

UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

ADMINISTRAÇÃO E EXPLORAÇÃO DE BASE DE DADOS

Monitor de avaliação de performance de uma BD Oracle

Bárbara Cardoso (A80453)

João Vilaça (A82339)

5 de Março de 2021

Resumo

É inegável que vivemos atualmente num mundo absolutamente ligado à tecnologia, a maioria das ações que fazemos no dia a dia estão de algum relacionados com algum sistema de informação, como quando consultamos o tempo no site do IPMA (Instituto Português do Mar e da Atmosfera) ou efetuamos uma comprar numa loja e imediatamente despoletamos o sistema de faturação, o sistema de stocks, o sistema de encomendas, etc. E o que a maioria deste sistema têm em comum? Um base de dados que suporta toda essa operação. As bases de dados assumem-se como componentes absolutamente essenciais no nosso estilo de vida atual, e é então importante que as mesmas funcionem no máximo das suas capacidades a maior parte do tempo e sem falhas para garantir que os serviços funcionem e, para isso, é necessário mantê-las sobre vigilância e monitorização apertada.

Neste trabalho, para fins educativos, no âmbito da unidade curricular de Administração e Exploração de Bases de Dados, pretende-se desenvolver um sistema de monitorização, sobre uma *Oracle Database*, que permite acompanhar a performance e a evolução das várias métricas da base de dados, quer em tempo real quer historicamente.

Conteúdo

1	Introdução	3
2	Análise do Problema	4
2.1	Descrição do Problema	4
2.2	Arquitetura	4
3	Análise e Construção da Base de Dados	5
3.1	Schema de Monitorização	5
3.2	Modelo Relacional	5
3.3	Recolha e preenchimento de dados	6
3.4	Processamento e consulta de histórico	6
4	Componente Aplicacional	7
4.1	Servidor de dados	7
4.1.1	Conexão à base de dados	7
4.1.2	Controladores	7
4.2	Interface	8
4.3	Controlo de Estado	8
4.4	Dashboards de apresentação de métricas	9
4.5	Exemplos	9
4.5.1	CPU	10
4.5.2	Memória	10
4.5.3	Tablespaces	11
4.5.4	Datafiles	11
4.5.5	Utilizadores	13
4.5.6	Sessões	13
5	Conclusão	15
6	Bibliografia	16
7	Anexos	17
7.1	Criação de Tablespace	17
7.2	Criação do utilizador Monitor	17
7.3	Criação de tabelas	17
7.4	Criação das views do histórico do CPU	18
7.5	Criação das views do histórico da Memória	19
7.6	Criação das views do histórico dos Datafiles	20
7.7	Criação das views do histórico dos Tablespaces	21
7.8	Criação do job de atualização do histórico	23
7.8.1	Procedure para atualização	23
7.8.2	Agendamento do procedure update_values procedure para ser executado a cada minuto	23

1 Introdução

Atualmente, os Sistemas Gestores de Bases de Dados (SGBD), devido às cada vez maiores necessidades de armazenamento e processamento de grandes quantidades de dados, estão a tornar-se cada vez mais poderosos e capazes. No entanto, nota-se um aumento inegável da sua complexidade que vai acompanhando este crescimento. Nesta medida, é fundamental a constante monitorização das bases de dados. Ao longo deste projeto, será desenvolvido um sistema de monitorização de uma *Oracle Database*, um dos grandes líderes do mercado de SGBDs a nível mundial.

Em primeiro lugar, será analisado o problema e será desenhada uma possível arquitetura que simplifique e otimize quer o processo de desenvolvimento da solução, mas também do seu *deploy* e do seu uso por parte dos utilizadores.

Desenhada a arquitetura, passaremos por todo o processo de decisão de quais dados deverão ser analisados e persistidos, da modelação e implementação física da base de dados e pela automatização do processo de recolha de dados.

Por fim, será desenvolvida toda a componente aplicacional para a consulta e apresentação das métricas de monitorização.

2 Análise do Problema

2.1 Descrição do Problema

Para uma correta monitorização da evolução e performance da base de dados, foi necessário analisar e discutir possíveis métricas e dados relevantes para acompanhar a base de dados, como serão tratados e recolhidos estes dados, como serão à posteriori consultados e depois apresentados ao utilizador final.

Será então necessário recolher dados importantes e com valor como, o estado do CPU em termos de hardware disponível e em termos de carga. Em relação à memória recolher qual a memória total disponível e a memória usada.

No que diz respeito a *storage*, é importante recolher todos os dados possíveis sobre quer tablespaces quer sobre datafiles, para monitorizar o seu uso, o seu estado e configurações e perceber a sua evolução em termos de consumo de espaço em disco.

Por fim, é também bastante importante verificar e analisar quais os utilizadores presentes no sistema, qual o seu estado e sobretudo as sessões que cada um tem no preciso momento, quer ativas quer inativas, e analisar que ações estão a efetuar sobre a base de dados.

