# Catalog inconsistency

Last updated by | Francisco Javier Pardillo Martin | Feb 3, 2022 at 2:14 AM PST

Contents

## Issue

Getting pg_dump errors that are preventing cx to take dumps:

```
pg_dump: last built-in OID is 16383
pg_dump: reading extensions
pg_dump: identifying extension members
pg_dump: reading schemas
pg_dump: reading user-defined tables
pg_dump: reading user-defined functions
pg_dump: error: schema with OID 307119712 does not exist
```

## Investigation/Analysis

```
select oid,nspname from pg_namespace WHERE oid = 307119712
```

Result is empty, there is no schema with oid 307119712

```
select oid, proname, pronamespace from pg_proc where pronamespace not in (select oid from pg_namespace);
```

Result will show those functions that are associated to missing schemas in the current database.

## Mitigation

Issue can be mitigated by deleting the offending functions or updating the pronamespace column to point to an already existing schema:

```
DELETE FROM pg_proc where pronamespace not in (select oid from pg_namespace);
or
UPDATE pg_proc set pronamespace=2200 where pronamespace not in (select oid from pg_namespace);
```

This will require an ICM to be opened (example ICM: https://portal.microsofticm.com/imp/v3/incidents/details/284895563/home) as those commands require the superuser privilege, if cx tries to execute those commands they will receive following errors:

```
ERROR: permission denied for table pg_proc
```

To prevent this to happen in future, please explain to cx that ddl sentences like “drop schema” should not be executed within a transaction block.

## Repro Steps

You can reproduce the issue by using 2 different sessions, and executing the steps in the following order:

**Session1**

```
1.BEGIN;
3.alter function <function> set schema dummy;
6.commit;
```

**Session2**

```
2.BEGIN;
4.drop schema dummy cascade;
5.commit;
```

If you want to reproduce, please do so in a new server and drop it later as you will create a postgresql catalog inconsistency that will need to involve PG to solve it.

## More Information

Issue is related to “drop schema” sentence not being transaction-safe, it´s not related exclusively to Azure Database for PostgreSQL Flexible or Single server, it´s related to community engine itself.

**How good have you found this content?**

🙂 🙁