**Universitatea Tehnică a Moldovei**

**Facultatea de Calculatoare, Informatică și Microeletronică**

**Departamentul Ingineria Software și Automatică**

Proiect de curs

**Disciplina: Baze de date**

**Tema:** *Elaborarea bazei de date pentru o magazin de carti*

**A efectuat: st.gr.TI-194, Zavorot Daniel**

**A verificat: asist. univ. Răducanu Octavian**

**Chișinău - 2021**

***Cuprins***

* 1. Noțiuni teoretice..........................................................................................3
  + 1.1. Limbajul de definire a datelor...........................................................6
  + 1.2. Microsoft SQL Server.......................................................................7
* 2. Definirea temei proiectului și conținutul problemei...................................8
* 3. Proiectarea modelului relațional al bazei de date........................................9
  + 3.1. Crearea și întreținerea Bazei de Date...............................................10
  + 3.2. Crearea și modificarea tabelelor......................................................13
  + 3.3. Diagrame, scheme, sinonime. Inserarea valori în baza de date.......13
  + 3.4. Crearea și gestiunea interogărilor SQL............................................17
  + 3.5. Instrucțiuni procedurale....................................................................22
  + 3.6. Viziuni..............................................................................................23
  + 3.7. Sinonime...........................................................................................24
  + 3.8. Proceduri stocate și funcții definite..................................................25
  + 3.9. Declanșatoare...................................................................................27
  + 3.10. Recuperarea bazei de date................................................................29
  + 3.11. Exportarea și importarea datelor......................................................30
  + 3.12. Rapoarte...........................................................................................33
* Concluzie.............................................................................................................34
* Surse bibliograficef..............................................................................................35
* Anexa...................................................................................................................36

# INTRODUCERE

## Noțiuni teoretice

O bază de date, uneori numită și bancă de date (abreviat BD), reprezintă o modalitate de stocare a unor [informații](https://ro.wikipedia.org/wiki/Informa%C8%9Bie) și [date](https://ro.wikipedia.org/wiki/Dat%C4%83) pe un suport extern (un dispozitiv de stocare), cu posibilitatea extinderii ușoare și a regăsirii rapide a acestora. La prima vedere sarcina poate părea banală. Totuși, în condițiile în care este vorba de a lucra cu milioane de elemente, fiecare putând consta din cantități de date care trebuie accesate simultan prin [Internet](https://ro.wikipedia.org/wiki/Internet) de către mii de utilizatori răspândiți pe întreg globul; și în condițiile când disponibilitatea aplicației și datelor trebuie să fie permanentă, soluțiile bune nu sunt de loc simple. De obicei o bază de date este memorată într-unul sau mai multe [fișiere](https://ro.wikipedia.org/wiki/Fi%C8%99ier). Bazele de date sunt manipulate cu ajutorul [sistemelor de](https://ro.wikipedia.org/wiki/Sistem_de_gestiune_a_bazelor_de_date) [gestiune a bazelor de date](https://ro.wikipedia.org/wiki/Sistem_de_gestiune_a_bazelor_de_date). Cel mai răspândit tip de baze de date este cel relațional, în care datele sunt memorate în [tabele](https://ro.wikipedia.org/wiki/Tabel). Pe lânga tabele, o bază de date relațională mai poate conține: indecși, proceduri stocate, declanșatori, utilizatori și grupuri de utilizatori, tipuri de date, mecanisme de securitate și de gestiune a tranzacțiilor etc. Alte tipuri de baze de date sunt modelul ierarhic, modelul orientat pe obiecte și, mai nou, modelul [XML.](https://ro.wikipedia.org/wiki/XML)

SQL (de la numele englez Structured Query Language - limbaj de interogare structurat) este un limbaj de programare specific pentru manipularea datelor în sistemele de manipulare a bazelor de date relaționale (RDBMS), iar la origine este un limbaj bazat pe algebra relațională. Acesta are ca scop inserarea datelor, interogații, actualizare și ștergere, modificarea și crearea schemelor, precum și controlul accesului la date. A devenit un standard în domeniu, fiind cel mai popular limbaj utilizat pentru crearea, modificarea, regăsirea și manipularea datelor de către SGBD-urile (Sistemele de Gestiune a Bazelor de Date) relaționale. Pe lângă versiunile standardizate ale limbajului, există o mulțime de dialecte și variante, unele proprietare, fiind specifice anumitor SGBD-uri și de asemenea conținând extensii pentru a suporta SBD-urile (Sistemele de Baze de Date) obiectuale (obiectual-relaționale). SQL permite atât accesul la conținutul bazelor de date, cât și la structura acestora.mSQL a fost inițial dezvoltat la compania [IBM](https://ro.wikipedia.org/wiki/IBM) de către [Donald D. Chamberlin](https://ro.wikipedia.org/w/index.php?title=Donald_D._Chamberlin&action=edit&redlink=1) și [Raymond F. Boyce](https://ro.wikipedia.org/w/index.php?title=Raymond_F._Boyce&action=edit&redlink=1) în anul 1974. Inițial numele era SEQUEL (tot de la *Structured Query Language*) și avea ca scop manipularea și regăsirea datelor stocate în bazele de date relaționale IBM.

Limbajul SQL este divizat în următoarele elemente:

-**Clauze,** care sunt componente ale instrucțiunilor și interogărilor.

-**Expresii,** al căror efect este producerea de valori [scalare](https://ro.wikipedia.org/wiki/Scalar) sau [tabele](https://ro.wikipedia.org/w/index.php?title=Tabele_(baze_de_date)&action=edit&redlink=1).

-**Predicates,** pot specifica condiții care sunt evaluate de SQL conform [logicii](https://ro.wikipedia.org/w/index.php?title=Logica_ternar%C4%83&action=edit&redlink=1) [ternare](https://ro.wikipedia.org/w/index.php?title=Logica_ternar%C4%83&action=edit&redlink=1) sau [logicii booleene](https://ro.wikipedia.org/wiki/Logic%C4%83_boolean%C4%83), în scopul limitării efectelor instrucțiunilor, sau pentru a influența cursul programului.

-**Interogările,** au ca scop regăsirea datelor după criterii specifice.

-**Instrucțiunile,** pot avea un efect persistent asupra datelor sau structurii datelor, sau pot controla tranzacțiile, conexiunile sau cursul programului. În general, instrucțiunile SQL se termină cu caracterul punct-virgulă (";"), deși acest lucru nu este obligatoriu în toate

platformele SQL. Spațiile albe suplimentare sunt ignorate, dar ele pot fi folosite pentru lizibilitatea codului SQL.

Oricărei coloane (sau câmp) dintr-un tabel SQL îi este asignat un *tip de dată*, la fel ca în toate celelalte limbaje de programare. Tipurile de date sunt următoarele:

-**CHARACTER (**sau **CHAR) –** șir de caractere

-**INTEGER** (sau **SMALLINT**) – număr întreg

-**FLOAT, REAL** sau **DOUBLE PRECISION –** număr real

-**NUMERIC** (precision, scale) sau **DECIMAL**(precision, scale) – număr zecimal , unde “precision” înseamnă numărul de cifre din partea întreagă, “scale” înseamnă numărul de zecimale.

-**DATE –** data zilei

-**TIME –** ora. Funcția system **NOW** întoarce data și ora curentă.

Interogarea unei baze de date reprezintă un proces de selecţie care restrânge informaţiile extrase din baza de date la acele înregistrări (numite și tuple) care îndeplinesc criteriile dorite. Cererile de interogare SQL, folosesc în exculsivitate comanda DML SELECT, ce poate fi utilizată atât pentru interogarea obiectelor create de utilizator cât și a obiectelor de la nivelul sistemului de gestiune.

Instrucțiunea SELECT este principalul instrument pentru regăsirea datelor dintr-o bază de date relațională, premițând selectarea de conținut dintr-una sau mai multe tabele ,după diferite criterii ce vor fi specificate cu ajutorul clauzelor (WHERE, GROUP BY, HAVING) ce sunt utilizate în structura instrucțiunii.

Atunci când se utilizează instrucţiunea SELECT, datele returnate pot fi controlate din două perspective. Pe de o parte, se pot selecta datele pe verticală, specificând atributele (coloanele) de interes, iar pe de altă parte, datele pot fi selectate pe orizontală, controlând astfel ce tuple (înregistrări) vor face parte din mulţimea rezultat.

Fraza SELECT , principial descrie patru proprietăţi ce caracterizează ale mulţimea rezultat, astfel :

1. coloanele pe care le va conţine mulţimea rezultat (atributele din tabele) 2.tabelele din care se selectează datele şi legăturile logice dintre ele (FROM)
2. condiţiile pe care trebuie să le îndeplinească liniile din tabele pentru a fi returnate (WHERE)
3. ordinea în care sunt afişate liniile din mulţimea rezultat (GROUP BY, ORDER BY)

O formă simplificată a frazei de interogare ce conține clauzele cele mai uzuale a cărei sintaxă respectă notația BNF (în care elementele opționale sunt marcate prin paranteze drepte, alternativele unui parametru prin bara verticală iar cuvintele rezervate de sistem MySQL referă clauzele și apar cu majuscule), iar ordinea de utilizare a acestora este cea specificată.

Clauzele prezente într-o formulare de interogare a datelor au următoarea semnificație :

* Clauza **WHERE** specifică condiţiile ce trebuie îndeplinite de atributele ale căror valori se doresc a fi afişate. Dacă această clauză lipseşte, sunt selectate toate informaţiile în cauză.
* Clauza **GROUP BY**. În situaţia în care se doreşte gruparea informaţiilor după valorile unui anumit atribut ,această clauză realizează suplimentar şi ordonarea valorilor, ceea ce poate determina o încetinire a vitezei de execuţie. Pentru a forţa motorul bazei de date să nu realizeze şi operaţia de sortare, trebuie adăugată clauza ORDER BY NULL. Parametrii pe care îi poate primi clauza sunt denumiri (respectiv alias-uri) de atribute (coloane) precum şi valori care indică ordinea lor în expresia SELECT.
* Clauza **ORDER BY** sortează conţinutul atributelor pe care le primeşte ca parametri crescător (implicit) sau descrescător, după cum sunt specificate cuvintele-cheie ASC sau DESC. În cazul interogărilor imbricate, dacă se realizează sortări după acelaşi atribut, va prevala opţiunea indicată în cazul interogării exterioare, dacă ordinea specificată este diferită.
* Clauza **HAVING** este folosită ,de regulă pentru atribute specificate în clauza GROUP BY sau pentru cele utilizate împreună cu funcţii agregate.

