

**JANEIRO, 2019**

# Banco de Dados



# RESUMO

Especializados no armazenamento de dados;  
Otimizados para elevar a performance;  
Funcionam separados da aplicação.





# CARACTERÍSTICAS

SEGUE UMA ESTRUTURA RÍGIDA

ARMAZENAM DADOS  
ESTRUTURADOS

ESTÃO EM TABELAS

# MODELO ENTIDADE E RELACIONAMENTO

## ENTIDADES

São os participantes;

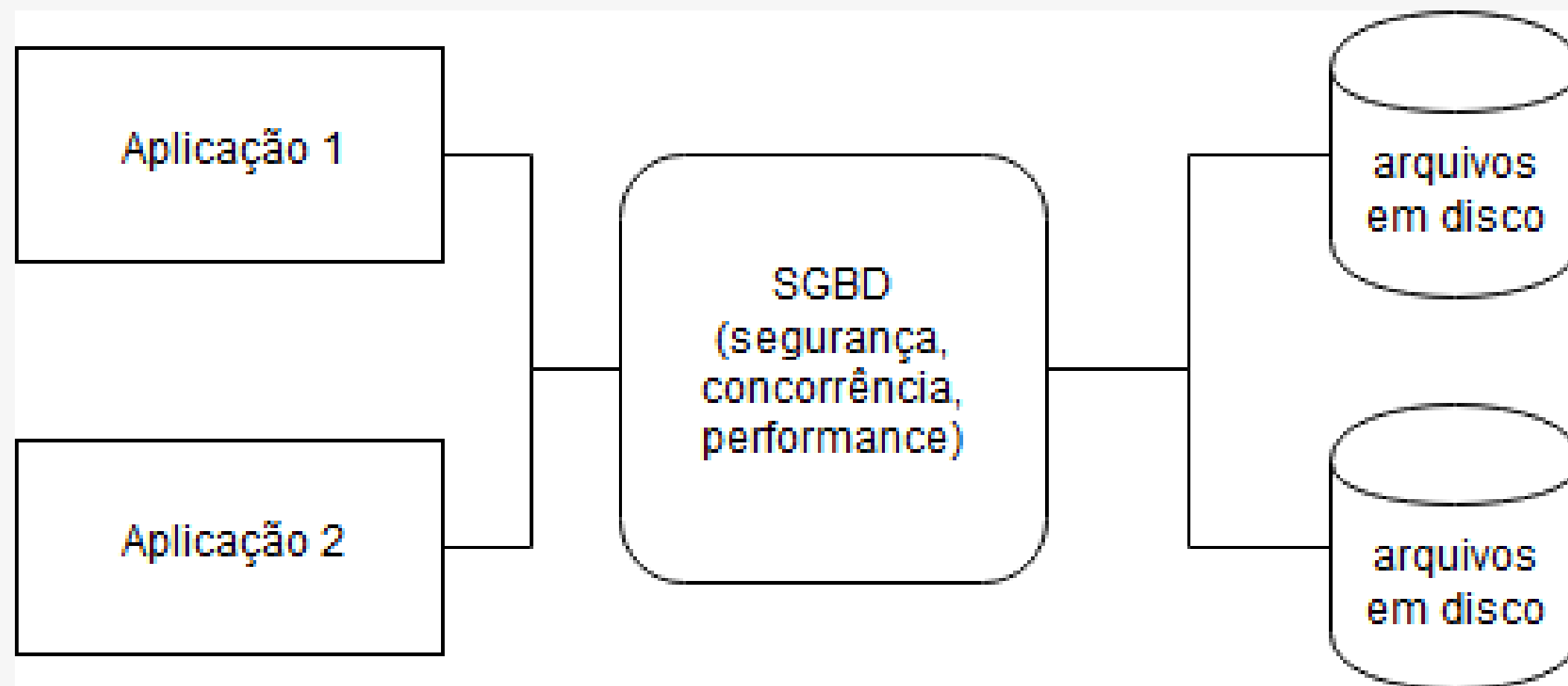
## RELACIONAMENTOS

As relações entre as entidades;



# SGBD

Sistema de gerenciamento de base de dados;



