

Ministerul Educației al Republicii Moldova
Universitatea Tehnică a Moldovei
Facultatea Calculatoare Informatică și Microelectronică
Catedra Automatică și Tehnologii Informaționale

RAPORT

Lucrare de laborator nr 5

Disciplina: Proiectarea Sistemelor Informaționale

Tema: Elaborarea modelului logic al Domeniului Obiectiv

A efectuat:

Vovc Artemie st. TI-133

A verificat:

Cojocarui Svetlana lector universitar

Chișinău 2016

Cuprins

Sarcina.....	3
1 Bazele aplicației AllFusion Erwin Data Modeller	4
2 Elaborarea domeniului logic	9
3 Modelarea sistemului	12
Concluzia	17
Bibliografia	18

Sarcina

Elaborarea modelului logic al Domeniului Obiectiv ales în notația IDEF1X.

În această lucrare trebuie de elaborate modelul logic al Domeniului Obiectiv în notația IDEF1X cu instrumentul CASE - ERwin Data Modeler, schema logică a datelor Domeniului Obiectiv – Business-procesele care au fost modelate în Lucrările de Laborator precedente.

1 Bazele aplicației AllFusion Erwin Data Modeller

CA ERwin Data Modeler (ERwin) – instrument din tehnologiile CASE- instrument pentru proiectarea Bazelor de Date, instrument care permite de a crea, documenta și monitoriza Baze de Date, repozitorii și galerii de date.

Lucrul cu aplicația se începe cu crearea unui model nou, pentru care trebuie de selectat tipul modelului (Logical; Physical; sau Logical/Physical) și SGBD -ul preferat din ferestruica ”Database” și versiunea ei ”Version” (figura 1.1).

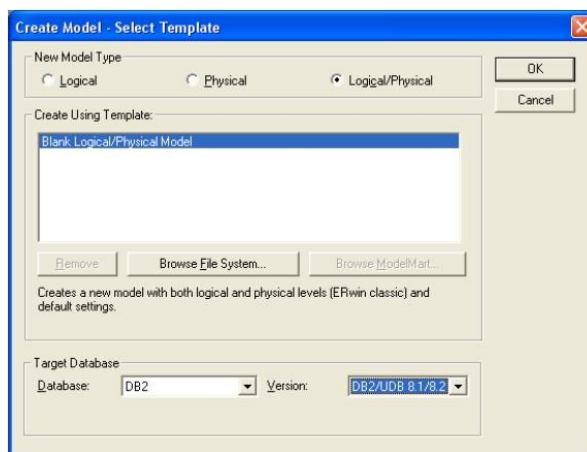


Figura 1.1 – Crearea unui Model nou

ERwin ne permite de a crea model la nivelul logic, sau model la nivelul fizic sau modelul combinat (Logical; Physical; sau Logical/Physical).

Modelul de nivel logic – este o abordare abstractă a datelor (informației) în acest model datele sunt prezentate așa cum sunt în realitate, și pot fi numite cu denumirea lor reală (”Furnizor”, ”Client”, ”Secție”, ”Comandă”).

Obiectele de pe acest model se numesc ”entități” și ”atribute”. Modelul logic al datelor este universal și nu este legat de aplicarea unei SGBD reale.

Modelul de nivel fizic - depinde de SGBD-ul concret selectat. În modelul fizic se conțin informații despre toate subiectele Bazei de Date. modelul fizic depinde de SGBD-ul ales în care noi vrem să creăm Baza de Date.

La nivelul logic ERwin suportă notațiile IE și IDEF1X, la nivelul fizic susține trei notații- IE, IDEF1X și DM. În continuare noi v-om modela lucrarea noastră în notația IDEF1X.

Trecerea de la modelul logic la modelul fizic se efectuează prin ferestruica de dialog din bara de instrumente (figura 1.2).

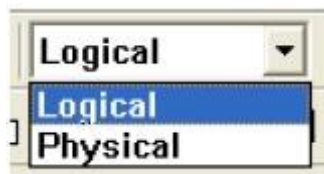


Figura 1.2 – Trecerea de la modelul logic la modelul fizic

Modelul de nivel logic

Pentru a construi un model la nivelul logic utilizăm bara de instrumente *Toolbox* cu instrumentele din această bară noi construim ”entități” și ”conexiunile” dintre ele (Figura 1.3, tabelul 1.1)

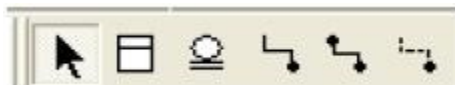


Figura 1.3 – Bara Toolbox

Tabelul 1.1 – Imaginea și funcția iconiței

Imaginea iconiței	Funcția iconiței
	Crearea unei Entități noi. Pentru a crea o entitate nouă butonăm pe iconiță și apoi pe un locul liber pe câmpul modelului.
	Crearea categoriilor. Pentru a rela o conexiune categorizată butonăm Iconița, apoi butonăm entitatea ”mamă” și în continuare butonăm entitatea – fiică.
	Crearea unei conexiuni ”identificatoare”. Pentru a conecta două entități cu o conexiune ”identificatoare” butonăm Iconița, apoi butonăm entitatea ”mamă” și în continuare butonăm entitatea – fiică.
	Creare conexiune ”mai mulți la mai mulți”.
	Crearea unei conexiuni ”neidentificatoare”.

