

# Modelo Conceitual de Banco de Dados para Controle de Playlists de Músicas

Este documento descreve o modelo conceitual de um banco de dados para o controle de playlists de músicas, com o objetivo de organizar e gerenciar coleções de músicas de forma eficiente. O modelo visa atender às necessidades de usuários que desejam criar, editar e compartilhar suas playlists, além de fornecer informações detalhadas sobre as músicas e artistas.

## Entidades e Atributos:

- **Usuário:**
  - ID (identificador único)
  - Nome
  - Email
  - Data de Nascimento
  - Data de Cadastro
- **Playlist:**
  - ID (identificador único)
  - Nome
  - Descrição
  - Data de Criação
  - ID do Usuário (criador da playlist) - chave estrangeira referenciando Usuário
- **Música:**
  - ID (identificador único)
  - Título
  - Duração
  - Ano de Lançamento
  - ID do Álbum (chave estrangeira referenciando Álbum)
- **Álbum:**
  - ID (identificador único)
  - Nome
  - Ano de Lançamento
  - ID do Artista (chave estrangeira referenciando Artista)
- **Artista:**
  - ID (identificador único)
  - Nome
- **Gênero Musical:**
  - ID (identificador único)
  - Nome
- **PlaylistMúsica: (entidade associativa para representar a relação muitos-para-muitos entre Playlist e Música)**
  - ID da Playlist (chave estrangeira referenciando Playlist)
  - ID da Música (chave estrangeira referenciando Música)
  - Ordem (ordem da música na playlist)
- **MúsicaGênero: (entidade associativa para representar a relação muitos-para-muitos entre Música e Gênero Musical)**
  - ID da Música (chave estrangeira referenciando Música)

- ID do Gênero Musical (chave estrangeira referenciando Gênero Musical)

### **Relacionamentos:**

- Um usuário pode criar várias playlists (1:N).
- Uma playlist pode conter várias músicas (1:N).
- Uma música pode estar em várias playlists (N:M).
- Uma música pertence a um álbum (N:1).
- Um álbum pertence a um artista (N:1).
- Uma música pode pertencer a vários gêneros musicais (N:M).
- Um gênero musical pode estar associado a várias músicas (N:M).

# Modelo Lógico Relacional para Oracle

## Considerações:

- O modelo abaixo foi elaborado com base nas informações fornecidas sobre as entidades e seus atributos.
- Restrições de integridade referencial garantem a consistência dos dados entre as tabelas.
- Chave primária de cada tabela em **negrito**.
- Chave estrangeira de cada tabela em *itálico*.

---

## Usuário

Coluna	Tipo de Dado	Restrições
<b>ID_USUARIO</b>	NUMBER	PRIMARY KEY, NOT NULL
NOME	VARCHAR2(255)	NOT NULL
EMAIL	VARCHAR2(255)	UNIQUE, NOT NULL
DATA_NASCIMENTO	DATE	NULL
DATA_CADASTRO	DATE	DEFAULT SYSDATE, NOT NULL

---

# Playlist

Coluna	Tipo de Dado	Restrições
ID_PLAYLIST	NUMBER	PRIMARY KEY, NOT NULL
NOME	VARCHAR2(255)	NOT NULL
DESCRICAO	VARCHAR2(4000)	NULL
DATA_CRIACAO	DATE	DEFAULT SYSDATE, NOT NULL
ID_USUARIO	NUMBER	NOT NULL, FOREIGN KEY REFERENCES USUARIO(ID_USUARIO)

---

# Música

Coluna	Tipo de Dado	Restrições
ID_MUSICA	NUMBER	PRIMARY KEY, NOT NULL
TITULO	VARCHAR2(255)	NOT NULL
DURACAO	NUMBER	NOT NULL
ANO_LANCAMENTO	NUMBER	NULL
ID_ALBUM	NUMBER	NOT NULL, FOREIGN KEY REFERENCES ALBUM(ID_ALBUM)

