

IBBSOFT®

Datentransfer mit

Data Bridge

Version 3.x

Dieses Handbuch dient als Arbeitsunterlage für Benutzer des Programms Data Bridge.

Die in diesem Handbuch enthaltenen Informationen können ohne Vorankündigung geändert werden und stellen keine Verpflichtung seitens IBBSOFT® dar. Die Software, die in diesem Handbuch beschrieben ist, wird im Rahmen einer Lizenzvereinbarung zur Verfügung gestellt.

Das Handbuch ist ausschließlich für unsere Kunden bestimmt. Alle Angaben erfolgen ohne Gewähr. Ohne ausdrückliche schriftliche Erlaubnis des Unternehmens IBBSOFT® darf dieses Handbuch oder ein Auszug hieraus für irgendwelche Zwecke oder in irgendeiner Form mit irgendwelchen Mitteln, elektronisch oder mechanisch, mittels Fotokopie, durch Auszeichnungen oder mit Informationsspeicherungs- bzw. Informationswiedergewinnungssystemen nicht reproduziert oder übertragen werden.

© Copyright 2007 IBBSOFT®

Herausgegeben von:

IBBSOFT®

IBB Industrieautomatisierung / Ingenieurbüro Bräuer

Liebenastr. 27
71111 Waldenbuch

Gewerbestraße 19
Technologiepark
08115 Lichtentanne / OT Stenn

Telefon: +49 (0) 7157 22285
Fax: +49 (0) 7157 22286
Email: info@ibbsoft.de

Stand der Dokumentation: März 2010

Inhalt

| | | |
|-------|--|----|
| 1 | Einführung | 5 |
| 2 | Installation..... | 6 |
| 3 | Data Bridge..... | 7 |
| 4 | Datentransfer | 8 |
| 4.1 | Erstellen der Datenbrücke mit Hilfe des Wizard..... | 8 |
| 4.2 | Speichern und Öffnen von Pipelines | 11 |
| 4.3 | Ausführen der Pipeline über die DOS-Eingabeaufforderung..... | 11 |
| 4.4 | Weitere Funktionalitäten des Wizard | 12 |
| 4.4.1 | Automatisches Verknüpfen gleichnamiger Felder | 12 |
| 4.4.2 | Voreinstellung der Datenbanken..... | 12 |
| 4.4.3 | UI Automation | 13 |
| 4.4.4 | UPDATE-Funktion | 14 |
| 4.4.5 | Kopieren einer Pipeline | 16 |
| 4.5 | WHERE-Bedingung..... | 17 |
| 4.6 | ORDER BY-Bedingung | 18 |
| 4.7 | Delete | 19 |
| 4.8 | Select..... | 19 |
| 4.9 | Log-Mitschnitt | 20 |
| 5 | Datenmanipulation | 20 |
| 5.1 | Generate Index | 20 |
| 5.1.1 | Fortlaufende Nummer generieren | 21 |
| 5.1.2 | Fortlaufende Nummerierung über mehrere Pipelines..... | 22 |
| 5.1.3 | Konstanten generieren..... | 23 |
| 5.2 | Umrechnungen und Berechnungen..... | 24 |
| 5.2.1 | Umrechnungen | 24 |
| 5.2.2 | Berechnungen | 25 |
| 5.3 | Formatierung | 29 |
| 5.4 | Kombinationen mit Konstanten..... | 29 |
| 6 | Automatisierung | 29 |
| 6.1 | Cron | 29 |
| 6.2 | Ausführen der Pipeline über Serverdienst..... | 32 |

Abbildungsverzeichnis

| | |
|---|----|
| Abbildung 1: Zusammenführung heterogen gehaltener Daten | 5 |
| Abbildung 2: Übersicht Datentransfer und Datenmanipulation | 6 |
| Abbildung 3: Hauptansicht des Programms Data Bridge..... | 7 |
| Abbildung 4: Datenbrücke zwischen Fahrzeug und PDB_leer | 8 |
| Abbildung 5: Ansicht Wizard..... | 9 |
| Abbildung 6: Wizard-Fenster mit ausgewählter Quelltablelle | 10 |
| Abbildung 7: Die Felder der Zieltabelle | 10 |
| Abbildung 8: Wizard Version | 12 |
| Abbildung 9: Makro-Datei eintragen | 13 |
| Abbildung 10: Wizard mit MAKRO Aktivierung | 14 |
| Abbildung 11: Feld ReNr ist Kriterium für Update | 15 |
| Abbildung 12: Darstellung der Update-Funktion in einer Pipeline..... | 16 |
| Abbildung 13: Where-Bedingung in der Pipeline..... | 17 |
| Abbildung 14: Order by-Bedingung in der Pipeline | 18 |
| Abbildung 15: Notation und Bedeutung für cron..... | 30 |
| Abbildung 16: Zeitangaben | 30 |
| Abbildung 17: Automatische und zeitgesteuerte Ausführung mittels cron | 31 |
| Abbildung 18: Dialog Cron Editor | 31 |
| Abbildung 19: Toolleiste signalisiert cron-Aktivierung | 32 |
| Abbildung 20: Funktion Autonumber | 21 |
| Abbildung 21: Zuordnung mit Autonumber..... | 22 |
| Abbildung 22: Notationen Autonumber | 22 |
| Abbildung 23: Beispiel von Konstanten | 23 |
| Abbildung 24: Resultat der Datenübertragung in Tabelle Dat_tTM mit Konstanten | 24 |
| Abbildung 25: Datenbrücke für Berechnungen | 26 |
| Abbildung 26: SQL-Statement | 26 |
| Abbildung 27: Resultat nach Ausführung der Pipeline | 27 |
| Abbildung 28: Resultat nach Formatierung..... | 28 |
| Abbildung 29: Datenmanipulation am SQL-Statement über den Benutzer | 28 |
| Abbildung 30: Kombination von Konstanten mit Feldinhalte | 29 |

1 Einführung

Riesige Datenmengen entstehen heute in Unternehmen aller Ausrichtungen, in Forschungsprojekten an Universitäten, in Verwaltungen oder sogar im Privathaushalt. Hierbei werden insbesondere in Unternehmen mit einer automatisierten Produktion Daten erzeugt, deren Auswertung und Darstellung für das Management die Grundlage für Investitionsentscheidungen des Produktions- und Verteilungsprozesses darstellen. In der Regel werden diese Daten in unterschiedlich gearteten Datenbanken oder in unterschiedlich strukturierten Formaten, wie XML oder CSV abgelegt.

Die Anwendung ist auf den Datenzugriff über ODBC-Treiber ausgelegt, der sich in vielen Bereichen als Standard etabliert hat. Es kann auf alle Datenhaltungssysteme zugegriffen werden, für die ODBC-Treiber verfügbar sind.

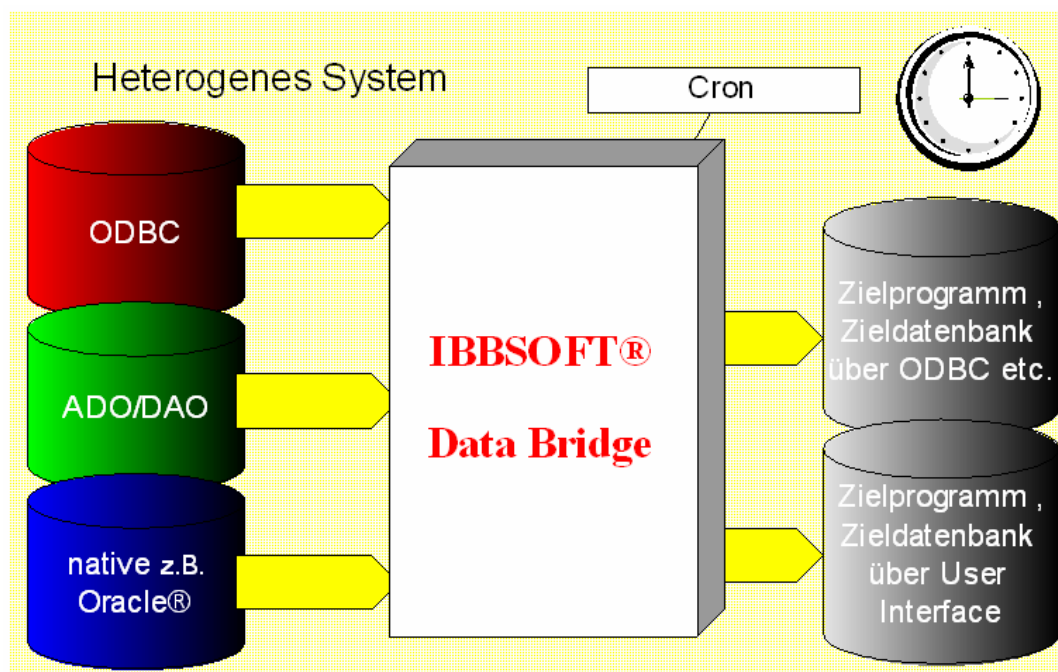


Abbildung 1: Zusammenführung heterogen gehaltener Daten

IBBSOFT® Data Bridge ist ein Programm zur Erstellung von Schnittstellen zwischen beliebigen Datenbanken und für den Austausch der Daten darüber. Der Benutzer definiert und verwaltet die Zuordnung der Felder der Tabellen der Quell- und Zieldatenbank. Ein Assistent (Wizard) unterstützt den Nutzer bei der Erstellung der Zuordnungen. Für den Transfer der Daten werden aus den getroffenen Zuordnungen Select-, Insert- und Update-Anweisungen generiert und ausgeführt.

Zudem steht dem Benutzer die Möglichkeit zur Verfügung, auf einfache Weise die Feldinhalte zu erzeugen, zu ändern oder zu formatieren, ohne auf Kenntnisse in SQL zurückzugreifen. Mittels integrierten Serverdiensts lassen sich Pipelines aus der „Ferne“ starten.

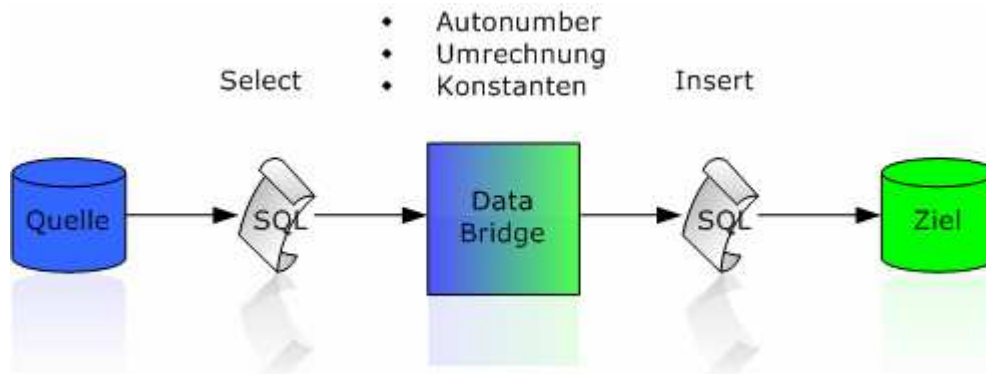


Abbildung 2: Übersicht Datentransfer und Datenmanipulation

Das Programm ist unter den Windows®-Betriebssystemen lauffähig. Hierbei werden keine besonderen Anforderungen an die Computerarchitektur gestellt.

Es wird vorausgesetzt, dass der Nutzer Kenntnis vom Aufbau der gewählten Datenbanken sowie der Verfahrensweise bei SQL-Anweisungen hat.

Das Programm Data Bridge ist ein Werkzeug zur Unterstützung des fachlich versierten Anwenders bei der Erstellung von vielfältigen Schnittstellen.

2 Installation

Für die Installation benötigen Sie Administrationsrechte auf dem System.

Legen Sie die Data Bridge-CD in Ihr CD- oder DVD-ROM-Laufwerk. Doppelklicken Sie auf die Datei *setup.exe* und folgen Sie den Anweisungen des Installationsassistenten.

3 Data Bridge

Nach Starten der Anwendung *Data Bridge.exe* erscheint die nachfolgende Programmansicht.

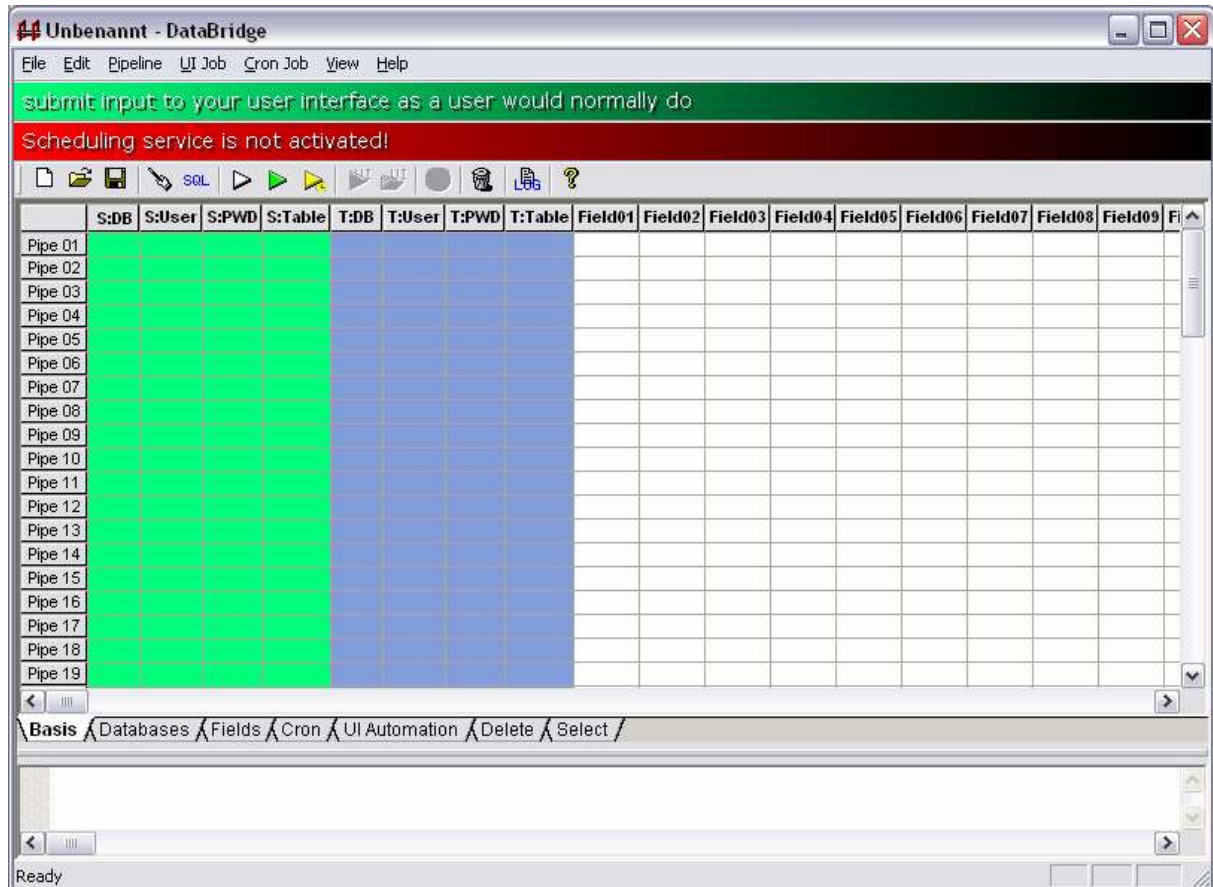


Abbildung 3: Hauptansicht des Programms Data Bridge

Hauptelement des Programms ist eine Tabellenansicht. Die Zeilen der Tabelle stellen die Daten-Verbindungen (engl. Data pipelines) zwischen den Tabellen jeweils einer Quell- und Zieldatendank dar. Die Tabelle besteht aus 256 Zeilen. Folglich sind 256 Datenbrücken möglich, die gleichzeitig ausgeführt, dass heißt, transferiert werden können. Das Registerblatt **Basis** zeigt die kompletten Datenbrücken, dagegen geben die anderen Registerblätter nur den Ausschnitt der benannten Spalten der Tabelle wieder.

