

Verbindung zu Notes-Datenbanken herstellen

Der Verbindungsaufbau zu einer Notes-Datenbank mit vielen Masken und Ansichten kann relativ langsam erfolgen, da der ODBC-Treiber von Lotus Notes während SQLConnect aus Gestaltungsdokumenten für Masken und Ansichten abgeleitete Systemkataloge aufbaut.

Lotus-Notes-ODBC-Setup-Dialogfeld

1. Geben Sie im Textfeld Data Source Name einen Namen für die Datenquelle ein. Fügen Sie zum Beispiel den Namen MITARB hinzu, um die ODBC-Verbindung zu einer Mitarbeiter-Datenbank zu kennzeichnen.
2. Optional: Geben Sie eine Beschreibung der Datenquelle im Textfeld Description ein. Geben Sie zum Beispiel die Beschreibung "Einstellungsdatum, Gehaltsentw. und aktuelle Übersicht aller Mitarbeiter" ein, um den Inhalt der Mitarbeiter-Datenbank zu beschreiben.
3. Geben Sie den Namen des Notes-Servers ein, der die Notes-Datenbank enthält, die Sie im Textfeld Server öffnen möchten. Lassen Sie das Textfeld leer, wenn sich die Notes-Datenbank auf einer lokalen Platte befindet.
4. Geben Sie den Pfad und Namen der .NSF-Datei ein, die Sie im Textfeld öffnen möchten. Ein Beispiel:

C:\PERSONAL\MITARB.NSF

NotesSQL Options Setup

Sie können jeden der folgenden optionalen Parameter ändern. Die Werte für diese Parameter beeinflussen die Speicherzuweisung zur Laufzeit:

Max Length of Text Fields

Dieser Parameter legt die maximale Anzahl Zeichen fest, die der Notes-Treiber in einer Zeichenfolge zulässt. Dies schließt die maximale Anzahl Zeichen ein, die von einem Notes-Textfeld zurückgeliefert wird, sowie die maximale Länge einer Zeichenfolge, die sich in ein Notes-Dokument einfügen lässt.

Der höchste zulässige Wert ist 10.240 (10K).

Der Vorgabewert ist 254.

Hinweis Notes hat lange Textfelder mit der Bezeichnung Rich-Text-Felder. Der Haupttext eines Notes-Dokuments besteht typischerweise aus einem einzigen Rich-Text-Feld. Falls Sie alle Daten in ein solches Feld lesen möchten, müssen Sie darauf achten, daß die von Ihnen angegebene maximale Zeichenfolgelänge hoch genug ist.

Max Number of Tables

Dieser Parameter gibt die maximale Tabellenanzahl pro Anfrage an.

Der höchste zulässige Wert ist 100.

Der Vorgabewert ist 20.

Max Number of Subqueries

Dieser Parameter gibt die maximale Anzahl verschachtelter Unterabfragen in einer SQL-Anweisung an.

Der höchste zulässige Wert ist 100.

Der Vorgabewert ist 20.

SQL Statement Length

Dieser Parameter gibt die maximale Länge einer SQL-Anweisung an, die an SQLPrepare/SQLExecDirect übergeben wird.

Der höchste zulässige Wert ist 32.767 (32K).

Der Vorgabewert ist 1.024.

Keep Temporary Indexes Until SQLDisconnect

Diese Optionsschaltfläche steuert die Speicherung temporärer Indizes. Durch die Auswahl dieser

Option werden temporäre Indizes bis SQLDisconnect gespeichert. Lassen Sie die Option ausgeschaltet, wenn Indizes am Ende jedes SELECT-Ergebnisses gelöscht werden sollen.

Siehe auch

[Hardware- und Software-Anforderungen](#)

[Lotus-Notes-Datenquelle hinzufügen, bearbeiten und löschen](#)

Lotus-Notes-Datenquelle

Eine Lotus-Notes-Datenquelle gibt den Notes-Server und die Notes-Datenbank an, die Sie öffnen möchten.

Der Lotus-Notes-ODBC-Treiber

Die folgenden Themen liefern Ihnen hilfreiche Informationen über die Anwendung des Lotus-Notes-ODBC-Treibers:

[Performance des Lotus-Notes-ODBC-Treibers optimieren](#)

[SQL-Tabellen, Ansichten und Indizes auf Notes abbilden](#)

[Definition der Spalten von Notes-Ansichten](#)

[Die Universal Relation Table](#)

[Spaltennamen](#)

[Verbindung zu Notes-Datenbanken herstellen](#)

[Reservierte Wörter](#)

Verbindung zu einer Lotus-Notes-Datenquelle herstellen

Ist Ihre Notes-ID Paßwort-geschützt, fordert Sie Notes jedesmal zur Eingabe eines Paßwortes auf, wenn Sie versuchen, sich mit dem Lotus-Notes-ODBC-Treiber bei einer entfernten Datenbank anzumelden. Das Paßwort ist nicht Datenbank-spezifisch und wird nicht gespeichert, so daß Sie es mehrmals eingeben müssen, wenn Sie eine Verbindung zu mehreren Datenbanken herstellen möchten.

Notes fordert Sie niemals zur Eingabe Ihrer Notes-ID auf, da diese ein Bestandteil Ihrer Workstation-Installation ist.

Eine Lotus-Notes-Datenquelle hinzufügen, bearbeiten und löschen

Bevor Sie Daten mit dem Lotus-Notes-ODBC-Treiber abfragen können, müssen Sie dafür eine Datenquelle hinzufügen. Sie können eine Datenquelle jederzeit hinzufügen oder löschen.

Eine Lotus-Notes-Datenquelle hinzufügen

1. Doppelklicken Sie auf das Symbol der Systemsteuerung, das Sie in der Hauptgruppe des Programm-Manager-Fensters finden.
2. Doppelklicken Sie auf das Symbol ODBC, das Sie im Systemsteuerung-Fenster finden.
Es erscheint das Dialogfeld Data Sources.
3. Wählen Sie Add.
Es erscheint das Dialogfeld Add Data Source.
4. Wählen Sie aus dem Listenfeld Installed ODBC Drivers die Option Lotus Notes aus.
5. Wählen Sie OK.
6. Geben Sie im Lotus-Notes-ODBC-Setup-Dialogfeld die Informationen zum Einrichten der Datenquelle ein.

Eine Lotus-Notes-Datenquelle bearbeiten

1. Doppelklicken Sie auf das Symbol der Systemsteuerung, das Sie in der Hauptgruppe des Programm-Manager-Fensters finden.
2. Doppelklicken Sie auf das Symbol ODBC, das Sie im Systemsteuerung-Fenster finden.
Es erscheint das Dialogfeld Data Sources.
3. Wählen Sie die Datenquelle aus dem Listenfeld Data Sources (Drivers) aus.
4. Wählen Sie Setup.
5. Ändern Sie die Information über die Datenquelle im Dialogfeld Lotus Notes ODBC Setup.

Eine Lotus-Notes-Datenquelle löschen

1. Doppelklicken Sie auf das Symbol der Systemsteuerung, das Sie in der Hauptgruppe des Programm-Manager-Fensters finden.
2. Doppelklicken Sie auf das Symbol ODBC, das Sie im Systemsteuerung-Fenster finden.
Es erscheint das Dialogfeld Data Sources.
3. Wählen Sie die Datenquelle aus dem Listenfeld Data Sources (Drivers) aus.
4. Wählen Sie Delete und dann Ja zur Bestätigung des Löschens.

Siehe auch

[Den Lotus-Notes-ODBC-Treiber installieren](#)
[Verbindung zu einer Lotus-Notes-Datenquelle herstellen](#)

Den Lotus-Notes-ODBC-Treiber installieren

1. Doppelklicken Sie auf das Symbol der Systemsteuerung, das Sie in der Hauptgruppe des Programm-Manager-Fensters finden.
Das Fenster der Systemsteuerung erscheint.
2. Doppelklicken Sie auf das ODBC-Symbol.
Das Dialogfeld Data Sources erscheint.
3. Wählen Sie Drivers.
Das Dialogfeld Drivers erscheint.
4. Wählen Sie Add.
Das Dialogfeld Add Drivers erscheint.
5. Geben Sie in dem Textfeld den Namen des Laufwerks und Verzeichnisses ein, das den Notes-Treiber enthält. Wählen Sie OK oder Browse, um einen Laufwerks- und Verzeichnisnamen auszuwählen.
Es erscheint das Dialogfeld Install Drivers.
6. Aus dem Listenfeld Available ODBC Drivers wählen Sie Lotus Notes.
7. Wählen Sie OK.
Der Notes-Treiber ist installiert.

Den Lotus-Notes-ODBC-Treiber löschen

1. Doppelklicken Sie auf das Symbol der Systemsteuerung, das Sie in der Hauptgruppe des Programm-Manager-Fensters finden.
2. Doppelklicken Sie auf das ODBC-Symbol.
Es erscheint das Dialogfeld Data Sources.
3. Wählen Sie Drivers.
Das Dialogfeld Drivers erscheint.
4. Wählen Sie Lotus Notes aus dem Listenfeld Installed ODBC Drivers.
5. Wählen Sie Delete.
Sie werden mit einer Mitteilung aufgefordert, zu bestätigen, daß Sie den Treiber und alle Datenquellen, die den Treiber verwenden, löschen möchten.
6. Wählen Sie Yes.

Lotus-Notes-ODBC-Treiber

Für alle Benutzer

Die folgenden Themen behandeln den Notes-Treiber und dessen Installation.

[Übersicht](#)

[Hardware- und Software-Anforderungen](#)

[Den Lotus-Notes-ODBC-Treiber installieren](#)

[Eine Lotus-Notes-Datenquelle hinzufügen, bearbeiten und löschen](#)

[Verbindung zu einer Lotus-Notes-Datenquelle herstellen](#)

[Der Lotus-Notes-ODBC-Treiber](#)

Für erfahrene Benutzer

Die folgenden Themen behandeln die direkte Anwendung des Notes-Treibers.

