Lucrarea 5.

Controlul accesului la BD

Clasificare comenzi SQL:

SELECT - Cautare informatii in BD

Grup DML(Data Manipulation Language)

UPDATE - Modificare continut inregistrari dintr-o tabela
INSERT INTO - Adaugare inregistrari intr-o tabela
DELETE FROM - Stergere inregistrari dintr-o tabela

Grup DDL (Data Definition Language)

CREATE obiect - Creare structura tabela, index, verdere
ALTER obiect - Modificare caracteristici tabela,index,sesiune,user,roll,..
DROP obiect - Stergere tabela,index,vedere,trigger,secventa,..
RENAME obiect - Redenumire tabela sau vedere
TRUNCATE TABLE - Sterge toate inregistrarile dintr-o tabela

Grup Control transactii

COMMIT - Confirma terminare tranzactie
ROLLBACK - Reface toate modificarile BD de la inceputul tranzactiei (ultimul COMMIT)
SAVEPOINT - Creeaza un punct de reluare in tranzactie

Grup DCL (Data Control language)

GRANT - Acorda drepturi pentru un **user**
REVOKE - Retrage drepturi de la la un user

Descriere comenzi:

O comanda se poate scrie pe mai multe randuri si se termina cu **;**

CREATE TABLE tabela (col1 tip1, col2 tip2,.....); - definire campuri tabela
CREATE TABLE tabela **AS SELECT * FROM** tab2; - copiere alta tabela

CREATE INDEX nume_index ON tabela(col1,col2,..); -- creare index pentru o tabela
CREATE INDEX inume ON pers (Nume); -- creare index INUME pentru tabela PERS

ALTER TABLE tabela **ADD** (col1 tip1,col2 tip2,..);
ALTER TABLE tabela **MODIFY** (col1 tip1,col2 tip2,..);
ALTER TABLE tabela **DROP COLUMN** col1,col2;

DROP TABLE tabela; -- sterge tabela fara a salva nimic in fisierul Rollback

INSERT INTO tabela (col1,col2,col3,...) **VALUES** (val1,val2,val3,....);
INSERT INTO tabela (col1,col2,col3,...) **SELECT** lista_col **FROM** tab2;

UPDATE tabela **SET** col1=expr1, col2=expr2,...**WHERE** conditie;

DELETE FROM tabela **WHERE** conditie;

Pentru toate modificarile facute in BD prin Insert, Update si Delete inregistrarile vechi
se salveaza in **fisierul ROLLBACK.**

TRUNCATE TABLE tabela; - sterge toate inregistrarile fara a le salva in fisierul Rollback

RENAME nume_vechi TO nume_nou; -- redenumeste o tabela sau vedere

CREATE USER Radu IDENTIFIED BY abc123; -- creaza un user al BD fara drepturi de acces
DROP USER Radu; -- sterge user-ul Radu

SELECT * FROM All_Users; --afiseaza toti userii BD

Se defineste ca **Role** un grup de drepturi de acces pentru o **clasa de utilizatori**.
CREATE ROLE an4s [IDENTIFIED BY parola]; -- creaza un grup de drepturi de acces
DROP ROLE an4s -- sterge role-ul an4s

SELECT * FROM Role_Sys_Privs; -- afiseaza toate role-urile din BD
SELECT * FROM User_Role_Privs; -- afiseaza toate drepturile utilizatorilor din BD

Drepturile acordate utilizatorilor sunt **de sistem** sau **pentru obiecte**

Drepturile de sistem se refera la comenzile pe care un user le poate folosi in propria schema a BD.
Lista privilegiilor se poate atribui unui Role definit anterior sau direct utilizatorilor.
Unei liste de utilizatori i se pot atribui drepturile dintr-un Role+alte drepturi.

	nume_role		nume_role	
GRANT	privilegii_system	**TO**	lista_useri	[**WITH ADMIN OPTION**];
			PUBLIC	

- Acordarea unor drepturi pentru role an4s
GRANT Select, Create table,Update,Delete,Drop table,Create index TO an4s;
GRANT an4s TO Radu; -- acorda drepturile de acces din Role-ul an4s pentru Radu
Pentru a executa comanda GRANT trebuie sa ai acest drept.

Drepturile asupra obiectelor specifica operatiile pe care le poate face un user sau role asupra unei tabele sau obiect.

	ALL		nume_role		
GRANT	privilegii_obiect	**ON** obiect	**TO** user	[**WITH GRANT OPTION]**;	
			PUBLIC		

GRANT Select, Update, Insert, Delete, Alter, Index ON Pers to an4s;

Drepturile asupra obiectelor se pot da la nivel de coloana:
GRANT Select, Update (Adresa, Salar) ON Pers TO Radu;

Suprimarea unor drepturi de acces se face prin commanda:

	nume_role		nume_role	
REVOKE	lista_privilegii	[**ON** obiect]	**FROM**	lista_useri ;
			PUBLIC	

Baza de date conține un mare numar de obiecte (tabele,vederi,indexi, proceduri,package-uri) aparținând mai multor useri, in care user-ul SYSTEM conține tabelele de control ale BD.
Schema este o parte a BD care conține obiectele create de un user, asupra carora are toate drepturile.
Un user are dreptul de a accesa obiecte din alte scheme daca a primit aceste drepturi din partea administratorului BD (DBA) sau din partea user-ului care este proprietarul schemei.

