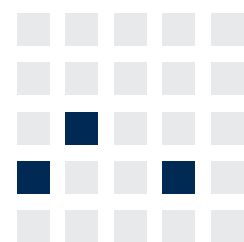


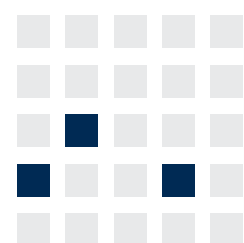
Einführung in die Wirtschaftsinformatik

Teil 9 – Unterabfragen

Wintersemester 2025/2026



**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 **Unterabfragen (Subqueries)** in SQL verwendet und wofür eignen sie sich?
- Was ist der Unterschied zwischen **Single-Row-** und **Multiple-Row-Unterabfragen**?
- Was für **alternative Ausdrücke** können in SQL verwendet werden?



Unterabfragen (Subqueries)

Alternative Ausdrücke

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



<https://quiz.lswi.de/>

pwd: ewinf

Unterabfragen und Lösungsansätze

Problemstellung

- Vergleiche zwischen zwei Werten einer Spalte

Vorgehensweise

1. Erste (innere) Abfrage – Aufruf (Abfrage) des Vergleichswertes
2. Zweite (äußere) Abfrage – Vergleich der Abfragewerte mit dem aus der inneren Abfrage ermittelten Wert (Vergleichswert)
3. Verbindung der beiden Schritte
—> Erste Abfrage in die zweite eingebettet

Äußere Abfrage

Vergleich aller abgefragten
Werte mit Vergleichswert



Innere Abfrage

Erzeugen des Vergleichswertes aus Abfrage

Syntax von Unterabfragen

Problemstellung – Datenauswahl von unbekannten Werten

- Ausführung der Unterabfrage (innere Abfrage), um Wert oder Werteliste zu generieren
- In Hauptabfrage (äußere Abfrage) Erzeugung der eigentlichen Ausgabeliste

SELECT spalte FROM tabelle WHERE ausdruck vergleichsoperator
(SELECT select_list FROM tabelle);

Single-Row Unterabfragen

Beispiel – Rückgabe eines Einzelwertes

- Wer arbeitet in der Abteilung, zu der auch Karl Plenk gehört?

SELECT name, vorname, position
FROM mitarbeiter
WHERE abt_nr =

(SELECT abt_nr
FROM mitarbeiter
WHERE name = 'Plenk'
AND vorname = 'Karl');

NAME	VORNAME	POSITION
Kettler	Gunter	Abteilungsleiter
Klein	Stefan	Konstrukteur
Berg	Christin	Konstrukteurin
Plenk	Karl	Sekretär

Single-Row Unterabfragen – Verfeinerung

Beispiel – modifizierte Filterung

- Wer arbeitet in der Abteilung, zu der auch der Mitarbeiter Plenk gehört?
Die Ergebnisausgabe soll ohne Herrn Plenk erfolgen.

