SQL
CREATE MATERIALIZED VIEW

Processamento Analítico de Dados

Profa. Dra. Cristina Dutra de Aguiar

Visão Materializada

```
CREATE MATERIALIZED VIEW nome_visão
[BUILD [DEFERRED | IMMEDIATE]]
[[REFRESH [COMPLETE | FAST | FORCE]]
        [ON COMMIT| ON DEMAND]]
        [[START WITH | NEXT] DATE ]]
[[ENABLE | DISABLE] QUERY REWRITE]
AS
<SELECT>
```

+ diversas outras opções

BUILD

- Quando a visão materializada é povoada
  - IMMEDIATE: imediatamente
  - DEFERRED: primeiro REFRESH

REFRESH

- Como é feita a atualização da visão
  - COMPLETE x FAST x FORCE
    - COMPLETE: atualiza completamente a visão, executando o comando SELECT
    - FAST: somente considera as alterações realizadas (atualização incremental)
    - FORCE: Oracle vai executar FAST sempre que possível, e COMPLETE caso contrário

REFRESH

- Como é feita a atualização da visão
  - ON COMMIT x ON DEMAND
    - ON COMMIT: atualiza a visão quando as operações realizadas na relação base forem finalizadas com sucesso
    - ON DEMAND: atualiza a visão somente quando um comando específico solicitar

REFRESH

- Como é feita a atualização da visão
  - START WITH x NEXT
    - START WITH: data na qual será realizada a primeira atualização automática
    - NEXT: intervalo de tempo entre duas atualizações automáticas consecutivas

QUERY REWRITE

- Se a visão materializada pode ser usada para reescrita de consultas
  - ENABLE: sim
  - DISABLE: não

Diversas outras opções encontram-se disponíveis, sendo que as funcionalidades dependem do SGBD

Exemplo

- Esquema
  - Campeonato de futebol
- Consulta base

```
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;
```

Visão Materializada 1 (1/5)

- Criar a visão materializada times que seja *povoada imediatamente* e *atualizada* sempre que houver um *commit* nas tabelas base

Visão Materializada 1 (2/5)

- Criar a visão materializada times que seja *povoada imediatamente* e *atualizada* sempre que houver um *commit* nas tabelas base

```
CREATE MATERIALIZED VIEW times
      ("Nome do Clube", "Apelido do Clube",
       "Nome do Estadio", "Capacidade do Estadio",
       "Nome da Equipe", "Numero de Titulos")
BUILD IMMEDIATE
REFRESH ON COMMIT
AS CONSULTA_BASE;
```

Visão Materializada 1 (3/5)

- Listar todos os dados de times

```
SELECT * FROM times;
```

- Inserir uma nova tupla

```
INSERT INTO EQUIPE (cnpjclube, nomeeq,
       nrojogadoreseq, nrotituloseq)
VALUES ('60.517.984/0001-04','MASTER',50,10);
```

Visão Materializada 1 (4/5)

- Listar todos os dados de times

```
SELECT * FROM times;
```

A visão estará atualizada? Por quê?

Visão Materializada 1 (5/5)

- Realizar *commit*

```
COMMIT;
```

- Listar todos os dados de times

```
SELECT * FROM times;
```

A visão estará atualizada? Por quê?

- Excluir times

```
DROP MATERIALIZED VIEW times;
```

Visão Materializada 2 (1/4)

- Criar a visão materializada times que seja *povoada* apenas quando o *usuário* solicitar.

Visão Materializada 2 (2/4)

- Criar a visão materializada times que seja *povoada* apenas quando o *usuário* solicitar.

```
CREATE MATERIALIZED VIEW times
      ("Nome do Clube", "Apelido do Clube",
       "Nome do Estadio", "Capacidade do Estadio",
       "Nome da Equipe", "Numero de Titulos")
BUILD DEFERRED
AS CONSULTA_BASE;
```

Visão Materializada 2 (3/4)

- Listar todos os dados de times

```
SELECT * FROM times;
```

A visão possui dados? Por quê?

Visão Materializada 2 (4/4)

- Povoar times

```
EXECUTE DBMS_MVIEW.REFRESH('times');
```

- Listar todos os dados de times

```
SELECT * FROM times;
```

A visão possui dados?
Por quê?

- Excluir times

```
DROP MATERIALIZED VIEW times;
```

Visão Materializada 3 (1/6)

- Criar a visão materializada times que *seja povoada imediatamente* e que *atualizada* apenas quando o *usuário* solicitar.

Visão Materializada 3 (2/6)

- Criar a visão materializada times que *seja povoada imediatamente* e que *atualizada* apenas quando o *usuário* solicitar.

```
CREATE MATERIALIZED VIEW times
       ("Nome do Clube", "Apelido do Clube",
        "Nome do Estadio", "Capacidade do Estadio",
        "Nome da Equipe", "Numero de Titulos")
BUILD IMMEDIATE
REFRESH ON DEMAND
AS CONSULTA_BASE;
```

Visão Materializada 3 (3/6)

- Listar todos os dados de times

```
SELECT * FROM times;
```

- Inserir uma nova tupla

```
INSERT INTO EQUIPE (cnpjclube, nomeeq,
       nrojogadoreseq, nrotituloseq)
VALUES ('60.517.984/0001-04','MASTER',50,10);
```

Visão Materializada 3 (4/6)

- Listar todos os dados de times

```
SELECT * FROM times;
```

A visão estará atualizada? Por quê?

Visão Materializada 3 (5/6)

- Realizar *commit*

```
COMMIT;
```

- Listar todos os dados de times

```
SELECT * FROM times;
```

A visão estará atualizada? Por quê?

Visão Materializada 3 (6/6)

- Solicitar a atualização de times

```
EXECUTE DBMS_MVIEW.REFRESH('times');
```

- Listar todos os dados de times

```
SELECT * FROM times;
```

A visão estará atualizada? Por quê?

- Excluir times

```
DROP MATERIALIZED VIEW times;
```