

student

	<u>id</u>	<u>name</u>	<u>dob</u>	<u>group</u>	<u>group_type</u>
1	1	Jim	1975-05-07	BE1	backend
2	11	Jack	1994-05-06	BE1	backend
3	20	Alex	1993-08-09	BE2	backend
4	42	Alexander II	1985-01-01	BE2	backend
5	3	Jackson	1996-04-06	BE3	backend
6	2	Sergio	1997-03-04	BE3	backend
7	41	Alexander I	1997-04-05	BE4	backend
8	30	Marion	1997-07-07	BE5	backend
9	31	Mario	1996-03-04	BE5	backend
10	10	Jeremy	1998-07-08	BE5	backend

grouptable

group

<u>id</u>	<u>type</u>	<u>number</u>
1	backend	BE1
		BE2
		3
		5

Students

	id	name	dob
1	1	Jim	1975-05-07
2	2	Sergio	1997-03-04
3	3	Jackson	1996-04-06
4	10	Jeremy	1998-07-08
5	11	Jack	1994-05-06
6	20	Alex	1993-08-09
7	30	Marion	1997-07-07
8	31	Mario	1996-03-04
9	41	Alexander I	1997-04-05
10	42	Alexander II	1985-01-01

Groups

	id	name	typ
1	BE1	backend	
2	BE2	backend	
3	BE3	backend	
4	BE4	backend	
5	BE5	backend	

	id	name	dob	group_id
1	Jim	1975-05-07		1
2	Sergio	1997-03-04		1
3	Jackson	1996-04-06		2
10	Jeremy	1998-07-08		2
11	Jack	1994-05-06		3
15	Mickey	2001-02-03		<null>
20	Alex	1993-08-09		3
30	Marion	1997-07-07		3
31	Mario	1996-03-04		4
41	Alexander I	1997-04-05		5
42	Alexander II	1985-01-01		5

	id	name	typ
1	BE1	backend	
2	BE2	backend	
3	BE3	backend	
4	BE4	backend	
5	BE5	backend	
6	FE1	frontend	
7	FE2	frontend	
8	FE3	frontend	



The diagram illustrates JOIN operations between three tables: Person, Group, and Type.

Tables:

- Person (p):** id (1-42), name, dob.
- Group (g):** id (1-5), group_id (1-5), name (BE1-BE5, FE1-FE3).
- Type (t):** id (1-8), typ (backend, frontend).

Annotations:

- Left Outer Join:** Shows a join from Person to Group. A circled 'P' is at the top of the Group table, with arrows pointing to each row. A circled '<null>' is at the bottom of the Group table, with an arrow pointing to the last row. Labels 'left' and 'Right' are on the left.
- Inner Join:** Shows a join from Person to Group. Labels 'left' and 'right' are above the tables. A circled 'P' is at the top of the Group table, with arrows pointing to each row. A circled '<null>' is at the bottom of the Group table, with an arrow pointing to the last row. Labels '100% ↔ 100%' are between the tables. Below the tables, it says 'INNER JOIN' and 'no orphan data'.
- Left Outer Join:** Shows a join from Person to Group. Labels 'left' and 'right' are below the tables. A circled 'P' is at the top of the Group table, with arrows pointing to each row. A circled '<null>' is at the bottom of the Group table, with an arrow pointing to the last row. Labels '100% → <100%' are between the tables. Below the tables, it says 'LEFT JOIN'.
- Left Outer Join with WHERE:** Shows a join from Person to Group where p.group_id is null. Labels 'left' and 'right' are below the tables. A circled 'P' is at the top of the Group table, with arrows pointing to each row. A circled '<null>' is at the bottom of the Group table, with an arrow pointing to the last row. Labels '100% → <100%' are between the tables. Below the tables, it says 'LEFT JOIN'.
- Final Result:** Shows the final result of the LEFT JOIN operation, combining all rows from Person with corresponding rows from Group and Type. It includes a circled 'P' at the top of the Group table and a circled '<null>' at the bottom.

```
select p.id, p.name, p.dob, g.name
from person p
right outer join group g on (g.id = p.group_id)
order by p.name;
```

≤ 100 , — 100
RIGHT JOIN

p.id	p.name	p.dob	g.name
20	Alex	1993-08-09	BE3
41	Alexander I	1997-04-05	BE5
42	Alexander II	1985-01-01	BE5
11	Jack	1994-05-06	BE3
3	Jackson	1996-04-06	BE2
10	Jeremy	1998-07-08	BE2
1	Jim	1975-05-07	BE1
31	Mario	1996-03-04	BE4
30	Marion	1997-07-07	BE3
2	Sergio	1997-03-04	BE1
<null>	<null>	<null>	FE1
<null>	<null>	<null>	FE2
<null>	<null>	<null>	FE3

```
select p.id, p.name, p.dob, g.name
from person p
full outer join group g on (g.id = p.group_id)
order by p.name;
```

p.id	p.name	p.dob	g.name
20	Alex	1993-08-09	BE3
41	Alexander I	1997-04-05	BE5
42	Alexander II	1985-01-01	BE5
11	Jack	1994-05-06	BE3
3	Jackson	1996-04-06	BE2
10	Jeremy	1998-07-08	BE2
1	Jim	1975-05-07	BE1
31	Mario	1996-03-04	BE4
30	Marion	1997-07-07	BE3
15	Mickey	2001-02-03	<null>
2	Sergio	1997-03-04	BE1
<null>	<null>	<null>	FE1
<null>	<null>	<null>	FE2
<null>	<null>	<null>	FE3

right join to be
All records from right
and absent from left
are null

we take
All from left
All from right
missing are null

name
Jim
Sergio
Jackson
Jeremy
Jack
Mickey
Alex
Marion
Mario
Alexander I
Alexander II

name
1 Marion
2 Jack
3 Jim
4 Alex
5 Sergio
6 Jackson
7 Jeremy
8 Alexander II
9 Mickey
10 Alexander III
11 Maria

INNER JOIN

OUTER JOIN

JOIN

LEFT OUTER JOIN

RIGHT OUTER JOIN

JOIN

```
select p.name, s.name
from person p
full outer join student s on (s.name = p.name)
where p.name is null OR s.name is null
```

p.name	s.name
1 <null>	Alexander III
2 <null>	Maria
3 Alexander I	<null>
4 Mario	<null>

```
select p.name, s.name
from person p
full outer join student s on (s.name = p.name)
```

SQL : f horizontal projection
+ vertical projection
f tables combining

cols

rows

join

rule for join

```
select p.id, p.name, p.dob, p.name, gt.name
from person p
left outer join group g on g.id = p.group_id
left outer join gtype gt on gt.id = g.type_id
where gt.name = 'backend'
```

+ aggregation

aggregation

- ✓ min
 - ✓ max
 - ✓ avg
 - ✓ count
 - ✓ sum
 - ✓ groupby

```
select count(*)  
from person p  
left outer join group g on g.id = p.group_id  
left outer join gtype gt on gt.id = g.type_id  
where gt.name = 'backend'  
AND g.name = 'BES';
```

count ↴
2

```
select min(g.name), max(g.name)
from person p
left outer join groupp g on g.id = p.group_id
left outer join gtype gt on gt.id = g.type_id
where gt.name = 'backend';
```

min max

```
select sum(p.id) from person p
```

206

```
select avg(p.id) from person p
```

avg ↴

```
select *  
from person p  
left outer join group g on g.id = p.group_id  
left outer join gtype gt on gt.id = g.type_id;
```

how many student
in each group

BF_1 -2
 BF_2 2
 BF_3 -3

BEG-1
BES-2