



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Henri Poikkijoki

KOLMANNEN OSAPUOLEN KÄYTTÖOHJEI-
DEN TIETOKANNAN SUUNNITTELU JA TO-
TEUTUS

Tekniikka
2022

TIIVISTELMÄ

Tekijä	Henri Poikkijoki
Opinnäytetyön nimi	Kolmannen osapuolen käyttöohjeiden tietokannan suunnittelu ja toteutus
Vuosi	2022
Kieli	suomi
Sivumäärä	33 + 2 liitettä
Ohjaaja	Mikko Västi

Tämän opinnäytetyön tarkoituksena oli tutustua tietokannan perusteisiin ja rakenteisiin, suunnitella ER-kaavio ja perustaa uusi tietokanta käyttöohjeille. VEOlla oli käytössä vanhempi tietokanta, jota oli työlästä pitää yllä ja käyttöohjeiden valitseminen projektille oli aikaa vievää. Pää tavoitteena työssä oli se, että käyttöohjeiden ylläpito helpottuisi.

Tietokanta tehtiin Microsoft Access 2016-sovelluksella. Tietokannan luomisessa käytettiin apuna makroja, jotka on tehty VBA:lla. VBA on Microsoftin kehittämä ohjelmointikieli Office-sovelluksiin.

Suurin osa työhön käytetystä ajasta meni ohjelmointikielen opettelemisessa ja sen toiminnan sisäistämisessä. Opinnäytetyön tavoitetta ei saavutettu, joten tietokannan kehittämistä tullaan jatkamaan.

ABSTRACT

Author	Henri Poikkijoki
Title	Design and Implementation of Database for Third-party Manuals
Year	2022
Language	Finnish
Pages	33 + 2 appendices
Name of Supervisor	Mikko Västi

The purpose of this thesis was to get familiar with basics and structures of database, design an ER-model and create a new database for manuals. VEO had an old database in use, which was troublesome to sustain and choosing manuals for project was time consuming. The priority of the thesis was to ease the maintenance of the database.

The database was made with the Microsoft Access 2016 software. VBA macros were used to create the database. VBA is a programming language for Office software developed by Microsoft.

Most of the time was spent to learning and adopting the programming language. The aims set for the thesis were not met, therefore, development will continue in the future.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

KUVALUETTELO

LIITELUETTELO

LYHENTEET	8
1 JOHDANTO	9
2 VEO OY	10
3 TIETOKANTA	11
4 ER-KAAVIO.....	12
4.1 ER-kaavion luominen.....	12
4.2 Käsitteet.....	12
4.2.1 Kohdejoukot	12
4.2.2 Yhteydet	13
4.2.3 Attribuutit.....	15
4.2.4 Perus- ja viiteavaimet.....	15
5 VBA	17
6 ACCESS-TIETOKANTA	19
6.1 Taulukkojen luominen	19
6.2 Taulukkojenväliset yhteydet.....	22
7 TIETOKANNAN SIIRTÄMINEN EXCELISTÄ ACCESIIN.....	24
7.1 Tiedostopolun lukeminen.....	24
7.2 Valmistajien nimien tuonti Accessiin	24
7.3 Tietojen kerääminen INI-tiedostoista	26
7.3.1 Tiedostonimien lukeminen.....	26
7.3.2 Käyttöohjeiden otsikoiden lukeminen.....	27
7.3.3 Päivämäärien ja nimikirjaimien lukeminen.....	27
7.3.4 Valmistajan kotisivut.....	29

7.3.5	Kaikkien valmistajien ja käyttöohjeiden lukeminen Accessiin	30
8	JATKOKEHITYS	31
9	YHTEENVETO	32
	LÄHTEET	33
	LIITTEET	34

KUVALUETTELO

Kuva 1. Erilaiset yhteystyypit tietokannassa	13
Kuva 2. Yhteystyypit ER-kaaviossa	14
Kuva 3. Perus- ja viiteavaimet ER-kaaviossa	16
Kuva 4. Esimerkki makron nauhoittamisesta Excelissä	17
Kuva 5. Esimerkki nauhoitetusta makrosta VBA:ssa	18
Kuva 6. "tblManufacturer"-taulukko Accessissa	19
Kuva 7. Kuvakaappaus perusavaimen määrittelyn ohjeista	20
Kuva 8. Perusavaimen määrittely Accessissa	20
Kuva 9. "tblComponent"-taulukko Accessissa	21
Kuva 10. Viiteavaimen määrittely Accessissa	22
Kuva 11. Taulukkojenväliset yhteydet Accessissa	22
Kuva 12. Esimerkkikuva kansiorakenteesta	25
Kuva 13. FSO:n hyödyntäminen tiedostonimien lukemisessa	26
Kuva 14. INI-tiedoston rakenne "Valmistaja1"-kansioista	27
Kuva 15. INI-tiedosto luettuna "tblManual"-taulukkoon	28
Kuva 16. INI-tiedosto luettuna "tblCheck"-taulukkoon	29
Kuva 17. Kansiorakenne "Valmistaja1"-kansioista sekä valmistajan kotisivun esitys INI-tiedostossa	29
Kuva 18. Valmistajan kotisivut luettuna "tblHomepage"-taulukkoon	30

LIITELUETTELO**LIITE 1.** Valmis ER-kaavio**LIITE 2.** Accessin taulukot ja niiden yhteydet määriteltyinä

LYHENTEET

ER-kaavio	Entiteetti-relaatiokaavio
VBA	Microsoft Office-sovelluksien ohjelmointikieli
FSO	Objekti VBA:ssa
INI-tiedosto	Windowsin alustustiedosto
DAX	Microsoftin toiminnanohjausjärjestelmä

1 JOHDANTO

Tämä opinnäytetyö on tehty VEOLle. Työn tarkoituksena on pääasiassa perehtyä tietokannan rakenteeseen ja sen suunnitteluun, jota käydään läpi luvussa 4. Sain aiheen kun ilmaisin, että mielenkiintoa olisi etenkin ohjelmointiin liittyviin aiheisiin.

Aiheena tietokannat on mielestäni mielenkiintoinen ja haastava, koska aiempaa taustaa ohjelmoinnista en juurikaan omaa. Tästä syystä myös ohjelmointikielen opiskeluun meni suuri osa ajasta.

Työllä saadaan nopeutettua ja helpotettua käyttöohjeiden tietokannan ylläpitoa, koska nykyinen tietokanta on tehty Exceliin, joka ei ole optimi suuren tietokannan ylläpitämiseen. Tämä tietokanta tullaan siirtämään Accessiin, joka on suunniteltu tietokantojen ylläpitoa varten. Tietokannan siirtoa käsitellään luvussa 7. Tietokannasta saadaan myös käyttäjäystävällisempi Accessin avulla, jolloin käyttöohjeiden selaaminen ja etsiminen on selkeämpää ja helpompaa.

2 VEO OY

Vaasa Engineering Oy on perustettu itsenäisyyspäivänä 6.12.1989. Perustajina olivat Harri Niemelä ja Mauri Holma. Vaasa Engineering Oy:tä on kutsuttu alkuvuosista lähtien alkukirjaimien mukaisesti VEOksi, mutta Vaasa Engineering -nimestä luovuttiin vuonna 2012, jolloin tytäryhtiöt Vaasa Kojeistot ja Vaasa Service fuusioitiin emoyhtiöön.¹

VEOn pääkonttori ja tuotantotilat sijaitsevat Vaasan Runsorissa. VEolla on Suomessa toimipisteitä myös Tampereella, Paimiossa, Rovaniemellä ja Seinäjoella. Lisäksi tytäryhtiöitä sijaitsee Ruotsissa, Norjassa ja Isossa-Britanniassa. Työntekijöitä VEolla on yhteensä noin 500, joista suurin osa työskentelee Vaasan toimipisteellä.

