BILETE EXAMEN

1. Ce este tranzactia? Ce incidente pot aparea relative la coerenta bazei de date?

O tranzacție este o unitate logică de lucru care constă dintr-o secvență de operațiuni de citire și/sau scriere asupra datelor. Tranzacțiile sunt utilizate pentru a asigura consistența și integritatea datelor în baza de date.

Incidente relative la coerenta bazei de date includ:

Concurență: Atunci când mai multe tranzacții încearcă să acceseze și să modifice aceleași date simultan, pot apărea conflicte și erori de coerență.

Blocare (Locking): Atunci când o tranzacție blochează resursele necesare pentru alte tranzacții, acest lucru poate duce la întârzieri sau blocări în sistem.

Eșecuri de sistem: Dacă un sistem se blochează sau se oprește în timpul unei tranzacții, poate exista un risc de pierdere a consistenței datelor.

Eșecuri de rețea: În cazul unei pierderi de conectivitate între server și clienți sau între servere, este posibil ca datele să nu fie sincronizate corespunzător, ceea ce poate duce la inconsistență.

1. Ce este o secventa si care sunt avantajele ei?
2. Ce sunt indecsii? Care sunt avantajele si dezavantajele lor?

Index – structură ordonată care permite:

• creşterea performanţelor ( localizarea directă, rapidă, a unui rând al unei tablele)

• asigurarea integrităţii datelor (unicitatea valorilor)

Operaţii:

- creare

- ştergere

- (actualizarea automată la operaţii de manipulare a datelor )

1. Se da urmatoarea baza de date relationala:

SOFER[Cods,Numes,Prens,Datan,Nr\_permis,Data\_permis,Nr\_curse]

AUTO[Coda,Numar\_inm,Marca,An\_inm,Capacitate,Nr\_culise]

TRASEU[Codt, Loc\_pl,Loc\_sos,Ora\_pl]

CURSA[Coda,Cods,Codt,Ora\_sos,Data]

1. Exprimati urmatoarea interogare intr-unul din limbajele studiate in Teoria bazelor de date si SQL.

Care sunt soferii care au facut ambele curse Galati-Tecuci si Tecuci-Galati in date de 22 mai 2011?

SELECT DISTINCT s.Numes

FROM SOFER s

JOIN CURSA c1 ON s.Cods = c1.Cods

JOIN CURSA c2 ON s.Cods = c2.Cods

JOIN TRASEU t1 ON c1.Codt = t1.Codt

JOIN TRASEU t2 ON c2.Codt = t2.Codt

WHERE (t1.Loc\_pl = 'Galati' AND t1.Loc\_sos = 'Tecuci' AND c1.Data = '2011-05-22')

AND (t2.Loc\_pl = 'Tecuci' AND t2.Loc\_sos = 'Galati' AND c2.Data = '2011-05-22');

SELECT DISTINCT Numes

FROM SOFER

WHERE Cods IN (

SELECT DISTINCT Cods

FROM CURSA

WHERE Data = '2011-05-22' AND Coda IN (

SELECT Codt

FROM TRASEU

WHERE Loc\_pl = 'Galati' AND Loc\_sos = 'Tecuci'

)

) AND Cods IN (

SELECT DISTINCT Cods

FROM CURSA

WHERE Data = '2011-05-22' AND Coda IN (

SELECT Codt

FROM TRASEU

WHERE Loc\_pl = 'Tecuci' AND Loc\_sos = 'Galati'

)

);

1. Creati o tabela virtuala care sa cuprinda toate cursele effectuate in luna mai 2011 care sa contina numele soferului, numele atuovechicului, data curesi,destinatia si ora sosirii.

SOFER[Cods,Numes,Prens,Datan,Nr\_permis,Data\_permis,Nr\_curse]

AUTO[Coda,Numar\_inm,Marca,An\_inm,Capacitate,Nr\_culise]

TRASEU[Codt, Loc\_pl,Loc\_sos,Ora\_pl]

CURSA[Coda,Cods,Codt,Ora\_sos,Data]

CREATE VIEW Curse\_Mai\_2011 AS

SELECT s.Numes AS Nume\_Sofer, a.Marca AS Nume\_Autovehicul, c.Data AS Data\_Cursei,

