

Projeto GDI

Mácio Meneses - mmmj@cin.ufpe.br

Morgana Galamba - mbfg2@cin.ufpe.br

Enrique Laborão - elm2@cin.ufpe.br

Júlio César Farias - jcfl@cin.ufpe.br

Consultas

Consultas - Group By/Having

- Mostra a quantidade de músicas e os títulos dos álbuns com mais de 5 músicas

```
SELECT  A.TITULO AS ALBUM ,COUNT(M.NOME) AS QUANTIDADE_DE_MUSICAS
FROM    MUSICA M, ALBUM A
WHERE   M.ID_ALBUM = A.ID_ALBUM
GROUP BY A.TITULO
HAVING  COUNT(M.NOME)>5;
```

Consultas - Junção Interna

- Exibir o nome de um artista e o nome das músicas que ele indicou

```
SELECT U.NOME_PRIMEIRO_NOME, B.NOME  
FROM USUARIO U  
INNER JOIN INDICA B  
ON U.ID=B.ID_ARTISTA;
```

Consultas - Junção Externa

- Retorna o primeiro nome dos artistas que ainda não postaram nenhum álbum

```
SELECT U.NOME_PRIMEIRO_NOME
FROM USUARIO U
WHERE U.ID IN (
    SELECT A.ID
    FROM ARTISTA A LEFT OUTER JOIN TEM T ON A.ID = T.ID
    WHERE T.ID IS NULL
);
```

Consultas - Semi Join

- Recuperando o primeiro nome dos artistas que participam de um álbum que tem uma música de muito sucesso (> 1 M de streamings)

```
SELECT U.NOME_PRIMEIRO_NOME
FROM USUARIO U
WHERE U.ID IN(
  SELECT T.ID
  FROM TEM T
  WHERE EXISTS (
    SELECT *
    FROM MUSICA M
    WHERE M.ID_ALBUM = T.ID_ALBUM AND M.NUMERO_DE_STREAMINGS > 1000000
  )
)
```

Consultas - Anti Join

- Recuperando o primeiro nome dos artistas que ainda não postaram nenhum álbum

```
SELECT U.NOME_PRIMEIRO_NOME
FROM USUARIO U
WHERE U.ID IN (
    SELECT A.ID
    FROM ARTISTA A
    WHERE NOT EXISTS(
        SELECT *
        FROM TEM T
        WHERE T.ID = A.ID
    )
)
```

Consultas - Subconsulta do tipo escalar

- Exibe os artistas que tem mais ouvintes mensais do que a média

```
SELECT U.NOME_PRIMEIRO_NOME AS Nome_do_Artista,U.NOME_SOBRENOME AS Sobrenome
FROM USUARIO U INNER JOIN ARTISTA A ON U.ID = A.ID
WHERE A.OUVINTES_MENSAIS > (
    SELECT AVG(OUVINTES_MENSAIS)
    FROM ARTISTA
);
```


Consultas - Subconsulta do tipo linha

- Retorna os outros usuários que têm o mesmo nome e sobrenome do usuário de ID '900000001'

```
SELECT U.NOME_PRIMEIRO_NOME, U.NOME_SOBRENOME , U.ID
FROM USUARIO U
WHERE U.ID<>'900000001' AND (U.NOME_PRIMEIRO_NOME,U.NOME_SOBRENOME) = (SELECT
    US.NOME_PRIMEIRO_NOME,US.NOME_SOBRENOME
    FROM USUARIO US
    WHERE US.ID='900000001'
);
```

Consultas - Subconsulta do tipo tabela

- Retorna o nome e sobrenome dos artistas que participaram do álbum de ID '12344666'

```
SELECT U.NOME_PRIMEIRO_NOME, U.NOME_SOBRENOME  
FROM USUARIO U  
WHERE U.ID IN (  
    SELECT TEM.ID  
    FROM TEM  
    WHERE TEM. ID_ALBUM='12344666'  
);
```

Consultas - Operação de Conjunto

- Retorna o ID do Ouvinte, Nome da Música e ID do Álbum das músicas que se tornaram favoritas do mês e haviam sido recomendadas à um ouvinte

```
SELECT R.ID_RECOMENDADO, R.NOME, R.ID_ALBUM  
FROM RECOMENDA R  
INTERSECT SELECT F.ID_OUVINTE, F.NOME, F.ID_ALBUM  
FROM FAVORITA_DO_MES F;
```

PL

PL - Trigger

- Não deixa um usuário seguir ele mesmo

```
CREATE OR REPLACE TRIGGER controle_segue
BEFORE INSERT ON SEQUE
FOR EACH ROW
BEGIN
    IF (:NEW.seguidor <> :NEW.seguido) THEN
        dbms_output.put_line('Seguindo');
    ELSE
        RAISE_APPLICATION_ERROR(-20101, 'UM USUARIO NÃO PODE SEGUIR ELE MESMO');
    END IF;
END;
```

PL - Função

- Contar quantos streamings um álbum tem, considerando todas as suas músicas

```
CREATE OR REPLACE FUNCTION F_STREAMINGS_ALBUM  
(ID_A ALBUM.ID_ALBUM%TYPE) RETURN INTEGER IS
```

```
    TOTAL INTEGER := 0;  
    NUMERO_ATUAL INTEGER;  
    CURSOR CUR_MUSICA IS  
        SELECT NUMERO_DE_STREAMINGS  
        FROM MUSICA  
        WHERE MUSICA.ID_ALBUM = ID_A;
```

```
BEGIN
```

```
    OPEN CUR_MUSICA;  
    FETCH CUR_MUSICA INTO NUMERO_ATUAL;  
    WHILE CUR_MUSICA%FOUND LOOP  
        TOTAL := TOTAL + NUMERO_ATUAL;  
        FETCH CUR_MUSICA INTO NUMERO_ATUAL;  
    END LOOP;
```

```
    CLOSE CUR_MUSICA;
```

```
    RETURN TOTAL;
```

```
END;
```

```
-- Chamando a função
```

```
BEGIN
```

```
    DBMS_OUTPUT.PUT_LINE('Numero de Streamings totais: ' || F_STREAMINGS_ALBUM('12345678'));
```

```
END;
```

PL - Procedure

- Printa todas as Músicas pertencentes a um Artista a partir do seu ID

```
CREATE OR REPLACE PROCEDURE musicas_artista (ID_ARTISTA VARCHAR2) IS
    CURSOR cur_pesq IS
        SELECT M.NOME,M.ID_ALBUM
        FROM MUSICA M
        WHERE M.ID_ALBUM IN(
            SELECT TEM.ID_ALBUM
            FROM TEM
            WHERE TEM.ID = ID_ARTISTA);
BEGIN
    FOR row_pesq IN cur_pesq LOOP
        DBMS_OUTPUT.PUT_LINE(row_pesq.NOME||' '|| row_pesq.ID_ALBUM);
    END LOOP;
END;
```

PL - Procedure

- Printa todas as Músicas pertencentes a um álbum

```
CREATE OR REPLACE PROCEDURE musicas_album (PK_ALBUM VARCHAR2) IS
    CURSOR cur_pesq IS
        SELECT M.NOME,M.ID_ALBUM
        FROM MUSICA M
        WHERE M.ID_ALBUM = PK_ALBUM;
BEGIN
    FOR row_pesq IN cur_pesq LOOP
        DBMS_OUTPUT.PUT_LINE(row_pesq.NOME||' '|| row_pesq.ID_ALBUM);
    END LOOP;
END;

BEGIN
    musicas_album('12345678');
END;
```


Obrigado!