Postgres

- PostgreSQL on macOS
- Backslash commands
- Cheatsheet
- Queries with IPs
- Dump / Restore
- Major Version Upgrade on Debian

PostgreSQL on macOS

Install PostgreSQL using brew

```
brew install postgresql
```

List services maintained by brew

```
$ brew services list
Name       Status  User Plist
postgresql started lr   ...
```

If you don't want it running as service you can manually start it with

```
postgres -D /usr/local/var/postgres
```

Backslash commands

	command
Toggle expanded display	`\x`
Toggle tuples only	`\t`
Connect to database	`\c db_name`
List databases	`\l`
Display table definition	`\d table_name`
Display triggers	`\dft`

Cheatsheet

Truncate/empty all tables

```
SELECT 'TRUNCATE TABLE ' ||  tablename || ' CASCADE;' FROM pg_tables WHERE
tableowner='acovix_devl';
```

Show process / query list

```
SELECT * FROM pg_stat_activity;

-- active queries
SELECT pid, age(query_start, clock_timestamp()), usename, query
    FROM pg_stat_activity
    WHERE query != '<IDLE>' AND query NOT ILIKE '%pg_stat_activity%'
    ORDER BY query_start desc;
```

Kill running query

```
SELECT pg_cancel_backend(procpid);

-- kill idle query
SELECT pg_terminate_backend(procpid);
```

List databases size

```
select datname, pg_size_pretty(pg_database_size(datname)) from pg_database order by
pg_database_size(datname) desc;
```

Show all triggers from a schema

```
SELECT
    event_object_table,
    trigger_name,
```

```
    event_manipulation,
    action_statement,
    action_timing
FROM information_schema.triggers ORDER BY event_object_table, event_manipulation;
```

Rename database

```
alter database old_name rename to new_name;
```

Set database owner

```
alter database sample_name owner to david;
```

Rename role

```
alter role role_name rename to new_role_name;
```

- set password, because after rename password is cleared

List enum

```
select n.nspname as enum_schema,
    t.typname as enum_name,
    string_agg(e.enumlabel, ', ') as enum_value
from pg_type t
    join pg_enum e on t.oid = e.enumtypid
    join pg_catalog.pg_namespace n ON n.oid = t.typnamespace
group by enum_schema, enum_name
```

Queries with IPs

Top networks

```
CREATE TABLE ip (
  id serial NOT NULL,
  addr cidr NOT NULL,
  hostname character varying(25) NOT NULL
);

SELECT * FROM ip WHERE NOT EXISTS
    (SELECT addr FROM net n WHERE n.addr >> net.addr);
```

Bigint to IP

```
create function bigint_to_inet(bigint) returns inet as
$$
select (($1>>24&255)||'.'||($1>>16&255)||'.'||($1>>8&255)||'.'||($1>>0&255))::inet
$$ language sql;
```

Dump / Restore

Dump All Functions (to a file)

```
SELECT pg_get_functiondef(f.oid) FROM pg_catalog.pg_proc f
INNER JOIN pg_catalog.pg_namespace n ON (f.pronamespace = n.oid)
WHERE n.nspname = 'public';
```

- run `\o functions.sql` first to save the output to a file or
- use `\copy (...) to`

Dump database

```
pg_dump -U username -h hostname databasename > dump.sql
```

Import info existing database

```
psql -d newdb -f dump.sql
```

Dump and Restore Sperling Database

```
pg_dump --no-owner -U sperling_prod -h mimir1.aco.net sperling_prod > dump.sql


psql -h localhosst sperling_devel postgres -f dump.sql
```

Major Version Upgrade on Debian

- make a snapshot/backup of the machine
- check the running clusters using `pg_lsclusters` command
- drop the new cluster (the one running the target version) `sudo -u postgres pg_dropcluster --stop 15 main`
- (optional) if you are ssh-ing from a macOS/Item2 the LC_CTYPE is wrongly set
  - run `export LC_CTYPE=en_US.UTF-8 export LC_ALL=en_US.UTF-8`
- upgrade the old cluster `sudo -u postgres pg_upgradecluster 13 main`
- check the running clusters using `pg_lsclusters` command
- drop the old cluster (be sure the upgrade was successful) `sudo -u postgres pg_dropcluster 13 main`
- uninstall old postgresql packages