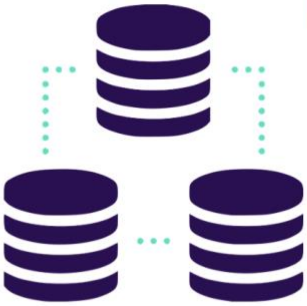
The Database of Databases



Created by the

Advanced Databases Students – TU059

Semester 1 – 2022-2023

Table of Contents

1. TimeScale DB …
2. Cassandra Database …
3. YugaByte Database …
4. Graph Data in Neo4j …
5. Oracle TimesTen …
6. Snowflake …
7. MySQL Heatwave
8. MariaDB …
9. Amazon Athena …

TimeScale DB …

Database	**TimescaleDB**		
Licence Type(s)	TimescaleDB Open Source is made available under the Apache 2.0 License [1]		
Original Developer(s)	Ajay Kulkarni, Mike Freedman	Date/Year Created	Initial release 2017. Version 1.0 in September 2018 [1]
Developed in Country	The United States [1]		
Current Owner	Timescale Inc [1]	Owner Since	Since release [1]
Owner Country	The United States [1]		
Previous Names (& Year)	NA		
Websites (Company & Product)	https://www.timescale.com/ TimescaleDB		
Database Category/Type	Time series /Temporal [1]		
Licence Types	TimescaleDB with an Apache 2 TimescaleDB Community Edition [15]		
Derivative Systems	What other databases are based on this database? What database is this database derived from? TimescaleDB is a time-series database, built on top of PostgreSQL. [1]		
Database Description	TimescaleDB is an open-source relational database for time-series/ temporal data. [1]		
History	Timescale first released TimescaleDB in 2017. The current release is version 2.6.0, which was released in February 2022. [1]		
Written in what language	Written in C [1]		
Supported Languages	.Net C C++ Delphi Java Javascript Perl PHP Python R		

	Ruby Scheme Tcl [1]
Operating Systems	Linux, OS X, Windows [1]
Support Platforms	(cloud, on-premises, Hybrid, etc) Cloud through Timescale Cloud or Managed Service with vendor Aiven. Self-managed TimescaleDB [15]
Isolation Levels	What isolation levels are support by the database. Since TimescaleDB is a postgreSQL extension, the postgreSQL isolation levels are supported. You can implement all four of the standard isolation levels (read uncommitted, read committed, repeatable read, serializable). Internally, however, three unique isolation levels are present. The read uncommitted isolation level operates like the read committed isolation level. [2] [8] [13]
Concurrency Control	What concurrency control is support by the database The concurrency control that is implemented and supported in PostgreSQL is what is supported in timescaleDB. PostgreSQL uses Multiversion Concurrency Control (MVCC). Practically this means that every SQL statement works with a database snapchat, no matter the current state of the database. This way inconsistent data from concurrent updates do not interfere with SQL statement and transaction isolation is provided. MVCC aims to minimise lock contention and allow for acceptable performance for multiuser environments. Additionally, two phase locking with deadlock detection is supported. Two phase locking is when (un)locking is done in two phases: growing and shrinking phase. Growing phase means that new locks can be created but none can be released, while shrinking phase means existing locks are released but no new ones are created. [12] [13]
Persistence Model(s)	What persistence models are supported by the database TimescaleDB follows PostgreSQL's persistence model. [2]
Data Models	What data models or combinations of models are supported Both wide-table and narrow table models are supported. With the wide-table model there are many columns where the columns have a distinct data variable. While the narrow-table model has a column containing the data variable name and the other containing its value. [6] [7] [13]
Language Capabilities	That is the main language used to interact with the database. How extensive is the implementation of the language? What language standards are supported. What things does it do differently? SQL Since it is a postgresql extension, it implements the same level SQL. Postgresql supports the major features of SQL:2016. Out of 177 mandatory features required for full Core conformance, it conforms to 170 features. [2]
Stored Procedures/Functions	Does the database support stored procedures/functions? Can external procedures be created/used? Can procedures/functions be created using other languages (C, Java, Python, etc) Yes, PostgreSQL functions/stored procedures can be implemented. Stored procedures can be created in PL/pgSQL, PL/ Tcl, PL/ Perl, PL/ Python, PL/ Java, PL/ PHP, PL/ R, PL/ Ruby, PL/ Scheme, PL/Unix shell. [1]

Database & Query Optimisation	How does the database manage database and query optimization? In timescaleDB querying chunks is avoided with constraint exclusion. Basically, the system uses the constraints to quickly determine which chunks need to be read. This way not all chunks and data are read to answer queries. Inserts are optimised by ensuring that most of them are done on chunks in memory, discussed more in Database Performance Features. The local chunk indexes are stored in memory as well. [16]
Database Performance Features	What performance features exist in the database to support better performance. TimescaleDB implements a hypertable abstraction, which user interacts with. Behind the scenes, all data is heavily partitioned in multiple dimensions. Automatic space and time partitioning is implemented, for time the default value is 7 days and allows for user configuration. The partitions are called chunks and are stored as tables. Since timeseries tends to involve primarily inserts and writes to recent time intervals, TimescaleDB implements 'right-sized' chunks. This means that the chunks concerning most recent time interval fit in memory. As discussed, querying chunks is avoided with constraint exclusion. Lastly, there is an efficient retention policy in place that's up to user configuration. Data is only dropped/deleted if all the rows in the chunk are over a certain time interval (for example older than 5 years). In this way, rows are never deleted, and chunks do not become fragmented. Moreover, expensive vacuuming does not have to be performed and management becomes more efficient. [16]
Indexes	What indexing methods are available? TimescaleDB supports the similar indexing methods as Postgresql. An index can be created on any combination of columns if you include the time column. TimescaleDB creates an index on the time column by default. It allows for custom index, after which you can drop the default generated index. Instead of creating global indexes like Postgresql, indexes are built on each chunk, which is based on time intervals. [17]
Database Interfaces to other languages and tools	What interfaces does the database support to other languages, tools and applications? Timescale offers Promscale as a product, which is the observability backend for Prometheus metrics and OpenTelemetry traces with SQL support. Prometheus is an open-source technology designed for systems monitoring and alerting toolkit in cloud-native environments. OpenTelemetry is an open-source framework that provides software development kits, APIs and other tools. These tools are vendor-agnostic or –neutral to gather telemetry data from cloud-native applications and the infrastructure to analyse the health and performance. [4] [5]
Database Logging	What database logging methods are used to support database operations. Database logging is a background process. In this case, timescaleDB follows PostgreSQL. Therefore, logical and physical logging are implemented using Write-Ahead Logging (WAL). Where the logical log represents all logical operations done in the database, the physical log shows all physical operations. WAL means that any operation in the database must be written, after the operation has been logged. This way in the event of a crash we are sure to recover the database with the logs. [14]
Autonomous/ Serverless	Can the database operate in an autonomous or serverless mode. If so, give details Yes, timescaledb can be made serverless with Amazon Aurora Serverless. [15]
Machine Learning	Does the Database support Machine Learning? What features exists to support machine learning.

	TimescaleDB supports Apache MADlib for scalable in-database analytics and machine and deep learning. Timescale claims that it may be efficient to process these machine learning algorithms against the dataset by running them within the database rather than pulling a much smaller sample to an external software. [10]
Data Compression	Details of what data compression techniques are available in the database. Recent data is kept as uncompressed database rows, after a customisable amount of time the partition is compressed into columnar format. From TimescaleDB 2.3 onward, you can perform INSERT's directly into compressed partitions of hypertables. [3]
Documentation	What documentation exists for the database. Give link and a short description. https://docs.timescale.com It provides information on installing, getting started with, overview, how-to guides and tutorials on timescaleDB. In addition, there's documentation on the API references and the different product types.
Training Materials/Resources	Give links and details of what training materials are available. These can include videos, tutorials, training courses (by vendor and others), etc Timescale provides the following guides and tutorials to get you started and working with time-series data. https://docs.timescale.com/timescaledb/latest/how-to-guides/ https://docs.timescale.com/timescaledb/latest/tutorials/
Source Code/GitHub	Given links to the Source Code (GitHub etc), and a short description for each link. https://github.com/timescale
References	List references used to complete this report and your research. This should be addition to those listed in previous sections. 1. https://db-engines.com/en/system/TimescaleDB 2. https://postgresql.org/docs/ 3. https://www.timescale.com/blog/timescaledb-2-3-improving-columnar-compression-for-time-series-on-postgresql/ 4. https://www.splunk.com/en_us/data-insider/what-is-opentelemetry.html#:~:text=OpenTelemetry%20is%20an%20open%20source,understand%20their%20performance%20and%20health. 5. https://prometheus.io/docs/introduction/overview/ 6. https://docs.timescale.com/timescaledb/latest/overview/data-model-flexibility/ 7. https://docs.griddb.net/tutorial/wide-narrow/ 8. https://levelup.gitconnected.com/understanding-isolation-levels-in-a-database-transaction-af78aea3f44 9. https://www.postgresql.org/docs/current/transaction-iso.html 10. Introduction to time-series forecasting | Timescale Docs 11. https://www.timescale.com/blog/use-composite-indexes-to-speed-up-time-series-queries-sql-8ca2df6b3aaa/ 12. https://www.postgresql.org/docs/current/mvcc.html 13. https://dbdb.io/db/timescaledb 14. https://www.postgresql.org/docs/current/wal-intro.html 15. Timescale Products | Timescale 16. Time Series Database Lectures #6 - Mike Freedman (TimescaleDB) - YouTube 17. Index | Timescale Docs

Cassandrea …

Database	**Apache Cassandra (Columnar Databases for Big Data)**		
Licence Type(s)	Open source – Apache 2.0		
Original Developer(s)	Facebook	**Date/Year Created**	2008
Developed in Country	United States of America		
Current Owner	Apache Software Foundation	**Owner Since**	2009
Owner Country	United States of America		
Previous Names (& Year)	N/A		
Websites (Company & Product)	https://cassandra.apache.org/_/index.html (Website for product). https://apache.org/ (Apache software foundation company website).		
Database Category/Type	NoSQL – Columnar		
Derivative Systems	Cassandra was derived on the DynamoDB system – the original concept and implementation of Dynamo was released in a paper in 2007. **Derived from Cassandra -** Scylla FiloDB DataStax Heroic FaunaDB BlueFlood		
Database Description	Cassandra is a free, open source, NoSQL wide-column database. Cassandra makes use of node clusters which are decentralised and act as datacentres (essentially making the node clusters masterless).		
History	Cassandra, (named after the Greek priestess/ oracle) was originally founded and developed by Facebook before being open sourced and adopted by Apache.		
Written in what language	Java		

Supported Languages	Main = CQL
	Supported = C#, C++, Clojure, Erlang, Go, Haskell, Java, JavaScript, Perl, PHP, Python, Ruby, Scala		

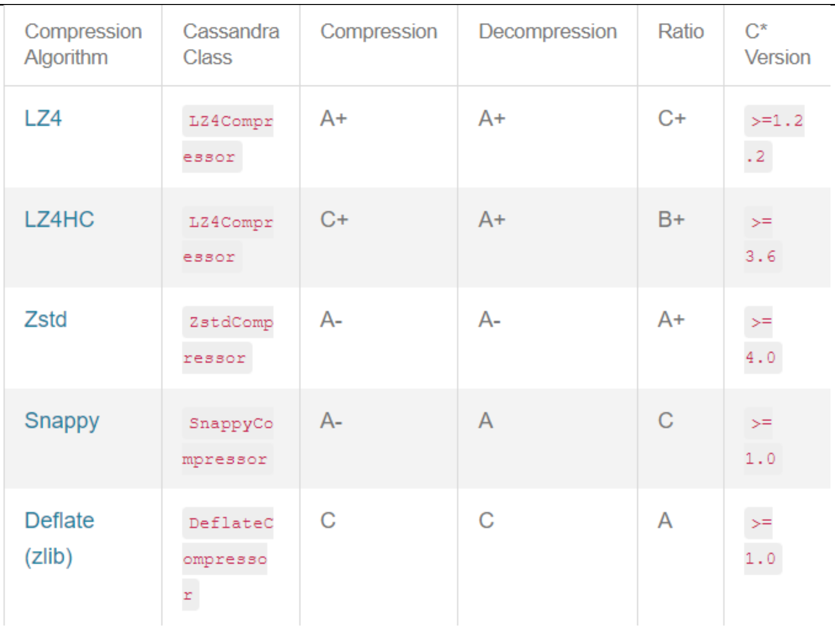
Operating Systems	All Operating systems with Java VM.		
Support Platforms	Available on all platforms – apache Cassandra can be installed on the cloud through a distributor, on linux machines, windows, docker and virtual box.		
Isolation Levels	Cassandra does not have RDBMS ACID transactions. It allows for light-weight transactions and all transactions are serializable not concurrent i.e., no multiple row transactions. Full -row isolation is in operation when using Cassandra – when writing to a row, only the client that is writing can see it. All other users will not see the update until it has been completed.		
Concurrency Control	Hybrid of Optimistic and Pessimistic Concurrency control is in place.		

