Sprint 1 – Banco de Dados

Normalização e Clusterização

Bárbara

Carlos

Renato

Saulo

Normalização

O objetivo principal é resolver problemas de atualização de bases de dados, **tornando-o mais íntegro, sem redundâncias e inconsistências**. É um processo no qual são eliminados esquemas de relações (tabelas) não satisfatórios, decompondo-os, através da separação de seus atributos em esquemas de relações menos complexas, mas que satisfaçam as propriedades desejadas.

Vantagens e Desvantagens

A análise da documentação de uma modelagem normalizada é muito mais clara, pois conhecendo os padrões, a compreensão é facilitada. Uma desvantagem seria o aumento do número de tabelas. Bancos devidamente construídos, ou seja, na terceira forma normal, apresentam um número maior de tabelas em comparação aos bancos não normalizados. Assim, as consultas em bancos com mais tabelas requerem uma complexidade maior na elaboração do SQL, fazendo necessário o uso dos joins e cláusulas para elaborarmos a consulta adequadamente. Devido a isso, a aplicação das regras de normalização de dados é altamente recomendada, pois os ganhos são consideravelmente relevantes.

Primeira Forma Normal – 1FN

Uma relação se encontra na 1FN se todos os domínios de atributos possuem apenas valores atômicos (simples e indivisíveis), e que os valores de cada atributo na tupla seja um valor simples, ou seja, a 1FN não permite que haja em um atributo um conjunto de valores ou uma tupla de valores, em outras palavras não são aceitas relações dentro de relações. Assim sendo todos os atributos compostos e multivalorados devem ser divididos em atributos atômicos.

Exemplo de 1FN: Livros

Tabela de Livros: Estrutura original

IdLivro	Título	Assunto	Autor1	Autor2	Autor3
21237	Os Sertões	Ficção	E. Cunha		
33455	Eletricidade básica	Física	A. Silva	B. Santos	
12312	Atlas do Brasil	Geografia	IBGE		

Estrutura normalizada na 1FN:

Tabela de Livros

IdLivro	Título	Assunto
21237	Os Sertões	Ficção
33455	Eletricidade básica	Física
12312	Atlas do Brasil	Geografia

Tabela Autores_Livros

IdLivro	Autor
21237	E. Cunha
33455	A. Silva
33455	B. Santos
12312	IBGE

Segunda Forma Normal – 2FN

Uma relação encontra-se na segunda forma normal quando estiver na primeira forma normal e todos os atributos que não participam da chave primária são dependentes desta. Assim devemos verificar todos os atributos que não são dependentes da chave primária e retirá-los da relação (tabela), dando origem a uma nova relação, que conterá esse atributo como não chave. Desta maneira, na segunda forma normal evita inconsistências devido a duplicidades.

Tabela de Empregado_Projeto: Estrutura original

Num_emp	Num_proj	Horas	Nome_emp	Nome_proj	Local_proj
00001	001	8	Maria	Versão Evolutiva 3.22	João Monlevade
00002	001	18	José	Versão Evolutiva 3.22	João Monlevade
00003	002	12	Samara	Versão Corretiva 3.21	Belo Horizonte

Estrutura normalizada - 2FN

Tabela: projetos

Num_proj	Nome_proj	Local_proj
001	Versão Evolutiva 3.22	João Monlevade
002	Versão Corretiva 3.21	Belo Horizonte

Tabela: Empregado_projeto

Num_emp	Num_proj	Horas
00001	001	8
00002	001	18
00003	002	12

Tabela: Empregado

Num_emp	Nome_emp
00001	Maria
00002	José
00003	Samara

Terceira Forma Normal – 3FN

Para estar na terceira forma normal a tabela não pode ter atributos não-chave se referindo a outros atributos não-chave. Assim devemos verificar se existe um atributo que não depende diretamente da chave, retirá-lo criando uma nova relação que conterá esse grupo de atributos, e definir com a chave, os atributos dos quais esse grupo depende diretamente.

