



TEKNIikka JA LIIKENNE

Tietotekniikka

Ohjelmistotekniikka

OPINNÄYTETYÖ

SQL SERVER 2005:N KÄYTTÖNOTTO VEIKKAUS OY:SSA

Työn tekijä: Sami Koivusaari
Työn ohjaaja: leht. Juhani Rajamäki
Työn ohjaaja: järj.pääl. Johannes Ollikainen

Työ hyväksytty: 2.12.2008

OPINNÄYTETYÖN TIIVISTELMÄ

Työn tekijä: Sami Koivusaari	
Työn nimi: SQL Server 2005:n käyttöönotto Veikkaus Oy:ssa	
Päivämäärä: 27.11.2008	Sivumäärä: 33 s
Koulutusohjelma: Tietotekniikka	Ammatillinen suuntautuminen: Ohjelmistotekniikka
Työn ohjaaja: lehtori Juhani Rajamäki Työn ohjaaja: järjestelmäpäällikkö Johannes Ollikainen	
<p>Insinööriyön tavoitteena oli tutkia SQL Server 2005:n ominaisuuksia ja dokumentoida sen käyttöönotto Veikkaus Oy:n sisäiset järjestelmät -osastolla. Uusi tuote hankittiin, koska vanha SQL Server 2000 alkoi olla jo vanhentunut järjestelmä.</p> <p>Tarkoituksena oli siirtää arkisto -sovellus uuteen järjestelmään sekä perehdyttää käyttäjiä sekä järjestelmän ylläpitäjiä sen käyttöön ja ylläpitoon. Insinööriyössä on dokumentoitu arkistosovelluksen siirto ja varmistaminen SQL Server 2005 -ympäristöön. On myös mietitty jatkotoimenpiteitä.</p> <p>SQL Server 2005:een siirtymisen edut näkyivät heti arkistosovelluksen nopeampana toimintana käyttäjille. Lisäksi virhetilanteet ovat vähentyneet huomattavasti.</p>	
Avainsanat: SQLServer2005, tietokanta, tietokannan siirtäminen, tietokannan varmistaminen	

ABSTRACT

Name: Sami Koivusaari	
Title: Implementation of SQL Server 2005 at Veikkaus Oy	
Date: 27 November 2008	Number of pages: 33
Department: Information Technology	Study Programme: Programming
Instructor: Johannes Ollikainen, Systems Manager	
Supervisor: Juhani Rajamäki, Senior Teacher	
<p>The aim of thesis paper was to study attributes of SQL Server 2005 and to document its implementation into the internal systems department at Veikkaus Oy. The new system was acquired because the old SQL Server 2000 was outdated.</p> <p>The purpose was to transfer arkisto –application into the new system as well as to familiarize users and system administrators in its use. The changeover and back-up of the arkisto –application into the SQLServer2005 environment have been documented in this thesis, including ideas for future actions.</p> <p>The advantages of the changeover into SQLSvrer2005 were evident as arkisto –application became faster for users. Error conditions have also decreased noticeably.</p>	
Keywords: SQLServer2005, database, database transfer, database back-up	

Lyhenne- ja käsiteluettelo

ACID	<i>Analysis Console for Intrusion Databases</i> , analyysimoottori, joka etsii ja käsittelee tietokantojen turvallisuuteen liittyviä tapahtumia.
Argent	Argent Software on amerikkalainen yritys, joka on valmistanut Argentin valvontajärjestelmän
Axapta	Microsoftin kehittämä liiketoiminnanohjausratkaisu.
BizTalk Server	Microsoftin kehittämä integrointipalvelin, joka helpottaa yrityksen liiketoimintaprosessien automatisointia ja hallintaa.
BSD	<i>Berkeley Software Distribution</i> , yksi käytetyimmistä avoimen lähdekoodin lisensseistä.
C	Ohjelmointikieli.
C++	Yksi tärkeimmistä kaupallisessa ohjelmistokehityksessä käytettävistä ohjelmointikielistä.
CRM	Microsoftin kehittämä liiketoiminnanohjausratkaisu.
DB2	IBM:n kaupallinen tietokantatuote.
EC	<i>External Connector</i> , palvelinohjelmistoissa käytettävä käyttöoikeus ulkoisille käyttäjille.
GNU GPL	<i>General Public License General Public License</i> , vapaa ohjelmistolisenssi.
JDBC	<i>Java Database Connectivity</i> , universaali ja toimittajariippumaton standardi, jonka mukaan Java-sovellukset voivat operoida SQL-tietokantojen kanssa.
MyODBC	MySQL -tietokantaan oleva rajapinta tietokantojen ja sovellusten välille.
Navision	Microsoftin kehittämä liiketoiminnanohjausratkaisu.
ODBC	<i>Open Database Connectivity</i> , Microsoftin kehittämä standardi rajapinta tietokantojen ja sovellusten välille.
Perl	<i>Practical Extraction and Report Language</i> , suosittu ohjelmointikieli unix-ylläpitäjien keskuudessa.
PHP	<i>Hypertext Preprocessor</i> , ohjelmointikieli, jota käytetään erityisesti Web-palvelinympäristöissä.
Python	Ohjelmointikieli, joka on alun perin kehitetty yhdistämään skriptikielten ja tavanomaisten ohjelmointikielten hyvät puolet.

RDBMS	<i>Relational database management system</i> , ohjelma jolla hallitaan relaatiotietokantoja.
Ruby	Ohjelmointikieli, joka muistuttaa Pythonia ja Perliä.
Smalltalk	Olio-ohjelmointikieli.
SQL	<i>Structured Query Language</i> , IBM:n kehittämä standardoitu kyselykieli.
System R	Maailman ensimmäinen relaatiotietokantaohjelma.
Visual Studio	Microsoftin ohjelmankehitysympäristö, joka tukee useita ohjelmointikieliä.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

LYHENNE- JA KÄSITELUETTELO

1	JOHDANTO	1
2	SQL SERVER 2005	1
2.1	Yleistä	1
2.2	Lisensiointivaihtoehdot	3
2.3	Versiot	5
2.4	SQL Server 2005:n alusta	5
3	TIETOKANTA	7
3.1	Tietokantojen historiaa	7
3.2	Tietokannat käytännössä	8
3.3	Transaktiot ja samanaikaisuus	9
3.4	Eri tietokantajärjestelmiä	10
3.4.1	<i>Yleistä</i>	10
3.4.2	<i>Oracle</i>	10
3.4.3	<i>MySQL</i>	11
3.4.4	<i>PostgreSQL</i>	11
4	ARKISTOSOVELLUKSEN SIIRTO KESKITETTYYN SQL SERVER 2005 - TIETOKANTARATKAISUUN	12
5	ARKISTOSOVELLUKSEN SIIRTO	14
5.1	Tietokantaan kirjautuminen palautuksen jälkeen	17
5.1.1	<i>Aikaisemmassa SQL Server 2000 -versiossa onnistuttiin näin:</i>	18
5.1.2	<i>SQL SERVER 2005:ssä tehdään</i>	20
5.2	Tietokannan varmistaminen	26
5.3	SQL Server Agent	29
6	YHTEENVETO	30
	VIITELUETTELO	32

1 JOHDANTO

Veikkaus Oy:n toimistoverkossa on käytössä SQL Server 2000 -tietokantaohjelmia asennettuna noin 12 palvelimessa. Palvelinalustana on käytössä Windows 2000 Server tai Windows Server 2003. Kaikki SQL Server -versiot ovat Standard-tasoisia. SQL Servereihin on pääsääntöisesti asennettu vain yksi kanta ja lisäksi palvelimessa pyörii aina kantaa käyttävä sovellus. Palvelimet eivät ole välttämättä tehokkaassa käytössä. Aina kun on tarve uudelle kannalle esimerkiksi vaikka väliaikaiseen käyttöön, aletaan rakentaa uutta palvelinta, koska tietokantapalvelimissa pyörii aina myös jokin tuotannollinen sovellus. Tämä ei ole kovinkaan käytännöllistä.

Insinööriyön tavoitteena oli tutkia uuden SQL Server 2005:n ominaisuuksia ja tuotteen soveltuvuutta Veikkaus Oy:n toimistoverkkoon ja ensisijaisesti dokumentoida uuden SQL Server 2005 tekninen käyttöönotto. Tutkittiin vaihtoehtoisena SQL Server -arkkitehtuurina järeämpiä tietokantapalvelimia, joissa toimisi useampi tietokanta omissa instansseissaan ja sovellukset olisivat omissa palvelimissaan. Tässä on otettava huomioon palveluiden jatkuvuus katastrofitilanteissa. Tutkittava replikoituja tai klusteroituja varapalvelinjärjestelyjä, koska näistä seuraa säästöä lisenssikustannuksissa.

2 SQL SERVER 2005

2.1 Yleistä

Microsoft SQL Server 2005 on kustannustehokas ja monipuolinen tietojenhallinta- ja -analysointiratkaisu, jonka avulla yrityksen tieto- ja analysointisovellusten turvallisuus, skaalautuvuus ja käytettävyys ovat aiempaa parempia ja näiden sovellusten luonti, käyttöönotto ja hallinta ovat entistä helpompia. SQL Server 2005 auttaa yritystoiminnan kolmella keskeisellä osa-alueella: yritystietojen hallinnassa, kehittäjien tuottavuudessa ja liiketoimintalogiikassa. [1.]

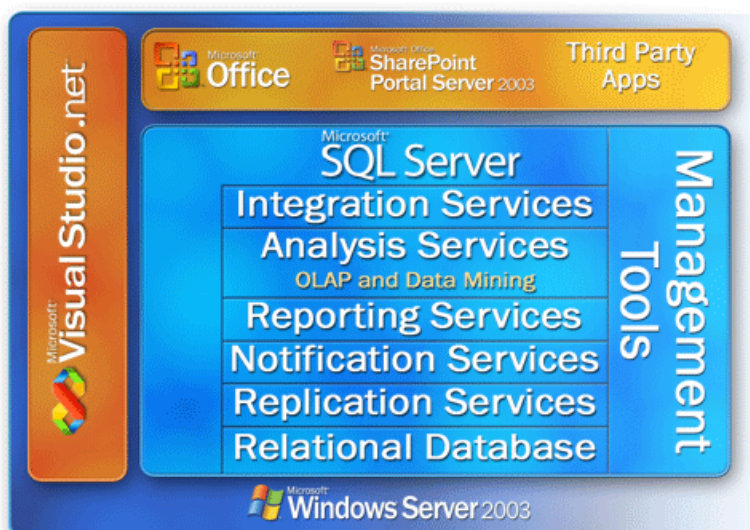
Yritystietojen hallinta - nykyisessä verkottuneessa maailmassa tietojen ja niitä hallitsevien järjestelmien on oltava aina käyttäjien käytettävissä. Kun SQL Server 2005 on käytössä, käyttäjät ja IT-ammattilaiset kaikkialla yrityksessä hyötyvät seisokkiaikojen vähentymisestä, skaalautuvuuden ja suoritusnopeuden parantumisesta sekä tehokkaista suojauksista.

Kehittäjien tuottavuus - yksi keskeisimpiä kehittäjien tuottavuuden tehostamisen esteitä on ollut se, että tietokantojen kehitys- ja virheenetsintätyössä ei ole aikaisemmin ollut käytettävissä integroituja työkaluja. SQL Server 2005:n uudet ominaisuudet mullistavat tietokantasovellusten kehityksen ja käyttöönoton.

Business Intelligence liiketoimintalogiikan haasteet ja mahdollisuudet perustuvat siihen, että työntekijöillä on käytettävissään oikeat tiedot oikealla hetkellä. Jotta tämä päämäärä saavutettaisiin, Business Intelligence ratkaisun on oltava kattava, turvallinen, integroitu operatiivisiin järjestelmiin ja käytettävissä jokaisen päivän jokaisena hetkenä. SQL Server 2005 auttaa yrityksiä saavuttamaan tämän päämäärän.

