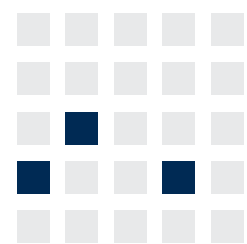




# Einführung in die Wirtschaftsinformatik

Teil 7 - Grundlagen der Datenmanipulation

Wintersemester 2020/2021



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*

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

*Tel* +49 331 977 3322

*Fax* +49 331 977 3406

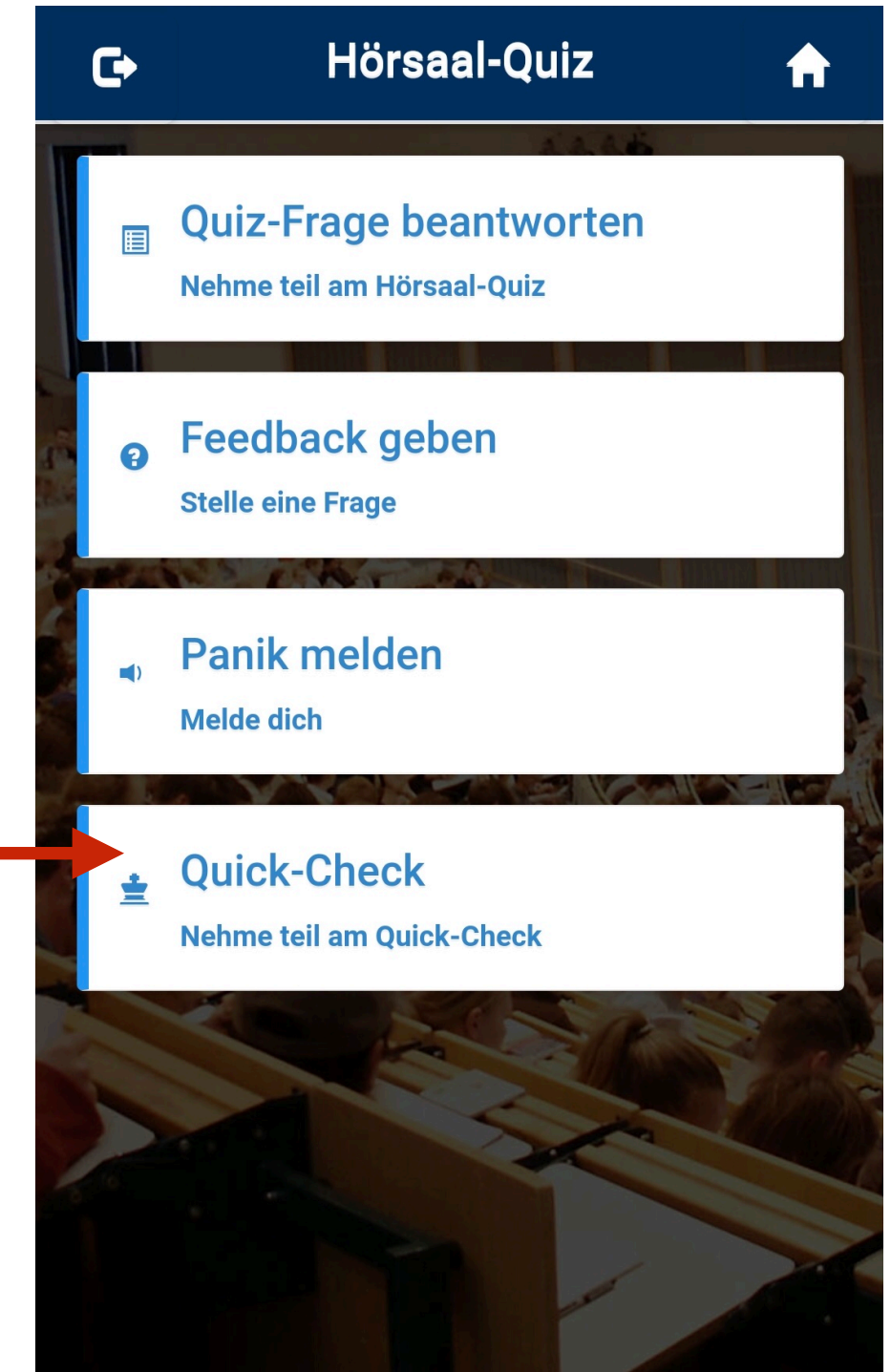
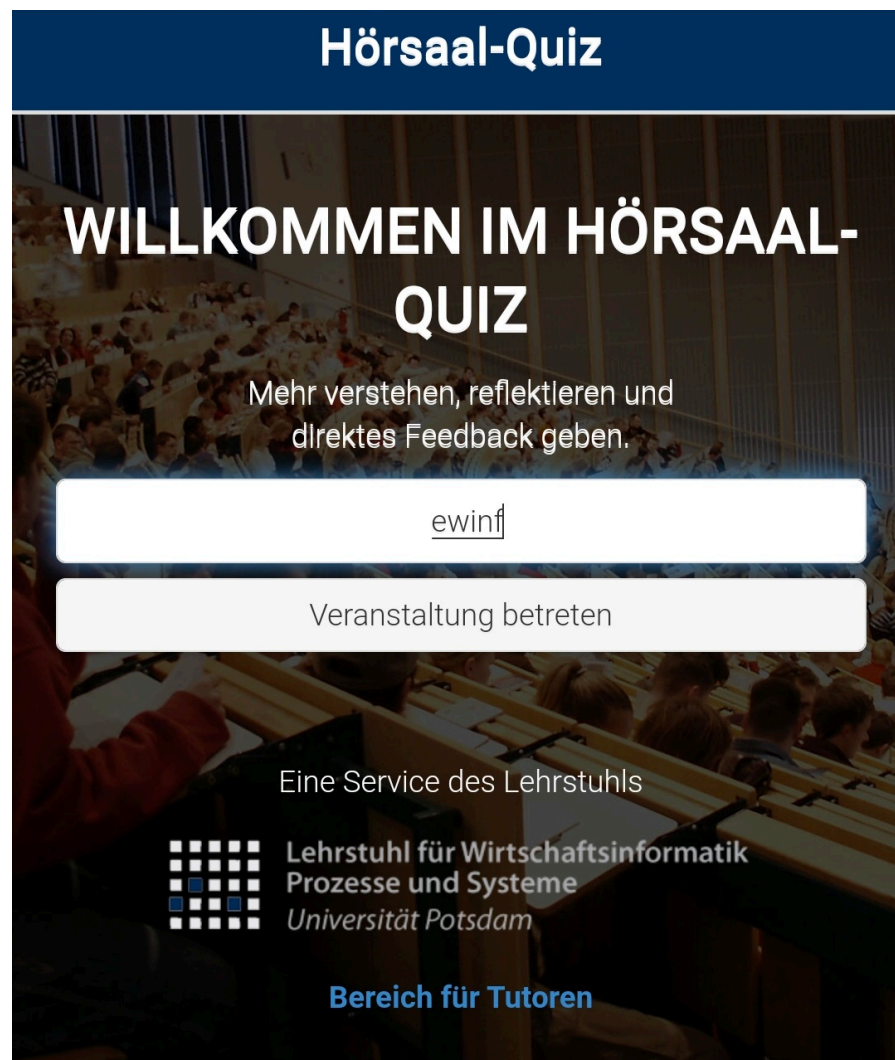
*E-Mail* [ngronau@lswi.de](mailto:ngronau@lswi.de)

