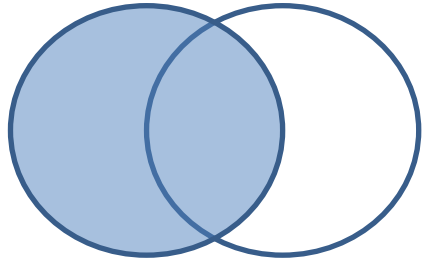


Join Sql

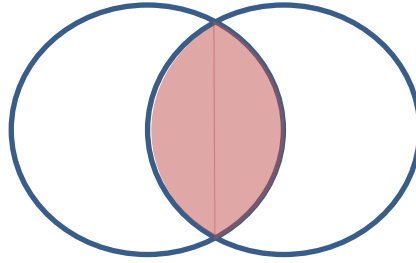
Vitor Vaz da Silva

Não são diagramas de Venn exactos!
 $(A-B) \cup (A \cap B) \neq A$
(porque inclui atributos de B e nulls)

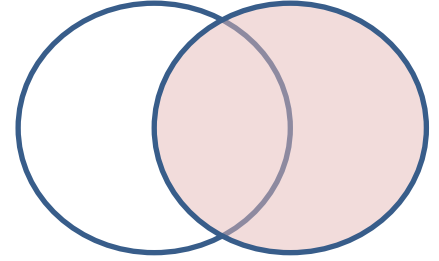


Left Join $(A-B) \cup (A \cap B)$

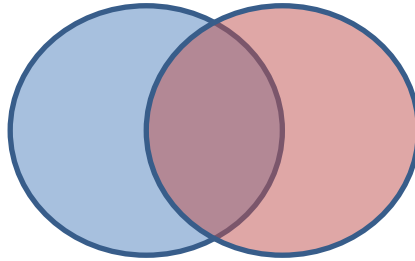
Join



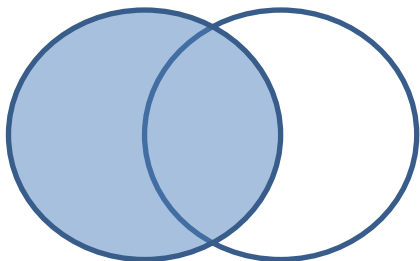
[Inner] Join $A \cap B$



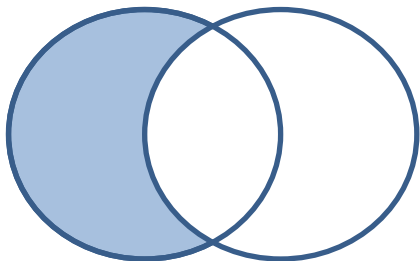
Right Join $(B-A) \cup (A \cap B)$



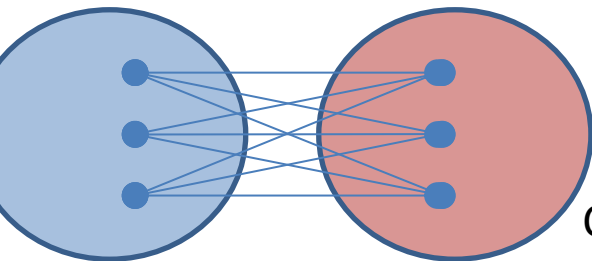
Full Outer Join $(A-B) \cup (B-A) \cup (A \cap B)$



