LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

# ETL RĪKU FUNKCIONALITĀTES PIELĀGOŠANAS IESPĒJAS DATU AVOTU IZMAIŅĀM

BAKALAURA DARBS

Autors: **Atis Ēriks Dumpis**
Studenta apliecības Nr.: ad16035
Darba vadītājs: Dr.dat. Darja Solodovņikova

RĪGA 2020

# ANOTĀCIJA

Šis bakalaura darbs “ETL rīku funkcionalitātes pielāgošanas iespējas datu avotu izmaiņām” tika izstrādāts ar mērķi dot lasītājam ieskatu, kas ir ETL rīki, un kādas ir iespējamās opcijas rīka funkcionalitātes pielāgošanai lietotāja interesēs.

Darba teorētiskajā daļā ir aprakstīts, kas ir ETL process, kā tas funkcionē un kādas ir tā galvenās trīs daļas

Darba praktiskajā daļā ir veidoti rīku funkcionalitātes pielāgošana dažādās kļūdas iespējamajās situācijās.

Darba noslēgumā apkopoti ETL rīku testēšanas rezultāti un apkopoti kādi secinājumi gūti darba izstrādes laikā.

Atslēgvārdi: ETL rīks, dati, transformācijas, ETL process.

# ABSTRACT

CUSTOMIZING ETL TOOL FUNCTIONALITY TO CHANGES IN DATA SOURCES

This Bachelor thesis „ Customizing ETL tool functionality to changes in data sources" was developed to explore what ETL tools are and what possible options are for customizing the functionality of the tool in the user's interest.

The theoretical part of the work describes what the ETL process is, how it works and what its main three parts are

In practical part author's customized the functionality of the tools in different situations of error.

In the conclusion there is a summary of the ETL tools and the conclusions reached during the process of this bachelor theses elaboration.



Keywords: ETL tools, data, transformations, ETL process.

# SATURS

# VĀRDNĪCA

| Nr. | Atslēgvārds | Skaidrojums |
|---|---|---|
| 1. | XLSX | (angl. – excel spreadsheet) Datnes formāts priekš tabulveida datiem ko izstrādājis Microsoft priekš darbībasd ar Microsoft Excel. |
| 2. | JSON | (angl. – JavaScript object notation) ir atvērta standarta datu formāts, kas izmanto cilvēkam salasāmu tekstu datu uzglabāšanai. JSON datu tips ir neatkarīgs no valodām un attīstīts no JavaScript. |
| 3. | CSV | (angl. – comma-separated values) ir ar komatiem atdalītu datu formāts, kas satur tabulveida datus, kur kolonas nodala komati un rindas ir tādas pašas kā datnes izkārtojumā. |
| 4. | Java | Java ir augta līmeņa programmēšanas valoda. Ir ļoti strikti objektorientēta programmēšanas valoda. |
| 5. | SQL | (angl. – structured query language) ir skriptu valoda, kas paredzēta datu bāžu datu piekļuvei un manipulācijai. |
| 6. | JavaScript | JavaScript ir programmēšanas valoda visbiežāk izmantota interneta lietotņu izveidei. |
| 7. | ETL | (angl. – extract transform load) ir process, kurā dati tiek savākti transformēti un ielādēti glabātuvē. |

# IEVADS

ETL rīki mūsdienu kultūrai piedāvā vieglu un ātri apgūstamu veidu, kā rūpēties par datiem, kas tipiski rodas teju visās nozarēs un darba vidēs. Šie rīki kaut arī iekļauj daudz noderīgu un ērtu funkcionalitāšu, tomēr ir situācijas, kur to darbība var tikt ierobežota ar ieejas datu kļūdām, kas var apdraudēt visu datu savākšanas procesu.

Šī darba izstrādes laikā tika pētīts vai ir iespējams pievienot rīkiem kļūdas iespējas samazinošas metodes.

Darba autors vēlas pētīt, vai ir iespējams modificēt vai papildināt ETL rīkus ar funkcionalitāti, kas spētu novērst datu trūkuma radītas kļūdas sistēmā.

Autora izvirzītā hipotēze ir, ka lielākajai daļai rīku, kuri paredzēti plašai lietošanai būs iekļauta sistēma kā modificēt rīku un pielāgot to lietotāja vajadzībai.

Darba mērķis ir vairākos testam izvēlētos rīkos izstrādāt papildinošus noteikumus, kas spētu novērst kļūdas situācijas, kas rodas ieejas datu avota nevēlamās izmaiņās, tipiski datu trūkuma gadījumos.

Lai sasniegtu darba mērķi, tika izvirzīti šādi uzdevumi:

- Iepazīties ar literatūru par ETL rīkiem un izprast to darbību.
- Aplūkot ETL rīku klāstu un izpētīt kādas funkcionalitātes tie spēj sniegt
- Izvēlēties ETL rīku grupu, kuriem mēģināt veikt modifikācijas iespējamo procesa kļūdu novēršanai.
- Testa grupas ETL rīkos mēģināt ieviest noteikumus, kas novērstu iespējamās ETL procesa kļūdas.

Autors darba izstrādes laikā veica gan teorētiskus pētījumus (literatūra), gan eksperimentālus (veidot nosacījumus rīkos, kas novērš kļūdas).

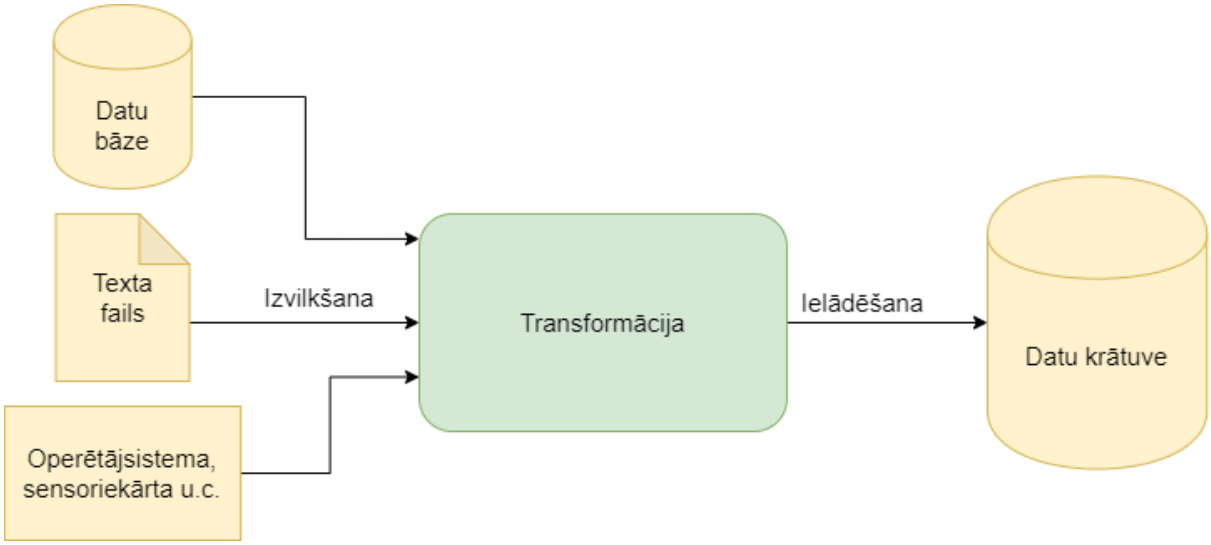
Kā faktoloģiskie materiālu avoti tiek izmantotas grāmatas, zinātniskie raksti un interneta resursi.

Pirmajā nodaļā ir dots skaidrojums, kas ir ETL rīki to darbības process kas paskaidrots, sadalot tā trīs daļās.
Otrajā nodaļā aprakstīts, kādas ir potenciālās iespējas pielāgot ETL rīku darbību un kuri rīki izvēlēti un kādēļ.

Trešajā nodaļā aprakstīts rīku veiktās modifikācijas un testpiemēri, kas veikti modifikāciju testēšanai.

# 1. ETL RĪKU DARBĪBA

ETL jeb “extract, transform, load” apzīmē trīs galvenos procesus datu izvilkšanu, transformāciju un ielādēšanu krātuvē, ko var aplūkot 1.1. attēlā. Mūsdienu kompānijām ar privātām datubāzēm pienāk brīdis, kad vainu tām nepieciešams iet laikam līdzi un sākt izmantot modernāku datu glabāšanas metodi vai tās ietver sevī jaunu datu avotu, kurš nav perfekti saderīgs ar pašreizējo datu bāzi. Šādās situācijās tiek izmantotas datu noliktavas, kuras pašas par sevi nav datu vācējas bet izmantojot ETL rīkus spēj apkopot datu avotus vienā kopējā noliktavā, kur šie dati tiek sakārtoti un organizēti uztveramākā formātā.[10,1]




*1.1. att. ETL procesa shematisks attēlojums*

ETL rīku nepieciešamība izveidojās tādēļ, ka bieži dati, kas nepieciešami var nākt no dažādiem avotiem šo situāciju piemēri:

- Atšķirīgām datu bāzēm – ja līdz kādam brīdim firmā izmantota Microsoft datu bāze, bet firmas lēmumu dēļ notiek pāreja uz Oracle datu bāzi vai iespējams notiek paralēla darbība ar abām.
- Atšķirīgām operētājsistēmām – Ja dati tiek ievākti no sensoru iekārtas, kas bāzēta uz Linux operētājsistēmu, savukārt pārējie dati nāk no Windows operētājsistēmas datnēm.
- Aparatūras atšķirības – Ja izmantotas tiek iekārtas kas pašas neapstrādā datus vai no to darbības tiek padoti statiski dati apstrādei.

Šie rīki ir pielāgoti vairāku informācijas avotu apvienošanai tajos var noteikt vajadzīgās attiecības, starp datiem veidojot vienojošus identifikatorus starp iepriekšēji

nesaistītiem datiem. Iespējams arī analizēt dažādus datu ievades veidus kā JSON formātu vai SQL pieprasījuma rezultātus tādējādi iekļaujot plašu datu atpazīšanas spektru.[1]

## 1.1. E jeb izvilkšana

Rīkā tiek definēti informācijas avoti, no kuriem tiks iegūti dati. Iekš rīka ir iespējams atzīmēt, kāda tipa vai veida dati tie būs, jo iespējams ka dati tiek iegūti no kādas datu bāzes vai vairākām, kas radītu datu avotu atšķirības, kā arī iespējams ka datu avoti ir tieši teksta bāzēta informācija, ko jāsadala tālāk nepieciešamajā informācijā, kā arī iespējams arī citi ievaddatu formāti JSON, log datnes, CSV, XLSX un citas.