---

## Álbum

Coluna	Tipo de Dado	Restrições
<b>ID_ALBUM</b>	NUMBER	PRIMARY KEY, NOT NULL
NOME	VARCHAR2(255)	NOT NULL
ANO_LANCAMENTO	NUMBER	NULL
ID_ARTISTA	NUMBER	NOT NULL, FOREIGN KEY REFERENCES ARTISTA(ID_ARTISTA)

---

## Artista

Coluna	Tipo de Dado	Restrições
<b>ID_ARTISTA</b>	NUMBER	PRIMARY KEY, NOT NULL
NOME	VARCHAR2(255)	NOT NULL

---

# Gênero Musical

Coluna	Tipo de Dado	Restrições
ID_GENERO	NUMBER	PRIMARY KEY, NOT NULL
NOME	VARCHAR2(255)	NOT NULL

---

## PlaylistMúsica

Coluna	Tipo de Dado	Restrições
ID_PLAYLIST	NUMBER	NOT NULL, PRIMARY KEY (ID_PLAYLIST, ID_MUSICA), FOREIGN KEY REFERENCES PLAYLIST(ID_PLAYLIST)
ID_MUSICA	NUMBER	NOT NULL, FOREIGN KEY REFERENCES MUSICA(ID_MUSICA)
ORDEM	NUMBER	NOT NULL



---

## MúsicaGênero

Coluna	Tipo de Dado	Restrições
ID_MUSICA	NUMBER	NOT NULL, PRIMARY KEY (ID_MUSICA, ID_GÊNERO), FOREIGN KEY REFERENCES MUSICA(ID_MUSICA)
ID_GENERO	NUMBER	NOT NULL, FOREIGN KEY REFERENCES GENERO_MUSICAL(ID_GENERO)

---

### Observações:

- A entidade associativa PlaylistMúsica possui a coluna ORDEM para controlar a ordem das músicas em uma playlist.
- A entidade associativa MúsicaGênero permite que uma música seja associada a múltiplos gêneros musicais.
- É possível adicionar outros atributos e restrições, como CHECK e UNIQUE, conforme a necessidade do sistema.

Este modelo lógico relacional fornece uma base sólida para a implementação do banco de dados em Oracle, garantindo a integridade e organização dos dados.

# Modelo Físico para Servidor Oracle - Script SQL

<https://livesql.oracle.com/ords/livesql/s/cj54jfdpt6wu6ru9gxevlydr9>

## -- 1. Criando tabelas

-- Tabela USUARIO

```
CREATE TABLE USUARIO (  
    ID_USUARIO NUMBER PRIMARY KEY,  
    NOME VARCHAR2(255) NOT NULL,  
    EMAIL VARCHAR2(255) UNIQUE NOT NULL,  
    DATA_NASCIMENTO DATE,  
    DATA_CADASTRO DATE DEFAULT SYSDATE NOT NULL  
);
```

-- Tabela PLAYLIST

```
CREATE TABLE PLAYLIST (  
    ID_PLAYLIST NUMBER PRIMARY KEY,  
    NOME VARCHAR2(255) NOT NULL,  
    DESCRICAO VARCHAR2(4000),  
    DATA_CRIACAO DATE DEFAULT SYSDATE NOT NULL,  
    ID_USUARIO NUMBER NOT NULL,  
    FOREIGN KEY (ID_USUARIO) REFERENCES USUARIO(ID_USUARIO)  
);
```