Limbajul de manipulare a datelor (DML) este un subset al SQL utilizat pentru a adăuga, actualize sau șterge date. Instrucțiunile limbajului de manipulare a datelor:

-**INSERT –** inserează un articol într-o tabelă

INSERT INTO table (camp1, camp2, . . .) VALUES (valoare1, valoare2, . .);

-**UPDATE –** actualizează un set de articole

UPDATE table SET camp = valoare [WHERE conditie];

-**DELETE –** șterge un set de articole DELETE FROM tabel [WHERE conditie];

Tranzacțiile sunt utilizate pentru a controla în ce condiții se desfășoară o succesiune a instrucțiunilor de manipulare a datelor.

-**START TRANSACTION** (sau **BEGIN WORK**, **BEGIN TRANSACTION**, în funcție de

dialectul SQL) Început de tranzacție.

-**SAVE TRANSACTION** (sau **SAVEPOINT**) salvează starea bazei într-un punct al transacției

-**COMMIT** Operează toate operațiile tranzacției ca fiind permanente.

-**ROLLBACK** Anulează toate operațiile tranzacției începând cu ultimul **COMMIT**.

Instrucțiunile **COMMIT** și **ROLLBACK** termină tranzacția curentă și deblochează datele.

## Limbajul de definire a datelor

În contextul SQL , definirea datelor sau limbajul de descriere a datelor ( DDL ) este o sintaxă pentru crearea și modificarea obiectelor bazei de date, cum ar fi tabele, indici și utilizatori. Instrucțiunile DDL sunt similare cu un limbaj de programare pentru computer pentru definirea structurilor de date , în special a schemelor de baze de date . Exemple obișnuite de DDL includ CREATE, ALTER și DROP.

Comanda **CREATE** este utilizată pentru a stabili o nouă bază de date, tabel, index sau procedură stocată . Instrucțiunea CREATE din SQL creează o componentă într-un sistem de gestionare a bazelor de date relaționale (RDBMS). În specificația SQL 1992, tipurile de componente care pot fi create sunt scheme, tabele , vizualizări , domenii, seturi de caractere , colații , traduceri și afirmații. Multe implementări extind sintaxa pentru a permite crearea de elemente suplimentare, cum ar fi indexuri și profiluri de utilizator. Unele sisteme, cum ar fi PostgreSQL și SQL Server , permit CREATE și alte comenzi DDL, în interiorul unei tranzacții de bază de date și astfel pot fi reduse.

*CREATE TABLE tabel(*

*Camp1 tip1, Camp2 tip2,*

*. . .*

*PRIMARY KEY (index1, index2, . . .)*

*);*

Instrucțiunea **DROP** distruge o bază de date, un tabel, un index sau o vizualizare existente. O instrucțiune DROP în SQL elimină o componentă dintr-un sistem de gestionare a bazelor de date relaționale (RDBMS). Tipurile de obiecte care pot fi abandonate depind de ce RDBMS este utilizat, dar majoritatea acceptă abandonarea tabelelor , a utilizatorilor și a bazelor de date. Unele sisteme permit ca comenzile DROP și alte comenzi DDL să apară în interiorul unei tranzacții și astfel să fie anulate. Sintaxa instrucțiunii: *DROP TABLE tabel;*

Instrucțiunea **ALTER** modifică un obiect de bază de date existent. O instrucțiune ALTER din SQL modifică proprietățile unui obiect din interiorul unui sistem de gestionare a bazelor de date relaționale (RDBMS). Tipurile de obiecte care pot fi modificate depind de ce RDBMS este utilizat.

## Redenumirea unui tabel

*ALTER TABLE tabel RENAME TO nume\_nou\_tabel;*

## Adăugarea de cîmpuri noi

*ALTER TABLE table\_name ADD (camp1 def1, col2 def2, . . .);*

## Modificarea structurii unui cîmp

*ALTER TABLE table\_name MODIFY (camp1 tip1, camp2 tip2, . . .);*

## Stergerea unui cîmp

*ALTER TABLE tabel DROP COLUMN camp;*

Instrucțiunea **TRUNCATE** este utilizată pentru a șterge toate datele dintr-un tabel. Este mult mai rapid decât ȘTERGERE. Sintaxa instrucțiunii: *TRUNCATE TABLE tabel;*

## Microsoft SQL Server

**Microsoft SQL Server** este un sistem de gestionare de [baze de date](https://ro.wikipedia.org/wiki/Baz%C4%83_de_date)relaționale (RDBMS) produs de compania [americană](https://ro.wikipedia.org/wiki/SUA) [Microsoft](https://ro.wikipedia.org/wiki/Microsoft) Corp. Limbajul de interogare este SQL iar extensia procedurală este T-SQL.

Suportă versiunea companiei Microsoft de [SQL](https://ro.wikipedia.org/wiki/SQL) (*Structured Query Language* - limbaj structurat de interogări), cel mai răspândit limbaj de interogare pentru bazele de date. Este considerat un sistem de gestionare a bazelor de date pentru întreprinderi - adică se poate aplica bazelor de date de dimensiuni foarte mari - însă la ultimele versiuni au venit cu mai multe tipuri de licențiere, de la cele free la unele avansate, astfel că în prezent putem spune că SQL Server se poate folosi pentru baze de date de orice dimensiune.

Codul de bază pentru Microsoft SQL Server își are originile în Sybase SQL Server și a reprezentat intrarea Microsoft pe piața bazelor de date relaționale, concurând cu Oracle, IBM și Sybase. Microsoft, Sybase si Ashton-Tate s-au unit pentru a crea și a scoate pe piață prima versiune numită SQL Server 4.2 pentru Win OS/2. Mai tarziu Microsoft a negociat pentru drepturi de exclusivitate la toate versiunile de SQL Server scrise pentru sistemele de operare Microsoft. Sybase și-a schimbat ulterior numele în Adaptive Server Enterprise, pentru a evita confuzia cu Microsoft SQL Server.

SQL Server 7.0 a fost primul server de baze de date bazat pe [GUI](https://ro.wikipedia.org/wiki/GUI). O variantă de SQL Server 2000 a fost prima variantă comercială pentru arhitectura [Intel](https://ro.wikipedia.org/wiki/Intel). In anii următori s-au lansat și alte versiuni, ce au adus îmbunătățiri de performanță, tool-uri IDE și ETL, suport pentru date XML, geometrice sau spațiale, căutare în text, self-tuning, clustering și HADR: SQL Server 2005, SQL Server 2008, SQL Server 2012 și SQL Server 2014. Ultima versiune este SQL Server 2016.

Microsoft SQL Server folosește o variantă de SQL numită T-SQL, sau Transact-SQL, o implementare de SQL-92 (standardul ISO pentru SQL) cu unele extensii. T-SQL în principal adaugă sintaxă adițională pentru procedurile stocate și pentru tranzacții. Standardele SQL necesită ACID; acesta este un acronim pentru cele 4 condiții ce trebuie îndeplinite de orice tranzacție: atomicitate, consistență, izolare, durabilitate. MS SQL Server suportă ODBC (Open Database Connectivity).

Stocarea datelor se face pe modelul relațional, sub forma unei colecții de tabele cu rânduri și coloane. Fiecare coloană este de un anumit tip: SQL Server suportă atât tipurile primare, precum întreg, zecimal, caracter, dată calendaristică, dar și tipuri mai complexe precum text, date binare, date geometrice, date spațiale, XML, etc.

## 2 Definirea temei proiectului și conținutul problemei

O carte este o colecție de [hârtii](https://ro.wikipedia.org/wiki/H%C3%A2rtie), [pergamente](https://ro.wikipedia.org/wiki/Pergament) sau alte astfel de materiale, în formă de coli sau foi de cele mai multe ori egale între ele și legate sau broșate într-un volum. Cărțile sunt de obicei [tipărite](https://ro.wikipedia.org/wiki/Tip%C4%83rire) (rareori scrise și de mână) și conțin diverse lucrări [scrise](https://ro.wikipedia.org/wiki/Scriere), pe cea mai mare diversitate de teme. O carte este de asemenea o operă literară sau științifică sau o parte semnificativă dintr-o astfel de operă.

In zilele noastre pentru a putea detine o carte este destul sa o cumparam de la un magazin de carti, deobicei pe rafturile magazinului se afla la moment cele mai populare carti din diferite domenii si de la diferiti autori.

La randul ei, cartea detine un ISBN, adica numarul de identitate a acestei carti, fara acest numar, cartea nu poate fi vanduta. Pe langa ISBN, cartea contine informatii despre Editura unde a fost tiparita, numarul de pagini, anul tiparirei si nu in ultimul de rand autorul cartii. Iar magazinul de carti unde se afla aceasta carte face o generalizare si indica pretul acestei carti care cititorul o poate procura.

Magazinele de carti cu fiecare an devin mai putin atractive, deoarece tehnologiile se dezvolta si acum aceiasi carte poate fi procurata in format electronic fara sa iesi din casa. Un plus pentru formatul electronic este ca nu se cheltuie foaie, cerneala si energie electrica pentru crearea cartii, dar un minus este ca in comparatie cu cartea obisnuita, cartea electronica nu poate fi citita un timp indelungat din cauza durerilor de ochi si descarcarea acumulatorului gadget-ului in care se afla cartea, iar din partea mea pot spune ca cartea digitala nu are mirosul specific care are cartea obisnuita, chiar si peste multi ani dupa procurarea si cititrea cartii, mirosul acesta ramane.

Cu dezvoltarea tehnologiilor, magazine de carti obisnuite inca mai exista. In Republica Moldova, magazine recunoscute de carti sunt: Librarius, ProNoi si Elefant. In aceste magazine sortimentul cartilor este unul destul de mare. Din propria experienta pot spune ca orice carte care o cautam, o gaseam in unul din aceste magazine.

Tema proiectului este elaborarea unei baze de date pentru un magazin de carti mediu. Baza de date trebuie să permită posibilitatea de stocare a datelor pe diverse categorii de informații și anume:

* Editurile care emit cartile
* Autorii care scriu cartile
* Cartile propriu-zis

## 3 Proiectarea modelului relațional al bazei de date

Proiectând baza de date am avut atenția ca datele stocate să fie rațional utilizate, adica datele nu vor fi duplicate nici într-un tabel. Legătura de mai mulți la mai mulți a fost realizată utilizând un tabel intermediar pentru a evita la fel duplicarea datelor în tabele.