După ce am construit entitățile creăm ”atributele ” pentru fiecare entitate. Pentru aceasta sunt două posibilități: dublu clic pe entitate sau, în meniul de context selectăm punctul *Attributes* (Figura 1.4).

În fereastra atributelor entității avem posibilitate de a vizualiza și redacta informația despre atributele create și de a crea noi atribute. Tot aici indicăm care atribut are statutul de cheie primară. Pentru a crea un atribut nou activăm butonul ” *New*”. În fereastra de dialog ce a apărut (Figura 1.5) putem: - selecta tipul atributului (BLOB, data/ora, număr, rând), - ai da un nume atributului (*Attribute Name*) și numele coloniței (Column Name), care va reprezenta atributul la nivelul fizic (Figura 1.5).

După ce construim entitățile efectuăm conexiunile între ele. La crearea unei conexiuni ”idendificatoare” attributele ce au statut de *cheie primară* a entității ”mamă” migrează în componența cheiei primare a entității ”fiică”, iar la Crearea unei conexiuni ”neidendificatoare” migrează, pur și simplu, în componența atributelor entității ”fiică”.

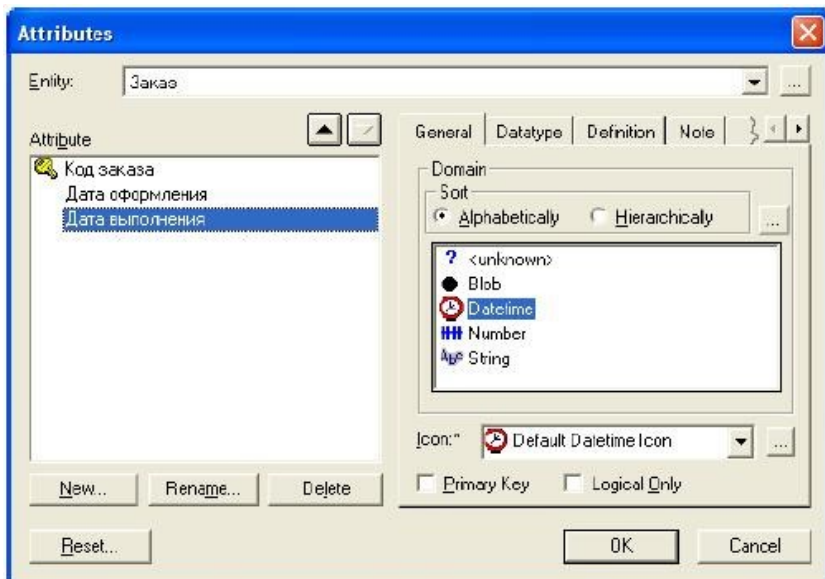


Figura 1.4 – Ferestra a atributelor entității selectate

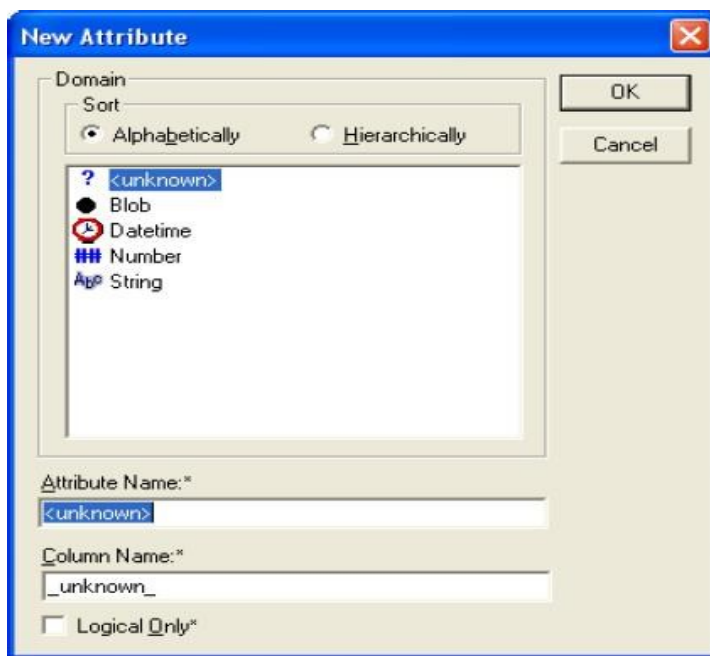


Figura 1.5 – Creare atribut

La determinarea statutului unei conexiuni, sau de ai modifica statutul trebuie de făcut dublu click pe conexiune sau de ales din meniul de context punctul *Relationship Properties* (figura 1.6). Aici în registrul opțiunilor dacă selectăm opțiunea *General* putem atribui numele conexiunii (în direcția ”mamă – fiică” și

direcția ”fiică – mamă”), nivelul puteri conexiunii (zero, una sau mai multe; una și mai multe (P); zero sau una (Z); fixă (un număr concret), putem schimba tipul conexiunii.