CONCAT(t.Loc\_pl, ' - ', t.Loc\_sos) AS Destinatie, c.Ora\_sos AS Ora\_Sosirii

FROM CURSA c

JOIN SOFER s ON c.Cods = s.Cods

JOIN AUTO a ON c.Coda = a.Coda

JOIN TRASEU t ON c.Codt = t.Codt

WHERE MONTH(c.Data) = 5 AND YEAR(c.Data) = 2011;

CREATE VIEW Curse\_Mai\_2011 AS

SELECT s.Numes AS Nume\_Sofer, a.Marca AS Nume\_Autovehicul, c.Data AS Data\_Cursei,

CONCAT(t.Loc\_pl, ' - ', t.Loc\_sos) AS Destinatie, c.Ora\_sos AS Ora\_Sosirii

FROM CURSA c

JOIN SOFER s ON c.Cods = s.Cods

JOIN AUTO a ON c.Coda = a.Coda

JOIN TRASEU t ON c.Codt = t.Codt

WHERE c.Data BETWEEN '2011-05-01' AND '2011-05-30';

1. Forumulati urmatoarea prelucrare ca script SQLPLUS: Sa se stearga toate autobuzele de o anumita marca indicate prin dialog (dar si toate cursele effectuate de aceastea!)

-- Setarea înregistrării de input/output pentru dialog

SET SERVEROUTPUT ON;

-- Declarațiile interogării pentru a obține marca de la utilizator

ACCEPT marca\_autobuzului CHAR PROMPT 'Introduceți marca autobuzului: ';

-- Declarațiile comenzilor pentru ștergerea datelor din tabele

BEGIN

-- Ștergerea datelor din tabela CURSA care corespund autobuzului specificat

DELETE FROM CURSA

WHERE Coda IN (SELECT Coda FROM AUTO WHERE Marca = '&marca\_autobuzului');

-- Ștergerea datelor din tabela AUTO pentru autobuzul specificat

DELETE FROM AUTO WHERE Marca = '&marca\_autobuzului';

-- Afișarea mesajului de confirmare

DBMS\_OUTPUT.PUT\_LINE('Toate autobuzele de marca ' || '&marca\_autobuzului' || ' și cursele asociate au fost șterse.');

END;

/

1. Creati in PLSQL o procedura stocata care calculeaza si afiseaza pentru un sofer (indicat prin nume si prenume) un autobuz (indicat prin numarul de inmatriculare), numarul de curse Galati-Tecuci effectuate de acel sofer cu acel autbuz.

CREATE OR REPLACE PROCEDURE NumarCurseGalatiTecuci (

nume\_sofer IN VARCHAR2,

prenume\_sofer IN VARCHAR2,

numar\_inmatriculare\_autobuz IN VARCHAR2

)

IS

numar\_curse NUMBER;

BEGIN

-- Calcularea numărului de curse Galati-Tecuci efectuate de șoferul cu autovehiculul specificat

SELECT COUNT(\*)

INTO numar\_curse

FROM CURSA c

JOIN SOFER s ON c.Cods = s.Cods

JOIN AUTO a ON c.Coda = a.Coda

JOIN TRASEU t ON c.Codt = t.Codt

WHERE s.Numes = nume\_sofer

AND s.Prens = prenume\_sofer

AND a.Numar\_inm = numar\_inmatriculare\_autobuz

AND t.Loc\_pl = 'Galati'

AND t.Loc\_sos = 'Tecuci';

-- Afișarea rezultatului

DBMS\_OUTPUT.PUT\_LINE('Numarul de curse Galati-Tecuci efectuate de ' || nume\_sofer || ' ' || prenume\_sofer || ' cu autobuzul ' || numar\_inmatriculare\_autobuz || ' este: ' || numar\_curse);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu există curse Galati-Tecuci efectuate de ' || nume\_sofer || ' ' || prenume\_sofer || ' cu autobuzul ' || numar\_inmatriculare\_autobuz);

END;

/

1. Scrieti un trigger in PL/SQL care sa actualizeze numarul de curse effectuate de soferi si autobuze, de cate ori se inregistreaza o cursa noua.

