



Kommunikation und Datenhaltung

1. Übung zur Datenhaltung

Relationale Algebra & SQL



Organisatorisches

- Übungsblätter erhalten Sie eine Woche vor den Übungsterminen auf der Webseite
 - <http://www.ipd.uni-karlsruhe.de/~komdat/>
- Besprechung der Übungsblätter in der Übung
- Heute Besprechung des ersten Übungsblatts zur Datenhaltung: Relationale Algebra & SQL
- Annotierte und nicht-annotierte Folien kurze Zeit nach der Übung auf Webseite verfügbar

Datenbankschema

Artist (id, name, type, begindate, enddate)

Track (id, name, artist, length)

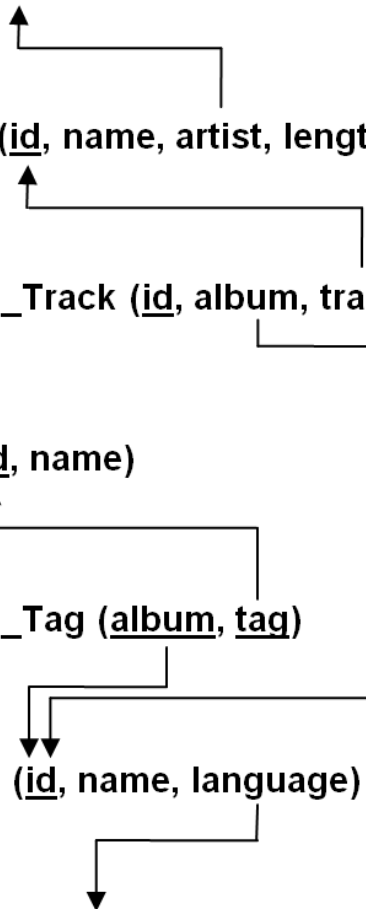
Album_Track (id, album, track)

Tag (id, name)

Album_Tag (album, tag)

Album (id, name, language)

Language (id, name)



Datenbankschema: Relation „Artist“

id	name	type	begindate	enddate
92744	A Tribe Called Quest	2	1988-00-00	<i>NULL</i>
34778	David Hasselhoff	1	1952-07-17	<i>NULL</i>
233	Dire Straits	2	1977-00-00	1995-00-00
2327	Johann Sebastian Bach	1	1685-03-21	1750-07-28
815	John Lennon	1	1940-10-09	1980-12-08
387	Johnny Cash	1	1932-02-26	2003-09-12
303	The Beatles	2	1957-00-00	1970-04-10
2333	The Clash	2	1976-00-00	1986-00-00
435359	The Felice Brothers	2	2006-00-00	<i>NULL</i>
23	Tom Waits	1	1949-12-07	<i>NULL</i>

Datenbankschema: Relation „Album“

id	name	language
2717	Abbey Road	120
22325	Sgt. Pepper's Lonely Hearts Club Band	120
70630	At Folsom Prison	120
370155	London Calling	120
370161	London Calling	120
369913	London Calling	120
227599	London Calling	<i>NULL</i>
23746	London Calling	120

Datenbankschema: Relation „Track“

id	name	artist	length
6534544	Looking for Freedom	34778	235000
7605384	Jump in My Car	34778	178000
1453345	Walk of Life	233	247706
439759	I'm Gonna Be (500 Miles)	829	216733
742185	The Man Comes Around	387	268826
190082	London Calling	2333	203826
190083	Brand New Cadillac	2333	130240
230136	Yesterday	303	0
198976	Let It Be	303	232400
1421141	Can I Kick It? (extended Boilerhouse mix)	92744	400426

Datenbankschema: Alben und deren Tracks

Relation „Album“

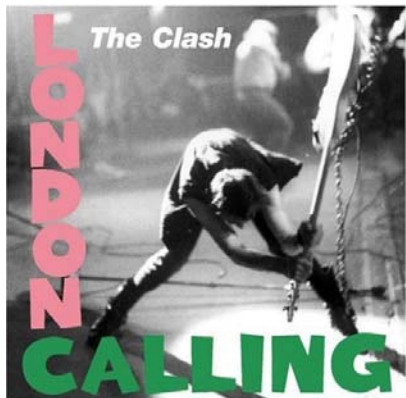
id	name	language
23746	London Calling	120

Relation „Track“

id	name	artist	length
190082	London Calling	2333	203826
190083	Brand New Cadillac	2333	130240

Relation „Album_Track“

id	album	track
190080	23746	190082
190081	23746	190083
190082	23746	190084
...		



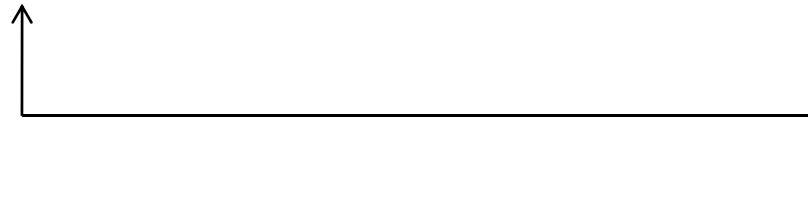
Datenbankschema: Sprachen

Relation „Language“

id	name
120	English
476	German, Swiss

Relation „Album“

id	name	language
2717	Abbey Road	120
22325	Sgt. Pepper's Lonely Hearts Club Band	120
70630	At Folsom Prison	120
370155	London Calling	120



Datenbankschema: Tags

Relation „Album“

id	name	language
23746	London Calling	120

Relation „Tag“

id	name
7	rock
88	punk
133	classic pop and rock

70s 80s alt rock alternative alternative rock **british**
 british punk britpop clash classic classic punk **classic**
rock dub eighties electronic england english experimental
 favorite favorites favourites funk funky garage rock grunge
 guitar hard rock hardcore indie indie rock japanese jazz joe
 strummer london male vocalists metal new wave nice old school
 old school punk oldies political pop post-punk progressive rock
punk punk 77 **punk rock** punk-rock
 reggae **rock** rock 'n' roll rockabilly ska ska punk stoner rock
 the clash the only band that matters uk 77

album	tag
23746	7
23746	88
23746	133

Relation „Album_Tag“



Aufgabe 1a)

Geben Sie die Namen aller Künstler aus!

