

Opinnäytetyö AMK

Elektroniikka

Tietoliikennejärjestelmät

2017

Aleksi Hakkarainen

# MODUULIVALMISTUKSEN TESTAUSPROSESSIN KEHITYS



Aleksi Hakkarainen

## MODUULIVALMISTUKSEN TESTAUSPROSESSIN KEHITYS

Elektroniikkatuotannossa esiintyy erityyppisiä vikoja piirilevyn valmistamisessa sekä lopullisessa kokoonpanovaiheessa. Yleisiä vikoja ovat prosessiviat, vialliset komponentit ja testausvirheet. Vikoja tilastoimalla saadaan ymmärrys yleisimmistä vioista, sekä voidaan tutkia kuinka vikoja saadaan vähennettyä. Vikojen määrällinen vähentäminen lisää tuotannon tehokkuutta.

Tämän opinnäytetyön aiheena oli kehittää moduulivalmistuksen testausprosessia tutkimalla yleisimpiä vikoja. Toimeksianto opinnäytetyöhön saatiin Telesteltä. Työn tavoitteena oli tutkia millaisia ovat Telesten laajakaistavahvistimien yleisimmät viat sekä piirilevyvalmistuksen puolella, että tuotannon puolella. Piirilevyvalmistuksen yhteydessä piirilevyt testataan korttitestauksella, jossa karsitaan vikoja ennen kuin moduuli menee kokoonpanoon. Saaduista tuloksista voidaan etsiä syitä ja löytää erilaisia ratkaisuja kehittää prosessia.

Opinnäytetyössä perehdytään tietokantoihin, SQL-kyselykieleen sekä Excelin taulukonhallintatyökaluihin. Lisäksi työssä selostetaan lyhyesti korttitestauksesta sekä eri tuotannonvaiheista ja muista työhön liittyvistä ohjelmista. Työn alussa tutkittiin vikatilastoja tietokannasta. Tietokannoista saaduista tiedoista rakennettiin Exceliin erilaisia taulukoita, joista nähdään eri vikojen prosenttiosuuksia ja lukumääriä. Kun taulukot saatiin muodostettua, syvennyttiin vikojen aiheuttajiin tarkemmin keskustelemalla korjaajien kanssa.

Lopputuloksena saatiin selville yleisimmät viat ja niiden aiheuttajat. Telesten yleisimpänä viallisena komponenttina ovat muuntajat ja suuntakytkimet. Vikoja tilastoihin tulee myös testauksien epäonnistumisesta ja muista yksittäisistä vioista. Tuloksien perusteella aihetta voidaan jatkokehittää tutkimalla kuinka vikojen määrää saadaan vähennettyä.

### ASIASANAT:

Kehitys, testaus, prosessi, viat, moduulivalmistus

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Electronics degree programme | Telecommunication systems

2017 | 35

Instructor: Juha Nikkanen

Aleksi Hakkarainen

## DEVELOPMENT OF A MODULE MANUFACTURING TESTING PROCESS

Different kinds of defects occur in the production of electronics in both board assembly and forward assembly. Common defects are process failures, defective components and testing failures. The efficiency of the production can be improved by minimizing the amount of defects.

The purpose of this thesis was to develop the testing process of module manufacturing by researching the most common defects. This thesis was commissioned by Teleste Plc. The objective of this thesis was to research the typical defects of Teleste's broadband amplifiers in both board assembly and forward assembly phases. During board assembly, the modules are tested in a test fixture to minimize defects passing on to forward assembly.

This thesis focuses on databases, SQL language and table management tools of Excel. In addition, the thesis explains the principles of a test fixture, Teleste's different production phases and used software. Excel tables with different defects' percentages were created with the data retrieved from the database. Also discussions with repairers about the most common defects helped to better understand the true causes of the defects.

As the final result in this thesis the most common defects were found. Teleste's most common defects are transformers and directional couplers. Other defects are testing failures and sporadic defects. Further development needs to focus on researching different methods to decrease the amount of defects with the help of this thesis' results.

### KEYWORDS:

Development, testing, process, defects, module manufacturing

# SISÄLTÖ

<b>SANASTO</b>	<b>6</b>
<b>1 JOHDANTO</b>	<b>7</b>
<b>2 TIETOKANNAT JA TIETOJEN KÄSITTELY</b>	<b>9</b>
2.1 Tietokannat	9
2.1.1 Tietokantatyypit	9
2.1.2 Tietokantamallit	10
2.1.3 Relaatiotietokantamalli	12
2.2 SQL-kieli	14
2.2.1 SQL-kielen käyttökohteet	14
2.2.2 SQL-kyselyt	16
<b>3 OHJELMISTO JA KORTTITESTAUS</b>	<b>18</b>
3.1 TestStand -ohjelmisto	18
3.2 Database Viewer työkaluna	19
3.3 Vikatyypit	19
3.4 BADS eli Board Assembly Defects Software	20
3.5 Tuotannon vaiheet	21
3.6 Korttitestaus	22
<b>4 VIKATILASTOT</b>	<b>24</b>
4.1 Vikatilastojen laatiminen BADSilla	24
4.2 Vaihtaminen BADSista Database Vieweriin	25
4.3 Tilastojen tarkentaminen	26
4.4 Vikasyiden selvittely	30
<b>5 TULOKSET JA POHDINTA</b>	<b>31</b>
<b>6 LOPUKSI</b>	<b>34</b>
<b>LÄHTEET</b>	<b>35</b>

## KUVAT

Kuva 1. Yksinkertaistettu kuva tietokannan toiminnasta. [4]	9
Kuva 2. Hierarkkisen tietokantamallin perusrakenne. [5]	11
Kuva 3. Verkkotietokantamallin rakenne. [5]	12
Kuva 4. Esimerkki relaatiotietokantamallin taulujen välisistä yhteyksistä. [5]	13
Kuva 5. SELECT ja FROM komentojen palauttama kyselytulos. [7]	16
Kuva 6. WHERE-komennon lisäys rajoittaa hakutulosta. [7]	17
Kuva 7. TestStandin esimerkkisekvenssi. [8]	18
Kuva 8. Database Viewerin kyselytulokset.	19
Kuva 9. BADSin hakukentät.	21
Kuva 10. Korttitestausjigi kannen ollessa auki.	23
Kuva 11. BADS hakutulokset laitteelle A.	25
Kuva 12. SQL-kysely tehtynä Database Viewerillä.	26
Kuva 13. Kysely korjaukseen menneiden laitteiden lukumäärän selvittämistä varten.	27
Kuva 14. Prosenttiosuudet selvitetystä tiedoista.	28
Kuva 15. Komponenttikategorian ja ladontateknologian osuudet.	29
Kuva 16. Duplikaattien haku ja syiden selvitys.	29
Kuva 17. Tyhjiä selityksiä tuotannon puolen vikakirjauksissa.	30
Kuva 18. Solussa viritetyn laitteen mittaustulos, jossa vaste ylittää maskin.	32

## TAULUKOT

Taulukko 1. Vikatyypien selitykset.	20
Taulukko 2. Työvaiheet.	22

# SANASTO

BA	Board Assembly. Tuotannon vaihe, jossa piirilevy valmistetaan.
BADS	Board Assembly Defects Software. Telesten tuotantoa varten kehitetty ohjelma, jolla kirjataan viat tietokantaan.
DBA	Database administrator, tietokannan järjestelmänvalvoja.
DBMS	Database Management System. Tietokannan hallintajärjestelmä.
Diplekseri	Passiivinen yli-/alipäästösuodin, jolla samassa siirtolinjassa olevat kahden eri taajuusalueen signaalit voidaan erottaa tai yhdistää.
FA	Final Assembly. Tuotannon vaihe, jossa laite saa lopullisen muotonsa. Tämä vaihe sisältää esimerkiksi loppukokoonpanon ja testauksen.
IBM	International Business Machines Corporation. Vuonna 1888 perustettu teknologiayritys, joka on tunnettu suurtietokoneiden ja raskaiden palvelimien valmistajana.
Microsoft	Yhdysvaltalainen ohjelmistoalan yritys, joka on perustettu 1975. Tunnetaan kannettavien- ja pöytätietokoneiden Microsoft Windows-käyttöjärjestelmien valmistajana ja myyjänä.
MySQL	Vuonna 1995 luotu relaatiotietokantaohjelmisto.
Oracle	Oracle Corporationin relaatiotietokanta.
PostgreSQL	Avoimen lähdekoodin relaatiotietokantapalvelin.
RF	Radio Frequency. Radiotaajuus. RF-etuliite viittaa komponentteihin, jotka toimivat jollain tietyllä radioalueella.
SMD	Surface mounted device. Pintaliitoskomponentti, joka yleensä juotetaan piirilevylle koneladonnassa.
SQL	Structured Query Language. IBM:n kehittämä kyselykieli.
THD	Through-hole device. Käsiladottu komponentti.

# 1 JOHDANTO

Tämän opinnäytetyön tavoitteena oli saada selville yleisimmin esiintyvät viat, joita esiintyy Telesten laajakaistavahvistimien eri tuotannon vaiheissa. Kun saadaan tietoa yleisimmistä vioista ja niiden aiheuttajista, voidaan syventyä siihen, miten tuotantoa voitaisi kehittää. Laajakaistavahvistimien eri tuotannon vaiheilla tässä tapauksessa tarkoitetaan kahta eri vaihetta. Ensimmäinen vaihe on piirilevyn valmistaminen, jolloin piirilevylle ensin koneladotaan komponentit ja sen jälkeen käsiladotaan viimeiset komponentit. Toisella vaiheella tarkoitetaan loppukokoonpanoa tuotannon puolella, jossa laite kasataan lopulliseen muotoonsa.

Lisäksi opinnäytetyössä arvioidaan korttitestauksen tarpeellisuutta vertailemalla vikatilastoja moduulivalmistuksen ja tuotannon puolilla keskenään. Eräs korttitestauksen eduista on havaita mahdolliset viat jo heti moduulin valmistumisen jälkeen, jolloin se saadaan korjattua ennen lopullista kokoonpanovaihetta. Korttitestauksella vähennetään tällä tavalla tuotannon aikaa. Jos saman laitteen moduuli on molemmilla puolilla viallinen samasta syystä, voidaan ajatella korttitestaus tarpeettomaksi.

