**MariaDB:**

MariaDB on vapaa avoimeen lähdekoodiin (GNU General Public License) perustuva SQL-pohjainen tietokannanhallintajärjestelmä. Tämä tarkoittaa sitä, että ohjelman lisenssi sallii käyttäjille oikeuden käyttää, tutkia, muuttaa ja jakaa ohjelmaa kenelle tahansa. MariaDB perustuu MySQL:ään, minkä vuoksi sillä on suureksi osaksi samat komennot ja toiminnallisuudet. MariaDB on varsinkin web devaajien suosiossa, ja pienet ja keskisuuret yritykset käyttävät sitä mielellään sen ilmaisuuden ja tehokkuuden vuoksi. Tämän lisäksi sitä käyttävät myös Wikipedia, Booking.com, UBS, sekä Google.

MariaDB on relaatiotietokantapohjainen. Relaatiomallilla tarkoitetaan sitä, että tieto on järjestetty riveihin ja sarakkeisiin taulukkomuodossa. Rivi kuvaa tietuetta ja sarakkeet kuvaavat tietueen ominaisuutta. Taulukoilla voi olla keskinäisiä yhteyksiä eli relaatioita. Relaatiot määritellään avain-sarakeperiaatteella. Ensinnäkin sarake määritellään pääavaimeksi, joka yksilö tietueen. Pääavaimen on oltava uniikki ja ei-tyhjä, mikä takaa tietokannan yhtenäisyyden/ konsistenssin. Vierasavaimilla taas viitataan toisiin taulukoihin ja ne aina kohdistuvat toisen taulukon pääavaimeen.

MariaDB on avoimen lähdekoodin ohjelmistona vapaasti käytettävissä, modifioitavissa ja levitettävissä. Tämän myötä sillä on laaja yhteisö devaajia ja käyttäjiä, mikä luo hyvät olosuhteet MariaDB:n jatkuvalle innovoinnille. Sillä onkin kattavat kirjastot ja dokumentaatiot, joita MariaDB:n yhteisö luo ja ylläpitää. MariaDB on yhteensopiva MySQL:n kanssa, joten MySQL-applikaatioita voi siirtää MariaDB:hen pienin muutoksin. MariaDB käyttää esimerkiksi Aria storage enginea, joka tekee siitä tehokkaan. MariaDB sietää virhetiloja hyvin ja sillä on korkea käytettävyys. MariaDB:llä on hyvät tietoturvallisuusominaisuudet, mikä mahdollistaa roolit käyttäjien mukaan ja sillä on hyvä salasanavalidaatio ja -enkryptio.

**MongoDB:**

MongoDB on NoSQL-pohjainen tietokannanhallintajärjestelmä. MongoDB varastoi dataa JSON:n kaltaisiin BSON-dokumentteihin (Binary JSON). MongoDB tallentaa dokumentteja taulukoiden sijaan ja ne ovat usein JSON:n tapaan luotuja olioita, mutta joskus dokumentit voivat olla myös esimerkiksi XML-pohjaisia. Tämän etuna on se, että monentyyppistä dataa voi joustavasti tallentaa samaan kokoelmaan ilman ennalta määriteltyä skeemaa. Joillakin tietueilla voi olla erityyppistä dataa ja ylimääräisiä kenttiä muihin tietueisiin verrattuna. Dokumentit voivat sisältää jopa toisia dokumentteja ja sisäkkäisiä tauluja. Tämä on merkittävä eroavaisuus SQL-pohjaisiin järjestelmiin, kuten MariaDB:hen verrattuna, joissa on ennalta määriteltävä datatyypit ja muut rajoitukset datan tallentamista varten. MongoDB:ssä ei ole samanlaista integriteettivaatimusta. MongoDB:tä käyttävät esimerkiksi Adobe, MetLife ja Cisco.