Tot aici în opțiunea *RI Action* putem atribui ” limita integrității”.

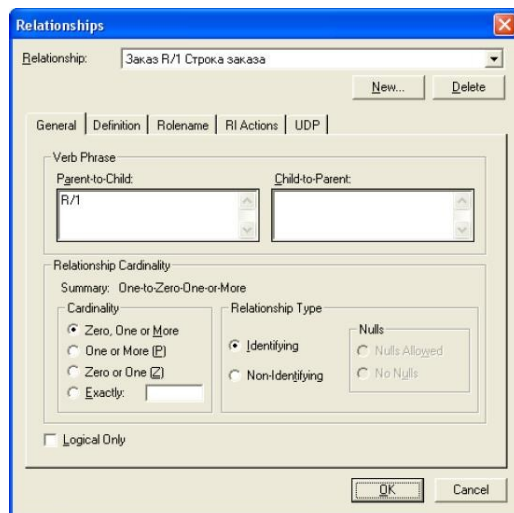


Figura 1.6 – Fereastra ”Relationship”

Exemplu a unui model logic a Bazei de Date este reprezentat în figura 1.7.

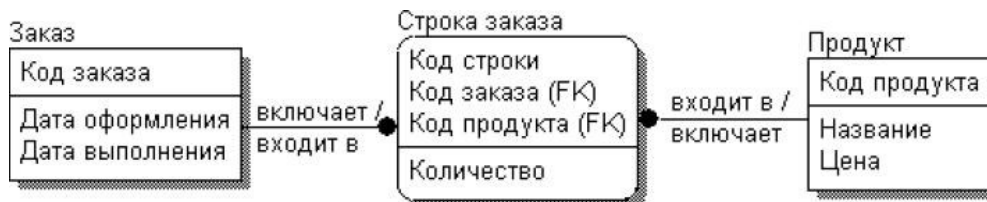


Figura 1.7 – Exemplul unei scheme logice a BD

Nivelul fizic al modelului de date

Dacă trecem de la nivelul logic la nivelul fizic atunci în mod automat se va crea schema fizică a Bazei de Date (figura 1.8)

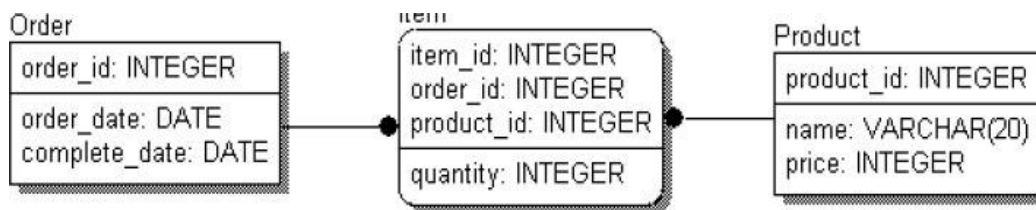


Figura 1.8 – Schema fizică a Bazei de Date create în mod automat

Ea poate fi completată, redactată sau a fi modificată. Principiul de lucru la crearea schemei fizice este analogic cu principiul de lucru la crearea schemei logice.

După ce am creat schema fizică a BD putem genera scripturile pentru SGBD - ul ales. Pentru aceasta utilizăm punctul meniu *Tools -> Forward Engineering/Schema Generation* (Figura 1.9).

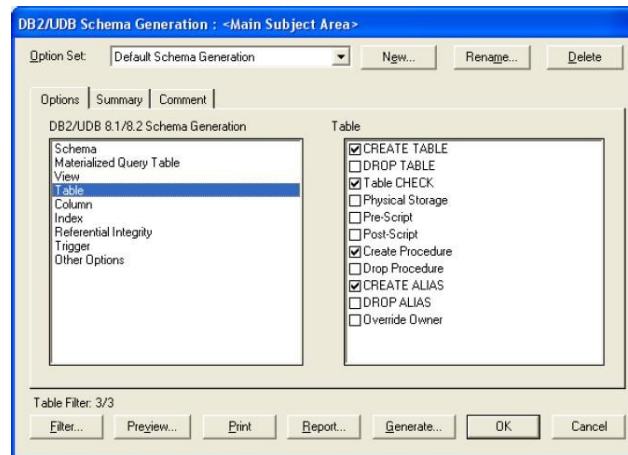


Figura 1.9 – Fereastra pentru a genera SQL-scripturi pentru un SGBD predestinat

Aici putem indica care scripturi trebuie de generat, de previzualiza scripturile și de ai genera (ERwin va efectua conexiune cu SGBD-ul selectat și în mod automat va genera SQL-scripturi).

