

Einführung in die Wirtschaftsinformatik

Teil 06 – SQL Einführung und Datendefinition

Wintersemester 2026/2027

Lehrstuhl für Wirtschaftsinformatik
Prozesse und Systeme
Universität Potsdam

Chair of Business Informatics
Processes and Systems
University of Potsdam

Univ.-Prof. Dr.–Ing. habil. Norbert Gronau
Lehrstuhlinhaber | *Chairholder*

Mail August-Bebel-Str. 89 | 14482 Potsdam | Germany
Visitors Digitalvilla am Hedy-Lamarr-Platz, 14482 Potsdam
Tel +49 331 977 3322

E-Mail ngronau@lswi.de
Web lswi.de

Lernziele

- Was unterscheidet DDL, DML, DCL und welche Aufgaben erfüllen sie?
- Wie werden Tabellen in SQL erzeugt, verändert und gelöscht (CREATE, ALTER, DROP)?
- Welche Datentypen und Wertebereiche stehen in SQL zur Verfügung und wie werden sie eingesetzt?
- Wie werden Datensätze eingefügt, geändert und gelöscht (INSERT, UPDATE, DELETE)?
- Welche Regeln gelten für Tabellen, Spalten, Schlüsselattribute und Schemata?
- Wie funktioniert die Vergabe von Benutzerrechten in SQL (GRANT, REVOKE)?



Abfragesprachen

Die Datenbanksprache SQL

Tabellen verwalten

Datensätze verwalten

Datenkontrolle und -steuerung

Hörsaal-Quiz - Einleitungsfragen

Öffnet die App über den QR-Code oder den Link:

https://quiz.lswi.de/

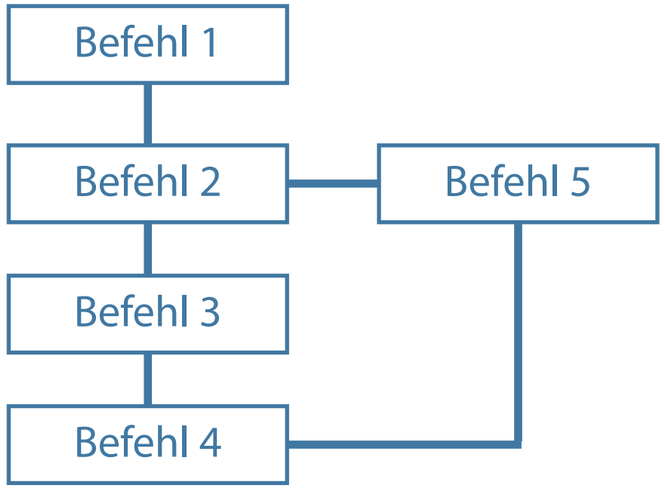
pwd: ewinf

Sprachen und Zugriffsmethoden

Geeignete Sprachmittel für Datenbankzugriffe

- Sprache für Spezifikation von Datentypen und physischen Datenstrukturen (Data Definition Language - DDL)
- Sprache für Formulierung von Anfragen und Änderungsoperationen (Query Languages)
- Kontrolle der Sicherheit und der Zugriffsrechte für Objekte oder Teile eines Datenbanksystems (Data Control Language - DCL)
- Sprache zur Modifikation von Datensätzen bereits bestehender Tabellen (Data Manipulation Language - DML)

Prozedurale Programmiersprachen




```
i = 1
while i <= 42 :
   if i < 42:
      i = i+1
   elif i = 42:
      print  ”Die Antwort ist: “, i
```

Prozedurale Sprachen

- Problemorientiert
- Imperatives Prinzip - Programm als Folge von Befehlen, Ausführung im Rechner in vorgegebener Reihenfolge
- Explizite Steuerung ⟷ Implizite Logik

Prozedurale Programmierung

- Aufbau von Computerprogrammen aus kleineren Teilproblemen (Aufgaben, Prozeduren)
- Kleinster und unteilbarer Schritt bei diesem Verfahren → die Anweisung

Jeder Programmalgorithmus besteht aus Logik und Steuerung.

Deklarative Sprachen

„Gib mir die Namen und Positionen aller Mitarbeiter aus Abteilung 260F“

```
SELECT vorname, name, position
FROM mitarbeiter
WHERE abt_nr = '260F';
```

Deklarative Sprachen

- Vereinfachung des Programmierprozesses durch natürliche Sprache
- Beschreibend (deklarativ) - beschreibt gewünschtes Ergebnis, nicht aber die Darstellung
- Explizite Logik ⟷ Implizite Steuerung

