

# Kirjanpitoerien avoimien erien tasaus- ja raportointiprosessin automatisointi Excel-makrolla

Vili Virtanen

OPINNÄYTETYÖ  
Marraskuu 2023

Liiketalouden tutkinto-ohjelma  
Taloushallinto

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Liiketalouden tutkinto-ohjelma  
Taloushallinto

VIRTANEN, VILI:

Kirjanpitoilien avoimien erien tasaus- ja raportointiprosessin automatisointi Excel-makrolla

Opinnäytetyö 52 sivua, joista liitteitä 2 sivua  
Marraskuu 2023

---

Opinnäytetyön tavoite oli automatisoida Yritys X:n Tavarantoimitusten vastaanotto / laskun vastaanotto -tilien tasaus- ja raportointiprosessi Excel-makroa hyödyntäen. Kyseinen prosessi suoritetaan kerran kuukaudessa manuaalisesti, ja siihen kuluu aikaa noin 2–3 tuntia. Automatisointi mahdollistaa prosessin tehokkaamman suorittamisen ja vapauttaa suorittajan muihin tehtäviin. Tavoiteajaksi prosessin suorittamiseen asetettiin Yritys X:n kanssa 45 minuuttia. Yritys X on suomalaisen kaupan alan yrityksen alaisuudessa operoiva taloushallinto-, palkanlaskenta- ja HR-palveluita tarjoava yksikkö. Tekijä oli opinnäytetyöprosessin aikana työsuhteessa toimeksiantajayritykseen, mutta makroa kehitettiin vapaa-ajalla.

Opinnäytetyössä kehitettiin makroa yhdeksän viikon ajan viikolta 37 viikolle 45. Makro valmistui tämän työn aikana kokonaisuudessaan. Se ajetaan osissa, jotka on jaettu prosessin eri vaiheiden mukaisesti. Osia on yhteensä 10. Makro tekee kaikki prosessiin kuuluvat työvaiheet itse tasauskirjauksia ja sähköpostien lähetyksestä lukuun ottamatta. Prosessin suorittajalle jää itselleen makron luomien luetteloiden ja sähköpostien oikeellisuuden tarkistus ja kirjausten tekeminen. Prosessi suoritettiin kokonaisuudessaan makroa käyttäen marraskuussa 2023 ja se todettiin toimivaksi. Jatkossa prosessia ei tarvitse siis enää suorittaa manuaalisesti. Makron ajoon kuluu aikaa noin kahdeksan minuuttia, joten ajajalle jää tarkistustyöhön ja kirjausten tekoon aikaa 37 minuuttia, jolloin ollaan vielä tavoitteissa.

Prosessin tehostamisen lisäksi tässä työssä luotua makroa voidaan mahdollisesti hyödyntää jatkossa toimeksiantajayrityksen muiden prosessien automatisoinnissa. Lisäksi tekijän oma osaaminen VBA-ohjelmoinnista ja Excel-makroista kasvoi huomattavasti. Myös opinnäytetyön julkisesta osuudesta voi olla hyötyä muille makrojen kehittämisessä ja luomisessa.

---

Asiasanat: makrot, Excel, automaatio, Visual Basic for Applications, prosessit, kirjanpito

## ABSTRACT

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Degree Programme in Business Administration  
Financial Administration

VIRTANEN, VILI:

Automation of the Process of Balancing and Reporting Open Items in Accounting with an Excel Macro

Bachelor's thesis 52 pages, appendices 2 pages  
November 2023

---

The objective of the thesis was to automate the balancing and reporting process of Company X's Goods Receipt/Invoice Receipt accounts using an Excel macro. The process is performed monthly and manually, and it takes approximately 2-3 hours. Automation enables a more efficient execution of the process, freeing up the operator for other tasks. The goal was to reduce the time required for the process to 45 minutes and it was determined in collaboration with Company X. Company X operates as a finance, payroll, and HR services unit under a Finnish retail company. During the thesis process, the author was employed by the client company but developed the macro in their own free time.

The macro development took place over nine weeks from week 37 to week 45, and it was completed entirely during this thesis. The macro is executed in parts, divided according to the different phases of the process, with a total of 10 parts. The macro performs all the steps of the process, except for balancing entries and email sending. The operator is left with the verification of lists and emails created by the macro and making necessary entries. The entire process was carried out using the macro in November 2023, and it was found to be functional. In the future, the process will no longer need to be performed manually. The macro execution takes about eight minutes, leaving the operator with 37 minutes for verification and entry, thus meeting the goal.

In addition to streamlining the process, the macro created in this thesis could potentially be utilized for automating other processes within the client company. Furthermore, the author's proficiency in VBA programming and Excel macros significantly increased. The publicly available portion of the thesis may also be beneficial for others in the development and creation of macros.

---

Key words: macros, Excel, automation, Visual Basic for Applications, processes, accounting

## SISÄLLYS

1	JOHDANTO .....	6
2	OPINNÄYTETYÖN TAVOITE JA TARKOITUS .....	7
3	OPINNÄYTETYÖN KUVAUS .....	8
	3.1 Talousprosessin tarkoitus .....	8
	3.2 Talousprosessin kuvaus.....	8
	3.3 Kehitettävä Excel-makro .....	11
4	EXCEL-MAKRO.....	13
5	PÄIVÄKIRJARAPORTOINTI.....	15
	5.1 Viikko 1 .....	15
	5.2 Viikko 2 .....	19
	5.3 Viikko 3 .....	22
	5.4 Viikko 4 .....	25
	5.5 Viikko 5 .....	29
	5.6 Viikko 6 .....	31
	5.7 Viikko 7 .....	32
	5.8 Viikko 8 .....	36
	5.9 Viikko 9 .....	42
	5.10 Yhteenveto koko tarkastelujaksolta .....	46
6	VALMIIN MAKRON KUVAUS .....	47
7	POHDINTA .....	49
	LÄHTEET.....	50
	LIITTEET .....	51
	Liite 1. VBA-editori Excelissä .....	51
	Liite 2. Excel-makron etusivu.....	52

**LYHENTEET JA TERMIT**

Funktio	Kaava, joka suorittaa tietyn laskennan tai toiminnon Excel-taulukossa
Koodi	Ohjelmaan kirjoitetut ohjeet tai komennot, jotka makro suorittaa
Kumppanikoodi	Käytetään yksilöimään ja erottamaan eri liiketoimintakumppanit toisistaan
Loop	Silmukka, joka sallii tietyn koodinpätkän suorittamisen useita kertoja, kunnes ehdot täyttyvät
Moduuli	VBA-ohjelmoinnissa moduulit ovat paikkoja, jonne VBA-koodi voidaan kirjoittaa ja järjestellä.
Muuttuja	Muuttujia käytetään hallitsemaan ja tallentamaan tietoa koodissa.
Objekti	Koodiesine, joka viittaa johonkin sovelluksen osaan, kuten soluun, taulukkoon tai käyttöliittymän elementtiin
Range	Objekti, joka viittaa solujen alueeseen Excel-taulukossa
SAP	Toiminnanohjausjärjestelmä
SAP-moduuli	Toiminnallinen osa SAP-järjestelmässä, joka kattaa tietyn liiketoiminta-alueen toiminnot
SAP Script	Ohjelmointikieli, jota voi käyttää yhdessä VBA:n kanssa SAP-järjestelmän raporttien ja lomakkeiden muotoiluun ja tulostukseen.
String	Muuttuja, joka tallentaa ja käsittelee merkkijonoja eli tekstidataa
Sub-proseduuri	Koodinosa, joka on määritelty suorittamaan tietty toiminto tai tehtävä
Tasauskirjaus	Vastakirjaus, jossa tapahtuman saldo menee tilillä nolille
Transaktio	Tietty liiketoiminnallinen tai tietokantatoiminto SAP-järjestelmässä
TV/LV-tilit	Tavaran vastaanotto / laskun vastaanotto -tilit
VBA	Visual Basic for Applications -ohjelmointikieli

## 1 JOHDANTO

Tässä opinnäytetyössä dokumentoidaan Yritys X:n tavaran vastaanotto / laskun vastaanotto -tilien tasaus- ja raportointiprosessin automatisointia Excel-makrolla. Opinnäytetyö on päiväkirjamuotoinen, ja sen tarkastelujakso on yhdeksän viikon pituinen viikkojen 37–45 ajalta. Automatisoitavassa prosessissa käsitellään asiakasyrityksen TV/LV- eli tavaran vastaanotto / laskun vastaanotto -tilejä ja tehdään yli kuusi kuukautta vanhoille ja -1000 €-300 € suuruisille avoimille ostotilauksille tasauskirjaukset. Tasauksella tarkoitetaan tässä tapauksessa kirjausta, jossa ostotilauksen saldo merkitään tilin toiselle puolelle, minkä jälkeen tilin molempien debet/kredit -puolien saldot ovat sen osalta yhtä suuret. Prosessi tehdään kerran kuukaudessa manuaalisesti, ja siihen kuluu noin 2–3 tuntia aikaa. Excel-makroa kehittäessäni raportoin työprosessia päivätasolla ja jokaisen viikon osalta olen koonnut yhteenvedon, jossa on makron kehityksen tärkeimmät vaiheet.

Yritys X:llä viitataan työssä suomalaisen kaupan alan yrityksen alaisuudessa operoivaan taloushallinto-, palkanlaskenta- ja HR-palveluita tarjoavaan yksikköön. Yritys X on työnantajayritykseni ja olen ollut siellä työsuhteessa vuoden 2022 keväästä lähtien. Prosessi, jota työssä käsitellään, on aikaisemmin kuulunut työtehtäviini, ja tuolloin sain tämän kehitysidean ja päätin toteuttaa sen opinnäytetyössäni.

Tämän työn julkisesta osuudesta salataan toimeksiantajayritys ja muita tunnistamiseen johtavia tekijöitä. Lisäksi makron toiminnan yleiskuva ja pätkiä sen koodista näytetään tässä raportissa, mutta kokonaisuutena sitä ei julkisteta.

## 2 OPINNÄYTETYÖN TAVOITE JA TARKOITUS

Tässä opinnäytetyössä käsitellään Yritys X:n kerran kuukaudessa tehtävää talousprosessia. Prosessi tehdään manuaalisesti ja se vie aikaa noin 2–3 tuntia. Kyseisessä prosessissa haetaan SAP-toiminnanohjausjärjestelmästä lista TV/LV-tilien avoimista eristä. Tämä listaus viedään Exceliin ja muokataan muotoon, jonka perusteella tehdään tasauskirjaukset ostotilauksille, joilla ei ole ollut tapahtumia yli kuuteen kuukauteen, ja joiden summat ovat joko alle 1000 € velan puolella tai alle 300 € saamisen puolella. Lopullinen tuotos avoimista eristä, tasauskirjauksista sekä niiden tulos- ja varastovaikutuksesta raportoidaan sähköpostitse yhtiön vastuuhenkilöille.

Prosessissa joutuu tekemään paljon manuaalista työtä Excelissä. SAP:ista haettua lähtötietoa täytyy muokata esimerkiksi suodattamalla vanhat ja uudet tapahtumat, laskemalla ostotilauksien summat ja suodattamalla liian suuret/pienet summat pois kirjattavien joukosta, poistamalla toimittajia, joiden tapahtumia ei kirjata ja poistamalla tiettyjä rahteja. Kirjauksien jälkeen tilien avoimet erät haetaan uudelleen ja lajitellaan ne vuosien perusteella uusimman tapahtuman mukaan. Lisäksi haetaan vielä uusi listaus SAP:ista kirjauksista syntyneiden tositenumeroiden avulla ja muodostetaan sen perusteella taulukko, jonka avulla lasketaan ja raportoidaan tulos- ja varastovaikutus.

Tavoitteena on tehostaa prosessin tekemistä automatisoimalla se lähes kokonaisuudessaan käyttäen Excel-makroa ja vapauttaa näin tekijä nopeammin muihin tehtäviin. Opinnäytetyön tarkoituksena on kehittää Excel-makro, joka tekee kaikki prosessin vaiheet lähtötietojen hakemisesta SAP-järjestelmästä, lopullisten sähköpostien luomiseen asti lukuun ottamatta itse tasauskirjauksia. Makron ajajalle jäisi vain makron laatimien listauksien oikeellisuuden tarkistus ja niiden perusteella tehtävät kirjaukset sekä myöhemmässä vaiheessa sähköpostien ja liitetietojen tarkistus ja lähetys. Tavoitteena on, että koko prosessin pystyisi suorittamaan noin 45 minuutissa mukaan lukien kirjausten tekemisen. Tähän on laskettu mukaan aika, joka saattaa kulua kirjausten tarkistuksessa ja tekemisessä.

### 3 OPINNÄYTETYÖN KUVAUS

#### 3.1 Talousprosessin tarkoitus

Opinnäytetyössä käsiteltävänä oleva prosessi on osa vaihto-omaisuuskirjanpidon suurempaa kokonaisuutta. Siinä käsitellään asiakasyrityksen Tavarantoimitus / Laskun vastaanotto -tilejä. Näille tileille tulevat debet-kirjaukset vastaanotetuista laskuista ja kredit-kirjaukset vastaanotetuista tavaroista. Mikäli ostotilauksen osalta on tavarat ja lasku vastaanotettu pitäisi sen mennä tileillä nolliille. Tilin avoimet tapahtumat ovat ostotilauksia, joista on joko vastaanotettu lasku, mutta ei tavaroita tai vain osa tavaroista, tai vastaanotettu tavarat, mutta ei laskua. Prosessi tehdään jokaisen kuukauden 12.–14. päivänä ja siinä on monia eri vaiheita ja se tehdään seuraamalla Yritys X:n laatimia ohjeita (Organisaation makro-ohjeet 2023). Prosessin tarkoituksena on tasata nolliille tileiltä pienet ja vanhat avoinna olevat ostotilaukset, joiden osalta ei ole kannattavaa lähteä selvittämään puuttuvaa laskua tai puuttuvia tavaroita, ja raportoida jäljelle jääneet arvoltaan suuremmat avoimet ostotilaukset, joiden osalta selvitystä on tehtävä. Siinä haetaan useita eri lähtötietoja SAP-järjestelmästä, tehdään pienille ja vanhoille tapahtumille tasauskirjaukset ja luodaan kaksi lopullista Excel-taulukkoa sisältäen tiedot tasatuista ostotilauksista ja jäljelle jääneistä avoimista ostotilauksista tileillä. Lopuksi lähetetään kolme eri sähköpostia raportoiden kirjausten jälkeen tileille avoimeksi jääneet tapahtumat ja kirjauksista syntynyt tulos- ja varasto vaikutus. Manuaalisesti prosessi on monimutkainen ja aikaa vievä.

#### 3.2 Talousprosessin kuvaus

TV/LV-tilien tasaus- ja raportointiprosessi suoritetaan kuukausittain. TV/LV-tileiltä tasataan vanhat ja pienet ostotilaukset. Vanhoilla ja pienillä tarkoitetaan ostotilauksia, joiden avoinna oleva summa on alle 1000 € velan puolella tai alle 300 € saamisen puolella ja niillä ei ole viimeisen kuuden kuukauden aikana ollut tapahtumia.



Prosessissa ajetaan ensin automaattinen kuittausajo, jonka jälkeen TV/LV-tilien avoimet tapahtumat haetaan kuluvaan päivään asti SAP-järjestelmästä ja vietään Exceliin, josta muokataan lopullinen tasattavat ostotilaukset sisältävä Excel. Viedystä Excel-luettelosta korjataan lukujen muodot ja poistetaan ylimääräiset rivit ja sarakkeet, minkä jälkeen aineistosta suodatetaan yli ja alle kuusi kuukautta vanhat tapahtumat ja vietään nämä omille välilehdilleen. Sitten yli kuusi kuukautta vanhoille tapahtumille tehdään PHAKU-funktio, jolla haetaan ostotilausnumeroiden avulla vastaavia alle 6 kk -välilehdeltä. Kaikki ostotilaukset, joiden tuorein tapahtuma löytyi funktion avulla alle 6 kk -välilehdeltä, poistetaan yli 6 kk -välilehdeltä. Kun yli 6 kk -välilehdellä on jäljellä vain ostotilaukset, joiden tuorein tapahtuma on yli kuusi kuukautta vanha, tehdään riveille Summa.Jos.Joukko -funktio, jolla lasketaan yhteen kaikkien saman ostotilausnumeron sisältävien rivien summa-sarakkeella olevat arvot. Luotu summasarake suodatetaan niin, että niiden arvot ovat välillä -1000 €-300 € ja nämä rivit vietään omalle välilehdelleen "Tasattavat".

