

```

SELECT * FROM public."RM20101" WHERE -- "CUSTNMBR" = 'EBS122656' -- "DOCNUMBR" =
'INV1217966'
select login,* from res_users where login ilike '%%';
select a.* from
(
select state,partner_id,count(partner_id) as ct from ebs_customer_gas_account group by
partner_id,state
)a where ct>1

select a.* from
(
select state,NEW_EBS_NO,count(partner_id) as ct from ebs_customer_gas_account group by
NEW_EBS_NO,state
)a where ct>1

select new_ebs_no from res_partner group by new_ebs_no having count(*) >1

select login from res_users where login in (
select email from res_partner group by new_ebs_no,name,email having count(*) >2
)

select new_ebs_no from ebs_customer_gas_account group by new_ebs_no ,state having count(*) >1
and state='activated'

select new_ebs_no,state,NAME from ebs_customer_gas_account group by new_ebs_no,state,NAME
having count() >1 and state='activated'
AND NEW_EBS_NO IN (
SELECT NEW_EBS_NO FROM RES_PARTNER group by new_ebs_no having count() >1 )

insert or update on table "ebs_payment_recon" violates foreign key constraint
"ebs_payment_recon_invoice_group_id_fkey"
DETAIL: Key (invoice_group_id)=(126118) is not present in table "invoice_gas_account_group". ---  

update ebs_payment_recon set invoice_group_id = NULL where id in (SELECT  

a.id  

FROM  

ebs_payment_recon as a  

LEFT JOIN invoice_gas_account_group as b  

ON b.id = a.invoice_group_id  

WHERE  

invoice_group_id IS NOT NULL  

AND b.id IS NULL)

```

```

update gas_connection_request set download_attached_passport = NULL where id in (SELECT
a.id
FROM
gas_connection_request as a
LEFT JOIN ir_attachment as b
ON b.id = a.download_attached_passport
WHERE
download_attached_passport IS NOT NULL
AND b.id IS NULL)

```

```

select a.* from
(
select NAME,count(partner_id) as ct from ebs_customer_gas_account
group by NAME
)a where ct>1

```

DUPLICATE DOCUMENT VERIFIED IS GETTING CREAED DELETED THAT

```

#
get_status_id=self.env['gas.track.status'].sudo().search([('status_def',
'in', ('doc_verified',''))])
#
upload_pending_doc_id=self.env['gas.track.status'].sudo().search([('status_d
ef', 'in', ('upload_pending_doc',''))])
    # # if get_status_id or upload_pending_doc_id:
    # for status in get_status_id:
    #     delete_doc_verified_duplicate =
self.env['tracking.status.details'].sudo().search([('tracking_gcr_ids', '=', 
self.gas_connection_req.id),('tracking_status', '=', status.id),
('status_active', '=', True)], limit=2,order='tracking_time desc')
        # if len(delete_doc_verified_duplicate) > 1:
        #     print('DUPLICATE DOCUMENT VERIFIED IS GETTING CREAED
DELETED THATttttttttttttttttttttttttttttttttttttttttt')
            # delete_doc_verified_duplicate =
self.env['tracking.status.details'].sudo().search([('tracking_gcr_ids', '=', 
self.gas_connection_req.id),('tracking_status', '=', status.id)], limit=1,
order='tracking_time desc')
                # dstr = """delete from tracking_status_details where
id=""" + str(delete_doc_verified_duplicate.id) + """
                # self.env.cr.execute(dstr)
                # for status in upload_pending_doc_id:
                #     delete_upload_pending_doc_duplicate =

```

inner join

if the column matches only it will return else it will not return the values

```
update ebs_customer_gas_account_status a set company_id = b.company_id
```

from

(select id,company_id from ebs_customer_gas_account)b where b.id=a.gas_id

```
refund_batch_reference= concat (refund_batch_reference , """" + str(customer_refund_approval.name) + """")
```

SELECT

t1.reference ,

t1.reading_date,

t1.entry_date

t1.source,

b.name as Customer,

c.name as Gas_Account,

t1.meter_reading,

d.meter_number

FROM

PORtal_Reading as t1

LEFT JOIN res_partner as b

ON t1.customer_id = b.id

LEFT JOIN ebs_customer_gas_account as c

ON t1.gas_account = c.id

LEFT JOIN ebs_customer_gas_meter_details as d

ON t1.new_meter_no = d.id

INNER JOIN

```
(SELECT MAX(id) as id,gas_account, customer_id FROM PORTAL_READING GROUP BY  
gas_account, customer_id) last_updates  
ON last_updates.id = t1.id and last_updates.gas_account = t1.gas_account  
and last_updates.customer_id = t1.customer_id  
and t1.entry_date between '2023-01-01' AND '2023-03-31'
```

```
update ebs_customer_gas_account set  
beneficiary_name=b.name from (select name,id from res_partner limit 100)b  
where partner_id=b.id and company_id=78
```

```
UPDATE ebs_customer_gas_account_status SET refund_mode_change =null  
customer_refund_id =  
null,allow_to_refund=null,refund_state='invoice_generated',iban_mismatch=False,bank_name_mismat  
ch=False  
,refund_batch_reference=",reason_to_allow_refund=",reference=null
```

```
DELETE FROM customer_refund_approval  
DELETE FROM ebs_customer_other_refund
```

```
DELETE FROM cashier_view  
DELETE FROM refund_debit_details
```

```
update ebs_customer_gas_account set bank_account_name=163  
,bank_account_no=11111,bank_iban_no=1111111,beneficiary_name='dddddd' ,swift_code=1 ,  
preferred_email='fffff'
```

```
update ebs_customer_gas_account_status set date_completed='2023-06-09' where id=118253
```

- Use the **INITCAP** function to replace the first letter of every word in a character string with an uppercase letter.
-