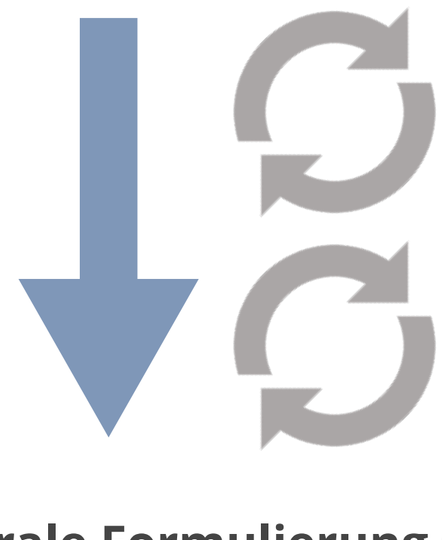
Deklarative Programmierung

- Beschreibung über Anfragen, welche Informationen gesucht werden
- Eine Anweisung beinhaltet eine Anfrage

Diese Sprachen werden als Sprachen der vierten Generation („4GL“) bezeichnet.

Unterschiede bei der Abbildung einer Aufgabenstellung

Aufgabenstellung: Gib eine Liste aller Männer aus.



Prozedurale Formulierung (3. Generation)

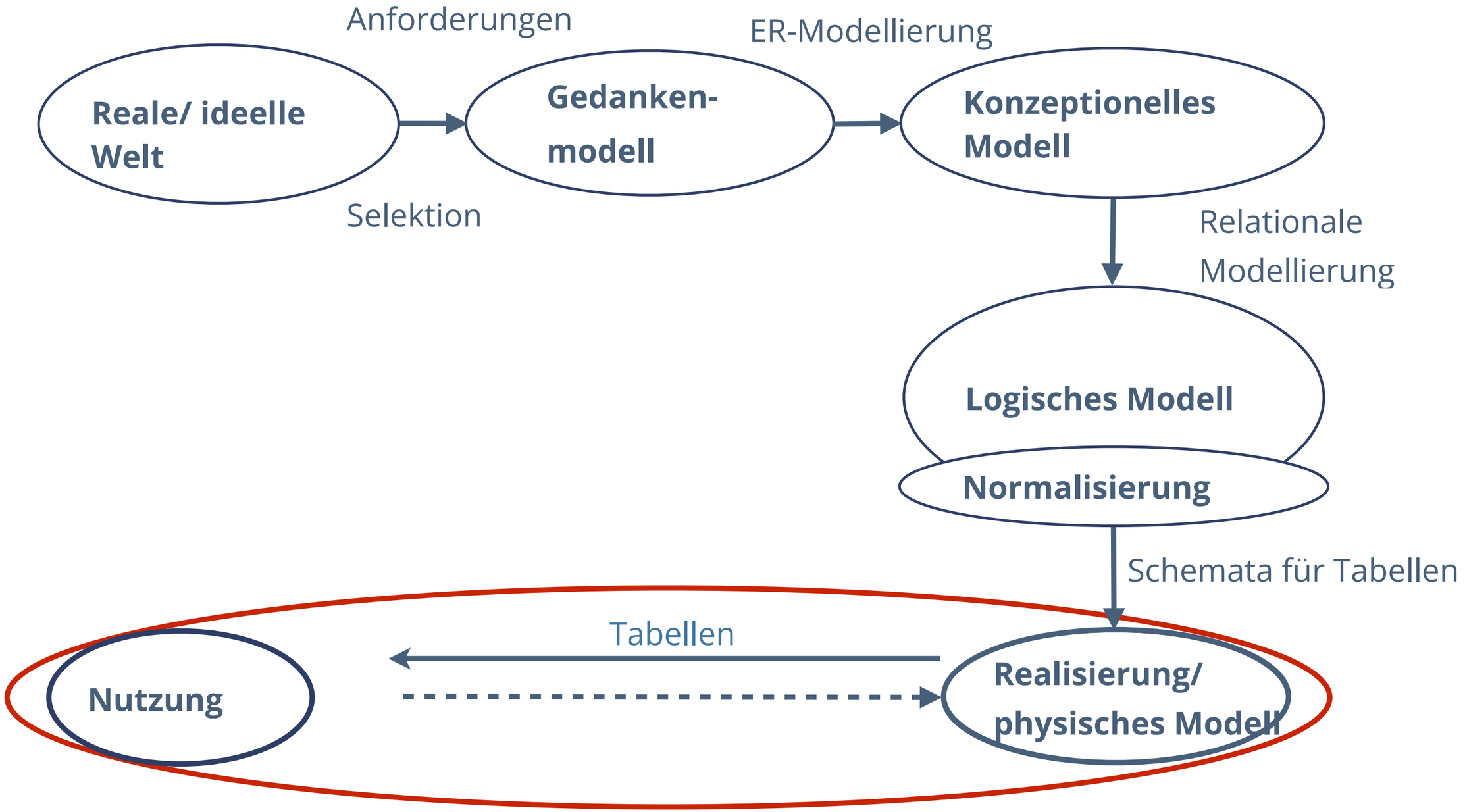
(1) Gehe zum ersten Mitarbeiter in der Tabelle
(2) Nimm Mitarbeiter aus Tabelle Personal
(3) Prüfe, ob männlich
(4) Falls JA, notiere Mitarbeiter
(5) Prüfe, ob letzter Mitarbeiter
(6) Falls NEIN, gehe zum nächsten Mitarbeiter und zu (3)
(7) Falls JA, ENDE

