

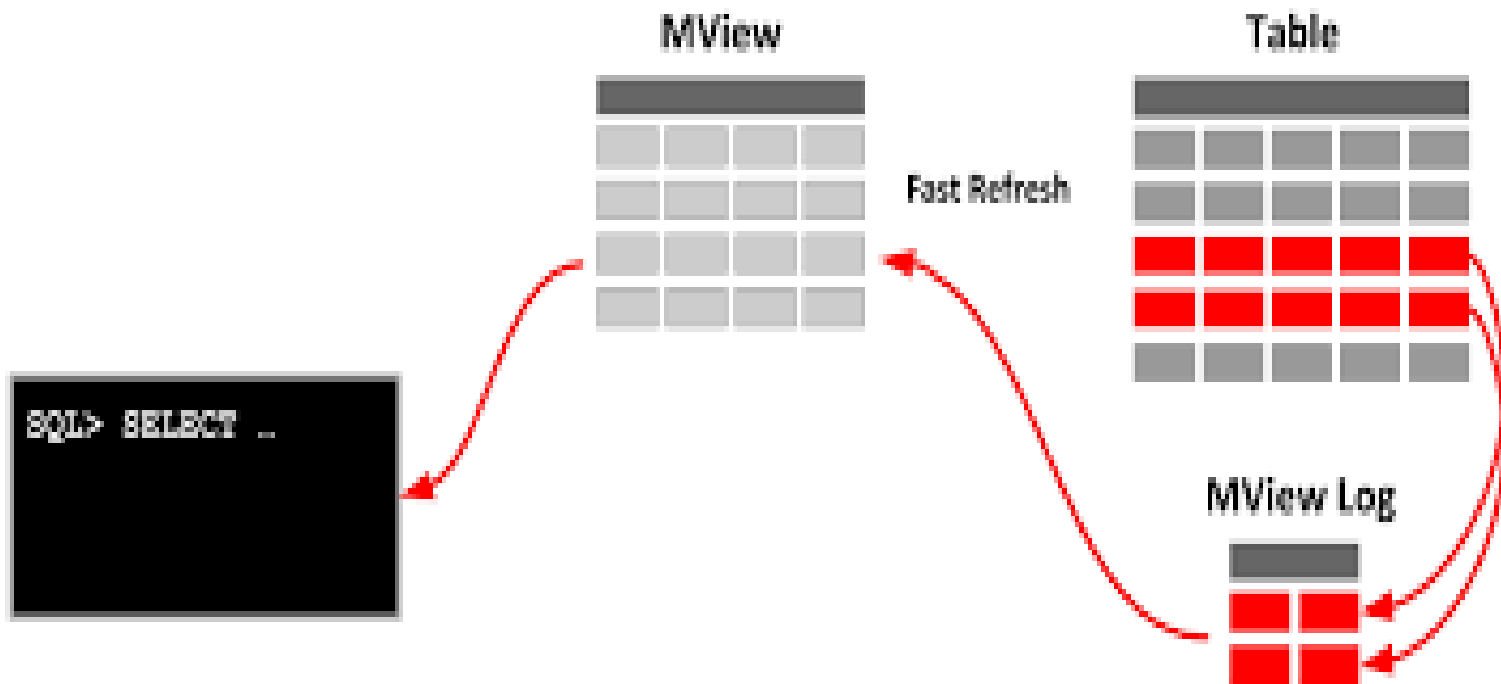


BANCO DE DADOS

Outros Objetos

Ma. Simone Maria Viana Romano

OUTROS OBJETOS



OUTROS OBJETOS

Objetos de banco de dados são as unidades lógicas que compõem os blocos de construção dos bancos de dados.

Um objeto de banco de dados se refere a qualquer objeto que é utilizado para armazenar ou referenciar dados. Alguns exemplos destes objetos são: tabelas, synonyms, visões, clusters, sequences e índices.



OUTROS OBJETOS: SEQUENCE

- Estruturas criadas no banco de dados que retornam um valor diferente a cada acesso. Sintaxe:

```
CREATE SEQUENCE nome_sequência  
[START WITH valor_inicial] (default 1)  
[INCREMENT BY incremento] (default 1)  
[MAXVALUE valor_maximo_sequencia/NOMAXVALUE]  
[MINVALUE valor_minimo_sequencia/NOMINVALUE]  
[CYCLE|NOCYCLE]  
[CACHE numero_cache|NOCACHE]  
[ORDER|NOORDER];
```



OUTROS OBJETOS: VIEW (Visão)

