

LATVIJAS UNIVERSITĀTES
DATORIKAS FAKULTĀTES

AUTOSERVISU DATU IMPORTS UN APSTRĀDE
KVALIFIKĀCIJAS DARBS

Autors: Einārs Koops

Studenta apliecības numurs: ek14029

Darba vadītājs: Docents Imants Gorbāns

RĪGA 2017

ANOTĀCIJA

Darba mērķis ir izveidot veidu, kā apdrošināšanas kompānijas X darbinieks var lejupielādēt jaunākos autoservisu un ekspertu datus, modificēt tos (pārskatāmāk nekā sākotnējā risinājumā) un visbeidzot augšupielādēt tos datu bāzē. Paralēli tam jāmaina autoservisu atlases mehānisms un tas jāpārceļ datu bāzē, lai to varētu lietot apdrošināšanas kompānijas mājaslapā.

Tika izskatīti dažādi iespējami risinājumi ar kuriem varētu veikt doto datu atjaunošanu. Visbeidzot veicot sarunas ar uzdevuma pasūtītāju un konsultējoties ar kolēģiem, tika izlemts izmantot CSV failus, kuri tiktu importēti Oracle formā, lai veiktu datu saglabāšanu datu bāzē. Paralēli šī risinājuma izstrādei, tika mainīts servisu atlases algoritms, kurš izmanto autoservisu un ekspertu datus.

Atslegvārdi: Oracle Form Builder, PL/SQL, JIRA, autoservisi, apdrošināšana

ABSTRACT

Import and processing of service station data

Work goal was to create a way for insurance company X worker to download newest transport service and expert data from database, modify them (in more convenient way than previous solution) and then upload them back into the database. At the same time need appeared to change transport service algorithm and transfer it into database, so that it could be used for insurance company homepage.

At start there were examined many different solutions with which data modification could be done. In the end it was decided, by talking with task requesters and by consulting with colleagues and, that CSV file will be used, which will be imported into Oracle Form, that way data would be saved into database. In the same time, algorithm for selecting transport services will be changed, which uses transport service and expert data.

Keywords: Oracle Form Builder, PL/SQL, JIRA, auto services, insurance

SATURS

APZĪMĒJUMI UN SKAIDROJUMI.....	1
1. IEVADS.....	2
2. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA.....	3
2.1. Ievads.....	3
2.1.1. Nolūks.....	3
2.1.2. Darbības sfēra.....	3
2.2. Vispārīgais apraksts.....	3
2.2.1. Sistēmas lietotāji.....	3
2.2.2. Lietotāja raksturiezīmes.....	3
2.2.3. Produkta funkcijas.....	3
2.2.4. Vispārējie ierobežojumi.....	4
2.3. Funkcionālās prasības.....	4
2.3.1. Lietotāju stāsti.....	4
2.4. Nefunkcionālās prasības.....	6
2.4.1. Ārējās saskarnes prasības.....	6
3. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS.....	7
3.1. Ievads.....	7
3.1.1. Nolūks.....	7
3.1.2. Darbības sfēra.....	7
3.1.3. Termins un pieņemtie apzīmējumi.....	7
3.1.4. Saistība ar citiem dokumentiem.....	7
3.2. Atkarību apraksts.....	7
3.2.1. Konceptuālais entītijas relāciju (ER) modelis.....	7
3.2.2. Fiziskais ER modelis.....	9
3.3. Detalizēts projektējuma apraksts.....	10
3.3.1. Datubāzes tabulas.....	10
3. TESTĒŠANAS DOKUMENTĀCIJA.....	30
3.1. Ievads.....	30
3.1.1. Nolūks.....	30
3.1.2. Saistības ar citiem dokumentiem.....	30
3.2. Testēšanas apraksts.....	30
3.3. Testēšanas žurnāls.....	30
3.3.1. Vienībtesti.....	30
3.3.2. Akcepttesti.....	31
3.3.3. Nefunkcionālo prasību testēšana.....	37

4. PROJEKTA PĀRVALDĪBA	38
4.1. Projekta organizācija.....	38
4.2. Izstrādes vide.....	38
4.3. Kvalitātes nodrošināšana.....	38
4.4. Konfigurācijas pārvaldība	38
4.5. Komandas organizācija	39
5. DARBIETILPĪBAS NOVĒRTĒŠANA	40
6. SECINĀJUMI	41
7. IZMANTOTĀ LITERATŪRA	42
9. Pielikums.....	43
1.pielikums. Autoservisu atlases forma.....	43
2.pielikums. Pakalpojumu sniedzēju datu atjaunošanas forma – datu eksports	43
3.pielikums. Pakalpojumu sniedzēju datu atjaunošanas forma – datu imports	44
4.pielikums. CSV faila datu struktūra un dati	44
5.pielikums. CSV faila datu struktūra un dati	44
6.pielikums. CSV faila datu struktūra un dati	45
7.pielikums. CSV faila datu struktūra un dati	45
8.pielikums. Pirmkoda piemērs no Oracle formas	45
9.pielikums. Pirmkoda piemērs no datubāzes	49

APZĪMĒJUMI UN SKAIDROJUMI

Tortoise SVN	Versiju kontroles pārvaldes rīks
Oracle Form Builder	Programmatūra paredzēta formu izveidei ar kurām var mijedarboties ar Oracle datubāzi
PL/SQL Developer	Apvienotā izstrādes vide priekš programmatūras izstrādes Oracle datubāzes vidē
TIA (The Insurance Application)	Apdrošināšanas lietojumprogrammatūra
JIRA	Projektu, izmaiņu pieprasījumu izstrādes un kļūdu administrēšanas pārvaldības rīks

1. IEVADS

Kvalifikācijas darba tēmu piedāvāja uzņēmums, kurā darba autors strādā. Dotais uzdevums bija turpinājums iepriekšējam projektam, kuru darba autors veica, līdz ar to darba autors uzsāka šī projekta realizāciju.

Uzņēmumā bija radusies vajadzība automatizēt servisu izvēlni, lai atvieglotu darbu apdrošināšanas kompānijas darbiniekiem, kā arī padarīt šo procesu automātisku, lai uzņēmuma klienti varētu izvēlēties servisu paši (no servisu saraksta, kuru apdrošināšanas kompānijas piedāvā). Kā arī bija radusies nepieciešamība atvieglot autoservisu un ekspertu datu modificēšanu.

Lai izpildītu projektam izvirzītos mērķus, bija jāveic vairāki uzdevumi. Jāpārceļ esošais servisu atlases mehānisms no formas uz datubāzi, tādējādi atļaut tā izmantošanu gan formā, gan apdrošināšanas kompānijas mājaslapā. Jāveic dotā algoritma izmaiņas, kuras atbilst pasūtītāja prasībām/vēlmēm. Jāatrod veids, kas ļauj pārskatāmi importēt, modificēt un eksportēt autoservisu datus. Jāveic atrastā risinājuma realizācija.

Darbs tika izstrādāts izmantojot spējo programmatūras izstrādi, jo šis izstrādes veids tiek izmantots dotajā apdrošināšanas kompānijā un uzdevuma pasūtītāji ir pieraduši pie tā, ka var tikt veiktas prasības izstrādes laikā.

Darbs sastāv no Programmatūras prasību specifikācijas, Programmatūras projektējuma apraksta, Testēšanas dokumentācijas, Projekta pārvaldības, Darbietilpības novērtēšana un Pielikuma.

2. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

2.1. Ievads

2.1.1. Nolūks

Programmatūras prasību specifikācijas mērķis ir apkopot un aprakstīt Autoservisu datu importa un apstrādes izveides prasības, funkcionalitāti un specifikāciju. Dotais dokuments ir paredzēts dotā produkta izstrādātājam, kurš veiks dotā produkta projektējumu un izstrādi.

2.1.2. Darbības sfēra

Autoservisu datu imports aizstās esošo risinājumu TIA, kurā lietotājam manuāli ir jāmeklē serviss, kuram lietotājs vēlas izmainīt konkrētus datus. Dotais risinājums atvieglos biežāk maināmo datu atjaunošanu, kā arī sniegs papildus informāciju par servisiem, kuru varēs izmantot to atlasē. Kā arī tiks mainīts servisu atlasē algoritms, kas sniegs šo algoritmu lietotājiem, to servisu sarakstu, kuri atbilstu apdrošināšanas kompānijas prasībām.

Programmprodukta mērķauditorija ir apdrošināšanas kompānijas darbinieki, kas nodarbojas ar transportlīdzekļu atlīdzību lietu administrēšanu un servisu datu atjaunošanu. Vienlaicīgi servisu atlasē algoritms, tiks lietots apdrošināšanas kompānijas pašapkalpošanas portālā.

2.2. Vispārīgais apraksts

2.2.1. Sistēmas lietotāji

Ar doto sistēmu var strādāt visi lietotāji, kuriem ir reģistrēts konts TIA sistēmā. Lai lietu formu, kura paredzēta autoservisu datu atjaunošanai ir nepieciešamas papildus tiesības, kuras nav visiem TIA sistēmas lietotājiem.

2.2.2. Lietotāja raksturiezīmes

Ar doto sistēmu var strādāt jebkurš lietotājs ar pamatzināšanām un iemaņām darbā ar datoru. Papildus tam lietotājam jābūt zināšanās par apdrošināšanu un tās niansēm dotajā apdrošināšanas kompānijā, lai lietotājs spētu lietderīgi izmantot TIA.

2.2.3. Produkta funkcijas

- 1) Veikt pašreizējo autoservisu datu eksportu
- 2) Veikt pašreizējo ekspertu datu eksportu
- 3) Veikt autoservisu vēsturisko datu eksportu
- 4) Veikt ekspertu vēsturisko datu eksportu
- 5) Veikt jaunāko autoservisu datu importu
- 6) Veikt jaunāko ekspertu datu importu

- 7) Autoservisu atlase
- 8) Ekspertu atlase
- 9) Saglabāt esošo funkcionalitāti

2.2.4. Vispārējie ierobežojumi

Darbam ar sistēmu TIA ir nepieciešams dators ar pieeju apdrošināšanas kompānijas iekšējam tīklam, ka arī jābūt nodrošinātam savienojumam ar serveri, kurā ir uzstādīta TIA. Ja tiek lietots apdrošināšanas kompānijas pašapkalpošanas portāls, tad ir nepieciešams interneta savienojums.

2.3. Funkcionālās prasības

2.3.1. Lietotāju stāsti

2.3.1.1. Ekspertu pazīme

Es kā TIA lietotājs vēlos, lai ekspertiem, tiktu pievienota pazīme, kas atļauj atšķirt autoservisu ekspertu no īpašuma eksperta.

Akceptēšanas kritērijs – Ekspertiem var pievienot pazīmi, kas atļauj atšķirt auto ekspertu no īpasuma ekspertu.

2.3.1.2. Vēsturisko autoservisu datu iegūšana

Es kā TIA lietotājs vēlos, lai es spētu iegūt iepriekš importēto autoservisu datus.

Akceptēšanas kritērijs – Atverot Pakalpojumu Sniedzēju Datu Atjaunošanas formu, tiek parādīts saraksts ar iepriekš veiktajiem autoservisu importiem un izvēloties attiecīgo importu tiek veikts korekts datu eksports.

2.3.1.3. Vēsturisko ekspertu datu iegūšana

Es kā TIA lietotājs vēlos, lai es spētu iegūt iepriekš importēto ekspertu datus.

Akceptēšanas kritērijs – Atverot Pakalpojumu Sniedzēju Datu Atjaunošanas formu, tiek parādīts saraksts ar iepriekš veiktajiem ekspertu importiem un izvēloties attiecīgo importu tiek veikts korekts datu eksports.

2.3.1.4. Iepriekšējās funkcionalitātes saglabāšana – atlase pēc manuāli ievadīties datiem

Es kā TIA lietotājs vēlos, lai tiktu saglabāta iepriekšējā funkcionalitāte. Es varu servisu atlases formas attiecīgajos laukos manuāli ievadīt vērtības, pēc kuriem tiktu veikta attiecīgā datu atlase.

Akceptēšanas kritērijs – Atverot servisu atlases formu lietotājs var manuāli ievadīt/mainīt informāciju attiecīgajos laukos, pēc kuriem var tikt veikta veiksmīga datu atlase.