# CARDINALIDADE

**1:1**  
**1:N**  
**N:N**

# 1:1

1 motorista possui uma CNH.

id	nome	cnh	dt_expedicao	dt_validade
1	Helena	123	18/01/2017	18/01/2020
2	Fernando	456	18/01/2017	18/01/2019

motoristas		
id	nome	id_cnh
1	Helena	1
2	Fernando	2

cnhs			
id	cnh	dt_expedicao	dt_validade
1	123	18/01/2017	18/01/2020
2	456	18/01/2017	18/01/2019

# 1:N

1 pessoa possui N emails.

pe<sup>so</sup>as

id	nome	data_nascimento
1	Helena	18/03/1993
2	Fernando	10/11/1985

emails

id	endereco	id_pessoa
1	<a href="mailto:h.strada@hotmail.com">h.strada@hotmail.com</a>	1
2	<a href="mailto:fernando.henrique@corujasdev.com.br">fernando.henrique@corujasdev.com.br</a>	2



# N:N

Alunos assistem N aulas.

alunos	
id	nome
1	Helena
2	Fernando

alunos_aulas	
id_aluno	id_aula
1	1
1	2

aulas	
id	aula
1	matemática
2	português

# CHAVE PRIMÁRIA - PK - PRIMARY KEY

Conjunto de um ou mais campos que identifica unicamente um registro da tabela;

# CHAVE ESTRANGEIRA - FK - FOREIGN KEY

Estabelece o relacionamento entre duas tabelas, garantindo não realizar uma relação inválida;