Deklarative Formulierung (4. Generation, SQL)

- Wähle Spalte aus, wähle Tabelle aus, lege Bedingung(en) fest

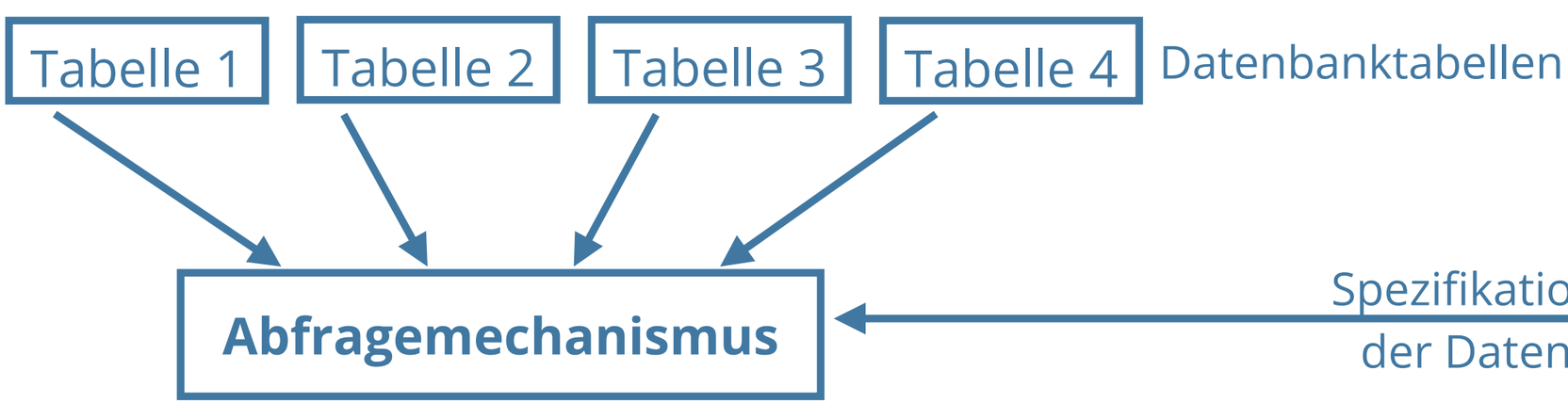
- Beispiel:
  SELECT Name FROM Mitarbeiter
  WHERE Geschlecht = „männlich"

Der Weg zur Datenbank (Abfragen)



Abfrageprinzip

- Abfragen liefern die gewünschten Daten in Tabellenstruktur
- Ausgabe - Kombination von Datensätzen und Spalten



Abfragen liefern eine Ausgabetabelle

Benutzer

Name	Vorname	PLZ
Müller	Anna	26121
Huber	Karl	46047
Müller	Amelie	21614
Hanser	Kurt	14482



Abfragesprachen

Die Datenbanksprache SQL

Tabellen verwalten

Datensätze verwalten

Datenkontrolle und -steuerung

Grundlegende Merkmale von SQL

- Befehle zur Definition, Manipulation und Abfrage von Daten
- Ermöglicht Benutzer, Online-Abfragen über ein Programm direkt am Bildschirm einzugeben
- Abfrageergebnis liegt sofort vor
- „Konstruktionsteile" - festgelegt durch Syntaxregeln, formuliert als Anfragen (Abfragen)

Name	Vornam	PLZ	...	...
Mülle	Anna	26121		
Huber	Karl	46047		
Mülle	Amelie	21614		
Hanse	Kurt	14482		



PLZ	Ort
26121	Oldenburg
46047	Oberhausen
21614	Buxtehude
14482	Potsdam

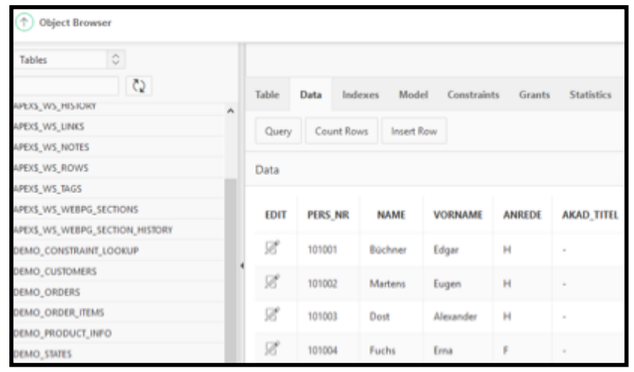


Tabelle ⟷ Anfragen

Sichten

(Tabelle + Formular)

SQL ist die Standardsprache relationaler Datenbanksysteme.

