



RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes un informācijas tehnoloģijas fakultāte

Informācijas tehnoloģijas institūts

1. praktiskais darbs

“CASE rīki datu bāzu projektēšanā”

Rīks „Toad Data Modeler”

Autori: L. Dombure

K. Vītols

Grupa: RDGD0

Pārbaudīja: prof. J. Eiduks

RĪGA, 2012

SATURS

Ievads.....	3
1. Teorētiskā daļa.....	4
1.1. Rīka tehniskās prasības	4
1.2. Rīka iegādes cena	4
1.3. Rīka piedāvātās iespējas.....	4
1.4. Rīka instrumentārijs	6
1.5. Notācijas.....	8
1.6. Modeļi	10
1.7. Dokumentācija	10
1.8. Papildus iespējas	11
1.8.1. Version Manager.....	11
1.8.2. To-Do list.....	11
1.8.3. XML.....	12
1.8.4. Rules	13
2. Praktiskā daļa.....	15
2.1. Jauna projekta izveide	15
2.2. Loģiskā modeļa izveide.....	16
2.3. Mantošanas realizācija	20
2.4. Loģiskā modeļa verificēšana.....	22
2.5. Kopējais loģiskais modelis.....	24
2.6. Loģiskā modeļa transformācija fiziskajā modelī	24
2.7. Modeļu salīdzināšana	25
2.8. PER -> LER	25
2.9. Loģisko modeļu salīdzinājums.....	26
2.10. Ģenerētais SQL skripts.....	26
Subjektīvs novērtējums	30
Izmantotā literatūra.....	31

IEVADS

Praktiskajā darbā „Toad Data Modeler” izstrādāts datu bāzes „Studentu reģistrācija” piemērs ar datu bāzi un lietojumprogrammu projektējuma rīka „Toad Data Modeler” 4.3.3.6. versiju.

Darbs sastāv no 2 nodaļām, kur darba pirmajā nodaļā tiek apskatīta „Toad Data Modeler” saskarne, piedāvātā funkcionalitāte, bet otrajā nodaļā tiek atspoguļota mācību datu bāzes modeļu izveide.

Darba īstenošanā iesaistīti 2 dalībnieki.

Kopējais darba apjoms 31 lapa.

1. TEORĒTISKĀ DAĻA

1.1. Rīka tehniskās prasības

Rīka „Toad Data Modeler” tehniskās minimālās un optimālās prasības definētas 1. tabulā.

	Minimālās prasības	Optimālās prasības
Platforma	Pentium	Pentium dual core
Atmiņa	512 MB	4 GB
Cietais disks	200 MB	400 MG
Operētājsistēmas	<ul style="list-style-type: none">• MS Windows XP;• MS Windows Vista;• MS Windows Server 2003;• MS Windows Server 2008;• MS Windows Server 2008 R2;• MS Windows 7.	

1.2. Rīka iegādes cena

Mājas lapā <https://shop.quest.com> publiskotajā informācijā par Toad Data Modeler vienas licences iegādes cenu norādītā vērtība ir 450 €, salīdzinājumam darba autori atrada 2009. gada studenšu izstrādātā praktiskā darba „Toad Data Modeler” minēto cenu 479 \$.

Darba autori darba izstrādei izmanto *trial* versiju, ar visiem no versijas izrietošajiem ierobežojumiem.

1.3. Rīka piedāvātās iespējas

Rīks „Toad Data Modeler” atbalsta sekojošu datu bāzu sistēmu izmantošanu, nodrošinot tabulā atzīmēto funkcionalitāti („+” - nodrošina funkcionalitāti konkrētajai db sistēmai, „-” - nenodrošina minēto funkcionalitāti):

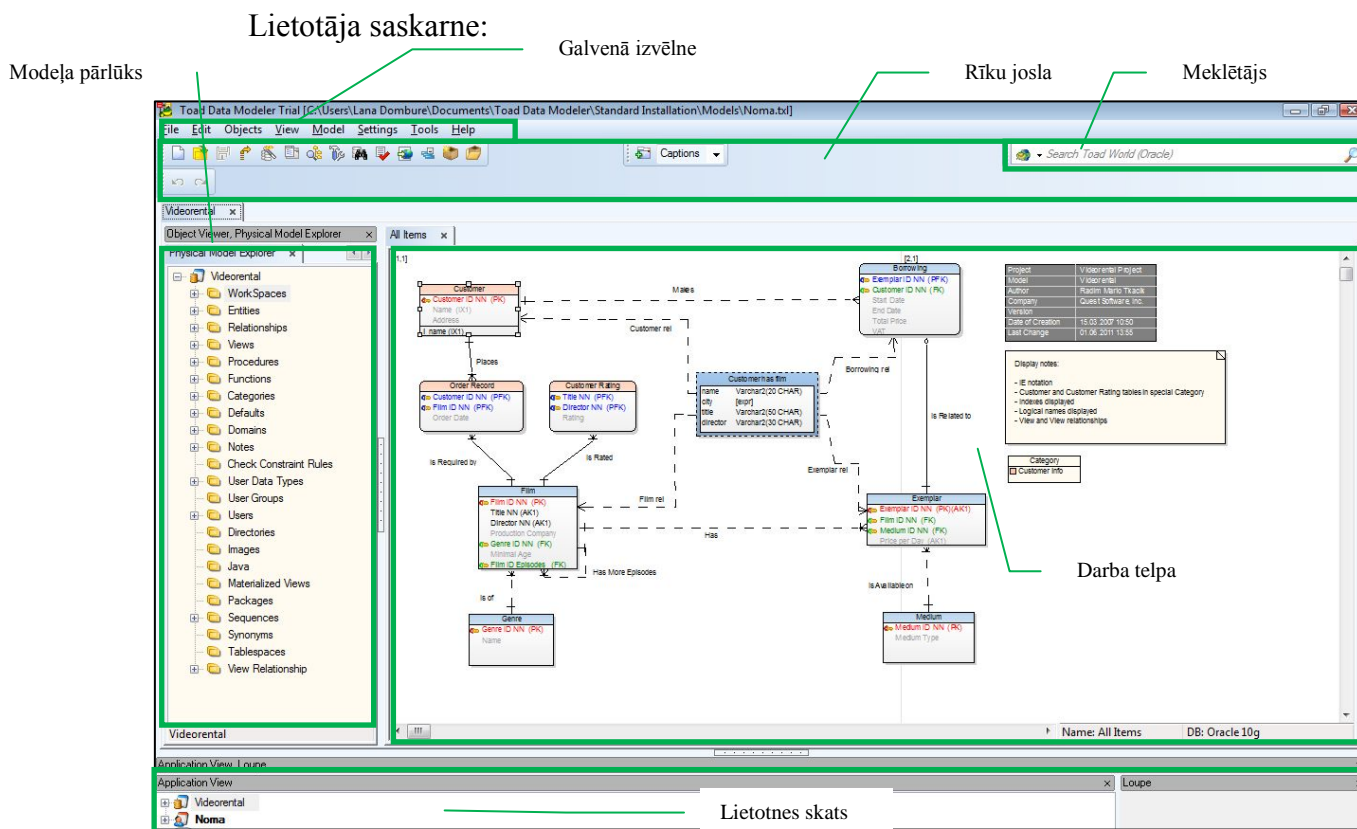
Atbalstītās DB sistēmas	Reverse Engineering	Import From SQL File	Alter Script Generation	SQL/DDDL Code Generation
DB2 z/OS v. 10	+	-	-	+
DB2 z/OS v. 9	+	-	-	+
DB2 v. 9.7 (LUW)	+	+	+	+
DB2 v. 9.5 (LUW)	+	+	+	+
DB2 v. 9 (LUW)	+	+	+	+
DB2 UDB v. 8 (LUW)	+	+	+	+
Ingres 10.0	+	-	-	+
Ingres 9.3	+	-	-	+
Microsoft Access 2007/2010	+	-	-	+
Microsoft Access 2000-2003	+	-	-	+
Microsoft SQL Azure	+	+	+	+
Microsoft SQL Server 2012	+	+	+	+
Microsoft SQL Server 2008	+	+	+	+

Microsoft SQL Server 2005	+	+	+	+
Microsoft SQL Server 2000	+	-	+	+
MySQL 5.5	+	+	+	+
MySQL 5.1	+	+	+	+
MySQL 5.0	+	+	+	+
Oracle 11g R2	+	+	+	+
Oracle 11g R1	+	+	+	+
Oracle 10g	+	+	+	+
Oracle 9i	+	+	+	+
PostgreSQL 9.1	+	-	+	+
PostgreSQL 9.0	+	-	+	+
PostgreSQL 8.4	+	-	+	+
PostgreSQL 8.3	+	-	+	+
PostgreSQL 8.2	+	-	+	+
PostgreSQL 8.1	+	-	+	+
SQLite 3.7	+	-	-	+
Sybase ASE 15.7	+	-	-	+
Sybase ASE 15.5	+	-	-	+
Sybase ASE 15	+	-	-	+
Sybase ASE 12.5	+	-	-	+
Sybase IQ 15.2	+	-	-	+
Sybase SQL Anywhere 11	+	-	-	+
Teradata 13	+	-	-	+
Other databases (Universal Model)	+	-	-	-

Turpmākā darba izstrādē darba autori izmanto Toad Data Modeler 4.3.3.6. versiju.

No iepriekšējām versijām darbā izmantotā atšķiras ar to, ka ir uzlabota lietotāja saskarne un modeļa cilnes ir iespēja aizvērt ar „krustiņu”, kā arī ir izveidota iespēja definēt jaunus Modeļu tipus (Settings | Options | Model). Kā noklusētais modeļa tips tiek iestatīt pēdējais izmantotais. Ir izveidota jauna Modeļu verificēšanas iespēja, kas piedāvā izvēlēties starp visa modeļa vai atsevišķu tā daļu verificēšanu. Verificēšanu tagad var veikt pirms skripta ģenerēšanas.

1.4. Rīka instrumentārijs



Rīkjoslas:



Attēlā apskatāmas sekojošas rīkjoslas un to nodrošinātā funkcionalitāte:


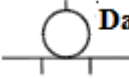
- ✓ Lietotnes rīks:
 - ~ atvērt jaunu modeli;
 - ~ eksistējošā modeļa atvēršanas dialoga logs;
 - ~ saglabāt modeli;
 - ~ apvērstās inženierijas veidne;
 - ~ pievienot modeli versiju menedžerim;
 - ~ drukāšanas dialoga logs;
 - ~ pirms drukas aplūkošanas dialoga logs;
 - ~ lietotnes pārlūka logs;
 - ~ opciju dialoga logs;
 - ~ lapas formāta izvēles logs;
 - ~ modeļa pārlūka logs;
 - ~ modeļa pārbaudes dialoga logs;
 - ~ skripta ģenerēšanas dialoga logs;
 - ~ atskaišu veidošanas veidne;
- ✓ Designer - instrumentārijs fiziskajam modelim:
 - ~ izvēles veikšanas kursori;
 - ~ pievienot entītijai modeli;

- ~ pievienot identificējošu saiti modelī;
- ~ pievienot neidentificējošu saiti modelī;
- ~ pievienot M:N saiti modelī;
- ~ pievienot Skatu modelī;
- ~ pievienot Materializētu skatu modelī;
- ~ pievienot piezīmi modelī;
- ~ pievienot piezīmju līniju modelī;
- ~ pievienot izstrādāšanas zīmogu modelī;
- ~ pievienot uzrakstu kategorijām modelī;
- ✓ Designer - instrumentārijs loģiskajam modelim:
 - ~ izvēles veikšanas kursori;
 - ~ pievienot entītijai modelī;
 - ~ pievienot identificējošo saiti modelī;
 - ~ pievienot neidentificējošu saiti modelī;
 - ~ pievienot mantošanas struktūru modelī;
 - ~ pievienot piezīmi modelī;
 - ~ pievienot piezīmju līniju modelī;
 - ~ pievieno izstrādāšanas zīmogu modelī;
 - ~ pievieno uzrakstu kategorijām modelī;
- ✓ Designer - instrumentārijs meta modelim:
 - ~ izvēles veikšanas kursori;
 - ~ pievienot klasi meta modelī;
 - ~ pievienot asociāciju meta modelī;
 - ~ pievienot ģeneralizāciju meta modelī;
 - ~ pievienot piezīmi meta modelī;
 - ~ pievienot piezīmju līniju meta modelī;
 - ~ pievienot izstrādāšanas zīmogu meta modelī;
 - ~ pievienot uzrakstu kategorijām meta modelī;
- ✓ Darba telpa:
 - ~ mērogošanas rīks;
 - ~ samazināt;
 - ~ palielināt;
 - ~ procentuāla mērogošana;
 - ~ attēlošanas līmeņa izvēlne;
 - ~ izklājuma dialoga logs;
 - ~ izlīdzina izvēlēto saišu punktus.