Unterhalb der Tabelle befindet sich ein Textfenster, in dem die Logbuch-Einträge sichtbar sind. Neben Statusangaben werden hier Testausgaben zu erzeugten SQL-Befehlen ausgegeben. Der Nutzer hat jederzeit Einblick in den Datenfluss.

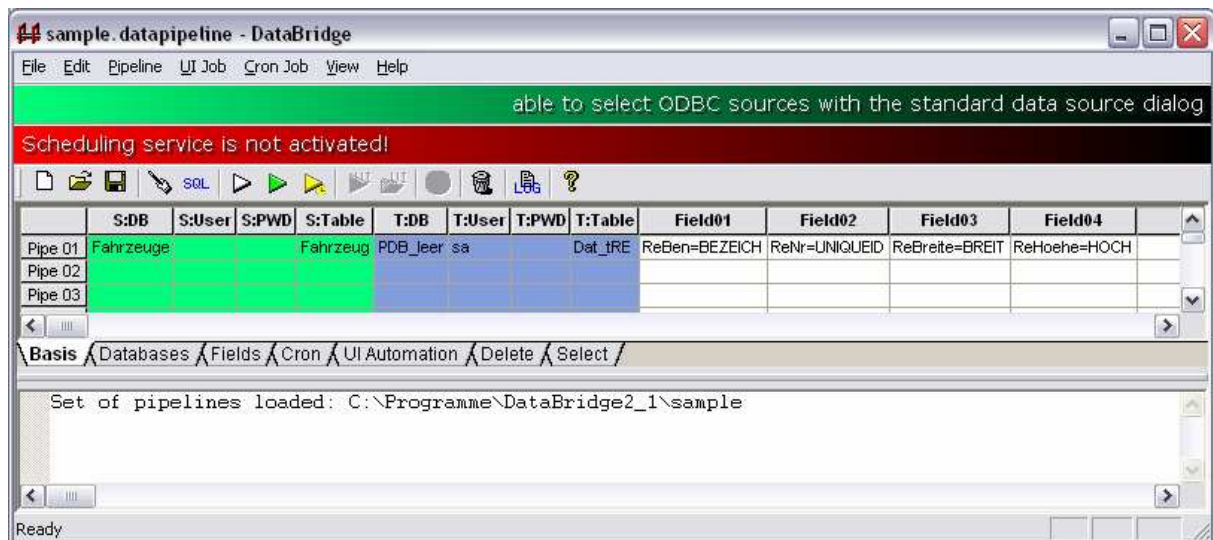




Abbildung 4: Datenbrücke zwischen Fahrzeug und PDB_jeer


In Abbildung 4 ist eine Datenbrücke dargestellt. Der Anwender kann die erforderlichen Eingaben direkt in die Zellen vornehmen oder den integrierten Assistenten (Wizard) benutzen. Die Zusammensetzung einer Pipeline wird im Abschnitt 4.1 erläutert.

Die erfassten Datenbrücken können durch das Symbol  *Execute without writing any data* getestet werden, ohne dass Daten transferiert werden. Das Programm setzt die SQL-Anweisungen zusammen und gibt diese in der Log-Datei aus. Dem Anwender bietet sich nun die Möglichkeit, die generierten SQL-Anweisungen zu überprüfen.

Der tatsächliche Datentransfer wird mit dem Symbol  *Execute* gestartet. In jeder Zeile werden eine Verbindung zur Quelldatenbank und eine Verbindung zur Zieldatenbank aufgebaut. Kann die Verbindung zu beiden Datenbanken erfolgreich aufgebaut werden, so wird aus den in den Feldspalten hinterlegten Bezeichnern ein SELECT-Statement generiert und ausgeführt. Die gewonnenen Daten werden mittels eines ebenfalls automatisch generierten INSERT-Statements in die Zieltabelle eingetragen.

4 Datentransfer

4.1 Erstellen der Datenbrücke mit Hilfe des Wizard

Data Bridge verfügt über einen integrierten Assistenten (Wizard), der den Benutzer bei der Erstellung der Zuordnungen der Datenquellen unterstützt. Durch Klicken auf das Symbol  wird der Wizard-Dialog Abbildung 5 geöffnet.

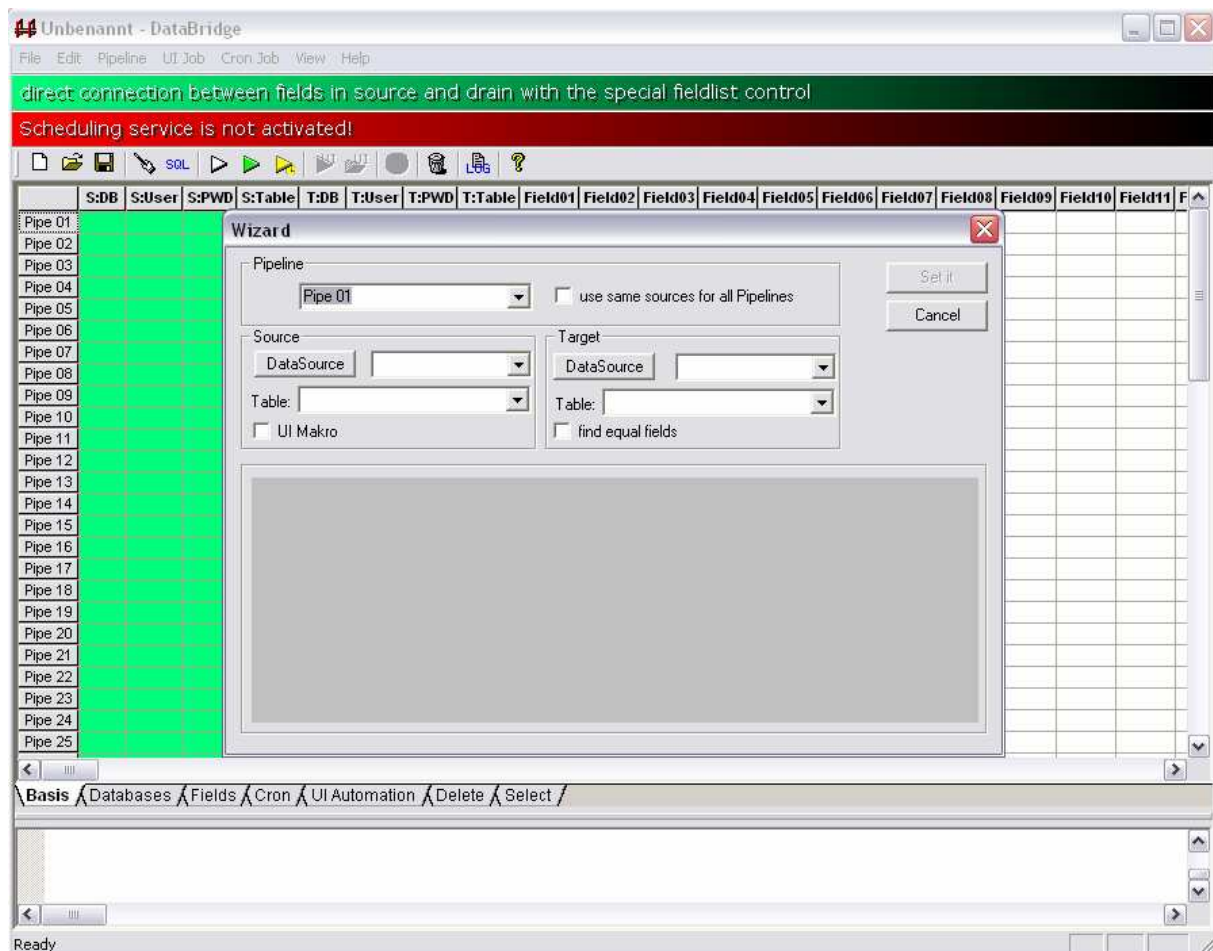



Abbildung 5: Ansicht Wizard

Der Wizard bezieht sich immer auf genau eine Zeile (Pipe) der Tabelle. Die gewünschte Pipelinezeile wird durch Selektion in der Dropdownbox des Wizard-Fensters festgelegt. Rechts neben der Pipeline-Auswahl ist eine Checkbox „*use same sources for all Pipelines*“, durch die bestimmt werden kann, ob alle Pipelines die gleiche Quell- und Zieldatenbank verwenden.

Unterhalb des Bereichs Pipeline befinden sich der Bereich *Source* und *Target* für die Quell- sowie die Zieldatenbank. In der jeweils oberen Dropdownbox kann die ODBC-Verbindung zur Datenbank ausgewählt werden. Es sind alle im System registrierten ODBC-Verbindungen auswählbar. Für den Fall, dass neue ODBC-Verbindungen angelegt werden sollen, kann der Standard-Dialog von Windows zur ODBC-Verwaltung geöffnet werden. Dies erfolgt mit dem Symbol . Der Inhalt des erscheinenden Dialogs ist von der jeweiligen Windows-Version abhängig. Daher soll hier nicht näher auf die Spezifikation der ODBC-Verwaltung eingegangen werden.

Nachdem die Datenbank ausgewählt wurde, muss die Tabelle ausgewählt werden. Das erfolgt über die Dropdownbox *Table*. In dieser Box sind alle Tabellen, Systemtabellen und Views der Datenquelle aufgelistet. Wird eine Tabelle ausgewählt, listen sich die darin enthaltenen Felder im unteren Abschnitt des Fensters auf (siehe Abbildung 6).

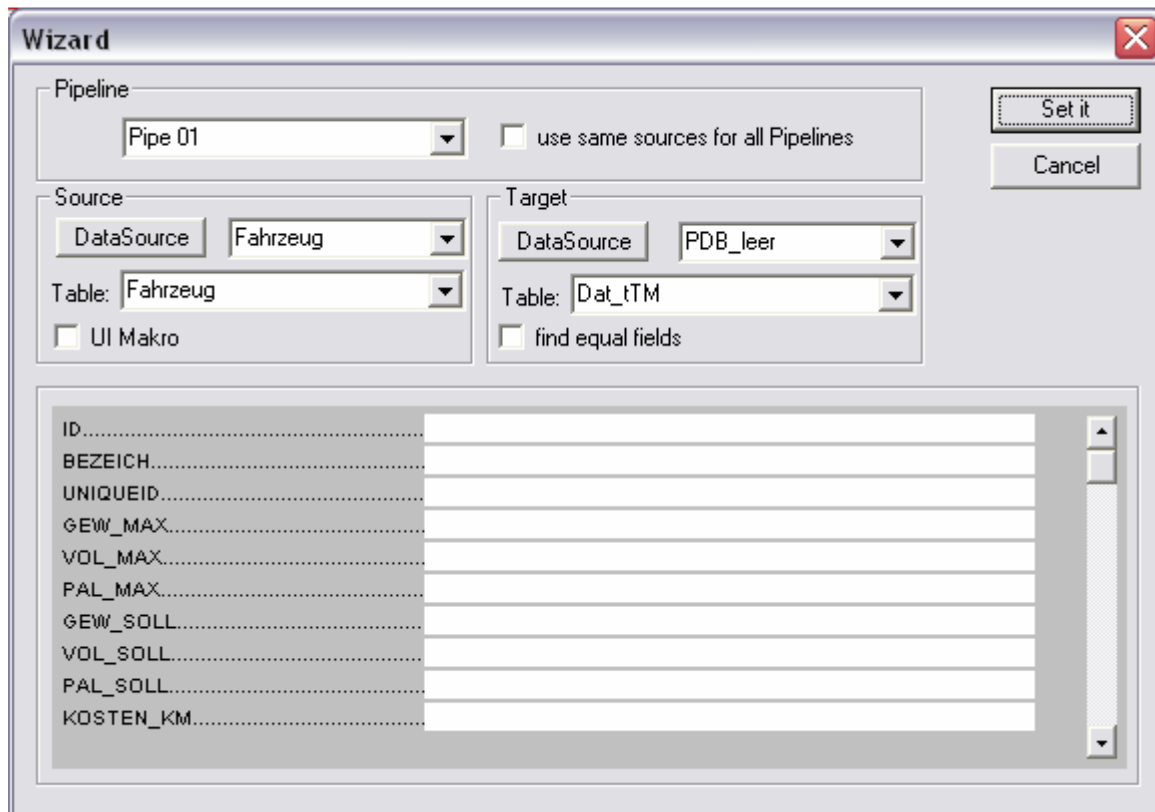


Abbildung 6: Wizard-Fenster mit ausgewählter Quelltable

Nun kann in der Target-Region die Zieltabelle aus der Dropdownbox gewählt werden.