Sprachumfang von SQL (1 von 2)

Definition des Datenbankschemas (Data Definition Language)

- Erzeugen, Ändern, Löschen von Tabellen
- Syntax:
  `CREATE TABLE Tabellenname`
- Beispiel:

```
CREATE TABLE Mitarbeiter (…)
```

Befehle zur Zugriffssteuerung (Data Control Language)

- Erzeugen, Ändern, Löschen von Benutzerrechten
- Syntax:
  `GRANT <Funktion(en)> ON <Tabelle> TO <Nutzer>`
- Beispiel:

```
GRANT SELECT ON schemaname.Projekt TO
USER1
```

Sprachumfang von SQL (2 von 2)

Befehle zur Datenabfrage (Data Manipulation Language)

- Auswahl Datensätze und Spalten, Verknüpfung von Tabellen
- Syntax:
  `SELECT <Spalte(n)> FROM <Tabelle(n)>`
- Beispiel:

```
SELECT Vorname, Name FROM Mitarbeiter
```

Befehle zur Datenmanipulation (Data Manipulation Language)

- Einfügen, Ändern, Löschen in Tabellen
- Syntax:
  `INSERT INTO <Tabelle> <Merkmale>`
  `VALUES <Werte>`
- Unvollständiges Beispiel:

```
INSERT INTO Mitarbeiter (Pers_Nr,
Name, Vorname, Geschlecht)
VALUES('102400', 'Mustermann',
'Max','M')
```

Organisation der Abbildung von Werten in Tabellen

Vorname	h/Woche	Geburtsdatum	…
Martin	24	01.09.89	…
Anne	21	23.04.93	…
Grit	35	17.12.76	…
Text	*Zahl*	*Datum*	

Vorname	H/Woche	Geburtstag	…
Martin	24	01.09.89	…
Anne	21	23.04.93	…
Grit	35	17.12.76	…

Beispiel: Ganzzahlige Werte 0…50

Formate und Werte

- Attribute - Spaltenorientierung
- Attributausprägung - Konkrete Werte von Attributen einzelner Entities
- Domäne - Wertebereiche von Attributen

Wertebereiche

- Wertevorrat durch Wertebereich festgelegt
- Wertebereich - numerischer, alpha-numerischer, Datums-/Uhrzeit- oder logischer (wahr, falsch) Datentyp

Jedes einfache Attribut eines Entitytyps ist mit einer Wertemenge verknüpft.

Quelle: Elmasri, 2002, S. 72

Datentypen

Numerische Werte

- Ganze Zahlen (in dezimalem oder hexadezimalem Format)
- Fließkommazahlen (Ziffernfolge - Dezimalpunkt - Ziffernfolge)
- Wissenschaftliche Notation (Darstellung von Exponentialzahlen)

Zeichenkettenwerte

- Alphanumerische Zeichen (in Hochkommata bzw. Anführungszeichen gesetzt)
- Sonderzeichen (nicht druckbare Zeichen, Steuerzeichen)

Datums- und Zeitwerte

- Datum (z. B. im Format 'Jahr-Monat-Tag')
- Uhrzeit (Stunden-Minuten-Sekunden)

NULL-Wert

- Typloser Wert ('kein Wert' oder 'unbekannter Wert' in einem Feld)

Quelle: Elmasri, 2002, S. 227f

Attributtypen

Numerische Werte

- Ganze Zahlen - Number (Oracle), INTEGER, SMALLINT (MySQL)
- Reelle Zahlen - FLOAT, REAL

Zeichenketten (Strings), alphanumerische Werte

- CHAR(Länge)
- VARCHAR2(Länge) - Strings mit max. Zeichenlänge (Oracle)

Datums- und Uhrzeitwerte

- DATE: Datentyp für Datumsanzeige „04.03.2020“
- TIME: spezieller Datentyp zur Anzeige der Zeit
- Achtung: Deutsches Datumsformat muss explizit gesetzt werden

`NUMBER(7,2)`
Festkommazahl
7 Stellen inklusive 2 Nachkommastellen

`VARCHAR2(7)`
beliebige alphanumerische Zeichen
Max. Länge 7

`DATE`
Datumswert

Die vorherrschenden Datentypen in Attributen sind numerische und Zeichenkettenwerte.