2.3.1.5. Iepriekšējās funkcionalitātes saglabāšana – automātiska datu atlase

Es kā TIA lietotājs vēlos, lai tikt saglabāta iepriekšējā funkcionalitāte. Atveros servisu atlases formu, tiktu automātiski ievietota attiecīgā transportlīdzekļa informācija.

Akceptēšanas kritērijs – Atverot doto formu tiek automātiski nolasīta un ievietota informācija attiecīgajos laukos par konkrēto transportlīdzekli, pēc kura tiek veikta veiksmīga datu atlase.

2.3.1.6. Eksperta apgabala izmantošana

Es kā TIA lietotājs vēlos, lai izmantojot formu priekš ekspertu atlases, tiktu automātiski iegūta transportlīdzekļa reģistrācijas apgabals, pēc kura tiktu veikts korektra ekspertu datu atlase.

Akceptēšanas kritērijs – Atverot doto formu tiek automātiski nolasīta un ievietota informācija attiecīgajā laukā par konkrēto transportlīdzekli, pēc kura tiek veikta veiksmīga datu atlase.

2.3.1.7. Vecās atlases mehānisms

Es kā TIA lietotājs vēlos, lai pastāvētu veids pēc kura var veikt servisu atlasī izmantojot sākotnējo autoservisu atlases mehānismu.

Akceptēšanas kritērijs- Atverot doto formu var manuāli izslēgt esošo autoservisu atlases mehānismu, kā vietā servisu atlase notiks pēc vecā servisa mehānisma.

2.3.1.8. Autoservisu marku vecumu ierobežojumi

Es kā TIA lietotājs vēlos, lai tiktu ieviests papildus vecuma ierobežojums katrai transporta markai un šis ierobežojums var atšķirties katram servisam. Šim vecuma ierobežojumam jātiek pielietotam servisu atlases mehānismā.

Akceptēšanas kritērijs – Veicot servisu atlases mehānismu visi atlasītie servisi nepārkāpj vecuma ierobežojumus attiecīgajai transporta markai.

2.3.1.9. Autoservisu datu eksports/imports

Es kā TIA lietotājs vēlos, lai būtu veids kā vienuviet varētu mainīt autoservisu datus un varētu tos viegli importēt datu bāzē.

Akceptēšanas kritērijs – Izmantojot Pakalpojumu Sniedzēju Datu Atjaunošanas formu tiek iegūts CSV fails, kurā lietotājs var modificēt autoservisu datus. Izmantojot šo pašu formu dotā CSV faila datus importē datu bāzē.

2.3.1.10. Ekspertu datu eksports/imports

Es kā TIA lietotājs vēlos, lai būtu veids kā vienuviet varētu mainīt ekspertu datus un varētu tos viegli importēt datu bāzē.

Akceptēšanas kritērijs – Izmantojot Pakalpojumu Sniedzēju Datu Atjaunošanas formu tiek iegūts CSV fails, kurā lietotājs var modificēt ekspertu datus. Izmantojot šo pašu formu dotā CSV faila datus importē datu bāzē.

2.3.1.11. Vienots atlasē mehānisms

Es kā TIA lietotājs vēlos, lai veicot servisu atlasē TIA un apdrošināšanas kompānijas interneta lapā tiktu lietots viens un tāds pats atlasē veids. Interneta lapā var tikt izmantoti vēl stingrāki atlasē kritēriji, līdz ar TIA atlasē var būt servisi, kuri neparādās apdrošināšanas kompānijas lapā.

Akceptēšanas kritērijs – Izmantojot TIA un apdrošināšanas kompānijas interneta lapu tiek lietots vienāds servisu atlasē mehānisms.

2.3.1.12. Transportlīdzekļu svārs

Es kā TIA lietotājs vēlos, lai atlasot datus priekš servisa atlasē mehānisma, tiktu ņemts vērā transportlīdzekļa svārs. Gadījumā ja transportlīdzekļa svārs nēsasniedz noteiktu vērtību tad tiktu pieņemta noteikts transportlīdzekļa tips dotajā servisu atlasē mehānismā.

Akceptēšanas kritērijs – Veicot servisu atlasē mehānismu tiek izmantots transportlīdzekļu svārs. Gadījumā ja svārs nēsasniedz noteiktu vērtību tiek izmantots noteikts transportlīdzekļa svārs dotajā servisu atlasē mehānismā.

2.4. Nefunkcionālās prasības

2.4.1. Ārējās saskārnēs prasības

2.4.1.1. Pārskatāmība

Formu dizainiem jābūt viegli pārskatāmiem un stilistiski līdzīgiem citām TIA formām.

3. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

3.1. Ievads

3.1.1. Nolūks

Šajā dokumenta sadaļā tiek attēlots programmatūras prasību specifikācijas (PPS) noteikto funkcionālo un nefunkcionālo prasību realizācija.

3.1.2. Darbības sfēra

Atsauce uz 1.1.2.

3.1.3. Terminu un pieņemtie apzīmējumi

Skatīt 1. Lapaspusi – Apzīmējumi un Skaidrojumi

3.1.4. Saistība ar citiem dokumentiem

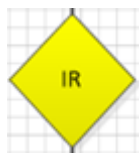
Dotais dokuments ir veidots pēc PPS.

3.2. Atkarību apraksts

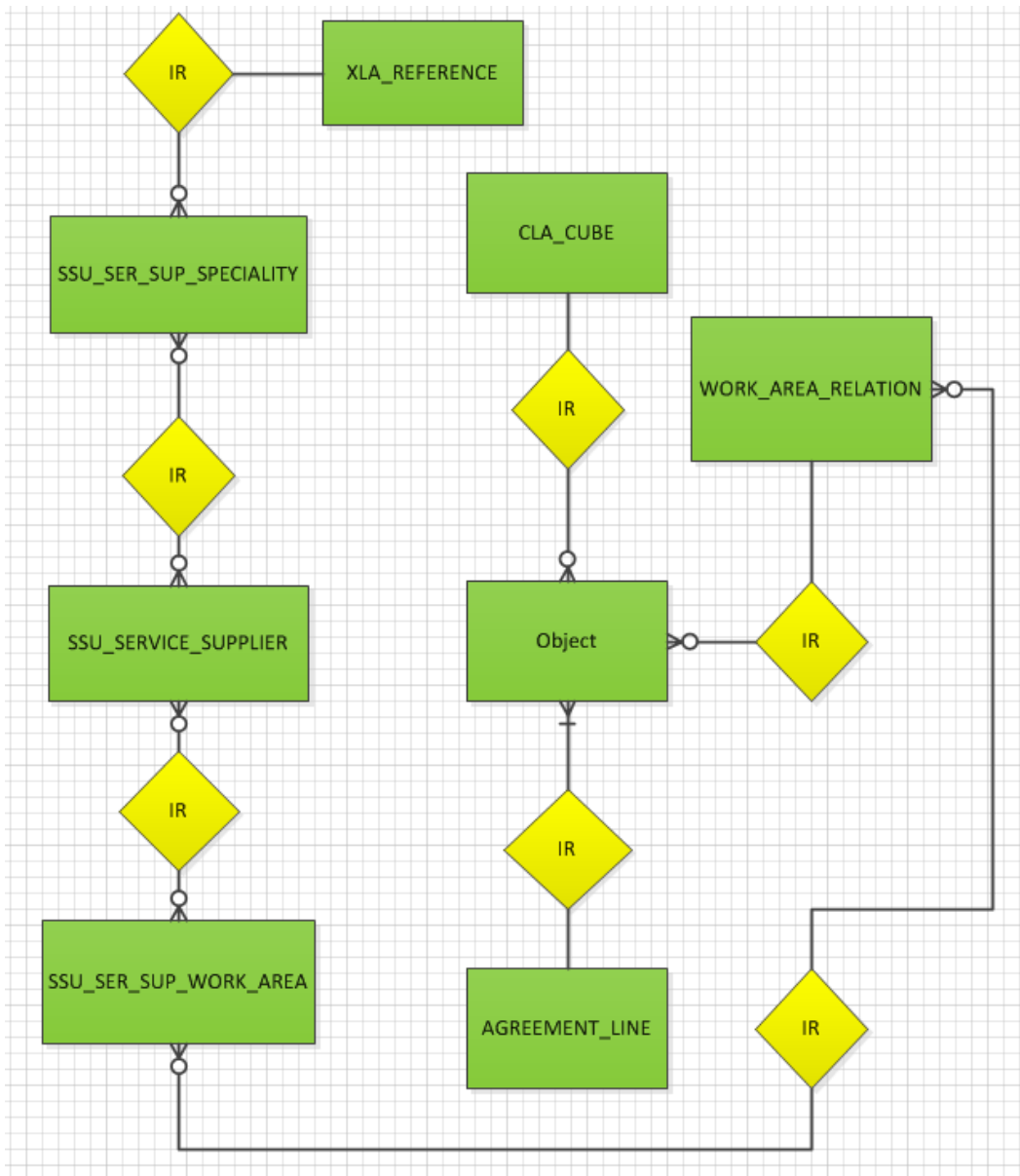
3.2.1. Konceptuālais entītiņu relāciju (ER) modelis



2.1. att. Datubāzes entītija



2.2. att. Datubāzes entītiņu savstarpējā attiecība



2.3. att. Konceptuālais ER modelis

3.2.2. Fiziskais ER modelis

Column	Type	Nullable	Default	Comments
NAME_ID_NO	NUMBER(10)			
SERVICE_SUPPLIER_TYPE	VARCHAR2(3)			Type of service
ALIAS	VARCHAR2(10)			Alias identification
DESCRIPTION	VARCHAR2(2000)	Y		A description of the person in the data from 1
CONTACT_PERSON	VARCHAR2(32)	Y		
START_DATE	DATE	Y		

Column	Type	Nullable	Default	Comments
IMPORT_ID	NUMBER			
SPECIALTY_TYPE	VARCHAR2(1)			
IMPORTED_FILE_TYPE	VARCHAR2(1)			
STATUS	VARCHAR2(1)			
USERID	VARCHAR2(8)			
TIMESTAMP	DATE	Y		sysdate
START_DATE	DATE	Y		

Column	Type	Nullable	Default	Comments
NAME_ID_NO	NUMBER(10)			
SPECIALTY_TYPE	VARCHAR2(3)			
TIMESTAMP	DATE	Y		
USERID	VARCHAR2(8)			
RECORD_VERSION	NUMBER			
PRIORITY	NUMBER			

Column	Type	Nullable	Default	Comments
CUST_NO	NUMBER(10)			
AGR_LINE_NO	NUMBER(10)			
OBJECT_NO	NUMBER(10)			
SKQ_NO	NUMBER(10)			
OBJECT_TYPE	VARCHAR2(12)			
TYPE_ID	VARCHAR2(12)			

Column	Type	Nullable	Default	Comments
NAME_ID_NO	NUMBER(10)			
ALIAS	VARCHAR2(10)			
DESCRIPTION	VARCHAR2(2000)	Y		
CONTACT_PERSON	VARCHAR2(64)	Y		
WORK_TIME	VARCHAR2(2000)	Y		

Column	Type	Nullable	Default	Comments
CLA_CASE_NO	NUMBER(10)			
CLA_SUBCASE_NO	VARCHAR2(10)	Y		
CLA_STATUS	VARCHAR2(10)	Y		
HANDLER	VARCHAR2(8)	Y		
C01	VARCHAR2(2000)	Y		
C02	VARCHAR2(2000)	Y		

Holds a list of item related to a claim case or a subcase. A claim assistant will use

Column	Type	Nullable	Default	Comments
RELATED_NAME	VARCHAR2(30)			
USERID	VARCHAR2(8)			
LOCATION_NAME	VARCHAR2(30)			
TIMESTAMP	DATE	Y		
RELATION_PK	Column(s)			
LOCATION_NAME	Column(s)			
RELATED_NAME	Column(s)			

Column	Type	Nullable	Default	Comments
NAME_ID_NO	NUMBER(10)			
WORK_AREA	VARCHAR2(10)			
TIMESTAMP	DATE	Y		
USERID	VARCHAR2(8)			
RECORD_VERSION	NUMBER			
PRIORITY	NUMBER			