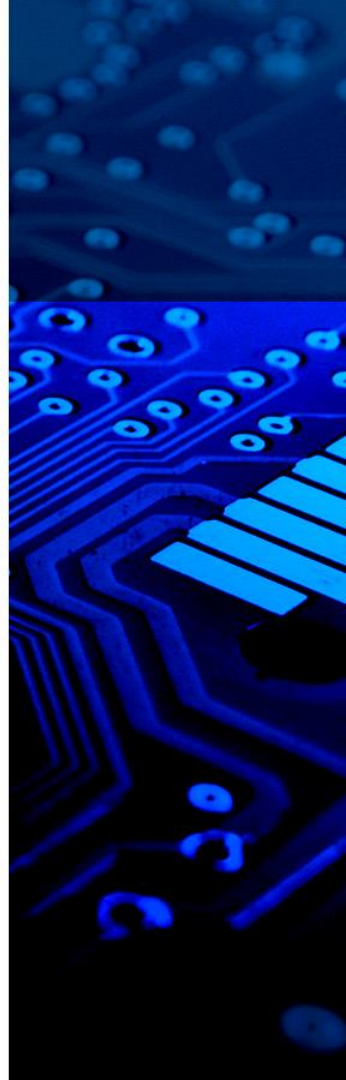
Tabela virtual na qual os dados não estão fisicamente armazenados.

Utilizamos visões para:

- Restringir o acesso a dados;
- Facilitar as consultas complexas;
- Permitir a independência dos dados;
- Apresentar diferentes visualizações dos mesmos dados.

Podem ser classificadas como:

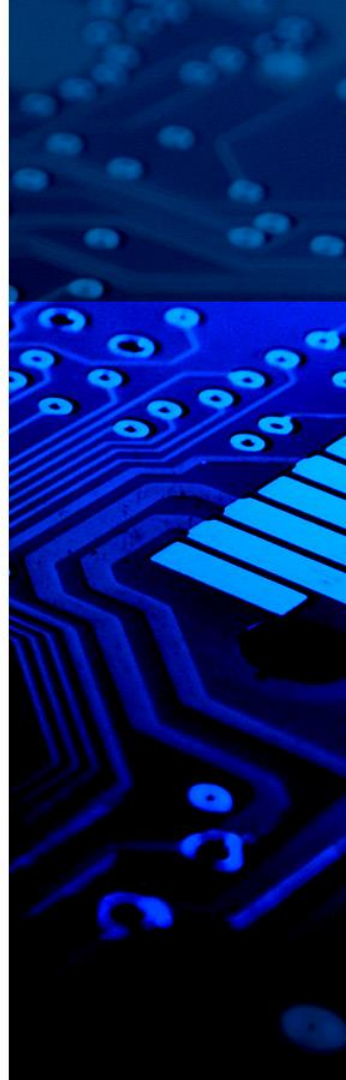
- **SIMPLES**: somente uma tabela, não contém funções e nem grupos e faz operação DML.
- **COMPLEXA**: uma ou mais tabelas, contém funções, grupos de dados e não faz uma operação DML.



OUTROS OBJETOS: VIEW (Visão)

Podemos executar operações DML nos dados através de uma *view* se essas operações não tiverem uma destas exceções:

- Funções de grupo;
- GROUP BY;
- DISTINCT;
- ROWNUM;
- Colunas definidas por expressões (exemplo: salário * 1.10);
- Colunas NOT NULL nas tabelas-base que não sejam selecionadas pela *view*.



OUTROS OBJETOS: VIEW (Visão)

Sintaxe:

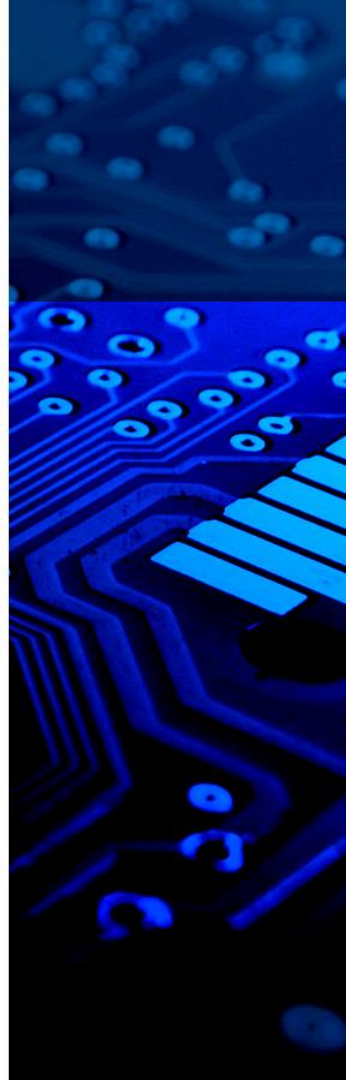
CREATE [OR REPLACE] [FORCE | NOFORCE] VIEW nome_view
[apelido1, apelido2, ...]

