

# KNIME Leitfaden für die Erweiterung

KNIME AG, Zürich, Schweiz

Version 5.7 (letzte Aktualisierung auf )



## Inhaltsverzeichnis

- [Einleitung . . . . .](#page3)
- [Hafenarten . . . . .](#page3)
- [Verbindung mit einer Datenbank . . . . .](#page6)
- [Verbinden mit vordefinierten DB . . . . .](#page6)
- [Verbindung mit anderen Datenbanken . . . . .](#page10)
- [Registrieren Sie Ihre eigene DB . . . . .](#page12)
- [Depridierte JDBC Driver #page . . . . .](#page17)
- [Erweiterte Datenbankoptionen . . . . .](#page18)
- [Beispiele . . . . .](#page23)
- [Lesen aus einer Datenbank . . . . .](#page43)
- [Datenbank Metadaten . . . . .](#page44)
- [Quere Erzeugung . . . . .](#page45)
- [Visual Query Generation . . . . .](#page45)
- [Advanced Query Building . . . . .](#page51)
- [Datenbankstruktur Manipulation . . . . .](#page54)
- [DB Tabelle Remove . . . . .](#page54)
- [DB Tabelle Create . . . . .](#page55)
- [DB Manipulation . . . . .](#page59)
- [DB Löschen . . . . .](#page59)
- [DB Writer . . . . .](#page62)
- [DB Insert . . . . .](#page63)
- [DB Update . . . . .](#page63)
- [DB Merge . . . . .](#page63)
- [DB Loader . . . . .](#page63)
- [DB Transaction Notes . . . . .](#page66)
- [Art Mapping. . . . .](#page68)
- [DB Type Mapper . . . . .](#page69)
- [Migration. . . . .](#page70)
- [Workflow Migration . . . . .](#page70)
- [Node Name Mapping . . . . .](#page74)
- [Registrieren Sie Ihre eigene DB . . . . .](#page77)
- [Business Hub / Server Setup . . . . .](#page78)
- [JDBC Treiber auf GitHub . . . . .](#page78)
- [Standard JDBC Parameter . . . . .](#page80)

< a href="#page82" style="color: #000000; text-decoration: underline;">> Reserviert JDBC Parameter für den URL-String  
< a href="#page83" style="color: #000000; text-decoration: underline;">> Verbindung Initialisierung "URL-String"  
< a href="#page84" style="color: #000000; text-decoration: underline;">> . . . . . . . . . . . . . . .  
< a href="#page85" style="color: #000000; text-decoration: underline;">> Beispiel: Apache HttpClient "#page85"  
< a href="#page87" style="color: #000000; text-decoration: underline;">> Beispiel: Apache HttpClient "#page87"  
< a href="#page88" style="color: #000000; text-decoration: underline;">> Beispiel: Microsoft SQL Server "#page88"  
< a href="#page90" style="color: #000000; text-decoration: underline;">> Beispiel: Oracle Database "#page90"  
< a href="#page90" style="color: #000000; text-decoration: underline;">> Beispiel: PostgreSQL "#page90"

# Einleitung

Die KNIME Datenbankerweiterung bietet eine Reihe von KNIME-Knoten, die die Verbindung zu JDBC-konforme Datenbanken. Diese Knoten liegen in der DB Kategorie im Node Repository, wo Sie eine Reihe von Datenbankzugriffen, Manipulationen und Schreibknoten finden können.

Die Datenbankknoten sind Teil jeder KNIME Analytics Platform Installation. Es ist nicht notwendig, um zusätzliche KNIME Extensions zu installieren.

Dieser Leitfaden beschreibt die KNIME Datenbankerweiterung und zeigt unter anderem, wie man verbinden Sie mit einer Datenbank und wie Sie Datenmanipulation in der Datenbank durchführen können.

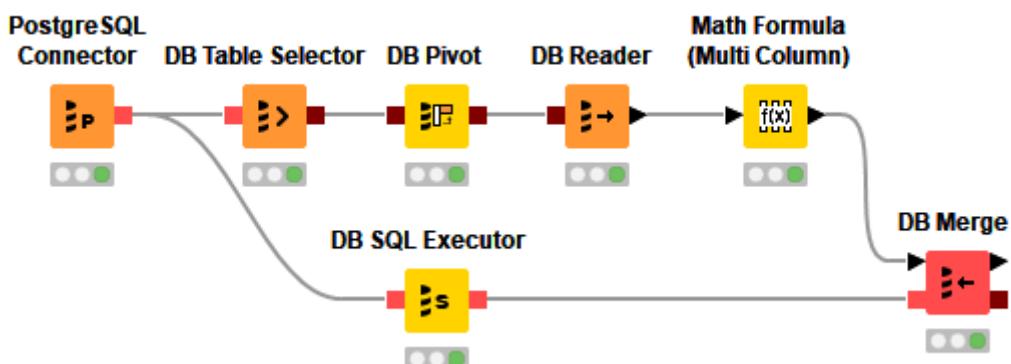


Abbildung 1. Beispiel-Workflow mit DB-Knoten

## Hafentypen

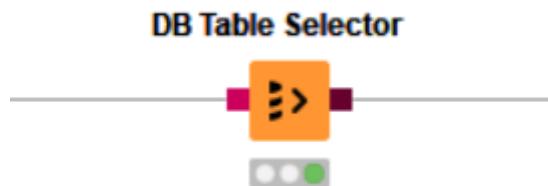


Abbildung 2. Zwei Arten von Datenbank-Port

Es gibt zwei Arten von Ports in der Datenbank-Erweiterung, der DB Connection Port (rot) und die DB Data port (dunkelrot).

Die DB-Verbindung port speichert Informationen über die aktuelle DB Session, z.B. Datentypen, Anschlüsseigenschaften, JDBC-Eigenschaften, Treiberinformationen, etc.

Die DB Daten port gibt Ihnen Zugriff auf eine Vorschau der Daten.

## Blick auf den Hafen

Nachdem Sie einen DB-Knoten ausgeführt haben, können Sie das Ergebnis in der Ausportansicht durch Rechtsklick auf den node and select the outport to inspect at the bottom of the menu. Für weitere Informationen

[über die Ausführung eines Knotens, bitte verweisen Sie auf die](#)

[Quickstart Anleitung](#).

## DB Verbindungsaukskunft

Die Aussicht auf einen DB-Anschluss hat die

Deutsche SitzungRegisterkarte, die die Informationen enthält

über die aktuelle Datenbanksitzung, wie Datenbanktyp und Verbindungs-URL.

## DB Datenauskunft

Beim Ausführen eines Datenbank-Manipulationsknotens, der einen

DB Daten z.B. einen

DB

Gruppe Knoten, was der Knoten tut, ist die notwendige SQL-Abfrage zu erstellen, um die

Gruppe Durch den Betrieb, der vom Benutzer ausgewählt wird und an den nächsten Knoten im Workflow weiterleitet. Es die Abfrage nicht tatsächlich durchgeführt. Es ist jedoch möglich, eine Vorschau einer Untermenge zu prüfen des Zwischenergebnisses und seiner Spezifikation.

Wählen Sie dazu den Knoten aus und klicken Sie auf Fetch 100 Tabellenzeilen

im Knotenmonitor am unteren Ende

die UI.

Standardmäßig werden nur die ersten 100 Zeilen geätzt, aber Sie können auch andere auswählen

Optionen durch Öffnen des Dropdown-Menüs der GebührenKnopf. Hoewever, sei

bewusst, dass je nach Komplexität der SQL-Abfrage bereits nur Cache

die ersten 100 Zeilen könnten eine lange Zeit dauern.

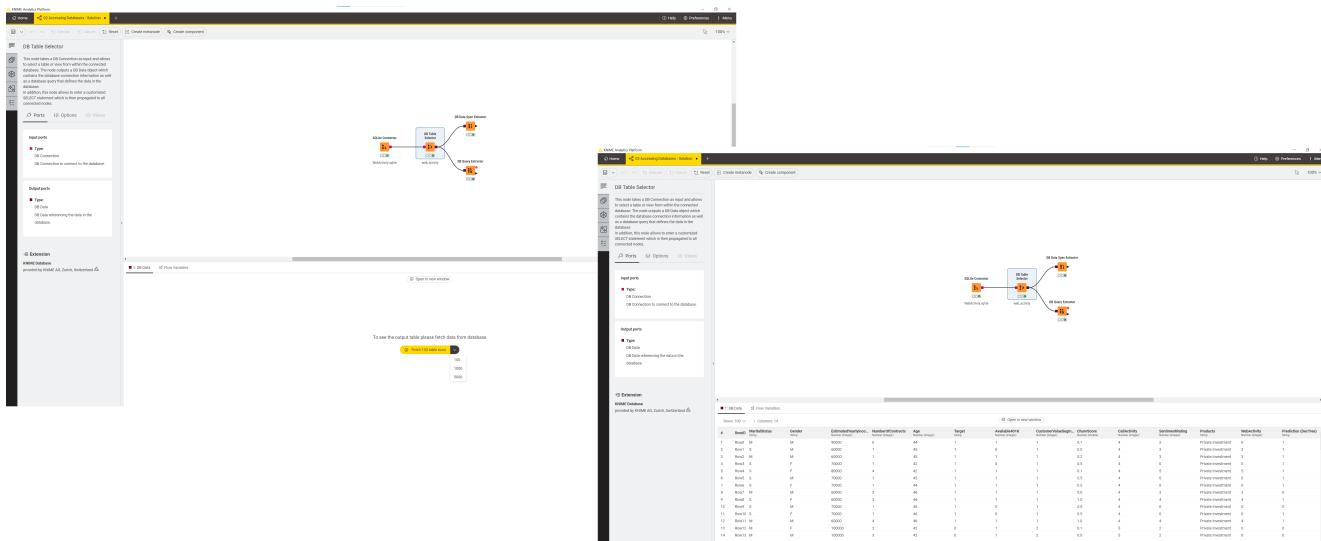


Abbildung 3. DB Outport Ansicht mit abgerufenen Zeilen

Die Tabellenspezifikation kann mit Hilfe der

[DB Data Spec Extractor](#)

Knotenpunkt Die

Ausgabe dieses Knotens zeigt die Liste der Spalten in der Tabelle, mit ihren Datenbanktypen und die entsprechenden KNIME-Datentypen.



Weitere Informationen zum Typ Mapping zwischen Datenbanktypen und KNIME  
[Abschnitt](#page68)

Die generierte SQL-Abfrage kann mithilfe der

[DB Query Extractor](#)

Knoten.

## Sitzungsabwicklung

Der DB Session Lebenszyklus wird von den Connector-Knoten verwaltet. Ausführung eines Connector-Knotens wird eine DB Session erstellen, und das Zurücksetzen des Knotens oder das Schließen des Workflows wird die entsprechende DB Session und damit die Verbindung zur Datenbank.

Um eine DB Session während der Workflow-Ausführung zu schließen [DB Anschluss schließen](#) Node kann verwendet. Dies ist auch der bevorzugte Weg, Datenbankressourcen zu befreien, sobald sie nicht länger benötigt durch den Workflow. Um eine DB Session zu schließen, verbinden Sie sie einfach mit der DB Verbindungsschließer-Knoten, der die DB Session zerstört und damit die Verbindung zum [Datenbank, sobald sie ausgeführt wird. Verwenden Sie die](#) [Eingangsstrom variabler Port](#) der DB-Verbindung Schließen Sie den Knoten, um ihn auszuführen, sobald es gespeichert ist, um die DB Session zu zerstören.

# Verbindung mit einer Datenbank

Die DB → Verbindung Unterkategorie im Node Repository enthält

- eine Reihe von datenbankspezifischen Verbindungsknoten für häufig verwendete Datenbanken wie Microsoft SQL Server, MySQL, PostgreSQL, H2, etc.
- sowie die generische Datenbankverbinder Knoten.

Ein Connector-Knoten erzeugt eine Verbindung zu einer Datenbank über seinen JDBC-Treiber. In der Konfiguration Dialog eines Verbindungsknotens, den Sie benötigen, um Informationen wie den Datenbanktyp, den Treiber, der Standort der Datenbank und die Authentifizierungsmethode, falls vorhanden.

Die meisten der datenbankspezifischen Verbindungsknoten enthalten bereits die notwendigen JDBC Treiber und bieten einen Konfigurationsdialog an, der auf das spezifische Datenbank. Es wird empfohlen, diese Knoten über die generische Knoten, wenn möglich.

DB Connector

## Anbindung an vordefinierte Datenbanken

Folgende Datenbanken haben einen eigenen Connector-Knoten:

- Amazon Redshift
- Amazon Athena
- Google BigQuery
- H2
- Microsoft Access
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL
- Schneeflocke
- SQLite
- Vertica

Einige dedizierte Connector-Knoten wie Google BigQuery oder Amazon

Redshift, kommen ohne einen JDBC-Treiber wegen Lizenzbeschränkung. Wenn Sie wollen

Verwenden Sie diese Knoten, müssen Sie zunächst den entsprechenden JDBC-Treiber registrieren.

[Treiber Plug-in Abschnitt](#page12)

Abschnitt über die Registrierung

[Treiber Plug-in Abschnitt](#page9)

Ihr eigener Fahrer. Für Amazon Redshift wenden Sie sich bitte an die

[Treiber Plug-in Abschnitt](#page9).

Wenn für Ihre Datenbank kein dedizierter Steckverbinderknoten vorhanden ist, können Sie die generische

[DB](#page10)

Anschluss Knoten. Weitere Informationen hierzu finden Sie in der

[Datenbanken](#page10)

Abschnitt.

Nachdem Sie den richtigen Connector-Knoten für Ihre Datenbank finden, doppelklicken Sie auf den Knoten, um zu öffnen

den Konfigurationsdialog. In der Verbindungseinstellungen Fenster können Sie die grundlegende

Parameter für Ihre Datenbank, wie z.B. Datenbanktreiber, Standort oder Authentifizierung. Dann

klicken Ok. und den Knoten ausführen, um eine Verbindung herzustellen.

Wenn Sie bei der Ausführung "die neueste Treiberversion verwenden" auswählen, wird der Knoten

automatisch den Treiber mit der neuesten (höchsten) Treiberversion verwenden, die

für den aktuellen Datenbanktyp verfügbar. Dies hat den Vorteil, dass Sie nicht

muss den Workflow nach einem Treiber-Update berühren. Allerdings könnte der Workflow

brechen im seltenen Fall, dass das Verhalten des Fahrers, z.B. Typ Mapping,

Änderungen mit der neueren Version.

KNIME Analytics Platform bietet im Allgemeinen drei verschiedene Arten von Verbindungsknoten die

[Dateibasiert](#page7), [Datenbank](#page8) und [Treiber](#page9)

[Knoten](#page10)

die in den folgenden Abschnitten erläutert werden.

## Dateibasierter Connector-Knoten

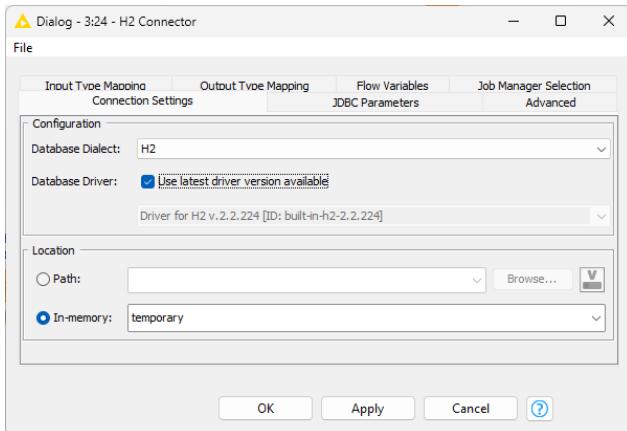


Abbildung 4. H2 Konnektorkonfiguration Dialog

Die Figur auf der linken Seite zeigt eine Beispiel des Knoten-Dialogs für eine Datei-basierte Datenbank wie SQLite, H2 oder MS Zugang. Die wichtigsten Knoteneinstellungen werden nachfolgend beschrieben:

**Konfiguration** : Im Konfigurationsfenster Sie können die registrierte Datenbank auswählen Dialekt und Fahrer.

**Standort** : Der Standort zur Datenbank. Du kann entweder die Pfad zu einem bestehenden Datenbank oder Auswahl In-Memory um ein temporäre Datenbank, die im Speicher gespeichert wird wenn die Datenbank diese Funktion unterstützt.

## Serverbasierter Verbindungsnode

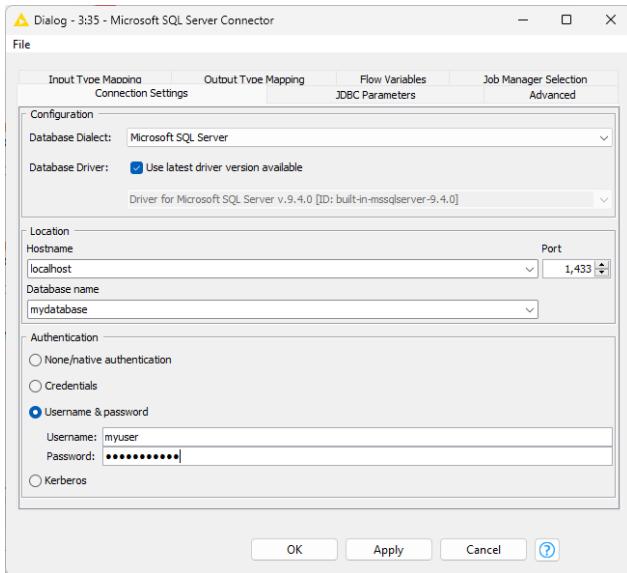


Abbildung 5. MS SQL Server Connector

Konfigurationsdialog

Die Figur auf der linken Seite zeigt eine Beispiel des Knotendialogs für einen Server-Datenbanken, wie MySQL, Oracle oder PostgreSQL. Der wichtigste Knoten im Folgenden werden Einstellungen beschrieben.

**Konfiguration** : Im Konfigurationsfenster Sie können die registrierte Datenbank auswählen Dialekt und Fahrer.

**Standort** : Der Standort zur Datenbank. Du sollte den Hostnamen und den Port bereitstellen der Maschine, die die Datenbank beherbergt, und auch den Namen der Datenbank, die je nach Datenbank optional sein.

**Authentifizierung** : Anmeldeinformationen können entweder über Anmeldestromgrößen bereitgestellt werden, oder direkt im Konfigurationsdialog im Form des Benutzernamens und Passworts. Kerberos Eine Authentifizierung ist auch für Datenbanken, die diese Funktion unterstützen, z. Hive oder Impala. Für weitere Informationen Kerberos Authentifizierung, bitte beziehen sich auf die [Kerberos Benutzerhandbuch](#).

Weitere Informationen zu den JDBC-Parametern und dem Erweiterten Tab finden Sie unter [Abschnitt "Typ Mapping Tabs und Advanced Tab"](#). Die Typ Mapping Tabs Abschnitt. Die JDBC-Parameter und Advanced Tab

## Drittanbieter-Datenbanktreiber Plug-in

Wie bereits erwähnt, enthalten die dedizierten datenbankspezifischen Steckverbinderknoten bereits die notwendigen JDBC-Treiber. Einige Datenbanken erfordern jedoch eine besondere Lizenzierung, die verhindert, dass wir die erforderlichen JDBC-Treiber automatisch installieren oder gar Bündeln mit die entsprechenden Anschlussknoten. Zum Beispiel bietet KNIME zusätzliche Plug-ins an, die installieren Sie die [Oracle Datenbank Treiber](#) und [Microsoft SQL Server Treiber](#) oder [Amazon Redshift Fahrer](#), die besondere Lizenzen benötigen.

Um die Plug-ins zu installieren, gehen Sie zu Datei → KNIME installieren Erweiterungen... . In der Installieren Fenster, Suche

für den Treiber, den Sie benötigen (Oracle, MS SQL Server oder Redshift), und Sie werden etwas sehen ähnlich wie die Abbildung unten. Wählen Sie dann das Plug-in, um es zu installieren. Wenn Sie das Plug-in nicht sehen dieses Fenster dann ist es bereits installiert. Nach der Installation des Plug-ins starten Sie KNIME neu. Danach, Wenn Sie den Konfigurationsdialog des dedizierten Verbindungsknotens öffnen, sollten Sie sehen dass der installierte Treiber der jeweiligen Datenbank in der Datenbanktreiberliste verfügbar ist.

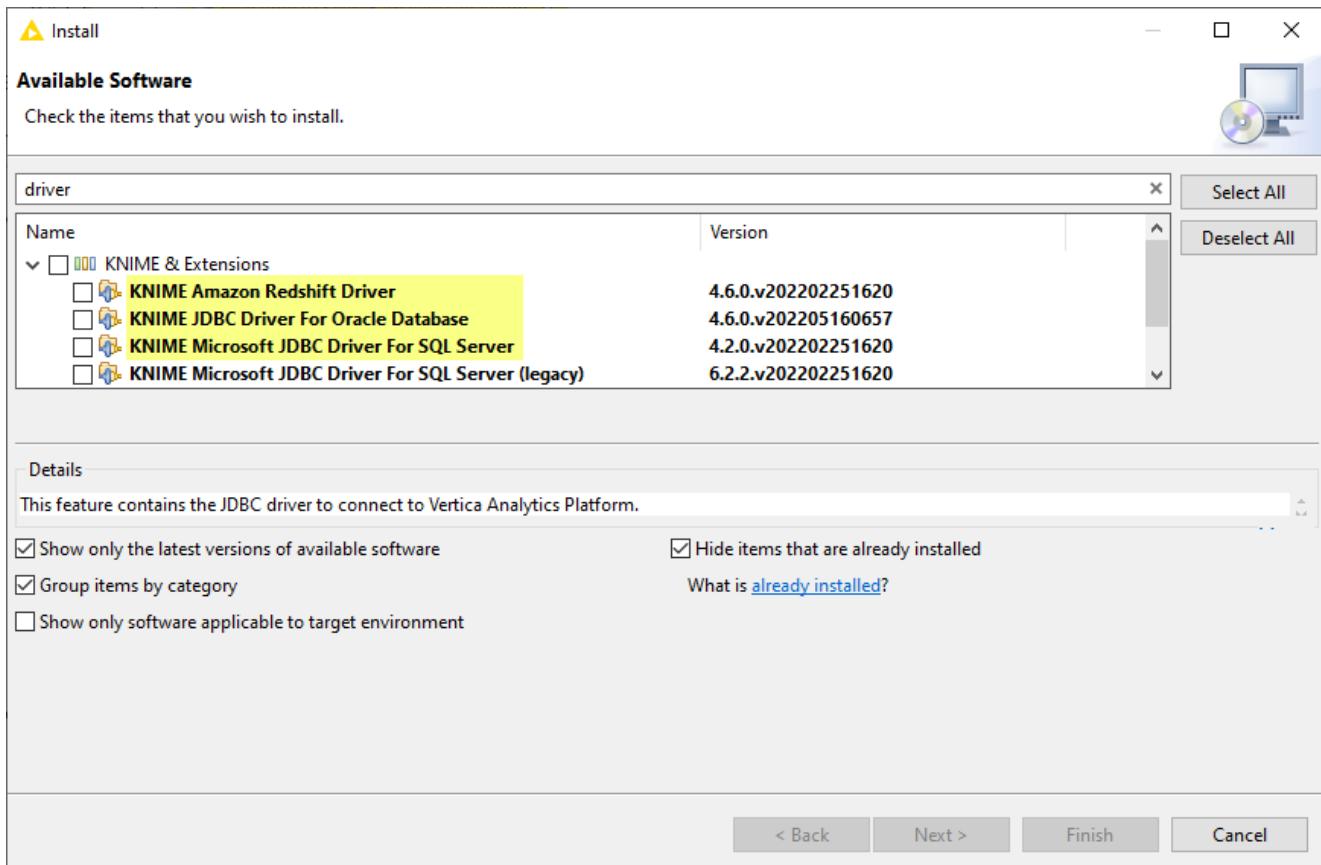


Abbildung 6. Fenster installieren

## Verbindung mit anderen Datenbanken

Die generische DB Connector node kann mit beliebigen JDBC-konformen Datenbanken verbinden. Die Im Folgenden werden die wichtigsten Knoteneinstellungen beschrieben.

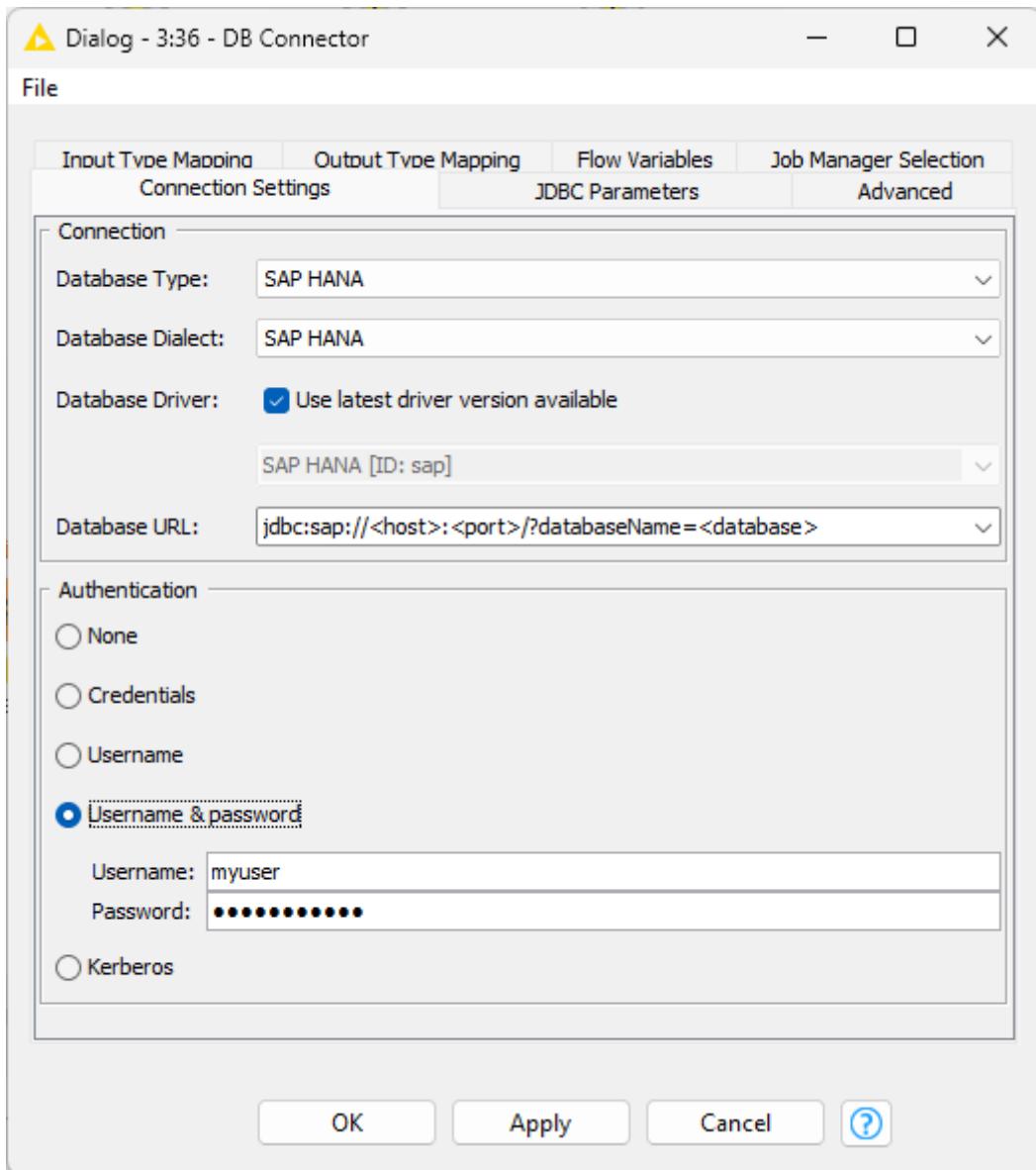


Abbildung 7. Konfigurationsdialog für Datenbankverbindung

Datenbanktyp : Wählen Sie den Typ der Datenbank aus, an die sich der Knoten anschließt. Zum Beispiel, wenn Datenbank ist eine PostgreSQL-Derivat auswählen Postgres als Datenbanktyp. Wenn Sie nicht die Typ wählen Sie den Standardtyp aus.

Datenbank Dialect : Wählen Sie den Datenbankdialekt aus, der definiert, wie die SQL-Anweisungen sind erzeugt.

Datenbank Treiber : Wählen Sie einen geeigneten Treiber für Ihre spezifische Datenbank aus oder wählen Sie "Benutzen neueste Treiberversion verfügbar" Option, in welchem Fall bei der Ausführung der Knoten wird automatisch den Treiber mit der neuesten (höchsten) Treiberversion verwenden, die für die aktuelle Datenbanktyp. Wenn es keinen passenden JDBC-Treiber gibt, muss er zuerst registriert werden, siehe [Registrieren Sie Ihre eigenen JDBC-Treiber](#page12). Nur Treiber, die für die ausgewählten Datenbanktyp wird zur Auswahl zur Verfügung stehen.

Datenbank URL : Eine treiberspezifische JDBC URL. Geben Sie die Datenbankinformationen im Platzhalter ein,

wie Host, Port und Datenbankname.

**Authentifizierung** : Login-Anmeldeinformationen können entweder über Anmeldeflussgrößen bereitgestellt werden, oder direkt im Konfigurationsdialog in Form von Benutzername und Passwort. Kerberos

Eine Authentifizierung ist auch für Datenbanken vorgesehen, die diese Funktion unterstützen, z.B. Hive oder Impala.

Weitere Informationen zur Kerberos-Authentifizierung finden Sie in der

[Kerberos Benutzerhandbuch](#)

Der ausgewählte Datenbanktyp und Dialekt bestimmen, welche Datentypen, Aussagen wie Insert-, Update- und Aggregationsfunktionen werden unterstützt.

Wenn Sie einen Fehler beim Verbinden mit einer Drittanbieter-Datenbank auftreffen, können Sie die JDBC logger

in das KNIME-Log, das Ihnen helfen könnte, die Probleme zu identifizieren. Um zu tweak wie

KNIME interagiert mit Ihrer Datenbank, z.B. zitiert Kennungen, die Sie den Standard ändern können entsprechend den Einstellungen Ihrer Datenbank. Zum Beispiel

KNIME verwendet " als Standardkennung, die nicht standardmäßig von einigen unterstützt wird

Datenbanken (z.B. Informix). Um dies zu lösen, ändern Sie einfach den Wert des Abgrenzer

Registrieren Sie Ihre eigenen JDBC Treiber

Für einige Datenbanken KNIME Analytics Platform enthält keinen gebrauchsfertigen JDBC Treiber.

In diesen Fällen muss zunächst ein herstellerspezifischer JDBC-Treiber in KNIME registriert werden.

Analyseplattform. Bitte konsultieren Sie Ihren Datenbankanbieter, um den JDBC-Treiber zu erhalten. Eine Liste einiger der beliebtesten JDBC Treiber finden

Der JDBC-Treiber muss [JDBC 4.1](#) oder höher konform.

Um JDBC-Treiber auf KNIME Server einzurichten, lesen Sie bitte den Abschnitt

[Hub und KNIME Server](#)

Um Ihren herstellerspezifischen JDBC Treiber zu registrieren, gehen Sie zu Datei → Vorlieben → KNIME → Datenbanken .

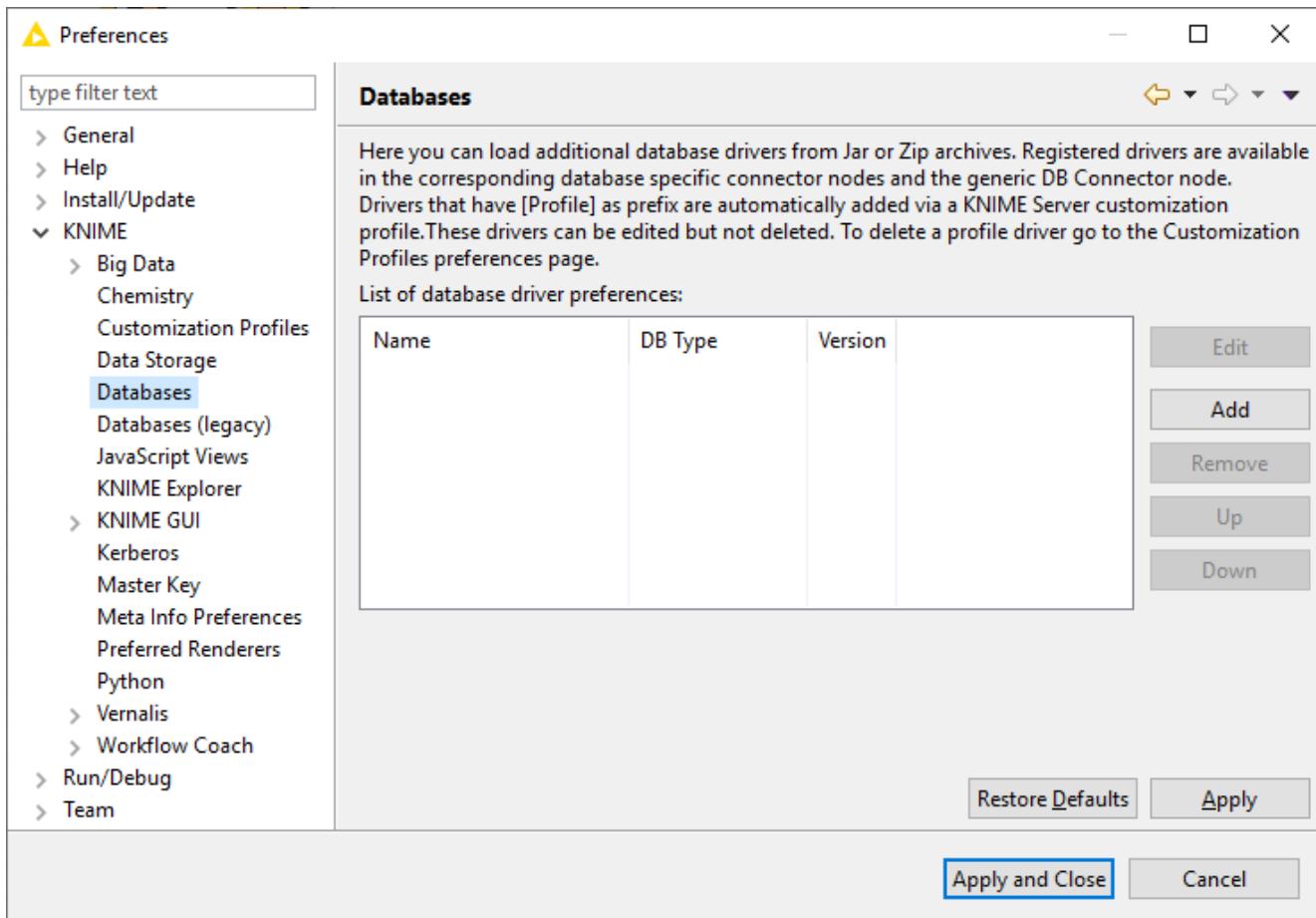


Abbildung 8. DB Seite

Klick [Hinzufügen](#) öffnet ein neues Datenbank-Treiberfenster, wo Sie den JDBC-Treiber bereitstellen können

Pfad und alle notwendigen Informationen, wie:

- ID : Die einzigartige ID des JDBC-Treivers, bestehend aus nur alphanumerischen Zeichen und underscore.

- Name : Der einzigartige Name des JDBC-Treibers.

- Datenbanktyp : Der Datenbanktyp. Wenn Sie einen bestimmten Datenbanktyp z. MySQL die

Der Treiber wird für die Auswahl im dedizierten Steckverbinderknoten z. MySQL

Connector. Wenn Ihre Datenbank jedoch nicht auf der Liste ist, können Sie wählen Ihnen alle verfügbaren Parameter in der

[Standard](#page20), die Fahrer, die sind

registriert für den Standardtyp sind nur im generischen DB Connector-Knoten verfügbar.

- Warenbezeichnung Optionale Beschreibung des JDBC-Treibers.

- URL-Vorlage : Das JDBC Treiber-Verbindungs-URL-Format, das im dedizierten

Verbindungsknoten. Wenn Sie eine andere Datenbank als

Standard in der Datenbanktyp , die

URL-Vorlage wird mit der Standardvorlage für die ausgewählte Datenbank vorgegeben. Bitte!

[Standardvorlage](#page16) oder [Standardvorlage](#page15)

für weitere Informationen.

- URL Vorlage Syntax Informationen : Klicken auf das Fragezeichen öffnet eine Infobox die im Allgemeinen Informationen über die URL-Template-Syntax liefert. Zusätzlich, wenn Sie wählen eine andere Datenbank Standard in der Datenbanktyp eine oder mehrere mögliche URL Vorlagenbeispiele werden für den ausgewählten Datenbanktyp bereitgestellt, den Sie kopieren können und in den URL-Vorlage Feld.

- Klasse : Der Weg zum JDBC Treiber. Klicken Sie auf Datei hinzufügen wenn der Fahrer als ein .jar-Datei oder Verzeichnis hinzufügen wenn der Treiber als Ordner vorgesehen ist, der mehrere enthält. Ja Dateien. Einige Anbieter bieten eine .zip-Datei zum Herunterladen, die auf eine entpackt werden muss Ordner zuerst.

Wenn der JDBC-Treiber native Bibliotheken z.B. DLLs benötigt, müssen Sie alle von ihnen setzen in einen einzigen Ordner und dann diesen Ordner über die Verzeichnis hinzufügen Knopf in zusätzlich zur JDBC Treiber .jar-Datei.

- Fahrerkasse : Die JDBC Treiberklasse und Version werden automatisch von Klicken Sie auf Finden Sie Fahrerklassen . Bitte wählen Sie nach dem Klick die entsprechende Klasse aus Knopf.

Wenn Ihre Datenbank in der Datenbanktyp Dropdown-Liste, es ist besser, Wählen Sie es statt es zu setzen Standard . Einstellung der Datenbanktyp bis Standard wird Sie können nur die generische DB Connector Verbindung zum Knoten Datenbank, auch wenn es einen dedizierten Connector-Knoten für diese Datenbank gibt.

