

PostGress Query

Thursday, March 2, 2023 6:42 PM

```
-- Check Table size
select pg_size_pretty (pg_relation_size('iq4.bus_sap_production'));
-- Check Database size
SELECT pg_database.datname,
pg_size_pretty(pg_database_size(pg_database.datname)) AS size
FROM pg_database;

select * from pg_indexes where schemaname in ('iq4') and tablename in ('released_checkpoint')

select now()
select * from pg_timezone_names

select * from information_schema.columns where column_name like '%operator%'

----- Creation of user
delete from pg_user where username = 'test'
create user shreya with password 'test123'
grant select on all tables in schema iq4 to shreya

GRANT USAGE ON SCHEMA iq4 to shreya
ALTER DEFAULT PRIVILEGES IN SCHEMA iq4 GRANT SELECT ON TABLES TO shreya;
GRANT SELECT ON ALL SEQUENCES IN SCHEMA iq4 TO shreya;
GRANT SELECT ON ALL TABLES IN SCHEMA iq4 TO shreya;

SELECT * FROM pg_extension WHERE extname = 'dblink'

-- check the active connection made to DB
SELECT * FROM pg_stat_activity order by datname desc
where datname in ('aggregates-scanning-dev')

--Below code to cast date as text to use like operator
and c.bb_number in ('BB4105401010') and CAST(a.modified_time::TEXT as character) ilike '%%'

-- Below query has logic for converting the date as text also how to use array so that multiple bbnumber and %
select a.serial_number_id,a.bus_id,a.serial_number,a.serial_number_with_agbm,a.part_number,a.part_number_id,
a.transaction_code,a.position,a.creation_time,a.modified_time,a.error_message,a.message_type,a.upload_success,
a.sap_upload_status,a.demo_mode,a.part_number_original,a.serial_number_original,a.agbm,b.vpd_ident,c.bb_number,
a.username,a.end_time,null as error_log_id
FROM serial_number_details as a
INNER JOIN part_number_vpd_ident_details as b on a.part_number_id=b.part_number_id
INNER JOIN bus_details as c on a.bus_id=c.bus_id
where a.message_type like '%' and a.demo_mode = false and a.end_time is null
and c.bb_number ILIKE ANY (ARRAY['BB4105401010', 'BB4105401020']) and ((CAST(a.modified_time as DATE))::TEXT) ilike CAST('%' as TEXT)
and a.serial_number_with_agbm ilike '%'
UNION
select distinct a.serial_number_id,a.bus_id,a.serial_number,a.serial_number_with_agbm,a.part_number,a.part_number_id,
a.transaction_code,a.position,a.creation_time,a.modified_time,a.error_message,a.message_type,a.upload_success,
a.sap_upload_status,a.demo_mode,a.part_number_original,a.serial_number_original,null,null,c.bb_number,
a.username,a.end_time,err.error_log_id
FROM serial_number_details as a
INNER JOIN error_log_details as err on a.serial_number_id=err.serial_number_id and err.end_time is null
INNER JOIN bus_details as c on a.bus_id=c.bus_id
where a.message_type like '%' and a.demo_mode = false and a.end_time is null
and c.bb_number ILIKE ANY (ARRAY['BB4105401010', 'BB4105401020']) and ((CAST(a.modified_time as DATE))::TEXT) ilike CAST('%' as TEXT)
and a.serial_number_with_agbm ilike '%'
order by serial_number_id desc

-----
update iq4.released_checkpoint set visible =false,
ok_by='IQ4SCRUB',
modified_user = 'IQ4SCRUB',
checkpoint_status = 'OK',
modified_time = NOW(),

where released_checkpoint_id in (select released_checkpoint_id from iq4.released_checkpoint where bb_number in ( 'BB4100100973')
and pda_id in ( 12993, 11475, 11425, 11529, 11530, 11554, 11422, 11423, 11381, 11555, 11424, 11474) order by bb_number) RETURNING *

select bb_number, operator_workplace,checkpoint_status,modified_user,visible,ok_by,hand_type,pda_id
from iq4.released_checkpoint where bb_number in ('BB4102010054')
and pda_id in (19263) order by bb_number
```

```
select bb_number, hierarchy_code, plant_id from iq4.released_checkpoint where modified_time > (date_trunc('hour', NOW())::timestamp) - INTERVAL '1 hour')
```

```
group by bb_number, hierarchy_code, plant_id
```

