

Opinnäytetyö AMK

Tieto- ja viestintäteknikka

2024

Juho Suopanki

Microsoft SQL - tuotantotietokannan kopiointi toiselle SQL-palvelimelle tietokantavarmistusta hyödyntäen

– tekninen toteutus Windows-toimialueympäristön
työkaluilla

Opinnäytetyö AMK | Tiivistelmä

Turun ammattikorkeakoulu

Tieto- ja viestintäteknikka

2024 | 37 sivua

Juho Suopanki

Microsoft SQL -tuotantotietokannan kopiointi toiselle SQL-palvelimelle tietokantavarmistusta hyödyntäen

- tekninen toteutus Windows-toimialueympäristön työkaluja hyödyntäen

Tuotantokäytössä oleviin tietokantoihin tallennetun tiedon hyödyntäminen organisaation toissijaisissa käyttötarkoituksissa on usein haastavaa jos tietoja ei saada kopioitua. Tuotantojärjestelmän käyttöä ei saa vaarantaa jatkuvalla kuormituksella ja tietoa pitäisi olla mahdollista yhdistellä muiden järjestelmien tietojen kanssa.

Työn tavoite oli automatisoida Microsoft SQL -tuotantotietokannan kopiointi toiselle SQL-tietokantapalvelimelle jatkohyödyntämistarkoitukseen. Opinnäytetyön toteutuksessa otettiin huomioon toimintaympäristön tekniset rajoitukset ja asiakasvaatimukset. Työ toteutettiin Windows-toimialueympäristössä Microsoftin tuotteilla. Opinnäytetyön tuloksena saavutettiin toimiva ja vaatimukset täyttävä ratkaisu SQL-tietokannan kopioimiseksi toiselle SQL-palvelimelle.

Työn teknisen toteutuksen hyödyntämisessä on tapauskohtaisesti arvioitava sen soveltuvuus asiakastarpeeseen. Opinnäytetyö ei ota kantaa tiedonkäsittelyä ja -hallintaa koskeviin lakeihin tai tietoturvaan.

Asiasanat:

SQL, tietokanta, varmuuskopiointi, automaatio, Windows

Bachelor's Thesis | Abstract

Turku University of Applied Sciences

Information and communication technology

2024 | 37 pages

Juho Suopanki

Copying of Microsoft SQL production database to another SQL-server using database backup

- technical implementation using Windows-domain tools

It is often challenging to utilize data stored in production databases for organizations secondary needs if the data cannot be copied. Production databases must not be jeopardized by constant load and data should be possible to merge with data from other systems.

Main objective of the thesis was automating the copying of Microsoft SQL production database to another SQL-server for further use. The implementation took into account the technical and customer imposed limitations. Work was carried out in Windows-domain environment using Microsoft products. As a result of the thesis, a functional and compliant solution was achieved for copying the SQL database to another SQL server.

In application of the thesis' solution to a specific use-case it is recommended that a thorough assessment of applicability is made. This thesis' does not take into account the laws, regulations and information security requirements concerning information management and governance.

Keywords:

SQL, database, backup, automation, Windows

Sisältö

1 Johdanto	8
2 Toimintaympäristö ja työkalut	9
2.1 Windows-toimialue	9
2.2 Windows Active Directory -toimialuepalvelu	9
2.3 SQL Server	10
2.3.1 SQL Server -käyttöoikeudet	11
2.3.2 SQL Server Agent	12
2.3.3 Varmuuskopiointi ja tietokannan palauttaminen	14
2.3.4 T-SQL	15
3 Tiedonsiirtoratkaisun vaatimukset ja rajoitukset	16
4 Käytettävän tiedonsiirtoratkaisun suunnittelu	17
5 Tuotantotietokantojen kopiointi ja siirto raportointikantaan	20
5.1 Verkkojaon tekeminen ja Active Directory -tunnuksen luvittaminen	20
5.2 Varmistusten ottaminen tuotantoympäristöstä	23
5.3 Tietokantojen ajaminen raportointipalvelimen kantaosioiksi tuotannon varmistuksista	25
5.3.1 Varmistusten siirto ja nimeäminen	26
5.3.2 Tietokantojen palautusajolla raportointipalvelimen kantaosioiden korvaaminen	27
5.3.3 Käytettyjen varmistustiedostojen poisto ja SQL Login -luvitukset	30
6 Yhteenveto	32
Lähteet	34

Kuvat

Kuva 1. SQL Server käyttöoikeushallinnan tiekartta (Lewis 2004, 3).	12
---	----

Kuva 2. SQL Server Agent tarkasteltuna SQL Server Management Studion kautta (Das 2020).	13
Kuva 3. Ympäristön ja perustarpeen lähtötilanne.	17
Kuva 4. Tietokantakopioinnin suunniteltu päätason rakenne.	19
Kuva 5. Raportointipalvelimen levyosio.	20
Kuva 6. Esimerkki AD-tunnuksen luvituksesta tuotantoympäristön kantaosion User Mapping -välilehdellä.	21
Kuva 7. Credential-tunnuksen luominen.	22
Kuva 8. Proxy-tunnuksen luominen ja käyttöoikeudet.	22
Kuva 9. AD-tunnuksen luvittaminen raportointipalvelimen kansiorakenteisiin.	23
Kuva 10. Tuotantopalvelimen varmistusajon skriptin sijainti ja nimi.	24
Kuva 11. Tuotantopalvelimen varmistusjobin asetukset.	25
Kuva 12. SQL Server Agent job -paketissa ajettavat vaiheet.	26
Kuva 13. PowerShell-skripti varmistusten siirtämiseen ja nimeämiseen.	27
Kuva 14. RESTORE FILELISTONLY -komennon ote palautusskriptistä.	27
Kuva 15. SINGLE_USER-komennon ote palautusskriptistä.	28
Kuva 16. RESTORE-komento palautusskriptissä.	28
Kuva 17. Kannan yhteyksien käsittelyn säätäminen ja omistajan vaihto skriptillä.	28
Kuva 18. Raportointikannan tiedostojen nimenvaihdot skriptillä.	29
Kuva 19. RECOVERY MODE -asetuksen muokkaaminen skriptillä.	29
Kuva 20. SHRINKFILE-komennon ajaminen skriptillä.	29
Kuva 21. CHECKDB-komennon ajaminen raportointipalvelimen kantaosiolle.	30
Kuva 22. PowerShell-komennot käytettyjen kansiorakenteiden poistamiseen.	30
Kuva 23. SQL Login -tunnuksen luvittaminen kantaosioon skriptillä.	31

Käytetyt lyhenteet ja sanasto

AD, Active Directory	Active Directory Domain Services (AD DS) on Windows-toimialueen hakemistopalvelu sekä käyttäjätietokanta, jonka kautta voidaan hallita toimialueeseen liitettyjä tunnuksia, laitteita ja resursseja. (Microsoft 2022a.)
Kollaatio	Kollaatio on kokoelma sääntöjä, joiden perusteella käytettävä merkistö asetetaan tiettyyn järjestykseen. (Microsoft 2022c.)
Microsoft SQL Server	Microsoftin kehittämä relaatiotietokannan hallintajärjestelmä.(McQuillan 2015, 1.)
PowerShell, PS	Microsoftin kehittämä moderni komentorivityökalu, jolla voidaan hallita tehokkaasti Windows-ympäristöä ja automatisoida tehtäviä. (Microsoft 2023d.)
Ryhmäkäytänne	Ryhmäkäytänneillä (Group Policy) hallitaan ja asetetaan keskitetysti sääntöjä Windows-ympäristön käyttäjille, työasemille ja palvelimille. (Microsoft 2024b.)
SQL	Structured Query Language, ISO-standardisoitu tietokantojen kyselykieli. (W3Schools, n.d.)
SQL-instanssi	Instanssi tai ilmentymä on yksi SQL Server -tuotteen asennus, joka suorittaa sqlservr.exe-ohjelmätiedostoa. (Microsoft 2022b.)
SQL-klusteri	Microsoft SQL Server Cluster on kahden tai useamman SQL-palvelimen kokoonpano, jossa palvelimilla on identtinen pääsy tietokannan tallentamiseen käytettyyn levytilaan. (Ford 2024.)
T-SQL, TSQL	Transact-SQL, Microsoft SQL Server -tuotteen pääasiassa sisäisesti käyttämä kyselykieli, joka pohjautuu SQL-kieleen laajentaen sen toiminnallisuuksia. (Clark 2021.)

Windows-toimialue	Samaan fyysiseen tai virtuaaliseen tietoverkkoon kuuluva kokoelma tietokoneita ja palvelimia, joiden käyttöä ja asetuksia voidaan hallita keskitetysti. (Hoffman 2014.)
WMI	Windows Management Instrumentation on Windows-käyttöjärjestelmissä oleva infrastruktuuri, joka tallentaa ja toimittaa hallintatietoa käyttöjärjestelmän muille osille. Sen avulla voidaan myös automatisoida hallinnallisia tehtäviä. (Microsoft 2023f.)

1 Johdanto

Organisaatioiden tietoa säilytetään useissa eri paikoissa ja useilla eri tavoilla. Data säilytettynä ei vielä luo arvoa, vaan se tulee saattaa sillostaan saataville ja käyttöön. Tietoa tuottavien järjestelmien tuotantokäyttöä vaarantamatta tiedon tuominen saataville niin, että siitä voidaan tehdä johtopäätöksiä, analyysiä, suunnitelmia, tunnistaa uhkia ja kehittämisen kohteita, on ensiarvoisen tärkeää organisaation toiminnalle. Tietoon perustuva päätöksenteko vaatii tiedon saattamista saataville. (Ceccobao 2021.) Tämän opinnäytetyön esittelemä ratkaisu on yksi tapa purkaa näitä silloja.