[Verbindungszeichenfolgen](#)

[SQL-Anweisungen](#)

[Datentypen](#)

[Fehlermeldungen](#)

Für Programmierer

Die folgenden Themen liefern Programmierinformationen zu dem Notes-Treiber. Diese Themen bieten Anwendungsprogrammierern mit Kenntnissen des ODBC API (Open Database Connectivity Application Programming Interface) hilfreiche Informationen.

[SQLGetInfo-Rückgabewerte](#)

[ODBC-API-Funktionen](#)

Hardware- und Software-Anforderungen

Zur Abfrage von Daten benötigen Sie

- Den Lotus-Notes-ODBC-Treiber
- Notes-Datenbankdateien
- Einen Computer, der mit MS-DOS ab Version 3.3 betrieben wird
- Microsoft Windows ab Version 3.1
- Den ODBC Driver Manager 1.0 (ODBC.DLL)

Die Workstation-Software Notes Version 3 muß installiert sein und das Verzeichnis im Pfad für ausführbare Dateien aufgenommen sein.

Die Notes-Datenbankdateien können sich auf einem Server befinden. Sie benötigen keine lokalen Kopien dieser Dateien, aber Sie müssen zumindest über Notes einen Lesezugriff darauf haben.

Zum Hinzufügen, Ändern oder Löschen von Treibern oder Datenquellen sollte die Option ODBC in der Systemsteuerung auf Ihrem Computer installiert sein.

Weitere Informationen über Notes-Datenbanken entnehmen Sie Ihrer Lotus-Notes-Dokumentation.

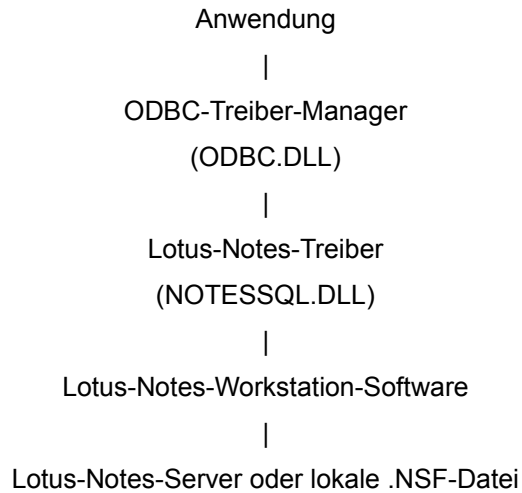
Siehe auch

[Den Lotus-Notes-ODBC-Treiber installieren](#)

Übersicht

Der Lotus-Notes-ODBC-Treiber ermöglicht es Ihnen, auf eine Notes-Datenbank über die Schnittstelle Open Database Connectivity (ODBC) zuzugreifen.

Die Anwendungs-/Treiberarchitektur ist:



Hinweis Der Notes-Treiber kommuniziert ausschließlich mit der Lotus-Notes-Workstation-Software. Lotus-Notes kümmert sich um den gesamten Netzwerkverkehr und die Sicherheit.

Siehe auch

[Hardware- und Software-Anforderungen](#)

[Den Lotus-Notes-ODBC-Treiber installieren](#)

[Eine Lotus-Notes-Datenquelle hinzufügen, ändern und löschen](#)

[Verbindung zu einer Lotus-Notes-Datenquelle herstellen](#)

[Der Lotus-Notes-ODBC-Treiber](#)

ODBC

ODBC (Open Database Connectivity) ist eine Schnittstelle, die Anwendungen ermöglicht, auf Daten in SQL-Datenbankverwaltungen zuzugreifen. Die Schnittstelle ermöglicht einer einzigen Anwendung, über ein Standard-Protokoll eine Verbindung zu vielen unterschiedlichen Datenbanktypen herzustellen. ODBC besteht aus einem Treiber-Manager und mehreren Treibern. Jeder Treiber verbindet die Anwendung mit einer bestimmten Datenbank.

Die Universal Relation Table

- Ein Feldname kann in einer Notes-Datenbank in mehreren Ausführungen mit unterschiedlichen Datentypen vorkommen. Daher müssen Sie in einer SELECT-Klausel Feldnamen angeben, da diese in der Universal Relation Table nachsieht. Zum Beispiel können Sie nicht SELECT * eingeben. Auf Felder in der Universal Relation Table können Sie nur Textoperationen ausführen, da der Datentyp für alle Felder Zeichen ist.
- Die Universal Relation Table kann an einer Selbst-Verbindung teilhaben, jedoch nicht an irgendeiner anderen Verbindungsart.
- Die Universal Relation Table unterstützt nur Anweisungen mit SELECT und CREATE VIEW. Sie können in einer Universal Relation Table keine INSERT-, DELETE- oder UPDATE-Anweisung ausführen.
- Sie können die Universal Relation Table nicht mit der Anweisung DROP TABLE löschen.

Die Universal Relation Table

Die Universal Relation Table enthält alle Felder in der Notes-Datenbank. SQL-Tabellen, die mit dem Notes-Treiber erstellt werden, ähneln mehr SQL-Ansichten statt traditioneller relationeller Datenbanktabellen.

Zum Beispiel können Sie mit dem Notes-Treiber und der Anweisung CREATE TABLE eine Notes-Maske erstellen. Die Anweisung DROP TABLE löscht die Notes-Maske, aber es werden keine Daten aus der Datenbank gelöscht. Die Anwendung von DROP TABLE mit dem Lotus-Notes-Treiber ist vergleichbar mit dem Löschen einer SQL-Ansicht. Die Daten verbleiben in der Datenbank. Sie können die Daten durch andere Masken oder Ansichten, die dieselben Feldnamen verwenden, ansehen oder durch Verweis über die Universal Relation Table.

Erstellen Sie mit dem Notes-Treiber eine neue Tabelle, die denselben Namen wie eine zuvor gelöschte Tabelle hat, und verwenden für die neue Tabelle einige Feldnamen der gelöschten Tabelle, kann die neue Tabelle schon Daten enthalten, bevor Sie irgendetwas eingegeben haben. Dies liegt daran, daß die Tabelle in Wirklichkeit eine Ansicht vorhandener Daten aus der Universal Relation Table darstellt. Felder in Notes-Datenbanken enthalten einen Verweis auf die Maske, die zu ihrer Erstellung verwendet wurde. Der Notes-Treiber verwendet den Maskennamen, der mit dem Feld gespeichert wird, um das Feld bei einer Auswahl aus einer Maske zu erkennen.

Die Universal Relation Table

Verbindung zu einer Lotus-Notes-Datenquelle herstellen

Ist Ihre Notes-ID kennwortgeschützt, fordert Sie Notes jedesmal zur Eingabe eines Kennwortes auf, wenn Sie versuchen, mit dem Lotus-Notes-ODBC-Treiber eine Verbindung zu einer entfernten Datenbank herzustellen. Das Kennwort ist nicht Datenbank-spezifisch und wird nicht behalten, so daß Sie es mehrfach eingeben müssen, wenn Sie eine Verbindung zu mehreren Datenbanken herstellen möchten.

Notes fordert Sie niemals zur Eingabe Ihrer Notes-ID auf. Ihre Notes-ID ist ein Bestandteil Ihrer Workstation-Installation.

Beispiel: Anwendung von Masken und Ansichten im Vergleich

Als Beispiel für einen Vergleich der Anwendung von Masken und Ansichten in einer Datenbank ist das Namen- und Adreßbuch von Notes gut eignet. Die Datenbank Namen- und Adreßbuch enthält:

- Eine Maske mit der Bezeichnung Person
- Eine Ansicht mit dem Namen Personen und einem Sortierschlüssel auf LastName.

Die folgende Anweisung ermöglicht die Suche nach Personen aus dem nach Nachnamen sortierten Namen- und Adreßbuch auf die effizienteste Weise:

```
SELECT LastName
FROM Personen
ORDER BY LastName
```

Personen ist eine Notes-Ansicht. Diese Abfrage ist effizient, da der Lotus-Notes-ODBC-Treiber den Index verwenden kann, der bereits mit der Ansicht Personen verknüpft ist, die die Nachnamen in der richtigen Reihenfolge auflistet.

Nehmen wir zum Beispiel an, daß Sie Personen nach deren Anschriften sortieren möchten. Sie könnten dazu die folgende Anweisung verwenden:

```
SELECT LastName, Mail_Address
FROM Personen
ORDER BY Mail_Address
```

Da die Ansicht Personen nicht nach Mail_Address sortiert ist, verwendet der Notes-Treiber den Personen-Index, erzeugt eine temporäre Datenbank und erstellt einen temporären Index nach Mail_Address. Das führt zu einer langsameren Performance.

Die folgende Anweisung erreicht dasselbe Ergebnis effizienter:

```
SELECT LastName, Mail_Address
FROM Personen
ORDER BY Mail_Address
```

Personen ist eine Notes-Maske. Gibt es keinen Index auf Mail_Address, erzeugt der Notes-Treiber einen temporären Index auf Mail_Address, muß aber keine temporäre Datenbank erzeugen. Diese Anweisung ist schneller als die vorherige Anweisung, die ORDER BY auf eine Tabelle angewendet hat, die auf einer Ansicht basiert. Diese Anweisung wird sogar noch schneller ausgeführt, wenn der Benutzer mit der Anweisung CREATE INDEX im Notes-Treiber einen Index erstellt oder über die Ansicht-Erstellung von Notes einen Index erstellt.

Hinweis Um eine Ansicht einer Datenbank zu erstellen, die sich auf einem Server befindet, benötigen Sie Zugangsrechte als Gestalter.

Reservierte Wörter

Vermeiden Sie die Verwendung der reservierten Wörter, die in Anhang C der *Microsoft ODBC Programmers Reference* als Bezeichner (Tabellen- oder Spaltennamen) aufgeführt sind.

