

# ÄLYKÄS AVAINJÄRJESTELMÄ

Antti Mehtälä

Opinnäytetyö

Tieto- ja viestintäteknikka  
Insinööri (AMK)

2022

Tieto- ja viestintäteknikka  
Insinööri (AMK)

---

<b>Tekijä</b>	Antti Mehtälä	<b>Vuosi</b>	2022
<b>Ohjaaja</b>	Ari Karjalainen		
<b>Toimeksiantaja</b>	Antti Mikael Mehtälä TMI		
<b>Työn nimi</b>	Älykäs avainjärjestelmä		
<b>Sivumäärä</b>	23		

---

Tämän opinnäytetyön aiheena on älykäs avainjärjestelmä. Avainjärjestelmän tavoitteena on tuottaa dataa palautetuista avaimista ja vähentää turhia tarkistuskäyntejä matkailualueilla, joissa turistit voivat palauttaa avaimia palautuslaatikkoihin. Opinnäytetyön tarkoituksena on vähentää turhaa liikkumista ja työajan käyttöä sellaisilta toimijoilta, jotka ovat riippuvaisia palautetuista avaimista, kuten majoitustoiminta ja autovuokraamot.

Järjestelmä tehtiin omalle yritykselle, joka on myynyt vastaavia järjestelmiä aiemmin. Opinnäytetyössä kehitetään tietopohjaa, datan varastointia ja päivitettävyyttä. Opinnäytetyössä tehty kehitys perustuu MariaDB:hen, Pythoniin, Apacheen ja Raspberry PI:hin. RFID-lukija kytketään Raspberryyhin, joka saa tiedon palautuneesta avaimesta. Kehitystyössä kiinnitetään huomiota kirjastoihin, joita järjestelmä tarvitsee toimiakseen.

Lopputuloksena on automaattinen järjestelmä, joka yhdistää avaimen yksilöllisen tunnisteen oikealle tiedolle. Tämä tieto päivittyy verkkosivuille, ja opinnäytetyön esimerkissä käytetään taulukkoa palautuneista autoista. Rajapinta on käytettävissä myös muille alustoille, kuten Power BI ja Excel.

Opinnäytetyö auttaa ymmärtämään SQL-palvelimien toiminnallisuuksia, virtuaalipalvelimen käyttömahdollisuuksia ja tarjoaa toimintatapoja kirjastojen tuomiselle Pythoniin. Yksinkertaisella toteutuksella voidaan vähentää turhaa työajankäyttöä kaikilla palautustapahtumista riippuvaisilla aloilla, ja järjestelmä on helposti laajennettavissa muihin käyttötarkoituksiin.

Study Programme in Information  
and Communication Technology  
Bachelor of Engineering

---

<b>Author</b>	Antti Mehtälä	<b>Year</b>	2022
<b>Supervisor</b>	Ari Karjalainen		
<b>Commissioned by</b>	Antti Mikael Mehtälä Sole Trader		
<b>Title</b>	Smart Returning System for Keys		
<b>Number of pages</b>	23		

---

The goal of this thesis study was to produce a smart returning system for keys, which would log the returning time of keys and display that data for end user. This study was commissioned by the author's own company which had sold similar systems before.

This thesis study included developing SQL environment, Python-scripting, and Apache configurations. With Raspberry PI the returning data was produced via RFID-reader.

The result of this thesis study was a system based on MariaDB, Python, Apache, and Raspberry. The scalable system with cloud integration and data can easily be modified if needed.

Keywords

Database programs, data types, Python

## SISÄLLYS

1 JOHDANTO .....	6
2 PROSESSIN LÄHTÖKOHDAT .....	7
2.1 Käyttökohteet.....	7
2.2 Prosessin vaiheet .....	8
3 JÄRJESTELMÄKOKONAISUUS .....	10
3.1 Pilvipalvelimen määrittely ja vaatimukset.....	10
3.2 Tietokannat pilvipalvelimella .....	11
3.3 Apachen asennus .....	13
3.4 Raspberry Pi ja skriptit.....	14
3.5 Hallinta ja ylläpito .....	19
3.6 Jatkokehitys .....	21
4 POHDINTA .....	22
LÄHTEET.....	23

## KÄYTETYT LYHENTEET

ER-malli	Entity-Relationship Model, malli tietotyyppien tai olioiden suhteesta
RFID	Radio Frequency Identification, radiotunniste
SSH	Secure Shell, salatun tietoliikenneyhteyden protokolla

## 1 JOHDANTO

Opinnäytetyöni keskittyi avainten hallintaan ja sen sujuvoittamiseen teknologiaa hyväksikäyttäen. Tavoitteena oli vähentää työresurssien hukkaamista muun muassa matkailualoilla, ja tämän tavoitteen saavuttamiseksi käytettiin hyvinkin yksinkertaisia metodeja.

Tarve opinnäytetyön runkona toimivalle järjestelmälle, joka ilmoittaa palautuneista autoista heräsi, kun työskentelin autovuokraamoilla. Kehitin perusversion, joka ilmoitti sähköpostitse palautuneista autoista, ilman etähallinta- tai päivitysmahdollisuuksia. Hallinnan yksinkertaistamiseksi ja skaalattavuuden lisäämiseksi ympäristö siirrettiin pilvialustalle ja pois vanhoilta skripteiltä. Ensimmäisenä määriteltiin vaatimukset pilviympäristölle ja etsittiin sopiva pilviympäristön tarjoaja. Seuraavana tehtiin tarvittavat asennukset tietokannan ja verkkosivujen osalta. Viimeisenä tuotettiin Raspberry PI -konfiguraatio, joka osaa lukea tietoa RFID-avaimenperistä ja syöttää tiedon pilvitietokantaan.

Opinnäytetyössä kerrotaan aluksi järjestelmän toimintalogiikka, käydään läpi pilviympäristöjen vaatimukset ja esitellään työvaiheet. Seuraavaksi toteutetaan pilviympäristön työvaiheet, joka vaaditaan järjestelmän toimimiseen. Sen jälkeen käydään toteutuksen muut vaiheet läpi ja puretaan eri työvaiheet lähdekooditasolle. Pohdintaosiossa arvioidaan tuotteen käyttöarvoa, jälleenmyyntiarvoa ja kehityskohteita.

Opinnäytetyön tarpeellisuus ilmeni aiemmassa työssäni autovuokraamolla. Rakensin siellä samoilla työkaluilla jäykemmän järjestelmän, joka ei skaalautunut resurssitarpeiden mukaan. Tämä järjestelmä toimi pohjana tarpeiden määrittelymiselle, sekä antoi mahdollisuuden miettiä tulevia markkinointikohteita.

