

Monitorização de uma Base de Dados



Bruno Miguel Salgado Machado João Paulo Ribeiro Alves João Rui de Sousa Miguel

A74941 A73524 A74237

Índice

[Introdução 1](#_Toc503749355)

[Métricas de Avaliação 2](#_Toc503749356)

[Desempenho do CPU 2](#_Toc503749357)

[Desempenho da Memória 3](#_Toc503749358)

[Análise de Datafiles 4](#_Toc503749359)

[Sessões de Utilizador 5](#_Toc503749360)

[Permissões de Utilizador 6](#_Toc503749361)

[Análise de Utilizadores 7](#_Toc503749362)

[Análise de Tablespaces 8](#_Toc503749363)

[Detalhes de Tabelas 9](#_Toc503749364)

[Leitura de valores 10](#_Toc503749365)

[Desenho de Base de Dados auxiliar 11](#_Toc503749366)

[Modelo Conceptual 11](#_Toc503749367)

[Modelo Lógico 12](#_Toc503749368)

[Modelo Físico 12](#_Toc503749369)

[Escrita de valores obtidos por programa gestor 12](#_Toc503749370)

[Desenvolvimento API Rest 13](#_Toc503749371)

[Pedidos GET 13](#_Toc503749372)

[Resposta JSON 13](#_Toc503749373)

[Implementação 13](#_Toc503749374)

[Interface Web 14](#_Toc503749375)

[Desenho 14](#_Toc503749376)

[Implementação 14](#_Toc503749377)

[Conclusões 15](#_Toc503749378)

[Trabalho Futuro 15](#_Toc503749379)

[ANEXOS 16](#_Toc503749380)

[ANEXO I – Modelo Conceptual 16](#_Toc503749381)

Introdução

Para a realização deste trabalho foi pedido que se desenvolvesse uma aplicação capaz de efetuar a gestão de uma Base de Dados. Para tal utilizou-se a linguagem de programação sob a qual o grupo estava mais apto, sendo esta JAVA.

Numa fase inicial procedeu-se à identificação de todas as métricas capazes de ser apresentadas a um utilizador de forma a que este consiga facilmente ter uma ideia de onde os seus recursos estão a ser utilizados. Posteriormente procedeu-se ao desenvolvimento e implementação de uma nova base de dados, capaz de albergar informações relativas à que se pretende monitorizar. Para tal foram efetuados os esquemas conceptual, lógico e o modelo físico, assim como a normalização destes.

Após finalização do tratamento de dados, começou-se a desenvolver uma API REST, capaz de receber pedidos HTTP GET e responder com informação especificada num formato JSON. Este formato é utilizado para que depois seja possível apresentar ao utilizador os dados relativos à base de dados em questão, de uma forma mais apelativa.

Este relatório serve, assim, para documentar todo o processo de desenvolvimento da aplicação.

Métricas de Avaliação

Nesta secção serão especificados/apresentados os dados considerados importantes para construir um software/aplicação capaz de apresentar a um utilizador o correto estado atual de um sistema de Bases de Dados.

Será importante referir que a recolha de dados foi efetuada através do utilizador “hr” e password “oracle”, que haviam sido disponibilizados em aula.

Desempenho do CPU

Uma das métricas utilizadas para medir o desempenho do nosso sistema, foi a utilização do CPU em tempos, assim como em topologia. Para isto efetuaram-se registos dos números de cores que estavam disponibilizados, tempos de processamento, input-output, *iddle*, etc.

Manteve-se um registo destes valores ao longo do tempo, permitindo traçar um perfil evolutivo do estado de evolução e de ocupação da base de dados.

Na figura 1 demonstra-se a query utilizada para obter estes dados, esta produz o resultado demonstrado na figura 2.

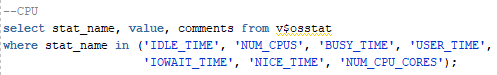


Figura 1 - Query efetuada para obter dados CPU

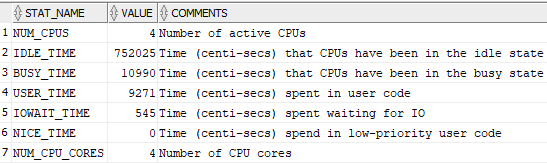


Figura 2 - Output produzido pela query relativa ao CPU (Figura 1)

Desempenho da Memória

A segunda métrica utilizada diz respeito aos valores de ocupação da memória relativa ao espaço de endereçamento do sistema e ao espaço de endereçamento do programa. Estes dizem respeito, respetivamente, ao espaço ocupado em estruturas de memória que sejam partilhadas, conhecidas como *SGA Components*, e espaço que contém informação e dados de controlo relativos a um processo do servidor.