-- Tabela ARTISTA

```
CREATE TABLE ARTISTA (  
    ID_ARTISTA NUMBER PRIMARY KEY,  
    NOME VARCHAR2(255) NOT NULL
```

```
);
```

```
-- Tabela ALBUM
```

```
CREATE TABLE ALBUM (
```

```
    ID_ALBUM NUMBER PRIMARY KEY,
```

```
    NOME VARCHAR2(255) NOT NULL,
```

```
    ANO_LANCAMENTO NUMBER,
```

```
    ID_ARTISTA NUMBER NOT NULL,
```

```
    FOREIGN KEY (ID_ARTISTA) REFERENCES ARTISTA(ID_ARTISTA)
```

```
);
```

```
-- Tabela MUSICA
```

```
CREATE TABLE MUSICA (
```

```
    ID_MUSICA NUMBER PRIMARY KEY,
```

```
    TITULO VARCHAR2(255) NOT NULL,
```

```
    DURACAO NUMBER NOT NULL,
```

```
    ANO_LANCAMENTO NUMBER,
```

```
    ID_ALBUM NUMBER NOT NULL,
```

```
    FOREIGN KEY (ID_ALBUM) REFERENCES ALBUM(ID_ALBUM)
```

```
);
```

```
-- Tabela GENERO_MUSICAL
```

```
CREATE TABLE GENERO_MUSICAL (
```

```
    ID_GENERO NUMBER PRIMARY KEY,
```

```
    NOME VARCHAR2(255) NOT NULL
```

```
);
```

```
-- Tabela PlaylistMusica
```

```
CREATE TABLE PlaylistMusica (  
    ID_PLAYLIST NUMBER NOT NULL,  
    ID_MUSICA NUMBER NOT NULL,  
    ORDEM NUMBER NOT NULL,  
    PRIMARY KEY (ID_PLAYLIST, ID_MUSICA),  
    FOREIGN KEY (ID_PLAYLIST) REFERENCES PLAYLIST(ID_PLAYLIST),  
    FOREIGN KEY (ID_MUSICA) REFERENCES MUSICA(ID_MUSICA)  
);
```

```
-- Tabela MusicaGenero
```

```
CREATE TABLE MusicaGenero (  
    ID_MUSICA NUMBER NOT NULL,  
    ID_GENERO NUMBER NOT NULL,  
    PRIMARY KEY (ID_MUSICA, ID_GENERO),  
    FOREIGN KEY (ID_MUSICA) REFERENCES MUSICA(ID_MUSICA),  
    FOREIGN KEY (ID_GENERO) REFERENCES GENERO_MUSICAL(ID_GENERO)  
);
```

## **-- 2. Scripts de criação das sequences:**

```
-- Sequence para a tabela USUARIO
```

```
CREATE SEQUENCE SEQ_USUARIO
```

```
START WITH 1
```

```
INCREMENT BY 1;
```

```
-- Sequence para a tabela PLAYLIST
```

```
CREATE SEQUENCE SEQ_PLAYLIST
```

```
START WITH 1
```

```
INCREMENT BY 1;
```

```
-- Sequence para a tabela MUSICA
```

```
CREATE SEQUENCE SEQ_MUSICA
```

```
START WITH 1
```

```
INCREMENT BY 1;
```

```
-- Sequence para a tabela ALBUM
```

```
CREATE SEQUENCE SEQ_ALBUM
```

```
START WITH 1
```

```
INCREMENT BY 1;
```

```
-- Sequence para a tabela ARTISTA
```

```
CREATE SEQUENCE SEQ_ARTISTA
```

```
START WITH 1
```

```
INCREMENT BY 1;
```

```
-- Sequence para a tabela GENERO_MUSICAL
```

```
CREATE SEQUENCE SEQ_GENERO
```

```
START WITH 1
```

```
INCREMENT BY 1;
```

### -- 3. Inserção de dados (exemplos):

-- Inserindo dados na tabela ARTISTA

```
INSERT INTO ARTISTA (ID_ARTISTA, NOME) VALUES (SEQ_ARTISTA.NEXTVAL,
'AnaVitória');
```

```
INSERT INTO ARTISTA (ID_ARTISTA, NOME) VALUES (SEQ_ARTISTA.NEXTVAL,
'Lagum');
```

```
INSERT INTO ARTISTA (ID_ARTISTA, NOME) VALUES (SEQ_ARTISTA.NEXTVAL,
'Imagine Dragons');
```

```
INSERT INTO ARTISTA (ID_ARTISTA, NOME) VALUES (SEQ_ARTISTA.NEXTVAL,
'Coldplay');
```

-- Inserindo dados na tabela ALBUM

```
INSERT INTO ALBUM (ID_ALBUM, NOME, ANO_LANCAMENTO, ID_ARTISTA) VALUES
(SEQ_ALBUM.NEXTVAL, 'Cor', 2022, 1);
```

-- AnaVitória

```
INSERT INTO ALBUM (ID_ALBUM, NOME, ANO_LANCAMENTO, ID_ARTISTA)
VALUES (SEQ_ALBUM.NEXTVAL, 'Memórias (De onde eu nunca fui)', 2023, 2);
```

