

Mestrado em Engenharia Informática

Projeto AEBD

Carlos Magalhães
(pg42817)

Jorge Brandão Gonçalves
(pg42838)

Miguel Cardoso
(a85315)

Miguel Ribeiro
(pg44419)

27 de janeiro de 2021

Resumo

Este projecto foi proposto na âmbito da UC de Administração e Exploração de Base de Dados do perfil de Engenharia de Conhecimento como parte do Mestrado de Engenharia Informatica.

Este projecto tem como âmbito construir um sistema de monitorização de Base de Dados que apresente de forma simples os principais parâmetros de avaliação e de performance de uma Base de Dados Oracle.

Área de Aplicação: Engenharia do Conhecimento, Administração e Exploração de Base de Dados, Oracle

Repositório Github com o código: <https://github.com/pg42817/aebd>

De modo a ser mais fácil partilhar o código com o professor, metemos neste repositório o código assim como na blackboard.

Conteúdo

1	Introdução	2
1.1	Contextualização	2
1.2	Estrutura do Relatório	2
2	Projecto	3
2.1	Requisitos do projeto	3
3	Implementação	4
3.1	Introdução	4
3.2	Modelo Relacional	4
3.3	Agente	6
3.4	Frontend	7
3.5	Problemas do Projeto	7
4	Conclusão	9

Capítulo 1

Introdução

1.1 Contextualização

Este projeto foi proposto no âmbito da UC de Administração e Exploração de Bases de dados no perfil de Engenharia Informática para o Mestrado em Engenharia Informática da Universidade do Minho.

O objetivo deste projeto é a criação de um agente que recolha informações sobre as nossas bases de dados oracle, e que as guarde num *schema* criado pelo grupo.

Era também pedido que fosse feita uma API Rest que se ligue á PDB criada no *schema* anterior e que devolvesse esses dados no formato necessário.

O projeto deverá ser complementado com uma Interface Web de modo a apresentar os dados retirados pelo agente de maneira simples e correta.

1.2 Estrutura do Relatório

Este relatório está dividido em 4 grandes partes.

1. Na primeira faz-se a contextualização do projeto.
2. Na segunda fala-se dos requisitos dados pelos docentes para o projeto.
3. Na terceira parte consiste em explicar a nossa implementação relativamente ao que foi pedido, esta parte é a mais importante e por isso será mais detalhada em relação ao resto do relatório.
4. Por fim, temos uma conclusão, com uma reflexão final sobre o projecto.

Capítulo 2

Projecto

2.1 Requisitos do projeto

Para este projeto foram comunicados aos alunos 4 pontos necessários á conclusão do projeto e que iriam ser avaliados.

Estes pontos são os seguintes:

1. Compreensão e correta criação do schema oracle que irá albergar os dados recolhidos pelo agente: Tablespaces, datafiles, users, grants, tabelas, diagrama ER e normalização da BD.
2. Compreensão e correta recolha dos dados necessários das views de DBA (Tablespaces, Datafiles, Users, Memory, Sessions, CPU,...).
3. Correta apresentação dos dados na Interface web.
4. Documentação (relatório final) de todo o processo.

Capítulo 3

Implementação

3.1 Introdução

Inicialmente, criamos uma PDB com o respetivo schema, tablespaces e datafiles.

De seguida foi criado um agente, em JAVA, com o objetivo de recolher a informação para ser armazenada na Base de Dados.

O passo seguinte foi criar uma API REST que se conecte á Base de Dados e devolva os resultados.

Por fim, foi criada uma interface que apresenta os dados recolhidos.

3.2 Modelo Relacional

O modelo escolhido pelo grupo foi o seguinte:

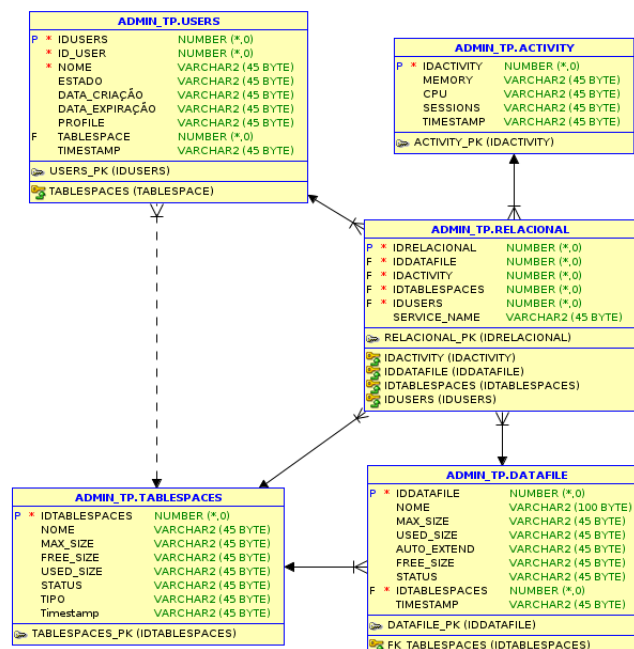


Figura 3.1: Modelo Relacional

Para isto foi criado um schema completo da seguinte forma.

