

Analysis Process Modeling Notation For Business Intelligence  
  
Technische Dokumentation

Eingereicht von

**Alexander Amstler**

**Philip Hohendanner**

**Maria Limberger**

Angefertigt am

**Institut für Wirtschaftsinformatik –**

**Data & Knowledge Engineering**

Beurteiler / Beurteilerin

**Dipl.-Ing. Thomas Neuböck**

***Univ.-Prof. Dipl.-Ing. Dr. techn. Michael Schrefl***

Projektstudie

[**Problem**](#_gz1m39r798k1) **3**

[Einleitung](#_82wctxspplpi) 3

[Problemnachweis](#_42wlvy54lxn1) 3

[Ziel](#_bweehurrefyv) 3

[Forschungsfragen](#_1pcjqmygbwpo) 4

[Fragestellungen](#_59da1zhsh3f7) 4

[Phase 1](#_ogk3wdupzi0i) 4

[Beschreibung des Unternehmens](#_nc9r1tu827e) 4

[Analyse der Marktposition](#_zemc1wkrm3ko) 5

[Einordnung ins Business Model Canvas](#_atnrmv3d652i) 5

[**Problemlösungsweg**](#_21lijhzf58tj) **9**

[Vorgehensmodell](#_qyz4m53mvtdv) 9

[**Ergebnis**](#_fqzgxkaxw2yx) **9**

[**Literaturverzeichnis**](#_e8uhld3q1fhq) **9**

[**Abbildungsverzeichnis**](#_oderkp9rv0mq) **9**

# DBContext

# Insert\_item

Die Klasse Insert\_item ermöglicht es einen Spaltennamen als String, und einen dazugehörigen Wert entweder als String oder Integer abzuspeichern.

Der Spaltenname wird unter column\_name gespeichert. Durch die Eingabe eines Strings oder Integers in den Konstruktor kann der jeweilige Wert entweder unter s\_value oder unter i\_value gespeichert werden. Wird ein String unter s\_value gespeichter, so bleibt in i\_value der Wert -404 gespeichter.

Durch die Methode intValue wird über den i\_value ermittelt, ob ein Integer oder einen String Wert gespeichert wurde. Wenn der Wert -404 in i\_value gefunden wird ist es ein String, sonst ein Integer.

Mittels isNull wird festgestellt ob das Insert\_item initialisiert wurde. Ist s\_value null und i\_value -404 so ist es leer und gibt true zurück.

Durch getColumnName wird der Spaltenname zurückgegeben.

Durch getSValue wird der String Wert retourniert.

Durch getIValue wird der Integer Wert retourniert.

# DBService

Die Klasse DBService speichert sich alle wichtigen Logindaten, um auf die Postgres Datenbank zugreifen zu können:

* DBHostname speichert den Servernamen
* DBPort speichert die Portnummer
* DBUsername speichert den Usernamen
* DBPasswort speichert das Passwort des Usernamen
* DBName speichert den Datenbanknamen
* NewTransatction gibt an ob eine neue Transaktion gestartet wird oder ob es noch Teil einer alten Transaktion ist
* nass\_dq\_sid speichert den Primary Key der Tabelle AGS\_NASS\_DIM\_QUAL zwischen
* nass\_sc\_sid speichert den Primary Key der Tabelle AGS\_NASS\_DIM\_QUAL\_SLICE\_COND zwischen
* ds speichert das aktuelle Datenset zwischen
* dt speichert die aktuelle Tabelle zwischen

Mittels dem Login werden sämtliche Daten für den Login für die Postgres Datenbank in den oben beschriebenen Attributen gespeichert.

Die folgenden Methoden sind speziell für die ComboItem Klasse gedacht:

GetData erstellt mittels Npgsql eine Verbindung zur Postgres Datenbank und führt eine Abfrage, die als Input Parameter übergeben wird, aus und retourniert eine Tabelle (DataTable). Es gibt drei Überladungen der Methode, wobei diese

* zwei Spalten einer Tabelle zurückliefert
* eine Spalte einer Tabelle zurückliefert
* als Input Parameter ein SQL Statement als String akzeptiert und eine Spalte einer Tabelle zurückgibt.

getStringFormStmt

Die folgenden Methoden sind Hilfsmethoden:

getLatestID liefert den Integer Wert des größten Primary Key oder 1, wenn die Tabelle noch leer ist.

getKeyfromTable liefert den Integer Wert einer gewählten Spalte, wo der mitgegebene Integer Schlüssel, dem Wert der mitgegebenen Spalte entspricht oder retourniert einen Error mit -1 falls nichts gefunden wird.

getSKeyfromTable liefert den String Wert einer gewählten Spalte, wo der mitgegebene Integer Schlüssel, dem Wert der mitgegebenen Spalte entspricht oder retourniert einen Error mit -1 falls nichts gefunden wird.

getConnection liefert eine geöffnete NpgsqlConnection zum Postgres Server.

Die folgenden Methoden sind für das Einfügen in Tabellen:

insinto holt sich den aktuellen Primary Key durch getLatestID und erstellt dann über StringBuilder ein SQL Insert-Statement. Danach weißt es der übergebenen NpgsqlConnection und NpgsqlTrancation ein NpgsqlCommand zu, das aus dem SQL Insert-Statement und den mitgegebenen Insert\_items besteht, zu.

Es gibt auch eine Überladung der Methode, wo man den Primary Key auch gleich mitgeben kann und nicht erst aus einer Tabelle auslesen muss.

insertWithoutPK funktioniert gleich wie insinto, außer dass hier der Primary Key das erste Listenelement der Insert\_item Liste ist.

insertDimQual ist für das Einfügen der Dimension Qualifications zuständig. Wenn es eine neue Transaktion ist werden einmalig die Primary Key Attribute mit den aktuellen Werten aus den Postgres Tabellen befüllt, ansonsten werden die bereits gespeicherten Primary Keys um 1 erhöht.

Danach werden die Werte der jeweiligen ComboBox und ListBox entnommen und in eine Insert\_item Liste gespeichert. Je nach ausgewählten Wert in der jeweiligen ComboBox bzw. ListBox wird dann insinto oder insertWithoutPk aufgerufen.

transactionComplete setzt den boolean newTransaction wieder auf true, wenn die Transaktion beendet wurde.

# ComboItem

# Login

# BaseMenu

# LoadForm

# UserInput

# SaveSchema

# Abbildungsverzeichnis