2

¹ Sippola, H. 2017. Energiakeskittymän ytimessä – VEOn tarina

² VEO Oy. VEOn kotisivut. Viitattu 20.11.2021

3 TIETOKANTA

Tietokanta on tallennettu kokonaisuus joka sisältää joukon järjesteltyjä, yhteen aihealueeseen kuuluvia tietoja. Tietokanta on yleisesti tiedonhaku varten sähköisessä muodossa ylläpidettävä järjestelmä. Käytetyin tietokanta on relaatiotietokanta, jota myös tässä työssä käytetään. Relaatiotietokanta perustuu relaatiomalliin, jonka avulla kuvataan tietokannan rakenteita ja niiden välisiä yhteyksiä.

Tietokantoja ovat esimerkiksi puhelinluettelo ja jäsenrekisteri. Tietokannat voivat aluksi olla pelkkiä laskentataulukoita tai luetteloita, mutta kun luettelo kasvaa, tietoihin alkaa tulla päällekkäisyyksiä, jolloin tietokannan ylläpitämiseen tulee ongelmia. Vastaavilta ongelmilta voidaan välttyä, kun käytetään tietokannan hallintajärjestelmää, kuten Microsoft Accessia.

Hyvässä tietokannassa yhteenkuuluvat tiedot esitellään loogisesti, jolloin ne on helposti tavoitettavissa, niissä ei ole päällekkäisyyksiä ja käyttölomake on käyttäjystävällinen ja tarpeeksi helposti käytettävissä. Tietokannan ylläpidon kannalta on myös tärkeää, että tietoa on helppo ylläpitää, eikä samaa tietoa tarvittaessa pidä korvata moneen paikkaan.³

³ Oracle. What is database? Viitattu 21.3.2022.

4 ER-KAAVIO

Tietokannan suunnittelu aloitetaan suunnittelemalla ER-kaavio, jonka tarkoituksena on esittää tietokannan rakenne (Liite 1). Rakenne esitetään yleisellä tasolla pääpiirteittäin ja yksinkertaisesti. ER-kaavio pidetään myös ajan tasalla tietokantaa tehdessä jatkuvasti, jos mahdollisia muutoksia tai parannuksia tulee. ER-kaavion suunnittelu on tärkeää jo tässä vaiheessa, koska se auttaa ja nopeuttaa tietokannan rakentamista ja auttaa ymmärtämään tietokannan rakennetta. Suunnittelun tärkeys korostuu sitä mukaan, mitä laajempi kokonaisuus on kyseessä.

4.1 ER-kaavion luominen

ER-kaavio suunniteltiin pitkälti vanhan tietokannan perusteella, josta tieto jaoteltiin kohdejoukoiksi. Kohdejoukot asiakas ja kieli lisättiin, jotta saadaan kerättyä lisää tietoa, josta on hyötyä jatkoa ajatellen, myös käyttäjän näkökulmasta. Komponenttikohdejoukko lisättiin, jotta saadaan linkitettyä tietylle komponentille tietty manuaali.

Attribuutteja lisättiin suhteellisen paljon verrattuna vanhaan tietokantaan, koska Accessissa tiedon kerääminen, jäsentely ja esittäminen on helpompaa.

4.2 Käsitteet

Tietokannoissa on neljä peruskäsitettä, jotka esitellään luvuissa 4.2.1 – 4.2.4. Luvuissa käytetään esimerkkinä työssä käytettyjä käsitteitä, sekä jokaisen luvun lopussa myös käytännön esimerkkejä, jotta käsitteiden tarkoitusta saadaan yksinkertaistettua.

4.2.1 Kohdejoukot

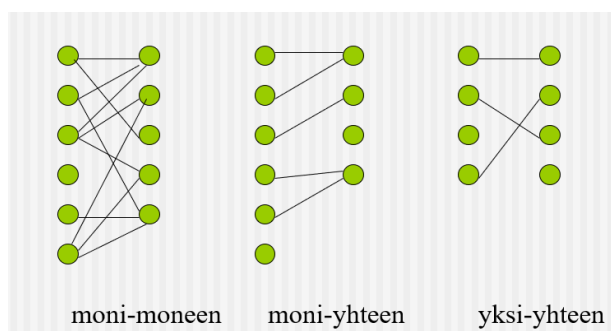
ER-kaavio sisältää kohdejoukkoja, jotka halutaan yksilöidä erillisenä kokonaisuutena. Tässä työssä kohdejoukoista manuaali on oleellisin. Kohdejoukot on esitetty tässä työssä suorakulmion muotoisina.

Käytetään käytännön esimerkkinä kohdejoukoista opiskelijaa, kurssia ja tutkintotutkintokirjastoja. Samoja esimerkkejä tullaan tarkastelemaan muissa käsitteissä.

4.2.2 Yhteydet

Yhteys on kahden tai useamman kohdejoukon välillä oleva riippuvuus, joiden avulla saadaan kohdejoukot linkitettyä toisiinsa loogisesti. Yhteydet ovat tietokannan rakenteelta hyvin tärkeitä, koska näillä pyritään välttämään ylimääräistä dataa.

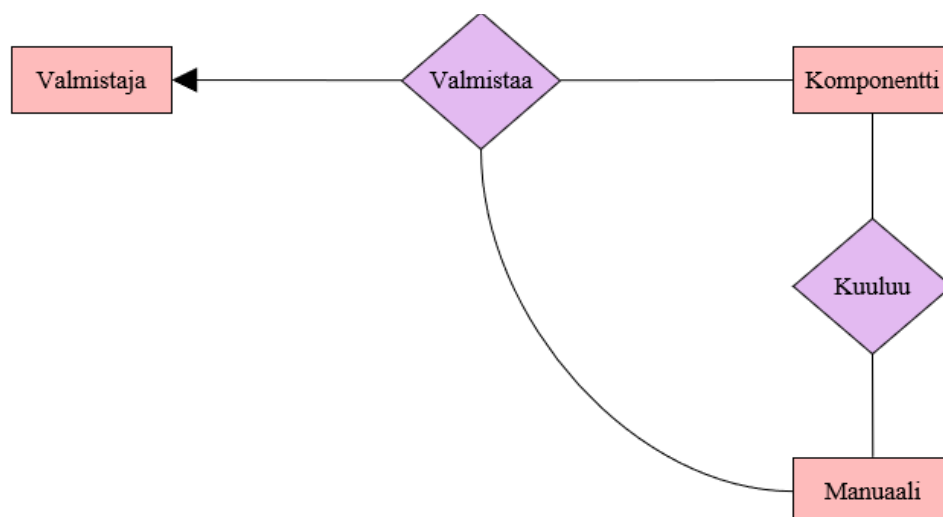
Kuvasta 1 nähdään kolme erilaista yhteystyyppiä: yksi-yhteen, moni-moneen ja moni-yhteen. Näistä yleisimpiä ovat moni-moneen ja moni-yhteen yhteydet, joita myös tässä työssä on käytetty.



Kuva 1. Erilaiset yhteystyytit tietokannassa

Moni-moneen yhteydet on kuvattu tässä työssä pelkillä viivoilla ja moni-yhteen yhteydet on kuvattu nuolella, joka saapuu ”yksi”-puolelle.

Esimerkkinä tässä työssä käytetyistä kohdejoukoista valmistaja-komponenttivali-nen yhteys on moni-yhteen, koska valmistajalla voi olla monta komponenttia, mutta komponentilla voi olla vain yksi valmistaja (**Kuva 2.**).



Kuva 2. Yhteystyypit ER-kaaviossa

Kuvasta 2 nähdään, että manuaali kohdejoukolla on myös yhteys valmistajaan, koska manuaali voi olla hallussa ilman fyysistä komponenttia. Kuvasta voidaan ottaa esimerkki myös moni-moneen yhteydestä kohdejoukkojen komponentti ja manuaali välillä. Eli komponentilla voi olla monta manuaalia ja manuaalissa voi olla myös monta eri komponenttia.

