

Katriina Pennanen

TIETOKANNAN SUUNNITTELU ELEHUOLTO OY:LLE

TIETOKANNAN SUUNNITTELU ELEHUOLTO OY:LLE

Katriina Pennanen
Opinnäytetyö
Kevät 2015
Tietojenkäsittely
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittely, Internet-palvelut ja digitaalinen media

Tekijä(t): Katriina Pennanen

Opinnäytetyön nimi: Tietokannan suunnittelu Elehuolto Oy:lle

Työn ohjaaja: Sinikka Viinikka

Työn valmistumislukukausi- ja vuosi: Kevät 2015

Sivumäärä: 42 + 9

Elehuolto on raahelainen yritys, joka on erikoistunut ajoneuvojen elektronisten ohjainlaitteiden korjaukseen. Yritys tarvitsi työntekijöiden käyttöön tietokannan, jonka avulla ohjainlaitteiden, korjausohjeiden ja vikakuvausten hallinnointi on helpompaa ja nopeampaa. Tehtävänäni oli suunnitella heille tarkoitukseen sopiva tietokanta. Tietokantaa käytetään ainoastaan Elehuollon toimitiloissa.

Tietoperustassa kerrottiin vaatimusmäärittelystä, tietokannoista ja tietokannan hallintajärjestelmästä sekä tietomalleista. Vaatimusmäärittelyn osalta on käyty läpi vaatimusten keräämiseen liittyviä asioita, oikeita menetelmiä oikeiden vastausten saamiseen, vaatimustyön vaikeuksia, toiminnallisia ja ei-toiminnallisia vaatimuksia.

Tietoperustaa varten tietoa löytyi sekä tuoreista englanninkielisistä julkaisuista, että suomenkielisestä kirjallisuudesta. Tietokantoihin liittyvää kirjallisuutta on saatavilla rajallisesti suomenkielellä ja kirjat ovat aika vanhoja. Relatiotietokantojen peruseriaatteet ovat kuitenkin pysyneet samoina, vaikka teknologia on muuttunut ja uusia tietokantojen hallintajärjestelmiä on tullut markkinoille paljon. Englannin kielistä kirjallisuutta sen sijaan löytyi paljon.

Opinnäytetyön tuloksena syntyi vaatimusmäärittely, käsitelmä, taulujen kuvaukset ja taulujen luontilauseet. Näiden avulla suunniteltu tietokanta voidaan myöhemmin toteuttaa. Accessilla toteutettu kokeiluversio on kuitenkin mahdollista siirtää yrityksen käyttöön pienien muutoksien jälkeen.

Asiasanat: Tietokanta, tietojärjestelmä, vaatimusmäärittely, käsitelmäanalyysi

ABSTRACT

Oulu University of Applied Sciences
Degree programme of business information systems, internet services and digital media

Author(s): Katriina Pennanen

Title of thesis: Data base design for Elehuolto Oy

Supervisor(s): Sinikka Viinikka

Term and year when the thesis was submitted: Spring 2015 Number of pages: 42 + 9

Elehuolto Oy based in Raahе, specializes in automotive electronic control module recoding and repairs. This is a fast growing company and a specialized database is needed for the management of spare component, repairs manual and customer service. The aim of this thesis is to develop a customized database to be applied in a business model such as Elehuolto Oy.

The knowledge base of this thesis consists of the requirements definition, data systems and ER modeling. Recently published English and Finnish literature make up the sources of this knowledge base. There is a limit to the amount of Finnish literature that is germane, hence English literature makes up the aggregate of this knowledge base.

To be defined in this thesis are the methods of collection, difficulties in gathering the required data, as well as the functional and non-functional requirements of the database.

The requirements, ER modeling, table descriptions and clauses necessary to create this database are all provided, and as a result a transferrable trial version of the database. This is a database that a company would be able to transfer to their server with only minor changes.

Keywords: Database, data system, requirement definition, ER modelling

SISÄLLYS

1	JOHDANTO	6
2	TOIMEKSIANTAJA JA TOIMEKSIANTO.....	7
2.1	Toimeksiantaja	7
2.1	Toimeksianto	7
3	VAATIMUSMÄÄRITTELY	9
3.1	Vaatimusten kerääminen.....	9
3.2	Menetelmiä oikeiden vastausten saamiseksi haastatteluista.....	10
3.3	Määrittelytyön vaikeuksia	12
3.4	Toiminnalliset ja ei-toiminnalliset vaatimukset	13
3.5	Vaatimusmäärittelyn dokumentointi.....	14
4	TIETOKANTA JA TIETOKANNAN HALLINTAJÄRJESTELMÄ	17
4.1	Tietokannan hallintajärjestelmä	17
4.2	Tietokannan rakenne.....	20
5	TIETOKANNAN SUUNNITTELU JA TIETOMALLIT	22
5.1	Käsiteanalyysi	23
5.2	Tietomallit	24
5.3	Käsitteet	24
5.4	Tiedot	25
5.5	Avaimet	26
5.6	Käsitteiden väliset yhteydet.....	27
5.7	UML:n luokkakaavio	30
5.8	Tarveanalyysi	33
5.9	Normalisointi.....	35
6	TIETOKANNAN RAKENTEEN TESTAUS.....	36
7	POHDINTA.....	39
	LÄHTEET.....	41
	LIITTEET	39

1 JOHDANTO

Viime vuosina elektroniikan osuus ajoneuvoissa on kasvanut valtavasti. Erilaiset ohjainlaitteet ohjaavat ajoneuvon toimintoja. Näistä ehkä tärkeimpiä ovat moottoritoimintojen-, jarrujen- ja turvatyynyjen ohjaus sekä monet muut toiminnot, jotka vaativat nykyaikaisia tietokonepohjaisia ohjainlaitteita. Kuluttajan kannalta ongelmallista on se, että näitä ohjainlaitteita on lähes mahdoton korjata itse. Autokorjaamossa rikkoutunut osa vaihdetaan yleensä uuteen. Tämä on kallista ja aikaa vievää, koska usein rikkoontuneen ohjainlaitteen tilalle joudutaan tilaamaan uusi laite ulkomailta.

Elehuolto Oy on Raahessa toimiva yritys, joka on erikoistunut korjaamaan autoelektroniikkaa. Yritys on perustettu vuonna 2013 ja se työllistää tällä hetkellä neljä henkilöä. Korjausmäärän kasvun takia yritys tarvitsee työntekijöitä varten tietokantaa korjausohjeille, vikakuvauksille ja ohjainlaitteille. Tulevan tietokannan avulla työntekijä löytää nopeasti oikean korjausohjeen ja yleisimmät vikakuvaukset korjattavana olevalle ohjainlaitteille.

Opinnäytetyön tavoitteena on suunnitella korjausohjetietokanta Elehuollon työntekijöiden käyttöön. Yrityksellä ei ole tällä hetkellä käytössä tietokantaa, johon henkilöstö voisi tallentaa laatimaan korjausohjeita. Opinnäytetyö toteutetaan tekemällä vaatimusmäärittely, jonka pohjalta laaditaan suunnitteluputken vaiheiden mukaisesti käsitemalli, josta johdetaan taulukuvaukset ja taulujen luontilauseet. Tietokannan rakenteen toimivuutta testataan suunnittelun edetessä. Testaus toteutetaan luomalla Accessiin tietokannan taulut ja testaamalla taulurakennetta luomalla toiminnallisten vaatimusten mukaisia kyselyjä tietokantaan.

2 TOIMEKSIANTAJA JA TOIMEKSIANTO

2.1 Toimeksiantaja

Opinnäytetyön toimeksiantaja on Elehuolto Oy. Se on raahelainen yritys, joka on perustettu vuonna 2013. Elehuolto aloitti toimintansa perustajan autotallista, josta yritys laajeni nopeasti neljän henkilön yritykseksi. Yritys korjaa ajoneuvoelektroniikkaa, kuten autojen ohjainyksiköitä ja mittaristoja sekä raskaan kaluston elektroniikkaa ja muita autoelektroniikan piirikortteja.

Autojen ohjainlaitteet ovat elektronisia yksiköitä, jotka ohjaavat ajoneuvon eri toimintoja. Näitä ohjainlaitteita löytyy merkittävä määrä nykyautoista. Perinteisesti autokorjaamot vaihtavat rikkoontuneen ohjainlaitteen tilalle uuden ohjainlaitteen. Nyt asiakas voi lähettää rikkoontuneen ohjainyksikön Elehuoltoon, jossa se korjataan ja tarvittaessa koodataan ammattitaidolla kuntoon. Ohjainlaitteiden pienestä koosta johtuen niiden toimittaminen huoltoon joko Postin tai Matkahuollon kautta on suhteellisen helppoa ja nopeaa. Tästä johtuen Elehuollon markkina-alueeseen kuuluu koko Suomi.

Elehuollon tyypillisimpiä asiakkaita ovat autokorjaamot sekä omatoimiset autokorjaajat. Autokorjaamon henkilökunnalla on ammattitaito lukea vikakoodit ja vaihtaa rikkoontuneen osan tilalle uusi tai korjattu osa. Autokorjaamoilla ei siis ole riittävää ammattitaitoa tehdä vaativaa vikadiagnosointia ohjainlaitteille, eikä kykyä korjata rikkoontuneita ohjainlaitteita. Toinen asiakaskunta on yksityiset, omatoimiset autonkorjaajat, joilla on alan tuntemusta. He voivat itse tai yhdessä Elehuollon henkilökunnan kanssa määritellä diagnostiikkaohjelmiston vikakoodeja ja paikallistaa viallisen ohjainlaitteen. Vian paikallistamisen jälkeen asiakas voi toimittaa rikkoontuneen osan Elehuoltoon korjattavaksi. Yrityksen tavoitteena on toimittaa osa korjattuna takaisin asiakkaalle kolmen päivän kuluessa asiakkaan toimituksen saapumisesta.

2.1 Toimeksianto

Yrityksen kasvun takia Elehuolto tarvitsee ohjainlaite-, vikakuvaus- ja korjausohjetietokantaa, josta työntekijä voi löytää helposti kulloinkin korjattavana olevan ohjainlaitteen tyypillisimmät vikakuvaukset ja korjausohjeet. Elehuollon työntekijät kirjaavat jokaisen korjattavana olevan ohjainyk-

sikön tyypillisimmät vikakuvaukset ja korjausohjeet PDF-muodossa. Tällä hetkellä heillä ei ole ohjeille tietokantaa, vaan vikakuvaukset ja korjausohjeet ovat kunkin työntekijän omalla tietokoneella sekalaisessa järjestyksessä. Tästä johtuen oikean korjausohjeen löytäminen työn alla olevalle ohjainlaitteelle on hankalaa ja työlästä.

Tietokantaa käytetään hyväksi jo asiakaspalvelun alkuvaiheessa, kun asiakas ensimmäisen kerran ottaa yhteyttä Elehuoltoon ongelmatilanteessa. Tässä vaiheessa työntekijä voi tarkistaa asiakkaan antamien ajoneuvon ja ongelmakuvauksen perusteella, mitkä voisivat olla kyseisen ohjainlaitteen tyypillisimmät ongelmat ja miten ne olisivat ratkaistavissa. Tämä helpottaa korjausarvion antamista asiakkaalle. Asiakkaan lähetettyä ohjainlaitteen Elehuoltoon, työntekijä etsii tietokannasta oikean korjausohjeen ohjainlaitetyypin perusteella. Joskus työntekijät tarvitsevat tietoa myös siitä, mille kaikille automalleille tietty ohjainlaitemerkki kuuluu. Elehuollolla on myös varaosakomponentteja varastossa. Työntekijät tarvitsevat varaosaohjainlaite- ja varaosakomponentti-tietokantaa, jotta he voivat tarvittaessa tarkistaa mitä varaosalaitteita ja -komponentteja heillä on saatavana. Lisäksi tarvitaan tietoa siitä, kuinka paljon varaosia on saatavana ja mihin eri ohjainlaitteisiin kyseiset varaosakomponentit käyvät.

3 VAATIMUSMÄÄRITTELY

Vaatimusmäärittely (requirements definition) on dokumentti, johon on koottu kehitettävän järjestelmän eri sidosryhmien sille asettamat vaatimukset. Vaatimukset määrittelevät eri sidosryhmien tarpeet suhteessa järjestelmään mutta eivät ota kantaa siihen, minkälainen tekninen toteutus järjestelmälle valitaan. Vaatimukset voidaan luokitella toiminnallisiin ja ei-toiminnallisiin vaatimuksiin. (Pohjonen 2007, 28.)

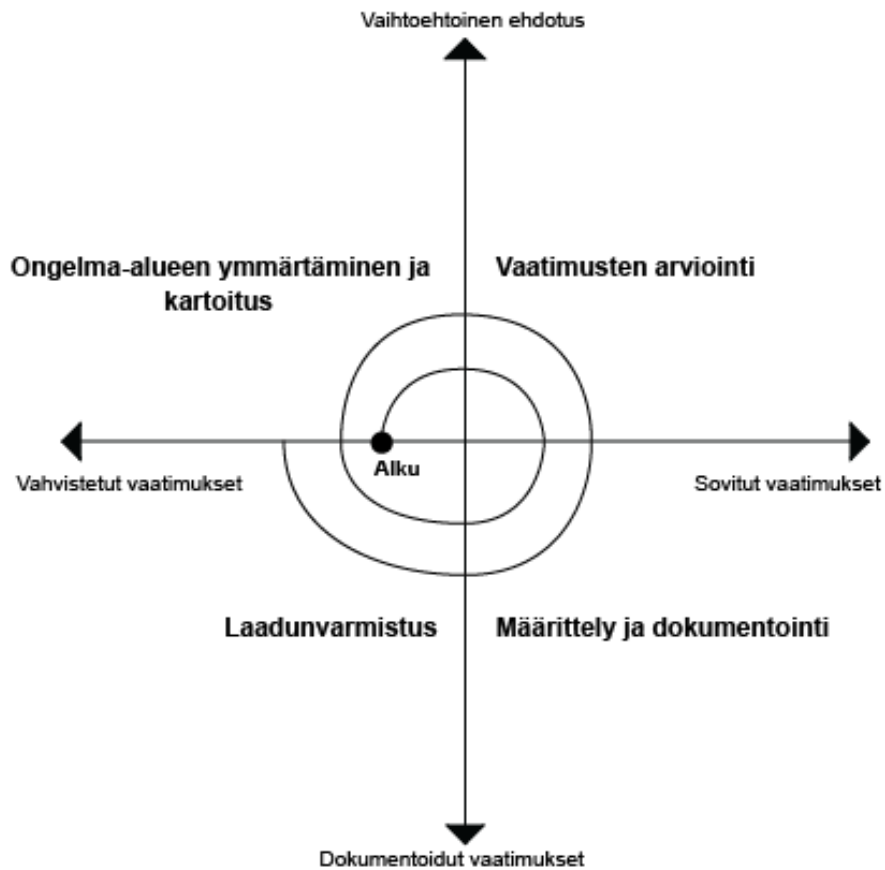
Voidakseen toteuttaa jonkin ongelman ratkaisevan järjestelmän, on ymmärrettävä ja määriteltävä, mikä ratkaistava ongelma tarkalleen ottaen on. Vaatimusmäärittelyssä vastataan kolmeen kysymykseen: mitä halutaan, miksi halutaan ja kuka kantaa vastuun? (Paakki 2011, 7.) Vaatimusmäärittely on monesti oma projektinsa ja siinä on pyrkimyksenä selvittää järjestelmän asiakasvaatimukset mahdollisimman perusteellisesti. Vaatimukset lähtevät liiketoiminnan tarpeista ja yleensä niiden tärkein lähde on asiakas. (Haikala & Mikkonen 2011, 65.)