Tabele conțin date după cum urmează:

# Tabelul Carti

# ID-ul cartii (Primary key)

# Titlul

# ID-ul autorului (Foreign key)

# ID-ul editurii (Foreign key)

# ISBN

# Anul aparitiei

# Nr. de pagini

# Pretul cartii

# Tabelul Autori

* ID-ul autorului (Primary key)
* Numele
* Prenumele
* Genul
* Anul nasterii

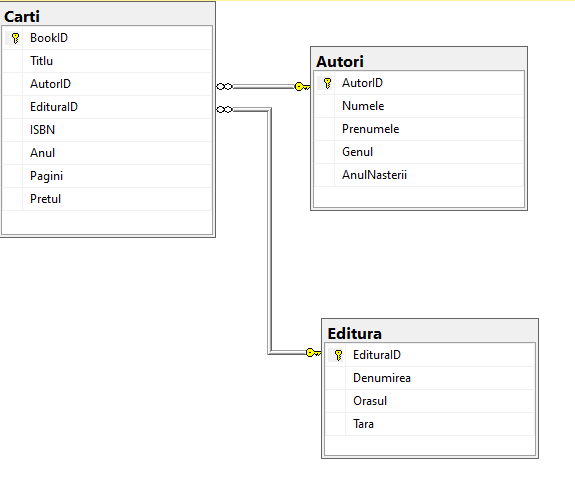
# Tabelul Editura

# ID-ul editurii (Primary key)

# Denumirea companiei

# Orasul

# Tara



**Figura 3 - Schema relațională a bazei de date proiectate**

În figura 3 este reprezentată schema bazei de date relaționale dezvoltată în SQL Server. În SQL Server relațiile și asocierile între ele sunt reprezentate vizual în diagrama de asocieri (Relationships).

Schema relațională pune în evidență legăturile dintre tabele. Definirea legăturilor dintre tabele se face logic construind asocieri între tabele cu ajutorul unor atribute de legătură. Atributul din tabela iniţială se numeşte cheie externă, iar cel din tabela finală este cheie primară. Potenţial, orice tabelă se poate lega cu orice tabelă, după orice atribute. Legăturile se stabilesc la momentul descrierii datelor prin limbaje de descriere a datelor (LDD), cu ajutorul restricţiilor de integritate. Practic, se stabilesc şi legături dinamice la momentul execuţiei.

***3.1 Crearea si intretinerea bazei de date***

Crearea bazei de date cu restricțiile următoare:

* Creșterea fișierului primar al bazei– 64MB, limita de creștere – nelimitat;
* Creșterea logului – 64MB, limita de creștere – 2MB;

Am creat noua bază de date utilizând Wizard-ul incorporat în SQL Server Management Studio pentru că acesta ne permite mai ușor să restricționăm limitele creșterilor.

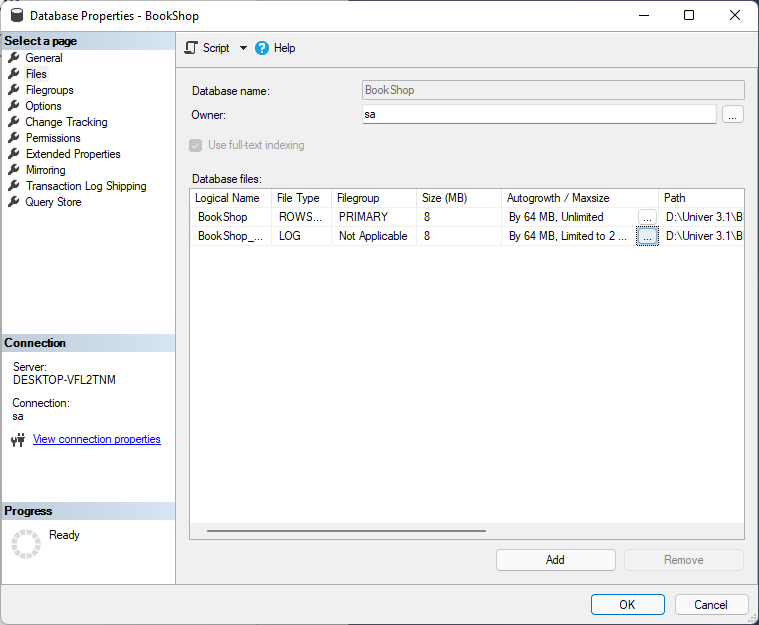
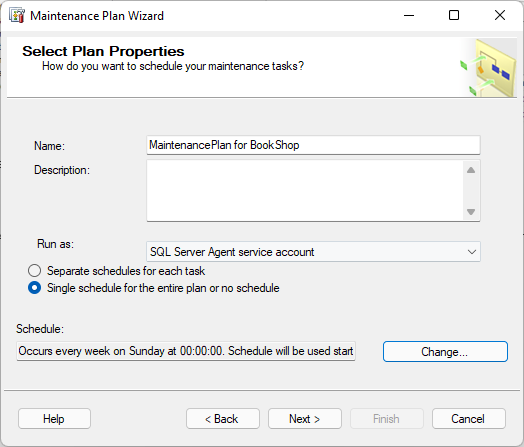
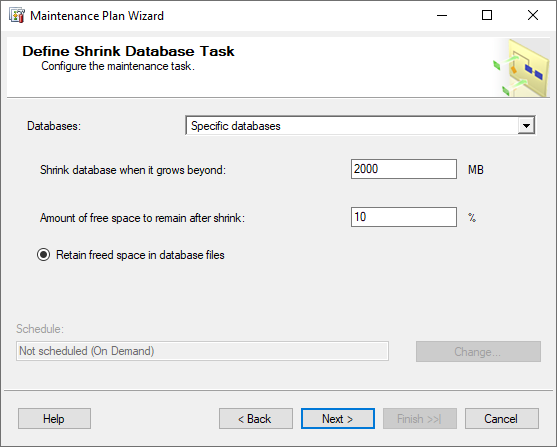


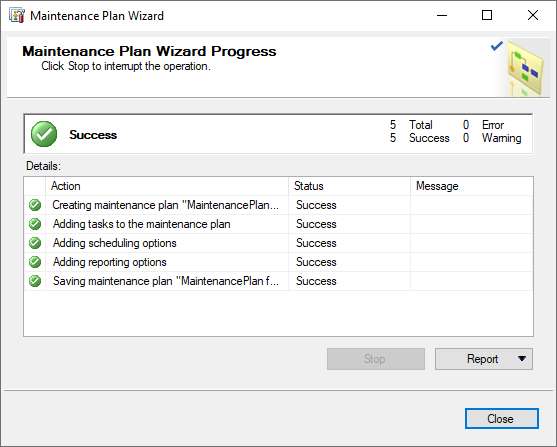
Figura 3.1.1 - Fereastra în care putem manipula cu creșterea fișierului primar și a logului

A fost creat și planul de întreținere a bazei de date, cu ajutorul Wizard-ului

din SQL Server Management Studio.



****



***3.2 Crearea si modificarea tabelelor***

CREATE TABLE Editura (

EdituraID int NOT NULL PRIMARY KEY,

Denumirea varchar(255) NOT NULL,

Orasul varchar(255) NOT NULL,

Tara varchar(255) NOT NULL

);

CREATE TABLE Autori (

AutorID int NOT NULL PRIMARY KEY,

Numele varchar(255) NOT NULL,

Prenumele varchar(255) NOT NULL,

Genul varchar(255) NOT NULL,

AnulNasterii int NOT NULL,

);

CREATE TABLE Carti (

BookID int NOT NULL PRIMARY KEY,

Titlu varchar(255) NOT NULL,

AutorID int NOT NULL FOREIGN KEY REFERENCES Autori(AutorID),

EdituraID int NOT NULL FOREIGN KEY REFERENCES Editura(EdituraID),

ISBN bigint NOT NULL,

Anul int NOT NULL,

Pagini int NOT NULL,

Pretul float

);

***3.3 Diagrame, scheme, sinonime. Inserarea valori în baza de date.***

INSERT INTO Editura

VALUES

(1,'RAO','Bucuresti','Romania'),

(2,'Corint','Bucuresti','Romania'),

(3,'TREI','Bucuresti','Romania'),

(4,'ART','Bucuresti','Romania'),

(5,'Steaua Nordului','Bucuresti','Romania'),

(6,'Mihail Sadoveanu','Bucuresti','Romania'),

(7,'Cartier','Chisinau','Moldova'),

(8,'Policadran','Chisinau','Moldova'),

(9,'Amaltea','Bucuresti','Romania'),

(10,'Bestseller','Chisinau','Moldova'),

(11,'Humanitas','Bucuresti','Romania'),

(12,'Liviu Rebreanu','Bucuresti','Romania'),

(13,'Litera','Bucuresti','Romania'),

(14,'Polirom','Bucuresti','Romania'),

(15,'Nemira','Bucuresti','Romania'),

(16,'Curtea veche','Bucuresti','Romania');

INSERT INTO Autori

VALUES

(1,'Rebreanu','Liviu','masculin',1885),

(2,'Sadoveanu','Mihail','masculin',1880),

(3,'Dostoevsky','Fiodor','masculin',1821),

(4,'Hawkins','Paula','femenin',1972),

(5,'Preda','Marin','masculin',1922),

(6,'Bulgakov','Mihail','masculin',1891),

(7,'Dabija','Nicolae','masculin',1948),

(8,'Blaga','Lucian','masculin',1895),

(9,'Kiyosaki','Robert','masculin',1947),

(10,'Clason','George','masculin',1874),

(11,'Boyne','John','masculin',1971),

(12,'Druta','Ion','masculin',1928),

(13,'Sapkowski','Andrzej','masculin',1948),

(14,'Rowling','Joanne','femenin',1928),

(15,'Brown','Dan','masculin',1964),

(16,'Slavici','Ioan','masculin',1848),

(17, 'Coleho','Paulo','masculin',1947),

(18, 'Zusak','Markus','masculin',1975);

INSERT INTO Carti

VALUES

(1,'Codul lui Da Vinci',15,1,9786066094931,2004,506,110),

(2,'Simbolul pierdut',15,1,9786067761740,2009,601,133),

(3,'Origini',15,1,9786060063759,2017,456,171),

(4,'Maestrul si Margareta',6,2,9786068723259,1967,464,154),

(5,'Fata din tren',4,3,9786067193503,2015,408,180),

(6,'Crima si pedeapsa',3,2,9786069368954,1866,590,165),

(7,'Idiotul',3,14,9789734621279,1869,696,150),

(8,'Padurea spanzuratilor',1,13,9786063367571,1922,352,70),

(9,'Baltagul',2,6,9786069335505,1930,204,108),

(10,'Frunze de dor',12,7,9789975863049,1957,260,110),

(11,'Horodiste',12,8,9789975442305,1953,233,60),

(12,'Hronicul si cantecul varstelor',8,11,9789735060404,1965,184,128),

(13,'Cel mai iubit dintre pamanteni vol.1',5,4,9786067100235,2014,378,70),

(14,'Cel mai iubit dintre pamanteni vol.2',5,4,9786067100242,2014,332,60),

(15,'Cel mai iubit dintre pamanteni vol.3',5,13,9789736755835,2004,428,50),

(16,'Tema pentru acasa',7,10,9789975334952,2020,506,211),

(17,'Tata bogat tata sarac',9,16,9786064402547,2018,232,215),

(18,'Cel mai bogat om din Babilon',10,9,9789737780027,2004,144,66),

(19,'Baiatul cu pijamale in dungi',11,1,9786067761849,2016,224,119),

(20,'Ultima dorinta. Seria Witcher. Cartea I',13,15,9786064305398,2019,296,175),

(21,'Sabia destinului. Seria Witcher. Cartea II',13,15,9786064305381,2019,352,175),

