a href="#">unmark()</a> Unmark the itemset.

#### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

(continued from last page)

## Constructors

### Itemset

```
public Itemset()
```

Creates a new empty itemset.

### Itemset

```
public Itemset(int c)
```

Create a new empty itemset of specified capacity.

**Parameters:**

c - the capacity of the itemset

**Throws:**

`IllegalArgumentException` - c is negative or zero

### Itemset

```
public Itemset(Itemset itemset)
```

Create a new itemset by copying a given one.

**Parameters:**

itemset - the itemset to be copied

**Throws:**

`IllegalArgumentException` - itemset is null

## Methods

### getSupport

```
public float getSupport()
```

Return support of itemset.

### getWeight

```
public long getWeight()
```

Return weight of itemset.

### getItem

```
public int getItem(int i)
```

Return i-th item in set.

**Parameters:**

i - the index of the item to get

**Returns:**

the i-th item

(continued from last page)

**Throws:**`IndexOutOfBoundsException` - `i` is an invalid index

---

**getFirstItem**

```
public int getFirstItem()
```

Return first item in set.

**Returns:**

first item

**Throws:**`IndexOutOfBoundsException` - there is no first item

---

**getNextItem**

```
public int getNextItem()
```

Return next item in set.

**Returns:**

next item

**Throws:**`IndexOutOfBoundsException` - there is no next item

---

**hasMoreItems**

```
public boolean hasMoreItems()
```

Return true if there are more items in the itemset. You can call this method to find out whether you can call `getNext` without raising an exception.

**Returns:**

true if there are more items, false if not

---

**size**

```
public int size()
```

Return size of itemset.

**Returns:**

size of itemset

---

**doesIntersect**

```
public boolean doesIntersect(Itemset itemset)
```

Return true if this itemset has items in common with `itemset`.

**Parameters:**

`itemset` - the itemset with which we compare

**Returns:**

true if `itemset` contains items of this itemset, false otherwise.

**Throws:**

---

(continued from last page)

---

`IllegalArgumentException - itemset is null`

---

## subtract

```
public Itemset subtract(Itemset itemset)
```

Return a new Itemset that contains only those items that do not appear in `itemset`.

**Parameters:**

`itemset` - the itemset whose items we want to subtract

**Returns:**

an Itemset containing only those items of this Itemset that do not appear in `itemset`.

**Throws:**

`IllegalArgumentException - itemset is null`

---

## add

```
public Itemset add(Itemset itemset)
```

Return a new Itemset that contains all those items that appear in this Itemset and in `itemset`.

**Parameters:**

`itemset` - the itemset whose items we want to add

**Returns:**

an Itemset containing all those items that appear in this Itemset and in `itemset`.

**Throws:**

`IllegalArgumentException - itemset is null`

---

## addItem

```
public boolean addItem(int item)
```

Add a new item to the itemset.

**Parameters:**

`item` - the item to be added

**Returns:**

true if item was added, false if it wasn't added (was already there!)

**Throws:**

`IllegalArgumentException - item is <= 0`

---

## removeItem

```
public boolean removeItem(int item)
```

Removes a given item from the itemset.

**Parameters:**

`item` - the item to remove

**Returns:**

true if item was removed, false if it wasn't removed (was not found in itemset!)

---

(continued from last page)

**Throws:**IllegalArgumentException - item is  $\leq 0$ 

---

**removeLastItem**public boolean **removeLastItem**()

Removes last item (which has the greatest value) from the itemset.

**Returns:**

true if item was removed, false if it wasn't removed (the itemset was empty)

---

**setSupport**public void **setSupport**(float newSupport)

Set the support of the itemset.

**Parameters:**

newSupport - the support of the itemset

**Throws:**IllegalArgumentException - newSupport is  $< 0$  or  $> 100$ 

---

**setWeight**public void **setWeight**(long newWeight)

Set the weight of the itemset.

**Parameters:**

newWeight - the weight of the itemset

**Throws:**IllegalArgumentException - newWeight is  $< 0$ 

---

**incrementWeight**public void **incrementWeight**()

Increment the weight of the itemset.

---

**isEqualTo**public boolean **isEqualTo**([Itemset](#) itemset)

Checks equality with a given itemset.

**Parameters:**

itemset - the itemset against which we test for equality

**Throws:**

IllegalArgumentException - itemset is null

---

**isIncludedIn**public boolean **isIncludedIn**([Itemset](#) itemset)

---

(continued from last page)

Checks inclusion in a given itemset.

**Parameters:**

itemset - the itemset against which we test for inclusion

**Throws:**

IllegalArgumentException - itemset is null

---

## mark

```
public boolean mark()
```

Mark the itemset.

**Returns:**

true if itemset was already marked, false otherwise

---

## unmark

```
public boolean unmark()
```

Unmark the itemset.

**Returns:**

true if itemset was marked, false otherwise

---

## isMarked

```
public boolean isMarked()
```

Return itemset mark.

**Returns:**

true if itemset is marked, false otherwise

---

## toString

```
public java.lang.String toString()
```

Return a String representation of the Itemset.

**Returns:**

String representation of Itemset

---

## canCombineWith

```
public boolean canCombineWith(Itemset itemset)
```

Check whether two itemsets can be combined. Two itemsets can be combined if they differ only in the last item.

**Parameters:**

itemset - itemset with which to combine

**Returns:**

true if the itemsets can be combined, false otherwise

**Throws:**

IllegalArgumentException - itemset is null

---

## combineWith

```
public Itemset combineWith(Itemset itemset)
```

Combine two itemsets into a new one that will contain all the items in the first itemset plus the last item in the second itemset.

**Parameters:**

itemset - itemset with which to combine

**Returns:**

an itemset that combines the two itemsets as described above

**Throws:**

`IllegalArgumentException` - itemset is null

---

## pruneNonMaximal

```
public static void pruneNonMaximal(java.util.Vector v)
```

Remove all non-maximal itemsets from the vector v

**Parameters:**

v - the collection of itemsets

---

## pruneDuplicates

```
public static void pruneDuplicates(java.util.Vector v)
```

Remove all duplicate itemsets from the vector v

**Parameters:**

v - the collection of itemsets

---

## com.psychofree.mining Interface KMeans

All Known Implementing Classes:

[BasicKMeans](#)

public interface **KMeans**  
extends java.lang Runnable

Simple K-Means clustering interface.

### Method Summary

void	<a href="#">addKMeansListener</a> ( <a href="#">KMeansListener</a> l)	Adds a KMeansListener to be notified of significant happenings.
<a href="#">Cluster[]</a>	<a href="#">getClusters</a> ()	Get the clusters computed by the algorithm.
void	<a href="#">removeKMeansListener</a> ( <a href="#">KMeansListener</a> l)	Removes a KMeansListener from the listener list.

### Methods inherited from interface java.lang Runnable

run

## Methods

### addKMeansListener

public void **addKMeansListener**([KMeansListener](#) l)

Adds a KMeansListener to be notified of significant happenings.

**Parameters:**

l - the listener to be added.

### removeKMeansListener

public void **removeKMeansListener**([KMeansListener](#) l)

Removes a KMeansListener from the listener list.

**Parameters:**

l - the listener to be removed.

### getClusters

public [Cluster\[\]](#) **getClusters**()

Get the clusters computed by the algorithm. This method should not be called until clustering has completed successfully.

(continued from last page)

**Returns:**

an array of Cluster objects.

---

## com.psychofree.mining Interface KMeansListener

---

public interface **KMeansListener**  
extends

Defines object which register with implementation of `KMeans` to be notified of significant events during clustering.

---

### Method Summary

void	<a href="#">kmeansComplete</a> ( <a href="#">Cluster[]</a> clusters, long executionTime) KMeans is complete.
void	<a href="#">kmeansError</a> (java.lang.Throwable t) An error occurred during KMeans clustering.
void	<a href="#">kmeansMessage</a> (java.lang.String message) A message has been received.

---

### Methods

#### **kmeansMessage**

public void **kmeansMessage**(java.lang.String message)

A message has been received.

**Parameters:**

message

---

#### **kmeansComplete**

public void **kmeansComplete**([Cluster\[\]](#) clusters,  
long executionTime)

KMeans is complete.

**Parameters:**

clusters - the output of clustering.

executionTime - the time in milliseconds taken to cluster.

---

#### **kmeansError**

public void **kmeansError**(java.lang.Throwable t)

An error occurred during KMeans clustering.

**Parameters:**

t

---

## com.psychofree.mining Class KMeansPoints

```
java.lang.Object
  |
  +- com.psychofree.mining.MiningFunction
      |
      +- com.psychofree.mining.KMeansPoints
```

```
public class KMeansPoints
extends MiningFunction
```

This class runs the K-means algorithm for Clustering provided by WEKA (weka.clusterers.SimpleKMeans) using a dataset containing only numeric attributes

### Constructor Summary

public	<a href="#">KMeansPoints()</a>
--------	--------------------------------

### Method Summary

static void	<a href="#">main</a> (java.lang.String[] args) Used for testing an instance of the class outside the PBMS Engine
java.lang.String	<a href="#">minePatterns</a> (java.lang.String[] args, java.sql.Connection conn) Start the process of rule extraction and store the result in the Pattern Base.

#### Methods inherited from class [com.psychofree.mining.MiningFunction](#)

[createOutputTable](#), [minePatterns](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Constructors

### **KMeansPoints**

```
public KMeansPoints()
```

## Methods

### **minePatterns**

```
public java.lang.String minePatterns(java.lang.String[] args,
    java.sql.Connection conn)
throws java.sql.SQLException,
    java.lang.Exception
```

Start the process of rule extraction and store the result in the Pattern Base.

(continued from last page)

**Parameters:**

args - The String array containing the mining function parameters  
conn - The database connection

---

**main**

```
public static void main(java.lang.String[] args)
```

Used for testing an instance of the class outside the PBMS Engine

## com.psychofree.mining Class MiningFunction

java.lang.Object

└-com.psychofree.mining.MiningFunction

Direct Known Subclasses:

[AprioriRules](#), [AprioriRulesWeka](#), [ExampleSimpleKMeans](#), [KMeansPoints](#), [UsoBasicKMeans](#)

public abstract class **MiningFunction**  
extends java.lang.Object

This is the superclass of all mining functions. User who wsh to create a new java mining function should extend this class. As all PSYCHOfree MUST return a table name containing mined patterns with (at least) an initialized structure method createOutputTable is supplied; therefore, user only need to create a Pattern array containg all mined patterns and everything else is left to this method

### Constructor Summary

public	<a href="#">MiningFunction()</a>
--------	----------------------------------

### Method Summary

static java.lang.String	<a href="#">createOutputTable</a> (java.sql.Connection conn, Pattern[] patterns) Creates the temporary output table which stores the extracted patterns
abstract java.lang.String	<a href="#">minePatterns</a> (java.lang.String[] args, java.sql.Connection conn) Method wich performs the patterns extraction

### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### MiningFunction

public **MiningFunction**()

## Methods

### minePatterns

```
public abstract java.lang.String minePatterns(java.lang.String[] args,
        java.sql.Connection conn)
    throws java.sql.SQLException,
        java.lang.Exception
```

Method wich performs the patterns extraction

## createOutputTable

```
public static java.lang.String createOutputTable(java.sql.Connection conn,  
        Pattern[] patterns)  
    throws java.sql.SQLException
```

Creates the temporary output table which stores the extracted patterns

**Parameters:**

conn - The existing database connection to be used

patterns - The array containing the extracted patterns

## com.psychofree.mining Class SET

java.lang.Object

└-com.psychofree.mining.SET

public class **SET**  
extends java.lang.Object

Implements a Set Enumeration Tree, which is a prefix tree used for storing and retrieving itemset information.

### Constructor Summary

public	<a href="#">SET()</a> Create a new empty SET.
--------	--

### Method Summary

java.util.Vector	<a href="#">getItemsets()</a> Return the itemsets of the SET.
java.util.Vector	<a href="#">getLargeItemsets()</a> Return the maximal itemsets of the SET.
float	<a href="#">getSupport(Itemset itemset)</a> Return the support for a given itemset.
void	<a href="#">insert(Itemset itemset)</a> Insert a new itemset in the SET.

### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### SET

public **SET**()

Create a new empty SET.

## Methods

### insert

public void **insert**([Itemset](#) itemset)

Insert a new itemset in the SET.

#### Parameters:

---

(continued from last page)

itemset - the itemset to be inserted

**Throws:**

IllegalArgumentException - itemset is null or is empty

---

## getSupport

```
public float getSupport(Itemset itemset)
    throws java.util.NoSuchElementException
```

Return the support for a given itemset.

**Parameters:**

itemset - the itemset for which we want to obtain the support

**Returns:**

support

**Throws:**

IllegalArgumentException - itemset is null or is empty

NoSuchElementException - itemset not found in SET

---

## getLargeItemsets

```
public java.util.Vector getLargeItemsets()
```

Return the maximal itemsets of the SET.