AS subconsulta

[WITH CHECK OPTION [*CONSTRAINT* restrição]]

[WITH READ ONLY [*CONSTRAINT* restrição]];

CLAUSULA	DESCRIÇÃO
REPLACE	Recria a view, se ela já existir.
FORCE	Cria a view, mesmo que a tabela não exista.
NOFORCE	Só cria a view se a tabela existir. (padrão)
CONSULTA	Comando SELECT. Não se usa o ORDER BY.
WITH CHECK OPTION	Somente linhas acessíveis pela view (cláusula where na sua definição), podemos inserir e alterá-las.
WITH READ ONLY	Indica que não podem ser executados comandos DML sobre a views.



OUTROS OBJETOS: VIEW (Visão)

- **Eliminar uma *view* :**

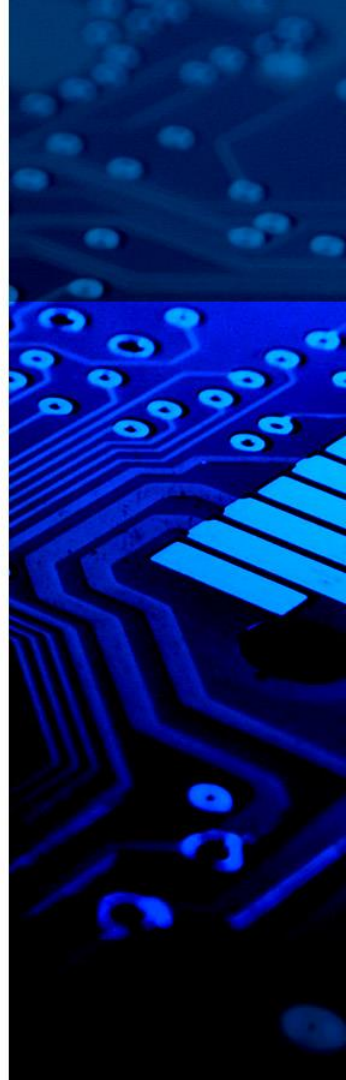
```
DROP VIEW nome_view;
```

- **Estrutura de uma *view*:**

```
DESC ou DESCRIBE nome_view;
```

- **Verificando dados de uma *view*:**

```
SELECT * FROM nome_view;
```



OUTROS OBJETOS: VIEW (Visão)

WITH READ ONLY

Nenhum comando DML será executado através da *view*. Qualquer tentativa com relação a um comando DML ocasionará um erro. Sintaxe:

```
CREATE OR REPLACE VIEW nome_view (coluna1, coluna2)
```

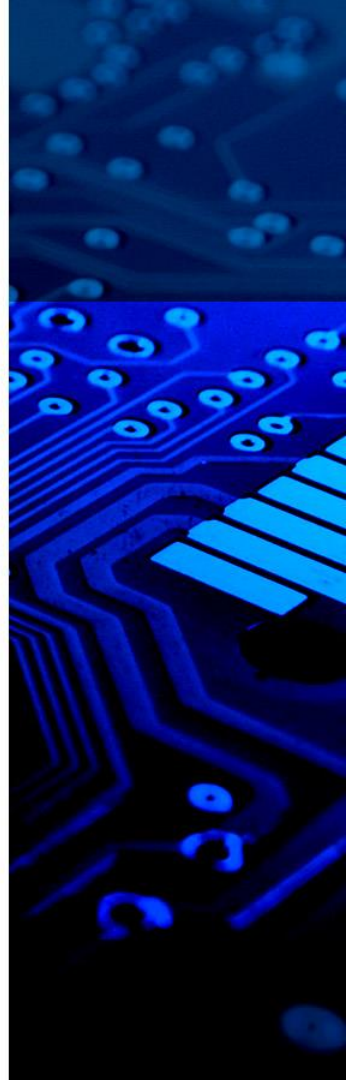
```
AS
```

```
SELECT coluna1, coluna2
```

```
    FROM tabela
```

```
    [WHERE condição]
```

```
WITH READ ONLY;
```



OUTROS OBJETOS: VIEW (Visão)

WITH CHECK OPTION

Verifica a integridade referencial através das *views*.

Podemos impor restrições no nível de banco de dados.