## 2 PROSESSIN LÄHTÖKOHDAT

Tarvekartoitusta ei erikseen tehty, vaan tarve kuvatun kaltaiselle järjestelmälle heräsi, kun pyrittiin vähentämään turhia työajoja. Autovuokraamolla työskennellessä talvisesongilla tuli viikoittain ajoja lentoasemalle, kun tarkistettiin, onko auto X palautunut kuten asiakkaan kanssa oli sovittu. Palautumaton auto aiheutti sesonkiaikana ylimääräisiä toimitusviivästyksiä, jotka olisi voitu välttää, mikäli ylimääräistä käyntiä lentoasemalla ei olisi tarvinnut tehdä.

Ensimmäinen versio kehitettiin niin, että avaimen palautus aktivoi skriptin, joka ilmoitti sähköpostilla, kun auto oli palautunut. Tämän järjestelmän heikkous oli huono päivitettävyyttä, eli jokainen auto tarvitsi erikseen koodata järjestelmään, jotta sähköpostissa oli oikean auton tiedot. Koska palautuslaitteelle ei ollut etähallintaa, autoja päivittääkseen oli oltava fyysisesti yhteydessä palautuslaitteeseen.

### 2.1 Käyttökohteet

Lapissa tarvetta avaintenpalautusjärjestelmälle voisi löytyä majoituspalveluista, välinevuokrauksesta ja autovuokraamoilta. Osalla Suomessa toimivista autovuokraamoista on käytössään selainpohjainen järjestelmä, johon palautustiedot olisivat helposti haettavissa SQL-palvelimelta. Skaalautuvuus korostuu siinä, että opinnäytetyössä esitellyillä menetelmillä, vastaava järjestelmä on kustannustehokas rakentaa myös pienelle liiketoiminnalle, kuten Airbnb-majoitusratkaisuille.

Järjestelmän logiikka tarjoaa myös laajennettavuutta muuhunkin toimintaan, kuin palautettaville avaimille. Potentiaalisia käyttökohteita ovat teollisuudessa toimivat IT-osastot, joilla tuotanto pyörii vuorokauden ympäri mutta IT-osasto on tavoitettavissa vain arkityöaikana. Laitteiden toimitus huoltoon onnistuisi toteuttaa keräyspisteillä, joista tieto tulisi tukipalveluihin. Autovuokraamojen lisäksi autojen maahantuonnissa voisi olla potentiaalista asiakaskuntaa, kun autoja puretaan laivoista. Vastaavanlaisia järjestelmiä on jo käytössä kirjastotoiminnassa.

Käyttötapauksessa yksi asiakas palauttaa mökin avaimet palautuspisteelle. Tässä tapauksessa käyttäjä on asiakas ja laukaisija on palautustapahtuma. Käyttötapauksessa asiakas päättää mökin vuokrauksen palauttamalla avaimet ennalta sovittuun palautuslaatikkoon. Mökin vuokraaja saa tiedon mökin vuokrauksen päättymisestä ja voi suunnitella mökkien siivousjärjestyksen. Poikkeuksellisenä toimintana voidaan pitää skenaariota, jossa asiakas ei palauta avainta sovittuun paikkaan, vaan jättää ne mökin pöydälle.

Käyttötapauksessa kaksi asiakas palauttaa auton vuokrauspisteelle. Tässä tapauksessa käyttäjä on asiakas ja laukaisija on palautustapahtuma. Käyttötapauksessa asiakas päättää auton vuokrauksen palauttamalla avaimet ennalta sovittuun palautuslaatikkoon. Autovuokraamo saa tiedon auton palautumisesta, ja voi myydä auton eteenpäin seuraavalle asiakkaalle. Poikkeuksellisenä toimintana voidaan pitää tilannetta, jossa asiakas ei palauta autoa ollenkaan, vaan jatkaa vuokrausta ilmoittamatta siitä vuokraamolle. Järjestelmä tukee lisävuorokauden veloitusta, kun palautusta ei ole tapahtunut.

Käyttötapauksessa kolme asiakas palauttaa auton etuajassa vuokrauspisteelle. Tässä tapauksessa käyttäjä on asiakas ja laukaisija on palautustapahtuma. Käyttötapauksessa asiakas päättää auton vuokrauksen aiemmin kuin sovittu palauttamalla avaimet ennalta sovittuun palautuslaatikkoon. Autovuokraamo saa tiedon auton palautumisesta, ja voi varmistaa pysäköintitilojen riittävyyden muille palautetuille autoille sekä myydä auton eteenpäin seuraavalle asiakkaalle. Poikkeuksellisenä toimintana voidaan pitää tilannetta, jossa asiakas ei palauta avaimia. Tietoa palautuneesta autosta ei tule sillä palautustapahtumaa ei ole luotu.

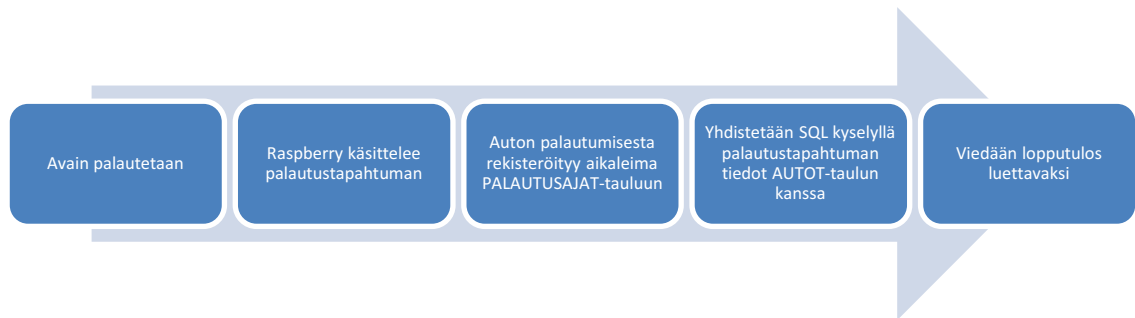
## 2.2 Prosessin vaiheet

Prosessi (Kuvio 1) koostuu viidestä eri vaiheesta:

1. Avain palautuu.
2. Avaimenperän tiedolla haetaan autoa vastaava ID tietokantapalvelimelta.



3. Auton ID tallentuu palautustapahtumana ja aikaleima tapahtumalle lisätään.
4. Tieto palautuneesta avaimesta ja aikaleimasta yhdistetään auton tietojen kanssa.
5. Tiedot esitetään verkkosivulla.