-- Lagum

```
INSERT INTO ALBUM (ID_ALBUM, NOME, ANO_LANCAMENTO, ID_ARTISTA)
VALUES (SEQ_ALBUM.NEXTVAL, 'Mercury - Act 1', 2021, 3);
```

-- Imagine Dragons

```
INSERT INTO ALBUM (ID_ALBUM, NOME, ANO_LANCAMENTO, ID_ARTISTA)
VALUES (SEQ_ALBUM.NEXTVAL, 'Music of the Spheres', 2021, 4);
```

-- Coldplay

-- Inserindo dados na tabela MUSICA

```
INSERT INTO MUSICA (ID_MUSICA, TITULO, DURACAO, ANO_LANCAMENTO, ID_ALBUM)
```

```

VALUES (SEQ_MUSICA.NEXTVAL, 'Amarelo, Azul e Branco', 234, 2022, 1);

-- AnaVitória

INSERT INTO MUSICA (ID_MUSICA, TITULO, DURACAO, ANO_LANCAMENTO, ID_ALBUM)
VALUES (SEQ_MUSICA.NEXTVAL, 'universo de coisas que eu desconheço', 187,
2022, 1);

-- AnaVitória

INSERT INTO MUSICA (ID_MUSICA, TITULO, DURACAO, ANO_LANCAMENTO, ID_ALBUM)
VALUES (SEQ_MUSICA.NEXTVAL, 'Deixa', 256, 2023, 2);

-- Lagum

INSERT INTO MUSICA (ID_MUSICA, TITULO, DURACAO, ANO_LANCAMENTO, ID_ALBUM)
VALUES (SEQ_MUSICA.NEXTVAL, 'Enemy', 173, 2021, 3);

-- Imagine Dragons

INSERT INTO MUSICA (ID_MUSICA, TITULO, DURACAO, ANO_LANCAMENTO, ID_ALBUM)
VALUES (SEQ_MUSICA.NEXTVAL, 'My Universe', 228, 2021, 4);

-- Coldplay

-- Inserindo dados na tabela GENERO_MUSICAL

INSERT INTO GENERO_MUSICAL (ID_GENERO, NOME) VALUES (SEQ_GENERO.NEXTVAL,
'MPB');

INSERT INTO GENERO_MUSICAL (ID_GENERO, NOME) VALUES (SEQ_GENERO.NEXTVAL,
'Pop');

INSERT INTO GENERO_MUSICAL (ID_GENERO, NOME) VALUES (SEQ_GENERO.NEXTVAL,
'Rock Alternativo');

INSERT INTO GENERO_MUSICAL (ID_GENERO, NOME) VALUES (SEQ_GENERO.NEXTVAL,
'Indie');

-- Inserindo dados na tabela MusicaGenero

INSERT INTO MusicaGenero (ID_MUSICA, ID_GENERO) VALUES (1, 1);

-- Amarelo, Azul e Branco - MPB

INSERT INTO MusicaGenero (ID_MUSICA, ID_GENERO) VALUES (1, 2);

