3/22/2021

3/22/2021

3/22/2021

DATABASE SYSTEM

CSC 472

MySQL\_in\_actionDATABASE SYSTEM

CSC 472

MySQL\_in\_actionDATABASE SYSTEM

CSC 472

MySQL\_in\_actionDATABASE SYSTEM

CSC 472

Table of Contents

[TABLE OF FIGURES 2](#_Toc67347808)

[MySQL 4](#_Toc67347809)

[Oracle Database 6](#_Toc67347810)

[MongoDB 8](#_Toc67347811)

[PostgreSQL 11](#_Toc67347812)

[Apache Cassandra DBMS 14](#_Toc67347813)

[References 17](#_Toc67347814)

# TABLE OF FIGURES

*[Figure 1 MySQL\_in\_action](../../../../../../../../C:/Users/USER/Desktop/CSC_ASS/CSC_472_Adefemi_Micheal_Afuwape_(206323).docx" \l "_Toc67347799)* 5

*[Figure 2 MongoDB\_in\_action](../../../../../../../../C:/Users/USER/Desktop/CSC_ASS/CSC_472_Adefemi_Micheal_Afuwape_(206323).docx" \l "_Toc67347800)* 10

[*Figure 3 PostgreSQL\_in\_action* 13](#_Toc67347801)

**ASSIGNMENT**

COURSE CODE: **CSC472**

COURSE TITLE: **DATABASE SYSTEMS**

**Question**

Write a brief note on 5 popular DBMSs and clearly and concisely discuss the strength and weaknesses of these DBMSs.

**Answer**

1. Five popular DBMS
   1. MySQL
   2. Oracle Database Management
   3. MongoDB
   4. PostgreSQL
   5. Apache Cassandra

# MySQL

MySQL is an open-source database that is under the Relational Database Management System (Relational Database are digital database based on the relational model of data). The “My” in the abbreviation is the name of one of the co-founders (Michael Widenius) daughter while the “SQL” is an abbreviation for Structured Query Language (which is a domain-specific language used in program design and intended for handling data detained in RDBMS, stream processes in RDBMS and lastly for data structuring) (Corporation, 2020). The application is free to download since it is open-source, even though MySQL is a stand-alone clients application that permit users to relate directly with MySQL DB using SQL, the DBMS is more often used with other programs to implement applications that need relational database compatibility. (GmbH, 2020)

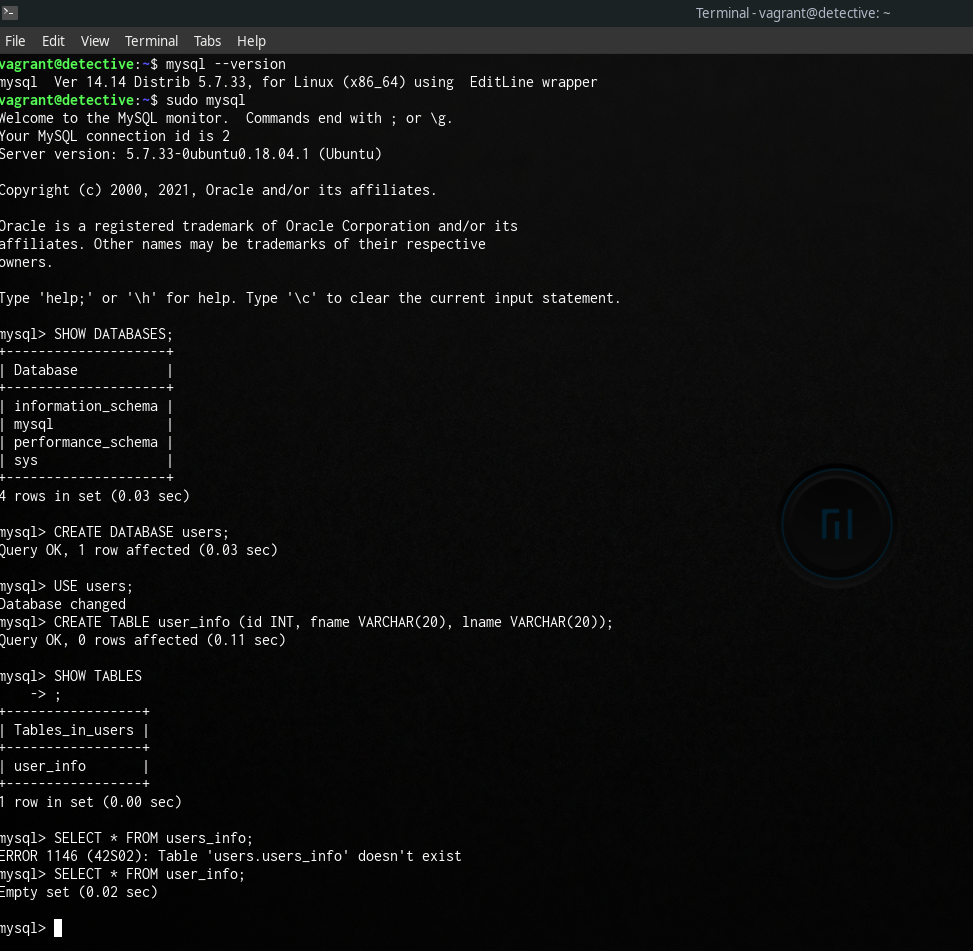
It’s of note that the M in the stacks above could also mean MariaDB (Which is forked of MySQL). There are a lot of DB driven web application that uses MySQL especially Content Management System applications such as WordPress, Drupal, Joomla and even Dreamweaver (while other DBs can be integrated with Dreamweaver, the default is MySQL).

