SiLVue OI SYSTEM DATABASE ENGINE

SCADA REKINTEK

****

Dobby Akhmadi Putra

**PT LEN INDUSTRI**

BANDUNG

2021

Content

[A. General Database Engines Comparison By (Features) 4](#_Toc76652733)

[a. Database Engine Ranking 4](#_Toc76652734)

[b. Comparison Between MariaDB VS MongoDB 4](#_Toc76652735)

[c. Comparison Between MariaDB VS MYSQL 5](#_Toc76652736)

[d. Comparison Between MariaDB VS PosgreSQL 6](#_Toc76652737)

[**e.** **Result Comparison For SILVUE OI Requirements** 7](#_Toc76652738)

[B. Comparasion Performance DBMS 7](#_Toc76652739)

[a. Insert Operation Query Per ms 7](#_Toc76652740)

[b. Select And Update Operation Query Per ms 8](#_Toc76652741)

[c. High Availability and Cloud Computing Replication 8](#_Toc76652742)

[**d.** **Comparasion Security Model** 9](#_Toc76652743)

[e. Total Maximum Size DBMS 10](#_Toc76652744)

[a. Mariadb/sql Maximum Size 10](#_Toc76652745)

[a. Mongodb Maximum Size 10](#_Toc76652746)

[C. Database Backup And Restore Compressed with Encryption 11](#_Toc76652747)

[a. Backup and Restore Encrypt With OpenSSL 11](#_Toc76652748)

[b. Compression gzip Backup and Restore Encrypting With Open 11](#_Toc76652749)

[D. Hardware Optimization 11](#_Toc76652750)

[a. Memory (RAM) 11](#_Toc76652751)

[b. Disks (HDD/SSD) 11](#_Toc76652752)

[a. CPU/Processor 12](#_Toc76652753)

[E. Database Benchmarking 12](#_Toc76652754)

[a. Select Query 10k Rows Mariadb Example Mariadb FEP Database 12](#_Toc76652755)

[b. Insert Query Example for 10 Rows Mariadb 12](#_Toc76652756)

[c. Insert Query Example for 3K Rows Mongodb 13](#_Toc76652757)

[d. Insert Query Example for 10k rows Postgreesql 15](#_Toc76652758)

[F. Database Module 16](#_Toc76652759)

[a. Database Design 16](#_Toc76652760)

[b. Function Structure Protocol 17](#_Toc76652761)

[c. Function Structure Slave Variable and Master Variable 17](#_Toc76652762)

[d. Function Redundant Settings 17](#_Toc76652763)

[e. Function Editor/General Settings 17](#_Toc76652764)

[f. Function Login with user Level 17](#_Toc76652765)

[G. Historian Module 17](#_Toc76652766)

[a. Realtime Graph Ex High Low Average with auto backup period. 17](#_Toc76652767)

[b. Archive Graph Ex High Low Average with Restore database from latest backup 17](#_Toc76652768)

[H. Log System Module 17](#_Toc76652769)

[a. Export And Import File To Ex: SQL , TXT ,CSV ,excel,Cloud Etc 17](#_Toc76652770)

[b. Event List and Log. 17](#_Toc76652771)

[I. Alarm Module 17](#_Toc76652772)

[a. Filter Event Alarm Status Ex: High Low Mid for each Category Function 17](#_Toc76652773)

[J. Redundancy Services Runtime Module 17](#_Toc76652774)

[**a.** Database Replication For Client Server **Dominant** And **Non Dominant** 17](#_Toc76652775)

[b. **Keep Alive** Replication Operating System. 17](#_Toc76652776)

[K. System Requirements Database Engine Module 17](#_Toc76652777)

[a. **Belum kepikiran lagi cek teknis dlu** 17](#_Toc76652778)