Schließen Sie reservierte Wörter in Anführungszeichen (" ") ein, wenn Sie diese verwenden müssen. Zur Verwendung des reservierten Wortes DATE geben Sie zum Beispiel "DATE" ein.

Spaltennamen

Verwenden Sie in SQL-Anweisungen keine Spaltennamen, die andere Zeichen enthalten als alphabetische Zeichen, numerische Zeichen, Dollarzeichen (\$) oder Unterstriche (_). Es ist jedoch jedes Zeichen zulässig, wenn Sie den Spaltennamen in Anführungszeichen einschließen (" ").

SQL-Tabellen, Ansichten und Indizes auf Notes abbilden

- Jede SQL-Tabelle entstammt einer Notes-Maske.
- Jeder SQL-Index entstammt einer Notes-Ansicht, in der alle sortierten Spalten auf Feldern in einer einzigen Maske abgebildet sind und bei der Dokumente nur über diese Maske ausgewählt werden.
- Jede SQL-Ansicht entstammt einer Notes-Ansicht, die Dokumente nur über eine einzige Maske auswählt.

Copyright

Kein Teil der Dokumentation oder Software darf kopiert, fototechnisch übertragen, reproduziert, übersetzt oder auf einem anderen elektronischen Medium gespeichert bzw. in maschinell lesbare Form gebracht werden (außer wie in der Dokumentation beschrieben). Hierzu ist in jedem Fall zuvor die ausdrückliche schriftliche Zustimmung von Lotus Development Corporation einzuholen.

©Copyright 1990, 1994

Lotus Development Corporation

55 Cambridge Parkway

Cambridge, MA 02142

Alle Rechte vorbehalten

Lotus und Lotus Notes sind eingetragene Warenzeichen von Lotus Development Corporation.

Microsoft ist ein eingetragenes Warenzeichen und Windows ist ein Warenzeichen von Microsoft Corporation.

© Copyright 1994 Casahl Technology, Inc. Alle Rechte vorbehalten.

Casahl ist ein eingetragenes Warenzeichen von Casahl Technology, Inc.

Definition der Spalten von Notes-Ansichten

Bezieht sich eine Spalte in einer Ansicht nicht direkt auf ein Feld, benennt der Lotus-Notes-ODBC-Treiber die Spalte im SQLColumns-Ergebnissatz mit dem Dollarzeichen (\$), dem eine Zahl folgt (zum Beispiel \$2).

Enthält eine Spaltendefinition einer Ansicht eine der nachfolgend aufgeführten @Funktionen, liefert der Notes-Treiber für diese Spalte keine Ergebnisse. Diese Spalten zeigen Daten an, wenn Sie von Notes aus betrachtet werden, die Daten stehen aber nicht durch eine SQL-Abfrage zur Verfügung.

@All

@DeleteDocument

@DeleteFields

@DocChildren

@DocLevel

@DocNumber

@DocParentNumber

@DocSiblings

@Error

@IsCategory

@IsExpandable

@Unavailable

Performance des Lotus-Notes-ODBC-Treibers optimieren

Die Performance des Treibers ist abhängig von Ihrer Hardware- und Software-Umgebung und anderen Faktoren wie:

- Zugriff auf die Datenbank
- Art der Tabelle, die Sie abfragen
- Entwurf der Notes-Datenbank
- Auswahlkriterien, die Sie angegeben haben

Datenbankzugriff

- Die Performance bei der Arbeit mit Server-Datenbanken wird von der Art der von Ihnen verwendeten Netzwerk-Hardware und -Software und der Menge der Aktivitäten auf dem Netzwerk beeinflusst. Eine Abfrage auf eine Server-Datenbank, die ein kleines Ergebnis liefert, hat in der Regel eine bessere Performance als dieselbe Abfrage auf eine lokale Datenbank.
- Haben Sie Gestalter-Zugriff auf die Server-Datenbank, kann der Notes-Treiber eine temporäre Ansicht auf dem Server erstellen, um die Daten zu sortieren. Wenn Sie keinen Gestalter-Zugriff auf die Datenbank haben und versuchen, ohne einen Index nach einem Feld zu sortieren, kopiert der Notes-Treiber die Daten in eine temporäre lokale Notes-Datenbank und erstellt die Ansicht auf die temporäre Kopie.

Notes-Tabellentypen

Der Notes-Treiber kann Notes-Masken und -Ansichten als Tabellen unterscheiden. Zusätzlich erkennt der Treiber die Universal Relation Table von Notes. Jedoch haben die Masken und Ansichten von Notes sehr unterschiedliche Merkmale, die sich auf die Geschwindigkeit des Datenzugriffs mit dem Lotus-Notes-ODBC-Treiber auswirken.

Hinweis Notes-Dokumente, die mit einer Maske erstellt wurden bei der die Option Store form in document in den Maskenattributen von Notes nicht ausgewählt war, erscheinen nicht in Ansichten, die Dokumente auswählen, die diese Maske verwenden.

Die Ansichten von Notes sind indiziert, und Abfragen auf Ansichten können die Indizes zur Verbesserung der Performance verwenden. Sie können die Performance optimieren, indem Sie vorhandene Notes-Ansichten ändern oder neue Notes-Ansichten erstellen, um die Daten in Notes zu sortieren statt die Daten mit dem Lotus-Notes-Treiber zu sortieren. (Sie benötigen den Gestalterzugriff, um Ansichten auf einem Server zu erstellen oder verändern).

Beispiel

Anwendung von Masken und Ansichten im Vergleich

Richtlinien zum Datenbankentwurf

Hier sind einige Richtlinien zum Erstellen von Ansichten in Notes, um die Performance des Notes-Treibers zu verbessern:

- Bevor Sie einen Index erstellen, sollten Sie die folgenden Informationen berücksichtigen. Ein Index zu einer Tabelle, die im Vergleich zu der Größe der gesamten Datenbank klein ist, beschleunigt den Zugriff. Hat die Datenbank eine einzelne Tabelle, kann der Index den Zugriff verlangsamen.

Verwendet eine WHERE-Klausel den Index und wählt der Index eine Untermenge (20% oder weniger) der Dokumente in der Datenbank aus, dann verbessert die Verwendung des Index die Zugriffszeit. Ein Index auf die gesamte Datenbank (SELECT @ALL) verlangsamt den Zugriff.

- Gestalten Sie die Notes-Ansicht so, daß die Spalten in der folgenden Reihenfolge erscheinen:

Kategorisierte Spalten

Sortierte, nicht kategorisierte Spalten

Nicht sortierte Spalten

- Vermeiden Sie die Verwendung der Option "Maske in Dokument speichern" aus dem Dialogfeld Maskenattribute. Dokumente, die mit dieser Option erstellt werden, erscheinen nicht in der Tabelle, die von der Maske abgeleitet ist.

Sie können Daten von Masken abrufen, die die Option "Maske in Dokument speichern" verwenden, indem Sie eine Ansicht erstellen und dabei eine der folgenden Formeln verwenden:

```
SELECT $TITLE = Maskenname
```

```
SELECT@ALL
```

- Erstellen Sie nicht mehr Masken und Ansichten als erforderlich. Die Verbindung zu einer Datenbank mit vielen Masken und Ansichten kann verhältnismäßig langsam sein.
- Vermeiden Sie in Masken die Verwendung komplizierter Formeln, da Notes diese Formeln auswertet, wenn Sie eine INSERT- oder UPDATE-Anweisung erteilen.
- Ändern Sie die Aktualisierungshäufigkeit im Dialogfeld Indexoptionen aus Gestaltung/Ansichtattribute ändern in Manuell/Hintergrund, wenn Sie mit dem Notes-Treiber viele Einfügungen durchzuführen haben. Wenn die Indexoption auf Automatisch eingestellt ist, aktualisiert Notes den Index jedesmal, wenn Sie ein Dokument mit einer INSERT- oder UPDATE-Anweisung hinzufügen oder ändern.
- Der Notes-Treiber unterscheidet die Groß-/Kleinschreibung. Schreiben Sie Masken- und Ansichtennamen in Großbuchstaben, um Probleme bei Vergleichen zu vermeiden.
- Zur Erkennung von Ansichten, die speziell für den SQL-Zugriff erstellt wurden, können Sie sich eine Namenscodierung überlegen. Zum Beispiel können Sie alle Ansichtennamen mit den Buchstaben "SQL" beginnen.

Richtlinien zu Abfragen

Diese Richtlinien sollten Sie beim Erstellen einer Abfrage berücksichtigen.

- Versuchen Sie, Auswahlkriterien nur auf kategorisierte und sortierte Spalten anzuwenden.
- Vermeiden Sie es, eine Tabelle auszuwählen, die auf einer Notes-Ansicht basiert, und eine andere Sortierreihenfolge anzugeben. Wenn Sie bei einer vorhandenen Ansicht eine andere Sortierreihenfolge wählen, erstellt Notes eine Kopie der Daten in einer lokalen Datenbank und eine Ansicht zum Sortieren der Daten.
- Wenn Sie in einer SELECT-Anweisung eine WHERE- oder ORDER-BY-Klausel verwenden, ist die Abfrage viel schneller, wenn ein Index für die Spalten vorhanden ist, die in der WHERE- oder ORDER-BY-Klausel angegeben sind.

Notes-Ansichten als Indizes verwenden

Der Notes-Treiber kann Notes-Ansichten wie Indizes verwenden, wenn diese die folgenden Merkmale erfüllen:

- Die Auswahlformel der Ansicht lautet SELECT Form = "Name". Sie können SELECT @All verwenden, aber dies ist viel langsamer.
- Jede Spalte in der Ansicht ist als einfacher Verweis auf ein Feld in dieser Maske definiert (keine Formeln).
- Zumindest eine Spalte ist sortiert.
- Jeder Name einer Ansicht, die als Index verwendet wird, muß eindeutig sein.