SQL Server 2005 integroituu mainiosti yhteen uusien Visual Studio 2005 - ja BizTalk Server 2006 -tuotteiden, kuten myös Microsoftin liiketoimintasovellusten Axaptan, CRM:n ja Navisionin kanssa.

Kuvassa 1 oleva kaavio valaisee ytimen osia SQL Server 2005:ssä esittäen, miten SQL Server 2005 on keskeinen osa Windowsia Server järjestelmää integroitumisessa Microsoft Windowsin. [2.]



Kuva 1. SQL Server 2005:n ydinkomponentit

SQL Server data-alusta käsittää seuraavat työkalut [1.]:

Relaatiokanta (Relational Database) on varmempi, luotettavampi, skaalautuvampi, hyvin saatavissa oleva relaatiokannan moottori parantuneen suorituksen kanssa ja tuki jäsenytyneelle ja suunnittelemattomalle datalle.

Kahdentumispalvelut (Replication Services): Datan kahdentuminen levintä tai liikkuvia tietotekniikkasovelluksia varten.

Ilmoituspalvelut (Notification Services): Kehittyneet ilmoituskyvyt skaalautuvien sovelluksien kehitystä ja sijoittamista varten, jotka voivat luovuttaa yksilölliset, ajankohtaiset tiedon päivitykset useille toisiinsa liitettyihin ja liikkuviin laitteisiin.

Yhdentymispalvelut (Integration Services): Datanpoisto, muutos ja lastauskyvyt tietovarastointia ja yritystä varten laaja datayhdentyminen.

Analyysipalvelut (Analysis Services): Kytkeytyvät analyttiset käsittelykyvyt suurien ja monimutkaisten datojen nopeaa ja pitkälle kehitettyä analyysiä varten, jotka käyttävät moniulotteista varastointia.

Raportointi palvelut (Reporting Services): Kokonaisvaltainen ratkaisu perinteisten paperi raporttien ja web-pohjaisten raporttien luomiseen, hallitsemiseen ja esittämiseen.

Hallinta työkalut (Management Tools): SQL Server käsittää integroitua hallinta työkaluja kehittyneelle tietokannan hallinnalle ja viritykselle sekä eheyttämiseksi työkaluilla kuten Microsoft Operations Manager ja Microsoft Systems Management Server.

2.2 Lisensiointivaihtoehdot

Luku 2.2 perustuu kokonaisuudessaan Microsoftin SQL Server –sivustoon [3].

SQL Server 2005:n lisenssimalleja on pääosin kaksi erilaista: palvelinkohtainen ja suoritinkohtainen.

Palvelinkohtainen käyttöoikeus (Per Server) + laitekohtainen (Device CAL) tai käyttäjäkohtainen (User CAL) työaseman palvelinkäyttöoikeus. Tällöin tarvitaan SQL Server 2005 -käyttöoikeus jokaiselle instanssille, jossa SQL

Server 2005 -ohjelmistoa ajetaan. Tämän lisäksi on joko laite- tai käyttäjäkohtainen työaseman palvelinkäyttöoikeus (CAL= Client Access License) jokaiselle käyttäjälle tai laitteelle, jolta palvelinohjelmistoa hyödynnetään. Instanssi tarkoittaa käyttöjärjestelmäympäristöä, jossa sitä suoritetaan. Yhteen instanssiin voidaan asentaa monta SQL Server 2005 -tietokantainstanssia. Lisäksi instanssi voi olla virtuaalinen palvelin tai fyysinen palvelin.

Kun SQL Server 2005 lisensoidaan palvelinkohtaisesti, on otettava huomioon, että palvelinkäyttöoikeus tarvitaan jokaiselle, jossa SQL Server 2005 -versiota tai jotakin sen komponenteista käytetään, koska SQL Server 2005 voidaan asentaa samaan palvelimeen rinnakkain. Palvelimessa olevien suorittimien määrällä ei ole merkitystä tässä lisenssissä.

Laitekohtainen käyttöoikeus (Device CAL) tarvitaan jokaiselle laitteelle, joka käyttää SQL Server 2005 -ohjelmiston palveluja tai toiminnallisuutta. Laitekohtaisen käyttöoikeuden myötä samaa laitetta voi käyttää useampi käyttäjä yhdellä käyttöoikeudella ja yhdellä käyttöoikeudella voi myös olla yhteydessä moneen SQL Serveriin. Kaksi SQL Serveriä voi kuitenkin käyttää toistensa palveluita ilman laitekohtaista käyttöoikeutta.

Käyttäjakohtainen käyttöoikeus (User CAL) tarvitaan käyttäjälle (työntekijä, asiakas tms.), joka käyttää SQL Serverin palveluita tai toiminnallisuutta. Käyttäjä voi käyttää yhdellä käyttöoikeudella useita laitteita ja useita SQL Servereita.

Työaseman palvelinkäyttöoikeus (CAL) tarvitaan vaikka käyttäjä ei suoranaisesti olisi yhteydessä tietokantaan, vaan käyttää palveluita ja ominaisuuksia toisen käyttöliittymän tai sovelluksen kautta. CAL:ia ei tarvita silloin, kun tieto siirretään manuaalisesti, esimerkiksi työntekijä välittää toiselle työntekijälle SQL Serveristä peräisin olevaa tietoa Excel-muodossa.

Työaseman palvelinkäyttöoikeus tarvitaan myös, mikäli SQL:n Personal Edition, Express Edition, MSDE tai Windows CE Edition ottaa yhteyden SQL-palvelimeen.

SQL Server 2005 Workgroup -versiolle on oma SQL Server Workgroup CAL, joka ei riitä SQL Server Standard- tai Enterprise-versioiden käyttöön. Jälkimmäisille tarkoitettu kokonainen SQL CAL kuitenkin käy Workgroup-version käyttöön.

Mikäli SQL Server 2005 -ohjelmistoa käytetään Internet-pohjaisen sovelluksen tietokantana, palvelinperustaista käyttöoikeusmallia ei voida käyttää, vaan SQL lisensoidaan aina suoritinpohjaista mallia käyttäen.

Suoritinkohtaiseen käyttöoikeuteen (Per Processor). Tällöin tarvitaan käyttöoikeus jokaiselle suorittimelle instanssissa, jossa SQL Server 2005 ohjelmistoa ajetaan tai jotakin sen komponenteista käytetään. Suoritinpohjainen käyttöoikeus sisältää käyttöoikeuden rajoittamattomalle käyttäjä- ja laitemäärälle ja käyttäjien määrää ei ole rajattu eikä jaettu sisäisiin tai ulkoisiin. SQL Server -ohjelmiston lisensointiin ei ole tarjolla External Connector -vaihtoehtoa. External Connector -käyttöoikeus ei kata yrityksen omia työntekijöitä, vaan heille hankitaan erilliset CAL-lisenssit.

Moniytimiset (multicore) ja monisäikeiset (hyperthreading) suorittimet laskeetaan yhdeksi suorittimeksi. Jos palvelinlaitteessa on ominaisuudet, joilla se voidaan partitioida laitteistotasolla erillisiin partitioihin, niitä käsitellään kuten fyysisesti erillisiä palvelimia.

2.3 Versiot

Workgroup on helppokäyttöisin ja edullisin tietokantaratkaisu pienille yrityksille tai pienille sovelluksille: 2 CPU, RAM 3 GB. [4.]

Standard on kokonainen tiedonhallinta- ja analysointialusta keskikokoisille organisaatioille tai suurille osastoille: 4 CPU, RAM ei ole rajoitettu, 64-bit. [4.]

Enterprise on integroitu tiedonhallinta- ja analyysialusta liiketoiminnallisesti kriittisiin tai suuriin ympäristöihin: CPU ei ole rajoitettu, RAM ei ole rajoitettu, 64-bit. [4.]

2.4 SQL Server 2005:n alusta

Kaksi palvelinta voidaan konfiguroida siten, että toisen palvelimen vikaantuneessa toinen jatkaa vikaantuneen palvelimen tehtäviä. Passiiviseen varakoneeseen ei tarvita erillistä käyttöoikeutta, kun

1. passiivikone ei aktiivisesti osallistu varsinaiseen prosessiin, vaan on vain tiedon vastaanottajana.

2. passiivikoneessa on sama tai pienempi määrä prosessoreita kuin aktiivikoneessa .

3. passiivikone on aktiivikoneen roolissa maksimissaan 30 vrk.

Edellä mainittuja aktiivi-passiivikokoonpanoja voivat olla esimerkiksi:

1. Tietokannan peilaus (Database Mirroring) on uutta SQL Server 2005 -ohjelmiston myötä. Tapahtumat suoritetaan automaattisesti sekä varsinaisessa että peilikoneessa. Molemmat koneet ovat jatkuvasti ajan tasalla, ja vikatilanteessa palvelu voidaan siirtää toiselle koneelle. Mahdollinen Standard- ja Enterprise-versioissa. [3.]

2. Klusterointi (Failover Clustering) on prosessi, jossa SQL Server ja käyttöjärjestelmä yhdessä varautuvat mahdolliseen vikatilanteeseen. Tietokanta-, laitteisto- tai käyttöjärjestelmävirheen sattuessa resurssit siirretään toiselle, vastaavalla tavalla konfiguroidulle palvelimelle. Mahdollinen Standard- ja Enterprise-versioissa. [3.]

3. Lokisiirto (Log Shipping) tapahtuu siten, että tietokannan tapahtumat kopioidaan varalla olevaan palvelimeen ajantasaisesti. Näin varapalvelimesta tulee identtinen kopio ensisijaisesta palvelimesta. Vikatilanteessa toissijainen palvelin voi lennosta ottaa tehtävät hoitoonsa. Mahdollinen Standard-, Enterprise- ja Workgroup -versioissa. [3.]

Usean SQL Server -ohjelmistokopion asentaminen samalle palvelimelle (multi-instance)

SQL Server 2005 Enterprise, Standard ja Workgroup-versiot mahdollistavat rajoittamattoman määrän tietokantainstansseja samassa virtuaali- tai fyysisessä operointiympäristössä. Aikaisemmin tämä oli mahdollista vain Enterprise -versiossa. [5.]

Runtime-lisenssit (Royalty-sopimus)

Joihinkin Microsoftin tuotteisiin sekä moniin kolmansien osapuolien ohjelmitoihin liittyy SQL Server -tuote. Tällainen toisen tuotteen mukana toimitettava SQL Server on tarkoitettu yksinomaan kyseisen sovelluksen käyttöön eikä sitä voi hyödyntää muuhun tietokantakäyttöön.

Palvelinasennusperustaisena tällainen SQL Server -tuote vaatii työaseman palvelinkäyttöoikeudet (CAL) samalla tavoin kuin erillisenä tuotteena hankittava SQL Server -ohjelmisto.

Kehitystyökaluina (Development Tools) SQL Server tarjoaa integroituja kehitystyökaluja tietokannan moottorille, datanpoistolle, muutokselle ja lastaukselle, joka on yhdistetty tiukasti Microsoft Visual Studioon toimittamaan päästä päähän sovelluksen kehityskykyä.

3 TIETOKANTA

Tietokanta on tietotekniikassa käytetty termi tietovarastolle. Se on kokoelma tietoja, joilla on yhteys toisiinsa. Tietokannan ei välttämättä tarvitse olla sähköisessä muodossa, vaan sellaista voidaan pitää esimerkiksi kynällä ja paperilla.

Tietokanta saattaa edustaa jotain selkeästi rajattua kohdetta reaali maailmasta. Tällainen kohde voi olla esimerkiksi yrityksen keräämät tiedot asiakkaisistaan. Jotta tietokanta olisi toimiva, on sen osien välillä oltava looginen yhteys.

Tietokantojen koot voivat vaihdella suuresti yhteen tiedostoon tallennetuista taulukoista hyvin suuriin tietokantoihin, joissa on useita miljoonia tietueita lukuisista kiintolevyistä koostuvilla levyakoilla. Tietokantaan voidaan tallentaa eri formaateissa olevaa tietoa, kuten tekstiä, ääntä ja videokuvaa.

