
sr.nimbus

SQL Server 2008

SQL07 – Recursos de Otimização para o desenvolvedor

Módulo 01 – Otimização, Tabelas e Consultas

Agenda

- **Processo de otimização**
  - Elementos
  - Processo
- **Consultas**
  - Ferramentas
  - Cursor-based vs. Set-based
- **Tabelas**
  - Tipos de dados
  - Estrutura
  - Modelagem
- **Conclusão**

Processo de otimização

Um pouco de processo não faz mal a ninguém…

Processo de otimização

- **Interação com o SQL Server se dá através de vários perfis**
  - Desenvolvedor
  - DBA
  - Usuário
- **Independentemente da fonte, uma coisa sempre é consenso:**

TODO MUNDO QUER UM BOM DESEMPENHO!!

Processo de otimização

- **Porém, existem várias peças nesse quebra-cabeça**
  - Sistema Operacional
  - Armazenamento (Storage)
  - Processamento da consulta
  - Indexação
  - Otimização
  - Caching (dados/procedimento)
  - Arquitetura dos dados/solução
- **Quando DBA e DEV trabalham juntos, fica mais simples resolver os problemas e evitá-los.**

Processo de otimização

- Aí chega a notícia...

  “A APLICAÇÃO ESTÁ LENTA!!”

  (Ou você deseja fazer um tuning pró-ativo)

- Por onde começar?
- Onde está a causa raiz do problema? Ou qual procedimento eu devo otimizar?
- Cada situação irá ditar as ferramentas e procedimento a ser adotado.

Processo de otimização

- Aí chega a notícia...

  “A APLICAÇÃO ESTÁ LENTA!!”

  (Ou você deseja fazer um tuning pró-ativo)

- Por onde começar?
- Onde está a causa raiz do problema? Ou qual procedimento eu devo otimizar?
- Cada situação irá ditar as ferramentas e procedimento a ser adotado.

Processo de otimização

- **Tuning é mais uma arte do que uma ciência”**
  - O que não quer dizer que não existe muita ciência por detrás das otimizações
  - Mas com o tempo você começa a ter alguns sentimentos em relação ao que está acontecendo.
- **Mas para ser efetivo é necessário um entendimento de como o SQL Server trabalha, senão a coisa vira uma série de palpites consecutivos. (Onde ocasionalmente um pode acabar resultando em um acerto...)**

Processo de otimização

- (0) Identifique o seu objetivo primário
- (1) Isole o problema / consulta
- (2) Reproduza a situação em ambiente controlado
- (3) Tenha um baseline para comparação
- (4) Crie uma hipótese para a situação atual
- (5) Proponha uma única solução
- (6) Implemente a solução e verifique resultado
- (7) Se deu certo, documente. Senão, volte para passo 3.

Consultas

Uma visão geral e elementos de otimização

Consultas

- **Como escrever consultas T-SQL está fora do escopo desse treinamento.**
  - SQL01 (T-SQL Querying) e SQL02 (T-SQL Programming)
- **Consultas são otimizadas e executadas pela engine relacional do SQL Server.**
  - Mais detalhes desse processo, ao longo do curso.
- **Uma consulta otimizada hoje pode gerar um plano diferente amanhã.**
  - Isso é normal!

Consultas

- **Toda consulta (manipulação de dados) possui um plano de execução.**
- **Primeiro é necessário entender como ler um plano de execução.**
- **SSMS: “Include Actual Execution Plan”**
  - SQL Server 2005 em diante: XML
  - Existem outras abordagens.
- **SET STATISTICS TIME**
- **SET STATISTICS IO**

Cursor based vs. Set based

- **Evolução do programador T-SQL**
  - Procedural: quando os programadores começam a trabalhar com o SQL Server, trazendo experiência de alguma linguagem de programação.
  - Ficando lúcido: a pessoa entende um pouco mais de programação orientada a conjuntos. Acreditam que tabelas temporárias, cursores e execução dinâmica são o mal da humanidade. ☺
  - Maturidade: já possui um conhecimento dos detalhes da arquitetura em que trabalha. Avalia prós e contras de cada abordagem, dando preferência a operações de conjuntos, porém sabendo reconhecer quando cursores, tabelas temporárias e execução dinâmicas são importantes.

