

Seleção de dados no SQL da Oracle: Indices e Subselects

Paulino Ng

2020-11-11

Indices

- ▶ Indices aceleram buscas.
- ▶ Como nos livros, o índice remissivo permite a procura rápida por algum assunto específico. Nos BDs, os índices permitem acelerar uma busca por um valor de uma coluna.
- ▶ Deve-se criar índices para as colunas que são mais usadas para buscas. Na maioria dos SGBDs, ao declarar uma chave primária, esta automaticamente é indexada.
- ▶ Pode-se criar índices para uma coluna, ou uma combinação delas. Se for criar para uma combinação, coloque na frente a coluna mais usada.

Exemplos de Indices

```
create index employee_dept_no_fk_idx  
on employees (deptno)
```

- ▶ employee_dept_no_fk_idx é o nome da chave
- ▶ employees é o nome da tabela
- ▶ deptno é o nome da coluna

Exemplo2

```
create unique index employee_ename_idx  
on employees (name)
```

- ▶ o índice `employee_ename_idx` é o nome dos empregados, ele tem de ser único. Se houverem 2, ou mais, empregados com o mesmo nome, será necessário colocar algo no nome para diferenciá-los.

Como, O que, Por que e Quando usar Índices

- ▶ Já vimos a sintaxe: `CREATE INDEX NOME_DO_INDICE ON NOME_DA_TABELA (COLUNAS)`
- ▶ A analogia para índices em BDs são os índices remissivos que aparecem no final dos livros. Mas em BDs, só o desenvolvedor precisa saber deles. Não há necessidade para os usuários da BD saberem da existência deles. Não veremos como SGBDs implementam os índices.
- ▶ A razão para definí-los é aumentar a velocidade das consultas. O preço é que modificações e inserções de dados tomam mais tempo.
- ▶ As colunas mais acessadas de uma tabela devem ser indexadas.
- ▶ As colunas usadas em operações de JOIN devem ser indexadas.
- ▶ Colunas usadas para buscas por intervalos de valores devem ser indexadas.

Não use índices para:

- ▶ Colunas que raramente são referenciadas.
- ▶ Colunas que têm poucos valores possíveis (sim, não, talvez).
- ▶ Tabelas pequenas.

Subqueries

- ▶ Vamos ver um tutorial

Sintaxe

```
SELECT [DISTINCT] lista_seleção
FROM lista_tabelas
WHERE
    {expressão { [NOT] IN | operador_comparação [ANY | ALL]
    | [NOT] EXISTS}
(SELECT [DISTINCT] lista_seleção_subquery
    FROM lista_tabelas_sub
    WHERE condições)
[GROUP BY lista_agrupamentos
[HAVING condições]]
[ORDER BY lista_ordem]
```


Subqueries em WHERE

- ▶ Subselects podem retornar zero ou mais itens (operações de IN, ANY ou ALL)
- ▶ Subselects podem retornar um único valor (usado em alguma operação de comparação)
- ▶ Subselects podem ser usados para teste de existência de resultado (EXISTS, ou NOT EXISTS)

Subselects podem ser usados no FROM

- ▶ No lugar de fornecer uma tabela, é possível fornecer um subselect cujo resultado é a tabela onde será feita consulta.

SELECT ...