



ALEKSI RAHOLA

# Tietojen migraatio Accessiin

SÄHKÖ- JA AUTOMAATIOTEKNIIKAN  
TUTKINTO-OHJELMA  
2024

## TIIVISTELMÄ

Rahola, Aleks: Tietojen migraatio Accessiin  
Opinnäytetyö, AMK  
Tutkinto-ohjelma: Sähkö- ja automaatiotekniikka  
Helmikuu 2024  
Sivumäärä: 42

Tämän opinnäytetyön tarkoituksena oli rakentaa tietokanta Olkiluoto 3 - laitoksen kenttälaitteista. Työn toimeksiantaja oli Teollisuuden Voima Oyj. Työn tavoitteena oli kerätä informaatiota monesta eri lähteestä yhdeksi yhteiseksi tietokannaksi. Erilaisten informaatiolähteiden ongelmana on oikean tiedon löytäminen kokonaisuudessaan sekä tietojen muokattavuus haluttuun muotoon.

Työn toteutuksessa suunniteltiin ensin tietokantaa. Suunnittelussa ensin tutkittiin haluttavat tiedot ja niiden prioriteetit. Niiden pohjalta luotiin luonnos tietokannan ulkonäöstä. Suunnitelman pohjalta rakennettiin tietokanta ja tuotiin esille keinoja toiminnallisuuden parantamiseksi.

Työn tavoitteeseen päästiin ja saatiin rakennettua tietokanta. Sen lisäksi tuli selvitettyä, mitä tietoja haluttiin tietokantaan. Se vaatii vielä loppujen laitepaikkojen lisäämisen, jotta saadaan tietokannasta sellainen kuin haluttiin.

Avainsanat: Relaatiotietokanta, Access, kenttälaitteet

## Abstract

Rahola, Aleks: Data migration to Access

Bachelor's thesis

Degree programme: Electrical and automation engineering

February 2024

Number of pages: 42

The purpose of this thesis was to build relational database for Olkiluoto 3 nuclear power plants field devices. Client of this thesis was Teollisuuden Voima Oyj. The objective for this was to gather data from different sources to one database. Big problem for the current system has been data being slow to find. Also, in TVOs own program, editing data has not been easy.

First task was to determine what data was needed to be imported to the database. Then the user interface had to be planned and draft for the forms was created. To create the database, features of Access had to be learned to help create the database and their usage were presented.

The objective was to create a database that contained data about field devices. Database was created and it contains part of the data what was wanted, and rest of the data is needed to be imported to database to get it finished.

Keywords: Relational database, Access, Field device

# SISÄLLYS

1 JOHDANTO .....	6
2 TEOLLISUUDEN VOIMA OYJ .....	8
2.1 Olkiluoto 3 .....	8
2.2 Kenttälaitteet .....	9
2.3 Relaatiotietokanta .....	10
3 MICROSOFT ACCESS .....	11
3.1 Taulukot .....	11
3.2 Taulukkojen yhdistykset .....	11
3.3 Käyttöliittymä .....	13
3.3.1 Kyselyt .....	14
3.3.2 Makrot .....	15
3.3.3 VBA .....	16
4 SUUNNITTELU JA TOTEUTUS .....	17
4.1 Suunnittelu .....	17
4.1.1 Tietojen määrittely .....	18
4.1.2 Taulukoiden suunnittelu .....	19
4.1.3 Käyttöliittymäsuunnittelu .....	19
4.2 Taulukoiden toteutus .....	21
4.2.1 Tietojen siirtäminen Accessiin .....	21
4.2.2 Taulukkojen jakaminen .....	22
4.2.3 Taulukoiden yhdistäminen .....	23
4.3 Lomakkeiden määrietykset .....	24
4.3.1 Lomakkeiden tietuelähteiden muodostaminen .....	25
4.3.2 Alilomakkeiden toteutus .....	27
4.4 Hakulomakkeen toteutus .....	30
4.4.1 Lomakkeen kenttien sijoittaminen .....	30
4.4.2 Toiminnallisuuden toteuttaminen .....	31
4.4.3 Excelin tuominen Accessiin .....	32
4.5 Laitapaikkalomakkeen toteutus .....	34
4.5.1 Laitapaikkalomakkeen ulkonäön toteuttaminen .....	35
4.5.2 Laitapaikkalomakkeen kenttien määrietykset .....	36
4.5.3 Toiminnallisuuden toteuttaminen .....	37
4.6 Komponenttilomakkeen muodostaminen .....	38
4.6.1 Komponenttilomakkeen ulkonäön toteuttaminen .....	38
4.6.2 Komponenttilomakkeen toiminnallisuuden toteuttaminen .....	39

5 YHTEENVETO.....	40
LÄHTEET.....	42

## 1 JOHDANTO

Informaatiota on nykyään kaikkialla. Kaikki tiedot, jotka on tallennettu koneelle tai kirjoitettu paperille on informaatiota. Datan kerääminen on tärkeää. Sen tarkoituksena on kerätä tavaroihin tai ihmisiin liittyviä tietoja myöhempää käyttöä varten. Dataa voi olla tietokannoissa, tiedostoissa tai paperilla. Tämän opinnäytetyön data koostuu Olkiluoto 3 -laitoksen kenttälaitteiden tiedoista.

Nykyään laitteiden data löytyy eri paikoista, kuten Exceleistä ja Teollisuuden voiman (TVO) käyttämästä tietokantasovelluksesta. TVO:n tietokantasovelluksessa on myös monia eri välilehtiä, josta löytyy haluttavat tiedot, ja niiden välillä on hieman hankalaa liikkua. Sen takia tietojen hakeminen nykyään on hidasta. Toinen merkittävä ongelma on ollut tietojen päivittämisessä, joka on hankalaa varsinkin TVO:n omassa sovelluksessa. Tämä työn tarkoituksena on tuoda ratkaisu ongelmille.

Tämän opinnäytetyön tavoitteena on siirtää Olkiluoto 3 -laitoksen kenttälaitteiden tiedot Excelistä Accessiin. Siirto on tarpeen tehdä, koska Excel ei ole paras paikka pitää suurta määrää tietoa, josta osa on vielä päällekkäistä. Koska monet rivit voivat sisältää sen saman tiedon, tallennetaan se turhaan moneen kertaan ja se taas suurentaa Excelin kokoa turhaan. Lisäksi jos tieto on koodi tai merkkijono, kirjoitusvirheen mahdollisuus suurentuu huomattavasti.

Tämän opinnäytetyön tarkoituksena on myös tutkia mitä tietoja laitteista halutaan tallentaa ja miten niitä tulee olemaan näkyvissä käyttöliittymässä. Tietojen priorisointi on yksi asia mitä tulee ottaa huomioon suunnittelussa. Niistä saatuja tuloksia voidaan tuoda esiin, mikäli TVO:n omia sovelluksia joskus tullaan päivittämään.

Työn aluksi käydään teoriaa lävitse kahdessa luvussa. Ensimmäisessä luvussa käydään yleisesti lävitse toimeksiantajaa, tietokantaan tulevia laitteita relaatiotietokantaa. Toisessa kappaleessa käsitellään Accessia ja ohjelmas- sa olevia ominaisuuksia. Työn toteutusta käsitellään teorialukujen jälkeen. Siinä määritetään tavoitteet, luodaan suunnitelma ja rakennetaan suunnitel- man pohjalta tietokanta. Viimeisessä luvussa käsitellään tavoitteiden saavut- tamista ja luodaan yhteenveto opinnäytetyöstä.

## 2 TEOLLISUUDEN VOIMA OYJ

Tämä opinnäytetyö on tehty Teollisuuden Voima Oyj:lle (TVO). TVO on listaamaton julkinen osakeyhtiö, joka toimii sähköntuotannon alalla. TVO tuottaa sähköä kolmen ydinvoimalaitoksen avulla. Teollisuuden Voima -konserniin kuuluvat tytäryhtiöt TVO Nuclear Services Oy (TVONS) ja Posiva Oy. TVO:lla on viisi omistajayhtiötä, joista suurin on Pohjolan Voima Oyj. (Teollisuuden Voima Oyj, n.d.b)

### 2.1 Olkiluoto 3

Olkiluoto 3 (OL3) on TVO:n uusiin laitosyksikkö, jonka sähköntuotanto alkoi käyttöönotolla maaliskuussa 2022 ja säännöllinen sähköntuotanto alkoi huhtikuussa 2023. OL3 tilattiin avaimet käteen periaatteella laitostoimittajalta, johon kuuluvat AREVA GmbH, AREVA NP SAS ja Siemens AG. AREVA on reaktoripuolen ja Siemens turbiinipuolen toimittaja. (Teollisuuden Voima Oyj, n.d.a) OL3 on ERP-tyyppinen painevesilaitos (eurooppalainen painevesireaktori), jonka nettoteho on noin 1600 MW.





Kuva 1. Olkiluoto 3 -laitosyksikkö (Teollisuuden Voima Oyj, 2018)

## 2.2 Kenttälaitteet

Kenttälaitteilla tarkoitetaan prosessiympäristössä prosessiin kytkettyjä elektronisia laitteita. Ne koostuvat käytännössä järjestelmän toimintaa valvovista laitteista sekä venttiileistä. Laitteita kannattaa hyödyntää vikasietoisuutta vaativissa järjestelmissä. Turva-automaatiossa mittalaitteet seuraavat järjestelmän tilaa ja toimilaitteilla, kuten venttiileillä, toteutetaan turvatoiminto järjestelmän tilan palauttamisen turvalliseksi. Turvatoimintoja voivat olla esimerkiksi paineenalennus, jossa venttiili aukeaa painerajan ylityksestä. (Tukes, 2021)

Laitteiden tehtävänä on esimerkiksi mittailta erilaisia asioita ja lähettää siitä saadut arvot prosessinohjaukselle. Laitteisiin kuuluvat esimerkiksi anturit ja lähetimet. Anturi muuttaa prosessisuureen sähköiseksi suureeksi. Lähetin muuttaa tuon sähköisen suureen standardimuotoiseksi signaaliksi, kuten 4-20mA tai 0-5V. (Rika sensor, 2021)

### 2.3 Relaatietietokanta

Relaatietietokantaa käytetään suurten datamäärien säilyttämiseen. Tarkoituksena relaatiotietokannalla on jakaa tiedot taulukkoihin ja yhdistää taulukot yhdistyksien avulla. Sillä tavalla tiedot esiintyvät vain kerran ja muihin taulukkoihin voidaan hakea viittauksella haluttava tieto. Teoriaa käydään tarkemmin luvussa 3 Accessin näkökulmasta.