2 Elaborarea domeniului logic

IDEF1X este bazat pe abordarea lui Chen și permite modelarea datelor ce corespunde Bazei de Date relaționale în a treia formă normală. Notăția Chen și procesul de construire a diagramelor a fost studiat în cursul ”organizarea Bazelor de Date” și noi în cadrul acestei lucrări vom studia numai diferențele dintre ele.

Entitate (Entity) – obiect real sau imaginar ce are o însemnată importanță pentru Domeniul Obiectiv. Fiecare Entitate trebuie să aibă o denumire, exprimată printr-un substantiv la singular. Fiecare Entitate trebuie să posede un identificator unic. Fiecare Entitate trebuie să se identifice univoc și să se deosebească de alte entități de tipul dat.

Atribut (Attribute) – oricăre caracteristică a entității, este importantă pentru Domeniul Obiectiv cercetat și ea servește pentru: calificare, identificare, clasificare, caracteristica cantitativă sau exprimarea stării entității. Denumirea atributului trebuie să fie exprimată printr-un substantiv la singular.

Conexiune (relație) (Relationship) – asociația nominativă între două entități al Domeniul Obiectiv cercetat.

În notația IDEF1X entitățile se împart în trei categorii dependente, independente de identificatori sau simplu independentă, dacă fiecare exemplar poate fi univoc identificată fără a determina relațiile cu alte entități.

Entitatea independentă grafic este reprezentată printr-un dreptunghi obișnuit, pe când Entitatea dependentă grafic este reprezentată printr-un dreptunghi cu colțurile rotunjite.

În notația IDEF1X există următoarele puteri de conexiuni (relații):

- puterea N - fiecare exemplar al entității - parentală poate avea 0 (zero), unu sau mai mult de cât un exemplar interconectat cu un exemplar entitate-moștenită (fără a specifica);

- puterea P - fiecare exemplar de entitate - parentală trebuie să aibă nu mai puțin de un exemplar de entitate-moștenită conectat cu exemplarul entității – parentală;

- puterea Z - fiecare exemplar de entitate - parentală trebuie să aibă nu mai mult de un exemplar entitate-moștenită conectată cu entitate – parentală;

- un număr concret - fiecare exemplar de entitate - parentală trebuie să fie conectat cu un număr fix de entitate-moștenită.

În ERwin la crearea conexiunii identificatoare attributele cheii primare a entității parentale în mod automat se transferă în componenta cheii primare a entității-moștenită. Aceasta se numește migrația atributelor. În entitate-moștenită aceste attribute se identifică ca cheie externă (FK). La crearea conexiunii

neidentificatoare atributelor cheii primare a entității- parentale în mod automat se transferă în câmpul atributelor non-cheie a entității-moștenită.

Elaborarea Modelului logic al datelor a companiei de asamblare și realizarea calculatoarelor PC și a notebook-uri.

În cazul acestei companii vom elabora modelul logic al procesului de ”asamblare calculatoare”.

Pentru a crea ”Modelul logic al datelor” (figura 6.1) trebuie să parcurgem următorii pași.

Pașii pentru crearea modelului logic:

- a) identificarea cerințelor de business;
- b) analiza cerințelor de business;
- c) crearea modelului conceptual al datelor, aprobarea lui de către reprezentanții companiei;
- d) crearea noului model logic de date care include următoarele:
 - 1) selectarea bazei de date țintă (pentru generare scripturi, pentru schema fizică);
 - 2) crearea unui document cu abrevieri standard pentru obiectele logice/fizice;
 - 3) crearea domeniilor;
 - 4) crearea regulilor;
 - 5) crearea valorilor implicite;
 - 6) crearea entităților și adaugarea definiții;
 - 7) asignarea tipurilor de date/domeniilor pt attribute;
 - 8) adăugarea de restricții check/reguli sau valori implicite;
 - 9) crearea de chei primare sau unice;
 - 10) crearea indecșilor;
 - 11) dacă e necesara, crearea subtipurilor și supertipurilor (moștenire);
 - 12) identificarea relațiilor între entități și crearea cheilor externe;
 - 13) validarea modelului de date;
 - 14) aprobarea modelului logic.

Crearea entităților și adaugarea definiții

Pentru a crea modelul logic trebuie să cream entitățile obiectului cercetat. Pentru domeniul cercetat vom identifica următoarele entități:

- client - persoană, ce procură calculatoare;
- comandă – lista calculatoarelor care sunt procurate de client;
- calculator;

- ansamble - elemente, din care sunt asamblate calculatoarele;
- colaborator – specialistul companiei, ce assemblează un calculator concret.