1.5. Notācijas

Modeļu veidošanai *Toad Data Modeler* rīkā ir iespējams izmantot divas notācijas: **IDEF1X** un **IE** (*Information Engineering*). **IDEF1X** metodoloģija tika izstrādāta ASV armijas vajadzībām un ir plaši pielietojama ASV valsts iestādes, finanšu un rūpniecības korporācijās. **IE** metodoloģija ir pielietojama, galvenokārt, rūpniecībā.

Notācijas elements	IDEF1X notācija	IE notācija	Apraksts
Realitāte			
Identificējošā saite			Realitātes nosaukums atrodas virs taisnstūra. Iekšā atrodas atribūtu saraksts, kur atslēgas atribūti iedalīti citā krāsā vai atdalīti ar svītru.
Neideintificējošā saite			
Saite 1 : 0,1			
Saite 1 : 1, N			
Saite 1 : 0.1 N			
Saite 1 : X			Tiek definēta konstanta vērtība.
Saite N : M			Saites tips, kas liecina par analīzes nepabeigtību; modelēšanas turpmākajos etapos tā tiek pārveidota citā saitē.
Hierarhiskā saite			Realitātes saite pašai ar sevi.
Pilnā kategoriju hierarhija			Izslēdzošā kategoriju hierarhija.

			Iekļaujoša kategoriju hierarhija.
Nepilnā hierarhija	kategoriju  Darbinieka tips		

1.6. Modeļi

Toad Data Modeler(TDM) vidē ir iespējams veidot fizisko un loģisko modeli.

Loģiskais līmenis – abstrakts skats uz datiem. Šajā līmenī dati tiek izskatīti tā kā tie ir reālajā dzīvē. Modeļa objekti, kas tiek piedāvāti loģiskajā līmenī, saucas par realitātēm (entītijām) un atribūtiem. Loģiskais modelis ir universāls un šajā līmenī nav saistīts ar konkrētu realizāciju DBV sistēmā.

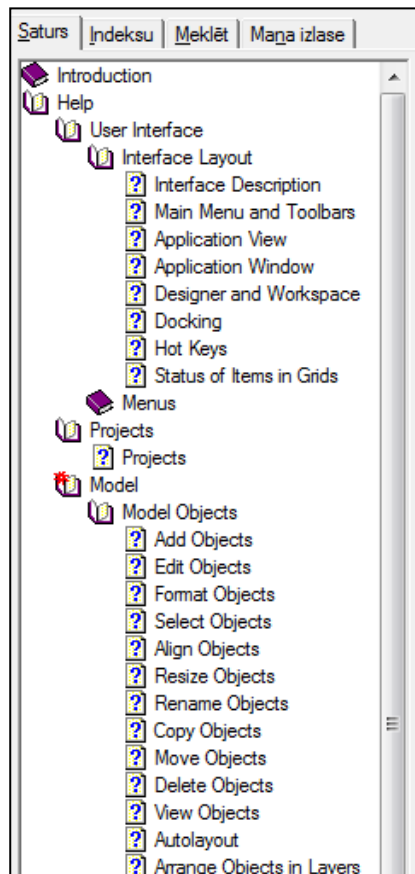
Fiziskais modelis ir loģiskais modelis, kurš ir piesaisīts kādai konkrētai datubāzu vadības sistēmai.

Tā kā *TDM* atbalsta vairākas DBVS, pirms modeļa veidošanas ir jānokonkretizē un jāizvēlas kāda no DB sistēmām. Pēc tam ir iespējams pievienot entītijas, un nodefinēt to īpašības.

1.7. Dokumentācija

Toad Data Modeler ir iekļauts plašs palīdzības fails. Tas detalizēti apraksta katru darbību, ko iespējams veikt. Tiek doti arī lietderīgi padomi dažādām situācijām. TDM palīdzības failā tiek plaši izmantoti attēli, kas ir reta prakse (tas gan arī nozīmē, ka instalācija aizņem vairāk vietas uz datora cietā diska). Dokumentācijā pat ir iekļauts gatavs JavaScript kods, kas ļauj ģenerēt HTML atskaites, SQL kodu utt.

Palīdzības faila „dziļuma” piemērs:



Piemērs ar attēlu izmantošanu:

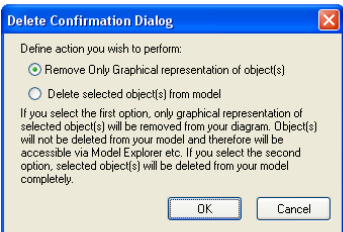
To delete objects from model (including all its shortcuts)

- Select the shortcut of object on Workspace and press **SHIFT+Delete**.

Tip: You can also delete the objects:

- In **Model Explorer** - select an object | right-click | **Delete Item**.
- In the Object dialog (**Model** menu | **Entities, Relationships** etc.) - select the object and click **Delete**.

When you press **Delete** or **SHIFT+Delete** in your ER diagram, the following message will display:

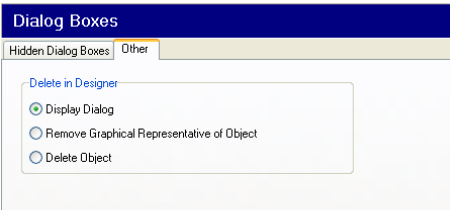


Select the action you want to perform.

Note: Key shortcuts: ALT+O for OK and ALT+C for Cancel.

To set the default Delete options in Toad Data Modeler:

- Select **Settings | Options | Dialog Boxes | Other** tab.



Option	Description
Display Dialog	Whenever you press Delete or SHIFT+Delete in your ER diagram, the Delete Confirmation Dialog will pop up and you will be able to select what action you want to perform in particular case.
Remove Graphical Representative of Object	If this option is selected, the Delete will always remove selected shortcut(s) of object from particular Workspace. SHIFT+Delete will have to be used to completely delete an object from your model.
Delete Object	If this option is selected, the Delete will completely delete selected object(s) from your model, including all the object shortcuts. There is no option to only remove selected shortcut(s) of object. (The Delete will replace SHIFT+Delete .)

1.8. Papildus iespējas

1.8.1. Version Manager

Toad Data Modeler rīkā ir iespējams veidot projektus, pievienot modeļus un citus failus, mainīt modeļus, veidot to versijas. Ir iespējams veidot neierobežotu projektu skaitu. Lai pievienotu esošo modeli kādam konkrētam projektam, ir jāizveido jauns projekts, izmantojot *Add Project*. Pēc tam projektam ir jāpievieno esošais modelis, izmantojot *Add to Version Manager*.

Pēc apstiprināšanas, izvēlētais modelis (tāpat kā katrs no jauna pievienotais modelis) tiks pievienots konkrētam projektam, līdz ar to vienmēr būs iespēja apskatīt projekta vairākas versijas, kuras ir bijušas pievienotas projektam ar *Version Manager*.

1.8.2. To-Do list

To-Do saraksts ļauj veidot ierakstus, kas atgādina uzdevumus, kurus ir jāveic vai atgādina par nepabeigtām darbībām.

To-Do list ir jāpievieno:

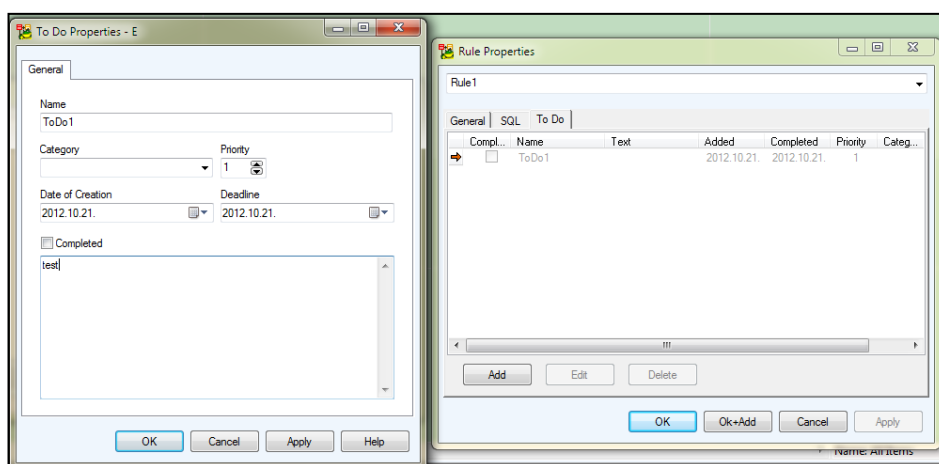
- ✓ kāda modeļa daļa, uzklikšķot uz objekta ar labo peles pogu un *Edit* izvēlnē izvēloties *Properties*, kurā tālāk jāizvēlas *To Do* cilne. Rezultātā tiks attēlots uzdevumu saraksts, kas ir saistīts ar izvēlēto objektu;

- ✓ galvenais *To Do* dialoga logs, kas atrodas zem *Model* → *To Do*. Rezultātā tiek parādīts pilns uzdevumu saraksts. Lai pievienotu jaunu uzdevumu, ir jāizvēlas *Add* opcija, lai izmainītu esošo uzdevumu, ir jāizvēlas opcija *Edit*, savukārt lai izdzēstu uzdevumu, ir jāizvēlas opcija *Delete*.

Toad Data Modeler vidē var piesaistīt uzdevumus ne tikai modelim kopumā, bet arī atsevišķiem objektiem.

Objektu saraksts, kuriem var pievienot uzdevumus:

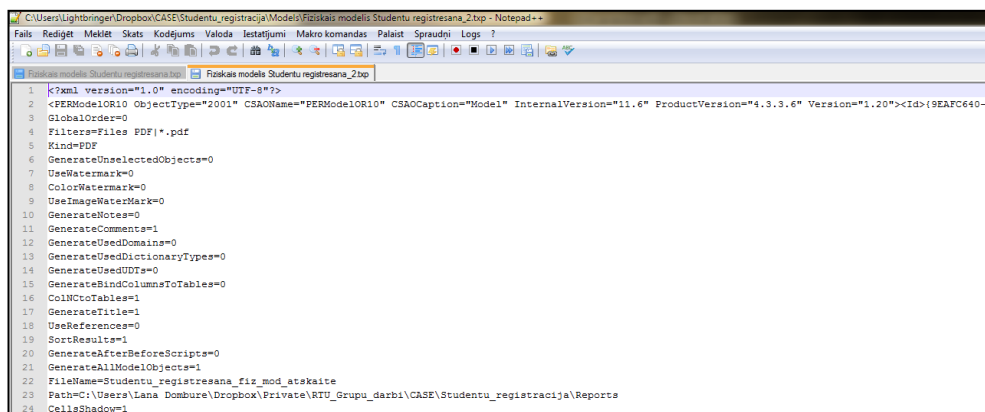
- | | | | | | |
|---|-----------|---|---------------|---|-------------------|
| ✓ | modelis, | ✓ | atslēgas, | ✓ | lietotāji, |
| ✓ | entītija, | ✓ | indeksi, | ✓ | lietotāju grupas. |
| ✓ | saite, | ✓ | ierobežojumi, | | |
| ✓ | atribūti, | ✓ | trigieri, | | |