Kuvio 1. Palautustapahtuman prosessikaavio

On tärkeä huomata, että yllä oleva prosessikaavio ei ole sama kuin työjärjestys. Työvaiheet menevät eri järjestyksessä kuin palautustapahtuman prosessikaavio, sillä Raspberyllle tarvitsee määrittellä SQL-tunnukset, joita ei voi määrittää ennen SQL-palvelinta. Työvaiheet käydään erikseen läpi omissa alaluvuissaan, mutta seuraavaksi on lyhyesti listattuna työvaiheiden toteutusjärjestys:

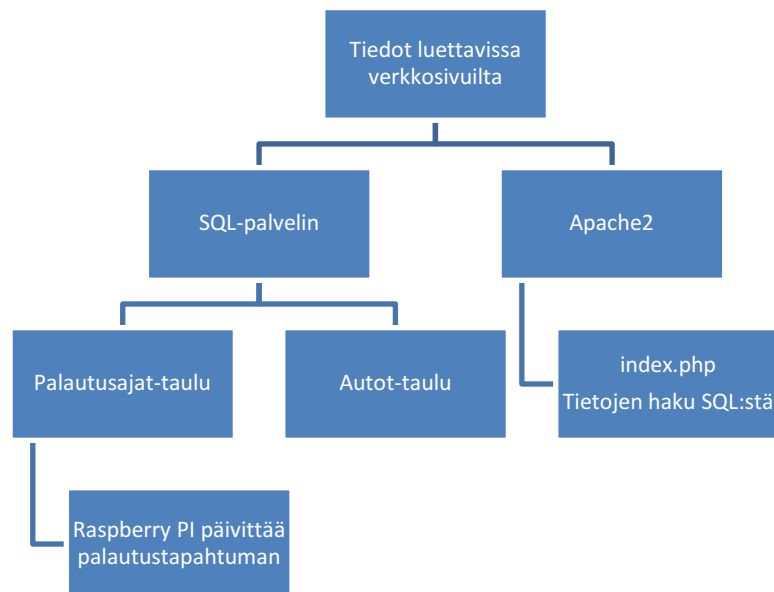
- pilvipalvelun vuokraus ja esiasennus
- tietokantojen määrittely pilvipalvelimelle
- apache2-asennus ja verkkosivujen tuottaminen pilvipalvelimelle
- Raspberryn asennus ja Python-skriptien määrittäminen.

### 3 JÄRJESTELMÄKOKONAISUUS

#### 3.1 Pilvipalvelimen määrittely ja vaatimukset

Tietokanta mistä löytyvät autojen tiedot sekä palautusajat, ei tarvitse paljon levytilaa tai prosessoritehoa. Etsin pilvipalvelualustaa, joka tukee Ubuntu Linuxia, ja antaa täyden hallinnan palvelimeen. Valitsin Kamateran, sillä näiden kahden vaatimuksen lisäksi, Kamatera tarjoaa mahdollisuuden valita käytettävän palvelimen fyysinen sijainti. (Kamatera 2022a.) Fyysisellä sijainnilla voi olla merkitystä, mikäli järjestelmää yhdistetään suoraan asiakastietoihin (Tietosuojavaltuutetun toimisto 2022).

Tässä toteutuksessa ei kuitenkaan yhdistetä järjestelmää suoraan olemassa oleviin järjestelmiin, joten asiakastietoja ei tarvitse miettiä enempää. Hierarkiakaavio (Kuvio 2) auttaa hahmottamaan palautusjärjestelmän toiminnallisuuksia.



Kuvio 2. Hierarkiakaavio järjestelmän osista

Valitsin Kamateralta Ruotsissa sijaitsevan palvelimen, ja itse palvelimen asensi Kamatera. Pilvipalvelimena vuokrasin Linux Ubuntu, 2048 MB:n keskusmuistilla ja kahdella prosessorilla. Kamatera tarjoaa julkisen IP-osoitteen palvelimilleen, joten tein asennukset SSH-yhteydellä. Veloitus tapahtuu aikaperusteisesti, ja itse

datasiirtoa palvelimelle tai palvelimelta pois on niin vähän, että arvioitu lasku kuukaudelta on alle kymmenen euroa. (Kamatera 2022b).

### 3.2 Tietokannat pilvipalvelimella

Käytin prosessikaaviota eri vaiheiden mallintamiseen. Asennukset suoritettiin prosessikaavion (Kuvio 3) mukaisesti, ja testattiin syöttämällä tietokannan tauluihin.

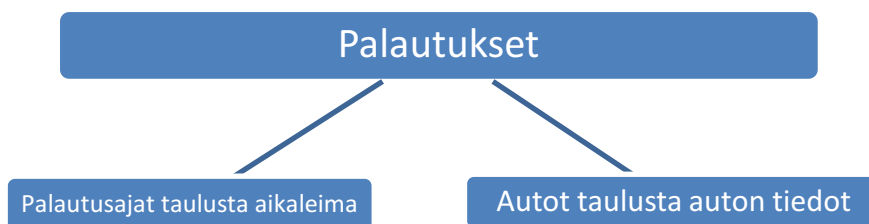


Kuvio 3. Tietokantapalvelimen prosessikaavio

MariaDB:n asennuksessa on huomioitava, että MariaDB:n pitää tukea SSH-yhteyksiä. Tämä onnistuu määrittämällä MariaDB:lle komennolla