SQLGetInfo Rückgabewerte (Programmierung)

In der folgenden Tabelle sind die #defines der C-Sprache für das Argument flnfoType und die entsprechenden Rückgabewerte von SQLGetInfo aufgeführt. Sie können diese Information abfragen, indem Sie das #define der C-Sprache im Argument flnfoType an SQLGetInfo übergeben.

<u>Wert von flnfoType (#define)</u>	<u>Rückgabewert des Lotus-Notes-Treibers</u>
SQL_ACCESSIBLE_PROCEDURES	Nein
SQL_ACCESSIBLE_TABLES	Nein
SQL_ACTIVE_CONNECTIONS	Keine Begrenzung
SQL_ACTIVE_STATEMENTS	Keine Begrenzung
SQL_CONCAT_NULL_BEHAVIOR	Ergebnis wird als NULL-Wert zurückgeliefert
SQL_CONVERT_BIGINT	Keiner
SQL_CONVERT_BINARY	Keiner
SQL_CONVERT_BIT	Keiner
SQL_CONVERT_CHAR	CHAR, VARCHAR, LONGVARCHAR, DATE, TIME oder TIMESTAMP
SQL_CONVERT_DATE	DATE, VARCHAR, LONGVARCHAR oder TIMESTAMP
SQL_CONVERT_DECIMAL	CHAR, NUMERIC, DECIMAL, INTEGER, SMALLINT, FLOAT, REAL, DOUBLE, VARCHAR oder LONGVARCHAR
SQL_CONVERT_DOUBLE	CHAR, NUMERIC, DECIMAL, INTEGER, SMALLINT, FLOAT, REAL, DOUBLE, VARCHAR oder LONGVARCHAR
SQL_CONVERT_FLOAT	CHAR, NUMERIC, DECIMAL, INTEGER, SMALLINT, FLOAT, REAL, DOUBLE, VARCHAR oder LONGVARCHAR
SQL_CONVERT_FUNCTIONS	Nicht unterstützt
SQL_CONVERT_INTEGER	CHAR, NUMERIC, DECIMAL, INTEGER, SMALLINT, FLOAT, REAL, DOUBLE, VARCHAR oder LONGVARCHAR
SQL_CONVERT_LONGVARBINARY	Keiner
SQL_CONVERT_LONGVARCHAR	CHAR, VARCHAR, LONGVARCHAR, DATE, TIME oder TIMESTAMP
SQL_CONVERT_NUMERIC	CHAR, NUMERIC, DECIMAL, INTEGER, SMALLINT, FLOAT, REAL, DOUBLE, VARCHAR oder LONGVARCHAR
SQL_CONVERT_REAL	CHAR, NUMERIC, DECIMAL, INTEGER, SMALLINT, FLOAT, REAL, DOUBLE, VARCHAR oder LONGVARCHAR
SQL_CONVERT_SMALLINT	CHAR, NUMERIC, DECIMAL, INTEGER, SMALLINT, FLOAT, REAL, DOUBLE, VARCHAR oder LONGVARCHAR
SQL_CONVERT_TIME	CHAR, VARCHAR, LONGVARCHAR, TIME

	oder TIMESTAMP
SQL_CONVERT_TIMESTAMP	CHAR, VARCHAR, LONGVARCHAR, DATE, TIME oder TIMESTAMP
SQL_CONVERT_TINYINT	Keiner
SQL_CONVERT_VARBINARY	Keiner
SQL_CONVERT_VARCHAR	CHAR, VARCHAR, LONGVARCHAR, DATE, TIME oder TIMESTAMP
SQL_CURSOR_COMMIT_BEHAVIOR	Vorbelegt
SQL_CURSOR_ROLLBACK_BEHAVIOR	Schließen/Löschen
SQL_DATA_SOURCE_NAME	Datenquellenname aus SQLConnect
SQL_DATA_SOURCE_READ_ONLY	NOTESSQL prüft die Quelle nicht daraufhin, ob diese schreibgeschützt ist.
SQL_DATABASE_NAME	Datenbankname, der über SQLConnect gefunden wird
SQL_DBMS_NAME	Lotus Notes
SQL_DBMS_VER	Lotus Notes Version 3
SQL_DEFAULT_TXN_ISOLATION	SQL_TXN_REPEATABLE_READ
SQL_DRIVER_NAME	NOTESSQL.DLL
SQL_DRIVER_VER	Version des installierten Treibers
SQL_EXPRESSIONS_IN_ORDERBY	Nein
SQL_FETCH_DIRECTION	NEXT
SQL_IDENTIFIER_CASE	Namen können in gemischter Schreibweise vorliegen, und die Groß-/Kleinschreibung wird beachtet.
SQL_IDENTIFIER_QUOTE_CHAR	Doppelte Anführungszeichen (""")
SQL_MAX_COLUMN_NAME_LEN	32
SQL_MAX_CURSOR_NAME_LEN	32
SQL_MAX_OWNER_NAME_LEN	0
SQL_MAX_PROCEDURE_NAME_LEN	0
SQL_MAX_QUALIFIER_NAME_LEN	0
SQL_MAX_TABLE_NAME_LEN	32
SQL_MULT_RESULT_SETS	Nicht unterstützt
SQL_MULTIPLE_ACTIVE_TXN	Nicht unterstützt
SQL_NUMERIC_FUNCTIONS	Nicht unterstützt
SQL_ODBC_API_CONFORMANCE	Level1
SQL_ODBC_SAG_CLI_CONFORMANCE	SAG-konform
SQL_ODBC_SQL_CONFORMANCE	Minimum
SQL_ODBC_SQL_OPT_IEF	Nein
SQL_ODBC_VER	Version des verwendeten ODBC-Treiber-

	Managers
SQL_OUTER_JOINS	Nein
SQL_OWNER_TERM	Nicht unterstützt
SQL_PROCEDURE_TERM	Nicht unterstützt
SQL_PROCEDURES	Nicht unterstützt
SQL_QUALIFIER_NAME_SEPARATOR	Keiner
SQL_QUALIFIER_TERM	Keiner
SQL_ROW_UPDATES	Nein
SQL_SCROLL_CONCURRENCY	SQL_SCCO_OPT_VALUES
SQL_SCROLL_OPTIONS	SQL_SO_FORWARD_ONLY
SQL_SEARCH_PATTERN_ESCAPE	Rückstrich (\)
SQL_SERVER_NAME	Server-Name von SQLConnect
SQL_STRING_FUNCTIONS	Nicht unterstützt
SQL_SYSTEM_FUNCTIONS	Nicht unterstützt
SQL_TABLE_TERM	FORM
SQL_TIMEDATE_FUNCTIONS	Nicht unterstützt
SQL_TXN_CAPABLE	Transaktionen werden nicht unterstützt
SQL_TXN_ISOLATION_OPTION	Nicht unterstützt
SQL_USER_NAME	Keiner

ODBC-API-Funktionen (Programmierung)

Der Lotus-Notes-ODBC-Treiber unterstützt alle Core- und Level-1-Funktionen mit Ausnahme der folgenden:

- SQLGetStmtOption
- SQLSetStmtOption
- SQLTransact

Nachfolgend wird beschrieben wie bestimmte Funktionen im Notes-Treiber implementiert sind.

SQLCancel

SQLSetStmtOption wird nicht unterstützt. Die Funktion SQLCancel (ohne die Option ASYNC_ENABLE von SQLSetStmtOption) ist hilfreich zur Unterbrechung einer SQLPutData-Funktion, wenn die Rückgabewerte eines langen Datenfeldes zusätzliche SQLPutData-Funktionen erfordert.

Rückgabewerte:

SQL_INVALID_HANDLE

SQLColumns

Bezieht sich eine Spalte einer Ansicht nicht direkt auf ein Feld, benennt der Notes-Treiber die Spalte im Ergebnissatz mit einem Dollarzeichen (\$), dem eine Zahl folgt (zum Beispiel \$2). Ein Aufruf von SQLColumns liefert nur die ersten 32 Zeichen des Bemerkungsfelds zurück. Das Bemerkungsfeld entspricht der Hilfebeschreibung für einen Feldsatz in der Maske, die mit dem Dialogfeld Field Definition erstellt wird.

SQLConnect

Der Notes-Treiber baut während der Ausführung von SQLConnect für die Datenbank Systemkataloge auf. Diese Kataloge werden aus den Notes-Masken- und -Ansicht-Gestaltungsdokumenten erstellt. Daher erscheint der Verbindungsaufbau zu einer Datenbank, die viele Masken und Ansichten enthält, langsam.

SQLSetConnectOption

Diese Funktion erlaubt die folgenden Einstellungen:

SQL_AUTOCOMMIT = 1

SQL_ACCESS_MODE = SQL_MODE_READ_WRITE

Alle anderen Einstellungen führen zu einem Fehler. SQLSetConnectOption prüft nur, ob Sie eine der zuvor gezeigten Einstellungen angefordert haben, und wirkt sich nicht auf den Status des Treibers aus.

SQLTables

Zulässige Tabellentypen sind:

FORM

SYNONYM

TABLE

VIEW

UNIVERSAL

Eine Notes-Datenbank enthält eine Tabelle, die denselben Namen wie die Datenbank trägt und als Universal Relation Table bezeichnet wird. Die Universal Relation Table enthält alle Felder der Notes-Datenbank. SQL-Tabellen, die mit dem Notes-Treiber erstellt werden, sind eher mit SQL-Ansichten vergleichbar als mit herkömmlichen relationalen Datenbanktabellen.

SQLTransact

SQLTransact wird eingeschränkt unterstützt. Da Notes keinen Transaktionsmechanismus anbietet, wird SQL_COMMIT unterstützt, aber SQL_ROLLBACK führt zu einem Fehler.