**Returns:**

a vector containing the maximal itemsets from the SET

---

## getItemsets

```
public java.util.Vector getItemsets()
```

Return the itemsets of the SET.

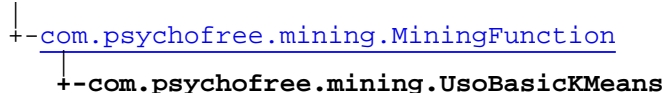
**Returns:**

a vector containing the itemsets from the SET

---

## com.psychofree.mining Class UsobasicKMeans

java.lang.Object



public class **UsobasicKMeans**  
extends [MiningFunction](#)

This class runs the K-means algorithm for Clustering provided by BasikMeans class

### Constructor Summary

public	<a href="#">UsobasicKMeans()</a>
--------	----------------------------------

### Method Summary

static void	<a href="#">main</a> (java.lang.String[] args) Used for testing the mining function outside the PMBS Engine
java.lang.String	<a href="#">minePatterns</a> (java.lang.String[] args, java.sql.Connection conn) Start the process of rule extraction and store the result in the Pattern Base.

#### Methods inherited from class [com.psychofree.mining.MiningFunction](#)

[createOutputTable](#), [minePatterns](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Constructors

### UsobasicKMeans

public **UsobasicKMeans**()

## Methods

### minePatterns

```

public java.lang.String minePatterns(java.lang.String[] args,
    java.sql.Connection conn)
    throws java.sql.SQLException,
    java.lang.Exception
  
```

Start the process of rule extraction and store the result in the Pattern Base.

(continued from last page)

**Parameters:**

args - The String array containing the mining function parameters  
conn - The database connection

---

**main**

```
public static void main(java.lang.String[] args)
```

Used for testing the mining function outside the PMBS Engine

---

Package

**com.psychofree.statements**

## com.psychofree.statements

### Class CreateClass

```

java.lang.Object
  |
  +- com.psychofree.statements.PBMSStatement
     |
     +- com.psychofree.statements.PDL_PMLStatement
        |
        +- com.psychofree.statements.CreateClass
  
```

#### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

```

public class CreateClass
extends PDL\_PMLStatement
  
```

This class wraps a CREATE CLASS statement of PDL creating a new class based on an existing pattern type

#### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

### Constructor Summary

public	<a href="#">CreateClass</a> (java.sql.Connection conn, java.lang.String cmd) Creates a new instance of CreateClass
public	<a href="#">CreateClass</a> (java.lang.String className, java.lang.String PatternTypeName) Creates a new instance of CreateClass

### Method Summary

java.lang.String	<a href="#">interpretStatement</a> () Translates the PDL statement in a call to the PL/pgSQL function CreateClass that performs the creation
void	<a href="#">parseStatement</a> () Parse the PDL command CREATE CLASS

#### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

#### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface com.jpbc.PBStatement

close, execute, execute

#### Methods inherited from interface com.jpbc.PdlStatement

executePDLcommand, executePDLcommand

#### Methods inherited from interface com.jpbc.PBStatement

close, execute, execute

#### Methods inherited from interface com.jpbc.PmlStatement

executePMLcommand, executePMLcommand

#### Methods inherited from interface com.jpbc.PBStatement

close, execute, execute

## Constructors

### CreateClass

```
public CreateClass(java.sql.Connection conn,
                  java.lang.String cmd)
```

Creates a new instance of CreateClass

### CreateClass

```
public CreateClass(java.lang.String ClassName,
                  java.lang.String PatternTypeName)
```

Creates a new instance of CreateClass

#### Parameters:

ClassName - the class name

PatternTypeName - the patter type name of the class

## Methods

### parseStatement

```
public void parseStatement()
    throws com.jpbc.PException
```

Parse the PDL command CREATE CLASS

#### Throws:

PException - if there is any syntax error in the command

### interpretStatement

```
public java.lang.String interpretStatement()
    throws com.jpbc.PException
```

Translates the PDL statement in a call to the PL/pgSQL function CreateClass that performs the creation

(continued from last page)

**Returns:**

A message containing the outcome of the execution

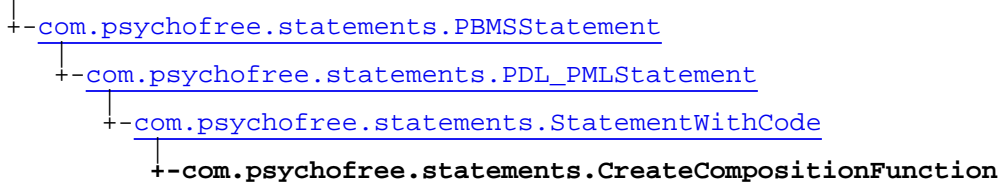
**Throws:**

StatementExecutionException - if something goes wrong

## com.psychofree.statements

# Class CreateCompositionFunction

java.lang.Object



### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

public class **CreateCompositionFunction**  
 extends [StatementWithCode](#)

This class wraps a CREATE COMPOSITION FUNCTION statement of PDL

### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

## Constructor Summary

public	<a href="#">CreateCompositionFunction</a> (java.sql.Connection conn, java.lang.String cmd) Creates a new instance of CreateCompositionFunction
--------	---

## Method Summary

java.lang.String	<a href="#">getFunctionName</a> ()
java.lang.String	<a href="#">interpretStatement</a> () A composition function is translated in a stored function of PL/pgSQL that takes in input two patterns and returns a new one combining them as described in the body of the function
void	<a href="#">parseStatement</a> () Parse the PDL command CREATE COMPOSITION FUNCTION

### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [execudePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Methods inherited from interface com.jpbc.PBStatement

close, execute, execute

#### Methods inherited from interface com.jpbc.PdlStatement

executePDLcommand, executePDLcommand

#### Methods inherited from interface com.jpbc.PBStatement

close, execute, execute

#### Methods inherited from interface com.jpbc.PmlStatement

executePMLcommand, executePMLcommand

#### Methods inherited from interface com.jpbc.PBStatement

close, execute, execute

## Constructors

### CreateCompositionFunction

```
public CreateCompositionFunction(java.sql.Connection conn,
                                java.lang.String cmd)
```

Creates a new instance of CreateCompositionFunction

## Methods

### getFunctionName

```
public java.lang.String getFunctionName()
```

**Returns:**

the name of the composition function

### parseStatement

```
public void parseStatement()
throws com.jpbc.PException
```

Parse the PDL command CREATE COMPOSITION FUNCTION

**Parameters:**

st - the StreamTokenizer containing the PDL command

**Throws:**

PException - if there is any syntax error in the command

### interpretStatement

```
public java.lang.String interpretStatement()
throws com.jpbc.PException
```

(continued from last page)

A composition function is translated in a stored function of PL/pgSQL that takes in input two patterns and returns a new one combining them as described in the body of the function

**Returns:**

A message containing the outcome of the execution

**Throws:**

`PException` - if something goes wrong

## com.psychofree.statements

### Class CreateCondition

```

java.lang.Object
  |
  +- com.psychofree.statements.PBMSStatement
     |
     +- com.psychofree.statements.PDL_PMLStatement
        |
        +- com.psychofree.statements.StatementWithCode
           |
           +- com.psychofree.statements.CreateCondition
  
```

#### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

```

public class CreateCondition
extends StatementWithCode
  
```

This class wraps a CREATE CONDITION statement of PML

#### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

### Constructor Summary

public	<a href="#">CreateCondition</a> (java.sql.Connection conn, java.lang.String cmd) Creates a new instance of CreateClass
--------	---

### Method Summary

java.lang.String	<a href="#">interpretStatement</a> () A condition is translated in a stored function of PL/pgSQL that takes in input a pattern and returns an integer.
------------------	---

void	<a href="#">parseStatement</a> () Parse the PDL command CREATE CONDITION
------	---

#### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

#### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

**Methods inherited from interface** com.jpbc.PdlStatement

executePDLcommand, executePDLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

**Methods inherited from interface** com.jpbc.PmlStatement

executePMLcommand, executePMLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

## Constructors

### CreateCondition

```
public CreateCondition(java.sql.Connection conn,
                      java.lang.String cmd)
```

Creates a new instance of CreateClass

## Methods

### parseStatement

```
public void parseStatement()
    throws com.jpbc.PException
```

Parse the PDL command CREATE CONDITION

**Throws:**

PException - if there is any syntax error in the PDL command

### interpretStatement

```
public java.lang.String interpretStatement()
    throws com.jpbc.PException
```

A condition is translated in a stored function of PL/pgSQL that takes in input a pattern and returns an integer.

**Returns:**

A message containing the outcome of the execution.

## com.psychofree.statements

### Class CreateFormula

```

java.lang.Object
  |
  +- com.psychofree.statements.PBMSStatement
     |
     +- com.psychofree.statements.PDL_PMLStatement
        |
        +- com.psychofree.statements.StatementWithCode
           |
           +- com.psychofree.statements.CreateFormula
  
```

#### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

```

public class CreateFormula
extends StatementWithCode
  
```

Creates a new formula function for a pattern type pt. The formula is implemented with a stored PL/pgSQL function which takes in input an instance of pt and a data source name and returns the name of a temporary table containing the results

#### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

### Constructor Summary

public	<a href="#">CreateFormula</a> (java.sql.Connection conn, java.lang.String cmd)
--------	--

### Method Summary

java.lang.String	<a href="#">interpretStatement</a> () Translates the formula definition in a PL/pgSQL function
------------------	---

void	<a href="#">parseStatement</a> () Parse the formula definition
------	---

#### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#),  
[getStatementType](#)

#### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

**Methods inherited from interface** `com.jpbc.PdlStatement``executePDLcommand, executePDLcommand`**Methods inherited from interface** `com.jpbc.PBStatement``close, execute, execute`**Methods inherited from interface** `com.jpbc.PmlStatement``executePMLcommand, executePMLcommand`**Methods inherited from interface** `com.jpbc.PBStatement``close, execute, execute`

## Constructors

### CreateFormula

```
public CreateFormula(java.sql.Connection conn,
                    java.lang.String cmd)
```

## Methods

### parseStatement

```
public void parseStatement()
    throws com.jpbc.PException
```

Parse the formula definition

**Throws:**

`PException` - if there is any syntax error in the command

### interpretStatement

```
public java.lang.String interpretStatement()
    throws com.jpbc.PException
```

Translates the formula definition in a PL/pgSQL function

**Returns:**

A message containing the outcome of the execution

**Throws:**

`PException` - if something goes wrong

## com.psychofree.statements Class CreateJoinCondition

```

java.lang.Object
  |
  +- com.psychofree.statements.PBMSStatement
      |
      +- com.psychofree.statements.PDL_PMLStatement
          |
          +- com.psychofree.statements.StatementWithCode
              |
              +- com.psychofree.statements.CreateJoinCondition
  
```

### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

```

public class CreateJoinCondition
extends StatementWithCode
  
```

This class wraps a CREATE JOIN CONDITION statement of PML

### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

## Constructor Summary

public	<a href="#">CreateJoinCondition</a> (java.sql.Connection conn, java.lang.String cmd) Creates a new instance of CreateClass
--------	---

## Method Summary

java.lang.String	<a href="#">interpretStatement</a> () A join condition is translated in a stored function of PL/pgSQL that takes in input two patterns and returns a boolean
------------------	---

void	<a href="#">parseStatement</a> () Parse the PDL command CREATE JOIN CONDITION
------	--

### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Methods inherited from interface com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

**Methods inherited from interface** com.jpbc.PdlStatement

executePDLcommand, executePDLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

**Methods inherited from interface** com.jpbc.PmlStatement

executePMLcommand, executePMLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

## Constructors

### CreateJoinCondition

```
public CreateJoinCondition(java.sql.Connection conn,
                          java.lang.String cmd)
```

Creates a new instance of CreateClass

## Methods

### parseStatement

```
public void parseStatement()
    throws com.jpbc.PException
```

Parse the PDL command CREATE JOIN CONDITION

**Throws:**

PException - if there is any syntax error in the command

### interpretStatement

```
public java.lang.String interpretStatement()
    throws com.jpbc.PException
```

A join condition is translated in a stored function of PL/pgSQL that takes in input two patterns and returns a boolean

**Returns:**

A message containing the outcome of the execution

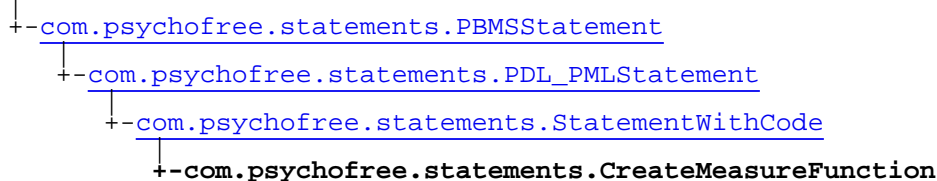
**Throws:**

PException - if something goes wrong

## com.psychofree.statements

### Class CreateMeasureFunction

java.lang.Object



#### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

public class **CreateMeasureFunction**  
 extends [StatementWithCode](#)