	Cassandra does not support multi-row transactions. Therefore, does isolated replacement of rows within the in-memory structure (where all read/writes are processed) making it an Optimistic Concurrency Control. When there is high conflict on a single partition Cassandra will implement "per-tuple" locks (switch to Pessimistic Concurrency Control) which counter potentially high abort rates.
Persistence Model(s)	Commit-log based persistence model.
Data Models	Uses the n-ary storage model -> stores data in a contractual manner – must share a border i.e., can only append data (*What Is N-Ary Storage Model (NSM) | IGI Global*, n.d.).

Language Capabilities	CQL is the main language used in the database, but it is very similar to SQL.
	CREATE – is used in SQL and CQL but implemented differently.
	CREATE DATABASE name;
	CREATE KEYSPACE name WITH replication = {'class': 'SimplyStrategy', 'replication_factor':1};
	UPDATE – same syntax is used for both.
	UPDATE table SET colour = green WHERE id = 10;
	READ – syntax is the same BUT should be wary when using this in CQL.
	SELECT * FROM table WHERE age > 18 AND age < 50; this would work in SQL but in CQL would result in an error because of too much data filtering.
	DELETE – both have the same syntax BUT CQL does not immediately delete the data from the database.
	DELETE FROM table WHERE id = 7;

Stored Procedures/Functions	Cassandra does not support stored procedures. Developers write their own business logic in the application-level code which then use client-drives to read + write data to Cassandra db.
Database & Query Optimiszation	The optimisation of the Cassandra database can be attributed to its "key cache". Key caching is allowed by default on Cassandra. It is cache that is part of the partition index. By using key cache reduces the number of disk activity – less disk reads needed.
Database Performance Features	"Gossiping" is a key feature of Cassandra – this refers to the lack of slave-master relationship between each of the nodes. This reduces the database/data failure.
Indexes	Data can be accessed using data attributes that are also partition keys. For example, User_id is the column name for a 'User' table, and as user_id is also the partition key of the table thus the data search can be aided by the partition key – meaning that you can use WHERE to search for the data based on the attribute condition. There are also secondary indexes present in Cassandra. These are indexes that are "built over" column values. For example, a user_id also has user_address, the primary index is attached to the partition key (user_id) therefore you can search for the address based on a user_id but not vice versa. The purpose of the secondary index is to enable this "vice versa". Secondary indexes are not advised as they can negatively affect the write.
Database Interfaces to other languages and tools	Cassandra is a widely used database therefore there are a lot of options when it comes to tool/ interfaces etc. Here is a list of tools available for Cassandra: • React Virtuoso – virtual list component. • Presto – SQL query engine allows for querying data from original database.

	• Temporal – microservice platform. • Metriql – metrics store enables companies to define and share their metrics easily. (Cassandra Tools | ANANT, n.d.).
Database Logging	Logging function provided by Simple Logging Façade for Java. Cassandra logging can be constructed systematically or manually. Logs are written to the *system.log* as well as the *debug.login*. Logging levels can be set using: ALL, TRACE, DEBUG, INFO (default), WARN, ERROR, OFF (*Configuring Logging | Apache Cassandra 3.0*, n.d.).
Autonomous/Serverless	A serverless database refers to the fact that developers can work without having to worry about the servers. Cassandra is a multi-cloud serverless database. DataStax Astra enables Cassandra to auto scale based on the traffic/app requirements. (*DataStax Just Took Cassandra Serverless. Here's Why That's a Major Breakthrough*, n.d.)
Machine Learning	Cassandra supports machine learning. Machine learning requires access to large data sets in order to properly train and execute various algorithms, therefore the database feeding the data must do so efficiently. Cassandra is designed to support machine learning by offering high availability and partition tolerance: Scalability – Cassandra is capable of scaling horizontally by adding nodes. The addition of nodes also increases the storage of the database with each node capable of storing approx. 1 TB of data. Decentralised – machine learning must be given data continuously, therefore Cassandra is ideal because it is extremely fault tolerant. Each of the nodes in a cluster are identical, and responsible for different partitions and communicate with each other via gossiping. Replication – data is also replicated around the nodes. If a node fails, there are other nodes that have the data therefore the database does not need downtime to fix the issue and usually the repair of the node is automated (*Real World Machine Learning with Apache Cassandra and Apache Spark (Part 1) | DataStax*, n.d.).
Data Compression	Cassandra offers 5 compression algorithms that make trade-offs between Compression speed, decompression speed, and ratio. The following image is rough performance grade for each of the 5 algorithms based on these areas. With A being better and F being worse.



Compression Algorithm	Cassandra Class	Compression	Decompression	Ratio	C* Version
LZ4	`LZ4Compressor`	A+	A+	C+	`>=1.2.2`
LZ4HC	`LZ4Compressor`	C+	A+	B+	`>= 3.6`
Zstd	`ZstdCompressor`	A-	A-	A+	`>= 4.0`
Snappy	`SnappyCompressor`	A-	A	C	`>= 1.0`
Deflate (zlib)	`DeflateCompressor`	C	C	A	`>= 1.0`

Image taken from Apache documentation (*Compression | Apache Cassandra Documentation*, n.d.)

Documentation	https://cassandra.apache.org/doc/latest/
Training Materials/Resources	https://www.javatpoint.com/cassandra-tutorial https://www.tutorialspoint.com/cassandra/index.htm
Source Code/GitHub	https://github.com/apache/cassandra

References

Cassandra Tools | ANANT. (n.d.). Retrieved November 11, 2022, from https://cassandra.tools/

Compression | Apache Cassandra Documentation. (n.d.). Retrieved November 10, 2022, from https://cassandra.apache.org/doc/latest/cassandra/operating/compression.html

Configuring logging | Apache Cassandra 3.0. (n.d.). Retrieved November 11, 2022, from https://docs.datastax.com/en/cassandra-oss/3.0/cassandra/configuration/configLoggingLevels.html

DataStax just took Cassandra Serverless. Here's why that's a major breakthrough. (n.d.). Retrieved November 10, 2022, from https://thestack.technology/datastax-astra-cassandra-serverless/

Real World Machine Learning with Apache Cassandra and Apache Spark (Part 1) | DataStax. (n.d.). Retrieved November 10, 2022, from https://www.datastax.com/blog/real-world-machine-learning-with-apache-cassandra-and-apache-spark-part-1

What is N-ary Storage Model (NSM) | IGI Global. (n.d.). Retrieved November 11, 2022, from https://www.igi-global.com/dictionary/n-ary-storage-model-nsm/39626

YugaByte Database …

Database	YugabyteDB		
Licence Type(s)	Opensource version :: Apache 2.0 License Priced versions :: **YugabyteDB Anywhere** :: pay for the number of cpu cores selected for the environment **YugabyteDB Managed** :: completely managed ,only pay per use Polyform Free Trial License 1.0.0		
Original Developer(s)	Kannan Muthukkaruppan, Karthik Ranganathan, and Mikhail Bautin	Date/Year Created	2016
Developed in Country	USA		
Current Owner	Yugabyte Inc	Owner Since	2016
Owner Country	USA		
Previous Names (& Year)	Nil		
Websites (Company & Product)	https://www.yugabyte.com/		
Database Category/Type	Distributed database and Database as a service		
Licence Types	Opensource version:: Apache 2.0 License Priced versions :: **YugabyteDB Anywhere**:: pay for the no of cpu cores selected for the environment **YugabyteDB Managed** :: completely managed ,only pay per use Polyform Free Trial License 1.0.0		
Derivative Systems	PostgreSQL and the YSQL API are compatible. This means that YugabyteDB may be used with PostgreSQL client drivers, the psql command line shell, IDE integrations like TablePlus and DBWeaver, and more. In relation to the Apache Cassandra Query Language, the same idea holds true for YCQL. Also it is similar to Google Spanner in most of the functionalities		
Database Description	Enterprise-grade relational database capabilities are combined in a special way by YugabyteDB with the robustness and horizontal scalability of cloud native systems. Similar to PostgreSQL but redesigned for the cloud native era is YugabyteDB.The DB also supports semi relational SQL API that is best suited for large-scale OLTP and HTAP applications that require rapid queries and vast amounts of data ingestion.		
History	YugabyteDB has had the following major (stable) releases: v2.14 in July 2022.		

	v2.12 in February 2022. (There was no v2.10 release.) v2.8 in November 2021. v2.6 in July 2021. v2.4 in January 2021. v2.2 in July 2020. v2.1 in February 2020. v2.0 in September 2019. v1.3 in July 2019. v1.2 in March 2019. v1.1 in September 2018. v1.0 in May 2018. v0.9 Beta in November 2017.
Written in what language	The distributed storage and transactions layer, as well as the YCQL query layer, and the YSQL layer, which is based on PostgreSQL, are written mostly in C++ and C, respectively. There are certain components of the build system and test suite are written in Python, Java, and Bash. A few data and network communication formats are also defined using Protocol Buffers.
Supported Languages	Java , Python , Go , Node.js, C++, C, C#,Ruby, Rust, PHP, Scala
Operating Systems	YugabyteDB :: the recommended operating systems are CentOS 7.n and RHEL 7.n , also supports a variety of OS systems YugabyteDB Anywhere :: AlmaLinux OS 8 (default), CentOS, Oracle Linux , Oracle Linux 8, Ubuntu 18, Ubuntu 20, Red Hat Enterprise Linux 8
Support Platforms	On premises , Cloud
Isolation Levels	Serializable Isolation :: For all committed transactions, this level simulates serial transaction execution, making it appear as though each transaction had been processed sequentially rather than concurrently. Snapshot Isolation :: When using snapshot isolation, transactions are shielded from other concurrently running transactions' uncommitted data and modifications made during transaction execution. Read Committed Isolation:: Before a statement is published, all committed data will be seen in the transaction as a whole (note that this implicitly also means that the statement will see a consistent snapshot)
Concurrency Control	Optimistic :: Without preventing any of the actions carried out as part of the transaction, optimistic locking postpones the evaluation of whether a transaction complies with the separation and other consistency criteria until after it has completed. Pessimistic :: No additional transaction would be able to lock that row as long as the initial transaction that locked it has not finished (either COMMIT or ABORT).
Persistence Model(s)	Storage layer is key to object/document store(DocDB key , DocDB value) DocDB key: keys in the DocDB document model are made up of one or more hash-organized components, then 0 or more ordered (range) components DocDB value: The DocDB document data model values include primitive types: such as double, text, timestamp, int32, int64, and so forth

	non-primitive types: Scalar keys are mapped to values via these objects, which may also be scalar or sorted maps.
Data Models	Relational Data Model Document Data Model
Language Capabilities	YSQL and YCQL are the language API's provided by YugabyteDB Since YSQL reuses the Postgre's query layer , most of its features are also present here. YCQL is similar to Cassandra Query language So, we believe it is fair to say that the language is extensively implemented
Stored Procedures/Functions	YSQL yugabyte Supports Procedures/functions YSQL being compatible with postgres supports procedures using external languages which is supported in postgres (C ,PERL)
Database & Query Optimization	YSQL provides following wo views you can use to identify SQL statements and their performance characteristics pg_stat_activity : A summary of current activity, including SQL, is provided by pg stat activity. It only indicates the current state of the PostgreSQL backend and does not provide any other information about the SQL that was executed besides the query text. pg_stat_statements: SQL statements that have been run in the past are detailed in pg stat statements. Because the PostgreSQL I/O codepath is not utilized, no 'block' or 'blk' information is currently collected (which is logical and physical I/O information). pg_stat_progress_copy: provides status information of a COPY command execution Similar to PostgreSQL, YugabyteDB offers the EXPLAIN statement to display the YSQL query execution plan for a specific SQL statement. You can use EXPLAIN to determine where in the query plan a query is spending the majority of its time, and then use this knowledge to choose the best strategy for enhancing query performance.
Database Performance Features	What performance features exist in the database to support better performance. • **Accelerate developer productivity** by quickly developing new features and services • **Unlock the potential of your apps** by adjusting the scale with them up, or down, as necessary • **Improve your customer experience** by going beyond uptime SLAs • **Reduce operational costs** with a lower TCO by only purchasing what is necessary • **Protect your critical data in production environments** with built-in security controls • **Seamlessly upgrade** utilizing cutting-edge migration technologies from your outdated database
Indexes	Primary keys, Foreign keys, Secondary indexes, Unique indexes, Partial indexes, Expression indexes, Covering indexes, GIN indexes
Database Interfaces to	You can connect to your YugabyteDB clusters using third-party clients