Kuvasta 2 nähdään myös kohdejoukkojen välille piirretyt yhteydet jokaiseen mukana olevaan kohdejoukkoon, jotka esitetään tässä työssä timantin muotoisina. Moni-moneen yhteyksistä tehdään myöhemmässä vaiheessa välitaulukkoja Accessiin.

Kohdejoukkojen opiskelija ja kurssi välillä yhteys on moni-moneen, koska opiskelijalla voi olla monta kurssia aktiivisena ja kurssille voi olla osallistunut monta opiskelijaa. Koska kohdejoukkojen opiskelija ja kurssi välinen yhteys on moni-moneen, näiden välille täytyy myös tehdä välitaulukko. Kohdejoukkojen opiskelija ja tutkintonimike välinen yhteys taas on yksi-moneen, koska yksi opiskelija voi opiskella vain yhtä tutkintonimikettä, mutta yhtä tutkintonimikettä voi opiskella monta opiskelijaa.

4.2.3 Attribuutit

Attribuutti on kohdejoukon ominaisuus, joka on tietokannan osalta kiinnostava ja oleellinen asia. Jokaisella attribuutilla täytyy olla jokin arvotyyppi, esimerkiksi numero- tai tekstikenttä. Attribuutit on esitetty tässä työssä ER-kaaviossa ellipsis muotoisina.

Opiskelijan attribuutteina voisi olla opiskelijanumero, etunimi, sukunimi. Kurssin attribuutteina ID, kurssin nimi ja opintopisteet. Tutkintonimikkeen attribuutteina opiskelijanumero ja tutkintonimike. Kohdejoukkojen opiskelijan ja kurssin välille välitaulukossa käytetään attribuutteina opiskelijanumeroa ja kurssin ID:tä.

4.2.4 Perus- ja viiteavaimet

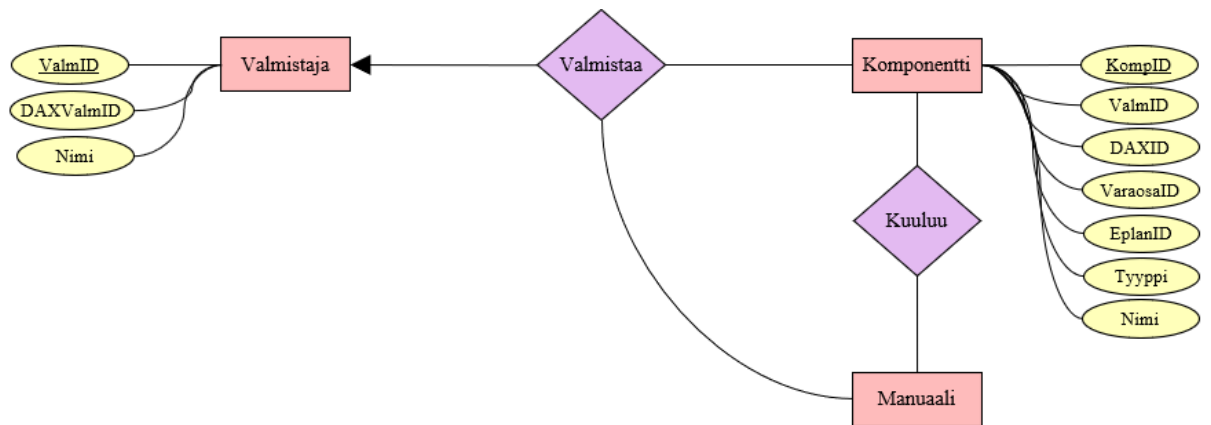
Jokaisen kohdejoukon attribuuteista valitaan perusavain tai viiteavain. Perusavaimen arvo on uniikki, jolloin arvoa ei voida esittää kahta kertaa samassa kohdejoukossa. Perusavaimella täytyy myös olla jokin arvo, eli sitä ei voi jättää tyhjäksi (Null).

Perusavain esitetään ER-kaaviossa alleviivattuna kuvassa 3. Kuvassa näkyy valmistaja kohdejoukon perusavain "ValmID" ja komponenttikohdejoukon perusavain "KompID". Myöhemmässä vaiheessa perusavain esitetään avaimen kuvalla (**Kuva 6.**).

Viiteavain puolestaan on kohdejoukon attribuutti, jonka arvo on sama kuin jonkun toisen kohdejoukon perusavain. Kuvasta 3 nähdään myös komponenttikohdejoukon attribuuteista viiteavain, joka on tässä tapauksessa "ValmID".⁴

⁴ InterviewBit. ER Model in DBMS. Viitattu 23.4.2022.

Opiskelija kohdejoukon perusavaimena voidaan käyttää opiskelijanumeroa, koska se on itsestään jo uniikki, eli kahta samaa numeroa ei voi olla. Kurssi kohdejoukon perusavaimena käytetään ID:tä. Tutkintonimike kohdejoukossa käytetään viiteavaimena opiskelijanumeroa. Välitaulukossa olevat ID:t ovat molemmat perusavaimia.



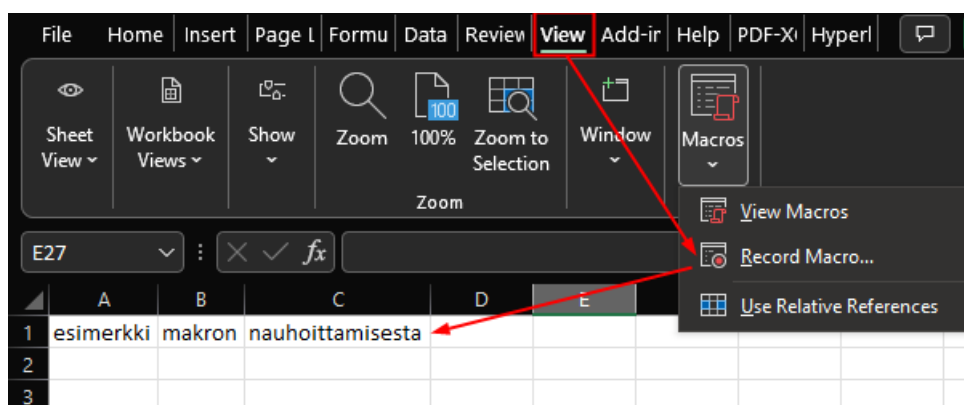
Kuva 3. Perus- ja viiteavaimet ER-kaaviossa

5 VBA

VBA on Microsoftin kehittämä ohjelmointikieli, joka sisältyy useimpiin Office-sovelluksiin. VBA:lla voidaan tehdä monipuolisesti käyttäjän määrittämiä toimintoja, helpottamaan sovelluksen haluttua käyttötapaa. Kaikissa Office-sovelluksissa VBA näkymän saa auki painamalla näppäinyhdistelmää ”Alt+F11”.

VBA:lla on mahdollista käyttää myös valmiita makroja tai käyttäjä voi itse luoda makron. Valmiit makrot ovat hyviä yksinkertaisten toimintojen suorittamiseen, mutta näissä toiminnot ovat hyvin rajattuja.⁵

Jos makrojen tekemisestä ei ole aikaisempaa kokemusta, yksinkertaisin tapa ymmärtää niitä, on nauhoittaa oma makro (**Kuva 4.**).