Left [outer] Join

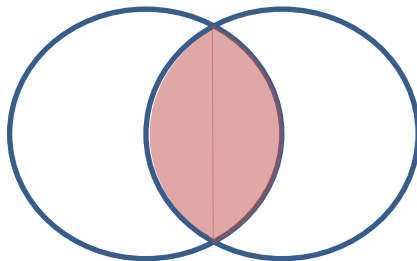


Left anti Join
Left excluding Join

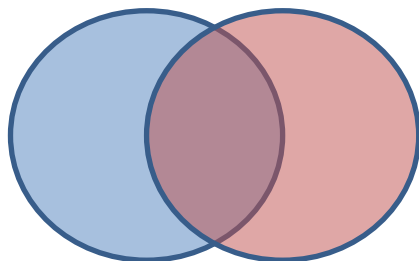


Cross Join

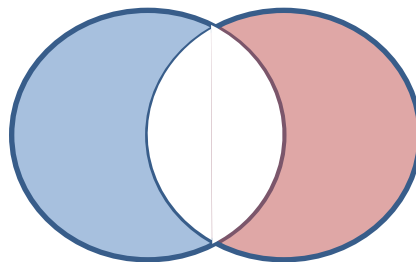
Join



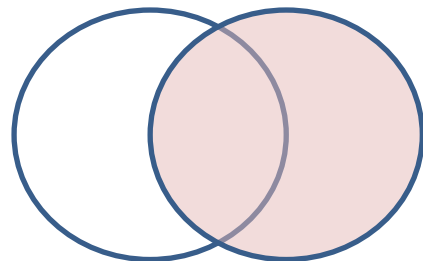
[Inner | Equi | Natural] Join



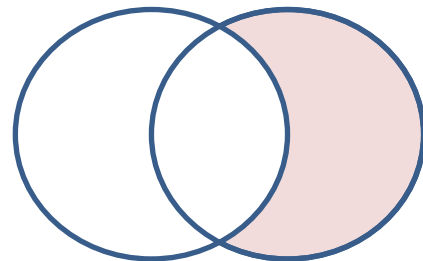
Full Outer Join



Full anti Join
Full Outer excluding Join



Right [outer] Join



Right anti Join
Right excluding Join

Exemplo

Tabela

Numero (1 a 24)

Par (2 a 20)

Impar (1 a 19)

Primo (1 a 19)

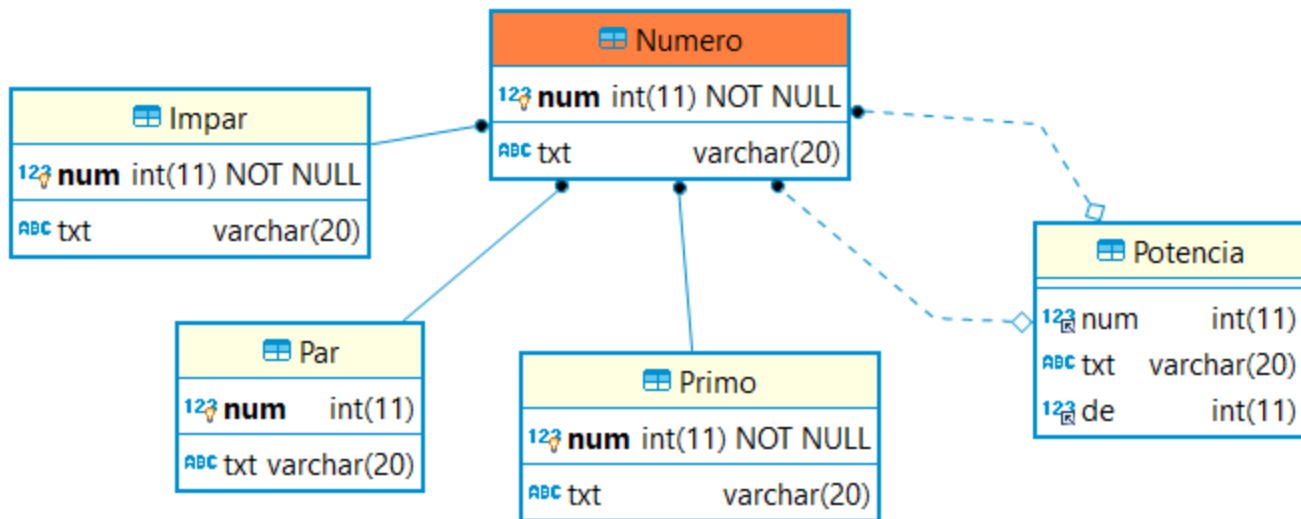
Potencia {1, 4, 9, 16}

```
create table Numero(  
    num int,  
    txt varchar(20),  
    constraint pk_numero primary key (num)  
) engine= InnoDB default character set=utf8; -- Para MySQL garantir as restrições de integridade
```

```
create table Par(  
    num int unique,  
    txt varchar(20),  
    constraint pk_par foreign key (num) references Numero(num)  
) engine= InnoDB default character set=utf8;
```

```
create table Impar(  
    num int primary key,  
    txt varchar(20),  
    foreign key (num) references Numero(num)  
) engine= InnoDB default character set=utf8;
```

```
create table Primo(  
    num int,  
    txt varchar(20),  
    foreign key (num) references Numero(num),  
    primary key(num)  
) engine= InnoDB default character set=utf8;  
  
create table Potencia(  
    num int,  
    txt varchar(20),  
    de int,  
    foreign key (num) references Numero(num),  
    foreign key (de) references Numero(num)  
) engine= InnoDB default character set=utf8;
```



insert into Numero(num,txt) values

(1,'um'), (2,'dois'), (3,'três'), (4, 'quatro'), (5,'cinco'), (6,'seis'), (7,'sete'), (8,'oito'), (9, 'nove'), (10,'dez'),
(11,'onze'), (12,'doze'), (13,'treze'), (14, 'catorze'), (15,'quinze'), (16,'dezseis'), (17,'dezsete'), (18,'dezoito'),
(19, 'deznove') , (20, 'vinte'), (21,'vinte e um'),(22, 'vinte e dois'), (23, 'vinte e três'), (24, 'vinte e quatro');

