

CAPITOLUL 2

INTRODUCERE IN SQL

Acest capitol realizeaza o introducere in limbajul de interogare utilizat pentru a accesa o baza de date Oracle. Multimea comenzilor SQL sunt conturate ca blocuri de interogare de baza. In particular, noi discutam declaratiile SQL folosite la:

- executia calculelor
- specificarea alternativa a capetelor de coloana
- concatenarea coloanelor
- sortarea rindurilor
- introducerea criteriilor de cautare.

Privire de ansamblu asupra SQL

Un sistem de management al bazei de date necesita un limbaj de interogare pentru a permite utilizatorului sa acceseze datele. SQL (limbaj de interogare structurata) este limbajul utilizat de majoritatea sistemelor de baza de date relational.

Limbajul SQL a fost dezvoltat intr-un prototip de sistem de management a bazelor de date relationale - System R - de IBM la mijlocul anilor 1970. In 1979, Corporatia Oracle introduce prima implementare a SQL in varianta comerciala.

Trasaturi caracteristice SQL

- SQL este prezentat in limba engleza. Foloseste cuvintele select, insert, delete ca parti ale setului de comenzi.
- SQL este un limbaj neprocedural :specifica ce informatii doresti, nu cum sa le obtii. Cu alte cuvinte SQL nu iti cere sa specifici metoda de acces la date. Toate cererile SQL folosesc optimizarea cererilor- o parte a RDBMS- pentru a determina rapid remedierea datelor specifi- cate. Aceste trasaturi usureaza obtinerea rezultatelor dorite.
- Procesarea SQL asupra inregistrarilor nu se poate face decit asupra unei singure inregistrari la un moment dat. Cea mai comuna forma a unui set de inregistrari este un tabel.
- SQL poate fi folosit de un sir de utilizatori incluzand DBA, programatori de aplicatii , personal de management si multe alte tipuri de utilizatori.
- SQL pune la dispozitie comenzi pentru o varietate de tascuri incluzand:
 - date interogate
 - inserarea, extragerea si stergerea rindurilor intr-un tabel.
 - crearea, modificarea si stergerea obiectelor de tip baza de date
 - controlul accesului la baza de date si la obiectele de tip baza de date.
 - garantarea consistentei bazei de date

La inceput sistemele de management a bazei de date a utilizat un limbaj separat pentru fiecare categorie in parte.SQL le-a unificat pe toate acestea intr-un singur limbaj.

SQL a devenit un limbaj standard industrial pentru bazele de date relationale . Institutul National American de Standarde(ANSI) a adoptat SQL ca limbaj standard pentru RDBMS in anul 1986.Organizatia Internationala de Standarde (ISO) a adoptat deasemenea SQL ca limbaj standard pentru RDBMS.Toate RDBMS-urile suporta unele forme de SQL si toti vinzatorii de RDBMS intentioneaza sa se alinieze la standardele ANSI.

Setul de comenzi SQL

Comanda + Descriere:

SELECT

este comanda cea mai utilizata ; este folosita pentru obtinerea datelor din bazele de date

INSERT

aceste trei comenzi sunt utilizate pentru a introduce noi

UPDATE

rinduri, pentru a schimba rindurile existente si stergerea

DELETE

rindurilor nedorite din tabelele bazelor de date respective. (Ele sunt cunoscute in ansamblu ca DML sau comenzi ale limbajului de manipulare a datelor.)

CREATE

aceste trei comenzi sunt utilizate dinamic pentru a crea ,

ALTER

utiliza si sterge orice structura de date, de exemplu, tabele,

DROP

expuneri , indecsi. (Ele sunt cunoscute sub numele colectiv DDL sau comenzi ale limbajelor de definire a datelor).

GRANT

aceste doua comenzi sunt utilizate pentru a da sau a lua

REVOKE

drepturile de acces pentru bazele de date si structurile din Oracle.

N.B. Acestea sunt o parte a comenzilor SQL. Pentru o lista mai completa de comenzi se recomanda - Manualul de Referinta a Limbajului SQL- .

Scrierea comenzilor SQL

Cand scriem comenzi SQL , este important sa ne reamintim cateva reguli simple pentru construirea unor declaratii valide care sunt si usor de citit si de editat:

- Comenzile SQL pot fi pe una sau mai multe linii.
- Clauzele sunt uzual plasate pe linii separate.
- Tabelarea poate fi folosita.
- Cuvintele de comanda nu pot fi separate pe mai multe linii.

- Comenzile SQL nu sunt 'case sensitive'.
- O comanda SQL este introdusa la promptul SQL si liniile subsecventelor sunt numarate.
- O singura declaratie poate fi considerata curenta cat timp ea este in buffer si poate fi rulata intr-un numar de moduri :
 - plasand un punct si virgula(;) la sfarsitul ultimei clauze.
 - plasand un punct si virgula /slash la ultima linie in buffer.
 - plasand un slash(/) la promptul SQL.
 - scriind un R[UN] (comanda SQL) la promptul SQL.

Fiecare din urmatoarele declaratii sunt valide:

```
SELECT * FROM EMP;
```

```
SELECT
*
FROM
EMP
;
```

```
SELECT *
FROM EMP;
```

In acest manual comenzile SQL sunt sparte in clauze pentru claritate.

Blocul de interogare de baza

Declaratia SELECT regaseste informatia din baza de date implementand toti operatorii din algebra relationala .

In cele mai simple forme trebuie sa contina:

1. O clauza SELECT ,care listeaza coloanele pentru afisare astfel incat este esentiala o Proiectie.
2. O clauza FROM care specifica tabela implicata.

Pentru a lista toate numerele departamentelor,numele angajatilor si numarul managerilor in tabela EMP introduceti urmatoarele:

```
SELECT      DEPTNO,ENAME,MGR
FROM        EMP;
```

DEPTNO	ENAME	MGR
20	SMITH	7902
30	ALLEN	7698
30	WARD	7698
20	JONES	7839
30	MARTIN	7698
30	BLAKE	7839
10	CLARK	7839

```

20 SCOTT          7566
10 KING
30 TURNER        7698
20 ADAMS          7788
30 JAMES          7698
20 FORD           7566
10 MILLER        7782

```

De remarcat ca numele coloanelor sunt separate prin spatiu.

