[ERRATA] Left Join vs Left Outer Join

codigolivro	nomelivro	nomeautor	datalancamento	codigoisbn	codigoeditora
8	Livro 1	[NULL]	[NULL]	354,897	100
14	Livro 2	[NULL]	[NULL]	546,578	100
1,589	Livro 3	[NULL]	[NULL]	15,869	200
2	Livro sem editora	[NULL]	[NULL]	85,472	400

codigoeditora	nome
100	Editora 1
200	Editora 2

```
Select
  L.*,
  ED.Nome as NomeEditora
From
  Livros L Left Join Editora ED on (L.CodigoEditora = ED.CodigoEditora)
Where
  (ED.codigoeditora is null) and
  (L.CodigoEditora is not null);
```

```
Select
  L.*,
  ED.Nome as NomeEditora
From
  Livros L LEFT OUTER Join Editora ED on (L.CodigoEditora = ED.CodigoEditora)
Where
  (ED.codigoeditora is null) and
  (L.CodigoEditora is not null);
```

codigolivro	nomelivro	nomeautor	datalancamento	codigoisbn	codigoeditora	nomeeditora
2	Livro sem editora	[NULL]	[NULL]	85,472	400	[NULL]

Inner Join

codigolivro	nomelivro	nomeautor	datalancamento	codigoisbn	codigoeditora
8	Livro 1	[NULL]	[NULL]	354,897	100
14	Livro 2	[NULL]	[NULL]	546,578	100
1,589	Livro 3	[NULL]	[NULL]	15,869	200
2	Livro sem editora	[NULL]	[NULL]	85,472	400

codigoeditora	nome
100	Editora 1
200	Editora 2

```
Select
  L.*,
  ED.Nome as NomeEditora
From
  Livros L INNER JOIN Editora ED on (L.CodigoEditora = ED.CodigoEditora)
;
```

Join

```
Select
  L.*,
  ED.Nome as NomeEditora
From
  Livros L JOIN Editora ED on (L.CodigoEditora = ED.CodigoEditora)
;
```

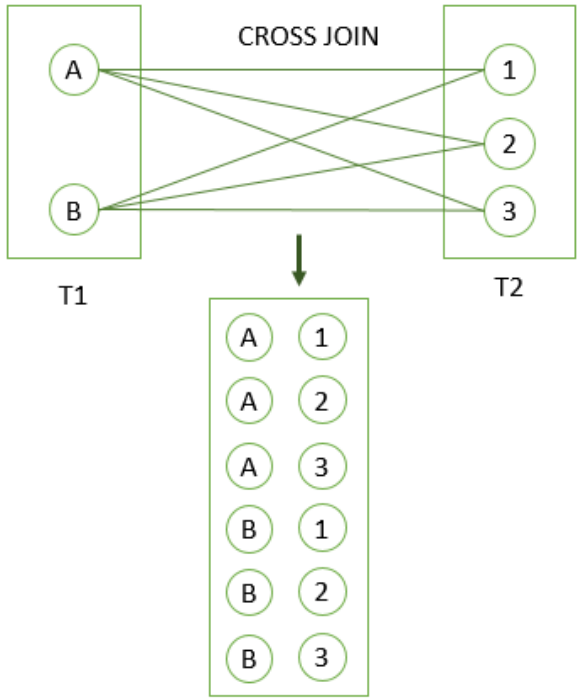
codigolivro	nomelivro	nomeautor	datalancamento	codigoisbn	codigoeditora	nomeeditora
14	Livro 2	[NULL]	[NULL]	546,578	100	Editora 1
8	Livro 1	[NULL]	[NULL]	354,897	100	Editora 1
1,589	Livro 3	[NULL]	[NULL]	15,869	200	Editora 2

