**PRESTO DB**

Presto (veya PrestoDB), herhangi bir boyuttaki verilere karşı hızlı analitik sorgular gerçekleştirmek için sıfırdan tasarlanmış, Klasik kitlesel paralel işleme (MPP) veritabanı yönetim sistemine benzer bir mimari kullanan, açık kaynaklı ve dağıtılmış bir SQL sorgu altyapısıdır. Java ile yazılmıştır. Hem Hadoop Dağıtılmış Dosya Sistemi (HDFS),  Cassandra, MongoDB ve  HBase gibi ilişkisel olmayan kaynakları hem de MySQL, PostgreSQL, Amazon Redshift , Microsoft SQL Server ve Teradata gibi ilişkisel veri kaynaklarını destekler. Presto, verileri bir yere aktarmaya gerek kalmadan depolandığı alanda sorgulayabilir. A black background with blue dots

Description automatically generated

Çalışma prensibinde ise birden çok çalışan düğümü ile senkronize olarak çalışan bir koordinatör düğümü vardır. Kullanıcılar SQL sorgularını, çalışan düğümleri arasında dağıtılmış bir sorgu planını ayrıştırmak, planlamak ve zamanlamak için özel bir sorgu ve yürütme motoru kullanan koordinatöre gönderir. Sorgu derlendikten sonra Presto; isteği, çalışan düğümleri arasında birden çok aşamada işler. Gereksiz G/Ç yükünü önlemek için tüm işlemler bellekte yapılır ve aşamalar arasında ağ boyunca işlem hattına yerleştirilir. Daha fazla çalışan düğümü eklenmesi, daha fazla paralellik ve daha hızlı işleme sağlar.

Tarihine bakacak olursak Facebook tarafından geliştirilmiş bir Distributed SQL  Query Engine’dir de diyebiliriz. Başlarda veri analistlerinin Apache Hadoop’taki büyük veri ambarında etkileşimli sorgular çalıştırması için tasarlandı ve geliştirildi. Presto Petabytler’a kadar işlevsel olarak çalışabilmektedir. 2013 yılında Facebook’ta açık kaynaklı sürümü duyuruldu. Sonrasında zamanla Netflix, Amazon gibi büyük firmalar bu sorgu motorunu kullandıklarını açıkladı. Hala aktif bir şekilde kullanılmaktadır.

Presto DB dökümantasyonuna buradan ulaşabilirsiniz: <https://prestodb.io/>

**DUCK DB**

A yellow letters on a black background

Description automatically generated

DuckDB, başlangıçta Hollanda'daki Centrum Wiskunde & Informatica'da (CWI) Mark Raasvelt ve  Hannes [Mühleisen](https://en-m-wikipedia-org.translate.goog/w/index.php?title=Hannes_M%C3%BChleisen&action=edit&redlink=1&_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc) tarafından geliştirilen ve ilk olarak 2019'da yayınlanan [açık](https://en-m-wikipedia-org.translate.goog/wiki/Netherlands?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc)[kaynaklı](https://en-m-wikipedia-org.translate.goog/wiki/Open-source_software?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc)[sütun odaklı](https://en-m-wikipedia-org.translate.goog/wiki/Column-oriented_DBMS?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc)[ilişkisel veritabanı yönetim sistemidir](https://en-m-wikipedia-org.translate.goog/wiki/Relational_database_management_system?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc).( RDBMS)

Kendi veri formatı ile tek dosya halinde veritabanı olarak kullanılabildiği gibi, farklı veri kaynaklarından sorgu yapılarak da kullanılabilir. [Yüzlerce sütun ve milyarlarca satır içeren tabloları](https://en.wikipedia.org/wiki/Table_(database)) birleştirmek gibi  gömülü yapılandırmadaki büyük veritabanlarına karşı karmaşık sorgularda yüksek performans sağlamak üzere tasarlanmıştır. Gömülü veritabanlarının (örneğin, [SQLite ) aksine DuckDB, işlemsel (](https://en-m-wikipedia-org.translate.goog/wiki/SQLite?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc)[OLTP](https://en-m-wikipedia-org.translate.goog/wiki/OLTP?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc) ) uygulamalara odaklanmaz ve bunun yerine [çevrimiçi analitik işleme](https://en-m-wikipedia-org.translate.goog/wiki/Online_analytical_processing?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc) ( OLAP ) iş yükleri için uzmanlaşmıştır .

[DuckDB vektörleştirilmiş bir sorgu işleme](https://en-m-wikipedia-org.translate.goog/wiki/Vectorwise?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc) motoru kullanır . DuckDB, veritabanı yönetim sistemleri arasında özeldir çünkü herhangi bir harici bağımlılığı yoktur ve yalnızca bir C++11 derleyicisiyle derlenebilir.  DuckDB ayrıca, bir ana bilgisayar işleminin içinde çalışarak geleneksel [istemci-sunucu modelinden ayrılır](https://en-m-wikipedia-org.translate.goog/wiki/Client%E2%80%93server_model?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc)

Birçok eklentisi bulunmaktadır. Amazon S3, Google Cloud Storage, PostreSQL gibi ortamlarda bulunan verilere eklentiler kullanarak kolayca aktarılabilmekte. Spatial eklentisi yüklenerek mekansal analizler yapılabilir.