*Web* [lswi.de](http://lswi.de)



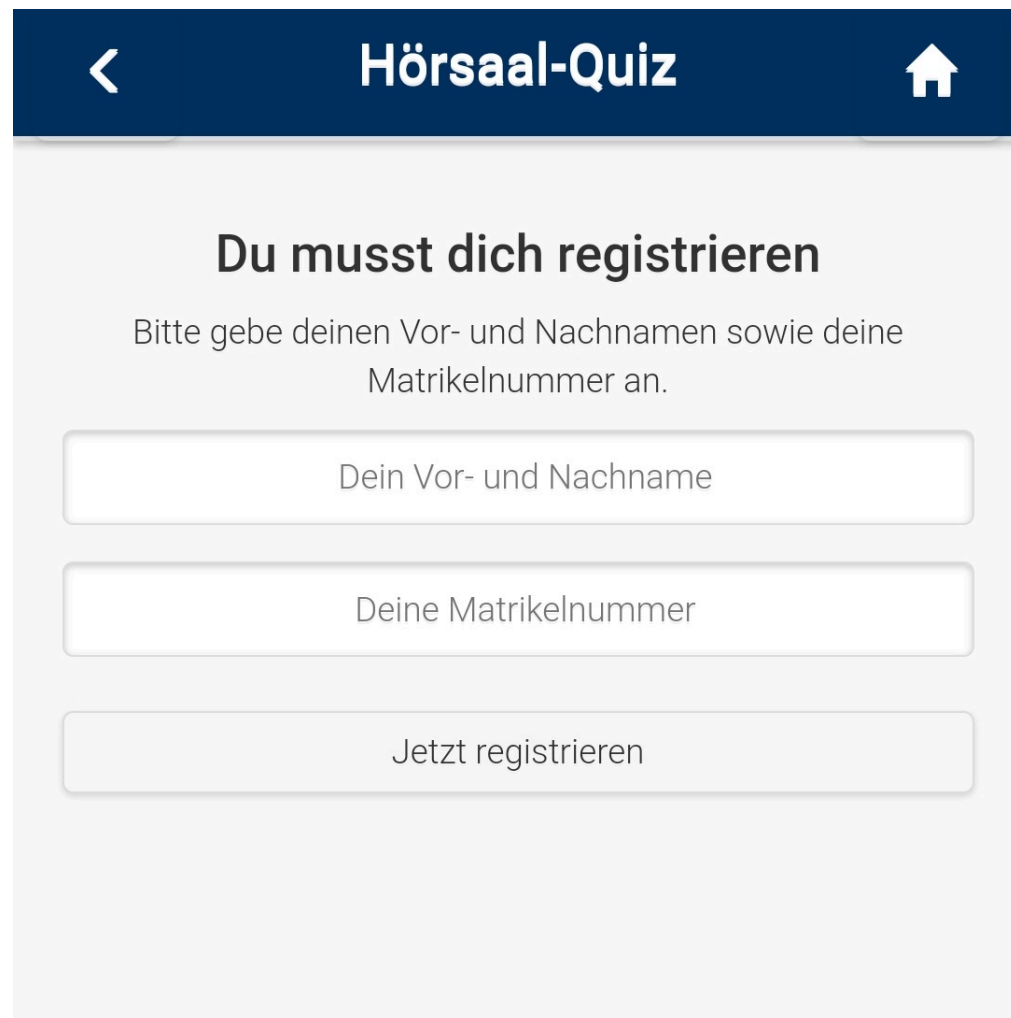
# Wiederholung Quick-Checks

- Zur Teilnahme am Quick-Check rufen Sie bitte unsere Lehrstuhl-App unter [quiz.lswi.de](https://quiz.lswi.de) auf
- Veranstaltungskürzel: ewinf
- Quick Checks bringen Bonuspunkte, welche in der Klausur angerechnet werden können
- Insgesamt 9 Bonuspunkte durch Quickchecks zu erreichen



# Wiederholung Quick-Checks

- Zur Anmeldung zum Quickcheck Name und Matr. Nummer angeben
- Achtung: Matrikelnummern mit einer 8 vorne werden momentan nicht akzeptiert
- Schreiben Sie zur Lösung ihre Matr. Nummer hinter Ihren Namen und ersetzen Sie die 8 vorne in ihrer Matr. Nummer durch eine 7 wie im Beispiel zu sehen



