

Proiectarea si implementarea unei baze de date a unui magazin - VinylMag folosind SQL Server Management Studio

> Cânpean Eugen-Cristian Database Development Școala Informală de IT

Cuprins

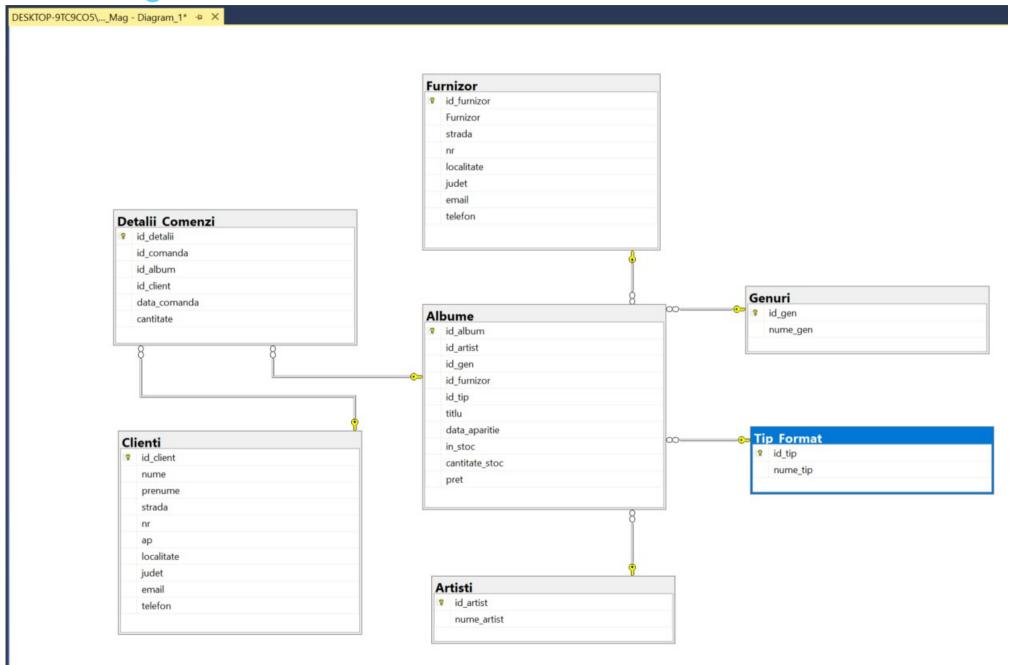
- Scopul Proiectului
- Diagrama bazei de date
- Crearea bazei de date
- Inserarea datelor
- Vizualizarea bazei de date
- Interogari ale bazei de date
- Procedură stocată
- Funcție
- View
- Trigger
- Job
- Concluzii

Proiectul urmărește realizarea unei baze de date a unui magazin care se ocupă cu vînzarea discurilor de vinyl, a cd-urilor si a casetelor audio.

Datele care sunt folosite oferă informații legate de:

- Stocul disponibil
- Albume, artişti, genuri muzicale, format muzical
- Furnizori
- Clienţi
- Comenzi

Diagrama bazei de date



Crearea bazei de date

```
create database Vinyl Mag;
use Vinyl Mag;
drop table if exists Artisti;
create table Artisti (
             int identity primary key not null,
id artist
nume artist Varchar(100)
);
drop table if exists Genuri;
create table Genuri (
id_gen int identity primary key not null,
nume_gen
             Varchar(50)
);
drop table if exists Tip Format;
create table Tip Format (
id tip int identity primary key not null,
nume tip
             Varchar(50)
);
```

```
drop table if exists Furnizor;
create table Furnizor (
id furnizor int identity primary key not null,
Furnizor
             Varchar(250),
strada Varchar(150),
      Varchar(150),
localitate
             Varchar(100),
judet Varchar(50),
email Varchar(150),
telefon
             Varchar(50),
);
drop table if exists Albume;
create table Albume(
id album
             int identity primary key not null,
id artist
             int REFERENCES artisti(id artist),
id gen int REFERENCES genuri(id gen),
id furnizor int REFERENCES furnizor(id furnizor),
id_tip int REFERENCES tip_format(id_tip),
```

