

Übungsbeispiel 2a - Lösung

Achtung – bei Hausübung / Prüfung muss ebenfalls der Lösungsweg ersichtlich sein.

- i) Schlüssel: ab, ac, NSA: d
3. Normalform
- ii) Schlüssel: ab, NSA: cd
BCNF
- iii) Schlüssel: ab, NSA: cde
1. Normalform
Zerlegung: $r_1: (abcd | ab \rightarrow c, ab \rightarrow d)$, $r_2: (be | b \rightarrow e)$
- iv) Schlüssel: b, NSA: acde
2. Normalform
Zerlegung: $r_1: (bcd | b \rightarrow d, b \rightarrow c, b \rightarrow e)$, $r_2: (ac | c \rightarrow a)$

Übungsbeispiel 2b - Lösungen

Achtung – bei Hausübung / Prüfung muss ebenfalls der Lösungsweg ersichtlich sein.

- $r_A: r_{A1}=(abd | ab \rightarrow d)$, $r_{A2}=(abc | b \rightarrow c)$
- $r_B: r_{B1}=(abc | ab \rightarrow c)$, $r_{B2}=(fg | f \rightarrow g)$, $r_{B3}=(cgd | cg \rightarrow d)$, $r_{B4}=(de | d \rightarrow e)$,
 $r_{B5} = (abfh)^1$

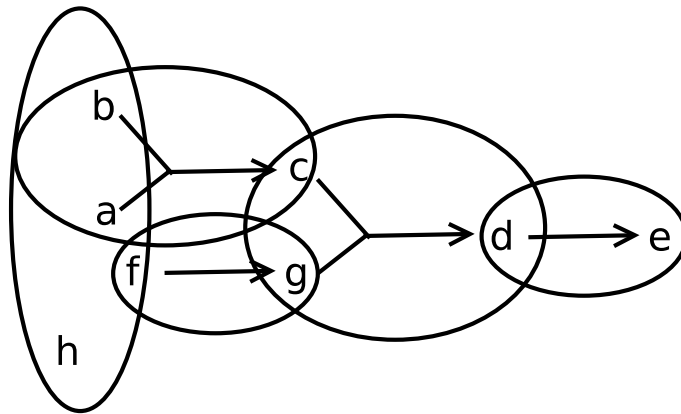


Abbildung 1: Zerlegung von r_B .

¹ r_{B5} enthält den Schlüssel der ursprünglichen Relation und stellt somit sicher, dass die Zerlegung verlustfrei ist

Übungsbeispiel 2c - Lösungen

```
1. CREATE TABLE Person (  
    SVNr CHAR(11) CHECK (VALUE LIKE '____ ____') PRIMARY KEY,  
    Name VARCHAR,  
    Wohnort VARCHAR  
);
```

```
2. CREATE TABLE Lektor (  
    SVNr CHAR(11) REFERENES Person(SVNr) PRIMARY KEY,  
    Berufsbezeichnung VARCHAR,  
    Abteilung VARCHAR  
);
```

```
CREATE TABLE haeltAb (  
    SVNr CHAR(11) REFERENCES Lektor(SVNr),  
    LVNr Integer REFERENCES Lehrveranstaltung(LVNr),  
    ECTS DECIMAL);
```

```
3. SELECT * FROM Hoersaal WHERE Bezeichnung LIKE '____'  
    AND Groesse>50 ORDER BY Gebaeude DESC, Bezeichnung DESC;
```

$\sigma_{[BezeichnungLIKE'____'\wedge Groesse>50]}Hoersaal$

```
4. SELECT LVNr, Name, COUNT(SVNr) AS NumTeilnehmer  
    FROM Besucht JOIN Lehrveranstaltung USING(LVNr)  
    GROUP BY LVNr, Name HAVING COUNT(SVNr) BETWEEN 30 AND 50;
```

Hinweis: (i) es ist notwendig nach LVNr **und** Name zu Gruppieren, da es sonst nicht möglich ist die Name auszugeben (alle Felder, die nicht in einer Aggregationsfunktion wie zum Beispiel COUNT, SUM und MIN verwendet werden, müssen mittels GROUP BY als Gruppierungsfelder definiert werden.
(ii) Der Term “AS NumTeilnehmer” ist optional und führt nur dazu, dass die entsprechende Spalte umbenannt wird.

```
5. -- Schritt 1: Lehrveranstaltung eingeben  
INSERT INTO Lehrveranstaltung (LVNr, Name, maxTeilnehmer)  
    VALUES (123, 'Datenbanksysteme', 20);
```

```
-- Schritt 2: Der Lehrveranstaltung einen Lektor zuordnen  
INSERT INTO haeltAb (SVNr, LVNr, ECTS) VALUES (2611, 123, 4);
```

```
6. UPDATE TABLE Person SET name = 'Kurz-Wyss' WHERE SVNr='2611';
```

Hinweis: Der Wert von SVNr steht unter Anführungszeichen, da wir das entsprechende Feld als CHAR(11) definiert haben. Hätten wir einen numerischen Typen (INTEGER, FLOAT, DECIMAL) vergeben, so müsste der Wert ohne Anführungszeichen stehen.

7. DELETE FROM Lektoren WHERE BERUFSBEZEICHNUNG = 'UNIV.DOZ'
OR SVNr IN
(SELECT SVNr FROM Person WHERE Name LIKE 'S%' AND NOT NAME LIKE '%ER');
8. SELECT DISTINCT SVNn FROM haeltAb;
 $\pi_{[SVNr]}(haeltAb)$
9. SELECT Studium FROM Student
GROUP BY Studium HAVING COUNT(SVNR)>2000;
10. SELECT Gebaeude, COUNT(Bezeichnung) FROM Hoersaal
GROUP BY Gebaeude;
11. SELECT Ort, COUNT(LVNr) FROM Lehrveranstaltung
GROUP BY Ort
HAVING COUNT(LVNr) > 5;

Hinweis: Das Attribut Ort ist ein Fremdschlüssel auf Hoersaal.Bezeichnung.

12. SELECT LVNr, Name, maxTeilnehmer, COUNT(SVNr) as countTeilnehmer,
COUNT(SVNr) / maxTeilnehmer * 100 AS Auslastung
FROM Lehrveranstaltung JOIN besucht USING(LVNr)
GROUP BY LVNr, Name, maxTeilnehmer
HAVING COUNT(SVNr) / maxTeilnehmer * 100 < 70;