Datenbankschema: Relation „Artist“

id	name	type	begindate	enddate
92744	A Tribe Called Quest	2	1988-00-00	<i>NULL</i>
34778	David Hasselhoff	1	1952-07-17	<i>NULL</i>
233	Dire Straits	2	1977-00-00	1995-00-00
2327	Johann Sebastian Bach	1	1685-03-21	1750-07-28
815	John Lennon	1	1940-10-09	1980-12-08
387	Johnny Cash	1	1932-02-26	2003-09-12
303	The Beatles	2	1957-00-00	1970-04-10
2333	The Clash	2	1976-00-00	1986-00-00
435359	The Felice Brothers	2	2006-00-00	<i>NULL</i>
23	Tom Waits	1	1949-12-07	<i>NULL</i>



Aufgabe 1a)

Geben Sie die Namen aller Künstler aus!

$$\pi_{\text{name}}(\text{Artist})$$



Aufgabe 1b)

Gesucht sind die Namen aller schweizerdeutschen Alben
(Sprache mit der ID 476).

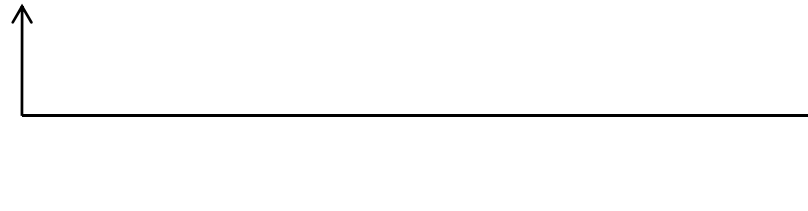
Datenbankschema: Sprachen

Relation „Language“

id	name
120	English
476	German, Swiss

Relation „Album“

id	name	language
2717	Abbey Road	120
22325	Sgt. Pepper's Lonely Hearts Club Band	120
70630	At Folsom Prison	120
370155	London Calling	120



Aufgabe 1b)

Gesucht sind die Namen aller schweizerdeutschen Alben (Sprache mit der ID 476).

$$\pi_{\text{name}} \left(\sigma_{[\text{language}=476]} (\text{Album}) \right)$$



Aufgabe 1c)

Geben Sie die Namen aller Alben aus, die einen Track enthalten, der länger als 100 Minuten ist.

Datenbankschema: Alben und deren Tracks

Relation „Album“

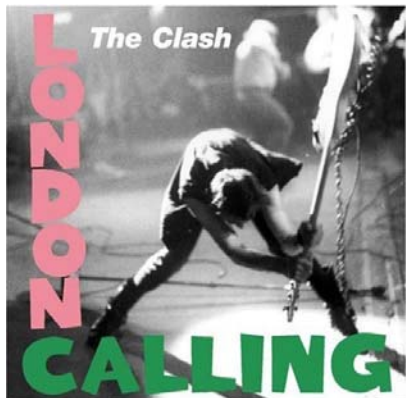
id	name	language
23746	London Calling	120

Relation „Track“

id	name	artist	length
190082	London Calling	2333	203826
190083	Brand New Cadillac	2333	130240

Relation „Album_Track“

id	album	track
190080	23746	190082
190081	23746	190083
190082	23746	190084
...		



Aufgabe 1c)

Geben Sie die Namen aller Alben aus, die einen Track enthalten, der länger als 100 Minuten ist.

$$\pi_{\text{albumname}}(\sigma_{[\text{length} > 1000 \cdot 60 \cdot 100]}((\text{Album_Track} \bowtie \beta_{[\text{album} \leftarrow \text{id}, \text{albumname} \leftarrow \text{name}]}(\text{Album})) \bowtie (\beta_{[\text{track} \leftarrow \text{id}, \text{trackname} \leftarrow \text{name}]}(\text{Track}))))$$



Aufgabe 1d)

Welche Alben enthalten keinen Track? Geben Sie Namen und IDs dieser Alben aus.

Datenbankschema: Alben und deren Tracks

Relation „Album“

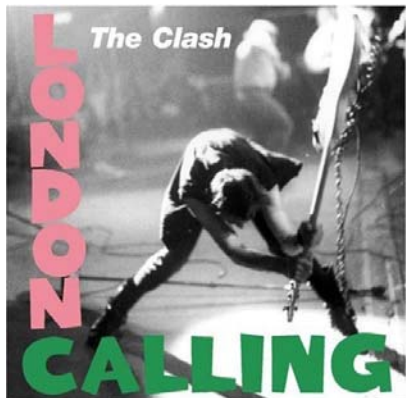
id	name	language
23746	London Calling	120

Relation „Track“

id	name	artist	length
190082	London Calling	2333	203826
190083	Brand New Cadillac	2333	130240

Relation „Album_Track“

id	album	track
190080	23746	190082
190081	23746	190083
190082	23746	190084
...		

