
L3. SQL – Proiecție. Redenumire coloane. Selecție. Ordonare rezultat.

Funcții de bibliotecă Oracle SQL

Operatorul de proiecție (π) din algebra relațională extrage o selecție de coloane dintr-o relație R și elimină duplicatele. Operatorul de selecție (σ) selectează un subset de rânduri dintr-o relație R pe baza unei condiții de selecție. Ambii operatori sunt implementați în SQL folosind instrucțiunea SELECT.

3.1. *Proiectarea coloanelor și redenumirea câmpurilor rezultatului*

Sintaxa generală a operatorului de proiecție este:

SELECT [**ALL** | **DISTINCT**] listă proiecție
FROM tablă;

Lista de proiecție este o listă de coloane din tabelă despărțite prin virgulă. Ca și alternativă poate fi înlocuită cu semnul '*' pentru a se obține proiecție 1:1. În acest caz numele și ordinea coloanelor în rezultat vor fi cele din tabela originală.

Dacă se dorește redenumirea câmpurilor se va folosi opțiunea **AS** urmată de noul nume pentru respectivul câmp.

Opțiunea **DISTINCT** forțează eliminarea duplicatelor. În mod implicit ele nu se elimină (implicit este **ALL**).



ACTIVITATE 1: Utilizând **SQL Workshop** -> **SQL Commands** rulați următoarele interogări SQL:

SELECT sid **AS** ID_student, nume, an **FROM** Student;

SELECT an **FROM** Student;

SELECT DISTINCT an **FROM** Student;

SELECT * FROM Student;

3.2. *Selecția rândurilor în SQL*

Sintaxa generală pentru implementare selecție este:

SELECT *
FROM tablă
WHERE condiție_selecție;

Deoarece lista de proiecție nu poate lipsi din sintaxa **SELECT**, pentru selecție pură se va folosi proiecția 1:1.

Condiția de selecție poate utiliza operatori de comparație (=, <, >, <=, >=, <>) și operatori logici (**AND**, **OR**, **NOT**). De asemenea se poate utiliza și operatorul de verificare tipar într-un șir de caractere **LIKE** în care caracterul '_' va înlocui un singur caracter în tipar, iar '%' poate înlocui o secvență de 0..n caractere.



ACTIVITATE 2: Utilizând **SQL Workshop** -> **SQL Commands** rulați următoarele interogări SQL:

SELECT * FROM Student **WHERE** sid=' SET001';

SELECT * FROM Student **WHERE** an=1;

3.3. *Ordonarea rezultatului în SQL*

La extragerea datelor din tabele sistemul de gestiune nu garantează o anumită ordine a înregistrărilor în rezultat. Sintaxa generală pentru forțarea ordonării la selecție este:

```
SELECT *  
  FROM tablă  
  [WHERE condiție_selecție]  
  [ORDER BY listă_ordonare]
```

În cazul în care lista de ordonare conține mai mult de o expresie/coloană, ordonarea se va face ierarhic. Se va lua în considerare prima expresie și doar la egalitate se va considera cea de a doua și așa mai departe.



ACTIVITATE 3: Utilizând **SQL Workshop** -> **SQL Commands** rulați următoarele interogări SQL:

```
SELECT * FROM Student ORDER BY an;  
  
SELECT sid, nume, an, media FROM Student  
  ORDER BY an, media;  
  
SELECT * FROM Student  
  WHERE bursa>0  
  ORDER BY an, media;
```

3.4. *Funcții de bibliotecă Oracle SQL*

Funcțiile de bibliotecă din Oracle SQL permit prelucrarea diverselor tipuri de date direct în cadrul unei interogări SQL.

A. Funcții pentru Șiruri de caractere

- Operatorul **like** (ex. sir **like** 'tipar' plus '%' și '_')
- Operatorul de concatenare '||' (ex. 'Ana' || ' are ' || 'mere')
- **lower**(sir) – transformă șirul cu litere mici

-
- **upper(sir)** – transformă șirul cu litere mari
 - **substr(sir, poz [, lng])** – returnează subșirul începând cu poziția *poz* de lungime *lng* (sau până la sfârșit dacă *lng* lipsește)
 - **instr(sir, subsir [, dela, aparitie_n])** – returnează poziția unde apare *subsir* în *sir*, căutarea poate eventual începe de la poziția *dela* sau poate căuta apariția *n* a subșirului în șirul dat
 - **trim(sir)** – șterge spațiile de început și sfârșit
 - **lpad(sir, lng, car)** - returnează *sir* completat în stânga cu caracterul *car* până la lungimea totală *lng*.
 - **length(sir)** – returnează lungimea șirului

B. Funcții Numerice

- Operatorul **between** – testează apartenența la un interval. Ex. *n1* between 1 and 10
- **abs(nr)** – returnează *nr* în valoare absolută fără semn
- **trunc(nr, zec)** - trunchiază un *nr* la un număr specificat de *zec* zecimale
- **round(nr, [zec]), ceil(nr), floor(nr)** – rotunjește *nr* la *zec* zecimale, respectiv în sus (ceil) sau în jos (floor)
- **remainder(n1, n2)** – restul împărțirii lui *n1* la *n2*

