

LABORATOR 1:

INTRODUCERE ÎN LIMBAJUL SQL, CERERI MONOTABEL

Conținut laboratoare:

Baze de date (baze de date relaționale)

O bază de date este o colecție organizată de informații, stocată într-un ansamblu de fișiere.

SGBD Oracle

Un sistem de gestiune a bazei de date (SGBD) este un produs software care asigură interacțiunea cu o bază de date, permițând definirea, consultarea și actualizarea datelor din baza de date.

SQL (Structured Query Language) este un limbaj neprocedural pentru interogarea și prelucrarea informațiilor din baza de date. Compilatorul limbajului SQL generează automat o procedură care accesează baza de date și execută comanda dorită. Limbajul SQL permite atât definirea, prelucrarea și interogarea datelor, cât și controlul accesului la acestea. Comenzile SQL pot fi integrate în programe scrise în alte limbaje, de exemplu C/C++, Java etc.

În funcție de tipul acțiunii pe care o realizează, instrucțiunile SQL se împart în mai multe categorii. Datorită importanței pe care o au comenzile componente, unele dintre aceste categorii sunt evidențiate ca limbaje în cadrul SQL, și anume:

- limbajul de definire a datelor (**LDD**) comenzile CREATE, ALTER, DROP;
- limbajul de prelucrare a datelor (**LMD**) comenzile INSERT, UPDATE, DELETE, SELECT;
- limbajul de control al datelor (**LCD**) comenzile COMMIT, ROLLBACK.

Instalare software:

- **Oracle Database 11g Release 2:**

<http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html?ssSourceSiteId=ocomfi>

- **SqlDeveloper:**

<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>

Arhitectura bazelor de date Oracle

http://docs.oracle.com/cd/E15586_01/server.1111/e25789/intro.htm#autoId15

instanțe, SGA, PGA;

structura fizică, structura logică etc.

Exercițiul 1: Consultați diagrama exemplu HR (Human Resources) pentru lucrul în cadrul laboratoarelor SQL. Identificați cheile primare și cele externe ale tabelelor existente în schemă, precum și tipul relațiilor dintre aceste tabele.

LIMBAJUL LMD, COMANDA SELECT

Clauza SELECT permite precizarea coloanelor care trebuie regăsite în baza de date.

**SELECT lista_expresii
FROM lista_tabele**

În lista_expresii se enumeră coloanele și expresiile ale căror valori trebuie obținute din baza de date. Pentru fiecare element al listei poate fi specificat câte un nume (alias).

Exemplul 2: Să se obțină numele, salariul și codul jobului angajaților

```
SELECT last_name as nume, salary as "Salariu", job_id "titlu job"  
FROM employees;
```

Se observă că alias-ul apare în rezultat, ca antet al coloanei corespunzătoare expresiei respective. Dacă un alias conține blank-uri, el va fi scris obligatoriu între ghilimele. Altfel, ghilimelele pot fi omise. Doar alias-urile specificate între ghilimele sunt case-sensitive, celelalte fiind scrise implicit cu majuscule. Cuvântul cheie "as" este opțional.

Exemplul 3: Să se obțină numele și prenumele angajaților

```
SELECT rownum nr_crt, first_name, last_name  
FROM employees;
```

O pseudocoloană se comportă ca o coloană a unui tabel, dar nu este stocată efectiv într-un tabel. Se pot face interogări asupra pseudocoloanelor, dar nu se pot insera, actualiza sau șterge valorile acestora. În exemplul anterior, rownum este o pseudocoloană căreia i s-a atribuit aliasul nr_crt. Prin intermediu acestei expresii fiecărei linii din tabelul employees i s-a atribuit un număr de ordine.

Pseudocoloana sysdate returnează data curentă. Tabelul dual este utilizat pentru a se completa sintaxa comenzii SELECT.

Exercițiul 4: Să se modifice exemplul 2 astfel încât denumirile coloanelor să fie Nr. crt, Numele angajatului, Prenumele angajatului.

Exemplul 5: Să se afișeze data curentă.

```
SELECT sysdate  
FROM dual;
```

Exercițiul 6: Să se afișeze pentru fiecare job: codul, titlul și diferența dintre salariul maxim și salariul minim.

Exercițiul 7: Să se afișeze pentru fiecare angajat codul, numele, codul departamentului în care lucrează fiecare angajat și numărul de zile care au trecut de la data angajării.

Exercițiul 8: Să se afișeze numele angajaților, codul_angajaților (employee_id) și salariul anual al acestora (salary * 12).

Exercițiul 9: Să se afișeze numele angajaților și valoarea comisionului raportată la salariu.