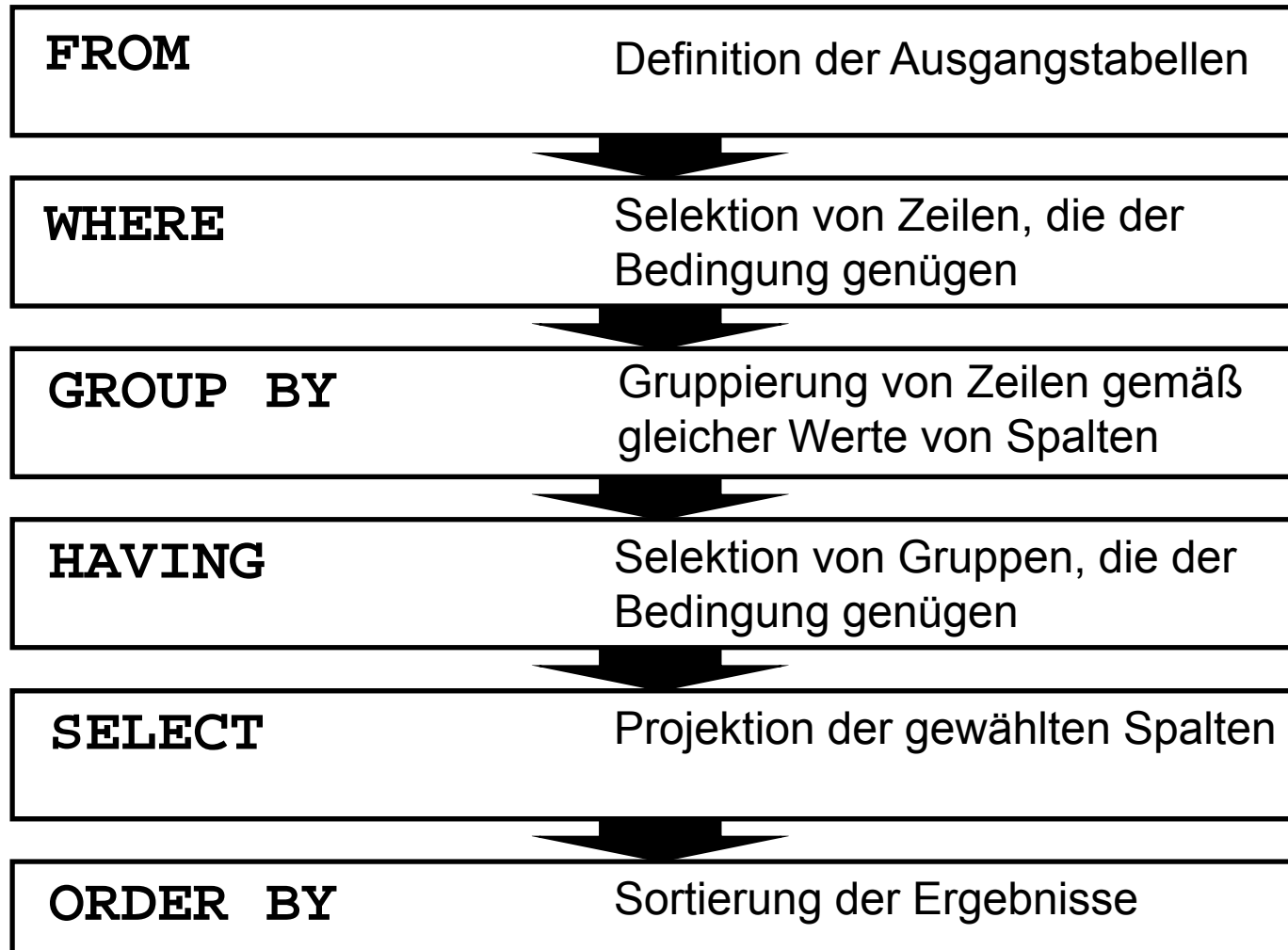


Aufgabe 1d)

Welche Alben enthalten keinen Track? Geben Sie Namen und IDs dieser Alben aus.

$$\pi_{\text{name, album}} \left(\begin{array}{l} \left(\left(\pi_{\text{album}} (\beta_{[\text{album} \leftarrow \text{id}]} (\text{Album})) \right) \right) \\ - \\ \left(\pi_{\text{album}} (\text{Album_Track}) \right) \\ \bowtie \\ \left(\beta_{[\text{album} \leftarrow \text{id}]} (\text{Album}) \right) \end{array} \right)$$

SQL: Reihenfolge





Web-Interface zur Datenbank

- URL: <http://i40virt00.ipd.uka.de/phpPgAdmin/>
 - Benutzer: student
 - Passwort: komdat2009
-
- Auch nach dieser Übung noch verfügbar.
 - Gutes Training für die Klausur!



Aufgabe 2a)

Geben Sie die Namen aller Künstler aus!

Datenbankschema: Relation „Artist“

id	name	type	begindate	enddate
92744	A Tribe Called Quest	2	1988-00-00	<i>NULL</i>
34778	David Hasselhoff	1	1952-07-17	<i>NULL</i>
233	Dire Straits	2	1977-00-00	1995-00-00
2327	Johann Sebastian Bach	1	1685-03-21	1750-07-28
815	John Lennon	1	1940-10-09	1980-12-08
387	Johnny Cash	1	1932-02-26	2003-09-12
303	The Beatles	2	1957-00-00	1970-04-10
2333	The Clash	2	1976-00-00	1986-00-00
435359	The Felice Brothers	2	2006-00-00	<i>NULL</i>
23	Tom Waits	1	1949-12-07	<i>NULL</i>



Aufgabe 2a)

Geben Sie die Namen aller Künstler aus!

```
SELECT DISTINCT(name)  
FROM Artist
```



Aufgabe 2b)

Gesucht sind die Namen aller schweizerdeutschen Alben
(Sprache mit der ID 476).