Opinnäytetyössä toteutetaan ja kuvataan Microsoft SQL Server -tietokannan kopiointi erilliselle SQL Server -palvelimelle, käyttäen hyväksi vain Windows-toimialueella olevia työvälineitä ja ohjelmia sekä SQL Server -tuotteen sisään rakennettuja ominaisuuksia. Toteutuksella palvellaan asiakasorganisaation tiedonkäytön toissijaisia toimintoja, kuten raportointia ja laskutusta. Tavoite on tehdä asiakkaan tarpeisiin sopiva kustannustehokas tietojen kopiointi- ja siirtoratkaisu ilman erillisiä integraatiotuotteita.

Toteutus soveltuu käyttöön mille tahansa organisaatiolle, jolla on tarve viedä SQL Server -pohjaisesta tuotantotietokannasta tiedot täysimuotoisena erilliseen SQL-ilmentymään tietyin reunaehdoin. Tiedonsiirrot eivät saa

- olla aikasensitiivisiä, kuten reaaliaikaisia tai hyvin tiheän päivityssyklin vaativia
- vaatia kumulatiivista käsittelyä tai tiedonsiirtotoimintoa
- vaatia konvertointia toiseen tietokantatuotteeseen sopivaksi.

Opinnäytetyössä käydään ensin läpi toteutuksen kannalta relevanttia teoriaa. Käytännön osiossa näytetään tietokantakopioinnin ja palauttamisen rakentamisen asetukset yksittäisen esimerkkisiirron osalta tavoitteena antaa lukijalle pohjatiedot toteutuksen luomiseksi itsenäisesti.

2 Toimintaympäristö ja työkalut

2.1 Windows-toimialue

Windows-toimialue on joukko samassa verkossa toimivia tietokoneita, joita hallitaan keskitetysti vähintään yhdeltä Domain Controller -roolissa olevalta Windows Server -palvelimelta. Tämä mahdollistaa verkon ja siihen liitettyjen laitteiden, käyttäjien ja muiden toimialueen jäsenlaitteiden keskitetyn hallinnan verkon ylläpitäjille. (Stegner, B. 2018.)

Toimialueeseen liitettyjen laitteiden käyttäjät eivät tyypillisesti käytä paikallisia kirjautumistunnuksia vaan niitä hallitaan ohjauspalvelimelta (Domain Controller, DC). Toimialueen laitteille kirjaututtaessa laite tarkistaa käyttäjän tunnuksen kirjautumisoikeudet ohjauskoneen hakemistopalvelusta (Active Directory, AD). (Stegner, B. 2018.) Näin ollen käyttäjä voi toimialueella kirjautua tunnuksellaan useisiin eri toimialueen laitteisiin. Toimialueen ylläpitäjät hallitsevat laitteiden asetuksia ryhmäkäytänteillä, joten laitteiden käyttäjillä voi olla erittäin rajoitetut oikeudet muuttaa laiteasetuksia. (Hoffman, C. 2014.)

Yksityishenkilöiden käyttämät tietokoneet ja laitteet eivät yleensä ole Windows-toimialueen jäseniä. Toimialueiden tyypillinen käyttötarkoitus on isojen laite- ja käyttäjämäärien keskitetty hallinta yritysten, koulujen ja valtion tietoverkoissa. Toimialueeseen liittyvissä tietokoneissa tulee olla Windows-käyttöjärjestelmän Professional tai Enterprise -versio asennettuna. (Hoffman, C. 2014.)

2.2 Windows Active Directory -toimialuepalvelu

Active Directory -toimialuepalvelu (Active Directory Domain Services, AD DS) on Windows Server -palvelimella toimiva hierarkkinen hakemistorakenne, joka tallentaa Windows-toimialueen verkossa olevaa tietoa ja myös antaa sen verkkoalueen käyttäjien saataville. Tiedot tallennetaan nimettyinä objekteina ja niihin liittyy attribuutteja eli ominaisuuksia. Tyypillisiä objekteja ovat toimialueen laitteet, käyttäjät, käyttäjäryhmät ja sovellukset. Active

Directory -toimialuepalvelu asennetaan Windows-palvelimelle, jolloin palvelimesta tulee toimialueen ohjauspalvelin. (Simister, A. 2024.) Korkean käytettävyyden ja vikasietoisuuden takaamiseksi Windows Server Administrationin parhaat käytännöt suosittelevat vähintään kahden ohjauspalvelimen asentamista Windows-toimialueelle (Euroopan tietotekniikan sertifiointilaitos EITCI, 2023).

Active Directory -toimialuepalvelu sisältää useita palveluja, jotka mahdollistavat hakemistoon tallennettujen tietojen tehokkaan jakamisen, hallinnan ja valvonnan. Palveluihin kuuluvat toimialuepalvelut (Domain Services), sertifikaattipalvelut (Certificate Services), sovelluksille tarkoitettu rajapinta hakemiston käyttämiseen (Lightweight Directory Services, AD LDS), autentikoinnin ohjauspalvelut (Directory Federation Services) sekä käyttöoikeushallinnan palvelut (Rights Management Services, AD RMS). Active Directory -toimialuepalvelun etuja ovat keskitetty hallinta, skaalautuvuus, rooli- ja käyttöoikeuspohjainen todentaminen ja valtuuttaminen sekä kertakirjautuminen verkkoresurssien käyttämiseksi. Palveluiden hyödyntäminen vaatii kuitenkin erikoisosaamista ja tuotelisenssit voivat nostavat ympäristön hintaa. Ympäristö on myös haavoittuvainen tietoturvahyökkäyksille ja vaatii jatkuvaa ylläpitoa. (GeeksforGeeks, 2023.)

2.3 SQL Server

SQL Server on Microsoftin kehittämä relaatiotietokannan hallintajärjestelmä. Kehitys aloitettiin yhteistyössä toisen yhtiön, Sybasen, kanssa 1980-luvun lopulla. Microsoft ja Sybase lopettivat yhteistyön 1990-luvun puolessa välissä, minkä jälkeen Microsoft jatkokehitti SQL Server -tuotetta ja julkaisi ensimmäisen oman versionsa (7.0) vuonna 1998. (McQuillan 2015, 1.) SQL Server -tuotteen kehityksellisesti merkittävimmät muutokset tulivat SQL Server 2005 -version myötä. Versio mahdollisti uudemman laitteiston ja sovelluksiin tehtyjen parannusten hyödyntämisen, kuten 64-bittisen arkkitehtuurin käytön, parannetun tuen useiden prosessorien käytölle ja paremman yhdenmukaisuuden .NET-kehityksen kanssa. (Radivojević ym. 2019, 1.) SQL

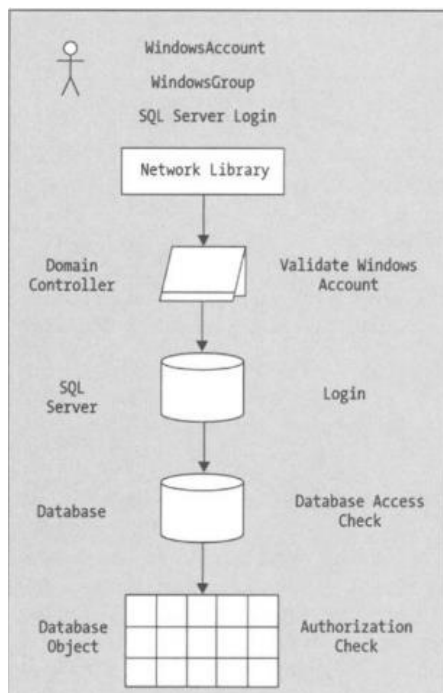
Server -tietokantojen käsittelyyn ja hallintaan on useita työkaluja, mutta Microsoftin kehittämä SQL Server Management Studio (SSMS) on yksi tunnetuimmista (Taylor 2023). SSMS on graafisella käyttöliittymällä varustettu monipuolinen työkalu SQL-ympäristön hallintaan (Microsoft 2023c).

SQL Server -tuotteesta on viisi eri päätyyppiä: Enterprise, Standard, Web, Developer ja Express. SQL Server Enterprise on ominaisuuksiltaan kattavin ja Express tyypisteyin ja ilmainen käyttää. Jokainen päätyyppi on suunniteltu tietynlaiseen käyttötarkoitukseen. (Peterson 2024.)

SQL Server sisältää useita komponentteja, joista tärkein on tietokantamoottori (Database Engine). Tietokantamoottori tallentaa, käsittelee ja suojaa tietoa mahdollistaen samalla tehokkaan tiedonvaihdon sovellusrajapinnan ja SQL-ilmentymän välillä. Muut SQL Server -komponentit keskittyvät tiedonhallinnan ja hyödyntämisen eri osa-alueisiin kuten koneoppimiseen, integraatioihin, raportointiin, laatuun, analysointiin ja monistettavuuteen saatavuuden takaamiseksi. (Microsoft 2024a.)

2.3.1 SQL Server -käyttöoikeudet

SQL Server -käyttöoikeushallinta koostuu kahdesta päätasosta: todennuksesta (authentication) ja valtuutuksesta (authorization). Todennuksella tarkoitetaan käyttäjän tunnistamista ja valtuutuksella käyttäjän käyttöoikeuksien ja -rajoitusten määrittämistä. Todennus tehdään joka kerta kun käyttäjä yrittää tehdä mitä tahansa toimenpidettä SQL Server -tietokantaan tai sen asetuksiin. Käyttöoikeushallinnan asetuksia ja ongelmia tutkittaessa ymmärrys todennuksen ja valtuutuksen solmukohdista (Kuva 1) auttaa hallitsemaan käyttäjäkohtaisia asetuksia. (Lewis, 2004, 2–3.)



