## Package

# com.psychofree.server

# com.psychofree.server
# Class PBMSEngine

```
java.lang.Object
   |
   +-com.psychofree.server.PBMSEngine
```

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

Class implementing the PBMS engine service

## Field Summary

| | |
|---|---|
| public | inputObject |
| public | outputObject |

## Constructor Summary

| | |
|---|---|
| public | PBMSEngine(java.lang.String patternBaseHost)<br>Creates a new instance of PBMSEngine |

## Method Summary

| | |
|---|---|
| void | close() |
| PBMSStatement | readStatement(com.psychofree.utilities.PSYCHOfreeConnection clientConnection)<br>Read an object from the input stream that contains a PDL, PML or PQL statement and convert it into a PBMSStatement |
| void | runStatement(com.psychofree.utilities.PSYCHOfreeConnection clientConnection, PBMSStatement statement)<br>This method read the given statement in order to understand if it is a PDL, PML or PQL operation and then translates it into a seqence of PL/SQL operations and commands. |

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

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

## Fields

### inputObject

public java.io.ObjectInput **inputObject**

(continued on next page)

(continued from last page)

### outputObject

```
public java.io.ObjectOutput outputObject
```

# Constructors

### PBMSEngine

```
public PBMSEngine(java.lang.String patternBaseHost)
```

Creates a new instance of PBMSEngine

**Parameters:**

`patternBaseHost` - The host on which the server runs

**Throws:**

`IOException` - If internal streams used by this engine can't be initialized

# Methods

### runStatement

```
public void runStatement(com.psychofree.utilities.PSYCHOfreeConnection
clientConnection,
         PBMSStatement statement)
  throws java.io.IOException,
         java.sql.SQLException
```

This method read the given statement in order to understand if it is a PDL, PML or PQL operation and then translates it into a seqence of PL/SQL operations and commands. If the operation is a query, a new window containing the selected items is shown. Otherwise a String containing the outcome of the execution of the command is sent via the opened socket to the connected object.

**Parameters:**

`clientConnection` - The object representing the client which performed the request

`statement` - PBMSStatement The statement to be translated and executed

**Throws:**

`IOException` - If errors occur while serializing result

`SQLException` - If transaction can't be commited

---

### readStatement

```
public final PBMSStatement readStatement(com.psychofree.utilities.PSYCHOfreeConnection
clientConnection)
  throws java.io.IOException,
         java.lang.ClassNotFoundException,
         UserConnectionClosedException,
         UserNotConnectedException
```

Read an object from the input stream that contains a PDL, PML or PQL statement and convert it into a PBMSStatement

**Parameters:**

`clientConnection` - The object representing the client which performed the request

**Returns:**

The translation of the object read from the input stream, null if client close his connection

(continued on next page)

(continued from last page)

**Throws:**

`IOException` - if any error occurs while reading input
`ClassNotFoundException` - if any error occurs while reading input

---

## close

```
public void close()
```

# com.psychofree.server
# Class PSYCHOFreeServer

```
java.lang.Object
   |
   +-com.psychofree.server.PSYCHOFreeServer
```

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

Entry point for psychofree server

## Constructor Summary

| | |
|---|---|
| public | PSYCHOFreeServer() |

## Method Summary

| | |
|---|---|
| static void | main(java.lang.String[] args) |

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

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

## Constructors

### PSYCHOFreeServer

`public` **`PSYCHOFreeServer`**`()`

## Methods

### main

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

# com.psychofree.server
# Class UserConnectionClosedException

```
java.lang.Object
   |-java.lang.Throwable
      |-java.lang.Exception
         |-com.psychofree.server.UserConnectionClosedException
```

**All Implemented Interfaces:**
java.io.Serializable

---

public class **UserConnectionClosedException**
extends java.lang.Exception

This exception is thrown if client closed his connection (for example closing client window)

---

## Constructor Summary

| | |
|---|---|
| public | UserConnectionClosedException() |
| public | UserConnectionClosedException(java.lang.String msg) |

| **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

### UserConnectionClosedException

public **UserConnectionClosedException**()

---

### UserConnectionClosedException

public **UserConnectionClosedException**(java.lang.String msg)

**Parameters:**
msg - The error message

# com.psychofree.server
# Class UserNotConnectedException

```
java.lang.Object
   |-java.lang.Throwable
      |-java.lang.Exception
         |-com.psychofree.server.UserNotConnectedException
```

**All Implemented Interfaces:**
java.io.Serializable

---

public class **UserNotConnectedException**
extends java.lang.Exception

This exception is thrown if the client connection is not alive while the server is processing a request

---

## Constructor Summary

| | |
|---|---|
| public | UserNotConnectedException() |
| public | UserNotConnectedException(java.lang.String msg) |

**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

### UserNotConnectedException

`public` **`UserNotConnectedException`**`()`

---

### UserNotConnectedException

`public` **`UserNotConnectedException`**`(java.lang.String msg)`

**Parameters:**
`msg` - The error message

## 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

---

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 | CreateClass(java.sql.Connection conn)<br>Creates a new instance of CreateClass |
| public | CreateClass(java.lang.String ClassName, java.lang.String PatternTypeName)<br>Creates a new instance of CreateClass |

## Method Summary

| | |
|---|---|
| java.lang.String | getClassName()<br> |
| java.lang.String | getPatternTypeName()<br> |
| java.lang.String | interpretStatement()<br>Translates the PDL statement in a call to the PL/pgSQL function CreateClass that performs the creation |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the PDL command CREATE CLASS |

| **Methods inherited from class** com.psychofree.statements.PDL_PMLStatement |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class** com.psychofree.statements.PBMSStatement |
|---|
| getStatementType, interpretStatement, parseStatement |

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

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

# Constructors

## CreateClass

```
public CreateClass(java.sql.Connection conn)
```

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

## getClassName

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

## getPatternTypeName

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

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

Parse the PDL command CREATE CLASS

**Throws:**

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

## interpretStatement

```
public java.lang.String interpretStatement()
  throws StatementExecutionException
```

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

**Returns:**

A message containing the outcome of the execution

**Throws:**

(continued from last page)

`StatementExecutionException` - if something goes wrong

# com.psychofree.statements
# Class CreateCompositionFunction

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

**All Implemented Interfaces:**
java.io.Serializable

---

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 | CreateCompositionFunction(java.sql.Connection conn)<br>Creates a new instance of CreateCompositionFunction |

## Method Summary

| | |
|---|---|
| java.lang.String | getCode() |
| java.lang.String | getFunctionName() |
| CreatePatternType | getNewType() |
| java.util.ArrayList | getParamFields() |
| java.lang.String | getPatternName() |
| java.lang.String | getPatternTypeName() |
| java.lang.String | getPNameInput1() |
| java.lang.String | getPNameInput2() |
| java.lang.String | getPTNameInput1() |
| java.lang.String | getPTNameInput2() |

| | |
|---|---|
| java.lang.String | interpretStatement()<br>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 |
| boolean | isDefinedNew() |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the PDL command CREATE COMPOSITION FUNCTION |

| Methods inherited from class com.psychofree.statements.PDL_PMLStatement |
|---|
| getStatementType, interpretStatement |

| Methods inherited from class com.psychofree.statements.PBMSStatement |
|---|
| getStatementType, interpretStatement, parseStatement |

| Methods inherited from class java.lang.Object |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

# Constructors

## CreateCompositionFunction

```
public CreateCompositionFunction(java.sql.Connection conn)
```

Creates a new instance of CreateCompositionFunction

# Methods

## getFunctionName

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

**Returns:**
the name of the composition function

## getPatternTypeName

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

**Returns:**
the the pattern type

## getPatternName

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

(continued from last page)