**Some strengths of using MySQL** are included below:

1. The first strength of MySQL is that the application can run effectively on minimal compute resources (Memory, Disk Space) unlike other DBMS. Although the higher the resources the better the performance, the full bundle of MySQL installer (community version) is 300MB, and of cos, there are minimal installations that go as much as 100MB.
2. Another great strength of MySQL is scalability (i.e. the ability to change in size). Because MySQL can be embedded in most application running on the web, mobile and even embedded systems, flexibility is therefore demanded and MySQL provides superb flexibility to manage and coordinate any sort of applications. If your scalability needs ever exceed the capabilities of a single server, you can use the internal replication capabilities of MySQL to create a cluster of systems and distribute the load by directing the writes to the master host and sending the reads to the slaves.
3. Among other strength, MySQL possess are the ability to run on a variety of architecture and operating systems. MySQL runs with ease on Windows, Linux and even on macOS.
4. MySQL provides transactional support because the DBMS offers full data integrity. It also supports unrestricted row-level locking, isolated, consistent, durable, completely atomic transaction and multi-version transaction support.
5. Lastly, MySQL is open source which means it’s constantly updated and bug fixes are rolled out as quickly as possible. Although there is the enterprise version which is not open source but the community version code is hosted on Github and is constantly updated.

Well, there’s no strength without weakness and outlined are **some of the weaknesses of MySQL**:

1. One of the issues I have faced using MySQL is that the functionality is heavily dependent on addons which make it easily bloated. Certain features like text search are not dependent on the core engine but other applications or add-ons, unlike other DBMS that supports out of the box functionality.
2. Another weakness of MySQL is that it currently does not support sub-queries, views, stored procedures, triggers, and foreign key enforcement. Imagine one database application having these features, definitely one will have to rewrite those portions before moving to MySQL and this is a daunting task in some cases.



*Figure 1 MySQL\_in\_action*

# **Oracle Database**

Oracle DB Management commonly referred to as Oracle DBMS or Oracle, is a multi-model DBMS that is fashioned and promoted by Oracle Corps. Oracle Database is known as a relational database that is widely used for successively online transaction processing, data warehousing and even diverse database jobs (Biggs, 1998). This type of DBMS is provided for use on-premise, in the cloud, as well as hybrid cloud setting up and can also be tried on third-party servers as well as on Oracle’s own hardware. Oracle Database version 6 was the first version to introduce **PL/SQL** language which is a proprietary procedural extension to SQL and allows storing of program units. The DBMS came built-in with an entity called **PUBLIC** which is neither a Role nor a User even though it is managed like both. In Oracle Database every user in the DB is a member of PUBLIC by default. Version 10g of Oracle Database introduced a cool feature called **SQL Model Clause** which complements calculation competence into SQL, this feature gives power of a conventional spreadsheet. Also, in version 10g Oracle DB supports text exploration and replace functions were introduced. This feature uses regular expressions and suggestively improve the ability to query and process, character data. This new feature of regular expressions extends the capability to outline the instructions for the sorts of strings one can search and it was also used as functions for manipulating character data.

