Contents

[MariaDB 1](#_Toc64028980)

[PHP 1](#_Toc64028981)

[Asenna mariadb-server 2](#_Toc64028982)

[Tila 2](#_Toc64028983)

[Konfigurointi 2](#_Toc64028984)

[Tietokantojen luonti 3](#_Toc64028985)

[PHP 6](#_Toc64028986)

[Tiedon hakeminen tietokannasta 7](#_Toc64028987)

[Ongelmanratkonta 8](#_Toc64028988)

[PHP lisäjuttuja 9](#_Toc64028989)

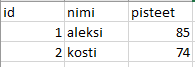
MariaDB ja PHP ei ole videota.

# MariaDB

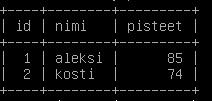
Tässä materiaalissa ja siihen liittyvässä tehtävässä luot yksinkertaisen tietokannan ja otat siihen yhteyden PHP-kielellä rakennetulla nettisivulla. Myöhemmissä harjoituksissa tietokantoja tulee enemmän, kun rakennetaan esimerkiksi WordPress ja Moodle ympäristöjä.

Tietokannoissa perusideana on, että meillä on erilaisia tietokantoja ja nämä tietokannat sisältävät tauluja. Nämä taulukot muodostuvat sarakkeista ja taulukoista samalla tavalla kuin Excel-taulukko.

Kuvassa on Excel-taulukko, jossa on 3 eri saraketta (id, nimi ja pisteet) sekä rivit (1, aleksi, 85 – 2, kosti, 74). Tietokannat ovat siis ihan samanlaisia taulukoita, mutta voivat olla laajuudeltaan todella suuria.



Tässä on sama, mutta mariadb-tietokannan taulussa:



Tietokannoissa joudutaan myös ensin luomaan ikään kuin pohja näille tiedoille. Siinä määritetään minkä tyyppisiä arvoja mihinkin sarakkeeseen voidaan asettaa. Esimerkiksi pisteet kohtaan hyväksytään vain numeroita, ei tekstiä lainkaan. Luodaan myös perusavain, joka on uniikki tunniste jokaiselle riville.

# PHP

PHP on ohjelmointikieli, jolla hoidetaan www-sivuilla backendiä eli sitä, mitä käyttäjä ei suoraan näe, kun hän käyttää nettisivua. PHP käsittelee esimerkiksi tilanteita, kun käyttäjä lataa sivulle tietoa tietokannasta. Frontend on taas se kaikki mikä näkyy käyttäjälle, kuten HTML ja CSS, jotka muodostavat sivun visuaalisen osuuden. Esimerkiksi sivun taustavärin ja fontit.

PHP avulla voitaisiin esimerkiksi näyttää käyttäjälle ylläoleva tietokanta selaimessa. Samaan tapaan kuin käytät Netflixiä tai Steamia, sisältö (eli elokuvat ja pelit) ladataan jostakin tietokannasta sinulle.

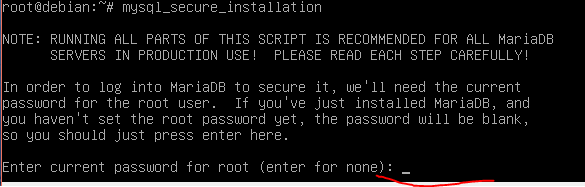
# Asenna mariadb-server

**apt-get install mariadb-server**

Asennuksen jälkeen tee kuvan mukaisesti koska rootille ei asetettu vielä salasanaa:

**mysql\_secure\_installation**

Tällä komennolla otetaan tietokannat käyttöön. Joudut aluksi antamaan rootin salasanan. Huomaa, että punaiseen kohtaan ei tule mitään merkkejä, kun kirjoitat salasanan siihen. Salasanan kirjoittamisen jälkeen paina Enter. Voit sitten vaihtaa salasanan, jos haluat.



Tuleviin kohtiin voi valita Y vaihtoehdon, mutta salasanaa ei kuitenkaan tarvitse vaihtaa.