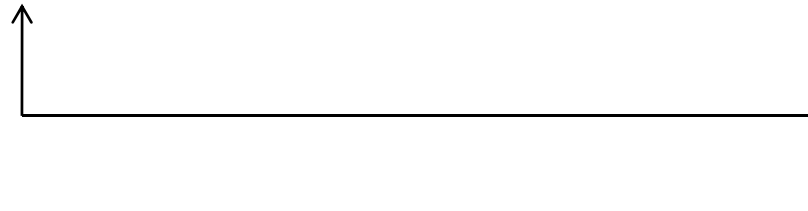
Datenbankschema: Sprachen

Relation „Language“

id	name
120	English
476	German, Swiss

Relation „Album“

id	name	language
2717	Abbey Road	120
22325	Sgt. Pepper's Lonely Hearts Club Band	120
70630	At Folsom Prison	120
370155	London Calling	120





Aufgabe 2b)

Gesucht sind die Namen aller schweizerdeutschen Alben
(Sprache mit der ID 476).

```
SELECT name  
FROM Album  
WHERE language = 476
```

Aufgabe 2b)

Gesucht sind die Namen aller schweizerdeutschen Alben (Sprache mit der ID 476).

name
Früsch vo de Läbere...
Nix wie Gäx
Gägsgüsi
Gigelisuppe
No Broblem
As isch nid immer alles crazy
I han es Zündhölzli azündt
Kriminalgschicht
...



Aufgabe 2c)

Geben Sie die Namen aller Alben aus, die einen Track enthalten, der länger als 100 Minuten ist.

Datenbankschema: Alben und deren Tracks

Relation „Album“

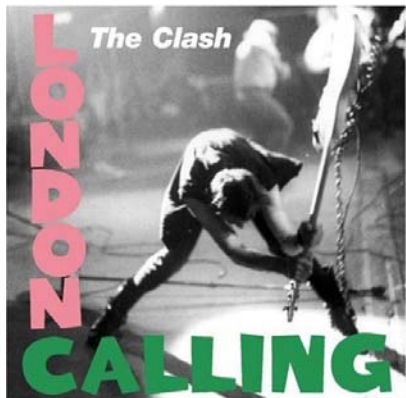
id	name	language
23746	London Calling	120

Relation „Track“

id	name	artist	length
190082	London Calling	2333	203826
190083	Brand New Cadillac	2333	130240

Relation „Album_Track“

id	album	track
190080	23746	190082
190081	23746	190083
190082	23746	190084
...		



Aufgabe 2c)

Geben Sie die Namen aller Alben aus, die einen Track enthalten, der länger als 100 Minuten ist.

```
SELECT DISTINCT(a.name)
FROM Album_Track at, Album a, Track t
WHERE at.album = a.id
AND at.track = t.id
AND t.length > (1000*60*100)
```



Aufgabe 2d)

Welche Alben enthalten keinen Track? Geben Sie Namen und IDs dieser Alben aus.

Datenbankschema: Alben und deren Tracks

Relation „Album“

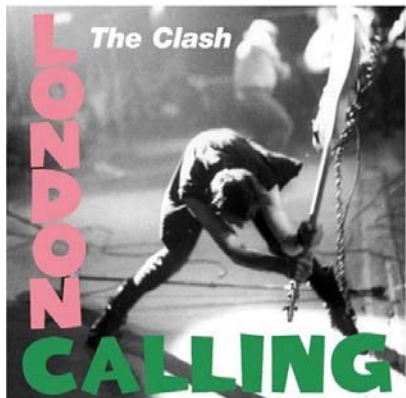
id	name	language
23746	London Calling	120

Relation „Track“

id	name	artist	length
190082	London Calling	2333	203826
190083	Brand New Cadillac	2333	130240

Relation
„Album_Track“

id	album	track
190080	23746	190082
190081	23746	190083
190082	23746	190084
...		



Aufgabe 2d)

Welche Alben enthalten keinen Track? Geben Sie Namen und IDs dieser Alben aus.

```
SELECT name, id
FROM Album
WHERE id NOT IN
      (SELECT DISTINCT(album)
       FROM Album_Track)
```



Aufgabe 3a)

Geben Sie die Anzahl der Künstler aus, deren Name mit der Zeichenkette „The “ („The“ gefolgt von einem Leerzeichen) beginnt.

Datenbankschema: Relation „Artist“

id	name	type	begindate	enddate
92744	A Tribe Called Quest	2	1988-00-00	<i>NULL</i>
34778	David Hasselhoff	1	1952-07-17	<i>NULL</i>
233	Dire Straits	2	1977-00-00	1995-00-00
2327	Johann Sebastian Bach	1	1685-03-21	1750-07-28
815	John Lennon	1	1940-10-09	1980-12-08
387	Johnny Cash	1	1932-02-26	2003-09-12
303	The Beatles	2	1957-00-00	1970-04-10
2333	The Clash	2	1976-00-00	1986-00-00
435359	The Felice Brothers	2	2006-00-00	<i>NULL</i>
23	Tom Waits	1	1949-12-07	<i>NULL</i>

Aufgabe 3a)

Geben Sie die Anzahl der Künstler aus, deren Name mit der Zeichenkette „The “ („The“ gefolgt von einem Leerzeichen) beginnt.

```
SELECT COUNT(*)  
FROM Artist  
WHERE name LIKE 'The %'
```



Aufgabe 3b)

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.