Attributtypen - Optionen und Sondertypen

Zusätzlich zum Datentyp sind als Optionen möglich:

- 'Not NULL' - das Feld darf nicht leer bleiben
- 'Default value' - wird als Vorgabe genommen
- 'Auto_increment' - für Primärschlüsselfelder, Wert wird mit neuem Datensatz automatisch erhöht
- Primary key - Primärschlüssel (zwingend für jedes Tupel erforderlich)

Große Datenobjekte - Grafiken, Textmengen

- Große, unstrukturierte Objekte (BLOB - Binary Large Objects)
- Große Zeichendatentypen (CLOB - Character Large Objects)

Datentyp - Logischer Wert

- BOOLEAN: kann nur Werte true oder false annehmen

Attribute des Entitytyps Artikel

Zulässige Datentypen in Oracle

Datentyp	Beschreibung
VARCHAR2 (wert)	Zeichendaten variabler Länge
CHAR (wert)	Zeichendaten fester Länge
NUMBER (p,s)	Numerische Daten variabler Länge (p: Länge insges., s: davon
DATE	Datums- und Zeitwerte
CLOB	Zeichendaten bis 4 GB

Beispiel-Schema einer Artikelstammdatei

Spaltennam	Artikelnum	Artikelnam	Hersteller	Artikelgrup	Bestand	Nettopreis
Datentyp	VARCHAR2 (6)	VARCHAR2	VARCHAR2	VARCHAR2 (8)	NUMBER	NUMBER
Auf NULL	No	No	Yes	Yes	No	Yes
Primärschlüsse	1	-	-	-	-	-
	Feld 1	Feld 2	Feld 3	Feld 4	Feld 5	Feld 6

SQL als deklarative Abfragesprache

Festlegung der Bedingungen für die gesuchten Daten durch Nutzer

SELECT attribut	FROM tabelle	WHERE attribut = 'Uta Herbst'
Welches Merkmal soll ausgewählt werden?	Aus welcher Tabelle stammt das Merkmal?	Nach welchen Auswahlbedingungen soll gesucht werden?

- Häufig leichtere Formulierungen → Unkomplizierte und effiziente Auswertung
- Viel kürzer als prozedurale Programmierung → Preiswertere Programmentwicklung
- Schnittstellen zu höheren Programmiersprachen vorhanden



Abfragesprachen

Die Datenbanksprache SQL

Tabellen verwalten

Datensätze verwalten

Datenkontrolle und -steuerung

Regeln für die Tabellenerstellung

Namensregeln für Tabellen und Spalten

- Zwingender Beginn mit einem Buchstaben
- Länge mindestens 1 und maximal 30 Zeichen
- Zulässige Zeichen: A-Z, a-z, 0-9, _, $, #
- Grundsätzlich unterschiedliche Namen der Objekte (z.B. Tabellen) eines Benutzers
- Nicht zulässige Namen - für Oracle reservierte Worte

Bedeutung von Schemata

- Beim erstmaligen Erstellen von Objekten in der Datenbank wird gleichzeitig ein Schema angelegt
- Ein Schema ist eine Sammlung von Datenbank-Objekten, die einem bestimmten Nutzer gehören
- Aufruf eines Objektes erfolgt intern immer durch Schema- und Objektnamen.

```
CREATE TABLE [schema.]tabelle
(spalte datentyp [DEFAULT standartwert1] [, ...]);
```

Quelle: Greenberg 2002, S. 372

Hörsaal-Quiz - Recap erste Hälfte

Öffnet die App über den QR-Code oder den Link:

https://quiz.lswi.de/

pwd: ewinf

Tabellenverwaltung mit CREATE

CREATE TABLE - Erstellen einer Tabelle

- Voraussetzung für Benutzer - Besitz des CREATE TABLE-Privilegs

```
CREATE TABLE tabelle (spalte datentyp [NOT NULL], ...,
PRIMARY KEY (spalte, ...),
FOREIGN KEY (spalte, ...)
                    REFERENCES referenztabelle (spalte));
```

Schlüsselwörter

- CREATE TABLE - Tabellenname, Spaltenbezeichner
- NOT NULL - Feld darf nicht leer sein (optional)
- PRIMARY KEY - Schlüsselattribut
- FOREIGN KEY - Verknüpfungen zu anderen Tabellen der Datenbank
- REFERENCES - Verweis, Attribut ist Primärschlüssel in Fremdtabelle

Alle Attributnamen innerhalb einer Tabelle müssen eindeutig sein.

Erzeugung einer neuen Tabelle

Tabelle mit CREATE erstellen

```
CREATE TABLE projektpartner (projekt_nr VARCHAR2(4) PRIMARY
KEY NOT NULL, projekt_partner VARCHAR2(35) NOT NULL, strasse
VARCHAR2(40), plz VARCHAR2(6), stadt VARCHAR2(30), land
VARCHAR2(20));
```

```
Table created.
```

Anschauen des Ergebnisses

```
DESCRIBE projektpartner;
```

Name	Null?	Type
Projekt_Nr	NO	VARCHAR2(4)
Projekt_Partner	NO	VARCHAR2(35)
Strasse		VARCHAR2(40)
PLZ		VARCHAR2(6)
Stadt		VARCHAR2(30)
Land		VARCHAR2(20)