Yrityksessä on kehitetty moduulivalmistusta aiemmin hieman eri tavoilla ja aiheesta löytyi yksi mainittava opinnäytetyö. Markus Virtasen vuonna 2007 tekemässä opinnäytetyössään Elektroniikkatuotannon apuaineiden ja korjausvälineiden käyttöönottoprosessi [1] hän arvioi ja dokumentoi piirilevymoduulien tuotannossa käytettävät prosessiapuaineet ja työvälineet, jotta kokonaisuutena saataisiin toimiva prosessi. Opinnäytetyössään hän on selittänyt moduulituotannon vaihe vaiheelta tarkasti, joten tässä työssä asiaa sivutaan vain pintapuolisesti. SQL-kieleen on olemassa paljon erilaisia opinnäytetöitä. Eräs esimerkki on Eero Karvasen vuonna 2016 tekemä insinöörityö SQL-tietokantojen valvonta [2], jossa käydään läpi SQL-komentoja ja vertaillaan eri työkaluja. Korttitestaukseen liittyvän Mauri Aallon diplomityön Development of a test fixture manufacturing process [3] tarkoituksena oli kehittää ohjelmistoa, jolla voidaan automatisoida korttitestauksen valmistusprosessia mahdollisimman paljon.

Työssä perehdytään tietokantoihin ja tutkitaan tuotannon vaiheita, sekä käydään läpi muita työn kannalta oleellisia asioita, kuten käytettyä ohjelmistoa. Opinnäytetyön aihe rajattiin tutkimustyöhön, jonka jälkeen työtä voidaan jatkokehittää konkreettisemmin etsimällä erilaisia ratkaisuja vikojen vähentämiseen.

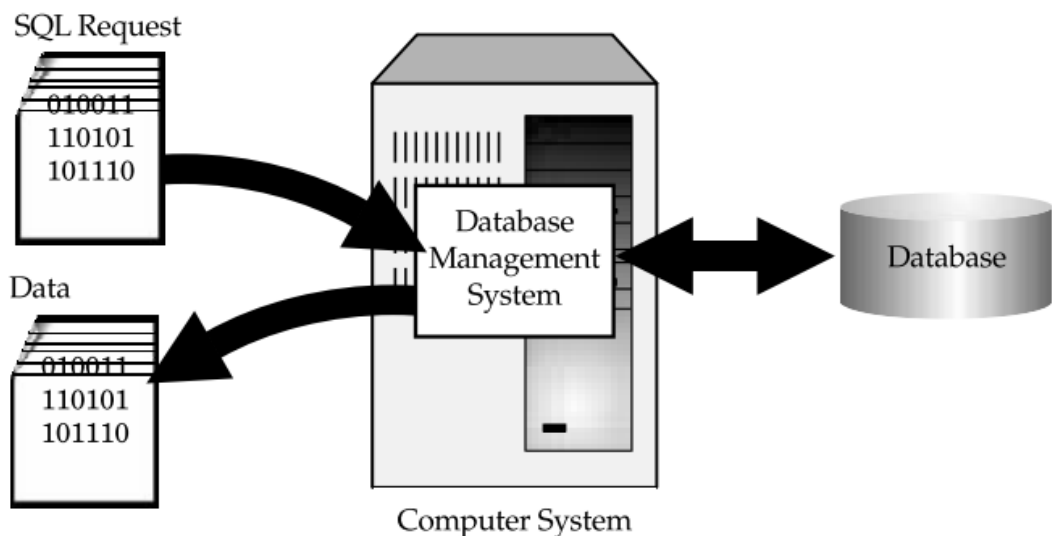
Alussa käsitellään teoriassa tietokantoja ja tietojen käsittelyä. Luvussa 3 käydään läpi laitteistoa ja ohjelmistoa, johon kuuluu Excelin lisäksi mm. National Instrumentsin TestStand-alusta, ja sen Database Viewer-ohjelma. Lisäksi käytössä oli Telesten omaan tuotantoon tehty BADS-ohjelma. Teoriaosion jälkeen varsinainen käytännön toteuttaminen alkaa kappaleesta 4. Käytännön osion jälkeen tulokset ja pohdinta.



## 2 TIETOKANNAT JA TIETOJEN KÄSITTELY

### 2.1 Tietokannat

Tietokannalla tarkoitetaan koostettua tietoa, joka on tietokoneen muistissa. Nämä koostetut tiedot ovat määrättyä tarkoitusta varten koottuja kokoelmia, joista saadaan erilaisilla hakumenetelmillä haettua, järjestettyä tai päivitettyä tietoa. Tietokannoilla on olemassa erilaisia tietokantatyyppejä sekä -malleja. Tietokantoja varten on markkinoilla olemassa erilaisia tietokantojen käsittelyyn tarvittavia ohjelmistoja, joista käytetään lyhennettä DBMS (Database Management System). DBMS ottaa vastaan kyselyitä ja palauttaa tietoa kuten kuvassa 1 on esitetty. Kun sille annetaan kyselypyyntö, hakee se tietokannasta tiedot ja palauttaa ne käyttäjälle tai ohjelmistolle. [4]



Kuva 1. Yksinkertaistettu kuva tietokannan toiminnasta. [4]

#### 2.1.1 Tietokantatyypit

Tietokantatyyppejä, joita nykyään käytetään, on olemassa kahdentyyppisiä: käyttötietokantoja ja analyttisiä tietokantoja. Nämä kaksi tietokantatyyppeä ovat saaneet nimityksensä niiden käyttökohteiden mukaan. [5]

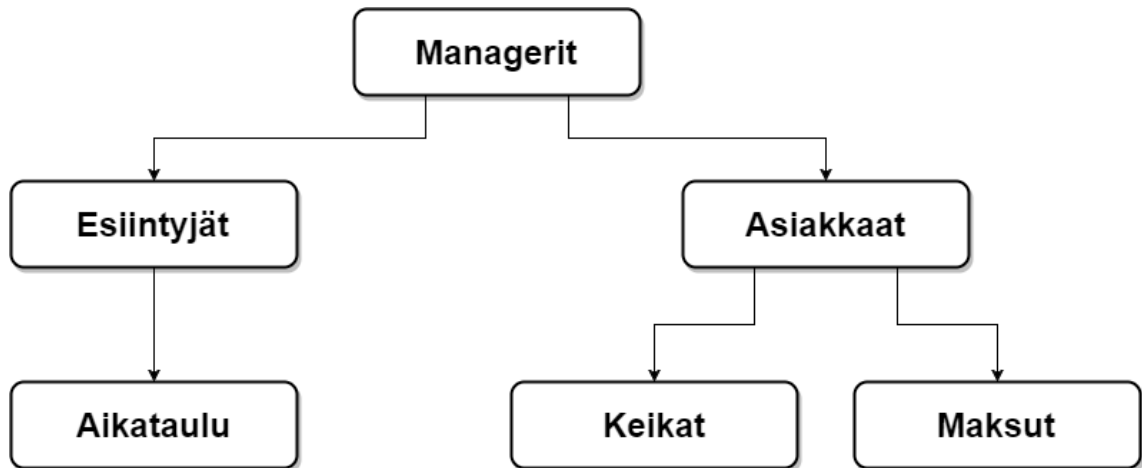
Käyttötietokantoja käytetään organisaatioissa, instituutioissa sekä liikeyrityksissä päivittäin. Tämän tyyppisiin tietokantoihin tallennetaan dynaamista dataa, eli tieto muuttuu jatkuvasti. Käyttötietokannat siis pitävät sisällään ajan tasalla olevaa tietoa. Esimerkkinä käyttötietokannoista voidaan pitää mm. inventaariotietokannat, potilaanseurantatietokannat ja tilaustenhoitotietokannat. [5]

Analyttiset tietokannat puolestaan ovat sellaisia tietokantoja, joihin tallennetaan ajasta riippumatonta tietoa. Tästä esimerkkinä liikeyrityksen kehityksen suunnan tarkastelu tai pitkäaikaisten suunnitelmien tekeminen. Analyttisessä tietokannassa tieto on staattista eli muuttumatonta. Tämä tarkoittaa sitä, että analyttisessä tietokannassa olevaa tietoa ei koskaan muuteta. [5]

### 2.1.2 Tietokantamallit

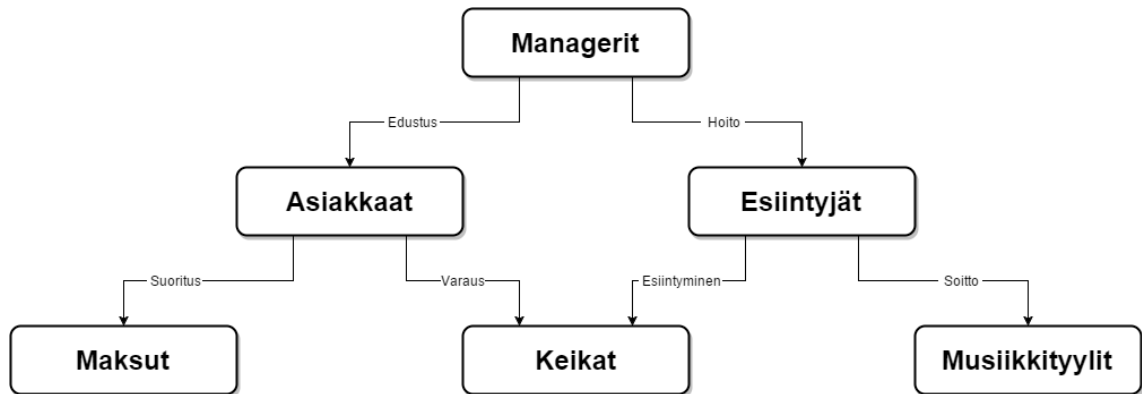
Tietokannoilla on olemassa myös erilaisia malleja. Ennen nykyisin eniten käytettyä tietokantamallia, relaatiotietokantamallia, on käytössä ollut yleisesti kahta tietokantamallia. Nämä mallit ovat hierarkkinen tietokantamalli sekä verkkotietokantamalli. [5]

Hierarkkinen tietokantamalli on nimensä mukaan järjestetty hierarkkisesti. Mallia voidaan kuvata helpoiten ylösalaisin käännetyllä puulla. Yksittäinen taulu toimii puun juurena ja muut taulut ovat juuresta lähteviä haaroja. Mallin suhteita kuvataan isä-lapsisuhteella. Tämä tarkoittaa sitä, että isätaululla voi olla monta lapsitaulua, mutta lapsitaululla vain yksi isätaulu. Taulujen linkitys tapahtuu yksiselitteisesti osoittimilla tai fyysisen järjestyksen perusteella. Tämän mallin ominaisuutena on se, että kohdetietoihin ei pääse muuten käsiksi kuin aloittamalla aina juuritaulusta. Eräs etu tässä mallissa on se, että tietoa voidaan hakea hyvin nopeasti, sillä rakenteet ovat yksiselitteisesti linkitettyjä. Haittapuoli on ylimääräisen tiedon tarve, mikä tarkoittaa että tässä mallissa ei voida rakentaa monimutkaisia suhteita taulujen välille. Kuvassa 2 esitetystä Aikataulu-taulusta löytyy samoja tietoja asiakkaasta kun Asiakkaat-taulusta, kuten esimerkiksi asiakkaan puhelinnumero. Kun yhteyksiä ei voida luoda erillisten taulujen välille, on vähintään välttämättömät tiedot siis sisällytettävä jokaiseen tauluun erikseen, jotta ne saadaan selville tarvittaessa. [5]