1.8.3. XML

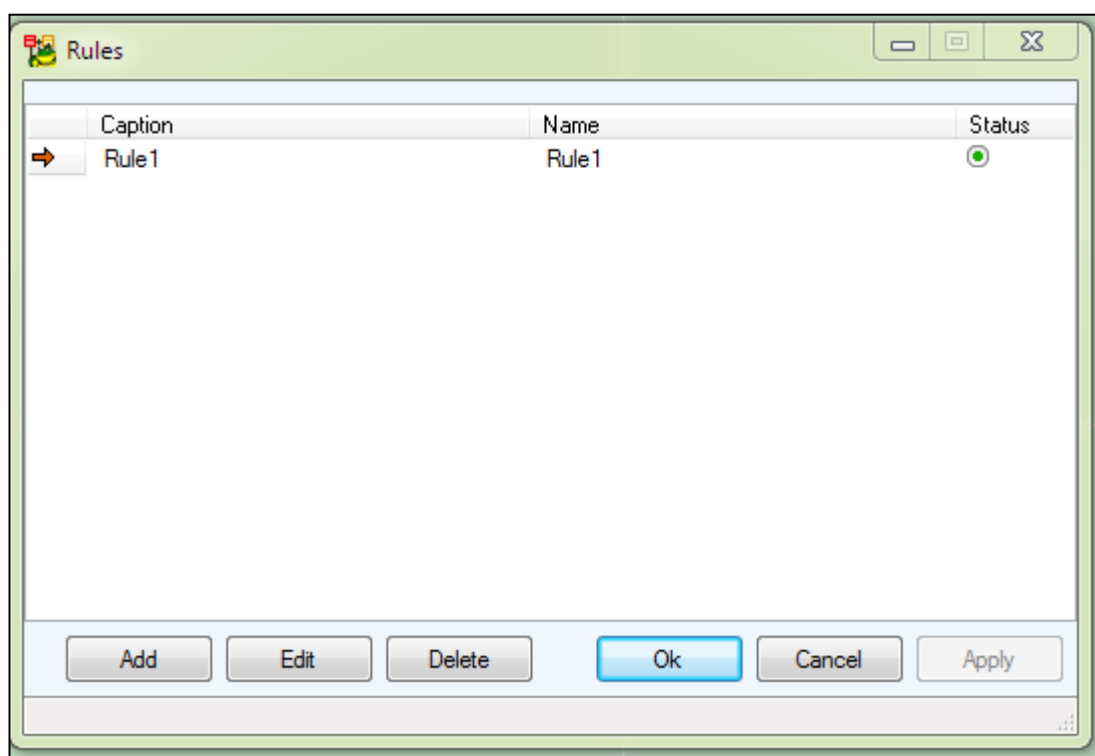
Toad Data Modeler modeļi tiek saglabāti kā *.tdm*. Tie ir XML faili, un tos var brīvi rediģēt ar jebkuru teksta redaktoru, kas atbalsta XML failus. Tas ir lietderīgi, ja ir nepieciešam pārskatīt modeļa struktūru.

Piemērs ar atvērtu TDM modeli programmā *Notepad++*

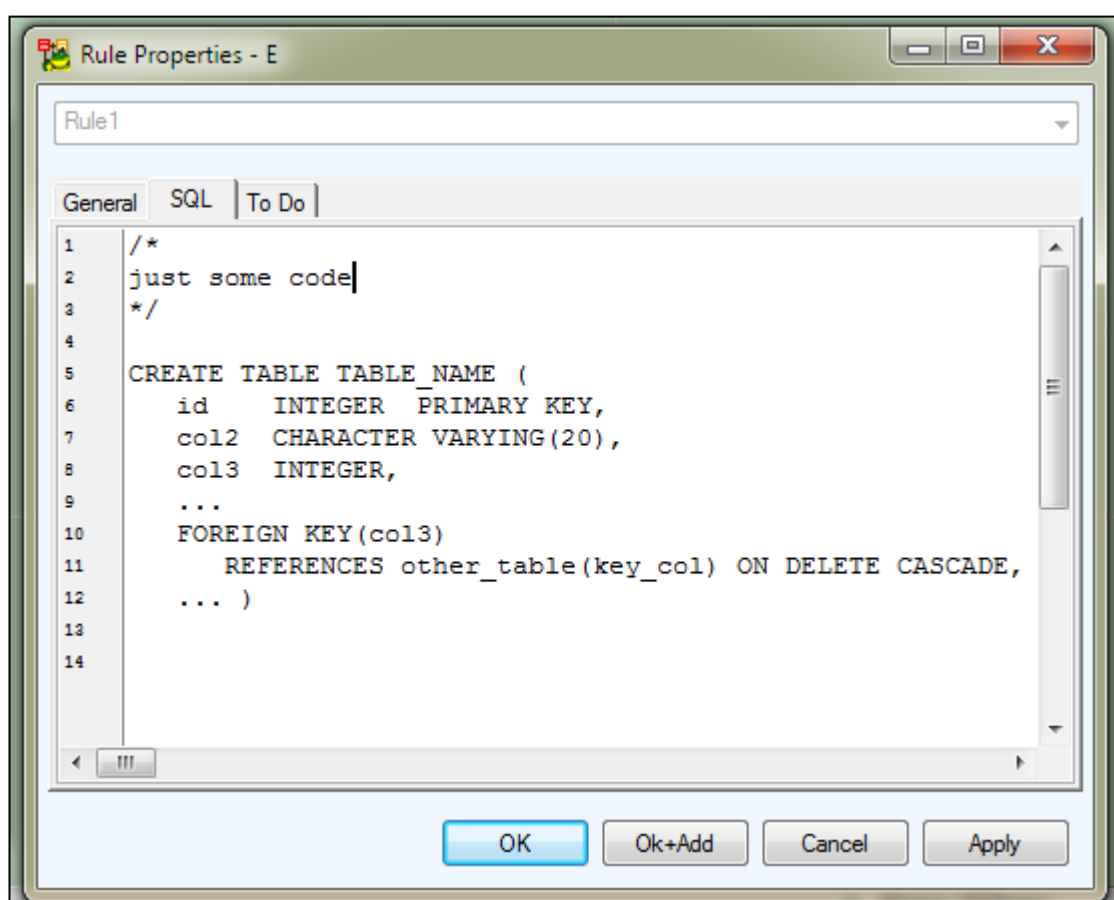


1.8.4. Rules

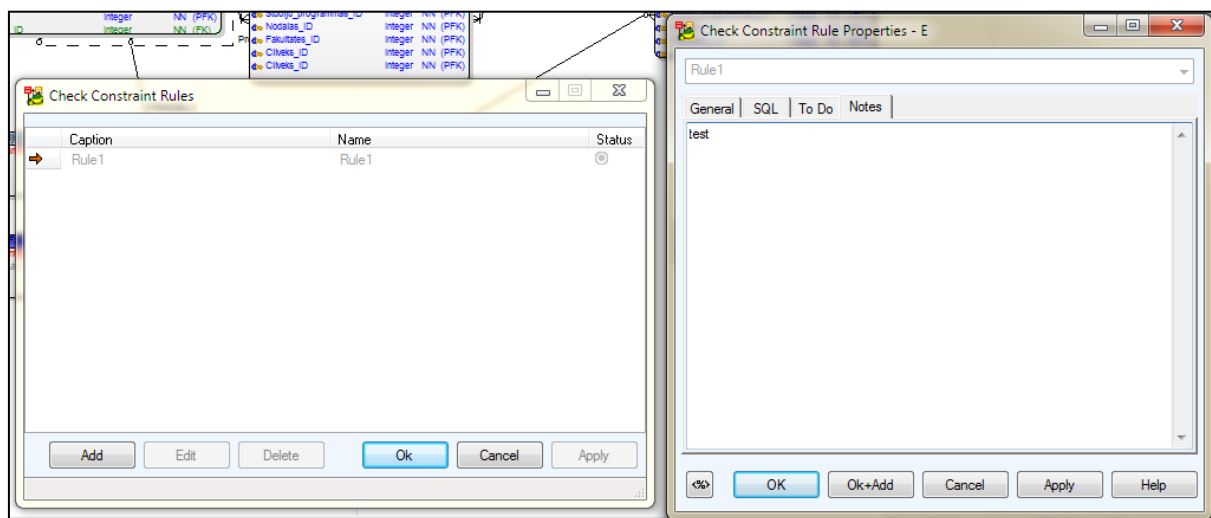
TDM modeļiem ir iespējams pievienot likumus.



Tas ir SQL kods, kas izpildās pie noteiktiem nosacījumiem.



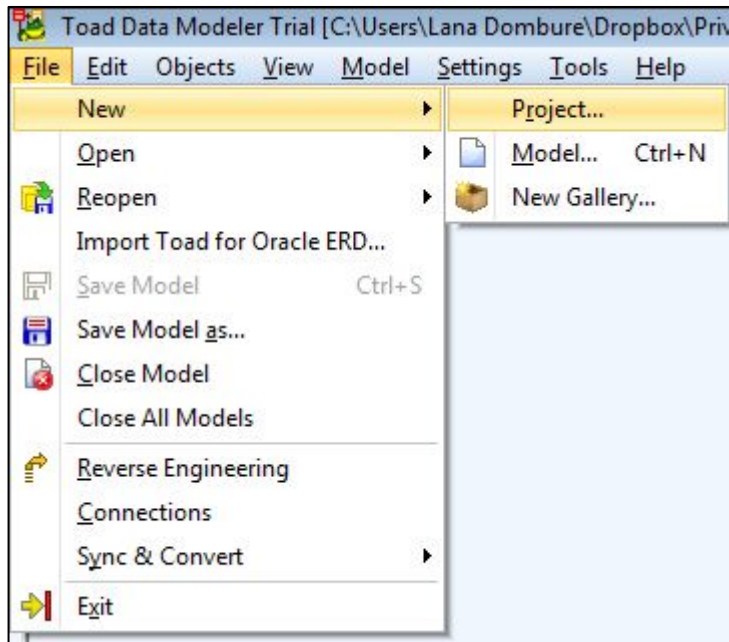
Ir arī iespējams izveidot Constraint Rules jeb ierobežojošos likumus. Tiem ir papildus iespēja pievienot piezīmes – Notes.