Este posibil sa selectam toate coloanele din tabela prin
specificarea
unui asterix ('*') dupa cuvantul SELECT .

```

SELECT      *
FROM        EMP;

```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	13-JUL-83	800.00		20
7499	ALLEN	SALESMAN	7698	15-AUG-83	1,600.00	300.00	30
7521	WARD	SALESMAN	7698	26-MAR-84	1,250.00	500.00	30
7566	JONES	MANAGER	7839	31-OCT-83	2,975.00		20
7654	MARTIN	SALESMAN	7698	05-DEC-83	1,250.00	1.400.00	30
7698	BLAKE	MANAGER	7839	11-JUN-84	2,850.00		30
7782	CLARK	MANAGER	7839	14-MAY-84	2,450.00		10
7788	SCOTT	ANALYST	7566	05-MAR-84	3,000.00		20
7839	KING	PRESIDENT		09-JUL-84	5,000.00		10
7844	TURNER	SALESMAN	7698	04-JUN-84	1,500.00	.00	30
7876	ADAMS	CLERK	7788	04-JUN-84	1,100.00		20
7900	JAMES	CLERK	7698	23-JUL-84	950.00		30
7902	FORD	ANALYST	7566	05-DEC-83	3,000.00		20

Alte elemente in clauza SELECT

Este posibil sa se includa si alte elemente in clauza SELECT.

- Expresii aritmetice
- Alias-uri de coloane
- Coloane concatenate
- Literal

Toate aceste optiuni ajuta utilizatorul sa ceara date si sa le manipuleze in functie de scopurile interogarii ; de exemplu,executia calculelor,legarea coloanelor impreuna,sau afisarea sirurilor de litere din text.

Expresii aritmetice

O expresie este o combinatie de una sau mai multe valori,operatori si functii care sa evalueaza la o valoare.

Expresiile aritmetice pot contine nume de coloane ,valori numerice constante si operatori aritmetici:

Operatori	Descriere
-----	-----
+	adunare
-	scadere
*	inmultire
/	impartire

```
SELECT  ENAME, SAL*12, COMM
FROM    EMP;
```

Daca expresia aritmetica contine mai mult decat un operator, prioritatile sunt *,/,la inceput,apoi +,- (de la stanga la dreapta pentru operatorii de aceeasi prioritate).

In urmatorul exemplu,inmultirea (250*12) este evaluata prima;apoi valoarea salariului este adunata la rezultatul multiplicarii(3000). Astfel pentru randul lui SMITH avem :800+3000=3800.

```
SELECT  ENAME, SAL + 250 * 12
FROM    EMP;
```

Parantezele pot fi utilizate pântru specificarea ordinii de executie a operatorilor ,daca,de exemplu ,adunarea e dorita inainte de inmultire:

```
SELECT  ENAME, (SAL + 250) * 12
FROM    EMP;
```

Aliasuri de coloana

Cand se afiseaza rezultatul unei interogari,SQL*Plus in mod normal utilizeaza numele coloanelor selectate ca titlu.In multe cazuri acest nume poate fi criptic sau fara inteles.

Puteti schimba un titlu de coloana utilizand un 'ALIAS'.

Un alias de coloana da unei coloane un nume de titlu alternativ la iesire.

Specificati aliasul dupa coloana in lista selectata.Prin default, titlurile alias vor fi fortate la litere mari si nu pot contine blankuri,numai daca aliasul este inclus intre ghilimele(" ").

Pentru a afisa titlul de coloana ANNSAL pentru salariul anual insemnand SAL*12,utilizati un alias de coloana:

```
SELECT  ENAME, SAL*12 ANNSAL, COMM
FROM    EMP;
```

Odata definit ,un alias poate fi utilizat de comenzile SQL care sunt tratate in capitolele 10 si 11.

Nota:

Intr-o declaratie SQL ,un alis de coloana poate fi utilizat numai in clauza SELECT.

Operatorul de concatenare

Operatorul de concatenare (||) permite coloanelor sa fie legate cu alte coloane,expresiilor aritmetice sau valorilor constante sa creeze o expresie de caractere.

Coloanele din cealalta parte a operatorului sunt combinate pentru a obtine o singura coloana.

Pentru a combina EMPNO si ENAME si sa se dea aliasul EMPLOYEE expresiei, introduceti:

```
SELECT EMPNO || ENAME EMPLOYEE
FROM EMP ;

EMPLOYEE
-----
7369SMITH
7499ALLEN
7521WARD
7566JONES
7654MARTIN
7698BLAKE
7782CLARK
7788SCOTT
7839KING
7844TURNER
7876ADAMS
7900JAMES
7902FORD
7934MILLER
```

Literali

Un literal este orice caracter ,expresie ,numar inclus in lista lui SELECT care nu este un nume de coloana sau un alias de coloana.

Un literal in lista lui SELECT este reprezentat de fiecare rand returnat la iesire.Sirurile de literali dintr-un text cu un format oarecare pot fi incluse in rezultatul interogarii si sunt tratate ca o coloana lista selectata.

Literalii de tip data calendaristica si caracter pot fi inchisi intre ghilimele simple(');literalii de tip numar nu au nevoie de ghilimele simple(').

Urmatoarea declaratie contine literali selectati prin concatenare si printr-un alias de coloana:

```
SELECT      EMPNO || '-' || ENAME  EMPLOYEE,
           'WORKS IN DEPARTMENT',
           DEPTNO
FROM        EMP;
```

EMPLOYEE	'WORKS IN DEPARTMENT'	DEPTNO
7369-SMITH	WORKS IN DEPARTMENT	20
7499-ALLEN	WORKS IN DEPARTMENT	30
7521-WARD	WORKS IN DEPARTMENT	30
7566-JONES	WORKS IN DEPARTMENT	20
7654-MARTIN	WORKS IN DEPARTMENT	30
7698-BLAKE	WORKS IN DEPARTMENT	30
7782-CLARK	WORKS IN DEPARTMENT	10
7788-SCOTT	WORKS IN DEPARTMENT	20
7839-KING	WORKS IN DEPARTMENT	10
7844-TURNER	WORKS IN DEPARTMENT	30
7876-ADAMS	WORKS IN DEPARTMENT	20
7900-JAMES	WORKS IN DEPARTMENT	30
7902-FORD	WORKS IN DEPARTMENT	20
7934-MILLER	WORKS IN DEPARTMENT	10

Tratarea valorilor nule

Daca unui rand ii lipseste o valoare pentru o anumita coloana ,despre acea valoare se spune ca este nula.