Relaatietietokannan käytöllä on monia etuja. Yksi suurimmista eduista on esittää haluttuja tietoja. Varsinkin suuren datamäärän suodattaminen onnistuu helposti relaatiotietokannan avulla. Muita etuja ovat muun muassa muokattavuus, datan normalisointi ja monta käyttäjää pääsee käyttää tietokantaa samanaikaisesti. Muokattavuudella tarkoitetaan datan lisäämistä, muokkaamista tai poistamista.

## 3 MICROSOFT ACCESS

Microsoft Access on Microsoftin kehittämä relaatiotietokantaohjelma. Datan hallinta ja sen esittäminen ovat Accessin päätehtävät. Ne saadaan toteutettua käyttämällä muun muassa taulukoita, lomakkeita ja kyselyitä. Tässä opinnäytetyössä käytetään Access for Microsoft 365 -versiota.

### 3.1 Taulukot

Taulukot sisältävät kaiken tiedon, mitä tietokannassa käytetään. Tietokannoissa käytetään yleensä useita taulukoita, jonka tavoitteena on vähentää päällekkäistä dataa. Eri taulukkojen tietoja voidaan yhdistellä relaatioiden avulla, joista kerrotaan enemmän luvussa 3.2. Myös taulukon sisällä kannattaa miettiä, mitkä tiedot kannattaa vielä jakaa useampaan kenttään. Esimerkiksi etu- ja sukunimet kannattaa jakaa omiin kenttiin.

Taulukkojen sarakkeet ja rivit ovat eri nimisiä Accessissa. Tietueella tarkoitetaan rivejä ja kentällä tarkoitetaan saraketta. Kentän arvolla tarkoitetaan tietueelle annettua arvoa. (Microsoft, n.d.a) Jokaiselle kentälle määritetään tietotyyppi, joka valitaan haluttavan tiedon perusteella.

### 3.2 Taulukkojen yhdistykset

Taulukot voivat olla yhdistyneet toisiinsa eli olla relaatiossa. Datan pystyy tällä tavoin jakamaan eri taulukoihin ja liittämään ne toisiinsa perus- ja viiteavaimien avulla. Accessissa on käytössä kolmenlaisia yhteystyyppejä:

- Yksi-moneen-yhteys
- Monta-moneen-yhteys
- Yksi-yhteen-yhteys

Yksi-moneen-yhteydessä toisesta taulukosta voi olla vain yksi arvo käytössä ja toisesta monta arvoa. Monta-moneen-yhteydessä kummastakin taulukosta voi olla monta arvoa käytössä. Taulukot yhdistetään toisiinsa kolmannen taulukon avulla käyttäen hyväksi kahta yksi-moneen-yhteyttä. Kolmanteen taulukoon on yhdistetty kummatkin moneen puolet. Yksi-yhteen-yhteydessä kummastakin voi olla vain yksi rivi käytössä. Tätä yhteyttä käytetään ainoastaan taulukon jakamiseen, eristää osan taulukosta tai tallentaa päätaulukon alijoukkoa. (Microsoft, n.d.c)

Yhteys perustuu perus- ja viiteavaimiin. Perusavain-kenttä sisältää yksilöllisen tunnisteen tietueelle. Perusavaimena toimii yleensä ID-kenttä, jonka Access luo automaattisesti. Perusavain-kenttänä voidaan myös käyttää käyttäjän omaa tunnistetta, kuten tuotekoodia. Vaatimuksena perusavaimen käytölle on, että kentän tiedot täytyy olla uniikkeja. Viiteavain sijaitsee toisessa taulukossa, johon viitataan päätaulukosta.

Yhdistyksissä voi käyttää kahta eri menetelmää. Taulukon puolelta voi luoda yhteyden Ohjatun hakukentän luomisen avulla. Access luo automaattisesti yhteyden kenttien välille. Toinen vaihtoehto on luoda yhteydet yhdistyksetikkunassa. Yhdistykset-ikkunassa tuodaan taulukot, joiden väliin haluaa luoda yhteyden ja vedetään halutun taulukon kentästä yhteys toisen taulukon kenttään. Kuvassa 2 vasemmanpuolinen taulukko on perusavainpuoli ja oikea viiteavainpuoli. Yhdistyksissä voidaan myös määrittää viite-eheyden käytettäväksi.

Table/Query:	Related Table/Query:
Nimike	Komponentti
Nimike ID	Nimike

Enforce Referential Integrity  
 Cascade Update Related Fields  
 Cascade Delete Related Records

Relationship Type: One-To-Many

## Kuva 2. Yhdistyksien luomisesta

Viite-ehydeillä estetään yhteydettömät tietueet. Arvoja, jotka eivät sijaitse perusavainpuolella, ei voi sijoittaa kyseiseen kenttään viiteavainpuolella. Kenttien johdannaisasetuksilla saadaan muutettua yksi-puolen arvoja. Johdannaispäivityksellä saadaan muokattua kentän arvoa ja se päivittyy kaikkiin viitattuihin arvoihin. Johdannaispoistolla saadaan poistettua kaikki perusavaimen viittaavat tietueet. (Microsoft. n.d.e)

### 3.3 Käyttöliittymä

Käyttöliittymän tarkoituksena on helpottaa ohjelmien käyttöä. Se toimii niin sanottuna tulkkina ohjelman ja käyttäjän välillä. Käyttäjää voidaan myös ohjata tekemään asioita tietyllä tavalla. Accessissa graafiset elementit toteutetaan lomakkeiden ja raporttien avulla. Toiminnalliset elementit toteutetaan kyselyiden, makrojen ja VBA-kielen avulla.

Lomakkeeseen voidaan liittää erityyppisiä kenttiä kuten taulukoita, painikkeita, tekstikenttiä ja valintaruutuja, joiden tarkoituksena on esittää taulukoiden sisältämiä tietoja. Käyttöliittymä toteutetaan pääsääntöisesti useiden lomakkeiden avulla, jossa lomakkeiden välillä siirtymiset toteutetaan makrojen tai VBA:n avulla.

Suunniteltaessa lomaketta tulee olla tiedossa halutut tiedot, sekä miten tärkeää jokainen tieto on. Käyttöliittymästä kannattaa sijoittaa tärkeimmät tiedot yläosaan tai vasempaan reunaan käyttäjien takia. Yleisesti tietoa etsitään ensin yläreunasta ja vasemmasta reunasta. Harvemmin tarvittavat tiedot voidaan tarvittaessa piilottaa painikkeiden taakse tai sijoittaa esimerkiksi alhaalle tai reunoihin.

Raporttien avulla voidaan kerätä tietoja ja tehdä niistä yhteenvetoja. Tiedot haetaan tietolähteestä, kuten taulukosta ja asetellaan raporttiin halutulla tavalla. Raporttia käytetään etenkin tulostamisen yhteydessä. Raportteja käytäessä tulee huomioida sen olevan vain lukua varten.

### 3.3.1 Kyselyt

Kyselyiden avulla saadaan haettua tietoa eri taulukoista ja esittää ne lomakkeessa tai raportissa. Kyselytyyppejä on monia erilaisia, joista tärkeimpiä ovat valinta- ja toimintokysely. Valintakyselyllä haetaan tietoja taulukoista ja toimintokyselyllä saadaan muokattua, lisättyä ja poistettua tietoa. (Microsoft, n.d.b) Kysely voidaan tehdä joko käyttäen Ohjattua kyselyn luomista tai rakennäkymässä. Sen voi myös tehdä SQL-kieltä käyttäen SQL-näkymässä.

SQL-kieli on tehokkaampi ja monipuolisempi keino tehdä kyselyitä. Kyselyiden toteuttamiseksi tulee kuitenkin tietoa SQL-kielestä ja sen toiminnasta. Ehtojen luominen voi olla yksinkertaisempaa SQL-kielen avulla, koska kaikki ehdot sijoitetaan WHERE-komentoon. Mutta mikäli on tarve tehdä yksinkertainen kysely, kannattaa se tehdä rakennäkymästä.

Valintakyselyllä haetaan kenttiä taulukoista. SQL-kielellä se tapahtuu SELECT- ja FROM-komennoilla. SELECT-komennolla valitaan kyselyssä näkyvät kentät ja FROM-komennolla taulukko tai taulukot, jossa kentät sijaitsevat. Siihen voidaan myös lisätä ehtoja WHERE-komennolla. Valintakyselyt toimivat etenkin hakujen yhteydessä.

Toimintokyselyiden avulla saadaan muokattua tietoja eri taulukoista. Liittämiskyselyllä saadaan sijoitettua tietoja taulukosta toiseen. SQL-kieltä käytettäessä tulee lisätä valintakyselyyn INSERT INTO -komento, joka sijoittaa siihen määritettyyn taulukkoon tiedot. Jos toisen taulukon tietoja tarvitsee päivittää käytetään UPDATE-komentoa, joka päivittää kentän arvon. Poistamiset tehdään DELETE-komennolla.

Kyselyssä käytettävät taulukot yhdistetään toisiinsa liitoksien avulla. Access luo liitokset automaattisesti, mikäli niiden välillä on jo yhteys. Kyselyillä on neljää eri perusliitostyyppiä:

- Sisäliitos
- Ulkoliitos
- Ristiliitos

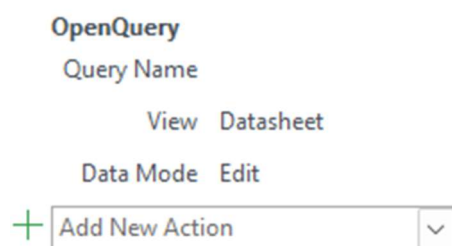
- Eriarvoinen liitos

Sisäliitoksessa tiedot sisällytetään vain, jos molemmista taulukoista löytyy vastaavat tiedot. Accessissa sisäliitos luodaan automaattisesti, jos taulukoilla löytyy jo yhteys tai jos toinen liitoskentistä on perusavain ja toisesta taulukosta löytyy samalla tai yhteensopivalla tietotyypillä oleva kenttä. Ulkoliitokset ovat melkein kuin sisäliitos, mutta siinä lisätään yhdistyksien lisäksi toisesta taulukosta kaikki tiedot. Ristiliitoksessa kahden taulukon tietueista näytetään kaikki mahdolliset yhdistelmät. Se antaa yleensä huonoja tuloksia, koska tietoja ei olla yhdistetty. Eriarvoisella liitoksella tarkoitetaan, kun käytetään jotakin muuta kuin yhtäsuuruusmerkkiä vertailussa. Sitä ei kuitenkaan voi tehdä rakennenäkymässä vaan se täytyy tehdä SQL-näkymässä. Ristiliitos ja eriarvoinen liitos ovat harvinaisia. (Microsoft, n.d.c)

SQL-kielellä sisäliitos luodaan INNER JOIN -komennolla. Ulkoliitos luodaan LEFT JOIN tai RIGHT JOIN -komennolla riippuen haluttavista taulukoiden arvoista.

