

BASE DE DADOS



Teórico-Práticas

**SQL – SELECT
Subqueries**

Linguagem SQL–Select-Subqueries

Sumário:

- Subqueries
 - Conceitos básicos (recordar da aula Teórica)
 - Exemplos

Linguagem SQL–Select-Subqueries

- Comando SELECT
 - Sintaxe

```
SELECT [ ALL | DISTINCT ] <select_list>  
FROM <table_list>  
[WHERE <where_expression>]  
[GROUP BY <groupby_list>]  
[HAVING <having_expression>]  
[ORDER BY <orderby_list> [DESC|ASC]]
```

Linguagem SQL–Select-Subqueries

- Subquery (subconsulta) (1/2)
 - É uma instrução SELECT (interior)
 - Está encaixada numa das seguintes cláusulas de outra instrução SELECT (exterior):
 - SELECT
 - FROM
 - WHERE
 - HAVING
- Pode retornar
 - Uma linha => pode ser usada em operação aritmética
 - Múltiplas linhas => pode ser usada em operadores: IN, ANY(SOME), ALL, EXISTS

Linguagem SQL–Select-Subqueries

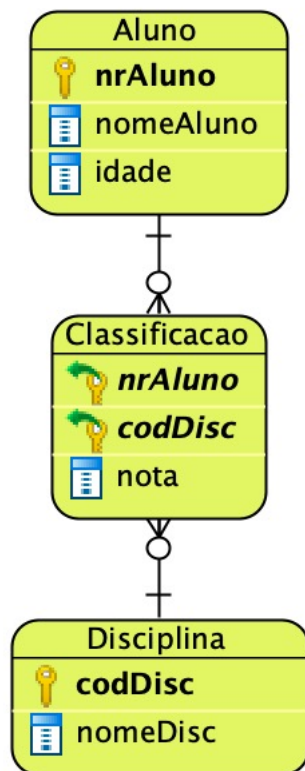
- Subquery (subconsulta) (2/2)
 - Execução
 - Depende do tipo:
 - Não-Correlacionada
 - Correlacionada

| Tipo de SubQuery | Select Interior | Sentido de Execução | Execução do select interior |
|-------------------------|--------------------------------|---|---|
| Não correlacionada | Não depende do select exterior | 1º o select Interior 2º select Exterior | Uma vez |
| Correlacionada | Depende do select exterior | Execução do interior intercalada com o exterior | nr. vezes = ao numero de vezes do select exterior |

Linguagem SQL–Select-Subqueries

Consultas de dados numa BDR

- Dado o UoD/domínio “notas de disciplinas” representado pelo seguinte modelo de dados relacional:



- Instância da BDR:

| | NRALUNO | NOMEALUNO | IDADE |
|---|---------|-----------|-------|
| 1 | 1 | Ana | 30 |
| 2 | 2 | João | 35 |
| 3 | 3 | Maria | 40 |
| 4 | 4 | Rui | 30 |
| 5 | 5 | Joana | 35 |
| 6 | 6 | José | 35 |
| 7 | 7 | Rita | 40 |
| 8 | 8 | Camilo | 33 |

| | CODDISC | NOMEDISC |
|---|---------|-------------|
| 1 | info | Informática |
| 2 | mate | Matemática |
| 3 | hist | História |
| 4 | ingl | Inglês |

| | NRALUNO | CODDISC | NOTA |
|----|---------|---------|------|
| 1 | 1 | hist | 16 |
| 2 | 1 | info | 15 |
| 3 | 1 | ingl | 18 |
| 4 | 1 | mate | 13 |
| 5 | 2 | hist | 14 |
| 6 | 2 | info | 18 |
| 7 | 2 | ingl | 16 |
| 8 | 2 | mate | 16 |
| 9 | 3 | hist | 17 |
| 10 | 3 | info | 18 |
| 11 | 3 | ingl | 19 |
| 12 | 3 | mate | 19 |
| 13 | 4 | hist | 12 |
| 14 | 4 | info | 14 |
| 15 | 4 | ingl | 13 |
| 16 | 5 | hist | 14 |
| 17 | 5 | info | 12 |
| 18 | 5 | ingl | 13 |
| 19 | 6 | hist | 16 |
| 20 | 6 | info | 18 |
| 21 | 6 | mate | 17 |
| 22 | 7 | hist | 12 |
| 23 | 7 | info | 14 |
| 24 | 7 | mate | 15 |
| 25 | 8 | hist | 15 |
| 26 | 8 | info | 14 |
| 27 | 8 | ingl | 14 |
| 28 | 8 | mate | 10 |

Linguagem SQL–Select-Subqueries

1. Mostrar o número, o nome e a idade dos alunos mais novos, usando duas estratégias diferentes:
 - a) Com função de agregação MIN
 - b) Com o operador ALL

```
--1.a)
SELECT nrAluno, nomealuno, idade
  FROM aluno
 WHERE idade = (SELECT MIN(idade) FROM aluno);
```



```
--1.b)
SELECT nrAluno, nomeAluno, idade
  FROM aluno
 WHERE idade <= ALL(SELECT idade FROM aluno);
```

| | NRALUNO | NOMEALUNO | IDADE |
|---|---------|-----------|-------|
| 1 | 1 | Ana | 30 |
| 2 | 4 | Rui | 30 |

Linguagem SQL–Select-Subqueries

2. Mostrar o número e o nome dos alunos que não têm nota a todas as disciplinas.

```
SELECT a.nrAluno, a.nomeAluno
  FROM aluno a
 INNER JOIN classificacao c ON c.nrAluno=a.nrAluno
  GROUP BY a.nrAluno, a.nomeAluno
  HAVING COUNT(c.codDisc) <> (SELECT COUNT(*) FROM disciplina)
 ORDER BY 1;
```

```
SELECT a.nrAluno, a.nomeAluno
  FROM aluno a
 WHERE a.nrAluno IN (
    SELECT c.nrAluno
      FROM classificacao c
     GROUP BY c.nrAluno
    HAVING COUNT(c.codDisc) <> (SELECT COUNT(*) FROM disciplina))
 ORDER BY 1;
```