Rīkā tiek ievadīts arī laika periods, kad uzsākt darbību un ievākt datus no visiem norādītajiem avotiem. Datu savākšanas procesā tiek pārbaudītas visas definētās vietas un adreses pēc ievācamiem datiem. Datu ievākšana iespējami apdraud datu avotu drošību ja ievākšanas laiks nav saskaņots ar paša avota darbību, piemēram datubāzes process ar savācamajiem datiem var apturēt datu vākšanas procesu vai arī apturēt datu bāzes darbību nevēlamā veidā.[1,2]

### 1.1.1. Datu savākšanas norise

Datu savākšana notiek pēc rīkā definētā avotu saraksta. Tipiski rīkā jāveic avotu definēšana norādot adresi vai lokāciju kur pieejami dati, kā arī kāda tipa dati tie ir datu bāzes gadījumā būtu jānorāda attiecīgās izvēlētās tabulas un datu lauki kas atlasīti, bet pie cita veida datu avotiem būtu jānorāda atbilstošs datu interpretācijas formāts kā JSON, .txt un citi Datu savākšana var notikt ar pilnu datu savākšanu kur tiek ievākti dati no visiem minētajiem avotiem paņemot viesus iespējamos datus, iespējams arī savākt datus daļēji izvēloties tikai jaunākos datus vai datus no kāda laika perioda tādējādi nedublējot datus kas iespējams atlasīti iepriekšējās datu savākšanas reizēs.

Vairākas no datu pārbaudēm tiek veiktas savākšanas solī. Datu avota salīdzinājums ar rīkā definēto struktūru tiek veikts, lai piefiksētu iespējamās izmaiņas vai problēmas, kas varētu rasties, mēģinot apstrādāt neeksistējošus datus. Tiek arī pārbaudīts, vai dati, kas jāņem, kā datu avots nav atkritum datnes ar informāciju, kas nav rīkā interpretējama. Ja rīkā pie datu avota ir norādīts arī datu tips, tiks pārbaudīts, vai tas atbilst ar sastaptu datnes datu tipu, piemēram, vai norādot, ka datnes, kas jāņem, ir ar tipu txt bet vienīgā datne norādītajā lokācijā ir csv formātā. Dažkārt tiek pārbaudīts arī, vai datnes nav, koruptētas vai fragmentēti tādējādi padarot to saturošo informāciju nelietojamu. Kā arī šinī solī iespējams pārbaudīt, vai ievācamajiem datiem ir visi atbilstošie identificējošās vērtības, kas vainu reprezentē datu ierakstus tālākās tabulās vai arī visu datu kopu kā pieņemamu datu avotu.[1,2]

## 1.2. T jeb transformācija

Transformācijas solī notiek vairums datu pārveidošanas un pārbaudes. Datiem uzreiz pēc to ielādēšanas vēl nav nekādas kopīgas definētas saistības, ja tie nākuši no dažādiem datu avotiem un šinī solī tie tiek izdalīti pārveidoti atbilstoši vainu formātam, vai saturam un tie tiek sasaistīti plašākā datu kartē, veidojot saiknes.[1,2]

### 1.2.1. Iespējamās darbības ar datiem pārveidošanas solī

Ar datiem, kas ir, ielādēti no dažādajiem datu avotiem, ir iespējams veikt darbības to modifikācijai. Bieži dati, kas iegūti no log datnēm vai arī ja to datu saturs ir ar vērtībām, kas nav vēlamas kopējam iznākuma datu apjomam tad šinī solī var pielāgot sistēmas darbību atbilstoši nepieciešamībām. Galvenās funkcionalitātes ko dod šis solis ir:[1]

**Tīrīšana –** apskatot ielādējamos datus, iespējams rediģēt, kā tie tiks ielādēti tālāk un situācijai specifiski noņemt tukšos datu laukus ja tādi, izveidojušies pie datu oriģinālās saglabāšanas, vai arī mainīt datu lauka tipu uz piemērotāku ar mazāku atmiņas datu lauka vienību. Piemēram, dati kas ielādēti no SQL datu bāzes ar tipu char[1]

**Filtrēšana –** kā jau minēts dati, kas ielādēti var saturēt lieku informāciju, kas nav vajadzīga ielādēt datu glabātuvē. Situācijās, kur ielādētie dati ir jau ielādēti un nav nepieciešama, atkārtota to ielādē vai ja datu lauki nespēj sniegt nekādu noderīgu vai izmantojamu informāciju, tad var izvēlēties attiecīgos laukus nelādēt.[1]

**Bagātināšana –** ar datiem iespējams veikt arī kalkulācijas šinī solī. Ja ir nepieciešama kāda biznesa loģika, kas apvienotu vairāku datu lauku doto informāciju piemēram vidējās vērtības iegūšana vai kopējo summu laika griezumā, tad šis solis ir tas, kurā jāievieš šīs formulas. Pēc nepieciešamības var arī ieviest laika atzīmes, kad veiktas darbības, kas tiek iegūtas ne no ārējiem datu avotiem bet gan no pašas sistēmas.[1]

**Dalīšana –** datus kas apzīmē vairākas zemāka līmeņa datu vienības var sadalīt vainu matemātiski ar kalkulācijām kā laika vienības nošķirot datumu no laika vai arī ar vienkāršākā metodēm kur tekstuālas vērtības, kā piemēram, cilvēka vārds un uzvārds tiek sadalītas šajās daļās atsevišķi ja nepieciešams. Dalīšanas darbība attiecināma arī uz kopējiem datu avotiem, kas ielādēti teksta formātā bez jau iepriekš definētas kārtības vai definīcijas, kas parasti sastopami datu bāžu datnēs. Šādiem tekstuāliem datu avotiem iespējams definēt struktūru pēc, kā tie tiek sadalīti tādējādi, atdalot iespējami dažādo informāciju.[1]

**Apvienošana –** ja ielādētajos datos nav nepieciešamība no individuāliem laukiem, piemēram, vārda lauka un uzvārda lauka nodalīšanai tad tos var apvienot, iegūstot tālāk

lādējamu tikai vienu lauku ar pilnu vārdu. Par apvienošanu var arī dēvēt matemātiskās darbības, kas tiek veiktas ar datiem, iegūstot kompleksus vērtību aprēķinus.[1]

## 1.3. L jeb ielādēšana

Lādēšana ir pēdējais no procesiem, kas notiek ETL rīkos. Tipiski dati, kas ielādēti pārbaudīti un transformēti, tiek ielādēti datu noliktavās sakarā ar to lielo apjomu, kas veidojas no regulāras datu ievākšanas. Datu ielādēšanai parasti ir pārbaudes, vai process noritējis veiksmīgi, jo situācijā kur dati nav ielādēti vai process apstājies, kādā no ielādes posmiem tas tiek palaists no jauna vai arī no pēdējā pieejamā momenta procesā, kurā iespējama darbības atsākšana. Datu ielādes restartēšanu var konfigurēt, pielāgojot katras sistēmas un datu kopas specifikācijai. Datu ielāde parasti darbojas ar liela izmēra datnēm un informācijas apjomu līdz ar to ir nepieciešama augsta līmeņa efektivizācija, un vajadzētu izvairīties no dažādām liekām darbībām vai datu vienībām, jo kopumā tās summējas un noslogo sistēmu svarīgajā ielādes procesā.[1,2]

### 1.3.1. Datu ielādēšanas veidi

Datu ielādēšanai ir dažādi veidi, kas notiek pie konkrētām situācijām attiecīgi tam, kas tiek nodefinēts rīka darbībā. Datu ielādēšanas var konfigurēt rīku darbībā, kontrolējot datu plūsmu, un kādi dati tiek pieņemti un saglabāti datu krātuvē.[1]

**Sākotnējā ielādēšana** – datu ielādēšana, kas notiek tukšā datu glabātuves sistēmā tiek ielādēti visi pieejamie dati, nepārrakstot nevienu ierakstu, jo tādu vēl nav sistēmā. Pie ielādēšanas nav nepieciešama glabātuvē esošo datu pārbaude sakarā ar to, ka glabātuve pirms ielādēšanas ir pilnībā tukša.[1]

**Pakāpeniskā ielādēšana** – ielādēšanas procesā tiek padoti tikai tie dati, kas mainījušies no tiem, kas ir jau pieejami sistēmā. Tiek ielādēta mainītā informācija, kā arī jaunā, ja dati jau nav pieejami krātuvē.[1]

**Pilnā pārlādēšana** – veicot pilnu pārlādēšanu tiek izdzēsti dati, kas bijuši vienā vai vairākās tabulās datu glabātuvē un tiek ielādēti pilnīgi jauni dati.[1]

# 2. PAPILDNOTEIKUMU IEVIEŠANA

Kaut arī liela funkcionalitātes bāze ir jau nosegta ar rīku pamatdarbību, tomēr var sastapties ar samērā biežu parādību, kad dati, kas paredzēti savākšanai un pārveidei, ir vainu nepilnīgi vai arī pārveidoti, tā ka nav vairs derīgi pie tanī brīdī aktīvās datu ievākšanas konfigurācijas. Tipiski rīki atrisina šādu situāciju ar dažādām iespējamām darbībām, piemēram, datu ievākšanas atkārtotu palaišanu, problemātiskās datnes nesavākšanu radot informācijas trūkumu tālākā datu pārveidē vai datu vākšanas un kopējā datu nogādāšanas procesa apturēšanu un neturpināšanu, līdz datu trūkuma vai neatbilstības situācija tikusi novērsta. Šādās situācijās būtu nepieciešams ieviest papildnosacījumus vai sistēmas atbildes darbību, kas ļautu reaģēt atbilstoši lietotāja vēlmēm. Tādēļ nepieciešams noskaidrot kādas ir tipiskākās situācijas kurās iespējams modificēt vai papildināt rīku funkcionalitāti.

