

1 GROUP BY

Ort	PLZ
Hamburg	20095
Hamburg	20537
Wedel	22880
Pinneberg	25421
Wedel	22880

(a) Orte

Sinnvolle Beispielabfragen

```
1 SELECT ort
2 FROM orte
3 GROUP BY ort
```

```
4 SELECT plz
5 FROM orte
6 GROUP BY plz
```

```
7 SELECT plz
8 FROM orte
9 GROUP BY ort
```

```
10 SELECT plz, ort
11 FROM orte
12 GROUP BY plz, ort
```

Antworten

Zeile 1-3: Alle Ortsnamen, im Endeffekt ein `SELECT DISTINCT ort`.

Zeile 4-6: Alle Postleitzahlen, im Endeffekt ein `SELECT DISTINCT plz`.

Zeile 7-9: Nicht möglich, Postleitzahlen sind nicht funktional vom Ort abhängig, daher gäbe es innerhalb der Gruppe „Hamburg“ mehrere Möglichkeiten eine Postleitzahl auszugeben.

Zeile 10-12: Alle einzigartigen Kombinationen aus Postleitzahl und Ort, im Endeffekt ein `SELECT DISTINCT plz, ort`.

Zusammenfassung

- Demonstriert die „Verwandtschaft“ zwischen GROUP BY und DISTINCT.
- Zeigt die Probleme bei der Selektion von Spalten, welche bei einem GROUP BY weder selbst gruppiert noch aggregiert werden.

2 COUNT DISTINCT

tag	raum	zeitslot	veranstaltung
Mo	R1	1	A
Mo	R1	2	B
Mo	R1	4	D
Mo	R1	4	E
Mo	R2	1	A

(a) Veranstaltungen

Sinnvolle Beispielabfragen

```
1 SELECT v.tag, v.raum, COUNT(DISTINCT v.zeitslot)
2 FROM Veranstaltungen v
3 GROUP BY v.tag, v.raum
```

```
1 SELECT v.tag, v.raum, COUNT(DISTINCT v.raum, v.zeitslot)
2 FROM Veranstaltungen v
3 GROUP BY v.tag
```

Zusammenfassung

- Demonstriert die Verwendung von **COUNT** und **DISTINCT**

3 JOIN Allgemein

kkid	name
1	A
2	B
3	C

(a) Krankenkasse

pin	name	kkid
1	X	1
2	Y	1
3	Z	3

(b) Person

Sinnvolle Beispielabfragen

```
1 SELECT *
2 FROM krankenkasse k, Person p
```

```
1 SELECT *
2 FROM krankenkasse k, Person p
3 WHERE k.kkid = p.kkid
```

```
1 SELECT *
2 FROM krankenkasse k
3 LEFT OUTER JOIN person p USING(kkid)
```

Antworten

Zeile 1-2: 9 Zeilen, 4 Spalten

Zeile 3-5: 3 Zeilen, 4 Spalten

Zeile 6-8: 4 Zeilen, 4 Spalten

Zusammenfassung

- Grundlegender Aufbau aller JOINS, inklusive dem Zusammenhang mit dem Kreuzprodukt.
- Von Hand vorrechnen lassen, eine Relation je Abfrage.
- Immer im Voraus nach Anzahl Zeilen / Spalten fragen.

4 OUTER JOIN

kkid	name
1	A
2	B
3	C

(a) Krankenkasse

pin	name	kkid
1	X	1
2	Y	1
3	Z	3
4	M	1
5	U	null

(b) Person

pin
1
2
3
5

(c) Student

Sinnvolle Beispielabfragen

```
1 SELECT *
2 FROM krankenkasse k
3   INNER JOIN person p USING (kkid)
```

```
4 SELECT *
5 FROM krankenkasse k
6   LEFT OUTER JOIN person p USING (kkid)
```

```
7 SELECT *
8 FROM krankenkasse k
9   LEFT OUTER JOIN person p USING (kkid)
10  INNER JOIN student s USING (pin)
```

```
11 SELECT *
12 FROM krankenkasse k
13   LEFT OUTER JOIN person p USING (kkid)
14   LEFT OUTER JOIN student s USING (pin)
```

Zusammenfassung

- Demonstriert, dass sich JOINS im Allgemeinen immer auf die ganze bisherige Zwischenrelation beziehen.

Antworten

k.kkid	k.name	p.pin	p.name	p.kkid
1	A	1	X	1
1	A	2	Y	1
1	A	4	M	1
3	C	3	Z	3

Tabelle 5: Ergebnis Zeile 1 - 3

k.kkid	k.name	p.pin	p.name	p.kkid
1	A	1	X	1
1	A	2	Y	1
1	A	4	M	1
3	C	3	Z	3
2	B	null	null	null

Tabelle 6: Ergebnis Zeile 4 - 6

k.kkid	k.name	p.pin	p.name	p.kkid	s.pin
1	A	1	X	1	1
1	A	2	Y	1	2
3	C	3	Z	3	3

Tabelle 7: Ergebnis Zeile 7 - 10

k.kkid	k.name	p.pin	p.name	p.kkid	s.pin
1	A	1	X	1	1
1	A	2	Y	1	2
1	A	4	M	1	null
3	C	3	Z	3	3
2	B	null	null	null	null

Tabelle 8: Ergebnis Zeile 11 - 14