

WIEDERHOLUNG

```
SELECT [ALL|DISTINCT] ausdruck
FROM {tabellenname|viewname}[aliasname]
    [{tabellenname|viewname}[ aliasname], ...]
[WHERE suchbedingungen]
[GROUP BY nicht aggregierender ausdruck
    [,nicht aggregierender ausdruck]...]
[HAVING suchbedingungen]
[ORDER BY {spaltenname | nummer in spaltenliste} [{ASC|DESC}]
    [, {spaltenname|nummer in spaltenliste} [{ASC|DESC}]]...]
```

SELECT

```
SELECT DISTINCT Name  
FROM Person
```

```
SELECT Buch.ISBN  
FROM Buch, Bestellung
```

```
SELECT Buch.ISBN AS "ISBN-Nummer"  
FROM Buch, Bestellung
```

FROM

```
SELECT *  
FROM Buch, Prof
```

```
SELECT *  
FROM Buch CROSS JOIN Prof
```

```
SELECT *  
FROM Buch quelle, Buch ziel
```

WHERE

- **EINFACHE OPERATOREN: =, <=, >=, <>, <, >**
- **BEDINGUNGEN MIT AND, OR, NOT**
- **IN - OPERATOR**
- **BETWEEN - OPERATOR**
- **LIKE - OPERATOR**
- **NULL - OPERATOR**

LOS GEHT'S

\o/

VERBUND (JOIN)

Autor	
Nr	Name
1201	Goethe
1202	Shakespeare
1203	Kafka

Buch			
Id	Preis	Titel	AutorId
302	13.99	Hamlet	1202
306	14.99	Faust	1201
310	44.99	Die Verwandlung	1203

FRAGE: WIE HEIßT DER AUTOR, DER 'HAMLET' GESCHRIEBEN HAT?

ERSTER SCHRITT:

```
SELECT *  
FROM Autor, Buch
```

Nr	Name	Id	Preis	Titel	AutorId
1201	Goethe	302	13.99	Hamlet	1202
1201	Goethe	306	14.99	Faust	1201
1201	Goethe	310	44.99	Die Verwandlung	1203
1202	Shakespeare	302	13.99	Hamlet	1202
1202	Shakespeare	306	14.99	Faust	1201
1202	Shakespeare	310	44.99	Die Verwandlung	1203
1203	Kafka	302	13.99	Hamlet	1202
1203	Kafka	306	14.99	Faust	1201
1203	Kafka	310	44.99	Die Verwandlung	1203

ZWEITER SCHRITT:

```
SELECT *  
FROM Autor, Buch  
WHERE Buch.Titel="Hamlet"
```


Nr	Name	Id	Preis	Titel	AutorId
1201	Goethe	302	13.99	Hamlet	1202
1202	Shakespeare	302	13.99	Hamlet	1202
1203	Kafka	302	13.99	Hamlet	1202

DRITTER SCHRITT:

```
SELECT *  
FROM Autor, Buch  
WHERE Buch.Titel="Hamlet"  
      AND Autor.Nr=Buch.AutorId
```

Nr	Name	Id	Preis	Titel	AutorId
1202	Shakespeare	302	13.99	Hamlet	1202

VIERTER SCHRITT:

```
SELECT Name  
FROM Autor, Buch  
WHERE Buch.Titel="Hamlet"  
      AND Autor.Nr=Buch.AutorId
```

	Name
	Shakespeare

CROSS JOIN

```
SELECT *  
FROM PERSONAL, ABTEILUNG;
```

```
SELECT *  
FROM PERSONAL CROSS JOIN ABTEILUNG;
```


(INNER) JOIN

Equi-Join

```
SELECT *  
FROM PERSONAL, ABTEILUNG  
WHERE PERSONAL.ABT_NR=ABTEILUNG.ABT_NR;
```

```
SELECT *  
FROM PERSONAL JOIN ABTEILUNG ON PERSONAL.ABT_NR=ABTEILUNG.ABT_NR;
```

```
SQL> SELECT * FROM PERSONAL JOIN ABTEILUNG ON PERSONAL.ABT_NR=ABTEILUNG.ABT_NR;
```

