

Joonas Kärkkäinen, TTK21SD

Pelitiedon tallennus ja visualisointi VPN-yhteydellä



Tradenomi
Tietojenkäsittely
Kevät 2025



KAMK • University
of Applied Sciences

Tiivistelmä

Tekijä(t): Kärkkäinen Joonas

Työn nimi: Pelitiedon tallennus ja visualisointi VPN-yhteydellä

Tutkintonimike: Tradenomi, Tietojenkäsittely

Asiasanat: suunnittelu, tietokannat, tietoverkot, visualisointi

Tämän opinnäytetyön tarkoitus oli etsiä toimiva ratkaisu pelitiedon tallentamiseksi ja visualisoimiseksi VPN-yhteyksiä hyväksi käyttäen. Työnaika käytiin läpi VPN-ratkaisujen, tietokantojen ja visualisoinnin teoriaa ja suoritettiin niihin liittyvien sovellusten asennukset. VPN-yhteyksiä kannattaa hyödyntää tallennusmekaniikoissa, sillä se salaa yhteyden, joka kulkee tietolähteestä tietokantaan. Tällöin myös tieto kulkee salattuna yhteyden läpi.

Teoriaosuuden aikana käytiin läpi verkkojen teoriaa, josta siirryttiin VPN-ratkaisujen teoriaan ja vertailtiin VPN-sovelluksia toisiinsa. Tämän jälkeen käytiin läpi tietokantoihin liittyvää teoriaa ja mihin tietokantoja käytetään. Lisäksi vertailtiin tietokantasovelluksia toisiinsa. Viimeinen osa teoriaa oli visualisoinnin teoria ja sovellusten vertailu.

Käytännön osuudessa suoritettiin projektiin liittyvät asennukset VPN-sovelluksen, tietokannan ja visualisointisovelluksen osalta. Asennuksiin kuului WireGuard-sovelluksen asentaminen kahdelle koneelle ja sovelluksien konfigurointi yhteyden luomiseksi, MongoDB-tietokannan asennus ja konfigurointi, tietokannan suunnittelu ja indeksointi, MongoDB Compass sovelluksen asennus ja konfigurointi ja visualisoinnin luominen. Lisäksi tallennettiin erillisellä sovelluksella satunnaista tietoa tietokantaan, minkä jälkeen voitiin tehdä tietokannan suunnittelu. Satunnainen tieto oli mongodbrdg sovelluksella luotuja teennäisiä henkilötietoja, mitkä pitivät sisällään nimiä, koordinaatteja ja aikamääreitä. Viimeisenä luotiin visualisointia tietokannan tiedoista. Visualisointiin käytettiin MongoDB Compass -sovellusta, jolla pystyttiin luomaan karttavisualisointi tietokantaan tallennetuista koordinaateista.

Lopuksi käytiin läpi, mikä opinnäytetyössä meni hyvin, kuten WireGuard-sovelluksen asennus ja yhteyksien luonti sekä MongoDB-tietokannan asennus, ja mikä olisi voinut mennä paremmin, kuten tiedon tallennus tietokantaan ja tiedon visualisointi. Lisäksi mainittiin, että projektia voisi kehittää eteenpäin asentamalla opinnäytetyössä käydyt järjestelmät oikeaan ympäristöön.

Abstract

Author(s): Kärkkäinen Joonas

Title of the Publication: Storing and visualization of game data using a VPN connection

Degree Title: Bachelor of Business Administration, Business Information Technology

Keywords: design, databases, information networks, visualization

The objective of this thesis was to find a working solution to store and visualize game data using VPN connections. The thesis includes a theory section of VPN solutions, databases, and visualization and completed software installations that correspond to those subjects. VPN solutions can be utilized during the storing of data, because it encrypts the connection between the data source and the database, which means that the information is also encrypted.

The theoretical section introduced some theory of information networks, from which the thesis moved on to the theory of VPN solutions and compared multiple VPN software. After that, it moved on to the theory and use cases of databases. In addition, multiple database software were compared. The last part of the theory section goes through the theory of visualisation with a comparison of multiple visualisation software.

In the practical section, the installation procedures of the chosen VPN, database and visualization software were discussed. The installations included the installation of WireGuard software on two machines and configuration of the software to create a connection, installation and configuration of a MongoDB database, designing and indexing of the database, installation and configuration of MongoDB Compass and creation of the visualisation. In addition, a separate software was used to store random data into the database, after which the database was designed. Mongodbrg was used for creating the random fake personal data, which contained names, coordinates and timestamps. Lastly, a visualization was created using the stored data from the database. MongoDB Compass was used for visualization, with which map visualization could be created using the stored coordinates from the database.

The final section includes a reflection of the successes, like installation of WireGuard and creation of connections and installation of MongoDB, and development needs, like storing the data to the database and data visualization. In addition, it was mentioned that the project could be developed further by installing the systems to a real environment.

Sisällys

1	Johdanto	1
2	Tietoverkot	2
3	VPN	3
3.1	VPN-tyypit	3
3.2	VPN-protokollat.....	4
3.3	VPN-vaihtoehdot.....	6
4	Tietokanta.....	8
4.1	Tietokantojen käyttö	8
4.2	Tietokannan suunnittelu	8
4.3	Tietokantapalvelimen valinta.....	12
5	Tiedon visualisointi	14
5.1	Visualisointitavat	14
5.2	Visualisointisovelluksen valinta.....	17
6	Työnteko.....	19
6.1	VPN-yhteyden luominen	19
6.1.1	Tietokantakoneen ja VPN sovelluksen asennus.....	20
6.1.2	Peli- ja Visualisointikoneen VPN-asennus.....	23
6.2	Tietokantapalvelimen asennus.....	26
6.3	Tiedon tallennus tietokantaan	28
6.4	Tietokannan suunnittelu	29
6.5	Visualisointisovelluksen asennus ja tiedon visualisointi	31
7	Yhteenveto	33
	Lähteet	34

Symboliluettelo

AES – Advanced Encryption Standard. Lohkosalausmenetelmä, jota käytetään tietotekniikassa.

ChaCha20 – Salausprotokolla.

DC-Labra – Kajaanin ammattikorkeakoulun datacenter-koulutuksen pääluokka.

Diffie-Hellman – Avaimenvaihtoprotokolla.

ECDSA – Elliptic Curve Digital Signature Algorithm. Sähköisen allekirjoituksen menetelmä.

Ed25519 – Salausprotokolla. Protokollan nimi on yhdistelmä EdDSA ja Curve25519 protokollien nimistä.

GPG – GNU Privacy Guard. Salausohjelmisto.

IP-osoite – (engl. Internet Protocol) internetprotokollaosoite. IP-osoitteita käytetään tietokoneiden välisessä viestinnässä.

JSON – JavaScript Object Notation. Tiedostomuoto

KamIT – Kajaani kaupungin ja koulujen tietohallinto, joka ylläpitää tietojärjestelmiä.

MPPE – Microsoft Point-to-Point Encryption. Salausprotokolla.

RSA – Julkisen avaimen salausalgoritmi, jonka nimi tulee kehittäjien sukunimien alkukirjaimista.

Site-to-Site – Kahden verkon välinen yhteys.

SQL – Structured Query Language. Standardoitu kyselykieli, jolla tehdään kyselyitä tietokantaan.

SSL/TLS – Secure Sockets Layer/Transport Layer Security. Salausprotokolla.

TCP – Transmission Control Protocol. Tietoliikenne protokolla tietokoneiden väliseen tiedonsiirtoon.

UDP – User Datagram Protocol. Yhteydetön tietoliikenne protokolla tietokoneiden väliseen tiedonsiirtoon.

VPN – Virtual Private Network. Virtuaalinen erillisverkko.

1 Johdanto

Loppuvuodesta 2023 useamman opiskelijan ryhmä alkoi miettiä robottitaistelupelin tekemistä Kajaanin ammattikorkeakoululla. Peli tehtäisiin toimimaan koululla sijaitsevalle liikealustalle ja käyttäen tavallisten näyttöjen sijaan VR-laseja, jotta peli olisi mahdollisimman immersiiivinen. Peliprojektia lähti vetämään eteenpäin yksi pelialan opettajista, jonka kanssa sovittiin, että pelistä tulee useamman vuoden kestävä projekti, johon nyt tehtäisiin konseptitodistuksia opinnäytetöiden muodossa.

Vuoden 2024 alussa tämän opinnäytetyön aihe saatiin rajattua ja sovittiin, että opinnäytetyössä luodaan VPN-yhteys koulun luokkatilasta datacenter koulutusalan konesaliin, jonne asennetaan tietokantapalvelin virtualisointiympäristöön. Tietokantaan sitten tallennetaan pelistä hahmojen koordinaatteja, olotilaa ja tyyppiä. Lisäksi luokkahuoneeseen asennetaan visualisointisovellus, jossa voidaan luoda karttavisualisointi käynnissä olevasta pelisessioista katsojien katseltavaksi tietokantaan tallentuvia tietoa hyödyntäen.

Opinnäytetyön tavoitteena on luoda konseptitodistus siitä, että pelistä on mahdollista tallentaa tietoa tietokantaan VPN-yhteyden yli ja visualisoida saatua tietoa katsojien tarkasteltavaksi. Työn aikana käytetään mahdollisuuksien mukaan ilmaisia ja avoimen lähdekoodin sovelluksia, joiden avulla projektin ympäristö pyritään rakentamaan.

Työn aluksi käydään läpi työhön liittyvien järjestelmien teoriaa. Aloitetaan verkoista ja VPN:stä, joista siirrytään tietokantoihin ja niiden suunnitteluun ja viimeisenä käydään läpi visualisoinnin teoriaa. Jokaisen teoriaosan lopuksi vertaillaan sovellusvaihtoehtoja ja valitaan niistä käyttöön otettavat sovellukset.

Työosuus on jaettu viiteen osa-alueeseen. Ensimmäisenä asennetaan VPN-yhteydet ja testataan ne toimiviksi, minkä jälkeen siirrytään tietokannan asentamiseen. Kolmantena osana tallennetaan tietokantaan tietoa, josta siirrytään sitten tietokannan suunnitteluun. Viimeisenä osa-alueena on visualisointisovelluksen asennus ja visualisointi.

2 Tietoverkot

Tietoverkot ovat järjestelmiä, jotka mahdollistavat kahden tietokoneen välisen kommunikaation. Jotta tietokoneet ymmärtävät toisiaan, kommunikaation sääntöjen tulee olla hyvin määriteltyjä. Tämä myös mahdollistaa sen, että koneet voivat kommunikoida keskenään viiveettömästi. Koneiden viestit kulkevat yleensä kuparikaapeleiden kautta, joilla koneet ovat yhdistettynä kytkimeen, jonka kautta tieto kulkee oikealle vastaanottaja koneelle. [1, s. 1–2.]

Tietoverkoilla on kolme pääkategoriaa: LAN eli Local Area Network on pieni verkko, joka koostuu vain yhden kerroksen tai rakennuksen laitteista. MAN eli Metropolitan Area Network on hieman suurempi verkko, joka koostuu kaupunkialueen kaikista LAN-verkoista. WAN eli Wide Area Network on suurin näistä verkoista, joka yhdistää kaupunkien verkkoja toisiinsa. [1, s. 2.]

LAN-verkoilla on tärkeä rooli erilaisten toimijoiden jokapäiväisessä toiminnassa, sillä se yhdistää tietokoneita ja muita verkkolaitteita toisiinsa, jolloin yksi toimisto voi jakaa resursseja keskenään. Jokainen laite, joka halutaan yhdistää verkkoon, tulee liittää kytkimeen, johon verkon muutkin koneet on yhdistetty. Jos yrityksellä on työntekijöitä useammassa kerroksessa, heillä voi olla rakennuksessa useampi LAN-verkko. Nämä verkot voidaan sitten yhdistää toisiinsa käyttämällä siltaa tai reititintä. [1, s. 3–4.]

MAN-verkko koostuu useista, toisiinsa yhdistetyistä LAN-verkoista, jotka sijaitsevat yhden kaupungin tai suurkaupungin alueella. MAN-verkko on hyvä tapa yhdistää paikallisen hallinnon tai suurien yritysten järjestelmiä toisiinsa, jolloin voidaan jakaa hyödyllisiä resursseja verkosta toiseen ja kommunikoida muiden hallinnon tai yrityksen osien kanssa. Yleensä MAN-verkossa on vain yksi yhteys jokaiseen LAN:iin, sillä uuden verkko infrastruktuurin rakentaminen tai ostaminen on kallista. [1, s. 4–5.]