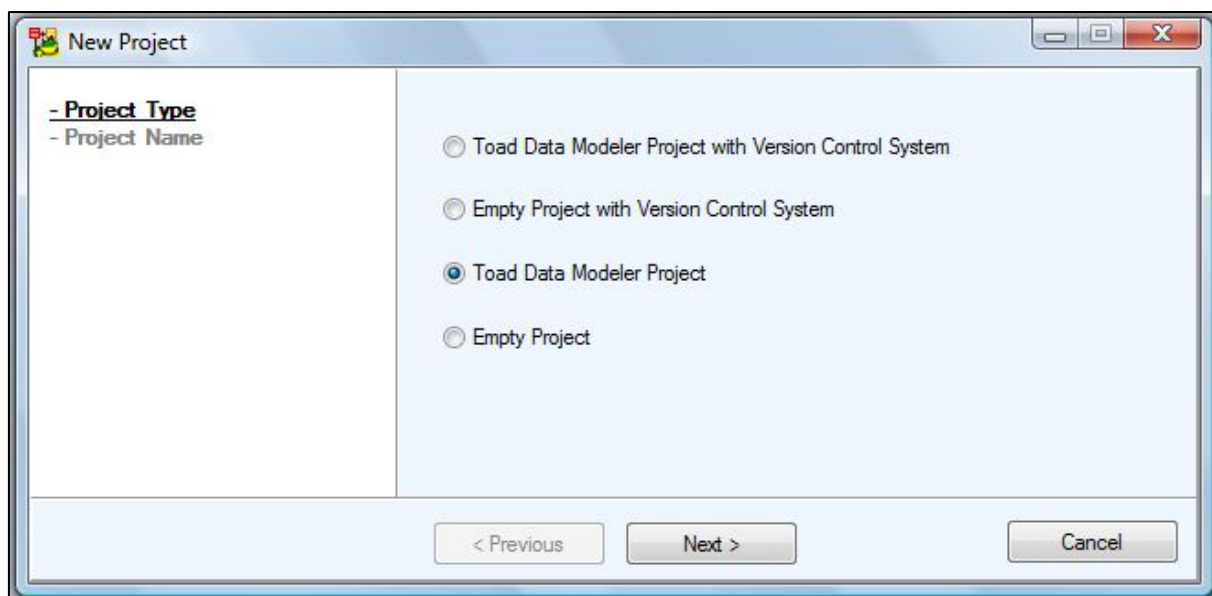
2. PRAKTISKĀ DAĻA

2.1. Jauna projekta izveide

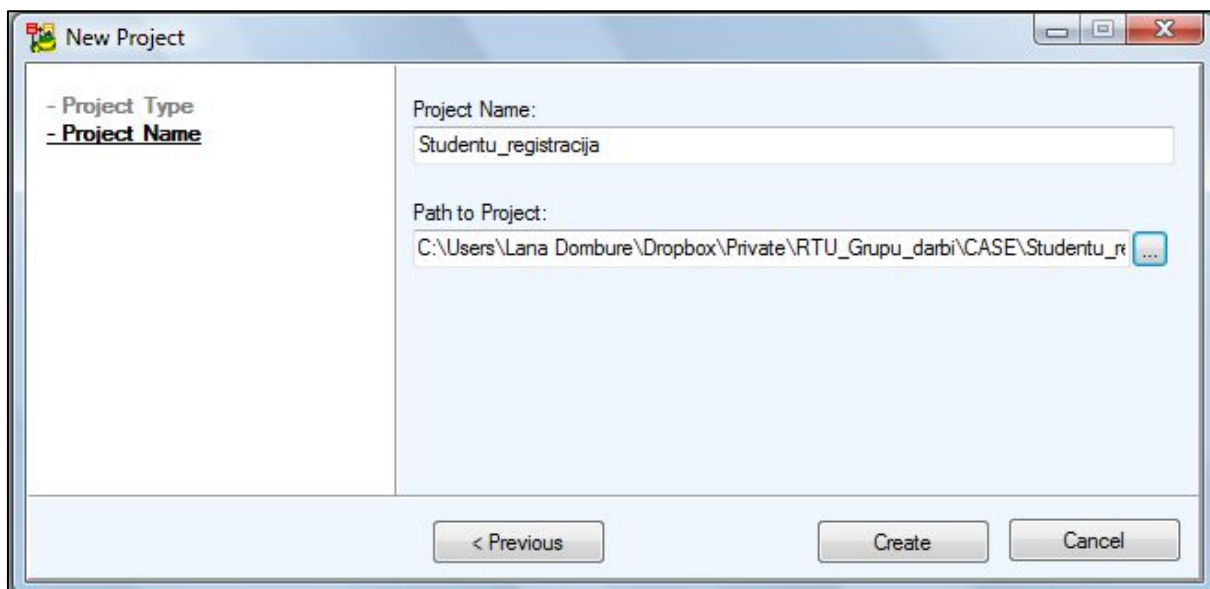
Lai sāktu darbu ir nepieciešams izveidot jaunu projektu, ko veic galvenajā izvēlnē izvēloties File | New | Project...



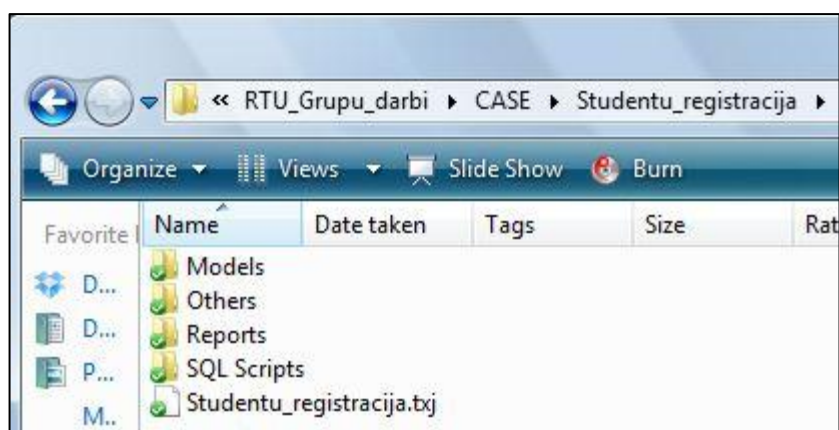
Sākotnēji tiek piedāvāts izvēlēties starp 4 projekta tipiem. Darba autori izvēlējās strādāt ar Toad Data Modeler Project.



Tiek definēts projekta nosaukums un tā atrašanās vieta.

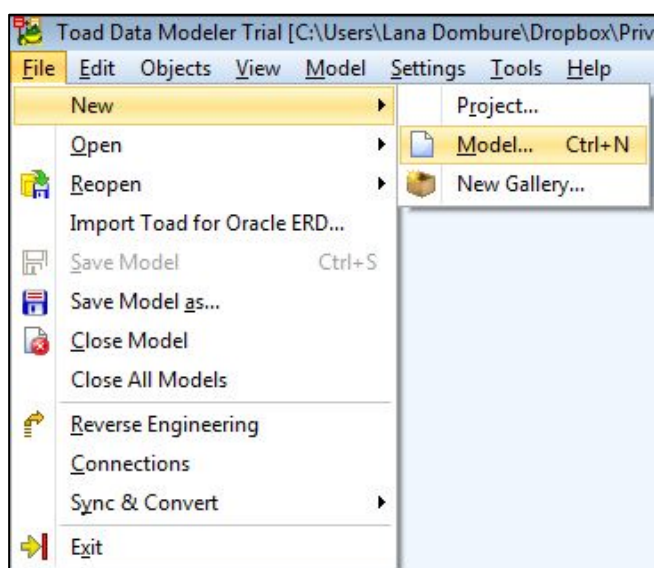


Automātiski izveidotā projekta mapju struktūra apskatāma zemāk.

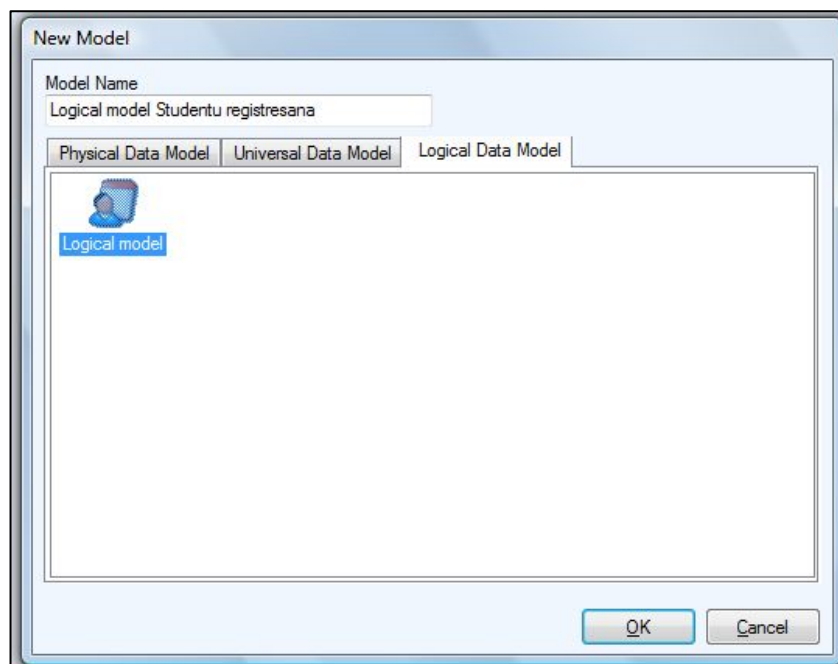


2.2. Loģiskā modeļa izveide

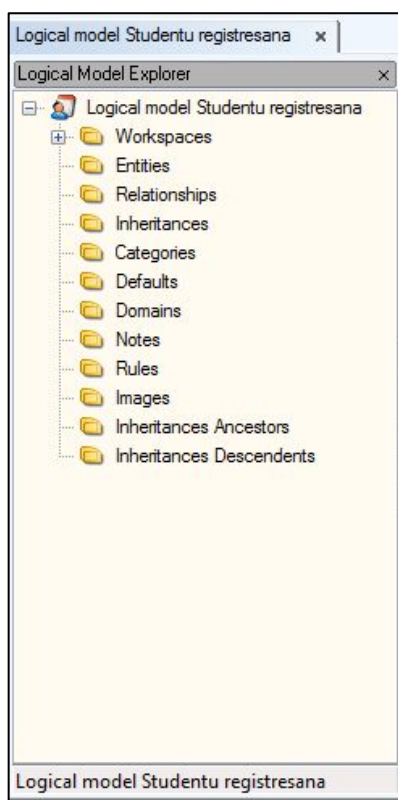
Lai izveidotu jaunu loģisko modeli, lietotājs galvenajā izvēlnē atver File | New | Model...



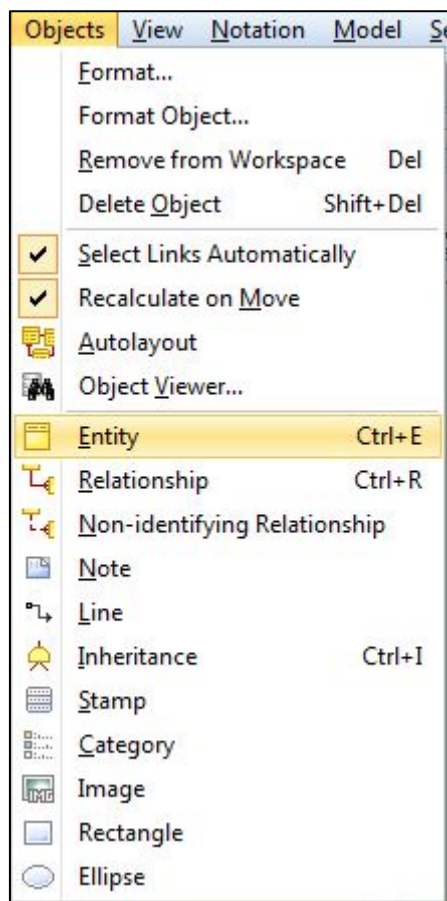
Lietotājam tiek piedāvāts logs loģiskā modeļa izvēlei.



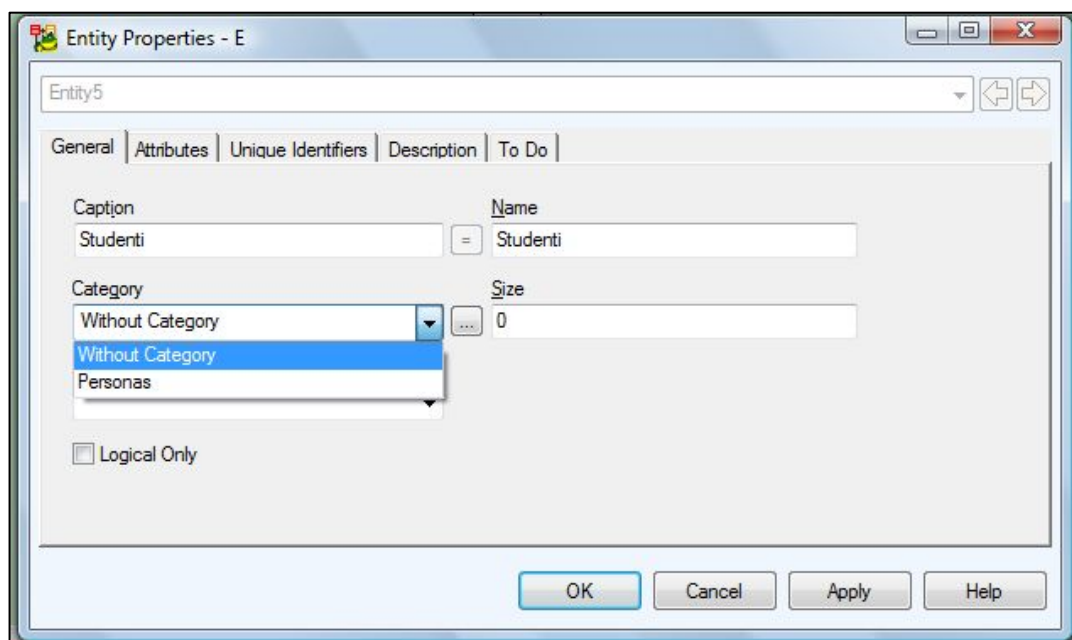
Pēc loģiskā modeļa izveides, lietotājam tiek atvērts darba virsma, kuras kreisajā pusē izvietots modeļa strukturēts pārlūks.



