select

distinct m.mem\_id,

m.member\_type\_id,

m.lname,

m.fname,

m.email,

m.addr,

m.zip,

m.join\_date,

p.product\_desc,

b.product\_name,

c.cln\_name,

m.member\_end\_date,

case

mc2.member\_cancel\_type\_id when 2 then mc2.member\_cancel\_type\_id

else mc1.member\_cancel\_type\_id

end as member\_cancel\_type\_id,

m.active\_fl,

case

when ca.mem\_id is null then null

else 66

end as purchase\_detail\_type\_num,

case

when gmc.mem\_id is not null then 1

else 0

end as member\_declassified\_fl,

case

when ccpa.mem\_id is not null then 1

else 0

end as ccpa\_member\_anonymized\_fl,

gmc.declassification\_start\_date,

gmc.anonymization\_date,

gmc.request\_source

from

wldbowner.MEMBER m

inner join wlcommon.product p on

m.product\_id = p.product\_id

inner join wldbowner.brand b on

m.brn\_id = b.brn\_id

inner join dbowner.campaigns on

cam\_campaign\_id = sol\_id

inner join dbowner.websites on

cam\_website\_id = wbs\_website\_id

inner join dbowner.clients c on

cln\_client\_id = wbs\_client\_id

left outer join wlmember.iot\_2009\_ca\_member ca on

ca.mem\_id = m.mem\_id

left outer join wldbowner.member\_cancel mc1 on

(m.mem\_id = mc1.mem\_id

and mc1.member\_cancel\_type\_id = 1)

left outer join wldbowner.member\_cancel mc2 on

(m.mem\_id = mc2.mem\_id

and mc2.member\_cancel\_type\_id = 2)

left outer join WLMEMBER.GDPR\_MEMBER\_CLASSIFICATION gmc on

(m.mem\_id = gmc.mem\_id

and gmc.current\_fl = 1

and (gmc.declassification\_start\_date is not null

and date\_trunc('day', gmc.declassification\_start\_date) <= TO\_DATE('9/17/2020', 'MM/DD/YYYY'))

and (gmc.declassification\_end\_date is null

or date\_trunc('day', gmc.declassification\_end\_date) > TO\_DATE('9/17/2020', 'MM/DD/YYYY')))

left outer join WLMEMBER.CCPA\_MEMBER\_CLASSIFICATION ccpa on

(m.mem\_id = ccpa.mem\_id

and ccpa.current\_fl = 1

and (ccpa.anonymization\_date is not null

and date\_trunc('day', ccpa.anonymization\_date) <= TO\_DATE('9/17/2020', 'MM/DD/YYYY')))

inner join wldbowner.solicitation s on

s.sol\_id = m.sol\_id

inner join wlsolicitation.solicitation\_template st on

st.sol\_template\_id = s.sol\_template\_id

where

1 = 1

and LOWER(unaccent\_string(lname)) like LOWER(unaccent\_string(:lname)) || '%'

and LOWER(unaccent\_string(fname)) like LOWER(unaccent\_string(:fname)) || '%'

and st.country\_id in (1, 2, 3, 121, 121, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14)

limit 102

Limit (cost=11646465.55..11646470.90 rows=102 width=209) (actual time=74352.929..74352.929 rows=0 loops=1)

-> Unique (cost=11646465.55..11646489.17 rows=450 width=209) (actual time=74352.928..74352.928 rows=0 loops=1)

-> Sort (cost=11646465.55..11646466.67 rows=450 width=209) (actual time=74352.927..74352.927 rows=0 loops=1)

Sort Key: m.mem\_id, m.member\_type\_id, m.lname, m.fname, m.email, m.addr, m.zip, m.join\_date, p.product\_desc, b.product\_name, c.cln\_name, m.member\_end\_date, (CASE mc2.member\_cancel\_type\_id WHEN '2'::numeric THEN mc2.member\_cancel\_type\_id ELSE mc1.member\_cancel\_type\_id END), m.active\_fl, (CASE WHEN (ca.mem\_id IS NULL) THEN NULL::integer ELSE 66 END), (CASE WHEN (gmc.mem\_id IS NOT NULL) THEN 1 ELSE 0 END), (CASE WHEN (ccpa.mem\_id IS NOT NULL) THEN 1 ELSE 0 END), gmc.declassification\_start\_date, gmc.anonymization\_date, gmc.request\_source

Sort Method: quicksort Memory: 25kB

-> Nested Loop Left Join (cost=1.46..11646445.72 rows=450 width=209) (actual time=74352.907..74352.907 rows=0 loops=1)

Join Filter: (m.mem\_id = ccpa.mem\_id)

-> Nested Loop Left Join (cost=1.46..11646427.04 rows=450 width=213) (actual time=74352.907..74352.907 rows=0 loops=1)

Join Filter: (m.mem\_id = gmc.mem\_id)

-> Nested Loop Left Join (cost=1.46..11645396.07 rows=450 width=167) (actual time=74352.906..74352.906 rows=0 loops=1)

-> Nested Loop (cost=1.02..11641589.19 rows=450 width=162) (actual time=74352.906..74352.906 rows=0 loops=1)

Join Filter: (s.sol\_template\_id = st.sol\_template\_id)

-> Nested Loop (cost=1.02..11603617.47 rows=515 width=167) (actual time=74352.906..74352.906 rows=0 loops=1)