insert into Par(num,txt) values (2,'two'), (4, 'four'), (6,'six'), (8,'eight'), (10,'ten'), (12,'twelve'),
(14, 'fourteen'), (16,'sixteen'), (18,'eighteen'),
(20, 'vinte'), (22, 'vinte e dois') -- dois últimos em português tal como em Numero

insert into Impar(num,txt) values (1,'one'), (3,'three'), (5,'five'), (7,'seven'), (9, 'nine'), (11,'eleven'),
(13,'thirteen'), (15,'fifteen'), (17,'seventeen'), (19, 'nineteen');

insert into Primo(num,txt) values (1,'I'), (2,'II'), (3,'III'), (5,'V'), (7,'VII'), (11,'XI'),(13,'XIII'), (17,'XVII'),
(19, 'XIX'), (23, 'XXIII') ;

insert into Potencia(num,txt,de) values (1,'1^2',1), (4,'2^2',2), (9,'3^2',3), (16,'4^2',4);

Numero	Par	Impar	Primo	Potencia
num--txt	num--txt	num--txt	num--txt	num---txt---de
1----um	2----two	1----one	1----I	1----1^2----1
2----dois	4----four	3----three	2----II	4----2^2----2
3----três	6----six	5----five	3----III	9----3^2----3
4----quatro	8----eight	7----seven	5----V	16----4^2----4
5----cinco	10----ten	9----nine	7----VII	
6----seis	12----twelve	11----eleven	11----XI	
7----sete	14----fourteen	13----thirteen	13----XIII	
8----oito	16----sixteen	15----fifteen	17----XVII	
9----nove	18----eighteen	17----seventeen	19----XIX	
10----dez	20----vinte	19----nineteen	23----XXIII	
11----onze	22----vinte e dois			
12----doze				
13----treze				
14----catorze				
15----quinze				
16----dezseis				
17----dezsete				
18----dezoito				
19----deznove				
20----vinte				
21----vinte e um				
22----vinte e dois				
23----vinte e três				
24----vinte e quatro				

Exemplo

-- Todos os Numero que também são Par

select * from Numero, Par where Numero.num=Par.num;

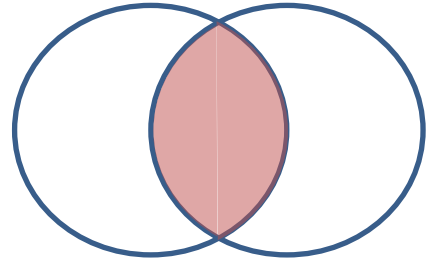
select * from Numero join Par on (Numero.num=Par.num);

select * from Numero join Par using (num);

select * from Numero join Par;

select * from Numero inner join Par;

select * from Numero natural join Par; -- *entre colunas com o mesmo nome*



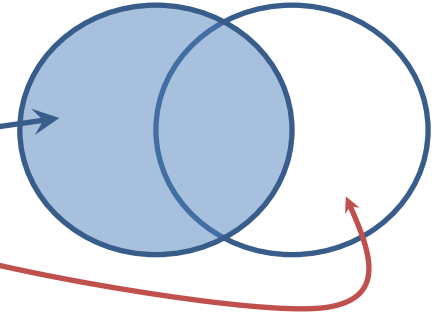
Exemplo

-- Todos os Primo que são ou não Par

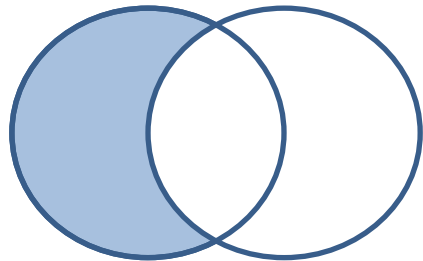
select * from Potencia left join Par **using(num) order by num;**

select * from Potencia left join Par **on(Potencia .num= Par.num) order by Potencia.num;**

**select * from (
 select N.num, N.txt, P.num as nnn, P.txt as ttt
 from Potencia as N, Par as P where N.num=P.num
 union
 select N.num, N.txt, null, null
 from Potencia as N where N.num not in (select num from Par)
) as R order by num;**



Exemplo



-- Todos os Numero que não são Par

```
select * from Numero as N where N.num not in (select num from Par);
```