WAN-verkko yhdistää kahta tai useampaa LAN- tai MAN-verkkoa toisiinsa kaupunkien, maakuntien tai valtioiden välillä. Tämä vaatii koordinaatiota ja oikeanlaisen laitteiston, joten yleensä puhelinoperaattorit tarjoavat oikeanlaisen kaapeloinnin WAN-verkkoihin. Jos tarvitaan yhteys toiselle puolelle maapalloa, voidaan fyysisten kaapelien sijaan käyttää satelliittiyhteyksiä. LAN-verkkoihin verrattuna WAN-verkot ovat hitaita, mutta niidenkin yhteysnopeudet voivat olla melkein 10 gigabittiä sekunnissa, joten kuva WAN:eista hitaina verkkoina on muuttumassa nopeasti. [1, s.5.]

3 VPN

VPN eli Virtual Private Network on yleensä julkisen verkon kautta toimiva salattu yhteys laitteesta johonkin tiettyyn verkkoon. Salatulla yhteydellä varmistetaan, että lähetetty tieto on turvassa eikä kukaan ulkopuolinen pysty näkemään, millaista tietoa julkisen verkon läpi lähetetään. [2.]

VPN-yhteyksien käyttämiseen on monia syitä. Julkista verkkoa käyttäessä VPN-yhteyttä voi käyttää salaamaan omia tietojaan, koska se salaa tietosi, kuten sähköpostin tiedot tai paikannustiedot, salausavaimella, jolloin kukaan verkossa oleva ei voi tarkastella tietoja. Nykyaikana myös etätöiden tekeminen on kasvattanut suosiotaan, jolloin VPN-yhteyden avulla työntekijät voivat saada yhteyden kodistaan yrityksen verkkoon ja resursseihin. VPN-yhteyksiä käytetään useasti myös alueellisten lukitusten kiertämiseen, kun halutaan katsoa jotakin mediaa, jota ei katselu hetkellä voi katsoa omassa maassa. Tällaisessa tilanteessa katsojalle näytetään lukittua mediaa, koska VPN-palvelin sijaitsee maassa, jossa media on saatavilla ja palvelu luulee yhteyden tulevan samasta maasta. Samaa ominaisuutta voidaan käyttää myös valtioiden asettamien sensuurien ja tarkkailujen kiertämiseen. Lisäksi VPN estää internet-palveluntarjoajia tarkkailemasta IP-osoitteen avulla selaushistoriaa, joka voitaisiin sitten yhdistää johonkin käyttäjään. [3.]

3.1 VPN-tyypit

Maailmasta löytyy monia erityyppisiä VPN-ratkaisuja erilaisiin käyttötarkoituksiin. Osa on tarkoitettu käytettäväksi tietokoneella ja osa puhelimilla, jotkut ovat maksullisia ja toiset ilmaisia ja osa on suunniteltu enemmän ammattilaisten käytettäväksi ja osa ihmisten henkilökohtaiseen käyttöön. Yksi suosituimmista VPN-tyypeistä on etäyhteys-VPN, jolla yhdistetään laite suoraan yksityiseen verkkoon tai palvelimeen. Tämän tyyppisiä ratkaisuja käytetään kotoa käsin yhdistämään esimerkiksi työpaikan verkkoon, kun tehdään etätöitä. Tällaisiin VPN-palveluihin on yleensä kirjauttava sisään, jotta yhteys saadaan avattua.

Yritykset sen sijaan käyttävät Site-to-Site -VPN-ratkaisua, joka yhdistää yrityksen sisäisiä verkkoja toisiinsa pitkilläkin etäisyyksillä. Näin varmistetaan, että yrityksen sisäiset yhteydet pysyvät salatuihin. Tämän tyyppisiä VPN-ratkaisuja on kahta erilaista, intranet ja extranet. Intranet yhdistää yhden yrityksen toisista irrallaan olevia verkkoja yhteen, jotta eri osastot voivat työskennellä tehokkaasti keskenään. Extranet sen sijaan yhdistää kahden eri yrityksen LAN-verkkoja toisiinsa,

jolloin useasti keskenään työskentelevät yritykset voivat jakaa tarvittavia resursseja toisilleen. Puhelimille on vastoitain kehitetty omia VPN-ratkaisuja, jotka ylläpitävät yhteyksiä yksityiseen verkkoon, vaikka yhteydet olisivat epävakaita tai jopa katkeilevia. Puhelimille tehtyjä VPN-sovelluksia käytetään yleensä matkoilla tai silloin, kun ei voida olla varmoja verkon toimivuudesta. [3.]

3.2 VPN-protokollat

Kun käyttäjä yhdistää VPN-sovelluksella etäpalvelimeen, palvelin todentaa käyttäjän, minkä jälkeen se luo salatun tunnelin, jonka läpi kaikki tieto sitten kulkee. Tunnelin läpi kulkeva tieto salataan koodiksi, jolloin tietoa on mahdotonta lukea, jos ei ole salausavainta. Kun tieto saapuu palvelimelle, salaus puretaan palvelimen omalla yksityisellä salausavaimella, jolloin tiedosta tulee jälleen luettavaa. Kun tiedon salaus on purettu, VPN-palvelin lähettää tiedon eteenpäin palvelimelle, jolle yhteyttä koitetaan ottaa. Samalla VPN-palvelin vaihtaa lähetykseen uuden IP-osoitteen. [3.]

Salausprosessin toiminta ja turvallisuus riippuvat siitä, mikä protokollatyyppi yhteydenotossa on käytössä, sillä VPN voi tarjota turvallisuutta vain, kun toiminnan takana on vahva protokolla. [3.] Seuraavaksi käyn läpi muutamia käytetyimpiä VPN-protokollia.

OpenVPN on yksi yleisimmin käytetyistä protokollista ja sitä pidetään yleisesti alan standardiprotokollana vakauden, joustavuuden ja turvallisuuden puolesta. Se hyödyntää salauksessa 256-bittistä AES-algoritmia (taulukko 1), tunneloi yhteyden SSL/TLS-yhteysprotokollalla käyttäen TCP- ja UDP-tietoliikenneprotokollien paketteja ja hyödyntää avoimen lähdekoodin teknologiaa. [3.]

SSTP on toinen yleisesti käytetty protokolla ja sitäkin pidetään alan standardina. Salauksessa käytetään 256-bittistä AES-algoritmia ja yhteys sertifioidaan SSL/TLS-yhteysprotokollan avulla käyttäen TCP-tietoliikenneprotokollan paketteja. Lisäksi SSTP on Microsoftin tukema ja tulee Windows-käyttöjärjestelmässä sisäänrakennettuna. [3.]

IKEv2 on protokolla, jota käytetään yleensä yhdessä IPSec-protokollan kanssa, joka mahdollistaa sen, että yhteyden turvallisuus ja nopeus ovat optimaalisia. Nämä protokollat yhdessä pitävät yhteyttä yllä, vaikka internet-yhteys olisi pätkivä. Yhteys salataan 256-bittistä AES-algoritmia hyödyntäen ja se toimii UDP-tietoliikenneprotokollan pakettien kautta. Yhteys voi pysyä päällä myös, kun vaihdetaan verkosta toiseen, esimerkiksi puhelinverkon ja langattoman verkon välillä. [3.]

L2TP on toinen protokolla, jota yleensä käytetään yhdessä IPSec-protokollan kanssa. Se myös on Windows-käyttöjärjestelmässä sisäänrakennettuna, kuten SSTP-protokolla. L2TP käyttää salaukseen 256-bittistä AES-algoritmia ja sen yhteys toimii UDP-tietoliikenneprotokollan pakettien kautta. L2TP on helposti asennettava protokolla, mutta monet palvelutarjoajat ovat luopuneet protokolla tukemisesta, kun parempia vaihtoehtoja on nykyaikana tarjolla. [3.]

PPTP on protokolla, joka edelsi L2TP-protokollaa, jonka vuoksi se ei ole enää laajassa käytössä. PPTP käyttää salauksessa 128-bittistä MPPE-algoritmia, joka on helppo murtaa. PPTP käyttää TCP-tietoliikenneprotokollan paketteja tiedon lähettämiseen. Lisäksi protokollassa on monia haavoituvuuksia, jonka takia sitä ei pidetä luotettavana vaihtoehtona VPN-yhteyksille. [3.]

WireGuard on uusi protokolla, jonka käyttö VPN-ympäristössä on jatkuvassa kasvussa. WireGuard mainostaa itseään kevyemmällä koodikannalla, moderneimmalla salausteknologialla ja paremmalla puhelinyhteensopivuudella. WireGuard hyödyntää salauksessaan 256-bittistä AES-algoritmia ja tiedonsiirto toimii UDP-tietoliikenneprotokollan paketeilla. OpenVPN-protokollan tavoin se on avoimen lähdekoodin projekti, jolloin jokainen käyttäjä voi tarkastella koodia ja ilmoittaa vioista. [3.]

Taulukko 1. Lista VPN-protokollista ja niiden toiminnoista [3][4].

Protokolla	Salaus	Reititys	Käyttö
OpenVPN	256 bittiä AES	TCP ja UDP	Yksi yleisimmin käytetyistä protokollista, avoimen lähdekoodin protokolla
SSTP	256 bittiä AES	TCP	Yksi yleisimmin käytetyistä protokollista, löytyy sisään rakennettuna Windows käyttöjärjestelmästä
IKEv2/IPSec	256 bittiä AES	UDP	Käytetään yleensä mobiililaitteiden VPN yhteyksiin
L2TP/IPSec	256 bittiä AES	UDP	Aiemmin yksi yleisimmistä ratkaisuista, muut ovat ohittaneet sen
PPTP	128 bittiä MPPE	TCP	Ei ole enää käytössä missään, koska suojaus on murrettu
WireGuard	256 bittiä AES	UDP	Uusi protokolla, jonka käyttöä jatkuvasti lisätään, avoimen lähdekoodin protokolla

3.3 VPN-vaihtoehdot

Ilmaisia ja avoimen lähdekoodin VPN-palvelimia löytyy useita erilaisia, joten valitsen yhden niistä projektiani varten, sillä käytän VPN-palvelinta projektissani yhteyden muodostamiseen tietokoneiden välille. Vaihtoehtoja ovat OpenVPN, strongSwan, SoftEther ja WireGuard.

OpenVPN sovelluksesta on monia vaihtoehtoja, mutta niistä muihin vertailtava versio on OpenVPN Community. Valituista vaihtoehdoista OpenVPN tukee pienintä määrää käyttöjärjestelmistä verrattuna muihin vaihtoehtoihin (taulukko 2.). Salaukseen käytetään AES-256-standardin salausta ja protokollana toimii samanniminen OpenVPN-protokolla. Tiedonsiirtonopeudeksi OpenVPN lupaa noin 1 gigabittiä sekunnissa. [5; 9.]

Toinen vaihtoehto VPN-sovellukselle on strongSwan. Se tukee useampaa käyttöjärjestelmää kuin OpenVPN, sillä tuettuna ovat myös iOS- ja Android-käyttöjärjestelmät. Diffie-Hellman-avaimenvaihtoprotokollan mukaan strongSwan käyttää salaukseen joko Ed25519- tai ECDSA-standardin salausta ja tunnelointiprotokollana toimii IKEv2.[6.]

Kolmas VPN-vaihtoehto on SoftEther. Se tukee samoja käyttöjärjestelmiä kuin strongSwan. Salaukseen SoftEtherissä käytetään AES256- ja RSA4096-standardin salauksia. SoftEther tarjoaa tuen moniin eri VPN-protokolliin tunnelin luomiseksi. Tiedonsiirron nopeudeksi SoftEther lupaa jopa 1 Gbps nopeutta vähäisellä muistin ja keskusprosessorin käytöllä. [7.]

Neljäs vaihtoehto VPN-sovellukseksi on WireGuard. Kahden edellisen vaihtoehdon tavoin se tukee useaa eri käyttöjärjestelmää. Salaukseen WireGuard käyttää ChaCha20-salausstandardia. Tunnelin luomiseen WireGuard käyttää samannimistä WireGuard-protokollaa. SoftEtherin tavoin WireGuard lupaa tiedonsiirtonopeudeksi jopa 1 Gbps nopeutta. [8.]

Valitsen näistä vaihtoehdoista projektiin käyttöön WireGuard-sovelluksen.

