

# **Subconsulta**

Professora: Juliana

19-9-2024 - SubConsulta

# Roterio:

- Preparação do Ambiente
- Subconsulta (uma linha)
- Subconsulta (várias linhas):
  - Exists;
  - Subquery múltiplas linhas (ALL, ANY e IN);
  - Expressão Case.

## ***Preparação do Ambiente***

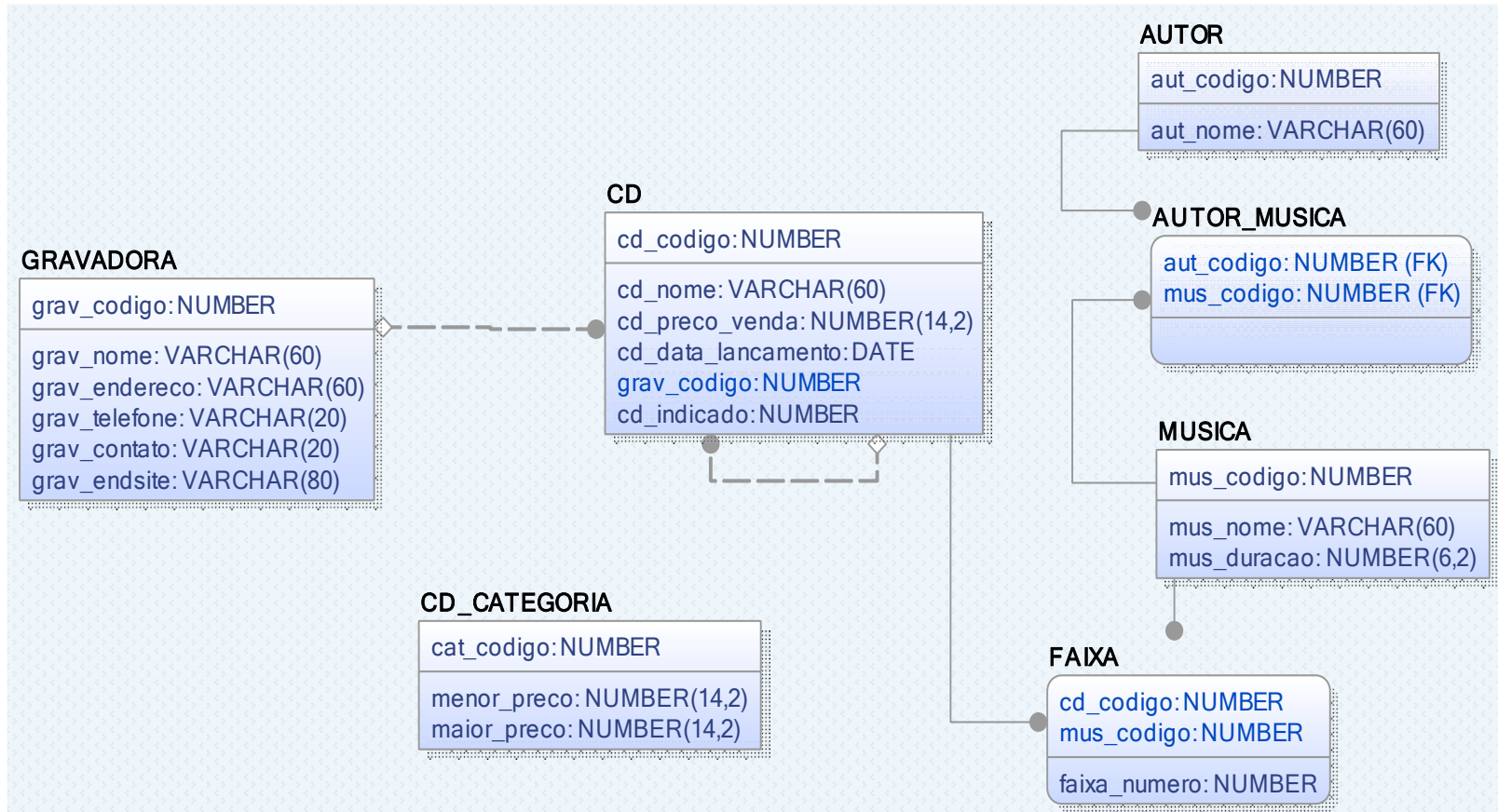
### ***➤ Logar com o usuário system e executar:***

- Cria\_tablespace\_usuario.sql;

### ***➤ Logar com o usuário USER\_catalogo, senha: aluno e executar:***

- catalogo-DDL;
- catalogo-DML.

# Catálogo de CDs



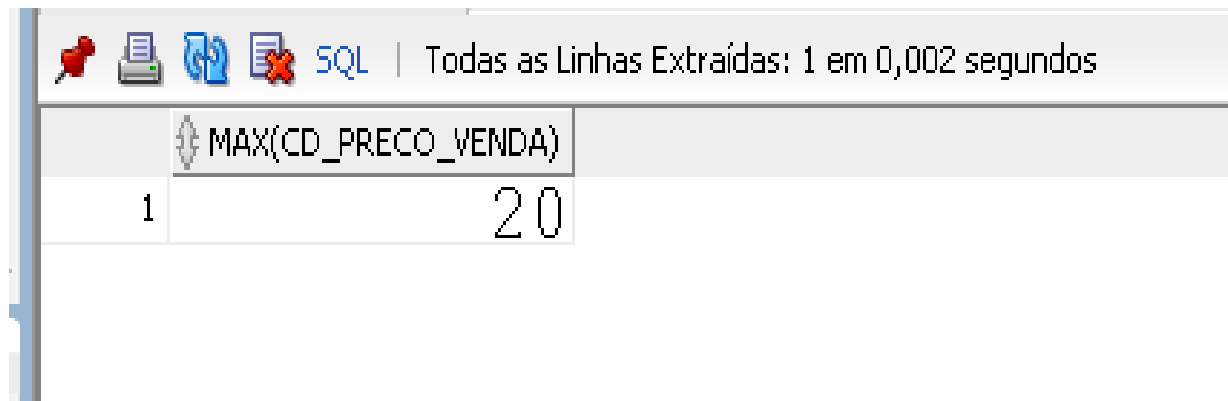
# Subqueries de Uma linha

- Os operadores utilizados em subquery de uma linha são:

=, <>, > >=, <, <=

# Qual o maior preço dos Cds?

```
select max(cd_preco_venda)  
from cd;
```

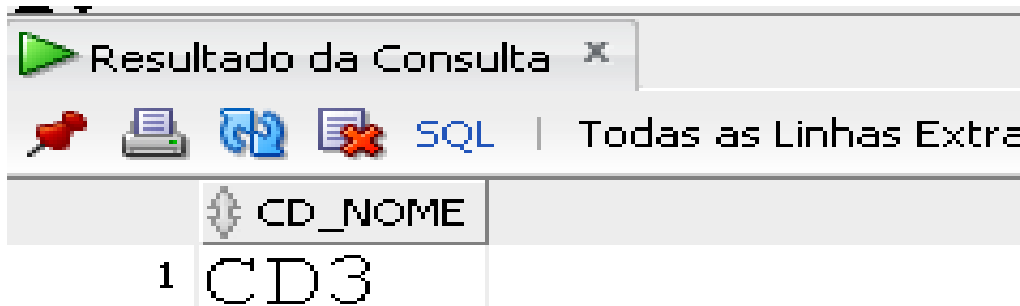


The screenshot shows a database query result in a software interface. The top bar contains icons for a pin, printer, copy, and delete, followed by the text 'SQL | Todas as Linhas Extraídas: 1 em 0,002 segundos'. Below this is a table with one column labeled 'MAX(CD\_PRECO\_VENDA)' and one row with the value '20'.

MAX(CD_PRECO_VENDA)
20

Liste o nome do CD de maior preço?

```
Select cd_nome from cd  
where cd_preco_venda=(select  
max(cd_preco_venda) from cd);
```



Resultado da Consulta	
SQL   Todas as Linhas Extra	
	CD_NOME
1	CD3

Criar uma tabela na qual constem os cds cujo o preço seja menor que a media.

```
create table cd_menor as  
select cd_nome, cd_preco_venda  
from cd where cd_preco_venda <  
    (select avg(cd_preco_venda) from cd);
```



# Exists

- Quais as gravadoras que tem CDs cadastrados?

```
Select grav_codigo,grav_nome from gravadora  
where EXISTS (select * from CD  
where cd.grav_codigo= gravadora.grav_codigo);
```

Resolver sem utilizar subconsulta...

```
select distinct g.grav_codigo,g.grav_nome  
from gravadora g, cd c  
where g.grav_codigo=c.grav_codigo;
```

```
select distinct g.grav_codigo,g.grav_nome  
from gravadora g inner join cd c  
on g.grav_codigo=c.grav_codigo
```

# IN

- Quais os CDs que têm o preço igual ao menor preço de cada gravadora?

```
Select cd_codigo, cd_nome, cd_preco_venda  
from CD  
where cd_preco_venda in  
(select min(cd_preco_venda)  
from CD  
group by grav_codigo);
```

# ANY

- Quais os CDs que têm preço inferior a qualquer um dos cds da gravadora com código 1?

```
Select cd_codigo, cd_nome, cd_preco_venda
```

```
From CD
```

```
Where cd_preco_venda < any (
```

```
Select cd_preco_venda
```

```
From CD
```

```
Where grav_codigo=1)
```

verdadeiro se o atributo comparado for  
menor que algum valor de atributo das tuplas resultantes da subconsulta

# ALL

- Quais CDs têm o preço de venda menor que todas as médias de preços de vendas por gravadoras?

```
Select cd_codigo, cd_nome, cd_preco_venda  
from CD  
where cd_preco_venda < ALL (  
select AVG(cd_preco_venda)  
from CD  
group by grav_codigo);
```

# Expressões CASE

**SINTAXE:**

```
SELECT colunas,  
  CASE  
    WHEN condição THEN ação  
  ...  
  [ELSE condição padrão]  
  END  
FROM tabela;
```

# CASE - SELECT

- Calcular desconto sobre o preço de venda dos CDs em função do preço.

```
SELECT cd_nome, cd_preco_venda,  
CASE  
WHEN cd_preco_venda < 10 then cd_preco_venda * 0.10  
WHEN cd_preco_venda >=10 and cd_preco_venda < 13 then cd_preco_venda * 0.20  
ELSE cd_preco_venda * 0.30  
END VENDA  
FROM CD;
```

Saída do Script

Resultado da Consulta

SQL

Todas as Linhas Extraídas: 3 em 0,015 seg

	CD_NOME	CD_PRECO_VENDA	VENDA
1	CD1	10	2
2	CD2	12	2,4
3	CD3	20	6

# CASE - UPDATE

```
UPDATE CD SET cd_preco_venda =  
CASE  
  WHEN cd_preco_venda < 10 then cd_preco_venda * 1.10  
  WHEN cd_preco_venda >=10 and cd_preco_venda < 13 then cd_preco_venda * 1.20  
  ELSE cd_preco_venda * 1.30  
END;  
  
Commit;
```



# Edit-> preferences-> SQLEditor