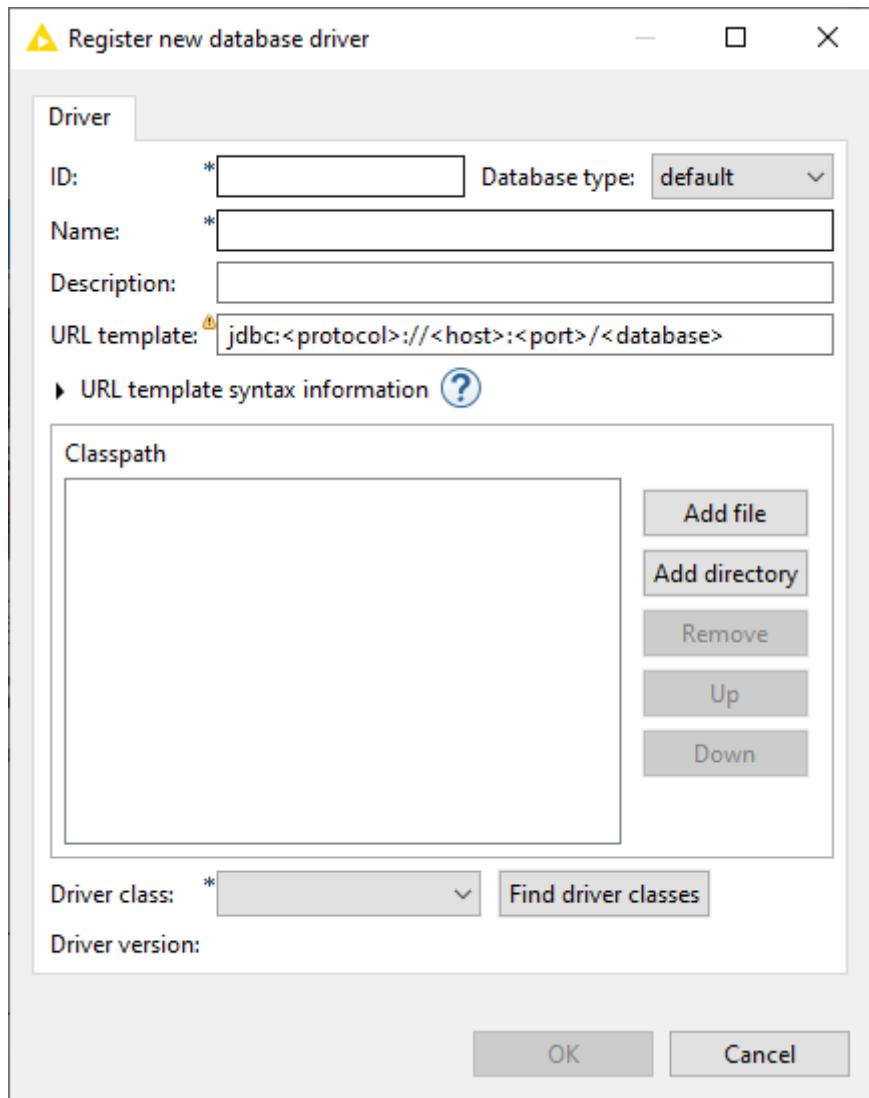


Abbildung 9. Registrieren Sie neues Datenbanktreiberfenster

KNIME Server kann JDBC Treiber automatisch an alle angeschlossenen KNIME verteilen  
[KNIME Datenbanktreiber](#)

## JDBC URL Vorlage

Bei der Registrierung eines JDBC-Treibers müssen Sie die JDBC-URL-Vorlage angeben, die verwendet durch den dedizierten Connector-Knoten, um die endgültige Datenbank-URL zu erstellen. Zum Beispiel `jdbc:oracle:thin:@:/` ist eine gültige Treiber-URL-Vorlage für die [Oracle dünner Fahrer](#). Für die meisten Datenbanken müssen Sie nicht selbst die passende URL-Vorlage finden, [URL Vorlage Syntax](#) weil mindestens ein URL-Template-Beispiel für eine Datenbank.

Die Werte der Variablen in der URL-Vorlage, z. Insgesamt, oder kann im Konfigurationsdialog des entsprechenden Verbindungsknotens angegeben.

Token:

- Pflichtwert (z. [Datenbank]): Der referenzierte Token muss einen Nicht-Blanch-Wert haben. Der Name zwischen den Klammern muss ein gültiger Tokenname sein (siehe unten für eine Liste von unterstützte Token).
- Optionaler Wert (z. [Datenbank]): Das referenzierte Zeichen kann einen Leerwert aufweisen. Die Name zwischen den Klammern muss ein gültiger Tokenname sein (siehe unten für eine Liste von unterstützte Token).
- Bedingungen (z. [Standort = in-Memory?mem:]): Dies gilt für Datei... Datenbanken, wie H2, oder SQLite. Die erste ? Charakter trennt den Zustand aus dem Inhalt, der nur in der URL enthalten ist, wenn die Bedingung wahr ist. Die einzige aktuell verfügbarer expliziter Betreiber =, um den genauen Wert einer Variablen zu testen. Links Operand muss ein gültiger Variablenname sein, und der richtige Operand der Wert, den die Variable ist erforderlich, um den Inhalt einzubeziehen. Der Inhalt kann obligatorisch sein und/oder optionale Token (/[Datenbank]), aber keine bedingten Teile. Es ist auch kann getestet werden, ob eine Variable vorliegt. Um dies zu tun, geben Sie den Variablenamen an z.B. Datenbank als Bedingung. Z. jdbc:mydb://:[database?/databaseName = ] wird entstehen jdbc:mydb://localhost:10000/databaseName = db1 wenn der Datenbankname angegeben ist der Knoten Dialog sonst wäre es jdbc:mydb://localhost:10000 .

Für **Serverbasiert** Datenbanken, die folgenden Token werden erwartet:

- Gastgeber: Der Wert des Hostname-Feldes auf der Verbindungseinstellungen Registerkarte eines Connectors Knoten.
- Hafen : Der Wert des Port-Feldes auf der Verbindungseinstellungen Registerkarte eines Verbindungsknotens.
- Datenbank : Der Wert des Datenbanknamensfelds auf der Verbindungseinstellungen Registerkarte einer Verbindungsknoten.

Für **Dateibasierte** Datenbanken, die folgenden Token werden erwartet:

- Standort : Die Standortwahl auf der Verbindungseinstellungen Registerkarte eines Verbindungsknotens. Die Dateiwert entspricht der Funktaste neben Pfad ausgewählt werden, und in-memory to die Funktaste neben In-Memory . Diese Variable kann nur unter Bedingungen verwendet werden.
- Datei : Der Wert des Pfad Feld auf dem Verbindungseinstellungen Registerkarte eines Verbindungsknotens. Das Variable ist nur gültig, wenn der Wert des Standorts Datei .
- Datenbank : Der Wert des In-Memory Feld auf dem Verbindungseinstellungen Registerkarte einer Verbindungsknoten. Diese Variable ist nur gültig, wenn der Wert des Standorts In-Memory .

Feldvalidierung im Konfigurationsdialog eines Verbindungsknotens hängt von  
ob die (inklusiven) Token, die sie beziehen, obligatorisch oder optional sind (siehe  
oben)

## Liste der gemeinsamen JDBC Treiber

Im Folgenden finden Sie eine ausgewählte Liste von gemeinsamen Datenbanktreibern, die Sie unter anderem zu KNIME hinzufügen können.

Analyseplattform:

- [Derby für die Zukunft](#)
- [Mehl](#)
- [Google BigQuery](#)
- [IBM DB2 / Informix](#)
- [SAP HANA](#)

Die oben genannte Liste zeigt nur ein Beispiel von Datenbanktreibern, die Sie hinzufügen können.

Wenn Ihr Treiber nicht in der oben genannten Liste ist, ist es immer noch möglich, es zu KNIME hinzuzufügen  
Analyseplattform.

## Depreierter JDBC Fahrer

Mit Version 5.3.0, KNIME Analytics Platform hat das Konzept der Abschreibung von JDBC vorgestellt  
Fahrer. Ältere Version von JDBC-Treibern wird abgeschrieben, wenn sie nicht von der  
Datenbank-Anbieter mehr oder wenn sie Sicherheitslücken enthalten. Abgeschriebene Treiber sind  
deutlich im Knoten-Dialog markiert und wenn ausgewählt, wird eine Warnung im Workflow-Editor angezeigt.

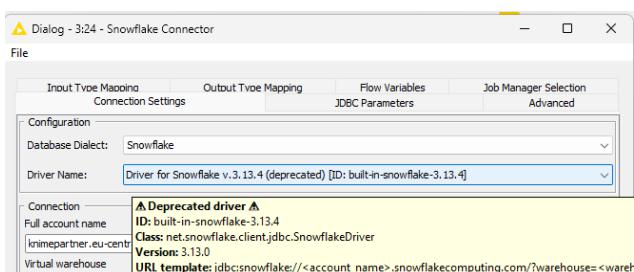


Abbildung 10. Tooltip eines deprecieren Treibers in  
die Treiberliste

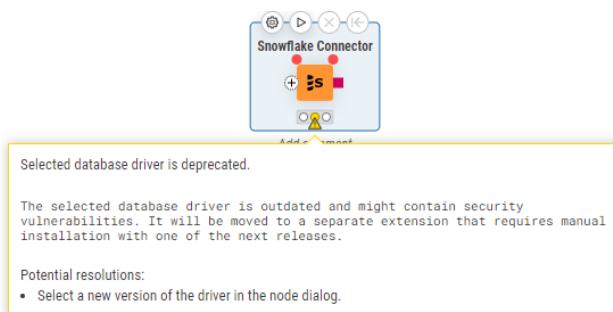


Abbildung 11. Keine Warnung, wenn eine Abschreibung  
Treiber wird ausgewählt

Nach einer gewissen Zeit werden abgeschriebene Treiber von der Plattform entfernt. Wenn ein solcher Treiber  
noch in einem Workflow referenziert, wird die KNIME Analytics Platform eine Ausnahme anzeigen

welche Erweiterung installiert werden muss, um den Fahrer wieder zur Verfügung zu stellen.

Um eine deprecated Treibererweiterung zu installieren, müssen Sie den Installationsdialog wie beschrieben öffnen

[https://docs.knime.com/latest/analytics\\_platform\\_installation\\_guide/](https://docs.knime.com/latest/analytics_platform_installation_guide/)

Index.html# install\_extensions\_and\_integrations[hier.] Wählen Sie im Installationsdialog die „Gruppenartikel nach Kategorie“ Option, suchen Sie nach der genannten Erweiterung und installieren Sie es. Nach Neustart von KNIME Analytics Platform wird der abgeschriebene Treiber wieder verfügbar sein.

Im Beispiel unten ist der deprecated Snowflake Treiber installiert.

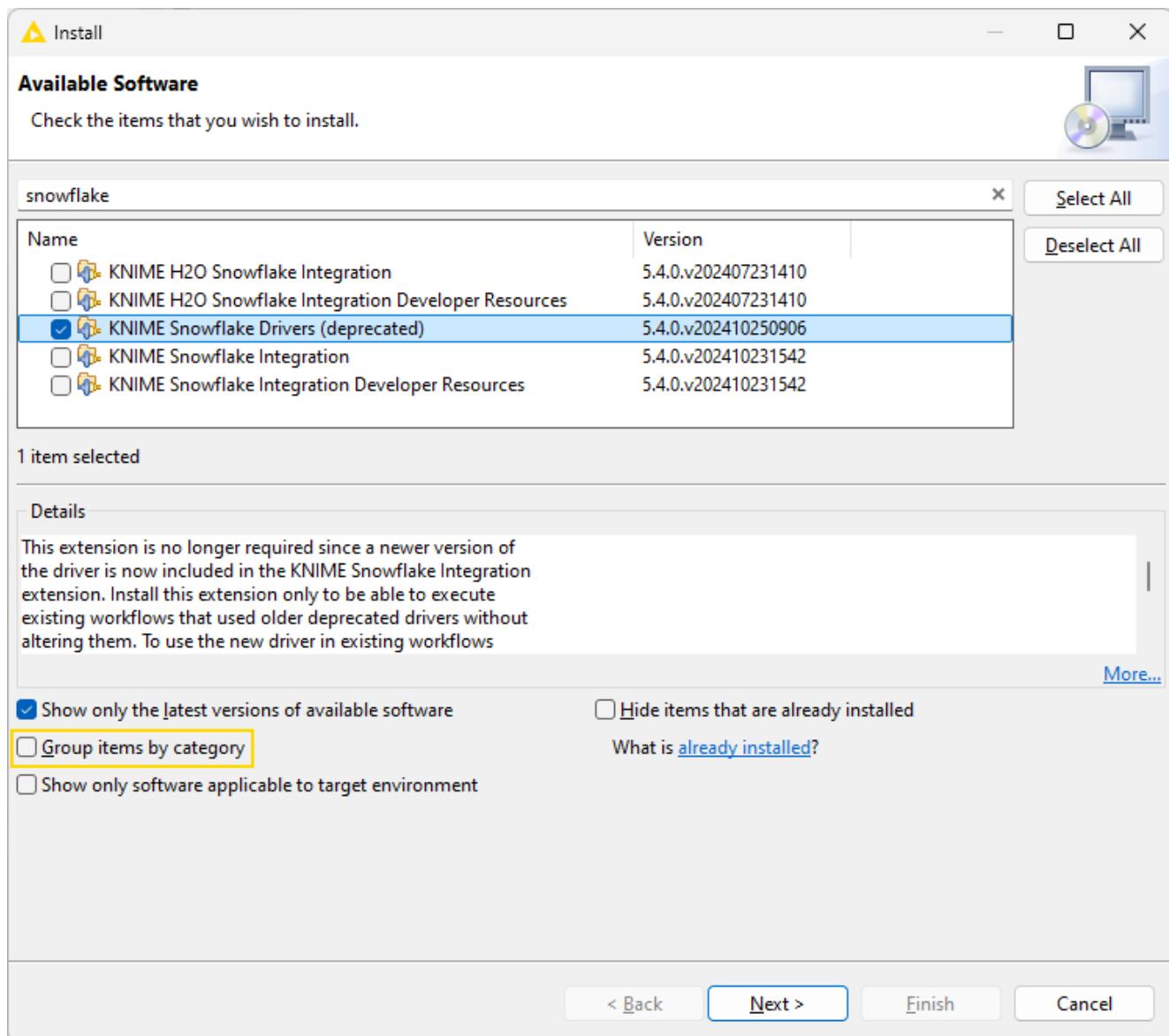


Abbildung 12. Installieren Sie deprecieerte Snowflake Treiber

## Erweiterte Datenbankoptionen

## JDBC Parameter

Mit den JDBC-Parametern können Sie benutzerdefinierte JDBC Treiberanschlussparameter definieren. Die ein Wert eines Parameters kann ein konstanter, variabler, Anmeldenutzer, Anmeldekennwort, KNIME sein URL oder [Wegflussvariable](#) . Bei einer Bahnflussgröße nur [Standard-Dateisysteme](#) werden unterstützt, aber nein [angeschlossene Dateisysteme](#). Für weitere Informationen über die unterstützten bitte auf Ihren Datenbank-Anbieter.

Die folgende Abbildung zeigt ein Beispiel für SSL JDBC-Parameter mit unterschiedlichen Variablenarten. Sie können einen Boolean einstellen, um zu aktivieren oder zu deaktivieren, Sie können auch eine KNIME relativ URL zu Der Präsident SLTrust Store Ort, oder verwenden Sie einen Anmeldeeingang für die Passwort vergessen? Parameter.

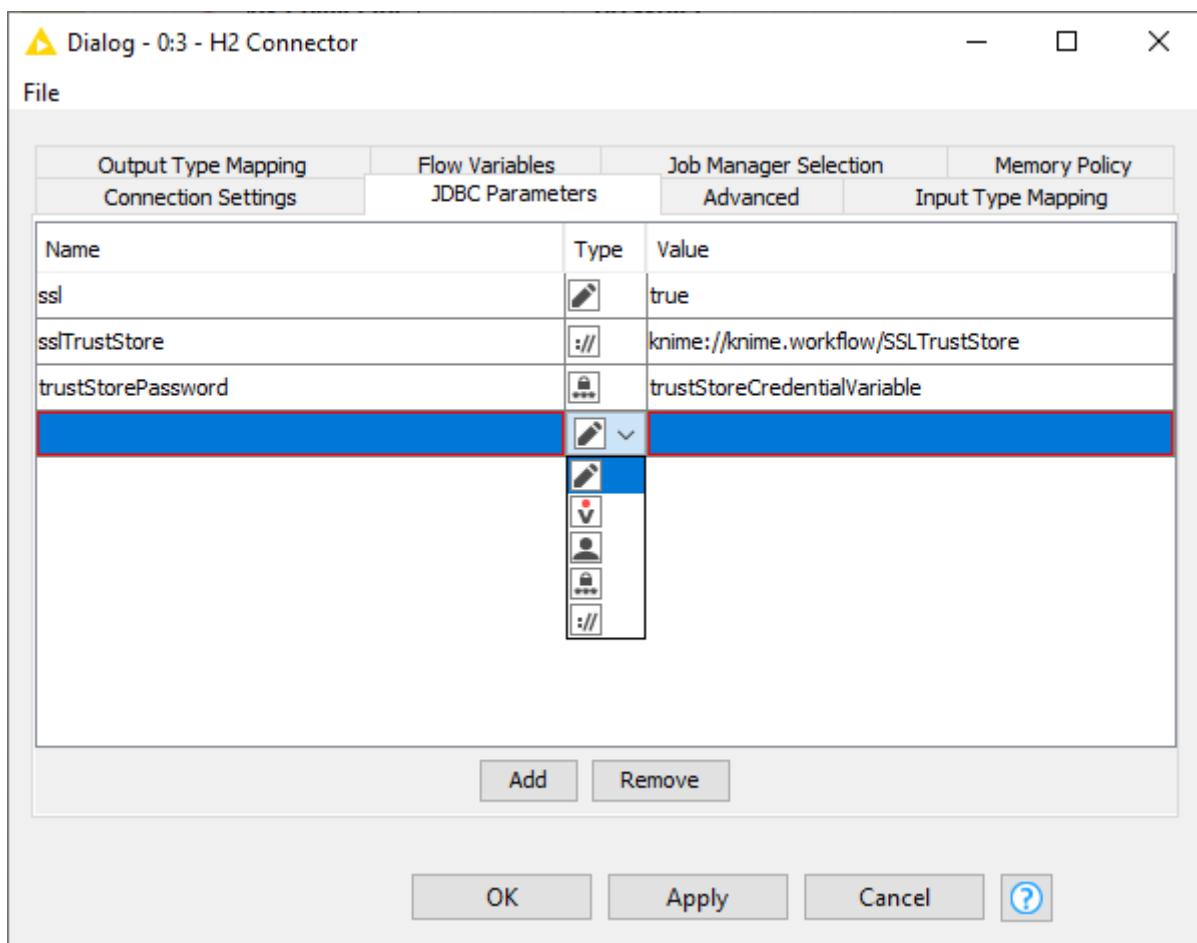


Abbildung 13. JDBC Parameter Register

Bitte beachten Sie, dass bei der Verbindung mit PostgreSQL mit SSL die Schlüsseldatei zuerst in beide umgewandelt Pkcs12 oder Pkcs8 Format. Für weitere Informationen über die unterstützten Treibereigenschaften [PostgreSQL Dokumentation](#)

## Erweiterte Registerkarte

Die Einstellungen in der Erweiterte Registerkarte ermöglichen Ihnen KNIME Framework-Eigenschaften wie Verbindungshandling, erweiterte SQL-Dialekt-Einstellungen oder Abfrage-Logging-Optionen. Das ist die Ort, an dem Sie z.B. mit der Datenbank interagieren können, wie die Abfragen sollte erstellt werden, die an die Datenbank gesendet werden. In der Metadaten Registerkarte können Sie auch Deaktivieren Sie das Metadaten-Fetching während der Konfiguration eines Knotens oder ändern Sie den Timeout, wenn Sie so was notwendig sein könnte, wenn Sie mit einer Datenbank verbunden sind, die mehr Zeit benötigt die Metadaten einer erstellten Abfrage berechnen oder über ein langsames Netzwerk mit ihr verbunden sind.

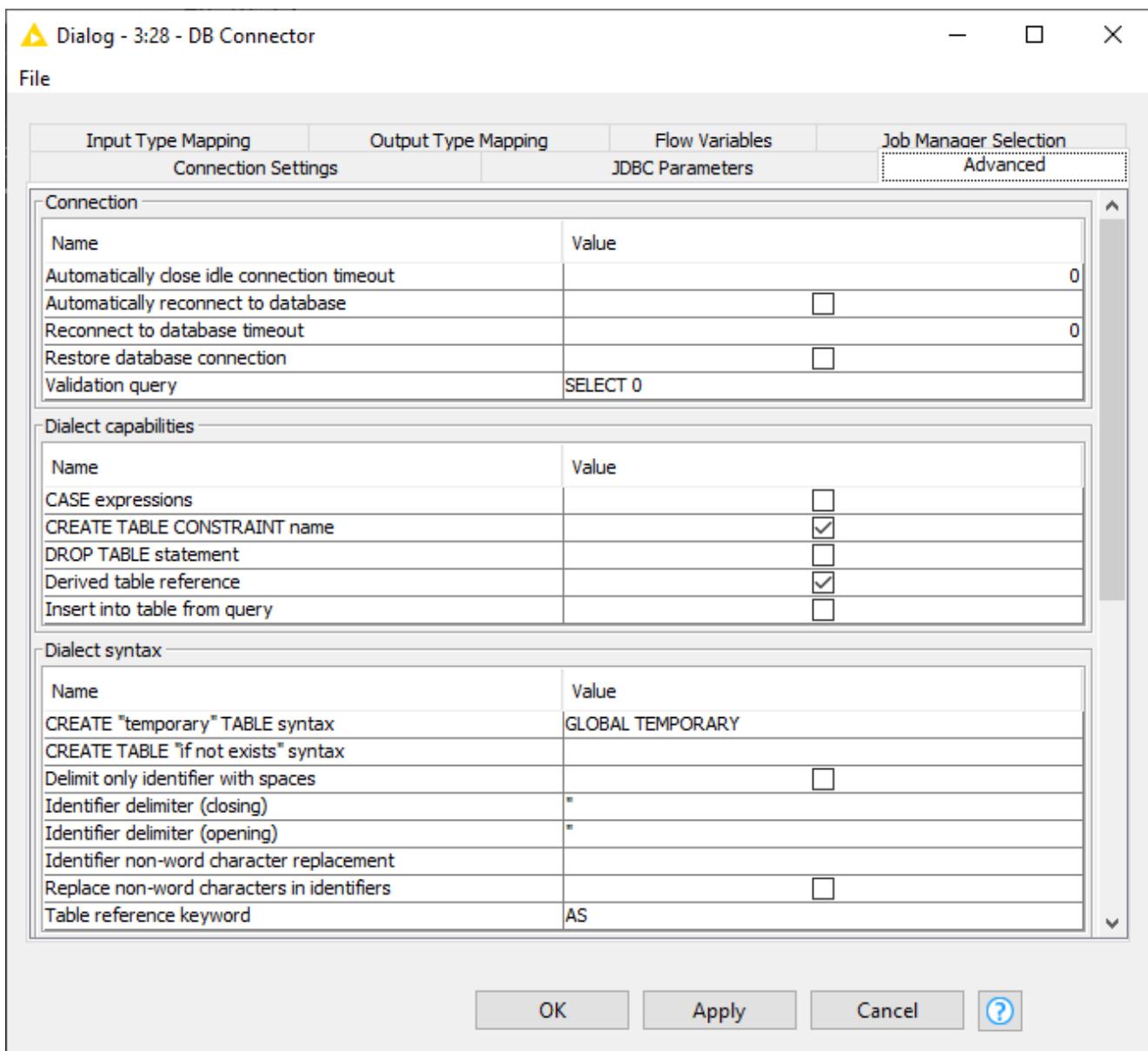


Abbildung 14. Erweiterte Registerkarte

Die vollen verfügbaren Optionen werden wie folgt beschrieben:

### Verbindung

- Automatisches Schließen der Leerverbindung Timeout : Zeitintervall in Sekunden, dass eine Datenbank die Verbindung kann leer bleiben, bevor sie automatisch geschlossen wird. Wert 0) Deaktivierung das automatische Schließen von Leeranschlüssen.
- Automatische Wiedervernetzung mit Datenbank : Aktiviert oder deaktiviert die Wiederverbindung mit dem Datenbank, wenn die Verbindung ungültig ist. Verbindung abhängiges Objekt nicht mehr vorhanden nach der Wiederverbindung.
- Wiederherstellen der Datenbank-Timeout : Zeitintervall in Sekunden zu warten, bevor Sie die Wiederverbindung zur Datenbank. Wert 0) gibt den Standard-Verbindungszeitout an.
- Datenbankverbindung speichern : Ermöglicht oder deaktiviert die Wiederherstellung der Datenbank Verbindung, wenn ein ausgeführter Steckverbinderknoten geladen wird.
- Validierungsanfrage : Die Abfrage zur Validierung einer Verbindung ist bereit für Verwendung. Wenn keine Abfrage angegeben ist KNIME ruft [Verbindung.isValid\(\)](#) Methode zur Validierung die Verbindung. Nur Fehler werden überprüft, kein Ergebnis ist erforderlich.

### Dialect Fähigkeiten

- CASE-Ausdrücke : Ob CASE-Ausdrücken in generierten Aussagen erlaubt sind.
- BEGRIFFSBESTIMMUNGEN : Ob Namen für CONSTRAINT definiert werden können Definitionen in CREATE-TABELLE-Anweisungen.
- DROP TABELLE Angabe : Ob DROP TABELLE Aussagen Teil der Sprache sind.
- Abgeleitete Tabellenreferenz : Ob Tabellenbezüge abgeleitet werden können.
- Einfügen in die Tabelle aus der Abfrage : Ob das Einfügen in eine Tabelle über eine ausgewählte Aussage unterstützt, z.B. INSERT INTO T1 (C1) (SELECT C1 FROM T2).

### Dialect syntax

- KREATE "temporary" TABELLE Syntax : Schlüsselwörter oder Schlüsselwörter für die Erstellung von temporären Tabellen.
- KREATE TABELLE "wenn nicht vorhanden" Syntax : Die Syntax für die Tabellenerstellungserklärung Zustand "wenn nicht vorhanden". Wenn leer, wird keine solche Erklärung automatisch erstellt, obwohl das gleiche Verhalten noch nicht-atomisch durch Knoten erreicht werden kann.
- Begrenzen Sie nur Kennung mit Leerzeichen : Wenn ausgewählt, nur Kennungen, z.B. Spalten oder Tabelle Namen, mit Räumen sind begrenzt.
- Identifier-Abgrenzer (geschlossen) : Schließabgrenzer für Kennung wie Spalte und Tabelle Name.
- Identifier-Abgrenzer (Öffnung) : Öffnungsbegrenzer für Kennung wie Spalte und Tabelle Name.

- Identifier Nicht-Wort-Zeichenwechsel : Der Ersatz für Nichtwortzeichen in Kennungen, wenn ihr Ersatz aktiviert ist. Ein Leerwert ergibt die Entfernung Nicht-Wort-Zeichen.
- Nicht-Wort-Zeichen in Kennungen ersetzen : Ob Nicht-Wort-Zeichen in ersetzen Kennungen, z.B. Tabellen- oder Spaltennamen. Nicht-Wort-Zeichen enthalten alle Zeichen andere als alphanumerische Zeichen (a-z, A-Z, 0-9) und Unterstrich (\_).
- Tabelle Referenzwort : Das Stichwort vor Korrelationsnamen in Tabellenreferenzen.

### JDBC-Rekorder

- Ermöglichen: Aktiviert oder deaktiviert Logger für JDBC-Operationen.

### JDBC Parameter

- JDBC-Parameter auf URL anhängen : Ermöglicht oder deaktiviert die Einstellung des Parameters zum JDBC URL anstatt sie als Eigenschaften zu übergeben.
- Benutzername und Passwort anhängen : Aktiviert oder deaktiviert die Appending des Benutzers Name und Passwort an die JDBC URL, anstatt sie als Eigenschaften zu übergeben.
- JDBC URL initial Parameter Separator : Das Zeichen, das den Beginn des Parameter in der JDBC URL.
- JDBC URL Parameter Separator : Das Zeichen, das zwei JDBC Parameter in die JDBC URL.

### JDBC-Anweisung Stornierung

- Ermöglichen: Aktiviert oder deaktiviert JDBC-Anweisung Stornierung versucht, wenn Knoten Ausführung wird abgesagt.
- Keine Stornobefragung Intervall : Die Menge der Millisekunden zwischen zwei warten Überprüfung, ob die Knotenausführung aufgehoben wurde. Gültigkeitsbereich: [100, 5000].

### Metadaten

- Flache Teilmengen, soweit möglich : Aktiviert oder deaktiviert Sub-Quer-Abflachung. Wenn aktiviert Subqueries z. B. SELECT \* AUS (SELECT \* FROM-Tabelle) KOLZ > 1 wird er SELECT \* Aus der Tabelle WHERE COL1 > 1 . Standardmäßig ist diese Option deaktiviert, da die Abfrage Abflachung ist in der Regel der Job des Datenbankabfrage-Optimierers. Allerdings einige Datenbank entweder Leistungsstörungen bei der Ausführung von Subqueries haben oder Subquitäten nicht unterstützen die Fragen überhaupt. In diesem Fall kann die Option helfen. Allerdings sind nicht alle Anfragen flachen, so dass sogar aktivierte Teil-Quries an die Datenbank gesendet werden können.
- Liste der Tabellentypen, die in Metadaten-Browser angezeigt werden : Komma getrennte Liste der Tabellentypen zu in Metadaten Browser anzeigen. Einige Datenbanken z.B. SAP HANA unterstützen mehr als die

Standard TABELLE und VI. Art wie CALC VIEW , HIERARCHIV und JOIN VIEW .

- Retrieve in Konfiguration : Aktiviert oder deaktiviert das Abrufen von Metadaten in der Konfigurationsmethode für Datenbankknoten.
- Abrufen im Konfigurations-Timeout : Zeitintervall in Sekunden zu warten, bevor eine metadata retrieval in konfigurieren Methode. Gültigkeitsbereich: [1, ).

## Umsetzung

- Ermöglicht : Aktiviert oder deaktiviert JDBC Transaktionsgeschäfte.

## Sonstiges

- Fehler, wenn WHERE-Klausel einen fehlenden Wert enthält : Alle Werte einer WHERE-Klausel überprüfen (z.B. Update, Löschen oder Zusammenführen) und scheitern, wenn man fehlt.
- Federgröße : Hinweis für den JDBC-Treiber über die Anzahl der Zeilen, die abgeholt werden sollten aus der Datenbank, wenn mehr Zeilen benötigt werden. Gültigkeitsbereich: [0,].
- Unterstützung mehrerer Datenbanken : Aktiviert oder deaktiviert Unterstützung für mehrere Datenbanken in einer ein einziges Statement.

Dedizierte DB-Steckverbinder (z.B. Microsoft SQL Server Connector) und eingebaute Treiber meist nur eine Teilmenge der oben genannten Optionen anzeigen, da die meisten Optionen vorgegeben sind, als ob die Datenbank CASE-Anweisungen unterstützt, etc.

## Beispiele

In diesem Abschnitt werden wir Beispiele dafür liefern, wie man sich mit einigen weithin bekannten Datenbanken verbindet.

### Verbindung mit Oracle

Der erste Schritt ist die Installation des Oracle Database JDBC Treibers, der als separater Treiber bereitgestellt wird.

mehr Informationen über das Plug-in und die Installation.

Es ist auch möglich, Ihren eigenen Oracle Database Treiber bei Bedarf zu verwenden. Für mehr Abschnitt.

Sobald der Treiber installiert ist, können Sie die dedizierte [Oracle Connector](#) Knoten. Bitte beachten Sie über die Verbindung mit dedizierten Connector-Nodes.

## Kerberos Authentifizierung

Um diesen Modus zu verwenden, müssen Sie wählen Kerberos als Authentifizierungsverfahren in der Verbindung EinstellungenTab des Oracle Connectors. Weitere Informationen zur Kerberos-Authentifizierung, Bitte beachten Sie die [Kerberos Benutzerhandbuch](#). Zusätzlich müssen Sie den folgenden Eintrag in die Tab: oracle.net.authentication\_services mit Wert (KERBEROS5). Bitte! nicht vergessen, den Wert in Klammern zu setzen. Für weitere Details siehe [Oracle-Dokumentation](#).

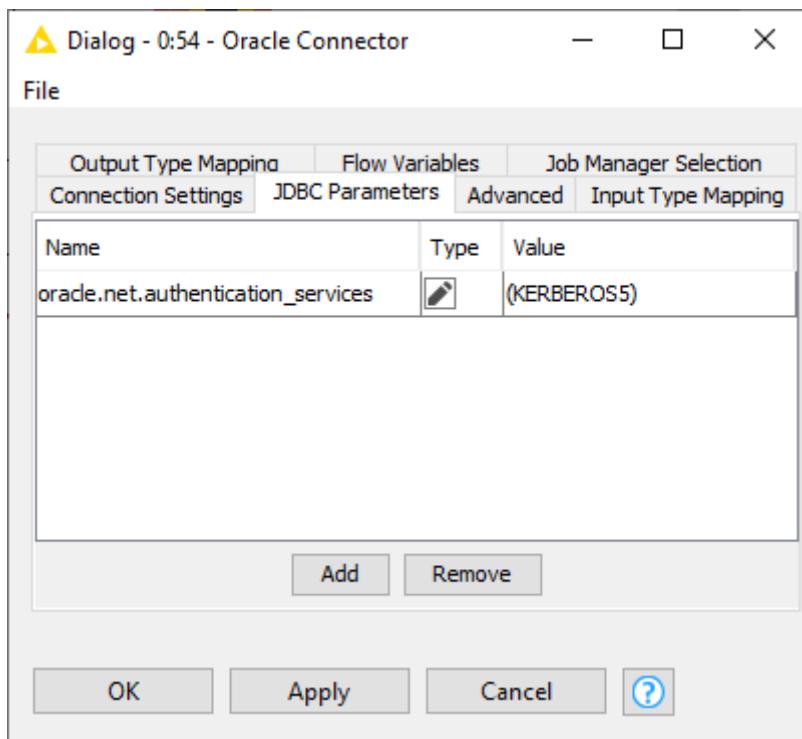


Abbildung 15. JDBC Registerkarte Parameter mit Kerberos Einstellungen

## Verbindung mit Databricks

Um mit Databricks zu verbinden, müssen Sie die [KNIME Integration von Databricks](#).

Der nächste Schritt ist das Herunterladen des Databricks Simba JDBC Treibers aus dem [Offizielle Website](#) bis Ihre Maschine. Dann gehen Sie zu Datei → Vorlieben → KNIME → Datenbanken, und klicken Hinzufügen.

KNIME bietet Apache Hive JDBC Treiber für Databricks, die Sie als einen Rückfalltreiber. Aber es wird dringend empfohlen, den offiziellen JDBC Treiber zu verwenden in der obenstehenden Verbindung vorgesehen.

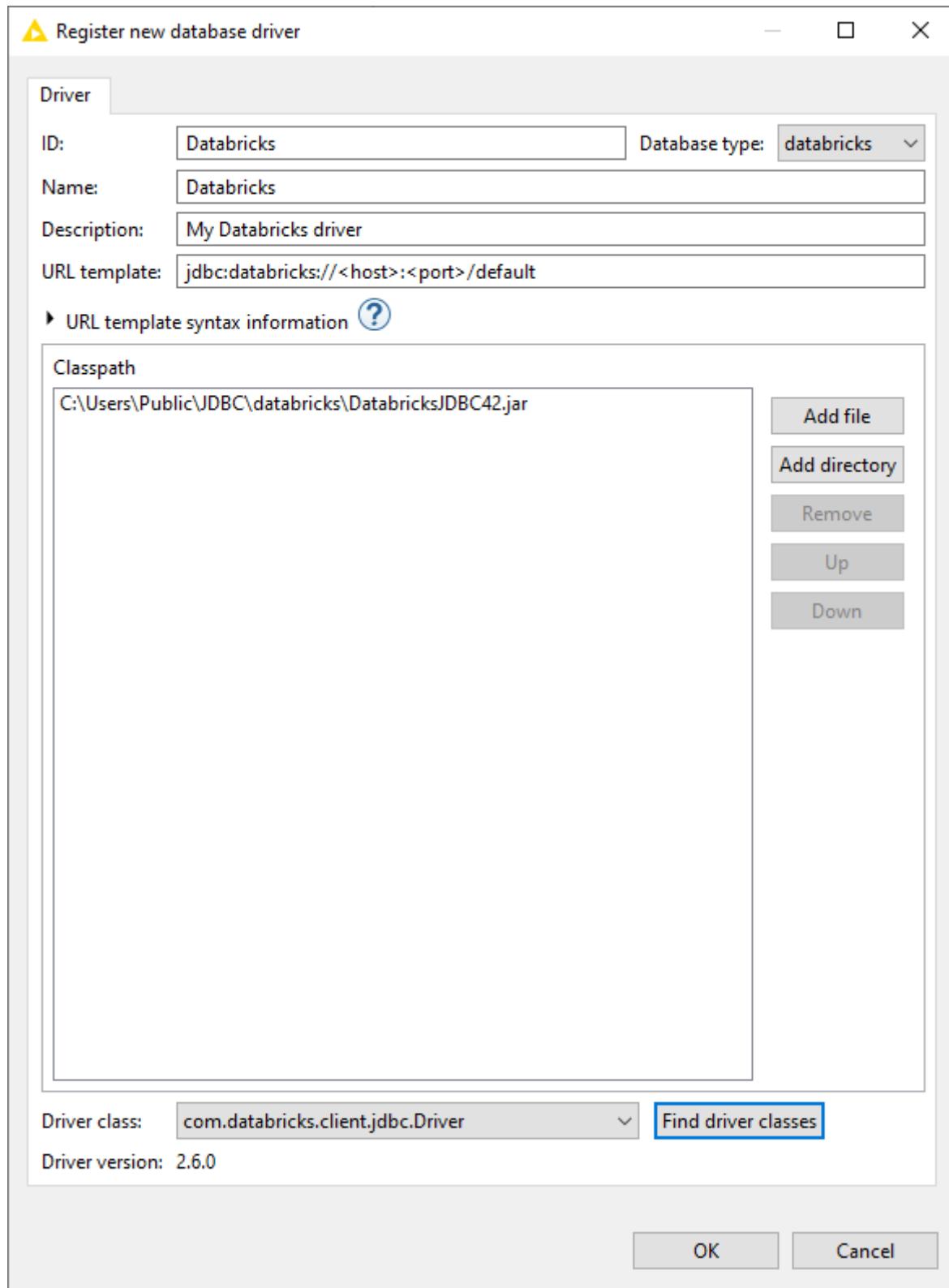


Abbildung 16. Registrieren Sie neue Databricks Treiber

Geben Sie im neuen Datenbanktreiberfenster folgende Informationen an:

- ID : Databricks, aber Sie können Ihre eigene Treiber-ID eingeben, solange es nur enthält alphanumerische Zeichen und Unterpunkte.

- Name : Databricks, aber Sie können Ihren eigenen Treibernamen eingeben.
- Datenbanktyp : Databricks ist in der Dropdown-Liste verfügbar, so dass der Datenbanktyp eingestellt wird bis Datenbrände .
- Warenbezeichnung: Mein Databricks-Treiber zum Beispiel.
- URL-Vorlage : Durch Auswahl Datenbrände in der Datenbanktyp , die URL-Vorlage ist automatisch auf die Standard-JDBC URL-Vorlage für Databricks voreingestellt, d.h.
 

`jdbc:spark://>port>/default` Für weitere mögliche Vorlagen klicken Sie einfach auf die [URL-Vorlage Syntaxle](#page13) direkt unten. Bitte beachten Sie die [Vorlage](#page15)

Abschnitt für weitere Informationen zu den unterstützten Token, z.B. Host, Port und Datenbank.
- Klasse : Klicken Sie auf Datei hinzufügen um die Databricks JDBC Treiberdatei hinzuzufügen. Der Weg zum Fahrer die Datei wird dann im Classpath Bereich erscheinen.
- Fahrerklasse : Anklicken Finden Sie Fahrerklassen automatisch alle verfügbaren JDBC erkennen Treiberklassen und Versionen, die in diesem Fall en.simba.spark.jdbc4.Fahrer in Version 2.6.0.

