Projeto de Bases de Dados

Parte 4

Grupo 19 – Turno BD225179L09

Prof. Gabriel Pestana

81082 – Nuno Gonçalves (15 horas)

81205 – Alice Dourado (15 horas)

81500 – Rodrigo Rato (15 horas)

1. **Índices**

Query #1 – A)

Índices na tabela **arrenda**:

- Índice do tipo **HASH** para a coluna morada;

- Índice do tipo **HASH** para a coluna codigo;

- Índice do tipo **BTREE** para a coluna nif;

Índices na tabela **fiscaliza**:

- Índice do tipo **HASH** para a coluna morada;

- Índice do tipo **HASH** para a coluna codigo;

Os índices hash são colocados devido à comparação a.morada = f.morada and a.codigo = f.código , pois estes tipo de índices são mais eficientes para operações de verificação de igualdade entre valores .

O índice BTREE sobre nif na tabela arrenda deve-se ao group by a.nif, que fica mais rápido ao percorrer a tabela por ordem, algo que é possível quando se tem este tipo de índice mas que é impossível de fazer num índice em hash onde determinar uma ordem de iteração da hash table é impossível.

b)

Para criar o índice BTREE do arrenda sobre a coluna nif usou-se a seguinte instrução:

**create index** nif\_idx **on** arrenda(nif) **using btree**;

Os índices HASH seriam criados com as seguintes instruções:

**create index** morada\_idx **on** arrenda(morada) **using hash**;

**create index** codigo\_idx **on** arrenda(codigo) **using hash;**

**create index** morada\_idx **on** fiscaliza(morada) **using hash;**

**create index** codigo\_idx **on** fiscaliza(codigo) **using hash;**

No entanto devido a limitações do software MySQL nao é possível criar índices hash uma vez que o storage engine usado nestas tabelas não suporta este tipo de índices.

Também não é necessário criar o índice nif\_idx pois está associado a uma primary key da tabela logo é criado um índice automaticamente.

O plano de execução desta query foi o seguinte:





É possível observar que o índice btree que críamos não foi usado neste caso porque no resultado obtido há apenas um nif distinto com o conteúdo que está nas tabelas, mas possivelmente com um conteúdo diferente seria usado se fosse obtida uma maior quantidade de nifs a partir da query

A primeira linha mostra que é usado um índice chamado morada que foi criado automaticamente na tabela fiscaliza, se fosse possível criar os índices hash que indicámos esses seriam usados em vez deste.

Query #2 – A)

Índice na tabela **estado**:

- Índice do tipo **HASH** para a coluna estado;

Índice na tabela **posto**:

- Índice do tipo **HASH** para as colunas morada, codigo\_espaco;

O índice do estado é criado porque, na query, é usada no where a condição “e.estado = ’aceite’” pelo que um índice do tipo hash será a melhor escolha para este caso pois este tipo de índice é mais eficiente para operações de verificação de igualdade entre valores.

O índice da tabela posto é devido ao facto de, na cláusula where da query efetuada, é usada a condição “(p.morada, p.codigo\_espaco) not in (…)”, o que implica que o valor irá ser comparado com todos os valores da tabela em segundo argumento e como isto é uma operação de verificação de igualdade o melhor índice para a situação é do tipo hash.

b)

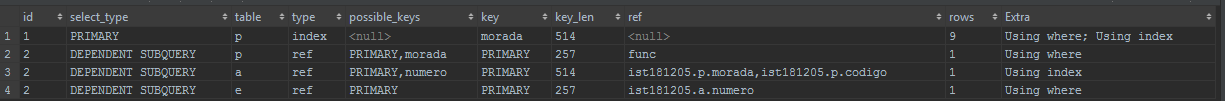
Para criar se criarem os índices usaram-se a seguintes instruções:

**create index** estado\_idx **on** estado(estado) **using hash**;

**create index** posto\_idx **on** posto(morada, codigo\_espaco) **using hash**;

Não é possível criar estes índices em MySQL devido às limitações mencionadas anteriormente na query 1

O plano de execução desta query foi o seguinte:



Para alem dos índices das primary keys o outro índice usado é o ‘morada’ que foi criado automaticamente na tabela posto, mas se fosse possível criar/usar índices hash o nosso índice iria ser escolhido em detrimento do índice predefinido em btree.