This class wraps a CREATE MEASURE FUNCTION statement of PDL

#### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

### Constructor Summary

public	<a href="#">CreateMeasureFunction</a> (java.sql.Connection conn, java.lang.String cmd) Creates a new instance of CreateMiningFunction
--------	--

### Method Summary

java.lang.String	<a href="#">interpretStatement</a> () A measure function is translated in a stored function of PL/pgSQL with exactly the behaviour defined in the body and that returns a measure object
void	<a href="#">parseStatement</a> () Parse the PDL command CREATE MEASURE FUNCTION

#### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

#### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

**Methods inherited from interface** `com.jpbc.PdlStatement`

executePDLcommand, executePDLcommand

**Methods inherited from interface** `com.jpbc.PBStatement`

close, execute, execute

**Methods inherited from interface** `com.jpbc.PmlStatement`

executePMLcommand, executePMLcommand

**Methods inherited from interface** `com.jpbc.PBStatement`

close, execute, execute

## Constructors

### CreateMeasureFunction

```
public CreateMeasureFunction(java.sql.Connection conn,
                             java.lang.String cmd)
```

Creates a new instance of CreateMiningFunction

## Methods

### parseStatement

```
public void parseStatement()
    throws com.jpbc.PException
```

Parse the PDL command CREATE MEASURE FUNCTION

**Parameters:**

st - the StreamTokenizer containing the PDL command

**Throws:**

PException - if there is any syntax error in the command

### interpretStatement

```
public java.lang.String interpretStatement()
    throws com.jpbc.PException
```

A measure function is translated in a stored function of PL/pgSQL with exactly the behaviour defined in the body and that returns a measure object

**Returns:**

A message containing the outcome of the execution

**Throws:**

StatementExecutionException - if something goes wrong

## com.psychofree.statements

### Class CreatePatternType

```

java.lang.Object
  |
  +- com.psychofree.statements.PBMSStatement
     |
     +- com.psychofree.statements.PDL_PMLStatement
        |
        +- com.psychofree.statements.StatementWithCode
           |
           +- com.psychofree.statements.CreatePatternType
  
```

#### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

```

public class CreatePatternType
extends StatementWithCode
  
```

This class wraps a CREATE PATTERN TYPE statement of PDL

#### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

### Constructor Summary

public	<a href="#">CreatePatternType</a> (java.sql.Connection conn, java.lang.String cmd) Creates a new instance of CreatePatternType
public	<a href="#">CreatePatternType</a> (java.sql.Connection conn, java.lang.String patternName, java.io.StreamTokenizer st)

### Method Summary

java.lang.String	<a href="#">interpretStatement</a> () The creation of a new pattern type is translated into several SQL instructions, that are the creation of the types (also those corresponding to the structure and the measure), the declaration of the Equals and Theta functions and the creation of the table of that pattern type
void	<a href="#">parseStatement</a> () Parse the PDL command CREATE PATTERN TYPE

#### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

#### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

**Methods inherited from interface** com.jpbc.PdlStatement

executePDLcommand, executePDLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

**Methods inherited from interface** com.jpbc.PmlStatement

executePMLcommand, executePMLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

## Constructors

### CreatePatternType

```
public CreatePatternType(java.sql.Connection conn,
                        java.lang.String cmd)
```

Creates a new instance of CreatePatternType

### CreatePatternType

```
public CreatePatternType(java.sql.Connection conn,
                        java.lang.String patternName,
                        java.io.StreamTokenizer st)
```

## Methods

### parseStatement

```
public void parseStatement()
    throws com.jpbc.PException
```

Parse the PDL command CREATE PATTERN TYPE

**Parameters:**

st - the StreamTokenizer containing the PDL command

**Throws:**

PException - if there is any syntax error in the command

(continued from last page)

## **interpretStatement**

```
public java.lang.String interpretStatement()  
    throws com.jpbc.PException
```

The creation of a new pattern type is translated into several SQL instructions, that are the creation of the types (also those corresponding to the structure and the measure), the declaration of the Equals and Theta functions and the creation of the table of that pattern type

**Returns:**

A message containing the outcome of the execution

**Throws:**

`PException` - if something goes wrong

## com.psychofree.statements Class DataCovering

```

java.lang.Object
  |
  +- com.psychofree.statements.PBMSStatement
      |
      +- com.psychofree.statements.PQLStatement
          |
          +- com.psychofree.statements.DataCovering
  
```

### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PqlStatement

```

public class DataCovering
extends PQLStatement
  
```

This class wraps a DATA COVERING statement of PQL

### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

### Constructor Summary

public	<a href="#">DataCovering</a> (java.sql.Connection conn, java.lang.String cmd) Creates a new instance of DataCovering
--------	---

### Method Summary

java.lang.String	<a href="#">getClassName</a> ()
java.lang.String	<a href="#">getCondition</a> ()
java.util.Vector	<a href="#">getCondParams</a> ()
java.lang.String	<a href="#">getDatasource</a> ()
<a href="#">Select</a>	<a href="#">getInnerSelect</a> ()
java.lang.String	<a href="#">getStoreAs</a> ()
com.jpbc.PResultSet	<a href="#">interpretStatement</a> () Translates the PQL command in a call to a PL/pgSQL proedure that performs the query
boolean	<a href="#">isSelect</a> ()
void	<a href="#">parseStatement</a> () Parse the PQL command DATA COVERING

### Methods inherited from class [com.psychofree.statements.PQLStatement](#)

[executePQLcommand](#), [executePQLcommand](#), [getStatementType](#), [interpretStatement](#)

**Methods inherited from class [com.psychofree.statements.PBMSStatement](#)**[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)**Methods inherited from class [java.lang.Object](#)**[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)**Methods inherited from interface [com.jpbc.PBStatement](#)**[close](#), [execute](#), [execute](#)**Methods inherited from interface [com.jpbc.PqlStatement](#)**[executePQLcommand](#), [executePQLcommand](#)**Methods inherited from interface [com.jpbc.PBStatement](#)**[close](#), [execute](#), [execute](#)

## Constructors

### DataCovering

```
public DataCovering(java.sql.Connection conn,
                    java.lang.String cmd)
```

Creates a new instance of DataCovering

## Methods

### parseStatement

```
public void parseStatement()
    throws com.jpbc.PException
```

Parse the PQL command DATA COVERING

**Parameters:**

st - the StreamTokenizer containing the PQL command

**Throws:**

PException - if there is any syntax error in the command

### getClassName

```
public java.lang.String getClassName()
```

### getCondition

```
public java.lang.String getCondition()
```

## getSource

```
public java.lang.String getSource()
```

---

## getStoreAs

```
public java.lang.String getStoreAs()
```

---

## isSelect

```
public boolean isSelect()
```

---

## getInnerSelect

```
public Select getInnerSelect()
```

---

## getCondParams

```
public java.util.Vector getCondParams()
```

---

## interpretStatement

```
public com.jpbc.PResultSet interpretStatement()  
    throws com.jpbc.PException
```

Translates the PQL command in a call to a PL/pgSQL procedure that performs the query

**Returns:**

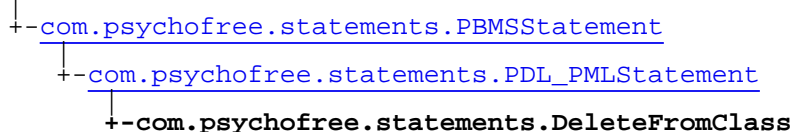
The result of the query

**Throws:**

`PException` - if something goes wrong

## com.psychofree.statements Class DeleteFromClass

java.lang.Object



### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

public class **DeleteFromClass**  
extends [PDL\\_PMLStatement](#)

This class wraps a DELETE FROM CLASS statement of PML

### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

### Constructor Summary

public	<a href="#">DeleteFromClass</a> (java.sql.Connection conn, java.lang.String cmd) Creates a new instance of CreateClass
public	<a href="#">DeleteFromClass</a> (java.lang.String ClassName, java.lang.String Condition)

### Method Summary

java.lang.String	<a href="#">getAlias</a> ()
java.lang.String	<a href="#">getClassName</a> ()
java.lang.String	<a href="#">getCondition</a> ()
java.lang.Object	<a href="#">interpretStatement</a> () Translates the PML command in a call to the PL/pgSQL procedure ClassDelete that performs the deletion
void	<a href="#">parseStatement</a> () Parse the PDL command DELETE FROM CLASS

### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

**Methods inherited from class** java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

**Methods inherited from interface** com.jpbc.PdlStatement

executePDLcommand, executePDLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

**Methods inherited from interface** com.jpbc.PmlStatement

executePMLcommand, executePMLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

## Constructors

### DeleteFromClass

```
public DeleteFromClass(java.sql.Connection conn,
                      java.lang.String cmd)
```

Creates a new instance of CreateClass

### DeleteFromClass

```
public DeleteFromClass(java.lang.String ClassName,
                      java.lang.String Condition)
```

## Methods

### getClassName

```
public java.lang.String getClassName()
```

### getCondition

```
public java.lang.String getCondition()
```

---

(continued from last page)

## **getAlias**

```
public java.lang.String getAlias()
```

---

## **parseStatement**

```
public void parseStatement()  
    throws com.jpbc.PException
```

Parse the PDL command DELETE FROM CLASS

**Throws:**

PException - if there is any syntax error in the command

---

## **interpretStatement**

```
public java.lang.Object interpretStatement()  
    throws com.jpbc.PException
```

Translates the PML command in a call to the PL/pgSQL procedure ClassDelete that performs the deletion

**Returns:**

The number of deleted patterns

**Throws:**

StatementExecutionException - if something goes wrong

---

## com.psychofree.statements

### Class DeletePatterns

```

java.lang.Object
  |
  +- com.psychofree.statements.PBMSStatement
      |
      +- com.psychofree.statements.PDL_PMLStatement
          |
          +- com.psychofree.statements.DeletePatterns
  
```

#### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

```

public class DeletePatterns
extends PDL_PMLStatement
  
```

This class wraps a DELETE PATTERNS statement of PML

#### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

### Constructor Summary

public	<a href="#">DeletePatterns</a> (java.sql.Connection conn, java.lang.String cmd) Creates a new instance of NewClass
--------	---

### Method Summary

java.lang.String	<a href="#">getAlias</a> ()
java.lang.String	<a href="#">getCondition</a> ()
java.lang.String	<a href="#">getPatternTypeName</a> ()
java.lang.Object	<a href="#">interpretStatement</a> () Translates the PML command in a call to the PL/pgSQL function Deletion that performs the deletion
void	<a href="#">parseStatement</a> () Parse the PDL command DELETE PATTERNS

#### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

#### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

#### Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

**Methods inherited from interface** `com.jpbc.PBStatement`

```
close, execute, execute
```

**Methods inherited from interface** `com.jpbc.PdlStatement`

```
executePDLcommand, executePDLcommand
```

**Methods inherited from interface** `com.jpbc.PBStatement`

```
close, execute, execute
```

**Methods inherited from interface** `com.jpbc.PmlStatement`

```
executePMLcommand, executePMLcommand
```

**Methods inherited from interface** `com.jpbc.PBStatement`

```
close, execute, execute
```

## Constructors

### DeletePatterns

```
public DeletePatterns(java.sql.Connection conn,  
                      java.lang.String cmd)
```

Creates a new instance of NewClass

## Methods

### getPatternTypeName

```
public java.lang.String getPatternTypeName()
```

### getCondition

```
public java.lang.String getCondition()
```

### getAlias

```
public java.lang.String getAlias()
```

### parseStatement

```
public void parseStatement()  
    throws com.jpbc.PException
```

(continued from last page)

Parse the PDL command DELETE PATTERNS

**Throws:**

`ParseException` - if there is any syntax error in the command

---

**interpretStatement**

```
public java.lang.Object interpretStatement()  
    throws com.jpbc.ParseException
```

Translates the PML command in a call to the PL/pgSQL function Deletion that performs the deletion

**Returns:**

The number of deleted patterns

**Throws:**

`ParseException` - if something goes wrong

## com.psychofree.statements Class DirectInsertPattern

```

java.lang.Object
  +- com.psychofree.statements.PBMSStatement
    +- com.psychofree.statements.PDL_PMLStatement
      +- com.psychofree.statements.DirectInsertPattern
  
```

### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

```

public class DirectInsertPattern
extends PDL\_PMLStatement
  
```

This class wraps a DIRECT INSERT PATTERNS statement of PML

### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

## Constructor Summary

public	<a href="#">DirectInsertPattern</a> (java.sql.Connection conn, java.lang.String cmd)
--------	--