Tabellenverwaltung mit ALTER

Pseudocode

Nachträgliches Ändern einer Tabellenstruktur

```
ALTER TABLE tabelle
  ADD (spalte datentyp DEFAULT ausdruck),...
MODIFY (spalte datentyp DEFAULT ausdruck),...
                DROP (spalte, …)
COLUMN ({spalte datentyp [NOT NULL]} , ...);
```

Mögliche Aktionen

- Hinzufügen neuer Spalten - ADD
  - Ändern vorhandener Spalten - MODIFY
  - Definition von Default-Werten für einzelne Spalten
- Löschen von Spalten - DROP
- COLUMN - ändert Datentyp Spalte

Quelle: Greenberg 2002, S. 391

Ändern von Spalten

Änderung von Datentyp, Größe und Default-Werte einer Spalte

Tatsächlicher SQL-Code

- Auswirkung des Default-Wertes nur auf nachfolgende Einfügungen

```
ALTER TABLE projektpartner
MODIFY (projekt_partner VARCHAR2(50));
```

```
Table altered.
```

- Struktur der erstellten Tabelle prüfen (mit DESCRIBE)

Name	Null?	Type
Projekt_Nr	NO	VARCHAR2(4)
Projekt_Partner	NO	VARCHAR2(50)
Strasse		VARCHAR2(40)
PLZ		VARCHAR2(6)
Stadt		VARCHAR2(30)
Land		VARCHAR2(20)

Quelle: Greenberg 2002, S. 394

Hinzufügen von Spalten

Hinzufügen mit ADD - neue letzte Spalte

```
ALTER TABLE abteilung
ADD (personalbudget NUMBER(8));
```

```
Table altered.
```

ABT_NR	ABT_NAME	...	PERSONALBUDGET
100V	Vorstandsstab	...	
105C	Controlling	...	
106Q	Qualitätssicherung	...	
107R	Interne Revision	...	
210E	Einkauf	...	
220L	Logistik	...	
...	...	...	

ADD fügt der bestehenden Tabelle eine neue Spalte hinzu.

Löschen von Spalten

Löschen nicht mehr benötigter Spalten

```
ALTER TABLE abteilung
DROP (personalbudget);
```

```
Table altered.
```

- Löschung nur für jeweils eine Spalte
- Anzeige der Änderung mit SELECT

ABT_NR	ABT_NAME	...
100V	Vorstandsstab	...
105C	Controlling	...
106Q	Qualitätssicherung	...
107R	Interne Revision	...
210E	Einkauf	...
220L	Logistik	...
...	...	...

Tabellenverwaltung mit DROP

Löschen einer Tabelle

- Ergebnis - Tabelle, Struktur und alle in ihr enthaltenen Datensätze sind danach gelöscht
- Vorgang ist NICHT umkehrbar:

```
DROP TABLE tabelle;
```

- Beispiel:

```
DROP TABLE projektpartner;
```

Achtung: Mit Ausführen des Befehls erfolgt eine unwiderrufliche Löschung (kein zusätzliches Speichern notwendig).



Abfragesprachen

Die Datenbanksprache SQL

Tabellen verwalten

Datensätze verwalten

Datenkontrolle und -steuerung

Einfügen von Datensätzen mit INSERT

Voraussetzung - bereits bestehende Tabelle

```
INSERT INTO tabelle [(spalte [, spalte...])]
        VALUES (wert [, wert...]);
```

Forderung - Verträglichkeit der Werte mit dem Datentyp des Feldes

- VALUES - Einfügen der Werte in Reihenfolge der Attribute
- CHAR- und DATE-Werte in Hochkommata

```
INSERT INTO kunde (kd_nr, bezeichnung, kd_typ_code, land,
                   plz_ort, strasse_nr)
VALUES ('10-0132', 'Saegewerk Plank', '500', 'Deutschland',
          '39340 Haldensleben', 'Gerikestr. 95');
```

```
INSERT INTO abteilung VALUES ('260Y', 'Teilelager 2', 'PO',
                              'Meisterbereich');
```

Quelle: Greenberg 2002, S. 321

Erzeugen eines neuen Datensatzes

```
INSERT INTO abteilung (abt_nr, abt_name, betr_teil)
       VALUES ('108M', 'Marketing', 'V');
```

```
1 row updated.
```

- Neue Zeile mit Werten für jede Spalte
- Angabe der Werte in der Default-Spaltenreihenfolge

ABT_NR	ABT_NAME	BETR_TEIL
100V	Vorstandsstab	ZK
105C	Controlling	ZK
106Q	Qualitätssicherung	PB
...	...	...
630E	Technik Endgeräte	PB
640S	Technik Service, Support	PB
108M	Marketing	V