The latest release of Oracle DB which is the 21c brings over 200 new capabilities to the converged enterprise database which allows developers to use one database for different workloads such as relational data, JSN, spatial, graph, blockchain and a lot more. The Oracle Database 21c, adds exciting new capabilities for developers and analysts including immutable blockchain tables, native JSON data types, AutoML for in-database machine learning, persistent memory support, higher-performance graph models, database in-memory, and an enhancement to the highly scalable sharding feature (Exadata, 2019).

There’s also the Oracle Autonomous Database which is an all-in-one cloud database solution for data marts, data lakes, operational reporting, and batch data processing. Oracle practices the use of machine learning to totally systematize all repetitive database task, ensuring higher performance, dependability, safety, and operational productivity.

Some **strengths of Oracle Database** are outlined below:

1. One lovable strength of Oracle DB is it’s out of the box security and compliance without the need for much configuration.
2. Like MySQL, Oracle DB also supports the ability to run on variety of architecture and operating systems. And unlike the Microsoft SQL Server (which is a competitor) that only runs on the Window OS, Oracle DB works well on Linux, macOS and even on Windows.
3. When it comes to anything that has to do with data, back up and recovery is a key and for Oracle DB, one of its strength lies in this. With the aid of using Oracle DB one can easily be able to make a point-in-time recovery if by chance there’s data loss or if there was an error while trying to restore the database it does what is known as flashback, that is it saves the work done on the database before the error occurred.
4. There are instances when more than one server or instances needs to be combined and connected to a single database this is known as DATABASE CLUSTERING, Oracle DDB has a strength for clustering servers which allows Oracle DB to perform data management using more than one server.
5. Even though there are numerous strengths of Oracle DB, the last one I will be pointing out is portability, data in the Oracle DB is easily moved/ported from onsite to cloud with the help of data pumps

Oracle DB even though its big in the DBMS market, it also has its weakness and some of **Oracle DB weakness** are outlined here:

1. One of the greatest weakness of Oracle DB is cost. Compared to other competitors Oracle DB, this of cos is not suitable for startups except when they need high DBMS specifications, else this becomes a waste of resources. Another type of cost is getting a specialist in Oracle DB, although there are a lot of Certified Oracle DB Administrators, finding a specialist is costly and time-consuming.
2. Another great issue with Oracle DB is that it is complex to use for beginners especially for someone that lacks technical ability which is therefore frowned upon in the beginner’s field for Database Admin. Yes, installation is quite easy but maintaining it is complex.

