

Opinnäytetyö AMK

Konetekniikka

2022

Joni Wahlroos

I/O-tietokannan kehittäminen

– Meyer Turku Oy



Opinnäytetyö AMK | Tiivistelmä

Turun ammattikorkeakoulu

Konetekniikka

2022 | 51 sivua

Joni Wahlroos

I/O-tietokannan kehittäminen

- Meyer Turku Oy

Tämä opinnäytetyö tehtiin Meyer Turku Oy:n toimeksiannosta. Tämän työn tavoitteena on perehtyä Meyer Turku Oy:n I/O-tietokanta (Input/Output) -järjestelmän käyttöön ja työtapoihin. Tarkoituksena on selvittää I/O-tietokantaa säännöllisin väliajoin käyttävien työntekijöiden kohtaamia ongelmia ja haasteita I/O-tietokannan käytössä. Selvityksen pohjalta on tarkoitus löytää parempi ratkaisu nykyisen I/O-tietokanta järjestelmän tilalle tai lähteä parantamaan vanhaa.

Työ aloitettiin kartoittamalla alustavasti, mitä markkinoilta löytyy, mutta etsinnöissä ei löytynyt yhtään juuri tähän tarkoitukseen sopivaa. Selvityksessä käsiteltiin järjestelmän nykytilanne ja uudelta järjestelmältä toivottavat ominaisuudet. Nykytilanteen kartoittamisessa löytyi jo olemassa oleva SQL (Structured Query Language) -servereillä toimiva kaapelitietokanta, jota voitaisiin myös käyttää mallina I/O-tietokannan rakentamisessa.

Kartoituksen perusteella päädyttiin SQL-pohjaiseen I/O-tietokantaan. Pohja on jo valmiina kaapelitietokannassa, ja siihen voitaisiin rakentaa rinnalle I/O-tietokanta. Exceliin rakennettaisiin VBA:n (Visual Basic for Applications) avulla käyttöliittymä, jolla päästään käyttämään SQL:n I/O-tietokannan tietoja.

Asiasanat: SQL, I/O-tietokanta, Visual Basic for Applications

Bachelor's Thesis | Abstract

Turku University of Applied Sciences

Mechanical Engineering

2022 | 51 pages

Joni Wahlroos

Development of I/O-database

- Meyer Turku Oy

The thesis was commissioned by Meyer Turku Oy. The aim of this thesis was to study the use of Meyer Turku Oy's I/O-database system, operation and working method. The purpose is to find out the problems and challenges faced by employees who use the I/O database at regular intervals when using the I/O database. Based on the report, the intention is to find a better solution to replace the current I/O database system or to start improving the old one.

The work began with a preliminary mapping of the software, what can be found on the market, but in the search did not find any suitable ones for this purpose. The report went through the current state of the system and what would be expected from the new system. In mapping the current situation, an existing SQL (Structured Query Language) server-based cable database was found, which is also used as a model for building the I/O database.

Based on the mapping, we decided on a SQL-based I/O database. Good foundation is already there in the cable database, where I/O-database could be built alongside it. Using VBA, an interface would be built in Excel to access the data in the SQL I/O-database.

Keywords: SQL, I/O-Database, Visual Basic for Applications

Sisältö

Käytetyt lyhenteet ja sanasto	8
1 Johdanto	9
1.1 Tavoite	9
1.2 Meyer Turku Oy	9
2 Lähtötilanne	10
2.1 Lähtötilanteen vahvuudet	10
2.2 Lähtötilanteen heikkoudet	10
2.3 Ohjelmiston alustava etsintä	11
2.4 Suomen Meyer Turun telakka	11
2.5 Saksan Meyer Werft Papenburg	12
2.6 Vaatimukset toimivalta järjestelmältä	12
2.7 Valittu tietokanta	12
3 Microsoft SQL -tietokanta	14
3.1 Microsoft SQL -serveri	14
3.2 Eheyssäntö	15
3.2.1 Chen-notaatio	15
3.2.2 Crow's foot -notaatiota	19
3.2.3 Tietokantaan sisäänrakennettu visuaalinen työkalu	21
3.2.4 Indeksointi	22
3.2.5 Tietokannan normalisointi	23
3.2.6 Tietokannan skeemat	24
3.2.7 Perusavain	25
3.2.8 Viiteavain	25
3.3 SQL-Serverin toimintoja	25
3.3.1 Proseduurit	25
3.3.2 Näkymät	26
3.3.3 Herättimet	27
3.3.4 Default-lause	27

3.3.5 Unique-lause	27
3.3.6 Check-lause	28
3.3.7 Is not null-lause	28
3.3.8 Character ja Variable character -tietotyypit	28
3.3.9 Numeeriset-tietotyypit	29
3.3.10 Päivämäärä-tietotyypit	29
3.4 SQL Server Management Studio (SSMS)	30
3.5 DML	30
3.5.1 SELECT-komento	30
3.5.2 INSERT-komento	31
3.5.3 UPDATE-komento	31
3.5.4 DELETE-komento	31
3.6 DDL	32
3.6.1 CREATE-komento	32
3.6.2 ALTER-komento	33
3.6.3 DROP-komento	34
4 Excel ja VBA	35
4.1 Excel	35
4.2 Excel kehitystyökalut	35
4.3 VBA	37
4.3.1 VBA:n projektihakemisto	37
4.3.2 Sub-proseduuri	37
4.3.3 Function-proseduuri	38
4.3.4 VBA UserForm	39
4.3.5 Muuttujat	40
4.4 Makrot	41
5 Nyt ja tulevaisuudessa	42
5.1 Käyttöliittymän rakentaminen	42
5.2 SQL-serverin edut ja haitat	43
5.2.1 Etuja	43
5.2.2 Haittoja	44

5.2.3 SQL I/O -tietokannan rakentaminen	44
5.3 Mahdolliset säästöt ja kustannukset tulevaisuudessa	44
5.3.1 Säästöjä virheiden ehkäisystä	44
5.3.2 Säästöjä työtehokkuudesta	45
5.3.3 Säästöjä Master I/O -listan kokoamisesta	45
5.3.4 Säästöt raportoinnin helpottamisesta	46
5.3.5 Kustannukset SQL I/O -tietokannan rakentamisesta ja ylläpidosta	48
6 Yhteenveto	49
Lähteet	50
Kuvat	
Kuva 1. Kohde (Entity) tyypit (Vertabelo 2014).	15
Kuva 2. Ominaisuus (Attribute) tyypit (Vertabelo 2014).	16
Kuva 3. Esimerkki Chen-notaatio (Vertabelo 2014).	16
Kuva 4. Chen-notaatio suhteet kohteiden välillä (Vertabelo 2014).	17
Kuva 5. Pakollinen tai valinnainen -suhde (Vertabelo 2014).	17
Kuva 6. Kohteiden välisiä suhteita (Vertabelo 2014).	18
Kuva 7. Osallistumisrajoitukset (Vertabelo 2014).	18
Kuva 8. Chen-notaatio esimerkki (Vertabelo 2014).	19
Kuva 9. Crow's foot -notaatio (Vertabelo 2016).	20
Kuva 10. Crow's foot -notaatio (Vertabelo 2016).	20
Kuva 11. Crow's foot -notaatio esimerkkejä (Vertabelo 2016).	20
Kuva 12. Kaapelitietokanta.	22
Kuva 13. B+ puu indeksi (ReseachGate 2019).	23
Kuva 14. Näkymän luonti SQL Server management studiossa.	26
Kuva 15. Näkymä tiedosto puussa SQL Server management studiossa.	26
Kuva 16. Onnistunut kysely SQL Server management studiolla.	30
Kuva 17. INSERT-komento (Microsoft 2022).	31
Kuva 18. UPDATE-komento (Microsoft 2022).	31
Kuva 19. DELETE-komento (Microsoft 2022).	32

Kuva 20. CREATE-komento (Microsoft 2022).	33
Kuva 21. ALTER-komento (Microsoft 2022).	33
Kuva 22. DROP-komento (Microsoft 2022).	34
Kuva 23. Kehitystyökalut työkalurivi Excelissä.	36
Kuva 24. Lisää valikko Excelin kehitystyökalurivillä.	36
Kuva 25. Painikkeet Excel työkirjassa.	36
Kuva 26. Projekti näkymä VBA:ssa.	37
Kuva 27. Yhdistäminen SQL-serverille VBA:n avulla.	38
Kuva 28. Function-proseduuri VBA:ssa (Microsoft 2022).	39
Kuva 29. Userform avattuna Exceliin.	39
Kuva 30. VBA:n Userform.	40
Kuva 31. Dim- ja Public-muuttujat (Microsoft 2022).	40
Kuva 32. Käyttöliittymä Excelissä.	42
Kuva 33. I/O-data flow.	43
Kuva 34. Global design projekti.	45
Kuva 35. SQL I/O -tietokannan käyttö suunnittelussa.	46
Kuva 36. Projektikohtainen raportointi.	47
Kuva 37. Asennuksen raportointi.	47
Kuva 38. Käytöönoton raportointi.	48

Käytetyt lyhenteet ja sanasto

Attribute	Ominaisuus, Sarake (Lahtonen 2004.)
DDL	Data Definition Language (Microsoft 2022.)
DML	Data Manipulation Language (Microsoft 2022.)
Entity	Kohde, Taulu (Lahtonen 2004.)
ER-malli	Entity-Relationship model, (Lahtonen 2004.)
I/O	Input/Output, Tulo/Lähtö (Techopedia 2020.)
RDBMS	Relational Database Management System (Microsoft 2022.)
SQL	Structured Query Language (Microsoft 2022.)
SSMS	SQL Server Management Studio (Microsoft 2022.)
T-SQL	Transact- Structured Query Language (Microsoft 2022.)
VBA	Visual Basic for Applications (Microsoft 2022.)

1 Johdanto

1.1 Tavoite

Tämän työn tavoitteena on perehtyä Meyer Turku Oy:n I/O-tietokanta-järjestelmän käyttöön ja työtapoihin. Tarkoitus on tutustua säännöllisin väliajoin I/O-tietokantaa käyttäviä työntekijöitä työtapoihin. Perehtymisestä saadun tiedon pohjalta on tarkoitus löytää parempi ratkaisu nykyisen I/O-tietokanta järjestelmän tilalle tai lähteä parantelemaan I/O-tietokantaa.

1.2 Meyer Turku Oy

Turussa on ollut jo vuodesta 1737 laivanrakennusosaamista. Turun telakka on sijainnut aikoinaan Aurajoen rannalla, mutta nyt toiminnot ovat kokonaan siirtyneet Pernon telakka-alueelle. Telakalla on vuosien saatossa rakennettu laivoja puisista höyrylaivoista aina loistoristeilijöihin. Omistajia on ollut vuosien varrella useita, ja nykyinen omistaja on saksalainen perheyhtiö Meyer Werft. (Museovirasto. 2009; Meyer Turku 2022a.)

Turun telakka on toiminut vuodesta 2014 Meyer Turku Oy -nimisenä omistuksen siirryttyä Meyer Werftille. Jan Meyer on johtanut Turun telakkaa aina vuodesta 2014 vuoden 2020 syyskuuhun, jonka jälkeen hänen veljensä Tim Meyer jatkoi Turun telakan toimitusjohtajana. Turku Oy työllistää yli 2000 henkilöä ja useita osa- ja laite- toimittajia ympäri Suomen. Meyer Turku Oy:n tytäryrityksiin kuuluu telakan yhteydessä toimiva laivojen yleisiin tiloihin avaimet käteen periaatteella toimittava Shipbuilding Completion Oy, Piikkiössä sijaitseva hyttitehdas Piikkiö Works Oy ja Raumalla laivanrakennus- ja offshore-alan suunnitteluyritys ENG'nD OY. Meyerin muut toimipisteet sijaitsevat Saksassa, Meyer Werft Papenburgissa ja Neptun Werft Rostockissa. (Meyer Turku 2022b.)

2 Lähtötilanne

Tällä hetkellä Meyerin Turun telakalla tehdään I/O-tietokanta Microsoftin Excel ohjelmistolla. Excelissä käytetään työtä helpottamaan työntekijöiden itse kehittämiä, Excel VBA-koodia ja nauhoitettuja makroja. Master-I/O-tietokantaan kerätään laivan kaikki eri I/O-järjestelmät yhteen Excel-tiedostoon. Kaikki valmiit tiedostot ja dokumentit tallennetaan Kronodoc-tiedostojen hallintajärjestelmäpalveluun.

2.1 Lähtötilanteen vahvuudet

Excelin hyvänä puolena voidaan pitää, että se on kaikille tuttu ja sen käyttöön ei tarvita laajaa perehdytystä. Valmiita telakan henkilöstön luomia mallipohjia voidaan käyttää useissa projekteissa suoraan tai pienellä muokkauksella, samoin kuin telakan henkilöstön nauhoittamia makroja ja luomaa VBA-koodia voidaan käyttää useissa projekteissa.

2.2 Lähtötilanteen heikkoudet

Tällä hetkellä samanaikainen tiedostojen käyttö on hankalaa. Kronodocissa tiedostot ovat järjestelmä kohtaisissa Excel-tiedostoissa, kuten esimerkiksi pääkoneen I/O-pisteet, voitelujärjestelmän I/O-pisteet, jäteveden puhdistus järjestelmän I/O-pisteet ja pilssijärjestelmän I/O-pisteet. Suunnittelijat lataavat Kronodocista tiedoston omalle koneelle ja alkavat työskennellä. Jos on tarvetta työskennellä saman järjestelmän tiedoston kanssa, se aiheuttaa päivitysongelman. Kun tiedosto palautetaan Kronodociin, viimeiseksi palautettu tiedosto jää voimaan. Päivitysongelmaa on yritetty välttää, sopimalla kuka työskentelee milloinkin minkäkin järjestelmän kanssa.