2.2 Arquitetura

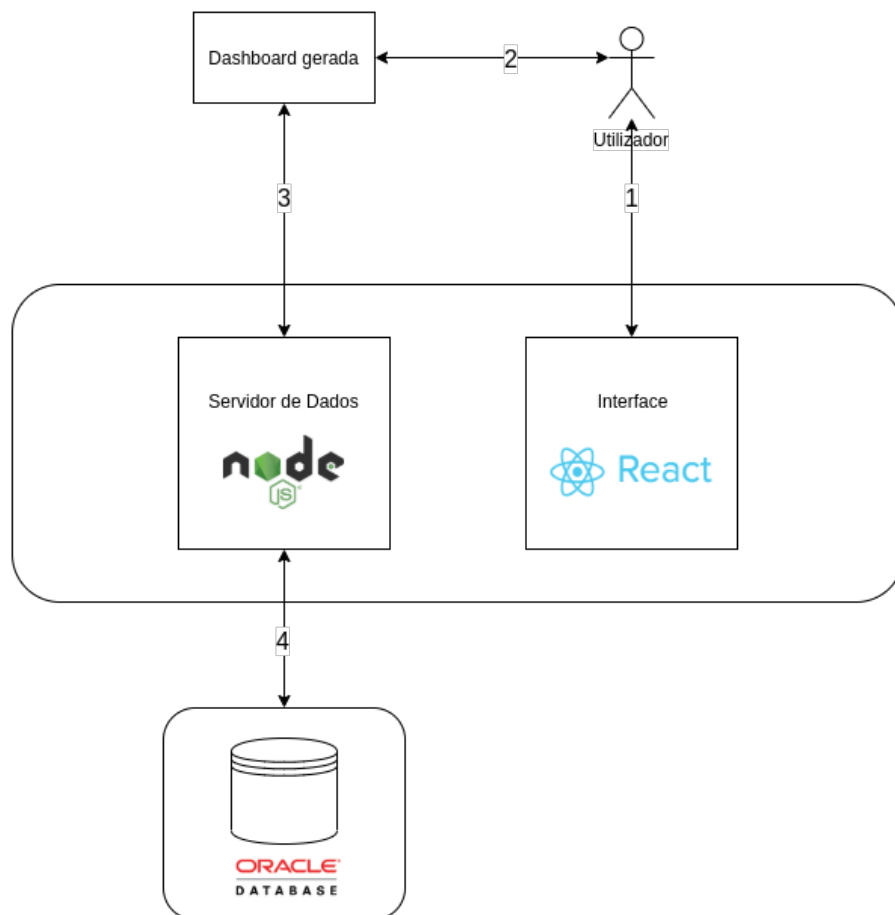


Figura 1: Arquitetura

3 Análise e Construção da Base de Dados

3.1 Schema de Monitorização

O primeiro passo, é criar um tablespace e datafiles para o armazenamento dos dados históricos (anexo: 7.1 - Criação do Tablespace) e criar um utilizador *MONITOR* que faça a gestão destes dados e atribuir-lhe o tablespace e permissões para a sua utilização (anexo: 7.2 - Criação do Utilizador).

Após a criação passou-se a executar todos os próximos passos com este utilizador através da ferramenta *Oracle SQL Developer*.

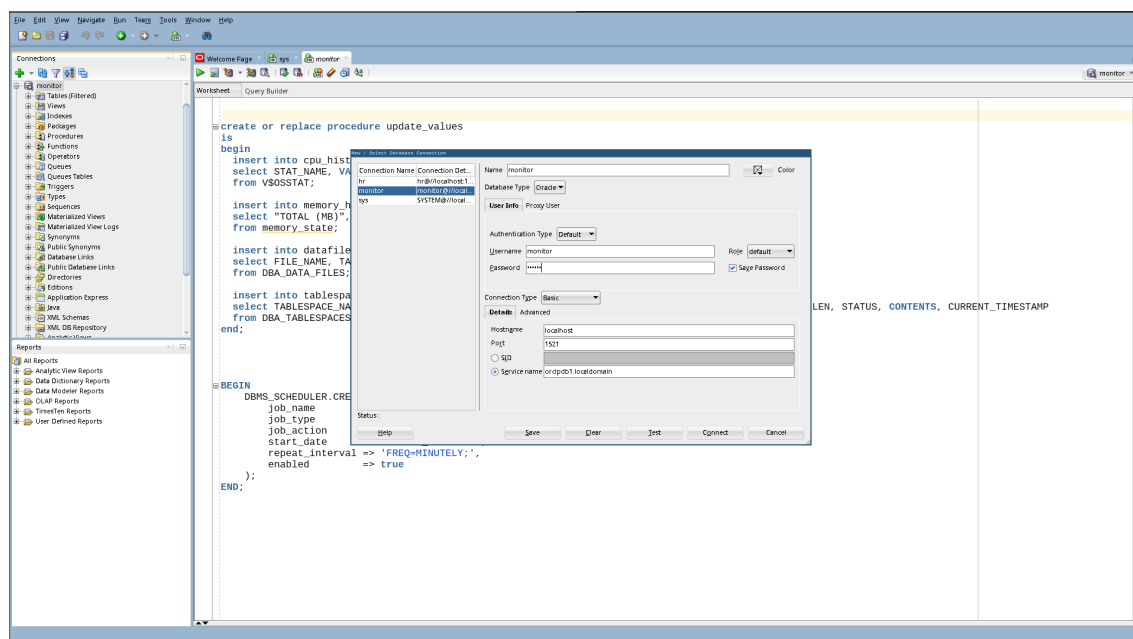


Figura 2: Oracle SQL Developer

3.2 Modelo Relacional

Desenhou-se o modelo relacional de dados sobre o qual serão guardados todos os dados históricos das métricas recolhidas.

