

Einführung in die Wirtschaftsinformatik

Teil 07 – Grundlagen der Datenmanipulation

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

- Wie werden Daten mit SELECT abgefragt und was ist der Unterschied zwischen Projektion (Spaltenauswahl) und Selektion (Zeilenauswahl)?
- Wie lassen sich Spalten umbenennen, sortieren und eindeutig identifizieren (Alias, DISTINCT, Reihenfolge)?
- Welche arithmetischen Operatoren, Verkettungsoperatoren und Datumsfunktionen können in SQL-Abfragen genutzt werden?
- Wie wird die Ausgabe über die WHERE-Klausel eingeschränkt und welche Vergleichsoperatoren (BETWEEN, IN, LIKE, IS NULL) stehen zur Verfügung?
- Wie können Bedingungen mit AND, OR und NOT kombiniert und priorisiert werden?



Abfragen mit SELECT

Operatoren und Ausdrücke

Datumsfunktionen

Einschränkung der Datenausgabe

Operatoren in zusammengesetzten Bedingungen

Hörsaal-Quiz - Einleitungsfragen

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

https://quiz.lswi.de/

pwd: ewinf

Anweisung SELECT - Projektion

Projektion - Spaltenauswahl

A	B	C	D	E

- Auswahl bestimmter Attribute (Spalten) einer Tabelle
- Anzeige der Werte aller Entities

Die Auswahl von Daten erfolgt bei der Projektion immer spaltenweise.

Betrachtung der SQL-Anweisungen am Beispiel

Ausgangssituation: Struktur der Datenbank „Firma WiProM AG“

mitarbeiter

Pers_Nr	Name	Vorname	Geschlecht	Leiter:in	Position	…	Abt_Nr*	…

abteilung

Abt_Nr	**Abt_Name**	**Betr_Teil**	**...**

betriebsteil

Betr_Tei	**Betriebsteilnam**	**PLZ**	**Or**	**Strass**	**Land**

artikel

Artikel_Nr	**Bezeichnung**	**Net_Preis**	**...**

bestellung

Bestell_Nr	**Bestelldatum**	**Kd_Nr***

bestellposten

Bestell_Nr	**Posten_Nr**	**...**	**Artikel_Nr**	**Anzahl**	**Rabatt**

kunde

Kd_Nr	**Bezeichnung**	**Kd_Typ_Code***	**Mail**	**Tel**	**Fax**	**Land**	**PLZ**	**...**

kundentyp

Kd_Typ_Code	**Bezeichnung**	**...**

projekt

Proj_Nr	Proj_Name	Proj_Leiter:in	Proj_Beginn	Pers.-kosten	…	Proj_Ende

eingesetzt_in

Pers_Nr	**Proj_Nr**

SQL-Anweisungen erstellen

- Keine Unterscheidung zwischen Groß- und Kleinschreibung in SQL-Schlüsselworten (Anweisungen)
- Semikolon - Abschluss einer Anweisung
- Keine Abkürzung oder Trennung von Schlüsselwörtern

Projektion - Spaltenauswahl

- Eine oder mehrere Zeilen
- Sinnvoll: Setzen jeder Klausel in eine eigene Zeile
- Verbesserung der Lesbarkeit durch Einrückungen

```
SCHLÜSSELWORT attribut, ...
  SCHLÜSSELWORT tabelle, ...
    SCHLÜSSELWORT bedingung, ...;
```

```
SELECT name
  FROM mitarbeiter
    WHERE vorname='Erna'
```

Nur die Feldinhalte sind casesensitiv.