# General Database Engines Comparison By (Features)

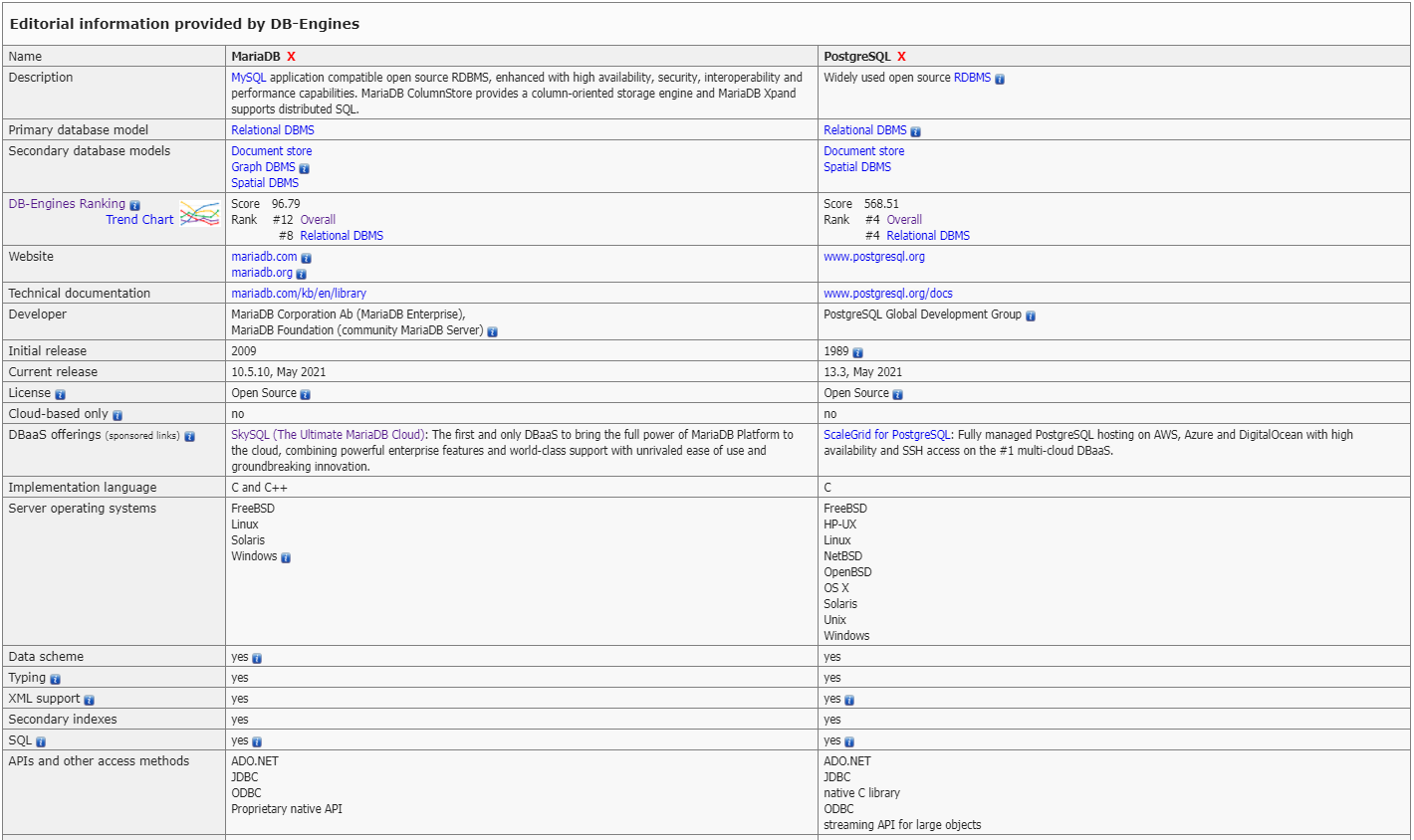
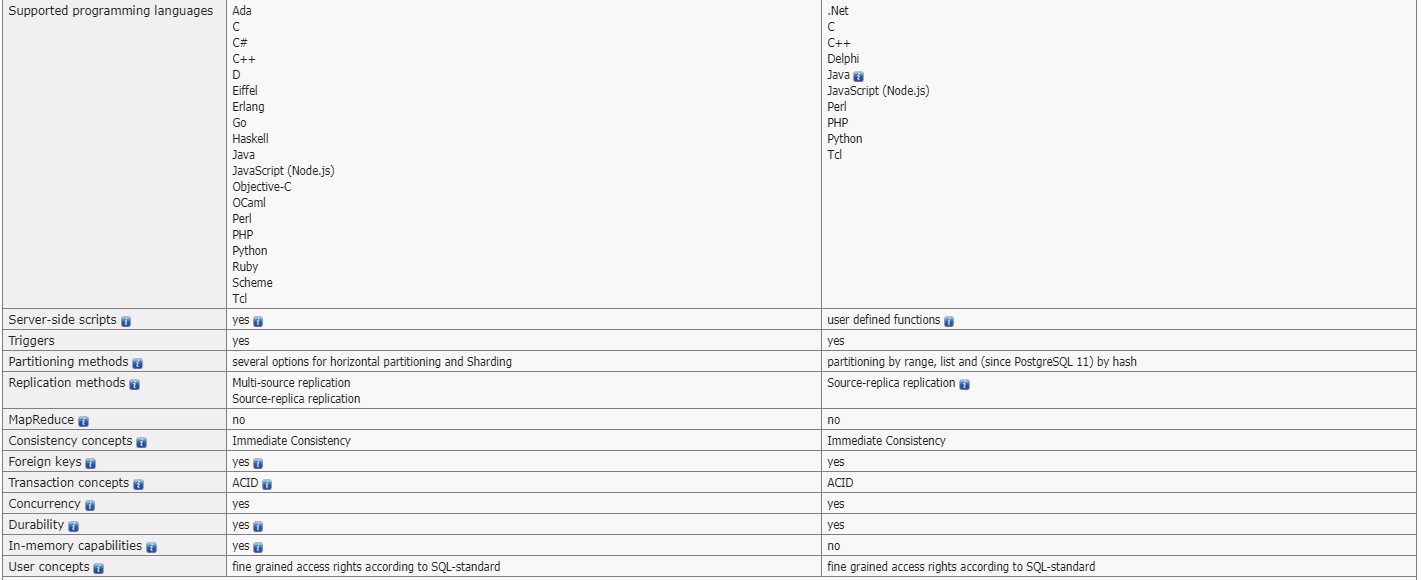
## Database Engine Ranking