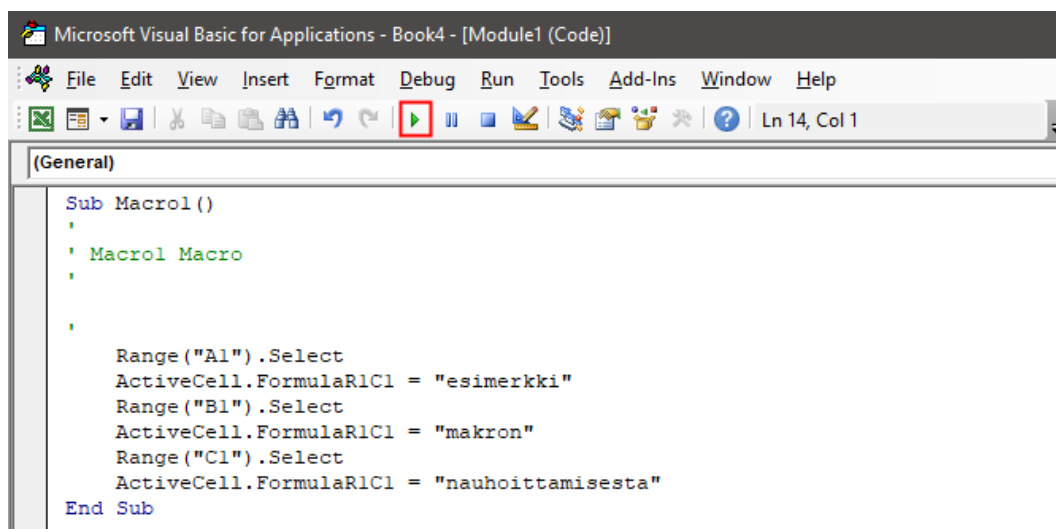


Kuva 4. Esimerkki makron nauhoittamisesta Excelissä

Kun nauhoitus on aloitettu, voidaan sen jälkeen tehdä mitä tahansa halutaan nauhoittaa. Kuvan 4 esimerkissä nauhoituksen jälkeen on kirjoitettu käsin tekstiä A1,

⁵ Microsoft Docs. Getting started with VBA in Office. Viitattu 26.2.2022

B1 ja C1 soluihin. Tämän jälkeen mennään "Macros"-valikkoon uudestaan ja valitaan "Stop Recording". Nyt voidaan avata VBA-näkymä, johon Excel on tehnyt makron, joka on nauhoitettu **(Kuva 5.)**.



```
Microsoft Visual Basic for Applications - Book4 - [Module1 (Code)]
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Ln 14, Col 1
(General)
Sub Macrol ()
'
' Macrol Macro
'
Range ("A1").Select
ActiveCell.FormulaR1C1 = "esimerkki"
Range ("B1").Select
ActiveCell.FormulaR1C1 = "makron"
Range ("C1").Select
ActiveCell.FormulaR1C1 = "nauhoittamisesta"
End Sub
```

Kuva 5. Esimerkki nauhoitetusta makrosta VBA:ssa

Painamalla "Run"-nappia, nauhoitettu makro kirjoittaa A1, B1 ja C1 soluihin automaattisesti sen, mitä nauhoitettiin. Yhdistelemällä ja muuttamalla nauhoitettuja makroja, on helpompi käsitellä, miten VBA toimii ja sitä kautta rakentaa monimutkaisempia makroja.

Makrojen nauhoittaminen ei ole kuitenkaan mahdollista Microsoft Accessissa, koska se sisältää monimutkaisempia toimintoja, joita ei ole mahdollista nauhoittaa niin, että sovellus sen ymmärtäisi.

6 ACCESS-TIETOKANTA

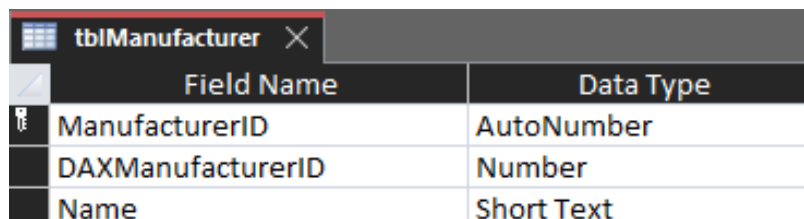
Kun ER-kaavio on saatu suunniteltua ja mallinnettua niin pitkälle kuin mahdollista, aletaan suunnittelemaan ja rakentamaan itse tietokantaa. Tässä työssä tietokanta tehdään Microsoft Access 2016-sovelluksella. Accessiin mallinnetaan tietokanta ER-kaavion mukaisesti, mutta yksityiskohtaisemmin.

6.1 Taulukkojen luominen

Ensimmäinen vaihe tietokannan luomisessa on taulukoiden luominen, jotka vastaavat ER-kaaviossa kohdejoukkoja.

Otetaan ER-kaavion kohdejoukoista valmistaja esimerkkinä Accessin taulukkona, joka on nimetty Accessissa nimellä "tblManufacturer" kuvassa 6.

"DAXManufacturerID"-kenttä on olemassa, koska valmistajien ID-numerot tullaan hakemaan DAXista myöhemmässä vaiheessa Accessiin.



Field Name	Data Type
ManufacturerID	AutoNumber
DAXManufacturerID	Number
Name	Short Text

Kuva 6. "tblManufacturer"-taulukko Accessissa

Taulukkoon on tuotu kaikki samat tiedot sarakkeisiin jotka olivat myös ER-kaaviossa, mutta taulukoiden nimet ja niiden attribuutit on muutettu englanniksi käytettävyyden ja yhtenäisyyksien vuoksi. Jokaiselle attribuutille täytyy myös antaa jokin arvo Accessissa. Taulukon "tblManufacturer" arvot näkyvät kuvasta 6 "Data Type" sarakkeesta.

Taulukosta nähdään myös aiemmin mainittu perusavain, joka esitettiin ER-kaaviossa alleviivaten, mutta Accessissa avaimen kuvalla, ”ManufacturerID”-kentän vasemmalla puolella.

- **To create a one-to-many relationship** The field on the one side (typically the primary key) of the relationship must have a unique index. This means that the **Indexed** property for this field should be set to **Yes (No Duplicates)**. The field on the many side should *not* have a unique index. It can have an index, but it must allow duplicates. This means that the **Indexed** property for this field should be set to either **No** or **Yes (Duplicates OK)**. When one field has a unique index, and the other does not, Access creates a one-to-many relationship.

Kuva 7. Kuvakaappaus perusavaimen määrittelyn ohjeista

Perus- ja viiteavaimet, joita käytetään moni-yhteen yhteyksissä, rakennettiin Microsoft Accessin ohjeiden mukaisesti (**Kuva 7.**)⁶

Perusavaimen arvo indeksoitiin, jolloin siitä tulee uniikki, eikä päällekkäisyyksiä sallita, joten taulukossa ei voi olla samaa arvoa kahta kertaa. Arvolle laitetaan myös valinta, että se vaaditaan, jolloin arvo ei voi jäädä tyhjäksi. Kuvasta 8 nähdään perusavaimen määrittely.

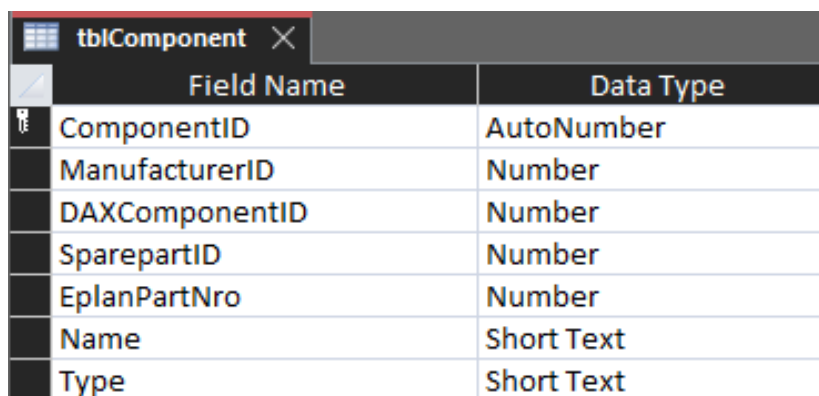
Required	Yes
Indexed	Yes (No Duplicates)

Kuva 8. Perusavaimen määrittely Accessissa

⁶ Microsoft Support. Create, edit or delete a relationship. Viitattu 28.6.2021.