Seuraavaksi tasattavien ostotilauksien joukosta poistetaan ne, joiden toimittaja on listattuna Yritys X:n laatimaan Exceliin "\*\*\*\*\* NÄITÄ EI SAA TASATA (toimittajat)". Ensin haetaan SAP:ista ostotilausnumeroilla toimittajanumerot ja vietään luettelo Exceliin. Viedylle Excelille tehdään PHAKU-funktio, jolla etsitään vastaavia toimittajainumeroita toisesta Excelistä, jossa on listattuna nämä toimittajat, joita ei tasata. Suodatetaan vain PHAKU-funktiolla löytyneet toimittajanumerot näkyviin ja tehdään vielä tasattaville ostotilauksille PHAKU, jolla tuodaan nämä ei-tasattavien toimittajien tiedot oikean ostotilauksen perään taulukkoon. Ostotilaukset, joiden toimittajat olivat ei-tasattavien joukossa, vietään omalle välilehdelleen. Tämän jälkeen tasattavien ostotilausten numeroilla haetaan SAP-järjestelmästä listaus, josta suodatetaan tietyt rahtirivit näkyviin ja vietään Exceliin. Viedystä Excelistä suodatetaan näkyviin nämä rahdit sillä SAP:issa tehty suodatus jättää vielä muitakin rivejä näkyviin. Näille suoritetaan vielä sama prosessi kuin ei-tasattaville toimittajille ja vietään omalle välilehdelleen. Tämän jälkeen tasattavat-välilehdellä ovat vain ne ostotilaukset, joille tasauskirjaukset tehdään. Seuraavaksi vietään ostotilausnumerot SAP-transaktioon, jossa kirjaukset tehdään. Tarkistetaan vielä, että summat täsmäävät Excelin ja SAP:in listausten välillä, minkä jälkeen tasataan ensin kumppanikoodilliset tapahtumat ja sitten kump-

panikoodittomat ja viedään kirjausten tositenumerot Excelliin tasattujen ostotilausten perään. Kumppanikoodilliset tapahtumat tasataan erikseen sillä, jos ne tasattaisiin samanaikaisesti, SAP lisäisi kumppanikoodit myös kumppanikoodittomien tapahtumien tasauskirjauksiin.

Tasausten jälkeen ajetaan automaattinen kuittausajo vielä uudelleen. Kuittausajon jälkeen haetaan SAP:ista tasauskirjausten tositenumeroilla luettelo, jonka avulla lasketaan tasausten tulos- ja varastovaikutus. Haettu luettelo viedään Excelliin. Tämän jälkeen luettelo siirretään uudelle välilehdelle tasattavat ostotilaukset sisältävään Excelliin ja tehdään riveille JOS-funktio, jonka avulla kredit-kirjausten summan etumerkki muutetaan -merkkiseksi. Tämän jälkeen suodatetaan näkyviin rivit, jotka ovat tilillä, jolta tulosvaikutuksen summa lasketaan, ja viedään tämä summa omalle välilehdelle. Seuraavaksi suodatetaan varastotilin rivit näkyviin ja viedään tämä summa samalle välilehdelle. Tämän jälkeen lasketaan vielä tasattavat-välilehdeltä summa velan puolella olevista tasauksista laskemalla – merkkiset ostotilaukset yhteen ja viedään välilehdelle summa tasauksista velan puolella. Samoin lasketaan + merkkiset ja viedään välilehdelle summa tasauksista saamisen puolella. Sitten lasketaan tulos- ja varastovaikutus yhteen ja kaikkien tasauksien summat yhteen. Summien pitäisi täsmätä.

Seuraavaksi haetaan avointen erien listaus TV/LV-tileiltä uudelleen ja viedään Excelliin. Tässä Excelissä lajitellaan ostotilaukset omille välilehdilleen vuosittain. Näille välilehdille tehdään samat PHAKU-funktiot kuin aikaisemmin yli 6 kk välilehdelle ja poistetaan ne ostotilaukset käsiteltävän vuoden välilehdeltä, jolla on tapahtumia uudemmilla vuosilla. Kun ostotilaukset on järjestetty vuosittain omille välilehdilleen tuoreimman tapahtuman mukaisesti, haetaan SAP:ista ostotilausnumeroilla vuosi kerrallaan luettelot samasta transaktiosta, jossa tasauskirjaukset tehtiin ja kootaan nämä yhteen Excelliin. Seuraavaksi muodostetaan otsikkorivi, lajitellaan taulukot ostotositteen mukaan ja tehdään ostotositteille välisumma, sekä lihavoidaan ja kiinnitetään otsikkorivi.

Tämän jälkeen valmis taulukko lähetetään sähköpostitse eteenpäin. Yhteen sähköpostiin laitetaan viestiksi tasausten summat ja niistä syntyneet tulos- ja varas-

tovaikutus. Seuraaviin viesteihin taulukon ensimmäiseltä välilehdeltä suodetaan tiettyjen tulosityksiköiden rivit ja lähetetään eteenpäin. Tämän jälkeen prosessi on tehty.

### 3.3 Kehitettävä Excel-makro

Opinnäytetyössä luodaan Excel-makro, joka hakisi avoimien erien luettelon SAP:ista Excel muodossa ja toisi tiedot luettelosta makrotyökirjaan ja muodostaisi välilehdet, joissa on kaikki avoimet tapahtumat, alle kuusi kuukautta vanhat ostotilaukset, yli kuusi kuukautta vanhat ja tasattavat. Tämän jälkeen makro hakisi tasattavien ostotilausten numeroilla toimittajien tiedot SAP:ista, veisi tiedot makrotyökirjaan ja siirtäisi pois tasattavien joukosta ne ostotilaukset, joiden toimittajat ovat listattuna erillisessä Excelissä ei-tasattavien toimittajien joukkoon. Seuraavaksi se hakisi SAP:ista tasattavien ostotilausnumeroiden avulla listauksen, josta suodattaisi tietyt rahtikirjaukset näkyviin. Myös nämä se poistaisi tasattavien joukosta ja siirtäisi omalle välilehdelleen makrotyökirjassa.

Tässä vaiheessa tasattavat-välilehdellä olisi vain ne ostotilaukset, joille tehtäisiin tasauskirjaukset. Tätä vaihetta makrolla ei automatisoida, jotta prosessin tekijä voi varmistaa kirjausten oikeellisuuden. Kun kirjaukset on tehty, makron ajaja merkitsisi niistä syntyneet tositenumerot makrotyökirjan tasattavat-välilehdelle. Tämän jälkeen makro hakisi tasausten kirjanpitositteiden avulla SAP:ista listauksen, jossa olisi kirjausten tositenumerot, debet/kredit -tunnus, summa ja pääkirjatili. Seuraavaksi makro veisi tämän listauksen omalle välilehdelleen makrotyökirjaan ja loisi uuden välilehden, jolle se laskisi kirjauksista syntyneet tulos- ja varastovaikutuksen haetun listauksen avulla tulos- ja varastotilien loppusummasta. Sitten samalle välilehdelle se laskisi velan ja saamisen puolella olevat tasaukset tasattavat-välilehdeltä, jonka jälkeen vielä tulos- ja varastovaikutukset yhteensä ja tasausten summat yhteensä. Tämän jälkeen makro tallentaisi kaikki luodut välilehdet lukuun ottamatta makro-välilehteä eri Exceliin .xlsx muodossa poistaen makrotoiminnot. Tämä olisi ensimmäinen valmis prosessissa luotava Excel, joka sisältäisi kaikki tarpeelliset tasauksiin liittyvät tiedot.

Seuraavaksi makro lähtisi luomaan toista lopullista Exceliä. Se hakisi TV/LV-tilien avoimet tapahtumat uudelleen tehtyjen kirjausten jälkeen ja veisi ne omaan Exceliin, jolla raportoidaan tilien avoimet erät. Tämän jälkeen se jakaisi ostotilaukset vuosittain omille välilehdilleen sen mukaan minä vuonna tuorein tapahtuma on. Sitten se hakisi vuosi kerrallaan SAP:ista eri transaktiosta, kuin avoimet erät haettiin, listaukset ja veisi ne avoimien erien Exceliin. Välilehdille se muodostaisi otsikkorivit, lajittelisi ja lisäisi välisummat ostotilauksien mukaan. Kun avoimien erien Excel olisi valmis, makro loisi sähköpostit ajajan luonnoksiin, joista hän voisi tarkistaa niiden oikeellisuuden ja lähettää ne eteenpäin.

## 4 EXCEL-MAKRO

Excel-makrot ovat automatisointityökaluja, jotka perustuvat Visual Basic for Applications (VBA) -ohjelmointikieleen ja ovat käytettävissä Microsoft Excelissä. Ne mahdollistavat monimutkaisten tehtävien ja toimintojen automatisoinnin, mikä säästää aikaa ja parantaa tehokkuutta. Excel-makrot ovat erityisen hyödyllisiä toistuvien tehtävien suorittamisessa ja monimutkaisten laskentakaavojen tai raporttien laatimisessa. VBA:n avulla makrot suorittavat toimintosarjoja eli prosedureja. Niillä tarkoitetaan ohjelmakokonaisuuksia, jotka suoritetaan järjestyksessä koodin alusta loppuun. Tässä työssä käytetään Sub-proseduuria, joka alkaa aina Sub-riviltä ja loppuu End Sub -riviin. Tällöin makro toistaa kaikki toiminnot koodista niiden rivien välillä. (Makrosta Office-ohj... n.d.)

VBA on sulautettu Exceliin ja mahdollistaa makrojen kirjoittamisen ja suorittamisen Excelin VBA-editorissa (Liite 1). VBA:ssa voit käyttää muuttujia, ehtolauseita, silmukoita ja muita ohjelmointirakenteita. Opinnäytetyön päiväkirjaraporttiosion sivuilla 14–15 on kerrottu, miten VBA otetaan käyttöön Excelissä, jonka jälkeen makron voi tallentaa Excelissä käyttämällä ”Tallenna makro” -toimintoa. Kun tallennus on aloitettu, Excel tallentaa kaikki toimet, joita sen aikana tehdään, ja muuntaa ne VBA-koodiksi. Kun makro on tallennettu, sen voi suorittaa Excelin ”Makrot” -valikosta tai määrittelemällä sille painikkeen tai pikanäppäimen, josta makro suoritetaan. Suorittaminen toistaa kaikki tallennetut toimet. Tallennettua makroa voi myös muokata avaamalla VBA-editori Excelissä. Sen saa auki ”Kehitystyökalut” -valikosta painamalla ”Näytä koodi” -painiketta. Siellä voit lisätä, poistaa tai muokata VBA-komentoja.

Excel-makroja voidaan käyttää moniin eri tarkoituksiin. Näitä ovat esimerkiksi tietojen käsittely, raporttien luominen, grafiikat ja kaaviot, toistuvat tehtävät ja kustomoidut käyttöliittymät. On tärkeää kuitenkin huomata, että vaikka Excel-makrot ovat erittäin hyödyllisiä, niiden käyttöön liittyy myös turvallisuusnäkökohtia. Makroja ei tulisi käyttää epäluotettavista lähteistä peräisin olevissa tiedostoissa, sillä ne voivat sisältää haitallista koodia. Excel-makroilla voi tehdä arjen työtehtävistä

paljon helpompia ja nopeampia. Kokeneet käyttäjät voivat luoda hyvin monimutkaisia ja tehokkaita makroja, mutta myös aloittelijat voivat hyödyntää makroja pienempien ja yksinkertaisempien tehtävien automatisointiin.

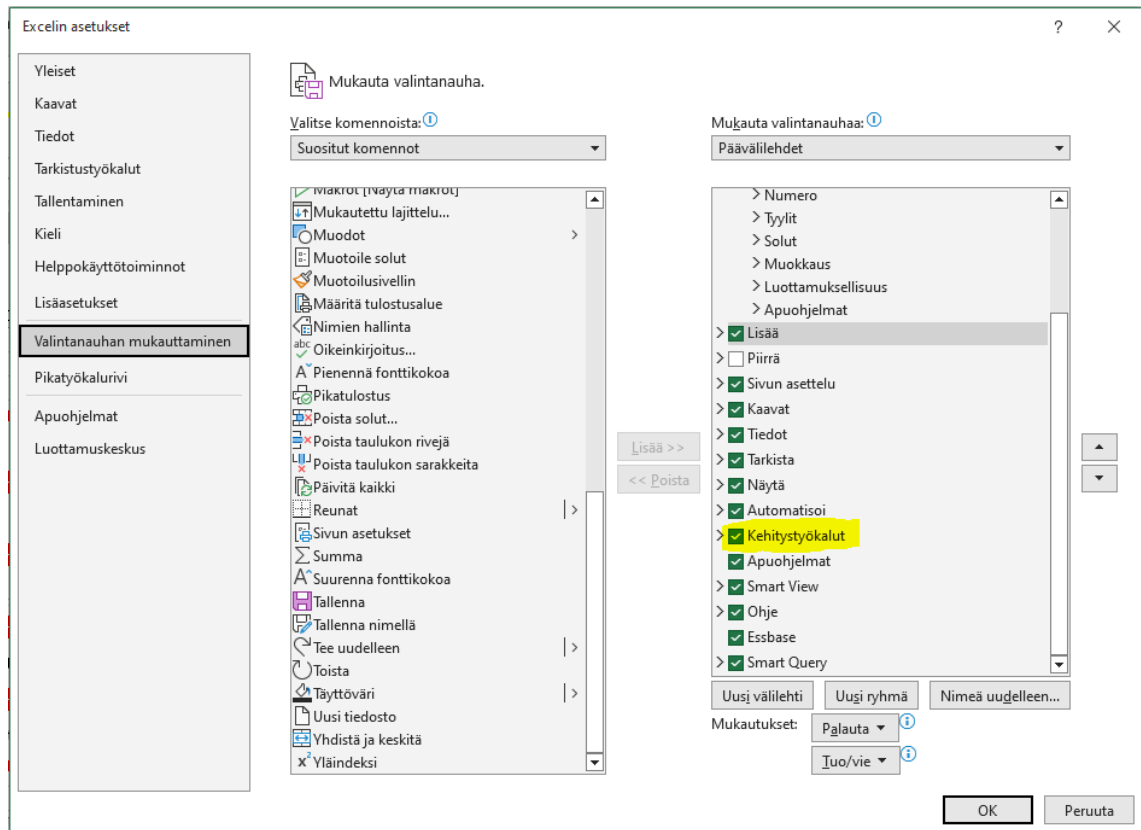
## 5 PÄIVÄKIRJARAPORTOINTI

### 5.1 Viikko 1

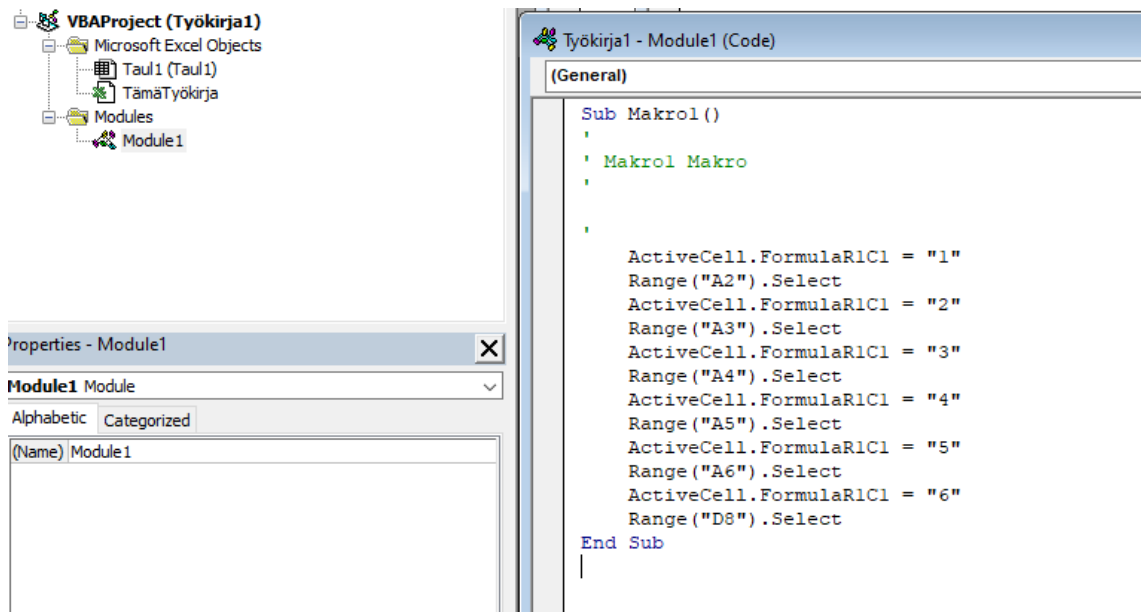
#### Tiistai 12.9.