# **MongoDB**

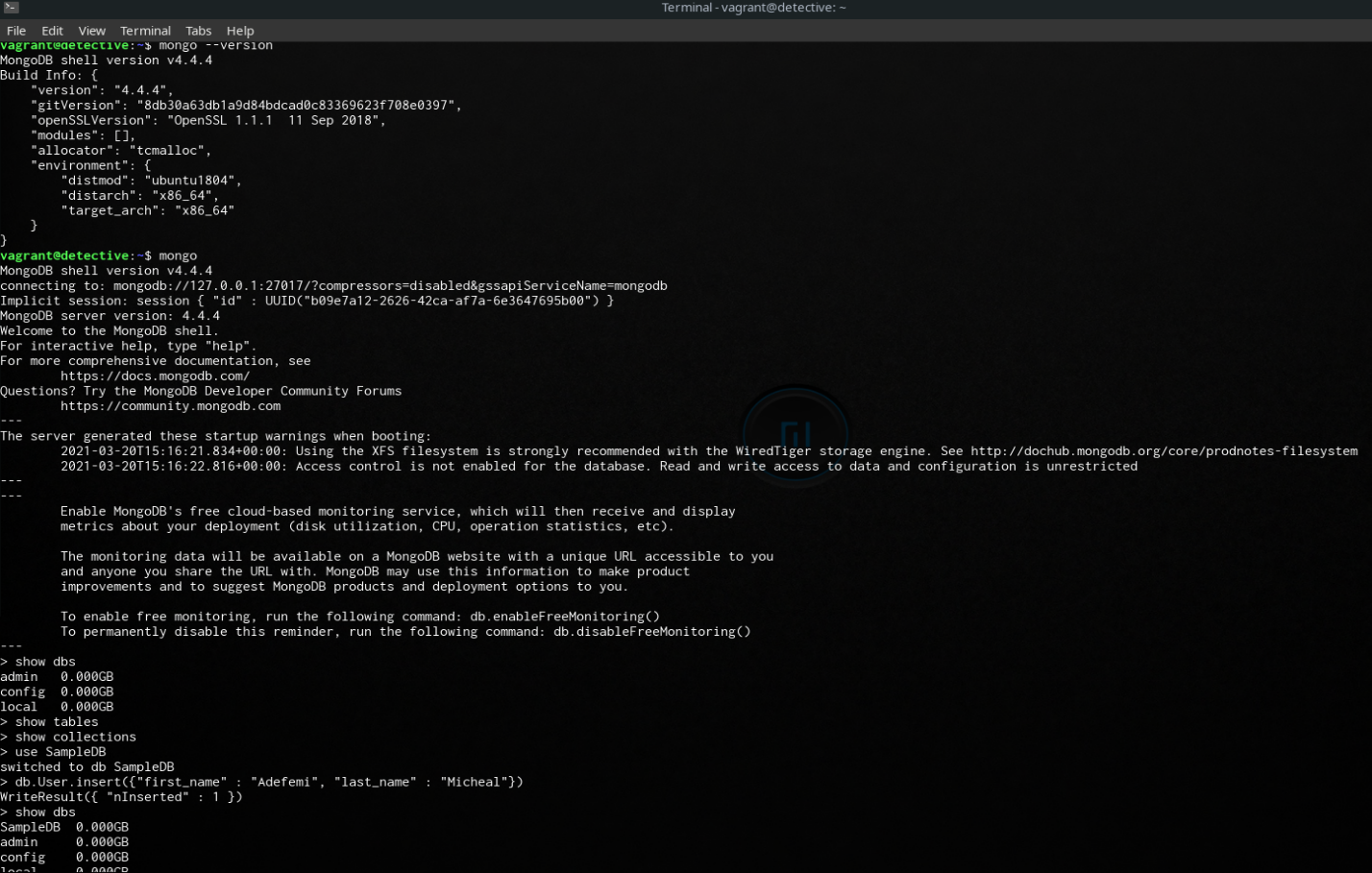
MongoDB (Mongo DataBase) popularly known as been the DBMS for contemporary applications. It is a general-purpose, document-based, circulated database made for recent application; developers also cloud development. MongoDB is open source and its available across all platform. MongoDB is classified under the NoSQL database program since it’s a non-relational database. Even though it's open-source (**MongoDB Community Server**) there’s also the enterprise edition called **MongoDB Enterprise Server** and then the one designed for cloud development called **MongoDB Atlas**. MongoDB stores data in JSON-like documents which is why it’s referred to like been a Document Database. MongoDB offers both GUI which makes it easy to search, visualize and work with data. It also supports CLI installation for the lovers of the command line. The application is written in C++ and is developed and owned by MongoDB Inc. The difference between MongoDB and old-style relational databases is that collections and documents are used in MongoDB. In MongoDB, a key-value pair is what makes a **document** and that’s the basic unit. Function and documents are what makes up **Collections,** and this is same as tables in relational database. Each database in the DBMS contains collections and each collection encompasses documents and individual document can have a diverse fluctuating amount of fields and the content and size of individual document can be dissimilar from each other. The structure of document in MongoDB is more related to how one construct classes and objects in OOP. Also, the documents are not necessarily schema defined during creation but rather this can be created on the fly unlike in a relational database where a table schema is compulsory before data insertion. Also, in the DBMS hierarchical relationships are allowed and more complex structure and array storage are allowed easily. Even though data integrity is enforced in relational database, in MongoDB it is explicitly required. In a relational database, data are required to be normalized (normalization refers to a technique that reduces data redundancy and ensures data is stored logically), this averts orphan records and copies. But when data is been normalized, it necessitates extra tables, this then allows added table joins, which result to additional keys and indexes. Now as the database start to grow, execution can be a problem and this is not an obvious requirement in the DBMS since it is resilient and normalization of data is not needed at first. Some of the features of MongoDB are Ad-hoc queries, indexing, replication, aggregation, load balancing, file storage (with the help of a file system known as GridFS), capped collections, transactions (supports multi-document ACID transactions).