r languages and tools	because YugabyteDB is compatible with PostgreSQL and Cassandra Pgadmin Apache Superset Arctype DBeaver TablePlus DbSchema Cassandra Workbench Yugabyte also provides numerous drivers for multiple languages to connect and interact with the ysql and ysql API.
Database Logging	pgAudit:: Detailed session and/or object audit logging is provided by YugabyteDB YSQL using PostgreSQL Audit Extension (pgAudit) via YugabyteDB YB-TServer logging. Session Logging :: For each user's session, session logging is enabled. Logging sessions should be enabled for all DML and DDL statements, as well as for all relations in DML statements. Object-level audit logging:: Statements that have an impact on a specific relation are logged by object audit logging. The only supported commands are SELECT, INSERT, UPDATE, and DELETE. TRUNCATE is not logged in the object audit.
Autonomous/Serverless	Serverless mode using YugabyteDB Anywhere and YugabyteDB Managed
Machine Learning	We have not come across any features supporting machine learning during our research
Data Compression	In essence, data compression is the process of reducing the size of on-disk data to save storage costs and accelerate backup times. Blocks of data can be compressed using techniques like Prefix, Snappy, LZO, LZ4, ZLib, and LZMA.
Documentation	https://docs.yugabyte.com/
Training Materials/Resources	https://vladmihalcea.com/yugabytedb/ https://university.yugabyte.com/ https://www.udemy.com/course/distributed-sql-and-yugabytedb-fundamentals/ https://forum.yugabyte.com/ https://app.slack.com/client/TG1HY4TGD/CG0KQF0GG
Links	
Source Code/GitHub	https://github.com/yugabyte/yugabyte-db :: the entire YugabyteDB code

	repo
Links	https://www.yugabyte.com/ https://www.yugabyte.com/about/ https://www.yugabyte.com/blog/ https://www.yugabyte.com/content-library/ https://www.yugabyte.com/success-stories/ https://www.yugabyte.com/compare-products/ https://www.yugabyte.com/events/
References	• YugabyteDB, Wikipedia. (2022, August 19). Retrieved from https://en.wikipedia.org/wiki/YugabyteDB • Legal information. Retrieved November 10, 2022, from https://docs.yugabyte.com/preview/legal/ • YugabyteDB coding style. Retrieved November 10, 2022, from https://docs.yugabyte.com/preview/contribute/core-database/coding-style/ • Supported operating systems and architectures. Retrieved November 10, 2022, from https://docs.yugabyte.com/preview/yugabyte-platform/configure-yugabyte-platform/supported-os-and-arch/ • Isolation Levels. Retrieved November 10, 2022, from https://docs.yugabyte.com/preview/explore/transactions/isolation-levels/ • Explicit locking. Retrieved November 10, 2022, from https://docs.yugabyte.com/preview/architecture/transactions/explicit-locking/ • Persistence in YugabyteDB. Retrieved November 10, 2022, from https://docs.yugabyte.com/preview/architecture/docdb/persistence/ • Get query statistics using pg_stat_statements. Retrieved November 10, 2022, from https://docs.yugabyte.com/preview/explore/query-1-performance/pg-stat-statements/ • View live queries with pg_stat_activity. Retrieved November 10, 2022, from https://docs.yugabyte.com/preview/explore/query-1-performance/pg-stat-activity/ • View COPY status with pg_stat_progress_copy. Retrieved November 10, 2022, from https://docs.yugabyte.com/preview/explore/query-1- -copy/ • Analyze queries with EXPLAIN. Retrieved November 10, 2022, from https://docs.yugabyte.com/preview/explore/query-1-performance/explain-analyze/#the-explain-statement • Choudhury, S. *A Busy Developer's Guide to Database Storage*

	Engines - Advanced Topics. (2022, July 05). Retrieved from https://www.yugabyte.com/blog/a-busy-developers-guide-to-database-storage-engines-advanced-topics/ - Indexes and constraints. Retrieved November 10, 2022, from https://docs.yugabyte.com/preview/explore/indexes-constraints/ - Third party tools. Retrieved November 10, 2022, from https://docs.yugabyte.com/preview/tools/#yugabytedb-prerequisites - Security. Retrieved November 10, 2022, from https://docs.yugabyte.com/preview/explore/security/security/#session-logging

Graph Data in Neo4j …

Database	Neo4j Graph Database		
Licence Type(s)	1. *Neo4j Community Edition* is suitable for personal and small-scale projects and is completely open-source and licensed under GPL v3. 2. *Neo4j Enterprise Edition* is built for more robust applications where scalability and availability are crucial and is available in 3 licensing options: i. *Neo4j Commercial License* – a subscription-based for closed-source commercial applications. ii. *Neo4j Developer License* – a free-to-use development experience of Enterprise Edition on local environments. iii. *Neo4j Evaluation License* - [1]		
Original Developer(s)	Neo 4j Inc. (CEO Emil Eifrem) [2]	Date/Year Created	2002 [2]
Developed in Country	United States [2]		
Current Owner	Neo4j Inc. [2]	Owner Since	2002
Owner Country	United States		
Previous Names (& Year)	N/A		
Websites (Company & Product)	Company (https://neo4j.com/) Product (https://neo4j.com/product/neo4j-graph-database)		
Database Category/Type	Neo4j is a Graph Database and is more broadly categorised as a NoSQL database. [3]		
Licence Types			
Derivative Systems	What other databases are based on this database? What database is this database derived from? AuraDB is a fully automated cloud database as a service built with Neo4j. Neo4j and graph databases in general a novel approach to storing data with Neo4j being a pioneer in that domain.[4]		
Database Description	Neo4j as a graph database stores data in a graph which is itself a network of *nodes* (or *vertices*) and the *relationships* between these nodes are called *edges*. Neo4j treats both its nodes and edges as first-class objects and they can possess properties in the form of key-value pairs, labels or metadata; these forms the build blocks of its property graph model. It uses a declarative SQL-inspired query language called *Cypher*. Neo4j is a fully ACID-compliant transactional database, that has been adopted for robust use-cases such as social network analysis, fraud-detection, AI/ML applications to name a few.		
History	Neo4j DB was created in 2007 and had its first release (version 1.0) in February 2010. In July of 2011, the first implementation of its *Cypher* query language was introduced in version 1.4. By December 2013, we saw the introduction of its visual IDE in its version 2.0 release. Its next major release (v.3.0) in 2016		

	implement features such as user-defined store procedures referred to as Awesome Procedures on Cypher (APOC) as well as language drivers for Java, .NET, JavaScript and Python. Subsequent release have gone on to add several enterprise level features such as multi-clustering, enterprise-level security, query monitoring, native indexing, full-text search etc. Its latest version v5.0 was released in October 2022. [5, p. 4][6]
Written in what language	Neo4j is written in mainly Java and partly in Scala. [7]
Supported Languages	Neo4j officially supports the drivers for .Net, Java, JavaScript, Go, and Python. [8]
Operating Systems	Linux, Windows, OS X, All OS with JVM. [9]
Support Platforms	Supported on systems with x86 and 64 and ARM architectures on physical, virtual, or containerized platforms. [9]
Isolation Levels	What isolation levels are support by the database. The default isolation level is *read-committed*. This means transactions only have access to data that has already been committed. While read-committed isolation is typically weaker than *serializable,* it is usually more performant at scale and fits most use cases. However, the Neo4j Java API makes provision for explicit locking of nodes and relationships which can mimic the effect of higher isolation levels. [10]
Concurrency Control	What concurrency control is support by the database Neo4j uses locks which may create the possibility of deadlocks. However, Neo4j will foresee any deadlock (caused by multiple transactions needing to access nodes or edges with locks) and throw an exception. Before the exception is thrown, the transaction is flagged for rollback. When the transaction is complete, all locks acquired by the transaction are removed. Other transactions waiting for locks held by the transaction causing the deadlock can move forward once the locks are released. The user can then re-execute the transaction if necessary. [11]
Persistence Model(s)	What persistence models are supported by the database Neo4j uses a *Polyglot Persistence Model* for persisting data. This means that it utilizes multiple data storage techniques to store data but it relies primarily on fixed-sized records that are stored linked lists to connect the nodes and relationships in the database. [12], [13]
Data Models	What data models or combinations of models are supported Neo4j as a graph database utilizes a graph property data model with nodes and relationship between the nodes which can have properties of their own. [14]
Language Capabilities	That is the main language used to interact with the database. How extensive is the implementation of the language? What language standards are supported. What things does it do differently? Neo4j's *Cypher* property graph query language was created to and has become the primary query language for the database. However, even before Cypher existed, it's Java API was used and is still fully functional. *Cypher* is inspired by SQL and SPARQL, but its uniqueness is in its almost-visual approach to query data. It represents nodes in rounded brackets and relationships with square brackets between arrows e.g. `(node)-[:IS _CONNECTED_TO]->(anotherNode)`

	Cypher's syntax resembles ASCII art and leans on the brain's innate propensity to recognize patterns, which is conceptually in-line with the concept of graph databases. Other notable features are graph concepts include paths queries (simple, variable length and shortest-path functions), the support for functions, operations and predicates on lists and query chaining. *Cypher* was first created at Neo4j Inc. in 2011 and was subsequently open-sourced in 2015 under the *openCypher* project. Since then, it has also been implemented in other graph databases such as Amazon Neptune, SAP's HANA graph and Redis graph. In an accepted proposal in September 2019, Cypher was marked as a language to model significantly in the creation of a standard Graph Query Language for property graphs. [15]–[18]
Stored Procedures/Functions	Does the database support stored procedures/functions? Can external procedures be created/used? Can procedures/functions be created using other languages (C, Java, Python, etc) Neo4j currently supports only stored procedures in Cypher and with its Java API. However, drivers still exist in JS, python, Go and other popular language for connecting to the database. [19]
Database & Query Optimisation	How does the database manage database and query optimization? Neo4j has a query optimizer which is called the Cypher query planner which optimizes and transforms queries using an execution plan. An execution plan is tree-like combination of operators, which represent a singular or unit task of an overall query operation. To minimize the resource-cost of an execution plan, it is recommended to use parameters instead of literals where possible, this way, rather than parsing and building new execution plans, Cypher can re-use the query. [20], [21]
Database Performance Features	What performance features exist in the database to support better performance. All Neo4j editions performance features include Native graph processing and storage, caching, cost-based query optimizer, native label indexes, composite indexes, full-text indexes for both nodes and relationships, auto-reuse of space. The Enterprise editions provides additional features like clustering, automatic cache warming and load balancing with Neo4j drivers. [22], [23]
Indexes	What indexing methods are available? Neo4j supports the following index types: • Fulltext index • Lookup index • Point index • Range index • Text index [24]
Database Interfaces to other languages and tools	What interfaces does the database support to other languages, tools and applications? Neo4j Browser (its just like SQL developer and ssms), Neo4j Bloom for data exploration, officially supports Dotnet, Java, JavaScript, Go, and Python and from community it supports ruby, PHP, R and Perl. [25]
Database Logging	What database logging methods are used to support database operations.