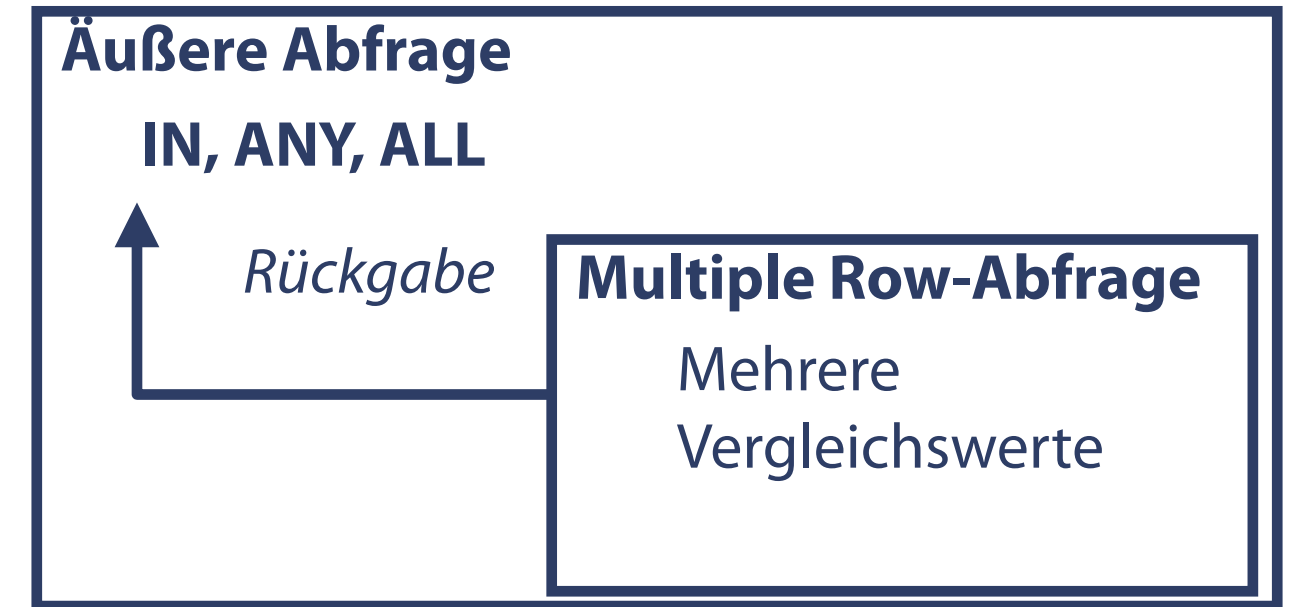
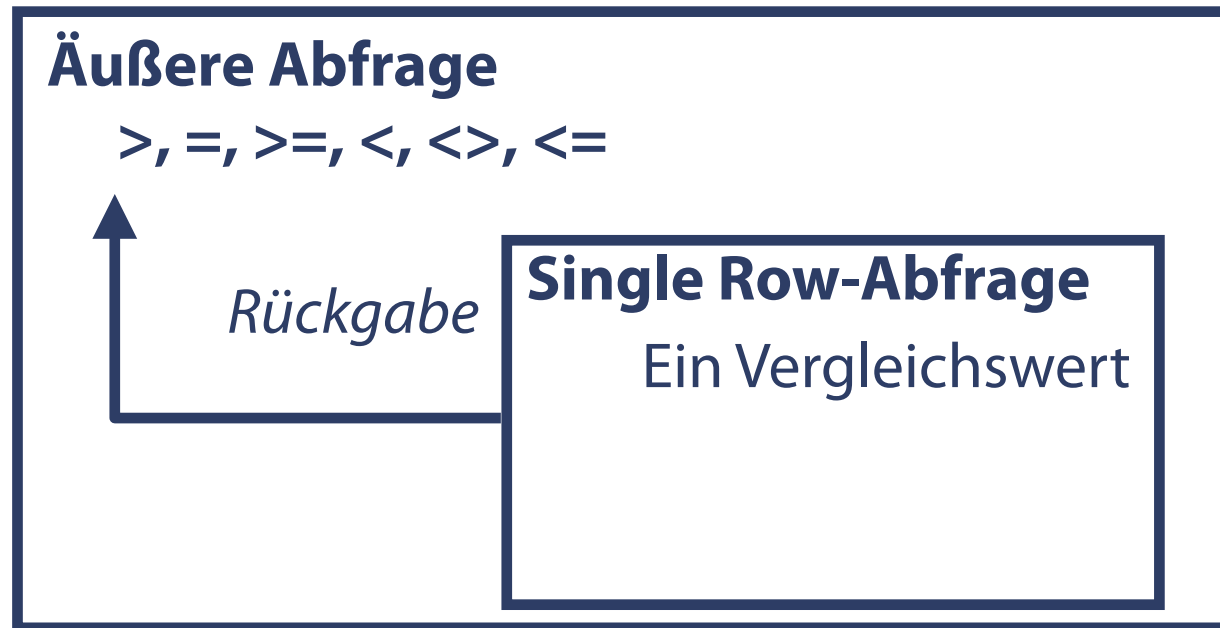
SELECT name, vorname, position
FROM mitarbeiter
WHERE abt_nr =