Zusammenfassung der unterstützten ODBC-SQL-Grammatik

Der Lotus-Notes-ODBC-Treiber unterstützt die folgende ODBC-SQL-Grammatik:

Unterstützte Anweisungen

ALTER TABLE

CREATE INDEX

CREATE TABLE

CREATE VIEW

DELETE mit Suche

DELETE mit Positionierung

DROP INDEX

DROP TABLE

DROP VIEW

INSERT

SELECT

FROM

WHERE

FOR UPDATE

GROUP BY

HAVING

ORDER BY

UPDATE mit Suche

UPDATE mit Positionierung

Unterstützte Ausdrücke, Funktionen und Operatoren

Numerische Operatoren

Prädikat-Operatoren

Spaltenfunktionen

<Unterabfrage> ist eine SELECT-Anweisung mit der Einschränkung, daß in der SELECT-Klausel nur eine Spalte bestimmt werden kann.

FOR UPDATE

Die Klausel FOR UPDATE bestimmt eine Liste von Spaltennamen.

Syntax

<FOR UPDATE Klausel> ist FOR UPDATE OF <*Spaltenliste*>

Siehe Konventionen

Beispiel

ORDER BY

Die Klausel ORDER BY legt fest, wie die Datensätze, die mit einer Abfrage abgerufen wurden, zu sortieren sind.

Syntax

<ORDER BY Klausel> ist ORDER BY <Spaltenname> | <Integer> [ASC | DESC] [, ...]

Siehe Konventionen

Beispiel

HAVING

Die Klausel HAVING bestimmt eine Suchbedingung für eine Gruppe.

Syntax

<HAVING Klausel> ist HAVING <Suchbedingung>

Siehe Konventionen

Die Suchbedingung muß eine Spaltenfunktion beinhalten.

Beispiel

GROUP BY

Die Klausel GROUP BY gruppiert die Daten aus den Quellentabellen nach einem oder mehreren Spaltennamen und produziert für jeden angegebenen Spaltennamen eine einzelne Summenzeile.

Syntax

<GROUP BY Klausel> ist GROUP BY <Spalte list>

Siehe Konventionen

Diese Liste kann keine abgeleiteten Spalten enthalten. Alle Spalten aus der Klausel GROUP BY müssen auch in der SELECT-Klausel erscheinen.

Beispiel

Numerische Operatoren

Der Lotus-Notes-ODBC-Treiber unterstützt in Ausdrücken die folgenden numerischen Operatoren.

<u>Operator</u>	<u>Bedeutung</u>
+	Addition
-	Subtraktion
*	Multiplikation
/	Division

Prädikat-Operatoren

Der Lotus-Notes-ODBC-Treiber unterstützt die folgenden Prädikat-Operatoren.

<u>Operator</u>	<u>Bedeutung</u>
<	Kleiner als
>	Größer als
<=	Kleiner als oder gleich
>=	Größer als oder gleich
=	Gleich
<>	Ungleich
BETWEEN	Gibt einen Wertebereich zwischen einer unteren und oberen Grenze an.
IN	Gibt ein Element einer Werteliste oder einer Unterabfrage an.
LIKE	Wird zum Vergleich mit einem Muster verwendet. Im LIKE-Prädikat gibt es folgende Jokerzeichen: Ein Unterstrich (_) steht für ein einzelnes Zeichen. Ein Prozentzeichen (%) stellt eine beliebige Anzahl Zeichen dar. Ein Rückstrich (\) ersetzt das Escape-Zeichen.
NOT	Den Operator NOT verwenden Sie zusammen mit einem anderen Operator, um eine negierte Suchbedingung anzugeben. Zum Beispiel: NOT IN, NOT LIKE oder NOT BETWEEN.
ANY	Vergleiche jeden Wert, der von einer Unterabfrage zurückgeliefert wird, mit dem angegebenen Wert. Die Operatoren =, <>, >, >=, <, <= und =. können davor angegeben werden. =ANY entspricht IN. <>ANY entspricht NOT IN.
SOME	SOME ist ein anderes Schlüsselwort für

	ANY.
ALL	Vergleiche jeden Wert, der von einer Unterabfrage zurückgeliefert wird, mit dem angegebenen Wert. Die Operatoren =, <>, >, >=, < und <= können davor angegeben werden.
EXISTS	"Wahr", wenn eine Unterabfrage zumindest einen Datensatz zurückgeliefert hat.

Spaltenfunktionen

Spaltenfunktionen können ein Bestandteil einer SELECT-Klausel sein. Eine Spaltenfunktion nimmt als Argument eine ganze Spalten mit Daten und produziert ein einzelnes Datenelement, das die Spalte aufsummiert. Zum Beispiel berechnet die Spaltenfunktion AVG den Mittelwert der Daten aus einer Spalte.

Sie können eine Spaltenfunktion mit einem Feldnamen oder in Kombination mit einem komplizierteren Ausdruck verwenden. Der Lotus-Notes-ODBC-Treiber unterstützt die folgenden Spaltenfunktionen.

<u>Funktion</u>	<u>Beschreibung</u>
AVG	Liefert in einem Ausdruck mit einem numerischen Feld den Mittelwert. Zum Beispiel liefert AVG(UMSATZ) den Mittelwert aller Umsatz-Felder.
COUNT	Liefert die Anzahl der Werte in einem beliebigen Ausdruck mit einem Feld.
MAX	Liefert den höchsten Wert in einem beliebigen Feldausdruck. Zum Beispiel liefert MAX(UMSATZ) den höchsten Wert aus dem Feld Umsatz.
MIN	Liefert aus einem beliebigen Feldausdruck den niedrigsten Wert. Zum Beispiel liefert MIN(UMSATZ) den niedrigsten Wert aus dem Feld Umsatz.
SUM	Liefert die Gesamtsumme der Werte eines Ausdrucks mit einem numerischen Feld. Zum Beispiel liefert SUM(UMSATZ) die Summe aller Werte im Feld Umsatz.

WHERE

Die Klausel WHERE legt die Bedingungen fest, die Datensätze erfüllen müssen, um in die Abfrage aufgenommen zu werden.

Syntax

<WHERE Klausel> ist WHERE [NOT] <Suchbedingung>[{ AND | OR } [NOT] <Suchbedingung> ...]

Siehe Konventionen

Beispiel

Argumente

<Suchbedingung> ist eines der folgenden Argumente:

- <Vergleichsprädikat> ist:
<Ausdruck> <Vergleichsoperator> <Ausdruck>
- <Bereichsprädikat> ist:
<Ausdruck> [NOT] BETWEEN <Ausdruck> AND <Ausdruck>
- <IN-Prädikat> ist:
<Ausdruck> [NOT] IN { <Ausdruck> | <Werteliste> }
- <Ähnlichkeitsprädikat> ist:
<Spaltenname> [NOT] LIKE <Suchmuster>
- <Mengenprädikat> ist:
<Ausdruck> <Vergleichsoperator> { ANY | SOME } Unterabfrage
- <ALL-Bedingung> ist:
<Ausdruck> <Vergleichsoperator> [NOT] ALL (<Unterabfrage>)
- <Existenzprädikat> ist:
WHERE [NOT] EXISTS (<Unterabfrage>)
- <IN-Vergleich mit Unterabfrage> ist:
WHERE <Ausdruck> [NOT] IN (<Unterabfrage>)
- <Verbindungsbedingung> ist:
<Spaltenausdruck> <Vergleichsoperator> <Spaltenausdruck>

Die Spaltenausdrücke sind im allgemeinen durch Tabellen- oder Aliasnamen näher bestimmt.
Spaltenausdrücke können kein Sternchen (*) enthalten.

Einschränkungen

Die WHERE-Klausel unterstützt die Rich-Text-Felder von Notes nicht. Notes läßt bei Rich-Text-Feldern keine Formeln zu. Da der Notes-Treiber die Suchbedingung an Notes übergibt und Notes nicht in der Lage ist, die Formel auszuwerten, werden keine Zeilen zurückgeliefert.

FROM

Die Klausel FROM bestimmt die Tabellennamen oder Ansichten der Quellentabellen für eine Abfrage.

Syntax

<FROM Klausel> ist FROM {<Tabellenverweis> [<Alias>]} [, ...]

Siehe Konventionen

Beispiel

Argumente

<Tabellenverweis> ist ein Tabellenname oder Ansichtename.

<Alias> ist ein Aliasname für den Tabellen- oder Ansichtennamen. Der Aliasname ist nur für diese Abfrage definiert.

UPDATE mit Positionierung

Die positionierte Update-Anweisung aktualisiert die zuletzt geholte Anweisung.

Syntax

```
UPDATE <Tabellenname> SET <Spaltenname> = <Wertausdruck> [ , ... ] WHERE CURRENT  
OF <Cursorname>
```

Siehe Konventionen

Beispiel

Argumente

<Tabellenname> ist der Name der Tabelle, die aktualisiert werden soll.

<Spaltenname> ist der Name einer Spalte in der Tabelle.

<Wertausdruck> ist der neue Wert für die Spalte. Es muß entweder ein Wertausdruck oder ein dynamischer Parameter sein.

<Cursorname> ist der Name des Cursors, der auf die Zeile zeigt, die aktualisiert werden soll.

UPDATE mit Suche

Die UPDATE-Anweisung mit Suche aktualisiert Werte in ausgewählten Zeilen einer Tabelle.

Syntax

```
UPDATE <Tabellenname> [ <Alias> ] SET <Spaltenname> = <Ausdruck> [ , ... ] [ <WHERE  
Klausel> ]
```

Siehe [Konventionen](#)

Beispiel

Argumente

<Tabellenname> ist der Name der Tabelle, die aktualisiert werden soll. Bei Verwendung von <Alias> kann dieses Argument nicht verwendet werden.

<Alias> ist ein optionaler Aliasname für die Tabelle. Bei Verwendung von <Tabellenname> kann dieses Argument nicht verwendet werden.

<Spaltenname> ist der Name einer Spalte in der Tabelle.