Master I/O -listan kanssa on samat ongelmat kuin järjestelmä listojen kanssa. Yksi henkilö on aina kerrallaan vastuussa Master I/O -listan kasaamisesta. Näin vältetään samanaikaisen käytön aiheuttama päivitysongelma. Master I/O -lista

kootaan järjestelmä listojen pohjalta. Master I/O -listan tiedosto koko paisuu suureksi jopa pienissä projekteissa, joka aiheuttaa hitautta Excelin käytössä.

2.3 Ohjelmiston alustava etsintä

Alustavissa etsinnöissä ainut löytynyt ohjelmisto, joka oli tarkoitettu I/O-listojen tekoon oli tehty VBA-koodilla Excelille. Löytyneen ohjelmiston huonona puolena oli se, että oli vain tietyn valmistajan komponenteille suunniteltu eli ei olisi ollut käyttöön sopiva ilman muokkaamista. Tämän ohjelmiston käytöstä olisi myös laskutettu kuukausimaksua. Etsinnöissä ei löytynyt muita ohjelmistoa I/O-tietokannan tekoon tai ylläpitoon. Aiheeseen liittyviltä sivustoilta ja keskustelualueilta (mm. Mr PLC, Reddit, Quora) kävi ilmi, että alan työntekijät käyttävät itse tekemiään Excel VBA-koodia ja makroja niin kuin Meyerilläkin jo tehdään.

2.4 Suomen Meyer Turun telakka

Toiveena olisi, että Master-listan koonti helpottuisi ja samanaikainen käyttö olisi mahdollista. Tehdyistä muutoksista kirjattaisiin ylös lokiin, kuka on tehnyt muutoksen ja milloin, jotta virheen sattuessa olisi helpompi selvittää miksi virhe tapahtui. Raportteja saisi helposti luotua suunnitteluun, toimittajalle, alihankkijoille, telakan käyttöönotto tiimille ja projektijohdolle ja pystyttäisiin tekemään yhteisiä projekteja Saksan Papenburg Meyer telakan ja Suomen Turun Meyer telakan välillä.

Kaapelin reitityksessä käytetään jo apuna datan tallennukseen Microsoft SQL -ympäristössä toimiva kaapelin listoille tehty tietokantaa, jota voitaisiin hyödyntää pohjana myös I/O-listoille. Tulevaisuuden suunnitelmassa oli myös mobiili raportoinnin työkalu kaapelin veto tiimille ja käyttöönotolle.

2.5 Saksan Meyer Werft Papenburg

Saksan Papenburgin Meyerin telakalla on vastaavasti myös käytössä Excel. Samat ongelmat ilmenivät Saksan Papenburgin telakalla. Master I/O -tietokanta on sama mutta siellä tehdään suoraan kaikki työ Master I/O -tietokantaan. Excelliin VBA:lla rakennettua käyttöliittymää oli jalostettu omaan käyttöön sopivaksi erittäin pitkälle, pidemmälle kuin Turun telakalla. Käyttöliittymään oli myös lisätty automaattinen täyttö ominaisuus, jolla sai nopeasti käyttöliittymästä lisättyä yksi tai viisi riviä I/O-pisteitä kerralla valitulla tunnuksella. Käyttöliittymään oli myös rakennettu systeemi kohtaisia näkymiä, joita sai lomakkeen ylälaidasta nopeasti nappia painamalla. Papenburgin telakalla on myös ollut käytössä 80-luvulta peräisin oleva ohjelmisto IBM AS/400, josta luovuttiin I/O-listojen luonnissa muutama projekti sitten.

2.6 Vaatimukset toimivalta järjestelmältä

Toimivalta järjestelmältä vaaditaan samanaikaisen työnteon mahdollisuus siten, ettei tarvitsisi varata tiedostoa itselle työn ajaksi. Uuden järjestelmän täytyisi mahdollistaa yhteisten projektien teon Saksan Papenburgin telakan kanssa. Projektit voisivat olla joko Turun telakalla tai Papenburgin telakalla. Suunnittelu voisi tapahtua omissa yksiköissä Papenburgissa ja Turussa. Käyttäjä virheiden vähentäminen ja estäminen olisi tärkeä osa uutta järjestelmää. Kuten myös erilaisten raporttien luonnin helpottaminen ja nopeuttaminen.

2.7 Valittu tietokanta

Kävi ilmi, että Meyer Turun telakalla on jo käytössä kaapelin- vedossa ja -suunnittelun apuna SQL-pohjainen tietokanta. Jo tehtyä pohjaa kaapeli tietokantaan voisi käyttää myös I/O-tietokannan luonnissa. Excelliin oli rakennettu käyttöliittymä VBA:n avulla päästään lukemaan ja kirjoittamaan tietokantaan. Käyttöliittymä rajoittaa suoraa pääsyä tietokantaan.

Meyer Turun telakan kehittämää tapaa tallentaa I/O-pisteitä käytettäisiin jatkossa myös Meyer Papenburgin telakalla. Näin saataisiin yhdenmukaistettua toimintatavat ja yhteistyö telakoiden välillä hiljaisempina hetkinä mahdollistuisi.

Käyttöliittymässä voisi käyttää ideoita Saksan Papenburgin telakan ja Turun telakan nykyisistä Excelliin luoduista makroista ja koodeista.

Päädyttiin käyttämään kaapelilistoille tarkoitettua SQL servereillä olevaa tietokantaa pohjana. Rakentamalla SQL kaapeli tietokannan osaksi käyttämällä jo olemassa olevaa SQL kaapeli tietokantaa voidaan käyttää SQL:ää I/O-tietokannan pohjana.

3 Microsoft SQL -tietokanta

Suurin osa maailman datasta oli tallennettu suurtietokoneille 1960-luvulla. Suurtietokoneen olivat kalliita käyttää ja suurin kustannus tuli monimutkaisesta hallintajärjestelmästä. Matemaatikko E. F. Coddia pyydettiin Englannista Yhdysvaltoihin IBM:n apuun selvittämään ongelmaa. Sen aikaiset tietokannat perustuivat puuta muistuttavaan hierarkkiseen tietokantamalliin. Coddin ideana oli, ettei käyttäjän tarvitsisi tietää tietokoneen fyysisen rakenteen yksityiskohtia tämä yksinkertaistaisi silloista tapaa hakea tietoa. Tietokoneen tulisi selvittää missä ja miten tiedot on tallennettu, ja miten tietoa voidaan hakea esittämällä tietokoneelle yksinkertainen kysymys. IBM:llä ollessaan Codd kirjoitti ideastaan artikkelin "A Relational Model of Data for Large Shared Data Banks.". Artikkelin pohjalta Don Chamberlin ja Ray Boyce kehittivät SQL kyselykielen, jota nykyään käytetään laajalti tiedon saantiin relaatiotietokannoista. (IBM 2022.)

Microsoft SQL -serveri sai alkunsa, kun se alkoi kehittää Ashton-Taten ja Sybasen kanssa vuonna 1982 Sybase SQL -serveriversiota IBM OS/2:lle. Vuonna 1993 Microsoft lopetti yhteistyön Sybasen kanssa ja jatkoi oman SQL-serverin kehitystä. Tällä hetkellä uusin versio on SQL-Server 2022. (Techtarget 2019; Microsoft 2022.)

3.1 Microsoft SQL -serveri

Microsoft SQL -serveri on yksi suurimmista markkinoilla olevista relaatiotietokannan hallintajärjestelmistä (RDBMS). Microsoft SQL -serveri on rakennettu standardoidun SQL-kielen pohjalta. Microsoft SQL -serverissä on käytetty transact-SQL (T-SQL) -kieltä, laajentamaan SQL-kieltä. Microsoftilta löytyy monta eri versiota SQL-serveristä, jotka on suunnattu niin pienelle kuin suurelle käyttäjäryhmälle. Tietokanta voi sijaita tietokoneella tai serverillä. (Techtarget 2019.)

3.2 Eheyssäntö

Tietokannan eheyttä voidaan tarkastella käsitteellisen mallintamisen avulla. Vuonna 1976 Chen esitteli tunnetuimman ja käytetyimmän käsittemallin ER-mallin (Entity-Relationship model), jota on myöhemmin laajennettu ja muunneltu. ER-mallissa tarkastellaan kohteeseen (entity) tulevat ominaisuudet (attribute) piirtämällä ja muodostamalla ali- ja ylityyppejä. Alityyppi perii aina ylityypin kaikki ominaisuudet. Muodostetaan suhteet näiden tietojen välille. Suhteen asteita kutsutaan kardinaalisuudeksi. Se kertoo, kuinka moneen suhteeseen tieto osallistuu ja voi osallistua. Kardinaalisuuksia ovat: 1 to 1, 1 to N ja M to N. Nämä voidaan ilmaista myös seuraavasti: yhden suhde yhteen, yhden suhde moneen ja monen suhde moneen. Jos löytyy suhteita tiedoista, jotka ovat monen suhde moneen, ne tulee poistaa ennen tietokannan rakennusta eheyden säilyttämiseksi. Perusavain ja viiteavain pitävät tietokannan eheyttä yllä. (Lahtonen 2004, 18–30.)

3.2.1 Chen-notaatio

Kuvassa 1 vasemmalta oikealle on esitetty Chen-notaation kohdetyypit: kohde, heikko kohde ja assosiatiivinen kohde. Kohde on aina tunnistettavissa. Heikon kohteen olemassaolo riippuu toisesta kohteesta. Heikkoa kohdetta ei voida tunnistaa pelkästään ominaisuuksien avulla. Assosiatiivinen kohdetta käytetään kuvaamaan M-to-M suhdetta. (Vertabelo 2014.)



Kuva 1. Kohde (Entity) tyypit (Vertabelo 2014).

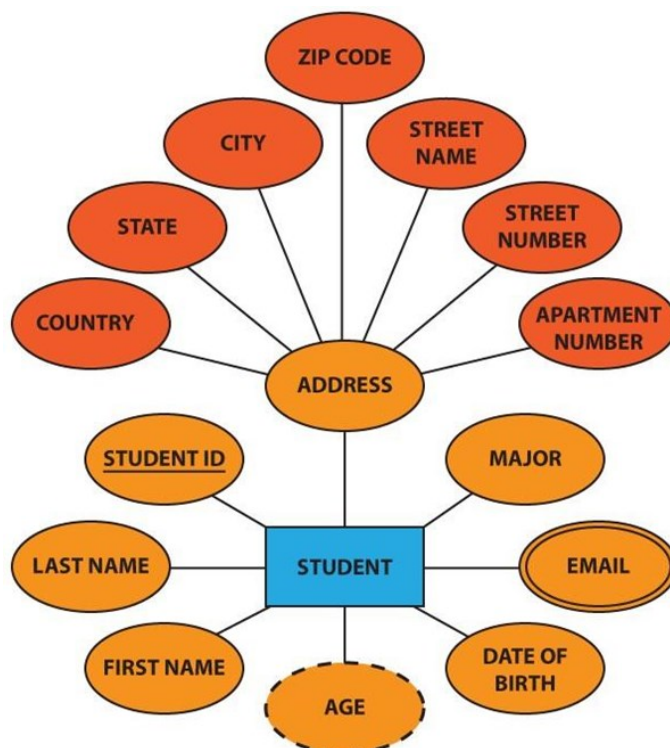
Kuvassa 2 vasemmalta oikealle on esitetty Chen-notaation ominaisuus (attributes) tyypit: tavallinen ominaisuus, avain ominaisuus, osittainen avain

ominaisuus, moniarvoinen ominaisuus ja johdettu ominaisuus. Avain ominaisuus identifioi kohteen. Avain ominaisuus on alleviivattu. Osittainen avain ominaisuus auttaa avain ominaisuutta yksilöimään ja identifioimaan heikon kohteen. Moniarvoisella ominaisuudella voi olla monta arvoa. Johdettu ominaisuus lasketaan (johdetaan) muista ominaisuuksista ja ei välttämättä ole tallennettu tietokantaan. (Vertabelo 2014.)



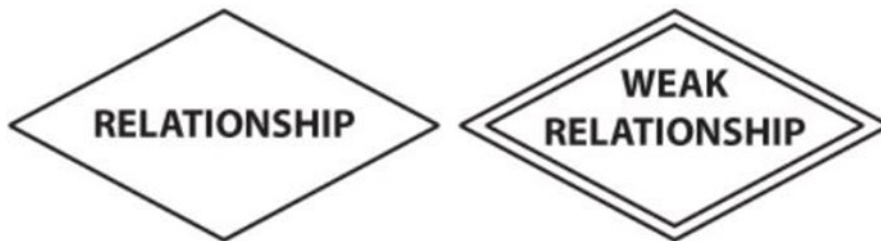
Kuva 2. Ominaisuus (Attribute) tyypit (Vertabelo 2014).

Kuvan 3 esimerkissä "STUDENT ID" on avain ominaisuus, "AGE" on johdettu ominaisuus, "EMAIL" on moniarvoinen ominaisuus ja loput ovat tavallisia ominaisuuksia. "COUNTRY", "STATE", "CITY", "ZIP CODE", "STREET NAME", "STREET NUMBER" ja "APARTMENT NUMBER" ominaisuudet ovat "ADDRESS" ominaisuuden alityyppejä. (Vertabelo 2014.)



Kuva 3. Esimerkki Chen-notaatio (Vertabelo 2014).

Kuvassa 4 vasemmalta oikealle on esitetty suhde merkintätapa kohteiden välillä: suhde ja heikko suhde. Vahvassa suhteessa kohteen olemassaolo on riippumaton muista kohteista ja löytyy avain ominaisuus. Heikossa suhteessa kohde on riippuvainen toisesta kohteesta. Kutsutaan myös vanhempi lapsi suhteeksi (Parent Child relationship). (Vertabelo 2014.)