Taulukko 2. Lista VPN-sovelluksista ja niiden ominaisuuksista. [5][6][7][8][9]

Nimi	Käyttöjärjestelmät	Salauus	Protokollat	Tiedonsiirtonopeus
OpenVPN	Linux, Windows, FreeBSD, macOS	AES 256	OpenVPN	noin 1Gbps
strongSwan	Linux, Android, FreeBSD, macOS, iOS, Windows	Ed25519, ECDSA	IKEv2	
SoftEther	Linux, Android, macOS, iOS, Windows, FreeBSD	AES 256, RSA 4096	SSL-VPN, OpenVPN, IPsec, L2TP, MS-SSTP, L2TPv3, EtherIP	Jopa 1Gbps
WireGuard	Linux, Android, FreeBSD, macOS, iOS, Windows	ChaCha20	WireGuard	Jopa 1Gbps

4 Tietokanta

Maailmassa tieto on tärkeä resurssi ja monet yritykset ovat kokonaan riippuvaisia tietokannoistaan, joten kaikki tietojärjestelmät käyttävät tietojen tallentamiseen tietokantatekniikkaa. [10, s. 4.]

Tietokanta on kokoelma tietoa, joka on järjestelty siten, että tietoa on helppo hakea, päivittää ja hallita. Tieto voi olla tietokannassa missä muodossa tahansa. [11.] Yleensä tieto tallennetaan tietokantoihin useisiin eri taulukkoihin riveinä ja sarakkeina. Tällä saavutetaan tiedon käsittelyn ja hakemisen tehokkuus. Suuri osa tietokannoista käyttää SQL- eli Structured Query Language -kieltä tiedon hakemiseen ja tallentamiseen. [12.] Kun tietokanta on SQL-pohjainen, se tarkoittaa, että kyseessä on relaatiotietokanta. Relaatiotietokannat ovat syrjäyttäneet perinteisemmät verkko-malliset ja hierarkkiset tietokannat, sillä relaatiotietokantoja on helpompi käyttää ja muuttaa. Relaatiotietokantojen lisäksi markkinoilla on saatavilla myös oliotietokantoja, jotka eivät seuraa normalisointisääntöjä tai relaatiomallia. Oliotietokannat sopivat monimutkaisien olioiden ja niiden operaatioiden luomiseen, siksi ne ovat hyvä jatke olio-ohjelmoinnille. Ne eivät kuitenkaan ole kasvaneet merkittäväksi osaksi kaupallisten sovelluksien joukossa. [10, s. 5–6.]

4.1 Tietokantojen käyttö

Tietokannoilla on monia käyttötarkoituksia. Eri yrityksillä on tietokannoille erilaisia käyttöjä, kuten tiedon säilytys, analysointi ja hallinta. Tallennetun tiedon avulla yritykset voivat sitten tehdä informoituja liiketalouspäätöksiä, kuten parantaa yrityksen kaupallisia prosesseja. Tietokantoihin kerätään tietoa esimerkiksi myynneistä, tilauksista ja asiakaspalvelusta, minkä jälkeen tietoa voidaan analysoida ja prosesseja parantaa. Samalla voidaan tallentaa tietoja asiakkaista ja käyttäjistä sekä heidän käyttäytymisestään, joiden avulla voidaan suositella käyttäjille asioita, jotka sopivat heidän käyttäytymiseensä ja parantavat heidän kokemustaan yrityksen kanssa asioimisesta. [11.]

4.2 Tietokannan suunnittelu

Tietokannat ovat modernien sovelluksien perusta. Sovellusten käyttäjät eivät näe tietokantoja suoraan, vaan he näkevät ne käyttöliittymän ja raporttien kautta. Silti on tärkeää suunnitella ja

rakentaa sovelluksen takana toimiva tietokanta hyvin. Tämä varmistaa, että sovellus on helppo ohjelmoida, jolloin sovelluksesta tulee toimiva ja helppokäyttöinen loppukäyttäjälle. [10, s. 20.]

Moderneilla tietokantatuotteilla on erittäin helppoa ja nopeaa luoda tauluja tietokantaan. Tämä mahdollistaa sen, tietokantatauluja on helppo luoda ilman, että ne on ensiksi suunniteltu kunnolla. Tämä on riski tietokantojen toimivuuden kannalta, sillä mitä suurempi tietokanta on, sitä huonommin se toimii, jos sitä ei ole suunniteltu kunnolla. Tietokannan suunnittelu on myös laajempi asia, kuin pelkästään tietokannan mallinnus. [10, s. 20.]

Ennen tietokannan suunnittelun aloittamista on hyvä miettiä, millaiseen tietokantarakenteeseen ollaan pyrkimässä. Hyvän rakenteen tulisi olla niin kattava, että se sisältää kaiken tarvittavan tiedon, jota järjestelmät tarvitsevat, mutta samalla siinä tulisi olla yksinkertainen rakenne, jolloin tietokanta on selkeä ja siihen on helppo tehdä kyselyitä. Rakenteen täytyisi myös olla selkeä ja kaikilla sarakkeilla tulisi olla tarkoitus tietokannassa. Tietokannan tulisi myös kestää muutoksia ja soveltua mahdollisimman moneen erilaiseen ympäristöön ilman, että tietokantaa tai ympäristöjä tarvitsee muuttaa suuresti. Kannan pitäisi olla lisäksi eheä, jolloin yksittäiset tiedot eivät toistu kannassa ja tiedot ovat varmasti oikeita, jolloin tietokanta vois myös olla tehokas ja se kykenee vastaamaan kaikki pyyntöihin. [10, s. 21.]

Tietokannan suunnitteluun kuuluu viisi vaihetta, joiden kokonaisuutta kutsutaan suunnitteluputkeksi. Vaiheet ovat käsiteanalyysi, tarveanalyysi, normalisointi, tietokannan toteutus ja suorituskyvyn viritys. Kaikki nämä vaiheet kannattaa suunnittelun aikana käydä läpi, vaikka osa niistä olisi nopeita tai mahdollisesti ohitettavia vaiheita. Vaiheet eivät myöskään välttämättä seuraa toisiaan, vaan vaiheita työstetään yhtä aikaa, jolloin siirrytään hiljalleen kohti oikeaa tietokantarakennetta. [10, s. 24–25.]

Ensimmäisenä vaiheena suunnittelussa toimii käsiteanalyysi. Sen tavoitteena on havainnollistaa kaavioiden avulla tietoa, jota tietokantaan ollaan tallentamassa. Käsiteanalyysissä on tarkoitus kuvata reaali maailman osaa, jota tietokannassa on sitten tarkoitus kuvata. Tämän tuloksena syntyy käsitelmä, joka sitten kuvaa sekä kohdealuetta että määrittelee tietokannan rakennetta. Käsitelmää sitten tarkennetaan, kunnes lopulta saavutetaan niin tarkka suunnitelma, että tietokannan rakentaminen voidaan aloittaa. Analyysiä tehdessä ei tule ajatella tietokannan suorituskykyä, sillä tässä vaiheessa on tärkeämpää saada kuvattua tietokantaan haluttavat asia selkeästi. [10, s. 32.]

Seuraava suunnittelun vaihe on tarveanalyysi. Tarveanalyysin tavoitteena on lisätä yksityiskohtia käsitteisiin sen perusteella, mitä tietoa tarvitaan. Käytännössä tämä tarkoittaa, että tarkastellaan

tietoa, jota tarvitaan tietokantaa ja verrataan sitä jo tehtyyn käsittemalliin, joka luotiin käsiteanalyysin aikana. Tarkoituksena on tarkistaa, löytyvätkö kaikki tarvittavat tiedot jo käsittemallista ja täytyykö jotakin lisätä malliin. Tällä tulisi päästä tilanteeseen, jossa käsittemalli sisältää kaiken tiedon yksityiskohtaisesti ja se on todettu toimivaksi. [10, s. 80–81.]

Kolmas vaihe suunnittelussa on normalisointi. Normalisoinnin tavoitteena on jalostaa tietorakenteita parempaan muotoon, jossa tietoa toistetaan mahdollisimman vähän ja jota on helppo pitää yhdenmukaisena. Samalla tietorakenteesta on saatava tehokas, jotta päivitykset ovat nopeita ja helppoja sekä muutoksia kestävä. [10, s. 86.]

Tästä siirrytään sitten neljänteen vaiheeseen, joka on tietokannan toteutus. Tämän aikana luodaan tietokannan taulurakenne, joka perustuu edellisten vaiheiden aikana suunniteltuun käsittemalliin. Taulut muodostetaan käsittemallista muuttamalla käsittemallin objektit tietokantaobjekteiksi seuraamalla listaa ”sääntöjä”.

Ensimmäisenä listassa ovat käsitteet, mikä tarkoittaa sitä, että jokaisesta käsitteestä tulee tehdä tietokantaan taulu. Toisena listasta löytyvät tiedot, jossa varmistetaan, että jokaisen käsitteen sisällä olevista tiedoista tulee tauluihin sarakkeita. Kolmas vaihe on yksi-moneen-yhteyksien tarkastaminen. Tämän aikana katsotaan, että isä-lapsi-yhteyksissä lapsitaulun päähän tulee viiteavaimiksi isätaulun perusavain. Neljäs vaihe suunnitteluputkessa on yksi-yhteen-yhteyksien tarkastaminen. Näissä tapauksissa vain toiseen tauluista lisätään viiteavain. Yleensä se on tauluista se, jossa se on pakollinen, eli esimerkiksi jos on henkilö ja esine, niin viiteavain tulee esineeseen, sillä jokaisella esineellä on henkilö, mutta jokaisella henkilöllä ei ole esinettä. Lisäksi viiteavainsarakkeesta voidaan tehdä yksilöivä, jolloin yhteys on varmasti yksi-yhteen-yhteys. Viides vaihe on nimeäminen, jonka aikana taulut ja sarakkeet nimetään. Nimien antaminen kannattaa tehdä hyvin ja huolella, sillä nimien muuttaminen jälkeenpäin saattaa olla työlästä, koska nimiin on voitu viitata ohjelmissa ja kyselyissä. Kuudes listan säännöistä on tietotyypit ja eheyshdot. Tämän aikana sarakkeille määritetään tuotekohtaiset tietotyypit. Lisäksi määritellään tietojen pakollisuus ja oletusarvot, jos niitä on. Seitsemäntenä vuorossa listassa ovat perusavaimet. Perusavaimilla on pakko olla jokin arvo. Tietokantatauluja tehdessä pitää siis määrittää perusavainsarake niin, että se ei saa olla tyhjä. Useissa tietokannoissa järjestelmä indeksoi perusavainsarakeen automaattisesti. Kahdeksas kohta säännöissä on viiteavaimet. Viiteavaimissa pitää tarkistaa, että pakollisten yhteyksien kohdalla arvo ei saa olla tyhjä. Viiteavainsarakkeille kannattaa luoda indeksi, koska ilman indeksiä isä- ja lapsitaulujen välillä olevat yhteydet ovat hitaita. Toiseksi viimeinen vaihe listassa on viite-eheydet. Tämän aikana varmistetaan, että tietokantaa luodaan viiteavaimet oikealla tavalla riippuen siitä, mitä viite-eheysääntöä noudatetaan. Viimeinen vaihe

tietokantojen toteutuksessa on poissulkevat yhteydet. Tämän vaiheen aikana tarkastetaan, että yhteyksissä, jotka eivät voi olla olemassa yhtä aikaa, toisen yhteyden viiteavain on tyhjä, esimerkiksi, jos autolla voi olla useampi erityyppinen omistaja, niin autolla voi silti olla vain yksi omistaja, jolloin muut omistaja tyypit jäävät tyhjäksi. [10, s. 103–110.]

Tietokannan viides ja viimeinen suunnittelu vaihe on suorituskyvyn viritys. Suorituskyvyn virityksen aikana tulee tehdä tietokannan fyysinen suunnittelu ja tietokannan virittäminen. Fyysinen suunnittelu on tärkeä osa tietokannan suorituskykyä. Fyysisen suunnittelu aikana täytyy suunnitella tietokannan indeksointi, määrittää tietokantasivujen koot, määrittellä tyhjän tilan määrän tulevaisuuden lisäyksiä varten, osioida taulut ja indeksit, suunnitella tietokantataulujen tilavaraukset, suunnitella taulu- ja indeksijoukkojen mahdolliset sijoittelurajoitukset sekä määrittellä mahdolliset tietokantapuskurialtaat ja niiden rajoitukset. Tietokantaa varten täytyy arvioida tarvittavan levytilan määrä. Tietokantasuunnittelijoita voi pyytää arviota tietokannan vaatiman tilan määrästä tai CASE-työkaluun voi syöttää arvion tietokantaan tulevien rivien määrästä, jolloin se laskee arvion tarvittavasta tilasta. Lisäksi on olemassa karkea arviointimenetelmä, jonka mukaan tietokanta vie viisi kertaa isoimpien taulujen nettokoon verran tilaa, jossa nettokoko on rivien määrä kertaa keskimääräinen rivien pituus. Taulutilaa tulee varata viisinkertainen tila, sillä aiemmin mainitut fyysisen suunnittelun vaiheet kuluttavat levytilaa sen verran. [10, s. 141–143.]

