# 1 musikDB

| AREA_TYPE | | | |
|---|---|---|---|
| id | name | id | name |
| 1 | Country | 5 | District |
| 2 | Subdivision | 6 | Island |
| 3 | City | 7 | County |
| 4 | Municipality | | |

| AREA | | |
|---|---|---|
| id | name | type |
| 432 | England | 1 |
| 1178 | London | 3 |
| 81 | Germany | 1 |
| 334 | Nordrhein-Westfahlen | 2 |
| 7709 | Düsseldorf | 3 |

| AREA_IN | | |
|---|---|---|
| | Contains | Within other |
| id | entity0 | entity1 |
| 964 | 432 | 1178 |
| 3587 | 432 | 3821 |
| 71 | 81 | 334 |
| 7475 | 334 | 7709 |

| PLACE | | | |
|---|---|---|---|
| id | name | type | area |
| 2418 | The Crystal Palace | 3 | 1178 |
| 2387 | BBC Radio Theatre | 1 | 1178 |
| 789 | ESPRIT area | 4 | 7709 |

| ARTIST_ARTIST | | | | | |
|---|---|---|---|---|---|
| | Relationship | Person | Band | | |
| id | r_type | entity0 | entity1 | begin_year | end_year |
| 32164 | member of band | 285850 | 1352 | 1974 | 1980 |
| 164036 | member of band | 290121 | 1352 | 1973 | |
| 54623 | sibling | 290121 | 316942 | | |

r_type possibilities

- is person
- sibling
- married
- member of band
- parent
- founder

| ARTIST | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | birth | | | death | | | | |
| id | name | b_year | b_month | b_day | e_year | type | area | gender | b_area |
| 938 | Eric Clapton | 1945 | 3 | 30 | | p | 221 | m | 53564 |
| 1352 | AC/DC | 1973 | 11 | | | g | 13 | | 5126 |
| 290121 | Angus Young | 1955 | 3 | 31 | | p | 80658 | m | 3855 |
| 341872 | Doro Pesch | 1964 | 6 | 3 | | p | 81 | f | 7709 |
| 285850 | Bon Scott | 1946 | 7 | 9 | 1980 | p | 221 | m | 80661 |

| RELEASE | | | | | | |
|---|---|---|---|---|---|---|
| release_id | name | artist_credit | country | date_year | date_month | date_day |
| 1610822 | Back in Black | 1352 | 81 | 1986 | | |
| 1085503 | Back in Black | 1352 | 13 | 1989 | | |
| 1386549 | Atemlos durch die Nacht | 397683 | 81 | 2013 | 11 | 29 |

| TRACK | | | | | |
|---|---|---|---|---|---|
| id | release_id | name | position | lenght | artist_credit |
| 14242945 | 1085503 | Hells Bells | 1 | 312000 | 1352 |
| 18709752 | 1610822 | Hells Bells | 1 | 311866 | 1352 |
| 18709759 | 1610822 | Have a Drink on Me | 8 | 238400 | 1352 |
| 20084778 | 1722097 | I Lost My Heart in Heidelberg | 18 | 198040 | 35001 |

# 2 SQL

EXISTS

```
SELECT artist.name FROM artist
WHERE EXISTS (SELECT track.name FROM track
WHERE track.artist_credit = artist.id
AND track.lenght BETWEEN 7200000 AND 14400000
AND artist.type = 'g')
```

| | name |
|---|---|
| 1 | Exodus |

String Functions Im Umgang mit Wörtern ist es am sinnvollsten wenn man **LIKE** statt = verwendet, da wenn man so was wie '%a%' haben möchte sucht = nach genau dieser Zeichenfolge und LIKE nach einem a mit sämtlichen Zeichen vor und nach einem a.
**UPPER** und **LOWER** sollten selbsterklärend sein
Die Zeichen % und _ sind jeweils 'Wildcards' sprich da kann jeder Buchstabe hin. % ist für 0 bis $\infty$ Charaktere und _ für genau ein Charakter

HAVING + ORDER BY + GROUP BY

```
SELECT area.name, COUNT(artist.name) FROM artist, area
WHERE artist.b_area = area.id
GROUP BY area.name
HAVING COUNT(artist.name) > 1000
ORDER BY area.name ASC
```

| | ↓ GROUP BY | |
|---|---|---|
| | name | count |
| 1 | London | 2094 |
| 2 | Los Angeles | 1319 |
| 3 | New York | 1403 |
| 4 | Tokyo | 1157 |