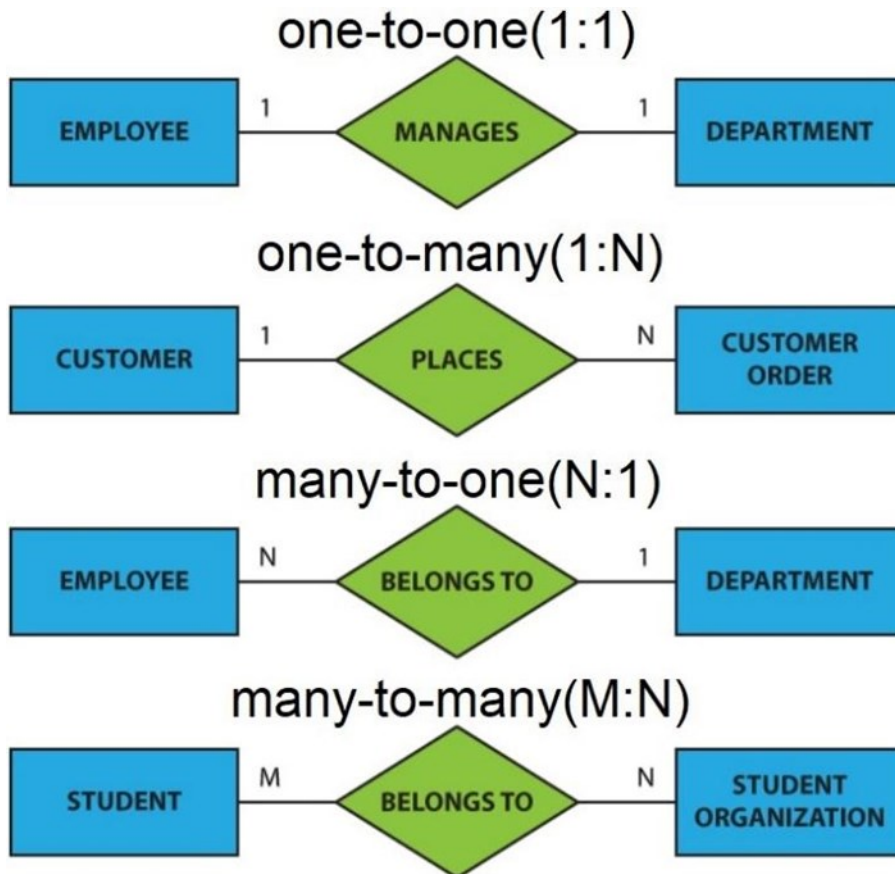
Kuva 4. Chen-notaatio suhteet kohteiden välillä (Vertabelo 2014).

Ehyt viiva suhteen ja kohteen välillä kuvaa pakollista suhdetta, kun taas katkoviiva ei pakollista suhdetta (kuva5). (Vertabelo 2014.)



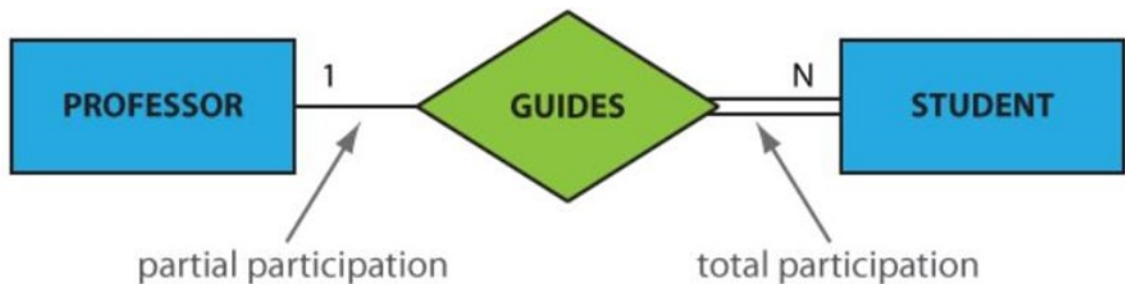
Kuva 5. Pakollinen tai valinnainen -suhde (Vertabelo 2014).

Kuvan 6 one-to-one-esimerkissä työntekijä hoitaa yhtä osastoa. One-to-many-kohdassa asiakas tekee tilauksen, jossa voi olla yhdessä tilauksessa useampi tuote. Many-to-one-kohdassa työntekijät kuuluvat osastolle. Many-to-many-kohdassa oppilaat kuuluvat opiskelijajärjestöihin. (Vertabelo 2014.)



Kuva 6. Kohteiden välisiä suhteita (Vertabelo 2014).

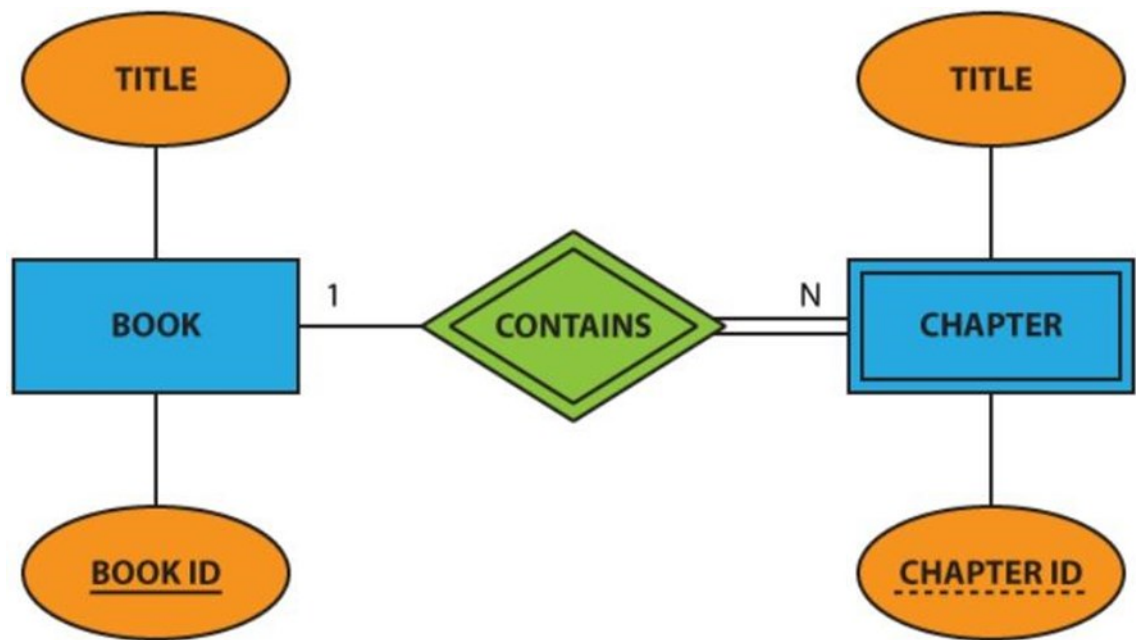
Kaksoisviiva kohteen ja suhteen välillä tarkoittaa kokonaisosallistumista, kun taas viiva osittaisesta osallistumisesta. Kuvassa 7 jokaista opiskelijaa ohjaa professori mutta jokainen professori ei ohjaa oppilasta. (Vertabelo 2014.)



Kuva 7. Osallistumisrajoitukset (Vertabelo 2014).

Kuvassa 8 kohteen kirja ja heikon kohteen luku välillä on heikko suhde sisältää. Kirja voi sisältää monta lukua mutta jokainen kirja ei sisällä lukua. Koska kirja ei

sisällä välttämättä lukua on kohde luku heikko. Kohteen olemassaolo riippuu kohteesta kirja. Kohteella kirja on ominaisuus otsikko ja avain ominaisuus kirja tunnus. Kohteella luku on ominaisuus otsikko ja osittainen avain ominaisuus luku tunnus. Osittainen avain ominaisuus luku tunnukselle tulee koska kohde luku on heikko kohde. (Vertabelo 2014.)

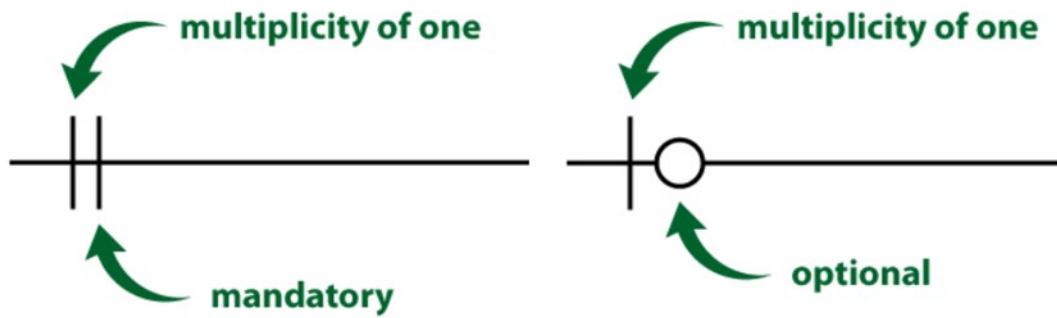


Kuva 8. Chen-notaatio esimerkki (Vertabelo 2014).

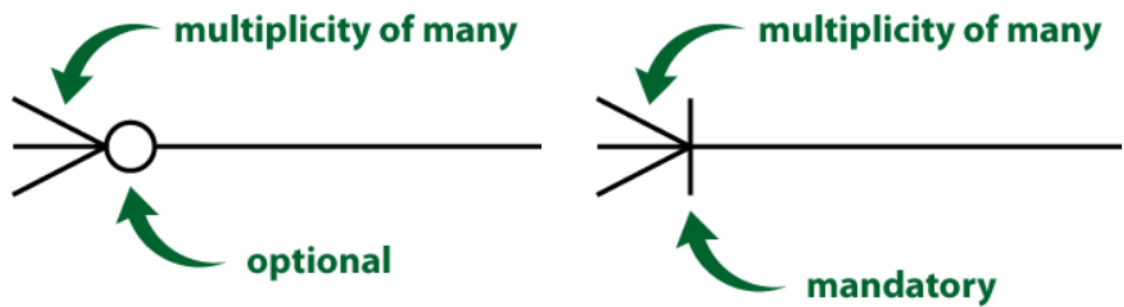
3.2.2 Crow's foot -notaatiota

Crow's foot -notaatio on yksi ER-mallin visuaalisista työkaluista. Crow's foot -notaatio tunnetaan myös nimellä IE-notaation. Crow's foot -notaatiossa ensimmäinen viiva suoralla tarkoittaa kuinka monta kertaa tieto esiintyy taulussa (kuva 9). Tieto voi esiintyä taulussa kerran tai monta kertaa. Toinen viiva suoralla tarkoittaa, onko tieto pakollinen vai ei. Kuvassa 9: yksi ja vain yksi sekä yksi tai nolla. Kuvassa 10: monta tai yksi ja monta tai ei yhtään. (Vertabelo 2016.)

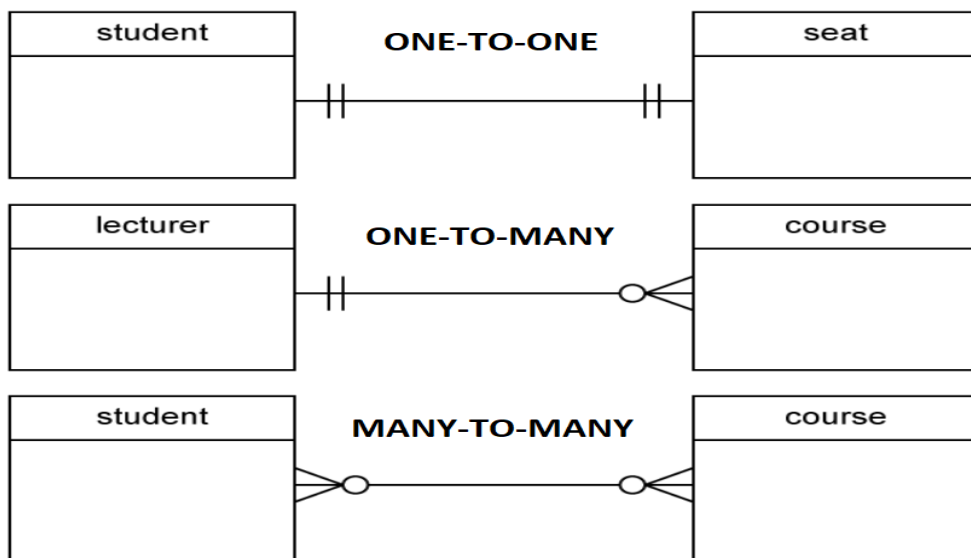
(Vertabelo 2016.)



Kuva 9. Crow's foot -notaatio (Vertabelo 2016).



Kuva 10. Crow's foot -notaatio (Vertabelo 2016).