**Returns:**

The pattern type name resulting from the composition

## getPTNameInput1

`public java.lang.String getPTNameInput1()`

**Returns:**

the first pattern type involved in the composition

## getPNameInput1

`public java.lang.String getPNameInput1()`

**Returns:**

the id of the first parameter used in the composition function

## getPTNameInput2

`public java.lang.String getPTNameInput2()`

**Returns:**

the second pattern type involved in the composition

## getPNameInput2

`public java.lang.String getPNameInput2()`

**Returns:**

the id of the second parameter used in the composition function

## getNewType

`public CreatePatternType getNewType()`

## isDefinedNew

`public boolean isDefinedNew()`

## getParamFields

`public java.util.ArrayList getParamFields()`

**Returns:**

(continued from last page)

the list of parameters used in the composition function

## getCode

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

**Returns:**

the code of the composition function

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

Parse the PDL command CREATE COMPOSITION FUNCTION

**Parameters:**

`st` - the StreamTokenizer containing the PDL command

**Throws:**

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

## interpretStatement

```
public java.lang.String interpretStatement()
  throws StatementExecutionException
```

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:**

`StatementExecutionException` - 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

---

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 | CreateCondition(java.sql.Connection conn)<br>Creates a new instance of CreateClass |

## Method Summary

| | |
|---|---|
| java.lang.String | getCode() |
| java.lang.String | getConditionName() |
| java.util.ArrayList | getParamFields() |
| java.lang.String | getPatternName() |
| java.lang.String | getPatternTypeName() |
| java.lang.String | interpretStatement()<br>A condition is translated in a stored function of PL/pgSQL that takes in input a pattern and returns an integer. |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the PDL command CREATE CONDITION |

| **Methods inherited from class com.psychofree.statements.PDL_PMLStatement** |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class com.psychofree.statements.PBMSStatement** |
|---|

| getStatementType, interpretStatement, parseStatement |
|---|

| Methods inherited from class java.lang.Object |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

# Constructors

## CreateCondition

```
public CreateCondition(java.sql.Connection conn)
```

Creates a new instance of CreateClass

# Methods

## getConditionName

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

## getPatternTypeName

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

## getPatternName

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

## getCode

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

## getParamFields

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

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

Parse the PDL command CREATE CONDITION

**Parameters:**

st - the StreamTokenizer containing the PDL command

(continued from last page)

**Throws:**

`ParseError` - if there is any syntax error in the PDL command

## interpretStatement

```
public java.lang.String interpretStatement()
  throws StatementExecutionException
```

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

---

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 | CreateFormula(java.sql.Connection conn) |

## Method Summary

| | |
|---|---|
| java.lang.String | getCode() |
| java.util.ArrayList | getParamFields() |
| java.lang.String | getPatternName() |
| java.lang.String | getPatternTypeName() |
| java.lang.String | interpretStatement()<br>Translates the formula definition in a PL/pgSQL function |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the formula definition |

| **Methods inherited from class com.psychofree.statements.PDL_PMLStatement** |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class com.psychofree.statements.PBMSStatement** |
|---|
| getStatementType, interpretStatement, parseStatement |

| Methods inherited from class java.lang.Object |
| --- |
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

# Constructors

## CreateFormula

```
public CreateFormula(java.sql.Connection conn)
```

# Methods

## getPatternTypeName

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

## getPatternName

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

## getParamFields

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

## getCode

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

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

Parse the formula definition

**Parameters:**

`st` - the StreamTokenizer containing the PDL command

**Throws:**

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

## interpretStatement

```
public java.lang.String interpretStatement()
  throws StatementExecutionException
```

(continued from last page)

Translates the formula definition in a PL/pgSQL function

**Returns:**

A message containing the outcome of the execution

**Throws:**

`StatementExecutionException` - 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

---

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 | CreateJoinCondition(java.sql.Connection conn)<br>Creates a new instance of CreateClass |

## Method Summary

| | |
|---|---|
| java.lang.String | getCode() |
| java.lang.String | getConditionName() |
| java.util.ArrayList | getParamFields() |
| java.lang.String | getPatternName1() |
| java.lang.String | getPatternName2() |
| java.lang.String | getPatternTypeName1() |
| java.lang.String | getPatternTypeName2() |
| java.lang.String | interpretStatement()<br>A join condition is translated in a stored function of PL/pgSQL that takes in input two patterns and returns a boolean |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the PDL command CREATE JOIN CONDITION |

| Methods inherited from class com.psychofree.statements.PDL_PMLStatement |
| --- |
| getStatementType, interpretStatement |

| Methods inherited from class com.psychofree.statements.PBMSStatement |
| --- |
| getStatementType, interpretStatement, parseStatement |

| Methods inherited from class java.lang.Object |
| --- |
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

# Constructors

## CreateJoinCondition

```
public CreateJoinCondition(java.sql.Connection conn)
```

Creates a new instance of CreateClass

# Methods

## getConditionName

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

## getPatternTypeName1

```
public java.lang.String getPatternTypeName1()
```

## getPatternName1

```
public java.lang.String getPatternName1()
```

## getCode

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

## getParamFields

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

## getPatternTypeName2

```
public java.lang.String getPatternTypeName2()
```

(continued from last page)

## getPatternName2

```
public java.lang.String getPatternName2()
```

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

Parse the PDL command CREATE JOIN CONDITION

**Parameters:**

`st` - the StreamTokenizer containing the PDL command

**Throws:**

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

## interpretStatement

```
public java.lang.String interpretStatement()
  throws StatementExecutionException
```

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:**

`StatementExecutionException` - if something goes wrong

# com.psychofree.statements
# Class CreateMeasureFunction

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

**All Implemented Interfaces:**
java.io.Serializable

---

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 | CreateMeasureFunction(java.sql.Connection conn)<br>Creates a new instance of CreateMiningFunction |

## Method Summary

| | |
|---|---|
| java.lang.String | getCode() |
| java.lang.String | getDataSourceName() |
| java.lang.String | getFunctionName() |
| java.util.ArrayList | getMeasureReturn() |
| java.util.ArrayList | getParamFields() |
| java.lang.String | getPatternName() |
| java.lang.String | getPatternTypeName() |
| java.lang.String | interpretStatement()<br>A measure function is translated in a stored function of PL/pgSQL with exaclty the behaviour defined in the body and that returns a measure object |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the PDL command CREATE MEASURE FUNCTION |

| Methods inherited from class com.psychofree.statements.PDL_PMLStatement |
| --- |
| getStatementType, interpretStatement |

| Methods inherited from class com.psychofree.statements.PBMSStatement |
| --- |
| getStatementType, interpretStatement, parseStatement |

| Methods inherited from class java.lang.Object |
| --- |
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

# Constructors

## CreateMeasureFunction

```
public CreateMeasureFunction(java.sql.Connection conn)
```

Creates a new instance of CreateMiningFunction

# Methods

## getFunctionName

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

## getPatternTypeName

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

## getPatternName

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

## getDataSourceName

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

## getParamFields

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

## getCode

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

(continued from last page)

## getMeasureReturn

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

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

Parse the PDL command CREATE MEASURE FUNCTION

**Parameters:**

`st` - the StreamTokenizer containing the PDL command

**Throws:**

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

## interpretStatement

```
public java.lang.String interpretStatement()
  throws StatementExecutionException
```

A measure function is translated in a stored function of PL/pgSQL with exaclty 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

---

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 | CreatePatternType(java.sql.Connection conn)<br>Creates a new instance of CreatePatternType |
| public | CreatePatternType(java.sql.Connection conn, java.lang.String patternName) |

