Basic Select

select name from customers where state = 'RS'

[Customer Address](https://www.beecrowd.com.br/judge/es/problems/view/2603)

select name, street from customers where city = 'Porto Alegre'

[Under 10 or Greater Than 100](https://www.beecrowd.com.br/judge/es/problems/view/2604)

select id, name from products where price<10 or price>100

[Executive Representatives](https://www.beecrowd.com.br/judge/es/problems/view/2605)

select products.name, providers.name

from products

join providers on products.id\_providers = providers.id

join categories on products.id\_categories = categories.id

where categories.id = 6

[Categories](https://www.beecrowd.com.br/judge/es/problems/view/2606)

select products.id, products.name from products

join categories on products.id\_categories = categories.id

where categories.name like 'super%'

[Providers' City in Alphabetical Order](https://www.beecrowd.com.br/judge/es/problems/view/2607)

select distinct city from providers order by city DESC

Provider Ajax SA

select products.name, providers.name

from products

join providers on products.id\_providers = providers.id

where providers.name = 'Ajax SA'

Higher and Lower Price

select max(price), min(price) from products

Products by Categories

select categories.name, sum(products.amount) from categories join products on categories.id = products.id\_categories group by categories.name

Average Value of Products

select cast(avg(price) as decimal (10,2)) as price from products

Action movies

select movies.id, movies.name from movies

join genres on genres.id = movies.id\_genres

where genres.description = 'Action'

September rentals

select customers.name, rentals.rentals\_date from customers

join rentals on customers.id = rentals.id\_customers

where rentals.rentals\_date between '2016-09-01' and '2016-09-30'

Expanding the business

select distinct city from customers

No rental

select customers.id, customers.name from customers where customers.id not in (select locations.id\_customers from locations)

Imported products

select products.name, providers.name, categories.name from products

join providers on products.id\_providers = providers.id

join categories on products.id\_categories = categories.id

where providers.name = 'Sansul SA' and categories.name = 'Imported'

Super luxury

select products.name, providers.name, products.price from products

join providers on products.id\_providers = providers.id

join categories on products.id\_categories = categories.id

where products.price > 1000 and categories.name = 'Super Luxury'

Orders in First Half

select customers.name, orders.id from orders

join customers on orders.id\_customers = customers.id

where orders.orders\_date between '2016-01-01' and '2016-06-30'

Amounts between 10 and 20

select products.name from products

join providers on providers.id = products.id\_providers

where (products.amount between 10 and 20) and providers.name like 'P%'

Legal person

select name from customers

join legal\_person on customers.id = legal\_person.id\_customers

Categories with various products

select products.name, categories.name from products

join categories on categories.id = products.id\_categories

where products.amount > 100

and categories.id in (1, 2, 3, 6, 9) order by categories.id asc

CPF Validation

select concat(substring(cpf,1,3),'.', substring(cpf,4,3),'.', substring(cpf,7,3), '-', substring(cpf,10,2)) from natural\_person

where id\_customers in (select id from customers)

Lawyers

select concat(substring(cselect name, customers\_number

from lawyers

where customers\_number = (

select max(customers\_number) from lawyers

)

union all

select name, customers\_number

from lawyers

where customers\_number = (

select min(customers\_number)

from lawyers

)

union all

select 'Average' AS name, Round(Avg(customers\_number), 0) AS customers\_number

from lawyers;

Contest

select

candidate.name,

cast((((score.math\*2) + (score.specific\*3) + (score.project\_plan\*5)) / 10) as decimal(10, 2)) avg

from candidate

join score on candidate.id = score.candidate\_id order by avg desc

Payday

SELECT name,cast(EXTRACT(DAY FROM payday) as integer)as day from loan;

League

(select concat('Podium: ',team)as name  
from league  
order by position  
limit 3)  
union all  
(select concat('Demoted: ',team)as name  
from league  
where position>any(select count(\*)-2 from league))

Students Grades

SELECT concat('Approved: ', name) as name, grade

FROM students

WHERE grade >= 7

ORDER BY grade DESC

Richard’s Multiverse

SELECT l.name, round((l.omega \* 1.618), 3) AS "Fator N"

FROM life\_registry l JOIN dimensions d ON

l.dimensions\_id = d.id

WHERE d.name in ('C875', 'C774') AND l.name LIKE 'Richard%'

ORDER BY l.omega ASC

Number of Characters

SELECT name, char\_length(name) AS length

FROM people

ORDER BY length DESC

Passwords

select a.id, a.password, MD5(a.password) as MD5 from account a;

Taxes

SELECT name, round((salary \* 0.10), 2) AS tax

FROM people

WHERE salary > 3000

Viruses

select replace(name,'H1','X')as name  
from virus;

Cearense Championship

SELECT

(

SELECT name

FROM teams t

WHERE t.id = team.id

) as name,

(

SELECT count(team\_1)

FROM matches

WHERE team\_1 = team.id

)+(

SELECT count(team\_2)

FROM matches WHERE

team\_2 = team.id

) as matches,

(

SELECT sum(case when team\_2\_goals > team\_1\_goals then 1 else 0 END) as victories

FROM teams t INNER JOIN matches m ON t.id = m.team\_2

WHERE t.id = team.id

)+(

SELECT sum(case when team\_1\_goals > team\_2\_goals then 1 else 0 END)

FROM teams t INNER JOIN matches m ON t.id = m.team\_1

WHERE t.id = team.id

) as victories,

(

SELECT sum(case when team\_2\_goals < team\_1\_goals then 1 else 0 END) as victories

FROM teams t INNER JOIN matches m ON t.id = m.team\_2

WHERE t.id = team.id

)+(

SELECT sum(case when team\_1\_goals < team\_2\_goals then 1 else 0 END)

FROM teams t INNER JOIN matches m ON t.id = m.team\_1

WHERE t.id = team.id

) as defeats,

(

SELECT sum(case when team\_2\_goals = team\_1\_goals then 1 else 0 END) as victories

FROM teams t INNER JOIN matches m ON t.id = m.team\_2

WHERE t.id = team.id

)+(

SELECT sum(case when team\_1\_goals = team\_2\_goals then 1 else 0 END)

FROM teams t INNER JOIN matches m ON t.id = m.team\_1

WHERE t.id = team.id

) as draws,

(

SELECT sum(case when team\_2\_goals > team\_1\_goals then 3 when team\_2\_goals = team\_1\_goals then 1 else 0 END) as victories

FROM teams t INNER JOIN matches m ON t.id = m.team\_2

WHERE t.id = team.id

)+(

SELECT sum(case when team\_1\_goals > team\_2\_goals then 3 when team\_1\_goals = team\_2\_goals then 1 else 0 END)

FROM teams t INNER JOIN matches m ON t.id = m.team\_1

WHERE t.id = team.id

) as score

FROM teams team

ORDER BY score DESC

Employees CPF

SELECT e.cpf, e.enome, d.dnome

FROM departamentos d INNER JOIN empregados e ON d.dnumero = e.dnumero

LEFT JOIN trabalha t ON t.cpf\_emp = e.cpf

LEFT JOIN projetos p ON t.pnumero = p.pnumero

WHERE p.pnumero IS NULL

ORDER BY e.cpf