Nachdem Sie alle Informationen ausgefüllt haben, klicken Sie auf OK , und der neu hinzugefügte Treiber erscheint in der Datenbank Treiber Präferenzen Tabelle. Klicken Sie auf Anwenden und schließen die Änderungen anzuwenden.

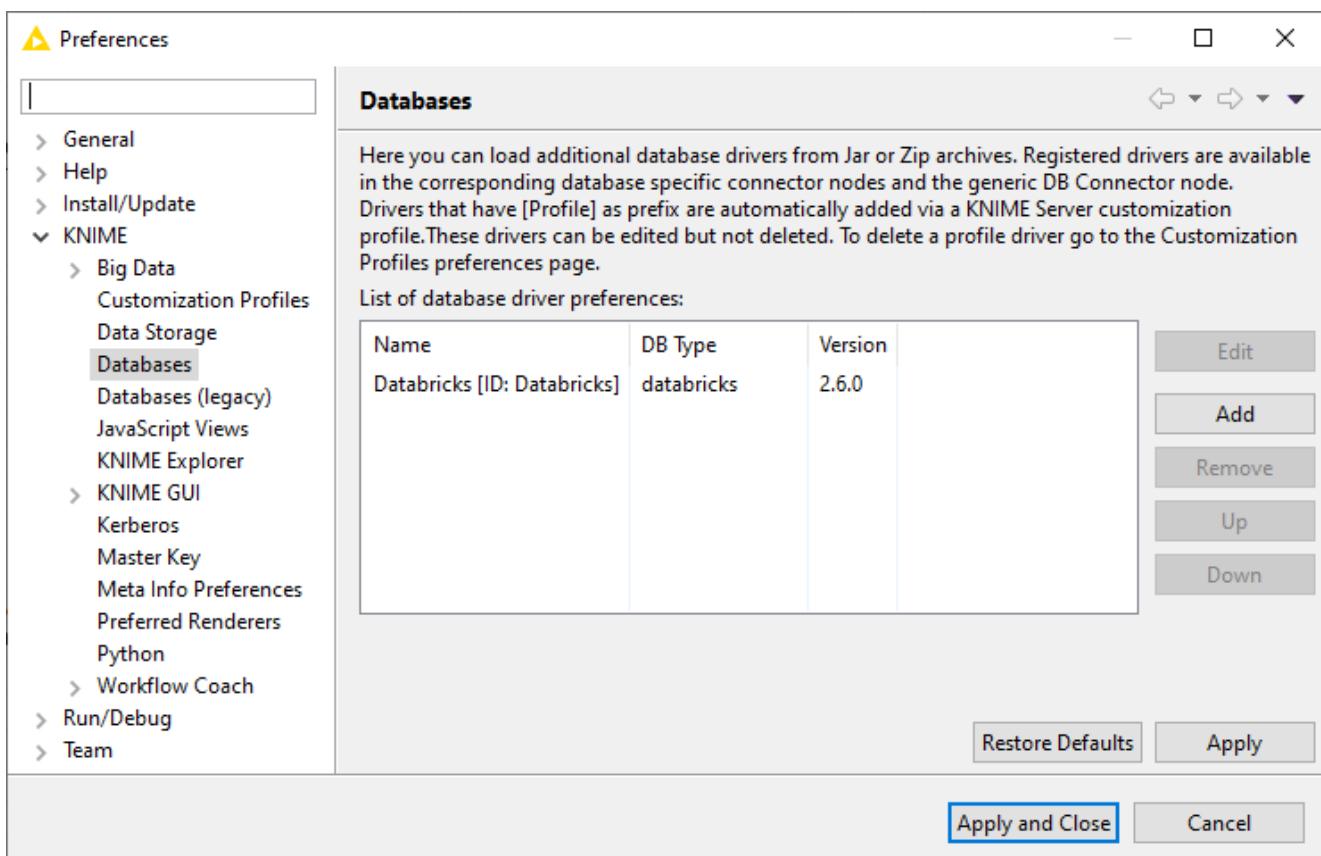


Abbildung 17. Datenbankeinstellungsseite

Um Databricks zu verbinden, müssen Sie zunächst eine Databricks-Umgebung erstellen, die angeschlossen ist

zu einem bestehenden Databricks-Cluster. Um dies zu tun, verwenden Sie die **Databricks Umwelt erstellen** Knoten. In das Konfigurationsfenster, das Sie bereitstellen müssen:

- **Datenschutzerklärung** : Vollständige URL der Databricks-Bereitstellung, die entweder <http://cloud.databricks.com> auf AWS oder <http://azuredatabricks.net> auf Azure.
- **Cluster ID** : Einzigartige Kennung eines Clusters im Databricks-Workspace.
- **Personalausweis** : Workspace-ID für Databricks auf Azure, lassen leer auf AWS.

In der **DB Port** → **Fahrer** Tab, Sie können den Datenbanktreiber auswählen, was in diesem Fall der Databricks Simba JDBC Treiber, den wir zuvor registriert haben.

Für die Authentifizierung empfiehlt Databricks die Verwendung von Token. Bitte beachten Sie die [Authentifizierung in Databricks](#) [AWS](#) oder [Azure](#) Dokumentation für weitere Informationen persönlicher Zugriff.

## Verbindung mit Google BigQuery

Um BigQuery zu verbinden, müssen Sie die [KNIME BigQuery Extension](#).

Aufgrund von Lizenzbeschränkungen ist der BigQuery JDBC-Treiber nicht Teil der KNIME Analytics Platform und muss separat heruntergeladen und registriert werden. Zum Download der BigQuery JDBC [Fahrer besuchen Sie bitte](#) [Offizielle Website](#). Dann gehen Sie zu Datei → Vorlieben → KNIME → Datenbanken, und klicken **Hinzufügen**.

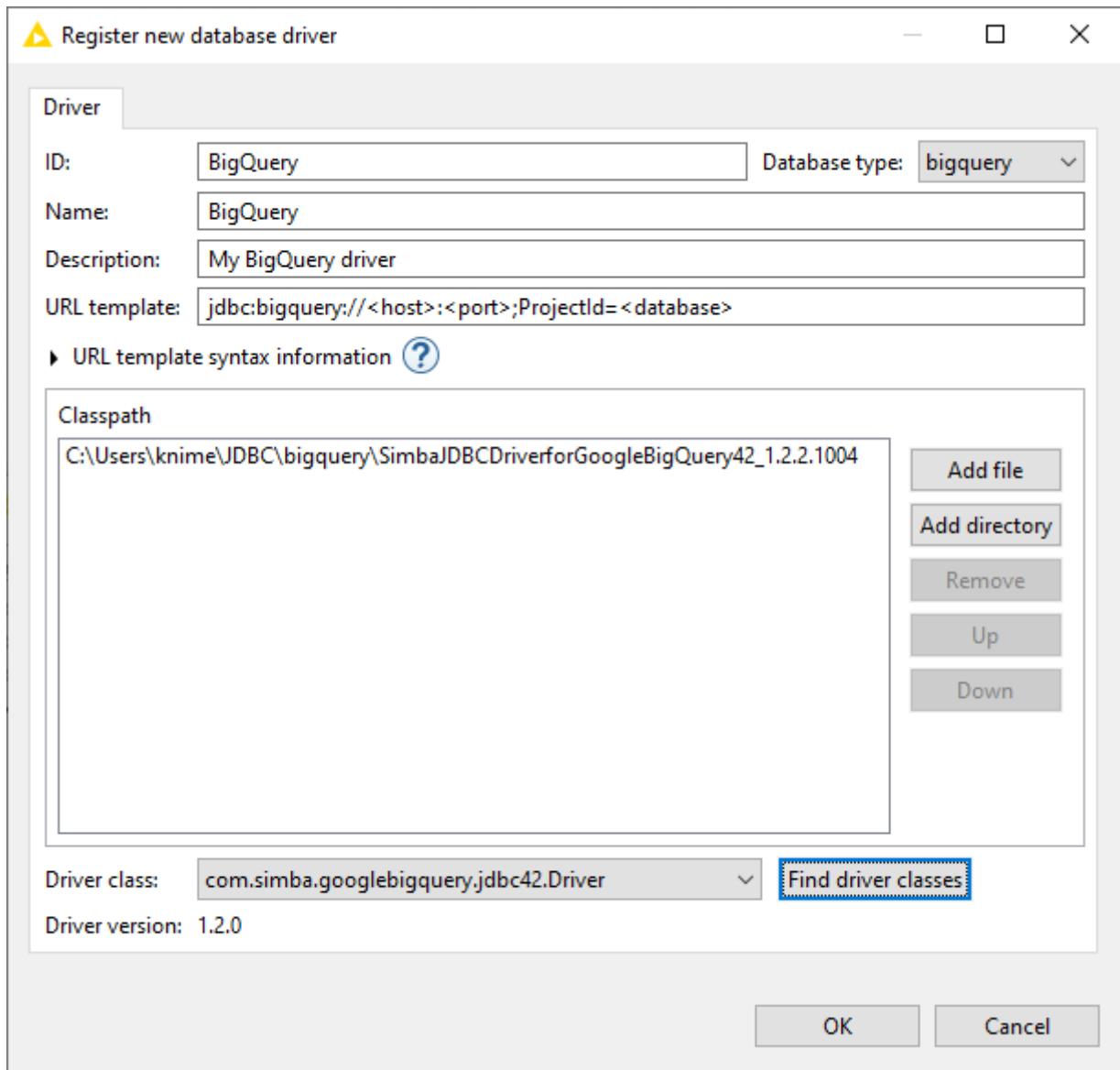


Abbildung 18. Neue BigQuery Treiber registrieren

Geben Sie im neuen Datenbanktreiberfenster folgende Informationen an:

- ID : BigQuery, aber Sie können Ihre eigene Fahrer-ID eingeben, solange es nur enthält alphanumerische Zeichen und Unterpunkte.
- Name : BigQuery, aber Sie können Ihren eigenen Fahrernamen eingeben.
- Datenbanktyp : BigQuery ist in der Dropdown-Liste verfügbar, so dass der Datenbanktyp auf Bigquery .
- Warenbezeichnung: Mein Google BigQuery Treiber zum Beispiel.
- URL-Vorlage : Durch Auswahl Bigquery in der Datenbanktyp , die URL-Vorlage ist automatisch auf die Standard-JDBC URL-Vorlage für BigQuery, d.h.

`jdbc:bigquery://:;ProjektId=`  
`<a href="#page13" style="color: #ff6600; text-decoration: underline;">Hilfe`

Bitte beachten Sie die

`Informationen #ff6600, text-decoration:underline;">Hilfe`  
 Abschnitt für mehr

Informationen zu den unterstützten Tokens wie Host, Port und Datenbank.

- Klasse : Klicken Sie auf Verzeichnis hinzufügen um das BigQuery JDBC-Treiberverzeichnis hinzuzufügen, das enthält die GoogleBigQueryJDBC42.jar-Datei und alle erforderlichen Begleitbibliotheken. Der Weg zum Fahrerverzeichnis erscheint dann im Bereich Classpath.
- Fahrerklasse : Anklicken Finden Sie Fahrerklassen automatisch alle verfügbaren JDBC erkennen Treiberklassen und Versionen, die in diesem Fall com.simba.googlebigquery.jdbc42.Driver in Version 1.2.0.

Achten Sie darauf, das Verzeichnis mit dem Google aufzunehmen BigQueryJDBC42.jar-Datei und alle erforderlichen Begleitbibliotheken.

Nachdem Sie alle Informationen ausgefüllt haben, klicken Sie auf **OK**, und der neu hinzugefügte Treiber erscheint in der Datenbank Treiber Präferenzen Tabelle. Klicken Sie auf **Anwenden** und schließen die Änderungen anzuwenden.

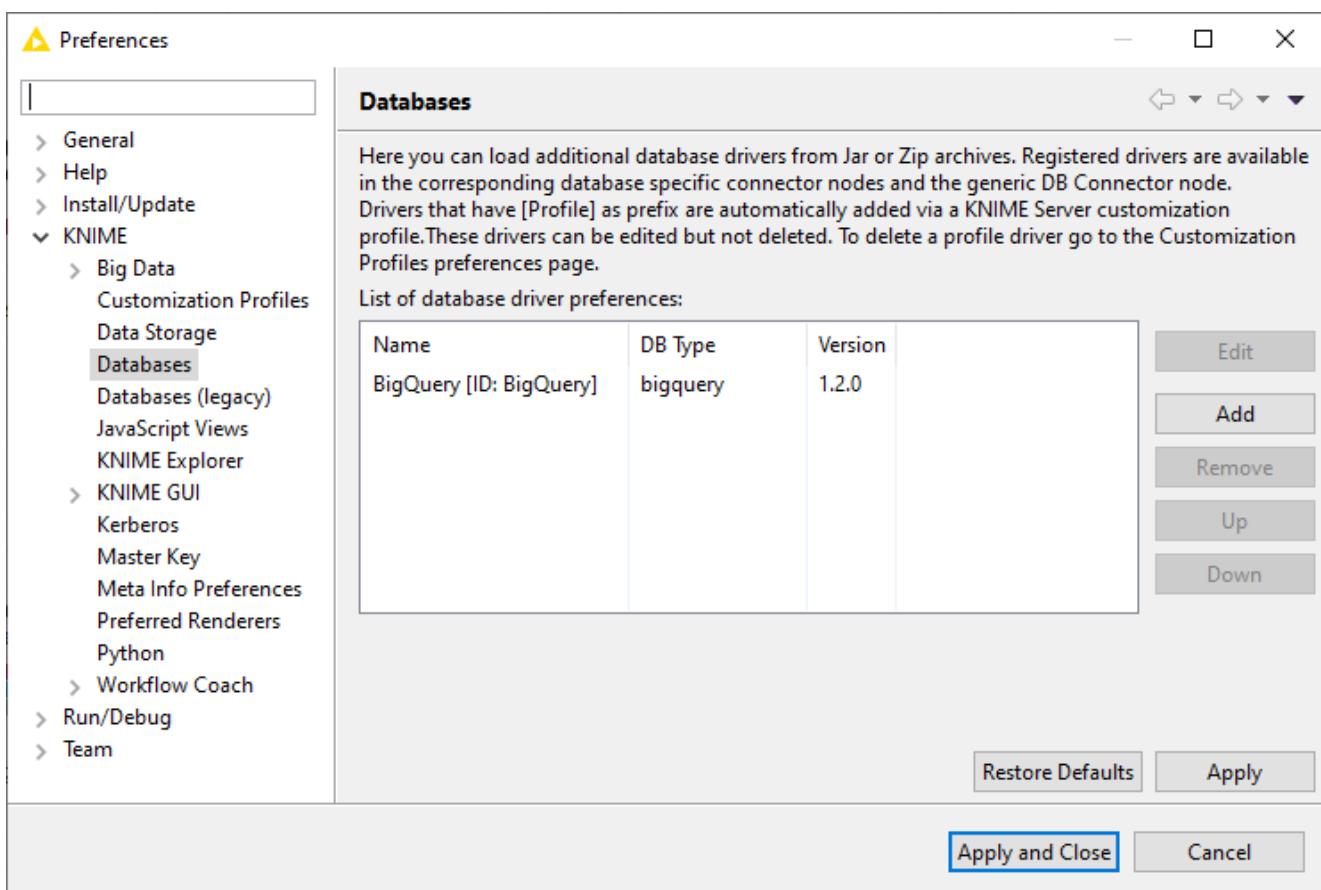


Abbildung 19. Datenbankeinstellungsseite

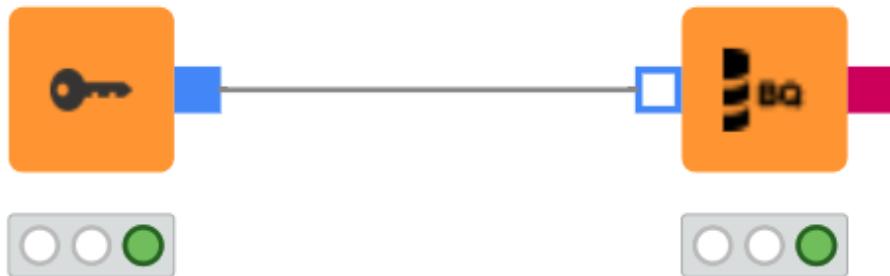
Um mit dem BigQuery Server zu verbinden, empfehlen wir, die authentifizieren und eine Verbindung zu den Google-APIs erstellen, und dann die

[Google Authentication](#)

Knoten zu

Anschluss Knoten zu BigQuery verbinden. Alternativ können Sie auch den Treiber angeben <a href="#page19" style="color: #000000; text-decoration: none; font-style: italic; font-weight: bold; font-size: 1em; border-bottom: 1px solid black; padding-bottom: 0.5em; margin-bottom: 0.5em;">Google BigQuery

# Google Authenticator      Google BigQuery Connector



## Service Account

Abbildung 20. Verbindung mit BigQuery

Anstelle des Google Authenticator-Knotens verwenden Sie auch den  
das Google-Geheimnis von der [Secrets Retriever](#)  
wie Sie ein Google-Geheimnis im Secret Store einrichten, lesen Sie bitte die [Gefällt mir](#). Für weitere Details  
[KNIME Secrets Benutzerhandbuch](#).

Im Konfigurationsfenster der [Google Authentication](#) node, you need to select Service  
Gesamt als Authentifizierungstyp. Zusätzlich müssen Sie den Authentifizierungsschlüssel eingeben  
Informationen (Siehe die [Google-Dokumentation](#) über die Erstellung einer Service-Account-Schlüsseldatei).  
Schließlich müssen Sie die Bereiche auswählen, die für diese Verbindung gewährt werden. Zum Beispiel  
Auswahl Google BigQuery Sie können Ihre Daten in Google BigQuery anzeigen und verwalten.

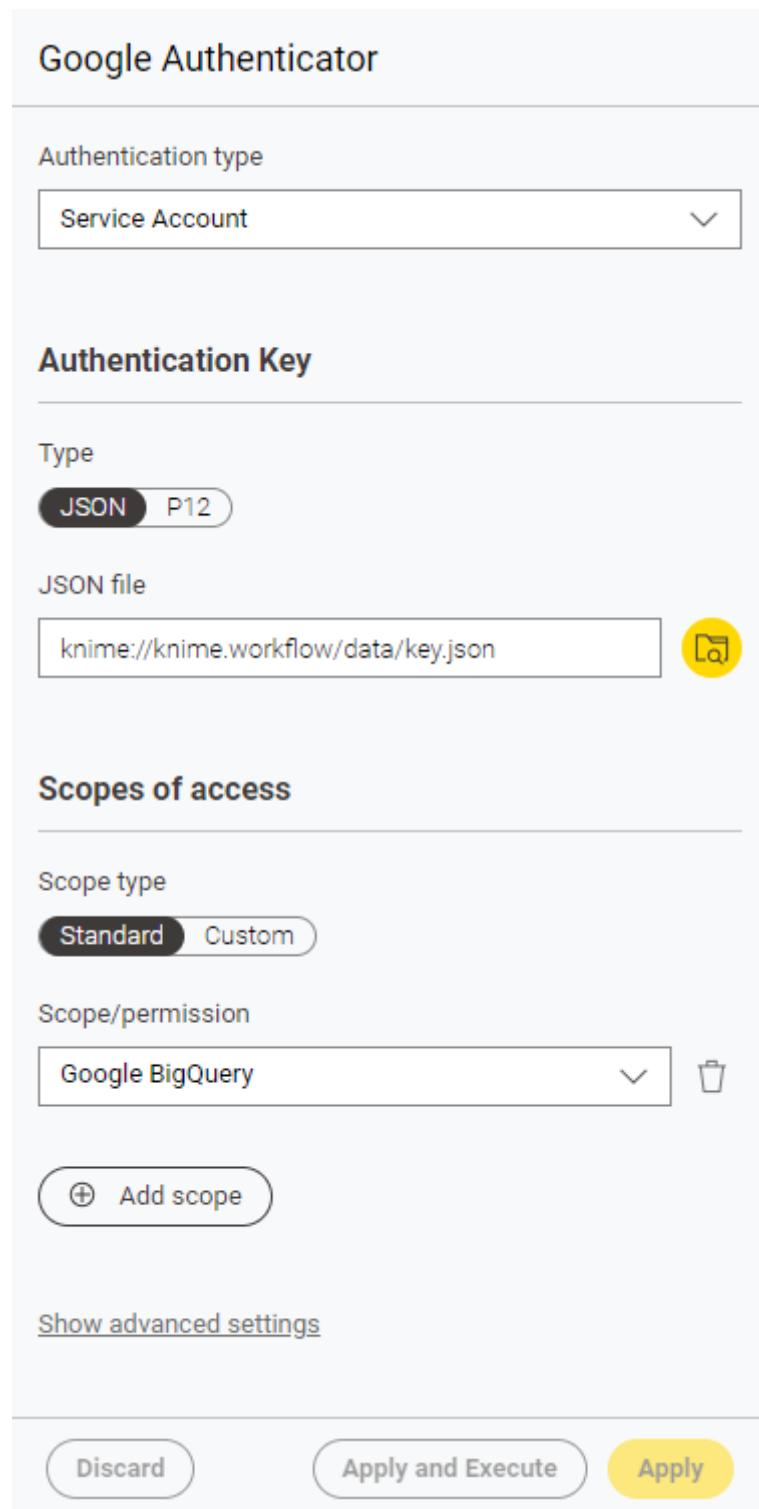


Abbildung 21. Konfigurationsfenster von Google Authenticator Node

Der Google BigQuery Connector-Knoten unterstützt nur die Authentifizierung über einen Service Konto oder Ihr eigener OAuth-Client (Zum Detail siehe die [Google-Dokumentation](#))

Nachdem die Verbindung erfolgreich aufgebaut ist, kann sie als Eingang verwendet werden  
 Verbindung für die Google BigQuery Connector Knoten. Das Konfigurationsfenster enthält als folgendes:

- Datenbank Treiber : Der Name des BigQuery Fahrers, den wir früher gegeben haben, wenn wir registriert den Fahrer. In unserem Beispiel ist es Großes Angebot
- Hostname : Der Hostname (oder IP-Adresse) eines Servers von Google BigQuery.
- Hafen : Der Port, auf dem der Google BigQuery Server zu hören ist. Der Standardport ist 443.
- Datenbankname : Der Name (Projekt-ID) der Datenbank, mit der Sie sich verbinden möchten.

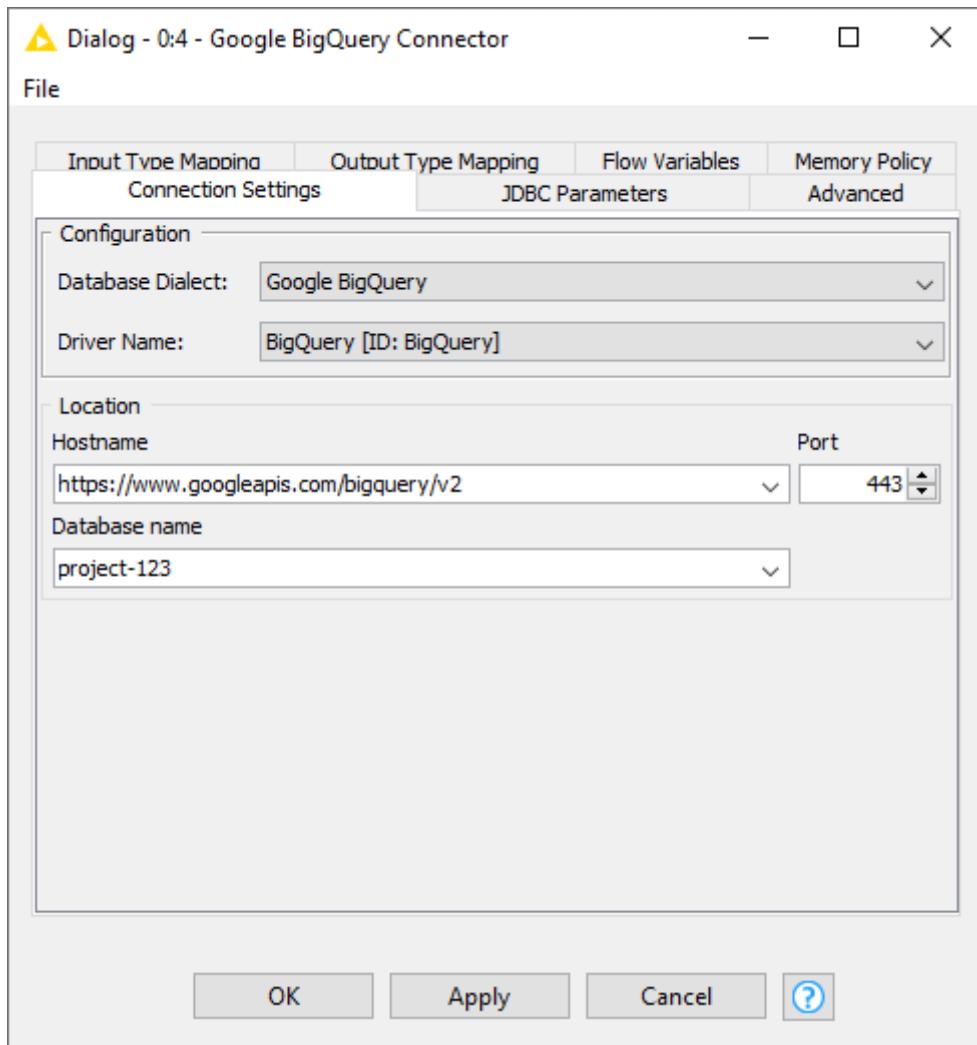


Abbildung 22. Konfigurationsfenster von Google BigQuery Connector Node

## Verbindung mit Microsoft SQL Server

Die gewidmet

Microsoft SQL Server Connector

Knoten wird standardmäßig mit dem

[iTDS für](#)[Microsoft SQL Server Treiber](#)

. Wenn Sie die

[offizieller Treiber für Microsoft SQL Server](#)

Stattdessen bietet KNIME ein zusätzliches Plug-in, um den Treiber zu installieren. Bitte beachten Sie [Datenbank Treiber Plug-in](#) für weitere Informationen zum Plug-in und zur Installation.

Es ist auch möglich, Ihren eigenen Microsoft SQL Server Treiber zu verwenden. So registrieren Sie Ihren eigenen Fahrer [Abschnitt](#). Der Microsoft SQL

Servertreiber benötigen möglicherweise mehrere native Bibliotheken, wie DLLs. In diesem Fall müssen Sie

alle benötigten nativen Bibliotheken in einen einzigen Ordner kopieren und diesen Ordner dann über den Verzeichnis hinzufügenKnopf zusätzlich zum JDBC .jar Datei im Datenbanktreiberfenster. Das Schritt ist nicht erforderlich, wenn Sie den bereitgestellten JTDS für Microsoft SQL Server Treiber oder den offiziellen Treiber für Microsoft SQL Server installiert durch das Plug-in.

 Alle notwendigen Dateien wie die sqljdbc\_auth.dll oder die ntlmauth. dll sind Teil von die bereitgestellten Treiber.

Nach der Installation des JDBC Treibers können Sie die Verbindung zu Microsoft SQL Server. Bitte beachten Sie mehr Informationen über die Verbindung mit dedizierten Connector-Knoten.

Microsoft SQL Server Connector  
[Knoten zu starten für](#page6)

Microsoft SQL Server unterstützt den sogenannten Windows native Authentifizierung Modus. Wenn Sie diesen Modus verwenden möchten, wählen Sie einfach Keine/native Authentifizierung in der Authentifizierung Einstellung im Konfigurationsfenster. Die folgenden Abschnitte erklären, wie dieser Modus abhängig verwendet werden kann auf welchem Treiber Sie verwenden.

 Windows native Authentifizierung funktioniert nur unter Windows. Wenn Sie laufen Workflows auf KNIME Hub oder KNIME Server wird der KNIME Executor höchstwahrscheinlich unter Linux laufen, was Benutzer-Login sowie zusätzliche JDBC-Parameter erfordert. Details finden Sie im folgenden Abschnitt.

Windows native Authentifizierung (NTLM) mit dem offiziellen Treiber für Microsoft SQL Server

Um diesen Modus mit dem bereitgestellten offiziellen Treiber für Microsoft SQL Server zu verwenden, KNIME Analytics Plattform muss auf einer Windows-Maschine ausgeführt werden und Sie müssen in einem Windows angemeldet werden Domain, die vom Microsoft SQL akzeptiert wird Server mit dem Sie verbinden. Zusätzlich, Sie [Tab:](#page19)

IntegrierteSicherheit = wahr

Für weitere Details siehe [Microsoft-Dokumentation](#).

wenn KNIME Analytics Platform läuft auf einem Nicht-Windows-Maschine, Sie müssen dem Benutzer zur Verfügung stellen Name und Passwort des Windows-Domain-Benutzers, den Sie zur Authentifizierung verwenden möchten. Zu tun Bitte wählen Sie entweder Angaben Option oder Benutzername und Passwort Option.

Zusätzlich müssen Sie die folgenden Einträge in den

[Tab:](#page19)

AuthentifizierungScheme = NTLM  
Domain-Name der Domain-Namen

Weitere Einzelheiten zu den erforderlichen Parametern finden Sie in der [Dokumentation](#).

## Microsoft JDBC Treiber

Windows native Authentifizierung (NTLM) mit dem jTDS Treiber für Microsoft SQL Server

Wenn Sie den bereitgestellten jTDS für Microsoft SQL Server Treiber und KNIME Analytics verwenden

Plattform läuft auf einer Windows-Maschine, die in eine Windows-Domain eingeloggt ist, die akzeptiert wird

von Microsoft SQL Server, mit dem Sie verbunden sind, dann müssen Sie keine JDBC angeben.

Parameter.

wenn KNIME Analytics Platform läuft auf einem Nicht-Windows-Maschine, Sie müssen dem Benutzer zur Verfügung stellen

Name und Passwort des Windows-Domain-Benutzers, den Sie zur Authentifizierung verwenden möchten. Zu tun ist dies nicht erforderlich.

Bitte wählen Sie entweder Angaben Option oder Benutzername und Passwort Option. Außerdem,  
Sie müssen den folgenden Eintrag in der [Tab:](#page19)

## Domain-Name der Domain-Namen

Weitere Informationen finden Sie in der Beschreibung der Domaineigenschaft in der

[JTDS FAQ](#)

oder sehen Sie

## README.sso

Kerberos Authentifizierung mit dem offiziellen Treiber für Microsoft SQL Server

Um diesen Modus mit dem bereitgestellten offiziellen Treiber für Microsoft SQL Server zu verwenden, wählen Sie einfach

Kerberos in der Authentifizierung Einstellung im Konfigurationsfenster. Zusätzlich müssen Sie [Tab:](#page19)

AuthentifizierungScheme = JavaKerberos

IntegrierteSicherheit = wahr

Für weitere Details siehe

[Microsoft-Dokumentation](#)

## Verbindung mit SQL Pools in Azure Synapse Analytics

Sie können die dedizierte

## Microsoft SQL Server Connector Node

eine Verbindung zu einem gewidmet

## SQL-Pool oder S

oder [Serverless SQL Pool](#) in Azure Synapse Analytics. Vor der Verbindung müssen Sie

installieren Sie die [offizieller Treiber für Microsoft SQL Server](#page9).  
B. als zusätzliches Einsticken vorgesehen ist.  
Bitte beachten Sie für weitere Informationen zum Plug-in und

wie zu installieren. Es ist auch möglich, Ihren eigenen Microsoft SQL Server Treiber zu verwenden. Anmeldung:  
[Abschnitt 12](#page12)

Um eine Verbindung herzustellen, müssen Sie den serverlosen und/oder dedizierten SQL Endpoint und einen Login-Benutzer erhalten z.B. zuerst der SQL-Administrator. Diese Informationen sind über die Synapse verfügbar.

### Arbeitsraumansicht

### Azure Portal

Name	Type	Size
Built-in	Serverless	Auto

Abbildung 23. Synapse Arbeitsraumansicht im Azure-Portal

Sobald Sie den offiziellen Treiber installiert haben und alle notwendigen Informationen verfügbar sind, können Sie

konfigurieren Sie [Microsoft SQL Server Connector Node](#)

. So öffnen Sie den Knoten Dialog und geben Sie ein

den serverlosen oder dedizierten SQL-Endpoint als

Wir sind hier mit Meister als Standard-Datenbank.

Geben Sie die Anmeldeinformationen in den Authentifizierungsabschnitt ein, wie unten gezeigt. Anstelle eines

Benutzername und Passwort können Sie auch Microsoft Entra ID-Login verwenden, wenn dies in

Ihr Synapse-Workspace mit

[Microsoft Authenticator Node](#)

und der dynamische Eingangsport

des Microsoft SQL Server Connectors.

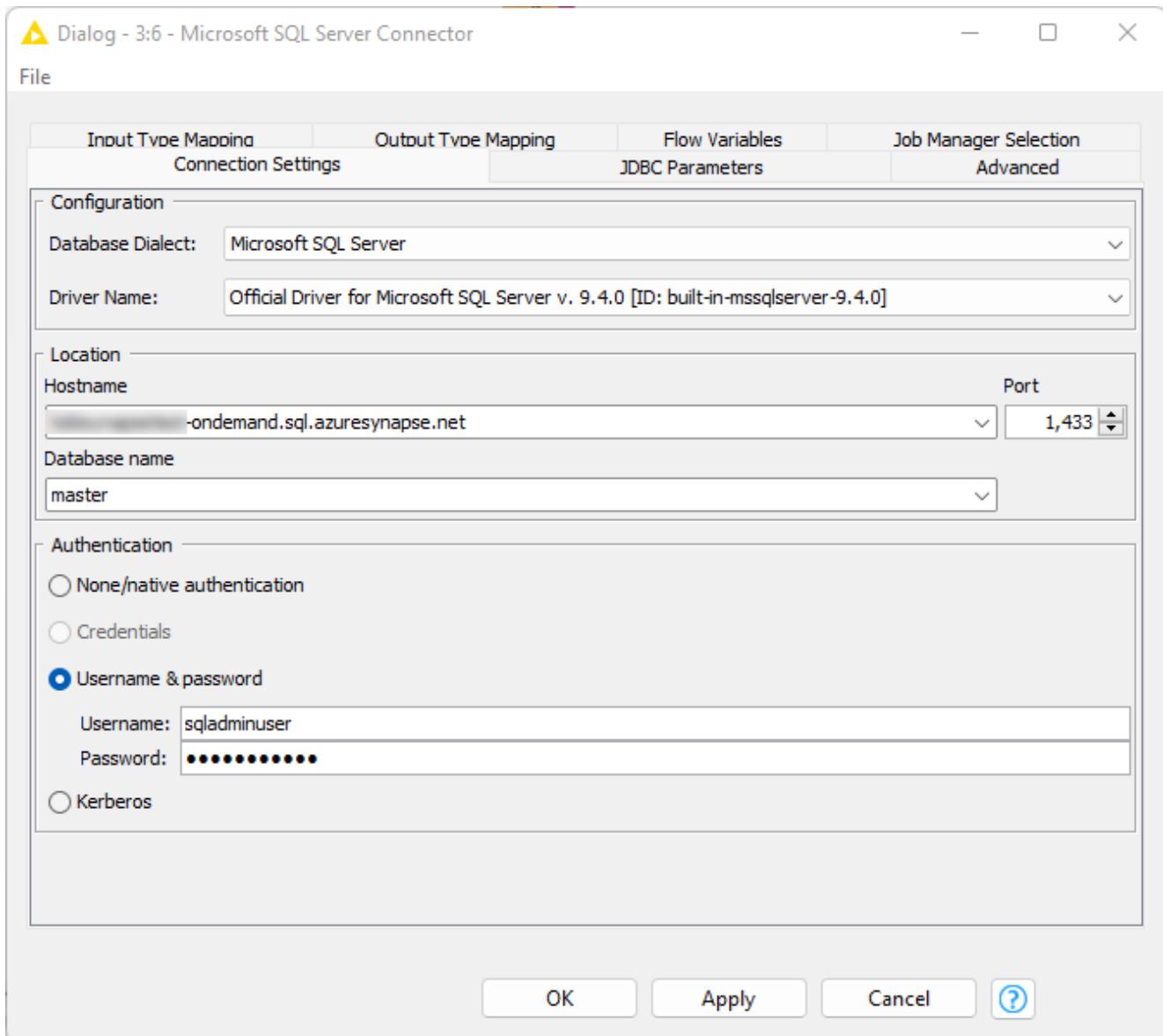


Abbildung 24. Microsoft SQL Server Connector Dialog mit Verbindungseinstellungen

Darüber hinaus müssen Sie die folgenden JDBC-Parameter über die Tab:

- TrustServerZertifikat : falsch
- Verschlüsselung wahr
- hostNameInCertificate : \*.sql.azure-synapse.net

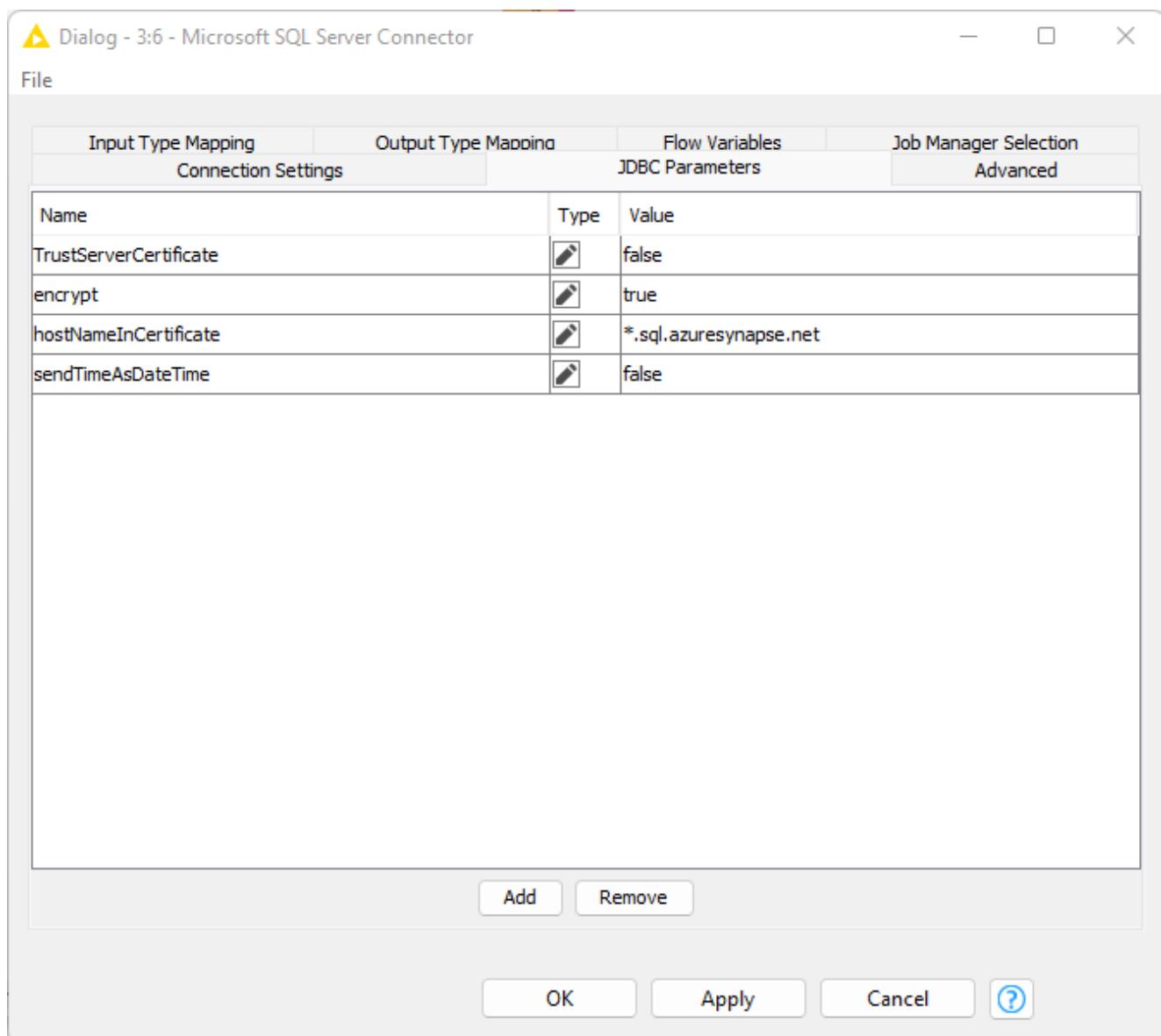


Abbildung 25. JDBC-Parameter für Azure Synapse

Weitere Informationen zu den JDBC-Parametern finden Sie in der

[Microsoft-Dokumentation](#)

Schließlich für **Serverless SQL Pools** Sie müssen die Transaktion deaktivieren, da sie nicht unterstützt werden  
 über die [Erweiterte Registerkarte](#page20) durch Auswählen der Ermöglicht Wert in der Umsetzung Abschnitt.