Kuva 2. Hierarkkisen tietokantamallin perusrakenne. [5]

Verkkotietokantamalli on ollut yritys kehittää ja korjata hierarkkisen tietokantamallin ongelmia. Parannuksena aikaisempaan verkkotietokantamallissa voidaan käyttää useampaa puuta, joilla on yhteisiä haaroja ja täten luoda useampia yhteyksiä. Lisäksi tässä mallissa voidaan tiedonhaku aloittaa mistä haarasta tahansa ja jatkaa tarpeen mukaan eteen- tai taaksepäin. Tämän mallin yhteyksiä luodaan ja niitä voidaan esittää joukkorakenteen avulla. Joukkorakenne on näkymätön rakenne, jonka tehtävä on liittää kaksi taulua toisiinsa siten, että toisesta taulusta tulee omistaja ja toisesta jäsen. Joukkorakenteiden avulla voidaan siis yhdistää yksi taulu useampaan kuin yhteen tauluun, eli niin sanottu yhdestä moneen -yhteys on mahdollista. Etuna tässä mallissa on hyvin nopea tiedonhaku ja monimutkaisemmat kyselyt kuin hierarkkisessa tietokantamallissa. Haittapuolia ovat, että käyttäjän tulee tuntee tietokannan rakenne erittäin hyvin, jotta käyttäjä osaa liikkua joukkorakenteiden läpi. Myöskään tietokannan rakennetta ei voi muuttaa ilman, että sen kanssa vuorovaikutuksissa olevia asiakasohjelmia muutetaan. Kuvassa 3 näkyvät joukkorakenteet verkkotietokantamallissa. Joukkorakenteita ovat mm. edustus Managerit-taulun ja Asiakkaat-taulun välillä, sekä soitto Esiintyjät-taulun ja Musiikkityylit-taulun välillä. [5]



Kuva 3. Verkkotietokantamallin rakenne. [5]

### 2.1.3 Relaatietietokantamalli

Kuten aiemmin todettu, nykyään yleisin käytössä oleva tietokantamalli on relaatiotietokantamalli. Se sai alkunsa, kun tri E. F. Codd, IBM:n tutkija, halusi 1960-luvun lopulta löytää uusia tapoja käsitellä suuriakin määriä tietoa. Koska sen aikaiset tietokantamallit ja -tyypit eivät täyttäneet hänen vaatimuksiaan, hän sai idean, että matematiikka ja rakenteiden käyttö voisivat ratkaista ongelmia, joita tietojen käsittelyssä kohdataan. Näitä ongelmia olivat mm. ylimääräinen tieto tai tietojen heikko eheys. [5]

Tiedot relaatiotietokantamallissa tallennetaan relaatioina, jotka näkyvät käyttäjälle tauluina. Jokainen relaatio muodostuu monikoista eli tietueista ja attribuuteista eli kentistä. Taulun tietueiden tai kenttien fyysisellä järjestyksellä ei ole mitään merkitystä. Jokainen tietue voidaan tunnistaa kentän avulla, jonka arvo poikkeaa muista. Kuinka relaatiotietokantamalli eroaa hierarkkisesta ja verkkotietokantamallista on se, että relaatiomallissa käyttäjän ei tarvitse tietää tietueen fyysistä sijaintia löytääkseen etsimänsä. Relaatiomallissa yhteydet voidaan jakaa yhdestä yhteen, yhdestä moneen sekä monesta moneen. Relaatiomallin yhteyksiä voidaan havainnollistaa kuvan 4 osoittamalla tavalla. Kuvassa laatikot ovat tauluja, laatikoiden otsikot ovat tietueita sekä tietueiden alla olevat arvot ovat kenttiä. Esimerkiksi Managerit-taulusta löytyy Managerin etunimi -tietue, jonka kentässä on tiedot Mikko, Keijo ja Katariina. [5]

**Managerit**

Managerin no.	Managerin etunimi	Managerin sukunimi	Palkkauspvm.	Managerin kotipuhelin
100	Mikko	Heinonen	16.05.95	555-3992
101	Keijo	Pietilä	15.10.95	790-3992
102	Katariina	Erkkilä	01.03.96	551-4993

**Asiakkaat**

Asiakasno.	Managerin no.	Asiakkaan etunimi	Asiakkaan sukunimi	Asiakkaan kotipuhelin	.....
9001	100	Seppo	Jalonen	553-3992	.....
9002	101	Sari	Mäkinen	790-3992	.....
9003	102	Ella	Putkisto	551-4993	.....

**Esiintyjät**

Esiintyjän no.	Managerin no.	Esiintyjän etunimi	Esiintyjän sukunimi	.....
3000	100	Juha	Sipilä	.....
3001	101	Marko	Jalo	.....
3002	102	Terhi	Virtanen	.....

**Keikat**

Asiakasno.	Esiintyjän no.	Keikan päivämäärä	Alkamisaika	Päätymisaika
9003	3001	01.04.96	13:00	15:30
9009	3000	13.04.96	21:00	01:30
9001	3002	02.05.96	15:00	18:00

Kuva 4. Esimerkki relaatiotietokantamallin taulujen välisistä yhteyksistä. [5]

Relaatiotietokantamallissa ei tarvitse samoja tietoja tallentaa jokaiseen tauluun erikseen, vaan ne voidaan hakea yhdellä kyselyllä, jossa linkitetään tauluja toisiinsa. [5]

## 2.2 SQL-kieli

Relaatiotietokantojen käyttöönoton myötä tietotekniikassa ja tiedonsiirrossa erääksi tukipilariksi on muodostunut SQL-kieli. Aluksi SQL-kieli on lähtenyt yksittäisestä kaupallisesta kokeilusta, mutta viimeisen kolmen vuosikymmenen aikana SQL on ansainnut itselleen miljardien arvoisen vuosittaisen markkinapaikan. Sadat markkinoilla olevista DBMS-ohjelmistoista (*Database Management System*, tietokannan hallintajärjestelmä) tukevat SQL-kieltä. Se on standardoitu osaksi jokapäiväistä tietotekniikkaa. Virallista kansainvälistä SQL-standardia on otettu käyttöön monissa käyttökohteissa ja sitä on laajennettu monesti. SQL löytyy kotona tietokoneista ja matkapuhelimista, sekä jopa autojen viihdejärjestelmistä. Jokainen laajasti tunnettu ohjelmistotuote on riippuvainen SQL:n tiedonkäsittelystä. SQL onkin muun muassa kolmen maailman suurimpien ohjelmistoyritysten Microsoftin, Oraclen ja IBM:n tietokantatuotteiden ydin. SQL on myös tärkeässä asemassa auttamassa avoimen lähdekoodin ohjelmistoja, kuten MySQL:ää ja PostgreSQL:ää. SQL on kulkenut pitkän matkan IBM:n tutkimusprojektista nykyiseen asemaansa tietokantojen käsittelyn ytimenä, sekä mahtavana markkinavoimana. [4]

SQL-kieli on työkalu, jonka avulla voidaan järjestellä, hallinnoida sekä hakea tietoa, jota on tallennettu johonkin tietokantaan. Alkujaan SQL tuli sanoista ”Structured English Query Language”, joka aluksi lyhennettiin ”SEQUEL”, mutta myöhemmin IBM sai tietää että kyseisellä lyhenteellä oli jo omistettu tavaramerkki. Lyhenteeksi tuli tämän jälkeen pelkästään nykyinen ”SQL”, ja jotta lyhenne saatiin vastaamaan nimeä, karsittiin sana ”English” nimestä pois kokonaan. Nimensä mukaan SQL on siis tietokonekieli, jota käytetään tietokantojen käsittelyssä. Tarkalleen ottaen SQL-kieltä käytetään ainoastaan aiemmin käsitellyn relaatiotietokantamallin kanssa. [4] [5]

### 2.2.1 SQL-kielen käyttökohteet

SQL-kieli oli aluksi tarkoitettu pelkästään kyselyitä varten, kuten nimessä sana ”query” (suom. kysely) siihen viittaakin. Nykyään kyselyt eivät kuitenkaan ole SQL-kielen ainoa tarkoitus. SQL hallitsee kaikkia funktioita, joita tietokannan hallitsemisjärjestelmä tarjoaa sen käyttäjille. Näitä funktioita ovat seuraavat:

- Tiedon määrittely, DDL (*Data Definition Language*):

SQL:n avulla käyttäjä voi määrittää tallennetun tiedon rakenteen ja järjestyksen, sekä tietojen väliset suhteet. DDL-komentoja ovat mm. CREATE, ALTER ja DROP. Komennolla CREATE voidaan luoda uusi taulu tai tietokanta, ALTER muuttaa taulua tai sen selitystä, sekä DROP poistaa valitut tiedot kokonaan.

- Tiedonhaku:  
SQL sallii käyttäjän tai ohjelmiston hakea tietoa tietokannasta, sekä käyttää sitä. Tässä tapauksessa tiedonhakua on esimerkiksi SQL-kyselyn tekeminen missä tahansa tietokannassa.
- Tiedonkäsittely, DML (*Data Manipulation Language*):  
SQL sallii käyttäjän tai ohjelmiston päivittää tietokantoja lisäämällä uutta tietoa, poistamalla vanhaa tietoa tai muokkaamalla jo jotain olemassa olevaa tietoa. Tähän kuuluvat mm. komennot SELECT, INSERT, UPDATE ja DELETE. Komennolla SELECT valitaan taulusta tietoja, INSERT lisää tietoa, UPDATE päivittää tai muuttaa tietoja, sekä DELETE poistaa tietoja.
- Käyttöoikeudet:  
SQL:n avulla voidaan suojella tallennettuja tietoja luvattomalta käytöltä rajoittamalla käyttäjän mahdollisuuksia hakea, lisätä tai muokata tietoja. Tietokannan käyttäminen edellyttää yhteyden luomista. Yhteyden luomiseen tarvitaan käyttäjä ja sille täytyy asettaa salasana. Lisäksi käyttäjien oikeuksia voidaan muuttaa asettamalla käyttäjille eri rooleja, kuten Connect, Resource ja DBA-rooli (*Database administrator*, tietokannan järjestelmänvalvoja). Connect-roolilla voidaan kysellä, lisätä ja päivittää. Resource-roolilla voidaan näiden lisäksi luoda proseduureja ja triggereitä. DBA-roolilla on järjestelmänvalvojana kaikki käyttöoikeudet ja tämä rooli on yleisesti vain harvoilla.
- Tiedon jakaminen:  
SQL hallinnoi tiedon jakamista siten, että yhtäaikaisten käyttäjät voivat tehdä tietokantaan muutoksia poistamatta toisen käyttäjän samanaikaisesti tekemiä muutoksia.
- Tiedon eheys:  
SQL luo tietokantaan rajoitteita, joilla suojellaan tietokantaa korruptoitumasta johtuen esimerkiksi järjestelmän vikaantumisesta.