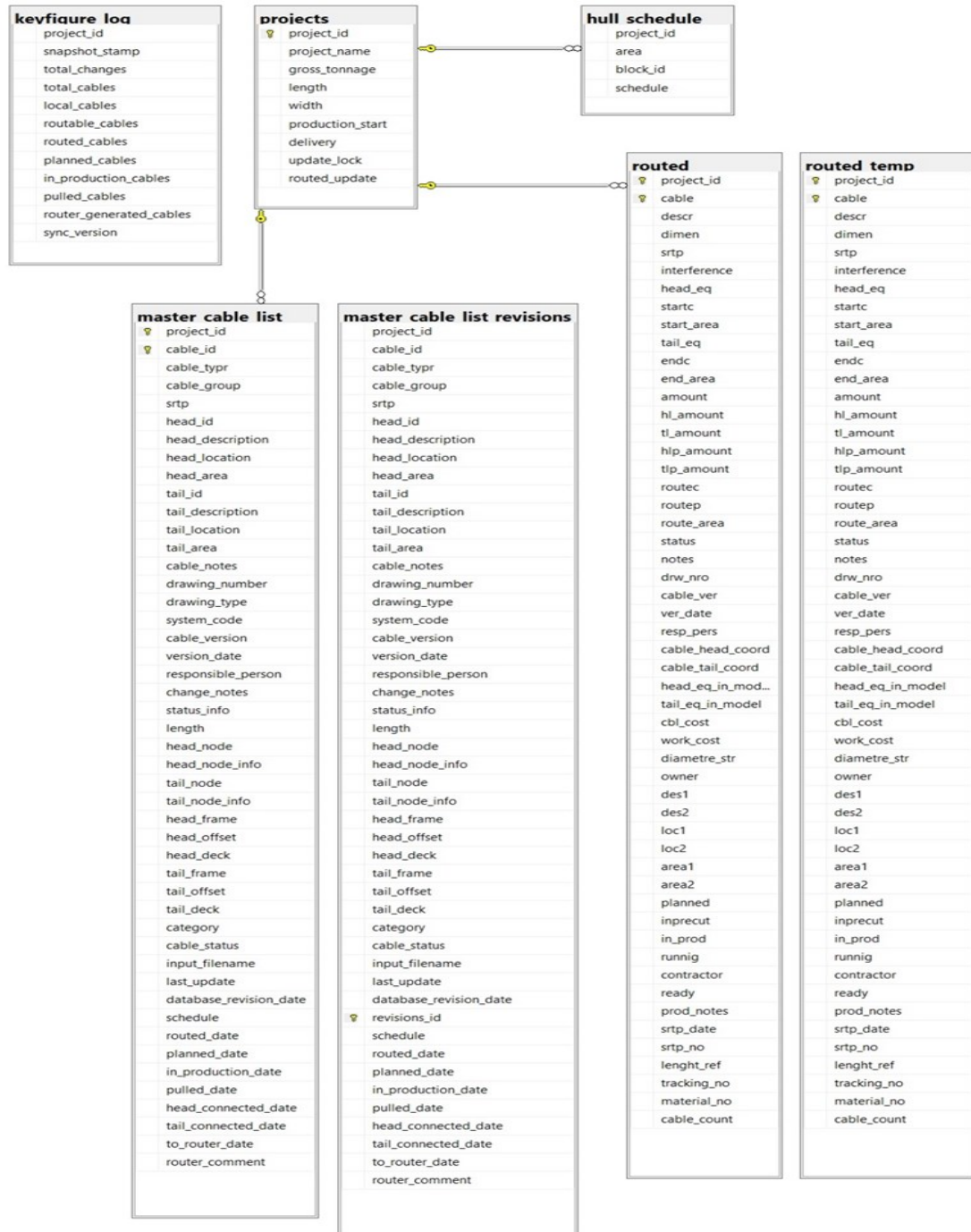
Kuva 11. Crow's foot -notaatio esimerkkejä (Vertabelo 2016).

Kuvan 11 esimerkeissä ylimpänä opiskelijalla voi olla yksi paikka ja paikalla voi olla yksi opiskelija, joten 1-to-1. Kuvan 11 keskimmaisessä esimerkissä luennoitsijalla voi olla monta luentoa ja luennolla voi olla vain yksi luennoitsija kerrallaan, joten 1-to-M. Kuvan 11 viimeisessä esimerkissä oppilaalla voi olla monta kurssia ja kurssilla voi olla monta opiskelijaa, joten M-to-M. (Vertabelo 2016.)

3.2.3 Tietokantaan sisäänrakennettu visuaalinen työkalu

SQL-tietokanta sisältää sisäänrakennetun visuaalisen työkalun tietokannan visualisointiin, tietokannan diagrammi. Jokaisessa tietokannassa on oma osio diagrammille. Diagrammeja voi tehdä useita yhdestä tietokannasta. Diagrammia voidaan muokata omien tarpeiden mukaan vaikka vain näyttämään kohteita ja kohteiden suhteita. Diagrammi ei ole pakollinen mutta helpottaa hahmottamaan kokonaisuutta paremmin kuten kohteiden välisiä suhteita ja ominaisuuksien toistoa kohteissa. (Microsoft 2022.)

Tietokannan diagrammissa on esitetty (kuva 12) kohteet ja sen ominaisuudet omissa tauluissa. Avain ominaisuudet on ilmoitettu avaimen kuvalla ominaisuuden vasemmalla puolella. Kohteiden välillä suhteet on ilmaistu viivoin, joiden päissä on avain ja kaksi ympyrää. Avain kuvaa, että kohteessa on avain ominaisuus ja toisessa päässä osittainen avain ominaisuus. Kohteessa projects on avain ominaisuus project_id, jonka vasta pari löytyy kohteesta hull schedule osittainen avain ominaisuus project_id. Samalla tavalla on yhdistetty kohteet master_cable_list ja master_cable_list_revision. Kohteen master cable list ja kohteen master cable list revision välillä avain ominaisuus ja osittainen avain ominaisuus on project_id ja cable_id. Kohteiden master cable list ja master cable list revision välille ei ole luotu viivaa kuvaamaan suhdetta.

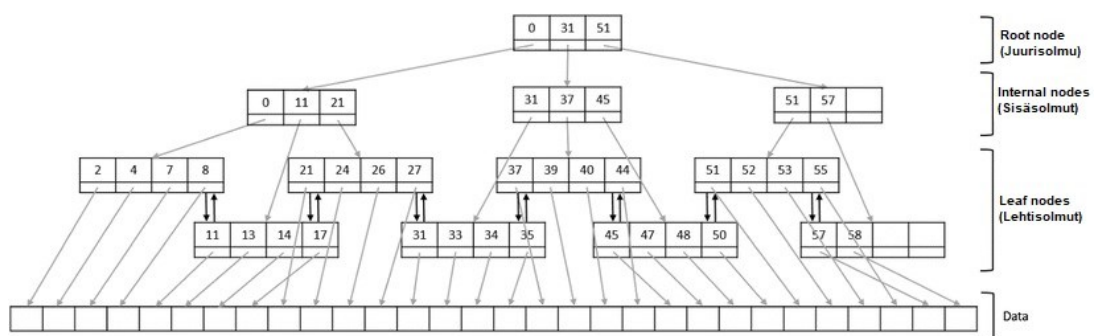


Kuva 12. Kaapelitietokanta.

3.2.4 Indeksointi

Tietokannan indeksi on verrattavissa kirja hakemistoon. Indeksointi nopeuttaa tietokannasta hakua mutta hidastaa tietokantaan lisäämistä, päivitystä ja

poistoa koska uudet ja päivitettyt tiedot on aina lisättävä ja muutettava indeksiin. Kun pääavain tai unique-lause luodaan, indeksi tulee automaattisesti kyseisille sarakkeeseen/sarakkeille. Kun käytetään useampaa kuin yhtä saraketta indeksi on näiden sarakkeiden yhdistelmä. SQL:ssä indeksi voi olla ryvästyvä (clustered) tai ei ryvästyvä (non clustered). Ryvästyvässä indeksissä arvojen fyysinen järjestys levyllä on indeksin mukaan, kun taas ei ryvästyvässä fyysistä järjestystä levyllä indeksin mukaan ei yllä pidetä. SQL:ssä käytetään B+ puu (kuva 13) rakennemalli indeksointiin. B+ puu rakennemallissa on juurisolmu, sisäsolmut, lehtisolmu ja datan fyysinen sijaintitieto levyllä. (Microsoft 2022.)



Kuva 13. B+ puu indeksi (ResearchGate 2019).

Esimerkki kuvasta 13. Jos haetaan vaikka numeroa 47. Aloitetaan juurisolmusta $0 \leq 31 < 51$. Edetään suoraan alas sisäsolmuun $31 \leq 37 < 45$ koska 47 menee numeroiden $31 \leq 51$ väliin. Edetään oikean puolimmaisenuolen puolelle koska 47 on isompi kuin 45. Lehtisolmusta löytyy numeron 47 paikka, joka osoittaa paikan datassa. Jos etsittäisiin numeroa 49 jolla on sama reitti puussa aina lehtisolmuun saakka. Numerolle 49 ei kuitenkaan hakutulosta löydy koska numerosta 49 ei ole tallennettu tietoa lehtisolmuun. Numeroa 49 ei ole tallennettu tietokantaan.

3.2.5 Tietokannan normalisointi

Kun suunnitelma tietokannan kohteista, ominaisuuksista ja kohteiden välisistä suhteista on ne hyvä normalisoida. Normalisoinnissa poistetaan turhatoisto, poistetaan turhat ominaisuudet ja korjataan epäjohdonmukaisuudet

ominaisuuksissa. On olemassa viisi normalisoinnin tasoa mutta kolmen ensimmäisen tason noudattaminen on riittävä. (Microsoft 2022.)

Ensimmäisessä normalisoinnin vaiheessa (1NF) ominaisuudet eivät saa muodostaa ryhmiä. Ominaisuus saa sisältää vain yhtä tietotyyppiä. Jos tarvetta on luotava kohteita sen mukaan, että saadaan poistettua ryhmät. Uudet kohteet liitetään toisiinsa avainominaisuuden avulla. (Microsoft 2022.)

Toisessa normalisoinnin vaiheessa (2NF) liitetään osittainen avainominaisuus ensimmäisessä vaiheessa tehtyyn kohteeseen. Kuitenkin tämän kohteen ominaisuudet ovat vain riippuvaisia avainominaisuudesta tai yhdistelmä avainominaisuudesta. (Microsoft 2022.)

Kolmannessa normalisoinnin vaiheessa (3NF) poistetaan ominaisuudet, jotka ei ole riippuvaisia avainominaisuudesta tai muodostetaan riippumattomille ominaisuuksille oma kohde, jos on järkevää tehdä niin. Joskus kolmannen vaiheen voi jättää pois, jos ei ole käytännöllistä suorittaa vaihetta. Esimerkiksi jos normalisointi aiheuttaa liian monen kohteen luonnin normalisoinnin saavuttamiseksi, joka taas hidastaa tietokantaa. (Microsoft 2022.)

Boyce Codd Normal Form (BCNF) poistetaan kaikki funktionaalisista riippuvuuksien tuloksista jäljelle jääneistä säännöttömyyksistä.

Neljännessä normalisoinnin vaiheessa poistetaan kaikki moniarvoiset riippuvuudet.

Viidennessä normalisoinnin vaiheessa poistetaan kaikki jäljelle jääneet säännöttömyydet. (Lahtonen 2004, 36.)

3.2.6 Tietokannan skeemat

Tietokannan skeemat ovat SQL:n tapa organisoida kohteita tietokannan sisällä. SQL-tietokannan oletus skeema on "dbo" joka tulee kohteen eteen pisteellä erotettuna. Tietokannan skeemojen määrää tietokannassa ei ole rajoitettu. Tietokannan skeemat lisäävät myös tietoturvaa. Tietokannan käyttäjille voidaan

antaa ja poistaa lupia käyttää yhtä tai useampaa skeemaa. Voidaan myös lupien sisällä säätää mitä voi tehdä tietokannan skeeman sisällä. Voidaan myöntää luku oikeus, muokkaus oikeus ja poisto oikeus (Microsoft 2022.).

3.2.7 Perusavain

Perusavain yksilöi taulukon joka rivin ja samalla pakottaa taulukon kokonaisuuden eheyttä. Pääavain voi olla sarakkeen tai maksimissaan 16 sarakkeen yhdistelmä ja kokonaispituus voi olla 900 tavua. Taulukossa ei voi olla kuin yksi pääavain. Pääavain sisältää aina Unique-lauseen ja Not null-lauseen. Tietokanta pitää huolen, ettei samaa pääavainta voida käyttää taulukossa uudelleen. (Microsoft 2022.)

3.2.8 Viiteavain

Viiteavain toimii nimensä mukaan linkkinä taulujen välillä viitaten aina pääavaimeen arvoon. Viiteavain yhdistetään aina pääavaimeen. Viiteavain voi olla sarakkeen tai sarakkeiden yhdistelmä. (Microsoft 2022.)

3.3 SQL-Serverin toimintoja

3.3.1 Proseduurit

Proseduurit ovat ennalta tallennettuja ohjelmia. Proseduurit voivat ottaa vastaan arvon ja palauttaa käsitellyn arvon takaisin kutsuvaan kohtaan ohjelmaa. Proseduurit voivat myös suorittaa toimintoja tietokannassa ja suorittaa toisi proseduureja. Proseduureja käytetään myös tietoturvallisuus syistä. (Microsoft 2022.)

