

# Aula sobre Unions e Views

## 1. Unions

**Objetivo:** Entender como combinar resultados de múltiplas consultas usando a cláusula **UNION**.

### 1.1. Teoria

O operador **UNION** é usado para combinar os resultados de duas ou mais consultas SQL em um único conjunto de resultados. Ele remove duplicatas por padrão. Se você quiser incluir duplicatas, use **UNION ALL**.

Para que o **UNION** funcione corretamente, todas as consultas devem ter o mesmo número de colunas e as colunas correspondentes devem ter tipos de dados compatíveis.

### 1.2. Exemplo de Tabelas

Vamos usar duas tabelas fictícias, **Vendas\_2023** e **Vendas\_2024**, para demonstrar o uso do **UNION**.

#### Estrutura da Tabela **Vendas\_2023**

sql

Copiar código

```
CREATE TABLE Vendas_2023 (  
    id_venda INT,  
    produto VARCHAR(100),  
    valor DECIMAL(10, 2)  
);  
  
INSERT INTO Vendas_2023 (id_venda, produto, valor)  
VALUES  
(1, 'Produto A', 150.00),  
(2, 'Produto B', 200.00);
```

#### Estrutura da Tabela **Vendas\_2024**

sql

Copiar código

```
CREATE TABLE Vendas_2024 (  
    id_venda INT,  
    produto VARCHAR(100),
```

```
        valor DECIMAL(10, 2)
    );

INSERT INTO Vendas_2024 (id_venda, produto, valor)
VALUES
(1, 'Produto A', 180.00),
(2, 'Produto C', 250.00);
```

### 1.3. Exemplos de Consultas com UNION

#### Combinar Vendas de 2023 e 2024

sql

Copiar código

```
SELECT id_venda, produto, valor
FROM Vendas_2023
UNION
SELECT id_venda, produto, valor
FROM Vendas_2024;
```

1. Esse comando combina os resultados das vendas de 2023 e 2024 em uma única lista, removendo duplicatas.

#### Combinar Vendas de 2023 e 2024 Incluindo Duplicatas

sql

Copiar código

```
SELECT id_venda, produto, valor
FROM Vendas_2023
UNION ALL
SELECT id_venda, produto, valor
FROM Vendas_2024;
```

2. Esse comando combina os resultados das vendas de 2023 e 2024, incluindo duplicatas.

## 2. Views

**Objetivo:** Entender como criar e usar views para simplificar consultas e encapsular lógica de negócios.

### 2.1. Teoria

Uma **VIEW** é uma tabela virtual baseada no resultado de uma consulta SQL. Você pode usar views para simplificar consultas complexas, esconder a complexidade das consultas do usuário, e fornecer uma interface consistente para os dados.

#### Criação de Views:

sql

Copiar código

```
CREATE VIEW nome_da_view AS  
SELECT colunas  
FROM tabela  
WHERE condição;
```

**Uso de Views:** Depois de criada, você pode consultar a view como se fosse uma tabela normal.

### Exemplo de View:

Vamos criar uma view que combina informações de `Vendas_2023` e `Vendas_2024`.

## 2.2. Exemplo de Tabelas

Usaremos as tabelas `Vendas_2023` e `Vendas_2024` novamente.

## 2.3. Criação de uma View

### Criar View para Vendas Combinadas

sql

Copiar código

```
CREATE VIEW Vendas_Combinadas AS  
SELECT id_venda, produto, valor  
FROM Vendas_2023  
UNION ALL  
SELECT id_venda, produto, valor  
FROM Vendas_2024;
```

1. A view `Vendas_Combinadas` combina todas as vendas de 2023 e 2024, incluindo duplicatas.

### Consultar a View

sql

Copiar código

```
SELECT * FROM Vendas_Combinadas;
```

2. Esse comando retorna todos os registros da view `Vendas_Combinadas`.

## 2.4. Alterar e Excluir Views

### Alterar uma View

Para alterar uma view, você pode simplesmente usar o comando `CREATE OR REPLACE VIEW:`

sql

Copiar código

```
CREATE OR REPLACE VIEW Vendas_Combinadas AS  
SELECT id_venda, produto, valor  
FROM Vendas_2023  
UNION  
SELECT id_venda, produto, valor  
FROM Vendas_2024;
```

1.

### Excluir uma View

sql

Copiar código

```
DROP VIEW Vendas_Combinadas;
```

2. Esse comando remove a view `Vendas_Combinadas` do banco de dados.

---

## Resumo

- **UNION** combina os resultados de múltiplas consultas, removendo duplicatas. Use `UNION ALL` para incluir duplicatas.
- **Views** são tabelas virtuais baseadas em consultas SQL que simplificam a consulta de dados complexos e proporcionam uma interface consistente.

Esses conceitos são úteis para gerenciar e consultar dados de forma eficiente em bancos de dados SQL.

4o mini