### 3.3.2 Makrot

Makrot ovat keino koodata Accessissa. Sen tarkoituksena on automatisoida ja lisätä toimintoja esimerkiksi lomakkeille. Makrojen luominen toimii koodaamisen tavoin, mutta tekeminen on grafiikkapohjaisempaa kuten kuvassa 3.



Kuva 3. Makrojen luonti.

### 3.3.3 VBA

Visual Basic for Applications (VBA) on Microsoftin kehittämä ohjelmistokieli. Sen tarkoituksena on luoda monimutkaisempia toimintoja ja ohjata käyttäjät tekemään asioita haluamalla tavalla. VBA:n avulla voidaan korvata makroja tai lausekkeenmuodostimen. Lausekkeenmuodostin on Accessin työkalu, jolla voidaan rakentaa lausekkeitä käyttäen enemmän grafiikallista tapaa.

Toteuttaminen on hankalampaa kuin Accessin työkalujen käyttäminen, mutta saadut ratkaisut voivat olla tehokkaampia työkaluihin verrattuna. Yksi merkittävä tekijä haastavuudessa on kieli itsessään. Kieltä käyttäessä täytyy tietää, miten halutun toiminnon saa muodostettua.

Asioita, joita kannattaa toteuttaa ovat esimerkiksi napit ja erilaiset toiminnot. VBA:n avulla käyttöliittymän toiminnasta saa helppokäyttöisempää. Esimerkiksi painikkeeseen voi ohjelmoida monia eri komentoja ja yhdellä painalluksella saa toteutettua ne kaikki.



## 4 SUUNNITTELU JA TOTEUTUS

Tässä luvussa käsitellään tietokannan suunnittelua ja toteutusta. Tietokannan toteutus alkaa suunnittelulla, jossa käsitellään haluttuja tietoja, taulukoiden jakamista sekä käyttöliittymän luonnosta. Suunnitelman tarkoituksena on esittää alustavasti tietokannan toimintaa ja ulkonäköä. Toteutuksessa rakennetaan tietokanta suunnitelman avulla.

Tavoitteena tällä opinnäytetyöllä on luoda tietokanta, johon sijoitetaan kenttä-laitteisiin liittyviä tietoja. Tietojenkäsittelylle olisi tarkoitus rakentaa käyttöliittymä, jonka kautta hallinta toteutuu.

### 4.1 Suunnittelu

Tietokannan toteutus alkaa suunnittelusta, jossa määritetään halutut tiedot ja luonnos käyttöliittymästä. Suunnittelu on tärkeä osa työtä, koska sen avulla saadaan selkeytettyä haluttavaa lopputulosta. Hyvässä suunnitelmassa on kaikki haluttavat tiedot otettu ylös ja jaettu ne ryhmiin taulukoihin sijoittamisen perusteella. Käyttöliittymän puolella suunnitelmassa on hyvä olla luonnos, josta näkee ideaa tietojen paikoista. Luonnoksella voidaan esittää käyttöliittymän ulkonäköä ja esittää se ennen työn aloittamista.

Yksi ensimmäisistä tehtävistä on määrittää kerättävä data. Määrittämisessä on otettava huomioon myös mahdollisesti tulevaisuudessa tarvittavat tiedot. Määrittelyssä käydään läpi kaikki tallennuspaikat ja päätetään, mitkä niistä tullaan tarvitsemaan.

#### 4.1.1 Tietojen määrittely

Tietojen määrittelyt ovat käyty läpi automaatiotekniikan kanssa. Kuvaan 4 on kerätty haluttavat tiedot, jotka ovat lajiteltu tyyppien perusteella. Lajittelussa on otettu huomioon, kuuluuko data laitepaikkaan vai komponenttiin. Nämä asiat ovat käsitelty jo tässä vaiheessa, koska se helpottaa käyttöliittymän ja taulukkojen muodostamista.

```

Hakukentät
Laitepaikka, komponenttitunnus ja nimi, nimike, valmistaja

Järjestys tuloksissa
Laitepaikka, nimike, komponentin nimi, komponentin tunnus, komponentin koodi,
valmistaja

Komponentin sisällä olevat tiedot
Tunnus, nimi, valmistaja, varaosat, HW, SW, Lisätiedot, lista asennuspaikoista,
dokumentit, tyyppi, nimike, connector

Laitepaikan sisällä olevat tiedot
Laitepaikka, komponentin tunnus, nimi, valmistaja, dokumentit, lisätiedot, nimike,
luokitukset, range, valokuvat, onko VD:ssä, huone, OLG, viritys- ja mittausalue

Laitepaikan dokumentit
EOMM, Terminal diagram, circuit diagram, Setting list, system suitability analysis,
PID, CFI, Määräaikaistestiohje, hookup

Komponentin dokumentit
Data sheet, Suitability analysis, Equipment description, TPQ

Luokitukset
Turvaluokka, kunnossapitoluokka, Seismic Category, Electrical safety class, Electrical
quality class, I&C quality class

```

#### Kuva 4. Halutut tiedot tietokantaan

Määritetyt tiedot tulee jakaa taulukoihin, kuten on mainittu luvussa 3.1. Taulukoiden tiedot kannattaa jakaa tarpeeksi pieniin osiin, jotta tietojen hallinta olisi helpompaa. Esimerkiksi yhteen taulukkoon kerätään tuotekoodi, tuotteen nimi ja tyyppi, koska ne eivät toistu muissa taulukoissa.

Tietojen kirjoitustyyliä muokataan samannäköisiksi. Esimerkiksi osassa tuotekoodista on välilyöntejä, joita ei ole kaikissa samantyyllisissä tuotekoodista. Sen avulla voidaan välttää päällekkäisiä tietoja ja myös helpotetaan tietojen etsimistä.

Näiden tietojen sijoittamiseksi taulukoihin ne tulee jakaa vielä pienempiin osiin. Taulukoiden luomiseksi kannattaa luoda päätaulukko, johon kerätään kaikki tiedot muista taulukoista. Päätaulukon avulla saadaan esitettyä tiedot

oikein. Tämä tietokanta tulee perustumaan laitepaikan ympärille, joten se tulee olemaan päätaulukko ja muiden taulukkojen tiedot sijoitetaan luvussa 3.2 esitetyn viittaamisen perusteella.

#### 4.1.2 Taulukoiden suunnittelu

Suunnittelu kannattaa aloittaa päätaulukoista ja jakaa tietoa muihin taulukoihin tarpeen tullen. Tavoitteena on kaikkien toistuvien tietojen jakaminen toisiin taulukoihin ja sieltä viittaaminen päätaulukoon. Näin ollen päätaulukossa näkyy ainoastaan viittaukset muista taulukoista. Myös kenttien tietotyypit tulee olla tiedossa, jotta yhdistämisvaiheessa ei tule ongelmia.

Lomakkeet voidaan tehdä taulukoiden avulla. Sen takia päätaulukot tulee olla tiedossa mahdollisemman varhaisessa vaiheessa. Komponentti- ja laitepaikkataulukot ovat päätaulukkoina tässä tietokannassa, koska lomakkeissa esitetään laitepaikkoihin ja komponentteihin liittyviä tietoja.

#### 4.1.3 Käyttöliittymäsuunnittelu

Käyttöliittymäsuunnittelulla tarkoitetaan sekä ulkonäön että toiminnallisuuden suunnittelua. Ulkonäössä tulee ottaa huomioon monia asioita, kuten selkeys ja käytännöllisyys. Esimerkiksi painikkeiden tulee erottaa tavallista kentistä.

Käyttöliittymäsuunnittelussa käytetään hyödyksi Jacob Nielsenin kehittämä periaatetta, jota kutsutaan Nielsenin heuristiikaksi. Siihen kuuluu kymmenen suunnitteluperiaatetta:

- Järjestelmän tilan näkyvyys
- Järjestelmän ja todellisen maailman vastaavuus
- Käyttäjän kontrolli ja vapaus
- Yhdenmukaisuus ja standardit
- Virheiden estäminen
- Tunnistaminen muistamisen sijaan
- Joustavuus ja käytön tehokkuus

- Esteettinen ja minimalistinen suunnittelu
- Auttaminen virhetilanteiden tunnistamisessa, vianmäärityksessä ja virheistä toipumisessa
- Ohjeet ja dokumentit

Listan perusteella voidaan tutkia käyttöliittymän käytettävyyttä sitä suunniteltaessa ja asettaa sille tavoitteet. (Haltu, 2023; Nielsen, 2024). Käyttäjiltä tuleva palaute kertoo enemmän ohjelman käytettävyydestä ja sen avulla voidaan tehdä muokkauksia.

Suunniteltaessa tulee ensin selvittää tietojen priorisointi, koska sen perusteella sijoitetaan tiedot lomakkeisiin. Käytetyimmät tiedot tulee sijoittaa selkeästi näkyville. Niiden paikka on yleensä ylhäällä, koska käyttäjät katsovat siihen yleensä ensimmäisenä. Harvoin tarvittavia tietoja voidaan sijoittaa esimerkiksi alareunaan tai painikkeiden taakse.

Käyttöliittymän luonnos on tehty Powerpointilla ja siihen on liitetty osa halutavista tiedoista, kuten on mallinnettu kuvaan 5. Siinä on otettu huomioon prioriteetit ja sijoiteltu tiedot sen perusteella. Laitapaikka-näkymässä tärkeimmät tiedot ovat itse laitepaikka, sen nimi, sijoituspaikka ja asennetut komponentit.

Laitapaikka				
Laitapaikka		Laitapaikan nimi		
Komponentti tunnus	Tuotekoodi	Nimike	Valmistaja	Huone
Lisätiedot				
Luokitukset	Dokumentit	Kuvat	Muut tiedot	

Kuva 5. Laitepaikkäkäkymän luonnos

Kuvaan 6 on suunniteltu nimikenkäkymän ulkonäköä. Siihen on tarkoituksena kerätä komponentteihin liittyviä tietoja, kuten tuotetunnuksia, asennuspaikko-

ja valmistajia. Kyseinen sijoittelu on tehty priorisoinnin avulla. Kuvissa 5 ja 6 olevat siniset laatikot ovat nappeja, jotka avaavat tietyn lomakkeen tai suorittavat toimintoja.

The image shows a form titled "Nimike" with the following elements:

- Input fields: Komponenttitunnus, Tyypä, Nimike, Nimi, Valmistaja.
- Buttons: Asennuspaikat, Dokumentit, Muut tiedot.
- A large text area labeled "Lisätiedot".
- A small blue square icon next to the text "QC".