Em primeiro lugar criou-se uma nova pdb com os seguintes comandos.

```
1 CREATE TABLESPACE trabalho_pratico DATAFILE 'tp_files.dbf' SIZE 500M;
2 CREATE TEMPORARY TABLESPACE tp_temp TEMPFILE 'tp_temp_files.dbf' SIZE
  ↳ 50m;
3 CREATE USER tp_admin IDENTIFIED BY "123456" DEFAULT TABLESPACE
  ↳ trabalho_pratico QUOTA UNLIMITED ON trabalho_pratico TEMPORARY
  ↳ TABLESPACE tp_temp;
4 GRANT CONNECT, RESOURCE, CREATE VIEW, CREATE SEQUENCE TO tp_admin ;
```

A criação das tabelas, foi a segunda e última parte do schema e foi feito da seguinte forma.

```
1 Create TABLE Tablespaces
2 (
3   idTablespaces INT GENERATED BY DEFAULT ON NULL AS IDENTITY,
4   Nome VARCHAR(45) NULL,
5   Max_size VARCHAR(45) NULL,
6   Free_size VARCHAR(45) NULL,
7   Used_size VARCHAR(45) NULL,
8   Status VARCHAR(45) NULL,
9   Tipo VARCHAR(45) NULL,
10  "Timestamp" VARCHAR(45) NULL,
11  PRIMARY KEY (idTablespaces)
12 );
13
14 CREATE TABLE Datafile (
15   idDatafile INT GENERATED BY DEFAULT ON NULL AS IDENTITY,
16   Nome VARCHAR(100) NULL,
17   Max_size VARCHAR(45) NULL,
18   Used_size VARCHAR(45) NULL,
19   Auto_extend VARCHAR(45) NULL,
20   Free_size VARCHAR(45) NULL,
21   Status VARCHAR(45) NULL,
22   idTablespaces INT NOT NULL,
23   Timestamp VARCHAR(45) NULL,
24   PRIMARY KEY (idDatafile),
25   CONSTRAINT fk_Tablespaces FOREIGN KEY (idTablespaces) REFERENCES
  ↳ Tablespaces(idTablespaces)
26 );
27
28 CREATE TABLE Users (
29   idUsers INT GENERATED BY DEFAULT ON NULL AS IDENTITY,
30   id_User INT NOT NULL,
31   Nome VARCHAR(45) NOT NULL,
32   Estado VARCHAR(45) NULL,
33   Data_criação VARCHAR(45) NULL,
```

```

34 Data_expiração VARCHAR(45) NULL,
35 Profile VARCHAR(45) NULL,
36 Tablespace INT NULL,
37 Timestamp VARCHAR(45) NULL,
38 PRIMARY KEY (idUsers),
39 CONSTRAINT Tablespace FOREIGN KEY (Tablespace) REFERENCES
    ↳ Tablespace (idTablespaces));
40
41 CREATE TABLE Activity (
42     idActivity INT GENERATED BY DEFAULT ON NULL AS IDENTITY,
43     Memory VARCHAR(45) NULL,
44     Cpu VARCHAR(45) NULL,
45     Sessions VARCHAR(45) NULL,
46     Timestamp VARCHAR(45) NULL,
47     PRIMARY KEY (idActivity));
48
49 CREATE TABLE Relacional (
50     idRelacional INT GENERATED BY DEFAULT ON NULL AS IDENTITY,
51     idDatafile INT NOT NULL,
52     idActivity INT NOT NULL,
53     idTablespaces INT NOT NULL,
54     idUsers INT NOT NULL,
55     service_name VARCHAR(45) NULL,
56     PRIMARY KEY (idRelacional),
57     CONSTRAINT idDatafile FOREIGN KEY (idDatafile) REFERENCES
        ↳ Datafile(idDatafile),
58     CONSTRAINT idActivity FOREIGN KEY (idActivity) REFERENCES
        ↳ Activity(idActivity),
59     CONSTRAINT idTablespaces FOREIGN KEY (idTablespaces) REFERENCES
        ↳ Tablespaces(idTablespaces),
60     CONSTRAINT idUsers FOREIGN KEY (idUsers) REFERENCES Users (idUsers)
61 );

```

3.3 Agente

O agente foi feito em Java.