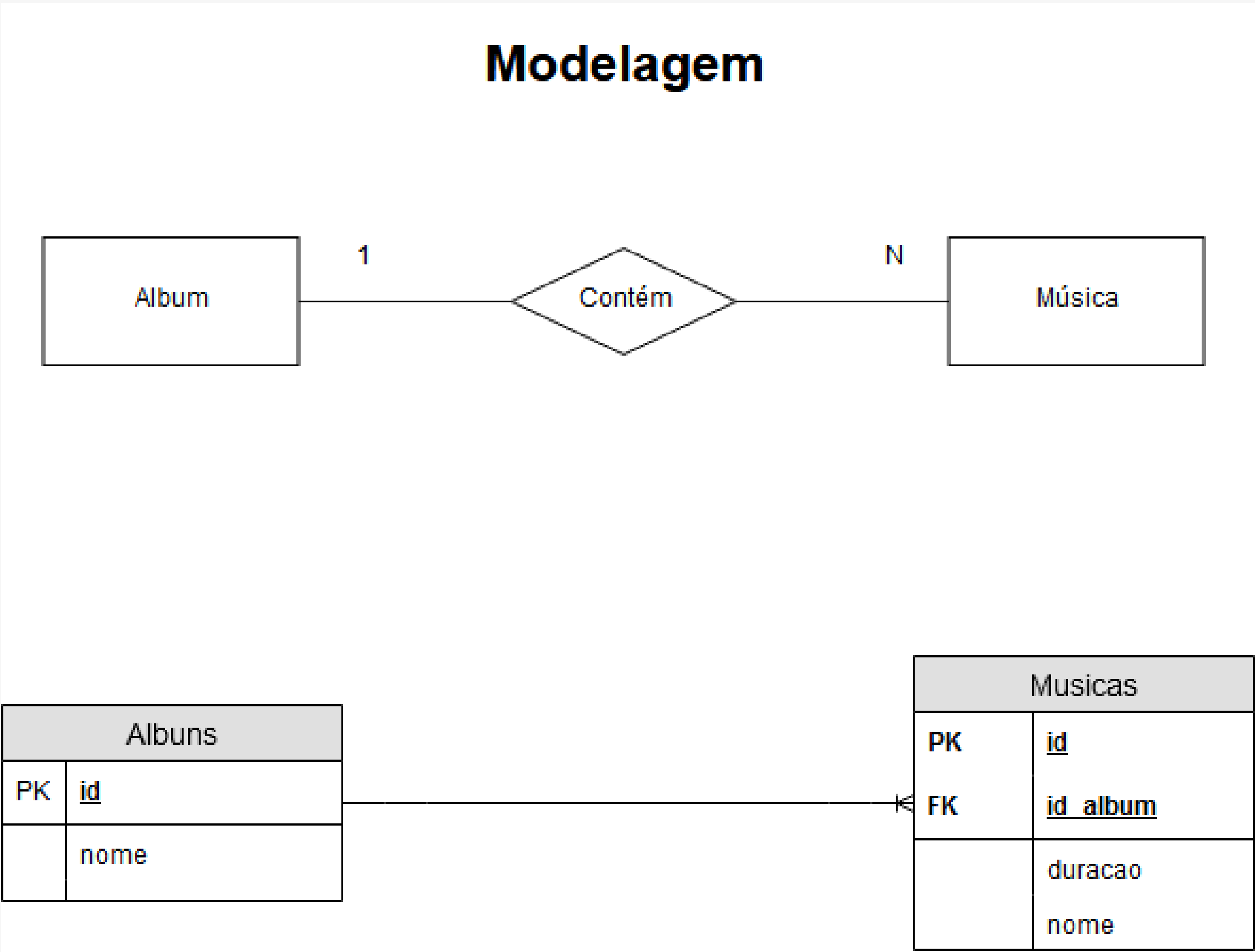
# TIPOS DE DADOS

<https://docs.microsoft.com/pt-br/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-2017>

# MODELAGEM + EXCEL

albuns	
id	nome
1	Album1
2	Album2
3	Album3

musicas			
id	nome	duracao	id_album
1	Musica1	2	1
2	Musica2	3	1





# DDL

---

LINGUAGEM DE DEFINIÇÃO DE DADOS  
INTERAGEM COM OS OBJETOS DO BANCO



```

CREATE DATABASE BUGSMUSIC;

USE BUGSMUSIC;

-- DDL - LINGUAGEM DE DEFINIÇÃO DE DADOS

CREATE TABLE NOME_TABELA (
  -- COLUMNA    TIPO_DADOS    CARACTERÍSTICAS    CHAVE
  ID            INT            IDENTITY            PRIMARY KEY
  ,NOME         VARCHAR(200)    UNIQUE NOT NULL
);

CREATE TABLE ALBUNS (
  -- COLUMNA    TIPO_DADOS    CARACTERÍSTICAS    CHAVE
  ID            INT            IDENTITY            PRIMARY KEY
  ,NOME         VARCHAR(200)    NOT NULL
);

CREATE TABLE MUSICAS (
  -- COLUMNA    TIPO_DADOS    CARACTERÍSTICAS    CHAVE
  ID            INT            IDENTITY            PRIMARY KEY
  ,NOME         VARCHAR(200)    NOT NULL
  ,ID_ALBUM     INT            NOT NULL            FOREIGN KEY REFERENCES ALBUNS(ID)
);

-- ADICIONANDO UMA NOVA COLUMNA NA TABELA DE MUSICAS

ALTER TABLE MUSICAS
ADD DURACAO FLOAT

```

# DML



---

LINGUAGEM DE MANIPULAÇÃO DE DADOS  
INTERAGEM COM OS DADOS DENTRO DA TABELA



```
-- DML - LINGUAGEM DE MANIPULAÇÃO DE DADOS

-- INSERT INTO TABELA (COLUNA1, COLUNA2, COLUNA3) VALUES (VALOR1, VALOR2, VALOR3)
INSERT INTO ALBUNS (NOME) VALUES ('ALBUM1');
INSERT INTO ALBUNS (NOME) VALUES ('ALBUM2'), ('ALBUM3');

-- UPDATE TABELA SET VALOR = 'NOVOVALOR' WHERE CONDICAÇÃO = SUACONDICAÇÃO
UPDATE ALBUNS SET NOME = 'ALBUM1 - RENOMEADO' WHERE ID = 1;

-- DELETE FROM TABELA WHERE CONDICAÇÃO = SUACONDICAÇÃO
DELETE FROM ALBUNS WHERE ID = 3;
```



