

Universitatea din București
Facultatea de Matematică și Informatică
Departmentul de Informatică
Master Inteligență Artificială

Gestionarea unei librării de închiriere jocuri
Aplicații profesionale în baze de date orientate pe obiecte
Proiect 2

Andreea-Daniela Ene
grupa 407

București
February 4, 2017

Cuprins

1	Descrierea modelului	5
2	Tratarea orientată pe obiecte	6
3	Crearea tabelelor	7
4	Transformarea modelului	10
5	Interogari	13

1 Descrierea modelului

In acest proiect este propus un model pentru baza de date in care se poate realiza gestionarea unei librării de inchiriere jocuri.

In model, entitatile au fost organizate in patru grupuri, joc, magazin, angajat si client. Joc, grupul principal, contine diferite informatii legate de jocurile disponibile pentru inchiriere, precum tipul, categoria, producatorul, numarul de jucatori recomandati (categorie de varsta), precum si numarul de jocuri disponibile. In grupul client se regasesc informatii legate de acesa si produsele pe care el le-a inchiriat. In cel de-al treilea grup, magazin, sunt informatii despre evenimentele ce se pot desfasura in magazine, precum si departamentele din acestea. In ultimul grup, angajat, se regasesc sunt disponibile informatii despre toti angajatii companiei, pozitia pe care acestia o ocupa, precum si contractul sau nivelul salarial.

Această librărie asigura clientilor sai o gama variata de jocuri, pe care acestia le pot inchiria pentru 30 de zile, din oricare magazin din cele disponibile. Clientii pot returna jocul in oricare dintre locatiile disponibile ale librăriei, si pot participa la diferitele evenimente organizate in cadrul acestora. Fiecare client are asociat un cont in baza de date, fiind neaparat necesar sa transmita datele sale de contact, pentru eventualele intarzieri ale returnarii. Acestia pot de asemenea sa dea o nota intre 1 si 5 tuturor jocurilor pe care le-au inchiriat.

Jocurile disponibile in baza de date sunt stocate impreuna cu informatii referitoare la tipurile, categoriile, producatorii, categoriile de varsta, dar si notele oferite de clienti. Jocurile fac parte dintr-un singur tip, categorie, sau producator, insa pot exista mai multe jocuri din fiecare dintre acestea. In magazine, pot fi organizate diferite evenimente pentru clienti, in care acestia pot participa la competitii de jocuri de societate si testa jocurile lor preferate. In magazine exista diferite departamente care se ocupa de logistica, inchiriere jocuri, sau achizitie. Toti angajatii au asignat cate un contract unic, si au un pachet salarial care face parte dintr-o clasa salariala. Acestia ocupa anumite pozitii in departamentele din care fac parte.

2 Tratarea orientată pe obiecte

Din tabelele folosite în proiectul precedent am ales să folosesc următoarele tabele:

- game
- game_review
- client
- client_loan
- loan_products
- game_is_in_store
- store

3 Crearea tabelelor

Tabela **game**

```
CREATE TABLE "game" (  
    "game_id"          INT NOT NULL,  
    "name"             VARCHAR(128) NOT NULL,  
    "thumbnail"        VARCHAR(256) NOT NULL,  
    "year_published"   SMALLINT NOT NULL,  
    "description"      VARCHAR(500) NOT NULL,  
    "category_id"      INT NOT NULL,  
    "producer_id"      INT NOT NULL,  
    "type_id"          INT NOT NULL,  
    "players_id"       INT NOT NULL,  
    PRIMARY KEY ("game_id"),  
    CONSTRAINT "category_id"  
        FOREIGN KEY ("category_id")  
        REFERENCES "category" ("category_id")  
        ENABLE,  
    CONSTRAINT "producer_id"  
        FOREIGN KEY ("producer_id")  
        REFERENCES "producer" ("producer_id")  
        ENABLE,  
    CONSTRAINT "type_id"  
        FOREIGN KEY ("type_id")  
        REFERENCES "type" ("type_id")  
        ENABLE,  
    CONSTRAINT "players_id"  
        FOREIGN KEY ("players_id")  
        REFERENCES "players" ("players_id")  
        ENABLE);
```

Tabela **game_review**

```
CREATE TABLE "game_review" (  
  "review_id"      INT NOT NULL,  
  "grade"          NUMBER(1) NOT NULL,  
  "client_id"      INT NOT NULL,  
  "game_id"        INT NOT NULL,  
  PRIMARY KEY ("review_id"));
```