Aloitin työni silmäilemällä Yritys X:n aineistoa Excel-makroista. Aineistossa oli myös linkki Aki Taanilan dokumenttiin Excel VBA-ohjelmoinnista, jossa on paljon hyödyllistä tietoa makron kehitystä varten (Taanila 2013). Yritys X:n aineistossa on viisi makron ohjeistusvideota, joita ryhdyin seuraavaksi katsomaan läpi. Videoiden aiheet olivat makron nauhoitus, makron koodaus ja dokumentointi, linkkejä oppaisiin. (Organisaation makro-ohjeet 2023.)

Excelissä täytyy olla kehitystyökalut käytössä, jotta makroja voi luoda. Polku niiden käyttöönottoon on Tiedosto – Asetukset – Valintanauhan mukauttaminen ja avoimesta ikkunasta merkitsee rastin ruutuun ”Kehitystyökalut” (Kuva 1). Tämän jälkeen Kehitystyökalut-osio tulee Excelin yläpalkkiin. Painoin kehitystyökalut-osioista ”Tallenna Makro” -painiketta, joka aloittaa nauhoituksen, sen jälkeen kaikki Excelissä tehdyt toiminnot ja muokkaukset nauhoitetaan VBA-editoriin koodinpätkäksi, kunnes nauhoitus lopetetaan. Lisäsin eri soluihin eri arvoja ja lopetin nauhoituksen. Nauhoituksen lopetettua avasin VBA-editorin painamalla ”Kehitystyökalut” välilehdeltä ”Näytä koodi”. Siellä nauhoitus on luonut moduulin, jonka sisältä koodinpätkä löytyy (Kuva 2).



Kuva 1. Polku kehitystyökalujen käyttöönottoon Excelissä



Kuva 2. Yksinkertainen nauhoitettu koodi, jossa makro syöttää luvut 1–6 yksi kerrallaan ensiksi aktiiviseen soluun, josta etenee A2-A6.

Pelkkä nauhoitus ei kuitenkaan ole hyvä tapa luoda makroja, sillä usein se saattaa suorittaa turhia ja ylimääräisiä toimintoja, eikä sopeudu muuttuviin lähtötietoihin monimutkaisempia tehtäviä tehdessä ilman muuttujien määrittelyä. Se on

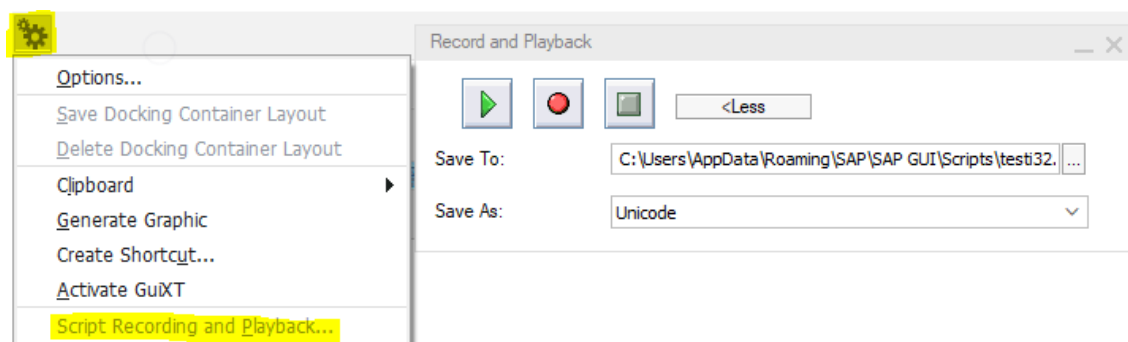


kuitenkin hyvä tapa katsoa mitä koodiin tulee kirjoittaa tietyissä toiminnoissa. Esimerkiksi nauhoituksen voi laittaa päälle, valita solu A2 ja lopettaa nauhoituksen. Tämän jälkeen voi avata koodin VBA-editorista ja katsoa minkälaisella koodilla makro valitsee solun A2. Kuvasta 1 näkee, että kyseinen koodinpätkä on: "Range ("A2").Select".

### **Torstai 14.9.**

Tämän työn aiheena olevassa talousprosessissa ei pelkästään käytetä Exceliä, vaan lähtötietoja haetaan SAP-järjestelmästä. VBA-ohjelmoinnilla pystyy käyttämään koodia, jolla makro ottaa yhteyden SAP:iin ja tekee toimintoja siellä. Näitä toimintoja voi lisätä makroon nauhoittamalla SAP:issa SAP Scriptiä ja lisäämällä Scriptin nauhoituksesta syntyneen koodin VBA-editoriin Excel-makrossa.

Tässä opinnäytetyössä kehitettävän makron ensimmäisenä tehtävänä olisi hakea yhtiön TV/LV-tilien avoimet erät kuluvaan päivään asti SAP:ista ja tallentaa luettelo Excel-muodossa samaan kansioon missä makrotäykirja sijaitsee. Lähdin nauhoittamaan SAP Scriptiä, jonka avulla makron pitäisi pystyä suorittamaan tämä vaihe. SAP:issa polku nauhoittamiseen on seuraava: Mutterin kuva tai Alt + F12 – Script Recording and Playback..., jonka jälkeen avautuu ruutu, jossa on painikkeet "Playback Script", "Record Script", "Stop" ja "More" (Kuva 3). Painamalla "More" voi valita mihin polkuun Script tallennetaan tietokoneelle ja missä muodossa. Syötin nämä tiedot ja aloitin tallennuksen. Kirjoitin hakupalkkiin transaktion, josta TVLV-tilien avoimet tapahtumat haetaan. Valitsin oikean variantin ja syötin kuluvan päivämäärän ja painoin "suorita". Variantit ovat SAP:in transaktioihin tallennettuja hakutietoja esimerkiksi käyttämäni variantti lisää SAP-hakuun oikeat tilinumerot, yhtiönumeron ja haettavan listan asettelun. Kun luettelo oli haettu, tallensin sen tietokoneelle Excel-muodossa ja lopetin nauhoituksen. Nauhoitetun Scriptin koodin saa auki menemällä tallennuskansioon, joka määriteltiin ennen nauhoitusta "More"-painikkeen alta, ja valitsemalla hiiren oikealla painikkeella "muokkaa".



Kuva 3. Polku Scriptin nauhoitukseen SAP:issa

Tämä koodi, jonka nauhoitin ei myöskään ole valmis koodi. Esimerkiksi tässä koodissa haetaan avoimia eriä päivämäärällä 14.9.2023 ja Excel tallennetaan minun tietokoneeni työpöydälle, jollaista polkua ei löydy, kun joku muu ajaa makroa eri tietokoneella. Tätä täytyisi siis vielä muokata niin, että se hakee avoimet erät aina kuluvalle päivämäärällä, ja että SAP:ista haettu luettelo tallennetaan aina siihen polkuun missä makrotyökirja sijaitsee. Näin sen pitäisi toimia kaikilla käyttäjillä, jotka makroa ajavat.

Seuraavaksi tein makrotesti-Excelin ja avasin VBA-editorin. Siellä löin työkirjaan moduulin, johon lähdin luomaan tätä Sub-proseduuria. Aloitin siis koodin Sub-rivillä Sub Makrotesti() ja lopetin End Sub -rivillä. Näiden rivien väliin sijoitin nauhoitetun SAP Scriptin -koodin, jonka jälkeen koodia voi kokeilla suorittaa. Sen suorittaminen alkaa järjestyksessä ensimmäisestä Sub-rivistä lopun End Sub -riviin asti. Ajo ei toiminut ja sain virheilmoituksen: "Compile error: Invalid use of property". Hetken tiedonhaun jälkeen löysin netistä syyn ongelmalle: "I figured that application is a reserved keyword in VBA and that was causing error by referring to incorrect objects (Troubleshooting connection... 2022)". Tämä tarkoittaa sitä, että "Application" on VBA-ohjelmointikielessä varattu avainsana eikä sitä voi siitä syystä käyttää muuttujana tai muuna määriteltävän elementin nimenä ohjelmassa. Muutin koodissa "Application" muotoon "sApplication" ja kokeilin ajaa uudelleen. Tällä kertaa ajo onnistui. Syy miksi koodi toimii, vaikka muutin "Application"-kohdat ovat se, että koodissa määritellään mitä se tarkoittaa. Käytännössä sitä käytetään siis vain nimenä. Makro haki avoimet erät SAP:ista ja tallensi ne määrittelemääni kansioon nimellä Avoimettesti1.

## **Viikon 1 yhteenveto**

Käytin ensimmäisellä tarkasteluviikolla suurimman osan ajasta tutustumiseen VBA-ohjelmointikieleen ja Excel-makrojen toimintaan. Kokeilin makrojen nauhoittamista ja suorittamista. Lisäksi tutustuin VBA-editoriin ja sen toimintoihin.

Makron ensimmäisestä vaiheesta sain toimimaan TV/LV-tilien avoimien erien listauksen hakemisen SAP-järjestelmästä ja sen tallentamisen Exceliin. Nauhoitin SAP:issa Scriptin, jonka avulla Excel-makro suorittaa SAP-haun. Tämä ei ole vielä valmis vaan vaatii muokkauksia, jonka avulla vaihe toimii tietokoneesta ja makron ajajasta riippumatta.

## **5.2 Viikko 2**

### **Maanantai 18.9.**

Seuraavaksi lähdin selvittämään, miten saisin viime viikolla nauhoitetun SAP Scriptin toimimaan oikein ja kaikilla käyttäjillä. Tällä hetkellä makro ei hae TV/LV-tilien avoimia eriä kuluvaan päivään, vaan päivään, jona nauhoitin SAP Scriptin 14.9.2023. Se ei myöskään tallenna haettua listausta oikeaan sijaintiin. Minun täytyy siis luoda ja määritellä muuttujia, joiden perusteella makro suorittaa haun ja tallentaa sieltä tuodun Excel-luettelon oikeaan sijaintiin. Lähdin tarkastelemaan Yritys X:n muita makroja, joiden toimintaperiaate on osin sama mitä kehittelemässäni makrossa tulee olemaan. Löysin makrosta apua päivämäärän muuttujaksi määrittelemiseen ja sen käyttämiseen nauhoittamassani koodissa. Määrittelin makroon muuttujan lauseella: "Dim alkupvm As Date" ja lisäsin koodiin rivin: "alkupvm = Date". Ensimmäinen lause luo alkupvm-muuttujan ja toinen lause määrittelee muuttujan alkupvm arvon kuluvaan päiväksi. Muuttujia määrittellessä voi vapaasti valita muuttujalle nimen esimerkiksi tässä tapauksessa käyttämäni muuttujan nimi olisi voinut olla vaikka pvm tai päivä. Koodissa rivi:  
`"session.findById("wnd[0]/usr/ctxtPA_STIDA").Text = "14.09.2023"` syöttää SAP-hakuun päivämäärän. Muokkasin koodiin päivämäärän 14.09.2023 tilalle alkupvm, jolla sen pitäisi toimia niin, että syöttää kuluvaan päivämäärän hakuun. (Organisaation makrotyökirja 2023.)

Tämän jälkeen täytyi muokata tallennuspolku. Lisäsin Global-muuttujat, jotka toimivat Sub-proseduurista riippumatta, mikäli ne sijaitsevat VBA-editorissa sen moduulin sisällä, johon kyseiset muuttujat on määritetty. Tämä tarkoittaa sitä, että vaikka minulla olisi käsiteltävässä moduulissa useampi Sub-proseduuri, nämä Global-muuttujat toimivat niissä kaikissa. Lisätyt muuttujat ovat: "Global MakroExcel As Workbook, Global apupolku As String, Global ApuExcel As Workbook". Tämän jälkeen määrittelin apupolku-muuttujan: "apupolku = "C:\Users\" & Environ("Username") & "\Desktop\""" ja lisäsin sen koodin kohtaan, jossa makro määrittelee tallennuspolun mihin haettu avoimet-listaus tallennetaan: "session.findById("wnd[1]/usr/ctxtDY\_PATH").Text = apupolku". Näin makro käyttää jatkossa apupolkua tallentaessaan luetteloja ja tämän polun pitäisi toimia kaikilla käyttäjillä.

Seuraavaksi määrittelin myös MakroExcel-muuttujan lauseella: "Set MakroExcel = Workbooks("makrotiedostonnimi.xlsm)". Seuraavaksi kirjoitin koodia, jolla makro avaa tallennetun tiedoston apupolku-muuttujan sijainnista ja tallentaa sen uudelleen samaan kansioon makrotyökirjan kanssa samalla vaihtaen tiedostomuodon .xls muotoisesta .xlsx-muotoon. Tässä käytin apuna aikaisemmin lisätyjä muuttujia. Koodissa määritellään ApuExcel ja avataan se käyttäen apupolku-muuttujaa. Sen jälkeen tallennetaan ApuExcel uudessa muodossa makrotyökirjan kanssa samaan polkuun käyttäen apuna MakroExcel-muuttujaa. Kokeilin ajaa ja ajo oli onnistunut. Nyt makro hakee ja tallentaa TVLV-tilien avoimet tapahtumat samaan kansioon makrotyökirjan kanssa. (Kuva 4).

```
Set ApuExcel = Workbooks.Open(apupolku & "Avoimet.xls")
ApuExcel.SaveAs Filename:=MakroExcel.Path & "\" & "Avoimet.xlsx", FileFormat:=51
ApuExcel.Close (False)
```

Kuva 4. Syntynyt koodinpätkä, joka tallentaa luettelon oikeaan kansioon käyttäen aikaisemmin määriteltäviä muuttujia.

### Keskiviikko 20.9.

Tällä hetkellä makron toimimista varten oikea SAP-moduuli täytyy olla avoinna ennen makron ajamista. Haluan, että SAP-moduulit voivat olla suljettuna aloitusnäyttöä SAP Logon -ikkunaa lukuun ottamatta, ja makron avaavan oikean moduulin itse. Tähän löysin koodin toimeksiantajan toiselta makrolta ja lisäsin sen

omaksi Sub-proseduuriksi makrotyökirjaan nimellä Sub SAP\_Logon(). Seuraavaksi täytyy määritellä minkä SAP-moduulin makro avaa. Lisäsin koodin Globalmuuttujaksi: "SAPmodule As String" ja määrittelin muuttujan lauseella: "SAPmodule = "(Avattava moduuli)". Lisäksi halusin, että makro ilmoittaa olevansa valmis, kun ajo on suoritettu ja tämä onnistui kirjoittamalla koodin loppuun: "MsgBox ("Makro on valmis!")". Nyt makro ilmoittaa ponnahdusikkunalla, kun ajo on päätynyt. Lisäsin vielä makron alkuun koodit, joiden avulla se lopettaa Excel-ikkunoiden päivytyksen ja varoitusten näyttämisen ja loppuun, joilla nämä asetukset palaavat ennalleen. Tämä nopeuttaa makron toimintaa. Nyt minulla oli valmiina makron ensimmäinen osa, joka hakee ja tallentaa lähtötiedot seuraavaa vaihetta varten. Tämän toiminta täytyy vielä testata muun käyttäjän toimesta. Lisäsin vielä napin makro-Exceeliin, jota painamalla se ajaa tehdyn Sub-proseduurin makrosta. (Organisaation makrotyökirja 2023.)

Seuraavaksi makron pitäisi muokata Avoimet-Excel niin, että valmiissa Excelissä on välilehdet: Yli 6 kk vanhat, alle 6 kk vanhat, tasattavat ja toimittajat ei tasata. Päätin tehdä tämän eri Sub-proseduurina, jotta makro voidaan ajaa osissa ja on helpompi huomata mahdollisia virheitä. Lisäsin toisen moduulin VBA-editoriin, johon ohjelmoin nyt seuraavaa vaihetta makrossa. Tein tämän, jotta on selkeämpää tarkastella eri osia makrosta ja ei vahingossa tule muokattua aikaisemmin jo toimivaksi tehtyä makron osaa. Lähdin taas katsomaan Yritys X:n ohjeistuksia, mallikoodia ja videoita (Organisaation makro-ohjeet 2023).