Näiden asioiden pohjalta voidaan siis sanoa, että SQL on toiminnoiltaan kieli, jolla voidaan ohjata tietokannan hallintojärjestelmää eri toiminnoilla.

Puhuttaessa tietokonekielistä, kuten C-kieli tai Java, on SQL-kielessä on merkittäviä eroja muihin kieliin nähden. SQL kertoo, mitä käyttäjä haluaa tietokoneen tekevän sen sijaan että kertoo, kuinka käyttäjä haluaa tietokoneen sen tekevän. Toisin sanottuna SQL ei ole kielenä niinkään menetelmällinen, vaan selittävä ja kuvaileva. SQL on erittäin tehokas tietokonekieli ja lopulta kuitenkin melko helppo oppia käyttämään. [4]

### 2.2.2 SQL-kyselyt

SQL kehitettiin vastaamaan relaatiotietokantamallia. Jotta relaatiotietokantaa pystyy ”komentamaan”, tulee käyttäjän osata joitain peruskomentoja. SQL:ssä käskyt voidaan kirjoittaa pienillä tai isoilla kirjaimilla. Lisäksi SQL ei vaadi tiettyä rivitystä tai välilyöntien paikkaa. Tästä syystä lausekkeiden muotoilu voi helpottaa sen ymmärtämistä. Myös lausekkeiden ylläpito tulee helpommaksi, jos niitä tarvitsee myöhemmin muuttaa. [6]

SQL:n peruskomentoja ja avainsanoja ovat SELECT, FROM ja WHERE. SELECT-komennolla valitaan, mitä haetaan tietokannasta. FROM-komennolla määritellään mistä tämä kyseinen tieto haetaan. WHERE-komento on määrittelevä komento, jolla annetaan kyselylle esimerkiksi rajoittavia tekijöitä. Yksinkertainen tietokanta voi koostua esimerkiksi yrityksen palkkalistoilla olevista työntekijöistä. Tämän tietokannan tauluista voi löytyä esimerkiksi työntekijän etu- ja sukunimi, palkkauspäivä ja kotikaupunki. Kuvassa 5 on kuvitteellisen yrityksen työntekijälista sen jälkeen, kun tietokantaan on suoritettu seuraava kysely:

```
SELECT EmployeeID, FirstName, LastName, HireDate, City FROM Employees
```

	EmployeeID	FirstName	LastName	HireDate	City
	1	Nancy	Davolio	1/5/1992 12:00:00 AM	Seattle
	2	Andrew	Fuller	14/8/1992 12:00:00 AM	Tacoma
	3	Janet	Leverling	1/4/1992 12:00:00 AM	Kirkland
	4	Margaret	Peacock	3/5/1993 12:00:00 AM	Redmond
	5	Steven	Buchanan	17/10/1993 12:00:00 AM	London
	6	Michael	Suyama	17/10/1993 12:00:00 AM	London
	7	Robert	King	2/1/1994 12:00:00 AM	London
	8	Laura	Callahan	5/3/1994 12:00:00 AM	Seattle
	9	Anne	Dodsworth	15/11/1994 12:00:00 AM	London

Kuva 5. SELECT ja FROM komentojen palauttama kyselytulos. [7]



Tällä tavalla saadaan palautettua tietokannasta löytyvät tiedot kyseisillä kriteereillä. Jos halutaan palauttaa kyselyllä kaikki sellaiset henkilöt, jotka asuvat Lontoossa, lisätään kyselyyn vielä komento WHERE. Lisätään seuraava lauseke kyselyyn ja siitä saadaan kuvan 6 mukainen tulos:

```
SELECT EmployeeID, FirstName, LastName, HireDate, City FROM Employees
WHERE City = 'London'
```

	EmployeeID	FirstName	LastName	HireDate	City
	5	Steven	Buchanan	17/10/1993 12:00:00 AM	London
	6	Michael	Suyama	17/10/1993 12:00:00 AM	London
	7	Robert	King	2/1/1994 12:00:00 AM	London
	9	Anne	Dodsworth	15/11/1994 12:00:00 AM	London

Kuva 6. WHERE-komennon lisäksi rajoittaa hakutulosta. [7]

SQL-kielessä voidaan lisäksi käyttää erilaisia operaattoreita. Näitä operaattoreita ovat esimerkiksi tähti (\*), jolla palautetaan kaikki taulun sarakkeet kyselystä. Toinen esimerkki on %-merkki, jolla voidaan korvata merkkejä merkkijonosta. Jos halutaan hakea jostain taulusta kaikki henkilöt, joiden sukunimi alkaa kirjaimella S, voidaan se toteuttaa lisäämällä kyselyyn johonkin väliin seuraava rivi:

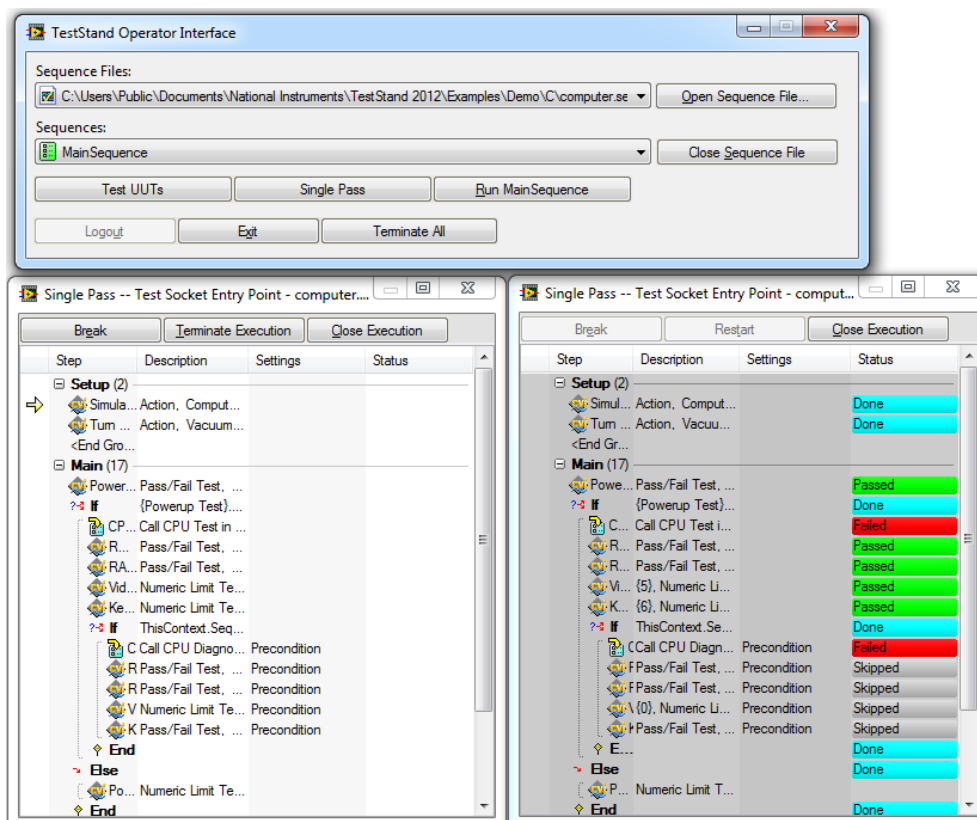
```
WHERE LastName like 'S%'
```

Lisäksi suurempi kuin- tai pienempi kuin -merkit (> ja <) käyvät SQL-kyselyissä. Loogiset operaattorit kuten AND, NOT, OR toimivat myös. Jos halutaan hakea tietoa jonkun kahden eri arvon väliltä, esimerkiksi aikaväliltä, voidaan aikojen välillä käyttää sanaa BETWEEN, jolloin kysely palauttaa kahden arvon väliset tiedot. [6]

## 3 OHJELMISTO JA KORTTITESTAUS

### 3.1 TestStand -ohjelmisto

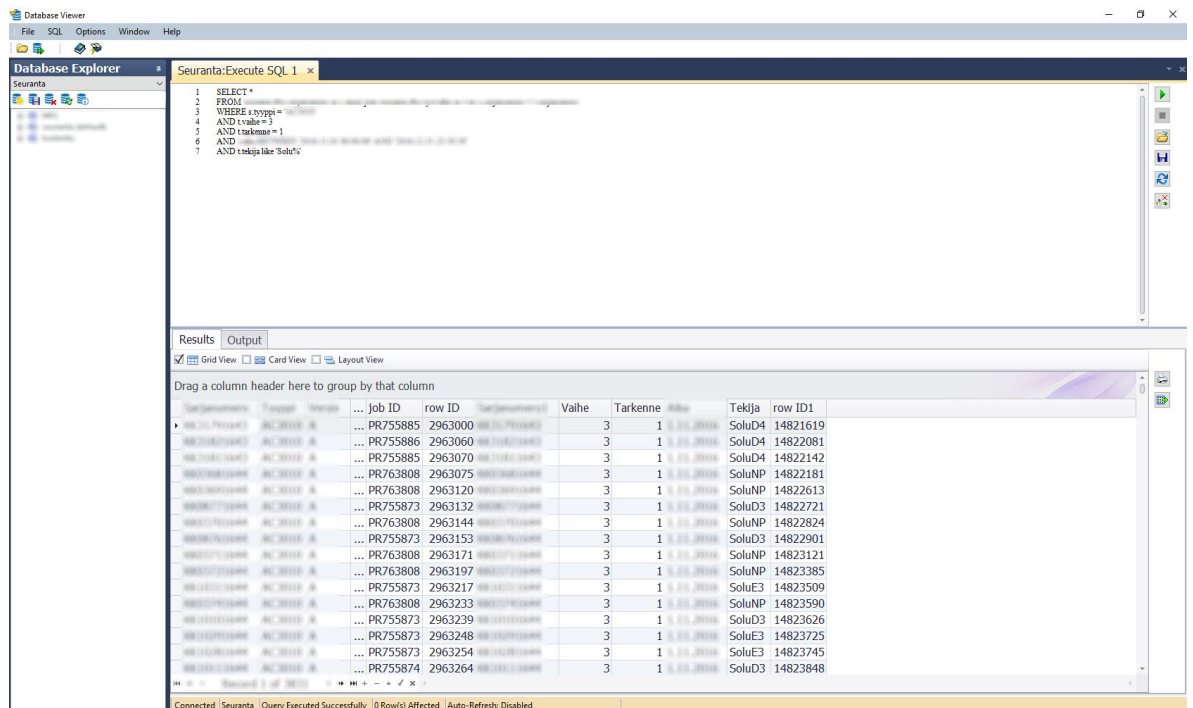
TestStand on National Instrumentsin kehittämä ohjelmisto testausten hallintaan. Sillä voidaan luoda testisekvenssejä, joilla saadaan suoritettua mitä tahansa koodikieltä käyttävää ohjelmamoduulia integroimalla koodikieliä keskenään. Kuvan 7 TestStand-ikkunassa vasemmalla puolella näkyy esimerkkinä sekvenssi ennen suoritusta ja oikealla puolella suorituksen jälkeen. Kuvan sekvenssissä on suoritettu demo TestStandin omasta esimerkikansiosta, jolla simuloidaan tietokoneen sisäistä diagnostiikkaa. [8]



Kuva 7. TestStandin esimerkisekvenssi. [8]

### 3.2 Database Viewer työkaluna

Database Viewer on osa TestStand -ohjelmistoa. Se on tietokannan hallintajärjestelmä, jolla voi tarkastella tai muuttaa tietoa tietokannan tauluista sekä suorittaa SQL-kyselyitä. Kuvassa 8 on suoritettu Database Viewerillä esimerkikysely, mistä näkyy mm. haetun laitteen sarjanumerot ja solu, missä työ on aloitettu. [9]



Kuva 8. Database Viewerin kyselytulokset.

