

# What's **new** in PostgreSQL 15

*Unless it's reverted before shipping..*

Daniel Gustafsson

# MERGE

Conditionally insert, update or delete rows of a table

Not an upsert, we have `INSERT .. ON CONFLICT DO ..` for that; it's a conditional bulkload for merging a dataset rather than adding it

Originally committed for PostgreSQL 11 but was reverted

Uses existing privileges for insert, update and delete

Not supported for materialized views, foreign tables or tables with rules

# MERGE is not UPSERT

An upsert is an atomic operation

Upsert with `INSERT .. ON CONFLICT DO ..` requires a collision with an existing committed and visible to snapshots tuple

Merge is a transformation tool for bulk loading with potential side effects

```
MERGE INTO target_table  
USING data_source ON join_condition  
WHEN when_clause
```

*First, produce candidate rows by  
joining data\_source to target\_table*



```
MERGE INTO target_table  
USING data_source ON join_condition  
WHEN when_clause
```

```
MERGE INTO target_table  
USING data_source ON join_condition  
WHEN when_clause
```



*For each candidate rows, check if  
matching and apply when\_clause*

```
MERGE INTO wines w
USING wine_stock_changes s
ON s.winename = w.winename
WHEN NOT MATCHED AND s.delta > 0 THEN
    INSERT VALUES(s.winename, s.delta)
WHEN MATCHED AND w.stock + s.delta > 0 THEN
    UPDATE SET stock = w.stock + s.delta
WHEN MATCHED THEN
    DELETE;
```

```
MERGE INTO wines w
USING wine_stock_changes s
ON s.winename = w.winename
```

*Update inventory  
based on changes*

```
WHEN NOT MATCHED AND s.delta > 0 THEN
```

```
    INSERT VALUES(s.winename, s.delta)
```

```
WHEN MATCHED AND w.stock + s.delta > 0 THEN
```

```
    UPDATE SET stock = w.stock + s.delta
```

```
WHEN MATCHED THEN
```

```
    DELETE;
```



```
MERGE INTO wines w
```

*Add new products*

```
USING wine_stock_changes s
```

```
ON s.winename = w.winename
```

```
WHEN NOT MATCHED AND s.delta > 0 THEN
```

```
    INSERT VALUES(s.winename, s.delta)
```

```
WHEN MATCHED AND w.stock + s.delta > 0 THEN
```

```
    UPDATE SET stock = w.stock + s.delta
```

```
WHEN MATCHED THEN
```

```
    DELETE;
```

```
MERGE INTO wines w
USING wine_stock_changes s
ON s.winename = w.winename
```

*Update inventory on  
existing products*

```
WHEN NOT MATCHED AND s.delta > 0 THEN
    INSERT VALUES(s.winename, s.delta)
```

```
WHEN MATCHED AND w.stock + s.delta > 0 THEN
    UPDATE SET stock = w.stock + s.delta
```

```
WHEN MATCHED THEN
    DELETE;
```

```
MERGE INTO wines w                                Remove sold-out  
USING wine_stock_changes s                        products  
ON s.winename = w.winename  
WHEN NOT MATCHED AND s.delta > 0 THEN  
    INSERT VALUES(s.winename, s.delta)  
WHEN MATCHED AND w.stock + s.delta > 0 THEN  
    UPDATE SET stock = w.stock + s.delta  
WHEN MATCHED THEN  
    DELETE;
```