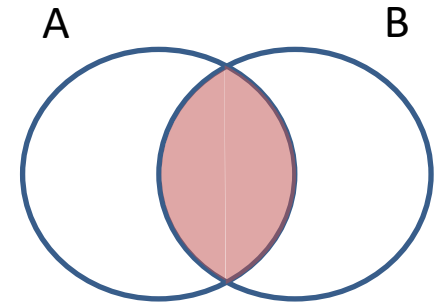
```
select N.num, N.txt, P.num, P.txt  
      from Numero as N left join Par as P on (N.num=P.num)  
      where P.num is null;
```

```
select * from Numero left join Par using(num) where Par.num is null;
```

inner join

```
select * from A join B on (A.chave=B.chave)
```

```
select * from A, B where A.chave=B.chave
```

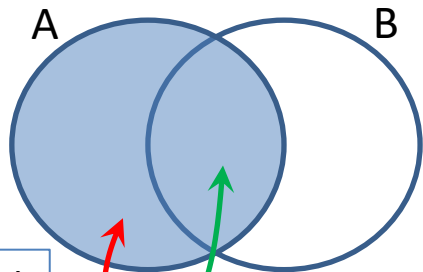


A(a, b, chave)
B(c, chave)

left outer join

```
select * from A left join B on (A.chave=B.chave)
```

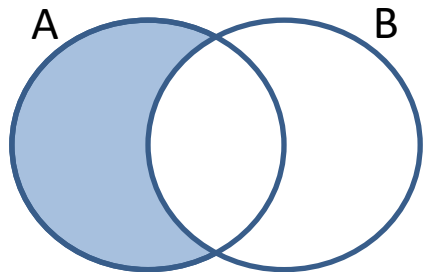
A(a, b, chave)
B(c, chave)



```
select a, b, c from A, B  
    where A.chave=B.chave  
union  
select a, b, null from A  
    where A.chave not in (select chave from A, B where A.chave=B.chave)
```

```
select a, b, c from A, B where A.chave=B.chave  
union  
select a, b, null from A where A.chave not in (select chave from B)
```

left outer anti join



```
select * from A left join B on (A.chave=B.chave) where B.chave is null
```

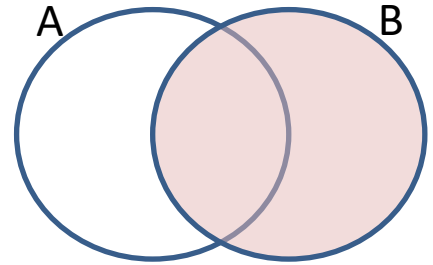
A(a, b, chave)
B(c, chave)

```
select a, b, null as c from A where A.chave not in (select chave from B)
```

```
select * from A where not exists (select * from B where A.chave=B.chave)
```

right outer join

```
select * from A right join B on (A.chave=B.chave)
```



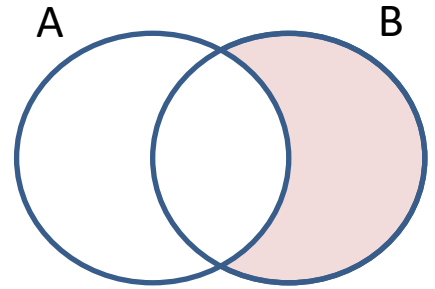
A(a, b, chave)
B(c, chave)

```
select a, b, c from A, B where A.chave=B.chave
```

```
union
```

```
select null, null, c from B where B.chave not in (select chave from A)
```

right outer anti join

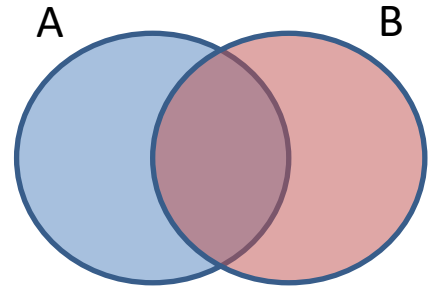


```
select * from A right join B on (A.chave=B.chave) where A.chave is null
```

A(a, b, chave)
B(c, chave)

```
select null as a, null as b, c from B where B.chave not in (select chave from A)
```


full outer join



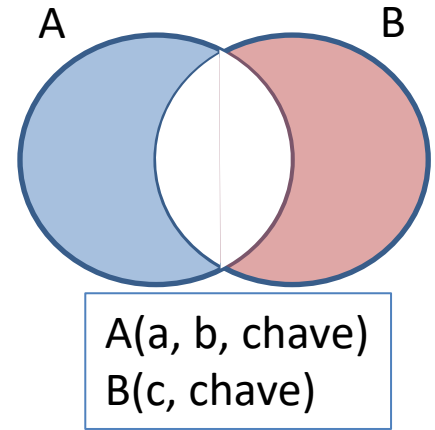
```
select * from A full join B on (A.chave=B.chave)
```

A(a, b, chave)
B(c, chave)

```
select * from A left join B on (A.chave=B.chave)  
union  
select * from A right join B on (A.chave=B.chave)
```

```
select a, b, c from A, B where A.chave=B.chave  
union  
select a, b, null from A where A.chave not in (select chave from B)  
union  
select null, null, c from B where B.chave not in (select chave from A)
```

full outer anti join



```
select * from A full join B on (A.chave=B.chave) where (B.chave is null or A.chave is null)
```