Cuvintele cheie DISTINCT și UNIQUE sunt sinonime și utilizarea lor determină sistemul Oracle să returneze o singură copie din fiecare mulțime de linii duplicate selectate. Opțiunea ALL permite ca sistemul să returneze toate liniile selectate, inclusiv toate duplicatele. Această opțiune este implicită.

Exemplul 10: Să se listeze, cu și fără duplicate, codurile job-urilor din tabelul employees.

```
SELECT DISTINCT job_id  
FROM employees;
```

```
SELECT UNIQUE job_id  
FROM employees;
```

```
SELECT ALL job_id  
FROM employees;
```

```
SELECT job_id  
FROM employees;
```

Exercițiul 11: Să se listeze, ce joburi au salariații și în ce departamente lucrează aceștia.

Operatorul de concatenare a șirurilor de caractere este ||. Șirurile de caractere se includ între apostrofuri. Ghilimelele delimitează alias-uri.

Exemplul 12: Să se obțină numele și prenumele angajaților precum și codul job-urilor acestora. O singură coloană intitulată "nume și funcție" va conține rezultatul cererii.

```
SELECT last_name || ' ' || first_name || ' ' || job_id "nume și funcție"  
FROM employees;
```

Exercițiul 13: Să se obțină codul angajaților, concatenat cu numele și cu salariul.

SELECT lista_expresii
FROM lista_tabele
WHERE conditie

Selecția din algebra relațională este realizată prin clauza WHERE. Această clauză este mai complexă decât operația de selecție din algebra relațională, deoarece expresia care o urmează poate conține atât comparații de attribute și/sau expresii aritmetice, cât și operatori logici (AND, OR, NOT), operatori pe mulțimi (UNION, INTERSECT, MINUS), operatori de apartenență la mulțimi cu negările acestora (IN, NOT IN) și operatorul de existență (EXISTS).

Operatorii prezentați în ordinea descrescătoare a priorității sunt:

- Operatori aritmetici
unari +, -
binari +, -, *, /
- concatenarea stringurilor ||
- operatori pentru comparații =, >, >=, <, <=, <>
- [NOT] BETWEEN AND, [NOT] IN (), LIKE, IS [NOT] NULL, EXISTS
- operatori logici NOT, AND, OR

Se pot folosi paranteze pentru a se forța o anumită ordine de evaluare.

Exemplul 14: Să se obțină numele și funcția angajaților care au un salariu mai mare decât 8000.

```
SELECT last_name, job_id, salary
FROM employees
WHERE salary > 8000;
```

Exercițiul 15: Să se obțină numele și venitul angajatului cu codul 170.

Exercițiul 16: Să se obțină informații despre angajații care au o vechime, exprimată în număr de zile mai mare decât 1000.

Apartenența la o mulțime finită de valori se poate testa prin intermediul operatorului IN, urmat de lista valorilor între paranteze și separate prin virgule:
expresie IN (valoare_1, valoare_2, ... , valoare_n)

Exemplul 17: Să se afișeze angajații care lucrează în departamentele 10 și 20.
Să se afișeze angajații care nu lucrează în departamentele 60, 40.

```
SELECT employee_id, last_name, first_name, department_id
FROM employees
WHERE department_id IN (10, 20)
```

```
SELECT employee_id, last_name, first_name, department_id
FROM employees
WHERE department_id NOT IN (60, 40)
```

Exercițiul 18: Să se afișeze numele și codul departamentului pentru angajații care au codul jobului IT_PROG sau HR_REP.

Exercițiul 19: Să se listeze numele și salariile angajaților care câștigă mai mult de 1500 \$ și lucrează în departamentul 10 sau 30. Se vor eticheta coloanele Angajat respectiv Salariu lunar.

Exemplul 20: Să se afișeze informații despre directorul general (se presupune că acesta este angajatul care nu are manager).

```
SELECT last_name, first_name, job_id
FROM employees
WHERE manager_id IS NULL;
```

Exercițiul 21: Să se afișeze venitul (salariu + comision) angajaților care câștigă comision?

Exercițiul 22: Care este codul jobului angajatului pentru care nu se cunoaște departamentul în care lucrează?

Pentru testarea apartenenței la un domeniu de valori se utilizează operatorul expresie [NOT] BETWEEN valoare_1 AND valoare_2

Exemplul 23: Să se listeze numele și salariul pentru toți angajații al căror salariu nu se află în intervalul 7000-17000\$.

```
SELECT last_name, salary
FROM employees
WHERE salary NOT BETWEEN 7000 AND 17000;
```

Pentru compararea șirurilor de caractere, împreună cu operatorul LIKE se utilizează caracterele wildcard:

- % - reprezentând orice șir de caractere, inclusiv șirul vid;
- _ (underscore) -reprezentând un singur caracter și numai unul.

