

Aprofundamento SQL

Labenu_



Alterando as propriedades das colunas

Labenu_



Alterando as colunas

- Usamos o comando **ALTER TABLE** para alterar as colunas da tabela

Adicionar coluna

```
ALTER TABLE Tabela  
ADD nova-coluna TIPO RESTRIÇÃO;
```

```
ALTER TABLE Pessoas  
ADD especialidade ENUM('front', 'back') NOT NULL;
```



Alterando as colunas

Deletar coluna

```
ALTER TABLE Tabela  
DROP COLUMN coluna-removida;
```

```
ALTER TABLE Pessoas  
DROP COLUMN email;
```



Alterando as colunas

Editar tipo da coluna

```
ALTER TABLE Tabela  
MODIFY coluna-editada NOVO-TIPO NOVA-RESTRIÇÃO;
```

```
ALTER TABLE Pessoas  
MODIFY id INT AUTO_INCREMENT;
```



Alterando as colunas

Renomear coluna

```
ALTER TABLE Tabela  
CHANGE coluna-editada novo-nome TIPO RESTRIÇÃO;
```

```
ALTER TABLE Pessoas  
CHANGE nome primeiro_nome VARCHAR(255) UNIQUE;
```



Alterando os valores das linhas

Labenu_



Atualizando valores

- Usamos o comando **UPDATE** para atualizar os valores das colunas.
- Quando usado sem **WHERE** altera **todas as linhas** da tabela. ⚠
- Recebe o nome da tabela, o novo valor e a condição.

```
UPDATE Tabela  
SET coluna = novo-valor  
WHERE condição;
```



Coding together

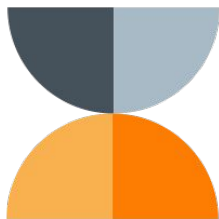


```
UPDATE Pessoas  
SET especialidade = 'back'  
WHERE id = 1;
```



Pausa para relaxar 🤪

10 min



- **ALTER TABLE**
 - Altera as colunas de uma tabela
- **UPDATE TABLE**
 - Atualiza uma ou mais linhas na tabela



Funções

Labenu_



Funções

- Assim como as linguagens de programação, a linguagem SQL também nos fornece funções prontas para performar cálculos comuns.

SOMA todos os valores da coluna

```
SELECT SUM(coluna)
FROM tabela;
```

Calcula a **MÉDIA** dos valores

```
SELECT AVG(coluna)
FROM tabela;
```

CONTA todas os registros de uma coluna

```
SELECT COUNT(coluna) FROM tabela;
```



Funções

Retorna os **menor** e **maior** valor da coluna, respectivamente

```
SELECT MIN(coluna)
FROM tabela;
```

```
SELECT MAX(coluna)
FROM tabela;
```

Retorna **data e hora atual**

```
SELECT NOW()
FROM DUAL;
```

Retorna a diferença entre
data atual e a **data informada**

```
SELECT
DATEDIFF(NOW(), "1995-08-31")
FROM DUAL;
```



Coding together

```
SELECT MIN(idade)
FROM Pessoas;
```

```
SELECT MAX(idade)
FROM Pessoas;
```

```
SELECT AVG(idade)
FROM Pessoas;
```

```
SELECT COUNT(*)
FROM Pessoas;
```



Cláusulas

Labenu_



Cláusulas - ORDER BY

- As instruções SQL possuem cláusulas que executam ações
 - Trabalhamos até aqui com as cláusulas SELECT, INTO, FROM, WHERE, vejamos outras:

Ordena os resultados

```
SELECT coluna  
FROM Tabela  
ORDER BY coluna ASC/DESC;
```



Coding together

Ordem crescente

```
SELECT primeiro_nome  
FROM Pessoas  
ORDER BY idade ASC;
```

Ordem decrescente

```
SELECT primeiro_nome  
FROM Pessoas  
ORDER BY idade DESC;
```



Cláusulas - GROUP BY

Agrupar as linhas que têm os **mesmos valores**, frequentemente utilizado com **funções**.

```
SELECT COUNT(*), coluna-com-valores-repetidos  
FROM tabela  
GROUP BY coluna-com-valores-repetidos;
```

```
SELECT COUNT(*), especialidade  
FROM Pessoas  
GROUP BY especialidade;
```



É isso por hoje!

Labenu_





Obrigado!