### 3.3 Vikatyypit

Telesten tietokannasta haetut viat on jaoteltu eri vikatyypeittäin. Nämä vikatyypit määritellään laitteen vikatietoihin korjaajan paikalla korjaajan toimesta. Riippuen vian aiheuttajasta ja vian paikasta korjaaja valitsee vikatyypeistä sille sopivimman. Vikatyypeillä on numerot 1-10 ja niille on jokaiselle oma selitys. Taulukossa 1 on lueteltu nämä vikatyypit.

Taulukko 1. Vikatyypin selitykset.

Vikatyypin	Selitys
1	Juotos auki
2	Juotos oikosulku
3	Komponentti viallinen
4	Komponentti väärinpäin
5	Komponentti puuttuu
6	Väärä komponentti
7	Väärin säädetty, testaus
8	Materiaalivika
9	Suunnitteluvika
10	Muu syy

### 3.4 BADS eli Board Assembly Defects Software

BADS on Telesten tuotantoon kehitetty ohjelma, jolla saadaan tuotannossa fyysinen tai ohjelmallinen vika kirjattua tietokantaan, sekä sillä voi myös tarkastella niitä. BADS:n avulla on tarkoitus tallentaa tuotteiden eri moduuleissa ilmeneviä virheitä lisäämällä virhetietoja vastaaviin kokoonpanokuviin ja tallentamalla tietoja tekstitiedostoihin ja tietokantaan. BADSilla saadaan moduulikohtaista dataa niissä esiintyvistä virheistä sekä visuaalisena datana että taulukoituna. Visuaalisena datana BADS esittää viat ympyröinä kyseessä olevan piirilevyn piirustuksissa. Esimerkiksi kun loppukokoonpanovaiheessa tuotetta testataan, eikä se täytä sille asetettuja vaatimuksia joltain testauksen osalta, täytyy se lähettää korjaukseen. Korjauksessa korjaaja paikantaa vian, korjaa laitteen ja kirjaa vikaselostuksen tietokantaan parhaansa mukaan. Kun tietokannasta halutaan tarkastella vikoja, BADS:n käyttäjä kirjoittaa kuvassa 9 näkyviin hakukenttiin haluamansa tiedot, kuten tuotteen nimen tai moduulin. [10]

The screenshot shows a software interface for searching records. It is divided into several sections:

- Hakutermit (Search terms):** Contains input fields for 'Sarjanumero' (Serial number), 'Tyyppi' (Type), and 'Moduuli' (Module). Below these are fields for 'Steppi' (Step) and 'Selitys' (Explanation).
- Aikaväli (Time interval):** Shows 'Alkaen' (Start) as 8.09.43,000 on 16.3.2017 and 'Päättyen' (End) as 9.09.43,980 on 5.5.2017.
- Tallenna tiedostoon (Save to file):** Includes a file path 'C:\Temp\Testi.txt' and a 'TALLENNA' button.
- Filters:** A section with 'Lisää olemassaolevaan' checkbox and a 'TALLENNA' button.
- Buttons:** 'HAE' (Search), 'TYHJENNÄ' (Clear), and 'POIS' (Exit) buttons are on the right side.
- Bottom section:** Contains four spinners for 'Vaihe' (0), 'Tarkenne' (0), 'Vika' (0), and 'Havaintopaikka' (0), followed by a 'Komponentti' field.

Kuva 9. BADS:n hakukentät.

Lisäksi BADS:lla voidaan rajata hakutuloksia esimerkiksi tietylle aikavälille tai eri havaintopaikkaan. Havaintopaikka voi olla numeroarvona joko 12 eli moduulivalmistus tai 2 eli tuotannon puoli. [10]

### 3.5 Tuotannon vaiheet

Telesten tuotannossa työskennellään soluissa. Yhden laitteen valmistus alkaa moduulivalmistuksesta. Moduulivalmistuksessa piirilevyille ensin koneladotaan komponentit ja sen jälkeen se menee käsiladontaan, jossa moduuli valmistuu. Tämän jälkeen, jos moduuli on kortsitestattava, se menee kortsitestaukseen. Mikäli moduuli ei läpäise kortsitestausta, se korjataan. Kun moduuli on läpäissyt kortsitestauksen, se siirtyy moduulivalmistuksesta välivarastoon. Välivarastosta valmiit moduulit siirtyvät kokoonpanoon tilausten mukaan. Kokoonpanossa laite valmistetaan kunkin laitteen ohjeiden mukaan. Laite voi esimerkiksi koostua virtalähteestä, moduulista, liittimistä ja erilaisista lisäosista. Aluksi laitteelle suoritetaan alkukokoonpano, jonka jälkeen laite siirtyy viritykseen ja testaukseen. Mikäli laite ei läpäise viritystä tai testauksia ja siinä havaitaan vika, se siirtyy korjaukseen. Kun laite läpäisee molemmat, se siirtyy loppukokoonpanoon ja pakkaukseen. Lopulta pakattu laite lähetetään solusta jälleen välivarastoon, josta se menee eteenpäin asiakkaalle. Kuten vikatyypeillä, on myös työvaiheilla omat numerot ja niiden selitykset. Taulukossa 2 on työvaiheet ja niiden selitykset.

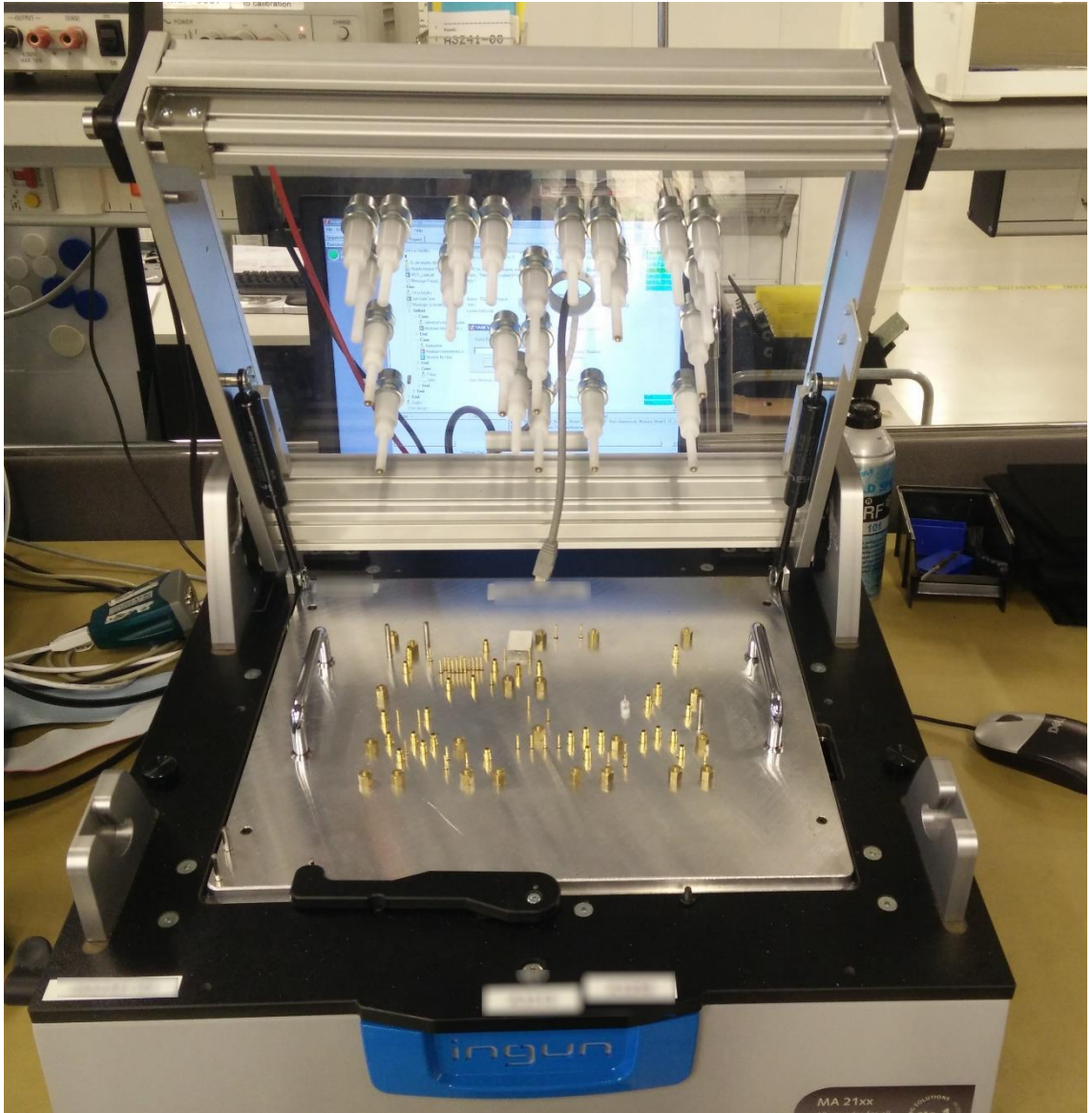
Taulukko 2. Työvaiheet.