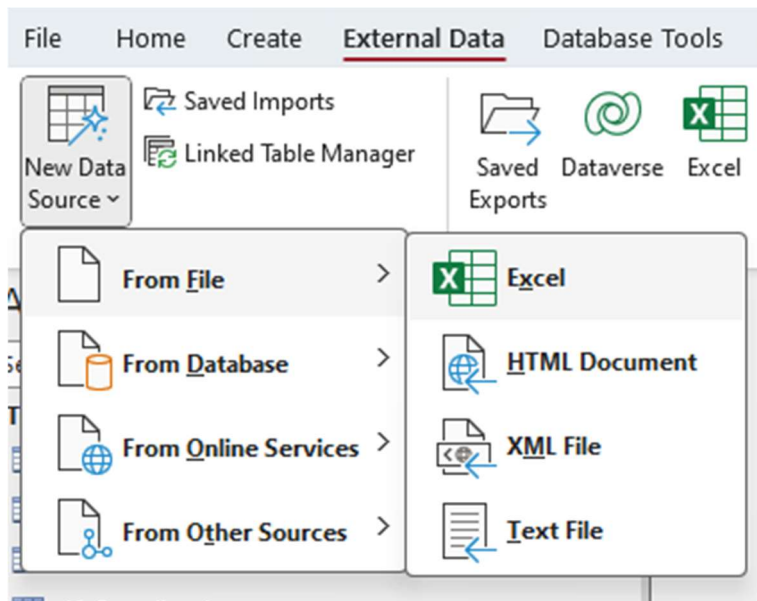
Kuva 6. Nimikenäkymän luonnos

## 4.2 Taulukoiden toteutus

Tietokannan toteutus alkaa taulukoiden tekemisellä. Taulukoilla on tärkeä merkitys tietokannan toimintaan, koska ne sisältävät tietokantojen datan. Tietokannan toteutuksesta tulee helpompaa, mitä paremmin on taulukot luotu. Accessissa voi luoda lomakkeen taulukon avulla vetämällä taulukko lomakkeeseen.

### 4.2.1 Tietojen siirtäminen Accessiin

Kenttälaitteiden sisältämät datat ovat kerätty yhteen Exceliin, koska sen voi ajaa helposti Accessiin. Jotta tiedot saadaan siirrettyä sinne, tulee Excelin ja Accessin sarakkeiden nimet olla samat. Siirto tapahtuu External data -välilehden New data source -työkalulla. Sieltä pystyy valitsemaan Excelin käytettäväksi, kuten on näytetty kuvassa 7.



Kuva 7. Datan tuominen Accessiin Excelin avulla

Myöhemmässä vaiheessa tavoitteena on rakentaa työkalu, jolla Excelin tiedot voidaan sijoittaa moneen eri taulukkoon ja sen lisäksi tarkistaa syötetyt tiedot. Työkalu asennetaan painikkeeseen ja painettaessa sitä käyttäjälle aukeaa resurssienhallinta, josta valitaan ajettava Excel-tiedosto.

#### 4.2.2 Taulukkojen jakaminen

Data on aluksi sijoitettu yhteen taulukkoon, jonka jakamiseen voidaan käyttää muutamaa eri vaihtoehtoa. Yksi vaihtoehtoista on käyttää Analyze table -työkalua, joka tarkistaa taulukon päällekkäiset tiedot ja antaa ehdotuksen, miten jakaa datan taulukoihin. Se on yksinkertainen vaihtoehto, mutta monimutkaisissa taulukoissa se ei välttämättä ole paras vaihtoehto.

Liittämiskyselyllä voidaan sijoittaa halutut kentät toiseen taulukkoon. Sen saa valittua kyselyn rakennäköstä Liitä. Rakennäkössä Kenttä-kohtaan sijoitetaan haluttu kenttä, jossa tiedot sijaitsevat. Taulukko-kohtaan sijoitetaan taulukko, jossa Kenttä-kohtaan kenttä sijaitsee. Liitä-kohtaan sijoitetaan toisen taulukon kenttä, johon tiedot siirretään. Mikäli kenttien nimet vastaavat toisiaan, Liitä-kohta täyttyy automaattisesti kentän nimellä. Kyseiset kohdat kun täyttää, pystyy suorittamaan liittämiskyselyn. Kyselyyn voi-

daan myös asettaa ehtoja ja lajitteluperusteita. Ennen kyselyn toteuttamista, kannattaa kokeilla valintakyselyn avulla kyselyn toiminta oikein.

Kyselyn voi toteuttaa myös SQL-kielellä, jonka avulla voidaan tehdä monimutkaisiakin kyselyitä. Liittämiskyselyssä käytetään INSERT INTO -komentoa, joka liittää kyselyssä saadut tiedot haluttuihin kenttiin. DISTINCT-komennolla saadaan uniikit vaihtoehdot etsittyä. Sitä käyttäessä tulee kuitenkin ottaa huomioon, että se vaikuttaa jokaiseen kenttään. Alla on esimerkki nimikkeiden siirtämisestä toiseen taulukkoon.

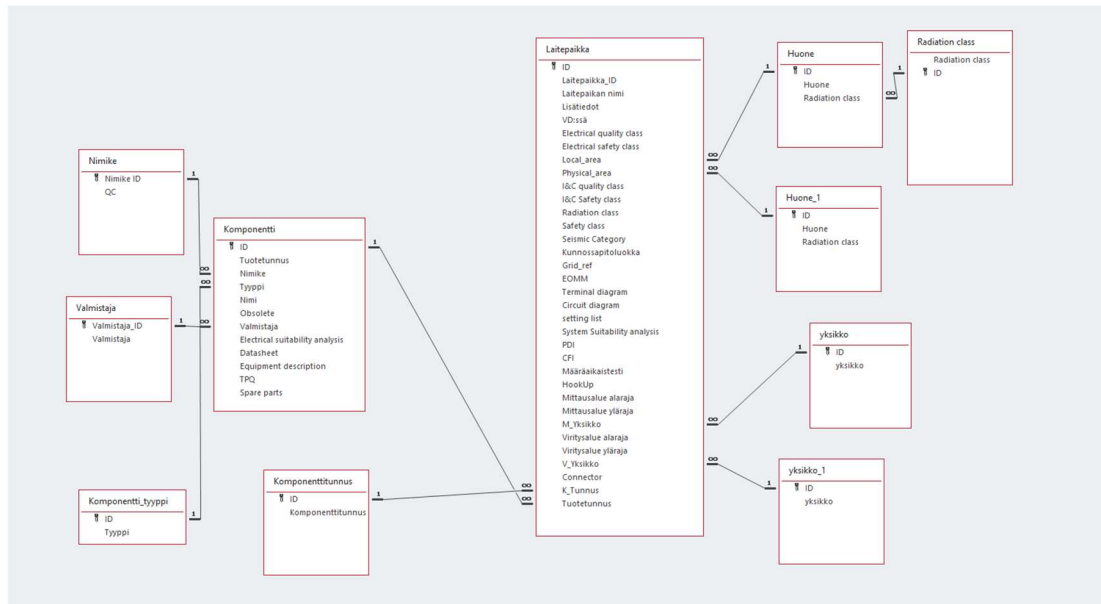
```
INSERT INTO Nimike ( [Nimike ID], QC )
SELECT DISTINCT Temp.Nimike, Temp.QC
FROM Temp
```

Taulukkojen tarkoituksena on, että päätaulukkoon kerättäisiin muiden taulukkojen perusavainkenttien arvot. Keino vaihtaa kentän arvo perusavainkentän arvoksi esitetään luvussa 4.4.3. Siihen kerätään myös muita tietoja, joita ei tarvinnut jakaa muihin taulukoihin. Jos kentällä on vain muutama mahdollinen arvo ja ne eivät muutu, voi sille muodostaa listan, joka sisältää valittavat vaihtoehdot. Sitä listaa tullaan käyttämään muutamassa kentässä.

#### 4.2.3 Taulukoiden yhdistäminen

Laitapaikkataulukko on tämän tietokannan päätaulukko. Siihen on tarkoituksenaan kerätä muiden taulukoiden tietoja perus- ja viiteavaimien avulla. Kyseiset yhdistykset ovat toteutettu Laskuri-tietotyyppisten ID-kenttien avulla. Sen takia kaikki viitattavat kentät sisältävät numeron, esimerkiksi laitepaikan sijasta. Kuvaan 8 on määritetty tämän tietokannan yhdistykset.

Viite-ehyettä ja johdannaispäivityksiä käytetään jokaisessa yhteydessä. Tarkoituksena on esimerkiksi komponentteihin tehdyt päivitykset tulee päivittämään jokaiseen laitepaikkaan. Työssä ei käytetä poistamista, koska on mahdollista poistaa vahingossa tietoa eikä poistamista voi peruuttaa. Poistamiset hoidetaan taulukkojen kautta, jotta pienennetään vahingossa poistamisen todennäköisyyttä.



Kuva 8. Tietokannan taulukkojen yhdistykset

### 4.3 Lomakkeiden määrittäykset

Käyttöliittymässä tulee olemaan kolme lomaketta: Haku, komponentti ja laitepaikka. Hakukentältä on mahdollisuus siirtyä komponentti- tai laitepaikkalomakkeeseen. Komponenttilomakkeelta on mahdollisuus siirtyä laitepaikkalomakkeeseen ja vastaavasti Laitepaikkalomakkeelta komponenttilomakkeeseen.

Tietokanta tulee avautumaan hakulomakkeelle. Sen saa määritettyä asetuksista, kuten kuvassa 9 on näytetty. Tarkoituksena on helpottaa tietokannan käyttöä, jotta käyttäjien ei itse tarvitsisi avata hakulomaketta.