# DQL

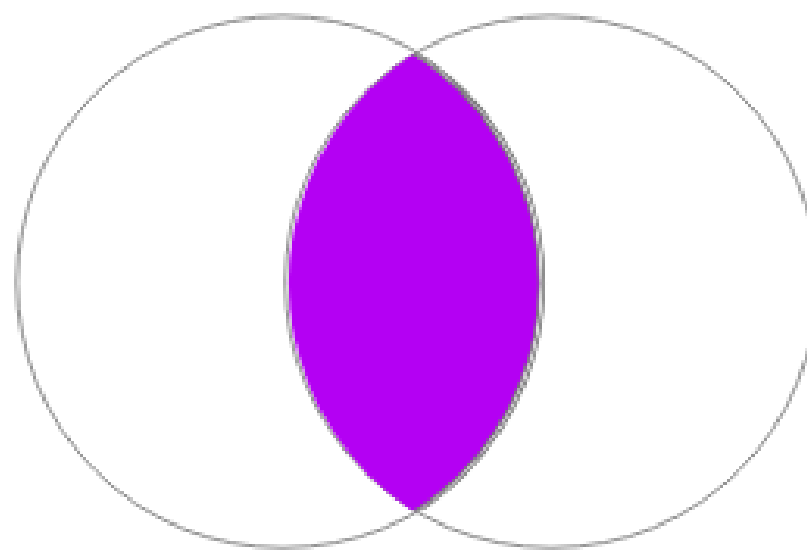
LINGUAGEM DE CONSULTA DE DADOS  
COMANDOS DE CONSULTA  
ALGUNS AUTORES COLOCAM COMO DML

```
-- DQL - LINGUAGEM DE CONSULTA DE DADOS

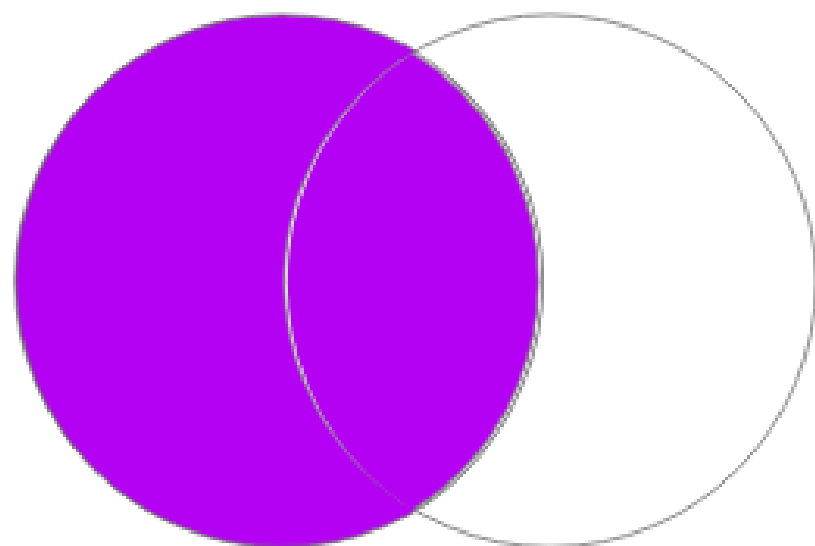
-- SELECT COLUNAS FROM TABELA
SELECT * FROM ALBUNS;
SELECT NOME FROM ALBUNS;
SELECT * FROM ALBUNS WHERE ID = 1;
SELECT * FROM ALBUNS WHERE ID > 1;
-- OR, AND, >, <, >=, <=, WHERE, COUNT, GROUP BY, MAX, MIN, AVG
```