3.1 Vaatimusten kerääminen

Vaatimusten kerääminen on vaativaa työtä, koska ei ole olemassa yhtä vaatimusten keräysmenetelmää, joka takaisi tarpeeksi kattavan lopputuloksen. Siksi vaatimusten kokoamisessa yleensä käytetään useita strategioita yhtäaikaisesti. Perinteisin tapa kerätä vaatimuksia on haastatella tulevan järjestelmän käyttäjiä. Pohjosen mukaan tällaiset haastattelut tarjoavat hyvän tilaisuuden kerätä vaatimusten lisäksi myös tärkeää taustatietoa esimerkiksi järjestelmän tulevista käyttäjistä, heidän työtehtävistään, yleisistä käytänteistä, organisaation rakenteista sekä vastuualueista. Keskeistä on löytää oikeanlaiset haastateltavat sidosryhmien sisältä. (2007, 28.)

Elehuollolle suunniteltavan tietokannan vaatimusmäärittelyä tehdessä vaatimuksia kerättiin pääasiassa haastatteleamalla yrityksen työntekijöitä. Vaatimusten kerääminen aloitettiin heti projektin alkuvaiheessa. Toimeksiantajan kanssa käytiin keskustelua tulevan tietokannan vaatimuksista. Vaatimukset listattiin aluksi karkealla tasolla ranskalaisilla viivoilla. Vaatimuksia tarkennettiin projektin edetessä. Vierailut toimeksiantajan tiloissa antoivat arvokasta tietoa työntekijöiden työtehtävistä ja siitä, mihin he tulevaa tietokantaa tarvitsevat. Lisäksi vierailut auttoivat hahmottamaan sitä, millä tavalla työntekijät tulevat käyttämään tulevaa tietokantaa. Vaatimusmäärittelyn teko

eteni pääpiirteissään kuvion 1. mukaisesti. Prosessi eteni iteratiivisesti ongelmakentän ymmärryksen ja vaatimusten tarkentuessa asteittain.



Kuvio 1. Vaatimusmäärittelyn työvaiheet (Laine 2014, viitattu 4.3.2015).

3.2 Menetelmiä oikeiden vastausten saamiseksi haastatteluista

Winantin mukaan hyvän vaatimuksen saamiseksi täytyy kysyä kysymyksiä. Tämä voi kuulostaa itsestäänselvyydeltä, mutta monesti haastattelija haluaa kysymysten sijaan antaa vastauksia. Tämä saa helposti loppukäyttäjän tuntemaan että heidän osuutensa asiaan on merkityksetön. Tärkeitä kysymyksiä hänen mukaansa on kuka, mitä, missä, miksi, milloin ja miten. Näitä kysymyksiä tulisi sitten tarkentaa. Winantin ohjeet kysymyksille ovat: älä korosta itseäsi, hae hyödyllisiä vastauksia sekä älä tee mitään oletuksia. (2002, viitattu 2.2.2015.)

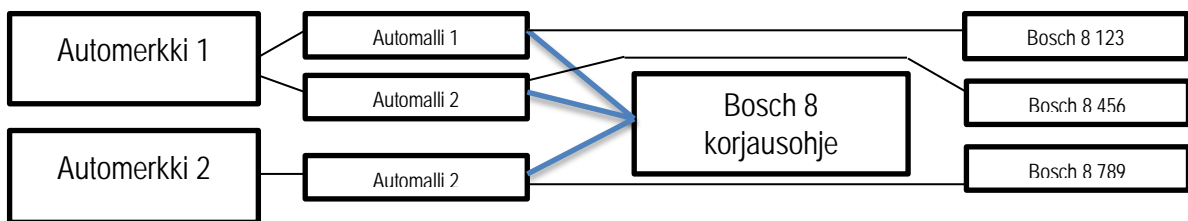
Crispin on todennut, että suunnittelijan tulisi unohtaa omat olettamukset ja sen sijaan asettua sidosryhmän maailmaan. Hänen mukaansa tulisi selvittää asiakkaan tarpeet ottamalla selvää ensiksikin mikä halutun ominaisuuden tarkoitus on. Toiseksi, kuka kyseistä toimintoa käyttää ja mitä he tekevät ominaisuuden käyttöä ennen ja mitä sen jälkeen, kun ominaisuutta on käytetty. Lisäksi tulisi selvittää, mitä tietoa käyttäjä hakee raportista. (2012, viitattu 2.2.2015.)

Halutessa laajaa näkökulmaa asioista, jotka voivat olla välttämättömiä hankkeen kannalta, tulisi selvittää mitä ongelmia järjestelmän avulla yritetään ratkaista ja mitä mahdollisia ongelmia ratkaisu voi tuoda. Tällaisia kysymyksiä kutsutaan "yhteydettömiksi" ("context-free") kysymyksiksi. Jos taas etsitään jotain erityistä tietoa, haastattelijan kannattaa kysyä kapeampia kysymyksiä. Kapeammilla kysymyksillä haetaan vastausta kysymyksiin, kuten miten tarkalleen, milloin tarkalleen ja mitä tarkalleen. Esimerkiksi miten lasku tarkalleen ottaen käsitellään? Jos haastattelija ei oikein tiedä mitä hänen tulisi tietää, voi asiakkaalta kysyä mitä kysymyksiä hänelle tulisi esittää. Tämä on ns. metakysymys ja tuo ihmistä lähemmäksi järjestelmää ja ongelmaa ja osoittaa mikä heille on tärkeää. (Winant 2002, viitattu 2.2.2015.)

Elehuollon henkilöstölle tehtyjen yhteydettömien kysymysten pohjalta saatiin kuva siitä, mitä ongelmia järjestelmän avulla yritetään ratkaista. Tietokannan avulla työntekijä saa nopeasti esille kulloinkin työn alla olevan ohjainlaitteen tyypillisimmät vikakuvaukset ja korjausohjeen. Metakysymysten avulla saatiin esimerkiksi tietoa siitä, miten tarkalleen kyseinen korjausohje tulisi olla löydettävissä tietokannasta. Työntekijä hakee oikeaa korjausohjetta auton merkin, mallin ja vuosimallin perusteella. Toisaalta haettaessa esimerkiksi pelkän automallin perusteella haku antaisi kaikki ne ohjainlaitteet jotka kuuluvat kyseiselle mallille.

Winantin mukaan haastattelijan ei tulisi tehdä olettamuksia. Sen sijaan, että tehtäisiin johdattelevia kysymyksiä, kuten ettehan pidä jostakin asiasta, tulisi avata vuoropuhelu käsiteltävästä ongelmasta. Joskus huomataan, että ongelmaksi koettu asia ei vaadikaan mitään ratkaisua. Winantin mukaan olettamuksien välttämiseksi kannattaa käyttää yhteydettömiä- ja metakysymyksiä. Tällaisilla kysymyksillä saadaan vastauksia esimerkiksi siihen, missä asiakas toivoo ominaisuutta käytettävän ja miksi kyseinen ongelma kannattaa ratkaista. Winantin mukaan asiakkaat antavat parempia vastauksia, kun heillä on tunne, että he itse ovat asiantuntijoita. (2002, viitattu 2.2.2015.)

Elehuollolle suunniteltavan tietokannan määrittelyä tehdessä mietittiin sitä, tuleeko hakuohjeissa ottaa huomioon myös auton moottorityyppi. Tämä tieto ei kuitenkaan olisi antanut mitään lisäarvoa hakuihin, joten se tieto jätettiin tietokannasta pois. Eri autovalmistajat käyttävät usein saman merkisiä ohjainlaitteita autoissaan. Esimerkiksi Bosch 8 merkinen ABS-ohjainlaite voi kuulua useampaan eri automerkkiin ja malliin. Nämä ohjainlaitteet on kategorisoitu tarkemmin osanumerolla. Osanumero viittaa tarkalleen tiettyyn automalliin ja vuosimalliin, johon ohjainlaite käy. Metakysymyksiä tehdessä ilmeni, että sama korjausohje sopii kaikkiin Bosch 8 ohjainlaitteisiin riippumatta sen osanumerosta (kuvio 2). Tästä johtuen osanumeroita ei oteta mukaan tietokantaan. Työntekijät ainoastaan listaavat eri osanumerot korjausohjeisiin.



Kuvio 2. Eri automallien ohjainlaitteille voi käydä sama korjausohje

3.3 Määrittelytyön vaikeuksia

Lehtimäen mukaan on todella vaikeaa saada selville, mitä järjestelmän tulisi tehdä eli miten sen pitäisi toimia. "Järjestelmänhän tulee toimia organisaatiossa samalla tai jopa suuremmalla ymmärryksellä kuin ne ihmiset, jotka ovat siellä vuosikausia olleet" (Lehtimäki 2006, 160). Harmillista Lehtimäen mukaan on myös se, että määrittelytyö tulee tehtäväksi heti projektin alussa. Määrittelytyö tehdään silloin, kun toiminta-alueesta on vähiten ymmärrystä. Jotta tästä haasteesta selviää, täytyy Lehtimäen mukaan tehdä kaksi asiaa: opiskella kohdealueen toimintaa ja opiskella määrittelytyötä. (2006. 161.)

Yksi tavallinen ongelma, johon vaatimuksia määriteltäessä törmätään, on vaatimusten keskenäisyys ja keskinäinen ristiriitaisuus. Vaatimusten tulisi olla helposti luettavissa, tutkittavissa ja ymmärrettävissä. Jos vaatimukset ovat epämääräisiä, ne aiheuttavat ongelmia kehitystyötä arvioitaessa. Jotta vältettäisiin epäselvyydet, vaatimukset tulisi kirjata niin tarkasti kuin mahdollista. (Pohjonen 2007, 29.)

Elehuollolle tulevaa tietokantaa määriteltäessä oli tärkeää tutustua kohdealueen toimintaan. Tutustuminen tapahtui vieraillemalla yrityksessä, haastattelemalla työntekijöitä ja tutustumalla yrityksen internet-sivuihin. Näin muodostui kuva siitä, minkälaisessa ympäristössä tulevaa järjestelmää tullaan käyttämään. Jotta välttyttäisiin myöhemmin ilmenevistä ongelmista, vaatimukset pyrittiin kirjaamaan niin, että ne olisi helposti luettavissa ja ne olisi mahdollisimman tarkasti ja ymmärrettävästi kirjoitettu.

3.4 Toiminnalliset ja ei-toiminnalliset vaatimukset

Toiminnallisiin vaatimuksiin määritellään se, mitä järjestelmän odotetaan tekevän. Ne kertovat sen, miten järjestelmä toimii ulkoapäin tarkasteltuna, miten se kommunikoi ympäristön kanssa ja miten eri sidosryhmät ovat yhteydessä järjestelmään sekä sen, miten ne työskentelevät sen kanssa. (Pohjonen 2007, 28.) Toiminnalliset vaatimukset vastaavat mitä-kysymykseen. (Paakki 2011, 27.)

Ei-toiminnalliset vaatimukset määrittävät sitä, mitä rajoituksia syntyvällä ohjelmistolla on. Ne vastaavat kysymykseen miten tekee? (Paakki 2011, 26). Ei-toiminnalliset vaatimukset määrittelevät, minkälaisen reunaehtoisen vallitessa järjestelmä toteuttaa toiminnalliset vaatimuksensa. Ei-toiminnalliset vaatimukset määrittelevät siis ominaisuuksia, kuten vastausajat, kapasiteetin ja käytettävyyden. Ei-toiminnallisia vaatimuksia ovat esimerkiksi käyttöliittymän kieli, salasanan tarpeellisuus tai järjestelmän budjetti. (Hamilton 2013, viitattu 4.3.2015.)

Toiminnallisten ja ei-toiminnallisten vaatimusten olennaisin ero on siinä, että toiminnallisiin vaatimuksiin kuuluu kaikki se, mitä järjestelmän on tehtävä, kun taas ei-toiminnalliset vaatimukset osoittavat ne olosuhteet, joissa toiminnallisten vaatimusten tulee toimia. Esimerkiksi, jos suunnitellun ohjelman tulee toimia järjestelmässä, jonka resurssit ovat rajalliset, kuuluu tämä ei-toiminnallisiin vaatimuksiin. Järjestelmän resursseja ei siis voida mitenkään parantaa muuttamalla toiminnallisia vaatimuksia. Tuleva sovellus voidaan suunnitella niin, että se kiertää tämän ongelman, mutta vaatimuksiin ei voida kirjata, että koodin tulisi olla tehokkaampaa suoriutuakseen ongelmasta. (Lecky-Thompson 2005, 101 – 102.)

Elehuollolle suunniteltavan tietokannan toiminnallisia vaatimuksia kerätessä kirjattiin taulukkomuotoon asiakkaan haluama vaatimus tai ominaisuus ja siihen liittyvä vaatimustaso. Vaatimusta-

so 1 tarkoittaa, että vaatimuksella on korkea prioriteetti. Vaatimustaso 2 tarkoittaa taas sitä, että se on ominaisuus, joka olisi hyvä löytyä toteutettavasta tietokannasta. Taulukossa 1. näkyy esimerkkejä tiedon hakuun liittyvistä toiminnallisista vaatimuksista. Järjestelmän ei-toiminnallisia vaatimuksia on esimerkiksi se, että tietokannan tulee toimia Windows 7 -ympäristössä.

Taulukko 1. Tiedon hakuun liittyvät toiminnalliset vaatimukset.

Vaimus tai ominaisuus	vaatimustaso
Tietokannasta tulee voida hakea tieto, mikä ohjainyksikkö kuuluu tietylle ajoneuvolle riippuen auton merkistä, mallista ja/tai vuosimallista.	1
Tietoja tulee voida hakea sanahauulla.	2
Järjestelmästä tulee saada tulostettua listaus, jossa näkyy kaikki automerkit ja mallit, jolle tietty ohjainlaite käy.	1
Tietokannasta tulee voida hakea ohjauslaitteen korjausohje ja vikakuvaus tietyn automerkin, mallin ja/tai vuosimallin perusteella	1
Tietokannasta tulee voida tulostaa listaus, kuinka paljon tiettyä ohjainlaitetyyppeä on varastossa	1

3.5 Vaimusmäärittelyn dokumentointi

Vaimusmäärittelyn tuloksena on yleensä dokumentti, jota kutsutaan toiminnalliseksi määrittelyksi. Se sisältää yleensä niin asiakas- kuin ohjelmistovaatimuksetkin. Dokumentoinnin määrä riippuu tulevan järjestelmän luonteesta. Ison ohjelmiston suunnittelu vaatii erilaista dokumentointia kuin pienemmän ja vaatimattomamman ohjelmiston suunnittelu. Tyypillisimpiä asioita, joita vaatimusmäärittelyyn dokumentoidaan ovat esimerkiksi luontipäivämäärä, tekijä, asiakas, vaatimuksen tyyppi ja kuvaus, suhde muihin vaatimuksiin, tarpeellisuus, pysyvyys, testattavuus ja aika-arvio. (Haikala ym. 2011. 64–68.)