## Method Summary

| | |
|---|---|
| java.lang.String | getEqualsCode() |
| java.util.List | getEqualsFields() |
| java.lang.String | getEqualsParameter() |
| java.lang.String | getFormulaCode() |
| java.util.List | getFormulaFields() |
| java.lang.String | getFormulaName() |
| java.lang.String | getFormulaParam() |
| java.util.List | getMeasureFields() |
| java.lang.String | getName() |

| | |
|---|---|
| java.lang.String | getReturnName()<br>Getter & Setter |
| java.util.List | getStructureFields() |
| java.lang.String | getThetaCode() |
| java.util.List | getThetaFields() |
| java.lang.String | getThetaParameter() |
| boolean | hasCode() |
| java.lang.String | interpretStatement()<br>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 | parseStatement(java.io.StreamTokenizer st)<br>Parse the PDL command CREATE PATTERN TYPE |
| void | setEqualsCode(java.lang.String EqualsCode) |
| void | setEqualsFields(java.util.List EqualsFields) |
| void | setEqualsParameter(java.lang.String EqualsParameter) |
| void | setFormulaCode(java.lang.String FormulaCode) |
| void | setFormulaFields(java.util.List FormulaFields) |
| void | setFormulaName(java.lang.String FormulaName) |
| void | setFormulaParam(java.lang.String FormulaParam) |
| void | setHasCode(boolean hasFormulaCode) |
| void | setMeasureFields(java.util.List MeasureFields) |
| void | setName(java.lang.String Name) |
| void | setReturnName(java.lang.String ReturnName) |
| void | setStructureFields(java.util.List StructureFields) |
| void | setThetaCode(java.lang.String ThetaCode) |
| void | setThetaFields(java.util.List ThetaFields) |

| | |
|---|---|
| void | setThetaParameter(java.lang.String ThetaParameter) |

**Methods inherited from class** com.psychofree.statements.PDL_PMLStatement

getStatementType, interpretStatement

**Methods inherited from class** com.psychofree.statements.PBMSStatement

getStatementType, interpretStatement, parseStatement

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

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

# Constructors

## CreatePatternType

```
public CreatePatternType(java.sql.Connection conn)
```

Creates a new instance of CreatePatternType

## CreatePatternType

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

# Methods

## getReturnName

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

Getter & Setter

## setReturnName

```
public void setReturnName(java.lang.String ReturnName)
```

## getName

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

## setName

```
public void setName(java.lang.String Name)
```

## getStructureFields

public java.util.List **getStructureFields**()

## setStructureFields

public void **setStructureFields**(java.util.List StructureFields)

## getMeasureFields

public java.util.List **getMeasureFields**()

## setMeasureFields

public void **setMeasureFields**(java.util.List MeasureFields)

## getFormulaFields

public java.util.List **getFormulaFields**()

## setFormulaFields

public void **setFormulaFields**(java.util.List FormulaFields)

## getEqualsFields

public java.util.List **getEqualsFields**()

## setEqualsFields

public void **setEqualsFields**(java.util.List EqualsFields)

## getEqualsCode

public java.lang.String **getEqualsCode**()

## setEqualsCode

public void **setEqualsCode**(java.lang.String EqualsCode)

(continued from last page)

## getThetaFields

```
public java.util.List getThetaFields()
```

## setThetaFields

```
public void setThetaFields(java.util.List ThetaFields)
```

## getThetaCode

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

## setThetaCode

```
public void setThetaCode(java.lang.String ThetaCode)
```

## getFormulaCode

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

## setFormulaCode

```
public void setFormulaCode(java.lang.String FormulaCode)
```

## setThetaParameter

```
public void setThetaParameter(java.lang.String ThetaParameter)
```

## setEqualsParameter

```
public void setEqualsParameter(java.lang.String EqualsParameter)
```

## getThetaParameter

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

(continued from last page)

## getEqualsParameter

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

## getFormulaName

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

## setFormulaName

```
public void setFormulaName(java.lang.String FormulaName)
```

## hasCode

```
public boolean hasCode()
```

## setHasCode

```
public void setHasCode(boolean hasFormulaCode)
```

## getFormulaParam

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

## setFormulaParam

```
public void setFormulaParam(java.lang.String FormulaParam)
```

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

Parse the PDL command CREATE PATTERN TYPE

**Parameters:**

`st` - the StreamTokenizer containing the PDL command

**Throws:**

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

(continued from last page)

## interpretStatement

```
public java.lang.String interpretStatement()
  throws StatementExecutionException
```

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:**

`StatementExecutionException` - 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

---

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 | DataCovering(java.sql.Connection conn)<br>Creates a new instance of DataCovering |

## Method Summary

| | |
|---|---|
| java.lang.String | getClassName() |
| java.lang.String | getCondition() |
| java.util.Vector | getCondParams() |
| java.lang.String | getDatasource() |
| Select | getInnerSelect() |
| java.lang.String | getStoreAs() |
| java.sql.ResultSet | interpretStatement()<br>Translates the PQL command in a call to a PL/pgSQL proedure that performs the query |
| boolean | isSelect() |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the PQL command DATA COVERING |

| **Methods inherited from class** com.psychofree.statements.PQLStatement |
|---|
| getStatementType, interpretStatement |

| Methods inherited from class com.psychofree.statements.PBMSStatement |
| --- |
| getStatementType, interpretStatement, parseStatement |

| Methods inherited from class java.lang.Object |
| --- |
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

# Constructors

## DataCovering

```
public DataCovering(java.sql.Connection conn)
```

Creates a new instance of DataCovering

# Methods

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

Parse the PQL command DATA COVERING

**Parameters:**
st - the StreamTokenizer containing the PQL command

**Throws:**
ParseError - if there is any syntax error in the command

## 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()
```

## interpretStatement

```
public java.sql.ResultSet interpretStatement()
  throws StatementExecutionException
```

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

**Returns:**

The result of the query

**Throws:**

`StatementExecutionException` - if something goes wrong

# com.psychofree.statements
# Class DeleteFromClass

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

**All Implemented Interfaces:**
java.io.Serializable

---

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 | DeleteFromClass(java.sql.Connection conn)<br>Creates a new instance of CreateClass |
| public | DeleteFromClass(java.lang.String ClassName, java.lang.String Condition) |

## Method Summary

| | |
|---|---|
| java.lang.String | getAlias() |
| java.lang.String | getClassName() |
| java.lang.String | getCondition() |
| java.lang.String | interpretStatement()<br>Translates the PML command in a call to the PL/pgSQL procedure ClassDelete that performs the deletion |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the PDL command DELETE FROM CLASS |

| **Methods inherited from class** com.psychofree.statements.PDL_PMLStatement |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class** com.psychofree.statements.PBMSStatement |
|---|
| getStatementType, interpretStatement, parseStatement |

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

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

# Constructors

## DeleteFromClass

```
public DeleteFromClass(java.sql.Connection conn)
```

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()
```

## getAlias

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

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws WrongKeywordException
```

Parse the PDL command DELETE FROM CLASS

**Parameters:**

`st` - the StreamTokenizer containing the PML command

**Throws:**

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

## interpretStatement

```
public java.lang.String interpretStatement()
  throws StatementExecutionException
```

(continued from last page)

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

**Returns:**

A message containing the outcome of the execution

**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

---

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 | DeletePatterns(java.sql.Connection conn)<br>Creates a new instance of NewClass |

## Method Summary

| | |
|---|---|
| java.lang.String | getAlias() |
| java.lang.String | getCondition() |
| java.lang.String | getPatternTypeName() |
| java.lang.String | interpretStatement()<br>Translates the PML command in a call to the PL/pgSQL function Deletion that performs the deletion |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the PDL command DELETE PATTERNS |

| **Methods inherited from class** com.psychofree.statements.PDL_PMLStatement |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class** com.psychofree.statements.PBMSStatement |
|---|
| getStatementType, interpretStatement, parseStatement |

| **Methods inherited from class** java.lang.Object |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

# Constructors

## DeletePatterns

