



# Subqueries



## SELECT-Anweisung

```
select-exp ::= SELECT [ALL | DISTINCT] select-item-commalist  
            FROM table-ref-commalist  
            [WHERE cond-exp]  
            [GROUP BY column-ref-commalist]  
            [HAVING cond-exp]
```

```
SELECT  <SELECT_LISTE>
        FROM  <TABELLE>
        WHERE <EXPR> <OPERATOR>
                (SELECT <SELECT_LISTE>
                  FROM <TABELLE>
                  WHERE... )
;
```

- <expr> ist ein Ausdruck oder im einfachsten Fall ein Attribut
- <operator> ist ein Vergleichsoperator wie z.B. =, >, oder IN
- Die Subquery (innere Query) wird zuerst ausgewertet. Das Ergebnis der Auswertung wird anschließend von der äußeren Query verwendet.
- Die Subquery muss immer in Klammern eingeschlossen werden.
- Die Klausel ORDER BY kann in einer Subquery nicht verwendet werden. In einer zusammengesetzten SELECT-Anweisung darf nur eine einzige ORDER BY Klausel verwendet werden und diese muss die letzte Klausel in der äußersten SELECT-Anweisung sein.

## Subquery-Typen

- Single-Row Subqueries
- Multiple-Row Subqueries
- Multiple-Column Subqueries
- Correlated Subqueries
- Subqueries mit EXISTS

# Single Row Subquery

Operator	Meaning
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to

```
SELECT NACHNAME
FROM MITARBEITER
WHERE FUNKTION =
    ( SELECT FUNKTION
      FROM MITARBEITER
      WHERE UPPER(NACHNAME) = 'WEISSNIX' );
```

**Alle Mitarbeiter, die zu einer  
Abteilung gehören,  
die etwas mit  
"Programmieren"  
zu tun hat?**

# Lösung I

**Alle Mitarbeiter, die zu einer Abteilung gehören, die etwas mit "Programmieren" zu tun hat?**

```
select m.personalnummer, m.nachname,    m.abteilungsnummer
from mitarbeiter m
where m.abteilungsnummer in
        (select a.abteilungsnummer
         from abteilung a
         where upper(a.abteilungsname)
               like '%PROGRAMMIER%')
order by m.personalnummer;
```

# Lösung II

Alle Mitarbeiter, die zu einer Abteilung gehören, die etwas mit "Programmieren" zu tun hat?

```
select m.personalnummer, m.nachname,  
m.abteilungsnummer  
from mitarbeiter m  
join abteilung a  
on m.abteilungsnummer = a.abteilungsnummer  
where upper(a.abteilungsname) like  
                                '%PROGRAMMIER%'  
order by m.personalnummer;
```



# Multiple Row Subquery

- **Geben mehr als eine Tabellenzeile zurück**
- **Benutzen „multiple-row“ Vergleichsoperatoren**

Operator	Bedeutung
IN	Ein Element der Unterabfrage muss <b>GLEICH (=)</b> dem geprüften Wert sein
ANY	Irgendein Element der Unterabfrage muss den Vergleich erfüllen (<, >)
ALL	Alle Elemente der Unterabfrage müssen den Vergleich erfüllen (<, >)

# Multiple Row Subquery

```
SELECT  NACHNAME, ABTEILUNGSNUMMER, GEHALT
FROM MITARBEITER
WHERE GEHALT > ANY
      (SELECT GEHALT
        FROM MITARBEITER
         WHERE ABTEILUNGSNUMMER = 3
           AND GEHALT > 4000);
```

## Multiple Column Subquery

```
SELECT NACHNAME, GEHALT, EINTRITTS_DATUM
FROM MITARBEITER
WHERE (GEHALT, EINTRITTS_DATUM) IN
SELECT GEHALT, EINTRITTS_DATUM
FROM MITARBEITER
WHERE ABTEILUNGSNUMMER = 3)
AND ABTEILUNGSNUMMER <> 3;
```

## Aufgabe

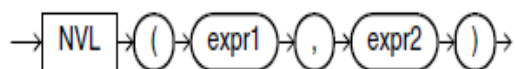
Gib das Ergebnis folgender SQL-Abfrage an:

```
SELECT M.NACHNAME
FROM   MITARBEITER M
WHERE  M.PERSONALNUMMER NOT IN
      (SELECT CHEF.VORGESETZTER
       FROM MITARBEITER CHEF);
```