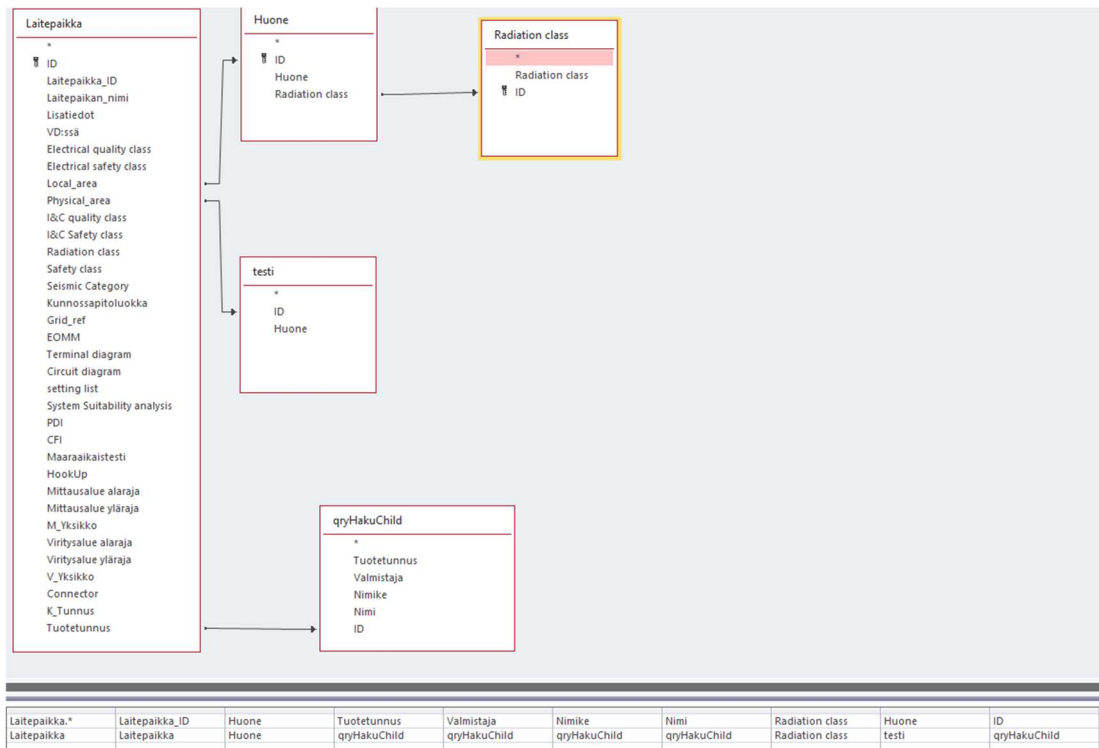
Kuva 9. Hakukentän määrittäminen avautumislomakkeeksi



#### 4.3.1 Lomakkeiden tietuelähteiden muodostaminen

Lomaketta luotaessa määritetään sille tietuelähte, joka on taulukko tai kysely. Tässä työssä käytetään kyselyä, jolla saadaan haettua tiedot monesta eri taulukosta ja liitettyä ne lomakkeeseen. Jokaiselle työn lomakkeelle rakennetaan oma kysely, jota käytetään lomakkeen tietuelähteenä. Lomakkeiden kyselyt luodaan käyttämällä Valitse-kyselymuotoa.

Hakulomakkeella on tarkoitus suodattaa tietoa. Kyselystä tarvitaan ulos haussa olevat tiedot. Lomakkeeseen muodostetaan viisi eri hakuvaihtoehtoa: Laitepaikka, nimike, tuotetunnus, tuotteen nimi ja valmistaja. Hakutuloksiin tulee näiden lisäksi huonetila ja huoneen säteilyluokka. Säteilyluokalla tarkoitetaan tilassa olevaan säteily määrää. Kuvaan 10 on kerätty haussa tarvittavat tiedot.

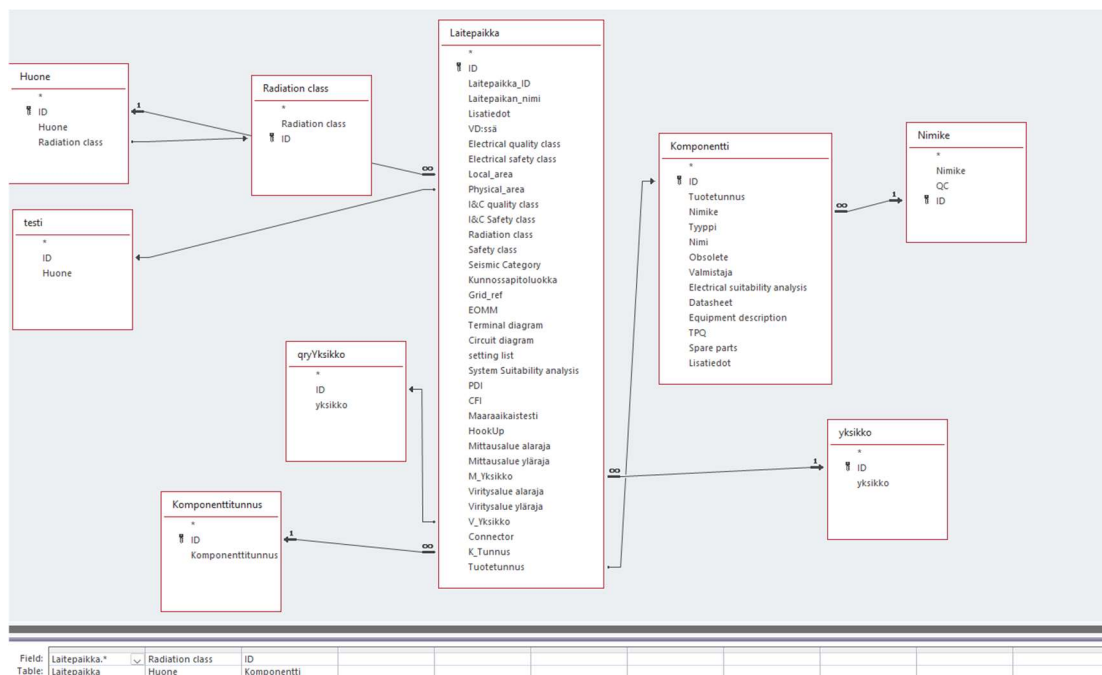


Kuva 10. Hakukentän tietuelähteen toteutus

Hakukentän kysely on toteutettu ulkoliitoksen avulla, jossa laitepaikkataulukosta otetaan kaikki tietueet. Ulkoliitos näkyy nuolena kyselyissä. Kyselyyn tehtiin alikyselyitä, jotta saadaan kyselystä yksinkertaisempi. Komponentille on luotu oma kysely, joka näkyy kuvassa 10 qryHakuChild -kyselynä. Huo-

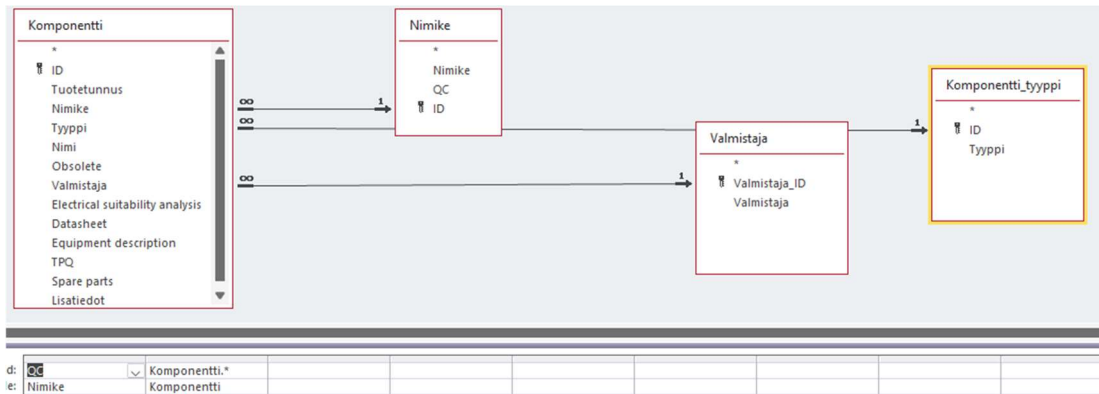
neita ovat Local\_area ja Physical\_area, joista toiselle niistä on rakennettu kysely huonetunnuksen saamiseksi. Huoneiden tunnukset olisivat samat, jos kumpaankin huoneeseen käytettäisiin samaa lähdettä.

Laitapaikkalomakkeella on tarkoitus esittää laitepaikkaan liittyviä tietoja. Laitapaikan tunnus, siihen sijoitettu komponentti ja dokumentit ovat osa mitä tulee liitettyä lomakkeeseen. Lomaketta toteutettaessa tulee esiteltä tarkemmin esiintyvät tiedot. Kuvaan 11 on kerätty laitepaikkaan tulevat tiedot. Laitepaikka.\* tarkoittaa laitepaikkataulukon kaikkien kenttien näkymistä kyselytuloksissa.



Kuva 11. Laitepaikan tietuelähteen kysely

Komponenttilomakkeella esitetään komponentteihin liittyviä tietoja. Kyseisiin tietoihin kuuluvat esimerkiksi tuotetunnus, valmistaja ja nimi. Kuvassa 12 on Komponenttilomakkeessa käytettävä kysely. QC:llä tarkoitetaan nimikkeen hyväksymistä, eli nimikkeen tietojen varmentamista oikeaksi.



Kuva 12. Komponentin tietuelähteen kysely.

#### 4.3.2 Alilomakkeiden toteutus

Lomakkeisiin tulee monia alilomakkeita. Laitepaikkalomakkeeseen tarvitaan kysely, jolla nähdään kaikki siihen laitepaikkaan asennetut komponentit. Komponenttilomakkeeseen tarvitaan kysely, jolla nähdään komponentin asennuspaikat. Sen lisäksi kumpaakin lomakkeeseen tarvitaan useampi lomake, joilla saadaan lisättyä taulukoihin tietoa.