Sintaxe:

```
CREATE OR REPLACE VIEW nome_view AS
```

```
consulta
```

```
WITH CHECK OPTION CONSTRAINT nome_restrição;
```



OUTROS OBJETOS: VIEW (Visão)

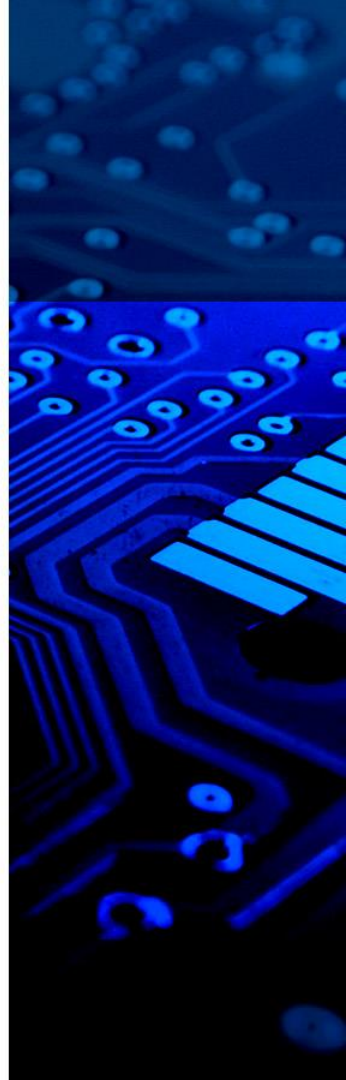
ANÁLISE TOP-N

Permite exibir apenas os n maiores ou os n menores registros de uma tabela com base em uma condição.

Exemplo: exibir os três maiores salários da empresa, os quatro mais novos contratados da empresa, os dois representantes de vendas que mais vendem produtos, etc.

Sintaxe:

```
SELECT lista_colunas, ROWNUM  
FROM (subconsulta com ORDER BY)  
WHERE ROWNUM <= N;
```



OUTROS OBJETOS: VIEW MATERIALIZADA

Materialized views (anteriormente, *Snapshot*);

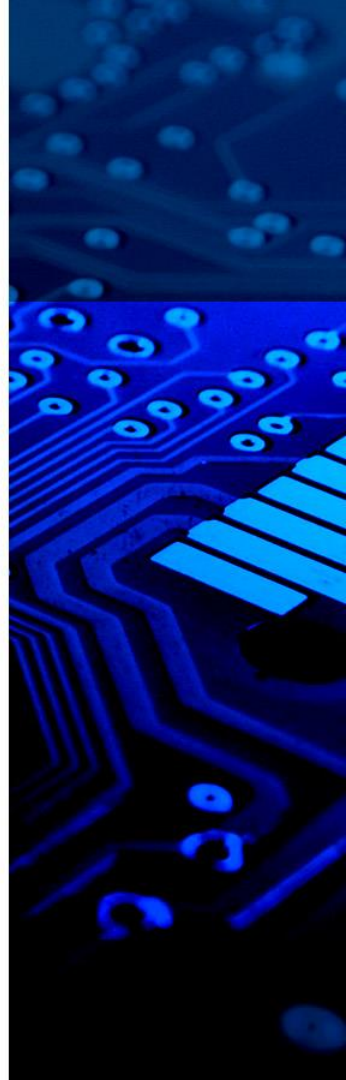
Armazenar dados de consultas que precisam ler uma grande quantidade de dados de diferentes tabelas/views;

Permite a execução de queries complexas, lendo um único objeto de banco de dados;

Diferença da view e da view materializada:

View: armazena a consulta;

View materializada: armazena o resultado desta consulta.

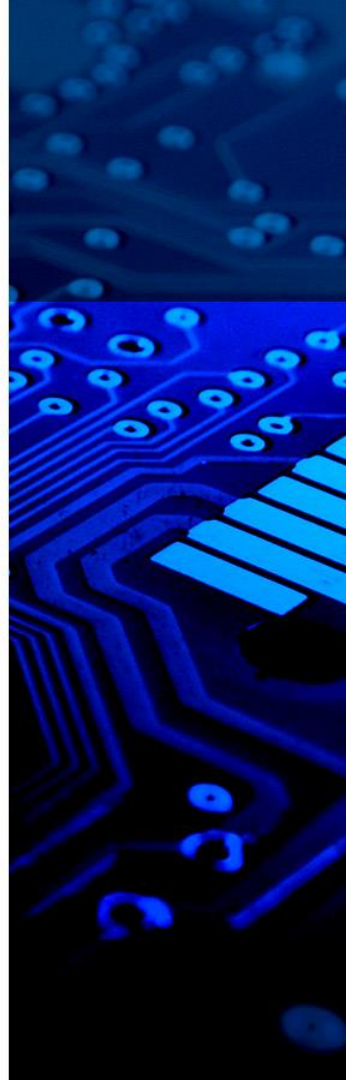


OUTROS OBJETOS: VIEW MATERIALIZADA

Sintaxe:

```
CREATE MATERIALIZED VIEW nome_view  
REFRESH FAST  
START WITH SYSDATE + 1  
NEXT SYSDATE + 1/24  
WIDTH PRIMARY KEY  
AS  
consulta;
```