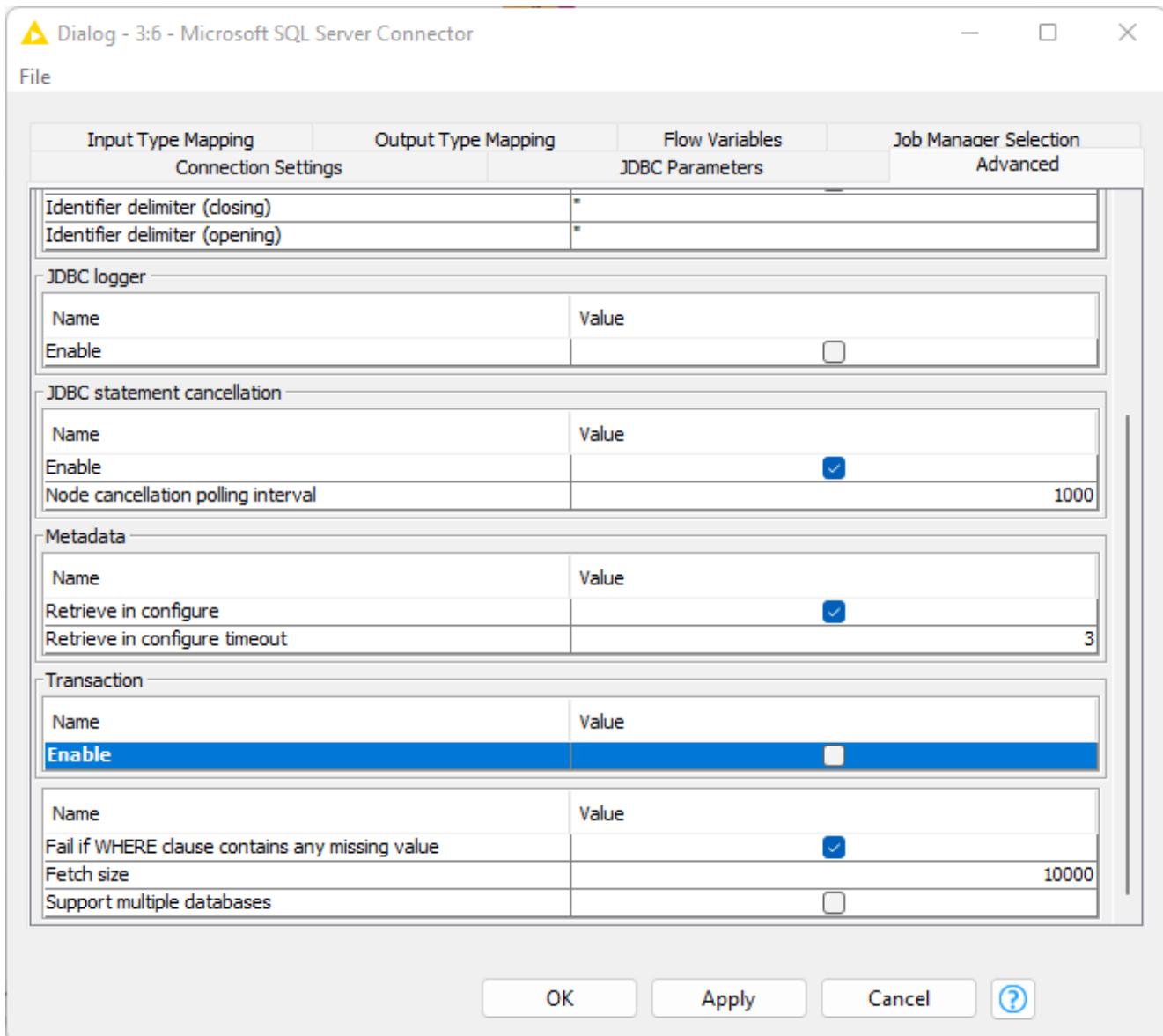


Abbildung 26. Deaktivieren Sie die Transaktionsunterstützung über die Erweiterte Registerkarte

### Verbindung mit Apache HiveTM

Um mit Hive zu verbinden, müssen Sie die

[KNIME Big Data Connectors Erweiterung](#)

Die gewidmet **Hive Connector** node wird standardmäßig mit dem open-source Apache Hive gebündelt JDBC Treiber. Auch proprietäre Treiber werden unterstützt, müssen aber zuerst registriert werden, wie z. den Hive JDBC-Steckverbinder von Cloudera.

In diesem Beispiel möchten wir mit dem proprietären Cloudera Hive JDBC Treiber mit Hive verbinden.

Der erste Schritt ist das Herunterladen der neuesten [Hive JDBC Treiber für Cloudera Enterprise](#). Dann gehen Sie zu Datei → Vorlieben → KNIME → Datenbanken, und klicken **Hinzufügen**.

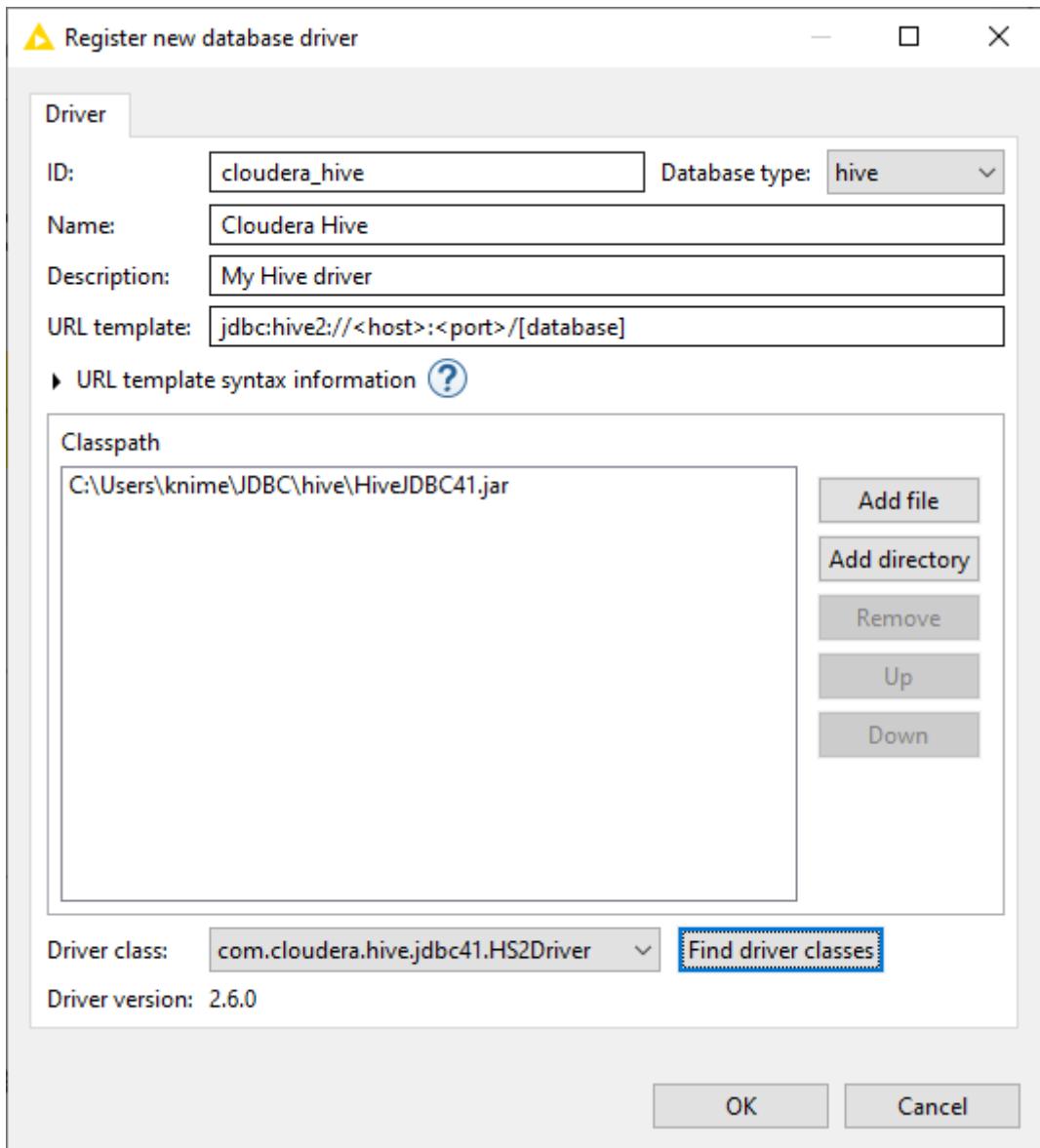


Abbildung 27. Registrieren Sie neue Apache Hive driver

Geben Sie im neuen Datenbanktreiberfenster folgende Informationen an:

- ID : cloudera\_hive, aber Sie können Ihre eigene Treiber-ID eingeben, solange es nur enthält alphanumerische Zeichen und Unterpunkte.
- Name : Die Welt Hive, aber Sie können Ihren eigenen Fahrernamen eingeben.
- Datenbanktyp : Hive ist in der Dropdown-Liste verfügbar, so dass der Datenbanktyp auf Hive .
- Warenbezeichnung: Mein Hive-Treiber zum Beispiel.
- URL-Vorlage : Bitte stellen Sie sicher, dass dieses Feld enthält jdbc:hive2://:[database] . Bitte beachten Sie die [Abschnitt für weitere Informationen zu den unterstützten Tokens wie Host, Port und Datenbank.](#page15)
- Klasse : Klicken Sie auf Datei hinzufügen, um die .jar Datei, die den Hive JDBC Treiber enthält. Die

Pfad zur Treiberdatei wird dann im Classpath Bereich angezeigt.

- Fahrerklasse : Anklicken Finden Sie Fahrerklassen automatisch alle verfügbaren JDBC erkennen Fahrerklassen und Versionen. Bitte stellen Sie sicher, dass Sie mit einem Gehalt an Fahrer .

Endlich, klicken Sie Ok. und der neu hinzugefügte Treiber erscheint in den Einstellungen des Datenbanktreibers Tisch. Klicken Sie auf Anwenden und schließen die Änderungen anzuwenden und Sie können die Verbindung zu Ihrem Hive Datenbank.

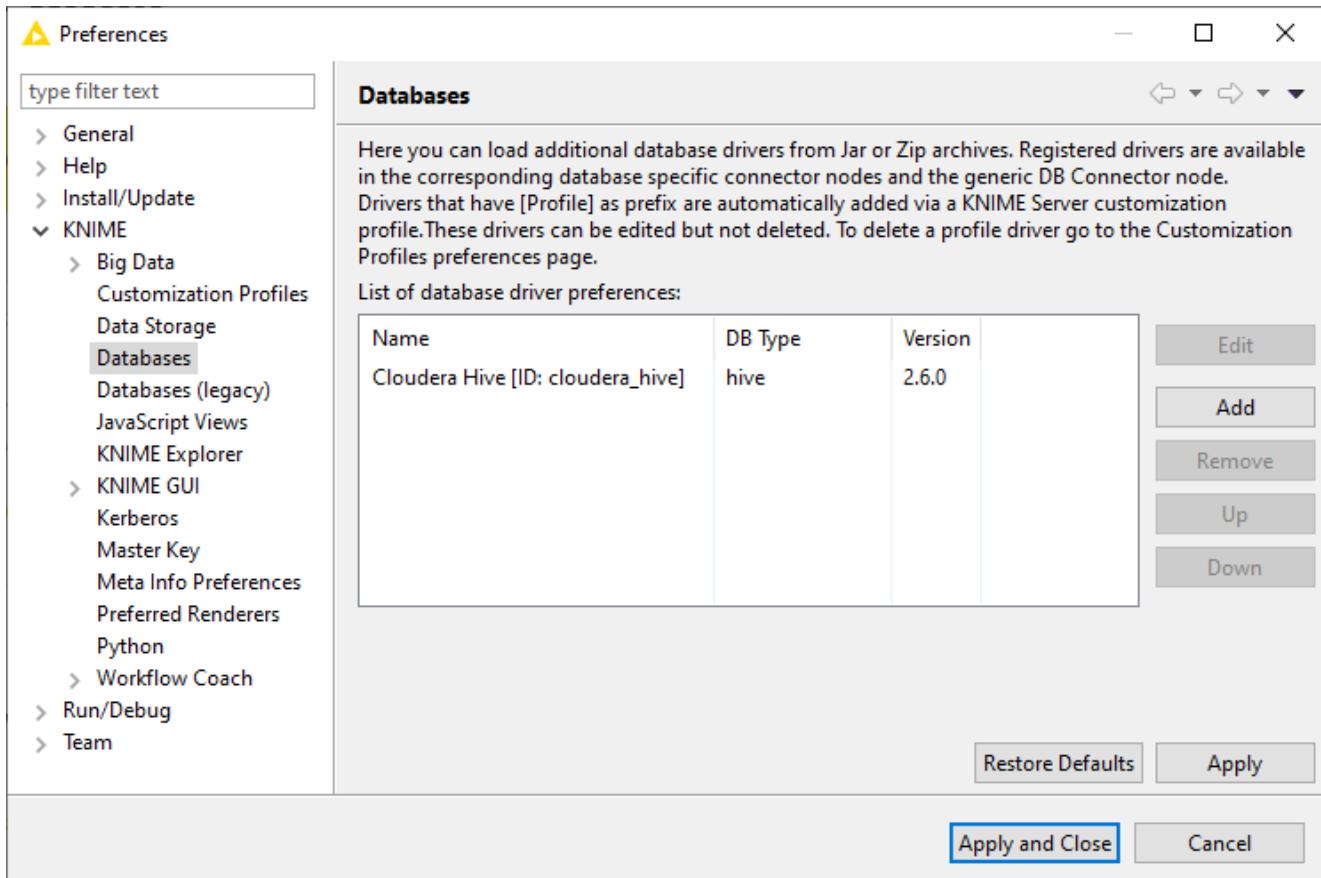


Abbildung 28. Datenbankeinstellungsseite

Hive hat einen dedizierten Connector-Knoten namens [Hive Connector](#page6), bitte verweisen über die Verbindung mit dedizierten Connector-Nodes. [vordefinierte Datenbanken](#)

Verbindung mit Apache ImpalaTM

Um Apache Impala zu verbinden, müssen Sie die [KNIME Big Data Connectors Erweiterung](#).

Die gewidmet Impala Connector node wird standardmäßig mit dem open-source Apache gebündelt. Hive JDBC Treiber, der mit Impala kompatibel ist. Auch proprietäre Treiber werden unterstützt, muss zuerst registriert werden, wie beispielsweise der von Cloudera bereitgestellte Impala JDBC Connector.

In diesem Beispiel wollen wir mit der proprietären Cloudera Impala JDBC eine Verbindung zu Impala herstellen

Fahrer. Der erste Schritt ist das Herunterladen der neuesten

[Impala JDBC Treiber für Cloudera Enterprise](#)

Dann gehen Sie zu Datei → Vorlieben

→ KNIME → Datenbanken , und klicken

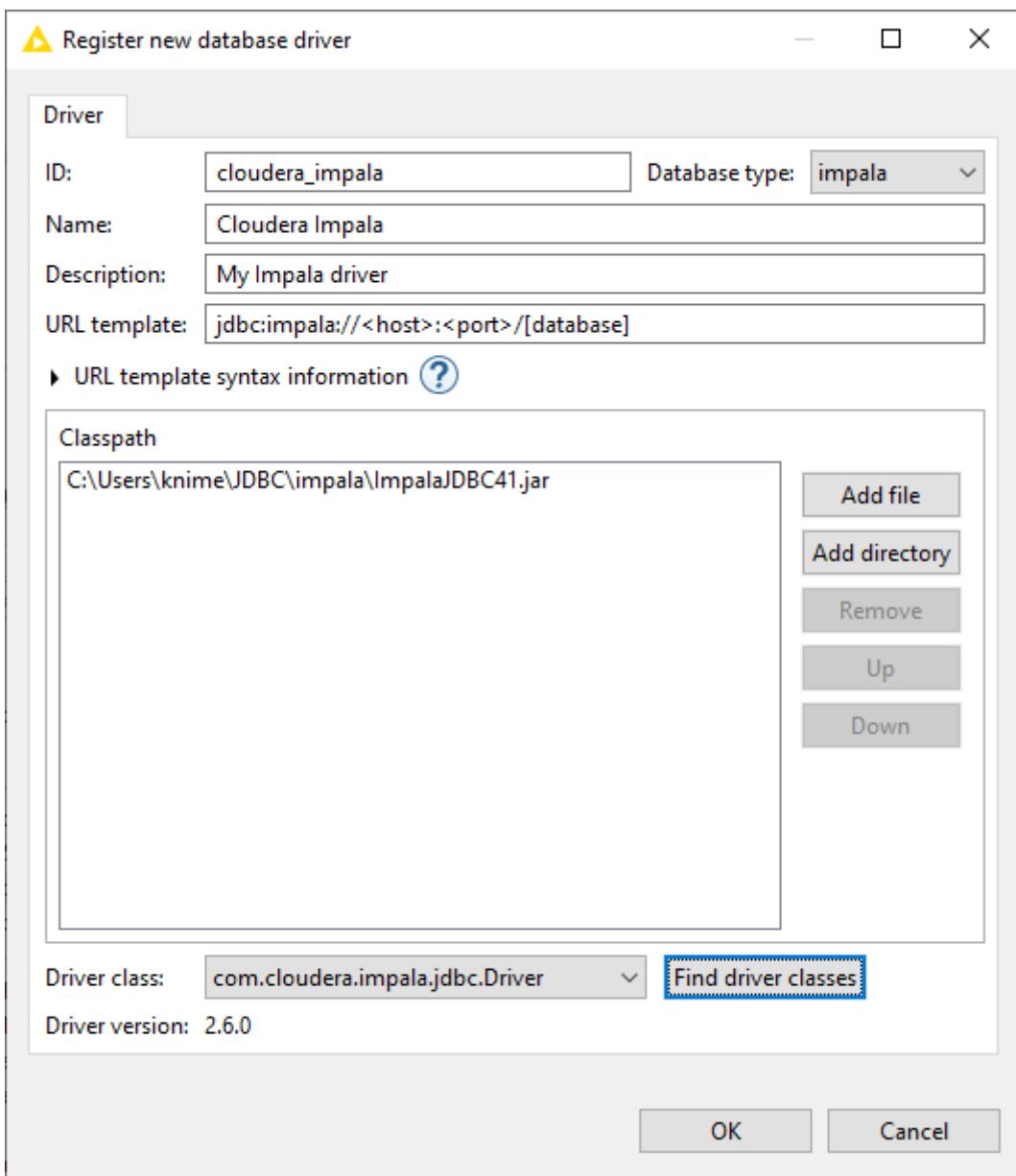


Abbildung 29. Registrieren Sie neuen Apache Impala Treiber

Geben Sie im neuen Datenbanktreiberfenster folgende Informationen an:

- ID : cloudera\_impala, aber Sie können Ihre eigene Treiber-ID eingeben, solange es nur enthält alphanumerische Zeichen und Unterpunkte.
- Name : Cloudera Impala, aber Sie können Ihren eigenen Treibernamen eingeben.
- Datenbanktyp : Impala ist in der Dropdown-Liste verfügbar, so dass der Datenbanktyp auf Impala .
- Warenbezeichnung Mein Impala-Treiber zum Beispiel.
- URL-Vorlage : Durch Auswahl Impala in der Datenbanktyp , die URL-Vorlage ist

automatisch auf die Standard-JDBC URL-Vorlage für Impala, d.h.

`jdbc:impala://:[database]` . Für weitere mögliche Vorlagen klicken Sie einfach auf der [„URL Vorlagen“](#page13) Seite direkt unten. Bitte beachten Sie die [Vorlage](#page15) Abschnitt für weitere Informationen zu den unterstützten Token, z.B. Host, Port und

Datenbank.

- Klasse : Klicken Sie auf Datei hinzufügen, um die .jar Datei, die die Impala JDBC Treiberdatei enthält.

Der Pfad zur Treiberdatei wird dann im Bereich Classpath angezeigt.

- Fahrerklasse : Anklicken Finden Sie Fahrerklassen automatisch alle verfügbaren JDBC erkennen Treiberklassen und Versionen, die in diesem Fall Com.cloudera.impala.jdbc.Driver

Endlich, klicken Sie Ok. und der neu hinzugefügte Treiber erscheint in den Einstellungen des Datenbanktreibers

Tisch. Klicken Sie auf Anwenden und schließen die Änderungen anzuwenden und Sie können die Verbindung zu Ihrem Impala beginnen

Datenbank.

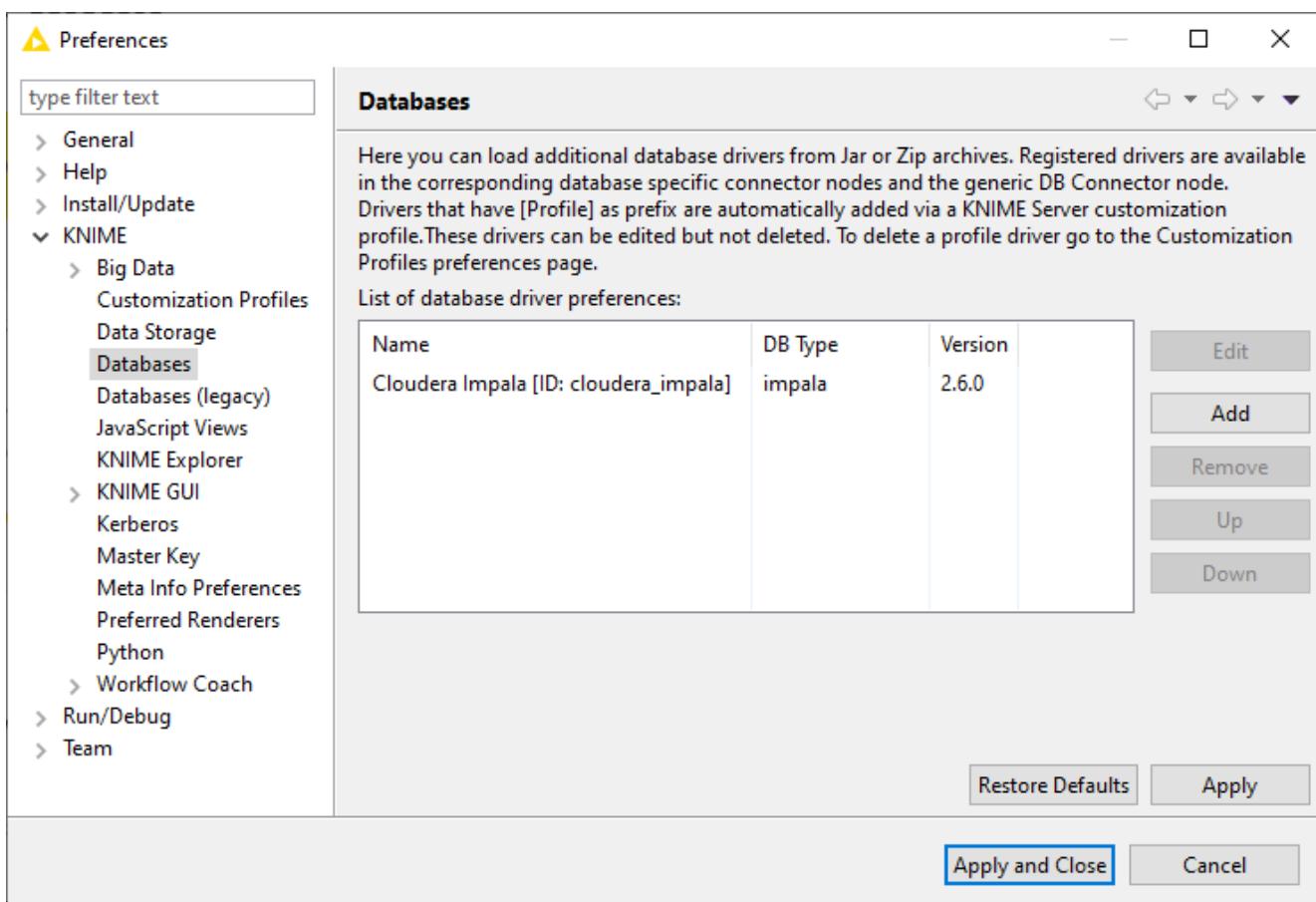


Abbildung 30. Datenbankeinstellungsseite

Impala hat einen dedizierten Connector-Knoten namens [Impala Connector](#page6), bitte verweisen über die Verbindung mit dedizierten Connector-Nodes. [zu vordefinierten Datenbanken](#page6)

## Lesen aus einer Datenbank

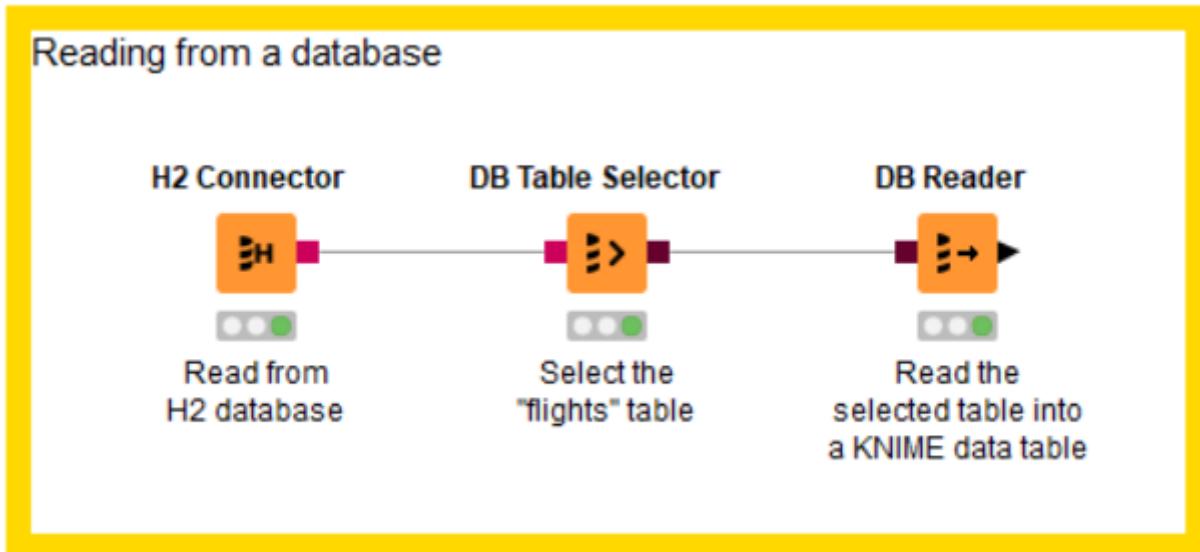


Abbildung 31. Auslesen aus einer Datenbank

Die obige Figur ist ein Beispiel dafür, wie man aus einer Datenbank liest. In diesem Beispiel wollen wir las die [Flüge](#) in einer H2-Datenbank gespeicherter Datensatz in eine KNIME-Datentabelle.

Zuerst benötigen Sie einen Verbindungsknoten, um eine Verbindung zur Datenbank herzustellen, im Beispiel darüber ist eine H2-Datenbank. Je nach

welche Datenbank wir verbinden wollen. Weitere Details zur Verbindung mit einer Datenbank [Verbindung mit einer Datenbank](#) auf die Abschnitt.

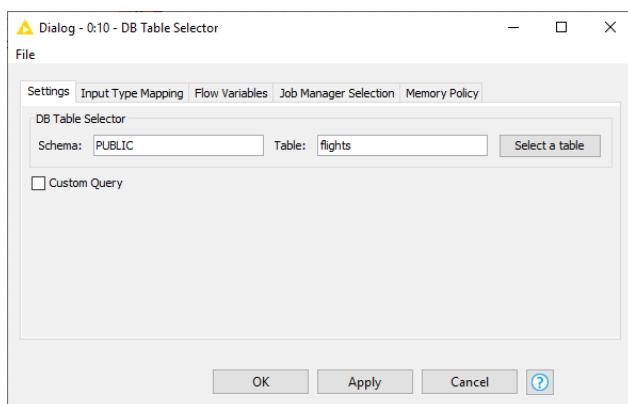


Abbildung 32. DB Tabellenauswahlkonfiguration

Presse Dialog Dialog Dialog  
Dialog Dialog Dialog

Wählen Sie eine Tabelle-Taste wird geöffnet  
verfügbare Tabellen/Ansichten in der Datenbank.

Darüber hinaus tickt die Individuelle Abfrage checkbox ermöglicht es Ihnen, Ihren eigenen benutzerdefinierten SQL zu schreiben

Nach dem Verbindungsaufbau wird die nächste Schritt ist, die DB Tabellenauswahl Knotenpunkt die die Auswahl einer Tabelle oder einer Ansicht ermöglicht interaktiv auf Basis der Eingabedatenbank Verbindung.

Die Figur auf der linken Seite zeigt die Konfigurationsdialog des DB Tabelle Auswählen Knoten. Am oberen Teil können Sie eingeben das Schema und der Name der Tabelle/Ansicht, Sie möchten wählen, in diesem Beispiel wollen wir um die "Flights"-Tabelle auszuwählen.

[Fenster, die Listen](#)

abfragen, um das Ergebnis zu verengen. Es akzeptiert jede SELECT-Erklärung, und der Platzhalter

#table# kann verwendet werden, um auf die über die

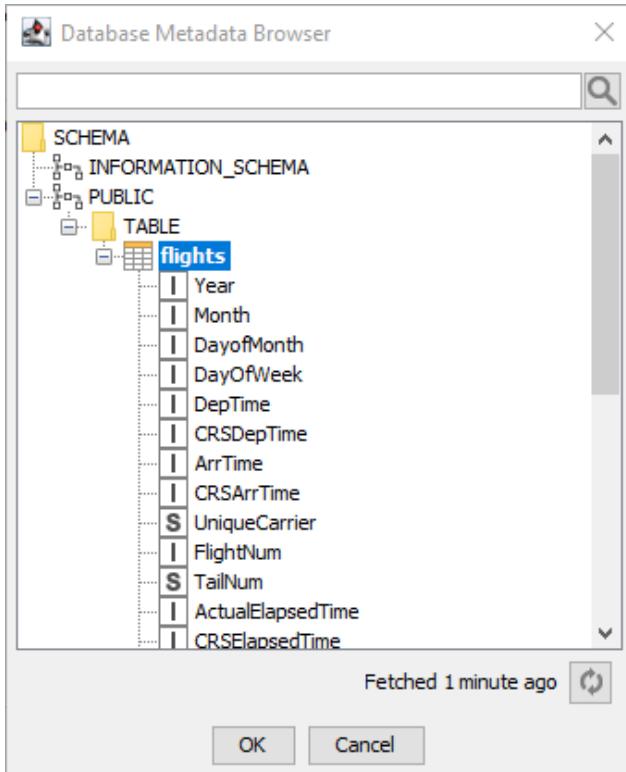
Wählen Sie eine Tabelle Knopf.

Die Eingangstyp Mapping tab ermöglicht es Ihnen, die Mapping-Regeln von Datenbanktypen zu definieren  
KNIME-Typen. Weitere Informationen hierzu finden Sie im Abschnitt [.](#page68)

[Hier geht es weiter](#page4) → **Ausgang:** dieser Knochen ist eine  
die Datenbankinformationen enthält und

die SQL-Abfrage automatisch durch den Rahmen erstellt, der die eingegebene Tabelle oder den Benutzer auswählt benutzerdefinierte Abfrage. Um die ausgewählte Tabelle oder Ansicht in die KNIME Analytics Platform zu lesen, Sie kann die DB Reader Knoten. Ausführen dieses Knotens führt die Eingabe SQL-Abfrage in der Datenbank und Ausgabe wird das in einer KNIME-Datentabelle gespeicherte Ergebnis sein, die gespeichert wird auf der Maschine, in der die KNIME Analytics Platform läuft.

# Datenbank Metadaten Browser



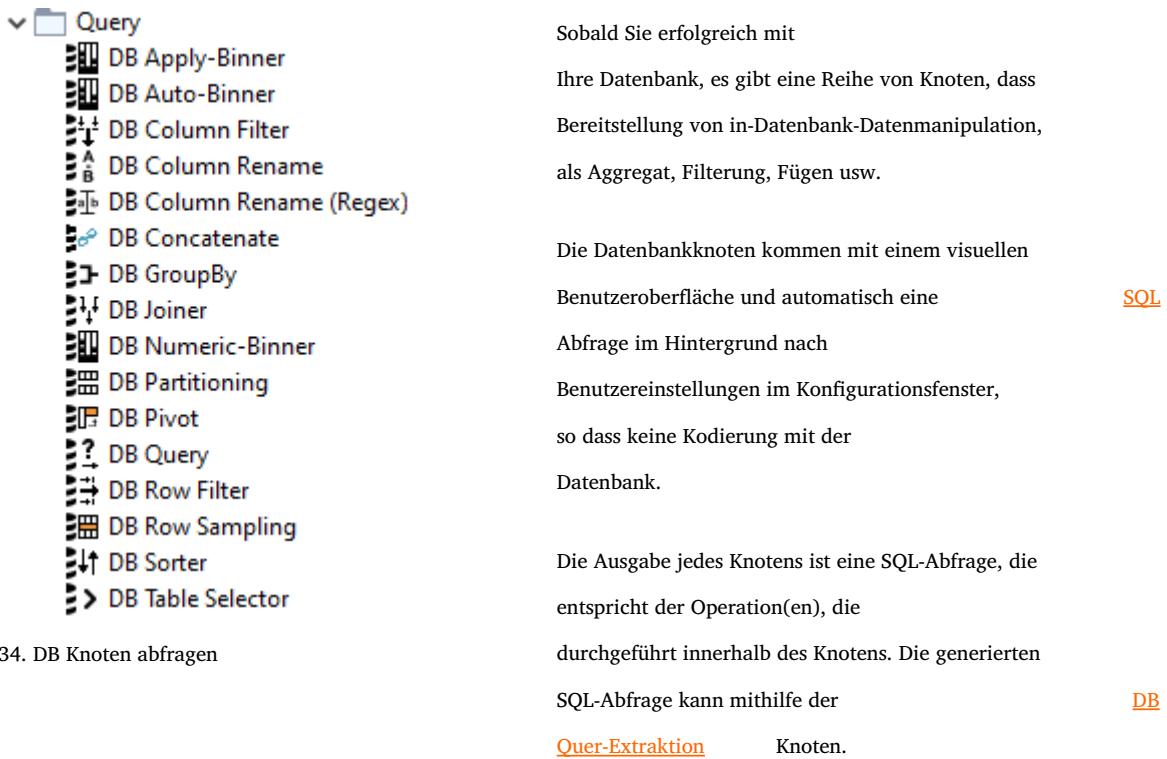
Die Datenbank Metadaten Browser zeigt  
Datenbankschema, einschließlich aller Tabellen /  
Ansichten und ihre entsprechenden Spalten und  
Spaltendatentypen. Am ersten Öffnen  
die Metadaten aus der Datenbank abrufen  
und kache es für die spätere Verwendung. von  
Klicken auf ein Element (Schema/Tabelle/Ansicht)  
es zeigt die enthaltenen Elemente. Zur Auswahl  
eine Tabelle oder Ansicht Wählen Sie den Namen und klicken Sie auf OK  
oder Doppelklicken Sie auf das Element.

Das Suchfeld oben am Fenster ermöglicht es Ihnen, nach jeder Tabelle oder Ansicht zu suchen in der Datenbank. Am unteren Ende befindet sich ein aktualisieren Sie die Schaltfläche, um die Schemaliste neu zu laden mit einer Zeit Referenz, wie lange vor der Das Schema wurde zuletzt aktualisiert.

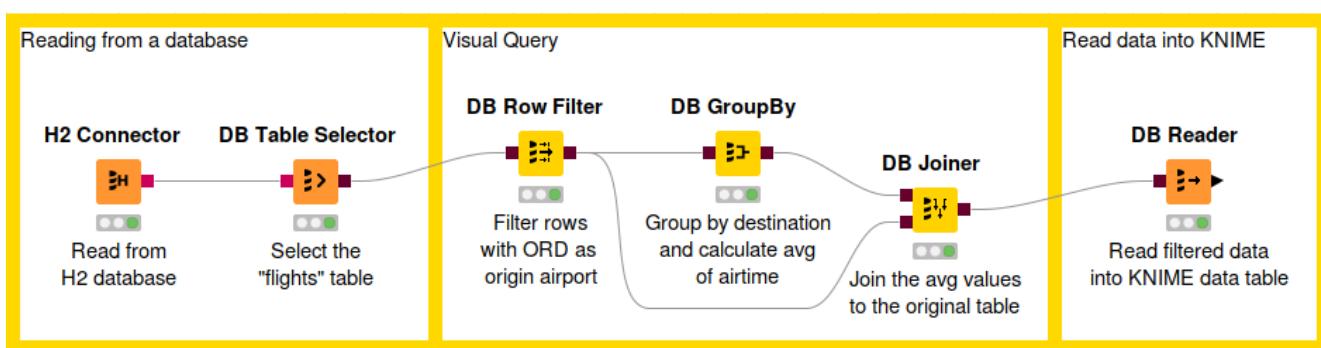
Abbildung 33. Datenbank Metadaten Browser

Wenn Sie gerade eine Tabelle erstellt haben und Sie sie nicht in der Schemaliste finden können, könnte es sei, dass der Metadaten-Browser-Cache nicht aktuell ist, also bitte versuchen, zu aktualisieren die Liste durch Anklicken der Refresh-Taste an der unteren rechten Ecke.