## Method Summary

java.lang.String	<a href="#">getDataSourceName</a> ()
------------------	--------------------------------------

java.lang.String	<a href="#">getIntoClass</a> ()
------------------	---------------------------------

java.lang.String	<a href="#">getPatternTypeName</a> ()
------------------	---------------------------------------

java.lang.String	<a href="#">getStructureValue</a> ()
------------------	--------------------------------------

java.lang.String	<a href="#">getThresoldDefinition</a> ()
------------------	--

java.lang.String	<a href="#">getTsFinal</a> ()
------------------	-------------------------------

java.lang.String	<a href="#">getTsInit</a> ()
------------------	------------------------------

java.lang.Object	<a href="#">interpretStatement</a> ()
------------------	---------------------------------------

Translates the PML command in a call to the PL/pgSQL function Insertion that performs the insertion

void	<a href="#">parseStatement</a> ()
------	-----------------------------------

Parse the PML command DIRECT INSERT PATTERN

### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#),  
[getStatementType](#)

**Methods inherited from class** [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

**Methods inherited from class** java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

**Methods inherited from interface** com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

**Methods inherited from interface** com.jpbc.PdlStatement

[executePDLcommand](#), [executePDLcommand](#)

**Methods inherited from interface** com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

**Methods inherited from interface** com.jpbc.PmlStatement

[executePMLcommand](#), [executePMLcommand](#)

**Methods inherited from interface** com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

## Constructors

### DirectInsertPattern

```
public DirectInsertPattern(java.sql.Connection conn,
                           java.lang.String cmd)
```

## Methods

### getPatternTypeName

```
public java.lang.String getPatternTypeName()
```

### getDataSourceName

```
public java.lang.String getDataSourceName()
```

(continued from last page)

---

## getTsInit

```
public java.lang.String getTsInit()
```

---

## getTsFinal

```
public java.lang.String getTsFinal()
```

---

## getThresoldDefinition

```
public java.lang.String getThresoldDefinition()
```

---

## getStructureValue

```
public java.lang.String getStructureValue()
```

---

## getIntoClass

```
public java.lang.String getIntoClass()
```

---

## parseStatement

```
public void parseStatement()  
    throws com.jpbc.PException
```

Parse the PML command DIRECT INSERT PATTERN

**Throws:**

PException - if there is any syntax error in the command

---

## interpretStatement

```
public java.lang.Object interpretStatement()  
    throws com.jpbc.PException
```

Translates the PML command in a call to the PL/pgSQL function Insertion that performs the insertion

**Returns:**

the pid of the new pattern

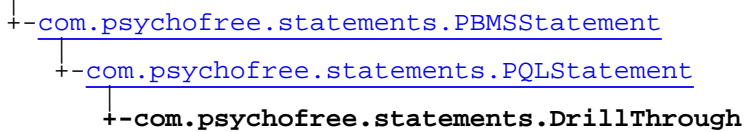
**Throws:**

PException - if something goes wrong

---

## com.psychofree.statements Class DrillThrough

java.lang.Object



### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PqlStatement

public class **DrillThrough**  
extends [PQLStatement](#)

This class wraps a DRILL THROUGH statement of PQL

### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

## Constructor Summary

public	<a href="#">DrillThrough</a> (java.sql.Connection conn, java.lang.String cmd) Creates a new instance of DrillThrough
--------	---

## Method Summary

java.lang.String	<a href="#">getClassName</a> ()
------------------	---------------------------------

java.lang.String	<a href="#">getCondition</a> ()
------------------	---------------------------------

java.util.Vector	<a href="#">getCondParams</a> ()
------------------	----------------------------------

<a href="#">Select</a>	<a href="#">getInnerSelect</a> ()
------------------------	-----------------------------------

java.lang.String	<a href="#">getTableName</a> ()
------------------	---------------------------------

com.jpbc.PResultSet	<a href="#">interpretStatement</a> () Translates the PQL command in a call to a PL/pgSQL procedure that performs the query
---------------------	---

boolean	<a href="#">isSelect</a> ()
---------	-----------------------------

void	<a href="#">parseStatement</a> () Parse the PQL command DRILL THROUGH
------	--

### Methods inherited from class [com.psychofree.statements.PQLStatement](#)

[executePQLcommand](#), [executePQLcommand](#), [getStatementType](#), [interpretStatement](#)

### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

#### Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

#### Methods inherited from interface com.jpbc.PBStatement

`close`, `execute`, `execute`

#### Methods inherited from interface com.jpbc.PqlStatement

`executePQLcommand`, `executePQLcommand`

#### Methods inherited from interface com.jpbc.PBStatement

`close`, `execute`, `execute`

## Constructors

### DrillThrough

```
public DrillThrough(java.sql.Connection conn,  
                   java.lang.String cmd)
```

Creates a new instance of DrillThrough

## Methods

### getCondition

```
public java.lang.String getCondition()
```

### getTableName

```
public java.lang.String getTableName()
```

### getClassName

```
public java.lang.String getClassName()
```

### getInnerSelect

```
public Select getInnerSelect()
```

(continued from last page)

## isSelect

```
public boolean isSelect()
```

---

## getCondParams

```
public java.util.Vector getCondParams()
```

---

## parseStatement

```
public void parseStatement()  
    throws com.jpbc.PException
```

Parse the PQL command DRILL THROUGH

**Throws:**

PException - if there is any syntax error in the command

---

## interpretStatement

```
public com.jpbc.PResultSet interpretStatement()  
    throws com.jpbc.PException
```

Translates the PQL command in a call to a PL/pgSQL procedure that performs the query

**Returns:**

The result of the query

**Throws:**

PException - if something goes wrong

---

## com.psychofree.statements

### Class DropClass

```

java.lang.Object
  |
  +- com.psychofree.statements.PBMSStatement
     |
     +- com.psychofree.statements.PDL_PMLStatement
        |
        +- com.psychofree.statements.DropClass
  
```

#### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

```

public class DropClass
extends PDL_PMLStatement
  
```

This class wraps a DROP CLASS statement of PML

#### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

### Constructor Summary

public	<a href="#">DropClass</a> (java.sql.Connection conn, java.lang.String cmd) Creates a new instance of DropClass
--------	---

### Method Summary

java.lang.String	<a href="#">getName</a> ()
------------------	----------------------------

java.lang.String	<a href="#">interpretStatement</a> () Translates the PML statement in a call in a call to the PL/pgSQL stored function DeleteClass that performs the deletion
------------------	--

void	<a href="#">parseStatement</a> () Parse the PDL command DROP CLASS
------	---

#### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

#### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

**Methods inherited from interface** com.jpbc.PdlStatement

executePDLcommand, executePDLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

**Methods inherited from interface** com.jpbc.PmlStatement

executePMLcommand, executePMLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

## Constructors

### DropClass

```
public DropClass(java.sql.Connection conn,
                 java.lang.String cmd)
```

Creates a new instance of DropClass

## Methods

### getName

```
public java.lang.String getName()
```

### parseStatement

```
public void parseStatement()
    throws com.jpbc.PException
```

Parse the PDL command DROP CLASS

**Throws:**

ParseError - if there is any syntax error in the command

### interpretStatement

```
public java.lang.String interpretStatement()
    throws com.jpbc.PException
```

Translates the PML statement in a call in a call to the PL/pgSQL stored function DeleteClass that performs the deletion

**Returns:**

A message containing the outcome of the execution

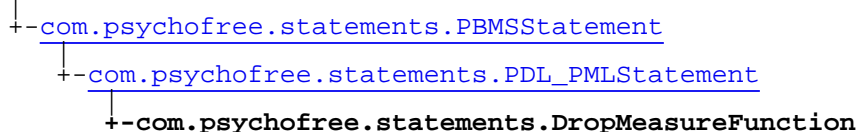
**Throws:**

StatementExecutionException - if something goes wrong

## com.psychofree.statements

# Class DropMeasureFunction

java.lang.Object



### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

public class **DropMeasureFunction**

extends [PDL\\_PMLStatement](#)

This class wraps a DROP MEASURE FUNCTION statement of PML

### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

## Constructor Summary

public	<a href="#">DropMeasureFunction</a> (java.sql.Connection conn, java.lang.String cmd) Creates a new instance of DropMiningFunction
--------	--

## Method Summary

java.lang.String	<a href="#">getName</a> ()
------------------	----------------------------

java.lang.String	<a href="#">interpretStatement</a> () Deletes the stored function which implements the measure function then updates the system catalogue PBMSMeasureFunctions deleting the relative entry
------------------	---

void	<a href="#">parseStatement</a> () Parse the PML command DROP MEASURE FUNCTION
------	--

### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Methods inherited from interface com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

**Methods inherited from interface** com.jpbc.PdlStatement

executePDLcommand, executePDLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

**Methods inherited from interface** com.jpbc.PmlStatement

executePMLcommand, executePMLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

## Constructors

### DropMeasureFunction

```
public DropMeasureFunction(java.sql.Connection conn,
                           java.lang.String cmd)
```

Creates a new instance of DropMiningFunction

## Methods

### getName

```
public java.lang.String getName()
```

### parseStatement

```
public void parseStatement()
    throws com.jpbc.PException
```

Parse the PML command DROP MEASURE FUNCTION

**Throws:**

WrongKeywordException - if there is any syntax error in the command

### interpretStatement

```
public java.lang.String interpretStatement()
    throws com.jpbc.PException
```

Deletes the stored function which implements the measure function then updates the system catalogue PBMSMeasureFunctions deleting the relative entry

**Returns:**

A message containing the outcome of the execution

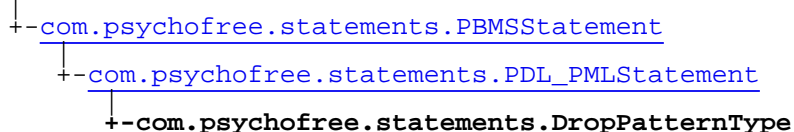
**Throws:**

StatementExecutionException - if something goes wrong

## com.psychofree.statements

### Class DropPatternType

java.lang.Object



#### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

```
public class DropPatternType
extends PDL_PMLStatement
```

This class wraps a DROP PATTERN TYPE statement of PML

#### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

### Constructor Summary

public	<a href="#">DropPatternType</a> (java.sql.Connection conn, java.lang.String cmd) Creates a new instance of CreatePatternType
--------	---

### Method Summary

java.lang.String	<a href="#">getName</a> () Getter & Setter
------------------	---

java.lang.String	<a href="#">interpretStatement</a> () The deletion of a pattern type is translated in the deletion of the three base types defined and the typed classes created
------------------	---

void	<a href="#">parseStatement</a> () Parse the PML command DROP PATTERN TYPE
------	--

#### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

#### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

**Methods inherited from interface** com.jpbc.PdlStatement

executePDLcommand, executePDLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

**Methods inherited from interface** com.jpbc.PmlStatement

executePMLcommand, executePMLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

## Constructors

### DropPatternType

```
public DropPatternType(java.sql.Connection conn,
                      java.lang.String cmd)
```

Creates a new instance of CreatePatternType

## Methods

### getName

```
public java.lang.String getName()
```

Getter & Setter

### parseStatement

```
public void parseStatement()
throws com.jpbc.PException
```

Parse the PML command DROP PATTERN TYPE

**Throws:**

ParseError - if there is any syntax error in the command

### interpretStatement

```
public java.lang.String interpretStatement()
throws com.jpbc.PException
```

The deletion of a pattern type is translated in the deletion of the three base types defined and the typed classes created

**Returns:**

A message containing the outcome of the execution

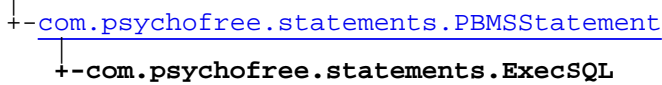
**Throws:**

StatementExecutionException - if something goes wrong

## com.psychofree.statements

# Class ExecSQL

java.lang.Object



### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement

public class **ExecSQL**  
 extends [PBMSStatement](#)

This class wraps an EXEC SQL statement, it executes pure SQL or PL/pgSQL code

### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

## Constructor Summary

public	<a href="#">ExecSQL</a> (java.sql.Connection conn, java.lang.String cmd) Creates a new instance of ExecSQL
--------	---

## Method Summary

java.lang.String	<a href="#">getSQLCode</a> ()
int	<a href="#">getStatementType</a> ()
java.lang.Object	<a href="#">interpretStatement</a> () Execute the SQL or PL/pgSQL statement
void	<a href="#">parseStatement</a> ()

### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Methods inherited from interface com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

## Constructors

(continued from last page)

## ExecSQL

```
public ExecSQL(java.sql.Connection conn,  
               java.lang.String cmd)
```

Creates a new instance of ExecSQL

## Methods

### getSQLCode

```
public java.lang.String getSQLCode()
```

---

### parseStatement

```
public void parseStatement()  
    throws com.jpbc.PException
```

This method parse a PBMSStatement from input

---

### interpretStatement

```
public java.lang.Object interpretStatement()  
    throws com.jpbc.PException
```

Execute the SQL or PL/pgSQL statement

**Returns:**

An object containing the query result

**Throws:**

PException - if there was any error in the statement execution

---

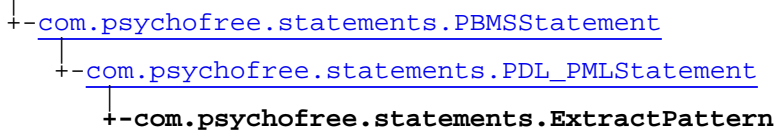
### getStatementType

```
public int getStatementType()
```