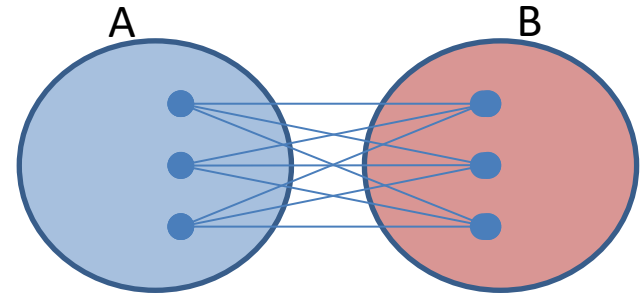
```
select * from A left join B on (A.chave=B.chave) where B.chave is null  
union  
select * from A right join B on (A.chave=B.chave) where A.chave is null
```

```
select a, b, null from A where A.chave not in (select chave from B)  
union  
select null, null, c from B where B.chave not in (select chave from A)
```

cross join

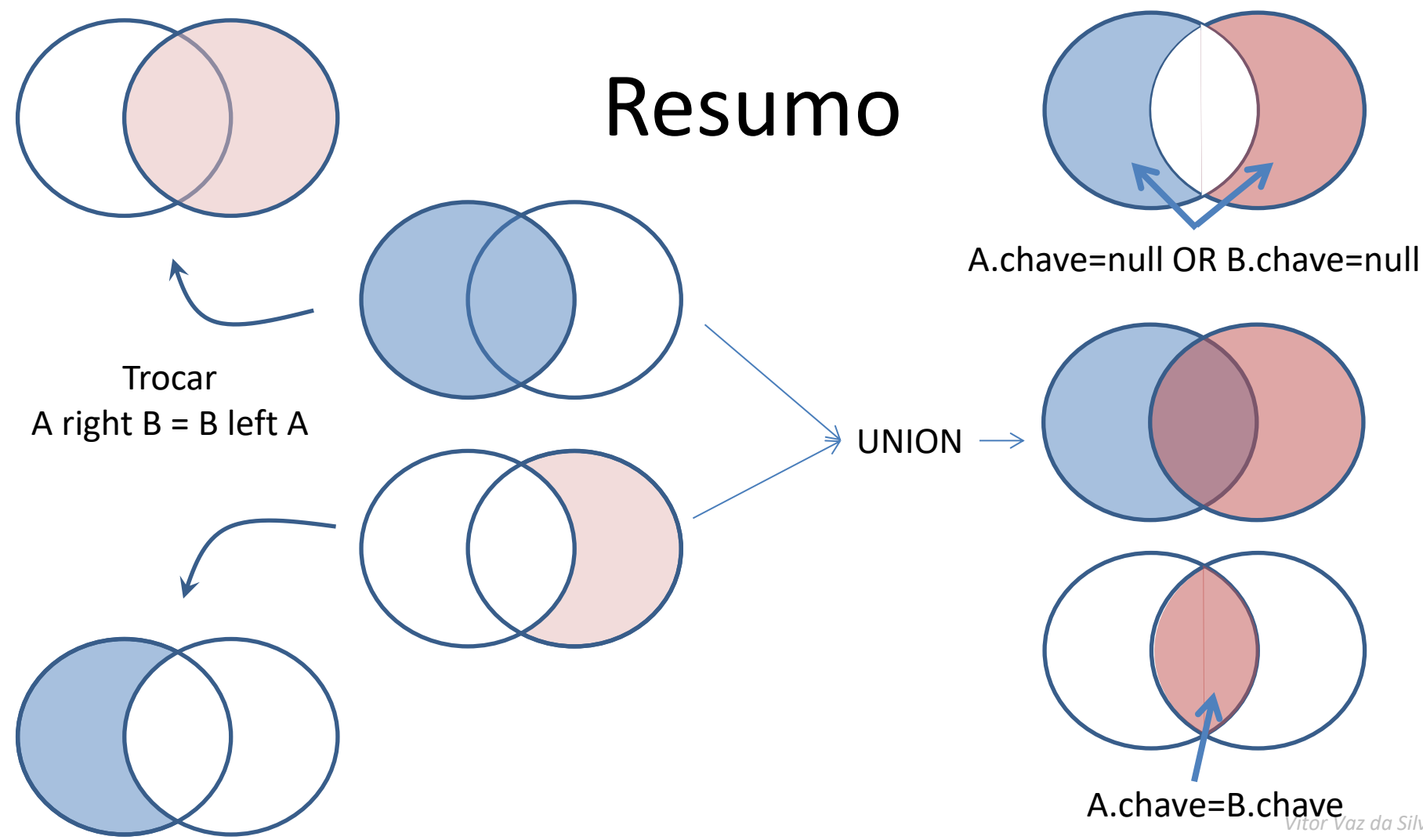
`select * from A cross join B`

`select * from A, B`



A(a, b, chave)
B(c, chave)