## 2.1. Paredzamās funkcionalitātes papildināšanas iespējas

Analizējot dažādos ETL rīkus, var novērot, ka lielumā rīkos ir iekļauta kāda līmeņa papildkoda iekļaušanas opcija, kas varētu būt viena no iespējamajām rīku papildināšanas iespējā. Daži rīki pieejami arī to pamata koda formā, kas arī atļautu iespēju modificēt rīka darbību. Kā arī paredzami būtu, kā šāda tipa situāciju risinājumi būtu sastopami arī rīku konfigurācijā iekš paša rīka vai tā konfigurācijas datnēm.

### 2.1.1. ETL rīkā iebūvēta opcija nosacījumu pievienošanai

Pieļaujot ka šāda tipa situācija, kur ETL rīkā iekļautā funkcionalitāte nav pietiekama lietotāja vajadzībām, būtu tipiska darbības videi kas ir ļoti situācijai specifiska ar daudziem elementiem kas nav viegli definējami, kā piemēram, dažāda tipa datnes vai sadarbības sistēmas no kurām jāiegūst apstrādājamās datnes. Tādēļ var uzskatīt, ka rīkos būtu iekļauta kāda veida sistēma vai funkcionalitāte, kas atļautu, papildināt esošo datu apstrādes procesu. Ja šāda sistēma būtu rīkā tad, varētu būt iespēja, ka ar to var arī pievienot papildinājumus nosacījumiem situācijās, kur ir trūkstoša informācija, kas potenciāli izraisītu rīka darbības apstāšanos vai nepilnīgu norisi.

### 2.1.2. ETL rīka manipulācija konfigurācijas datnes

Vairumam programmu un sistēmu, kuras paredzētas plašai lietotāju skaitam tiek iekļauta iespēja konfigurēt sistēmas darbību. Šāda iespēja tiek plaši izmantota, sakarā ar to, ka visas sistēmas izmantošanas pieejas nav vienādas. Dažādiem lietotājiem varētu būt dažādas

vajadzības no sistēmas vainu tas būtu tieši iespējamās funkcionalitātes ziņā ko spēj piedāvāt sistēma, kur nepieciešams kādu no tām atslēgt sakarā ar tās nelietderību situācijā vai nomainīt sistēmas darbības parametrus, piemēram, darbību skaitu ko paredzēts veikt vienā darbības ciklā. Šādā datnē būtu potenciāli iespējams arī pievienot noteikumus vai iespējot funkcionalitāti, kas būtu pielāgota kļūdas situācijas apturēšanai.

### 2.1.3. ETL rīka pamata koda papildināšana

Vairākām programmām, kuras ir paredzētas publiskai piekļuvei tiek publicēts arī sistēmas kods projekta formātā. Ja sistēmai ir brīvi pieejams kods, tad tā papildināšanai ir nepieciešams atrast datu ievākšanas un apkopošanas procesu un papildināt ar attiecīgo funkcionalitātes kodu. Iespēja pastāv, ka manipulēt ar rīka pamat kodu tiešā veidā var arī nepastāvēt tad potenciāli vel varētu būt variants izveidot individuālu procesa kodu, kurš tiktu izsaukts datu ievākšanas solī un ar šo procesu pārbaudīt un pie nepieciešamības pielāgot padotos datus.

## 2.2. Rīku izvēles kritēriji

Lai noskaidrotu vai dažādās rīku funkcionalitātes papildināšanas iespējas ir iespējamas un īstenojamas no rīka lietotāja puses, ir nepieciešams izvēlēties rīku testa grupu, kas sastāvēs no četriem ETL rīkiem. Ir nepieciešams izvēlēties rīkus, kas reprezentētu plašo rīku klāstu tādēļ kā kritērijus, darba autors izvēlējās popularitāti, ko reprezentē rīka lietotāju potenciālais skaits, rīka pieejamību, izslēdzot rīkus, kuru piekļuve nav iespējama bez specifikām licencēm. Tiks ņemts vērā arī rīka atbalstītās datu ievākšanas vides, izslēdzot rīkus, kurus ierobežotu pārāk šaurs pielietojamo datu savākšanas iespēju klāsts. ETL rīkam, kas tiek testēts jāspēj arī saglabāt vai nogādāt datus datu krātuvē. Netiks ņemti vērā arī rīki kas veic tikai daļēju ETL rīku procesu piemēram rīki kas tikai savāc un nogādā datus glabātuvē izlaižot transformāciju balstoties uz atbalstošu rīku vai sistēmu, kas veiktu šo procesu vai rīki kuri paredzēti tikai savākšanai un pārstāj procesu pie sakārtotas ievākto datu kopas kuru paredzēts nogādāt citam rīkam, kas veic datu ielādi glabātuvē.

## 2.3. Izvēlētie rīki

Izvēlētie rīki atbilda visām prasībām, kas bija nepieciešamas rīka izvēlei. No četriem izvēlētajiem rīkiem viens ir tiešsaistes rīks, kura darbība ļoti atšķiras no pārējiem.

### 2.3.1. Talend ETL

Talend ETL ir bezmaksas datu integrācijas rīks tas ir daļa no lielāka datu apstrādes un manipulācijas sistēma, kas sastāv no vairākiem moduļiem un Talend ETL, ir viens no tiem.[7]

Talend ETL rīks pamātā ir manipulējams ar Grafisku vidi, kur iespējams aplūkot dažādos datu elementus un pārveidojumus, kas norisinās ETL procesā šis, rīks arī iekļauj vairākas citas noderīgas funkcionalitātes:

- Iespēju definēt komplicētus datu kartēšanas gadījumus izmantojot datu transformācijas iespējas, kas spēj iekš sistēmas mainīt datu tipu un formātu.
- Iekš rīka iespējams arī veidot java vai perl koda datnes kuras var izpildīt jebkurā iekārtā kurā ir ETL rīks.
- Iespējams implementēt atkļūdošanas elementus rīka darbības reāllaikā.
- Rīks atpazīst plašu klāstu ar datu formātiem un sniedz ieskatu iespējamajā datu saņemšanas un transformācijas veidnē.

### 2.3.2. Altova Maptool

Altova Maptool ir maksas rīks, kurš piedāvā bezmaksas izmēģinājuma periodu, kurā bija iespējams testēt sistēmas darbības paplašināšanas iespējas. Rīka iespējamai apstrādājamo datu formātu klāsts ir ļoti plašs. Šinī rīkā datus var manipulēt un definēt dažādos datu avotus un to ielādēšanas galapunktus un to visu iespējams veikt grafiskā vidē ar vizuāliem elementiem, kas reprezentē dažādās darbības. Rīka priekšrocības un papildu funkcionalitātes:

- Liels pieejamo datu bibliotēku skaits, kas atļauj manipulēt un pārveidot datus no dažādiem datu avotiem.
- Rīkā var detalizēti veidot darbību hierarhiju, kas var detalizēti pārveidot ieejas datus.
- Iespēja aplūkot potenciālos datu ievākšanas un pārveidošanas rezultātus.

[6]

### 2.3.3. Fivetran

Fivetran ir tiešsaistes rīks ar ļoti vienkāršu un ātri apgūstamu darbības mehānismu. Rīka darbība notiek ārēji ne lietotāja iekārtā līdz ar to rīka darbībai var piekļūt no jebkuras lokācijas ar interneta piekļuvi. Rīka priekšrocības un funkcionalitātes:[4]

- Iespēja veidot robustas datu transformācijas shēmas
- Loti vienkārša jau datu avotu pievienošanas metode.
- Iekļauts datu transformācijas rīks kas atļauj veidot sql skriptu valodas kodu datu manipulācijām.
- Visi ielādētie dati pieejami ar sql komandām.

### 2.3.4. CloverDX desiner

CloverDX desiner ir publiskas piekļuves rīks, un tas ir viens no pirmajiem rīkiem, kas publicēts kā brīvas piekļuves. Šim rīkam ETL procesu veidošana notiek grafiskā vidē, kur tiek veidotas darbības no blokiem, kas reprezentē ienākošos datus transformācijas un izvades datus. Iespējams arī tieši aplūkot ETL procesa programmas kodu un modificēt to. Šis rīks grūti identificējams ar kādām unikālām īpašībām līdz ar to nav uzskaitāmas papildu priekšrocības vai funkcionalitātes.[5]

# 3. Rīku uzlabojumi un testi

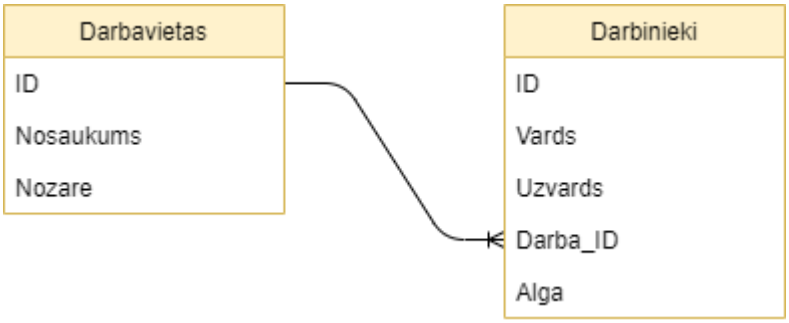
## 3.1. Testēšanas datnes

Priekš rīku testēšanas izveidoja vairākus datu formāta datnes ar testpiemēru informāciju par darbiniekiem un darba vietām.

Testpiemēru datņu formāti:

- XLSX – izveidota viena datne ar divām lapām, kur katra satur savus datus viena darbiniekus otra darbavietas.
- CSV – Divi teksta tipa datnes kuros ir dati, atdalīti, ar komatiem iekļaujoties CSV datu tipam.
- JSON – Divi teksta tipa datnes ar datiem atbilstošā formā datu atpazīšanai.
- PostgreSQL – datubāze ar divām tabulām.

Bija nepieciešams izveidot vairākus datu formātus testa datiem, jo pastāvēja iespēja, ka visi rīki nespēs atpazīt datu formātu vai arī nebūs iespējams pielāgot ETL rīku kļūdas risinājuma situācijai. Rīkos, ja būs iespējams, papildināt kļūdas apstrādes funkcionalitāti jāpārbauda, vai šī kļūda tiek, atpazīta vairākos datu formātos, tādēļ visi datu avoti testiem satur vienādu informāciju.




