Important Database Administrator Commands

In this post, I am sharing few important and basic commands for Database Administrator of PostgreSQL and PSQL Console.

Below are the important PSQL command:

# \l : to list all databases

# \? : to list help for all commands

# \h create table : show help with topics

# \d : to list all tables

# \df : to list all functions

# \da : to list all aggregate functions

# \dn : to list all schemas

# \du : to list all users

# \r : to reset the query buffer

# \db : to list all table space

To show current data directory.

```
SHOW data_directory;
```

To change current data directory and it requires a server restart.

```
SET data_directory TO NewPath;
```

To show current server configuration setting.

```
SHOW ALL;
```

```
SELECT name, setting, unit, context FROM pg_settings;
```

Get PostgreSQL Version name:

```
SELECT VERSION();
```

How to find size of Database and Table in PostgreSQL

In this post, I am sharing few important function for finding the size of database, table and index in PostgreSQL.

Script 1 : Find a total size of the database.

```
SELECT pg_size_pretty(pg_database_size('db_employee'));
```

Script 2: Find a total size of the table with an index.

```
SELECT pg_size_pretty(pg_total_relation_size('Employee_Table'));
```

Script 3: Find a total size of the table without an index.

```
SELECT pg_size_pretty(pg_relation_size('Employee_table'));
```

Script 4: Find a total size of the index.

```
SELECT pg_size_pretty(pg_indexes_size('index_empid'));
```

Script 5: Find a total number of rows in a table.

```
SELECT COUNT(1) FROM Employee_table;
```

Script 6 : Find all the table and index size in the current database.

```
SELECT
    TableName
    ,pg_size_pretty(pg_table_size(TableName)) AS TableSize
    ,pg_size_pretty(pg_indexes_size(TableName)) AS IndexSize
    ,pg_size_pretty(pg_total_relation_size(TableName)) AS TotalSize
FROM
(
     SELECT ('"' || table_schema || '"."' || table_name || '"') AS TableName
     FROM information_schema.tables
) AS Tables
ORDER BY 4 DESC
```

Copy Database to another Server in Windows (pg_dump – backup & restore)

In our organization, junior database developers are learning PostgreSQL, and they require a sample database from our Validation Database Server.

In this scenario, Database Administrator takes PostgreSQL backup using pg_dump utility and restore that dump into the development environment.

A pg_dump is a PostgreSQL utility which is used for PostgreSQL database backup and restore.

Another option is, directly copy the database from one server to another server using pg_dump + psql. In this scenario, Database Administrator does not require any intermediate database backup file.

Copy database, Using pg_dump utility for backup and restore:

Take full database backup:

```
pg_dump -U username database_name > filepath\backup.sql
```

Restore database backup:

```
pg_dump -U username database_name < filepath\backup.sql
```

Script to find the count of objects for each Database Schema

In this post, I am sharing a script to find the total count of objects for each database schemas of PostgreSQL.
If you Postgres DBA and want to do accounting on a number of total objects per schemas, you can use this script.

Using this script, you can find different types of objects count like table, view, index, sequence

```sql
SELECT
       n.nspname as schema_name
       ,CASE c.relkind
          WHEN 'r' THEN 'table'
          WHEN 'v' THEN 'view'
          WHEN 'i' THEN 'index'
          WHEN 'S' THEN 'sequence'
          WHEN 's' THEN 'special'
       END as object_type
       ,count(1) as object_count
FROM pg_catalog.pg_class c
LEFT JOIN pg_catalog.pg_namespace n ON n.oid = c.relnamespace
WHERE c.relkind IN ('r','v','i','S','s')
GROUP BY  n.nspname,
       CASE c.relkind
          WHEN 'r' THEN 'table'
          WHEN 'v' THEN 'view'
          WHEN 'i' THEN 'index'
          WHEN 'S' THEN 'sequence'
          WHEN 's' THEN 'special'
       END
ORDER BY n.nspname,
       CASE c.relkind
          WHEN 'r' THEN 'table'
          WHEN 'v' THEN 'view'
          WHEN 'i' THEN 'index'
          WHEN 'S' THEN 'sequence'
          WHEN 's' THEN 'special'
       END;
```