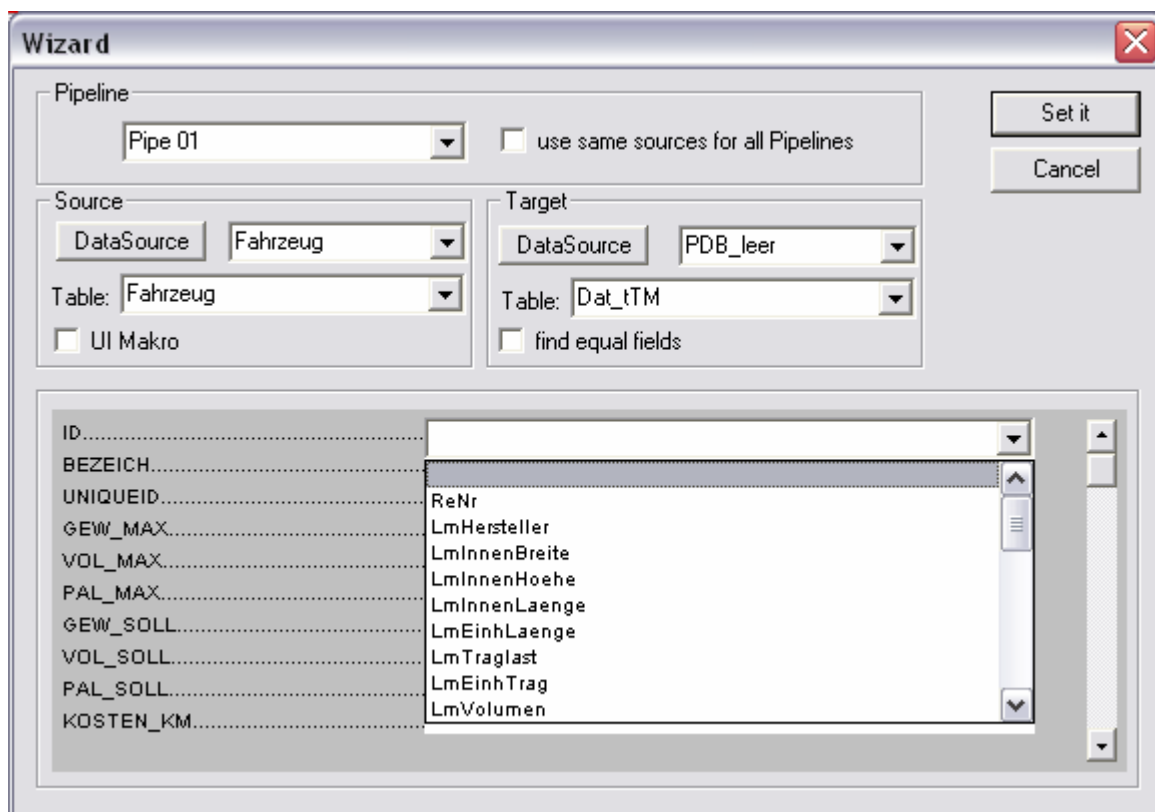
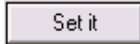


Abbildung 7: Die Felder der Zieltabelle

Nach Wahl der Zieltabelle sind auf der rechten Seite des Controls Dropdownboxen mit den Feldern der Zieltabelle angelegt worden. Einem Quellfeld wird genau ein Zielfeld zugeordnet.

Nachdem allen interessierenden Feldern aus der Quelltablelle Ziele zugeordnet wurden, kann durch Betätigen des Symbols  die erstellte Pipeline in die Tabelle der Hauptansicht übernommen werden.

4.2 Speichern und Öffnen von Pipelines

Die in der Tabelle der Hauptansicht erstellten Pipelines können gemeinsam in einer Datei mit der Standardendung *.pipeline* gespeichert werden. Die Ablage der Datei erfolgt im Allgemeinen im Programmverzeichnis. Über die Standard-Dateidialoge von Windows kann auch jedes andere Verzeichnis gewählt werden, für das ausreichende Zugangsrechte vorliegen.

Die abgelegten Pipeline-Dateien können mit der Anwendung Data Bridge verknüpft werden. Im „Datei öffnen“-Dialog des Programms können bestehende Pipeline-Dateien ausgewählt und geöffnet werden.

4.3 Ausführen der Pipeline über die DOS-Eingabeaufforderung

Data Bridge kann bestehende Pipeline-Dateien (*.pipeline) über die DOS-Eingabeaufforderung ausführen. Der DOS-Befehl lautet z.B.:

```
C:\Programme\DataBridge>databridge.exe "mypipe.pipeline"
```

Daraufhin führt Data Bridge die Datei mypipe.pipeline aus. Das Programm Symbol ist dabei in der Systray vertreten. Nach erfolgter Ausführung beendet sich das Programm selbständig.

Log-Mitschnitte sind in diesem Modus deaktiviert. Es ist möglich, die Anwendung mehrmals parallel zu öffnen und so mehrere Pipelines nebeneinander auszuführen.

4.4 Weitere Funktionalitäten des Wizard

4.4.1 Automatisches Verknüpfen gleichnamiger Felder

Im Bereich *Target* befindet sich ein Kontrollkästchen „*find equal fields*“. Bei Aktivierung und anschließender Auswahl einer Tabelle werden die Zuordnungen gleichnamiger Felder beider Datenquellen automatisch gefüllt (siehe Abbildung 8).

Eine Auswahl oder Entfernen der unerwünschten Zuordnung per Hand ist weiterhin möglich.

4.4.2 Voreinstellung der Datenbanken

Prinzipiell können in jeder Zeile andere Quell- und Zieldatenbanken benutzt werden. Beschränken sich die Zuordnungen auf die bereits festgelegten Datenbanken, können diese für jede Datenpipeline voreingestellt werden. Dazu muss in das Kontrollkästchen „*use same sources for all Pipelines*“ ein Häkchen gesetzt werden. Die blau und grün gefärbten Spalten für Quelle und Ziel werden anschließend ausgeblendet.

Entfernen des Hakens aus dem Kästchen blendet die Spalten wieder ein.

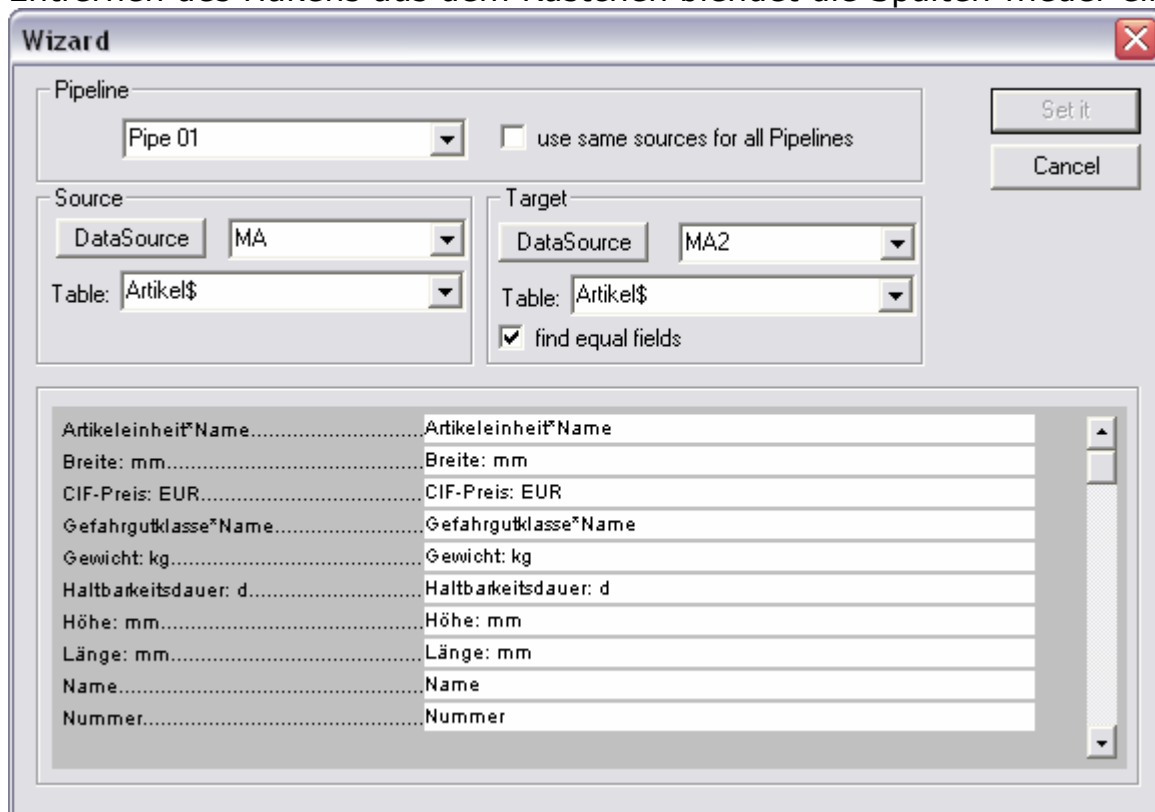


Abbildung 8: Wizard Version

4.4.3 UI Automation

Die UI Automation ist der Datentransfer über die Benutzeroberfläche eines Programms, die über keinen Importfilter verfügen. Es wird im Data Bridge lediglich die Quelle festgelegt. Das Ziel ist zum Beispiel ein Formular, in denen die Werte der Quelle eingetragen werden. Für das Ziel ist zusätzlich ein MAKRO zu erstellen und in die Spalte *UI Automation* mit Pfad und Name der Makro-Datei eingetragen (siehe Abbildung 9).

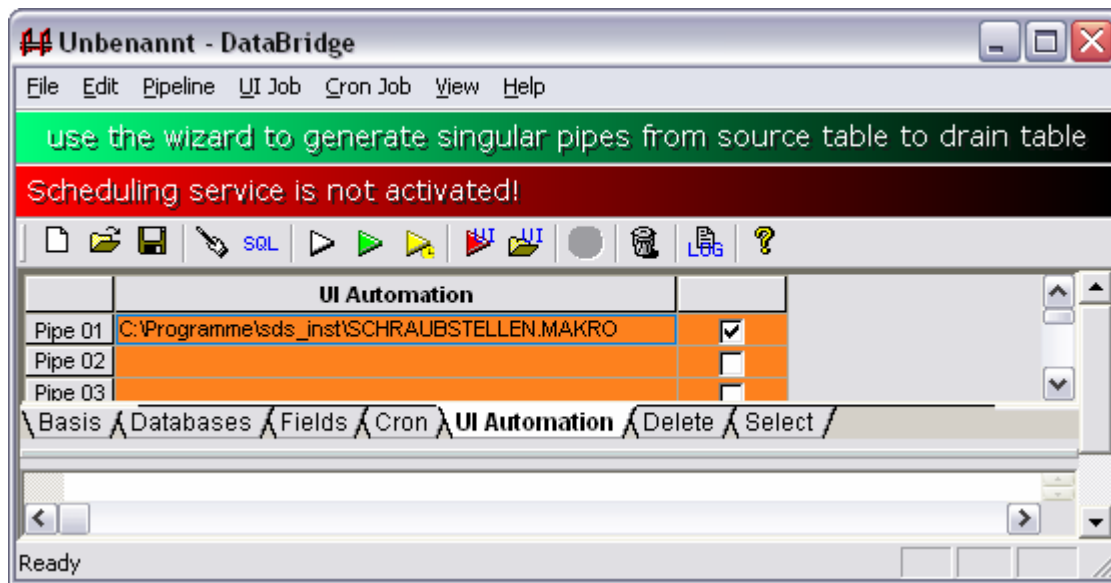


Abbildung 9: Makro-Datei eintragen

Die MAKRO-Erstellung erfolgt außerhalb von Data Bridge in einem Texteditor. Die Syntax ist einschlägigen Dokumentationen zum Thema Windows scripting und Macro scripting zu entnehmen.

Nach Auswahl einer beliebigen Zeile ist der Wizard zu starten. Im Dialogfenster des Wizard werden die Quelldatenbank und die gewünschte Tabelle ausgewählt. Daraufhin ist der Haken im Kontrollkästchen „UI Makro“ zu setzen.

Der Wizard ist so konzipiert, dass die Quelldatenbank nicht mit der Zieldatenbank identisch sein kann, da dieses Vorgehen nicht von jedem ODBC-Treiber unterstützt wird. Wird jedoch die UI Makrosteuerung aktiviert, setzt das Programm Ziel und Quelle gleich und stellt damit die Felder für die Makro Automation bereit (siehe Abbildung 10).