Vaimusmäärittelydokumentin runko muodostuu johdannosta, toiminnoista, tiedoista, liittymistä ja muista ominaisuuksista. Johdannossa kerrotaan kenen käyttöön järjestelmä tehdään ja mitkä

ovat sen tavoitteet. Johdanto-osiossa määritellään myös dokumentissa esiintyvät vieraat termit. Toiminnot kuvataan yleisellä tasolla ja ne numeroidaan tärkeysjärjestykseen esim. mitkä vaatimukset ovat pakollisia, mitkä tärkeitä ja mitkä toivottavia. Tiedot kuvataan yleisellä tasolla ja niitä tarkennetaan käsite- ja tarveanalyysillä. Dokumentissa kuvataan mahdolliset järjestelmän vaativat liittymät muihin järjestelmiin sekä järjestelmän käyttäjät ja käyttöympäristö. Lopuksi kuvataan vielä järjestelmän muut ominaisuudet, kuten suorituskyky, käytettävyys, toipuminen virhetilanteista, turvallisuus, ylläpidettävyys, siirrettävyys jne. (Hovi, Huotari & Lahdenmäki. 2005. 29.)

Huolimatta siitä, että laadukkaan dokumentoinnin tuottaminen on yksi tärkeä osa-alue tietojärjestelmän kehittämisessä, se jätetään usein liian vähälle huomiolle. Tähän on syynä usein kiire. Sen takia määrittelyyn ja suunnitteluun liittyviä asioita ei dokumentoida kunnolla, vaan ne esitellään vain pintapuolisesti. Kiireen takia myös dokumentoinnin ylläpito voi jäädä liian vähälle huomiolle. Toisaalta vaarana on turhan laaja dokumentointi, jolloin kehityshanke voi jumiutua niin, ettei se kykene etenemään elinkaaren seuraavaan vaiheeseen. Dokumentaation määrä ja laatu riippuvat kehityshankkeen laajuudesta ja hankaluudesta. Pienen projektin tiedonkulku ja dokumentointi voi tarvita vain vähän dokumentaatiota. Kuitenkin tietty perusdokumentaatio tulisi jokaisella rakennettavalla tietojärjestelmällä olla. (Haikala ym. 2011. 79.)

Vaatimusten dokumentointi riippuu myös lukijakunnasta. Mikäli dokumentin lukijakunta on teknisen osaamisen omaavia ihmisiä, voidaan vaatimusmäärittelyn dokumentoinnissa käyttää ammattikieltä. Jos taas lukijalla ei ole teknillistä osaamista, on hyvä laatia helppolukuinen asiakirja. Dokumentoinnissa on suotavaa käyttää sekä normaalia tekstinkäsittelyohjelmaa, että CASE-työkaluja. CASE-työkalujen käyttö parantaa kehitysprosessia, koska niiden käyttö helpottaa tulevan järjestelmän mallintamista. (Foster 2014, viitattu 4.3.2015.)

Elehuollon henkilöstölle tehtyjen haastatteluiden perusteella luotiin vaatimusmäärittelydokumentti, josta kävi ilmi asiakkaan vaatimukset yleisellä tasolla. Vaatimuksia tarkennettiin projektin edetessä. Dokumentista pyrittiin tekemään selkeä ja kieli pyrittiin pitämään helposti ymmärrettävänä. Vaatimusmäärittelydokumentissa kerrottiin lyhyesti tietokannan tarkoitus, kohdejärjestelmä ja yleisimmät käsitteet. Lisäksi dokumentissa käydään läpi tulevan tietokannan tietosisältö (taulukko 2.) sekä toiminnalliset ja ei-toiminnalliset vaatimukset. Dokumentoinnissa pyrittiin siihen, että sieltä löytyy kaikki tarpeellinen tieto, mutta mitään turhaa ei dokumentoida.

Taulukko 2. Esimerkkejä tulevan tietokannan tietosisällöstä.

NIMI	tietotyyppi	merkkien määrä
Merkkinumero	int	4
Automerkki	char	20

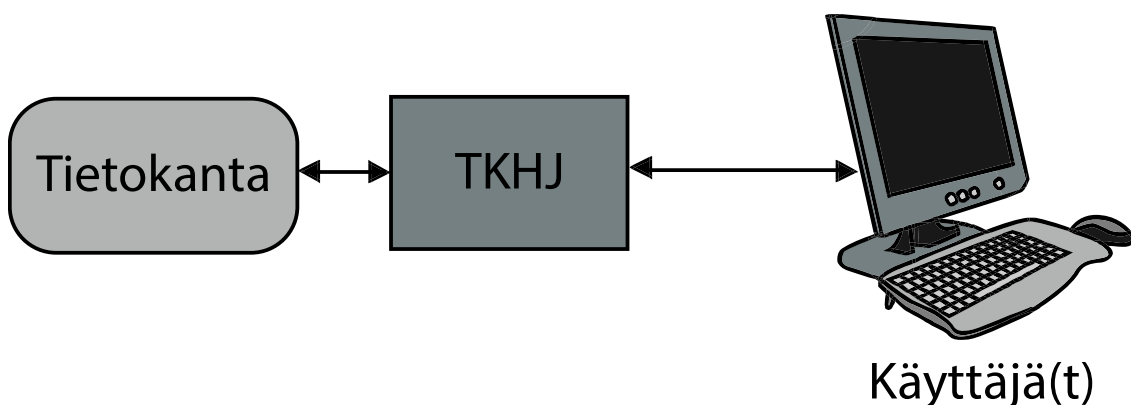
Automerkki-taulu

NIMI	tietotyyppi	merkkien määrä
Mallinumero	int	4
Automalli	char	30
Merkkinumero	int	4

Automalli-taulu

4 TIETOKANTA JA TIETOKANNAN HALLINTAJÄRJESTELMÄ

Tietokanta (Database, DB) on loogisesti yhteenkuuluvien, tallennettujen tietojen joukko, jota voidaan käsitellä jollakin tietokantakielellä, kuten SQL. Tietokantaan tallennettuja tietoja hallinnoi siihen tarkoitettu erityinen ohjelmisto, jota kutsutaan tietokannan hallintajärjestelmäksi TKHJ (Database Management System, DBMS) (kuvio 3). (Hovi, Huotari & Lahdenmäki 2005, 4.)



Kuvio 3 Tietokannan hallintajärjestelmä

4.1 Tietokannan hallintajärjestelmä

Tietokannan hallintajärjestelmissä käsitellään kahdenlaisia tietokantoja: käyttötietokantoja ja analyttisiä tietokantoja. Käyttötietokantoja käytetään silloin, kun tietoja pitää kerätä, säilyttää tai muokata. Tällaisiin tietokantoihin tallennetaan dynaamista tietoa. Tällainen tieto muuttuu jatkuvasti ja sieltä tulee saada ajantasaista tietoa. Esimerkiksi tilaustenhoitotietokannat ja potilaanseurantatietokannat ovat käyttötietokantoja. Analyttisissä tietokannoissa tiedot ovat staattisia. Se tarkoittaa sitä, että tietoja harvoin muutetaan ja että tietokanta koskee tiettyä ajankohtaa. Tällaista tietokantaa käytetään historiallisten ja ajasta riippuvien tietojen seuraamiseen ja tallentamiseen. Esimerkiksi kemiallisten testien tietokannat, geologisten näytteiden tietokannat ja kyselytutkimustietokannat ovat analyttisiä tietokantoja. (Hernandez 2000, 3,4.)

Tietokannan hallintajärjestelmät voidaan jakaa kahteen eri ryhmään: desktop tietokantoihin ja palvelintietokantoihin. Yleisesti ajateltuna desktop tietokannat on tarkoitettu yhden käyttäjän sovelluksiin. Desktop tietokantoja on mm. Microsoft Access ja FileMarker Pro. Desktop tietokannat tarjoavat edullisia ja yksinkertaisia ratkaisuja vaatimattomampien tietomäärien tallentamiseen ja

käsittelyyn. Desktop tietokantojen hyötyjä on edullisuus ja käyttäjäystävällisyys. Niiden käyttämiseen ei tarvitse hallita perusteellisesti SQL-kieltä. Lisäksi niissä on yleensä helppokäyttöinen käyttöliittymä. (Chapple, viitattu 1.4.2015.)

Palvelintietokannat on sen sijaan suunniteltu monen käyttäjän sovelluksiin, joissa vaaditaan kovaa suorituskykyä. Esimerkkejä palvelintietokannoista on Microsoft SQL Server, Oracle ja IBM DB2. Palvelintietokantaa voi käyttää ja päivittää usea käyttäjä yhtä aikaa. Näiden tietokannan hallintajärjestelmien hyödyt riippuvat palvelun toimittajasta, mutta yhteistä näille on se, että ne ovat joustavia, tehokkaita, suorituskykyisiä ja skaalautuvia. (Chapple, viitattu 1.4.2015.)

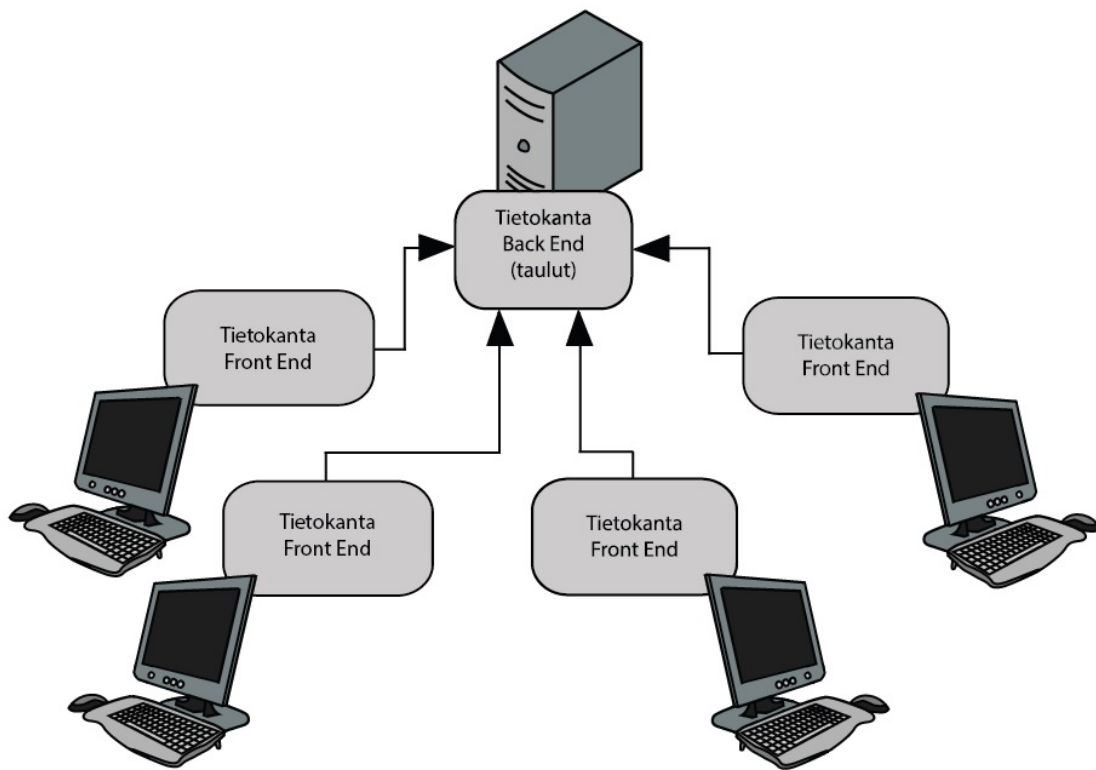
Lisäksi on olemassa avoimen lähdekoodin tietokannan hallintajärjestelmiä, joita voi ladata ilmaiseksi internetistä. Näistä esimerkkeinä PostgreSQL, MariaDB, SQLite ja MySQL. Lähdekoodi on näissä vapaasti käytettävissä ja muokattavissa. Yksi suosituimmista avoimen lähdekoodin tietokannan hallintajärjestelmistä on MySQL.

Tietokannan hallintajärjestelmän valinnassa kannattaa kiinnittää huomiota ainakin seuraaviin asioihin: Kuka tietokantaa käyttää ja mitä tehtäviä he suorittavat, kuinka usein tietoja muutetaan ja kuka muutokset tekee, mikä laitteisto on käytettävissä, kuka vastaa tietojen ylläpidosta ja tuleeko tietokannan toimia internetissä?

Microsoft Access on tietokantaohjelma, joka kuuluu Microsoftin Office-pakettiin. Yksi suurimmista eroista Accessin ja muiden tietokantatuotteiden välillä on se, että Access on tiedostopohjainen tietokanta. Sitä vastoin esimerkiksi SQL Server on palvelinohjainen tietokanta, joka varastoi tietoa suuritehoisille palvelimille. Näihin tietoihin pääsee käsiksi omalta tietokoneelta joko lähiverkon tai internetin kautta. Lisäksi palvelinohjaiset tietokannat ovat paljon monimutkaisempia perustaa ja ylläpitää, vaikkakin ne tarjoavat parempaa suorituskykyä, jota tarvitaan silloin, kun niillä on tuhansia käyttäjiä samaan aikaan. (MacDonald 2013, viitattu 25.3.2015.)

Toinen ero Accessin ja palvelinohjaisen tietokantatuotteen välillä on, että Accessilla on ratkaisu sekä tiedon säilyttämiseen, että vuorovaikutteisuuden tiedon kanssa. Sen sijaan palvelinohjainen tietokantamoottori, kuten aiemmin mainittu SQL Server, keskittyy ainoastaan tiedon tallentamiseen ja lähettää vain pyydettyä tietoa muihin koneisiin. Tämä johtaa siihen, että jos päättää käyttää palvelinohjaista tietokantaa, täytyy siihen koodata sovellus, ennen kuin tietokantaa voi käyttää tehokkaasti. (MacDonald 2013, viitattu 25.3.2015.)

Jaettu tietokanta tarkoittaa sitä, että tietokanta on jaettu kahteen erilliseen tiedostoon. Taustatietokanta (Back-end-database) sisältää vain raakatietoa, eli tietokannan taulut. Edustatietokanta (Front-end-database) taas sisältää kaiken sen mitä tietokannan kanssa työskentelemiseen tarvitaan. Siellä on kaikki muut tietokannan objektit, kuten raportit, kyselyt, lomakkeet ja makrot. Jaetussa Access-tietokannassa taulut ovat yhdellä serverillä, johon kullakin käyttäjällä on yhteys (kuvio 4). (MacDonald 2013, viitattu 25.3.2015.)



Kuvio 4. Jaettu Access-tietokanta. (MacDonald 2013, viitattu 25.3.2015.)