Tietokannan virittämiselle on monia tapoja. Virittäminen kannattaa aloittaa indeksien suunnittelusta, sillä indeksiä paksuntamalla saavutetaan todella hyvä suorituskyvyn nousu. Tämän jälkeen taulujen rivejä voi uudelleen järjestellä suorituskyvyn parantamiseksi ja tämä voi tulla tarpeelliseksi myös myöhemmässä vaiheessa, kun taulut ja indeksit alkavat olla vanhentuneita. Tietokannan ulkopuolella sovellusta voidaan muuttaa niin, että se ei esimerkiksi tekisi turhia tietokantahakuja, jolloin suorituskyky paranee. Lisäksi sovelluksessa voidaan parantaa SQL-kyselyiden rakennetta, sillä eri sovellukset toimivat eri tavoin. Tietokannan parametrejä voidaan myös muokata säätämällä esimerkiksi tyhjiä tiloja tai väliaikaisen tietokannan kokoa. Tietokantaa voi myös denormalisoida, jolloin liitosten määrä vähenee ja tietokannan suorituskyky voi parantua huomattavasti. Tämä kuitenkin vaatii yleensä myös sovellusten muokkausta. Pidemmällä ajanjaksolla tietokannan tietoja voidaan myös summata yhteen, jolloin esimerkiksi koko kuukauden tiedot ovat yhdessä tietokannassa. Tällöin pidemmältä aikaväliltä haettujen tietojen kyselyt nopeutuvat levytilan kustannuksella. Tietokannan alla olevaan laitteistoon voi myös tulla tarvetta lisätä tehoa, jolloin tietokanta samalla tehostuu. Lisäksi yhtäaikaista lukituksia kannattaa tarkastella, sillä ne voivat aiheuttaa pullonkauloja tietokannassa. Viimeisenä tietokoneen käyttöjärjestelmän tasolla

voidaan nostaa tietokannan prioriteettia, jolloin tietokanta saa lisää resursseja käyttöönsä. [10. s. 144–145.]

4.3 Tietokantapalvelimen valinta

Tietokantapalvelimille on monia vaihtoehtoja. Valitsen niistä vertailuun Oracle Database-, MySQL-, Microsoft SQL Server-, PostgreSQL- ja MongoDB-tietokantapalvelimet, sillä ne ovat DB-Enginesin mukaan suosituimmat tietokantapalvelimet maailmassa. [13.]

Oraclen tietokanta Oracle Database on maksullinen tietokantapalvelin, josta on olemassa ilmainen kokeiluversio (taulukko 3). Se voidaan asentaa Linux-, Windows- ja macOS-käyttöjärjestelmien päälle. Oracle Database on relaatiotietokanta, joten se tukee SQL-kyselyitä tietokantaan. Se myös tukee tietokannan osiointia horisontaalisti ja pirstaloiden, jolloin tietokannan sisältöä voidaan säilyttää eri tietokannassa eri palvelimella. [14.]

MySQL on ilmainen tietokantapalvelin, josta on myös maksullinen versio yritysten käyttöön. Se voidaan asentaa Oraclen tietokannan tavoin Linux-, Windows- ja macOS-käyttöjärjestelmille, mutta lisäksi MySQL tukee myös Solaris-käyttöjärjestelmää. MySQL-tietokannan pää tietokantatyyppi on relaatiotietokanta, joten se tukee SQL-kyselyitä tietokantaa. Myös MySQL tukee tietokannan osiointia horisontaalisti sekä pirstalointia. [15.]

Microsoft SQL Server on Microsoftin tietokantapalvelin, josta on mahdollista saada ilmainen kokeiluversio käyttöön. Se voidaan asentaa joko Linux- tai Windows-käyttöjärjestelmälle. Sen pää tietokantatyyppi on relaatiotietokanta, jolloin se tukee SQL-kyselyiden tekemistä tietokantaa. Myös Microsoft SQL Server tukee tietokannan pirstalointia ja horisontaalista osiointia. [16.]

PostgreSQL on ilmainen tietokantapalvelin, jonka voi asentaa Linux-, macOS-, Windows-, BSD- ja Solaris-käyttöjärjestelmille. Sen pää tietokantatyyppi on relaatiotietokanta, joten PostgreSQL tukee SQL-kyselyiden tekemistä tietokantaa. PostgreSQL tukee tietokantojen osiointia horisontaalisti ja vertikaalisti, mutta se ei tue pirstalointia, toisin kuin aiemmat tietokantapalvelimet. [17.]

MongoDB on ilmainen tietokantapalvelin, jonka voi asentaa Linux-, macOS- ja Windows-käyttöjärjestelmille. MongoDB:stä on myös valmis konttiratkaisu Docker-ympäristöön. MongoDB:en pää tietokantatyyppi on dokumenttitietokanta, joten se ei täysin tue SQL-kyselyiden tekemistä tietokantaan, vaan tietyissä tilanteissa tietokannasta voidaan lukea tietoa SQL-kyselyllä. MongoDB tukee tietokantojen pirstalointia. [18.]

Taulukko 3. Lista tietokantapalvelimista ja niiden ominaisuuksista. [14][15][16][17][18]

Nimi	Hinnoittelu	Käyttöjärjestelmät	Päätietokanta tyyppi	SQL	Osiointi
Oracle Database	Ilmainen versio olemassa	Linux, Windows, macOS	Relatiionaalinen	Kyllä	Pirstalointi, horisontaalinen osiointi
MySQL	Ilmainen	Linux, macOS, Windows, Solaris	Relatiionaalinen	Kyllä	Pirstalointi. horisontaalinen osiointi
MS SQL Server	Ilmainen versio olemassa	Windows, Linux	Relatiionaalinen	Kyllä	Pirstalointi, horisontaalinen osiointi
PostgreSQL	Ilmainen	Linux, macOS, Windows, BSD, Solaris	Relatiionaalinen	Kyllä	Horisontaalinen ja vertikaalinen osiointi
MongoDB	Ilmainen	Linux, macOS, Windows, Docker	Dokumentti	Vain luku	Pirstalointi

Näistä tietokantapalvelimista valitsen projektiin MongoDB-tietokantapalvelimen, sillä se on ainoa tietokantapalvelin, jota varten Unreal Enginen kauppasivulta löytyy ilmaiseksi ladattava tietokantalisäosa.

5 Tiedon visualisointi

Tiedon visualisointi on tiedon esittämistä grafiikan, kuten kaavioiden, diagrammien ja animaatioiden, avulla. Visualisoinnin tarkoitus on näyttää monimutkaista dataa helposti ymmärrettävässä muodossa. [19.]

Tiedon visualisointia voidaan käyttää hyväksi monella eri tavalla eikä visualisointia ole tarkoitettu vain tietoa tutkivien tiimien käyttöön. Esimerkiksi yritysten johdossa olevat ihmiset voivat käyttää visualisointia esittämään yrityksen rakennetta ja datatieteilijät voivat löytää ja selittää kuvioita visualisoinnin avulla. [19.]

Tiedon visualisoinnissa on hyviä ja huonoja puolia. Visualisoinnin hyvä puoli on se, että näytetystä tiedosta on helppo löytää toistuvia kuvioita ja tavallisesta poikkeavia asioita, koska ihmisten silmät ovat niin hyviä erottamaan värejä ja kuvioita toisistaan. Lisäksi ihmiset sisäistävät tiedon nopeasti visualisoinnin kautta ja visualisointi helpottaa tiedon jakamista eteenpäin. Huonona puoleena visualisoinnissa on se, että jos visualisointi on huonosti suunniteltu tai toteutettu, tiedosta voi ymmärtää vääriä asioita tai tieto voi olla puolueellista. Lisäksi tiedon viesti saattaa hävitä kokonaan huonosti toteutetussa visualisoinnissa. [20.]

Visualisointi on tärkeää, sillä se helpottaa tiedon näkemistä, tutkimista ja ymmärtämistä. Niin kauan kuin visualisointi on tehty oikein, ei sen monimutkaisuudella ole väliä. Hyvä visualisointi tuo tiedon jokaisen ymmärrettäväksi. Visualisointia myös tarvitaan jokaisella alalla siksi, että se tekee tiedosta helposti ymmärrettävää. Mitä paremmin osaat näyttää haluamaasi tietoa hyvin visualisoinnin avulla, sitä helpommin tietoa voi käyttää hyväkseen. [20.]

5.1 Visualisointitavat

Tiedon visualisointiin on monia tapoja. Kun puhutaan visualisoinnista, yleensä ihmisille tulee ensimmäisenä mieleen yksinkertaiset pylväs- ja ympyrädiagrammit. Tämä johtuu siitä, että ne ovat olennainen ja yleinen osa tiedon visualisointia. Ne eivät kuitenkaan sovi kaikille tietotyypeille, vaan jokaiselle tietotyypille täytyy osata valita oikea visualisointitapa. Tiedon tehokkaaseen esittämiseen on siksi useita eri vaihtoehtoja. Yleisiä visualisointitapoja ovat kaaviot, taulukot ja kartat. Näiden lisäksi yleisiä tapoja ovat infograafit ja kojelaudat, joihin kerätään useita eri visualisointitapoja yhteen. [20.]

Kaaviot esittävät tietoa käyrä-, diagrammi-, kartta- tai taulukkomuodossa. Yleisimpiä kaavio tyyppiä ovat pistekaavio, pylväsdiagrammi, lineaarinen graafi ja ympyrädiagrammi. Nämä kaavio tyyppit, yksin tai yhdessä, vastaavat suureen osaan relationaalisen tiedon omaavista kysymyksistä. Kaaviot käyttävät visualisointiin karteesta koordinaatistoa, jossa koordinaatit edustavat mittasuhteita ja arvoja. Mittasuhteita käytetään koordinaatistossa ryhmittämään arvot siten, että niitä voidaan analysoida, minkä jälkeen arvoja voidaan käyttää visualisointiin. Visualisointi tyyppit eriyvät toisistaan siinä, kuinka monta mittasuhdetta ne voivat onnistuneesti näyttää. Esimerkiksi pylväsdiagrammit ovat hyviä usean eri mittasuhteen näyttämiseen, kun taas ympyrädiagrammeissa visualisoinnin selkeys häviää muutaman mittasuhteen jälkeen. Jokaisella kaaviotyypillä on omat vahvuutensa ja heikkoutensa. Oikein käytettynä jokainen tyyppi auttaa ihmisiä ymmärtämään tietoa paremmin. Kun samalla kiinnittää huomiota myös visualisoinnin ulkonäköön, visualisoinnin tyyppi ja ulkonäkö vaikuttavat siihen, kuinka katsoja näkee tietoa. [21.]

Kaavioiden päätyyppejä on useita. Ympyrädiagrammit ovat yleinen tapa vertailla asioita keskenään toisiinsa sekä kokonaisuuteen. Pylväsdiagrammeissa asiat jaetaan suuruuksien mukaan ja ne voidaan järjestellä mittasuhteiden tarpeen mukaan. Viivadiagrammeilla on helppo näyttää suhdanteita jollakin tietyllä aikavälillä ja niillä voi saman aikaisesti vertailla useampaa eri luokkaa samassa aikavälissä käyttäen useampaa viivaa. Pistekaaviolla voi tarkastella kahden arvon riippuvuutta toisiinsa eri toisistaan riippumattomilla akseleilla ja pistekaavioon voi myös lisätä suhdanneviivan. Histogrammia käyttäessä jaetaan yksi pitkän aikavälin mittaus erilaisiin ryhmiin, jotta nähdään, miten arvot jakautuvat mittauksessa. Pallokaaviossa pallojen kokoja käytetään esittämään eri arvoja vertailun vuoksi. [21.]

Taulut ovat yksi käytetyimmistä tavoista tiedon näyttämiseen. Niissä keskitytään yksittäisten arvojen näyttämiseen eikä niin paljon visuaaliseen tyyliin. Samalla ne ovat yksi tärkeimmistä tavoista, joilla ihmiset analysoivat tietoa. Vaikka tauluissa ei keskitytäkään visuaalisesti näyttävään muotoon, voidaan tauluun silti lisätä visuaalisia elementtejä, jotta tiedon tutkimisesta tulee helpompaa. Yleisesti ihmiset käyttävät tauluja, kun ne ovat sisällytettynä verkkosivuilla, ne ovat ravintolan valikoiman muodossa tai kun töissä täytyy käyttää Microsoft Excel -sovellusta. Tämä tarkoittaa sitä, että tauluja on joka puolella elämässä, joten niiden lukeminen ja ymmärtäminen on tärkeä taito. [22.]