Työvaihe	Selitys
2	Ladonta
3	Alkukokoonpano
4	Korttitestaus
5	Kokoonpano
6	Viritys
7	Lopputestaus
8	Loppukokoonpano
10	Pakkaus
11	Lähetys, välivarasto
12	Romutus
13	Vika
14	Palautus

Eri tuotteilla on olemassa eri työvaiheet, joten kaikille tuotteille ei välttämättä tule esimerkiksi työvaihetta 5 tietoihin.

### 3.6 Korttitestaus

Osa Telesten laitteiden moduuleista testataan korttitestauksella. Korttitestaus tapahtuu moduulivalmistuksen ohessa. Kuvassa 10 näkyy korttitestausjigi kansi avattuna. Korttitestausjigi on kannellinen laite, jossa on neulapeti. Neulapeti koostuu johtimista, jotka ovat yhteydessä piirilevyn testipisteisiin, kun kansi suljetaan. Valmistettu moduuli laitetaan korttitestausjigiin, jossa sille suoritetaan testaus. Eri moduulityypeille on olemassa eri testisekvenssit, jotka niille suoritetaan.



Kuva 10. Korttitestausjigi kannen ollessa auki.

Piirilevyjen layout eli pohjapiirros määrittää jokaiselle neulalle oman sijaintinsa. Kannessa on pieni reikä USB-johdolle, joka kytketään piirilevyn testauksen yhteydessä.

## 4 VIKATILASTOT

Työssä tiedonhankinta aloitettiin BADS-ohjelmasta. Sen käyttäminen alkuun oli hyvä ratkaisu, sillä sen käyttäminen ei edellytä juuri minkäänlaista kokemusta tietokannoista. Sitä käyttämällä pääsi hyvin tarkastelemaan tyypillisimpiä vikoja. Tarkastelussa oli kahden laitteen vikatilastot, jotka nimetään laitteeksi A ja B. Molemmat laitteet ovat laajakaistavahvistimia. Laitteelle A suoritetaan korttitestaus. Laitteelle B ei suoriteta korttitestausta moduulivalmistuksen aikana. Sen moduuli kulkee moduulivalmistuksesta suoraan välivarastoon ja sieltä tuotannon puolelle, kun kyseistä laitetta aletaan valmistaa asiakkaalle. Vikatilastoille valittiin tietty ajanjakso, jolta niitä tarkasteltiin.

### 4.1 Vikatilastojen laatiminen BADSilla

BADSin hakutulosten perusteella laitteen A suurin vikojen aiheuttaja on tyypiltään numero 3, eli viallinen komponentti. Tämän kyseisen vikatyypin tiedoista löytyi suurimpana vikojen aiheuttajana komponentti T3, joka on testipisteen muuntaja. Lisätiedoissa ei tälle komponentille oltu kerrottu mitään tarkentavaa, eli ei siis ollut tässä vaiheessa tietoa, mihin komponentin viallisuus vaikuttaa tai miten se oltiin havaittu. Tämä tarvitsi lisäselvitystä. Toisena vikatilastoissa oli vikatyypin 5 eli puuttuva komponentti. Tämän kategorian viat johtuvat vikaselitysten mukaan suurimmaksi osaksi siitä, että komponenttien huono sijoittelu on aiheuttanut harmia tuotannon puolella. Piirilevyissä käytetyt komponentit ovat nykyään pintaliitoskomponentteja, jotka eivät kestä suurta mekaanista rasitusta. Jos komponentti on sijoitettu lähelle ruuvipaikkaa, voi se tuotannossa saada iskuja esimerkiksi ruuvinvääntimestä, jolloin se irtoaa joko osittain tai kokonaan. Seuraavana tilastoissa on vikatyypin 7, joka on selitykseltään väärin säädetty, joka tarkkaan ottaen viittaa solussa testauksen epäonnistumiseen. Osaa laitteista ei saada viritettyä työpisteellä vaatimusten mukaisiksi, joten ne lähetetään siitä syystä korjaukseen. Laite ei välttämättä ole viallinen, vaan se vaatii pientä hienosäätöä, kuten komponenttien irrottelu tai niiden lisäämistä. Sarja- ja rinnankomponenttien arvoja muuttelemalla saadaan laitteen vahvistus muuttumaan samalla. Näillä toimenpiteillä saadaan laite asiakkaan vaatimusten mukaiseksi. Kuvassa 11 näkyy tulokset heti haun suorituksen jälkeen.





tilastoa. Aluksi tilastossa oli yleiskatsaus laitteiden A ja B vioista. Aloitetuista tuotteista ja korjaukseen menneistä tuotteista saatiin prosenttiosuudet laskettua. Vikatyypeistä saatiin myös omat prosenttiosuudet.

### 4.3 Tilastojen tarkentaminen

Korjaukseen menneiden ja vikatyypien prosenttiosuuksien lisäksi tarvitaan muuta tilastoa, joten tilastoja täytyi laajentaa. Näitä tilastoja olivat mm. Telesten ladotun komponentin vikatodennäköisyys per vikatyyppi, ladontateknologia sekä komponentti. Lisäksi haluttiin tietää, mikä on korttitestauksen kiinni saama osuus viallisista. Tilastoihin tuli myös komponenttikategorian yleisin vikasyy. Haluttiin myös tietää, että esiintyykö sama moduuli viallisena sekä moduulivalmistuksen että tuotannon puolella. Näihin tilastoihin kaikki tieto on haettu Database Viewerillä. Sen avulla tietojen hakeminen onnistuu muutaman yksinkertaisen lausekkeen opetteluun jälkeen. Lisäksi Database Viewerillä pystyy yhdistämään tauluja monimutkaisemmilla kyselyillä. Kuvassa 12 näkyvässä kuvakaappauksessa on SQL-kysely tehtynä Database Viewerillä.

```

1 SELECT
2 FROM
3 WHERE s.tyyppi =
4 AND t.vaihe = 3
5 AND t.tarkenne = 1
6 AND t.tekija like 'Solu%'
7 AND t.aika BETWEEN
8 AND s.sarjanumero like 'KK%'

```

Results Output

Grid View  Card View  Layout View

Drag a column header here to group by that column

id	versio	vaihe	tekija	tarkenne	...
1	A		3 SoluNP		1
2	A		3 SoluD1		1
3	A		3 SoluD1		1
4	A		3 SoluD1		1
5	A		3 SoluD1		1
6	A		3 SoluE3		1
7	A		3 SoluD2		1
8	A		3 SoluD1		1
9	A		3 SoluD2		1
10	A		3 SoluC4		1
11	A		3 SoluD1		1
12	A		3 SoluC4		1
13	A		3 SoluD2		1
14	A		3 SoluD3		1
15	A		3 SoluD3		1
16	A		3 SoluD3		1
17	A		3 SoluD3		1
18	A		3 SoluD1		1
19	A		3 SoluA4		1
20	A		3 SoluD1		1

Kuva 12. SQL-kysely tehtynä Database Viewerillä.

Kyselyssä on haettu seuranta-tietokannasta aloitetut tuotteet. Jotta nämä tuotteet saadaan listattua, täytyy yhdistää kaksi eri taulua tietokannasta inner join-komennolla. Tämä on yksi huomattava etu Database Viewerissä BADSiin verrattuna, sillä BADSilla ei voi suorittaa yhdistettyjä kyselyitä. Solussa oleva tieto aloitetusta tuotteesta tallentuu tietokantaan sarjanumerona ja työvaihe tallentuu erilliseen tauluun. Nämä tiedot yhdistämällä saadaan siis työvaihe-taulusta tuotteen työvaihe ja sarjanumero-taulusta laitteen tyyppi. AND-komennoilla voidaan kyselylle antaa tarkentavia määritelmiä, kuten esimerkiksi aikaväli, sekä aloitusvaiheeksi 3. Lisäksi varmistetaan että kyseessä on Telesten tuotannossa aloitetut tuotteet antamalla sarjanumerolle määritelmä, että se alkaa muodossa KK sekä tekijä alkaa muodossa Solu. Kyselyn alapuolella kuvassa 12 näkyy osa hakutuloksista. Kyselystä saadaan pelkkä aloitettujen määrä korvaamalla SELECT-sarakkeen jälkeinen osio tekstillä count(\*), jolloin kysely palauttaa taulusta löytyneiden kenttien lukumäärän numerona. Aloitettujen tuotteiden lukumäärän selvittämisen jälkeen haettiin korjaukseen menneiden määrä kuvan 13 mukaisella kyselyllä.

```

1  SELECT count(*)
2  FROM s
3  WHERE s.moduli = 'SOLU'
4  AND s.aika BETWEEN '2017-01-01' AND '2017-12-31'
5  AND s.havaintopaikka like '2'
```

Kuva 13. Kysely korjaukseen menneiden laitteiden lukumäärän selvittämistä varten.

Aloitettujen tuotteiden ja korjaukseen menneiden lukumäärästä saatiin laskettua prosenttiosuudet. Nämä kyselyt suoritettiin molemmille tuotteille. Samasta taulusta mistä saatiin määrät löytyi myös tieto viallisista komponenteista. Nämä komponentit listattiin molemmille tuotteille, mutta laitteelle B ainoastaan tuotannon puolelta, koska sillä ei ole korttitestausta. Soluvika-taulusta saadaan myös eriteltyä erillinen tilasto eri vikatyypeille. Kun on tiedossa aloitetut tuotteet, korjaukseen menneet, vikatyypit ja komponentit, voidaan näille laskea kuvan 14 mukaiset prosenttiosuudet.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Tuote	Aloitettu	Korjaukse %			Kompone	Yht. Laitte %			Aikaväli		
2				3,12 %				0,0032 %				
3				3,86 %				0,0039 %				
4	Yht.			6,98 %								
5	Vikatyyppi	Määrä	%	% / Aloitettu		Kompone	Määrä	%	% Aloiteti	% Kategori	% Komponentti	
6	1		35,7 %	1,11 %		T16		17,9 %	0,56 %	0,051 %	0,0006 %	
7	3		31,0 %	0,97 %		D14		13,1 %	0,41 %	0,012 %	0,0004 %	
8	5		16,7 %	0,52 %		T3		7,1 %	0,22 %	0,020 %	0,0002 %	
9	11		9,5 %	0,30 %		L14		6,0 %	0,19 %	0,000 %	0,0002 %	
10	2		4,8 %	0,15 %		V27		4,8 %	0,15 %	0,004 %	0,0002 %	
11	4		2,4 %	0,07 %		L148		4,8 %	0,15 %	0,001 %	0,0002 %	
12						X31		3,6 %	0,11 %	0,005 %	0,0001 %	
13						D31		3,6 %	0,11 %	0,003 %	0,0001 %	
14						D16		3,6 %	0,11 %	0,003 %	0,0001 %	
15						T6		2,4 %	0,07 %	0,007 %	0,0001 %	
16						D36		2,4 %	0,07 %	0,002 %	0,0001 %	

Kuva 14. Prosenttiosuudet selvitetystä tiedoista.

