# Resolving pg_attribute Corruption and XID Wraparound in PostgreSQL

## 1. Root Cause Analysis

Corruption in pg_attribute: The system catalog pg_attribute is essential for PostgreSQL's operation. Errors like 'cache lookup failed for relation' suggest corruption in this table.

XID Wraparound Risks: Databases must perform regular vacuuming to prevent transaction ID wraparound. Your logs show transaction ages exceeding autovacuum_freeze_max_age.

## 2. Steps to Mitigate

### (a) Immediate Resolution (Post-Corruption Detection)

Since pg_attribute is a system catalog, direct modifications are restricted, and operations like VACUUM or REINDEX may fail due to the corruption. Attempt these steps:

1. Start PostgreSQL in Single-User Mode:

```
postgres --single -D /data/patroni HES
```

2. Check and Fix pg_attribute XID:

```
SELECT oid, relname, relfrozenxid FROM pg_class WHERE relname = 'pg_attribute';
```

```
UPDATE pg_class SET relfrozenxid = (SELECT datfrozenxid FROM pg_database WHERE datname = 'HES')
WHERE oid = <pg_attribute_oid>;
```

3. Attempt Vacuum Again:

```
VACUUM FREEZE pg_catalog.pg_attribute;
```

### (b) Restore Backup

If the corruption persists and is irrecoverable, restoring a clean backup is necessary:

1. Stop PostgreSQL and Restore Backup:

- Use pg_basebackup or another consistent backup tool.

- Ensure the restored data doesn't include the corrupted catalog entries.

2. Ensure Replication Slots are Cleaned:

- List stale replication slots:

  SELECT * FROM pg_replication_slots;

- Drop unused slots:

  SELECT pg_drop_replication_slot('<slot_name>');

**(c) Prevent Future Issues**

1. Increase autovacuum Parameters:

- Ensure that autovacuum processes keep up with XID usage:

  SET autovacuum_work_mem = '1GB';  -- Adjust based on available memory

  SET maintenance_work_mem = '2GB';

- Increase autovacuum_vacuum_cost_limit for faster vacuums:

  SET autovacuum_vacuum_cost_limit = 2000000000;  -- Example value

2. Schedule Regular Vacuum:

- Run regular database-wide VACUUM operations, particularly before reaching autovacuum_freeze_max_age.

**3. Follow-Up Post-Restoration**

1. Verify Data Consistency:

SELECT datname, age(datfrozenxid) FROM pg_database;

Ensure no database is close to the autovacuum_freeze_max_age.

2. Resolve Pending Transactions:

SELECT * FROM pg_prepared_xacts;

Commit or rollback any lingering prepared transactions.

3. Ensure System Stability:

- After restoration, monitor replication, autovacuum logs, and system catalog health to avoid recurrence.