Tabela **client**

```
CREATE TABLE "client" (  
  "client_id"      INT NOT NULL,  
  "name"           VARCHAR(45) NOT NULL,  
  "phone"          NUMBER(10) NOT NULL,  
  "e-mail"         VARCHAR(20) NOT NULL,  
  "address"        VARCHAR(50) NULL,  
  PRIMARY KEY ("client_id"));
```

Tabela **client_loan**

```
CREATE TABLE "client\_loan" (  
  "client\_loan_id" INT NOT NULL,  
  "start_date"      DATE NOT NULL,  
  "end_date"        DATE NOT NULL,  
  "returned_date"   DATE NULL,  
  "client_id"       INT NOT NULL,  
  PRIMARY KEY ("bill_id"));
```

Tabela **loan_products**

```
CREATE TABLE "loan_products" (  
  "loan_products_id" INT NOT NULL,  
  "game_is_in_store" INT NOT NULL,  
  "client_loan_id"   INT NOT NULL,  
  PRIMARY KEY ("loan_products_id"));
```


Tabela `game_is_in_store`

```
CREATE TABLE "game\_is\_in\_store" (  
  "game_id"          INT NOT NULL,  
  "store_id"         INT NOT NULL,  
  "is_available"     NUMBER(1) DEFAULT 1 NOT NULL,,  
  PRIMARY KEY ("game\_id", "store\_id")),
```

Tabela `store`

```
CREATE TABLE "store" (  
  "store_id"         INT NOT NULL,  
  "name"             VARCHAR(64) NOT NULL,  
  "address"          VARCHAR(45) NOT NULL,  
  PRIMARY KEY ("store_id"));
```

Drop Tables Script

```
drop table game          cascade constraints purge;  
drop table game_review   cascade constraints purge;  
drop table client        cascade constraints purge;  
drop table client_loan   cascade constraints purge;  
drop table loan_products cascade constraints purge;  
drop table game_is_in_store cascade constraints purge;  
drop table store         cascade constraints purge;
```

4 Transformarea modelului

```
CREATE TYPE T_game as object (  
    "game_id"          INT,  
    "name"             VARCHAR(128),  
    "thumbnail"        VARCHAR(256),  
    "year_published"   SMALLINT,  
    "description"       VARCHAR(500),  
    "category_id"      INT,  
    "producer_id"      INT,  
    "type_id"          INT,  
    "players_id"       INT);  
  
CREATE TYPE T_game_review as object (  
    "review_id"        INT,  
    "grade"            NUMBER(1),  
    "client_id"        INT,  
    "game_id"          INT);  
  
CREATE TYPE T_client as object (  
    "client_id" INT,  
    "name"      VARCHAR(45),  
    "phone"     NUMBER(10),  
    "e-mail"    VARCHAR(20),  
    "address"   VARCHAR(50));  
  
CREATE TYPE T_client_loan as object (  
    "client\_loan_id" INT,  
    "start_date"     DATE,  
    "end_date"       DATE,
```

```
"returned_date"    DATE,  
"client_id"        INT);
```

```
CREATE TYPE T_loan_products as object (  
    "loan_products_id" INT,  
    "game_is_in_store" INT,  
    "client_loan_id"   INT);
```

```
CREATE TYPE T_game_is_in_store as object (  
    "game_id"        INT,  
    "store_id"       INT,  
    "is_available"   NUMBER(1) DEFAULT 1);
```

```
CREATE TYPE T_store as object (  
    "store_id"       INT,  
    "name"           VARCHAR(64),  
    "address"        VARCHAR(45));
```

Vizualizările pot fi create pornind de la tabelele de tip obiect create anterior. Cererea prin care acestea vor fi create va reprezenta extragerea coloanelor din tabelele de la care am pornit. Cheia primară pe care o vom folosi pentru identificarea obiectelor va fi cheia primară a tabelelor asociate, astfel voi folosi opțiunea WITH OBJECT IDENTIFIER (identificator) atunci când vizualizările vor fi create.

