



2.3.1 Datenbanksysteme I

Lerninhalte 231-10 Abfragen erstellen (Teil 1)

Abfragen erstellen

Mit Hilfe von Abfragen kann man Daten in einer Datenbank anzeigen, hinzufügen, löschen oder ändern. Sie werden vom Benutzer im Datenbankmanagementsystem formuliert. Das Ergebnis einer Abfrage (*engl. Query*) ist eine Teilmenge des zugrundeliegenden Datenbestandes. Eine Abfrage kann die Informationen aus verschiedenen Tabellen abrufen und zusammensetzen, damit sie beispielsweise in einem Formular oder Bericht angezeigt werden können.

Zur Erstellung von Abfragen in relationalen Datenbanken hast du grundsätzlich zwei Möglichkeiten:

1. Die grafisch gestützte Entwurfsansicht (oder den Assistenten) des DBMS deiner Wahl
2. Die textuelle Datenbanksprache SQL (*engl. Structured Query Language*)

SQL – Structured Query Language

SQL steht für *Structured Query Language* und ist eine Datenbanksprache zur Definition, Abfrage und Manipulation von Daten in relationalen Datenbanken. SQL ist ein weltweiter Standard¹, um auf alle wichtigen Datenbanksysteme² zugreifen zu können.

Die Sprachregeln von SQL sind ziemlich einfach aufgebaut und mit Grundkenntnissen der englischen Sprache gut verständlich. Durch seine Rolle als Standardsprache ist SQL von großer Bedeutung, da damit eine weitgehende Unabhängigkeit von der benutzten Datenbanksoftware erzielt werden kann.

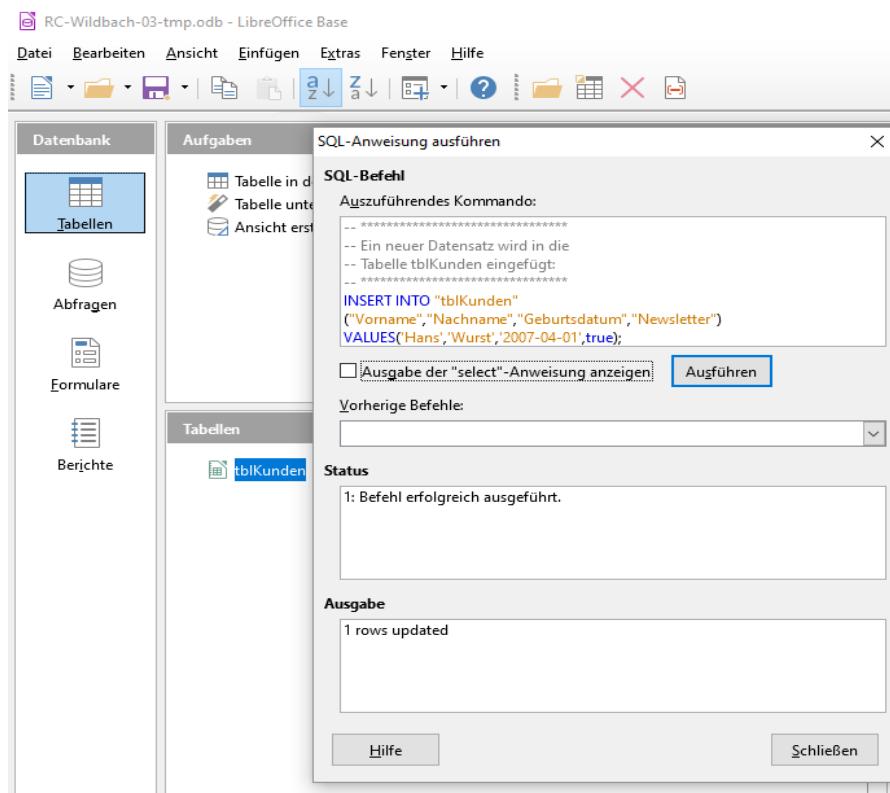


Abb. 01: SQL-Anweisung in LibreOffice Base ausführen (Menü Extras >> SQL)

¹ Die erste SQL-Standardversion wurde bereits 1987 durch die Internationale Organisation für Normung – kurz ISO (von gr.: „isos“; dt. „gleich“, engl. *International Organization for Standardization*) anerkannt.

² Das wären beispielsweise ORACLE, MySQL, MS Access, SQL-Server, PostGres oder HSQLDB.



2.3.1 Datenbanksysteme I

Lerninhalte 231-10 Abfragen erstellen (Teil 1)

SQL ist eine Programmiersprache und unterliegt somit bestimmten Satzbauregeln (Syntax), die ausführlich in den jeweiligen Standards und Handbüchern beschrieben werden. Nicht jeder Hersteller von Datenbankmanagementsystemen (DBMS) hat auf die gleiche SQL-Standardversion³ gebaut. Die in diesem Kapitel verwendeten einfachen Beispiele sind aber so gewählt, dass sie bei allen gängigen DBMS funktionieren.