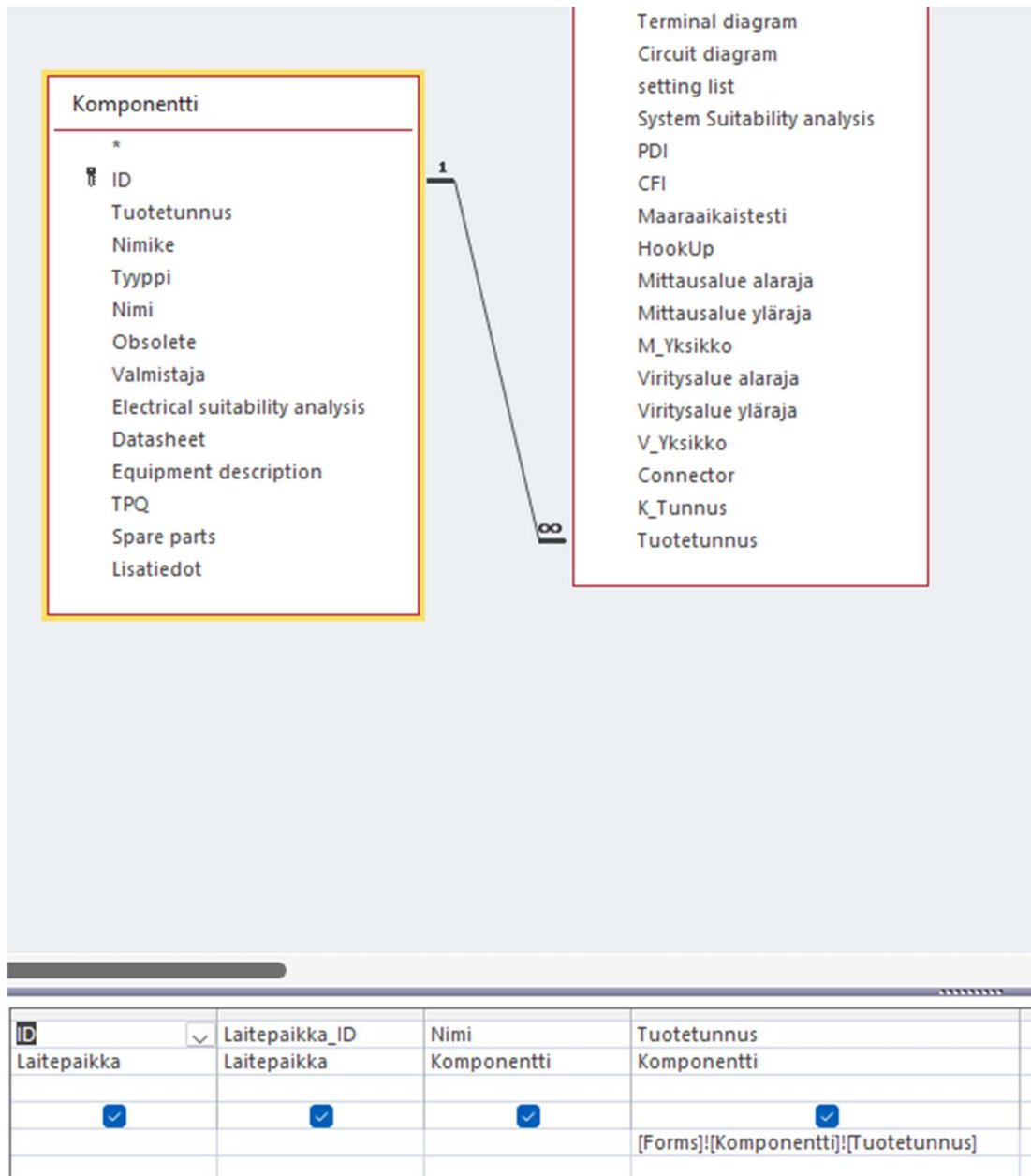
Kuten edellisessä kappaleessa mainittiin, laitepaikkalomakkeessa tulee näkyä kaikki siihen laitepaikkaan kuuluvat komponentit. Kyseinen kysely saadaan toteutettua sisäliitoksen avulla, jossa laitepaikkataulukon tuotetunnuskenttä yhdistetään komponenttitaulukon ID-kenttään. Näkyville halutaan Laitepaikan ID-kenttä, laitepaikka ja tuotetunnus. Laitepaikkaan tarvitaan myös ehto, jolla näytetään vain kyseisen laitepaikan komponentit. Ehto saadaan tehtyä, määrittelemällä laitepaikkalomakkeessa olevan laitepaikkakentän arvon ehdoksi. Kyseinen lomake on määritetty aukeavan taulukkonäkymässä. Kuvaan 13 on mallinnettu kysely.

The screenshot shows a software interface with a list of fields on the left and a 'Komponentti' form on the right. A line connects the 'Tuotetunnus' field in the list to the 'Komponentti' form. Below the interface is a table with query criteria.

Field:	Table:	Sort:	Show:	Criteria:
Laitapaikka ID	Laitapaikka		<input checked="" type="checkbox"/>	[Forms]![Laitapaikka]![IDkentta]
Tuotetunnus	Komponentti		<input checked="" type="checkbox"/>	
ID	Laitapaikka		<input checked="" type="checkbox"/>	
			<input type="checkbox"/>	

Kuva 13. Laitepaikkalomakkeen alilomakkeen kysely.

Komponenttilomakkeen alilomakkeelle muodostetaan kysely, jolla saadaan kyseisen komponentin asennuspaikat esitettyä. Kyselyyn otetaan mallia kuvasta 13. Siinä tarvitsee siirtää ehto tuotetunnukselle ja lisätä komponentti-  
taulukon Nimi-kenttä kyselyyn. Kuvaan 14 on toteutettu kysely.



Kuva 14. Komponenttilomakkeen alilomakkeen kysely

Laitapaikka- ja komponenttilomakkeiden yhdistelmäruuduille tulee luoda mahdollisuus lisätä uusi tieto, mikäli sitä ei ole listassa. Se saadaan toteutettua asettamalla ehto Mikäli ei listassa -kohtaan. Ehtoon määritellään avattavaksi lisäämislomake, joka lisää tiedon taulukkoon. Laitapaikkalomakkeesta Komponenttitunnus-, Local area – ja Physical area -kentille ja komponenttilomakkeesta Valmistaja-, Nimike-, ja Tyyppi-kentille tulee luoda lomakkeet. Lomakkeisiin liitetään haluttu taulukko ja se esitetään lomakenäkymässä. Tietojen syöttötilaksi täytyy asettaa lisääminen, joka määritetään kappaleissa 4.5.3 ja 4.6.2.

#### 4.4 Hakulomakkeen toteutus

Kuten kappaleessa 4.3.1 mainittiin, hakuvaihtoehtoja on viisi: Laitapaikka, nimike, tuotetunnus, tuotteen nimi ja valmistaja. Kyseisillä hakuvaihtoehdoilla saadaan tuotettua tarpeeksi tarkka hakutulokset.

Hakutulokset voidaan toteuttaa kahdella eri tavalla. Lomakkeeseen voidaan sijoittaa kysely tai tehdään lomakkeesta jatkolomake ja sijoitetaan tekstilaatikot kentiksi. Jatkolomakkeella voidaan näyttää useita tietueita kerrallaan.

##### 4.4.1 Lomakkeen kenttien sijoittaminen

Hakulomake toteutetaan jatkolomakkeen avulla. Jatkolomakkeessa Tiedot-kohtaan lisätään hakutulosten kentät. Kaikki muut objektit täytyy sijoittaa ylä- tai alatunnisteeseen. Hakukentät ja painikkeet ovat sijoitettu ylä- tai alatunnisteeseen. Rajaamalla Tiedot-kentän korkeuden tekstikenttien korkeudeksi saadaan toteutettua hakutulokset halutulla tavalla. Kuvassa 15 on sijoitettu kentät ja painikkeet rakennäkymässä. Huone.Huone -kentän oikealla puolella olevat kentät ja alatunnisteessa oleva kenttä on piilotettu lomakenäkymässä, koska niitä käytetään apuna toteutuksessa, mutta niiden ei tarvitse olla näkyvissä.

The screenshot shows a web form with the following elements:

- Form Header:** Search criteria fields for Laitapaikka, Nimi, Valmistaja, and Tunnus, all currently showing 'Unbound'. Action buttons: Etsi, Tuo Excelistä, Tyhjennä.
- Detail:** A table with columns: Laitapaikka, Tuotetunnus, Nimi, Nimike, Valmistaja, Local area, Physical area.
- Form Footer:** A section containing a 'Text112' field and a 'qryHakuChild.ID' field.

Kuva 15. Hakukentän ulkonäkö.

Kenttien sisällä näkyvät nimet ovat kyseiseen kenttään määritetty lähde. Lähde voidaan ottaa taulukosta tai kyselystä ja kentän arvot sijoitetaan tekstikenttään. Lähde määritetään kentän Ominaisuus-ikkunan Tiedot-välilehdeltä.

#### 4.4.2 Toiminnallisuuden toteuttaminen

Jotta hakukentillä voidaan suodattaa tietueita, tulee tietuelähteeseen lisätä ehtoja jokaiselle kentälle. Jokaiselle kentälle tulee asettaa samanlainen ehto, jonka avulla suodatus toimii. Ehdossa tulee huomioida myös tyhjien kenttien näyttäminen vain, jos hakukenttä on tyhjä. Hakukentän tietuelähteeseen tulee sijoittaa ehdot, jotta saadaan haku toimimaan. Ennen ehtojen toteuttamista, kannattaa antaa jokaiselle kentälle kuvaava nimi. Nimeämisen avulla on helpompi viitata oikeaan kenttään.

Kyselyn Laitapaikka-, Nimike-, Komponentti-, Nimi- ja Valmistaja-kohtiin tul- laan lisäämään sama ehto eri kenttien nimillä. Suodattaminen toteutetaan Like-komennon avulla, joka tarkistaa sisältääkö hakutulosten kenttä haku- kenttään kirjoitetun merkkijonon. Nz sijoittaa Null arvon tilalle tyhjän merkki- jonon.

```
Like "*" & Forms!Hakukentat!Nimikentta & "*" Or  
Nz (Forms!Hakukentat!Nimikentta, "") = "" );
```

Kyseisellä ehdolla saadaan toteutettua haluttu tulos. Forms!Hakukentat!Nimikentta:llä viitataan hakukentät-lomakkeen Nimikentta- tekstilaatikkoon. Muiden kenttien ehdoissa vaihdetaan Nimikentta halutun hakukentän nimeen.

Etsi-painikkeella toteutetaan kysely, jolloin hakukenttiin sijoitettuja arvoja etsi- tään kyselystä. Se toteutetaan VBA-kielellä komennolla:

```
DoCmd.Requery (qryHaku)
```

Painike on myös asetettu oletukseksi, jolloin näppäimistöllä Enter-painikkeen painaminen vaikuttaa kyseiseen painikkeeseen.

Tyhjennä-painike toimii muuten samalla tavalla, mutta se ensin sijoittaa tyh- jän arvon jokaiseen kenttään. Se toteutetaan komennolla:

```
Laitepaikkakentta = ""  
Nimikekentta = ""  
Nimikentta = ""  
Komponenttikentta = ""  
Valmistajakentta = ""
```

DoCmd.Requery (qryHaku)

Lomakkeesta pitää pystyä siirtymään laitepaikka- ja komponenttilomakkeisiin. Se tapahtuu asettamalla hakutulosten Laitepaikka- ja Komponentti-kenttiin määrittäminen, joka avaa kyseiset lomakkeet. Määrittäminen pystytään tekemään joko makron tai VBA:n avulla.

Siirtymät tässä tietokannassa on tehty makrojen avulla. OpenForm-komennolla saadaan avattua tietty lomake ja siihen määritellään avattava lomake, näkymä, ehdot ja tietojen syöttötila. Näkymällä tarkoitetaan, onko avattava lomake esimerkiksi lomake- tai rakennenäkyvässä. Tietojen syöttötilalla tarkoitetaan, avautuuko lomake esimerkiksi lisää- tai vain luku -näkyvässä. Kuvaan 16 on määritetty lomakkeen avautuminen. Ehtoon on määritetty komponenttilomakkeen avautuminen siihen ID-arvoon, mikä on hakutuloksessa. Laitepaikkalomakkeelle on määritetty vastaava makro, missä on vaihdettu komponentti-ID:t laitepaikka-ID:ksi.