```
public DeletePatterns(java.sql.Connection conn)
```

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(java.io.StreamTokenizer st)
  throws WrongKeywordException
```

Parse the PDL command DELETE PATTERNS

**Parameters:**

`st` - the StreamTokenizer containing the PML command

**Throws:**

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

## interpretStatement

```
public java.lang.String interpretStatement()
  throws StatementExecutionException
```

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

**Returns:**

A message containing the outcome of the execution

**Throws:**

`StatementExecutionException` - 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

---

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 | DirectInsertPattern(java.sql.Connection conn) |

## Method Summary

| | |
|---|---|
| java.lang.String | getDataSourceName() |
| java.lang.String | getIntoClass() |
| java.lang.String | getPatternTypeName() |
| java.lang.String | getStructureValue() |
| java.lang.String | getThresoldDefinition() |
| java.lang.String | getTsFinal() |
| java.lang.String | getTsInit() |
| java.lang.String | interpretStatement()<br>Translates the PML command in a call to the PL/pgSQL function Insertion that performs the insertion |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the PML command DIRECT INSERT PATTERN |

| **Methods inherited from class** com.psychofree.statements.PDL_PMLStatement |
|---|

| getStatementType, interpretStatement |
| --- |

| **Methods inherited from class** com.psychofree.statements.PBMSStatement |
| --- |
| getStatementType, interpretStatement, parseStatement |

| **Methods inherited from class** java.lang.Object |
| --- |
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

# Constructors

### DirectInsertPattern

```
public DirectInsertPattern(java.sql.Connection conn)
```

# Methods

### getPatternTypeName

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

### getDataSourceName

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

### 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(java.io.StreamTokenizer st)
  throws ParseError
```

Parse the PML command DIRECT INSERT PATTERN

**Parameters:**

`st` - the StreamTokenizer containing the PML command

**Throws:**

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

## interpretStatement

```
public java.lang.String interpretStatement()
  throws StatementExecutionException
```

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

**Returns:**

A message containing the outcome of the execution

**Throws:**

`StatementExecutionException` - if something goes wrong

# com.psychofree.statements
# Class DrillThrough

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

**All Implemented Interfaces:**
java.io.Serializable

---

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 | DrillThrough(java.sql.Connection conn)<br>Creates a new instance of DrillThrough |

## Method Summary

| | |
|---|---|
| java.lang.String | getClassName() |
| java.lang.String | getCondition() |
| java.util.Vector | getCondParams() |
| Select | getInnerSelect() |
| java.lang.String | getTableName() |
| java.sql.ResultSet | interpretStatement()<br>Translates the PQL command in a call to a PL/pgSQL procedure that performs the query |
| boolean | isSelect() |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the PQL command DRILL THROUGH |

| **Methods inherited from class com.psychofree.statements.PQLStatement** |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class com.psychofree.statements.PBMSStatement** |
|---|

| getStatementType, interpretStatement, parseStatement |
|---|

| **Methods inherited from class** java.lang.Object |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

# Constructors

## DrillThrough

```
public DrillThrough(java.sql.Connection conn)
```

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()
```

## isSelect

```
public boolean isSelect()
```

## getCondParams

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

(continued from last page)

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

Parse the PQL command DRILL THROUGH

**Throws:**

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

---

## interpretStatement

```
public java.sql.ResultSet interpretStatement()
  throws StatementExecutionException
```

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

**Returns:**

The result of the query

**Throws:**

`StatementExecutionException` - 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

---

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 | DropClass(java.sql.Connection conn)<br>Creates a new instance of DropClass |

## Method Summary

| | |
|---|---|
| java.lang.String | getName() |
| java.lang.String | interpretStatement()<br>Translates the PML statement in a call in a call to the PL/pgSQL stored function DeleteClass that performs the deletion |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the PDL command DROP CLASS |

| **Methods inherited from class** com.psychofree.statements.PDL_PMLStatement |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class** com.psychofree.statements.PBMSStatement |
|---|
| getStatementType, interpretStatement, parseStatement |

| **Methods inherited from class** java.lang.Object |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

---

## Constructors

(continued from last page)

## DropClass

```
public DropClass(java.sql.Connection conn)
```

Creates a new instance of DropClass

# Methods

## getName

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

---

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
```

Parse the PDL command DROP CLASS

**Throws:**

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

---

## interpretStatement

```
public java.lang.String interpretStatement()
  throws StatementExecutionException
```

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
   |-com.psychofree.statements.PBMSStatement
      |-com.psychofree.statements.PDL_PMLStatement
         |-com.psychofree.statements.DropMeasureFunction
```

**All Implemented Interfaces:**
java.io.Serializable

---

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 | DropMeasureFunction(java.sql.Connection conn)<br>Creates a new instance of DropMiningFunction |

## Method Summary

| | |
|---|---|
| java.lang.String | getName() |
| java.lang.String | interpretStatement()<br>Deletes the stored function which implements the measure function then updates the system catalogue PBMSMeasureFunctions deleting the relative entry |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the PML command DROP MEASURE FUNCTION |

| **Methods inherited from class** com.psychofree.statements.PDL_PMLStatement |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class** com.psychofree.statements.PBMSStatement |
|---|
| getStatementType, interpretStatement, parseStatement |

| **Methods inherited from class** java.lang.Object |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

---

## Constructors

(continued from last page)

## DropMeasureFunction

```
public DropMeasureFunction(java.sql.Connection conn)
```

Creates a new instance of DropMiningFunction

# Methods

## getName

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

---

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws WrongKeywordException
```

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 StatementExecutionException
```

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
   |-com.psychofree.statements.PBMSStatement
      |-com.psychofree.statements.PDL_PMLStatement
         |-com.psychofree.statements.DropPatternType
```

**All Implemented Interfaces:**
java.io.Serializable

---

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 | DropPatternType(java.sql.Connection conn)<br>Creates a new instance of CreatePatternType |

## Method Summary

| | |
|---|---|
| java.lang.String | getName()<br>Getter & Setter |
| java.lang.String | interpretStatement()<br>The deletion of a pattern type is translated in the deletion of the three base types defined and the typed classes created |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the PML command DROP PATTERN TYPE |

| **Methods inherited from class** com.psychofree.statements.PDL_PMLStatement |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class** com.psychofree.statements.PBMSStatement |
|---|
| getStatementType, interpretStatement, parseStatement |

| **Methods inherited from class** java.lang.Object |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

---

## Constructors

(continued from last page)

## DropPatternType

```
public DropPatternType(java.sql.Connection conn)
```

Creates a new instance of CreatePatternType

# Methods

## getName

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

Getter & Setter

---

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws WrongKeywordException
```

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 StatementExecutionException
```

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 EmptyBin

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

**All Implemented Interfaces:**
java.io.Serializable

---

public class **EmptyBin**
extends PDL_PMLStatement

This particular statement is to be used when user wants to purge his tablespace from bin tables It's not used in PSYCHOfree

---

| **Fields inherited from class** com.psychofree.statements.PBMSStatement |
|---|
| OTHER_STATEMENT, PDL_OR_PML_STATEMENT, QUERY_STATEMENT |

## Constructor Summary

| | |
|---|---|
| public | EmptyBin(java.sql.Connection conn, java.lang.String user) |

## Method Summary

| | |
|---|---|
| java.lang.String | interpretStatement() |
| void | parseStatement(java.io.StreamTokenizer st) |

| **Methods inherited from class** com.psychofree.statements.PDL_PMLStatement |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class** com.psychofree.statements.PBMSStatement |
|---|
| getStatementType, interpretStatement, parseStatement |

| **Methods inherited from class** java.lang.Object |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

---

## Constructors

### EmptyBin