O valoare nula este o valoare care este sau incorecta,sau necunoscuta,sau inaplicabila.O valoare nula nu este la fel ca 'zero'.Zero este un numar.Valoarea nula ocupa un octet in reprezentarea interna.

Valoarea nula este tratata corect de catre SQL.

Daca orice valoare de coloana intr-o expresie este nula atunci rezultatul este nul.In urmatoarea declaratie numai Salesman are un rezultat al salariului:

```
SELECT      ENAME, SAL*12 + COMM ANNUAL_SAL
FROM        EMP;
```

ENAME	ANNUAL_SAL
SMITH	
ALLEN	19500
WARD	15500
JONES	
MARTIN	16400
BLAKE	
CLARK	

```

SCOTT
KING
TURNER          18000
ADAMS
JAMES
FORD
MILLER

```

Daca dorim sa obtinem un rezultat pentru toti angajatii,este necesar sa convertim valoarea nula la un numar . Noi folosim functia NVL pentru a converti o valoare nula la o valoare nenula.

Folositi functia NVL pentru a converti valoarea nula de la declaratia precedenta la 0.

```

SELECT      ENAME, SAL*12+NVL (COMM,0) ANNUAL_SAL
FROM        EMP;

```

```

ENAME      ANNUAL_SAL
-----
SMITH      9600
ALLEN      19500
WARD       15500
JONES      35700
MARTIN     16400
BLAKE      34200
CLARK      29400
SCOTT      36000
KING       60000
TURNER     18000
ADAMS      13200
JAMES      11400
FORD       36000
MILLER     15600

```

NVL asteapta doua argumente:

1. o expresie
2. o valoare nenula

De notat ca puteti folosi functia NVL pentru a converti un numar nul , data calendaristica sau sir de caractere la un alt numar , data calendaristica sau sir de caractere de aceiasi lungime si de acelasi tip de date asteptate.

```

NVL (DATECOLUMN, '01-JAN-88')

NVL (NUMBERCOLUMN, 9)

NVL (CHARCOLUMN, 'STRING')

```

Prevenirea selectiei rindurilor duplicate

Daca nu se indica altfel, SQL*Plus afiseaza rezultatul unei interogari fara eliminarea intrarilor duplicate .

Pentru a lista toate numerele de departament din tabela EMP, introduceti:

```
SELECT   DEPTNO
FROM     EMP;
```

```
DEPTNO
-----
      20
      30
      30
      20
      30
      30
      10
      20
      10
      30
      20
      30
      20
      10
```

Clauza DISTINCT

Pentru eliminarea valorilor duplicate din rezultat, includeti restrictia DISTINCT in comanda SELECT.

Pentru a elimina valorile duplicate afisate in exemplul urmator introduceti:

```
SELECT   DISTINCT   DEPTNO
FROM     EMP;
```

```
DEPTNO
-----
      10
      20
      30
```

Coloane multiple pot fi specificate dupa restrictia DISTINCT si restrictia DISTINCT afecteaza toate coloanele selectate.

Pentru a afisa valorile distincte ale lui DEPTNO si JOB, introduceti:

```
SELECT   DISTINCT DEPTNO, JOB
FROM     EMP;
```

```
DEPTNO   JOB
```

```

-----
10  CLERK
10  MANAGER
10  PRESIDENT
20  ANALYST
20  CLERK
20  MANAGER
30  CLERK
30  MANAGER
30  SALESMAN

```

Aceasta afiseaza o lista a tuturor combinatiilor diferite de ocupatie si numere de departamente.

De notat ca restrictia DISTINCT poate sa fie referita numai o singura data si trebuie sa urmeze imediat dupa cuvantul de comanda SELECT.

Clauza ORDER BY

In mod normal ordinea rindurilor intoarse in rezultatul unei cereri este nedefinita .Clauza ORDER BY poate fi utilizata pentru a sorta rindurile.

Daca o folosim, clauza ORDER BY trebuie sa fie intotdeauna ultima in declaratia SELECT.

Pentru a sorta dupa ENAME, introduceti:

```

SELECT  ENAME, JOB, SAL*12, DEPTNO
FROM    EMP
ORDER BY ENAME;

```

ENAME	JOB	SAL*12	DEPTNO
ADAMS	CLERK	13200	20
ALLEN	SALESMAN	19200	30
BLAKE	MANAGER	34200	30
CLARK	MANAGER	29400	10
FORD	ANALYST	36000	20
JAMES	CLERK	11400	30
JONES	MANAGER	35700	20
KING	PRESIDENT	60000	10
MARTIN	SALESMAN	15000	30
MILLER	CLERK	15600	10
SCOTT	ANALYST	36000	20
SMITH	CLERK	9600	20
TURNER	SALESMAN	18000	30
WARD	SALESMAN	15000	30

Ordonarea de default a datelor

Ordinea sortarii de default este ascendenta:

- Valorile numerice cele mai mici primele
- Valorile de tip date calendaristice cele mai mici primele.
- Valorile de tip caracter in ordinea alfabetica.

Inversarea ordinii de default

Pentru a inversa aceasta ordine cuvintul de comanda DESC este specificat dupa numele coloanei in clauza ORDER BY.

Pentru a inversa ordinea coloanei HIREDATE, deci datele cele mai tirzii sa fie afisate primele,introduceti:

```
SELECT  ENAME, JOB, HIREDATE
FROM    EMP
ORDER BY HIREDATE DESC;
```

ENAME	JOB	HIREDATE
JAMES	CLERK	23-JUL-84
KING	PRESIDENT	09-JUL-84
BLAKE	MANAGER	11-JUN-84
TURNER	SALESMAN	04-JUN-84
ADAMS	CLERK	04-JUN-84
CLARK	MANAGER	14-MAY-84
WARD	SALESMAN	26-MAR-84
SCOTT	ANALYST	05-MAR-84
MARTIN	SALESMAN	05-DEC-83
FORD	ANALYST	05-DEC-83
MILLER	CLERK	21-NOV-83
JONES	MANAGER	31-OCT-83
ALLEN	SALESMAN	15-AUG-83
SMITH	CLERK	13-JUN-83

Ordonarea dupa mai multe coloane

Este posibil sa se ordoneze dupa mai multe coloane.Limita este numarul de coloane din tabela.In clauza ORDER BY se specifica coloanele pentru ordonat separate prin virgula.Daca una sau toate coloanele trebuie sa fie inversate specificati DESC dupa fiecare coloana.