**Hörsaal-Quiz**

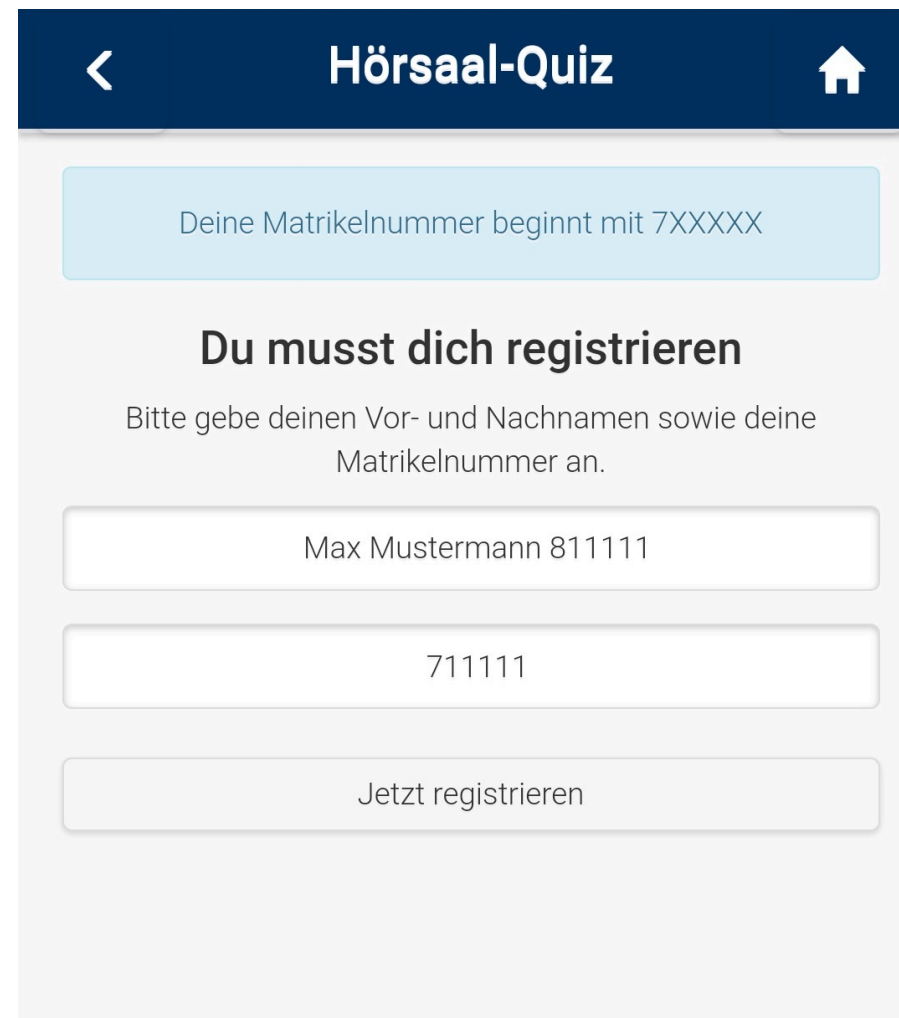
**Du musst dich registrieren**

Bitte gebe deinen Vor- und Nachnamen sowie deine Matrikelnummer an.

Dein Vor- und Nachname

Deine Matrikelnummer

Jetzt registrieren



**Hörsaal-Quiz**

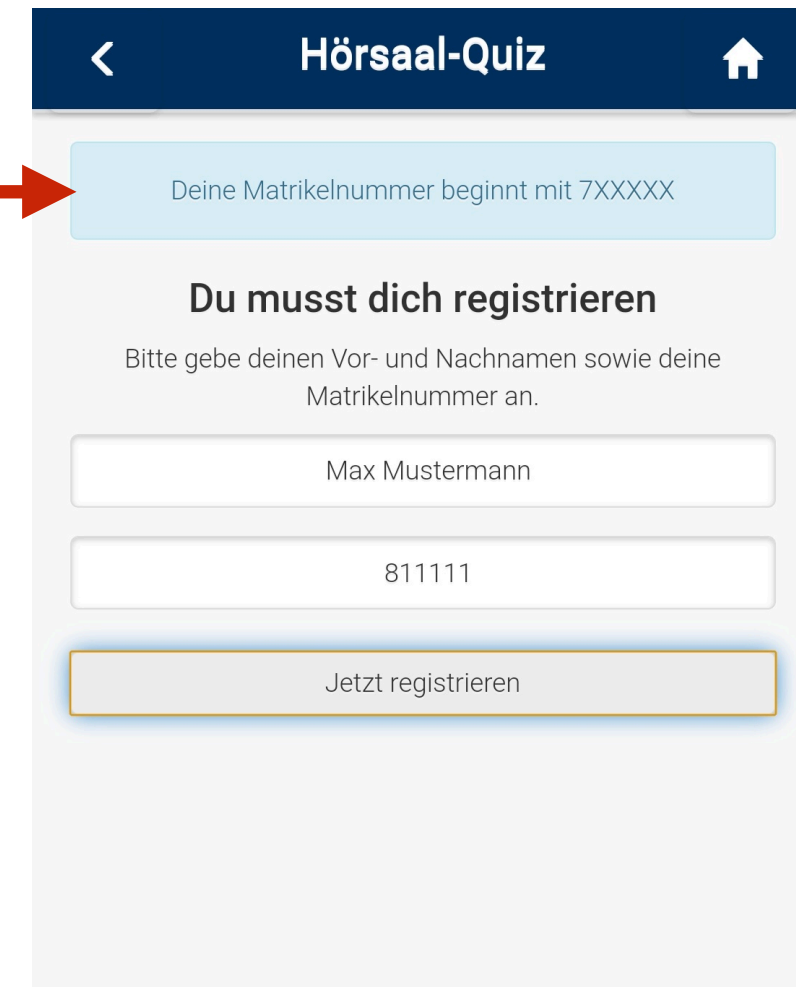
**Du musst dich registrieren**

Bitte gebe deinen Vor- und Nachnamen sowie deine Matrikelnummer an.