PNR	VORNAME	NACHNAME	GEH_S	ABT_N	KRA	ABT_N	NAME
123	Ada	Lovelace	it5	abt13	dek	abt13	Produktion
124	Leonhard	Euler	it2	abt11	aok	abt11	Verwaltung
126	Ren??	Descartes	it1	abt12	dek	abt12	Projektierung
127	Alan	Turing	it2	abt14	dek	abt14	Lagerung
132	Grace	Hopper	it4	abt15	dek	abt15	Verkauf
145	Marie	Curie	it4	abt15	dek	abt15	Verkauf
147	Hedy	Lamarr	it5	abt11	dek	abt11	Verwaltung
133	Radia	Perlman	it3	abt12	dek	abt12	Projektierung
161	Martin	Fowler	it2	abt13	dek	abt13	Produktion
163	Erich	Gamma	it1	abt14	dek	abt14	Lagerung

10 rows selected.

```
SQL> SELECT * FROM PERSONAL, ABTEILUNG WHERE PERSONAL.ABT_NR=ABTEILUNG.ABT_NR;
```

PNR	VORNAME	NACHNAME	GEH_S	ABT_N	KRA	ABT_N	NAME
123	Ada	Lovelace	it5	abt13	dek	abt13	Produktion
124	Leonhard	Euler	it2	abt11	aok	abt11	Verwaltung
126	Ren??	Descartes	it1	abt12	dek	abt12	Projektierung
127	Alan	Turing	it2	abt14	dek	abt14	Lagerung
132	Grace	Hopper	it4	abt15	dek	abt15	Verkauf
145	Marie	Curie	it4	abt15	dek	abt15	Verkauf
147	Hedy	Lamarr	it5	abt11	dek	abt11	Verwaltung
133	Radia	Perlman	it3	abt12	dek	abt12	Projektierung
161	Martin	Fowler	it2	abt13	dek	abt13	Produktion
163	Erich	Gamma	it1	abt14	dek	abt14	Lagerung

10 rows selected.

NATURAL JOIN

```
SELECT *  
FROM PERSONAL NATURAL JOIN ABTEILUNG;
```

```
SQL> SELECT * FROM PERSONAL NATURAL JOIN ABTEILUNG;
```

ABT_N	PNR	VORNAME	NACHNAME	GEH_S	KRA	NAME
ab13	123	Ada	Lovelace	it5	dek	Produktion
ab11	124	Leonhard	Euler	it2	aok	Verwaltung
ab12	126	Ren??	Descartes	it1	dek	Projektierung
ab14	127	Alan	Turing	it2	dek	Lagerung
ab15	132	Grace	Hopper	it4	dek	Verkauf
ab15	145	Marie	Curie	it4	dek	Verkauf
ab11	147	Hedy	Lamarr	it5	dek	Verwaltung
ab12	133	Radia	Perlman	it3	dek	Projektierung
ab13	161	Martin	Fowler	it2	dek	Produktion
ab14	163	Erich	Gamma	it1	dek	Lagerung

10 rows selected.

OUTER JOIN

LEFT OUTER JOIN

```
SELECT *  
FROM PERSONAL LEFT OUTER JOIN ABTEILUNG ON PERSONAL.ABT_NR=ABTEILUNG.ABT_NR;
```

```
SQL> SELECT * FROM PERSONAL LEFT OUTER JOIN ABTEILUNG ON PERSONAL.ABT_NR=ABTEILUNG.ABT_NR;
```

PNR	VORNAME	NACHNAME	GEH_S	ABT_N	KRA	ABT_N	NAME
147	Hedy	Lamarr	it5	abt11	dek	abt11	Verwaltung
124	Leonhard	Euler	it2	abt11	aok	abt11	Verwaltung
133	Radia	Perlman	it3	abt12	dek	abt12	Projektierung
126	Ren??	Descartes	it1	abt12	dek	abt12	Projektierung
161	Martin	Fowler	it2	abt13	dek	abt13	Produktion
123	Ada	Lovelace	it5	abt13	dek	abt13	Produktion
163	Erich	Gamma	it1	abt14	dek	abt14	Lagerung
127	Alan	Turing	it2	abt14	dek	abt14	Lagerung
145	Marie	Curie	it4	abt15	dek	abt15	Verkauf
132	Grace	Hopper	it4	abt15	dek	abt15	Verkauf
173	Richard	Helm	it1		dek		

11 rows selected.

