
Technical Report CS-01-18
Contract-based Query Rewriting under Privacy
Constraints
Hannes Grunert

Universität Rostock
Fakultät für Informatik und Elektrotechnik
Institut für Informatik
Lehrstuhl für Datenbank- und Informationssysteme



Additional material for VLIoT @ VLDB 2018

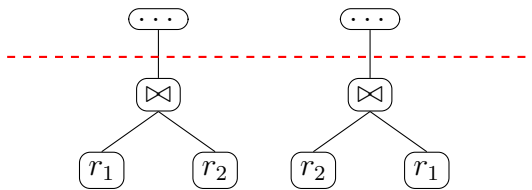
Contract-based Query Rewriting under Privacy Constraints

Hannes Grunert
 Database Research Group
 University of Rostock
 18051 Rostock
 (hs|nf)(at)informatik.uni-rostock.de

1 Classic Rules

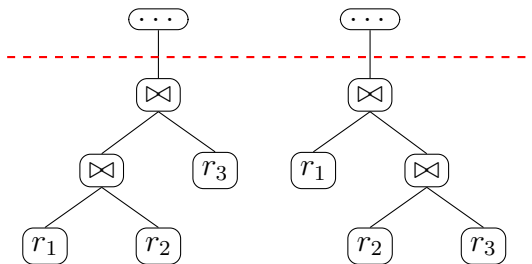
K0-1: Kommutativität von Joins

Regel: $r_1 \bowtie r_2 \equiv r_2 \bowtie r_1$
Invariante: —
Vorbedingungen: —
Nachbedingungen: —
Anmerkung: Wird nur für interne Vertauschungen benutzt.



K0-2: Assoziativität von Joins

Regel: $(r_1 \bowtie r_2) \bowtie r_3 \equiv r_1 \bowtie (r_2 \bowtie r_3)$
Invariante: —
Vorbedingungen: —
Nachbedingungen: —
Anmerkung: Wird nur für interne Vertauschungen benutzt.



K1: Entfernen redundanter Operationen

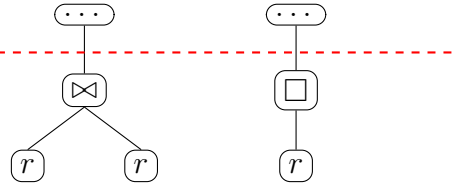
Regel: $r \bowtie r \equiv r$

Invariante: —

Vorbedingungen: \bowtie wird nicht unterstützt

Nachbedingungen: —

1	BEGIN;
2	SELECT t1.id, t1.length
3	FROM tracks t1 JOIN tracks t2
	ON (t1.id=t2.id)
4	COMMIT;
<hr/>	
1	BEGIN;
2	SELECT t1.id, t1.length
3	FROM tracks t1
4	COMMIT;



K2: Kommutativität von Selektion und Join

Regel: $\sigma_Y(\sigma_X(r_1) \bowtie r_2) \sqsubseteq_K \sigma_{X \wedge Y}(r_1 \bowtie r_2)$

Invariante: $\text{attr}(X) \subseteq r_1$

Vorbedingungen: σ_X wird nicht unterstützt

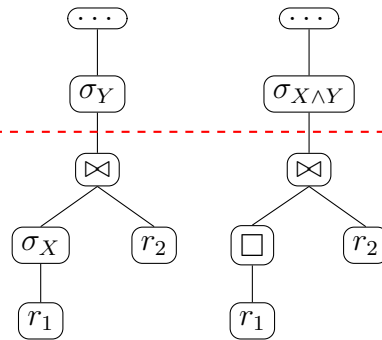
Nachbedingungen: —

```

1 BEGIN;
2 SELECT tracks.title, albums.
  name
3 FROM tracks JOIN albums ON (
  tracks.album = albums.id)
4 WHERE albums.name LIKE '
  Distant Worlds%' AND length >
  250000;
5 COMMIT;

1 BEGIN;
2 WITH X AS (
3   SELECT tracks.album, tracks
  .title
4   FROM tracks
5   WHERE length > 250000
6 ),
7 Y AS (
8   SELECT X.title, albums.name
9   FROM X JOIN albums ON (X.
  album = albums.id)
10 )
11 SELECT *
12 FROM Y
13 WHERE name LIKE 'Distant
  Worlds%';
14 COMMIT;

```



K3: Kommutativität von Projektion und Verbund – V1

Regel: $\pi_X(\pi_X(r_1) \bowtie r_2) \sqsubseteq_K \pi_X(r_1 \bowtie r_2)$

Invariante: $\text{attr}(X) \subseteq r_1$

Vorbedingungen: π_X wird nicht unterstützt

Nachbedingungen: —

```

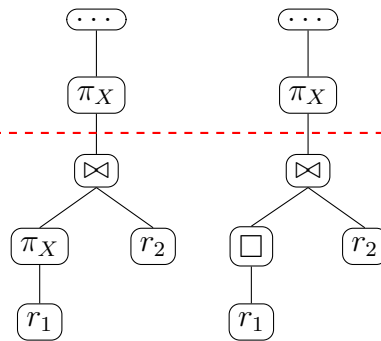
1 BEGIN;
2 WITH X AS (
3   SELECT id AS tid, album,
4     title
5   FROM tracks
6 ),
7 Y AS (
8   SELECT *
9   FROM X JOIN albums ON (X.
10    album = albums.id)
11 )
12 SELECT tid, album, title
13 COMMIT;

```

```

1 BEGIN;
2 WITH X AS (
3   SELECT *
4   FROM tracks
5 ),
6 Y AS (
7   SELECT X.id AS tid, *
8   FROM X JOIN albums ON (X.
9    album = albums.id)
10 )
11 SELECT tid, album, title
12 FROM Y;
13 COMMIT;

```



K4: Kommutativität von Projektion und Verbund – V2

Regel: $\pi_Y(r_1) \bowtie \pi_Z(r_2) \sqsubseteq_K \pi_X(r_1 \bowtie \pi_Z(r_2))$

Invariante: $X := Y \cup Z$

Vorbedingungen: π_Y wird nicht unterstützt,

$Y \cap Z \neq \emptyset \vee R(r_1) \cap R(r_2) = \emptyset$

Nachbedingungen: —

```

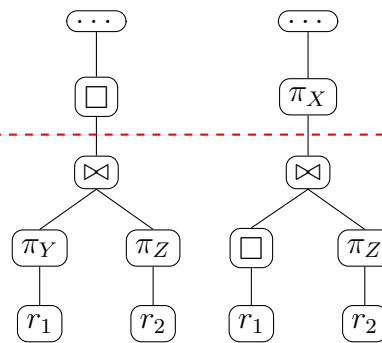
1 BEGIN;
2 WITH X AS (
3   SELECT id AS tid, album,
4     title
5   FROM tracks
6 ),
7 Y AS (
8   SELECT albums.id AS aid ,
9     name
10  FROM albums
11 ),
12 Z AS (
13   SELECT *
14   FROM X JOIN Y ON (X.album =
15     Y.aid)
16 )
17 SELECT *
18 FROM Z;
19 COMMIT;

```

```

1 BEGIN;
2 WITH X AS (
3   SELECT *
4   FROM tracks
5 ),
6 Y AS (
7   SELECT X.id AS tid, *
8   FROM X JOIN albums ON (X.
9     album = albums.id)
10 )
11 SELECT tid, album, title
12 FROM Y;
13 COMMIT;

```



K5: Dominanz der äußeren Projektion

Regel: $\pi_X(\pi_Y(r)) \sqsubseteq_K \pi_X(r)$

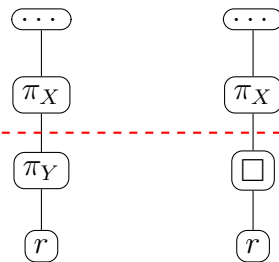
Invariante: $X \subseteq Y$

Vorbedingungen: π_Y wird nicht unterstützt

Nachbedingungen: —

```
1 BEGIN;  
2 WITH X AS (  
3   SELECT id AS tid, album,  
4     title  
5   FROM tracks  
6 ),  
7 Y AS (  
8   SELECT albums.id AS aid ,  
9     name  
10  FROM albums  
11 ),  
12 Z AS (  
13   SELECT *  
14   FROM X JOIN Y ON (X.album =  
15     Y.aid)  
16 )  
17 SELECT *  
18 FROM Z;  
19 COMMIT;
```

```
1 BEGIN;  
2 WITH X AS (  
3   SELECT *  
4   FROM tracks  
5 ),  
6 Y AS (  
7   SELECT X.id AS tid, *  
8   FROM X JOIN albums ON (X.  
9     album = albums.id)  
10 )  
11 SELECT tid, album, title  
12 FROM Y;  
13 COMMIT;
```



K6: Kommutativität von zwei Selektionen – V1

Regel: $\sigma_X(\sigma_Y(r)) \sqsubseteq_K \sigma_{X \wedge Y}(r)$

Invariante: —

Vorbedingungen: σ_Y wird nicht unterstützt

Nachbedingungen: —

```

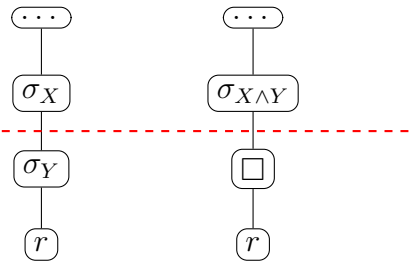
1 BEGIN;
2 WITH X AS (
3   SELECT *
4   FROM tracks
5   WHERE title LIKE '%song%'
6 )
7
8 SELECT *
9 FROM X
10 WHERE length > 250000;
11 COMMIT;

```

```

1 BEGIN;
2 WITH X AS (
3   SELECT *
4   FROM tracks
5 )
6
7 SELECT *
8 FROM X
9 WHERE title LIKE '%song%' AND
10 length > 250000;
11 COMMIT;

```



K7: Kommutativität von zwei Selektionen – V2

Regel: $\sigma_{X \wedge Y}(r) \sqsubseteq_K \sigma_Y(\sigma_X(r))$

Invariante: —

Vorbedingungen: σ_Y oder \wedge wird nicht unterstützt

Nachbedingungen: σ_X wird unterstützt

```

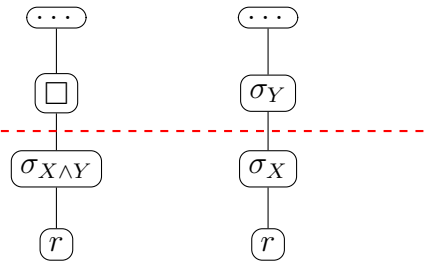
1 BEGIN;
2 WITH X AS (
3   SELECT *
4   FROM tracks
5   WHERE title LIKE '%song%'
6 )
7
8 SELECT *
9 FROM X
10 WHERE length > 250000;
11 COMMIT;

```

```

1 BEGIN;
2 WITH X AS (
3   SELECT *
4   FROM tracks
5 )
6
7 SELECT *
8 FROM X
9 WHERE title LIKE '%song%' AND
10 length > 250000;
11 COMMIT;