Suunniteltava tietokanta on suhteellisen pieni ja sisältää alle kymmenen taulua. Tietokantaan tallennettava tieto muuttuu harvoin sen jälkeen, kun se on tallennettu tietokantaan. Tietokantaan lisätään uutta tietoa esimerkiksi silloin, kun korjattavaksi tulee uuden automallin ohjainyksikkö tai kun tietokantaan tallennetaan uusi korjausohje. Tietokannasta tulee harvoin tarvetta poistaa sinne tallennettua tietoa. Tietokantaan tallennettava tieto on siis stabiilia, eli kyseessä on analyttinen tietokanta.

Koska tuleva tietokanta on pieni ja se sisältää stabiilia tietoa, on johdonmukaista valita jokin kevyempi ratkaisu tietokannan hallintajärjestelmäksi. Tulevaa tietokantaa käyttää samanaikaisesti

maksimissaan neljä työntekijää. Nämä harvoin muuttavat tietoa samaan aikaan. Siksi tietokanta olisi järkevä toteuttaa Accessilla. Accessin valintaa puoltaa myös se, että se sopii hyvin pienten tietomäärien tallentamiseen ja muokkaamiseen. Lisäksi sen avulla käyttöliittymän, kyselyiden ja syöttölomakkeiden luominen on helppoa.

4.2 Tietokannan rakenne

Tietokannan rakenne koostuu tauluista ja niiden välisistä yhteyksistä, kentistä, tietueista, näkymistä ja avaimista. Taulu (table) on relaatiokannan pään rakenne. Se muodostuu kentistä ja tietueista, joiden järjestyksellä ei ole väliä. Taulu edustaa kohdetta, esimerkiksi henkilöä, paikkaa tai esinettä. Kenttä (attribuutti) on relaatiotietokannan pienin rakenne. Tietokantaan tallennetaan tietoa kentän avulla. Kenttä edustaa aiheen tiettyä ominaisuutta siinä taulussa, jossa kenttä sijaitsee. Tietue (monikko) on taulun sisäinen rakenne. Se edustaa taulun aiheen tiettyä esiintymää. Näkymä taas on näennäistaulu, joka koostuu yhden tai useamman taulun kentistä. Näkymää pidetään siksi näennäisenä, koska se ei säilytä itse tietoa, vaan se saa tietonsa tauluista, joihin se perustuu. Avaimet ovat erikoiskenttiä, jotka palvelevat taulussa tiettyä tarkoitusta. Avaimen tyyppi ratkaisee sen käytön taulussa. Esimerkkejä avaimista on pääavain ja viiteavain. Pääavaimen avulla taulun sisältämä tietue tunnistetaan yksiselitteisesti. Viiteavaimen avulla kahden taulun välille voidaan luoda yhteys. (Hernandez 2000, 39–45.)

Yhteydellä tarkoitetaan kahden taulun välille luotua liitosta. Yhteys syntyy, kun kaksi taulua yhdistetään pääavaimen ja viiteavaimen avulla. Yhteydet ovat datan eheyden kannalta erittäin tärkeitä, koska ne auttavat vähentämään ylimääräistä ja kaksinkertaista dataa. Yhteyksien avulla voidaan myös määritellä näkymiä. (Hernandez 2000, 46.)

Kuviossa 5 on esitetty tietokannan taulun rakenne. Kuviossa on siis kaksi taulua: yritys ja henkilö. Yritystaulun perusavain on Ytunnus ja henkilö-taulun perusavain on Hnro. Ytunnus toimii viiteavaimena henkilö-taulusta yritys-tauluun. Kentät ovat pystysuoria sarakkeita. Tietueet taas ovat vaakasuoria rivejä. Esimerkiksi kuviossa 5 Yritys Oy edustaa yritystaulun ainutlaatuista esiintymää. Yritys Oy:n tietue on siis kenttien kokoelma, jota kohdellaan yhtenä yksikkönä. Näiden kenttien arvot edustavat Yritys Oy:öön liittyviä tietoja.

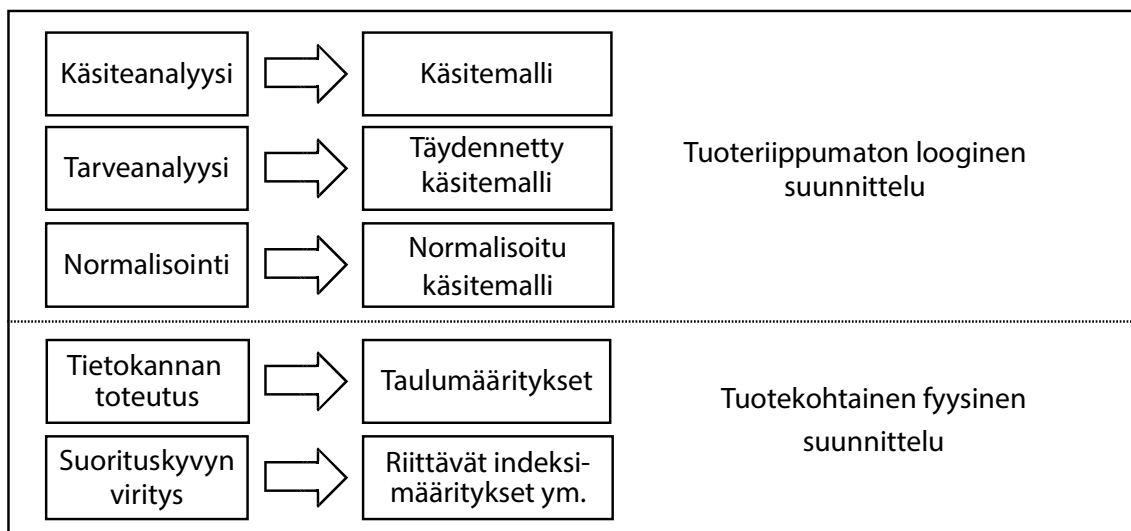


Kuvio 5. Tietokannan taulun rakenne.

5 TIETOKANNAN SUUNNITTELU JA TIETOMALLIT

Tietomalli (data model) on kuvausjärjestelmä ja ryhmä sääntöjä, joiden avulla voidaan kuvata tietorakenteita ja niiden välisiä yhteyksiä. Relaatiotietokannat perustuvat E. F. Coddin julkaisemaan relaatiomalliin (the relational model). Relaatiomalli ei ota kantaa relaatiokannan fyysiseen toteutukseen, vaan sen toteuttaa tietokantatuotteiden toimittajat. Tietomalleilla (käsitemalli, ER-malli) havainnollistetaan käsiteltävän tiedon sisältöä, rakennetta ja suhteita. Tietomallit eivät yleensä ota kantaa siihen, minkälaisia rakenteita tietojen käsittelyyn tarvitaan. Käsiteltävät tiedot yritetään jaotella loogisiin käsitteisiin ja kuvaamaan näiden välisiä suhteita. (Pohjonen 2007, 63,120.)

Tietokannan suunnittelu rakentuu tietyistä vaiheista, jotka on tehtävä kaikissa tietokantojen suunnitteluhankkeissa. Vaiheiden muodostamaa kokonaisuutta kutsutaan suunnitteluputkeksi (kuvio 6). Joskus jokin suunnitteluputken vaiheista voi olla hyvin nopeita ja lähes ohitettavia. (Hovi ym. 2005, 24.)



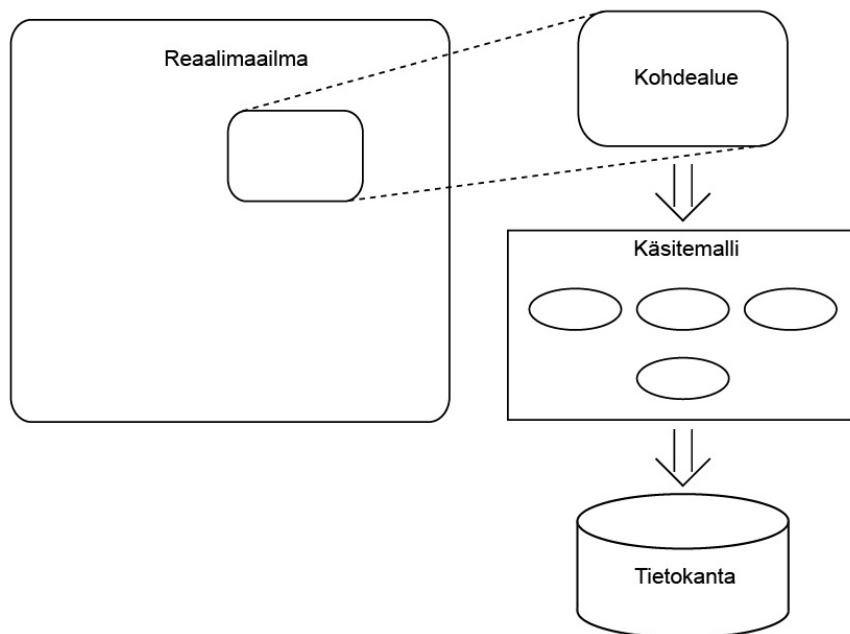
Kuvio 6. Tietokantojen suunnitteluputki (Hovi ym.2005, 24).

Suunnitteluputken ensimmäinen vaihe on käsiteanalyysin (ER-modelling, conceptual modelling, conceptual design, data modelling) teko. Tässä vaiheessa tietokanta suunnitellaan loogisella tasolla ja muodostetaan käsitemalli. Tarveanalyysissä käsitemallia tarkennetaan tietotarpeiden perusteella ja lisätään kaikki puuttuvat tiedot. Normalisointivaiheessa tarkistetaan, ettei mallissa

ole turhaa toisteisuutta. Normalisoinnin jälkeen on vuorossa tuotekohtainen suunnittelu, jolloin tulevan tietokannanhallintajärjestelmän tulisi olla tiedossa. Tietokannan toteutusvaiheessa käsitellään käsitteet muutetaan tietokannan tauluiksi. Viimeinen vaihe on suorituskyvyn viritys, mikä tarkoittaa lähinnä indeksointia. (Hovi ym. 2005, 24,25.)

5.1 Käsiteanalyysi

Tietokantojen suunnitteluprosessin ensimmäinen vaihe on siis käsiteanalyysin teko. Siinä selvitetään, mitä tietoja tulevaan tietokantaan talletetaan. Käsiteanalyysissä kuvataan kohdealuetta jota on tarkoitus kuvata tietokannassa (kuvio 7). Lopputuloksena syntyy käsitelmä, joka kuvaa toisaalta kohdealuetta ja toisaalta määrittelee tietokannan fyysiselle rakenteelle pohjan. Tässä vaiheessa tarkoituksena on saada kuvattua riittävän karkealla ja yleisellä tasolla tulevaan tietokantaan haluttavat asiat. Käsiteanalyysiä toteutetaan aluksi karkealla tasolla. (Hovi ym. 2005, 32,33.)



Kuvio 7. Käsitelmä kuvaa tietokannan toteutusta (Hovi ym. 2005, 32.)

Elehuollon tietokannan suunnittelu eteni suunnitteluputken mukaisesti. Käsiteanalyysin teko aloitettiin jo määrittelytyön aikana. Käsiteanalyysiä pystyi tekemään samaan aikaan kohdealueeseen

tutustumisen kanssa. Määrittelytyössä otettiin siis selvää, mitä tietoja tulevaan tietokantaan halutaan tarkentaa. Käsitemallin näiden tietojen perusteella pyrittiin havainnollistamaan tulevaan tietokantaan tallennettavia tietoja. Käsitemallin lopputuloksena syntyvää käsitemallia käytetään myöhemmin lopullisen tietokannan fyysisenä pohjana.

5.2 Tietomallit

Tietomalleista tyypillisimpiä ovat käsitemalli, oliomalli ja relaatiomalli. Relaatiomalli jakaantuu vielä Coddin esittämään abstraktiin relaatiomalliin ja fyysiseen taulumalliin. Relaatiomalli perustuu matematiikkaan, joukko-oppiin ja predikaattilogiikkaan. Relaatiomallin kolme osaa ovat rakenne, käsittely ja eheysäännöt. (Hovi ym. 2005, 6 – 8.) Tietokannan rakenteesta kerrottiin aiemmin luvussa 4.1.

Tietokannan käsittelyssä on taustalla joukko-oppi. Tietoja siis käsitellään joukko-opillisesti. Taulun riveille voidaan kohdistaa joukko-operaatioita, kuten hae kaikkien espoolaisten asiakkaiden nimi ja osoite. Joukko-operaatioilla voidaan hakea joko yhtä tai useampaa taulua kerrallaan. Joukko-opillisuutta toteutetaan SQL-kielellä (Structured Query Language). Suurin osa relaatiotietokannoista ymmärtää vain SQL-kieltä. SQL-kielellä voidaan päivittää ja lukea tietokantaa sekä lisätä ja poistaa rivejä. Tietokannan tulee olla hyvin suunniteltu, että SQL:n avulla sieltä saadaan tietoa ulos. (Hovi ym. 2005, 10.)

Tietokanta on eheä silloin, kun sen tiedot ovat oikeita ja ristiriidattomia ja ne vastaavat reaali maailmaa. Eheys on vaarantunut esimerkiksi silloin, kun tietokannasta löytyy sama asiakas kahteen kertaan tai yhdellä asiakkaalla on kaksi eri osoitetta, eikä voida tietää kumpi niistä on oikea. Avainuus tarkoittaa sitä, että perusavaimen arvo ei saa koskaan olla tyhjä, eli NULL-arvo. Viiteuus taas tarkoittaa sitä, että esimerkiksi isätaulusta ei saa poistaa tietoja jos lapsitaulussa on kyseiseen isään liittyviä rivejä. (Hovi ym. 2005, 11, 12.)

5.3 Käsitteet

Käsite (entity) kuvaa jotakin konkreettista tai abstraktia asiaa, esinettä, henkilöä, tapahtumaa tai vaikkapa paikkaa, josta halutaan tallentaa tietoa myöhempää käyttöä varten. Käsitteen määrittely voi joskus olla hankalaa, joten yksi käsitemallin vaikeimpia asioita on löytää oikeat käsitteet.

Käsite kuvataan käsitekaaviossa esimerkiksi ovaalilla tai laatikolla, jonka sisälle merkitään käsitteen nimi. Yhden käsitteen tiedot voi esittää myös käsitekaavalla ilman graafista notaatiota. Käsitteen perään sulkuihin kirjoitetaan tiedot pilkulla erotettuna ja perusavain alleviivattuna. Esimerkiksi Osasto(osastoID, osastonNimi). Käsittemallin avulla käsitteet ja niiden tiedot saa esitettyä tiiviissä muodossa, mutta niistä on vaikea hahmottaa käsitteiden välisiä yhteyksiä. (Hovi ym. 2005, 35, 74.)

Elehuollolle suunniteltavan tietokannan käsiteanalyysissä pyrittiin määrittelemään ja kuvaamaan tulevan tietokannan käsitteet. Ydinkäsitteiksi löytyi tässä vaiheessa automerkki, automalli, ohjainlaitekattegoria, ohjainlaite, vuosimalli ja varaosa. Näitä käsitteitä tarkennettiin suunnittelun myöhemmissä vaiheissa. Tarkennetuista käsitteistä syntyi myöhemmin tulevan tietokannan taulut.