CLAUSULA	DESCRIÇÃO
REFRESH	<p>Tipo de atualização que sofrerão os dados da visão. Há 3 estados ou métodos:</p> <p>FAST: Atualiza somente as linhas da tabela original que sofreram atualizações.</p> <p>COMPLETE: Recria toda a estrutura da view materializada mesmo que não seja necessário.</p> <p>FORCE: por padrão o método é FORCE. Tentará fazer uma atualização FAST se não for possível irá fazer a COMPLETE.</p>
NEXT	Determina o intervalo entre a próxima replicação.
WIDTH PRIMARY KEY	É utilizado para criar a Chave-Primária, baseada na tabela máster.



OUTROS OBJETOS: SEQUÊNCIAS (SEQUENCE)

CLAUSULA	DESCRIÇÃO
SEQUENCIA	Nome da sequência. Obs. não pode ser o mesmo de uma tabela ou view.
INCREMENT BY n	Especifica de quanto será o incremento da sequência. O default é 1.
START WITH n	Especifica o primeiro número a ser gerado. O default é 1.
MAXVALUE n	Especifica o valor máximo da sequência. Default é NOMAXVALUE, indo até 10^{27}
MINVALUE n	Especifica o valor mínimo para sequências.
CYCLE NOCYCLE	Quando atingir o valor máximo, a numeração continuará a partir do valor inicial. O default é NOCYCLE.
CACHE NOCACHE	Número de valores inteiros a manter na memória (padrão 20). O número mínimo pode ser 2. NOCACHE significa sem memória cachê.
ORDER NOORDER	Garante que os números inteiros sejam gerados na ordem da solicitação (ORDER). O NOORDER não garante esta ordem (é o padrão).

OUTROS OBJETOS: SEQUÊNCIAS (SEQUENCE)

- **Confirmando a existência de sequences:**
`SELECT * FROM user_sequences;`
- **Eliminando uma sequence:**
`DROP SEQUENCE nome_sequencia;`
- **OBSERVAÇÕES:**
 - Se for executado um ROLLBACK, o número da sequência não retornará ao valor anterior;
 - Duas pseudo-colunas são utilizadas nas sequências:
 - ***NEXTVAL***: retorna o próximo valor da sequência;
 - ***CURRVAL***: retorna valor corrente da sequência.
- **SINTAXE:**
 - **NEXTVAL**: `INSERT INTO tabela (coluna1, coluna2)
VALUES (nome_sequencia.NEXTVAL, 'valor');`
 - **CURRVAL**: `SELECT nome_sequencia.CURRVAL FROM dual;`



OUTROS OBJETOS: SINÔNIMOS (SYNONYM)

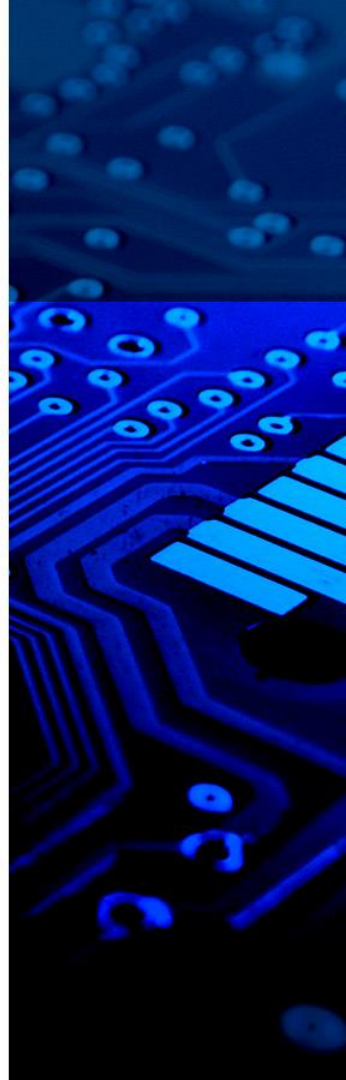
- Um banco de dados é utilizado por dezenas de usuários que têm vários objetos;
- Usado para facilitar a manipulação desses objetos.
- SINTAXE: CREATE [**PUBLIC**] SYNONYM sinônimo FOR objeto;

OBSERVAÇÕES:

PUBLIC: o sinônimo poderá ser usado por todos usuários.