Datenbankschema: Tags

Relation „Album“

id	name	language
23746	London Calling	120

Relation „Tag“

id	name
7	rock
88	punk
133	classic pop and rock

70s 80s alt rock alternative alternative rock **british**
 british punk britpop clash classic classic punk **classic**
rock dub eighties electronic england english experimental
 favorite favorites favourites funk funky garage rock grunge
 guitar hard rock hardcore indie indie rock japanese jazz joe
 strummer london male vocalists metal new wave nice old school
 old school punk oldies political pop post-punk progressive rock
punk punk 77 **punk rock** punk-rock
 reggae **rock** rock 'n' roll rockabilly ska ska punk stoner rock
 the clash the only band that matters uk 77

album	tag
23746	7
23746	88
23746	133

Relation „Album_Tag“

Aufgabe 3b)

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.

```
(SELECT album
  FROM Album_Tag
  WHERE tag = 637)
INTERSECT
(SELECT album
  FROM Album_Tag
  WHERE tag = 150)
```



Aufgabe 3c)

Geben Sie anstatt der ID den Namen aller Alben aus Teilaufgabe b) aus.

3b) 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.

Aufgabe 3c)

Geben Sie anstatt der ID den Namen aller Alben aus Teilaufgabe b) aus.

3b) 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.

```
SELECT name FROM Album
WHERE id IN
    (
        (SELECT album FROM Album_Tag WHERE tag = 637)
        INTERSECT
        (SELECT album FROM Album_Tag WHERE tag = 150)
    )
```



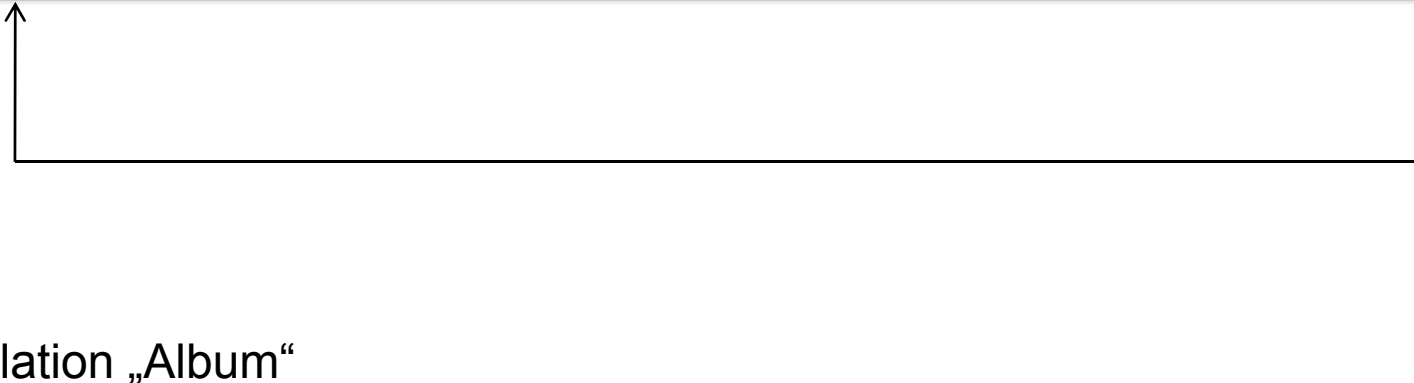
Aufgabe 3d)

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.

Datenbankschema: Künstler von Album

Relation „Artist“

id	name	type	begindate	enddate
303	The Beatles	2	1957-00-00	1970-04-10



Relation „Album“

id	name	language	artist
2717	Abbey Road	120	303
22325	Sgt. Pepper's Lonely Hearts Club Band	120	303

Aufgabe 3d)

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.

```
SELECT a.artist AS "Artist ID", b.name AS Artist,  
       a.name AS Album  
FROM Album a, Artist b  
WHERE a.artist = b.id  
GROUP BY a.artist, a.name, b.name  
HAVING COUNT(*) >= 7  
ORDER BY Artist
```

Aufgabe 3e)

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.

Korrektur: Join mit Relation „Album_Track“ unnötig!

Datenbankschema für Aufgabe 3e)