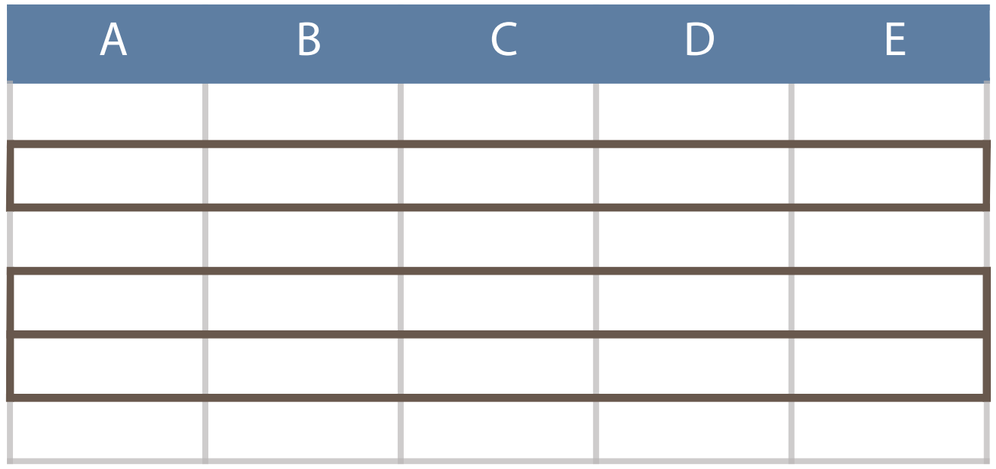
Default-Einstellungen

- Default-Ausrichtung für Überschriften: zentriert und in Großbuchstaben

ABT_NR	ABT_NAME	BETR_TEIL
…	…	…

Anweisung SELECT - Selektion

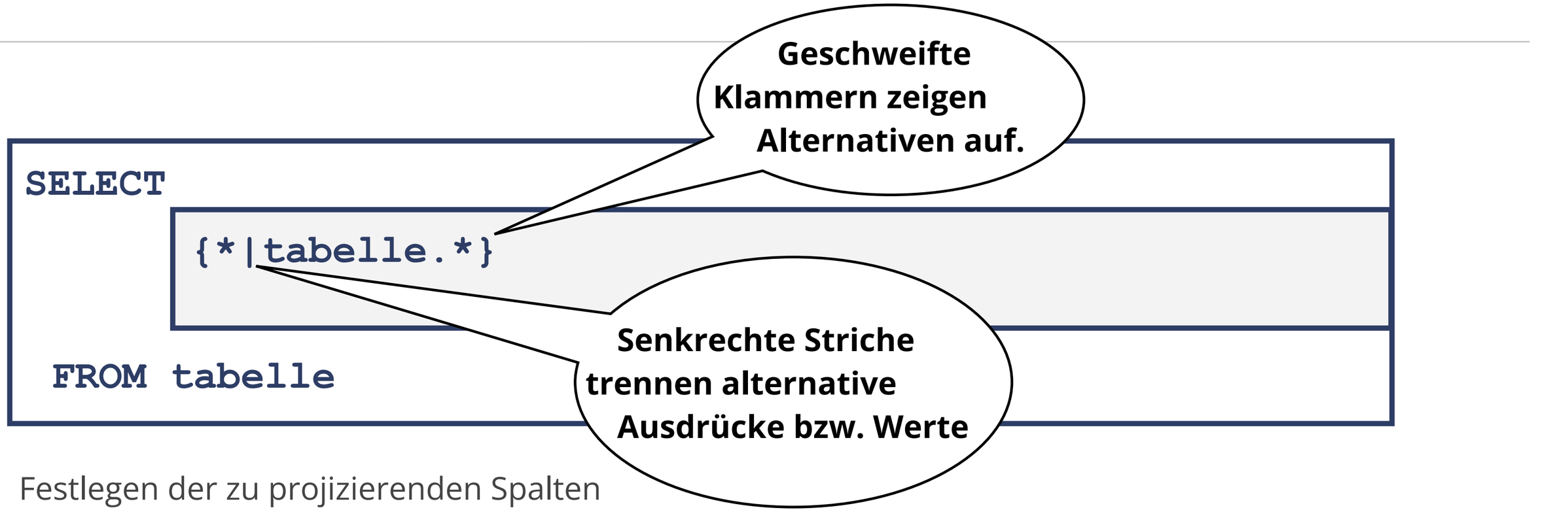
Selektion - Zeilenauswahl



- Auswahl bestimmter Tupel (Zeilen, Datensätze) einer Tabelle
- Anzeige ausgewählter Entities entsprechend einer Auswahlbedingung

Die Auswahl von Daten erfolgt bei der Selektion immer zeilenweise.

Tabellenausgabe



- Festlegen der zu projizierenden Spalten
- Keine Einschränkung - Ausgabe aller Spalten einer Tabelle
- Beispiele:
  SELECT * FROM abteilung;
  SELECT mitarbeiter.* FROM mitarbeiter;

Der Spaltenname kann durch vorangestellte Angabe des Tabellennamens (Trennung mittels Punkt) eindeutig gemacht werden.

Auswahl aller Spalten - das Sternchen *

```
SELECT * FROM mitarbeiter;
```

PERS_NR	NAME	VORNAME	GESCHLECHT	…	LEITER:IN	POSITION	…	ABT_NR	…
101001	Büchner	Edgar	männlich	…	101060	Hauptabteilungsleitung	…	260F	…
101002	Martens	Eugen	männlich	…	101060	Drehtechnik	…	260D	…
101003	Dost	Alexander	männlich	…	101059	Einkauf	…	210E	…
101004	Fuchs	Erna	weiblich	…	101060	Sekretariat	…	260D	…
101005	Rösch	Konrad	männlich	…	101047	Abteilungsleitung	…	510L	…
…	…	…	…	…	…	…	…	…	…
102059	Schulz	Daniel	männlich	…	102005	Vertriebsbeauftragte:r	…	310V	…
102060	Froese	Sascha	männlich	…	102010	Vertriebsleitung	…	410V	…
102061	Seljukow	Nikolai	männlich	…	102004	Vertriebsbeauftragte:r	…	410V	…

Nach Auswahl aller Spalten durch * wird die gesamte Tabelle angezeigt.