O default é que somente o criador possa utilizá-lo.

Para criar este sinônimo devemos estar conectados como usuário SYSTEM.



OUTROS OBJETOS: SINÔNIMOS(SYNONYM)

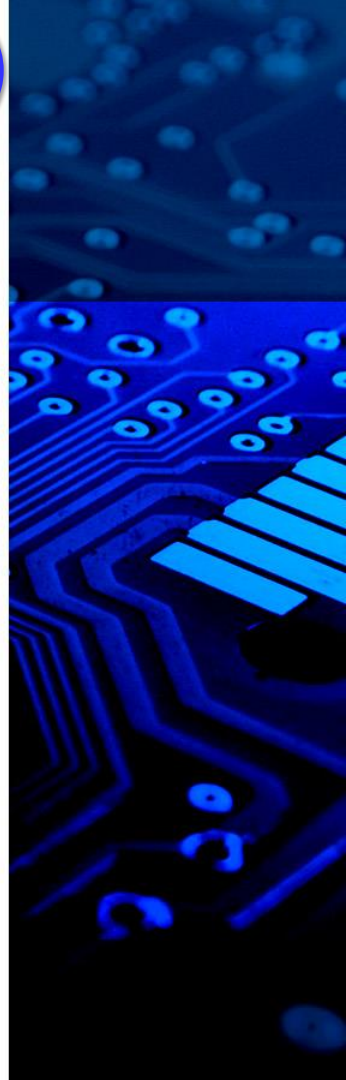
- Um banco de dados é utilizado por dezenas de usuários que têm vários objetos;
- Usado para facilitar a manipulação desses objetos.
- SINTAXE: CREATE [**PUBLIC**] SYNONYM sinônimo FOR objeto;

OBSERVAÇÕES:

PUBLIC: o sinônimo poderá ser usado por todos usuários.

O default é que somente o criador possa utilizá-lo.

Para criar este sinônimo devemos estar conectados como usuário SYSTEM.



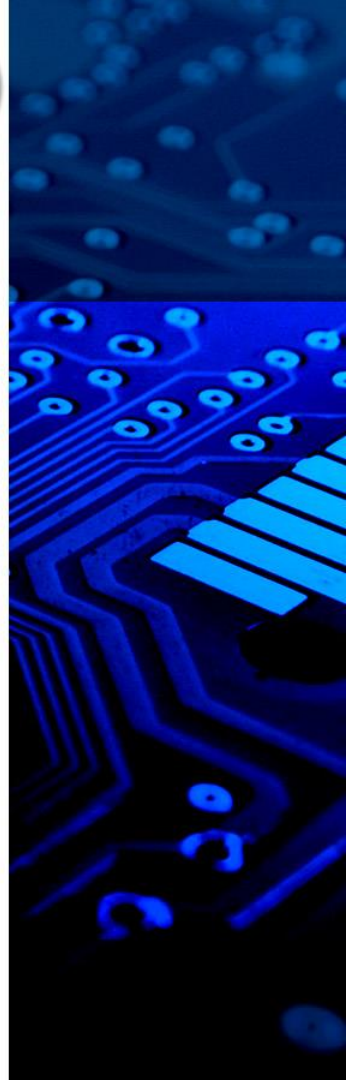
OUTROS OBJETOS: SINÔNIMOS(SYNONYM)

Consultar sinônimos existentes:

```
SELECT SYNONYM_NAME, TABLE_OWNER  
      FROM ALL_SYNONYMS  
[WHERE TABLE_NAME = 'nome_tabela'];
```

Excluir sinônimos:

```
DROP [PUBLIC] SYNONYM nome_sinônimo;
```



OUTROS OBJETOS: INDICE (INDEX)

Chamado de *B-TREE* (ARVORE B*)

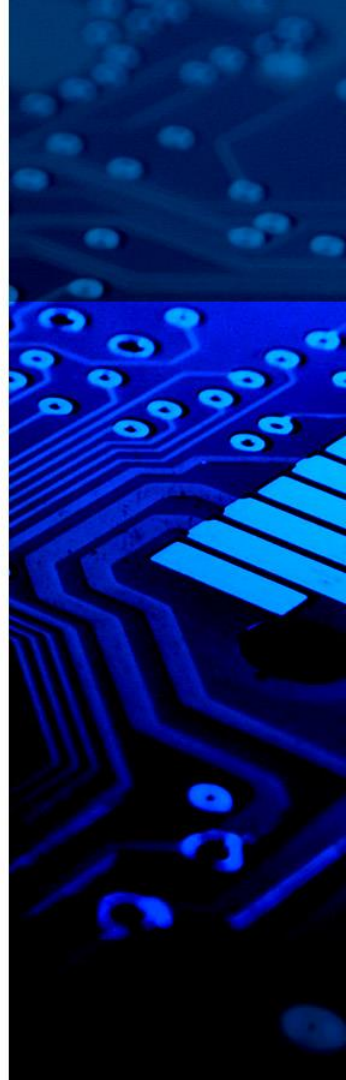
Objeto de um esquema que pode acelerar a recuperação de linhas usando um ponteiro.