# Tila

Mariadb-paketin tilan näet:

**systemctl status mariadb**

Mariadb-paketin voit käynnistää uudelleen:

**systemctl restart mariadb**

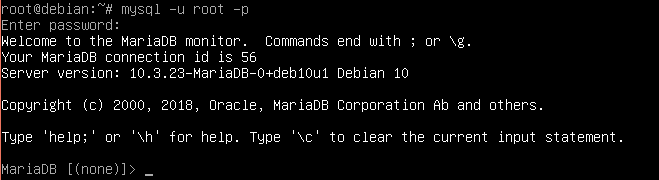
Mariadb version näet:

**mysqladmin version**

# Konfigurointi

Seuraavaksi muodostetaan yhteys tietokantaan:

**mysql -u root -p**



Komentokehote on nyt erilainen.

Tästä komentokehotteesta pääset pois näin:

**exit**



Tietokantaan halutaan luoda käyttäjä vain sitä varten, sillä rootin käyttöä ei suositella. Tässä luodaan uusi käyttäjä nimeltään taitaja, jonka salasana on Sala1234. Hänellä on lupa käyttää kaikkia tietokantoja. Muista lopussa oleva ; puolipiste.

**GRANT ALL ON \*.\* TO taitaja@'localhost' IDENTIFIED BY ‘Sala1234’ WITH GRANT OPTION;**



”Tallennetaan” asetukset:

**FLUSH PRIVILEGES;**



# Tietokantojen luonti

Jos ; puolipisteen komennon lopusta tulee -> komentokehote. Tästä pääset eteenpäin, kun lisäät siihen puolipisteen. Tämä täydentää komennon loppuun.



Tässä luodaan taitajapist niminen tietokanta, johon laitetaan kisaajien nimiä ja pisteitä.

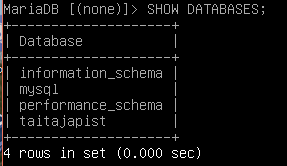
Tietokanta luodaan näin:

**CREATE DATABASE taitajapist;**



Näin näet mitä tietokantoja sinulla on:

**SHOW DATABASES;**



Siirrytään taitajapist tietokantaan:

**USE taitajapist;**



Luodaan sinne uusi taulu nimeltään hyriantaitajat. Sinne luodaan sarake nimeltään id, joka on tyypiltään INT eli numero, se on myös perusavain ja numero kasvaa automaattisesti, kun lisää rivejä luodaan.

nimi sarake on tyypiltään TEXT, eli siihen voidaan kirjoittaa melkein mitä vain.

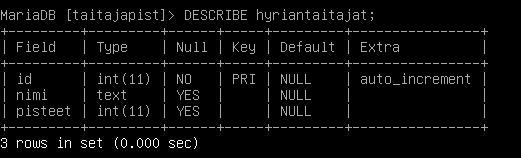
pisteet sarake on myös INT, eli siihen voi laittaa vain numeroita.

lopuksi sulkeet suljetaan ja päätetään ; puolipisteellä.



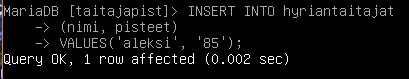
DESCRIBE komennolla nähdään mitä juuri luotiin:

**DESCRIBE hyriantaitajat;**

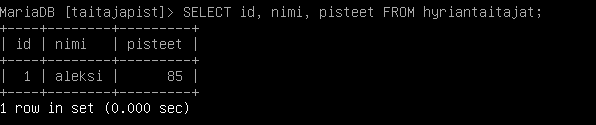


Seuraavaksi lähdetään syöttämään tauluun tietoja.

Syötetään tietoja hyriantaitajat tauluun. Sulkujen sisällä määritetään mihin sarakkeisiin arvot menevät. Huomioi, että id sarakkeeseen ei laiteta mitään, koska sen arvo kasvaa automaattisesti. VALUES kohdalla määritetään, mitkä arvot mihinkin sarakkeeseen laitetaan. Eli ”nimi” sarakkeeseen syötetään arvo ”aleksi”. Ja ”pisteet” sarakkeeseen syötetään arvo ”85”.



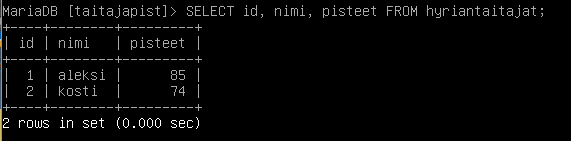
Tarkistetaan, että tiedot tallennettiin. Eli valitaan SELECT komennolla id, nimi ja pisteet sarakkeet taulusta nimeltään hyriantaitajat.