<Ausdruck> ist ein Ausdruck, dessen Auswertung einen neuen Wert für die Spalte ergibt.

<WHERE Klausel> gibt die Zeilen an, die aktualisiert werden sollen (siehe [SELECT](#)).

INSERT

Die INSERT-Anweisung fügt zu einer Tabelle oder Ansicht eine Zeile hinzu. Sie können die Werte explizit angeben oder aus einer SELECT-Anweisung erhalten. Der Wert, den Sie einer Spalte zuweisen, muß zu dem Datentyp der Spalte kompatibel sein. Geben Sie keinen Wert an, wird ein Vorgabewert zugewiesen, wenn ein solcher zur Verfügung steht.

Syntax

```
INSERT INTO <Tabellenname>[ ( <Spaltenliste> ) ] { VALUES ( <Werteliste> ) |  
<Abfragespezifikation> }
```

Siehe [Konventionen](#)

Beispiel

Argumente

<Tabellenname> ist der Name der Tabelle, in die eingefügt werden soll.

<Spaltenliste> ist eine optionale Liste der Spalten, die zu aktualisieren sind. Geben Sie keine Spaltenliste an, werden die Spalten in der Reihenfolge zugewiesen, in der sie in der Tabellendefinition erscheinen. Sie müssen die Spaltennamen durch Kommas trennen und die gesamte Liste in Klammern setzen.

<Werteliste> ist eine Liste von Werten, die in die Tabelle eingefügt werden sollen. Sie müssen die Werte mit Kommas voneinander trennen und die gesamte Liste in Klammern einschließen. Alle Werte müssen wie folgt angegeben werden:

- Zeichenfolgen-Konstante in Anführungszeichen
- Numerische Konstante
- Dynamischer Parameter
- NULL

Die INSERT-Anweisung muß entweder eine VALUES- oder eine SELECT-Klausel enthalten.

Funktionen zur Eingabe-Validierung

Wenn der Notes-Treiber einen neuen Datensatz in eine Tabelle einfügt, wertet er für jedes Feld drei Funktionen aus dem Maskenentwurf aus (Notes geht ebenso vor, wenn Sie ein Dokument zusammensetzen):

- Vorgabe-Wert
- Eingabe-Übersetzung
- Eingabe-Validierung

Die Felddefinition kann jede dieser Funktionen oder keine von ihnen enthalten. Diese Funktionen werden in der folgenden Reihenfolge ausgewertet:

1. Die Formel für den Vorgabe-Wert wird (falls vorhanden) ausgewertet, um einen Wert zu liefern, wenn die INSERT-Anweisung für dieses Feld keine Daten enthält.
2. Die Eingabe-Übersetzung-Formel wird ausgewertet.

Hinweis Die Eingabe-Übersetzung-Formel kann den Wert der eingefügten Daten verändern.

3. Die Eingabe-Validierungs-Formel wird ausgewertet.

Wenn eine INSERT-Anweisung mehrere Datensätze betrifft und eine oder mehrere der Einfügungen eine Eingabe-Validierungsprüfung nicht bestehen, liefern SQLExecute oder SQLExecDirect den Code SQL_SUCCESS_WITH_INFO zurück. Dadurch erfährt der Benutzer, was fehlgeschlagen ist, und

ermöglicht die Fortsetzung der zulässigen Einfügungen.

DROP VIEW

Die Anweisung DROP VIEW löscht eine Ansicht aus der aktuellen Datenbank.

Syntax

DROP VIEW <Ansichtename>

Siehe Konventionen

Beispiel

Argumente

<Ansichtename> ist der Name der Ansicht, die gelöscht werden soll.

Einschränkungen

- RESTRICT und CASCADE werden nicht unterstützt.
- Abhängige Objekte und Dokumente werden nicht gelöscht.

DROP TABLE

Die Anweisung DROP TABLE löscht eine Tabelle aus der aktuellen Datenbank.

Hinweis Die DROP TABLE-Anweisung löscht keine Daten aus der Datenbank. Weitere Informationen dazu siehe [Die Universal Relation Table](#).

Syntax

```
DROP TABLE <Tabellenname>
```

Siehe [Konventionen](#)

Beispiel

Argumente

<*Tabellenname*> ist der Name der Tabelle, die gelöscht werden soll.

Einschränkungen

- RESTRICT und CASCADE werden nicht unterstützt.
- Abhängige Objekte und Dokumente werden nicht gelöscht.

DROP INDEX

Die Anweisung DROP INDEX löscht einen Index aus der aktuellen Datenbank.

Syntax

```
DROP INDEX <Indexname>
```

Siehe Konventionen

Beispiel

Argumente

<*Indexname*> ist der Name des Indexes, der gelöscht werden soll.

Einschränkungen

- RESTRICT und CASCADE werden nicht unterstützt.
- Abhängige Objekte und Dokumente werden nicht gelöscht.

DELETE mit Positionierung

Die positionierte DELETE-Anweisung löscht die zuletzt erhaltene Zeile (die aktuelle Zeile).

Syntax

```
DELETE FROM <Tabellenname> WHERE CURRENT OF <Cursorname>
```

Siehe Konventionen

Beispiel

Argumente

<Tabellenname> ist der Name der Tabelle oder Ansicht, aus der eine Zeile gelöscht werden soll.

<Cursorname> ist der Cursor, der auf die Zeile zeigt, die gerade verarbeitet wird.

DELETE mit Suche

Die DELETE-Anweisung mit Suche löscht ausgewählte Zeilen aus einer Tabelle.

Syntax

```
DELETE FROM <Tabellenname> [ WHERE <WHERE-Klausel> ]
```

Siehe Konventionen

Beispiel

Argumente

<Tabellenname> ist der Name der Tabelle, aus der Sie Daten löschen möchten.

<WHERE-Klausel> gibt die Zeilen an, die gelöscht werden sollen. Geben Sie keine WHERE-Klausel an, werden alle Zeilen in der Tabelle gelöscht.

SELECT

Die SELECT-Anweisung wählt Zeilen und Spalten aus Tabellen aus, um diese entweder anzuzeigen oder diese als Eingabe für andere SQL-Anweisungen zu verwenden.

Syntax

```
SELECT <Vollauswahl> [ <ORDER BY Klausel> | <FOR UPDATE Klausel> ]
```

Siehe [Konventionen](#)

Argumente

<Vollauswahl> ist eines der folgenden Argumente:

- <SELECT Klausel>
- < [FROM Klausel](#) >
- [<[WHERE Klausel](#)>]
- [<[GROUP BY Klausel](#)>]
- [<[HAVING Klausel](#)>]

<SELECT Klausel> ist SELECT { * | <*Ausdruckliste*> }

<*Ausdruckliste*> ist eine Liste von Spaltennamen und anderer <Ausdrücke>, deren Werte in der Ergebnistabelle (Ergebnismenge) erscheinen.

Einschränkung

- Die UNION-Operation wird nicht unterstützt.
- Die Klausel DISTINCT wird in einer Spaltenfunktion nicht unterstützt.

CREATE VIEW

Die Anweisung CREATE VIEW definiert eine neue Ansicht der aktuellen Datenbank.

Syntax

```
CREATE VIEW <Ansichtname> [(<Spaltenliste>)] AS <Abfragespezifikation>
```

Siehe Konventionen

Beispiel

Argumente

<Ansichtname> ist der Name einer neuen Ansicht.

<Spaltenliste> ist eine Liste durch Kommas begrenzter Spaltennamen.

Einschränkungen

- Nur einzelne Basis-Tabellen (keine Ansicht auf eine Ansicht) werden unterstützt.
- Ausgewählte Spalten können keine Spaltenfunktionen sein.
- In einer Abfragespezifikation werden keine GROUP BY- oder HAVING-Klauseln unterstützt.
- In einer Ansichtendefinition werden keine verschachtelten Abfragen unterstützt.

CREATE TABLE

Die Anweisung CREATE TABLE definiert in der aktuellen Datenbank eine neue Tabelle.

Syntax

```
CREATE TABLE <Basis-Tabellenname> ( <Spaltenname> <Datentyp> [ DEFAULT  
<Vorgabewert> ] [ CHECK <Suchbedingung> ] )
```

Siehe Konventionen

Beispiel

Argumente

<Basis-Tabellenname> ist der Name der Tabelle, die erstellt werden soll.

<Spaltenname> ist der Name einer Spalte.

<Datentyp> ist einer der folgenden Datentypen:

- CHAR(n) oder CHARACTER(n) wobei $1 \leq n \leq 254$
- DECIMAL(p, s) wobei p (Genauigkeit) $1 \leq p \leq 15$ und s (Maßstab) $0 \leq s \leq p$ ist.
- NUMERIC(p, s) (wie DECIMAL)
- SMALLINT
- INTEGER
- REAL
- FLOAT
- DOUBLE PRECISION
- VARCHAR(n) oder CHARACTER VARYING(n) (wie CHAR)
- DATE (Format JJJJ-MM-TT)
- TIME (Format HH-MM-SS)
- TIMESTAMP (Format JJJJ-MM-TT HH:MM:SS)

Hinweis Die verkürzten ODBC-Escape-Sequenzen {d 'jjjj-mm-dd'}, {t 'hh:mm:ss'} und {dt 'jjjj-mm-dd hh:mm:ss'} werden nicht unterstützt.

Hinweis Der Millisekundenteil des TIMESTAMP-Wertes wird nicht unterstützt.

Einschränkungen

- NULL und NOT NULL werden nicht unterstützt.
- UNIQUE und PRIMARY KEY werden nicht unterstützt.
- Die Klausel REFERENCES wird nicht unterstützt.
- Eine Tabelleneinschränkungsdefinition wird nicht unterstützt.

CREATE INDEX

Die Anweisung CREATE INDEX definiert einen Index für eine Tabelle.