Criados de duas formas: explícita ou implícita.

O índice é utilizado e mantido automaticamente pelo banco de dados.

Desvantagem: quando é inserido um registro novo há um gasto um tempo para atualizar o índice da nova linha.

É criado um índice em uma coluna que é recuperado um pequeno número de registros. Uma boa regra: ***Crie um índice quando uma consulta recuperar no máximo 10% do total de registros de uma tabela.***

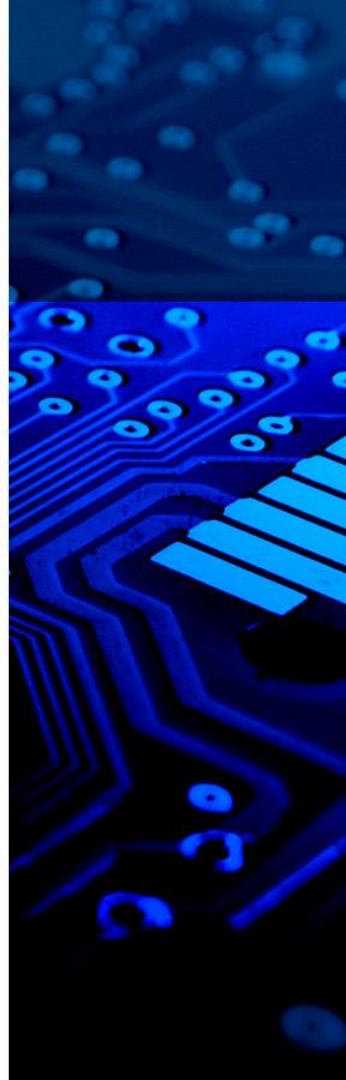


OUTROS OBJETOS: INDICE (INDEX)

É muito importante lembrar que ao eliminar uma tabela, os índices correspondentes também são eliminados.

Há dois tipos de índices:

- **índice exclusivo:** criado automaticamente quando é definido uma PRIMARY KEY ou UNIQUE KEY, onde nome do índice é o nome da *constraint*;
- **índice não-exclusivo:** criar, por exemplo, um índice na FOREIGN KEY para uma junção em uma consulta para melhorar a performance.



OUTROS OBJETOS: INDICE (INDEX)

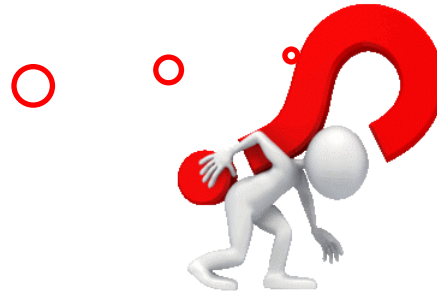
Quando é que devemos
criar um índice?



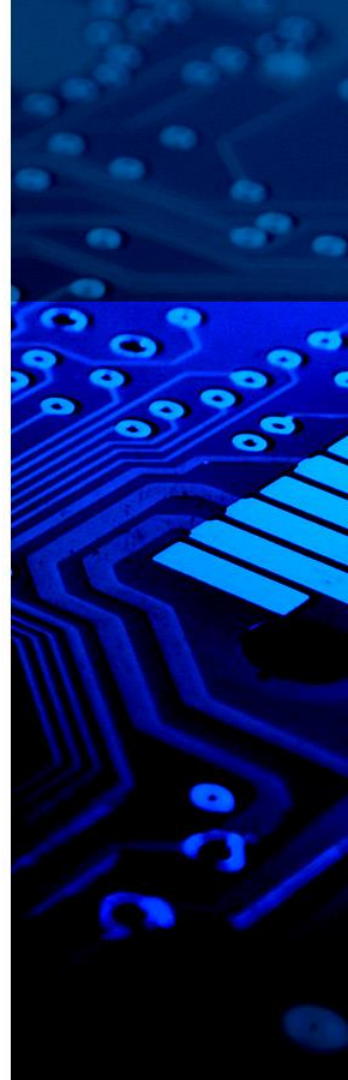
- Quando a coluna for usada frequentemente na cláusula WHERE ou em uma condição de junção.
- Quando a coluna contiver uma ampla faixa de valores.
- Quando a coluna contiver um grande número de valores nulos.
- Quando duas ou mais colunas forem usadas juntas com frequência em uma cláusula WHERE ou em uma condição de junção.
- Quando a tabela for grande e se esperar que a maioria das consultas recupere menos que 2 a 4% das linhas.

