

## INDICES

O que são?

Objetos do banco de dados usados para referenciar diretamente determinado registro, sem a necessidade de percorrer toda a tabela para isso (FULL TABLE SCAN).

A melhor maneira de entender esse processo é fazendo uma analogia com um livro qualquer. Pois bem, imagine um livro com 830 páginas e divididos em 18 capítulos que por sua vez são divididos em vários assuntos.

Imagine agora, que você deva acessar o capítulo 13 – Hints e buscar o assunto "Especificando múltiplos Hints", então você pensa é fácil! Basta procurar no índice do livro e acessar a página indicada. Pois bem, essa também é a maneira do banco trabalhar, ou seja, quando um registro é solicitado, o banco procura no índice da tabela e acessa o registro diretamente.

Ok, mas e se não existisse índice como seria?

No nosso exemplo, para achar a página do livro que contém o assunto desejado teríamos que folhear página por página até encontrá-lo, fazendo assim uma leitura completa do livro. Dá para imaginar o esforço necessário para executar essa tarefa. Podemos projetar essa mesma idéia para o modo como o banco trabalha e termos uma noção do quanto ela é dispendiosa, reduzindo a sua performance.

Por que não usar Índices em todas as tabelas?

Em primeiro lugar o índice está relacionado a uma (ou mais) coluna(s) e não a toda tabela. E depois, uma tabela do banco de dados não é estática como o livro, ou seja, estamos sempre inserindo novos dados, alterando dados existentes e também apagando-os, isso gera a necessidade de reorganizar o índice a cada alteração, acarretando um custo para mantê-lo atualizado e sincronizado. É fácil perceber que é muito pior ter um índice errado do que não ter nenhum, pois seríamos levados a informações erradas.

Sobre os índices existe um consenso de quando NÃO devemos criá-los.

- As colunas não são usadas frequentemente como condição nas consultas
- A tabela for pequena ou se os resultados das consultas forem maiores que 5-10% dos registros.
- A tabela for atualizada com frequência

Para criar um índice deve-se usar o comando:

CREATE INDEX <nome\_indice> ON <nome\_tabela(coluna (s))>;

Exemplo:

1 – vamos criar as tabelas abaixo:

SETOR	
COD_SETOR	NUMBER (4)
NOME_SETOR	VARCHAR2 (30)

COLABORADOR	
COD_COLABORADOR	NUMBER
NOME_COLABORADOR	VARCHAR2 (50)
COD_SETOR	NUMBER

2 - Insira os registros na tabela SETOR:

```
insert into setor values(1,'Diretoria');  
insert into setor values(2,'Recursos Humanos');  
insert into setor values(3,'Projetos');  
insert into setor values(4,'Produção');  
insert into setor values(5,'Manutenção');
```

e na tabela COLABORADOR:

```
insert into setor values(1,'Monica',1);  
insert into setor values(2,'Cebolinha',1);  
insert into setor values(3,'Magali',2);  
insert into setor values(4,'Cascão',2);  
insert into setor values(5,'Bidu',3);  
insert into setor values(1,'Teste',1);
```

3 – Agora a veja o plano de execução do comando abaixo:

```
select * from colaborador;
```

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			2
TABLE ACCESS	COLABORADOR	FULL	2

Perceba que o ORACLE optou por fazer acesso FULL na tabela. Isso por dois motivos:

Não há índice para a tabela

A quantidade de registros retornados é maior que 10% do total de linhas

4 – Vamos criar o índice IDX\_COD\_COLABORADOR na tabela COLABORADOR.

```
create index idx_cod_colaborador on colaborador(cod_colaborador);
```

Se você refizer o plano de execução do select anterior perceberá que o acesso também será por FULL TABLE SCAN.

5 – Agora vamos executar o plano de execução para o comando:

```
select nome_setor, nome_colaborador from colaborador x, setor s
where x.cod_setor = s.cod_setor and cod_colaborador = 2;
```

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			5
HASH JOIN			5
Acessar Predicados X.COD_SETOR			
TABLE ACCESS	COLABORADOR	FULL	2
Filtrar Predicados COD_CC			
TABLE ACCESS	SETOR	FULL	2

Note que mesmo com o índice criado o acesso é FULL nas duas tabelas.

6 – Crie o índice IDX\_COD\_SETOR na tabela SETOR

```
create index idx_cod_setor on setor(cod_setor);
```

7 – Ao refazer o plano de execução do select anterior perceba que o plano de execução mudou pois o índice ficou mais seletivo.

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			3
NESTED LOOPS			3
NESTED LOOPS			3
TABLE ACCESS	COLABORADOR	FULL	2
Filtrar Predicados COD_COLABORADOR=2			
INDEX	IDX_COD_SETOR	RANGE SCAN	0
Acessar Predicados X.COD_SETOR=S.COD_SETOR			
TABLE ACCESS	SETOR	BY INDEX ROWID	1

O plano mostra o uso do índice da tabela SETOR para trazer os registros solicitados.