Să identificăm care sunt relațiile dintre aceste entități:

- client – comandă, un client poate plasa mai multe comenzi, trebuie de menționat că dacă un client este în baza de date atunci el a plasat măcar o comandă, asta ne arată că puterea conexiunii este - P, Conexiune identificatoare, de oarece comandă fără client nu poate exista;
- comandă – calculator, în cadrul unei comenzi clientul poate comanda mai multe calculatoare, dar ca minimum în comandă trebuie să fie măcar un calculator, asta ne arată că puterea conexiunii este - P. Conexiune identificatoare, de oare ce calculator fără comandă nu poate exista;
- calculator – ansamble, într-un calculator sunt mai multe componente diferite, unul și același tip de componente poate fi parte a diferitor calculatoare, puterea conexiunii este mulți la – mulți, în IDEF1X astfel tip de conexiune nu există pentru a rezolva situația, introducem noțiunea de entitate asociativă – Configurația;
- putere conexiunii între entitățile Calculator și Configurația este P, deoarece orișice Calculator obligatoriu are o Configurație, puterea conexiunii între entitățile Ansamble și Configurația este - N, putem avea cazul că unele componente încă nu sunt instalate în nici un calculator, conexiunea în ambele cazuri este identificatoare pentru că nu poate exista Configurația calculatorului fără legătura directă cu calculatorul și cu ansamble;
- ansamble – tip de elemente, este evident că numărul de tipuri de ansamble care pot fi instalate este limitat dar sunt des utilizate, vom introduce o entitate nouă - tipul de ansamble, puterea conexiunii este - P, conexiunea este identificatoare;
- calculator – colaborator, fiecare Calculator este asamblat de un specialist, unii dintre ei pot asambla mai multe Calculatoare, puterea Conexiunii – N, tipul conexiunii este neidentificatoare, pentru că exemplarul entității Calculator deja poate exista dar el nu este încă legat de nici un specialist, anume din aceste considerente în proprietățile acestei conexiuni noi am ales comutatorul "Nulls Allowed" (pe diagramă un romb din partea entității parentale specialist.

3 Modelarea sistemului

Mai jos este reprezentat în figura 3.1 începutul lucrului cu mediu ERwin DM. Se vede o entitate numită Utilizator pe suprafața de lucru și procesul de adăugare a noilor atribute în entitate.