```
public EmptyBin(java.sql.Connection conn,
                java.lang.String user)
```

(continued from last page)

# Methods

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

This method parse a PBMSStatement from input

---

## interpretStatement

```
public java.lang.String interpretStatement()
  throws StatementExecutionException
```

# com.psychofree.statements
# Class ExecSQL

```
java.lang.Object
   |-com.psychofree.statements.PBMSStatement
      |-com.psychofree.statements.ExecSQL
```

**All Implemented Interfaces:**
java.io.Serializable

---

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 | ExecSQL(java.sql.Connection conn)<br>Creates a new instance of ExecSQL |

## Method Summary

| | |
|---|---|
| java.lang.String | getSQLCode() |
| int | getStatementType() |
| java.lang.Object | interpretStatement()<br>Execute the SQL or PL/pgSQL statement |
| void | parseStatement(java.io.StreamTokenizer st) |

| **Methods inherited from class** com.psychofree.statements.PBMSStatement |
|---|
| getStatementType, interpretStatement, parseStatement |

| **Methods inherited from class** java.lang.Object |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

---

## Constructors

### ExecSQL

```
public ExecSQL(java.sql.Connection conn)
```

Creates a new instance of ExecSQL

# Methods

## getSQLCode

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

---

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

This method parse a PBMSStatement from input

---

## interpretStatement

```
public java.lang.Object interpretStatement()
  throws ExecSQLStatementException
```

Execute the SQL or PL/pgSQL statement

**Returns:**

An object containing the query result

**Throws:**

`ExecSQLStatementException` - if there was any error in the statement execution

---

## getStatementType

```
public int getStatementType()
```

# com.psychofree.statements
# Class ExecSQLStatementException

```
java.lang.Object
   ┗-java.lang.Throwable
      ┗-java.lang.Exception
         ┗-com.psychofree.statements.StatementExecutionException
            ┗-com.psychofree.statements.ExecSQLStatementException
```

**All Implemented Interfaces:**
java.io.Serializable

---

public class **ExecSQLStatementException**
extends StatementExecutionException

Exception throwable if an ExecSQL statements fails

---

## Constructor Summary

| | |
|---|---|
| public | ExecSQLStatementException() |
| public | ExecSQLStatementException(java.lang.String msg) |

**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

### ExecSQLStatementException

`public` **`ExecSQLStatementException`**`()`

---

### ExecSQLStatementException

`public` **`ExecSQLStatementException`**`(java.lang.String msg)`

# com.psychofree.statements
# Class ExtractPattern

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

**All Implemented Interfaces:**
java.io.Serializable

---

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 | ExtractPattern(java.sql.Connection conn)<br>Creates a new instance of ExtractPatternType |

## Method Summary

| | |
|---|---|
| java.lang.String | getAlias() |
| java.lang.String | getCondition() |
| java.lang.String | getDataSourceName() |
| java.lang.String | getExtParams() |
| java.lang.String | getIntoClass() |
| java.lang.String | getMeasureFunction() |
| java.util.ArrayList | getMeasureValues() |
| java.lang.String | getMiningFunctionName() |
| java.lang.String | getPatternTypeName() |
| java.lang.String | getThresoldDefinition() |
| java.lang.String | getTsFinal() |

| | |
|---|---|
| java.lang.String | getTsInit() |
| java.lang.String | interpretStatement()<br>Extract the patterns using the mining fuction specified, then process the result using the PL/pgSQL stored procedure ProcessExtractedPatterns |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the PML command EXTRACT PATTERNS |

| Methods inherited from class com.psychofree.statements.PDL_PMLStatement |
|---|
| getStatementType, interpretStatement |

| Methods inherited from class com.psychofree.statements.PBMSStatement |
|---|
| getStatementType, interpretStatement, parseStatement |

| Methods inherited from class java.lang.Object |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

# Constructors

## ExtractPattern

```
public ExtractPattern(java.sql.Connection conn)
```

Creates a new instance of ExtractPatternType

# Methods

## 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()
```

## getExtParams

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

## getAlias

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

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

Parse the PML command EXTRACT PATTERNS

**Throws:**

ParseError - if there is any syntax error in the command

## interpretStatement

```
public java.lang.String interpretStatement()
  throws StatementExecutionException
```

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

**Returns:**

A message containing the outcome of the execution

**Throws:**

`StatementExecutionException` - if something goes wrong

# com.psychofree.statements
# Class InsertIntoClass

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

**All Implemented Interfaces:**
java.io.Serializable

---

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 | InsertIntoClass(java.sql.Connection conn)<br>Creates a new instance of CreateClass |
| public | InsertIntoClass(java.lang.String ClassName, java.lang.String Condition) |

## Method Summary

| | |
|---|---|
| java.lang.String | getAlias() |
| java.lang.String | getClassName() |
| java.lang.String | getCondition() |
| java.lang.String | interpretStatement()<br>Translates the PML command into a call to the PL/pgSQL procedure ClassInsert which performs the insertion |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the PML command INSERT INTO CLASS |

| **Methods inherited from class** com.psychofree.statements.PDL_PMLStatement |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class** com.psychofree.statements.PBMSStatement |
|---|
| getStatementType, interpretStatement, parseStatement |

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

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

# Constructors

## InsertIntoClass

```
public InsertIntoClass(java.sql.Connection conn)
```

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()
```

## getAlias

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

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws WrongKeywordException
```

Parse the PML command INSERT INTO CLASS

**Throws:**

ParseError - if there is any syntax error in the command

## interpretStatement

```
public java.lang.String interpretStatement()
  throws StatementExecutionException
```

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

**Returns:**

(continued from last page)

A message containing the outcome of the execution

**Throws:**

`StatementExecutionException` - if something goes wrong

# com.psychofree.statements
# Class NoStatementRecognizedException

```
java.lang.Object
   |-java.lang.Throwable
      |-java.lang.Exception
         |-com.psychofree.statements.ParseError
            |-com.psychofree.statements.NoStatementRecognizedException
```

**All Implemented Interfaces:**
java.io.Serializable

---

public class **NoStatementRecognizedException**
extends ParseError

Exception to be risen when a statement isn't recognized

---

## Constructor Summary

| | |
|---|---|
| public | NoStatementRecognizedException(java.lang.String cmd) |

**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

### NoStatementRecognizedException

`public` **`NoStatementRecognizedException`**`(java.lang.String cmd)`

# com.psychofree.statements
# Class ParseError

```
java.lang.Object
   |
   +-java.lang.Throwable
      |
      +-java.lang.Exception
         |
         +-com.psychofree.statements.ParseError
```

**All Implemented Interfaces:**
java.io.Serializable

**Direct Known Subclasses:**
NoStatementRecognizedException, UnexpectedKeywordException, WrongKeywordException

---

public class **ParseError**
extends java.lang.Exception

General Exception throwable while parsing statements

---

## Constructor Summary

| | |
|---|---|
| public | ParseError(java.lang.String msg) |

**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

### ParseError

`public ParseError(java.lang.String msg)`

# 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 | ParserUtilities() |

## Method Summary

| | |
|---|---|
| static void | consumeToken(java.io.StreamTokenizer st, java.lang.String token)<br>Consumes a given token of the input stream. |
| static void | consumeToken(java.io.StreamTokenizer st, java.lang.String[] token)<br>Consumes tokens of the input stream |
| static java.lang.String | nextStringToken(java.io.StreamTokenizer st)<br>Retreive the next string token in the stream |
| static java.lang.String | nextToken(java.io.StreamTokenizer st)<br>Retreive the next token in the stream |
| static TypedField | parseFieldDeclaration(java.io.StreamTokenizer st, java.lang.String delim)<br>Parse a field declaration part of a command from the input stream starting with a given keyword |
| static TypedField | parseFieldDeclaration(java.io.StreamTokenizer st, java.lang.String[] delim)<br>Parse a field declaration part of a command from the input stream bounded by the given keywords. |
| static java.lang.String | parseWhere(java.io.StreamTokenizer st, java.lang.String[] delim, java.lang.String alias, java.lang.String patternTypeName)<br>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**()

(continued from last page)

# Methods

## nextStringToken

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

Retreive 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)
```

Retreive 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 UnexpectedKeywordException
```

Consumes a given token of the input stream.

