SELECT username,nombre,email, GROUP\_CONCAT(roles.name) FROM usuario

JOIN role\_user ON role\_user.user\_id = usuario.id JOIN roles ON roles.id=role\_user.role\_id

WHERE email LIKE "%fintra.co"

GROUP BY role\_user.user\_id

**SELECT\* FROM caso**

**WHERE created\_at BETWEEN "2019-02-08 14:00:00" AND "2019-02-08 14:30:00"**

**AND asignado = 5**

**Casos creados entre las 2:00 p.m. y las 2:30 p.m. y asignados al usuario 5**

**SELECT** **COUNT**(created\_at) **FROM** registro\_canal

**WHERE** created\_at **LIKE** "2019-03-27%" **AND** contenido = "Llamada Entrante"

**LIMIT** 1500

Conteo de llamadas ingresadas el día 27/03/2019

**SELECT** \* **FROM** registro\_canal

**WHERE** created\_at **LIKE** "2019-03-27%" **AND** contenido = "Llamada Entrante"

**LIMIT** 1500

Muestra las llamadas ingresadas el día 27/03/2019

SELECT campo1, count(\*)

FROM tabla1

GROUP BY campo1

HAVING count(\*) > 1

Conteo y busqueda de duplicados

SELECT \* FROM asterisk.cdr

WHERE LENGTH(src)=4 AND LENGTH(dst)>8

AND clid NOT LIKE '%<5380>'

AND disposition LIKE 'answer%'

AND calldate BETWEEN '2019-04-01 00:00:00' AND '2019-04-30 23:59:59'

AND lastapp LIKE 'Dial'

ORDER BY calldate DESC

Encontrar las llamadas salientes de la base de datos de un pbx basado en asterisk

SELECT cdr.clid as origen,

cdr.dst as destino,

CEIL(billsec/60) as minutos

FROM asterisk.cdr

WHERE cdr.calldate > '2019-04-01 00:00:00'

AND cdr.calldate < '2019-04-30 23:59:59'

AND cdr.src LIKE '53%' AND LENGTH(cdr.src) = 4

AND cdr.billsec > 0 AND cdr.dst != '2600'

AND LENGTH(dst)> 8 AND src != '5380'

AND disposition LIKE 'answer%'

Encontrar las llamadas salientes de la base de datos de un pbx basado en asterisk v.2.0

SELECT tmp.usuario\_id,

u.nombre,

u.email,

u.deleted\_at,

u.habilitado,

GROUP\_CONCAT(DISTINCT r.name),

GROUP\_CONCAT(DISTINCT c.nombre),

GROUP\_CONCAT(DISTINCT tc.nombre)

FROM

(

SELECT aea.usuario\_id, cc.campana\_id, cc.canal\_id

FROM agente\_estado\_agente aea

INNER JOIN agente\_campana\_canal acc ON acc.usuario\_id = aea.usuario\_id

LEFT JOIN campana\_canal cc ON cc.id = acc.campana\_canal\_id

WHERE aea.created\_at BETWEEN '2019-04-01 00:00:00' AND '2019-04-30 23:59:59'

GROUP BY aea.usuario\_id, cc.campana\_id, cc.canal\_id

UNION

SELECT cs.usuario\_id, cc.campana\_id, cc.canal\_id

FROM campana\_supervisor cs

INNER JOIN campana\_canal cc ON cc.campana\_id = cs.campana\_id

WHERE ((cs.created\_at <= '2019-04-30 23:59:59') AND

(cs.deleted\_at IS NULL OR cs.deleted\_at BETWEEN '2019-04-01 00:00:00' AND '2019-04-30 23:59:59'))

UNION

SELECT u2.id, null, null

FROM usuario u2

LEFT JOIN role\_user ru2 on u2.id = ru2.user\_id

WHERE ru2.user\_id = u2.id AND ru2.role\_id = 1

) tmp

LEFT JOIN usuario u ON u.id = tmp.usuario\_id

LEFT JOIN role\_user ru ON ru.user\_id = u.id

LEFT JOIN roles r ON r.id = ru.role\_id

LEFT JOIN campana c ON c.id = tmp.campana\_id

LEFT JOIN canal cn ON cn.id = tmp.canal\_id

LEFT JOIN tipo\_canal tc on cn.tipo\_canal\_id = tc.id

GROUP BY tmp.usuario\_id

Lista de usuarios de plataforma XXXX para CC

SELECT c.id,c.cliente\_id,primer\_nombre,primer\_apellido,descripcion,c.created\_at,COUNT(\*) FROM caso c

JOIN cliente cl ON c.cliente\_id = cl.id

WHERE c.created\_at >'2019-06-01%'

AND descripcion NOT LIKE ''

AND campana\_id = 10

GROUP BY cl.id, descripcion

HAVING COUNT(\*)>1

ORDER BY created\_at DESC

select c.id nro\_caso,u.nombre as asesor, *coalesce*(*json\_unquote*(*json\_extract*(dato\_gestion,'$.gestión\_contacto')),'') datos,  
 g.created\_at f\_gestion,  
 c.created\_at f\_lead  
from caso c  
FORCE INDEX (caso\_created\_at\_idx)  
join gestion g on g.caso\_id = c.id  
join usuario u on u.id=g.usuario\_id  
where c.campana\_id in (2,14)  
and c.created\_at between '2020-01-31 17:00:00' and '2020-02-29 17:00:00'  
and g.created\_at between '2020-01-31 17:00:00' and '2020-02-29 17:00:00'

Use of Coalesce primer valor no nulo