Tabelas

A estrutura básica dos nossos bancos de dados

Tabelas

- Tabelas são o coração do SQL Server e do modelo relacional em geral, pois é onde o dado é armazenado.
- Cada instância de um dado na tabela representa uma entidade simples ou registro (formalmente chamado de tupla)
- A maioria das tabelas serão relacionadas entre si. Por exemplo: A tabela Cliente possuí um identificador único IDCliente que é usado como chave estrangeira no relacionamento com a tabela Pedido.

Tabelas

- As tabelas devem ser modeladas de acordo com a teoria de banco de dados relacionais, respeitando as formas normais.
- Não existe modelagem certa ou errada, existem aquelas modelagens que vão consumir maior espaço, dar mais trabalho de manutenção, acelerar as pesquisas, etc.
- Quantas tabelas grandes existe no seu ambiente?
- E as tabelas de lookup, quantas são?
- Múltiplos bancos de dados para subsistemas?

Tabelas

```
CREATE TABLE [ database_name . [ schema_name ] . |
    schema_name . ] table_name ( {
    <column_definition> |
    <computed_column_definition>
    |<column_set_definition> }  [ <table_constraint> ] [
    ,...n ] ) [ ON { partition_scheme_name (
    partition_column_name ) | filegroup | "default" } ] [
    { TEXTIMAGE_ON { filegroup | "default" } ] [
    FILESTREAM_ON { partition_scheme_name |
    filegroup | "default" } ] [ WITH ( <table_option> [
    ,...n ] ) ] [ ; ]
```

- Veja definição completa no BOL

Tipos de dados

- **Numéricos exatos**
  - Bit: 1 bit
  - Tinyint: 1 byte (0..255)
  - Smallint: 2 bytes (-215 .. 215-1)
  - Int: 4 bytes (-233 .. 233-1)
  - Bigint: 8 bytes (-263 .. 263-1)
  - Smallmoney: 4 bytes (214.748,3648)
  - Money: 8 bytes (922.337.203.685.477,5807)

Tipos de dados

- **Numéricos exatos**
  - Decimal
    - Precisão 1 a 9: 5 bytes
    - Precisão 10 a 19: 9 bytes
    - Precisão 20 a 28: 13 bytes
    - Precisão 29 a 38: 17 bytes
- **Vardecimal foi introduzido com o SQL Server 2005 SP2.**
- **Numéricos aproximados**
  - Float (n) = 24 ou 53 / 4 ou 8 bytes
  - Real = 4 bytes (float(24))

Tipos de dados

- **Data e hora**
  - Datetime: 8 bytes (.003333s)
  - Smalldatetime: 4 bytes (1m)
  - Datetime2: 6 a 8 bytes (100n)
  - Date: 3 bytes (1d)
  - Time: 3 a 5 bytes (100n)
  - Datetimeoffset: 8 a 10 bytes (100n)

Tipos de dados

- **Caractere**
  - Char(n)
  - Varchar(n)
  - Nchar(n)
  - Nvarchar(n)
  - Ntext – Deprecated, usar NVARCHAR(MAX)
- **Imagem**
  - Binary
  - VarBinary
  - Image – Deprecated, usar VARBINARY(MAX)

Tipos de dados

- **Outros tipos**
  - Cursor (não utilizado com tabela, mas um tipo que aponta para um cursor).
  - Rowversion/Timestamp – valor único incremental para o BD.
  - Sql_variant – permite armazenar praticamente qualquer tipo (até 8K).
  - Tabela (TVP no SQL Server 2008)
  - XML (até 2GB)
  - Uniqueidentifier – 16 bytes

NULL

- **Adiciona complexidade para a engine que mantém um mapa de bits para colunas nulas.**
- **Aplicação deve saber como tratar nulos.**
- **Idealmente todas as colunas seriam NOT NULL!**
  - Se possível, podemos trabalhar com o DEFAULT.
- **Colunas esparsas podem ajudar em cenários onde existem múltiplas colunas com null.**

- **E vocês, já tiveram problema com nulos? Como foi e como foi resolvido?**

Tipos definidos pelos usuário

- **Maneira conveniente de garantir a consistência entre os desenvolvedores e DBAs dentro do mesmo domínio de negócio.**
- **Ex.: definir tipo CPF ou Telefone.**
  - CHAR(12), DECIMAL(11,0), DD e telefone separados?
- **Sintaxe: CREATE TYPE <nome> FROM ....**

- **Vale a pena usar UDTs?**
  - Quem já possui experiência com eles?

