# Database Migration Analysis Report

## 1️⃣ Introduction

This report outlines the process and results of migrating data from a MySQL database to a PostgreSQL database. The primary objectives were to ensure a smooth transition, preserve data integrity, and maintain compatibility for future operations.

## 2️⃣ Migration Goals

- Successfully transfer schema and data from MySQL to PostgreSQL  
- Ensure all relationships, constraints, and indexes are preserved  
- Maintain 100% data accuracy and integrity  
- Minimize downtime during migration

## 3️⃣ Migration Process

- Step 1: Export data from MySQL using `mysqldump`.

- Step 2: Create target database in PostgreSQL.

- Step 3: Convert MySQL dump to PostgreSQL-compatible format using `pgloader`.

- Step 4: Run migration and import data into PostgreSQL.

- Step 5: Perform data integrity checks to ensure consistency.

- Step 6: Log all migration steps for auditing purposes.

## 4️⃣ Data Integrity Checks

After migration, integrity checks were conducted to confirm the accuracy of the data transfer:

- Row counts matched for all tables between MySQL and PostgreSQL.  
- Primary keys contained no NULL values.  
- Foreign key relationships were preserved.  
- Random spot checks confirmed accurate data values.

## 5️⃣ Migration Summary

|  |  |  |
| --- | --- | --- |
| Step | Description | Status |
| 1 | Exported data from MySQL | Completed |
| 2 | Created PostgreSQL database | Completed |
| 3 | Converted schema using pgloader | Completed |
| 4 | Migrated data to PostgreSQL | Completed |
| 5 | Performed data integrity checks | Completed |

## 6️⃣ Key Observations

- Migration was completed without data loss.

- PostgreSQL performance improved for analytical queries.

- Some MySQL-specific data types (e.g., `TINYINT`) required conversion to PostgreSQL equivalents.

- Indexes were rebuilt to optimize query performance post-migration.

## 7️⃣ Recommendations

- Test application queries against the new PostgreSQL database before production switch.

- Schedule regular backups post-migration.

- Use schema migration tools for future upgrades to reduce manual work.

## 8️⃣ Conclusion

The migration from MySQL to PostgreSQL was successfully completed with no data loss and all integrity constraints preserved. This transition enables better performance, advanced SQL capabilities, and scalability for future business needs.