Max Mustermann 811111

711111

Jetzt registrieren



**Hörsaal-Quiz**

Deine Matrikelnummer beginnt mit 7XXXXX

**Du musst dich registrieren**

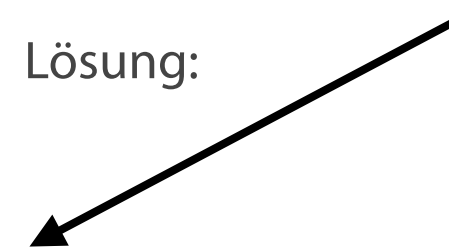
Bitte gebe deinen Vor- und Nachnamen sowie deine Matrikelnummer an.

Max Mustermann

811111

Jetzt registrieren

Lösung:







## **Abfragen mit SELECT**

Operatoren und Ausdrücke

Datumsfunktionen

Einschränkung der Datenausgabe



# Betrachtung der SQL-Anweisungen am Beispiel

## Ausgangssituation: Struktur der Datenbank „Firma WiProM AG“

mitarbeiter

Pers\_Nr | Name | Vorname | Anrede | Leiter | Position | ... | Abt\_Nr\* | ...

abteilung

Abt\_Nr | Abt\_Name | Betr\_Teil\* | ...

betriebsteil

Betr\_Teil | Betriebsteilname | PLZ | Ort | Strasse | Land

artikel

Artikel\_Nr | Bezeichnung | Net\_Preis | Artikelgruppe | ...

bestellung

Bestell\_Nr | Bestelldatum | Kd\_Nr\* | Vertrieb\_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 | Proj\_Beginn | Pers.-kosten | ... | Proj\_Ende

eingesetzt\_in

Pers\_Nr | Proj\_Nr

# SQL-Anweisungen erstellen

---

- Keine Unterscheidung zwischen Groß- und Kleinschreibung in SQL-Schlüsselwörtern (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, ...;
```

**Nur die Feldinhalte sind casesensitiv.**

# Default-Einstellungen

---

- Default-Ausrichtung für Überschriften: zentriert und in Großbuchstaben

ABT_NR	ABT_NAME	BETR_TEIL
...	...	...

# 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.**



# Anweisung SELECT - Selektion

---

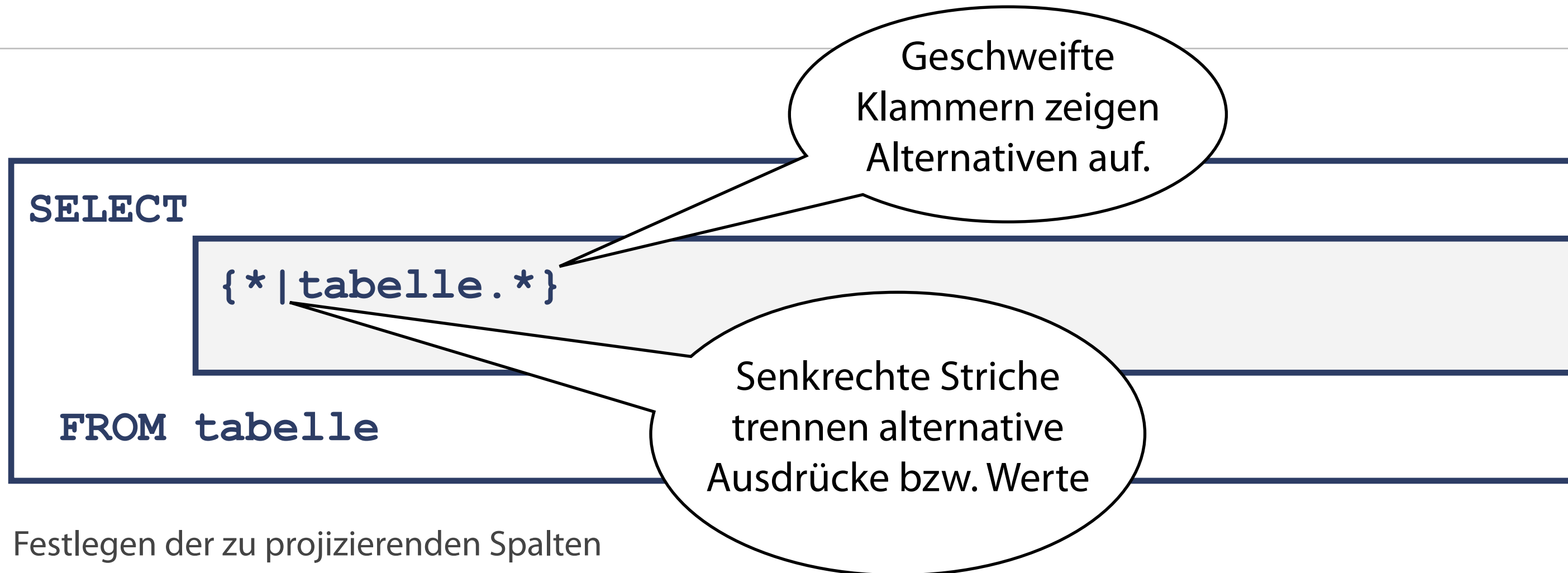
## Selektion - Zeilenauswahl

A	B	C	D	E

- 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	ANREDE	...	LEITER	POSITION	...	ABT_NR	...
101001	Büchner	Edgar	Herr	...	101060	Hauptabteilungsleiter	...	260F	...
101002	Martens	Eugen	Herr	...	101060	Dreher	...	260D	...
101003	Dost	Alexander	Herr	...	101059	Einkäufer	...	210E	...
101004	Fuchs	Erna	Frau	...	101060	Sekretärin	...	260D	...
101005	Rösch	Konrad	Herr	...	101047	Abteilungsleiter	...	510L	...
...	...	...	...	...	...	...	...	...	...
102059	Schulz	Daniel	Herr	...	102005	Vertriebsbeauftragter	...	310V	...
102060	Froese	Sascha	Herr	...	102010	Vertriebsleiter	...	410V	...
102061	Seljukow	Nikolai	Herr	...	102004	Vertriebsbeauftragter	...	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
```