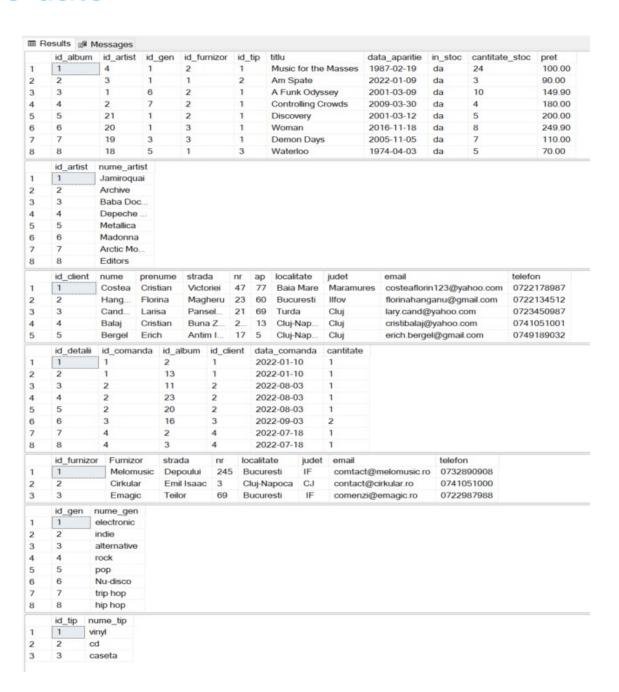
Inserarea datelor

```
insert into artisti values ('Jamiroquai');
insert into artisti values ('Archive');
insert into artisti values ('Baba Dochia');
insert into artisti values ('Depeche Mode');
insert into artisti values ('Metallica');
insert into artisti values ('Madonna');
insert into artisti values ('Arctic Monkeys');
insert into artisti values ('Editors');
insert into artisti values ('Partizan');
insert into artisti values ('Varna');
insert into artisti values ('Travka');
insert into artisti values ('Wax Tailor');
insert into artisti values ('Etienne de Crecy');
insert into artisti values ('Sebastien Tellier');
insert into artisti values ('Soulwax');
insert into artisti values ('Genesis');
```

```
insert into genuri values ('electronic');
insert into genuri values ('indie');
insert into genuri values ('alternative');
insert into genuri values ('rock');
insert into genuri values ('pop');
insert into genuri values ('Nu-disco');
insert into genuri values ('trip hop');
insert into genuri values ('hip hop');
insert into tip format values ('vinyl');
insert into tip format values ('cd');
insert into tip format values ('caseta');
insert into furnizor values ('Melomusic', 'Depoului', '245',
'Bucuresti', 'IF', 'comtact@melomusic.ro', '0732890908');
insert into furnizor values ('Cirkular', 'Emil Isaac', '3
'Clui-Napoca', 'CJ', 'contact@cirkular.ro', '0741051000');
insert into furnizor values ('Emagic', 'Teilor', '69', 'Bucuresti',
          ', 'comenzi@emagic.ro', '0722987988');
```

Vizualizarea bazei de date

```
select * from Albume
select * from Clienti
select * from Artisti
select * from Detalii_Comenzi
select * from Furnizor
select * from Genuri
select * from Tip_Format
```



Interogari ale bazei de date Funcții folosite: SUM, month

```
--suma pt fiecare comanda
select dc.id comanda, sum(a.pret*dc.cantitate)
 as suma comanda
from detalii comenzi dc
join albume a on dc.id album = a.id album
group by dc.id comanda
---suma comenzilor pt luna 08
select month(data comanda) as data comanda,
sum(a.pret*dc.cantitate) as suma comanda
from detalii comenzi dc
join albume a on dc.id album = a.id album
where month(data comanda) = '08'
group by month(data comanda)
```

⊞R	esults 🖟 Mess	sages	
	id_comanda	suma_comanda	
1	1	240.00	
2	2	309.90	
3	3	120.00	
4	4	449.90	
5	5	309.90	
6	6	80.00	
	data comand	la suma comanda	1
_			
1	8	699.80	