Auswahl bestimmter Spalten

Syntaxregel

- Spaltenauswahl über korrekte Angabe der Spaltennamen

```
SELECT pers_nr, name, vorname, abt_nr FROM mitarbeiter;
```

PERS_NR	NAME	VORNAME	ABT_NR
101001	Büchner	Edgar	260F
101002	Martens	Eugen	260D
101003	Dost	Alexander	210E
101004	Fuchs	Erna	260D
101005	Rösch	Konrad	510L
...	...	...	...
102059	Schulz	Daniel	310V
102060	Froese	Sascha	410V
102061	Seljukow	Nikolai	410V

Werden die Spaltennamen falsch geschrieben, erfolgt keine Ausgabe von Daten.

Schreibweise der Anweisungen

Auf Groß-/Kleinschreibung kommt es bei der Formulierung der Abfrage nicht an:

```
SELECT PERS_NR, VORNAME, NAME, ABT_NR FROM MITARBEITER;
```

```
SELECT pers_nr, Vorname, Name, Abt_Nr FROM Mitarbeiter;
```

```
select pers_nr, vorname, name, abt_nr from mitarbeiter;
```

- Schlüsselworte und Attributnamen nicht Case Sensitiv
- Text einer Anweisung auch über mehrere Zeilen möglich
- Semikolon am Ende der Anweisung kann weggelassen werden

PERS_NR	VORNAME	NAME	ABT_NR
101001	Edgar	Büchner	260F
101002	Eugen	Martens	260D
101003	Alexander	Dost	210E
101004	Erna	Fuchs	260D
101005	Konrad	Rösch	510L
…	…	…	…

Spalten-Aliasnamen

Bedeutung

- Umbenennung von Spaltenüberschriften
- Sinnvoll bei Berechnungen - Spaltentitel

Regeln

- Angabe direkt hinter Spaltennamen
- Optionale Angabe des Schlüsselwortes AS zwischen Spalten- und Aliasnamen
- Bei Angabe von Leer- oder Sonderzeichen oder Groß-/ Kleinschreibung - doppelte Anführungszeichen (" „)

Pseudocode

```
SELECT spalte|Ausdruck [alias], ...}
FROM tabelle
```

Eckige Klammern zeigen optional auswählbare Ausdrücke.

Greenberg 2002, S. 66

Spaltenreihenfolge und Alias

Modifizierte Spaltenauswahl

```
SELECT pers_nr personalnummer, abt_nr abteilungsnummer, name
FROM mitarbeiter;
```

PERSONALNUMMER	ABTEILUNGSNUMMER	NAME
101001	260F	Büchner
101002	260D	Martens
101003	210E	Dost
101004	260D	Fuchs
101005	510L	Rösch
...	...	...
102059	310V	Schulz
102060	410V	Froese
102061	410V	Seljukow

Die Reihenfolge und Spaltennamen können je nach Bedarf modifiziert werden.

Angabe von Spalten-Aliasnamen

```
SELECT name AS Familienname, gehalt  AS Verdienst
FROM mitarbeiter;
```

FAMILIENNAME	VERDIENST
Büchner	
...	...
Froese	5520
Seljukow	1500

```
SELECT pers_nr Personalnr, abt_nr Abteilungsnr, name
FROM mitarbeiter;
```

PERSONALNR	ABTEILUNGSNR	NAME
101001	260F	Büchner
...	...	...

Das Schlüsselwort AS weist explizit auf den Alias hin.

Greenberg 2002, S. 67

Alias mit Groß-/Kleinschreibung

- Alternative bei Groß-/Kleinschreibung über Anführungszeichen " "
- Notwendig bei Auftreten von Leerzeichen im Alias

```
SELECT pers_nr AS "Personalnr.", abt_nr, name AS "Familienname"
FROM mitarbeiter;
```

Personalnr.	ABT_NR	Familienname
101001	260F	Büchner
101002	260D	Martens
...	...	...

```
SELECT name "Familienname", gehalt "Jährl. Gehalt"
FROM mitarbeiter;
```

Familienname	Jährl. Gehalt
...	...
Hofmann	43320
Würz	74280
Joseph	21600
Cho	8400
Saizew	27600



Abfragen mit SELECT

Operatoren und Ausdrücke

Datumsfunktionen

Einschränkung der Datenausgabe

Operatoren in zusammengesetzten Bedingungen

Arithmetische Ausdrücke

- Erstellen von Ausdrücken mit Hilfe arithmetischer Operatoren
- Zulässig in Werten sind nur Daten vom Typ NUMBER und DATE

Operator	Beschreibung
+	Addieren
-	Subtrahieren
*	Multiplizieren
/	Dividieren

```
SELECT attribut, attribut * Wert
FROM tabelle;
```

Operatoren können in jeder Klausel einer SQL-Anweisung mit Ausnahme der FROM-Klausel verwendet werden.

Greenberg 2002, S. 59

Arithmetische Operatoren

- Zusätzliche Spalte mit Berechnungsergebnis
- Existiert nicht in der Tabelle "mitarbeiter"

```
SELECT name, gehalt, gehalt * 1.02 Gehaltserhöhung
FROM mitarbeiter;
```

NAME	GEHALT	GEHALTSERHÖHUNG
Büchner	10430	10638,6
Martens	2400	2448
Dost	3100	3162
Fuchs	3600	3672
Rösch	6590	6721,8
...	...	...