**Standardmäßig werden Aliasnamen in Überschriften in Großbuchstaben angezeigt.**

Eckige  
Klammern zeigen  
optional auswählbare  
Ausdrücke



# 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*12*(1 + provision) AS  
Jahresgehalt  
FROM mitarbeiter;
```

FAMILIENNAME	JAHRESGEHALT
Büchner	
...	...
Froese	66240
Seljukow	18000

```
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.**

# 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 "Familiennamen"  
FROM mitarbeiter;
```

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

```
SELECT name "Familiennamen", gehalt*12  
"Jährl. Gehalt" FROM mitarbeiter;
```

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





Abfragen mit SELECT

**Operatoren und Ausdrücke**

Datumsfunktionen

Einschränkung der Datenausgabe



# 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.**

# 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**

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

# 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
Walker	John A.	
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 einer der nächsten  
VL lösen**

# 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

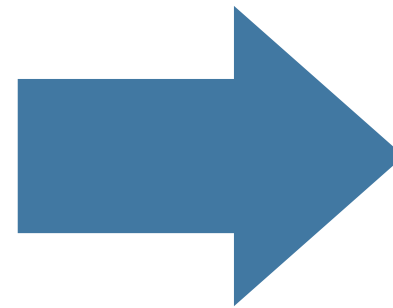
```
SELECT spalte1 || spalte2 [...] || spalte_n  
FROM tabelle;
```

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

# Text über Verkettungsoperatoren verbinden

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

NAME	VORNAME	ANREDE
Plenk	Karl	H
Engel	Lothar	H
Roth	Katharina	F
Riekhoff	Monika	F
Groß	Hildegard	F
Eckert	Hans	H
...	...	...



Mitarbeiter
HKarlPlenk
HLotharEngel
FKatharinaRoth
FMonikaRiekhoff
FHildegardGroß
HHansEckert
...

*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  
Verkettungsoperators

```
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.**



# Literale und Lesbarkeit von Felddaten

---

```
SELECT anrede || '.' || name || ' ist beschäftigt als ' ||  
Position "Beschäftigtenliste"  
FROM mitarbeiter;
```

## Beschäftigtenliste

H. Plenk ist beschäftigt als Konstrukteur
H. Engel ist beschäftigt als Buchhalter
F. Roth ist beschäftigt als Controller
F. Riekhoff ist beschäftigt als Chefsekretärin
F. Groß ist beschäftigt als Sekretärin
H. Eckert ist beschäftigt als Wirtschaftsingenieur
...





