2. Limitarea înregistrărilor folosind o selecție:
Să presupunem, conform exemplului de mai sus, că se doreşte afişarea tuturor angajaților din departame ntul 10 (prezintă interes doar setul de linii care au valoarea 10 în coloana DEPTNO). Această metodă de r estricționare reprezintă baza clauzei WHERE în SQL.

2.1 Limitarea liniilor selectate:
Se poate restricționa numărul de linii returnate de o interogare folosind clauze WHERE. O clauză WHERE conține o condiție ce trebuie îndeplinită de fiecare linie din rezultat şi urmează imediat după clauza FRO M.
Forma:
SELECT [DISTINCT] {*, coloana [alias], …}
FROM tabel
[WHERE conditie];
Unde:
WHERE restricționează interogarea la liniile ce îndeplinesc condiția conditie. conditie reprezintă condiția c e trebuie satisfăcută de fiecare înregistrare ce apare în rezultat; este compusă din nume de coloane, expr esii, constante şi operatori de comparație.
Clauza WHERE poate compara valorile din coloane, valori literale, expresii aritmetice sau funcții, fiind co mpusă din trei elemente:
-numele coloanei
-operatorul de comparație
-nume de coloană, constantă sau listă de valori

2.1.1. Folosirea clauzei WHERE:
Exemplu:
SELECT ename, job, deptno
FROM emp
WHERE job='CLERK';
Exemplul anterior afişează toți angajații care au funcția CLERK.

Şiruri de caractere şi date calendaristice:
Şirurile de caractere şi datele calendaristice utilizate în clauza WHERE trebuie incluse între apostrofuri (‘ ‘). Toate căutările la nivel de caracter sunt case-sensitive (i.e. se face distincție între litere mici şi majuscul e). Datele calendaristice sunt memorate de Oracle în formatul secol, an, luna, zi, ore, minute şi secunde. Afişarea implicită a datei este DD-MON-YY.
Notă: valorile numerice nu trebuie incluse între apostrofuri.

Operatori de comparație:
Operatorii de comparație (=, >, >=, <, <=, <>) sunt folosiți în condiții care compară două expresii.
Utilizarea lor în clauza WHERE respectă următorul format:
WHERE expresie operator valoare
Exemple:
WHERE hiredate='01-JAN-95'
… WHERE sal>=1500
… WHERE ename='SMITH'

Alți operatori de comparație ce pot fi utilizați într-o clauză WHERE:
BETWEEN … AND …; Semnificatie : Între două valori (inclusiv)
IN (listă); Semnificatie : Potriveşte orice valoare din listă
LIKE ; Semnificatie : Potriveste un caracter
IS NULL; Semnificatie : Este valoare null

2.1.1.1 Operatorul BETWEEN
Operatorul BETWEEN se utilizează pentru selectarea valorilor dintr-un interval.
Exemplu:

```
SELECT ename, sal
FROM emp
WHERE sal BETWEEN 1000 AND 1500; //1000 este limita inferioara, 1500 este limita superioara
```

2.1.1.2 Operatorul IN
Operatorul IN este utilizat pentru căutare într-o listă de valori. El poate fi utilizat cu orice tip de dată.
Exemplu:
```
SELECT empno, ename, sal, mgr
FROM emp
WHERE mgr IN (7902, 7566, 7788);
```

Următorul exemplu returnează câte o linie din tabelul emp pentru fiecare angajat al cărui nume este inclus în lista de nume din clauza WHERE.
```
SELECT empno, ename, mgr, deptno
FROM emp
WHERE ename IN ('FORD', 'ALLEN');
```
Notă: dacă în listă sunt folosite caractere sau date calendaristice, acestea trebuie incluse între apostrofuri (' ').

2.1.1.3 Operatorul LIKE
Exemplu:
```
SELECT ename
FROM emp
WHERE ename LIKE 'S%';
```

Nu întotdeauna se cunoaşte valoarea exactă pe baza căreia se va efectua căutarea. Instrucțiunea SELE CT permite selectarea liniilor care corespund unui tipar de caractere cu ajutorul operatorului LIKE. Operați a de potrivire după un tipar de caractere este referită drept căutare cu caractere wildcard. Pentru construir ea şirurilor de căutare pot fi utilizate două simboluri:
- % Reprezintă orice secvență de caractere;
- _ (underscore) Reprezintă un singur caracter;

Instrucțiunea SELECT de mai sus returnează numele angajaților din tabelul emp al căror nume începe cu “S”. Numele care încep cu “s” nu vor fi returnate.

