EDB
Postgres for the AI Generation

PostgreSQL 17 new features

Boriss Mejías
Solutions Architect
10th December 2024

Key features and performance improvements in PostgreSQL 17

Releases

Community Releases

PostgreSQL 17 released 2024-09-26

PostgreSQL 17.1 released 2024-11-14

PostgreSQL 17.2 released 2024-11-21

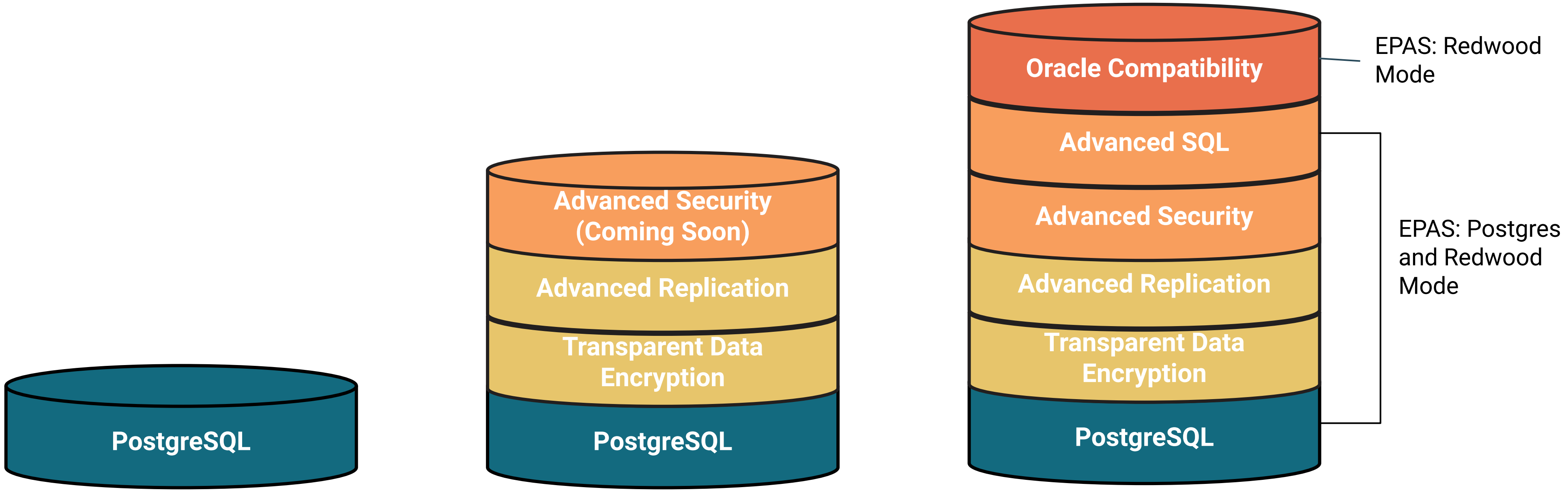
EDB Postgres 17

EDB Postgres Extended (PGE) Server 17.2 released 2024-11-21

EDB Postgres Advanced Server (EPAS) 17.2 released 2024-11-21

EDB tools and supported extensions certified for v17 ~2024-11-21

EDB Postgres databases



Community 360

Standard
EDB Postgres Extended (PGE)

Enterprise
EDB Postgres Advanced Server (EPAS)

Incremental Backups

- Only the changed blocks are copied

- pg_basebackup **`-i`** `old_manifest_file`

  pg_basebackup **`--incremental=`**`old_manifest_file`

- It requires **summarize_wal** enabled

- **pg_combinebackup** is used to reconstruct a full backup

Incremental Backups

- Test Challenge for PGDay Lowlands

Incremental Backups

- Barman 3.11.0 released 2024-08-22 supports it for PG 17 Beta3
- Barman 3.12.1 released 2024-11-21 latest