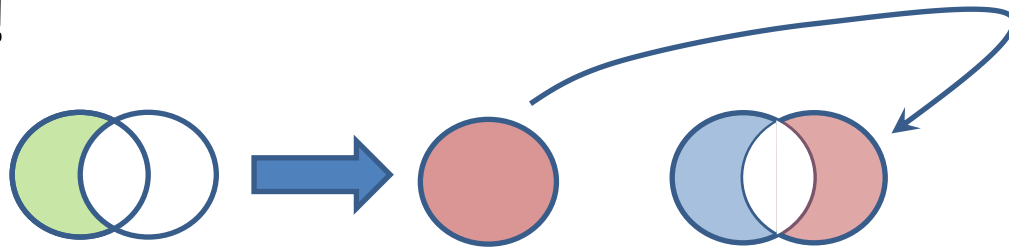
Resumo



Escreva as expressões SQL para responder a:

1. Todos os Primo sejam ou não Impar
2. Todos os Numero que são Primo e não Par
3. Todos os Par que são Potencia
4. Invente!!!!

Recorde:



Surrogate Key

R1(a,b,c,d)

R2(a,b,c, e, f)

R3(a,b,c, g,h,j)

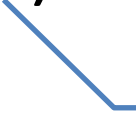
S(a,b,c,x)

R1(x,d)

R2(x, f)

R3(x, g,h,j)

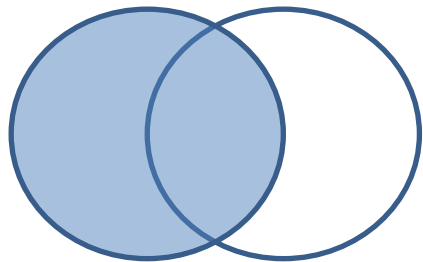
Unique not
null



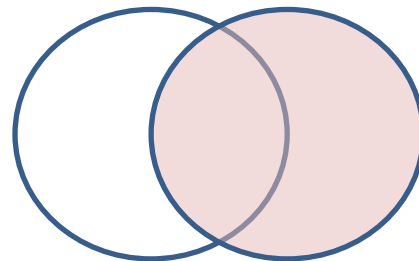
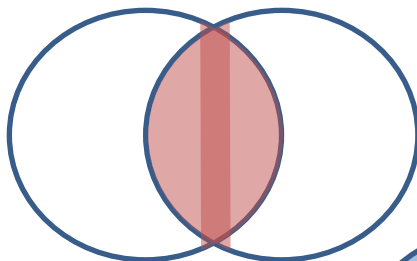
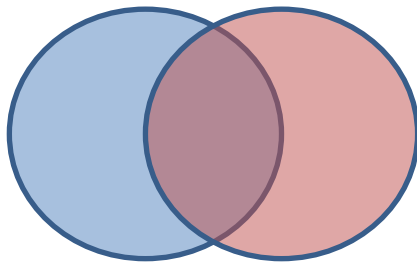
- Exemplo de atributo derivado (no select)
- $cc = \{ \{a,b\}, \{c\}, \{d\} \}$ $cp = \{a,b\}$

Join

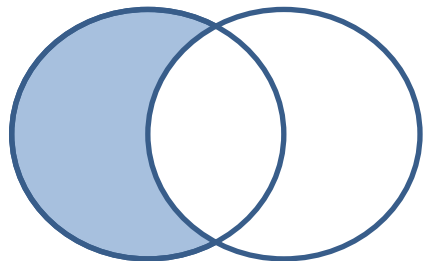
Não são diagramas de Venn!