Vertailun vuoksi tarkastellaan myös komponenttikohdejoukkoa, joka on vastavasti nimetty Accessissa ”tblComponent”-taulukoksi.

”ComponentID”-attribuutilla halutaan yksilöidä Access-tietokantaan tuodut komponentit. ”ManufacturerID:llä” halutaan yksilöidä valmistajat jotka lisätään Access-tietokantaan. ”DAXComponentID”-kenttään tuodaan DAXissa oleva ID. ”SparepartID”-kenttään tuodaan varaosatiekannasta komponentin ID. ”EplanPartNro”-attribuutille ei tällä hetkellä ole tarvetta, mutta se lisätään jo tässä vaiheessa tulevaisuuden tarpeita ajatellen. ”Name”-kenttä on komponentin nimi ja ”Type”-kenttä on komponentin tyyppi.



Field Name	Data Type
ComponentID	AutoNumber
ManufacturerID	Number
DAXComponentID	Number
SparepartID	Number
EplanPartNro	Number
Name	Short Text
Type	Short Text

Kuva 9. ”tblComponent”-taulukko Accessissa

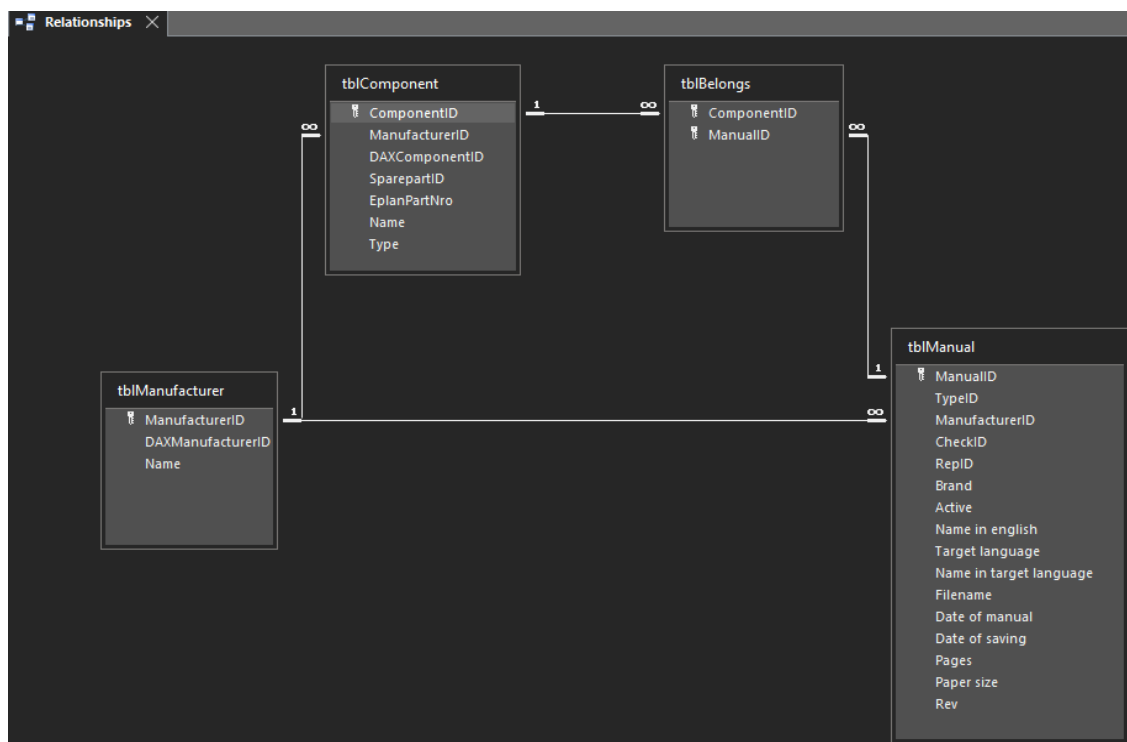
”tblComponent”-taulukon viiteavaimena on käytetty ER-kaaviossa vastaavaa attribuuttia ”ValmID”, joka nähdään kuvasta 9 ”ManufacturerID”. Viiteavaimen arvo on indeksoitu kuten perusavaimenkin, mutta päällekkäisyydet sallitaan, jotta saadaan rakennettua moni-yhteen yhteys valmistaja- ja komponenttitaulukoiden välille. Viiteavaimelle laitetaan perusavaimen tavoin myös valinta, että se vaaditaan **(Kuva 10.)**.

Required	Yes
Indexed	Yes (Duplicates OK)

Kuva 10. Viiteavaimen määrittely Accessissa

6.2 Taulukkojenväliset yhteydet

Taulukkojenväliset yhteydet esitetään Accessissa yksityiskohtaisemmin kuin ER-kaaviossa. Jokainen moni-moneen yhteys vaatii myös välitaulukon. Accessissa yhteyksien merkintä tulee automaattisesti, riippuen yhteystyypistä. ”Yksi”-puolen merkintä Accessissa on loogisesti numero yksi ja ”moni”-puolta merkitään ∞-merkillä (**Kuva 11.**).



Kuva 11. Taulukkojenväliset yhteydet Accessissa

Kuvasta 11 nähdään samat kohdejoukot kuin ER-kaaviossa (**Kuva 3.**), mutta Accessiin tehtyinä. Kuvasta nähdään myös "tblBelongs"-välitaulukko, joka vastaa ER-kaaviossa kohdejoukkojen välisiä moni-moneen yhteystyyppejä. Välitaulukko vaaditaan, jotta Accessiin saadaan luotua moni-moneen yhteys.

7 TIETOKANNAN SIIRTÄMINEN EXCELISTÄ ACCESSIIN

Tällä hetkellä olemassa oleva käyttöohjeiden tietokanta on Excelissä. Excel ei ole optimaalinen sovellus tietokantojen hallitsemiseen, varsinkin kun käyttöohjeita tulee jatkuvasti lisää ja niitä päivitetään tarpeen mukaan ajankohtaisiksi.

Tässä työssä on käytetty Excelissä olevaa VBA-koodia pohjana, jota on lähdetty muokkaamaan yhteensopivaksi Accessin kanssa.

Vaikka Office-sovellukset käyttävät samaa VBA-ohjelmointikieltä, niin eroavaisuuksia silti on paljon, jotka pitää korjata sopivaksi Accessiin, esimerkiksi Excelissä viitataan VBA:ssa tiettyyn välilehteen ja välilehdessä tiettyyn soluun, niin Accessissa täytyy viitata tiettyyn taulukkoon ja taulukossa tiettyyn kenttään.

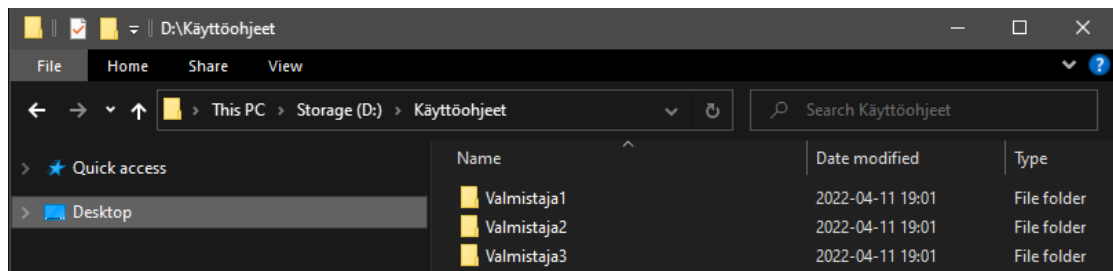
VBA:ssa, niin kuin muissakin ohjelmointikielissä, on hyvä kommentoida yksinkertaisesti jokaista makroa joka tehdään, jotta VBA:ta voi kuka tahansa katsoa ja ymmärtää, mitä missäkin kohtaa tapahtuu. Tämä helpottaa myös mahdollisten vikojen etsimisessä tai makrojen optimoinnissa, kun aikaa ei tarvitse käyttää makron tai tietyn rivin etsimiseen, vaan kommentista näkee suoraan mitä kyseinen makro tai rivi tekee. Tietokannan toimintaa lähdettiin testaamaan pienin paloin kerralla testiversiossa.