*3.1. att.* ***Testa tabulu datu struktūra***

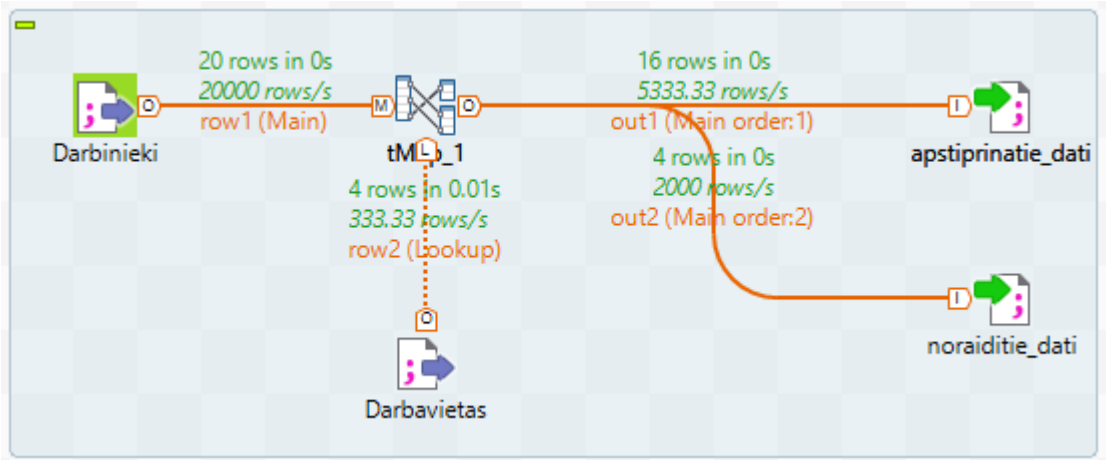
Testēšana tika veikta, izveidojot vietējā direktorijā testa failus, testa faili sastāv no diviem individuāliem failiem JSON CSV datu tipiem un viena faila XLSX gadījumā, Rīkos tika izveidots process kas norādot faila direktoriju spētu savākt tā informāciju un transformēt to apvienojot abas datu tabulas vienā, kur pie katra darbinieka norādīta arī tā darbavieta un nozare, ko atpazīst pēc ārējās atslēgas lauka Darba_ID, pēc datu apstrādes dati tiek ielādēti vietējā direktorijā kur var aplūkot gūtos datus no ETL procesa. Rīkos tika izvēlēts atbilstošais datu atlases veids priekš katra datu tipa. Pirmais tests tiks veikts ar visiem datiem, bez paredzamam kļūdām. Nākošie testi iekļaus vairākas paredzamas datu transformācijas kļūdas sākot ar trūkstošu informāciju pāris datu laukos no ievadātiem, tālāk ievadātos trūkstošu datu kolonu un beidzot ar trūkstošu darbavietu tabulu, kas ir daļa no gala nepieciešamajiem datiem.

Rīkos tiks mēģināts ar pieejamajām metodēm pielāgot rīka darbību, lai trūkstošo datu izraisītās problēmas neradītu sistēmas apstāšanos un tiktu izvadīti gala dati ar atbilstošu saturu.

Gala datus var uzskatīt par atbilstošiem ja ir izveidots gala datu kopa un visus trūkstošos datus ir bijusi iespēja aizvietot ar tekstu “Nav datu”.

## 3.2. Talend ETL testēšana

Iepazīstoties ar darba vidi rīkā varēja uzreiz saprast darbības iespējas un kā veidot ETL datu savākšanas un pārveidošanas procesu. Lai pārbaudītu datu apstrādi tika izveidots ETL process, kura dati tiek savākti no divām tabulām darbinieki un darbavietas, pēc tam tika apvienoti dati no tabulām katram darbinieka ierakstam, pievienojot darbavietas nosaukumu un darba nozari, kā arī tika atlasīti atsevišķi darbinieki, kuriem nav darba vietas.




*3.2. att.* ***Talend ETL rīkā izveidots datu apstrādes process***

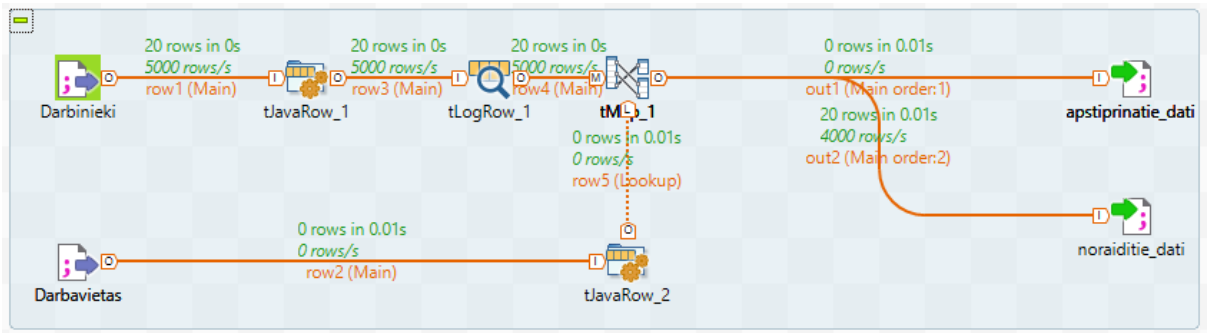
Procesa darbība apstiprinājās procesa rezultātā tika iegūts viens datne ar apvienotiem darbinieku datiem ar darba vietas datiem, un otrs datne kura satur darbiniekus, kuru darba vietas indekss neatbilst esošajām darba vietām darba vietu datnē.

### 3.2.1. Rīka manipulācija ar iebūvētajam opcijām

Rīkā ir iekļautas vairākas opcijas nestandarta papildnoteikumu ieviešanai. Rīka lietotājam tiek doti vairāki procesa plāna bloki, kuros var izveidot Java programmēšanas valodas kodu, kas spēj darboties ar ieejas datiem un datu pārveidošanas gadījumā nodot izejas datus pārveidotus.

Rīka datu apstrādes procesu iespējams arī aplūkot java programmēšanas valodas formā kur katrs no ievietotajiem darbības blokiem ir tā programmas koda formā. Ja izvēlas šo datni atvērt rediģēšanas veidā iespējams arī brīvi pārveidot visa procesa kodu pievienojot papildu pārbaudes un noteikumus kādi nepieciešamai katrā situācijā.

Datu savākšanas un pārveidošanas procesam tika ievienots rīkā iekļautais java programmēšanas valodas elements kurā tika uzrakstīts programmas kods kurš saņemot ienākošos datus no datu avota lasītāja veic pārbaudi vai ienākošie dati atbilst paredzētajam formātam kurš arī ir definēts java koda fragmentā, ja vērtība, kas saņemta nav atbildusi tad trūkstošās informācijas laukā tiek padota vērtība "Nav datu" rīka procesa vizualizācijā kas redzamā 3.3. attēlā var redzēt divus ievades elementus "Darbinieki" un "Darbavietas" pēc katra no šiem elementiem seko koda elements kurā ir iespēja pievienot rīkā vēl neeksistējošu kodu. Rīka pielāgošanā tika izmantots arī rīka procesa pamata koda manipulācija pievienojot papildus nosacījumu kas padotu tukšu kopu kāda no ievades faila trūkuma gadījumā.




*3.3. att.* ***Talend ETL rīkā izveidots process ar java koda elementiem***

**Situācija nr.1:**

Talend ETL rīkam tiek padoti testa datnes dažādos datu formātos ar pilnu informāciju. Nav veiktas detnes modifikācijas līdz ar to nav paredzama rīka kļūdas situācija.

**Prognozētais iznākums:**

Process noasināsies bez problēmām, un tiks atgriezti visi dati un nebūs neviena datu lauka, kurš aizvietots ar trūkstošu datu tekstu vai ar tukšu lauku.

**Iznākums:**

Process noritēja kā paredzēts bez problēmām, un tika atgriezti dati to paredzētajā formā ar visiem nepieciešama datu laukiem aizpildītiem. Rezultāta datu kopu var aplūkot 3.4. attēlā.

```
1;Janis;Kalnins;FutureTec;Tec;600
2;Peteris;Ozolins;AlfaBiz;Fiz;500
3;Martins;Osis;Omego;Bio;780
4;Sems;Klava;Tesla;Tec;560
5;Laila;Ziemelis;FutureTec;Tec;340
6;Zigis;Kalnins;FutureTec;Tec;890
7;Maruta;Ozolins;AlfaBiz;Fiz;780
8;Tirza;Osis;Omego;Bio;560
9;Memele;Klava;Tesla;Tec;860
10;Toms;Ziemelis;AlfaBiz;Fiz;740
11;Timijs;Kalnins;FutureTec;Tec;760
12;Janis;Ozolins;AlfaBiz;Fiz;450
13;Peteris;Osis;Omego;Bio;330
14;Martins;Klava;Tesla;Tec;770
15;Sems;Ziemelis;Omego;Bio;660
16;Laila;Kalnins;FutureTec;Tec;940
17;Zigis;Ozolins;AlfaBiz;Fiz;370
18;Maruta;Osis;Omego;Bio;110
19;Tirza;Klava;Tesla;Tec;440
```

*3.4. att.* ***Talend ETL rīka iznākuma dati pirmajā testā***

**Situācija nr.2:**

Talend ETL rīkam tiek padotas testa datnes atbilstošos datu formātos un lai simulētu trūkstošu informāciju darbinieku datnē tiek izņemts vārds un uzvārds katrai otrajai personai.

**Prognozētais iznākums:**

Process noasināsies bez problēmām, un tiks atgriezti visi dati un pusei no personām vārda un uzvārda vietā būs “Nav datu” teksts..

