Last edited by **Isabel F Freitas** 9 months ago

Useful Commands

Find the size of tables in a specific database in postgresql

```
SELECT
    relname AS "relation",
    pg_size_pretty (
        pg_total_relation_size (C .oid)
    ) AS "total_size"
FROM
    pg_class C
LEFT JOIN pg_namespace N ON (N.oid = C .relnamespace)
WHERE
    nspname NOT IN (
        'pg_catalog',
        'information_schema'
    )
AND C .relkind <> 'i'
AND nspname !~ '^pg_toast'
ORDER BY
    pg_total_relation_size (C .oid) DESC;
```

Copy table from one database to another

```
insert into public.measure_definition_23
select *
from dblink('dbname=alp host=alp4.demarq.uk password=password user=isabel port=54320',
            'select * from dbt_isabel_marts.measure_definition')
       as t1(measure_id text,measure_label text, v_from_ts timestamp without time zone, v_to_ts time
```

Check for active background tasks in postgresql and kill them

- Connect to postgresql from the command line using:

```
psql -U isabel -h alp4.demarq.uk -p 54320 alp
```

- Check active background tasks:

```
SELECT * FROM pg_stat_activity WHERE state = 'active';
```

- Exit it: `\q`
- cancel all requests except itself use:

```
SELECT pg_cancel_backend(pid) FROM pg_stat_activity WHERE state = 'active' and pid <> pg_backend_pi
```

- Check for dead tuples:

```
SELECT
        relname AS TableName
        ,n_live_tup AS LiveTuples
        ,n_dead_tup AS DeadTuples
FROM pg_stat_user_tables;
```

or

```
SELECT
        relname AS ObjectName
        ,pg_stat_get_live_tuples(c.oid) AS LiveTuples
        ,pg_stat_get_dead_tuples(c.oid) AS DeadTuples
FROM pg_class c;
```

or

```
SELECT relname, n_dead_tup FROM pg_stat_user_tables;
```

Kill celery worker

- Check for active workers:

```
ps aux|grep 'celery worker'
```

- Kill all celery workers:

```
ps auxww | grep 'celery worker' | awk '{print $2}' | xargs kill -9
```

Grant Permissions on all tables in a specific schema

```
GRANT SELECT ON ALL TABLES IN SCHEMA dbt_isabel_marts TO alp_developer
GRANT SELECT ON ALL TABLES IN SCHEMA dbt_isabel_data TO alp_developer
```

- https://tableplus.com/blog/2018/04/postgresql-how-to-grant-access-to-users.html

Git error sorting

- Error: `master and origin/master branch have diverged`
- Sorting it out:

```
git reset --hard origin/master
```

- https://stackoverflow.com/questions/2452226/master-branch-and-origin-master-have-diverged-how-to-undiverge-branches/2452610

Create SAS Views

```
DROP VIEW if exists alpraw_vw.acc;
DROP VIEW if exists alpraw_vw.acc1;
DROP VIEW if exists alpraw_vw.ar_st;
DROP VIEW if exists alpraw_vw.ar_tx;
DROP VIEW if exists alpraw_vw.gl;

CREATE OR REPLACE VIEW alpraw_vw.acc as select * from alp.dbt_isabel_alp_target.stg_acc;
-- CREATE OR REPLACE VIEW alpraw_vw.acc1 as select * from alp.dbt_isabel_alp_target.stg_acc;
CREATE OR REPLACE VIEW alpraw_vw.ar_st as select * from alp.dbt_isabel_alp_target.stg_ar_st;
CREATE OR REPLACE VIEW alpraw_vw.ar_tx as select * from alp.dbt_isabel_alp_target.stg_ar_tx;
CREATE OR REPLACE VIEW alpraw_vw.gl as select * from alp.dbt_isabel_alp_target.stg_gl;
-- CREATE OR REPLACE VIEW views.acc as select * from alp.dbt_isabel_alp_target.stg_acc;
-- CREATE OR REPLACE VIEW dbt_isabel_alp_target.acc as select * from alp.dbt_isabel_alp_target.stg_a
-- GRANT SELECT ON dbt_isabel_alp_target.* TO "alp_dev_ro" ;
GRANT SELECT ON ALL TABLES IN SCHEMA alpraw_vw TO alp_dev_ro;
GRANT SELECT ON ALL TABLES IN SCHEMA alpraw_vw TO sasro;
GRANT SELECT ON ALL TABLES IN SCHEMA dbt_isabel_alp_target TO alp_dev_ro;
GRANT USAGE on SCHEMA dbt_isabel_alp_target TO sasro;
-- GRANT SELECT ON ALL TABLES IN SCHEMA dbt_isabel_alp_target TO sasro;
-- GRANT SELECT ON TABLE dbt_isabel_alp_target.stg_acc TO "alp_dev_ro";
```