7.1 Tiedostopolun lukeminen

Kaikki tarvittavat tiedot, mitä Accessiin luetaan, ovat yhden kansion alla. Tämän kansion tiedostopolku lisätään Accessin "tblSettings"-taulukon kenttään "FileLocation", jotta VBA:ssa voidaan käyttää yhtä muuttujaa. Näin vältetään siltä, että tietoa ei tarvitse korjata moneen paikkaan, jos esimerkiksi tiedostopolku vaihtuu.

7.2 Valmistajien nimien tuonti Accessiin

Ensimmäisenä Accessiin lähdettiin tuomaan komponenttien valmistajien nimiä, jotka on kaikki kerätty yhteen kansioon (**Kuva 12.**).

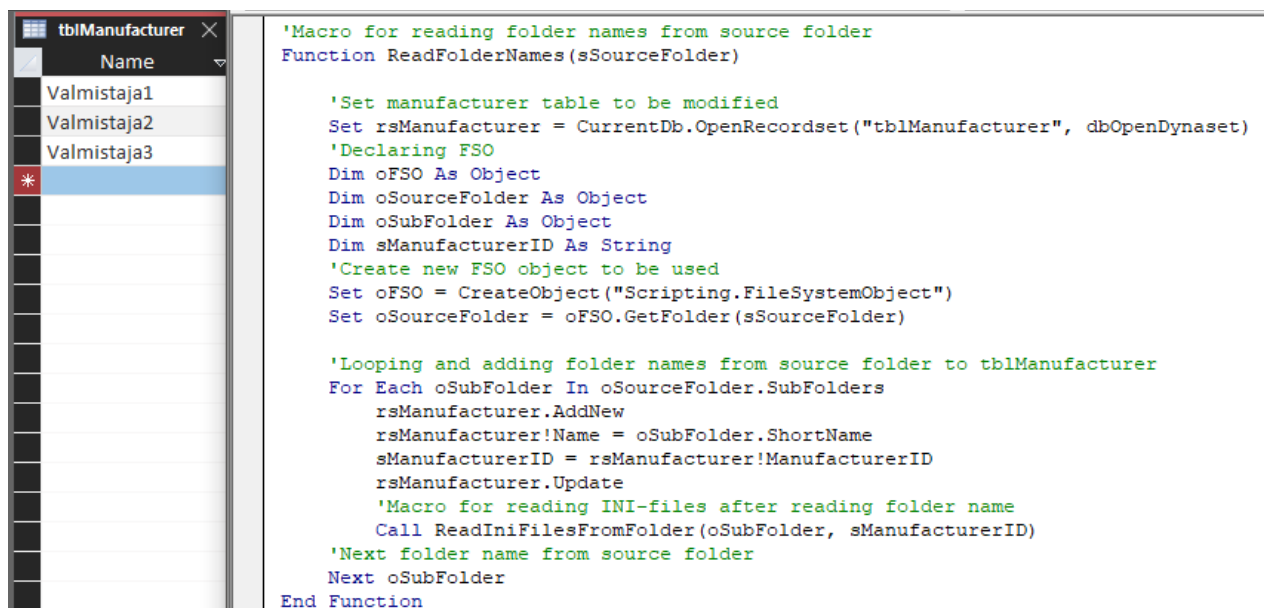


Kuva 12. Esimerkkikuva kansiorakenteesta

Valmistajien nimien lukemisessa on käytetty FSO:ta, joka on valmis objekti Microsoftin VBA-kirjastossa. FSO:lla on mahdollista tehdä monenlaisia toimintoja tiedostoihin liittyen, esimerkiksi lukea tiedostopolku, luoda tiedostoja, sekä lukea tiedostonimiä, jota tässä työssä käytettiin.⁷

Kuvan 13 makrossa FSO:lla luetaan kaikkien alikansioiden nimet "SourceFolder"-muuttujan perusteella ja lisätään ne yksi kerrallaan "tblManufacturer"-taulukon kenttään "Name". Tässä vaiheessa otetaan myös talteen "ManufacturerID", jota tarvitaan myöhemmässä vaiheessa.

⁷ Microsoft Docs. FileSystemObject object. Viitattu 12.4.2022.



Kuva 13. FSO:n hyödyntäminen tiedostonimien lukemisessa

7.3 Tietojen kerääminen INI-tiedostoista

INI-tiedostosta luettavat tiedot käydään läpi luvuissa 7.3.1 – 7.3.4. INI-tiedostot on valmiiksi olemassa ja niitä ylläpidetään jatkuvasti. Tällä saadaan hyöty, että tieto pitää ainoastaan lisätä INI-tiedostoon ja makrot lisäävät tiedon Accessiin, kunhan formaatti pidetään kuvan 14 mukaisena.

7.3.1 Tiedostonimien lukeminen

Tiedostonimien lukemisessa käytettiin pohjana vanhan Excel-tietokannan VBA-makroa, johon muokattiin rivit, jotka vanhassa tietokannassa viittasivat Excelissä tiettyyn soluun niin, että ne viittaavat Accessissa tiettyyn taulukkoon ja taulukossa tiettyyn kenttään. Vanhaa makroa muokattiin myös hieman siistimmäksi ja selkeämmäksi.

Tiedostonimet luettiin suoraan olemassa olevista INI-tiedostoista. INI-tiedosto rakentuu kuvan 14 mukaisesti, jossa keltaisella on merkattu tiedostonimi, eli "tiedostonimi_1.pdf"

VBA-makrossa tiedostonimi luetaan muuttujaan, alusta siihen asti kunnes makro huomaa symbolin ”=”. Tiedostonimi kirjoitetaan Accessissa ”tblManual”-taulukon kenttään ”FileName” (**Kuva 15.**).

```
tiedostonimi_1.pdf=1 Kayttoohjeen otsikko; 2022-04-25; 2022-04-24; 2022-04-26; HP
tiedostonimi_2.pdf=2 Kayttoohjeen otsikko; 2022-04-25; 2022-04-24; 2022-04-26; HP
tiedostonimi_3.pdf=3 Kayttoohjeen otsikko
```

Kuva 14. INI-tiedoston rakenne ”Valmistaja1”-kansioista

7.3.2 Käyttöohjeiden otsikoiden lukeminen

Käyttöohjeiden otsikoiden lukemisessa käytettiin samaa tyyliä kuin tiedostonimien lukemisessakin, eli ne saatiin suoraan INI-tiedostosta. Kuvasta 14 nähdään vihreällä merkattuna käyttöohjeiden otsikko, eli ”1 Kayttoohjeen otsikko”.

VBA-makrossa käyttöohjeiden otsikko luetaan muuttujaan, siihen asti kunnes makro huomaa ensimmäisen puolipilkun. Tästä muuttujasta vähennetään tiedostonimen muuttuja, jolloin saadaan pelkästään otsikko muuttujaksi. Otsikko kirjoitetaan Accessissa ”tblManual”-taulukon kenttään ”NameInEnglish” (**Kuva 15.**).