Cross Join

```
Select
  *
From
  Livros L CROSS JOIN Editora ED
;
```

codigolivro	nomelivro	nomeautor	datalancamento	codigoisbn	codigoeditora	codigoeditora	nome
8	Livro 1	[NULL]	[NULL]	354,897	100	100	Editora 1
14	Livro 2	[NULL]	[NULL]	546,578	100	100	Editora 1
1,589	Livro 3	[NULL]	[NULL]	15,869	200	100	Editora 1
2	Livro sem editora	[NULL]	[NULL]	85,472	400	100	Editora 1
8	Livro 1	[NULL]	[NULL]	354,897	100	200	Editora 2
14	Livro 2	[NULL]	[NULL]	546,578	100	200	Editora 2
1,589	Livro 3	[NULL]	[NULL]	15,869	200	200	Editora 2
2	Livro sem editora	[NULL]	[NULL]	85,472	400	200	Editora 2

Retorna um Produto Cartesiano entre as Tabelas



Total Recebido por Período

```sql
SELECT
    sum(amount)
FROM
    payment p
WHERE
    payment_date BETWEEN '2007-03-01' AND '2007-03-31';
```

Results 1

SELECT sum(amount)

Grid

	sum
1	23,886.56

Group By – Ranking por Customer/Cliente

```sql
SELECT
    customer_id ,
    sum(amount) AS customer_total
FROM
    payment p
WHERE
    payment_date BETWEEN '2007-03-01' AND '2007-03-31'
GROUP BY
    customer_id
ORDER BY
    customer_total desc, customer_id asc
```

payment 1

SELECT customer_id , sum(amount) AS customer_total

Grid

Text

	customer_id	customer_total
1	148	87.82
2	410	86.83
3	21	79.83
4	15	79.82
5	119	77.82
6	526	76.87
7	147	76.85
8	259	75.84
9	137	74.85
10	144	72.84
11	569	72.83
12	373	71.86
13	468	71.86
14	181	71.84
15	266	71.83

Group By – Ranking por Customer/Cliente

```sql
SELECT
    customer_id ,
    sum(amount) AS customer_total
FROM
    payment p
WHERE
    payment_date BETWEEN '2007-03-01' AND '2007-03-31'
GROUP BY
    customer_id
ORDER BY
    customer_total desc, customer_id asc
```

payment 1

```
SELECT customer_id , sum(amount) AS customer_total
```

Grid | Text

	customer_id	customer_total
1	148	87.82
2	410	86.83
3	21	79.83
4	15	79.82
5	119	77.82
6	526	76.87
7	147	76.85
8	259	75.84
9	137	74.85
10	144	72.84
11	569	72.83
12	373	71.86
13	468	71.86
14	181	71.84
15	266	71.83

Apenas quem comprou acima de X

```sql
SELECT
    customer_id ,
    sum(amount) AS customer_total
FROM
    payment p
WHERE
    payment_date BETWEEN '2007-03-01' AND '2007-03-31'
GROUP BY
    customer_id
HAVING sum(amount) > 70
ORDER BY
    customer_total desc, customer_id ASC
```

	customer_id	customer_total
1	148	87.82
2	410	86.83
3	21	79.83
4	15	79.82
5	119	77.82
6	526	76.87
7	147	76.85
8	259	75.84
9	137	74.85
10	144	72.84
11	569	72.83
12	373	71.86
13	468	71.86
14	181	71.84
15	266	71.83
16	78	70.86
17	198	70.84

Ranking dos 5 que mais alugaram

```sql
SELECT
    customer_id ,
    sum(amount) AS customer_total
FROM
    payment p
WHERE
    payment_date BETWEEN '2007-03-01' AND '2007-03-31'
GROUP BY
    customer_id
ORDER BY
    customer_total desc, customer_id ASC
LIMIT 5
```

customer_id	customer_total
148	87.82
410	86.83
21	79.83
15	79.82
119	77.82

Join Normal

```sql
--Join normal
SELECT a.first_name,
    a.last_name,
    f.title,
    f.description,
    f.release_year
   FROM actor a
     JOIN film_actor fa ON a.actor_id = fa.actor_id
     JOIN film f ON fa.film_id = f.film_id;
```

```sql
EXPLAIN ANALYZE SELECT a.first_name,
    a.last_name,
    f.title,
    f.description,
    f.release_year
   FROM actor a
     JOIN film_actor fa ON a.actor_id = fa.actor_id
     JOIN film f ON fa.film_id = f.film_id;
```