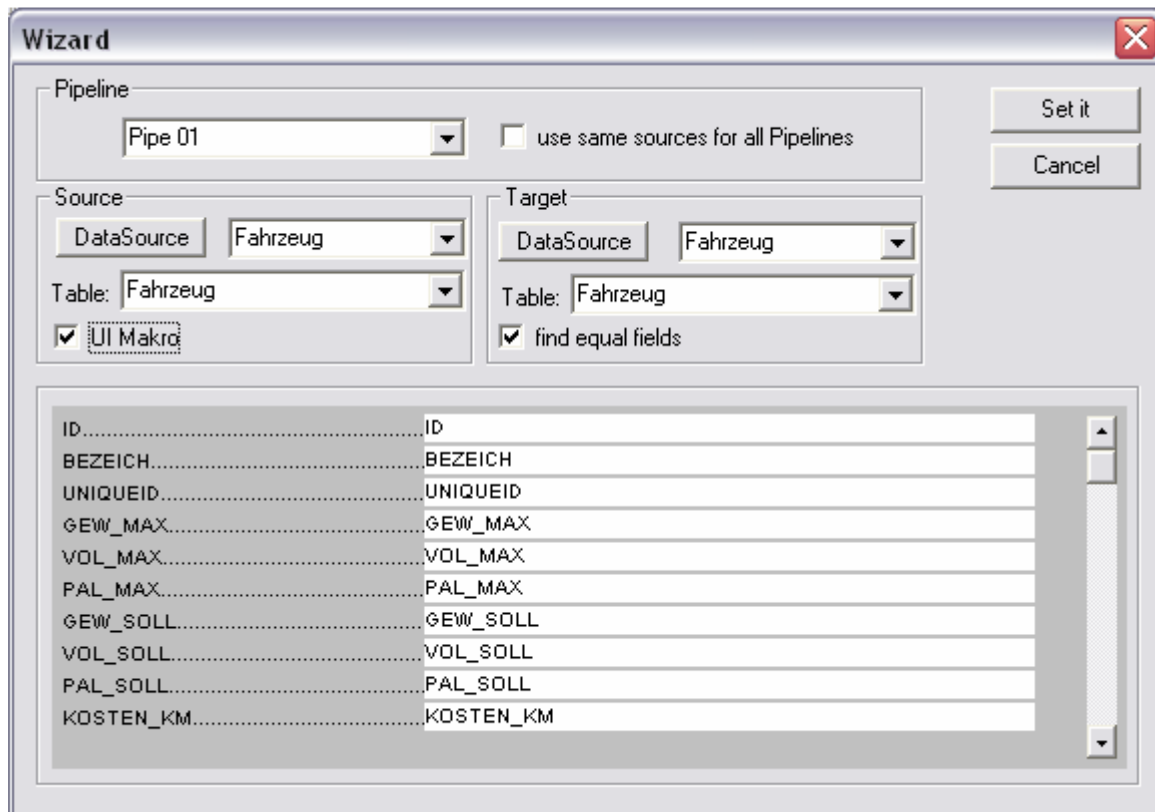




Abbildung 10: Wizard mit MAKRO Aktivierung

Alle Felder der Tabelle werden in der Hauptansicht ausgegeben. Wenn die Zelle der Spalte UI Automation mit dem Cursor ausgewählt wird, aktiviert sich das Symbol  „UI File selection“ zum Öffnen des Datei-Dialoges, in dem die Makrodatei ausgewählt werden kann.

Das Makro kann zum Testen für die ausgewählte Datenverknüpfung gestartet werden. Steht der Cursor in einer Zeile der Spalte UI Automation, aktiviert sich das Symbol  „Test selected UI Job“ in der Menüzeile. Mit Klick auf das Symbol wird die aktuelle Zeile (Datenpipeline) mit dem Makro ausgeführt. Es ist darauf zu achten, dass für die Ausführung des Makros alle verwendeten Anwendungen offen sind.

4.4.4 UPDATE-Funktion

Mit der UPDATE Funktion können bestehende Daten in der Zieldatenbank überschrieben werden. Dazu wurden im Wizard drei weitere Felder UPDATE Key 1, UPDATE Key 2 und UPDATE Key 3 ergänzt. Zu UPDATE Key werden die Felder zugeordnet, die als Vergleichskriterium dienen.

Beachte: Jedes Feld, welches dem UPDATE Key zugeordnet wird, muss vorher einem konkreten (beliebigen) Feld zugeordnet werden. Diese Zuordnung dient allein als Platzhalter und wird im Weiteren nicht beachtet. Der erzeugte SQL-Befehl wird korrekt umgesetzt.

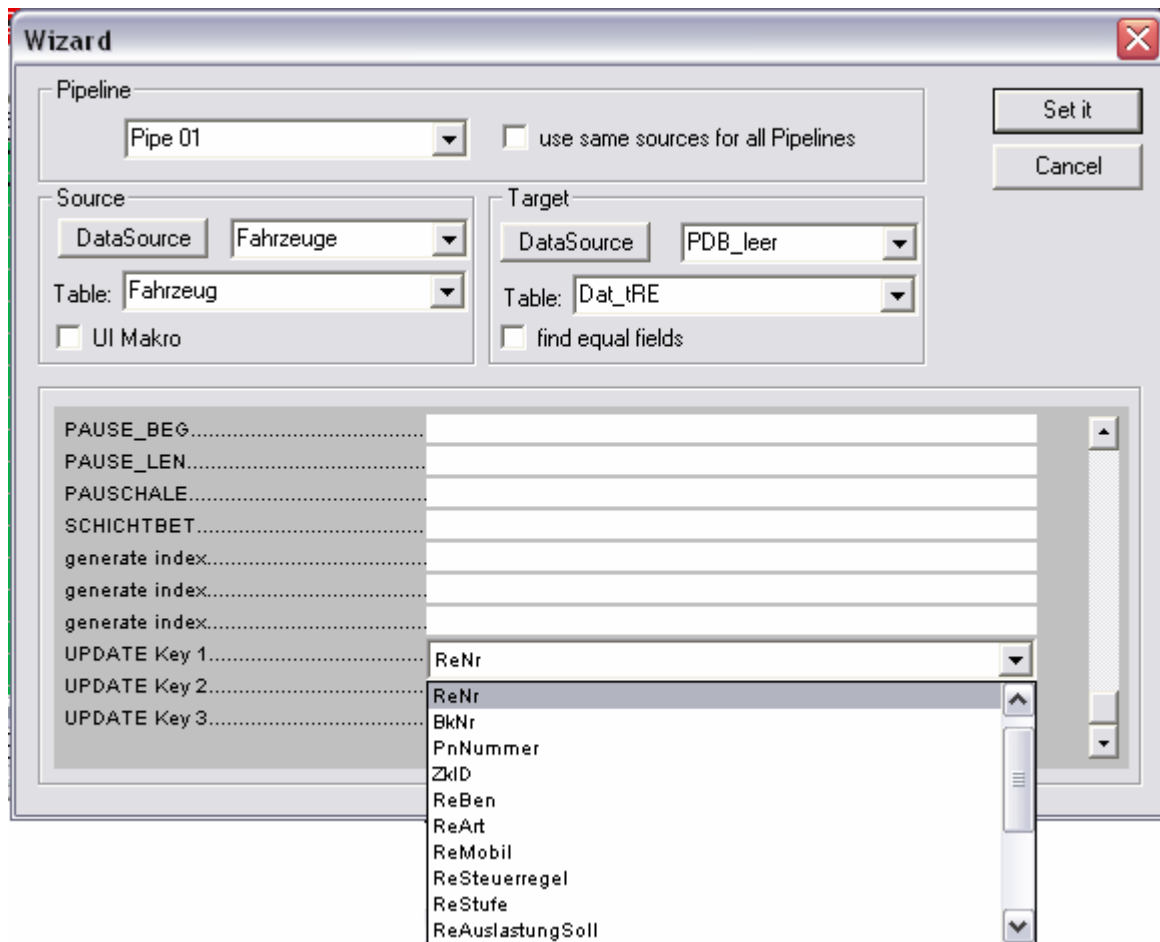


Abbildung 11: Feld ReNr ist Kriterium für Update

Im Beispiel wurde das Feld ReNr als Update Key 1 ausgewählt. Die SQL-Anweisung zum Update des Datensatzes in der Zieldatenbank könnte dann wie folgt lauten:

```
Update Dat_tRE
SET

ReBen='UL-MJ222',
ReBreite='0',
ReHoehe='0'

WHERE ReNr='124'
```

Die Where-Bedingung der Update-Anweisung wird aus den UPDATE Keys erzeugt. Die einzelnen Keys sind mit einer UND-Verknüpfung versehen.

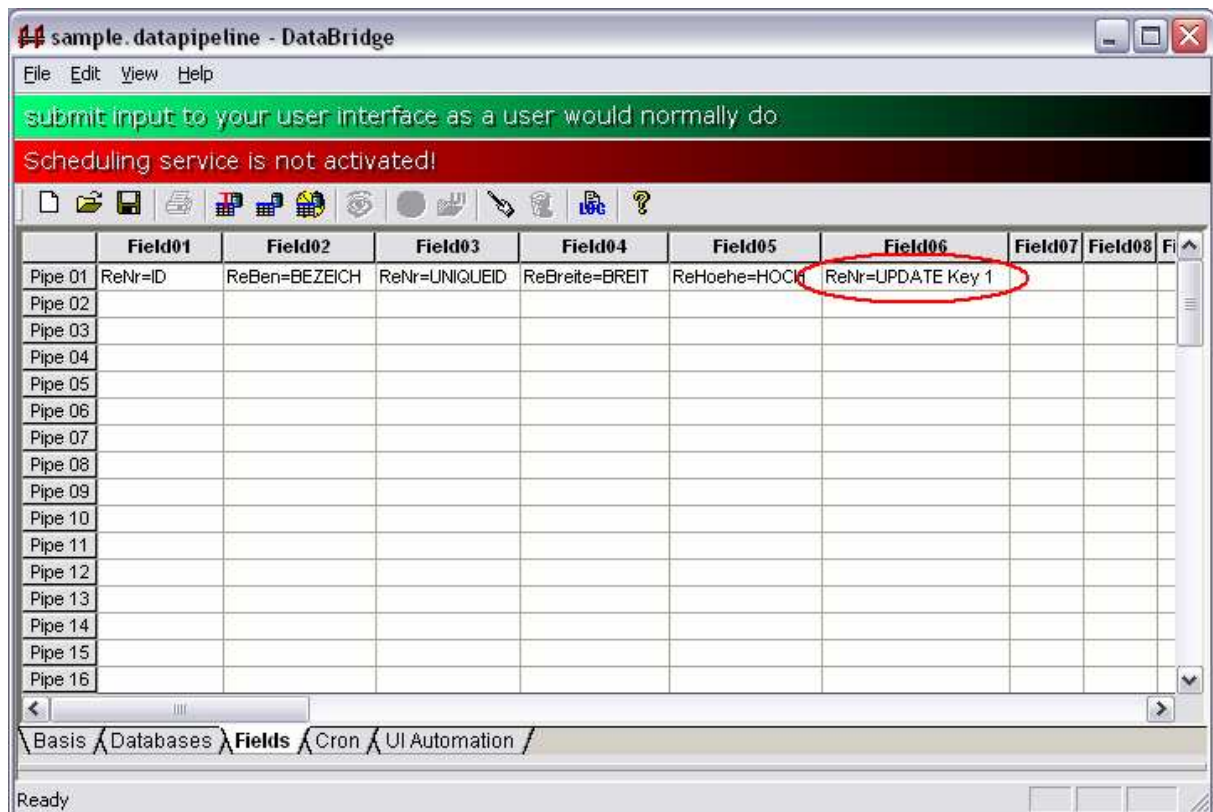


Abbildung 12: Darstellung der Update-Funktion in einer Pipeline

Wird der Update-Key in der Tabelle nicht gefunden, erfolgt keine Aktion. Wird kein Wert in der Where-Bedingung angezeigt, liegt es womöglich daran, dass das Feld nach UPDATE Key nicht zusätzlich einem konkreten Tabellenfeld zugeordnet wurde (siehe Beachte).

Eine mögliche Strategie zur Sicherstellung, dass ein Datensatz nur einmal in eine Tabelle vorhanden ist, wird im Weiteren angegeben.

1. Pipeline mit Update
2. Pipeline

Dabei werden zwei gleiche Pipelines angelegt, die sich lediglich um einen Update Key unterscheiden. Existiert der Datensatz im Ziel bereits, wird dieser überschrieben. Existiert dieser nicht, so gewährleistet das Insert die erfolgreiche Datenübernahme.

Existierte der Datensatz bereits und es wird ein Insert-Befehl ausgeführt, so schlägt dieser fehl und keine Aktion wird ausgeführt.

4.4.5 Kopieren einer Pipeline

Wird die Zeile einer bereits ausgefüllten Pipeline markiert, ehe der Wizard aufgerufen wird, so füllen sich alle bekannten Felder im Wizard selbst aus. Wird anschließend in der Region Pipeline eine andere (freie) Pipeline

eingestellt, überträgt das Programm nach Betätigung von die selektierte und im Wizard eventuell modifizierte Pipeline auf den neuen Platz. Hierdurch ist es möglich, einer Quelldatenbank mehrere Zieldatenbanken komfortabel zuzuordnen, wobei sich der Schreibaufwand der Pipelines erheblich reduziert. In diesem Fall wäre nur die Zieldatenbank zu ändern.

4.5 WHERE-Bedingung

Die Datenbestände der Quelldatenbank können mit einer WHERE-Bedingung eingegrenzt werden. Die WHERE-Bedingung ist manuell nachzutragen. Es ist darauf zu achten, dass der Nachtrag hinter den Zuordnungsfeldern der Pipeline eingetragen wird.