## Quer-Generation



## Visual Query Generation



Die obige Figur zeigt ein Beispiel für die in-Datenbank-Datenmanipulation. In diesem Beispiel haben wir las die **Flüge** Datensatz aus einer H2-Datenbank. Zuerst filtern wir die Zeilen, so dass wir nur die Flüge, die bestimmte Bedingungen erfüllen. Dann berechnen wir die durchschnittliche Luftzeit zu jedem einzelnen Flughafen. Schließlich verbinden wir die Durchschnittswerte zusammen mit den Originalwerten und dann das Ergebnis in die KNIME Analytics Platform lesen.

Der erste Schritt ist, die entsprechende Tabelle, die wir bearbeiten möchten mit.

## DB Row Filter

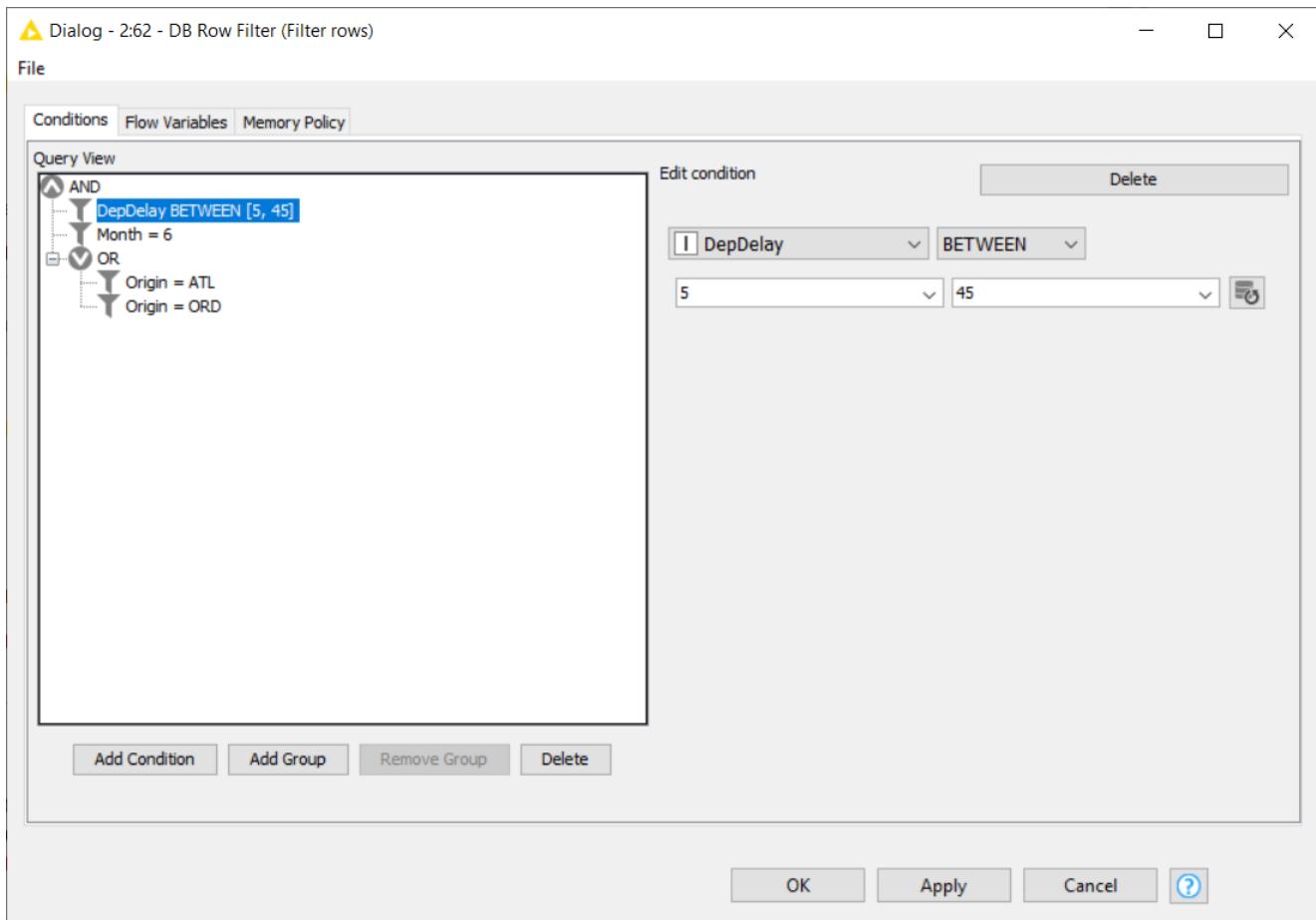


Abbildung 36. DB Row Filter Konfiguration Dialog

Nach der Auswahl der Tabelle können Sie mit den Daten arbeiten. Zuerst verwenden wir die

DB Row Filter

Knoten zu filtern Zeilen nach bestimmten Bedingungen. Die Abbildung oben zeigt die Konfiguration

Dialog des Knotens DB Row Filter . Auf der linken Seite gibt es einen Vorschaubereich, der alle Bedingungen von der Filter für die Eingabedaten. Filter können über logische Operatoren kombiniert und gruppiert werden wie UND oder OR. Nur Zeilen, die die angegebenen Filterbedingungen erfüllen, werden in den Ausgabedatentabelle. Am unteren Ende gibt es Optionen zu:

- In den Warenkorb : Fügen Sie mehr Zustand in die Liste
- Gruppe : Erstellen eines neuen logischen Operators (AND oder OR)
- Gruppe : Löschen des aktuell ausgewählten logischen Operators
- Löschen : Löschen Sie den ausgewählten Zustand aus der Liste

Um eine neue Bedingung zu erstellen, klicken Sie auf die Knopf. Um eine Bedingung zu bearbeiten, wählen Sie im Zustandsliste, die den ausgewählten Zustand im Zustands-Editor rechts zeigt. Die Editor besteht aus mindestens zwei Dropdown-Listen. Die linksste enthält die Spalten aus die Eingabedatentabelle und die darauf folgende enthält die mit der ausgewählter Spaltentyp, wie =, !=, <, >. Je nach gewähltem Betrieb ein Drittel und

Vielleicht wird das vierte Eingabefeld angezeigt, um die Filterwerte einzugeben oder auszuwählen. Die Schaltfläche nächstes die Wertefelder alle möglichen Werte für die ausgewählte Spalte, die dann zur Auswahl im Wertfeld erhältlich.

Durch Klicken auf einen logischen Operator in der Vorschauliste können Sie zwischen UND wechseln oder ODER und diesen Bediener durch Klicken zu löschen Gruppe .

Wie in unserem Beispiel wollen wir alle Zeilen zurückgeben, die die folgenden Bedingungen erfüllen:

- Originat vom Flughafen Chicago O'Hare (ORD) OR Hartsfield-Jackson Atlanta Flughafen (ATL)
- Im Monat Juni 2017 auftreten
- Sie haben eine milde Ankunftszeit zwischen 5 und 45 Minuten

#### DB GroupBy

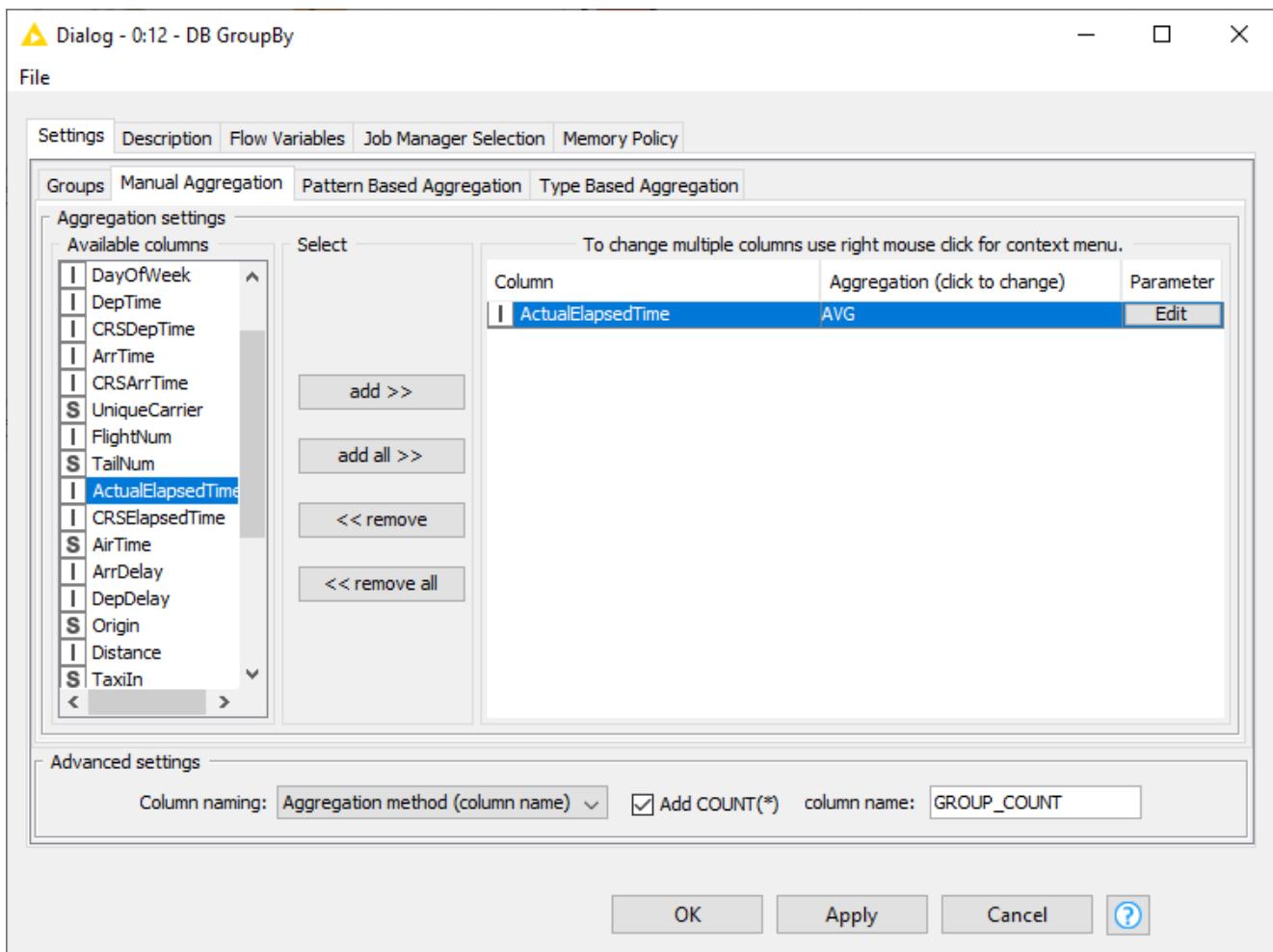


Abbildung 37. DB GroupBy: Manuelle Aggregation

Der nächste Schritt ist die Berechnung der durchschnittlichen Luftzeit zu jedem einzelnen Zielflughafen mit Hilfe der DB GroupBy Knoten. Um die Anzahl der Zeilen pro Gruppe abrufen Count(\*) hinzufügen Kontrollkästchen

die Erweiterte Einstellungen . Der Name der Gruppenzählsäule kann über das Ergebnis geändert werden Spaltenname Feld.

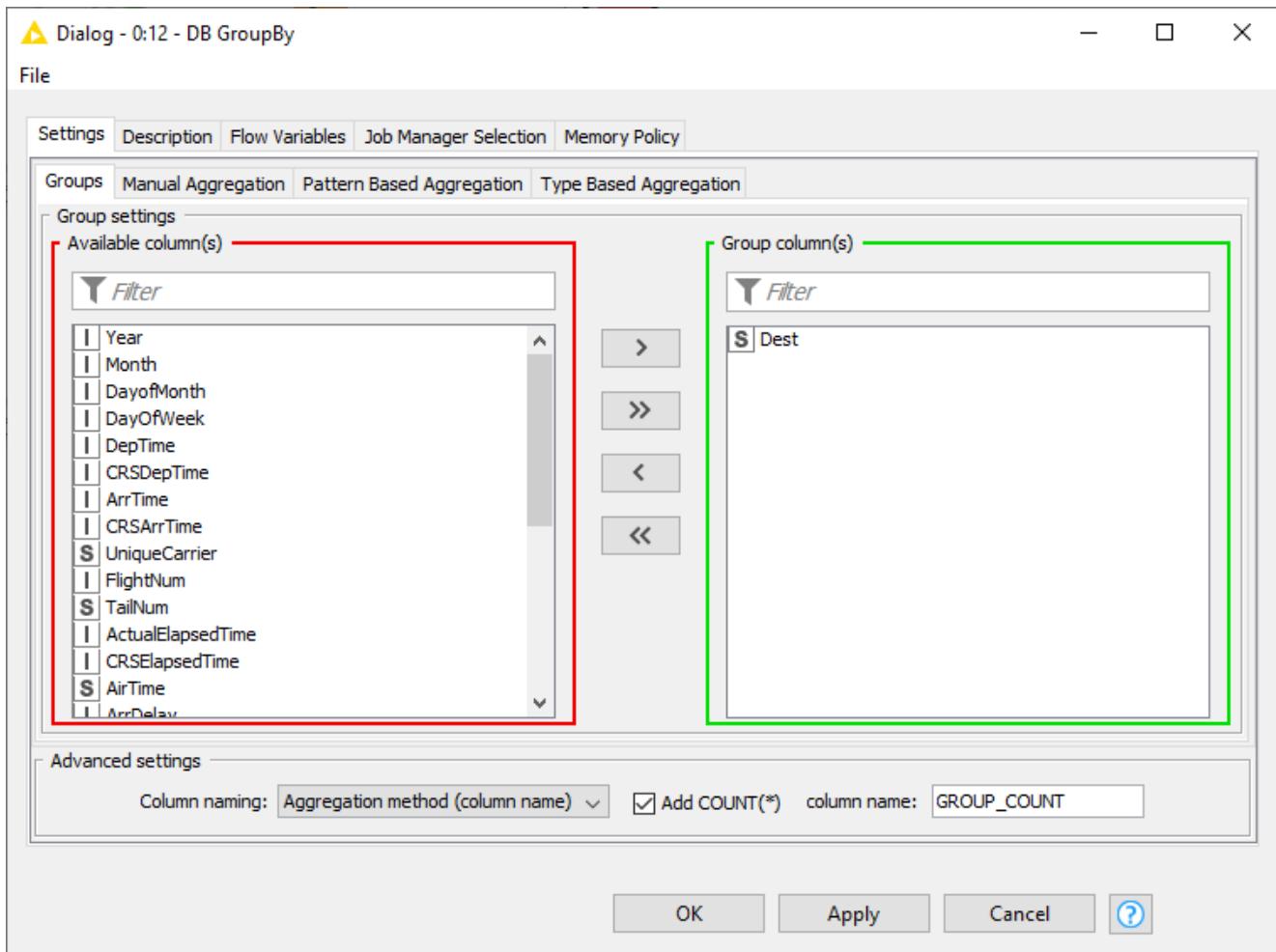


Abbildung 38. DB GroupBy: Gruppeneinstellungen

Um die durchschnittliche Luftzeit für jeden Zielflughafen zu berechnen, müssen wir von der Spalte in der Gruppen Tab, und in Manuelle Aggregation Tab wählen wir AktualElapsedTime Spalte (Luftzeit) und AVG als Aggregationsmethode.

DB Joiner

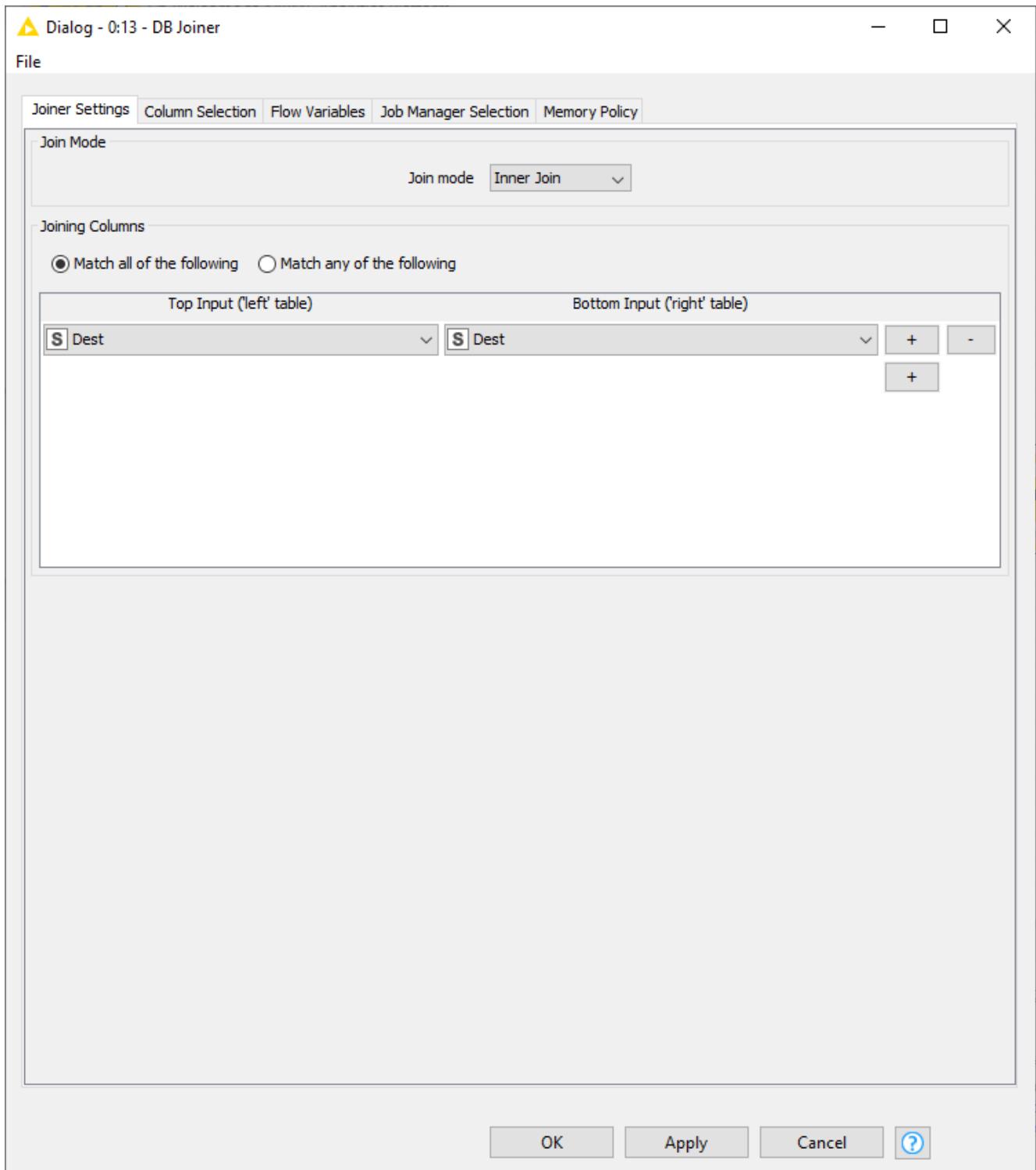


Abbildung 39. DB Joiner: Joiner Einstellungen

Um das Ergebnis zurück zu den ursprünglichen Daten zu verbinden, verwenden wir **DB Joiner** Knoten, der zwei Datenbanktabellen basierend auf dem Fügen von Spalte(n) beider Tabellen. In der **Mitglied Einstellungen** Tab, dort sind Optionen, um den Join-Modus zu wählen, ob Inner Join, Full Outer Join, etc. und das Fügen Spalte(n)

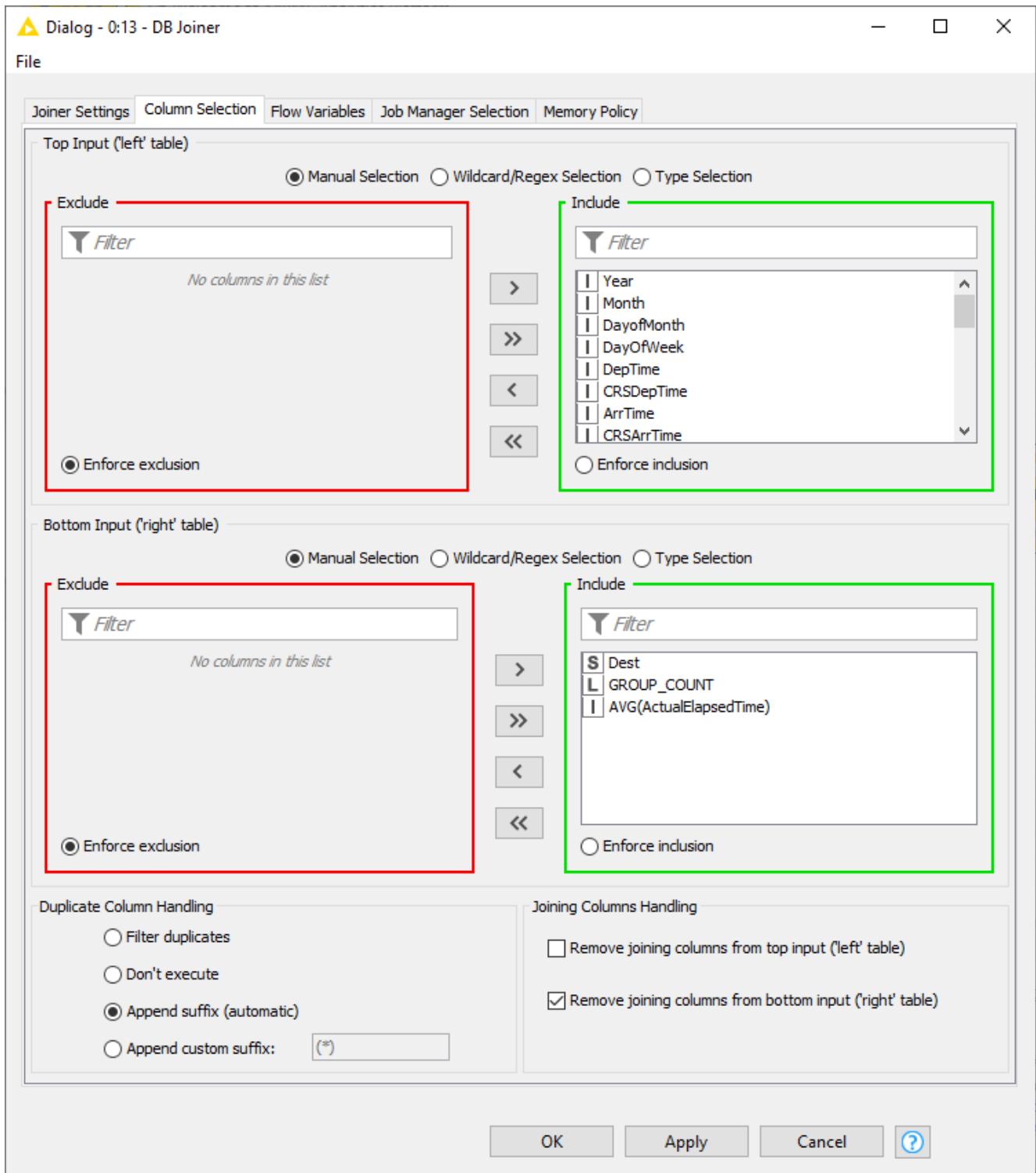


Abbildung 40. DB Joiner: Spalte Auswahl

In der **Auswahl der Spalte** Tab können Sie auswählen, welche Spalten aus jeder Tabelle Sie möchten in der Ausgabetafelle enthalten. Standardmäßig werden die Fügespalten von der unteren Eingabe nicht angezeigt in der Ausgabetafelle.

## Fortgeschrittenes Quer-Gebäude

Manchmal reicht die Verwendung der vordefinierten DB-Knoten zur Manipulation von Daten in der Datenbank nicht aus.

Dieser Abschnitt wird einige der DB-Knoten erklären, die es Benutzern ermöglichen, ihren eigenen SQL zu schreiben

Fragen, wie DB Abfrage , DB Query Reader , und Parametrized DB Mehr erfahren Knoten.

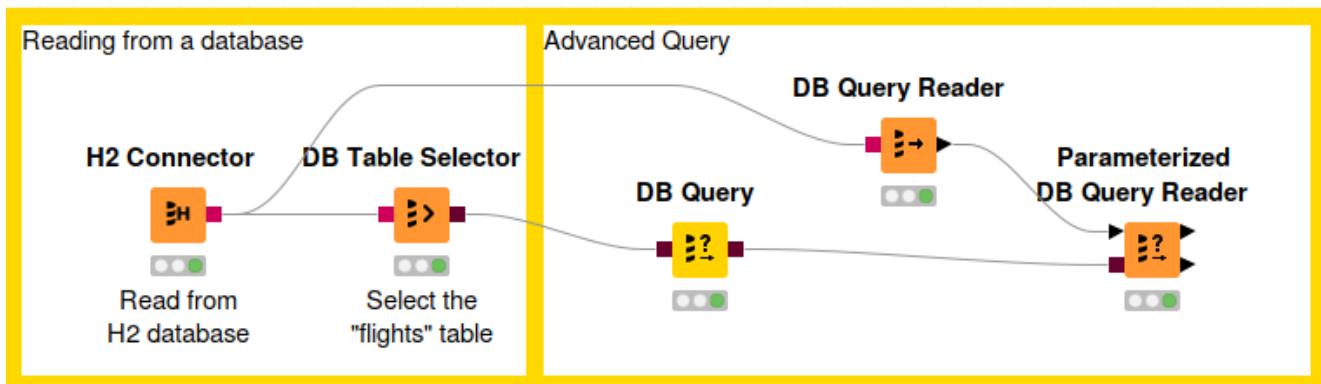


Abbildung 41. Beispiel-Workflow mit erweiterten Abfrageknoten

Jeder DB-Manipulationsknoten, der ein DB-Datenobjekt als Eingabe erhält und ein

DB-Datenobjekt als Ausgabe,wickelt die eingehende SQL-Abfrage in eine Sub-Quote um.

Einige Datenbanken unterstützen jedoch keine Subqueries, und wenn das der Fall ist,

Bitte benutzen Sie DB Query Reader um Daten aus der Datenbank zu lesen.

Die folgende Figur zeigt den Konfigurationsdialog der

DB Abfrage Knoten. Die Konfiguration

Dialog von anderen erweiterten Abfrageknoten, die es dem Benutzer erlauben, SQL-Anweisungen zu schreiben, geben einen ähnliche Benutzererfahrung. Es gibt einen Textbereich, um Ihre eigene SQL-Anweisung zu schreiben, die

Syntax-Herstellung und Code-Vervollständigung durch Treffer Ctrl + Space . Auf der Unterseite befindet sich eine

Bewertung Taste, wo Sie die SQL-Anweisung auswerten und die ersten 10 Zeilen der

Ergebnis. Wenn in der SQL-Anweisung ein Fehler vorliegt, wird eine Fehlermeldung in der

Bewerten Sie das Fenster. Auf der linken Seite ist die

[das Fenster](#page44)

Sie können die Datenbank-Metadaten wie Tabellen und Ansichten und deren

entsprechende Spalten. Die Datenbankliste enthält die Spalten, die in

die angeschlossene Datenbanktabelle. Doppelklicken eines der Elemente wird seinen Namen in der

aktuelle Cursorposition im SQL-Anweisungsbereich.

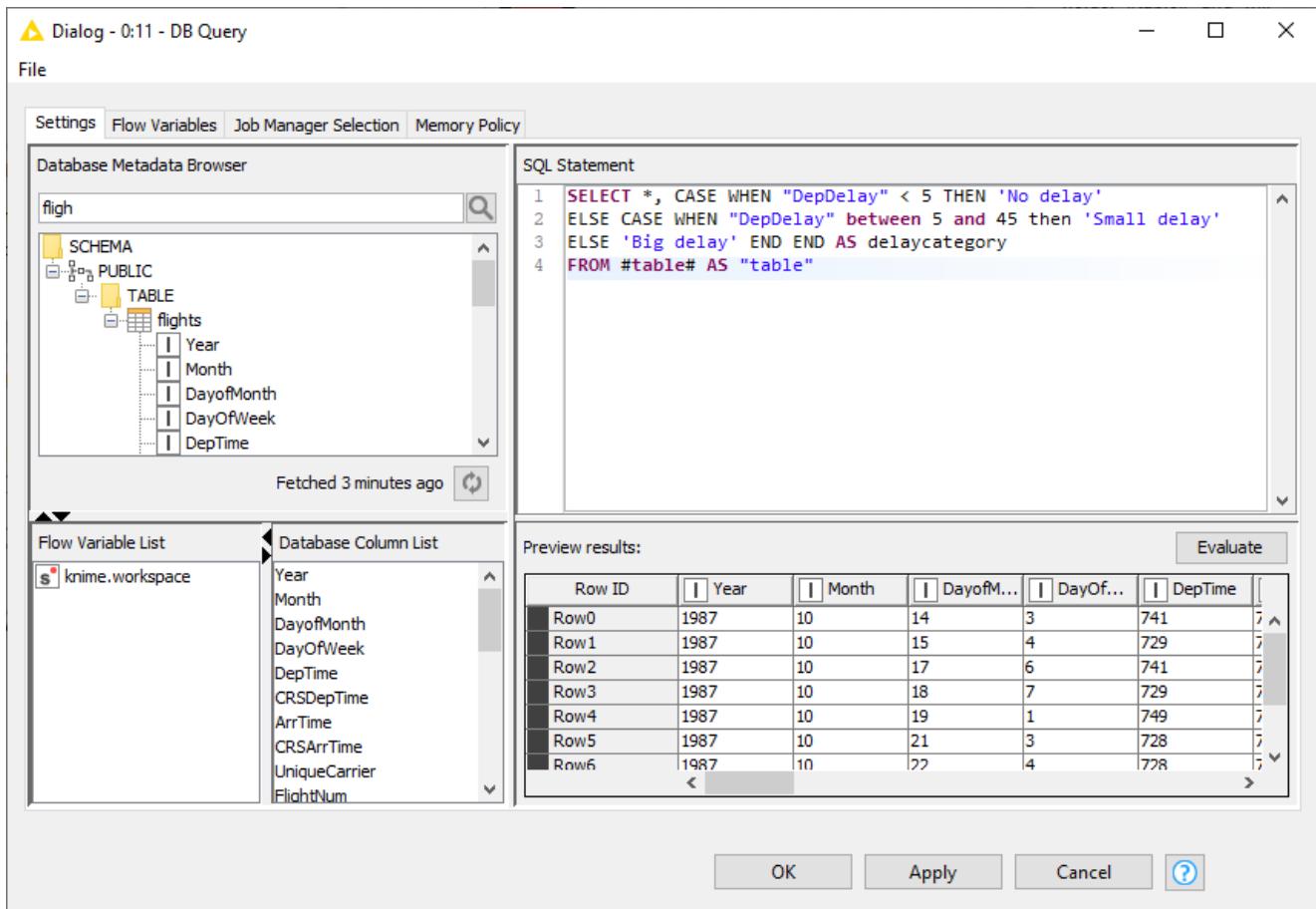


Abbildung 42. Konfigurationsdialog des DB Query-Knotens

## DB Abfrage

Die DB Abfrageknoten die Eingabe SQL-Abfrage von einer eingehenden Datenbankdaten modifiziert Verbindung. Die SQL-Abfrage vom Vorgänger wird durch den Platzhalter dargestellt und wird während der Ausführung ersetzt. Die geänderte Eingabeabfrage ist dann an der outport.

## DB Query Reader

Vollt eine eingegebene SQL-Abfrage aus und gibt das Ergebnis als KNIME-Datentabelle zurück. Dieser Knoten tut die Abfrage nicht ändern oder wickeln und unterstützt so alle Arten von Aussagen, die Daten zurückgeben.

Dieser Knoten unterstützt andere SQL-Anweisungen neben SELECT, wie TABELLE.

DESCRIBE

## Parameter DB Mehr erfahren

Mit diesem Knoten können Sie eine SQL-Abfrage mit verschiedenen Parametern ausführen. Es schleift über die Eingabe KNIME-Tabelle und nimmt die Werte aus der Eingabetabelle zur Parametrierung der Eingabe SQL abfragen. Da der Knoten eine KNIME-Datentabelleneingabe hat, stellt er eine Typ-Mapping-Tab zur Verfügung, die Sie können die Mapping-Regeln ändern. Für weitere Informationen über die Registerkarte Typ Mapping, [Bitte beachten Sie Typ Mapping](#page68) Abschnitt.

## DB Looping

Dieser Knoten führt SQL-Abfragen in der angeschlossenen Datenbank durch die möglichen Werte begrenzt durch die Eingabetabelle. Es begrenzt jede SQL-Abfrage, so dass nur Zeilen, die dem Möglichen entsprechen Werte aus der Eingabetabelle abgerufen werden, während die Anzahl der Werte pro Abfrage definiert. Dieser Knoten wird normalerweise verwendet, um auszuführen Abfragen z.

```
SELECT * FROM Tabelle  
IN DER ERWÄGUNG ($Col1_values$)
```

Während der Ausführung der Spaltenplatzhalter \$Col1\_values\$ wird durch ein Komma ersetzt getrennte Werteliste aus der Eingabetabelle. Da der Knoten eine KNIME-Datentabelle eingegeben hat stellt eine Typ-Mapping-Register zur Verfügung, mit der Sie die Mapping-Regeln ändern können. Für mehr Informationen über die Registerkarte Typ Mapping entnehmen Sie bitte der [Abschnitt.](#page68)

Ein Beispiel für die Verwendung dieses Knotens ist auf

[KNIME Hubraum](#).

## Datenbankstruktur Manipulation

Datenbankstruktur Manipulation bezieht sich auf jede Manipulation auf die Datenbanktabellen. Die

Folgender Workflow zeigt, wie eine bestehende Tabelle aus einer Datenbank mit der

DB Tischentferner

und eine neue Tabelle mit der

DB Table Creator

Knoten.

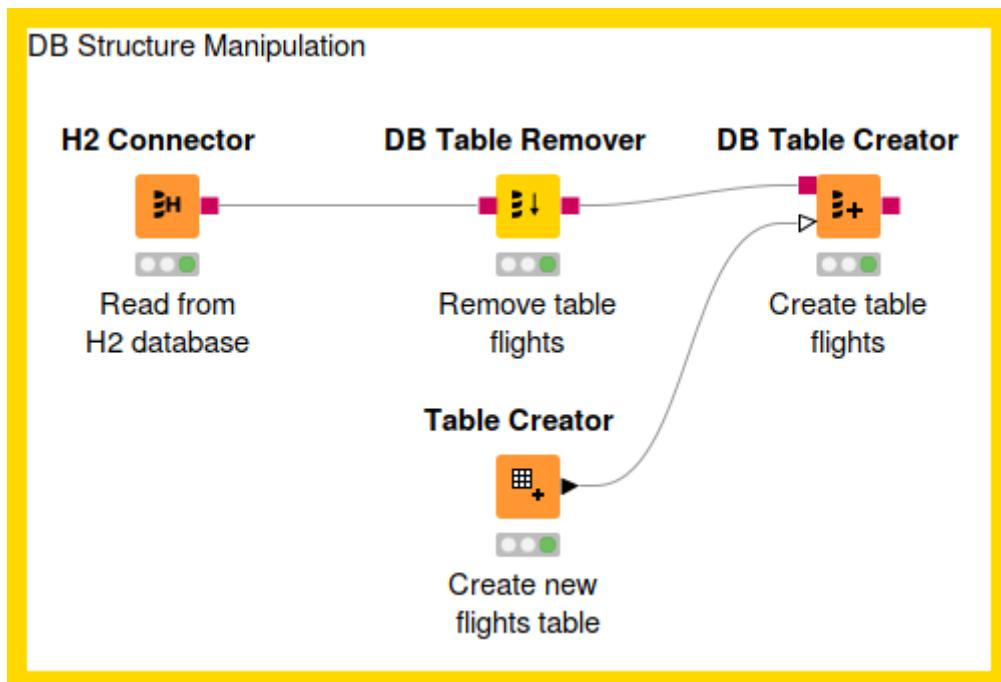


Abbildung 43. Beispiel eines Datenbankstruktur-Handhabungs-Workflows

DB Tischentferner

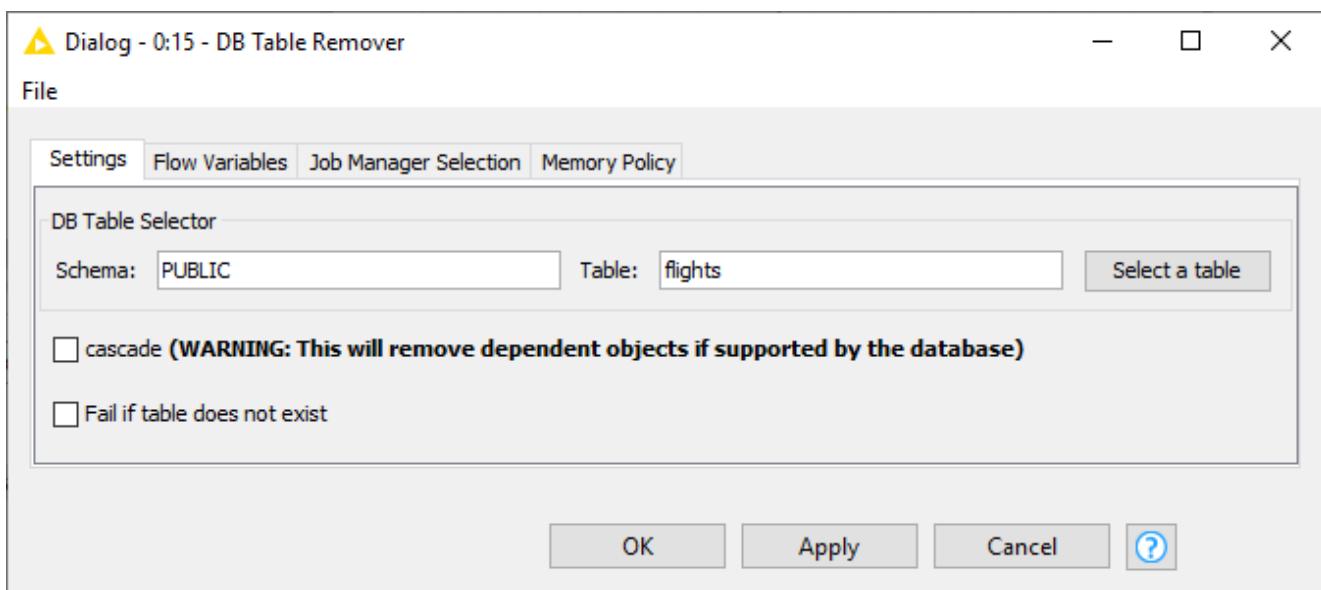


Abbildung 44. DB Tabelle Entferner Konfiguration Dialog

Dieser Knoten entfernt eine Tabelle aus der durch die eingehende Datenbankverbindung definierten Datenbank.

