PL5 - SQL DML SELECT - QUERIES COM AGRUPAMENTO DE REGISTOS

Category	Exercícios
Files	PL04B.S1-CREATE e ALTER TABLE.sql PL04B.S2-INSERT.sql PL04B.zip PL04B-SQL-DML-SELECT-Agrupamento-Ex CD de Musica 2.pdf

▼ Quick Access

PL

▼ Executar os scripts disponibilizados para criar uma base de dados (BD) de uma coleção de CD de música, baseada no modelo relacional da Figura 1.

▼ PL04B.S1-CREATE e ALTER TABLE.sql

```
-- ** eliminar tabelas se existentes **
-- CASCADE CONSTRAINTS para eliminar as restrições de i
-- PURGE elimina a tabela da base de dados e da "recicl
DROP TABLE cd       CASCADE CONSTRAINTS PURGE;
DROP TABLE musica   CASCADE CONSTRAINTS PURGE;
DROP TABLE editora  CASCADE CONSTRAINTS PURGE;

-- tabela Editora
CREATE TABLE editora (
    idEditora  INTEGER      CONSTRAINT pkEditoraIdEdito
    nome        VARCHAR(20) CONSTRAINT nnEditoraNome NO
);

-- tabela CD
CREATE TABLE cd (
    codCd       INTEGER     CONSTRAINT pkCdCodCd PRIMAR
    idEditora   INTEGER,
    titulo      VARCHAR(40) CONSTRAINT nnCdTitulo NOT N
```

```
    dataCompra  DATE,
    valorPago   NUMERIC(5,2),
    localCompra VARCHAR(20)
);

-- tabela musica
CREATE TABLE musica (
    nrMusica    INTEGER,
    codCd       INTEGER,
    titulo      VARCHAR(40) CONSTRAINT nnMusicaTitulo N
    interprete  VARCHAR(30) CONSTRAINT nnMusicaInterpre
    duracao     NUMERIC(5,2),

    CONSTRAINT pkMusicaNrMusicaCodCd  PRIMARY KEY (codC
);

-- ** alterar tabelas para definição de chaves estrange
ALTER TABLE cd      ADD CONSTRAINT fkCdIdEditora    FOR
ALTER TABLE musica  ADD CONSTRAINT fkMusicaCodCd    FOR

-- ** guardar em DEFINITIVO as alterações na base de da
-- COMMIT;
```

▼ PL04B.S2-INSERT

```
-- ** inserir dados nas tabelas **

Alter session set nls_language = portuguese;

-- tabela Editora
INSERT INTO editora(idEditora, nome) VALUES (1, 'BMG');
INSERT INTO editora(idEditora, nome) VALUES (2, '4AD');
INSERT INTO editora(idEditora, nome) VALUES (3, 'Polydo
INSERT INTO editora(idEditora, nome) VALUES (4, 'Univer
INSERT INTO editora(idEditora, nome) VALUES (5, 'Warner
INSERT INTO editora(idEditora, nome) VALUES (6, 'Island
INSERT INTO editora(idEditora, nome) VALUES (7, 'Parlap
INSERT INTO editora(idEditora, nome) VALUES (8, 'ADF');
```

```
INSERT INTO editora(idEditora, nome) VALUES (9, 'Valent

-- tabela CD
INSERT INTO cd(codCd, idEditora, titulo, dataCompra, va
INSERT INTO cd(codCd, idEditora, titulo, dataCompra, va
INSERT INTO cd(codCd, idEditora, titulo, dataCompra, va
INSERT INTO cd(codCd, idEditora, titulo, dataCompra, va
INSERT INTO cd(codCd, idEditora, titulo, dataCompra, va
INSERT INTO cd(codCd, idEditora, titulo, dataCompra, va
INSERT INTO cd(codCd, idEditora, titulo, dataCompra, va
INSERT INTO cd(codCd, idEditora, titulo, dataCompra, va

-- tabela Musica
--CD 1
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
--CD 2
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
--CD 3
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
--CD 4
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
```

```
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
--CD 5
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
--CD 6
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
--CD 7
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
--CD 8
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
```

```
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,
INSERT INTO musica(nrMusica, codCd, titulo, interprete,

-- ** guardar em DEFINITIVO as alteraï¿½iï¿½es na base d
-- COMMIT;
```

▼ 1) Mostrar a quantidade de CD comprados em cada um dos locais de compra registados

```
SELECT localCompra, COUNT(*) as QUANTIDADE_CD
FROM cd
GROUP BY localCompra;
```

LOCALCOMPRA	QUANTIDADE_CD
iTunes	1
-	2
FNAC	3
Worten	2

▼ `GROUP BY`

- O `GROUP BY` é uma cláusula em SQL usada para agrupar linhas que têm os mesmos valores em determinadas colunas em uma ou mais tabelas. Ele é frequentemente usado em conjunto com funções de agregação (como `COUNT`, `SUM`, `AVG`, etc.) para realizar operações em grupos de linhas.