Monien kaavioiden tavoin taulut järjestävät tietoa akseleilla. Näissä taulussa rivejä kuvaa x-akseli ja sarakkeita kuvaa y-akseli. Tauluja tehdessä on yleisesti tapana sijoittaa mitattavat ryhmät x-akselille ja niitä kuvaavat arvot y-akselille. Taulut ovat hyviä kuvaamaan määrällistä tietoa järjestellyssä muodossa ja niistä on helppo nähdä, kuinka eri tiedot ovat riippuvaisia toisistaan. Tiedon

analysoijat käyttävät yleensä tauluja, kun he haluavat nähdä yksittäisiä arvoja. Taulut mahdollistavat sen, että tiedosta on helppo nähdä haluttavat arvot, kuten vaikka jokin tieto tietyllä aikavälillä tai mittasuhteella. Tauluihin voi myös lisätä yhteenvetoja, jolloin suuresta tiedon määrästä voi nähdä välisummia tai loppusummia aikaväleista tai mittasuhteista. Taulujen huono puoli on kuitenkin se, että ne eivät skaalaudu hyvin. Kun rivejä ja sarakkeita on taulussa suuri määrä, se muuttuu vaikeasti luettavaksi eikä tietoa ole helppo nähdä taulusta. Visuaalisten osien lisääminen tauluun auttaa tiedon tutkijaa näkemään hyödyllisiä asioita taulusta nopeammin. Eri värien ja kokojen käyttäminen helpottaa kuvioiden ja poikkeuksien huomaamista. Taulut ovat erityisen hyviä silloin, kun halutaan tehdä vertailevaa analysointia tietojen välillä. Taulujen päätyyppejä on pari kappaletta. Painotettuja tauluja ja lämpökarttatauluja käytetään silloin, kun tiedon värillä ja koolla halutaan vertailla mittauksia. Tekstitauluja käytetään silloin, kun halutaan kuvailla tai koota mittauksista tietoa. [22.]

Karttavisualisoinneissa keskitytään tiedon ja fyysisen sijainnin suhteeseen ymmärryksen luomisessa. Karttojen koot voivat vaihdella todella pienistä, esimerkiksi yksi tie, kaupunki tai puisto, todella suuriin, kuten kokonainen maa, maanosa tai jopa planeetta. Kartat toimivat säiliönä kaikelle lisätiedolle. Tämä mahdollistaa sen, että erilaisia muotoja ja värejä käyttäen tietoa voidaan näyttää halutulla tavalla. Niiden avulla voidaan sitten huomata ongelmia, seurata ja ymmärtää muutoksia ja suhdanteita sekä ennustaa, mitä jollakin alueella voisi tapahtua ja milloin. Karttavisualisoinnit korostavat tiedon fyysisiä yhteyksiä toisiinsa. Tämä voi kuitenkin aiheuttaa muutamman virheen, joita karttavisualisointeja tehdessä yleensä tapahtuu. Kun karttaa skaalaa, se voi muuttaa sitä, kuinka katsojat ymmärtävät kartalla näkyvän tiedon. Lisäksi toisiaan lähellä olevia tietopisteet voidaan nähdä toisiinsa sidonnaisina, vaikka ne eivät ole millään tavalla toisiinsa liitännäisiä. Karttavisualisoinnin voivat näyttää kaikenlaisia tietoja. Ne voivat kertoa esimerkiksi ihmisten historiaa tai näyttää nopeimman reitin kahden kaupungin välillä. Karttavisualisoinnissa on kaksi päätyyppiä. Ensimmäinen niistä on suhteellisten symbolien kartta, jossa symbolien sijainnit merkitsevät jotakin kiinnostavaa tietoa. Lisäksi symbolien koko ja väri muuttuvat tiedon mukana. Toisena päätyyppinä toimivat aluekartat, joissa tieto rajautuu maantieteellisten rajojen mukaan eri värein. Värit voivat kartalla ilmaista jonkin tietyn alueen sisäistä tietoa tai tiedon tiheyttä jollakin alueella. [23.]

5.2 Visualisointisovelluksen valinta

Visualisointia varten on olemassa monia eri sovelluksia. Tähän projektiin valitsen kuitenkin sellaisen sovelluksen, joka tukee MongoDB-tietokannasta visualisointia. Valitsen vertailtavaksi viisi eri sovellusta, Tableau Desktop, Power BI, MongoDB Charts, Qlik Sense ja MongoDB Compass. Nämä sovellukset ovat Hevo Data Inc:in mukaan suosittuja vaihtoehtoja tiedonvisualisoinnille, jossa käytetään MongoDB-tietokantaa [24].

Ensimmäinen vaihtoehto on Tableau Desktop, joka on maksullinen sovellus. Siitä on kuitenkin mahdollista saada ilmainen kokeiluversio, jos sen ominaisuuksia haluaa kokeilla (taulukko 4). Tableau Desktopin saa asennettua Windows- ja macOS-käyttöjärjestelmille. [25.]

Toisena vaihtoehtona on Microsoftin Power BI. Sekin on maksullinen sovellus ja siitä on mahdollista saada ilmainen kokeilu, jos kuuluu jonkin yrityksen työntekijöihin, sillä se vaatii kirjautuessa yrityssähköpostia. Power BI on mahdollista asentaa vain Windows-käyttöjärjestelmälle. [26.]

Kolmas vaihtoehto on MongoDB:n Atlas Charts. Se sisältyy MongoDB Atlakseen. [27.] MongoDB Atlas puolestaan on maksullinen sovellus, josta on mahdollista saada ilmainen kokeiluversio. Se voidaan asentaa Windows-, macOS- ja Linux-käyttöjärjestelmille. [28.]

Neljäs vaihtoehto on Qlik Sense, joka on myös maksullinen visualisointisovellus, josta on mahdollista saada ilmainen kokeiluversio, kun Qlikiin ottaa yhteyttä omilla tiedoillaan. Qlik Sense voi asentaa vain Windows-käyttöjärjestelmälle. [29.]

Viimeisenä vaihtoehtona on MongoDB Compass, joka on täysin ilmainen sovellus ja sen voi asentaa Windows-, macOS- ja Linux-käyttöjärjestelmille. [30.]

Valitsen näistä sovelluksista käyttööni MongoDB Compass sovelluksen, sillä se on täysin ilmainen sovellus.

Taulukko 4. Visualisointi sovellus vaihtoehdot [25][26][27][28][29][30]

Nimi	Hinnoittelu	Käyttöjärjestelmä
Tableau Desktop	Maksullinen, ilmainen kokeilu.	Windows, macOS
Microsoft Power BI	Maksullinen, yritys käyttäjät voivat kokeilla ilmaiseksi	Windows
MongoDB Atlas Charts	Sisältyy Atlas, joka on maksullinen, ilmainen kokeilu	Windows, macOS, Linux
Qlik Sense	Maksullinen, ilmainen kokeilu	Windows
MongoDB Compass	Ilmainen	Windows, macOS, Linux

6 Työnteko

Projektin työ on helppo jakaa neljään osaa: VPN-sovellusten asennus ja yhteyden muodostaminen, tietokantasovelluksen asennus ja tietokannan suunnittelu, tiedon tallennus pelistä tietokantaan ja visualisointi sovelluksen asentaminen ja tiedon visualisointi.

Projektissa tietokanta ja VPN-palvelin sijaitsevat DC-Labran konesalissa ja peli- ja visualisointikone sijaitsee DC-Labran luokahuoneessa. Tarkoitus on luoda VPN-yhteys DC-Labran luokahuoneen verkosta DC-Labran konesalin verkkoon ja kuljettaa yhteys kahta julkista IP-osoitetta käyttäen julkisen verkon läpi salatussa VPN-tunnelissa. Lisäksi luodaan toinen tunneli konesalissa, jolloin liikenne kulkee salattuna myös VPN- ja tietokantapalvelimien välillä.

6.1 VPN-yhteyden luominen

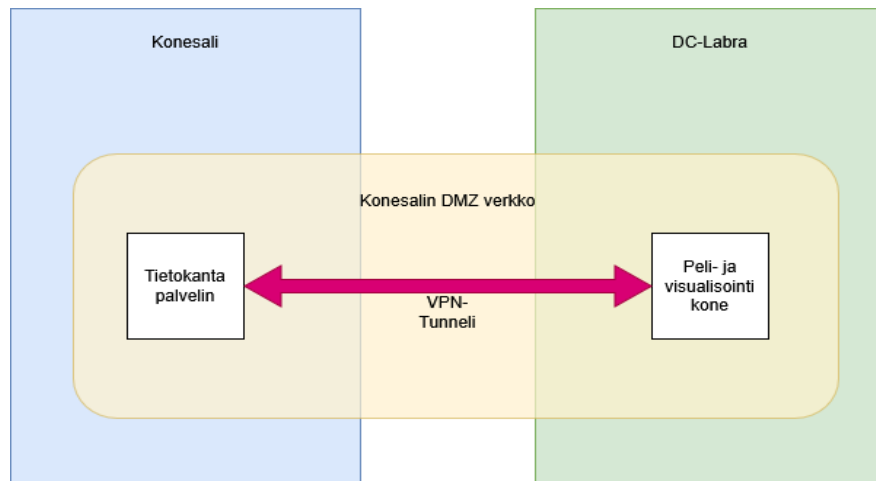
Tässä asennusosiossa asennetaan WireGuard-sovellus fyysiselle tietokoneelle, johon asennetaan myöhemmin visualisointisovellus ja tietolähde, sekä virtuaalikoneelle, johon asennetaan myöhemmin MongoDB-tietokantapalvelin.

Olen varannut IP-osoitteet molemmille koneille sekä WireGuard-sovelluksen asennuksessa tehtäville virtuaalisille verkkokorteille. IP-osoitteita tulee käyttöön yhteensä 4 kappaletta. Lisäksi olen varannut 2 kappaletta julkisia IP-osoitteita, joihin liikenne ohjataan palomuurilla.

Heti ennen asennuksien aloittamista vastaan tuli este, kun KamIT ei tietoturvasyistä halunnut avata portteja koulun luokasta ulkoverkkoon, sillä tietokone, jolle peli, MongoDB Compass ja WireGuard olisi asennettu, on koulun oppilaiden ja opettajien muussa käytössä samanaikaisesti. Näin päädyttiin ratkaisuun, jossa KamIT asensi projektia varten tietokoneen, joka sitten asetettiin DC-Labran takaosaan.

Lisäksi DC-Labrassa ei haluttu avata porttia DC-Labran luokan verkosta konesalin DMZ-verkkoon, joten KamIT:ilta saatu kone asetettiin suoraan DMZ-verkkoon (kuva 1), jolloin se on samassa verkossa kuin tietokantapalvelin. Samalla myös luovuttiin julkisten IP-osoitteiden käytöstä, mutta pidän projektissa mukana WireGuard-sovelluksen, jolloin projektin alkuperäinen idea toteutuu siltä osin.

Alkuperäiseen asennussuunnitelmaan kuului kaksi virtuaalikonetta: tietokantakone ja VPN-palvelinkone. Huomasin kuitenkin asennuksia tehdessäni, että VPN-palvelinkone on projektissa turha välikäsi, joten luovuin siitä asennuksien yhteydessä.



Kuva 1. Projektin lopullinen topologia

6.1.1 Tietokantakoneen ja VPN sovelluksen asennus

Tietokantakonetta varten asensin virtualisointiympäristöön virtuaalikoneen Ubuntu Server 24.04.2 -käyttöjärjestelmän käyttäen Ubuntu Server 24.04.2 Live -asennustiedostoa.