Abfragen mit SELECT

Operatoren und Ausdrücke

**Datumsfunktionen**

Einschränkung der Datenausgabe



# 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;
```

<b>SYSDATE</b>
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

# Errechnen von Differenzen bei Datumswerten

- Beispiel: Anzeige des aktuellen Alters der einzelnen Mitarbeiter

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

Results Explain Describe Saved SQL History

 ORA-00923: FROM keyword not found where expected

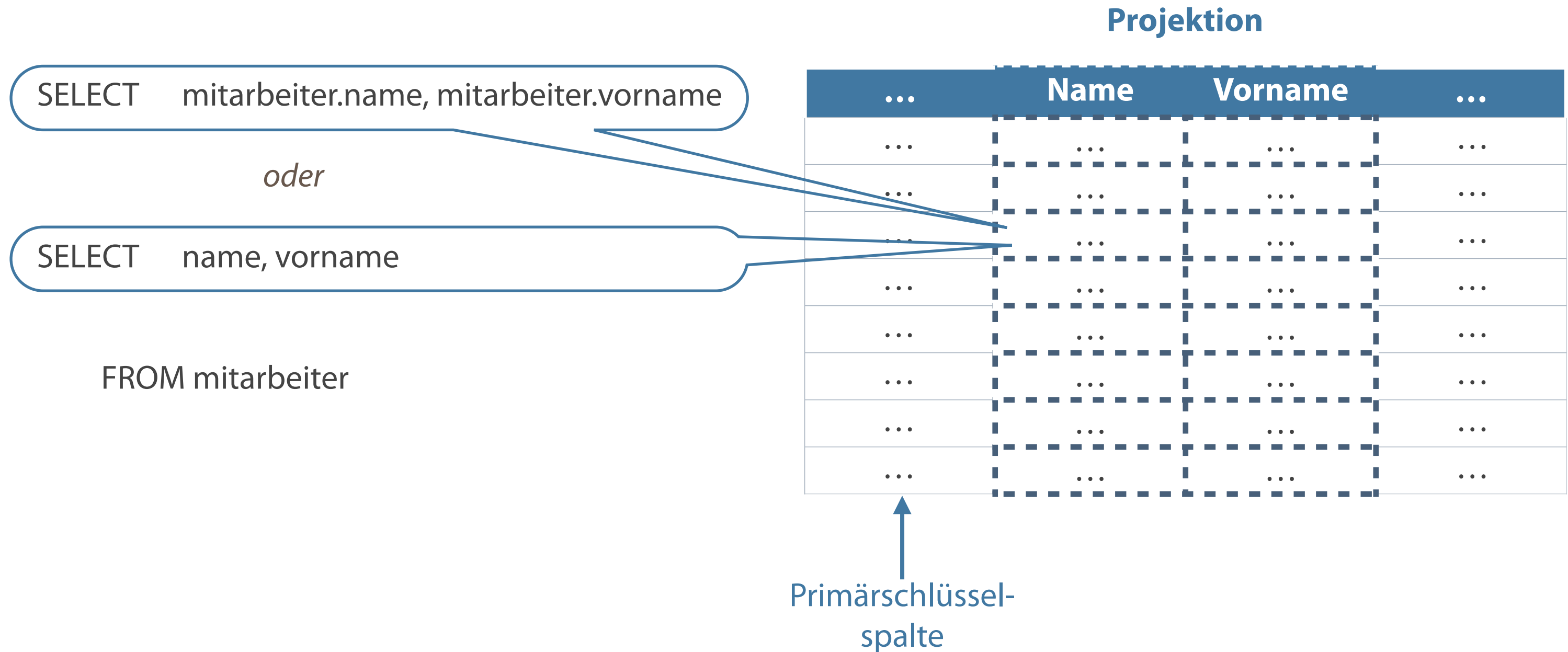
```
SELECT name, vorname, (SYSDATE - geburtstag) / 365 AS "Alter"  
FROM mitarbeiter;
```

NAME	VORNAME	ALTER
Schwarz	Konstantin	59,71777555809233891
Klemm	Ljudmilla	62,38352898274987316
Jörgensen	Jane	41,35887144850329781
Schäfer	Juri	48,33695364028411973
Herrmann	Nadja	52,12873446220192795
...	...	...

**„Alter“ kann nicht als Spaltennamen oder Alias verwendet werden, da es auch als Schlüsselwort identifiziert werden könnte.**



# Zusammenfassung - Projektion



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





Abfragen mit SELECT

Operatoren und Ausdrücke

Datumsfunktionen

**Einschränkung der Datenausgabe**



# Einschränkung von Zeilen durch Auswahl

---

- DISTINCT-Befehl steht vor dem Spaltennamen und blendet redundante Daten aus
- Einschränkung der zurückgegebenen Zeilen über die WHERE-Klausel
- Angabe der WHERE-Klausel grundsätzlich nach der FROM-Klausel

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

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

# 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
...
Elektriker
Elektriker
Elektroniker
Elektroniker
Elektronikerin
Endkontrolleur
Entwicklungsingenieur
Entwicklungsingenieurin
...

POSITION
...
Elektriker
Elektroniker
Elektronikerin
Endkontrolleur
Entwicklungsingenieur
Entwicklungsingenieurin
Geschäftsleiter
Hauptabteilungsleiter
...

*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
...	...
Dreher	260D
Einkäufer	210E
Sekretärin	260D
Abteilungsleiter	510L
Abteilungsleiter	320M
Abteilungsleiter	310T
Schleifer	260K
Buchhalterin	420F
...	...

*Mehrfachausgabe von  
Zeilen mit den selben  
Wertekombinationen wird  
unterdrückt*



# Anwendung der WHERE-Klausel

Bedingung der WHERE-Klausel schränkt Ausgabe ein

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

Beispiel: Nur Auswahl aller männlichen Mitarbeiter

PERS_NR	ANREDE	NAME	POSITION	ABT_NR
102006	Herr	Malossek	Vertriebsbeauftragter	110V
102007	Herr	Karomi	Vertriebsbeauftragter	210V
102009	Herr	Stadelmann	Vertriebsbeauftragter	110V
102010	Herr	Kellner	Geschäftsleiter	10VL
102011	Herr	Peplinski	Vertriebsgruppenleiter	410V
102017	Herr	Hinz	Technischer Mitarbeiter	10VL
102018	Herr	Zengel	Vertriebsbeauftragter	310V
102020	Herr	Kunze	Vertriebsleiter	310V
102021	Herr	Straube	Technischer Mitarbeiter	10VL
...	...	...	...	...

# Zeichenfolgen und Datumsangaben

---

- Unterscheidung bei Zeichenwerten in Groß-/Kleinschreibung

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

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