5.4 Tiedot

Tieto (attribute) kuvaa käsitettä. Käsitteeseen kuuluu ryhmä tietoja ja kukin tieto on käsitteen ominaisuus. Tietoja voivat olla esimerkiksi asiakkaan nimi, katuosoite, puhelinnumero ja saldo. Yleensä tieto on teksti-, nimi- tai päivämäärätyyppinen. Käsitteisiin liittyy myös ilmentymiä (instance, occurrence). Esiintymät ovat käsitteen mukaisia tietoja ja pitävät sisällään tietojen arvot. (Hovi ym. 2005. 35–36.) Esimerkiksi kuvion 5 Henkilö-käsitteen mukaisia esiintymiä ovat esimerkiksi Taisto, Aune ja Pekka.

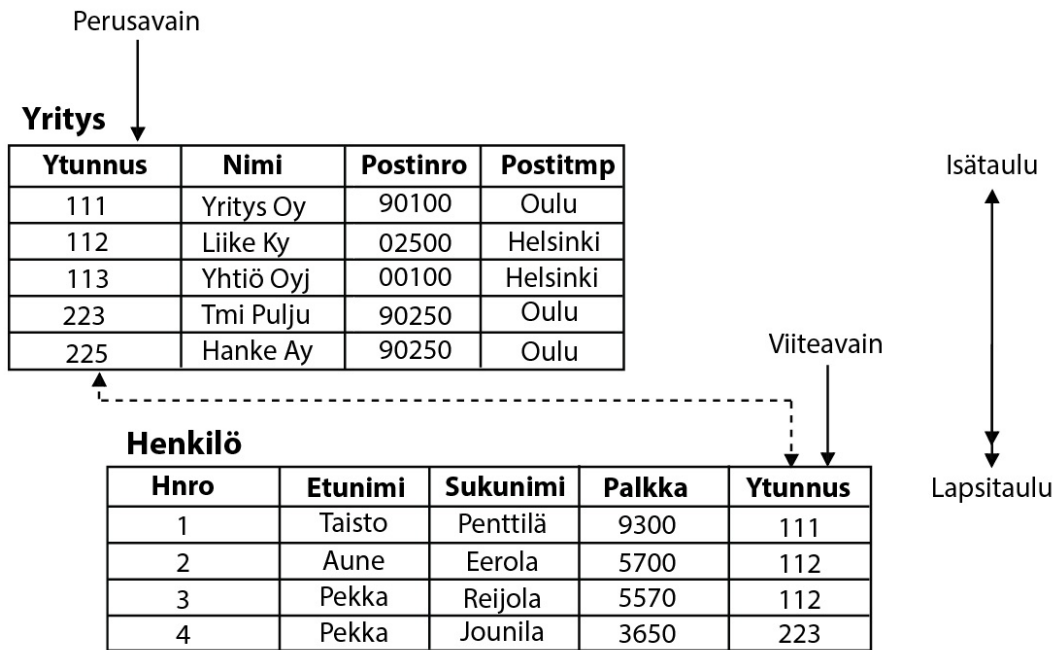
Elehuollolle suunniteltavan tietokannan auto-käsitteen attribuutteja ovat merkkinumero ja automerkki. Automerkki- ja automalli -käsitteiden ilmentymiä ovat luonnollisesti automerkit ja automallit, joiden osia Elehuolto korjaa. Koska eri automalleihin kuuluvat ohjainlaitteet on yksilöity osanumeron mukaan, varaosalaitteiden yksilöimiseen ei riitä pelkkä ohjainlaitteen nimi. Tästä syystä varaosakäsitteen tietoja ovat osanumero ja määrä. Osanumeron ilmentymiä ovat ohjainlaitteen osanumerot. Koska sama korjausohje käy kaikille saman merkkisille ohjainlaitteille, ohjainlaite-käsitteen ilmentymiä ovat eri ohjainlaitteiden merkit ja niiden korjausohjeet. Ohjainlaitekattegorian tietoja ovat kategorianumero ja ohjainlaitekattegoria. Käsitteen ilmentymiä ovat esimerkiksi moottorinohjainyksikkö tai ABS-jarrut.

5.5 Avaimet

Taulurakenteen kannalta avaimet ovat ratkaiseva asia, koska ne ensinnäkin varmistavat, että taulun kaikki tietueet voidaan tunnistaa. Toiseksi ne auttavat pakottamaan ja luomaan eheyden eri lajeja. Tämä tarkoittaa esimerkiksi sitä, että avaimen avulla voidaan varmistaa, että taululla on selkeästi erotettavia tietueita ja että kahden taulun yhdistämiseen käytettyjen kenttien arvot täsmäävät aina keskenään. Kolmas syy avaimien tärkeydelle on se, että niiden avulla luodaan yhteyksiä eri taulujen välillä. Avainten avulla muodostetaan yhteys kahden taulun välillä. (Hernandez 2000, 212.)

Perusavain on taulun tärkein avain. Se pakottaa taulutason eheyden voimaan ja auttaa yhteyden muodostamisessa muihin tauluihin. Lisäksi sen avulla taulun tietty tietue voidaan tunnistaa ja siihen kyetään viittaamaan tarkasti. (Hernandez 2000, 220.) Perusavaimella on viisi ominaisuutta. Ensinnäkin sen tulee olla yksilöllinen eli uniikki. Tämä tarkoittaa, että perusavaimessa ei saa olla kahta samanlaista arvoa. Toiseksi se voi olla joko luonnollinen tai surrogaatti. Luonnollinen perusavain on esimerkiksi henkilötunnus, surrogaatti taas on tekninen numero, joka ei sisällä muutuvaa tietoa. Yleensä surrogaatti on juokseva numero. Perusavaimen kolmas ominaisuus on, että sitä ei tarvitse päivittää. Päivittämisen ongelmana olisi, että jos perusavaimen arvo muuttuu, jouduttaisiin muuttamaan myös viiteavainta. Neljänneksi, perusavain voi olla moniosainen, toisin sanoen se voi koostua monesta eri sarakkeesta ja viidenneksi, perusavaimen arvo ei saa koskaan olla tyhjä. (Hovi ym. 2005, 62 – 66.)

Viiteavaimia käytetään yksi-moneen -yhteyksissä viittaamaan lapsitaulusta isätauluun (kuvio 8). Viiteavain viittaa isätaulun perusavaimen. Viittavaa taulua kutsutaan lapsitauluksi, kun taas viittauksen kohteena oleva taulu on isätaulu. Viiteavaimia tarvitaan siinä vaiheessa, kun tauluja tulee yhdistää, eli kun halutaan tehdä liitoksia. (Hovi ym. 2005, 9.)



Kuvio 8. Perus- ja viiteavain.

Elehuollolle suunniteltavan tietokannan kaikille käsitteille, lukuun ottamatta varaosa -käsitettä, annettiin perusavaimeksi juokseva numero eli surrogaatti. Esimerkiksi automerkki-käsitteen perusavaimena on merkkinumero, joka muodostuu automaattisesti juoksevana numerona, kun uusi automerkki lisätään tietokantaan. Varaosa -käsitteelle annettiin luonnollinen perusavain eli perusavaimena käytetään ohjainlaitteen osanumeroa.

5.6 Käsitteiden väliset yhteydet


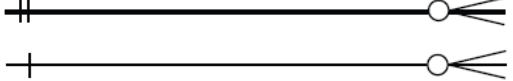




Käsitteiden välillä on suhteita eli yhteyksiä eli notaatioita. Tärkein ja yleisin suhdetyyppi on yksi-moneen-yhteys eli isä-lapsi-yhteys. Muita yhteystyyppejä on yksi-yhteen- ja moni-moneen -yhteydet. Yhteyksiä voidaan kuvailla verbillä. Esimerkiksi asiakas tekee tilauksen, auto kuuluu yritykselle tai opiskelija osallistuu kurssille. (Hovi ym. 2005, 37.)

Yhteys syntyy, kun kaksi taulua on yhdistetty pääavaimen ja viiteavaimen avulla tai kun ne on linkitetty yhteen kolmannen taulun avulla. Yhteydet ovat tärkeitä datan eheyden kannalta. Ne auttavat vähentämään ylimääräistä ja kaksinkertaista dataa. Yksi-yhteen- yhteys on olemassa taulujen välillä, jos ensimmäisen taulun yksittäinen tietue voi liittyä vain yhteen toisen taulun tietueeseen ja toisen taulun yksittäinen tietue voi liittyä vain yhteen ensimmäisen taulun tietueeseen. Yksi-moneen-yhteydestä on kyse silloin, jos ensimmäisen taulun yksittäinen tietue voi liittyä yh-

teen tai useampaan toisen taulun tietueeseen, mutta toisen taulun yksittäinen tietue voi liittyä vain yhteen ensimmäisen taulun tietueeseen. Moni-moneen-yhteys tarkoittaa sitä, että ensimmäisen taulun yksittäinen tietue voi liittyä yhteen tai useampaan toisen taulun tietueeseen ja päinvastoin. (Hernandez 2000, 46 – 50.)

Moni-moneen -yhteys syntyy esimerkiksi silloin, kun tiettyyn projektiin voi osallistua monta henkilöä ja yksi henkilö voi kuulua moneen projektiin. Moni-moneen -yhteydet on hyvä purkaa auki muodostamalla jokin välikäsite. Jokainen moni-moneen -yhteys voidaan purkaa kahdeksi yksi-moneen -yhteydeksi, muodostamalla alkuperäisten käsitteiden väliin uusi käsite. Uusi käsite on lapsisuhteessa molempiin alkuperäisiin käsitteisiin ja sen perusavain on lähes aina aiempien käsitteiden perusavaimien yhdistelmä. Yksi-yhteen- yhteydet ovat aika harvinaisia. Esimerkkinä yksi-yhteen- yhteydestä on henkilö- ja myyjä-käsitteet, jotka on kuvattu erikseen. Myyjä-käsite sisältää paljon tietoa, jotka liittyvät vain myyjään ja heitä on henkilöstössä vain vähän. Yleensä yksi-yhteen- ratkaisua kannattaa välttää ja sen voi ratkaista yhdistämällä taulut. (Hovi ym. 2005, 44 – 45, 67, 106).

Käsitekaavio voidaan esittää monilla eri kuvaustekniikoilla. Malleista on ajan saatossa syntynyt monia eri variaatioita. Yksi tunnetuimmista menetelmistä on James Martinin menetelmä Information Engineering. Muita kuvausmenetelmiä on IDEFIX, UML ja NIAM -menetelmät. UML:n luokakaavion lisäetuna on se, että luokkiin voi liittää metodeja eli operaatioita. (Hovi ym. 2005, 74, 75.)

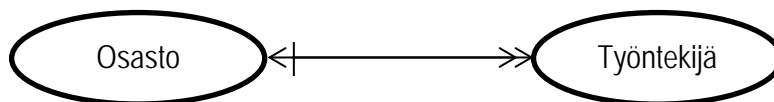
Yhteyden kuvaustapa	Selitys:
	Bruse: IDEF1X
	Martin: Information Engineering SSADM
	Soini: Kohdeanalyysi
	Chen: Entity-Relationship Diagram (ERD)
	Booch, Jacobson, Rumbaugh: UML
	Oracle Design

Kuvio 9. Notaatioiden erilaisia merkitsemistapoja. (Hovi ym. 2005.)

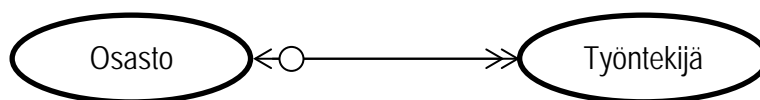
Käsitteiden välisiä yhteyksiä voidaan siis kuvata monella eri tavalla. Kuviossa 9 näkyy notaatioiden erilaisia merkitsemistapoja. Kuviossa näkyvät merkintätavat kuvaavat kaikki yksi-moneen-yhteyttä. Joissakin kuvaustekniikoissa on mahdollista ilmaista myös yhteyden lukumääräsuhte. (Hovi ym. 2005, 75.)

Yhteyksiä voidaan tarkentaa määrittelemällä onko yhteydet pakollisia vai ehdollisia. Kuvio 10 kuvaa pakollista yhteyttä. Tässä tapauksessa työntekijän on aina kuuluttava johonkin osastoon. Toisin sanoen työntekijää ei voi tallentaa tietokantaan, ellei häntä kytketä johonkin osastoon. Kyseessä on siis pakollinen yhteys. Jos työntekijän taas tulee pystyä tallentamaan tietokantaan vaikka hänelle ei ole vielä määrätty osastoa tai hän ei tule kuulumaan mihinkään osastoon, käytetään ehdollista yhteyttä (kuvio 11). Työntekijä voi tässä tapauksessa kuulua yhteen tai nollaan osastoon. (Hovi ym. 2005, 42,43.)

Hernandezin mukaan taulu voi osallistua yhteyteen kahdella eri tavalla, joko pakollisesti ja vapaaehtoisesti. Esimerkiksi, jos taulu A:n tietueiden täytyy olla olemassa ennen kuin taulu B:hen voidaan tallentaa uusia tietueita, taulu A:n osallistuminen uuteen yhteyteen on pakollista. Jos taas tietueiden olemassaolo taulu A:ssa ei ole välttämätöntä, jotta taulu B:hen voidaan syöttää tietueita, taulu A: osallistuminen yhteyteen on vapaaehtoista. Yhteyksien pakollisuus riippuu yleensä siitä, miten kunkin taulun tiedot liittyvät muihin tietoihin ja miten tietoja käytetään. (2000, 50,51.)



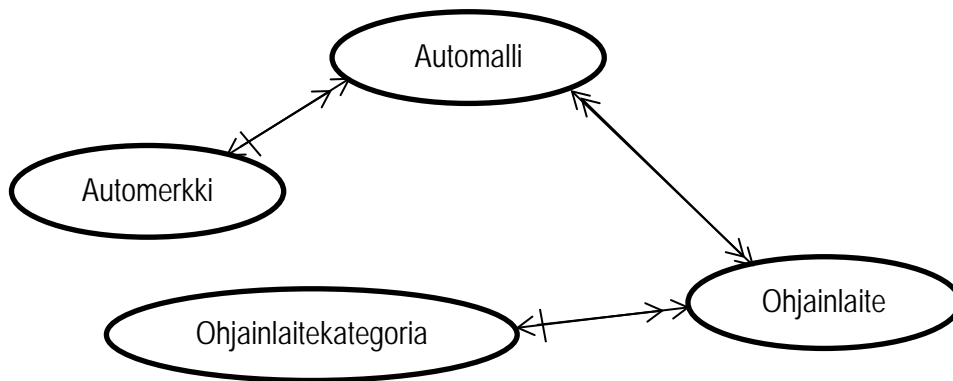
Kuvio 10. Pakollinen yhteys.



Kuvio 11. Ehdollinen yhteys.