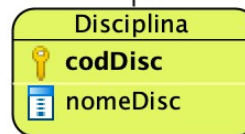
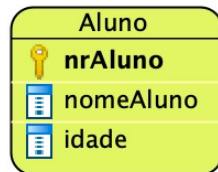


| | NRALUNO | NOMEALUNO |
|---|---------|-----------|
| 1 | | 4 Rui |
| 2 | | 5 Joana |
| 3 | | 6 José |
| 4 | | 7 Rita |

Linguagem SQL–Select-Subqueries

3. Mostrar o número e o nome dos alunos que não têm nota da disciplina de matemática.

```
SELECT a.nrAluno, a.nomeAluno
  FROM aluno a
 WHERE a.nrAluno NOT IN (
    SELECT c.nrAluno
      FROM classificacao c
 INNER JOIN disciplina d ON d.codDisc=c.codDisc
    WHERE UPPER(d.nomeDisc) LIKE 'MATEMÁTICA');
```



| | NRALUNO | NOMEALUNO |
|---|---------|-----------|
| 1 | 4 | Rui |
| 2 | 5 | Joana |

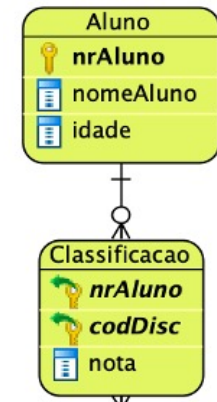
```
SELECT a.nrAluno, a.nomeAluno
  FROM aluno a
 WHERE a.nrAluno NOT IN (
    SELECT c.nrAluno
      FROM classificacao c
    WHERE c.codDisc = (
      SELECT d.codDisc
        FROM disciplina d
      WHERE UPPER(d.nomeDisc) LIKE 'MATEMÁTICA'));
```

Linguagem SQL–Select-Subqueries

4. Mostrar a menor nota de cada aluno (número e nome), com ordenação decrescente.

```
SELECT a.nrAluno,  
       a.nomeAluno,  
       (SELECT MIN(nota)  
        FROM classificacao c  
        WHERE c.nrAluno=a.nrAluno) "menor nota"  
FROM aluno a  
ORDER BY 3 DESC;
```

```
SELECT a.nrAluno, a.nomeAluno, MIN(c.nota)  
FROM aluno a  
INNER JOIN classificacao c ON c.nrAluno=a.nrAluno  
GROUP BY a.nrAluno, a.nomeAluno  
ORDER BY 3 DESC;
```

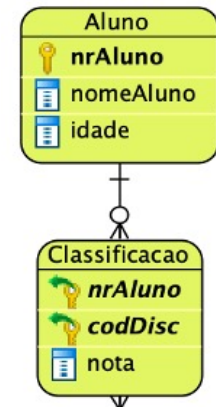


| | NRALUNO | NOMEALUNO | menor nota |
|---|---------|-----------|------------|
| 1 | 3 | Maria | 17 |
| 2 | 6 | José | 16 |
| 3 | 2 | João | 14 |
| 4 | 1 | Ana | 13 |
| 5 | 5 | Joana | 12 |
| 6 | 4 | Rui | 12 |
| 7 | 7 | Rita | 12 |
| 8 | 8 | Camilo | 10 |

Linguagem SQL–Select-Subqueries

5. Mostrar o número e o nome dos alunos que têm todas as notas superiores a 15.

```
SELECT a.nrAluno, a.nomeAluno
FROM aluno a
WHERE 15 <= ALL ( SELECT nota
                  FROM classificacao c
                  WHERE c.nrAluno=a.nrAluno);
```



| | NRALUNO | NOMEALUNO |
|---|---------|-----------|
| 1 | 3 | Maria |
| 2 | 6 | José |

Linguagem SQL–Select-Subqueries

6. Mostrar a quantidade de alunos que têm nota às disciplinas de matemática e de inglês.

```
SELECT COUNT(*) "Qtd Alunos"
  FROM (    SELECT c.nrAluno
            FROM classificacao c
            WHERE c.codDisc = (SELECT d.codDisc
                               FROM disciplina d
                               WHERE UPPER(d.nomeDisc) LIKE 'MATEMÁTICA')

  INTERSECT
  SELECT c.nrAluno
  FROM classificacao c
  WHERE c.codDisc = (SELECT d.codDisc
                     FROM disciplina d
                     WHERE UPPER(d.nomeDisc) LIKE 'INGLÊS'));
```

```
SELECT COUNT(*) "Qtd Alunos"
  FROM aluno a
 WHERE EXISTS (    SELECT *
                  FROM classificacao c1
                  INNER JOIN classificacao c2 ON c2.nrAluno=c1.nrAluno
                  WHERE c1.codDisc = (SELECT d.codDisc
                                       FROM disciplina d
                                       WHERE UPPER(d.nomeDisc) LIKE 'MATEMÁTICA')
                  AND c2.codDisc = (SELECT d.codDisc
                                     FROM disciplina d
                                     WHERE UPPER(d.nomeDisc) LIKE 'INGLÊS')
                  AND c1.nrAluno=a.nrAluno);
```



| Qtd Alunos | |
|------------|---|
| 1 | 4 |