Syntax

```
CREATE INDEX <Indexname> ON <Basis-Tabellename>( <Spaltenname> [ ASC | DESC ] [ ,  
<Spaltenname> [ ASC | DESC ] ] ... )
```

Siehe [Konventionen](#)

Beispiel

Argumente

<Indexname> ist der Name des Index, der erstellt werden soll.

<Basis-Tabellename> ist der Name der Tabelle, die indiziert werden soll.

<Spaltenname> ist der Name einer Spalte, die in den Index aufgenommen werden soll. Die gesamte Liste der Spaltenspezifikationen wird in Klammern eingeschlossen und die Elemente durch Kommas voneinander getrennt. Der Datentyp der Spalte darf nicht VARCHAR sein.

ASC | DESC bestimmt die Reihenfolge des Index als aufsteigend (ASC) oder absteigend (DESC). Die aufsteigende Sortierreihenfolge ist voreingestellt.

Einschränkungen

Kein eindeutiger Index.

Ausnahmen von der ODBC-SQL-Grammatik

Der Lotus-Notes-ODBC-Treiber unterstützt die meisten SQL-Anweisungen und -Klauseln in der ODBC-Minimum- und -Core-Grammatik. Die folgende Tabelle beschreibt Ausnahmen:

<u>Grammatik</u>	Beschränkung
ALTER TABLE	Die folgenden Schlüsselwörter werden nicht unterstützt: NULL NOT NULL PRIMARY KEY REFERENCES Keine Definition von Tabelleneinschränkungen
CREATE INDEX	Das Schlüsselwort UNIQUE wird nicht unterstützt.
CREATE TABLE	Die folgenden Schlüsselwörter werden nicht unterstützt: NULL NOT NULL UNIQUE PRIMARY KEY REFERENCES Keine Definition von Tabelleneinschränkungen
CREATE VIEW	Nur einzelne Basis-Tabellen (keine Ansicht auf eine Ansicht) SELECT-Spalten können keine Spaltenfunktionen sein Keine GROUP BY- oder HAVING-Klausel in einer Abfragespezifikation Keine verschachtelten Abfragen in einer Ansichtdefinition
RESTRICT oder CASCADE	Nicht unterstützt in DROP INDEX-, DROP VIEW- und DROP TABLE-Anweisungen. Abhängige Objekte und Dokumente werden nicht gelöscht.
GRANT und REVOKE	Nicht unterstützt. Die gesamte Zugriffskontrolle erfolgt implizit durch Notes.
UNION	Nicht unterstützt in der SELECT-Anweisung
DISTINCT	Nicht unterstützt in einer select-Klausel oder einer Spaltenfunktion

DATE, TIME- und
TIMESTAMP-ODBC-
Escape-Klauseln

Die verkürzten ODBC-Escape-
Sequenzen {d 'jjjj-mm-tt'}, {t
'hh:mm:ss'} und {dt 'jjjj-mm-tt
hh:mm:ss'} werden nicht
unterstützt.

Rich-Text-Felder

Der Lotus-Notes-Treiber liefert von einem Rich-Text-Feld nur den Textteil (die Textteile) zurück. Der Notes-Treiber kann kein Rich-Text-Feld erstellen.

Die WHERE-Klausel unterstützt die Rich-Text-Felder von Notes nicht. Notes läßt keine Formeln bei Rich-Text-Feldern zu. Da der Notes-Treiber die Suchbedingung an Notes übergibt und Notes nicht in der Lage ist, die Formel auszuwerten, werden keine Zeilen zurückgeliefert.

Listenfelder (Felder mit Mehrfachwerten)

Der Lotus-Notes-Treiber unterstützt Mehrfachwerte in Feldern. Der Notes-Treiber enthält Informationen darüber, welche Felder mehrere Werte enthalten können und welche Datentypen diese Werte haben.

Der Notes-Treiber unterstützt keine Listenfelder in DDL-Anweisungen (CREATE TABLE, DROP TABLE, ALTER TABLE, CREATE INDEX, CREATE VIEW oder DROP VIEW).

Verwenden Sie in DML-Anweisungen (SELECT, INSERT, UPDATE mit Suche, UPDATE mit Positionierung, DELETE mit Suche, DELETE mit Positionierung) die folgende Notation zur Angabe einer Werteliste:

```
'String;String;String'
```

Textfelder

Der Notes-Treiber liefert alle Daten aus Feldern mit Mehrfachwerten, die aus Text bestehen, als einzelne Zeichenfolge, wobei die Elemente durch Semikolons voneinander getrennt sind. Zum Beispiel:

```
'a;b;c'
```

Hinweis Im Ergebnis kann ein zusätzliches Semikolon auftauchen, wenn bei der Eingabe der Information in Notes ein anderes Zeichen als ein Semikolon als Listentrennzeichen verwendet wurde und die Liste ein Element mit einem Semikolon darin enthält.

Der Notes-Treiber akzeptiert eine durch Semikolons begrenzte Liste von Zeichenfolgen zum Einfügen in Felder mit Mehrfachwerten, die aus Text bestehen. Zum Beispiel:

```
'a;b;c'
```

Dieser Wert erstellt im Notes-Dokument eine Liste, wenn ein Feld Mehrfachwerte zulässt. Nimmt ein Feld keine Mehrfachwerte auf, erscheint der Wert in Notes als einzelne Zeichenfolge. Daten, die über den Notes-Treiber abgefragt werden, ergeben in jedem Fall dasselbe Ergebnis.

Numerische Felder und Datumfelder

Der Notes-Treiber liefert nur den ersten Wert in der Liste, wenn die Felder mit Mehrfachwerten numerische Felder oder Datumfelder sind.

Der Notes-Treiber akzeptiert zur Einfügung in ein numerisches Feld für Mehrfachwerte nur einen einzigen numerischen Wert.

SQL-Anweisungen

Der Lotus-Notes-Treiber unterstützt einen Großteil der ODBC-Minimum- und Core-Level-SQL-Syntax. Einschränkungen und Ausnahmen sind bei jeder Anweisung beschrieben.

[Zusammenfassung der unterstützten ODBC-SQL-Grammatik](#)

[Ausnahmen von der ODBC-SQL-Grammatik](#)

Siehe auch

[Der Lotus-Notes-ODBC-Treiber](#)

Verbindungszeichenfolgen

Die folgenden Schlüsselwörter werden von einem SQLDriverConnect-Aufruf unterstützt:

Schlüsselwort	Beschreibung
DSN	Der Name der Datenquelle
Database	Der Name der Notes-Datenbank mit Pfadangabe (falls erforderlich)
Server	Der Name des Notes-Servers, auf dem sich die Datenbank befindet Lassen Sie das Feld leer, wenn sich die Datenbank auf der lokalen Arbeitsstation befindet.

Zum Beispiel verwenden Sie die folgende Verbindungszeichenfolge, wenn Sie eine Verbindung zur Datenquelle Personal im Verzeichnis C:\PERSONAL auf dem Server HR_1 herstellen möchten:

DSN=Personal; Database=c:\personal\personal.nsf; Server=HR_1

Datentypen

Die folgende Tabelle zeigt, wie die Lotus-Notes-Datentypen auf den ODBC-SQL-Datentypen abgebildet werden. Notes unterstützt zusätzlich zu den SQL-Datentypen noch zwei weitere Datentypen:

- Listenfelder
- Rich-Text-Felder

Abbildung der SQL_Datentypen auf den Notes-Datentypen

ODBC-SQL-Datentypen	<u>Lotus-Notes-Datentypen</u>
SQL_CHAR	Text
SQL_VARCHAR	Text
SQL_LONGVARCHAR	Notes Rich Text
SQL_DECIMAL	Zahl
SQL_NUMERIC	Zahl
SQL_SMALLINT	Zahl
SQL_INTEGER	Zahl
SQL_REAL	Zahl
SQL_FLOAT	Zahl
SQL_DOUBLE	Zahl
SQL_DATE	Zeit
SQL_TIME	Zeit
SQL_TIMESTAMP	Zeit

Hinweis SQLGetTypeInfo liefert ODBC-SQL-Datentypen zurück. Alle in Anhang D des *Microsoft ODBC SDK Programmer's Reference* beschriebenen Umwandlungen werden für die oben aufgeführten ODBC-SQL-Datentypen unterstützt.

Abbildung der Notes-Datentypen auf den SQL-Datentypen

Lotus-Notes-Datentyp	<u>ODBC-SQL-Datentyp</u>
Zahl	SQL_FLOAT
Zeit	Abhängig vom Format SQL_TIME, SQL_DATE oder SQL_TIMESTAMP
Text	SQL_VARCHAR
Schlüsselwort	SQL_VARCHAR
Mehrfachwerte-Liste	SQL_VARCHAR
Rich-Text-Feld	nur der Textteil wie bei SQL_LONGVARCHAR
Abschnitt	Nicht unterstützt

Fehlermeldungen

Fehlermeldungen können in einem von drei Layern entstehen:

- Der Treiber-Manager-Layer fängt jede nicht korrekte Folge von ODBC-API-Aufrufen und anderen unzulässigen Werten ab.
- Der Lotus-Notes-ODBC- und -SQL-Engine-Layer fängt jeden Fehler bei der Verarbeitung einer SQL-Anweisung ab.
- Der interne Lotus-Notes-Layer fängt Fehlermeldungen ab, die von einem Notes-API-Aufruf zurückgeliefert werden.

Fehlermeldungen haben das folgende Format:

[Vendor][ODBC-Komponente][Datenquelle]Meldungstext

Die Präfixe in Klammern ([]) weisen auf die Herkunft des Fehlers hin.