And next we will do comparation database between Maria DB and another DBMS for pros and cons.

## Comparison Between MariaDB VS MongoDB

## Comparison Between MariaDB VS MYSQL

## Comparison Between MariaDB VS PosgreSQL

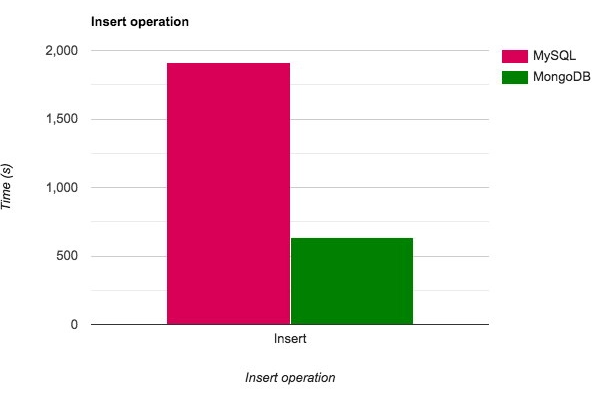


## **Result Comparison For SILVUE OI Requirements**

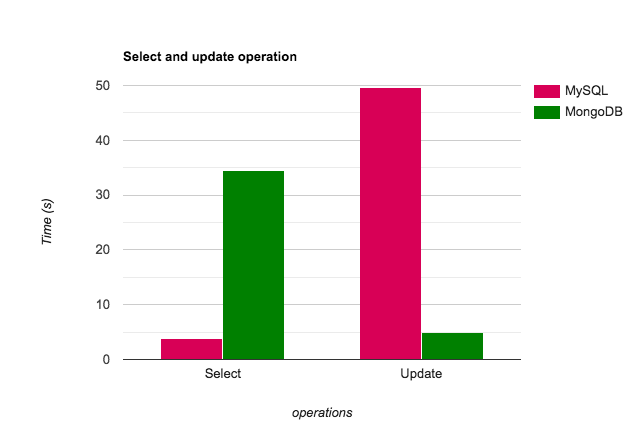
MariaDB Replication Features : MariaDB can do **multi source replication** like a **master to slave** and **source replication** like a client and server, compare with **PosgreSQL** just support **source replication with 3rd party.**

MariaDB and MongoDB can do transactional data using **JSON**. And **PostgreSQL,MYSQL** cannot do that.