QUERY PLAN
Hash Join (cost=83.00..317.83 rows=5462 width=126) (actual time=0.689..3.447 rows=5462 loops=1)
Hash Cond: (fa.film_id = f.film_id)
-> Hash Join (cost=6.50..166.22 rows=5462 width=15) (actual time=0.104..1.689 rows=5462 loops=1)
Hash Cond: (fa.actor_id = a.actor_id)
-> Seq Scan on film_actor fa (cost=0.00..84.62 rows=5462 width=4) (actual time=0.017..0.391 rows=5462 loops=1)
-> Hash (cost=4.00..4.00 rows=200 width=17) (actual time=0.075..0.075 rows=205 loops=1)
Buckets: 1024 Batches: 1 Memory Usage: 19kB
-> Seq Scan on actor a (cost=0.00..4.00 rows=200 width=17) (actual time=0.011..0.035 rows=205 loops=1)
-> Hash (cost=64.00..64.00 rows=1000 width=117) (actual time=0.579..0.579 rows=1000 loops=1)
Buckets: 1024 Batches: 1 Memory Usage: 157kB
-> Seq Scan on film f (cost=0.00..64.00 rows=1000 width=117) (actual time=0.010..0.292 rows=1000 loops=1)
Planning time: 0.556 ms
Execution time: 3.659 ms

Join Com Where

```sql
--Join usando Where
SELECT a.first_name,
    a.last_name,
    f.title,
    f.description,
    f.release_year
   FROM actor a, film f, film_actor fa
    WHERE  a.actor_id = fa.actor_id
    AND  fa.film_id = f.film_id;
```

```sql
EXPLAIN ANALYZE SELECT a.first_name,
    a.last_name,
    f.title,
    f.description,
    f.release_year
   FROM actor a, film f, film_actor fa
    WHERE  a.actor_id = fa.actor_id
    AND  fa.film_id = f.film_id;
```

QUERY PLAN
Hash Join (cost=83.00..317.83 rows=5462 width=126) (actual time=0.605..7.040 rows=5462 loops=1)
Hash Cond: (fa.film_id = f.film_id)
-> Hash Join (cost=6.50..166.22 rows=5462 width=15) (actual time=0.093..3.609 rows=5462 loops=1)
Hash Cond: (fa.actor_id = a.actor_id)
-> Seq Scan on film_actor fa (cost=0.00..84.62 rows=5462 width=4) (actual time=0.014..0.829 rows=5462 loops=1)
-> Hash (cost=4.00..4.00 rows=200 width=17) (actual time=0.066..0.066 rows=205 loops=1)
Buckets: 1024 Batches: 1 Memory Usage: 19kB
-> Seq Scan on actor a (cost=0.00..4.00 rows=200 width=17) (actual time=0.009..0.032 rows=205 loops=1)
-> Hash (cost=64.00..64.00 rows=1000 width=117) (actual time=0.504..0.504 rows=1000 loops=1)
Buckets: 1024 Batches: 1 Memory Usage: 157kB
-> Seq Scan on film f (cost=0.00..64.00 rows=1000 width=117) (actual time=0.008..0.239 rows=1000 loops=1)
Planning time: 0.511 ms
Execution time: 7.429 ms

Select

```
SELECT a.first_name,
    a.last_name
   FROM actor a
     JOIN film_actor fa ON a.actor_id = fa.actor_id
     JOIN film f ON fa.film_id = f.film_id
ORDER BY a.first_name
```

atores | atores distinct

SELECT a.first_name, a.last_name FROM actor a JC

Grid | Text | Record

	first_name	last_name
1	Adam	Grant
2	Adam	Grant
3	Adam	Hopper
4	Adam	Hopper
5	Adam	Hopper
6	Adam	Hopper
7	Adam	Hopper
8	Adam	Hopper
9	Adam	Hopper
10	Adam	Hopper
11	Adam	Hopper
12	Adam	Hopper
13	Adam	Hopper
14	Adam	Hopper
15	Adam	Hopper
16	Adam	Hopper
17	Adam	Hopper
18	Adam	Hopper
19	Adam	Hopper
20	Adam	Hopper
21	Adam	Grant
22	Adam	Grant