Useful Scripts

```
# ssh into apl6
ssh isabel_freitas@alp5.demarq.uk

# Copying files from the local machine to the server through ssh
scp -rp "/mnt/c/Users/Isabel F Freitas/virtual_environments/etl/" "isabel_freitas@10.2.0.99:/opt/dat

scp -rp "/mnt/c/Users/Isabel F Freitas/alp/alp_isabel/final_code" "isabel_freitas@10.2.0.99:/opt/dat

scp -rp "/mnt/c/Users/Isabel F Freitas/alp/alp_isabel/final_code/*.py" "isabel_freitas@10.2.0.99:/op

scp -rp "/mnt/c/Users/Isabel F Freitas/alp/columns/Vanquis ALP Data Dictionary.csv" "isabel_freitas@
```

```
# Copying files from server to the local machine through ssh
sftp isabel_freitas@10.2.0.99:/opt/dataops/envs/isabel/vanquis_xlsx_reports/data_delivery_report.xls

# Copy scripts
cp /opt/dataops/data/raw/vanquis_bank/*.csv /opt/dataops/envs/isabel/vanquis_data/

# Working directories
cd /opt/dataops/envs/isabel/vanquis_data/

cd /opt/dataops/envs/isabel/vanquis_dictionary/

cd /opt/dataops/envs/isabel/vanquis_xlsx_reports/

# Activate virtual environment
source /opt/dataops/envs/isabel/virtual_envir/alp/bin/activate

# Moving to my directory
cd /opt/dataops/envs/isabel/final_code/


# Converting sas7bdat files to csv from the command line
# This command was run locally /opt/dataops/envs/isabel/sas_files
sas7bdat_to_csv i2sbStatementbasesegment.sas7bdat i2sbStatementbasesegment.csv

# Spark add package
pyspark --packages com.crealytics:spark-excel_2.12:0.13.1

# Check the Spark packages
ll /home/isabel_freitas/.ivy2/jars/

# postgresql shell command
psql -U isabel -h alp4.demarq.uk -p 54320 alp

# copy table from postgresql to local drive
\copy public.feb_trans to '/mnt/c/alp/feb_trans.csv' DELIMITER ',' CSV HEADER
\copy (select * from dbt_isabel_marts.measure_data where measure_id = 'A0000001' limit 10) to '/mnt/

# Back up the alp database
pg_dump -U isabel -h alp4.demarq.uk -p 54320 alp > /mnt/c/alp/alp.sql
```

pgAdmin4 Keyboard Shortcuts

- https://www.pgadmin.org/docs/pgadmin4/development/keyboard_shortcuts.html

List procedures

```
select n.nspname as schema_name,
       p.proname as specific_name,
       l.lanname as language,
       case when l.lanname = 'internal' then p.prosrc
            else pg_get_functiondef(p.oid)
            end as definition,
       pg_get_function_arguments(p.oid) as arguments
from pg_proc p
left join pg_namespace n on p.pronamespace = n.oid
left join pg_language l on p.prolang = l.oid
left join pg_type t on t.oid = p.prorettype
where n.nspname not in ('pg_catalog', 'information_schema')
      and p.prokind = 'p'
order by schema_name,
         specific_name;
```

- OR:

```
SELECT *
 FROM alp.INFORMATION_SCHEMA.ROUTINES
 WHERE ROUTINE_TYPE = 'PROCEDURE'
```

- https://dataedo.com/kb/query/postgresql/list-stored-procedures

Install yarn

- `curl -sL https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -`
- `echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee /etc/apt/sources.list.d/yarn.list`
- `sudo apt update && sudo apt install yarn`