3.1 Tietokantojen historiaa

Ensimmäiset nykyisten tietokantojen edeltäjät kehitettiin 60-luvulla. Alan pioneeri oli Charles Bachman. Relaatiotietokantamallin ja käsittelyteorian kehitti Edgar F. Codd vuonna 1970. Codd työskenteli IBM:n tutkimuslaboratoriossa, ja hän johti ensimmäisten relaatiotietokantojen testiversioiden kehitystyötä. Vuosina 1973 - 1976 IBM:n laboratoriossa kehitettiin relaatiotietokannan prototyyppi nimeltään System R. IBM:n kaupallinen tietokantatuote DB2 julkaistiin vuonna 1982. [5.]

Hyvin pian kuitenkin sen potentiaali muissakin tietokantasovelluksissa huomattiin. Sen jälkeen siitä on tullut lähes standardi tietokantojen ylläpidossa. SQL ei ole rakenteellinen kieli, joten sen avulla tehtävät voidaan määritellä

kysymyksellä "mitä" ennemminkin kuin "miten". Esimerkiksi SQL määrittelee, mitkä tiedot haetaan tai lisätään, eikä miten toiminto tapahtuu. [6.]

3.2 Tietokannat käytännössä

Tietokantoja käytetään monissa sovelluksissa liki jokaisella sovellusalueella. Tietokantoja suositellaan suurien, monen käyttäjän järjestelmien tiedon tallennusmekanismiksi silloin, kun tarvitaan koordinaatiota usean käyttäjän välillä. Jopa yksittäiselle käyttäjälle tietokanta voi olla sopiva ratkaisu, ja monet sähköpostiohjelmat ja kalenterisovellukset perustuvat standardinmukaiseen tietokantateknologiaan. [6.]

Tietokannat on käytännöllisintä luokitella tuetun ohjelmointimallin mukaan. Muutamia malleista ovat olleet laajalti käytössä jo jonkin aikaa. Hierarkkinen malli toteutettiin ensimmäisenä, jota seurasi verkkomalli. Sen jälkeen relaatiomalli ohitti ne niin kutsutun "Flat-File"-mallin kanssa, joka oli helppo toteuttaa vaatimattomiinkin alustoihin. Hierarkkinen-, verkko- ja Flat -tiedostomallit eivät perustu vahvaan teoreettiseen pohjaan niin kuin relaatiomalli, vaan ne ovat syntyneet laitteiston ja ohjelmointitekniikoiden rajoitteiden vaikutuksesta.

Olio-ohjelmoinnin yleistyttyä myös tietokantoja on kehitetty vastaamaan oliomallin asettamiin tarpeisiin. On kehitetty puhtaita oliotietokantoja ja perinteisiin relaatiokantoihin on lisätty olio-ohjelmointia tukevia ominaisuuksia. Jälkimmäisistä käytetään nimeä olio-relaatiotietokannat.

SQL on standardi, kun käsitellään ja haetaan tietoja relaatiotietokannasta. Kaikki keskeiset relaatiotietokantoja hyväksyvät käyttävät ratkaisut tukevat ja hyödyntävät sitä jollain tapaa. SQL:aa voi hyödyntää niin yhden käyttäjän Windows-sovelluksessa, joka käyttää MS Access -tietokantaa, kuin 5 000 käyttäjän SQL Server -ohjelmassa. Sen avulla ohjelmoija tai tietokannan hoitaja voi mm. muokata tietokannan rakennetta, muuttaa järjestelmän turva-asetuksia sekä lisätä käyttäjän oikeuksia tietokantaan, suorittaa kyselyjä tietokannasta ja päivittää tietokantaa.

Suosittuja SQL-toteutuksia on useita. Perustason ratkaisuksista löytyvät kotija toimistokäyttöön suunnatut tietokantasovellukset, kuten MS Access tai Corel Paradox. Nämä sisältävät usein helppokäyttöisen graafisen käyttöliittymän sekä ison joukon velhoja, jotka helpottavat tietokannan ylläpitoa ja hallintaa.

Microsoft tarjoaa useita kehittäjille ja yrityskäyttöön suunnattuja SQL-työkaluja ja -ohjelmia. Näistä tunnetuin lienee yrityksen lippulaiva, MS SQL Server, joka on jo useana vuotena peräkkäin äänestetty parhaaksi RDBMS-ratkaisuksi.

Tarjolla on toki useita muitakin ammattimaiseen käyttöön suunnattuja tietokantaratkaisuja. Näistä kaikkein eniten mainetta ovat saavuttaneet Borlandin FireBird sekä ilmainen MySQL.

3.3 Transaktiot ja samanaikaisuus

Lisänä useimmat käytännön tietokannat pyrkivät toteuttamaan transaktiomallin, jolla on halutut tiedon oikeellisuuden säilytysominaisuudet. Ihanneta-pauksessa tietokantaohjelmisto noudattaa ACID-sääntöjä.

Atomisuus (Atomicity) - kaikki tai ei mitään. Jokainen operaatio pitää suorittaa tai sitten ei niistä suoriteta ainuttakaan. Transaktiot, joita ei pystytä vie-mään loppuun, täytyy peruuttaa kokonaisuudessaan.

Oikeellisuus (Consistency) - jokaisen transaktion jäljiltä tietokannan tulee olla oikeellisessa tilassa.

Eristys (Isolation) - transaktiot eivät saa vaikuttaa toisiinsa eikä keskeneräinen suoritus saa näkyä muille transaktioille.

Kestävyys (Durability) - onnistuneiden transaktioiden pitää säilyä, jos ohjelmisto kaatuu.

Käytännön toteutuksissa useimmat tietokantajärjestelmät joustavat joidenkin näiden sääntöjen noudattamisessa, jotta ne pystyvät saavuttamaan paremman suorituskyvyn.

Samanaikaisuuden hallinta on menetelmä, jolla varmistetaan, että transaktiot ajetaan turvallisella tavalla ACID-sääntöjä noudattaen. Tietokantajärjestelmän pitää pystyä varmistamaan, että vain sarjallistettavat, palautettavat operaatiot ovat sallittuja ja että mitään toimintoja toteutetuista transaktioista ei menetetä silloin, kun palautetaan kanta tilaan ennen peruutettua transaktiota.

3.4 Eri tietokantajärjestelmiä

3.4.1 Yleistä

Seuraavassa on kerrottu muutamista relaatiojärjestelmistä. Kaikissa niissä käytetään SQL-kieltä, mikä on IBM:n kehittämä standardoitu kyselykieli, jolla relaatiotietokantaan voi tehdä erilaisia hakuja, muutoksia ja lisäyksiä. Käytännössä kaikki relaatiotietokannat ymmärtävät SQL-kieltä. SQL on standardoitu ISO/IEC:n toimesta.

SQL ei määritä verkkoprotokollaa tai standardia, jolla SQL-komentoja välitetään. ODBC tarjoaa jonkinäköisen tietokantariippumattoman ajuriratkaisun, mutta sekin vaatii samaisen ODBC-ajurin jokaiselle asiakkaalle. Sen avulla sovellukset voivat kommunikoida tietokantapalvelimen kanssa.

Tärkeimpiä käyttökäskyjä SQL:ssa ovat mm. SELECT, UPDATE, INSERT ja DELETE. Tärkeimmät ylläpitokäskyt ovat CREATE TABLE, CREATE INDEX, ALTER TABLE ja ALTER INDEX. [7.]

3.4.2 Oracle

Oracle on sekä tunnettu relaatiotietokanta että Oracle tietokantaa valmistava Larry Ellisonin johtama yhtiö. Tarkkaan ottaen yhtiö myy tietokannan hallintaohjelmistoa, mutta normaalissa kielenkäytössä samaa nimeä käytetään sekä kannasta että ohjelmistosta. Oracle on ammattilaisten tietokantatyökalu ja kuuluu samaan sarjaan kuin MS SQL Server. [8.]

Oracle tallettaa datan loogisesti taulualueisiin ja fyysisesti tiedostoihin. Taulualueilla voi olla erilaisia lohkoja esimerkiksi indeksilohkoja ja datalohkoja. Segmentit puolestaan koostuvat ilmenemistä. Ilmenemät ryhmitellään yhtenevien datalohkojen perusteella. Datalohko on tiedon talletuksen perusyksikkö. Fyysisellä tasolla tiedostot muodostuvat yhdestä tai useammasta datalohkosta, jossa lohkokoko voi vaihdella.

Oracle hallinnoi kaikkea talletettua tietoa sen tiedon avulla, joka on talletettu systeemitaulualueelle.

Tietokantaan voi tallentaa myös funktioita, ja ne voidaan suorittaa tietokantaohjelmiston sisällä. Funktiot kirjoitetaan SQL-, PL/SQL- tai Java-ohjelmointikielellä.

3.4.3 MySQL

MySQL-tietokannan loivat vuonna 1995 suomalainen Michael Widenius ja ruotsalaisen David Axmarkin. MySQL:n ensimmäinen versio julkaistiin 1996. [9.]

MySQL on suosittu ja yksinkertaisissa tehtävissä melko tehokas SQL-tietokannan hallintajärjestelmä. MySQL:ää kehittää ruotsalainen yritys MySQL AB. MySQL on saatavissa vapaalla GNU GPL:llä tai kaupallisella lisenssillä.

Monista kaupallisista tietokantajärjestelmistä poiketen MySQL:n hallinnointi tapahtuu komentoriviltä tai tekstipohjaisella asiakasohjelmalla. Sille on myös valmistajan sivulta saatavat graafiset MySQL Administrator ja MySQL Query Browser sekä suosittu vaihtoehtoinen phpMyAdmin.

Aikaisempina vuosina MySQL:ää on karsastettu ammattilaispiireissä sen vajavaisuuksien, kuten esimerkiksi transaktioiden sekä aitojen vierasavaimien puutteen takia, mutta nykyään MySQL:ää on rajusti kehitetty eteenpäin ja vajavaisuudet ovat vähentyneet.

MySQL-tietokanta on hyvin suosittu web-palveluiden tietokantana. MySQL-tietokannan päälle rakennettava ohjelmalogiikka tehdään usein PHP, Python- tai Perl-ohjelmointikielellä. Myös muilla ohjelmointikielillä on mahdollista käyttää MySQL -tietokantaa. MySQL sisältää rajapinnan muun muassa C:lle, C++:lle, Smalltalkille, Javalle ja Rubyille. MySQL:lle on olemassa MyODBC-niminen ODBC-rajapinta.

3.4.4 PostgreSQL

PostgreSQL on avoimena lähdekoodina jaettava olio-relaatiotietokantapalvelin, joka on lisensoitu joustavalla BSD-tyyppisellä lisenssillä. Se on vaihtoehto muille vapaan lähdekoodin tietokantajärjestelmille, kuten MySQL:lle, sekä myös kaupallisille järjestelmille kuten Oraclelle tai Microsoft SQL Serverille. Kuten toiset vastaavat avoimen lähdekoodin projektit, PostgreSQL ei ole yksittäisen yrityksen tai henkilön kontrolloima, vaan perustuu kansainväliseen ohjelmoijien ja yritysten muodostaman yhteisön tekemään kehitystyöhön. Tätä kehitystyötä johtaa ns. core-team, jolla on valta päättää, otetaanko muutokset mukaan koodipuuhan vai ei. [10.]