În anumite cazuri, operatorul LIKE poate fi utilizat în locul operatorului BETWEEN. Următorul exemplu afiş easză numele şi data angajării tuturor angajaților a căror angajare s-a făcut între ianuarie 1981 şi decembri e 1981.
Exemplu:
```
SELECT ename, hiredate
FROM emp
WHERE hiredate LIKE '%81';
```

Pot fi combinate diferite tipuri de potriviri pe caracter:
```
SELECT ename
FROM emp
WHERE ename LIKE '_A%';
```

Dacă se doreşte căutarea caracterelor ‘%’ sau ‘_’ se va folosi opțiunea ESCAPE, care precizează de fapt caracterul care va fi utilizat drept Escape.
Următoarea instrucțiune SELECT afişează numele tuturor angajaților al căror nume conține secvența de c aractere “A_S”.
Exemplu:
```
SELECT ename
FROM emp
```

WHERE ename LIKE '%A_S' ESCAPE '\';

2.1.1.4 Operatorul IS NULL
Operatorul IS NULL este utilizat pentru căutarea valorilor null. Deoarece valoarea null are semnificația un ei valori indisponibile, neatribuite, necunoscute sau neaplicabile ea nu poate apare în cadrul unei operații de comparație, deoarece ar conduce la un rezultat null. De exemplu, pentru a afişa numele, funcția şi com isionul tuturor angajaților care nu au dreptul la comision se va folosi
următoarea instrucțiune SELECT:
Exemplu:
SELECT ename, job, comm
FROM emp
WHERE comm IS NULL;

2.1.1.5 Operatori logici (AND, OR, NOT)
Un operator logic combină două componente de tip condiție pentru a produce un singur rezultat bazat pe acestea sau inversează rezultatul unei singure condiții. În SQL sunt disponibili trei operatori logici: AND, OR şi NOT.
-AND Returnează TRUE dacă ambele componente ale condiției sunt adevărate;
-OR Returnează TRUE dacă una din componentele condiției este adevărată;
-NOT Returnează TRUE dacă respectiva condiție este falsă;

Folosirea operatorului AND:
SELECT empno, ename, job, sal
FROM emp
WHERE sal>=1100 AND job='CLERK';
În exemplul de mai sus ambele condiții trebuie să fie adevărate pentru a fi selectată o înregistrare.
De aceea, un angajat care are funcția CLERK şi câştigă mai mult de $1100 va fi selectat.
Notă: -toate căutarile de tip caracter sunt case-sensitive.
- şirurile de caracter trebuie incluse între apostrofuri (' ').

Folosirea operatorului OR:
SELECT empno, ename, job, sal
FROM emp
WHERE sal>=1100 OR job='CLERK';
În exemplul de mai sus, vor fi selectate înregistrările care îndeplinesc cel puțin o condiție: fiesal>=1100, fi e job=’CLERK’.

Folosirea operatorului NOT:
SELECT ename, job
FROM emp
WHERE job NOT IN ('CLERK', 'MANAGER','ANALYST');
În exemplul de mai sus este afişat numele şi funcția tuturor angajaților a căror funcție nu este CLERK, MA NAGER sau ANALYST.

Notă: operatorul NOT poate fi combinat şi cu alți operatori SQL, cum ar fi BETWEEN, LIKE şi IS NULL.
… WHERE job NOT IN (‘CLERK’, ‘ANALYST’)
… WHERE sal NOT BETWEEN 1000 AND 1500
… WHERE comm IS NOT NULL

2.1.2 Clauza ORDER BY
Liniile returnate de o interogare sunt afişate într-o ordine oarecare. Pentru sortarea liniilor se utilizează cla uza ORDER BY, care, dacă este folosită, trebuie să apară ultima în instrucțiunea SELECT. Sortarea se p oate face dupa o coloană, o expresie sau după un alias de coloană.

Sintaxă:

SELECT expresie
FROM tabel
[WHERE conditie]
[ORDER BY {coloana, expresie} [ASC|DESC];
unde:
ORDER BY specifică ordinea în care sunt afişate liniile.
ASC ordonează liniile ascendent – implicit.
DESC ordonează liniile descendent.

Dacă nu este folosită clauza ORDER BY ordinea sortării este nedefinită şi Serverul Oracle poate afişa linii le în ordine diferită pentru două interogări identice.

SELECT-ul următor afişează rezultatele interogării ordonate descrescător după coloana hiredate.
SELECT ename, job, deptno, hiredate
FROM emp
ORDER BY hiredate DESC;

Următoarea instrucțiune SELECT ordonează liniile după aliasul coloanei sal*12 (de tip expresie aritmetică).
SELECT empno, ename, sal*12 annsal
FROM emp
ORDER BY annsal;

În următorul exemplu liniile sunt sortate după coloanele deptno şi sal, iar liniile având aceeaşi valoare pentru deptno fiind sortate descendent după sal.
SELECT ename, deptno, sal
FROM emp
ORDER BY deptno, sal DESC;

Notă: se poate face sortare şi după o coloană care nu este în lista clauzei SELECT.