(SELECT abt_nr
FROM mitarbeiter
WHERE name = 'Plenk'
AND vorname = 'Karl')

AND name <> 'Plenk';

NAME	VORNAME	POSITION
Kettler	Gunter	Abteilungsleiter
Klein	Stefan	Konstrukteur
Berg	Christin	Konstrukteurin

Operatoren in Unterabfragen



Syntax

- Unterabfragen grundsätzlich in Klammern
- Vergleichsoperator vor (links von) Unterabfrage
- Anwendung ORDER BY-Klausel in Unterabfrage bei Realisierung einer Top-N-Analyse (Abfrage mit Ranking)

Vergleichsoperatoren

Single Row-Unterabfragen

größer als	kleiner als	gleich	ungleich	größer oder gleich	kleiner oder gleich
>	<	=	<>	>=	<=

Multiple Row-Unterabfragen

Gleich einem Element aus der Liste	Vergleich mit jedem von Unterabfrage zurückgegebenen Wert	Vergleich mit allen von Unterabfrage zurückgegebenen Werten
IN	ANY	ALL

Multiple-Row Anfragen

Beispiel – Rückgabe mehrerer Werte

- Wie heißen die Mitarbeiter in den Abteilungen, in denen Personen mit dem Namen Grimm arbeiten?

SELECT name, vorname, abt_nr

FROM mitarbeiter

WHERE abt_nr IN

ORDER BY abt_nr, name;

(SELECT abt_nr FROM mitarbeiter

WHERE name = 'Grimm')

NAME	VORNAME	ABT_NR
Adler	Jana	100V
Grimm	Bernd	100V
Hofmann	Katja	100V
Melzer	Thomas	100V
Sonntag	Christof	100V
Walther	Stefanie	100V
Grimm	Alexander	260Z
...

Leere Werte aus Unterabfragen

Problemstellung – Rückgabe von NULL-Werten

- Leere Ergebnisse aus Unterabfragen ergeben leere Werte der äußeren Abfrage

SELECT name, position	
FROM mitarbeiter	
WHERE position =	(SELECT position
	FROM mitarbeiter
	WHERE name = 'Penk') ;

No data found

Kombination mehrerer Single-Row Unterabfragen

SELECT name, position, gehalt

FROM mitarbeiter

WHERE position =

AND gehalt >

;

Einkäufer

2800

(SELECT position

FROM mitarbeiter

WHERE pers_nr = 101003)

(SELECT gehalt

FROM mitarbeiter

WHERE pers_nr = 101107)

NAME	POSITION	GEHALT
Dost	Einkäufer	3100
Petersen	Einkäufer	2890

In der WHERE-Klausel können auch mehrere innere Abfragen nacheinander verwendet werden.

Operator ALL in Unterabfragen

Innere Abfrage

- Ausgabe 19 Zeilen (Gehälter der 19 Sekretärinnen)

Äußere Abfrage – Übernahme aus Vergleich in der WHERE-Klausel

- <ALL – Kleiner als alle Werte (weniger als minimaler Wert – 1780)
- >ALL – Größer als alle Werte (mehr als maximaler Wert – 3600)
- =ALL – Übereinstimmung zwischen Vergleichswert mit allen Rückgabewerten – praktisch nicht möglich

```
SELECT pers_nr, name, position, gehalt
```

```
FROM mitarbeiter
```