```



K8: Kommutativität von zwei Selektionen – V3

Regel: $\sigma_X(\sigma_Y(r)) \sqsubseteq_K \sigma_Y(\sigma_X(r))$

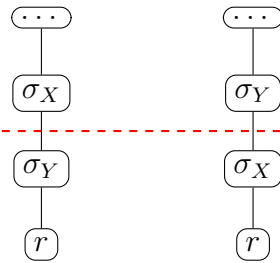
Invariante: —

Vorbedingungen: σ_Y wird nicht unterstützt

Nachbedingungen: σ_X wird unterstützt

```
1 BEGIN;  
2 WITH X AS (  
3   SELECT *  
4   FROM tracks  
5   WHERE length > 250000  
6 )  
7  
8 SELECT *  
9 FROM X  
10 WHERE title LIKE '%song%';  
11 COMMIT;
```

```
1 BEGIN;  
2 WITH X AS (  
3   SELECT *  
4   FROM tracks  
5   WHERE title LIKE '%song%'  
6 )  
7  
8 SELECT *  
9 FROM X  
10 WHERE length > 250000;  
11 COMMIT;
```



K9: Kommutativität von Selektion und Projektion – V1

Regel: $\sigma_Y(\pi_X(r)) \sqsubseteq_K \pi_X(\sigma_Y(r))$

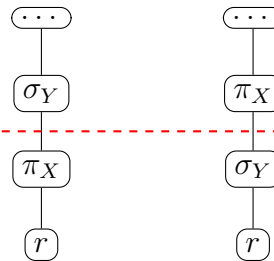
Invariante: —

Vorbedingungen: π_X wird nicht unterstützt

Nachbedingungen: σ_Y wird unterstützt

```
1 BEGIN;  
2 WITH X AS (  
3   SELECT title, length  
4   FROM tracks  
5 )  
6  
7 SELECT *  
8 FROM X  
9 WHERE length > 250000;  
10 COMMIT;
```

```
1 BEGIN;  
2 WITH X AS (  
3   SELECT *  
4   FROM tracks  
5   WHERE length > 250000  
6 )  
7  
8 SELECT title, length  
9 FROM X;  
10 COMMIT;
```



K10: Kommutativität von Selektion und Projektion – V2

Regel: $\pi_Y(\sigma_X(r)) \sqsubseteq_K \sigma_X(\pi_Y(r))$

Invariante: —

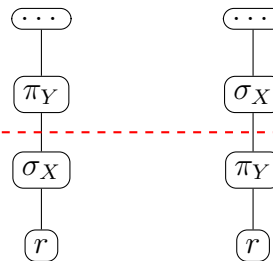
Vorbedingungen: σ_X wird nicht unterstützt,

$attr(X) \subseteq Y$

Nachbedingungen: π_Y wird unterstützt

```
1 BEGIN;
2 WITH X AS (
3   SELECT *
4   FROM tracks
5   WHERE title LIKE '%song%'
6 )
7
8 SELECT title, length
9 FROM X;
10 COMMIT;
```

```
1 BEGIN;
2 WITH X AS (
3   SELECT title, length
4   FROM tracks
5 )
6
7 SELECT *
8 FROM X
9 WHERE title LIKE '%song%';
10 COMMIT;
```



K11: Kommutativität von Selektion und Projektion – V3

Regel: $\pi_X(\sigma_Z(\pi_{X,Y}(r))) \subseteq_K \pi_X(\sigma_Z(r))$

Invariante: —

Vorbedingungen: $\pi_{X,Y}$ wird nicht unterstützt,

$attr(Z) \subseteq X \cup Y$

Nachbedingungen: —

```

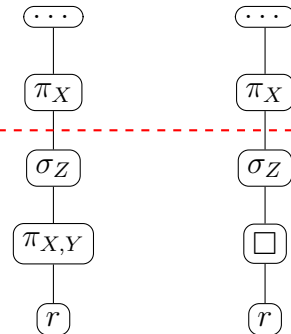
1 BEGIN;
2 WITH X AS (
3   SELECT title, length
4   FROM tracks
5 ), Y AS (
6   SELECT *
7   FROM X
8   WHERE length < 250000
9 )
10
11 SELECT title
12 FROM Y;
13 COMMIT;

```

```

1 BEGIN;
2 WITH X AS (
3   SELECT *
4   FROM tracks
5   WHERE length < 250000
6 )
7
8 SELECT title
9 FROM X;
10 COMMIT;

```



K12: Kommutativität von Selektion und Projektion – V4

Regel: $\pi_X(\sigma_Z(r)) \quad \sqsubseteq_K$
 $\pi_X(\sigma_Z(\pi_{X,Y}(r)))$

Invariante: —

Vorbedingungen: σ_Z wird nicht unterstützt

Nachbedingungen: $\pi_{X,Y}$ wird unterstützt,

$Y := attr(Z) \setminus X$

```

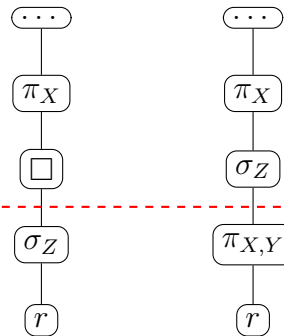
1 BEGIN;
2 WITH X AS (
3   SELECT *
4   FROM tracks
5   WHERE title LIKE '%song%'
6 )
7
8 SELECT title
9 FROM X;
10 COMMIT;

```

```

1 BEGIN;
2 WITH X AS (
3   SELECT title, length
4   FROM tracks
5 ), Y AS (
6   SELECT *
7   FROM X
8   WHERE title LIKE '%song%'
9 )
10
11 SELECT title
12 FROM Y;
13 COMMIT;

```



K13: Kommutativität von Selektion und Mengenvereinigung – V1

Regel: $\sigma_X(r_1 \cup r_2) \sqsubseteq_K \sigma_X(r_1) \cup \sigma_X(r_2)$

Invariante: —

Vorbedingungen: \cup wird nicht unterstützt

Nachbedingungen: σ_X wird unterstützt

```

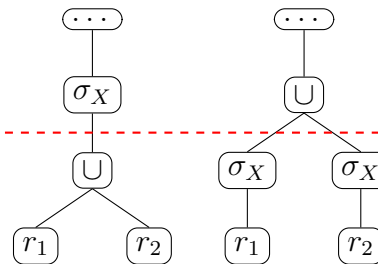
1 BEGIN;
2 WITH X AS (
3   (SELECT name
4    FROM composers) UNION
5   (SELECT name
6    FROM artists)
7 )
8
9 SELECT name
10 FROM X
11 WHERE name LIKE '%&%' ;
12 COMMIT;

```

```

1 BEGIN;
2 WITH X AS (
3   SELECT name
4   FROM composers
5   WHERE name LIKE '%&%'
6 ), Y AS (
7   SELECT name
8   FROM artists
9   WHERE name LIKE '%&%'
10 )
11
12 SELECT *
13 FROM X
14 UNION
15 SELECT *
16 FROM Y;
17 COMMIT;

```



K14: Kommutativität von Selektion und Mengenvereinigung – V2

Regel: $\sigma_X(r_1) \cup \sigma_X(r_2) \sqsubseteq_K \sigma_X(r_1 \cup r_2)$

Invariante: —

Vorbedingungen: σ_X wird nicht unterstützt

Nachbedingungen: \cup wird unterstützt

```

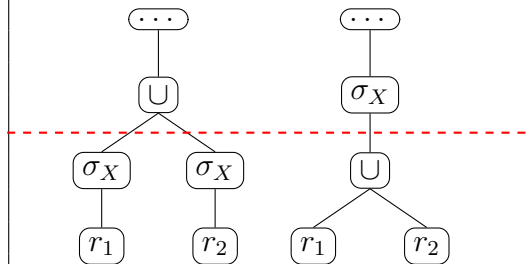
1 BEGIN;
2 WITH X AS (
3   SELECT name
4   FROM composers
5   WHERE id < 2500
6 ), Y AS (
7   SELECT name
8   FROM artists
9   WHERE id < 2500
10 )
11
12 SELECT *
13 FROM X
14 UNION
15 SELECT *
16 FROM Y;
17 COMMIT;

```

```

1 BEGIN;
2 WITH X AS (
3   (SELECT id, name
4   FROM composers) UNION
5   (SELECT id, name
6   FROM artists)
7 )
8
9 SELECT name
10 FROM X
11 WHERE id < 2500;
12 COMMIT;

```



K15: Kommutativität von Selektion und Mengendifferenz – V1

Regel: $\sigma_X(r_1 - r_2) \sqsubseteq_K \sigma_X(r_1) - r_2$

Invariante: —

Vorbedingungen: — wird nicht unterstützt

Nachbedingungen: σ_X wird unterstützt

```

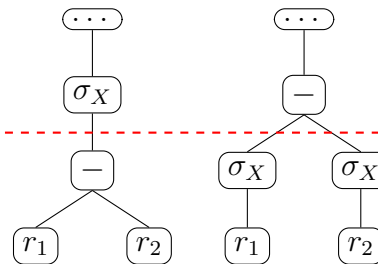
1 BEGIN;
2 WITH X AS (
3   (SELECT name
4    FROM composers) EXCEPT
5   (SELECT name
6    FROM artists)
7 )
8
9 SELECT *
10 FROM X
11 WHERE name LIKE '%&%' ;
12 COMMIT;

```

```

1 BEGIN;
2 WITH X AS (
3   SELECT name
4   FROM composers
5   WHERE name LIKE '%&%'
6 ), Y AS (
7   SELECT name
8   FROM artists
9   WHERE name LIKE '%&%'
10 )
11
12 SELECT *
13 FROM X
14 EXCEPT
15 SELECT *
16 FROM Y;
17 COMMIT;

```



K16: Kommutativität von Selektion und Mengendifferenz – V2

Regel: $\sigma_X(r_1) - \sigma_X(r_2) \sqsubseteq_K \sigma_X(r_1) - r_2$

Invariante: —

Vorbedingungen: σ_X wird im rechten Teilbaum nicht unterstützt

Nachbedingungen: —

```

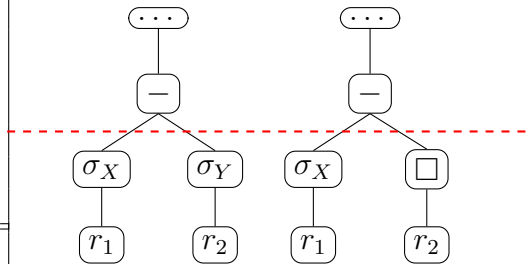
1 BEGIN;
2 WITH X AS (
3   SELECT name
4   FROM composers
5   WHERE name LIKE '%&%'
6 ), Y AS (
7   SELECT name
8   FROM artists
9   WHERE name LIKE '%&%'
10 )
11
12 SELECT *
13 FROM X
14 EXCEPT
15 SELECT *
16 FROM Y;
17 COMMIT;

```

```

1 BEGIN;
2 WITH X AS (
3   SELECT name
4   FROM composers
5   WHERE name LIKE '%&%'
6 ), Y AS (
7   SELECT name
8   FROM artists
9   WHERE name LIKE '%&%'
10 )
11
12 SELECT *
13 FROM X
14 EXCEPT
15 SELECT *
16 FROM Y;
17 COMMIT;