Identity

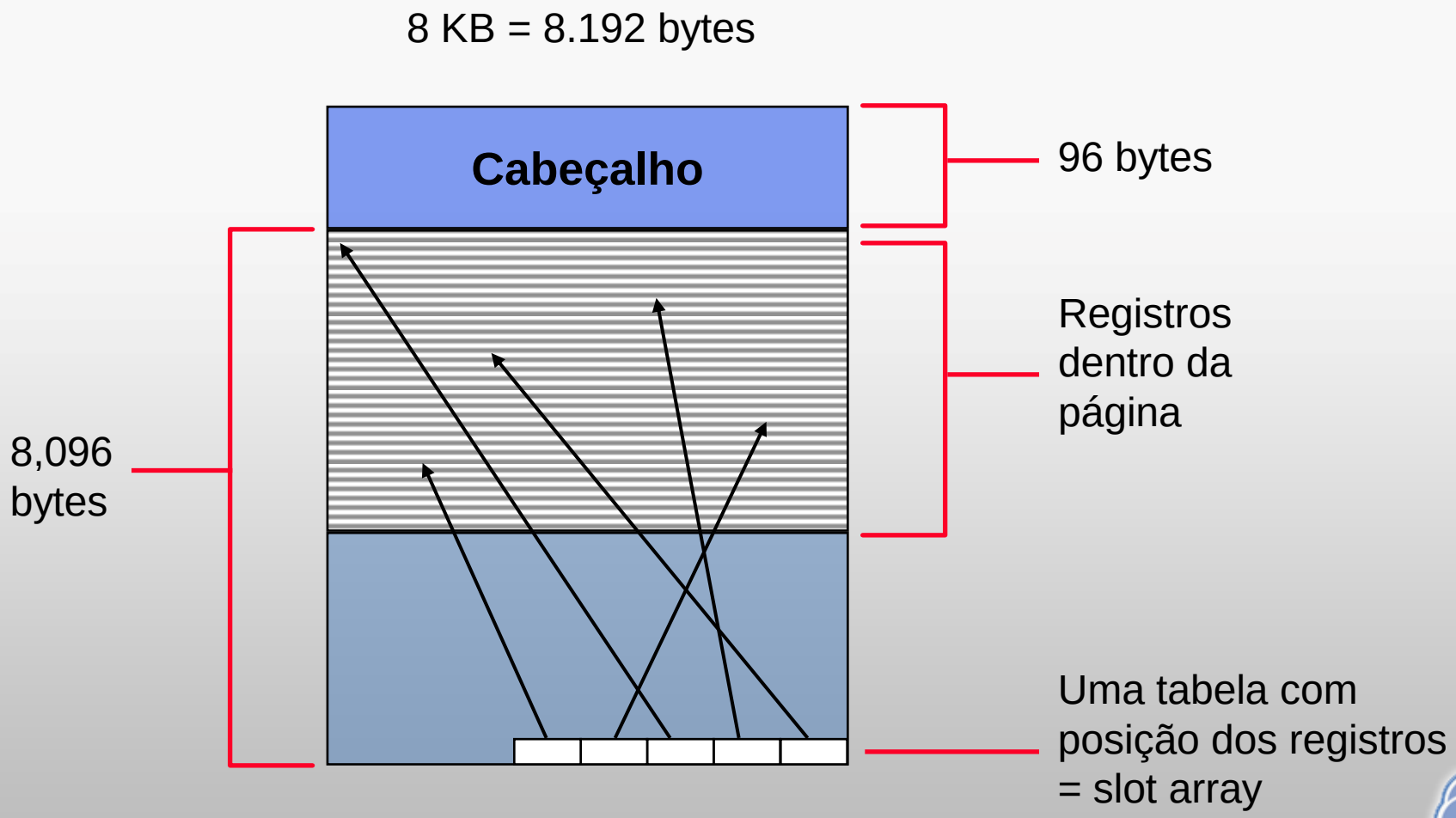
- **Também conhecido como auto-numeração.**
- **Gera valores sequenciais para uma coluna específica na tabela.**
  - Excelente para surrogate keys!
  - Porque utilizá-lo como índice cluster?
- **Não garante unicidade dos registros e nem que não haverá falha de numeração.**
- **Não necessita ser uma numeração crescente. Pode-se definir o valor inicial e o incremento.**
- **IDENT_SEED e IDENT_INCR**
- **Internamente utiliza um spinlock para gerar próximo valor.**

