

# BANCO DE DADOS

## SUB-CONSULTA E UNION

RICARDO SONAGLIO ALBANO



# SQL - SUBCONSULTA E UNION

## TIPOS DE SUBCONSULTA - SUBQUERY

- Com uma linha
- Operador any / some
- Operador in
- No having
- Correlacionada
- Com operador exists
- Intersect (interseção)
- Except (diferença)
- Divisão



# SQL - SUBCONSULTA E UNION

## SUBQUERY com uma linha

- São comandos SELECT aninhados em outros comandos SELECT, INSERT, UPDATE ou DELETE ou outras SUB-CONSULTAS (SUBQUERIES);
- Muitas sentenças SQL que incluem SUBQUERIES podem ser formuladas de modo alternativo através de JOINS;
- A sentença SELECT de uma SUBQUERY é sempre colocada dentro de parênteses;
- Pode estar inserida na lista de SELECT, numa cláusula WHERE ou HAVING da query externa;



# SQL - SUBCONSULTA E UNION

## SUBQUERY com uma linha

- Qual o funcionário que tem o maior salário?

```
select fun_id, fun_nome, fun_salario  
from funcionario  
where fun_salario = (select max(fun_salario) from funcionario);
```



# SQL - SUBCONSULTA E UNION

## SUBQUERY com o operador ALL

- Qual funcionário ganha mais que a média salarial de todos os departamentos?

```
select fun_id, fun_nome, fun_salario  
from funcionario  
where fun_salario > all  
(select avg(fun_salario) from funcionario group by dep_id);
```



# SQL - SUBCONSULTA E UNION

## SUBQUERY com o operador ANY

- Que funcionário ganha mais que a média salarial de algum departamento?

```
select fun_id, fun_nome, fun_salario  
from funcionario  
where fun_salario > any  
(select avg(fun_salario) from funcionario group by dep_id);
```



# SQL - SUBCONSULTA E UNION

## SUBQUERY com o operador IN

- Mostrar todos os funcionários que pertencem aos departamentos subordinados ao departamento 'Reitoria academica' (dep\_id = 2)?

```
select fun_id, fun_nome, dep_id  
from funcionario  
where dep_id in (select dep_id  
                 from departamento where dep_Superior = 2);
```



# SQL - SUBCONSULTA E UNION

## SUBQUERY HAVING

- Quais departamentos tem média salarial maior que a media salarial da empresa?

```
select dep_id, avg(fun_salario)
from funcionario
group by dep_id
having avg(fun_salario) > (select avg(fun_salario) from funcionario);
```



# SQL - SUBCONSULTA E UNION

## SUBQUERY CORRELACIONADA

- Liste os nomes dos empregados cujo salário seja maior que a média salarial do seu departamento?

```
select fun_nome, fun_salario, dep_id
from funcionario f
where fun_salario > (select avg(fun_salario)
                     from funcionario
                     where dep_id = f.dep_id);
```



# SQL - SUBCONSULTA E UNION

## SUBQUERY: o operador EXISTS

- Retorna VERDADEIRO ou FALSO;
- É um operador que tem como argumento um SELECT;
- A SUBQUERY não devolve nenhuma tabela resultante;
- EXISTS pode ser negado por um NOT;
- Exemplo: Caso existam funcionarios ganhando mais de 10000, há algum funcionario recebendo menos de 2.000??

```
select fun_nome, fun_salario from funcionario
where fun_salario < 2000
And exists (select * from funcionario where fun_salario > 10000);
```

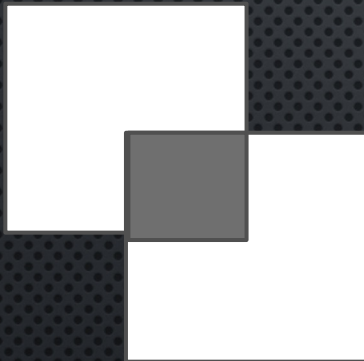


# SQL - SUBCONSULTA E UNION

## SUBQUERY: OPERADOR INTERSECT (Interseção)

- Quais departamentos possuem funcionários?

**Departamento**



**Funcionario**

```
select dep_id from Departamento;  
intersect  
select dep_id from funcionario;
```

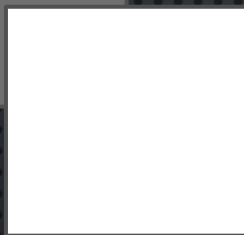


# SQL - SUBCONSULTA E UNION

## SUBQUERY: OPERADOR EXCEPT (Diferença)

- Existe algum departamento sem funcionários?

**Departamento**



**Funcionario**

```
select dep_id from Departamento;  
except  
select dep_id from funcionario;
```



# SQL - SUBCONSULTA E UNION

## SUBQUERY: OPERADOR EXCEPT (Diferença)

- Quais funcionários tiraram férias em todos os anos de emprego na empresa?

```
select fun_id, fun_nome
from funcionario f
where not exists
    (select distinct year(fer_periodo_aquisitivo ) from ferias
    except
    select year(fer_periodo_aquisitivo )
    from ferias where fer_matricula = f.fun_id)
```



# SQL - SUBCONSULTA E UNION

## UNION e UNION ALL

- Consultar informações pertencentes a mais de uma tabela e que são obtidas com a execução de mais de um comando SELECT;
- Regras:
  - As colunas devem ser do mesmo tipo, ou seja, (string com string, int com int...);
  - É necessário que os comandos SELECT apresentem o mesmo número de colunas;
  - Elimina linhas duplicadas do resultado.
  - A inclusão de WHERE pode ser feita em qualquer um dos comandos SELECT;
  - O apelido (alias) das colunas, quando necessário, deve ser incluído no primeiro SELECT;
  - Para dados ordenados, o último SELECT deve ter uma cláusula ORDER BY;
  - UNION ALL: não elimina as linhas duplicadas.



# SQL - SUBCONSULTA E UNION

## UNION

- Exemplos:

```
select fun_nome from funcionario
union
select pro_nome from professor
union
select alu_nome from aluno
```

```
select fun_nome, 'funcionario' from funcionario
union
select pro_nome, 'professor' from professor
union
select alu_nome, 'aluno' from aluno
```

```
select fun_nome, fun_nascimento, 'funcionario' from funcionario;
union
select pro_nome, pro_nascimento, 'professor' from professor;
union
select alu_nome, alu_nascimento, 'aluno' from aluno order by 2;
```



**Obrigado por sua atenção!**