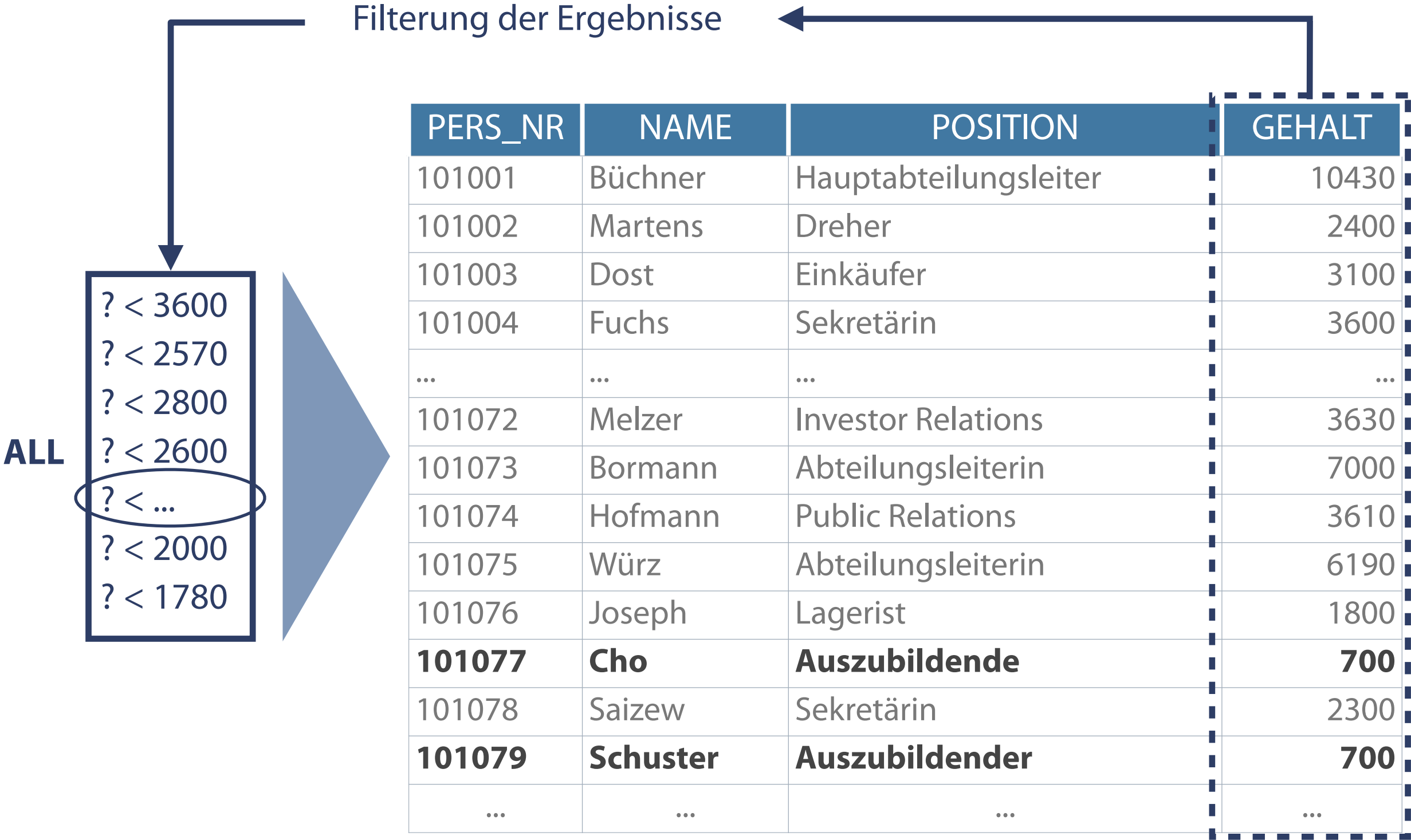
```
WHERE gehalt < ALL
```

(3600, 2570, 2800, 2600, ..., 1780)

```
(SELECT gehalt FROM mitarbeiter  
WHERE position = 'Sekretärin')
```

```
ORDER BY pers_nr;
```