OUTROS OBJETOS: INDICE (INDEX)

Quando é que **NÃO**
devemos criar um índice?



- Quando a tabela for pequena.
- Quando as colunas não forem utilizadas com frequência como uma condição na consulta.
- Quando se esperar que a maioria das consultas recupere mais que 2 a 4% das linhas.
- Quando a tabela for atualizada com frequência. Se você tiver um ou mais índices em uma tabela, as instruções DML que acessarem a tabela serão mais lentas.



OUTROS OBJETOS: ÍNDICE (INDEX)

CRIAR ÍNDICE ARVORE B:

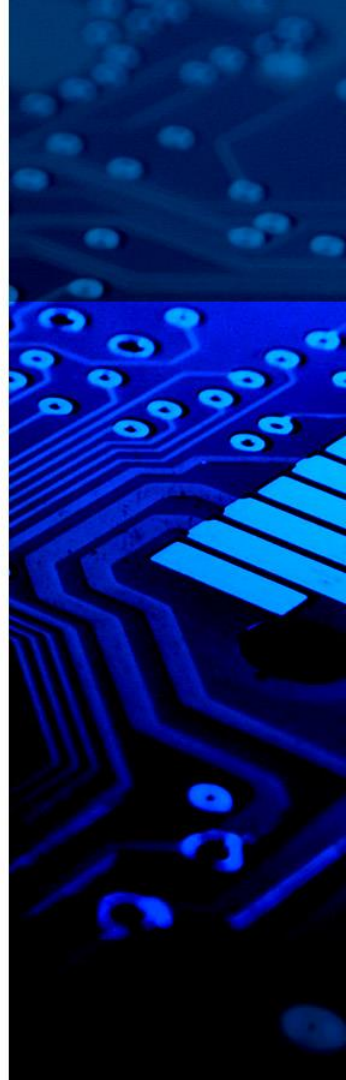
**CREATE [UNIQUE] INDEX índice ON tabela (coluna[,
coluna]...);**

- UNIQUE permitidos somente valores exclusivos na coluna indexada.
- coluna é o nome da coluna na tabela a ser indexada.

• RENOMEAR ÍNDICE:

ALTER INDEX nome_antigo RENAME TO nome_novo

- **EXCLUIR ÍNDICE: DROP INDEX nome_indice;**



OUTROS OBJETOS: INDICE (INDEX)

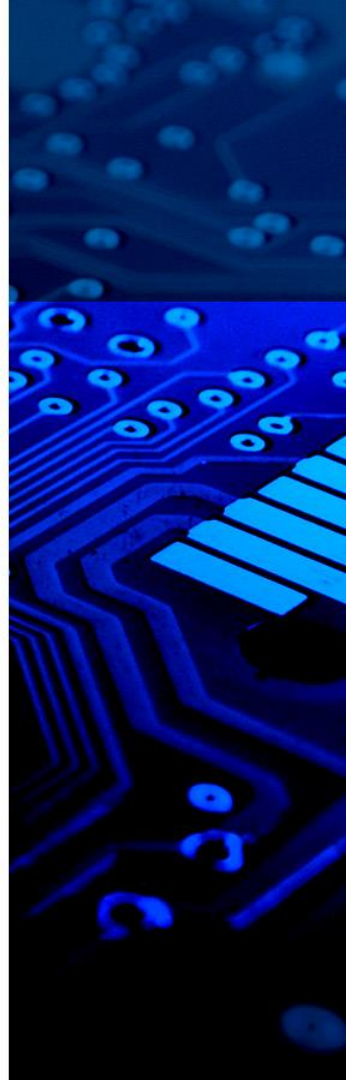
CONSULTAR INDICES EXISTENTES NO BANCO DE DADOS:

```
SELECT index_name, table_owner, table_name, uniqueness,  
       status  
FROM USER_INDEXES;
```

```
SELECT index_name, table_name e column_name  
FROM USER_IND_COLUMNS;
```

INDICES QUE TEMOS ACESSO:

```
SELECT * FROM ALL_IND_COLUMNS;
```



SAIBA MAIS

- https://docs.oracle.com/cd/E11882_01/server.112/e25494/views.htm#ADMIN020
- https://docs.oracle.com/cd/B10500_01/server.920/a96521/views.htm
- <https://www.devmedia.com.br/gerenciando-objetos-de-banco-de-dados-revista-sql-magazine-90/22004>