**Parameters:**
`st` - The stream to parse.
`token` - The starting keyword

**Throws:**
`UnexpectedKeywordException` - 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)
```

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

---

(continued from last page)

## 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 UnexpectedKeywordException
```

Consumes tokens of the input stream

**Parameters:**

`st` - The stream to parse
`token` - Array containing the keywords to consume

**Throws:**

`UnexpectedKeywordException` - If the current token in the stream is differnt from the given keywords

# 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

---

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 | PatternCovering(java.sql.Connection conn) |

## Method Summary

| | |
|---|---|
| java.lang.String | getClassName() |
| java.lang.String | getCondition() |
| java.util.Vector | getCondParams() |
| java.lang.String | getDatasource() |
| Select | getInnerSelect() |
| java.lang.String | getStoreAs() |
| java.sql.ResultSet | interpretStatement()<br>Translates the PQL command in a call to a PL/pgSQL proedure that performs the query |
| boolean | isSelect() |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the PQL command PATTERN COVERING |

| **Methods inherited from class com.psychofree.statements.PQLStatement** |
|---|
| getStatementType, interpretStatement |

| Methods inherited from class com.psychofree.statements.PBMSStatement |
| --- |
| getStatementType, interpretStatement, parseStatement |

| Methods inherited from class java.lang.Object |
| --- |
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

# Constructors

## PatternCovering

```
public PatternCovering(java.sql.Connection conn)
```

# Methods

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

Parse the PQL command PATTERN COVERING

**Throws:**

ParseError - if there is any syntax error in the command

## interpretStatement

```
public java.sql.ResultSet interpretStatement()
  throws StatementExecutionException
```

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

**Returns:**

The result of the query

**Throws:**

StatementExecutionException - if something goes wrong

## getClassName

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

## getCondition

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

(continued from last page)

## 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

**Direct Known Subclasses:**
ExecSQL, PDL_PMLStatement, PQLStatement, WrongStatement

---

public abstract class **PBMSStatement**
extends java.lang.Object
implements java.io.Serializable

The superclass of all PSYCHOfree statements

---

## Field Summary

| | |
|---|---|
| public static | OTHER_STATEMENT |
| public static | PDL_OR_PML_STATEMENT |
| public static | QUERY_STATEMENT |

## Constructor Summary

| | |
|---|---|
| public | PBMSStatement() |
| public | PBMSStatement(java.sql.Connection conn) |

## Method Summary

| | |
|---|---|
| abstract int | getStatementType() |
| abstract java.lang.Object | interpretStatement() |
| abstract void | parseStatement(java.io.StreamTokenizer st)<br>This method parse a PBMSStatement from input |

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

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

---

## Fields

(continued from last page)

### 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)
```

## Methods

### parseStatement

```
public abstract void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

This method parse a PBMSStatement from input

**Parameters:**

`st` - The StreamTokenizer Object used to get tokens from input stream

**Throws:**

`ParseError` - If this statement's syntax is not correct

### interpretStatement

```
public abstract java.lang.Object interpretStatement()
  throws StatementExecutionException
```

### getStatementType

```
public abstract int getStatementType()
```

(continued from last page)

# com.psychofree.statements
# Class PDL_PMLStatement

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

**All Implemented Interfaces:**
java.io.Serializable

**Direct Known Subclasses:**
CreateClass, DeleteFromClass, DeletePatterns, DirectInsertPattern, DropClass, DropMeasureFunction, DropPatternType, EmptyBin, ExtractPattern, InsertIntoClass, RecomputePattern, StatementWithCode, Update, UpdatePattern

---

public abstract class **PDL_PMLStatement**
extends PBMSStatement

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 | PDL_PMLStatement() |
| public | PDL_PMLStatement(java.sql.Connection conn) |

## Method Summary

| | |
|---|---|
| int | getStatementType() |
| abstract java.lang.String | interpretStatement() |

| **Methods inherited from class** com.psychofree.statements.PBMSStatement |
|---|
| getStatementType, interpretStatement, parseStatement |

| **Methods inherited from class** java.lang.Object |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

## Constructors

### PDL_PMLStatement

```
public PDL_PMLStatement()
```

(continued from last page)

### PDL_PMLStatement

```
public PDL_PMLStatement(java.sql.Connection conn)
```

## Methods

### interpretStatement

```
public abstract java.lang.String interpretStatement()
  throws StatementExecutionException
```

### getStatementType

```
public int getStatementType()
```

# com.psychofree.statements
# Class PQLStatement

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

**All Implemented Interfaces:**
java.io.Serializable

**Direct Known Subclasses:**
DataCovering, DrillThrough, PatternCovering, Select, Show

---

public abstract class **PQLStatement**
extends PBMSStatement

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 | PQLStatement() |
| public | PQLStatement(java.sql.Connection conn) |

## Method Summary

| | |
|---|---|
| int | getStatementType() |
| abstract java.sql.ResultSet | interpretStatement() |

| Methods inherited from class com.psychofree.statements.PBMSStatement |
|---|
| getStatementType, interpretStatement, parseStatement |

| Methods inherited from class java.lang.Object |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

---

## Constructors

### PQLStatement

```
public PQLStatement()
```

### PQLStatement

```
public PQLStatement(java.sql.Connection conn)
```

## Methods

### getStatementType

```
public int getStatementType()
```

### interpretStatement

```
public abstract java.sql.ResultSet interpretStatement()
  throws StatementExecutionException
```

# com.psychofree.statements
# Class RecomputePattern

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

**All Implemented Interfaces:**
java.io.Serializable

---

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 | RecomputePattern(java.sql.Connection conn)<br>Creates a new instance of ExtractPatternType |

## Method Summary

| | |
|---|---|
| java.lang.String | getAlias() |
| java.lang.String | getCondition() |
| java.lang.String | getDataSourceName() |
| java.lang.String | getIntoClass() |
| java.lang.String | getMeasureFunctionName() |
| java.lang.String | getPatternTypeName() |
| java.lang.String | getTsFinal() |
| java.lang.String | getTsInit() |
| java.lang.String | interpretStatement()<br>Translates the command in a call to the PL/pgSQL stored procedure Recomputation that performs the operation |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the PML command RECOMPUTE PATTERNS |

| Methods inherited from class com.psychofree.statements.PDL_PMLStatement |
| --- |
| getStatementType, interpretStatement |

| Methods inherited from class com.psychofree.statements.PBMSStatement |
| --- |
| getStatementType, interpretStatement, parseStatement |

| Methods inherited from class java.lang.Object |
| --- |
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

# Constructors

## RecomputePattern

```
public RecomputePattern(java.sql.Connection conn)
```

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()
```

(continued from last page)

## getMeasureFunctionName

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

## getIntoClass

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

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

Parse the PML command RECOMPUTE PATTERNS

**Throws:**

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

## interpretStatement