Halusin, että makro hakee Avoimet-tiedoston, ja siirtää sen ensimmäisen välilehden makrotyökirjan toiselle välilehdelle, jossa se sitten lähtee muokkaamaan sitä oikeanlaiseksi. Aluksi piti taas aloittaa koodi lauseella: "Sub makrotesti2()", jonka jälkeen määrittelin taas muuttujia ja polkuja, mitä kautta makro hakee ja avaa tiedoston Avoimet.xlsx. Tämän jälkeen kopioin ja muokkasin ohjeistuksista löytynyttä koodia, jotta se toimii tämän makron kanssa (Organisaation makro-ohjeet 2023). Muokatessa koodinpätkässä makro avaa Excel-tiedoston Avoimet, siirtää sieltä välilehden makrotyökirjaan, sulkee Avoimet-tiedoston tallentamatta ja nimeää tuodun välilehden "Avoimet" (Kuva 5).

```
Workbooks.Open Filename:=Avoimet 'open workbook
Sheets(1).Copy After:=Workbooks(makrotiedosto).Sheets(1) 'and move it's 1st sheet to macro workbook's 2nd sheet

Workbooks("Avoimet").Close False 'close opened workbook

ActiveSheet.Name = "Avoimet" 'rename the worksheet in macro workbook
```

## Kuva 5. Muokattu koodinpätkä

### Viikon 2 yhteenveto

Toisella viikolla sain muokattua toimivaksi koodia, joka jäi ensimmäisellä viikolla kesken. Nyt makro suorittaa TV/LV-tilien avoimien erien SAP-haun määriteltyjä muuttujia apuna käyttäen, joilla se hakee avoimet erät kuluvan päivän päivämäärällä ja tallentaa sen ensin apupolku-muuttujaa käyttäen makron ajajan tietokoneelle .xls muodossa. Tämän jälkeen avaa tallennetun tiedoston ja tallentaa sen vielä samaan kansioon makrot yökirjan kanssa .xlsx muodossa. Seuraavaksi makro tuo tallennetun listauksen makrot yökirjan toiselle välilehdelle. Ensimmäinen prosessissa syntyvä lopullinen Excel-tiedosto olisi tarkoituksena luoda makrot yökirjan välilehdille, josta se sitten tallennetaan valmistuessaan omaksi Excel-tiedostoksi ilman makrotoimintoja.

Ymmärrys makrojen tekemisestä ja VBA-editorin toiminnoista edistyi paljon. Ymmärrän myös muuttujien merkityksen, miten niitä määritellään ja miksi niitä on tärkeä käyttää. Kun muuttujat on määritelty oikein, ja makro suorittaa tämän talousprosessin niitä käyttäen, pitäisi sen toimia jatkossa ilman toimenpiteitä käyttäjältä. Työn tarkoituksena on automatisoida käsiteltävänä olevasta prosessista lähes kaikki vaiheet, joten on tärkeää, että se toimii jatkossa itsenäisesti.

## 5.3 Viikko 3

### Maanantai 25.9.

Makron toiselle välilehdelle tuodussa TV/LV-tilien avoimien tapahtumien luettelossa on tyhjiä rivejä ja sarakkeita. Seuraavaksi makron tehtävänä olisi poistaa nämä tyhjät rivit ja sarakkeet, sekä levittää sarakkeet ja korjata ostotilauksien summat sisältävien solujen desimaalit. Tämä oli suhteellisen yksinkertainen, sillä SAP:ista tuodussa luettelossa on aina samat rivit ja sarakkeet tyhjänä. Lisäsin koodiin: "Range("1:2").Delete", joka poistaa rivit 1 ja 2. Tein tämän myös muille

tyhjille riveille ja sarakkeille. Seuraavaksi täytyi korjata Summa-sarakkeen E desimaalit. Tämäkin oli yksinkertainen ja käytin tähän makronnauhoitusta. Valitsin sarakkeen E ja käytin näppäinyhdistelmää Ctrl, Shift + 1, joka korjaa desimaalit. Lisäsin vielä koodiin: "Columns.AutoFit", joka levittää kaikki sarakkeet solujen sisältävien arvojen levyiseksi.

Seuraavaksi makron pitäisi lajitella avoimien rivien -taulukko kirjauspäivän mukaisesti, jotta mahdolliset tyhjät rivit väleistä poistuisi. Tämä oli hieman haastavampi tehtävä, sillä en löytänyt ohjeista tai internetistä koodia, joka toimisi omani kanssa hyvin. Kokeilin siis tähänkin nauhoitusta. Se onnistui muuten, mutta koodissa Range-objekti oli vain sen pituinen, kuin luettelossa oli arvoja sisältäviä rivejä. Tämä ei toimisi jatkossa, sillä rivien määrä muuttuu joka kerta, kun prosessi tehdään. Tämä ei kuitenkaan vaatinut paljoa muokkausta ja löysin verkosta tähän apua. Lisäsin koodiin muuttujalausekkeen: "Dim SelectedArea As Range" ja "Set SelectedArea = Selection". Tämä määrittelee Rangeksi valittuna olevan alueen luettelosta. Sen jälkeen lisäsin virheelliseen koodinpätkään: ".SetRange SelectedArea", joka valitsee aikaisemmin määritellyn muuttujan Rangeksi. Tämän jälkeen nimesin sen omaksi Sub-proseduuriksi: "Sortbypostingdate" ja lisäsin alun perin käsiteltävänä olevaan Sub Osa 1 -proseduuriin lauseen: "Call Sortbypostingdate", joka ajaa tämän uuden Sub Sortbypostingdate siinä kohtaa Sub Osa 1 -proseduuria, johon tuo Call-lause on kirjoitettu. Muutoksien jälkeen tein testiajon ja se oli onnistunut. (Using Sort in VBA... 2014.)

Lisäsin myös makroon Yritys X:n ohjeissa olevan reset Sub -proseduurin (Organisaation makro-ohjeet 2023). Tämä poistaa kaikki paitsi ensimmäisen eli makrovälilehden makrotyökirjasta. Lisäsin makrotyökirjan makro-välilehdelle erillisen napin reset, jolla voi ajaa tämän kyseisen proseduurin ja resetoita makrotyökirjan alkupisteeseen. Laitoin tämän makron Osa 1 -koodiin myös Call reset, joka tyhjentää makrotyökirjan, jos sitä alkaa ajamaan niin, että siellä on jo muita välilehtiä avoinna. Näin se ei jää tästä syystä virheeseen ajon aikana ja resetoi aina automaattisesti makrotyökirjan ennen ajojen aloitusta.

## Torstai 28.9.

Makron seuraavana vaiheena olisi lajitella ostotilaukset niiden uusimman tapahtuman mukaan omille välilehdilleen yli 6 kk tai alle 6 kk. Lähdin työstämään tätä. Löysin Yritys X:n aineistosta koodia, minkä pitäisi muokkauksilla toimia tässä makrossa. Aluksi kuitenkin lisäsin koodia, jolla makro kopioi Avoimet-välilehden ja tekee siitä kaksi uutta välilehteä nimillä yli 6 kk ja alle 6 kk. Tämän jälkeen muokkasin ja lisäsin koodit, joilla makro poistaa rivit, joiden päivämäärä on joko yli tai alle 183 päivää tästä päivästä, riippuen kumpi välilehti on käsiteltävänä. (Organisaation makro-ohjeet 2023.)

Alla näkyvässä koodissa määritellään ensin muuttujan dateToday arvoksi tämä päivä. Sitten se valitsee Excel-taulukosta solun C2, joka on ensimmäisen ostotilauksen tapahtumarivin kirjauspäivän sisältävä solu. Tämän jälkeen makro tekee Loopin, jossa se vertailee rivin päivämäärää tähän päivään, ja poistaa sen, mikäli ero on alle 183 päivää. Jos ero on yli se jatkaa seuraavaan soluun. Makro tekee tätä siihen asti, kunnes aktiivinen solu on tyhjä. Kopioin tämän myös tekemään saman alle 6 kk -välilehdelle vaihtamalla koodista vain pienempi kuin 183 (< 183) suuremmaksi kuin 183 (> 183). Näin se poistaa rivit, joiden päivämäärä on yli 183 päivää tästä päivästä ja jäljelle jäävät rivit ovat alle 183 päivää kuluva päivästä. (Kuva 6.)

```
dateToday = Date 'define variable dateToday to be today
Range("C2").Select
Do
    dateSearched = ActiveCell 'define variable dateSearched to be active cell's value
    If DateDiff("d", dateSearched, dateToday) < 183 Then 'if date difference is less than 183 days then
        ActiveCell.EntireRow.Delete 'delete row
    Else
        ActiveCell.Offset(1, 0).Select 'otherwise move to next row
    End If
Loop Until ActiveCell = "" 'continue loop until active cell is empty
```

Kuva 6. Koodi, jolla poistetaan tapahtumat, joiden päivämäärä on alle 6 kk. Näin jäljelle jäävät rivit ovat yli 6 kk.

## Viikon 3 yhteenveto

Makro edistyi hyvin viikolla 3. Jokaista vaihetta tehdessä oma osaamiseni makron kehityksestä ja VBA-ohjelmoinnista kasvoi. Nyt makrossa on toiminto, jolla se nollataan alkupisteeseen. Lisäksi se järjestee avoimet rivit kirjauspäivän mukaisesti vanhimmasta uusimpaan. Tämä lajittelu poistaa myös väleistä mahdolliset



tyhjät rivit. Seuraavaksi makro kopioi avoimet rivit sisältävän välilehden omille välilehdilleen ja nimeää ne yli 6 kk ja alle 6 kk, jonka jälkeen se poistaa yli 6 kk -välilehdeltä kaikki tapahtumat, jotka ovat alle 183 päivää kuluva päivästä käyttäen Loop-komentoa, millä se toistaa toiminnon, kunnes määritelty ehto on tosi. Tässä tapauksessa se suorittaa Loopia, kunnes aktiivinen solu on tyhjä. Samoin se poistaa alle 6 kk -välilehdeltä kaikki tapahtumat, jotka ovat yli 183 päivää kuluva päivästä. Tämän vaiheen koodi oli vaikea tehdä, sillä en voinut hyödyntää siinä nauhoitusta. Kuitenkin pitkän ajan ja useiden kokeiluiden jälkeen sain muokattua koodista sellaisen, joka vertailee rivien päivämääriä nykyiseen päivään ja poistaa ne perustuen siihen onko ero yli tai alle 183 päivää.

#### **5.4 Viikko 4**

##### **Tiistai 3.10.**

Makron täytyy seuraavaksi tehdä ostotilausnumeron perusteella yli 6 kk -välilehdelle PHAKU-funktio, jolla haetaan samoja ostotilauksia alle 6 kk -välilehdeltä. Ostotilaukset, jotka löytyvät myös alle 6 kk -välilehdeltä poistetaan yli 6 kk -välilehdeltä. Kokeilin tätä aluksi nauhoittamalla ja katsomalla minkälaista koodia siitä syntyy. Nauhoitus on hyvä keino tehdä funktioita makrolla, sillä se muuttaa yhtälön koodissa VBA-muotoiseksi koodiksi.

Loin ja avasin apumakrotesti-Excelin, johon teen jatkossa testinauhoituksia, ettei nauhoitus tee koodia tai uusia moduuleja suoraan lopulliseen makroon. Näin on siis helpompaa sitten kopioida nauhoitetusta koodista pätkiä oikeaan makroon. Toin tähän apu-Exceliin välilehdet yli- ja alle 6 kk. Lisäsin alle 6 kk -välilehteen, jokaisen rivin perässä olevaan soluun alle6kk ja tätä tietoa haetaan seuraavaksi PHAKU-funktiolla. Tein tämän siksi, että seuraavassa vaiheessa tiedon suodattaminen on helpompaa tämän alle6kk-tekstin avulla.

Seuraavaksi aloitin nauhoituksen, valitsin M2 solun yli 6 kk -välilehdeltä ja tein PHAKU-funktion, jolla etsitään samoja ostotilausnumeroita alle 6 kk -välilehdeltä ja sijoitetaan kaavan soluun teksti alle6kk, mikäli sama ostotilaus löytyy. Kopioin kaavan jokaiselle riville. Tämän jälkeen lopetin nauhoituksen ja menin katsomaan

VBA-editorista miltä se näyttää. Nauhoitettu koodi on huono eikä toimi, kun lähötiedot muuttuvat. Kuitenkin funktion lisääminen selvisi nauhoittamalla ja sen koodi on seuraavanlainen: "ActiveCell.FormulaR1C1 = "=VLOOKUP(RC[-4], 'alle 6kk'!C9:C13,5,0)"". Sain nauhoittamalla koodia, joka valitsee oikean solun ja tekee siihen edellä mainitun kaavan. Seuraavaksi tein koodinpätkän, joka määrittelee muuttujan arvon A-sarakkeen viimeisen tietoa sisältävän rivin perusteella. Tätä muuttujaa voi käyttää Rangena koodissa. Sitten lisäsin koodiin rivin, joka täyttää nauhoitetun funktion taulukon jokaiselle riville M-sarakkeessa käyttäen aikaisemmin määriteltyä muuttujaa Rangena. Tämän se jälkeen kopioi koko M-sarakkeen ja liittää sen arvoina. Tämä poistaa siis kaavat ja nopeuttaa makron toimintaa. (Kuva 7.)

```
Sheets("alle 6kk").Select
Range("M2") = "alle6kk"
LR = Cells(Rows.Count, "A").End(xlUp).Row 'count last row based on column A
Range("M2").Select
Selection.AutoFill Destination:=Range("M2:M" & LR) 'copy alle6kk to all the rows in column M

Sheets("yli 6kk").Select
Range("M2").Select
ActiveCell.FormulaR1C1 = "=VLOOKUP(RC[-4], 'alle 6kk'!C9:C13,5,0)" 'vlookup that searches yli6kk sheets values from sheet alle6kk
Selection.AutoFill Destination:=Range("M2:M" & LR) 'copy vlookup to all the rows in column M

Columns("M").Copy
Range("M1").Select
Selection.PasteSpecial Paste:=xlPasteValues
Application.CutCopyMode = False
```

Kuva 7. Luotu koodinpätkä, joka lisää kaavat, joilla etsitään samoja ostotilauksia välilehdillä yli- ja alle 6 kk

Seuraavaksi makron täytyisi poistaa rivit, joille kaava toi tekstin "alle6kk". Kokeilin tehdä tätä nauhoituksella useampaan kertaan, mutta koodista tuli monimutkainen ja epäonnistunut. Onnistuin kuitenkin lopulta nauhoittamaan sen niin, että se vaati vain pientä muokkausta toimiakseen. Jouduin taas määrittelemään viimeisen rivin ja lisäämään sen arvon koodiin Rangeksi, jotta se osaa valita jatkossakin koko taulukon riippumatta sen rivimäärästä. Tämän jälkeen makro suodattaa kaikki rivit näkyviin, joiden M-sarakkeessa on "alle6kk". Sitten makro valitsee kaikki näkyvät rivit ja poistaa ne. Tämän jälkeen on jäljellä vain rivit, joiden ostotilauksia ei löydy alle 6 kk -välilehdeltä. Makro myös poisti otsikkorivin, joten lopuksi se vielä käy kopioimassa sen toiselta välilehdeltä ja liittämässä ylimmäksi riviksi. Loin tästä koodista oman Sub-proseduurin nimellä Alle6kk. Lisäsin tämän jälkeen Makrotesti2() loppuun Call Alle6kk, joka ajaa tuon luodun proseduurin.

Nyt yli 6 kk -välilehdellä oli ostotilauksia, joilla ei ole ollut mitään tapahtumia yli puoleen vuoteen. Näistä ostotilauksista täytyi vielä suodattaa pois ne, joiden summat ovat -1000 €-300 € välillä. Lähdin tätäkin tekemään nauhoittamalla ja loputkin vaiheet olivat lähes samanlaisia kuin aikaisemmin. Sama viimeisen rivin määrittäminen ja sen rajauksen lisääminen nauhoitettuun koodiin. Lisäksi täytyi korjata desimaaleja summista ja kopioida rajauksen jälkeen vain näkyvät solut ja liittää ne omalle välilehdelleen, joka nimetään "Tasattavat". Lopuksi vielä poistaa rajaus välilehdeltä yli 6 kk. Kun koodi oli taas hiottu, lisäsin sen lopulliseen makrokoodiin.

#### **Keskiviikko 4.10.**

Seuraavana vuorossa oli poistaa tasattavien joukosta toimittajat, joita ei tasata. SAP:ista täytyy hakea tasattavien ostotilausten numeroilla toimittajatiedot, jonka jälkeen poistaa ne toimittajat, jotka ovat listattuna toiseen Exceliin "Näitä ei tasata". Ensitöikseni lähdin nauhoittamaan SAP Scriptiä, jolla haetaan ostotilauksien toimittajatiedot.