Zur besseren Lesbarkeit können Leerzeichen vor und nach dem arithmetischen Operator eingefügt werden.

Operatorpriorität

- Punkt- vor Strichrechnung: Multiplikationen und Divisionen vor Additionen und Subtraktionen
- Auswertung von Operatoren derselben Priorität von links nach rechts
- Einsatz von Klammern zur Priorisierung der Auswertung
- Bessere Lesbarkeit von Anweisungen

*** / + -**

a+b+c →

a + b * c <> (a + b) * c

Greenberg 2002, S. 61

Berechnung mit priorisierten Operatoren

- Berechnung des Jahresgehaltes plus Einmalzahlung

```
SELECT name, gehalt, 50 + gehalt * 12
FROM mitarbeiter;
```

oder alternativ

```
SELECT name, gehalt, 50 + (gehalt * 12 )
FROM mitarbeiter;
```

NAME	GEHALT	50+GEHALT*12
Büchner	10430	125210
Martens	2400	28850
Dost	3100	37250
Fuchs	3600	43250
Rösch	6590	79130
...	...	...

NULL-Werte in Feldern

- Repräsentation nicht verfügbarer, nicht zugewiesener, unbekannter oder nicht anwendbarer Werte
- Nicht dasselbe wie der Wert 0 (Zahl Null) oder Leerzeichen

```
SELECT name, vorname, akad_titel
FROM mitarbeiter;
```

NAME	VORNAME	AKAD_TITEL
Weber	Edzard	Dr.
Melzer	Thomas	
Bormann	Samira	Dr.
Hofmann	Katja	
Würz	Hannah	
…	…	…
Petersen	Helmut	
Schulz-	Paul	Dr. Ing.
Plenk	Karl	
Engel	Lothar	
Roth	Katharina	
…		

Leerwerte:
Felder ohne Inhalte

Dieses Problem werden wir in Vorlesung 8 „erweiterte Funktionen“ lösen.

Greenberg 2002, S. 64

Verkettungsoperator

- Verkettet Spalten oder Zeichenfolgen mit anderen Spalten
- Darstellung durch zwei senkrechte Striche (||)
- Erstellt Ergebnisspalte mit einem Zeichenausdruck
- Ausgabe der Zeichenkette ohne Leerzeichen zwischen verketteten Elementen

```
SELECT spalte1 || spalte2 [... || spalte_n]
FROM tabelle;
```

Hinweis: Im weiteren Verlauf wird statt **Attribut** *synonym der Begriff* ***Spalte*** *genutzt.*

Greenberg 2002, S. 68

Text über Verkettungsoperatoren verbinden

```
SELECT vorname || name || geschlecht AS "Mitarbeiter"
FROM mitarbeiter;
```

NAME	VORNAME	GESCHLECHT
Plenk	Karl	männlich
Engel	Lothar	männlich
Roth	Katharina	weiblich
Riekhoff	Monika	weiblich
Groß	Hildegard	weiblich
Eckert	Hans	männlich
...	...	...

Mitarbeiter
KarlPlenkmännlich
LotharEngelmännlich
KatharinaRothweiblich
MonikaRiekhoffweiblich
HildegardGroßweiblich
HansEckertmännlich
...

Die Spalten können in beliebiger Reihenfolge verkettet werden

Ergebnis liefert noch keine befriedigende Ausgabe

Das Schlüsselwort AS vor dem Aliasnamen verbessert die Lesbarkeit der SELECT-Klausel.

Literale Zeichenfolgen

- Zeichen, Zahl oder Datum in der SELECT-Liste
- Literale Datums- und Zeichenwerte stehen in Hochkommata
- Ausgabe jeder Zeichenfolge einmal für jede zurückgegebene Zeile

Achtung: Dies ist Teil des Pseudocodes, nicht Teil des Verkettungs-operators

```
SELECT spalte1 || 'zeichen'|'datum'|zahl || spalte2
[… || spalte_n]
FROM tabelle;
```

Verkettungsmöglichkeiten zwischen Spalten entweder über Zeichen oder Datumswert oder numerischen Wert

Literale gestalten Spaltenüberschriften in besser lesbarer Form.

Greenberg 2002, S. 70

Literale und Lesbarkeit von Felddaten

```
SELECT 'Mitarbeiter ' || name || ' (' || geschlecht || ') hat
die Position ' || position || '.' AS "Beschäftigtenliste"
FROM mitarbeiter;
```

Beschäftigtenliste
Mitarbeiter Plenk (männlich) hat die Position Konstruktion.
Mitarbeiter Engel (männlich) hat die Position Buchhaltung.
Mitarbeiter Roth (weiblich) hat die Position Controlling.
Mitarbeiter Riekhoff (weiblich) hat die Position Chefsekretariat.
Mitarbeiter Groß (weiblich) hat die Position Sekretariat.
Mitarbeiter Eckert (männlich) hat die Position Wirtschaftsingenieurwesen.
...