```



K17: Kommutativität von Selektion und Mengendifferenz – V3

Regel: $\sigma_X(r_1) - r_2 \sqsubseteq_K \sigma_X(r_1 - r_2)$

Invariante: —

Vorbedingungen: σ_X wird nicht unterstützt

Nachbedingungen: — wird unterstützt

```

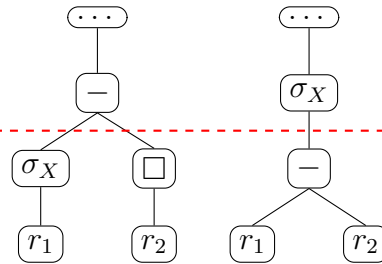
1 BEGIN;
2 WITH X AS (
3   SELECT name
4   FROM composers
5   WHERE name LIKE '%&%'
6 ), Y AS (
7   SELECT name
8   FROM artists
9 )
10
11 SELECT *
12 FROM X
13 EXCEPT
14 SELECT *
15 FROM Y;
16 COMMIT;

```

```

1 BEGIN;
2 WITH X AS (
3   SELECT name
4   FROM composers
5   EXCEPT
6   SELECT name
7   FROM artists
8 )
9
10 SELECT *
11 FROM X
12 WHERE name LIKE '%&%';
13 COMMIT;

```



K18: Kommutativität von Projektion und Mengenvereinigung – V1

Regel: $\pi_X(r_1 \cup r_2) \sqsubseteq_K \pi_X(r_1) \cup \pi_X(r_2)$

Invariante: —

Vorbedingungen: \cup wird nicht unterstützt

Nachbedingungen: π_X wird unterstützt

```

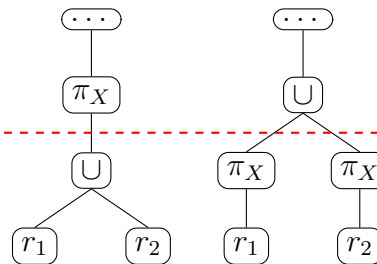
1 BEGIN;
2 WITH X AS (
3   SELECT *
4   FROM composers
5   UNION
6   SELECT *
7   FROM artists
8 )
9
10 SELECT DISTINCT name
11 FROM X;
12 COMMIT;

```

```

1 BEGIN;
2 WITH X AS (
3   SELECT name
4   FROM composers
5 ), Y AS (
6   SELECT name
7   FROM artists
8 )
9
10 SELECT *
11 FROM X
12 UNION
13 SELECT *
14 FROM Y;
15 COMMIT;

```



K19: Kommutativität von Projektion und Mengenvereinigung – V2

Regel: $\pi_X(r_1) \cup \pi_X(r_2) \sqsubseteq_K \pi_X(r_1 \cup r_2)$

Invariante: —

Vorbedingungen: π_X wird nicht unterstützt

Nachbedingungen: \cup wird unterstützt,
 $R(r_1) \equiv R(r_2)$

```

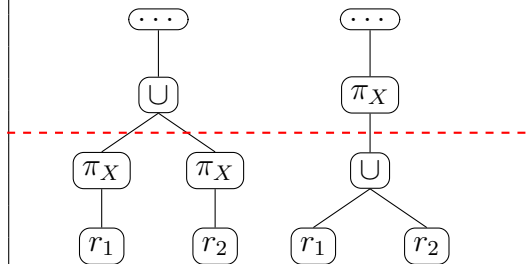
1 BEGIN;
2 WITH X AS (
3   SELECT name
4   FROM composers
5 ), Y AS (
6   SELECT name
7   FROM artists
8 )
9
10 SELECT *
11 FROM X
12 UNION
13 SELECT *
14 FROM Y;
15 COMMIT;

```

```

1 BEGIN;
2 WITH X AS (
3   SELECT *
4   FROM composers
5 UNION
6   SELECT *
7   FROM artists
8 )
9
10 SELECT DISTINCT name
11 FROM X;
12 COMMIT;

```



K20: Distributivität von Mengenvereinigung und Verbund – V1

Regel: $(R_1 \cup R_2) \bowtie (S_1 \cup S_2) \sqsubseteq_K (R_1 \bowtie S_1) \cup (R_1 \bowtie S_2) \cup (R_2 \bowtie S_1) \cup (R_2 \bowtie S_2)$

Invariante: —

Vorbedingungen: \cup wird nicht unterstützt

Nachbedingungen: \bowtie wird unterstützt

```

1 BEGIN;
2 WITH X AS (
3   (SELECT id AS aid, name
4     FROM artists
5     WHERE name LIKE '%&%' )
6   UNION
7   (SELECT id AS aid, name
8     FROM artists
9     WHERE name LIKE '%Nob%' )
10 ), Y AS (
11   (SELECT id AS tid, title,
12     artist
13     FROM tracks
14     WHERE length > 250000)
15   UNION
16   (SELECT id AS tid, title,
17     artist
18     FROM tracks
19     WHERE title LIKE '%song%' )
20 )
21 SELECT *
22 FROM X JOIN Y ON(X.aid=Y.
  artist);
  COMMIT;

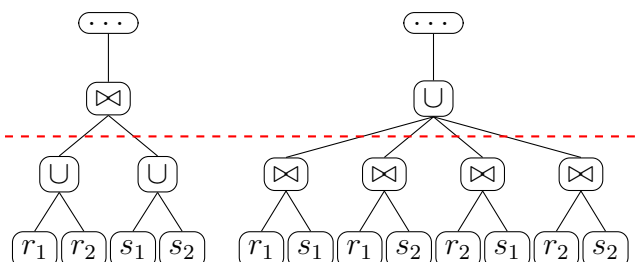
```

```

1 BEGIN;
2 WITH F1 AS (
3   SELECT *
4   FROM
5   (SELECT id AS aid, name
6     FROM artists
7     WHERE name LIKE '%&%' ) AS
8   F1X JOIN
9   (SELECT id AS tid, title,
10    artist
11    FROM tracks
12    WHERE length > 250000) AS
13   F1Y ON(F1X.aid=F1Y.artist)
14 ), F2 AS (
15   SELECT *
16   FROM
17   (SELECT id AS aid, name
18     FROM artists
19     WHERE name LIKE '%&%' ) AS
20   F2X JOIN
21   (SELECT id AS tid, title,
22    artist
23    FROM tracks
24    WHERE title LIKE '%song%' )
25   AS F2Y ON(F2X.aid=F2Y.

```

21



K21: Distributivität von Mengenvereinigung und Verbund – V2

Regel: $(r_1 \bowtie s_1) \cup (r_1 \bowtie s_2) \cup (r_2 \bowtie s_1) \cup (r_2 \bowtie s_2) \sqsubseteq_K (r_1 \cup r_2) \bowtie (s_1 \cup s_2)$

Invariante: —

Vorbedingungen: \bowtie wird nicht unterstützt

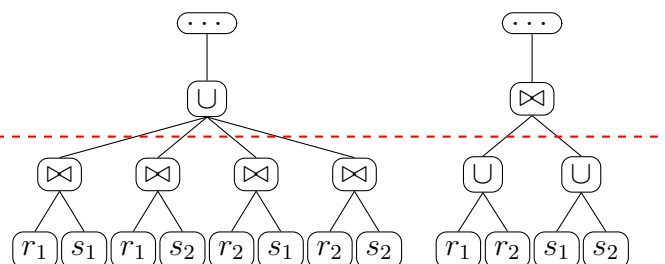
Nachbedingungen: \cup wird unterstützt

```

1 BEGIN;
2 WITH F1 AS (
3   SELECT *
4   FROM
5     (SELECT id AS aid, name
6      FROM artists
7      WHERE name LIKE '%&%' ) AS
8     F1X JOIN
9     (SELECT id AS tid, title,
10      artist
11     FROM tracks
12     WHERE length > 250000) AS
13     F1Y ON(F1X.aid=F1Y.artist)
14 ), F2 AS (
15   SELECT *
16   FROM
17     (SELECT id AS aid, name
18      FROM artists
19      WHERE name LIKE '%&%' ) AS
20     F2X JOIN
21     (SELECT id AS tid, title,
22      artist
23     FROM tracks
24     WHERE title LIKE '%song%' )
25     AS F2Y ON(F2X.aid=F2Y.
26     artist)
27 ), F3 AS (
28   SELECT *
29   FROM
30     (SELECT id AS aid, name
31      FROM artists
32      WHERE name LIKE '%Nob%' ) AS
33     F3X JOIN
34     (SELECT id AS tid, title,
35      artist
36     FROM tracks
37     WHERE length > 250000) AS
38     F3Y ON(F3X.aid=F3Y.artist)
39 ), F4 AS (
40   SELECT *
41   FROM
42     (SELECT id AS aid, name
43      FROM artists
44      WHERE name LIKE '%Nob%' ) AS
45     F4X JOIN
46     (SELECT id AS tid, title,
47      artist
48     FROM tracks
49     WHERE title LIKE '%song%' )
50     AS F4Y ON(F4X.aid=F4Y.

```

22



2 LAC1 + LAC2

L1: Kleiner zu Kleiner-gleich (Attribut-Konstante)

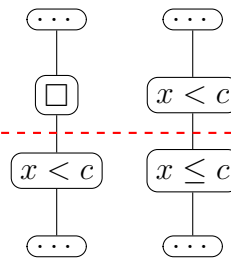
Regel: $x < c \sqsubseteq_K x \leq c$

Invariante: —

Vorbedingungen: $<$ wird nicht unterstützt

Nachbedingungen: $<$ wird unterstützt

```
1 BEGIN;  
2  
3 SELECT album, length  
4 FROM tracks  
5 WHERE length <= 250000;  
6  
7 COMMIT;  
-----  
1 BEGIN;  
2 WITH X AS (  
3   SELECT album, length  
4   FROM tracks  
5   WHERE length <= 250000  
6 )  
7 SELECT album, length  
8 FROM X  
9 WHERE length < 250000;  
10  
11 COMMIT;
```



L2: Gleich zu Kleiner-gleich (Attribut-Konstante)

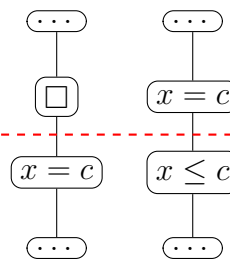
Regel: $x = c \sqsubseteq_K x \leq c$

Invariante: —

Vorbedingungen: = wird nicht unterstützt

Nachbedingungen: < wird unterstützt

```
1 BEGIN;
2
3 SELECT album, length
4 FROM tracks
5 WHERE length = 250000;
6
7 COMMIT;
-----
1 BEGIN;
2
3 WITH X AS (
4   SELECT album, length
5   FROM tracks
6   WHERE length <= 250000
7 )
8 SELECT album, length
9 FROM X
10 WHERE length = 250000;
11
12 COMMIT;
```



L3: Gleich zu Größer-gleich (Attribut-Konstante)