Die Ausführung dieses Knotens entspricht der Ausführung des SQL Befehls

DROP . In der Konfiguration

Dialog, es gibt eine Option, um die Datenbanktabelle zu entfernen. Die Konfiguration ist die

wie im DB Tabellenauswahl node, wo Sie das entsprechende Schema und

<a href="#page44" style="color: #0000ff; text-decoration: underline;">Tabellenentfernung und Tabellenanlegen unter einem Node

Im Konfigurationsfenster stehen folgende Optionen zur Verfügung:

Cascade : Auswählen dieser Option bedeutet, dass das Entfernen einer Tabelle, auf die andere verwiesen wird

Tabellen/Ansichten entfernen nicht nur die Tabelle selbst, sondern auch alle abhängigen Tabellen und Ansichten. wenn diese Option wird nicht von Ihrer Datenbank unterstützt, dann wird sie ignoriert.

Versagen, wenn Tabelle nicht existiert : Auswählen dieser Option bedeutet, dass der Knoten scheitert, wenn die gewählte Tabelle existiert nicht in der Datenbank. Standardmäßig ist diese Option nicht aktiviert, so dass der Knoten noch erfolgreich ausführen, auch wenn die ausgewählte Tabelle nicht in der Datenbank vorhanden ist.

## DB Table Creator

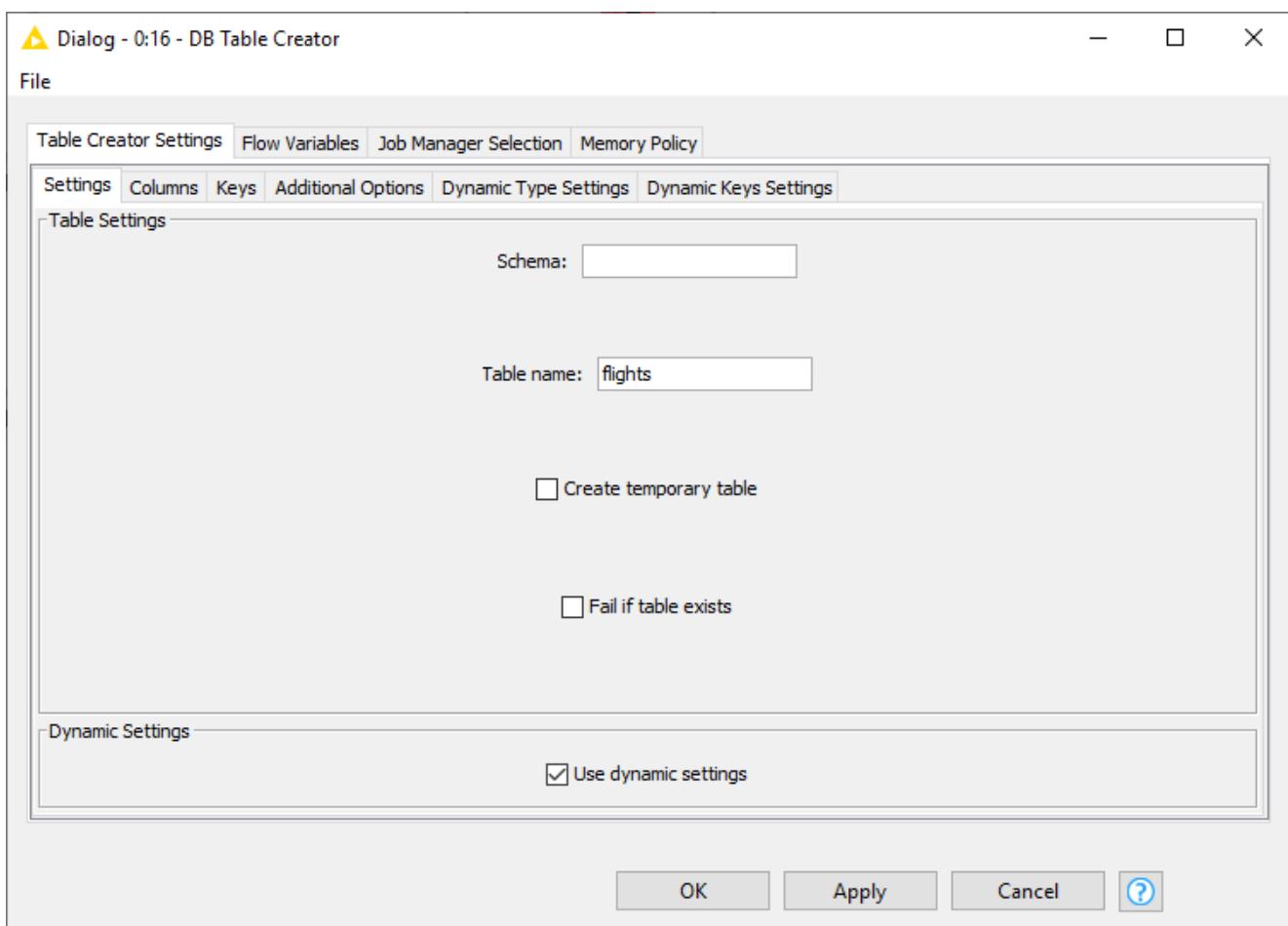


Abbildung 45. DB-Tabellen Creator: Einstellungen

Dieser Knoten erstellt eine neue Datenbanktabelle. Die Tabelle kann entweder manuell erstellt werden oder dynamisch basierend auf den Eingabedatentabellenspezifikationen. Es unterstützt erweiterte Optionen wie

Angabe, ob eine Spalte Nullwerte enthalten kann oder Primärschlüssel oder eindeutige Schlüssel als sowie der SQL-Typ.

Wenn Dynamische Einstellungen verwendet die Datenbanktabellenstruktur wird durch die Struktur der Eingabe-KNIME-Datentabelle. Die Spalten und SchlüsselTabs werden nur gelesen und nur helfen, die Struktur der Tabelle zu überprüfen, die erstellt wird. Die erstellte Datenbanktabelle Struktur kann durch Änderung der Typ-Mapping beeinflusst werden, z.B. indem definiert wird, dass KNIME-Doppel Spalten sollten als Stringspalten in die Datenbank geschrieben werden DB Table Creator wird Wählen Sie den String-Äquivalent-Datenbanktyp für alle Doppelspalten. Dieses Mapping und auch die Schlüsselerzeugung über die Dynamische Typeneinstellungen und Dynamischer Schlüssel EinstellungenTabs.

In der Registerkarte Einstellungen können Sie das entsprechende Schema und den Tabellennamen eingeben. Folgender Optionen sind verfügbar:

Zwischentabelle erstellen : Diese Auswahl wird eine temporäre Tabelle erstellen. Die Handhabung temporäre Tabellen, wie lange es existiert, der Umfang davon usw. hängt von der Datenbank ab, die Sie Verwendung. Bitte beachten Sie Ihren Datenbank-Anbieter für weitere Details dazu.

Fehler, wenn Tabelle existiert Auswählen wird der Knoten mit datenbankspezifischer Fehlermeldung ausfallen wenn die Tabelle bereits existiert. Standardmäßig ist diese Option deaktiviert, so dass der Knoten ausgeführt wird erfolgreich und keine Tabelle erstellen, wenn es bereits existierte.

Dynamische Einstellungen verwenden Das Auswählen ermöglicht es dem Knoten, die Struktur dynamisch zu definieren die Datenbanktabelle z.B. Spaltennamen und Typen basierend auf der Eingabe KNIME-Tabelle und der dynamische Einstellungen Tabs. Nur wenn diese Option aktiviert ist, wird Dynamische Typeneinstellungen und Dynamische Spalteneinstellungen Tab verfügbar. Die in der Namebasiertes SQL Typ Typ Mapping eine höhere Priorität als die in den KNIME-basiertes SQL Typ Mapping Wenn in beiden Registerkarten keine Mapping definiert ist, wird die Standard Mapping basierend auf der Es werden Definitionen des Datenbankanschlussknotens verwendet. Beachten Sie, dass während der dynamischen Einstellungen moderieren Spalten und SchlüsselTab wird nur lesen, um Ihnen eine Vorschau der Wirkung der dynamischen Einstellungen.

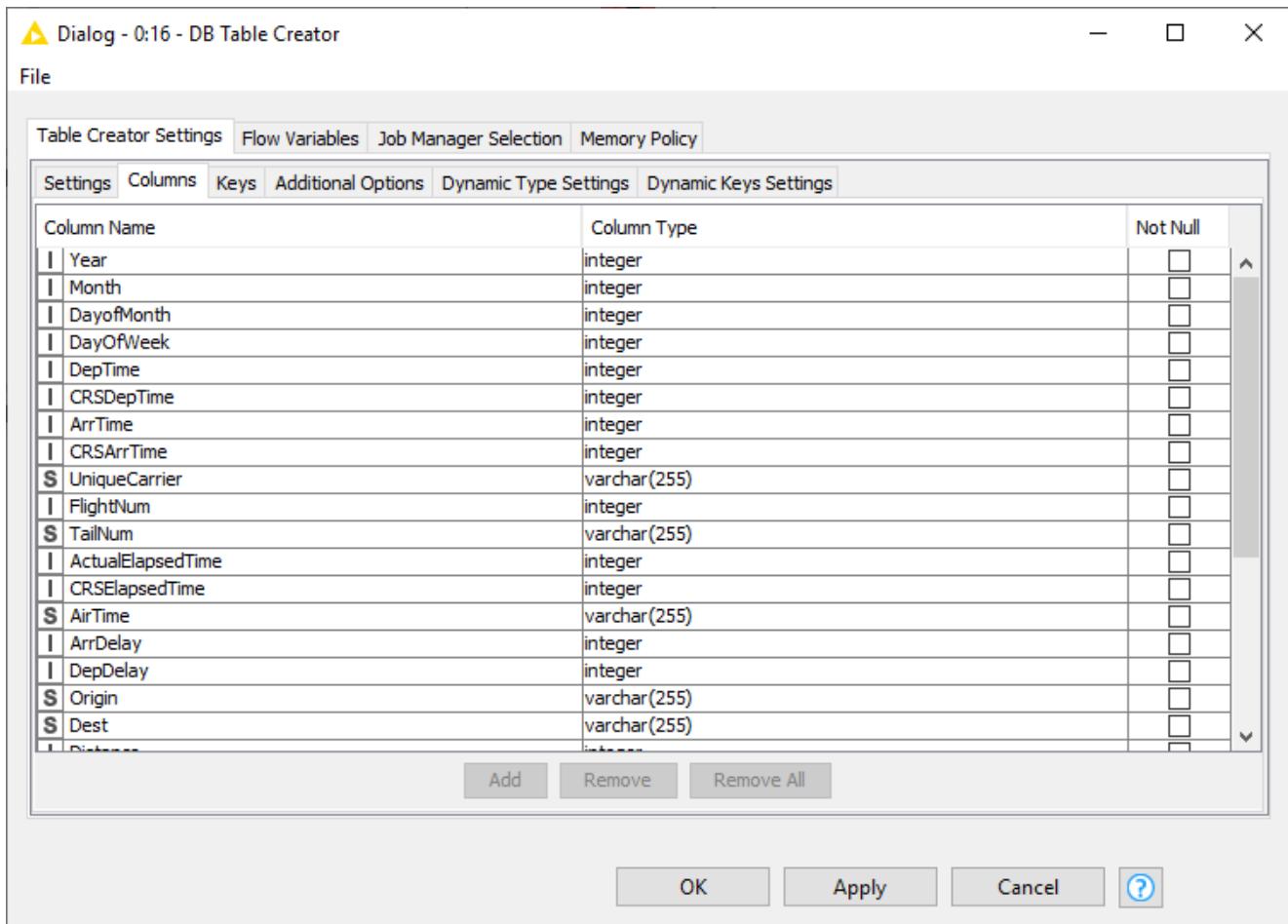


Abbildung 46. DB-Tabellenschöpfer: Säulen

In der Spalten Tab können Sie die Zuordnung zwischen den Spaltennamen vom Eingang ändern Tabelle und der entsprechende SQL-Typ manuell. Sie können Spalte hinzufügen oder entfernen und die gegebenenfalls SQL-Typ für eine bestimmte Spalte. Wenn Dynamische Einstellungen verwendet ausgewählt wird, Diese Registerkarte wird nur gelesen und dient als Vorschau der dynamischen Einstellungen.

In der **Schlüssel** Registerkarte können Sie bestimmte Spalten als primäre/unique Schlüssel manuell festlegen. Wie in der Spalten Tab, wenn die Dynamische Einstellungen verwendet aktiviert ist, wird diese Registerkarte eingelesen und dient als eine Vorschau der dynamischen Einstellungen.

In der **Zusätzliche Optionen** Tab können Sie zusätzliche SQL-Anweisung schreiben, die angehängt werden nach der CREATE-TABELLE-Anweisung, z.B. Speicherparameter. Diese Erklärung wird angefügt bis zum Ende der automatisch generierten CREATE-TABELLE-Anweisung und ausgeführt als ein einziges Statement.

In der **Dynamische Spalteneinstellungen** es gibt zwei Arten von SQL Type Mapping, der Name-Based und der KNIME-Based.

- In der **Namebasiertes SQL Typ Mapping** Sie definieren die standardmäßige SQL-Typ-Mapping für einen Satz von Spalten basierend auf den Spaltennamen. Sie können eine neue Reihe hinzufügen, die Namensmuster der Spalten, die abgebildet werden sollten. Das Namensmuster kann entweder ein

String mit Wildcard oder einem regelmäßigen Ausdruck. Die in der	Namebasiert
SQL Type Mapping                 eine höhere Priorität als die in den	KNIME-basiert
SQL Type Mapping                 .	

- In der KNIME-Typ-basiertes SQL Type Mapping                 Sie können den Standard SQL-Typ definieren  
Mapping basierend auf einem KNIME-Datentyp. Sie können eine neue Reihe mit dem KNIME hinzufügen  
Datentyp, der abgebildet werden sollte.

In der Dynamische Schlüsseleinstellungen Sie können die Schlüsseldefinitionen basierend auf der Spaltennamen. Sie können eine neue Zeile hinzufügen, die das Namensmuster der Spalten enthält, die sollte verwendet werden, um einen neuen Schlüssel zu definieren. Das Namensmuster kann entweder ein String mit Wildcard sein oder einen regelmäßigen Ausdruck.

Unterstützte Wildcards sind \* (passt jede Anzahl von Zeichen) und ? (Matches)  
ein Zeichen) z.B. KNI\* würde alle Strings entsprechen, die mit KNI beginnen, wie  
KNIME, während KNI? würde nur Strings entsprechen, die mit KNI beginnen und  
4. Charakter.

# DB Manipulation

Dieser Abschnitt beschreibt verschiedene DB-Knoten für In-Datenbank-Manipulation, wie DB Löschen (Tabelle) , DB Writer , DB-Einsatz , DB Update , DB Merge , und DB Loader Knoten, sowie der Datenbanktransaktionsknoten.

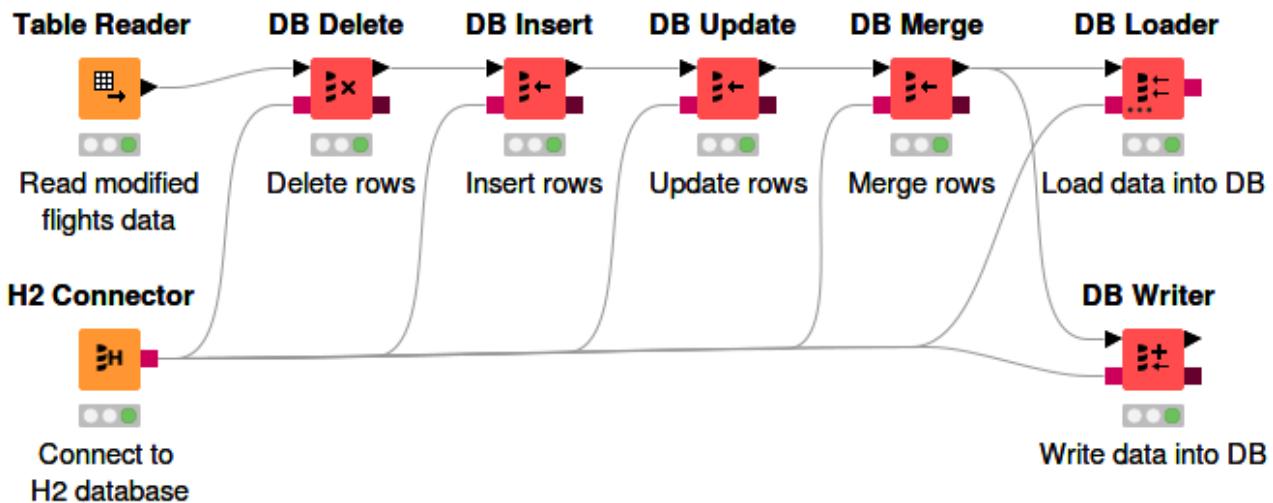


Abbildung 47. Beispiel der DB Manipulation

## DB Löschen

Die Datenbankerweiterung bietet zwei Knoten an [Löschen](#) Zeilen aus einer ausgewählten Tabelle in der Datenbank. Die [DB Löschen \(Tabelle\)](#) node löscht alle Zeilen aus der Datenbanktabelle, die übereinstimmen die Werte einer Eingabe-KNIME-Datentabelle, während [DB Löschen \(Filter\)](#) node löscht alle Zeilen aus der Datenbanktabelle, die den angegebenen Filterbedingungen entspricht.

### DB Löschen (Tabelle)

Dieser Knoten [Löschen](#) Zeilen aus einer ausgewählten Tabelle in der Datenbank. Der Eingang ist ein Port, der die Datenbank beschreibt, sowie eine KNIME-Datentabelle, die die Werte enthält, die definieren, welche Zeilen aus der Datenbank gelöscht werden. Es löscht Datenzeilen in der Datenbank basierend auf die ausgewählten Spalten aus der Eingabetabelle. Daher müssen alle ausgewählten Spaltennamen genau den Spaltennamen in der Datenbank entsprechen. Nur die Zeilen in der Datenbanktabelle die den Wertkombinationen der ausgewählten Spalten aus der KNIME-Eingabedatentabelle entsprechen wird gelöscht.

Die folgende Figur zeigt den Konfigurationsdialog der [DB Löschen \(Tabelle\)](#) Knoten. Die Konfigurationsdialog der anderen Knoten für die DB Manipulation ist sehr ähnlich. Sie können eingeben der Tabellenname und sein entsprechendes Schema oder den Tabellennamen in der

[<a href="#page3" style="color: orange;">#page3](#page3)

<#page44>

<a href="#page44" style="color: #ff6600; text-decoration: underline;">Metadaten Browser</a>  
durch Klicken Wählen Sie eine Tabelle

Zusätzlich müssen die Identifikationsspalten aus der Eingabetabelle ausgewählt werden. Die Namen die ausgewählten KNIME Tabellenspalten müssen den Namen in der ausgewählten Datenbanktabelle entsprechen.

Alle Zeilen in der Datenbanktabelle mit übereinstimmenden Werten für die ausgewählten Spalten aus dem Eingang

Die Datentabelle von KNIME wird gelöscht. In SQL entspricht dies dem **HINWEIS**Spalten. Es gibt drei Optionen:

- Fehler bei Fehlern : wenn ausgewählt, wird der Knoten scheitern, wenn bei der Ausführung Fehler auftreten sonst es wird erfolgreich ausgeführt, auch wenn eine der Eingabezeilen eine Ausnahme in Datenbank.
- Löschen von Statusspalten aktivieren : wenn ausgewählt, wird es zwei zusätzliche Spalten in der Ausgabe hinzufügen **DEUTSCHLAND**Erklärung.  
Tisch. Die erste Spalte enthält die Anzahl der von der Eine Zahl größer oder gleich Null gibt an, dass die Operation durchgeführt wurde erfolgreich. Ein Wert von -2 gibt an, dass die Operation erfolgreich durchgeführt wurde, aber Die Anzahl der betroffenen Zeilen ist unbekannt. Die zweite Spalte enthält eine Warnung Meldung, falls vorhanden.
- Deaktivieren Sie DB Datenausgabeport : Wenn ausgewählt, wird es den DB Data Ausgangsport und den Ausführung der Metadatenabfrage am Ende der Knotenausführung, die Probleme mit Datenbanken, die keine Subqueries unterstützen.

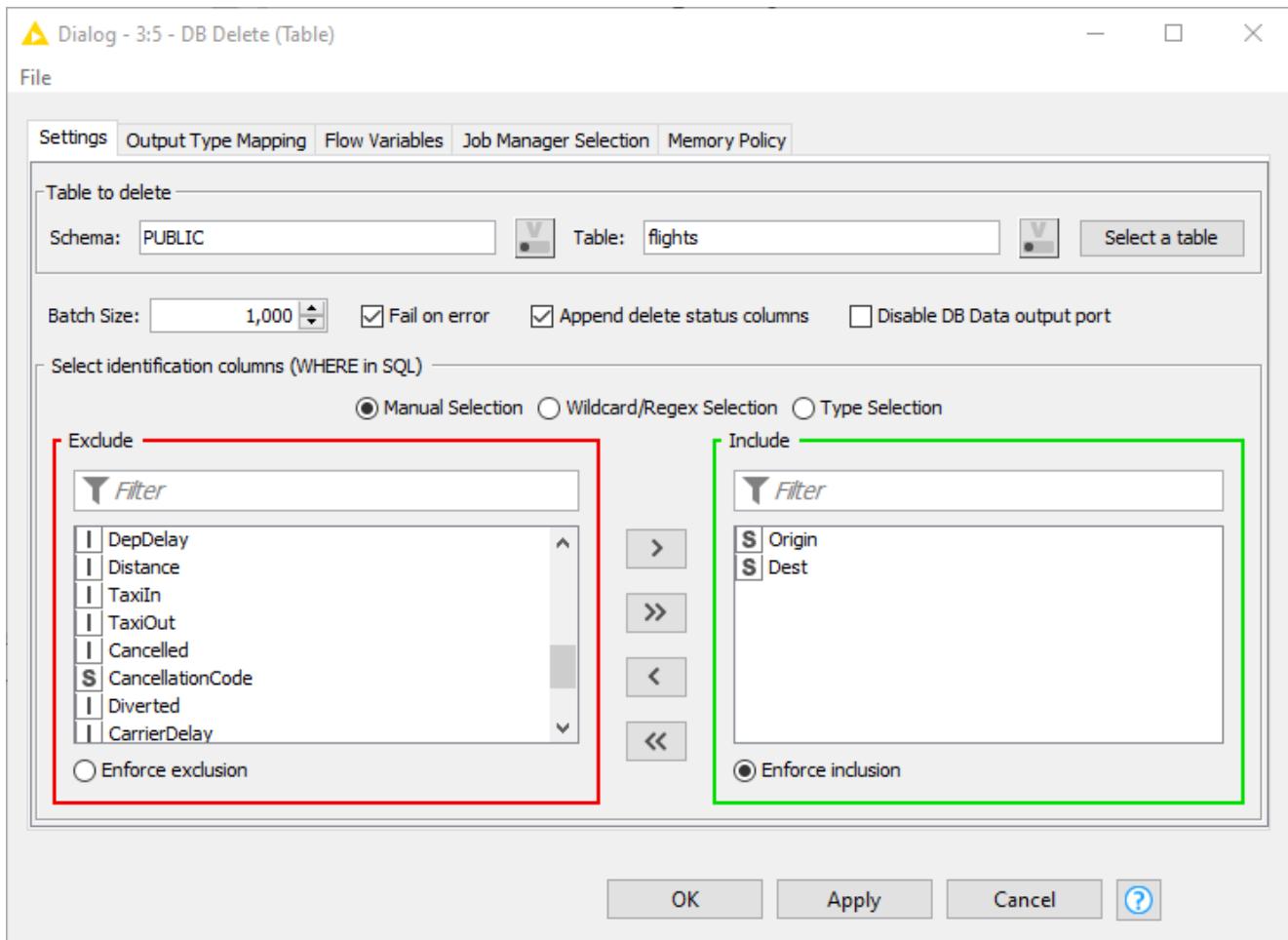


Abbildung 48. Konfigurationsdialog des DB Löschen (Tabellen) Knotens

Die Ausgangstyp Mapping tab ermöglicht es Ihnen, die Mapping-Regeln von KNIME-Typen zu definieren Datenbanktypen. Weitere Informationen hierzu finden Sie in der [Abschnitt.](#page68)

#### DB Löschen (Filter)

Dieser Knoten [Löschen](#) Zeilen aus einer ausgewählten Tabelle in der Datenbank, die dem angegebenen Filter entsprechen. [Bedingungen für Deletion](#) und [DB Verbindungen](#)

beide Fälle beschreibt der Eingangsport die Datenbank. Der Ausgangsport kann von einem [DB Verbindungen](#) oder [Daten](#) port. In

Im Knotendialog der [DB Löschen \(Filter\)](#) Knoten können Sie den Tabellennamen und dessen entsprechendes Schema oder wählen Sie den Tabellennamen in der [von](#)  
Klicken Sie auf Wählen Sie eine Tabelle Zusätzlich können Sie die Filterbedingungen festlegen, die verwendet werden, um die Zeilen in der Datenbanktabelle identifizieren, die gelöscht werden sollen. Die Bedingungen werden verwendet erzeugen [HINWEIS Klausel der DEUTSCHLAND](#) Erklärung. Alle Zeilen, die den Filterbedingungen entsprechen wird gelöscht.

Zum Beispiel werden die in der unten dargestellten Figur angegebenen Einstellungen alle Zeilen aus der Flüge Tabelle, die vor dem Jahr 2017 und das, wo weder Verschieden oder gekündigt .

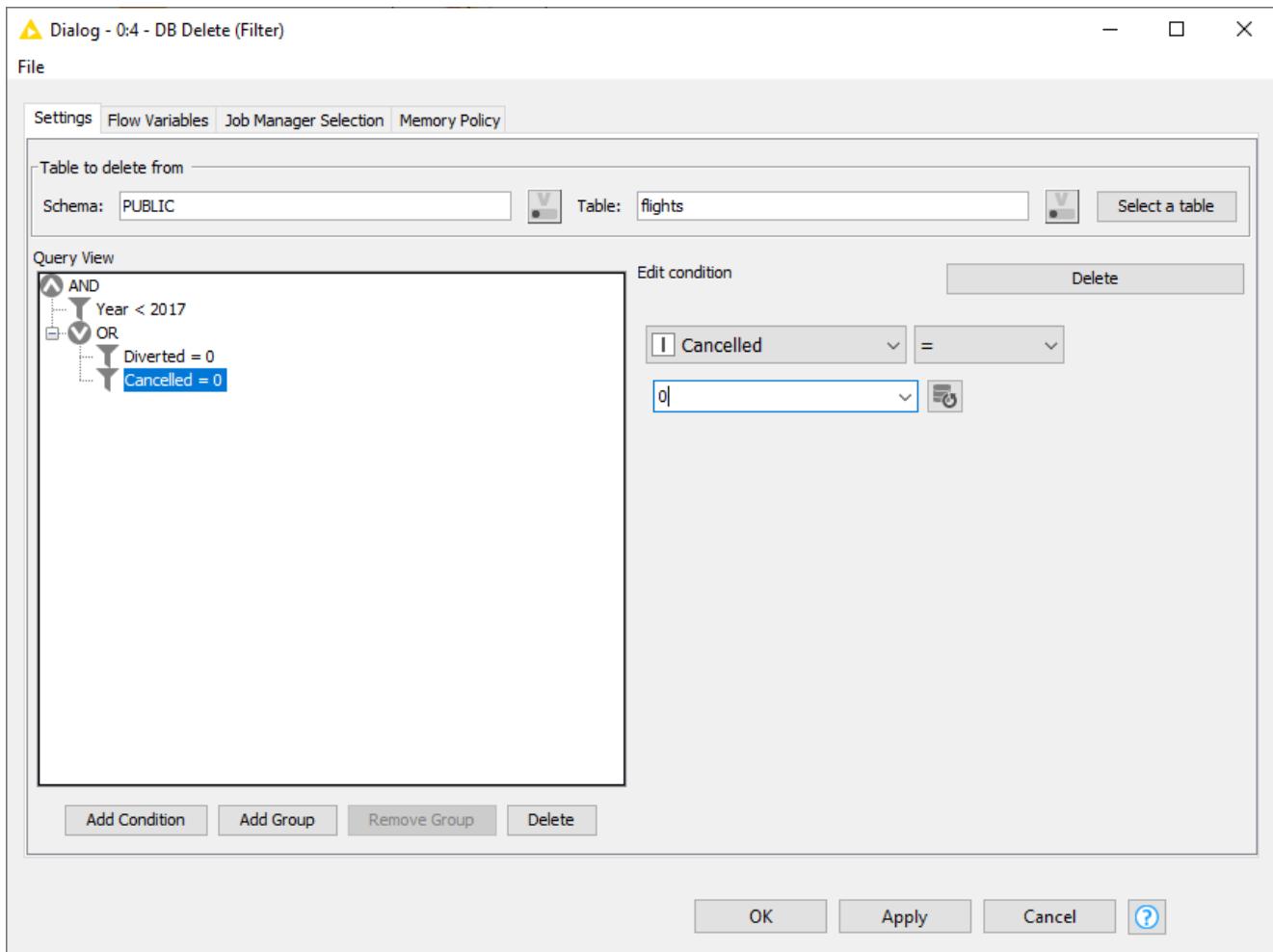


Abbildung 49. Konfigurationsdialog des DB Löschen (Filter)-Knotens

Ein Beispiel für die Verwendung dieses Knotens ist auf

[KNIME Hubraum](#).

## DB Writer

Dieser Knoten [Einsätze](#) die ausgewählten Werte aus der Eingabe-KNIME-Datentabelle in die angegebene Datenbanktabellen. Es erfüllt die gleiche Funktion wie die [DB Table Creator](#)-Knoten, aber zusätzlich auch erstellt die Datenbanktabelle automatisch, wenn sie vor dem Einfügen der Werte nicht existiert. Die neu erstellte Tabelle hat eine Spalte für jede ausgewählte Eingabe KNIME Spalte. Die Datenbank Spaltennamen werden die gleichen sein wie die Namen der Eingabe KNIME Spalten. Die Datenbank Säulentypen stammen aus den angegebenen KNIME-Typen und den [Konfiguration](#).

Alle Datenbankspalten erlauben fehlende Werte (z.

NULL )

Bitte benutzen Sie [DB Table Creator](#) node, wenn Sie die Eigenschaften des erstellten steuern möchten Datenbanktabelle.

Es besteht auch die Möglichkeit, eine bestehende Tabelle zu überschreiben, indem die Option aktiviert wird [Entfernen vorhanden](#) Tabelle im Konfigurationsfenster. Die Aktivierung der Option wird jede Tabelle mit dem angegebenen entfernen Name aus der Datenbank und dann ein neues erstellen. Wenn diese Option nicht gewählt wird, die neue

Datenzeilen werden an eine bestehende Tabelle angefügt. Sobald die Datenbanktabelle existiert, wird der Knoten alle KNIME-Eingangszeilen in die Datenbanktabelle in der gleichen Weise wie die

[<a href="#page63" style="color: #0000ff; text-decoration: none;">Knoten.](#page63)

## DB-Einsatz

Dieser Knoten [Einsätze](#) die ausgewählten Werte aus der Eingabe-KNIME-Datentabelle in die angegebene Datenbanktabellen. Alle ausgewählten Spaltennamen müssen genau den Spaltennamen innerhalb entsprechen die Datenbanktabelle.

## DB Update

Dieser Knoten [Aktualisierung](#) Zeilen in der angegebenen Datenbanktabelle mit Werten aus dem ausgewählten Spalten der Eingabe-KNIME-Datentabelle. Die Identifizierungsspalten werden in den [HINWEIS](#) Teil der SQL-Anweisung und identifizieren Sie die Zeilen in der Datenbanktabelle, die aktualisiert werden. Die In der [SET](#) Teil der SQL-Anweisung und enthalten die Werte, die wird in der ausgewählten Datenbanktabelle in die passenden Zeilen geschrieben.

## DB Merge

Die DB Merge Knoten ist eine Kombination der DB Update und DB-Einsatz Knoten. Wenn die Datenbank unterstützt die Funktionalität, die er ausführt [MERGE](#) Erklärung, dass alle neuen Zeilen oder Updates einfügen alle vorhandenen Zeilen in der ausgewählten Datenbanktabelle. Wenn die Datenbank die Zusammenführung nicht unterstützt Funktion der Knoten versucht zunächst alle Zeilen in der Datenbanktabelle zu aktualisieren und dann alle Zeilen einfügen wo während des Updates keine Übereinstimmung gefunden wurde. Die Namen der ausgewählten KNIME-Tabelle Spalten müssen den Namen der Datenbanktabelle entsprechen, in der die Zeilen aktualisiert werden sollen.

## DB Loader

Ab 4.3 nutzt der DB Loader-Knoten die neue Dateihandlung  
□ Rahmen, der eine nahtlose Migration zwischen verschiedenen Dateisystemen ermöglicht. Für mehr Details, bitte check out the [KNIME Leitfaden für die Bearbeitung von Dateien](#)

Dieser Knoten führt datenbankspezifische Bulkload-Funktionalität aus, die nur einige Datenbanken (z.B. Hive, Impala, MySQL, PostgreSQL und H2) unterstützen, große Datenmengen in eine bestehende Datenbanktabelle.

Die meisten Datenbanken führen keine Datenüberprüfungen durch, wenn die Daten in die Tabelle, die zu einer korrupten Datentabelle führen könnte. Der Knoten führt einige Vorabkontrollen wie Überprüfung der Spaltenfolge und Spaltennamen der Eingabe-KNIME-Datentabelle sind mit der ausgewählten Datenbanktabelle kompatibel. Es überprüft jedoch nicht die Spaltentypkompatibilität oder die Werte selbst. Bitte stellen Sie sicher, dass die Spaltentypen und Werte der KNIME-Tabelle kompatibel mit der Datenbanktabelle.

In Abhängigkeit von der Datenbank wird häufig ein Zwischendateiformat (z.B. CSV, Parkett, ORC) verwendet. für Effizienz, die erforderlich sein könnte, um die Datei auf einen Server hochzuladen. Wenn eine Datei sein muss hochgeladen, kann jedes der Protokolle, die von den Dateiverwaltungsknoten und der Datenbank unterstützt werden verwendet, z.B. SSH/SCP oder FTP. Nach dem Laden der Daten in eine Tabelle erhält die hochgeladene Datei gelöscht, wenn die Datenbank nicht mehr benötigt wird. Wenn die Datei nicht hochgeladen oder gespeichert werden muss aus irgendeinem Grund verhindert eine Dateiverbindung die Ausführung.

Einige Datenbanken wie MySQL und PostgreSQL unterstützen datenbasierte und speicherbasierte Hochladen, die unterschiedliche Rechte in der Datenbank erfordern. Zum Beispiel, wenn Sie nicht die Rechte, die datenbasierte Belastung der Daten auszuführen, versuchen stattdessen die speicherbasierte Methode.

Wenn die Datenbank verschiedene Lademethoden unterstützt (Datei- oder Speicher-  
Sie können die Methode in der Optionen Tab, wie im Beispiel dargestellt  
unten. Andernfalls Lademodus Option wird nicht in der Konfiguration angezeigt  
Dialog.

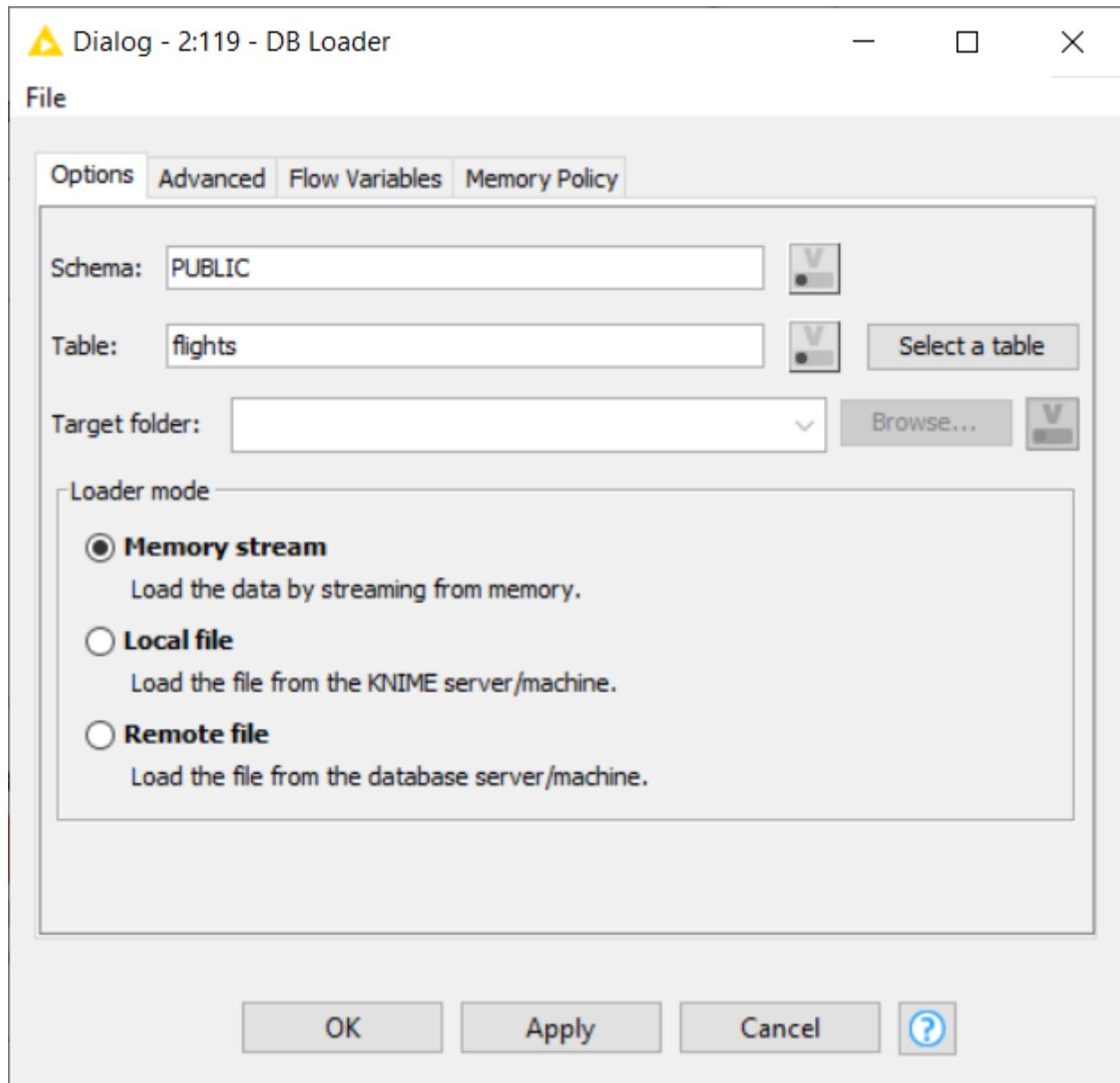


Abbildung 50. DB Loader: Option

Je nach angeschlossener Datenbank können sich die Dialogeinstellungen ändern. Zum Beispiel MySQL und PostgreSQL verwenden eine CSV-Datei für die Datenübertragung. Um zu ändern, wie die CSV-Datei ist schuf es zu den Erweiterte Tab.