Interogari ale bazei de date

Funcții folosite: CONCAT, COUNT

```
---clientul cu cele mai multe comenzi
select top 1 concat(c.nume, ' ', c.prenume) as nume_client, count(distinct id_comanda) as nr_comenzi
from detalii_comenzi dc
join clienti c on dc.id_client = c.id_client
group by concat(c.nume, ' ', c.prenume)
order by nr comenzi desc
---comenzi provincie vs bucuresti
select case when localitate = 'Bucuresti' then localitate else 'Provincie' end as regiune,
count(distinct id_comanda) as numar_comenzi from detalii_comenzi dc
join clienti c on dc.id client = c.id client
group by case when localitate = 'Bucuresti' then localitate else 'Provincie' end;
 ■ Results Messages
       nume client
                          nr comenzi
 1
        Hanganu Florina
       regiune
                   numar comenzi
        Bucuresti
 2
        Provincie
                   4
```

Interogari ale bazei de date Elemente folosite: Select TOP

```
---cel mai vechi album
select top 1 nume artist, nume gen, titlu, data aparitie from albume a
join genuri g on a.id gen = g.id gen
join artisti ar on a.id artist = ar.id artist
order by data aparitie asc
---cel mai nou album
select top 1 nume_artist, nume_gen, titlu, data_aparitie from albume a
join genuri g on a.id gen = g.id gen
join artisti ar on a.id artist = ar.id artist
order by data aparitie desc
data aparitie
      nume artist
                   nume gen
                              titlu
      Abba
                               Waterloo
                                         1974-04-03
                   pop
 1
```

titlu

Midnights

nume gen

pop

data aparitie

2022-10-21

nume artist

Taylor Swift

Procedură stocată Elemente folosite: WITH

```
--- procedura cu parametru de intrare id comanda care returneaza pretul transportului
CREATE PROCEDURE PretTransport
@id_comanda int
AS
with pret per comanda as (
select dc.id_comanda, sum(a.pret*dc.cantitate) as suma_comanda
from detalii_comenzi dc
join albume a on dc.id_album = a.id_album
group by dc.id_comanda)
select id comanda, suma comanda,
case when suma_comanda >= 200 then 0
else 16 end pret_transport
from pret_per_comanda
where id_comanda = @id_comanda;
exec PretTransport @id comanda = 3
pret transport
       id comanda
                     suma comanda
       3
                      120.00
                                        16
 1
```

Funcție Elemente folosite: subquery

```
---functie care calculeaza totalul pt fiecare comanda (total albume+pret transport)
CREATE FUNCTION dbo.Test_functie (@id int)
RETURNS table AS
return(
select suma_comanda+
(case when suma comanda >= 200 then 0
else 16 end ) pret_total
from (
select dc.id_comanda, sum(a.pret*dc.cantitate) as suma_comanda
from detalii comenzi dc
join albume a on dc.id album = a.id album
group by dc.id_comanda) t
where id_comanda = @id)
SELECT *
FROM dbo.Test_functie(3)
                 Messages

    ■ Results

          pret_total
           136.00
```