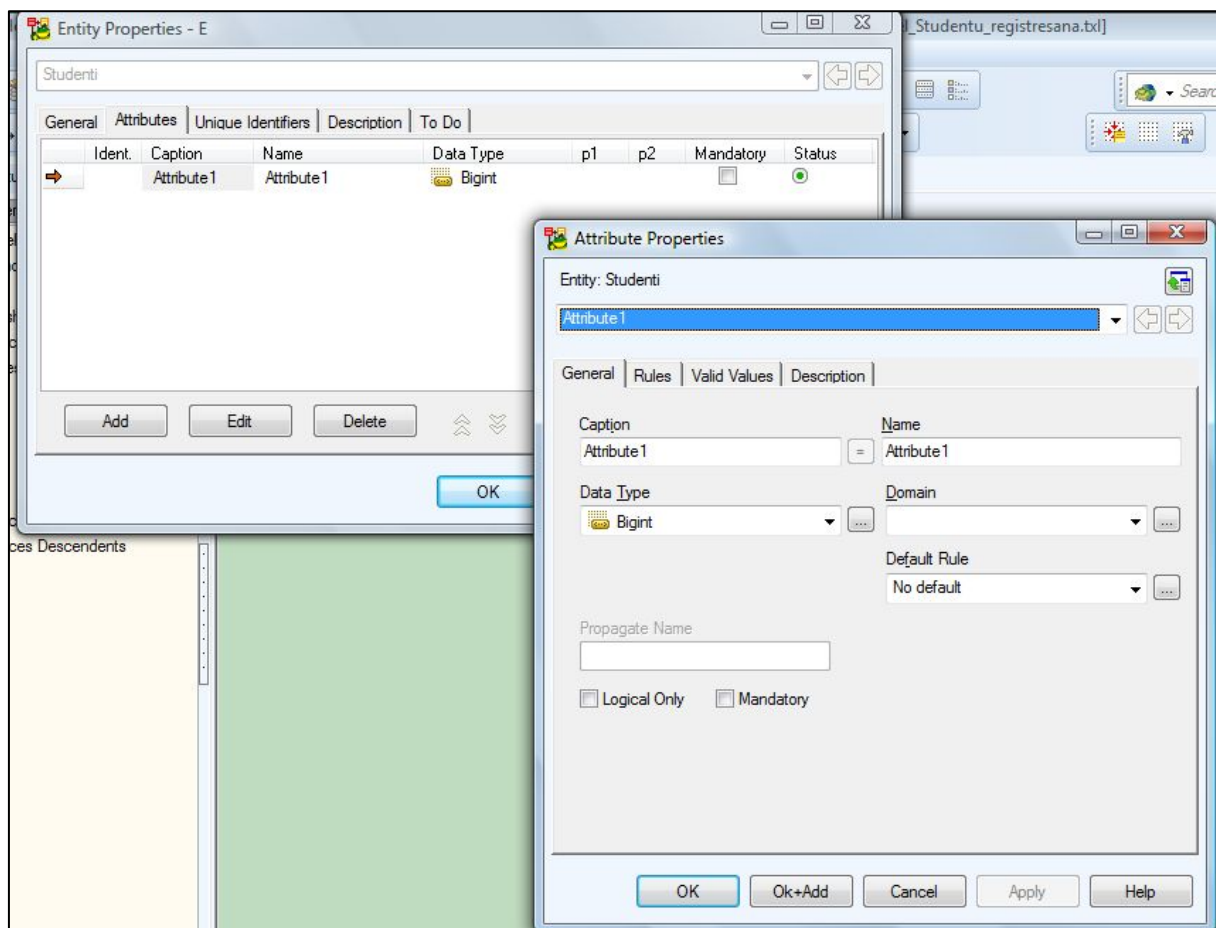
Lai modeli izveidotu tiek izmantoti piedāvātie objekti, kuri pieejami galvenajā izvēlnē Objects vai arī objektu rīkjoslā Designer.



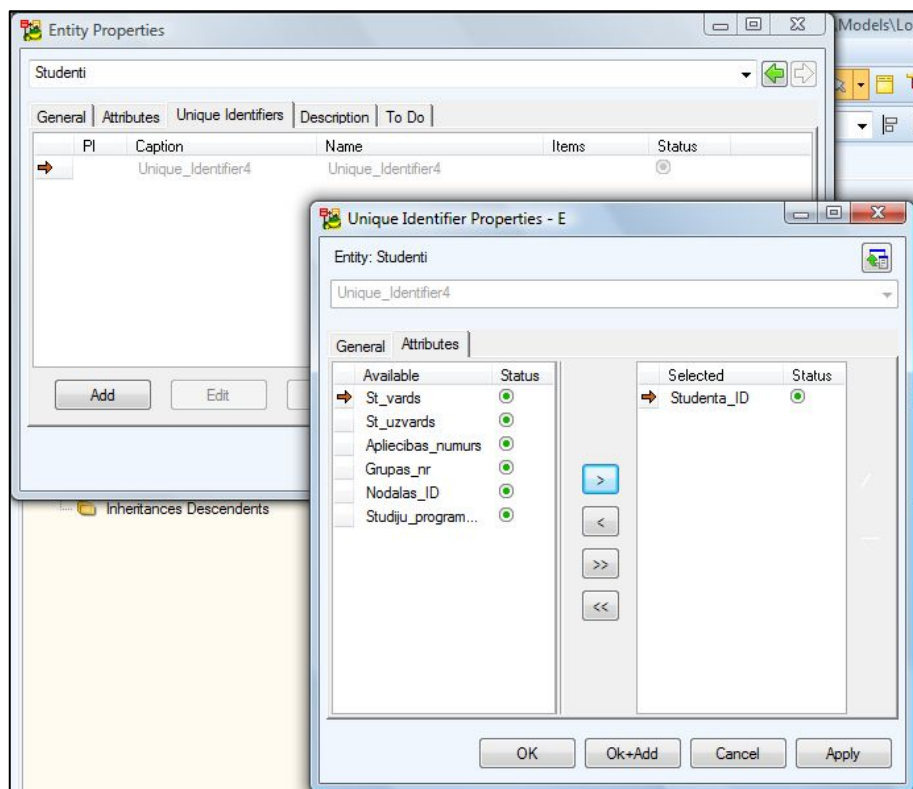
Izvēloties izveidot jaunu realitāti, un ar dubultklikšķi atverot jaunās realitātes īpašības, tiek piedāvāta iespēja definēt realitātes nosaukumu, laukus, kā arī pievienot konkrēto realitāti kādai kategorijai.



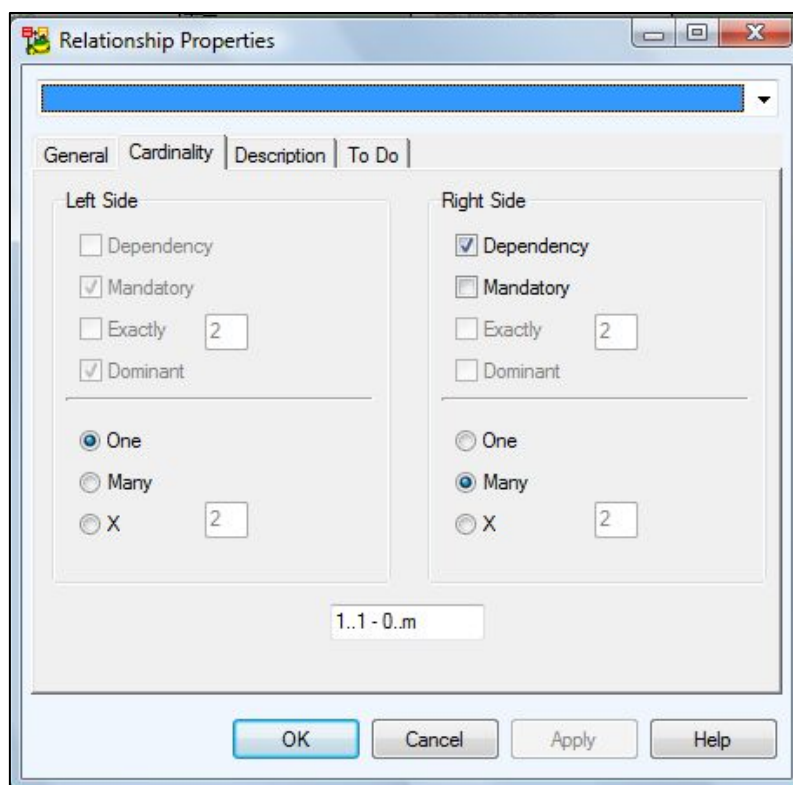
Definējot jaunus realitātes laukus, tiem izvēlas arī datu tipu.



Unikālo atslēgu definēšanas logi.



Pēc visu realitāšu definēšanas, darba autori izveido saites.

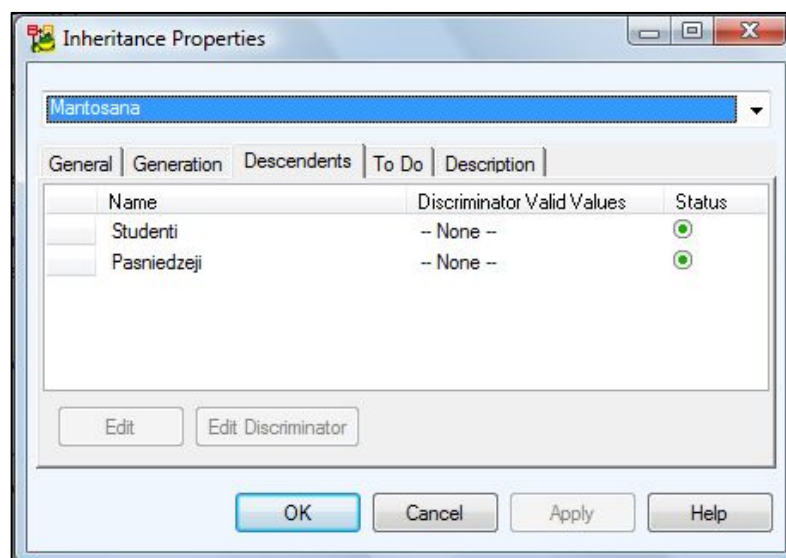
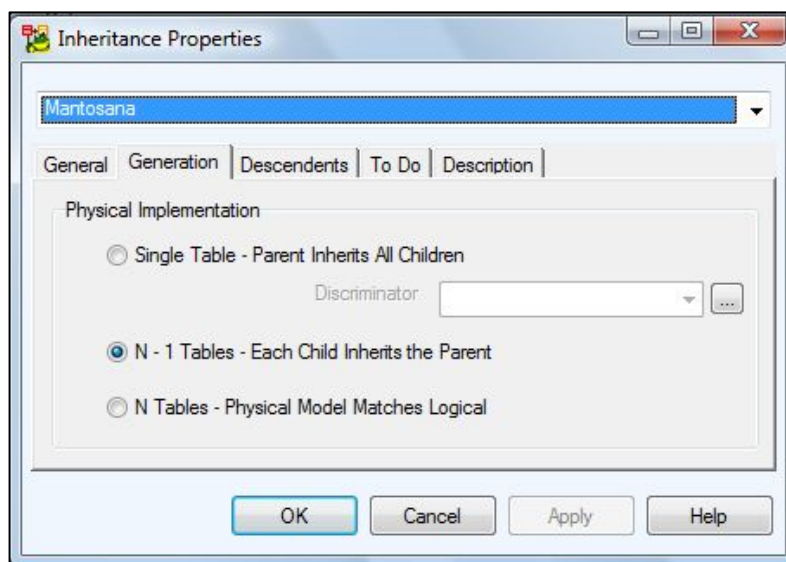
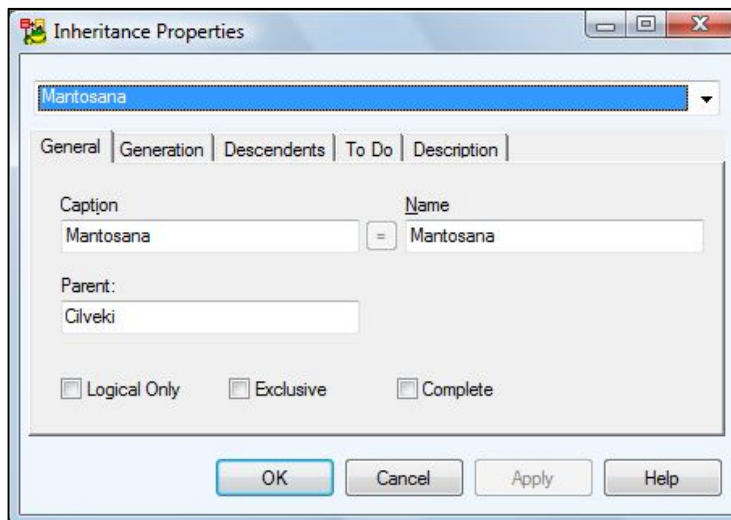


Konkrētajā loģiskajā modelī tiek izmantotas binārās un urālās saites, un Toad Data Modeler nodrošina iespēju izveidot arī ternārās saites.