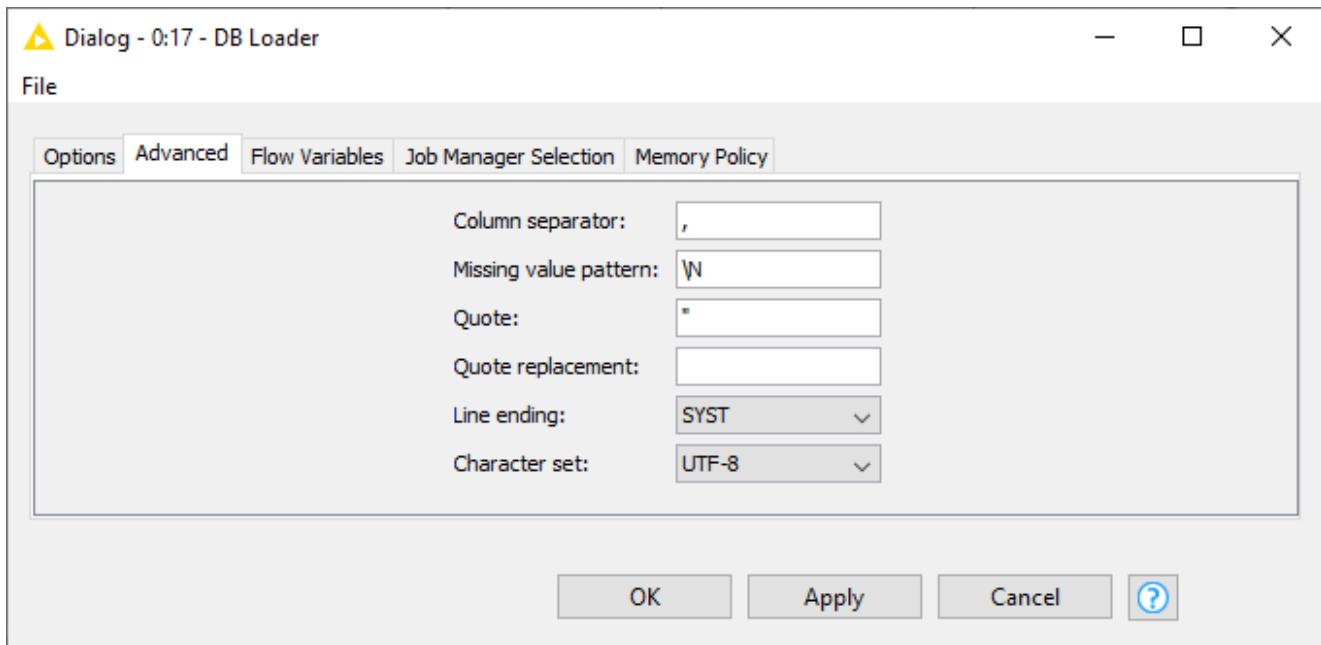


Abbildung 51. DB Loader: Fortgeschritten

## DB Transaction Nodes

Die Datenbankerweiterung bietet auch Knoten zur Simulation von Datenbanken [Transaktion](#). Eine Transaktion ermöglicht es Ihnen, mehrere Datenbank-Daten-Manipulationsoperationen in eine einzige Arbeitseinheit zu gruppieren. Diese Einheit führt entweder ganz oder gar nicht aus.

### DB Transaction Start

Die [DB Transaction Start](#) node startet eine Transaktion mit der Eingabedatenbankverbindung. Wie Solange die Transaktion abgeschlossen ist, kann die Eingabedatenbankverbindung nicht außerhalb verwendet werden [der Transaktion. Je nach Isolationsgrad](#), andere Verbindungen können keine sehen Änderungen in der Datenbank während der Transaktion. Die Transaktion verwendet den Standard Isolationsgrad der angeschlossenen Datenbank.

### DB Transaction End

Die [DB Transaction End](#) node beendet die Transaktion der Eingabedatenbankverbindung. Die [Knoten beendet die Transaktion mit einem bereinigt](#) die alle Änderungen für andere Benutzer sichtbar macht, erfolgreich ausgeführt. Ansonsten beendet der Knoten die Transaktion mit einem [Zurück zur Übersicht](#) zurück zur Datenbank zum Zustand zu Beginn der Transaktion.

Dieser Knoten hat 2 Eingangssports. Der erste ist der Transaktions-DB-Verbindungsport, der vom Ende der Transaktionskette zu verbinden. Der zweite Hafen sollte die Transaktions-DB-Verbindung vom Ausgang der

[DB Transaction Start](#) Knoten. wenn

die Transaktion erfolgreich ist und ein Commit ausgeführt wird, die DB-Verbindung von der ersten Eingabeport wird an den Ausgangsport weitergeleitet, andernfalls bei einem Rollback, die DB eine Verbindung vom zweiten Eingangsport wird weitergeleitet.

Die folgende Figur zeigt ein Beispiel der Transaktionsknoten. In diesem Beispiel Transaktion besteht aus zwei DB Writer-Knoten, die Daten auf dieselbe Tabelle nacheinander schreiben. Wenn während eines beliebigen Schrittes des Schreibens ein Fehler auftritt, werden die Änderungen nicht ausgeführt und die Datenbank wird zu Beginn der Transaktion in den vorherigen Zustand zurückgegeben. Wenn kein Fehler die Änderungen in der Datenbank erfolgen.

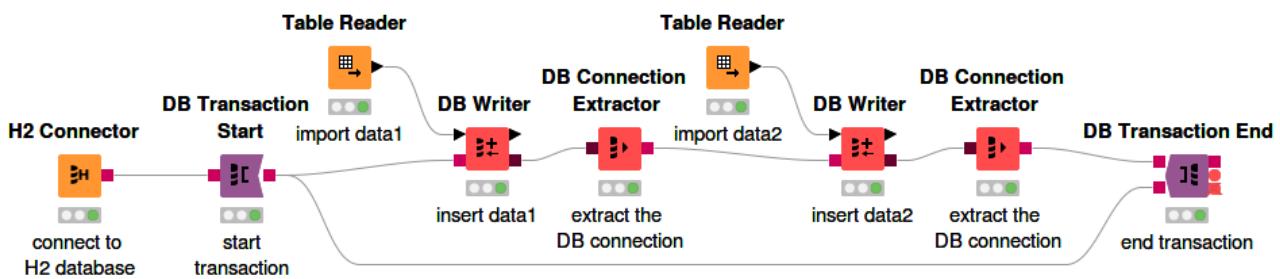


Abbildung 52. DB Transaction Nodes

# Typ Mapping

Das Datenbank-Framework ermöglicht es Ihnen, Regeln zu definieren, um von [Datenbanken](#) zu KNIME Typen und umgekehrt. Dies ist notwendig, weil Datenbanken verschiedene Artensätze unterstützen z. Oracle hat nur einen numerischen Typ mit unterschiedlichen Genauigkeiten, um ganze Zahlen zu repräsentieren, aber auch Floating-Point-Nummern, während KNIME verschiedene Typen (Integer, lang, doppelt) verwendet sie repräsentieren.

Besonders Datums- und Zeitformate werden in verschiedenen Datenbanken unterschiedlich unterstützt. Für Beispiel: [Datum und Uhrzeit \(Zoned\)](#) Typ, der in KNIME verwendet wird, um einen Zeitpunkt innerhalb eines Eine definierte Zeitzone wird nur von wenigen Datenbanken unterstützt. Aber mit der Art Mapping Framework Sie können KNIME zwingen, den Typ Date&time (Zoned) automatisch in String zu konvertieren, bevor Schreiben in eine Datenbanktabelle und Umwandeln des Strings in einen Date&time (Zoned)-Wert wenn du es liest.

Die Art Mapping Framework besteht aus einer Reihe von Mapping-Regeln für jede angegebene Richtung aus der Sicht der KNIME Analytics Plattform:

- Ausgangstyp Mapping : Die Zuordnung von KNIME-Typen zu Datenbanktypen
- Eingangstyp Mapping : Die Zuordnung von Datenbanktypen zu KNIME-Typen

Jede der Mapping-Richtung hat zwei Regeln:

- Mapping von Name : Mapping-Regeln basierend auf einem Spaltennamen (oder regulärer Ausdruck) und Typ. Nur Spalte, die beiden Kriterien entsprechen, werden berücksichtigt.
- Mapping nach Typ : Mapping-Regeln auf Basis eines KNIME- oder Datenbanktyps. Alle Spalten von der angegebene Datentyp berücksichtigt wird.

Die Typ-Mapping kann an verschiedenen Stellen im Analyse-Workflow definiert und verändert werden. Die [#page6](#) . Sie kommen mit einem vernünftigen

datenbankspezifische Standard-Mapping. Die Art Mapping-Regeln sind Teil der DB-Verbindung und DB Daten Verbindungen und von vorangehenden Knoten geerbt. Zusätzlich zum Anschluss nodes bieten alle Datenbankknoten mit einer KNIME-Datentabelle a Ausgangstyp Mapping Tab to map die Datentypen der Knoten geben den Typen der entsprechenden Datenbank KNIME-Spalten ein Spalten.

Die Zuordnung von Datenbanktypen zu KNIME-Typen kann für jede DB Daten Verbindung über die DB Type Mapper Knoten.

## DB Type Mapper

Die DB Type Mapper node ändert die Datenbank auf die KNIME-Typ-Mapping-Konfiguration für anschließende Knoten durch Auswahl eines KNIME-Typs auf den angegebenen Datenbanktyp. Die Konfiguration Dialog ermöglicht es Ihnen, neue oder bestehende Typ-Mapping-Regeln zu ändern. Alle neuen oder geänderten Regeln sind als fett markiert.

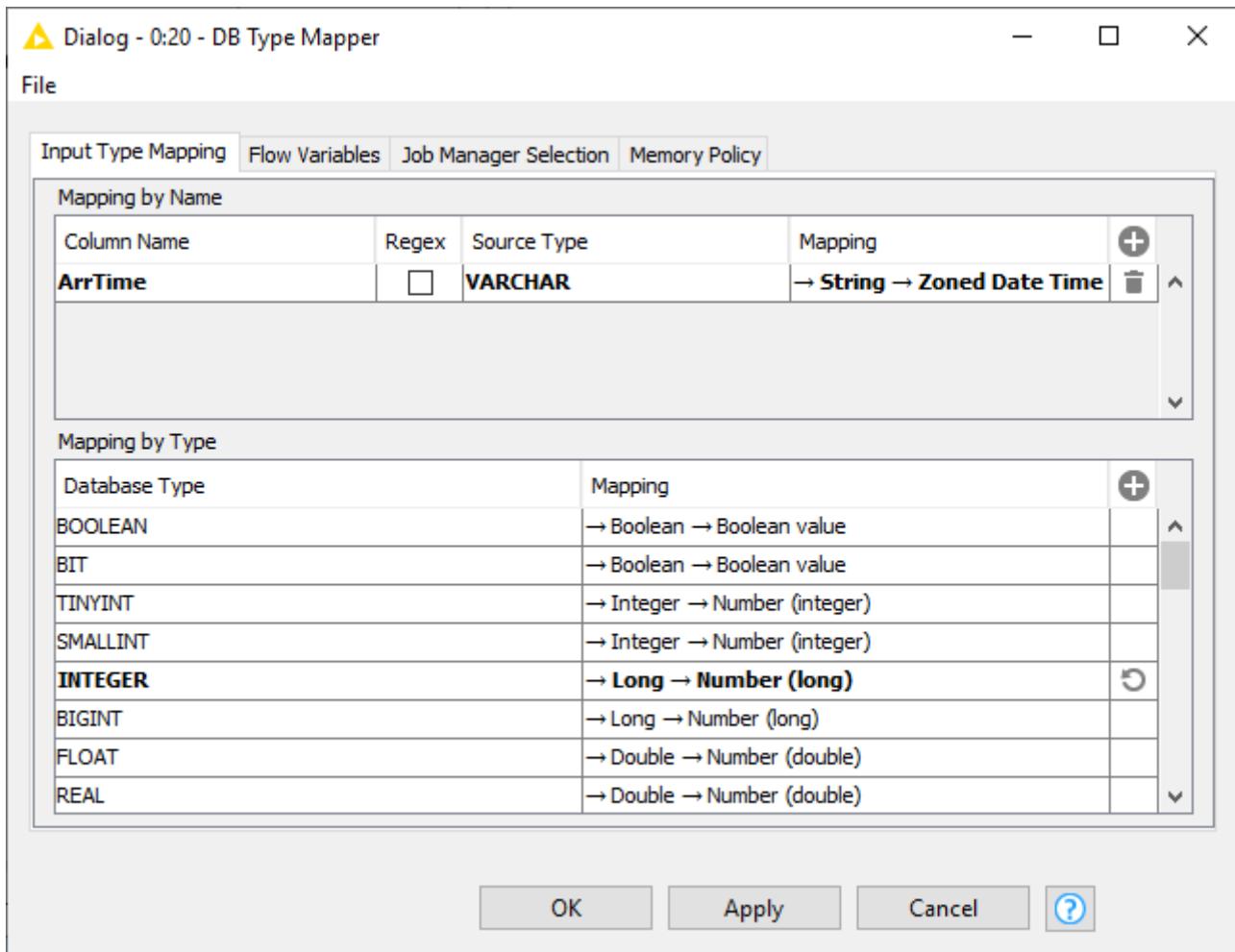


Abbildung 53. DB Type Mapper Konfiguration Dialog



Regeln aus vorhergehenden Knoten können nicht gelöscht, sondern nur geändert werden.

# Migration

Dieser Abschnitt erklärt, wie Sie Ihren Workflow migrieren, der eine deprecierter Datenbank enthält.

[Klicken Sie auf den Datenbankknoten](#page70), um zur Führung verwendet werden

Sie durch den Prozess und wandeln die deprecierten Datenbankknoten in die entsprechenden

neue Datenbankknoten. Für die Zuordnung zwischen den depreciereten und den neuen Knoten bitte [Klicken Sie auf den Datenbankknoten](#page74), um die Zuordnung zu ändern.

Alle zuvor registrierten JDBC-Treiber müssen erneut registriert werden. Für mehr

Informationen zur Registrierung eines Treibers im neuen Datenbank-Rahmen, bitte [Klicken Sie auf den Datenbankknoten](#page112), um die Registrierung zu ändern.

## Arbeitsablauf Migrationstool

Das Workflow Migrationstool ist noch in der Vorschau. Wir werden weiterhin neue hinzufügen und die bestehende Funktionalität überarbeiten.

Das Workflow Migrationstool unterstützt Sie bei der Migration bestehender Workflows, die enthalten deprecate Datenbankknoten zu den neuen Datenbankknoten. Das Werkzeug ändert keine bestehender Workflow führt aber die Migration auf einer Kopie des ursprünglichen Workflows durch.

Als Beispiel können wir in der Abbildung unterhalb eines Workflows sehen, der deprecate Datenbank enthält Knoten. Ziel ist es, das Workflow-Migrationstool zu nutzen, um uns dabei zu helfen, die abgeschriebenen Knoten zu den neuen Datenbankknoten.

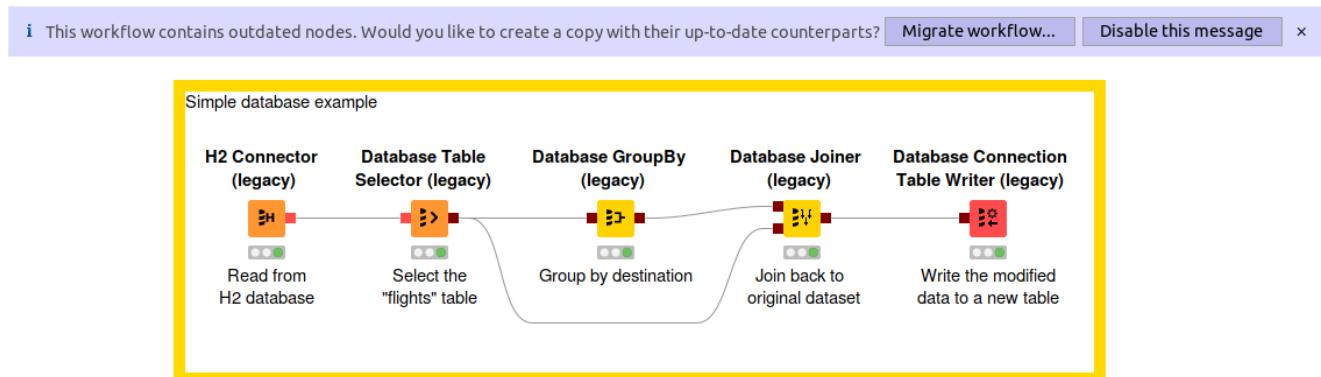


Abbildung 54. Workflow mit deprecieren Datenbankknoten

Um das Tool Workflow Migration zu starten, müssen wir einfach den Workflow öffnen, der die deprecieren Datenbankknoten, die Sie migrieren möchten. Eine Nachricht wird oben auftauchen der Workflow mit der Option, den Workflow zu migrieren (siehe Abbildung oben).

Anklicken [Workflow migrieren...](#) öffnet das Migrationsassistentfenster wie unten gezeigt. In diesem Fenster, Sie können den Workflow zu migrieren (die, die die deprecate

Datenbankknoten) und geben Sie den Namen für den neuen Workflow ein, der eine Kopie des alten einen Workflow, jedoch durch die depreciereten Datenbankknoten ersetzt durch die neuen (falls vorhanden).

Der Standardname für den neuen Workflow ist der Name des alten Workflows mit (Migration)  
als Suffix befestigt.



Der ursprüngliche Workflow wird während des Migrationsprozesses nicht verändert.

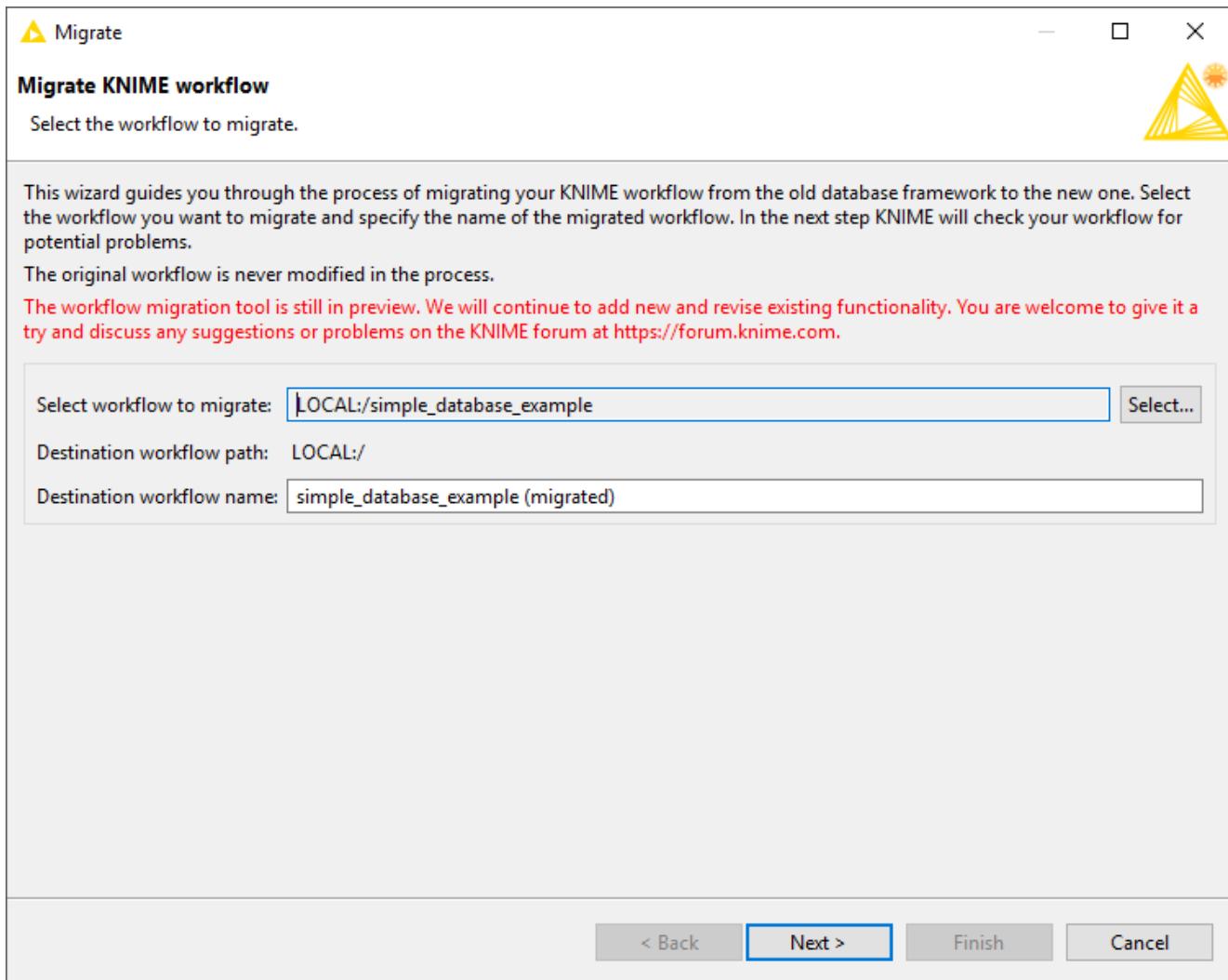


Abbildung 55. Migrationstool: Wählen Sie den Workflow aus

Klicken Sie auf **Nächste** zur nächsten Seite, wie unten gezeigt. In diesem Stadium wird der Workflow analysiert und alle deprekierten Datenbankknoten, für die eine Migrationsregel existiert, aufgeführt werden hier, zusammen mit ihren äquivalenten neuen Knoten. Das Werkzeug führt auch eine Vorprüfung durch und zeigt mögliche Probleme. Wenn Sie mit dem Mapping-Vorschlag einverstanden sind, klicken Sie den Migrationsprozess durchführen.

Nächste bis

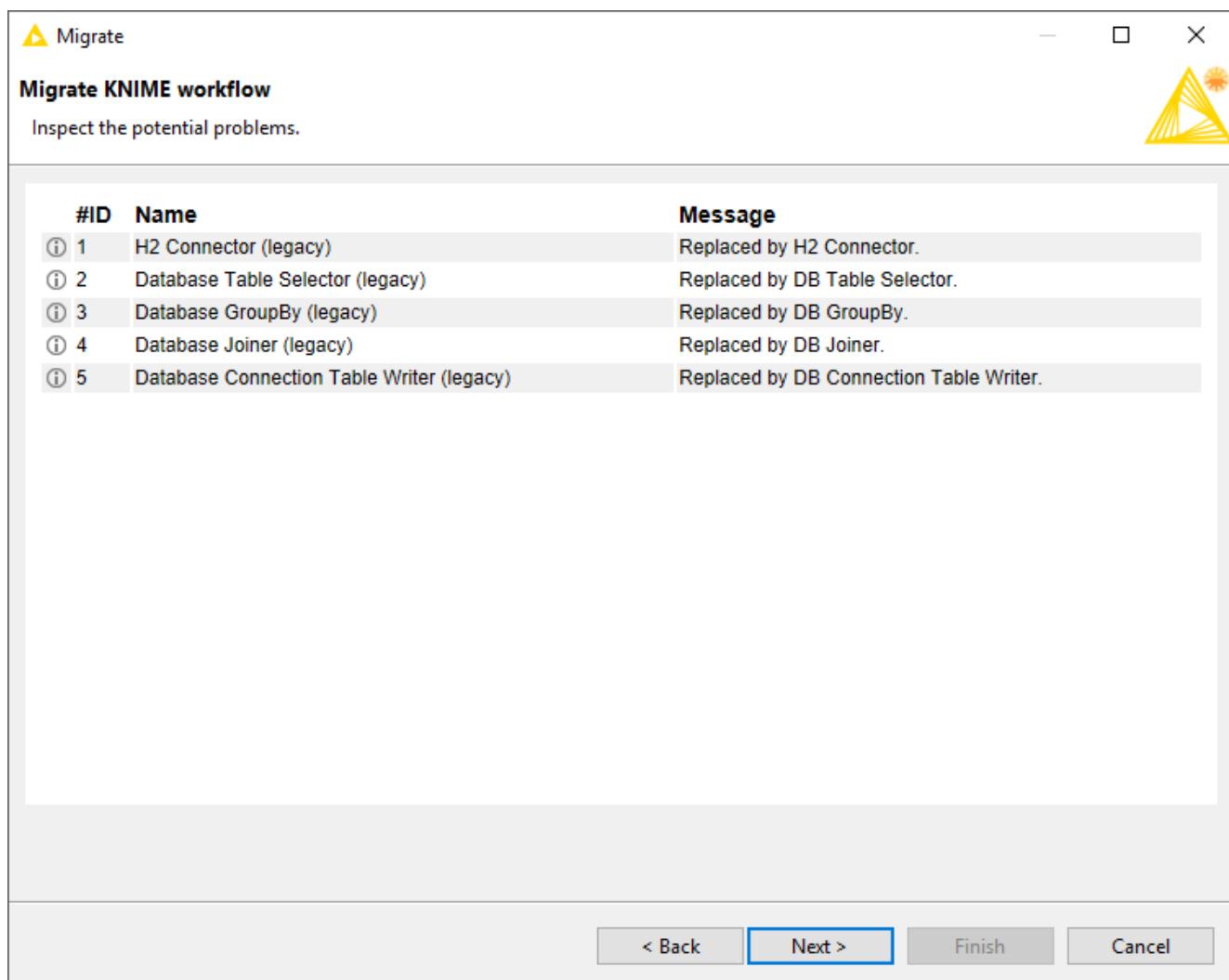


Abbildung 56. Migrationstool: Die potenzielle Kartierung zeigen

Nachdem der Migrationsprozess abgeschlossen ist, können Sie den Migrationsbericht wie der gezeigte sehen unten. Wenn während des Migrationsprozesses Warnungen oder Probleme auftreten, entsprechend Nachrichten werden im Bericht angezeigt. Sie haben auch die Möglichkeit, die Migrationsbericht im HTML-Format.

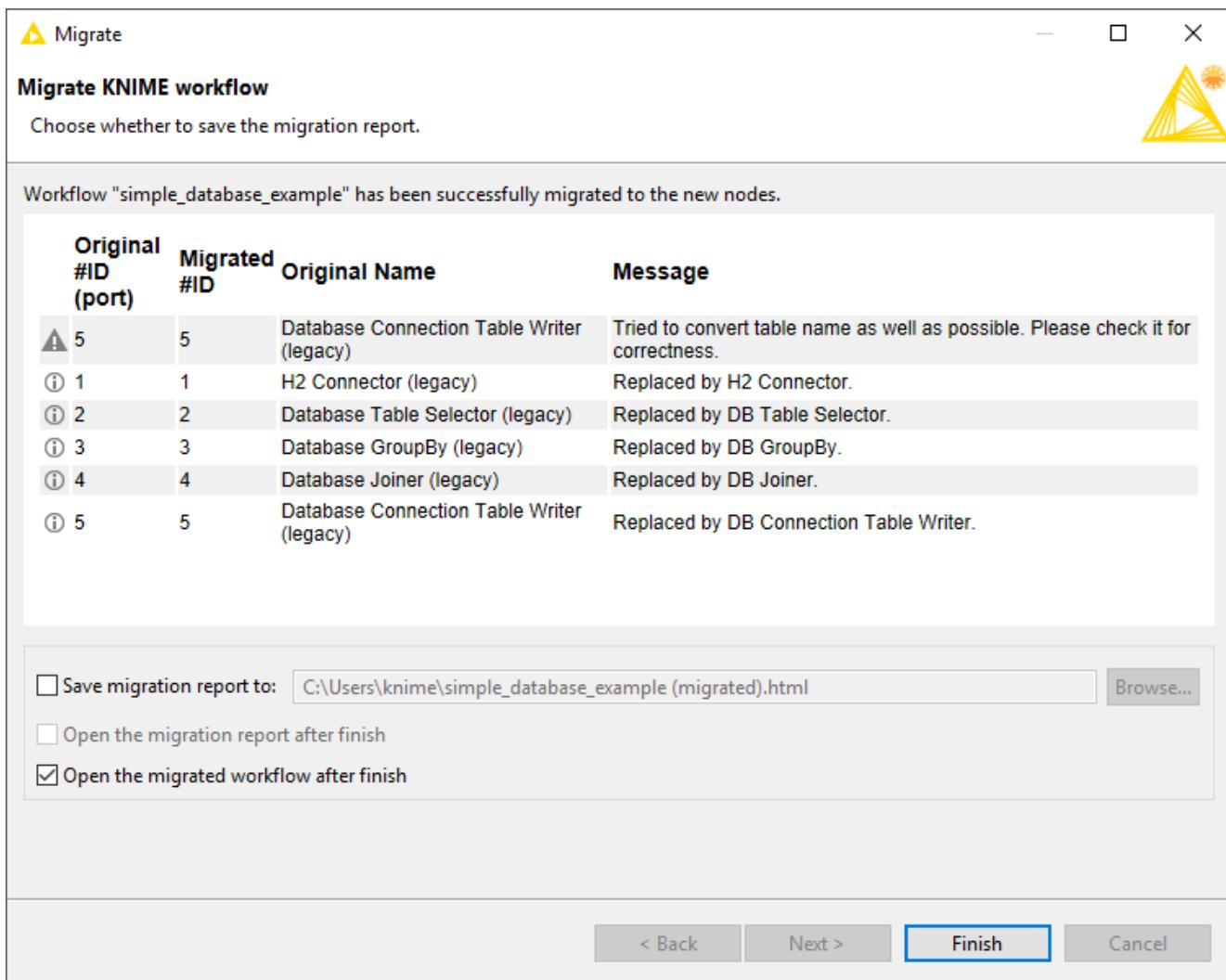


Abbildung 57. Migrationstool: Migrationsbericht

Die folgende Abbildung zeigt den migrierten Workflow, bei dem alle deprecieren Datenbankknoten ersetzt durch die neuen Datenbankknoten, während alle Einstellungen intakt bleiben.

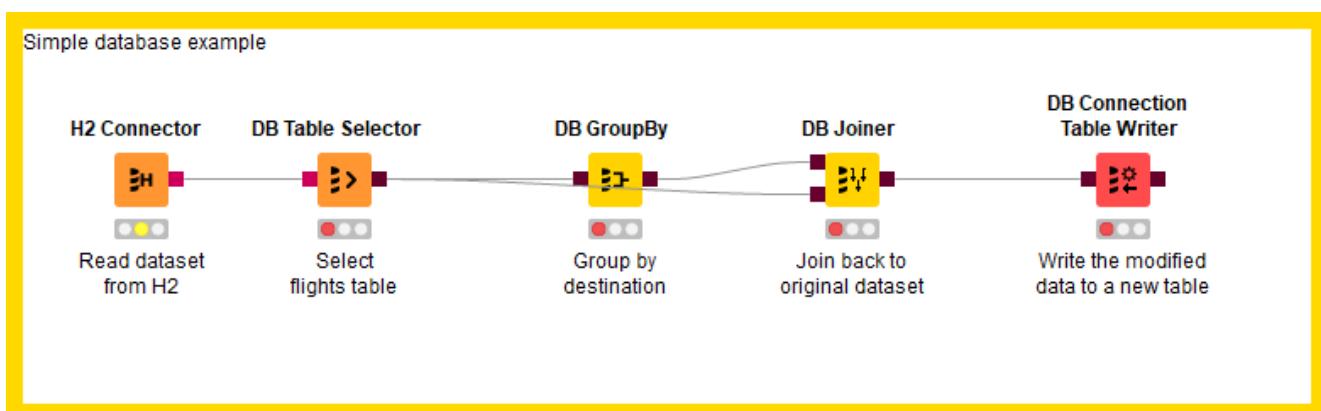


Abbildung 58. Neuer Workflow mit migrierten DB-Knoten

## Deaktivieren der Workflow-Migrationsnachricht

Wenn Sie die Migrationsnachricht nicht migrieren und deaktivieren möchten, klicken Sie auf Dies ist möglich Nachricht in der Nachricht. Die Vorlieben → Datenbanken Seite wird geöffnet, wo Sie können die Option deaktivieren Bieten Sie die Migration von Workflows an, die deprecierter Datenbankknoten enthalten wie in der Abbildung unten gezeigt. Klicken Sie auf Anwenden und schließen die Einstellung zu speichern und die Nachricht wird erscheinen nicht mehr, wenn Sie einen Workflow mit deprecierter Datenbankknoten öffnen. Zu Zurück diese Einstellung, einfach öffnen Vorlieben → Datenbanken Seite wieder und aktivieren Sie den Check Box.

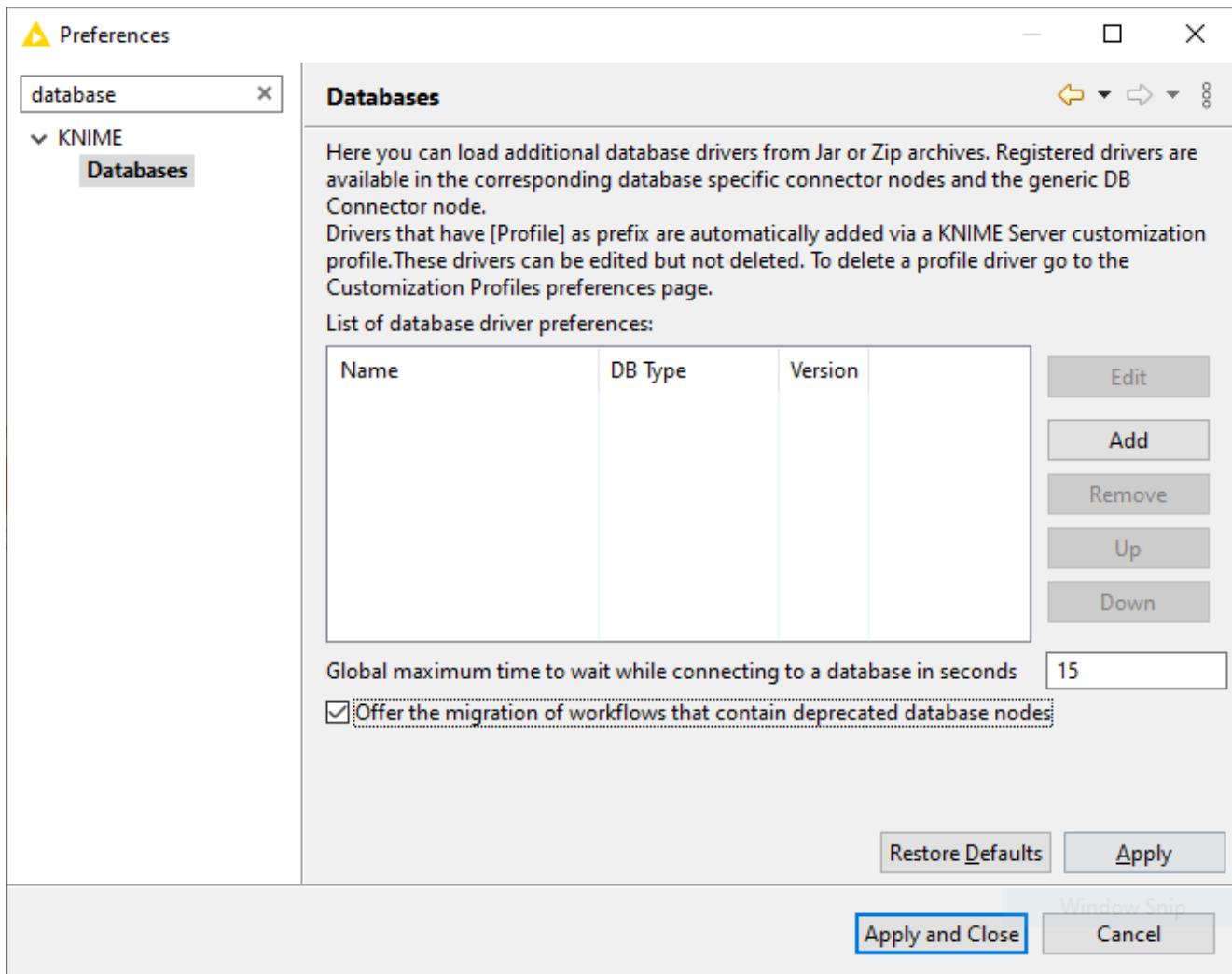


Abbildung 59. Migrationsmeldung deaktivieren

## Node Name Mapping

Die folgende Tabelle zeigt die Kartierung zwischen den deprekierten Datenbankknoten und den neuen Datenbankknoten.

Deprecated Datenbankknoten	Neue Datenbankknoten
Amazon Athena Connector	<a href="#">Amazon Athena Connector</a>
Amazon Redshift Connector	<a href="#">Amazon Redshift Connector</a>
Datenbank Apply-Binner	<a href="#">DB Apply-Binner</a>
Datenbank Auto-Binner	<a href="#">DB Auto-Binner</a>
Datenbank-Spaltenfilter	<a href="#">DB Kolumne Filter</a>
Umbenennung der Datenbank	<a href="#">Umbenennung von DB-Kolben</a>
Datenbankanschluss Tischleser	<a href="#">DB Reader</a>
Datenbank-Verbindungstabellenschreiber	<a href="#">DB Verbindungstisch Schreiber</a>
Datenbankverbinder	<a href="#">DB Connector</a>
Datenbank löschen	<a href="#">DB Löschen (Tabelle)</a> oder DB Löschen (Filter)
Datenbank Drop Tabelle	<a href="#">DB Tischentferner</a>
Datenbank GroupBy	<a href="#">DB GroupBy</a>
Datenbank-Beitrag	<a href="#">DB Joiner</a>
Datenbank-Looping	<a href="#">DB Looping</a>
Datenbank Numeric-Binner	<a href="#">DB Numeric-Binner</a>
Datenbank Pivot	<a href="#">DB Pivot</a>
Datenbankabfrage	<a href="#">DB Abfrage</a>
Datenbank Reader	<a href="#">DB Query Reader</a>
Datenbank Row Filter	<a href="#">DB Row Filter</a>
Datenbank-Sampling	<a href="#">DB Row Samping</a>
Datenbank Sortieren	<a href="#">DB Sorter</a>
Datenbank SQL Executor	<a href="#">DB SQL Ausführen</a>

Deprecated Datenbankknoten	Neue Datenbankknoten
Datenbanktabelle Connector	Kann durch <a href="#">DB Connector</a> und <a href="#">DB Tabellenauswahl</a>
Datenbank Tabellen-Autor	<a href="#">DB Table Creator</a>
Datenbanktabellenauswahl	<a href="#">DB Tabellenauswahl</a>
Datenbank-Update	<a href="#">DB Update</a>
Datenbankschreiber	<a href="#">DB Writer</a>
H2 Connector	<a href="#">H2 Connector</a>
Hive Connector	<a href="#">Hive Connector</a>
Hive Loader	<a href="#">DB Loader</a>
Impala Connector	<a href="#">Impala Connector</a>
Impala Loading	<a href="#">DB Loader</a>
Microsoft SQL Server Connector	<a href="#">Microsoft SQL Server Connector</a>
MySQL Connector	<a href="#">MySQL Connector</a>
Parameterbasierte Datenbankabfrage	<a href="#">Parameter DB Mehr erfahren</a>
PostgreSQL Anschluss	<a href="#">PostgreSQL Anschluss</a>
SQL-Extrakt	<a href="#">DB Query Extractor</a>
Injizieren	<a href="#">DB Quer-Injektor</a>
SQLite Connector	<a href="#">SQLite Connector</a>
Vertica Connector	<a href="#">Vertica Connector</a>
Ja	<a href="#">Microsoft Access Connector</a>
Ja	<a href="#">DB-Einsatz</a>
Ja	<a href="#">DB Merge</a>
Ja	<a href="#">DB Column Rename (Regex)</a>

Deprecated Datenbankknoten	Neue Datenbankknoten
Ja	<a href="#">DB Partitionierung</a>
Ja	<a href="#">DB Type Mapper</a>

## Registrieren Sie Ihre eigenen JDBC Treiber für die deprecierter Datenbank Rahmen

Die JDBC-Treiberregistrierungsseite für den deprecierter Datenbankrahmen ist nicht mehr sichtbar in den KNIME Präferenzen. Jedoch können die Fahrer noch für die Abschreibung registriert werden Datenbankrahmen.