```

```

-- Amarelo, Azul e Branco - Pop

INSERT INTO MusicaGenero (ID_MUSICA, ID_GENERO) VALUES (2, 1);

-- universo de coisas que eu desconheço - MPB

INSERT INTO MusicaGenero (ID_MUSICA, ID_GENERO) VALUES (3, 3);

-- Deixa - Rock Alternativo

INSERT INTO MusicaGenero (ID_MUSICA, ID_GENERO) VALUES (4, 3);

-- Enemy - Rock Alternativo

INSERT INTO MusicaGenero (ID_MUSICA, ID_GENERO) VALUES (5, 2);

-- My Universe - Pop


-- Inserindo dados na tabela USUARIO

INSERT INTO USUARIO (ID_USUARIO, NOME, EMAIL, DATA_NASCIMENTO)

VALUES (SEQ_USUARIO.NEXTVAL, 'João Silva', 'joao.silva@email.com',
TO_DATE('1990-05-10', 'YYYY-MM-DD'));

INSERT INTO USUARIO (ID_USUARIO, NOME, EMAIL, DATA_NASCIMENTO)

VALUES (SEQ_USUARIO.NEXTVAL, 'Maria Oliveira', 'maria.oliveira@email.com',
TO_DATE('1985-12-22', 'YYYY-MM-DD'));


-- Inserindo dados na tabela PLAYLIST

INSERT INTO PLAYLIST (ID_PLAYLIST, NOME, DESCRICAO, ID_USUARIO)

VALUES (SEQ_PLAYLIST.NEXTVAL, 'MPB Favoritas', 'Minhas músicas favoritas
de MPB', 1);

INSERT INTO PLAYLIST (ID_PLAYLIST, NOME, DESCRICAO, ID_USUARIO)

VALUES (SEQ_PLAYLIST.NEXTVAL, 'Rock Internacional', 'Músicas de rock
internacional que eu gosto', 2);

INSERT INTO PlaylistMusica (ID_PLAYLIST, ID_MUSICA, ORDEM) VALUES (1, 1,
1);

-- MPB Favoritas - Amarelo, Azul e Branco

INSERT INTO PlaylistMusica (ID_PLAYLIST, ID_MUSICA, ORDEM) VALUES (1, 2,
2);

```



```
-- MPB Favoritas - universo de coisas que eu desconheço

INSERT INTO PlaylistMusica (ID_PLAYLIST, ID_MUSICA, ORDEM) VALUES (2, 4,
1);

-- Rock Internacional - Enemy

INSERT INTO PlaylistMusica (ID_PLAYLIST, ID_MUSICA, ORDEM) VALUES (2, 5,
2);

-- Rock Internacional - My Universe
```

## Atividade de revisão de conteúdo - SQL Oracle

### Enunciado:

Com base no esquema de banco de dados fornecido, escreva as declarações SQL para responder às seguintes perguntas:

### Questões:

1. **(Subconsulta - Única linha)** Liste o nome de todos os artistas que lançaram álbuns no mesmo ano que o álbum "Cor" da AnaVitória.
2. **(Subconsulta - Múltiplas linhas)** Liste o nome de todas as músicas que estão na playlist "MPB Favoritas".
3. **(Subconsulta - Correlacionada)** Liste o nome de todos os artistas que possuem mais de um álbum cadastrado.
4. **(INNER JOIN - USING)** Liste o título da música e o nome do gênero musical de todas as músicas que possuem gênero cadastrado.
5. **(INNER JOIN - ON)** Liste o nome do artista e o nome do álbum para todos os álbuns lançados.
6. **(LEFT JOIN)** Liste o nome de todos os usuários e o nome das playlists que eles criaram, incluindo usuários que não criaram playlists.
7. **(RIGHT JOIN)** Liste o nome de todas as playlists e o nome do usuário que a criou, incluindo playlists que não foram criadas por nenhum usuário (se houver).
8. **(FULL JOIN)** Liste o nome de todos os usuários e o nome das playlists que eles criaram, incluindo usuários que não criaram playlists e playlists que não foram criadas por nenhum usuário (se houver).
9. **(AUTO JUNÇÃO)** Liste os pares de músicas que possuem o mesmo ano de lançamento.
10. **(GROUP BY - ROLLUP)** Liste a quantidade de músicas por gênero musical, incluindo um subtotal para cada gênero e um total geral.
11. **(GROUP BY - CUBE)** Liste a quantidade de músicas por gênero musical e por ano de lançamento, incluindo subtotais e totais gerais.
12. **(GROUP BY - GROUPING)** Liste a quantidade de músicas por gênero musical e por ano de lançamento, identificando as linhas de subtotal e total geral.
13. **(GROUP BY - GROUPING SET)** Liste a quantidade de músicas por gênero musical, a quantidade de músicas por ano de lançamento e o total geral de músicas.
14. **(INTERSECT)** Liste as músicas que estão presentes em todas as playlists.
15. **(MINUS)** Liste as músicas que estão na playlist "MPB Favoritas" mas não estão na playlist "Rock Internacional".
16. **(UNION)** Liste todos os nomes de artistas e usuários, sem repetições.
17. **(UNION ALL)** Liste todos os nomes de artistas e usuários, incluindo repetições.
18. **(TO\_CHAR(null))** Liste o nome e a data de nascimento de todos os usuários, formatando a data de nascimento como 'DD/MM/YYYY' e exibindo 'Sem data' caso a data de nascimento seja nula.
19. **(Igualando a lista SELECT)** Liste o nome das músicas que têm a mesma duração que a música "Amarelo, Azul e Branco".
20. **(Subconsulta - WHERE – GROUP BY – HAVING)** Liste o nome dos artistas que possuem mais álbuns lançados do que o artista "Imagine Dragons".