	Neo4j utilizes *Apache Log4j 2* for logging as part of its performance monitoring strategy. All default Log4j components are supported by Neo4j and logging is configured using two XML files *conf/server-logs.xml* and *conf/user-logs.xml.* These configuration files comprise of 3 major components: • *Appenders* — for defining output locations such as a file, the console, network socket, etc. • *Layouts* — for specifying how the output is formatted e.g. plain-text, JSON, CSV, etc. • *Loggers* — for routing log events to one or more appenders. Logs are appended to generated rolling files, which include: The user log, debug log, garbage collection log, HTTP log, security log, etc. [26]
Autonomous/Serverless	Can the database operate in an autonomous or serverless mode. If so, give details. AuraDB is a fully automated cloud database as a service built with Neo4j that runs on AWS and GCP cloud providers. Its enterprise tier offering includes extra features such as enterprise-class security, frequent backups, and clustering amongst others. [4]
Machine Learning	Does the Database support Machine Learning? What features exists to support machine learning. Neo4j supports machine Learning through its *Graph Data Science Library.* The library process access to a collection of graph algorithm and ML pipelines and models. [27]
Data Compression	Details of what data compression techniques are available in the database. Out of the box there is no compression available but with an APOC (*Awesome Procedures on Cypher*) plugin strings can be compressed into byte arrays
Documentation	What documentation exists for the database. Give link and a short description. Neo4j has an extensive documentation page that can be found at (https://neo4j.com/docs).
Training Materials/Resources	Give links and details of what training materials are available. These can include videos, tutorials, training courses (by vendor and others), etc - Neo4j YouTube channel (https://www.youtube.com/c/neo4j/videos) - Neo4j Official Docs (https://neo4j.com/docs)
Links	Other product related links and a short description for each link. • Neo4j AuraDB (https://neo4j.com/cloud/platform/aura-graph-database) - fully managed cloud service DB • Neo4j Graph Data Science (https://neo4j.com/product/graph-data-science) - connected data analytics and machine learning platform • Neo4j AuraDS (https://neo4j.com/cloud/platform/aura-graph-data-science) - fully managed graph data science engine [28]
Source Code/GitHub	Given links to the Source Code (GitHub etc), and a short description for each link. Neo4j GitHub Repository (https://github.com/neo4j/neo4j)

References	List references used to complete this report and your research. This should be addition to those listed in previous sections. [1] ‘Neo4j Licensing’, *Neo4j Graph Data Platform*. https://neo4j.com/licensing/ (accessed Nov. 10, 2022). [2] ‘Company’, *Neo4j Graph Data Platform*. https://neo4j.com/company/ (accessed Nov. 10, 2022). [3] ‘Neo4j Graph Database’, *Neo4j Graph Data Platform*. https://neo4j.com/product/neo4j-graph-database/ (accessed Nov. 10, 2022). [4] ‘Neo4j AuraDB overview - Neo4j Aura’, *Neo4j Graph Data Platform*. https://neo4j.com/docs/aura/auradb/ (accessed Nov. 11, 2022). [5] ‘Database – Neo4j Graph Data Platform’. https://neo4j.com/release-notes/database/ (accessed Nov. 10, 2022). [6] ‘Impossible Is Nothing: The History (& Future) of Graph Data [GraphConnect Recap] - Neo4j Graph Data Platform’. https://neo4j.com/blog/history-and-future-of-graph-data/ (accessed Nov. 10, 2022). [7] ‘What is a Graph Database? - Developer Guides’, *Neo4j Graph Data Platform*. https://neo4j.com/developer/graph-database/ (accessed Nov. 10, 2022). [8] ‘Drivers & Language Guides - Developer Guides’, *Neo4j Graph Data Platform*. https://neo4j.com/developer/language-guides/ (accessed Nov. 10, 2022). [9] ‘System requirements - Operations Manual’, *Neo4j Graph Data Platform*. https://neo4j.com/docs/operations-manual/5/installation/requirements/ (accessed Nov. 10, 2022). [10] ‘Transaction management - Java Reference’, *Neo4j Graph Data Platform*. https://neo4j.com/docs/java-reference/5/transaction-management/ (accessed Nov. 10, 2022). [11] ‘Transaction management - Java Reference’, *Neo4j Graph Data Platform*. https://neo4j.com/docs/java-reference/5/transaction-management/ (accessed Nov. 10, 2022). [12] ‘Understanding Neo4j’s data on disk - Knowledge Base’, *Neo4j Graph Data Platform*. https://neo4j.com/developer/kb/understanding-data-on-disk/ (accessed Nov. 08, 2022). [13] ‘Product DNA: Master Data Graph Enabling the Digital Transformation at Lockheed Martin @ Neo4j GraphConnect 2018’. https://neo4j.com/graphconnect-2018/session/master-data-graph-digital-transformation-lockheed-martin/ (accessed Nov. 08, 2022). [14] ‘Graph Modeling Guidelines - Developer Guides’, *Neo4j Graph Data Platform*. https://neo4j.com/developer/guide-data-modeling/ (accessed Nov. 10, 2022). [15] ‘Introduction - Neo4j Cypher Manual’, *Neo4j Graph Data Platform*. https://neo4j.com/docs/cypher-manual/5/introduction/ (accessed Nov. 11, 2022). [16] N. Francis *et al.*, ‘Cypher: An Evolving Query Language for Property Graphs’, in *Proceedings of the 2018 International Conference on Management of Data*, Houston TX USA, May 2018, pp. 1433–1445. doi: 10.1145/3183713.3190657. [17] ‘Cypher Query Language - Developer Guides’, *Neo4j Graph Data Platform*. https://neo4j.com/developer/cypher/ (accessed Nov. 11, 2022). [18] ‘Cypher (query language)’, *Wikipedia*. Nov. 02, 2022. Accessed: Nov. 11, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Cypher_(query_language)&oldid=1119702724#cite_note-BSI_39075_GQL-22

[19] ‘User defined procedures and functions - Getting Started’, *Neo4j Graph Data Platform*. https://neo4j.com/docs/getting-started/5/cypher-intro/procedures-functions/ (accessed Nov. 11, 2022).

[20] ‘Query tuning - Neo4j Cypher Manual’, *Neo4j Graph Data Platform*. https://neo4j.com/docs/cypher-manual/5/query-tuning/ (accessed Nov. 11, 2022).

[21] ‘Execution plans - Neo4j Cypher Manual’, *Neo4j Graph Data Platform*. https://neo4j.com/docs/cypher-manual/5/execution-plans/ (accessed Nov. 11, 2022).

[22] ‘Introduction - Operations Manual’, *Neo4j Graph Data Platform*. https://neo4j.com/docs/operations-manual/5/introduction/ (accessed Nov. 11, 2022).

[23] ‘Performance - Operations Manual’, *Neo4j Graph Data Platform*. https://neo4j.com/docs/operations-manual/5/performance/ (accessed Nov. 11, 2022).

[24] ‘Indexes for search performance - Neo4j Cypher Manual’, *Neo4j Graph Data Platform*. https://neo4j.com/docs/cypher-manual/5/indexes-for-search-performance/ (accessed Nov. 11, 2022).

[25] ‘Neo4j documentation - Drivers and APIs - Neo4j Documentation’, *Neo4j Graph Data Platform*. https://neo4j.com/docs/docs/drivers-apis/ (accessed Nov. 11, 2022).

[26] ‘Logging - Operations Manual’, *Neo4j Graph Data Platform*. https://neo4j.com/docs/operations-manual/5/monitoring/logging/ (accessed Nov. 11, 2022).

[27] ‘Machine learning - Neo4j Graph Data Science’, *Neo4j Graph Data Platform*. https://neo4j.com/docs/graph-data-science/2.2/machine-learning/machine-learning/ (accessed Nov. 11, 2022).

[28] ‘Product’, *Neo4j Graph Data Platform*. https://neo4j.com/product/ (accessed Nov. 11, 2022).

[3] https://neo4j.com/labs/apoc/4.3/overview/apoc.util/apoc.util.compress/

Oracle TimesTen …

Database	**Oracle TimesTen**		
Licence Type(s)	Commercial and Free Options Oracle TimesTen Database In-Memory license grants unrestricted access to some of the TimesTen feature set, however, there is a separate license for the Application-Tier DB Cache which grants access to the entire feature set. This feature set is designed for TimesTen to be used alongside Oracle DB as a cache, and also requires an Oracle Enterprise license. The TimesTen Express (XE) is free of charge community edition but has limitations in places such as a maximum perm-size set at 16GB, and K-safety for TimesTen Scaleout capped at 2. https://docs.oracle.com/en/database/other-databases/timesten/22.1/licensing/licensing-information.html **https://docs.oracle.com/database/timesten-18.1/TTLIC/toc.htm#TTLIC345**		
Original Developer(s)	Hewlett-Packard labs	Date/Year Created	1994
Developed in Country	United States of America		
Current Owner	Oracle	Owner Since	2005
Owner Country	United States of America		
Previous Names (& Year)	Smallbase 1994, TimesTen 1996, Oracle TimesTen https://en.wikipedia.org/wiki/TimesTen		
Websites (Company & Product)	https://www.oracle.com/ie/database/technologies/related/timesten.html https://docs.oracle.com/cd/E18283_01/timesten.112/e13065/using.htm https://www.oracle.com/ie/		
Database Category/Type	In-Memory Row-Store OLTP RDBMS/Data Store		
Derivative Systems	Oracle DB can be configured to use TimesTen as a cache. This is part of the Application-Tier license, and also requires an Oracle Enterprise license. See previous section regarding licensing. https://datacadamia.com/db/timesten/cache		
Database Description	TimesTen is a relational In-Memory database that aimed at improving OLTP operations by storing data in main memory. This is supposed to reduce the latency for I/O operations associated with disk storage.		
History	https://docs.rackspace.com/blog/introduction-to-oracle-times-ten/ TimesTen was developed by Hewlett Packard in 1994, and then proceeded to split off in 1996 as a startup. Eventually it was acquired by Oracle in 2005. Since then, it has had gone through the following releases. -> 6.0 -> 7.0 -> 11gR2 -> 18c -> 22.1		

Written in what language	Unsure, but would hazard a guess that it is written in C, considering that Oracle DB is also written in C.
Supported Languages	C++, Java, PL/SQL, SQL
Operating Systems	- TimesTen Scaleout & Classic for Linux x86 x64 - TimesTen Classic for Oracle Solaris x86 x64, Oracle Solaris SPARC x64, and IBM AIX Power x64 - TimesTen Client for Windows x64 and Mac OS X - TimesTen XE is only available for Linux
Support Platforms	Oracle does not offer a cloud service license for TimesTen themselves, however TimesTen can be deployed to other cloud providers. For example, it can be installed on a cloud VM or even to Kubernetes containers. Oracle also offers TimesTen as an additional service to their Exalogic product, which is a hardware system that is pre-configured and then installed on customer premise. https://www.oracle.com/a/tech/docs/wp-timesten-exalogic-2215631.pdf https://blogs.oracle.com/timesten/post/introduction-to-oracle-timesten-in-memory-database
Isolation Levels	TimesTen supports two levels of transaction isolation. Read-Committed and Serializable isolation. Read-Committed isolation means that when data is read by a 'reader', it is a separate copy of data separate from 'writers'. Writers block only other writers, and readers under the serializable isolation option. Readers are not blocked. ANSI Read-Committed Isolation is achieved through using multiple-version concurrency control, where multiple versions of data are created, allowing read/write operations to continue in parallel. This is the default option for the system. This is the second loosest isolation level of four. The next level of isolation is serializable where locks are acquired for a transaction. This means that if a transaction performs any CUD operations, that transaction must complete before another transaction can perform any CRUD operations. The reason for only having two isolation levels is to improve OLTP performance on data. https://dbdb.io/db/timesten/revisions/10 http://luna-ext.di.fc.ul.pt/oracle11g/timesten.112/e13065/trans.htm https://docs.oracle.com/database/timesten-18.1/TTCIN/concurren.htm#TTCIN172
Concurrency Control	Further than the previously mentioned Isolation Levels, there are also multiple options for the locking mechanism for serialized isolation and read-committed isolation. There are shared and exclusive locks, where shared locks mean the resource can be shared but does not allow alteration of the row/table/view. Exclusive locks mean that only one transaction can be performed on the database at a time, and do not allow any sharing of the data. This lock is for modifying data. Simply put shared locks = read, exclusive locks = write. When a read-committed transaction is made, they acquire exclusive locks on the items they write to, but not on the items that they read. Locks are released once a transaction is committed or rolled back. Additionally, there are multiple locking levels: row, table, and database. As the names suggest, these levels lock a row, a full table, or an entire database for a specified transaction. Obviously, row-based locking is beneficial when there are multiple concurrent transactions taking place, but in the event of bulk-loading data, database locking is sometimes needed, but only when concurrent transactions are not operating.