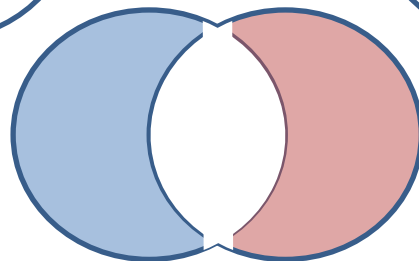
Left Join



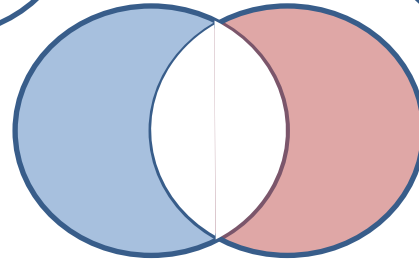
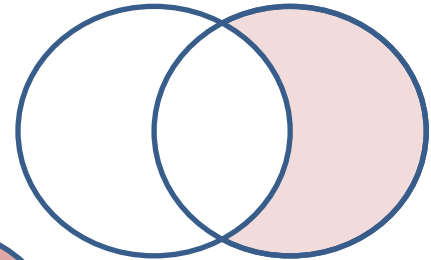
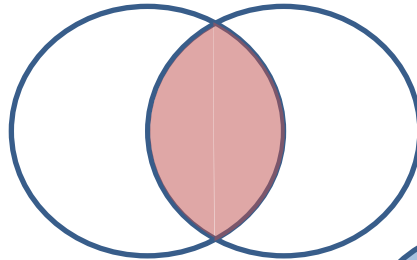
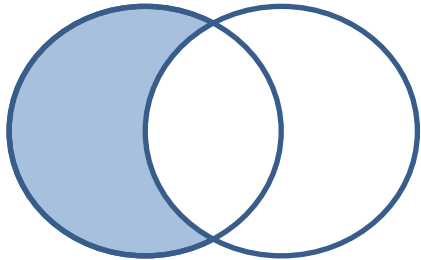
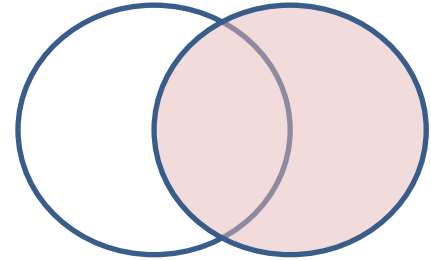
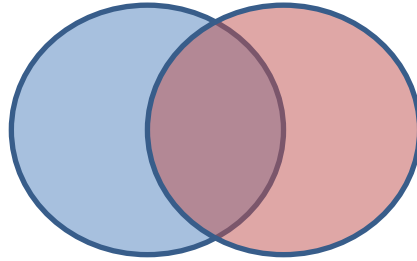
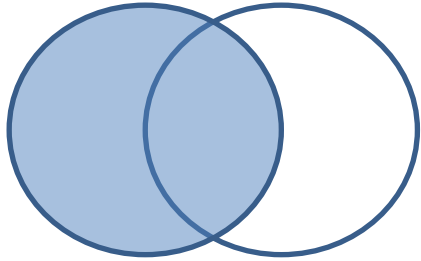
Right Join



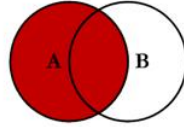
Left Join



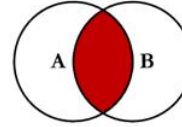
Join



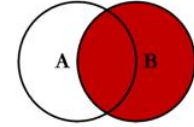
N1QL JOINS



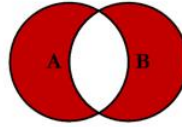
```
SELECT <select_list>
FROM Table_A A
LEFT JOIN Table_B B
ON KEYS <keys-clause(A)>
```



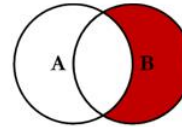
```
SELECT <select_list>
FROM Table_A A
INNER JOIN Table_B B
ON KEYS <keys-clause(A)>
```



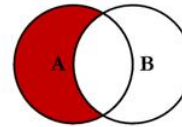
```
SELECT <select_list>
FROM Table_B B
LEFT JOIN Table_A A
ON KEYS <keys-clause(B)>
```



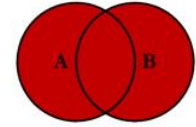
```
SELECT <select_list>
FROM Table_B B
LEFT JOIN Table_A A
ON KEYS <keys-clause(B)>
WHERE META(A).id IS MISSING
UNION ALL
SELECT <select_list>
FROM Table_A A
LEFT JOIN Table_B B
ON KEYS <keys-clause(A)>
WHERE META(B).id IS MISSING
```



```
SELECT <select_list>
FROM Table_B B
LEFT JOIN Table_A A
ON KEYS <keys-clause(B)>
WHERE META(B).id IS MISSING
```



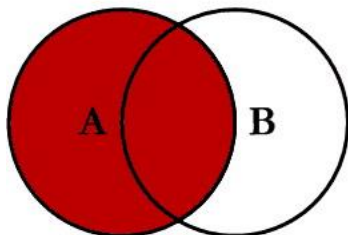
```
SELECT <select_list>
FROM Table_A A
LEFT JOIN Table_B B
ON KEYS <keys-clause(A)>
WHERE META(B).id IS MISSING
```