Some **strengths of MongoDB** are outlined below:

1. Scalability is such a huge deal in the database world and MongoDB offers the ability to scale out the reads and write throughputs using a sharded cluster. Auto-balancing and auto sharding is supported in the sharded cluster this makes scaling of the application horizontally possible, and it does this by accumulating thousands of nodes which then allows it respond to mass operations per second over a very large data set.
2. One strength of MongoDB that stands out is how easy it is to set up the environment and its provision of JavaScript client for queries. Given that they have an interactive GUI application, most developers tend to use MongoDB when they need to decide which NoSQL database to use.
3. Every database developer knows how hard it is to work with **join** in a relational database, MongoDB is designed in such a way that you don’t need complex join and this has to do with flexibility where conversion or mapping of application object to the database object is not needed unlike in relational database. Mapping of this objects are easy since they are more natural BSON documents. The schema-less feature in the DBMS makes it developer-friendly since the data structure can be changed on the fly by introducing any kind of fields at any time.
4. MongoDB also has the ability for deep queries capabilities such as full-text search, regular expressions, dynamic queries on documents, aggregation (using map-reduce or the aggregation pipeline framework), and sorting. The aggregation and the queries are executed at the collection-level which might be an issue if one want to run global queries across the entire database since joins aren't supported.

MongoDB might not be the future for all database in that it also has some **weakness** which are outlined below:

1. Most DB Admins frown at MongoDB when it comes to working with relational data since join is not supported. If the data does not fit into the document data model, then either the database schema will need to be normalized as it’s done in relational databases and this will result in several follow up queries that decrease the output or add code complexity. Therefore, it is not always recommended to use MongoDB for a use case where one needs to model relevant data and then run complex join-intensive queries.
2. Another weakness of MongoDB is that the free version does not support more than 16MB document size and the document cannot perform nesting more than 100 levels.