Incremental Backup

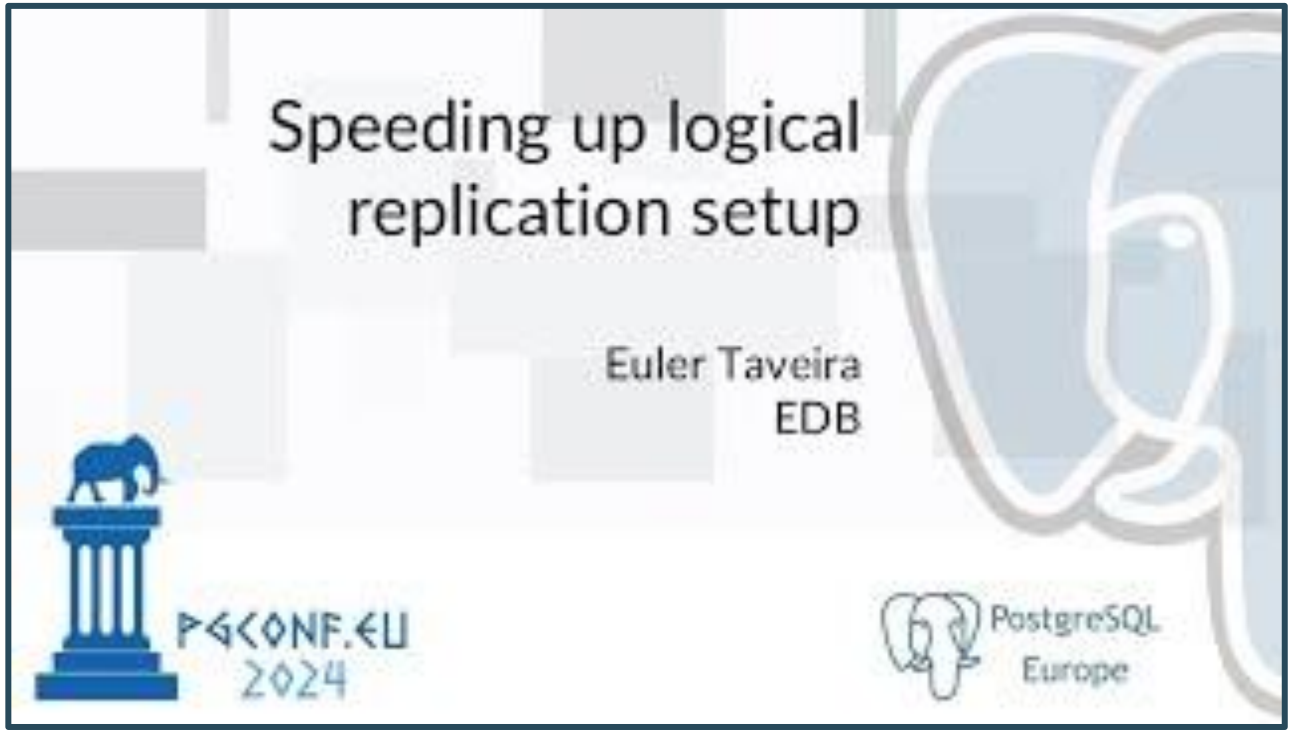
Incremental Backup
Robert Haas
EnterpriseDB

https://www.youtube.com/watch?v=F_uqARRaqNw

Logical Replication

- Failover of replication slots
- New tool **pg_createsubscriber**
- pg_upgrade preserves replication slots

pg_createsubscriber



Speeding up logical replication setup

Euler Taveira
EDB

PGCONF.EU
2024

PostgreSQL
Europe

https://www.youtube.com/watch?v=LPtmM2Dkdvg

Partitioning

- Allow partitions to be merged using

```
ALTER TABLE … MERGE PARTITIONS
```

- Allow partitions to be split using

```
ALTER TABLE … SPLIT PARTITION
```

And much more…

- Security
  - `pg_maintain` avoids superuser privileges for following: VACUUM, ANALYZE, CLUSTER, REFRESH MATERIALIZED VIEW, REINDEX, and LOCK TABLE on all relations
  - Make TLS connections without a network round-trip negotiation
- SQL
  - `COPY ON ERROR 'ignore'`
  - `JSON_TABLE()`
- Monitoring
  - New views:
    - `pg_wait_events`
    - `pg_stat_checkpointer`
  - Index Vacuum progress in `pg_stat_progress_vacuum`

CloudNativePG

- Postgres 17 supported
- Every single commit can be tested

More software PG17 ready

- TPA Trusted Postgres Architect
- PGD Postgres Distributed

What about AI

- pg_vector grows independently
- EDB AIDB integrates pg_vector with retrievers and generators

More details are here

https://www.postgresql.org/docs/17/release-17.html

Reducing Migration Times: EPAS 17

EDB

EDB Postgres Advanced Server (EPAS)

