CEUB

EDUCAÇÃO SUPERIOR

ceub.br

BANCO DE DADOS II

AULA 08

VIEWs

Prof. Leonardo R. de Deus

1. INTRODUÇÃO

Uma VIEW (visão) é uma forma de definir uma consulta que pode ser executada a qualquer momento, sem a necessidade de escrever todo o código da instrução SQL.

1. INTRODUÇÃO

Uma VIEW (visão) é uma forma de definir uma consulta que pode ser executada a qualquer momento, sem a necessidade de escrever todo o código da instrução SQL.

É uma tabela virtual que armazena uma consulta SELECT, que é executada toda vez que a VIEW é chamada.

1. INTRODUÇÃO

Uma VIEW (visão) é uma forma de definir uma consulta que pode ser executada a qualquer momento, sem a necessidade de escrever todo o código da instrução SQL.

É uma tabela virtual que armazena uma consulta SELECT, que é executada toda vez que a VIEW é chamada.

Vamos utilizar uma VIEW para:

- simplificar consultas complexas;
- criar camadas de segurança;
- otimizar o desempenho de relatórios analíticos.

2. VIEW para Simplificar Consultas Complexas

A equipe de vendas precisa, constantemente, de um relatório detalhado dos pedidos da loja, que mostre nome do cliente, o número do pedido, a data, o nome do produto, a quantidade e o valor total do item.

2. VIEW para Simplificar Consultas Complexas

A equipe de vendas precisa, constantemente, de um relatório detalhado dos pedidos da loja, que mostre nome do cliente, o número do pedido, a data, o nome do produto, a quantidade e o valor total do item.

```
SELECT
    c.nome_cliente,
    p.numero_pedido,
    p.data_pedido,
    pr.nome_produto,
    ip.quantidade,
    ip.valor_unitario,
    (ip.quantidade * ip.valor_unitario) AS valor_total_item
FROM loja.tb04_pedido AS p
JOIN loja.tb01_cliente AS c ON p.id_cliente = c.id_cliente
JOIN loja.tb05_item_pedido AS ip ON p.id_pedido = ip.id_pedido
JOIN loja.tb02_produto AS pr ON ip.id_produto = pr.id_produto;
```

2. VIEW para Simplificar Consultas Complexas

A equipe de vendas precisa, constantemente, de um relatório detalhado dos pedidos da loja, que mostre nome do cliente, o número do pedido, a data, o nome do produto, a quantidade e o valor total do item.

```
SELECT
    c.nome_cliente,
    p.numero_pedido,
    p.data_pedido,
    pr.nome_produto,
    ip.quantidade,
    ip.valor_unitario,
    (ip.quantidade * ip.valor_unitario) AS valor_total_item
FROM loja.tb04_pedido AS p
JOIN loja.tb01_cliente AS c ON p.id_cliente = c.id_cliente
JOIN loja.tb05_item_pedido AS ip ON p.id_pedido = ip.id_pedido
JOIN loja.tb02_produto AS pr ON ip.id_produto = pr.id_produto;
```

Escrever esse JOIN entre 4 tabelas toda vez é repetitivo e propenso a erros

2. VIEW para Simplificar Consultas Complexas

A equipe de vendas precisa, constantemente, de um relatório detalhado dos pedidos da loja, que mostre nome do cliente, o número do pedido, a data, o nome do produto, a quantidade e o valor total do item.

SOLUÇÃO: criar uma VIEW para armazenar a consulta

```
CREATE VIEW loja.vw_relatorio_pedidos_detalhado AS
SELECT
    c.nome_cliente,
    p.numero_pedido,
    p.data_pedido,
    pr.nome_produto,
    ip.quantidade,
    ip.valor_unitario,
    (ip.quantidade * ip.valor_unitario) AS valor_total_item
FROM loja.tb04_pedido AS p
JOIN loja.tb01_cliente AS c ON p.id_cliente = c.id_cliente
JOIN loja.tb05_item_pedido AS ip ON p.id_pedido = ip.id_pedido
JOIN loja.tb02_produto AS pr ON ip.id_produto = pr.id_produto;
```

CEUB
EDUCAÇÃO SUPERIOR

Pergunta

#1

Qual o valor total gasto pela cliente “Ana Carolina Souza”?