# Comparasion Performance DBMS

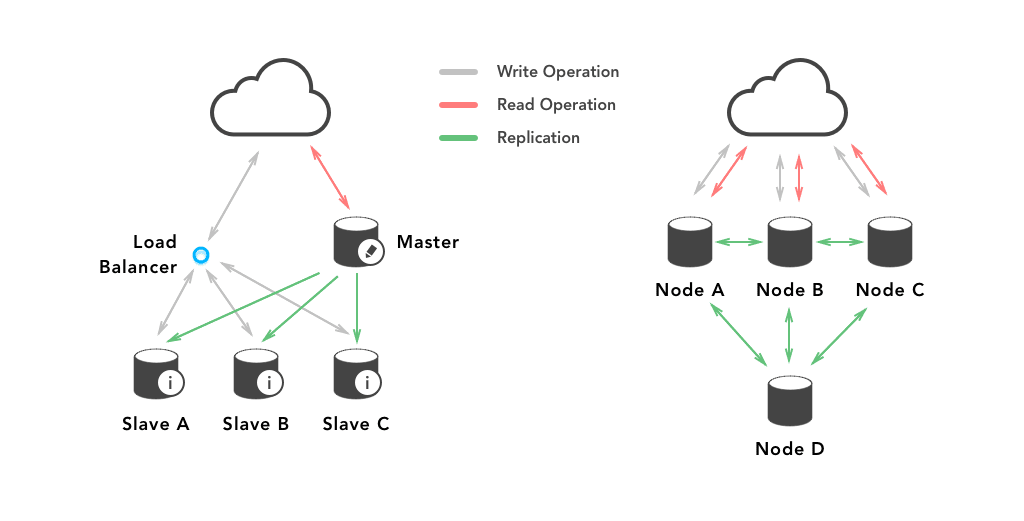
1. Insert Operation Query Per ms  
     
   What you’re looking at here is the amount of time it takes for MySQL (red) and MongoDB (green) to insert data inside the database. **The data at hand was calculated using a benchmark of one million inserts.**
   1. **Scalability.** Want to transfer your data across multiple servers? MongoDB encourages this behavior as it supplements even faster performance. On top of that, all data is saved in JSON format.
   2. **Mobile Development.** A new kind of tech-stack is called [MERN](https://click.linksynergy.com/deeplink?id=jf7w44yEft4&mid=39197&murl=https%3A%2F%2Fwww.udemy.com%2Fcourse%2Fmern-stack-front-to-back%2F) (MongoDB, Express, React, and Node), which is intended for the development of high-grade mobile apps. Since mobile applications tend to produce an enormous amount of data at a very fast pace, it only makes sense that MongoDB would be the default choice for this purpose.
   3. **Dynamic Structure.** Issues can arise without any foresight. In the case of MongoDB, you get extreme flexibility to dynamically move or transfer your database without needing to address any issues relating to the inner data structure.

## Select And Update Operation Query Per ms



In the case of updating which is a write operation, MongoDB takes 0.002 seconds to update all student emails whereas MySQL takes 0.2491s to execute the same task.

## High Availability and Cloud Computing Replication



**MongoDB:** This supports only **master-slave replication.** It uses replica sets to create multiple copies of the data. Each member of the replica set will be assigned primary or secondary role at any point in the process.

By default, read/writes are done on primary replica and then replicated on the secondary replicas.

**MySQL:** This supports both **master-slave replication and master-master replication**. Multi-source replication gives you the ability to replicate data from several masters in parallel. In master-slave replication, consistency is not too difficult as each piece of data has exactly one owing master.

While in master-master replication, if you can make it work, they seem to offer everything you want with no single point of failure.

## **Comparasion Security Model**

**MongoDB:** This uses a**role-based access control** with a flexible set of privileges. Its security features include authentication, auditing and authorization.

Moreover, it is also possible to use **Transport Layer Security TLS and Secure Sockets Layer SSL** for encryption purposes. This ensures that it is only accessible and readable by the intended client.

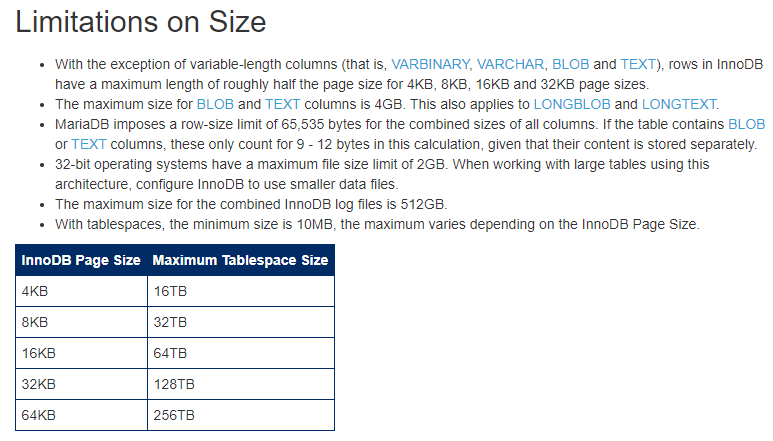
**MySQL:** This uses a **privilege based security model**. This means it authenticates a user and facilitates it with user privileges on a particular database such as CREATE, SELECT, INSERT, UPDATE and so on.