Pentru a ordona dupa doua coloane si afisa in ordinea inversa a salariului, introduceti:

```
SELECT  DEPTNO, JOB, ENAME
FROM    EMP
ORDER BY DEPTNO, SAL DESC;
```

DEPTNO	JOB	ENAME
10	PRESIDENT	KING

10	MANAGER	CLARK
10	CLERK	MILLER
20	ANALYST	SCOTT
20	ANALYST	FORD
20	MANAGER	JONES
20	CLERK	ADAMS
20	CLERK	SMITH
30	MANAGER	BLAKE
30	SALESMAN	ALLEN
30	SALESMAN	TURNER
30	SALESMAN	WARD
30	SALESMAN	MARTIN
30	CLERK	JAMES

Pentru a ordona dupa o coloana nu este necesar sa o avem SELECT-ata.

ORDER BY si valorile nule

In Oracle7,valorile nule sunt afisate ultimele pentru secventele ascendente si sint raportate primele cind rindurile sunt sortate in ordine descendenta.

Atentie:

Clauza ORDER BY este utilizata intr-o interogare cind se doreste sa se afiseze rindurile intr-o ordine specifica .Fara clauza ORDER BY rindurile sunt returnate intr-o ordine convenita de ORACLE si va trebui sa ne bazam pe el- ordinea determinata fiind consistenta de la cerere la cerere.De notat ca ordinea de afisare a rindurilor nu influenteaza ordinea interna a rindurilor asa cum sunt stocate in tabela.

Clauza WHERE

Clauza WHERE corespunde operatorului 'Restrictie' din algebra relationala.

Contine o conditie pe care rindurile trebuie sa o indeplineasca in ordinea afisarii lor.

Clauza WHERE ,daca este folosita , trebuie sa urmeze clauzei FROM :

```
SELECT      coloane
FROM        tabela
WHERE       anumite conditii sunt intilnite
```

Clauza WHERE poate compara valori in coloana ,valori literale,expresii aritmetice sau functii.

Clauza WHERE asteapta trei elemente:

1. Un nume de coloana
2. Un operator de comparatie
3. Un nume de coloana, constanta sau lista de valori.

Operatorii de comparatie sunt utilizati in clauza WHERE si pot fi impartiti in doua categorii: logici si SQL.

Operatorii logici

Acesti operatori verifica urmatoarele conditii:

Operator	Semnificatie
=	egal cu
>	mai mare decit
>=	mai mare sau egal
<	mai mic decit
<=	mai mic sau egal

Sirurile de caractere si datele calendaristice in clauza WHERE

Coloanele din ORACLE pot avea urmatoarele tipuri:caracter,numar sau data calendaristica.

Sirurile de caractere si datele calendaristice din clauza WHERE trebuie sa fie inchise in ghilimele simple('). Sirurile de caractere trebuie sa se supra- puna cu valoarea coloanei daca nu, trebuie modificate de o functie. Utilizati "Functii pe Caractere" din capitolul 4.

Pentru a afisa numele,numerele,ocupatia si departamentele tuturor functionarilor, introduceti:

```
SELECT      ENAME, EMPNO, JOB, DEPTNO
FROM        EMP
WHERE       JOB = 'CLERK';
```

ENAME	EMPNO	JOB	DEPTNO
SMITH	7369	CLERK	20
ADAMS	7876	CLERK	20
JAMES	7900	CLERK	30
MILLER	7934	CLERK	10

Pentru a gasi toate numele de departamente cu numerele de departament mai mare ca 20 ,introduceti:

```
SELECT      DNAME, DEPTNO
FROM        DEPT
WHERE       DEPTNO >20;
```

DNAME	DEPTNO
SALES	30

Compararea unei coloane cu alta in cadrul aceluiasi rand

Puteti compara o coloana cu o alta coloana in acelasi rand, la fel ca si cu o valoare constanta.

De exemplu ,presupunem ca dorim sa obtinem acei angajati al caror comision estã mai mare decat salariul lor:

```

SELECT      ENAME, SAL, COMM
FROM        EMP
WHERE       COMM > SAL;

```

ENAME	SAL	COMM
MARTIN	1,250.00	1,400.00

Operatori SQL

Sint patru operatori SQL care opereaza pe toate tipurile de date:

Operatori SQL

Operator	Semnificatie
BETWEEN..AND...	intre doua valori (inclusiv)
IN(list)	compara cu o lista de valori
LIKE	compara cu un model de tip caracter
IS NULL	estã o valoare nula

Operatorul BETWEEN

Realizeaza teste pentru valori intre, si inclusiv, o valoare minima si o valoare maxima.

Presupunind ca dorim sa vedem angajatii ai caror salariu este intre 1000 si 2000:

```

SELECT      ENAME, SAL
FROM        EMP
WHERE       SAL BETWEEN 1000 AND 2000;

```

ENAME	SAL
ALLEN	1,600.00
WARD	1,250.00
MARTIN	1,250.00
TURNER	1,500.00
ADAMS	1,100.00

MILLER 1,300.00

De notat ca valorile specificate sunt inclusive si ca limita minima trebuie specificata prima.

Operatorul IN

Testeaza valorile dintr-o lista specificata.

Presupunem ca dorim sa gasim angajatii care au unul din cele trei numere de marca(MGR):

```
SELECT EMPNO, ENAME, SAL, MGR
FROM EMP
WHERE MGR IN (7902,7566,7788);
```

EMPNO	ENAME	SAL	MGR
7369	SMITH	800.00	7902
7788	SCOTT	3,000.00	7566
7876	ADAMS	1,100.00	7788
7902	FORD	3,000.00	7566

Daca, caracterele sau datele calendaristice sunt utilizate,ele trebuie introduse intre ghilimele(' ').