**Iznākums:**

Process noritēja kā paredzēts bez problēmām, un tika atgriezti dati to paredzētajā formā, ar katras otrās personas vāra un uzvārda vietā tekstu “Nav datu”. Rezultāta datu kopu var aplūkot 3.5. attēlā.

```
1;Janis;Kalnins;FutureTec;Tec;600
2;Nav datu;Nav datu;AlfaBiz;Fiz;500
3;Martins;Osis;Omego;Bio;780
4;Nav datu;Nav datu;Tesla;Tec;560
5;Laila;Ziemelis;FutureTec;Tec;340
6;Nav datu;Nav datu;FutureTec;Tec;890
7;Maruta;Ozolins;AlfaBiz;Fiz;780
8;Nav datu;Nav datu;Omego;Bio;560
9;Memele;Klava;Tesla;Tec;860
10;Nav datu;Nav datu;AlfaBiz;Fiz;740
11;Timijs;Kalnins;FutureTec;Tec;760
12;Nav datu;Nav datu;AlfaBiz;Fiz;450
13;Peteris;Osis;Omego;Bio;330
14;Nav datu;Nav datu;Tesla;Tec;770
15;Sems;Ziemelis;Omego;Bio;660
16;Nav datu;Nav datu;FutureTec;Tec;940
17;Zigis;Ozolins;AlfaBiz;Fiz;370
18;Nav datu;Nav datu;Omego;Bio;110
19;Tirza;Klava;Tesla;Tec;440
20;Nav datu;Nav datu;Tesla;Tec;360
```

*3.5. att.* ***Talend ETL rīka iznākuma dati otrajā testā***

**Situācija nr.3:**

Talend ETL rīkam tiek padotas testa datnes atbilstošos datu formātos un lai simulētu trūkstošu informāciju darbavietu datnē tiek noņemta nozares kolona.

**Prognozētais iznākums:**

Process noasināsies bez problēmām, un tiks atgriezti visi dati, bet pirmspēdējās kolonas dati būs aizvietoti visās instancēs ar tekstu "Nav datu".

**Iznākums:**

Process noritēja kā paredzēts bez problēmām, un tika atgriezti dati to paredzētajā formā ar visiem nepieciešama datu laukiem aizpildītiem. Rezultāta datu kopu var aplūkot 3.6. attēlā.

```
1;Janis;Kalnins;FutureTec;Nav datu;600
2;Peteris;Ozolins;AlfaBiz;Nav datu;500
3;Martins;Osis;Omego;Nav datu;780
4;Sems;Klava;Tesla;Nav datu;560
5;Laila;Ziemelis;FutureTec;Nav datu;340
6;Zigis;Kalnins;FutureTec;Nav datu;890
7;Maruta;Ozolins;AlfaBiz;Nav datu;780
8;Tirza;Osis;Omego;Nav datu;560
9;Memele;Klava;Tesla;Nav datu;860
10;Toms;Ziemelis;AlfaBiz;Nav datu;740
11;Timijs;Kalnins;FutureTec;Nav datu;760
12;Janis;Ozolins;AlfaBiz;Nav datu;450
13;Peteris;Osis;Omego;Nav datu;330
14;Martins;Klava;Tesla;Nav datu;770
15;Sems;Ziemelis;Omego;Nav datu;660
16;Laila;Kalnins;FutureTec;Nav datu;940
17;Zigis;Ozolins;AlfaBiz;Nav datu;370
18;Maruta;Osis;Omego;Nav datu;110
19;Tirza;Klava;Tesla;Nav datu;440
20;Memele;Ziemelis;Tesla;Nav datu;360
```

*3.6. att.* ***Talend ETL rīka iznākuma dati trešajā testā***

**Situācija nr.4:**

Talend ETL rīkam tiek padotas testa datnes atbilstošos datu formātos un lai simulētu trūkstošu informāciju procesam netiek padota darbavietu datne.

**Prognozētais iznākums:**

Process noasināsies bez problēmām, un tiks atgriezta datne ar tikai personas ID, vārda un uzvārda laukiem.

**Iznākums:**

Process noritēja, kā paredzēts bez problēmām un tika atgriezti dati to paredzētajā formā ar tikai trīs paredzētajām vērtībām par katru personu. Rezultāta datu kopu var aplūkot 3.7. attēlā.

```
1;Janis;Kalnins;
2;Peteris;Ozolins;
3;Martins;Osis;
4;Sems;Klava;
5;Laila;Ziemelis;
6;Zigis;Kalnins;
7;Maruta;Ozolins;
8;Tirza;Osis;
9;Memele;Klava;
10;Toms;Ziemelis;
```

*3.7. att.* ***Talend ETL rīka iznākuma datu fragments ceturtajā testā***

### 3.2.2. Rīka manipulācija ar konfigurācijas datnēm un citām metodēm

Talend ETL rīkam ir ļoti plaša iekšēja datu atlases struktūra un iespējamās darbības, un to pielāgošanas iespējas ir jau iekļautas standarta rīka darbībā. Visas iespējamās pielāgošanas iespējas, kas tika, piedāvātas konfigurācijas datnēs, neietekmēja iespējamās kļūdas situācijas, kas varētu notikt trūkstošu datu sakarā.

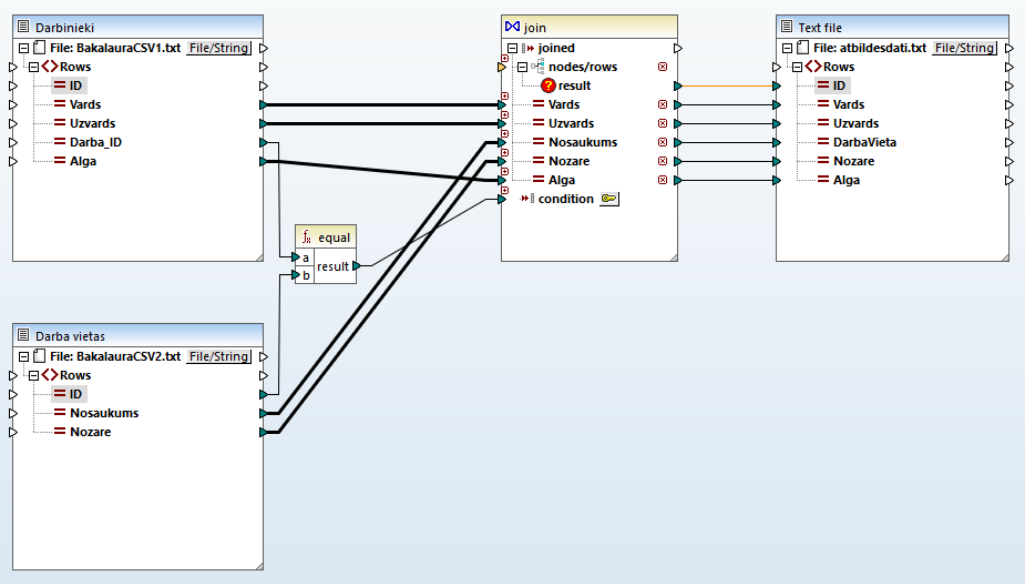
Rīka piedāvātie elementi datu savākošanas vajadzībām deva lielu iespēju manipulēt un pielāgot to darbību dažādām prasībām, toties, tiem bez tiešas koda papildināšanas nevarēja veikt nepieciešamo kļūdas situāciju risināšanu.

Šim rīkam nav reālistiski iespējams izmainīt programmas pamat kodu sakarā ar to, ka tas nav pieejams publisks koda avots.

## 3.3. Altova Maptool testēšana

Uzsākt darbu Altova Maptool darba vidē nav viegli un nav uzreiz saprotams, kā notiek datu apstrādes process, lai gan viss process notiek grafiskā darba vidē, kur katru darbību apzīmē ar brīvi pārvietojamu bloku kuram, var pielāgot tā nepieciešamo darbību.
Rīkā izveidotos procesus iespējams aplūkot tikai grafiskajā vidē un nav opcijas rediģēt detalizētākus elementus katrā darbības blokā. Darbību bloki paši ir ļoti detalizēti sadalīti, katrai loģiskajai darbībai kā opcijai ja tad vai salīdzināšanas darbības visas ir nodalītas individuāli, ļaujot veidot kompleksas sistēmas ar ievāktajiem datiem.

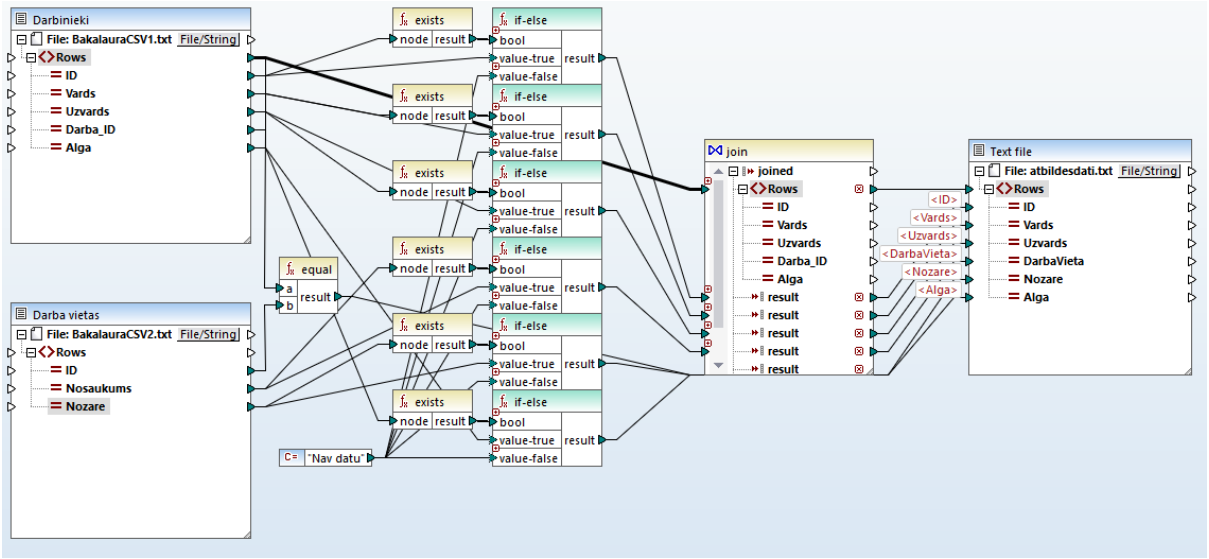
Rīkā tika izveidots process, kas paredzēts divu datu tabulu apvienošanai un datu izvadei teksta datnē kas aplūkojams 3.8. attēlā. Process sastāv no divien ievaddatu laukiem viena loģikas salīdzinājuma lauka viena datu apvienošanas lauka un datu izvades lauka.