(22,'Sangele elfilor. Seria Witcher. Cartea III',13,15,9786064305800,2019,312,175),

(23,'Vremea dispretului. Seria Witcher. Cartea IV',13,15,9786064305817,2019,328,158),

(24,'Botezul focului. Seria Witcher. Cartea V',13,15,9786064304872,2019,352,175),

(25,'Turnul randunicii. Seria Witcher. Cartea VI',13,15,9786064308979,2020,400,175),

(26,'Harry Potter 1. Piatra filosofala',14,4,9786060861317,2016,260,200),

(27,'Harry Potter 2. Camera secretelor',14,4,9786060861324,2016,270,225),

(28,'Harry Potter 3. Prizonierul din Azkaban',14,4,9786060861973,2016,265,205),

(29,'Harry Potter 4. Pocalul de foc',14,4,9786060862338,2017,450,267),

(30,'Harry Potter 5. Ordinul Phoenix',14,4,9786060862161,2017,990,300),

(31,'Harry Potter 6. Printul Semisange ',14,4,9786067998436,2017,900,295),

(32,'Harry Potter 7. Talismanul mortii',14,4,9786060861331,2018,335,50),

(33,'Mara',16,5,6422600000831,2007,335,50),

(34,'Ion',1,7,9789975860253,2015,464,179),

(35,'Ciuleandra',1,12,9789731898346,2017,152,67),

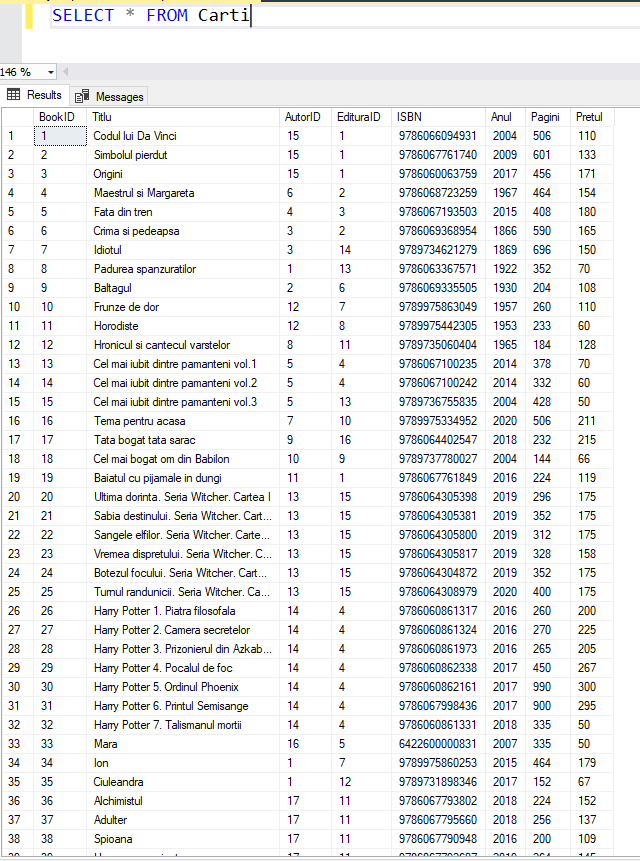
(36,'Alchimistul',17,11,9786067793802,2018,224,152),

(37,'Adulter',17,11,9786067795660,2018,256,137),

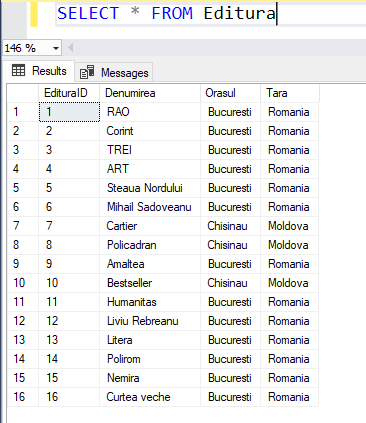
(38,'Spioana',17,11,9786067790948,2016,200,109),

(39,'Unsprezece minute',17,11,9786067792607,2019,264,145),

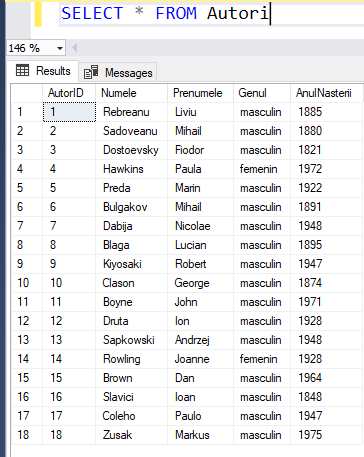
(40,'Hotul de carti. Curaj mai presus de cuvinte',18,1,9786066096133,2011,440,154);



**Figura 3.3.1 - Datele în tabelul Carti**

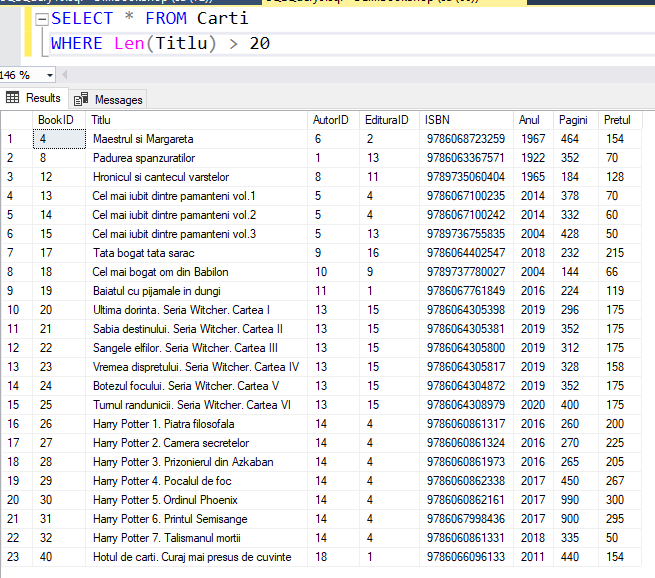


**Figura 3.3.2 - Datele în tabelul Editura**

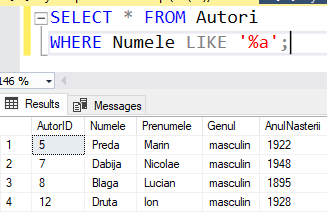


**Figura 3.3.3 - Datele în tabelul Autori**

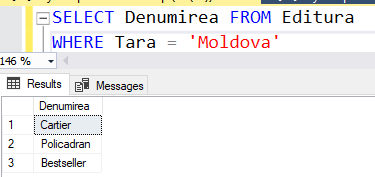
**3.4 Crearea și gestiunea interogărilor SQL**



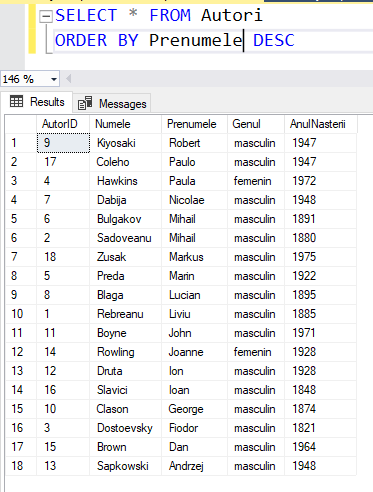
**Figura 3.4.1 – Afisarea tuturor cartilor cu titlu mai mare de 20 caractere**



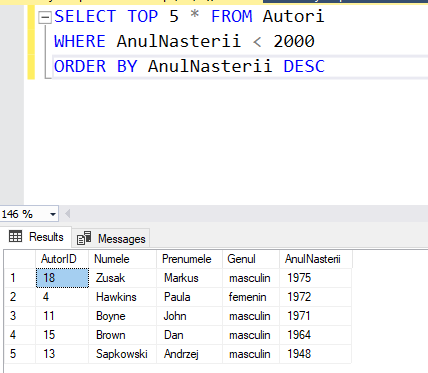
**Figura 3.4.2 – Afisarea tuturor autorilor numele carora se termina cu litera „a”**



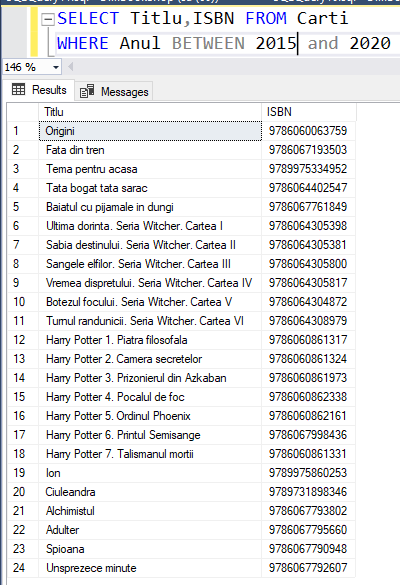
**Figura 3.4.3 – Afisarea tuturor editurilor din Moldova**



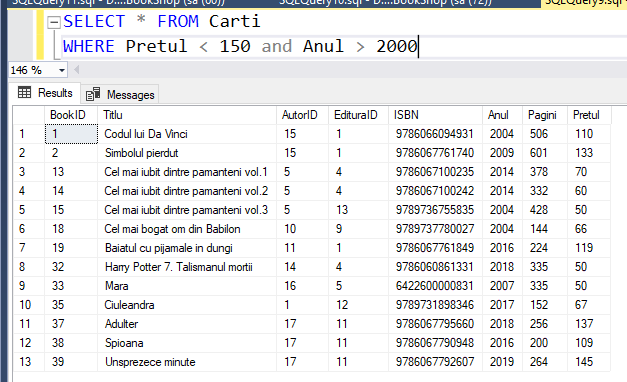
**Figura 3.4.4 – Ordonarea descrescator dupa prenumele autorilor**



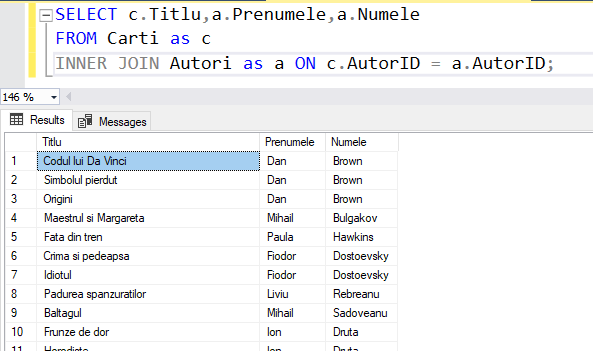
**Figura 3.4.5 – Afisarea primii 5 autori care s-au nascut inainte de anul 2000, ordonati descrescator**