Though, it fails to explain why a given user is denied specific access. On the transport layer, it uses encrypted connections between clients and the server using the SSL.

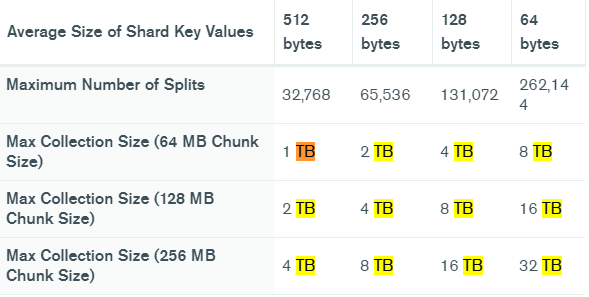
## Total Maximum Size DBMS

* 1. Mariadb/sql Maximum SizeInnoDB tables can have a maximum of 1,017 columns. This includes virtual generated columns.

InnoDB tables can have a maximum of 64 secondary indexes.

A multicolumn index on InnoDB can use a maximum of 16 columns. If you attempt to create a multicolumn index that uses more than 16 columns, MariaDB returns an Error 1070.

### Mongodb Maximum Size



# Database Backup And Restore Compressed with Encryption

1. Backup and Restore Encrypt With OpenSSL  
   The following example creates an AES-encrypted backup, protected with the password "mypass" and stores it in a file "backup.xb.enc":

**Command Encrypt** : mariabackup --user=root --backup --stream=xbstream | openssl enc -aes-256-cbc -k mypass > backup.xb.enc

**Command Decrypt** : openssl enc -d -aes-256-cbc -k mypass -in backup.xb.enc | mbstream -x

1. Compression gzip Backup and Restore Encrypting With OpenThis example adds a compression step before the encryption, otherwise looks almost identical to the previous example:

**Command Encrypt** : mariabackup --user=root --backup --stream=xbstream | gzip | openssl enc -aes-256-cbc -k mypass > backup.xb.gz.enc  
  
**Command Decrypt**: openssl enc -d -aes-256-cbc -k mypass -in backup.xb.gz.enc |gzip -d| mbstream -x

# Hardware Optimization

### Memory (RAM)

Memory is the most important factor as it allows you to adjust the [Server System Variables](https://docs.w3cub.com/mariadb/server-system-variables/index). More memory means larger key and table caches can be stored in memory so that disk access, an order of magnitude slower, is reduced.

Simply adding more memory may not result in drastic improvements if the server variables are not set to make use of the extra available memory.

Using more RAM slots on the motherboard increases the bus frequency, and there will be more latency between the RAM and the CPU. So, using the highest RAM size per slot is preferable.

## Disks (HDD/SSD)

Fast disk access is critical, as ultimately it's where the data resides. The key figure is the disk seek time, a measurement of how fast the physical disk can move to access the data, so choose disks with as low a seek time as possible.

You can also add dedicated disks for temporary files and transaction logs.

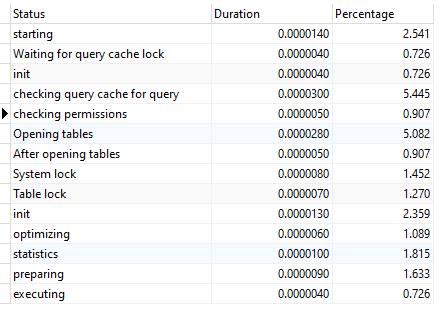
### CPU/Processor

Althrough hardware bottlenecks often fall elsewhere, faster processors allow calculations to be performed more quickly , and the results sent back to the client more quickly. Besides processor speed, the processor bus speed and cache size are also important factors to consider.