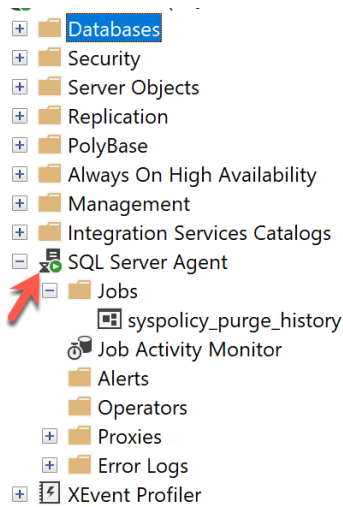
Kuva 1. SQL Server käyttöoikeushallinnan tiekartta (Lewis 2004, 3).

SQL Server -käyttöoikeushallinta on monitasoinen ja ensimmäisessä vaiheessa käyttäjä yritetään tunnistaa (Lewis, 2004, 1). Käyttäjä voi kirjautua toimialuetunnuksella jolloin SQL Server käyttää Windows-tunnuksen todennusta tai SQL Server -tietokannan SQL Server -todennuksella (SQL Server Authentication). (Potgieter 2022.) SQL Server -ilmentymän tasolla hallitaan käyttäjän todennusta, mutta SQL Server -tietokantojen käyttäjätunnukset ja niihin liittyvät oikeudet ovat tietokantatasolla. Käyttäjän todennuksen jälkeen tarkistetaan käyttäjän oikeudet tietokantoihin ja niissä oleviin objekteihin. (Lewis, 2004, 1.) Ilmentymän tasolla olevaan tunnukseen voi sijoittaa useita eri käyttäjiä eri tietokannoista ilmentymän sisällä, mutta sitä ei voi sijoittaa useisiin käyttäjätunnuksiin tietyn tietokannan sisällä (Microsoft 2023b).

2.3.2 SQL Server Agent

SQL Server Agent -palvelu on SQL Server -palvelimen Windows-palvelu, joka mahdollistaa tehtävien ajastuksen SQL Server -ilmentymän sisällä. SQL Server Agentin käyttötarkoituksena on tehdä rutiininomaisista, toistuvasti suoritettavista

tehtävistä automaattisesti suoritettavia. SQL Server Agent -palvelua voi käyttää SQL Server Management Studion kautta (Kuva 2). (Das 2020.)



Kuva 2. SQL Server Agent tarkasteltuna SQL Server Management Studion kautta (Das 2020).

SQL Server Agent -palvelu koostuu viidestä komponentista (Microsoft 2023d):

1. *Job* – ohjelma joka sisältää ja määrittää säännöt toistettaville toiminnoille SQL Server -ympäristössä
2. *Step* – Jobin sisällä suoritettava yksittäinen tehtävä (*job step*) joka voidaan ajaa yhden tai useamman kerran. Suorituksen onnistuminen ja epäonnistuminen on valvottavissa. *Stepit* suoritetaan sarjassa ja jokainen *step* voidaan ajaa eri käyttöoikeuskontekstissa.
3. *Schedules* – Aikataulut määrittävät milloin *job* suoritetaan. Yksi aikataulu voi olla käytössä useissa *jobeissa* ja jokaisella *jobilla* voi olla useampia aikatauluja.
4. *Alerts* – Hälytykset reagoivat tiettyihin tapahtumiin SQL Server -ilmentymässä tai Microsoft Windows Management Instrumentation (WMI) -palvelun tapahtumien kautta palvelimella mihin SQL Server -ilmentymä on asennettu.
5. *Operators* – Operaattorilla määritetään vastuullisen ylläpitäjän tiedot, mutta se ei ole kiinnitetty käyttövaltuuksiin. Operaattoreille voidaan

lähettää automaattisia hälytyksiä esimerkiksi *net send* -komennolla tai sähköpostilla.

SQL Server Agent -palveluun voidaan lisätä proxy-käyttäjiä. Proxy-käyttäjien avulla voidaan määrittää suoritettavien *job stepien* käyttövaltuuskonteksti. Proxy-käyttäjälle määritetään alajärjestelmäoikeudet ja sitä voidaan käyttää sellaisten *job stepien* suorittamiseen, jotka on määritetty käyttämään proxy-käyttäjälle luvitettua alajärjestelmää. (Microsoft 2023d.)

2.3.3 Varmuuskopiointi ja tietokannan palauttaminen

Järjestelmän käyttämien tietojen säilyvyyden varmistaminen on järjestelmän toiminnan jatkuvuuden kannalta tärkeimpiä asioita. Varmuuskopio on tuotannon tietojen kopio, joka tallennetaan eri paikkaan kuin missä tuotannon käyttämät tiedot sijaitsevat. Säännöllisen varmuuskopion ottamisen tavoitteena on että tuotannon tietojen kohdatessa minkä tahansa tapahtuman joka tuhoaa tai korruptoi käytössä olevan tuotantojärjestelmän tiedon, voidaan tiedot palauttaa varmuuskopiosta ja järjestelmän toiminta palauttaa normaaliksi. (Jayaram 2018a.)

SQL Server -varmuuskopioita on 7 erilaista, joista tyypillisimpiä ovat (Jayaram 2018c):

- FULL BACKUP eli täysi varmuuskopio, joka on samalla pohja kaikille muille varmuuskopioille ja tulee aina ottaa ensimmäisenä
- DIFFERENTIAL BACKUP joka sisältää kaikki muutokset edellisen FULL-varmuuskopion jälkeen
- TRANSACTION LOG BACKUP jolla varmistetaan transaktioloki eli tietokannan tietojen käsittelyyn liittyvät lokitiedot
- TAIL LOG BACKUP jolla otetaan virhetilanteissa talteen reaaliaikainen tuotantokannan lokitieto ennen kannan palauttamista.

SQL Server -tietokannan palautusmalli on tietokannan ominaisuus, jonka perusteella määräytyy mitä tapahtumia lokitiedostoihin kirjataan, voiko

lokitiedostoja varmuuskopioida ja mitä palautustoimintoja on käytettävissä. Palautusmalleja ovat FULL (täysi), SIMPLE (yksinkertainen) ja BULK-LOGGED. Tuotantotietokannoille suositellaan käytettäväksi FULL-palautusmallia. (Jayaram 2018b.) Täysi varmuuskopio (FULL BACKUP) sisältää kaiken tietokannan tiedon lisäksi transaktiolokit ja on varmuuskopion valmistumishetkellä täydellinen kopio kaikesta tarvittavasta tiedosta tietokannan palauttamiseksi samaan tai erilliseen SQL Server -ympäristöön (Microsoft 2023e).

2.3.4 T-SQL

Transact-SQL on Microsoftin ja Sybasen kehittämä jatke standardisoidulle SQL (Structured Query Language) -kyselykielelle. T-SQL-kieltä käytetään pääasiassa Microsoft SQL Serverin sisällä ja poikkeaa hyvin vähän SQL-kielestä sisältäen kuitenkin enemmän toiminnallisuuksia. Muuttujien määrittämisen mahdollisuus, proseduraalinen lohkoissa tapahtuva tietojen käsittely, lisäfunktioiden käyttämisen mahdollisuus ja massalisäykset tiedostostoista ovat mahdollisia T-SQL-kielellä tehden siitä SQL-kieltä tehokkaamman. (Clark 2021.)

3 Tiedonsiirtoratkaisun vaatimukset ja rajoitukset

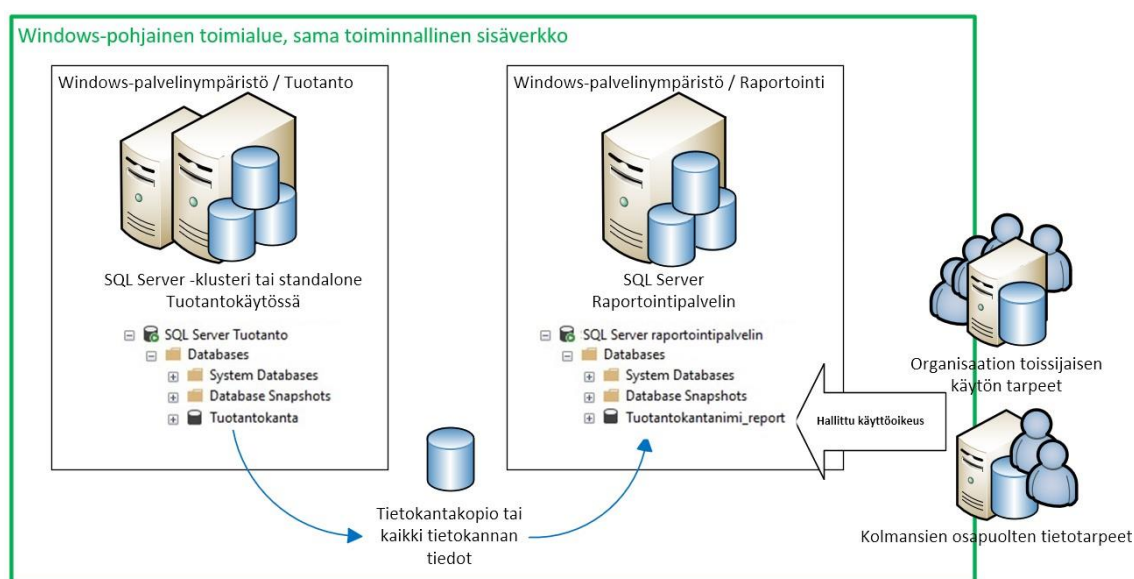
Tuotantotietokannasta raportointitietokantaan kopioitavan tietosisällön käyttötarkoitus on palvella asiakasorganisaation omia toissijaisia käyttötarpeita ja niistä johdettavia kolmansien osapuolien tietotarpeita. Kolmansien osapuolien käyttötarpeen perusteella tietojen tulisi päivittyä raportointipalvelimelle kerran vuorokaudessa ja olla ajantasaisia jokaisen päivän aamuyönä. Koska jatkokäytön tarpeiden palveleminen koskee asiakkaan toissijaisia toimintoja, siirrot eivät ole toiminnallisesti kriittisiä asiakasorganisaation ensisijaisen tehtävän toteuttamiseksi.