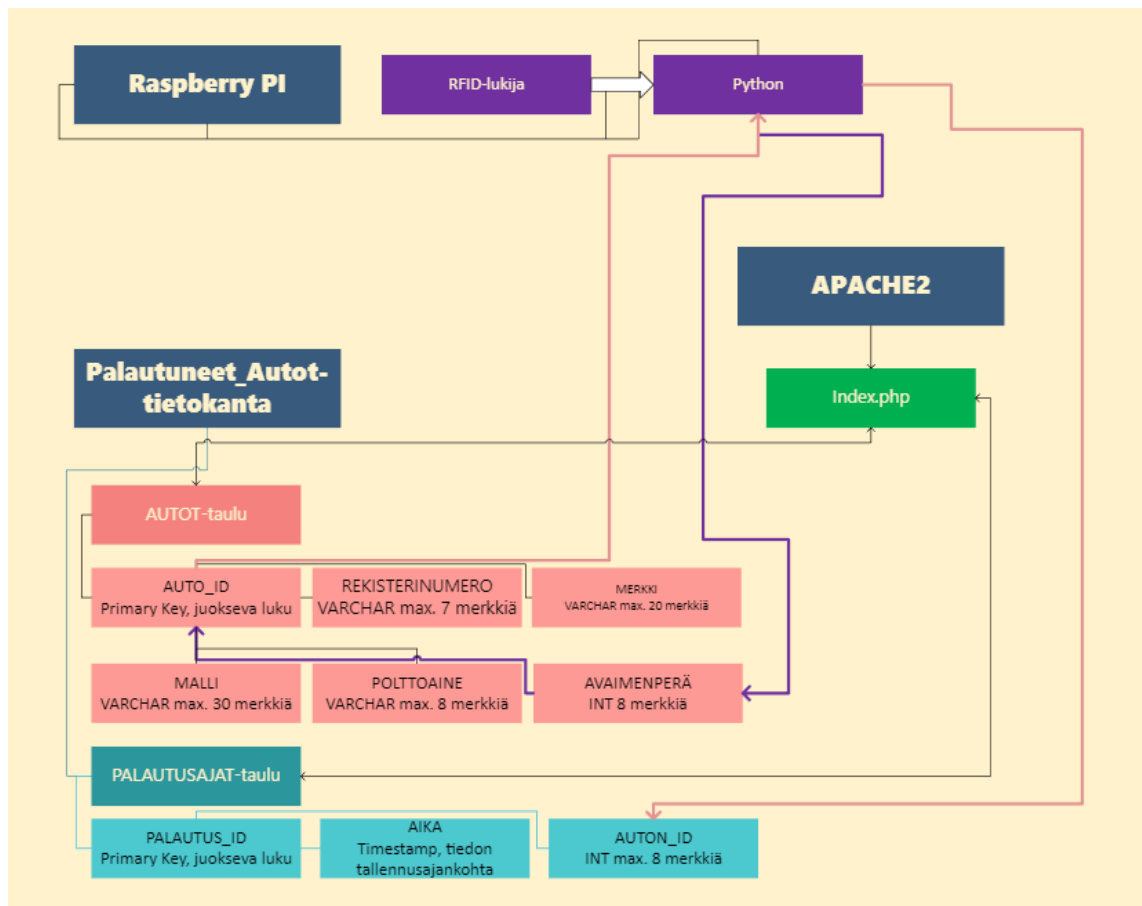
*mysql\_secure\_installation*

Samalla voidaan määrittellä root-tunnukset tietokantaan, ja muut etähallintaominaisuudet. Seuraavana luodaan MariaDB:lle tarvittavat tietokannat ja taulut. Yksinkertaisuudessaan tietokannan rakenne näyttää seuraavalta kaaviokuvulta (Kuvio 4).



Kuvio 4. Tietokantapalvelimen hierarkia

Palvelimelle luodaan tietokanta ja siihen tietokantaan kaksi taulua: Palautusajat ja Autot. Palautusajat-tauluun kirjautuu vain yksilöllinen palautus ID, auton ID ja kellonaika sekä päivämäärä, milloin auto palautui. Yksinkertaisella tietokanta rakenteella mahdollistetaan laajennettavuus ja päivitettävyyys myöhemmin. Sijoittamalla kaikki autot erilliseen Autot-tauluun, voidaan avaimenperälle päivittää toinen auto, ilman että Raspberryä tarvitsee uudelleen konfiguroida. Autot-tauluun määritellään auton ID, rekisterinumero, auton merkki, auton malli, polttoaine ja avaimenperän tunnistenumero. Ensisijaisena avaimena käytetään auton ID-saraketta. ER-malli (Kuvio 5) auttaa hahmottamaan yhteydet tietokannan taulujen ja fyysisten osien välillä.



Kuvio 5. ER-malli tietokantojen rakenteesta ja järjestelmästä

Kun taulut on luotu, voidaan tarkistaa niiden määrä tietokannasta komennolla SHOW tables; ja eheys syöttämällä tauluihin testitietoja. Palautusajat-tauluun ei tarvitse kirjata muuta kuin auton ID. Tietokanta itse antaa palautustapahtumalle

ID:n ja täyttää kirjaushetken kellonajan. Tietojen kirjautumisen tarkistaminen voidaan tehdä kyselyllä:

```
SELECT * FROM [taulun nimi];
```

Tietojen tuontiin verkkosivulle voidaan käyttää root-tunnusta, mutta on parempi luoda uudet tunnukset MariaDB:ssä mitä käytetään PHP:lle. Näille tunnuksille täytyy myöntää erikseen oikeudet tietokantaan mistä tietoja tuodaan. Kommentoimilla oikeudet voidaan myöntää

```
GRANT ALL PRIVILEGES ON [tietokannan nimi]. TO '[PHP-käyttäjä nimi]@localhost IDENTIFIED BY '[PHP-käyttäjän salasana]';
```

Tämän jälkeen tietokannalle päivitetään äsken myönnetty oikeudet komennolla

```
FLUSH PRIVILEGES; (Chartio 2022.)
```

### 3.3 Apachen asennus

Ennen Apachen asennusta on hyvä varmistaa SSH-yhteyden toimivuus tietokantapalvelimelle. Testauksen voi suorittaa ottamalla SSH-yhteyden omalta työpöydältä pilvipalvelimelle. Apachen perusasennus suoritetaan samoin kuin muillekin Linux Ubuntu jakeluille

```
apt install apache2
```

Apachen sallimiseksi palomuurin läpi käytetään komentoa

```
ufw allow 'Apache'
```

Sen jälkeen index.html on tavoitettavissa julkisesta verkosta, kun tietää pilvipalvelimen IP-osoitteen.

Verkkosivuja varten asennetaan paketti

```
php libapache2-mod-php php-mysql
```

Tällä paketilla verkkosivuilla käytettävä PHP-skripti yhdistää SQL-palvelimeen ja hakee tietoja palvelimelta. Asennuksen jälkeen PHP-Version voi tarkistaa komentamalla

```
php -v
```

Luodaan uusi index.php tiedosto polkuun /var/www/html. Oletuksena polusta löytyvän index.html tiedosto poistetaan. Index.php-tiedostoon määritellään haettavat tiedot, ja tuodaan ne taulukkona verkkosivulle. Tietokantaan yhdistäminen tehdään PHP:ssä sisäänrakennetulla operaattorilla

```
mysqli_connect.
```

SQL-syntaksissa haetaan palautusajoista löytyvillä auton ID-tunnuksilla tietoja autot-taulusta, ja määritellään tietojen järjestys seuraavassa taulukossa.