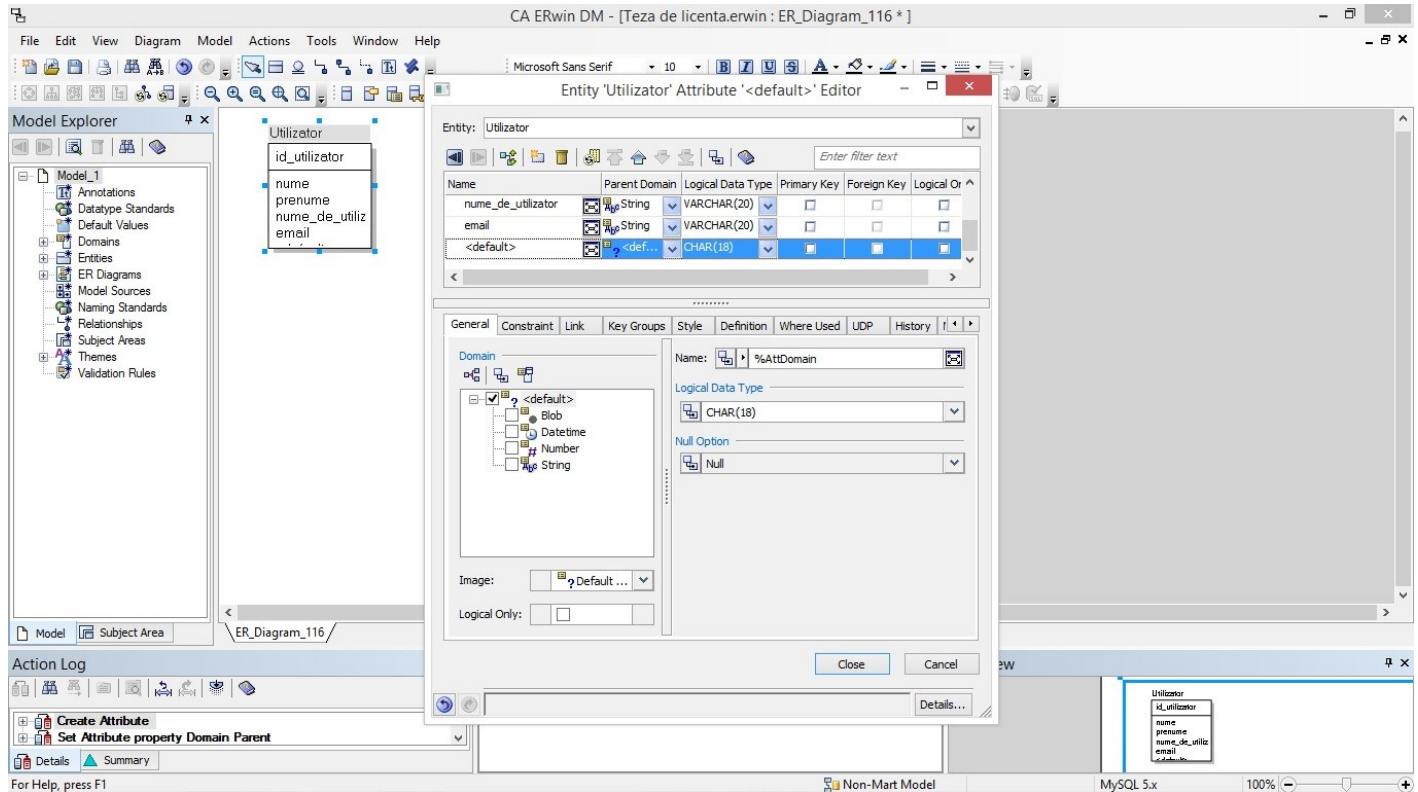


Figura 3.1 – Erwin DM

Când creem un nou atribut el are tipul default noi trebuie să-l determinăm cu un tip din cele patru: Blob, Datetime, Number, String. Acestea patru tipuri mai au și subtipuri ca de exemplu pentru number avem Byte, Integer și etc. Putem determina atribut ca fiind vid sau nevid. Dacă definim tipul primitiv al atributului mai putem și determina lungimea acestui primitiv de exemplu dacă e String și char atunci vază lungimea de la 0 la maximum pentru caractere. Maximum și minimum depinde de standardele fiecărei SGBD în parte. Pe lângă proprietăți generale putem și seta constrângeri de check etc. Cel mai important este de determinat carei primari key (cheia primară) pentru entitate curentă. Scopul determinării cheii primare este pentru a putea face relații de tipul unu la mai mulți sau mai mulți la mai mulți.

În figura 3.2 este reprezentată baza de date propriu zisă pentru sistemul informațional „SDK rețele neuronale”.