Nauhoitin prosessin ja kopioin Scriptin omaan moduuliin VBA-editorissa. Tein sinne Sub-proseduurin "Toimittajat". Scripti oli muuten valmis, mutta siihen täytyi taas määrittellä muuttujien avulla tallennuspolut. Tämä oli jo tuttua sillä tein samat asiat jo aikaisemmin avoimia rivejä hakevan Scriptin muokkauksessa. Lisäksi täytyi lisätä koodinpätkä makroon, jolla se käy ensin kopioimassa Tasattavat-välilehdeltä ostotilausnumerot. Tämä oli yksinkertainen:

- Sheets("Tasattavat").Select (valitsee oikean välilehden)
- Range("I2").Select (valitsee solun I2)
- Range(Selection, Selection.End(xlDown)).Select (valitsee kaikki solut sarakkeelta, joissa on tietoa)
- Selection.Copy (kopioi valinnan)

Kun olin tehnyt tarvittavat muokkaukset, kokeilin ajaa sen ja se näytti toimivan hyvin.

Tein uuden moduulin mihin lähdin seuraavaksi tekemään koodia, jolla makro hakee aikaisemmin haetun Toimittajat-Excelin ja tuo sen makrottyökirjaan omalle välilehdelleen, jonka jälkeen poistaa siitä tyhjät rivit ja sarakkeet. Tämäkin on lähes sama prosessi kuin aikaisemmin, joten sain kopioitua aikaisemmista vaiheista lähes valmiin koodin tähänkin kohtaan.

Tämän jälkeen makro avaa tiedoston "NÄITÄ EI SAA TASATA (toimittajat)". Tämäkin vaihe oli suurimmaksi osaksi kopioitavissa aikaisemmista osista, joissa makro avasi muita Excel-tiedostoja. Muokattava oli osat, joissa määriteltiin mikä tiedosto avataan. Sitten lähdin taas nauhoittamalla tekemään vaihetta, jossa tehdään PHAKU-funktio aikaisemmin haetuille toimittajatiedoille, jonka avulla etsitään toimittajia NÄITÄ EI SAA TASATA (toimittajat) -Excelistä. Tämän jälkeen suodatin "#PUUTTUU!" sisältävät solut näkyviin ja poistin nämä rivit. Jäljelle jäi siis ostotilaukset, joiden toimittajat löytyivät ei-tasattavien joukosta. Sitten kopioin Tasattavat-välilehden uudelle välilehdelle ja nimesin sen "Toimittajat ei tasata", johon tein vielä uuden PHAKU-funktion, jolla hain Toimittajat-välilehdeltä ne ostotilaukset, joita ei toimittajien takia tasata. Tämän jälkeen vielä kertaalleen suodatin vain "#PUUTTUU!" näkyviin ja poistin ne, jolloin jäljelle jäivät vain nämä ei-tasattavat rivit, jotka löytyivät PHAKU-funktiolla. Seuraavaksi lisäsin näiden rivien perään "Ei tasata" ja tein vielä yhden PHAKU-funktion Tasattavat-välilehdelle. Tällä funktiolla hain osumia Toimittajat ei tasata -välilehdeltä ja osumista kaava tuo tekstin "Ei tasata". Tämän jälkeen suodatin näkyviin vain rivit, joilla oli tuo teksti ja poistin ne tasattavien joukosta. Sitten poistin koko sarakkeen, jolla kaavat olivat.

Tarkastelin nauhoitettua koodia ja se näyttäisi siltä, että toimii pienillä muokkauksilla. Jouduin useampaan kohtaan määrittelemään viimeisen solun käyttäen aikaisemmin käytettyä koodia: "LR = Cells(Rows.Count, "A").End(xlUp).Row", joka määrittelee viimeisen tietojen sisältävän rivin A-sarakkeen avulla. Jouduin lisäämään tämän koodin kohtiin, joissa vaihdetaan tarkasteltavaa välilehteä, jolloin tämän arvo muuttuu. Lisäksi lisäsin tämän LR-muuttujan kohtiin, joissa tarvittiin Rangen määrittelyä, kuten PHAKU-funktion lisääminen kaikille riveille ja suodattaessa puuttuvat näkyviin, että makro suodattaa koko taulukon eikä vain osaa siitä. Muokkauksien jälkeen kokeilin makron toimintaa ja se onnistui. Lisäsin vielä napin makrotiedoston ensimmäiselle välilehdelle, jonka nimesin "Osa 2".

#### **Viikon 4 yhteenveto**

Viikolla 4 makro eteni hyvää vauhtia. Se poistaa nyt yli 6 kk -välilehdeltä ne ostotilaukset, jolta löytyy tuoreempia tapahtumia myös alle 6 kk -välilehdeltä. Tämän jälkeen se luo Tasattavat-välilehden yli 6 kk vanhoista ostotilauksista suodattamalla vielä ne, joiden yhteenlasketut summat ovat yli -1000 € ja alle 300 €. Sitten

makro hakee seuraavassa vaiheessa SAP:ista tasattavien ostotilausten numeroilla toimittajatiedot. Näillä tiedoilla se etsii toimittajia Excelistä, jossa ei-tasattavat toimittajat ovat listattuna. Löydetyt ostotilaukset makro poistaa tasattavien joukosta ja vie omalle välilehdelleen makrottyökirjassa. Tasattavien joukosta pitää vielä poistaa tiettyjä rahteja sisältävät ostotilaukset, mutta tämän jätän myöhemmän vaiheeseen makron teossa. Viikolla neljä oli paljon työtä, testiajoo ja oppimista Excel-makron kehityksen suhteen.

## 5.5 Viikko 5

### **Keskiviikko 11.10.**

Makro muodostaa nyt valmiin tasattavien ostotilausten taulukon pois lukien rah-tirivien poistamista. Seuraavaksi vuorossa olisi tasauskirjausten tekeminen ja niiden tositteiden lisääminen taulukkoon. Tämän vaiheen makron ajaja tekee itse. Tasausten jälkeen kuitenkin makron tehtävänä on hakea uudelleen avoimet erät ja muodostaa näistä listaus, joka raportoidaan yhtiöön. Lisäksi muodostaa raportti kirjauksista syntyvästä tulos- ja varastovaikutuksesta. Lisäsin vielä makroon koodia, jossa se lisää tasattavat-välilehdelle otsikot tasaustosite ja kirjanpitosite, ostotilausrivien perään. Näiden alle merkataan ostotilauksien tasauksista syntyvät tositenumerot.

Tulos- ja varastovaikutusta haettaessa SAP:ista täytyy kyseiseen SAP-moduuliin kirjautua manuaalisesti. Yritin ohjelmoida makroa niin, että ajaessa Sub-proseduuria se kysyisi käyttäjätunnuksen ja salasanan, jonka avulla kirjautuisi moduuliin, mutta tämä ei jostain syystä onnistunut. Käytin paljon aikaa tähän yrittäessäni korjata ongelmaa, mutta en saanut sitä tällä kertaa toimimaan. Siirryin siis tois-taiseksi eteenpäin ja palaan tähän, mikäli haluan sen saada vielä toimimaan niin. Tällä ei kuitenkaan ole makron toimintaan suurta vaikutusta, sillä silti kirjautumistunnuksia olisi pitänyt syöttää makron ponnahdusikkunaan. Nyt siis täytyy kirjautua oikeaan SAP-moduuliin ensin ja sitten ajaa makroa.

Nauhoitin SAP Scriptin ja lisäsin sen makron koodiin, jolla se hakee kirjanpitositteiden avulla oikeasta transaktiosta listauksen niiden tulos- ja varastovaikutuk-

sesta ja tallentaa tämän apupolkua käyttäen Exceliin. Tätä hakiessa täytyy syöttää SAP:iin myös kuluva vuosi, jota ei vielä aikaisemmin ole täytynyt täyttää. Jouduin määrittelemään tätä varten muuttujan, johon löysin apua Yritys X:n jakamasta aineistosta. Muuttujan määrittely meni seuraavasti:

- Dim pvm As String
- Dim vuosi As String
- pvm = Now()
- vuosi = Format(pvm, "yyyy")

Tässä luotiin ensin kaksi muuttujaa pvm ja vuosi. Sen jälkeen pvm-muuttuja määriteltiin kuluva päiväksi, jonka jälkeen vuosi-muuttuja määriteltiin pvm-muuttujaa käyttäen myös kuluva päiväksi, mutta muotoon, jossa se näyttää vain vuosiluvun. Näin sen arvoksi tulee kuluvan vuoden vuosiluku. Lisäsin sen tämän jälkeen nauhoitettuun SAP Scripttiin kohtaan, jossa se syöttää haettavan vuoden SAP-hakuun. (Organisaation makro-ohjeet 2023.)

Lisäsin vielä loppuun koodia niin, että makro avaa tallennetun .xls tiedoston ja tallentaa sen makrokirjan kanssa samaan kansioon .xlsx muodossa. Tämän jälkeen makro ilmoittaa ponnahdusikkunalla "Makro on valmis!".

### **Viikon 5 yhteenveto**

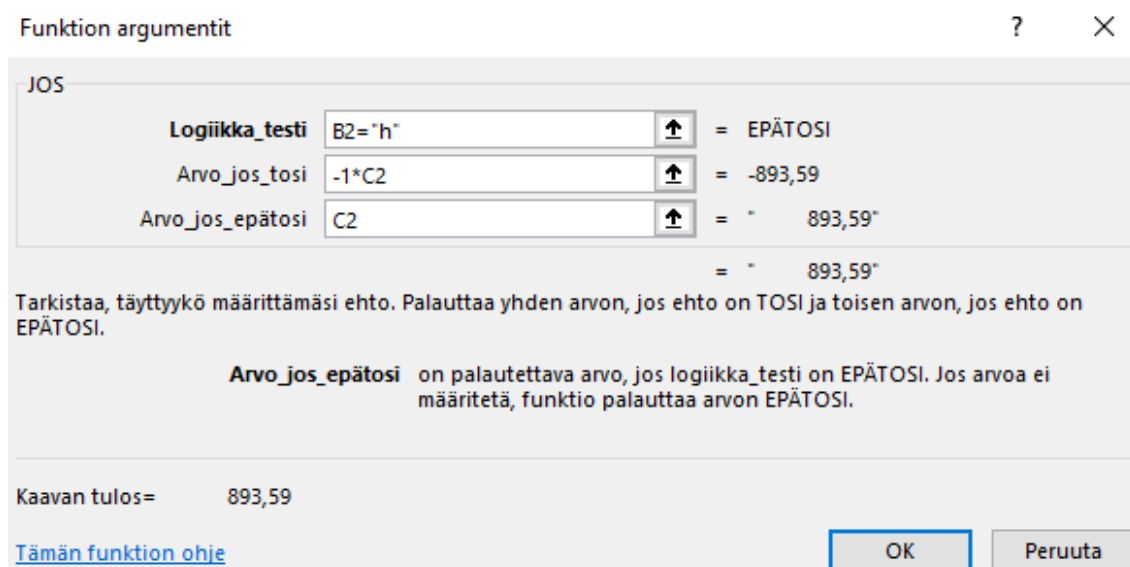
Viikolla 5 makrosta valmistui vaihe, jossa se hakee SAP:ista tasauskirjausten tositenumeroilla listauksen ja tallentaa sen makron ajajan tietokoneelle. Sieltä se avaa vielä tallennetun Excel-tiedoston ja tallentaa sen kertaalleen oikeassa .xlsx muodossa makrottyökirjan kanssa samaan kansioon. Haetun listauksen avulla lasketaan tasauskirjauksista syntynyt tulos- ja varastovaikutus.

Käytin viikolla 5 paljon aikaa ongelman selvittämiseen, jossa en saanut makroa avaamaan itsenäisesti oikeaa SAP-moduulia, sillä avatessa sitä täytyi moduuliin kirjautua käyttäjän omilla tunnuksilla. Tämä ongelma ei vielä ratkennut ja palaan siihen myöhemmin, mikäli keksin ratkaisun tai on aikaa perehtyä lisää ongelmaan.

## 5.6 Viikko 6

### Maanantai 16.10.

Seuraavaksi makron täytyy tuoda aikaisemmin tallennettu listaus makrotyökirjalle omalle välilehdelle ja muokata sitä. Tämän jälkeen laskea vielä uudelle välilehdelle tasauksista syntynyt tulos- ja varastovaikutus ja täsmäyttää näitä tasattavien tositteiden yhteissummaan. Kirjoitin aluksi koodia, jolla makro tuo lähtötiedot makrotyökirjaan ja poistaa siitä tyhjät rivit ja sarakkeet väleistä. Tämän jälkeen makron täytyy muuttaa kredit-kirjausten etumerkiksi -merkki sillä lähtötiedoissa kaikki luvut ovat positiivisessa muodossa. Tämän tein nauhoittamalla Excelissä. Laitoin nauhoituksen päälle ja tein =JOS kaavan (Kuva 8). Sitten kopioin kaavan jokaiselle riville ja poistin siitä funktiot kopioimalla ja liittämällä arvot.



Kuva 8. JOS-kaava mikä muuttaa kredit-kirjausten etumerkit miinus-merkkiseksi.

Seuraavaksi nauhoitin makrolla Tulos- ja varastovaikutus -välilehden luonnin ja muokkauksen. Lisäksi löysin netistä apua koodiin, jolla loin reunaviivoja luotuihin soluihin (Excel VBA - ... 2012). Sitten lisäsin koodiin taas useamman funktion, joilla raportoidaan tulosvaikutus, varastovaikutus, tasaukset yhteensä, tasaukset velan puolella ja – saamisen puolella. Tulosvaikutuksen sai tekemällä Summa.Jos.Joukko -funktion missä se laskee yhteen kaikki rivit missä on tulos-tilit. Varastovaikutuksen niin, että samanlainen funktio laskee yhteen rivit, joissa on varastotili. Tasaukset saamisen ja velan puolella lasketaan myös samanlaisella funktiolla, mutta ne lasketaan tasattavien välilehdeltä, niin että saamisen

puolella olevan funktion ehtona on ostotilauksen summa  $>0$  ja velan puolella ostotilauksien summat ovat  $<0$ . Tasaukset yhteensä ehtona laitoin ostotilausten summa  $>-1001$ , sillä tasattavat ostotilaukset on jo rajattu  $-1000$  €- $300$  € välille. Tällöin laskee kaikki yhteen. (Kuva 9.)

<b>Tulos- ja varastovaikutus</b>	
Tulosvaikutus	
Varastovaikutus	
<b>Yhteensä</b>	
<b>Tasaukset yhteensä</b>	
Tasauksista velan puolella	
Tasauksista saamisen puolella	
<b>Yhteensä</b>	

Kuva 9. Luotu välilehti. Tulos- ja varastovaikutus yhteensä täytyy täsmätä Tasaukset yhteensä summaan.

## Viikon 6 yhteenveto

Tällä viikolla makrosta valmistui vaihe, jossa makro tuo viime viikolla haetun listauksen makrotyökirjaan omalle välilehdelle. Sen jälkeen luo välilehden Tulos- ja varastovaikutus, johon se laskee funktioiden avulla tuodusta listauksesta yhteen tulostilien ja varastotilien summat. Lisäksi se laskee tasausten summat tasattavat-välilehdeltä, joiden täytyy täsmätä tulos- ja varastovaikutusten yhteensä summaan. Nämä summat makron ajaja tarkistaa itse ajon valmistuttua.

## 5.7 Viikko 7

### Keskiviikko 25.10.

Lisäsin makrotyökirjan ensimmäiselle välilehdelle viime viikolla luotuihin vaiheisiin painikkeet. Nimesin toisen "Hae \*\*\*\*", jota painamalla makro hakee SAP:ista listauksen, jonka avulla tulos- ja varastovaikutus tehdyistä tasauskirjauksista lasketaan. Ohjelma tallentaa tämän luettelon makrotyökirjan kanssa samaan kansioon. Toinen painike on "Tulos- ja varastovaikutus", jota painamalla makro laskee äsken haetusta tiedosta tulos- ja varastovaikutuksen omalle välilehdelle makrotyökirjaan.



Kun tasauskirjaukset on tehty, ja niiden tulos- ja varastovaikutus on makron toimesta laskettu, seuraava vaihe on se, että haetaan uudelleen TV/LV-tilien avoimet rivit ja muokataan siitä uusi Excel, joka raportoidaan eteenpäin tasausten kanssa. Ensimmäisessä vaiheessa tehty Sub-proseduuri, jossa makro hakee tilien avoimet erät, toimii myös tässä vaiheessa eikä sitä tarvitse kertaalleen tehdä. Voin kuitenkin makroa viimeistellessä luoda sille toisen napin, jotta se tekee makron ajamisesta selkeämpää.