Jotta tilastoihin saatiin laskettua jokaiselle komponenttityypille oma osuus, täytyi ensin selvittää, että kuinka paljon kyseistä komponenttia on laitteen piirilevyssä. Nämä tiedot löytyvät Telesten Agile PDM-tietokannasta. Agile PDM on Oraclen ohjelmisto, jolla hallinnoidaan tuotteen elinkaarta. Tiedot saa vietyä Agilesta suoraan Exceliin erillisenä tiedostona, josta saa suodatettua haluamansa tiedot. Samasta tiedostosta saadaan myös selville laitteen komponenttien ladontateknologia, eli onko komponentti käsiladottava vai koneladottava. Laitteelle A saatiin yhteensä 982 komponenttia, joista 49 on käsiladottavia. Laitteelle B puolestaan 885 yhteensä komponenttia, joista 30 on käsiladottavia. Lisäksi tilastosta saadaan eriteltyä komponenttikategoriat ja komponenttien lukumäärät yhdessä piirilevyssä. Näillä tiedoilla saatiin kuvan 15 mukaiset osuudet laskettua.

Ladontate	Kompone	Yht	Viallisia	%
THD				0,0266 %
SMD				0,0121 %
Kompone	Laitteessa	Yht.	Viallisia	%
C				0,0012 %
D				0,0435 %
F				0,0000 %
L				0,0074 %
M				0,0000 %
R				0,0002 %
T				0,4329 %
V				0,0090 %
X				0,0449 %

Kuva 15. Komponenttikategorian ja ladontateknologian osuudet.

Kuvan 15 sarakkeissa näkyy laitteessa oleva komponenttimäärä, yhteensä ladotut komponentit, vialliset komponentit sekä viallisten komponenttien osuus ladotuista komponenteista. THD:llä tarkoitetaan käsiladottuja komponentteja ja SMD:llä koneladottuja komponentteja.

Seuraavana tutkittiin löytyykö sama moduuli viallisena sekä moduulivalmistuksen että tuotannon puolelta. Tässä tapauksessa tärkeää on vielä tietää, että onko saman moduulin sama komponentti viallisena molemmilla puolilla. Kyselyssä käytettiin jälleen inner join-komentoa. Tuotannon puolella tuotteet kulkevat tietokannassa sarjanumerolla, kun moduulivalmistuksen puolella ne näkyvät moduulinumeroina. Moduulivalmistuksesta tullut moduulinumero säilyy edelleen tuotannon puolella olevan sarjanumeron tiedoissa. Duplikaattien eli kaksoiskappaleiden hakeminen onnistui hakemalla tuotannosta inner join-komennolla sarjanumeroista moduulinumerot. Näin saatiin selville vialliset moduulinumerot molemmilta puolilta. Nämä tiedot vietiin Exceliin ja suoritettiin seuraavanlainen lauseke:

```
=IF (ISERROR (MATCH (A2 ; $C$2 : $C$147 ; 0 ) ; "" ; TRUE)
```

Tällä lausekkeella saatiin vertailtua kahden eri sarakkeen tietoja toisiinsa. Jos vasemman puoleisen sarakkeen kentässä oleva tieto löytyi myös oikeanpuoleisesta sarakkeesta, tuli siihen tieto TRUE. Duplikaatteja löytyi yhteensä 5 kappaletta laitteesta A. Kuvan 16 tiedoista kuitenkin selvisi, että viat olivat erilaisia molemmilla puolilla. Samoja vikoja ei siis esiynny molemmilla puolilla samalla moduulilla.

	A	B	C	D	E	F	G
1	BA	MATCH	FA		MATCHIT	Syy BA	Syy FA
2						R99,R114,R115,R117 // vika 5 // 19: kaikki feilaa	T1 // vika 3
3						T16 // vika 1 // 15,16,19: vaste, gain ja slope feilaa	Testaus // vika 7 // ei vikaa
4						V27 // vika 5 // 24V jännite, V27 puuttui, rikkonut D18 ja D21	Testaus // 7
5						T16 // vika 3 // s.15,16,19 gainit alhaalla	T3 // vika 3
6						Y19 // vika 2 // s.7 ei vastetta	levy kärehtänyt
7							
8							
9							
10		TRUE					
11							
12							

Kuva 16. Duplikaattien haku ja syiden selvitys.

#### 4.4 Vikasyiden selvittely

Vikatilastoja muodostettaessa suoraan tietokannoista ei voitu täysin varmuudella sanoa vikojen todellisia syitä. Tietokannasta löytyvistä tiedoista vain moduulivalmistuksen puolelta tulleet vikakirjaukset sisälsivät selityksen. Tuotannon puolelta tulleiden vikakirjausten kohdalla suurin osa selityksistä olivat tyhjiä, kuten kuvasta 17 näkyy. Vikojen todellisten syiden selvittämiseksi täytyi käydä keskusteluja korjaajien kanssa.

Vaihe	Tarkenne	Steppi	Selitys	Komponentti	Vika	Havai
13	2		Liittimen jalka taittunut.	X4	11	2
13	2			L88	5	2
13	2		Diplexeri osunut.	L72	5	2
13	2			T3	3	2
13	2			C136	3	2
13	2			C123	3	2
13	2			L169	3	2
13	2			L36	3	2
13	2			T3	3	2
13	2			C265	3	2
13	2			L36	3	2
13	2			C199	3	2
13	2			L168	3	2
13	2			T3	3	2
13	2			X25	3	2
13	2			T3	3	2
13	2			T3	3	2
13	2			T3	3	2
13	2			C415	3	2
13	2			L136	4	2

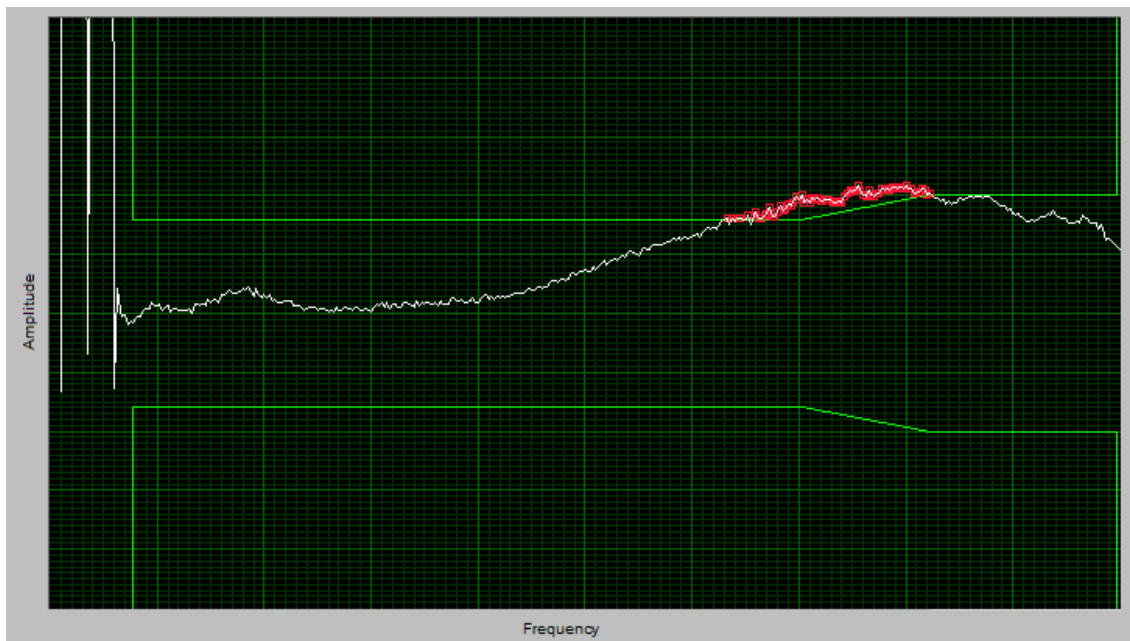
Kuva 17. Tyhjiä selityksiä tuotannon puolen vikakirjauksissa.

Tietokannasta saatiin kummallekin tuotteelle laskettua vikaantumisprosentti aloitettujen ja korjaukseen menneiden määrästä. Lisäksi tietokannasta saaduista vikatilastoista saadaan yleisimmät vikatyypit, vikatyypien tilastoista löytyvät vialliset komponentit ja edelleen viallisten komponenttien tiedoista löytyi niiden vikaselostukset. Tietokannoista löytyneet vikatilastot eivät kuitenkaan täysin heijastaneet todellisuutta. Vikojen todelliset syyt täytyi käydä selvittämässä korjaajilta. Keskusteluissa oli tärkeää selvittää, mistä viat johtuvat. Johtuvatko viat esimerkiksi komponenttien viallisuudesta vai mahdollisista prosessioista, kuten huonoista juotoksista. Lisäksi haluttiin tietää, miten suurimmat vianaiheuttajat jäävät kiinni mittauksissa. Yksittäisten ja satunnaisten vikojen täsmällisempi syyselvittely oli tarpeetonta. Pääpaino oli vikatilastoissa noin viiden ensimmäisen komponentin vikasyiden selvittäminen. Eri aikavälillä tutkittuja tilastoja vertailtiin myös ja vaikka niissä on pieniä eroja keskenään, ovat vikatyypit kuitenkin eri aikaväleillä samat. Komponenttien väliset prosenttiosuudet vaihtelevat eri aikaväleillä, mutta suurimpina vianaiheuttajina säilyvät aina samat vikatyypit.

## 5 TULOKSET JA POHDINTA

Vikatilastoista selvitettiin vikatyypit ja vianaiheuttajat. Laitteella A moduulivalmistuksen vikaprocentti oli 3,12% aloitetuista laitteista ja tuotannon puolella 3,86% tutkimusjaksolla. Yhteensä vikaprocentti oli siis 6,98%. Laitteen B vikaprocentti oli samalta aikaväliltä 11,16%. Laitteella B oli kyseisellä aikavälillä enemmän testausvikoja, mikä näkyy tilastossa rajumpana prosenttiosuutena. Yleisin viallinen komponenttityyppi on molemmissa laitteissa T-alkuinen komponentti. Näitä ovat muuntajat ja suuntakytkimet. Toinen merkittävä komponenttityyppi on D-alkuiset komponentit. Näitä ovat RF-vahvistimet, -vaimentimet sekä -kytkimet. Laitteen B tapauksessa nämä viat ovat vähentyneet ja ne ovat nykyään satunnaisia. Laitteen A tilastoissa kaksi RF-kytkintä ja RF-vaimenninta pitää tilaston kärkipaikkoja. Vikojen määrät ovat tippuneet paljon, koska vikojen havainnoimisessa on tapahtunut jatkuvasti parannusta. Eräs parannus on tarkempi juotostarkastus heti koneladonnan jälkeen. D-alkuiset komponentit jäävät kuitenkin kiinni moduulivalmistuksen puolella kordtitestauksessa. Näitä vikoja ei enää tuotannon puolella ilmene muuten kuin satunnaisesti.