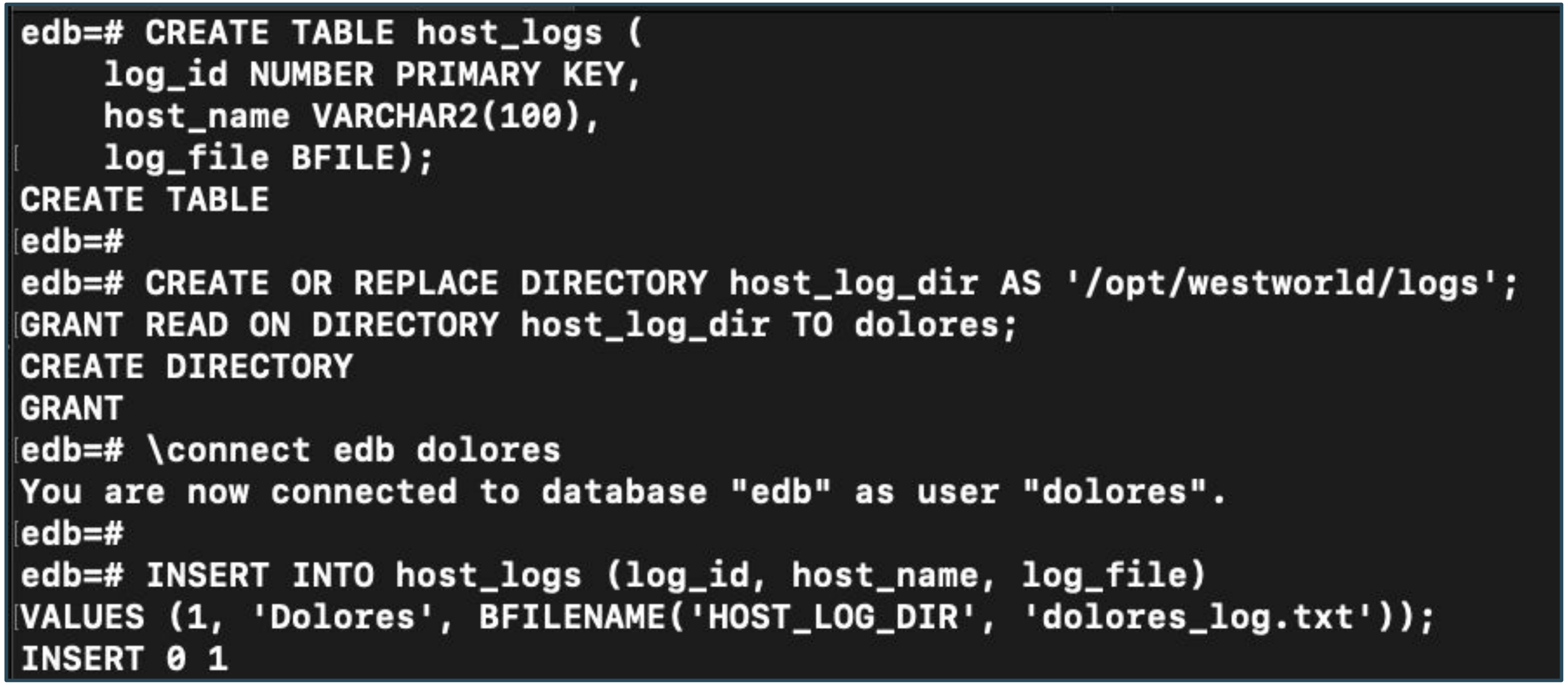
Q4 2024 (v17)

- Merge with upstream PostgreSQL:
  - Community Enhancements
- Simplified Migration for Legacy Oracle Applications:
  - XMLType Support
  - BFILE Support
- Oracle Compatibility enhancements based on Migration Portal:
  - SQL: NLS_UPPER, NLS_LOWER, NLS_INITCAP, SAVE EXCEPTION FOR ALL, FOR ALL MERGE
- Customer Requests and Miscellaneous:
  - JSON Support for EDB Audit Log, Directory Permissions, EDB Loader improvements, OCI Import Foreign Schema, Conceal Password