	It is also worth noting that all database locks are exclusive locks. For OLTP performance, Row level locking is used to maximise throughput. https://dbdb.io/db/timesten *https://docs.oracle.com/database/timesten-18.1/TTCIN/concurrent.htm#TTCIN171*
Persistence Model(s)	TimesTen designates two segments within memory at runtime as data stores. The temporary store holds data structures that queries rely on for execution such as locks and compiled commands. The second segment is the Permanent data store and contains the actual table data and indexes. Data is loaded from disk into the permanent store when a database is loaded. Data is then persisted at designated check points and stored on disk. TimesTen uses checkpoints to store snapshots of the database at regular intervals, which can lead to a significant amount of I/O operations depending on the database size and number of changes since the last checkpoint. This checkpointing helps maintain durability, as the database can more quickly recover from a crash because up-to-date information is stored to the disk and can be loaded. Additionally, TimesTen creates transaction logs, which are automatically persisted to disk. When checkpoints are loaded, it reduces how far back the transaction logs must go to recover from a crash. Depending on the version of TimesTen being used, classic or scaleout, each database or node has its own set of checkpoint and transaction files respectively. Again, like transaction isolation, there are two types of checkpoints: Non-Blocking (Fuzzy) and Blocking. Fuzzy checkpoints don't acquire locks, and so may include non-committed transactions. If a recovery is to be performed with a Fuzzy checkpoint file, the most recent version should be used alongside the transaction logs. However, blocking checkpoints acquire locks, preventing data from being modified. They may take longer, and will disrupt and prevent other transactions from occurring. https://docs.oracle.com/database/timesten-18.1/TTOPR/trans.htm#TTOPR403
Data Models	Oracle TimesTen follows the SQL standard as a relational database. It stores data in the row format. https://docs.oracle.com/database/timesten-18.1/TTOPR/comp.htm#TTOPR329
Language Capabilities	PL/SQL, unfortunately was unable to explicitly confirm which SQL standard TimesTen conforms to, but Oracle Database 18c conforms to SQL:2016, which implies that TimesTen supports a similarly recent standard if maximum compatibility is desired between Oracle and TimesTen. https://docs.oracle.com/database/timesten-18.1/TTSQL/TTSQL.pdf
Stored Procedures/Functions	TimesTen supports PL/SQL, which itself allows for stored procedures. The PREPARE keyword is used before a statement for it to be prepared, then to execute and clear the prepared statement the exec; and free; commands are used respectively. https://docs.oracle.com/database/timesten-18.1/TTPLS/overview.htm#TTPLS118
Database & Query Optimization	Oracle uses a cost-based optimizer for queries, with priority being given to performance rather than memory usage, where multiple paths can be generated which are based on indexes, sorting and constraints. The costs variables include: - Table and column statistics - Constraints such as primary keys - Indexes - Amount of data - Number of unique values

	- Predicates - Scan methods (full-table, rowid, range/hash index) - Join algorithm Optimization is the third step of code generation: 1. Parsing, 2. Semantic Analysis, 3. Optimizer, and 4. Code generation. The optimizer is not guaranteed to generate the most optimal plan, as it must balance time and resources used to optimize the query, with the actual execution of the query. With this considered, the optimizer can take user-generated hints into account, with hints at differing levels such as statement-level, transaction-level, and connection-level, in descending precedence. https://www.geeksforgeeks.org/cost-based-optimization/ https://docs.oracle.com/database/timesten-18.1/TTCIN/query.htm#TTCIN181
Database Performance Features	Oracle has an entire manual section in performance tuning a TimesTen database, including tuning the database itself, SQL tuning, Transaction tuning, Cache and Replication tuning, and Materialized View tuning. Again, this is an extensive list, so it doesn't make sense to dump it all in this document. Some of the most obvious tuning include: the correct configuration of checkpointing and logging, providing the database enough memory to work with, and avoiding connection overhead by ensuring the database always remains in memory. Other features about the design of the database include the fact that it only supports two isolation levels. Because read-committed isolation is less restrictive than repeatable-reads and serializable isolation, it means that OLTP operations can occur with less blocking. The use of MVCC also contributes to OLTP performance. As an in-memory database, there is less use of I/O with disk meaning that data access and writing is much faster. Again, this helps to improve OLTP performance. See for further details: https://docs.oracle.com/database/timesten-18.1/TTOPR/perform.htm#TTOPR783
Indexes	There are two types of Indexing that TimesTen uses in query execution, both pre-existing and temporary indexes that are generated during the optimization phase of query execution. 1. Hash Indexes: A hash index is better for equality searches. They are faster for exact match lookups but are not as space efficient, and can only be used for finding exact values, and are not useful in sorting results from a table-scan. 2. Range (B+-Tree) Indexes: Range-indexes as their name suggests are better for finding rows where a certain column value falls within a specified range. They are optimized for in-memory data management. Worth noting that TimesTen originally used T-Tree indexes. Indexes can be specified as unique, meaning that each value of a column is unique. TimesTen can also create temporary indexes of both types during the processing stage of query execution to improve performance.

	https://docs.oracle.com/database/timesten-18.1/TTOPR/comp.htm#TTOPR379 https://docs.oracle.com/database/timesten-18.1/TTCIN/query.htm#TTCIN186 https://docs.oracle.com/cd/E17952_01/mysql-5.7-en/index-btree-hash.html https://docs.oracle.com/database/timesten-18.1/TTOPR/comp.htm#TTOPR380
Database Interfaces to other languages and tools	Connection APIs: JDBC ODBC ODP.NET Oracle Call Interface (OCI) Languages: C C++ Java PL/SQL Tools: SQL Developer https://db-engines.com/en/system/TimesTen
Database Logging	As mentioned in the Persistency section, TimesTen performs logging on transactions that occur within the database and persists these logs to disk. However, logs are not generated for read-only transactions because they do not modify the data in any capacity, so it is not necessary information when performing a recovery. Logs are purged regularly by TimesTen, typically when a new checkpoint file is generated. https://docs.oracle.com/database/timesten-18.1/TTOPR/trans.htm#TTOPR681
Autonomous/Serverless	Could not find information indicating support for autonomous/serverless deployments.
Machine Learning	No information supplied from vendor about Machine Learning capabilities, no other resources displayed its use as a machine learning database. Mostly people used it as a cache, as it is optimized for OLTP. It would not be great for handling large amount of data for analytical purposes, RAM is expensive.
Data Compression	For efficient data storage in memory, and on disk, TimesTen can perform table compression through columns, both individual columns and column groups. Essentially unique values of a column are stored within a separate dictionary table, with the values in the original table column replaced with pointers to the key in the dictionary table where is original value is kept. https://docs.oracle.com/database/timesten-18.1/TTOPR/perform.htm#TTOPR411
Documentation	https://docs.oracle.com/database/timesten-18.1/index.htm
Training Materials/Resources	Oracle VM supplied with hands on labs. https://www.oracle.com/ie/database/technologies/timesten-vm.html TimesTen Tutorial and Quick Guide:

	https://www.oracle.com/database/technologies/timesten-tutorials-demos.html			
Source Code/GitHub	N/A Proprietary			
Links	GitHub Sample Code: https://github.com/oracle-samples/oracle-timesten-samples			
References	https://docs.oracle.com/database/timesten-18.1/index.htm https://docs.oracle.com/database/timesten-18.1/TTOPR/perform.htm#TTOPR411 https://docs.oracle.com/database/timesten-18.1/TTCIN/query.htm#TTCIN186 https://docs.oracle.com/database/timesten-18.1/TTPLS/overview.htm#TTPLS118 https://www.oracle.com/a/tech/docs/wp-timesten-exalogic-2215631.pdf https://blogs.oracle.com/timesten/post/introduction-to-oracle-timesten-in-memory-database https://docs.oracle.com/en/database/other-databases/timesten/22.1/licensing/licensing-information.html https://docs.oracle.com/cd/E11882_01/timesten.112/e21633/using.htm#TTOPR120 https://docs.oracle.com/cd/E18283_01/timesten.112/e13065/using.htm https://docs.oracle.com/cd/E17952_01/mysql-5.7-en/index-btree-hash.html https://www.geeksforgeeks.org/cost-based-optimization/ https://dbdb.io/db/timesten https://skylandtech.net/category/database-vendors/oracle/timesten/ http://luna-ext.di.fc.ul.pt/oracle11g/timesten.112/e13065/trans.htm https://datacadamia.com/db/timesten/timesten https://docs.rackspace.com/blog/introduction-to-oracle-times-ten/ https://dbdb.io/db/timesten/revisions/10			

Snowflake …

Database	**Snowflake**		
Licence Type(s)	On Demand: Usage-based pricing with no long-term licensing requirements. Capacity: Discounted pricing based on an up-front Capacity commitment.		
Original Developer(s)	Benoit Dageville, Thierry Cruanes and Marcin Żukowski	Date/Year Created	July 23rd 2012
Developed in Country	San Mateo, California, United States		
Current Owner	Benoit Dageville, Thierry Cruanes and Marcin Żukowski	Owner Since	2012
Owner Country	United States		
Previous Names (& Year)	Snowflake Inc. 2012		
Websites (Company & Product)	https://www.snowflake.com/en/		
Database Category/Type	Snowflake is a relational database system that uses Snowflake Schema, where low cardinality multiplicities divide into separate tables.		
Licence Types	On Demand: Usage-based pricing with no long-term licensing requirements.		
	Capacity: Discounted pricing based on an up-front Capacity commitment.		
Derivative Systems	Based on SQL-92 ANSI Information Schema		
Database Description	The snowflake schema is represented by centralized fact tables which are connected to multiple dimensions. "Snowflaking" is a method of normalizing the dimension tables in a star schema. When it is completely normalized along all the dimension tables, the resultant structure resembles a snowflake with the fact table in the middle. The principle behind snowflaking is normalization of the dimension tables by removing low cardinality attributes and forming separate tables.		
History			
Written in what language	Developed in C++ and Java		
Supported Languages	Go, Java, .NET, Python, C, Node.js, php, python,		
Operating Systems	Cloud based system. Hosted on one of three platforms: Amazon S3, Microsoft Azure, Google Cloud		
Support Platforms	Used on cloud platforms; Amazon S3, Microsoft Azure, Google Cloud		
Isolation Levels	Single Isolation level is supported in Snowflake, READ COMMITTED.		
Concurrency Control	Multi-cluster warehouses with a maximum of 10, including an option of automatic concurrency scaling.		

Persistence Model(s)	Object oriented
Data Models	Snowflake Schema
Language Capabilities	Complete ANSI SQL language support
Stored Procedures/Functions	Functions/Procedures can be written and stored on the web API. External functions and procedures can be used. Python, Spark, Kafka, NodeJS, Go, .NET, JDBC, ODBC, PHP and SQL API prebuild drivers and connectors can be used.
Database & Query Optimiszation	Query optimization is done internally, cannot be accessed.
Database Performance Features	Black box system, the performance cannot be manipulated by users.
Indexes	Indexing is not available.
Database Interfaces to other languages and tools	SnowPipe, enables loading and unloading data from other platforms. Snowflake supports development with supported languages.
Database Logging	Users can view the entire activity taken over database from query history, copy history or task history.
Autonomous/Serverless	Snowflake support autonomous SQL transaction.
Machine Learning	Snowflake has been created to support machine learning, so it has many tools.
Data Compression	Four data compression methods can be used: SNAPPY, ZLIB, ZSTD and BZ2.
Documentation	Documentation book/pdf with everything inside and a web documentation.
Training Materials/Resources	Give links and details of what training materials are available. These can include videos, tutorials, training courses (by vendor and others), etc
Links	Other product related links and a short description for each link.
Source Code/GitHub	Given links to the Source Code (GitHub etc), and a short description for each link.
Links	https://www.snowflake.com/en/

References	[1] - Snowflake Inc.. (2022, October 22). Retrieved November 9, 2022, from https://en.wikipedia.org/wiki/Snowflake_Inc. [2] - Engines ranking. (n.d.). Retrieved November 9, 2022, from https://db-engines.com/en/ranking [3] - Cloud computing. (2022, November 08). Retrieved November 9, 2022, from https://en.wikipedia.org/wiki/Cloud_computing [4] - Data as a Service. (2022, October 12). Retrieved November 9, 2022, from https://en.wikipedia.org/wiki/Data_as_a_service [5] - Native programmatic interfaces¶. (n.d.). Retrieved November 9, 2022, from https://docs.snowflake.com/en/user-guide/ecosystem-lang.html [6] - Snowflake information schema¶. (n.d.). Retrieved November 9, 2022, from https://docs.snowflake.com/en/sql-reference/info-schema.html [7] - Key Concepts & Architecture¶. (n.d.). Retrieved November 9, 2022, from https://docs.snowflake.com/en/user-guide/intro-key-concepts.html [8] - Gigoyan, S. (2022, May 22). Snowflake transactions vs SQL Server Transactions. Retrieved November 9, 2022, from https://www.mssqltips.com/sqlservertip/7257/snowflake-transactions-vs-sql-server-transactions/ [9] - Transactions¶. (n.d.). Retrieved November 9, 2022, from https://docs.snowflake.com/en/sql-reference/transactions.html#isolation-level [10] - Multi-cluster warehouses¶. (n.d.). Retrieved November 9, 2022, from https://docs.snowflake.com/en/user-guide/warehouses-multicluster.html [11] - Snowflake community. (n.d.). Retrieved November 9, 2022, from https://community.snowflake.com/s/article/Putting-Snowflake-s-Automatic-Concurrency-Scaling-to-the-Test [12] - Four keys to success with snowflake. (n.d.). Retrieved November 9, 2022, from https://www.slalom.com/insights/snowflake-implementation-success#:~:text=It%20provides%20support%20for%20programming,day%2Dto%20%2Dday%20operations. [13] - Snowflake schema. (n.d.). Retrieved November 9, 2022, from https://www.sciencedirect.com/topics/computer-science/snowflake-schema [14] - The data cloud. (n.d.). Retrieved November 9, 2022, from https://www.snowflake.com/en/ [15] - Compress¶. (n.d.). Retrieved November 9, 2022, from https://docs.snowflake.com/en/sql-reference/functions/compress.html [16] - Machine learning tools. (n.d.). Retrieved November 9, 2022, from https://www.snowflake.com/trending/machine-learning-tools [17] - Transactions¶. (n.d.). Retrieved November 9, 2022, from https://docs.snowflake.com/en/sql-reference/transactions.html			