Taulukko 1. Tiedot verkkosivulla



AUTON_ID	REKISTERINUMERO	MERKKI	MALLI	POLTTOAINE	AIKA
6	ABC-678	VOLVO	V70	DIESEL	2022-11-05 09:42:18
1	ABC-123	TOYOTA	COROLLA	95E10	2022-11-02 19:14:28
3	ABC-345	TESLA	MODEL S	SAHKO	2022-11-02 19:14:28
6	ABC-678	VOLVO	V70	DIESEL	2022-11-02 19:14:28
2	ABC-234	NISSAN	QASHQAI	DIESEL	2022-11-02 19:14:28
4	ABC-456	HONDA	CIVIC	ETANOLI	2022-11-02 19:14:28

### 3.4 Raspberry PI ja skriptit

Raspberry PI asennetaan ilman graafista käyttöliittymää. Graafisella käyttöliittymällä asennetussa Raspberryssä on sisäänrakennettuja virransäästöominaisuuksia, joita ei voi ottaa pois käytöstä. Käyttöjärjestelmän tulee olla 64-bittinen, sillä Pythonin yhdistäminen SQL-palvelimelle ei toimi 32-bittisellä käyttöjärjestelmällä. Vaatimukset Raspberryn asennukselle ovat seuraavaksi taulukoituna

(Taulukko 2). Näppäimistöasettelu tulee asettaa omaa kokoonpanoa vastaavaksi. Langaton verkkoyhteys on mahdollista, mutta asennusvaiheessa käytetään langallista yhteyttä. Langattoman verkkoyhteyden voi mahdollistaa myöhemmin `raspi-config`-toiminnolla.

Taulukko 2. Raspberry PI:n vaatimukset

Käyttöjärjestelmä:	Raspbian 64-bittinen ilman graafista käyttöliittymää (Lite).
Näppäimistöasettelu:	Suomi 105 Windows näppäimillä.
Internet-yhteys:	Langaton tai langallinen
SSH:	Etähallinta SSH-yhteydellä

Asennuksen määrytykset asetetaan ensimmäisellä käynnistyksellä. Jos määrytyksiin tulee virhe, voidaan määrytyksiä myöhemmin muuttaa komennolla

*`sudo raspi-config`*

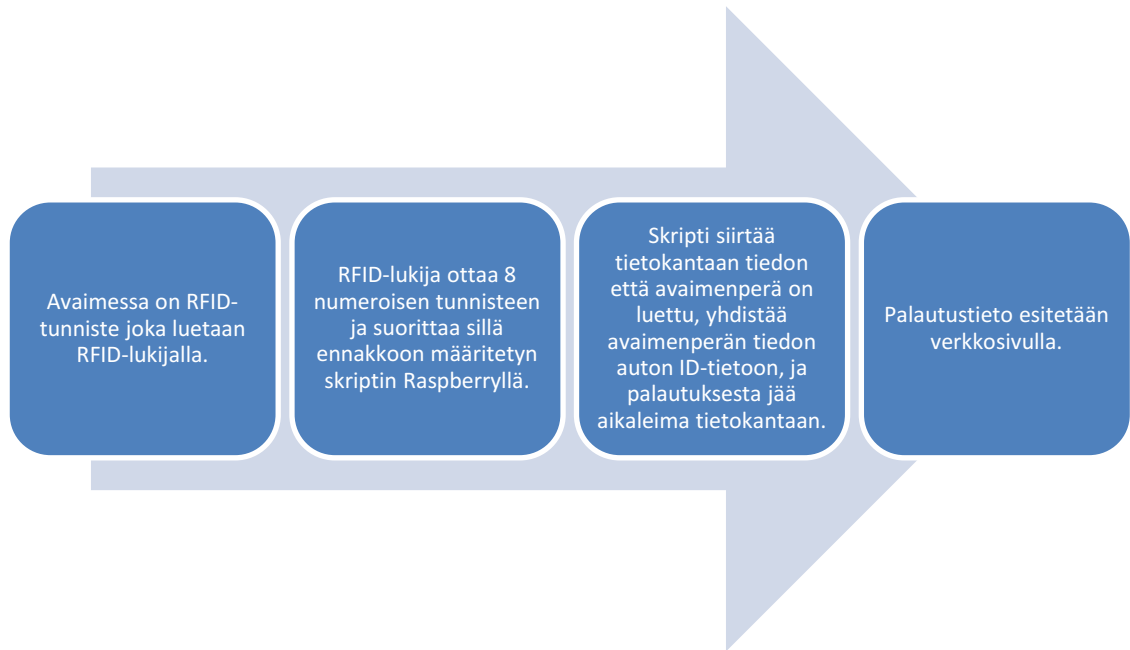
Raspberry tarvitsee Pythonille lisäpaketteja. Pythonin perusasennus ei tue SSH-yhteyksiä eikä SQL-kyselyitä. On hyvientapojen mukaista tuoda paketit Python-skriptiin alussa (Python 2022). Paketit mitä asennetaan, ovat seuraavassa taulukossa (Taulukko 3).

Taulukko 3. Asennettavat paketit

<code>sudo apt-get install python3-pip</code>	Pythonin asennuspakettien asentamiseen
<code>sudo pip3 install mysql-connector</code>	Pythonin yhdistämiseen SQL:ään pilvipalvelimella
<code>sudo apt remove libmariadb3 libmariadb-dev &amp;&amp; sudo apt autoremove</code>	Vanhojen MariaDB-kirjastojen poistamiseen yhteensopivuuden varmistamiseksi
<code>wget https://downloads.mariadb.com/MariaDB/mariadb_repo_setup</code>	Uusien MariaDB-kirjastojen lataaminen
<code>sudo chmod +x mariadb_repo_setup</code>	Suoritusoikeudet MariaDB-kirjastojen asentamiselle
<code>sudo ./mariadb_repo_setup --mariadb-server-version="mariadb-10.6"</code>	MariaDB 10.6 asentaminen
<code>sudo apt install libmariadb3 libmariadb-dev</code>	Kirjastojen asentaminen MariaDB:lle
<code>sudo pip3 install mariadb</code>	Pythonin yhteensopivuus MariaDB:lle

Pakettien asentamisen jälkeen Raspberry tulee konfiguroida käsittelemään palautustietoa. Seuraavassa prosessikaaviossa (Kuvio 7) on määritelty tapahtumajärjestys palautustapahtumalle.