View Elemente folosite: CASE WHEN; functiile concat, sum

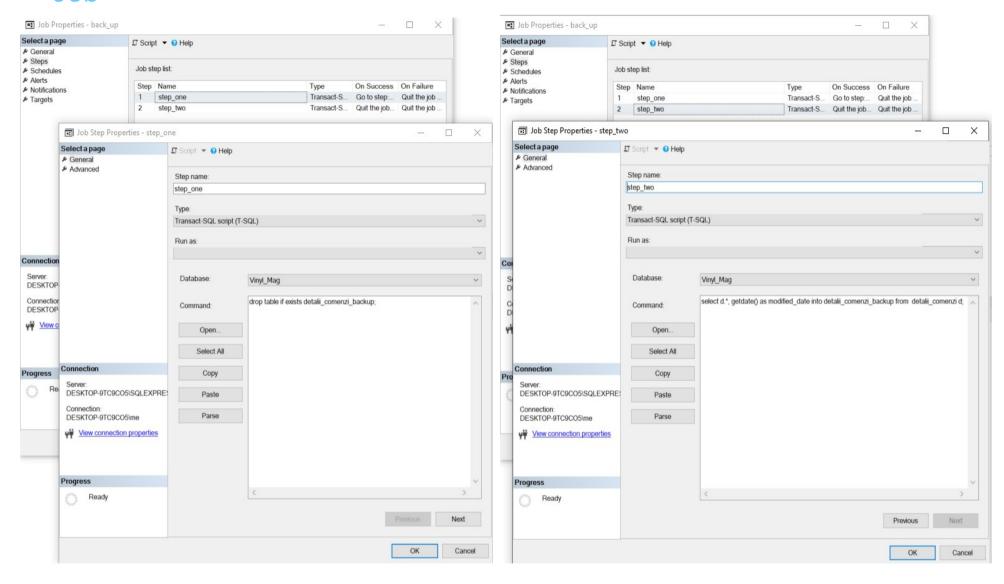
```
-- view care contine toate informatiile financiare necesare generarii unei facturi
-- id comanda, data comanda, nume, prenume client, cost transport, cost produse, cost total
create or alter view Detalii_financiare as
select dc.id_comanda,
dc.data comanda,
concat (c.nume, ' ', c.prenume) as nume_prenume,
sum(a.pret*dc.cantitate) as pret_albume,
case when sum(a.pret*dc.cantitate) >= 200 then 0
else 16 end as cost transport,
sum(a.pret*dc.cantitate) + (case when sum(a.pret*dc.cantitate) >= 200 then 0
else 16 end) as cost_total
from detalii_comenzi dc
join albume a on dc.id_album = a.id_album
join clienti c on dc.id_client = c.id_client
group by id_comanda, data_comanda, nume, prenume;
select * from Detalii_financiare
■ Deculte - ■ M
```

■ Results									
	id_comanda	data_comanda	nume_prenume	pret_albume	cost_transport	cost_total			
1	1	2022-01-10	Costea Cristian	240.00	0	240.00			
2	2	2022-08-03	Hanganu Florina	309.90	0	309.90			
3	3	2022-09-03	Candea Larisa	120.00	16	136.00			
4	4	2022-07-18	Balaj Cristian	449.90	0	449.90			
5	5	2022-08-08	Bergel Erich	309.90	0	309.90			
6	6	2022-08-08	Hanganu Florina	80.00	16	96.00			

Trigger - care impiedica introducerea in baza de date a unei comenzi care contine albume care nu se afla pe stoc sau a caror stoc este mai mic decat cantitatea comandata

```
create or alter trigger dbo.InsertComenzi
on dbo.detalii comenzi
after insert as
set nocount on;
if exists (select 1 from inserted i join Detalii Comenzi dc on i.id detalii = dc.id detalii
join albume a on dc.id album = a.id album
where (a.cantitate stoc = 0) or (i.cantitate > a.cantitate stoc))
begin
raiserror ('nu se poate comanda, nu este pe stoc', 16,1);
rollback transaction;
return
end
go
--comanda cu numar de albume indisponibile
insert into detalii comenzi values (7, 24, 2, '2023-01-12', 1);
-- comanda cu numar de albume mai mare decat stocul disponibil
insert into detalii_comenzi values (8, 23, 2, '2023-01-12', 4);
select * from Albume where id_album in (23,24);
Messages
   Msq 50000, Level 16, State 1, Procedure InsertComenzi, Line 12 [Batch Start Line 20]
   nu se poate comanda, nu este pe stoc
   Msg 3609, Level 16, State 1, Line 21
   The transaction ended in the trigger. The batch has been aborted.
```

Job



Concluzii

Baza de date creată oferă informații relevante despre vânzări, cumpărători, stocuri. Prin diverse interogări se pot afla toate informațiile generării unor facturi sau rapoarte de activitate. Am identificat următoarele îmbunătățiri posibile:

- Adăugarea coloanelor modified_date şi modified_by_user pentru a identifica cine şi când a efectuat ultima dată modificări;
- Job-ul creat face o copie zilnică a celui mai important tabel, ceea ce pentru tabele de dimensiuni mari ar putea afecta performanța. Am putea modifica acest job astfel încât doar ultimele modificări (incremental) să fie inserate în tabelul de back-up;
- > Datele personale care identifică clienții ar putea fi protejate/criptate.