

## Slide 1: Título

### **Tipos de Dados e Constraints no MySQL**

- Diferenças entre tipos numéricos, texto, data e binários
  - Uso avançado de ENUM, SET e JSON
  - Constraints: UNIQUE, DEFAULT, CHECK, NOT NULL, AUTO\_INCREMENT
- 

## **Slide 2: Visão Geral**

- Por que escolher tipos adequados?
  - Impactos em armazenamento, validação e performance
- 

## **Slide 3: Tipos Numéricos**

- Exemplos:  
INT, SMALLINT, TINYINT, DECIMAL(12,2), FLOAT, DOUBLE
  - Quando usar inteiro vs decimal vs float
  - Exemplos de declaração
  - Quando for valores em reais com precisão de duas casas decimais, use o decimal e para mais de duas casas decimais, use o float (6 casas decimais). Quando ter mais de 6 casas decimais, então use o double.
- 

## **Slide 4: Tipos Texto**

- CHAR(n) vs VARCHAR(n)
    - CHAR para string com menos de 10 caracteres e acima, use o varchar até 256 caracteres.
  - TEXT, MEDIUMTEXT, BLOB, VARBINARY
  - Explicação clara das capacidades de armazenamento de cada tipo [Devart Blog+7Devart Blog+7Stack Overflow+7GeeksforGeeks+4hasura.io+4DEV Community+4dev.mysql.com+2w3schools.com+2w3schools.com+2](#)
- 

## **Slide 5: Tipos Data/Hora e Binários**

- DATE, DATETIME, TIMESTAMP e suas faixas válidas [dev.mysql.com](#)
- BINARY(n), VARBINARY(n), BLOB – armazenamento de dados binários

---

## Slide 6: ENUM e SET – Conceitos

- **ENUM:** uma escolha única dentre valores pré-definidos (ex: status 'pending', 'processing', 'completed') [DEV Community+4dev.mysql.com+4w3schools.com+4Stack Overflow+2Devart Blog+2dev.mysql.com+2](#)
- **SET:** permite múltiplas escolhas combinadas (ex: dias disponíveis 'monday,tuesday') [Stack Overflow+4Prisma+4Stack Overflow+4](#)

---

## Slide 7: ENUM – Exemplo Prático

```
CREATE TABLE shirts (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(40),  
  size ENUM('x-small','small','medium','large','x-large')  
);  
  
INSERT INTO shirts (name,size) VALUES ('polo','medium');  
  
SELECT name, size FROM shirts WHERE size='medium';
```

Discussão: impactos de armazenamento e ordenação [dev.mysql.com](#)

---

## Slide 8: SET – Exemplo Prático

```
CREATE TABLE volunteers (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(50),  
  availability  
SET('monday','tuesday','wednesday','thursday','friday','saturday','sunday')  
);  
  
INSERT INTO volunteers (name,availability) VALUES ('Ana','monday,wed,friday');
```

Como o MySQL normaliza e ordena internamente  
[w3schools.com+12Prisma+12dev.mysql.com+12](#)

---

### Slide 9: Tipo JSON – Uso Avançado

```
CREATE TABLE products (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  data JSON  
);  
  
INSERT INTO products (data)  
VALUES ('{"color":"red","sizes":["M","L"],"price":49.90}');  
  
SELECT JSON_EXTRACT(data,'$.sizes[0]') FROM products;
```

Armazenamento binário eficiente e funções JSON disponíveis  
[mysqltutorial.orgPrisma](https://mysqltutorial.org/Prisma)

---

### Slide 10: Constraints Básicas (NOT NULL, DEFAULT)

- **NOT NULL:** impede valores nulos
  - **DEFAULT:** valor padrão usado quando não é informado  
[dev.mysql.com+4DEV Community+4hasura.io+4](https://dev.mysql.com+4DEV+Community+4hasura.io+4)
  - Sintaxe e exemplos práticos
- 

### Slide 11: Constraint UNIQUE

- Impede duplicatas em colunas ou conjunto de colunas
- Pode aceitar **NULL** em colunas únicas (ao contrário da PK) [DEV Community+4hasura.io+4dev.mysql.com+4dev.mysql.com+6GeeksforGee+6dev.mysql.com+6](https://dev.mysql.com+4dev.mysql.com+6GeeksforGee+6dev.mysql.com+6)
- Exemplo:

```
CREATE TABLE products (  
  product_code VARCHAR(20) UNIQUE,  
  name VARCHAR(100)  
);
```

---

### Slide 12: Constraint CHECK

- Permite validace de valores com expresses booleanas
- Exemplo:

sql

CopiarEditar

```
CREATE TABLE employees (
    salary DECIMAL(10,2),
    bonus DECIMAL(10,2),
    CHECK (salary >= bonus * 2)
);
```

```
```:contentReference[oaicite:34]{index=34}
```

---

### ## Slide 13: AUTO\_INCREMENT

- Restrito a tipos numricos e usualmente usado como PK
- Comea em 1 por padro, pode ser alterado ( ` ALTER TABLE ...  
AUTO\_INCREMENT = 100 ` ) :contentReference[oaicite:35]{index=35}
- Exemplo de uso:

```
```sql
```

```
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY
```

---

### Slide 14: Resumo Comparativo (Tabela)

Constraint / Tipo	Uso tpico	Observaes
NOT NULL	evitar nulos	obrigatrio na insero
DEFAULT	valor padro	quando no informado
UNIQUE	evitar duplicatas	aceita NULL
CHECK	validar expresso lgica	MySQL 8.0+ suporta

Constraint / Tipo	Uso típico
-------------------	------------

Observações
-------------

AUTO_INCREMENT	chave sequencial automática só para numéricos
----------------	---

---

### Slide 15: Exercícios Práticos

1. Crie tabela users com:
    - id INT NOT NULL AUTO\_INCREMENT PK
    - username VARCHAR(50) NOT NULL UNIQUE
    - status ENUM('Ativo','Inativo','Pendente') DEFAULT 'Pendente'
    - preferences JSON
  2. Insira alguns registros e faça queries como:
    - Listar usuários ativos
    - Extrair campo preferences.language de JSON
  3. Crie tabela orders validando que total >= subtotal + tax com CHECK
- 

### Slide 16: Conclusão e Dicas

- Escolha de **tipo de dado** e **constraint** adequada melhora integridade e performance
- Use **ENUM/SET** para valores fixos; **JSON** para estruturas flexíveis
- Constraints ajudam a evitar erros e sujeira nos dados