



# ORACLE

## Academy



# Database Programming with SQL

16-2

Índices e Sinônimos

**ORACLE**  
Academy



# Objetivos

- Esta lição abrange os seguintes objetivos:
  - Definir um índice e seu uso como um objeto de esquema
  - Nomear as condições que fazem um índice ser criado automaticamente
  - Criar e executar uma instrução CREATE INDEX e DROP INDEX
  - Criar e executar um índice baseado em funções
  - Criar sinônimos privados e públicos

# Finalidade

- Já pensou se você fosse a uma livraria ou biblioteca e encontrasse todos os livros enfileirados nas prateleiras sem seguirem uma ordem?
- O único jeito de achar o que você está procurando será examinar todos os livros em todas as prateleiras!
- Isso não é algo muito eficiente, não é?
- Talvez você se surpreenda em saber que a recuperação de dados de um banco de dados é exatamente como a situação dos livros
- Para cada consulta, uma verificação completa da tabela ocorre

# Finalidade

- Felizmente, existe uma maneira de tornar a busca de dados mais eficiente
- O Oracle usa um índice para acelerar a recuperação de linhas
- Nesta lição, você aprenderá a criar e excluir um índice e quando fazer isso
- Você também saberá como criar nomes amigáveis para os objetos de banco de dados
- Do mesmo modo que os endereços Web, que eliminam a necessidade de saber o endereço de protocolo IP, os sinônimos oferecem uma maneira de dar um nome mais simples a um objeto (Parecido com os aliases, não é?)

# Índices

- Um índice do Servidor Oracle é um objeto de esquema que pode acelerar a recuperação de linhas com o uso de um ponteiro. Os índices podem ser criados de maneira explícita ou automática
- Se você não tiver um índice na coluna que está selecionando, uma verificação completa da tabela ocorrerá
- Um índice fornece acesso rápido e direto às linhas em uma tabela
- Seu propósito é reduzir a necessidade de entrada/saída de disco, usando um caminho indexado para localizar os dados rapidamente

# Índices

- O índice é usado e mantido automaticamente pelo Servidor Oracle. Após a criação do índice, nenhuma atividade direta do usuário é necessária
- ROWID é uma representação básica com 64 strings do endereço de linha e contém o identificador de bloco, a localização da linha no bloco e o identificador de arquivo do banco de dados
- Os índices usam ROWIDs porque são a maneira mais rápida de acessar qualquer linha



# Índices

- Os índices são lógica e fisicamente independentes da tabela que indexam.
- Isso significa que podem ser criados ou eliminados a qualquer momento e não afetam as tabelas básicas ou outros índices

COUNTRY_ID	COUNTRY_NAME	CAPITOL	REGION_ID
1	United States of America	Washington, DC	21
2	Canada	Ottawa	21
3	Republic of Kazakhstan	Astana	143
7	Russian Federation	Moscow	151
12	Coral Sea Islands Territory	-	9
13	Cook Islands	Avarua	9
15	Europa Island	-	18
20	Arab Republic of Egypt	Cairo	15
...	...	...	...



# Índices

- Observação: quando você elimina uma tabela, os índices correspondentes também são eliminados

COUNTRY_ID	COUNTRY_NAME	CAPITOL	REGION_ID
1	United States of America	Washington, DC	21
2	Canada	Ottawa	21
3	Republic of Kazakhstan	Astana	143
7	Russian Federation	Moscow	151
12	Coral Sea Islands Territory	-	9
13	Cook Islands	Avarua	9
15	Europa Island	-	18
20	Arab Republic of Egypt	Cairo	15
...	...	...	...