*Figure 2 MongoDB\_in\_action*

# **PostgreSQL**

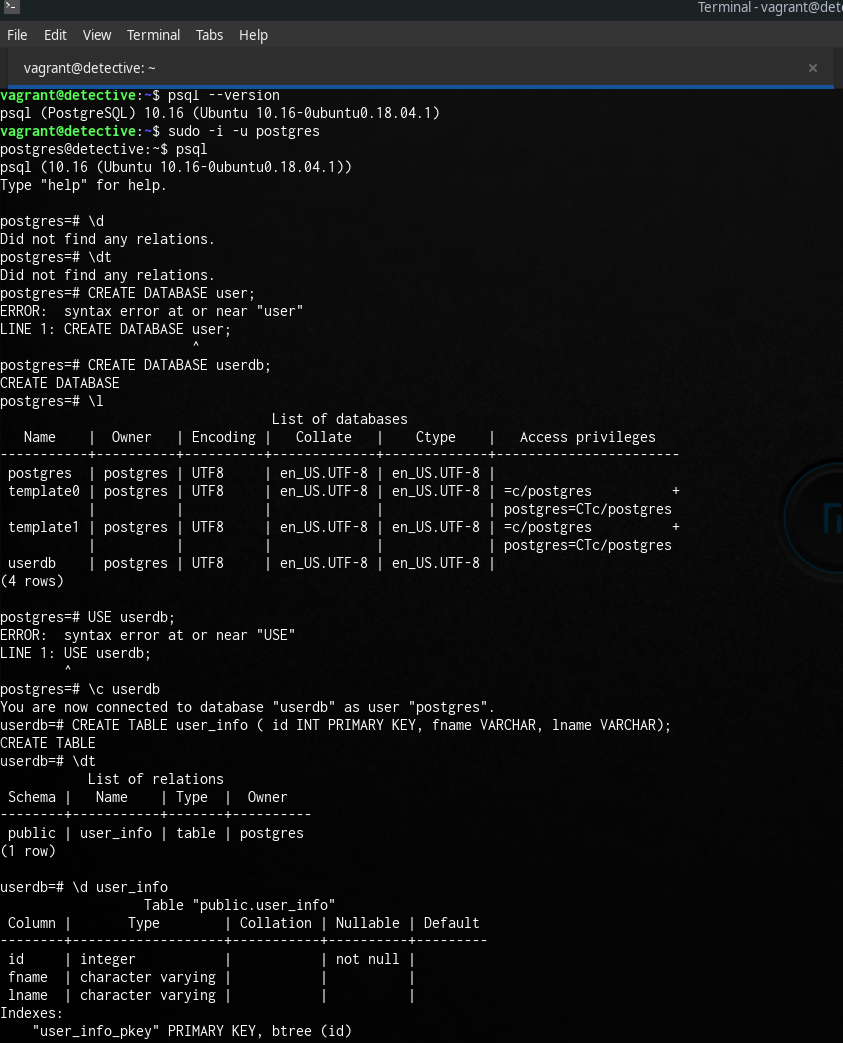
PostgreSQL, also known as Postgres, is the most sophisticated open-source relational database in the globe. Unlike other database listed above, SQL (relational) and JSON (non-relational) querying are also supported by Postgres. Like other DBMS Postgres also supports multi-platform that is it can run on Windows OS, Linux OS and macOS. PostgreSQL was initially named POSTGRES, which alluded towards its beginnings as a replacement for the INGRES database. It was renamed PostgreSQL in 1996 to show its SQL support. Transactions with ACID (Atomicity, Consistency, Isolation, and Durability) properties, triggers, materialized views, stored procedures, and automatically updatable views are all features of Postgres (Stonebraker). Postgres is macOS' default database, and it's designed to manage a wide variety of workloads, from single machines to data centers or web servers with a large number of concurrent users. Postgres handles concurrency with MVCC (MultiVersion Concurrency Control), which allows each transaction to take a snapshot of the database and make improvements without impacting other transactions. Since Postgres server supports concurrency, it is process-based (not thread-based), which means that each database session uses one OS process, and multiple sessions are automatically shared through all available CPUs by the OS. Apart from concurrency support, Postgres also provides three degrees of transaction isolation: Read committed, Repeatable Read, and Serializable (via the serializable snapshot isolation). Postgres has a web version (phpPgAdmin), GUI version (pgAdmin) and also the CLI version (psql) and also the cloud version (PostgreSQL Studio), which mean developers or database admin are free to choose between this range of support for use. When it comes to a stack that uses Postgres, we have the LAPP (Linux, Apache, PostgreSQL, PHP/Perl/Python) stack which means Postgres is widely used as a robust back-end database (Kerstiens, 2013). This is useful for workloads that don't need such assurances, but it's not ideal for all data because it slows down efficiency by requiring confirmation of the transaction's arrival at synchronous standby.

Some **strengths of Postgres** are outlined below:

1. One major strength of Postgres is its support across all programming language and how flexible it is to use this language. No programming language does not have a driver for Postgres. The community is so large that it builds a driver for all popular programming language.
2. Postgres like all other non-relational DBMS also support flexible full-text search, regular expressions, allows replication, supports ACID, dynamic queries, with richer semantics, geospatial types, complex numbers and so on, even though it’s still SQL compliant, unlike MySQL. It uses a variety of indexing methods for full-text searches, including B-trees, Genetic Algorithms-based indexes, and GIN indexes.
3. Postgres is also well-liked in the BI (Business Intelligence) community, as it integrates well with BI applications and is better suited for data warehousing and data analysis due to its quick read-write speed, which these applications require. And the write-head logging in Postgres makes it a highly fault-tolerant database.
4. The greatest power/strength of Postgres lies in the fact that it is easy to use even though it SQL like but it’s very easy to pick for beginners either using the GUI or the CLI so learning Postgres does not require much training.

