



Unidade 3

Bancos de Dados Relacionais e Linguagem SQL

3.5 Sequences, Índices e Sinônimos

Sequences

- Conhecemos dois tipos de objetos de banco de dados: TABLE e VIEW.
- Um outro objeto importante do Banco de dados relacional é a ***SEQUENCE*** que é usado para gerar automaticamente a sequencia de números exclusivos como os de chave primária, por exemplo.
- *Sequences* são geradas por uma rotina automática interna dos SGBDs.
- Sua utilização economiza tempo porque reduz a quantidade de códigos que você precisa gravar e controlar identificadores numéricos únicos.

Sequences

- Os números da *sequence* são armazenados e gerados de maneira independente às tabelas.
- Verifica-se duas formas de trabalhar com este objeto nas distribuições de SGBD's:

```
CREATE SEQUENCE chave_primaria_int_seq  
INCREMENT BY 1  
START WITH 1  
MAXVALUE 50000  
NOCACHE  
NOCYCLE;
```

Definindo a estrutura de *sequence*.

```
CREATE TABLE IF NOT EXISTS employees.access_menu (  
    menu_id INT(11) NOT NULL AUTO_INCREMENT,  
    list_order INT(11) NULL DEFAULT NULL,  
    name VARCHAR(15) NULL DEFAULT NULL,  
    parent_id INT(11) NULL DEFAULT NULL,  
    PRIMARY KEY (menu_id),
```

Apontando a coluna como AUTO_INCREMENT

Sequences

- O MySQL não implementa sequence em seu conceito tradicional porém, caso você queira emular esta condição siga este manual:
- <https://www.convert-in.com/docs/mysql/sequence.htm>
- Ou este código que também é de grande ajuda:
- <https://gist.github.com/arieljannai/acc72888750b8167faad>

Índices

- **Índices:** são objetos de esquema que podem acelerar a recuperação de linhas com o uso de um **ponteiro** e são criados de maneira explícita ou automática.
- Caso você não tenha um índice na coluna que está selecionando, uma verificação completa da tabela (*full scan*) e se houver o índice haverá o fornecimento de acesso rápido e direto às linhas em uma tabela.
- O propósito é a redução da necessidade de entrada/saída de disco, usando um caminho indexado (apontamento) para localizar os dados rapidamente.

Índices

- Índices são usados e mantidos automaticamente pelo SGBD e após a criação do índice, nenhuma atividade direta do usuário é necessária.
- São lógica e fisicamente independentes da tabela que indexam significando que podem ser criados ou eliminados a qualquer momento e não afetam as tabelas básicas ou outros índices.
- Eliminando uma tabela, os índices correspondentes também são eliminados.

Índices - Tipos de índices

- **Índice exclusivo:** é criado automaticamente quando há definição de uma coluna em uma tabela para ter uma *constraint PRIMARY KEY* ou *UNIQUE KEY*.
- **Índice não exclusivo:** é o índice que o usuário pode criar para acelerar o acesso às linhas por exemplo, para otimizar junções.
- Pode-se criar um índice na coluna FOREIGN KEY, o que acelera a pesquisa para relacionar as linhas à coluna PRIMARY KEY, melhorando o tempo de resposta à consulta.

Índices - Quando criar

- Recomenda-se a criação de um índice observando-se a variedade de valores da colunas e além disto se contiver uma grande quantidade de valores nulos.
- Outra estratégia interessante é se uma ou mais colunas forem frequentemente usadas juntas em uma cláusula WHERE ou condição de junção.
- Uma observação de performance importante é caso a tabela seja grande e a maioria das consultas recuperarem menos de 2 a 4% das linhas.

Índices - Quando criar

- Já ouvimos dizer que tudo que fazem em excesso não nos faz bem, certo? Esta frase se encaixa perfeitamente quando estamos decidindo se é necessário ou não criar um índice.
- Nas 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.
- Eles são organizados com conceitos de árvores balanceadas.

Índices - Quando não criar

- A criação de um índice não costuma valer a pena se:
- A tabela for pequena ou for atualizada com frequência;
- As colunas não forem frequentemente usadas como uma condição na consulta;
- Esperar-se recuperação entre 2 a 4% das linhas na tabela ou ainda se as colunas indexadas forem referenciadas como parte de uma expressão.

Índice composto

- Índice composto também chamado de concatenado, é criado em várias colunas e podem aparecer em qualquer ordem além de não precisarem ser adjacentes na tabela.
- Eles podem acelerar a recuperação de dados para instruções *SELECT* em que a cláusula *WHERE* reference todas as colunas ou a parte inicial delas no índice composto.
- Valores nulos não são incluídos no índice composto já que o otimizador não usa um índice se a cláusula *WHERE* contém a expressão *IS NULL*.

SYNONYM

- Sinônimos são palavras ou expressões que substituem outras palavras, exatamente como na linguagem comum e no caso dos SGBDs são usados para criar outro nome para o objeto e simplificar o acesso a ele.
- Podem facilitar a referência a uma tabela pertencente a outro usuário e reduzir nomes de objetos longos.
- Esse método pode ser muito útil no caso de nomes de objetos longos, como os das *views* e o DBA pode criar um sinônimo público acessível para todos os usuários.

Referências bibliográficas

ELMASRI, R., NAVATHE, S. B., Sistemas de Banco de Dados: Fundamentos e Aplicações. 3ª Ed., Editora LTC, 2002.

MANNINO, Michael V. Projeto, Desenvolvimento de Aplicações e Administração de Banco de Dados. 3ª. Ed. Porto Alegre. Bookman. 2008.