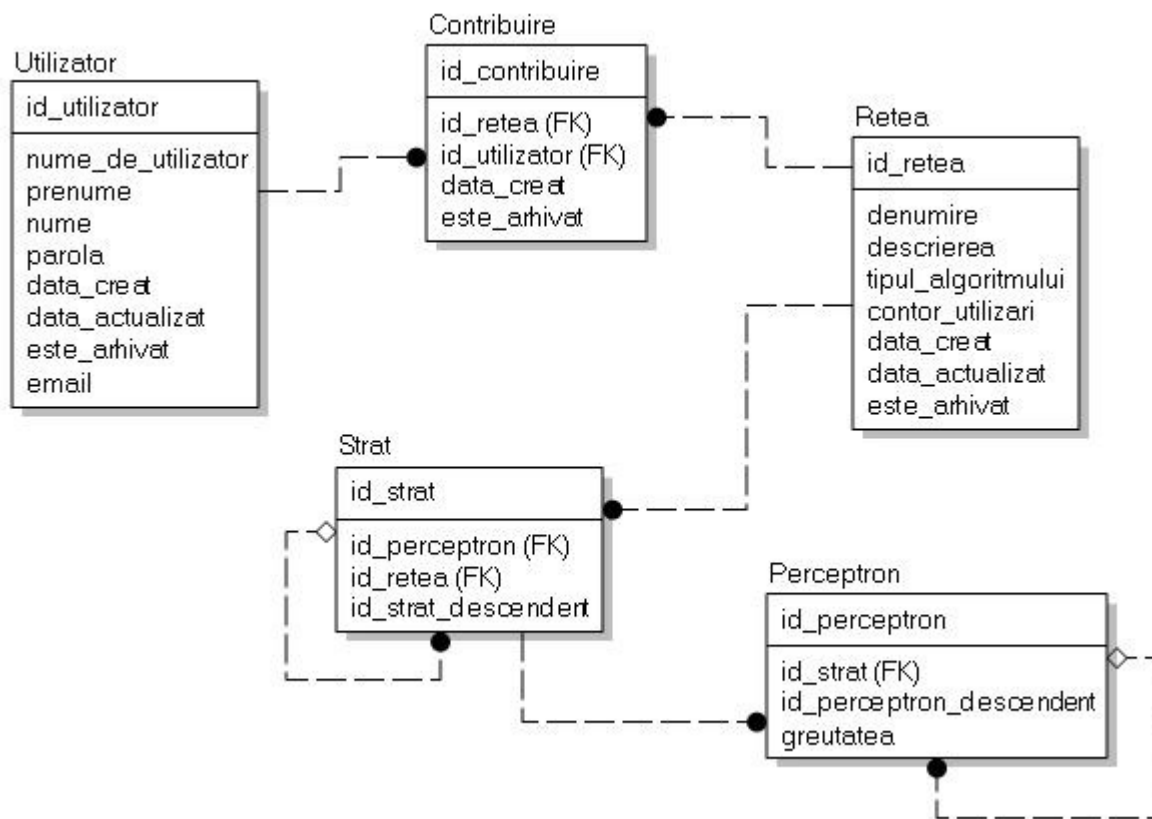


Figura 3.2 – Baza de date

Urmează explicarea relațiilor între entități. Așa cum sistemul informațional va fi prezentă posibilitatea de a contribui la o rețea este nevoie de a uni logic tabelele Utilizator și Rețea cu o relație de mulți la mulți. Pentru a realiza această relație de mulți la mai mulți între cele două entități este necesitatea de a introduce o nouă entitate care e numită contribuie care va avea două cîmpuri chei externe care va conține referință la utilizator și la rețea la care contribuie. Strat este o entitate care descrie nivelurile dintr-o rețea neuronală, în strate pot să se conțină o mulțime de perceptroni, pe lângă acest fapt straturile se conectează și între ele, de aceea este prezent atributul perceptron și strat descendent. În entitatea Perceptron se descrie greutatea fiecărui perceptron în parte și o mulțime de perceptroni descendenți cărui se va transmite un semnal.

Figura 3.3 prezintă o parte din script generat cu ajutorul programei erwin data modeller.

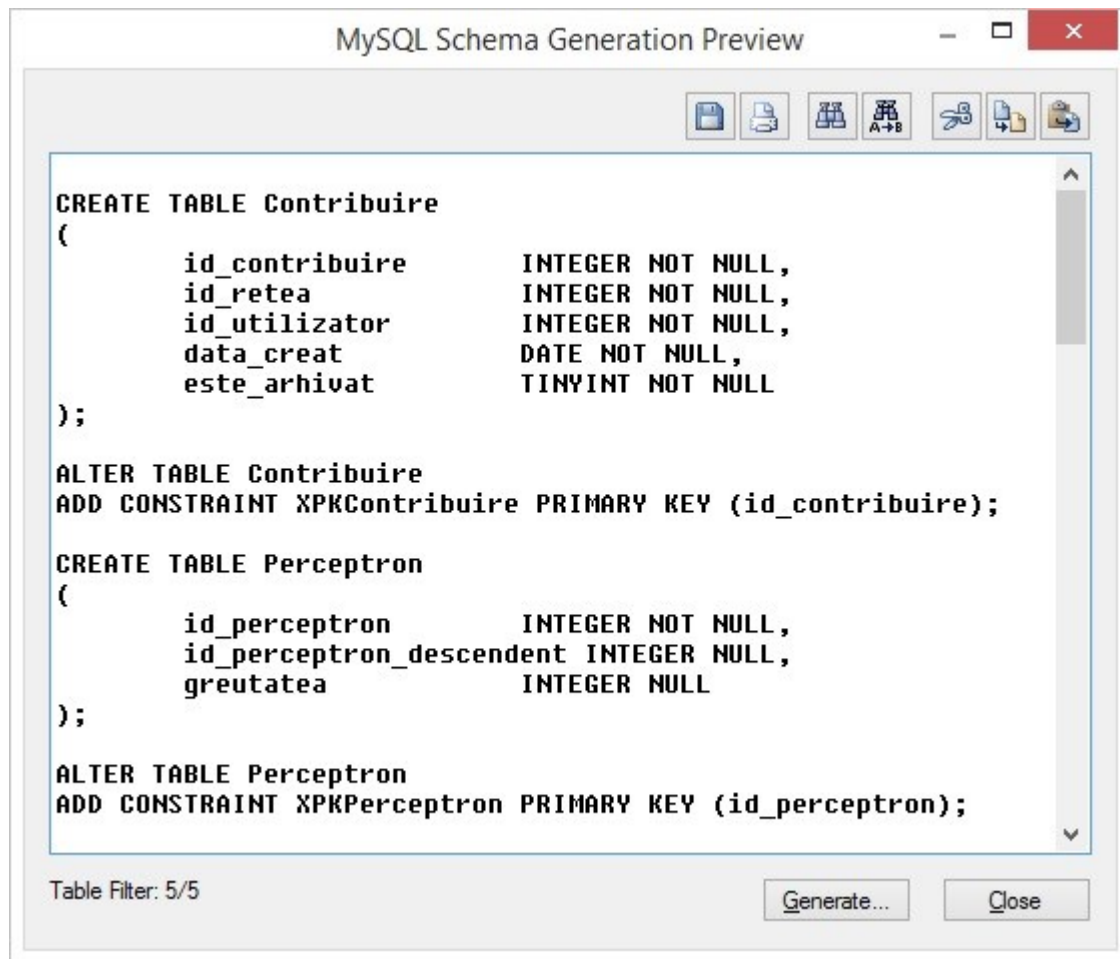


Figura 3.3 – Generarea scriptului

Erwin data modeller permite prin două-trei clicuri generarea unui script valid pentru orice SGBD actual. Vezi mai jos scriptul generat.