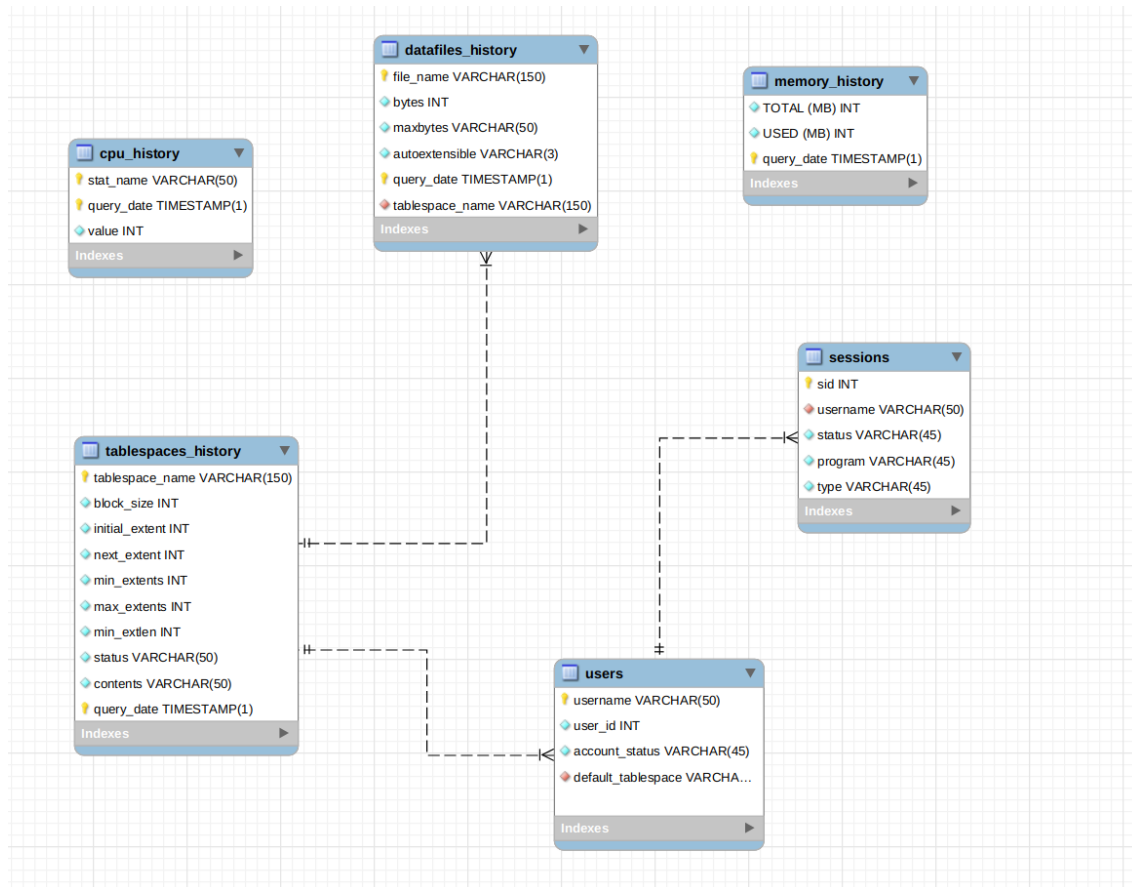


Figura 3: Modelo relacional

3.3 Recolha e preenchimento de dados

Para manter e atualizar o histórico de dados, foi desenvolvido e aplicado um 'procedure' que consulta as tabelas de sistema com as informações sobre os vários 'resources' e insere os dados considerados para histórico nas respectivas tabelas que foram apresentadas anteriormente. (anexo: 7.8.1 - Criação do 'procedure')

Para que o histórico possa dar uma visão geral em termos de análise de dados daquilo que realmente foi a evolução da performance e do uso da base de dados ao longo do tempo, é, no mínimo, importante que sejam recolhidos dados em intervalos de tempo curtos de maneira automática. Para atingir este objetivo, através do 'DBMS_SCHEDULER' foi criado um 'job' que a cada minuto executa o 'procedure' desenvolvido. (anexo: 7.8.2 - Agendamento da recolha de dados)

3.4 Processamento e consulta de histórico

Para processar os dados estatisticamente em termos de períodos temporais foram desenvolvidos para cada recurso *views* que fazem esse tratamento e que serão depois consultadas pelo servidores de dados para servir de maneira simples e rápida as métricas de monitorização da base de dados pedidas. (anexo: 7.5 - Exemplo das views das métricas de Memória)

4 Componente Aplicacional

4.1 Servidor de dados

Uma vez que todo o trabalho de seleção e lógica temporal de análise de dados já se encontra efetuado no lado da base de dados através da criação de views para as tabelas de histórico, a API de dados fica muito facilitada.

Este servidor, desenvolvido em Node.js, limita-se apenas a receber os pedidos do frontend, a perceber a que resource é que ele se refere e se existem as não especificações em termos de unidade temporal e quantidade de dados. Feito este processamento e depois da análise dos resultados, o servidor faz uma 'query' à base de dados quer para obter os dados atuais de determinado 'resource' ou dados históricos dos mesmos, dependendo tal como referido do pedido efetuado.

4.1.1 Conexão à base de dados

A conexão à base de dados é feita recorrendo ao uso do módulo 'oracledb' que suporta várias funcionalidades básicas e avançadas da *Oracle Database* e do *Oracle Client*.

```
const oracledb = require('oracledb');
oracledb.outFormat = oracledb.OUT_FORMAT_OBJECT;

const DB = module.exports;

const config = {
  connectString: process.env.DB_CONNECT_STRING,
  password: process.env.DB_PASSWORD,
  poolIncrement: process.env.DB_POOL_INCREMENT,
  poolMax: process.env.DB_POOL_MAX,
  poolMin: process.env.DB_POOL_MIN,
  user: process.env.DB_USER
};

DB.exec = async (query) => {
  const connection = await oracledb.getConnection(config);

  return connection.execute(query);
};
```