NoSQL-tietokannat sallivat datan tallentamisen ilman ennaltamääriteltyä skeemaa. Tallennetut tietueet tai dokumentit voivat olla rakenteeltaan ja tyypiltään melkein mitä tahansa ja uusia kenttiä voi lisätä vaivattomasti. Tämä joustavuus sopii erittäin hyvin tilanteisiin, joissa on dynaamista ja ajan myötä kehittyvää dataa, jolloin tietokanta voi helposti sopeutua uusiin olosuhteisiin. MongoDB skaalautuu hyvin horisontaalisesti. MongoDB pystyy jakamaan dataa useiden tietoverkkojen yli ja pystyy käsittelemään suuria määrää dataa ja vilkasta tietoliikennettä. Tämän takia MongoDB soveltuu hyvin applikaatioille, joilla on nopeasti kasvavat datamäärät. MongoDB:tä käytetäänkin usein Big Datan analysoinnin yhteydessä.

MongoDB tukee automaattista tietokannan pirstaloimista siten, että dataa voi jakaa useiden koneiden kesken. MongoDB tukee useantyyppisiä indeksejä, joiden avulla voidaan tehostaa tietokantakyselyitä ja datan lajittelemista. MongoDB:llä on SQL:n tapaan tehokas kyselykieli, joka edesauttaa tietokantojen rakentamista. MongoDB:llä on hyvä kyky toimia myös silloin, kun tulee ongelmia hardwaren tai tietoyhteyksien kanssa. Kaiken tämän lisäksi MongoDB:llä on laaja ja aktiivinen yhteisö toimintansa taustalla, jotka kehittävät sitä jatkuvasti. Sillä on hyvä dokumentaatio, työkalut ja kirjastot, ja sen voi integroida toimimaan useiden kielien ja frameworkien kanssa.

MongoDB:llä on konsistenssin suhteen malli, jonka englanninkielinen nimi on ”eventual consistency”. Sen heikkoutena on, että kirjoitusoperaation jälkeen ei ole takeita siitä, milloin data on näkyvissä muissa servereissä tai klienttien koneissa. MongoDB sallii mahdollisuuden toteuttaa korkeamman tasoista konsistenssia, mutta sen kääntöpuolena on hitaampi suorituskyky. Silloin kun vaaditaan enemmän yhtenäisyyttä datastruktuurissa, voi niitä säätävien konfiguraatioiden tekeminen olla monimutkaista ja vaikuttaa tietokannan tehokkuuteen negatiivisesti. MongoDB:llä on myös vaikeaa tehdä monimutkaisia kyselyitä koskien useampia joineja ja aggregaatioita. MongoDB käyttää huomattavasti muistia esimerkiksi indeksoinnin yhteydessä ja datan määrä voi kasvaa huomattavasti. Transaktiot eivät ole yhtä tehokkaita kuin esimerkiksi MariaDB:ssä.

**Yhteenveto**

MariaDB:tä on hyvä käyttää silloin, kun datan on syytä olla strukturoitua (asiakastiedot yms.) ja rahaa käsittelevissä applikaatioissa, joissa ACID-periaatteella toimivat transaktiot ovat hyvä ratkaisu liiketoiminnan toteuttamisessa. MariaDB sopii hyvin tilanteisiin, joissa on monimutkaisia kyselyitä ja liitoksia taulujen välillä. SQL-kieltä käyttävänä järjestelmänä MariaDB sopii hyvin tilanteisiin, joissa datan on oltava yhtenäistä, jolloin SQL-kielen vaatimat validaatiosäännöt ja datan rajoitukset sopivat hyvin tällaiseen tarkoitukseen. Kaiken kaikkiaan MariaDB tarjoaa paljon toimintoja ja työkaluja toimivalle tietokantaympäristölle.

MongoDB taas on hyvä ratkaisu silloin, jos tallennetaan strukturoitumatonta tai puoliksistrukturoitua dataa. MongoDB:n joustava skeema on hyvä ratkaisu, jos on ennakoitu, että datan muoto saattaa muuttua tulevaisuudessa. MongoDB:n skaalautuvuus tarjoaa mahdollisuuden suurten data määrien lukemiselle ja kirjoittamiselle, minkä vuoksi sitä voi hyödyntää Big Data analysoinnissa.

Näiden eri tietokannanhallintajärjestelmien käyttöönotossa on siis huomioitava oman ydintoiminnan, oli kyseessä liiketoiminta tai asiakaspalvelu, keskeiset tarpeet ja sen mukaan valita, kumpaa tietokantaa käyttää. Ei ole epätavanomaista, että molempia järjestelmiä voi käyttää riippuen siitä, minkälaista tietoa tallennetaan ja käsitellään.