

Übungsblatt D1

(Relationale Algebra & SQL)

Ausgabe: 18.05.2009
Besprechung: 25.05.2009

Alle Aufgaben auf diesem Übungsblatt beziehen sich auf ein leicht abgewandeltes Datenbankschema des *MusicBrainz*-Projektes¹, einer Datenbank über Musik.

Abbildung 1 zeigt alle zur Bearbeitung der Aufgaben relevanten Relationen. Unterstrichene Attribute markieren Schlüssel, Pfeile markieren Fremdschlüsselbeziehungen.

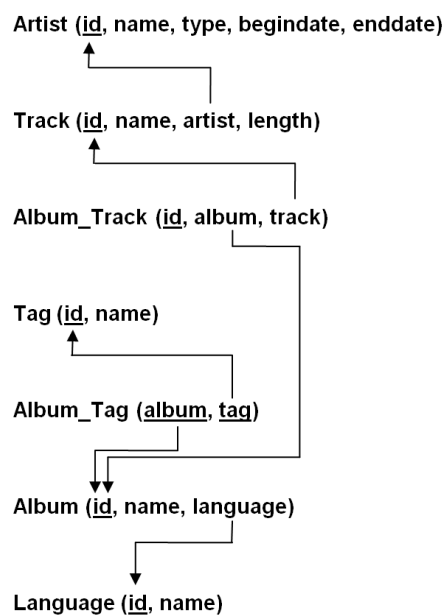



Abbildung 1: Schema “MusicBrainz”

¹<http://musicbrainz.org/>

Anmerkungen zum Schema

- Ein Album besteht aus einem oder mehreren Tracks (Liedern), wobei jeder Track einen Namen hat und genau einem Künstler (Relation “Artist”) zugeordnet ist.
- Das Attribut “length” der Relation “Track” ist ein numerischer Wert und gibt die Länge eines Tracks in Millisekunden an. Ein Track kann über die Relation “Album_Track” einem Album zugewiesen werden.
- Das Attribut “type” der Relation “Artist” ist ein numerischer Wert aus der Menge {0, 1, 2}. 1 kennzeichnet Personen (z. B. “Joe Strummer” oder “Gisbert zu Knyphausen”), 2 Gruppen (z. B. “The Clash” oder “The Felice Brothers”) und 0 Tupel, für die keine Angabe vorliegt. Das Attribut “type” kann auch NULL sein, was in diesem Fall semantisch gleichzusetzen ist mit 0.
- Das Attribut “begindate” der Relation “Artist” enthält das Geburtsdatum eines Künstlers respektive das Gründungsdatum einer Gruppe. Das Attribut “enddate” enthält das Todesdatum eines Künstlers oder das Datum der Auflösung einer Gruppe. Beide Attribute können den Wert NULL annehmen, falls das Datum nicht bekannt ist, nicht spezifiziert wurde oder das entsprechende Ereignis noch nicht eingetreten ist.
- Die Relation “Tag” enthält Schlagwörter (Tags), die über die Relation “Album_Tag” Alben zugeordnet werden können. Die folgende Abbildung ist eine grafische Repräsentation von Tags wie sie der Band “Nirvana” zugewiesen wurden:



90 90s alternative rock american
americana grunge legendary northwest
punk rock rock and indie seattle usa

Jedem dieser Schlagwörter wird genau ein Tupel in der Relation “Tag” zugewiesen, das Attribut “name” enthält das Schlagwort selbst. So gibt es z.B. ein Tupel mit der ID 175 dessen Attribut “name” mit “grunge” belegt ist.

Web-Interface zur Datenbank

Wir haben für Sie die Möglichkeit eingerichtet, SQL-Anfragen gegen dieses Schema laufen zu lassen:

Webseite: <http://i40virt00.ipd.uka.de/phpPgAdmin/>
Benutzername: student
Passwort: komdat2009

Auf der Webseite sehen Sie links eine Liste der Server. Klicken Sie hier auf den Eintrag *MusicBrainz* und melden sich mit obigen Zugangsdaten am System an. Wenn Sie nach erfolgreichem Einloggen oben rechts den Link *SQL* verfolgen, öffnet sich ein neues Fenster, in dem Sie SQL-Anfragen absetzen können. Das Ergebnis einer Anfrage wird Ihnen im Hauptfenster angezeigt.

Bitte beachten Sie, dass die Datenbank mit über 400 000 Künstlern, 650 000 Alben und 7,5 Millionen Tracks relativ groß ist. Bestimmte Anfragen können daher einige Zeit benötigen, bis sie abgearbeitet sind. Auch empfiehlt es sich für Anfragen, die viele Tupel zum Ergebnis haben, die Option *Paginate results* im SQL-Abfragefenster zu aktivieren.