Käyttöjärjestelmän asennuksen jälkeen suoritetaan Ubuntu komentokehoteessa kaksi päivityskomentoa (koodiesimerkki 1), joilla päivitetään paketin hallinta ja paketit, jonka jälkeen WireGuard voidaan asentaa koneelle noudattaen WireGuardin verkkosivuilta löytyviä ohjeita (koodiesimerkki 2).

```
sudo apt-get update
sudo apt-get upgrade
```

Koodiesimerkki 1. Päivityskomennot

```
sudo apt-get install wireguard
```

Koodiesimerkki 2. WireGuard asennuskomento [31]

Seuraavaksi konfiguroidaan WireGuardille virtuaalinen verkkoliitäntä ja annetaan sille IP-osoite. (koodiesimerkki 3).

```
sudo ip link add dev wg0 type wireguard
sudo ip address add dev wg0 10.10.24.42/16
```

Koodiesimerkki 3. WireGuard-verkkokomentoja [32]

Seuraavaksi luodaan salausavainpari, jota käytetään VPN-tunnelin salauksen luomiseen ja purkamiseen (koodiesimerkki 4). [32.]

```
wg genkey > private
wg pubkey < private
```

Koodiesimerkki 4. Salausavainparin luominen [32]

Kun WireGuardia varten on luotu virtuaalinen verkkoliitäntä, voidaan seuraavaksi luoda konfiguraatitiedosto, johon kirjoitetaan kaikki tarvittavat WireGuard-konfiguraatiot WireGuardin verkkosivuilta löytyvien esimerkkien mukaan. [32.]

```
sudo nano wgclient.conf
[Interface]
PrivateKey = (Aiemmin luotu yksityinen avain)
ListenPort = 51823

[Peer]
PublicKey = (Julkinen avain Windows koneelta)
AllowedIPs = 0.0.0.0/0

Endpoint = 10.10.24.10:51821
```

Koodiesimerkki 5. WireGuardin konfiguraatitiedoston luominen. [32]

Tallennetaan konfiguraatitiedosto ja sitten asetetaan se WireGuardia varten tehdyille virtuaali-verkkoportille käyttöön ja laitetaan verkkokortti päälle (koodiesimerkki 6).

```
sudo wg setconf wg0 wgclient.conf
```

```
sudo ip link set up dev wg0
```

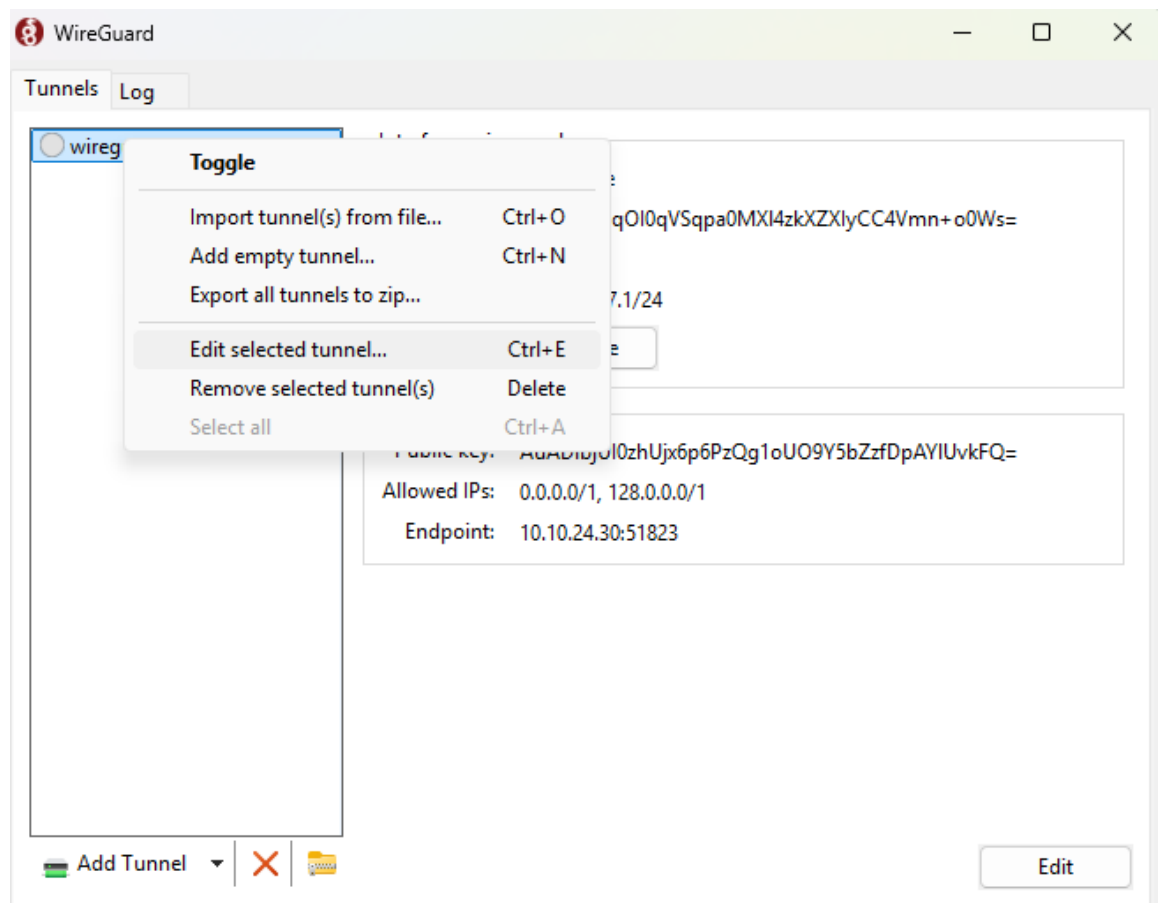
Koodiesimerkki 6. Konfiguraation asetus ja verkkokortin käynnistys. [32]

Kun molemmat VPN asennukset on suoritettu, yhteyden pitäisi toimia.

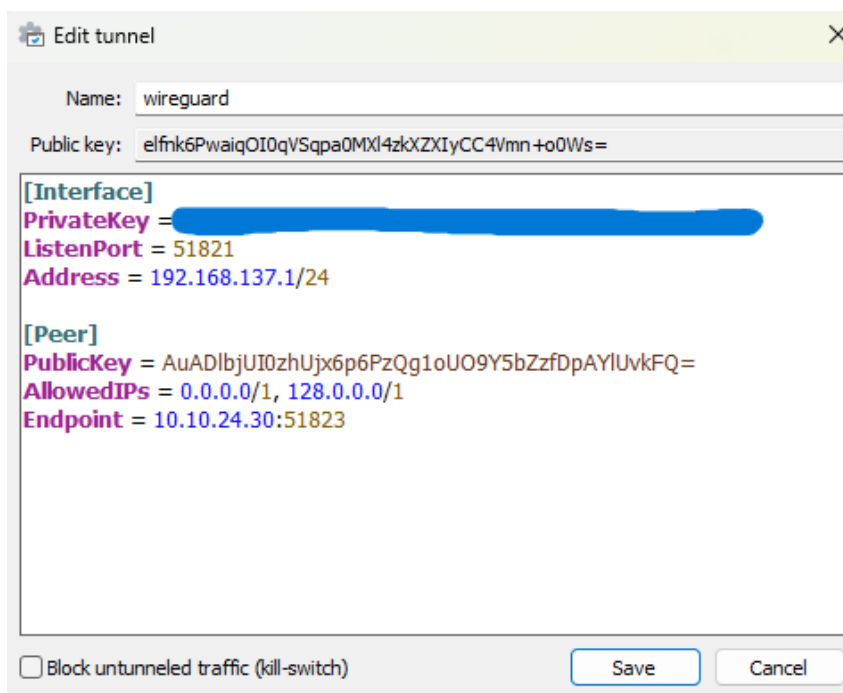
6.1.2 Peli- ja Visualisointikoneen VPN-asennus

Tietokone, jolle peli ja visualisointisovellus asennetaan, on Windows 11-käyttöjärjestelmän omaava kone. WireGuardilla on Windows-käyttöjärjestelmälle asennuspaketti, jonka avulla WireGuard saadaan asennettua tietokoneelle [31].

Windows-koneella tyhjän tunnelin luomisen yhteydessä WireGuard luo automaattisesti julkisen ja yksityisen avaimen, jota käytetään VPN-tunnelin salauksessa. Tyhjän tunnelin luomisen jälkeen sitä täytyy muokata (kuva 2) ja IP-osoite täytyy lisätä itse suoraan konfiguraatioon, jotta WireGuard toimii oikein [27]. Tämän jälkeen loput konfiguraatiot (kuva 3) voidaan määrittellä WireGuardin sivuilta löytyvien esimerkkien avulla oikean näköisiksi [32].

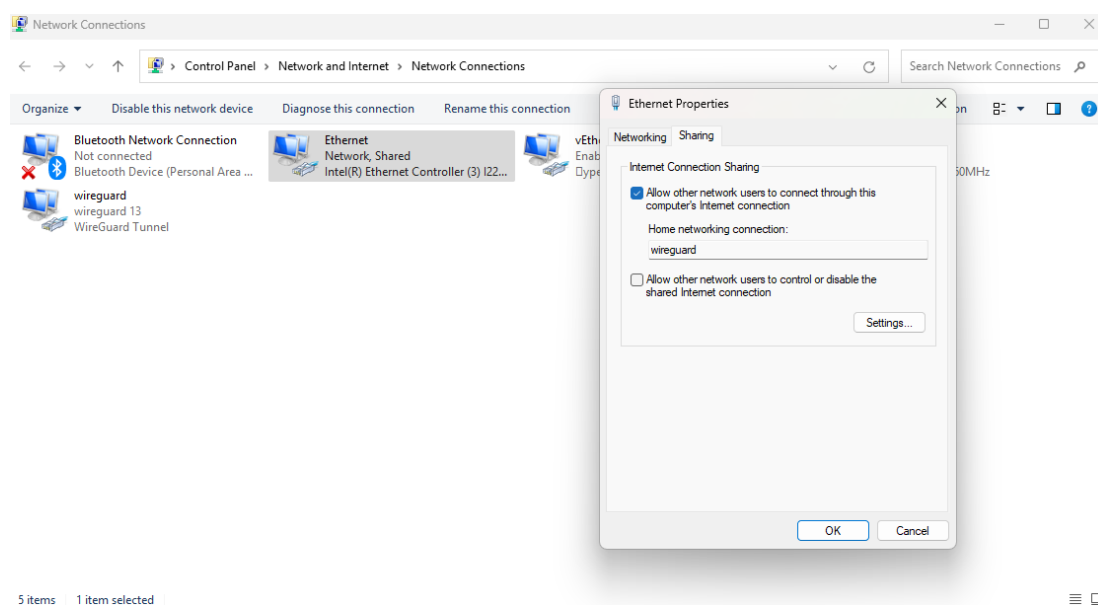


Kuva 2. Tyhjän tunnelin muokkaus.



Kuva 3. Tunnelin konfiguraatio.

Jotta WireGuardin virtuaalinen verkkokortti saa yhteyden Windows-koneen fyysisen verkkokortin kautta, täytyy fyysinen verkkokortti ensin jakaa WireGuardin verkkokortin käyttöön. Tämä voidaan tehdä Windowsin ohjauspaneelin verkkoasetuksista muokkaamalla verkkosovittimen asetuksia (kuva 4). Kun fyysisen verkkokortin jakaa virtuaalisen verkkokortin käyttöön, Windows automaattisesti muuttaa virtuaalisen verkkokortin IP-osoitteen. Tämä IP-osoite sitten lisätään Windows-koneella WireGuardin IP-osoitteeksi. [33.]



Kuva 4. Verkkosovittimen asetukset.

Näiden konfiguraatioiden jälkeen yhteys ei kuitenkaan vielä lähtenyt toimimaan. Vasta sitten, kun Windows-tietokoneen verkkokortista käytiin ottamassa pois käytöstä Forwarding-ominaisuus Windows PowerShellin avulla [34], WireGuard sai luotua tunnelin tietokoneiden välille (kuva 5) ja tietokanta koneeseen sai testattua yhteyden ping-komentoa käyttäen (kuva 6).

```
Peer
  Public key: AuADlUbjUI0zhUjx6p6PzQg1oUO9Y5bZzfDpAYIUvkFQ=
  Allowed IPs: 0.0.0.0/1, 128.0.0.0/1
  Endpoint: 10.10.24.30:51823
  Latest handshake: 1 minute, 30 seconds ago
  Transfer: 348 B received, 273.39 KiB sent
```

Kuva 5. VPN-tunnelin neuvottelu toimii.

```
Pinging 10.10.24.42 with 32 bytes of data:
Reply from 10.10.24.42: bytes=32 time<1ms TTL=64
Reply from 10.10.24.42: bytes=32 time<1ms TTL=64
Reply from 10.10.24.42: bytes=32 time<1ms TTL=64
Reply from 10.10.24.42: bytes=32 time<1ms TTL=64

Ping statistics for 10.10.24.42:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Kuva 6. Koneiden välisen yhteyden testaus Ping-komennolla.

Kun koneiden välinen yhteys on saatu toimimaan, voidaan siirtyä tietokannan asennukseen.

6.2 Tietokantapalvelimen asennus

MongoDB:stä asennan uusimman Community -version aiemmin asennetun Ubuntu Linux -käyttöjärjestelmän päälle. MongoDB:n lataus vaatii Ubuntuun asetettavaksi MongoDB:n julkisen GPG-avaimen. Tämän avaimen lisääminen Ubuntuun vaatii curl- ja gnupg-paketit asennetuiksi. Curl ja gnupg asennetaan molemmat käyttäen Ubuntu mukana tulevaa apt-paketinhallintasovellusta (koodiesimerkki 5). Samalla asetetaan GPG-avain, jotta saamme ladattua MongoDB-sovelluksen. [34.]