## com.psychofree.statements

### Class ExtractPattern

java.lang.Object



#### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

public class **ExtractPattern**  
 extends [PDL\\_PMLStatement](#)

This class wraps an EXTRACT PATTERNS statement of PML

#### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

#### Constructor Summary

public	<a href="#">ExtractPattern</a> (java.sql.Connection conn, java.lang.String cmd) Creates a new instance of ExtractPatternType
--------	---

#### Method Summary

java.lang.String	<a href="#">getAlias</a> ()
java.lang.String	<a href="#">getCondition</a> ()
java.lang.String	<a href="#">getDataSourceName</a> ()
java.lang.String	<a href="#">getExtParams</a> ()
java.lang.String	<a href="#">getIntoClass</a> ()
java.lang.String	<a href="#">getMeasureFunction</a> ()
java.util.ArrayList	<a href="#">getMeasureValues</a> ()
java.lang.String	<a href="#">getMiningFunctionName</a> ()
java.lang.String	<a href="#">getPatternTypeName</a> ()
java.lang.String	<a href="#">getThresoldDefinition</a> ()
java.lang.String	<a href="#">getTsFinal</a> ()

java.lang.String	<a href="#">getTsInit()</a>
java.lang.Object	<a href="#">interpretStatement()</a> Extract the patterns using the mining fuction specified, then process the result using the PL/pgSQL stored procedure ProcessExtractedPatterns
void	<a href="#">parseStatement()</a> Parse the PML command EXTRACT PATTERNS

#### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

#### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface [com.jpbc.PBStatement](#)

[close](#), [execute](#), [execute](#)

#### Methods inherited from interface [com.jpbc.PdlStatement](#)

[executePDLcommand](#), [executePDLcommand](#)

#### Methods inherited from interface [com.jpbc.PBStatement](#)

[close](#), [execute](#), [execute](#)

#### Methods inherited from interface [com.jpbc.PmlStatement](#)

[executePMLcommand](#), [executePMLcommand](#)

#### Methods inherited from interface [com.jpbc.PBStatement](#)

[close](#), [execute](#), [execute](#)

## Constructors

### ExtractPattern

```
public ExtractPattern(java.sql.Connection conn,
                     java.lang.String cmd)
```

Creates a new instance of ExtractPatternType

## Methods

(continued from last page)

---

**getPatternTypeName**

```
public java.lang.String getPatternTypeName()
```

---

**getDataSourceName**

```
public java.lang.String getDataSourceName()
```

---

**getMiningFunctionName**

```
public java.lang.String getMiningFunctionName()
```

---

**getCondition**

```
public java.lang.String getCondition()
```

---

**getTsInit**

```
public java.lang.String getTsInit()
```

---

**getTsFinal**

```
public java.lang.String getTsFinal()
```

---

**getThresoldDefinition**

```
public java.lang.String getThresoldDefinition()
```

---

**getMeasureValues**

```
public java.util.ArrayList getMeasureValues()
```

---

**getMeasureFunction**

```
public java.lang.String getMeasureFunction()
```

---

**getIntoClass**

```
public java.lang.String getIntoClass()
```

---

---

(continued from last page)

---

## getExtParams

```
public java.lang.String getExtParams()
```

---

## getAlias

```
public java.lang.String getAlias()
```

---

## parseStatement

```
public void parseStatement()  
    throws com.jpbc.PException
```

Parse the PML command EXTRACT PATTERNS

**Throws:**

PException - if there is any syntax error in the command

---

## interpretStatement

```
public java.lang.Object interpretStatement()  
    throws com.jpbc.PException
```

Extract the patterns using the mining function specified, then process the result using the PL/pgSQL stored procedure ProcessExtractedPatterns

**Returns:**

The number of extracted patterns

**Throws:**

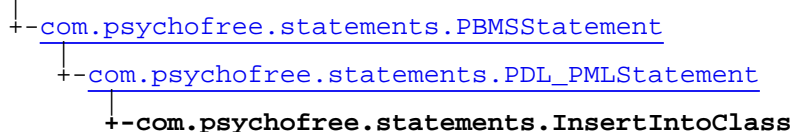
PException - if something goes wrong

---

## com.psychofree.statements

### Class InsertIntoClass

java.lang.Object



#### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

public class **InsertIntoClass**  
 extends [PDL\\_PMLStatement](#)

This class wraps an INSERT INTO CLASS statement of PML

#### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

### Constructor Summary

public	<a href="#">InsertIntoClass</a> (java.sql.Connection conn, java.lang.String cmd) Creates a new instance of CreateClass
public	<a href="#">InsertIntoClass</a> (java.lang.String ClassName, java.lang.String Condition)

### Method Summary

java.lang.String	<a href="#">getAlias</a> ()
java.lang.String	<a href="#">getClassName</a> ()
java.lang.String	<a href="#">getCondition</a> ()
java.lang.Object	<a href="#">interpretStatement</a> () Translates the PML command into a call to the PL/pgSQL procedure ClassInsert which performs the insertion
void	<a href="#">parseStatement</a> () Parse the PML command INSERT INTO CLASS

#### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

#### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

**Methods inherited from class** java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

**Methods inherited from interface** com.jpbc.PdlStatement

executePDLcommand, executePDLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

**Methods inherited from interface** com.jpbc.PmlStatement

executePMLcommand, executePMLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

## Constructors

### InsertIntoClass

```
public InsertIntoClass(java.sql.Connection conn,  
                       java.lang.String cmd)
```

Creates a new instance of CreateClass

### InsertIntoClass

```
public InsertIntoClass(java.lang.String ClassName,  
                       java.lang.String Condition)
```

## Methods

### getClassName

```
public java.lang.String getClassName()
```

### getCondition

```
public java.lang.String getCondition()
```

---

(continued from last page)

## **getAlias**

```
public java.lang.String getAlias()
```

---

## **parseStatement**

```
public void parseStatement()  
    throws com.jpbc.PException
```

Parse the PML command INSERT INTO CLASS

**Throws:**

PException - if there is any syntax error in the command

---

## **interpretStatement**

```
public java.lang.Object interpretStatement()  
    throws com.jpbc.PException
```

Translates the PML command into a call to the PL/pgSQL procedure ClassInsert which performs the insertion

**Returns:**

The number of inserted patterns

**Throws:**

PException - if something goes wrong

---

## com.psychofree.statements

### Class ParserUtilities

java.lang.Object

└-com.psychofree.statements.ParserUtilities

public class **ParserUtilities**  
 extends java.lang.Object

Contains static utilities methods used for parsing input streams

#### Constructor Summary

public	<a href="#">ParserUtilities()</a>
--------	-----------------------------------

#### Method Summary

static void	<a href="#">consumeToken</a> (java.io.StreamTokenizer st, java.lang.String token) Consumes a given token of the input stream.
static void	<a href="#">consumeToken</a> (java.io.StreamTokenizer st, java.lang.String[] token) Consumes tokens of the input stream
static <a href="#">PDL_PMLStatement</a>	<a href="#">getPdlStatementTypeFrom</a> (java.sql.Connection conn, java.lang.String cmd)
static <a href="#">PDL_PMLStatement</a>	<a href="#">getPmlStatementTypeFrom</a> (java.sql.Connection conn, java.lang.String cmd)
static <a href="#">PQLStatement</a>	<a href="#">getPqlStatementTypeFrom</a> (java.sql.Connection conn, java.lang.String cmd)
static <a href="#">PBMSStatement</a>	<a href="#">getStatementTypeFrom</a> (java.sql.Connection conn, java.lang.String cmd)
static java.lang.String	<a href="#">nextStringToken</a> (java.io.StreamTokenizer st) Retreive the next string token in the stream
static java.lang.String	<a href="#">nextToken</a> (java.io.StreamTokenizer st) Retreive the next token in the stream
static TypedField	<a href="#">parseFieldDeclaration</a> (java.io.StreamTokenizer st, java.lang.String delim) Parse a field declaration part of a command from the input stream starting with a given keyword
static TypedField	<a href="#">parseFieldDeclaration</a> (java.io.StreamTokenizer st, java.lang.String[] delim) Parse a field declaration part of a command from the input stream bounded by the given keywords.
static java.lang.String	<a href="#">parseWhere</a> (java.io.StreamTokenizer st, java.lang.String[] delim, java.lang.String alias, java.lang.String patternTypeName) Parse a where clause declaration of a command from the input stream

Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

## Constructors

### ParserUtilities

```
public ParserUtilities()
```

## Methods

### nextStringToken

```
public static java.lang.String nextStringToken(java.io.StreamTokenizer st)
```

Retrieve the next string token in the stream

**Parameters:**

st - The stream to parse

**Returns:**

The next token

### nextToken

```
public static java.lang.String nextToken(java.io.StreamTokenizer st)
```

Retrieve the next token in the stream

**Parameters:**

st - The stream to parse.

**Returns:**

The next token

### consumeToken

```
public static void consumeToken(java.io.StreamTokenizer st,  
    java.lang.String token)  
throws com.jpbc.PException
```

Consumes a given token of the input stream.

**Parameters:**

st - The stream to parse.

token - The starting keyword

**Throws:**

PException - If the current token in the stream is differnt from the given keyword

### parseFieldDeclaration

```
public static TypedField parseFieldDeclaration(java.io.StreamTokenizer st,  
    java.lang.String[] delim)
```

(continued from last page)

Parse a field declaration part of a command from the input stream bounded by the given keywords.

**Parameters:**

st - The stream to parse  
delim - Array of string keywords

**Returns:**

The parsed field

---

## parseFieldDeclaration

```
public static TypedField parseFieldDeclaration(java.io.StreamTokenizer st,  
        java.lang.String delim)
```

Parse a field declaration part of a command from the input stream starting with a given keyword

**Parameters:**

st - The stream to parse  
delim - The starting keyword

**Returns:**

The parsed field

---

## parseWhere

```
public static java.lang.String parseWhere(java.io.StreamTokenizer st,  
        java.lang.String[] delim,  
        java.lang.String alias,  
        java.lang.String patternTypeName)
```

Parse a where clause declaration of a command from the input stream

**Parameters:**

st - The stream to parse  
delim - The string keywords  
alias - The alias used in the where statement  
patternTypeName - The pattern type involved in the statement

**Returns:**

The parsed where statement

---

## consumeToken

```
public static void consumeToken(java.io.StreamTokenizer st,  
        java.lang.String[] token)  
throws com.jpbc.PException
```

Consumes tokens of the input stream

**Parameters:**

st - The stream to parse  
token - Array containing the keywords to consume

**Throws:**

PException - If the current token in the stream is different from the given keywords

(continued from last page)

---

## getStatementTypeFrom

```
public static PBMSStatement getStatementTypeFrom(java.sql.Connection conn,  
        java.lang.String cmd)  
    throws com.jpbc.PException
```

---

## getPdlStatementTypeFrom

```
public static PDL\_PMLStatement getPdlStatementTypeFrom(java.sql.Connection conn,  
        java.lang.String cmd)  
    throws com.jpbc.PException
```

---

## getPmlStatementTypeFrom

```
public static PDL\_PMLStatement getPmlStatementTypeFrom(java.sql.Connection conn,  
        java.lang.String cmd)  
    throws com.jpbc.PException
```

---

## getPqlStatementTypeFrom

```
public static PQLStatement getPqlStatementTypeFrom(java.sql.Connection conn,  
        java.lang.String cmd)  
    throws com.jpbc.PException
```

---

## com.psychofree.statements Class Pattern

java.lang.Object

└-com.psychofree.statements.Pattern

### All Implemented Interfaces:

java.io.Serializable

```
public class Pattern
extends java.lang.Object
implements java.io.Serializable
```

## Constructor Summary

public	<a href="#">Pattern()</a> Creates a new instance of Pattern
public	<a href="#">Pattern(java.lang.String patternType)</a>
public	<a href="#">Pattern(java.lang.String patternType, java.lang.String dataSource)</a>

