Major Features: Postgres 15

BRUCE MOMJIAN

POSTGRESQL is an open-source, full-featured relational database. This presentation gives an overview of the Postgres 15 release.

https://momjian.us/presentations

Last updated: April 2023

Postgres 15 Feature Outline

1. MERGE
2. Logical replication
3. Compression
4. Granular permissions
5. Memory
6. COPY headers

Full item list at https://momjian.us/pgsql_docs/release-15.html and https://www.postgresql.org/docs/15/release-15.html.

1. MERGE

- Part of the SQL standard, often requested
- Similar to INSERT … ON CONFLICT, except
  - join oriented, not row oriented
  - does not require a unique index
  - can error on concurrent changes

INSERT … ON CONFLICT

```
CREATE TABLE test (x INTEGER, y BOOLEAN);

INSERT INTO test VALUES (1), (3), (5);

INSERT INTO test VALUES (1), (2), (3), (4), (5), (6)
ON CONFLICT (x) DO UPDATE SET y = TRUE;
ERROR:  there is no unique or exclusion constraint matching the ON CONFLICT specification

CREATE UNIQUE INDEX i_test ON test (x);

INSERT INTO test VALUES (1), (2), (3), (4), (5), (6)
ON CONFLICT (x) DO UPDATE SET y = TRUE;

SELECT * FROM test;
 x |   y
---+--------
 1 | t
 2 | (null)
 3 | t
 4 | (null)
 5 | t
 6 | (null)
```

Merge

```
DELETE FROM test;

INSERT INTO test VALUES (1), (3), (5);

MERGE INTO test
USING (VALUES (1), (2), (3), (4), (5), (6)) m (x)
ON test.x = m.x
WHEN NOT MATCHED THEN
     INSERT (x) VALUES (m.x)
WHEN MATCHED THEN
     UPDATE SET y = TRUE;

SELECT * FROM test;
 x |   y
---+--------
 1 | t
 2 | (null)
 3 | t
 4 | (null)
 5 | t
 6 | (null)
```

2. Logical Replication

Logical replication is now more flexible by allowing

- Publication of entire schemas, including future table additions
- Publication row control with a WHERE clause
- Publication column control
- Subscribers to skip specific transactions

Additional features are:

- Support for prepared transactions
- Suppress replication of empty transactions
- Possible replication termination on error

3. Compression

- Add LZ4 compression to the base backup protocol (gzip was already supported)
- Add LZ4 and Zstandard compression of pg_basebackup files (gzip was already supported)
- Allow pg_basebackup to control if compression happens server-side or client-side
- Add LZ4 compression to pg_receivelog (gzip was already supported)
- Add LZ4 and Zstandard compression of full page writes (LZ was already supported)

4. Granular Permissions

- Allow view to be run with the permissions of the view user, not owner
- Allow GRANT to control changes to server-side variables
- Add predefined role with checkpoint permission

5. Memory

- Improve performance of sorts that exceed work_mem
- Improve performance and reduce memory usage of in-memory sorts
- Store run-time server statistics in shared memory, rather than on disk
- Make hashing by default use twice as much memory as other operations
- Add server variable to report the amount of used shared memory and huge pages

6. COPY Headers

```
CREATE TABLE copytest (x INTEGER, y TEXT);

INSERT INTO copytest VALUES (1, 'My term paper'), (2, 'Crossword puzzle');

COPY copytest to STDOUT;
1       My term paper
2       Crossword puzzle

COPY copytest to STDOUT WITH (HEADER);
x       y
1       My term paper
2       Crossword puzzle
```

Previously only COPY's CSV mode supported headers.

COPY Headers

```
COPY copytest TO '/tmp/p' WITH (HEADER);

DELETE FROM copytest;

COPY copytest FROM '/tmp/p';
ERROR:  invalid input syntax for type integer: "x"
CONTEXT:  COPY copytest, line 1, column x: "x"

COPY copytest FROM '/tmp/p' WITH (HEADER);

SELECT * FROM copytest;
 x |        y
---+-----------------
 1 | My term paper
 2 | Crossword puzzle
```

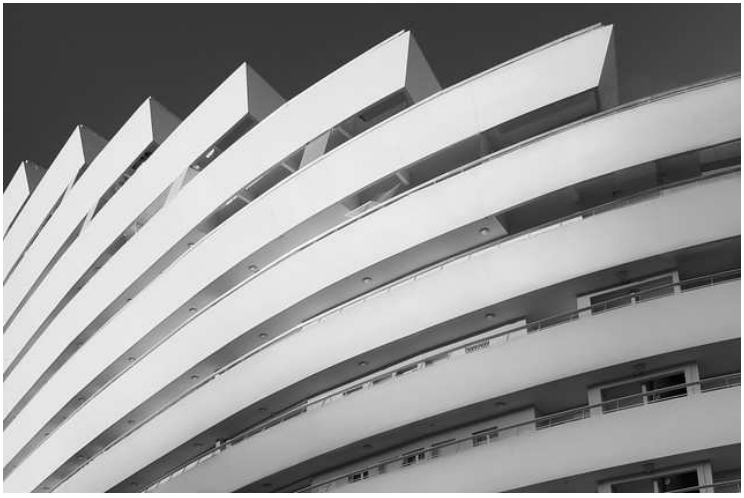
COPY HEADER Verification

```
ALTER TABLE copytest RENAME y TO z;

COPY copytest FROM '/tmp/p' WITH (HEADER);

COPY copytest FROM '/tmp/p' WITH (HEADER MATCH);
ERROR:  column name mismatch in header line field 2: got "y", expected "z"
CONTEXT:  COPY copytest, line 1: "x     y"
```

Conclusion



https://momjian.us/presentations

https://www.flickr.com/photos/thomasletholsen/