Although Postgres is considered the world most advanced open-source relational database, it still has some **weakness** which is outlined below:

1. Postgres has a very large focus on compatibility rather than speed, unlike MySQL that has high reading speed. Even though over the years its performance has always been worked on but more is still required form Postgres.
2. This might not be a major weakness but Postgres docs are only English written, it’s hard for non-English speakers to pick up the DBMS.
3. The Postgres replication method for holding slaves up to date with the primary, according to anecdotal evidence, is a little buggy. Since this does not happen for all, but only in certain usage cases, anecdotal evidence is used.



*Figure 3 PostgreSQL\_in\_action*

# **Apache Cassandra DBMS**

Cassandra, also known as Apache Cassandra, is an open-source, distributed NoSQL database developed by Facebook and published as an open-source project in July 2008. Cassandra provides modern applications with the continuous availability (zero downtime), high efficiency, and linear scalability they demand, as well as operational flexibility and easy replication through data centers and geographies. Cassandra can accommodate petabytes of data and thousands of simultaneous operations per second, allowing businesses to process vast volumes of data through hybrid and multi-cloud environments. Avinash Lakshman (one of the writers of Amazon's Dynamo) and Prashant Malik, two Facebook engineers, created Cassandra to power Facebook's inbox search feature, which made conversation search and other content easier for users. Cassandra's design merged Amazon's Dynamo paper's distribution model for horizontal scaling across multiple nodes with Google's BigTable paper's log-structured storage engine. As a result, a highly scalable database was developed that could accommodate even the most data-intensive and performance-demanding use cases. The decentralized system of the database ring architecture is peer-to-peer in which it has no single point of disaster. It has a masterless architecture and an equal node. This distributed nature of its architecture allows for better scalability, high availability and operations ease. Additionally, the database makes provision for a language related to SQL known as CQL and can be used effortlessly for modelling and retrieving of stored data. (Al-Saeedi, 2016)

1. **Node**: This is where data is stored.
2. **Data Center**: This is a collection of associated nodes. Diverse data centres are frequently used to serve dissimilar workloads to avert transactions from being obstructed by other disparate transactions. The replication conformation in the database can be set on the data centre level.
3. **Cluster**: This encompasses numerous data centres that comprise multiple nodes or machines
4. **Peer-to-Peer Gossip Protocol**: This can be used by each node to determine the current state of a node and other nodes in the cluster. Since the pass down of information is moved from one node to another in a chatter manner, this makes a node knows about another node in the cluster. The chatter details are reserved and persevered.
5. **Partitioner**: There is automatic sharding and circulation across node when one writes data to any node in the cluster. The database use partitioner to regulate how to share this data across node. Steady hashing is done in the database to improve performance since the partitioners are hash function that are used to compute tokens and these tokens are the partition key.
6. **Snitch**: This is used to listen and monitor nodes, network performance and machines in the cluster which helps to deliver details of the node state and make a decision on the best node it can use for data replication. The default is dynamic snitch and this monitors the performance of diverse nodes an example is the read latency.
7. **CQL**: This is the language the database uses for communication, its similar to SQL in format and it also has a CLI known as cqlsh that one uses interrelate with CQL. Datastax which is a startup developed a GUI tool for the language known as “Datastax DevCenter”.
8. **Keyspace**: This is a container that stores numerous CQL tables and can be compare to database schema in relational database. A keyspace per application is the standard even though numerous keyspace are welcomed.
9. **CQL** **Table**: Storage of data in this database are in columns and are made available by a primary row key. When a numerous well-organized column is gathered and have a primary row key then we have CQL Table.