```
sudo apt-get install gnupg curl

curl -fsSL https://www.mongodb.org/static/pgp/server-8.0.asc | \
sudo gpg -o /usr/share/keyrings/mongodb-server-8.0.gpg \
--dearmor
```

Koodiesimerkki 5. Gnupg- ja Curl-asennus ja MongoDB GPG -avaimen asetus. [34]

Seuraavaksi lisätään MongoDB latauslinkki listaan, josta apt löytää sen ja päivitetään apt-lista ja asennetaan MongoDB Community (koodiesimerkki 6). [34.]

```
echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-8.0.gpg ]
https://repo.mongodb.org/apt/ubuntu noble/mongodb-org/8.0 multiverse" | sudo tee
/etc/apt/sources.list.d/mongodb-org-8.0.list

sudo apt-get update

sudo apt-get install -y mongodb.org
```

Koodiesimerkki 6. Latauslinkin lisäys, apt-listan päivitys ja MongoDB-asennus. [34]

Tämän jälkeen käydään muuttamassa MongoDB-konfiguraatitiedostoa, jonka sijainti on /etc/mongod.conf ja vaihdetaan tiedostosta kohdat port ja bindip niin, että ne ohjaavat oikeaan IP-osoitteeseen ja porttiin, jotka tässä tapauksessa ovat WireGuardin osoite ja jokin portti, johon MongoDB:hen menevä liikenne ohjataan. Tämän jälkeen tarkistetaan, mitä komentoa käyttöjärjestelmässä käytetään sovellusten käynnistämiseen (koodiesimerkki 7). [34.]

```
ps - -no-headers -o comm 1
```

Koodiesimerkki 7. Käynnistyssovelluksen tarkastus. [34]

Tässä tapauksessa komento palautti vastauksen systemd, joten käytämme sitä MongoDB-tietokannan käynnistämiseen (koodiesimerkki 8). Tämän jälkeen tarkistamme, että se varmasti käynnistyi ja varmistamme, että MongoDB käynnistyy, kun kone käynnistyy uudelleen. [34.]

```
sudo systemctl start mongod  
  
sudo systemctl status mongod  
  
sudo systemctl enable mongod
```

Koodiesimerkki 8. MongoDB käynnistys, tilan tarkistus ja käynnistykseen varmistut komennot. [28]

Tämän jälkeen voidaan yhdistää omaan tietokantaa suorittamalla komento, joka avaa komentorivin tietokannan sisällä (koodiesimerkki 9). [34.]

```
mongosh [määritetty IP-osoite]:[määritetty portti]
```

Koodiesimerkki 9. Tietokantaan yhdistäminen [34][35]

Näin tietokannan asennus on valmis ja voidaan siirtyä tietokannan suunnitteluun.

6.3 Tiedon tallennus tietokantaan

Minusta riippumattomista tekijöistä en saanut oikeuksia pelin lataukseen ajoissa, joten tietolähteenä käytän mongodbrdg-sovellusta, jonka käyttäjä jdrumgoole on luonut github-verkkosivulle. Mongodbrdg on sovellus, joka on suunniteltu luomaan satunnaisia henkilötietoa MongoDD-tietokantaan. Sovelluksen luoma tieto on satunnaista, mutta ristiriidatonta, joten käyttäjä voi sen avulla luoda satunnaista tietoa, joka on samassa muodossa jokaisella luontikerralla. Sovellus luo tiedon JSON-muodossa ja sillä voi luoda useampia teennäisiä henkilöitä kerralla. Sovellus asennetaan Python 3.13 -tulkkiin ja Python-ohjelmointikieltä hyväksikäyttäen komentokehoteesta. Sovellus generoi satunnaista tietoa valmiiksi määritellyillä tunnisteilla. [38.]

Suuri osa sovelluksen generoimasta tiedosta on hyödytöntä tämän projektin kannalta, mutta tiedon seasta voidaan poimia nimi, koordinaatit ja rekisteröinnin aika (kuva 7), jotka ovat tietotyypiltään samanlaista tietoa, kuin mitä pelistä alun perin haluttiin.

```
{
  "first_name": "Donnetta",
  "last_name": "Page",
  "gender": "FEMALE",
  "company": "Syntel",
  "email": "Donnetta.Page@syntel.rio",
  "registered": "2010-06-09 11:06:05.882643",
  "user_id": 0,
  "country": "United States",
  "city": "Hayward",
  "phone": "1-171-738-1641",
  "location": {
    "type": "Point",
    "coordinates": [
      164.393576,
      59.535072
    ]
  },
  "language": "Dari",
  "interests": [
    "Reading",
    "politics"
  ]
}
```

Kuva 7. Sovelluksen generoimaa tietoa. [38]

Muokkaamalla sovelluksen konfiguraatitiedostoa voidaan poistaa kaikki ylimääräinen tieto, jota sovellus haluaa generoida, jolloin jäljelle jäävät vain nimi ja koordinaatit kuvaamaan jokaista pelissä olevaan hahmoa sekä rekisteröinnin aika, jota voidaan käyttää kuvaamaan pelissä tapahtuvaa yhtä hetkeä.

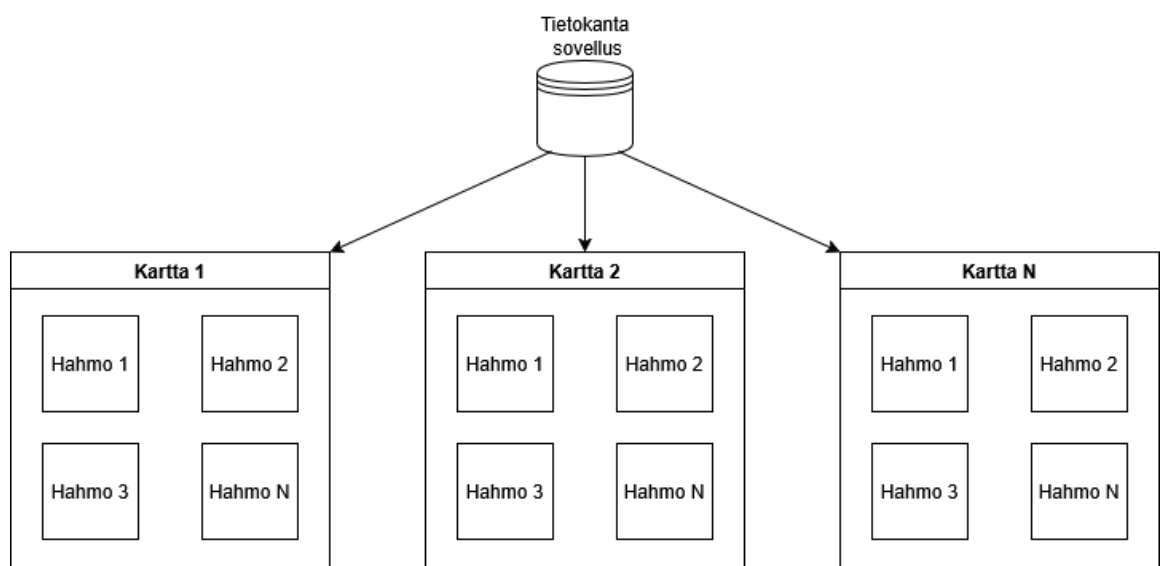
Tämän jälkeen vain suoritetaan sovellus Windows-koneen komentokehotteesta muutaman lisäargumentin kanssa (koodiesimerkki 11), jolloin sovellus luo tietoa oikeaan tietokantaan ja kokoelmaan.

```
mongodbrdg.exe - -mongodb mongodb://(IP-osoite):(portti) - -database (tietokannan nimi) - -collection (kokoelman nimi) - -idend 10
```

Koodiesimerkki 11. Sovelluksen suorituskomento ja lisäargumentit. [38]

6.4 Tietokannan suunnittelu

Tietokantaan olisi pelistä haluttu tallentaa jokaisesta pelisessioista pelaajahahmo ja jokainen vihollishahmo. Jokaiselle hahmolle olisi tallennettu koko pelisession ajalta hahmon tyyppi, hahmon tila, koordinaatit. Tietokantaan tulisi jokaiselle pelin kartalle oma tietokanta (kuva 8), johon jokainen pelisessio tulisi omana kokoelmanaan ja jokaiseen kokoelmaan tallentuisi jokainen hahmo omana dokumenttinaan, joka sisältäisi kyseisen hahmon tiedot jokaiselta pelisession sekunnilta (kuva 9).



Kuva 8. Tietokannan topologia.

Hahmo N
Aikaleima: 10.02.2025 15:00
Olotila: Elossa
Koordinaatti X: 50
Koordinaatti Y: 30
Tyyppi: Robotti
Aikaleima: 10.02.2025 15:01
Olotila: Elossa
Koordinaatti X: 45
Koordinaatti Y: 35
Tyyppi: Robotti

Kuva 9. Yksittäisen hahmon dokumentti tietokannassa.

Pelissä hahmoja voi olla monia kymmeniä, joten tietokantaan tulee paljon dataa koko pelisession ajan. Jotta tiedon etsiminen pelisession ajalta nopeutuu, tietokanta voidaan indeksoida aikaleimojen mukaan, jolloin kyselyiden tekeminen aikaleimojen mukaan nopeutuu. Indeksini voidaan luoda yhdistämällä MongoDB-sovellukseen (koodiesimerkki 9), siirtymällä muokkaamaan tietokantaa ja tietokannan sisällä suoritetaan komento, joka luo indeksin (koodiesimerkki 10) [36].