▼ `COUNT`

- Função de agregação que conta o número de linhas em cada grupo. Neste caso, conta o número de CDs para cada local de compra

---

▼ 2) Copiar e alterar o comando da alínea anterior, de forma a mostrar o resultado por ordem decrescente da quantidade de CD comprados

```
SELECT localCompra, COUNT(*) as QUANTIDADE_CD
FROM cd
GROUP BY localCompra
ORDER BY QUANTIDADE_CD DESC;
```

LOCALCOMPRA	QUANTIDADE_CD
FNAC	3
Worten	2
-	2
iTunes	1

▼ `ORDER BY`

- Usada para ordenar o conjunto de resultados de uma consulta com base em uma ou mais colunas. A ordem pode ser ascendente (`ASC`) ou descendente (`DESC`). Ex:

```
SELECT coluna1, coluna2
FROM sua_tabela
ORDER BY coluna1 ASC;  -- ASC é opcional, pois é a orde
```

- Ainda dá para ordenar várias colunas, especificando a ordem para cada uma delas:

```
SELECT coluna1, coluna2
FROM sua_tabela
ORDER BY coluna1 ASC, coluna2 DESC;
```

- O `ORDER BY` é geralmente a última cláusula na consulta, após o `WHERE` e o `GROUP BY`.

▼ 3) Mostrar a quantidade de editoras diferentes que produziram os CD comprados, em cada um dos locais de compra

```
SELECT cd.localCompra, COUNT(DISTINCT editora.idEditora) A
FROM cd
JOIN editora ON cd.idEditora = editora.idEditora
GROUP BY cd.localCompra;
```

LOCALCOMPRA	QUANTIDADE_EDITORAS
iTunes	1
-	2
Worten	1
FNAC	3

- `JOIN` para combinar as tabelas cd e editora com base na coluna idEditora
- Depois o `COUNT(DISTINCT editora.idEditora)` para contar a quantidade de editoras diferentes (ignorando duplicados) para cada local de compra
- Por fim o `GROUP BY` para agrupar os resultados pelo local de compra

---

▼ 4) Copiar e alterar o comando da alínea 2, de forma a mostrar também, para cada local de compra conhecido, o valor total pago e o maior valor pago.

```
SELECT localCompra,
    COUNT(*) as QUANTIDADE_CD,
    SUM(cd.valorPago) AS VALOR_TOTAL_PAGO,
    MAX(cd.valorPago) AS MAIOR_VALOR_PAGO
FROM cd
```

```
GROUP BY localCompra
ORDER BY QUANTIDADE_CD DESC;
```

LOCALCOMPRA	QUANTIDADE_CD	VALOR_TOTAL_PAGO	MAIOR_VALOR_PAGO
FNAC	3	93.7	55
Worten	2	72.8	42.3
-	2	20.09	10.99
iTunes	1	9.9	9.9

- Foi adicionado o `SUM` para calcular o valor total pago para cada local de compra
- O `MAX` para obter o maior valor pago para cada local de compra

---

▼ 5) Mostrar, para cada CD e respetivos intérpretes, a quantidade de músicas do CD em que o intérprete participa. Além da quantidade referida, também deve ser apresentado o código do CD e o intérprete

```
SELECT
    cd.codCd,
    musica.interprete,
    COUNT(*) as quantidade_de_musicas
FROM cd
JOIN musica ON cd.codCd = musica.codCd
GROUP BY cd.codCd, musica.interprete;
```

CODCD	INTERPRETE	QUANTIDADE_DE_MUSICAS
2	Loring Maazel	1
8	Lana Del Rey	12
2	Zubin Mehta	1
3	Pink Floyd	5
4	Eurythmics	1
1	The Vandals	3
1	Lana Del Rey	1
4	Electric Light Orchestra	1
4	Wham!	1
4	Lana Del Rey	1
4	Kate Bush	1
6	Miguel AraÃ¯Â¿Â½jo	10
4	Fine Young Cannibals	1
5	U2	8
1	AFI	3
1	Rufio	1
7	Xutos e PontapÃ¯Â¿Â½s	10

- Foi usado o `JOIN` para combinar as tabelas cd e musica com base na coluna codCd
- `COUNT` para contar a quantidade de músicas para cada combinação de CD e intérprete
- `GROUP BY` para agrupar os resultados por CD e intérprete