Operator ALL in Unterabfragen – Ergebnistabelle



Operator ANY in Unterabfragen – Beispiel

Innere Abfrage

- Ausgabe 19 Zeilen (Gehälter der 19 Sekretärinnen)

Äußere Abfrage – Übernahme aus Vergleich in WHERE-Klausel

- $< ANY$ – Kleiner als irgendein Wert (weniger als maximaler Wert = 3600)
- $> ANY$ – Größer als irgendein Wert (mehr als minimaler Wert = 1780)
- $= ANY$ – Entspricht einem der 14 Werte (3600, 2570, 2800, 2600, ..., 1780) – analog der Funktion IN

```
SELECT pers_nr, name, position, gehalt
```

```
FROM mitarbeiter
```

```
WHERE gehalt < ANY
```

(3600, 2570, 2800, 2600, ..., 1780)



```
ORDER BY pers_nr;
```

```
(SELECT gehalt
```

```
FROM mitarbeiter
```

```
WHERE position = 'Sekretärin')
```

Operator ANY in Unterabfragen – Ausgangstabellen

Innere Abfrage: Suche nach dem Wert 'Sekretärin'

POSITION	GEHALT
Sekretärin	1780
Sekretärin	1950
Sekretärin	2405
Sekretärin	1780
...	...

ANY – nimm irgendeinen Wert aus rechter Seite

Vergleich Gehalt aus äußerer Abfrage mit jedem Ergebniswert aus innerer Abfrage?
 $\text{gehalt}_{\text{außen}} \leq \text{ANY gehalt}_{\text{innen}}$