4.1.2 Controladores

Interface genérico de controlador responsável pela receção e tratamento dos pedidos e pelo envio das respostas.

```
const GenericController = module.exports;

const isSqlInjection = require('is-sql-injection');
const connectOracle = require('../db');

GenericController.index = (table) => {
  if(isSqlInjection(table)) {
    return { error: 'possible sql injection detected' };
  }

  return connectOracle.exec(`SELECT * FROM ${table}`);
};
```



```

GenericController.filtered = (table, quantity) => {
  if(isSqlInjection(table) || isSqlInjection(quantity)) {
    return { error: 'possible sql injection detected' };
  }

  return connectOracle.exec(`SELECT * FROM ${table} FETCH FIRST ${quantity} ROWS ONLY`);
};

GenericController.raw = (query) => {
  return connectOracle.exec(query);
};

```

Exemplo de implementação do controlador genérico para o caso específico da Memória.

```

const Memory = module.exports;

const GenericController = require('./genericController');

Memory.index = () => {
  return GenericController.index('memory_state');
};

Memory.filtered = (unit, quantity) => {
  return GenericController.filtered(`memory_${unit}`, quantity);
};

```

4.2 Interface

Para interface foi usada outra framework de JavaScript, React, mas desta feita para Frontend. Esta aplicação tem dois principais componentes, o primeiro para a gestão do estado do lado do cliente, que é responsável por comunicar com a API referida anteriormente e fazer pedidos para receber os últimos dados do estado da base de dados. O segundo componente, prende-se com a renderização da interface com o utilizador em si, em que, baseado nos dados recebidos pelo componente anterior, gera a dashboard com gráficos do estado dos vários 'resources'.

4.3 Controlo de Estado

Interface genérico responsável pela comunicação para consulta de dados ao servidor em Node.js.

```

import axios from 'axios';

const host = process.env.REACT_APP_API_HOST;

// resource : type of data to fetch (p.e. 'cpu', 'datafiles', ...)
// unit      : type of data aggregation (p.e. 'daily', 'hourly', ...)
// quantity  : amount of data to fetch (p.e. '10', '100', '3123123', ... data points)

export const getData = async (resource, unit, quantity) => {
  const url = createUrl(resource, unit, quantity);

  return await axios.get(url);
};

function createUrl(resource, unit, quantity) {

```

```

let key = `${host}/${resource}`;

if (unit !== undefined) {
  key += `?unit=${unit}&quantity=${quantity}`;
}

return key;
}

```

Exemplo de implementação do interface para comunicação para o caso específico do CPU.

```

import { getData } from './genericState';

export const getCurrentCpu = () => {
  return getData('cpu');
};

export const getCpuHistory = (unit = 'daily', quantity = 7) => {
  return getData('cpu', unit, quantity);
};

```

4.4 Dashboards de apresentação de métricas

As dashboard foram desenvolvidas com base nos módulos `'react-layout-grid'` que permite a divisão do ecrã em várias secções e dá a possibilidade de o utilizador reorganizar a dashboard a seu gosto. E também no módulo `'react-chartjs-2'` um wrapper de React para a criação dos vários gráficos em HTML/JS.

4.5 Exemplos

Nesta secção iremos apresentar diversos exemplos da interface desenvolvida bem como explicar as decisões tomadas.

Tendo por base apresentar uma interface simples e intuitiva, decidimos dividir os parâmetros a avaliar por 3 categorias diferentes, por forma a facilitar a utilização desta: *Configuration*, onde estão as informações do *CPU* e *Memory*, *Storage* que contém os dados sobre *Tablespaces* e *Datafiles* e *Security* que contém as informações dos *Users* e *Sessions*.

4.5.1 CPU

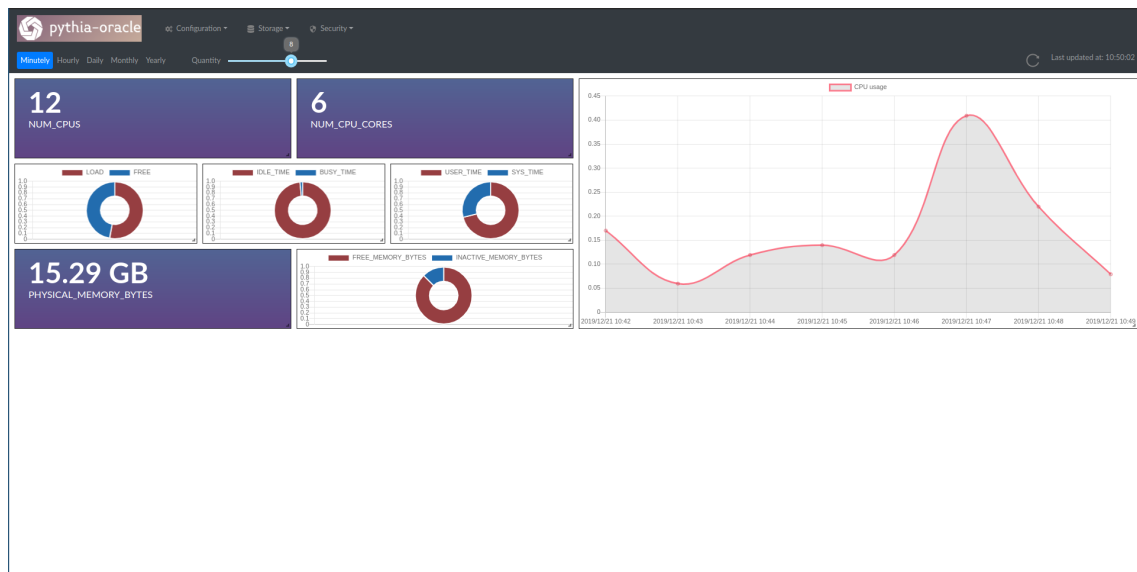


Figura 4: Dashboard de análise de métricas relacionadas com CPU e gráfico da evolução histórica da carga do mesmo