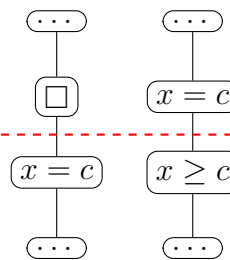
Regel: $x = c \sqsubseteq_K x \geq c$

Invariante: —

Vorbedingungen: = wird nicht unterstützt

Nachbedingungen: > wird unterstützt

```
1 BEGIN;  
2  
3 SELECT album, length  
4 FROM tracks  
5 WHERE length = 250000;  
6  
7 COMMIT;  
-----  
1 BEGIN;  
2  
3 WITH X AS (  
4   SELECT album, length  
5   FROM tracks  
6   WHERE length >= 250000  
7 )  
8 SELECT album, length  
9 FROM X  
10 WHERE length = 250000;  
11  
12 COMMIT;
```



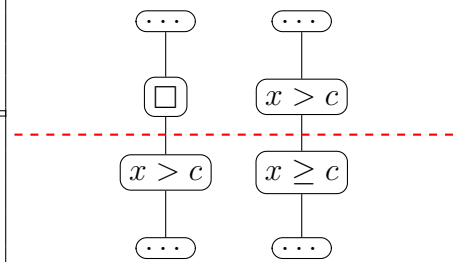
L4: Größer zu Größer-gleich (Attribut-Konstante)

Regel: $x > c \sqsubseteq_K x \geq c$

Vorbedingungen: $>$ wird nicht unterstützt

Nachbedingungen: $>$ wird unterstützt

```
1 BEGIN;  
2  
3 SELECT album, length  
4 FROM tracks  
5 WHERE length > 250000;  
6  
7 COMMIT;  
-----  
1 BEGIN;  
2  
3 WITH X AS (  
4   SELECT album, length  
5   FROM tracks  
6   WHERE length >= 250000  
7 )  
8 SELECT album, length  
9 FROM X  
10 WHERE length > 250000;  
11  
12 COMMIT;
```



L5: Kleiner zu Kleiner-gleich (Attribut-Attribut)

Regel: $x < y \sqsubseteq_K x \leq y$

Invariante: —

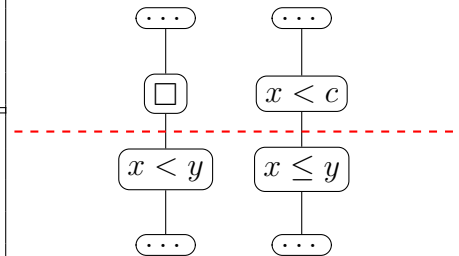
Vorbedingungen: $<$ wird nicht unterstützt

Nachbedingungen: $<$ wird unterstützt

```

1 BEGIN;
2
3 SELECT album, genre, bitrate
4 FROM tracks
5 WHERE length < 250000;
6
7 COMMIT;
-----
1 BEGIN;
2
3 WITH X AS (
4   SELECT album, genre,
5     bitrate
6   FROM tracks
7   WHERE genre <= bitrate
8 )
9 SELECT album, genre, bitrate
10 FROM X
11 WHERE genre < bitrate;
12 COMMIT;

```



L6: Gleich zu Kleiner-gleich (Attribut-Attribut)

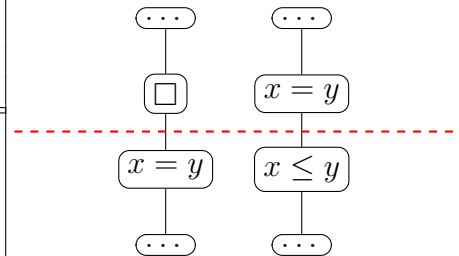
Regel: $x = y \sqsubseteq_K x \leq y$

Invariante: —

Vorbedingungen: = wird nicht unterstützt

Nachbedingungen: < wird unterstützt

```
1 BEGIN;  
2  
3 SELECT album, genre, bitrate  
4 FROM tracks  
5 WHERE genre = bitrate;  
6  
7 COMMIT;  
-----  
1 BEGIN;  
2  
3 WITH X AS (  
4   SELECT album, genre,  
5     bitrate  
6   FROM tracks  
7   WHERE genre <= bitrate  
8 )  
9 SELECT album, genre, bitrate  
10 FROM X  
11 WHERE genre = bitrate;  
12 COMMIT;
```



L7: Gleich zu Größer-gleich (Attribut-Attribut)

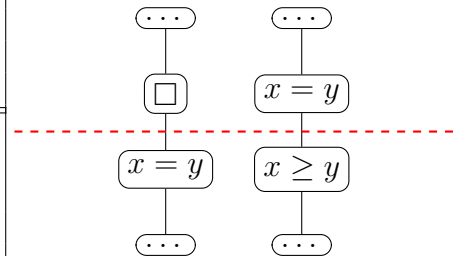
Regel: $x = y \sqsubseteq_K x \geq y$

Invariante: —

Vorbedingungen: = wird nicht unterstützt

Nachbedingungen: > wird unterstützt

```
1 BEGIN;
2
3 SELECT album, genre, bitrate
4 FROM tracks
5 WHERE genre = bitrate;
6
7 COMMIT;
-----
1 BEGIN;
2
3 WITH X AS (
4   SELECT album, genre,
5         bitrate
6   FROM tracks
7   WHERE genre >= bitrate
8 )
9 SELECT album, genre, bitrate
10 FROM X
11 WHERE genre = bitrate;
12 COMMIT;
```



L8: Größer zu Größer-gleich (Attribut-Attribut)

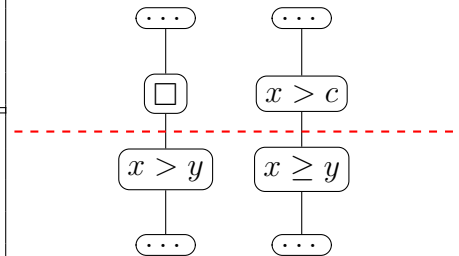
Regel: $x > y \sqsubseteq_K x \geq y$

Invariante: —

Vorbedingungen: $>$ wird nicht unterstützt

Nachbedingungen: $>$ wird unterstützt

```
1 BEGIN;  
2  
3 SELECT album, genre, bitrate  
4 FROM tracks  
5 WHERE genre > bitrate;  
6  
7 COMMIT;  
-----  
1 BEGIN;  
2  
3 WITH X AS (  
4   SELECT album, genre,  
5     bitrate  
6   FROM tracks  
7   WHERE genre >= bitrate  
8 )  
9 SELECT album, genre, bitrate  
10 FROM X  
11 WHERE genre > bitrate;  
12 COMMIT;
```



3 Aggregate Constraints

A1: Minimum-gleich zu Maximum-größer-gleich (Attribut-Konstante)

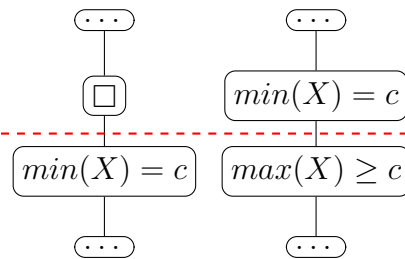
Regel: $\min(X) = c \sqsubseteq_K \max(X) \geq c$

Invariante: —

Vorbedingungen: \min oder $=$ wird nicht unterstützt

Nachbedingungen: \max und \geq wird unterstützt

```
1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING min(length) = 250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING max(length) >=
10    250000)
11 ) AS X
12 GROUP BY album
HAVING min(length) = 250000
```



A2: Minimum-größer-gleich zu Maximum-größer-gleich (Attribut-Konstante)

Regel: $\min(X) \geq c \sqsubseteq_K \max(X) \geq c$

Invariante: —

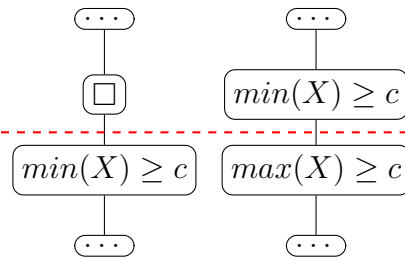
Vorbedingungen: \min wird nicht unterstützt

Nachbedingungen: \max wird unterstützt

```

1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING min(length) >= 250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING max(length) >=
10      250000)
11   ) AS X
12 GROUP BY album
13 HAVING min(length) >= 250000

```



A3: Minimum-größer zu Maximum-größer (Attribut-Konstante)

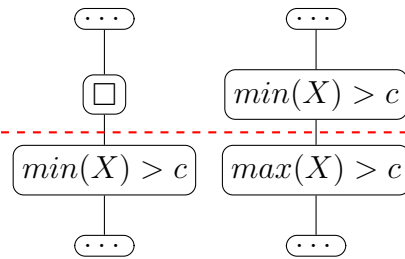
Regel: $\min(X) > c \sqsubseteq_K \max(X) > c$

Invariante: —

Vorbedingungen: \min wird nicht unterstützt

Nachbedingungen: \max wird unterstützt

```
1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING min(length) > 250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING max(length) >
10    250000)
11   ) AS X
11 GROUP BY album
12 HAVING min(length) > 250000
```



A4: Maximum-gleich zu Minimum-kleiner-gleich (Attribut-Konstante)

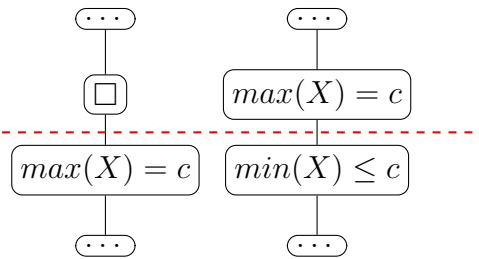
Regel: $\max(X) = c \sqsubseteq_K \min(X) \leq c$

Invariante: —

Vorbedingungen: \max oder $=$ wird nicht unterstützt

Nachbedingungen: \min und \leq wird unterstützt

```
1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING max(length) = 250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING min(length) <=
10    250000)
11   ) AS X
12 GROUP BY album
13 HAVING max(length) = 250000
```



A5: Maximum-kleiner-gleich zu Minimum-kleiner-gleich (Attribut-Konstante)

Regel: $\max(X) \leq c \sqsubseteq_K \min(X) \leq c$

Invariante: —

Vorbedingungen: \max wird nicht unterstützt

Nachbedingungen: \min wird unterstützt

```

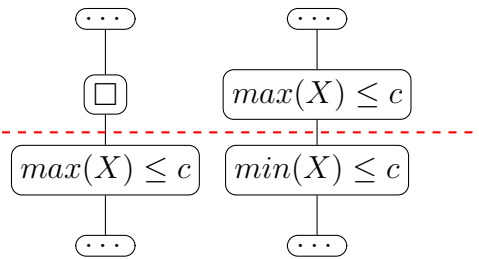
1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING max(length) <= 250000

```

```

1 SELECT album
2 FROM (
3   SELECT album, length
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING min(length) <=
10    250000)
11   ) AS X
12 GROUP BY album
HAVING max(length) <= 250000

```



A6: Maximum-kleiner zu Minimum-kleiner (Attribut-Konstante)

Regel: $\max(X) < c \sqsubseteq_K \min(X) < c$

Invariante: —

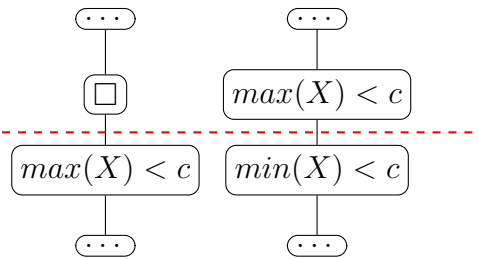
Vorbedingungen: \max wird nicht unterstützt