Duck DB dökümantasyonuna buradan ulaşabilirsiniz: <https://duckdb.org/>

**ARANGO DB**

**ArangoDB** , ArangoDB Inc. tarafından geliştirilen bir [grafik veritabanı](https://en-m-wikipedia-org.translate.goog/wiki/Graph_database?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc) sistemidir. ArangoDB , tek bir veritabanı çekirdeği ve birleşik bir [sorgu dili AQL (ArangoDB Query Language) ile üç veri modelini (grafikler,](https://en-m-wikipedia-org.translate.goog/wiki/Query_language?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc)[JSON](https://en-m-wikipedia-org.translate.goog/wiki/JSON?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc) belgeleri, [key-value](https://en-m-wikipedia-org.translate.goog/wiki/Key%E2%80%93value_database?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc) ) desteklediği için [çok modelli](https://en-m-wikipedia-org.translate.goog/wiki/Multi-model_database?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc) bir veritabanı sistemidir. AQL esas olarak bir bildirimsel dildir ve tek bir sorgu içinde farklı veri erişim modellerinin birleştirilmesine olanak tanır.

A black background with green letters

Description automatically generated

ArangoDB **bir NoSQL veritabanı sistemidir** ancak AQL, birçok yönden SQL'e benzer. Veri depolama yöntemleri ACID (Atomiklik, Tutarlılık, İzolasyon, Güvenilirlik) gereksinimlerini karşılar, işlemleri destekler ve hem yatay hem de dikey ölçeklenebilirlik sağlar.[AQL birçok yönden SQL'e](https://en-m-wikipedia-org.translate.goog/wiki/SQL?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc) benzer ve depolama motoru olarak [RocksDB'yi](https://en-m-wikipedia-org.translate.goog/wiki/RocksDB?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc) kullanır. ArangoDB varsayılan depolama biçimi olarak [JSON kullanır](https://en.wikipedia.org/wiki/JSON) ancak dahili olarak serileştirme ve depolama için hızlı ve kompakt bir ikili biçim olan ArangoDB VelocyPack kullanır.

ArangoDB GmbH, 2014 yılında Claudius Weinberger ve Frank Celler tarafından kuruldu.  Başlangıçta veritabanı sistemine "Çok Yönlü Nesne Konteyneri" veya kısaca AVOC adını verdiler ve bu da onları veritabanına AvocadoDB adını vermeye yöneltti.  Daha sonra, adı ArangoDB olarak değiştirdiler.  "Arango" kelimesi, Küba'da yetiştirilen az bilinen bir avokado çeşidini ifade eder.

 ArangoDB, JavaScript tabanlı web uygulamaları için bir sunucu olarak kullanılabilir ve REST/Web API üzerinden veritabanına erişim sağlar. Çok iş parçacıklı mimarisi, yükü tüm CPU çekirdeklerine dağıtarak performansı optimize eder. Ayrıca, farklı veri sunum modellerini tek bir sorguda karıştırma ve heterojen verileri bir araya getirme imkanı sunar.

Arango DB dökümantasyonuna buradan ulaşabilirsiniz: <https://arangodb.com/>

**NEO4J**

Neo4j Java ve Scala dosyası yazılmış olan [Neo4J Technology](https://tr.wikipedia.org/w/index.php?title=Neo4J_Technology&action=edit&redlink=1) isimli firma tarafından açık kaynak olarak dağıtılan, yüksek performanslı ve ölçeklenebilen ve geliştirilen [graph](https://tr.wikipedia.org/w/index.php?title=Graph&action=edit&redlink=1) veritabanıdır. NASA, eBay, cisco gibi kuruluşlar tarafından kullanılmaktadır.

A black background with a black square

Description automatically generated with medium confidence

Geliştiricilerin yerel grafik depolama ve işleme özelliğine sahip ACID uyumlu bir işlem veri tabanı olarak tanımladığı Neo4j, DB-Engines sıralamasına göre en popüler graph tabanlı ve 22.en popüler genel veri tabanıdır.

İlişkisel veritabanlarından farklı olarak veri arasındaki ilişkileri vurgulamak için graph teorisine kullanan, node (düğüm) ve ilişki (ilişkiler) olmak üzere iki temel unsurdan oluşan bir NoSql veritabanıdır. Daha çok graf yapısına göre çalıştığından dolağı ağ, harita, sosyal ağ gibi bir birine bağlı düğümlerin saklandığı ve işlendiği alanlarda kullanılmaktadır. Oracle NoSQL Database, OrientDB, HypherGraphDB, GraphBase, InfiniteGraph ve AllegroGraph gibi diğer popüler grafik veritabanları da vardır. Aşağıdaki görseldeki gbi düşünebiliriz.