Referirea la un obiect din alta schema decât cea curenta la care ne-am conectat trebuie sa specifice si numele proprietarului (user).

```
SELECT * FROM Scott.Emp;
```

Afişarea tabelelor utilizator si proprietarii lor se face din vederea All_Tables sau Tab:

```
SELECT Owner, Table_name FROM All_Tables WHERE Owner='SCOTT';
SELECT * FROM tab;
SELECT * FROM Col;          -- Afiseaza toate coloanele din tabelele utilizator
```

Utilizare funcții si expresii in SQL

In SQL se permite utilizarea expresiilor în lista comenzilor de interogare.
Aceste comenzi sunt legate de tabela la care se referă.

```
SELECT Ename Nume, Sal Salar, Comm Comision, Sal+NVL(Comm,0) Impozit FROM Emp;
SELECT ' Salut  ', Ename FROM Emp;       -- afişează un rând pentru fiecare înregistrare (N)
SELECT ' Salut '  FROM EMP;       -- este corect şi afişează N rânduri
```

Pentru calculul unor expresii independente de tabelele bazei de date există o tabelă virtuală Dual care se poate referi.

Nu există comandă corespunzătoare instrucțiunii de atribuire.

In expresii se pot referii operatorii şi funcţiile sistem pentru fiecare tip de operand.

Operatorii aritmetici respectă ordinea operațiilor din algebră.

** - ridicare la putere

* şi / - pentru înmulțire şi împărțire

+ şi - - pentru adunare şi scădere

<, <=, =, >=, >, <> - operatorii de relație (!= poate fi folosit pentru diferit de)

NOT, AND, OR - operatori logici pentru expresiile logice

Funcții standard pe tipuri

Type Error	Number	Character	Conversion	Date	Obj Ref	Misc
SQLCODE	ABS	ASCII	CHARTOROWID	ADD_MONTHS	DEREF	BFILENAME
SQLERRM	ACOS	CHR	CONVERT	LAST_DAY	REF	DECODE
	ASIN	CONCAT	HEXTORAW	MONTHS_BETWEEN	VALUE	DUMP
	ATAN	INITCAP	RAWTOHEX	NEW_TIME		EMPTY_BLOB
	ATAN2	INSTR	ROWIDTOCHAR	NEXT_DAY		EMPTY_CLOB
	CEIL	INSTRB	TO_CHAR	ROUND		GREATEST
	COS	LENGTH	TO_DATE	SYSDATE		LEAST
	COSH	LENGTHB	TO_MULTI_BYTE	TRUNC		NLS_CHARSET_DECL_LEN
	EXP	LOWER	TO_NUMBER			NLS_CHARSET_ID
	FLOOR	LPAD	TO_SINGLE_BYTE			NLS_CHARSET_NAME
	LN	LTRIM				NVL
	LOG	NLS_INITCAP				SYS_CONTEXT
	MOD	NLS_LOWER				SYS_GUID
	POWER	NLS_UPPER				UID
	ROUND	NLSSORT				USER
	SIGN	REPLACE				USERENV
	SIN	RPAD				VSIZE
	SINH	RTRIM				
	SQRT	SOUNDEX				
	TAN	SUBSTR				
	TANH	SUBSTRB				
	TRUNC	TRANSLATE				
		TRIM				
		UPPER				

Conversiile din numeric în caracter şi invers se fac implicit.

SELECT * FROM Emp WHERE Empno=7244; -- este echivalent cu Empno='7244'

SELECT ' SIN(X)= ', Sin(0.5) FROM DUAL;

Tipul Date indică un moment în timp: an-luna-zi-ora-minut-secunda

Pentru afişarea unui câmp dată trebuie folosită funcția de conversie dată-character cu format:

SELECT **TO_CHAR(Vdate, 'Format')** .... Format indică ce afişăm şi cum afişăm.

SELECT SYSDATE FROM DUAL; -- afişează data în formatul implicit 23-MAR-06

SELECT TO_CHAR(SYSDATE,'dd-month-mon-yyyy-year-day-hh24-mi-ss') FROM DUAL;

-- afişează 23-march -mar-2006-two thousand six-thursday -18-58-11

SELECT TO_CHAR(hiredate, 'dd-month-mon-yyyy-year-day-hh24-mi-ss')
FROM Emp WHERE Ename='SMITH';

-- afişează 17-december -dec-1980-nineteen eighty-wednesday-00-00-00

SELECT TO_CHAR(’30-05-1978’, ’dd-month-mon-yyyy-year-day-hh24-mi-ss’) FROM DUAL;
Dă eroare fiindcă ’30-05-1978’ este de tip char şi nu dată. Trebuie întâi transformat în dată:

SELECT TO_CHAR(**TO_DATE('30-05-1978','dd-mm-yyyy'),**
'dd-month-mon-yyyy-year-day-hh24-mi-ss') FROM DUAL;
-- afişează 30-may -may-1978-nineteen seventy-eight-tuesday -00-00-00

Formatul pentru numere este de forma ’9999.99’sau ’$9,999.99’, iar pentru Char ’AAAA’.
SELECT Ename Nume, TO_CHAR(sal,'$9,999.99') Salar FROM EMP;
NUME SALAR
---------- ----------
SMITH $800.00
ALLEN $1,600.00

SELECT Ename, 'SALAR='||SAL Salar FROM Emp; -- || concatenare si conversie implicita
ENAME SALAR
---------- ----------
SMITH SALAR=800
ALLEN SALAR=1600

Operaţii cu operanzi date
Date – Number → date
Date + Number → date
Date – Date → numar de zile
Round (Date,’Month’) → data rotunjita la prima zi din luna, cea mai apropiata
Trunc (Date,’Month’) → data rotunjita la prima zi din luna curenta
Last_day(Date) → ultima zi din luna
Next_Day(Date,’monday’) → urmatoare data pentru o zi de luni.

SELECT SYSDATE, Round(Sysdate,'month') Rotunjire,
Last_day(SYSDATE) Ultima_zi, Trunc(sysdate,'month') Trunc,
Next_Day(SysDate,'monday') Luni from dual;

SYSDATE ROTUNJIRE ULTIMA_ZI TRUNC LUNI
--------- --------- --------- --------- ---------
24-MAR-06 01-APR-06 31-MAR-06 01-MAR-06 27-MAR-06

SELECT Ename, SYSDATE-Hiredate Zile FROM Emp WHERE Ename=’SMITH’;

ENAME ZILE

SMITH 9227.82456 -- data1-data2 rezulta numar de zile de la angajare

In **clauza WHERE** se pot folosi pentru selecția înregistrărilor operatorii logici.

Logical Operators	
=	Equal to SELECT * FROM Emp **WHERE Job='ANALIST'**
!= or <>	Not equal to **WHERE Job!='ANALIST'**
>	Greater than **WHERE Sal > 2000**
>=	Greater than or equal to **WHERE Sal >= 2000**
<	Less than **WHERE Sal < 2000**
<=	Less than or equal to **WHERE Sal <= 2000**
in	Equal to any item in a list **WHERE Ename IN (Smith,Scott, King)**
not in	Not equal to any in a list **WHERE Ename NOT IN (Smith, King)**
between	Between two values, **WHERE sal Between 500 and 1000** greater than or equal to one and less than or equal to the other
not between	Not between two values **WHERE sal Not Between 500 and 1000**
Exists	Exists specified value **WHERE Ename EXISTS (SELECT….)**
is null	Is blank **WHERE Mgr IS NULL**
is not null	Is not blank **WHERE Mgr IS NOT NULL**
like	Like a specified pattern. % means any series of characters. _ means any single character. **WHERE Ename LIKE 'Pop%'**
not like	Not like a specified pattern. % means any series of characters. _ means any single char. **WHERE Ename NOT LIKE'Pop%'**

Funcția DECODE

Permite afişarea unor valori modificate ale unor câmpuri funcție de valoarea unui alt câmp. Funcția corespunde cu instrucțiunea IF – THEN – ELSE din limbajele de programare clasice. Se exemplifica pe cazul afişării unor noi salarii mărite diferențiat pe funcții (Job) afişate în coloana Salar_nou. Cei a căror funcție nu a fost specificată vor avea afişat salarul actual.

```
SELECT Ename Nume ,Job, sal Salar,
DECODE(job,'ANALYST',   sal*1.1,
                'CLERK',        sal*1.3,
                'MANAGER',  sal*1.2,
                                sal) Salar_nou FROM Emp;
```

```
NUME      JOB               SALAR  SALAR_NOU
---------- --------- ---------- - ----------
SMITH          CLERK              800             1040
ALLEN          SALESMAN       1600            1600
JONES          MANAGER        2975            3570
MARTIN         SALESMAN       1250            1250
BLAKE          MANAGER         2850            3420
CLARK          MANAGER         2450            2940
SCOTT          ANALYST         3000            3300
KING           PRESIDENT       5000            5000
```

Funcția DECODE poate fi folosită şi în comanda UPDATE la actualizarea salariului majorat.
Vom copia tabela Emp în Pers pentru a nu o modifica.
CREATE TABLE Pers AS SELECT * FROM Emp;

```
UPDATE pers
set sal=DECODE(job, 'ANALYST',  sal*1.1,
                    'CLERK', sal*1.3,
                    'MANAGER', sal*1.2,
                               sal)                    -- valoarea Default
```

```
NUME      FUNCTIA    SALAR_NOU
---------- --------- ----------
SMITH          CLERK             1040
ALLEN          SALESMAN          1600
WARD           SALESMAN          1250
JONES          MANAGER           3570
MARTIN         SALESMAN          1250
BLAKE           MANAGER           3420
CLARK           MANAGER           2940
SCOTT          ANALYST           3300
KING           PRESIDENT         5000
```