Elehuollolle suunniteltavan tietokannan automerkki- ja automalli-käsitteiden välillä on yksi-moneen-yhteys. Yhdellä automerkillä voi olla useita automalleja, mutta yksi automalli kuuluu aina yhdelle automerkille. Samoin yhteen ohjainlaitekategoriaan voi kuulua usea ohjainlaite, mutta yksi

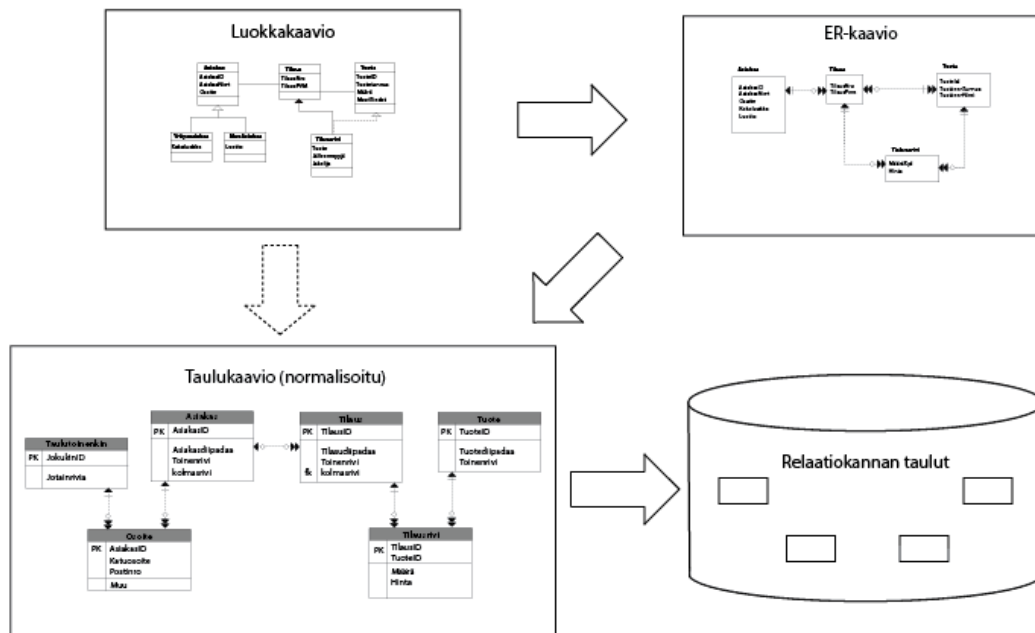
ohjainlaite voi kuulua vain yhteen ohjainlaitekategorian. Yhdellä automallilla voi olla monta erilais- ta ohjainlaitetta ja yksi ohjainlaite voi kuulua monelle eri automallille. Näiden käsitteiden välillä on moni-moneen-yhteys. Jotta tietojen lisääminen, päivittäminen ja poistaminen eivät myöhemmin olisi hankalaa, on moni-moneen-yhteys purettava. Kuviossa 12 on esimerkki muutamasta käsi- temallin käsitteestä ja niiden välisestä yhteydestä.



Kuvio 12. Käsitteiden välisiä yhteyksiä.

5.7 UML:n luokkakaavio

Oliopohjainen suunnittelu on yleistynyt viime vuosina. Suunniteltavat tietokannat ovat suurim- maksi osaksi relaatiokantoja. Käytössä olevat oliosuunnittelumenetelmät eivät muodosta ehyttä polkua relaatiokannan taulukaavioihin asti. Oliosuunnittelumenetelmiin kuuluvat luokka, - toiminto - ja tilakaaviot eivät ole riittäviä tai eivät suoraan sovi relaatiokannan pohjaksi. Luokkakaavio on yksi oliosuunnittelun lopputulos. Se sisältää usein periytymistä kuvaavia rakenteita, joita ei ole käsitemallissa. Niitä ei voi myöskään suoraan toteuttaa relaatiokannoilla. Jos luokkakaavion to- teuttaa suoraan taulurakenteeksi, johtaa se helposti tehottomaan toteutukseen. Sen sijaan luok- kakaavio kannattaa ensin muuttaa käsitemalliksi ja edetä tämän jälkeen suunnitteluputken mu- kaisesti (kuvio 13). (Hovi ym. 2005, 118.) Yksi oliomallinnuksen menetelmä on UML (Unified Mo- deling Language).



Kuvio 13. Polut luokkakaaviosta taulujen toteutukseen. (Mukailtu kirjasta Hovi ym. 2005, 118.)

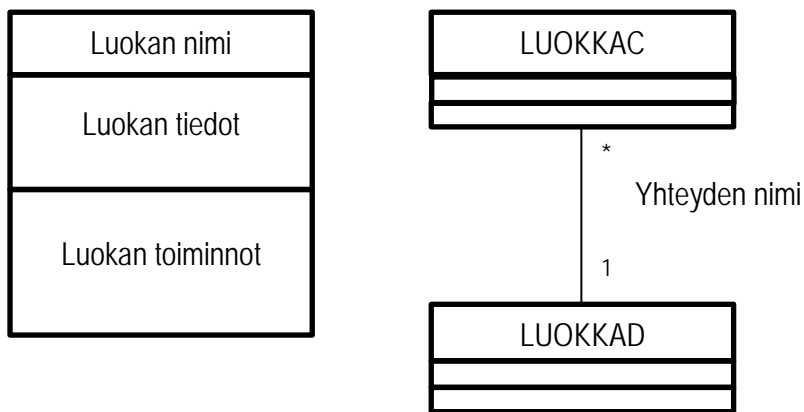
Luokkakaavio koostuu luokista, niiden tiedoista ja toiminnoista sekä luokkien välisistä yhteyksistä (kuviot 14). Yhteystyyppinä ovat yhteyssuhde, koostesuhde sekä yleistysuhde eli periminen. Yhteydet rakennetaan aina kahden luokan välille, vaikka luokka olisi yhteydessä moneen eri luokkaan. Luokkakaaviossa yhteystyyppit kuvataan seuraavalla tavalla:

- Yksi-yhteen: 1:1
- Yksi-moneen 1:M tai
- Moni-moneen M:N

Muodostettaessa yhteyssuhteita luokkien välille, kerrotaan yhteyden nimi ja kerrannaisuus. Kerrannaisuus voidaan merkitä seuraavalla tavalla:

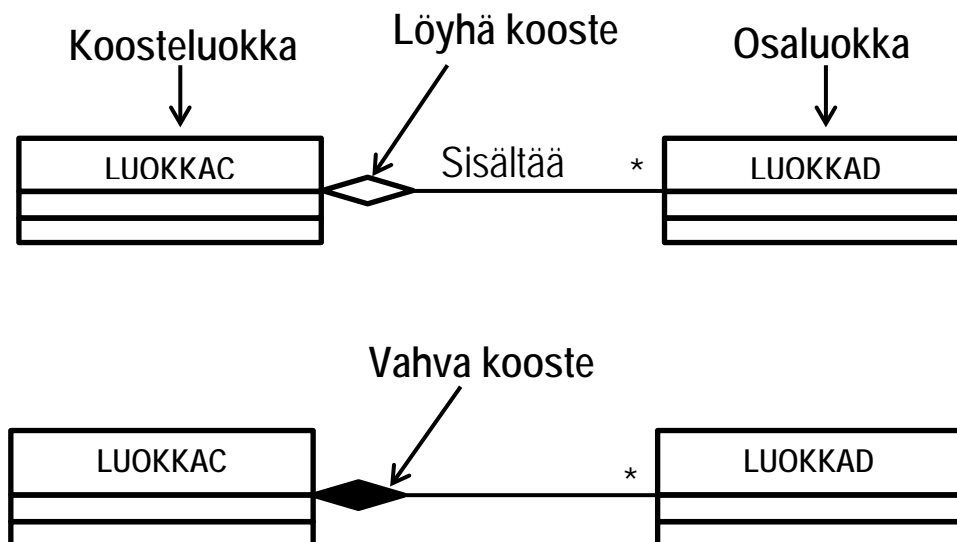
- 1 merkitsee yhtä yhteyttä luokkien välillä
- * merkitsee määrittelemättömän monta yhteyttä
- 0...1 merkitsee, että kertautumisia on yksi tai ei yhtään
- 0..* merkitsee, että kertautumisia ei ole yhtään tai niitä on määrittelemättömän monta
- kerrannaisuuksien määrää voidaan ilmaista myös tarkemmin esimerkiksi 10...100

(Nousiainen, viitattu 19.2.2015.)



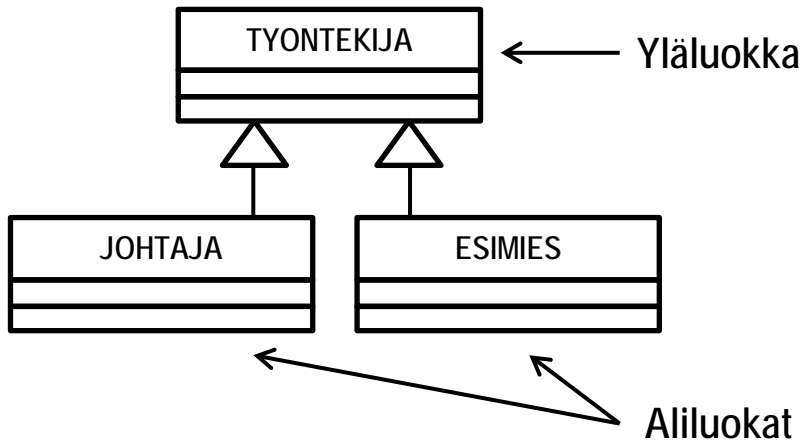
Kuvio 14. Luokkakaavio. Oikealla esimerkki yhteyssuhteesta.

Yhteyssuhteen erikoismuoto on koostesuhte. Siinä on kyse omistamisesta, jossa koosteluokka omistaa osaluokkansa. Koosteisia suhteita on löyhä kooste (aggregation) ja vahva kooste (composition). Löyhässä koosteessa osaluokan olemassaolo ei ole riippuvainen koosteluokasta. Vahvassa koosteessa taas osaluokkaa ei voi olla ilman koosteluokkaa. Vahvassa koosteessa koosteluokan hävitessä myös osaluokat häviävät. Löyhä kooste merkitään ontolla vinoneliöllä ja vahva kooste täytetyllä vinoneliöllä (kuvio 15). (Nousiainen, Viitattu 19.2.2015.)



Kuvio 15. Löyhä- ja vahva kooste.

Yleistäminen eli periminen on yläluokan ja aliluokan välinen suhde. Aliluokka on yhteensopiva yläluokan kanssa ja sisältää siihen liittyvää lisätietoa sekä perii yläluokan ominaisuuksia. Yleistäminen kuvataan ontton kolmion avulla (kuvio 16). (Nousiainen, viitattu 19.2.2015).



Kuvio 16. Yleistäminen.

Luokkakaavio kannattaa muuttaa käsitemalliksi. Peruseriaate on, että kustakin luokasta tulee käsitemallin käsite. Luokan yksinkertaisista tiedoista tulee käsitteen ominaisuuksia. Luokkien olioita yksilöi ID (OID object identifier), josta voidaan luoda käsitteen yksilöivä perusavain. Monimoneen-yhteydet tulee myös purkaa. Jos suhde on esimerkiksi nimetty "osallistuu", voidaan tätä nimeä käyttää uuden käsitteen nimenä, kun muutetaan se substantiiviksi, esimerkiksi osallistuminen. Kooste, joka on yksi-moneen-suhteen erikoistapaus, voidaan muuttaa suoraan ER-kaavion yksi-moneen -yhteydeksi. Luokkahierarkian muuttamiseen käsitemalliksi on kolme vaihtoehtoa. Ensimmäinen ja yleisin vaihtoehto on tehdä jokaisesta luokasta oma käsite. Tätä vaihtoehtoa kannattaa käyttää vain silloin, kun luokilla on paljon toisistaan poikkeavia tietoja. Toisena vaihtoehtona on yläluokan ominaisuuksien kopioiminen aliluokkiin ja yläluokkien poistaminen. Tätä vaihtoehtoa kannattaa käyttää silloin, kun yläluokilla ei ole paljon ominaisuuksia, mutta alaluokilla on. Kolmas vaihtoehto on yhdistää aliluokat yläluokkaan, jolloin syntyy yksi käsite. (Hovi ym. 2005, 118–131.)

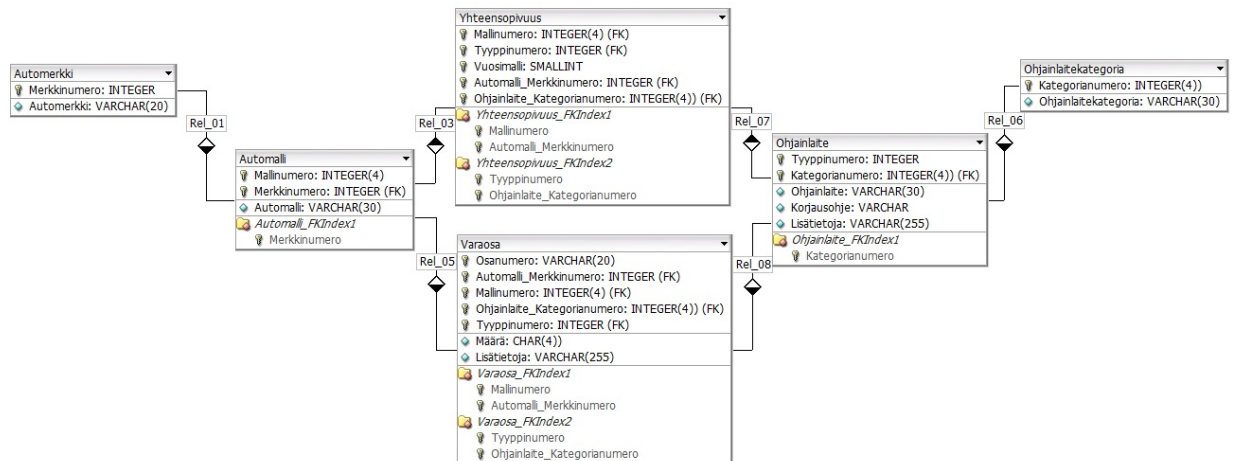
5.8 Tarveanalyysi

Tarveanalyysiä tehdessä käsitemalli voi olla vielä aika puutteellinen. Varsinkin käsitteiden tietoluettelot voivat olla vielä aika keskeneräisiä. Tarveanalyysissä käsitemalli täydennetään tarkalle

tasolle. Tarveanalyysissä on tarkoitus ensinnäkin tarkistaa ja tehdä rakenteellinen testaus käsittemallin tiedossa olevilla tietotarpeilla. Toinen tavoite on täydentää käsittemalli lisäämällä siihen uusia tietoja ja tarvittaessa myös uusia käsitteitä ja yhteyksiä. Käytännössä tarveanalyysissä käydään läpi jokainen tietotarve ja tarkastetaan käsittemallista, löytyykö käsittemallista kyseisen raportin tai ikkunan tarvitsemat yksittäiset tiedot. Jokainen tietotarve käydään siis erikseen läpi ja käsittemallia tarkennetaan tarpeen mukaan. (Hovi ym. 2005, 80, 81.)

