

Lucrarea 6.

Utilizare vederi, Outer Join si variabile SQL+

Definire vederi View

Vederile definesc o macheta prin care se vad numai anumite coloane din BD.

O vedere se defineste printr-un select care poate contine unul sau mai multe operatii de JOIN.

Vederile sunt tratate ca niste tabele virtuale pentru care nu se alocă spațiu.

Select-ul prin care se definește o vedere se ține într-un rând din tabela All_Views ca o coloană

Text de tip LONG(echivalent cu tipul memo din dBase).

- Vederile
- simplifică scrierea interogărilor complexe
 - asigură posibilitatea de a acorda drepturi de acces pe vedere și nu pe tabele.
 - permite definirea unor tabele virtuale pentru aplicații

```
CREATE VIEW nume_view (lista_coloane) AS SELECT lista_col  
FROM tab1,tab2 WHERE cond [WITH READ ONLY];
```

```
CREATE VIEW Vemp (Nume,Salar,Departament) AS SELECT E.Ename Nume,  
E.Sal Salar , D.Dname Nume_dept  
FROM EMP E, DEPT D WHERE E.Deptno=D.Deptno;
```

```
SELECT * FROM Vemp WHERE salar>1200; -- Afiseaza tabela virtuala executand JOIN-  
ul.
```

Descr All_Views -- afiseaza structura tabelii cu descrierea vederilor.

Afisarea vederilor existente in BD se face din tabela All_Views unde definirea este un Select memorat in coloana Text de tip LONG.

```
SET LONG 200 -- seteaza lungime afisata pentru coloane de tipul LONG  
SELECT Owner, View_Name,Text FROM All_Views WHERE Owner= 'SCOTT' ;
```

Modificările și inserările(DML) sunt permise numai pentru View-rile simple:

- Definite pe o tabelă
- Fără funcții agregat sau clauza GROUP BY

Drepturile de acces pentru grupuri de utilizatori se recomandă să se dea pe vederi și nu pe tabele.

```
Create View Stud_AC AS Select * From stud Where cods='AC';  
Grant SELECT, INSERT, UPDATE, DELETE ON Stud_AC TO secr_ac;  
Grant [UPDATE|INSERT] (Nume,Adresa,Tel) on Stud TO Public;
```

Secretara de la AC are drepturi complete asupra studentilor de la AC.

Cămpurile Nume, Adresa, Tel din fișierul studenți pot fi citite de oricine dar nu modificate.

```
Select * From stud-ac Where cods='AC4'; -- Afiseaza studentii din anul 4 AC
```

Outer JOIN

Left Outer Join - ia toate inregistrarile din prima tabela si

- Le completeza coloanele din a doua tabela daca au corespondent conform conditiei de Join
- Coloanele din a doua tabela raman goale daca nu exista inregistrare corespondenta

Right Outer Join – lucreaza asemanator luind toate inregistrarile din tabela a doua.

In exemplul urmator se afiseaza toti salariatii dar care nu au telefon o sa aiba coloana TEL si IMP necompletate:

```
Select P.Nume, P.Adresa, P.Tel, T.IMP From Pers P, Telefon T  
Where P.Tel=+T.Tel;
```

Semnul + se pune inaintea coloanei din conditia de Join corespunzatoare tablei care este incompleta.

Self JOIN

Este JOIN-ul realizat in cadrul aceleiasi tabele. Un camp din tabela refera alt camp din aceiasi tabela. Se realizeaza deschizand aceeași tabela de 2 ori cu alias-uri diferite.

Afisare date salariatii, numele si salariul sefului folosind tabela EMP unde fiecare salariat are codul sefului (MGR), care este tot un salariat cu codul Empno.

```
SELECT E.Ename Nume, E.Sal Salar, S.Ename Nume_sef, S.Sal Salar_sef  
FROM EMP E, EMP S WHERE E.Mgr=S.Empno;
```

Select-uri imbricate

Consideram o BD universitara avand tabelele:

STUD(Cods, Nume, Adresa, Datan,Bursa) - Date personale studenti

NOTE(Cods,Codc,Nota) - tabela note cu cheia Cods,Codc

CURS(Codc,Titlu,Codp) - tabela cursuri cu referinta la tabela PROF prin Codp

PROF(Codp,Nume,Grad,Salar) - tabela de profesori

Afisare studenti care urmeaza cursurile profesorului 'POP':

```
SELECT Nume,Cods,Datan FROM Stud  
WHERE Cods=ANY (SELECT Cods FROM Note  
WHERE Codc=ANY (SELECT Codc FROM Curs  
WHERE Codp=ANY (SELECT Codp WHERE Nume='POP'))));
```

Operatorul ' =ANY' poate fi inlocuit cu **IN** si considera ca se iau toate valorile selectate.