4.5.2 Memória

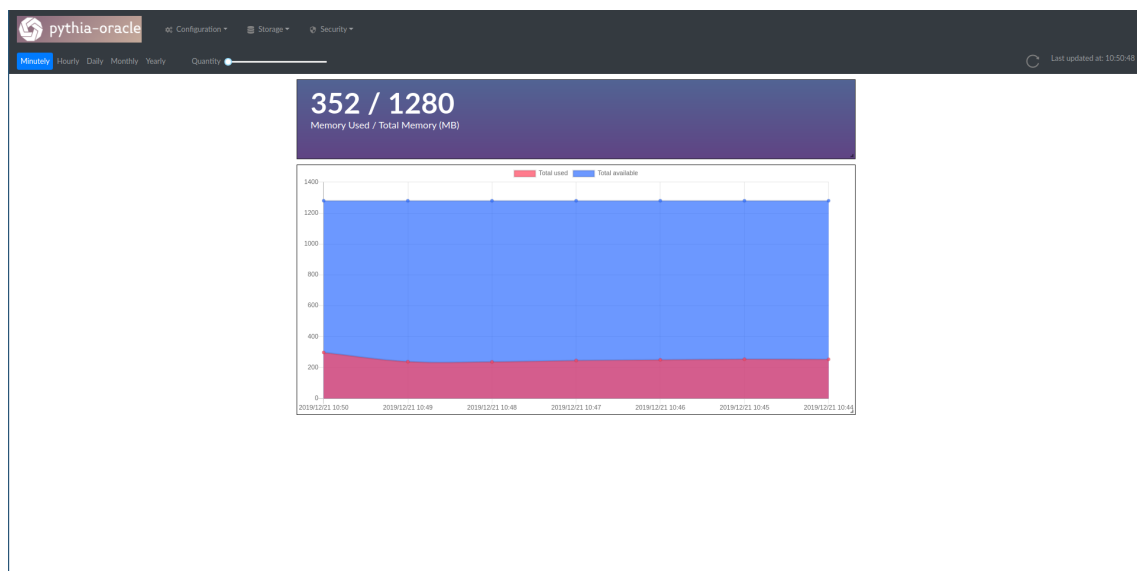


Figura 5: Métrica da memória do sistema total e a usada e gráfico com a sua evolução

4.5.3 Tablespaces

pythia-oracle

ConfigurationStorageSecurity

Last updated at: 10:51:03

TABLESPACE_NAME	BLOCK_SIZE	INITIAL_EXTENT	NEXT_EXTENT	MIN_EXTENTS	MAX_EXTENTS	MAX_SIZE	PCT_INCREASE	MIN_EXTLEN	STATUS	CONTENTS	LOGGING	FORCE_LOGGING	EXTENT_MANAGEMENT	ALLOCATION_TYPE	PLUGGED_IN	SEGMENT_SPACE_MANAGEMENT
SYSTEM	8192	65536		1	2147483645	2147483645		65536	ONLINE	PERMANENT	LOGGING	YES	LOCAL	SYSTEM	NO	MANUAL
SYSAUX	8192	65536		1	2147483645	2147483645		65536	ONLINE	PERMANENT	LOGGING	YES	LOCAL	SYSTEM	NO	AUTO
UNDOTBS1	8192	65536		1	2147483645	2147483645		65536	ONLINE	UNDO	LOGGING	NO	LOCAL	SYSTEM	NO	MANUAL
TEMP	8192	1048576	1048576	1		2147483645	0	1048576	ONLINE	TEMPORARY	NOLOGGING	NO	LOCAL	UNIFORM	NO	MANUAL
USERS	8192	65536		1	2147483645	2147483645		65536	ONLINE	PERMANENT	LOGGING	NO	LOCAL	SYSTEM	NO	AUTO
PYTHIA_ORACLE_TABLES	8192	65536		1	2147483645	2147483645		65536	ONLINE	PERMANENT	LOGGING	NO	LOCAL	SYSTEM	NO	AUTO

Figura 6: Listagem dos Tablespaces do sistema e das suas propriedades

4.5.4 Datafiles

pythia-oracle

ConfigurationStorageSecurity

Quantity

Last updated at: 10:51:21

FILE NAME	FILE ID	TABLESPACE_NAME	BYTES	BLOCKS	STATUS	RELATIVE_FNO	AUTOEXTENSIBLE	MAXBYTES	MAXBLOCKS	INCREMENT_BY	USER_BYTES	USER_BLOCKS	ONLINE STATUS	LOGS_WRITE_PROTECT
/u02/app/oracle/instant/ORCL/system01.dbf	1	SYSTEM	1132462080	138240	AVAILABLE	1	YES	34339721984	4194302	1280	1131411504	138112	SYSTEM	OFF
/u02/app/oracle/instant/ORCL/syaux01.dbf	3	SYSAUX	775946240	94720	AVAILABLE	3	YES	34339721984	4194302	1280	774897664	94192	ONLINE	OFF
/u02/app/oracle/instant/ORCL/undotbs01.dbf	12	UNDOTBS1	54525952	6656	AVAILABLE	12	YES	302544000	30000	128	53477276	6528	ONLINE	OFF
/u02/app/oracle/instant/ORCL/users01.dbf	7	USERS	5242880	640	AVAILABLE	7	YES	34339721984	4194302	160	4194304	512	ONLINE	OFF
/u01/app/oracle/product/12.2.0.8/flash_1/du01/app/oracle/instant/du01/pythia_oracle_01.dbf	21	PYTHIA_ORACLE_TABLES	104857600	12800	AVAILABLE	21	NO	0	0	0	103869024	128172	ONLINE	OFF