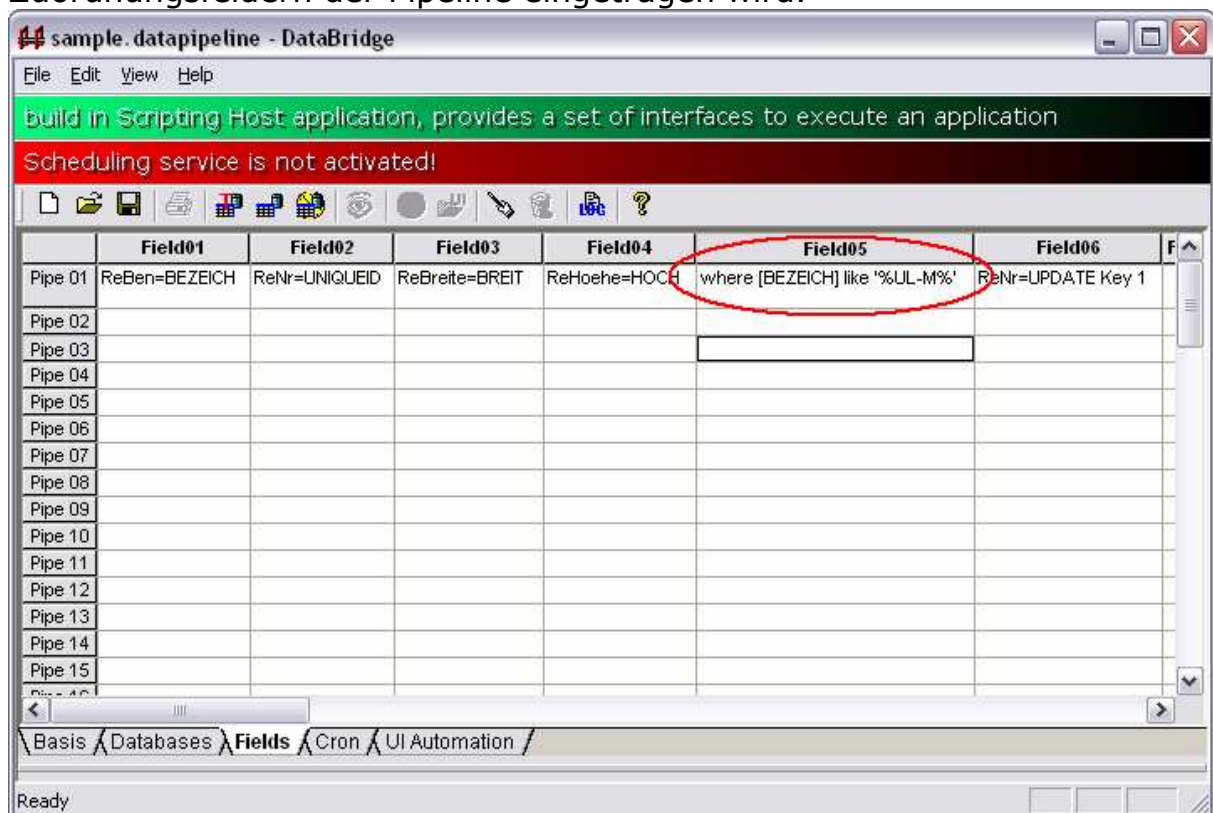


Abbildung 13: Where-Bedingung in der Pipeline

In Abbildung 13 ist der SQL-Bestandteil

Where [BEZEICH] like '%UL-M%'

eingetragen worden. Es sollen nur die Daten aus der Quelle ausgewählt werden, die im Feld BEZEICH den Wortteil 'UL-M' enthalten.

Resultat der Log-Datei: (vgl. Abbildung 13)

```

11/25/07 18:35:09 : Select `BEZEICH`, `UNIQUEID`, `BREIT`, `HOCH` from `Fahrzeug` where [BEZEICH]
like '%UL-M%'
11/25/07 18:48:00 : Insert into Dat_tRE (ReBen, ReNr, ReBreite, ReHoehe) VALUES ('UL-MJ 183', '124', '0',
'0')
11/25/07 18:48:00 : Insert into Dat_tRE (ReBen, ReNr, ReBreite, ReHoehe) VALUES ('UL-MN 426', '112', '0',
'0')
11/25/07 18:48:00 : Insert into Dat_tRE (ReBen, ReNr, ReBreite, ReHoehe) VALUES ('UL-MM 297', '114', '0',
'0')
11/25/07 18:48:00 : Insert into Dat_tRE (ReBen, ReNr, ReBreite, ReHoehe) VALUES ('UL-MJ 183', '125', '0',
'0')
11/25/07 18:48:00 : Insert into Dat_tRE (ReBen, ReNr, ReBreite, ReHoehe) VALUES ('UL-MJ 183', '126', '0',
'0')
11/25/07 18:48:00 : Insert into Dat_tRE (ReBen, ReNr, ReBreite, ReHoehe) VALUES ('UL-MJ 183', '127', '0',
'0')
11/25/07 18:48:00 : Insert into Dat_tRE (ReBen, ReNr, ReBreite, ReHoehe) VALUES ('UL-MJ 183', '128', '0',
'0')
11/25/07 18:48:01 : Insert into Dat_tRE (ReBen, ReNr, ReBreite, ReHoehe) VALUES ('UL-MJ 183', '129', '0',
'0')

```

Es gilt die Zuordnung Ziel=Quelle: ReBen=BEZEICH. Die Quelle BEZEICH wurde eingeschränkt auf bestimmte Werte. Demnach sind diese Werte in die Zieltabelle im Feld ReBen eingetragen worden.

Die Where-Bedingung kann mit AND und OR beliebig erweitert werden.

Alternativ kann ein SQL-Statement benutzt werden. Weitere Informationen sind in den Abschnitten Select bzw. Berechnungen zu finden.

Die eckigen Klammern in der WHERE-Klausel sollten möglichst immer um die Feldbezeichnung stehen. Erfahrungsgemäß weichen die Formate einiger Datenbanken vom SQL-Standard ab. Indessen wird das Format bei der automatisch generierten SQL-Abfrage berücksichtigt.

4.6 ORDER BY-Bedingung

Anstelle oder zusammen mit der WHERE-Bedingung kann eine ORDER BY Klausel nachträglich vom Nutzer eingetragen werden.

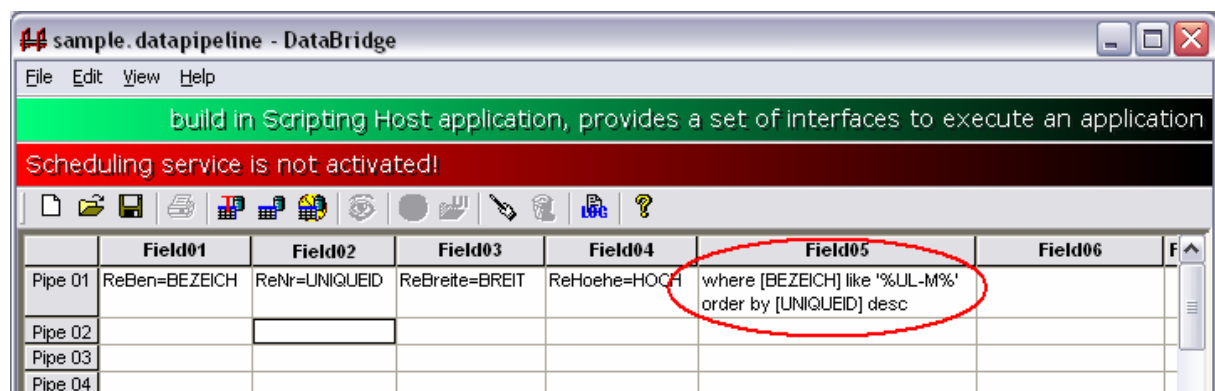


Abbildung 14: Order by-Bedingung in der Pipeline

Die Ausgabe der Log-Datei:

```
11/25/07 18:53:38 : Select `BEZEICH`, `UNIQUEID`, `BREIT`, `HOCH` from `Fahrzeug` where [BEZEICH]
like '%UL-M%' order by [UNIQUEID] desc
11/25/07 18:53:38 : Insert into Dat_tRE (ReBen, ReNr, ReBreite, ReHoehe) VALUES ('UL-MJ 183', '129', '0',
'0')
11/25/07 18:53:38 : Insert into Dat_tRE (ReBen, ReNr, ReBreite, ReHoehe) VALUES ('UL-MJ 183', '128', '0',
'0')
11/25/07 18:53:38 : Insert into Dat_tRE (ReBen, ReNr, ReBreite, ReHoehe) VALUES ('UL-MJ 183', '127', '0',
'0')
11/25/07 18:53:38 : Insert into Dat_tRE (ReBen, ReNr, ReBreite, ReHoehe) VALUES ('UL-MJ 183', '126', '0',
'0')
11/25/07 18:53:38 : Insert into Dat_tRE (ReBen, ReNr, ReBreite, ReHoehe) VALUES ('UL-MJ 183', '125', '0',
'0')
11/25/07 18:53:38 : Insert into Dat_tRE (ReBen, ReNr, ReBreite, ReHoehe) VALUES ('UL-MJ 183', '124', '0',
'0')
```


4.7 Delete

Die Funktion Delete erlaubt dem Benutzer, die Zieltabelle vollständig zu leeren, bevor die Daten erneut eingefügt werden. Die Funktion kann für jede Datenbrücke gesetzt werden. Dazu ist in der ausgewählten Zeile in der Spalte Delete durch Doppelklick ein Häkchen zu setzen.


Achtung! Ist diese Funktion eingestellt, wird jede Datenzeile aus der Tabelle entfernt.

Vor Ausführung wird dem Nutzer eine Sicherheitsabfrage gestellt. Die Delete Funktion ist im automatisierten Modus und beim Update nicht verfügbar.

4.8 Select

Im Registerblatt *Select* können SELECT-Anweisungen selbst eingetragen werden oder mit dem Symbol  für die Pipeline erzeugt werden. Dieser SQL-Befehl wird direkt an die Quell-Datenbank weitergegeben. Demnach ist jede Funktionalität, die mit SQL beschreibbar ist, möglich und beschränkt sich nur durch den SQL-Befehlsvorrat der Datenbank.

Die Anzahl der Spalten müssen mit den Einträgen der Felder im Registerblatt *Fields* übereinstimmen.

Gebräuchlich ist die Erstellung der Pipeline mit Hilfe des Wizard. Anschließend kann sich der Benutzer den SQL-Befehl mit Klicken auf das Symbol  „SQL-Statement“ erstellen lassen. Dieser SQL-Befehl befindet sich dann in der ausgewählten Zeile in der Spalte SQL-Statement bzw. im Registerblatt *Select*.


Eine Anwendung wird im Kapitel Berechnungen beschrieben.

Doppelklicken in die Zelle der SQL-Anweisung öffnet einen SQL-Editor. Im Editor können die erstellten SQL-Anweisungen mit einer Testausführung überprüft werden. Eine automatische Formatierung erleichtert die Übersicht.

Es wird in erster Linie die SQL-Anweisung in der SQL-Statement-Spalte ausgeführt, sofern für die Pipeline eine SQL-Anweisung angelegt wurde. Andernfalls wird aus den einzelnen Feldern die SQL-Anweisung generiert und ausgeführt. Dazu muss die SQL-Statement-Spalte leer sein.

4.9 Log-Mitschnitt

Um die fehlerfreie Ausführung der Datenbrücken überwachen zu können und um den Anwendern die Einrichtung der so genannten Pipelines zu erleichtern, wurde eine Log-Funktionalität in das Programm implementiert. Im Unterverzeichnis `.\LOGS` vom Installationsverzeichnis von Data Bridge werden für jeden Kalendertag Dateien abgelegt, die die SQL-Anweisungen sowie die über den ODBC-Treiber übermittelten Fehler- und Ausführungsmeldungen der Datenbanken enthalten. Die Log-Dateien werden angelegt, sofern eine Datei mit Namen `logflag.log` im Log-Verzeichnis existiert.

Über das Symbol  kann direkt von der Oberfläche aus die Log-Datei des aktuellen Tages angesehen werden.

Aktuelle Einträge werden permanent im Textfenster im unteren Bereich der Programmansicht wiedergegeben.

5 Datenmanipulation

Neben der Übertragung vorhandener Daten können auch neue Daten erzeugt oder bestehende Daten verändert werden. Mit diesen Funktionen können fehlende Primärschlüssel ergänzt, Umrechnungen oder Einträge von Konstanten vorgenommen werden.

5.1 Generate Index

Die Liste der Felder der Quell-Tabelle im Wizard (Abbildung 15) enthält zusätzliche Zeilen mit der Bezeichnung „generate Index“. Dadurch ist die Möglichkeit gegeben, einen automatisch generierten Feldinhalt zu erzeugen. Es umfasst die Erzeugung neuer Daten durch

- Fortlaufende Nummerierung (Autonumber)
- Konstanten.

5.1.1 Fortlaufende Nummer generieren

Die Notwendigkeit einer eindeutigen Nummerierung resultiert aus abweichenden Datenbankstrukturen zwischen Quell- und Zieldatenbank. Beispielsweise wird in der Zieltabelle ein eindeutiger Primärschlüssel verlangt, der in der Quelltable nicht vorhanden ist. Dieser nicht vorhandene Feldinhalt kann nicht übertragen, sondern muss durch das Programm erzeugt werden. Im Wizard wird das entsprechende Feld dem *Generate Index* Feld zugeordnet.

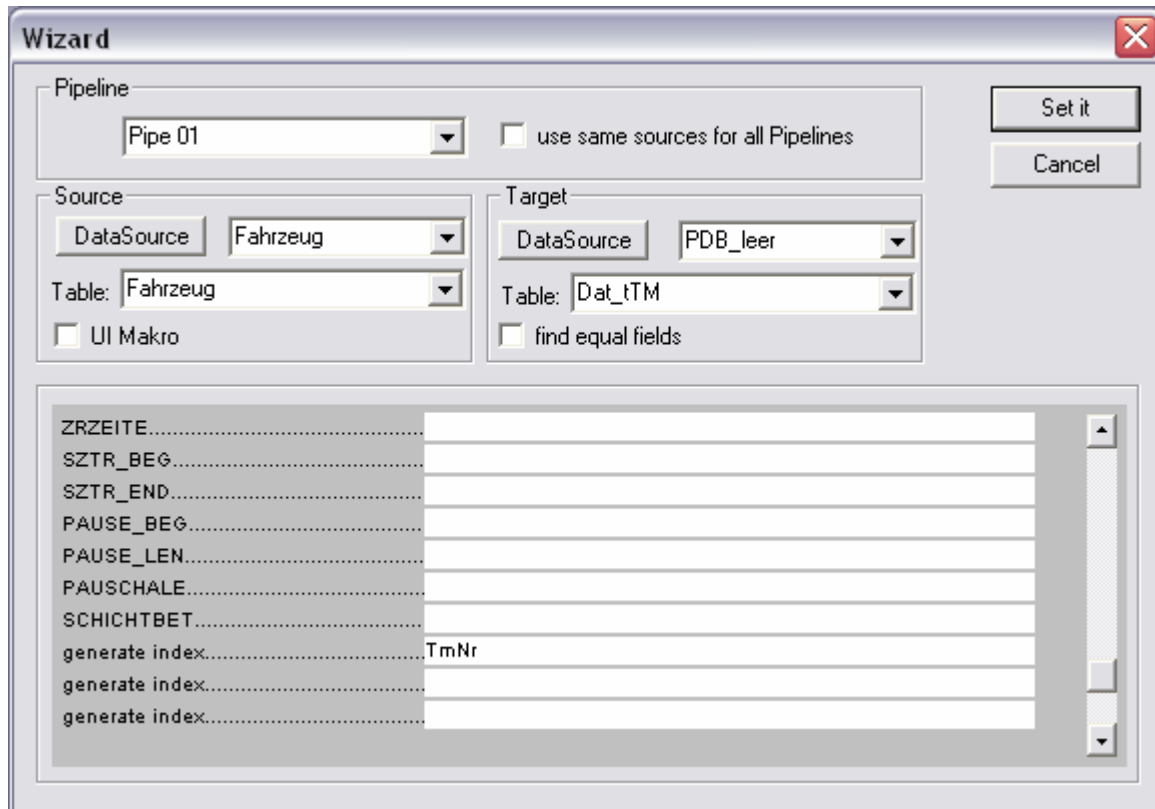


Abbildung 15: Funktion Autonumber

In Abbildung 15 und Abbildung 16 wird in das Feld *TmNr* eine aufsteigend gezählte Zahl eingetragen. Die Notation für den einfachsten Fall ist **TmNr=:#:**