Asiakas ei ole valmis investoimaan erilliseen integraatiosovellukseen tai -työkaluun vaan toivoo, että tietojen kopiointi on toteutettavissa mahdollisimman kustannustehokkaasti tarpeeseen nähden. Mahdollisia tietosisällön rakenteellisia muutoksia ei tarvitse ottaa huomioon. Kyseisten muutosten seuranta- ja toteutusvastuu on siirretty raportointipalvelimen tietosisältöjä hyödyntävien kolmansien osapuolten vastuulle. Kustannustehokkuusvaatimukseen perustuen alustakapasiteetin muutokset toivotaan myös pidettävän mahdollisimman tarkoituksenmukaisena.

Tietokantojen kopioinnista ja siirrosta ei saa aiheutua riskiä tuotantotietokantojen tai -järjestelmän toiminnalle. Tekninen toteutus tulee olla toiminnallisesti irrallaan tuotannon prosesseista.

4 Käytettävän tiedonsiirtoratkaisun suunnittelu

Tarve saada useita eri tietokantoja siirrettyä raportointipalvelimelle automatisoidusti kerran päivässä ilman erillisiä integraatiotyökaluja vaati alustavalta suunnittelulta ympäristön työkalujen ja toteutusratkaisujen arviointia. Tekninen ympäristö ja siihen liittyvät tuotteet, hallintakäytännöt ja käytettävissä olevat palvelinalustat (Kuva 3) loivat asiakasvaatimusten lisäksi rajoituksia toteutuksen mahdollisille ratkaisuvaihtoehdoille.



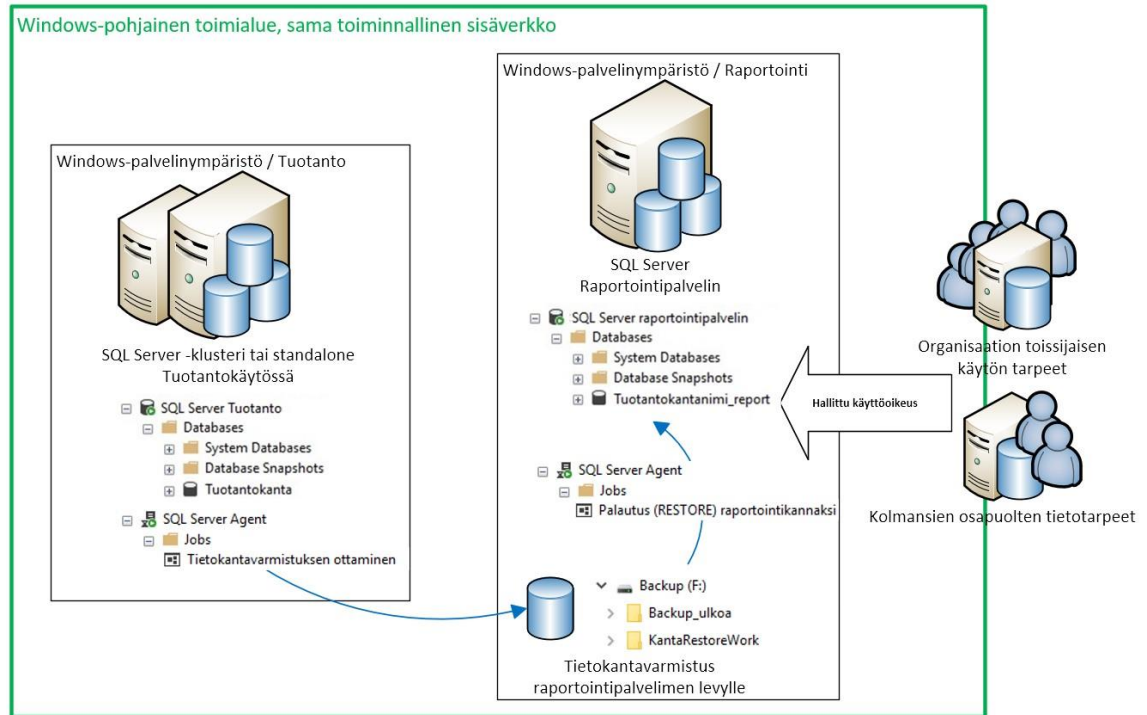
Kuva 3. Ympäristön ja perustarpeen lähtötilanne.

Vaihtoehdot tietojen siirtämiseen liittyivät automaattisesti joko SQL Server Integration Service -ominaisuuksien hyödyntämiseen tai kannan kaiken tiedon siirtämiseen jollain muulla tavalla. Kumulatiiviselle siirrolle, jossa siirrettävää tietoa verrattaisiin jo aiemmin siirrettyyn, ei ollut tarvetta. Kannan täydessä varmistuksessa otetaan kaikki tieto talteen joka kerta, joten SQL Server Integration Service oli tarpeeseen nähden huomattavasti työläämpi toteutusvaihtoehto. SQL Server Replication -tekniikkaa vaihtoehtona poissuljettiin välittömästi, koska asiakasvaatimuksena oli toteutuksen irrallisuus tuotantokäytöstä.

Raportointipalvelimen palvelinalusta oli jo olemassa SQL Server -asennuksena. Tuotantotietokannat käyttävät samaa tuotetta, joten varmistusten käyttämiseksi siirtomuotona ei ollut estettä. Siirto ei vaatisi tietojen konvertointia vaan suoraviivaisen palautuksen varmistuksesta uudelle palvelimelle.

Alustapalveluyrityksen ei-klusteroiduissa (standalone) SQL Server -palvelimissa varmistusten ottamiseen on, ennen erillisen varmistuksista vastaavan sovelluksen käyttöönottoa, käytetty Ola Hallengrenin luomaa julkisesti saatavilla olevaa T-SQL-skriptiä ja siihen liittyviä SQL Server Agent jobeja. Kyseinen varmistuksia ajava skripti oli täysin soveltuva tietokantakopiointia varten tuotettavan varmistuksen ottamiseen. Käytettävä toimintalogiikka pysyisi myös linjassa yrityksen aiemmin käyttämän ja hyväksymän toimintatavan kanssa.

Opinnäytetyössä harkittiin myös siirtoprosessien ajamista keskitytetystä yhdestä pisteestä. Tehtävien ajoitus (Task Scheduler) -toiminto on sisäänrakennettuna kaikissa Windows-tuotteissa ja sen kautta voidaan ajaa PowerShell-komentoja, batch-komentotiedostoja ja kutsua SQL-tietokantaan tallennettuja alirutiineja (Stored Procedure). Nämä kyvykkyydet tarvittavien luvitusten kera riittäisivät suorittamaan tietokantakopiointi hallitusti. Tehtävien ajoitus -toiminnolla on kuitenkin yrityksen sisällä havaittu olevan paljon ongelmia prosessien käynnistämisessä, joten siirtoprosessit päätettiin hajauttaa eri paikkoihin toimintavarmuuden lisäämiseksi. Opinnäytetyön tekijä päätyi ratkaisuun, jossa varmistusajot suorittaa tuotantopalvelin, tiedostokäsittelyn sekä kantaosoiden palautukset varmistuksista hoitaa raportointipalvelin (Kuva 4).



Kuva 4. Tietokantakopioinnin suunniteltu päätason rakenne.

Toteutuksen ylläpitovastuu on alustapalvelua tuottavalla yrityksellä, jolla ei ole omaa sovelluskehitystä ja hyvin minimaalinen integraatio-osaaminen. SQL Server -perusosaaminen on alustapalveluyrityksessä hyvin yleistä, joten kaikkien prosessien ajaminen SQL Server -tuotteen sisällä SQL Server Agentilla varmistaa että yritys kykenee jatkossakin toteuttamaan ylläpidon opinnäytetyössä rakennetulle kokonaisuudelle.

5 Tuotantotietokantojen kopiointi ja siirto raportointikantaan

Päätason suunnitelman pohjalta tuotettava tiedonsiirtoratkaisu vaatii toimiakseen käyttäjätunnuksen, joka luvitetaan tietokantoihin ja tarvittaviin kansioihin palvelimilla. Tietokannan varmistuksen ottaminen, varmistustiedoston käsitteleminen ja kohdepalvelimella tehtävän tietokantapalautuksen suorittaminen epäonnistuvat jos tunnuksella ei ole riittäviä oikeuksia. Tarvittava tunnus luodaan ja luvitetaan ensin, minkä jälkeen siirron asetukset rakennetaan systemaattisesti lähteestä kohteeseen.

5.1 Verkojaon tekeminen ja Active Directory -tunnuksen luvittaminen

SQL-tuotantotietokannoista ajettavat varmistukset tulee tallentaa fyysiseen levyjärjestelmään. Tallennusta varten raportointipalvelimelle tehtiin erillinen fyysinen levy, johon varmistukset voidaan jatkokäyttöä varten tallentaa. Varmistukset ajetaan siis suoraan raportointipalvelimen tallennuslevylle (Kuva 5), ei tuotantopalvelimen omille levyille. Tällä ratkaisulla vältetään tuotantopalvelimen levyn lukemista verkon yli, ylimääräistä erillistä tiedostosiirtoa varmistuksen ottamisen jälkeen ja tuotantopalvelimen kapasiteetin turhaa käyttöä.