Relation „Artist“

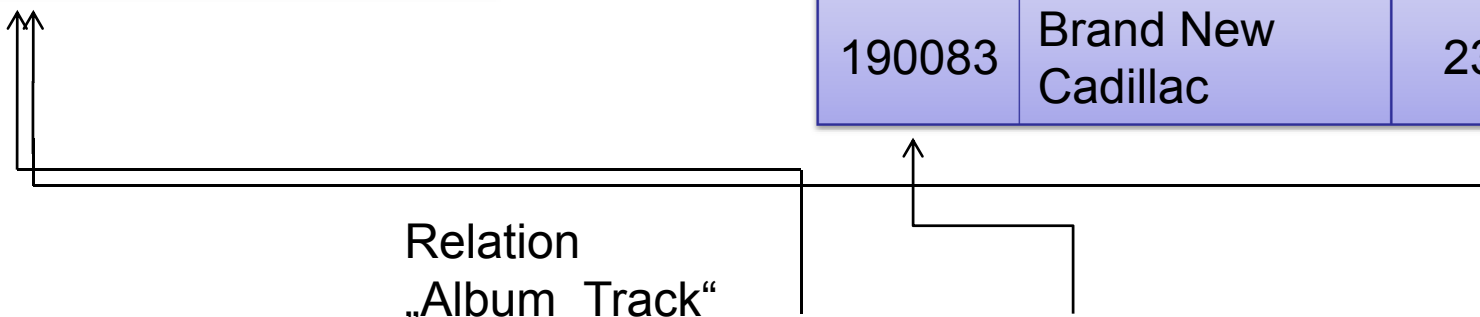
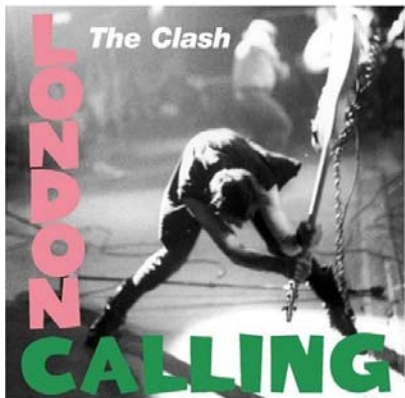
id	name
2333	The Clash

Relation „Track“

id	name	artist	length
190082	London Calling	2333	203826
190083	Brand New Cadillac	2333	130240

Relation „Album_Track“

id	album	track
190080	23746	190082
190081	23746	190083
190082	23746	190084
...		



Aufgabe 3e)

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.

```
SELECT a.id, a.name,  
       ROUND((AVG(length) / 1000 / 60), 2) as  
       "Average Track Length"  
FROM Track t, Artist a  
WHERE a.id = t.artist  
AND length != 0 AND length IS NOT NULL  
GROUP BY a.id, a.name  
HAVING COUNT(*) > 100  
ORDER BY "Average Track Length" DESC
```

Korrektur: Der Join über die Relation „Album_Track“ (aus der Lösung wie sie in der Übung auf Folie 48 vorgestellt wurde) ist unnötig.



Aufgabe 4a i)

Lösung über Join-Anfrage:

Welche Alben wurden mit mindestens einem Tag
versehen? Geben Sie die ID und den Namen dieser
Alben aus.

Datenbankschema: Tags

Relation „Album“

id	name	language
23746	London Calling	120

Relation „Tag“

id	name
7	rock
88	punk
133	classic pop and rock

70s 80s alt rock alternative alternative rock **british**
 british punk britpop clash classic classic punk **classic**
rock dub eighties electronic england english experimental
 favorite favorites favourites funk funky garage rock grunge
 guitar hard rock hardcore indie indie rock japanese jazz joe
 strummer london male vocalists metal new wave nice old school
 old school punk oldies political pop post-punk progressive rock
punk punk 77 **punk rock** punk-rock
 reggae **rock** rock 'n' roll rockabilly ska ska punk stoner rock
 the clash the only band that matters uk 77

album	tag
23746	7
23746	88
23746	133

Relation „Album_Tag“

Aufgabe 4a i)

Lösung über Join-Anfrage:

Welche Alben wurden mit mindestens einem Tag versehen? Geben Sie die ID und den Namen dieser Alben aus.

```
SELECT DISTINCT(a.id), a.name  
FROM Album a, Album_Tag at  
WHERE a.id = at.album
```

alternativ

```
SELECT DISTINCT(a.id), a.name  
FROM Album a JOIN Album_Tag at  
ON a.id = at.album
```



Aufgabe 4a ii)

Lösung über (NOT) IN-Anfrage:

Welche Alben wurden mit mindestens einem Tag versehen? Geben Sie die ID und den Namen dieser Alben aus.

Datenbankschema: Tags

Relation „Album“

id	name	language
23746	London Calling	120

Relation „Tag“

id	name
7	rock
88	punk
133	classic pop and rock

70s 80s alt rock alternative alternative rock **british**
 british punk britpop clash classic classic punk **classic**
rock dub eighties electronic england english experimental
 favorite favorites favourites funk funky garage rock grunge
 guitar hard rock hardcore indie indie rock japanese jazz joe
 strummer london male vocalists metal new wave nice old school
 old school punk oldies political pop post-punk progressive rock
punk punk 77 **punk rock** punk-rock
 reggae **rock** rock 'n' roll rockabilly ska ska punk stoner rock
 the clash the only band that matters uk 77

album	tag
23746	7
23746	88
23746	133

Relation „Album_Tag“

Aufgabe 4a ii)

Lösung über (NOT) IN-Anfrage:

Welche Alben wurden mit mindestens einem Tag versehen? Geben Sie die ID und den Namen dieser Alben aus.

```
SELECT id, name
FROM Album
WHERE id IN (
    SELECT Album
    FROM Album_Tag
)
```



Aufgabe 4a iii)

Lösung über (NOT) EXISTS-Anfrage:

Welche Alben wurden mit mindestens einem Tag versehen? Geben Sie die ID und den Namen dieser Alben aus.