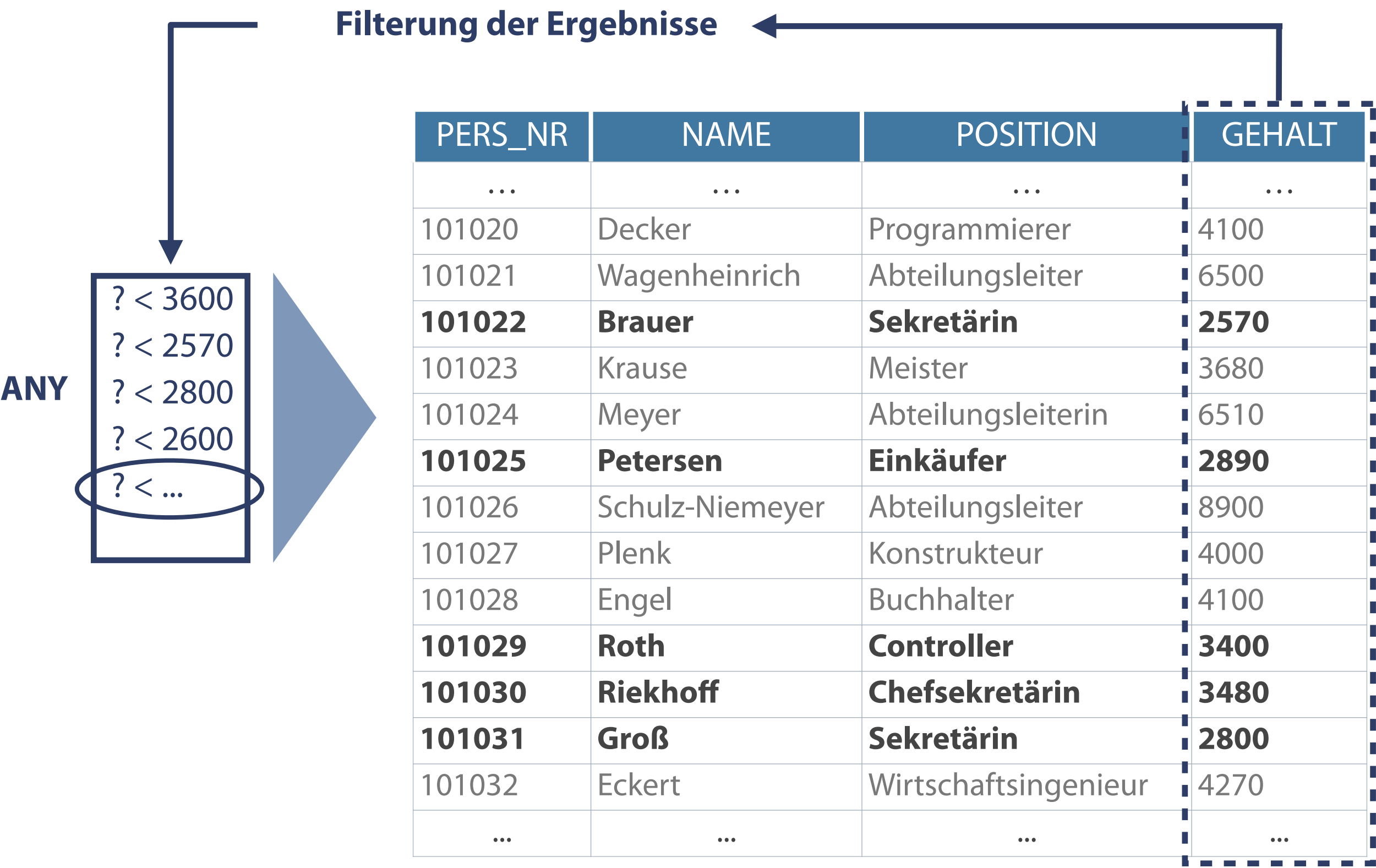
-Vergleich der Werte über ANY-

Äußere Abfrage: Rückgabewerte aus der inneren Abfrage, um Vergleich zu formulieren

PERS_NR	NAME	POSITION	GEHALT	ABT_NR
...
101020	Decker	Programmierer	4100	620W
101021	Wagenheinrich	Abteilungsleiter	6500	420F
101022	Brauer	Sekretärin	2570	210E
101023	Krause	Meister	3680	260F
101024	Meyer	Abteilungsleiterin	6510	540P
101025	Petersen	Einkäufer	2890	210E
101026	Schulz-Niemeyer	Abteilungsleiter	8900	105C
101027	Plenk	Konstrukteur	4000	310T
101028	Engel	Buchhalter	4100	420F
101029	Roth	Controller	3400	105C
101030	Riekhoff	Chefsekretärin	3480	
101031	Groß	Sekretärin	2800	230P
101032	Eckert	Wirtschaftsingenieur	4270	230P
...

WHERE-Klausel – Abfrage Werte Gehalt

Operator ANY in Unterabfragen – Ergebnistabelle



NULL-Werte in einer Unterabfrage – Ungeeignete Multiple Row-Funktion

Problem

- Abgefragte Spalte in Unterabfrage enthält mindestens einen NULL-Wert -->
- Hauptabfrage kann keine Ergebnistabelle erzeugen

Bedingungen, die einen NULL-Wert vergleichen...

- ...liefern einen NULL-Wert zurück
- Entspricht der Wirkung von "<> ALL"

```
SELECT name  
FROM mitarbeiter  
WHERE pers_nr NOT IN
```

```
(SELECT leiter  
FROM mitarbeiter);
```

No data found

NULL-Werte in einer Unterabfrage – Alternative

Bedingung – Wirkung verschiedener Operatoren

- Bei Erwartung der Rückgabe von NULL-Werten keine Verwendung von NOT IN
- Alternatives Vorgehen: Einsatz des Operators IN
- Wirkweise von IN entspricht “=ANY”

SELECT name
FROM mitarbeiter
WHERE pers_nr IN

(SELECT leiter
FROM mitarbeiter);

NAME
Köhler
Ernst
Klemm
Krajcsir
Michalke
...