## Exercícios Complementares - Oracle SQL

### Views:

1. **Criar uma view:** Crie uma view chamada vw\_musicas\_populares que exiba o título da música, o nome do artista e o nome do álbum para todas as músicas que pertencem ao gênero "Pop".
2. **Inserir dados através de uma view:** Insira uma nova música na tabela MUSICA através da view vw\_musicas\_populares. Utilize a cláusula WITH CHECK OPTION na definição da view para garantir que apenas músicas do gênero "Pop" possam ser inseridas.
3. **Atualizar dados através de uma view:** Atualize o ano de lançamento de uma música na tabela MUSICA através da view vw\_musicas\_populares.
4. **Deletar dados através de uma view:** Delete uma música da tabela MUSICA através da view vw\_musicas\_populares.
5. **Criar uma view READ ONLY:** Crie uma view chamada vw\_artistas\_albums que exiba o nome do artista e o nome do álbum para todos os artistas, mas que não permita operações DML (INSERT, UPDATE, DELETE).

### Sequences:

1. **Criar uma sequence:** Crie uma sequence chamada SEQ\_PLAYLISTMUSICA que inicie com 1 e incremente de 1 em 1.
2. **Utilizar NEXTVAL:** Insira um novo registro na tabela PlaylistMusica utilizando SEQ\_PLAYLISTMUSICA.NEXTVAL para gerar o valor da coluna ORDEM.
3. **Utilizar CURRVAL:** Após inserir um registro na tabela PlaylistMusica, exiba o valor atual da sequence SEQ\_PLAYLISTMUSICA utilizando CURRVAL.

### Índices:

1. **Criar um índice:** Crie um índice na coluna EMAIL da tabela USUARIO para otimizar as consultas que utilizam essa coluna na cláusula WHERE.
2. **Explicar o impacto do índice:** Explique como o índice criado na questão anterior afeta o desempenho da consulta SELECT \* FROM USUARIO WHERE EMAIL = 'joao.silva@email.com'.
3. **Identificar índices:** Utilize o dicionário de dados do Oracle para identificar os índices existentes no seu esquema de banco de dados.

### Controle de acesso e privilégios:

1. **Criar um usuário:** Crie um novo usuário chamado usuario\_playlist com uma senha.
2. **Conceder privilégios:** Conceda ao usuário usuario\_playlist o privilégio de selecionar dados da tabela MUSICA.
3. **Revogar privilégios:** Revogue o privilégio de selecionar dados da tabela MUSICA do usuário usuario\_playlist.
4. **Criar um role:** Crie um role chamado role\_playlist que tenha o privilégio de inserir dados na tabela PLAYLIST.
5. **Atribuir role a um usuário:** Atribua o role role\_playlist ao usuário usuario\_playlist.

## Controle de transações:

1. **Executar uma transação:** Insira uma nova música na tabela MUSICA e um novo álbum na tabela ALBUM dentro de uma transação. Utilize COMMIT para confirmar as alterações ou ROLLBACK para desfazer as alterações.
2. **Utilizar SAVEPOINT:** Dentro de uma transação, insira uma nova música na tabela MUSICA, crie um SAVEPOINT chamado sp1, insira um novo álbum na tabela ALBUM e utilize ROLLBACK TO sp1 para desfazer a inserção do álbum, mantendo a inserção da música.
3. **Explicar as propriedades ACID:** Explique cada uma das propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade) no contexto de um banco de dados Oracle e como elas garantem a integridade dos dados.