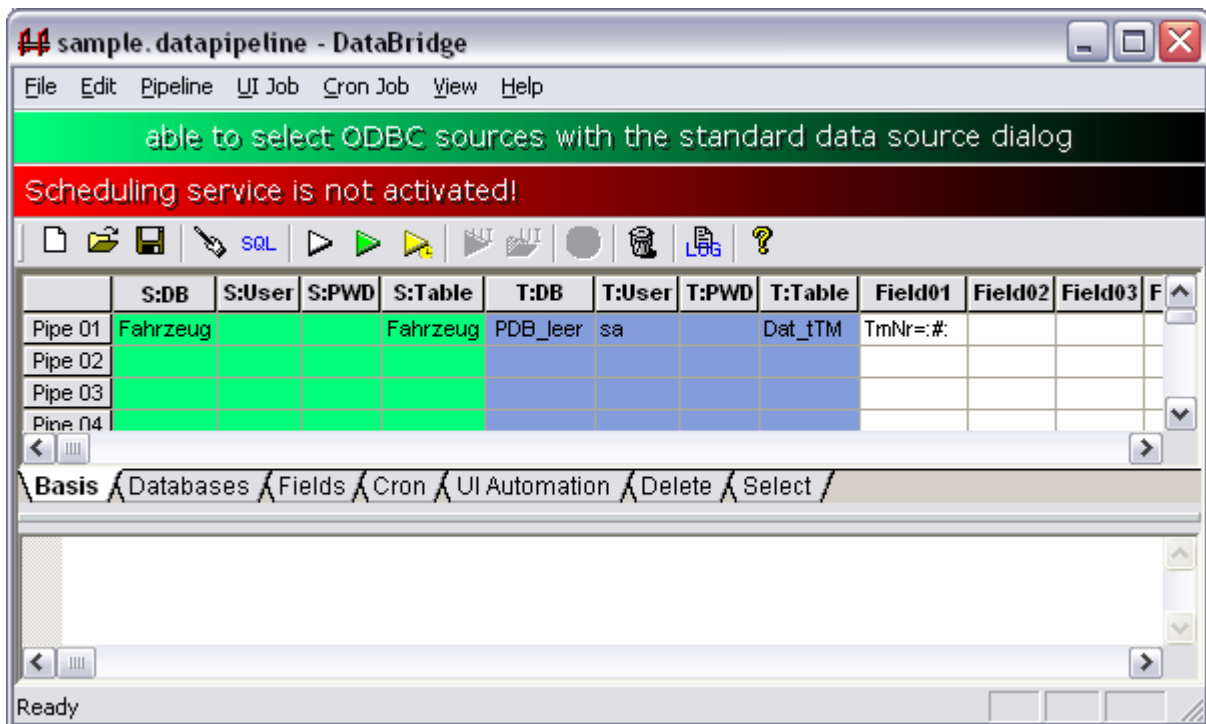


Abbildung 16: Zuordnung mit Autonumber

Durch einfache Textergänzungen lassen sich zusammengesetzte Felder aus Text und Zahlenfolgen erstellen. In Abbildung 17 sind mögliche Notationen und das erzeugte Ergebnis aufgeführt.

| Notation | Ergebnis | Notation | Ergebnis |
|----------|----------|-------------|----------|
| TM:#: | TM1 | :#: %1 | 1 |
| | ... | | ... |
| | TM9 | | 99 |
| | TM10 | | 100 |
| :#: %3 | 001 | :#: %1+25 | 26 |
| | ... | | ... |
| | 009 | | 99 |
| | 010 | | 100 |
| TM:#: %5 | TM00001 | TM:#: %5+12 | TM00013 |
| | ... | | ... |
| | TM00020 | | TM00020 |
| | TM00021 | | TM00021 |

Abbildung 17: Notationen Autonumber

5.1.2 Fortlaufende Nummerierung über mehrere Pipelines

Die Nummerierung kann über mehrere Pipelines fortgesetzt werden. In der ersten Pipeline wird mit der Notation :#: oder einer der Formate aus Abbildung 17 begonnen. In der darauf folgenden Pipeline bekommt das hochzählende Feld das Format :~#: zugewiesen. Das kann so fortgeführt werden, bis ein neuer Zahlenindex begonnen werden soll. Dann wird wieder mit Format :#: angefangen.

Die Formate aus Abbildung 17 sind ebenfalls anwendbar. Dazu muss # durch ~ getauscht werden. Es ist für das Hochzählen nicht zwingend erforderlich, das gleiche Format fortzusetzen.

5.1.3 Konstanten generieren

Unter Umständen soll einem nicht in der Quell-Datenbank vorhandener Wert ein konstanter Wert zugewiesen werden. Dies erfolgt entweder durch den Eintrag [MeinZielFeldName]='[Konstante]' in die Zelle einer Pipe oder mittels Wizard und der Generate Field Zuordnung.

Im Wizard wird das entsprechende Feld dem *Generate Index* Feld zugeordnet. Nach Bestätigung der Zuordnung mit Set it entsteht die Standard-Notation [MeinZielFeldName]=:#: in der Tabellenansicht. Die Zeichen :#: sind durch ein Hochkomma und der Konstante zu ersetzen.

Beispiel:

TmTyp='9

Abbildung 18 zeigt eine Pipeline, in der in der 5. Spalte sowie in der 7. Spalte eine Konstante verwendet wird.

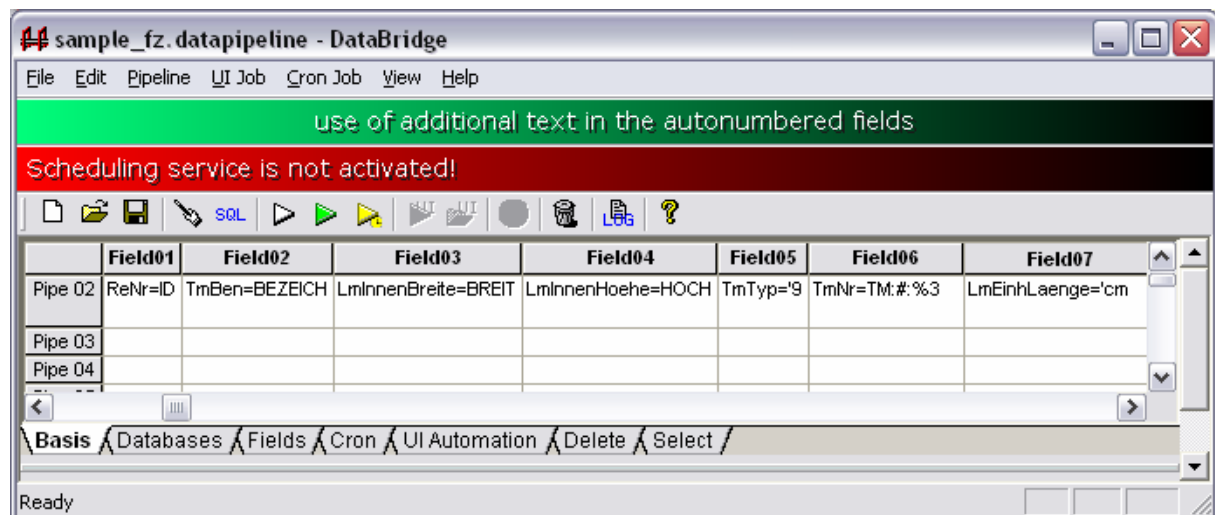


Abbildung 18: Beispiel von Konstanten

Das Resultat nach Ausführung der Pipe 02 ist in Abbildung 19 ersichtlich. Spalte TmTyp bzw. Spalte LmEinhLaenge besitzen in jeder Zeile den gleichen Wert.

| Dat_tTM | | | | | |
|---------|-----------|---------------|-------|-------|--------------|
| ReNr | TmBen | LmInnenBreite | TmTyp | TmId | LmEinhLaenge |
| 1 | UL-ZU 325 | 0.0 | 9 | TM023 | cm |
| 2 | UL-DK 541 | 0.0 | 9 | TM010 | cm |

| | | | | | |
|---|-----------|-----|---|-------|----|
| 3 | UL-HH 213 | 0.0 | 9 | TM011 | cm |
| 4 | UL-MN 426 | 0.0 | 9 | TM012 | cm |
| 5 | UL-HU 387 | 0.0 | 9 | TM013 | cm |

Abbildung 19: Resultat der Datenübertragung in Tabelle Dat_tTM mit Konstanten

5.2 Umrechnungen und Berechnungen

5.2.1 Umrechnungen

Mit Hilfe einer bestimmten Notation können Umrechnungen vorgenommen werden. Umrechnungen sind beispielsweise bei unterschiedliche Maßeinheiten oder Währungen notwendig. Da vielfältige Schreibweisen für Zahlen existieren, müssen weitere Angaben zum Format übergeben werden, die durch ein Ausrufe-Zeichen getrennt sind. Die Formatierung wird nach dem zugeordneten Zielfeld in der Tabelle ergänzt.

Das Format ist wie folgt:

[ZielfeldName]=[QuellfeldName]![Tausender-Trenner]![Dezimal-Zeichen]![Operation]![Berechnungs-Konstante]

z.B.: emp_Salary_brutto=emp_Salary!.,!*!1,19

Als Tausender-Trenner kann ein Leerzeichen, ein Hochkomma, bei Währungen ein Punkt stehen. Ist kein Tausender-Trenner nötig, wird der Trenner weggelassen. Die Formatangabe beginnt demzufolge mit zwei Ausrufezeichen.

Das Dezimal-Zeichen trennt die Ganzzahl vom Bruchteil und wird als Komma, zum Teil auch als Punkt, ausgegeben.

An dritter Stelle wird die Operation benannt (siehe Tabelle 1).


| Zeichen | Operation |
|---------|----------------|
| + | Addition |
| - | Subtraktion |
| * | Multiplikation |
| : | Division |

Tabelle 1: Operationen


Die Berechnungs-Konstante enthält den Umrechnungsfaktor entsprechend der voran gestellten Formatierung, mit der die Umrechnung vorgenommen wird. Die Anzahl der verwendeten Nachkomma-Stellen wird in der Ausgabe übernommen. Wird keine Nachkommastelle angegeben, wird unabhängig vom Format der Quelldaten eine ganze Zahl erzeugt. Die Stellenzahl nach dem Trenner wird als Anweisung für die zu erzeugende

Genauigkeit verstanden. Intern wird mit maximaler Genauigkeit gerechnet und die Ausgabe für die Datenbank formatiert.

Es wird gegenwärtig davon ausgegangen, dass das Zahlenfeld in den verknüpften Datenbanken jeweils das gleiche Format aufweist.

Ist sich der Benutzer nicht im Klaren, welches Format das angegebene Feld besitzt, so kann er sich den erzeugten SQL-Befehl mit Button  *Execute without writing any data* im Editorfenster ausgeben lassen.

5.2.2 Berechnungen

Berechnungen sind mit SQL zu realisieren. Dazu muss eine Datenbrücke erstellt werden, in der das zu berechnende Feld enthalten ist. Mit Klicken auf Symbol  SQL-Statement wird ein SQL-Befehl erzeugt und in die Spalte *SQL-Statement* abgelegt.

Die benannte Spalte steht am Ende der Tabelle in der Basisansicht bzw. befindet diese sich auf dem Registerblatt *Select*.

Dieser SQL-Befehl kann nun bearbeitet werden.

Beispiel: Ermittlung des Volumens aus den Feldern LANG, BREIT, HOCH

Für die Ermittlung des Volumens werden die Felder LANG, BREIT, HOCH der Quelldatenbank benötigt. Dazu kommt ein Schlüsselfeld ID sowie ein Platzhalterfeld für das Volumen, hier nochmals das Feld LANG (siehe Abbildung 20).

Der Platzhalter ist nötig, da die gleiche Anzahl an Feldern in der Pipeline, als auch in der SQL-Abfrage vorliegen muss. Die Feldbenennung spielt jedoch keine Rolle.

