

Structured Query Language

SQL - Kommunikation mit relationalen Datenbanken

Grundlagen

Was zeichnet SQL aus?

- Standardisiert & Plattformübergreifend
- Im wesentlichen englische Sprachelemente
- Aufteilung in:
 - DDL
 - DML
 - DCL

Data Definition Language

DDL

- **CREATE TABLE ...** - für das Anlegen von Tabellen
- **DROP INDEX ...** für das Löschen eines Index
- **ALTER TABLE ...** für das Ändern von Tabellen

Data Manipulation Language

DML

- **INSERT INTO ...** für das Einfügen von Daten in Tabellen
- **SELECT FROM ...** für das Lesen von Daten aus Tabelle
- **UPDATE SET ...** für das Lesen von Daten aus Tabellen
- **DELETE FROM ...** für das Löschen von Daten in Tabellen

SQL - Statements

Read-Statement I

Beispiel:

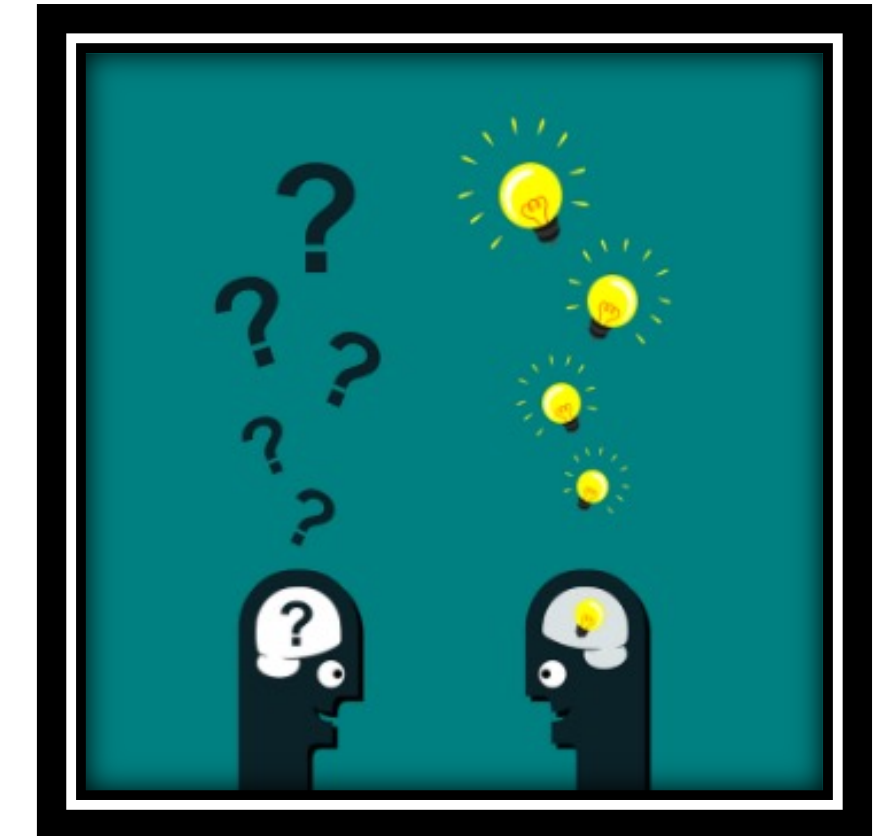
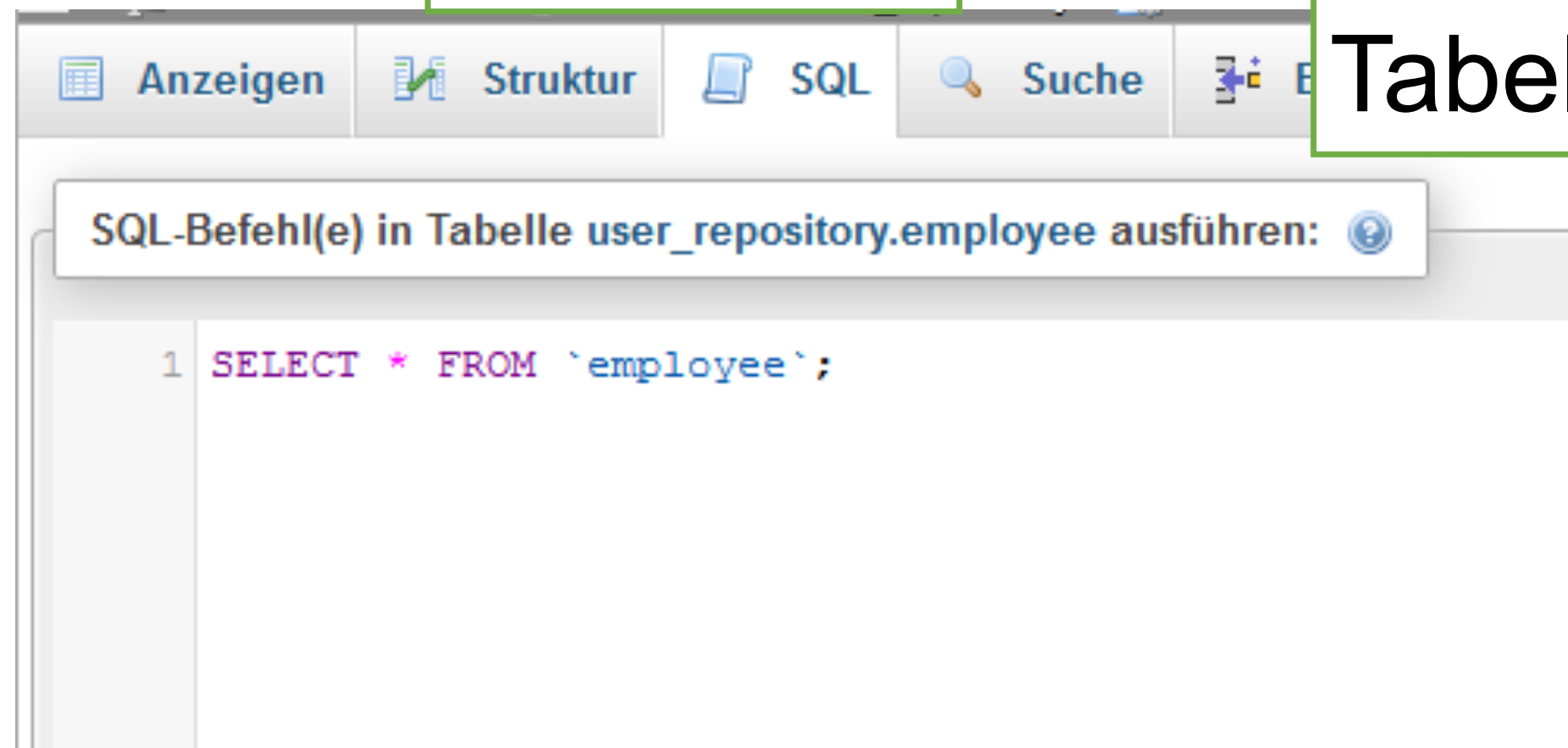
SELECT * **FROM** **EMPLOYEE**;