ORDER BY ASC

WITH AS + LIMIT

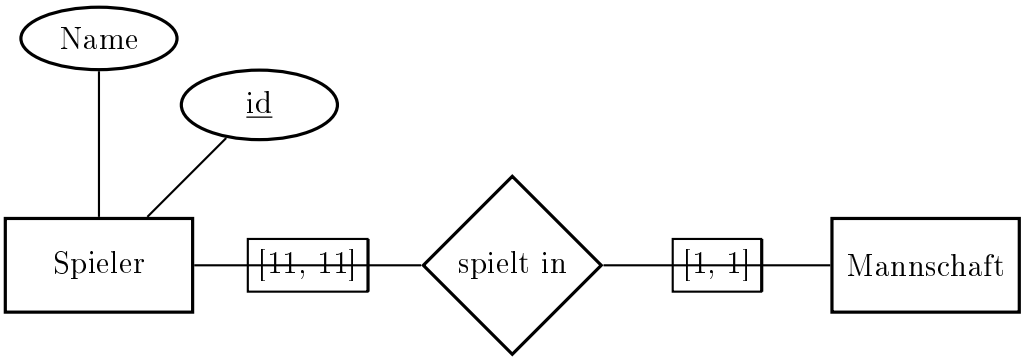
```
WITH tmp AS (
SELECT artist.name as artist, r.name as album
FROM artist, release as r
WHERE r.artist_credit = artist.id )
SELECT * FROM tmp
LIMIT 1
```

| | artist | album |
|---|---|---|
| 1 | Damn Seagulls | Soul Politics |

## 2.1 DDL/DML

```
create Table Unikurs(
KursID int,
Name varchar(30) not null,
AnzahlStudis int check (AnzahlStudis >=5),
primary key(KursID)
foreign key(Name) references Vorlesung(Name)
);
    insert into unikurs
values(0, 'LA1', 777)
    insert into unikurs (KursID, Name)
values(1, 'Ana1')
```

# 3 ER-Schema




# 4 Relationale Algebra

Umbenennung $\beta$

Ganze Tabelle: $\beta_b$(Band). Einzelne Spalte: $\beta_{ph \leftarrow Photo}$(R)

Gruppierung

$\gamma_{A,sum(a)}$(R) Gruppiert nach A mit der Summe über a (oder andere Aggregat Funktionen)

min/max

min: $\Pi_a$(R) - $(\beta_{a \leftarrow a2}(\Pi_{a2}(\sigma_{a1<a2}(\beta_{a1 \leftarrow a}(R) \times \beta_{a2 \leftarrow a}(R)))))$ $\Leftrightarrow$ $\gamma_{A,min(a)}$(R) - Man selektiert alle Elemente die noch ein kleineres Element haben und zieht dann alle anderen ab

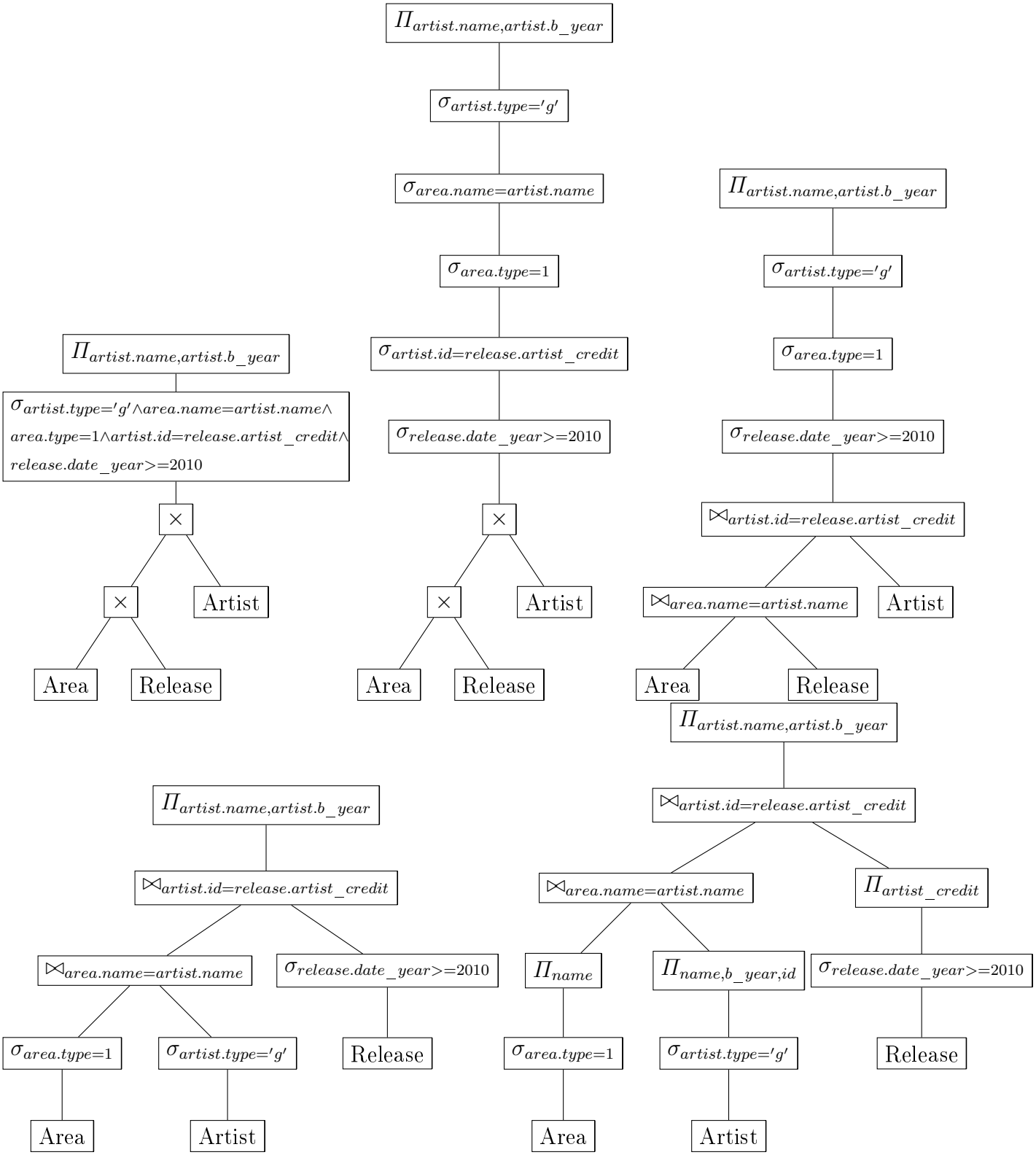
max: $\Pi_a$(R) - $(\beta_{a \leftarrow a1}(\Pi_{a2}(\sigma_{a1<a2}(\beta_{a1 \leftarrow a}(R) \times \beta_{a2 \leftarrow a}(R)))))$ $\Leftrightarrow$ $\gamma_{A,max(a)}$(R) - Man selektiert alle Elemente die noch ein größeres Element haben und zieht dann alle anderen ab