Kuvio 7. Raspberryn tapahtumajärjestys

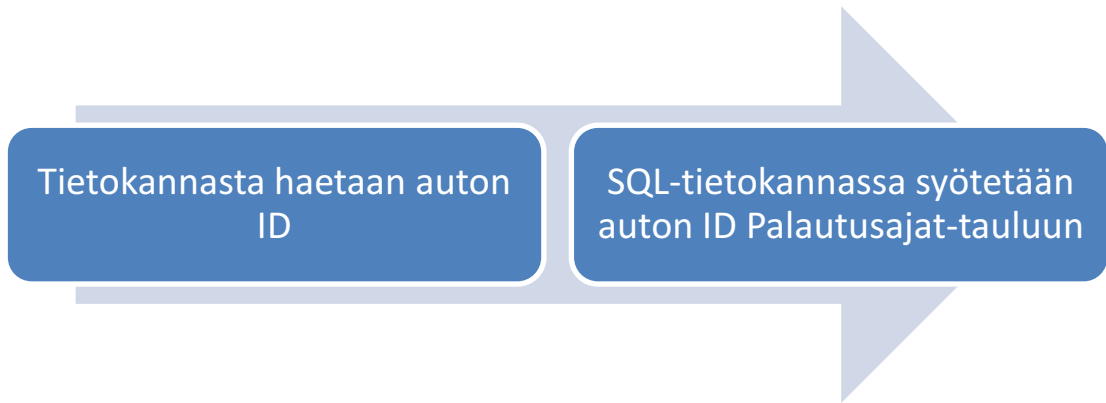
Näiden tapahtumien ulkopuolelle jää julkisesta verkosta etäyhteyden määrittäminen Raspberryyn. Tämä tehdään Dataplicity-palvelulla, joka tukee verkkoselaimen kautta tapahtuvaa Raspberry PI:n hallintaa. Ohjeet Dataplicityn käyttöön löytyvät Dataplicityn verkkosivuilta. (Dataplicity 2022).

Aliasten määrittely Raspberrylle tapahtuu suoraan komentoriviltä avaamalla tiedoston `.bash_aliases`. Jokaiselle avaimenperälle lisätään oma alias, joka antaa komennon `python3 [avaimenperän numerosarja].py`. Esimerkiksi avaimenperälle 12345678, syötetään tiedot seuraavalla komennolla

```
alias 12345678='python3 12345678.py'
```

Kun avaimenperä luetaan RFID-lukijalla, se käynnistää skriptin joka on määritelty tämän avaimenperän numerosarjalle. Aliasten kirjoittamisen jälkeen, Raspberry pitää uudelleen käynnistää. Seuraavana määritellään Python-skriptit.

Python-skriptien määrittelyä varten käytetään oheista prosessikaaviota (Kuvio 8). Skripteillä määritellään yhteys tietokantapalvelimeen, haetaan tiedot autosta ja lisätään palautustapahtuma.



Kuvio 8. Python-skriptin tapahtumajärjestys

Pythonilla tietojen lukeminen SQL-palvelimelta vaatii aiemmin määriteltyjen Python-pakettien asennukset. Kun paketit on asennettu, ne tuodaan Python-koodiin, ja muodostetaan yhteys tietokantapalvelimeen. Pythonin yhdistäminen SQL-palvelimeen on laajasti käsiteltyä internetissä, mutta tässä tapauksessa SQL-yhteys muodostetaan SSH-tunnelin kautta. (BroadOak Data 2020). Ennen SSH-tunnelin avausta asennetaan paketit mitkä ovat määriteltyinä taulukossa 2.

Tuodaan SSHTunnel-paketti Pythoniin, avataan yhteys pilvipalvelimelle ja pilvipalvelimella avataan tietokanta. Tästä tietokannasta haetaan avaimenperälle määritelty auton ID, ja lisätään tämä auto palautuneeksi Palautusajat-tauluun. Sen jälkeen yhteys palvelimelle suljetaan. SQL-kysely millä kysely suoritetaan, on seuraava:

```
"insert into PALAUTUSAJAT(AUTON_ID) select AUTO_ID from AUTOT where AVAIMENPERA=12345678"; (W3Schools 2022).
```

Jokainen avaimenperä tarvitsee oman aliaksen ja oman Python-skriptin, joka on kytketty aliakseen. Helpoin tapa toteuttaa tämä, on ottaa yksi toimiva Python-skripti ja luoda sen pohjalta uusi skripti uuden avaimenperän tiedoilla. Autot tauluun määritellään jokaiselle autolle oma avaimenperä. Mikäli RFID-lukijaan keikelee antaa jonkin muun kuin .bash\_aliases-tiedostoon määritetyn numerosarjan, konsoli antaa virheilmoituksen:

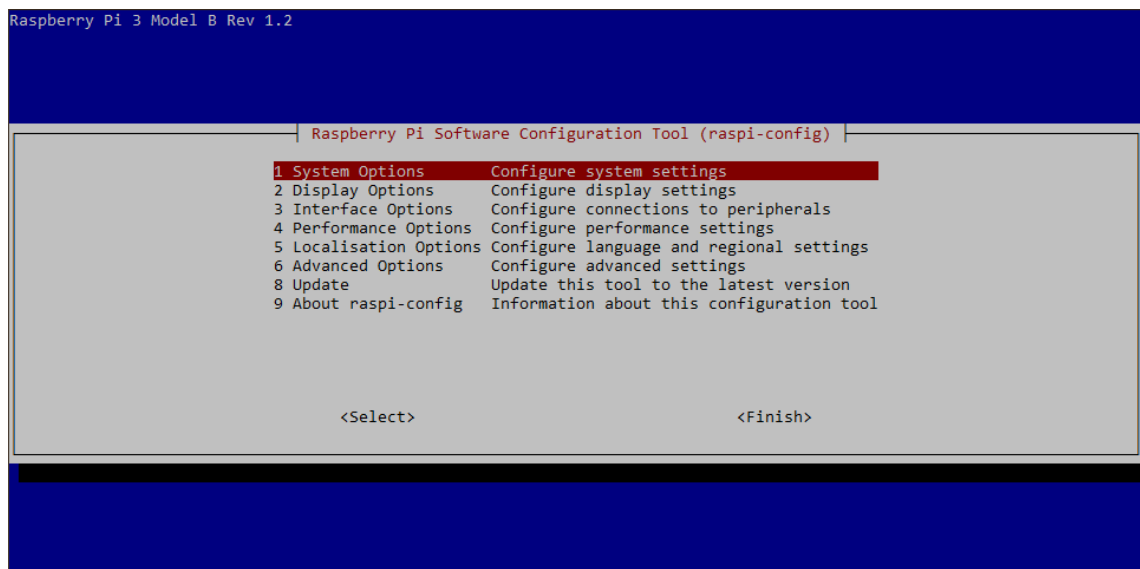
```
bash: [virheellinen numerosarja]: command not found.
```

### 3.5 Hallinta ja ylläpito

Järjestelmän ollessa käytössä, ylläpito suoritetaan etänä. Mikäli Raspberrylle tarvitsee lisätä avaintunnisteita, tai laitetta tarvitsee päivittää, on huoltotoimet helppointa suorittaa Dataplicityn kautta. Yhteys Raspberryn onnistuu muodostaa selaimesta, kun Dataplicity on asennettu laitteelle. Kirjautumalla Dataplicity-tilille, voidaan valita Raspberry ja avata suoraan konsolinäkymä.