Operatorul LIKE

Uneori nu se cunosc valorile exacte pe care le cautam.Utilizand operatorul LIKE este posibil sa selectam randurile care se potrivesc cu un model specificat de caractere.Operatia de petter-matching a caracterelor poate fi asemanata cu o cautare 'wild-card'.Doi simbolii se pot utiliza la construirea sirului de cautare.

Simbol	Reprezentare
%	orice secventa de zero sau mai multe caractere
-	un singur caracter oarecare

Pentru a lista toti angajatii al caror nume incepe cu un S,introduceti:

```
SELECT ENAME
FROM EMP
WHERE ENAME LIKE 'S%';
```

ENAME
SMITH
SCOTT

Caracterul '_' poate fi utilizat pentru cautarea unui anumit numar de caractere.

De exemplu pentru a lista toti angajatii care au un nume exact de patru caractere lungime:

```

SELECT  ENAME
FROM    EMP
WHERE   ENAME LIKE '____';

```

```

ENAME
-----
WARD
KING
FORD

```

Semnele '%' si '_' pot fi utilizate in orice combinatie de caractere.

Operatorul IS NULL

Operatorul IS NULL face teste specifice pentru valorile care sunt NULL.

```

SELECT  ENAME, MGR
FROM    EMP
WHERE   MGR IS NULL;

```

```

ENAME    MGR
-----
KING

```

Negarea expresiilor

Urmatorii operatori fac teste de negatie:

Operator	Descriere
!=	diferit de (VAX, UNIX, PC)
^=	diferit de (IBM)
<>	diferit de (toate o/s)
NOT NUMECOL=	diferit de
NOT NUMECOL>	mai mic sau egal

Operatori SQL

Operator	Descriere
NOT BETWEEN	nu se afla intre doua valori date
NOT IN	nu se afla intr-o lista data de valori
NOT LIKE	diferit de sirul
IS NOT NULL	nu este o valoare nula

Pentru a gasi angajatii ai caror salariu nu este intr-un interval, introdu- ceti:

```

SELECT  ENAME, SAL
FROM    EMP
WHERE   SAL NOT BETWEEN 1000 AND 2000;

```

```

ENAME          SAL
-----
SMITH          800.00
JONES         2,975.00
BLAKE         2,850.00
CLARK         2,450.00
SCOTT         3,000.00

```

KING	5,000.00
JAMES	950.00
FORD	3,000.00

Pentru a afla acei angajati a caror meserie nu incepe cu M, introduceti:

```

SELECT  ENAME, JOB
FROM    EMP
WHERE   JOB NOT LIKE 'M%';

ENAME   JOB
-----  -
SMITH   CLERK
ALLEN   SALESMAN
WARD    SALESMAN
MARTIN  SALESMAN
SCOTT   ANALYST
KING    PRESIDENT
TURNER  SALESMAN
ADAMS   CLERK
JAMES   CLERK
FORD    ANALYST
MILLER  CLERK

```

Pentru a afla toti angajatii care au un manager(MGR), introduceti:

```

SELECT  ENAME, MGR
FROM    EMP
WHERE   MGR IS NOT NULL;

ENAME   MGR
-----  -
SMITH   7902
ALLEN   7698
WARD    7698
JONES   7839
MARTIN  7698
BLAKE   7839
CLARK   7839
SCOTT   7566
TURNER  7698
ADAMS   7788
JAMES   7698
FORD    7566
MILLER  7782

```

Nota:

Daca o valoare NULL este utilizata intr-o comparatie ,atunci operatorul de comparatie trebuie sa fie IS sau IS NOT NULL. Daca acesti operatori nu sunt utilizati si valoarea NULL este comparata,atunci rezultatul este intotdeauna FALSE

De exemplu, COMM!=NULL este intotdeauna falsa.Rezultatul este fals deoarece o valoare NULL poate sa nu fie egala sau diferita cu orice alta valoare alta decat NULL.

De notat ca o astfel de eroare nu este semnalata,rezultatul fiind intotdeauna fals.

Interogarea datelor cu conditii multiple

Operatorii AND sau OR pot fi utilizati pentru a compune expresii logice.

Predicatul AND este adevarat numai daca ambele conditii sunt 'adevarate'; predicatul OR este adevarat daca cel puțin una din conditii este 'adevarata'.

In urmatoarele doua exemple,conditiile sunt aceleasi,dar predicatelor difera Priviti cum rezultatul este dramatic modificat.

Pentru a gasi toti functionarii care castiga intre 1000 si 2000,introduceti:

```
SELECT EMPNO,ENAME, JOB, SAL
FROM EMP
WHERE SAL BETWEEN 1000 AND 2000
AND JOB = 'CLERK';
```

EMPNO	ENAME	JOB	SAL
7876	ADAMS	CLERK	1,100.00
7934	MILLER	CLERK	1,300.00

Pentru a afla toti angajatii care sunt si functionari si/sau functionari care castiga intre 1000 si 2000,introduceti:

```
SELECT EMPNO,ENAME, JOB, SAL
FROM EMP
WHERE SAL BETWEEN 1000 AND 2000
OR JOB = 'CLERK';
```

EMPNO	ENAME	JOB	SAL
7369	SMITH	CLERK	800.00
7499	ALLEN	SALESMAN	1,600.00
7521	WARD	SALESMAN	1,250.00
7654	MARTIN	SALESMAN	1,250.00
7844	TURNER	SALESMAN	1,500.00
7876	ADAMS	CLERK	1,100.00
7900	JAMES	CLERK	950.00
7934	MILLER	CLERK	1,300.00

Puteti combina AND sau OR in aceiasi expresie logica. Cand AND sau OR apar in aceiasi clauza WHERE, toti operatorii AND sunt evaluati mai intai si apoi toti operatorii OR. Vom spune ca operatorii AND au o precedenta mai mare decat OR.