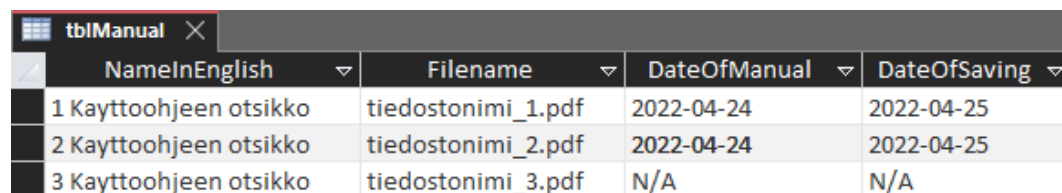
7.3.3 Päivämäärien ja nimikirjaimien lukeminen

INI-tiedoston rakenteeseen on merkattu käyttöohjeen tallennuspäivämäärä järjestelmään syaanin värisellä, käyttöohjeen päivämäärä magentan värisellä ja harmaalla on merkattu tarkistuspäivämäärä. Viimeisenä on merkattu nimikirjaimilla, kuka on lisännyt käyttöohjeen järjestelmään (**Kuva 14.**).

Eli, kuvan 14 esimerkin mukaisesti ”1 Kayttoohjeen otsikko” on tallennettu järjestelmään 25.4.2022, käyttöohjeessa oleva päivämäärä on 24.4.2022 ja 26.4.2022 on tarkistettu onko kyseisestä käyttöohjeesta uudempaa versioita saatavilla.

VBA:ssa päivämäärien ja nimikirjaimien lukemisessa käytettiin makroa, joka laskee merkkijonojen pituudet ja vähentää merkkijonosta puolipilkkuun asti merkkejä. Jos päivämäärä(t) tai nimikirjaimet puuttuvat, kirjoitetaan Accessin vastaaviin kenttiin "N/A". Toinen hyvä vaihtoehto lukea merkkijonoa, kun puolipilkku toisuu, on split-funktio, jolla tämän toiminnon saisi vähän selkeämmäksi.⁸

Käyttöohjeen tallennuspäivämäärä järjestelmään kirjoitetaan Accessissa "tblManual"-taulukon kenttään "DateOfSaving", käyttöohjeeseen merkattu päivämäärä kirjoitetaan saman taulukon kenttään "DateOfManual" (**Kuva 15.**).



	NameInEnglish	Filename	DateOfManual	DateOfSaving
1	Kayttoohjeen otsikko	tiedostonimi_1.pdf	2022-04-24	2022-04-25
2	Kayttoohjeen otsikko	tiedostonimi_2.pdf	2022-04-24	2022-04-25
3	Kayttoohjeen otsikko	tiedostonimi_3.pdf	N/A	N/A

Kuva 15. INI-tiedosto luettuna "tblManual"-taulukoon

Kun tässä vaiheessa kaikki tarvittava on tuotu "tblManual"-taulukoon, niin lisätään myös "ManufacturerID", joka luettiin luvussa 7.2. ID lisätään sen takia, jotta myöhemmässä vaiheessa saadaan linkitettyä tiedot yhteen, että kenen valmistajan käyttöohje on kyseessä.

Tarkistuspäivämäärä kirjoitetaan "tblCheck"-taulukon kenttään "DateOfCheck" ja nimikirjaimet saman taulukon kenttään "Initials" (**Kuva 16.**).

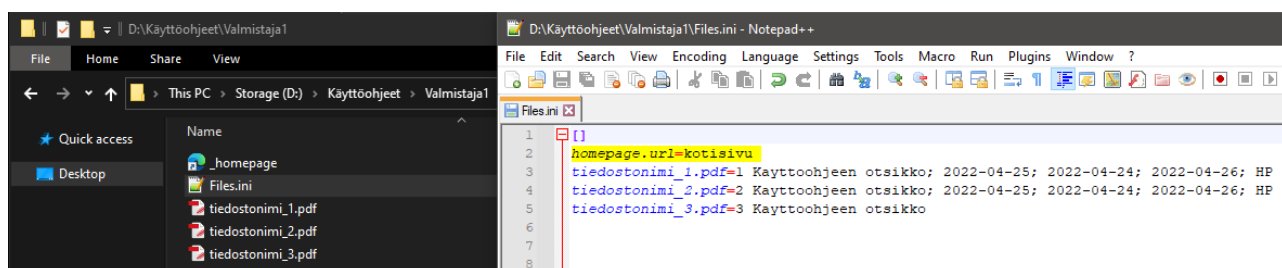
⁸ Microsoft Docs. Split function. Viitattu 23.4.2022.

tblCheck	Initials	DateOfCheck
	HP	2022-04-26
	HP	2022-04-26
	N/A	N/A

Kuva 16. INI-tiedosto luettuna ”tblCheck”-taulukkoon

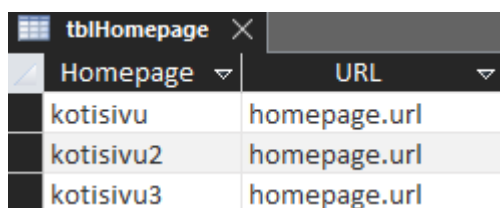
7.3.4 Valmistajan kotisivut

Valmistajien kotisivut on lisätty jokaisen valmistajan kansioon URL-linkityksellä. Jokaiseen INI-tiedostoon on myös lisätty tämä URL-tiedosto ja valmistajan nimi **(Kuva 17.)**.



Kuva 17. Kansiorakenne ”Valmistaja1”-kansioista sekä valmistajan kotisivun esitys INI-tiedostossa

Valmistajien kotisivut tuodaan vastaavalla tavalla kuin aikaisemmatkin tiedot on tuotu INI-tiedostoista. Kotisivuille on tehty oma ”tblHomepage”-taulukko, koska yhdellä valmistajalla voi olla useampi kotisivu, joten näin ne saadaan linkitettyä yhteen valmistajaan **(Kuva 18.)**. Taulukkoon lisätään myös ”ManufacturerID”, jotta myöhemmässä vaiheessa saadaan linkitettyä kotisivut ja valmistajat yhteen.



Homepage	URL
kotisivu	homepage.url
kotisivu2	homepage.url
kotisivu3	homepage.url

Kuva 18. Valmistajan kotisivut luettuna "tblHomepage"-taulukkoon

7.3.5 Kaikkien valmistajien ja käyttöohjeiden lukeminen Accessiin

Tässä vaiheessa tarvittavat makrot on tehty ja testiversion sijasta luettiin tiedot järjestelmässä olemassa olevien valmistajien ja valmistajien käyttöohjeiden lukeminen Accessiin. Tietokanta kasvoi tässä vaiheessa tuhansiin riveihin. Lukemisessa tuli muutamia ongelmia, liittyen erikoismerkkeihin, joita makrot eivät tunnista- neet samalla tavalla kuin normaaleja merkkejä, mutta nämä saatiin kuitenkin kui- tattua ja tiedot luettua.

8 JATKOKEHITYS

Työhön jäi paljon jatkokehittävää, joten tässä luvussa käydään läpi pääpiirteittäin kehitystä vaativat aiheet.

Tietokannassa on hyvä olla vertailuoperaattoreita, tärkeimpänä varmasti se, että onko tietokannassa kaikki tiedostot listattuna, jotka löytyy VEOn järjestelmästä.

Käyttäjälle näytettävän lomakkeen tekeminen on myös tärkeä vaihe tehdä huolella, jotta tietokantaa on miellyttävä käyttää ja se on selkeä. Lomake rakennetaan SQL-kyselyillä, jotta olemassa olevista taulukoista saadaan käyttäjälle vain oleelliset tiedot näkyviin. Tietokantaa ei voi julkaista käytettäväksi, ennen kuin lomake on saatu valmiiksi.