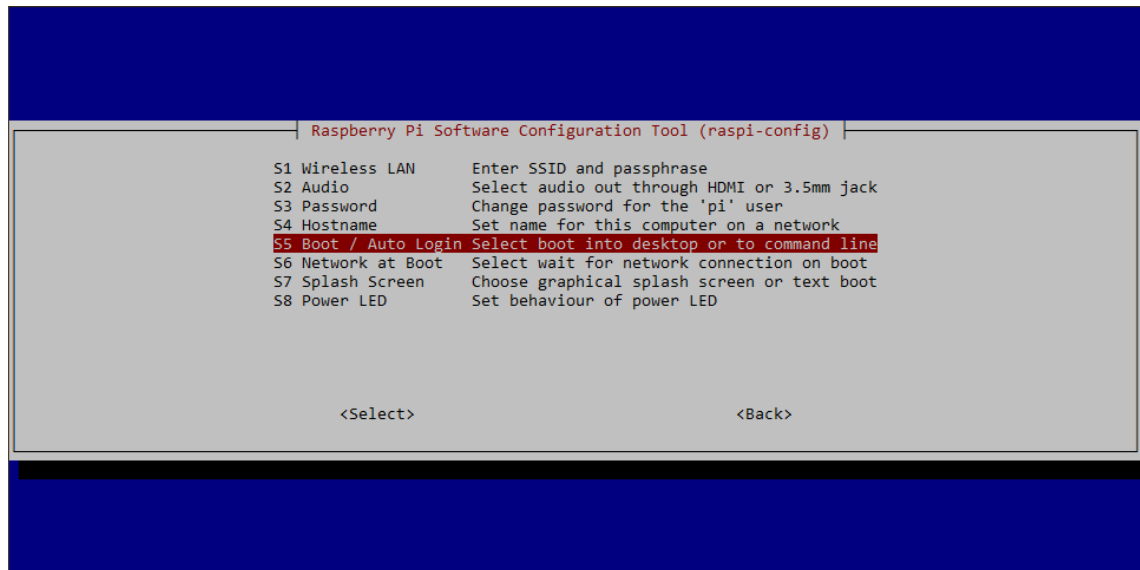
Ennen etähallinnan varmistamista, on hyvä asettaa automaattinen sisäänkirjautuminen. Raspberry pyytää oletuksena käyttäjätunnusta ja salasanaa uudelleenkäynnistyksen yhteydessä. Tämä ominaisuus tarvitsee ottaa pois, sillä uudelleenkäynnistys konsoliin täytyy tapahtua automaattisesti, muuten mahdollisen sähkökatkon jälkeen järjestelmä ei enää toimi. Automaattinen sisäänkirjautumisen mahdollistaminen tulee suorittaa Raspberrillä System Options valikosta (Kuvio 9). Raspberry Pi Software Configuration Tool (raspi-config) avautuu komennolla:

```
sudo raspi-config
```



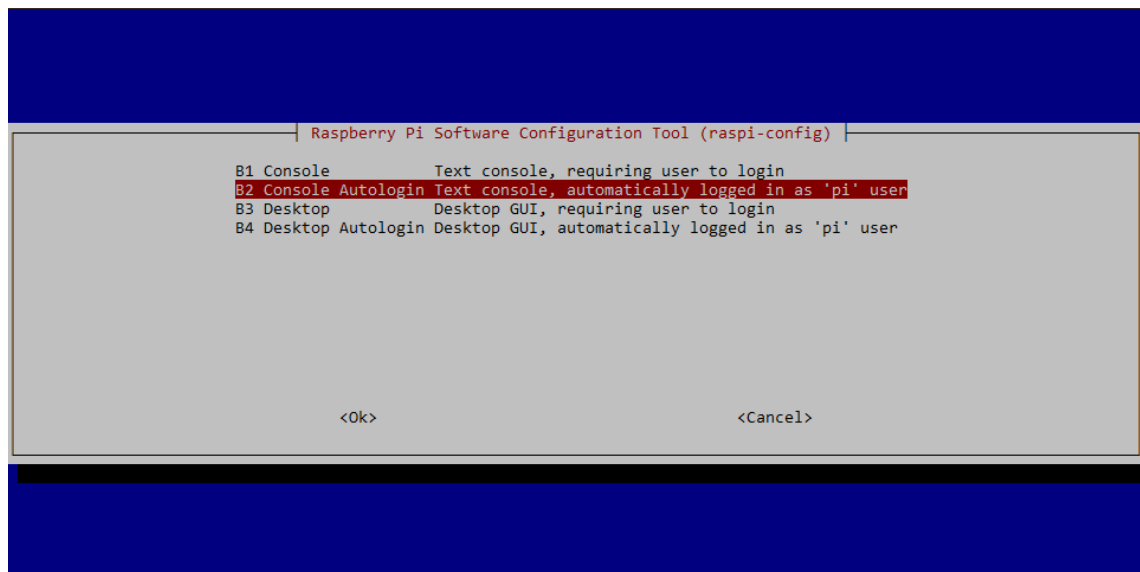
Kuvio 9. Raspi-config valikko

Raspi-config valikosta voi tehdä myös muita määrittäyksiä. System Options kohdasta (Kuvio 10) löytyvät verkkoasetukset ja käynnistysasetukset.



Kuvio 10. System Options-valikko

Boot / Auto login valikossa (Kuvio 11) tulee valita Console Autologin. Valinnat B3 ja B4 eivät toimisi, sillä kokoonpanossa ei ole graafista käyttöliittymää.



Kuvio 11. Console Autologin-valinta

Kun *Console Autologin* on valittu, suljetaan *raspi-config* ja käynnistetään Raspberry uudelleen. Dataplicityä käyttäessä on huomioitava, että oletus kansio polku konsolissa on *dataplicity@raspberrypi:/\$*.

Jotta mahdolliset muutokset ja päivitykset menevät perille ja ovat käytettävissä, pitää muutokset tehdä samalle käyttäjälle jolle Raspberry käynnistyy. Raspbian tukee toisella käyttäjällä kirjautumista konsolin kautta, komento tähän on

*su pi*

Konsoli pyytää käyttäjätunnuksen *pi*-salasanaa, joka on määritelty Raspberryn käyttöönnotossa. Kaikki muutokset laitteelle tulee tehdä *pi*-käyttäjällä.