CREATE OR REPLACE TRIGGER ActualizareNumarCurse

AFTER INSERT ON CURSA

FOR EACH ROW

BEGIN

-- Actualizarea numărului de curse pentru șoferul implicat în cursă

UPDATE SOFER

SET Nr\_curse = Nr\_curse + 1

WHERE Cods = :NEW.Cods;

-- Actualizarea numărului de curse pentru autobuzul implicat în cursă

UPDATE AUTO

SET Nr\_culise = Nr\_culise + 1

WHERE Coda = :NEW.Coda;

END;

/

Bilet Nr.4

1. Ce sunt fisiere de comenzi SQL\*PLUS si ce categorii de comenzi sau instructiuni pot ele contine?

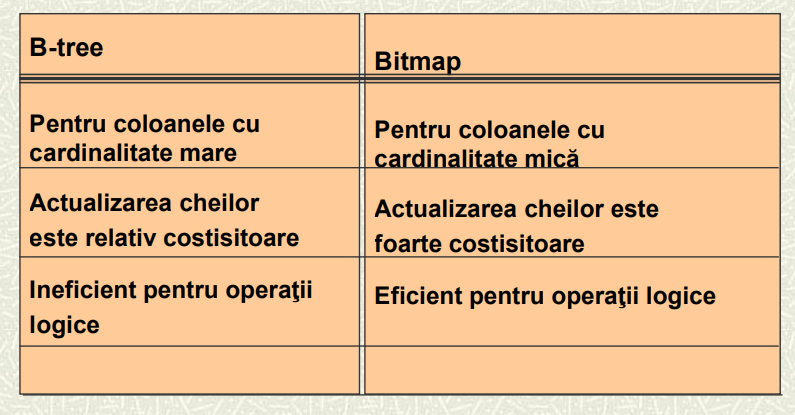
Fişier de comenzi (script) : - forma cea mai simplă de program

- secvenţă de comenzi - pot fi lansate din orice program de interfaţă în mod comandă;

1. Ce este tranzactia si ce este consistenta la citire?

O tranzacție este o unitate logică de lucru care este efectuată asupra unei baze de date.

1. Care este diferenta intre indecsii BITMAP SI indecsii B-Tree, cand se recomanda utilizarea fiecaruia dintre ei?



1. Se da urmatoarea baza de date relationala:

CLIENT [Codcl, Nume, Pren, Datan, Adresa]

CONT [Nrcont, Tipcont, Datacare, Sold\_la\_zi,Codcl]

Operatiune[Nrcont,Tipop,Data,Suma,Operator]

1. Exprimati urmatoarea interogare in SQL:

Care sunt conturile pentru care nu s-a facut operatiune tip “extragere”?

SELECT c.Nrcont

FROM CONT c

WHERE NOT EXISTS (

SELECT 1

FROM Operatiune o

WHERE o.Nrcont = c.Nrcont

AND o.Tipop = 'extragere'

);

SELECT c.Nrcont

FROM CONT c

LEFT JOIN Operatiune o ON c.Nrcont = o.Nrcont AND o.Tipop = 'extragere'

WHERE o.Nrcont IS NULL;

1. Creati o tabela virtuala care sa contina toate operatiunile effectuate pe conturi de tip <<la Vedere>>:numele clientului, data operatiei si numele operatorului CARDP[Numele,Prenume,Nrcont,Data\_op,Operator]

CREATE VIEW Operatiuni\_La\_Vedere AS

SELECT c.Nume AS Nume\_Client, c.Prenume AS Prenume\_Client, o.Data AS Data\_Operatiei, cp.Nume AS Nume\_Operator, cp.Prenume AS Prenume\_Operator

FROM Operatiune o

JOIN CONT co ON o.Nrcont = co.Nrcont

JOIN CLIENT c ON co.Codcl = c.Codcl

JOIN CARDP cp ON o.Operator = cp.Operator AND o.Data = cp.Data\_op

WHERE co.Tipcont = 'la Vedere';

1. Formualati urmatoarea prelucrare in SQL (sau SQLPLUS):Sa se stearga toate comenzie cu sold 0 ale unui client numele si prenumele sunt introduce prin dialog.