Select Distinct

```
SELECT DISTINCT a.first_name,
    a.last_name
   FROM actor a
     JOIN film_actor fa ON a.actor_id = fa.actor_id
     JOIN film f ON fa.film_id = f.film_id
ORDER BY a.first_name
```

atores | atores distinct

SELECT DISTINCT a.first_name, a.last_name FRO

Grid | Text | Record

	first_name	last_name
1	Adam	Grant
2	Adam	Hopper
3	Al	Garland
4	Alan	Dreyfuss
5	Albert	Johansson
6	Albert	Nolte
7	Alec	Wayne
8	Angela	Witherspoon
9	Angela	Hudson
10	Angelina	Astaire
11	Anne	Cronyn
12	Audrey	Bailey
13	Audrey	Olivier
14	Bela	Walken
15	Ben	Harris
16	Ben	Willis
17	Bette	Nicholson
18	Bob	Fawcett
19	Burt	Temple
20	Burt	Posey
21	Burt	Dukakis
22	Cameron	Streep
23	Cameron	Wray

Join

```
SELECT
    f.title,
    f.description,
    l.name
FROM
    film f
INNER JOIN "language" l ON
    (f.language_id = l.language_id)
ORDER BY
    f.title ;
```

Join | Subquery

SELECT f.title, f.description, l.name FROM film f INNER JOIN | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text

	title	description	name
1	Academy Dinosaur	A Epic Drama of a Feminist And a Mad Scientist who must Battle a Teacher in The Canadian Rockies	English
2	Ace Goldfinger	A Astounding Epistle of a Database Administrator And a Explorer who must Find a Car in Ancient China	English
3	Adaptation Holes	A Astounding Reflection of a Lumberjack And a Car who must Sink a Lumberjack in A Baloon Factory	English
4	Affair Prejudice	A Fanciful Documentary of a Frisbee And a Lumberjack who must Chase a Monkey in A Shark Tank	English
5	African Egg	A Fast-Paced Documentary of a Pastry Chef And a Dentist who must Pursue a Forensic Psychologist in The Gulf of Mexico	English
6	Agent Truman	A Intrepid Panorama of a Robot And a Boy who must Escape a Sumo Wrestler in Ancient China	English
7	Airplane Sierra	A Touching Saga of a Hunter And a Butler who must Discover a Butler in A Jet Boat	English
8	Airport Pollock	A Epic Tale of a Moose And a Girl who must Confront a Monkey in Ancient India	English
9	Alabama Devil	A Thoughtful Panorama of a Database Administrator And a Mad Scientist who must Outgun a Mad Scientist in A Jet Boat	English
10	Aladdin Calendar	A Action-Packed Tale of a Man And a Lumberjack who must Reach a Feminist in Ancient China	English
11	Alamo Videotape	A Boring Epistle of a Butler And a Cat who must Fight a Pastry Chef in A MySQL Convention	English
12	Alaska Phantom	A Fanciful Saga of a Hunter And a Pastry Chef who must Vanquish a Boy in Australia	English
13	Ali Forever	A Action-Packed Drama of a Dentist And a Crocodile who must Battle a Feminist in The Canadian Rockies	English
14	Alice Fantasia	A Emotional Drama of a A Shark And a Database Administrator who must Vanquish a Pioneer in Soviet Georgia	English
15	Alien Center	A Brilliant Drama of a Cat And a Mad Scientist who must Battle a Feminist in A MySQL Convention	English
16	Alley Evolution	A Fast-Paced Drama of a Robot And a Composer who must Battle a Astronaut in New Orleans	English
17	Alone Trip	A Fast-Paced Character Study of a Composer And a Dog who must Outgun a Boat in An Abandoned Fun House	English
18	Alter Victory	A Thoughtful Drama of a Composer And a Feminist who must Meet a Secret Agent in The Canadian Rockies	English
19	Amadeus Holy	A Emotional Display of a Pioneer And a Technical Writer who must Battle a Man in A Baloon	English
20	Amelie Hellfighters	A Boring Drama of a Woman And a Squirrel who must Conquer a Student in A Baloon	English

Sub-Consulta / Sub-Query

```
SELECT
    f.title,
    f.description,
    (SELECT name FROM "language" l WHERE f.language_id = l.language_id) AS name
FROM
    film f
ORDER BY
    f.title ;
```