Daten aus
Tabelle
lesen

Alle
Spalte

Von
welcher
Tabelle

Tabellenname



SQL - Statements

Insert-Statement



```
INSERT INTO TABLE (SPALTE1, SPALTE2, SPALTE3) VALUES (1, „TEXT“, „TEXT“);
```

SQL-
Keyword
s für
Datensat
z
eintragen

Name
der
Tabelle,
in die
man
Daten
eintrage
n
möchte

In Klammern
die Spalten
eintragen, die
mit Werten
befüllt werden
sollen

SQL-
Keyword,
dass nun
die
Werte
folgen

Einzutra
gende
Werte

SQL - Statements

Update-Statement



UPDATE TABLE SET SPALTE1 = „Wert1“, SPALTE2 = „AndererWert“ WHERE SPALTE_ID = 1;

SQL-Keywords für
Veränderung von
einem oder
mehreren
Datensätzen

die Spalten
benennen und eine
Wertzuweisung
vornehmen

SQL-
Keyword für
die
Eingrenzung

Eingrenzungskriterium

SQL - Statements

Delete-Statement

DELETE FROM

SQL-Keyword für das Löschen eines oder mehrerer Datensätze

TABLE

Ausgewählte Tabelle

WHERE

SQL-Keyword für die Eingrenzung

ID = 1;