# Database Benchmarking

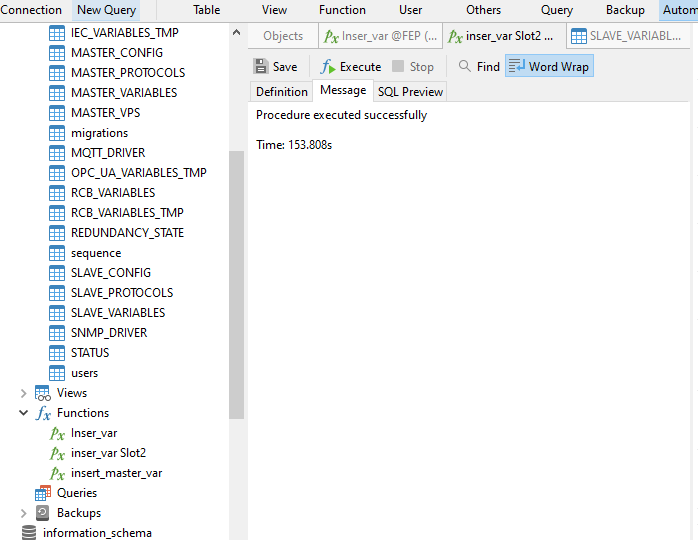
For this test we using ryzen 3700x on virtual box 8 core processor

## Select Query 10k Rows Mariadb Example Mariadb FEP Database

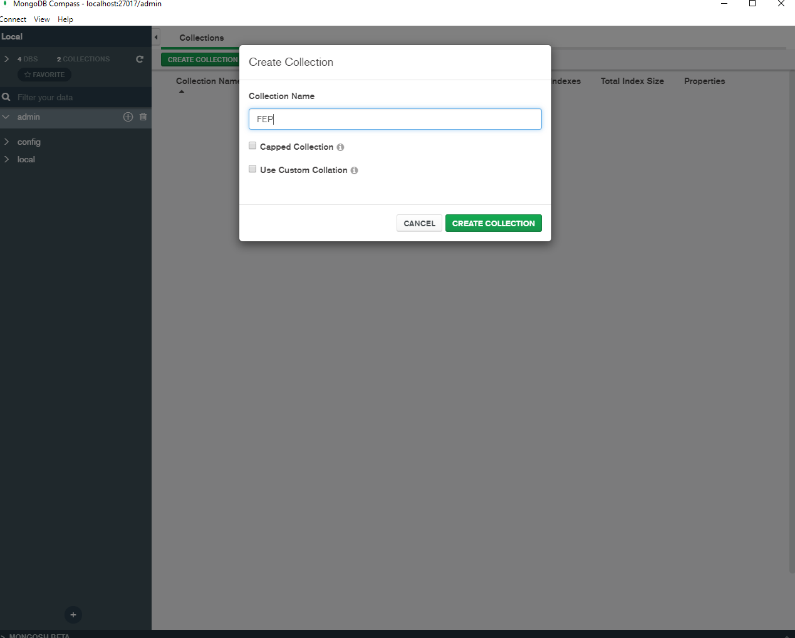
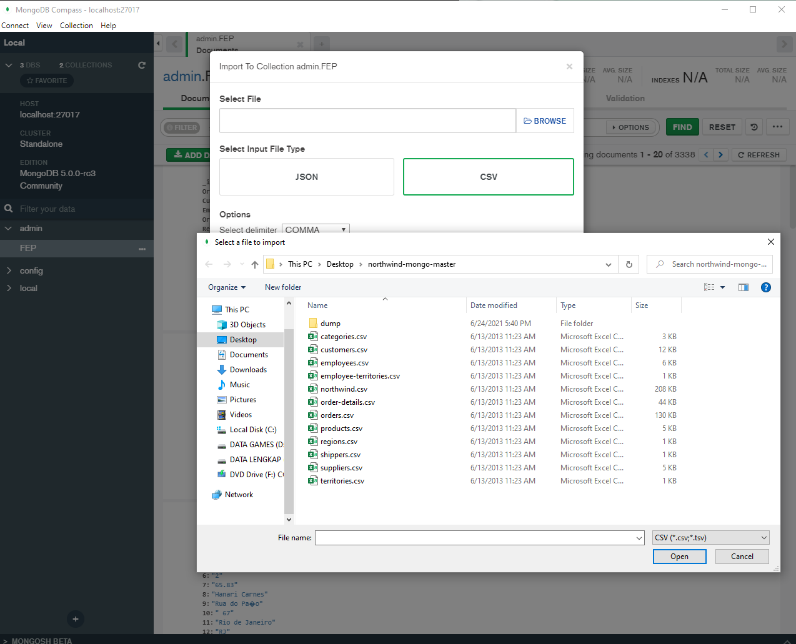
  