Nachbedingungen: \min wird unterstützt

```

1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING max(length) < 250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING min(length) <
10    250000)
11   ) AS X
12 GROUP BY album
13 HAVING max(length) < 250000

```



A7: Minimum-gleich zu Durchschnitt-größer-gleich (Attribut-Konstante)

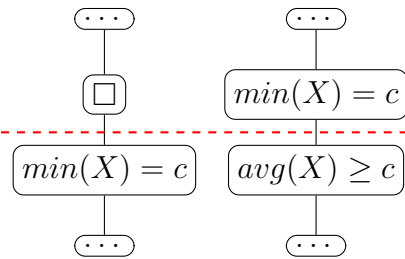
Regel: $\min(X) = c \sqsubseteq_K \text{avg}(X) \geq c$

Invariante: —

Vorbedingungen: \min oder $=$ wird nicht unterstützt

Nachbedingungen: avg und \geq wird unterstützt

```
1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING  $\min(\text{length}) = 250000$ 
-----
1 SELECT album
2 FROM (
3   SELECT album, length
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING  $\text{avg}(\text{length}) \geq$ 
10    250000)
11   ) AS X
12 GROUP BY album
13 HAVING  $\min(\text{length}) = 250000$ 
```



A8: Minimum-größer-gleich zu Durchschnitt-größer-gleich (Attribut-Konstante)

Regel: $\min(X) \geq c \sqsubseteq_K \text{avg}(X) \geq c$

Invariante: —

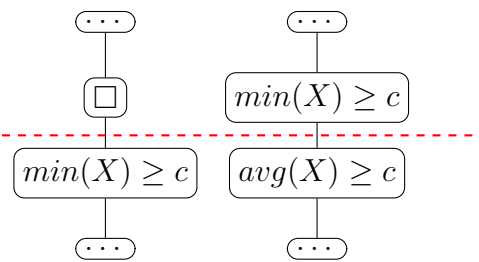
Vorbedingungen: \min wird nicht unterstützt

Nachbedingungen: avg wird unterstützt

```

1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING min(length) >= 250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING avg(length) >=
10      250000)
11   ) AS X
12 GROUP BY album
13 HAVING min(length) >= 250000

```



A9: Minimum-größer zu Durchschnitt-größer (Attribut-Konstante)

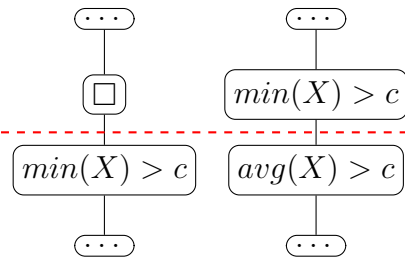
Regel: $\min(X) > c \sqsubseteq_K \text{avg}(X) > c$

Invariante: —

Vorbedingungen: \min wird nicht unterstützt

Nachbedingungen: avg und $>$ wird unterstützt

```
1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING  $\min(\text{length}) > 250000$ 
-----
1 SELECT album
2 FROM (
3   SELECT album, length
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING  $\text{avg}(\text{length}) >$ 
10    250000)
11   ) AS X
12 GROUP BY album
13 HAVING  $\min(\text{length}) > 250000$ 
```



A10: Maximum-gleich zu Durchschnitt-kleiner-gleich (Attribut-Konstante)

Regel: $\max(X) = c \sqsubseteq_K \text{avg}(X) \leq c$

Invariante: —

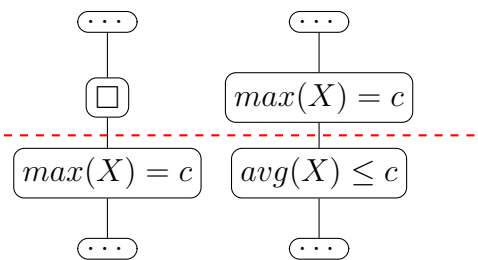
Vorbedingungen: \max oder $=$ wird nicht unterstützt

Nachbedingungen: avg und \leq wird unterstützt

```

1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING max(length) = 250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING avg(length) <=
10    250000)
11   ) AS X
12 GROUP BY album
13 HAVING max(length) = 250000

```



A11: Maximum-kleiner-gleich zu Durchschnitt-kleiner-gleich (Attribut-Konstante)

Regel: $\max(X) \leq c \sqsubseteq_K \text{avg}(X) \leq c$

Invariante: —

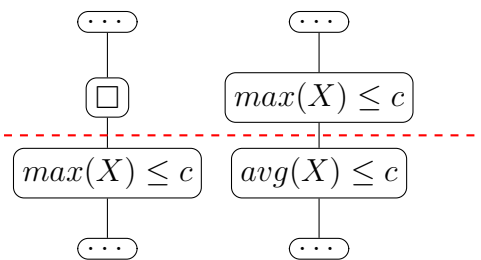
Vorbedingungen: \max wird nicht unterstützt

Nachbedingungen: avg wird unterstützt

```

1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING max(length) <= 250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING avg(length) <=
10    250000)
11   ) AS X
12 GROUP BY album
13 HAVING max(length) <= 250000

```



A12: Maximum-kleiner zu Durchschnitt-kleiner (Attribut-Konstante)

Regel: $\max(X) < c \sqsubseteq_K \text{avg}(X) < c$

Invariante: —

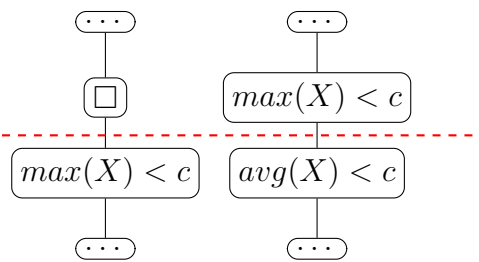
Vorbedingungen: \max wird nicht unterstützt

Nachbedingungen: avg wird unterstützt

```

1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING max(length) < 250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING avg(length) <
10    250000)
11   ) AS X
12 GROUP BY album
13 HAVING max(length) < 250000

```



A13: Minimum-gleich zu Summe-größer-gleich (Attribut-Konstante)

Regel: $\min(X) = c \sqsubseteq_K \text{sum}(X) \geq c$

Invariante: $c \geq 0$

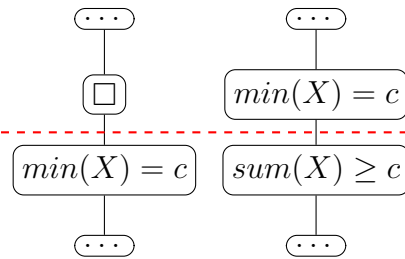
Vorbedingungen: \min oder $=$ wird nicht unterstützt

Nachbedingungen: sum und \geq wird unterstützt

```

1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING min(length) = 250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING sum(length) >=
10    250000)
11 ) AS X
12 GROUP BY album
13 HAVING min(length) = 250000

```



A14: Minimum-größer-gleich zu Summe-größer-gleich (Attribut-Konstante)

Regel: $\min(X) \geq c \sqsubseteq_K \text{sum}(X) \geq c$

Invariante: $c \geq 0$

Vorbedingungen: \min wird nicht unterstützt

Nachbedingungen: sum wird unterstützt

```

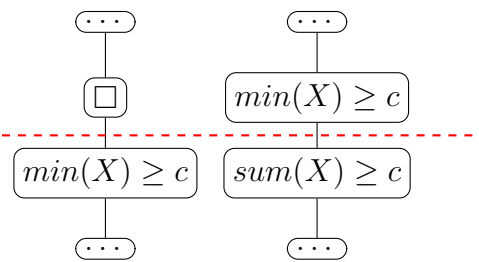
1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING  $\min(\text{length}) \geq 250000$ 

```

```

1 SELECT album
2 FROM (
3   SELECT album, length
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING  $\text{sum}(\text{length}) \geq 250000$ )
10  ) AS X
11 GROUP BY album
12 HAVING  $\min(\text{length}) \geq 250000$ 

```



A15: Minimum-kleiner zu Summe-größer (Attribut-Konstante)

Regel: $\min(X) > c \sqsubseteq_K \text{sum}(X) > c$

Invariante: $c \geq 0$

Vorbedingungen: \min wird nicht unterstützt

Nachbedingungen: sum wird unterstützt

```

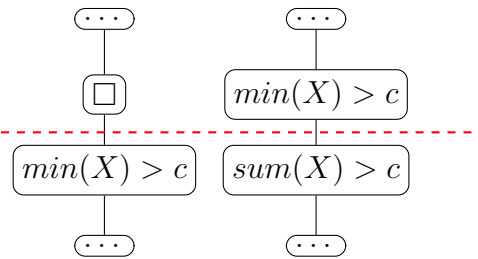
1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING  $\min(\text{length}) > 250000$ 

```

```

1 SELECT album
2 FROM (
3   SELECT album, length
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING  $\text{sum}(\text{length}) >$ 
10    250000)
11   ) AS X
12 GROUP BY album
13 HAVING  $\min(\text{length}) > 250000$ 

```



A16: Maximum-gleich zu Summe-kleiner-gleich (Attribut-Konstante)

Regel: $\max(X) = c \sqsubseteq_K \text{sum}(X) \leq c$

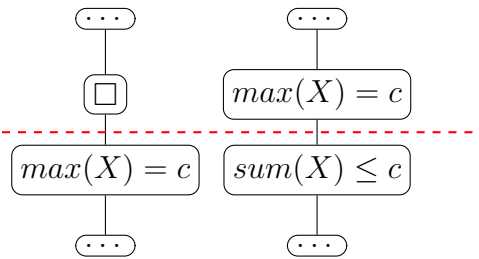
Invariante: $c \leq 0$

Vorbedingungen: \max oder $=$ wird nicht unterstützt

Nachbedingungen: sum und \leq wird unterstützt

```

1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING max(length*-1) =
  -250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length*-1 AS
  L
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING sum(length*-1) <=
  -250000)
10  ) AS X
11 GROUP BY album
12 HAVING max(L) = -250000
  
```



A17: Maximum-größer-gleich zu Summe-kleiner-gleich (Attribut-Konstante)

Regel: $\max(X) \geq c \sqsubseteq_K \text{sum}(X) \leq c$

Invariante: $c \leq 0$

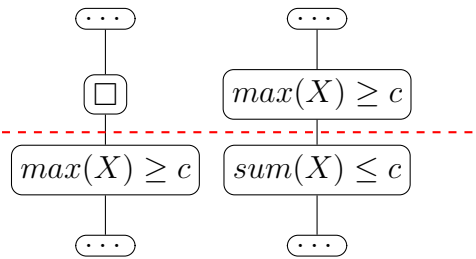
Vorbedingungen: \max oder \geq wird nicht unterstützt

Nachbedingungen: sum und \leq wird unterstützt

```

1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING max(length*-1) >=
  -250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length*-1 AS
  L
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING sum(length*-1) <=
  -250000)
10  ) AS X
11 GROUP BY album
12 HAVING max(L) >= -250000