Column	Type	Nullable	Default	Comments
PROGRAM_NAME	VARCHAR2(30)			
LANGUAGE	VARCHAR2(3)			
ITEM_NAME	VARCHAR2(250)			
ITEM_TYPE	VARCHAR2(3)			
ITEM_LABEL	VARCHAR2(250)			
GLOBAL_LABEL	VARCHAR2(1)			
IN_USE	VARCHAR2(1)			
TIMESTAMP	DATE			
USERID	VARCHAR2(8)			
LABEL_LENGTH	NUMBER(3)			
GROUP_NAME	VARCHAR2(30)			
RECORD_VERSION	NUMBER			
BLOCK_NAME	VARCHAR2(250)			
T8704_2ND_IDX	Index			
ITEM_NAME	Column(s)			
LANGUAGE	Column(s)			
ITEM_LABEL	Column(s)			

Column	Type	Nullable	Default	Comments
TABLE_NAME	VARCHAR2(30)			
LANGUAGE	VARCHAR2(3)			
CODE	VARCHAR2(10)			
DISPLAY_CODE	VARCHAR2(10)			
DESCRIPTION	VARCHAR2(60)			

Column	Type	Nullable	Default	Comments
TABLE_NAME	VARCHAR2(30)			
DIM01	VARCHAR2(10)			
DIM02	VARCHAR2(10)			
DIM03	VARCHAR2(10)			
DIM04	VARCHAR2(10)			
DIM05	VARCHAR2(10)			
DIM06	VARCHAR2(10)			

Column	Type	Nullable	Default	Comments
CLA_CASE_NO	NUMBER(10)			
THIRD_PARTY_NO	NUMBER(10)	Y		Identification of the third party
CLA_SUBCASE_NO	NUMBER(10)			Unique identification of a subcase
SUBCASE_TYPE	VARCHAR2(2000)	Y		Description of a subcase
CLA_CLASS	VARCHAR2(10)	Y		Classification code for a subcase
QUESTION_CLASS	VARCHAR2(10)	Y		Classification code for a question
QUESTION_STATUS	VARCHAR2(3)	Y		Code indicating whether a question is active
TASKLIST_CLASS	VARCHAR2(10)	Y		Classification code for a tasklist

Column	Type	Nullable	Default	Comments
NAME_ID_NO	NUMBER(10)			
WORK_AREA	VARCHAR2(10)			
TIMESTAMP	DATE	Y		
USERID	VARCHAR2(8)			
RECORD_VERSION	NUMBER			
PRIORITY	NUMBER			

2.4. att. Fiziskais ER modelis

3.3. Detalizēts projektējuma apraksts

3.3.1. Datubāzes tabulas

3.3.1.1. Tabula "Object"

Tabula paredzēta informācijas uzglabāšanai un iegūšanai par doto objektu, šīs dokumentācijas kontekstā par transportlīdzekli.

Jau esoša apdrošināšanas kompānijas tabula, kuru šī darba autors nav veidojis un/vai modificējis.

2.1. tabula

Kolonna	Datu tips	Cita informācija	Komentārs
CUST_NO	Number(10)		Klienta numurs
AGR_LINE_NO	Number(10)		Norāda uz polises apakšlīniju uz kuru attiecas dotais objekts
NEWEST	Varchar2(1)		Norāda vai dotajā ierakstā ir jaunākā informācija
OBJECT_TYPE	Varchar2(12)	NOT NULL	Norāda kāda veida objekts tiek aprakstīts dotajā ierakstā
C10	Varchar2(2000)		Apraksta lauks, dotās dokumentācijas kontekstā tiek glabāts pasta indekss
N03	Number		Apraksta lauks, dotās dokumentācijas kontekstā tiek glabāts automašīnas svars. Attiecas tikai uz KASKO lietām.
C13	Varchar2(2000)		Apraksta lauks, dotās dokumentācijas kontekstā tiek glabāta

			informācija par transportlīdzekļa marku
C17	Varchar2(2000)		Apraksta lauks, dotās dokumentācijas kontekstā tiek glabāta informācija par transportlīdzekļa reģistrācijas numuru. Attiecas tikai uz KASKO lietām.
C16	Varchar2(2000)		Apraksta lauks, dotās dokumentācijas kontekstā tiek glabāta informācija par transportlīdzekļa tipu (vieglā, motocikls, traktors...)
C36	Varchar(2000)		Apraksta lauks, dotās dokumentācijas kontekstā tiek glabāta apgabala kods. Attiecas tikai uz KASKO lietām.
N59	Number		Apraksta lauks, dotā dokumenta kontekstā glabā informāciju par to vai klientam pienākas dīler servisu pakalpojumi, vai arī klients pats var izvēlēties servisu. Attiecas tikai uz KASKO lietām.

C34	Varchar2(2000)		Apraksta lauks, dotā dokumenta kontekstā glabā transportlīdzekļa izlaišanas gadu. Attiecas tikai uz KASKO lietām.
D04	Date		Apraksta lauks, dotā dokumenta kontekstā glabā transportlīdzekļa reģistrācijas datumu formā 'DD.MM.YYYY'. Attiecas tikai uz KASKO lietām.
D18	Date		Apraksta lauks, dotā dokumenta kontekstā glabā transportlīdzekļa reģistrācijas datumu formā 'DD.MM.YYYY'. Attiecas tikai uz OCTA lietām.
N13	Date		Apraksta lauks, dotā dokumenta kontekstā glabā transportlīdzekļa izlaišanas gadu. Tiek glabāts formātā 'YYYY'. Attiecas tikai uz OCTA lietām.

TIMESTAMP	Date		Norāda, kad ir izveidots dotais ieraksts vai arī kad ir pēdējo reizi modificēts. Tiek glabāts formātā 'DD.MM.YYYY hh.mm.ss'
USERID	Varchar2(8)		Norāda uz lietotāju, kurš ir izveidojis/rediģējis doto ierakstu

3.3.1.2. Tabula "AGREEMENT_LINE"

Tabula paredzēta, lai uzglabātu informāciju par polises apakšlīgumu punktiem.

Jau esoša APDROŠINĀŠANAS KOMPĀNIJAS tabula, kuru šī darba autors nav veidojis un/vai modificējis.

2.2. tabula

Kolonna	Datu tips	Cita informācija	Komentārs
AGR_LINE_NO	Number(10)	NOT NULL	Polises līnijas numurs
POLICY_NO	Number(10)	NOT NULL	Polises numurs uz kuru attiecas dotais ieraksts
PRODUCT_LINE_ID	Varchar2(5)	NOT NULL	Norāda uz to, kāda veida apdrošināšana ir dotajā ierakstā
NEWEST	Varchar2(1)		Norāda vai dotajā ierakstā ir jaunākā informācija
COVER_START_DATE	Date		Norāda no kura datuma sākot dotais

			objekts tiek apdrošināts. Datuma formāts 'DD.MM.YYYY'
COVER_END_DATE	Date		Norāda līdz kuram datumam dotais objekts ir apdrošināts. Datuma formāts 'DD.MM.YYYY'
TIMESTAMP	Date		Norāda, kad ir izveidots dotais ieraksts vai arī kad ir pēdējo reizi modificēts. Tiek glabāts formātā 'DD.MM.YYYY hh.mm.ss'
USERID	Varchar2(8)		Norāda uz lietotāju, kurš ir izveidojis/rediģējis doto ierakstu

3.3.1.3. Tabula "SSU_SERVICE_SUPPLIER"

Tabula paredzēta, lai uzglabātu informāciju par pakalpojuma sniedzējiem.

Jau esoša APDROŠINĀŠANAS KOMPĀNIJAS tabula, kuru šī darba autors nav veidojis un/vai modificējis.

2.3. tabula

Kolonna	Datu tips	Cita informācija	Komentārs
NAME_ID_NO	Number(10)	NOT NULL	Unikāls ID numurs pēc kura var iegūt sīkāku informāciju par servisu

ALIAS	Varchar2(10)	NOT NULL	Dotā pakalpojuma sniedzēja vārda saīsinājums
SERVICE_SUPPLIER_TYPE	Varchar2(3)	NOT NULL	Dotā vērtība nosaka pakalpojuma sniedzēja tipu
DESCRIPTION	Varchar2(2000)		Papildus apraksts pa doto pakalpojumu sniedzēju brīvā veidā
GROUP_ID	Int	NOT NULL	Norāda, kurā grupā konkrētais pakalpojumu sniedzējs atrodas
PRIORITY	Number		Norāda prioritāti dotajam servisam
CONTACT_PERSON	Varchar2(64)		Tiek glabāti telefonnumuri, vārdi un uzvārdi, lai varētu sazināties ar doto servisu
QUEUE	Varchar2(3)		Norāda aptuveno dienu skaitu cik ilgā laikā pakalpojuma sniedzējs varēs apkalpot klientu
STARS	Varchar2(1)		Skaitlis no 1 līdz 3. Norāda to, cik ļoti apdrošināšanas kompānija iesaka doto autoservisu.

START_DATE	Date	NOT NULL	Datumu, kad sāka sadarbību ar doto pakalpojuma sniedzēju. Dati glabāsies 'DD.MM.YYYY' formā
END_DATE	Date		Datums, kad beidza sadarbību ar doto pakalpojumu sniedzēju ja nav ievadīts datums tad sadarbība vēl projām turpinās. Dati glabāsies 'DD.MM.YYYY' formā
BLACKLISTED	Varchar2(1)	NOT NULL	Norāda vai dotais serviss atrodas melnajā sarakstā
BLAKLISTED_REASON	Varchar2(2000)		Norāda iemeslu, kādēļ serviss atrodas melnajā sarakstā
USER_ID	Varchar2(8)		Kurš lietotājs ir pēdējo reizi modificējis doto ierakstu
TIMESTAMP	Date		Norāda, kad ir izveidots dotais ieraksts vai arī kad ir pēdējo reizi modificēts. Tiek glabāts formātā

			'DD.MM.YYYY hh.mm.ss'
RECORD VERSION	Number		Norāda versijas numuru

2.3.1.4. Tabula "SSU_SER_SUP_SPECIALITY"

Tabulā paredzēta, lai tajā uzglabātu informāciju par servisu sniegtajiem pakalpojumiem.

Jau esoša APDROŠINĀŠANAS KOMPĀNIJAS tabula, kuru šī darba autors nav veidojis un/vai modificējis.

2.4. tabula

Kolonna	Datu tips	Cita informācija	Komentārs
NAME_ID_NO	Number(10)	NOT NULL	Norāda uz konkrētu autoservisu tabulā SSU_SERVICE_SUPPLIER
SPECIALITY_TYPE	Varchar2(3)	NOT NULL	Norāda kāda veida pakalpojumus dotais serviss sniedz, atšifrējums ir tabulā XLA_REFERENCES
TIMESTAMP	Date		Norāda kad ir izveidots/modificēts pēdējais ieraksts. Tiek glabāts formātā 'DD.MM.YYYY'
USERID	Varchar2(8)		Norāda uz lietotāju, kurš ir izveidojis/rediģējis doto ierakstu
RECORD_VERSION	Number		Norāda versijas numuru
PRIORITY	Number		Norāda prioritāti dotajam pakalpojumam

2.3.1.5. Tabula "XLA_REFERENCE"

Tabula paredzēta, lai tajā uzglabātu virtuālās tabulas. Dotā dokumenta kontekstā uzglabā kodus un to apzīmējumus, kuri tiek lietoti SSU_SER_SUP_SPECIALITY, CLA_CUBE.

Jau esoša APDROŠINĀŠANAS KOMPĀNIJAS tabula, kuru šī darba autors nav veidojis un/vai modificējis.

2.5. tabula

Kolonna	Datu tips	Cita informācija	Komentārs
TABLE_NAME	Varchar2(30)	NOT NULL	Virtuālās tabulas vārds
LANGUAGE	Varchar2(3)	NOT NULL	Valodas kods
CODE	Varchar2(10)	NOT NULL	Ieraksta kods
DESCRIPTION	Varchar2(60)		Ieraksta koda apraksts
MISC_VALUE	Varchar2(60)		Dotās dokumentācijas kontekstā tiek uzglabāti vecuma intervāli.
START_DATE	Date		Datuma formāts 'DD.MM.YYYY'
END_DATE	Date		Datuma formāts 'DD.MM.YYYY'
TIMESTAMP	Date		Norāda uz to, kad ir izveidots dotais ieraksts vai arī, kad tas ir pēdējo reizi modificēts. Tiek glabāts formātā 'DD.MM.YYYY hh.mm.ss'
USERID	Varchar2(8)		Norāda uz lietotāju, kurš ir izveidojis/redīgējis

			doto ierakstu
SORT_NO	Number(10)		Satur informāciju par secību, kā ir jāsakārto dotie ieraksti
HELP_TEXT	Varchar2(2000)		Satur papildus informāciju par doto ierakstu, kura varētu noderēt nākotnē
RECORD_VERSION	Number		Norāda versijas numuru

2.3.1.6. Tabula “VZD_FULL_ADDRESSES”

Tabula paredzēta, lai varētu iegūt pilsētu un/vai novadu nosaukumu izmantojot pasta indeksu no OBJECT tabulas C10 lauka.