## Method Summary

void	<a href="#">addToMeasure(java.lang.String el)</a>
void	<a href="#">addToStructure(java.lang.String el)</a>
java.lang.String	<a href="#">getDataSource()</a>
java.util.ArrayList	<a href="#">getMeasure()</a>
java.lang.String	<a href="#">getMeasureElement(int index)</a>
int	<a href="#">getMeasureSize()</a>
java.util.ArrayList	<a href="#">getStructure()</a>
java.lang.String	<a href="#">getStructureElement(int index)</a>
int	<a href="#">getStructureSize()</a>
java.lang.String	<a href="#">getType()</a>
java.lang.String[]	<a href="#">getValidity()</a>

java.lang.String	<a href="#">getValidityEnd()</a>
java.lang.String	<a href="#">getValidityStart()</a>
void	<a href="#">setDataSource</a> (java.lang.String dataSource)
void	<a href="#">setMeasure</a> (java.util.ArrayList Measure)
void	<a href="#">setStructure</a> (java.util.ArrayList Structure)
void	<a href="#">setType</a> (java.lang.String patternType)
void	<a href="#">setValidity</a> (java.lang.String start, java.lang.String end)
void	<a href="#">setValidityEnd</a> (java.lang.String end)
void	<a href="#">setValidityStart</a> (java.lang.String st)
java.lang.String	<a href="#">toString()</a>

#### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### Pattern

```
public Pattern()
```

Creates a new instance of Pattern

### Pattern

```
public Pattern(java.lang.String patternType)
```

### Pattern

```
public Pattern(java.lang.String patternType,  
              java.lang.String dataSource)
```

## Methods

### getType

```
public java.lang.String getType()
```

(continued from last page)

---

## **setType**

```
public void setType(java.lang.String patternType)
```

---

## **getStructureSize**

```
public int getStructureSize()
```

---

## **getStructure**

```
public java.util.ArrayList getStructure()
```

---

## **setDataSource**

```
public void setDataSource(java.lang.String dataSource)
```

---

## **getDataSource**

```
public java.lang.String getDataSource()
```

---

## **getStructureElement**

```
public java.lang.String getStructureElement(int index)
```

---

## **setStructure**

```
public void setStructure(java.util.ArrayList Structure)
```

---

## **addToStructure**

```
public void addToStructure(java.lang.String el)
```

---

## **getMeasureSize**

```
public int getMeasureSize()
```

---

(continued from last page)

---

## getMeasure

```
public java.util.ArrayList getMeasure()
```

---

## getMeasureElement

```
public java.lang.String getMeasureElement(int index)
```

---

## setMeasure

```
public void setMeasure(java.util.ArrayList Measure)
```

---

## addToMeasure

```
public void addToMeasure(java.lang.String el)
```

---

## setValidityStart

```
public void setValidityStart(java.lang.String st)
```

---

## setValidityEnd

```
public void setValidityEnd(java.lang.String end)
```

---

## setValidity

```
public void setValidity(java.lang.String start,  
                        java.lang.String end)
```

---

## getValidityStart

```
public java.lang.String getValidityStart()
```

---

## getValidityEnd

```
public java.lang.String getValidityEnd()
```

---

## getValidity

```
public java.lang.String[] getValidity()
```

---

(continued from last page)

---

## **toString**

```
public java.lang.String toString()
```

## com.psychofree.statements Class PatternCovering

```

java.lang.Object
  |
  +- com.psychofree.statements.PBMSStatement
     |
     +- com.psychofree.statements.PQLStatement
        |
        +- com.psychofree.statements.PatternCovering
  
```

### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PqlStatement

```

public class PatternCovering
extends PQLStatement
  
```

This class wraps a PATTERN COVERING statement of PQL

### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

### Constructor Summary

public	<a href="#">PatternCovering</a> (java.sql.Connection conn, java.lang.String cmd)
--------	--

### Method Summary

java.lang.String	<a href="#">getClassName</a> ()
java.lang.String	<a href="#">getCondition</a> ()
java.util.Vector	<a href="#">getCondParams</a> ()
java.lang.String	<a href="#">getDatasource</a> ()
<a href="#">Select</a>	<a href="#">getInnerSelect</a> ()
java.lang.String	<a href="#">getStoreAs</a> ()
com.jpbc.PResultSet	<a href="#">interpretStatement</a> () Translates the PQL command in a call to a PL/pgSQL proedure that performs the query
boolean	<a href="#">isSelect</a> ()
void	<a href="#">parseStatement</a> () Parse the PQL command PATTERN COVERING

### Methods inherited from class [com.psychofree.statements.PQLStatement](#)

[executePQLcommand](#), [executePQLcommand](#), [getStatementType](#), [interpretStatement](#)

**Methods inherited from class** [com.psychofree.statements.PBMSStatement](#)[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)**Methods inherited from class** java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

**Methods inherited from interface** com.jpbc.PqlStatement

executePQLcommand, executePQLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

## Constructors

### PatternCovering

```
public PatternCovering(java.sql.Connection conn,
                      java.lang.String cmd)
```

## Methods

### parseStatement

```
public void parseStatement()
    throws com.jpbc.PException
```

Parse the PQL command PATTERN COVERING

**Throws:**

PException - if there is any syntax error in the command

### interpretStatement

```
public com.jpbc.PResultSet interpretStatement()
    throws com.jpbc.PException
```

Translates the PQL command in a call to a PL/pgSQL procedure that performs the query

**Returns:**

The result of the query

**Throws:**

PException - if something goes wrong

(continued from last page)

**getClassName**

```
public java.lang.String getClassName()
```

---

**getCondition**

```
public java.lang.String getCondition()
```

---

**getDatasource**

```
public java.lang.String getDatasource()
```

---

**getStoreAs**

```
public java.lang.String getStoreAs()
```

---

**isSelect**

```
public boolean isSelect()
```

---

**getInnerSelect**

```
public Select getInnerSelect()
```

---

**getCondParams**

```
public java.util.Vector getCondParams()
```

---

## com.psychofree.statements Class PBMSStatement

java.lang.Object

└─com.psychofree.statements.PBMSStatement

### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement

### Direct Known Subclasses:

[ExecSQL](#), [PDL\\_PMLStatement](#), [PQLStatement](#)

public abstract class **PBMSStatement**  
 extends java.lang.Object  
 implements com.jpbc.PBStatement, java.io.Serializable

The superclass of all PSYCHOfree statements

### Field Summary

public static	<a href="#">OTHER_STATEMENT</a>
public static	<a href="#">PDL_OR_PML_STATEMENT</a>
public static	<a href="#">QUERY_STATEMENT</a>

### Constructor Summary

public	<a href="#">PBMSStatement</a> ()
public	<a href="#">PBMSStatement</a> (java.sql.Connection conn, java.lang.String cmd)

### Method Summary

void	<a href="#">close</a> ()
java.lang.Object	<a href="#">execute</a> ()
java.lang.Object	<a href="#">execute</a> (java.lang.String str)
abstract int	<a href="#">getStatementType</a> ()
abstract java.lang.Object	<a href="#">interpretStatement</a> ()
abstract void	<a href="#">parseStatement</a> () This method parse a PBMSStatement from input

### Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

**Methods inherited from interface** `com.jpbc.PBStatement`

```
close, execute, execute
```

## Fields

### **PDL\_OR\_PML\_STATEMENT**

```
public static int PDL_OR_PML_STATEMENT
```

### **QUERY\_STATEMENT**

```
public static int QUERY_STATEMENT
```

### **OTHER\_STATEMENT**

```
public static int OTHER_STATEMENT
```

## Constructors

### **PBMSStatement**

```
public PBMSStatement()
```

### **PBMSStatement**

```
public PBMSStatement(java.sql.Connection conn,  
                      java.lang.String cmd)
```

## Methods

### **parseStatement**

```
public abstract void parseStatement()  
    throws com.jpbc.PException
```

This method parse a PBMSStatement from input

**Throws:**

`PException` - If this statement's syntax is not correct

(continued from last page)

## **interpretStatement**

```
public abstract java.lang.Object interpretStatement()  
    throws com.jpbc.PException
```

---

## **getStatementType**

```
public abstract int getStatementType()
```

---

## **close**

```
public void close()
```

---

## **execute**

```
public java.lang.Object execute()  
    throws com.jpbc.PException
```

---

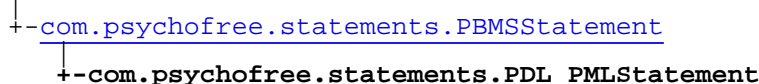
## **execute**

```
public java.lang.Object execute(java.lang.String str)  
    throws com.jpbc.PException
```

---

## com.psychofree.statements Class PDL\_PMLStatement

java.lang.Object



### All Implemented Interfaces:

com.jpbc.PmlStatement, com.jpbc.PdlStatement, java.io.Serializable, com.jpbc.PBStatement

### Direct Known Subclasses:

[CreateClass](#), [DeleteFromClass](#), [DeletePatterns](#), [DirectInsertPattern](#), [DropClass](#), [DropMeasureFunction](#), [DropPatternType](#), [ExtractPattern](#), [InsertIntoClass](#), [RecomputePattern](#), [StatementWithCode](#), [Update](#), [UpdatePattern](#)

public abstract class **PDL\_PMLStatement**

extends [PBMSStatement](#)

implements [com.jpbc.PBStatement](#), [java.io.Serializable](#), [com.jpbc.PdlStatement](#), [com.jpbc.PmlStatement](#)

General superclass for all PDL and PML statement whose execution will return a message (generally success or failure)

### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

### Constructor Summary

public	<a href="#">PDL_PMLStatement</a> ()
public	<a href="#">PDL_PMLStatement</a> (java.sql.Connection conn, java.lang.String cmd)

### Method Summary

java.lang.String	<a href="#">executePDLcommand</a> ()
java.lang.String	<a href="#">executePDLcommand</a> (java.lang.String str)
int	<a href="#">executePMLcommand</a> ()
int	<a href="#">executePMLcommand</a> (java.lang.String str)
int	<a href="#">getStatementType</a> ()

### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

### Methods inherited from class [java.lang.Object](#)

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

**Methods inherited from interface** com.jpbc.PdlStatement

executePDLcommand, executePDLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

**Methods inherited from interface** com.jpbc.PmlStatement

executePMLcommand, executePMLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

## Constructors

### PDL\_PMLStatement

```
public PDL_PMLStatement()
```

### PDL\_PMLStatement

```
public PDL_PMLStatement(java.sql.Connection conn,  
                        java.lang.String cmd)
```

## Methods

### getStatementType

```
public int getStatementType()
```

### executePDLcommand

```
public java.lang.String executePDLcommand()  
    throws com.jpbc.PException
```

### executePDLcommand

```
public java.lang.String executePDLcommand(java.lang.String str)  
    throws com.jpbc.PException
```

---

## **executePMLcommand**

```
public int executePMLcommand()  
    throws com.jpbc.PException
```

---

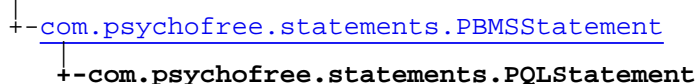
## **executePMLcommand**

```
public int executePMLcommand(java.lang.String str)  
    throws com.jpbc.PException
```

## com.psychofree.statements

# Class PQLStatement

java.lang.Object



### All Implemented Interfaces:

com.jpbc.PqlStatement, java.io.Serializable, com.jpbc.PBStatement

### Direct Known Subclasses:

[DataCovering](#), [DrillThrough](#), [PatternCovering](#), [Select](#), [Show](#)

public abstract class **PQLStatement**

extends [PBMSStatement](#)

implements com.jpbc.PBStatement, java.io.Serializable, com.jpbc.PqlStatement

General superclass for all querying statement whose execution will return a ResultSet object

### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

## Constructor Summary

public	<a href="#">PQLStatement</a> ()
public	<a href="#">PQLStatement</a> (java.sql.Connection conn, java.lang.String cmd)

## Method Summary

com.jpbc.PResultSet	<a href="#">executePQLcommand</a> ()
com.jpbc.PResultSet	<a href="#">executePQLcommand</a> (java.lang.String str)
int	<a href="#">getStatementType</a> ()
abstract com.jpbc.PResultSet	<a href="#">interpretStatement</a> ()

### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Methods inherited from interface com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

**Methods inherited from interface** com.jpbc.PqlStatement

executePQLcommand, executePQLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

## Constructors

### PQLStatement

```
public PQLStatement()
```

### PQLStatement

```
public PQLStatement(java.sql.Connection conn,  
                    java.lang.String cmd)
```

## Methods

### getStatementType

```
public int getStatementType()
```

### interpretStatement

```
public abstract com.jpbc.PResultSet interpretStatement()  
    throws com.jpbc.PException
```

### executePQLcommand

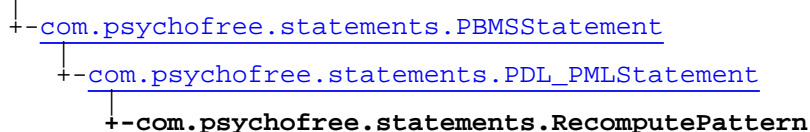
```
public com.jpbc.PResultSet executePQLcommand()  
    throws com.jpbc.PException
```

### executePQLcommand

```
public com.jpbc.PResultSet executePQLcommand(java.lang.String str)  
    throws com.jpbc.PException
```

## com.psychofree.statements Class RecomputePattern

java.lang.Object



### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

```
public class RecomputePattern
extends PDL\_PMLStatement
```