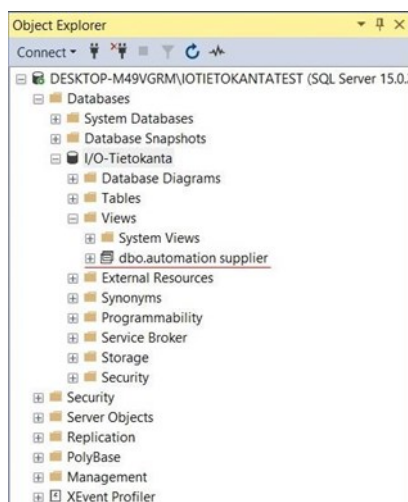
3.3.2 Näkymät

Näkymät määritellään tekemällä kysely SQL-tietokantaan (kuva 14). Näkymästä viitataan jo olemassa oleviin tietoihin. Näkymän tiedot voivat olla useasta taulukosta tai jopa useasta tietokannasta. Näkymillä on oma kohta tiedostopuussa (kuva 15.) erillään taulukoista. Näkymää voidaan käyttää myös turvallisuus syistä. Antamalla lukuoikeus näkymään ilman pääsyä näkymässä viitattuihin tauluihin. Näkymää voidaan myös käyttää yksinkertaistamaan, tarkentamaan ja mukauttamaan käyttäjien näkemystä tietokannasta. Näkymä tyyppiä on kolmea erilaista indeksoitu näkymä, osittainen näkymä ja järjestelmä näkymä. Indeksoitu näkymä on tallennettu fyysisesti tietokantaan ja on indeksoitu ryvästyvällä indeksoinnilla. Ositettu näkymä on kokoelma eri servereillä olevista kohteista, jotka on koottu yhdeksi kohteeksi. Järjestelmä näkymä on SQL:llä luoma järjestelmä näkymä metadatatista. (Microsoft 2022.)



```
SQLQuery1.sql - D_49VGRM\joniw (60)* - X
CREATE VIEW [automation_supplier] AS
SELECT * AS "ias_esd", i_o_cabinet, tag_no, last_updated, "desc", automation_desc1, automation_desc2, board_type, board, maker_card_no, channel, termina
FROM master_signal_list
```

Kuva 14. Näkymän luonti SQL Server management studiossa.



Kuva 15. Näkymä tiedosto puussa SQL Server management studiossa.

3.3.3 Herättimet

Herättimet ovat toiminto sarjoja, jotka suoritetaan automaattisesti, kun tietokannassa tehdään muutoksia. On olemassa DML (Data Manipulation Language), DDL (Data Definition Language) ja kirjautumisherätin. DML-herätin on kohteen tason herätin, DDL on tietokanta tason herätin ja kirjautumisherätin aktivoituu nimensä mukaan, kun kirjaututaan sisälle tietokantaan. DML-herätin aktivoituu, kun kohteessa tehdään muutoksia INSERT, UPDATE tai DELETE komennon kanssa. DDL-herättimet aktivoituvat, kun tietokantaan tehdään komennoilla CREATE, ALTER tai DROP muutoksia. DML ja DDL-herättimissä voidaan muuttaa herättimen käyttäytymistä jättämällä pois komento tai komentoja. (Microsoft 2022.)

3.3.4 Default-lause

Default-lause eli oletusarvo määritellään kenttään automaattisesti tietyllä arvolla, jos kenttä jää tyhjäksi. Arvo annetaan taulua luodessa tai lisäämällä erikseen taulun määrittelyyn. (Microsoft 2022.)

3.3.5 Unique-lause

Unique-lause tarkoittaa, että määritellyllä sarakkeella taulukossa ei saa olla yhtään samaa arvoa. Tietokanta hyväksyy arvoksi NULL, mutta vain kerran. Voidaan estää NULL arvon antaminen unique-lauseeseen kanssa lisäämällä sarakkeen ehtoihin NOT NULL. Tietokanta antaa virheilmoituksen, jos yrittää syöttää jo kertaalleen käytettyä arvoa kenttään. Unique-lause ehkäisee näin virheet, kun on tarve luoda yksilöllinen tunnus. Unique-lause kuuluu pääavaimen ominaisuuksiin. (Microsoft 2022.)

3.3.6 Check-lause

Check-lause voidaan asettaa annettujen arvojen tarkastussääntö sarakkeille. Voidaan asettaa yksi tai useampi check-lause yhdelle ja useammalle sarakkeelle. Toimii tietotekniikassa esiintyvän BOOLEAN:n tavoin. Voidaan rajata esimerkiksi luku alue, merkkien määrä, täytyy sisältää ja voidaan myös tarkistaa, jos kenttä saa jonkin tietyn arvon, sen seurauksena tarkastetaan myös jokin tietyn kentän arvo. NULL arvo voi jossain tapauksissa aiheuttaa ongelmia check-lauseelle koska tulkitsee sen arvoksi tosi näin ollen menee tarkastus läpi, vaikka ei pitäisi. (Microsoft 2022.)

3.3.7 Is not null-lause

Is not null-lause tarkoittaa, ettei kenttää saa jättää tyhjäksi tietoa lisätessä. Kentälle on pakko antaa jokin tietotyypin mukainen arvo, jos ei ole määritetty oletusarvoa kyseiselle sarakkeelle. SQL-tietokanta ilmoittaa virheestä, jos jätetään kenttä tyhjäksi. Is not null-lause kuuluu pääavaimen ominaisuuksiin. (Microsoft 2022.)

3.3.8 Character ja Variable character -tietotyypit

Character- ja variable character -tietotyypit ovat yleisimpiä tietotyyppiejä. Character (char) on kiinteämittainen merkkijono, kun taas variable character (varchar) on vaihtuvamittainen merkkijono. Molempiin saa tallennettua maksimissaan 8000 merkkiä. Eroa toiminnallisuuteen tulee tallennus tavassa. Jos char (100) ja varchar (100) määritellään sisältämään 100 merkkiä ja tallennetaan vain kaksi merkkiä: char tallettaa 2 merkkiä ja täyttää loput 98 merkki paikkaa, joten se vie kaikki ennalta määritellyn tilan tässä tapauksessa 100 tavua. Varchar toimii toisin, se tallettaa 2 merkkiä ja 2 tavua kuluu merkkijonon tiedon ylläpitoon, joten yhteensä vain 4 tavua kuluu tämän tiedon tallennukseen. Char-tietotyyppiä on hyvä käyttää vain, jos ennalta tiedetään, että merkkien määrä on aina tietty. Määrittelemällä tietotyyppi pystytään

estämään virheitä tietyissä tapauksissa rajaamalla merkkien määrää. (Microsoft 2022.)

3.3.9 Numeeriset-tietotyypit

Numeerist-tietotyypit Microsoftin (2022) mukaan ovat seuraavat:

1. Bigint on kokonaislukutyyppi SQL:ssä. BIGINT luku alue on -263 (-9 223 372 036 854 775 808) – 263–1 (9 223 372 036 854 775 807).
2. Dec(p/s) on desimaaliluku SQL:ssä. Desimaaliluvussa oleva p on numeroiden kokonaismäärä ja s on desimaaliosan numeroiden määrä.
3. Int on käytetyin kokonaislukutyyppi SQL:ssä. INT luku alue on -2^{31} (-2 147 483 648) to $2^{31}-1$ (2 147 483 647).
4. Money on valuutta tyyppi SQL:ssä. MONEY luku alue on -922 337 203 685 477,5808–922 337 203 685 477,5807.
5. Smallint on kokonaislukutyyppi SQL:ssä. SMALLINT luku alue on -215 (-32 768) – 215–1 (32 767).
6. Smallmoney on valuutta tyyppi SQL:ssä. SMALLMONEY luku alue on (-214 748,3648–214 748,3647).
7. Tinyint on kokonaislukutyyppi SQL:ssä. TINYINT luku alue on 0–255.

3.3.10 Päivämäärä-tietotyypit

Päivämäärä-tietotyypit Microsoftin (2022) mukaan ovat seuraavat:

1. Date-tietotyyppi on muodossa YYYY-MM-DD.
2. Time on muotoa hh:mm:ss.
3. Datetime on muotoa YYYY-MM-DD hh:mm:ss.
4. Current_Timestamp tallentaa sen hetkisen ajan, jolloin kysely timestamp tietotyyppistä tehdään. Tämä tallennetaan muodossa YYYY-MM-DD hh:mm:ss.

3.4 SQL Server Management Studio (SSMS)

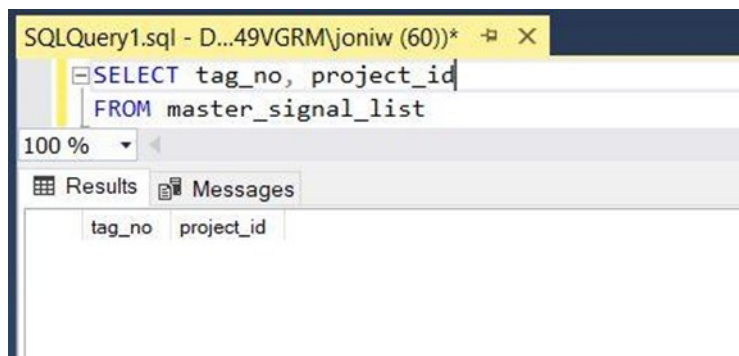
SSMS toimii kaikissa Microsoftin SQL -serveri, Azure SQL -tietokanta, Azure SQL hallitut instanssit, SQL-serveri Azure Virtuaalikone ja Azure Synapsi Analytiikka versioissa. SSMS on työkalu, jolla luodaan tauluja, näkymiä ja tietokannan rakenne kuvauksia tietokantaan, ja myös haetaan tietokannasta tietoa kyselyiden avulla. (Microsoft. 2022.)

3.5 DML

DML (Data Manipulation Language) on tarkoitettu tietokannan käsittelyyn ja tietojen hakemiseen SQL-tietokannassa tai SQL-serverissä. DML:llä voidaan suorittaa kohteeseen kyselyitä, lisäyksiä, päivityksiä ja poistoja. (Microsoft 2022.)

3.5.1 SELECT-komento

SELECT-komennolla haetaan tietoa kohteista ja niiden ominaisuuksista kyselyiden avulla. Voidaan hakea useasta eri kohteesta ja voidaan rajata ominaisuuksien näkyvyyttä kyselyn loppu tuloksessa. SELECT * hakee tiedon kaikilla ominaisuuksilla kohteesta. Kirjoittamalla SELECT jälkeen ominaisuuksien nimet, komento näyttää vain rajatun (kuva 16) loppu tuloksen. (Microsoft 2022.)



Kuva 16. Onnistunut kysely SQL Server management studiolla.

3.5.2 INSERT-komento

INSERT-komennolla lisätään uutta tietoa kohteeseen ja kohteen ominaisuuksille (kuva 17). Tietoa voi lisätä useamman rivin kerrallaan. (Microsoft 2022.)

```
SQL Copy  
  
INSERT INTO Production.UnitMeasure  
VALUES (N'FT', N'Feet', '20080414');
```

Kuva 17. INSERT-komento (Microsoft 2022).

3.5.3 UPDATE-komento

UPDATE-komennolla päivitetään kohteen jo olemassa olevia tietoja (kuva 18). UPDATE-komennolla päivitetään kohteen rivi kohtaisia tietoja rivi kerrallaan. (Microsoft 2022.)

```
SQL  
  
-- Uses AdventureWorks  
  
UPDATE DimEmployee  
SET FirstName = 'Gail'  
WHERE EmployeeKey = 500;
```

Kuva 18. UPDATE-komento (Microsoft 2022).

3.5.4 DELETE-komento

DELETE-komennolla poistetaan kohteista tietoja (kuva 19). DELETE-komennolla voidaan poistaa kaikki kohteen tiedot tai yksittäisiä rivejä kohteesta. (Microsoft 2022.)

```
SQL

DELETE TOP (20)
FROM Purchasing.PurchaseOrderDetail
WHERE DueDate < '20020701';
GO
```

Kuva 19. DELETE-komento (Microsoft 2022).

3.6 DDL

DDL (Data Definition Language) on tarkoitettu tietokannan rakenteen määrittelyyn. DDL komennoilla voidaan luoda uusia kohteita, muokata kohteita ja poistaa kohteita. (Microsoft 2022.)

3.6.1 CREATE-komento

CREATE-komennolla luodaan aina uutta tietokantaan. CREATE-komennolla voi luoda uuden tietokannan, kohteen (kuva 20) ja sen ominaisuudet, indeksejä, herättimiä, proseduurit, näkymiä ja monia muita tietokanta tason toimintoja. (Microsoft 2022.)

SQL

```
CREATE TABLE dbo.PurchaseOrderDetail
(
    PurchaseOrderID int NOT NULL
        REFERENCES Purchasing.PurchaseOrderHeader(PurchaseOrderID),
    LineNumber smallint NOT NULL,
    ProductID int NULL
        REFERENCES Production.Product(ProductID),
    UnitPrice money NULL,
    OrderQty smallint NULL,
    ReceivedQty float NULL,
    RejectedQty float NULL,
    DueDate datetime NULL,
    rowguid uniqueidentifier ROWGUIDCOL NOT NULL
        CONSTRAINT DF_PurchaseOrderDetail_rowguid DEFAULT (NEWID()),
    ModifiedDate datetime NOT NULL
        CONSTRAINT DF_PurchaseOrderDetail_ModifiedDate DEFAULT (GETDATE()),
    LineTotal AS ((UnitPrice*OrderQty)),
    StockedQty AS ((ReceivedQty-RejectedQty)),
    CONSTRAINT PK_PurchaseOrderDetail_PurchaseOrderID_LineNumber
        PRIMARY KEY CLUSTERED (PurchaseOrderID, LineNumber)
        WITH (IGNORE_DUP_KEY = OFF)
)
ON PRIMARY;
```

Kuva 20. CREATE-komento (Microsoft 2022).

3.6.2 ALTER-komento

ALTER-komennolla muokataan aina jotain jo olemassa olevaa tietokanta tason objektia. ALTER-komennolla voidaan muokata tietokantaa ja sen ominaisuuksia (kuva 21). (Microsoft 2022.)

SQL

```
CREATE TABLE dbo.doc_exa (column_a INT) ;
GO
ALTER TABLE dbo.doc_exa ADD column_b VARCHAR(20) NULL ;
GO
```

Kuva 21. ALTER-komento (Microsoft 2022).

3.6.3 DROP-komento

DROP-komennolla poistetaan jo olemassa olevaa tietokanta tason objektia. Esimerkiksi DROP TABLE taulunnimi (kuva 22) poistaa koko kohteen ja kaikkien sen ominaisuudet. (Microsoft 2022.)

```
SQL  
  
DROP TABLE ProductVendor1 ;
```

Kuva 22. DROP-komento (Microsoft 2022).