Bfile

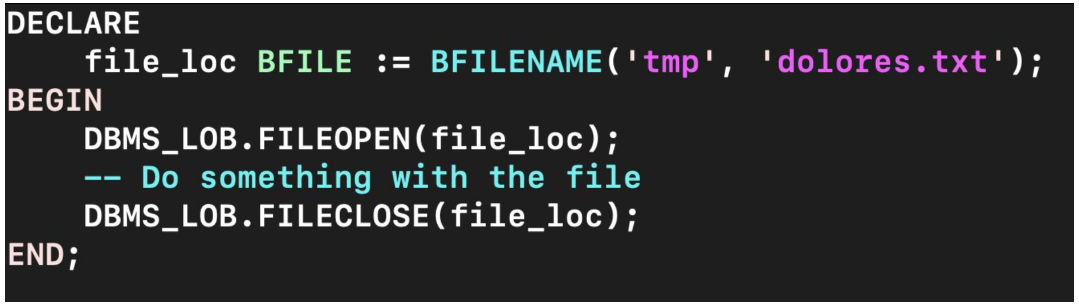
New native data type in EPAS 17

- Needed to expand DBMS_LOB support
    - Reached limit with other Binary Data types



```
edb=# CREATE TABLE host_logs (
    log_id NUMBER PRIMARY KEY,
    host_name VARCHAR2(100),
    log_file BFILE);
CREATE TABLE
edb=#
edb=# CREATE OR REPLACE DIRECTORY host_log_dir AS '/opt/westworld/logs';
GRANT READ ON DIRECTORY host_log_dir TO dolores;
CREATE DIRECTORY
GRANT
edb=# \connect edb dolores
You are now connected to database "edb" as user "dolores".
edb=#
edb=# INSERT INTO host_logs (log_id, host_name, log_file)
VALUES (1, 'Dolores', BFILENAME('HOST_LOG_DIR', 'dolores_log.txt'));
INSERT 0 1
```

DBMS_LOB



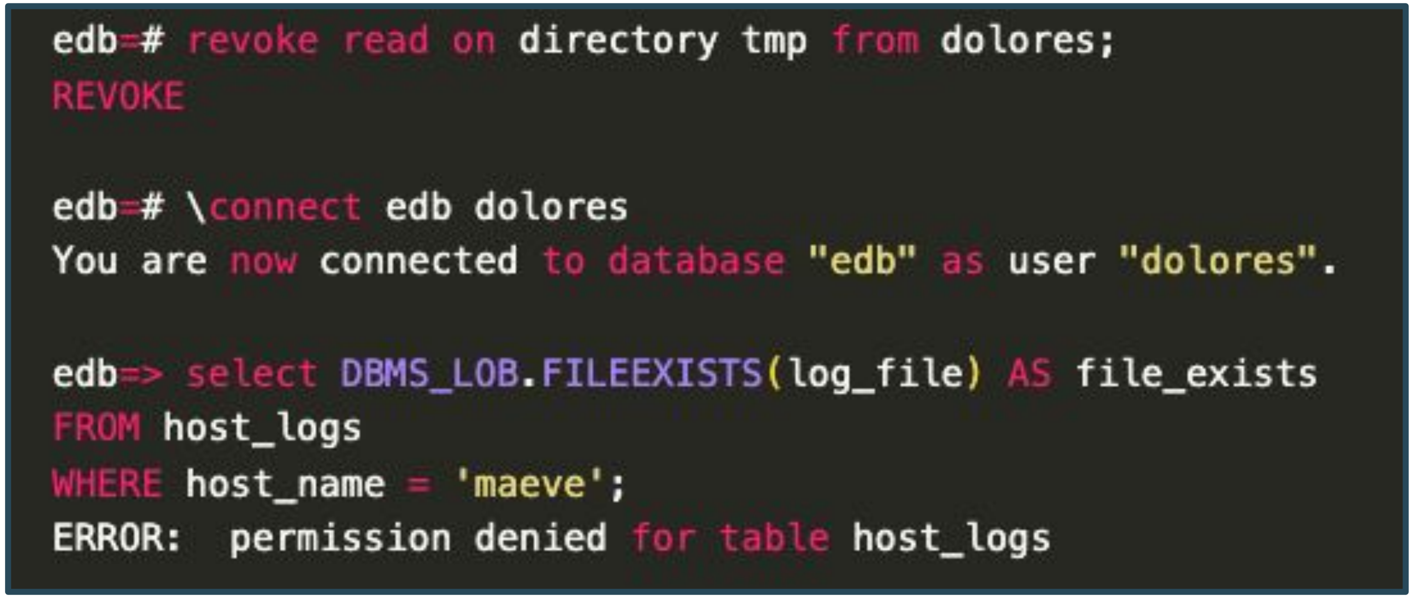
```
DECLARE
    file_loc BFILE := BFILENAME('tmp', 'dolores.txt');
BEGIN
    DBMS_LOB.FILEOPEN(file_loc);
    -- Do something with the file
    DBMS_LOB.FILECLOSE(file_loc);
END;
```

	v16	v17
APPEND	✓	✓
COMPARE	✓	✓
CONVERTTOBLOB	✓	✓
CONVERTTOCLOB	✓	✓
COPY	✓	✓
ERASE	✓	✓
FILECLOSE, CLOSE		✓
FILECLOSEALL		✓
FILEEXISTS		✓
FILEGETNAME		✓
FILEISOPEN, ISOPEN		✓
FILEOPEN, OPEN		✓
GET_STORAGE_LIMIT	✓	✓
GETLENGTH	✓	✓
INSTR	✓	✓
LOADBLOBFROMFILE		✓
LOADCLOBFROMFILE		✓
READ	✓	✓
SUBSTR		
TITLE NEEDED		✓
TRIM	✓	✓
WRITE	✓	✓
WRITEAPPEND	✓	✓