# JUNÇÕES

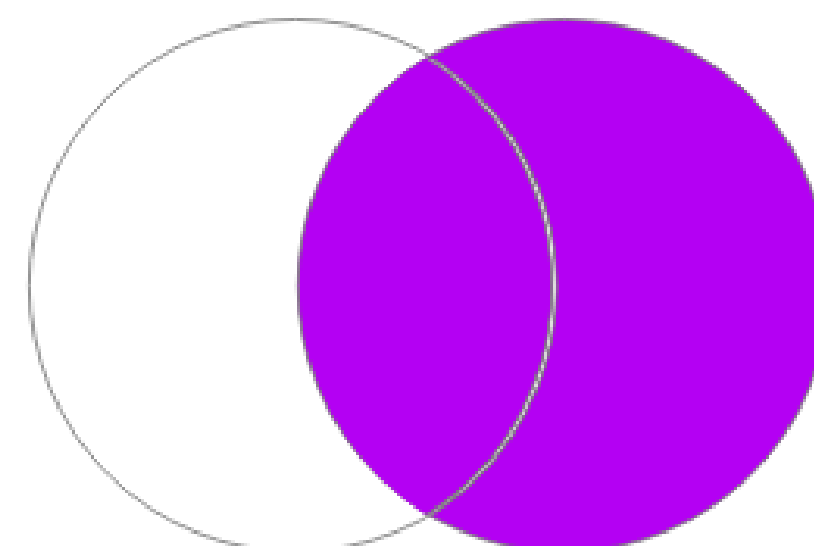
INNER JOIN



LEFT JOIN



RIGHT JOIN



JUNÇÕES

# JUNÇÕES - INNER JOIN

```
-- JUNCOES
-- SELECT * FROM PRIMEIRA_TABELA TIPO_JUNCAO SEGUNDA_TABELA ON PRIMEIRA_TABELA.IDENTIFICADOR = SEGUNDA_TABELA.IDENTIFICADOR
SELECT * FROM ALBUNS INNER JOIN MUSICAS ON ALBUNS.ID = MUSICAS.ID_ALBUM;
SELECT * FROM ALBUNS LEFT JOIN MUSICAS ON ALBUNS.ID = MUSICAS.ID_ALBUM;
SELECT * FROM ALBUNS RIGHT JOIN MUSICAS ON ALBUNS.ID = MUSICAS.ID_ALBUM;
```

91 %

 Resultados  Mensagens

	ID	NOME	ID	NOME	ID_ALBUM	DURACAO
1	1	ALBUM1 - RENOMEADO	1	MUSICA1	1	NULL
2	1	ALBUM1 - RENOMEADO	2	MUSICA2	1	NULL

# JUNÇÕES - LEFT JOIN

```
-- JUNCOES
-- SELECT * FROM PRIMEIRA_TABELA TIPO_JUNCAO SEGUNDA_TABELA ON PRIMEIRA_TABELA.IDENTIFICADOR = SEGUNDA_TABELA.IDENTIFICADOR
SELECT * FROM ALBUNS INNER JOIN MUSICAS ON ALBUNS.ID = MUSICAS.ID_ALBUM;
SELECT * FROM ALBUNS LEFT JOIN MUSICAS ON ALBUNS.ID = MUSICAS.ID_ALBUM;
SELECT * FROM ALBUNS RIGHT JOIN MUSICAS ON ALBUNS.ID = MUSICAS.ID_ALBUM;
```

91 %

Resultados Mensagens

	ID	NOME	ID	NOME	ID_ALBUM	DURACAO
1	1	ALBUM1 - RENOMEADO	1	MUSICA1	1	NULL
2	1	ALBUM1 - RENOMEADO	2	MUSICA2	1	NULL
3	2	ALBUM2	NULL	NULL	NULL	NULL

# JUNÇÕES - RIGHT JOIN

```
-- JUNCOES
-- SELECT * FROM PRIMEIRA_TABELA TIPO_JUNCAO SEGUNDA_TABELA ON PRIMEIRA_TABELA.IDENTIFICADOR = SEGUNDA_TABELA.IDENTIFICADOR
SELECT * FROM ALBUNS INNER JOIN MUSICAS ON ALBUNS.ID = MUSICAS.ID_ALBUM;
SELECT * FROM ALBUNS LEFT JOIN MUSICAS ON ALBUNS.ID = MUSICAS.ID_ALBUM;
SELECT * FROM ALBUNS RIGHT JOIN MUSICAS ON ALBUNS.ID = MUSICAS.ID_ALBUM;
```

91 %

Resultados Mensagens

	ID	NOME	ID	NOME	ID_ALBUM	DURACAO
1	1	ALBUM1 - RENOMEADO	1	MUSICA1	1	NULL
2	1	ALBUM1 - RENOMEADO	2	MUSICA2	1	NULL