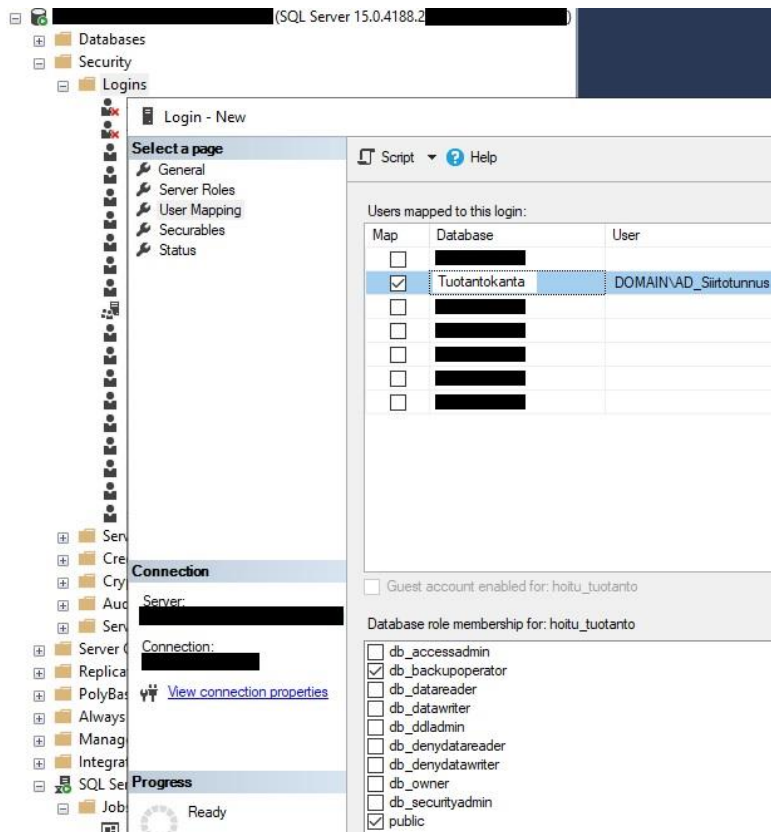


Kuva 5. Raportointipalvelimen levyosio.

Tietokantojen siirtoprosessin eri vaiheista vastaava AD-käyttäjätunnus luotiin AD_Siirtotunnus-nimellä. Tunnus toimii palvelutunnuksena yrityksen voimassaolevien käytäntöjen mukaisesti. Palvelutunnus ei tarvitse Active Directoryssä erikoisluvituksia, tärkeintä on että se on domainiin liitetty tunnus.

Käyttäjätunnus nimettiin käyttötarkoitusta kuvaavalla nimellä yrityksen nimeämispolitiikan mukaisesti ja sen tietoihin kirjoitettiin käyttötarkoitus.

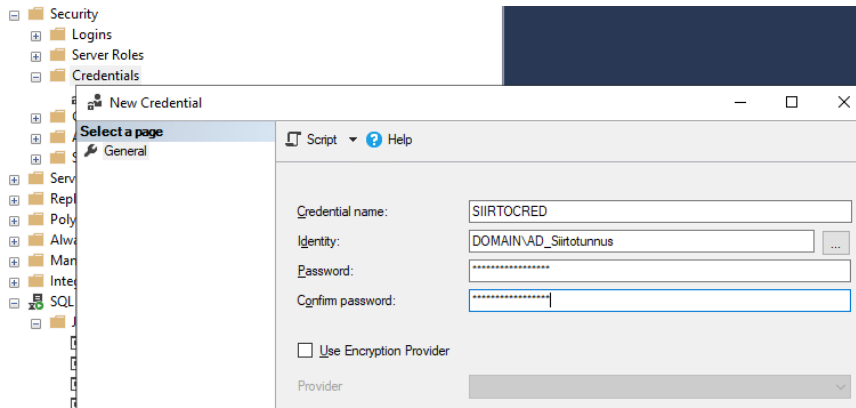
AD-käyttäjätunnus nimeltään DOMAIN\AD_Siirtotunnus luvitettiin tämän jälkeen tuotantotietokantoihin. Tunnus tulee lisätä jokaisen instanssin päätasolle *Security*-osioon ja sille tulee antaa palvelintasolla *public*-rooli ja User-Mapping-välilehdellä *db_backupoperator*-oikeus (Kuva 6).



Kuva 6. Esimerkki AD-tunnuksen luvituksesta tuotantoympäristön kantaosion User Mapping -välilehdellä.

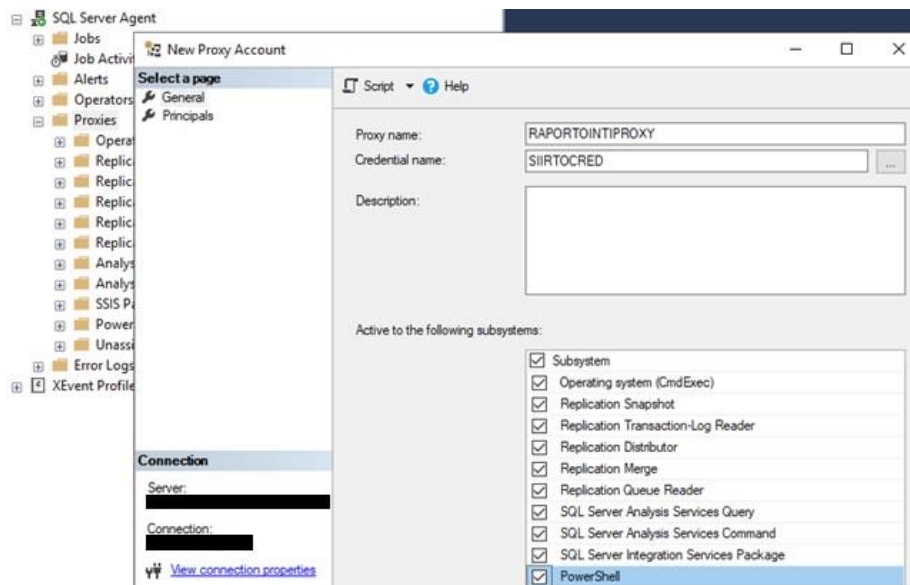
Tuotantotietokantojen varmistusten luomiseen käytetään SQL Server -tuotteen sisällä olevaa SQL Server Agent -komponenttia. SQL-instanssin päätasolle luotiin *Credentials*-tunnus, joka liitettiin tietokantainstanssiin luvitettuun AD-tunnuksen DOMAIN\AD_Siirtotunnus (Kuva 7). Credentials-tyyppinen tunnus perii käyttöoikeuksia Security-tason tunnuksen mukaisesti. Credentials-osioon

tarvitaan tunnus, jotta SQL Server Agentin alle voidaan luoda siihen liitetty proxy-käyttäjä.



Kuva 7. Credential-tunnuksen luominen.

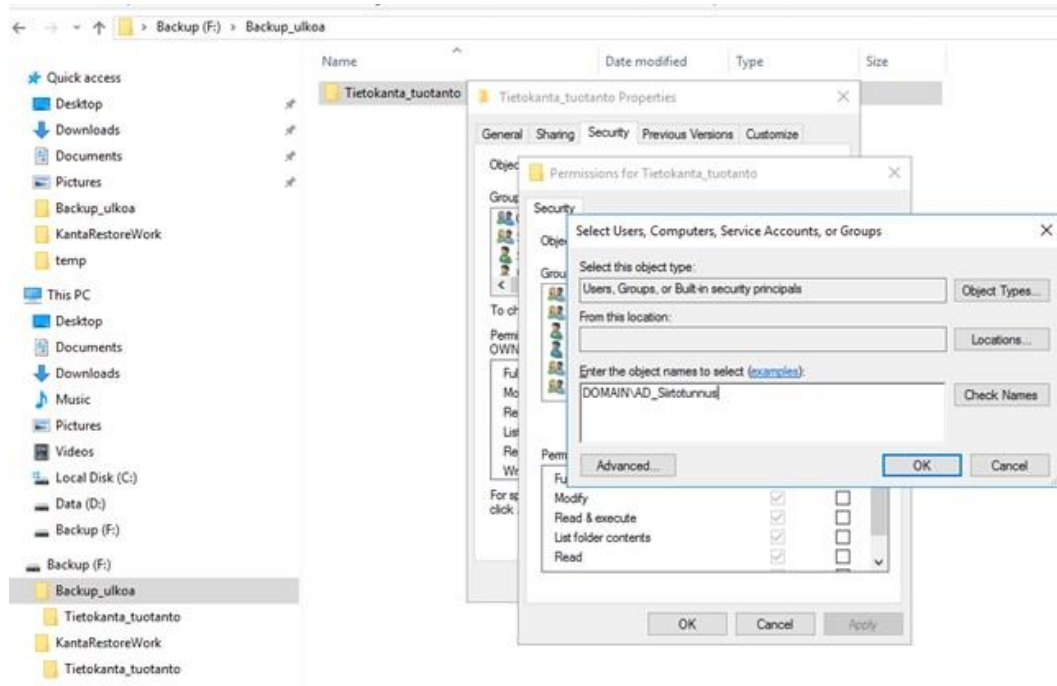
SQL Server Agentin alta *Proxies*-kohdasta luotiin uusi proxy-käyttäjä, joka liitettiin aiemmin tehtyyn *Credentials*-tunnukseen SIIRTOCRED. Proxy-tunnukselle annettiin tässä tapauksessa oikeudet kaikkiin subsystem-osiin (Kuva 8). Tarpeen mukaan oikeuksia voidaan vähentää.