Jau esoša APDROŠINĀŠANAS KOMPĀNIJAS tabula, kuru šī darba autors nav veidojis un/vai modificējis.

2.6. tabula

Kolonna	Datu tips	Cita informācija	Komentārs
LOWESTLEVELID	Number(20)	NOT NULL	Dotā ieraksta ID
LOWESTLEVELTYPE	Number(3)		Norāda uz to, kāda veida teritorija ir dotais ieraksts (pilsēta, ciems, novads...)
LOWESTLEVELSORTNAME	Varchar2(516)		Teritorijas pilnais nosaukums
ISLASTCHILD	Char(1)		Norāda uz to

			vai dotajā teritorijā ir vecāks kādai citai teritorijai
APPROVED	Varchar2(1)		Norāda uz to vai dotā teritorija ir tikusi apstiprināta
CITY_NAME	Varchar2(516)		Norāda uz pilsētas nosaukumu, kurā dotais ieraksts ietilpst
NOVADSNAME	Varchar2(516)		Norāda uz novada nosaukumu, kurā dotais ieraksts ietilpst.

2.3.1.7. Tabula "TB5_WORK_AREA_RELATION"

Tabula paredzēja, lai uzglabātu saites starp dažādām atrašanās vietām.

Darba autora veidotā tabula.

2.7.tabula

Kolonna	Datu tips	Cita informācija	Komentārs
LOCATION_NAME	Varchar2(30)	NOT NULL	Vietas/teritorijas nosaukums, kuru sasaista ar RELATED_NAME
RELATED_NAME	Varchar2(30)	NOT NULL	Vietas/teritorijas nosaukums ar kuru tiek sasaistīts LOCATION_NAME

NEWEST	Varchar2(1)		Norāda vai dotajā ierakstā ir jaunākā informācija
USERID	Varchar2(8)		Norāda uz lietotāju, kurš ir izveidojis/rediģējis doto ierakstu
TIMESTAMP	Date	NOT NULL	Norāda kad ir izveidots/modificēts pēdējais ieraksts. Tiek glabāts formātā 'DD.MM.YYYY'

2.3.1.8. Tabula "SSU_SER_SUP_WORK_AREA"

Tabula paredzēta, lai uzglabātu informāciju par pakalpojumu sniedzēju darba apgabaliem, kurus tie apkalpo.

Jau esoša APDROŠINĀŠANAS KOMPĀNIJAS tabula, kuru šī darba autors nav veidojis un/vai modificējis.

2.8. tabula

Kolonna	Datu tips	Cita informācija	Komentārs
NAME_ID_NO	Number(10)	NOT NULL	Norāda uz kuru servisu dotais ieraksts attiecās. Atsaucās uz tabulas SSU_SERVICE_SUPPLIER kolonnu NAME_ID_NO
WORK_AREA	Varchar2(10))	NOT NULL	Norāda apgabalu, kurā dotais serviss strādā
TIMESTAMP	Date		Norāda, kad ir izveidots dotais ieraksts vai arī kad ir pēdējo reizi modificēts. Tiek glabāts formātā

			'DD.MM.YYYY hh.mm.ss'
USERID	Varchar2(8)		Norāda uz lietotāju, kurš ir izveidojis/rediģējis doto ierakstu
RECORD_VERSION S	Number		Norāda versijas numuru
PRIORITY	Number		Norāda ar kādu prioritāti dotais serviss strādā šajā apgabalā

2.3.1.9. Tabula "CLA_CUBE"

Tabula paredzēta, lai varētu no CSDD transporta apzīmējumiem un apdrošināšanas kompānijas transporta apzīmējumiem iegūt vienotus apzīmējumus, kuri tiek izmantoti SSU_SER_SUP_SPECIALITY.

Jau esoša APDROŠINĀŠANAS KOMPĀNIJAS tabula, kuru šī darba autors nav veidojis un/vai modificējis.

2.9. tabula

Kolonna	Datu tips	Cita informācija	Komentārs
TABLE_NAME	Varchar2(30)	NOT NULL	Virtuālās tabulas nosaukums
DIM01	Varchar2(10)		Virtuālās tabulas kolonna, dotā darba kontekstā norāda uz produkta kodu
DIM02	Varchar2(10)		Virtuālās tabulas kolonna, dotā darba kontekstā norāda uz kodu object tabulā, kurš apzīmē transporta tipu
DIM03	Varchar2(10)		Virtuālās tabulas kolonna, dotā darba

			kontekstā norāda uz kodu/kodiem xla_reference tabulā
DIM04	Varchar2(10)		Virtuālās tabulas kolonna, dotā darba kontekstā norāda uz kodu/kodiem xla_reference tabulā
TIMESTAMP	Date		Norāda, kad ir izveidots dotais ieraksts vai arī kad ir pēdējo reizi modificēts. Tiek glabāts formātā 'DD.MM.YYYY hh.mm.ss'
USERID	Varchar2(8)		Norāda uz lietotāju, kurš ir izveidojis/rediģējis doto ierakstu
RECORD_VERSION	Number		Norāda versijas numuru

2.3.1.10. Tabula "XLA_PGM_ITEM_LABEL"

Tabula paredzēta, lai nodrošinātu formām valodas maiņas funkcionalitāti.

Jau esoša APDROŠINĀŠANAS KOMPĀNIJAS tabula, kuru šī darba autors nav veidojis un/vai modificējis, tikai papildinājis.

2.10. tabula

Kolonna	Datu tips	Cita informācija	Komentārs
PROGRAM_NAME	Varchar2(30)	NOT NULL	Norāda uz programmas nosaukumu, kur dotais ieraksts tiks izmantots
LANGUAGE	Varchar2(3)		Valodas kods
ITEM_NAME	Varchar2(250)	NOT NULL	Norāda uz vienuma nosaukumu dotajā programmā
ITEM_TYPE			Norāda uz vienuma tipu dotajā programmā
ITEM_LABEL	Varchar2(250)		Norāda uz vienuma vērtību dotajā programmā
GLOBAL_LABEL	Varchar2(1)		Vajadzības gadījumā norāda uz konkrētu grupu uz kuru elementiem ir jāattiecas dotais tulkojums
IN_USE	Varchar2(1)		Norāda uz to vai dotais ieraksts tiek lietots
TIMESTAMP	Date		Norāda, kad ir izveidots dotais ieraksts vai arī kad ir pēdējo reizi modificēts. Tiek glabāts formātā 'DD.MM.YYYY hh.mm.ss'
USERID	Varchar2(8)		Norāda uz lietotāju,

			kurš ir izveidojis/rediģējis doto ierakstu
LABEL_LENGTH	Number(3)		Norāda garumu ITEM_LABELK
GROUP_NAME	Varchar2(30)		Norāda pie kuras grupas ieraksts attiecas
RECORD_VERSION	Number		Norāda versijas numuru
BLOCK_NAME	Varchar2(250)		Norāda uz to, kurā blokā dotajam vārdam ir jāatrodas

2.3.1.11. Tabula "TB5_SERVICE_LOG"

Tabula paredzēta, lai uzglabātu datus par notikušajiem datu importa notikumiem.

Autora veidota tabula.

2.11. tabula

Kolonna	Datu tips	Cita informācija	Komentārs
IMPORT_ID	Number	NOT NULL	Norāda importa numuru, kurš pēc kārtas ir ticis veikts
IMPORTED_FILE_TYPE	Varchar2(1)	NOT NULL	Norāda kāda veida dati tiek importēti. 1 – Servisu dati, 2 – Ekspertu dati
STATUS	Varchar2(1)	NOT NULL	Norāda, kādā statusā ir dotais imports. 1 – Imports ir sācies, nav tikuši ielādēti nekādi dati 2 – Kļūda veicot datu importu TB5_SERVICE_TEMP 3 – Datu imports TB5_SERVICE_TEMP

			<p>veiksmīgi veikts</p> <p>4- Sācies datu imports uz TB5_SERVICE_HISTORY</p> <p>5 – Kļūda veicot datu importu uz TB5_SERVICE_HISTORY</p> <p>6- Datu imports uz TB5_SERVICE_HISTORY</p> <p>veiksmīgi izdarīts</p> <p>7 – Sācies datu imports uz SSU_SERVICE_SUPPLIER, SSU_SER_SUP_SPECIALIT Y, SSU_SER_SUP_WORK_AR EA</p> <p>8 – Kļūda veicot datu importu uz SSU_SERVICE_SUPPLIER, SSU_SER_SUP_SPECIALIT Y, SSU_SER_SUP_WORK_AR EA</p> <p>9 – Datu imports uz SSU_SERVICE_SUPPLIER, SSU_SER_SUP_SPECIALIT Y, SSU_SER_SUP_WORK_AR EA veiksmīgi veikts</p>
USERID	Varchar(8)	NOT NULL	Norāda uz lietotāju, kurš ir izveidojis/rediģējis doto ierakstu
TIMESTAMP	Date	NOT NULL	Norāda, kad ir izveidots dotais ieraksts. Tiek glabāts formātā

			'DD.MM.YYYY hh.mm.ss'
START_DATE	Date		Norāda, kad ir sācies datu imports. Tiek glabāts formātā 'DD.MM.YYYY hh.mm.ss'
END_DATE	Date		Norāda, kad ir beidzies datu imports. Tiek glabāts formātā 'DD.MM.YYYY hh.mm.ss'
ERROR_CODE	Number		Paredzēts kļūdas koda saglabāšanai
ERROR_MESSAGE	Varchar2(512)		Paredzēts kļūdas teksta saglabāšanai.

2.3.1.12. Tabula "CLA_SUBCASE"

Tabula paredzēta, lai uzglabātu datus par cietušajiem transportlīdzekļiem. Dotās dokumentācijas kontekstā attiecas tikai uz OCTA lietām.

Jau esoša APDROŠINĀŠANAS KOMPĀNIJAS tabula, kuru šī darba autors nav veidojis un/vai modificējis.

2.12. tabula

Kolonna	Datu tips	Cita informācija	Komentārs
CLA_CASE_NO	Number(10)	NOT NULL	Norāda lietas numuru
CLA_SUBCASE_NO	Number(10)	NOT NULL	Norāda apakšlietas numuru
HANDLER	Varchar2(8)	NOT NULL	Norāda uz TIA lietotāju, kurš ir atbildīgs pa doto lietu
C26	Varchar2(2000)		Apraksta lauks, dotā dokumenta kontekstā glabā transportlīdzekļa tipu

2.3.1.13. Tabula "IT56CLA_CASE"

Jau esoša APDROŠINĀŠANAS KOMPĀNIJAS tabula, kuru šī darba autors nav veidojis un/vai modificējis.

2.12. tabula

Kolonna	Datu tips	Cita informācija	Komentārs
CLA_CASE_NO	Number(10)	NOT NULL	Norāda lietas numuru.
CLA_SUBCASE_NO	Number(10)	NOT NULL	Norāda apakšlietas numuru.
HANDLER	Varchar2(8)		Norāda uz TIA lietotāju, kurš ir atbildīgs pa doto lietu.
N01	Number		Apraksta lauks, dotās dokumentācijas kontekstā norāda uz transportlīdzekļa svaru. Attiecas uz OCTA lietām.

2.3.1.14. Tabula "TB5_SERVICE_HISTORY"

Tabula paredzēta, lai uzglabātu datus par notikušajiem datu importiem. Dotā tabula saglabā dažādus datus, kurus interpretējot tie atspoguļojas tabulās SSU_SERVICE_SUPPLIER, SSU_SER_SUP_SPECIALITY, SSU_SER_SUP_WORK_AREA, CLA_CUBE, līdz ar to TB5_SERVICE_HISTORY tabulai ir 476 kolonnas.