-> Nested Loop Left Join (cost=0.73..11603376.03 rows=515 width=174) (actual time=74352.905..74352.905 rows=0 loops=1)

-> Nested Loop Left Join (cost=0.29..11599019.27 rows=515 width=169) (actual time=74352.905..74352.905 rows=0 loops=1)

Join Filter: (ca.mem\_id = m.mem\_id)

-> Nested Loop (cost=0.29..11585313.89 rows=515 width=151) (actual time=74352.905..74352.905 rows=0 loops=1)

Join Filter: (websites.wbs\_client\_id = c.cln\_client\_id)

-> Seq Scan on clients c (cost=0.00..46.75 rows=2375 width=21) (actual time=0.007..0.369 rows=2344 loops=1)

-> Materialize (cost=0.29..11566921.55 rows=515 width=142) (actual time=31.720..31.720 rows=0 loops=2344)

-> Nested Loop (cost=0.29..11566918.98 rows=515 width=142) (actual time=74352.049..74352.049 rows=0 loops=1)

Join Filter: (campaigns.cam\_website\_id = websites.wbs\_website\_id)

-> Seq Scan on websites (cost=0.00..97.68 rows=3768 width=12) (actual time=0.003..0.658 rows=3722 loops=1)

-> Materialize (cost=0.29..11537714.79 rows=515 width=142) (actual time=19.976..19.976 rows=0 loops=3722)

-> Nested Loop (cost=0.29..11537712.21 rows=515 width=142) (actual time=74350.594..74350.594 rows=0 loops=1)

-> Nested Loop (cost=0.00..11534399.41 rows=515 width=130) (actual time=74350.594..74350.594 rows=0 loops=1)

Join Filter: (m.brn\_id = b.brn\_id)

-> Nested Loop (cost=0.00..11532815.44 rows=515 width=116) (actual time=74350.594..74350.594 rows=0 loops=1)

Join Filter: (m.product\_id = p.product\_id)

-> Seq Scan on member m (cost=0.00..11531931.36 rows=515 width=100) (actual time=74350.593..74350.593 rows=0 loops=1)

Filter: ((lower(unaccent\_string((lname)::text)) ~~ 'tugav%'::text) AND (lower(unaccent\_string((fname)::text)) ~~ 'miamo%'::text))

Rows Removed by Filter: 20595367

-> Materialize (cost=0.00..3.71 rows=114 width=27) (never executed)

-> Seq Scan on product p (cost=0.00..3.14 rows=114 width=27) (never executed)

-> Materialize (cost=0.00..24.03 rows=202 width=26) (never executed)

-> Seq Scan on brand b (cost=0.00..23.02 rows=202 width=26) (never executed)

-> Index Scan using pk\_campaigns on campaigns (cost=0.29..6.42 rows=1 width=12) (never executed)

Index Cond: (cam\_campaign\_id = m.sol\_id)

-> Materialize (cost=0.00..36.55 rows=1770 width=18) (never executed)

-> Seq Scan on iot\_2009\_ca\_member ca (cost=0.00..27.70 rows=1770 width=18) (never executed)

-> Index Scan using pk\_member\_cancel on member\_cancel mc1 (cost=0.44..8.45 rows=1 width=11) (never executed)

Index Cond: (m.mem\_id = mem\_id)

Filter: (member\_cancel\_type\_id = '1'::numeric)

-> Index Scan using pk\_solicitation on solicitation s (cost=0.29..0.46 rows=1 width=11) (never executed)

Index Cond: (sol\_id = campaigns.cam\_campaign\_id)

-> Materialize (cost=0.00..285.92 rows=4880 width=5) (never executed)

-> Seq Scan on solicitation\_template st (cost=0.00..261.52 rows=4880 width=5) (never executed)

Filter: (country\_id = ANY ('{1,2,3,121,121,4,5,6,7,8,9,10,11,12,13,14}'::numeric[]))

-> Index Scan using pk\_member\_cancel on member\_cancel mc2 (cost=0.44..8.45 rows=1 width=11) (never executed)

Index Cond: (m.mem\_id = mem\_id)

Filter: (member\_cancel\_type\_id = '2'::numeric)

-> Materialize (cost=0.00..25.59 rows=149 width=46) (never executed)

-> Seq Scan on gdpr\_member\_classification gmc (cost=0.00..24.85 rows=149 width=46) (never executed)

Filter: ((declassification\_start\_date IS NOT NULL) AND (current\_fl = 1) AND (date\_trunc('day'::text, declassification\_start\_date) <= to\_date('9/17/2020'::text, 'MM/DD/YYYY'::text)) AND ((declassification\_end\_date IS NULL) OR (date\_trunc('day'::text, declassification\_end\_date) > to\_date('9/17/2020'::text, 'MM/DD/YYYY'::text))))

-> Materialize (cost=0.00..10.81 rows=1 width=18) (never executed)

-> Seq Scan on ccpa\_member\_classification ccpa (cost=0.00..10.80 rows=1 width=18) (never executed)

Filter: ((anonymization\_date IS NOT NULL) AND (current\_fl = 1) AND (date\_trunc('day'::text, anonymization\_date) <= to\_date('9/17/2020'::text, 'MM/DD/YYYY'::text)))

Planning time: 8.077 ms

Execution time: 74353.230 ms

From, above explain plan it’s been observed that application is waiting for more than minute to get response from database because of the slow running query which might be the cause of the application exception.