Directory Permissions

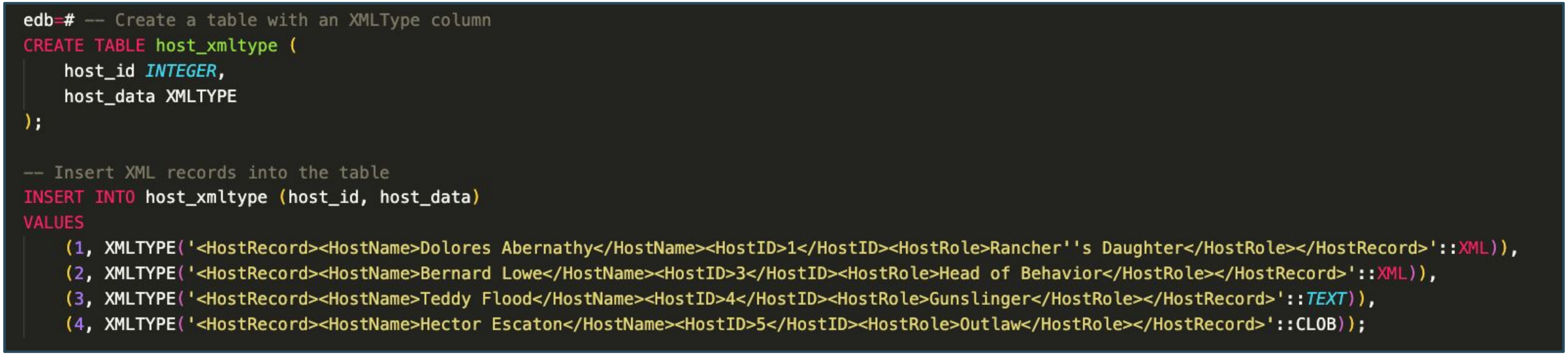
Before PostgreSQL v17, only superusers could create or drop directories, and there was no mechanism to grant these privileges to non-superusers

```
GRANT { CREATE | DROP }
        ANY DIRECTORY
        TO role_specification [, ...]
        [ GRANTED BY role_specification ]

GRANT { READ | WRITE | ALL [ PRIVILEGES ] }
        ON DIRECTORY directory_name [, ...]
        TO role_specification [, ...] [ WITH GRANT OPTION ]
        [ GRANTED BY role_specification ]

REVOKE [ GRANT OPTION FOR ]
        { READ | WRITE | ALL [ PRIVILEGES ] }
        ON DIRECTORY  directory_name [, ...]
        FROM role_specification [, ...]
        [ GRANTED BY role_specification ]
        [ CASCADE | RESTRICT ]
```



```
edb=# revoke read on directory tmp from dolores;
REVOKE

edb=# \connect edb dolores
You are now connected to database "edb" as user "dolores".

edb=> select DBMS_LOB.FILEEXISTS(log_file) AS file_exists
FROM host_logs
WHERE host_name = 'maeve';
ERROR:  permission denied for table host_logs
```