*3.8. att.* ***Altova Maptool rīkā izveidots datu apstrādes process***

### 3.3.1. Rīka manipulācija ar iebūvētajam opcijām

Altova Maptool rīks nepiedāvāja veidus, kā papildināt procesu nestandarta veidos iekš rīka darbības. Nebija pieejams ne tieša koda ievadīšanas elements ko pievienot procesam ne arī visa procesa pārskatīšanas un modificēšanas iespējas programmas koda formā.
Rīka funkcionalitātes pielāgošanai tika izmantoti rīkā iekļautie loģikas elementi, kas sniedza iespēju veidot loģikas darbības ar datu pārbaudēm un bija iespējams pievienot noteikumus trūkstošu datu gadījumos. Process tika papildināts ar vairākiem loģikas elementiem, kas pārbauda ievaddatu pareizību jeb vai tie ir padoti ielādei un pēc tā, atbilde tiek nodota loģikas elementam ja tad kurā pie eksistējošiem datiem tie tiek padoti tālāk un ja neeksistē tad padota atbilde “Nav datu” process attēlots 3.9. attēlā.




*3.9. att.* ***Altova ETL rīkā izveidots datu apstrādes process ar papildnoteikumiem***

**Situācija nr.1:**

Altova ETL rīkam tiek padota testa datne dažādos datu formātos ar pilnu informāciju. Nav veiktas datnes modifikācijas līdz ar to nav paredzama rīka kļūdas situācija.

**Prognozētais iznākums:**

Process noasināsies bez problēmām, un tiks atgriezti visi dati un nebūs neviena datu lauka, kurš aizvietots ar trūkstošu datu tekstu vai ar tukšu lauku.

**Iznākums:**

Process noritēja kā paredzēts bez problēmām, un tika atgriezti dati to paredzētajā formā ar visiem nepieciešama datu laukiem aizpildītiem. Rezultāta datu kopu var aplūkot 3.10. attēlā.

```
1   1,Janis,Kalnins,FutureTec,Tec,600
2   2,Peteris,Ozolins,AlfaBiz,Fiz,500
3   3,Martins,Osis,Omego,Bio,780
4   4,Sems,Klava,Tesla,Tec,560
5   5,Laila,Ziemelis,FutureTec,Tec,340
6   6,Zigis,Kalnins,FutureTec,Tec,890
7   7,Maruta,Ozolins,AlfaBiz,Fiz,780
8   8,Tirza,Osis,Omego,Bio,560
9   9,Memele,Klava,Tesla,Tec,860
10  10,Toms,Ziemelis,AlfaBiz,Fiz,740
11  11,Timijs,Kalnins,FutureTec,Tec,760
```

*3.10. att.* ***Altova ETL rīka iznākuma datu fragments pirmajā testā***

**Situācija nr.2:**

Altova ETL rīkam tiek padotas testa datnes atbilstošos datu formātos un lai simulētu trūkstošu informāciju darbinieku datnē tiek izņemts vārds un uzvārds katrai otrajai personai.

**Prognozētais iznākums:**

Process noasināsies bez problēmām, un tiks atgriezti visi dati un pusei no personām vārda un uzvārda vietā būs “Nav datu” teksts..

**Iznākums:**

Process noritēja kā paredzēts bez problēmām, un tika atgriezti dati to paredzētajā formā ar katras otrās personas vārdu un uzvārdu vietā tekstu “Nav datu”. Rezultāta datu kopu var aplūkot 3.11. attēlā.

```
1,Janis,Kalnins,FutureTec,Tec,600
2,Nav datu,Nav datu,AlfaBiz,Fiz,500
3,Martins,Osis,Omego,Bio,780
4,Nav datu,Nav datu,Tesla,Tec,560
5,Laila,Ziemelis,FutureTec,Tec,340
6,Nav datu,Nav datu,FutureTec,Tec,890
7,Maruta,Ozolins,AlfaBiz,Fiz,780
8,Nav datu,Nav datu,Omego,Bio,560
9,Memele,Klava,Tesla,Tec,860
10,Nav datu,Nav datu,AlfaBiz,Fiz,740
```

*3.11. att.* ***Altova ETL rīka iznākuma datu fragments otrajā testā***

**Situācija nr.3:**

Altova ETL rīkam tiek padotas testa datnes atbilstošos datu formātos un lai simulētu trūkstošu informāciju darbavietu datnē tiek noņemta nozares kolona.

**Prognozētais iznākums:**

Process noasināsies bez problēmām, un tiks atgriezti visi dati, bet darba kolonas dati būs aizvietoti visās instancēs ar tekstu “Nav datu”.

**Iznākums:**

Process noritēja kā paredzēts bez problēmām, un tika atgriezti dati to paredzētajā formā ar visiem nepieciešama datu laukiem aizpildītiem. Rezultāta datu kopu var aplūkot 3.12. attēlā.

```
1,Janis,Kalnins,FutureTec,Nav datu,600
2,Peteris,Ozolins,AlfaBiz,Nav datu,500
3,Martins,Osis,Omego,Nav datu,780
4,Sems,Klava,Tesla,Nav datu,560
5,Laila,Ziemelis,FutureTec,Nav datu,340
6,Zigis,Kalnins,FutureTec,Nav datu,890
7,Maruta,Ozolins,AlfaBiz,Nav datu,780
8,Tirza,Osis,Omego,Nav datu,560
9,Memele,Klava,Tesla,Nav datu,860
10,Toms,Ziemelis,AlfaBiz,Nav datu,740
```

*3.12. att.* ***Altova ETL rīka iznākuma datu fragments trešajā testā***

**Situācija nr.4:**

Altova ETL rīkam tiek padotas testa datnes atbilstošos datu formātos un lai simulētu trūkstošu informāciju procesam netiek padota darbavietu datne.

**Prognozētais iznākums:**

Process noasināsies bez problēmām, un tiks atgriezti datnes ar tikai personas ID, vārda un uzvārda laukiem.

**Iznākums:**

Process apstājās un nespēja paveikt datu ielādi un apstrādi. Sakarā ar to, ka datu apstrādes elementi var būt tikai tādi, kādu piedāvā sistēmā iekļautie, tādēļ nav elementu, kas spētu noteikt datnes trūkumu ievaddatos. Procesa darbības laikā tika pieprasīti datu lauki no datnes, kurš neeksistē tādēļ, arī nav potenciāls veids kā novērst datnes trūkumu ielādējamo datu savākšanas solī.

### 3.3.2. Rīka manipulācija ar konfigurācijas datnēm un citām metodēm

Altova Maptool rīka konfigurācijas datnēs var ietekmēt vairākus darbības elementus, kā definēt dažādo ievaddatu datnes formātu versijas un kuras no tām būs standartā pieņemtās, kā esošas, konfigurācijas datnēs, iespējams arī definēt globālas vērtības, kuras tiek izmantotas datu apstrādes procesos. Vienīgā opcija konfigurācijas datos, kas atbilda meklētajām, ir apstādināt rīka darbību situācijā, kad tiek sastapta sistēmas kļūda. Šāda opcija var palīdzēt situācijās, kur svarīgi ir neapturēt citus procesus viena rīka vainas dēļ.

Šim rīkam nav pieejams pamata kods, kurā varētu veikt vispārējas izmaiņas datu savākšanas procesā.

## 3.4. Fivetran testēšana

Fivetran tiešsaistes rīks lietotājam viegli uztverams, un ir viegli izveidot ETL procesu, kas savāktu datus un nogādātu tos glabātuvē. Rīka lielākais mīnuss ir tas, ka tas radīts ļoti vienkāršai lietošanai. Rīkā nav iekļauts ienākošo datu transformācijas mehānisms, bet gan iespējams pārveidot datus to glabātuvē ar SQL skriptiem.

Rīkā definētās transformācijas iespējams aplūkot SQL skriptu formā un iespējams uzstādīt regulāru darbību atkārtošanu neatkarīgu no datu savākšanas procesa.

### 3.4.1. Rīka manipulācija ar iebūvētajām opcijām

Ņemot vērā, ka datu transformācija notiek pēc datu ielādēšanas ar datiem, kas, nogādāti glabātuvē to pielāgošanas iespējas trūkstošiem datiem, datu savākšanas procesā ir anulējamas. Kaut arī rīkam ir iespējams pārlasīt datus un analizēt iespējams trūkstošus laukus,

un papildināt vajadzīgās vērtības, toties ja dati kas nogādāti glabātuvē un pārrakstījuši iepriekšēju ierakstu, neatstājot tukšu vērtību jaunajā ierakstā. Vairākas situācijas paredz, ka šāda tipa pieeja datu transformācijai nespēj atrisināt trūkstošu datu problēmas. Rīkā tika veikti testi, bet rīka funkcionalitāti nevarēja pielāgot testa kļūdas situāciju labošanai.

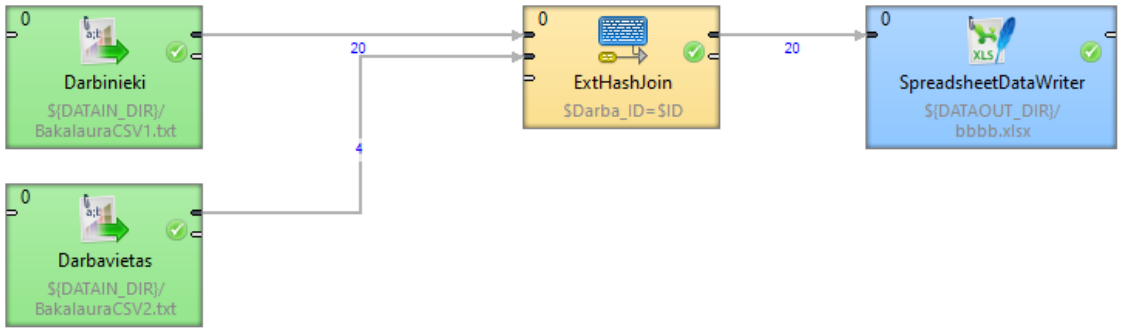
### 3.4.2. Rīka manipulācija ar konfigurācijas datnēm un citām metodēm

Rīka viss process notiek tiešsaistē, līdz art to nav pieejamas konfigurācijas datnes, kas uzreiz izslēdz iespēju pievienot noteikumus tādā veidā, kas palīdzētu ar datņu savākšanu savākšanas potenciālajām kļūdām.

