

## 5. SQL

### - Conjunto de Operações para Manipulação de Dados -

→ Comparações envolvendo NULL e lógica de três valores:

→ Quando NULL está envolvido em uma operação de comparação, o resultado é considerado UNKNOWN (ou desconhecido) e pode ser TRUE ou FALSE:

Conectivos lógicos na lógica de três valores.

<b>(a) AND</b>	TRUE	FALSE	UNKNOWN
TRUE	TRUE	FALSE	UNKNOWN
FALSE	FALSE	FALSE	FALSE
UNKNOWN	UNKNOWN	FALSE	UNKNOWN

<b>(b) OR</b>	TRUE	FALSE	UNKNOWN
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	UNKNOWN
UNKNOWN	TRUE	UNKNOWN	UNKNOWN

<b>(c) NOT</b>			
TRUE	FALSE		
FALSE	TRUE		
UNKNOWN	UNKNOWN		

## 5. SQL

- Conjunto de Operações para Manipulação de Dados -

### ❑ Checando valores nulos

↳ Predicado IS NULL

↳ Exemplo

```
select * from Empregado  
where dt-nasc is null
```

↳ Predicado IS NOT NULL

↳ Exemplo

```
select * from Empregado  
where dt-nasc is not null
```

## 5. SQL

- Conjunto de Operações para Manipulação de Dados -

### ❑ Consulta SQL aninhada (subconsulta)

↳ Consulta SQL especificada dentro de uma outra consulta SQL

↳ Exemplo

⇒ Listar todos os empregados que têm salário maior que a média salarial da empresa

```
select e.nome  
from Empregado e  
where salário > (select avg(salário) from Empregado)
```

subconsulta retorna um conjunto de valores para a consulta mais externa (executada primeiro)

## 5. SQL

- Conjunto de Operações para Manipulação de Dados -

### ❑ Consulta SQL aninhada (cont.)

⇒ Listar o primeiro e segundo maiores salários da empresa

```
select max(salário)
from Empregado
union
select max(salário)
from Empregado
where salário <> (select max(salário) from Empregado)
```

⇒ Listar nome dos departamentos com média salarial maior que a média salarial da empresa

```
select d.nome
from Departamento d, Empregado e
where d.cod_depart=e.lotação
group by d.nome
having avg(e.salário) > (select avg(salário) from Empregado)
```

subconsulta na cláusula *having*

## 5. SQL

### - Conjunto de Operações para Manipulação de Dados -

➡ Uso de Consulta SQL aninhada com “Group By” e Funções Agregadas:

➡ Exemplo:

➡ De acordo com a Tabela SomeTable (dada abaixo), selecionar o código da marca com o maior somatório de marcas da tabela mostrada:

**SomeTable**

CODE	Marks
A	50
A	50
B	60
B	60
C	20
C	20

**Consulta SQL**

```
SELECT Code, MAX(SumMarks)
FROM
  (SELECT
    SUM(Marks) as SumMarks, Code
  FROM SomeTable
  GROUP BY Code
  ) foo
GROUP BY Code
```

## 5. SQL

- Conjunto de Operações para Manipulação de Dados -

### ❏ Consulta SQL aninhada (cont.)

#### ↳ Consulta correlacionada

⇒ Listar todos os empregados que possuem salário maior que a média salarial de seus departamentos

```
select nome  
from Empregado  
where salário > (select avg(salário) from Empregado  
                 where lotação=???? )
```

A subconsulta precisa utilizar o valor do atributo **lotação** de cada tupla da consulta mais externa como parâmetro de entrada

⇒ Variável de correlação

⇒ Variável da consulta mais externa utilizada pela consulta mais interna

```
select e.nome  
from Empregado e  
where salário > (select avg(salário) from Empregado  
                 where lotação=e.lotação)
```

Executado para cada tupla da consulta mais externa

## 5. SQL

- Conjunto de Operações para Manipulação de Dados -

### ❑ Consulta SQL aninhada (cont.)

#### → Predicado IN

⇒ Verifica a pertinência de elementos em um conjunto

⇒ Exemplo

```
select nome  
from Empregado  
where matr in (1,5,8,9)
```

⇒ Listar os empregados lotados nos departamentos localizados em Fortaleza, considerando o seguinte esquema

Departamento(cod\_depart, nome, ender, cidade)  
Empregado(matr, nome, ender, cpf, salário, lotação)

```
select nome  
from Empregado  
where lotação in (select cod_depart from Departamento  
where cidade='Fortaleza' )
```



## 5. SQL

- Conjunto de Operações para Manipulação de Dados -

### ❑ Consulta SQL aninhada (cont.)

↳ Predicado IN (cont.)