XMLType

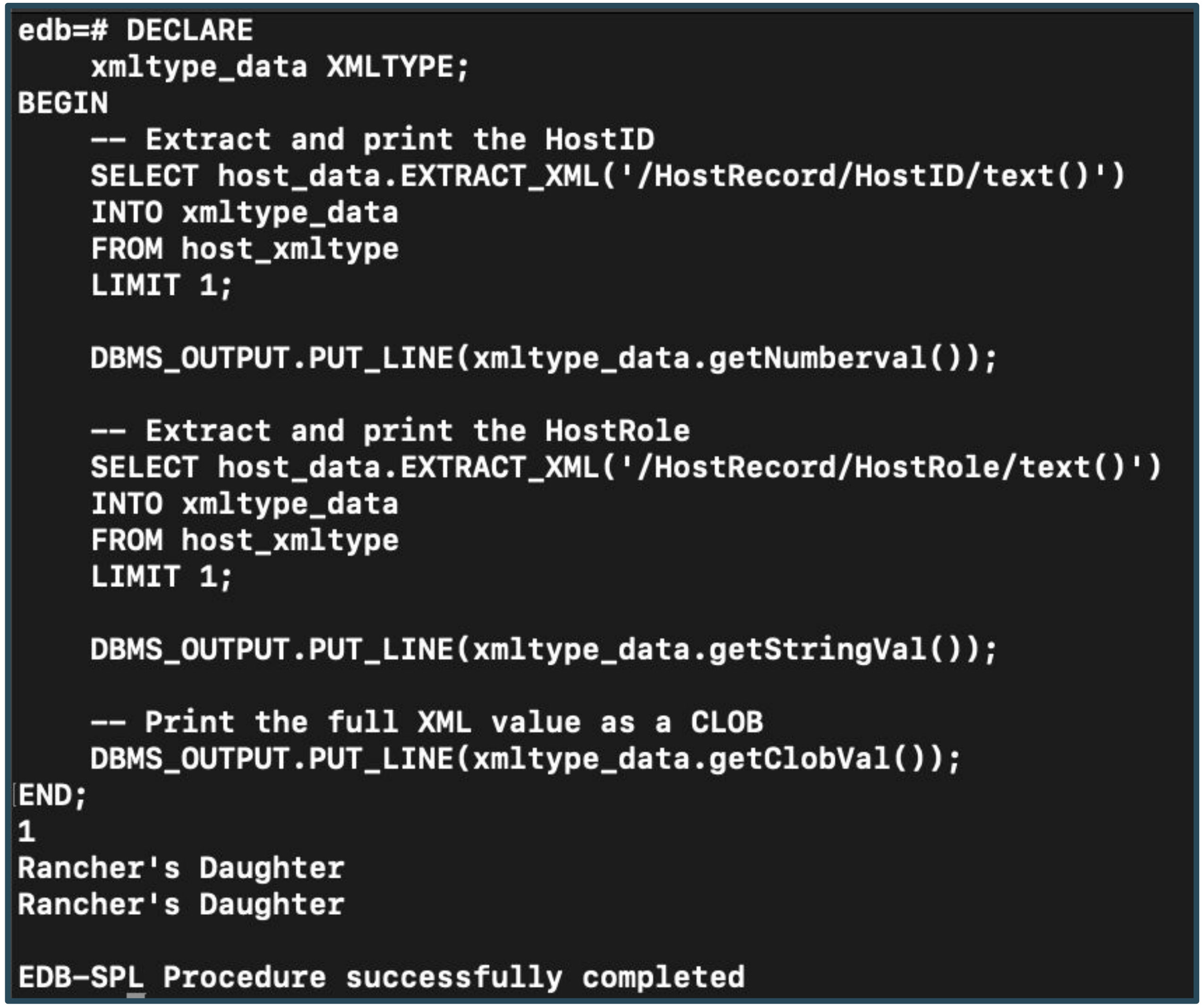


```
edb=# -- Create a table with an XMLType column
CREATE TABLE host_xmltype (
    host_id INTEGER,
    host_data XMLTYPE
);

-- Insert XML records into the table
INSERT INTO host_xmltype (host_id, host_data)
VALUES
    (1, XMLTYPE('<HostRecord><HostName>Dolores Abernathy</HostName><HostID>1</HostID><HostRole>Rancher''s Daughter</HostRole></HostRecord>'::XML)),
    (2, XMLTYPE('<HostRecord><HostName>Bernard Lowe</HostName><HostID>3</HostID><HostRole>Head of Behavior</HostRole></HostRecord>'::XML)),
    (3, XMLTYPE('<HostRecord><HostName>Teddy Flood</HostName><HostID>4</HostID><HostRole>Gunslinger</HostRole></HostRecord>'::TEXT)),
    (4, XMLTYPE('<HostRecord><HostName>Hector Escaton</HostName><HostID>5</HostID><HostRole>Outlaw</HostRole></HostRecord>'::CLOB));
```

XMLType

Example

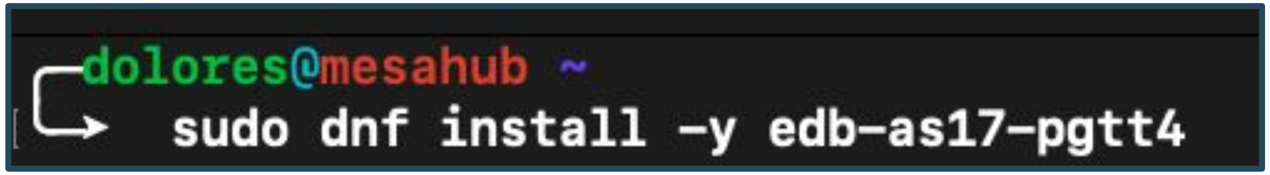
- Use member function to extract and print the host’s ID as a NUMBER
- Extract and print the host’s role as a string
- Return the entire XML value as a CLOB



```
edb=# DECLARE
    xmltype_data XMLTYPE;
BEGIN
    -- Extract and print the HostID
    SELECT host_data.EXTRACT_XML('/HostRecord/HostID/text()')
    INTO xmltype_data
    FROM host_xmltype
    LIMIT 1;

    DBMS_OUTPUT.PUT_LINE(xmltype_data.getNumberval());

    -- Extract and print the HostRole
    SELECT host_data.EXTRACT_XML('/HostRecord/HostRole/text()')
    INTO xmltype_data
    FROM host_xmltype
    LIMIT 1;

    DBMS_OUTPUT.PUT_LINE(xmltype_data.getStringVal());

    -- Print the full XML value as a CLOB
    DBMS_OUTPUT.PUT_LINE(xmltype_data.getClobVal());
END;
1
Rancher's Daughter
Rancher's Daughter

EDB-SPL Procedure successfully completed
```