Scriptul generat

```
CREATE TABLE Contribuire
(
    id_contribuire      INTEGER NOT NULL,
    id_retea            INTEGER NOT NULL,
    id_utilizator        INTEGER NOT NULL,
    data_creat          DATE NOT NULL,
    este_arhivat         TINYINT NOT NULL
);
```

```
ALTER TABLE Contribuire
ADD CONSTRAINT XPKContribuire PRIMARY KEY (id_contribuire);
```

```
CREATE TABLE Perceptron
(
    id_perceptron          INTEGER NOT NULL,
    id_perceptron_descendent INTEGER NULL,
    greutatea              INTEGER NULL
);
```

```
ALTER TABLE Perceptron
ADD CONSTRAINT XPKPerceptron PRIMARY KEY (id_perceptron);
```

```
CREATE TABLE Retea
(
    id_retea              INTEGER NOT NULL,
    denumire              VARCHAR(20) NOT NULL,
    descrierea            VARCHAR(20) NULL,
    tipul_algoritmului    VARCHAR(20) NOT NULL,
    contor_utilizari      INTEGER NOT NULL,
    data_creat            DATE NOT NULL,
    data_actualizat       DATE NOT NULL,
    este_arhivat          CHAR(18) NOT NULL
);
```

```
ALTER TABLE Retea
ADD CONSTRAINT XPKRetea PRIMARY KEY (id_retea);
```

```
CREATE TABLE Strat
(
    id_strat              INTEGER NOT NULL,
    id_retea              INTEGER NOT NULL,
    id_strat_descendent   INTEGER NULL,
    id_perceptron         INTEGER NOT NULL,
    ld_perceptron         INTEGER NULL
);
```

```
ALTER TABLE Strat
ADD CONSTRAINT XPKStrat PRIMARY KEY (id_strat);
```

```

CREATE TABLE Utilizator
(
    id_utilizator      INTEGER NOT NULL,
    nume               VARCHAR(20) NOT NULL,
    prenume            VARCHAR(20) NULL,
    nume_de_utilizator VARCHAR(50) NOT NULL,
    parola              VARCHAR(20) NOT NULL,
    data_creat          DATE NOT NULL,
    data_actualizat     DATE NULL,
    este_arhivat        TINYINT NOT NULL,
    email               VARCHAR(50) NOT NULL
);

ALTER TABLE Utilizator
ADD CONSTRAINT XPKUtilizator PRIMARY KEY (id_utilizator);

ALTER TABLE Contribuire
ADD CONSTRAINT R_1 FOREIGN KEY (id_utilizator) REFERENCES Utilizator (id_utilizator);

ALTER TABLE Contribuire
ADD CONSTRAINT R_2 FOREIGN KEY (id_retea) REFERENCES Retea (id_retea);

ALTER TABLE Perceptron
ADD CONSTRAINT R_6 FOREIGN KEY (id_perceptron) REFERENCES Perceptron (id_perceptron);

ALTER TABLE Strat
ADD CONSTRAINT R_3 FOREIGN KEY (id_retea) REFERENCES Retea (id_retea);

ALTER TABLE Strat
ADD CONSTRAINT R_4 FOREIGN KEY (id_strat) REFERENCES Strat (id_strat);

ALTER TABLE Strat
ADD CONSTRAINT R_5 FOREIGN KEY (id_perceptron) REFERENCES Perceptron (id_perceptron);

ALTER TABLE Strat
ADD CONSTRAINT R_7 FOREIGN KEY (id_perceptron) REFERENCES Perceptron (id_perceptron);

```


Concluzia

Lucrarea dată a avut ca scop să formeze o viziune despre sistemul erwin data modeller care formează baza de date pentru orice sistem informațional. La general este un instrument de modelare a bazelor de date, care necesită unele configurări în dependență de ce baza de date se va folosi în viitor pentru sistemul informațional. Ca rezultat am generat un script care formează tabelele bazei de date, care nu conține scriptul ce generează însuși identificatorul bazei de date.

Bibliografia

1. Indrumarul metodic al universității tehnice din Moldova. AllFusion process modeler.
2. Resursă electronică bazele programului AllFusion process modeler. [regim de acces]:
<http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/lab05.html>