O processo de normalização deve ser aplicado em uma relação por vez, pois durante o processo de normalização vamos obtendo quebras, e, por conseguinte, novas relações. No momento em que o sistema estiver satisfatório, do ponto de vista do analista, este processo iterativo é interrompido.

Exemplo de 3FN: Empregados trabalhando em departamentos

Tabela de Empregado_Depto: Estrutura original

Num_emp	Nome	Data_nasc	Num_Depto	Nome_Depto	Emp_Ger_Depto
00001	Maria	06/03/1977	001	Homologação	018
00002	José	27/05/1973	002	Homologação	018
00003	Samara	24/08/1984	003	Desenvolvimento	005

Tabela: Empregado_Depto

Num_emp	Nome	Data_nasc	Num_Depto
00001	Maria	06/03/1977	001
00002	José	27/05/1973	002
00003	Samara	24/08/1984	003

Tabela: Departamento

Num_Depto	Nome_Depto	Emp_Ger_Depto
001	Homologação	018
002	Homologação	018
003	Desenvolvimento	005

Clusterizacao

Índice

Um índice é uma estrutura em disco associada a uma tabela ou exibição, que agiliza a recuperação das linhas de uma tabela ou exibição. Um índice contém chaves criadas de uma ou mais colunas da tabela ou exibição. Essas chaves são armazenadas em uma estrutura que habilita o SQL Server a localizar a linha ou as linhas associadas aos valores de chave de forma rápida e eficaz.

Tanto os índices clusterizados quanto os não clusterizados podem ser exclusivos. Isso significa que duas linhas não podem ter o mesmo valor que a chave de índice. Caso contrário, o índice não será exclusivo e várias linhas poderão compartilhar o mesmo valor de chave.

Os índices são mantidos automaticamente para uma tabela ou exibição sempre que os dados da tabela são modificados.

Os índices são criados automaticamente quando as restrições PRIMARY KEY e UNIQUE são definidas em colunas de tabelas. Por exemplo, ao criar uma tabela e identificar determinada coluna como a chave primária, o Mecanismo de Banco de Dados cria automaticamente uma restrição PRIMARY KEY e o índice nessa coluna.

Aplicação

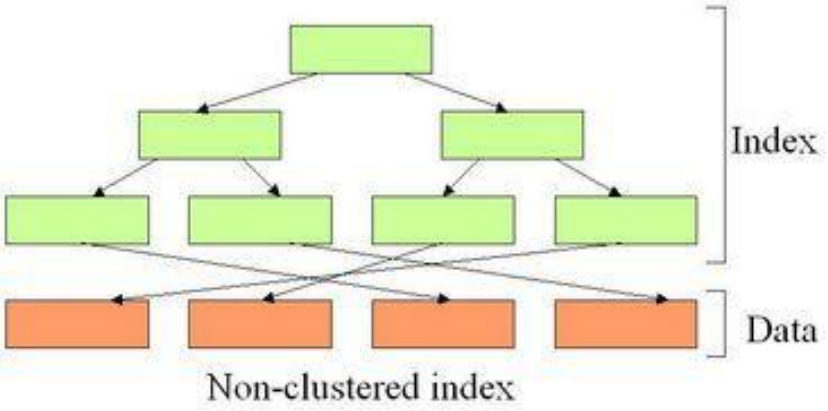
Índices bem projetados podem reduzir as operações de E/S de disco e consumir menos recursos de sistema, aprimorando o desempenho das consultas. Os índices podem ser úteis para uma série de consultas que contêm instruções SELECT, UPDATE, DELETE ou MERGE.

O otimizador de consulta avalia cada método disponível para recuperar os dados e seleciona o mais eficaz. O método pode ser uma verificação de tabela ou verificação de um ou mais índices, se houver.