# Tipos de Índices

- Dois tipos de índices podem ser criados:
  - Índice exclusivo: o Servidor Oracle cria automaticamente esse índice quando você define uma coluna em uma tabela para ter uma constraint PRIMARY KEY ou UNIQUE KEY.
    - O nome do índice é o nome dado à constraint
    - Embora você possa criar manualmente um índice exclusivo, é recomendado criar uma constraint exclusiva na tabela, que cria um índice exclusivo explicitamente
  - Índice não exclusivo: é o índice que o usuário pode criar para acelerar o acesso às linhas
    - Por exemplo, para otimizar junções, você pode criar um índice na coluna FOREIGN KEY, o que acelera a pesquisa para relacionar as linhas à coluna PRIMARY KEY

# Criando um Índice

- Crie um índice em uma ou mais colunas executando a instrução CREATE INDEX:

```
CREATE INDEX index_name  
ON table_name( column...,column)
```

- Para criar um índice no seu esquema, você deve ter o privilégio CREATE TABLE
- Para criar um índice em qualquer esquema, você precisa do privilégio CREATE ANY INDEX ou do privilégio CREATE TABLE para a tabela na qual está criando o índice. Valores nulos não são incluídos no índice

# Criando um Índice

- Por exemplo, para aumentar a velocidade do acesso da consulta à coluna REGION\_ID na tabela WF\_COUNTRIES:

```
CREATE INDEX wf_cont_reg_id_idx  
ON wf_countries(region_id);
```

# Quando Criar um Índice

- Um índice deve ser criado somente se:
  - A coluna contiver uma grande variedade de valores
  - A coluna contiver uma grande quantidade de valores nulos
  - Uma ou mais colunas forem frequentemente usadas juntas em uma cláusula WHERE ou condição de junção
  - A tabela for grande e esperar-se que a maioria das consultas recupere menos de 2 a 4% das linhas

# Quando Não Criar um Índice

- Quando se está decidindo se é necessário ou não criar um índice, nem sempre é melhor pecar pelo excesso
- Cada operação DML (INSERT, UPDATE, DELETE) que é executada em uma tabela com índices faz com que eles sejam atualizados
- Quanto mais índices estiverem associados à tabela, mais esforço será necessário para atualizar todos eles após a operação DML

# Quando Não Criar um Índice

- A criação de um índice não costuma valer a pena se:
  - A tabela for pequena
  - As colunas não forem frequentemente usadas como uma condição na consulta
  - Esperar-se que a maioria das consultas recupere mais de 2 a 4% das linhas na tabela
  - A tabela for atualizada com frequência
  - As colunas indexadas forem referenciadas como parte de uma expressão



# Índice Composto

- Um índice composto (também chamado de "concatenado") é aquele que você cria em várias colunas de uma tabela
- As colunas em um índice composto podem aparecer em qualquer ordem e não precisam ser adjacentes na tabela
- Os índices compostos podem acelerar a recuperação de dados para instruções SELECT em que a cláusula WHERE referencia todas as colunas ou a parte inicial delas no índice composto

```
CREATE INDEX emps_name_idx  
ON employees(first_name, last_name);
```

**ORACLE**

Academy

# Índice Composto

- Valores nulos não são incluídos no índice composto
- Para otimizar junções, você pode criar um índice na coluna FOREIGN KEY, o que acelera a pesquisa para relacionar as linhas à coluna PRIMARY KEY
- O otimizador não usa um índice se a cláusula WHERE contém a expressão IS NULL

# Confirmando Índices

- Confirme a existência dos índices a partir da view `USER_INDEXES` do dicionários de dados
- Você também pode consultar a view `USER_IND_COLUMNS` para verificar as colunas envolvidas em um índice
- A consulta mostrada no próximo slide é uma junção entre a tabela `USER_INDEXES` (nomes dos índices e sua exclusividade) e a tabela `USER_IND_COLUMNS` (nomes dos índices, nomes das tabelas e nomes das colunas)

# Confirmando Índices

```
SELECT DISTINCT ic.index_name, ic.column_name,  
                ic.column_position, id.uniqueness  
FROM user_indexes id, user_ind_columns ic  
WHERE id.table_name = ic.table_name  
AND ic.table_name = 'EMPLOYEES';
```