RIGHT OUTER JOIN

```
SELECT *  
FROM PERSONAL RIGHT OUTER JOIN ABTEILUNG ON PERSONAL.ABT_NR=ABTEILUNG.ABT_NR;
```

```
SQL> SELECT * FROM PERSONAL RIGHT OUTER JOIN ABTEILUNG ON PERSONAL.ABT_NR=ABTEILUNG.ABT_NR;
```

PNR	VORNAME	NACHNAME	GEH_S	ABT_N	KRA	ABT_N	NAME
123	Ada	Lovelace	it5	abt13	dek	abt13	Produktion
124	Leonhard	Euler	it2	abt11	aok	abt11	Verwaltung
126	Ren??	Descartes	it1	abt12	dek	abt12	Projektierung
127	Alan	Turing	it2	abt14	dek	abt14	Lagerung
132	Grace	Hopper	it4	abt15	dek	abt15	Verkauf
145	Marie	Curie	it4	abt15	dek	abt15	Verkauf
147	Hedy	Lamarr	it5	abt11	dek	abt11	Verwaltung
133	Radia	Perlman	it3	abt12	dek	abt12	Projektierung
161	Martin	Fowler	it2	abt13	dek	abt13	Produktion
163	Erich	Gamma	it1	abt14	dek	abt14	Lagerung
						abt16	Forschung

11 rows selected.

FULL OUTER JOIN

```
SELECT *  
FROM PERSONAL FULL OUTER JOIN ABTEILUNG ON PERSONAL.ABT_NR=ABTEILUNG.ABT_NR;
```

```
SQL> SELECT * FROM PERSONAL FULL OUTER JOIN ABTEILUNG ON PERSONAL.ABT_NR=ABTEILUNG.ABT_NR;
```

PNR	VORNAME	NACHNAME	GEH_S	ABT_N	KRA	ABT_N	NAME
123	Ada	Lovelace	it5	abt13	dek	abt13	Produktion
124	Leonhard	Euler	it2	abt11	aok	abt11	Verwaltung
126	Ren??	Descartes	it1	abt12	dek	abt12	Projektierung
127	Alan	Turing	it2	abt14	dek	abt14	Lagerung
132	Grace	Hopper	it4	abt15	dek	abt15	Verkauf
145	Marie	Curie	it4	abt15	dek	abt15	Verkauf
147	Hedy	Lamarr	it5	abt11	dek	abt11	Verwaltung
133	Radia	Perlman	it3	abt12	dek	abt12	Projektierung
161	Martin	Fowler	it2	abt13	dek	abt13	Produktion
163	Erich	Gamma	it1	abt14	dek	abt14	Lagerung
173	Richard	Helm	it1		dek		
						abt16	Forschung

12 rows selected.

SELF JOIN

```
SELECT *  
FROM PERSONAL p1 INNER JOIN PERSONAL p2 ON p1.ABT_NR=p2.ABT_NR  
WHERE p1.PNR < p2.PNR;
```



```
SQL> SELECT * FROM PERSONAL p1 INNER JOIN PERSONAL p2 ON p1.ABT_NR=p2.ABT_NR WHERE p1.PNR < p2.PNR;
```

PNR	VORNAME	NACHNAME	GEH_S	ABT_N	KRA	PNR	VORNAME	NACHNAME	GEH_S	ABT_N	KRA
132	Grace	Hopper	it4	abt15	dek	145	Marie	Curie	it4	abt15	dek
124	Leonhard	Euler	it2	abt11	aok	147	Hedy	Lamarr	it5	abt11	dek
126	Ren??	Descartes	it1	abt12	dek	133	Radia	Perlman	it3	abt12	dek
123	Ada	Lovelace	it5	abt13	dek	161	Martin	Fowler	it2	abt13	dek
127	Alan	Turing	it2	abt14	dek	163	Erich	Gamma	it1	abt14	dek