▼ 6) Copiar o comando da alínea 1 e alterar de modo a mostrar apenas os locais de compra com a quantidade superior a 2

```
SELECT localCompra, COUNT(*) as QUANTIDADE_CD
FROM cd
GROUP BY localCompra
HAVING COUNT(*) > 2;
```

LOCALCOMPRA	QUANTIDADE_CD
FNAC	3

▼ `HAVING COUNT`

- Filtra os resultados para incluir apenas os locais de compra onde a quantidade de CDs é superior a 2

▼ `HAVING`

- É usada em conjunto com o `GROUP BY` para filtrar os resultados de um consulta agragada com base em condições específicas.
- Enquanto o `WHERE` é usado para filtar linhas asntes do processo de agregação, o `HAVING` é usado para fazer essa filtragem após a agregação.

▼ 7) Mostrar os locais de compra, cuja média do valor pago por CD é inferior a 10, juntamente com o respetivo total do valor pago.

```
SELECT
    localCompra,
    AVG(valorPago) AS MEDIA_VALOR_PAGO_POR_CD,
    SUM(valorPago) AS TOTAL_VALOR_PAGO
FROM cd
GROUP BY localCompra
HAVING AVG(valorPago) < 10;
```

LOCALCOMPRA	MEDIA_VALOR_PAGO_POR_CD	TOTAL_VALOR_PAGO
iTunes	9.9	9.9

- `AVG` calcula a média do valor pago por CD para cada local de compra
- `SUM` calcula o total de valor pago por CD para cada local de compra
- `GROUP BY` agrupa os resultados por local de compra
- `HAVING` filtra os resultados, incluindo apenas os locais de compra cuja média do valor pago por CD é inferior a 10

▼ 8) Mostrar o valor total pago nos locais de compra, cuja quantidade de CD comprados é superior a 2. O local de compra também deve ser visualizado

```
SELECT
    localCompra,
    COUNT(*) AS QUANTIDADE_CD,
    SUM(valorPago) AS TOTAL_VALOR_PAGO
FROM cd
GROUP BY localCompra
HAVING COUNT(*) > 2;
```

LOCALCOMPRA	QUANTIDADE_CD	TOTAL_VALOR_PAGO
FNAC	3	93.7

- `COUNT` → conta a quantidade de CDs para cada local de compra
- `SUM` → calcula o valor total pago por CD para cada local de compra
- `GROUP` → agrupa os resultados por local de compra
- `HAVING COUNT` → filtra os resultados, incluindo apenas locais de compra onde a quantidade de CDs comprados é superior a 2.

▼ 9) Copiar o comando da alínea 5 e alterar de modo a mostrar apenas o intérprete e o código do CD em que o intérprete participa apenas em 1 música. O resultado deve ser apresentado por ordem crescente do código do CD e, em caso de igualdade, por ordem alfabética do intérprete.

```
SELECT cd.codCd, musica.interprete
FROM cd
JOIN musica ON cd.codCd = musica.codCd
GROUP BY cd.codCd, musica.interprete
HAVING COUNT(*) = 1
ORDER BY cd.codCd ASC, musica.interprete ASC;
```

CODCD	INTERPRETE
1	Lana Del Rey
1	Rufio
2	Loring Maazel
2	Zubin Mehta
4	Electric Light Orchestra
4	Eurythmics
4	Fine Young Cannibals
4	Kate Bush
4	Lana Del Rey
4	Wham!

- `HAVING COUNT(*) = 1` filtra os resultados para incluir apenas os intérpretes que participam em exatamente uma música.
- `ORDER BY cd.codCd ASC, musica.interprete ASC` ordena os resultados por código do CD em ordem crescente e, em caso de igualdade, por ordem alfabética do intérprete em ordem crescente.

▼ 10) Copiar o comando da alínea anterior e alterar de modo a mostrar apenas os intérpretes começados por E ou L (letras maiúsculas ou minúsculas).

```
SELECT cd.codCd, musica.interprete
FROM cd
JOIN musica ON cd.codCd = musica.codCd
WHERE UPPER(musica.interprete) LIKE 'E%' OR UPPER(musica.i
GROUP BY cd.codCd, musica.interprete
HAVING COUNT(*) = 1
ORDER BY cd.codCd ASC, musica.interprete ASC;
```

CODCD	INTERPRETE
1	Lana Del Rey
2	Loring Maazel
4	Electric Light Orchestra
4	Eurythmics
4	Lana Del Rey

- `UPPER` para tornar as comparações de intérpretes maiúsculas/minúsculas incensíveis.
- O `LIKE` filtra os intérpretes cujos nomes começam com E ou L