SQL bietet einen kompletten Satz von Anweisungen, um mit relationalen Datenbanken arbeiten zu können. Man unterscheidet grundsätzlich **vier Aufgabenbereiche** oder Teilmengen von **SQL-Anweisungen**:

- **DCL** (engl. *data control language*): Sie wird verwendet, um Berechtigungen für ein Datenbanksystem zu vergeben oder zu entziehen (z. B. **GRANT, REVOKE**).
- **DDL** (engl. *data definition language*): Mit den Anweisungen der DDL werden Datenstrukturen (z. B. Tabellen) angelegt, geändert oder entfernt (z. B. **CREATE, ALTER, DROP**).
- **DML** (engl. *data manipulation language*): Mit den Befehlen der Datenbearbeitungssprache werden Daten geschrieben, gelesen, geändert und gelöscht (z. B. **INSERT, UPDATE, DELETE**).
- **DQL** (engl. *data query language*): Die Befehle der Datenabfragesprache umfassen alles, was man für eine Suchabfrage benötigt (z. B. **SELECT, FROM, WHERE, ORDER BY**).

Data Manipulation Language (DML; dt. Datenbearbeitungssprache)

Während die SQL-DCL und die SQL-DDL hauptsächlich von Datenbankadministratoren verwendet werden, sind die Anweisungen der SQL-DML dazu da, damit Daten geschrieben, geändert und gelöscht werden können.

Auf die Veränderung von Tabellendaten mithilfe von SQL-Anweisungen aus der DML soll an diesem Kapitel nur kurz eingegangen werden. Der gewöhnliche Datenbankbenutzer möchte ja hauptsächlich nützliche Informationen aus der Datenbank ziehen. Deshalb bildet das Auslesen von Informationen mit den SQL-Anweisungen der **DQL** einen Schwerpunkt in Teil 2 dieses Kapitels (*Lerninhalte 231-11 Abfragen erstellen - Teil 2*).

Einfügen eines Datensatzes mit INSERT

Neue Datensätze in eine Datenbank einzufügen, ist eine elementare Aufgabe jedes DBMS. SQL sieht dafür den Befehl **INSERT INTO** vor.

Das Beispiel findest du als Textdatei *insert-into-tblKunden.txt* im Ordner *231-materialien/SQL/DML*. Zeilen, die mit einem doppelten Mittestrich beginnen, werden vom DBMS ignoriert. So kann man erläuternde Kommentare beifügen.

```
-- ****
-- Ein neuer Datensatz wird in die
-- Tabelle tblKunden eingefügt:
-- ****

INSERT INTO "tblKunden"
("Vorname", "Nachname", "Geburtsdatum", "Newsletter")
VALUES('Hans', 'Wurst', '2007-04-01', true);4
```

³ Die meisten DBMS beruhen im Kern noch auf SQL92. Aktuelle, erweiterte Standards wären: SQL2016, SQLf und SQLSTATE.

⁴ Die syntaktisch vollständig korrekte Form der Datumsdarstellung lautet: {D 'YYYY-MM-TT'}, hier also {D '2007-04-01'}. In LibreOffice Base SQL funktionieren beide Varianten.



2.3.1 Datenbanksysteme I

Lerninhalte 231-10 Abfragen erstellen (Teil 1)

Aktualisieren von Datensätzen mit UPDATE

Vorhandene Datensätze müssen natürlich bei Bedarf wieder geändert werden können. Die Preise von Produkten ändern sich ständig nach der aktuellen Marktlage. Menschen ziehen um, heiraten und ändern ihre Namen. Es gibt viele Anlässe, Datensätze in einer Datenbank zu aktualisieren. Dafür bietet SQL die Anweisung **UPDATE**.

Das Beispiel findest du als Textdatei *update-tblKunden.txt* im Ordner *231-materialien/SQL/DML*.

```
-- ****
-- Aktualisiert den Datensatz von
-- von Hans Wurst (KdNr = 101).
-- ****
UPDATE "tblKunden"
SET "Anrede" = 'Herr',
    "Vorname" = 'Hansi',
    "Newsletter" = false
WHERE "KdNr" = 101;
```

Löschen von Datensätzen mit DELETE

Die dritte Möglichkeit, Tabellendaten zu verändern, besteht im Löschen von Datensätzen. Gelöscht werden dabei – wie beim Einfügen – grundsätzlich ganze Datensätze⁵. Dafür gibt es den SQL-Befehl **DELETE FROM**.

Das Beispiel findest du als Textdatei *delete-from-tblKunden.txt* im Ordner *231-materialien/SQL/DML*.

```
-- ****
-- Löscht alle Datensätze mit einer
-- KdNr größer als 99.
-- ****
DELETE FROM "tblKunden"
WHERE "KdNr" > 99;
```

Wie man die SQL-Anweisungen INSERT, UPDATE und DELETE in der Datenbank RC Wildbach ausführt, kannst du mit Hilfe dieses **Lernvideos** nachvollziehen:



Abb. 02: Startbild des Videotutoriums *SQL Teil 1*

⁵ Soll ein einzelnes Feld gelöscht werden, geht das so: **ALTER TABLE** Tabellenname **DROP** Feldname;