Daudzo kolonnu skaitu dēļ, netiks uzskaitītas visas tabulas kolonnas. Papildus aprakstītajām kolonnām ir kolonnas kuru nosaukumi un datu struktūra dublējas ar kolonnām no SSU_SERVICE_SUPPLIER. Pārējās kolonnas apzīmē servisu piedāvātos pakalpojumus vai šo servisu vecumu ierobežojumus. Šo kolonnu nosaukumi ir ņemti no XLA_REFERENCES tabulas ierakstiem, kuri ir saistīti ar pakalpojumu sniedzējiem un to datu tipi ir Varchar2(1). Katrai kolonnai, kura ir saistīta ar auto markām ir vēl 2 kolonnas, kura atšķiras no sākotnēja nosaukuma ar _K vai _O (piemērs, kolonna AA5, kura apzīmē AUDI ir papildus kolonnas AA5_K, AA5_O). Šo kolonnu tipi ir Varchar2(12), kuri ir paredzēti vecuma kodu ierobežojumiem.

Autora veidota tabula.

2.14. tabula

Kolonna	Datu tips	Cita informācija	Komentārs
SEQ_NO	Number	NOT NULL	Norāda ieraksta numuru pēc kārtas.
IMPORT_ID	Number	NOT NULL	Norāda importa numuru uz kuru dotais ieraksts attiecas.
NAME_ID_NO	Number(10)	NOT NULL	Norāda uz kuru pakalpojumu sniedzēju dotais ieraksts attiecas.
TIMESTAMP	Date	NOT NULL	Norāda, kad dotais ieraksts tika ievietots datu bāzē. Tiek glabāts formātā 'DD.MM.YYYY'
USERID	Varchar2(8)	NOT NULL	Norāda uz TIA lietotāju, kurš veicis dotā ieraksta importu.

2.3.1.15. Tabula "TB5_SERVICE_TEMP"

Tabula paredzēta datu pagaidu uzglabāšanai un validācijai pirms dati tiek saglabāti TB5_SERVICE_HISTORY tabulā, kā arī datu importam tabulās SSU_SERVICE_SUPPLIER, SSU_SER_SUP_SPECIALITY, SSU_SER_SUP_WORK_AREA, CLA_CUBE. Dotajai tabulai ir tādas pašas kolonas kā TB5_SERVICE_HISTORY, vienīgi šajā tabulā nav kolonna SEQ_NO un IMPORT_ID.

Autora veidota tabula.

3. TESTĒŠANAS DOKUMENTĀCIJA

3.1. Ievads

3.1.1. Nolūks

Šajā nodaļā tiek aprakstīta programmatūras testēšanas gaita un testa rezultāti.

3.1.2. Saistības ar citiem dokumentiem

Testēšanas dokumentācija un testēšana veikta balstoties uz PPS funkcionālajam prasībām un to akceptēšanas kritērijiem.

3.2. Testēšanas apraksts

Testēšana notika pēc t.s. "pelēkās kastes" principa. Testi tika veikti gan izstrādes laikā, gan pēc izstrādes. Visi testi tika veikti manuāli. Testēšanas žurnālā atrodamie testi tika veikti pēc pēdējās iterācijas pabeigšanas. Testa veiksmīga iznākuma gadījuma tiek ierakstīts "+", neveiksmes gadījumā "-".

Testēšanu veica darba autors, testētājs no apdrošināšanas uzņēmuma X un apdrošināšanas uzņēmuma X darbinieks, kurš akceptēja izmaiņu likšanu produkcijā. Turpmāk dokumentācijā tiks uzskaitīti testi, kurus veica darba autors.

3.3. Testēšanas žurnāls

3.3.1. Vienībtesti

Testa numurs	Skaidrojums	Iznākums
1.	Veic datu eksportu Datu Atjaunošanas formā. Modificē CSV failu, tā lai tas neatbilstu nosacījumiem. Veic datu importu, tiek attēlots kļūdas paziņojums, imports tiek pārtraukts.	+
2.	Veic datu importu Datu Atjaunošanas formā, izmantojot CSV failu, kuram ir kolonna, kura apzīmē pakalpojuma veidu, kurš vairs netiek atbalstīts. Veicot datu importu, tiek attēlots kļūdas paziņojums, imports tiek apturēts.	+
3.	Atver KASKO lietu un dodas uz Pakalpojumu sniedzēju formu. Izvēloties sadaļu Autoservisi tiek veikta automātiska datu atlase pēc kuriem, tiek	+

	izvēlēti atbilstošie servisi.	
3.	<p>Atver CASCO lietu un dodas uz Pakalpojumu sniedzēju formu.</p> <p>Izvēloties sadaļu Autoservisi tiek veikta automātiska datu atlase pēc kuriem, tiek izvēlēti atbilstošie servisi. Aizveru doto formu, izvēlos citu cietušo transportlīdzekli tai pašai lietai.</p> <p>Atkārtoti atveru Pakalpojumu sniedzēju formu, tiek atlasīti servisi, kuri atbilst tikko izvēlētajai transportlīdzekļa datiem.</p>	+
4.	<p>Atveru Pakalpojuma sniedzēju formu.</p> <p>Tiek atlasīti servisi, kuri atbilst izvēlētajai transportlīdzekļa datiem, veicu manuālu parametru maiņu, lai tiktu izvēlēti citi servisi. Atlase strādā ar jaunajiem datiem.</p>	+

3.3.2. Akcepttesti

Prasība	Darbības	Sagaidāmais rezultāts	Iznākums
1.3.1.1. Ekspertu pazīme	Atver pakalpojumu sniedzēju Datu Atjaunošanas formu.	Tiek atvērta dotā forma bez problēmām	+
	<p>Izvēlas eksporta sadaļu. Izvēlas <i>CSV file name - Eksperti</i>.</p> <p>Izvēlas vietu kur saglabāt CSV failu.</p> <p>Veic faila ģenerēšanu.</p>	Tiek uzģenerēts dotais fails ar ekspertu datiem.	+
	Atver un modificē doto failu, pievieno nepieciešamajiem	Dotais CSV fails ir ticis modificēts.	+

	ekspertiem auto pazīmi. Saglabā izmaiņas.		
	Dotajā formā izvēlas importa sadaļu. Izvēlas <i>CSV file name – Eksperti</i> . Izvēlas tikko modificēto CSV failu. Veic datu importu.	Tiek veikts veiksmīgs datu imports.	+
	Veic atkārtotu datu eksportu un salīdzina ar tikko importētajiem datiem.	Abos CSV failos sakrīt esošie dati pa ekspertu pazīmēm.	+
1.3.1.2. Vēsturisko autoservisu datu iegūšana	Atver pakalpojumu sniedzēju Datu Atjaunošanas formu.	Tiek atvērta dotā forma bez problēmām.	+
	Izvēlas iepriekš importēto failu sadaļu.	Tiek attēloti iepriekš veikti autoservisu un ekspertu importi.	+
	Izvēlas iepriekš veiktu autoservisu importu un veic tā eksportu.	Tiek uzģenerēts CSV fails ar izvēlētā importa datiem.	+
1.3.1.3. Vēsturisko ekspertu datu iegūšana	Atver pakalpojumu sniedzēju Datu Atjaunošanas formu.	Tiek atvērta dotā forma bez problēmām.	+
	Izvēlas iepriekš importēto failu sadaļu.	Tiek attēloti iepriekš veikti autoservisu un ekspertu importi.	+
	Izvēlas iepriekš veiktu ekspertu importu un veic tā eksportu.	Tiek uzģenerēts CSV fails ar izvēlētā importa datiem.	+
1.3.1.4.	Atver pakalpojumu	Dotā forma atveras	+

Iepriekšējās funkcionalitātes saglabāšana – atlase pēc manuāli ievadītiem datiem	sniedzēju atlases formu. Izvēlas <i>Saņēmējs – Autoserviss.</i>	bez problēmām. Izvēloties sadaļu Autoserviss tiek attēlots autoservisu saraksts.	
	Ievadot vērtības attiecīgajos laukos, tiek ņemti vērā autoservisu atlasē.	Mainās piedāvāto autoservisu saraksts, kurš atbilst ievadītajām vērtībām.	+
1.3.1.5. Iepriekšējās funkcionalitātes saglabāšana – automātiska datu atlase	Atver pakalpojumu sniedzēju atlases formu.	Dotā forma atveras bez problēmām.	+
	Izvēlas sadaļu <i>Saņēmējs – Autoserviss.</i>	Tiek automātiski nolasīti un ievadīti dati pa transporta līdzekli, izmantojot šos datus tiek veikta servisu atlase.	+
1.3.1.6. Eksperta apgabala izmantošana	Atver pakalpojumu sniedzēju atlases formu.	Dotā forma atveras bez problēmām.	+
	Izvēlas sadaļu <i>Saņēmējs – Eksperti.</i>	Tiek automātiski nolasīta informācija pa transporta līdzekļa darbības apgabalu, pēc kura tiek veikta veiksmīga ekspertu atlase.	
1.3.1.7.	Atver pakalpojumu	Dotā forma atveras	+

Vecās atlasēs mehānisms	sniedzēju atlasēs formu.	bez problēmām.	
	Izvēlas sadaļu <i>Saņēmējs – Autoserviss.</i>	Tiek attēloti autoservisi balstoties pēc automātiski atlasītajiem datiem.	+
	Uzspiež pogu <i>Izslēgt filtru.</i>	Tiek izslēgts jaunais auto atlasēs mehānisms. Atlase notiek ievadot vērtības attiecīgajos apraksta laukos.	+
1.3.1.8. Autoservisu marku vecuma ierobežojumi	Atver pakalpojumu sniedzēju atlasēs formu.	Dotā forma atveras bez problēmām.	+
	Izvēlas sadaļu <i>Saņēmējs – Autoserviss.</i>	Tiek attēloti autoservisi balstoties pēc automātiski atlasītajiem datiem. Attēlotie servisi atbilst vecuma ierobežojošajiem kritērijiem	+
1.3.1.9. Autoservisu datu eksports/imports	Atver pakalpojumu sniedzēju Datu Atjaunošanas formu.	Tiek atvērta dotā forma bez problēmām	+
	Izvēlas eksporta sadaļu. Izvēlas <i>CSV file name - Autoservisi.</i> Izvēlas vietu kur saglabāt CSV failu. Veic faila ģenerēšanu.	Tiek uzģenerēts dotais fails ar autoservisu datiem.	+
	Atver un modificē	Dotais CSV fails ir	+

	doto failu. Saglabā izmaiņas	ticis modificēts.	
	Dotajā formā izvēlas importa sadaļu. Izvēlas <i>CSV file name – Autoservisi</i> . Izvēlas tikko modificēto CSV failu. Veic datu importu.	Tiek veikts veiksmīgs datu imports.	+
	Veic atkārtotu datu eksportu un salīdzina ar tikko importētajiem datiem.	Abos CSV failos sakrīt visi dati.	+
1.3.1.10. Ekspertu datu eksports/imports	Atver pakalpojumu sniedzēju Datu Atjaunošanas formu.	Tiek atvērta dotā forma bez problēmām	+
	Izvēlas eksporta sadaļu. Izvēlas <i>CSV file name - Eksperti</i> . Izvēlas vietu kur saglabāt CSV failu. Veic faila ģenerēšanu.	Tiek uzģenerēts dotais fails ar ekspertu datiem.	+
	Atver un modificē doto failu. Saglabā izmaiņas	Dotais CSV fails ir ticis modificēts.	+
	Dotajā formā izvēlas importa sadaļu. Izvēlas <i>CSV file name – Eksperti</i> . Izvēlas tikko modificēto CSV failu. Veic datu importu.	Tiek veikts veiksmīgs datu imports.	+
	Veic atkārtotu datu eksportu un salīdzina ar tikko importētajiem	Abos CSV failos sakrīt visi dati.	+

	datiem.		
1.3.1.11. Vienots atlasē mehānisms	Veic autoservisu datu atlasē apdrošināšanas kompānijas lapā.	Tiek veikta servisu atlasē, kura atbilst konkrētā transportlīdzekļa datiem.	+
	Atver pakalpojumu sniedzēju atlasē formu, izmantojot tos pašus datus ar kuriem tika veikta servisu atlasē apdrošināšanas kompānijas lapā.	Dotā forma atveras bez problēmām.	+
	Izvēlas sadaļu <i>Saņēmējs – Autoserviss.</i>	Tiek automātiski atlasīti tie paši servisi, kā arī iespējams tiek attēloti vēl papildus servisi.	+
1.3.1.12. Transportlīdzekļu svars	Atver pakalpojumu sniedzēju atlasē formu.	Dotā forma atveras bez problēmām.	+
	Izvēlas sadaļu <i>Saņēmējs – Autoserviss.</i>	Tiek attēloti autoservisi balstoties pēc automātiski atlasītajiem datiem. Gadījumā ja automašīnas svars nesasniedz noteikto svaru, tad tiek mainīts transporta tips.	+

3.3.3. Nefunkcionālo prasību testēšana

Prasība	Darbības	Sagaidāmais rezultāts	Iznākums
1.4.1.1. Pārskatāmība	Tiek apskatītas citas TIA formas un salīdzinātas ar Datu Atjaunošanas formu formu	Esošā forma īpaši neatšķiras no citām TIA formām	+
1.4.1.1. Pārskatāmība	Tiek apskatītas citas TIA formas un salīdzinātas ar autoservisu atlases formu	Esošā forma īpaši neatšķiras no citām TIA formām	+

4. PROJEKTA PĀRVALDĪBA

4.1. Projekta organizācija

Sistēmas izstrāde tika izmantota spējā izstrāde (agile), jo apdrošināšanas uzņēmumā X tiek izmantots dotais izstrādes veids un darba pasūtītāji ir pieraduši un rēķinās ar to, ka izstrādes procesā var tikt sniegtas papildus prasības vai tikt mainītas iepriekšējās prasības.