OpenForm	
Form Name	Komponentti
View	Form
Filter Name	
Where Condition	= [Komponentti.ID]=[Forms]![Hakukentat]![IDK]
Data Mode	
Window Mode	Normal

Kuva 16. Komponenttilomakkeen avaaminen.

#### 4.4.3 Excelin tuominen Accessiin

Excelin tuomiseksi Accessiin on luotu oma painike. Sen tarkoituksena on ensin kysyä, mikä Excel-tiedosto tuodaan. Tiedosto tuodaan ensin temp-tilaan, josta sen tiedot siirretään muihin taulukoihin kyselyiden avulla. Excel-tiedostossa ja Accessin temp-tilassa tulee olla samat sarakkeiden otsikot, jotta tuominen onnistuu.

Kyselyinä käytetään Append- ja Update-kyselyitä. Ensin temp-tilasta siirretään uudet tiedot muihin taulukoihin. Mikäli jokin kentän arvo on viitattuna



ID-kentän arvolla toisessa taulukossa, sen arvo päivitetään temp-taulukossa ID-kentän arvoksi.

Tietojen lisäämiseksi toisiin taulukoihin käytetään Append-kyselyä. Taulukoihin tulee ensin määrittää indeksi, jolla estetään duplikaattien lisääminen. Indeksiin määritetään halutut kentät ja määritetään Yksilölliset arvot -kohtaan kyllä. Esimerkiksi komponentin indeksiin kuuluvat tuotetunnus, nimike ja nimi. Kyseiset kentät on määritetty, koska tuotetunnuksella voi olla eri teknisiä arvoja. Kuvan 17 kyselyssä siirretään komponenttitunnuksen arvo komponenttitunnus-taulukkoon. Lisäksi siihen on määritetty, että lisättävä arvo ei voi olla NULL.

Field:	Komponenttitunn
Table:	temp
Sort:	
Append To:	Komponenttitunnus
Criteria:	Is Not Null

Kuva 17. Append-kysely

Komponenttitunnus on myös ID-arvona laitepaikkataulukossa. Sen takia temp-taulukon arvo tulee päivittää ID-kentän arvoksi, joka toteutetaan Update-kyselyllä. Kyselyssä kummankin taulukon komponenttitunnus-kentät yhdistetään sisäliitoksella. Kuvassa 15 on muodostettu kysely, jossa vaihdetaan temp-taulukon komponenttitunnus-kentän arvoksi komponenttitunnus-taulukon ID-kenttä.

Field:	Komponenttitunnus
Table:	temp
Update To:	[Komponenttitunnus].[ID]
Criteria:	

Kuva 18. Update-kysely

Append-kysely täytyy toteuttaa jokaiselle taulukolle erikseen. Sen lisäksi Update-kysely tarvitsee toteuttaa niille temp-taulukon kentille, jotka tarvitsevat kääntää ID-kentän arvoksi.

Kyselyt yhdistetään VBA:n avulla. Kyselyiden järjestyksellä on merkitystä tietojen saamiseksi oikein, esimerkiksi Append-kyselyt tulee olla ennen Update-kyselyitä. Kuvaan 8 määritetyt yhdistykset määrittävät suoritusjärjestyksen. Yksi-puolen taulukot täytyy suorittaa ennen monta-puolta. Muuten yksi-

puolen taulukkoon sijoitetaan ID-kentän arvo, mikäli esimerkiksi komponentti-tunnus on jo sijoitettuna taulukkoon. Tässä koodissa käytetään Execute-komentoa OpenQuery-komennon sijasta, koska sillä saadaan poistettua varoitukset käytöstä. Kyselyt kannattaa ensin kokeilla virheiden varalta.

```
CurrentDb.Execute "qryK_tunnusAdd"
CurrentDb.Execute "qryK_tunnusConv"
```

VBA-koodin tarkoituksena on ensin kysyä käyttäjältä käytettävä Excel-tiedosto. Kyseisen Excel-tiedoston polku tallennetaan tekstikenttään, joka sijoitetaan hakulomakkeeseen ja laitetaan näkymättömäksi. Sitä polkua tarvitaan tietojen tuomiseksi temp-taulukkoon. Excel-tiedostosta on luotu pohja, jonka avulla tietojen siirtäminen onnistuu. Polun hakeminen kannattaa tehdä, koska muuten Excel-tiedosto täytyy pysyä yhdessä paikassa tuomisen onnistumiseksi.

Ensin tarvitsee luoda keino selvittää tiedoston sijainti. Ennen koodin kirjoittamista, pitää ottaa käyttöön Microsoft Office 16.0 Object Library, joka löytyy VBA-editorin Tools-välilehden references-kohdasta. Application.FileDialog(msoFileDialogFilePicker) komennon avulla saadaan haettua tiedoston polku, joka syötetään muuttujaan. Komennolle on myös määritetty ehto, jolla voidaan tuoda vain yksi tiedosto. Syöttäminen tapahtuu For Each avulla, jossa tekstikentän arvoksi syötetään tiedoston polku.

Taulukon tuominen tapahtuu DoCmd.TransferSpreadsheet:in avulla.

```
DoCmd.TransferSpreadsheet acImport, acSpreadsheetTypeExcel12, tableName, fileName, True
```

Komennolla tuodaan Excel-tiedosto Accessiin. TableNameen tilalle sijoitetaan temp-taulukko ja fileNameen tilalle tekstikentän arvo.

#### 4.5 Laitepaikkalomakkeen toteutus

Laitepaikkalomakkeen on tarkoitus esittää laitepaikkaan liittyviä tietoja. Lomakkeeseen tulee muun muassa laitepaikan tunnus, nimi, komponentti ja dokumentit. Tarkoituksena on esittää tiedot selkeästi ja kattavasti, jotta pystytään löytämään haluttavat tiedot nopeasti. Esimerkiksi vikatilanteissa on tär-

keää varmentaa oikean laitteen asennus oikeaan paikkaan. Asennusta varten voi olla parametreja ja asennusohjeita, joilla saadaan asennettua se oikealla tavalla. Kaikki nämä tiedot kannattaa sisällyttää yhteen paikkaan, jotta voidaan käyttää vähemmän aikaa etsimiseen.

Lomakkeen suunnitelmaa tullaan hyödyntämään pienillä muutoksilla. Kuvassa 5 olevia siirtopainikkeita ei lisätä lomakkeeseen, koska tiedot sai selkeästi esille. Lisäksi tietojen sijainteihin tehtiin muutoksia, jotta lomakkeesta tulisi luettavampi.

#### 4.5.1 Laitapaikkalomakkeen ulkonäön toteuttaminen

Laitapaikkalomakkeen toteutus alkaa tuomalla tietuelähteeksi kysely, joka tehtiin kappaleessa 4.3.1. Lomakkeeseen tulevat tiedot jaetaan eri ryhmiin ja ne sijoitetaan omiin kohtiin. Jotta saadaan selkeytettyä näkymää, kukin ryhmä asetellaan laatikon päälle. Laatikot tulee olemaan eriväriset kuin taustaväri ja niille asetetaan erikoisefekti, jolla saadaan varjot ala- ja oikealle puolelle.

Tiedot jaetaan neljään eri ryhmään: yleiset, dokumentit, luokitukset ja mitaus- ja viritysarvot. Näiden ryhmien tiedoista yleiset ovat ensimmäisenä prioriteettina, joten se tulee olemaan vasemmalla ja ylhäällä. Muut tiedot asetellaan miten ne parhaiten sopivat. Kuva 19 on laitepaikkalomakkeen ulkonäkö.

**Laitepaikka**

Laitepaikka  Laitepaikan nimi  VD

Komponenttitunnus  Tuotetunnus  Local Area  Physical Area  Connector  ID

Laitepaikka_ID	Tuotetunnus	ID
*		(New)

Record: 1 of 1 | No Filter | Search

**Luokitukset**

Safety class

Electrical quality class

Electrical safety class

I&C quality class

I&C Safety class

Kunnossapitoluokka

Seismic Category

**Dokumentit**

PDI <input type="text"/>	Grid Ref <input type="text"/>	Terminal diagram <input type="text"/>	Maaraikaistesti <input type="text"/>
EOMM <input type="text"/>	Circuit diagram <input type="text"/>	CFI <input type="text"/>	
HookUp <input type="text"/>	Setting list <input type="text"/>	System Suitability analysis <input type="text"/>	

**Mittaus- ja viritysarvot**

Mittausalue alaraja <input type="text"/>	Mittausalue yläraja <input type="text"/>	Yksikkö <input type="text"/>
Viritysalue alaraja <input type="text"/>	Viritysalue yläraja <input type="text"/>	Yksikkö <input type="text"/>

Kuva 19. Laitepaikkalomake.

#### 4.5.2 Laitepaikkalomakkeen kenttien määrittelyt

Laitepaikkalomakkeessa on alilomake, tekstikenttiä ja yhdistelmäruutuja. Alilomake esittää samaan laitepaikkaan kuuluvia komponentteja. Alilomakkeen kyselyyn on määritetty Laitepaikka\_ID, Tuotetunnus ja Laitepaikkataulukon ID-kenttä näkyväksi. Ehdoksi on määritetty, että Laitepaikka-kentän arvoa vastaavat laitepaikat näkyvät siinä. Alilomake on luotu vetämällä kysely lomakkeeseen. Tekstikentät ovat itse sijoiteltu lomakkeeseen ja kullekin on valittu lähteeksi tietuelähteen kenttä.

Yhdistelmäruudut on luotu käyttämällä yhdistelmäruutu wizardia. Siinä ensin määritellään mistä taulukosta tai kyselystä haetaan tiedot. Sitten valitaan halutut kentät ja missä järjestyksessä ne esitetään. Yhdistelmäruudun arvon voi myös tallentaa haluttuun kenttään. Yhdistelmäruudut määriteltiin tähän, jotta saadaan luettelon kaikki kuuluvat tiedot. Sen avulla voidaan tarkastaa, löytyykö tieto jo taulukosta.

### 4.5.3 Toiminnallisuuden toteuttaminen

Laitepaikkalomakkeessa tuotetunnusta painettaessa tulee päästä komponenttilomakkeeseen kyseisen komponentin kohdalle. Toisin kuin luvussa 4.4.2 siirtyminen on tehty käyttämällä VBA:ta. Siirtyminen toteutetaan lauseella:

```
DoCmd.OpenForm "komponentti", , , "Komponentti.ID=" &
_Forms!Laitepaikka!IDK
```

Jossa FORMS:in edellä oleva \_ tarkoittaa rivinvaihtoa. Hipsujen sisällä oleva komponentti on avattava lomake ja kolmen pilkun jälkeen oleva koodinpätkä on ehto, jolla avataan tietty komponentti. Alilomakkeessa olevan Laitepaikka\_ID saattaa sisältää useampia komponentteja, jolloin niiden välillä voidaan siirtyä Laitepaikka\_ID-kenttää painamalla.