  


## Insert Query Example for 10 Rows Mariadb

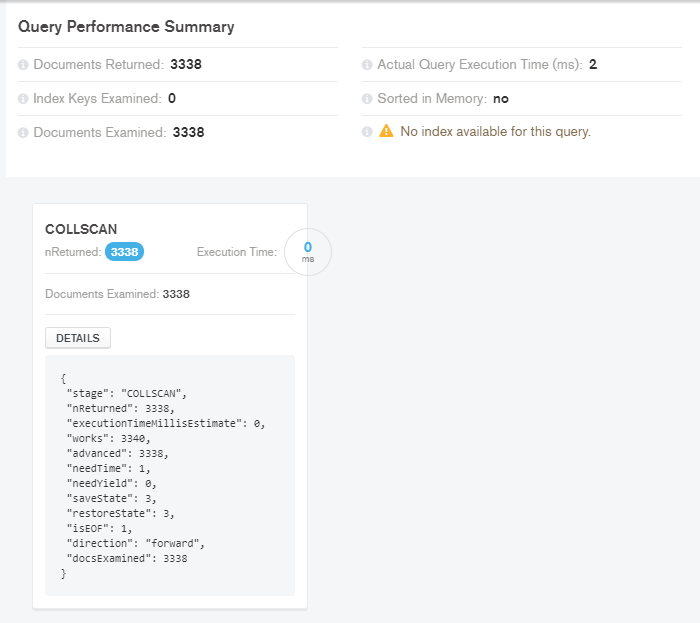
Example Mariadb FEP Database



## Insert Query Example for 3K Rows Mongodb

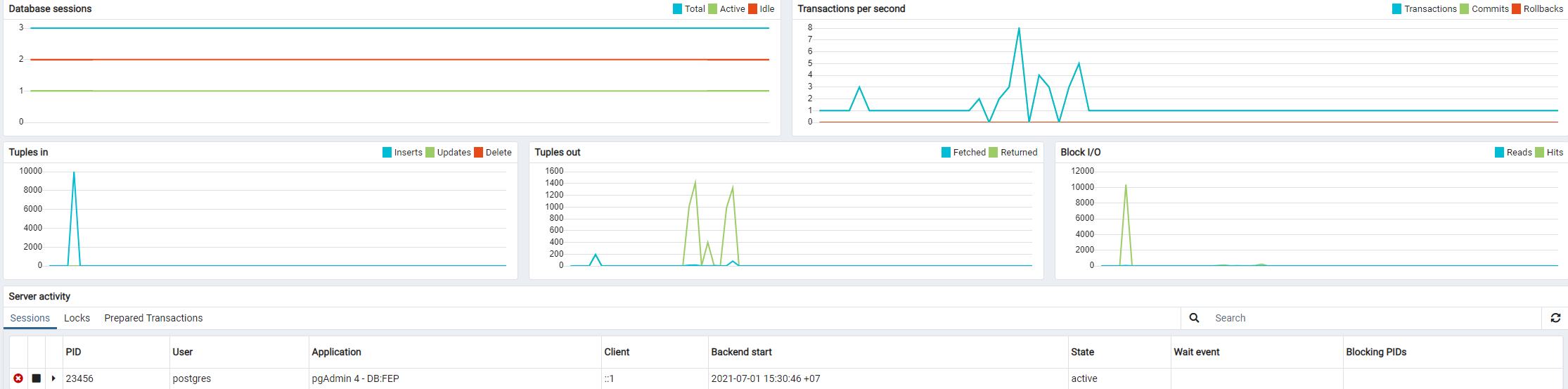
  
creating database  


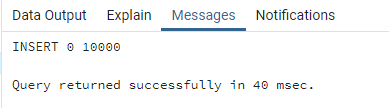
Import field/table example using CSV “database norhwind”