Suunniteltavan tietokannan tarveanalyysivaiheessa tietotarpeita käytiin läpi haastattelemalla henkilökuntaa. Haastatteluissa pyrittiin saamaan selville, mitä tietoja työntekijä tarkalleen tarvitsee näkyviin milläkin hakuehdoilla. Näiden tietojen perusteella käsittemallia pystyttiin tarkentamaan paremmalle tasolle. Tarveanalyysiä tehdessä tuli ilmi, että esimerkiksi varaosien tietoihin tulisi saada lisätietokenttä, johon työntekijä voi merkitä tietoja kyseiseen varaosaan liittyen. Samoin ohjainlaitteisiin haluttiin lisätietokenttä, johon voi lisätä huomautuksia kyseiseen ohjainlaitteeseen liittyen. Tarveanalyysivaiheessa käsittemalli sai lopullisen muodon. Kuviossa 17 näkyy tulevan tietokannan ER-malli. Malli on toteutettu DBDesigner 4-ohjelmalla, joka on tietokantojen suunnitteluun tarkoitettu ilmainen ohjelma. ER-mallissa moni-moneen-yhteydet on purettu. Automalli- ja ohjainlaite-käsitteen väliin on lisätty yhteensopivuus-käsite. Tämä on oikeastaan tulevan tietokannan tärkein taulu, koska sitä kautta selviää, mikä ohjainlaite kuuluu millekin automerkille riippuen auton vuosimallista. Lisäksi vuosimalli-käsite on poistettu ja se on siirretty attribuutiksi yhteensopivuustauluun. Yhteensopivuus-käsitteen tietoja ovat mallinumero, tyyppinumero ja vuosimalli. Nämä yhdessä toimivat pääavaimena.

Automalli-käsitteen viiteavaimena automerkkitauluun toimii merkkinumero. Varaosa- ja yhteensopivuus-käsitteistä mallinumero toimii viiteavaimena automalli-käsitteeseen ja tyyppinumero ohjainlaite-käsitteeseen. Kategorianumero toimii viiteavaimena ohjainlaite-käsitteestä ohjainlaitekategoria-käsitteeseen.



Kuvio 17. Tietokannan ER-malli.

5.9 Normalisointi

Normalisoinnissa isot taulut hajotetaan pienemmiksi tauluiksi, jotta voidaan poistaa ylimääräistä ja kaksinkertaista tietoa. Lisäksi tämä auttaa välttämään tietojen lisäämiseen, päivittämiseen ja poistamiseen liittyviä ongelmia. Normaalimuoto on sääntöjoukko, jolla taulurakenteen kestävyys testataan ongelmien välttämiseksi. (Hernandez 2000, 30). Normalisoinnin (normalisation) tarkoitus on jalostaa tietorakenteita parempaan tallennusmuotoon. Normalisointiteoria käsittää viisi eri normaalimuotoa. Niistä kolme ensimmäistä on yleisimmät muodot ja riittää, että tietokanta saavuttaa kolmannen normaalimuodon. (Hovi ym. 2005, 86.)

Normalisointivaihe oli nopea tehdä kyseistä tietokantaa suunniteltaessa, koska taulujen tietorakenne oli kohtalaisen pieni. Jokaiseen tauluun sisältyi vain vähän tietoja ja taulut saavuttivat käsitellin loppuvaiheessa kolmannen normaalimuodon ilman, että niihin tarvitsi tehdä muutoksia. Suunnitteluputken mukaisesti normalisoinnin jälkeen seuraa tietokannan toteutus ja suorituskyvyn viritys. Toteutusvaihe on rajattu pois tästä opinnäytetyöstä. Taulut muodostuvat käsitellin käsitteistä ja käsitteiden tiedoista syntyy taulun sarakkeet. Yksi-moneen-yhteyksissä tulee viitevain siihen tauluun, joka on isä-lapsi-yhteyden lapsi-päässä. Myös tietotyypit ja eheyshdot määritellään siinä vaiheessa, kun käsitteistä muodostetaan tauluja.

6 TIETOKANNAN RAKENTEEN TESTAUS

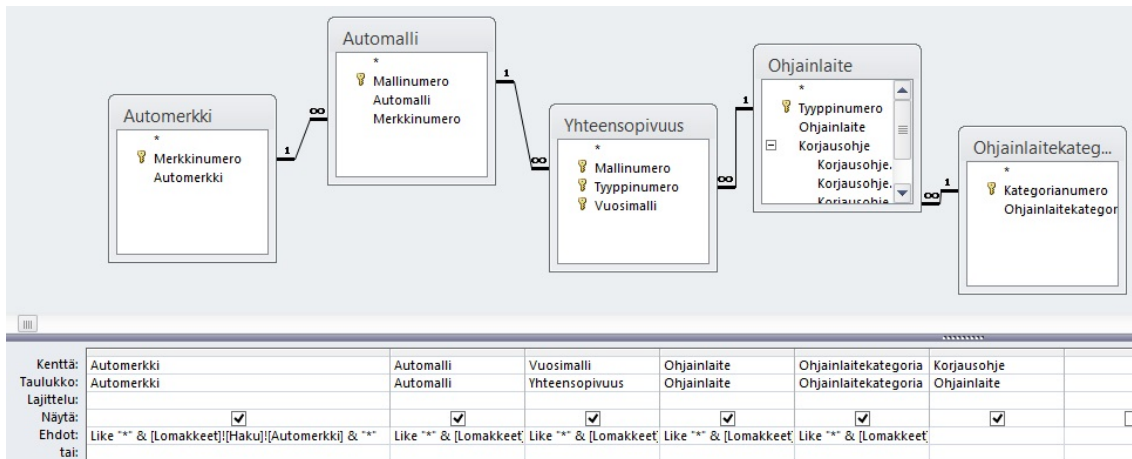
Tietokannan rakennetta on hyvä testata suunnittelun edetessä. Elehuollolle suunniteltavaa tietokantaa testattiin suunnittelun eri vaiheissa. Rakenteen testausta varten luotiin testitietokanta Accessiin. Accessin avulla tietokannan taulujen luonti on helppoa ja nopeaa. Tietokantaan luotiin kaikki tulevan tietokannan taulut. Tauluihin syötettiin sekä aitoa, että keksittyä tietoa. Seuraavissa kuvioissa näkyvät tiedot eivät siis välttämättä vastaa todellisuutta, vaan ne ovat testikäyttöön syötettyä tietoa. Kuviossa 18 on esimerkki automerkki- ja automalli -tauluista. Tauluihin on tallennettu eri automerkkejä ja -malleja. Viiteavaimena automalli -taulusta automerkki -tauluun toimii merkinnumero, joka viittaa, mille automerkille mikäkin automalli kuuluu.

Merkkinum	Automerkki
1	Audi
2	BMW
6	Citroen
5	Fiat
3	Ford

Mallinume	Automalli	Merkkinumero
1	A3	1
2	A5	1
3	A6	1
4	A4	1
5	X3	2
6	X5	2
7	Focus	3
8	Bravo	5
9	Idea	5

Kuvio 18. Automerkki- ja automalli -taulut.

Jotta tietokannan rakennetta pystyttiin testaamaan käytännössä, täytyi testausta varten luoda erilaisia testitapauksia. Testitapaukset luotiin vaatimusmäärittelyn toiminnallisten vaatimusten perusteella. Kuviossa 19 on esimerkki kyselystä, jota työntekijä voi käyttää asiakkaan ottaessa yhteyttä ohjainlaitteen korjauksen merkeissä. Kyselyyn valittiin automerkki-, automalli-, yhteensopivuus-, ohjainlaite- ja ohjainlaitekategoria-taulut.



Kuvio 19. Korjausohje-kysely Accessin rakennäkössä.

Kyselystä tehtiin myös hakulomake, jonka avulla kyselyn tekeminen on mielekkäämpää. Työntekijä voi hakea oikeaa vikakuvausta ja korjausohjetta asiakkaan antamien tietojen perusteella. Oikeaa korjausohjetta, johon myös tyypillisimmät vikakuvaukset on kirjattu, voi hakea joko kaikilla kuviossa 20 näkyvillä hakuehdoilla tai vain jollakin niistä. Hakuikkunan alla näkyy haun tulos. Tämä haku näyttää 2000 vuosimallin Ford Focusin abs-yksikön korjausohjeen. Hakutuloksen oikeassa laidassa olevaan sarakkeeseen on tallennettu linkki korjausohjeeseen.

Haku

Automerkki:

Automalli:

Vuosimalli:

Ohjainlaitekat...:

Ohjainlaite:

Automerkki	Automalli	Vuosimalli	Ohjainlaite	Ohjainlaitekat...
Ford	Focus	2000	Bosch 8	ABS moduuli

Liitteet

Liitteet (ajaa kohde kaksoklikkaamalla)

Bosch_8_Ohje.pdf

Kuvio 20. Korjausohjehaku.

Elehuollolla on varastossa ohjainlaitteita, joita voidaan vaihtaa asiakkaiden rikkoontuneiden ohjainlaitteiden tilalle esimerkiksi silloin, kun ohjainlaitetta ei pysty korjaamaan. Testausta varten varaosatauluun tallennettiin muutama ohjainlaite, joiden osanumerot ovat keksittyjä. Tietokantaan tehtiin kysely, joka palauttaa kaikki varastossa olevat ohjainlaitteet ja niiden määrät. Lisäksi kysely antaa tiedon siitä, mille automerkille ja mallille kyseinen ohjainlaite kuuluu. Kysely tehtiin kyselyn rakennenaikymässä. Kyselyyn otettiin mukaan automerkki, automalli ja varaosataulut. Tauluis- ta valittiin kyselyyn automerkki-, automalli-, osanumero-, määrä- ja lisätietoja -kentät. Kyselyn ehdoksi valittiin yksi tallennetuista osanumeroista. Kuviossa 21 näkyy kyselyn rakennenaikymä ja kyselyn tulos. Kysely antaa vastauksen sekä siihen, montako kappaletta kyseistä osaa on varas- tossa, että siihen, mille automallille osa käy. Lisäksi nähdään työntekijän mahdollisesti lisäämät lisätiedot kyseiseen osaan liittyen.

Kenttä:	Automerkki	Automalli	Osanumero	Määrä	Lisätietoja
Taulukko:	Automerkki	Automalli	Varaosa	Varaosa	Varaosa
Lajittelu:					
Näytä:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Ehdot:			"0022"		
tai:					

Automerkki	Automalli	Osanumero	Määrä	Lisätietoja
Audi	A3	0022	2	

Kuvio 21. Varaosakysely.

7 POHDINTA

Opinnäytetyön tarkoituksena oli suunnitella Elehuolto Oy:lle tietokanta, jonka avulla työntekijät voivat hallinnoida ohjainlaitteiden korjausohjeita. Tietokannan suunnittelu ei ollut minulle entuudestaan kovinkaan tuttu aihealue. Tämä teki työstä jonkin verran haastavaa. Haastavuutta lisäsi se, että tulevan tietokannan kohdealue oli minulle entuudestaan tuntematon.

Tietokannan suunnittelu aloitettiin tekemällä vaatimusmäärittely. Vaatimusmäärittelyä tehtiin haastatteleamalla Elehuollon työntekijöitä, tutustumalla heidän toimintaympäristöönsä ja nettisivustoon. Haastattelukertoja kertyi kaksi ja molemmilla kerroilla haastattelut tehtiin Elehuollon toimitiloissa. Ensimmäisellä kerralla sain karkean kuvan siitä, millaisesta tietokannasta olisi kyse. Keräsin vaatimuksia ranskalaisilla viivoilla paperille ja laadin niiden perusteella myöhemmin vaatimusmäärittelydokumentin.

Vaatimukset tarkentuivat sitä mukaa, kun ongelma-alueen ymmärrys lisääntyi. Vaatimusmäärittelyn pohjalta tehtiin käsiteanalyysi, jonka tuloksena oli karkea versio käsitemallista. Siinä vaiheessa ei vielä ollut tiedossa kaikkia tietokantaan tulevia tietoja. Tietotarpeita tarkennettiin tarveanalyysillä. Tarveanalyysivaiheessa vierailin Elehuollon toimitiloissa jolloin haastattelin toista kertaa henkilöstöä. Haastatteluissa pyrin saamaan vastauksia siihen, minkälaisia näkymiä työntekijät haluavat milläkin hakusanoilla. Muun muassa ohjainlaite ja varaosa-tauluihin haluttiin lisätä lisätietokenttä, johon työntekijät voivat tallentaa haluamaansa tietoa. Haastatteluiden perusteella käsitemalli voitiin tarkentaa lopulliselleen muotoon.

Jotta käsitemallin rakenteen toimivuus voitiin varmistaa, loin Accessilla testitietokannan. Tietokantaan loin käsitemallin mukaisesti kaikki taulut ja kentät sekä loin niiden väliset yhteydet. Tämän jälkeen lisäsin kaikkiin tauluihin tietoa. Osan tiedoista jouduin keksimään, koska minulla ei ollut tietoa esimerkiksi siitä minkä vuosimallin automalleille mikäkin ohjainlaite kuuluu. Seuraavaksi tein erilaisia kyselyitä testatakseni tietokannan toimivuutta. Loin myös haku- toiminnon, jonka avulla voi hakea oikeaa korjausohjetta automerkin, -mallin, vuosimallin ohjainlaitekategorian ja/tai ohjainlaitteen perusteella. Testauksen perusteella tietokannan rakenne toimi hyvin. Tietokannan avulla työntekijät löytävät korjausohjeet helposti ja nopeasti. Tein tietokannan tauluille myös taulumääritykset ja taulujen luontilauseet siltä varalta, että toimeksiantaja tarvitsee niitä tulevaisuudessa. Taulujen määritykset löytyvät liitteestä 1 ja luontilauseet liitteistä 2.

Opin projektin aikana paljon tietokannan suunnittelusta käytännössä. Vaikka suunniteltava tietokanta oli pieni, senkin suunnitteluun tuli kiinnittää huomiota, koska mikään tietokanta ei voi toimia hyvin, jos rakenne ei ole toimiva. Suunnittelun aikana oli aika haastavaa saada toimeksiantajalta tarkkaa kuvaa siitä, mitä tietokannasta tarkalleen ottaen halutaan hakea ja minkälaisia näkymiä hakutulosten perusteella tulisi näkyä.

Raportin kirjoittaminen vei paljon enemmän aikaa, kuin olin aluksi ajatellut. Tein mielestäni virheen siinä, että aloin suoraan kirjoittamaan raporttia samalla, kun suunnittelin tietokantaa. Olisi kannattanut mieluummin keskittyä alussa lukemaan erilaisia lähteitä ja tekemään muistiinpanoja niistä ja kirjoittaa tietoperusta kokonaan valmiiksi, ennen varsinaisen suunnittelun aloittamista. Näin olisin säästänyt paljon aikaa ja olisin pystynyt keskittymään paremmin yhteen asiaan kerrallaan.