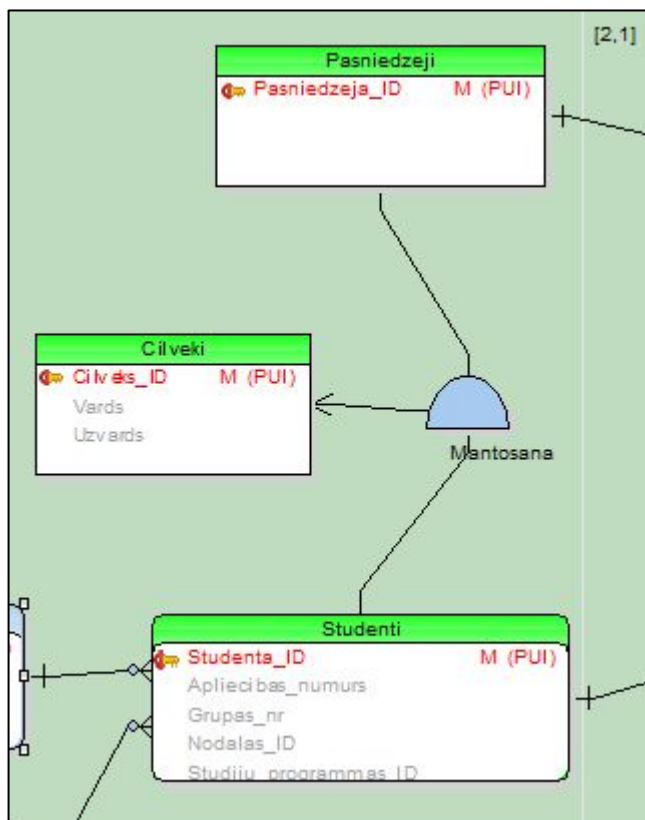
2.3. Mantošanas realizācija

Loģiskajā modelī tika izveidota realitāte Cilveks, kurai definēti atribūti Vards un Uzvars. Realitāte Cilveks ir Parent realitāte, bet realitātes Pasniedzēji un Studenti ir Child.

No Objektu saraksta ir jāizvēlas rīks Inheritance. Ar peles kreiso taustiņu jāuzspiež uz Parent realitāti, un pēc tam uz vienu no Child realitātēm. Pēc tam atkal jāaktivizē rīks Inheritance un no Mantošanas saites jānovelk uz otru Child realitāti.

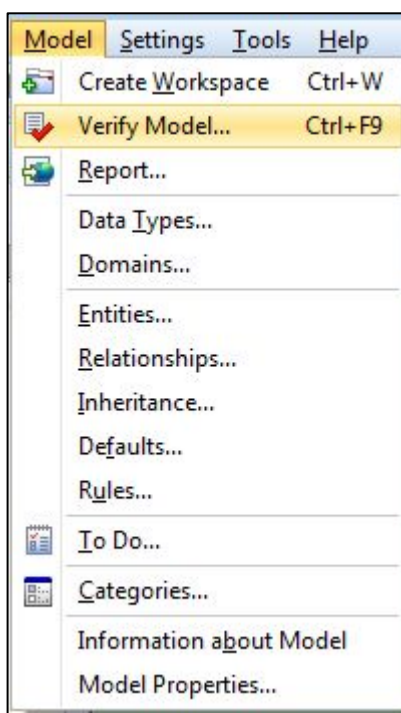


Mantošanas rezultāts loģiskajā modelī vizuāli tiek attēlots:

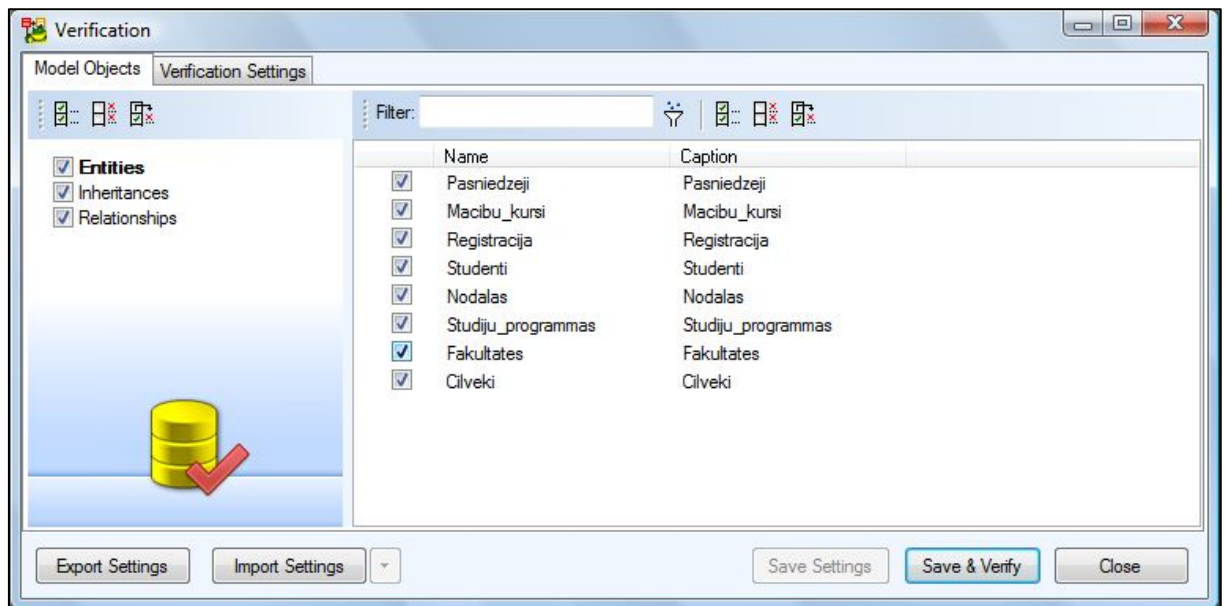


2.4. Loģiskā modeļa verificēšana

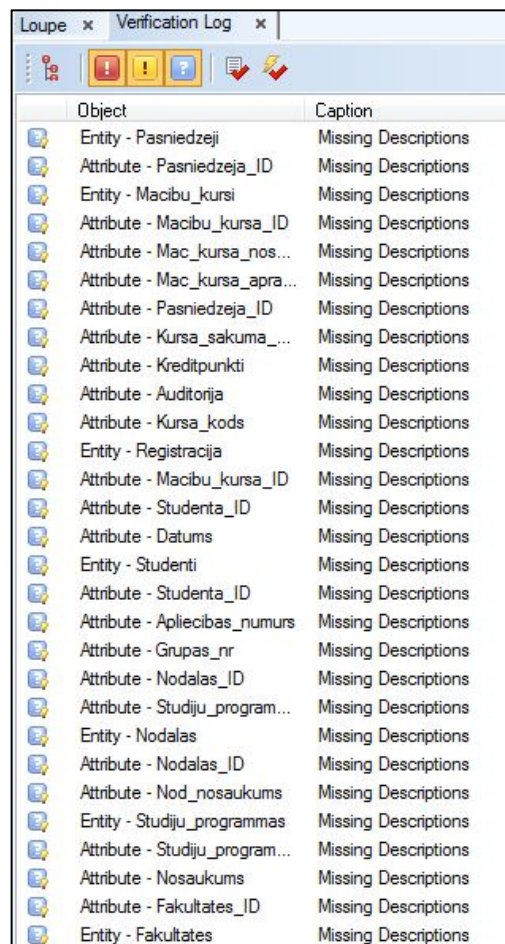
Lai verificētu loģisko modeli no galvenās izvēlnes jāizvēlas Model | Verify Model...



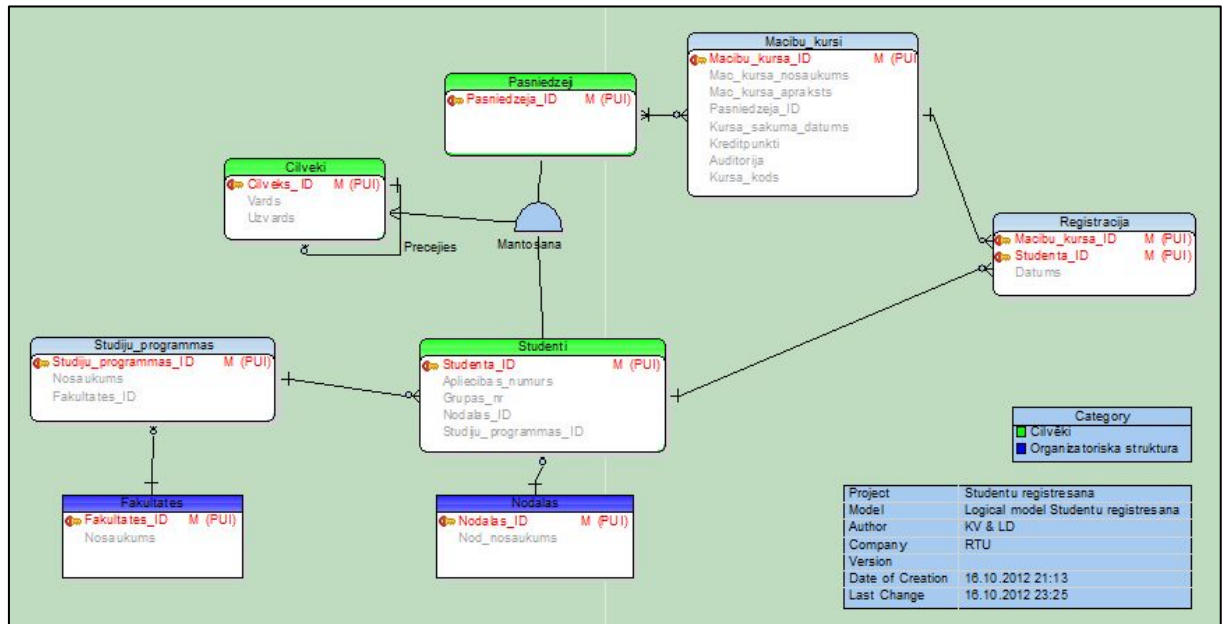
Tiek piedāvāta iespēja verificēt visu modeli vai tikai daļu tā elementus



