Visões Materializadas

Cristina Dutra de Aguiar Ciferri
Bruno Tomazela
Renata Miwa Tsuruda

Agosto/2010

Agenda

- Visão Geral
  - http://download.oracle.com/docs/cd/B28359_01/server.111/b28318/schema.htm#CNCPT411
- Sintaxe
- Exemplo
- Exercícios

Agenda

- Visão Geral
  - http://download.oracle.com/docs/cd/B28359_01/server.111/b28318/schema.htm#CNCPT411
- Sintaxe
- Exemplo
- Exercícios

Visão Geral

- Tabela simples que é derivada de outras tabelas
- Existe necessariamente em sua forma física
  - não é uma tabela virtual
- Discussão
  - replicação dos dados
  - armazenamento de dados agregados
  - custo de consultas x custo de atualização

Visão Geral

- Relações que podem ser usadas para
  - sumarizar
  - computar
  - replicar
  - distribuir
- Útil em vários ambientes computacionais
  - data warehousing
  - computação distribuída e móvel

Casos de Uso

- Data warehouses
  - usado para computar e armazenar dados agregados tal como somatórios e médias
  - geralmente referenciadas como agragações
  - usadas para computar *joins* com ou sem agregações

- Otimizador de consultas pode utilizar visões materializadas
  - para melhorar o desempenho de consultas, reconhecendo automaticamente quando uma visão materializada pode ser utilizada
  - para reescrever consultas, de maneira transparente para o usuário, de forma que as consultas utilizem visões materializadas ao invés das tabelas ou visões base

Casos de Uso

- Ambientes distribuídos
  - replicar dados pelos *sites* (nós)
  - sincronizar atualizações de diversos *sites* usando métodos de resolução de conflitos
  - manter réplicas locais para evitar acesso a *sites* remotos

- Computação móvel
  - recuperar um subconjunto de dados de servidores centrais para clientes móveis
  - receber atualizações periódicas dos servidores
  - propagar atualizações feitas pelos clientes de volta para os servidores

Comparação com índices

- São similares a índices em diversos aspectos
  - consomem considerável espaço de armazenamento
  - precisam ser atualizadas quando os dados das tabelas base são alterados
  - melhoraram o desempenho de consultas SQL quando são usadas para reescrever consultas
  - são transparentes para aplicações e usuários de SQL

Comparação com índices

- Por outro lado
  - podem ser acessadas diretamente usando o comando *select*
  - podem também ser acessadas, e alteradas, por comandos *insert*, *update* e *delete*

Discussão

- Replicação dos dados
- Armazenamento de dados agregados
- Custo de consultas x custo de atualização

Agenda

- Visão Geral
  - http://download.oracle.com/docs/cd/B28359_01/server.111/b28318/schema.htm#CNCPT411
- Sintaxe
- Exemplo
- Exercícios

Sintaxe

create materialized view *nome_visão*
[build [deferred | immediate]]
[refresh [complete | fast | force]
[on commit | on demand]]
[start with *date*]
[next *date*]]
[for update]
[[enable | disable] query rewrite]
as *select*

Sintaxe

create materialized view *nome_visão*
[build [deferred | immediate]]
[refresh [complete | fast | force]
[on commit | on demand]]
[start with *date*]
[next *date*]]
[for update]
[[enable | disable] query rewrite]
as *select*

Sintaxe

create materialized view *nome_visão*
[build [deferred | immediate]]
[refresh [complete | fast | force]
[on commit | on demand]]
[start with *date*]
[next *date*]]
[for update]
[[enable | disable] query rewrite]
as *select*

Sintaxe

create materialized view *nome_visão*
[build [deferred | immediate]]
[refresh [complete | fast | force]
[on commit | on demand]
[start with *date*]
[next *date*]]
[for update]
[[enable | disable] query rewrite]
as *select*

Sintaxe

create materialized view *nome_visão*
[build [deferred | immediate]]
[refresh [complete | fast | force]
[on commit | on demand]
[start with *date*]
[next *date*]]
[for update]
[[enable | disable] query rewrite]
as *select*

Sintaxe

create materialized view *nome_visão*
[build [deferred | immediate]]
[refresh [complete | fast | force]
[on commit | on demand]
[start with *date*]
[next *date*]]
[for update]
[[enable | disable] query rewrite]
as *select*

Sintaxe

create materialized view *nome_visão*
[build [deferred | immediate]]
[refresh [complete | fast | force]
[on commit | on demand]
[start with *date*]
[next *date*]]
[for update]
[[enable | disable] query rewrite]
as *select*

Sintaxe

create materialized view *nome_visão*
[build [deferred | immediate]]
[refresh [complete | fast | force]
[on commit | on demand]
[start with *date*]
[next *date*]]
[for update]
[[enable | disable] query rewrite]
as *select*