4 Excel ja VBA

4.1 Excel

Excel on Microsoftin taulukkolaskentaohjelmisto, jonka ensimmäinen versio julkaistiin Macintosh-tietokoneille vuonna 1985 ja Windowsille kaksi vuotta myöhemmin 1987.

Excelistä löytyy kaikki taulukkolaskennan perusominaisuudet. Excelistä löytyy tilasto, suunnittelu ja rahoitus alan tarpeisiin kaavioita, viivakaavioita, histogrammeja, ja pivot-taulukoita.

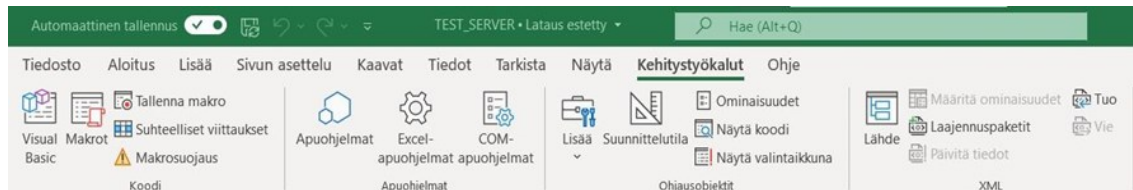
Excelissä rivit on numeroitu ja sarakkeet on kirjaimilla merkitty. Nämä yhdessä muodostavat solujen sijaintitiedon. (JavaTpoint 2021.)

Tietojen kelpoisuuden tarkistus (Data Validation) -toiminnolla voidaan estää tai hallita mitä käyttäjä saa syöttää valituille soluille tai valitulle alueelle. Tietojen kelpoisuuden tarkistuksella voidaan antaa käyttäjän syöttää soluun vain tekstiä tai numeroita, vain tietyn pituista tekstiä tai numeroita, numeroita jollain välillä, pienempi kuin, suurempi kuin, estää käyttäjää antamasta ennalta määriteltyjen arvojen ulkopuolelta drop-down list toiminnossa tai estää antamasta tiettyjen päivämäärien ja kellonajan ulkopuolelta. Tietojen kelpoisuuden tarkistuksella voidaan myös avata pop-up virhesanoma ikkunan. Virhe ikkuna vaihtoehtoja on varoitus, stop ja ilmoitus. Tietojen kelpoisuuden tarkistuksesta löytyy myös sanoma, joka näytetään, kun solu, jossa on aktiivisena tietojen kelpoisuus tarkastus toiminto. (JavaTpoint 2021.)

4.2 Excel kehitystyökalut

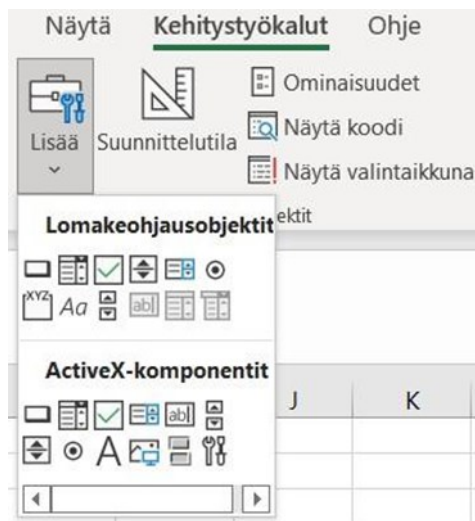
Excelin Windows versio tukee VBA-ohjelmointikieltä. Exceliin saa näkymään kehitystyökalut työkalurivin (kuva 23), josta löytyy makrot ja VBA painikkeet. Tärkeimmät tämän työn kannalta työkaluriviltä löytyvät ominaisuudet ovat: Visual Basic, makrot, lisää ja suunnittelutila. Visual Basic painike avaa VBA:n

koodi editorin. Makrot painike avaa listan Excel tiedoston sisältämistä VBA koodeista. Makrot sivun kautta voi lisätä painikkeille suoritettava koodin.

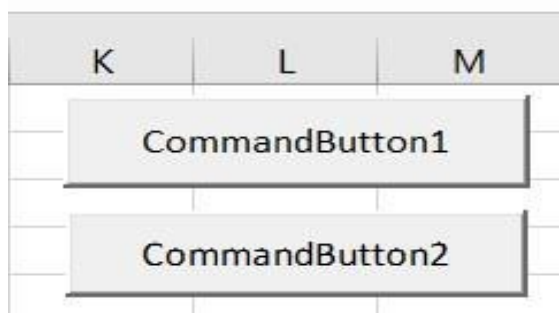


Kuva 23. Kehitystyökalut työkalurivi Excelissä.

Lisää-painikkeen (kuva 24) alta löytyy lomakeohjausobjektit ja ActiveX-komponentit. Molemmista kategorioista löytyy esimerkiksi painonapit (kuva 24), tekstikentät, valintapainike ja yhdistelmäruutu. (Microsoft 2022.)



Kuva 24. Lisää valikko Excelin kehitystyökalurivillä.



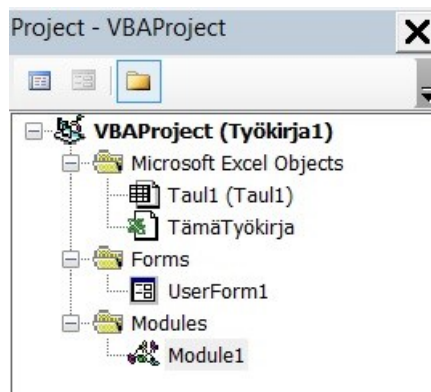
Kuva 25. Painikkeet Excel työkirjassa.

4.3 VBA

VBA on toteutus Microsoftin tapahtuma pohjaisesta ohjelmointikielestä Visual Basic 6.0: sta. VBA:ta päivitetään edelleen tukemaan uusia Office- ominaisuuksia, vaikka Microsoft ei enää tue tai päivitä Visual Basic 6.0: aa. VBA:n etuna on helppokäyttöisyys, prosessien automatisointi, omien makrojen nauhoitus, oman käyttöliittymän luominen. VBA lisäksi toimii useissa Office- paketin ohjelmissa. AutoCAD, SolidWorks ja monet muut suunnitteluohjelmit tukevat osittain VBA-ominaisuuksia. (Microsoft 2022.)

4.3.1 VBA:n projektihakemisto

VBA:n projekti hakemisto näyttää projektien hierarkian ja mitä projekti pitää sisällään. Projekti hakemisto sisältää kansiot (kuva 26) eri VBA komponenteille: Microsoft Excel -objektit, -lomakkeet ja -moduulit. Microsoft Excel -objekteista löytyy työkirja ja työkirjan sisältämä välilehti. Lomakkeista löytyy käyttäjälomake. Moduuleista löytyy moduuli1- lomakkeita, ja moduuleja voidaan lisätä ja muokata ominaisuuksia projekti hakemiston kautta. (Microsoft 2022.)

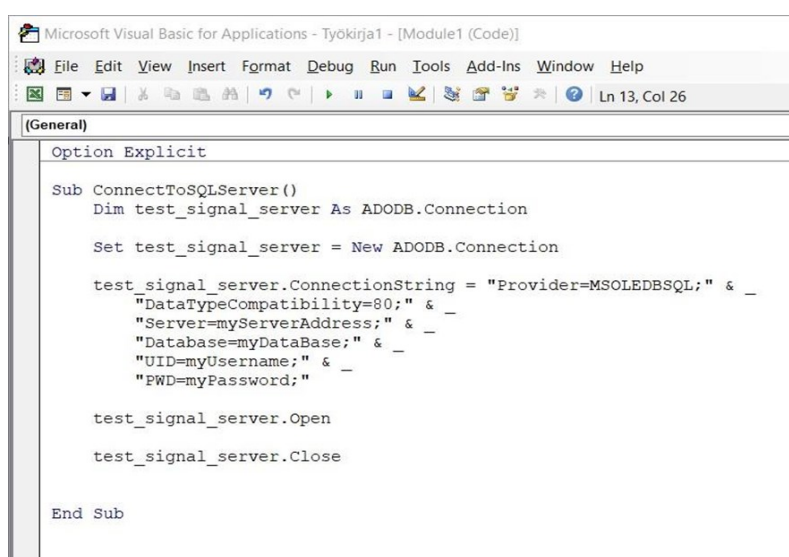


Kuva 26. Projekti näkymä VBA:ssa.

4.3.2 Sub-proseduuri

Sub-proseduuri suorittaa koodin Sub-proseduurin sisällä. Exit Sub poistuu Sub-proseduurista, jos ohjelmassa on erikseen niin määriteltä. Kun Sub-proseduuri

on suoritettu loppuun lopettaa End Sub -suorituksen. Sub-proseduuri ei voi palauttaa muuttujien arvoja. Sub-proseduurin sisällä voi olla kutsuja muihin Sub-proseduuriohjelmiin tai funktion-proseduuriohjelmiin. Sub-proseduurin (kuva 27) rakenne on seuraava: Sub-komento, jota seuraa käyttäjän antama nimi. Nimen perään lisätään sulkeet. Muuttujien määrittely ja asetus. Koodi, joka suoritetaan ja lopuksi suljetaan proseduurin End Sub -komennolla. (Microsoft 2022.)



```
Microsoft Visual Basic for Applications - Työkirja1 - [Module1 (Code)]
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Ln 13, Col 26
(General)
Option Explicit
Sub ConnectToSQLServer()
    Dim test_signal_server As ADODB.Connection
    Set test_signal_server = New ADODB.Connection
    test_signal_server.ConnectionString = "Provider=MSOLEDBSQL;" & _
        "DataTypeCompatibility=80;" & _
        "Server=myServerAddress;" & _
        "Database=myDataBase;" & _
        "UID=myUsername;" & _
        "PWD=myPassword;"
    test_signal_server.Open
    test_signal_server.Close
End Sub
```

Kuva 27. Yhdistäminen SQL-serverille VBA:n avulla.

4.3.3 Function-proseduuri

Function-proseduuri suorittaa koodin Function-proseduurin sisällä. Exit Function poistuu Function-proseduurista, jos ohjelmassa on erikseen niin määriteltä. Kun Sub-proseduuri on suoritettu loppuun lopettaa End Function -suorituksen. Function-proseduuri palauttaa muuttujien arvoja (kuva 28). (Microsoft 2022.)

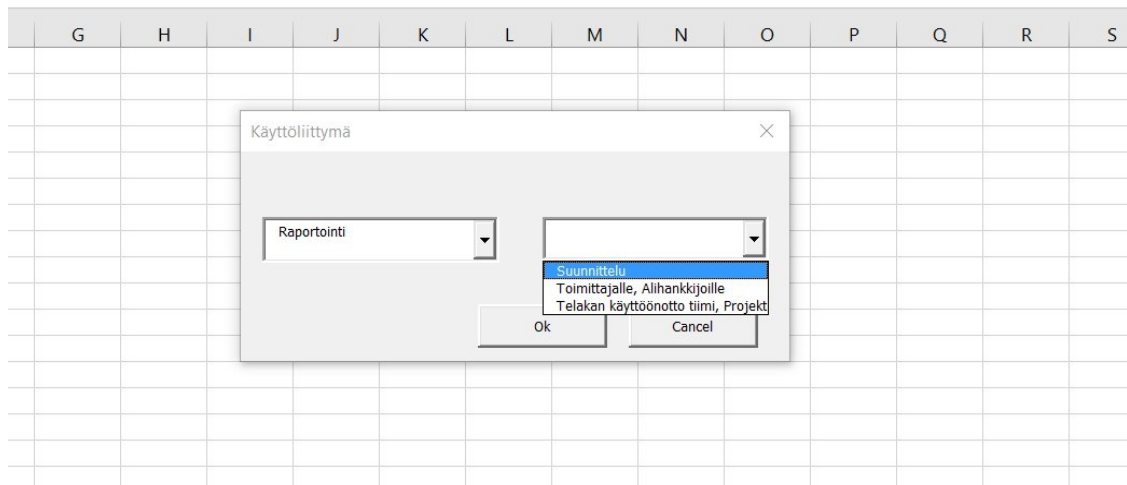
```
VB

Function BinarySearch(. . .) As Boolean
' . . .
' Value not found. Return a value of False.
If lower > upper Then
    BinarySearch = False
    Exit Function
End If
' . . .
End Function
```

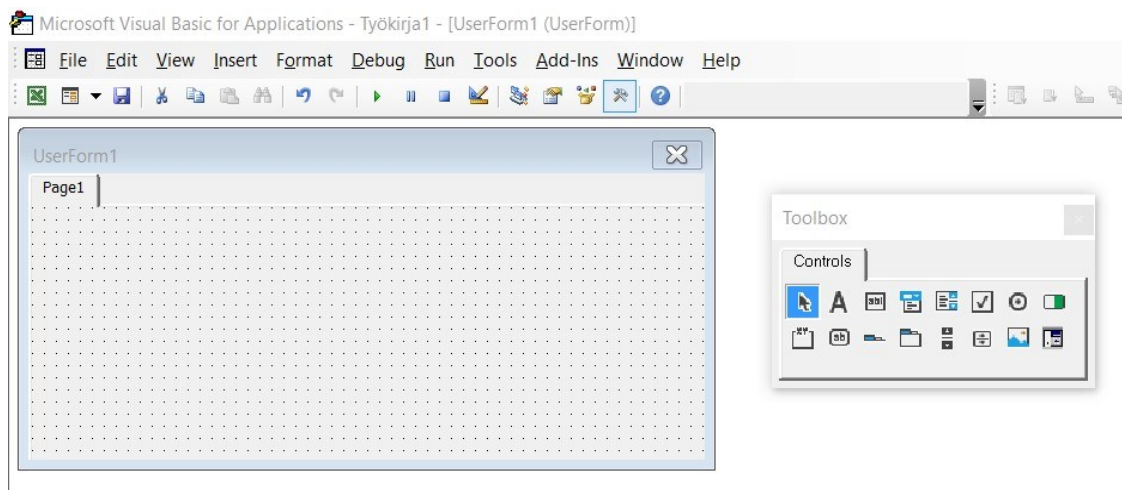
Kuva 28. Function-proseduuri VBA:ssa (Microsoft 2022).