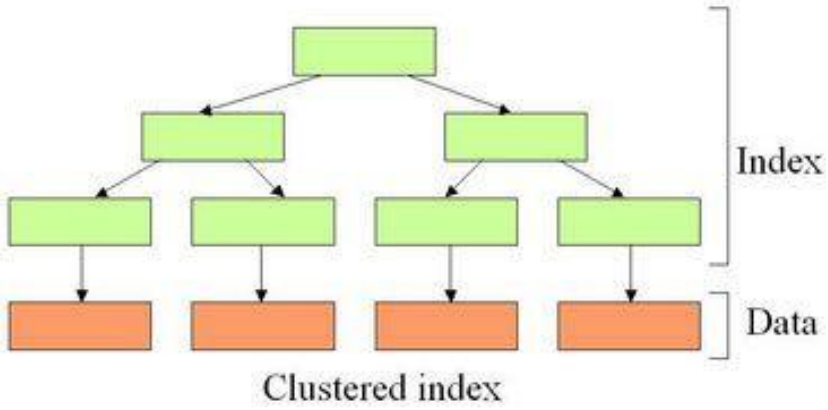
Ao executar uma verificação de tabela, o otimizador de consulta lê todas as linhas da tabela e extrai as linhas que atendem os critérios da consulta. Uma verificação de tabela gera várias operações de E/S de disco e pode utilizar muitos recursos. No entanto, a verificação de tabela poderá ser o método mais eficaz se, por exemplo, o conjunto de resultados da consulta contiver um alto percentual de linhas da tabela.

Quando o otimizador de consulta utiliza um índice, ele pesquisa as colunas de chave do índice, encontra o local de armazenamento das linhas necessárias à consulta e extrai as linhas que correspondem àquele local. Em geral, fazer pesquisas no índice é muito mais rápido do que na tabela, porque diferentemente da tabela, o índice contém, com frequência, poucas colunas por linha e as linhas ficam na ordem de classificação.

Diferenças



Non-clustered index



Clustered index

Vantagens e Desvantagens

Uma das grandes vantagens na utilização de índices é o ganho de performance na busca de dados.

A desvantagem é quando se utiliza índices clusterizados e os dados são atualizados com certa frequência, pois isto geraria um alto custo de otimização e escrita.

Como criar índices

Sintaxe para criação de um índice não clusterizado

```
CREATE NONCLUSTERED INDEX NOME_INDEX
ON     NOME_TABELA (NOME_COLUNA)
```

Sintaxe para criação de um índice clusterizado

```
CREATE CLUSTERED INDEX NOME_INDEX
ON     NOME_TABELA (NOME_COLUNA)
```

Obs.: Não é possível criar um índice, seja clusterizado ou não, caso já exista outro índice com o mesmo nome.

Bibliografia

http://aprendaplsql.com/modelagem-de-dados/normalizacao-banco-de-dados/

http://www.dsc.ufcg.edu.br/~pet/jornal/maio2011/materias/recapitulando.html

https://www.devmedia.com.br/indices-clusterizados-e-nao-clusterizados-no-sql-server/30288

https://docs.microsoft.com/pt-br/sql/relational-databases/indexes/clustered-and-nonclustered-indexes-described?view=sql-server-2017

https://www.devmedia.com.br/projetando-e-criando-indices-clusterizados/6964

https://docs.microsoft.com/pt-br/sql/relational-databases/indexes/create-clustered-indexes?view=sql-server-2017

Repositório GitHub

https://github.com/Saulomsantos/indices_clusterizados_nao_clusterizados_normalizacao.git

Quadro Trello

https://trello.com/b/CgJSvPmu/%C3%ADndices-clusterizados-n%C3%A3o-clusterizados-normaliza%C3%A7%C3%A3o

Arquivos Google Drive

https://drive.google.com/drive/folders/1XWQwZ9euZ8mHI-9Hl3boQ7NXLMQLjq0P?usp=sharing