### 3.6 Jatkokehitys

Järjestelmän toimivuus on hyvä, joten sitä ei kannata lähteä muuttamaan. Ensimmäinen iso jatkokehityskohde, on SSH-kirjautumisen vaihtaminen salasanatunnisteisesta *key*-tiedostoksi, sillä nykyisessä tunnistautumisessa tunnukset ovat luettavassa muodossa Raspberrillä. Tämä on suora tietoturva-aukko, kun kytkeväällä näytön ja näppäimistön Raspberryn voi löytää pilvipalvelimen Linux-tunnukset. Tämän vuoksi toteutus nykyisellä kokoonpanolla vaatii yhden pilvipalvelimen yritystä kohden, kun taas *key*-tiedostoilla voidaan vähentää pilvipalvelimien määrää.

Muita kehityskohteita ovat sijaintitiedon lisääminen, järjestelmän laajentaminen useammalle toimipisteelle, ja verkkosivuille tunnistautumisen lisääminen. Jatkokehityskohteiden priorisointina näen ensimmäisenä verkkosivutunnistautumisen, ja toisena SSH-kirjautumisen anonymisointi. Myös laitteiston kehittäminen pieni virtaisemmaksi olisi potentiaalinen jatkokehityskohde, optimitilanteessa kokoonpanon saisi pakattua LoRa-verkossa toimivaksi niin, että palautuspaikka voisi olla, vaikka erämaamökillä (Digita 2022). Raspberryn monipuolisuuden ansiosta hallintaa voi laajentaa GPIO-pinneillä myös kulunvalvontaan ja muuhun turvallisuuteen.

#### 4 POHDINTA

Lopputuloksena opinnäytetyöstä tuli erittäin toimiva automaattinen järjestelmä, joka ilmoittaa, kun avaimet ovat palautuneet. Järjestelmä perustui aiemmin kehittämälleni avainten palautusjärjestelmälle, mutta kuten opinnäytetyön tavoitteena oli, on huomattavasti joustavampi ja helpommin ylläpidettävä. Järjestelmän toimivuutta voisi vielä varmistaa varavirtalähteellä, mutta juuri sopivan kokoista varavirtalähdettä ei ollut saatavilla opinnäytetyötä suorittaessani.

Raspberry PI lukee RFID-tunnistetiedon, käynnistää Python-skriptin ja päivittää SQL-palvelimelle palautusajan. Apachella oleva PHP-tiedosto hakee palautusajan, auton tunnisteen ja auton lisätiedot, ja tuo ne näkyville verkkosivulle. Mikäli RFID-lukijalle annetaan avaimenperä, jota ei ole rekisteröity järjestelmään, ei vahinkoa pääse tapahtumaan. Lukunopeus on suuri ja luettavan tiedon määrä pieni, joten sekä skriptit että virheilmoitukset suoritetaan nopeasti.

Opinnäytetyötä tehdessä sain syvennettyä osaamistani Python-ohjelmoinnissa, sekä SQL-tietokannan määrittelyssä. Haasteina mainittakoon *MariaDB connector*:in toimimattomuus 32-bittisessä ympäristössä, ja tämän havaitseminen vei aikaa. Kun *MariaDB connector*-moduuli siirrettiin 64-bittiselle alustalle, Python-skriptillä sai yhdistettyä tietokantaan ja kyselyitä suoritettua.

Ennen prototyypin valmistamista en ollut aiemmin havainnut Raspberryn virransäästöasetuksien joustamattomuutta. Tutustuin laajasti Raspberry PI:n käyttöjärjestelmään selvittäessäni virranhallintaa, automaattista käynnistystä ja muita ominaisuuksia. Koska Raspbian-käyttöjärjestelmä perustuu Linuxille, sain tästä osaamista, joka hyödyttää myös lukuisissa muissa Linux-järjestelmissä.

Järjestelmän jatkokehitystä voisi tehdä siirtämällä hallintaa enemmän selainpainotteiseksi. Autojen lisääminen Autot-tauluun voisi tapahtua selaimella, ja verkkosivulle voi lisätä salasanalla tunnistautumisen. Lisäksi järjestelmän liiketoiminnallinen heikkous on RFID-lukijoissa. On lukijoita, jotka lukevat avaimenperän kahdeksan numeroa ja lisäävät "Enterin" loppuun, kuin myös lukijoita, jotka vain lukevat kahdeksan numeroa. Mikäli RFID-lukijoita tilaisi massana, pitäisi lukijoiden toiminnallisuudesta saada varmuus ennen tilausta, ja tarvittaessa muuttaa Python-skripti toimimaan ilman Enteriä.

## LÄHTEET

Broadoak Data 2020. Accessing MySQL Database Through A SSH Tunnel. Viitattu 05.11.2022 <http://broadoakdata.uk/accessing-mysql-database-through-an-ssh-tunnel/>.

Chartio 2022. How to Grant All Privileges on a Database in MySQL. Viitattu 4.11.2022 <https://chartio.com/resources/tutorials/how-to-grant-all-privileges-on-a-database-in-mysql/>.

Dataplicity 2022. Etusivu. Viitattu 19.11.2022 <https://www.dataplicity.com/>.

Home assistant 2022. Unable to load mariadb module. Viitattu 05.11.2022 <https://community.home-assistant.io/t/unable-to-load-the-mariadb-module/459791/2>. TARKISTA VIITTAUS.

Kamatera 2022a. Cloud Servers - Pricing. Viitattu 19.11.2022 [https://www.kamatera.com/Products/201/Cloud\\_Servers#page\\_250](https://www.kamatera.com/Products/201/Cloud_Servers#page_250).

Kamatera 2022b. Company Profile. Viitattu 19.11.2022 [https://www.kamatera.com/Global\\_Data\\_Centers](https://www.kamatera.com/Global_Data_Centers).

Tietosuojavaltuutetun toimisto 2022. Henkilötietojen siirrot Euroopan talousalueen ulkopuolelle. Viitattu 19.11.2022 <https://tietosuoja.fi/henkilotietojen-siirrot-etan-ulkopuolelle>.

W3Schools 2022. SQL INSERT INTO statement. Viitattu 28.11.2022 [https://www.w3schools.com/sql/sql\\_insert.asp](https://www.w3schools.com/sql/sql_insert.asp).

Python 2022. 6. Modules. Viitattu 28.11.2022 <https://docs.python.org/3/tutorial/modules.html>.

Digita 2022. IoT:n kartta. Viitattu 29.11.2022 <https://www.digita.fi/iotn-kartta/>.