NULL-Werte in einer Unterabfrage – Lösung

Bedingung

- Verhinderung der Rückgabe von NULL-Werten für Einsatz des Operators NOT IN

Lösung

- WHERE-Klausel in Unterabfrage
- Wirkweise von IS NOT NULL – Filtern aller nicht leeren Ergebnisse

Beispiel Aufgabenstellung

- Ausgabe aller Positionen (Berufe), die keine Leitungsfunktion besitzen

```
SELECT name, position  
FROM mitarbeiter  
WHERE pers_nr NOT IN
```

```
(SELECT leiter FROM mitarbeiter  
WHERE leiter IS NOT NULL);
```



Unterabfragen (Subqueries)
Alternative Ausdrücke

Hörsaal-Quiz - Recap erste Hälfte

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



<https://quiz.lswi.de/>

Pwd: ewinf

Bedingte Ausdrücke

CASE-Ausdrücke

```
CASE ausdruck WHEN comp_ausdruck1 THEN return_ausdruck_1
              [WHEN comp_ausdruck_2 THEN return_ausdruck_2
               WHEN comp_ausdruck_n THEN return_ausdruck_n
               ELSE else_ausdruck]
END
```

DECODE

```
DECODE (col|ausdruck, such1, ergebnis_1
        [, such2, ergebnis_2,...,]
        [, default]);
```

...stellen eine IF-THEN-ELSE-Logik innerhalb der SQL-Anweisung bereit.

Anwendung von CASE-Ausdrücken

- Bedingte Abfragen in der Form von IF-THEN-ELSE-Anweisung

```
SELECT name, position, gehalt,
       CASE position WHEN 'Konstrukteur' THEN gehalt*1.10
                    WHEN 'Konstrukteurin' THEN gehalt*1.10
                    WHEN 'Monteur' THEN gehalt*1.15
                    WHEN 'Meister' THEN gehalt*1.05
       ELSE gehalt END Gehaltserhöhung
FROM mitarbeiter ORDER BY position;
```

NAME	POSITION	GEHALT	GEHALTSERHÖHUNG
...
Klein	Konstrukteur	4100	4510
Adam	Konstrukteurin	4700	5170
...
Ascheid	Meister	3990	4189,5
Krause	Meister	3680	3864
...
Thal	Monteur	2400	2760
Möhlmann	Monteur	2740	3151
...

Hinweis: CASE-Klausel schließt mit END-Ausdruck.
WHEN-Klausel bezieht sich nur auf den Vergleich mit jeweils einem konkreten Wert.

Anwendung der Funktion DECODE

- Wirkung der Anweisung ähnlich dem CASE-Ausdruck

```
SELECT name, position, gehalt,  
       DECODE(position, 'Konstrukteur', gehalt*1.10,  
                    'Konstrukteurin', gehalt*1.10,  
                    'Monteur', gehalt*1.15,  
                    'Meister', gehalt*1.05, gehalt) "Neues  
       Gehalt" FROM mitarbeiter ORDER BY position;
```

NAME	POSITION	GEHALT	NEUES GEHALT
...
Klein	Konstrukteur	4100	4510
Adam	Konstrukteurin	4100	4510
...
Ascheid	Meister	3990	4189,5
Krause	Meister	3680	3864
...
Thal	Monteur	2400	2760
Möhlmann	Monteur	2740	3151
...