INDEX NAME	COLUMN NAME	COL POS	UNIQUENESS
EMP_EMAIL_UK	EMAIL	1	UNIQUE
EMP_EMP_ID_PK	EMPLOYEE_ID	1	UNIQUE
EMP_DEPARTMENT_IX	DEPARTMENT_ID	1	NONUNIQUE
EMP_JOB_IX	JOB_ID	1	NONUNIQUE
EMP_MANAGER_IX	MANAGER_ID	1	NONUNIQUE
EMP_NAME_IX	LAST_NAME	1	NONUNIQUE
EMP_NAME_IX	FIRST_NAME	2	NONUNIQUE

# Índices Baseados em Funções

- Um índice baseado em funções armazena os valores indexados e usa o índice com base em uma instrução SELECT para recuperar os dados
- Um índice baseado em funções é aquele que se baseia em expressões
- A expressão do índice é feita a partir de colunas de tabela, constantes, funções SQL e funções definidas pelo usuário

# Índices Baseados em Funções

- Os índices baseados em funções são úteis quando você não sabe se os dados foram armazenados com letras maiúsculas ou minúsculas no banco de dados
- Por exemplo, você pode criar um índice baseado em funções que possa ser usado com uma instrução SELECT que utiliza UPPER na cláusula WHERE
- O índice será usado nesta pesquisa

```
CREATE INDEX upper_last_name_idx  
ON employees (UPPER(last_name));
```

```
SELECT *  
FROM employees  
WHERE UPPER(last_name) = 'KING';
```

# Índices Baseados em Funções

- Os índices baseados em funções definidos com as palavras-chave `UPPER(nome_coluna)` ou `LOWER(nome_coluna)` permitem pesquisas sem distinção entre maiúsculas e minúsculas
- Se não souber como os sobrenomes dos funcionários foram incluídos no banco de dados, você ainda poderá usar o índice colocando letra maiúscula na instrução `SELECT`
- Quando uma consulta for modificada usando uma expressão na cláusula `WHERE`, o índice não a utilizará, a não ser que você crie um índice baseado em funções para corresponder à expressão



# Índices Baseados em Funções

- Por exemplo, a instrução abaixo permite pesquisas sem distinção entre maiúsculas e minúsculas usando o índice:

```
CREATE INDEX upper_last_name_idx  
ON employees (UPPER(last_name));
```

```
SELECT *  
FROM employees  
WHERE UPPER(last_name) LIKE 'KIN%';
```

# Índices Baseados em Funções

- Para garantir que o Servidor Oracle use o índice, em vez de fazer uma verificação completa da tabela, certifique-se de que o valor da função não seja nulo nas consultas subsequentes
- Por exemplo, é garantido que a seguinte instrução usará o índice, mas, sem a cláusula WHERE, o Servidor Oracle poderá fazer uma verificação completa da tabela:

```
SELECT *  
FROM employees  
WHERE UPPER (last_name) IS NOT NULL  
ORDER BY UPPER (last_name);
```

- O Servidor Oracle trata os índices com colunas marcadas com DESC como índices baseados em funções.

# Índices Baseados em Funções

- As colunas marcadas com DESC são classificadas em ordem decrescente
- Todos esses exemplos usam as funções UPPER e LOWER, mas vale observar que, embora sejam muito frequentemente usadas em Índices Baseados em Funções, o banco de dados Oracle não está limitado a elas
- Pode ser usada qualquer função incorporada válida do Oracle (por exemplo, TO\_CHAR)

# Índices Baseados em Funções

- Outro exemplo de Índices Baseados em Funções é mostrado aqui
- A tabela de funcionários é consultada para encontrar os funcionários contratados em 1987

```
SELECT first_name, last_name, hire_date  
FROM employees  
WHERE TO_CHAR(hire_date, 'yyyy') = '1987'
```

- Essa consulta resulta em uma Verificação Completa da Tabela, o que pode ser uma operação muito custosa se a tabela for grande
- Mesmo se a coluna hire\_date estiver indexada, o índice não será usado devido à expressão TO\_CHAR

# Índices Baseados em Funções

- Após criarmos o Índice Baseado em Funções a seguir, poderemos executar a mesma consulta, mas evitar a custosa Verificação Completa da Tabela

```
CREATE INDEX emp_hire_year_idx  
ON employees (TO_CHAR(hire_date, 'yyyy'));
```

```
SELECT first_name, last_name, hire_date  
FROM employees  
WHERE TO_CHAR(hire_date, 'yyyy') = '1987'
```