```



A18: Maximum-größer zu Summe-kleiner (Attribut-Konstante)

Regel: $\max(X) > c \sqsubseteq_K \text{sum}(X) < c$

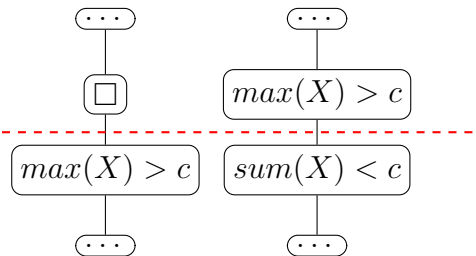
Invariante: $c \leq 0$

Vorbedingungen: \max oder $>$ wird nicht unterstützt

Nachbedingungen: sum und $<$ wird unterstützt

```

1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING max(length*-1) >
  -250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length*-1 AS
  L
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING sum(length*-1) <
  -250000)
10  ) AS X
11 GROUP BY album
12 HAVING max(L) > -250000
  
```



A19: Durchschnitt-größer-gleich zu Summe-größer-gleich (Attribut-Konstante)

Regel: $avg(X) \geq c \sqsubseteq_K sum(X) \geq c$

Invariante: $c \geq 0$

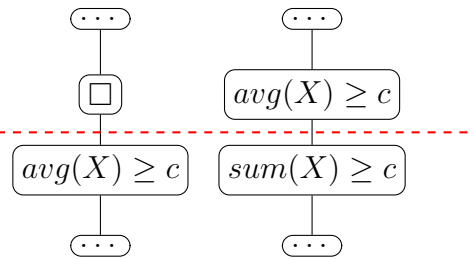
Vorbedingungen: avg wird nicht unterstützt

Nachbedingungen: sum wird unterstützt

```

1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING avg(length) >= 250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING sum(length) >=
10    250000)
11   ) AS X
12 GROUP BY album
13 HAVING avg(length) >= 250000

```



A20: Durchschnitt-größer zu Summe-größer (Attribut-Konstante)

Regel: $avg(X) > c \sqsubseteq_K sum(X) > c$

Invariante: $c \geq 0$

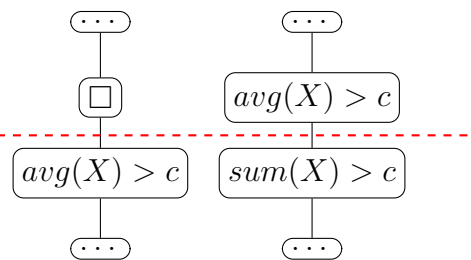
Vorbedingungen: avg wird nicht unterstützt

Nachbedingungen: sum wird unterstützt

```

1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING avg(length) > 250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING sum(length) >
10    250000)
11   ) AS X
11 GROUP BY album
12 HAVING avg(length) > 250000

```



A21: Durchschnitt-kleiner-gleich zu Summe-kleiner-gleich (Attribut-Konstante)

Regel: $avg(X) \leq c \sqsubseteq_K sum(X) \leq c$

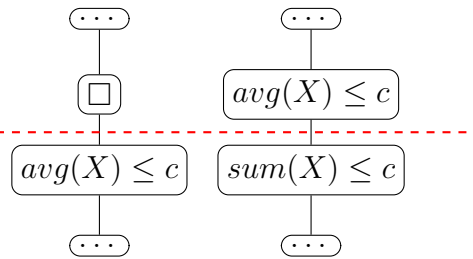
Invariante: $c \leq 0$

Vorbedingungen: avg wird nicht unterstützt

Nachbedingungen: sum wird unterstützt

```

1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING avg(length*-1) <=
  -250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length*-1 AS
  L
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING sum(length*-1) <=
  -250000)
10  ) AS X
11 GROUP BY album
12 HAVING avg(L) <= -250000
  
```



A22: Durchschnitt-kleiner zu Summe-kleiner (Attribut-Konstante)

Regel: $avg(X) < c \sqsubseteq_K sum(X) < c$

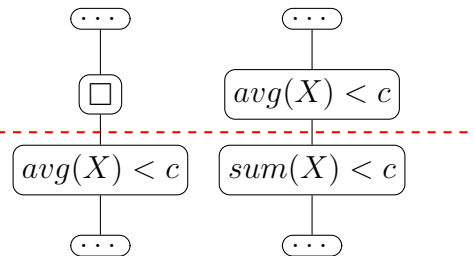
Invariante: $c \leq 0$

Vorbedingungen: avg wird nicht unterstützt

Nachbedingungen: sum wird unterstützt

```

1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING avg(length*-1) <
  -250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length*-1 AS
  L
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING sum(length*-1) <
  -250000)
10  ) AS X
11 GROUP BY album
12 HAVING avg(L) < -250000
  
```



A23: Summe-größer-gleich zu Durchschnitt-größer-gleich-0 (Attribut-Konstante)

Regel: $sum(X) \geq c \sqsubseteq_K avg(X) \geq 0$

Invariante: $c \geq 0$

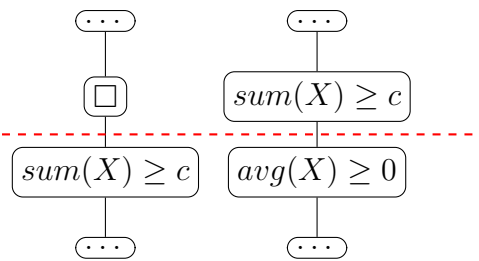
Vorbedingungen: sum wird nicht unterstützt

Nachbedingungen: avg wird unterstützt

```

1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING sum(length) >= 250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING avg(length) >= 0)
10  ) AS X
11 GROUP BY album
12 HAVING sum(length) >= 250000

```



A24: Summe-größer zu Durchschnitt-größer-0 (Attribut-Konstante)

Regel: $sum(X) > c \sqsubseteq_K avg(X) > 0$

Invariante: $c \geq 0$

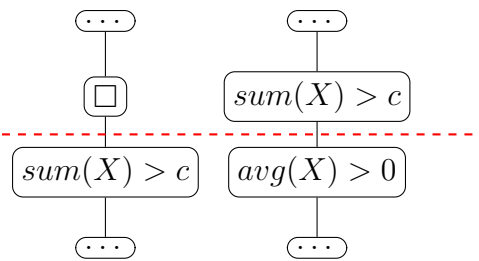
Vorbedingungen: sum wird nicht unterstützt

Nachbedingungen: avg wird unterstützt

```

1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING sum(length) > 250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING avg(length) > 0)
10  ) AS X
11 GROUP BY album
12 HAVING sum(length) > 250000

```



A25: Summe-kleiner-gleich zu Durchschnitt-kleiner-gleich-0 (Attribut-Konstante)

Regel: $sum(X) \leq c \sqsubseteq_K avg(X) \leq 0$

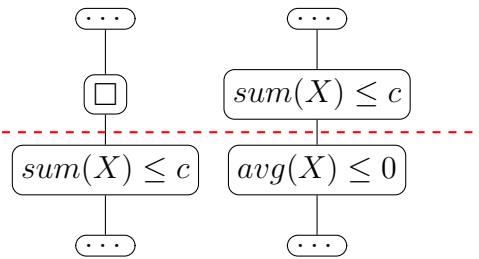
Invariante: $c \leq 0$

Vorbedingungen: sum wird nicht unterstützt

Nachbedingungen: avg wird unterstützt

```

1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING sum(length*-1) <=
  -250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length*-1 AS
  L
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING avg(length*-1) <=
  0)
10  ) AS X
11 GROUP BY album
12 HAVING sum(L) <= -250000
  
```



A26: Summe-kleiner zu Durchschnitt-kleiner-0 (Attribut-Konstante)

Regel: $sum(X) < c \sqsubseteq_K avg(X) < 0$

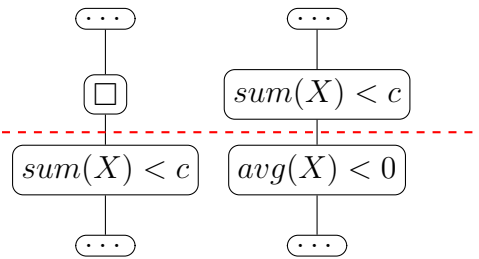
Invariante: $c \leq 0$

Vorbedingungen: sum wird nicht unterstützt

Nachbedingungen: avg wird unterstützt

```

1 SELECT album
2 FROM tracks
3 GROUP BY album
4 HAVING sum(length*-1) <
  -250000
-----
1 SELECT album
2 FROM (
3   SELECT album, length*-1 AS
  L
4   FROM tracks
5   WHERE album IN (
6     SELECT album
7     FROM tracks
8     GROUP BY album
9     HAVING avg(length*-1) <
  0)
10  ) AS X
11 GROUP BY album
12 HAVING sum(L) < -250000
  
```



A27: Kleiner zu Maximum-kleiner (Attribut-Konstante)

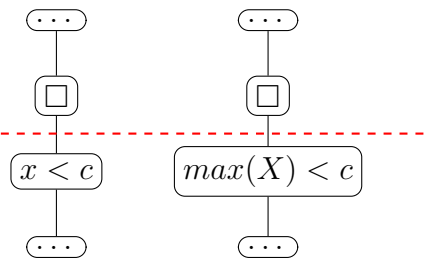
Regel: $\forall x \in X : x < c \sqsubseteq_K \max(X) < c$

Invariante: —

Vorbedingungen: \forall wird nicht unterstützt

Nachbedingungen: \max wird unterstützt

1	SELECT DISTINCT album
2	FROM tracks
3	WHERE album NOT IN (
4	SELECT album
5	FROM tracks
6	WHERE NOT (length < 250000)
7)
1	SELECT DISTINCT album
2	FROM tracks
3	GROUP BY album
4	HAVING max (length) < 250000



A28: Kleiner-gleich zu Maximum-kleiner-gleich (Attribut-Konstante)

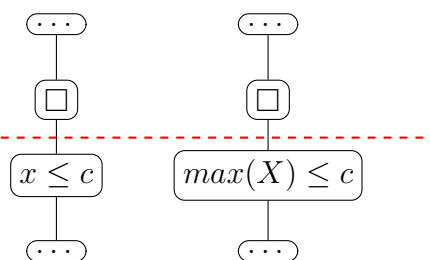
Regel: $\forall x \in X : x \leq c \sqsubseteq_K \max(X) \leq c$

Invariante: —

Vorbedingungen: \forall wird nicht unterstützt

Nachbedingungen: \max wird unterstützt

1	SELECT DISTINCT album
2	FROM tracks
3	WHERE album NOT IN (
4	SELECT album
5	FROM tracks
6	WHERE NOT (length <= 250000)
7)
1	SELECT DISTINCT album
2	FROM tracks
3	GROUP BY album
4	HAVING max (length) <= 250000



A29: Gleich zu Minimum-gleich (Attribut-Konstante)