Quelle: Greenberg 2002, S. 322

Ändern von Datensätzen mit UPDATE

Anweisung UPDATE - Änderung des Inhalts einer Spalte

- Möglichkeit der gleichzeitigen Änderung einer oder mehrerer Spalten

```
UPDATE tabelle
SET spalte = wert [, spalte = wert, ...]
[WHERE bedingung];
```

- WHERE-Klausel - Auswahl bestimmter Datensätze zum Ändern
- Ohne Einschränkung - Aktualisierung aller Datensätze

Quelle: Greenberg 2002, S. 329

Aktualisierung von Datensätzen

WHERE-Klausel - Einschränkung auf einzelne Zeile oder Gruppe von Zeilen

```
ALTER TABLE mitarbeiter
MODIFY (geschlecht VARCHAR2(4));
```

Änderung des Datentyps der Spalte Anrede

```
UPDATE mitarbeiter
SET geschlecht = 'männlich'
  WHERE geschlecht = 'M';
```

```
86 rows updated.
```

```
SELECT pers_nr,
geschlecht, name, position
  FROM mitarbeiter;
```

Anzeige der Änderung mit SELECT

PERS_NR	ANREDE	NAME	POSITION
101001	Herr	Schulze	Assistent
101002	Herr	Lange	Geschäftsführer
101003	Frau	Metz	Sekretärin
...	...	...	...
101049	Herr	Junge	Auszubildender
101050	Herr	Altermann	Praktikant

Die Auswahl der Zeilen kann über AND- bzw. OR-Operatoren eingeschränkt bzw. erweitert werden.

Löschen von Datensätzen mit DELETE

Anweisung DELETE - Löschung von Zeilen

- Standardmäßig Löschung aller Datensätze einer Tabelle
- Aber: Keine Löschung der Tabelle und ihrer Struktur

```
DELETE [FROM] tabelle;
```

- Über die Formulierung einer Bedingung (sog. WHERE-Klausel) kann die Auswahl von Datensätzen eingeschränkt werden
—> Löschung einzelner Zeilen - Bedingung: konkreter Wert aus Primärschlüssel

```
DELETE [FROM] tabelle
  [WHERE bedingung];
```

Das Löschen von einzelnen oder mehreren Datensätzen muss mittels Angabe einer WHERE-Klausel erfolgen!

Quelle: Greenberg 2002, S. 336

Löschung einzelner Datensätze

WHERE-Klausel - Abfrage der Werte aus Primärschlüssel

```
DELETE FROM abteilung
WHERE abt_name = 'Marketing';
```

```
1 row deleted.
```

Hinweis: Löschung erfolgt unter Beachtung der Datenintegrität

```
SELECT *
FROM abteilung;
```

Anzeige der Änderung mit SELECT

Gelöschte Datensätze werden nicht mehr angezeigt

ABT_NR	ABT_NAME	...
100V	Vorstandsstab	...
106Q	Qualitätssicherung	...
...	...	...
630E	Technik Endgeräte	...
640S	Technik Service, Support	...

Zeilen mit einem Primärschlüssel, der in einer anderen Tabelle als Fremdschlüssel dient, können nicht gelöscht werden.