```
public java.lang.String interpretStatement()
  throws StatementExecutionException
```

Translates the command in a call to the PL/pgSQL stored procedure Recomputation that performs the operation

**Returns:**

A message containing the outcome of the execution

**Throws:**

`StatementExecutionException` - 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

---

public class **Select**
extends PQLStatement

This class wraps a SELECT statement of PQL

---

## Field Summary

| | |
|---|---|
| public static final | CJOIN<br>Value: **3** |
| public static final | INTERSECT<br>Value: **2** |
| public static final | NATURAL<br>Value: **0** |
| public static final | UNION<br>Value: **1** |

**Fields inherited from class com.psychofree.statements.PBMSStatement**

OTHER_STATEMENT, PDL_OR_PML_STATEMENT, QUERY_STATEMENT

## Constructor Summary

| | |
|---|---|
| public | Select(java.sql.Connection conn)<br>Creates a new instance of Select |

## Method Summary

| | |
|---|---|
| java.util.ArrayList | getClassLists() |
| java.lang.String | getClassName() |
| java.lang.String | getCompFuntionName() |
| java.lang.String | getCondition() |

| | |
|---|---|
| java.util.ArrayList | getFields() |
| java.lang.String | getFirstAlias() |
| java.lang.String | getJoinedAlias() |
| java.lang.String | getJoinedClass() |
| int | getJoinType() |
| boolean | getOpType() |
| java.lang.String | getTableName() |
| java.lang.String | getTypeName() |
| java.sql.ResultSet | interpretStatement()<br>Translates the Selection in a call to a PL/pgSQL procedure that performs the query |
| boolean | isJoin() |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the PQL command SELECT |

| **Methods inherited from class com.psychofree.statements.PQLStatement** |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class com.psychofree.statements.PBMSStatement** |
|---|
| getStatementType, interpretStatement, parseStatement |

| **Methods inherited from class java.lang.Object** |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

# 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)
```

Creates a new instance of Select

## Methods

### getFields

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

(continued from last page)

## isJoin

```
public boolean isJoin()
```

## getTypeName

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

## getJoinType

```
public int getJoinType()
```

## getCompFuntionName

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

## getOpType

```
public boolean getOpType()
```

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

Parse the PQL command SELECT

**Throws:**

ParseError - if there is any syntax error in the command

## getJoinedAlias

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

## getFirstAlias

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

(continued from last page)

## interpretStatement

```
public java.sql.ResultSet interpretStatement()
  throws StatementExecutionException
```

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

**Returns:**

The result of the query

**Throws:**

`StatementExecutionException` - 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

---

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 | Show(java.sql.Connection conn)<br>Creates a new instance of Show |

## Method Summary

| | |
|---|---|
| java.sql.ResultSet | interpretStatement()<br>Translates the SHOW statement into the correspondent SQL query |
| boolean | isIsClasses() |
| boolean | isIsPatternTypes() |
| void | parseStatement(java.io.StreamTokenizer st)<br>Parse the SHOW command |

| **Methods inherited from class** com.psychofree.statements.PQLStatement |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class** com.psychofree.statements.PBMSStatement |
|---|
| getStatementType, interpretStatement, parseStatement |

| **Methods inherited from class** java.lang.Object |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

---

(continued from last page)

# Constructors

## Show

```
public Show(java.sql.Connection conn)
```

Creates a new instance of Show

# Methods

## isIsClasses

```
public boolean isIsClasses()
```

---

## isIsPatternTypes

```
public boolean isIsPatternTypes()
```

---

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

Parse the SHOW command

**Throws:**

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

---

## interpretStatement

```
public java.sql.ResultSet interpretStatement()
  throws StatementExecutionException
```

Translates the SHOW statement into the correspondent SQL query

**Returns:**

The resultset cointaining the query result

**Throws:**

`StatementExecutionException` - if something goes wrong

# com.psychofree.statements
# Class StatementExecutionException

```
java.lang.Object
   |-java.lang.Throwable
      |-java.lang.Exception
         |-com.psychofree.statements.StatementExecutionException
```

**All Implemented Interfaces:**
java.io.Serializable

**Direct Known Subclasses:**
ExecSQLStatementException

---

public class **StatementExecutionException**
extends java.lang.Exception

General Exception throwable while interpreting a statement

---

## Constructor Summary

| | |
|---|---|
| public | StatementExecutionException() |
| public | StatementExecutionException(java.lang.String msg) |

**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

### StatementExecutionException

`public StatementExecutionException()`

---

### StatementExecutionException

`public StatementExecutionException(java.lang.String msg)`

# com.psychofree.statements
# Class StatementWithCode

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

**All Implemented Interfaces:**
java.io.Serializable

**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) |

| **Methods inherited from class** com.psychofree.statements.PDL_PMLStatement |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class** com.psychofree.statements.PBMSStatement |
|---|
| getStatementType, interpretStatement, parseStatement |

| **Methods inherited from class** java.lang.Object |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

---

## Constructors

### StatementWithCode

```
public StatementWithCode(java.sql.Connection conn)
```

# com.psychofree.statements
# Class UnexpectedKeywordException

```
java.lang.Object
   |
   +-java.lang.Throwable
      |
      +-java.lang.Exception
         |
         +-com.psychofree.statements.ParseError
            |
            +-com.psychofree.statements.UnexpectedKeywordException
```

**All Implemented Interfaces:**
java.io.Serializable

---

public class **UnexpectedKeywordException**
extends ParseError

Exception to be risen when, while parsing a statement, an incorrect keyword is found

---

## Constructor Summary

| | |
|---|---|
| public | UnexpectedKeywordException(java.lang.String token) |

**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

### UnexpectedKeywordException

`public` **`UnexpectedKeywordException`**`(java.lang.String token)`

# 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

---

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 | Update(java.sql.Connection conn) |

## Method Summary

| | |
|---|---|
| java.lang.String | getAlias() |
| java.lang.String | getPatternTypeName() |
| java.lang.String | interpretStatement()<br>Translates the PML command in a call to a PL/pgSQL procedure that performs the update |
| void | parseStatement(java.io.StreamTokenizer st) |

| **Methods inherited from class** com.psychofree.statements.PDL_PMLStatement |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class** com.psychofree.statements.PBMSStatement |
|---|
| getStatementType, interpretStatement, parseStatement |

| **Methods inherited from class** java.lang.Object |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

---

## Constructors

(continued from last page)

## Update

```
public Update(java.sql.Connection conn)
```

# Methods

## getPatternTypeName

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

---

## getAlias

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

---

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

This method parse a PBMSStatement from input

---

## interpretStatement

```
public java.lang.String interpretStatement()
  throws StatementExecutionException
```

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

**Returns:**

A message containing the outcome of the execution

**Throws:**

`StatementExecutionException` - 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

**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 instanciate 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 | UpdatePattern(java.sql.Connection conn, java.lang.String patternTypeName, java.lang.String alias) |

## Method Summary

| | |
|---|---|
| java.lang.String | getAlias() |
| java.lang.String | getCondition() |
| java.lang.String | getPatternTypeName() |

| **Methods inherited from class** com.psychofree.statements.PDL_PMLStatement |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class** com.psychofree.statements.PBMSStatement |
|---|
| getStatementType, interpretStatement, parseStatement |

| **Methods inherited from class** java.lang.Object |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