Vikojen syyt ovat molemmissa laitteissa suurimpien vianaiheuttajien tapauksessa samanlaisia. Muuntajien ja suuntakytkimien viallinen toiminta näkyy mittauksissa sille asetettujen raja-arvojen eli maskin ylittymisenä. Laitteissa on virityskomponentteja, joilla saadaan säädettyä vasteita hieman jompaan kumpaan suuntaan, mutta yleensä laitteen vaste ylittää maskin juuri väärään suuntaan. Esimerkiksi säätökondensaattorilla saataisiin vastetta säädettyä ylemmäs, kun vaste ylittää maskin jo ennen säätämistä. Korjaajan toimenpide tällaisissa tapauksissa on vaihtaa komponentti, jolloin vaste saadaan asettumaan maskin sisälle. RF-vahvistimien, -vaimentimien sekä -kytkimien tapauksessa viallinen toiminta näkyy yleensä kalibrointiongelmana. Tämä ongelma aiheuttaa sen, että laitteen vahvistus ei riitä tai sitä on liikaa. Myös tällaisessa tapauksessa komponentin vaihtaminen korjaa ongelman. Kuvassa 18 näkyvässä mittaustuloksessa näkyy laitteen vaste epäonnistuneen virityksen jälkeen. Vaste on säädetty säätökondensaattorilla niin lähelle maskin rajaa kun mahdollista. Sen säätövara ei kuitenkaan riittänyt. Tämä laite lähetettiin korjaukseen, jossa komponentti vaihdettiin ja se korjasi ongelman.



Kuva 18. Solussa viritetyn laitteen mittaustulos, jossa vaste ylittää maskin.

Toisentyypisiä vikoja ovat tuotannon puolen testausviat ja puuttuvat komponentit. Testausviat johtuvat osaksi siitä, että solussa laitteita ei saada viritettyä kunnolla. Solussa työskenneltäessä ei voida yhteen laitteeseen tuhjata liian paljoa aikaa, vaan se on lähetettävä korjaukseen, jos sitä ei saada viritettyä. Tuotannon puolella puuttuvat komponentit ovat sellaisia, jotka lähtevät irti kokoonpanovaiheessa. Komponenttien sijoittelu lähelle ruuvipaikkaa tai dipleksereitä altistaa ne mahdolliselle osumalle. Komponentti voi ottaa osun ruuvinvääntimestä, kun piirilevyä ruuvataan valuu kiinni. Lisäksi diplekserin ohjurin reikien läheisyydessä olevat komponentit ovat alttiina osumille. Nämä viat johtuvat suunnitteluviasta. Moduulien kärähtämisestä tapahtuu myös ajoittain johtuen huolimattomuudesta. Moduulit kärähtävät, kun ne kytketään väärin.

Prosessivikoja ovat esimerkiksi komponenttien puuttuminen, väärin komponenttien latominen tai huonot juotokset. Prosessivikojen osuus vikatilastoissa on erittäin marginaalista. Kummankaan laitteen yleisimmät vikatyypit eivät ole prosessivikoja. T- ja D-alkuisten komponenttien viallisuus johtuu mahdollisesti niiden toleranssista. Testausviat sekä puuttuvat komponentit eivät ole myöskään prosessivikoja. Muiden yksittäisten vikojen syitä ei voida todeta, koska niitä on niin vähän ja ne ovat satunnaisia.

Seuraavana toimenpiteenä on lähteä tutkimaan mitä voidaan tehdä, ettei kyseisiä vikoja tulisi. Valmistajan vaihtamista voidaan miettiä komponenttien tapauksessa, joiden toleranssi aiheuttaa harmia. Voidaan tutkia millaisia vaihtoehtoja on komponentille



olemassa. Komponentin valmistajan vaihdossa täytyy luultavasti kuitenkin ottaa huomioon kustannukset ja muut suunnitteluun liittyvät asiat. Toinen asia mitä voidaan tehdä on lähteä tutkimaan tuotannosta syitä miksi viritys ei onnistu. Jostain syystä tuotannossa ilmenee paljon testausvikoja, mutta korjauspaikalla laite saadaan menemään testeistä läpi. Varsinkin laitteen B tapauksessa testausvikojen osuus on suuri. Komponenttien uudelleen sijoittelu on myös sellainen asia, joka voi vaikuttaa vikatilastoihin. Jos piirilevylle ladottujen komponenttien sijaintia saadaan muutettua vaarantamatta laitteen toimintaa, saadaan karsittua vikoja, jotka aiheutuvat mekaanisesti irtoavista komponenteista. Yksittäisille vioille ei voida tehdä mitään, sillä niiden syitä ei voida selvittää. Korttitestaus moduulivalmistuksen aikana lyhentää tuotannon aikaa hyvässä tapauksessa. Kun vika havaitaan ja korjataan jo moduulivalmistuksen aikana, saadaan se menemään tuotannosta läpi vaivattomasti. Tästä on esimerkkinä havaittu D-alkuinen komponentti laitteessa A. Se esiintyy moduulivalmistuksen puolella viallisenä, mutta loppukokoonpanossa kyseisen komponenttityypin osalta löytyy vain harvoin. Korttitestauksen huonona puolena on se, että sillä ei voida mitata kuin tiettyyn rajaan asti. Korttitestauksessa testipisteet alkavat reagoimaan muutoksiin herkemmin korkeammilla taajuuksilla ja pienikin epäpuhtaus voi aiheuttaa suuria heittoja mittaustuloksissa. Jos siis laite on viallinen juurikin korkeilla taajuuksilla, ei korttitestauksella sitä huomata, koska korttitestauksen mittauksien rajat on mitoitettu pienemmille taajuuksille. Tiedonhankinta vikatilastoihin oli vaikeaa tuotannon puolelta ja tämä asia on kerrottu eteenpäin. Tuotannon puolen vikakirjauksien selitykset ovat puutteellisia ja korjaajia tulisi kehottaa parempiin selityksiin, jotta jatkossa vikojen selvittely ja tilastointi helpottuisi.

## 6 LOPUKSI

Tämän opinnäytetyön aiheena oli selvittää Telestelle tyypillisimmät vianaiheuttajat ja niiden syyt. Tavoitteena oli tutkia millaisia vikoja löytyy moduulivalmistuksessa sekä tuotannon puolella loppukokoonpanosta. Korttitestauksen tarpeellisuus on vaikea määrittää suoraan tutkimalla vikatilastoja. Tilastoista selvisi laitteessa A oleva D-alkuinen komponentti, joka esiintyy viallisena ainoastaan moduulivalmistuksessa. Sen komponentin tapauksessa korttitestaus on hyödyllinen. Tässä työssä perehdyttiin SQL-kieleen ja sen käyttöön tietokannoissa sekä Telesten käytössä oleviin ohjelmistoihin. Tietokantaan talletetuista tiedoista saatiin muodostettua vikatilastot. Vikatilastoja vertailemalla korjaajien havaintoihin saatiin selvitettyä vikojen syitä. Suurimmaksi vianaiheuttajaksi selvisivät muuntajat ja suuntakytkimet. Lisäksi tilastoissa tuotannon testausviat vievät suuren prosenttiosuuden. Selvitetyllä vikatilastolla ja vikojen aiheuttajilla voidaan arvioida mitä on tehtävissä vikojen minimoimiseksi.

Opinnäytetyössä haasteita tuotti tiedonhankinta ja vikojen syiden täsmällinen selvittäminen. Moduulivalmistuspuolen korjaajat kirjaavat tarkemmin vikaselitykset tietokantaan, kun taas tuotannosta korjaajat jättävät suurimman osan tyhjäksi. Tämä hankaloittaa tilannetta siten, että esimerkiksi pitkältä aikaväliltä vikojen tutkiminen on vaikeaa, koska korjaajat ei välttämättä itsekään muista enää täysin, että miten viat ovat ilmenneet mittauksissa. Tuotannon puolelle on kehoitettu vikojen selitysten tarkentamista. Viallisten komponenttien syyt saatiin kuitenkin selville. Viallisimpien komponenttien syitä ovat niiden toleranssista johtuvat heittelyt, jotka näkyvät vasteiden maskien ylittymisinä tai alittumisina. Korjaustoimenpiteenä näissä tapauksissa on komponentin vaihtaminen, jolloin laitteen vaste pysyy maskin rajojen sisällä. Kun muut syyt saadaan selville, voidaan tutkia erilaisia vaihtoehtoja vikojen vähentämiseksi.

## LÄHTEET

- [1] Virtanen, M. 2007. Elektroniikkatuotannon apuaineiden ja korjausvälineiden käyttöönottoprosessi. Opinnäytetyö. Turun ammattikorkeakoulu.
- [2] Karvanen, E. 2016. SQL-tietokantojen valvonta. Insinöörityö. Metropolia Ammattikorkeakoulu.
- [3] Aalto, M. 2005. Development of a test fixture manufacturing process. Diplomityö. Åbo Akademi University.
- [4] Groff, J. & Weinberg, P. 2009. SQL: The Complete Reference. 3., uudistettu painos. New York: McGraw-Hill.
- [5] Hernandez, M. 2000. Tietokannat - suunnittelu ja toteutus. Suom. Kajala, T. Jyväskylä: Gummerus Kirjapaino Oy.
- [6] Stephens, R.; Plew, R.; Morgan, B. & Perkins, J. 1999. SQL-tietokantaohjelmointi – Trainer. Suom. Arola, J. Jyväskylä: Gummerus Kirjapaino Oy.
- [7] Microsoft, "Writing SQL Queries: Let's Start with the Basics", [www-dokumentti]. Saatavilla: [https://technet.microsoft.com/en-us/library/bb264565\(v=sql.90\).aspx](https://technet.microsoft.com/en-us/library/bb264565(v=sql.90).aspx) (Luettu 26.4.2017).
- [8] National Instruments. "What Is TestStand?". [www-dokumentti]. Saatavilla: <http://www.ni.com/teststand/whatis/> (Luettu 26.4.2017).
- [9] National Instruments, "Database Viewer Application". [www-dokumentti]. Saatavilla: <http://zone.ni.com/reference/en-XX/help/370052H-01/tsref/infotopics/dbviewer/> (Luettu 26.4.2017).
- [10] Holmberg, A. 2014. BADS-ohjelmiston päivitys ja käyttöönotto. Opinnäytetyö. Turun ammattikorkeakoulu