Abfragesprachen

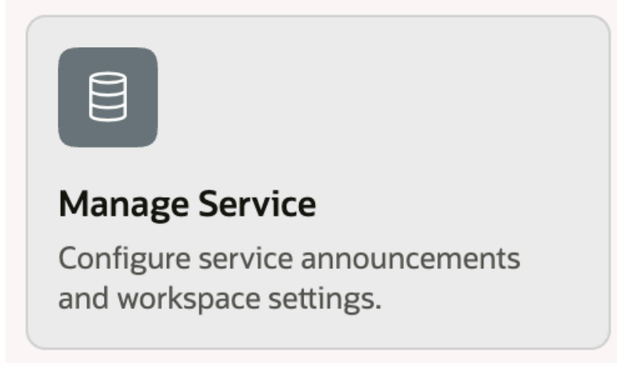
Die Datenbanksprache SQL

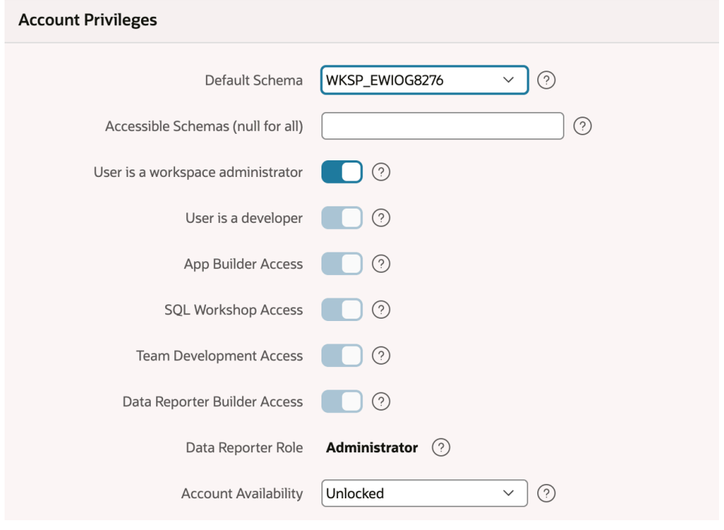
Tabellen verwalten

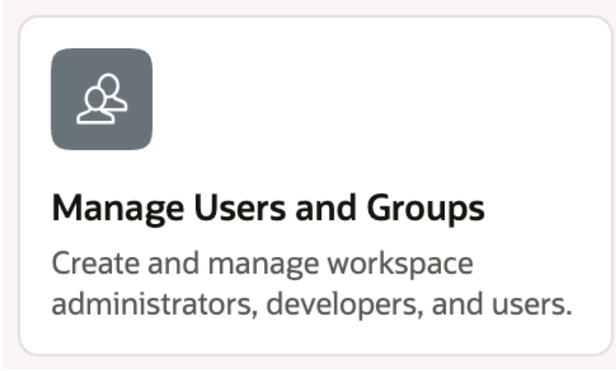
Datensätze verwalten

Datenkontrolle und -steuerung

Steuerung der Benutzerrechte







Datenbanksicherheit

- Systemsicherheit - Zugriff auf System (Organisation)
- Datensicherheit - Zugriff auf Datenbankobjekte

Privilegien

- Zugriff auf Datenbank - Systemprivilegien
- Zugriff auf Datenbankobjekte und Bearbeitung - Objektprivilegien

Datenbanknutzer

- Systemadministrator
- (Einfacher) Benutzer

Vergabe von Benutzerrechten (Privilegien)

Teilmenge von Privilegien, die vergeben werden können

- Hinzufügen
- Ändern
- Löschen

Voraussetzung für Benutzer - Objektprivilegien

- Berechtigung zur Durchführung bestimmter Aktionen für ausgewählte Tabellen
- Eigentümer hat alle Privilegien für ein Objekt (z.B. Tabelle)
- Möglichkeit der Rechtevergabe an andere Nutzer

Datenbank - Kontrolle der Zugriffsrechte

Datenkontrollsprache (Data Control Language)

- Kontrolle der Sicherheit und der Zugriffsrechte für Objekte oder Teile eines Datenbanksystems
- Befehle liegen näher bei der Sprache des DBMS

Typische SQL-Befehle der DCL

- GRANT – vergibt Zugriffsrechte
- DENY – verweigert Zugriffsrechte
- REVOKE – löscht vorher vergebene oder verweigerte Zugriffsrechte

Zusammenfassung - Übersicht der SQL-Anweisungen

Datenmanipulationssprache (Data Manipulation Language)

- INSERT - Einfügen
- UPDATE - Ändern
- DELETE - Löschen

Datendefinitionssprache (Data Definition Language)

- CREATE - Erzeugen
- ALTER - Ändern
- DROP - Löschen
- RENAME - Umbenennen

```
INSERT INTO artikel
(artikel_nr, bezeichnung, net_preis, mwst_red)
VALUES ('297', 'Trennsaege Stenner MHS 12', '46210', '0')
```

```
Hier ein Beispiel zu DELETE
```

```
DELETE FROM artikel
WHERE artikel_nr = '297';
```

Hörsaal-Quiz - Wissensvertiefung

Öffnet die App über den QR-Code oder den Link:

https://quiz.lswi.de/

pwd: ewinf

Kontrollfragen

- Welche Aufgaben können mit DDL-Operationen realisiert werden?
- Kann über DML-Sprachelemente ein neuer Datensatz erzeugt werden?
- Können neue Datensätze auch eingefügt werden, obwohl nicht für alle Spalten Daten vorhanden sind und eingetragen werden können?
- Wo finden Datumsfunktionen ihre Anwendung?

Literatur

Kemper, A./Eickler, A.: Datenbanksysteme; 6. Auflage, 2006, Oldenbourg Verlag

Greenberg, N./Nathan, P: Professioneller Einstieg in Oracle9i SQL - Band 1; 2002, Oracle

Elmazri, R./Navathe, S. B.: Grundlagen von Datenbanksystemen; 3. Auflage, 2002, Addison-Wesley

Zum Nachlesen



Gronau, N., Gäbler, A.:
Einführung in die Wirtschaftsinformatik, Band 1
8. überarbeitete Auflage

GITO Verlag Berlin 2019, ISBN 978-3-95545-233-9

Kontakt

Univ.-Prof. Dr.-Ing. Norbert Gronau

Center for Enterprise Research
Universität Potsdam
August-Bebel-Str. 89 | 14482 Potsdam
Germany

Tel. +49 331 977 3322
E-Mail ngronau@lswi.de