Inicialmente começamos por criar uma classe relativa a cada tabela (do modelo relacional em 3.2) para posteriormente criarmos o seu respetivo objeto(a ser enviado para a base de dados).

Após ter-mos criado estes objetos (que no nosso projeto dizem respeito aos métodos com o prefixo "trata") começamos a criar o nosso método main. Neste primeiramente conectamo-nos à base de dados usando o driver JDBC.

Agora com ligação á base de dados usamos os métodos para tratar cada um dos dados das tabelas.

O trabalho destes métodos é correr e fazer um parsing de um comando sql que devolve as informações necessárias ao povoamento da tabela em questão.

De modo a serializar/mapear os obeitos em java para o formato JSON usamos a biblioteca "jackson".

No final de ter-mos as informações necessárias fazemos o insert no schema criado na secção acima.

De modo a povoar a base de dados, corremos estes métodos num loop de modo a ir buscar informações de 1 em 1 minuto para termos um histórico dos dados (requisito fornecido no enunciado).

3.4 Frontend

O frontend foi feito em javascript, nodeJs.

Este cria rotas que irão devolver as informações necessárias conforme a rota em questão.

Estas rotas comunicam com um *controller*, através de pedidos.

O *controller* irá redireccionar os pedidos á base de dados retornando assim as informações á rota.

Foram feitas uma série de *views* em pug que é um template do nodejs.

Por fim, para o *design* foi usada a stylesheet da w3schools.

Administração e Exploração de Bases de Dados			
API para Monitorização de uma Base de Dados Oracle			
<div>Activity</div> <div>Datfiles</div> <div>Tablespaces</div> <div>Users</div>			
Dados mais recentes			
Activity	Datfiles	Tablespaces	Users
ID Activity : 92 Memória : 635.76 CPU : 0 Sessão : 64 Timestamp : 2021-01-27 18:51:30	ID Datfile : 40 Nome : /u02/app/oracle/oradata/ORCLCDB/orclpdb1/users01.dbf Max Size : 32767 Used Size : 32758 Auto Extend : YES Free Size : 9 Status : ID Tablespace : 40 Timestamp : 2021-01-27 18:51:58	ID Tablespace : 40 Nome : USERS Max Size : 32767 Free Size : 32766 Used Size : 1 Status : ONLINE Tipo : PERMANENT Timestamp : 2021-01-27 18:51:30	ID User : 127 Nome : ADMIN_TP Estado : OPEN Data Criação : 2021-01-17 18:31:43 Data Expiração : 2021-07-16 18:31:43 Profile : DEFAULT Tablespace : 36 Timestamp : 2021-01-27 18:52:26

Figura 3.2: Design do site

Na imagem acima, vemos o nosso *frontend* onde temos 4 botões clicáveis onde se pode ver mais informação sobre cada tabela.

Em baixo disso temos os dados mais recentes adicionados á base de dados.

3.5 Problemas do Projeto

O primeiro grande problema do grupo foi entender o enunciado, pois pareceu nos um pouco confuso no que era pedido.

Após consultar o docente, foi nos esclarecido prontamente o que era pedido para fazer no projeto.

O segundo problema que o grupo teve foi que informações relativas às bases de dados.

Após diversas reuniões de grupos, e feedback (por mail) do docente este problema foi ultrapassado.

O problema que mais trabalho deu para resolver, foi o relativo à junção das diferentes tópicos do trabalho. Isto é, passar o agente de um dos membros para outros nem todas as livrarias usadas (JDBC, Jackson) funcionavam.

Com ajuda entre a parte do grupo que tratou do agente, este problema foi resolvido.

Capítulo 4

Conclusão

Com este trabalho propusemo-nos a fazer um sistema de monitorização de base de dados que apresente os principais parametros de avaliação de performace e podemos concluir que este trabalho foi um sucesso visto que no geral,fomos capazes de cumprir com o tudo o que nos foi proposto.

Nomeadamente, criar um schema capaz de albergar de forma correta e organizada todos os dados que posteriormente foram recolhidos pelo nosso agente.

Também,conseguimos criar uma API que se conecta á nossa base de dados e vai buscar esses resultados que depois são mostrados de forma explicita na nossa interface.

Apesar de alguns, que maioritariamente aconteceram na interpretação do enunciado, conseguimos realizar o projeto com sucesso e o balanço é, portanto, francamente positivo.