MySQL Heatware

Database	MySQL HeatWave		
Licence Type(s)	Copyright © 1997, 2022, Oracle and/or its affiliates		
Original Developer(s)	A team led by Nipun Agarwal	Date/Year Created	December 2020
Developed in Country			
Current Owner	Oracle Corp (ORCL. N)	Owner Since	December 2020
Owner Country			
Previous Names (& Year)	No previous names are available. The service was named MySQL Heatwave.		
Websites (Company & Product)	This service was developed by Oracle.		
Database Category/Type	In-memory Query accelerator with Built-in ML		
Licence Types	MySQL Heatwave is not open source. It is protected under Copyright © 1997, 2022, Oracle and/or its affiliates. The software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws.		
Derivative Systems	MySQL Autopilot was developed for the MySQL Heatwave service to improve performance, scalability, and uptime while reducing manual database administration tasks		
Database Description	MySQL Heatwave is designed to increase the performance of MySQL Database service for analytics and mixed workloads. It eliminates the need for a separate Analytics database, separate Machine Learning (ML) tools, and extract, transform, and load (ETL) duplication. MySQL HeatWave is available on Oracle Cloud Infrastructure (OCI), Amazon Web Services (AWS), and Microsoft Azure.		
History	The service was released to the public on December 2020. Following are the version releases for Heatwave HeatWave 8.0.31 (2022-10-11, General Availability) HeatWave 8.0.30-u1 (2022-09-06, General Availability) HeatWave 8.0.30 (2022-07-26, General Availability) HeatWave 8.0.28-u3 (2022-04-19, General Availability) HeatWave 8.0.28-u2 (2022-03-29, General Availability) HeatWave 8.0.28-u1 (2022-02-15, General Availability) HeatWave 8.0.27-u3 (2021-12-15, General Availability) HeatWave 8.0.27-u2 (2021-12-07, General Availability) HeatWave 8.0.26-u2 (2021-09-21, General Availability) HeatWave 8.0.26-u1 (2021-08-10, General Availability) HeatWave 8.0.26 (2021-07-23, General Availability) HeatWave 8.0.25 (2021-05-11, General Availability) HeatWave 8.0.24 (2021-04-20, General Availability) HeatWave 8.0.23-u2 (2021-03-15, General Availability) HeatWave 8.0.23-u1 (2021-02-09, General Availability)		

Written in what language	No details are available about MySQL HeatWave. InnoDB, the storage engine used by MySQL HeatWave is written in C language.
Supported Languages	SQL, Python, R, Scala
Operating Systems	
Support Platforms	Available only on the cloud via OCI (Oracle Cloud Infrastructure), AWS (Amazon Web Services), and Microsoft Azure
Isolation Levels	Isolation levels are not provided by MySQL HeatWave but instead InnoDB provides the following isolation levels since InnoDB is the storage engine used hand-in-hand with MySQL HeatWave. InnoDB offers all four transaction isolation levels described by the SQL:1992 standard: READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, and SERIALIZABLE. The default isolation level for InnoDB is REPEATABLE READ.
Concurrency Control	MySQL Heatwave offers auto thread pooling which queues incoming transactions to give sustained throughput during high transaction concurrency. Where multiple clients are running queries concurrently, Auto Thread Pooling applies workload-aware admission control to eliminate resource contention caused by too many waiting transactions. Auto Thread Pooling automatically manages the settings for the thread pool control variables.
Persistence Model(s)	Since MySQL DB and InnoDB are relational database services they persist data in the form of records and tables. C++, Java can be used to define and persist entity objects and their relationships.
Data Models	EER (Enhanced Entity Relationship) Models can be created using the MySQL Database Service. With this feature, you can plan databases more thoroughly by delving into the properties and constraints with greater precision.
Language Capabilities	The Database uses SQL primarily for querying and CRUD operations. Python, R, Scala are used by Heatwave ML for running ML algorithms via Jupyter or Apache Zeppelin.
Stored Procedures/Functions	Stored procedures and functions are supported in MySQL Cluster. COBOL, C++, and Java can be used to define stored procedures and functions.
Database & Query Optimisation	MySQL Autopilot enables database and query optimization with auto query plan improvement and auto query time estimation. The service estimates/predict the time it takes to run a certain query before executing the query which helps the customer to decide whether to run the query. It also learns various statistics from the historical execution of queries and improves the query plan for future queries.
Database Performance Features	MySQL Autopilot is a feature that enables many additional features which make MySQL Heatwave function autonomously without any user intervention. Features like – auto-provisioning, auto parallel load, auto data placement, auto encoding, auto query plan improvement, auto query time estimation, auto change propagation, auto-scheduling, and auto error recovery.

Indexes	Indexing is not required in MySQL Heatwave as MySQL autopilot features auto data placement which enables partitioning of the data based on the query beforehand to optimize query execution time.
Database Interfaces to other languages and tools	MySQL Heatwave supports tools like Apache Zeppelin and Jupyter for the Machine Learning service.
Database Logging	Database logging is available in InnoDB, which is called as undo and redo logs. The redo log is physically represented on disk by redo log files. An undo log is a collection of undo log records associated with a single read-write transaction. Using these undo and redo logs the database keeps track of the records affected by any alteration/modification.
Autonomous/Serverless	Yes, most of the optimizations are autonomous and handled by MySQL Heatwave via the MySQL Autopilot service.
Machine Learning	Yes, the service offers native in-database machine learning. The users can build ML Models using SQL commands. They can also automate the ML lifecycle.
Data Compression	HeatWave compresses data as it is loaded, which permits HeatWave nodes to store more data. It dynamically allocates the data to the nodes to reduce the cost by minimizing the size of the Heatwave Cluster required to store your data. Decompression operations that occur as data is accessed affect performance to a small degree
Documentation	https://dev.mysql.com/doc/heatwave/en/ - This guides the users/developers on operating and navigating the MySQL Heatwave Analytics software service.
Training Materials /Resources	1. https://dev.mysql.com/doc/heatwave/en/heatwave-analytics.html - Introduction and user guide related to MySQL heatwave service 2. https://dev.mysql.com/doc/heatwave/en/heatwave-machine-learning.html - Introduction and user guide related to the MySQL heatwave Machine Learning facility 3. https://dev.mysql.com/doc/refman/8.0/en/innodb-storage-engine.html - User Guide for InnoDB Storage engine 4. https://www.oracle.com/ie/mysql/heatwave/#:~:text=Nipun%20Agarwal%2C%20Oracle%20senior%20vice,time%20for%20developers%20and%20DBAs. – Heatwave video demos
Links	
Source Code/GitHub	https://github.com/oracle/heatwave - Codebase for MySQL heatwave benchmarking
Links	

References	1. https://dev.mysql.com/doc/refman/8.0/en/innodb-transaction-isolation-levels.html 2. https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/CHAP_SettingUp_Aurora.html 3. https://dev.mysql.com/doc/heatwave/en/ 4. https://www.oracle.com/ie/mysql/heatwave/ 5. https://blogs.oracle.com/mysql/post/mysql-autopilot-machine-learning-automation-for-mysql-heatwave 6. https://www.oracle.com/ie/a/ocom/docs/mysql/mysql-heatwave-on-aws-brief.pdf - Heatwave on AWS User Guide 7. https://downloads.mysql.com/docs/heatwave-en.pdf - MySQL Heatwave User Guide 8. https://www.oracle.com/a/ocom/docs/mysql-heatwave-technical-brief.pdf - MySQL Heatwave Technical Brief 9. https://aws.amazon.com/rds/features/ - Amazon RDS features

MariaDB …

Database	Maria DB		
Licence Type(s)	GPL 2.0, mostly open source		
Original Developer(s)	Michael “Montey” Widenius, the founder of MySql and a group of developers which formed after the purchase of MySQL by Oracle and David Axmark	**Date/Year Created**	29th October 2009
Developed in Country	Sweden		
Current Owner	Michael Howard, CEO of MariaDB Corporation, Kaj Arnö, CEO of MariaDB Foundation	**Owner Since**	Dec 2015 Jan 2019
Owner Country	Redwood California, United States		
Previous Names (& Year)	MySQL (first release in 1995 - 2009)		
Websites (Company & Product)	MariaDB-Foundation: https://mariadb.org/ MariaDB-Corporation: https://mariadb.com/ Knowledgebase: https://mariadb.com/kb/en/		
Database Category/Type	Relational Database Management System (RDBMS) ⇨ Table based data management		
Licence Types	GNU GPL 2 license		
Derivative Systems	MariaDB is a fork of the MySQL Database. It is still compatible with most of the MySQL features and the MySQL language.		
Database Description	MariaDB is in its core an open-source relational database management system. There are enterprise, cloud, and scale versions available.		
History	• MariaDB is a fork of the MySQL company. The split happened because of the acquisition of MySQL by Sun which later got purchased by Oracle • In February 2009 Monty & others left Sun to work on Maria engine in Monty Program Ab • Oracle acquired Sun and so MySQL in April the same year • The split was accompanied by the slogan: “Save the People, Save the Product”, because they wanted to keep the code under a free licence and to ensure there would always be a free version existing		

	• Montey describes his motivation as giving back to the coding community, because the team also used open-source code for their project • To ensure that there would always be an open version the MariaDB Foundation was created in 2010, so everybody could contribute to the project on equal terms • The foundation is apart from the cooperation so there is no influence on each other
Written in what language	C and C++
Supported Languages	Java, Perl, C, C++ Mostly used are Java and C (ca. 90%)
Operating Systems	Linux, Windows, macOs
Support Platforms	SkySQL is the built-in cloud solution
Isolation Levels	• MariaDB is using GALERA clusters ◦ synchronous multi-master database cluster So, the following isolation levels are supported: • READ-UNCOMMITTED ◦ see changes of transactions which have not been committed → no real isolation level • READ-COMMITTED ◦ dirty reads are impossible • REPEATABLE-READ ◦ Select always return the same query result • SERIALZABLE is only honoured between transactions on the same node (should be avoided) ◦ all records accessed by a transaction are locked
Concurrency Control	Maria DB offers full ACID-Control • Atomicity: entire transaction must complete, so there are no unfulfilled transactions • Consistency: Data must be consistent thought the database and transactions • Isolation: Any data being used during the processing of one transaction cannot be used by another transaction until the first transaction is complete • Durability: Once data from a transaction has been committed, its effects will remain
Persistence Model(s)	• Primary key is necessary due to underlying InnoDB storage engine • Also mandatory for efficient clustering https://vettabase.com/blog/why-tables-need-a-primary-key-in-mariadb-and-mysql/

Data Models	What data models or combinations of models are supported Relational Data Model: • Data is organized in tables • Data is identified via primary keys • Key constrains display relationships between data
Language Capabilities	- Standard SQL: - NoSQL: Offers some statements which are close to NoSQL standard statements like HANDLER - SkySQL: Connectors: Java, C++, C, node.js, ODBC and Python
Stored Procedures/Functions	Maria DB offers the opportunity to create stored procedures with SQL and stored functions Stored procedure: Routine invoke with a CALL statement Stored function: Defined functions for use with SQL-statements Stored routine: SQL statements related to creating and stored routines Maria DB has various connectors e.g. Python so therefore there is the opportunity to use it with another language than SQL
Database & Query Optimiszation	Maria DB Knowledgebase offers various articles about the different types and strategies of optimization. Underneath there is an extract of the general information to optimize with MariaDB Storage Engine: InnoDB ⇨ partition data on faster drives ⇨ increase/decrease query cache size Query optimization: - Normalize tables - Use the right data type - Indexes - 'Explain' keyword: See querys behaviour and get the information of how the statement is operating behind the scenes Subquery optimization
Database Performance Features	Details to the database are held in the my.cnf file There are also performance enhancements possible when accessing the storage engine directly -> InnoDB as a default and Aria for temporal files. Parameters to improve performance: • InnoDB file-per-table • InnoDB Buffer Pool Size • Disable Swapiness In MySQL • Max Connections