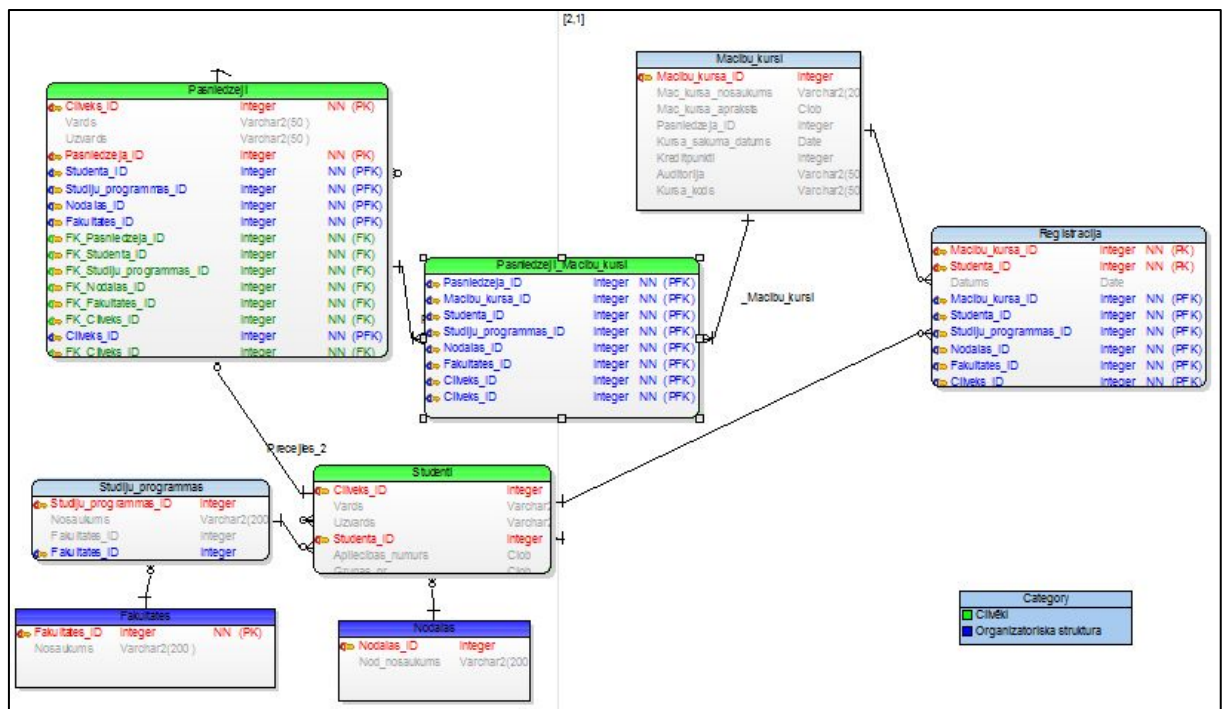
Kā redzams verificēšanas atskaites logā, ar izveidoto loģisko modeli viss ir kārtībā un kļūdas netika atrastas.



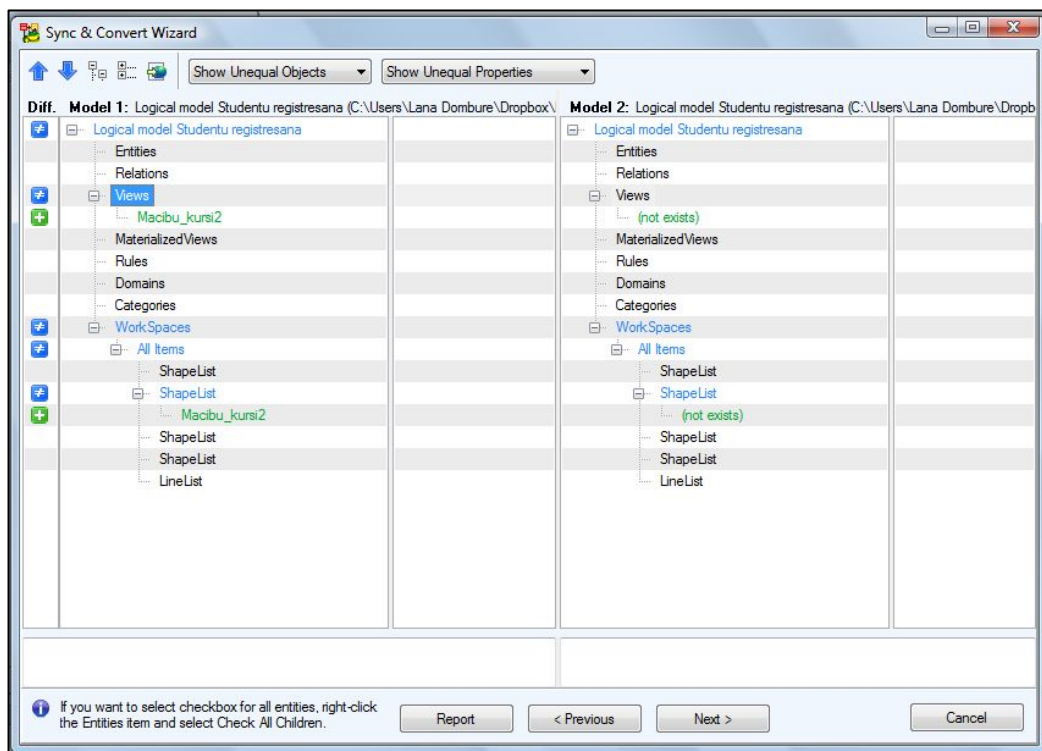
2.5. Kopējais loģiskais modelis



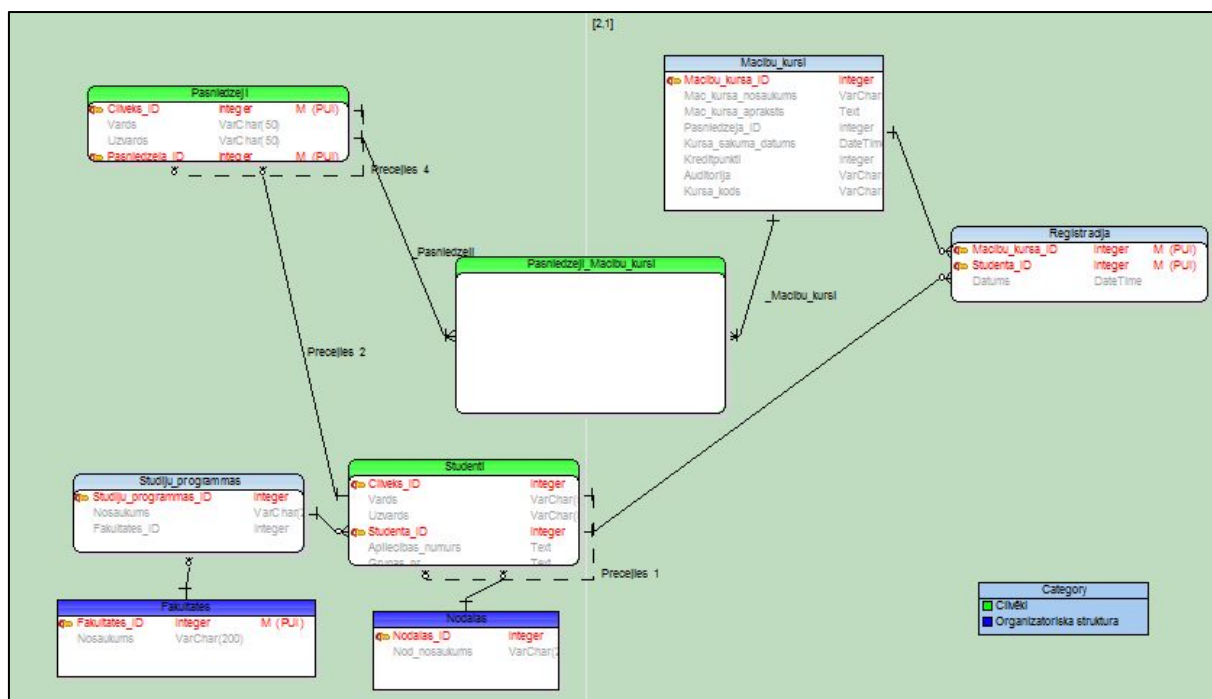
2.6. Loģiskā modeļa transformācija fiziskajā modelī