Loin makroon uuden Sub-proseduurin, jossa se luo uuden Excelin "TVLV Avoimet erät" ja tuo kyseiseen tiedostoon tämän uudelleen haetun avoimien erien - listauksen. Kopioin makron aikaisemmasta koodista myös tähän ohjelmaan päätäkää, sillä avoimet-Excelin ensimmäinen välilehti muokataan samanlaiseksi myös tässä työvaiheessa. Tämän jälkeen kuitenkin ostotilaukset järjestellään omille välilehdilleen vuosien eikä kuukausien perusteella. Tämän työvaiheen ohjelmointi tuotti paljonkin vaikeuksia, sillä halusin sen toimivan jatkossa ilman muokkauksia, vaikka vuodet vaihtuvatkin. Pitkän ihmettelyn ja kokeiluiden jälkeen sain kuitenkin tehtyä Loopin, jolla makro poistaa kaikki muut paitsi nykyisen vuoden rivit. Tätä varten minun piti muokata lähtötietoja niin, että year/month-sarakkeessa on vain vuosiluvut. Tämän jälkeen lisäsin koodiin muuttujan "vuosi" ja määrittelin sen täksi vuodeksi. Sitten muokkasin aikaisemmin käytettyä Loopia, jossa vertailtiin päiviä toisiinsa niin, että se vertailee vuotta ja poistaa kaikki rivit, joiden ero on > 0 tämän vuoden kanssa. Näin jäljelle jäävät rivit ovat vain tämän vuoden rivejä. Seuraavaksi täytyy tehdä uusia välilehtiä ja muokata niihin tätä koodia niin, että se jättää vain edellisen vuoden rivit, sitä edellisen vuoden ja niin edelleen.

### **Torstai 26.10.**

Lähdin ohjelmoimaan osuutta, jossa ostotilaukset jaetaan vuosien perusteella omille välilehdilleen. Jonka jälkeen vuosi kerrallaan haetaan ostotilauksilla uudet luettelot SAP:ista, samasta transaktiosta missä tasauskirjaukset tehtiin, ja sitä muokataan ja lähetetään eteenpäin yhtiöön. Poistin makrosta toiminnon, joka muokkaa year/month sarakkeen vain vuosiluvuksi, sillä lisään sen vasta seuraavassa vaiheessa muokkaamaan luotuja uusia välilehtiä. Tämän jälkeen määrittelin muuttujia niin kuin aikaisemmissakin vaiheissa. Muuttujiksi piti määritellä nykyinen vuosi ja siitä 3 vuotta taaksepäin, sillä tilillä on useammalta vuosilta tapahtumia. (Kuva 10.)

```

Dim vuosi As String
Dim evuosi As String
Dim ivuosi As String
Dim uvuosi As String
Dim pvm As String
Dim LR As Long 'last row
Dim kk As Long
Dim dateToday As Date 'today

dateToday = Date

vuosi = year(Date)
evuosi = (year(Date) - 1)
ivuosi = (year(Date) - 2)
uvuosi = (year(Date) - 3)

```

Kuva 10. Tähän on määritelty vuosi-muuttujat. Vuosi on tämä vuosi, evuosi on vuosi takaperin, ivuosi kaksi vuotta takaperin ja niin edelleen.

Seuraavaksi kirjoitin koodia, jolla makro lisää neljä kopiota ensimmäisestä välilehdestä ja nimeää ne yhdestä neljään. Tämän jälkeen se poistaa ylimääräiset rivit lopusta ja muokkaa year/month-sarakkeet näyttämään vain vuosilukua. Sitten suorittaa Loopin jokaiselle välilehdelle aikaisempien vuosi-muuttujien avulla, joka poistaa kaikkien muiden vuosien rivit. Esimerkiksi välilehdeltä 4 Loop poistaa kaikki rivit, joiden vuosi eroaa muuttujasta uvuosi, joka on kolme vuotta nykyvuodesta taaksepäin eli tässä tapauksessa se poistaa kaikki rivit, jotka eivät ole vuoden 2020.

Kun makro on tehnyt edellisen vaiheen, on seuraavaksi taas PHAKU-funktion tekemistä kaikille aikaisempien vuosien välilehdille. PHAULLA haetaan esimerkiksi Vuoden 2020 ostotilauksia vuoden 2021 välilehdeltä ja jos niitä löytyy makro poistaa nämä rivit välilehdeltä 2020, sillä tuorein tapahtuma on vuodella 2021. Tämän vaiheen makro suorittaa kaikille paitsi tuoreimman vuoden välilehdelle. Lisäsin tähän vaiheeseen myös If-lauseen, jolla makro poistaa koko välilehden, mikäli siitä ei löydy yhtään ostotilausta. Esimerkiksi tätä makroa tehdessäni kaikki vuoden 2020 ostotilaukset löytyivät myös välilehdeltä 2021, jolloin 4-välilehti jäi tyhjäksi. Tämä If-lause poistaa koko välilehden, mikäli se on tyhjä ja jos se ei ole tyhjä se poistaa M-sarakkeen, jossa PHAKU-funktiot olivat. (Kuva 11.)

```

Range("I2").Select

    If ActiveCell = "" Then 'if active cell is empty then
    ActiveSheet.Delete 'delete row
    Else: Columns("M:M").Delete 'delete column containing vlookup
    End If

```

Kuva 11. If-lause, joka poistaa tyhjän välilehden tai funktiot sisältävät sarakkeen.

Seuraavaa vaihetta varten ostotilauksien perästä pitää poistaa ylimääräiset nollat kaikilta välilehdiltä. Tämän vaiheen nauhoitin makron nauhoitus -toiminnolla Excelissä käyttäen teksti sarakkeisiin -toimintoa. Tässä kohtaa makro tulee virheeseen, mikäli viimeistä välilehteä ei löydy, jos se aikaisemmassa vaiheessa poistettiin. Lähdin etsimään netistä apua, miten saan koodattua makron niin, ettei se pysähdy tähän virheeseen vaan suorittaa toiminnot, mikäli välilehti löytyy, ja hypää seuraavaan vaiheeseen, mikäli sitä ei löydy. Löysin ratkaisun ongelmaani verkosta ja lisäsin: "On Error GoTo eh" -koodin ennen 4-välilehden valintaa, jolloin jos tässä vaiheessa makro kohtaa virheen se hyppää kohtaan "eh:" ilman makron pysähtymistä (Kelly n.d.).

Tässä vaiheessa makro hakee eri vuosien ostotilauksien avulla luettelot samasta transaktiosta missä aikaisemmassa vaiheessa tehtiin tasauskirjaukset. Lähdin nauhoittamaan SAP Scriptiä, jossa hain vuosi kerrallaan nämä listat ja tallensin ne tietokoneelleni. Tein tämän neljä kertaa, sillä joskus näitä vuosia voi olla neljä, vaikka tekohetkellä viimeinen välilehti jäikin tyhjäksi. Nauhoituksen jälkeen lisäsin sen koodin VBA-editoriin omalle moduulilleen. Tämän jälkeen määrittelin taas muuttujat listojen tallennusta varten, johon kuului esimerkiksi apupolku- ja vuosi-muuttujat. Jaoin SAP Scriptin vaiheisiin koodissa joka vuosien listan haun perusteella. Muokkasinkin koodia niin, että se tallentaa luettelot aluksi apupolkuun nimillä 1-4.xls. Tämän jälkeen se avaa nämä haetut Excel-luettelot ja tallentaa ne makrotyökirjan kanssa samaan kansioon käyttäen vuosi-muuttujia. 1 tallennetaan vuosi-muuttujalla ja 2 taas evuosi-muuttujalla ja niin edelleen. Tähänkin vaiheeseen jouduin laittamaan On Error GoTo -komennon, jolla se hyppää 4-välilehden SAP-haut ja tallennukset yli, mikäli sitä välilehteä ei löydy.

## **Viikon 7 yhteenveto**

Tällä viikolla oli paljon ihmettelemistä sen suhteen, miten saan makron toimimaan, mikäli ostotilaukset jakautuvat eri vuosille jatkossa. Löysin ratkaisun ongelmaan käyttämällä On Error -komentoa ja kokeilin tähän vaiheeseen useampia variaatioita. Kokeilin aluksi On Error Resume Next -komentoa, mutta tajusin nopeasti, että tällä komennolla makro hyppää vain sen rivin yli, jossa virhe tuli. Tarvitsin kuitenkin komennon, jolla makro hyppää koko vaiheen yli, mikäli käsiteltävänä olevalla vuodella ei ole yhtään ostotilausta. Päädyin On Error GoTo -komentoon, jolla voi määrittellä itse, mihin kohtaan makro hyppää virheen sattuessa.

Muutoin makro edistyi hyvin tällä viikolla. Nyt makron vaihe on valmis, jossa se jakaa ensin TV/LV-tilien avoimet rivit vuosittain omille välilehdilleen. Tämän jälkeen se suorittaa SAP-haut vuosittain välilehtien ostotilausnumeroilla ja tallentaa ne makrotyökirjan kanssa samaan polkuun nimeämällä tiedostot vuosi-muuttujien perusteella.

## **5.8 Viikko 8**

### **Keskiviikko 1.11.**

Seuraava työvaihe oli tuoda viime viikolla vuosi kerrallaan haetut ja tallennetut avoimet rivit "TVLV avoimet erät" -työkirjaan välilehtien 1–4 tilalle ja muokata ne lopulliseen muotoon. Lähdin luomaan uutta Sub-proseduuria ja määrittelin taas muuttujia aikaisempaan tapaan, kuten vuosi-muuttujat ja MakroExcel-muuttujan, jota käytetään tiedostojen avaamisessa ja välilehtien viennissä. Tämän jälkeen hain Reset-proseduurin, jota kuvailin päiväkirjaraportoinnin maanantai 25.9. sivuilla 19–20. Tämä ohjelma siis poistaa työkirjan kaikki välilehdet viimeisestä lähietien, kunnes jäljellä on vain ensimmäinen kaikki avoimet rivit sisältävä välilehti. Tämän jälkeen kopioin ja muokkasinkin taas aikaisemmista vaiheista koodia, jolla makro avaa kuluvan vuoden työkirjan ja kopioi ja liittää sen "TVLV avoimet erät" -työkirjaan toiseksi välilehdeksi nimen sen vuosi-muuttujaa käyttäen.

Tämän jälkeen seuraava vaihe oli muokata välilehdet lopulliseen muotoon. Menin kuluvan vuoden välilehdelle ja laitoin nauhoituksen päälle. Poistin välilehdeltä yli-

määräiset tyhjät rivit ja sarakkeet, muodostin otsikkorivin, tein mukautetun lajittelun ostotositteen mukaan ja tein välisummat ostotositteiden mukaan, joka laskee eron arvo -sarakkeen arvot yhteen ja viimeiseksi lihavoin ja kiinnitin otsikkorivin. Eron arvo -sarakkeessa on summa, joka rivin osalta on avoimena tilillä. Lopetin nauhoituksen ja tarkistelin syntynyttä koodia. Koodi vaikuttaa siltä, että se toimii tällaisenaan ilman suurempia muokkauksia. Jälleen kerran vaadittavat muokkaukset olivat viimeisen rivin määrittäminen käyttäen LR-muuttujaa, jotta makro osaa muokata kaikkia tietoja sisältäviä rivejä lähtötietojen muuttuessa. Nauhoituksessa syntyneessä koodissa muokataan välilehteä 2023, jonka tilalle lisäsin vuosi-muuttujan, jolloin se osaa taas jatkossa muokata kuluvan vuoden mukaan nimettyä välilehteä. Tämän jälkeen ajoin koodia rivi kerrallaan tarkistaen, että se toimii suunnitellusti, jonka jälkeen kopioin tämän koodin vielä kolme kertaa muuttaen siihen vuosi-muuttujat, jolloin se suorittaa saman toiminnon kaikille vuosille. Ja kertaalleen lisäsin On Error GoTo -komennon koodiin, jolla se ohittaa toiminnot viimeisen vuoden osalta, mikäli sitä tiedostoa ei löydy. Ajoin luodun Sub-proseduurin kokonaisuudessaan ja se toimi moitteetta.

Huomasin vielä tarkastellessa syntynyttä tiedostoa, että välisummissa ostotilauksien lopullinen summa näyttää jokaisen ostotilauksen osalta 0 ja virheenä oli se, että luvut luettelossa ovat tekstimuodossa. Laitoin nauhoituksen päälle ja valitsin sarakkeen, jossa luvut ovat. Tämän jälkeen valitsin toiminnon teksti sarakkeisiin, valitsin ”kiinteä leveys” ja sarakkeen tietojen muodoksi ”yleinen”. Tämän muuttaa luvut numeroiksi, päivämäärät päivämääriksi ja muut arvot tekstiksi. Painoin valmis ja lopetin nauhoituksen. Lisäsin syntyneen koodinpätkän kohdan loppuun, jossa poistettiin tyhjät sarakkeet/rivit ja muodostettiin otsikkorivi. Kokeilin ajaa tämän kohdan ja se toimi, joten kopioin sen jokaisen välilehden koodiin. Nyt välisummat muodostuivat oikein.

### **Torstai 2.11.**

Makron viimeinen työvaihe oli käsiteltävänä. Makron pitäisi luoda sähköpostit, joiden liitteenä on luotu TVLV Avoimet erät -tiedosto ja tekstinä on tasauksien summat ja niiden tulos- ja varastovaikutus. Löysin toimeksiantajan aineistosta ohjeet sähköpostien lähettämiseen Excel-makrolla. Kopioin koodinpätkän ja lähdin ihmettelemään sitä. (Organisaation makro-ohjeet 2023.)

Ohjeista otetussa koodissa on aina sama teksti, mutta minulla olisi tarkoituksena ottaa tekstiin luvut makrotiedoston Tulos- ja varastovaikutus -välilehdeltä, joten loin muuttujat tulos, varasto, velka, saaminen ja määrittelin ne välilehden soluiksi, jossa nuo luvut ovat. Tämän jälkeen lähdin muokkaamaan tekstiä ja lisäsin sinne nuo muuttujat. Tekstiä tehdessä "<BR>" on rivinvaihto ja "&nbsp;" on välilyönti, näitä käytin sähköpostien luomisessa. Määrittelin koodiin, kenelle se lähetetään ja testauksen vuoksi käytin omaa sähköpostiani. Aiheen lisäykseen lisäsin myös päivämäärä muuttujaksi: "datetoday" ja määrittelin sen kuluvaksi päiväksi, jonka jälkeen laitoin aiheen ja: "& datetoday", jolla se lisää aiheen perään päivämäärän loppuun. Seuraavaksi liitteen lisäämiseksi lisäsin vielä liite-muuttujan ja määrittelin sen avoimien erien tiedostoksi. Lisäsin koodiin: ".attachments.add liite", jolla se lisää tuon liitteen sähköpostiin.

Tein testiajon tälle makron ohjelmalle ja se toimi, mutta ei täydellisesti. Liitteeksi laitettun tiedoston "TVLV avoimet erät 2.11.2023" välilyöntien tilalle tuli "&20". Lähdin etsimään netistä tietoa mistä tämä ongelma voi johtua ja löysin vastauksen. Sillä liitetiedosto sijaitsee SharePointissa, on sen sijainti URL-osoite eikä perinteinen tiedostosijainti, jolloin välilyöntien tilalle tulee tuo "%20". Samalla sivulla oli kuitenkin korjaus ongelmaan ja lisäsin tuohon liitteen lisäys kohtaan With-lauseen:

- With .Attachments.Add(Liite)
- .DisplayName = Replace(.DisplayName, "%20", " ")
- End With

Tällä koodilla se korvaa näytettävässä tiedostonimessä kohdat "&20" välilyönneillä. Kokeilin ajaa koodia ja tämä osuus toimi. (Excel VBA attaching... 2023.)

Sitten siirryin seuraavaan ongelmaan. Välilehdellä Tulos- ja varastovaikutus olevat luvut näytetään kahden desimaalin muodossa, mutta ne sisältävät useampia desimaaleja ja makron ajaessa nuo muuttujat, se lisää sähköpostiin nuo luvut kokonaisuudessaan kaikilla desimaaleilla. Aluksi ajattelin käyttäväni teksti sarakkeisiin -toimintoa ja valitsemalla tietyn pituuden solussa oleville luvuille, mutta tajusin tämän olevan huono ratkaisu, sillä lukujen muuttuessa esimerkiksi satojen ja tuhansien välillä se katkaisee luvut väärästä kohdasta. Lähdin etsimään netistä tietoa ja löysin sivuston, jolla oli ratkaisu tähän ongelmaan. Jouduin muuttamaan SUMMA-funktioita aikaisempaan tulosjavarastovaikutus Sub-proseduuriin, jossa

nuo luvut lasketaan. Näiden funktioiden eteen lisäsin ROUND-funktion ja perään, kuinka moneen desimaaliin se pyöristetään. Esimerkkinä SUMMA-funktio, jolla lasketaan tasaukset velan puolella, muokkasin alkuperäisestä muodosta: "ActiveCell.FormulaR1C1 = "=SUMIFS(Tasattavat!C5,Tasattavat!C13,""<0"")"" Muotoon: "ActiveCell.FormulaR1C1 = "=ROUND(SUMIFS(Tasattavat!C5,Tasattavat!C13,""<0""),2)"". Ajoin tämän jälkeen muokatun osan makrosta kokonaisuudessaan ja se toimi moitteetta. Tämän jälkeen kokeilin ajaa uudelleen ja nyt luvut tulivat oikein sähköpostiin. Tein tästä osiosta oman Sub sähköposti -proseduurin. (How to Combine... 2022.)