# Oracle Builtin Function

## NVL

### Syntax



### Purpose

NVL lets you replace null (returned as a blank) with a string in the results of a query. If *expr1* is null, then NVL returns *expr2*. If *expr1* is not null, then NVL returns *expr1*.

```
SELECT last_name, NVL(TO_CHAR(commission_pct), 'Not Applicable')  
       "COMMISSION" FROM employees  
WHERE last_name LIKE 'B%'  
ORDER BY last_name;
```

LAST_NAME	COMMISSION
Baer	Not Applicable
Baida	Not Applicable
Banda	.1
Bates	.15
Bell	Not Applicable
Bernstein	.25

## Subqueries an Stelle einer Spalte

```
Select gehalt-  
    (select avg(gehalt)  
     from mitarbeiter)  
from mitarbeiter;
```

# Aufgaben

1. Gegeben seien die Tabellen  
STUDENT und DOZENT

Gib die Betreuungsrelation an!

2. Um welchen Betrag unterscheidet sich das GEHALT der weiblichen Mitarbeiter im Mittel vom GEHALT der männlichen Mitarbeiter?

# Correlated Subquery

```
SELECT NACHNAME, ABTEILUNGSNUMMER,  
GEHALT  
FROM MITARBEITER A  
WHERE GEHALT >  
      (SELECT AVG(GEHALT)  
       FROM MITARBEITER B  
       WHERE B.ABTEILUNGSNUMMER =  
            A.ABTEILUNGSNUMMER) ;
```



## Subquery mit EXISTS

```
SELECT NACHNAME
FROM MITARBEITER A
WHERE EXISTS
    ( SELECT *
      FROM MITARBEITER B
      WHERE B.VORGESETZTER =
            A.PERSONALNUMMER ) ;
```

# Top-N Queries

# Microsoft SQL Server

```
SELECT TOP 10 column  
FROM table
```

# PostgreSQL and MySQL

```
SELECT column FROM table  
LIMIT 10
```

# Oracle

```
SELECT column FROM table  
WHERE ROWNUM <= 10
```

**Falls ROWNUM in  
Kombination mit einer  
ORDER BY Klausel  
eingesetzt wird, sortiert diese  
das Ergebnis um!**

**ORDER BY wird immer  
ZULETZT vor der Anzeige  
ausgeführt!**

```
SELECT ROWNUM, PERSONALNUMMER, NACHNAME
FROM MITARBEITER
ORDER BY NACHNAME;
```

ROWNUM	PERSONALNUMMER	NACHNAME
12	999	Besserwisser
2	2	Buchhalter
3	3	Fehler
1	1	Geldberg
6	6	Gründlich
4	4	Hacker
10	802	Kannix
7	7	Klugscher
8	8	Lueger
5	5	Nieda
9	801	Tutnix
11	803	Weissnix

```
SELECT  ROWNUM, T.* FROM
(SELECT PERSONALNUMMER, GEHALT
      FROM MITARBEITER
      ORDER BY GEHALT DESC) T
WHERE ROWNUM < 4
```

ROWNUM	PERSONALNUMMER	GEHALT
-----	-----	-----
1	1	6000
2	999	5000
3	8	4500

# NULLWerte in einer Subquery

## Anfrage:

“Alle Mitarbeiter, die nicht Vorgesetzte anderer Mitarbeiter sind“

```
SELECT M.NACHNAME  
FROM   MITARBEITER M  
WHERE  M.PERSONALNUMMER NOT IN  
       (SELECT CHEF.VORGESETZTER  
        FROM MITARBEITER CHEF)
```

Liefert:

0 rows selected ???



# Subquery abschließendes Beispiel

Voraussetzungen:

- Tabelle Mitarbeiter wurde um eine Spalte GEHALTSKLASSE erweitert
- eine Tabelle KLASS\_TABELLE (UG, OG, Klasse) wurde eingerichtet.

```
UPDATE MITARBEITER_MIT_GK M1
SET GEHALTSKLASSE =
    (SELECT K.KLASSE
     FROM KLASS_TABELLE K, MITARBEITER_MIT_GK M
     WHERE M.GEHALT BETWEEN K.UG AND K.OG
     AND M.PERSONALNUMMER = M1.PERSONALNUMMER
    )
```



## Aufgabe

**Welcher Mitarbeiter (Personalnummer, Nachname)  
hatte die meisten Kundenkontakte?**