3. VIEW para Implementar Segurança

O departamento de marketing precisa de uma lista de clientes para uma campanha regional, mas eles não podem ter acesso a dados sensíveis como CPF e endereço. Eles só precisam do nome, email, cidade e estado dos clientes.

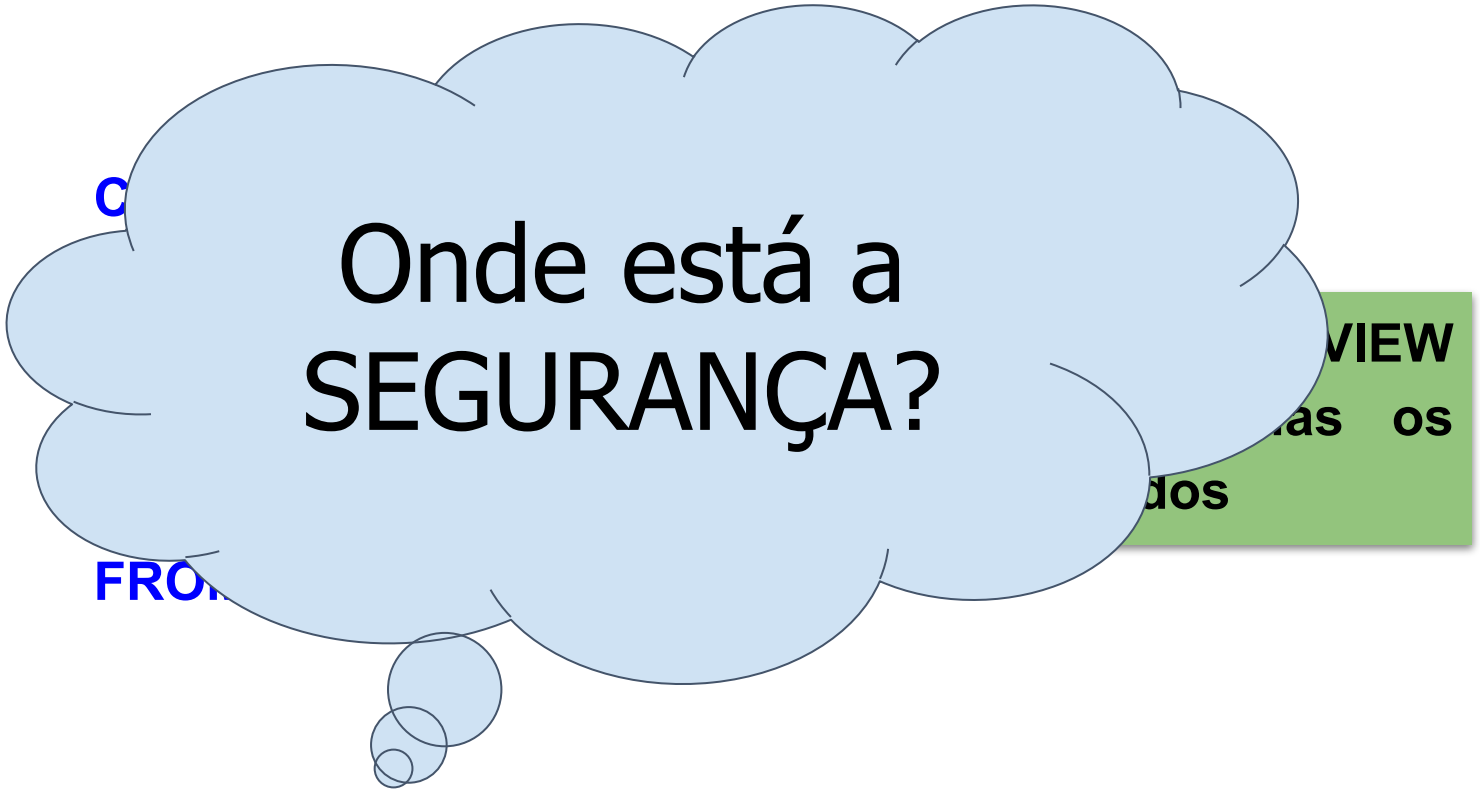
3. VIEW para Implementar Segurança

O departamento de marketing precisa de uma lista de clientes para uma campanha regional, mas eles não podem ter acesso a dados sensíveis como CPF e endereço. Eles só precisam do nome, email, cidade e estado dos clientes.

```
CREATE VIEW loja.vw_clientes_marketing AS
SELECT
    nome_cliente,
    email,
    cidade,
    estado
FROM loja.tb01_cliente;
```

SOLUÇÃO: criar uma VIEW para mostrar apenas os campos desejados

O departamento de marketing precisa de uma lista de clientes para uma campanha regional, mas eles não podem ter acesso a dados sensíveis como CPF e endereço. Eles só precisam do nome, email, cidade e estado dos clientes.



3. VIEW para Implementar Segurança

O departamento de marketing precisa de uma lista de clientes para uma campanha regional, mas eles não podem ter acesso a dados sensíveis como CPF e endereço. Eles só precisam do nome, email, cidade e estado dos clientes.

com Privilégios específicos para cada grupo de usuários.

```
GRANT SELECT ON loja.vw_clientes_marketing TO ROLE_MARKETING;

REVOKE SELECT ON loja.tb01_cliente FROM ROLE_MARKETING;
```

Pergunta

#2

Quais são os clientes do Estado de São Paulo?

Atenção: o usuário não pode ter acesso ao CPF e Endereço do cliente.

3. VIEW para Análise de Dados

Banco Transacional

normalizado

Banco Análitico

desnormalizado; modelagem multidimensional

3. VIEW para Análise de Dados