4.3.4 VBA UserForm

VBA Userformia käytetään osana Excelin käyttöliittymää. Userformin voi avata koodin avulla tai lisätä napin Excelin välilehdelle (kuva 29), josta saa userformin avautumaan. Userform sisältää useita eri toimintoja vaihtoehtoja kuten tekstikentän, alas vetovalikon, valintanapit ja napit (kuva 30). (Microsoft 2022.)



Kuva 29. Userform avattuna Exceliin.



Kuva 30. VBA:n Userform.

4.3.5 Muuttujat

Kun määritellään proseduriin tason tai moduulin tason muuttujaa käytetään Dim lauseketta (kuva 31). Proseduri tason muuttuja asetetaan proseduurin sisään proseduurin alkuun. Moduulitason muuttuja asetetaan moduulissa moduulin yläosaan sille varatulle declarations osioon. Muuttujan näkyvyys määreen voi asettaa joko Public tai Private. Public näkyvyys määre näkyy moduulin ulkopuolelle, kun taas Private näkyvyys määre näkyy vain moduulissa. Dim vastaa näkyvyys määreeltään Private. Näkyvyys määreet asetetaan Dim lauseen tilalle. Muuttujan tietotyyppiä ovat: integer, string, variant, boolean, long, currency, byte, single, double, date, string*length ja object. (Microsoft 2022.)

VB	VB
<code>Dim strName As String</code>	<code>Public strName As String</code>

Kuva 31. Dim- ja Public-muuttujat (Microsoft 2022).

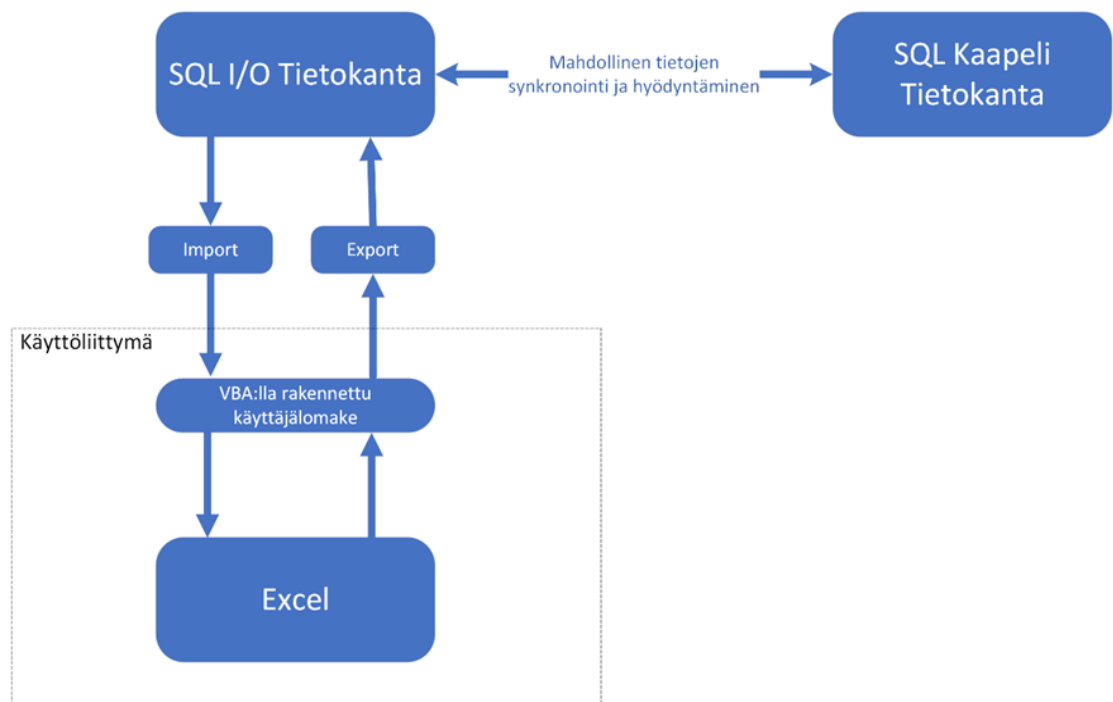
4.4 Makrot

Makrot ovat joko VBA-koodilla tai nauhoittamalla tehty toiminto tai toiminto sarja Excelissä. Nauhoitus tallentaa VBA-koodiksi jokaisen hiiren painalluksen ja näppäimen painalluksen. Makrojen avulla pystytään automatisoimaan toimintoja. Makron voi tallentaa osaksi työkirjaa tai Excelin omaan makro työkirjaan. (Tutorialspoint. 2022.)

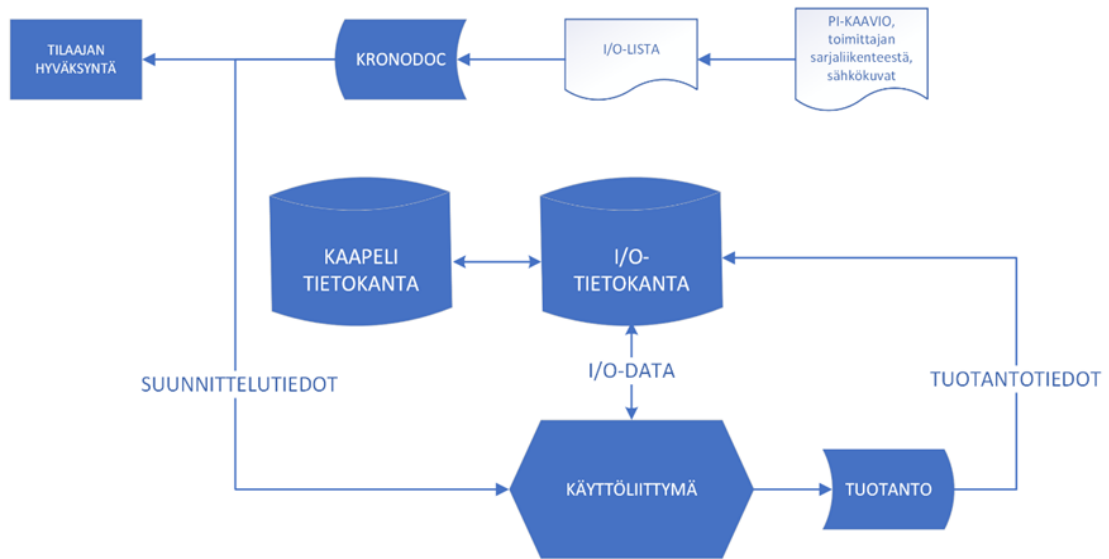
5 Nyt ja tulevaisuudessa

5.1 Käyttöliittymän rakentaminen

Tarkoituksena on rakentaa käyttöliittymä (kuva 32) VBA:n userformin avulla Exceliin. Käyttöliittymän avulla tietoja haetaan ja kirjoitetaan SQL-tietokantaan. Käyttöliittymän avulla käyttäjän ei tarvitse opetella SQL-tietokannan käyttöä vaan kaikki SQL-komennot tietokantaan (kuva 33) on hoidettu käyttöliittymän koodissa. Käyttöliittymän olisi tarkoitus olla helppokäyttöinen ja estää käyttäjää syöttämästä väärää tietoa aina kun se on mahdollista toteuttaa järkevästi.



Kuva 32. Käyttöliittymä Excelissä.



Kuva 33. I/O-data flow.

5.2 SQL-serverin edut ja haitat

5.2.1 Etuja

SQL-tietokanta mahdollistaa samanaikaisen työnteon ja nopean tallennuksen servereille. SQL-servereiden avulla voidaan myös tehdä yhteisiä projekteja Saksan Papenburgin telakan kanssa.

SQL-tietokannassa on monta hyvää tapaa ehkäistä virheitä kuten: tietotyypit, check-lause, is not null-lause ja herättimet. Virheiden ehkäisyyn voidaan myös laskea eheyssääntö. Kun aloitetaan suunnittelemaan tietokantaa noudattamalla eheyssääntöä, siten säästytään virheiltä myöhemmin tietokannassa.

SQL-tietokantaan voidaan jakaa myös käyttäjille roolin mukaan lukuoikeus tai kirjoitusoikeus sen mukaan, onko tarve muokata tietoja tai vain päästä katsomaan tietoja. Käyttäjien roolittaminen lisää tietoturva.

5.2.2 Haittoja

SQL-tietokannan ainoana huonona puolena on yhtäaikainen tietokantaan päivittäminen. Muutokset tietokantoihin tapahtuu Exceliin rakennetun VBA käyttöliittymän avulla. Jos käyttäjät päivittävät Excelin kautta tietoa samanaikaisesti tiedot muutoksesta menevät jonoon tietokannalle siirrettäväksi. Tietokannan jonosta tiedot päivitetään tietokantaan siinä järjestyksessä, kun ne ovat tulleet jonoon tietokannalle, jolloin vain viimeisin päivitys jää voimaan. Todennäköisyys tapahtua näin, on todella pieni.

5.2.3 SQL I/O -tietokannan rakentaminen

Tarkoitus olisi rakentaa I/O-tietokanta olemassa olevan kaapeli tietokannan yhteyteen. Osa kaapeli tietokannalle rakennetuista taulukoista voidaan käyttää I/O-tietokantaa rakennettaessa kuten keyfigure_log, projects ja hull_schedule - taulukot ja Master cable-list taulukosta cable_id voidaan käyttää I/O-tietokannassa.

5.3 Mahdolliset säästöt ja kustannukset tulevaisuudessa

5.3.1 Säästöjä virheiden ehkäisystä

Vähentämällä virheitä joudutaan palaamaan vähemmän jo käsitellyn asian pariin. Työtunteja menisi vähemmän virheiden syyn etsintään ja korjaukseen niin suunnittelussa kuin tuotannossa.

Tiedon päivittämisen yhteydessä lokiin tallentuisi aika ja tieto siitä, ketä on tietoa päivittänyt. Näin virheen sattuessa olisi helpompi selvittää syytä miksi päivitys tehtiin. Toinen vaihtoehto on käyttää revisiointi tietoja virheiden sattuessa. Revisiointi tietoon tallentuu käyttäjätunnus revisiointin tekijästä.

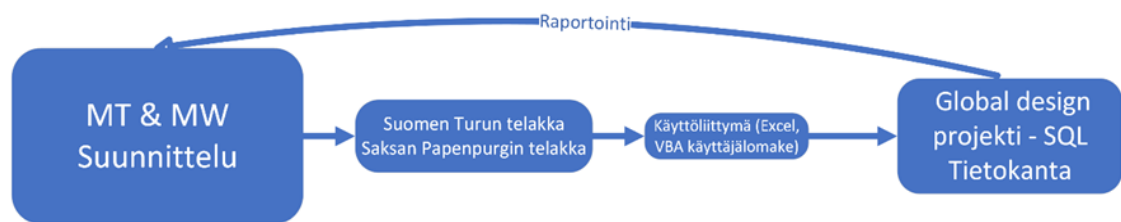
Virheet voivat vaikuttaa aikatauluihin ja sitä kautta tuleviin sakkoihin ja muihin kuluihin.

5.3.2 Säästöjä työtahokkuudesta

SQL I/O -tietokannan mahdollistama samanaikainen työnteko. Kun pystytään tekemään työtä samanaikaisesti ei tarvitse odotella tiedoston vapautumista. Järjestelmä kohtaisten listojen kokoaminen helpottuisi.

SQL I/O -tietokanta mahdollistaisi myös yhteiset projektit Meyerin telakoiden kesken (kuva 34). Voittaisiin jakaa työtehtäviä osaamisen perusteella ei sijainnin. Jos jossain telakalla on vähän hiljaisempaa, voitaisiin käyttää henkilöstö resursseja ja osaamista tehokkaasti hyödyksi.

Ajatus käyttää SQL:ää Global design ympäristössä Meyer Groupin toimipaikoista.

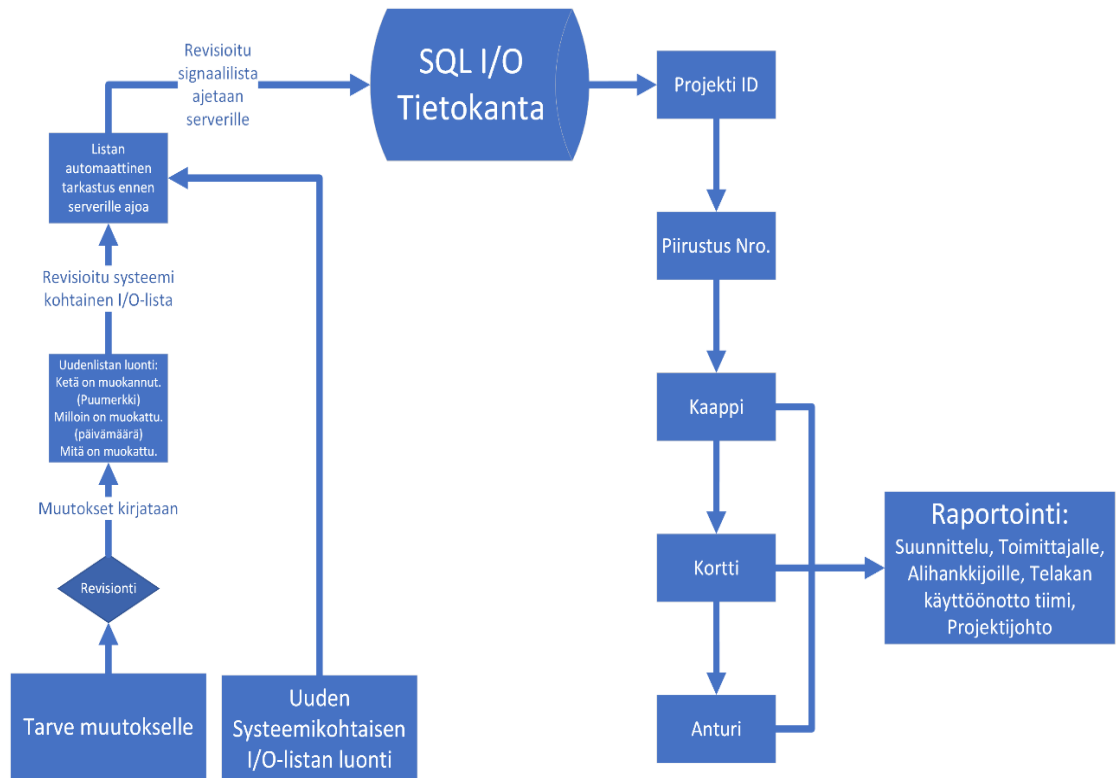


Kuva 34. Global design projekti.