Regel: $\forall x \in X : x = c \sqsubseteq_K \min(X) = c$

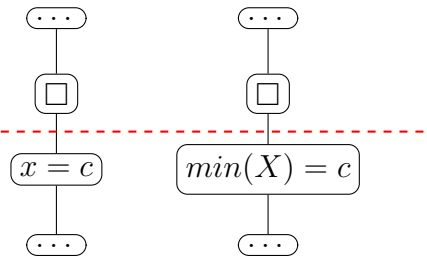
Invariante: —

Vorbedingungen: \forall wird nicht unterstützt

Nachbedingungen: \min wird unterstützt

```

1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN (
4   SELECT album
5   FROM tracks
6   WHERE NOT (length = 250000)
7 )
-----
1 SELECT DISTINCT album
2 FROM tracks
3 GROUP BY album
4 HAVING min(length) = 250000
  
```



A30: Gleich zu Maximum-gleich (Attribut-Konstante)

Regel: $\forall x \in X : x = c \sqsubseteq_K \max(X) = c$

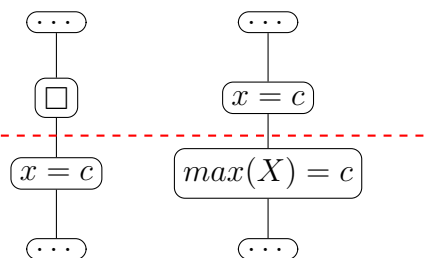
Invariante: —

Vorbedingungen: \forall wird nicht unterstützt

Nachbedingungen: \max wird unterstützt

```

1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN (
4   SELECT album
5   FROM tracks
6   WHERE NOT (length = 250000)
7 )
-----
1 SELECT DISTINCT album
2 FROM tracks
3 GROUP BY album
4 HAVING max(length) = 250000
  
```



A31: Ungleich zu Minimum-ungleich (Attribut-Konstante)

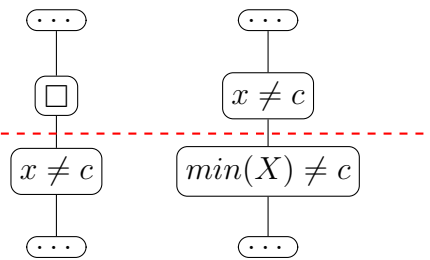
Regel: $\forall x \in X : x \neq c \sqsubseteq_K \min(X) \neq c$

Invariante: —

Vorbedingungen: \forall wird nicht unterstützt

Nachbedingungen: \min wird unterstützt

1	SELECT DISTINCT album
2	FROM tracks
3	WHERE album NOT IN (
4	SELECT album
5	FROM tracks
6	WHERE NOT (length != 250000)
7)
1	SELECT DISTINCT album
2	FROM tracks
3	GROUP BY album
4	HAVING min (length) != 250000



A32: Ungleich zu Maximum-ungleich (Attribut-Konstante)

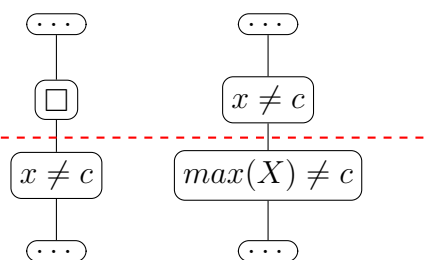
Regel: $\forall x \in X : x \neq c \sqsubseteq_K \max(X) \neq c$

Invariante: —

Vorbedingungen: \forall wird nicht unterstützt

Nachbedingungen: \max wird unterstützt

1	SELECT DISTINCT album
2	FROM tracks
3	WHERE album NOT IN (
4	SELECT album
5	FROM tracks
6	WHERE NOT (length != 250000)
7)
1	SELECT DISTINCT album
2	FROM tracks
3	GROUP BY album
4	HAVING max (length) != 250000



A33: Größer-gleich zu Minimum-größer-gleich (Attribut-Konstante)

Regel: $\forall x \in X : x \geq c \sqsubseteq_K \min(X) \geq c$

Invariante: —

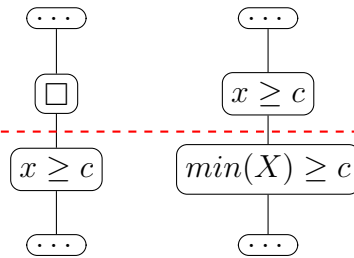
Vorbedingungen: \forall wird nicht unterstützt

Nachbedingungen: \min wird unterstützt

```

1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN (
4   SELECT album
5   FROM tracks
6   WHERE NOT (length >= 250000)
7 )
-----
1 SELECT DISTINCT album
2 FROM tracks
3 GROUP BY album
4 HAVING min(length) >= 250000

```



A34: Größer zu Minimum-größer (Attribut-Konstante)

Regel: $\forall x \in X : x > c \sqsubseteq_K \min(X) > c$

Invariante: —

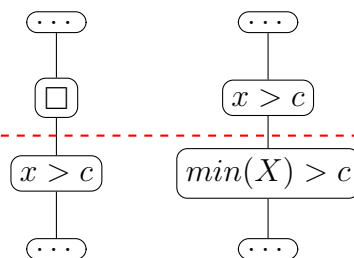
Vorbedingungen: \forall wird nicht unterstützt

Nachbedingungen: \min wird unterstützt

```

1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN (
4   SELECT album
5   FROM tracks
6   WHERE NOT (length > 250000)
7 )
-----
1 SELECT DISTINCT album
2 FROM tracks
3 GROUP BY album
4 HAVING min(length) > 250000

```



A35: Kleiner zu Durchschnitt-kleiner (Attribut-Konstante)

Regel: $\forall x \in X : x < c \sqsubseteq_K \text{avg}(X) < c$

Invariante: —

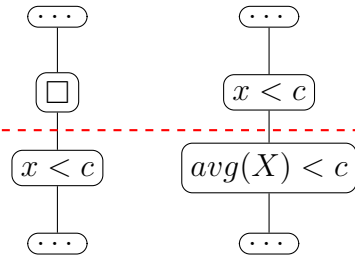
Vorbedingungen: \forall wird nicht unterstützt

Nachbedingungen: *avg* wird unterstützt

```

1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   WHERE NOT(length < 250000)
7 )
-----
1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   GROUP BY album
7   HAVING NOT(avg(length) <
8     250000))
9 AND album NOT IN(
10  SELECT album
11  FROM tracks
12  WHERE NOT(length < 250000)
13 )

```



A36: Kleiner-gleich zu Durchschnitt-kleiner-gleich (Attribut-Konstante)

Regel: $\forall x \in X : x \leq c \sqsubseteq_K \text{avg}(X) \leq c$

Invariante: —

Vorbedingungen: \forall wird nicht unterstützt

Nachbedingungen: *avg* wird unterstützt

```

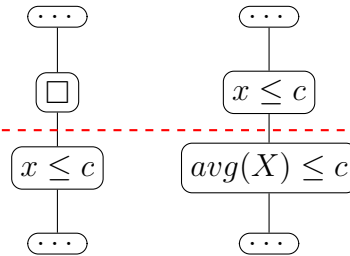
1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   WHERE NOT(length <= 250000)
7 )

```

```

1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   GROUP BY album
7   HAVING NOT(avg(length) <=
8     250000))
9 AND album NOT IN(
10  SELECT album
11  FROM tracks
12  WHERE NOT(length <= 250000)

```



A37: Gleich zu Durchschnitt-gleich (Attribut-Konstante)

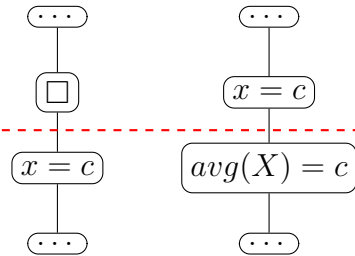
Regel: $\forall x \in X : x = c \sqsubseteq_K \text{avg}(X) = c$

Invariante: —

Vorbedingungen: \forall wird nicht unterstützt

Nachbedingungen: *avg* wird unterstützt

```
1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   WHERE NOT(length = 250000)
7 )
-----
1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   GROUP BY album
7   HAVING NOT(avg(length) =
8     250000))
9 AND album NOT IN(
10  SELECT album
11  FROM tracks
12  WHERE NOT(length = 250000)
13 )
```



A38: Größer-gleich zu Durchschnitt-größer-gleich (Attribut-Konstante)

Regel: $\forall x \in X : x \geq c \sqsubseteq_K \text{avg}(X) \geq c$

Invariante: —

Vorbedingungen: \forall wird nicht unterstützt

Nachbedingungen: *avg* wird unterstützt

```

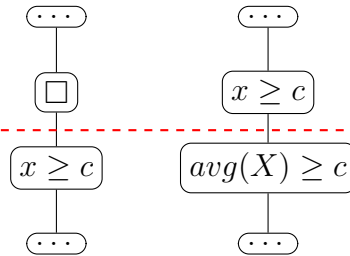
1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   WHERE NOT(length >= 250000)
7 )

```

```

1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   GROUP BY album
7   HAVING NOT(avg(length) >=
8     250000))
9 AND album NOT IN(
10  SELECT album
11  FROM tracks
12  WHERE NOT(length >= 250000)

```



A39: Größer zu Durchschnitt-größer (Attribut-Konstante)

Regel: $\forall x \in X : x > c \sqsubseteq_K \text{avg}(X) > c$

Invariante: —

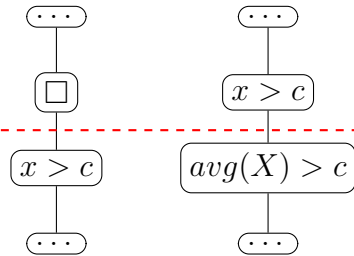
Vorbedingungen: \forall wird nicht unterstützt

Nachbedingungen: *avg* wird unterstützt

```

1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   WHERE NOT(length > 250000)
7 )
-----
1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   GROUP BY album
7   HAVING NOT(avg(length) >
8     250000))
9 AND album NOT IN(
10  SELECT album
11  FROM tracks
12  WHERE NOT(length > 250000)
13 )

```



A40: Kleiner-gleich zu Summe-kleiner-gleich (Attribut-Konstante)

Regel: $\forall x \in X : x \leq c \sqsubseteq_K \text{sum}(X) \leq c$

Invariante: $c \leq 0$

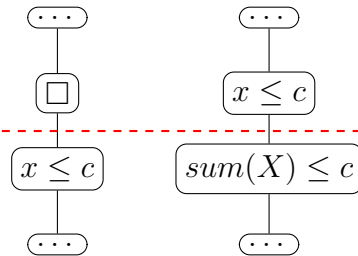
Vorbedingungen: \forall wird nicht unterstützt

Nachbedingungen: *sum* wird unterstützt

```

1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   WHERE NOT(length*-1 <=
7     -250000)
8 )
9
10 SELECT DISTINCT album
11 FROM tracks
12 WHERE album NOT IN(
13   SELECT album
14   FROM tracks
15   GROUP BY album
16   HAVING NOT(sum(length*-1)
17     <= -250000))
18 AND album NOT IN(
19   SELECT album
20   FROM tracks
21   WHERE NOT(length*-1 <=
22     -250000)
23 )

```



A41: Kleiner zu Summe-kleiner (Attribut-Konstante)