---Update Multiple column using case

```
update iq4.master_workplace_group set group_name=
```

```
CASE
```

```
WHEN group_name='88_1 Agrégats Ext + Batterie' THEN '88_1 Agrégats Ext et Batterie'
```

```
WHEN group_name='88_7 + PE E-FACH Haut + Bas' THEN '88_7 E-FACH Haut et Bas'
```

```
WHEN group_name= '88_8 distribution_Avt + Sat' THEN '88_8 Distribution Avt. et Sat.'
```

```
WHEN group_name= '76_Girouettes Lat. & ARR' THEN '76_Girouettes Lat. et Arr.'
```

```
WHEN group_name='87_Prep. PDB & Module' THEN '87_Prep. PDB et Module'
```

```
WHEN group_name= '80_SAE & Ethylostest' THEN '80_SAE et Ethylostest'
```

```
WHEN group_name = '80_Vidéo & Comptage' THEN '80_Vidéo et Comptage'
```

```
END
```

```
where group_name in ('88_6 + V NT-FACH + VDV','88_7 + PE E-FACH Haut + Bas','88_8 distribution_Avt + Sat','76_Girouettes Lat. & ARR','87_Prep. PDB & Module','80_SAE & Ethylostest','80_Vidéo & Comptage','88_1 Agrégats Ext + Batterie')
```

```
returning *
```

=====

```
with table_1 as(select bb_number,pda_id,count(*) from iq4.released_checkpoint
```

```
where checkpoint_type ='qualitätsnachweis-mit-ds-merkmal'
```

```
--and checkpoint_status in ('OPEN')
```

```
--and hand_type != 'HAND1'
```

```
group by pda_id, bb_number
```

```
having count(*) =1
```

```
EXCEPT
```

```
select bb_number,pda_id,count(*) from iq4.released_checkpoint
```

```
where checkpoint_type ='qualitätsnachweis-mit-ds-merkmal'
```

```
--and checkpoint_status in ('OPEN')
```

```
and hand_type = 'HAND1'
```

```
group by pda_id, bb_number
```

```
having count(*) =1)
```

```
select b.bb_number, a.operator_workplace,a.checkpoint_status,a.modified_user,a.visible,a.ok_by,a.hand_type,a.pda_id,a.creation_time,
```

```
a.modified_time,a.checkpoint_type
```

```
from iq4.released_checkpoint a, table_1 b
```

```
where a.pda_id = b.pda_id
```

```
and a.bb_number = b.bb_number
```

```
and checkpoint_status in ('OPEN')
```

Creation of partition in Postgresql

```
CREATE TABLE public.partition_2 PARTITION OF iq4.released_checkpoint_partition
```

```
FOR VALUES FROM ('BB4107100340') TO ('BB4107100341')
```

```
TABLESPACE pg_default;
```

If no table space is assigned then pg_default table space is will be assigned

Indexing is a data structure that you build and assigned on existing table , Tried to analyze and build the shortcut

Indexes are smaller referential tables that holds your reference against the indexed value

Basically it is 2 column table one is valued where we created the index on other one is id of index

Index columns are sorted

Btree and 7

Explain analyze select * from table

Once u do that it will display Index only if query is effectively indexes

If Heap Fetches is 0 that means it did not go through heap

It also display planning time and execution time

When u query the table with where condition where index does not exists then it will do full table scan , So u can see in below screen that parallel sequence scan on employees

U can also see workers planned is 2 that means postgres parallelly started executing 2 jobs

If you using index will not be used

Database Indexing Explained (with PostgreSQL)

QUERY P

LAN

```
Gather (cost=1000.00..217539.56 rows=114 width=4) (actual time=
72.006..3026.990 rows=563 loops=1)
```

Workers Planned: 2

Workers Launched: 2

```
-> Parallel Seq Scan on employees (cost=0.00..216528.16 rows
=48 width=4) (actual time=30.987..2989.872 rows=188 loops=3)
```

```
Filter: ((name)::text = 'Zs'::text)
```

Rows Removed by Filter: 7000147

Planning Time: 0.083 ms

JIT:

Functions: 12

```
Options: Inlining false, Optimization false, Expressions true,
Deforming true
```

2.278 ms, Emission 58.535 ms, Total 73.312 ms