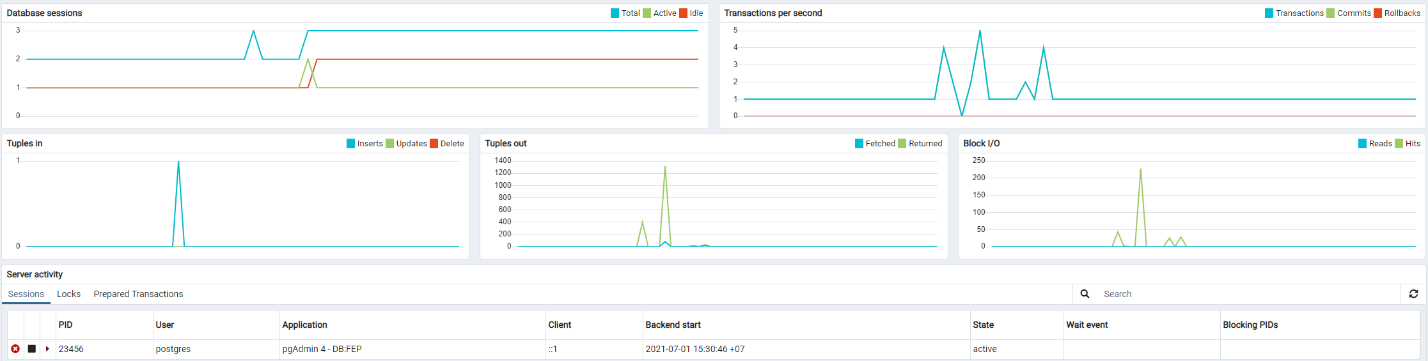
Result the query on 3k rows the execution time is 0 ms

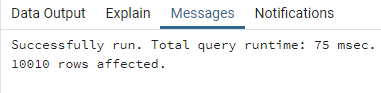
## Insert Query Example for 10k rows Postgreesql





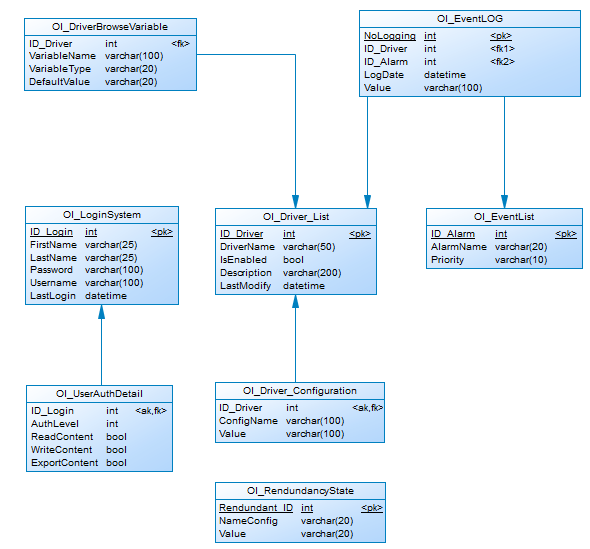
1. Select Query Example for 10k rows Postgreesql





# Database Module

## Database Design



## Function Structure Protocol

## Function Structure Slave Variable and Master Variable

## Function Redundant Settings

## Function Editor/General Settings

## Function Login with user Level

# Historian Module

## Realtime Graph Ex High Low Average with auto backup period.

## Archive Graph Ex High Low Average with Restore database from latest backup

# Log System Module

## Export And Import File To Ex: SQL , TXT ,CSV ,excel,Cloud Etc

## Event List and Log.

# Alarm Module

## Filter Event Alarm Status Ex: High Low Mid for each Category Function

# Redundancy Services Runtime Module

## Database Replication For Client Server **Dominant** And **Non Dominant**

## ***Keep Alive*** Replication Operating System.

# System Requirements Database Engine Module

## ***Belum kepikiran lagi cek teknis dlu***

References :

[DB-Engines Ranking - popularity ranking of database management systems](https://db-engines.com/en/ranking)

[MongoDB vs MySQL NoSQL - Why Mongo is Better | Severalnines](https://severalnines.com/database-blog/mongodb-vs-mysql-nosql-why-mongo-better)  
[InnoDB Limitations - MariaDB Knowledge Base](https://mariadb.com/kb/en/innodb-limitations/)

[MongoDB Limits and Thresholds — MongoDB Manual](https://docs.mongodb.com/manual/reference/limits/#Database-Size)  
  
[Using Encryption and Compression Tools With Mariabackup - MariaDB Knowledge Base](https://mariadb.com/kb/en/using-encryption-and-compression-tools-with-mariabackup/#encrypting-and-decrypting-backup-with-openssl)

[Mariadb-check - MariaDB - W3cubDocs](https://docs.w3cub.com/mariadb/mariadb-check/index)

[Hardware Optimization - MariaDB Knowledge Base](https://mariadb.com/kb/en/hardware-optimization/)