ACCEPT NumeClient CHAR PROMPT 'Introduceți numele clientului: '

ACCEPT PrenumeClient CHAR PROMPT 'Introduceți prenumele clientului: '

DELETE FROM Operatiune

WHERE Nrcont IN (

SELECT Nrcont

FROM CONT

WHERE Codcl = (

SELECT Codcl

FROM CLIENT

WHERE Nume = '&NumeClient' AND Pren = '&PrenumeClient'

)

)

AND Suma = 0;

ACCEPT NumeClient CHAR PROMPT 'Introduceți numele clientului: '

ACCEPT PrenumeClient CHAR PROMPT 'Introduceți prenumele clientului: '

DELETE o

FROM Operatiune o

JOIN CONT c ON o.Nrcont = c.Nrcont

JOIN CLIENT cl ON c.Codcl = cl.Codcl

WHERE cl.Nume = '&NumeClient' AND cl.Pren = '&PrenumeClient'

AND o.Sold = 0;

1. Scrieti un trigger in PL/SQL care sa actualizeze soldul unui cont, de cate ori se efectueaza o operatiune pe contruile de tipul operatinuii daca este de depunere sau extragere).

CREATE OR REPLACE TRIGGER ActualizareSoldCont

AFTER INSERT OR UPDATE ON Operatiune

FOR EACH ROW

DECLARE

v\_SumaOperatie NUMBER;

BEGIN

-- Verificăm dacă operația este de tipul "depunere" sau "extragere"

IF :NEW.Tipop IN ('depunere', 'extragere') THEN

-- Calculăm suma operației

v\_SumaOperatie := :NEW.Suma;

-- Actualizăm soldul contului

IF :NEW.Tipop = 'depunere' THEN

UPDATE CONT

SET Sold\_la\_zi = Sold\_la\_zi + v\_SumaOperatie

WHERE Nrcont = :NEW.Nrcont;

ELSIF :NEW.Tipop = 'extragere' THEN

UPDATE CONT

SET Sold\_la\_zi = Sold\_la\_zi - v\_SumaOperatie

WHERE Nrcont = :NEW.Nrcont;

END IF;

END IF;

END;

/

1. Concepeti un system de baze de date unde structura precedenta ocupa locul central. Descrieti etapele necesare ale sistemului, iar pentru fiecare etapa formulate exemple de comenzi si programe (valabile in sistemul Oracle) dintre cele ca….

CREATE TABLE CLIENT (

Codcl INT PRIMARY KEY,

Nume VARCHAR2(50),

Prenume VARCHAR2(50),

Datan DATE,

Adresa VARCHAR2(100)

);

CREATE TABLE CONT (

Nrcont INT PRIMARY KEY,

Tipcont VARCHAR2(50),

Datacare DATE,

Sold\_la\_zi NUMBER,

Codcl INT,

FOREIGN KEY (Codcl) REFERENCES CLIENT(Codcl)

);

CREATE TABLE Operatiune (

Nrcont INT,

Tipop VARCHAR2(50),

Data DATE,

Suma NUMBER,

Operator VARCHAR2(50),

FOREIGN KEY (Nrcont) REFERENCES CONT(Nrcont)

);

CREATE OR REPLACE PROCEDURE AdaugaClient (

v\_Codcl INT,

v\_Nume VARCHAR2,

v\_Prenume VARCHAR2,

v\_Datan DATE,

v\_Adresa VARCHAR2

) AS

BEGIN

INSERT INTO CLIENT (Codcl, Nume, Prenume, Datan, Adresa)

VALUES (v\_Codcl, v\_Nume, v\_Prenume, v\_Datan, v\_Adresa);

COMMIT;

END;

/

CREATE VIEW Operatiuni\_La\_Vedere AS

SELECT c.Nume AS Nume\_Client, c.Prenume AS Prenume\_Client, o.Data AS Data\_Operatiei, cp.Nume AS Nume\_Operator, cp.Prenume AS Prenume\_Operator

FROM Operatiune o

JOIN CONT co ON o.Nrcont = co.Nrcont

JOIN CARDP cp ON o.Operator = cp.Operator AND o.Data = cp.Data\_op

WHERE co.Tipcont = 'la Vedere';

Curs 4:

Sistem informatic :

- model al unui sistem real

- conţine reprezentări informatice ale obiectelor şi proceselor reale

- pentru a urmări evoluţia sistemului real, a lua decizii,..

- facilitează controlul şi coordonarea sistemului real;

Sistem cu baze de date :

- sistem integrat de date memorate şi organizate sub un control centralizat;

integrare: structuri de date intercorelate;

control centralizat : SGBD, cu ajutorul schemei.