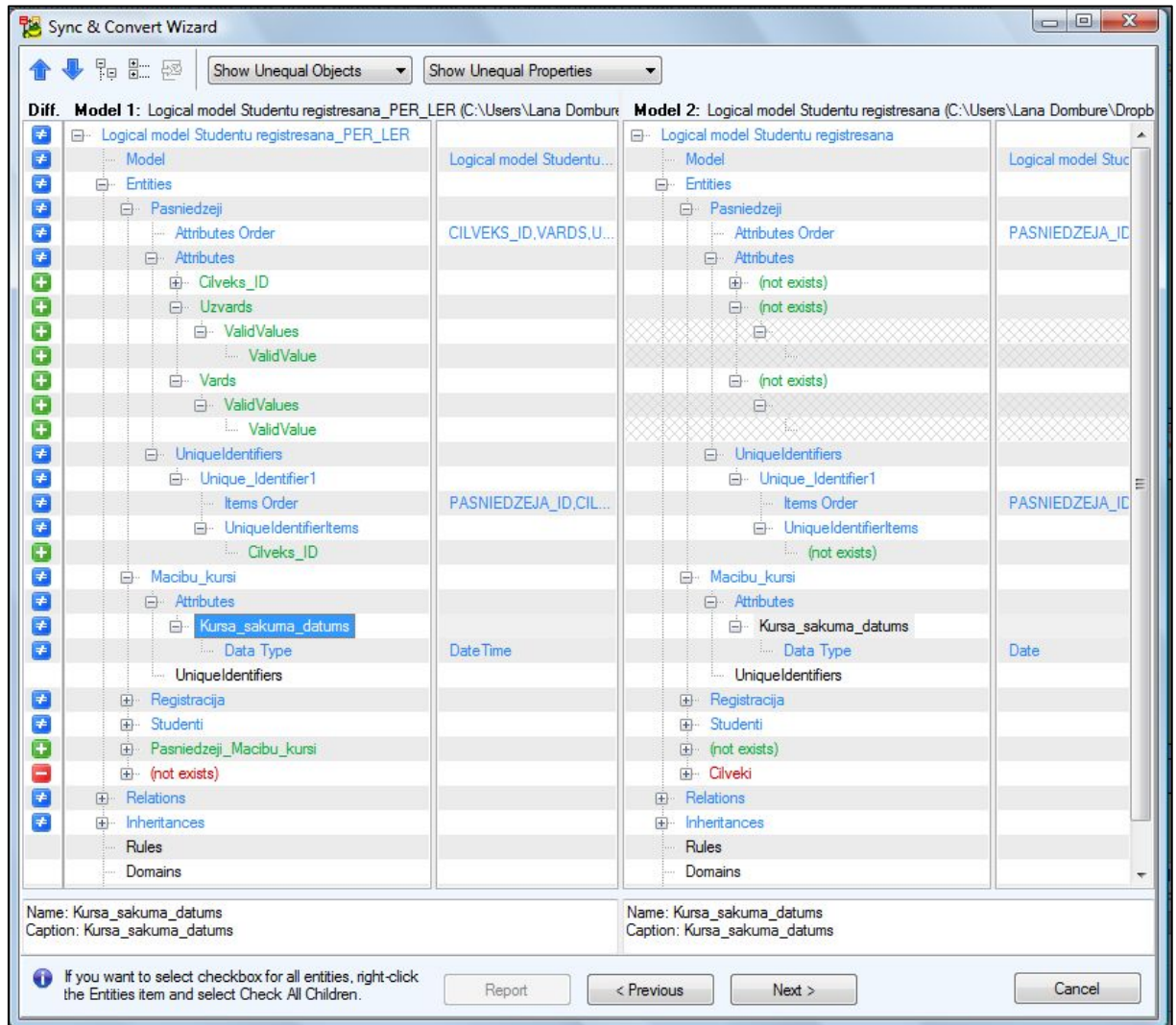
2.7. Modeļu salīdzināšana



2.8. PER -> LER



2.9. Loģisko modeļu salīdzinājums



2.10. Ģenerētais SQL skripts

/*

Created: 17.10.2012

Modified: 17.10.2012

Project: Studentu registresana

Model: Logical model Studentu registresana

Company: RTU

Author: KV & LD

Database: Oracle 10g

*/

-- Create tables section -----

-- Table Pasniedzēji

```
CREATE TABLE "Pasniedzēji"(
  "Cilvēks_ID" Integer NOT NULL,
  "Vards" Varchar2(50),
  "Uzvards" Varchar2(50),
  "Pasniedzēja_ID" Integer NOT NULL,
  "Studenta_ID" Integer NOT NULL,
  "Studiju_programmas_ID" Integer NOT NULL,
  "Nodalas_ID" Integer NOT NULL,
  "Fakultates_ID" Integer NOT NULL,
  "FK_Pasniedzēja_ID" Integer NOT NULL,
  "FK_Studenta_ID" Integer NOT NULL,
  "FK_Studiju_programmas_ID" Integer NOT NULL,
  "FK_Nodalas_ID" Integer NOT NULL,
```

```

"FK_Fakultates_ID" Integer NOT NULL,
"FK_Cilveks_ID" Integer NOT NULL,
"Cilveks_ID" Integer NOT NULL,
"FK_Cilveks_ID" Integer NOT NULL
)
/
-- Add keys for table Pasniedzaji
ALTER TABLE "Pasniedzaji" ADD CONSTRAINT "Unique_Identifier1" PRIMARY KEY
("Pasniedzaja_ID","Studenta_ID","Studiju_programmas_ID","Nodalas_ID","Fakultates_ID","Cilveks_ID","Cilveks_ID")
/
-- Table Macibu_kursi
CREATE TABLE "Macibu_kursi"(
"Macibu_kursa_ID" Integer NOT NULL,
"Mac_kursa_nosaukums" Varchar2(200 ),
"Mac_kursa_apraksts" Clob,
"Pasniedzaja_ID" Integer,
"Kursa_sakuma_datums" Date,
"Kreditpunkti" Integer,
"Auditorija" Varchar2(50 ),
"Kursa_kods" Varchar2(50 )
)
/
-- Add keys for table Macibu_kursi
ALTER TABLE "Macibu_kursi" ADD CONSTRAINT "Unique_Identifier2" PRIMARY KEY ("Macibu_kursa_ID")
/
-- Table Registracija
CREATE TABLE "Registracija"(
"Macibu_kursa_ID" Integer NOT NULL,
"Studenta_ID" Integer NOT NULL,
"Datums" Date,
"Macibu_kursa_ID" Integer NOT NULL,
"Studenta_ID" Integer NOT NULL,
"Studiju_programmas_ID" Integer NOT NULL,
"Nodalas_ID" Integer NOT NULL,
"Fakultates_ID" Integer NOT NULL,
"Cilveks_ID" Integer NOT NULL
)
/
-- Add keys for table Registracija
ALTER TABLE "Registracija" ADD CONSTRAINT "Unique_Identifier3" PRIMARY KEY
("Macibu_kursa_ID","Studenta_ID","Macibu_kursa_ID","Studenta_ID","Studiju_programmas_ID","Nodalas_ID","Fakultates_I
D","Cilveks_ID")
/
-- Table Studenti
CREATE TABLE "Studenti"(
"Cilveks_ID" Integer NOT NULL,
"Vards" Varchar2(50 ),
"Uzvards" Varchar2(50 ),
"Studenta_ID" Integer NOT NULL,
"Apliecibas_numurs" Clob,
"Grupas_nr" Clob,
"Nodalas_ID" Integer,
"Studiju_programmas_ID" Integer,
"Studiju_programmas_ID" Integer NOT NULL,
"Nodalas_ID" Integer NOT NULL,
"Fakultates_ID" Integer NOT NULL,
"FK_Studenta_ID" Integer NOT NULL,
"FK_Studiju_programmas_ID" Integer NOT NULL,
"FK_Nodalas_ID" Integer NOT NULL,
"FK_Fakultates_ID" Integer NOT NULL,
"FK_Cilveks_ID" Integer
)
/
-- Add keys for table Studenti
ALTER TABLE "Studenti" ADD CONSTRAINT "Unique_Identifier4" PRIMARY KEY
("Studenta_ID","Studiju_programmas_ID","Nodalas_ID","Fakultates_ID","Cilveks_ID")
/
-- Table Nodalas

```

```

CREATE TABLE "Nodalas"(
  "Nodalas_ID" Integer NOT NULL,
  "Nod_nosaukums" Varchar2(200 )
)
/
-- Add keys for table Nodalas
ALTER TABLE "Nodalas" ADD CONSTRAINT "Unique_Identifier5" PRIMARY KEY ("Nodalas_ID")
/
-- Table Studiju_programmas
CREATE TABLE "Studiju_programmas"(
  "Studiju_programmas_ID" Integer NOT NULL,
  "Nosaukums" Varchar2(200 ),
  "Fakultates_ID" Integer,
  "Fakultates_ID" Integer NOT NULL
)
/
-- Add keys for table Studiju_programmas
ALTER TABLE "Studiju_programmas" ADD CONSTRAINT "Unique_Identifier6" PRIMARY KEY
("Studiju_programmas_ID","Fakultates_ID")
/
-- Table Fakultates
CREATE TABLE "Fakultates"(
  "Fakultates_ID" Integer NOT NULL,
  "Nosaukums" Varchar2(200 )
)
/
-- Add keys for table Fakultates
ALTER TABLE "Fakultates" ADD CONSTRAINT "Unique_Identifier7" PRIMARY KEY ("Fakultates_ID")
/
-- Table Pasniedzaji_Macibu_kursi
CREATE TABLE "Pasniedzaji_Macibu_kursi"(
  "Pasniedzaja_ID" Integer NOT NULL,
  "Macibu_kursa_ID" Integer NOT NULL,
  "Studenta_ID" Integer NOT NULL,
  "Studiju_programmas_ID" Integer NOT NULL,
  "Nodalas_ID" Integer NOT NULL,
  "Fakultates_ID" Integer NOT NULL,
  "Cilveks_ID" Integer NOT NULL,
  "Cilveks_ID" Integer NOT NULL
)
/
-- Create views section -----
CREATE VIEW "Macibu_kursi2" AS
SELECT
/
-- Create materialized views section -----
CREATE MATERIALIZED VIEW "MaterializedView1"
AS /*select_statement*/
/
-- Create relationships section -----
CREATE INDEX "IX_Pasniedzaji" ON "Pasniedzaji_Macibu_kursi" ("Pasniedzaja_ID", "Studenta_ID",
"Studiju_programmas_ID", "Nodalas_ID", "Fakultates_ID", "Cilveks_ID", "Cilveks_ID")
/
ALTER TABLE "Pasniedzaji_Macibu_kursi" ADD CONSTRAINT "_Pasniedzaji" FOREIGN KEY ("Pasniedzaja_ID",
"Studenta_ID", "Studiju_programmas_ID", "Nodalas_ID", "Fakultates_ID", "Cilveks_ID", "Cilveks_ID") REFERENCES
"Pasniedzaji" ("Pasniedzaja_ID", "Studenta_ID", "Studiju_programmas_ID", "Nodalas_ID", "Fakultates_ID", "Cilveks_ID",
"Cilveks_ID")
/
CREATE INDEX "IX_Macibu_kursi" ON "Pasniedzaji_Macibu_kursi" ("Macibu_kursa_ID")
/
ALTER TABLE "Pasniedzaji_Macibu_kursi" ADD CONSTRAINT "_Macibu_kursi" FOREIGN KEY ("Macibu_kursa_ID")
REFERENCES "Macibu_kursi" ("Macibu_kursa_ID")
/
CREATE INDEX "IX_" ON "Registracija" ("Macibu_kursa_ID")
/
ALTER TABLE "Registracija" ADD CONSTRAINT "" FOREIGN KEY ("Macibu_kursa_ID") REFERENCES "Macibu_kursi"
("Macibu_kursa_ID")
/

```

```

CREATE INDEX "IX_" ON "Registracija" ("Studenta_ID", "Studiju_programmas_ID", "Nodalas_ID", "Fakultates_ID",
"Cilveks_ID")
/
ALTER TABLE "Registracija" ADD CONSTRAINT "" FOREIGN KEY ("Studenta_ID", "Studiju_programmas_ID",
"Nodalas_ID", "Fakultates_ID", "Cilveks_ID") REFERENCES "Studenti" ("Studenta_ID", "Studiju_programmas_ID",
"Nodalas_ID", "Fakultates_ID", "Cilveks_ID")
/
CREATE INDEX "IX_" ON "Studenti" ("Studiju_programmas_ID", "Fakultates_ID")
/
ALTER TABLE "Studenti" ADD CONSTRAINT "" FOREIGN KEY ("Studiju_programmas_ID", "Fakultates_ID")
REFERENCES "Studiju_programmas" ("Studiju_programmas_ID", "Fakultates_ID")
/
CREATE INDEX "IX_" ON "Studenti" ("Nodalas_ID")
/
ALTER TABLE "Studenti" ADD CONSTRAINT "" FOREIGN KEY ("Nodalas_ID") REFERENCES "Nodalas"
("Nodalas_ID")
/
CREATE INDEX "IX_" ON "Studiju_programmas" ("Fakultates_ID")
/
ALTER TABLE "Studiju_programmas" ADD CONSTRAINT "" FOREIGN KEY ("Fakultates_ID") REFERENCES
"Fakultates" ("Fakultates_ID")
/
CREATE INDEX "IX_Precejies_1" ON "Studenti" ("FK_Studenta_ID", "FK_Studiju_programmas_ID", "FK_Nodalas_ID",
"FK_Fakultates_ID", "FK_Cilveks_ID")
/
ALTER TABLE "Studenti" ADD CONSTRAINT "Precejies_1" FOREIGN KEY ("FK_Studenta_ID",
"FK_Studiju_programmas_ID", "FK_Nodalas_ID", "FK_Fakultates_ID", "FK_Cilveks_ID") REFERENCES "Studenti"
("Studenta_ID", "Studiju_programmas_ID", "Nodalas_ID", "Fakultates_ID", "Cilveks_ID")
/
CREATE INDEX "IX_Precejies_2" ON "Pasniedzeji" ("Studenta_ID", "Studiju_programmas_ID", "Nodalas_ID",
"Fakultates_ID", "Cilveks_ID")
/
ALTER TABLE "Pasniedzeji" ADD CONSTRAINT "Precejies_2" FOREIGN KEY ("Studenta_ID", "Studiju_programmas_ID",
"Nodalas_ID", "Fakultates_ID", "Cilveks_ID") REFERENCES "Studenti" ("Studenta_ID", "Studiju_programmas_ID",
"Nodalas_ID", "Fakultates_ID", "Cilveks_ID")
/
CREATE INDEX "IX_Precejies_4" ON "Pasniedzeji" ("FK_Pasniedzeja_ID", "FK_Studenta_ID",
"FK_Studiju_programmas_ID", "FK_Nodalas_ID", "FK_Fakultates_ID", "FK_Cilveks_ID", "FK_Cilveks_ID")
/
ALTER TABLE "Pasniedzeji" ADD CONSTRAINT "Precejies_4" FOREIGN KEY ("FK_Pasniedzeja_ID", "FK_Studenta_ID",
"FK_Studiju_programmas_ID", "FK_Nodalas_ID", "FK_Fakultates_ID", "FK_Cilveks_ID", "FK_Cilveks_ID") REFERENCES
"Pasniedzeji" ("Pasniedzeja_ID", "Studenta_ID", "Studiju_programmas_ID", "Nodalas_ID", "Fakultates_ID", "Cilveks_ID",
"Cilveks_ID")
/

```

SUBJEKTĪVS NOVĒRTĒJUMS

Darba autori, praktiskās datu bāzes modeļu izstrādes laikā izvirzīja subjektīvu vērtējumu par rīka lietošanas iespējām, kas apkopots tabulā.

Pozitīvi	Negatīvi
Iespēja salīdzināt modeļus	Ierobežota LER funkcionalitāte
Atskaišu ģenerēšana	Tikai 2 notācijas
Rīka dokumentācija	Pēc automātiskā izkārtojuma modelis nav pārskatāms
Versiju vadība	Strādā lēni
Transformācija LER -> PER; PER -> LER	Licences izmaksas – 450 €
Kodu ģenerēšana	Ierobežota trial versijas funkcionalitāte
Atbalsta daudz DBVS	
Apvērsta inženierija	
Intuitīva lietotāja saskarne	

IZMANTOTĀ LITERATŪRA

1. <https://shop.quest.com/682/?scope=checkout&id=x1ZX7jpb8e>
2. J. Eiduks, 2012./2013. māc. g. Izdales materiāli no diska.
3. *Toad Data Modeler* Help.