	- Thread Cache Size - Disable MySQL DNS Lookups - Query Cache Size - Tmp Table Size & Max Heap Table Size - Slow Query Logs - Idle Connections
Indexes	- MariaDB create index - MariaDB index types - MariaDB create index multiple columns - MariaDB create index if not exists - MariaDB create index no lock - MariaDB create index using btree - MariaDB index json or json column - MariaDB index column - MariaDB index hint - MariaDB index statistics - MariaDB index on view - MariaDB index length and index size https://databasefaqs.com/mariadb-index/
Database Interfaces to other languages and tools	There are 12 different connectors: - C & C++ Connector - Java Connector - .NET Connector - Node.js Connector - ODBC Connector - Perl DBI - PHP - Python - Ruby - Erlang - The MariaDB Jupyter Kernel - Other Connectors (Excel, Swift, R) https://mariadb.com/kb/en/connectors/
Database Logging	In MariaDB 10.4 and later, the following server-side authentication plugins are installed by default: - The mysql_native_password and mysql_old_password authentication plugins authentication plugins are installed by default in all builds. - The unix_socket authentication plugin is installed by default in all builds on Unix and Linux. - The named_pipe authentication plugin is installed by default in all builds on Windows. Plug-ins for server-side :

	- mysql_native_password - mysql_old_password - ed25519 - gssapi - pam (Unix only) - unix_socket (Unix only) - named_pipe (Windows only) Client-side authentication plug-ins: - mysql_native_password - mysql_old_password - client_ed25519 - auth_gssapi_client - dialog - mysql_clear_password - sha256_password - caching_sha256_password https://mariadb.com/kb/en/pluggable-authentication-overview/
Autonomous/Serverless	A way to access and secure your datas efficiently and with security : https://mariadb.com/products/enterprise/xpand/ This use replica like we can find in RAID-4 (a storage technologies).
Machine Learning	It supports machine learning. https://mariadb.com/kb/en/machine-learning-with-mindsdb/ Minidsdb is a third-party application that interfaces with MariaDB Server to provide Machine Learning capabilities through SQL. The interface is done via the Connect Storage Engine. Usage: Take a JSON configuration file. Create Table who fit with a csv file you want to learn with. LOAD DATA INFILE <file_path>
Data Compression	Compressed row format (with block as we see in lecture) from 1 to 16 kb. Supported on Barracuda file format. ImmoDB : https://mariadb.com/kb/en/innodb-page-compression/ none, zlib, lz4, lzo, lzma, bzip2, snappy.
Documentation	For Cloud : https://mariadb.com/products/skysql/docs/quickstart/install-mariadb/
	From normal installation : https://mariadb.org/download/ follow steps on this site : https://www.mariadbtutorial.com/getting-started/

	To understand features of mariadb : https://mariadb.com/kb/en/documentation/ Release notes : https://mariadb.com/kb/en/mariadb-server-release-dates/ Read features' tutorial from beginners to advanced users : https://mariadb.com/kb/en/training-tutorials/ All documentation about Entreprise Server : https://mariadb.com/docs/ MaxScale is a database proxy that extends the high availability, scalability, and security of MariaDB Server while at the same time simplifying application development by decoupling it from underlying database infrastructure : https://mariadb.com/kb/en/maxscale/ ColumnStore is a columnar storage engine that utilizes a massively parallel distributed data architecture : https://mariadb.com/kb/en/mariadb-columnstore/ To find every connectors from other language: https://mariadb.com/kb/en/connectors/ The cloud database service (DBaaS) SkySQL : https://mariadb.com/products/skysql/docs/			
	Docker installation find MariaDB image : docker pull mariadb Start a container called "mariadbtest" on 3306 port: docker run --name mariadbtest -e MYSQL_ROOT_PASSWORD=mypass -p 3306:3306 -d docker.io/library/mariadb See all your containers running : docker ps Start the cmd : docker exec -it mariadbtest mysql -uroot -p (password here is mypass)			
Training Materials/Resources	We can find 3 Books, 4 articles (Beginner, Basic, Intermediate, Advanced). Every article that looks like notebooks to learn and try by yourself. You can find everything here: https://mariadb.com/kb/en/training-tutorials/			
Links	Other product related links and a short description for each link.			
Source Code/GitHub	From github repository:			

	https://github.com/MariaDB/server `git clone https://github.com/MariaDB/server.git`			
Links	Other product related links and a short description for each link.			
References	List references used to complete this report and your research. This should be addition to those listed in previous sections. • https://mariadb.com/kb/en/acid-concurrency-control-with-transactions/ • Anel Husakovic, 2021. Installing plugins in the MariaDB Docker Library Container: https://mariadb.org/installing-plugins-in-the-mariadb-docker-library-container/ • Amit Verma, 2018, MySql Joins – LEFT JOIN, RIGHT JOIN, INNER and OUTER JOIN: https://www.beingidea.com/mysql-left-join-and-right-join-inner-join-and-outer-join/ • Budima, U. H. a. E., 2021. Inner Join Performance: MariaDB vs. PostgreSQL. Journal of Physics: Conference Series, 1844(2020 2nd ICoST). • Codd, E., 1981. Relational Database - A Practical Foundation of Productivity. ACM Turing Award Lecture, 9 November, pp. 109-117.			

Amazon Athena …

Query Engine	**AWS Amazon Athena**		
Licence Type(s)	Cloud Licensing is used for Amazon Athena as AWS manages everything. The user can just easily log in to their account on the AWS management console, and use Athena along with other AWS cloud-based products such as S3, Glue, Quicksight.		
Original Developer(s)	Amazon Web Services	Date/Year Created	2016
Developed in Country	Unites States of America		
Current Owner	Amazon Web Services	Owner Since	2016
Owner Country	Unites States of America		
Previous Names (& Year)	Amazon Athena did not have a previous name; it was introduced as Amazon Athena from the beginning. No reports online were found in regards to initial names before Athena.		
Websites (Company & Product)	https://aws.amazon.com/athena/?nc=sn&loc=0&whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc		
Database Category/Type	Athena is categorised as a query engine service, and **not a database**. Athena is not a stand-alone product and cannot be used without a database connected to it. Data is stored in a database like S3, and computational queries are run by Athena against it.		
Derivative Systems	Amazon Athena is based on Presto. It has the standard functionalities of Presto, but Athena is behind a couple of versions. Presto is currently on 0.277, and Athena is on 0.217. Although the AWS states that the latest update to Athena version 3 is relatively up to date with Presto's current version. However, research has found no information on which version of Presto is Athena currently integrating in Athena version 3. However, it does say that the current version 3 implements Trino functions. But like the above, there is no further information on which Trino version is being implemented.		
Database Description	Amazon Athena is an interactive, serverless query engine that uses SQL language to make up these queries. It is mainly used to query AWS S3, an AWS database. However, Amazon Athena can also be integrated with other data sources by using data source connectors. Athena can query structured, semi-structured, and unstructured data, such as standard data formats like JSON, CSV, columnar formats. It is a pay-by-query service, in which you are charged for successful queries. The underlying cost comes from the amount of bytes of data are scanned by each query.		
History	Athena was first released in 2016. Athena engine version 2 was released in 2020, and version 3 was announced last month. Over the past 6 years, Athena has integrated many more features. When Athena was first released it was only able to analyse data in S3. Version 2 made it possible for users to connect to many more data sources in the backend of Athena. It has also evolved with PrestoDB, as this is what Athena relies on under the hood.		

Written in what language	Amazon Athena is based on Facebook's Presto, another SQL query engine. Presto was written in the language Java. Although there is no information on which language Athena is written in, we can assume that it was with Java due to its dependency of Presto.
Supported Languages	Structured Query Language (SQL) by IBM, is generally used in relational database management systems. It is used to manage databases and perform query operations on data.
Operating Systems	Amazon Athena is compatible with all operating systems (Windows, MacOS, Linux etc.) as it is a web-based service. There is no need to download anything, it can be accessed through their web console – AWS Management Console
Support Platforms	(cloud, on-premises, Hybrid, etc) Amazon Athena is a serverless query engine connecting to a cloud-based storage and infrastructure to operate. AWS has a whole cloud computing ecosystem, and Athena is a component of that. This makes it easier to access and use these products, in particular Athena as there was no set up whatsoever when you are connecting to AWS S3 database, and using AWS Glue to bridge the gap.
Isolation Levels	What isolation levels are support by the database. Athena supports ACID transactions. However, there is no further information if all isolation levels are supported. What our research found was that Athena fully supports ACID, therefore we can assume it supports all 5 isolation levels; Read Uncommitted, Read Committed, Repeatable Read, Serializable, Snapshot. It uses Apache Iceberg to implement ACID transactions.
Concurrency Control	What concurrency control is support by the database AWS limits how much data, query power and concurrent queries you can run in Athena. Each account is limited to 100 databases (where each database can only have 100 tables), 20 concurrent queries and 5500 S3 requests per second. Queries also time out after 30 mins.
Persistence Model(s)	What persistence models are supported by the database Athena uses S3 as its underlying storage engine. A query output location in S3 must be specified before you run your query. Athena caches all query results in this output location, such as `s3://path/to/query/bucket/.` Athena automatically saves the output files from each query that runs. Users can download these files from the console.
Data Models	What data models or combinations of models are supported As Athena is not a database, but only a query engine, it does not necessarily follow a particular data model. It does, however, support multiple data formats from structured, semi-structured, and unstructured data
Language Capabilities	That is the main language used to interact with the database. How extensive is the implementation of the language? What language standards are supported. What things does it do differently? Structured Query Language (SQL) is used to make queries in Athena. However, not the full and updated SQL functionalities are implemented. It supports basic functions and statements of Data Definition Language (DDL) and Data Manipulation Language (DML).

Stored Procedures/Functions	User Defined Functions (UDF) are supported in Athena, in which users can create and reuse functions. Examples are compressing or decompressing data. UDFs are fuelled by AWS Lambda (a serverless computing platform by Amazon) and can be made with Java.
Database & Query Optimisation	How does the database manage database and query optimization? Athena relies on partitioning and compression for data optimisation. These techniques reduce the amount of data that is scanned per query. Storing files in parquet format means data is compressed. Parquet format also allows column label filtering - Athena only has to scan columns that are needed. Partitioning breaks the file down into smaller files. This reduces row level scan because the query will only scan the file that it needs.
Database Performance Features	What performance features exist in the database to support better performance. Athena is a distributed query engine. It uses S3 as its underlying storage engine. Unlike ETL tools / full database products, Athena does not have its own optimized storage layer. Its performance features depend on S3. E.g., queries can be very slow if data is not sorted in S3 to allow for metadata-based filtering.
Indexes	What indexing methods are available? Athena has no indexes. No indexing methods available. It performs full table scans instead of relying on indexes. Because there are no indexes, you cannot optimise data – you can only optimise queries.
Database Interfaces to other languages and tools	What interfaces does the database support to other languages, tools and applications? There are 3 ways to connect to Athena. 1. Through the console. 2. Through the JDBC and ODBC driver. 3. Through an API. The JDBC/ODBC driver connection allow Athena to support other reporting/visualisation tools. E.g., Tableau uses ODBC Driver in order to access data in Athena.
Database Logging	What database logging methods are used to support database operations. Database logging is a background process. AWS CloudTrail stores API activity and calls from the console. It captures the request that was made to Athena, the IP address from which the request was made, who made it and when. You can query CloudTrail logs using Athena. Athena also publishes metrics to Amazon CloudWatch for each workgroup (check is enabled). These include: QueryQueueTime, QueryPlanningTime, TotalExecutionTime and more. You can check these metrics in Athena. You can also configure rules for Cloudwatch log groups.
Autonomous/Serverless	Can the database operate in an autonomous or serverless mode. If so, give details. Athena has a serverless architecture. PrestoDB is used to perform queries. Athena deploys Presto (an Apache project) without the need to oversee infrastructure or spin servers. All SQL queries in Athena use Presto, an Apache project that developers use to query data stored in Amazon S3 using standard SQL.
Machine Learning	Does the Database support Machine Learning? What features exists to support machine learning. Version 2 introduced Amazon Sagemaker. This brought Machine Learning capabilities to Athena. Users can deploy all the models in Amazon Sagemaker for inference in SQL Queries. The USING EXTERNAL FUNCTION deploys the Sagemaker