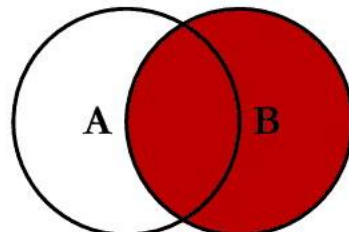
```
SELECT <select_list>
FROM Table_B B
LEFT JOIN Table_A A
ON KEYS <keys-clause(B)>
UNION ALL
SELECT <select_list>
FROM Table_A A
LEFT JOIN Table_B B
ON KEYS <keys-clause(A)>
WHERE META(B).id IS MISSING
```

@atom_yang, 2017

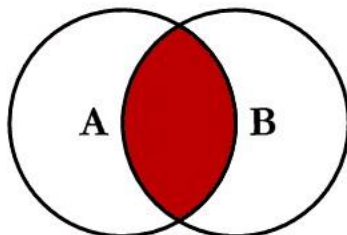
SQL JOINS



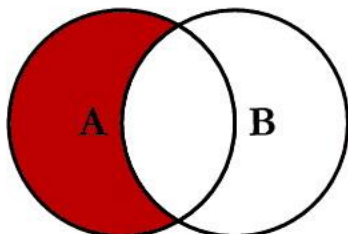
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



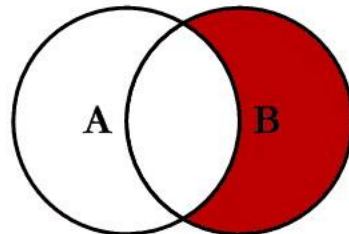
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



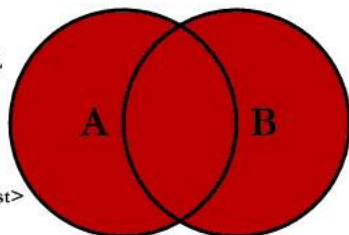
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



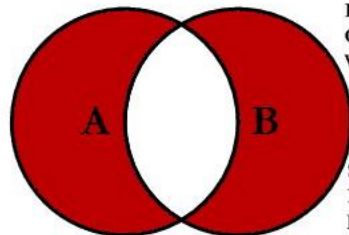
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```

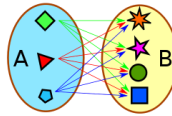


```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

Select (campos)
From A InnerJoin B
On A.Clave = B.Clave

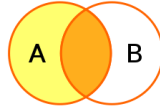


Select (campos)
From A Cross Join B

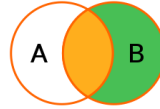


Joins del SQL

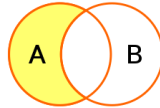
Select (campos)
From A Left Join B
On A.Clave = B.Clave



Select (campos)
From A Right Join B
On A.Clave = B.Clave



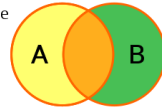
Select (campos)
From A Left Join B
On A.Clave = B.Clave
Where B.Clave is Null



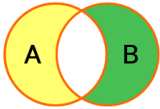
Select (campos)
From A Right Join B
On A.Clave = B.Clave
Where A.Clave is Null



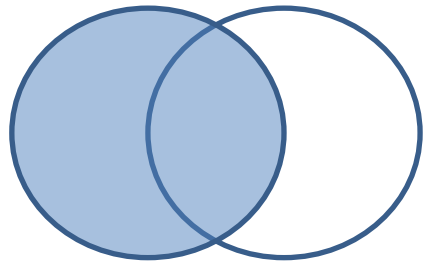
Select (campos)
From A Full Join B
On A.Clave = B.Clave



Select (campos)
From A Right Join B
On A.Clave = B.Clave
Where (A.Clave is Null) Or (B.Clave is Null)



Exemplo



-- Todos os Numero que são ou não Par

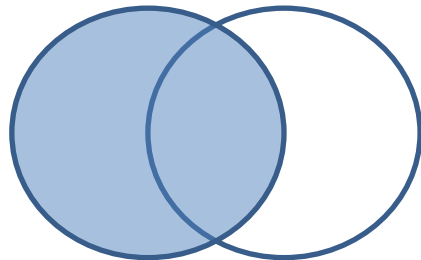
select * from Numero as N where N.num in (select num from Par)

**select N.num, N.txt, P.num, P.txt
from Numero as N left join Par as P on (N.num=P.num)**

select * from Numero left join Par using(num)

Exemplo

-- Todos os Primo que são ou não Par



```
select * from Potencia left join Par using(num) order by num
```

```
select * from Potencia left join Par on(Potencia .num= Par.num) order by Potencia.num
```

```
select * from (  
  select N.num, N.txt, P.num as nnn, P.txt as ttt  
    from Potencia as N, Par as P where N.num=P.num  
  union  
  select N.num, N.txt, null as n, null as t  
    from Potencia as N where N.num not in (  
      select P.num from Potencia as N, Par as P where N.num=P.num  
    )  
) as R order by num
```