Datenbankschema: Tags

Relation „Album“

id	name	language
23746	London Calling	120

Relation „Tag“

id	name
7	rock
88	punk
133	classic pop and rock

70s 80s alt rock alternative alternative rock **british**
 british punk britpop clash classic classic punk **classic**
rock dub eighties electronic england english experimental
 favorite favorites favourites funk funky garage rock grunge
 guitar hard rock hardcore indie indie rock japanese jazz joe
 strummer london male vocalists metal new wave nice old school
 old school punk oldies political pop post-punk progressive rock
punk punk 77 **punk rock** punk-rock
 reggae **rock** rock 'n' roll rockabilly ska ska punk stoner rock
 the clash the only band that matters uk 77

album	tag
23746	7
23746	88
23746	133

Relation „Album_Tag“

Aufgabe 4a iii)

Lösung über (NOT) EXISTS-Anfrage:

Welche Alben wurden mit mindestens einem Tag versehen? Geben Sie die ID und den Namen dieser Alben aus.

```
SELECT id, name
FROM Album a
WHERE EXISTS (
    SELECT *
    FROM Album_Tag at
    WHERE at.album = a.id
)
```



Aufgabe 4a iv)

Lösung über eine Anfrage mit Mengenoperationen:
Welche Alben wurden mit mindestens einem Tag
versehen? Geben Sie die ID und den Namen dieser
Alben aus.

Datenbankschema: Tags

Relation „Album“

id	name	language
23746	London Calling	120

Relation „Tag“

id	name
7	rock
88	punk
133	classic pop and rock

70s 80s alt rock alternative alternative rock **british**
 british punk britpop clash classic classic punk **classic**
rock dub eighties electronic england english experimental
 favorite favorites favourites funk funky garage rock grunge
 guitar hard rock hardcore indie indie rock japanese jazz joe
 strummer london male vocalists metal new wave nice old school
 old school punk oldies political pop post-punk progressive rock
punk punk 77 **punk rock** punk-rock
 reggae **rock** rock 'n' roll rockabilly ska ska punk stoner rock
 the clash the only band that matters uk 77

album	tag
23746	7
23746	88
23746	133

Relation „Album_Tag“

Aufgabe 4a iv)

Lösung über eine Anfrage mit Mengenoperationen:
Welche Alben wurden mit mindestens einem Tag versehen? Geben Sie die ID und den Namen dieser Alben aus.

```
SELECT id, name
FROM album
WHERE id IN (
    (SELECT id FROM Album)
    INTERSECT
    (SELECT album from Album_Tag)
)
```



Aufgabe 4b i)

Lösung über Join-Anfrage:

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 4b i)

Artist

id	name	type	begindate	enddate
34778	David Hasselhoff	1	1952-07-17	NULL
377499	Macaulay Culkin	1	1980-08-26	NULL

Track

id	name	artist	length
6534544	Looking for Freedom	34778	235000
7605384	Jump in My Car	34778	178000

LEFT JOIN vor Projektion/Selektion auf obigen Datensätzen:

id	name	type	begindate	enddate	id	name	artist	length
34778	D.H.	1	1952-07-17	NULL	6534544	Looking [...]	34778	235000
34778	D.H.	1	1952-07-17	NULL	7605384	Jump [...]	34778	178000
377499	M.C.	1	1980-08-26	NULL	NULL	NULL	NULL	NULL

Aufgabe 4b i)

id	name	type	begindate	enddate	id	name	artist	length
34778	D.H.	1	1952-07-17	NULL	6534544	Looking [...]	34778	235000
34778	D.H.	1	1952-07-17	NULL	7605384	Jump [...]	34778	178000
377499	M.C.	1	1980-08-26	NULL	NULL	NULL	NULL	NULL

```
FROM Artist a LEFT JOIN Track t
ON a.id = t.artist
WHERE [...] AND t.artist IS NULL
```

- Für jedes Tupel aus Artist wird nach Join-Partnern in der Relation Track gesucht ($a.id = t.artist$). Werden Tracks gefunden, enthält das Ergebnis für jeden Track zum Künstler eine Zeile mit dem Verbund.
- Wurde kein Track gefunden, so wird der Künstler dennoch ein mal in das Ergebnis übernommen. Diese Zeile enthält NULL-Werte für die Attribute aus der Relation Track.

Aufgabe 4b i)

Lösung über Join-Anfrage:

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.

```
SELECT a.name, begindate  
FROM Artist a LEFT JOIN Track t  
ON a.id = t.artist  
WHERE begindate IS NOT NULL  
AND enddate IS NULL  
AND t.artist IS NULL
```



Aufgabe 4b ii)

Lösung über (NOT) IN-Anfrage:

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.

Datenbankschema: Künstler zu Tracks

Relation „Artist“

id	name	type	begindate	enddate
34778	David Hasselhoff	1	1952-07-17	<i>NULL</i>



Relation „Track“

id	name	artist	length
6534544	Looking for Freedom	34778	235000
7605384	Jump in My Car	34778	178000

Aufgabe 4b ii)

Lösung über (NOT) IN-Anfrage:

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.

```
SELECT name, begindate
FROM Artist
WHERE begindate IS NOT NULL
AND enddate IS NULL
AND id NOT IN (
    SELECT DISTINCT(artist)
    FROM track
)
```



Aufgabe 4b iii)

Lösung über (NOT) EXISTS-Anfrage:

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.