▼ 11) Mostrar os dias de semana em que foram comprados, em locais conhecidos, pelo menos dois CD

```
SELECT TO_CHAR(dataCompra, 'Day') as dia_semana, COUNT(*) 
FROM cd
WHERE localCompra IS NOT NULL
GROUP BY TO_CHAR(dataCompra, 'Day')
HAVING COUNT(*) >= 2;
```

DIA_SEMANA	QUANTIDADE_CD
Domingo	2
Sexta-Feira	3

- `TO_CHAR(dataCompra, 'Day')` é usado para obter o dia da semana a partir da data de compra.
- `WHERE localCompra IS NOT NULL` filtra os registros onde o local de compra é conhecido.
- `GROUP BY TO_CHAR(dataCompra, 'Day')` agrupa os resultados pelo dia da semana.

- `HAVING COUNT(*) >= 2` filtra os resultados, incluindo apenas os dias de semana em que foram comprados pelo menos dois CDs.

▼ 12) Mostrar, para cada CD, o título e a quantidade de músicas

```
SELECT
    cd.codCd,
    cd.titulo,
    COUNT(*) as quantidade_de_musicas
FROM cd
JOIN musica ON cd.codCd = musica.codCd
GROUP BY cd.codCd, cd.titulo;
```

CODCD	TITULO	QUANTIDADE_DE_MUSICAS
6	Giesta 2	10
1	Punkzilla	8
7	O Mundo ao Contrï¿½rio	10
2	Classic Duets	2
8	Born to Die	12
4	Hits 4	6
5	Songs of Experience	8
3	The Wall	5

- `cd.codCd` e `cd.titulo` são selecionados para mostrar o código e o título de cada CD.
- `JOIN musica ON cd.codCd = musica.codCd` unir as tabelas `cd` e `musica` com base no código do CD.
- `COUNT(*)` contar a quantidade de músicas para cada CD.
- `GROUP BY cd.codCd, cd.titulo` agrupa os resultados pelo código e título do CD.

▼ 13) Mostrar, para cada CD, o código, o título e a quantidade de músicas.

```
SELECT
    cd.codCd,
    cd.titulo,
    COUNT(*) as quantidade_de_musicas
FROM cd
LEFT JOIN musica ON cd.codCd = musica.codCd
GROUP BY cd.codCd, cd.titulo;
```

CODCD	TITULO	QUANTIDADE_DE_MUSICAS
6	Giesta 2	10
1	Punkzilla	8
7	O Mundo ao Contrï¿½rio	10
2	Classic Duets	2
8	Born to Die	12
4	Hits 4	6
5	Songs of Experience	8
3	The Wall	5

- `cd.codCd` e `cd.titulo` são selecionados para mostrar o código e o título de cada CD.
- `LEFT JOIN musica ON cd.codCd = musica.codCd` incluir CDs mesmo que não tenham músicas associadas.
- `COUNT(*)` contar a quantidade de músicas para cada CD.
- `GROUP BY cd.codCd, cd.titulo` agrupa os resultados pelo código e título do CD.

▼ 14) Mostrar, para cada CD que tenha músicas com duração superior a 5, o código, o título e a quantidade de músicas cuja duração é superior a 5.

```
SELECT
    cd.codCd,
    cd.titulo,
    COUNT(*) as quantidade_de_musicas_duracao_superior_a_5
```

```
FROM cd
JOIN musica ON cd.codCd = musica.codCd
WHERE musica.duracao > 5
GROUP BY cd.codCd, cd.titulo;
```

CODCD	TITULO	QUANTIDADE_DE_MUSICAS_DURACAO_SUPERIOR_A_5
7	O Mundo ao Contrï¿½rio	1
6	Giesta 2	1
3	The Wall	1

- `cd.codCd` e `cd.titulo` são selecionados para mostrar o código e o título de cada CD.
- `JOIN musica ON cd.codCd = musica.codCd` unir as tabelas `cd` e `musica` com base no código do CD.
- `WHERE musica.duracao > 5` filtra as músicas com duração superior a 5 minutos.
- `COUNT(*)` contar a quantidade de músicas com duração superior a 5 minutos para cada CD.
- `GROUP BY cd.codCd, cd.titulo` agrupa os resultados pelo código e título do CD.