Abfragen mit SELECT

Operatoren und Ausdrücke

Datumsfunktionen

Einschränkung der Datenausgabe

Operatoren in zusammengesetzten Bedingungen

Hörsaal-Quiz - Recap erste Hälfte

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

https://quiz.lswi.de/

pwd: ewinf

Datumsfunktionen

- Möglichkeiten der Arbeit mit Datumswerten
- Verwendung des Datentyps DATE
- Eingabe der Datumswerte in Hochkommata
- Default-Anzeigeformat - DD.MM.YYYY
- Einfachste Form: Darstellung des Datums (Feldtyp DATE) als Zeichenkette

```
SELECT SYSDATE
FROM DUAL;
```

SYSDATE
14.12.2020

Alle Datumsfunktionen geben einen Wert vom Datentyp DATE zurück.

Rechnen mit Datumswerten

- Addition/Subtraktion einer Anzahl von Tagen von einem Datum
- Differenzbildung zweier Datumswerte - Anzahl der dazwischen liegenden Tage

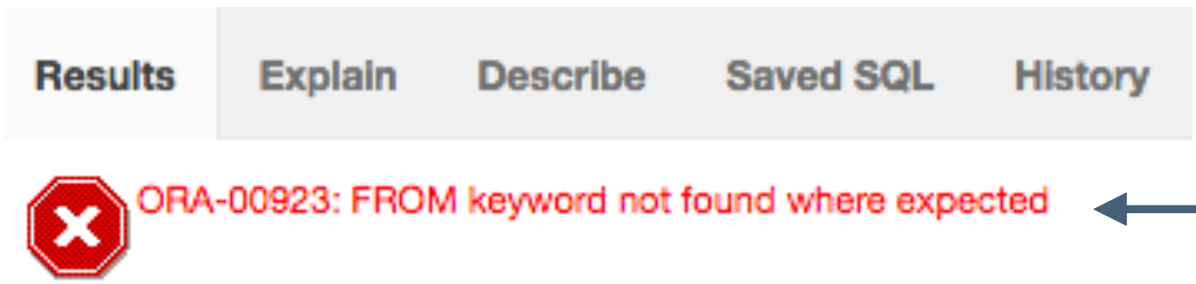
Operation	Ergebnis	Beschreibung
Datum + Zahl	Datum	Addition Tage zu Datum
Datum - Zahl	Datum	Subtraktion Tage von Datum
Datum1 - Datum2	Anzahl von Tagen	Subtraktion Datum2 von Datum1

Greenberg 2002, S. 141

Errechnen von Differenzen bei Datumswerten

- Beispiel: Anzeige des aktuellen Alters der einzelnen Mitarbeiter

```
SELECT name, vorname, (SYSDATE - geburtstag)/365 AS Alter
FROM mitarbeiter;
```




```
SELECT name, vorname, (SYSDATE - geburtstag)/365 AS "Alter"
FROM mitarbeiter;
```

NAME	VORNAME	ALTER
Schwarz	Konstantin	59,717775558092338914256
Klemm	Ljudmilla	62,383528982749873160832
Jörgensen	Jane	41,358871448503297818366
Schäfer	Juri	48,336953640284119736174
Herrmann	Nadja	52,12873446220192795352
...	...	...

‚Alter' kann nicht als Spaltennamen oder Alias verwendet werden, da es auch als Schlüsselwort identifiziert werden könnte.