Modelando sua tabela

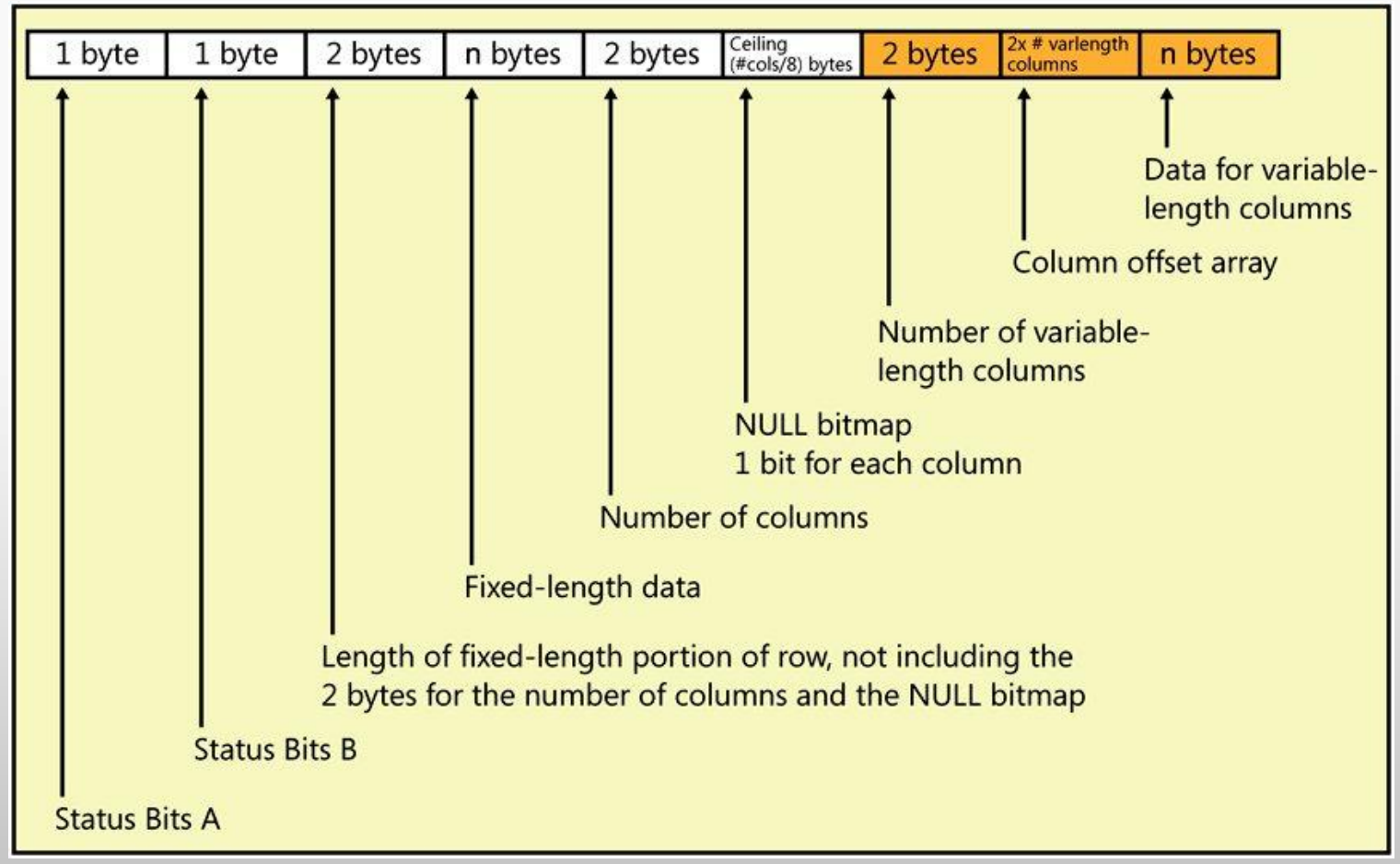
- **Quanto mais registros em uma página, maior a eficiência de IO e acertos em cache!**
- **Ao escolher um o tipo de dados e tamanho das suas colunas, não gaste muito espaço e não seja pão-duro!**
- **A escolha apropriada de um tipo é importante também durante a indexação.**
  - Indexação será discutida no próximo módulo