C. Funcții tip Dată calendaristică

- **CURRENT_DATE** sau **SYSDATE** – returnează data și ora curentă de pe server
- **extract({ year | month | day | hour | minute | second } from expd)** – extrage anul, luna, ziua, ora, minutul, secunda din data *expd* (dată sau timestamp)
- **months_between(d1, d2)** – calculează numărul de luni între cele două date

D. Funcții generice

- Operatorii convertește *nr* la șir de caractere, eventual cu formatul *fmt* (poate include L pentru moneda locală, 9 pentru cifre și G și D pentru virgule la mii sau la zecimale). Ex. `to_char(16356, 'L99G999D99')`
- **IS NULL / IS NOT NULL** – returnează TRUE / FALSE dacă valoarea este NULL și invers în caz contrar
- **NVL(exp1, exp2)** – returnează exp1 dacă nu este NULL și exp2 în caz contrar. Ex. `NVL(ume, 'Necunoscut')`
- **NVL2(exp1, exp2, exp3)** – testează exp1 și dacă nu este NULL returnează exp2 iar în caz contrar exp3. Ex. `NVL(varsta, 'Cunoscuta', 'Nu exista informatia')`
- **to_char(d)** – convertește *d* la șir de caractere, eventual cu formatul *fmt* (poate include Y, M, D, HH24, MI, SS pentru an, luna, zi, ora minut secundă). Ex. `to_char(d1, 'DD-MM-YYYY HH24:MI:SS')`
- **to_date(sir, fmt)** – convertește șirul la o dată calendaristică considerând formatul *fmt*. Ex. `to_date('23/04/2022', 'DD/MM/YYYY')`
- **to_timestamp(sir, fmt)** – la fel ca și `to_date` dar ia în considerare și timpul pe lângă dată
- **to_char(nr, fmt)** - convertește *nr* la șir de caractere, eventual cu formatul *fmt* (poate include L pentru moneda locală, 9 pentru cifre și G și D pentru virgule la mii sau la zecimale). Ex. `to_char(16356, 'L99G999D99')`
- **to_number(sir, fmt)** - convertește *șir* în număr, eventual cu formatul *fmt* (poate include L pentru moneda locală, 9 pentru cifre și G și D pentru virgule

la grupe mii sau la zecimale). Ex. `to_char(16356, 'L99G999D99')`

- **CASE WHEN** exp1 **THEN** val1
 WHEN exp2 **THEN** val2
 ...
 ELSE valN
END - testează expresiile logice exp1 .. expN-1 și returnează valoarea asociată primei expresii adevărate. Dacă nici una nu este adevărată returnează cea corespunzătoare valorii **ELSE**

Obs: pentru a testa funcțiile de bibliotecă acestea pot fi incluse într-un select din tabela Dual (o tablă sistem falsă cu un singur rând și o singură coloană).

Ex. **SELECT** upper('aBsscdC') **FROM** Dual;



ACTIVITATE 4: Utilizând **SQL Workshop** -> **SQL Commands** rulați următoarele interogări SQL:

```
SELECT nume, upper(nume) AS numS,  
          length(nume) AS lngName  
FROM Student;
```

```
SELECT sid, nume, an FROM Student  
      WHERE nume LIKE 'N%';
```

```
SELECT extract(MONTH from SYSDATE) FROM  
Dual;
```

```
SELECT nume, media,  
      CASE WHEN media<5 THEN 'NS'  
          WHEN media<7 THEN 'S'  
          WHEN media<10 THEN 'B'  
          ELSE 'FB'  
      END AS calificativ  
FROM Student;
```

3.5. *Exerciții:*

L3.Ex1. Să se listeze toți studenții a căror nume de familie se termină cu litera ‘u’.

L3.Ex2. Considerând că toți studenții au un singur prenume, să se afișeze o lista cu numele de familie, prenumele și anul de studiu a tuturor studenților.

L3.Ex3. Să se listeze toți studenții a căror prenume este mai lung de 7 caractere inclusiv. Se va afișa numele complet, prenumele și lungimea prenumelui pentru verificare.

L3.Ex4. Să se afișeze toți studenții care au împlinit deja sau vor împlini anul acesta 20 de ani. Se va afișa numele, anul si data nașterii. Anul curent se consideră cel obținut cu SYSDATE.

L3.Ex5. Să se afișeze toți studenții care își serbează ziua de naștere în luna curentă. Obs: luna curentă se va obține din SYSDATE.

L3.Ex6. Să se afișeze toți studenții care nu au completat câmpul bursa (a căror bursă nu se cunoaște). Cu o nouă interogare afișați apoi studenții care nu au bursă.

L3.Ex7. Să se afișeze toți studenții din anul 2. Se va afișa numele, anul si CNP-ul. Pentru studenții care nu au completat CNP-ul se va afișa ‘CNP necompletat’.

L3.Ex8. Să se scrie două interogări SQL, una care listează toți studenții de sex feminin și cealaltă pe cei de sex masculin. Se va afișa numele, cnp-ul și anul de studiu.

L3.Ex9. Se consideră studenții împărțiți în două grupuri – fete și respectiv băieți. Să se afișeze toți studenți afișând numele, anul de studiu și grupul (fete sau băieți) din care fac parte.