

Stellen Sie als erstes mit eigenen Worten folgende Anfrage auf:

Sie möchten wissen, welche Kleidung zu welchem Anlass an welchem Tag getragen wurde!

Malen sie Tabellen inklusive Daten der letzten Übung auf!

Führen sie folgende Statements aus:

```
SELECT a.nr, farbe, k.nr, bezeichnung
FROM kleidung k, anlass a, anlass_kleidung
WHERE a.nr = anlass_nr
AND Kleidung_Nr = k.nr
```

Alias:

Ein Alias ist ein temporärer Name, der in einer SQL-Anweisung einer Tabelle oder Spalte zugewiesen und in der selben Anweisung (bei Tabellen) verwendet wird.

Warum muss ein Alias-Name vergeben werden?

Führen sie hierzu folgendes Statement aus. Notieren sie sich die Fehlermeldung!

```
SELECT nr, farbe, k.nr, bezeichnung
FROM kleidung k, anlass a, anlass_kleidung
WHERE a.nr = anlass_nr
AND Kleidung_Nr = k.nr
```

Was ist der Unterschied zwischen den folgenden zwei Abfragen?

```
SELECT a.nr, farbe, k.nr, bezeichnung
FROM kleidung k, anlass a, anlass_kleidung
WHERE a.nr = anlass_nr
AND Kleidung_Nr = k.nr
```

```
SELECT a.nr, farbe, k.nr, bezeichnung
FROM kleidung k, anlass a, anlass_kleidung
```

Join-Spalte

Eine Join-Spalte wird für die Verknüpfung zweier Tabellen verwendet. Sie wird innerhalb der WHERE-Klausel identifiziert, in der die Beziehung der Join-Spalten der beiden Tabellen spezifiziert werden. Anderes ausgedrückt sind das die Spalten, die in beiden Tabellen vorhanden sind. Hinweis: Diese können einen unterschiedlichen Namen besitzen.

Welche Spalten sind typische Join_Spalten?

Malen sie die zusammengehörigen Spalten farbig aus!

Fremdschlüssel-Spalten

Stellen sie folgende Abfragen auf:

Sie möchten wissen, an welchem Tag ein rotes Kleidungsstück getragen wurde.

```
SELECT *
FROM kleidung k, anlass_kleidung
WHERE Kleidung_Nr = k.nr
```

```
AND farbe = 'rot';
```

Sie möchten wissen, an welchem Tag ein blaues Kleidungsstück zu welchem Anlass getragen wurde.

```
SELECT *
FROM kleidung k, anlass a, anlass_kleidung
WHERE a.nr = anlass_nr
AND Kleidung_Nr = k.nr
AND farbe = 'blau';
```

Sie möchten wissen, an welchem Tag ein gelbes Abendkleid zu welchem Anlass getragen wurde.

**Schreiben sie sich bitte die Abfrage erst wieder in Worten auf!
In welchem Tabellen stehen die Informationen für unsere Abfrage?
Was sind in diesem Fall die Join-Spalten?**

```
Kleidungsart – Nr
Kleidung – Nr
Anlass_Kleidung – Anlass_Nr
Anlass_Kleidung – Kleidung_Nr
```

```
SELECT *
FROM kleidung k, anlass a, anlass_kleidung, Kleidungsart ka
WHERE a.nr = anlass_nr
AND Kleidung_Nr = k.nr
AND art = ka.nr
AND farbe = 'gelb';
AND ka.bezeichnung = 'Abendkleid';
```

GROUP BY

Mit Hilfe von GROUP BY können Tabellen zu Gruppen mit zusammengehörigen Zeilen zusammengefasst werden.

HAVING BY

HAVING BY legt fest, welche Gruppe durch GROUP BY einzuschließen sind.

Was macht folgende Abfrage?

```
SELECT count(Farbe)
FROM Kleidung
GROUP BY Farbe;
```

```
SELECT *
FROM Kleidung
ORDER BY Farbe;
```

```
SELECT count(Farbe)
FROM Kleidung
GROUP BY Farbe
HAVING Farbe = 'grün';
```

Zeigen sie ihre Kleidungsstücke an, welche nach der Art des Anlasses (angezeigt mit Name) gruppiert sind!

```
SELECT *  
FROM Kleidungsart, Kleidung k  
WHERE art = k.Nr  
GROUP BY art;
```

Warum muss hier nur ein Alias für die Tabelle Kleidung vergeben werden?

Weil die Nr in beiden Tabellen vorkommt, die Art aber nur in einer Tabelle!

Kleidungsart

Nr	Bezeichnung
1	T-Shirt
3	Abendkleid
2	Rock

Kleidung

Nr	Art	Farbe
1	2	rot
2	2	grün
3	3	gelb
4	1	blau
5	1	rot
6	1	grün

Anlass

Nr	Bezeichnung	Bemerkung
1	Prüfung	muss sehr schick sein
2	Vorlesung	niedrige Relevanz
3	Sonntagsausflug	
4	Date	

Anlass_Kleidung

Kleidung_Nr	Anlass_Nr	Datum
1	1	12.02.2002
1	1	18.02.2002
3	1	12.05.2002
4	2	09.05.2002
4	4	10.05.2002

```
CREATE TABLE Anlass_Kleidung (  
  Kleidung_Nr number  
    constraint Anlass_Kleidung1 references Kleidung (nr)  
  Anlass_Nr number  
    constraint Anlass_Kleidung2 references Anlass (nr)  
  Datum date;  
);
```

```
INSERT INTO Anlass_Kleidung VALUES (1, 1,  
to_date('12.02.2002'));  
INSERT INTO Anlass_Kleidung VALUES (1, 1,  
to_date('18.02.2002'));  
INSERT INTO Anlass_Kleidung VALUES (3, 1,  
to_date('12.05.2002'));  
INSERT INTO Anlass_Kleidung VALUES (4, 2,  
to_date('09.05.2002'));  
INSERT INTO Anlass_Kleidung VALUES (4, 4,  
to_date('10.05.2002'));
```