This class wraps a RECOMPUTE PATTERNS statement of PML

### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

### Constructor Summary

public	<a href="#">RecomputePattern</a> (java.sql.Connection conn, java.lang.String cmd) Creates a new instance of ExtractPatternType
--------	---

### Method Summary

java.lang.String	<a href="#">getAlias</a> ()
java.lang.String	<a href="#">getCondition</a> ()
java.lang.String	<a href="#">getDataSourceName</a> ()
java.lang.String	<a href="#">getIntoClass</a> ()
java.lang.String	<a href="#">getMeasureFunctionName</a> ()
java.lang.String	<a href="#">getPatternTypeName</a> ()
java.lang.String	<a href="#">getTsFinal</a> ()
java.lang.String	<a href="#">getTsInit</a> ()
java.lang.Object	<a href="#">interpretStatement</a> () Translates the command in a call to the PL/pgSQL stored procedure Recomputation that performs the operation
void	<a href="#">parseStatement</a> () Parse the PML command RECOMPUTE PATTERNS

**Methods inherited from class** [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#),  
[getStatementType](#)

**Methods inherited from class** [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

**Methods inherited from class** java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

**Methods inherited from interface** com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

**Methods inherited from interface** com.jpbc.PdlStatement

[executePDLcommand](#), [executePDLcommand](#)

**Methods inherited from interface** com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

**Methods inherited from interface** com.jpbc.PmlStatement

[executePMLcommand](#), [executePMLcommand](#)

**Methods inherited from interface** com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

## Constructors

### RecomputePattern

```
public RecomputePattern(java.sql.Connection conn,  
                        java.lang.String cmd)
```

Creates a new instance of ExtractPatternType

## Methods

### getPatternTypeName

```
public java.lang.String getPatternTypeName()
```

### getDataSourceName

```
public java.lang.String getDataSourceName()
```

## getCondition

```
public java.lang.String getCondition()
```

---

## getTsInit

```
public java.lang.String getTsInit()
```

---

## getTsFinal

```
public java.lang.String getTsFinal()
```

---

## getAlias

```
public java.lang.String getAlias()
```

---

## getMeasureFunctionName

```
public java.lang.String getMeasureFunctionName()
```

---

## getIntoClass

```
public java.lang.String getIntoClass()
```

---

## parseStatement

```
public void parseStatement()  
    throws com.jpbc.PException
```

Parse the PML command RECOMPUTE PATTERNS

**Throws:**

PException - if there is any syntax error in the command

---

## interpretStatement

```
public java.lang.Object interpretStatement()  
    throws com.jpbc.PException
```

Translates the command in a call to the PL/pgSQL stored procedure Recomputation that performs the operation

**Returns:**

The number of recomputed patterns

**Throws:**

PException - if something goes wrong

---

## com.psychofree.statements Class Select

```

java.lang.Object
  |
  +- com.psychofree.statements.PBMSStatement
      |
      +- com.psychofree.statements.PQLStatement
          |
          +- com.psychofree.statements.Select
  
```

### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PqlStatement

```

public class Select
extends PQLStatement
  
```

This class wraps a SELECT statement of PQL

Field Summary	
public static final	<a href="#">CJOIN</a> Value: 3
public static final	<a href="#">INTERSECT</a> Value: 2
public static final	<a href="#">NATURAL</a> Value: 0
public static final	<a href="#">UNION</a> Value: 1

Fields inherited from class <a href="#">com.psychofree.statements.PBMSStatement</a>
<a href="#">OTHER_STATEMENT</a> , <a href="#">PDL_OR_PML_STATEMENT</a> , <a href="#">QUERY_STATEMENT</a>

Constructor Summary	
public	<a href="#">Select</a> (java.sql.Connection conn, java.lang.String cmd) Creates a new instance of Select
public	<a href="#">Select</a> (java.sql.Connection conn)

Method Summary	
java.util.ArrayList	<a href="#">getClassLists</a> ()
java.lang.String	<a href="#">getClassName</a> ()
java.lang.String	<a href="#">getCompFuntionName</a> ()

java.lang.String	<a href="#">getCondition()</a>
java.lang.String	<a href="#">getFields()</a>
java.lang.String	<a href="#">getFirstAlias()</a>
java.lang.String	<a href="#">getJoinedAlias()</a>
java.lang.String	<a href="#">getJoinedClass()</a>
int	<a href="#">getJoinType()</a>
boolean	<a href="#">getOpType()</a>
java.lang.String	<a href="#">getTableName()</a>
java.lang.String	<a href="#">getTypeName()</a>
com.jpbc.PResultSet	<a href="#">interpretStatement()</a> Translates the Selection in a call to a PL/pgSQL procedure that performs the query
boolean	<a href="#">isJoin()</a>
void	<a href="#">parseStatement()</a> Parse the PQL command SELECT
void	<a href="#">parseStatement()</a> (java.io.StreamTokenizer st)

**Methods inherited from class** [com.psychofree.statements.PQLStatement](#)

[executePQLcommand](#), [executePQLcommand](#), [getStatementType](#), [interpretStatement](#)

**Methods inherited from class** [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

**Methods inherited from class** java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

**Methods inherited from interface** com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

**Methods inherited from interface** com.jpbc.PqlStatement

[executePQLcommand](#), [executePQLcommand](#)

**Methods inherited from interface** com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

## Fields

### NATURAL

```
public static final int NATURAL
```

Constant value: 0

### UNION

```
public static final int UNION
```

Constant value: 1

### INTERSECT

```
public static final int INTERSECT
```

Constant value: 2

### CJOIN

```
public static final int CJOIN
```

Constant value: 3

## Constructors

### Select

```
public Select(java.sql.Connection conn,  
              java.lang.String cmd)
```

Creates a new instance of Select

### Select

```
public Select(java.sql.Connection conn)
```

## Methods

### getFields

```
public java.lang.String getFields()
```

### getClassLists

```
public java.util.ArrayList getClassLists()
```

---

## **getCondition**

```
public java.lang.String getCondition()
```

---

## **getTableName**

```
public java.lang.String getTableName()
```

---

## **getClassName**

```
public java.lang.String getClassName()
```

---

## **getJoinedClass**

```
public java.lang.String getJoinedClass()
```

---

## **isJoin**

```
public boolean isJoin()
```

---

## **getTypeName**

```
public java.lang.String getTypeName()
```

---

## **getJoinType**

```
public int getJoinType()
```

---

## **getCompFuntionName**

```
public java.lang.String getCompFuntionName()
```

---

## **getOpType**

```
public boolean getOpType()
```

---

---

(continued from last page)

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
    throws com.jpbc.PException
```

---

## parseStatement

```
public void parseStatement()
    throws com.jpbc.PException
```

Parse the PQL command SELECT

**Throws:**

PException - if there is any syntax error in the command

---

## getJoinedAlias

```
public java.lang.String getJoinedAlias()
```

---

## getFirstAlias

```
public java.lang.String getFirstAlias()
```

---

## interpretStatement

```
public com.jpbc.PResultSet interpretStatement()
    throws com.jpbc.PException
```

Translates the Selection in a call to a PL/pgSQL procedure that performs the query

**Returns:**

The result of the query

**Throws:**

PException - if something goes wrong

---

## com.psychofree.statements

### Class Show

```

java.lang.Object
  |
  +- com.psychofree.statements.PBMSStatement
     |
     +- com.psychofree.statements.PQLStatement
        |
        +- com.psychofree.statements.Show
  
```

#### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PqlStatement

public class **Show**  
 extends [PQLStatement](#)

This class wraps a SHOW statement

#### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

### Constructor Summary

public	<a href="#">Show</a> (java.sql.Connection conn, java.lang.String cmd) Creates a new instance of Show
--------	---

### Method Summary

com.jpbc.PResultSet	<a href="#">interpretStatement</a> () Translates the SHOW statement into the correspondent SQL query
boolean	<a href="#">isIsClasses</a> ()
boolean	<a href="#">isIsPatternTypes</a> ()
void	<a href="#">parseStatement</a> () Parse the SHOW command

#### Methods inherited from class [com.psychofree.statements.PQLStatement](#)

[executePQLcommand](#), [executePQLcommand](#), [getStatementType](#), [interpretStatement](#)

#### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface [com.jpbc.PBStatement](#)

[close](#), [execute](#), [execute](#)

**Methods inherited from interface** `com.jpbc.PqlStatement``executePQLcommand, executePQLcommand`**Methods inherited from interface** `com.jpbc.PBStatement``close, execute, execute`

## Constructors

### Show

```
public Show(java.sql.Connection conn,  
             java.lang.String cmd)
```

Creates a new instance of Show

## Methods

### isIsClasses

```
public boolean isIsClasses()
```

### isIsPatternTypes

```
public boolean isIsPatternTypes()
```

### parseStatement

```
public void parseStatement()  
    throws com.jpbc.PException
```

Parse the SHOW command

**Throws:**

`PException` - if there is any syntax error in the command

### interpretStatement

```
public com.jpbc.PResultSet interpretStatement()  
    throws com.jpbc.PException
```

Translates the SHOW statement into the correspondent SQL query

**Returns:**

The resultset containing the query result

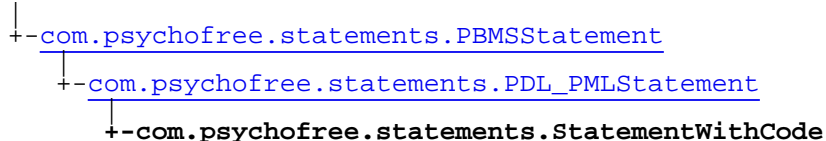
**Throws:**

`PException` - if something goes wrong

## com.psychofree.statements

### Class StatementWithCode

java.lang.Object



#### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

#### Direct Known Subclasses:

[CreateCompositionFunction](#), [CreateCondition](#), [CreateFormula](#), [CreateJoinCondition](#), [CreateMeasureFunction](#), [CreatePatternType](#)

public abstract class **StatementWithCode**

extends [PDL\\_PMLStatement](#)

General superclass for all PDL and PML statements which contain code

#### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

### Constructor Summary

public [StatementWithCode](#)(java.sql.Connection conn, java.lang.String cmd)

#### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

#### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

#### Methods inherited from interface com.jpbc.PdlStatement

[executePDLcommand](#), [executePDLcommand](#)

#### Methods inherited from interface com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

**Methods inherited from interface** com.jpbc.PmlStatement

executePMLcommand, executePMLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

---

## Constructors

### StatementWithCode

```
public StatementWithCode(java.sql.Connection conn,  
                          java.lang.String cmd)
```

## com.psychofree.statements Class Update

```

java.lang.Object
  |
  +- com.psychofree.statements.PBMSStatement
     |
     +- com.psychofree.statements.PDL_PMLStatement
        |
        +- com.psychofree.statements.Update
  
```

### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

```

public class Update
extends PDL\_PMLStatement
  
```

This class wraps an UPDATE PATTERNS statement of PML

### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

### Constructor Summary

public	<a href="#">Update</a> (java.sql.Connection conn, java.lang.String cmd)
--------	---

### Method Summary

java.lang.String	<a href="#">getAlias</a> ()
------------------	-----------------------------

java.lang.String	<a href="#">getPatternTypeName</a> ()
------------------	---------------------------------------

java.lang.Object	<a href="#">interpretStatement</a> () Translates the PML command in a call to a PL/pgSQL procedure that performs the update
------------------	--

void	<a href="#">parseStatement</a> ()
------	-----------------------------------

### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Methods inherited from interface com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

**Methods inherited from interface** com.jpbc.PdlStatement

executePDLcommand, executePDLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

**Methods inherited from interface** com.jpbc.PmlStatement

executePMLcommand, executePMLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

## Constructors

### Update

```
public Update(java.sql.Connection conn,
               java.lang.String cmd)
```

## Methods

### getPatternTypeName

```
public java.lang.String getPatternTypeName()
```

### getAlias

```
public java.lang.String getAlias()
```

### parseStatement

```
public void parseStatement()
    throws com.jpbc.PException
```

This method parse a PBMSStatement from input

### interpretStatement

```
public java.lang.Object interpretStatement()
    throws com.jpbc.PException
```

Translates the PML command in a call to a PL/pgSQL procedure that performs the update

**Returns:**

The number of updated patterns

(continued from last page)

**Throws:**

PException - if something goes wrong

## com.psychofree.statements

# Class UpdatePattern