Tika izveidots darba uzdevums JIRA 7.2., kurā tika saglabāti faili saistība ar pasūtītāju prasībām.

4.2. Izstrādes vide

Datubāze – Oracle Database 11g Enterprise Edition Release 11.2.0.4.0. -64bit Production.

Programmatūras izstrādes laika tika izmantotas 3 izstrādes vides:

- 1) Lokālā izstrādes vide – PL/SQL Developer, TIA, Oracle Form Builder 6.0.8.27
- 2) Testa vide – kopīga testa vide visiem testētājiem un izstrādātājiem
- 3) Akceptesta vide – testa vide ar jaunākajiem datiem no produkcijas

4.3. Kvalitātes nodrošināšana

Kvalitātes nodrošināšanai dotā programma tika izstrādāta vadoties pēc apdrošināšanas kompānijas X principiem. Programmatūras kods tika noformēts vienotā stilā, mainīgajiem tika doti saprotami apzīmējumi un tika pievienoti komentāri. Kods, kas tika rakstīts datu bāze tika formēts izmantojot PL/SQL Beautifier. Gan komentāri, gan pats kods tika rakstīts angļu valodā.

4.4. Konfigurācijas pārvaldība

Konfigurācijas pārvaldība tika nodrošināta ar Tortoise SVN palīdzību, ka arī Workbench – cits versijkontroles rīks.

Izstrādes sākotnējā stadijā tika lietots lokāls repozitorijs līdz programmatūra nonāca līdz stadijai, lai to varētu likt uz testa vides. Pēc tam sekojot apdrošināšanas kompānijas izstrādes nosacījumiem, katru reizi liekot jaunas izmaiņas uz testa vides izmaiņas tika saglabātas atsevišķā repozitorijā.

4.5. Komandas organizācija

Dotajā projektā bija tikai viens izstrādātājs – darba autors. Autors tieši komunicēja ar galveno arhitektu, IT komandas vadītāju, testētāju un citiem kolēģiem, lai informētu viņus par savām izmaiņām, prasītu padomus par doto projektu. Autors arī komunicēja ar darba pasūtītājiem un dotā produkta lietotāju, lai uzzinātu viņu prasības/vēlmes, komentārus/ieteikumus par izdarīto.

5. DARBIETILPĪBAS NOVĒRTĒŠANA

Darbietilpības novērtējumam tika izmantota COCOMO II darbietilpības novērtējuma kalkulators. Pieejams: <http://csse.usc.edu/tools/COCOMOII.php>

Aprēķinot darbietilpības novērtējumu, tika izmantots aptuvenais rindiņu skaits (saukts par SLOC) – 4500

5.1. tabula

Parametrs	Vērtība
Produkta sarežģītība	Ļoti Zema
Pieredze ar izstrādes platformu	Zema
Valodas pieredze	Zema
Pieredze ar lietojumprogrammu	Zema
Dokumentācijas saskaņotības ar dzīves cikla vajadzībām	Zema

Aprēķinātais darbietilpības rezultāts: 13,3 personmēneši

Dotais projekts tika veikts no 2016.gada jūnija vidus līdz 2016.gada oktobra beigām, kā arī tika veikti labojumi/izmaiņas pēc tam kad dotais produkts tika izlaists produkcijā. Līdz ar to reālais darbietilpības novērtējums ir 3,4 personmēneši.

Ļoti atšķiras aprēķinātais no reālā darbietilpības novērtējuma. To varētu skaidrot ar to, ka COCOMO II nav īpaši efektīvs šāda izmēra projektu darbietilpības noteikšanai. Netiek arī ņemtas vērā izstrādes valodas īpatnības, visbeidzot daudzi no parametriem ir subjektīvi vērtējami, kas var ļoti izmainīt doto novērtējumu.

6. SECINĀJUMI

Darba izstrādes laikā, tika izpildīti Ievadā izrunātie uzdevumi un sasniegti projekta mērķi. Tika izveidota Datu Atjaunošanas forma, par pamatu tika ņemta cita TIA forma (tikai vizuālais noformējums, pārējo funkcionalitāti veidoja darba autors), paralēli tam tika pārcelts servisu atlases algoritms no Servisu atlases formas uz datubāzi, dotajam algoritmam tika veikta veiktas vairākas izmaiņas (piemēram, meklējot autoserviss tiek ņemta vērā konkrētās markas vecuma ierobežojumi, meklējot ekspertus tiek ņemta vērā transportlīdzekļa apgabals, kur tas ticis reģistrēts u.c.), kā arī tika papildināta datubāze ar jaunām tabulām, kas bija nepieciešama, lai veiktu datu importu un modificēšanu saistībā ar servisiem un ekspertiem.

Darba autoram šī projekta laikā bija minimālas zināšanas un prasmes darbā ar Oracle formām un PL/SQL datubāzi. Veicot šo projektu darba autors ieguva jaunas zināšanas un prasmes saistībā ar šīm tehnoloģijām. Problēmas sagādāja automātisko testu neesamība, kas būtu atvieglējusi kļūdu atrašanu veicot izmaiņas servisu atlases mehānismā.

7. IZMANTOTĀ LITERATŪRA

1. **Database PL/SQL Language References** [tiešsaiste] – [atsauce 27.05.2017]

Pieejams: <https://docs.oracle.com/cloud/latest/db112/LNPLS/toc.htm>

2. **LVS 68:1996 Programmatūras prasību specifikācijas ceļvedis**, Latvijas Valsts standarts

3. **Einārs Koops**, Automatizēta autoservisu izveide : kvalifikācijas darbs. LU Datorikas fakultāte, Rīga Latvija 2016

4. **Robert C. Martin**. *Clean Code. A Handbook of Agile Software Craftsmanship* : PRENTICE HALL, 2008

5. **Terminu datubāze** [tiešsaiste] – [atsauce 27.05.2017] Pieejams: <http://termini.lza.lv/>

6. **Form Builder Reference, Volume 1** [tiešsaiste] – [atsauce 28.05.2017]

Pieejams: http://download.oracle.com/otn_hosted_doc/forms/forms/A73074_01.pdf

7. **LVS 68:1996 Programmatūras testēšanas dokumentācija sniegtās vadlīnijas**, Latvijas Valsts standarts

8. **LVS 72:1996 Ieteicamā prakse programmatūras projektējuma aprakstīšanai**, Latvijas Valsts standarts

9. Pielikums

1. pielikums

Autoservisu atlasas forma

The screenshot shows the 'LOV' application window with a search filter set to 'Autoserviss'. The search criteria include: Transporta gads: 2012, Transporta vecums: 4, Transporta tips: Viegļā, Transporta marka: OPEL, and Registrācijas datums: 17.12.2012. There are checkboxes for 'Apskate', 'Riepas', 'Stikls', 'Noma', 'Dileris', and 'Izslēgt filtru'. Below the search criteria is a table with columns: Darbības vieta, Nosaukums, Prior., Kontaktpersona, Apraksts, Iestāde, Veids, Rinda, and Zvaigznes. The table contains multiple rows of service records, with some cells redacted by black boxes.

Darbības vieta	Nosaukums	Prior.	Kontaktpersona	Apraksts	Iestāde	Veids	Rinda	Zvaigznes
Rīga		Zaļš				Viegļā;;Visas markas virs 3-50 pilni g.	5	
Rīga		Zaļš				HONDA;Viegļā;Dileris 0-3 pilni g.;Visa	1	
Rīga		Zaļš				Viegļā;;Visas markas 0-3 pilni g.;Visa	10	
Rīga		Zaļš				Apskate (vieglās a/m);Viegļā;;Visas m	0	
Rīga		Zaļš				Viegļā;;;Auto noma;Visas markas virs	3	
Rīga		WEB Dzelten				Apskate (vieglās a/m);Viegļā;;Visas m	7	
Rīga		WEB Dzelten				VW ;CITROEN ;SKODA ;Viegļā;;Dileris	7	
Rīga		WEB Dzelten				Apskate (vieglās a/m);Stikli (kravas a		
Rīga		WEB Dzelten				MAZDA ;FORD;Viegļā;;Dileris 0-3 pilni		
Rīga		WEB Dzelten				HONDA;Viegļā;Dileris 0-3 pilni g.;Visa	3	
Rīga		WEB Dzelten				Apskate (vieglās a/m);Apskate (krava	11	
Rīga		WEB Dzelten				Apskate (vieglās a/m);Visas markas v		
Rīga		WEB Dzelten				Apskate (vieglās a/m);Viegļā;Visas m	9	
Rīga		WEB Dzelten				Viegļā;;Visas markas virs 3-50 pilni g.	0	
Rīga		WEB Dzelten				Apskate (vieglās a/m);Visas markas v		
Rīga		WEB Dzelten				Apskate (vieglās a/m);Viegļā;Visas m	0	
Rīga		WEB Dzelten				Viegļā;;Visas markas virs 3-50 pilni g.	14	
Rīga		WEB Dzelten				TOYOTA ;LEXUS;BMW;Viegļā;Dileris 0-4		
Rīga		Nesūtīt Zaļš				Viegļā;Visas markas virs 3-50 pilni g.;		
Rīga		Nesūtīt Zaļš				T/A rezerves daļu sagāde;Visas marka		

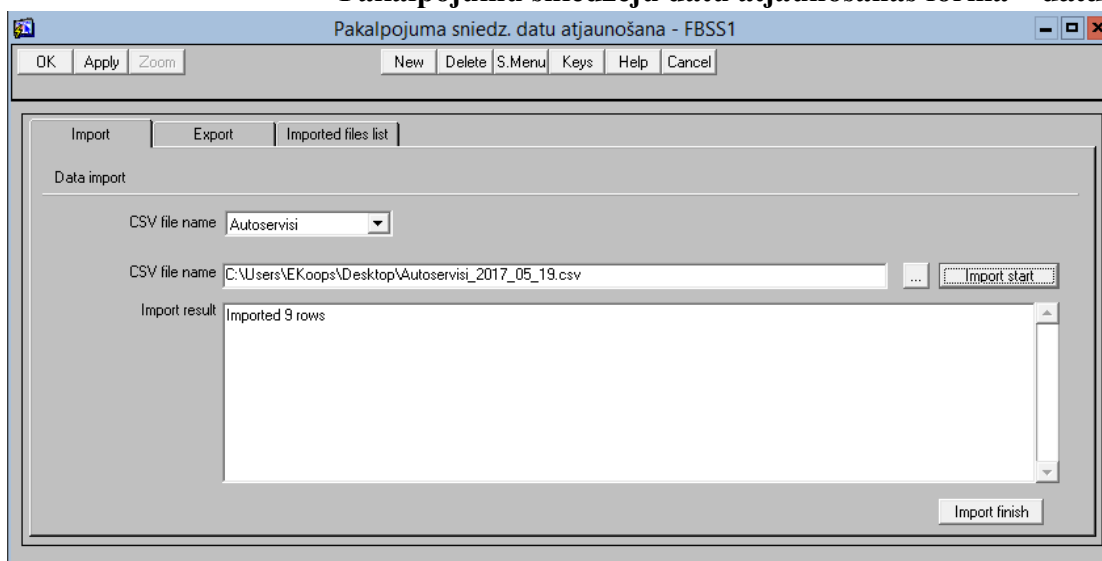
2. pielikums

Pakalpojumu sniedzēju datu atjaunošanas forma – datu eksports

The screenshot shows the 'Pakalpojuma sniedz. datu atjaunošana - FBSS1' application window. It has tabs for 'Import', 'Export', and 'Imported files list'. The 'Export' tab is active, showing a 'Data export' section. The 'CSV file name' is set to 'Autoservisi'. The 'CSV file name' field contains the path 'C:\Users\VEKoops\Desktop\Autoservisi_2017_05_19.csv'. There is an 'Export' button and a status bar that says 'Export finished successfully!'.