AUFGABE

**PROBIEREN SIE DIE VERSCHIEDENEN `join` ARTEN MIT
VERSCHIEDENEN TABELLEN AUS.**

- 1. SCHREIBEN SIE AUF, WAS SIE FESTSTELLEN.**
- 2. MIT WELCHEN TABELLENKOMBINATIONEN UND SPALTEN GEHT ES
NICHT?**
- 3. AUF WAS MUSS MAN ACHTEN, WENN MAN MEHR ALS ZWEI
TABELLEN NIMMT?**
- 4. KONSTRUIEREN SIE DAZU EIN BEISPIEL.**

AGGREGATSFUNKTIONEN

COUNT

```
SELECT COUNT(*) AS "PERSONALANZAHL"  
FROM PERSONAL;
```

```
SELECT COUNT(*) AS "Anzahl", ABT_NR  
FROM PERSONAL  
GROUP BY ABT_NR  
HAVING COUNT(*) > 1  
ORDER BY 1 DESC;
```

```
SQL> SELECT COUNT(*) AS "PERSONALANZAHL" FROM PERSONAL;
```

```
PERSONALANZAHL
```

```
-----
```

```
12
```

```
SQL> SELECT COUNT(*) AS "Anzahl", ABT_NR FROM PERSONAL GROUP BY ABT_NR HAVING COUNT(*) > 1 ORDER BY 1 DESC;
```

```
Anzahl ABT_N
```

```
-----
```

```
3 abt13
```

```
2 abt12
```

```
2 abt14
```

```
2 abt15
```

```
2 abt11
```

AUFGABEN

- 1. WELCHER MITARBEITER HAT DIE MEISTEN KINDER?**
- 2. WIEVIELE MITARBEITER HABEN SCHON MAL EINE PRÄMIE BEKOMMEN?**

MIN – UND MAX


```
SELECT MAX(P_BETRAG)
FROM PRAEMIE;
```

```
SELECT ABT_NR
FROM PERSONAL
GROUP BY ABT_NR
HAVING MAX (PNR) > 160;
```

```
SQL> SELECT MAX(P_BETRAG) FROM PRAEMIE;
```

```
MAX(P_BETRAG)
```

```
-----
```

```
2300
```

```
SQL> SELECT ABT_NR FROM PERSONAL GROUP BY ABT_NR HAVING MAX (PNR) > 160;
```

```
ABT_N
```

```
-----
```

```
abt13
```

```
abt14
```

AUFGABEN

- 1. WELCHE MACHINE HAT DEN HOCHSTEN NEUWERT?**
- 2. WAS WAR DIE NIEDRIGSTE PRÄMIE?**

SUM

```
SELECT SUM(P_BETRAG)  
FROM PRAEMIE;
```

```
SELECT PNR  
FROM PRAEMIE  
GROUP BY PNR  
HAVING SUM(P_BETRAG) > 1000;
```

```
SQL> SELECT SUM(P_BETRAG) FROM PRAEMIE;
```

```
SUM(P_BETRAG)
```

```
-----
```

```
6700
```

```
SQL> SELECT PNR FROM PRAEMIE GROUP BY PNR HAVING SUM(P_BETRAG) > 1000;
```

```
PNR
```

```
-----
```

```
123
```

AUFGABEN

- 1. WIE VIEL SIND ALLE MASCHINEN ZUM AKTUELLEN ZEITPUNKT WERT?**
- 2. WER ARBEITET AN MASCHINEN, DIE ZUSAMMEN EINEN NEUWERT VON MEHR ALS 60.000 EUR HABEN?**

AVG


```
SELECT PNR, AVG (P_BETRAG)
FROM PRAEMIE
GROUP BY PNR
HAVING AVG (P_BETRAG) > 500;
```

```
SQL> SELECT PNR, AVG (P_BETRAG) FROM PRAEMIE GROUP BY PNR HAVING AVG (P_BETRAG) > 300;
```

PNR	AVG(P_BETRAG)
123	1500
124	350
132	1000
133	600
145	500
147	400
161	500

7 rows selected.