Syötetään tauluun vielä lisää tietoja:



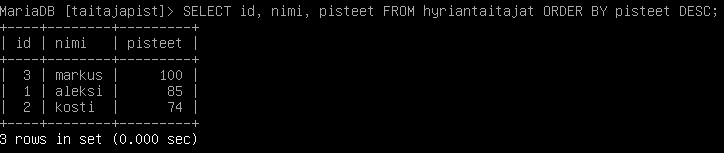
Tarkistetaan, että tiedot tallentuivat:



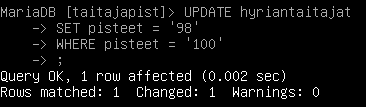
Syötetään tauluun vielä yhdet tiedot:



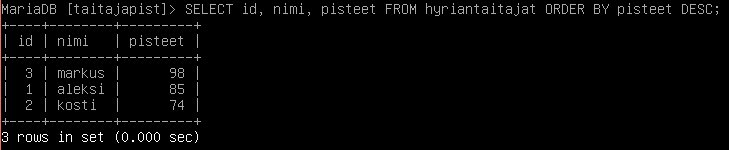
Luetaan tiedot, mutta lisätään vielä ORDER BY komento ja DESC. Näillä taulukko saadaan lajiteltua järjestykseen pisteiden mukaan DESC, eli laskevassa järjestyksessä.



Tarkistuksen yhteydessä huomattiin, että markus ei ollutkaan tehnyt jotakin asiaa oikein ja hänen pistemääränsä päivitettiin. Tähän käytetään UPDATE komentoa hyriantaitajat tauluun. SET komennolla asetetaan pisteet arvoksi 98, WHERE eli siinä taulukon kohdassa jossa pisteet olivat 100.



Pisteet ovat muuttuneet:



Aleksi ei haluakaan kisata, joten hänet poistetaan taulusta kokonaan.

DELETE FROM, eli mistä taulusta poistetaan. WHERE komennolla määritetään mitä kohtaa etsitään, ja se poistetaan. Tässä haetaan rivi, josta löytyy aleksin nimi ja se poistetaan.

Lopuksi tarkistetaan, että se poistui.



# PHP

Asenna php:



Asenna myös php-mysqli paketti, ilman tätä ei tietojen hakeminen php:llä onnistu tietokannasta:



Seuraavaksi testataan php toiminta käyttämällä phpinfo() metodia. Käytetään apache2 webpalvelimen oletushakemistoa (/var/www/html) ja poistetaan sieltä index.html tiedosto.



Luodaan sen tilalle index.php.

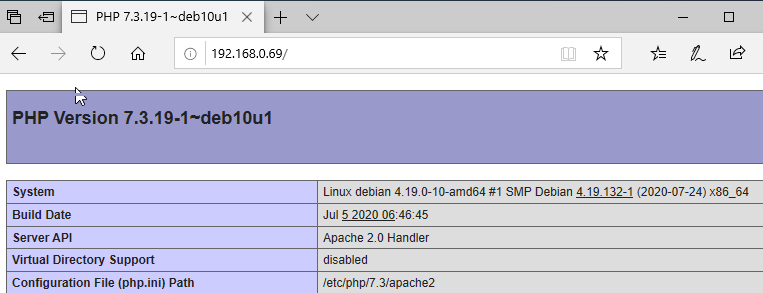


Jonka sisältö on seuraava:



php koodi tulee aina tagien <?php ?> väliin.

Käynnistä uudelleen apache2 palvelu. Selaimella voidaan testata toimivuus (palvelimen IP):



# Tiedon hakeminen tietokannasta

Seuraavaksi haetaan PHP:lla tietoja juuri luodusta tietokannasta. Tässä kerron mitä index.php tiedostosta löytyy nyt sisältönä. Siitä kuva alla. **Tämä on sama tiedosto, jossa oli alun perin phpinfo() koodinpätkä.**

**Kun testaat tämän toimintaa, muista päivittää selain esim. CTRL + F5, jotta tietoja ei haeta välimuistista!**

Rivillä 1 aloitetaan php

Riveillä 2-5 määritetään muuttuja, joita tullaan käyttämään. Esimerkiksi $palvelin muuttujaan tallennetaan palvelimen sijainti ja $kayttaja muuttujaan tietokannan käyttäjän tunnus. Näiden tietojen tulee siis vastata niitä, joita olet palvelimella käyttänyt, kun teit tietokannan ja sen käyttäjät. localhost kohdan tilalle voit myös kirjoittaa palvelimen IP-osoitteen.