3. pielikums

Pakalpojumu sniedzēju datu atjaunošanas forma – datu imports



4. pielikums

CSV faila datu struktūra un dati

Autoservisi	Saisinātais	Nosaukums	Apraksts	Kontaktin	Darba laiks	Prioritāte	Prioritāte: Rinda	Zvaigznēs	Darbības ē	Istā adrese	Koordināt	Koordināt	Sākuma da	Beigu dat	Blacklist	Blacklist	Modificēša
19569					Darba laiks	2	WEB Dzelt	7	Cēsis				01.01.2005		Y		Dace.Fri 09.05.2017
20257					Darba laiks	1	Zaļš	2	Talsi				01.01.2005		Y		Dace.Fri 09.05.2017
20265					Darba laiks	2	WEB Dzeltens		Talsi				01.01.2005		Y		Dace.Fri 09.05.2017
20273						9	Nešūt WEB Sarkans		Daugavpils				01.01.2005		Y		Dace.Fri 09.05.2017
20672						5	Nesūt WEB Dzeltens		Madona				01.01.2005		Y		Dace.Fri 09.05.2017
20761					Darba laiks	1	Zaļš	3	Rīga				13.01.2005		Y		Dace.Fri 09.05.2017
30953					Darba laiks	2	WEB Dzelt	7	Līvāni				01.01.2002		Y		Dace.Fri 09.05.2017
30961					Darba laiks	1	Zaļš	21	Kuldīga				01.01.2005		Y		Dace.Fri 09.05.2017
40851					Darba laiks	2	WEB Dzeltens		Daugavpil				01.01.2005		Y		Dace.Fri 09.05.2017

5. pielikums

CSV faila datu struktūra un dati

13	14	15	16	17	18	19	20	22	23	24	25	26	27	BA1	BA2	BA3	BA4	BA5	BA9
Autostiki	Riepu rei	Ritenu di	T/I signa	Automag	Cits	Auto non	Apskate	Stikli (kri	Stikli (au	Riepas (I	Riepas (I	Apskate	Visu mar	Vieglā	Kravas	Autobusi	Motociki	Traktors	Piekabes
X													X						
X						X			X				X						
									X				X						
						X							X						
X						X	X		X	X	X		X	X	X			X	X
X						X	X		X	X	X		X	X	X			X	X
X									X				X						
													X						
													X						
													X						
													X						
X								X	X				X	X	X				


```

LOOP
  l_TABLE_COLUMN(i) := fb6_x_take2(l_temp, l_delimiter);
  l_TABLE_COLUMN(i) := '1';
END LOOP;

for i IN 20..222 --first 19 are ignored
LOOP
  BEGIN
    l_TABLE_COLUMN(i) := fb6_x_take2(l_temp, l_delimiter);
    l_TABLE_COLUMN(i) := rtrim(ltrim(l_TABLE_COLUMN(i)));
    SELECT SUBSTR(l_TABLE_COLUMN(i), 1, 1) INTO l_usable FROM DUAL;
    if l_usable='A' AND l_TABLE_COLUMN(i) <> 'A02' then ---if column is
for car model then skip next 2 columns
      l_TABLE_COLUMN(1) := fb6_x_take2(l_temp, l_delimiter);
      l_TABLE_COLUMN(1) := fb6_x_take2(l_temp, l_delimiter);
      l_TABLE_COLUMN(1) := '3';
    end if;
    if l_TABLE_COLUMN(i)='1' or l_TABLE_COLUMN(i)='2' or
l_TABLE_COLUMN(i)='3' or l_TABLE_COLUMN(i)='4'
      or l_TABLE_COLUMN(i)='5' or l_TABLE_COLUMN(i)='6' or
l_TABLE_COLUMN(i)='7' or l_TABLE_COLUMN(i)='8'
      or l_TABLE_COLUMN(i)='9' then
      SELECT ROWNUM AS RNM INTO L_USABLE FROM XLA_REFERENCE X
      WHERE X.TABLE_NAME='SSU_SPECIALITY_TYPE' AND
      1 = x_date_between(sysdate, x.start_date, x.end_date) AND
      X.CODE = '0' || l_TABLE_COLUMN(i) AND
      X.LANGUAGE='LV';
      l_TABLE_COLUMN(i) := '0' || l_TABLE_COLUMN(i);
    elsif l_TABLE_COLUMN(1)=3 then
      l_TABLE_COLUMN(1) := 2;
    else
      SELECT ROWNUM AS RNM INTO L_USABLE FROM XLA_REFERENCE X
      WHERE X.TABLE_NAME='SSU_SPECIALITY_TYPE' AND
      1 = x_date_between(sysdate, x.start_date, x.end_date) AND
      X.CODE = l_TABLE_COLUMN(i) AND
      X.LANGUAGE='LV';
    end if;
  EXCEPTION
    WHEN NO_DATA_FOUND THEN
      IF l_TABLE_COLUMN(i) IS NULL or l_TABLE_COLUMN(i)='' then
        exit;
      else
        :header.IMPORT_RESULT := 'Error with column check. Column ' ||
l_table_column(i);
        bb5_service_suppliers.service_log_error(:header.import_id, '',
'Error with column check. Column ' || l_table_column(i), '2');
        return;
      end if;
  END;
END LOOP;

:header.import_result := 'Column check done';
synchronize;
TEXT_IO.Get_Line(l_file, l_temp); --reads row where all column names
are
TEXT_IO.Get_Line(l_file, l_temp);
LOOP --reads all the data from excel into arrays and inserts it into
table

for column_count IN 1..19 -- column count
LOOP

  l_TABLE_ROW(column_count) := fb6_x_take2(l_temp, l_delimiter);

```

```

        l_TABLE_ROW(column_count) :=
rtrim(ltrim(l_TABLE_ROW(column_count)));
    END LOOP;
    L_TABLE_ROW(4) := REPLACE(L_TABLE_ROW(4), '&', ';'); --- replace
characters again, because needed to do it in data export
    L_TABLE_ROW(5) := REPLACE(L_TABLE_ROW(5), '&', ';');
    L_TABLE_ROW(18) := REPLACE(L_TABLE_ROW(18), '&', ';');
    L_TABLE_ROW(18) := REPLACE(L_TABLE_ROW(18), '&', ';');

    if l_TABLE_ROW(1) IS NULL OR l_TABLE_ROW(2) IS NULL then ---Check for
mandatory values
        :header.IMPORT_RESULT := 'Not all mandatory values are set';
        bb5_service_suppliers.service_log_error(:header.import_id,'',
        'Not all mandatory values are set. Values' || l_TABLE_ROW(1) || ', '
|| l_TABLE_ROW(2) || ', ' || l_TABLE_ROW(9), '2');
        return;
    end if;

    for i IN 20..222
    LOOP
        l_TABLE_ROW(i) := fb6_x_take2(l_temp, l_delimiter);
        l_TABLE_ROW(i) := rtrim(ltrim(L_TABLE_ROW(i)));
        if l_TABLE_ROW(i)='x' then
            l_TABLE_ROW(i):='X';
        elsif l_TABLE_ROW(i)<>'X' AND l_TABLE_ROW(i) IS NOT NULL then
            :header.import_result := 'Incorrect data found! Import
canceled. Problems with speciality! Wrong value:= ' || l_TABLE_ROW(i);
            bb5_service_suppliers.service_log_error(:header.import_id,'',
'Incorrect data found! Import canceled. Problems with speciality! Wrong
value:= ' || l_TABLE_ROW(i), '2');
            return;
        end if;
        SELECT SUBSTR(l_TABLE_COLUMN(i), 1, 1) INTO l_usable FROM DUAL;
        if l_usable='A' AND l_TABLE_COLUMN(i)<>'A02' then ---processes array
when taking information about car models
            l_number_2 := i+1;
            l_number := 1;
            OPEN MODEL_AGE_CHECK;
            l_MODEL_AGE(l_number) := fb6_x_take2(l_temp, l_delimiter);
            l_MODEL_AGE(l_number) :=
upper(rtrim(ltrim(l_MODEL_AGE(l_number))));
            l_usable := NULL;
            j:=1;
            While j<9
            LOOP
                SELECT SUBSTR(l_MODEL_AGE(l_number), j, 3) INTO l_code FROM
DUAL;

                FETCH MODEL_AGE_CHECK INTO l_usable;
                if l_usable is NULL AND l_MODEL_AGE(l_number)<>' then
                    CLOSE MODEL_AGE_CHECK;
                    :header.import_result := 'Incorrect data found! Import
canceled. Problems with car model ages! Wrong value:= ' ||
l_MODEL_AGE(l_number);
                    bb5_service_suppliers.service_log_error(:header.import_id,'',
'Incorrect data found! Import canceled. Problems with car model ages! Wrong
value:= ' || l_MODEL_AGE(l_number), '2');
                    return;
                end if;
                j := j+3;
            END LOOP;

            l_number := l_number + 1;
            l_MODEL_AGE(l_number) := fb6_x_take2(l_temp, l_delimiter);

```

```

        l_MODEL_AGE(l_number) :=
upper(rtrim(ltrim(l_MODEL_AGE(l_number))));
        l_number := l_number + 1;
        l_usable := NULL;
        j:=1;
        While j<9
        LOOP
            SELECT SUBSTR(l_MODEL_AGE(l_number), j, 3) INTO l_code FROM
DUAL;

            FETCH MODEL_AGE_CHECK INTO l_usable;
            if l_usable is NULL AND l_MODEL_AGE(l_number)<>' ' then
                CLOSE MODEL_AGE_CHECK;
                :header.import_result := 'Incorrect data found! Import
canceled. Problems with car model ages! Wrong value:= ' ||
l_MODEL_AGE(l_number);
                bb5_service_suppliers.service_log_error(:header.import_id,'',
'Incorrect data found! Import canceled. Problems with car model ages! Wrong
value:= ' || l_MODEL_AGE(l_number), '2');
                return;
            end if;
            j := j+3;
        END LOOP;

for m IN l_number_2..222
LOOP
    if l_TABLE_COLUMN(m) IS NULL then
        exit;
    end if;
    l_TABLE_ROW(m) := fb6_x_take2(l_temp, l_delimiter);
    l_TABLE_ROW(m) := rtrim(ltrim(L_TABLE_ROW(m)));
    if l_TABLE_ROW(m)='x' then
        l_TABLE_ROW(m) := 'X';
    elsif l_TABLE_ROW(m)<>'X' AND l_TABLE_ROW(m) IS NOT NULL then
        CLOSE MODEL_AGE_CHECK;
        :header.import_result := 'Incorrect data found! Import
canceled. Problems with car models! Wrong value := ' || l_TABLE_ROW(m);
        bb5_service_suppliers.service_log_error(:header.import_id,'',
'Incorrect data found! Import canceled. Problems with car models! Wrong
value := ' || l_TABLE_ROW(m), '2');
        return;
    end if;
    l_MODEL_AGE(l_number) := fb6_x_take2(l_temp, l_delimiter);
    l_MODEL_AGE(l_number) := rtrim(ltrim(l_MODEL_AGE(l_number)));
    l_usable := NULL;
    j:=1;
    While j<9
    LOOP
        SELECT SUBSTR(l_MODEL_AGE(l_number), j, 3) INTO l_code FROM
DUAL;

        FETCH MODEL_AGE_CHECK INTO l_usable;
        if l_usable is NULL AND l_MODEL_AGE(l_number)<>' ' then
            CLOSE MODEL_AGE_CHECK;
            :header.import_result := 'Incorrect data found! Import
canceled. Problems with car model ages! Wrong value:= ' ||
l_MODEL_AGE(l_number);

bb5_service_suppliers.service_log_error(:header.import_id,'', 'Incorrect
data found! Import canceled. Problems with car model ages! Wrong value:= '
|| l_MODEL_AGE(l_number), '2');
            return;
        end if;
        j := j+3;
    END LOOP;