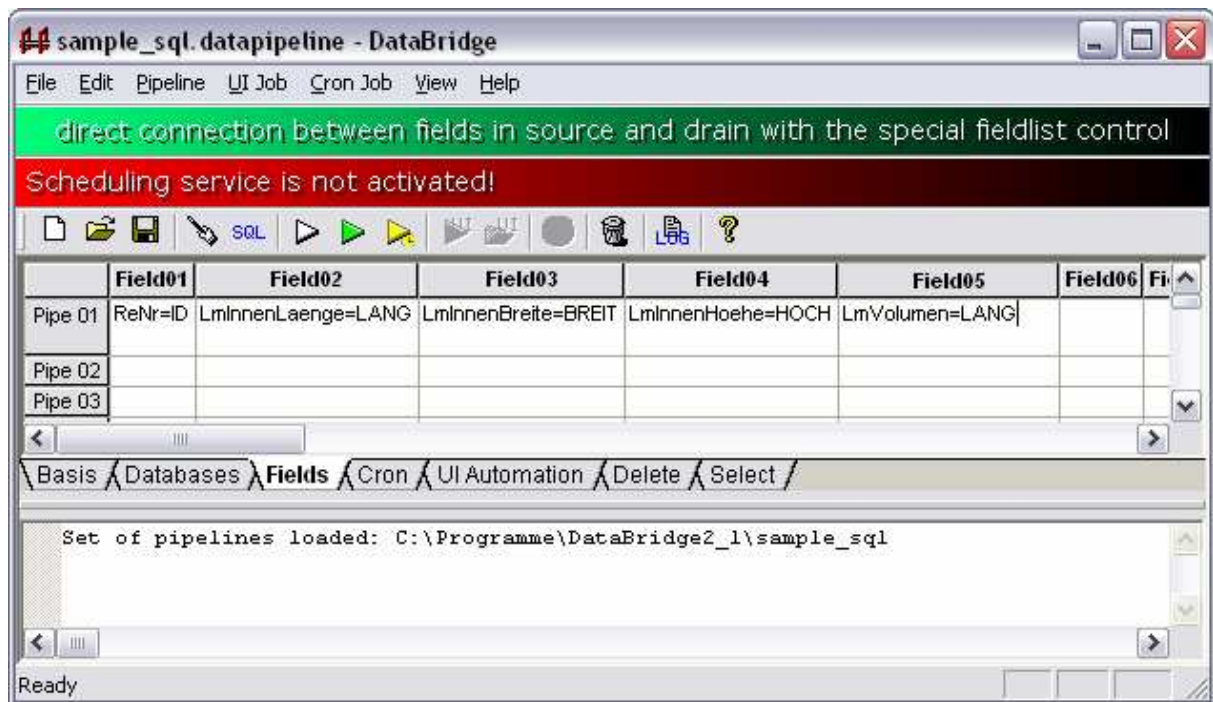



Abbildung 20: Datenbrücke für Berechnungen

Ist die Pipeline mittels Wizard erstellt, wird mit Klicken auf das Symbol  SQL-Statement der Select-Befehl generiert und in der Spalte SQL Select Statement angezeigt (siehe Abbildung 21).

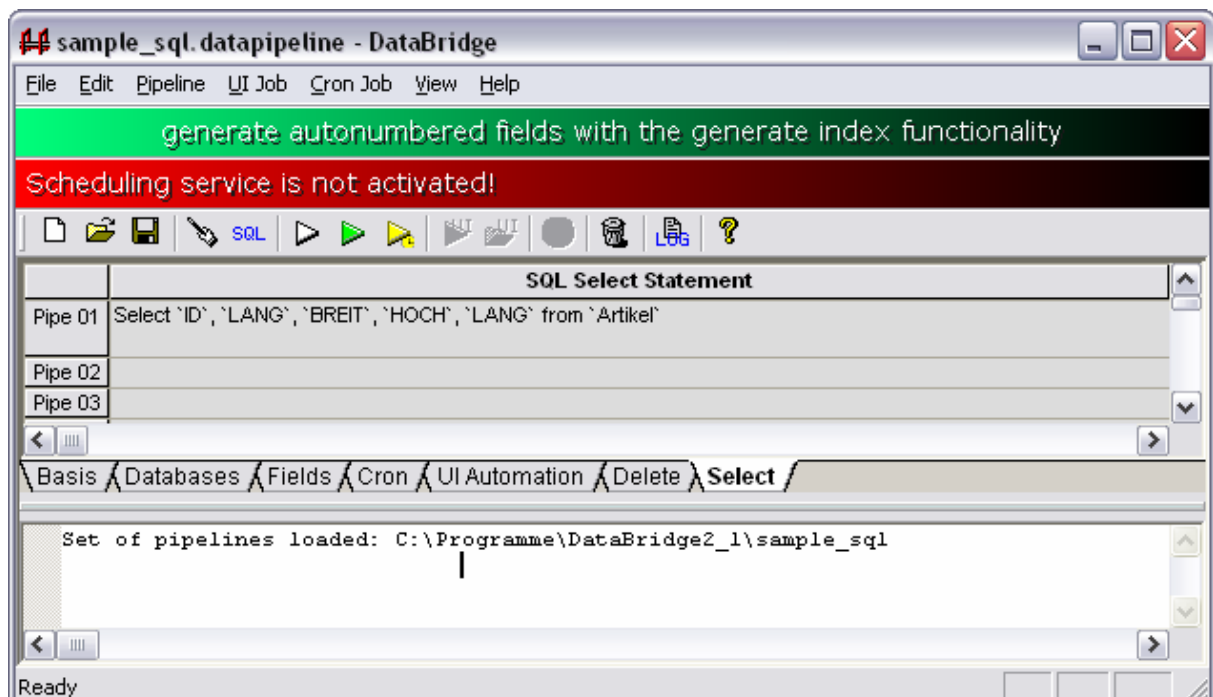


Abbildung 21: SQL-Statement

Nun wird in der Zeile das letzte Feld LANG ersetzt durch (siehe Abbildung 22):

LANG*BREIT*HOCH AS LmVolumen

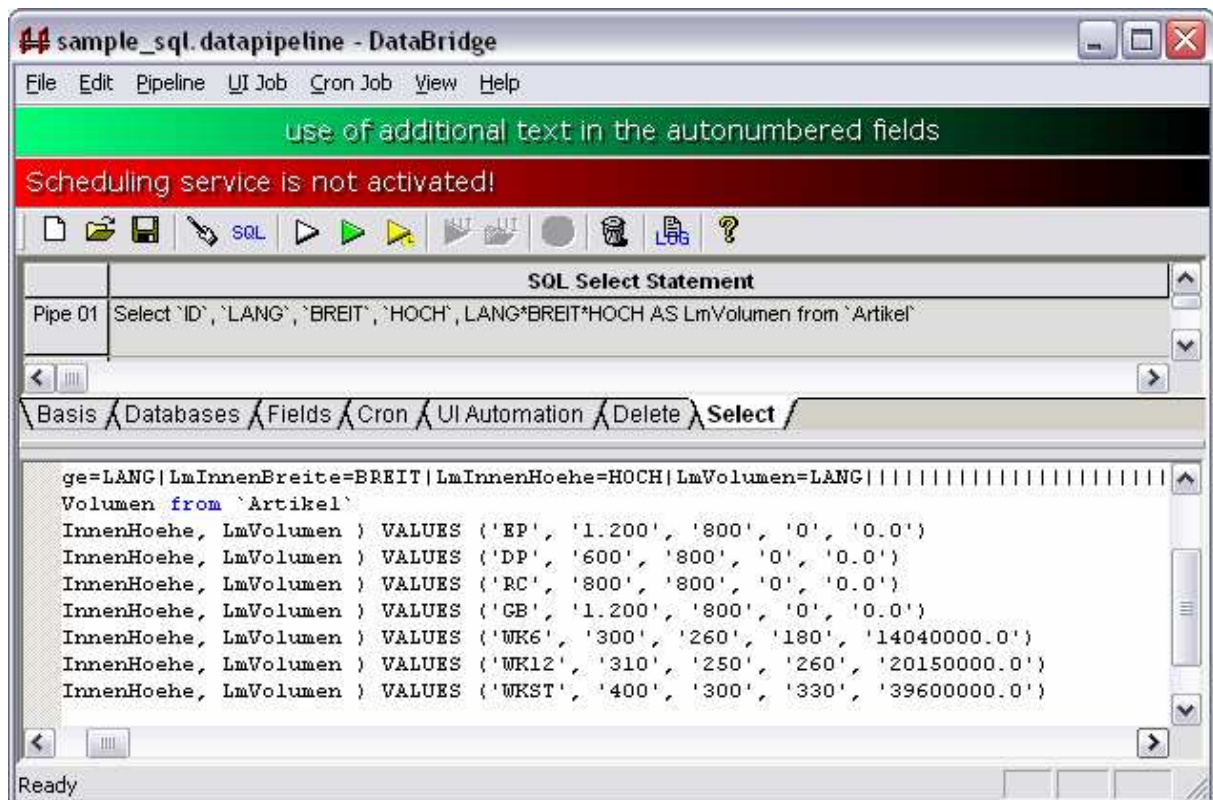


Abbildung 22: Resultat nach Ausführung der Pipeline

Abbildung 22 zeigt, dass das Volumen berechnet wurde. Die Formatierung des Ergebnisses ist von ODBC-Treiber abhängig, da die Ausführung des SQL-Befehls direkt in der Quell-Datenbank stattfindet.

Sobald bei der Ausführung ein SQL-Befehl in der Spalte SQL-Statement enthalten ist, wird dieser der ursprünglichen Datenbrücke vorgezogen und an deren Stelle ausgeführt. Getroffene Formatierungen der Spalten bezüglich Autonumber, Konstanten oder Umrechnungen werden dennoch berücksichtigt, auch wenn diese nicht im SQL-Befehl sichtbar sind. Dies ist der Grund, dass einmal erstellte Zuordnungen in der Reihenfolge und Anzahl nicht geändert werden dürfen (Felder als Platzhalter dienen).

Abbildung 23 zeigt die Pipeline zum beschriebenen SQL-Befehl aus Abbildung 22. Zusätzlich erhielten einige Felder neue Formate.

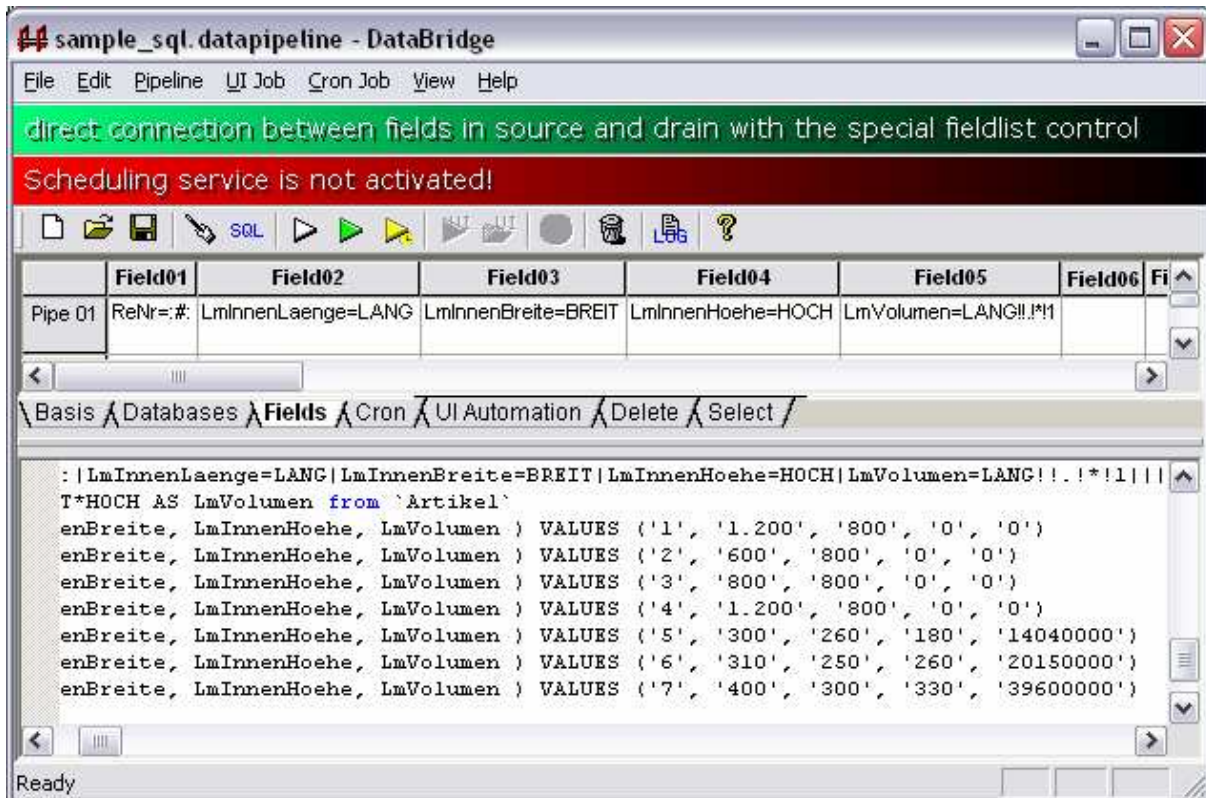


Abbildung 23: Resultat nach Formatierung

ReNr wird nun generiert. Das Format der Volumenberechnung wird mit der Formatierung zur Umrechnung ganzzahlig. Abbildung 23 zeigt im Ausgabefenster die erzeugten Daten.

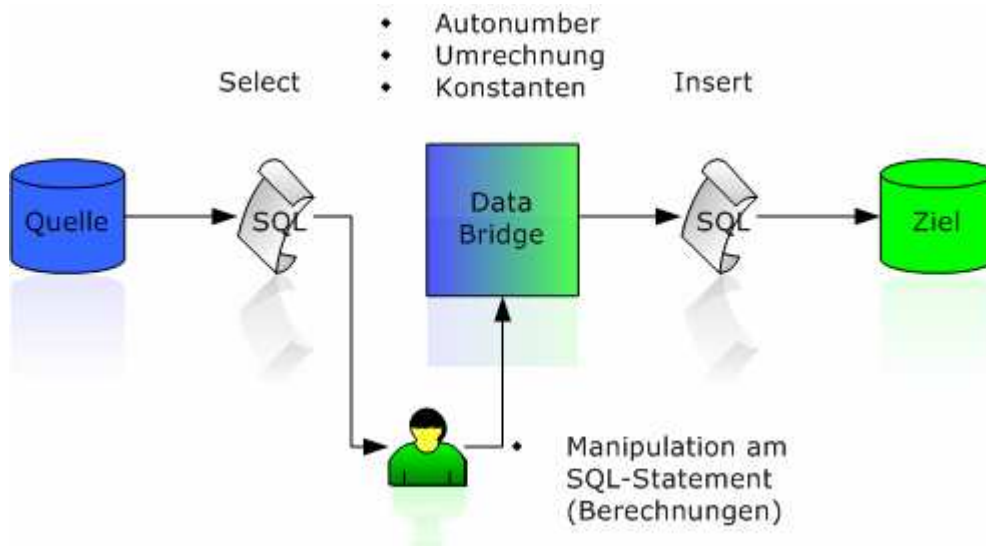


Abbildung 24: Datenmanipulation am SQL-Statement über den Benutzer

Die Bearbeitung des SQL-Statements der Quell-Datenbank (Select) lässt jede Möglichkeit zu, die dem SQL-Sprachumfang der Quell-Datenbank entspricht.

