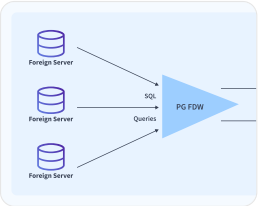
Liquibase vs pgloader

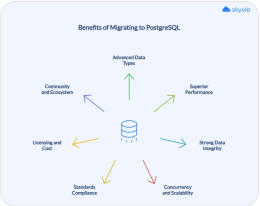
can we do the same thing when we want to migrate the date from one data base to another databse with liquibase that the pgloader do

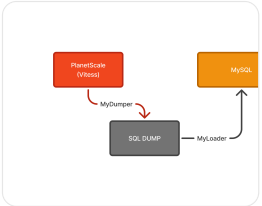
Short answer: **No — Liquibase cannot fully replace what `pgloader` does.**
Long answer: **They solve different problems**, and you usually **use them together**, not instead of each other.

I'll explain this **very clearly and deeply**, step by step, so you have **zero doubts**.

🔹 What pgloader actually does (Data Migration Tool)







pgloader is designed for:

- ✔ Migrating **entire databases**
- ✔ Moving **schema + data together**
- ✔ One-time or bulk migrations

Example:

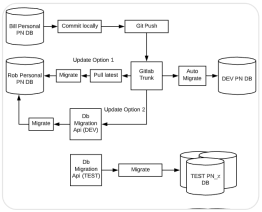
- MySQL → PostgreSQL
- Oracle → PostgreSQL
- SQLite → PostgreSQL

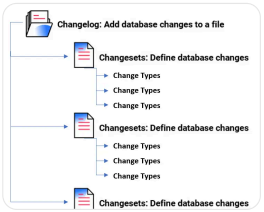
What pgloader migrates automatically:

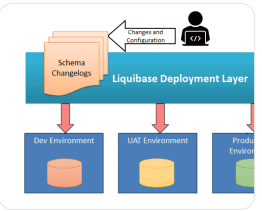
Item	Migrated
Tables	✅
Columns	✅
Data (rows)	✅
Indexes	✅
Primary / Foreign Keys	✅
Sequences	✅
Data type conversion	✅
Millions of rows (fast)	✅

👉 **pgloader is a DATA MIGRATION ENGINE**

🔹 What Liquibase actually does (Schema Versioning Tool)







Liquibase is designed for:

- ✔ Managing **database schema changes over time**
- ✔ Keeping **Dev / QA / Prod in sync**
- ✔ Version-controlling DB changes

Liquibase focuses on:

Item	Supported
Create / Alter tables	✅

Item	Supported
Add / remove columns	✅
Indexes & constraints	✅
Rollback support	✅
Tracking changes	✅
Bulk data migration	❌
Cross-DB auto conversion	❌

👉 **Liquibase is a SCHEMA CHANGE MANAGEMENT TOOL**

🔹 Direct Comparison (Very Important)

Feature	pgloader	Liquibase
Migrate MySQL → PostgreSQL	✅	❌
Move large volumes of data	✅	❌
Auto convert data types	✅	❌
One-time full DB migration	✅	❌
Track schema changes	❌	✅
CI/CD integration	❌	✅
Rollback support	❌	✅

❓ Can Liquibase migrate data from one DB to another?

🔴 Not like pgloader ❌

Liquibase **cannot**:

- Read from Database A
- Transform data
- Load into Database B automatically

There is **NO built-in mechanism** for:

- Cross-database data extraction
- Data type auto-conversion
- Bulk row movement

🟡 What Liquibase *can* do (Limited Data Migration)

Liquibase **can move small/static data**, like:

- Default roles
- Configuration tables
- Master/reference data

Example (valid use case):

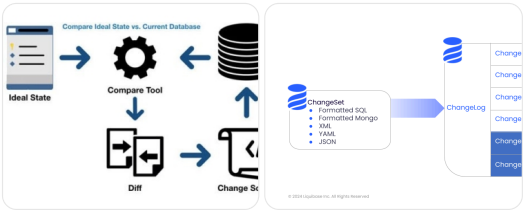
```sql
INSERT INTO roles (id, name) VALUES (1, 'ADMIN');
```

❌ Bad use case:

- Migrating 1 million users
- Migrating transactional history
- Migrating logs or analytics data

✅ Correct & Industry-Standard Approach (REAL WORLD)

🔥 Best Practice Architecture



Step 1: Use pgloader

```
text

Old DB (MySQL)
     ↓
pgloader
     ↓
New DB (PostgreSQL)
```

- ✔ Schema created
- ✔ Data migrated
- ✔ Types converted

Step 2: Put Liquibase on top

```
text

Liquibase starts here
↓
Future schema changes
↓
Dev → QA → Prod
```

- ✔ Version control
- ✔ Safe deployments
- ✔ Team collaboration

🧠 Why Liquibase SHOULD NOT be used for full data migration

If you try:

- Huge SQL insert scripts
- Export → insert manually
- Data movement via Liquibase

You will face:

❌ Slow builds
❌ Transaction failures
❌ Memory issues
❌ Rollback nightmares
❌ CI/CD pipeline crashes

Liquibase is **not designed for this job**.

🧩 When to use WHAT (Simple Rule)

Use pgloader when:

- ✔ Migrating **data**
- ✔ Switching **database engines**
- ✔ One-time bulk migration

Use Liquibase when:

- ✔ Managing **schema evolution**
- ✔ Working in **teams**
- ✔ Maintaining **environments**

🎯 Final Verdict (Very Clear)

> ❌ **Liquibase cannot do what pgloader does**
> ✅ **pgloader + Liquibase together = perfect solution**

They are **complementary tools**, not competitors.

If you want, next I can:

- Show a **real MySQL → PostgreSQL migration example**
- Explain **how teams safely combine pgloader + Liquibase**

- Design a **production-grade migration plan**
- Explain **why Flyway vs Liquibase also matters**

Just tell me 👍

Is this conversation helpful so far?