Enough said about Cassandra’s infrastructure, some **strengths of Cassandra** are outlined below:

1. Since data is automatically replicated to multiple nodes, Cassandra is said to be fault-tolerant. Replication occurs across numerous data centres and unsuccessful ones can be substituted with no interruption. If by chance one has an abortive node, the users are rerouted automatically to a healthy node without noticing the node failure. What this mean is that application is designed to always be reachable with data been accessible. The database also has an option for built-in repair service that fix an instant occurrence of a problem with no manual interference. Therefore, at the event of a failed node this does not affect efficiency (DataStax, 2021).
2. Among other popular NoSQL database, Cassandra consistently outperforms them in benchmarks and real application and this is because of the fundamental architecture of the DBMS. It also outperforms other DBMS in throughput and also in latency. The database is also designed for heavy writes, with any insert or update being written immediately without the need to lock or read existing data for pattern in constraints violations, allowing for extremely quick writes. Upserts are a form of update that is very fast since new data is written with a different timestamp.
3. Decentralization and elasticity are the core of Cassandra, there are no sole facts of disaster. There is no condition for system failbacks. All node in the cluster is indistinguishable. In addition, as new devices are added, the read and write volume both increase linearly, with no interruption or disturbance to the applications.
4. Lastly one of Cassandra’s strength is consistency, the database is organized to have either strong consistency “or” eventual consistency, because it allows what is known as **TUNABLE CONSISTENCY**. The database is typically constructed to be eventual consistency. If the database is organized to be strongly consistent, one might influence the system availability based on “CAP Theorem”. The amount of replicas that should be admitted for the write before it is measured as positive determines the consistency level (Al-Saeedi, 2016).

Apache Cassandra also comes with some **weakness** which I outlined below:

1. Cassandra's first flaw is related to storage, since data is only circulated using the partition key, which means that if you select the wrong partition keys, you risk having flashpoint problems, where one node receives the majority of the weight since each partition can only be stored in one node computer. Furthermore, each partition's maximum number of cells is limited to 2 billion; additionally, individual column values are limited to 2GB, and collection values are limited to 64KB.

1. The DBMS takes some feebleness when data collection arises, since collection is done on the “partition level”, if one wants to track collective queries on the DBMS then it is a necessity to stipulate the partition key because the collection occur on partition level. Typically, collection is convenient if it can be tested on the entire database. And to do this, one might primarily need to acquire the partition keys, and run the combined query in contradiction with each partition distinctly and then collectively.

# References

Al-Saeedi, B. (2016, April). *Factors-Influencing-NoSQL-Adoption: alronz.github.io*. Retrieved from alronz.github.io: http://alronz.github.io/Factors-Influencing-NoSQL-Adoption/site/Cassandra/Basics/Underline%20Structure/

Biggs, M. (1998, October 5). *Books*. Retrieved from Oracle8 on Linux shows promise: https://books.google.com/books?id=sFEEAAAAMBAJ&pg=PA129

Corporation, M. 8. (2020, April 3). *History of MySQL*. Retrieved from MySQL is named after co-founder Monty Widenius's daughter, My.: MySQL 8.0 Reference Manual. Oracle Corporation

DataStax. (2021). *Home: DataStax*. Retrieved from DataStax: https://www.datastax.com/cassandra

Exadata, O. D. (2019, February 13). *Blogs: Oracle*. Retrieved from Oracle: https://blogs.oracle.com/database/oracle-database-19c-available-exadata

GmbH, D.-E. s. (2020, April 3). Retrieved from https://db-engines.com/en/ranking/relational+dbms

Kerstiens, C. (2013, September 26). *WAL-E and Continuous Protection with Heroku Postgres*. Retrieved from Heroku Blog: https://blog.heroku.com/wal\_e\_and\_continuous\_protection\_with\_heroku\_postgres

Stonebraker, M. (n.d.). Proceedings of the 13th International Conference on Very Large Data Bases. In M. Stonebraker, & L. Rowe, *The POSTGRES data model* (pp. 83–96). Brighton, England: Morgan Kaufmann Publishers.