Von SQL zu Relationale Algebra

SELECT DISTINCT artist.name,
artist.b_year
FROM area, release, artist
WHERE artist.type = 'g'
AND area.name = artist.name
AND area.type = 1;

$\Pi_{artist.name,artist.b\_year}(\sigma_{artist.type='g' \wedge area.name=artist.name \wedge area.type=1 \wedge artist.id=release.artist_credit \wedge release.date_year>=2010}$
$(AREA \times RELEASE \times ARTIST))$

Operator Baum


$\Pi_{artist.name,artist.b\_year}$
$\sigma_{artist.type='g'}$
$\sigma_{area.name=artist.name}$
$\sigma_{area.type=1}$
$\sigma_{artist.id=release.artist\_credit}$
$\sigma_{release.date\_year>=2010}$
×
×
Artist
Area
Release
$\Pi_{artist.name,artist.b\_year}$
$\sigma_{artist.type='g'\wedge area.name=artist.name\wedge}$
$area.type=1\wedge artist.id=release.artist\_credit\wedge$
$release.date\_year>=2010$
×
×
Artist
Area
Release
$\Pi_{artist.name,artist.b\_year}$
$\sigma_{artist.type='g'}$
$\sigma_{area.type=1}$
$\sigma_{release.date\_year>=2010}$
$\bowtie_{artist.id=release.artist\_credit}$
$\bowtie_{area.name=artist.name}$
Artist
Area
Release
$\Pi_{artist.name,artist.b\_year}$
$\bowtie_{artist.id=release.artist\_credit}$
$\bowtie_{area.name=artist.name}$
$\sigma_{release.date\_year>=2010}$
$\sigma_{area.type=1}$
$\sigma_{artist.type='g'}$
Area
Artist
Release
$\Pi_{artist.name,artist.b\_year}$
$\bowtie_{artist.id=release.artist\_credit}$
$\bowtie_{area.name=artist.name}$
$\Pi_{artist\_credit}$
$\Pi_{name}$
$\Pi_{name,b\_year,id}$
$\sigma_{release.date\_year>=2010}$
$\sigma_{area.type=1}$
$\sigma_{artist.type='g'}$
Area
Artist
Release

# 5 Tupelkalküle

Name und Wohnort aller Mitarbeiter, welche in der Forschung arbeiten
{<m.name, m.wohnort> | m ∈ Mitarbeiter ∧∃$a$ ∈ Abteilung (a.abtname = 'Forschung' ∧ a.nr = m.abtnr)}

# 6 Formeln

## 6.1 Selectivität

Erwartete größe der resultierenden Tabelle:
sel(P, R) = $\frac{|\sigma_P(R)|}{|R|}$, sel(B ⋈ C) = $\frac{|B \bowtie C|}{|B|*|C|}$, sel = $\frac{\text{Output - Relation}}{\text{Input - Relation}}$