Operatorul **ALL** se foloseste cand toate elementele selectate trebuie sa indeplineasca conditia.

Afisare salariatii care au salariul mai mare decat toate salariile medii pe departamente:

```
SELECT * FROM Emp WHERE  
sal > ALL (SELECT AVG(Sal) FROM EMP GROUP BY Deptno);
```

Selectul imbricat poate avea coloane multiple:

```
SELECT lista_coloane FROM tabela  
WHERE (col1,col2) IN (SELECT col1,col2 FROM tabela WHERE conditie);
```

Un Select poate apare si in clauza FROM:

Afisare persoane cu salarul > salarul mediu pe departamentul din care fac parte:

```
SELECT E.Ename Nume ,E.Sal Salr ,E.Deptno B.Salavg Salar_mediu  
FROM Emp E , (SELECT Deptno,Avg(sal) Salavg FROM Emp GROUP BY Deptno) B  
WHERE E.deptno=B.deptno AND E.sal>B.salavg;
```

Variabile SQL+

In SQL nu exista variabile.

In SQL+ se accepta definirea de variabile mai ales pentru dialog cu utilizatorul.

DEFINE var= valoare - creaza o variabila CHAR ce poate fi initializata

DEFINE var - afiseaza valoarea variabilei

DEFINE - afiseaza toate variabilele

UNDEFINE var - strge variabila

Variabilele se pot introduce si prin dialog in comanda ACCEPT:

ACCEPT vdept PROMPT ' Dati numele departamentuli: '

Sintaxa pentru commanda ACCEPT este:

```
NUMBER '99.999'  
ACCEPT var CHAR FORMAT 'An' PROMPT ' Mesaj' [Hide]  
DATE
```

In comenzile SQL pot apare variabile temporare care daca nu sun definite cer valoarea de la consola. Acest mod simplifica comenzile care pot fi parametrizate.

```
INSERT INTO Stud (Nume,Adresa,datan,Bursa)  
VALUES (&nume,&adresa,&datan,&bursa);  
'&nume','&adresa',
```

Dialogul va fi de forma:

Introduceti nume: 'POPESCU'

Introduceti Datan: ', '15-ian-1981'

Introduceti bursa: 800

Daca s-a dat '&nume' se poate introduce numele ca text fara apostroafe:

SET VERIFY OFF - nu se mai afiseaza vloare veche si noua a variabilei

Variabilele se pot introduce in lista de campuri sau conditii si vor fi cerute prin dialog:

```
SELECT &camp1,&camp2 FROM stud WHERE &cond;
```

Utilizare rapoarte

TTITLE 'text' ON|OFF - titlul de sus
BTITLE 'text' ON|OFF - titlul din partea de jos a paginii
SET PAGESIZE 40 - Numar de linii pe pagina
SET LINESIZE 120 - Numar de caractere pe pagina
SET LONG 200 - Numar de caractere afisate la tipul LONG
SET HEADING ON|OFF - afisare sau nu titlul raportului
SET COLSEP ' | ' - separatorul de coloane (implicit spatiu)
SHOW ALL - afisare toate setarile variabilelor sistem

Definirea coloanelor dintr-un raport:

COLUMN lista_optiuni

COLUMN	Nume	HEADING	'Nume Prenume	' FORMAT	A20	JUST	Center
COLUMN	Bursa		' Bursa '		L9.999		Left
COLUMN	Datan		'Data nasterii'				

Afisarea raportului cu Select:

SELECT Nume,Adresa,Datan,Bursa FROM Stud;

Setarile se fac inainte de Select si trebuie ca secventa sa fie memorata intr-un fisier.SQL

EDIT prog1 - editeaza un program in directorul curent (script SQL)

Lansarea unui script SQL se face cu:

START prog1