Arī sakarā ar to, ka rīks ir tikai tiešsaistē pieejams, nav iespējams piekļūt tā pamat kodam, kas liedz jebkādas izmaiņas tiešā rīka darbībā.

## 3.5. CloverDX desiner testēšana

Uzsākot darbu CloverDX ETL rīkā viegli, izveidot vienkāršus ETL procesus, kas var ielādēt transformēt un noglabāt datus. Visa rīka darbība notiek grafiskā vidē ar darbībām, ko apzīmē darbību bloki. Rīka procesa vizualizācija attēlota 3.13. attēlā.

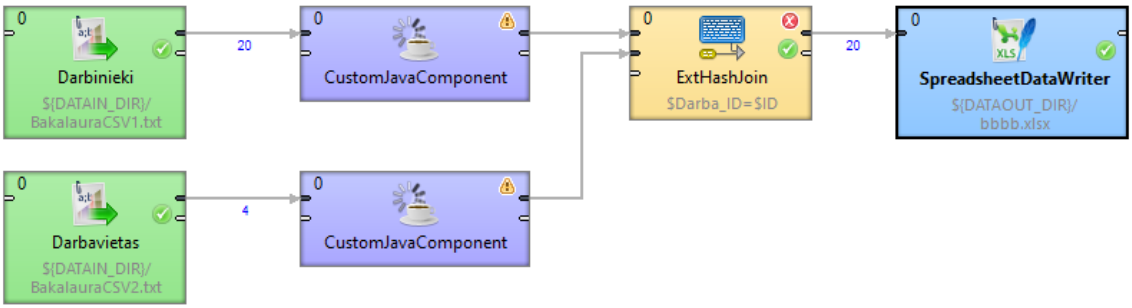



*3.13. att.* ***CloverDX ETL rīkā izveidots datu apstrādes process***

### 3.5.1. Rīka manipulācija ar iebūvētajām opcijām

CloverDX rīkā Nebija iespēja aplūkot un modificēt darbības procesu, aplūkojot to programmas koda formā. Vienīgais veids, kā varēja pievienot, papildus noteikumus, kad nav iekļauti jau standarta rīka funkcionalitātē ir izmantot pielāgojamus java koda elementus kuros varēja iegūt ieejas datus no datu nolasītājiem un pievienot pārbaudes, vai ievaddatiem ir ne nulles vērtības. Rīka iekļautajos java koda blokos tika izveidots programmas kods, kas atlasa

datus, kas definēti kā ievaddati un pārbauda vai tie satur vērtību pēc kā ja dati ir pareizi padod tos tālāk ja nē tad definē to kā neesošus. Procesa vizualizāciju var aplūkot 3.14 attēlā.




*3.14. att.* ***CloverDX ETL rīkā izveidots process ar java koda elementiem***

**Situācija nr.1:**

CloverDX ETL rīkam tiek padotas testa datnes dažādos datu formātos ar pilnu informāciju. Nav veiktas datņu modifikācijas līdz ar to nav paredzama rīka kļūdas situācija.

**Prognozētais iznākums:**

Process noasināsies bez problēmām, un tiks atgriezti visi dati un nebūs neviena datu lauka, kurš aizvietots ar trūkstošu datu tekstu vai ar tukšu lauku.

**Iznākums:**

Process noritēja kā paredzēts bez problēmām, un tika atgriezti dati to paredzētajā formā ar visiem nepieciešama datu laukiem aizpildītiem. Rezultāta datu kopu var aplūkot 3.15. attēlā.

| 1 | ID | Vards | Uzvards | Darbavieta | Nozare | Alga |
|---|---|---|---|---|---|---|
| 2 | 1 | Janis | Kalnins | FutureTec | Tec | 600 |
| 3 | 2 | Peteris | Ozolins | AlfaBiz | Fiz | 500 |
| 4 | 3 | Martins | Osis | Omego | Bio | 780 |
| 5 | 4 | Sems | Klava | Tesla | Tec | 560 |
| 6 | 5 | Laila | Ziemelis | FutureTec | Tec | 340 |
| 7 | 6 | Zigis | Kalnins | FutureTec | Tec | 890 |

*3.15. att.* ***CloverDX ETL rīka rezulta datu fragments pirmajā testā***

**Situācija nr.2:**

CloverDX ETL rīkam tiek padotas testa datnes atbilstošos datu formātos un lai simulētu trūkstošu informāciju darbinieku datnēs tiek izņemts vārds un uzvārds katrai otrajai personai.

**Prognozētais iznākums:**

Process noasināsies bez problēmām, un tiks atgriezti visi dati un pusei no personām vārda un uzvārda vietā būs "Nav datu" teksts..

**Iznākums:**

Process noritēja kā paredzēts bez problēmām, un tika atgriezti dati to paredzētajā formā, ar katru otrās personas vāru un uzvārdu vietā tekstu "Nav datu". Rezultāta datu kopu var aplūkot 3.16. attēlā.

| 1 | ID | Vards | Uzvards | Darbavieta | Nozare | Alga |
|---|---|---|---|---|---|---|
| 2 | 1 | Janis | Kalnins | FutureTec | Tec | 600 |
| 3 | 2 | Nav datu | Nav datu | AlfaBiz | Fiz | 500 |
| 4 | 3 | Martins | Osis | Omego | Bio | 780 |
| 5 | 4 | Nav datu | Nav datu | Tesla | Tec | 560 |
| 6 | 5 | Laila | Ziemelis | FutureTec | Tec | 340 |
| 7 | 6 | Nav datu | Nav datu | FutureTec | Tec | 890 |
| 8 | 7 | Maruta | Ozolins | AlfaBiz | Fiz | 780 |

*3.16. att.* ***CloverDX ETL rīka rezulta datu fragments otrajā testā***

**Situācija nr.3:**

CloverDX ETL rīkam tiek padotas testa datnes atbilstošos datu formātos un lai simulētu trūkstošu informāciju darbavietu datnē tiek noņemta nosaukuma kolona.

**Prognozētais iznākums:**

Process noasināsies bez problēmām, un tiks atgriezti visi dati, bet darba kolonas dati būs aizvietoti visās instancēs ar tekstu "Nav datu".

**Iznākums:**

Process noritēja kā paredzēts bez problēmām, un tika atgriezti dati to paredzētajā formā ar visiem nepieciešama datu laukiem aizpildītiem. Rezultāta datu kopu var aplūkot 3.17. attēlā.

| 1 | ID | Vards | Uzvards | Darbavieta | Nozare | Alga |
|---|---|---|---|---|---|---|
| 2 | 1 | Janis | Kalnins | FutureTec | Nav datu | 600 |
| 3 | 2 | Peteris | Ozolins | AlfaBiz | Nav datu | 500 |
| 4 | 3 | Martins | Osis | Omego | Nav datu | 780 |
| 5 | 4 | Sems | Klava | Tesla | Nav datu | 560 |
| 6 | 5 | Laila | Ziemelis | FutureTec | Nav datu | 340 |
| 7 | 6 | Zigis | Kalnins | FutureTec | Nav datu | 890 |
| 8 | 7 | Maruta | Ozolins | AlfaBiz | Nav datu | 780 |
| 9 | 8 | Tirza | Osis | Omego | Nav datu | 560 |
| 10 | 9 | Memele | Klava | Tesla | Nav datu | 860 |

*3.17. att.* ***CloverDX ETL rīka rezulta datu fragments trešajā testā***

**Situācija nr.4:**

CloverDX ETL rīkam tiek padotas testa datnes atbilstošos datu formātos un lai simulētu trūkstošu informāciju procesam netiek padota darbavietu datne.

**Prognozētais iznākums:**

Process noasināsies bez problēmām, un tiks atgriezta datne ar tikai personas ID, vārda un uzvārda laukiem.

**Iznākums:**

Process tika apturēts, jo nav nepieciešamie dati. Kaut arī ir pieejams, koda rakstīšanas elements rīkā. tomēr nav iespēja pārveidot datu ievākšanas elementu. un pievienot noteikumu, kas atļautu turpināt rīka darbību pēc datu avota datnes neatrašanas.

### 3.5.2. Rīka manipulācija ar konfigurācijas datnēm un citām metodēm

CloverDX ETL rīkam ir pieejams ļoti plašs konfigurējamo datu klāsts, bet darba autoram nav izdevies atrast iespējamu veidu kā no šīm datnēm pielāgot rīka darbību potenciālā kļūdas situācijā.

Kaut arī bija pieejams kods tas, nesniedza vēlamās opcijas un kaut arī ir iespējams modificēt pamata datu ievades elementus ir labāka alternatīva, izmantojot rīkā iekļautās koda ievietošanas opcijas.

## 3.6. Kopsavilkums

ETL rīkos iekļautās funkcionalitātes nosedz lielu rīka darbības diapazonu, bet nav iespējams paredzēt visas situācijas, kas var atgadīties rīka darbības laikā. ETL rīkos, kuri paredzēti izmantošanai ne tiešsaistē pielāgošanas iespējas un izmantošanas diapazons bija daudz plašāks ar vairāk opcijām. Rīkā ko var izmantot tikai tiešsaistē, nebija pietiekama funkcionalitāte, ko var izskaidrot ar rīka resursu patēriņu sistēmas izstrādātāju pusē, kur rīkam jāatbild par nenoteiktu daudzumu lietotāju, un sniegt katram iespējas manipulēt ar datu pārveidošanas procesu tā savākšanas solī būtu bezatbildīgi no patērēto resursu apjoma.

Rīkiem veiktu testu un situāciju pārskats aplūkojams 3.1. un 3.2. tabulās.