▼ 15) Mostrar, para cada CD com menos de 6 músicas, o código, o título e a quantidade de músicas do CD.

```
SELECT
    cd.codCd,
    cd.titulo,
    COUNT(*) as quantidade_de_musicas
FROM cd
LEFT JOIN musica ON cd.codCd = musica.codCd
GROUP BY cd.codCd, cd.titulo
HAVING COUNT(*) < 6 OR COUNT(*) IS NULL;
```

CODCD	TITULO	QUANTIDADE_DE_MUSICAS
2	Classic Duets	2
3	The Wall	5

- `cd.codCd` e `cd.titulo` são selecionados para mostrar o código e o título de cada CD.
- `LEFT JOIN musica ON cd.codCd = musica.codCd` incluir CDs mesmo que não tenham músicas associadas.
- `COUNT(*)` contar a quantidade de músicas para cada CD.
- `GROUP BY cd.codCd, cd.titulo` agrupa os resultados pelo código e título do CD.
- `HAVING COUNT(*) < 6 OR COUNT(*) IS NULL` filtra os resultados, incluindo apenas CDs com menos de 6 músicas ou CDs sem músicas associadas.

▼ 16) Mostrar, para cada CD cujas músicas têm uma duração média superior a 4, o código, o título e a quantidade de músicas do CD.

```
SELECT cd.codCd, cd.titulo, COUNT(*) as quantidade_de_musi
FROM cd
JOIN musica ON cd.codCd = musica.codCd
GROUP BY cd.codCd, cd.titulo
HAVING AVG(musica.duracao) > 4;
```

Output: No Data Found

▼ 17) Mostrar, numa coluna, o título de cada uma das músicas e o título de cada CD que tem pelo menos 3 interpretes

```
SELECT
    cd.codCd,
    cd.titulo,
    COUNT(*) as quantidade_de_musicas
FROM cd
```

```
JOIN musica ON cd.codCd = musica.codCd
GROUP BY cd.codCd, cd.titulo
HAVING AVG(musica.duracao) > 4;
```

Output: No Data Found

- `cd.codCd` e `cd.titulo` são selecionados para mostrar o código e o título de cada CD.
- `JOIN musica ON cd.codCd = musica.codCd` unir as tabelas `cd` e `musica` com base no código do CD.
- `AVG(musica.duracao)` calcular a duração média das músicas para cada CD.
- `GROUP BY cd.codCd, cd.titulo` agrupa os resultados pelo código e título do CD.
- `HAVING AVG(musica.duracao) > 4` filtra os resultados, incluindo apenas CDs cujas músicas têm uma duração média superior a 4.

▼ 18) Copiar e alterar o comando da alínea anterior, de modo a não mostrar os registos repetidos

```
SELECT
    DISTINCT cd.codCd,
    cd.titulo,
    COUNT(*) as quantidade_de_musicas
FROM cd
JOIN musica ON cd.codCd = musica.codCd
GROUP BY cd.codCd, cd.titulo
HAVING AVG(musica.duracao) > 4;
```

Output: No Data Found

- O uso do `DISTINCT` garante que apenas linhas únicas são retornadas. Dessa forma, se houver CDs com a mesma duração média, apenas uma entrada para cada combinação única de `codCd` e `titulo` sera mostrada.

▼ 19) Copiar e alterar o comando da alínea anterior, de modo a apresentar também o comprimento de cada título e por ordem decrescente.

```
SELECT
    cd.codCd,
    cd.titulo,
    LENGTH(cd.titulo) as comprimento_do_titulo,
    COUNT(*) as quantidade_de_musicas
FROM cd
JOIN musica ON cd.codCd = musica.codCd
GROUP BY cd.codCd, cd.titulo
HAVING AVG(musica.duracao) > 4
ORDER BY comprimento_do_titulo DESC;
```

Output: No Data Found

- `LENGTH(cd.titulo) as comprimento_do_titulo` calcula o comprimento de cada título.
- `ORDER BY comprimento_do_titulo DESC` ordena os resultados por ordem decrescente com base no comprimento do título.

▼ 20) Mostrar os títulos de CD com duração superior a 35 e que são iguais a títulos de músicas

```
SELECT DISTINCT cd.titulo
FROM cd
JOIN musica ON cd.titulo = musica.titulo
WHERE cd.duracao > 35;
```

Output: No Data Found

- `JOIN musica ON cd.titulo = musica.titulo` unir as tabelas `cd` e `musica` com base no título.
- `WHERE cd.duracao > 35` filtra os CDs com duração superior a 35.
- `DISTINCT` garante que apenas títulos únicos sejam retornados.

PL

---

https://prod-files-secure.s3.us-west-2.amazonaws.com/3b829444-cd0b-4b9b-a3c6-750b0f6c18d6/e1cef50f-ff30-4daf-ba7b-a0e1a8474ff0/PL04B-SQL-DML-SELECT-Agrupamento-Ex_CD_de_Musica_2.pdf