Figura 7: Listagem de todos os Datafiles

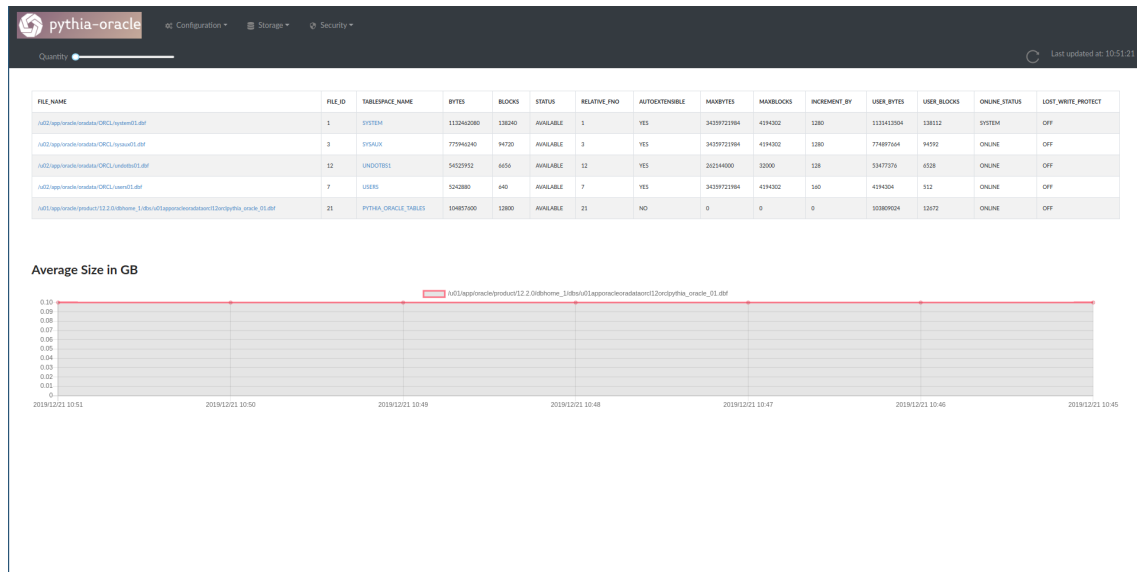


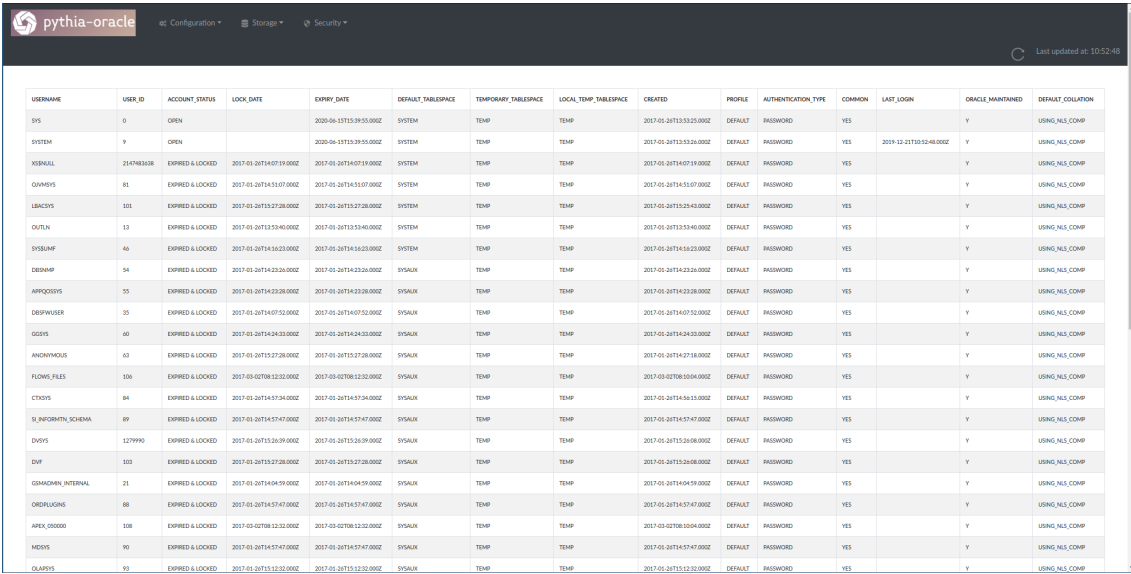
Figura 8: Apresentação de gráfico com evolução do tamanho quando é selecionado o nome do Datafile no exemplo anterior

The screenshot shows the 'pythia-oracle' interface displaying the properties of the 'PYTHIA_ORACLE_TABLES' tablespace.

TABLESPACE_NAME	BLOCK_SIZE	INITIAL_EXTENT	NEXT_EXTENT	MIN_EXTENTS	MAX_EXTENTS	MAX_SIZE	PCT_INCREASE	MIN_EXTLEN	STATUS	CONTENTS	LOGGING	FORCE_LOGGING	EXTENT_MANAGEMENT	ALLOCATION_TYPE	PLUGGED_IN	SEGMENT_SPACE_MANAGEMENT
PYTHIA_ORACLE_TABLES	8192	65536		1	2147483645	2147483645		65536	ONLINE	PERMANENT	LOGGING	NO	LOCAL	SYSTEM	NO	AUTO

Figura 9: Quando é clicado o link do Tablespace correspondente nos exemplos anterior o utilizador é redirecionado para as propriedades desse Tablespace