Datenbankschema: Künstler zu Tracks

Relation „Artist“

id	name	type	begindate	enddate
34778	David Hasselhoff	1	1952-07-17	<i>NULL</i>



Relation „Track“

id	name	artist	length
6534544	Looking for Freedom	34778	235000
7605384	Jump in My Car	34778	178000

Aufgabe 4b iii)

Lösung über (NOT) EXISTS-Anfrage:

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.

```
SELECT name, begindate
FROM Artist a
WHERE begindate IS NOT NULL
AND enddate IS NULL
AND NOT EXISTS (
    SELECT *
    FROM Track
    WHERE artist = a.id
)
```



Aufgabe 4b iv)

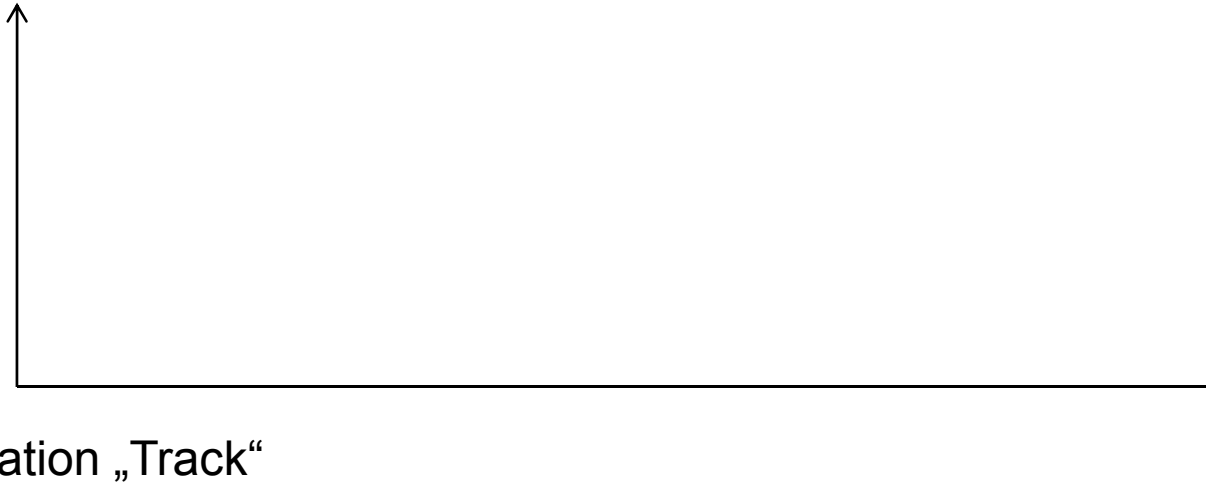
Lösung über Anfrage mit Mengenoperationen:

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.

Datenbankschema: Künstler zu Tracks

Relation „Artist“

id	name	type	begindate	enddate
34778	David Hasselhoff	1	1952-07-17	<i>NULL</i>



Relation „Track“

id	name	artist	length
6534544	Looking for Freedom	34778	235000
7605384	Jump in My Car	34778	178000

Aufgabe 4b iv)

Lösung über Anfrage mit Mengenoperationen:

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.

```
SELECT name, begindate
FROM Artist
WHERE id IN (
    (SELECT id FROM Artist a
     WHERE begindate IS NOT NULL
     AND enddate IS NULL)
    EXCEPT
    (SELECT artist FROM Track)
)
```

Aufgabe 5a)

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)
```

Wo liegt der Syntax/Semantik-Fehler?

- Rückgabewert der durch die Unteranfragen verknüpften Anfragen stimmt nicht überein!
- id = numerische ID, name = String

Aufgabe 5a)

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

Richtig:

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

Aufgabe 5b)

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
```

Wo liegt der Syntax/Semantik-Fehler?

- Durch HAVING werden Einschränkungen auf durch GROUP BY gebildete Gruppen formuliert
- GROUP BY-Klausel fehlt

Aufgabe 5b)

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

Richtig:

```
SELECT name, COUNT(*)  
FROM Artist  
GROUP BY name  
HAVING COUNT(*) > 1
```

Aufgabe 5c)

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')
```

Wo liegt der Syntax/Semantik-Fehler?

- Unteranfrage wird für jedes Tupel der Überanfrage ausgewertet. Unteranfrage liefert hier immer etwas zurück => EXISTS immer wahr. Daher enthält das Ergebnis **alle** Alben!
- Unkorrelierte EXISTS-Anfragen meistens sinnlos oder falsch!

Aufgabe 5c)

Geben Sie ID und Namen aller Alben zurück, die mit dem Tag „motown“ versehen wurden.

Richtig:

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

Aufgabe 5d)

Geben Sie die ID aller Alben zurück, die nicht mit dem Tag „metal“ versehen wurden.

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

Wo liegt der Syntax/Semantik-Fehler?

- In der Aufgabenstellung nur Name des Tags genannt => Join mit Relation „Tag“ nötig.
- Relation „Album_Tag“ enthält für ein Album möglicherweise mehrere Tags => Überprüfung in WHERE-Klausel unzureichend.

Aufgabe 5d)

Geben Sie die ID aller Alben zurück, die nicht mit dem Tag „metal“ versehen wurden.

Richtig:

```
SELECT id
FROM Album
WHERE id NOT IN (
    SELECT DISTINCT(at.album)
    FROM Album_Tag at, Tag t
    WHERE at.tag = t.id
    AND t.name = 'metal'
)
```

Aufgabe 5e)

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
```

Wo liegt der Syntax/Semantik-Fehler?

- In SELECT-Klausel nur Aggregatfunktionen oder Attribute nach denen gruppiert wurde.
- In Ausgangsrelation wird jede Gruppe nur durch ein Tupel repräsentiert.

Aufgabe 5e)

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

Richtig:

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

Worte zur Klausur

- Aufgabenstellung **genau** lesen!
 - Was soll die Anfrage zurückliefern?
 - Welche Relationen brauche ich?
- Nicht mehr machen als verlangt → **Zeit!**
- Auf Korrektheit der Syntax achten