PostgreSQL on tullut suuresti tunnetuksi MySQL:n varjossa. Kun MySQL on tullut tunnetuksi helppona ja nopeana tietokantana, PostgreSQL on tullut tunnetuksi pienillä tietomäärillä hieman hitaampana mutta kuitenkin ominaisuuksiltaan parempana ja luotettavampana tietokantana. Toistaiseksi se katkaakin suuremman osan SQL-2003-standardista kuin MySQL. PostgreSQL tukee mm. transaktioita ja viittauseheyden tarkistusta. PostgreSQL:n suuria hienouksia edustaa tyyppien määrittely: esimerkiksi datan muunto oliomallista relaatiomalliin ja takaisin on suuri ohjelmointitekniinen ongelma, mutta PostgreSQL pystyy tarjoamaan mahdollisuuden rakentaa uusia monimutkaisia tietotyyppejä. Version 8.0 myötä myös replikointiominaisuuksiin on panostettu suuresti, joten useita palvelimia käyttämällä PostgreSQL voi selvitä hyvinkin suuresta käyttökuormasta. [10.]

Yhteistä kaikille tässä mainituille tietokantaratkaisuille on, että niitä kaikkia voidaan hallita SQL:n avulla. Osa on kehittänyt omia murteitaan tehokkuuden parantamiseksi, mutta pohjimmiltaan lähes kaikki tietokantaratkaisut ovat ANSI-SQL yhteensopivia. Tämä tarkoittaa, että meillä on aina käytössä tietty komentojen perusjoukko, joka toimii kaikissa tietokannoissa. [10.]

4 ARKISTOSOVELLUKSEN SIIRTO KESKITETTYYN SQL SERVER 2005 - TIETOKANTARATKAISUUN

Arkistosovelluksessa on keskus- ja sopimusarkisto. Sovelluksella on käyttäjiä noin sataviisikymmentä. Tietokannassa on metatietoa sopimuksista ja linkki skannattuihin sopimuksiin. Metatietoja ovat mm. sopimuksennimi, asiakirjatyypit, yrityksen nimi, vastuuhenkilöt, muistutus vanhenemisesta, voimassaolo alkaa ja päättyy, sopimusnumero, projekti- ja hankenumero sekä lisätietoja.

Lähtökohtana on että Arkisto-sovellus ja arkiston tietokanta ovat samalla palvelimella. Lähtötilanteessa tietokannan versio on SQLServer 2000. Ongelmana tässä ratkaisussa on, että sovellus ja tietokanta kilpailevat palvelimen muistista, ja tämä ilmenee sovellusta käytettäessä mm. pidentyneinä hakuajoina. Halusimme myös eroon vanhentuneista SQL Server 2000:sta, koska halusimme vaihtaa ne SQL Server 2005:ksi. Tätä ennen luodaan keskitetty tietokantapalvelin, jonne on tarkoitus siirtää yksittäiset tietokannat.

Tämän seurauksena kysyimme sovellustoimittajalta, mitä muutoksia sovelluksen ohjaustiedostoihin tehdään, mikäli tietokanta siirretään omalle palvelimelle keskitettyyn tietokantaratkaisuun. Käytössä ei ole Windows-autentikointia vaan sopimus- ja keskusarkisto käyttävät tietokantaa JDBC-yhteyden läpi ja yhteysosoitteessa annetaan parametrina instanssin nimi. Default instanssia käytettäessä olisi tietokantaan tehtävä muutoksia. Veikkauksessa kaikki ympäristöt on kahdennettu siten, että varaympäristöä voidaan käyttää testipalvelimena. Sama koskee myös keskitettyjä tietokantapalvelimia eli niilläkin on varaympäristö. Sovellustoimittajan kanssa sovittiin, että testaamme siirtoa ensin testipuolella. Kun kaikki toimii hyvin, tehdään sama tuotannossa. Sovimme, että toimittajan konsultti tekee sovelluksen puolelle tarvittavat muutokset ja me siirrämme tietokannan SQLServer2005:een. Koska toimittaja ei luvannut tukea sovellukselle 2005:ssä niin asensimme tietokannan serverille SQL Server 2000 -tilaan. Otimme ensin testipalvelimella varmistuksen tietokannasta ja tallensimme sen meidän levyjärjestelmään ja sitten noudimme tietokannan varmistuksen keskitetylle tietokantapalvelimelle ja teimme uuden tietokannan ja palautimme tietokantavarmistuksen sen päälle. Yllätyksenä tuli, että tietokannan ja tietokantapalvelimen sisäisenkäyttäjän kirjautuminen täytyi muuttaa eri tavalla kuin mitä oli SQL Server 2000:ssa. Tämä herätti alkuun ihmetystä, mutta se ratkaistiin myöhemmin kuvatulla tavalla.

Tämän jälkeen muutetaan sovelluksen ohjauksen uuteen keskitettyyn tietokantapalvelimeen ja päästään testaamaan uutta ratkaisua. Sitten muutettiin tietokantoja valvova Argent-sovellusjärjestelmän ohjaukset keskitettyyn tietokantaan, jotta Argent ei herjaa pysäytetyistä tietokannoista. Argent-sovellus valvoo kantojen ja SQLServerAgentin päälläoloa, jotta varmistukset menevät levyille. Uutta ratkaisua testattiin kaksi viikkoa testiympäristössä ennen kuin siirryttiin tuotantoon. Havaittiin jo testivaiheessa, että uudella ratkaisulla saavutettiin parempaa suorituskykyä sovelluksessa. Tuotantoon siirtymisen jälkeen käyttäjien valitukset ohjelman hitaudesta loppuivat. [11]

Tulevaisuudessa, kun arkistosovelluksen sekä testi- että tuotantopuolella olevat sovelluspalvelimet ovat vanhempia ja virtuaalipalvelimilla on enemmän tilaa virtuaaliympäristössä, tullaan sovelluspalvelimet virtualisoimaan. Tästä seuraa lisenssisäästöjä.

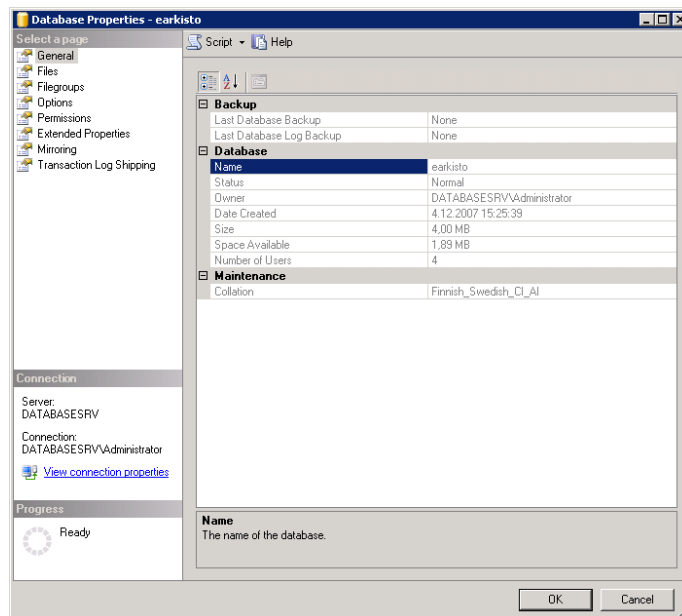
Seuraavaksi poistetaan vanhat tietokannat nykyisiltä sovelluspalvelimilta ja sen jälkeen poistetaan SQL Server 2000:set, koska on varaympäristö keskittystä tietokantapalvelimesta.

5 ARKISTOSOVELLUKSEN SIIRTO

Ilmoitetaan käyttökatkosta käyttäjille, että illalla ja seuraavana aamuna on puoleenpäivään saakka käyttökatkos sekä ilmoitetaan sähköpostilla, kun arkistosovellus on käytettävissä. Otetaan tietokannasta varmistus illalla työajan jälkeen. Siirretään varmistus tietokantapalvelimelle kopiaamalla se levyympäristöltä tai hakemalla varmistusjärjestelmän kautta.

Kannan asennus tehdään tietokantapalvelimelle paikallisella administrator-tunnuksella. Avataan Microsoft SQL Server Management Studio -ohjelma. Avataan tietokantapalvelin/Databases-haara ja hiiren oikealla painikkeella valitaan New Database. Luodaan uusi kanta nimeltään Arkisto.

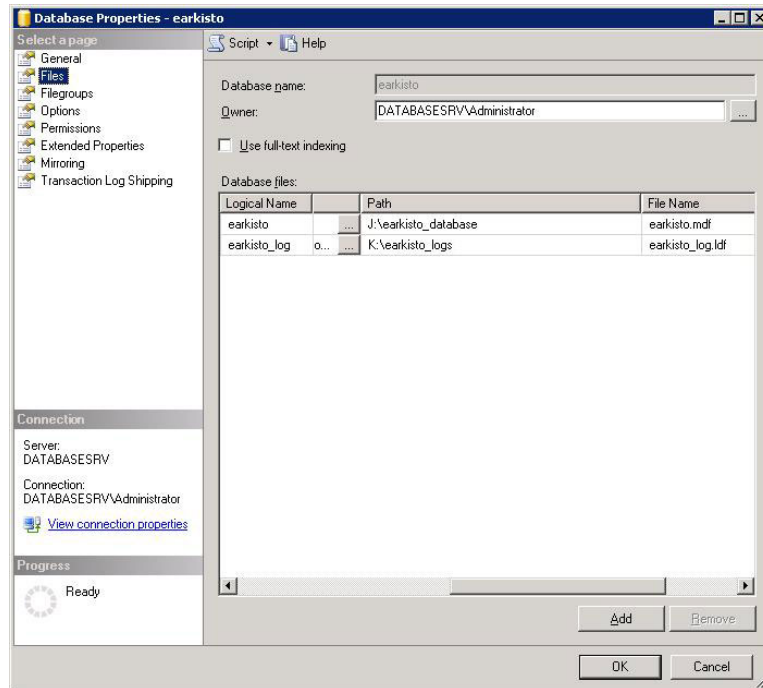
Tietokannan ominaisuuksien yleiseltä välilehdeltä nähdään seuraavia tietoja: viimeisin varmistus, nimi, tila, omistaja, luontipäivä, tila, käytettävissä oleva tila, käyttäjät, tietokannan kielikoodi.



Kuva 2. Tietokannan ominaisuudet -välilehti

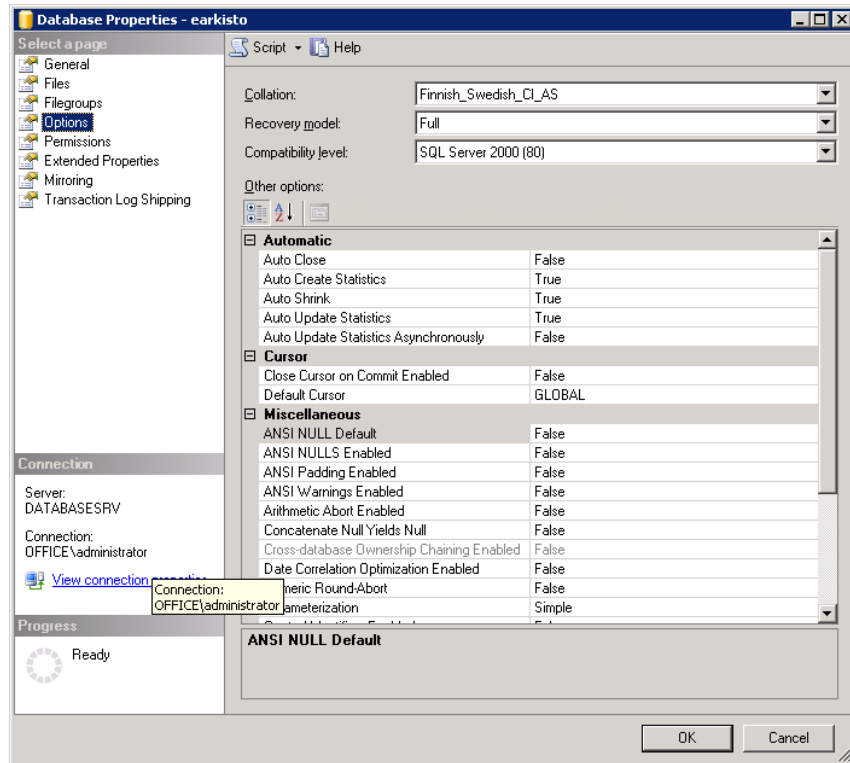
Tietokannan tiedostot välilehdeltä nähdään tiedot kannan ja kannan login sijainnista. Kanta ja logi laitettiin eri levyille, koska näin tietokanta toimii nopeampana, koska levyjärjestelmän toinen lukupää lukee kantaa ja toinen kirjoit-

taa logia. Tärkeintä on, että tietokantaa ja logia ei laiteta C-levylle, koska sitä käytetään pelkästään käyttöjärjestelmää ja sovelluksia varten. Se, että käytetään levyjärjestelmän levyjä kantaa ja logia varten helpottaa niiden laajentamista, ja varmistuksia varten on oma levyasemansa.