A equipe de Business Intelligence (BI) precisa analisar as vendas da loja.

Uma abordagem é criar um **Modelo Estrela** usando Views para que eles possam conectar suas ferramentas (Power BI, Tableau) e responder a perguntas de negócio facilmente.

Um Modelo Estrela tem uma tabela de **Fatos** no centro (os números, o que aconteceu) e várias tabelas de **Dimensões** ao redor (quem, o quê, quando, onde)

3. VIEW para Análise de Dados

Dimensão Cliente

```
CREATE VIEW loja.vw_dim_cliente AS
SELECT
    id_cliente,
    nome_cliente,
    cidade,
    estado,

    -- Atributo Derivado 1: Calculando a idade atual do cliente
    EXTRACT(YEAR FROM AGE(CURRENT_DATE, data_nascimento)) AS idade,

    -- Atributo Derivado 2: Criando uma segmentação por faixa etária
    CASE
        WHEN EXTRACT(YEAR FROM AGE(CURRENT_DATE, data_nascimento)) < 25 THEN '18-24 anos'
        WHEN EXTRACT(YEAR FROM AGE(CURRENT_DATE, data_nascimento)) BETWEEN 25 AND 34 THEN '25-34 anos'
        WHEN EXTRACT(YEAR FROM AGE(CURRENT_DATE, data_nascimento)) BETWEEN 35 AND 44 THEN '35-44 anos'
        ELSE '45+ anos'
    END AS faixa_etaria
FROM loja.tb01_cliente;
```

3. VIEW para Análise de Dados

Dimensão Produto

```
CREATE VIEW loja.vw_dim_produto AS
SELECT
    id_produto, nome_produto, categoria, preco,

    -- Atributo Derivado: Criando uma classificação de preço
    CASE
        WHEN preco < 100 THEN 'Acessível'
        WHEN preco BETWEEN 100 AND 500 THEN 'Intermediário'
        WHEN preco > 500 THEN 'Premium'
        ELSE 'Não classificado'
    END AS faixa_de_preco,

    -- Atributo Derivado: Indicador de disponibilidade
    CASE
        WHEN estoque > 0 THEN 'Disponível'
        ELSE 'Indisponível'
    END AS status_estoque
FROM loja.tb02_produto;
```

