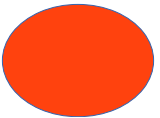


# Modelo Relacional

---

## SQL – Consultas aninhadas 2

Prof. André Britto



# Consultas aninhadas (subselect)

---

- Podemos utilizar consultas aninhadas e assim construir consultas mais complexas.
- De forma simples:  
SELECT <campo>  
FROM <tabela>  
WHERE <condição>
- Podes colocar uma consulta dentro
- das três cláusulas.

# Operador IN

---

- O operador IN define se um valor faz parte de uma coleção de valores.
- Sintaxe:
  - atributo(s) IN(R) – Retorno é true ou false.
    - É possível utilizar também NOT IN.
    - A cardinalidade dos atributos a esquerda do IN deve ser a mesma do conjunto a direita do IN
  - R é um conjunto de valores, podendo ser obtido através de uma consulta SQL.

# Operador IN

---

- Listar todas as turmas das disciplinas COMP0197, COMP198 e COMP0199

# Operador IN

---

- Listar todas as turmas das disciplinas COMP0197, COMP198 e COMP0199

```
SELECT *
```

```
FROM turma t JOIN disciplina d USING(cod_disc)
```

```
WHERE cod_disc IN ('COMP0197', 'COMP0198', 'COMP0199')
```

# Operador IN

---

- Listar todas as turmas das disciplinas COMP0197, COMP0198 e COMP0199 lecionados pelo professor P200

# Operador IN

---

- Listar todas as turmas das disciplinas COMP0197, COMP198 e COMP0199 lecionados pelo professor P100

```
SELECT *
```

```
FROM turma t JOIN disciplina d USING(cod_disc)
```

```
JOIN leciona l USING (id_turma)
```

```
WHERE (cod_disc,mat_professor) IN (('COMP0197','P200'),  
('COMP0198','P200'), ('COMP0199','P200'))
```

# Operador IN

---

- Listar a cpf dos professores que lecionam alguma disciplina



# Operador IN

---

- Listar o cpf dos professores que lecionam alguma disciplina

```
SELECT cpf
```

```
FROM professor
```

```
WHERE mat_professor IN (SELECT mat_professor FROM  
leciona)
```

# Operador IN

---

- Para todo nome da consulta mas externa, o operador IN testa se esse valor existe na consulta interna.
- Retorna *true* se existe ou *false* se não existe.

# Operador IN

---

- Listar a matricula dos professores que não lecionam alguma disciplina

```
SELECT cpf
```

```
FROM professor
```

```
WHERE mat_professor NOT IN (SELECT mat_professor  
FROM leciona)
```

# Operador EXISTS

---

- SQL permite testarmos se o resultado de uma subconsulta possui alguma tupla ou não.
- O operador EXISTS retorna *true* se a subconsulta é não vazia.
- Sintaxe:  
EXISTS(R)

# Operador EXISTS

---

Liste todos os estudantes, se existir algum registro na tabela cursa

```
SELECT e.mat_estudante
```

```
FROM estudante AS e
```

```
WHERE EXISTS(SELECT * FROM cursa)
```

# Consultas correlacionadas

---

- É possível utilizar nomes das consultas mais externas, nas consultas mais internas.
- Operador EXISTS fica mais poderoso com essa opção.
- Podemos checar se um valor da consulta externa existe na consulta interna.

# Consultas correlacionadas

---

- Liste todos os alunos que estão matriculados em alguma disciplina:

# Consultas correlacionadas

---

- Liste todos os alunos que estão matriculados em alguma disciplina:

```
SELECT e.mat_estudante
```

```
FROM estudante e
```

```
WHERE EXISTS (SELECT * FROM cursa c
```

```
WHERE e.mat_estudante = c.mat_estudante)
```



# ALL e SOME

---

- Podemos checar se um valor, ou uma tupla de valores, possui alguma relação de quantidade com o conjunto retornado pela consulta.
  - Podemos verificar se essa relação se aplica a alguns elementos ou se ela é verdade para todos os elementos.

# ALL e SOME

---

- Considere que queremos todos os professores que recebem um salário maior do que média de salários por departamento.

# ALL e SOME

---

- Média de salários por departamento

```
SELECT avg(salario)
```

```
FROM professor pi JOIN cargo c ON (pi.cargo = r.id_cargo)
```

```
GROUP BY departamento
```

# ALL e SOME

---

- Valores de salário maior que **alguma** média

SELECT \*

FROM professor p JOIN cargo c ON (p.cargo = c.id\_cargo)

WHERE c.salario > **SOME** (SELECT avg(salario)

FROM professor p JOIN cargo c ON (p.cargo = c.id\_cargo)

GROUP BY departamento)

# ALL e SOME

---

- Valores de salário maior que **todas** as média
- **SELECT \***
- **FROM professor p JOIN cargo c ON (p.cargo = c.id\_cargo)**
- **WHERE c.salario > ALL** (SELECT avg(salario)
- **FROM professor p JOIN cargo c ON (p.cargo = c.id\_cargo)**
- **GROUP BY departamento)**

# Consultas correlacionadas

---

- Podemos restringir a consulta interna aos valores da consulta externa.
- Ex: professores que recebem mais do que a média de salários do departamento que ele faz parte.
  - Tenho que garantir que a média calculada seja a do departamento do professor.

# Consultas correlacionadas

---

SELECT \*

FROM professor p JOIN cargo c ON (p.cargo = c.id\_cargo)

WHERE c.salario > (SELECT avg(salario)

FROM professor pi JOIN cargo c ON (pi.cargo = c.id\_cargo)

WHERE p.departamento = pi.departamento )

Por que não usamos o ALL ou SOME nessa situação?

## Leitura recomendada

---

ELMASRI, R; NAVATHE, S.B. **Sistemas de Banco de Dados**, Addison Wesley, 6ª Edição.

- Capítulo 4 e 5

Silberschatz, A; Korth H.F.; Sudarshan S. **Sistemas de Banco de Dados**, Editora Campus, 6ª Edição.

- Capítulo 3, 4 e 5

