1. change current user password \password
2. password file ~/.pgpass check for file permission (600 idealy)

host:port:dbname:user:password

1. connection service file

First, create a file named pg\_service.conf with the following contents:

[dbservice1]host=postgres1port=5432dbname=postgres

You can then either copy it into place at /etc/pg\_service.conf or another agreed central location. You can then set the environment variable PGSYSCONFDIR to that directory location.

Now, you can then specify a connection string like the following:

service=dbservice1 user=sriggs

The service can also be set using an environment variable named PGSERVICE.

4) server uptime:

SELECT pg\_postmaster\_start\_time();

SELECT date\_trunc('second',current\_timestamp - pg\_postmaster\_start\_time()) as uptime;

5) data directory

| **Subdirectory** | **Purpose** |
| --- | --- |
| base | Main data directory. Beneath this directory each database has its own directory within which are the files for each database table or index. |
| global | Database server catalog tables that are shared across all databases. |
| pg\_clog | Transaction status files. |
| pg\_multixact | Row-level lock status files |
| pg\_subtrans | Subtransaction status files |
| pg\_tblspc | Links to external tablespaces |
| pg\_twophase | "2-phase commit", or Prepared transaction status |
| pg\_xlog | Transaction log (or Write Ahead Log - WAL) |

6). Server log

/var/log/postgresql or /var/lib/pgsql/data/pg\_log

sys log

/var/log/syslog

7) pg\_controldata is a PostgreSQL server application that shows the contents of a server's control file. The control file is located within the data directory of a server, and is created at database initialization time. Some of the information within it is updated regularly, or when certain major events occur.

8) \x change the output format as one column per line

9) show tables:

SELECT count(\*) FROM information\_schema.tablesWHERE table\_schema NOT IN ('information\_schema','pg\_catalog');

Or

\d

10) show disk space of a database

SELECT pg\_database\_size(current\_database());

SELECT sum(pg\_database\_size(datname)) from pg\_database; (all databases size)

11) table space

We can find out the size of a table using the following query:

select pg\_relation\_size('accounts');

We can also find out the total size of a table including indexes and other related space using the following query:

select pg\_total\_relation\_size('accounts');

or we can also use a psql command as follows:

\dt+ accounts

12) Top 10 Biggest Tables

SELECT table\_name,pg\_relation\_size(table\_name) as sizeFROM information\_schema.tablesWHERE table\_schema NOT IN ('information\_schema','pg\_catalog')ORDER BY size DESCLIMIT 10;

13) get table size

CREATE OR REPLACE FUNCTION estimated\_row\_count(text) RETURNS bigint LANGUAGE sql AS $$ SELECT (CASE WHEN reltuples > 0 THEN pg\_relation\_size($1)/(8192\*relpages/reltuples) ELSE 0 END)::bigint FROM pg\_class WHERE oid = $1::regclass; $$;

CREATE OR REPLACE FUNCTION pg\_relation\_size\_nolock(tablename regclass) RETURNS BIGINT LANGUAGE plpgsql AS $$ DECLARE classoutput RECORD; tsid INTEGER; rid INTEGER; dbid INTEGER; filepath TEXT; filename TEXT; datadir TEXT; i INTEGER := 0; tablesize BIGINT; BEGIN -- -- get data directory -- EXECUTE 'SHOW data\_directory' INTO datadir; -- -- get relfilenode and reltablespace -- SELECT reltablespace as tsid ,relfilenode as rid INTO classoutput FROM pg\_class WHERE oid = tablename AND relkind = 'r'; –- –- Throw an error if we can't find the tablename specified -- IF NOT FOUND THEN RAISE EXCEPTION 'tablename % not found', tablename; END IF; tsid := classoutput.tsid; rid := classoutput.rid; -- -- get the database object identifier (oid) -- SELECT oid INTO dbid FROM pg\_database WHERE datname = current\_database(); -- -- Use some internals knowledge to set the filepath –- IF tsid = 0 THEN filepath := datadir || '/base/' || dbid || '/' || rid; ELSE filepath := datadir || '/pg\_tblspc/' || tsid || '/' || dbid || '/' || rid; END IF; -- -- Look for the first file. Report if missing -- SELECT (pg\_stat\_file(filepath)).size INTO tablesize; -- -- Sum the sizes of additional files, if any -- WHILE FOUND LOOP i := i + 1; filename := filepath || '.' || i; -- -- pg\_stat\_file returns ERROR if it cannot see file -- so we must trap the error and exit loop -- BEGIN SELECT tablesize + (pg\_stat\_file(filename)).size INTO tablesize; EXCEPTION WHEN OTHERS THEN EXIT; END; END LOOP; RETURN tablesize; END; $$;

14) dependency:

if a table has views depend on it, it can not be dropped.

Check dependency constraints:

SELECT \* FROM pg\_constraint WHERE confrelid = 'orders'::regclass;

Or \dt+

15) check parameter settings:

SELECT name, setting, reset\_val, source FROM pg\_settings WHERE source = 'session';