```
db.collection.createIndex( {timestamp: 1});
```

Koodiesimerkki 10. Indeksini luominen. [36]

Tietokantaa halutaan luoda asetus, joka poistaa 2 viikkoa vanhat ja sitä vanhemmat kokoelmat, jotta tietokannan koko ei pääse paisumaan todella suureksi ja vanhojen pelisessioiden tietoja ei haluta säilyttää tietokannassa ikuisuutta. Asetus voidaan luoda yksinkertaisella komennolla, jossa määritellään poistettavaksi kaikki tieto, joka on yli kaksi viikkoa vanhaa (koodiesimerkki 11).

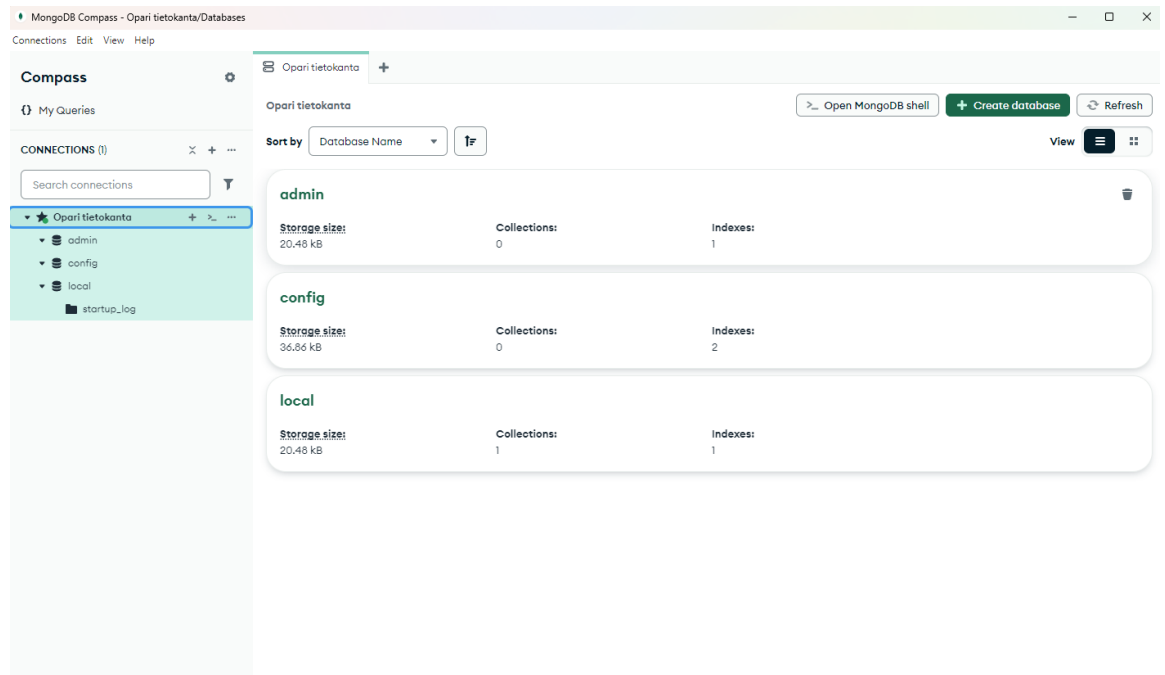
```
db.collection.deleteMany( { orderExpDate : { "$lt" : new Date(Date.now() - 14*24*60*60 * 1000) } })
```

Koodiesimerkki 11. Vanhojen tietojen poisto. [37]

6.5 Visualisointisovelluksen asennus ja tiedon visualisointi

Visualisointisovellus asennetaan samalle Windows-tietokoneelle, kuin mistä tietoa tallennetaan tietokantaan. MongoDB Compass -asennus on helppo tehdä, sillä sen saa asennustiedostona ladattua MongoDB:n sivuilta. Ladataan vain uusin versio asennustiedostosta ja suoritetaan tiedosto. [39.]

Asennuksen jälkeen voidaan avata MongoDB Compass ja luoda uusi yhteys. Tämän projektin tapauksessa yhteyden luomiseksi ei tarvitse muuttaa muita asetuksia kuin IP-osoite ja portti MongoDB-tietokannan IP-osoitteeseen ja porttiin uuden yhteyden luomisvaiheessa. Tämän jälkeen visualisointisovellus saa yhteyden tietokantaan palvelimeen ja löytää sieltä tietokantoja (kuva 10). [40.]



Kuva 10. Yhteys tietokantaan visualisointisovelluksesta.

Visualisointisovellukseen saadaan näkymään haluttua koordinaattidataa (kuva 11), mutta tietopisteitä ei saa näkymään nimiin liitännäisinä. Lisäksi visualisoinnissa ei ole koordinaatistoa, joten pelkkiä tietopisteitä katsomalla ei pysty näkemään, missä koordinaateissa jokainen tietopiste sijaitsee.



Kuva 11. Visualisointi koordinaattidatasta.

Visualisointisovelluksessa ei myöskään saa visualisoitua tietoa yhdessä muiden tietojen kanssa, vaan sovellus on ennemminkin suunnattu tietokannassa olevan tiedon tutkimiseen ja tarkastamiseen kuin varsinaiseen tiedon visualisointiin. Kaikki tietokantaan tallennetut tiedot löytyvät visualisointisovelluksen avulla, mutta ne ovat erillisinä osioinaan, eikä tietojen välisiä yhteyksiä ole mahdollista tutkia. Esimerkiksi koordinaattitietoa olisi haluttu tutkia aika- ja nimitietoihin liitännäisinä, jolloin olisi nähty, missä jokainen hahmo sijaitsi jonkin tietyn aikamäärään kohdalla.

7 Yhteenveto

Opinnäytetyön tavoitteena oli luoda konseptitodistus pelitiedon tallentamisesta tietokantaa ja tiedon visualisoimisesta VPN-yhteyttä käyttäen. Työosuus jakautui neljään osaan: VPN-sovelluksen asennus ja yhteyksien luominen, tietokantasovelluksen asennus ja tiedon tallennus tietokantaan, tietokannan suunnittelu sekä tiedon visualisointi.

VPN-sovellukseksi valitun WireGuardin asennus sujui hyvin ja tavoitteeseen, eli toimivaan VPN-yhteyteen päästiin. Asennuksen aika tuli vastaan pari ongelmaa, jotka estivät yhteyden luomista, mutta ongelmat saatiin lopulta ratkottua etsimällä vastauksia eri keskustelupalstoilta.

Tietokannan asennus sujui hyvin enkä törmännyt ongelmiin MongoDB:n asennuksen aikana. Tiedon tallennus tietokantaa sujui kohtalaisesti, mutta kaikkia tavoitteita ei saavutettu, sillä jouduin odottamaan, että saisin oikeuden ladata pelin, josta tietoa haluttiin tallentaa tietokantaa. Lopulta jouduin luopumaan odottamisesta ja etsimään erillisen sovelluksen tiedon tallentamista varten. Sovellukseksi valikoitui mongodbrdg ja sillä tietokantaan saatiin osa halutusta tiedosta.

Tietokannan suunnittelu sujui kohtalaisesti sillä tiedolla, mitä tietokantaan varasovelluksesta saatiin. Tietoa saatiin indeksoitua ja tietokannan rakenne saatiin luotua halutun näköiseksi. Suunnitteluun olisi voinut käyttää enemmän aikaa, jos sitä olisi riittänyt.

Tiedon visualisoinnin kannalta projekti sujui kohtalaisesti, sillä varasovelluksesta ei saatu tallennettua kaikkea tietoa, mitä olisi visualisoinnissa haluttu käyttää ja valitsemani visualisointisovellus MongoDB Compass ei sisältänyt tarvittavia ominaisuuksia tiedon kunnolliseen visualisointiin. Työn aikana kuitenkin saatiin aikaan pieni visualisointi koordinaattidatasta ja sovellukseen saatiin näkymään tietokannasta kaikki tietokannat.

Yhtenäisenä kokonaisuutena projekti onnistui kohtalaisen hyvin, vaikka kaikkia tuloksia ei saavutettu. Jatkokehityksenä ratkaisu voidaan asentaa oikeaan ympäristöön, jossa tietolähteenä toimii varsinainen peli ja visualisointia varten voisi mahdollisesti tutkia maksullisiakin sovelluksia. VPN-yhteydet luotiin opinnäytetyössä niin, että ne toimisivat myös kahden eri verkon välillä, kuten alkuperäisessä suunnitelmassa.

Lähteet

1. Ciccarelli P, Faulkner C. Networking Foundations: Technology Fundamentals for IT Success. Yhdysvallat: Sybex; 2004
2. What Is a Virtual Private Network (VPN)? Cisco. [Internet]. [viitattu 25.11.2024]. Saatavilla: <https://www.cisco.com/c/en/us/products/security/vpn-endpoint-security-clients/what-is-vpn.html>
3. What is a VPN? Microsoft. [Internet]. [viitattu 9.1.2025]. Saatavilla: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-vpn>
4. What is the PPTP (Point-to-Point Tunneling Protocol)? NordVPN. [Internet]. [viitattu 9.1.2025]. Saatavilla: <https://nordvpn.com/blog/what-is-pptp-protocol/>
5. OpenVPN Community. OpenVPN Technologies, Inc. [Internet]. [viitattu 11.3.2025]. Saatavilla: <https://openvpn.net/community/>
6. strongSwan. strongSwan. [Internet]. [viitattu 11.3.2025]. Saatavilla: <https://strongswan.org/>
7. SoftEther VPN Project. SoftEther Project. [Internet]. [viitattu 11.3.2025]. Saatavilla: <https://www.softether.org/>
8. WireGuard: Fast, modern, secure VPN tunnel. WireGuard. [Internet]. [viitattu 11.3.2025]. Saatavilla: <https://www.wireguard.com/>
9. Performance testing OpenVPN. OpenVPN Community. [Internet]. [viitattu 30.3.2025]. Saatavilla: <https://community.openvpn.net/openvpn/wiki/PerformanceTestingOpenVPN>
10. Hovi A, Huotari J, Lahdenmäki T. 1. painos. Tietokantojen suunnittelu ja indeksointi. Jyväskylä: Docendo Finland Oy; 2005
11. Definition: database (DB). TechTarget. [Internet]. [viitattu 6.1.2025]. Saatavilla: <https://www.techtarget.com/searchdatamanagement/definition/database>

12. What Is a Database? Oracle. [Internet]. [viitattu 26.11.2024]. Saatavilla: <https://www.oracle.com/database/what-is-database/>
13. DB-Engines Ranking. Redgate Software. [Internet]. [viitattu 10.4.2025]. Saatavilla: <https://db-engines.com/en/ranking>
14. Oracle Database 23ai. Oracle. [Internet]. [viitattu 13.4.2025]. Saatavilla: <https://docs.oracle.com/en/database/oracle/oracle-database/23/index.html>
15. Documentation. MySQL. [Internet]. [viitattu 13.4.2025]. Saatavilla: <https://dev.mysql.com/doc/>
16. SQL Server technical documentation. Microsoft. [Internet]. [viitattu 13.4.2025]. Saatavilla: <https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16>
17. Documentation. PostgreSQL. [Internet]. [viitattu 13.4.2025]. Saatavilla: <https://www.postgresql.org/docs/>
18. What is MongoDB? MongoDB. [Internet]. [viitattu 13.4.2025]. Saatavilla: <https://www.mongodb.com/docs/manual/>
19. What is data visualization? IBM. [Internet]. [viitattu 10.1.2025]. Saatavilla: <https://www.ibm.com/think/topics/data-visualization>
20. What is data visualization? Definition, examples, and learning resources. Tableau. [Internet]. [viitattu 13.3.2025]. Saatavilla: <https://www.tableau.com/visualization/what-is-data-visualization>
21. A guide to charts. Tableau. [Internet]. [viitattu 22.3.2025]. Saatavilla: <https://www.tableau.com/chart>
22. A guide to tables. Tableau. [Internet]. [viitattu 29.3.2025]. Saatavilla: <https://www.tableau.com/visualization/what-is-table>
23. A guide to geospatial visualizations. Tableau. [Internet]. [viitattu 29.3.2025]. Saatavilla: <https://www.tableau.com/visualization/what-is-geospatial-visualization>
24. The best MongoDB visualization tool for 2025. Hevo Data. [Internet]. [viitattu 29.4.2025]. Saatavilla: <https://hevodata.com/learn/mongodb-visualization/>

25. Welcome to Tableau Desktop. Tableau. [Internet]. [viitattu 5.5.2025]. Saatavilla: <https://www.tableau.com/learn/get-started/creator>
26. Power BI. Microsoft. [Internet]. [viitattu 5.5.2025]. Saatavilla: <https://www.microsoft.com/fi-fi/power-platform/products/power-bi>
27. MongoDB Atlas Charts. MongoDB. [Internet]. [viitattu 5.5.2025]. Saatavilla: <https://www.mongodb.com/docs/charts/>
28. MongoDB Atlas. MongoDB. [Internet]. [viitattu 5.5.2025]. Saatavilla: <https://www.mongodb.com/docs/atlas/>
29. Qlik Sense on Windows help. Qlik. [Internet]. [viitattu 5.5.2025]. Saatavilla: https://help.qlik.com/en-US/sense/November2024/Content/Sense_Helpsites/Home.htm
30. What is MongoDB Compass? MongoDB. [Internet]. [viitattu 5.5.2025]. Saatavilla: <https://www.mongodb.com/docs/compass/current/>
31. Installation. WireGuard. [Internet]. [viitattu 11.03.2025]. Saatavilla: <https://www.wireguard.com/install/>
32. Quick Start. WireGuard. [Internet]. [viitattu 11.3.2025]. Saatavilla: <https://www.wireguard.com/quickstart/>
33. WireGuard Windows setup. Introserv. [Internet]. [viitattu 14.4.2025]. Saatavilla: <https://introserv.com/tutorials/wireguard-windows-setup/>
34. Install MongoDB Community Edition on Ubuntu. MongoDB. [Internet]. [viitattu 3.5.2025]. Saatavilla: <https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu/>
35. Wireguard client not working on Windows. Reddit. [Internet]. [viitattu 3.5.2025]. Saatavilla: https://www.reddit.com/r/WireGuard/comments/t8cqyp/wireguard_client_not_working_on_windows_transfer/
36. What is Indexing in a Database? MongoDB. [Internet]. [viitattu 8.5.2025]. Saatavilla: <https://www.mongodb.com/resources/basics/databases/database-index>

37. How can I remove older records from a collection in MongoDB? Stackoverflow. [Internet]. [viitattu 9.5.2025]. Saatavilla: <https://stackoverflow.com/questions/46441006/how-can-i-remove-older-records-from-a-collection-in-mongodb>
38. Mongodbrdg – The MongoDB Random Data Generator. Jdrumgoole. [Internet]. [viitattu 7.5.2025]. Saatavilla: <https://github.com/jdrumgoole/mongodbrdg>
39. Connection Strings. MongoDB. [Internet]. [viitattu 3.5.2025]. Saatavilla: <https://www.mongodb.com/docs/manual/reference/connection-string/#find-your-self-hosted-deployment-s-connection-string>
40. Try MongoDB Tools Free. MongoDB. [Internet]. [viitattu 3.5.2025]. Saatavilla: <https://www.mongodb.com/try/download/compass>