Dieser Abschnitt ist nur gültig, wenn Sie die

[Datenbank Connector \(deprecated\)](#)

Knoten.

Um den Treiber für das neue Datenbank-Framework zu registrieren, z.  
[~~#page1121-style121rst10000text1000,readdecodation;uradres:/aRegistrieren~~](#)

[DB Connector](#)

Knotenpunkt

Abschnitt.

Um neue Treiber zu registrieren, müssen Sie eine neue Textdatei mit dem Namen erstellen

Fahrer.epf und

Folgende Inhalte:

```
*  
/instance/org.knime.workbench.core/database_drivers =  
Datei_export_version = 3.0
```

Hier verweist auf den Pfad, der auf die Treiber jar-Datei verweist. Du kannst mehrere Treiber registrieren, indem die Pfade mit ;als Separator. Die folgenden Shows ein Beispiel, das zwei verschiedene JDBC-Treiber (DB2 und Neo4J) mit Windows-Dateipfad registriert Notation.

```
*  
/instance/org.knime.workbench.core/database_drivers = C:\\\\KNIME\\\\JDBC\\\\db2\\\\db2jcc4.jar;C  
\\\\KNIME\\\\JDBC\\\\Neo4J\\\\neo4j-jdbc-driver-3.4.0.jar  
Datei_export_version = 3.0
```

Wenn Sie Windows Pfad Notation in der  
enthält Gegenschläge, die jeder Gegenschlag von einer Sekunde entkommen werden muss  
Backslash.

dass

Einmal erstellt, müssen Sie die Fahrer.epf Datei in der KNIME Analytics Platform via Datei→ Importeinstellungen . Nach dem Import der Treibereinstellungen-Datei müssen Sie neu starten KNIME Analytics Plattform, um die alten Datenbank-Frameworks die neuen Treiberdateien abholen zu lassen.

# Business Hub / Server Setup

Dieser Abschnitt enthält alles, was mit der Ausführung von Workflows zusammenhängt, die Datenbankknoten enthalten auf KNIME Hub und KNIME Server.

## JDBC Treiber für KNIME Hub und KNIME Server

KNIME Hub ermöglicht das Hochladen

[Anpassungsprofile](#)

JDBC-Treiber auf KNIME einrichten

Hubausführungen und Kunden der KNIME Analytics Platform.

KNIME Server ermöglicht es Ihnen auch zu definieren

[Anpassungsprofile](#)

JDBC automatisch einrichten

Treiber auf eigenen Ausführenden sowie Kunden der KNIME Analytics Platform.

Anstelle von Anpassungsprofilen können auch JDBC-Treiber direkt auf der

[Executor mit einem](#) [Präferenzen Datei](#). In diesem Fall jedoch die Präferenzen des Serverausführers

und alle KNIME Analytics Platform-Clients müssen manuell synchronisiert werden, so dass das gleiche

Fahrer stehen an beiden Enden zur Verfügung.

## Serverseitige Schritte

ANHANG Erstellen einer Profilordner innen /config/client-profile . Der Name  
der Ordner entspricht dem Namen des Profils. Der Ordner hält Präferenzen und andere Dateien zu verteilen.

2. Kopieren Sie die jar Datei, die den JDBC-Treiber in den Profilordner enthält.

3. Erstellen Sie innerhalb des Profilordners eine Präferenzdatei (Dateiname endet mit .epf ) mit folgende Inhalte:

```
/instance/org.knime.database/drivers//database_type=
	instance/org.knime.database/drivers//driver_class=
	instance/org.knime.database/drivers//paths/0=${profile:location}/
ERGEBNISSE
	instance/org.knime.database/drivers//url_template=
	instance/org.knime.database/drivers//version=
```

Wo:

- =====: Eine einzigartige ID für den JDBC-Treiber, bestehend nur aus alphanumerischen Charaktere und Unterpunkte.

- : Der Datenbanktyp. Bitte konsultieren Sie die in [Registrieren Sie Ihre e](#) für die Liste der derzeit verfügbaren Typen.

- Die JDBC-Treiberklasse beispielsweise  
Oracle.jdbc.OracleDriver für Oracle.
  - <===== Der Name des .jar Datei (einschließlich Dateierweiterung) dass enthält die JDBC Treiberklasse. Beachten Sie die Variable \${profile:location} steht für den Standort des Profilordners auf jedem Client, der die Anpassungsprofil. Es wird automatisch durch die richtige ersetzt Standort von jedem Client.
  - : Die JDBC URL-Vorlage für den Treiber. Bitte beachten Sie die <a href="#page15" style="color: #ff6600; text-decoration: underline;">JDBC URL Vorlage

```
jdbc\oracle\thin:@\:
```
  - <===== Die Version des JDBC Treibers z. 12.2.0 . Der Wert kann ausgewählt.  
E Server-Executoren müssen über ein Anpassungsprofil informiert werden, indem Sie Informationen an die knime.ini Datei, so dass sie es von KNIME Server anfordern können. konsultieren Sie den jeweiligen Abschnitt des [KNIME Leitfaden für die Verwaltung von Servern](#) auf man das aufstellt.
  - Beachten Sie bitte, dass am Ende dieser Datenbank-spezifische Beispiele vorliegen. Abschnitt.

### Clientseitige Schritte

KNIME Analytics Platform-Clients müssen auf ein Anpassungsprofil aufmerksam gemacht werden, damit sie können es von KNIME Server anfordern. Bitte konsultieren Sie den jeweiligen Abschnitt des [Serververwaltung Leitfadens](#) für eine vollständige Bezugnahme auf die Einrichtung dieser.

In der KNIME Analytics Platform können Sie Datei → Vorlieben → KNIME → Anpassung Profile . Das öffnet Anpassungsprofile Seite, wo Sie wählen können, welche KNIME Server und Profil zu verwenden. Die Änderungen werden nach dem Neustart von KNIME Analytics wirksam Plattform.

Um zu sehen, ob der Fahrer hinzugefügt wurde, gehen Sie zu Datei → Vorlieben → KNIME → Datenbanken .

In dieser Seite werden Treiber, die über ein Anpassungsprofil hinzugefügt werden, als Herkunft: Profil

nach der Fahrer-ID (siehe Abbildung unten). Diese Treiber können bearbeitet, aber nicht gelöscht werden. Um eine Profiltreiber, bitte zum Anpassungsprofile Seite und Auswahl der jeweiligen Profil.

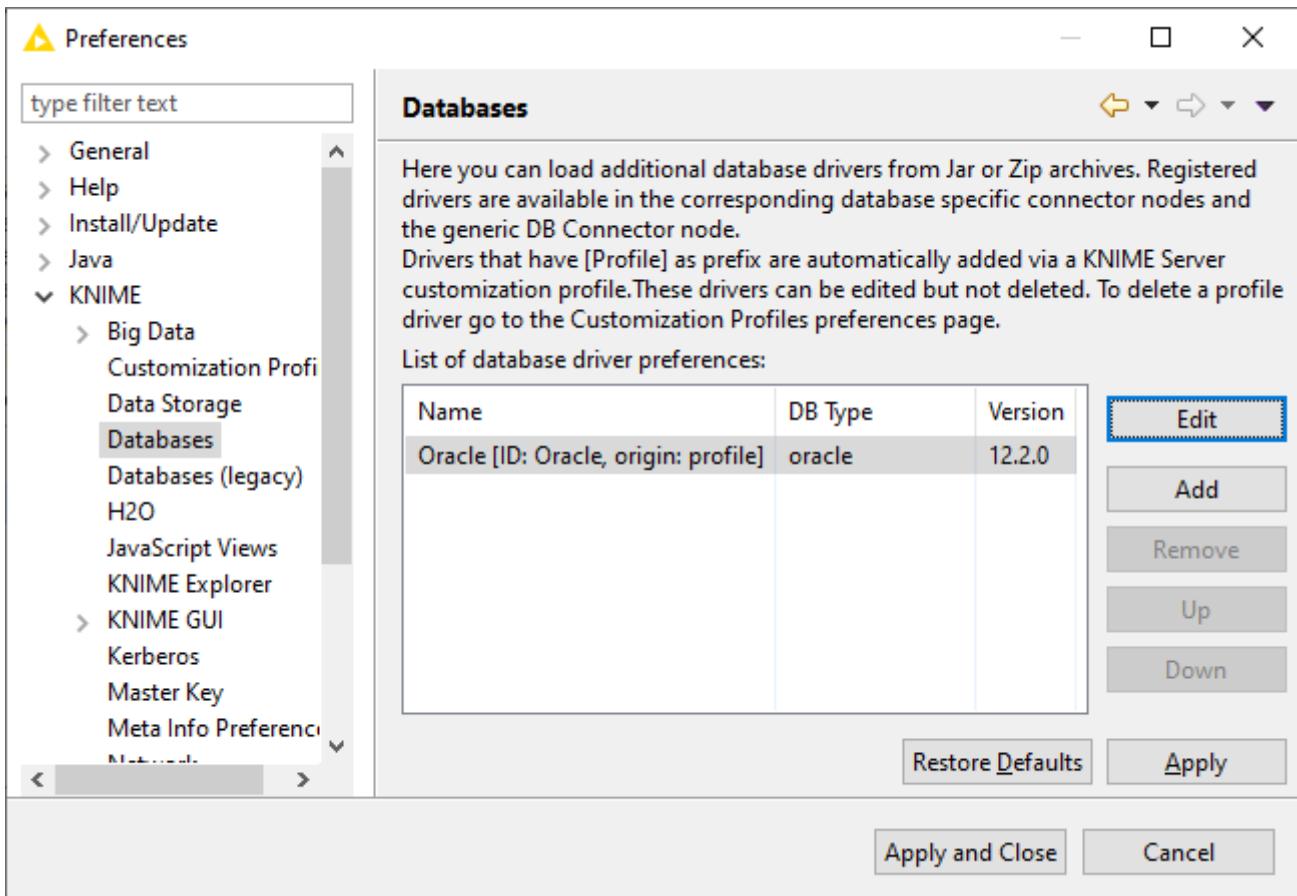


Abbildung 60. DB Vorliebenseite

## Standard JDBC Parameter

Standard JDBC Parameter bieten einen Weg für Hub-Admins, JDBC-Parameter auf JDBC zu injizieren

Verbindungen aus Workflows, die auf dem KNIME Hub laufen. Diese Parameter nehmen

Vorrang vor Werten, die im Verbindungsknoten angegeben sind. Zur Angabe eines zusätzlichen JDBC

Parameter, fügen Sie die folgenden Zeilen in die .epf Datei Ihres Anpassungsprofils:

```
/instance/org.knime.database/drivers//attributes/additional/org.knime.databas
e.util.DerivableProperties/knime.db.connection.jdbc.properties//type=
HERSTE
LLUNG
/instance/org.knime.database/drivers//attributes/additional/org.knime.databas
e.util.DerivableProperties/knime.db.connection.jdbc.properties//value=
ALTER
```

Wo:

- `<=====` : Die einzigartige ID für den JDBC-Treiber.
- `=====` : Der Name des zu setzenden JDBC-Parameters.
- `===== und =====` : Ein Typ und ein Wert, der angibt, worauf der JDBC-Parameter eingestellt werden soll.

und kann wie folgt gewählt werden:

Einstellung bis KONTEXT\_PROPERTY erlaubt die Angabe des Workflow-Kontexts

Eigenschaften. kann auf einen von:

- kontext.workflow.name : Der Name des KNIME Workflows.
- kontext.workflow.path : Der Mountpoint-relative Workflow-Pfad.
- kontext.workflow.absolute-path : Der absolute Workflow-Pfad.
- kontext.workflow.username : Der Name des KNIME Hub oder KNIME Servers Benutzer, der den Workflow ausführt.
- kontext.workflow.temp.location : Der Weg des Workflow temporär Standort.
- kontext.workflow.autor.name : Der Name des Workflow-Autors.
- kontext.workflow.last.editor.name : Der Name des Benutzers, der zuletzt bearbeitet wurde den Workflow.
- kontext.workflow.creation.date : Das Erstellungsdatum des Workflows.
- kontext.workflow.last.time.modifiziert : Die letzte geänderte Zeit der Arbeitsablauf.
- kontext.job.id : Die Job-ID bei der Ausführung auf KNIME Hub oder KNIME Server.

Einstellung bis ENTWICKLUNG erlaubt die Angabe der Anmeldename von stromvariabel. Set zum Namen der Strömungsgröße. Bitte!

Beachten Sie, dass der Wert **Name** der Anmeldeinformationen fließen variabel, um die Anmeldung zu erhalten aus, aber nicht die Anmeldung selbst.

Einstellung bis ANWENDUNGSBEREICH erlaubt die Angabe der Passwort vergessen? von stromvariabel. Set zum Namen der Strömungsgröße. Bitte!

Beachten Sie, dass der Wert **Name** der Anmeldeinformationen fließen variabel, um die Passwort aus, aber nicht das Passwort selbst.

Einstellung bis DELEGATED\_GSS\_CREDENTIAL kann in einem delegierten GSS passieren beglaubigt. Der Anmeldetag ist nur für die Kerberos-Authentifizierung und während der KNIME Hubausführung. Das KNIME Hub-Executor erstellt den delegierten GSS Anmeldeinformationen im Namen des aktuellen KNIME Hub-Benutzers, der den Workflow ausführt.

nicht gesetzt für diesen Parametertyp. Zum Beispiel, wie man dies verwendet  
~~<a href="#page88" style="color: #0000ff; text-decoration: underline;">Parameter sind hier nicht gesetzt~~

Einstellung bis FLOW\_VARIABLE erlaubt eine Durchflussgröße anzugeben. Set den Namen der Strömungsgröße.

Einstellung bis GSS\_PRINCIPAL\_NAME im Hauptnamen des

(delegiert) Kerberos Benutzer z.B., Benutzer@REALM.COM. Der Hauptname von Kerberos ist nur für die Kerberos-Authentifizierung verfügbar. Nicht gesetzt dafür [<a href="#page90" style="color: #ff6600; text-decoration: underline;">Beispiel](#page90)

Parametertyp. Zum Beispiel, wie dieser Parameter verwendet wird, siehe [Beispiel](#page90)

- Einstellung bis GSS\_PRINCIPAL\_NAME\_WITHOUT\_REALM kann in der Hauptname ohne die REALM des (delegierten) Kerberos Benutzers. Zum Beispiel wenn der Kerberos Hauptname Benutzer@REALM.COM der Wert wird Benutzer. Die Kerberos Hauptname ist nur für die Kerberos Authentifizierung verfügbar. nicht gesetzt eine für diesen Parametertyp. Beispiel für die Verwendung dieses Parameters [Beispiel](#page90) PostgreSQL
- Einstellung bis LITERAL erlaubt, einen buchstäblichen Wert anzugeben, der nicht unterschritten wird jede weitere Substitution. Set auf den wörtlichen Wert.
- Einstellung bis LOCAL\_URL eine URL in URL. , wie ein "Knie"



Beachten Sie bitte, dass am Ende dieser Datenbank-spezifische Beispiele vorliegen.  
Abschnitt.

## Reserviert JDBC Parameter

Bestimmte JDBC-Parameter können Sicherheitsprobleme verursachen, wenn ein Workflow auf KNIME ausgeführt wird Hub, z. DelegationUID für Impala/Hive-Verbindungen mit [Simba](#) basierter Treiber. Solche Parameter können als Zurück zur Übersicht zu verhindern, dass Workflows sie auf KNIME verwenden Hub. Um einen Parameter als reserviert einzustellen, fügen Sie die folgenden Zeilen in die .epf Ihre Datei Anpassungsprofil:

```
/instance/org.knime.database/drivers//attributes/reserved/org.knime.database.util.DerivableProperties/knime.db.connection.jdbc.properties/=true
```

Oder die kürzere Version:

```
/instance/org.knime.database/drivers//attributes/reserve/*/knime.db.connect auf.jdbc.properties/=true
```

Wo:

- `<=====`: Die einzigartige ID für den JDBC-Treiber.

- : Der Name des JDBC Parameters.

Beachten Sie bitte, dass am Ende dieser Datenbank-spezifische Beispiele vorliegen.  
Abschnitt.

## Verbindung Initialisierung Angaben

Die Verbindungs-Initialisierungs-Anweisung bietet einen Weg für Nabens-Admins ein SQL injizieren Erklärung, die ausgeführt wird, wenn eine JDBC-Verbindung aus einem Workflow erstellt wird, der auf KNIME Hub. Diese Aussage geht über alle anderen innerhalb der Arbeitsablauf. Um eine Initialisierungserklärung anzugeben, fügen Sie die folgende Zeile in die Anpassungsprofil: .epf Ihre Datei

```
/instance/org.knime.database/drivers//attributes/additional/java.lang.String/
knime.db.connect.init_statement/value=
```

Wo:

- `=`: Die einzigartige ID für den JDBC-Treiber.
- : Ein Wert, der die auszuführende Aussage angibt. Es ist möglich, KONTEXT\_PROPERTY Variablen innerhalb der workflow kontextbezogene Eigenschaften direkt vor der Ausführung der Erklärung. Diese Variablen haben das folgende Format: \${variablename} . Die folgenden Variablennamen sind verfügbar:

- |   |   |
|---|---|
| <input type="checkbox"/> kontext.workflow.name                  | : Der Name des KNIME Workflows.   |
| <input type="checkbox"/> kontext.workflow.path                  | : Der Mountpoint-relative Workflow-Pfad.  |
| <input type="checkbox"/> kontext.workflow.absolute-path         | : Der absolute Workflow-Pfad.   |
| <input type="checkbox"/> kontext.workflow.username              | : Der Name des KNIME Hub oder KNIME Server Benutzers die den Workflow ausführt. |
| <input type="checkbox"/> kontext.workflow.temp.location         | : Der Weg des Workflow temporäre Lage.  |
| <input type="checkbox"/> kontext.workflow.autor.name            | : Der Name des Workflow-Autors.   |
| <input type="checkbox"/> kontext.workflow.last.editor.name      | : Der Name des Benutzers, der die Arbeitsablauf.                                |
| <input type="checkbox"/> kontext.workflow.creation.date         | : Das Erstellungsdatum des Workflows.   |
| <input type="checkbox"/> kontext.workflow.last.time.modifiziert | : Die letzte modifizierte Zeit des Workflows.                                   |
| <input type="checkbox"/> kontext.job.id                         | : Die Job-ID bei der Ausführung auf KNIME Hub oder KNIME Server.                |

## Kerberos Zuständige Delegation

Dieser Abschnitt beschreibt die Konfiguration von KNIME Hub zur Ausführung

[Kerberos eingeschränkt](#)

[Delegation](#) (oder [Benutzerimperson](#)) bei Verbindung mit einer Kerberos-versicherten Datenbank wie

Apache Hive, Apache Impala, Microsoft SQL Server oder PostgreSQL. Zulässige Delegation

ermöglicht es dem KNIME Hub, im Auftrag des Nutzers Datenbankoperationen auszuführen, die a

Workflow auf einem KNIME Hub Executor statt dem KNIME Executor Kerberos Ticket-Benutzer.

Um loszulegen, müssen Sie den KNIME Hub konfigurieren, um sich gegen Kerberos zu authentifizieren.

Dazu müssen Sie die KNIME Hub Executors wie in der

[In den Warenkorb](#)

[Leitfaden](#).

Sobald alle KNIME Hub Executors eingerichtet sind, um ein Kerberos Ticket zu erhalten, können Sie Kerberos aktivieren eine eingeschränkte Delegation für einzelne JDBC-Treiber mit einer der folgenden Methoden.

 Beachten Sie bitte, dass am Ende dieser Datenbank-spezifische Beispiele vorliegen.

Abschnitt.

### Standard JDBC Parameter

[\*\*Standard JDBC Parameter\*\*](#) Bereitstellung einer Möglichkeit für Hub-Admins, JDBC-Parameter auf JDBC injizieren

Verbindungen aus Workflows, die auf dem KNIME Hub laufen. Je nach Datenbank a

verschiedene Parameter und Werte werden verwendet, um eine eingeschränkte Delegation durchzuführen. Nur einige Fahrer den Namen des Benutzers zu verlangen, der impersoniert werden sollte. Dies kann mit dem Namen geschehen

des KNIME Hub-Benutzers, der den Workflow ausführt (kontext.workflow.username) als [DELEGATED\\_GSS\\_CREDENTIAL](#) zum Beispiel. Andere Fahrer benötigen

die delegierte GSS-Berechtigung als Parameter, der automatisch an den Treiber übergeben werden kann

über den Werttyp [DELEGATED\\_GSS\\_CREDENTIAL](#). Die GSS-Berechtigung wird mit Hilfe der [MS-SFU Kerberos 5 Erweiterung](#) zum Beispiel.

### Service Ticket Delegation

Wenn der Fahrer keinen dedizierten Parameter für eine eingeschränkte Delegation des KNIME bereitstellt

Der Ausführende kann im Auftrag des Nutzers ein Kerberos-Service-Ticket anfordern, das die Ausführung des Service-Tickets durchführt.

[Workflow mit Hilfe des MS-SFU Kerberos 5 Erweiterung](#). Das erhaltene Serviceticket wird dann von

den JDBC-Treiber beim Herstellen der Verbindung zur Datenbank.

Um das Service-Ticket anzufordern, benötigt der KNIME Executor den Service-Namen und die

qualifizierter Hostname. Um den Dienstnamen anzugeben, fügen Sie die folgende Zeile in die

.epf Ihre Datei

KNIME Ausführungsbeispiele Anpassungsprofil:

```
/instance/org.knime.database/drivers//attributes/additional/java.lang.String/
knime.db.connection.kerberos_delegation.service/value =
```

in Erwägung nachstehender Gründe:

ist der Name des Dienstes, um das Kerberos-Ticket zu verlangen. Vgl.

[PostgreSQL](#page90) zum Beispiel.

Der vollständig qualifizierte Hostname wird automatisch aus der Hostname-Einstellung für alle extrahiert.

~~<a href="#page10" style="color: #000000; text-decoration: underline;">dediziertes JDBC-Verbindungsstring. Der Name wird aus~~

die JDBC-Verbindungsstring unter Verwendung eines Standard-Standard-Ausdrucks ( `.*(?:@|/)([^:;,\n]*).*` )

die für die meisten JDBC-Strings aus der Box herausarbeiten sollte. Falls erforderlich, kann es

geändert, indem die folgende Zeile dem `.epf` Datei Ihres KNIME Anpassung der Ausführungsbeispiele

Profil:

```
/instance/org.knime.database/drivers//attributes/additional/java.lang.String/
knime.db.connection.kerberos_delegation.host_regex/value =
```

Die `host_regex` sollte einen regelmäßigen Ausdruck enthalten, der den vollständig qualifizierten Hostnamen extrahiert aus der JDBC URL mit der ersten Gruppe, die dem Hostnamen entspricht.

Verbindung Initialisierung Angaben

[Verbindung Initialisierungserstellung](#page83) Die `host_regex` bietet einen Weg für Hub-Admins ein SQL injizieren

Erklärung, die ausgeführt wird, wenn eine JDBC-Verbindung aus einem Workflow erstellt wird, der auf

KNIME Hub. Diese Funktion kann für eine eingeschränkte Delegation in einigen Datenbanken verwendet werden, so

wie Mehr durch Ausführen einer bestimmten SQL-Anweisung mit der `kontext.workflow.username` wie z.

```
IMPERSONAL ${context.workflow.username};
```

Beispiel: Apache HiveTM

Verbindungen zu Apache Hive erfordert weitere Setup-Schritte abhängig vom verwendeten JDBC-Treiber.

In diesem Beispiel zeigen wir, wie:

[ANHANG Registrieren Sie den Inhaber](#) [Hive JDBC Treiber zur Verfügung gestellt von Cloudera](#) auf KNIME Hub.

[2. Konfigurieren Benutzerimperson](#) auf KNIME Hub (für eingebettete und proprietäre Hive JDBC Treiber).

## Eigene Simba-basierte JDBC-Treiberregistrierung

Wenn die Verwendung von eingebetteter Open-Source Apache Hive JDBC Driver bevorzugt ist, skip [Abschnitt](#).

[ANHANG Download des Eigentums](#) [Hive JDBC Treiber](#) von der Cloudera-Website.

2. Erstellen Sie den Profilordner innen /config/client-profile und Name In den Warenkorb Hive (zum Beispiel).

3. Kopie HiveJDBC41.jar vom heruntergeladenen JDBC-Treiber in das neu erstellte Profil Ordner.

L 347 vom 20.12.2013, S. 1). Erstellen Sie im Profilordner eine neue Präferenzdatei namens .epf (zum Beispiel) mit folgende Inhalte:

```
/instance/org.knime.database/drivers/cloudera_hive/database_type=hive
/instance/org.knime.database/drivers/cloudera_hive	driver_class=com.cloudera.hive.
jdbc41.HS2 Fahrer
/instance/org.knime.database/drivers/cloudera_hive/paths/0=${profile:location}/Hiv
eJDBC41.jar
/instance/org.knime.database/drivers/cloudera_hive/url_template= jdbc\hive2\://
st> \:[Datenbank]
/instance/org.knime.database/drivers/cloudera_hive/version=2.6.0
```

5. Wenn, wie empfohlen, KNIME Hub muss Workflow-Benutzer zu verkörpern, gehen Sie auf die [Abschnitt](#).

## Benutzer-Imitation auf Hive

Dieses Beispiel stellt den Hive JDBC-Treiber (embedded or proprietary) ein, so dass KNIME Hub [Benutzer von Workflows](#) auf JDBC-Verbindungen.

Die Aktivierung von Benutzer-Imitation auf Hive hängt vom verwendeten JDBC-Treiber ab:

- Fügen Sie für den eingebetteten Apache Hive JDBC Treiber die folgenden Zeilen zum KNIME Hub hinzu [Präferenzen Datei](#)

```
/instance/org.knime.database/drivers/hive/attributes/additional/org.knime.database.util.  
DerivableProperties/knime.db.connection.jdbc.properties/hive.server2.proxy.user/type = CON  
TEXT_PROPERTY  
/instance/org.knime.database/drivers/hive/attributes/additional/org.knime.database.util.  
DerivableProperties/knime.db.connection.jdbc.properties/hive.server2.proxy.user/value = co  
ntext.workflow.username  
/instance/org.knime.database/drivers/hive/attributes/reserved/*/knime.dbconnection.jdbc  
.properties/hive.server2.proxy.user = true
```

- Für den proprietären Simba-basierten JDBC-Treiber fügen Sie die folgenden Zeilen den Präferenzen hinzu  
[<a href="#page86" style="color: #000000; text-decoration: none;">Datei #1600, Schritt 4.](#page86)  
(Schritt 4).

```
/instance/org.knime.database/drivers/cloudera_hive/attributes/additional/org.knime.datab  
ase.util.DerivableProperties/knime.db.connection.jdbc.properties/DelegationUID/type = CONT  
ERGEBINNISSE  
/instance/org.knime.database/drivers/cloudera_hive/attributes/additional/org.knime.datab  
ase.util.DerivableProperties/knime.db.connection.jdbc.properties/DelegationUID/value = con  
Text.workflow.username  
/instance/org.knime.database/drivers/cloudera_hive/attributes/reserve/*/knime.db.connec  
.jdbc.properties/DelegationUID = true
```

## Beispiel: Apache ImpalaTM

In diesem Beispiel werden wir die Inhaberschaft registrieren

[Impala JDBC Treiber zur Verfügung gestellt von Cloudera](#) auf

KNIME Hub. Dieses Beispiel setzt den Fahrer ein, damit KNIME Hub

[impersonierender Workflow](#)

[Benutzer](#) auf JDBC-Verbindungen.

Wenn Sie die eingebettete Open-Source verwenden      Apache Hive JDBC Driver (für Impala)      , du  
  muss diesen Schritt nicht tun. Bitte beachten Sie jedoch, dass in diesem Fall Benutzer  
Imitation auf KNIME Hub ist aufgrund von Einschränkungen des Fahrers nicht möglich.

[ANHANG Download des Eigentums](#)      [Impala JDBC](#)      von der Cloudera-Website.

2. Erstellen Sie den Profilordner innen      /config/client-profile und  
Name      In den Warenkorb Impala (zum Beispiel).

3. Kopie      ImpalaJDBC41.jar      vom heruntergeladenen JDBC-Treiber in das neu erstellte Profil  
Ordner.

L 347 vom 20.12.2013, S. 1). Erstellen Sie im Profilordner eine neue Präferenzdatei namens [Impala.epf](#) (zum Beispiel) mit  
die folgenden Inhalte:

```

/instance/org.knime.database/drivers/cloudera_impala/database_type = impala
/instance/org.knime.database/drivers/cloudera_impala	driver_class = com.cloudera.impala.jdbc.Fahrer
/instance/org.knime.database/drivers/cloudera_impala/paths/0 = ${profile:location}/ImpalaJDBC41.jar
/instance/org.knime.database/drivers/cloudera_impala/url_template = jdbc\:impala\://[Datenbank]
/instance/org.knime.database/drivers/cloudera_impala/version = 2.6.0
/instance/org.knime.database/drivers/cloudera_impala/attributes/additional/org.knime.database.util.DerivableProperties/knime.db.connection.jdbc.properties/DelegationUID/Typ = CONTEXT_PROPERTY
/instance/org.knime.database/drivers/cloudera_impala/attributes/additional/org.knime.database.util.DerivableProperties/knime.db.connection.jdbc.properties/DelegationUID/value = context.workflow.username
/instance/org.knime.database/drivers/cloudera_impala/attributes/reserved/*/knime.db.connection.jdbc.properties/DelegationUID = true

```

Beachten Sie, dass die letzten drei Zeilen die	DelegationUID	JDBC-Parameter zur Kraft	<a href="#">Benutzer</a>
<a href="#">Verkörperung</a>	( <b>empfohlen</b> ) Wenn Sie nicht wollen, dass KNIME Hub		

Workflow-Benutzer können Sie die letzten drei Zeilen entfernen.

## Beispiel: Microsoft SQL Server

Verbindungen zu Microsoft SQL Server benötigen weitere Setupschritte. In diesem Beispiel werden zeigen, wie:

ANHANG Registrieren Sie den SQL Server JDBC Treiber von Microsoft auf KNIME Hub.

[2. Konfigurieren](#) [Benutzerimperson](#) auf KNIME Hub, der für Kerberos empfohlen wird  
Authentifizierung.

## Microsoft Treiber Installation

Der SQL Server JDBC Treiber von Microsoft erfordert eine spezielle Lizenzierung, die Sie benötigen zustimmen. Deshalb bietet KNIME ein zusätzliches Plug-in, um den Treiber zu installieren. Um Installieren Sie das Plug-in den Schritten, wie in der Abschnitt.

## Beschränkte Delegation auf Microsoft SQL Server

Wenn Sie Kerberos basierende Authentifizierung für Microsoft SQL Server verwenden, empfehlen wir auch Benutzer-Imitation einrichten. Dieses Beispiel erstellt den Microsoft SQL Server JDBC Treiber so dass [KNIME Hub](#) [Benutzer von Workflows](#) auf JDBC-Verbindungen. Für weitere Details

über die Verwendung von Kerberos integrierter Authentifizierung mit Microsoft SQL Server

[Microsoft](#)

[SQL Serverdokumentation](#)

Aktivieren Sie die Benutzer-Imitation für den eingebetteten Microsoft SQL Server Treiber, indem Sie den

[nach dem KNIME Hub](#)

[Präferenzen Datei](#)

```
/instance/org.knime.database/drivers/built-in-msqlserver-
9.4.0/zusätzlich/org.knime.database.util.DerivableProperties/knime.dbconnect
ion.jdbc.properties/gscredential/type = DELEGATED_GSS_CREDENTIAL
```

In der [Microsoft SQL Server Connector](#)

sicherstellen, dass Sie Kerberos in der

Authentifizierung Einstellen im Konfigurationsfenster und Hinzufügen der folgenden zwei Parameter  
[<a href="#page19" style="color: #ff6600; text-decoration: underline;">JDBC Parameter](#)</a>

in der Tab:

AuthentifizierungScheme = JavaKerberos

IntegrierteSicherheit = wahr

Diese Parameter können automatisch als

[<a href="#page80" style="color: #ff6600; text-decoration: underline;">über den KNIME Hub](#)

[Präferenzen Datei](#) durch Hinzufügen der folgenden Zeilen:

```
/instance/org.knime.database/drivers/built-in-msqlserver-
9.4.0/zusätzlich/org.knime.database.util.DerivableProperties/knime.dbconnect
ion.jdbc.properties/authenticationScheme/Typ = literal
/instance/org.knime.database/drivers/built-in-msqlserver-
9.4.0/zusätzlich/org.knime.database.util.DerivableProperties/knime.dbconnect
ion.jdbc.properties/authenticationScheme/Wert = JavaKerberos
```

```
/instance/org.knime.database/drivers/built-in-msqlserver-
9.4.0/zusätzlich/org.knime.database.util.DerivableProperties/knime.dbconnect
ion.jdbc.properties/integrierteSicherheit/Typ = literal
/instance/org.knime.database/drivers/built-in-msqlserver-
9.4.0/zusätzlich/org.knime.database.util.DerivableProperties/knime.dbconnect
ion.jdbc.properties/integrierteSicherheit/Wert = wahr
```

Wenn Sie Ihren eigenen Treiber verwenden möchten, müssen Sie den eingebauten Treiber id (

gebaut...

In-msqlserver-9.4.0

) mit der ID Ihres Fahrers. Für weitere Informationen über

um Ihren eigenen Treiber zu installieren

[<a href="#page78" style="color: #ff6600; text-decoration: underline;">Abschnitt.](#)

Abschnitt.

Weitere Einzelheiten zu den Parametern, die für eine eingeschränkte Delegation verwendet werden, siehe

[Microsoft SQL Serverdokumentation](#)

## Beispiel: Oracle Datenbank

Verbindungen zur Oracle Datenbank erfordern weitere Setupschritte. In diesem Beispiel zeigen wir wie folgt:

ANHANG Registrieren Sie den Oracle Database Treiber von Oracle auf KNIME Hub.

2. Konfigurieren    [Benutzerimperson](#)        auf KNIME Hub, der für Kerberos empfohlen wird  
Authentifizierung.

### Oracle Datenbank Treiberinstallation

Der Oracle Database JDBC Treiber erfordert eine spezielle Lizenzierung, die Sie zustimmen müssen.

Deshalb bietet KNIME ein zusätzliches Plug-in, um den Treiber zu installieren. Um die [Authentifizierung für Oracle Database](#) zu ermöglichen, müssen Sie den Oracle JDBC Driver in der KNIME Installation installieren. Um dies zu tun, gehen Sie zu den [Systemeinstellungen](#) und klicken auf den [Setup](#)-Abschnitt.

### Beschränkte Delegation auf Oracle Datenbank

[Kerberos](#)  
Wenn Sie basierende Authentifizierung für Oracle Database, empfehlen wir auch Setup  
eine eingeschränkte Delegation.

Aktivieren Sie die Kerberos eingeschränkte Delegation für den installierten Oracle Treiber, indem Sie die folgende Zeilen zum KNIME Hub        [Präferenzen Datei](#)        (z. Das ist eine gute Idee)

```
/instance/org.knime.database/drivers/built-in-oracle-  
19.14.0/zusätzlich/java.lang.String/knime.db.connection.kerberos_delegation.s  
ervice/value = oracle
```

Wenn Sie Ihren eigenen Treiber verwenden möchten, müssen Sie den eingebauten Treiber id ( gebaut...

in-oracle-19.14.0 ) mit der ID Ihres Fahrers. Weitere Informationen zum Thema

Ihren eigenen Treiber einrichten sehen

Abschnitt.

Der Dienstname selbst wird nicht verwendet, so dass jede nicht-leere String eingeschränkt ermöglicht delegieren.

## Beispiel: PostgreSQL

Wenn Sie Kerberos basierende Authentifizierung für PostgreSQL verwenden, empfehlen wir auch Setup  
eine eingeschränkte Delegation.

Aktivieren Sie die Kerberos eingeschränkte Delegation für den eingebetteten PostgreSQL Treiber durch Hinzufügen  
[die folgende Zeile zum KNIME Hub](#) [Präferenzen Datei](#).

```
/instance/org.knime.database/drivers/built-in-postgres-
42.3.5/zusätzlich/java.lang.String/knime.db.connection.kerberos_delegation.se
rvice/value=postgres
```

Der Standard-Dienstname für PostgreSQL ist Postgrieß, aber es könnte anders sein je nach Ihr Datenbank-Setup. In diesem Fall müssen Sie sich ändern Postgrieß in der oben genannten Zeile auf den Dienstnamen.

Standardmäßig verwendet der PostgreSQL-Treiber den Betriebssystembenutzer während des Logins, wenn Verwendung der Kerberos-Authentifizierung, die Probleme verursacht, wenn der Workflow ausgeführt wird der KNIME Hub. Je nach Datenbank-Setup möchten Sie es entweder auf den Kerberos setzen

Hauptname oder ohne REALM oder kontext.workflow.username je nach Ihrer Datenbank-Setup.

Um den Benutzernamen auf den Hauptnamen einschließlich der REALM z. Benutzer@REALM.COM Verwendun  
 die folgende Zeile:

```
/instance/org.knime.database/drivers/built-in-postgres-
42.3.5/attributes/additional/org.knime.database.util.DerivableProperties/knime.db.connec
:jdbc.properties/user/type=GSS_PRINCIPAL_NAME
```

Um den Hauptnamen ohne die REALM zu verwenden, verwenden Sie diese Zeile stattdessen:

```
/instance/org.knime.database/drivers/built-in-postgres-
42.3.5/attributes/additional/org.knime.database.util.DerivableProperties/knime.db.connec
:jdbc.properties/user/type=GSS_PRINCIPAL_NAME_WITHOUT_REALM
```

Wenn Sie Ihren eigenen Treiber verwenden möchten, müssen Sie den eingebauten Treiber id ( gebaut...  
 In-postgres-42.3.5 ) mit der ID Ihres Fahrers. Weitere Informationen zum Thema  
 Ihren eigenen Treiber einrichten sehen [Abschnitt.](#page78)



KNIME AG  
Talacker 50  
8001 Zürich, Schweiz  
[www.knime.com](http://www.knime.com)  
[Info@knime.com](mailto:Info@knime.com)