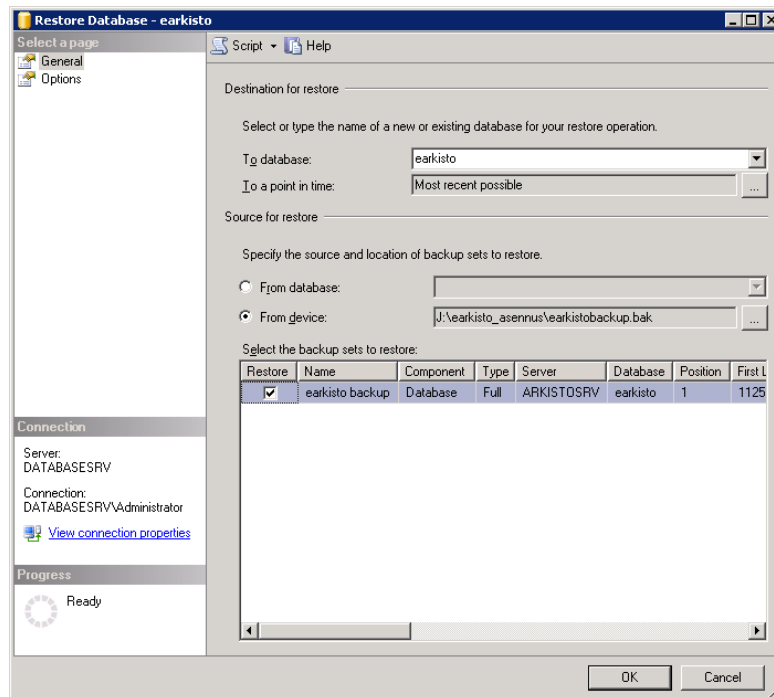
Kuva 3. Kuvassa tietokanta ja logi eri levyillä

Options-välilehdeltä valitaan Compatibility level: SQL Server 2000(80), koska arkiston sovellustoimittaja ei tue vielä uudempaa. Recovery modeliksi valitaan Full, koska silloin voidaan palata tietokannan palautuksessa haluttuun ajanhetkeen. Joissakin tietokannoissa on paljon huonosti tehtyjä kyselyjä tai tietomalli on huonosti suunniteltu. Tästä seuraa, että tietokannan logi voi kasvaa suhteessa valtavaksi verrattuna tietokannan kokoon. Näissä tapauksissa on käytetty recovery model simpleä, mutta sitä käytettäessä ei pystytä palautumaan kuin viimeiseen varmistukseen. Tämän takia varmistuksia täytyy ottaa useammin ja tietokannan pitää olla luonteeltaan sellainen, että tietoa voi hävitä tai se on helposti luotavissa uudelleen esimerkiksi latauksilla. Välilehdeltä valittava kielimoodin on tärkeä olla Finnish_Swedish_CI_AS, koska muuten skandinaavisten kirjainten ja desimaalipilkun kanssa saattaa tulla ongelmia.



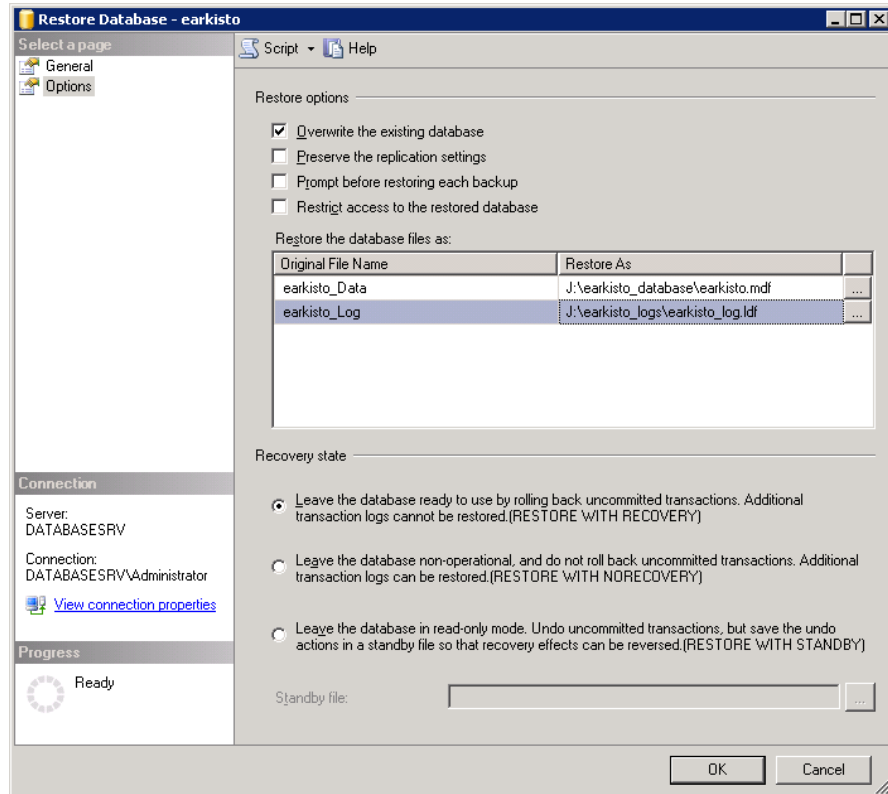
Kuva 4. Tietokannan asetukset

Otetaan arkistopalvelimen kannasta full-backup ja siirretään tietokantapalvelimelle ja palautetaan se restorella aiemmin luodun kannan päälle.



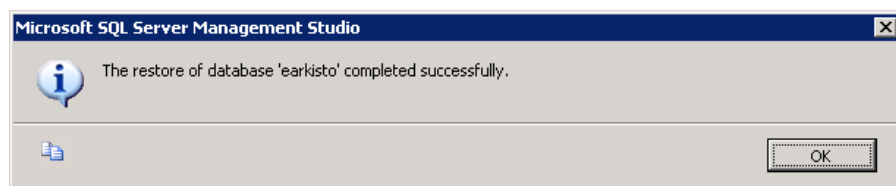
Kuva 5. Tehdään tietokannan palautus

Tässä vaiheessa on tärkeää laittaa ruksi Overwrite existing database, jotta tietokanta ja logi menevät oikeiden tiedostojen paikalle. Mikäli tietokannan ei halua heti kaikille käyttäjille näkyväksi, niin se voidaan laittaa Restrict access to the restored database -tilaan ja tehdä kaikki toimenpiteet valmiiksi ja sen jälkeen vapautetaan tietokanta kaikille näkyväksi.



Kuva 6. Tietokannan palautus

Lopuksi saadaan ilmoitus onnistuneesta palautuksesta.



Kuva 7. Ilmoitus onnistuneesta tietokannan palautuksesta

5.1 Tietokantaan kirjautuminen palautuksen jälkeen

Kanta on palautettu ja se on pystyssä. Tämän jälkeen täytyy tehdä useita toimenpiteitä, jotta SQL Server -autentikoinnin käyttäjätunnus ja salasana samanlaiseksi kuin vanhassa tietokantapalvelimessa. Tietokannan palautuk-

sen mukana tulee tietokannankäyttäjä nimeltään sir, mutta SQL Serveriltä puuttuu kuitenkin samanniminen tietokantapalvelimen käyttäjätunnus, jolla kantayhteys luodaan. Jos tämän tunnuksen luo ja yrittää yhdistää Arkistokannan sir-käyttäjän uuden SQL Serverin käyttäjän tunnukseseen, oletuskannan käyttäjäksi se ei onnistu.

5.1.1 *Aikaisemmassa SQL Server 2000 -versiossa onnistuttiin näin:*

Arkisto MS SQL Server -tietokanta voidaan siirtää toiseen palvelimeen suoraan varmuuskopiotiedostosta. Tämä kuitenkin vaatii tietokannan järjestelmätauluihin muutoksia, koska MS SQL Serverin sisäiset käyttäjätunnukset (Arkisto käyttää sir-tunnusta) on uniikkina tunnisteena (SID) järjestelmäkohmainen.

Järjestelmäkohtaiset käyttäjien SID on tallennettu SQLServerin master tietokantaan. Ne saadaan näkyviin komennolla SQL Query Analyzer:

```
select * from master..syslogins
```

Tämän jälkeen name-sarake kertoo käyttäjätunnuksen nimen ja sid-sarake tarvittavan uniikin avaimen. Ensin kopioidaan privuserin sid leikepöydälle ja tämän jälkeen syötetään seuraava komento (arkisto tietokannan nimi on Earkisto):

```
select * from earkisto..sysusers
```

Tästä nähdään tietokantakohtainen uid jokaiselle järjestelmätason käyttäjälle. Nyt kun on tiedossa privuserin uid ja sid, niin ne pitää enää yhdistää komennolla:

```
update earkisto..sysusers set sid=[paste leikepöydältä tähän] where uid=[privuserin uid]
```

Alla on esimerkkinä Earkisto-nimisestä järjestelmästä vastaavan tilanteen kuvaruutukaappaukset.


```
select sid, name from master..syslogins
```

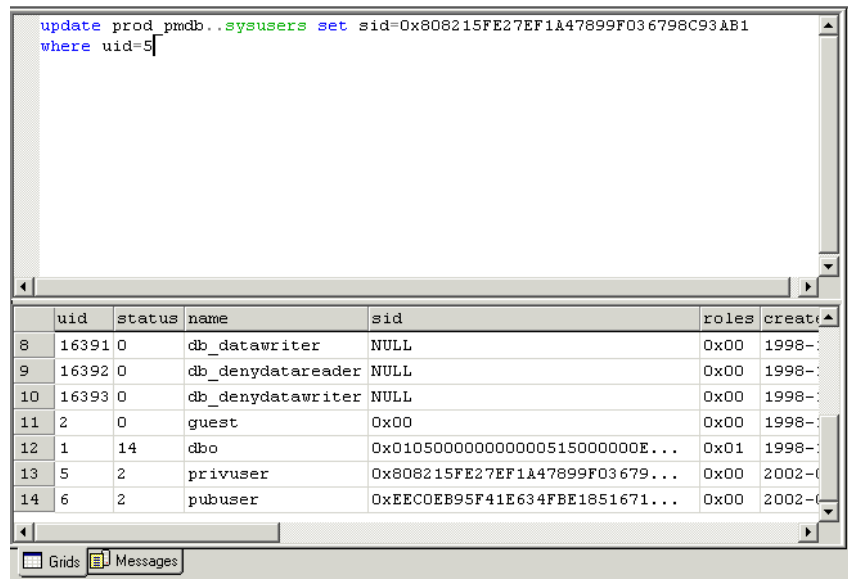
	sid	name
1	Ox01	sa
2	Ox01020000000000052000000020020000	BUILTIN\Administrators
3	Ox010500000000000515000000EE94020...	
4	Ox3B1311FB20B03E4996DCB59F27CF0558	
5	Ox7F09E0AFA300654FA4FAF13F5772002E	
6	Ox808215FE27EF1A47899F036798C93AB1	privuser
7	Ox85442B791B15074C8EC0BBA9D5EFC73D	
8	Ox8F68B548B8F849A4184BF5A493533C	