Deoarece AND are o precedenta mai mare decat OR urmatoarea declaratie SQL intoarce toti managerii cu salarii peste 1500\$ si toti vanzatorii.

```
SELECT EMPNO,ENAME, JOB, SAL, DEPTNO
FROM EMP
```

```

WHERE SAL > 1500
AND JOB = 'MANAGER'
OR JOB = 'SALESMAN';

```

EMPNO	ENAME	JOB	SAL	DEPTNO
7499	ALLEN	SALESMAN	1,600.00	30
7521	WARD	SALESMAN	1,250.00	30
7566	JONES	MANAGER	2,975.00	20
7654	MARTIN	SALESMAN	1,250.00	30
7698	BLAKE	MANAGER	2,850.00	30
7782	CLARK	MANAGER	2,450.00	10
7844	TURNER	SALESMAN	1,500.00	30

Daca doriti sa selectati toti managerii si vanzatorii cu salarii peste 1500\$ puteti introduce:

```

SELECT EMPNO, ENAME, JOB, SAL, DEPTNO
FROM EMP
WHERE SAL > 1500
AND (JOB = 'MANAGER'
OR JOB = 'SALESMAN');

```

EMPNO	ENAME	JOB	SAL	DEPTNO
7499	ALLEN	SALESMAN	1,600.00	30
7566	JONES	MANAGER	2,975.00	20
7698	BLAKE	MANAGER	2,850.00	30
7782	CLARK	MANAGER	2,450.00	10

Parantezele specifica ordinea in care operatorii vor fi evaluati. In al doilea exemplu operatorul OR este evaluat inaintea operatorului AND.

TIPURI DE DATE CARACTER SI CONDITII

```

WHERE ENAME = 'SCOTT'

```

ENAME (VARCHAR2)
SCOTT V V
SCOTT
MILLER
~~~~ ~ ~ ~

Oracle **nu face** umplerea cu blancuri la compararea cu coloanele VARCHAR2

ENAME (CHAR)
SCOTT
MILLER

```

WHERE ENAME = 'SCOTT'
| SCOTT V V|
|      . . .|
| SCOTT|V V|
|      |. |
| MILLER| V|
~~~~ ~~~~~ ~

```

Oracle **face** umplerea cu blancuri la compararea cu coloanele CHAR.

## Tipurile de date caracter si conditii

Tipurile de baza ale datelor stocate intr-o tabela oracle sunt:caracter, valoare numerica sau data calendaristica.Vom discuta toate variantele in detaliu mai tarziu.De cate ori rezultatele unei conditii implica date de tip caracter, acestea pot varia in functie de tipul coloanei;ORACLE inzestreaza coloaneta cu tipul CHAR pentru valori de lungime fixa si cu tipul VARCHAR2 pentru valori de lungime variabila.

Pentru coloanele cu tipul VARCHAR2 ,Oracle nu umple sirul de comparare si de aceea va face o potrivire exacta.In primul exemplu,doar un singur rand este intors pentru conditia:

```
WHERE ENAME = 'SCOTT'
```

cand un alt rand stocat in coloana ENAME are mai multe caractere decat sirul de comparat.

Pentru coloanele cu tipul CHAR ,oricum,Oracle face umplere cand valorile coloanelor sunt initial stocate,facandu-le pe toate de aceiasi lungime.

Aceiasi conditie va intoarce ambele randuri pentru SCOTT ,indiferent de cate spatii de sfarsit au fost adaugate cand valorile au fost stocate in tabela.

Oracle umple cu blancuri sirul de comparat in cel deal doilea caz si de aceea spatiile stocate sunt nesemnificative.

## Precedenta operatorilor

Toti operatorii sunt aranjati intr-o ierarhie ceea ce le determina precedenta .Intr-o expresie operatiile sunt executate in ordinea precedentei lor de la mare la mica.

Cand operatorii au precedenta egala atunci ei se evalueaza de la stanga la dreapta.

1. Toti operatorii de comparatie si SQL au precedenta egala:  
=,!=,<,>,<=,>=, BETWEEN...AND,IN,LIKE,IS NULL.
2. NOT(pentru a inversa rezultatul unei expresii logice.De ex: WHERE not(sal>2000))
3. AND
4. OR.

De fiecare data cand sunteti in dubiu despre care dintre doua operatii vor fi executate mai intai cand o expresie este evaluata, sunteti liberi sa utilizati parantezele pentru a clarifica semnificatia dorita si pentru a va asigura ca SQL*Plus face ceea ce doriti.

Sa presupunem ca doriti sa gasiti toti managerii, din orice departament, si toti functionarii din departamentul 10:

```
SELECT *
FROM EMP
WHERE JOB='MANAGER' OR (JOB = 'CLERK' AND DEPTNO = 10);
```

Parantezele de deasupra sunt necesare, AND are o precedenta mai mare decat OR, dar ele clarifica semnificatia expresiei.

## SELECT-Sumar

Urmatoarele clauze sunt inchise in comanda SELECT:

```
SELECT [DISTINCT] [* ,coloana alias],...]
FROM tabela
WHERE conditie (ii)
ORDER BY [coloana, expr] [ASC/DESC];
```

### SELECT

selecteaza cel putin o coloana

### Alias

poate fi folosit pentru coloanele din lista selectata

*

desemneza toate coloanele

### DISTINCT

poate fi utilizat pentru eliminarea duplicatelor

### FROM Tabela

desemneaza tabela din care provin coloanele

### WHERE

restrictioneaza cererea la randurile care indeplinesc o conditie. Poate contine valori de coloane, expresii si literali

### AND/OR

poate fi utilizat intr-o clauza WHERE pentru a construi conditii mai complexe. AND are prioritate peste OR.

()

pot fi utilizate pentru a forta prioritatea

### ORDER BY

intotdeauna apare la sfarsit. Specifica ordinea de sortare. Una sau mai multe coloane pot fi specificate aici.

### ASC

ordinea ascendenta este ordinea de sortare (implicita) si nu trebuie specificat.

DESC

inverseaza ordinea de sortare de default si trebuie specificat dupa un nume de coloana.

Clauzele pot fi introduse pe linii separate in buffer si tabelarea este utilizata pentru claritate si in editare.

## **Logarea la SQL*Plus**

Aceasta sectiune explica cum sa ne logam la SQL*Plus si liniile de iesire ale tipurilor de comenzi ce pot rezulta in cadrul sau.