Tal como efetuado nas medições do CPU, foram guardados os valores de memória SGA e PGA relativos a um dado instante. Para tal utilizou-se a técnica de *timestamping*. Na figura 3 poder-se-á visualizar a query utilizada para obtenção destes dados, e a figura 4 apresenta o *output* obtido.

Note-se, ainda, que a primeira linha dirá respeito ao valor de SGA enquanto a segunda linha remete para o valor de PGA.

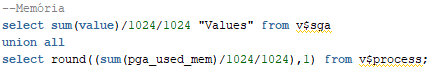


Figura 3 - Query efetuada para obtenção de dados de memória



Figura 4 - Resultado obtido pela execução da query presente na figura 3

Análise de Datafiles

Segue-se o terceiro conjunto de dados que se considerou importantes registar, os datafiles, isto dado que é nestes que serão guardadas as tabelas, índices, procedures e views de um sistema de base de dados. No que toca a estes ficheiros, obteve-se o nome do tablespace ao qual um datafile está associado (isto ajudar-nos-á mais à frente), o nome de cada datafile, o seu tamanho total, o espaço que está atualmente a ser utilizado e a percentagem de utilização.

Para que fosse possível obter dados relativos a estes mesmos ficheiros, utilizou-se a query SQL disponibilizada na figura 5. Esta produz o output vivível na figura 6.

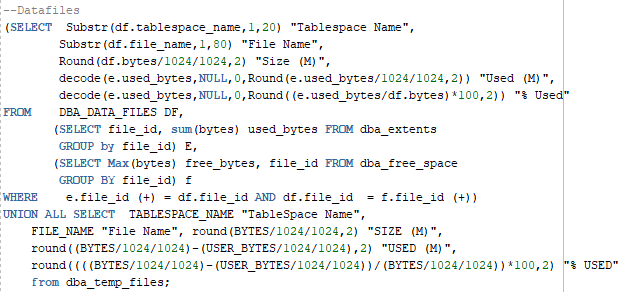


Figura 5 - Query de obtenção de informações relativas ao estado dos datafiles

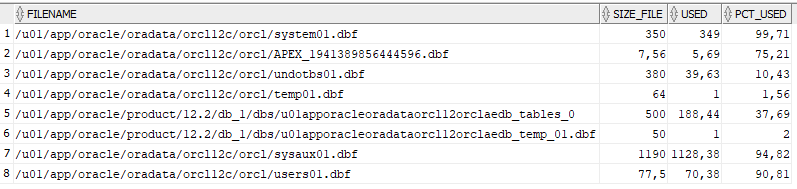


Figura 6 - Output obtido pela query presente na figura 5

Sessões de Utilizador

As sessões de utilizador dizem respeito ao registo de informações de sessão efetuadas por estes, estando associadas a um id (sid), ao nome de utilizador (OS\_USER) e às máquinas e programas sob a qual se definem as sessões existentes.

Utilizou-se a query da figura 7 para obter estes dados, o resultado está definido na figura 8, onde é possível visualizar um exemplo de sessão. A query foi realizada no SQL Developer na máquina Captain-PC.

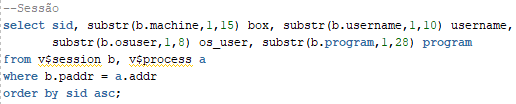


Figura 7 - Query utilizada para obter dados relativos às sessões de utilizadores

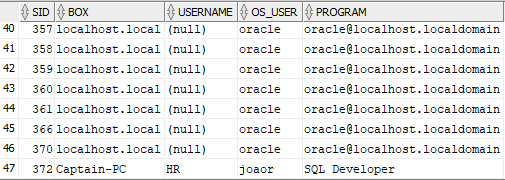


Figura 8 - Output produzido pela query na figura 7

Permissões de Utilizador

As permissões de utilizador são o quinto elemento que o grupo considerou necessário manter registo. Estes registos dizem respeito às permissões que certos utilizadores têm dentro do sistema de base de dados. As informações obtidas para a utilização e apresentação do estado do sistema passa por: a quem a permissão se destina, que tipo de permissão é que se esta a referir, se esta tem a opção de administrador, common, ou se resulta de uma relação de herança (no sentido de hierarquia).