PGTT

Added Support for Postgres Global Temporary Tables (pgtt)



```
dolores@mesahub ~
    sudo dnf install -y edb-as17-pgtt4
```

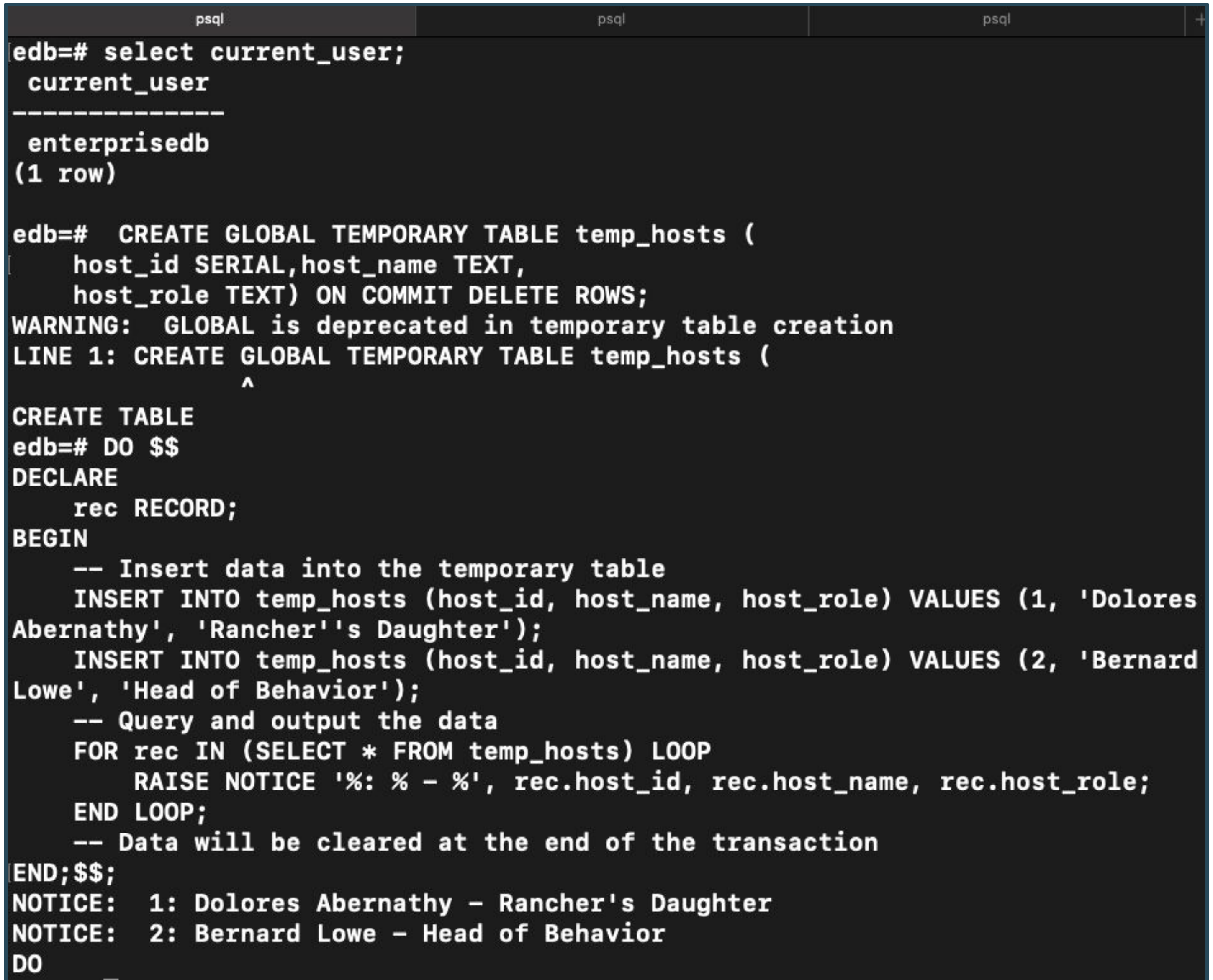
Overview

- Available for PostgreSQL, PGE, and EPAS 13-17
- A PostgreSQL extension providing functionality similar to Oracle Global Temporary Tables (GTT)
- Enables session- or transaction-specific temporary data storage