Kuva 7. Etsitään privuserin oikea SID järjestelmätason taulusta

```
select * from PROD_PMDB..sysusers
```

	uid	status	name	sid	roles	create...
8	16391	0	db_datawriter	NULL	0x00	1998-
9	16392	0	db_denydatareader	NULL	0x00	1998-
10	16393	0	db_denydatawriter	NULL	0x00	1998-
11	2	0	guest	0x00	0x00	1998-
12	1	14	dbo	Ox010500000000000515000000E...	0x01	1998-
13	5	2	privuser	Ox808215FE27EF1A47899F03679...	0x00	2002-
14	6	2	pubuser	OxEE0EB95F41E634FBE1851671...	0x00	2002-

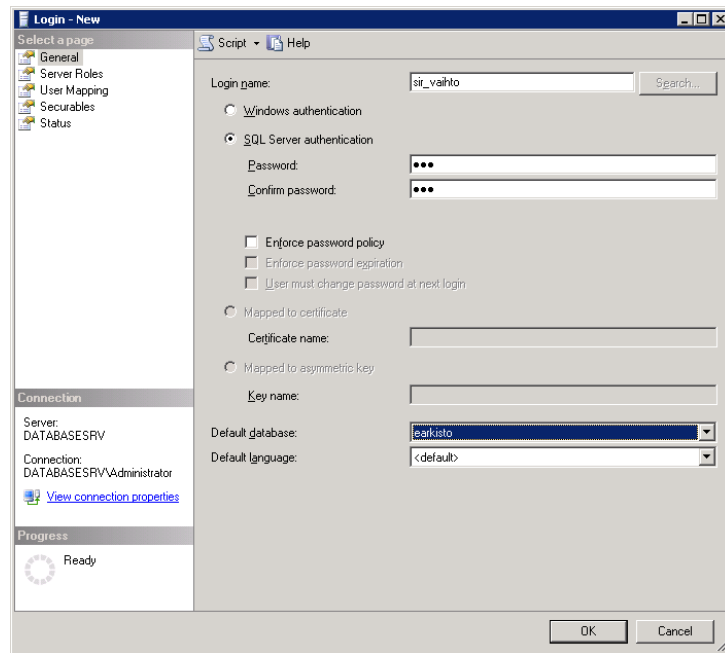
Kuva 8. Etsitään privuserin UID palautettavan tietokannan käyttäjätaulusta



Kuva 9. Korvataan tietokannan virheellinen SID järjestelmätason oikealla SID:llä

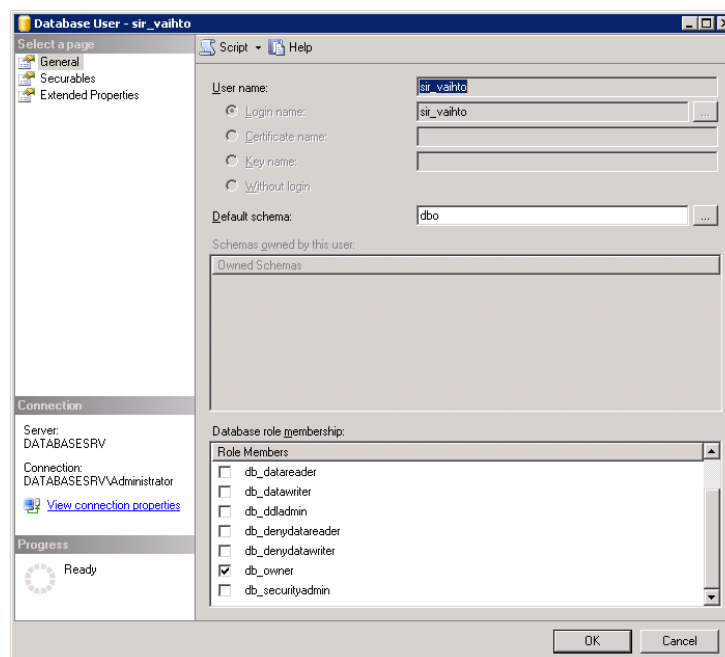
5.1.2 SQL SERVER 2005:ssä tehdään

SQL Server -autentikoinnin käyttäjätunnusta ja salasanaa varten luodaan SQL Serveriin apukäyttäjäksi login sir_vaihto. SQL Server Management Studio /kantapalvelin/Security/Logins -> New Login. Jätetään laittamatta ruki kohtaan Enforce password policy, koska tämän jälkeen jouduttaisiin keksimään hankala salasana ja tässä vaiheessa tunnusta käytetään vain hetki (ja kanta voisi olla restricted tilassa). Muuten tässä voitaisiin vielä määritellä salasanan vanhentumisaika ja se, että käyttäjä joutuu vaihtamaan salasanan ensimmäisellä kirjautumisella. Default database tulee automaattisesti oikein, koska olemme tehneet tunnuksen oikean kannan päällä.



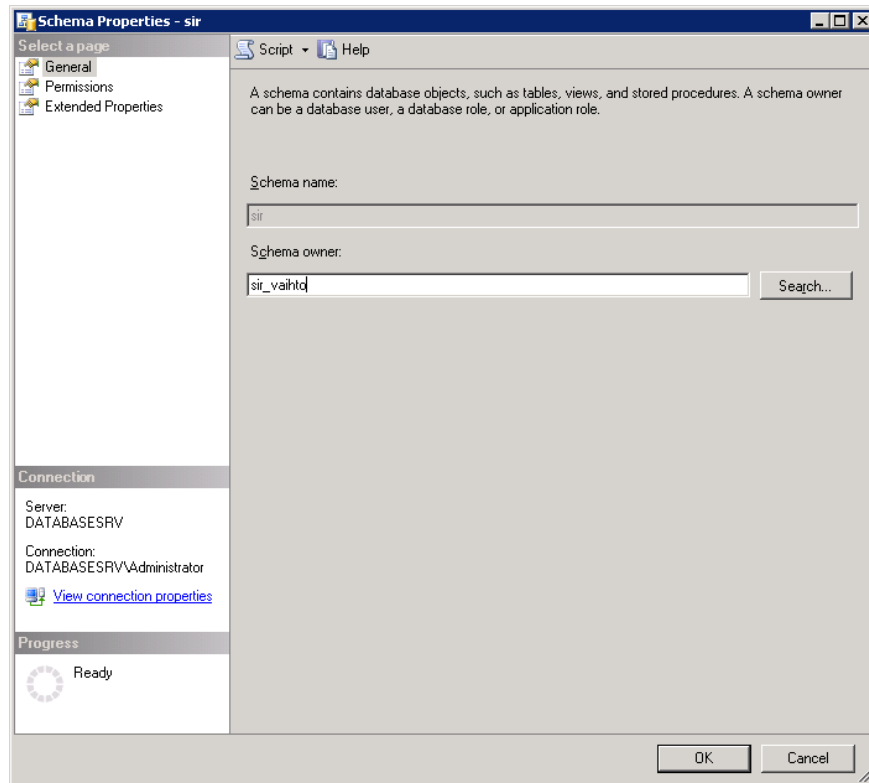
Kuva 10. Luodaan apukäyttäjä nimeltään sir_vaihto

Luodaan kannalle sir_vaihto käyttäjä. SQL Server Management Studio /kantapalvelin/Databases/Arkisto/Security/Users -> New User. Tässä on tärkeää laittaa tunnus tietokannan omistajaksi, koska muuten vaihto ja alkupe-
räisen tunnuksen poistaminen ei onnistu.



Kuva 11. Laitetaan tunnus kannan omistajaksi

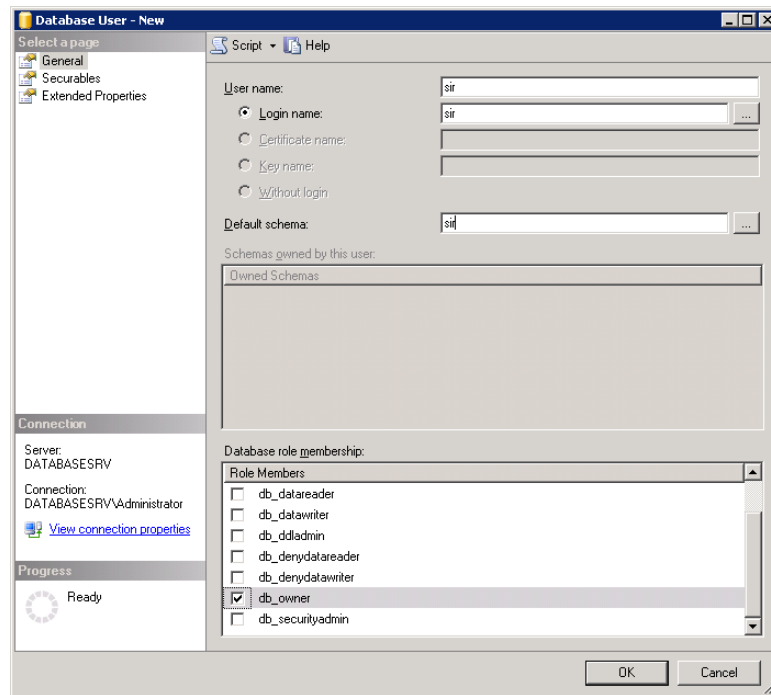
Vaihdetaan sir-schema sir_vaihto tunnuksen omistamaksi. SQL Server Management Studio /kantapalvelin/Databases/Arkisto/Security/Schemas polusta valitaan sir-schema ja valitaan properties.



Kuva 12. Vaihdetaan omistaja

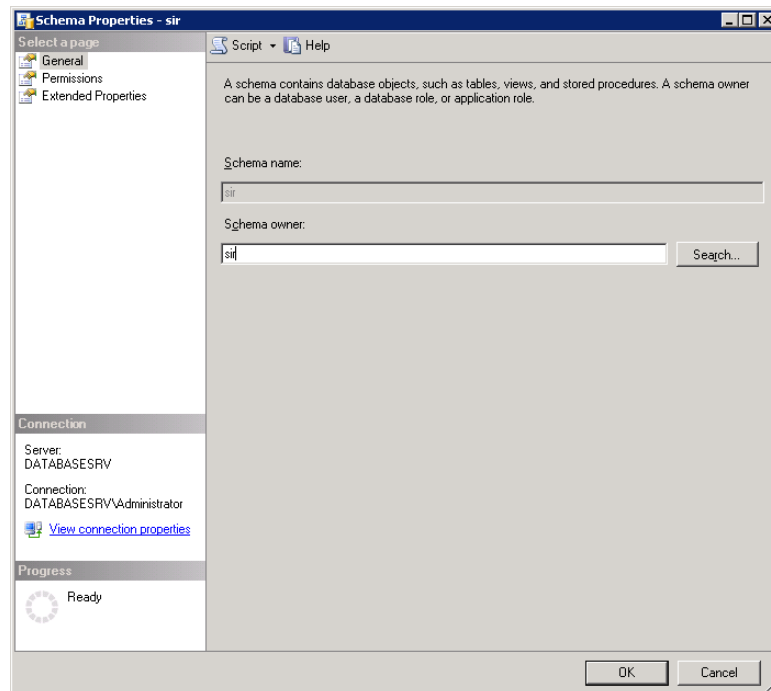
Tämän jälkeen sir-käyttäjän saadaan poistettua kannasta. SQL Server Management Studio /kantapalvelin/Databases/Arkisto/Security/Users-polusta valitaan käyttäjä aktiiviseksi ja poistetaan delete-vaihtoehdolla.

Luodaan sir-käyttäjää uudelleen. Tärkeää on, että login nameksi pitää nyt laittaa sir. Laitetaan rooli db_owner samoin kuin aiemmalla sir-käyttäjällä oli. SQL Server Management Studion polusta /kantapalvelin/Databases/Arkisto/Security/Users valitaan New User.



