

DATENBANKEN

Datenbankdesign

Rohdaten für eine Song – Datenbank:

Title	Album	Artist 1	Artist 2
Bella Ciao	Bella ciao	El Profesor	HUGEL
Lucky You	Kamikaze	Eminem	
Girls Like You	Red Pill Blues	Maroon 5	Cardi B
Denim Jacket	Red Pill Blues	Maroon 5	
Revenge	Beautiful Trauma	Pink	Eminem

1. Normalform:

- Spalten mit gleichartigem Inhalt müssen entfernt werden.
- Für jede Gruppe zusammengehöriger Daten muss eine eigene Tabelle gebildet werden.
- Jeder Datensatz muss durch einen Primärschlüssel eindeutig identifizierbar sein.

id (primary key)	Title	Album	Artist
1	Bella Ciao	Bella ciao	El Profesor
2	Bella Ciao	Bella ciao	HUGEL
3	Lucky You	Kamikaze	Eminem
4	Girls Like You	Red Pill Blues	Maroon 5
5	Girls Like You	Red Pill Blues	Cardi B
6	Denim Jacket	Red Pill Blues	Maroon 5
7	Revenge	Beautiful Trauma	Pink
8	Revenge	Beautiful Trauma	Eminem

2. Normalform

- Wenn sich Inhalte in Spalten wiederholen, müssen die Tabellen in mehrere Teiltabellen zerlegt werden.
- Die Tabellen müssen durch Fremdschlüssel (foreign keys, „Querverweise“) miteinander verbunden werden. (Diese Verknüpfungen heißen **Relationen**.)

2. Normalform – 1. Versuch:

Tabelle Titles:

titleID (prim. key)	Title	Album
1	Bella Ciao	Bella ciao
2	Lucky You	Kamikaze
3	Girls Like You	Red Pill Blues
4	Denim Jacket	Red Pill Blues
5	Revenge	Beautiful Trauma

Tabelle Artists:

artistID (prim. key)	titleID (foreign key)	Artist
1	1	El Profesor
2	1	HUGEL
3	3	Eminem
4	5	Eminem
5	3	Maroon 5
6	4	Maroon 5
7	3	Cardi B
8	5	Pink

2. Normalform – 2. Versuch:

Tabelle Titles:

titleID (prim. key)	Title	Album
1	Bella Ciao	Bella ciao
2	Lucky You	Kamikaze
3	Girls Like You	Red Pill Blues
4	Denim Jacket	Red Pill Blues
5	Revenge	Beautiful Trauma

Tabelle Artists:

artistID (prim. key)	Artist
1	El Profesor
2	HUGEL
3	Eminem
4	Maroon 5
5	Cardi B
6	Pink

Tabelle Relation Titles – Artists:

titleID	ArtistID
1	1
1	2
2	3
3	4
3	5
4	4
5	3
5	6

3. Normalform

- Spalten, die nicht in unmittelbarer Abhängigkeit zum Primärschlüssel einer Tabelle stehen, müssen eliminiert werden (d.h. in eine eigene Tabelle ausgelagert werden).

Betroffen ist hier die Tabelle Titles (Spalte Album auslagern).

Tabelle Titles:

titleID (prim. key)	Title	AlbumID (foreign key)
1	Bella Ciao	1
2	Lucky You	2
3	Girls Like You	3
4	Denim Jacket	3
5	Revenge	4

Tabelle Albums:

albumID (prim. key)	Album
1	Bella ciao
2	Kamikaze
3	Red Pill Blues
4	Beautiful Trauma

Umsetzung in SQL

Einrichtung der Datenbank als Administrator

(auf schueler.bulme.at schon vom Administrator erledigt):

```
CREATE DATABASE db_uebung;  
GRANT ALL PRIVILEGES ON db_uebung.*  
    TO '$franz'@'localhost' IDENTIFIED BY 'babablub';"
```

Alles ab hier als normaler User (franz):

```
USE db_uebung;
```

Einfache, nicht normierte Tabelle ohne Primary Key erstellen:

```
CREATE TABLE songs (  
    title VARCHAR(100) NOT NULL,  
    album VARCHAR(100) NOT NULL,  
    artist1 VARCHAR(100) NOT NULL,  
    artist2 VARCHAR(100)  
);
```

Überprüfen:

```
DESCRIBE songs;
```

Daten einfügen:

```
INSERT INTO songs (title, album, artist1)  
VALUES ('Bella Ciao', 'Bella Ciao', 'El Profesor');
```

Überprüfen:

```
SELECT * FROM songs;
```

Wie oben, aber Tabelle mit Primary Key:

```
CREATE TABLE songs (
    id INT NOT NULL AUTO_INCREMENT,
    title VARCHAR(100) NOT NULL,
    album VARCHAR(100) NOT NULL,
    artist VARCHAR(100) NOT NULL,
    PRIMARY KEY(id)
);
```

Verknüpfung mit einer Tabelle songs_albums mittels Foreign Key:

```
CREATE TABLE songs_titles (
    titleID INT NOT NULL AUTO_INCREMENT,
    title VARCHAR(100) NOT NULL,
    albumID INT,
    PRIMARY KEY(titleID),
    FOREIGN KEY(albumID) REFERENCES songs_albums(albumID)
);
```

Abfragen aus verknüpften Tabellen (Title/Album):

```
SELECT title, album
    FROM songs_titles, songs_albums
    WHERE songs_titles.albumID = songs_albums.albumID;
```

oder:

```
SELECT title, album
    FROM songs_titles
    LEFT JOIN songs_albums USING (albumID);
```

Abfragen aus drei verknüpften Tabellen (Title/Artist, über songs_rel_titles_artists) sortiert nach Title:

```
SELECT title, artist
    FROM songs_titles, songs_rel_titles_artists, songs_artists
    WHERE songs_titles.titleID = songs_rel_titles_artists.titleID
    AND songs_artists.artistID = songs_rel_titles_artists.artistID
    ORDER BY title;
```

Welche Titel – in einer Zeile angezeigt - enthält ein Album?

```
SELECT album, GROUP_CONCAT(title)
    FROM songs_titles, songs_albums
    WHERE songs_albums.albumID = songs_titles.albumID
    GROUP BY album;
```

Welcher Artist hat die meisten Titel?

```
SELECT COUNT(*), artist
    FROM songs_titles, songs_rel_titles_artists, songs_artists
    WHERE songs_titles.titleID = songs_rel_titles_artists.titleID
    AND songs_artists.artistID = songs_rel_titles_artists.artistID
    GROUP BY artist ORDER BY 1 DESC
```

TIPP:

Wähle eine sinnvolle Namenskonvention bei der Erzeugung der Tabellen, z.B. `projektname_tabellenname` und `projektname_rel_tabellenname1_tabellenname2`