Erwartete größe der Suche:
sel(A = v, R) = $\frac{1}{val_{A,r}}$, sel(A < v, R) = $\frac{v - A_{min}}{A_{max} - A_{min}}$, sel(A > v, R) = $\frac{A_{max} - v}{A_{max} - A_{min}}$

Projektion ohne Duplikateliminierung: |r| = |$\Pi$(r)| ⇒ sel = 1
Mit Duplikateliminierung: |$\Pi_k$(r)| = $val_{k,r}$
Spezialfälle sind: Keine gemeinsamen Werte |r ⋈ s| = 0, Fremdschlüsselbeziehung (r → s): |r ⋈ s| = |r|, Im Verbundsattribut ist nur ein verschiedenes Element: |r ⋈ s| = |r * s|, sonst |r ⋈ s| = $\frac{|r|*|s|}{\max\{val_{B,r}, val_{B,s}\}}$

## 6.2 B+Bäume

d := Daten, k := (Einordnungs-)Attribut, s := Seite (=Knoten), z := Zeiger, n := Anzahl Datensätze, x := Zellen inneres Knoten, y := Zellen Blattknotens

Ordnung des Blattknoten i
i = $\lfloor \frac{s-2z}{2(d+k)} \rfloor$

Ordnung des Innerknotens j
j = $\lfloor \frac{s-z}{2(d+k)} \rfloor$

Höhe
$1 + \lceil \log_{2x+1}(\frac{n}{2y}) \rceil \leq n \leq 1 + \lceil \log_{2x+1}(\frac{n}{y}) \rceil$

# 7 Anfrage Optimierung

Variablen

| | |
|---|---|
| $b_{size}$ | Blockgröße |
| mem | Puffergröße (in Anzahl Blöcken) |
| \|r\| | Anzahl Tupel in Relation r |
| $b_r$ | Anzahl Blöcke für Tupel aus r |
| $size_r$ | Mittlerer Tupel-Größe |
| $f_r$ | Blockungsfaktor $\frac{b_{size}}{size_r}$ - Wie viele Relationen in einen Block passen |
| $val_{A,r}$ | Anzahl verschiedener Werte für A in r |
| $lev_{I(R(A))}$ | Anzahl Indexebenen im $B^+$Baum für Index auf R(A) |

Externes Sortieren

Anzahl Mischläufe: $\lceil \log_{mem-1}(\frac{b_r}{mem}) \rceil$

Blockzugriff pro Mischlauf: $(L + S) * b_r * (1 + \lceil \log_{mem-1}(\frac{b_r}{mem}) \rceil)$

Bottleneck bei Anfragebearbeitung sind immer Blockzugriffe. Operationen im Hauptspeicher werden vernachlässigt.

Scans

Scans durchlaufen alle Tupel/Sucht nach einem bestimmten Tupel mit den entsprechenden Bedingungen

Kosten eines Relationsscans: $b_r$

Kosten eines Indexscans: $lev_{I(R(A))} + \frac{|r|}{val_{A,r}}$

Join-Methods

Block-Nested-Loop-Join: liest so viele Blöcke von r ein, dass der Buffer fast voll ist (mem-1). In dem letzten Buffer Slot wird Sequenziell jeder Block von S eingelesen. Dies wird mit r gejoined in den Hauptspeicher wieder gehashed. Kosten sind

$$b_r + \lceil \frac{b_r}{mem - 1} \rceil * b_s$$

Merge-Join: Zunächst werden beide Tabellen nach Join-Attribut sortiert. Dann werden beide linear durchlaufen und gemerged. Kosten sind

$[sortieren] + b_r + b_s$ wobei das Sortieren abgeschätzt werden kann mit $b_x \log_{mem} b_x$

Classic-Hash-Join: Die ganze Relation r wird durchgegangen und ein block nach dem anderen gehashed. Nach jedem hash schritt wird überprüft, ob ein block aus s zusammen gesetzt werden kann, mit der selben hash Funktion. Kosten:

$$b_r + p * b_s \text{ mit } p = \lceil \frac{b_r}{mem - 1} \rceil$$

# 8 Codd'schen Regeln

1. Integration: Einheitliche nicht redundante Datenverwaltung
2. Operationen: Speicher, Suchen, Ändern
3. Katalog: Zugriff auf Datenbankbeschreibung im Data Dictionary
4. Benutzerschichten
5. Integritätssicherung: Korrektur des Datenbankinhaltes
6. Datenschutz: Ausschluss unautorisierter Zugriffe
7. Synchronisation: Parallele Transaktionen Koordinierung
8. Datensicherung: Wiederherstellung von Daten nach Systemfehlern

# 9 B+ Bäume

Eingabe = Dallas, Miami, Memphis, Atlanta, Phoenix, Portland, Detroit, Austin, Columbus, Houston, Chicago, Philadelphia
m = 1