



**INSTITUTO
FEDERAL**
Pernambuco

Linguagem de 3^a Geração

Banco de Dados 2

Professora Carolina Torres

Linguagem de 3^a Geração

- Linguagem de programação sofisticada, utilizada para ter acesso a uma base de dados a partir de vários ambientes
- Combina o poder e a flexibilidade de SQL com as estruturas de código de procedimentos encontradas nas linguagens de programação de 3a. geração

Linguagem de 3^a Geração

- Permite algumas estruturas de procedimento, como:
 - Variáveis e tipos (pré-definidos ou não)
 - Estruturas de controle (IF-THEN-ELSE e laços)
 - Procedimentos e funções

Esquema Adotado

SGBD Postegres

```
create table setor(  
    cod_setor integer primary key,  
    nome_setor varchar (30) );
```

```
create table produto(  
    cod_produto integer primary key,  
    nome_produto varchar(30),  
    preço decimal (8,2),  
    categoria varchar(30),  
    cd_setor integer,  
    foreign key (cd_setor) references  
    setor (cod_setor) );
```

```
create table fornecedor (  
    cod_fornecedor integer primary key,  
    nome_fornecedor varchar(30),  
    cidade varchar(30) );
```

```
create table forn_Prod(  
    cd_fornecedor integer,  
    cd_produto integer,  
    primary key (cd_fornecedor,cd_produto),  
    foreign key (cd_fornecedor) references  
    fornecedor (cod_fornecedor),  
    foreign key (cd_produto) references  
    produto (cod_produto) );
```

Visões

...

São tabelas virtuais que não ocupam espaço físico
Otimizam consultas muito utilizadas

Visões

- Sintaxe de definição -



**INSTITUTO
FEDERAL**
Pernambuco

```
CREATE VIEW <nome_da_visão>  
    [(<nome_das_colunas>)]  
    AS <consulta_SQL>
```

Visões

- Acesso -

- Acessada como uma tabela comum

```
SELECT * FROM <nome_da_visao>
```

- Ainda permite aplicações de todas as cláusulas (*where, group by, order by e having*)

```
SELECT <campos>  
FROM <nome_da_visao>  
WHERE <condições>  
GROUP BY <alguns_campos>  
HAVING <condição_filtro_agrupamento>  
ORDER BY <campos>
```

Visões

- Operações -

- Alterar visões:
 - ALTER VIEW *nome_da_visao* ALTER *column_name* SET DEFAULT *expressao*
 - ALTER VIEW *nome_da_visao* ALTER *column_name* DROP DEFAULT
 - ALTER VIEW *nome_antigo* RENAME TO *nome_novo*
- Deletar visões:
 - DROP VIEW *nome_da_visao*

Visões

- Exemplos -

1. Criar a visão ***listagem_produtos*** que liste o nome dos produtos e seu preço ordenados por preço
2. Selecionar apenas os dados dessa visão com nomes de produto que comecem com C.
3. Deletar essa visão

```
1. Create view listagem_produtos AS  
select nome_produto, preço from produto  
order by preço;
```

```
2. Select * from listagem_produtos  
WHERE nome_produto like 'C%';
```

```
3. DROP VIEW listagem_produtos
```

Visões

- Exercícios-

1. Criar a visão **produtos_setor** que:
 - Apresente o nome dos produtos, o preço e o nome do setor ao qual pertencem
 - As colunas devem ser exibidas como **produto**, **preco** e **setor**.
2. Alterar o nome da visão que acabou de criar para **produtos_preco_setor**;
3. Selecionar apenas os dados da visão que possuam preço acima de R\$300,02;
4. Apagar a visão **produtos_preco_setor**;
5. Criar a visão **v_prod_forn** que liste o nomes dos produtos e os nomes dos seus fornecedores;

Consulte a visão **v_prod_forn**, exibindo:

6. O nome do fornecedor e a quantidade de produtos que ele fornece;
7. A listagem em ordem alfabética pelo nome dos produtos depois pelos nomes dos fornecedores em ordem contrária à alfabética;
8. Quantos produtos são oferecidos pelo fornecedor 'Unilever'.



**INSTITUTO
FEDERAL**
Pernambuco

Linguagem de 3^a Geração

Banco de Dados 2

Professora Carolina Torres