

TEME PROPUSE PT. ANUL III UTM (SGBD)

Obs. Studenții vor alege sa facă un referat din cele propuse la punctul I (se pot cupla 2 studenți) sau o problemă(1 student) dintre cele prezentate la punctul II.

I. TEME DE CASĂ CU ELABORARE DE REFERATE

- 1) Monitorizarea performanțelor Microsoft SQL Server.
- 2) Importul-exportul de date pt. Microsoft SQL Server (utilizare **Sql Server Integration Services**)
- 3) Implementarea și gestionarea indecșilor pentru bazele de date Microsoft SQL Server.
- 4) Implementarea vederilor pentru bazele de date Microsoft SQL Server.
- 5) Implementarea procedurilor stocate pentru bazele de date Microsoft SQL Server.
- 6) Implementarea funcțiilor pentru bazele de date Microsoft SQL Server.
- 7) Implementarea triggerelor pentru bazele de date Microsoft SQL Server.
- 8) Optimizarea performanțelor interogărilor pentru bazele de date Microsoft SQL Server.
- 9) Definirea și coordonarea lucrului tranzacțional pentru bazele de date Microsoft SQL Server.
- 10) Implementarea securității bazelor de date Microsoft SQL Server.
- 11) Criptarea în bazele de date Microsoft SQL Server.
- 12) Replicarea datelor utilizând Microsoft SQL Server.
- 13) Salvarea/restaurarea bazelor de date utilizând Microsoft SQL Server.
- 14) Interogarea serverului Microsoft SQL Server utilizând standardul XML de prezentare a datelor.

- 15) Accesarea bazelor de date utilizând tehnologia Microsoft ADO.NET din aplicații Windows/ASP.NET.
- 16) Dezvoltarea de aplicații pentru accesarea datelor folosind Entity Framework (modelul conceptual, modelul de stocare a datelor și modelul de mapare a datelor).
- 17) Utilizarea limbajului de interogare LINQ pentru accesarea surselor de date în aplicații .NET(C#, VB.NET).

II. PROBLEME

1. Problema 1 -----Cărti

Carti	Autori	Edituri
Cod_ISBN(PK) Titlu Cantitate Data_expirare	Id_autor(PK) Nume Prenume Data_nasterii	Id_editura(PK) Nume_editura Adresa

Constrangeri: data_nasterii < data_sistem, cantitate >=0

- 1.Să se construiască baza de date Carti. Se va avea în vedere (la crearea celor trei tabele) că o carte poate fi publicată de mai multe edituri și că ea poate avea mai mulți autori.
- 2.Să se populeze baza de date cu cel puțin 3 intrări corespunzătoare cerințelor
- 3.Să se afișeze toate cărțile și toți autorii lor
- 4.Să se creeze o procedură stocată de căutare a cărților după codul ISBN
- 5.Să se creeze o procedură stocată de afișare a cărților detaliată
(date cărți, date autori, date edituri)
- 6.Să se creeze o vedere care să conțină date detaliate despre autorii care au o vârstă mai mare de 60 de ani (date autor, date cărți scrise, date edituri)
- 7.Să se creeze un trigger care să nu permită să se steargă o carte, publicată de o anumită editură, dacă nu s-a depășit data_expirare(data expirare) în acel moment
- 8.Să se facă un backup full al bazei de date

2. Problema 2 -----Produse

Produse	Comanda	Cumparator
Id_produș(PK) Nume_produș Cantitate_stoc	Id_comanda(PK) Data_cda { Cantitate Pret_u Achitat }	Id_cumparator(PK) Nume Prenume

Constrangeri : data_expirare >= data_sistem , achitat (de tip bit)

- 1.Sa se construiasca baza de date Produse. Se va avea în vedere (la crearea celor trei tabele) că un cumpărător poate face mai multe comenzi, iar o comandă poate conține unul sau mai multe produse. Acoladele {...} semnifică existența unuia sau a mai multor produse într-o comandă- o multime de produse cumpărate.
- 2.Sa se populeze baza de date cu cel puțin 3 intrari corespunzatoare cerintelor
- 3.Sa se afiseze toti cumparatorii, toate comenzile facute si toate produsele comandate
- 4.Sa se creeze o procedura stocata de cautare a produselor vandute dupa nume (partial)
- 5.Sa se creeze o procedura stocata de afisare a comenzilor (date comenzi, date produse) facute de un anumit cumparator(dupa ID-ul lui), ordonate crescator calendaristic.
- 6.Sa se creeze o vedere care sa contina date detaliate despre produsele neachitate (date comanda, date cumparator, date produs)
- 7.Sa se creeze un trigger care sa modifice cantitatea in tabela produse(cantitate_stoc) in momentul cand a fost achitat un produs într-o comandă(achitat=1) - la cumparare sau ulterior.
8. Să se facă un backup full al bazei de date.

3. Problema 3 -----Banca

Banca	Client	Imprumut
Id_banca (PK)	Id_client(PK)	Id_imprumut(PK)
Nume_banca	Nume	Suma
Adresa	Prenume	Data_eliberare
Nr_telefon	Adresa	Data_expirare
	Salariu	Rata_lunara

Constrangeri : data_eliberarii <= data_sistem, data_expirare > data_sistem, salariu >0

- 1.Sa se construiasca baza de date Banca. Se va avea în vedere (la crearea celor trei tabele) că un client poate face imprumuturi la una sau mai multe banci (tabela Imprumut va trebui sa reflecte acest lucru).
- 2.Sa se populeze baza de date cu cel puțin 3 intrari corespunzatoare cerintelor
- 3.Sa se afiseze toate bancile si toti clientii acestora printr-o vedere creata in acest scop
- 4.Sa se creeze o procedura stocata de cautare a bancilor dupa nume(partial)
- 5.Sa se creeze o procedura stocata de afisare a bancilor si clientilor acestora (date banca, date imprumuturi, date client)
- 6.Sa se creeze o vedere care sa contina date detaliate despre imprumuturile neexpirate (date banca, date client, date imprumut)
- 7.Sa se creeze un trigger care sa se declanseze la inserarea unui nou imprumut si care sa calculeze rata lunara a clientului cu o dobanda fixa de 0.5% pe luna
8. Să se facă un backup full al bazei de date.