```

java.lang.Object
  |
  +- com.psychofree.statements.PBMSStatement
     |
     +- com.psychofree.statements.PDL_PMLStatement
        |
        +- com.psychofree.statements.UpdatePattern
  
```

### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

### Direct Known Subclasses:

[UpdatePatternWithMeasureFunction](#), [UpdateValidity](#)

public abstract class **UpdatePattern**  
 extends [PDL\\_PMLStatement](#)

Updating patterns may follow three distinct ways: SYNCHRONIZE patterns, VALIDATE patterns, SET patterns VALIDITY; this is the superclass of these three different statements. An object belonging this class is owned by Update statement which should instantiate it correctly reading what the user typed in input

### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

## Constructor Summary

public	<a href="#">UpdatePattern</a> (java.sql.Connection conn, java.lang.String patternTypeName, java.lang.String alias)
--------	--

## Method Summary

java.lang.String	<a href="#">getAlias</a> ()
------------------	-----------------------------

java.lang.String	<a href="#">getCondition</a> ()
------------------	---------------------------------

java.lang.String	<a href="#">getPatternTypeName</a> ()
------------------	---------------------------------------

### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

**Methods inherited from interface** com.jpbc.PdlStatement

executePDLcommand, executePDLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

**Methods inherited from interface** com.jpbc.PmlStatement

executePMLcommand, executePMLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

## Constructors

### UpdatePattern

```
public UpdatePattern(java.sql.Connection conn,  
                    java.lang.String patternTypeName,  
                    java.lang.String alias)
```

## Methods

### getPatternTypeName

```
public java.lang.String getPatternTypeName()
```

### getAlias

```
public java.lang.String getAlias()
```

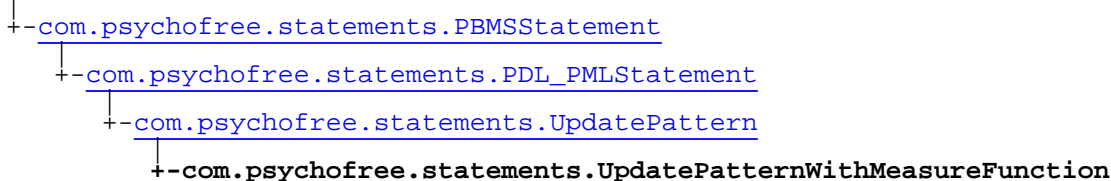
### getCondition

```
public java.lang.String getCondition()
```

## com.psychofree.statements

# Class UpdatePatternWithMeasureFunction

java.lang.Object



### All Implemented Interfaces:

[java.io.Serializable](#), [com.jpbc.PBStatement](#), [com.jpbc.PmlStatement](#), [com.jpbc.PdlStatement](#)

### Direct Known Subclasses:

[UpdateSynchronize](#), [UpdateValidate](#)

public abstract class **UpdatePatternWithMeasureFunction**  
 extends [UpdatePattern](#)

This is the superclass of UpdateValidate and UpdateValidity as they both refers to a measure function which is commonly handled here

#### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

## Constructor Summary

public	<a href="#">UpdatePatternWithMeasureFunction</a> ( <a href="#">java.sql.Connection</a> conn, <a href="#">java.lang.String</a> patternTypeName, <a href="#">java.lang.String</a> alias)
--------	--

## Method Summary

<a href="#">java.lang.String</a>	<a href="#">getFunctionName</a> ()
----------------------------------	------------------------------------

#### Methods inherited from class [com.psychofree.statements.UpdatePattern](#)

[getAlias](#), [getCondition](#), [getPatternTypeName](#)

#### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

#### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

#### Methods inherited from class [java.lang.Object](#)

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

**Methods inherited from interface** com.jpbc.PdlStatement

executePDLcommand, executePDLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

**Methods inherited from interface** com.jpbc.PmlStatement

executePMLcommand, executePMLcommand

**Methods inherited from interface** com.jpbc.PBStatement

close, execute, execute

## Constructors

### UpdatePatternWithMeasureFunction

```
public UpdatePatternWithMeasureFunction(java.sql.Connection conn,  
                                       java.lang.String patternTypeName,  
                                       java.lang.String alias)
```

## Methods

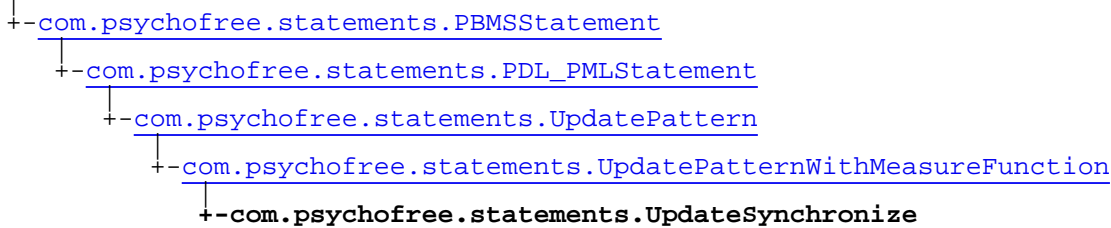
### getFunctionName

```
public java.lang.String getFunctionName()
```

## com.psychofree.statements

### Class UpdateSynchronize

java.lang.Object



#### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

```

public class UpdateSynchronize
extends UpdatePatternWithMeasureFunction
  
```

This class wraps and Update statement dealing with synchronization of patterns

#### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

#### Constructor Summary

public	<a href="#">UpdateSynchronize</a> (java.sql.Connection conn, java.lang.String patternTypeName, java.lang.String alias, java.io.StreamTokenizer st)
--------	--

#### Method Summary

java.lang.Object	<a href="#">interpretStatement</a> () Translates the PML statement in a call to the PL/pgSQL procedure Synchronize that performs patterns synchronization recomputing the measures with the specified measure function
void	<a href="#">parseStatement</a> ()

#### Methods inherited from class [com.psychofree.statements.UpdatePatternWithMeasureFunction](#)

[getFunctionName](#)

#### Methods inherited from class [com.psychofree.statements.UpdatePattern](#)

[getAlias](#), [getCondition](#), [getPatternTypeName](#)

#### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

#### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

#### Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

#### Methods inherited from interface com.jpbc.PBStatement

`close`, `execute`, `execute`

#### Methods inherited from interface com.jpbc.PdlStatement

`executePDLcommand`, `executePDLcommand`

#### Methods inherited from interface com.jpbc.PBStatement

`close`, `execute`, `execute`

#### Methods inherited from interface com.jpbc.PmlStatement

`executePMLcommand`, `executePMLcommand`

#### Methods inherited from interface com.jpbc.PBStatement

`close`, `execute`, `execute`

## Constructors

### UpdateSynchronize

```
public UpdateSynchronize(java.sql.Connection conn,
                        java.lang.String patternTypeName,
                        java.lang.String alias,
                        java.io.StreamTokenizer st)
```

## Methods

### parseStatement

```
public void parseStatement()
    throws com.jpbc.PException
```

This method parse a PBMSStatement from input

### interpretStatement

```
public java.lang.Object interpretStatement()
    throws com.jpbc.PException
```

Translates the PML statement in a call to the PL/pgSQL procedure Synchronize that performs patterns synchronization recomputing the measures with the specified measure function

#### Returns:

The number of updated patterns

(continued from last page)

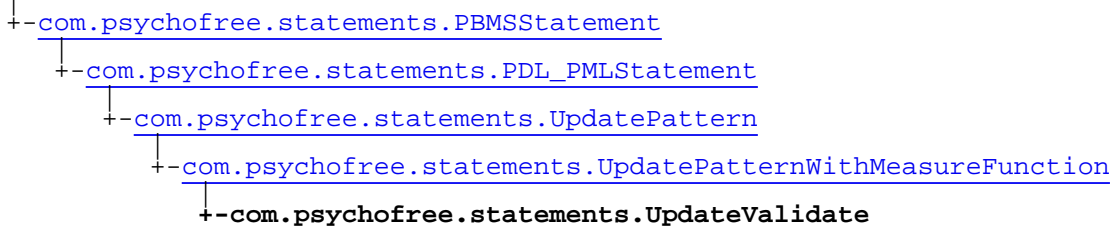
**Throws:**

PException - if something goes wrong

## com.psychofree.statements

### Class UpdateValidate

java.lang.Object



#### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

```

public class UpdateValidate
extends UpdatePatternWithMeasureFunction
  
```

This class wraps and Update statement dealing with validation of patterns

#### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

### Constructor Summary

public	<a href="#">UpdateValidate</a> (java.sql.Connection conn, java.lang.String patternTypeName, java.lang.String alias, java.io.StreamTokenizer st)
--------	---

### Method Summary

java.lang.String	<a href="#">getIntoClass</a> ()
java.lang.Object	<a href="#">interpretStatement</a> () Translates the PML statement in a call to the PL/pgSQL procedure Validate that performs patterns validation
void	<a href="#">parseStatement</a> ()

#### Methods inherited from class [com.psychofree.statements.UpdatePatternWithMeasureFunction](#)

[getFunctionName](#)

#### Methods inherited from class [com.psychofree.statements.UpdatePattern](#)

[getAlias](#), [getCondition](#), [getPatternTypeName](#)

#### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

**Methods inherited from class** [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

**Methods inherited from class** java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

**Methods inherited from interface** com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

**Methods inherited from interface** com.jpbc.PdlStatement

[executePDLcommand](#), [executePDLcommand](#)

**Methods inherited from interface** com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

**Methods inherited from interface** com.jpbc.PmlStatement

[executePMLcommand](#), [executePMLcommand](#)

**Methods inherited from interface** com.jpbc.PBStatement

[close](#), [execute](#), [execute](#)

## Constructors

### UpdateValidate

```
public UpdateValidate(java.sql.Connection conn,  
                      java.lang.String patternTypeName,  
                      java.lang.String alias,  
                      java.io.StreamTokenizer st)
```

## Methods

### getIntoClass

```
public java.lang.String getIntoClass()
```

### parseStatement

```
public void parseStatement()  
    throws com.jpbc.PException
```

This method parse a PBMSStatement from input

(continued from last page)

## **interpretStatement**

```
public java.lang.Object interpretStatement()  
    throws com.jpbc.PException
```

Translates the PML statement in a call to the PL/pgSQL procedure `Validate` that performs patterns validation

**Returns:**

The number of updated patterns

**Throws:**

`PException` - if something goes wrong

## com.psychofree.statements

# Class UpdateValidity

```

java.lang.Object
  |
  +- com.psychofree.statements.PBMSStatement
     |
     +- com.psychofree.statements.PDL_PMLStatement
        |
        +- com.psychofree.statements.UpdatePattern
           |
           +- com.psychofree.statements.UpdateValidity
  
```

### All Implemented Interfaces:

java.io.Serializable, com.jpbc.PBStatement, com.jpbc.PmlStatement, com.jpbc.PdlStatement

```

public class UpdateValidity
extends UpdatePattern
  
```

This class wraps an Update statement dealing with pattern validity period

### Fields inherited from class [com.psychofree.statements.PBMSStatement](#)

[OTHER\\_STATEMENT](#), [PDL\\_OR\\_PML\\_STATEMENT](#), [QUERY\\_STATEMENT](#)

## Constructor Summary

public	<a href="#">UpdateValidity</a> (java.sql.Connection conn, java.lang.String patternTypeName, java.lang.String alias, java.io.StreamTokenizer st)
--------	---

## Method Summary

java.lang.String	<a href="#">getTsFinal</a> ()
java.lang.String	<a href="#">getTsInit</a> ()
java.lang.Object	<a href="#">interpretStatement</a> () Translates the PML statement in a SQL update statement that performs update of the validity period
void	<a href="#">parseStatement</a> ()

### Methods inherited from class [com.psychofree.statements.UpdatePattern](#)

[getAlias](#), [getCondition](#), [getPatternTypeName](#)

### Methods inherited from class [com.psychofree.statements.PDL\\_PMLStatement](#)

[executePDLcommand](#), [executePDLcommand](#), [executePMLcommand](#), [executePMLcommand](#), [getStatementType](#)

### Methods inherited from class [com.psychofree.statements.PBMSStatement](#)

[close](#), [execute](#), [execute](#), [getStatementType](#), [interpretStatement](#), [parseStatement](#)

**Methods inherited from class** `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

**Methods inherited from interface** `com.jpbc.PBStatement`

`close`, `execute`, `execute`

**Methods inherited from interface** `com.jpbc.PdlStatement`

`executePDLcommand`, `executePDLcommand`

**Methods inherited from interface** `com.jpbc.PBStatement`

`close`, `execute`, `execute`

**Methods inherited from interface** `com.jpbc.PmlStatement`

`executePMLcommand`, `executePMLcommand`

**Methods inherited from interface** `com.jpbc.PBStatement`

`close`, `execute`, `execute`

## Constructors

### UpdateValidity

```
public UpdateValidity(java.sql.Connection conn,
                     java.lang.String patternTypeName,
                     java.lang.String alias,
                     java.io.StreamTokenizer st)
```

## Methods

### getTsInit

```
public java.lang.String getTsInit()
```

### getTsFinal

```
public java.lang.String getTsFinal()
```

### parseStatement

```
public void parseStatement()
    throws com.jpbc.PException
```

This method parse a PBMSStatement from input

## **interpretStatement**

```
public java.lang.Object interpretStatement()  
    throws com.jpbc.PException
```

Translates the PML statement in a SQL update statement that performs update of the validity period

**Returns:**

The number of updated patterns

**Throws:**

`PException` - if something goes wrong