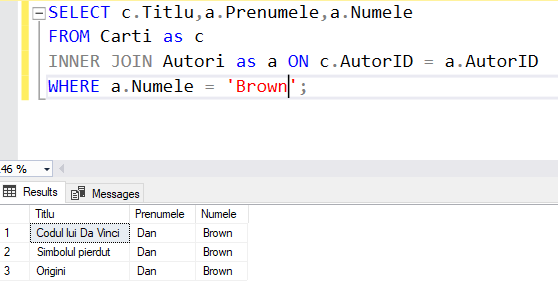
**Figura 3.4.6 – Afisarea titlului si ISBN-ului a cartilor care au aparut intre anii 2015 si 2020**



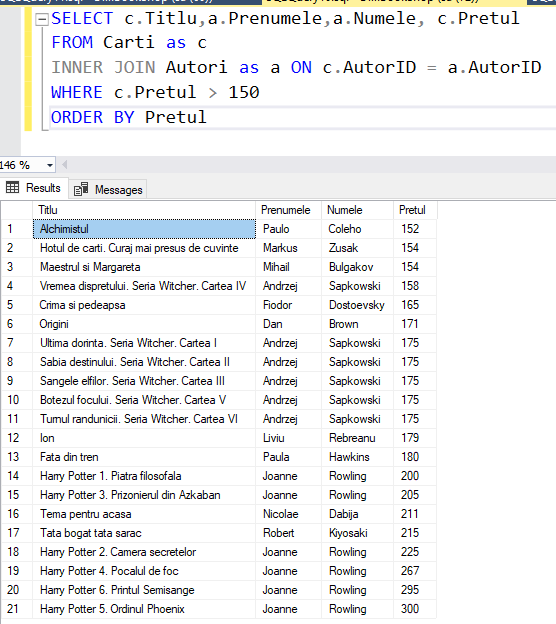
**Figura 3.4.7 – Afisarea cartilor care pretul este mai mic de 150 si anul aparitiei mai mare decat 2000**



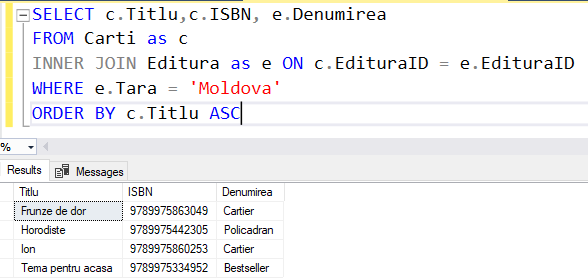
**Figura 3.4.8 – Afisarea titlu cartilor si numele/prenumele autorului**



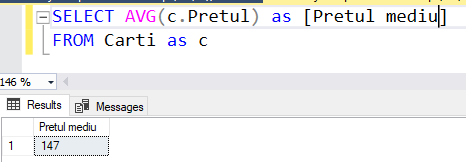
**Figura 3.4.9 – Afisarea titlu cartilor si numele/prenumele autorului care numele este „Brown”**



**Figura 3.4.10 – Afisarea titlu cartilor si numele/prenumele autorului care pretul cartii este mai mare de 150, ordonate crescator dupa pretul cartii**

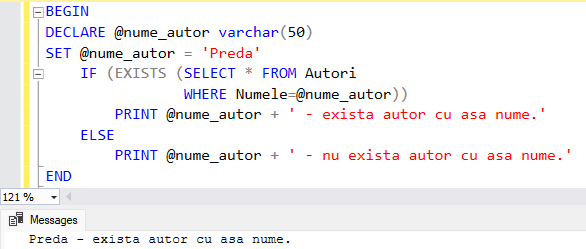


**Figura 3.4.11 – Afisarea titlu cartilor, ISBN-ul si denumirea editurii care tara este Moldova, ordonate crescator dupa titlu**

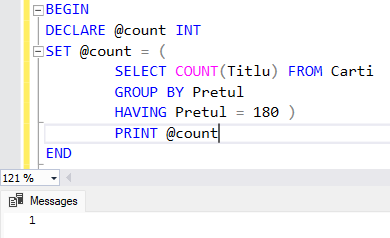


**Figura 3.4.12 – Afisarea pretului mediu a cartilor intr-un tabel temporar**

***3.5 Instrucțiuni procedurale***

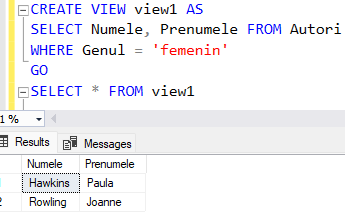


**Figura 3.5.1 – Verificarea existentei autorului cu numele „Preda”**

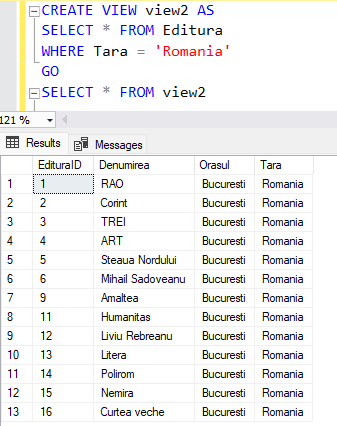


**Figura 3.5.2 – Afisarea numarului de carti cu pretul egal cu 180**

***3.6 Viziuni***

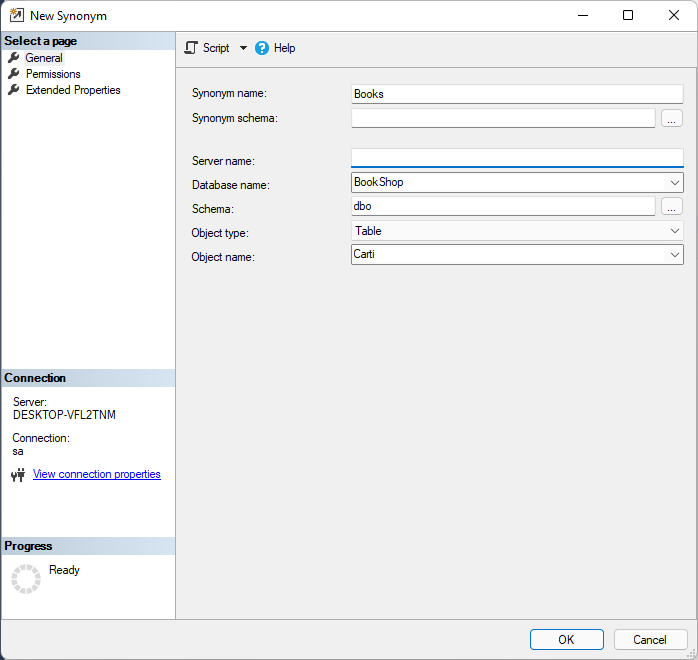


**Figura 3.6.1 – Viziunea view1**

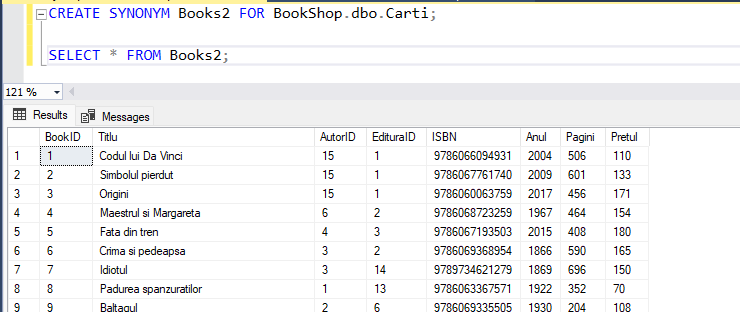


**Figura 3.6.2 – Viziunea view2**

***3.7 Sinonime***

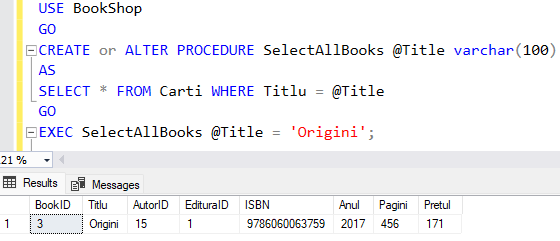


**Fig.3.7.1 Crearea unui sinonim in Microsoft SQL Server Management**

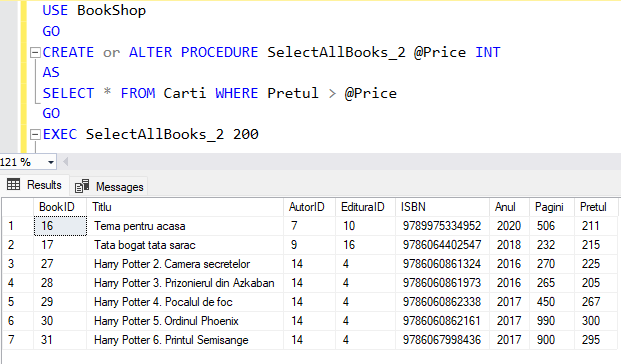


**Fig.3.7.2 Crearea unui sinonim in Transact-SQL**

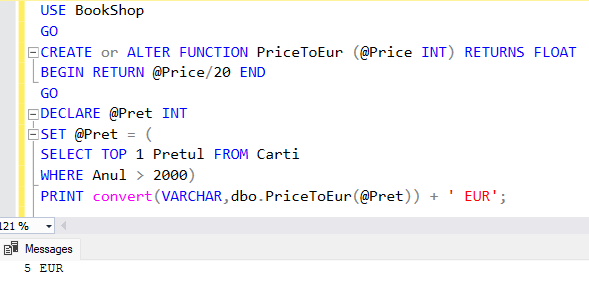
***3.8 Proceduri stocate și funcții definite***



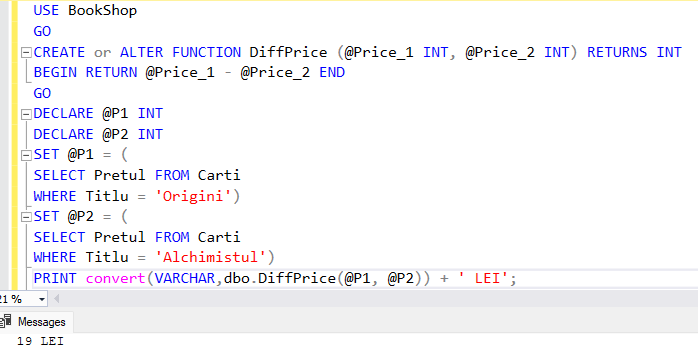
**Figura 3.8.1 - Rezultatele execuției procedurii SelectAllBooks**



**Figura 3.8.2 - Rezultatele execuției procedurii SelectAllBooks\_2**



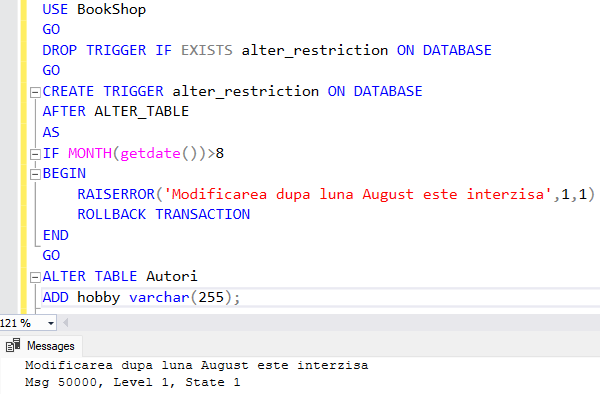
**Figura 3.8.3 - Rezultatele execuției funcției PriceToEur**



**Figura 3.8.4 - Rezultatele execuției funcției DiffPrice**

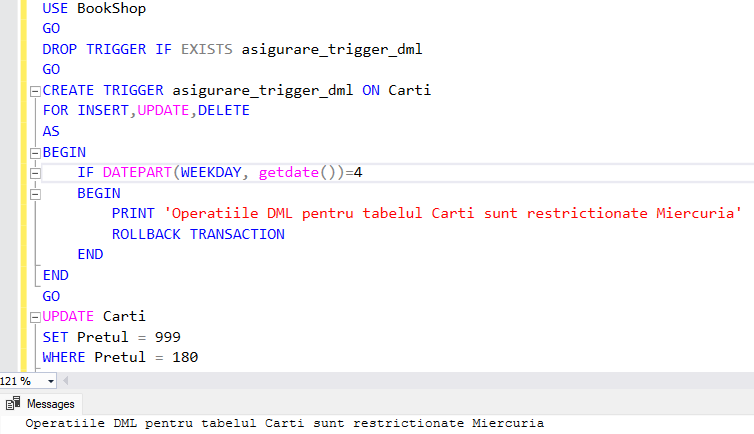
***3.9 Declanșatoare***

1. Crearea unui declanșator DDL care interzice modificarea datelor dupa data de 8 a lunii curente



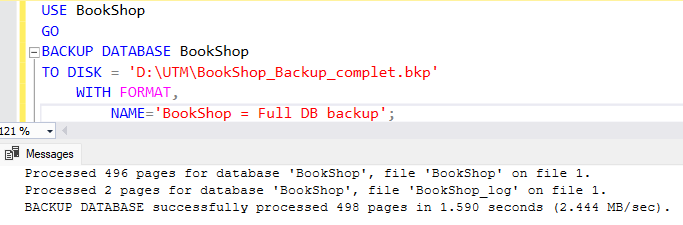
**Figura 3.9.1 - Mesajul afișat la adăugarea coloanei hobby**

1. Crearea unui declanșator DML, în care operațiile asupra tabelului Asigurarile sunt interzise sâmbăta.

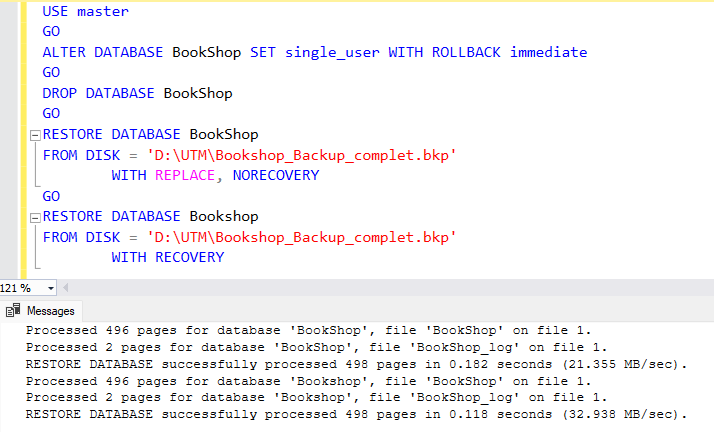


**Figura 3.9.2 - Operațiile asupra tabelului Carti sunt interzise Miercuria**

***3.10 Recuperarea bazei de date***



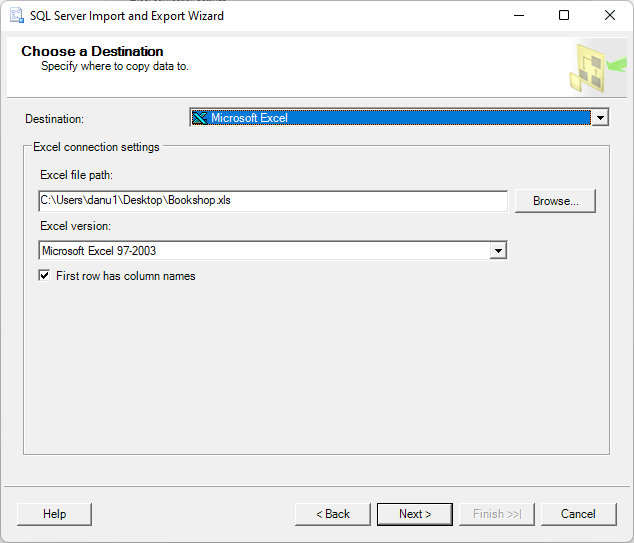
**Figura 3.10.1 – Backup complet a bazei de date**



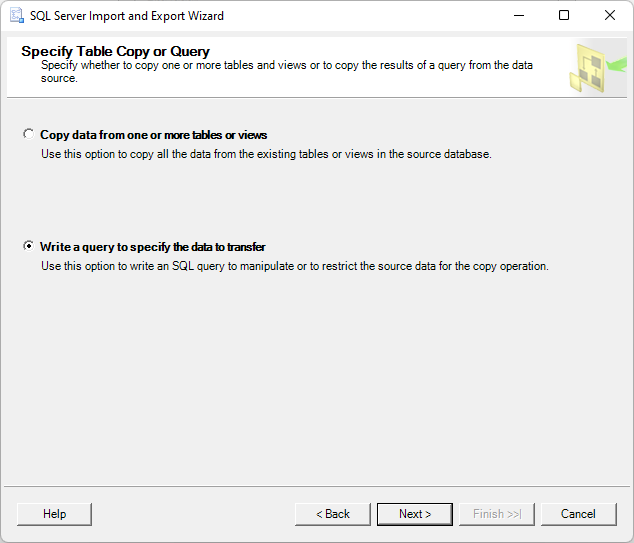
**Figura 3.10.2 - Mesajul de succes precum că backup-ul a fost realizat cu succes**

***3.11 Exportarea și importarea datelor***

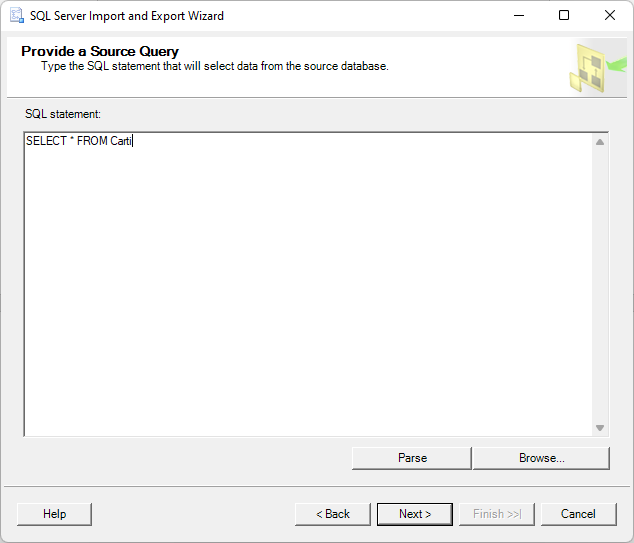
Deseori, se întâmplă așa că este nevoie să se efectueze un proces de import sau export de o singură dată fără a consacra mult timp pentru dezvoltarea, depanarea și asigurare continuității de dezvoltare.



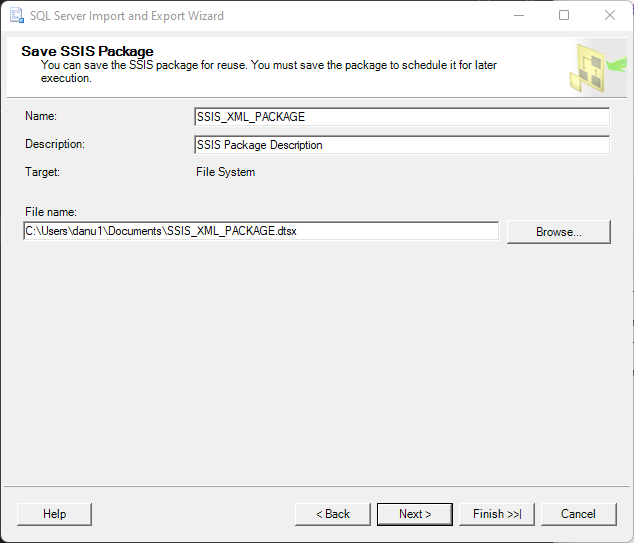
**Figura 3.11.1 Definirea exportării într-un fișier Excel**

****

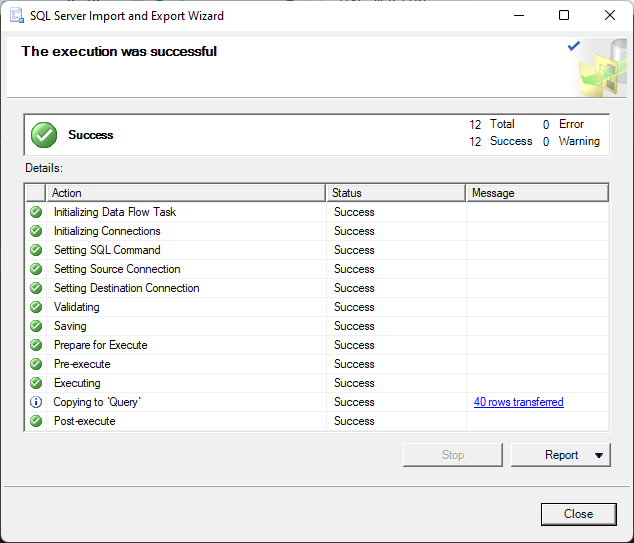
**Figura 3.11.2 Indicarea formei de extragere**

****

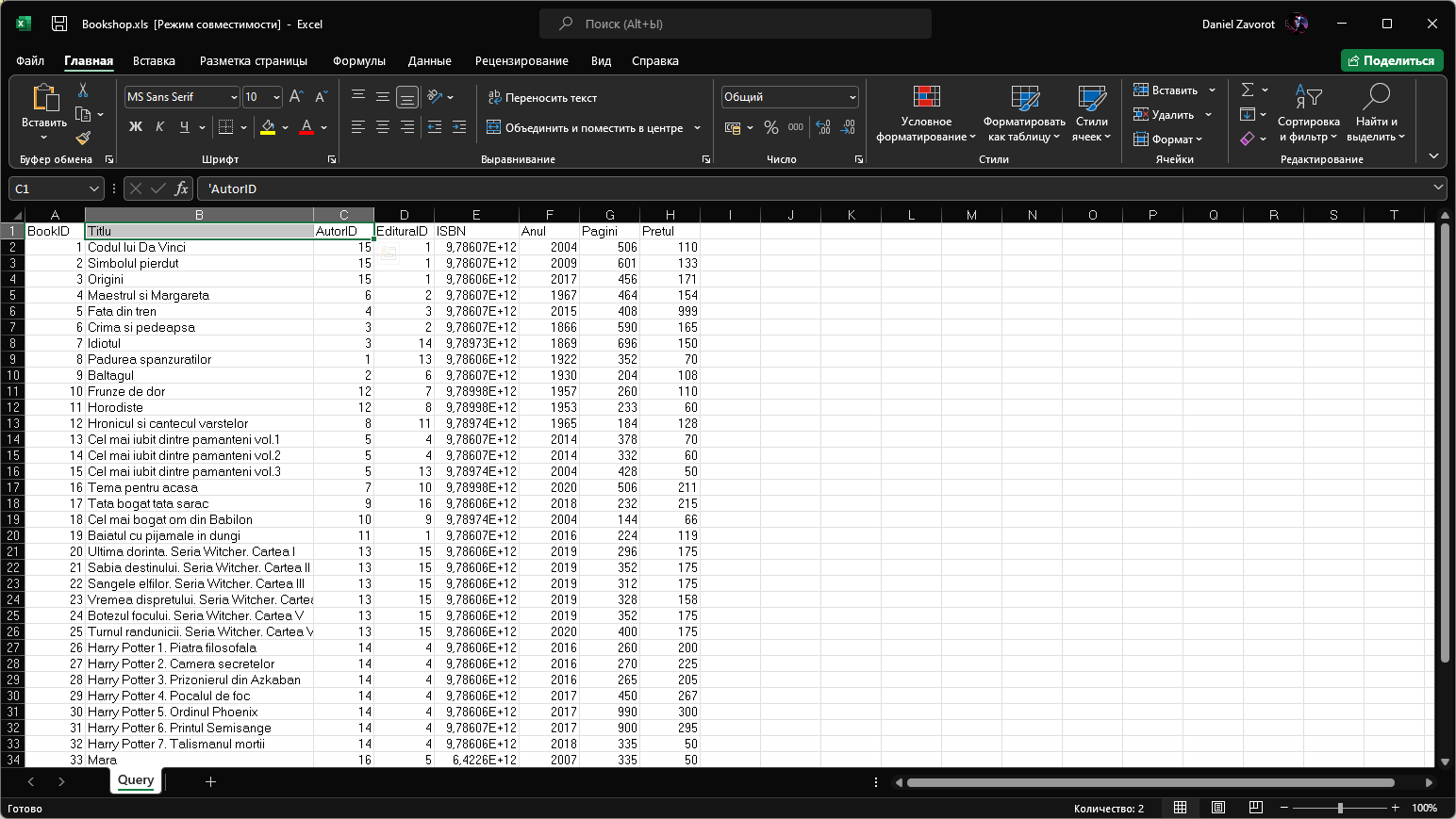
**Figura 3.11.3 Interogarea în casetă**

****

**Figura 3.11.4 Numele, descrierea și locația fișierului**

****

**Figura 3.11.5 Executarea cu succes**

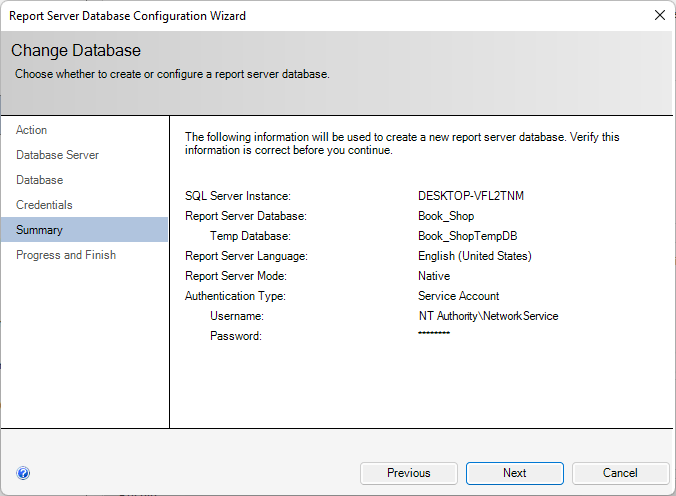
****

**Figura 3.11.6 Exportarea în MS Excel**

***3.12 Rapoarte***

Microsoft SQL Server 2019 Reporting Services (SSRS) oferă o gamă completă de instrumente și servicii pentru crearea, distribuirea și gestiunea rapoartelor despre datele extrase din baza de date și pune la dispoziție o serie de facilități ce permit ajustarea și extinderea funcționalităților de generare a rapoartelor pe cale programabilă.

SSRS reprezintă o platformă de generare a rapoartelor bazată pe server, care oferă funcționalități comprehensive de generare a rapoartelor pentru o varietate largă de surse de date. Serviciile includ un set complet de instrumente pentru crearea, gestiunea și expedierea rapoartelor. Interfețe API permit programatorilor să integreze sau să-și extindă posibilitățile de procesare a datelor și rapoartelor în aplicațiile utilizatorilor.



**Figura 3.12.1 - Configurarea serverului de raportare a bazei de date**

După configurarea serverului de raportare a bazei de date, în *SQL Management Studio* în lista bazelor de date, pot fi văzute 2 baze de date noi - Book**\_**Shop și Book**\_**ShopTempDB*.* Aceste baze de date conțin metadate despre rapoartele desfășurate în sistemul SSRS, istoricul utilizării rapoartelor și alte informații de serviciu necesare pentru asigurarea funcționării SSRS.

## Concluzie

Efectuând acest proiect am căpătat abilități de gestionare a bazelor de date. Am înțeles că în spatele la majotatea aplicațiilor stă o bază de date destul de impunătoare.

Pentru funcționarea eficientă și corectă a aplicațiilor este necesar ca baza de date care stă în spate să fie proiectată corect. Am însușit cum este corect de proiectat o bază de date, care sunt algoritmii de normalizare și anume de ce este importantă forma normală a unei baze de date. Am căpătat practici utile de lucru cu Microsoft SQL Server și limbajul Transact-SQL. Am încercat să reconstituiesc o bază de date a unui magazin de carti.

Numărul foarte mare de informaţii şi evoluţia lor în timp face ca la un moment dat evidenţa acestora să nu mai poată fi controlată fără ajutorul unei aplicatii specializate şi uşor de utilizat, în funcţie de specificul societaţii comerciale. De aceea se recomandă utilizarea unei baze de date sau a unui program informatic special conceput.

Limbajul SQL are la bază operaţiile algebrei relaţionale. Printr-o singură instrucţiune SQL, se poate exprima o interogare care presupune o secvenţă de operaţii *SELECT*, *PROJECT* şi *JOIN*, nefiind necesară o anumită ordine a acestora. Deşi forma de exprimare a unei instrucţiuni SQL pare a fi imperativă, în esenţă, instrucţiunea este de tip declarativ. Drept urmare limbajul SQL îl scuteşte pe utilizator de necesitatea dezvoltării unei secvenţe de paşi care trebuie parcurşi pentru obţinerea informaţiei dorite – tot ce are de făcut este să descrie informaţia de care are nevoie. Majoritatea instrucţiunilor SQL sunt executabile, ele putând fi interpretate şi executate imediat în mod interactiv sau putând fi incluse în diferite aplicaţii scrise în limbaje de programare de nivel înalt, executându-se în momentul rulării programului respectiv.

Dialectul SQL folosit în SQL Server este numit Transact- SQL și este implementat de Microsoft ca și componenta de bază a SQL Server. Transact – SQL este caracterizat ca fiind suplu și puternic. Cuprinde instrucțiuni pentru crearea unităților logice de stocare, a fișierelor care conțin bazele de date, a tabelelor care se memorează în cadrul bazelor de date. Instrucțiunile Transact – SQL se pot utiliza în scopul adăugării și manipulării datelor și a altor obiecte din bazele de date. Există patru cuvinte cheie care se folosesc pentru compunerea instrucțiunilor ce efectuează stocarea, modificarea și regăsirea datelor: INSERT (adaugă noi linii într-o tabelă), DELETE (șterge linii dintr-o tabelă), UPDATE (modifică liniile dintr-o tabelă), SELECT permite regăsirea datelor din una sau mai multe tabele. Extensiile aduse de Transact-SQL privesc în principal controlul fluxului, urmărind determinarea ordinii de execuție a instrucțiunilor.

***Surse bibliografice:***

1. V. Cotelea, M. Cotelea – îndrumar lucrări de laborator ʺMicrosoft SQL Server 2017 pas cu pasʺ
2. Curs SQL [resurse electronice]: <https://www.w3schools.com/sql/>

***Anexa***

CREATE TABLE Editura (

EdituraID int NOT NULL PRIMARY KEY,

Denumirea varchar(255) NOT NULL,

Orasul varchar(255) NOT NULL,

Tara varchar(255) NOT NULL

);

CREATE TABLE Autori (

AutorID int NOT NULL PRIMARY KEY,

Numele varchar(255) NOT NULL,

Prenumele varchar(255) NOT NULL,

Genul varchar(255) NOT NULL,

AnulNasterii int NOT NULL,

);

CREATE TABLE Carti (

BookID int NOT NULL PRIMARY KEY,

Titlu varchar(255) NOT NULL,

AutorID int NOT NULL FOREIGN KEY REFERENCES Autori(AutorID),

EdituraID int NOT NULL FOREIGN KEY REFERENCES Editura(EdituraID),

ISBN bigint NOT NULL,

Anul int NOT NULL,

Pagini int NOT NULL,

Pretul float

);

INSERT INTO Editura

VALUES

(1,'RAO','Bucuresti','Romania'),

(2,'Corint','Bucuresti','Romania'),

(3,'TREI','Bucuresti','Romania'),

(4,'ART','Bucuresti','Romania'),

(5,'Steaua Nordului','Bucuresti','Romania'),

(6,'Mihail Sadoveanu','Bucuresti','Romania'),

(7,'Cartier','Chisinau','Moldova'),

(8,'Policadran','Chisinau','Moldova'),

(9,'Amaltea','Bucuresti','Romania'),

(10,'Bestseller','Chisinau','Moldova'),

(11,'Humanitas','Bucuresti','Romania'),

(12,'Liviu Rebreanu','Bucuresti','Romania'),

(13,'Litera','Bucuresti','Romania'),

(14,'Polirom','Bucuresti','Romania'),

(15,'Nemira','Bucuresti','Romania'),

(16,'Curtea veche','Bucuresti','Romania');

INSERT INTO Autori

VALUES

(1,'Rebreanu','Liviu','masculin',1885),

(2,'Sadoveanu','Mihail','masculin',1880),

(3,'Dostoevsky','Fiodor','masculin',1821),

(4,'Hawkins','Paula','femenin',1972),

(5,'Preda','Marin','masculin',1922),

(6,'Bulgakov','Mihail','masculin',1891),

(7,'Dabija','Nicolae','masculin',1948),

(8,'Blaga','Lucian','masculin',1895),

(9,'Kiyosaki','Robert','masculin',1947),

(10,'Clason','George','masculin',1874),

(11,'Boyne','John','masculin',1971),

(12,'Druta','Ion','masculin',1928),

(13,'Sapkowski','Andrzej','masculin',1948),

(14,'Rowling','Joanne','femenin',1928),

(15,'Brown','Dan','masculin',1964),

(16,'Slavici','Ioan','masculin',1848),

(17, 'Coleho','Paulo','masculin',1947),

(18, 'Zusak','Markus','masculin',1975);

INSERT INTO Carti

VALUES

(1,'Codul lui Da Vinci',15,1,9786066094931,2004,506,110),

(2,'Simbolul pierdut',15,1,9786067761740,2009,601,133),

(3,'Origini',15,1,9786060063759,2017,456,171),

(4,'Maestrul si Margareta',6,2,9786068723259,1967,464,154),

(5,'Fata din tren',4,3,9786067193503,2015,408,180),

(6,'Crima si pedeapsa',3,2,9786069368954,1866,590,165),

(7,'Idiotul',3,14,9789734621279,1869,696,150),

(8,'Padurea spanzuratilor',1,13,9786063367571,1922,352,70),

(9,'Baltagul',2,6,9786069335505,1930,204,108),

(10,'Frunze de dor',12,7,9789975863049,1957,260,110),

(11,'Horodiste',12,8,9789975442305,1953,233,60),

(12,'Hronicul si cantecul varstelor',8,11,9789735060404,1965,184,128),

(13,'Cel mai iubit dintre pamanteni vol.1',5,4,9786067100235,2014,378,70),

(14,'Cel mai iubit dintre pamanteni vol.2',5,4,9786067100242,2014,332,60),

(15,'Cel mai iubit dintre pamanteni vol.3',5,13,9789736755835,2004,428,50),

(16,'Tema pentru acasa',7,10,9789975334952,2020,506,211),

(17,'Tata bogat tata sarac',9,16,9786064402547,2018,232,215),

(18,'Cel mai bogat om din Babilon',10,9,9789737780027,2004,144,66),

(19,'Baiatul cu pijamale in dungi',11,1,9786067761849,2016,224,119),

(20,'Ultima dorinta. Seria Witcher. Cartea I',13,15,9786064305398,2019,296,175),

(21,'Sabia destinului. Seria Witcher. Cartea II',13,15,9786064305381,2019,352,175),

(22,'Sangele elfilor. Seria Witcher. Cartea III',13,15,9786064305800,2019,312,175),

(23,'Vremea dispretului. Seria Witcher. Cartea IV',13,15,9786064305817,2019,328,158),

(24,'Botezul focului. Seria Witcher. Cartea V',13,15,9786064304872,2019,352,175),

(25,'Turnul randunicii. Seria Witcher. Cartea VI',13,15,9786064308979,2020,400,175),

(26,'Harry Potter 1. Piatra filosofala',14,4,9786060861317,2016,260,200),

(27,'Harry Potter 2. Camera secretelor',14,4,9786060861324,2016,270,225),

(28,'Harry Potter 3. Prizonierul din Azkaban',14,4,9786060861973,2016,265,205),

(29,'Harry Potter 4. Pocalul de foc',14,4,9786060862338,2017,450,267),

(30,'Harry Potter 5. Ordinul Phoenix',14,4,9786060862161,2017,990,300),

(31,'Harry Potter 6. Printul Semisange ',14,4,9786067998436,2017,900,295),

(32,'Harry Potter 7. Talismanul mortii',14,4,9786060861331,2018,335,50),

(33,'Mara',16,5,6422600000831,2007,335,50),

(34,'Ion',1,7,9789975860253,2015,464,179),

(35,'Ciuleandra',1,12,9789731898346,2017,152,67),

(36,'Alchimistul',17,11,9786067793802,2018,224,152),

(37,'Adulter',17,11,9786067795660,2018,256,137),

(38,'Spioana',17,11,9786067790948,2016,200,109),

(39,'Unsprezece minute',17,11,9786067792607,2019,264,145),

(40,'Hotul de carti. Curaj mai presus de cuvinte',18,1,9786066096133,2011,440,154);

SELECT \* FROM Carti

SELECT \* FROM Autori

SELECT \* FROM Editura

SELECT \* FROM Carti

WHERE Len(Titlu) > 20

SELECT \* FROM Autori

WHERE Numele LIKE '%a'

SELECT \* FROM Editura

WHERE Tara = 'Moldova'

SELECT \* FROM Autori

ORDER BY Prenumele DESC

SELECT TOP 5 \* FROM Autori

WHERE AnulNasterii < 2000

ORDER BY AnulNasterii DESC

SELECT Titlu,ISBN FROM Carti

WHERE Anul BETWEEN 2015 and 2020

SELECT c.Titlu,a.Prenumele,a.Numele

FROM Carti as c

INNER JOIN Autori as a ON c.AutorID = a.AutorID

SELECT c.Titlu,a.Prenumele,a.Numele

FROM Carti as c

INNER JOIN Autori as a ON c.AutorID = a.AutorID

WHERE a.Numele = 'Brown'

SELECT c.Titlu,a.Prenumele,a.Numele,c.Pretul

FROM Carti as c

INNER JOIN Autori as a ON c.AutorID = a.AutorID

WHERE c.Pretul > 150

ORDER BY Pretul

SELECT c.Titlu,c.ISBN,e.Denumirea

FROM Carti as c

INNER JOIN Editura as e ON c.EdituraID = e.EdituraID

WHERE e.Tara = 'Moldova'

ORDER BY c.Titlu ASC

SELECT AVG(c.Pretul) as [Pretul mediu]

FROM Carti as c

BEGIN

DECLARE @nume\_autor varchar(50)

SET @nume\_autor = 'Preda'

IF(EXISTS (SELECT \* FROM Autori

WHERE Numele=@nume\_autor))

PRINT @nume\_autor + ' - exista autor cu asa nume.'

ELSE

PRINT @nume\_autor + ' - nu exista autor cu asa nume.'

END

BEGIN

DECLARE @count INT

SET @count = (

SELECT COUNT(Titlu) FROM Carti

GROUP BY Pretul

HAVING Pretul = 180 )

PRINT @count

END

USE BookShop

GO

CREATE VIEW view1 AS

SELECT Numele, Prenumele FROM Autori

WHERE Genul = 'femenin'

GO

SELECT \* FROM view1

USE BookShop

GO

CREATE VIEW view2 AS

SELECT \* FROM Editura

WHERE Tara = 'Romania'

GO

SELECT \* FROM view2

USE BookShop

GO

CREATE or ALTER PROCEDURE SelectAllBooks @Title varchar(100)

AS

SELECT \* FROM Carti WHERE Titlu = @Title

GO

EXEC SelectAllBooks @Title = 'Origini';

USE BookShop

GO

CREATE or ALTER PROCEDURE SelectAllBooks\_2 @Price INT

AS

SELECT \* FROM Carti WHERE Pretul > @Price

GO

EXEC SelectAllBooks\_2 200;

USE BookShop

GO

CREATE or ALTER FUNCTION PriceToEur (@Price INT) RETURNS FLOAT

BEGIN RETURN @Price/20 END

GO

DECLARE @Pret INT

SET @Pret = (

SELECT TOP 1 Pretul FROM Carti

WHERE Anul > 2000)

PRINT convert(VARCHAR, dbo.PriceToEur(@Pret)) + ' EUR'

USE BookShop

GO

CREATE or ALTER FUNCTION DiffPrice (@Price\_1 INT, @Price\_2 INT) RETURNS INT

BEGIN RETURN @Price\_1 - @Price\_2 END

GO

DECLARE @P1 INT

DECLARE @P2 INT

SET @P1 = (

SELECT Pretul FROM Carti

WHERE Titlu = 'Origini')

SET @P2 = (

SELECT Pretul FROM Carti

WHERE Titlu = 'Alchimistul')

PRINT convert(VARCHAR, dbo.DiffPrice(@P1, @P2)) + ' LEI'

USE BookShop

GO

DROP TRIGGER IF EXISTS alter\_restriction ON DATABASE

GO

CREATE TRIGGER alter\_restriction ON DATABASE

AFTER ALTER\_TABLE

AS

IF MONTH(getdate())>8

BEGIN

RAISERROR('Modificarea dupa luna August este interzisa',1,1)

ROLLBACK TRANSACTION

END

GO

ALTER TABLE Autori

ADD hobby varchar(255);

USE BookShop

GO

DROP TRIGGER IF EXISTS asigurarea\_trigger\_dml

GO

CREATE TRIGGER asigurarea\_trigger\_dml ON Carti

FOR INSERT,UPDATE,DELETE

AS

BEGIN

IF DATEPART(WEEKDAY, getdate())=4

BEGIN

PRINT 'Operatiile DML pentru tabelul Carti sunt restrictionate Miercuria'

ROLLBACK TRANSACTION

END

END

GO

UPDATE Carti

SET Pretul = 999

WHERE Pretul = 180

USE BookShop

GO

BACKUP DATABASE BookShop

TO DISK = 'D:\UTM\BookShop\_Backup\_complete.bkp'

WITH FORMAT,

NAME='BookShop = Full DB backup'

USE master

GO

ALTER DATABASE BookShop SET single\_user WITH ROLLBACK immediate

GO

DROP DATABASE BookShop

GO

RESTORE DATABASE BookShop

FROM DISK = 'D:\UTM\BookShop\_Backup\_complete.bkp'

WITH REPLACE, NORECOVERY

GO

RESTORE DATABASE BookShop

FROM DISK = 'D:\UTM\BookShop\_Backup\_complete.bkp'

WITH RECOVERY