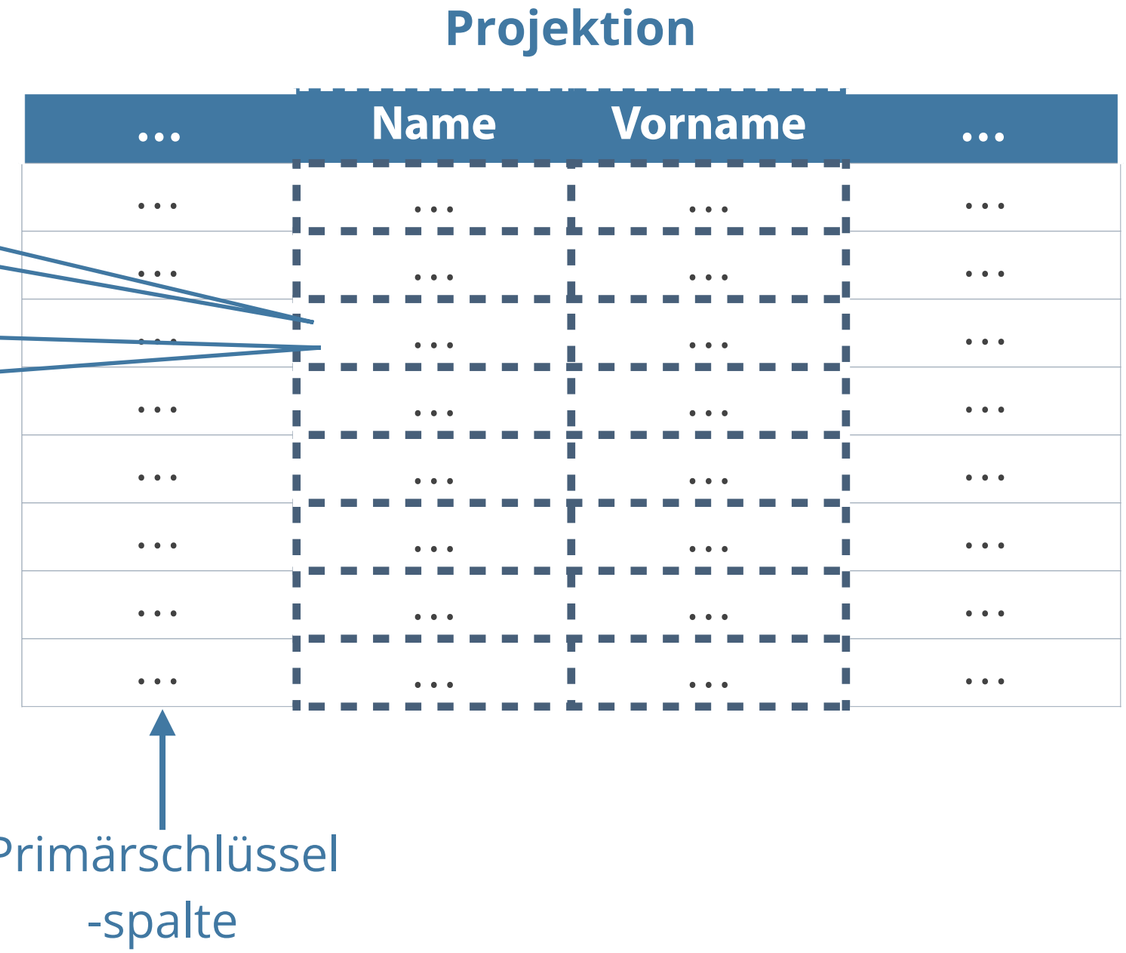
Zusammenfassung - Projektion

SELECT mitarbeiter.name, mitarbeiter.vorname

oder

SELECT name, vorname

FROM mitarbeiter



Mit der Projektion werden alle Daten aus den „angesprochenen“ Spalten einer Datenbanktabelle in einer temporären Ausgabetabelle angezeigt.



Abfragen mit SELECT

Operatoren und Ausdrücke

Datumsfunktionen

Einschränkung der Datenausgabe

Operatoren in zusammengesetzten Bedingungen

Einschränkung von Zeilen durch Auswahl

- Einschränkung der zurückgegebenen Zeilen über die WHERE-Klausel
- Angabe der WHERE-Klausel grundsätzlich nach der FROM-Klausel

```
SELECT * | {spalte | ausdruck, ...}
FROM tabelle
WHERE Bedingung(en);
```

Die Reihenfolge der Schlüsselworte ist durch die Syntax festgelegt und nicht änderbar.

Anwendung der WHERE-Klausel



Bedingung der WHERE-Klausel schränkt Ausgabe ein

```
SELECT pers_nr, geschlecht, name, position,
abt_nr
FROM mitarbeiter
WHERE geschlecht = 'männlich';
```

Zur Erinnerung...

```
SELECT name
FROM mitarbeiter
WHERE vorname = 'Erna';
```

Beispiel: Nur Auswahl aller männlichen Mitarbeiter

PERS_NR	GESCHLECHT	NAME	POSITION	ABT_NR
102006	männlich	Malossek	Vertriebsbeauftragte:r	110V
102007	männlich	Karomi	Vertriebsbeauftragte:r	210V
102009	männlich	Stadelmann	Vertriebsbeauftragte:r	110V
102010	männlich	Kellner	Geschäftsführung	10VL
102011	männlich	Peplinski	Vertriebsgruppenleitung	410V
102017	männlich	Hinz	Technische Mitarbeit	10VL
102018	männlich	Zengel	Vertriebsbeauftragte:r	310V
102020	männlich	Kunze	Vertriebsleitung	310V
102021	männlich	Straube	Technische Mitarbeit	10VL
…	…	…	…	…

Zeichenfolgen und Datumsangaben

- Unterscheidung bei Zeichenwerten in Groß-/Kleinschreibung

```
SELECT pers_nr, name, position, abt_nr
FROM mitarbeiter
WHERE position = 'Sekretariat';
```

- Datumswerte sind formatabhängig
- Default-Ausgabeformat für Datumsangaben - DD.MM.YYYY

```
SELECT pers_nr, name, position, abt_nr
FROM mitarbeiter
WHERE geburtstag = '12.12.1967';
```

PERS_NR	NAME	POSITION	ABT_NR
101070	Krajcsir	Chief Technical Officer	-

Zeichenfolgen und Datumswerte werden prinzipiell in Hochkommata '...' gesetzt.