Hinweis: DECODE-Klausel schließt mit default-Wert ab. Wird dieser weggelassen, dann wird bei fehlender Übereinstimmung in allen Argumenten NULL-Wert ausgegeben

Abfragereihen mit der Funktion DECODE

- Anzeige des Steuersatzes aller Angestellten der Abteilung 310T

```
SELECT name, position, abt_nr, gehalt,  
       DECODE (TRUNC (gehalt/1000, 0),  
               0, '00%',  
               1, '20%',  
               2, '25%',  
               3, '30%',  
               4, '35%',  
               5, '40%',  
               '45%') AS "Steuersatz"  
FROM mitarbeiter WHERE abt_nr = '310T';
```

TRUNC(x,y) schneidet von einem Wert **x** Nachkommastellen ab, sodass nur noch **y** Nachkommastellen verbleiben

NAME	POSITION	ABT_NR	GEHALT	STEUERSATZ
Berg	Sekretärin	310T	2250	25%
Kettler	Abteilungsleiter	310T	8080	45%
Klein	Konstrukteur	310T	4100	35%
Plenk	Konstrukteur	310T	4000	35%

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



<https://quiz.lswi.de/>

pwd: ewinf

Kontrollfragen

- Welche Aufgabe hat eine Unterabfrage?
- Worin besteht der Unterschied zwischen Single Row- und Multiple Row-Abfragen?
- Wann werden CASE-Ausdrücke in SQL-Anweisungen verwendet?

Literatur

Kemper, A./Eickler, A.: Datenbanksysteme; 10. Auflage, 2015, Oldenbourg Verlag

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

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

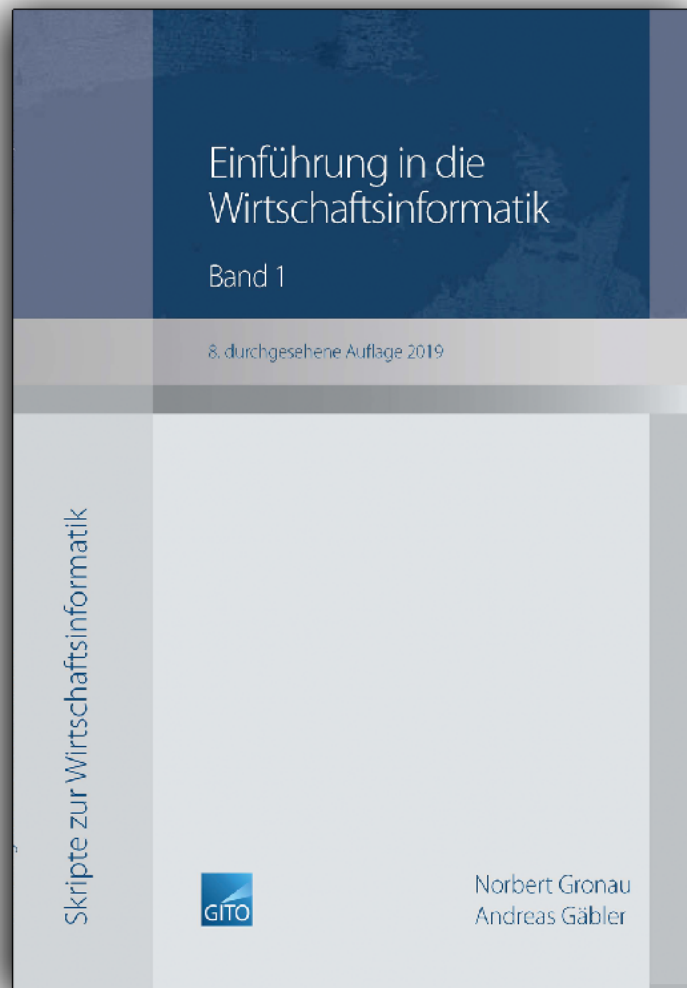
Kähler, W.-M.: SQL mit Oracle; 3. Auflage, 2008, ViewegHeuer, A./Saake, G.: Datenbanken, Konzepte und Sprachen; 2. Auflage, 1995, Thomson

Vossen, G.: Datenbankmodelle, Datenbanksprachen und Datenbankmanagementsysteme; 4. Aufl. - Oldenbourg Verlag München 2000

Mertens P. et. al: Grundzüge der Wirtschaftsinformatik; 9. Auflage; 2005, Springer Verlag

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 978-3-95545-233-9

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