Join | Subquery

SELECT f.title, f.description, (SELECT name FROM "languag | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text

	title	description	name
1	Academy Dinosaur	A Epic Drama of a Feminist And a Mad Scientist who must Battle a Teacher in The Canadian Rockies	English
2	Ace Goldfinger	A Astounding Epistle of a Database Administrator And a Explorer who must Find a Car in Ancient China	English
3	Adaptation Holes	A Astounding Reflection of a Lumberjack And a Car who must Sink a Lumberjack in A Baloon Factory	English
4	Affair Prejudice	A Fanciful Documentary of a Frisbee And a Lumberjack who must Chase a Monkey in A Shark Tank	English
5	African Egg	A Fast-Paced Documentary of a Pastry Chef And a Dentist who must Pursue a Forensic Psychologist in The Gulf of Mexico	English
6	Agent Truman	A Intrepid Panorama of a Robot And a Boy who must Escape a Sumo Wrestler in Ancient China	English
7	Airplane Sierra	A Touching Saga of a Hunter And a Butler who must Discover a Butler in A Jet Boat	English
8	Airport Pollock	A Epic Tale of a Moose And a Girl who must Confront a Monkey in Ancient India	English
9	Alabama Devil	A Thoughtful Panorama of a Database Administrator And a Mad Scientist who must Outgun a Mad Scientist in A Jet Boat	English
10	Aladdin Calendar	A Action-Packed Tale of a Man And a Lumberjack who must Reach a Feminist in Ancient China	English
11	Alamo Videotape	A Boring Epistle of a Butler And a Cat who must Fight a Pastry Chef in A MySQL Convention	English
12	Alaska Phantom	A Fanciful Saga of a Hunter And a Pastry Chef who must Vanquish a Boy in Australia	English
13	Ali Forever	A Action-Packed Drama of a Dentist And a Crocodile who must Battle a Feminist in The Canadian Rockies	English
14	Alice Fantasia	A Emotional Drama of a A Shark And a Database Administrator who must Vanquish a Pioneer in Soviet Georgia	English
15	Alien Center	A Brilliant Drama of a Cat And a Mad Scientist who must Battle a Feminist in A MySQL Convention	English
16	Alley Evolution	A Fast-Paced Drama of a Robot And a Composer who must Battle a Astronaut in New Orleans	English
17	Alone Trip	A Fast-Paced Character Study of a Composer And a Dog who must Outgun a Boat in An Abandoned Fun House	English
18	Alter Victory	A Thoughtful Drama of a Composer And a Feminist who must Meet a Secret Agent in The Canadian Rockies	English
19	Amadeus Holy	A Emotional Display of a Pioneer And a Technical Writer who must Battle a Man in A Baloon	English
20	Amelie Hellfighters	A Boring Drama of a Woman And a Squirrel who must Conquer a Student in A Baloon	English

Delete No Action / Cascade / Restrict

```
CREATE TABLE order_items (
    product_no integer REFERENCES products ON DELETE RESTRICT,
    order_id integer REFERENCES orders ON DELETE CASCADE,
    quantity integer,
    PRIMARY KEY (product_no, order_id)
);

CREATE TABLE order_items (
    product_no integer REFERENCES products
    order_id integer REFERENCES orders
    quantity integer,
    PRIMARY KEY (product_no, order_id)
);
```

Delete No Action / Cascade / Restrict

No Action:

Nenhuma ação é realizada e um erro é disparado caso exista registro correspondente na tabela Referenciada/relacionada.

Restrict:

Impede que dados relacionados por chave sejam excluídos. A diferença com a opção “No Action” é o momento quando tal verificação é realizada. Com “No Action” ela é realizada após a verificação da existência ou não de dados relacionados.

Cascade:

Os dados relacionados são deletados em cascata – ou seja, em todas as tabelas relacionadas.

Quando não defino a regra, como fazer para deletar em cascata?

Truncate Cascade:

TRUNCATE some_table CASCADE;

Function

Referências:

https://www.postgresql.org/docs/9.5/ddl-constraints.html

https://stackoverflow.com/a/19103574