Funcțiile UPPER(sir), LOWER(sir) convertesc șirul astfel încât acesta să fie scris cu majuscule respectiv minuscule.

Exemplul 24: Expresia last_name LIKE 'A%' este adevărată pentru numele care încep cu litera A; expresia UPPER(last_name) LIKE '_B%' este adevărată pentru numele care au a doua literă b sau B.

Exercițiul 25: Să se afișeze numele, job-ul și salariul pentru toți salariații al căror job conține șirul CLERK sau REP și salariul nu este egal cu 1000, 2000 sau 3000 \$.

Exercițiul 26: Să se listeze numele tuturor angajaților care au 2 litere 'l' în nume și lucrează în departamentul 30 sau managerul lor este 102.

Recapitulare:

SINTAXĂ COMANDĂ SELECT

SELECT {{{DISTINCT|UNIQUE}|ALL}lista_campuri[*]}

FROM [nume_schema.]nume_obiect]

[,[nume_schema.]nume_obiect ...]

[WHERE conditie_clauza_where]

[START WITH conditie_clauza_start_with

CONNECT BY conditie_clauza_connect_by]

[GROUP BY expresie[, expresie ...]

[HAVING conditie_clauza_having]]

[ORDER BY {expresie|pozitie}{[, {expresie|pozitie} ...}]

[FOR UPDATE

[OF [[nume_schema.]nume_obiect.]nume_coloana

[, [[nume_schema.]nume_obiect.]nume_coloana] ...]

[NOWAIT | WAIT numar_intreg]];

Care sunt clauzele obligatorii ale comenzii SELECT?

Opțional

STRUCTURA BAZELOR DE DATE ORACLE

La nivel **fizic** o bază de date Oracle presupune existența următoarelor tipuri de fișiere:

1. Datafiles: fișiere de date
2. Redo Log Files: fișiere de reluare
3. Control Files: fișiere de control

La nivel **logic** date sunt grupate în

1. blocuri

Din punct de vedere structural un bloc este compus din: antet, spațiu liber, spațiu pentru date. Unui bloc îi corespunde un număr de bytes. (DB_BLOCK_SIZE – multiplu al dimensiunii blocului sistemului de operare)

Pentru alocare manuală a spațiului se folosesc parametrii: PCTFREE și PCTUSED (procent din spațiul total).

2. Extensii

O extensie este compusă dintr-o succesiune continuă de blocuri alocate unui tip specific de informație. O extensie conține date dintr-un singur fișier.

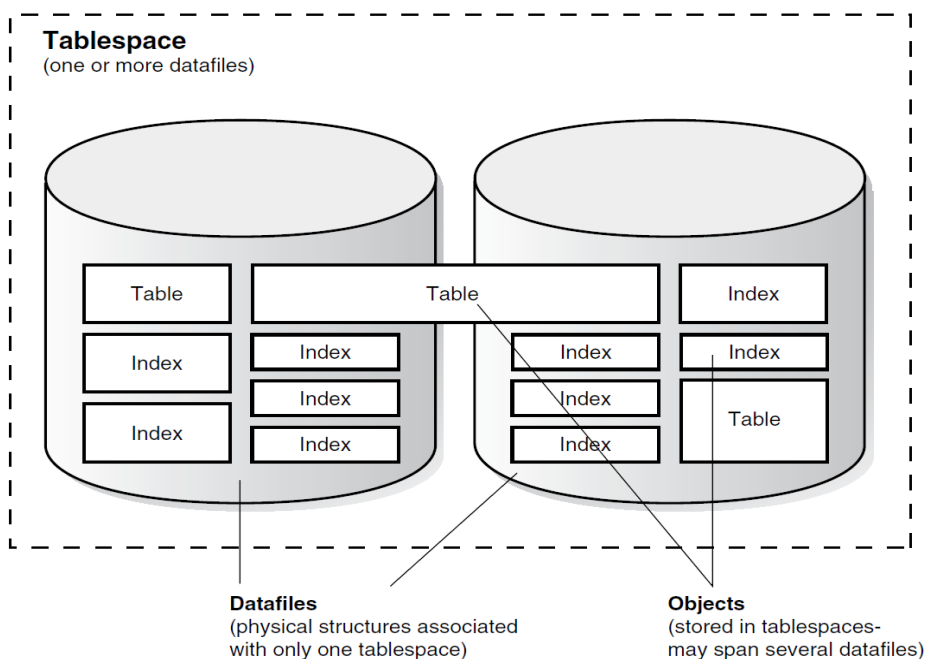
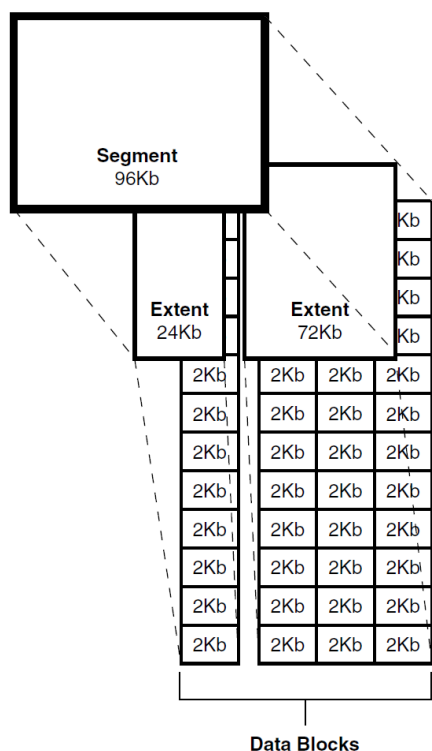
3. Segmente

