

# SEN4

Felix Hofinger

October 2022

## Inhaltsverzeichnis

<b>1</b>	<b>Datenabfrage</b>	<b>3</b>
1.1	Einfache Abfrage . . . . .	3
1.2	Bedingung . . . . .	3
1.3	Datensätze sortieren . . . . .	4
1.4	Datensätze gruppieren . . . . .	4
1.5	Verschachtelte Abfragen . . . . .	5
1.6	Tabellen verknüpfen (Joining) . . . . .	5
1.6.1	Die 4 Join-Arten . . . . .	6

# 1 Datenabfrage

```
SELECT [DISTINCT]
      { * | Attributliste | mathematische Ausdrücke } Bezeichner
FROM Tabelle1 Bezeichner1, Tabelle 2 Bezeichner2, ...
[WHERE BEDINGUNG]
[GROUP BY Attributliste] [HAVING Bedingung]
[ORDER BY Attributliste] [ASC | DESC]
```

## 1.1 Einfache Abfrage

```
SELECT *
FROM Kursthemen
```

TNr	Themengebiet
1	Sicherheit und Umweltschutz
2	...

Tabelle 1: Beispiel Ergebnis

---

```
SELECT PNr, Name, Vorname,
      (Lohnstufe - 1) * 10000 + 60000 Salary
FROM Personen
```

PNr	Name	Vorname	Salary
100001	Steffen	Felix	100000
...			

Tabelle 2: Beispiel Ergebnis

## 1.2 Bedingung

```
SELECT PNr, Name, Vorname
FROM Personen
WHERE FNr = 1
```

### 1.3 Datensätze sortieren

```
SELECT *
FROM Funktionen
ORDER BY Funktion
```

FNr	Funktion
4	Bereichsleiter
...	

Tabelle 3: Beispiel Ergebnis

### 1.4 Datensätze gruppieren

PNr	VN	KN	FNr	Salary		FNr
1			3			
2			3			
3			4			
4			5		=>	3
5			4			4
6			4			5
7			4			
8			3			

Tabelle 4: Beispiel: „GROUP BY“

```
SELECT FNr, COUNT(FNr) Anzahl,
       AVG(Lohnstufe - 1) * 10000 + 60000 Salary
FROM Personen
GROUP BY FNr
ORDER BY FNr DESC
```

In Verbindung mit dem Schlüsselwort „GROUP BY“ gibt es noch das Schlüsselwort „HAVING“ welches das definieren von Gruppenbedingungen ermöglicht. Im Gegensatz zu „WHERE“ werden die mit „HAVING“ angegebenen Bedingungen nicht auf einzelne Datensätze, sondern auf Datensatzgruppen angewendet.

## 1.5 Verschachtelte Abfragen

```

SELECT KNr, Kursbezeichnung
FROM Kurse
WHERE KNr IN (
    SELECT KNr
    FROM Kursbesuche
    WHERE PNr = (
        SELECT PNr
        FROM Personen
        WHERE Name = 'Steffen'
        AND Vorname = 'Felix'
    )
)
ORDER BY Kursbezeichnung

```

---

```

SELECT FNr, Name, Vorname, Lohnstufe
FROM Personen
WHERE (FNr, Lohnstufe) IN (
    SELECT FNr, MAX(Lohnstufe)
    FROM Personen
    GROUP BY FNr
)
ORDER BY FNr DESC, Name ASC

```

## 1.6 Tabellen verknüpfen (Joining)

A	B	C		D	E	F
I	~	~	X	$\alpha$	~	~
II	~	~		$\beta$	~	~

A	B	C	D	E	F	
I	~	~	$\alpha$	~	~	SELECT * FROM Funktionen, Personen
I	~	~	$\beta$	~	~	
II	~	~	$\alpha$	~	~	
II	~	~	$\beta$	~	~	

Tabelle 5: Beispiel: „CROSS JOIN“

A	B	C'		C''	D	E
I	~	$\alpha$	X	$\alpha$	~	~
II	~	$\alpha$		$\beta$	~	~
III	~	$\beta$				

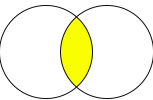
A	B	C'	C''	D	E	
I	~	$\alpha$	$\alpha$	~	~	SELECT * FROM Funktionen, Personen WHERE Funktionen.FNr = Personen.FNr
II	~	$\alpha$	$\alpha$	~	~	
III	~	$\beta$	$\beta$	~	~	

Tabelle 6: Beispiel: „INNER JOIN“

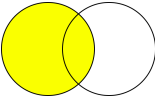
### 1.6.1 Die 4 Join-Arten

A	B	C		D	E	F
I	~	1	X	a	1	~
II	~	2		b	2	~
III	~	2		d	55	~
IV	~	99				

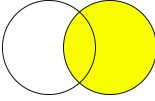
Tabelle 7: Beispiel Tabellen für die 4 Join-Arten

inner join: 

A	B	C	D	E	F
I	~	1	a	1	~
II	~	2	b	2	~
III	~	2	b	2	~

left join: 

A	B	C	D	E	F
I	~	1	a	1	~
II	~	2	b	2	~
III	~	2	b	2	~
IV	~	99	NULL	NULL	NULL

right join: 

A	B	C	D	E	F
I	~	1	a	1	~
II	~	2	b	2	~
III	~	2	b	2	~
NULL	NULL	NULL	d	55	~

cross join: Jedes Tupel mit jedem