

Registrar acessos do tipo FULL TABLE

MODIFICAÇÕES NESTE DOCUMENTO

Item: Criação do Documento

Alteração/Inclusão: todo o documento

Responsável: Gilberto A. Oliveira

Data: 05/08/2023

Índice

Contexto.....	4
Objetivo.....	4
Descrição.....	4

Contexto

Com o objetivo de registrar comandos que fazem acesso do tipo FULL TABLE e que com tempo de atividade acima de 1 minuto foi idealizado uma rotina que composta por tabelas de registro dos eventos (comando e ocorrência desses comandos), procedure que faz o registro das ocorrências e comandos e SCHEDULER que dispara a procedure a cada minuto.

Objetivo

Os objetivos dessa rotina são:

- registrar qual comando SQL está fazendo acesso FULL TABLE
- registrar a ocorrência desse comando com informações suficientes para identificação da origem do comando

Descrição

Para atender os objetivos dessa rotina foram criadas duas tabelas:

ACESSO_FULL – nessa tabela ficam registrados os dados do comando SQL que fez o acesso full.

```
CREATE TABLE "ACESSO_FULL"  
  ("SQL_HASH_VALUE" NUMBER,  
   "SQL_ID" VARCHAR2(13 BYTE),  
   "SQL_FULLTEXT" CLOB  
  );
```

OCORRENCIA_FULL – nessa tabela ficam registrados os dados da sessão que fez o acesso full, bem como a data/hora e a quantidade de ocorrências por dia.

```
CREATE TABLE "OCORRENCIA_FULL"  
  ("OCORRENCIA" DATE,  
   "SCHEMANAME" VARCHAR2(30 BYTE),  
   "OSUSER" VARCHAR2(30 BYTE),  
   "OBJECT_NAME" VARCHAR2(30 BYTE),  
   "MODULE" VARCHAR2(64 BYTE),  
   "ACTION" VARCHAR2(64 BYTE),  
   "SQL_HASH_VALUE" NUMBER,  
   "SQL_ID" VARCHAR2(13 BYTE),  
   "QUANTIDADE" NUMBER  
  );
```

Também foi criada a procedure PR_ACESSO_FULL que tem por objetivo capturar os comandos que estão fazendo o acesso FULL TABLE naquele momento e registrá-lo nas tabelas descritas.

O leitor deve perceber que cada comando SQL só é registrado uma única vez na tabela ACESSO_FULL, já cada comando gera uma ocorrência na tabela OCORRENCIA_FULL. Se for a primeira vez que o comando foi capturado naquele dia ele fará um registro completo (sysdate, schemaname, osuser, object_name, module, action, sql_hash_value, sql_id, 1) se não apenas incrementará o campo quantidade. Dessa forma, espera-se minimizar a quantidade de registros nas tabelas.

Também é possível perceber a existência de 3 relatórios (MENSAL, DIA-1 e DIA) que extraem informações capturadas e armazenadas nas tabelas como descrito anteriormente. Para todos os 3 tipos de relatórios obtêm-se resultados similares exceto pelo período pesquisado.

Exemplo:

```
*****
Usuario: MM_PLUS
SQL_HASH: 354216852
SQL_ID: 6xy17q4ajtuwn
Schema: MM_USER
Module: SQL Developer
Action:
Tabelas: CONTROLE
Hora do 1o registro: 04/09/2023 14:03:02
Núm de Ocorrências: 12
Comando:
SELECT * FROM CONTROLE

*****
Usuario: Holland
SQL_HASH: 3216043374
SQL_ID: f37sdu2zv1vbf
Schema: THESAMPLES
Module: SQL Developer
Action:
Tabelas: PAGAMENTO, NOTA_FISCAL
Hora do 1o registro: 05/09/2023 17:46:02
Núm de Ocorrências: 1
Comando:
SELECT NVL(SUM(ROUND(I.ICECP,2)),0.0) AS ICECP
       FROM PAGAMENTO I, NOTA_FISCAL N
       WHERE-- I.NRSEQ  = P_NRSEQ
              -- AND
              I.CDFILIAL      = N.CDFILIAL
              AND I.NRLANCTONF = N.NRLANCTONF
              AND N.IDENTRSAIDOP = 'S'
              AND I.CDFISCALNF  = '1'

*****
```