1. Vizualizare ce contine informatii despre toti clientii

```
CREATE VIEW clients_info of T_client WITH OBJECT IDENTIFIER(client_id) AS
SELECT
    c."name"      as "Nume",
    c."phone"     as "Numar telefon",
    c."e-mail"    as "Adresa e-mail",
    c."address"   as "Adresa"
FROM
    "client" c;
```

2. Vizualizare ce contine informatii despre notele jocurilor si de cine au fost oferite.

```
CREATE VIEW reviewed_games of T_client WITH OBJECT IDENTIFIER(client_id) AS
SELECT
    g."name"      as "Denumire joc",
    r."grade"     as "Nota",
    c."name"      as "Reviewer"
FROM
    "game" g, "game_review" r, "client" c
WHERE
    c."client_id" = r."client_id" AND
    r."game_id"   = g."game_id";
```

3. Vizualizare cu jocurile care sunt disponibile in magazine, si locul de unde se pot inchiria.

```
CREATE VIEW available_games of T_client WITH OBJECT IDENTIFIER(game_id) AS
SELECT
    g."name"      as "Denumire joc",
    s."name"      as "Nume magazin"
FROM
    "game" g, "store" s, "game_is_in_store" gs
WHERE
    gs."game_id"   = g."game_id" AND
    gs."store_id"  = s."store_id" AND
    gs."is_available" = 0;
```

5 Interogari

1. Afisarea mediei tuturor jocurilor

```
SELECT
    g."name"                as "Denumire joc",
    ROUND(AVG(r."grade"), 1) as "Nota",
    COUNT(r."grade")        as "Numar de note"
FROM "game" g, "game_review" r
WHERE
    r."game_id" = g."game_id"
GROUP BY g."name"
ORDER BY AVG(r."grade") DESC;
```

2. Persoana cu cele mai multe review-uri

```
SELECT
    c."name"                as "Nume client",
    COUNT(r."grade")        as "Numar",
    c."client_id"           as "ID"
FROM "game" g, "game_review" r, "client" c
WHERE
    r."game_id" = g."game_id" AND
    r."client_id" = c."client_id"
GROUP BY c."name"
ORDER BY COUNT(r."grade") DESC;
```

3. Jocul cu cele mai multe review-uri

```
SELECT
    g."name"                as "Denumire joc",
    COUNT(r."grade")        as "Numar de note"
FROM "game" g, "review" r
WHERE
    r."game_id" = g."game_id"
GROUP BY g."name"
ORDER BY COUNT(r."grade") DESC;
```

4. Jocurile ordonate in functie de anul publicarii

```
SELECT
    g."name"                as "Denumire joc",
    g."year_published"      as "Anul publicarii"
FROM "game" g
ORDER BY g."year_published" DESC;
```

5. Afisarea tuturor jocurilor inchiriate

```
SELECT
    g."name"                as "Denumire joc",
    g."game_id"             as "ID joc",
    c."name"                as "Nume client",
    c."client_id"           as "ID client"
FROM "game" g, "client" c, "client_loan" cl, "loan_products" lp, "game_is_in_store" gs
WHERE
    cl."returned_date"      is null AND
    cl."client_id"          = c."client_id" AND
    lp."client_loan_id"     = cl."client_loan_id" AND
    lp."game_is_in_store_id" = gs."game_id" AND
    gs."game_id"            = g."game_id"
ORDER BY c."name";
```

6. Afisarea tuturor jocurilor ce nu sunt inca disponibile pentru inchiriere. (Exista in baza de date, dar nu au ajuns inca in magazine)

```
SELECT
    g."name" as "Nume joc",
    count(g."game_id") as "Numar jocuri"
FROM "game" g
FULL JOIN "game_is_in_store" gs ON(g."game_id" = gs."game_id")
WHERE g."game_id" is null or gs."game_id" is NULL
GROUP BY g."name";
```

7. Afisarea tuturor jocurilor disponibile, ordonate in functie de anul publicarii.

```
SELECT UNIQUE
    g."name" as "Denumire joc",
    g."year_published" as "Anul publicarii"
FROM "game" g
ORDER BY g."year_published" DESC;
```

8. Afisarea celor mai populare jocuri, in ordine crescatoare, in functie de numarul de note primite.

```
SELECT
    g."name" as "Denumire joc",
    COUNT(r."grade") as "Numar de note"
FROM "game" g, "game_review" r
WHERE
    r."game_id" = g."game_id"
GROUP BY g."name"
ORDER BY COUNT(r."grade") DESC;
```

9. Afisarea jocurilor cu notele cele mai mari, ordonate crescator.

```
SELECT
    g."name" as "Denumire joc",
    ROUND(AVG(r."grade"), 1) as "Nota",
    COUNT(r."grade") as "Numar de note"
FROM "game" g, "game_review" r
WHERE
    r."game_id" = g."game_id"
GROUP BY g."name"
ORDER BY AVG(r."grade") DESC;
```