Un segment grupează mai multe extensii stocate în același table-space și păstrează datele dintr-un tabel, partiția unui tabel sau cluster.

Un segment poate conține date din mai multe fișiere. Când o extensie este completată, Oracle alocă spațiu pentru o nouă extensie.

4. Tablespace-uri

Un tablespace este format dintr-unul sau mai multe segmente și corespunde mai multor fișiere (datafiles – structurile fizice specifice sistemului de operare).



UTILITARUL SQL*PLUS

SQL*Plus

SQL*Plus este un utilitar Oracle, având comenzi proprii specifice, care recunoaște instrucțiunile SQL și le trimite server-ului Oracle pentru execuție.

Comenzile Sql*Plus nu permit manipularea datelor din baza de date. Dintre funcționalitățile mediului SQL*Plus, se pot enumera:

- editarea, executarea, salvarea și regăsirea instrucțiunilor SQL și a blocurilor PL/SQL;
- calculul, stocarea și afișarea rezultatelor furnizate de cereri;
- listarea structurii tabelelor;
- accesarea și copierea de informații dintr-o bază de date în alta;
- administrarea bazei de date.

În tabelul următor sunt evidențiate diferențele dintre comenzile SQL și comenzile SQL*Plus.

SQL	SQL*Plus
Este un limbaj de comunicare cu server-ul Oracle pentru accesarea datelor.	Recunoaște instrucțiunile SQL și le transferă server-ului Oracle.
Prelucrează date și definește obiecte din baza de date.	Nu permite prelucrarea informațiilor din baza de date.
Nu are un caracter de continuare.	Acceptă "-" drept caracter de continuare pentru comenzile scrise pe mai multe linii.
Instrucțiunile nu pot fi abreviate.	Comenzile pot fi abreviate.
Utilizează funcții pentru a efectua formatare.	Utilizează comenzi pentru formatarea datelor.
Caracterul de terminare a unei comenzi este ";"	Nu necesită caracter de terminare a unei comenzi.
Se bazează pe standardul ANSI pentru SQL.	Este o interfață specifică sistemului Oracle pentru execuția instrucțiunilor SQL.

Exemple de comenzi SQL*Plus:

RUN abreviere /

Comenzile Sql*Plus nu sunt depuse în buffer-ul Sql. Ultima cerere SQL lansată de către client este păstrată în buffer-ul SQL. Pentru rularea conținutului buffer-ului, se utilizează comanda RUN

DESC[RIBE] nume_obiect

Comanda DESCRIBE listează atributele unui tabel sau ale unui alt obiect.

SET LINE n

Comanda SET LINE stabilește dimensiunea paginii (numărul de caractere).

SET PAGES[IZE] n

Se stabilește numărul de linii afișat pe o pagină.

SET SPACE n

Se stabilește numărul de spații dintre coloane.

SET COLSEP 'x'

Separatorul dintre coloane va fi x.

SAVE path\nume.sql

Comanda SAVE salvează conținutul buffer-ului (ultima comanda SQL) în fișierul specificat. Nu este obligatorie precizarea extensiei sql.

START path\nume.sql (sau @path\nume.sql)

Comanda START execută comenzile din fișierul specificat.

EDIT path\nume.sql

Comanda EDIT deschide un editor de texte pentru modificarea scriptului specificat. Dacă lipsește numele fișierului, se editează ultima comanda din buffer.

Exemplul 27: Să se listeze conținutul tabelului DEPARTMENTS.

Să se listeze structura tabelului DEPARTMENTS, observând tipurile de date ale coloanelor.

Fără a se rescrie prima comandă tastată, să se listeze încă o dată conținutul tabelului DEPARTMENTS.

SELECT * FROM DEPARTMENTS;

DESCRIBE DEPARTMENTS

RUN