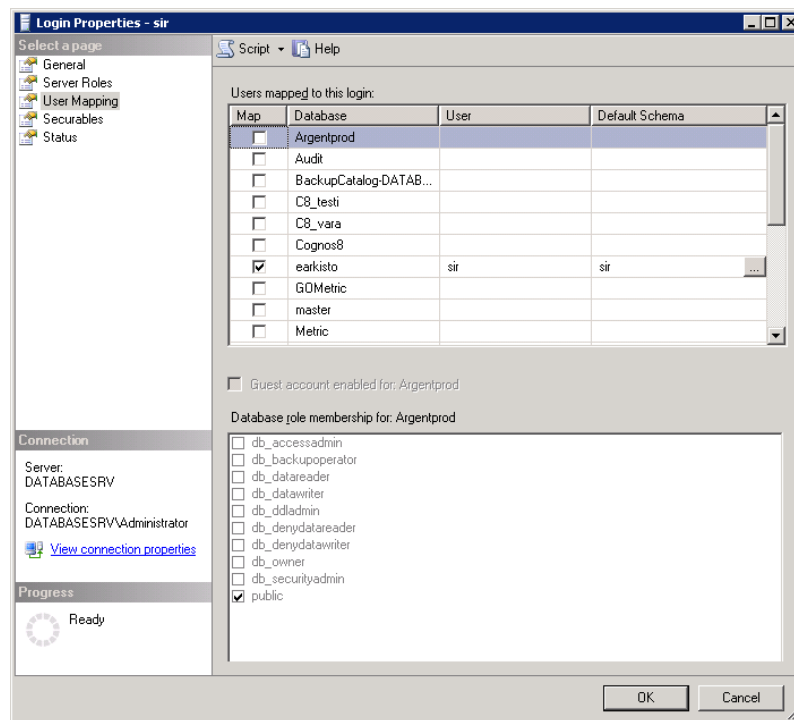
Kuva 13. Luodaan sir-käyttäjä ja laitetaan tietokannan omistajaksi

Vaihdetaan schema takaisin sir-kantakäyttäjän omistamaksi. Polusta SQL Server Management Studio /kantapalvelin/Databases/Arkisto/Security/Schemas valitaan sir-schema ja valitaan properties.



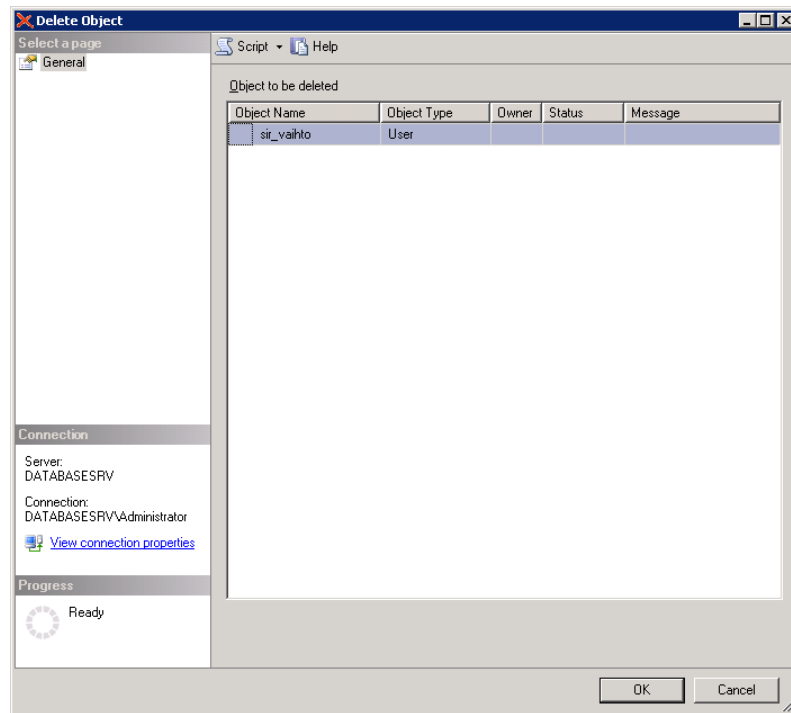
Kuva 14. Vaihdetaan schema omistaja

Heti perään huomataan, että scheman omistajan vaihtaminen ei onnistu scheman ominaisuuksista, vaan pitää se ensin vaihtaa polusta SQL Server Management Studio /kantapalvelin/Security/Logins, josta valitaan sir-käyttäjä ja properties ja sitten välilehti User Mapping ja laitetaan arkisto kantaan ruksi. Tämän jälkeen voidaan vielä tarkistetaa scheman ominaisuuksista, että omistajaksi on vaihtunut sir.



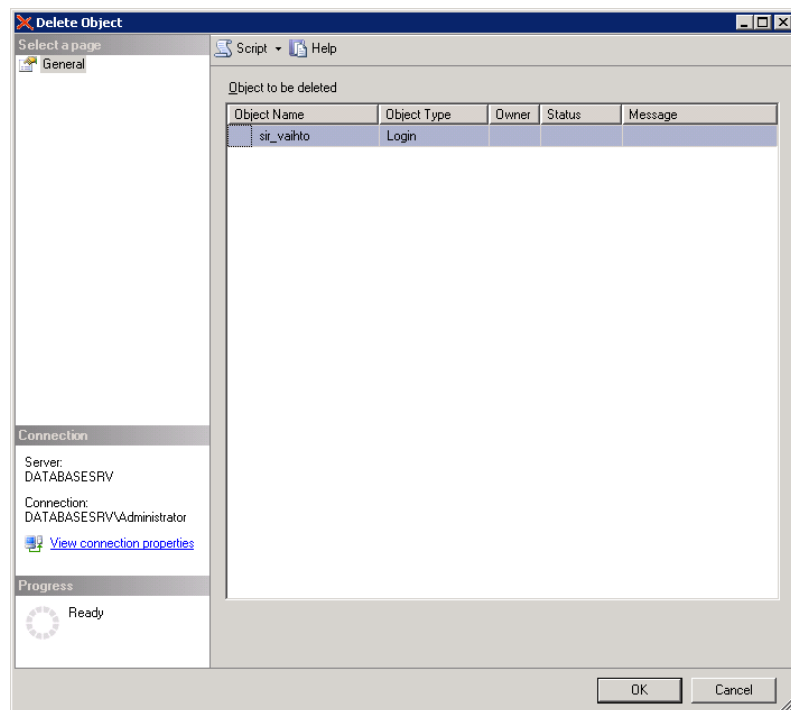
Kuva 15. Vaihdetaan omistaja

Seuraavaksi poistetaan sir_vaihto kannan login.



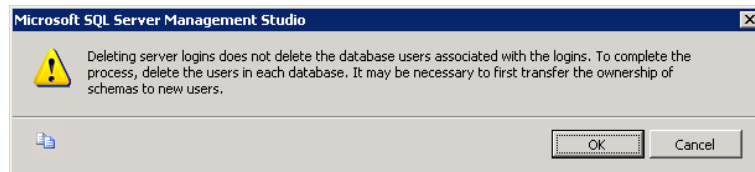
Kuva 16. Sir_vaihto login poistaminen kannasta

Kannasta poistamisen jälkeen poistetaan vielä SQLServeristä sir_vaihto login.



Kuva 17. Sir_vaihto login poistaminen SQLServeristä

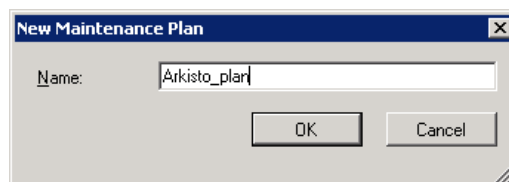
SQLServer antaa vielä varmistusviestin, että aikaisemmat toimenpiteet pitää olla tehtynä ennen poistoa, mutta ne on tehty. Tämän jälkeen tietokantaan saadaan yhteys arkistosta.



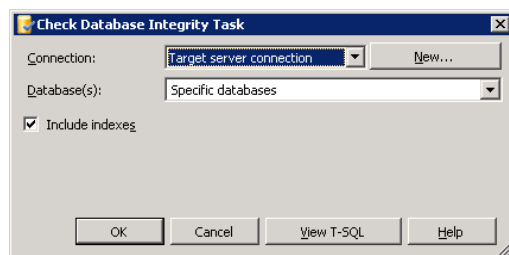
Kuva 18. Varmistusviesti sir_vaihton poistamisesta

5.2 Tietokannan varmistaminen

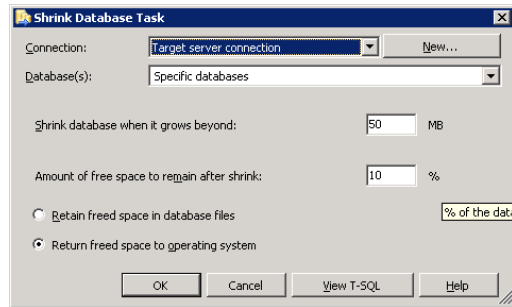
Tietokannan varmistaminen on hieman muuttunut verrattuna aikaisempiin versioihin. Tämän seurauksena dokumentoimme myös tietokannan varmistamisen osana tätä päättötyötä. Tietokannan varmistuksen voi tehdä automaattisella toiminnolla, mutta teimme varmistuksen manuaalisesti raahaamalla hiirellä tarvittavat tehtävät varmistusajoon. SQL Server Management Studiossa valitaan polku /kantapalvelin/Management/Maintenance Plans ja hiiren oikealla painikkeella New Maintenance Plan. Kuvista 19 - 25 nähdään, miten tietokannan varmistaminen tehdään.



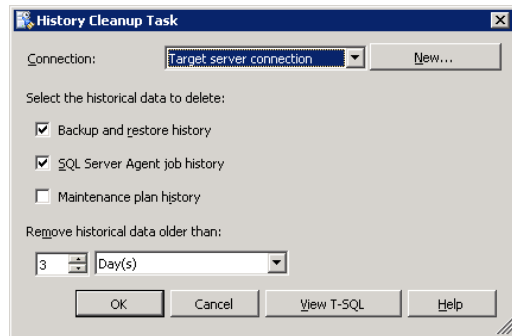
Kuva 19. Annetaan varmistusajolle nimi



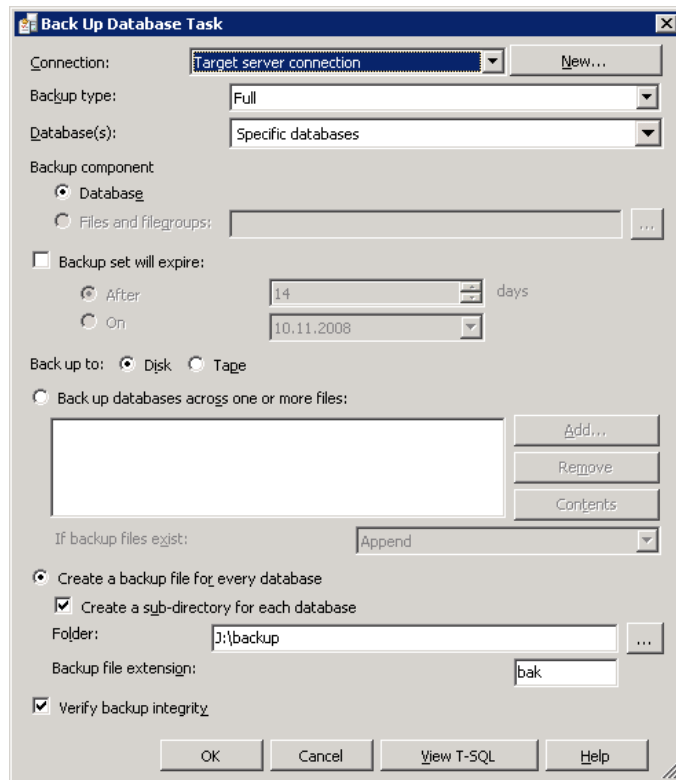
Kuva 20. Tietokannan eheyden tarkistaminen. Tarkistaa myös indeksit.