Tietokantaan tulee integroida muista tietokannoista ja järjestelmistä tietojen hakeminen, esimerkiksi luvussa 6.1 mainittu ”DAXManufacturerID”. Tällä tavalla tietokanta pystyy palvelemaan paremmin myös suunnittelun tarpeita tulevaisuudessa.

9 YHTEENVETO

Tämän opinnäytetyön päätavoitteena oli helpottaa käyttöohjeiden ylläpitoa, luomalla uusi tietokanta käyttäen Microsoft Accessia. Olemassa olevaa vanhaa tietokantaa joka on tehty Microsoft Exceliin, käytettiin runkona makroja tehdessä. Työtä ei saatu siihen vaiheeseen asti, että sen voisi julkaista käytettäväksi VEOlla.

Työssä käydään läpi myös tietokannan, ER-kaavion ja VBA:n perusteet ja pääkäsitteet. Työssä on käytetty myös esimerkkejä käytännön tietokannoista havainnollistamisen vuoksi.

Oppimisen kannalta työ oli erittäin hyvä. Koen, että opin paljon makrojen tekemisestä ja tietokantojen rakenteesta ja siitä, mitä pitää ottaa huomioon niitä suunnitellessa.

Suurimpia haasteita olivat aluksi selvästi ohjelmointikielen sisäistäminen ja miten makrot käyttäytyvät ylipäätään. Tähän ratkaisuna oli testaaminen ja ongelmatilanteissa varsinkin yleensä huomasi mitä oikeasti tapahtuu, kun makroja lähti käymään läpi vaihe vaiheelta, mitä missäkin tapahtuu ja miksi ei toimi.

Toinen iso haaste varsinkin alussa oli se, että ajattelin pelkästään kokonaisuutta, en sitä mitä sillä hetkellä pitäisi tehdä, eli pientä palaa isossa kokonaisuudessa. Tämä johti siihen, että en saanut keskittyttyä pelkästään siihen pieneen palaan mitä pitäisi tehdä, vaan yritin tehdä kerralla koko tietokannan makrot alusta loppuun, jolloin luonnollisesti se ei voi onnistua.

Kolmas haaste oli tasapainottelu työn, opinnäytetyön ja yksityiselämän välillä, mutta työn loppuvaiheessa tähän tuli mielestäni hyvin tasapainoa, jolloin mikään ei kärsinyt huomattavan paljoa.

Lopuksi haluaisin kiittää tuesta ja opastuksesta erityisesti esimiestäni Sami Hurinia ja mentoriani Rami Aihista.

LÄHTEET

InterviewBit. ER Model in DBMS. Viitattu 23.4.2022. <https://www.interviewbit.com/blog/er-model-in-dbms/>

Microsoft Docs. FileSystemObject object. Viitattu 12.4.2022. <https://docs.microsoft.com/en-us/office/vba/language/reference/user-interface-help/filesystemobject-object>

Microsoft Docs. Split function. Viitattu 23.4.2022. <https://docs.microsoft.com/en-us/office/vba/language/reference/user-interface-help/split-function>

Microsoft Docs. Getting started with VBA in Office. Viitattu 26.2.2022. <https://docs.microsoft.com/en-us/office/vba/library-reference/concepts/getting-started-with-vba-in-office#macros-and-the-visual-basic-editor>

Microsoft Support. Create, edit or delete a relationship. Viitattu 28.6.2021. <https://support.microsoft.com/en-us/office/create-edit-or-delete-a-relationship-dfa453a7-0b6d-4c34-a128-fdebc7e686af?ui=en-us&rs=en-us&ad=us>

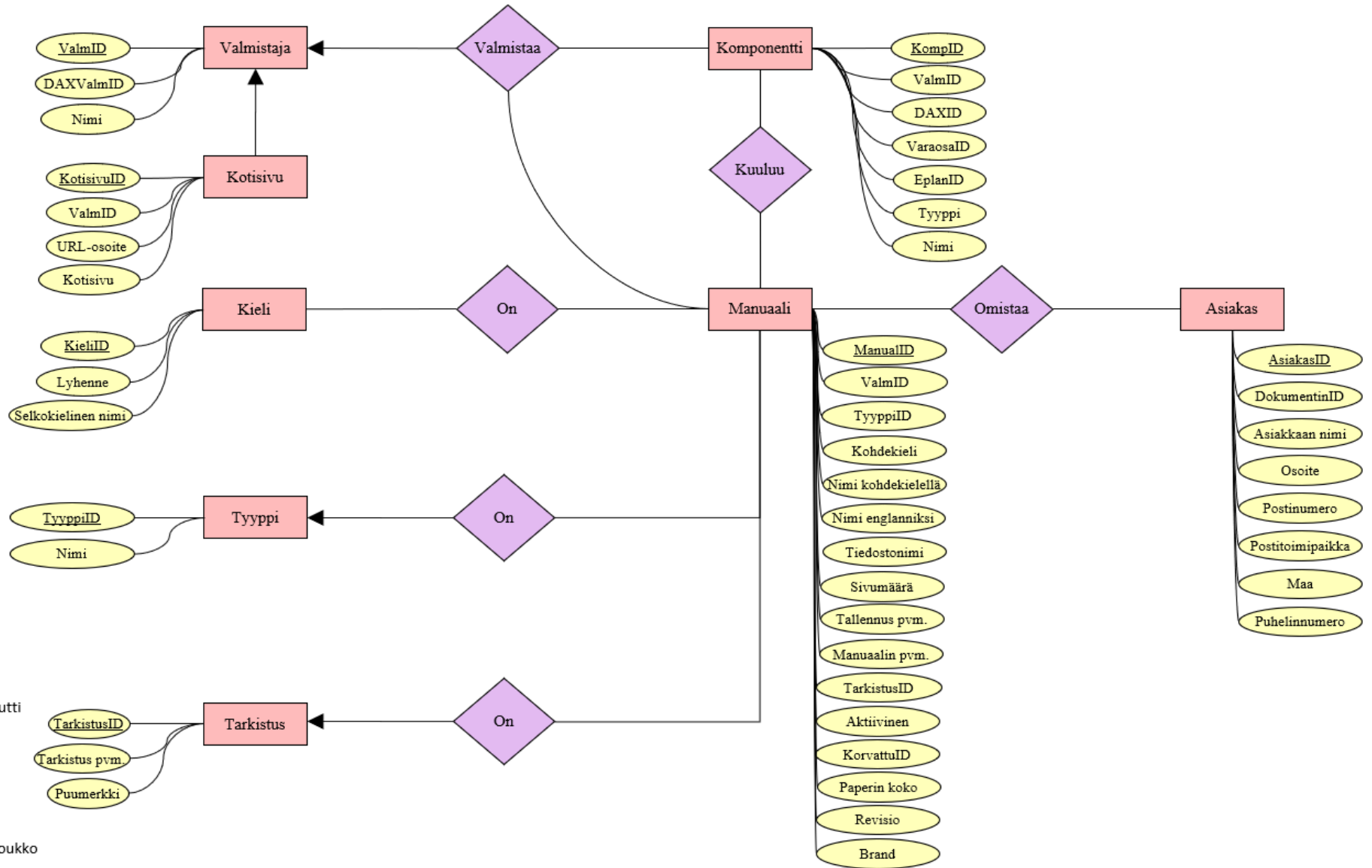
Oracle. What is database? Viitattu 21.3.2022. <https://www.oracle.com/database/what-is-database/>

Sippola, H. 2017. Energiakeskittymän ytimessä – VEOn tarina. Vaasa. Waasa Graphics Oy

VEO Oy. VEOn kotisivut. Viitattu 20.11.2021. <https://www.veo.fi>

LIITTEET

LIITE 1. Lopullinen ER-kaavio



LIITE 2. Accessin taulukot ja niiden yhteydet määriteltyinä