### **SQL*Plus**

SQL*Plus este un program scris de Corporatia Oracle, ce produce un mediu pentru comenzile SQL ce pot fi tastate direct sau rulate dintr-un fisier de comanda. In plus ,comenzile SQL pot fi derivate. Ele sunt folosite pentru:

- formatarea rezultatelor
- setarea optiunilor
- editarea si stocarea declaratiilor SQL

Odata ce v-ati logat la sistemul vostru de operare, sunt 3 moduri pentru a va loga la SQL*Plus:

1. SQLPLUS
  - o Vetii vedea dupa aceasta un mesaj ca acesta:
  - o SQL*Plus: Version 3.1.1 Production on Mon Oct 4 1993 Copyright (c) 1992, Oracle Corporation, California, USA. All rights reserved. Enter Username:
  - o Introduceti numele vostru de user si apasati RETURN: SQL*Plus va afisa promptul : "Enter Password:". Introduceti parola voastra si apasati din nou RETURN.
  - o Pentru protectia voastra ,parola nu va apare pe ecran. SQL*Plus va afisa promptul sau: SQL>
  - o Aceasta indica linia de comanda .Acolo sunt doua feluri de comenzi si puteti introduce pe aceasta linie de comanda: comenzi SQL sau comenzi SQL*Plus.
2. SQLPLUS username
  - o Vetii fi indemnati sa va dati parola.
3. SQLPLUS username/password
  - o Vetii fi logati la SQL*Plus. In acest caz parola va fi afisata.

Alegeti metoda pe care o preferati.

## **Editarea declaratiilor SQL utilizand comenzi SQL*Plus**

1. Cand veti introduce o comanda SQL,aceasta este stocata intr-o zona de memorie utilizata de buferul SQL si ramane acolo pana ce veti introduce o noua comanda.
2. Daca apasati [RETURN] inainte de a completa o comanda,SQL*Plus va afisa un numar de linie.
3. Terminatorul pentru declaratiile SQL este un ';':
4. Cat timp declaratia SQL este in bufer,sunt cateva operatii de editare directe care pot fi executate utilizand comenzi SQL*Plus:

Comanda	Abrevierea	ROL
APPEND text	A text	adauga 'text' la sfarsitul liniei curente.
CHANGE	C/old/new	schimba vechiul text cu noul text in linia
CHANGE	C/text/	curenta. sterge 'text'-ul din linia curenta
CLEAR BUFFER	CL BUFF	sterge toate liniile din buferul SQL.
DEL		sterge linia curenta.
INPUT	I	insereaza un numar nedefinit de linii.
INPUT 'text'-ul	I text	insereaza o linie constituita din respectiv
LIST	L	listeaza toate liniile din buferul SQL.
LIST n	Ln	listeaza o linie (specificata de n)
LIST m, n	L m n	listeaza un numar de linii(de la m la n).
RUN	R	afiseaza si executa comanda SQL curenta din
/		buffer. executa comanda SQL care este
		bufer.

## Diversitatea comenzilor SQL*Plus

Comenzile SQL (precum SELECT) sunt mijloace de acces la date prin kernelul Oracle.Comenzile SQL sunt utilizate in special pentru controlul mediului, formatarea rezultatelor interogarilor si controlul fisierelor.Comenzile identificate aici sunt amestecate si trebuie sa le folositi in urmatoarele exercitii.

Comenzile SQL sunt introduse la promptul SQL> pe o linie ,ele nu vor deschide un buffer.

Comanda + Descriere

SAVE numefis

permite salvarea intr-un fisier a continutului buferului SQL.

GET numefis  
incarca continutul unui fisier salvat in prealabil in buffer.

START numefis  
ruleaza un fisier de comanda salvat in prealabil. Fisiererele de comanda sunt tratate in capitolul 10.

ED numefis  
utilizeaza un editor de default pentru a edita continutul unui fisier salvat.

EXIT  
paraseste SQL*Plus.

## Capitolul 2 Exerciitii-Introducere in SQL

Aceste exercitii intentioneaza sa dezvaluie toate subiectele neobservate la o lectura anterioara. Daca aveti timp incercati intrebarea 13.

Workshop

### 1. Selectati toate informatiile din tabela SALGRADE.

GRADE	LOSAL	HISAL
-----	-----	-----
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

### 2. Selectati toate informatiile din tabela EMP.

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
-----	-----	-----	----	-----	-----	-----	-----
7369	SMITH	CLERK	7902	13-JUN-83	800.00		20
7499	ALLEN	SALESMAN	7698	15-AUG-83	1,600.00	300.00	30
7521	WARD	SALESMAN	7698	26-MAR-84	1,250.00	500.00	30
7566	JONES	MANAGER	7839	31-OCT-83	2,975.00		20
7654	MARTIN	SALESMAN	7698	05-DEC-83	1,250.00	1,400.00	30
7698	BLAKE	MANAGER	7839	11-JUL-84	2,850.00		30
7782	CLARK	MANAGER	7839	14-MAY-84	2,450.00		10
7788	SCOTT	ANALYST	7566	05-MAR-84	3,000.00		20
7839	KING	PRESIDENT		09-JUL-84	5,000.00		10
7844	TURNER	SALESMAN	7698	04-JUN-84	1,500.00	.00	30
7876	ADAMS	CLERK	7788	04-JUN-84	1,100.00		20
7900	JAMES	CLERK	7698	23-JUL-84	950.00		30
7902	FORD	ANALYST	7566	05-DEC-83	3,000.00		20
7934	MILLER	CLERK	7782	21-NOV-83	1,300.00		10

14 inregistrari selectate.

### 3. Listati toti angajatii care au salariul intre 1000 si 2000.

ENAME	DEPTNO	SAL
-----	-----	-----
ALLEN	30	1,600.00

WARD	30	1,250.00
MARTIN	30	1,250.00
TURNER	30	1,500.00
ADAMS	20	1,100.00
MILLER	10	1,300.00

6 inregistrari selectate.

**4. Listati numerele de departament si numele in ordinea numelor departamentelor.**

DEPTNO	DNAME
10	ACCOUNTING
40	OPERATIONS
20	RESEARCH
30	SALES

**5. Afisati toate tipurile diferite de job-uri.**

JOB
ANALYST
CLERK
MANAGER
PRESIDENT
SALESMAN

**6. Listati detaliile angajatilor din departamentele 10 si 20 in ordinea alfabetica a numelui.**

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7876	ADAMS	CLERK	7788	04-JUN-84	1,100.00		20
7782	CLARK	MANAGER	7839	14-MAY-84	2,450.00		10
7902	FORD	ANALYST	7566	05-DEC-83	3,000.00		20
7566	JONES	MANAGER	7839	31-OCT-83	2,975.00		20
7839	KING	PRESIDENT		09-JUL-84	5,000.00		10
7934	MILLER	CLERK	7782	21-NOV-83	1,300.00		10
7788	SCOTT	ANALYST	7566	05-MAR-84	3,000.00		20
7369	SMITH	CLERK	7902	13-JUN-83	800.00		20

8 inregistrari selectate.