Para a manutenção dos grants concedidos, utilizou-se a técnica de *timestamp* para verificar quando uma autorização foi concedida a um utilizador.

Uma vez mais, para que seja possível obter estas métricas utilizou-se a query visível na figura 10, sendo o output obtido pela execução desta, o disponibilizado na figura 11.



Figura 9 - Comando SQL utilizado para obter informações de permissões de utilizadores

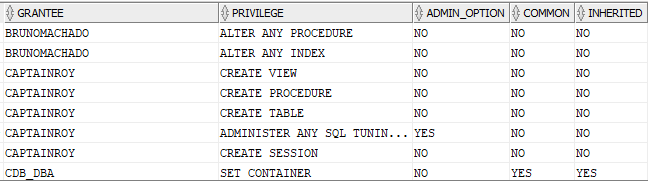


Figura 10 - Resultado obtido pela execução da query presente na figura 9

Análise de Utilizadores

De seguida, obteve-se um conjunto de informações relativas aos utilizadores existentes na base de dados. Estes possuem dados como o nome de utilizador, dois tablespaces (sendo um destes temporário), e a indicação de quando foi efetuado o último login (que poderá nunca ter ocorrido). Estes serão posteriormente referenciados aos tablespaces e aos grants, para que se consiga obter, num panorama mais abrangente, um conjunto mais amplo de informações relativas a cada utilizador.

A figura 11 apresenta o código SQL utilizado para obter os dados que o grupo considerou importantes relativamente aos utilizadores. A figura 12 apresenta o resultado obtido por esta mesma query, sendo que são estes valores que serão, de certa forma, introduzidos na nova base de dados de monitorização.



Figura 11 - Query utilizada para obter resultados relativos aos dados dos utilizadores

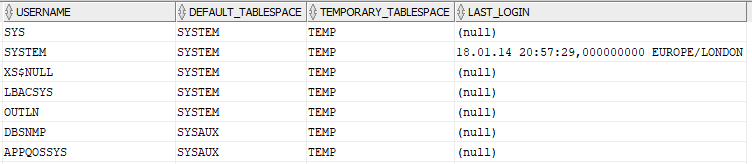


Figura 12 - Resultado representante dos dados obtidos pela query da figura 11

Análise de Tablespaces

O sétimo conjunto de dados que se decidiu tratar foram os tablespaces, estes dizem respeito ao armazenamento lógico dos dados presentes nas tabelas e outros objectos de sql. Estão relacionados com os datafiles (utilizados para guardar os dados fisicamente). Estes contêm informações relativas ao nome do tablespace, à percentagem utilizada destes, ao espaço total, utilizado e livre assim como a presença ou ausência de datafiles associados.

Para obter informações relativamente a este tipo de estruturas, produziu-se a query presente na figura 13, sendo que esta retorna o output especificado na figura 14.

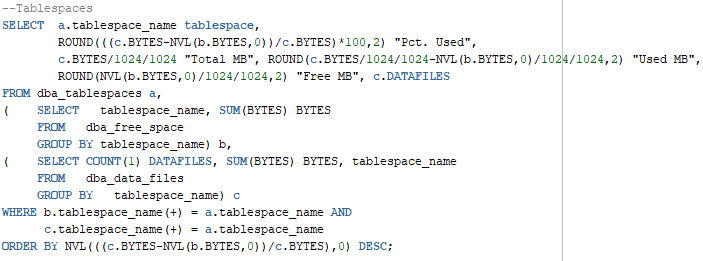


Figura 13 - Query utilizada para obter dados relativos aos tablespaces

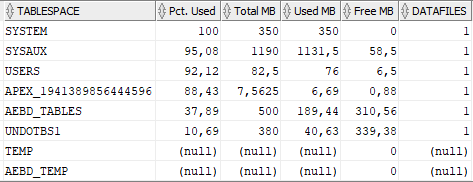


Figura 14 - Resultado obtido pela execução da query SQL presente na figura 13

Detalhes de Tabelas

As tabelas compreendem todos os dados que serão guardados num sistema de base de dados. Deste modo procedeu-se à recolha de informação relativa ao estado destas. Obteve-se o schema sob a qual uma tabela diz respeito, o seu nome, o seu tamanho e informação relativa ao tablespace sob a qual estas estão definidas.