Rivillä 7 muodostetaan yhteys tietokantaan mysqli\_connect metodilla. Sille annetaan tarvittavat tiedot aikaisemmilla muuttujille (eli mikä on palvelimen sijainiti, käyttäjä, käyttäjän salasana ja tietokannan nimi).

Rivillä 10 luodaan kysely, joka toimii täysin samalla tavalla kuin aikaisemmin teit tietokannassa. Tässä valitaan kaikki sarakkeet taulusta nimeltään hyriantaitajat.

Rivillä 11 $vastaukset muuttujalle määritetään että yhteys otetaan kohteeseen, joka on määritelty muuttujassa $yhteystietokantaan ja kysely, joka suoritetaan on määritelty muuttujassa $kysely

Rivillä 13 luodaan ehto lauseke, joka hakee tietoja tietokannasta niin kauan kunnes rivien määrä on suurempi kuin 0, eli päästään loppuun.

Rivillä 14 luodaan toistorakenne, jolla haetaan tietokannasta tiedot, jotka vastaavat $rivi[’id’] kohdissa määritettyjä arvoja.

Rivillä 15 tehdään tuo tietojen tulostus. Esimerkiksi $rivi[’nimi’] hakee tietoja tämän nimisestä sarakkeesta tietokannasta. Haetaan siis tiedot sarakkeista id, nimi ja pisteet.

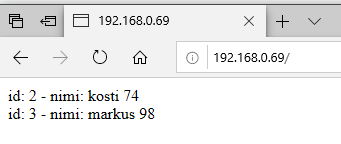
Rivillä 16 ja 17 suljetaan ehto ja toisto lausekkeet.

Rivillä 19 suljetaan yhteys tietokantaan.

Rivillä 21 lopetetaan php.



Lopputulos:

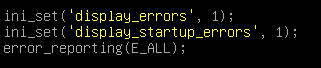


Koodissa olevaa mysqli asennettiin aikaisemmin. PHP käyttää mysqli metodeja, kun se toimii tietokannan kanssa. Se on ikään kuin PHP:n laajennus/lisäosa.

# Ongelmanratkonta

Tyypillisesti ongelmat ovat pieniä virheitä, kuten ; tai ” tai } merkkien puuttumisia joistakin kohdista. Yllä oleva koodinpätkä kyllä toimii, mutta kirjoita se huolella ja muista tehdä tarvittavat muutokset, jotta se toimii sinun tehtävässäsi.

Oletuksena selaimessa ei näe PHP: virheilmoituksia. Lisäämällä ne index.php tiedostoon kuvan mukaisesti selain näyttää virheet selaimessa. Laita nämä tietenkin php tagien sisälle:



Luodaan keinotekoinen ongelma, jotta nähdään, että error-viestit varmasti tulostuvat:



Nyt nähdään millä rivillä ongelma on ja voidaan tehdä tarvittavat muokkaukset:



Nanossa saat rivinumerot näkyville näin:



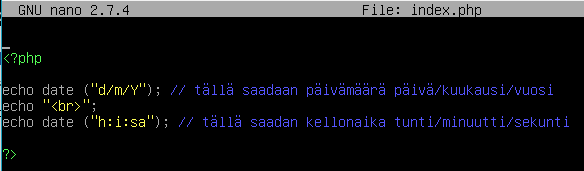
Nähdään ongelmarivi:

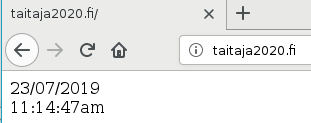


# PHP lisäjuttuja

Tässä muutama PHP kohta, joita kisoissa on ollut.

Kellonaika ja päivämäärä PHP:llä.

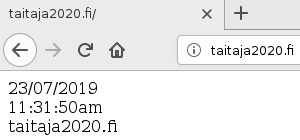




Fully qualified domain name.

Jos halutaan tulostaa pelkkä domain nimi riittää alla oleva komento.





Jos halutaan tietokoneen nimi + domain nimi, sen voisi toteuttaa näin:

