**MINISTERUL EDUCAŢIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Ingineria Software și Automatică**

**Programul de studii: Tehnologia informației**

RAPORT

# LUCRARE DE LABORATOR NR. 1

# la Baze de Date

A efectuat:

st. gr. TI-211 Popa Cătălin

A verificat: Olga Grosu

UTM, Chișinău 2023

**Diagramele în baza de date SQL Server**

Diagramele în baza de date SQL Server sunt instrumente vizuale care oferă o reprezentare grafică a structurii și relațiilor dintre entitățile dintr-o bază de date. Aceste diagrame sunt utile pentru proiectarea, înțelegerea și administrarea schemelor de bază de date. Pentru a crea diagrame în SQL Server Management Studio (SSMS), poți utiliza funcționalitățile integrate de proiectare a bazelor de date. Este important să menții diagramele actualizate pe măsură ce faci modificări în structura bazei de date pentru a asigura coerența și înțelegerea clară a acesteia.

Pentru a crea o diagram, este nevoie să urmăm următorii pași:

1. Navigăm spre baza noastră de date și dăm click dpreapta pe Database Diagrams precum în figura 1.

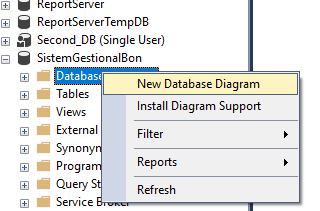


Figura 1 – Creare diagramă

1. După este nevoie să alegem din lista afișată în figura 2, tabele care dorim să le folosim în diagram și le adăugăm.

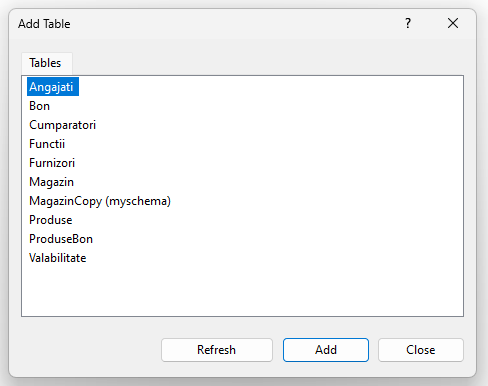


Figura 2 – Selectare tabele

1. Întru-un final, ar trebui să se afișeze diagrama bazei de date, precum este în figura 3.

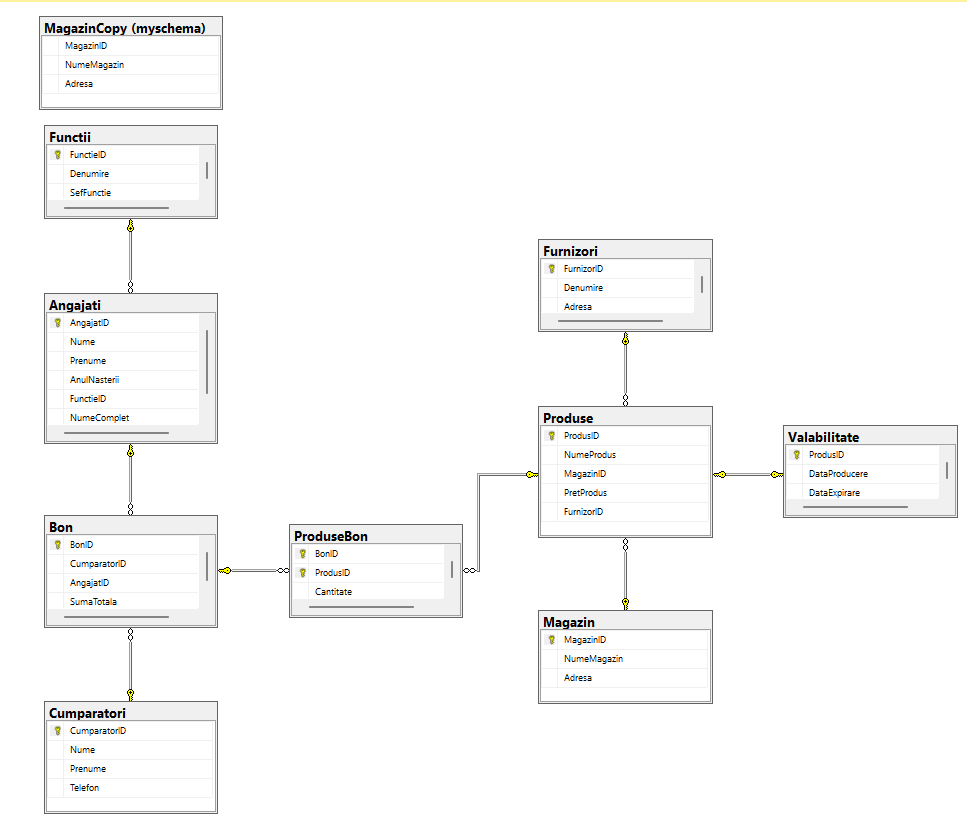


Figura 3 – Diagrama bazei de date ,,SistemGestionalBon’’.

**Administrarea schemelor**

Administrarea schemelor în SQL se referă la procesul de gestionare a structurii și organizării obiectelor în cadrul unei baze de date. O schemă este un container logic pentru obiecte, precum tabele, vederi, proceduri stocate și alte tipuri de obiecte dintr-o bază de date.

Pentru a crea o schemă nouă, este nevoie de urmat următorul cod SQL:

USE SistemGestionalBon

go

Create schema myschema AUTHORIZATION Catalin;

Astfel, după executare, putem observa că în baza de date a apărut o nou schemă cu denumirea new\_schema. Afișarea ei este în figura 4.



Figura 4 – Afișarea schemei noi

**Modificarea schemelor**

Pentru alocarea permisiunilor sau a incredita altui user careva imputerniciri asupra schemei este necesar sa utilizam „GRANT”. Intrucat schema noastra o detine utilizatorul „Catalin” o sa oferim imputerniciri de „Select și Insert” pentru utilizatorul admin.

Pentru aceasta, vom folosi următorul cod sql:

USE SistemGestionalBon

go

grant select, insert on schema :: myschema to Catalin3;

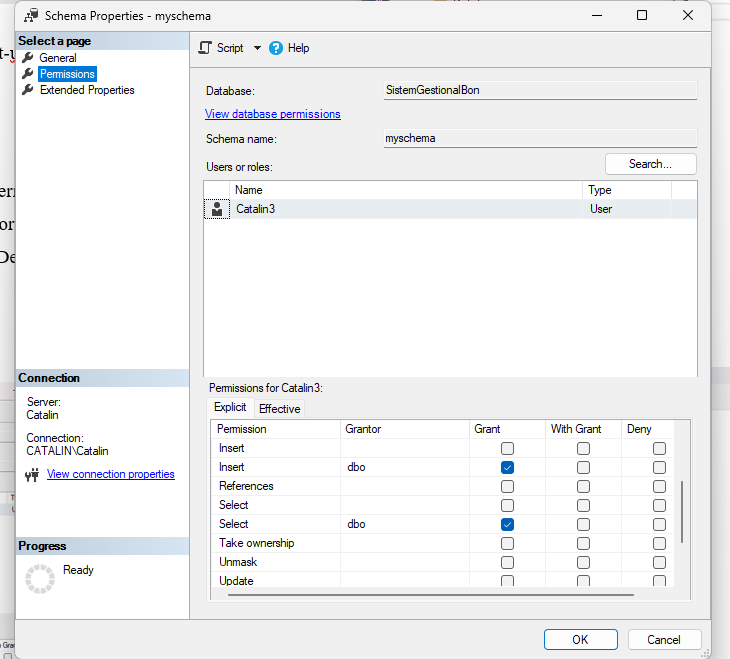


Figura 5 – Acordarea drepturilor pentru Catalin3 pe schema new\_schema.

**Crearea și suprimarea sinonimelor**

În contextul unei baze de date MSSQL (Microsoft SQL Server), termenul "sinonim" se referă la un obiect de bază de date care permite unui alt obiect să fie denumit folosind un alt nume. Practic, un sinonim este o referință simplificată către un obiect dintr-o bază de date.

Pentru crearea unui sinonim, folsim următorul cod SQL:

CREATE SYNONYM MARKET

FOR new\_schema.Magazin

go

select \* from MARKET

În figura 6, observă că sinonimul creat a apărut în baza de date.



Figura 6 – Sinonim

La fel cum am creat un sinonim, putem și să îl ștergem din baza de date. Pentru aceasta, executăm următorul cod:

drop synonym MARKET

Astfel, sinonimul va fi șters din baza de date.

**Indecși**

**Indexuri cluster:**

Un index cluster reordonează datele tabelului pe baza cheii de index. Tabelul însuși este organizat în jurul cheii de index, iar datele sunt stocate în ordinea acelei chei. Un tabel poate avea doar un singur index cluster.

În exemplu de mai jos ,această comandă creează un index pe tabelul MagazinCopy pentru a optimiza interogările care utilizează MagazinID ca chei de căutare. Aceasta poate îmbunătăți performanța interogărilor care implică căutări sau sortări pe aceste coloane.

USE SistemGestionalBon

CREATE CLUSTERED INDEX nr\_magazin

on SistemGestionalBon.new\_schema.MagazinCopy (MagazinID)



Figura 7 – Clustered index

**Indexuri non-clustered:**

Un index non-clustered este o structură separată de date, care conține o listă de chei de index și adresele de memorie ale rândurilor corespunzătoare din tabelă. Aceste indexuri facilitează căutările rapide, dar datele rămân stocate într-un alt ordine decât cheia de index. Un tabel poate avea mai multe indexuri non-clustered.

Pentru a crea un index nonclustered, putem folosi următorul T-SQL

USE SistemGestionalBon

CREATE INDEX index\_non

on SistemGestionalBon.new\_schema.MagazinCopy (MagazinID, NumeMagazin)

Această comandă creează un index non-clustered pe tabelul MagazinCopy din schema new\_schema în baza de date SistemGestionalBon.

În final, putem observa că au fost creați doi indecși diferiți, unul clustered și unul non-clustered.



Figura 8 – Indecși creați

Pentru a șterge un idex, folosim următorul T-SQL:

USE SistemGestionalBon

go

drop index nr\_magazin on SistemGestionalBon.new\_schema.MagazinCopy

**Vederi**

În SQL, o vedere (view) reprezintă rezultatul unei interogări salvate sub formă de tabel virtual în baza de date. Practic, o vedere este o interogare SQL predefinită, stocată în baza de date, care poate fi apelată și utilizată ca un tabel obișnuit în alte interogări sau operațiuni.

Vederile pot fi de 3 tipuri:

1. Vederi standart
2. Vederi indexate
3. Vederi partiționate

Mai jos am realizat un exemplu de creare a unei cederi.

USE SistemGestionalBon

GO

CREATE VIEW new\_view AS

SELECT \* FROM new\_schema.MagazinCopy

where NumeMagazin not like '%t'

Astfel se v-a crea o vedere cu numele new\_view, are are la bază datele din tabela MagazinCopy doar că fără rândurile care au la sfârșitul numelui litera t.

**Concluzie**

Interogările T-SQL demonstrează crearea și gestionarea schemelor, acordarea de permisiuni, utilizarea sinonimelor și a indecșilor, precum și crearea de vederi într-o bază de date SQL Server. Aceste operațiuni sunt fundamentale pentru administrarea eficientă a datelor și pentru asigurarea securității și performanței în medii de bază de date complexe.