3. VIEW para Análise de Dados

Dimensão Tempo

```
CREATE VIEW loja.vw_dim_tempo AS
SELECT DISTINCT
      EXTRACT(YEAR FROM data_pedido) AS ano,
      EXTRACT(MONTH FROM data_pedido) AS mes,
      EXTRACT(QUARTER FROM data_pedido) AS trimestre
FROM loja.tb04_pedido;
```

3. VIEW para Análise de Dados

Tabela Fato Vendas

```
CREATE VIEW loja.vw_fato_vendas AS
SELECT
    -- Chaves Estrangeiras (Links para as Dimensões)
    p.id_cliente,
    ip.id_produto,
    p.data_pedido,
    p.id_status_pedido,

    -- Fatos / Métricas Numéricas (O que queremos medir)
    ip.quantidade AS fato_quantidade_vendida,
    ip.valor_unitario AS fato_valor_unitario_venda,
    (ip.quantidade * ip.valor_unitario) AS fato_faturamento_total
FROM  loja.tb05_item_pedido AS ip
JOIN  loja.tb04_pedido AS p ON ip.id_pedido = p.id_pedido;
```

CEUB
EDUCAÇÃO SUPERIOR

Pergunta

#3

Qual o mês de maior faturamento da loja?

Pergunta

#4

Qual a faixa etária que mais gasta com produtos da categoria "Premium"?

4. VIEW Materializada

Se o banco crescer bastante ou as VIEWS forem muito complexas, o banco pode ficar lento para executar estas consultas.

4. VIEW Materializada

Se o banco crescer bastante ou as VIEWS forem muito complexas, o banco pode ficar lento para executar estas consultas.

VIEW MATERIALIZADA

4. VIEW Materializada

É uma tabela física que armazena o resultado de uma consulta.

Ponto Positivo: é rápida

Ponto Negativo: pode ficar desatualizada

VIEW MATERIALIZADA

Vamos criar uma view que mostre o número de itens vendidos por categoria.

VIEW MATERIALIZADA

NA PRÁTICA

```
CREATE VIEW loja.vw_qtd_itensvendidos_categoria AS
SELECT
    pr.categoria,
    SUM(ip.quantidade) AS total_itens_vendidos
FROM
    loja.tb05_item_pedido AS ip
JOIN
    loja.tb02_produto AS pr ON ip.id_produto = pr.id_produto
GROUP BY pr.categoria
ORDER BY total_itens_vendidos DESC;
```

Vamos criar uma view que mostre o número de itens vendidos por categoria.

VIEW MATERIALIZADA

NA PRÁTICA

```
CREATE MATERIALIZED VIEW
loja.vw_mat_qtd_itensvendidos_categoria AS
SELECT
    pr.categoria,
    SUM(ip.quantidade) AS total_itens_vendidos
FROM
    loja.tb05_item_pedido AS ip
JOIN
    loja.tb02_produto AS pr ON ip.id_produto = pr.id_produto
GROUP BY pr.categoria
ORDER BY total_itens_vendidos DESC;
```

Incluindo um novo pedido no banco:

VIEW MATERIALIZADA

NA PRÁTICA

```
INSERT INTO loja.tb04_pedido
(numero_pedido, data_pedido, id_cliente, id_status_pedido)
VALUES ('PED-2025-00026', '2025-09-16',20, 3);

INSERT INTO loja.tb05_item_pedido
(id_pedido, id_produto, quantidade, valor_unitario) VALUES
-- Adicionando 5 itens na categoria 'Livros'
(26, 8, 5, 89.90),
-- Adicionando 10 itens na categoria 'Esportes'
(26, 16, 10, 199.90),
-- Adicionando 2 itens na categoria 'Eletrônicos'
(26, 5, 2, 450.00);
```

Atualizando a view materializada:

VIEW MATERIALIZADA

NA PRÁTICA

```
REFRESH MATERIALIZED VIEW loja.vw_mat_qtd_itensvendidos_categoria;
```

OBRIGADO
A TODOS!

CEUB

EDUCAÇÃO SUPERIOR