*ETL rīku un testa piemēru iznākumi* Tabula 3.1.

| ETL rīks | Tests ar pareiziem ieejas datiem. | Tests ar trūkstošiem pāris datu laukiem. | Tests ar Trūkstošu datu kolonu. | Tests bez darbinieku faila. |
|---|---|---|---|---|
| Talend ETL | Rīks veiksmīgi atgriež paredzētās vērtības. | Rīks veiksmīgi atgriež paredzētās vērtības | Rīks veiksmīgi atgriež paredzētās vērtības | Rīks veiksmīgi atgriež paredzētās vērtības. |
| Altova Maptool | Rīks veiksmīgi atgriež paredzētās vērtības. | Rīks veiksmīgi atgriež paredzētās vērtības. | Rīks veiksmīgi atgriež paredzētās vērtības. | Rīkā nav izdevies novērst kļūdas situāciju. |
| Fivetran | Rīks veiksmīgi atgriež paredzētās vērtības. | Rīkā nav izdevies novērst kļūdas situāciju. | Rīkā nav izdevies novērst kļūdas situāciju. | Rīkā nav izdevies novērst kļūdas situāciju. |
| CloverDX desiner | Rīks veiksmīgi atgriež paredzētās vērtības. | Rīks veiksmīgi atgriež paredzētās vērtības. | Rīks veiksmīgi atgriež paredzētās vērtības. | Rīkā nav izdevies novērst kļūdas situāciju. |

*ETL rīku alternatīvo metožu iespējas* Tabula 3.2.

| ETL rīks | Konfigurācijas failu labošana | Rīka pamata koda manipulācijas |
|---|---|---|
| Talend ETL | Neietekmē kļūdu novēršanu | Nav pieejams pirmkods |
| Altova Maptool | Neietekmē kļūdu novēršanu | Nav pieejams pirmkods |
| Fivetran | Nav pieejamu konfigurāciju faili | Nav pieejams pirmkods |
| CloverDX desiner | Neietekmē kļūdu novēršanu | Pirmkodā veicamās izmaiņas nepietiekami ietekmē rīka procesu darbību. |

# REZULTĀTI

Darbā plaši aprakstīts ETL rīku darbība tās visos posmos, jeb savākšana transformācija un ielādēšana.

Darba rezultātā darba autors ir izstrādājis papildinājumus testa grupas ETL rīkos, kas spētu novērst vairākus kļūdas iespējas faktorus, kas saistīti ar datu avota izmaiņām vai datu trūkumu.

Darba teorētiskajā daļā darba autors aprakstījis ETL procesu un svarīgākos faktorus, kas ietekmē ETL procesa veiksmīgu norises procesu.

Darba praktiskās daļas beigās apkopotas metodes un attiecīgi norādīts kuros no rīkiem metodes ar kuriem no testiem metodes izdevušās.

# SECINĀJUMI

Ievadā izvirzītā hipotēze darba izstrādes laikā daļēji apstiprinājās – ETL rīkiem, kas paredzēti plašai lietošanai, ir iekšējas opcijas papildnoteikumu ieviešanai un pielāgošanai lietotāja vajadzībām.

ETL procesa datu ievākšanas solis ir ļoti komplicēts un situācijai specifisks, kas nozīmē, ka pārveidot šo soli globālā veidā, kas ietekmētu visas šī soļa parādīšanās izpausmes rīkā, ir teju neiespējama.

Vairums ETL rīku ir, veidoti ar domu, lai to procesi būtu modulāri nodalot katru iekšējo darbību atsevišķi, ļaujot konstruēt procesus lietotāja interesēs.

Konfigurācijas datnes, kas bija pieejamas rīku pielāgošanai, nesatur noteikumus, kas būtu spējīgi ietekmēt darbības procesa iespējamās kļūdas situācijas.

Tiešsaistes rīku piedāvātā funkcionalitāte ir ļoti ierobežota sakarā ar to, ka procesa darbības patērētie datu resursi nav atkarīgi no lietotāja iekārtas, kurā tiek izveidots ETL process.

Darba izstrādes laikā viens no grūtākajiem momentiem bija iegūt piekļuves tiesības vairākiem no ETL rīkiem, jo šo opciju bloķēja licenču trūkums vai, oficiāls nozares identifikācijas trūkums, un bieži rīku piekļuvi noteica tieša, individuāla pieteikuma izvērtēšana, kas nedeva pozitīvu iznākumu vairumā gadījumu.

Iespēja aplūkot, un rediģēt ETL procesu tiešā programmas koda veidā ir ļoti noderīga un ļauj izprast dažādos iekšējos procesus un iespējamās nepilnības sistēmā.

# IZMANTOTĀ LITERATŪRA


1. **Ralph Kimball**. TheData WarehouseETL Toolkit. Wiley Publishing, Inc. 10475 Crosspoint Boulevard Indianapolis, IN 46256. 2004. 29-113lpp.
2. **Matt C., Roland B., Jos van D.**. Pentaho Kettle Solutions Building Open Source ETL Solutions with Pentaho Data Integration. Wiley Publishing, Inc. 10475 Crosspoint Boulevard Indianapolis, IN 46256. 2010. 111-127lpp.
3. **ETL Tools And Processes 2019 Overview** [tiešsaiste] – [atsauce 05.05.2020.]. Pieejams internetā: https://datavirtuality.com/blog-etl-tools-and-processes/
4. **Core Concepts** [tiešsaiste] – [atsauce 08.05.2020.]. Pieejams internetā: https://fivetran.com/docs/getting-started/core-concepts
5. **Creating and Deploying Custom Components in CloverDX** [tiešsaiste] – [atsauce 28.04.2020.]. Pieejams internetā: https://www.cloverdx.com/blog/creating-custom-component
6. **Data Mapping Tools** [tiešsaiste] – [atsauce 05.05.2020.]. Pieejams internetā: https://www.altova.com/mapforce
7. **ELT components** [tiešsaiste] – [atsauce 15.04.2020.]. Pieejams internetā: https://help.talend.com/reader/jomWd_GKqAmTZviwG_oxHQ/8eRAWdyylAR2q4zFF_as4A
8. **ETL (Extract, Transform, and Load) Process** [tiešsaiste] – [atsauce 22.04.2020.]. Pieejams internetā: https://www.guru99.com/etl-extract-load-process.html
9. **Validating the Extract, Transform, Load Process Used to Populate a Large Clinical Research Database** [tiešsaiste] – [atsauce 01.05.2020.]. Pieejams internetā: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5556907/
10. **Extract, transform, load** [tiešsaiste] – [atsauce 12.05.2020.]. Pieejams internetā: https://en.wikipedia.org/wiki/Extract,_transform,_load

# PIELIKUMI

1. Pielikums. Testa dati darbinieku tabulai JSON datu formā


```
{"darbinieki":
[
 {
  "ID": 1,  "Vards": "Janis",  "Uzvards": "Kalnins",  "Darba_ID": 1,  "Alga": 600
 },
 {
  "ID": 2,  "Vards": "Peteris",  "Uzvards": "Ozolins",  "Darba_ID": 2,  "Alga": 500
 },
 {
  "ID": 3,  "Vards": "Martins",  "Uzvards": "Osis",  "Darba_ID": 3,  "Alga": 780
 },
 {
  "ID": 4,  "Vards": "Sems",  "Uzvards": "Klava",  "Darba_ID": 4,  "Alga": 560
 },
 {
  "ID": 5,  "Vards": "Laila",  "Uzvards": "Ziemelis",  "Darba_ID": 5,  "Alga": 340
 },
 {
  "ID": 6,  "Vards": "Zigis",  "Uzvards": "Kalnins",  "Darba_ID": 1,  "Alga": 890
 },
 {
  "ID": 7,  "Vards": "Maruta",  "Uzvards": "Ozolins",  "Darba_ID": 2,  "Alga": 780
 },
 {
  "ID": 8,  "Vards": "Tirza",  "Uzvards": "Osis",  "Darba_ID": 3,  "Alga": 560
 },
 {
  "ID": 9,  "Vards": "Memele",  "Uzvards": "Klava",  "Darba_ID": 4,  "Alga": 860
 },
 {
  "ID": 10,  "Vards": "Toms",  "Uzvards": "Ziemelis",  "Darba_ID": 5,  "Alga": 740
 },
 {
```

```
  "ID": 11,  "Vards": "Timijs",  "Uzvards": "Kalnins",  "Darba_ID": 1,  "Alga": 760
 },
 {
  "ID": 12,  "Vards": "Janis",  "Uzvards": "Ozolins",  "Darba_ID": 2,  "Alga": 450
 },
 {
  "ID": 13,  "Vards": "Peteris",  "Uzvards": "Osis",  "Darba_ID": 3,  "Alga": 330
 },
 {
  "ID": 14,  "Vards": "Martins",  "Uzvards": "Klava",  "Darba_ID": 4,  "Alga": 770
 },
 {
  "ID": 15,  "Vards": "Sems",  "Uzvards": "Ziemelis",  "Darba_ID": 5,  "Alga": 660
 },
 {
  "ID": 16,  "Vards": "Laila",  "Uzvards": "Kalnins",  "Darba_ID": 1,  "Alga": 940
 },
 {
  "ID": 17,  "Vards": "Zigis",  "Uzvards": "Ozolins",  "Darba_ID": 2,  "Alga": 370
 },
 {
  "ID": 18,  "Vards": "Maruta",  "Uzvards": "Osis",  "Darba_ID": 3,  "Alga": 110
 },
 {
  "ID": 19,  "Vards": "Tirza",  "Uzvards": "Klava",  "Darba_ID": 4,  "Alga": 440
 },
 {
  "ID": 20,  "Vards": "Memele",  "Uzvards": "Ziemelis",  "Darba_ID": 5,  "Alga": 360
 }
]
}
```

2.Pielikums. Testa dati darbavietu tabulai JSON datu formā


```
{
"darbavietas":
 [
 {
  "ID": 1,
  "Nosaukums": "FutureTec",
  "Nozare": "Tec"
 },
 {
  "ID": 2,
  "Nosaukums": "AlfaBiz",
  "Nozare": "Fiz"
 },
 {
  "ID": 3,
  "Nosaukums": "Omego",
  "Nozare": "Bio"
 },
 {
  "ID": 4,
  "Nosaukums": "Tesla",
  "Nozare": "Tec"
 }
]
}
```

Bakalaura darbs „ ETL rīku funkcionalitātes pielāgošanas iespējas datu avotu izmaiņām ” izstrādāts LU Datorikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: _____ Atis Ēriks Dumpis

Rekomendēju/~~nerekomendēju~~ darbu aizstāvēšanai *(nederīgo svītro vadītājs)*

Vadītājs: Dr.dat. Darja Solodovņikova _____ __.05.2020.

Recenzents: Doc. Maksims Kravcevs

Darbs iesniegts Datorikas fakultātē __.05.2020.

Dekāna pilnvarotā persona: vecākā metodiķe Ārija Sproģe _____

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē

____.06.2020. prot. Nr. _____

Komisijas sekretārs(-e): _____