4. Problema 4 ----- Banca

Banca	Client	Imprumut
Id_banca(PK) Nume_banca Adresa	Id_client(PK) Nume Prenume Adresa Salariu	Id_imprumut (PK) Suma_init Suma_restit Rata_lunara Data_expirare Data_eliberarii

Constrangeri : data_eliberarii <= data sistem, data_expirare > data sistem, salariu > 0

1. Sa se construiasca baza de date Banca. Se va avea în vedere (la crearea celor trei tabele) că un client poate face imprumuturi la una sau mai multe banci (tabela Imprumut va trebui sa reflecte acest lucru).
2. Sa se populeze baza de date cu cel puțin 3 intrari corespunzatoare cerintelor
3. Sa se afiseze toate bancile, ordonate dupa data deschiderii, si clientii acestora
4. Sa se creeze o procedura stocata de cautare a bancilor dupa nume (partial)
5. Sa se creeze o vedere care sa contina date detaliate despre banci si clientii acestora (date banca, date client, date imprumuturi)
6. Sa se creeze o procedura stocata care sa calculeze suma pe care o mai are de platit un anumit client al unei banci, pentru imprumuturile neexpirate si expirate (la cerere).
7. Sa se creeze un trigger care sa nu îmi permita sa sterg un imprumut daca data expirarii este mai mare decat cea sistem.
8. Să se facă un backup full al bazei de date.

5. Problema 5 -----Magazin

Magazin	Stoc_Produse	Vanzari
Id_magazin(PK) Nume Adresa	Id_produs (PK) Nume Pret Cantitate	Id_vanzare (PK) Data_vanzare { Pret_vanz Cant_vanz }

Constrangeri : data_vanzare <= data sistem , Pret > 0, Pret_vanz > 0 (pretul vanzarii, pret_vanz, este pretul la care un produs a fost vandut de un magazin, iar pretul produsului, pret, este pretul la care un produs a fost achizitionat - de tip money).

1. Sa se construiasca baza de date Magazin. Se va avea în vedere (la crearea celor trei tabele) că un magazin poate face mai multe vanzari, iar o vanzare poate conține unul sau mai multe produse – acelasi produs se poate vinde in mai multe magazine. Acoladele

{...} semnifică existența unuia sau a mai multor produse într-o vânzare- o multime de produse.

- 2.Sa se populeze baza de date cu cel puțin 3 intrari corespunzatoare cerintelor
- 3.Sa se afiseze toate produsele care nu s-au vandut in niciun magazin si cantitatea lor
- 4.Sa se creeze o procedura stocata de afisare a vanzarilor(data vanzarii, nume produs, cantitate) unui magazin dupa id produs, in ordinea ascendenta a datelor de vanzare
- 5.Sa se creeze o procedura stocata de afisare a vanzarilor detaliate (date despre vanzare, date despre produse si date despre magazin), in ordinea alfabetica a numelui de magazin
- 6.Sa se creeze o vedere care sa contina date detaliate despre produse si in ce magazine pot fi gasite
- 7.Sa se creeze un trigger care sa se declanseze la inserarea unui nou produs si care sa scumpeasca respectivul produs cu 0.5 %
8. Să se facă un backup full al bazei de date.

6. Problema 6 ----- Magazin

Magazin	Stoc_Produse	Vanzari
Id_magazin(PK) Nume Adresa	Id_produs (PK) Nume Pret Cantitate	Id_vanzare (PK) Data_vanzare { Pret_vanz Cant_vanz }

Constrangeri : data_vanzare <= data sistem , Pret > 0, Pret_vanz > 0 (pretul vanzarii, pret_vanz, este pretul la care un produs a fost vandut de un magazin, iar pretul produsului, pret, este pretul la care un produs a fost achizitionat - de tip money).

- 1.Sa se construiasca baza de date Magazin. Se va avea în vedere (la crearea celor trei tabele) că un magazin poate face mai multe vanzari, iar o vanzare poate conține unul sau mai multe produse – acelasi produs se poate vinde in mai multe magazine. Acoladele {...} semnifică existența unuia sau a mai multor produse într-o vânzare- o multime de produse.
- 2.Sa se populeze baza de date cu cel puțin 3 intrari corespunzatoare cerintelor
- 3.Sa se afiseze toate magazinele si toate produsele vandute de ele
- 4.Sa se creeze o procedura stocata de cautare a produselor vandute dupa numele magazinului
- 5.Sa se creeze o procedura stocata de afisare a vanzarilor detaliate (date despre vanzare, date despre produs si date despre magazin)
- 6.Sa se creeze o vedere care sa contina date detaliate despre produsele vandute de magazine, care au suferit modificari de pret
- 7.Sa se creeze un trigger care sa se declanseze la efectuarea unei vanzari si care sa modifice cantitatea in stoc a fiecarui produs din vanzarea respectiva
8. Să se facă un backup full al bazei de date.

Observație: *Tema de casă se va trimite profesorului in format electronic, până în ultima săptămână de curs a semestrului(săptămâna 15.01.18-19.01.18).*

(adresa e-mail: mihai_popes2000@yahoo.com)

NU SE MAI PRIMESC TEME DE CASĂ DUPĂ ACEEA!