Schema : - descrierea datelor, obiectelor şi a accesului la ele

SGBD : - ansamblu coordonat de programe ce permite descrierea, memorarea, manipularea, prelucrarea ansamblurilor de date ce constituie baze de date

Avantaje

coerenţă (se reduce redundanţa sau este strict controlată + restricţii de integritate)

independenţa datelor faţă de programe

accesibilitate (limbaje de interogare de nivel înalt)

partajarea datelor

protecţia datelor

Dezavantaje

costuri (hardware, software, instruire, personal)

probleme de securitate şi integritate (din partajarea datelor)

dificultatea modelării datelor

Obiective Performanţa la nivel de:

* Utilizare
* programare
* proiectare
* administrare

Se urmăreşte:

* minimizarea timpului de răspuns
* gestiunea optimă a spaţiului de stocare
* asigurarea integrităţii datelo

Index – structură ordonată care permite:

• creşterea performanţelor ( localizarea directă, rapidă, a unui rând al unei tablele)

• asigurarea integrităţii datelor (unicitatea valorilor)

Operaţii: - creare - ştergere - (actualizarea automată la operaţii de manipulare a datelor )

Clasificare indecşi

Clasificare din punct de vedere logic:

- după o coloană (single column) sau mai multe coloane (concatenated index)

- unic (Unique) sau neunic (nonUnique)

Clasificare din punct de vedere fizic:

- partiţionat sau nepartiţionat - de tip arbore binar (B-tree) sau bitmap

Crearea indecşilor:

• Automat Un index unique este creat automat când se defineşte o restricţie de integritate de tip PRIMARY KEY sau UNIQUE.

• Manual Utilizatorul poate crea indecşi nonunique pentru anumite coloane pentru a micşora timpul de acces la rânduri.

Utilitatea indecşilor Indecşii se creează atunci când:

• Coloana (coloanele) este (sunt) frecvent folosită(e) în clauze WHERE sau în condiţii de join.

• Coloana conţine o diversitate mare de valori. • Coloana conţine un număr mare de null-uri.

• Tabela este mare şi cererile întorc uzual cel mult 2-4% dintre rânduri.

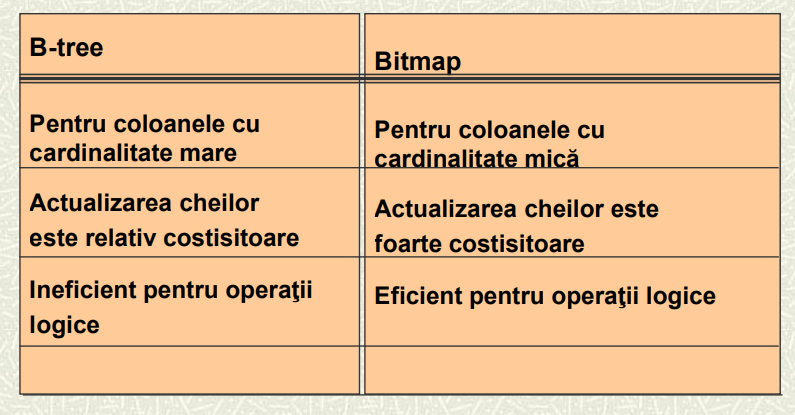
Indecşii NU se creează atunci când:

• Tabela este mică

• Coloanele nu sunt des folosite în interogări.

• Cele mai multe cereri întorc mai mult de 2–4% dintre rânduri.

• Tabela este actualizată frecvent.



Diferența principală dintre JOIN și INNER JOIN este că INNER JOIN returnează doar înregistrările care au o corespondență în ambele tabele, în timp ce simpla mențiune a JOIN poate fi considerată echivalentă cu INNER JOIN în majoritatea bazelor de date moderne.

Creaţi o tabelă virtuală, plecând de la tabelele cu angajaţi (emp, dept, salgrade), care sa contina toti angajatii de la Dallas, cu numele, salariul si treapta de salariu, avand ca nume de coloane

NAME, SAL, TS.

EMP [EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO]

DEPT [DEPTNO, DNAME, LOC]

SALGRADE [GRADE, LOSAL, HISAL]

CREATE VIEW Angajati\_Dallas AS

SELECT e.ENAME AS NAME, e.SAL AS SAL, s.GRADE AS TS

FROM EMP e

JOIN DEPT d ON e.DEPTNO = d.DEPTNO

JOIN SALGRADE s ON e.SAL BETWEEN s.LOSAL AND s.HISAL

WHERE d.LOC = 'Dallas';