```
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.**

# 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?

# Literatur

---

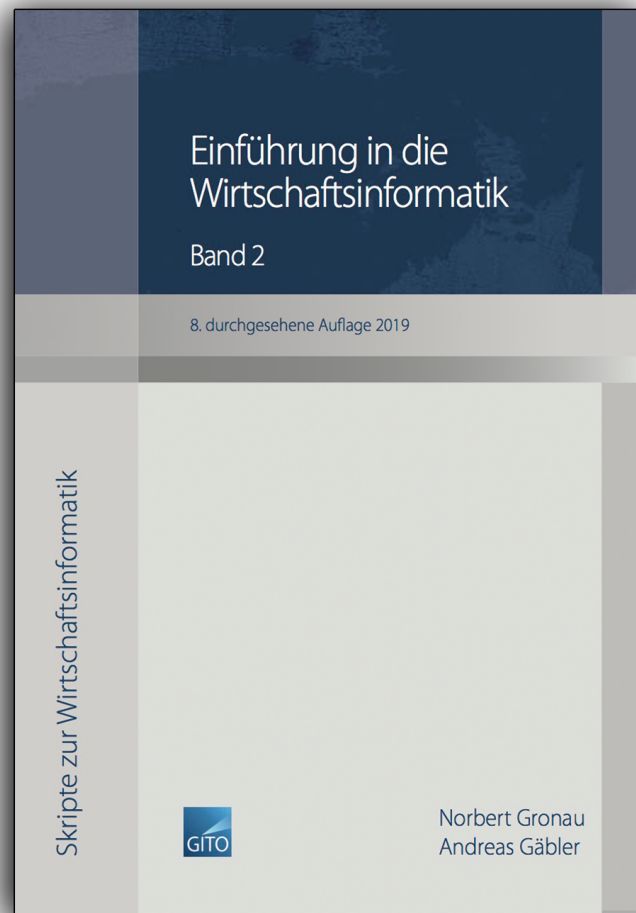
Heuer, A./Saake, G.: Datenbanken - Konzepte und Sprachen; 6. Auflage, 2018, mitp

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](mailto:ngronau@lswi.de)