Die Anzahl der Spalten und deren Reihenfolge darf im generierten SQL-Befehl nicht geändert werden.

5.3 Formatierung

Die Funktionalität der Berechnung kann als Formatierungsvorschrift genutzt werden. Durch Multiplikation mit 1 können ganzzahlige Werte mit der korrekten Rundung erhalten werden. Ebenso können mit Multiplikation von 1,00 zwei Kommastellen erzwungen werden.

5.4 Kombinationen mit Konstanten

Eine Kombination von Konstanten mit Feldinhalten, etwa ein vorangestelltes Währungszeichen, wird im SQL-Statement vorgenommen. Dabei ist nach Generierung des SQL-Statements vor bzw. nach dem Feld die Konstante in einfachem Hochkomma zu setzen. Verknüpft werden die Einträge mit dem Zeichen „&“ oder „+“ - abhängig von der Datenbank (Abbildung 25).

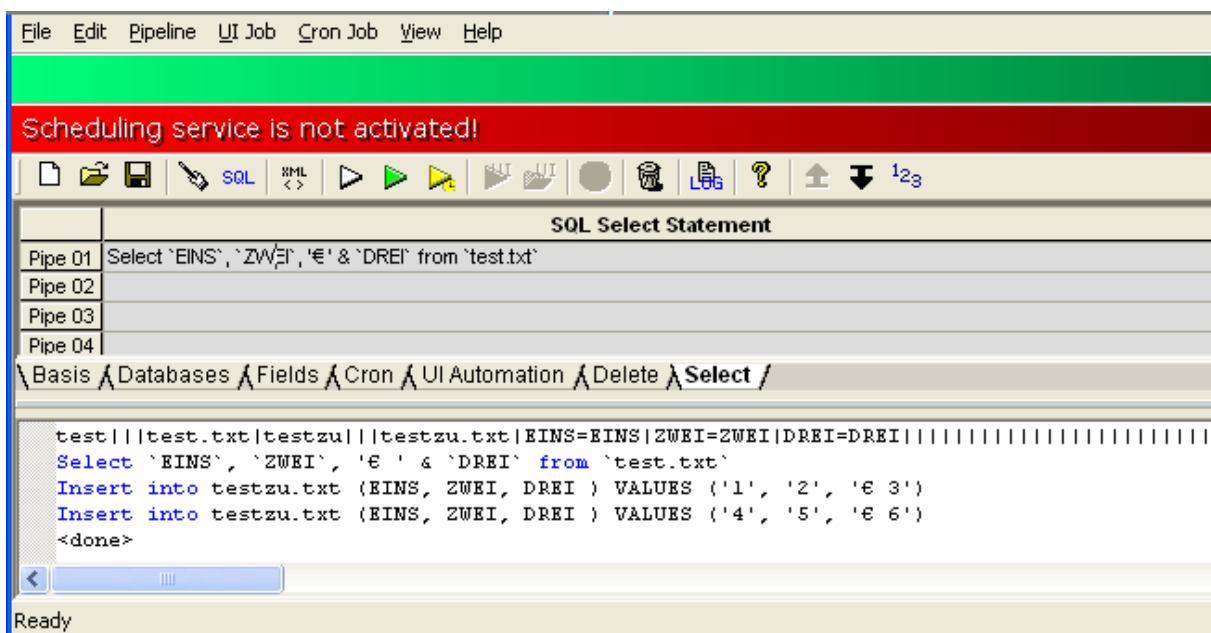



Abbildung 25: Kombination von Konstanten mit Feldinhalte

6 Automatisierung

6.1 Cron

Müssen Daten in regelmäßigen Abständen, z.B. stündlich, täglich oder wöchentlich in die Datenbank übertragen werden, wie bei einer Datensicherung, kann dies automatisiert erfolgen. Der Nutzer stellt den

Zeitpunkt der Ausführung in jeder Datenpipeline ein und startet mit dem Symbol  die Abarbeitung.

Die zeitliche Steuerung der Abarbeitung der Pipelines erfolgt in der Spalte *cron* bzw. auf dem gleichnamigen Registerblatt. Dabei ist ein Zeitpunkt bis zu einer Minute genau bestimmbar. Ebenso sind Zeitintervalle möglich.

In der Spalte *cron* wird der Eintrag vorgenommen, wobei der Notation aus Abbildung 26 zu folgen ist. Hierbei sind folgende Angaben möglich.

| Notation | Beschreibung |
|-----------|-------------------------------|
| * * * * * | Auszuführende Befehl |
| _____ | Wochentag (0 - 6) (Sonntag=0) |
| _____ | Monat (1 - 12) |
| _____ | Tag (1 - 31) |
| _____ | Stunde (0 - 23) |
| _____ | Minute (0 - 59) |

Abbildung 26: Notation und Bedeutung für cron

Beispiele für Einträge zu Zeitangaben sind:

| Zeitangabe | Beschreibung |
|----------------------|---|
| 0 12,15 * * * | Ein Komma „,” lässt mehrere Zeitangaben zu. Jeden Tag um 12 und 15 Uhr erfolgt eine Abfrage. |
| 0 12-24 * * * | Ein Bindestrich „-“ gibt einen Zeitraum an. Jeden Tag von 12-24 Uhr (jede Stunde) wird eine Abfrage durchgeführt. |

Abbildung 27: Zeitangaben

Abbildung 28 enthält einen *cron*-Eintrag. Die automatische Abarbeitung ist eingestellt, welches mit dem Häkchen gewirkt wird. Jeden Tag um 12 und 15 Uhr erfolgt eine Abfrage.

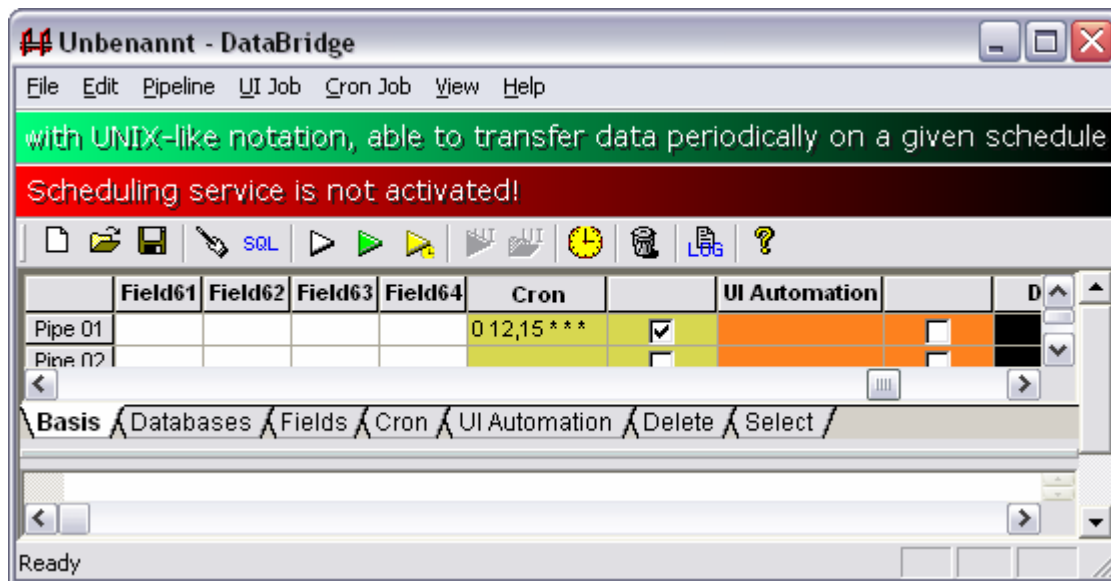



Abbildung 28: Automatische und zeitgesteuerte Ausführung mittels cron

Befindet sich der Cursor in einer Zelle der Spalte cron, aktiviert sich das Symbol , „Cron Edit“. Mit Hilfe des Dialogs „Cron Editor“ kann der Benutzer sich cron-Einträge erstellen bzw. werden die erstellten Einträge im Cron Editor-Fenster angezeigt.

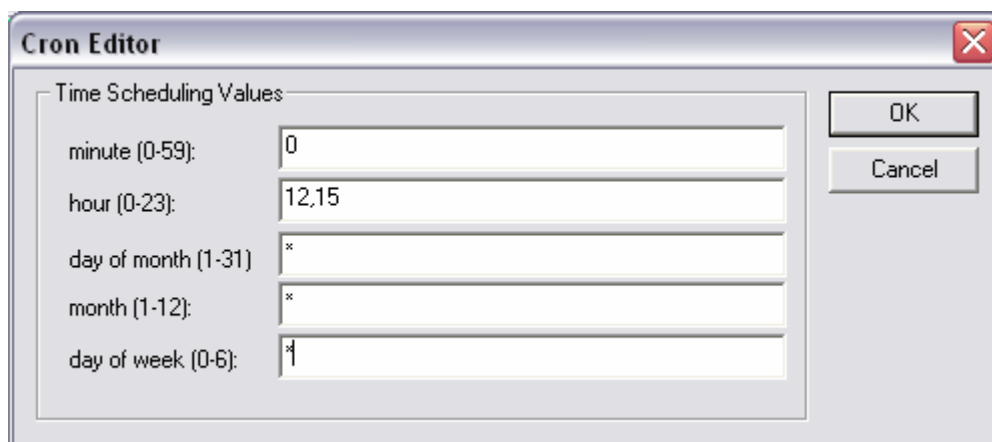



Abbildung 29: Dialog Cron Editor

Der Cron-Editor (Abbildung 29) enthält die Angaben aus dem Cron-Eintrag aus Abbildung 28.

Die automatische zeitgesteuerte Abarbeitung wird mit dem Symbol  *Execute Cron Job* aktiviert und läuft im Hintergrund. Den Status des Programms zeigt die von rot auf grün umschaltende Toolleiste mit der Aufschrift „Scheduling service is activated“ an. Durch nochmalige Betätigung wird die Abarbeitung gestoppt.

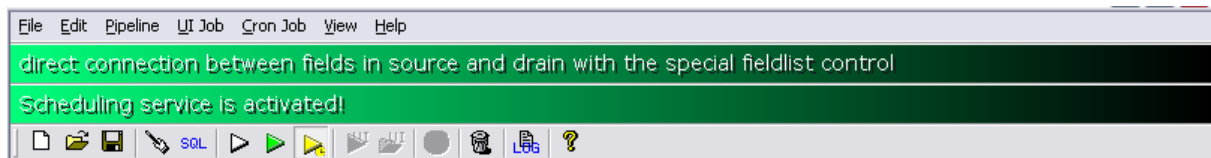



Abbildung 30: Tooleiste signalisiert cron-Aktivierung

Sind die eingestellten Zeitpunkte der Cron-Einträge erreicht, werden die SQL-Anweisungen der Pipeline erzeugt und ausgeführt und in die Logdatei aufgenommen.

6.2 Ausführen der Pipeline über Serverdienst

Data Bridge besitzt einen integrierten Server. Über einen Clienten kann die Nummer der Pipeline an Data Bridge übermittelt werden, die als nächstes ausgeführt werden soll.

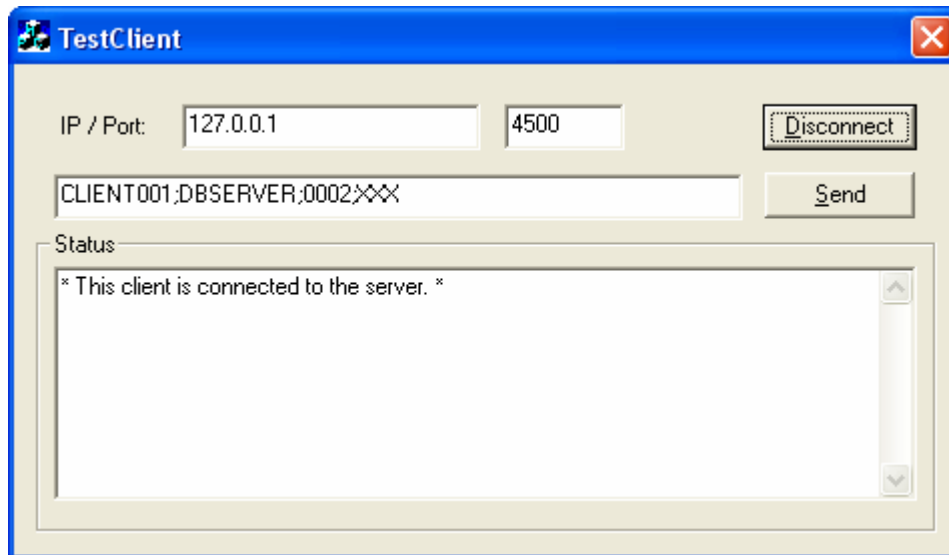
Der Serverdienst wird gestartet, wenn der Button  „Execute Cron Job“ betätigt wird.

Nachdem der Serverdienst gestartet ist, kann sich ein Client verbinden. Die Verbindungsdaten und Struktur der Mitteilung ist dem nachfolgenden Beispiel zu entnehmen.

Der Serverdienst wird beendet, wenn der Button „Execute Cron Job“ ein weiteres Mal betätigt wird.

Beispiel:

| | |
|-------------|--|
| IP-Adresse: | 127.0.0.1 (IP-Adresse des Systems, auf dem Data Bridge läuft.) |
| Port: | 4500 (standardmäßig) |
| Mitteilung: | CLIENT001;DBSERVER;0002;XXX |



Der Client versendet die Mitteilung, in der enthalten ist, dass die zweite Pipeline ausgeführt werden soll.

Wir möchten uns bei Dr. Sebastian Horbach und Dr. Jörg Ackermann von der Technischen Universität Chemnitz, Professur Fabrikplanung und Fabrikbetrieb, Institut für Betriebswissenschaften und Fabriksysteme (IBF) für die vielen kritischen Hinweise bedanken, die beim Einsatz des Programms innerhalb einer simulierten Praxisanwendung wesentlich zur Verbesserung und Ausweitung des Leistungsumfangs beigetragen haben. Wesentliche Denkanstöße flossen in das Programm ein und förderten die Zunahme der Einsatzfähigkeit auf größere Gebiete.

Lichtentanne, im März 2010