```

```

        l_number := l_number + 1;
        l_MODEL_AGE(l_number) := fb6_x_take2(l_temp, l_delimiter);
        l_MODEL_AGE(l_number) := rtrim(ltrim(l_MODEL_AGE(l_number)));
        l_number := l_number + 1;
        l_usable := NULL;
        j:=1;
        While j<9
        LOOP
            SELECT SUBSTR(l_MODEL_AGE(l_number), j, 3) INTO l_code FROM
DUAL;

            FETCH MODEL_AGE_CHECK INTO l_usable;
            if l_usable is NULL AND l_MODEL_AGE(l_number)<>' ' then
                CLOSE MODEL_AGE_CHECK;
                :header.import_result := 'Incorrect data found! Import
canceled. Problems with car model ages! Wrong value:= ' ||
l_MODEL_AGE(l_number);

bb5_service_suppliers.service_log_error(:header.import_id,'', 'Incorrect
data found! Import canceled. Problems with car model ages! Wrong value:= '
|| l_MODEL_AGE(l_number), '2');
                return;
            end if;
            j := j+3;
        END LOOP;
    END LOOP;
    CLOSE MODEL_AGE_CHECK;
    exit;
end if;
END LOOP;

```

9. pielikums

Pirmkoda piemērs no datubāzes

```

/**
    Function is used at form FBSS1. Transfers data from TB5_SERVICE_HISTORY
to SSU_SERVICE_SUPPLIER, SSU_SER_SUP_SPECIALITY, SSU_SER_SUP_WORK_AREA
    p_import_id - shows UNIQUE number which will be used for this data
import

    Returns 1 if data transfer done without any problems and 0 if some kind
of error occurred.
**/
function transfer_to_prod_service(p_import_id number) return number is
    l_prew_id_num        number := 0;
    l_check_row_big      number;
    l_service_num        number := 0;
    l_spec_type          varchar2(20);
    l_value_spec         varchar2(20);
    l_pl_sql_block       varchar2(400);
    l_spec_type_number   varchar2(10);
    l_age_1              varchar2(12);
    l_age_2              varchar2(12);
    l_work_area          varchar2(10);
    l_count              number := 0;

begin

    update_log(p_import_id, 7);
    DELETE FROM CLA_CUBE C WHERE C.TABLE_NAME = 'SSU_SERVICE_SUPPLIER';

    FOR R IN (SELECT *
              FROM TB5_SERVICE_HISTORY

```

```

        WHERE IMPORT_ID = p_import_id
        ORDER BY NAME_ID_NO) LOOP
l_service_num := R.name_id_no;
if l_prew_id_num <> l_service_num then
    SELECT COUNT(*)
    INTO l_check_row_big
    FROM TB5_SERVICE_HISTORY T
    WHERE t.name_id_no = l_service_num
        and t.import_id = p_import_id;
end if;
If l_check_row_big >= 1 then
    --updates existing service
    l_check_row_big := -1;
    l_work_area := NULL;
    UPDATE SSU_SERVICE_SUPPLIER S ---main information about service
        SET S.ALIAS = R.alias,
            S.DESCRPTION = R.DESCRPTION,
            S.CONTACT_PERSON = R.CONTACT_PERSON,
            S.WORK_TIME = R.WORK_TIME,
            S.START_DATE = R.START_DATE,
            S.END_DATE = R.END_DATE,
            S.BLACKLISTED = R.BLACKLISTED,
            S.BLACKLISTED_REASON = R.BLACKLISTED_REASON,
            S.PRIORITY = R.PRIORITY,
            S.STARS = R.STARS,
            S.REAL_ADDRESS = R.REAL_ADDRESS,
            S.COORDINATE_LAT = R.COORDINATE_LAT,
            S.COORDINATE_LAN = R.COORDINATE_LAN,
            S.QUEUE = R.QUEUE
    WHERE S.NAME_ID_NO = l_service_num;

    DELETE FROM SSU_SER_SUP_WORK_AREA a
    WHERE a.NAME_ID_NO = l_service_num;
    l_work_area := R.work_area;

    if r.work_area IS NOT NULL then
        INSERT INTO SSU_SER_SUP_WORK_AREA W
            (W.NAME_ID_NO, W.WORK_AREA)
        VALUES
            (l_service_num, R.work_area);
    end if;

    for i in 1 .. 37 loop
        ---first service specialities
        if i <= 9 then
            l_spec_type := 'Z0' || i;
            l_spec_type_number := '0' || i;
        else
            l_spec_type := 'Z' || i;
            l_spec_type_number := i;
        end if;

        l_pl_sql_block := 'SELECT ' || l_spec_type ||
            ' FROM TB5_SERVICE_HISTORY WHERE IMPORT_ID='
            ||
            p_import_id || ' AND NAME ID NO=' ||
            l_service_num || ' AND ROWNUM=1';
        EXECUTE IMMEDIATE l_pl_sql_block
            INTO l_value_spec;
        if l_value_spec = 'X' AND l_value_spec IS NOT NULL then
            SELECT COUNT(*)
            INTO l_count
            FROM SSU_SER_SUP_SPECIALITY

```

```

WHERE NAME_ID_NO = l_service_num
AND SPECIALITY_TYPE = l_spec_type_number;
if l_count >= 1 then
UPDATE SSU_SER_SUP_SPECIALITY
SET RECORD_VERSION = RECORD_VERSION + 1
WHERE NAME_ID_NO = l_service_num
and SPECIALITY_TYPE = l_spec_type_number;
else
INSERT INTO SSU_SER_SUP_SPECIALITY
(NAME_ID_NO, SPECIALITY_TYPE)
VALUES
(l_service_num, l_spec_type_number);
end if;
else
DELETE FROM SSU_SER_SUP_SPECIALITY
WHERE NAME_ID_NO = l_service_num
AND SPECIALITY_TYPE = l_spec_type_number;
end if;
end loop;

for g in (SELECT UNIQUE CODE
FROM XLA_REFERENCE X
WHERE X.TABLE_NAME = 'SSU_SPECIALITY_TYPE'
AND 1 =
x_date_between(sysdate, x.start_date, x.end_date)
AND (CODE LIKE 'B%' OR CODE LIKE 'C%' OR CODE =
'A02')) LOOP
---service specialities without bonus age fields
l_pl_sql_block := 'SELECT ' || g.code ||
' FROM TB5_SERVICE_HISTORY WHERE IMPORT_ID='
||
p_import_id || ' AND NAME_ID_NO=' ||
l_service_num || ' AND ROWNUM=1';
EXECUTE IMMEDIATE l_pl_sql_block
INTO l_value_spec;
if l_value_spec = 'X' AND l_value_spec IS NOT NULL then
SELECT COUNT(*)
INTO l_count
FROM SSU_SER_SUP_SPECIALITY
WHERE NAME_ID_NO = l_service_num
AND SPECIALITY_TYPE = g.code;
if l_count >= 1 then
UPDATE SSU_SER_SUP_SPECIALITY
SET RECORD_VERSION = RECORD_VERSION + 1
WHERE NAME_ID_NO = l_service_num
and SPECIALITY_TYPE = g.code;
else
INSERT INTO SSU_SER_SUP_SPECIALITY
(NAME_ID_NO, SPECIALITY_TYPE)
VALUES
(l_service_num, g.code);
end if;
else
DELETE FROM SSU_SER_SUP_SPECIALITY
WHERE NAME_ID_NO = l_service_num
AND SPECIALITY_TYPE = g.code;
end if;
END LOOP;

for k in (SELECT UNIQUE CODE
FROM XLA_REFERENCE X
WHERE X.TABLE_NAME = 'SSU_SPECIALITY_TYPE'
AND 1 =

```

```

        x_date_between(sysdate, x.start_date, x.end_date)
        AND CODE LIKE 'A%'
        AND CODE <> 'A02') LOOP
l_pl_sql_block := 'SELECT ' || k.code ||
        ' FROM TB5_SERVICE_HISTORY WHERE IMPORT_ID='
||
        p_import_id || ' AND NAME_ID_NO=' ||
        l_service_num || ' AND ROWNUM=1';
EXECUTE IMMEDIATE l_pl_sql_block
INTO l_value_spec;
if l_value_spec = 'X' AND l_value_spec IS NOT NULL then
SELECT COUNT(*)
INTO l_count
FROM SSU_SER_SUP_SPECIALITY
WHERE NAME_ID_NO = l_service_num
AND SPECIALITY_TYPE = k.code;
if l_count >= 1 then
UPDATE SSU_SER_SUP_SPECIALITY
SET RECORD_VERSION = RECORD_VERSION + 1
WHERE NAME_ID_NO = l_service_num
and SPECIALITY_TYPE = k.code;
else
INSERT INTO SSU_SER_SUP_SPECIALITY
(NAME_ID_NO, SPECIALITY_TYPE)
VALUES
(l_service_num, k.code);
end if;
else
DELETE FROM SSU_SER_SUP_SPECIALITY
WHERE NAME_ID_NO = l_service_num
AND SPECIALITY_TYPE = k.code;
end if;

l_pl_sql_block := 'SELECT ' || k.code ||
        ' FROM TB5_SERVICE_HISTORY WHERE IMPORT_ID='
||
        p_import_id || ' AND NAME_ID_NO=' ||
        l_service_num || ' AND ROWNUM=1';
EXECUTE IMMEDIATE l_pl_sql_block
INTO l_value_spec;
l_pl_sql_block := 'SELECT ' || k.code ||
        '_K FROM TB5_SERVICE_HISTORY WHERE IMPORT_ID='
||
        p_import_id || ' AND NAME_ID_NO=' ||
        l_service_num || ' AND ROWNUM=1';
EXECUTE IMMEDIATE l_pl_sql_block
INTO l_age_1;
l_pl_sql_block := 'SELECT ' || k.code ||
        '_O FROM TB5_SERVICE_HISTORY WHERE IMPORT_ID='
||
        p_import_id || ' AND NAME_ID_NO=' ||
        l_service_num || ' AND ROWNUM=1';
EXECUTE IMMEDIATE l_pl_sql_block
INTO l_age_2;
if (l_age_1 is not null or l_age_2 is not null) THEN
--- if there isnt a single record about this services car
model, then no information is transfered to other tables (ages)
INSERT INTO CLA_CUBE C
(C.TABLE_NAME, C.DIM01, C.DIM02)
VALUES
('SSU_SERVICE_SUPPLIER', l_service_num, k.code);
UPDATE CLA_CUBE
SET DIM03 = UPPER(REPLACE(l_age_1, ';', '')),

```

```

        DIM04 = UPPER(REPLACE(l_age_2, ';', ''))
    WHERE TABLE_NAME = 'SSU_SERVICE_SUPPLIER'
        AND DIM01 = l_service_num
        AND DIM02 = k.code;
    END IF;
END LOOP;

elsif l_check_row_big = -1 then
    --in case when multiple rows have same id (happens when service is
    "located" in multiple work_areas)
    if l_work_area <> r.work_area then
        if r.work_area is not null then
            DELETE SSU_SER_SUP_WORK_AREA
            WHERE NAME_ID_NO = l_service_num
            AND WORK_AREA = R.WORK_AREA;
            INSERT INTO SSU_SER_SUP_WORK_AREA W
            (W.NAME_ID_NO, W.WORK_AREA)
            VALUES
            (l_service_num, r.work_area);
        end if;
    end if;
end if;
l_prew_id_num := l_service_num;
END LOOP;
DELETE FROM TB5_SERVICE_TEMP;
commit;
return 1;
EXCEPTION
    WHEN OTHERS THEN
        service_log_error(p_import_id,
            '',
            'Error while importing to PROD - ' || sqlerrm,
            '8');

        rollback;
        DELETE FROM TB5_SERVICE_TEMP;
        COMMIT;
        return 0;
end;

```

Kvalifikācijas darbs „Autoservisu datu imports un apstrāde” izstrādāts Latvijas Universitātes Datorikas fakultātē.

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

Autors: Einārs Koops _____ .05.2017.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: Docents Imants Gorbāns _____ .05.2017.

Recenzents: M.Dat, Maija Ļaksa

Darbs iesniegts 29.05.2017.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: Darja Solodovņikova _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2017. prot. Nr. _____

Komisijas sekretārs(-e): _____