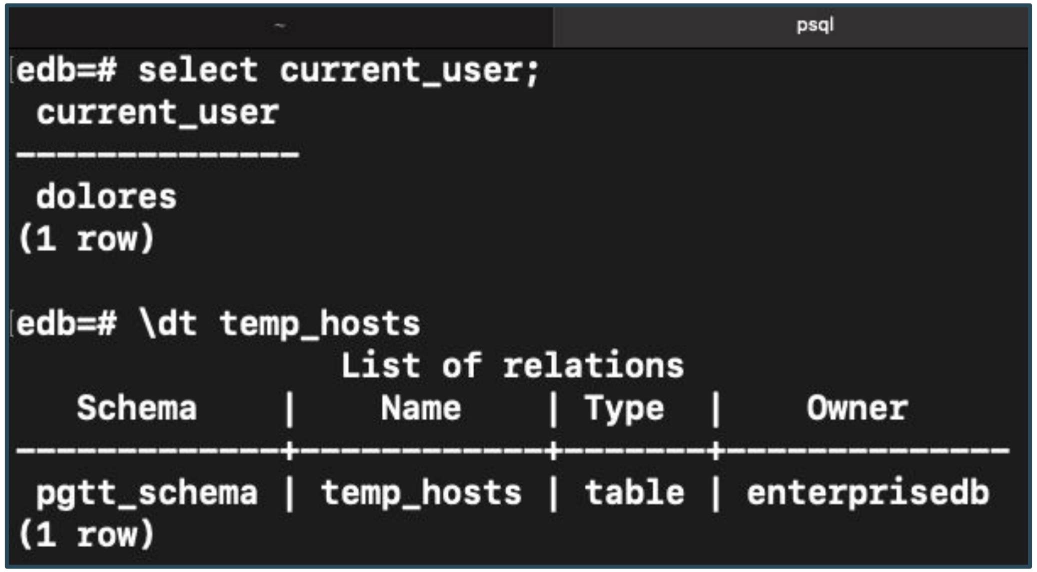
Why Add PGT to the Supported List?

- Enhanced Oracle Compatibility
- Developer Productivity

PGTT Example - Two Sessions



```
edb=# select current_user;
 current_user
--------------
 enterprisedb
(1 row)

edb=#  CREATE GLOBAL TEMPORARY TABLE temp_hosts (
    host_id SERIAL,host_name TEXT,
    host_role TEXT) ON COMMIT DELETE ROWS;
WARNING:  GLOBAL is deprecated in temporary table creation
LINE 1: CREATE GLOBAL TEMPORARY TABLE temp_hosts (
               ^
CREATE TABLE
edb=# DO $$
DECLARE
    rec RECORD;
BEGIN
    -- Insert data into the temporary table
    INSERT INTO temp_hosts (host_id, host_name, host_role) VALUES (1, 'Dolores
Abernathy', 'Rancher''s Daughter');
    INSERT INTO temp_hosts (host_id, host_name, host_role) VALUES (2, 'Bernard
Lowe', 'Head of Behavior');
    -- Query and output the data
    FOR rec IN (SELECT * FROM temp_hosts) LOOP
        RAISE NOTICE '%: % - %', rec.host_id, rec.host_name, rec.host_role;
    END LOOP;
    -- Data will be cleared at the end of the transaction
END;$$;
NOTICE:  1: Dolores Abernathy - Rancher's Daughter
NOTICE:  2: Bernard Lowe - Head of Behavior
DO
```



```
edb=# select current_user;
 current_user
--------------
 dolores
(1 row)

edb=# \dt temp_hosts
               List of relations
   Schema    |    Name    | Type  |    Owner
-------------+------------+-------+--------------
 pgtt_schema | temp_hosts | table | enterprisedb
(1 row)
```

Closing Words

- Plenty of interesting improvements on PG17
  - Incremental backups
  - Logical Replication
  - Table Partitioning
- Postgres does not have a community,
  it is a community

EDB
Postgres® for the AI Generation

Survey & Prize Draw

Scan the QR code and share your feedback with us - we value your opinion!

Submit the form now and win the

PostgreSQL 16 Administration Cookbook

Thank you