A diagram of a person

Description automatically generated

Neo4j'de her şey düğüm, kenar veya nitelik biçiminde tutulur. Her bir düğüm ve kenar istenildiği kadar niteliğe sahip olabilir. Düğümler ve kenarlar etiketlenebilir ve etiketlenmesi arama daraltılması adına kullanışlı olmaktadır.

Neo4j DB dökümantasyonuna buradan ulaşabilirsiniz: <https://neo4j.com/>

**DB Karşılaştırma Tablosu**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Özellik/DB** | **Presto DBPresto DBPre** | **DuckDB** | |  | | --- | | **ArangoDB** | |  |  |  | | --- | |  | | **Neo4j** |
| |  | | --- | | **Veri Modeli** |  |  | | --- | |  | | |  | | --- | | Distributed SQL Query Engine |  |  | | --- | |  | | |  | | --- | | Columnar Database |  |  | | --- | |  | | |  | | --- | | Multi-Model (Document, Key-Value, Graph) |  |  | | --- | |  | | |  | | --- | | Graph Database |  |  | | --- | |  | |
| |  | | --- | | **Sorgu Dili** |  |  | | --- | |  | | |  | | --- | | SQL |  |  | | --- | |  | | SQL | |  | | --- | | AQL (ArangoDB Query Language), Gremlin, Cypher |  |  | | --- | |  | | Cypher |
| **Ölçeklenebilirlik** | |  | | --- | | Çok Yüksek (Distributed, Clustered) |  |  | | --- | |  | | |  | | --- | | Tek Node (Single-Node, Embedded) |  |  | | --- | |  | | |  | | --- | | Yüksek (Clustered Deployment) |  |  | | --- | |  | | |  | | --- | | Yüksek (Clustered Deployment) |  |  | | --- | |  | |
| |  | | --- | | **Veri Saklama** |  |  | | --- | |  | | |  | | --- | | Parquet, ORC, Hive, Other Distributed Formats |  |  | | --- | |  | | |  | | --- | | Columnar Storage |  |  | | --- | |  | | |  | | --- | | JSON (Documents), Graphs, Key-Value |  |  | | --- | |  | | |  | | --- | | Nodes, Relationships |  |  | | --- | |  | |
| |  | | --- | | **Kullanım Alanları** |  |  | | --- | |  | | |  | | --- | | Büyük Veri Analitiği, BI, Veri Gölü İşlemleri |  |  | | --- | |  | | |  | | --- | | Veri Analizi, BI, Gözetimli Öğrenme |  |  | | --- | |  | | |  | | --- | | Multi-Model Analiz, Graph-Document Kombinasyonları |  |  | | --- | |  | | |  | | --- | | Sosyal Ağlar, Yol Bulma, Öneri Sistemleri |  |  | | --- | |  | |
| **Performans** | |  | | --- | | Büyük Veri Kümeleri İçin Yüksek Performans |  |  | | --- | |  | | |  | | --- | | OLAP İşlemleri için Yüksek Performans |  |  | | --- | |  | | |  | | --- | | Grafik, Belge ve Key-Value İşlemlerinde Dengeli |  |  | | --- | |  | | |  | | --- | | Grafik Sorguları için Optimize |  |  | | --- | |  | |
| |  | | --- | | **Öne Çıkan Özellikler** |  |  | | --- | |  | | |  | | --- | | Büyük veri kümeleri için esnek ve dağıtık sorgular |  |  | | --- | |  | | |  | | --- | | Kolay entegrasyon, hafif ve gömülü yapı |  |  | | --- | |  | | |  | | --- | | Çoklu veri modeli desteği, esnek sorgu seçenekleri |  |  | | --- | |  | | Yüksek performanslı graph işlemleri, ACID desteği |

Kaynakça:

<https://aws.amazon.com/tr/what-is/presto/>

<https://en.wikipedia.org/wiki/Presto_(SQL_query_engine)>

<https://en.wikipedia.org/wiki/DuckDB>

<https://en.wikipedia.org/wiki/ArangoDB>

<https://tr.wikipedia.org/wiki/Neo4j>

<https://medium.com/5bayt/neo4j-nedir-e7160602211e>

<https://medium.com/turknettech/neo4j-graph-database-hakk%C4%B1nda-15e13f365490>

[https://miro.medium.com/v2/resize:fit:1100/format:webp/1\*9290nSxCb8EjE6aPKKiT7w.png](https://miro.medium.com/v2/resize:fit:1100/format:webp/1*9290nSxCb8EjE6aPKKiT7w.png)

<https://chatgpt.com/>