4.5.5 Utilizadores

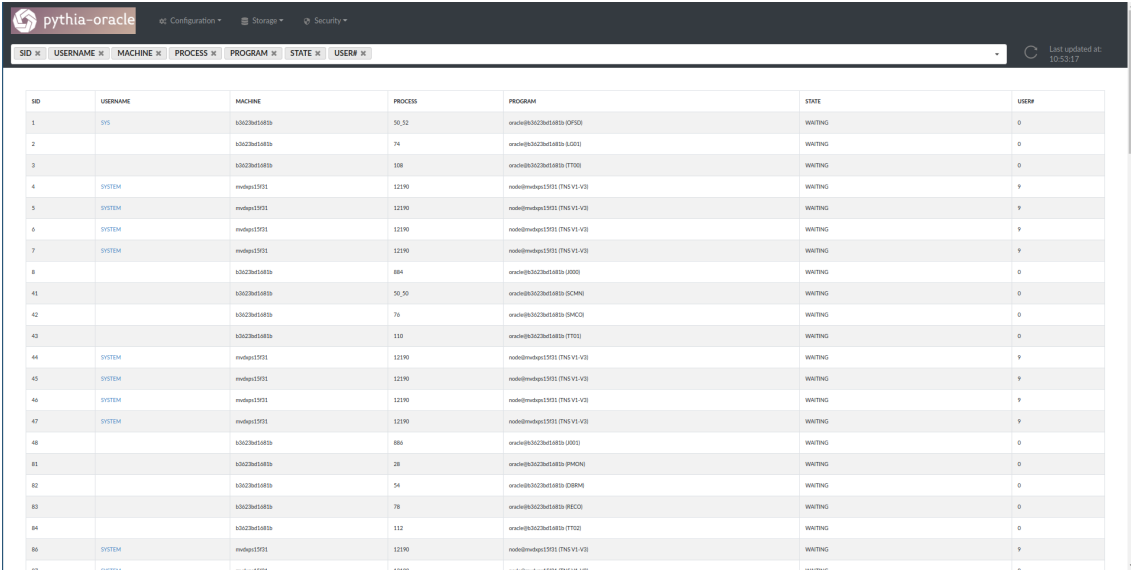


The screenshot shows the 'pythia-oracle' application interface. At the top, there are navigation tabs: 'Configuration', 'Storage', and 'Security'. The main content area displays a table with 15 columns: USERNAME, USER_ID, ACCOUNT_STATUS, LOCK_DATE, EXPIRY_DATE, DEFAULT_TABLESPACE, TEMPORARY_TABLESPACE, LOCAL_TEMP_TABLESPACE, CREATED, PROFILE, AUTHENTICATION_TYPE, COMMON, LAST_LOGIN, ORACLE_MAINTAINED, and DEFAULT_COLLATION. The table lists various system users, including SYS, SYSTEM, XS\$NULL, QMAN\$SYS, LBAC\$SYS, OUTLN, SYSMAN, DBSNMP, APPQOSSYS, DESKVIEWER, ODSYS, ANONYMOUS, FLOWC_FILES, CTXSYS, S_INFOMATH_SCHEMA, DVSYS, DWP, GPMACHAN_INTERNAL, OKEDPLUGINS, APEX_00000, MDSYS, and CLAPSYS. Each row provides details about the user's status, dates, tablespaces, and other system attributes.