**7. Listati numele si ocupatiile tuturor functionarilor in departamentul 20.**

ENAME	JOB
SMITH	CLERK
ADAMS	CLERK

**8. Afisati toti angajatii ai caror nume contine TH sau LL in interior.**

ENAME
SMITH
ALLEN
MILLER

**9. Listati urmatoarele detalii pentru toti angajatii care**

**au un manager.**

ENAME	JOB	SAY
SMITH	CLERK	800.00
ALLEN	SALESMAN	1,600.00
WARD	SALESMAN	1,250.00
JONES	MANAGER	2,975.00
MARTIN	SALESMAN	1,250.00
BLAKE	MANAGER	2,850.00
CLARK	MANAGER	2,450.00
SCOTT	ANALYST	3,000.00
TURNER	SALESMAN	1,500.00
ADAMS	CLERK	1,100.00
JAMES	CLERK	950.00
FORD	ANALYST	3,000.00
MILLER	CLERK	1,300.00

13 inregistrari selectate.

**10. Afiseaza numele si totalul remuneratiei pentru toti angajatii.**

ENAME	REMUNERATION
SMITH	9600
ALLEN	19500
WARD	15500
JONES	35700
MARTIN	16400
BLAKE	34200
CLARK	29400
SCOTT	36000
KING	60000
TURNER	18000
ADAMS	13200
JAMES	11400
FORD	36000
MILLER	15600

14 inregistari selectate.

**11. Afiseaza toti salariatii care au fost angajati in anul 1983.**

ENAME	DEPTNO	HIREDATE
SMITH	20	13-JUN-83
ALLEN	30	15-AUG-83
JONES	20	31-OCT-83
MARTIN	30	05-DEC-83
FORD	20	05-DEC-83
MILLER	10	21-NOV-830

6 inregistrari selectate.

**12. Afisati numele, salariul anual si comisionul pentru toti vanzatorii**

**ai caror salariu lunar este mai mare decat comisionul lor. Iesirea va fi**

ordonata dupa salariu , cele mai mari primele. Daca doi sau mai multi angajati au acelasi salariu trebuie sortati dupa nume in ordinea celor mai mari salarii.

ENAME	ANNUAL_SAL	COMM
-----	-----	-----
ALLEN	19200	300.00
TURNER	18000	.00
WARD	15000	500.00

Incercati-va aptitudinile cu acesta.

### 13. Selectati informatiile dupa cum sunt selectate.

Cine, cand si cum

SMITH	HAS HELD	THE POSITION OF	CLERK	IN DEPT	20	SINCE	13-JUN-83
ALLEN	HAS HELD	THE POSITION OF	SALESMAN	IN DEPT	30	SINCE	15-AUG-83
WARD	HAS HELD	THE POSITION OF	SALESMAN	IN DEPT	30	SINCE	26-MAR-84
JONES	HAS HELD	THE POSITION OF	MANAGER	IN DEPT	20	SINCE	31-OCT-83
MARTIN	HAS HELD	THE POSITION OF	SALESMAN	IN DEPT	30	SINCE	05-DEC-83
BLAKE	HAS HELD	THE POSITION OF	MANAGER	IN DEPT	30	SINCE	11-JUN-84
CLARK	HAS HELD	THE POSITION OF	MANAGER	IN DEPT	10	SINCE	14-MAY-84
SCOTT	HAS HELD	THE POSITION OF	ANALYST	IN DEPT	20	SINCE	05-MAR-84
KING	HAS HELD	THE POSITION OF	PRESIDENT	IN DEPT	10	SINCE	09-JUL-84
TURNER	HAS HELD	THE POSITION OF	SALESMAN	IN DEPT	30	SINCE	04-JUN-84
ADAMS	HAS HELD	THE POSITION OF	CLERK	IN DEPT	20	SINCE	04-JUN-84
JAMES	HAS HELD	THE POSITION OF	CLERK	IN DEPT	30	SINCE	23-JUL-84
FORD	HAS HELD	THE POSITION OF	ANALYST	IN DEPT	20	SINCE	05-DEC-83
MILLER	HAS HELD	THE POSITION OF	CLERK	IN DEPT	10	SINCE	21-NOV-83

14 inregistrari selectate.

## Capitolul 2 - Rezolvari

1. SELECT *  
FROM SALGRADE;
2. SELECT *  
FROM EMP;
3. SELECT ENAME, DEPTNO, SAL  
FROM EMP  
WHERE SAL BETWEEN 1000 AND 2000;
4. SELECT DEPTNO, DNAME  
FROM DEPT  
ORDER BY DNAME;
5. SELECT DISTINCT JOB  
FROM EMP;
6. SELECT *  
FROM EMP

```

WHERE DEPTNO IN (10,20)
ORDER BY ENAME;

7. SELECT ENAME, JOB
 FROM EMP
 WHERE JOB = 'CLERK'
 AND DEPTNO = 20;

8. SELECT ENAME
 FROM EMP
 WHERE ENAME LIKE '%TH%'
 OR ENAME LIKE '%LL%';

9. SELECT ENAME, JOB, SAL
 FROM EMP
 WHERE MGR IS NOT NULL;

10. SELECT ENAME, SAL*12+NVL(COMM,0) REMUNERATION
 FROM EMP;

11. SELECT ENAME, DEPTNO, HIREDATE
 FROM EMP
 WHERE HIREDATE LIKE '%83';

12. SELECT ENAME, SAL*12 ANNUAL SAL, COMM
 FROM EMP
 WHERE SAL > COMM
 AND JOB = 'SALESMAN'
 ORDER BY SAL DESC, ENAME;

13. SELECT ENAME||
 ' HAS HELD THE POSITION OF '||
 JOB||
 ' IN DEPT '||
 DEPTNO||
 ' SINCE '||
 HIREDATE "Who, what and when"
 FROM EMP;

```