Armazenamento interno

- **Toda página contém cabeçalho, corpo e slot array.**



Armazenamento interno

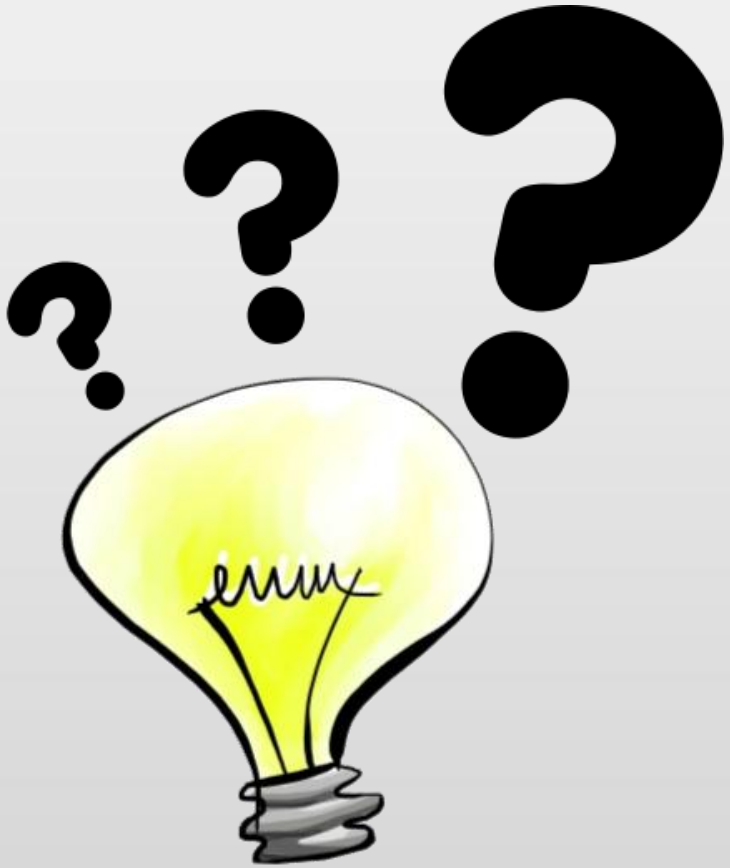


Constraints (Restrições)

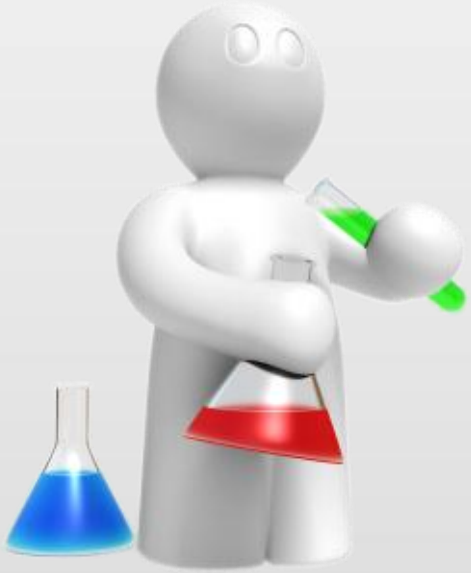
- **Integridade da entidade, de domínio e referencial.**
- **Tipos de constraints:**
  - PRIMARY KEY
  - UNIQUE
  - FOREIGN KEY
  - CHECK
  - DEFAULT
- **Unicidade é garantida através de índices.**
- **Importantes para o query optimizer!**

Conclusão

- **É importante seguir um processo bem definido durante a otimização de uma consulta.**
  - Entender o que resultou em o maior ganho é vital para fazer escolhas acertadas do que implementar.
- **Usualmente abordagens baseadas em conjuntos são mais eficientes que utilização de cursores.**
  - Mas nem sempre, então faça os testes.
- **Escolha sabiamente os tipos de dados da sua tabela, para evitar problemas no futuro, tanto no armazenamento como no crescimento.**



Dúvidas



Laboratório

Módulo 01

Recursos

- Database Design for Mere Mortals - Michael Hernandez
- Refactoring Databases – Scott Ambler
- Beginning Database Design: From Novice to Professional - Clare Churcher