---

# 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
   ┗-com.psychofree.statements.PBMSStatement
      ┗-com.psychofree.statements.PDL_PMLStatement
         ┗-com.psychofree.statements.UpdatePattern
            ┗-com.psychofree.statements.UpdatePatternWithMeasureFunction
```

**All Implemented Interfaces:**
java.io.Serializable

**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 | UpdatePatternWithMeasureFunction(java.sql.Connection conn, java.lang.String patternTypeName, java.lang.String alias) |

## Method Summary

| | |
|---|---|
| java.lang.String | getFunctionName() |

| **Methods inherited from class** com.psychofree.statements.UpdatePattern |
|---|
| getAlias, getCondition, getPatternTypeName |

| **Methods inherited from class** com.psychofree.statements.PDL_PMLStatement |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class** com.psychofree.statements.PBMSStatement |
|---|
| getStatementType, interpretStatement, parseStatement |

| **Methods inherited from class** java.lang.Object |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

---

# 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
   |-com.psychofree.statements.PBMSStatement
      |-com.psychofree.statements.PDL_PMLStatement
         |-com.psychofree.statements.UpdatePattern
            |-com.psychofree.statements.UpdatePatternWithMeasureFunction
               |-com.psychofree.statements.UpdateSynchronize
```

**All Implemented Interfaces:**
java.io.Serializable

---

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 | UpdateSynchronize(java.sql.Connection conn, java.lang.String patternTypeName, java.lang.String alias) |

## Method Summary

| | |
|---|---|
| java.lang.String | interpretStatement()<br>Translates the PML statement in a call to the PL/pgSQL procedure Syncronize that performs patterns synchronization recomputing the measures with the specified measure function |
| void | parseStatement(java.io.StreamTokenizer st) |

| **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** |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class com.psychofree.statements.PBMSStatement** |
|---|
| getStatementType, interpretStatement, parseStatement |

| Methods inherited from class java.lang.Object |
| --- |
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

# Constructors

## UpdateSynchronize

```
public UpdateSynchronize(java.sql.Connection conn,
                         java.lang.String patternTypeName,
                         java.lang.String alias)
```

# Methods

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

This method parse a PBMSStatement from input

## interpretStatement

```
public java.lang.String interpretStatement()
  throws StatementExecutionException
```

Translates the PML statement in a call to the PL/pgSQL procedure Syncronize that performs patterns synchronization recomputing the measures with the specified measure function

**Returns:**

A message containing the outcome of the execution

**Throws:**

StatementExecutionException - if something goes wrong

# com.psychofree.statements
# Class UpdateValidate

```
java.lang.Object
   |-com.psychofree.statements.PBMSStatement
      |-com.psychofree.statements.PDL_PMLStatement
         |-com.psychofree.statements.UpdatePattern
            |-com.psychofree.statements.UpdatePatternWithMeasureFunction
               |-com.psychofree.statements.UpdateValidate
```

**All Implemented Interfaces:**
java.io.Serializable

---

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 | UpdateValidate(java.sql.Connection conn, java.lang.String patternTypeName, java.lang.String alias) |

## Method Summary

| | |
|---|---|
| java.lang.String | getIntoClass() |
| java.lang.String | interpretStatement()<br>Translates the PML statement in a call to the PL/pgSQL procedure Validate that performs patterns validation |
| void | parseStatement(java.io.StreamTokenizer st) |

| **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** |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class com.psychofree.statements.PBMSStatement** |
|---|

| getStatementType, interpretStatement, parseStatement |
|---|

| **Methods inherited from class** java.lang.Object |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

# Constructors

## UpdateValidate

```
public UpdateValidate(java.sql.Connection conn,
                      java.lang.String patternTypeName,
                      java.lang.String alias)
```

# Methods

## getIntoClass

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

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

This method parse a PBMSStatement from input

## interpretStatement

```
public java.lang.String interpretStatement()
  throws StatementExecutionException
```

Translates the PML statement in a call to the PL/pgSQL procedure Validate that performs patterns validation

**Returns:**

A message containing the outcome of the execution

**Throws:**

StatementExecutionException - 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

---

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 | UpdateValidity(java.sql.Connection conn, java.lang.String patternTypeName, java.lang.String alias) |

## Method Summary

| | |
|---|---|
| java.lang.String | getTsFinal() |
| java.lang.String | getTsInit() |
| java.lang.String | interpretStatement()<br>Translates the PML statement in a SQL update statement that performs update of the validity period |
| void | parseStatement(java.io.StreamTokenizer st) |

| **Methods inherited from class com.psychofree.statements.UpdatePattern** |
|---|
| getAlias, getCondition, getPatternTypeName |

| **Methods inherited from class com.psychofree.statements.PDL_PMLStatement** |
|---|
| getStatementType, interpretStatement |

| **Methods inherited from class com.psychofree.statements.PBMSStatement** |
|---|
| getStatementType, interpretStatement, parseStatement |

| Methods inherited from class `java.lang.Object` |
| --- |
| `equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait` |

# Constructors

## UpdateValidity

```
public UpdateValidity(java.sql.Connection conn,
                      java.lang.String patternTypeName,
                      java.lang.String alias)
```

# Methods

## getTsInit

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

## getTsFinal

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

## parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

This method parse a PBMSStatement from input

## interpretStatement

```
public java.lang.String interpretStatement()
  throws StatementExecutionException
```

Translates the PML statement in a SQL update statement that performs update of the validity period

**Returns:**
A message containing the outcome of the execution

**Throws:**
`StatementExecutionException` - if something goes wrong

# com.psychofree.statements
# Class WrongKeywordException

```
java.lang.Object
   |
   +-java.lang.Throwable
      |
      +-java.lang.Exception
         |
         +-com.psychofree.statements.ParseError
            |
            +-com.psychofree.statements.WrongKeywordException
```

**All Implemented Interfaces:**
java.io.Serializable

---

public class **WrongKeywordException**
extends ParseError

Exception to be risen when, while parsing a statement, a wrong keyword is found

---

## Constructor Summary

| | |
|---|---|
| public | WrongKeywordException(java.lang.String token)<br>Creates a new instance of WrongKeywordException |

| **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

### WrongKeywordException

```
public WrongKeywordException(java.lang.String token)
```

Creates a new instance of WrongKeywordException

# com.psychofree.statements
# Class WrongStatement

```
java.lang.Object
   |-com.psychofree.statements.PBMSStatement
      |-com.psychofree.statements.WrongStatement
```

**All Implemented Interfaces:**
java.io.Serializable

---

public class **WrongStatement**
extends PBMSStatement

Wraps a statement that, if parsed, will produce a ParseError

---

| **Fields inherited from class** com.psychofree.statements.PBMSStatement |
|---|
| OTHER_STATEMENT, PDL_OR_PML_STATEMENT, QUERY_STATEMENT |

## Constructor Summary

| | |
|---|---|
| public | WrongStatement(java.lang.String stat) |
| public | WrongStatement(java.lang.String stat, java.lang.String cause) |

## Method Summary

| | |
|---|---|
| java.lang.String | getCause() |
| java.lang.String | getStatement() |
| int | getStatementType() |
| java.lang.Object | interpretStatement() |
| void | parseStatement(java.io.StreamTokenizer st)<br>This statement should never parse anything at all!!! |

| **Methods inherited from class** com.psychofree.statements.PBMSStatement |
|---|
| getStatementType, interpretStatement, parseStatement |

| **Methods inherited from class** java.lang.Object |
|---|
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

---

## Constructors

(continued from last page)

### WrongStatement

```
public WrongStatement(java.lang.String stat)
```

---

### WrongStatement

```
public WrongStatement(java.lang.String stat,
                      java.lang.String cause)
```

## Methods

### getStatement

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

---

### getCause

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

---

### parseStatement

```
public void parseStatement(java.io.StreamTokenizer st)
  throws ParseError
```

This statement should never parse anything at all!!!

---

### interpretStatement

```
public java.lang.Object interpretStatement()
  throws StatementExecutionException
```

---

### getStatementType

```
public int getStatementType()
```