Para obter estas métricas utilizou-se o comando SQL explicito na figura 15, sendo o seu output o existente na figura 16.

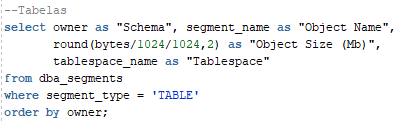


Figura 15 - Query utilizada para obter informação relativamente às tabelas

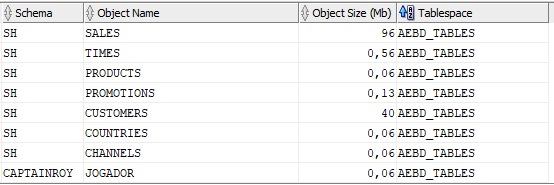


Figura 16 - Output obtido pela execução do commando disponibilizado na figura 15

Leitura de valores

Tal como foi dito anteriormente, este monitor foi desenvolvido através da linguagem de programação JAVA, deste modo foi necessário utilizar bibliotecas que nos permitissem a ligação a uma base de dados oracle. O grupo decidiu utilizar uma biblioteca conhecida, a JDBC. Após a sua incorporação no projeto, foi apenas necessário iniciar uma ligação com a base de dados, para que depois se pudessem realizar todos estes pedidos, procedendo ao tratamento dos dados.

Utilizaram-se PreparedStatements, disponibilizados por esta biblioteca para que o nosso código não ficasse vulnerável a SQL Injection. Além disto também se utilizou o ResultSet para conseguir tratar os dados recebidos no programa.

Para armazenar os dados foram desenvolvidas classes definidas para cada uma das tabelas, e subclasses relativas às informações que estas poderiam apresentar. Refira-se ainda que todos os pedidos de valores às tabelas são efetuados por Threads independentes entre si, sendo que podem existir pedidos efetuados em simultâneo.

A figura 17 apresenta um excerto do código sob a qual se pode observar a utilização destes mesmos elementos. Este excerto poderá ser adaptado para quaisquer outra query que se pretenda obter resultados, sendo que a única alteração necessária será sob a forma de como os dados são tratados, isto porque o ResultSet, em norma, produz outputs diferentes para querys distintas.

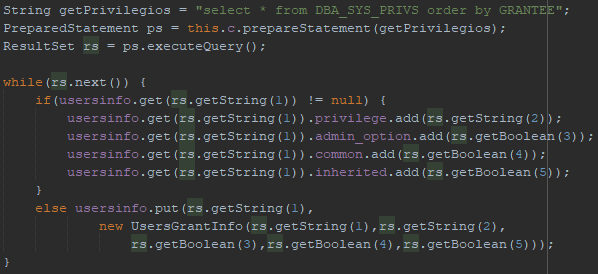


Figura 17 - Excerto de código representante da utilização dos recursos PreparedStatement e ResultSet disponibilizados pela biblioteca JDBC

Desenho de Base de Dados auxiliar

A segunda etapa de desenvolvimento deste trabalho, contemplou a especificação e resolução de um novo esquema para uma base de dados, capaz de albergar as informações obtidas pela aplicação de monitorização desenvolvida anteriormente. Este passo foi desenvolvido após se obter um código java que continha todas as informações em objetos, sendo que para proceder ao registo em memória “permanente” apenas foi necessário adicionar um novo pacote, que conseguisse precisamente ler estas informações dos objetos e escrever para a base de dados.

Tal abordagem demonstra a capacidade de modularização da aplicação, sendo que em case de alteração de alguma métrica é relativamente simples proceder à alteração de valores, ou funções especificas, diminuindo o trabalho necessário por parte do programador.

Modelo Conceptual

A primeira fase de desenho de uma base de dados, contempla o desenho de um esquema conceptual que seja compatível com todos os dados disponibilizados nos objetos java.