Yhdistelmäruutuihin on asetettu sääntö, joka avaa lisäämislomakkeen, mikäli esimerkiksi huonetta ei ole taulukossa. Se toteutetaan asettamalla makro tai koodi yhdistelmäruudun ominaisuusikkunan Tapahtumat-välilehden Luettelosta puuttuessa -kohtaan. Kyseinen sääntö on tehty makrolla ja se avaa halutun lomakkeen lisäämistilassa. Esimerkiksi Local area -kenttään syöttäessä arvon mitä siitä ei löydy, Huone-taulukosta tehty lomake aukeaa ja siihen voi lisätä halutun huoneen.

Painikkeista Uusi ja Tallenna on luotu Ohjattu komentopainikkeen luomisen avulla. Uusi-painikkeeseen on määritetty Lisää uusi tietue ja Tallenna-painikkeeseen on määritetty Tallenna tietue. Kuvat-painikkeen tarkoituksena on mennä nettisivulle, joka sisältää kuvia laitepaikoista. VBA:n avulla saadaan avattua nettisivu käyttämällä Application.FollowHyperlink -komentoa. Kyseisen sivun kansiorakenteen takia, osoitteeseen pitää lukea myös laitepaikan tunnusta. Se toteutetaan Left-komennon avulla, jolla voidaan määrittää, kuinka monta merkkiä luetaan.

## 4.6 Komponenttilomakkeen muodostaminen

Komponenttilomake esittää komponentteihin liittyviä tietoja. Komponenttien tiedot koostetaan yhteen paikkaan, jotta niiden löytäminen olisi nopeampaa. Siihen kerätään esimerkiksi tuotetunnus, valmistaja, nimike ja asennuspaikat. Tuotetunnuksella tarkoitetaan komponentilla olevaa tunnistetta, joka voi olla esimerkiksi merkkijono. Nimikkeellä saadaan varmistettua komponentin olevan oikea. Osa tuotetunnuksista saattaa olla erilaisia teknisiltä tiedoilta, joten nimikkeellä varmistetaan oikeellisuus. Komponenttien tiedoille on monia eri käyttökohteita, joista edellä olevat on osa niistä.

### 4.6.1 Komponenttilomakkeen ulkonäön toteuttaminen.

Komponenttilomakkeessa on vähemmän tietoa verrattuna laitepaikkalomakkeeseen, joten tietojen ryhmittely on helpompaa. Lomakkeeseen tulee dokumentit, yleiset tiedot ja asennuspaikat. Asennuspaikat on tuotu alilomakkeeksi lomakkeeseen. Sen tarkoituksena on esittää mihin laitepaikkoihin kyseistä komponenttia on asennettu. Kuvaan 20 on aseteltu haluttavat tiedot.

Suunnitelmassa olleita painikkeita ei laitettu lomakkeeseen, koska se ei ollut tarpeellista haluttavien tietojen perusteella. Tietojen vähyyden takia kentät sai sijoitettua lomakkeeseen selkeästi.

**Komponentti**

Komponenttitunnus	<input type="text"/>	Valmistaja	<input type="text"/>	Tyyppi	<input type="text"/>
Nimi	<input type="text"/>	Nimike	<input type="text"/>	QC	<input type="checkbox"/>
		Obsolete	<input type="text"/>	ID	<input type="text"/>

Lisätiedot

ID	Laittepaikka_ID	Nimi	Tuotetunnus
*	(New)		

Record: 1 of 1    No Filter    Search

**Dokumentit**

Electrical suitability analysis

Datasheet

Equipment description

TPQ

Kuva 20. Komponenttilomakkeen ulkonäkö.

#### 4.6.2 Komponenttilomakkeen toiminnallisuuden toteuttaminen

Toiminnallisuus tehdään samalla tyylillä laitepaikkalomakkeeseen verrattuna. Nimike-, Valmistaja- ja Tyyppi-kentässä käytetään hyväksi luvussa 4.5.3 esiteltyä yhdistelmäruutuihin lisäämistä. Alilomakkeen Laittepaikka\_ID-kentästä tulee päästä kyseiseen laitepaikkaan.

Kuvassa 20 näkyvät Uusi- ja Tallenna-painikkeet ovat luotu samalla tyylillä kuin 4.5.3, käyttäen hyväksi Ohjattu komentopainikkeen luomista.

## 5 YHTEENVETO

Tavoitteena tällä opinnäytetyöllä oli rakentaa Olkiluoto 3 kenttälaitteiden datasta tietokanta. Sen tarkoituksena on kerätä kenttälaitteisiin liittyvät tiedot yhteen paikkaan ja täten kokeilla tietokantaa automaatiotekniikan työkaluna.

Tietokannan rakentaminen onnistui luvussa 3 käydyn teorian perusteella. Teorian läpi käymisessä käytettiin pitkälti Microsoftin luomaa Accessin ohjesivustoa, josta löytyy perusteita tietokannan luomiseksi. Luvussa 4 toteutettiin rakentaminen ja siinä esiteltiin keinoja tiettyjen ominaisuuksien luomiseksi. Toteutuksessa käytiin ensin tietojen määrittelyt ja tietokannan suunnittelua. Niiden perusteella tietokantaan rakennettiin taulukot, kyselyt ja lomakkeet ja niistä tehtiin toimiva käyttöliittymä.

Tavoitteeseen päästiin ja saatiin rakennettua tietokanta kenttälaitteille. Tietokantaan rakennettiin keino hallita dataa ja ratkaisulla saatiin keino ratkaista vanhan tallennustavan ongelmat. Kenttälaitteiden tietojen määrittelyille otettiin huomioon kunnossapidon ja tekniikan toiveet ja niiden sijoittelussa käytettiin tietojen priorisointia.

Opinnäytetyön tekemisessä tuli opittua relaatiotietokantaa liittyvää teoriaa sekä Accessin käyttöä. Access ohjelmalla en ollut ennen rakentanut tietokantaa, joten tutustuin sen ominaisuuksiin ennen työn alkua. Työn tekemisessä tuli huomattua tietokannan rakentamisen vaiheiden tärkeys. Ennen seuraavaan vaiheeseen siirtymistä kannattaa toteuttaa valmiiksi kyseinen vaihe. Toteutuksessa myös huomasi suunnittelun tärkeyden. Tietokannan taulukoiden sisällöt ja lomakkeiden ulkonäkö oli helppo toteuttaa suunnitelman pohjalta.



Tätä tietokantaa tarvitsee vielä kehittää ja lisätä loputkin tiedot. Tähän tietokantaan tuotiin ainoastaan noin neljännes kaikista kenttälaitteista. Sen lisäksi tiedoissa oli puutteita etenkin dokumenttien osalta. Ennen muiden tietojen lisäämistä tulee selvittää, tarvitaanko tietokantaan tehdä jotain muutoksia. Käyttäjiltäkin saatavan palautteen perusteella voi myös tulla kehitysideoita. Esimerkiksi tätä tietokantaa ei olla testattu monen samanaikaisen käyttäjän toimesta. Tulevaisuudessa tulee myös miettiä kannattaako rakentaa suoraan tietokantasovellus kyseisen Access tietokannan tilalle. Access toimii nykyisellä data- ja käyttäjämäärällä, mutta mikäli määrät lisääntyvät merkittävästi, tietokannan käyttämisessä voi tulla ongelmia.

## LÄHTEET

Haltu. (14.3.2023). Käyttöliittymäsuunnittelu - mitä se on ja onko siitä hyötyä?  
<https://www.haltu.fi/blogi/kayttoliittymasuunnittelu>

Microsoft. (n.d.a). Johdanto taulukoihin. Haettu 20.10.2023 osoitteesta  
<https://support.microsoft.com/fi-fi/office/johdanto-tilakkeihin-78ff21ea-2f76-4fb0-8af6-c318d1ee0ea7>

Microsoft. (n.d.b). Kyselyiden esittely. Haettu 7.12.2023 osoitteesta  
<https://support.microsoft.com/fi-fi/office/kyselyiden-esittely-a9739a09-d3ff-4f36-8ac3-5760249fb65c>

Microsoft. (n.d.c). Taulukoiden ja kyselyjen liittäminen. Haettu 7.12.2023 osoitteesta  
<https://support.microsoft.com/fi-fi/office/taulukoiden-ja-kyselyjen-liitt%C3%A4minen-3f5838bd-24a0-4832-9bc1-07061a1478f6>

Microsoft. (n.d.d). Taulukoiden yhteyksien opas. Haettu 20.10.2023 osoitteesta  
<https://support.microsoft.com/fi-fi/office/taulukoiden-yhteyksien-opas-30446197-4fbe-457b-b992-2f6fb812b58f>

Microsoft. (n.d.e). Yhteyden luominen, muokkaaminen tai poistaminen. Haettu 23.1.2024 osoitteesta  
[https://support.microsoft.com/fi-fi/topic/yhteyden-luominen-muokkaaminen-tai-poistaminen-dfa453a7-0b6d-4c34-a128-fdebc7e686af#\\_enforce\\_referential\\_integrity](https://support.microsoft.com/fi-fi/topic/yhteyden-luominen-muokkaaminen-tai-poistaminen-dfa453a7-0b6d-4c34-a128-fdebc7e686af#_enforce_referential_integrity)

Nielsen. J. (30.1.2024). 10 Usability Heuristics for User Interface Design.  
<https://www.nngroup.com/articles/ten-usability-heuristics/>

Rika sensor. (19.11.2021). What is the difference between a sensor and a transmitter?.  
<https://www.rikasensor.com/what-is-the-difference-between-a-sensor-and-a-transmitter.html>

Teollisuuden Voima Oyj. (14.3.2018). Ulkokuvat OL1 katolla [valokuva]. TVO.  
<https://www.tvo.fi/media/materiaalipankki.html>

Teollisuuden Voima Oyj. (n.d.a). OL3. Haettu 12.10.2023 osoitteesta  
<https://www.tvo.fi/tuotanto/laitosyksikot/ol3.html>

Teollisuuden Voima Oyj. (n.d.b). TVO-konserni. Haettu 6.10.2023 osoitteesta  
<https://www.tvo.fi/yhtio/hallintojajohtaminen/tvo-konserni.html>

Tukes. (22.12.2021). Turva-automaatio prosessiteollisuudessa.  
<https://tukes.fi/turva-automaatio-prosessiteollisuudessa>