Aufgabe 1: Relationale Algebra

Formulieren Sie folgende Anfragen mit Hilfe der relationalen Algebra:

- a) Geben Sie die Namen aller Künstler aus!
- b) Gesucht sind die Namen aller schweizerdeutschen Alben (Sprache mit der ID 476).
- c) Geben Sie die Namen aller Alben aus, die einen Track enthalten, der länger als 100 Minuten ist.
- d) Welche Alben enthalten keinen Track? Geben Sie Namen und IDs dieser Alben aus.

Aufgabe 2: Relationale Algebra \rightarrow SQL

Setzen Sie die Anfragen aus Aufgabe 1 nun mittels SQL um. Achten Sie darauf, dass der relationale Ausdruck und der SQL-Ausdruck das gleiche Ergebnis liefern!

Aufgabe 3: SQL

Formulieren Sie folgende Anfragen mit Hilfe von SQL:

- a) Geben Sie die Anzahl der Künstler aus, deren Name mit der Zeichenkette "The " ("The" gefolgt von einem Leerzeichen) beginnt.
- b) Geben Sie die ID aller Alben aus, die sowohl mit dem Tag "hip-hop" (Tag mit der ID 150) als auch mit dem Tag "schlager" (Tag mit der ID 637) versehen wurden.

- c) Geben Sie anstatt der ID den Namen aller Alben aus Teilaufgabe b) aus.
- d) Manche Alben werden mehrfach veröffentlicht, z.B. als limitierte Editionen mit Bonustracks. Geben Sie den Namen aller Alben aus von Künstlern, die sieben oder mehr Alben mit dem selben Namen in ihrem Katalog haben. Geben Sie zusätzlich zum Namen des Albums den Namen des Künstlers und dessen ID aus und sortieren Sie das Ergebnis nach dem Namen des Künstlers.
- e) Gesucht sind die Künstler, deren Tracks im Schnitt am längsten sind. Berücksichtigen Sie für das Ergebnis nur Tracks, zu denen die Länge bekannt ist. Desweiteren sind Künstler auszuschliessen, die auf weniger als 100 Tracks kommen, zu denen die Länge bekannt ist. Sortieren Sie das Ergebnis absteigend nach der durchschnittlichen Länge der Tracks und geben Sie die ID und den Namen der Künstler und die durchschnittliche Länge derer Tracks in Minuten aus.

Aufgabe 4: SQL – Flexibilität

Setzen Sie die beiden Anfragen auf folgende vier verschiedene Arten um:

- i) Join-Anfrage
 - ii) (NOT) IN-Anfrage
 - iii) (NOT) EXISTS-Anfrage
 - iv) Anfrage mit Mengenoperationen
- a) Welche Alben wurden mit mindestens einem Tag versehen? Geben Sie die ID und den Namen dieser Alben aus.
 - b) Welchem noch lebenden Künstler ist kein Track zugeordnet? Ein lebender Künstler zeichnet sich durch `begindate` ungleich `NULL` und einem `enddate` gleich `NULL` aus. Geben Sie Namen und “`begindate`” des Künstlers aus.

Aufgabe 5: SQL – Häufige Fehler

Im Folgenden sind Anfragen sowohl in Freitext als auch in SQL gegeben. In die SQL-Anfragen haben sich syntaktische und/oder semantische Fehler eingeschlichen. Ändern Sie die SQL-Anfragen insoweit ab, dass sowohl die Syntax korrekt ist als auch die Anfrage im Freitext semantisch richtig umgesetzt wird.

- a) Geben Sie die ID aller Gruppen zurück, deren Gründungszeitpunkt nicht gesetzt ist.

```
(SELECT id FROM Artist WHERE begindate IS NULL)
INTERSECT
(SELECT name FROM Artist WHERE type = 2)
```

- b) Welche Namen für Künstler treten mehrfach in der Datenbank auf? Geben Sie zusätzlich zum Namen die Anzahl der Aufkommen an.

```
SELECT name, COUNT(*)  
FROM Artist  
HAVING COUNT(*) > 0
```

- c) Geben Sie ID und Namen aller Alben zurück, die mit dem Tag “motown” versehen wurden.

```
SELECT name, id  
FROM Album a  
WHERE EXISTS  
    (SELECT *  
     FROM Album_Tag at, tag t  
     WHERE at.tag = t.id  
     AND t.name = 'motown')
```

- d) Geben Sie die ID aller Alben zurück, die nicht mit dem Tag “metal” versehen wurden.

```
SELECT Album  
FROM Album_Tag  
WHERE tag != 'metal'
```

- e) Geben Sie für jeden Künstler die Anzahl der zum Künstler gehörigen Tracks zurück und sortieren Sie das Ergebnis absteigend nach dieser Zahl.

```
SELECT a.id, a.name, COUNT(*)  
FROM Artist a, Track t  
WHERE t.artist = a.id  
GROUP BY a.name  
ORDER BY COUNT(*) DESC
```