Sintaxe

create materialized view *nome_visão*
[build [deferred | immediate]]
[refresh [complete | fast | force]
[on commit | on demand]
[start with *date*]
[next *date*]]
[for update]
[[enable | disable] query rewrite]
as *select*

Sintaxe

create materialized view *nome_visão*
[build [deferred | immediate]]
[refresh [complete | fast | force]
[on commit | on demand]
[start with *date*]
[next *date*]]
[for update]
[[enable | disable] query rewrite]
as *select*

Materialized View Log

- Utilizado para permitir atualização incremental (refresh fast) da visão

Materialized View Log

create materialized view log on *nome_tabela*
[with [rowid]
[primary_key]
[sequence]
[(*column*,...)]
[[including | excluding] new values]];

Materialized View Log

create materialized view log on *nome_tabela*

[with [rowid]

[primary_key]

[sequence]

[(*column*,...)]

[[including | excluding] new values]];

Materialized View Log

create materialized view log on *nome_tabela*

[with [rowid]

[primary_key]

[sequence]

[(*column*,...)]

[[including | excluding] new values]];

Materialized View Log

create materialized view log on *nome_tabela*

[with [rowid]

[primary_key]

[sequence]

[(*column*,...)]

[[including | excluding] new values]];

Materialized View Log

create materialized view log on *nome_tabela*

[with [rowid]

[primary_key]

[sequence]

[(*column*,...)]

[[including | excluding] new values]];

Materialized View Log

create materialized view log on *nome_tabela*

[with [rowid]

[primary_key]

[sequence]

[(*column*,...)]

[[including | excluding] new values]];

Materialized View Log

create materialized view log on *nome_tabela*

[with [rowid]

[primary_key]

[sequence]

[(*column*,...)]

[[including | excluding] new values]];

Exercícios

- Usar a seguinte consulta nos exercícios

```
SELECT cl.nomeClube, cl.apelidoClube, es.nomeest
       es.capacidadeest, eq.nomeeq, eq.nrotituloseq
FROM   clube cl, equipe eq, estadio es,
       clubepossuiest cles
WHERE  es.nomeest     = cles.nomeest AND
       cles.cnpjclube = cl.cnpjclube AND
       cl.cnpjclube   = eq.cnpjclube;
```

Exercícios

- Criar uma VM *times* que seja povoada imediatamente e atualizada completamente sempre que houver um *commit* nas tabelas base
- Listar todos os dados da VM *times*
- Inserir a tupla
  - equipe(cnpjclube,nomeeq,nrojogadoreseq,nrotituloseq)
  - equipe ('60.517.984/0001-04',‘MASTER‘,50,10)
- Listar todos os dados da VM *times*
- Realizar *commit*
- Listar todos os dados da VM *times*
- Excluir a VM *times*

Exercícios

- Criar uma VM *times* que será povoada apenas quando o usuário solicitar
- Listar todos os dados da VM *times*
- Desafio: povoar a VM *times*
- Listar todos os dados da VM *times*
- Excluir a VM *times*

Exercícios

- Criar uma VM *times* que será povoada imediatamente e atualizada apenas quando o usuário solicitar
- Listar todos os dados da VM *times*
- Inserir a tupla
  - equipe(cnpjClube, nomeEq, nroJogadores, nroTitulos)
  - equipe('60.517.984/0001-04', 'NOVA EQUIPE', 1, 2)
- Listar todos os dados da VM *times*
- Realizar um *commit*
- Listar todos os dados da VM *times*
- Atualizar a VM *times*
- Listar todos os dados da VM *times*
- Excluir a VM *times*

Exercícios

- Criar uma VM *times* que sempre seja atualizada incrementalmente
- Listar todos os dados da VM *times*
- Inserir a tupla
  - equipe(cnpjClube,nomeEq,nroJogadoresEq,nroTitulosEq)
  - equipe('60.517.984/0001-04','JUVENIL',50,10);
- Realizar um *commit*
- Listar todos os dados da VM *times*
- Excluir a VM *times*

Exercícios

- Criar uma VM *times* que seja atualizada incrementalmente quando ocorrerem alterações nas tabelas *equipe*, *estadio* e *clubepossuiest* mas quando a alteração for em *clube* fazer uma atualização completa

- Desafio: mostrar que tipo de atualização é feito na VM para cada uma das seguintes inserções

  - equipe (cnpjclube,nomeeq,nrojogadoreseq,nrotituloseq)
  - equipe ('60.517.984/0001-04','teste',1,1)

  - clube (cnpjclube, NOMECLUBE, APELIDOCLUBE)
  - clube ('1111111111', ‘novo clube', ‘apelido');

- Excluir a VM *times* e os materialized view logs

Exercícios

- Criar uma VM *times* na qual seja possível executar comandos DML
- Listar todos os dados da VM *times*
- Fazer um *insert* na VM *times*
- Listar todos os dados da VM *times*
- Listar todos os dados da tabela base
- Excluir a VM *times*