Tietokannan suunnittelu ja koko opinnäytetyön tekeminen oli mielenkiintoinen projekti. Projektin aikana minua harmitti, etten sisällyttänyt myös tietokannan toteutusta opinnäytetyöhöni. Projektin seuraava askel on toteuttaa tietokannalle toimiva käyttöliittymä, jonka avulla työntekijöiden on helppoa käyttää tietokantaa. Käyttöliittymän teossa kannattaa kiinnittää huomiota myös käytettävyyteen. Käyttöliittymän teko voisi onnistua kohtalaisen helposti Accessilla. Opinnäytetyön loppuvaiheessa tutustuin myös Oraclen Application Expressiin ja myös testasin sitä jonkin verran. Myös tämän ohjelman avulla voitaisiin saada hyvä ja toimiva käyttöliittymä tietokannalle.

LÄHTEET

Chapple, M. Choosing a Database for Your Organization. Viitattu 1.4.2015, http://databases.about.com/od/administration/a/choosing_a_db.htm.

Crispin, L. 2012. Ideas for Eliciting Examples and Specifications. Viitattu 2.2.2012, <http://www.stickyminds.com/article/ideas-eliciting-examples-and-specifications>.

Foster, E. 2014. Software Engineering: A Methodical Approach. Viitattu 4.3.2015, [http://proquest.safaribooksonline.com.ezp.oamk.fi:2048/book/software-engineering-and-development/9781484208472/part-b-software-investigation-and-analy-ly-sis/9781484208489_ch04_xhtml?query=\(\(\(%22requirement+specification%22\)\)\)+AND+\(CATID%3ditdev+AND+PUBDATEYEAR%3d2014\)#snippet](http://proquest.safaribooksonline.com.ezp.oamk.fi:2048/book/software-engineering-and-development/9781484208472/part-b-software-investigation-and-analy-ly-sis/9781484208489_ch04_xhtml?query=(((%22requirement+specification%22)))+AND+(CATID%3ditdev+AND+PUBDATEYEAR%3d2014)#snippet).

Haikala, I. & Mikkonen, T. 2011. Ohjelmistotuotannon käytännöt. 12., uudistettu painos. Helsinki: Talentum media Oy.

Hamilton, R. 2013. Managing Writers. Viitattu 4.3.2015, [http://proquest.safaribooksonline.com.ezp.oamk.fi:2048/book/communications/writing/9781457182297/acquiring-technology/ch16s03_xhtml?query=\(\(\(%22The+requirements+specification%22\)\)\)#snippet](http://proquest.safaribooksonline.com.ezp.oamk.fi:2048/book/communications/writing/9781457182297/acquiring-technology/ch16s03_xhtml?query=(((%22The+requirements+specification%22)))#snippet).

Hernandez, M. 2000. Tietokannat. Suunnittelu ja toteutus. Suom. T. Kajala. Helsinki: Oy Edita AB.

Hovi, A., Huotari, J. & Lahdenmäki, T. 2005. Tietokantojen suunnittelu ja indeksointi. 2. laitos. Jyväskylä: Docendo Finland Oy.

Laine, H.a. Tietokantasovellus. Viitattu 24.3.2015, <https://www.cs.helsinki.fi/u/laine/tikas/material/ohjelmointi.html>.

Lecky-Thompson, G. 2005. Corporate Software Project Management. Hingham: Charles River Media, INC.

Lehtimäki, T. 2006. Ohjelmistoprojektit käytännössä. Helsinki: Readme.fi Oy.

MacDonald, M. 2013. Access 2013: The Missing Manual. Viitattu 25.3.2015, http://proquest.safaribooksonline.com.ezp.oamk.fi:2048/book/databases-and-reporting-tools/9781449359447/introduction/the_changes_in_access_2013_html#X2ludGVybmFsX0h0bWxWaWV3P3htbGikPTk3ODE0NDkzNTk0NDcIMkZ3aGF0X3lvdV9jYW5fZG9fd2l0aF9hY2Nlc3NfaHRtbCZxdWVyeT0=.

Nousiainen, E. UML-mallinnus. Luokkakaavio. Viitattu 19.2.2015, <http://www2.amk.fi/digma.fi/www.amk.fi/material/attachments/vanhaamk/etuotanto/5hNnpWMRw/luokkakaavio.pdf>.

Paakki, J. 2011. Ohjelmistojen vaatimusmäärittely. Viitattu 30.1.2015, <http://www.cs.helsinki.fi/u/paakki/Vaatimus-11-Luentokalvot-1.pdf>.

Pohjonen, R. 2007. Tietojärjestelmien kehittäminen. 3. painos. Helsinki: WSOY.

Winant, B. 2002. Requirement #1: Ask Honest Questions. Viitattu 2.2.2015, <http://www.stickyminds.com/article/requirement-1-ask-honest-questions?page=0%2C0>.

Automerkki-taulu

Kuvaus: Tauluun tallennetaan eri automerkit

Avainsarakkeet: Taulun perusavain on Merkkinumero

Taulun sarakkeet:

Merkkinumero	tietotyyppi:	int
	pituus:	4 merkkiä
	esitysmuoto:	1234
	arvoalue:	0-9
	merkitys:	yksilöi automerkin
	yksikäsitteinen:	on
	null-arvo:	ei sallittu
	oletusarvo:	seuraava vapaa numero

Automerkki	tietotyyppi:	varchar
	pituus:	20 merkkiä
	esitysmuoto:	vapaa
	arvoalue:	AA - ÖÖ
	merkitys:	automerkki
	yksikäsitteinen:	ei
	null-arvo:	ei sallittu
	oletusarvo:	ei ole

Automalli-taulu

Kuvaus: Tauluun tallennetaan eri automallit

Avainsarakkeet: Taulun perusavain on mallinumero, viiteavaimena toimii merkkinumero
automerkki-tauluun

Taulun sarakkeet:

Mallinumero	tietotyyppi:	int
	pituus:	4 merkkiä
	esitysmuoto:	1234
	arvoalue:	0-9

merkitys: yksilöi automerkin
yksikäsitteinen: on
null-arvo: ei sallittu
oletusarvo: seuraava vapaa numero

Automalli

tietotyyppi: varchar
pituus: 30 merkkiä
esitysmuoto: vapaa
arvoalue: AA -ÖÖ
merkitys: automallin nimi
yksikäsitteinen: ei
null-arvo: ei sallittu
oletusarvo: ei ole

Merkkinumero

tietotyyppi: int
pituus: 4 merkkiä
esitysmuoto: 1234
arvoalue: 0-9
merkitys: viiteavain automerkki-tauluun
yksikäsitteinen: on
null-arvo: ei sallittu
oletusarvo: seuraava vapaa numero

Ohjainlaitekategoria-taulu

Kuvaus: Tauluun tallennetaan eri ohjainlaitekategoriat

Avainsarakkeet: Taulun perusavain on kategorianumero

Taulun sarakkeet:

Kategorianumero

tietotyyppi: int
pituus: 4 merkkiä
esitysmuoto: 1234
arvoalue: 0-9
merkitys: yksilöi ohjainlaitetyypin

yksikäsitteinen: on
null-arvo: ei sallittu
oletusarvo: seuraava vapaa numero

Ohjainlaitekattegoria tietotyyppi: varchar
pituus 30 merkkiä
esitysmuoto vapaa
merkitys: ohjainlaitetyypin nimi
yksikäsitteinen: on
null-arvo: ei sallittu
oletusarvo: ei ole

Ohjainlaite-taulu

Kuvaus: Tauluun tallennetaan eri ohjainlaitetyypit

Avainsarakkeet: Taulun perusavain on tyyppinumero, viiteavaimena toimii kategorianumero ohjainlaitekattegoria-tauluun.

Taulun sarakkeet:

Tyyppinumero tietotyyppi: int
pituus: 4 merkkiä
esitysmuoto: 1234
arvoalue: 0 - 9
merkitys: yksilöi ohjainlaitetyypin
yksikäsitteinen: on
null-arvo: ei sallittu
oletusarvo: seuraava vapaa numero

Ohjainlaite tietotyyppi: varchar
pituus: 30 merkkiä
esitysmuoto: vapaa
Arvoalue: AA-ÖÖ
merkitys: ohjainlaitteen nimi
yksikäsitteinen: on
null-arvo: ei sallittu
oletusarvo ei ole

Korjausohje	tietotyyppi:	varchar (max)
	pituus:	ei määritelty
	esitysmuoto:	tiedostoliite
	Arvoalue:	ei määritelty
	merkitys:	sisältää korjausohjeen
	yksikäsitteinen:	on
	null-arvo:	sallittu
	oletusarvo	ei ole

Lisätietoja	tietotyyppi:	varchar (255)
	pituus:	255
	esitysmuoto:	vapaa
	Arvoalue:	AA-ÖÖ
	merkitys:	lisätekenttä
	yksikäsitteinen:	ei
	null-arvo:	sallittu
	oletusarvo	ei ole

Kategorianumero	tietotyyppi:	int
	pituus:	4 merkkiä
	esitysmuoto:	1234
	arvoalue:	0-9
	merkitys:	viiteavain ohjainlaitekategoria-tauluun
	yksikäsitteinen:	on
	null-arvo:	ei sallittu
	oletusarvo:	seuraava vapaa numero

Varaosa-taulu

Kuvaus: Tauluun tallennetaan varaosalaitteet

Avainsarakkeet: Taulun perusavain on osanumero, viiteavaimena toimii mallinumero automalli-tauluun ja tyyppinumero ohjainlaite-tauluun

Taulun sarakkeet:

Osanumero	tietotyyppi:	varchar
-----------	--------------	---------

	<p>pituus: 1 - 20 merkkiä</p> <p>esitysmuoto: vapaa</p> <p>arvoalue: 0 – 9, AA-ÖÖ</p> <p>merkitys: yksilöi ohjainlaitteen osanumeron</p> <p>yksikäsitteinen: on</p> <p>null-arvo: ei sallittu</p> <p>oletusarvo: ei ole</p>
Maara	<p>tietotyyppi: char</p> <p>pituus: 4 merkkiä</p> <p>esitysmuoto: 1234</p> <p>arvoalue: 0 – 9</p> <p>merkitys: varastossa olevien laitteiden määrä</p> <p>yksikäsitteinen: ei</p> <p>null-arvo: sallittu</p> <p>oletusarvo: ei ole</p>
Lisätietoja	<p>tietotyyppi: varchar (255)</p> <p>pituus: 255</p> <p>esitysmuoto: vapaa</p> <p>Arvoalue: AA-ÖÖ</p> <p>merkitys: lisätietokenttä</p> <p>yksikäsitteinen: ei</p> <p>null-arvo: sallittu</p> <p>oletusarvo: ei ole</p>
Tyypinnumero	<p>tietotyyppi: int</p> <p>pituus: 4 merkkiä</p> <p>esitysmuoto: 1234</p> <p>arvoalue: 0 - 9</p> <p>merkitys: viiteavain ohjainlaitetauluun</p> <p>yksikäsitteinen: on</p> <p>null-arvo: ei sallittu</p> <p>oletusarvo: seuraava vapaa numero</p>

Mallinumero	tietotyyppi:	int
	pituus:	4 merkkiä
	esitysmuoto:	1234
	arvoalue:	0-9
	merkitys:	viiteavain automallitauluun
	yksikäsitteinen:	on
	null-arvo:	ei sallittu
	oletusarvo:	seuraava vapaa numero

Yhteensopivuus-taulu

Kuvaus: Taulu toimii linkkitauluna ohjainlaite- ja automallitauluun.

Avainsarakkeet: Taulun perusavaimena toimii mallinumero, tyyppinumero ja vuosimalli.

Viiteavaimina toimii mallinumero automalli-tauluun ja tyyppinumero ohjainlaite-tauluun

Taulun sarakkeet:

Mallinumero	tietotyyppi:	int
	pituus:	4 merkkiä
	esitysmuoto:	1234
	arvoalue:	0-9
	merkitys:	viiteavain automallitauluun
	yksikäsitteinen:	on
	null-arvo:	ei sallittu
	oletusarvo:	seuraava vapaa numero

Tyyppinumero	tietotyyppi:	int
	pituus:	4 merkkiä
	esitysmuoto:	1234
	arvoalue:	0 - 9
	merkitys:	viiteavain ohjainlaitetauluun
	yksikäsitteinen:	on
	null-arvo:	ei sallittu
	oletusarvo:	seuraava vapaa numero

Vuosimalli	tietotyyppi:	smallint
------------	--------------	----------

pituus: 4 merkkiä
esitysmuoto: 1234
arvoalue: 1990 - 2100
merkitys: vuosimalli
yksikäsitteinen: ei
null-arvo: ei sallittu
oletusarvo: ei ole

Tietokantaan tulee yhteensä kuusi taulua: automerkki-, automalli-, ohjainlaitekategoria-, ohjainlaite-, yhteensopivuus- ja varasto-taulut. Seuraavassa on esitelty tietokantaan tulevien taulujen luontilauseet.

```
CREATE TABLE AUTOMERKKI
(
MERKKINUMERO INT NOT NULL AUTO_INCREMENT,
AUTOMERKKI VARCHAR(20) NOT NULL
PRIMARY KEY (MERKKINUMERO)
)
```

```
CREATE TABLE AUTOMALLI
(
MALLINUMERO INT NOT NULL AUTO_INCREMENT,
AUTOMALLI VARCHAR(30) NOT NULL,
PRIMARY KEY (MALLINUMERO),
FOREIGN KEY (MERKKINUMERO) REFERENCES AUTOMERKKI
ON DELETE CASCADE
)
```

```
CREATE TABLE OHJAINLAITEKATEGORIA
(
KATEGORIANUMERO INT NOT NULL AUTO_INCREMENT,
OHJAINLAITEKATEGORIA VARCHAR(30) NOT NULL
PRIMARY KEY (KATEGORIANUMERO)
)
```

```
CREATE TABLE OHJAINLAITE
(
TYYPPIUMERO INT(4) NOT NULL AUTO_INCREMENT,
OHJAINLAITE VARCHAR(30) NOT NULL,
KORJAUSOHJE VARCHAR(MAX),
LISATIETOJA VARCHAR(255),
PRIMARY KEY(TYYPPIUMERO),
FOREIGN KEY (KATEGORIANUMERO) REFERENCES (OHJAINLAITEKATEGORIA)
ON DELETE CASCADE
)
```

```
CREATE TABLE YHTEENSOPIVUUS
(
MALLINUMERO INT(4) NOT NULL,
TYYPPIUMERO INT(4) NOT NULL,
```

```
VUOSIMALLI SMALLINT,  
PRIMARY KEY(MALLINUMERO, TYYPPINUMERO, VUOSIMALLI),  
FOREIGN KEY (MALLINUMERO) REFERENCES AUTOMALLI,  
ON DELETE CASCADE,  
FOREIGN KEY (TYYPPINUMERO) REFERENCES (OHJAINLAITE)  
ON DELETE CASCADE  
)
```

```
CREATE TABLE VARAOSA  
(  
OSANUMERO VARCHAR(20) NOT NULL,  
MAARA CHAR(4) NOT NULL,  
LISATIEETOJA VARCHAR (255),  
PRIMARY KEY (OSANUMERO),  
FOREIGN KEY (MALLINUMERO) REFERENCES (AUTOMALLI)  
ON DELETE CASCADE,  
FOREIGN KEY (TYYPPINUMERO) REFERENCES (OHJAINLAITE)  
ON DELETE CASCADE  
)
```