Kuva 8. Proxy-tunnuksen luominen ja käyttöoikeudet.

Käytettävän AD-käyttäjätunnuksen käyttöoikeudet tulee varmistaa varmistusten kohdesijainnista raportointipalvelimelta. Varmistusten tallennuspaikkana

toimivan palvelimen paikallisen levyn kansiorakenteisiin annetaan tunnukselle *Modify*- eli *Muokkaus*-oikeudet (Kuva 9).

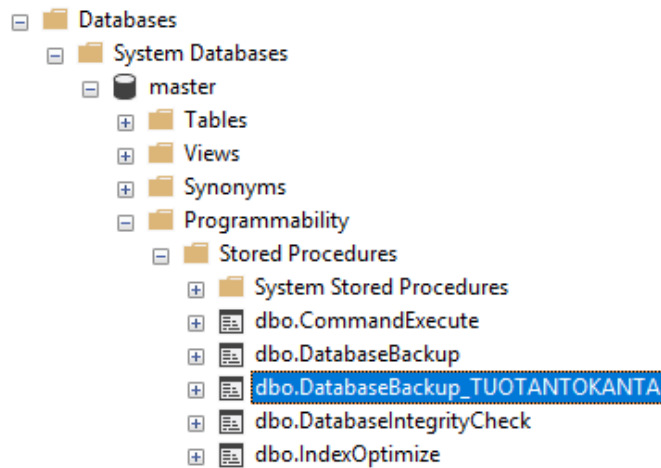


Kuva 9. AD-tunnuksen luvittaminen raportointipalvelimen kansiorakenteisiin.

Raportointipalvelimen tietokantaan AD-käyttäjätunnus luvitettiin *sysadmin*-oikeuksilla. Tarvittaessa oikeuksia voidaan vähentää, mutta tämän opinnäytetyön työympäristössä tunnuksen oikeudet ovat tähän opinnäytetyöhön liittymättömistä syistä korotetut.

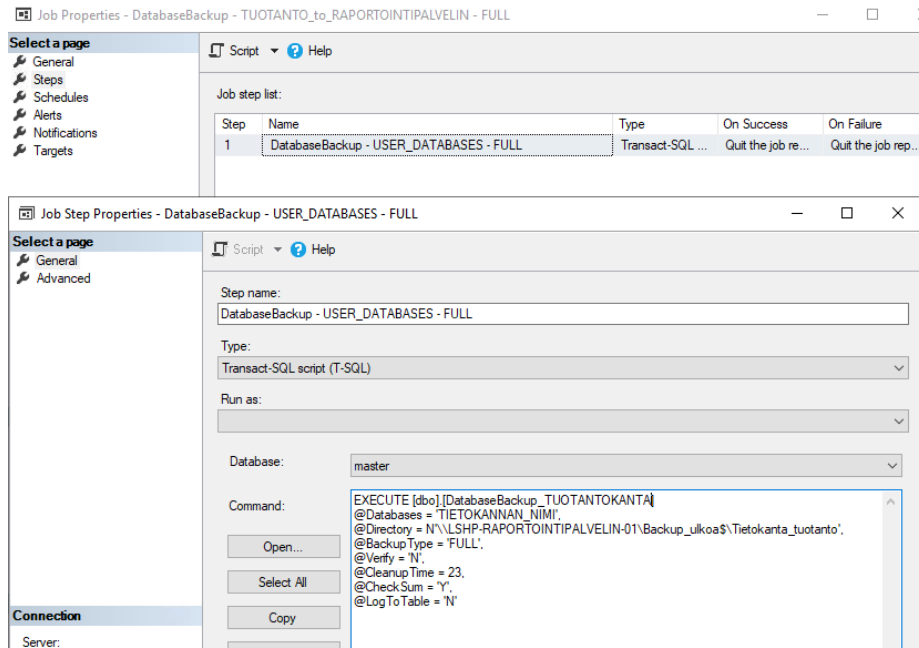
5.2 Varmistusten ottaminen tuotantoympäristöstä

Tietokantojen siirron suorittamisen asiakasvaatimuksena on prosessien pitäminen erillään tuotannon toiminnasta, joten varmistusten ajamiseen käytettävä skripti luotiin erikseen jokaiseen SQL-instanssiin, josta varmistuksia tuli siirtoa varten ottaa. Kysymyksessä on Ola Hallengrenin kehittämä ja ylläpitämä julkisesti saatavilla oleva T-SQL-skripti. Kyseistä skriptiä käytetään kuitenkin ainoastaan tämän tietokantasiirron tarpeisiin, joten se on nimetty kuvaavasti (Kuva 10).



Kuva 10. Tuotantopalvelimen varmistusajon skriptin sijainti ja nimi.

Tuotantokäytössä olevassa SQL-instanssissa on jokaista siirrettävää tietokantaosiota varten luotu oma SQL Server Agent jobinsa, joka vastaa tietokannan varmistuksen ottamisesta raportointipalvelimen levyille. Jobia ajetaan aiemmin luodulla proxy-tunnuksella. Varmistusajossa on tärkeää ottaa huomioon, että varmistus on FULL-tyyppinen, polku tallennuskansioon on oikein ja CleanupTime-asetus on alle 24 tuntia (Kuva 11). Jos varmistusajon tyyppi on jotain muuta kuin FULL, ei tietokannasta saada kaikkia tarvittavia tietoja siirrettyä. CleanupTime-asetuksen määrittämisellä kerrotaan varmistusskriptille vanhojen varmistusten poistamisen aikaväli, eikä siirron toiminnan kannalta ole tärkeää pitää useita varmistuksia.



Kuva 11. Tuotantopalvelimen varmistusjobin asetukset.

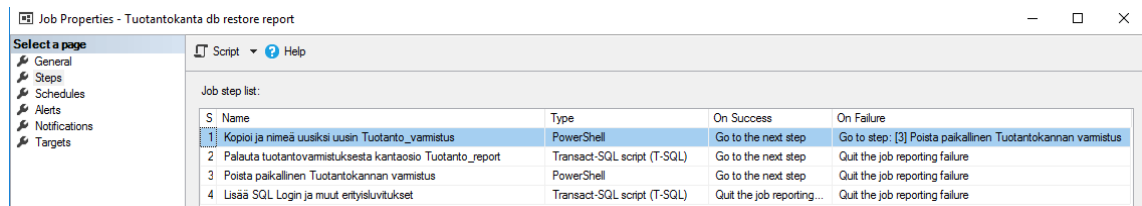
Varmistusten ajaminen raportointipalvelimen tallennuslevylle on ainoa tietokantojen siirron tekninen osa, jossa toimintoon käytettävät palvelinresurssit tulevat tuotannon kapasiteetista. Varmistusten ottaminen ei aiheuta haittaa tuotantokäytölle.

5.3 Tietokantojen ajaminen raportointipalvelimen kantaosioiksi tuotannon varmistuksista

Tietokantavarmistusten käyttäminen palautuksessa eri palvelimella ja eri kantaosion korvaamisessa asettaa muutamia huomioon otettavia poikkeuksia. Tuotantokannan ajama FULL-tyyppinen varmistus menee aina automaattisesti luodun kansiorakenteen alle varmistusajon kohdekansiossa. Varmistustiedoston nimi ei ole aina samanmuotoinen, joten siihen viittaaminen palautusta tekevässä skriptissä on vaikeaa ilman valmistelevia toimenpiteitä. Palautusta tehdessä on myös huomioitava että alkuperäinen tietokantanimi ei ole sama kuin raportointikannassa olevan tietokannan nimi, joten SQL-

tietokantapalautusta tehtäessä tulee varmistustiedoston nimet ja fyysiset tallennuspaikat muuntaa vastaamaan raportointipalvelimen vastaavia asetuksia.

Jokaiselle tuotannosta raportointipalvelimelle tuotavalle tietokannalle on raportointipalvelimella oma SQL Server Agent jobinsa, joka koostuu neljästä *stepistä* (Kuva 12). Erittely helpottaa vianselvitystä ja antaa selkeän perusrakenteen siirrolle, mikä helpottaa ratkaisun monistettavuutta.



Kuva 12. SQL Server Agent job -paketissa ajettavat vaiheet.

Raportointipalvelimella on jokaiselle siirrettävälle tuotannon kantaosiolle tehty ennakoita sitä vastaava raportointikanta samalla kollaatiolla kuin mitä tuotantokantaosio käyttää. Nimeämisessä on käytetty alkuperäisen tuotantokannan nimeä ja lisätty sen perään ”_report”.

