

Baze de date - Anul 1

Tutoriatul 3: Joinuri SQL

1 Joinuri SQL

Joinurile în SQL sunt operații care permit combinarea rândurilor din două sau mai multe tabele pe baza unei condiții de legătură. Acestea sunt esențiale pentru lucrul cu baze de date relaționale. Principalele tipuri de joinuri în SQL sunt:

- INNER JOIN
- LEFT (OUTER) JOIN
- RIGHT (OUTER) JOIN
- FULL (OUTER) JOIN
- CROSS JOIN
- SELF JOIN

1.1 INNER JOIN

Acest join returnează rândurile când există cel puțin o potrivire în ambele tabele. Este cel mai utilizat tip de join.

1.1.1 Sintaxă

```
SELECT coloane  
FROM tabela1  
INNER JOIN tabela2  
ON tabela1.coloana = tabela2.coloana;
```

1.1.2 Exemplu

Pentru exemplificare, să presupunem că avem două tabele: ANGAJATI și DEPARTAMENTE.

```
SELECT e.first_name, e.last_name, d.department_name  
FROM employees e  
INNER JOIN departments d  
ON e.department_id = d.department_id;
```

Această interogare returnează numele și prenumele angajaților împreună cu numele departamentului în care lucrează, dar doar pentru angajații care au un departament asociat.

1.2 LEFT (OUTER) JOIN

Acest join returnează toate rândurile din tabela din stânga și rândurile corespunzătoare din tabela din dreapta. Dacă nu există potriviri, rezultatul va conține valori NULL pentru coloanele tabelului din dreapta.

1.2.1 Sintaxă

```
SELECT coloane
FROM tabela1
LEFT JOIN tabela2
ON tabela1.coloana = tabela2.coloana;
```

1.2.2 Exemplu

```
SELECT e.first_name, e.last_name, d.department_name
FROM employees e
LEFT JOIN departments d
ON e.department_id = d.department_id;
```

Această interogare returnează toți angajații, chiar și cei care nu au un departament asociat. Pentru angajații fără departament, valoarea `department_name` va fi NULL.

1.3 RIGHT (OUTER) JOIN

Acest join returnează toate rândurile din tabela din dreapta și rândurile corespunzătoare din tabela din stânga. Dacă nu există potriviri, rezultatul va conține valori NULL pentru coloanele tabelului din stânga.

1.3.1 Sintaxă

```
SELECT coloane
FROM tabela1
RIGHT JOIN tabela2
ON tabela1.coloana = tabela2.coloana;
```

1.3.2 Exemplu

```
SELECT e.first_name, e.last_name, d.department_name
FROM employees e
RIGHT JOIN departments d
ON e.department_id = d.department_id;
```

Această interogare returnează toate departamentele, chiar și cele fără angajați. Pentru departamentele fără angajați, valorile pentru `first_name` și `last_name` vor fi NULL.

1.4 FULL (OUTER) JOIN

Acest join returnează toate rândurile când există o potrivire în oricare dintre tabele. Dacă nu există potriviri, rezultatul va conține valori NULL pentru coloanele tabelului care nu are potriviri.

1.4.1 Sintaxă

```
SELECT coloane
FROM tabela1
FULL JOIN tabela2
ON tabela1.coloana = tabela2.coloana;
```

1.4.2 Exemplu

```
SELECT e.first_name, e.last_name, d.department_name
FROM employees e
FULL JOIN departments d
ON e.department_id = d.department_id;
```

Această interogare returnează toți angajații și toate departamentele, inclusiv angajații fără departamente și departamentele fără angajați.

1.5 CROSS JOIN

Acest join returnează produsul cartezian al celor două tabele, adică fiecare rând din prima tabelă este combinat cu fiecare rând din a doua tabelă. Nu necesită o condiție de join.

1.5.1 Sintaxă

```
SELECT coloane
FROM tabela1
CROSS JOIN tabela2;
```

1.5.2 Exemplu

```
SELECT e.first_name, e.last_name, d.department_name
FROM employees e
CROSS JOIN departments d;
```

Această interogare returnează toate combinațiile posibile între angajați și departamente, indiferent dacă există sau nu o relație între ele. Numărul de rânduri rezultate va fi produsul dintre numărul de angajați și numărul de departamente.

1.6 SELF JOIN

Un self join este un join în care o tabelă este unită cu ea însăși. Este util pentru a găsi relații între rândurile din aceeași tabelă.

1.6.1 Sintaxă

```
SELECT coloane
FROM tabela t1
JOIN tabela t2
ON t1.coloana = t2.coloana;
```

1.6.2 Exemplu

```
SELECT e1.first_name || ' ' || e1.last_name AS "Angajat",
       e2.first_name || ' ' || e2.last_name AS "Manager"
FROM employees e1
JOIN employees e2
ON e1.manager_id = e2.employee_id;
```

Această interogare returnează numele angajaților împreună cu numele managerilor lor, folosind tabela EMPLOYEES unită cu ea însăși.

2 Exerciții

2.1 Problema 1 (Ușoară)

Afișați numele angajaților, jobul și numele departamentului în care lucrează.

```
SELECT e.last_name, e.job_id, d.department_name
FROM employees e
INNER JOIN departments d
ON e.department_id = d.department_id;
```

2.2 Problema 2 (Difilă)

Afișați numele angajaților care lucrează în același oraș cu managerul lor, împreună cu numele orașului respectiv. Utilizați mai multe joinuri pentru a rezolva această problemă.

```
SELECT e1.last_name AS "Angajat",
       e2.last_name AS "Manager",
       l.city AS "Oraș"
FROM employees e1
JOIN employees e2 ON e1.manager_id = e2.employee_id
JOIN departments d1 ON e1.department_id = d1.department_id
JOIN departments d2 ON e2.department_id = d2.department_id
```

```
JOIN locations l ON d1.location_id = l.location_id  
WHERE d1.location_id = d2.location_id  
ORDER BY l.city, e2.last_name, e1.last_name;
```

Această interogare complexă folosește mai multe joinuri pentru a:

- Uni angajații cu managerii lor (SELF JOIN)
- Asocia fiecare angajat și manager cu departamentul său
- Obține locațiile departamentelor
- Filtra pentru a găsi doar perechile angajat-manager care lucrează în același oraș