Die folgende Tabelle zeigt die Formate der Fehlermeldungen, die vom Treiber-Manager, dem Lotus-Notes-ODBC- und -SQL-Engine-Layer und dem internen Layer von Lotus Notes stammen:

Fehlermeldung	Fehlerherkunft
[Microsoft][ODBC DLL]Meldungstext	Treiber-Manager (ODBC.DLL)
[Casahl/Lotus][ODBC Lotus Notes]Meldungstext	Lotus-Notes-Treiber (NOTESSQL.DLL)
[Casahl/Lotus][ODBC Lotus Notes] [Lotus Notes Server]Notes API error:Meldungstext	Notes-API-DLLs

Fehler, die im Notes-Treiber oder dem Treiber-Manager auftreten

Der Treiber liefert eine Fehlermeldung mit dem SQL-Status (SQLSTATE) zurück, wenn der Fehler im ODBC-Lotus-Notes-Treiber oder dem Treiber-Manager auftritt. Wenn der Fehler im internen Layer von ODBC Lotus Notes auftritt, geben die Präfixe [Vendor] und [ODBC-Komponente] Vendor und Namen der ODBC-Komponente an, die den Fehler aus der Datenquelle erhalten hat.

Fehler, die in der Datenquelle auftreten

Bei Fehlern, die in der Datenquelle auftreten, liefert der Notes-Treiber eine Fehlermeldung mit SQLSTATE S1000, der durch den Notes-API-Aufruf zurückgeliefert wird. Tritt der Fehler im Treiber-Manager, dem Notes-Treiber oder dem SQL-Engine-Layer auf, wird die Datenquelle nicht angegeben.

ALTER TABLE

Die Anweisung ALTER TABLE fügt zu einer Tabelle eine oder mehrere Spalten hinzu.

Syntax

```
ALTER TABLE <Basis-Tabellenname>{ ADD <Spaltenname> <Datentyp>| ADD (<Spaltenname>  
<Datentyp> [ , <Spaltenname><Datentyp> ]... }
```

Siehe Konventionen

Beispiel

Argumente

<Basis-Tabellenname> ist die zu ändernde Tabelle.

<Spaltenname> ist der Name der Spalte, die hinzugefügt werden soll.

<Datentyp> ist einer der folgenden Datentypen:

- CHAR(n) oder CHARACTER(n) wobei $1 \leq n \leq 254$
- DECIMAL(p, s) wobei p (Stellenanzahl) $1 \leq p \leq 15$ und s (Dezimalzeichen) $0 \leq s \leq p$ ist.
- NUMERIC(p, s) (wie DECIMAL)
- SMALLINT
- INTEGER
- REAL
- FLOAT
- DOUBLE PRECISION
- VARCHAR(n) oder CHARACTER VARYING(n) (wie CHAR)
- DATE (Format JJJJ-MM-TT)
- TIME (Format HH-MM-SS)
- TIMESTAMP (Format JJJJ-MM-TT HH:MM:SS)

Hinweis Die verkürzten ODBC-Escape-Sequenzen {d 'jjjj-mm-tt'}, {t 'hh:mm:ss'} und {dt 'jjjj-mm-tt hh:mm:ss'} werden nicht unterstützt.

Hinweis Der Millisekunden-Teil des TIMESTAMP-Wertes wird nicht unterstützt.

Einschränkungen

- NULL oder NOT NULL wird nicht unterstützt.
- PRIMARY KEY wird nicht unterstützt.
- Die REFERENCES-Klausel wird nicht unterstützt.
- Tabelleneinschränkungsdefinitionen werden nicht unterstützt.

In der nachfolgenden Tabelle sind die zur Syntax-Beschreibung der SQL-Anweisungen verwendeten Konventionen aufgeführt.

<u>Konvention</u>	<u>Beschreibung</u>
<i>Argument</i>	Information, die von der Anwendung bereitgestellt werden muß
SQLTransact	Syntax, die genau wie dargestellt eingegeben werden muß; dazu gehören auch die Namen von Funktionen.
[]	Optionale Elemente oder Klammern (Darstellung in Fettschrift), die zur Syntax gehören und eingegeben werden müssen.
	Trennzeichen für zwei sich gegenseitig ausschließende Wahlmöglichkeiten in einer Syntax-Zeile
{ }	Begrenzung für zwei sich gegenseitig ausschließende Wahlmöglichkeiten in einer Syntax-Zeile
...	Argumente, die mehrfach wiederholt werden können
Benutzereingabe	In Beispielen weist diese Darstellung auf Informationen hin, die der Benutzer eingeben muß.

Ein Tabellenname kann bis zu 32 Zeichen lang sein und darf nicht mit dem Namen einer anderen Tabelle oder Ansicht in der Datenbank übereinstimmen. Schließen Sie den Tabellennamen in Anführungszeichen (" ") ein, kann er beliebige Zeichen einschließlich Leerzeichen enthalten. Ansonsten dürfen Tabellennamen nur aus Buchstaben, Ziffern, Unterstrichen (_) und Dollarzeichen (\$) bestehen und müssen mit einem Buchstaben beginnen. Verwenden Sie kein reserviertes SQL-Wort als Tabellenname.

Ein Spaltenname kann bis zu 32 Zeichen lang sein und darf nicht mit dem Namen einer anderen Spalte in der Tabelle übereinstimmen. Spaltennamen dürfen nur aus Buchstaben, Ziffern, Unterstrichen (_) und Dollarzeichen (\$) bestehen und müssen mit einem Buchstaben beginnen. Verwenden Sie kein reserviertes SQL-Wort als Spaltenname.

Ein Indexname kann bis zu 32 Zeichen lang sein und darf nicht mit dem Namen eines anderen Index, der von diesem Benutzer für die Tabelle erstellt wurde, übereinstimmen. Schließen Sie den Indexnamen in Anführungszeichen (" ") ein, kann er beliebige Zeichen einschließlich Leerzeichen enthalten. Ansonsten dürfen Indexnamen nur aus Buchstaben, Ziffern, Unterstrichen (_) und Dollarzeichen (\$) bestehen und müssen mit einem Buchstaben beginnen. Verwenden Sie kein reserviertes SQL-Wort als Indexname.

Ein Ansichtenname kann bis zu 32 Zeichen lang sein und darf nicht mit dem Namen einer anderen Tabelle oder Ansicht in der Datenbank übereinstimmen. Schließen Sie den Ansichtennamen in Anführungszeichen (" ") ein, kann er beliebige Zeichen einschließlich Leerzeichen enthalten. Ansonsten dürfen Ansichtennamen nur aus Buchstaben, Ziffern, Unterstrichen (_) und Dollarzeichen (\$) bestehen und müssen mit einem Buchstaben beginnen. Verwenden Sie kein reserviertes SQL-Wort als Ansichtenname.

<Ausdruck> ist eine beliebige Kombination aus Konstanten, Spaltenfunktionen und Spaltennamen.

<Vergleichsoperator> ist einer der folgenden Operatoren:

<u>Operator</u>	<u>Bedeutung</u>
<	Kleiner als
>	Größer als
<=	Kleiner als oder gleich
>=	Größer als oder gleich
=	Gleich
<>	Ungleich

<Suchmuster> ist entweder eine Zeichenfolgenkonstante (String-Konstante), das Schlüsselwort USER oder ein dynamischer Parameter. Zeichenfolgenkonstanten müssen in einfache (' ') oder doppelte (" ") Anführungszeichen eingeschlossen sein und können Jokerzeichen enthalten.

```
UPDATE LAGER SET MENGE = 100.00 WHERE CURRENT OF SQL_CUR_0
```

UPDATE LAGER SET STCKKOST=STCKKOST * 1.2

UPDATE PERSONAL SET PROVISION = (PROVISION * 1.25) WHERE EINSDAT < '1982-07-05'

```
SELECT LAGERORT, AVG(STCKKOST) FROM LAGER GROUP BY LAGERORT ORDER BY  
LAGERORT DESC
```

```
SELECT LAGERORT, AVG(STCKKOST) FROM LAGER GROUP BY LAGERORT HAVING  
AVG(STCKKOST) > 600
```

```
SELECT LAGERORT, AVG(STCKKOST) FROM LAGER GROUP BY LAGERORT
```

Select * FROM LAGER WHERE LAGERORT='New York' FOR UPDATE OF BESTAND


```
SELECT * FROM KUNDEN WHERE ORT='New York'
```

```
SELECT ARTIKEL_NR, BESCHR, BESTAND, LAGERORT, STCKKOST FROM LAGER WHERE  
BESTAND > 50
```

```
SELECT ARTIKEL_NR, BESCHR, LAGERORT, BESTAND, STCKKOST FROM LAGER WHERE  
LAGERORT = 'Los Angeles' OR LAGERORT = 'New York' AND BESTAND < 20 AND STCKKOST <  
1000
```

```
SELECT * FROM KUNDEN
```

```
INSERT INTO PERSONAL VALUES ('000001', 'Zambini', 'Rick', '1980-02-15', 'Los Angeles', '000000',  
6000, 5.0)
```

```
INSERT INTO NEUGEHTAB SELECT GEHALT, NACHNAME FROM PERSONAL,VERKAUF WHERE  
PERSONAL.PERS_NR = VERKAUF.PERS_NR AND GEHALT BETWEEN 5500 AND 6000
```

DROP VIEW NYKUND

DROP TABLE NEUKUND

DROP INDEX KUNDNDX

```
DELETE FROM LAGER WHERE CURRENT OF SQL_CUR_0
```

DELETE FROM NYKUND WHERE FIRMA='Interior Designs'


```
CREATE VIEW NYKUND (FIRMA,ANSCHRIFT) AS SELECT FIRMA,ANSCHRIFT FROM KUNDEN  
WHERE STAAT='NY'
```

```
CREATE TABLE PERSONAL  
  (PERS_NR CHAR(6),  
   NACHNAME CHAR(15),  
   VORNAME CHAR(10),  
   EINSDAT DATE,  
   ABTLG CHAR(15),  
   LEITER CHAR(6),  
   GEHALT FLOAT,  
   PROVISION FLOAT)
```

```
CREATE INDEX KUNDNDX ON KUNDEN (NACHNAME ASC)
```

```
ALTER TABLE KUNDEN ADD LAND VARCHAR(20)
```