Muokkasin seuraavaksi oikeat sähköpostit koodiin ja kokeilin ajaa sitä uudelleen. Makro toimi muuten, mutta sähköpostisovellus ei tunnistanut sähköposteja kahdeksi eri sähköpostiksi, vaikka niiden välissä oli välilyönti. Kokeilin sähköpostin vastaanottajat-kohdassa lisätä niiden välille "; " ja tämä näytti toimivan. Lisäsin pätkän myös makroon ja ajoin uudelleen. Tällä kertaa se toimi oikein. Lisäsin vielä koodiin uudet rivit toisen sähköpostin luontiin, johon tulee sama liite, mutta eri teksti ja vastaanottajat. Tein tarvittavat muokkaukset ja kokeilin ajaa uudelleen. Nyt makro luo sähköpostin luonnoksiin kaksi erilaista sähköpostia, jonka jälkeen makron ajaja voi vielä tarkistaa niiden oikeellisuuden ja lähettää ne eteenpäin.

### **Perjantai 3.11.**

Laitoin prosessin vakituiselle tekijälle kyselyä vielä sähköpostien mallista, jotta makro luo varmasti oikeanlaiset. Hän lähetti minulle mallisähköpostit, joiden perusteella muokkasin vielä makron koodia. Ensimmäinen viesti oli hyvä sellaisenaan, mutta toinen viesti jaetaan vastaanottajien kesken omiksi viesteiksi, koska heille lähetettyihin viesteihin suodatetaan näkyviin tietyt tulosityksiköt TVLV Avoimet erät -tiedoston ensimmäiseltä välilehdeltä. Kopioin toisen viestin koodin ja liitin sen uudelleen koodin perään. Nauhoitin makroa, jolla kävin ensimmäisen sähköpostin luonnin jälkeen, suodattamassa tietyn tulosityksikön näkyviin TVLV avoimet erät -tiedostosta ja tallensin sen. Muokkasin toiseen viestiin otsikon perään tulosityksikön numeron ja viestitekstin muotoon: "Ohessa TV/LV-tilien avoimet erät (kuluva päivä) tulosityksikölle \*\*\*\*". Tämän jälkeen kopioin nauhoitetun osuuden uudelleen viimeisen viestin eteen ja muutin siihen tulosityksikön. Lisäksi

muokkasin viestin tiedot samalla tavalla kuin edellisen. Ajoin ohjelman ja tarkistin viestien oikeellisuuden ja liitteiden suodatukset. Kaikki toimi niin kuin pitikin.

Seuraavaksi siirryin takaisin makrotiedoston muokkaamiseen. Lähdin tekemään osaa, jonka jätin tarkoituksella myöhemmälle. Tässä osassa poistetaan tasattavien joukosta tietyt rahdit, joita ei tasata. Ensimmäinen vaihe on hakea SAP:ista luettelo samasta transaktiosta, missä tasauskirjaukset tehdään. Tämä luettelo täytyy suodattaa rahtien mukaan, jonka jälkeen viedä Excelliin. Päätin tehdä tämän haun samaan makron osaan missä haetaan SAP:ista ostotilauksien toimittajatiedot. Käsittelin kyseistä makron vaihetta 4.10. sivulla 22. Lähdin nauhoittamaan SAP Scriptiä. Nauhoituksen jälkeen lisäsin Scriptin makron koodiin. Koodiin muokkasin taas päivämäärät ja tallennuspolun käyttäen jo tässä makron osassa olemassa olevia muuttujia. Samoin tallennus makrotyökirjan kanssa samaan kansioon onnistui olemassa olevien muuttujien avulla.

Tämän jälkeen lähdin tekemään vaihetta, jossa haetun listauksen perusteella poistetaan nuo rahdit tasattavien joukosta. Tämän lisäsin makron kohtaan, jossa poistetaan toimittajat, joita ei tasata. Tätä makron kohtaa käsittelin 4.10. sivulla 26. Lähdin nauhoittamaan myös tätä vaihetta. Avasin uuden luettelon ja poistin siitä ylimääräiset sarakkeet ostotilauksien ja rahtitietojen välistä. Sitten suodatin vain rahtirivit näkyviin, sillä SAP-suodatuksen jälkeen siellä on silti joukossa ylimääräisiä rivejä. Tämän jälkeen tein uuden välilehden makrotiedostoon ja nimesin sen "\*\*\*\* Rahti". Kopioin näkyvät solut uudesta luettelosta ja liitin ne luodulle välilehdelle. Tämän jälkeen tein tasattavat-välilehdelle PHAKU-funktion, jolla hain samoja ostotilauksia rahdit välilehdeltä. Tämän jälkeen lisäsin funktion kaikille riveille, kopioin ja liitin arvoina. Poistin tiedot uudelta rahdit sisältävältä välilehdeltä ja kopioin tasattavien joukosta vain näkyvät solut, joilla on nämä rahdit. Liitin nämä uudelle välilehdelle ja poistin ne tasattavien joukosta. Testauksen jälkeen tämäkin vaihe toimi. Nyt makro sisältää kaikki prosessin vaiheet ja vaatii enää hienosäätöä.

Seuraavaksi lisäsin makroon kohdan, jossa se kopioi makrotiedoston kaikki välilehdet lukuun ottamatta makrovälilehteä, ja tallentaa omana tiedostonaan: "\*\*\*\* TVLV erät alle 1000e vel. tai 300e hyv. yli 6 kk (kuluva päivä)". Tämä tulee olemaan toinen lopullinen makron tuottama Excel-tiedosto. Tämä vaatii vielä säätöä



sillä, jos makron ajaja lisää omia välilehtiä makrotyökirjaan esimerkiksi tasauskirjauksia tehdessä, tämä ei tuo sitä välilehteä uuteen tiedostoon. Lisäsin myös kaikkien makronosien napit makro-välilehdelle, joita painamalla makro tullaan ajamaan.

Käsittelin 11.10. sivulla 28, makron vaihetta missä haetaan SAP:ista tietoja moduulista, johon tarvitsi kirjautua syöttämällä manuaalisesti tunnukset. Keskustelin asiasta prosessin vakituisen tekijän kanssa ja selvisi, että vain minun täytyy kirjautua tällä tavalla kyseiseen moduuliin. Laitoin Yritys X:n IT-tuelle viestiä asiasta ja he korjasivat ongelman, jonka jälkeen tätä manuaalista kirjautumista ei tarvitse tehdä. Lisäsin kaikkiin makron ohjelmiin, jotka tekevät toimintoja SAP:issa, Subproseduurin SAP\_Logon, jota käsittelin 20.9. sivulla 19–20. Näin kyseisiä makron osia ajaessa tarvitsee olla vain Sap Logon -ikkuna auki ja makro avaa oikean moduulin itse.

### **Viikon 8 yhteenveto**

Tällä viikolla sain makroa kehitettyä paljon. Se toimii tässä vaiheessa jo kokonaisuudessaan, mutta tarvitsee vielä hienosäätöä. Makrosta valmistui vaiheet, jossa se muokkaa toisen lopullisen Excel-tiedoston valmiiksi ”TVLV Avoimet erät”. Makro tuo tiedostoon vuosittain haetut TV/LV-tilien avoimet rivit. Muokkaa välilehdet lopulliseen muotoon poistamalla ylimääräiset rivit, lisäämällä välisummat ostotilauksiin ja muodostamalla otsikkorivit. Tämän jälkeen makro luo sähköpostiluonnokset ajajan omaan sähköpostiin valmiina lähetettäväksi. Sähköposteissa on liitteenä luotu TVLV Avoimet erät -tiedosto ja yhdessä on raportoitu myös tasauksen summat ja niiden tulos- ja varastovaikutus. Ajajalle jää tehtäväksi tarkistaa sähköpostien ja liitteiden oikeellisuus.

Lisäksi makro tallentaa nyt myös toisen lopullisen Excel-tiedoston ”\*\*\*\* TVLV erät alle 1000e vel. tai 300e hyv. yli 6 kk (kuluva päivä)”, joka sisältää kaikki makron tekemät välilehdet, kuten tasatut erät ja niistä syntyneet tulos- ja varastovaikutus. Tämä vaatii vielä hiomista sillä tässä vaiheessa makro ei vielä vie manuaalisesti luotuja välilehtiä kyseiseen Excel-tiedostoon.

Yritys X:n IT-tuki korjasi ongelman, jota käsittelin viikolla 2 ja sanoin palaavani asiaan. Ongelmassa en saanut makroa kirjautumaan itse oikeaan SAP-moduuliin, mutta selvisi, että ongelma olikin minulla ja nyt korjauksen jälkeen moduuliin ei tarvitse kirjautua manuaalisesti. Tämän jälkeen lisäsin makron kaikkiin SAP:ia käytettäviin kohtiin koodia, jolla makro avaa oikean moduulin itse. Tämä on sama koodi, jota olen hyödyntänyt työssä aikaisemmin SAP-hakuja tehdessä.

## 5.9 Viikko 9

### **Maanantai 6.11.**

Lähdin korjaamaan viime viikolla tehtyä kohtaa makrossa, joka vie välilehdet uuteen työkirjaan ja tallentaa sen omana tiedostonaan. Nauhoittamalla tehtävän syntyi koodi, missä makro valitsee välilehdet nimien perusteella ja näin manuaalisesti lisätyt välilehdet eivät siirtyisi uuteen tiedostoon. Löysin googlettamalla ratkaisun tähän ongelmaan. Lisäsin koodiin komennon: "ThisWorkbook.Sheets.Copy", joka kopioi ja vie kaikki makrotöykirjan välilehdet uuteen tiedostoon. Lisäsin myös koodia, jolla ensimmäinen eli makro-välilehti poistetaan ja sitten tiedosto tallennetaan nimellä: "\*\*\*\* TVLV erät alle 1000e vel. tai 300e hyv. yli 6 kk (kuluva päivä)". Tarkoituksena on siis tallentaa makrolla luotu tasattujen ostotilauksien -tiedosto .xlsx muodossa, joka ei sisällä itse makroa. (VBA: ThisWorkbook... 2021.)

Tämän jälkeen testasin makron toiminnan ajamalla sen kokonaisuudessaan nappeja käyttäen. Huomasin, että tulos- ja varastovaikutuksen luvut SAP:ista tuodessa on joillain riveillä väärässä muodossa, eikä näin ollen makro laske niitä rivejä mukaan tehdessään SUMMA-funktioita. Lisäsin makron koodiin kohdan, jossa se muuttaa nämä arvot luvuiksi käyttäen teksti sarakkeisiin -toimintoa. Lisäksi makron loppuvaiheilla, liitetiedoston muokkauksessa sähköposteja varten makro muokkasi väärää välilehteä. Lisäsin koodia, joka valitsee ensimmäisen välilehden ennen näiden muokkausten tekoa. Tämän jälkeen makro toimi moitteetta.



## Torstai 9.11

Lähdin vielä tekemään prosessia manuaalisesti ja vertaamaan vaiheiden tuloksia makron kanssa. Vielä ennen sitä päätin muokata makron koodia Osa\_1-proseduurin kohdassa, jossa se vertailee rivien päivämäärää kuluvaan päivään ja jos ero on yli tai alle 183 päivää se poistaa rivit riippuen käsitelläänkö yli 6 kk vai alle 6 kk -välilehteä. Manuaalisesti tämä tehdään eri tavalla ja suodatetaan rivit vain kuukausitasolla yli 6 kuukautta tai alle 6 kuukautta, eikä näin ollen saman kuukauden tapahtumia päädy eri välilehdille. Käsittelin tätä vaihetta työssäni 28.9. sivulla 22–23. Halusin, että makro suodattaa tapahtumat samalla tavalla kuin manuaalisesti on tehty, joka myös helpottaa, kun seuraavaksi vertailen makron tuottamaa aineistoa manuaalisesti tuotettuun.

Loin uuden muuttujan kuukaudet: "Dim kuukaudet As Date." Tämän jälkeen määrittelin muuttujan: "kuukaudet = DateSerial(year(Date), Month(Date) – 6, 0)". Kuukaudet-muuttujan arvo on nyt kuluvasta päivästä kuusi kuukautta taaksepäin ja sitä edeltävän kuukauden viimeinen päivä. Eli marraskuussa kuukaudet-muuttuja sai arvon 30.4.2023. Tämän jälkeen muokkasin kohtaa, jossa vertaillaan rivin päivämäärää kuluvaan päivään (Kuva 6). Muokkasin sen niin, että se vertailee rivin päivämäärää luotuun kuukaudet-muuttujaan ja poistaa rivin, mikäli rivin päivä on pienempi tai suurempi kuin kuukaudet-muuttuja riippuen kumpi välilehdistä alle 6 kk/yli 6 kk on käsiteltävänä. (Kuva 13).

```

Do

    dateSearched = ActiveCell 'define variable dateSearched to be active cell's value

    If DateDiff("d", dateSearched, kuukaudet) < 0 Then 'if date difference is less than
ActiveCell.EntireRow.Delete 'delete row
    Else
ActiveCell.Offset(1, 0).Select 'otherwise move to next row
    End If

Loop Until ActiveCell = "" 'continue loop until active cell is empty

```

Kuva 13. Muokattu koodi, jolla aktiivisen solun päivämäärää vertaillaan kuukaudet-muuttujaan ja poistetaan mikäli Then-lause on tosi. Muulloin siirtyy seuraavaan riviin.

Seuraavaksi lähdin ajamaan makroa samanaikaisesti, kun suoritin prosessia manuaalisesti. Vertailin kaikkia luotuja välilehtiä, että makrolla tehdyt aineistot sisältävät samat tiedot eikä se tee missään vaiheessa virheitä. Huomasinkin ensim-

mäisen virheen nopeasti, kun avoimia tapahtumia oli vähemmän makron ajamassa listauksessa, kuin manuaalisesti ajamassani. Koodista selvisi nopeasti, että olin unohtanut lisätä SAP-hakuun kaikkien TV/LV-tilien numerot, joten nauhoitin nopeasti SAP Scriptin, jossa lisään hakuun myös puuttuneet tilinumerot. Lisäsin Scriptin makronkoodiin kohtaan, jossa avoimia rivejä haetaan ja testasin, että se toimii. Tämän jälkeen jouduin muokata koodia hieman lisää, sillä useampia tilejä hakiessa, tuodussa luettelossa on enemmän ylimääräisiä rivejä luettelon lopussa, joka vaikuttaa esimerkiksi siihen, että kuvassa 13 oleva koodi jää virheeseen siinä pisteessä, kun se etenee ylimääräiseen riviin, jossa tietona ei olekaan päivämäärä. Lisäsin rivien poistoja vaiheisiin, jossa tuotiin ja muokattiin Avoimet-tiedostoa. Näitä vaiheita käsittelin 25.9. sivulla 21–22 ja 25.10. sivulla 32. Näiden muokkauksien jälkeen ajoin makron kokonaisuudessaan ja tein prosessin myös manuaalisesti. Vertailin kaikkia luotuja aineistoja ja ne olivat identtisiä toistensa kanssa. Makro siis toimii suunnitellusti.

### **Viikon 9 yhteenveto**

Makro oli yhdeksännen viikon jälkeen täysin valmis. Tällä viikolla sain korjattua kaikki makron ongelmat ja tekemään vielä viimeisiä hiomisia. Lisäsin makroon kaikkien osien ajamista varten napit, nimesin ne, järjestelin ne ja tein koodia, millä aina seuraavaksi ajettavan osan nappi on vihreänä ja muut punaisena. Lisäksi lisäsin samoihin kohtiin koodia, missä jokaisen makron vaiheen jälkeen tulee muutaman lauseen ohjetekstit, mitkä sisältävät lyhyen tiedon seuraavasta ajosta tai tehtävästä mikä ajajan pitää suorittaa itse ennen seuraavaa ajoa, kuten taksaukset.

Muokkasin myös kohdan, jossa ostotilaukset jaettiin omille välilehdilleen yli 6 kk tai alle 6 kk sen mukaan onko rivin kirjauspäivämäärä yli/alle 183 päivää kuluva päivästä. Tämä tyyli erosi manuaalisesta tekemisestä, minkä takia päätin vielä muuttaa tämän vastaamaan sitä. Nyt makro vertailee rivien kirjauspäiviä uuteen muuttujaan, joka on kuusi kuukautta taaksepäin kuluva päivästä ja siitä edellisen kuukauden viimeinen päivä, ja poistaa rivit sen mukaan onko ero yli vai alle siitä päivästä.