⇒ Sintaxe

⇒  $\text{expr} [\text{NOT}] \text{IN} (\text{subconsulta}) \mid \text{expr} [\text{NOT}] \text{IN} (\text{val} [, \text{val} \dots])$

↳ Predicados SOME, ANY e ALL

⇒ Listar empregados que ganham salário maior ou igual a média salarial de um departamento

⇒ Sintaxe

⇒  $\text{expr} \theta \{\text{SOME} \mid \text{ANY} \mid \text{ALL}\} (\text{subconsulta})$

⇒  $\theta \in \{<, <=, >, >=, =, <>\}$

```
select nome
from Empregado
where salário >=some (select avg(salário) from Empregado
                      group by lotação)
```

OBS: “NOT IN” não é um alias para “<> ANY”, mas sim para “<> ALL”.



## 5. SQL

- Conjunto de Operações para Manipulação de Dados -

### ❑ Consulta SQL aninhada (cont.)

↳ Predicados SOME, ANY e ALL (cont.)

⇒  $\theta$ SOME (subconsulta) e  $\theta$ ANY (subconsulta)

⇒ Retornam verdade se e somente se

♦ Para pelo menos um elemento  $s$  retornado pela subconsulta, **expr  $\theta$  s é verdade**

⇒ São equivalentes

⇒  $\theta$ ALL (subconsulta)

⇒ Retorna verdade se e somente se,

♦ Para todo elemento  $s$  retornado pela subconsulta, **expr  $\theta$  s é verdade**

⇒ Listar empregados que ganham salários maior ou igual a média salarial de cada departamento

```
select nome
from Empregado
where salário >=all (select avg(salário) from Empregado
group by lotação)
```

## 5. SQL

- Conjunto de Operações para Manipulação de Dados -

### ❑ Consulta SQL aninhada (cont.)

↳ Predicados SOME, ANY e ALL (cont.)

⇒ Listar o departamento com maior média salarial

⇒ Não é permitido função agregada composta

```
select d.nome  
from Empregado e, Departamento d  
where d.cod_depart=e.lotação  
group by d.nome  
having avg(salário) >=all (select avg(salário) from Empregado  
group by lotação)
```

↳ Predicado EXISTS

⇒ Verifica se o conjunto retornado por um subconjunto é vazio ou não

⇒ Listar empregados que possuem dependentes

```
select e.nome  
from Empregado e  
where exists (select * from Dependente where matr_resp=e.matr)
```

## 5. SQL

- Conjunto de Operações para Manipulação de Dados -

### ❏ Consulta SQL aninhada (cont.)

#### ↳ Predicado EXISTS (cont.)

##### ⇒ Sintaxe

⇒ [NOT] EXISTS (subconsulta)

##### ⇒ EXISTS (subconsulta)

⇒ Retorna verdade se e somente se

‣ O conjunto retornado por subconsulta não é vazio

##### ⇒ NOT EXISTS (subconsulta)

⇒ Retorna verdade se e somente se

‣ O conjunto retornado por subconsulta é vazio

⇒ Listar nome de departamentos com empregados ganhando duas vezes mais que a média do departamento

```
select d.nome  
from Departamento d  
where exists (select * from Empregado e  
              where e.lotacao=d.cod_depart and  
                    salário > (2*(select avg(salário) from Empregado  
                                   where lotação=e.lotação)))
```

## 5. SQL

- Conjunto de Operações para Manipulação de Dados -

### ❑ Consulta SQL aninhada (cont.)

↳ Subconsulta escalar

⇒ Subconsulta que retorna apenas um valor

⇒ Pode aparecer na lista de argumentos da cláusula *select* e na cláusula *where*

⇒ Exemplo

⇒ Listar nome dos empregados com a quantidade de dependentes de cada empregado

```
select e.nome, (select count(*) From dependente where matr_resp=e.matr)
from Empregado e
```

### ❑ Predicado Between

↳ Sintaxe

⇒ *expr1* [NOT] BETWEEN *expr2* and *expr3*

⇒ Exemplo

⇒ *matr* between 2 and 10  $\Leftrightarrow$  *matr*  $\geq$  2 and *matr*  $\leq$  10