Quelle: Greenberg 2002, S. 96

Anzeige und Ausblenden mehrfach vorhandener Werte

Abfragen → Anzeige aller Zeilen, auch mehrfach vorhandene

```
SELECT position FROM mitarbeiter
ORDER BY position;
```

Ausblenden von Doppelungen mit DISTINCT („verschieden“)

```
SELECT DISTINCT position FROM mitarbeiter
ORDER BY position;
```

Mehrere Mitarbeiter sind als „Elektriker“ bzw. als „Elektroniker“ angestellt

*→ **Anzeige** redundanter Daten*

POSITION
…
Elektrotechnik
Elektrotechnik
Elektronik
Elektronik
Elektronik
Endkontrolle
Entwicklungsingenieurwesen
Entwicklungsingenieurwesen
…

POSITION
…
Elektrotechnik
Elektronik
Elektronik
Endkontrolle
Entwicklungsingenieurwesen
Entwicklungsingenieurwesen
Geschäftsführung
Hauptabteilungsleitung
…

*Mit Hilfe von DISTINCT **einmalige Ausgabe** der Position „Elektriker" bzw. „Elektroniker“*

*→**Ausblenden** redundanter Daten*

Beeinflussung der Ergebnisausgabe mit DISTINCT

- Angabe mehrerer Spalten nach DISTINCT
- Auswirkung auf alle gewählten Spalten
- Ausgabe aller unterschiedlichen Kombinationen der Spalten

```
SELECT DISTINCT position, abt_nr
FROM mitarbeiter;
```

POSITION	ABT_NR
Drehtechnik	260D
Einkauf	210E
Sekretariat	260D
Abteilungsleitung	510L
Abteilungsleitung	320M
Abteilungsleitung	310T
Schleiftechnik	260K
Buchhaltung	420F
...	...

Mehrfachausgabe von Zeilen mit den selben Wertekombinationen wird unterdrückt

Operatoren zum Vergleich von Werten

- Innerhalb der WHERE-Klausel - Vergleich für Zeichen sowie Zeichenketten, numerische und Datumswerte

```
SELECT ... FROM ...
WHERE ausdruck operator wert;
```

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

```
SELECT name, gehalt FROM mitarbeiter
WHERE gehalt >= 2500;
```

Quelle: Greenberg 2002, S. 97

Spezielle Vergleichsoperatoren

- Operatoren mit stärkerer bzw. erweiterter Filterwirkung
- Explizite Erfassung leerer Felder über IS NULL-Operator

Operator	Bedeutung
BETWEEN ... AND ...	**Zwischen zwei Werten (einschließlich dieser Werte)**
IN (*Werteliste*)	**Entspricht einem oder mehreren Werte aus einer Werteliste**
LIKE	**Entspricht einem zu definierenden Zeichenmuster**
IS NULL	**Ist ein NULL-Wert**

Quelle: Greenberg 2002, S. 99

Einsatz der Operatoren BETWEEN und IN

BETWEEN – Anzeigen von Zeilen basierend auf Wertebereich

```
SELECT name, gehalt
FROM mitarbeiter
WHERE gehalt BETWEEN 2500 AND 3500;
```

NAME	GEHALT
Gast	2600
Dost	3100
...	...
Fritzsche	2910
Bode	2550

Zugehörigkeitsoperator IN – Prüfung einer Werteliste

```
SELECT name, gehalt, abt_nr
FROM mitarbeiter
WHERE abt_nr IN
('230P','240P','107R');
```

NAME	GEHALT	ABT_NR
Groß	2800	230P
Eckert	4270	230P
...	...	...
Van Der Biest	5180	107R
Nograsek	3770	107R
...	...	...
Schröder	4400	240P

Der Operator LIKE für eine bessere Suche

LIKE für Zeichenfolgenwerte

```
SELECT name, vorname, geschlecht
FROM mitarbeiter
WHERE name LIKE 'Sch%';
```

NAME	VORNAME	GESCHLECHT
Scherz	Nicole	weiblich
Schöneck	Sascha	divers
...	...	...
Schulz	Carola	weiblich
Schlank	Nils	männlich

Die Platzhalterzeichen % (0 bis n Zeichen) und _ (exakt 1 Zeichen) in LIKE können kombiniert werden.



Abfragen mit SELECT

Operatoren und Ausdrücke

Datumsfunktionen

Einschränkung der Datenausgabe

Operatoren in zusammengesetzten Bedingungen

Logische Operatoren

Verknüpfen mehrerer Bedingungen in einer Abfrage

Operator	Bedeutung
AND	**Gibt TRUE zurück, wenn beide Komponentenbedingungen wahr sind**
OR	**Gibt TRUE zurück, wenn mindestens eine der beiden Komponentenbedingungen wahr ist**
NOT	**Gibt TRUE zurück, wenn die nachfolgende Bedingung falsch ist**

Quelle: Greenberg 2002, S. 105

Operator AND

AND – Erfüllung beider Bedingungen (wahr)

```
SELECT pers_nr, name, position, gehalt
FROM mitarbeiter
WHERE gehalt >= 6500 AND position LIKE '%leitung%';
```

PERS_NR	NAME	POSITION	GEHALT
101001	Büchner	Hauptabteilungsleitung	10430
101007	Kettler	Abteilungsleitung	8080
101016	Klein	Abteilungsleitung	7960
101042	Schmiedel	Abteilungsleitung	7210
101015	Grauer	Abteilungsleitung	6600

AND-Wahrheitstabelle

AND		Erste Bedingung ist…		
		WAHR	FALSCH	NULL
Zweite Bedingung ist…	WAHR	WAHR	FALSCH	NULL
	FALSCH	FALSCH	FALSCH	FALSCH
	NULL	NULL	FALSCH	NULL