O desenvolvimento da base de dados foi realizado em torno dos dados em si, sendo que adaptamos esta em função das informações e não o oposto. Deste modo podemos considerar que foram necessárias 8 tabelas, sendo que cada uma destas diz respeito aos valores obtidos anteriormente. Cada uma destas tabelas tem ainda atributos específicos a si, sendo que algumas mantém o *timestamp* que será utilizado para manter um registo temporal da evolução da base de dados.

O modelo conceptual a que se chegou pode ser observado na figura 18, sendo que para uma melhor resolução se deverá observar o anexo I.

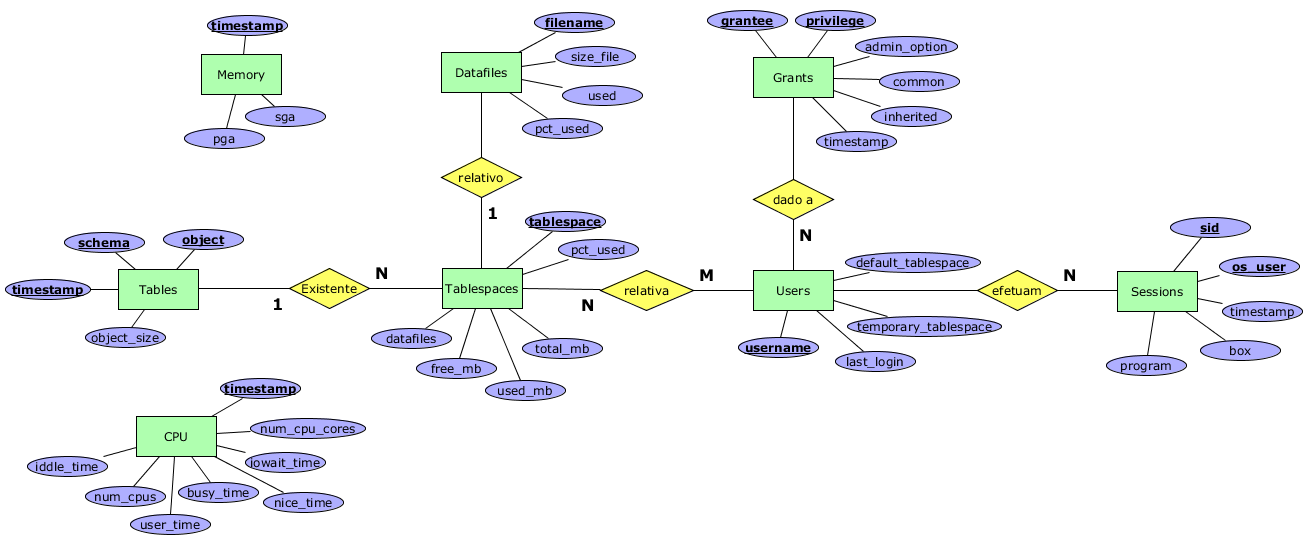


Figura 18 - Modelo conceptual desenvolvido para esta nova base de dados

Nesta identificam-se com clareza as relações existentes entre as diferentes tabelas, sendo que é necessário introduzir um pouco de redundância no que toca ao *timestamp* das tabelas Grants e Sessions uma vez que estas podem não ter correspondência com a tabela Users. É ainda possível observar que existem duas tabelas que serão utilizadas para manter registos e que não serão relacionadas com nenhuma outra, estas são a Memory e a CPU, as suas informações dizem respeito a um panorama geral e abrangente, sendo que não existem dados específicos a ser tratados sobre estas.

Modelo Lógico

Yh fhjr ghg rhr hg rb rh

Modelo Físico

Yh fhjr ghg rhr hg rb rh

Escrita de valores obtidos por programa gestor

Blá blá

Desenvolvimento API Rest

Blá blá blá

Pedidos GET

Yh fhjr ghg rhr hg rb rh

Resposta JSON

Yh fhjr ghg rhr hg rb rh

Implementação

Yh fhjr ghg rhr hg rb rh

Interface Web

Blá blá blá

Desenho

Yh fhjr ghg rhr hg rb rh

Implementação

Yh fhjr ghg rhr hg rb rh

Conclusões

Blá blá blá

Trabalho Futuro

Yh fhjr ghg rhr hg rb rh

ANEXOS

ANEXO I – Modelo Conceptual