	model that you want. You can run inference on data on any data source, it does not matter whether the data contains inherent analysis or not.
Data Compression	Details of what data compression techniques are available in the database. Data compression speeds up queries in Athena. When you use files in Apache Parquet or Apache ORC format, these are compressed by default in Athena. Compression reduces the file size. This has significant benefits in Athena because it means you scan less data and use less storage.
Documentation	What documentation exists for the database. Give link and a short description. https://docs.aws.amazon.com/athena/latest/ug/what-is.html This is the document of Amazon Athena, containing the steps to getting started, linking to data sources, creating tables, querying, and troubleshooting. The document has all the relevant information to use Athena.
Training Materials/Resources	Give links and details of what training materials are available. These can include videos, tutorials, training courses (by vendor and others), etc Athena overview – Deep Dive - https://www.youtube.com/watchv=tzoXRRCVmIQ&ab_channel=AWSEvents What's new in Athena - https://www.youtube.com/watchv=k9kHVj1dxaA&ab_channel=AWSEvents Athena tutorial - https://youtu.be/ACkLSdzHJLU Athena, S3, Glue tutorial - https://youtu.be/haLRGmMTOtY Querying S3 Server Access Logs - https://youtu.be/uDVhx2IO9WE Data Preparation for Athena - https://youtu.be/Dmw7HOOmiJQ
Links	Other product related links and a short description for each link. Using Quicksight with Athena – this shows the integration between Athena and quicksight https://youtu.be/pcl6ObEEaU8 What is AWS Glue – explains the role of AWS glue, and how it is integrated to other AWS products https://youtu.be/qgWMfNSN9f4 Quicksight with Athena - https://youtu.be/8AMagSTe0l8
Source Code/GitHub	Given links to the Source Code (GitHub etc), and a short description for each link. AWS Athena query federation source code on GitHub - https://github.com/awslabs/aws-athena-query-federation

	This will allow users to customise Amazon Athena to suit their architecture or technical needs. Users can link multiple data sources, create UDFs and more.
References	List references used to complete this report and your research. This should be addition to those listed in previous sections. Access through JDBC and ODBC connections - Amazon Athena. (2022). Retrieved November 11, 2022, from Amazonaws.cn website: https://docs.amazonaws.cn/en_us/athena/latest/ug/policy-actions.html Ahana and AWS | Ahana | Ahana. (2021, May 20). Retrieved November 9, 2022, from Ahana website: https://ahana.io/aws/ Ahana Cloud. (2022, August 4). Why Data Engineers Are Moving Away From AWS Athena - Ahana Cloud - Medium. Retrieved November 7, 2022, from Medium website: https://medium.com/@ahana.io/why-data-engineers-are-moving-away-from-aws-athena-3d2b0abca580 Amazon Athena - Big Data Analytics Options on AWS. (2022). Retrieved November 7, 2022, from Amazon.com website: https://docs.aws.amazon.com/whitepapers/latest/big-data-analytics-options/amazon-athena.html Amazon Athena FAQs – Serverless Interactive Query Service – Amazon Web Services. (2022). Retrieved November 10, 2022, from Amazon Web Services, Inc. website: https://aws.amazon.com/athena/faqs/ Amazon Web Services. (2018). AWS Knowledge Center Videos: How do I analyze my S3 logs using Athena? [YouTube Video]. Retrieved from https://www.youtube.com/watch?v=uoLsrKZha0E&ab_channel=AmazonWebServices Amazon Athena Pricing – Serverless Interactive Query Service – Amazon Web Services. (2022). Retrieved November 6, 2022, from Amazon Web Services, Inc. website: https://aws.amazon.com/athena/pricing/ Analyzing Amazon S3 server access logs using Amazon OpenSearch Service | Amazon Web Services. (2020, September 15). Retrieved November 7, 2022, from Amazon Web Services website: https://aws.amazon.com/blogs/big-data/analyzing-amazon-s3-server-access-logs-using-amazon-opensearch-service/#:~:text=Amazon%20S3%20lets%20you%20monitor,own%20in%20the%20same%20Region. Athena Basics: Running Queries | The Athena Guide. (2020). Retrieved November 10, 2022, from Athena.guide website: https://athena.guide/articles/athena-basics-running-queries/ Athena engine version 3 - Amazon Athena. (2020). Retrieved November 11, 2022, from Amazon.com website: https://docs.aws.amazon.com/athena/latest/ug/engine-versions-reference-0003.html Athena vs. Redshift Spectrum vs. Presto | Firebolt. (2022). Retrieved November 11, 2022, from Firebolt.io website: https://www.firebolt.io/blog/athena-vs-redshift-spectrum-vs-presto

AWS Athena Alternatives: Best Amazon Athena Alternatives. (2022, October 24). Retrieved November 7, 2022, from Ahana website: https://ahana.io/blog/aws-athena-alternatives/

AWS Athena engine V2 vs V1 - all the differences | Firebolt. (2022a). Retrieved November 11, 2022, from Firebolt.io website: https://www.firebolt.io/blog/amazon-athena-version-2-whats-new

AWS Athena Limitations. (2022, June 7). Retrieved November 11, 2022, from Ahana website: https://ahana.io/blog/aws-athena-limitations/

AWS Events. (2021). AWS re:Invent 2021 - What's new with Amazon Athena [YouTube Video]. Retrieved from https://www.youtube.com/watch?v=k9kHVj1dxaA&t=2289s

AWS Glue Data Catalog: Architecture, Components, Crawlers. (2022). Retrieved November 11, 2022, from Atlan website: https://atlan.com/aws-glue-data-catalog-explained/

AWS Tutorials. (2021). AWS Tutorials - Amazon Athena Query Cost Optimization [YouTube Video]. Retrieved from https://www.youtube.com/watch?v=hzRelR_GB4o&t=1036s

BryteFlow. (2020, July 21). Face off: AWS Athena vs Redshift Spectrum - BryteFlow. Retrieved November 9, 2022, from BryteFlow website: https://bryteflow.com/face-off-aws-athena-vs-redshift-spectrum/

burtcorp. (2020, July 30). GitHub - burtcorp/athena-jdbc: A JDBC driver for AWS Athena. Retrieved November 11, 2022, from GitHub website: https://github.com/burtcorp/athena-jdbc

Considerations and limitations for SQL queries in Amazon Athena - Amazon Athena. (2022). Retrieved November 10, 2022, from Amazon.com website: https://docs.aws.amazon.com/athena/latest/ug/other-notable-limitations.html

CloudZero. (2021). What Is AWS Athena? Here's Everything You Need To Know. Retrieved November 6, 2022, from Cloudzero.com website: https://www.cloudzero.com/blog/aws-athena

Derežić, D. (2020, June). Blog | Importance of Big Data: understanding the 5 Vs of big data. Retrieved November 10, 2022, from Bornfight website: https://www.bornfight.com/blog/understanding-the-5-vs-of-big-data-volume-velocity-variety-veracity-value/#:~:text=Big%20data%20value%20refers%20to,where%20data%20processing%20steps%20in.

Difference between Traditional data and Big data - GeeksforGeeks. (2020, September). Retrieved November 10, 2022, from GeeksforGeeks website: https://www.geeksforgeeks.org/difference-between-traditional-data-and-big-data/

Edureka. (2014, October 13). AWS Solutions Architect Certification Training Course. Retrieved November 5, 2022, from Edureka website: https://www.edureka.co/blog/understanding-amazon-s3/

Enabling Amazon S3 server access logging - Amazon Simple Storage Service. (2012). Retrieved November 7, 2022, from Amazon.com website: https://docs.aws.amazon.com/AmazonS3/latest/userguide/enable-server-access-logging.html

Getting Deterministic Performance Out Of Amazon Athena Guide | Ahana. (2021, March 6). Retrieved November 7, 2022, from Ahana website: https://ahana.io/answers/how-do-i-get-deterministic-performance-out-of-amazon-athena/

Levy, E. (2022, June 6). How to Improve AWS Athena Performance - Upsolver. Retrieved November 11, 2022, from Upsolver website: https://www.upsolver.com/blog/aws-athena-performance-best-practices-performance-tuning-tips

Jean-Christian LLOBET. (2016, June 28). The NoSQL databases implement various techniques to meet the need for even faster access to (big) data: the... Retrieved November 11, 2022, from Linkedin.com website: https://www.linkedin.com/pulse/asynchronous-vs-synchronous-api-nosql-databases-31-ratio-llobet

Luis Caro Perez. (2017, November 10). Visualize AWS Cloudtrail Logs using AWS Glue and Amazon Quicksight | Noise. Retrieved November 7, 2022, from Getoto.net website: https://noise.getoto.net/2017/11/10/visualize-aws-cloudtrail-logs-using-aws-glue-and-amazon-quicksight/

Meyer, R. (2022). Athena Error: Query exhausted resources at this scale factor. Retrieved November 8, 2022, from Firebolt.io website: https://www.firebolt.io/blog/aws-athena-error-query-exhausted-resources-at-this-scale-factor

Prakash, A. (2022, March 31). Use Amazon Athena Federated Query to query data from Aurora PostgreSQL running in Private Subnet. Retrieved November 11, 2022, from Medium website: https://awstip.com/use-amazon-athena-federated-query-to-query-data-from-aurora-postgresql-running-in-private-subnet-ae8d9441b1c8

Price-Performance Ratio of Athena vs Ahana. (2022, June 7). Retrieved November 9, 2022, from Ahana website: https://ahana.io/answers/how-does-the-price-performance-ratio-of-aws-athena-presto-compare-to-ahana-cloud-for-presto/

Query Presto Cluster with Apache Superset | Ahana Cloud for Presto - Official Documentation. (2022). Retrieved November 9, 2022, from Ahana.io website: https://ahana.io/docs/query-with-superset

Schwartz, B. (2019, January 28). What Is Concurrency in a Database? Retrieved November 8, 2022, from Orange Matter website: https://orangematter.solarwinds.com/2019/01/28/what-is-concurrency-in-a-database/

Serverless Data Integration – AWS Glue – Amazon Web Services. (2022). Retrieved November 6, 2022, from Amazon Web Services, Inc. website: https://aws.amazon.com/glue/

Tomar, A. (2020, December). How to use Indexing for SQL Query Optimization | Towards Data Science. Retrieved November 11, 2022, from Medium website: https://towardsdatascience.com/indexing-for-sql-query-optimization-139b57db9fc6

Treehouse Technology Group. (2020, June 4). Big Data vs. Traditional Data: What's the Difference? - Treehouse Tech Group. Retrieved November 10, 2022, from Treehouse Tech Group website: https://treehousetechgroup.com/big-data-vs-traditional-data-whats-the-difference/

Using Amazon Athena Federated Query - Amazon Athena. (2022). Retrieved November 6, 2022, from Amazon.com website: https://docs.aws.amazon.com/athena/latest/ug/connect-to-a-data-source.html

Using Machine Learning (ML) with Amazon Athena - Amazon Athena. (2019). Retrieved November 11, 2022, from Amazon.com website: https://docs.aws.amazon.com/athena/latest/ug/querying-mlmodel.html

Víctor Pérez Pereira. (2021, June 24). Query Logs the AWS WAF using Amazon Athena. Retrieved November 7, 2022, from DEV Community website: https://dev.to/aws-builders/query-logs-the-aws-waf-using-amazon-athena-3dld

What is AWS S3? Your guide to powerful features with minimal complexity. (2022). Retrieved November 5, 2022, from Stitchdata.com website: https://www.stitchdata.com/resources/aws-s3/

What Is Big Data? (2021). Retrieved November 10, 2022, from Oracle.com website: https://www.oracle.com/ie/big-data/what-is-big-data/

When should I use Athena? - Amazon Athena. (2022). Retrieved November 11, 2022, from Amazon.com website: https://docs.aws.amazon.com/athena/latest/ug/when-should-i-use-ate.html

Working with object metadata - Amazon Simple Storage Service. (2022). Retrieved November 11, 2022, from Amazon.com website: https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingMetadata.html

Zoltán Borók-Nagy. (2020, January 21). Speeding Up SELECT Queries with Parquet Page Indexes - Cloudera Blog. Retrieved November 11, 2022, from Cloudera Blog website: https://blog.cloudera.com/speeding-up-select-queries-with-parquet-page-indexes/