5.3.1 Varmistusten siirto ja nimeäminen

Jokaiselle tuotantotietokannasta ajettulle varmistukselle on raportointipalvelimen tallennuslevyllä oma kansionsa. Tuotantotietokannan tekemän varmistuksen nimi tulee saada vakioitua, joten ensimmäinen SQL Server Agent jobin *step* käyttää PowerShell-skriptiä hakeakseen tallennuspolusta uusimman .bak-tyyppisen tiedoston, vie sen uuteen paikkaan samalla tallennuslevyllä ja nimeää sen vakiona aina samalla nimellä (Kuva 13). Näin polkuun ja nimeen voidaan jatkossa viitata käyttäen staattista viittausta.

```

#Mistä uusin kantaosion backup haetaan
$BackupDir = "F:\Backup_ulkoa\tietokanta_tuotanto\TuotantopalvelimenAutogeneroima_po\ku\Kantaosio_automattinen_nimi\FULL"
#Mihin kantaosion backup viedään luettavaksi
$WorkDir = "F:\KantaRestoreWork\tietokanta_tuotanto"
#Mikä tiedosto siirretään nimeämisen jälkeen
$ToMove = "F:\Backup_ulkoa\tietokanta_tuotanto\TuotantopalvelimenAutogeneroima_po\ku\Kantaosio_automattinen_nimi\FULL\tietokantaNimi_raportointi|.bak"
Set-Location $WorkDir
$UusinBackupNimi = (Get-ChildItem $BackupDir\*.bak | sort LastWriteTime | select -last 1)
Rename-Item $UusinBackupNimi -NewName "TietokantaNimi_raportointi.bak"
Move-Item -path $ToMove -Destination $WorkDir\tietokantaNimi_raportointi.bak -Force

```

Kuva 13. PowerShell-skripti varmistusten siirtämiseen ja nimeämiseen.

Tietokantavarmistuksessa tuotettavat tiedostot saavat aina nimekseen tietyn pitkässä muodossa olevan nimen, joka vaihtelee varmistuksen ottamisajankohdan ja muiden tietojen perusteella. Staattisen nimen saaminen käytettävälle varmistustiedostolle helpottaa palautuksen tekemistä, koska palautuksessa käytettävät viittaukset voivat olla suoria viittauksia nimeen joka ei muutu joka kerta.

5.3.2 Tietokantojen palautusajolla raportointipalvelimen kantaosioiden korvaaminen

Step 2 käyttää T-SQL-skriptiä tehdäkseen raportointipalvelimen tietokantaosiolle *RESTORE*-toiminnolla palautuksen. Siirretystä ja uusiksi nimetystä varmistustiedostosta tarkistetaan oikeat tiedostonimet (Kuva 14), joita sitten käytetään korvaamaan loogiset tiedostonimet raportointipalvelimella ja ajetaan kannella *RESTORE*.

```

RESTORE FILELISTONLY
FROM DISK = 'D:\KantaRestoreWork\tietokantavarmistukseninnimi.bak'

```

Kuva 14. RESTORE FILELISTONLY -komennon ote palautusskriptistä.

Korvattava tietokanta raportointipalvelimelta tulee muuttaa sellaiseen muotoon, että palautusajo on ainoa yhteys tietokantaan. Tällä vältetään yhteiskäyttöestot, jotka voivat pysäyttää kannan palautustoiminnon (Kuva 15).

```
USE [master]
GO
ALTER DATABASE [RaportointikannanNimi] SET SINGLE_USER WITH ROLLBACK IMMEDIATE
GO
```

Kuva 15. SINGLE_USER-komennon ote palautusskriptistä.

Palautuskomennon (RESTORE) käyttämisessä on otettava huomioon että *RESTORE FILELISTONLY*-komennolla on haettu tiedot varmistuksessa olevista fyysisistä .mdf ja .ldf -tiedostoista, jotta ne voidaan kirjoittaa uusiksi fyysisiksi tiedostoiksi raportointikannalle. Skriptissä käytetään *REPLACE*-komentoa, joka ohjaa RESTORE-komentoa korvaamaan olemassa olevan tietokannan (Kuva 16). Ilman *REPLACE*-komentoa *RESTORE* yrittää tehdä uuden kantaosion sillä nimellä, mitä varmistuksessa on tietokannalle käytetty.

```
USE [master]
RESTORE DATABASE [RaportointikannanNimi]
FROM DISK = N'D:\KantaRestoreWork\TuotantokantaNimi_raportointi.bak'
WITH FILE=1,
-- Oikeiden .mdf ja .ldf -tiedostojen sijainti riippuu SQL Server -asennuksen sijainnista ja versiosta.
MOVE N'TuotantokantaNimi' TO N'D:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\DATA\RaportointikannanNimi.mdf',
MOVE N'TuotantokantaNimi_Log' TO N'D:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\DATA\RaportointikannanNimi.ldf',
NOUNLOAD, REPLACE, STATS = 1
GO
```

Kuva 16. RESTORE-komento palautusskriptissä.

Palautuksen jälkeen tietokantaan sallitaan useampia yhteyksiä ja tietokannan omistaja vaihdetaan (Kuva 17). Tietokannan omistajan roolilla on laajat käyttöoikeudet SQL-tietokannassa, joten sen korvaaminen raportointipalvelimella käytössä olevalla tunnuksella on tärkeää.

```
USE [master]
GO
ALTER DATABASE [RaportointikannanNimi] SET MULTI_USER
GO
USE [RaportointikannanNimi]
GO
EXEC sp_changedbowner 'Domain/TietokannanOmistajanTunnus'
```

Kuva 17. Kannan yhteyksien käsittelyn säätäminen ja omistajan vaihto skriptillä.

Varmistuksen mukana mukana tulleiden tiedostojen nimet muutetaan vastaamaan raportointipalvelimen kannan tiedostonimiä (Kuva 18). Ilman muutosta tiedostojen nimet vastaavat varmistuksessa olevan tuotantokannan nimiä.

```
USE master
GO
ALTER DATABASE [RaportointikannanNimi]
MODIFY FILE (NAME='TuotantokantaNimi', NEWNAME=' RaportointikannanNimi ')
GO
ALTER DATABASE [RaportointikannanNimi]
MODIFY FILE (NAME='TuotantokantaNimi_log', NEWNAME=' RaportointikannanNimi_log')
```

Kuva 18. Raportointikannan tiedostojen nimenvaihdot skriptillä.

Vaihdetaan kannan palautustila (RECOVERY MODE) eli se, missä laajuudessa sen voi palauttaa, yksinkertaiseksi (SIMPLE) (Kuva 19). Tämä minimoi lokitiedostojen käyttämän tilan ja sopii käyttötarkoitukseen, jossa ei ole tarvetta tietokannan palauttamiselle tiettyyn ajanhetkeen.

```
USE [master]
GO
ALTER DATABASE [RaportointikannanNimi] SET RECOVERY SIMPLE WITH NO_WAIT
GO
```

Kuva 19. RECOVERY MODE -asetuksen muokkaaminen skriptillä.

Alkuperäisen tietokannan loki sisältää mahdollisesti paljon tietoa, joka siirron ja uudelleenkirjoituksen jälkeen varaa ylimääräistä tilaa. Lokitiedoston varaamasta tilasta otetaan käyttämätön tilavarauus pois suorittamalla SHRINKFILE-komento (Kuva 20).

```
USE [RaportointikannanNimi]
GO
DBCC SHRINKFILE (N'RaportointikannanNimi_log', 1024)
GO
```

Kuva 20. SHRINKFILE-komennon ajamainen skriptillä.

Tietokannan korvaaminen täydellisesti varmistuksesta toistuvasti suoritettuna aiheuttaa korvattavalle kantaosiolle hiljalleen korruptoitumista. Ajamalla DBCC CHECKDB -komento (Kuva 21) jokaisen tietokantapalautuksen jälkeen varmistetaan palautetun tietokannan loogisesta ja fyysisestä eheydestä.

```
DBCC CHECKDB([RaportointikannanNimi]) WITH NO_INFOMSGS
```

Kuva 21. CHECKDB-komennon ajaminen raportointipalvelimen kantaosiolle.

Tietokantojen huoltoajoissa tehdään tyypillisesti ajastetusti DBCC CHECKDB -komennon ajaminen. Koska komento ajetaan jokaisen tietokannan palautuksen yhteydessä, voidaan huoltoajoa supistaa palautettavien tietokantojen osalta tai tarvittaessa poistaa kokonaan. Erittäin suurien tietokantojen palauttamisissa tämä voidaan jättää ajamatta ja suorittaa huoltoajolla erillisenä ajankohtana.

5.3.3 Käytettyjen varmistustiedostojen poisto ja SQL Login -luvitukset

SQL Server Agent jobin step 3 poistaa käytetyt varmistustiedostot ja polut, jotta ne eivät palautuksen jälkeen varaa tilaa raportointipalvelimen tallennuslevyltä. Poistamiseen käytetään PowerShell-komentoa, joka poistaa annetusta polusta lähtien kaikki tiedostot ja kansiorakenteet pakotetusti (Kuva 22). Tallennuspolun pääkansion alaista tietokantakohtaista kansiota ei saa poistaa, koska varmistuksia ajavan SQL Server Agent jobin polku viittaa siihen.

```
Remove-Item F:\Backup_ulkoa\TuotantokantaNimi_tuotanto\* -Recurse -Force  
Remove-Item F:\KantaRestoreWork\tietokannan_nimi_raportointi \* -Recurse -Force
```

Kuva 22. PowerShell-komennot käytettyjen kansiorakenteiden poistamiseen.

SQL-instanssin päätasolla Security-osiossa annettavat luvitukset pätevät kantaosioihin myös palautuksen jälkeen. SQL Login -tyyppisten tunnusten käyttöoikeudet kuitenkin poistuvat palautuksen yhteydessä. Käyttöoikeudet

tulee palautuksen jälkeen asettaa raportointipalvelimen tietokantaosioon erillisillä komennoilla. Käyttöoikeuksia voidaan rajata tarpeen mukaisesti. Käytössä oleva SQL Login-tunnus *TEST_SQL_Login* on luotu SQL-instanssin päätasolle. Varmistuksesta palautettuun tietokantaan asetetaan tässä tapauksessa SQL Login -tunnukselle tarvittavat oikeudet lukea tietokannan kaikki tiedot *db_datareader*-roolin kautta (Kuva 23).

```
ALTER DATABASE [RaportointikannanNimi] SET TRUSTWORTHY ON
USE [RaportointikannanNimi]
GO
CREATE USER [SQLLoginTunnuksenNimi] FOR LOGIN [SQLLoginTunnuksenNimi]
GO
USE [RaportointikannanNimi]
GO
ALTER ROLE [db_datareader] ADD MEMBER [SQLLoginTunnuksenNimi]
GO
```

Kuva 23. SQL Login -tunnuksen luvittaminen kantaosioon skriptillä.