Quelle: Greenberg 2002, S. 106

Operator OR

OR – mindestens eine der beiden Bedingungen muss wahr sein

```
SELECT pers_nr, name, position, gehalt
FROM mitarbeiter
WHERE gehalt >= 4500 OR position LIKE '%kauf%';
```

PERS_NR	NAME	POSITION	GEHALT
101001	Büchner	Hauptabteilungsleitung	10430
101003	Dost	Einkauf	3100
…	…	…	…
101056	Hein	Einkauf	2600
101059	Ernst	Abteilungsleitung	6500
…	…	…	…

OR-Wahrheitstabelle

OR		Erste Bedingung ist…		
		WAHR	FALSCH	NULL
Zweite Bedingung ist…	WAHR	WAHR	WAHR	WAHR
	FALSCH	WAHR	FALSCH	NULL
	NULL	WAHR	NULL	NULL

Quelle: Greenberg 2002, S. 107

Operator NOT

NOT – im Sinne von "keine Übereinstimmung"

```
SELECT name, vorname, position
FROM mitarbeiter
WHERE position NOT IN
('Abteilungsleitung','Hauptabteilungsleitung');
```

NAME	VORNAME	POSITION
Martens	Eugen	Drehtechnik
Dost	Alexander	Einkauf
…	…	…
Neumann	Michael	Systementwicklungsingenieurwesen
Altmann	Bernd	Lagerlogistik

NOT-Wahrheitstabelle

NOT	Erste Bedingung ist…		
	WAHR	FALSCH	NULL
Ergebnis ist	FALSCH	WAHR	NULL

Quelle: Greenberg 2002, S. 108

Prioritätsregeln der Operatoren

Auswertungsreihenfolge	Operator
1	Arithmetische Operatoren
2	Verkettungsoperator
3	Vergleichsoperatoren
4	IS [NOT] NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	Logischer Operator NOT
7	Logischer Operator AND
8	Logischer Operator OR

Die Auswertungsreihenfolge lässt sich durch Setzen von Klammern ändern.

Quelle: Greenberg 2002, S. 109

Zusammenfassung – Projektion und Selektion

```
SELECT name, vorname
```

Projektion

```
FROM mitarbeiter
```

```
WHERE abt_nr <> '20' AND position = 'Buchhaltung'
```

Selektion

Durch Projektion werden die Spalten, durch Selektion die Zeilen einer Tabelle ausgewählt.

Hörsaal-Quiz - Wissensvertiefung

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

https://quiz.lswi.de/

pwd: ewinf

Kontrollfragen

- Welche Grundregeln gibt es bei der Anwendung von Schlüsselwörtern?
- Worin unterscheiden sich Selektion und Projektion?
- Gibt es die Möglichkeit, über eine Anweisung die Struktur einer Tabelle auszugeben?
- Kann neben der Spaltenausgabe auch eine mathematische Operation mit Werten in einer Spalte ausgeführt werden?
- Kann ein Datumswert mit einer mathematischen Operation verknüpft werden?
- Welche Bedeutung kommt der Einschränkung der Ausgabe redundanter Daten zu?

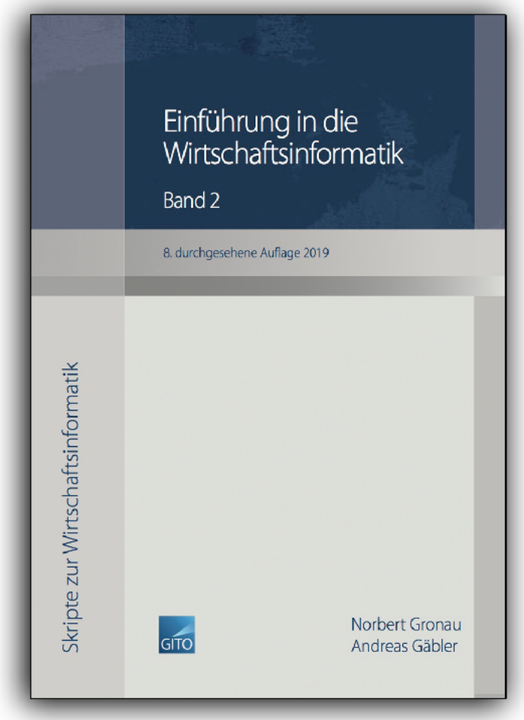
Literatur

Heuer, A./Saake, G.: Datenbanken - Konzepte und Sprachen; 6. Auflage, 2018, mitp

Elmazri, R./Navathe, S. B.: Grundlagen von Datenbanksystemen; 3. Auflage, 2009, Pearson

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

Zum Nachlesen



Gronau, N., Gäbler, A.:
Einführung in die Wirtschaftsinformatik, Band 2
8. überarbeitete Auflage

GITO Verlag Berlin 2019. ISBN 978-3-95545-285-8

Kontakt

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

Universität Potsdam
Karl-Marx-Str. 67 | 14482 Potsdam
Germany

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