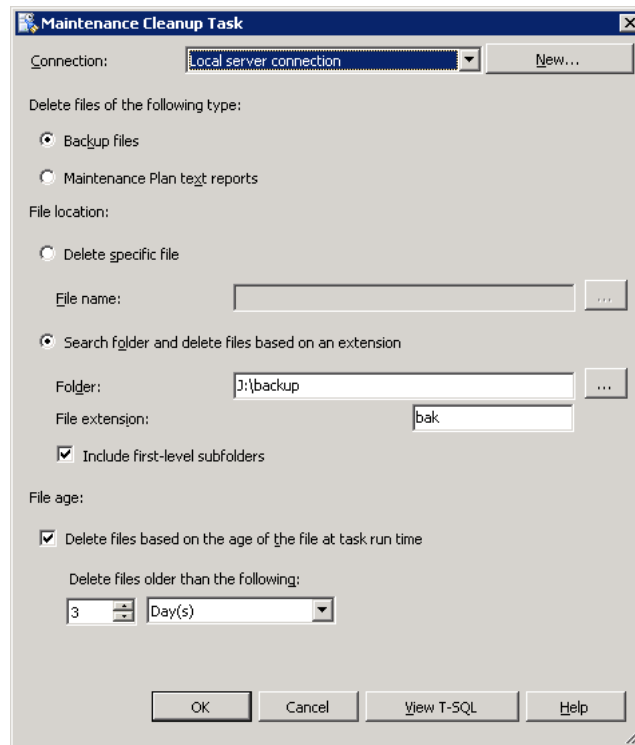
Kuva 21. Tietokannan kutistaminen



Kuva 22. Historian siivoaminen. Backupit, restoret, Sql Server agentin historia ja Maintenance planin historia.

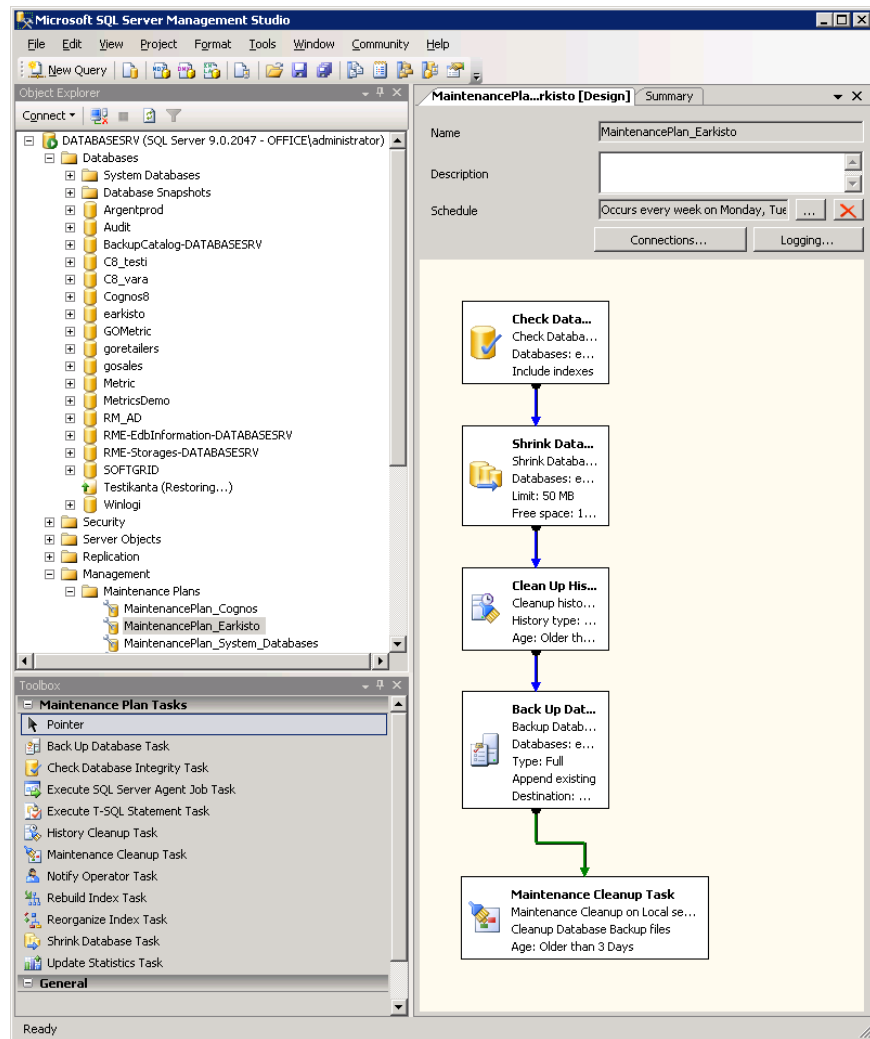


Kuva 23. Tietokannan varmistaminen. Backup type: full, simple tai Bulk-logged. Valitaan Full-vaihtoehto.



Kuva 24. Varmistustiedostojen poisto

Seuraavassa kuvassa näkyy koko varmistusajo tehtävineen. Maintenance Plan Tasks ikkunasta valitaan aiemmin esitellyt työvaiheet.

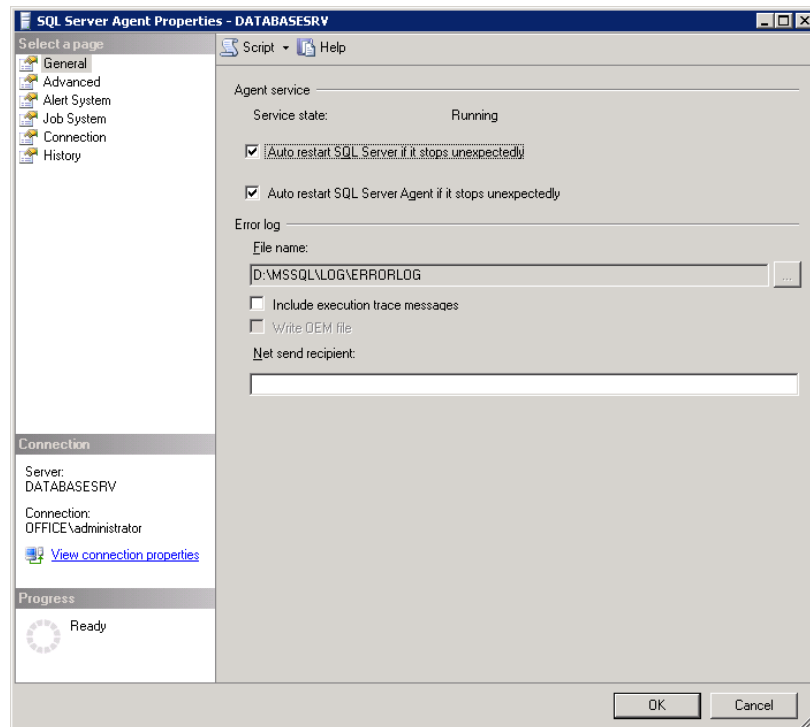


Kuva 25. Varmistusajon tehtävät

Tämän jälkeen tietokannan varmistukset menevät ajastetusti valitulle levyille, joka pitää tallentaa nauhalle ajastetusti.

5.3 SQL Server Agent

SQL Server Agent on SQL palvelimen sisällä toimiva ohjelma, joka pyörittää varmistusajoja ja muita ajastettuja toimintoja. SQL Server agentin asetuksiin kannattaa laittaa ruksit kohtiin Auto restart SQL Server if it stops unexpectedly ja Auto restart SQL Server Agent if it stops unexpectedly. Jos Windows- ja tietokantapalvelin käynnistetään uudelleen, niin Agent ei nouse ylös automaattisesti vaan joudutaan käynnistämään manuaalisesti. Tällöin eivät varmistuksetkaan toimi. Tämä pitää olla päällä, jos palvelinta käynnistää joku muu kuin tietokannahoitaja.



Kuva 26. SQL Server Agentin asetukset

Lopuksi vielä poistetaan SQL Server 2000 arkistopalvelimelta. Tämä tehdään Windows Serverin lisää tai poista sovellus kohdasta ja vaatii palvelimen uudelleenkäynnistämisen, jotta se poistuu kokonaan.

6 YHTEENVETO

Siirrettiin onnistuneesti arkistojärjestelmä jo aiemmin asennettuun keskitettyyn SQLServer -palvelimeen. Hieman ongelmia aiheutti se, että SQLServer osaaminen oli siirtoa tehneillä henkilöillä SQLServer2000:sta, mutta nyt kyseiset henkilöt ovat käyneet kurssilla. Kurssilla huomattiin, että muutaman asian olisi voinut tehdä helpomminkin tai eritavalla mm. tässä arkiston siirrossa käyttäjien siirron olisi voinut tehdä skriptillä, mutta meillä siihen osamista ei silloin ollut.

Käyttäjille konkreettinen ero aikaisempaan ratkaisuun verrattuna oli sovelluksen käytön helpottuminen, koska sovelluksen haut tietokannasta nopeutuivat huomattavasti.

Jatkossa lisäämme keskitettyihin servereihin lisätietokantoja ja virtualisoimme sovelluspalvelimia. Keskitettyjä kantapalvelimia on helpompi jatkossa

valvoa. Keskitetyt kantapalvelimet helpottavat vuosittaisia jatkuvuustestauksia, kun sovelluspalvelimet voidaan palauttaa virtuaalisena.

SQL Server 2008:aan siirrytään myöhemmin kun ensin sovellustoimittajat tukevat sitä. Siirtyminen SQL Server 2008:aan on helpompaa kuin SQL Server 2000:sta SQL Server 2005:een, koska käyttöliittymä on lähes samanlainen kuin SQL Server 2005:ssä.

VIITELUETTELO

- [1] SQLServer overview. Microsoft. [Verkkodokumentti.] [Viitattu 27.11.2008.]
[<http://www.microsoft.com/technet/prodtechnol/sql/2005/overview.mspx>].
- [2] SQLServerin ydinosaat. (Kuva.) Microsoft. [Viitattu 27.11.2008.]
[http://www.microsoft.com/library/media/1033/technet/images/prodtechnol/sql/2005/sqlserver001_big.png].
- [3] SQLServer2005 lisenssivaihtoehdot. Microsoft. [Verkkodokumentti.] [Viitattu 27.11.2008.]
http://www.microsoft.com/finland/license/business/palvelinohjelmistot/product_4.mspx.
- [4] SQLServer2005 tuotteiden vertailu. Microsoft. [Verkkodokumentti.] [Viitattu 27.11.2008.]
[<http://www.microsoft.com/sqlserver/2005/en/us/compare-features.aspx>].
- [5] Tietokantojen historiaa. Wikipedia. [Verkkodokumentti.] [Viitattu 27.11.2008.] [http://en.wikipedia.org/wiki/Charles_Bachman].
- [6] Tietokantojen historiaa. Wikipedia. [Verkkodokumentti.] [Viitattu 27.11.2008.] [<http://fi.wikipedia.org>]. Haku sanoilla "tietokantojen historiaa".
- [7] SQL-kielen komentoja. Wikipedia. [Verkkodokumentti.] [Viitattu 27.11.2008.] [<http://en.wikipedia.org/wiki/Sql>].
- [8] Yleistä Oraclesta. Wikipedia. [Verkkodokumentti.] [Viitattu 27.11.2008.] [http://en.wikipedia.org/wiki/Larry_Ellison].
- [9] Yleistä MySQL:stä. Wikipedia. [Verkkodokumentti.] [Viitattu 27.11.2008.] [<http://www.mysql.com/products/>].
- [10] Yleistä PostgreSQL:stä. Wikipedia. [Verkkodokumentti.] [Viitattu 27.11.2008.] [<http://www.postgresql.org/about/>].

- [11] Juurinen, Susanna. Insinööriyö: Argent-valvontajärjestelmä.
[Viitattu 1.6.2008.]