SQL Login -tunnuksia voidaan luvittaa useita peräkkäin. Oikeudet voivat tunnuskohtaisesti vaihdella tarpeen mukaan.

6 Yhteenveto

Opinnäytetyön tavoitteena oli tuottaa kustannustehokas SQL-tuotantotietokantojen kopiointi- ja siirtotapa erilliseen SQL-instanssiin, jotta järjestelmään syötettyjä tietoja voidaan hyödyntää vaarantamatta palvelutuotannon toimintaa. Tietojen kopiointin tekniset toiminnot tuli prosessina pitää erillään tuotannon muista toiminnoista ja tietojen tuli päivittyä joka yö. Opinnäytetyössä kuvattu toimintamalli toteuttaa asiakkaan vaatimukset ja ympäristön asettamat reunaehdot.

Varmistusten koon kasvaessa on kuitenkin uudelleenarvioitava tietojen päivityssyklin tiheyttä. Mitä suurempi tietokantavarmistus on, sitä kauemmin tarvitaan aikaa opinnäytetyössä esitellyn prosessin läpiviemiseen. Näin ollen tieto päivityssyklistä ja siirrettävä tietomäärä määrittelevät siirtomenetelmän sopivuuden.

Tietokantavarmistusten palautuksessa siirtyvät myös tietokantakohtaiset luvutukset. Näitä ei tässä opinnäytetyössä esitellyssä tietokannan siirtoprosessissa käsitelty millään tavalla, mutta ne aiheuttavat matalan tason käyttöoikeusriskin siirron kohdepalvelimella. Luvutukset tulisi olla aina tarkoituksenmukaiset väärinkäytösten minimoimiseksi.

Siirrettävien tietokantojen tietosisällöjen perusteella tulee myös ottaa huomioon lokituksen tarve kohdepalvelimella ja sen tietokannassa sekä mahdollinen tarve tietojen pseudonymisoinnille tai poistamiselle. Sensitiivisten tietojen käsittelystä säädetään useassa laissa, ja on rekisterinpitäjän vastuulla, että vaatimukset toteutuvat.

Opinnäytetyössä kuvattu SQL-tietokantojen siirtotapa on kustannustehokas tapa kopioida tuotantokantojen tietoa erilliseen instanssiin ja sopii hyvin matalan kynnyksen toteutuksena useisiin käyttötapauksiin. Tapauskohtainen arviointi sopivasta tietojen siirtotavasta on kuitenkin aina tehtävä erikseen. Vaikka päädyttäisiinkin hyödyntämään tämän opinnäytetyön tapaa, tulisi teknistä

toimintamallia arvioida ja jatkojalostaa, jotta se täyttää lain, tietoturvan ja viime kädessä asiakkaan vaatimukset.

Lähteet

Ceccobao, K. 10.2.2021. Unleash the the power of data with knowledge and insights. Viitattu 7.4.2024. <https://www.microsoft.com/en-gb/industry/blog/cross-industry/2021/02/10/unleash-the-power-of-data/>.

Clark, D. 4.3.2021. SQL vs T-SQL: Understanding the Differences. Viitattu 26.5.2024. <https://www.dataquest.io/blog/sql-vs-t-sql/>.

Das, A. 15.12.2020. Introduction to the SQL Server Agent. Viitattu 26.5.2024. <https://www.sqlshack.com/introduction-to-the-sql-server-agent/>.

Euroopan tietotekniikan sertifiointilaitos EITCI, 15.12.2023. Joten lause "Windows-toimialueen rakentamiseen tarvitaan vähintään 2 toimialueohjainta." tarkoittaa parasta käytäntöä?. Viitattu 25.5.2024.

<https://fi.eitca.org/tietoverkkojen/eitc-on-wsa-Windows-Server--hallinta/ikkunoiden-k%C3%A4ytt%C3%B6%C3%B6notto/johdatus-Windowsin-toimialueeseen-ja-toimialueen-ohjaimen/joten-lauseke-Windows-toimialueen-rakentamiseen-tarvitaan-v%C3%A4hint%C3%A4n-2-toimialueen-ohjausyksikk%C3%B6%C3%A4%2C-tarkoittaa-parasta-k%C3%A4yt%C3%A4nt%C3%B6%C3%A4/>.

Ford, T. n.d. Getting started with SQL Server clustering. Viitattu 7.4.2024. <https://www.mssqltips.com/sqlservertip/1541/getting-started-with-sql-server-clustering/>.

GeeksforGeeks, 4.5.2023. Introduction of Active Directory Domain Services. Viitattu 25.5.2024. <https://www.geeksforgeeks.org/introduction-of-active-directory-domain-services/>.

Hoffman, C. 4.8.2014. What is a Windows Domain and How Does It Affect My PC?. Viitattu 25.5.2024. <https://www.howtogeek.com/194069/what-is-a-windows-domain-and-how-does-it-affect-my-pc/>.

Jayaram, P. 1.3.2018a. An overview of the process of SQL Server backup-and-restore. Viitattu 26.5.2024. <https://www.sqlshack.com/overview-sql-server-backup-restore-process/>.

Jayaram, P. 10.4.2018b. Understanding SQL Server database recovery models. Viitattu 26.5.2024. <https://www.sqlshack.com/understanding-database-recovery-models/>.

Jayaram, P. 19.4.2018c. Understanding SQL Server Backup Types. Viitattu 26.5.2024. <https://www.sqlshack.com/understanding-sql-server-backup-types/>.

Lewis, M. 2004. SQL Server Security Distilled, Second Edition. Yhdysvallat: Springer-Verlag New York, Inc.

McQuillan, M. 2015. Introducing SQL Server. Kalifornia: Apress Media, LLC.

Microsoft, 17.8.2022a. Active Directory Domain Services Overview. Viitattu 7.4.2024. <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>.

Microsoft, 19.11.2022b. Database Engine Instances (SQL Server). Viitattu 7.4.2024. <https://learn.microsoft.com/en-us/sql/database-engine/configure-windows/database-engine-instances-sql-server?view=sql-server-ver16>.

Microsoft, 10.4.2022c. Collation Types. Viitattu 7.4.2024. [https://learn.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms175194\(v=sql.105\)](https://learn.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms175194(v=sql.105)).

Microsoft, 3.3.2023a. SQL Server Agent. Viitattu 25.5.2024. <https://learn.microsoft.com/en-us/sql/ssms/agent/sql-server-agent?view=sql-server-ver16>.

Microsoft, 16.3.2023b. Get started with Database Engine permissions. Viitattu 26.5.2024. <https://learn.microsoft.com/en-us/sql/relational-databases/security/authentication-access/getting-started-with-database-engine-permissions?view=sql-server-ver16>.

Microsoft, 31.3.2023c. What is SQL Server Management Studio (SSMS)?. Viitattu 26.5.2024. <https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16>.

Microsoft, 8.3.2023d. What is PowerShell?. Viitattu 7.4.2024. <https://learn.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7.4>.

- Microsoft, 4.12.2023e. Full database backups (SQL Server). Viitattu 26.5.2024. <https://learn.microsoft.com/en-us/sql/relational-databases/backup-restore/full-database-backups-sql-server?view=sql-server-ver16>.
- Microsoft, 8.3.2023f. Windows Management Instrumentation. Viitattu 10.6.2024. <https://learn.microsoft.com/en-us/windows/win32/wmisdk/wmi-start-page>.
- Microsoft, 14.2.2024a. What is SQL Server?. Viitattu 10.6.2024. <https://learn.microsoft.com/en-us/sql/sql-server/what-is-sql-server?view=sql-server-ver16>.
- Microsoft, 22.4.2024b. Group Policy overview. Viitattu 10.6.2024. <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/group-policy/group-policy-overview>.
- Peterson, R. 24.2.2024. What is SQL Server? Introduction, History, Types, Versions. Viitattu 10.6.2024. <https://www.guru99.com/sql-server-introduction.html>.
- Potgieter, J. 10.11.2022. Setup Security for a SQL Server Database via SSMS and T-SQL. Viitattu 26.5.2024. <https://www.mssqltips.com/sqlservertip/7447/sql-server-database-security-logins-users/>.
- Radivojević, M.; Sarka, D.; Durkin, W.; Coté, C. & Lah, M. 2019. Mastering SQL Server 2017. Iso-Britannia: Packt Publishing Ltd.
- Simister, A. 24.4.2024. What Is Active Directory and How Does It Work?. Viitattu 25.5.2024. <https://www.lepide.com/blog/what-is-active-directory-and-how-does-it-work/>.
- Stegner, B. 22.3.2018. What is a Windows Domain and What Are Its Advantages?. Viitattu 25.5.2024. <https://www.makeuseof.com/tag/windows-domain/>.
- Taylor, P. 2023. Ranking of the most popular database management systems worldwide, as of September 2023. Viitattu 26.5.2024. <https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems/#:~:text=As%20of%20September%202023%2C%20the%20most%20popular%20database,Microsoft%20SQL%20server%20rounded%20out%20the%20top%20three.>

W3Schools, n.d. What is SQL?. Viitattu 7.4.2024.
https://www.w3schools.com/whatis/whatis_sql.asp.