Eingrenzungskriterium



Data Control Language

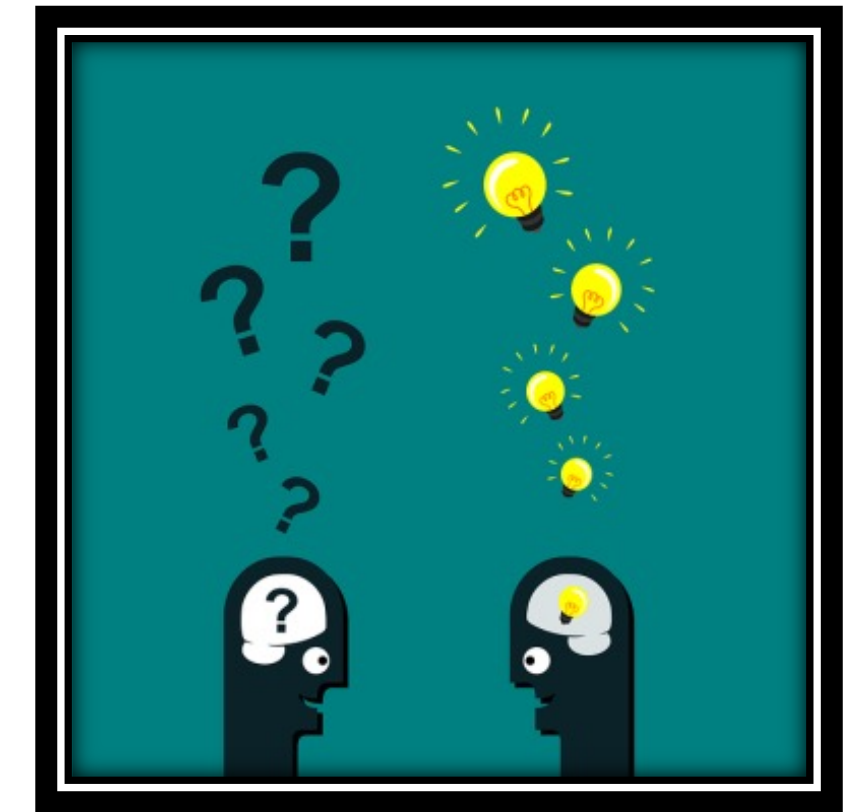
DCL

- **GRANT ON TO** ... für die Weitergabe von Rechten an einer Tabelle zu einem bestimmten User
- **REVOKE** ... für das Widerrufen von Rechten

SQL – Statements II

Create-Statement II - Tabelle anlegen I

```
CREATE TABLE IF NOT EXISTS CUSTOMER
(
  ID int(11) NOT NULL AUTO_INCREMENT,
  FORENAME varchar (100),
  SURNAME varchar(100) NOT NULL,
  TITLE varchar(100),
  ADDRESS varchar(250),
  POSTALCODE int(5),
  CITY varchar(100),
  ACTIVE BOOLEAN NOT NULL,
  PRIMARY KEY (ID)
);
```

A screenshot of a database management tool interface. At the top, there is a toolbar with icons and labels for 'Struktur', 'SQL', 'Suche', 'Abfrage', and 'Exportieren'. Below the toolbar, a text box says 'SQL-Befehl(e) in Datenbank company_relations ausführen:'. The main area displays an SQL command with line numbers 1 through 12, matching the code in the previous block. The command is: '1 CREATE TABLE IF NOT EXISTS CUSTOMER', '2 (', '3 ID int(11) NOT NULL AUTO_INCREMENT,', '4 FORENAME varchar (100),', '5 SURNAME varchar(100) NOT NULL,', '6 TITLE varchar(100),', '7 ADDRESS varchar(250),', '8 POSTALCODE int(5),', '9 CITY varchar(100),', '10 ACTIVE BOOLEAN NOT NULL,', '11 PRIMARY KEY (ID)', '12);'.

SQL-Tutorial