AUFGABEN

- 1. WER ARBEITET AN MASCHINEN, DEREN ZEITWERT WENIGER ALS 20.000 EUR WERT IST?**
- 2. WAS IST DER DURCHSCHNITTLLICHE ANSCHAFFUNGSWERT EINER MASCHINE?**

FUNKTIONEN IN DER SELECT-KOMPONENTE

```
SELECT GEH_STUFE, BETRAG * 1.06 AS "Gehaltserhoehung"  
FROM GEHALT;
```

```
SQL> SELECT GEH_STUFE, BETRAG * 1.06 AS "Gehaltserhoehung" FROM GEHALT;
```

```
GEH_S Gehaltserhoehung
```

```
-----
```

it1	2674.38
it2	3045.38
it3	3208.62
it4	3541.46
it5	4008.92

UNION

REGELN:

- **ALLE SELECT-ANWEISUNGEN BESITZEN DIE GLEICHE ANZAHL SPALTEN**
 - **DIE SPALTEN BESITZEN DIE GLEICHEN DATENTYPEN**
 - **NUR DIE LETZTE SELECT-ANWEISUNG DARF DIE ORDER BY-KOMPONENTE ENTHALTEN, SORTIERT WIRD AUF DER GRUNDLAGE DES ENDERGEBNISSES**
- **DOPPELTE ZEILEN WERDEN AUTOMATISCH GELÖSCHT. DISTINCT DARF NICHT VORKOMMEN.**


```
SELECT NACHNAME, VORNAME  
FROM PERSONAL  
WHERE PNR=132
```

```
UNION
```

```
SELECT NACHNAME, VORNAME  
FROM PERSONAL  
WHERE PNR=163
```

```
UNION
```

```
SELECT NACHNAME, VORNAME  
FROM PERSONAL  
WHERE PNR=173
```

```
ORDER BY 1;
```

```
SELECT NACHNAME, VORNAME  
FROM PERSONAL  
WHERE PNR=132 OR PNR=163 OR PNR=173  
ORDER BY 1;
```

```
SQL> SELECT NACHNAME, VORNAME FROM PERSONAL WHERE PNR=132
UNION
SELECT NACHNAME, VORNAME FROM PERSONAL WHERE PNR=163
UNION
SELECT NACHNAME, VORNAME FROM PERSONAL WHERE PNR=173
ORDER BY 1;
```

NACHNAME	VORNAME
----------	---------

Gamma	Erich
Helm	Richard
Hopper	Grace

SUBQUERY

- **VOLLSTAENDIGE SELECT-ANWEISUNG, DIE ALS RECHTSSEITIGER AUSDRUCK IN EINER WHERE-BEDINGUNG VERWENDET WIRD.**
 - **BEI OPERATOREN WIE IN, ANY, ALL, EXISTS:**
 - **SELECT-KOMPONENTE DARF NUR EINEN SPALTENAUSDRUCK ENTHALTEN**
 - **DISTINCT NICHT ERLAUBT — ORDER BY NICHT ERLAUBT**
- **ATTRIBUTNAMEN GELTEN IN DER SELECT-ANWEISUNG, IN DER IHRE RELATION ANGEGEBEN IST, UND IN ALLEN ZUGEHÖRIGEN UNTERABFRAGEN.**

WIE OFT WURDE DER MINIMALE PRAEMIENBETRAG GEZAHLT ?

```
SELECT COUNT(*)  
FROM PRAEMIE  
WHERE P_BETRAG =  
      (SELECT MIN(P_BETRAG)  
        FROM PRAEMIE);
```

```
SQL> SELECT COUNT(*) FROM PRAEMIE WHERE P_BETRAG = (SELECT MIN(P_BETRAG) FROM PRAEMIE);
```

COUNT(*)
1

ORDER BY

- **SORTIERT DIE ZEILE AUF DER GRUNDLAGE DER ANGEgebenEN ATTRIBUTE**
- **DIE ATTRIBUTE WERDEN DURCH IHRE NAMEN ODER DURCH DIE ANGABE DER SPALTENNUMMER GEKENNZEICHNET**
- **NUMMER MUß STEHEN, WENN SPALTENAUSDRUCK AUS EINER FUNKTION, EINER KONSTANTEN ODER EINEM NUMERISCHEN AUSDRUCK BESTEHT**
 - **STANDARDMAEßIG AUFSTEIGEND (ASC) SORTIERT**
 - **ABSTEIGEND SORTIERT BEI ANGABE VON DESC**

AUFGABENBLATT



**DAS WAR'S FÜR
HEUTE**