5.3.3 Säästöjä Master I/O -listan kokoamisesta

SQL-tietokannan mahdollistama Master-listan automaattinen kokoaminen samalla kuin täytetään systeemi kohtaisia tietoja. Tässä tapauksessa voidaan käyttää hyväksi SQL-tietokannan näkymiä. Näkymillä voitaisiin luoda järjestelmä kohtaiset taulukot, josta tieto tallentuisi Master I/O -tietokantaan samalla kun järjestelmä kohtaisia tietoja täytetään.

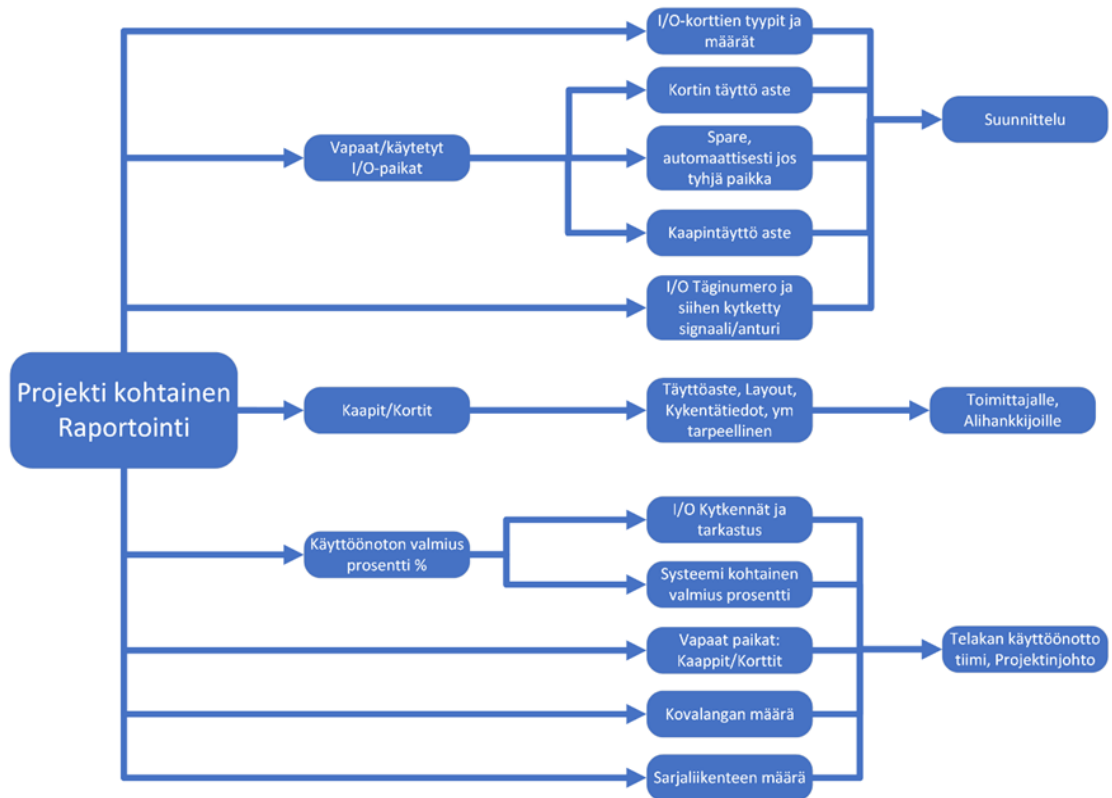
Säästönä tässä tavassa olisi kaikki se aika mikä käytetään Master I/O -listan kokoamiseen järjestelmä kohtaisista I/O-listoista.



Kuva 35. SQL I/O -tietokannan käyttö suunnittelussa.

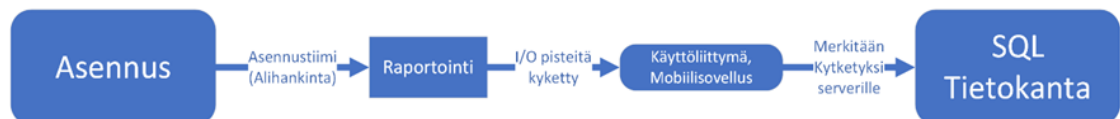
5.3.4 Säästöt raportoinnin helpottamisesta

SQL-tietokannan näkymien avulla voitaisiin luoda erilaisia valmiita raporttipohjia. SQL mahdollistaisi nopean raportoinnin eri tahoille kuten projektin johdolle, telakan käyttöönotto tiimille, toimittajille, alihankkijoille ja suunnittelulle (kuva 35). Kun pohjat ovat tehty ne olisivat aina saatavilla ja ajan tasalla. Raporttien tietojen päivitys tapahtuisi suoraan Master I/O -listan tiedoista.

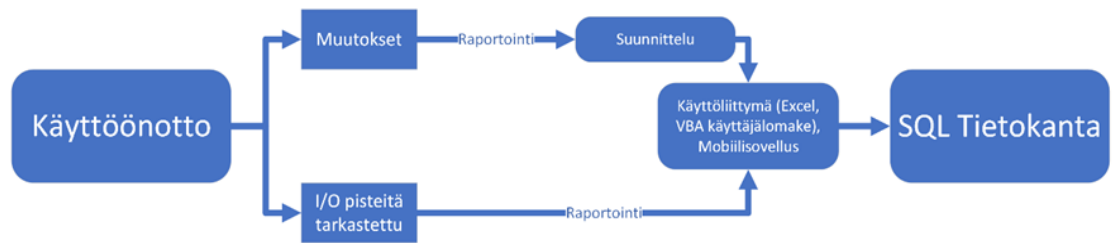


Kuva 36. Projekti kohtainen raportointi.

Kaapelinvetotiimille oli jo suunnitteilla raportoinnin työkaluksi mobiilisovellus. Sitä voitaisiin käyttää myös I/O-puolen asennuksessa (kuva 37) kuin myös käyttöönotossa (kuva 38) valmiiden kytkentöjen ja tarkastettujen I/O-pisteiden raportoinnissa.



Kuva 37. Asennuksen raportointi.



Kuva 38. Käyttöönoton raportointi.

5.3.5 Kustannukset SQL I/O -tietokannan rakentamisesta ja ylläpidosta

Kustannuksia puolestaan tulisi SQL I/O -tietokannan rakentamisesta. I/O-tietokannan voisi rakentaa jo olemassa olevan Master kaapelitietokannan rinnalle. Pohja on jo tehty, työn määrä ei ole niin iso verrattuna siihen, että jouduttaisiin rakentamaan kaikki alusta.

Excel käyttöliittymän I/O-puolen käyttöön joutuu tekemään melkein alusta, jos ei pystytä käyttämään Saksan Papenburgin Master I/O -listaa pohjana. Jo olemassa olevasta Excel-käyttöliittymästä ei voida käyttää kuin kirjautumisominaisuuden ja voisi antaa lukuoikeudet I/O-puolen tekijöille kaapeli tietokantaan.

Serverin lisä kapasiteetin tarve I/O-tietokannalle tuo yhden kustannus erän lisää. Karkea alustava arvio oli noin 2 €/kuukaudessa lisää jo olemassa olevaan noin 100 €/kuukausimaksuun. 100 €/kk piti sisällään jo kaapeli tietokannan ja muita toimintoja.

6 Yhteenveto

I/O-tietokantaa kannattaa lähteä kehittämään jo olemassa olevan kaapeli tietokannan yhteyteen. Hyvä pohja on jo tehty ja Meyerilta on omasta takaa osaamista I/O-tietokannan rakentamiseen. Kaapelitietokannasta voitaisiin käyttää suurinta osaa tauluista myös I/O-tietokannalle. Joitakin Master-kaapelitaulun tietoja voidaan yhdistää Master I/O -taulun kanssa.

Turun telakalla kehitettävää VBA käyttöliittymä Exceliin, jolla luetaan ja kirjoitetaan tietoja SQL-tietokannasta ja Master I/O -lista pohjaa käytetään jatkossa Master I/O -listojen luonnissa myös Saksan Papenburgissa. Yhteinen malli Master I/O -listalle ja VBA-käyttöliittymä Excelin kautta SQL-tietokantaan antavat edellytyksen yhteisien projektien suorittamiseen Saksan Papenburgin ja Turun telakoiden välillä.

Luomalla näkymiä SQL-tietokantaan voitaisiin rakentaa raporttipohjia valmiiksi SQL-tietokantaan. Raportit voisi saada käyttöliittymän kautta painamalla vain yhtä painiketta. Kehitteillä olevaa mobiiliraportoinnin työkalua kaapelinveto-tiimeille voisi jatkokehittää myös I/O-puolen käyttöön. Päivän päätteeksi voisi merkitä tehdyt kytkennät, jotta käyttöönotto tiimi tietäisi, mitkä kytkennät voisi tarkistaa. Myös käyttöönotolle voisi tehdä oman mobiili-tarkastus-listan siitä, mitä on jo tarkastettu. Samalla tieto valmiusasteesta olisi ajan tasalla.

SQL I/O -tietokannalla saavutettaisiin säästöjä ainakin työtunneissa ja työn tehokkuudessa, minkä lisäksi virheiden määrä vähenisi. Kuluja tulisi serverin ylläpidosta ja kertaluonteisia kuluja I/O-tietokannan rakentamisesta.

Todennäköisesti säästöt ja hyödyt olisivat suuremmat kuin serveristä syntyvät kulut.

Lähteet

Boyce, R., Codd, E., Chamberlin, D., Date, C. J. & Selinger, P. G. 2022. Relational Database. Viitattu 11.5.2022.

<https://www.ibm.com/ibm/history/ibm100/us/en/icons/reldb/>

Cong, V. N. H. & Huh, J-H. 2017. B+-tree construction on massive data with Hadoop. Cluster Computing. Vol 22, 1011–1021. Viitattu 27.11.2022.

https://www.researchgate.net/figure/Example-of-B-tree-with-branching-factor-equal-3_fig1_319966450

Dybka, P.2014. Chen's notation. Viitattu 19.11.2022.

<https://vertabelo.com/blog/chen-erd-notation/>

Dybka, P.2016. Crow's foot notation. Viitattu 19.11.2022.

<https://vertabelo.com/blog/crow-s-foot-notation/>

Hughes, A. 2019. Microsoft SQL Server. Viitattu 31.5.2022

<https://www.techtarget.com/searchdatamanagement/definition/SQL-Server>

JavaTpoint 2021. Data Validation in Excel. Viitattu 16.9.2022.

<https://www.javatpoint.com/data-validation-in-excel>

JavaTpoint 2021. MS Excel Definition. Viitattu 05.12.2022.

<https://www.javatpoint.com/ms-excel-definition>

Lahtonen, T. 2004. SQL Toolkit. Jyväskylä: Docendo Finland Oy

Meyer Turku Oy. 2022a. Lähes 300 vuotta kokemusta ja perinteitä. Viitattu 14.12.2022. <https://www.meyerturku.fi/fi/yritys/historia/index.jsp>

Meyer Turku Oy. 2022b. MEYER TURKU lyhyesti. Viitattu 12.5.2022.

https://www.meyerturku.fi/fi/meyerturku_com/shipyard/company/about_the_shipyard_1/about_the_shipyard.jsp

Microsoft 2022. Viitattu 14.12.2022.

<https://opdhsblobprod02.blob.core.windows.net/contents/264cf0df66eb4cbf86eac6677c504c4d/9f23772e46bd6fab7842c92c35a92127?skoid=2d004ef0-5468-4cd8-a5b7-14c04c6415bc&sktid=975f013f-7f24-47e8-a7d3-abc4752bf346&skt=2022-12-12T07%3A40%3A23Z&ske=2022-12-19T07%3A45%3A23Z&sks=b&skv=2021-10-04&sv=2021-10-04&se=2022-12-15T17%3A30%3A59Z&sr=b&sp=r&sig=a4VFVOUYexKiiV4RdFfqUbejM41RCHFwba1xizhcatk%3D>

Museovirasto 2009. Aurajokisuun satama-, telakka- ja teollisuusalue. Viitattu 29.5.2022. http://www.rky.fi/read/asp/r_kohde_det.aspx?KOHDE_ID=4095

Techopedia 2020. Input/Output (I/O). Viitattu 15.12.2022. [What is Input/Output \(I/O\)? - Definition from Techopedia](#)

Tutorialspoint 2022. Excel Macros – Overview. Viitattu 14.5.2022.

https://www.tutorialspoint.com/excel_macros/excel_macros_overview.htm#