## 5.10 Yhteenveto koko tarkastelujaksolta

Makron kehittäminen kesti yhdeksän viikkoa, jonka aikana sitä kehitettiin 17 päivänä. Kehitin makroa töiden ja opiskelujen ohella, ja osana päivistä pystyin käyttämään enemmän aikaa kehitystyöhön ja osana taas vähemmän.

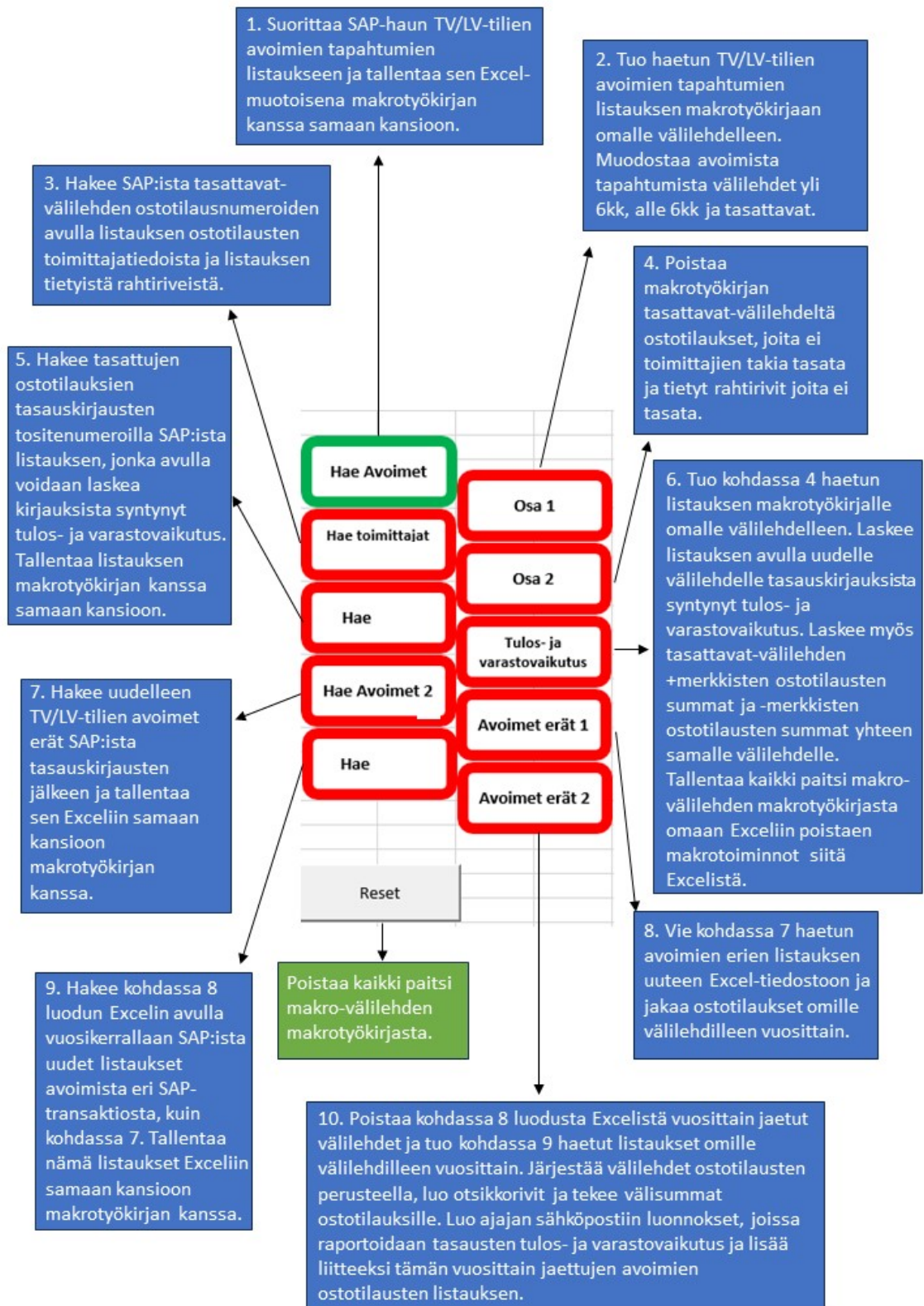
Makrosta valmistuivat näiden yhdeksän viikon aikana kaikki sille toivotut toiminnot. Makro tekee tässä työssä käsiteltävänä olleen prosessin kaikki vaiheet lukuun ottamatta tasauskirjauksia ja sähköpostien lähetystä. Kirjauksia varten makro kuitenkin luo valmiin luettelon kirjattavista ostotilauksista ja sähköposteja varten se luo valmiit luonnokset makron ajajan sähköpostiin. Prosessin tekijän tehtäväksi jää makron luomien luetteloiden ja sähköpostien oikeellisuuden tarkistus ja kirjausten tekeminen.

Koska makro valmistui ajallaan, pääsimme prosessin vakituisen tekijän kanssa suorittamaan se makroa hyödyntäen marraskuun osalta. Otin sekuntikellolla aikaa, kuinka pitkään prosessin suoritus kesti kokonaisuudessaan. Makro pysähtyi kerran virheeseen, joka johtui SAP:issa olevista asetuksien eroista minun ja ajajan välillä. Ongelma korjaantui vaihtamalla häneltä asetuksia. Toisen kerran pysähtyi SAP-hakujen kohdalla, joka korjaantui nopeasti poistamalla ylimääräisen toiminnon sisältävän rivin koodista. Näiden jälkeen makro saatiin ajettua ja prosessi suoritettua kokonaisuudessaan sitä hyödyntäen. Tähän kului aikaa 40 minuuttia, joka on parempi kuin tavoitteeksi asetettu 45 minuuttia. Lisäksi vakituisen tekijän mukaan tällä suorituskerralla, prosessissa oli normaalista poikkeavaa selvitystyötä, miksi sen suorittaminen kesti kauemmin kuin normaalisti.

Makron kehittäminen on ollut pitkä ja opettavainen prosessi. Olen helpottunut ja ylpeä siitä, että asetettuihin tavoitteisiin päästiin ja onnistuin kehittämään makron, joka suorittaa kaikki sille suunnitellut toiminnot onnistuneesti.

## 6 VALMIIN MAKRON KUVAUS

Valmis makro on jaettu osiin, ja se suoritetaan makrotyökirjan ensimmäisellä välilehdellä olevista painikkeista, ylhäältä alaspäin etenevässä järjestyksessä. Ennen makron ajamista on hyvä painaa Reset-painiketta, joka poistaa kaikki paitsi makro-välilehden makrotyökirjasta ja muuttaa ensimmäiseksi ajettavan osan painikkeen vihreäksi. Jokaisen onnistuneen ajon jälkeen seuraavan osion painike muuttuu vihreäksi ja muut painikkeet pysyvät tai muuttuvat punaiseksi. Tämän avulla pystyy helpommin hahmottamaan, mikä osa on ajettava seuraavaksi. Lisäksi välilehdellä on lyhyt ohjeistus ja kuvaus makron ajajalle, seuraavaksi ajettavan osan toiminnasta. Myös nämä ohjeistukset ja kuvaukset muuttuvat jokaisen osan ajon jälkeen. Painikkeet on jaettu niin, että kaikkien SAP-hakuja käyttävien osien painikkeet ovat vasemmalla ja loput oikealla toisiinsa katsottuna. Alla olen kuvannut jokaisen painikkeesta suoritettavan makron osan toiminnot lyhyesti (Kuvio 1). (Liite 2.)



KUVIO 1. Kuvaus kaikista makron osista lyhyesti.



## 7 POHDINTA

Tämän opinnäytetyön tavoitteena oli automatisoida Yritys X:n TV/LV-tilien ta-saus- ja raportointiprosessi Excel-makroa hyödyntäen. Automatisointi mahdollis-taa prosessin tehokkaamman suorittamisen ja vapauttaa tekijän nopeammin mui-hin tehtäviin. Konkreettiseksi tavoitteeksi Yritys X:n kanssa asetettiin, että koko prosessi saataisiin automatisoinnin avulla suoritettua 45 minuutissa.

Opinnäytetyö oli päiväkirjamuotoinen, ja sen tarkastelujakso oli yhdeksän viikkoa. Viikkojen ajalta työssä raportoitiin makronkehityksen edistymistä ja eri työvai-heita. Teoreettinen tieto on sisällytetty päiväkirjaraportointi -osioon, minkä lisäksi työssä on kappale, jossa Excel-makroja ja niiden toimintaa käydään lyhyesti läpi. Työ tehtiin yritykselle, joka on pidetty nimettömänä, ja siitä on piilotettu tunnista-miseen johtavia tietoja sekä tiedot, jotka kuuluvat tekijän vaitiolovelvollisuussopi-muksen alle.

Lähdin työhön ilman aikaisempaa kokemusta VBA-ohjelmoinnista tai makrojen luonnista. Tästä syystä aikataulun ja työnlaajuuden arvioiminen etukäteen oli haastavaa. Opinnäytetyössä kuitenkin päästiin asetettuihin tavoitteisiin. Prosessi suoritettiin sen vakituisen tekijän kanssa kokonaisuudessaan makron avulla mi-taten samalla suoritukseen kulunutta aikaa. Prosessi suoritettiin 40 minuutissa, joka on 5 minuuttia alle konkreettiseksi asetetun tavoitteen. Tästä ajasta noin 8 minuuttia kului makron ajamiseen ja loput tasauskirjausten tekoon ja makron luo-mien listojen oikeellisuuden tarkistamiseen. Tällä prosessin suorituskerralla ta-sattavat ostotilaukset vaativat normaalista poikkeavaa selvitystyötä, minkä vuoksi prosessin suorittamiseen kului enemmän aikaa. Jatkossa prosessi on mahdol-lista suorittaa vielä nopeammin, kun selvitystyöhön kuluva aika vastaa normaalia tilannetta ja makro on tutumpi sen ajajalle.

Opinnäytetyöstä syntynyt hyöty on suurempi kuin sille asetetut tavoitteet. Tekijän osaaminen makrojen kehityksen ja prosessien automatisoinnin suhteen kasvoi huomattavasti, mistä voi olla hyötyä jatkossa niin Yritys X:lle, kuin tekijälle itsel-leen. Luodun makron koodia on myös mahdollista hyödyntää Yritys X:n sisällä muiden prosessien automatisoinnissa, mikäli tarvetta.

## LÄHTEET

Excel VBA attaching files from SharePoint adding %20 for spaces in filenames. 2023. Verkkosivu. Viitattu 2.11.2023. <https://stackoverflow.com/questions/77152732/excel-vba-attaching-files-from-sharepoint-adding-20-for-spaces-in-filenames>

Excel VBA - total line to add border on top and bottom only. 2012. Verkkosivu. Viitattu 14.11.2023. <https://stackoverflow.com/questions/9731587/excel-vba-total-line-to-add-border-on-top-and-bottom-only>

How to Combine the ROUND and SUM Functions in Excel. 2022. Verkkosivu. Viitattu 2.11.2023. <https://www.lifewire.com/combine-round-and-sum-functions-in-excel-4178187>

Kelly, P. n.d. VBA Error Handling – A Complete Guide. Verkkosivu. Viitattu 26.10.2023. <https://excelmacromastery.com/vba-error-handling/>

Makrosta Office-ohjelmien käytön tehostajana. n.d. Verkkosivu. Viitattu 14.11.2023. <https://www.excel-ohjelmointi.fi/makrot.html>

Organisaation makro-ohjeet. 2023. Yritys X. Viitattu 12.9.2023. Vaatii käyttöoikeuden.

Organisaation makrotyökirja. 2023. Yritys X. Viitattu 18.9.2023. Vaatii käyttöoikeuden

Taanila A. 2013. EXCEL VBA-OHJELMOINTI. Pdf-dokumentti. Viitattu 12.9.2023. <http://myy.haaga-helia.fi/~taaak/vba/vba.pdf> viitattu 12.9.2023

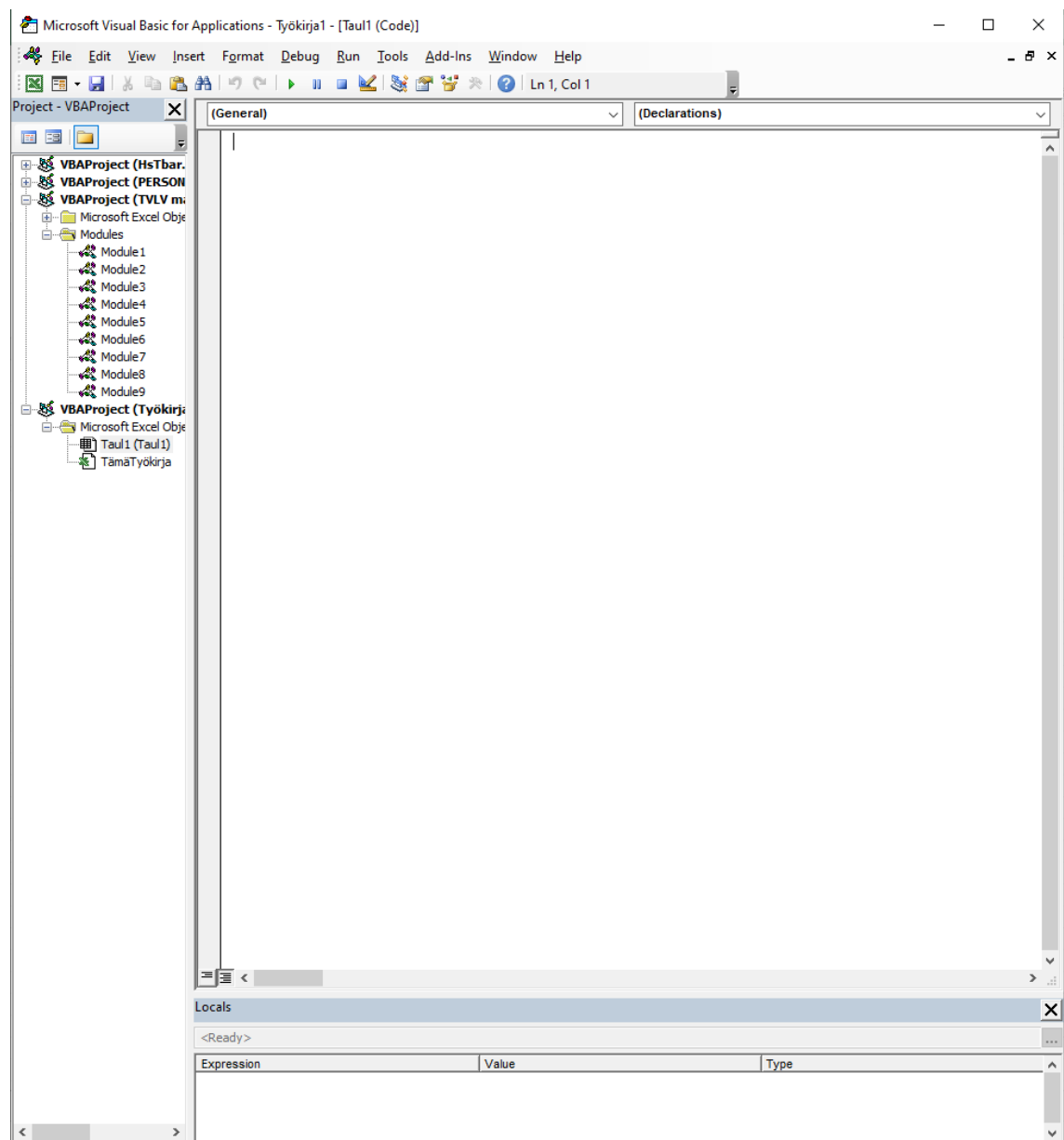
Troubleshooting connection Excel VBA to SAP. 2022. Verkkosivu. Viitattu 14.9.2023. <https://stackoverflow.com/questions/70551806/troubleshooting-connection-excel-vba-to-sap>

Using Sort in VBA for a Range that Changes. 2014. Verkkosivu. Viitattu 25.9.2023. <https://stackoverflow.com/questions/22174908/using-sort-in-vba-for-a-range-that-changes>

VBA: ThisWorkbook.SaveCopyAs will not allow xlsx file format. 2021. Verkkosivu. Viitattu 6.11.2023. <https://www.mrexcel.com/board/threads/vba-thisworkbook-savecopyas-will-not-allow-xlsx-file-format.1165358/>

## LIITTEET

### Liite 1. VBA-editori Excelissä



## Liite 2. Excel-makron etusivu.