Regel: $\forall x \in X : x < c \sqsubseteq_K \text{sum}(X) < c$

Invariante: $c \leq 0$

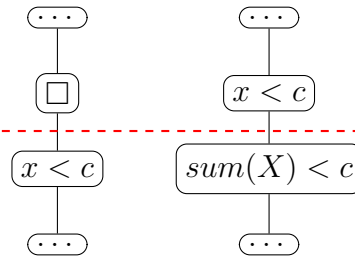
Vorbedingungen: \forall wird nicht unterstützt

Nachbedingungen: *sum* wird unterstützt

```

1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   WHERE NOT(length*-1 <
7     -250000)
8 )
9
10 SELECT DISTINCT album
11 FROM tracks
12 WHERE album NOT IN(
13   SELECT album
14   FROM tracks
15   GROUP BY album
16   HAVING NOT(sum(length*-1) <
17     -250000))
18 AND album NOT IN(
19   SELECT album
20   FROM tracks
21   WHERE NOT(length*-1 <
22     -250000)
23 )

```



A42: Gleich zu Summe-kleiner-gleich (Attribut-Konstante)

Regel: $\forall x \in X : x = c \sqsubseteq_K \text{sum}(X) \leq c$

Invariante: $c \leq 0$

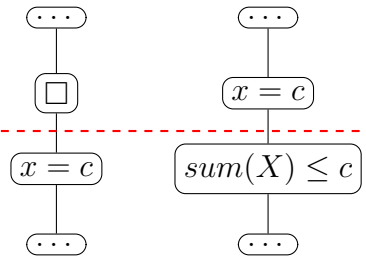
Vorbedingungen: \forall oder $=$ wird nicht unterstützt

Nachbedingungen: sum und \leq wird unterstützt

```

1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   WHERE NOT(length*-1 =
7     -250000)
8 )
9
10 SELECT DISTINCT album
11 FROM tracks
12 WHERE album NOT IN(
13   SELECT album
14   FROM tracks
15   GROUP BY album
16   HAVING NOT(sum(length*-1)
17     <= -250000))
18 AND album NOT IN(
19   SELECT album
20   FROM tracks
21   WHERE NOT(length*-1 =
22     -250000)
23 )

```



A43: Größer zu Summe-größer (Attribut-Konstante)

Regel: $\forall x \in X : x > c \sqsubseteq_K \text{sum}(X) > c$

Invariante: $c \geq 0$

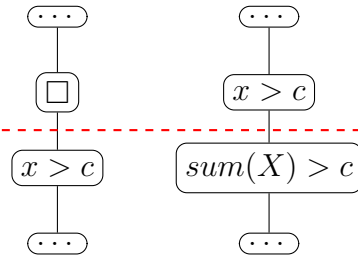
Vorbedingungen: \forall oder $=$ wird nicht unterstützt

Nachbedingungen: sum und \leq wird unterstützt

```

1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   WHERE NOT(length = 250000)
7 )
-----
1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   GROUP BY album
7   HAVING NOT(sum(length) <=
8     250000))
9 AND album NOT IN(
10  SELECT album
11  FROM tracks
12  WHERE NOT(length = 250000)
13 )

```



A44: Größer-gleich zu Summe-größer-gleich (Attribut-Konstante)

Regel: $\forall x \in X : x \geq c \sqsubseteq_K \text{sum}(X) \geq c$

Invariante: $c \geq 0$

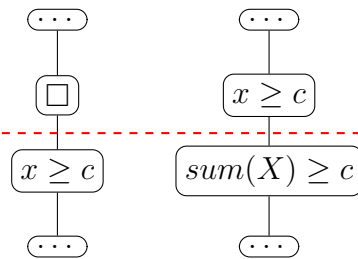
Vorbedingungen: \forall wird nicht unterstützt

Nachbedingungen: sum wird unterstützt

```

1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   WHERE NOT(length >= 250000)
7 )
-----
1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   GROUP BY album
7   HAVING NOT(sum(length) >=
8     250000))
9 AND album NOT IN(
10  SELECT album
11  FROM tracks
12  WHERE NOT(length >= 250000)
13 )

```



A45: Gleich zu Summe-größer-gleich (Attribut-Konstante)

Regel: $\forall x \in X : x = c \sqsubseteq_K \text{sum}(X) \geq c$

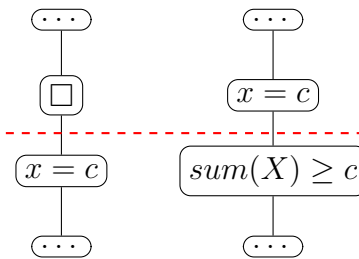
Invariante: $c \geq 0$

Vorbedingungen: \forall oder $=$ wird nicht unterstützt

Nachbedingungen: sum und \geq wird unterstützt

```

1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   WHERE NOT(length = 250000)
7 )
-----
1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   GROUP BY album
7   HAVING NOT(sum(length) >=
8     250000))
9 AND album NOT IN(
10  SELECT album
11  FROM tracks
12  WHERE NOT(length = 250000)
13 )
  
```



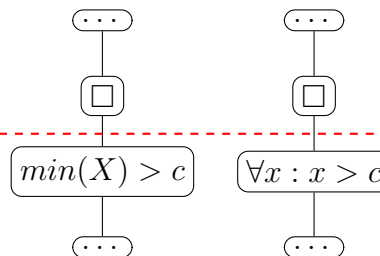
A46: Minimum-größer zu Größer (Attribut-Konstante)

Regel: $\text{min}(X) > c \sqsubseteq_K \forall x \in X : x > c$

Invariante: —

Vorbedingungen: min wird nicht unterstützt

Nachbedingungen: \forall wird unterstützt (Anfragen siehe A34)



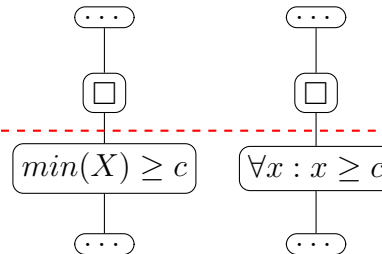
A47: Minimum-größer-gleich zu größer-gleich (Attribut-Konstante)

Regel: $\min(X) \geq c \sqsubseteq_K \forall x \in X : x \geq c$

Invariante: —

Vorbedingungen: \min wird nicht unterstützt

Nachbedingungen: \forall wird unterstützt (Anfragen siehe A33)



A48: Minimum-gleich zu größer-gleich (Attribut-Konstante)

Regel: $\min(X) = c \sqsubseteq_K \forall x \in X : x \geq c$

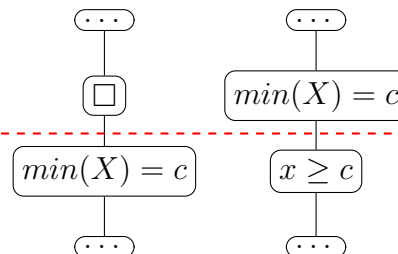
Invariante: —

Vorbedingungen: \min oder $=$ wird nicht unterstützt

Nachbedingungen: \geq und \forall wird unterstützt

```

1 SELECT DISTINCT album
2 FROM tracks
3 GROUP BY album
4 HAVING min(length) = 250000
-----
1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   WHERE NOT(length >= 250000)
7 )
8 AND album IN(
9   SELECT album
10  FROM tracks
11  GROUP BY album
12  HAVING min(length) = 250000
13 )
  
```



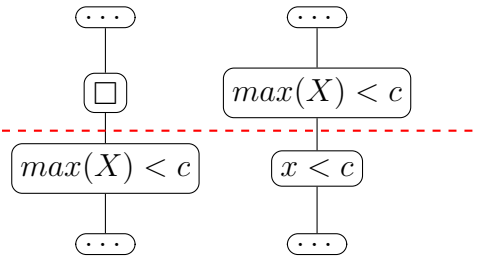
A49: Maximum-kleiner zu kleiner (Attribut-Konstante)

Regel: $\max(X) < c \sqsubseteq_K \forall x \in X : x < c$

Invariante: —

Vorbedingungen: \max wird nicht unterstützt

Nachbedingungen: \forall wird unterstützt (Anfragen siehe A27)



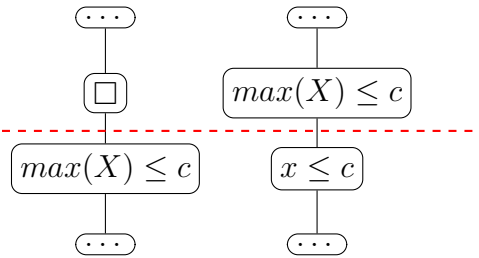
A50: Maximum-kleiner-gleich zu kleiner-gleich (Attribut-Konstante)

Regel: $\max(X) \leq c \sqsubseteq_K \forall x \in X : x \leq c$

Invariante: —

Vorbedingungen: \max wird nicht unterstützt

Nachbedingungen: \forall wird unterstützt (Anfragen siehe A28)



A51: Maximum-gleich zu kleiner-gleich (Attribut-Konstante)

Regel: $max(X) = c \sqsubseteq_K \forall x \in X : x \leq c$

Invariante: —

Vorbedingungen: max oder $=$ wird nicht unterstützt

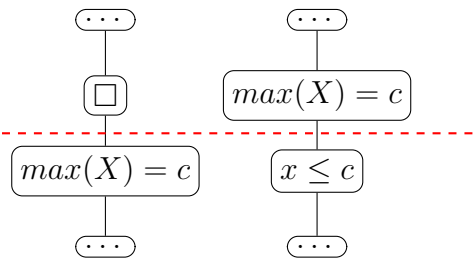
Nachbedingungen: \leq und \forall wird unterstützt

```

1 SELECT DISTINCT album
2 FROM tracks
3 GROUP BY album
4 HAVING max(length) = 250000
-----
1 SELECT DISTINCT album
2 FROM tracks
3 WHERE album NOT IN(
4   SELECT album
5   FROM tracks
6   WHERE NOT(length <= 250000)
7 )
8 AND album IN(
9   SELECT album
10  FROM tracks
11  GROUP BY album
12  HAVING max(length) = 250000
13 )

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXX



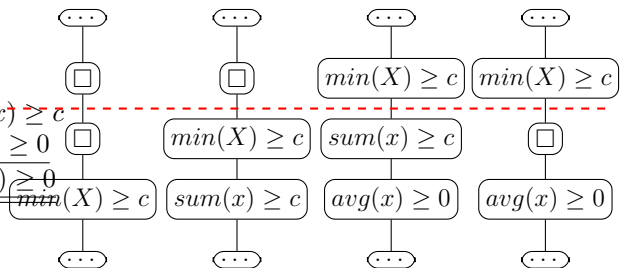
Beispiel für Regelinferenz

Sei $c \geq 0$

Regel 1: $min(X) \geq c \sqsubseteq_K sum(x) \geq c$

Regel 2: $sum(x) \geq c \sqsubseteq_K avg(x) \geq 0$

Neue Regel: $min(X) \geq c \sqsubseteq_K avg(x) \geq 0$



Impressum

Universität Rostock
Institut für Informatik
Lehrstuhl für Datenbank- und Informationssysteme
Albert-Einstein-Straße 22
18059 Rostock

Vertreten durch: Prof. Dr. rer. nat. habil. Andreas Heuer