FIRST_NAME	LAST_NAME	HIRE_DATE
Steven	King	17-Jun-1987
Jennifer	Whalen	17-Sep-1987

- Agora, o Oracle pode usar o índice na coluna hire\_date

# Removendo um Índice

- Você não pode modificar índices
- Para mudar um índice, você deve eliminá-lo e recriá-lo em seguida
- Remova uma definição de índice do dicionário de dados executando a instrução DROP INDEX



# Removendo um Índice

- Para eliminar um índice, você deve ser o proprietário dele ou ter o privilégio DROP ANY INDEX
- Se você eliminar uma tabela, os índices e as constraints serão eliminados automaticamente, mas as views e sequências permanecerão

```
DROP INDEX upper_last_name_idx;
```

```
DROP INDEX emps_name_idx;
```

```
DROP INDEX emp_hire_year_idx;
```



# SYNONYM

- Em SQL, um sinônimo é uma palavra ou expressão aceita como substituta de outra palavra, exatamente como na linguagem comum
- Os sinônimos são usados para criar outro nome para o objeto e simplificar o acesso a ele
- Os sinônimos podem facilitar a referência a uma tabela pertencente a outro usuário e reduzir nomes de objetos longos
- Por exemplo, para se referir à tabela `amy_copy_employees` no esquema de um colega de classe, você pode iniciar o nome da tabela com o nome do usuário que a criou, seguido por um ponto e, em seguida, o nome da tabela: `USMA_SBHS_SQL01_S04. amy_copy_employees`

# SYNONYM

- A criação de um sinônimo elimina a necessidade de qualificar o nome do objeto com o esquema e lhe fornece um nome alternativo para uma tabela, view, sequência, procedure ou outro objeto
- Esse método pode ser muito útil no caso de nomes de objetos longos, como os das views
- O administrador do banco de dados pode criar um sinônimo público acessível para todos os usuários e conceder especificamente o privilégio CREATE PUBLIC SYNONYM a qualquer usuário, e esse usuário pode criar sinônimos públicos

# SYNONYM

- Sintaxe:

```
CREATE [PUBLIC] SYNONYM synonym  
FOR object;
```

- PUBLIC: cria um sinônimo acessível a todos os usuários
- sinônimo: é o nome do sinônimo a ser criado
- objeto: identifica o objeto para o qual o sinônimo é criado

```
CREATE SYNONYM amy_emps  
FOR amy_copy_employees;
```

# Diretrizes de SYNONYM

- Diretrizes:
  - O objeto não pode ser contido em um pacote
  - O nome de um sinônimo privado deve ser diferente de todos os outros objetos pertencentes ao mesmo usuário
- Para remover um sinônimo:

```
DROP [PUBLIC] SYNONYM name_of_synonym
```

```
DROP SYNONYM amy_emps;
```

# Confirmando um sinônimo

- A existência de sinônimos pode ser confirmada consultando a view `USER_SYNONYMS` do dicionário de dados

Nome da Coluna	Conteúdo
Nome_sinônimo	Nome do sinônimo.
Nome_tabela	Proprietário do objeto referenciado pelo sinônimo.
Proprietário_tabela	Nome do objeto referenciado pelo sinônimo.
Link_bd	Link de banco de dados referenciado em um sinônimo remoto.

# Terminologia

- Estes são os principais termos usados nesta lição:
  - Índice composto
  - Confirmando índice
  - CREATE PUBLIC SYNONYM
  - DROP INDEX
  - Índice baseado em funções
  - Índice não exclusivo
  - Sinônimo
  - Índice exclusivo

# Resumo

- Nesta lição, você deverá ter aprendido a:
  - Definir um índice e seu uso como um objeto de esquema
  - Nomear as condições que fazem um índice ser criado automaticamente
  - Criar e executar uma instrução CREATE INDEX e DROP INDEX
  - Criar índices baseados em funções
  - Criar sinônimos privados e públicos



# ORACLE

## Academy