USERNAME	USER_ID	ACCOUNT_STATUS	LOCK_DATE	EXPIRY_DATE	DEFAULT_TABLESPACE	TEMPORARY_TABLESPACE	LOCAL_TEMP_TABLESPACE	CREATED	PROFILE	AUTHENTICATION_TYPE	COMMON	LAST_LOGIN	ORACLE_MAINTAINED	DEFAULT_COLLATION
SYS	0	OPEN		2020-06-13T15:39:51.000Z	SYSTEM	TEMP	TEMP	2017-02-28T13:53:25.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
SYSTEM	9	OPEN		2020-06-13T15:39:51.000Z	SYSTEM	TEMP	TEMP	2017-02-28T13:53:26.000Z	DEFAULT	PASSWORD	YES	2019-12-21T10:52:46.000Z	Y	USING_NLS_COMP
XS\$NULL	2147483648	EXPIRED & LOCKED	2017-02-28T14:07:19.000Z	2017-02-28T14:07:19.000Z	SYSTEM	TEMP	TEMP	2017-02-28T14:07:19.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
QMAN\$SYS	81	EXPIRED & LOCKED	2017-02-28T14:51:07.000Z	2017-02-28T14:51:07.000Z	SYSTEM	TEMP	TEMP	2017-02-28T14:51:07.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
LBAC\$SYS	101	EXPIRED & LOCKED	2017-02-28T15:27:28.000Z	2017-02-28T15:27:28.000Z	SYSTEM	TEMP	TEMP	2017-02-28T15:27:28.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
OUTLN	13	EXPIRED & LOCKED	2017-02-28T13:53:40.000Z	2017-02-28T13:53:40.000Z	SYSTEM	TEMP	TEMP	2017-02-28T13:53:40.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
SYSMAN	46	EXPIRED & LOCKED	2017-02-28T14:56:23.000Z	2017-02-28T14:56:23.000Z	SYSTEM	TEMP	TEMP	2017-02-28T14:56:23.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
DBSNMP	54	EXPIRED & LOCKED	2017-02-28T14:23:26.000Z	2017-02-28T14:23:26.000Z	SYSTEM	TEMP	TEMP	2017-02-28T14:23:26.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
APPQOSSYS	55	EXPIRED & LOCKED	2017-02-28T14:23:28.000Z	2017-02-28T14:23:28.000Z	SYSTEM	TEMP	TEMP	2017-02-28T14:23:28.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
DESKVIEWER	35	EXPIRED & LOCKED	2017-02-28T14:07:52.000Z	2017-02-28T14:07:52.000Z	SYSTEM	TEMP	TEMP	2017-02-28T14:07:52.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
ODSYS	60	EXPIRED & LOCKED	2017-02-28T14:24:33.000Z	2017-02-28T14:24:33.000Z	SYSTEM	TEMP	TEMP	2017-02-28T14:24:33.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
ANONYMOUS	63	EXPIRED & LOCKED	2017-02-28T15:27:28.000Z	2017-02-28T15:27:28.000Z	SYSTEM	TEMP	TEMP	2017-02-28T15:27:28.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
FLOWC_FILES	106	EXPIRED & LOCKED	2017-02-28T16:12:32.000Z	2017-02-28T16:12:32.000Z	SYSTEM	TEMP	TEMP	2017-02-28T16:12:32.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
CTXSYS	84	EXPIRED & LOCKED	2017-02-28T14:57:34.000Z	2017-02-28T14:57:34.000Z	SYSTEM	TEMP	TEMP	2017-02-28T14:57:34.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
S_INFOMATH_SCHEMA	89	EXPIRED & LOCKED	2017-02-28T14:57:47.000Z	2017-02-28T14:57:47.000Z	SYSTEM	TEMP	TEMP	2017-02-28T14:57:47.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
DVSYS	127990	EXPIRED & LOCKED	2017-02-28T15:26:59.000Z	2017-02-28T15:26:59.000Z	SYSTEM	TEMP	TEMP	2017-02-28T15:26:59.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
DWP	103	EXPIRED & LOCKED	2017-02-28T15:27:28.000Z	2017-02-28T15:27:28.000Z	SYSTEM	TEMP	TEMP	2017-02-28T15:27:28.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
GPMACHAN_INTERNAL	21	EXPIRED & LOCKED	2017-02-28T14:54:59.000Z	2017-02-28T14:54:59.000Z	SYSTEM	TEMP	TEMP	2017-02-28T14:54:59.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
OKEDPLUGINS	86	EXPIRED & LOCKED	2017-02-28T14:57:47.000Z	2017-02-28T14:57:47.000Z	SYSTEM	TEMP	TEMP	2017-02-28T14:57:47.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
APEX_00000	108	EXPIRED & LOCKED	2017-02-28T16:12:32.000Z	2017-02-28T16:12:32.000Z	SYSTEM	TEMP	TEMP	2017-02-28T16:12:32.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
MDSYS	90	EXPIRED & LOCKED	2017-02-28T14:57:47.000Z	2017-02-28T14:57:47.000Z	SYSTEM	TEMP	TEMP	2017-02-28T14:57:47.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP
CLAPSYS	93	EXPIRED & LOCKED	2017-02-28T15:12:32.000Z	2017-02-28T15:12:32.000Z	SYSTEM	TEMP	TEMP	2017-02-28T15:12:32.000Z	DEFAULT	PASSWORD	YES		Y	USING_NLS_COMP

Figura 10: Listagem dos utilizadores do sistema e suas propriedades (p.e. DEFAULT_TABLESPACE)

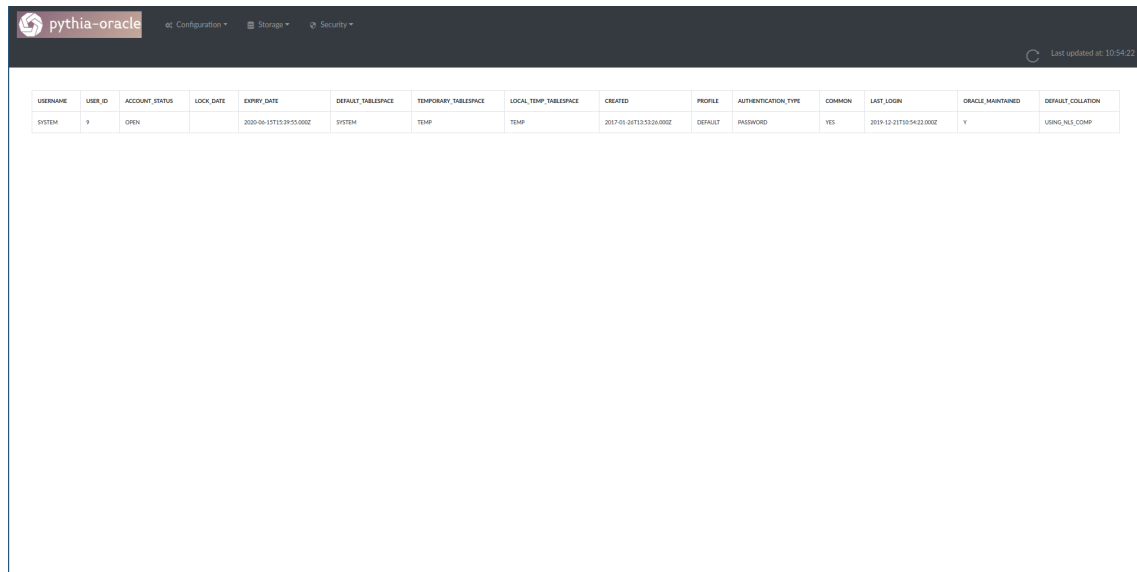
4.5.6 Sessões



The screenshot shows the 'pythia-oracle' application interface with a filter bar at the top. The filter bar includes tabs for 'SID', 'USERNAME', 'MACHINE', 'PROCESS', 'PROGRAM', 'STATE', and 'USER#'. The main content area displays a table with 7 columns: SID, USERNAME, MACHINE, PROCESS, PROGRAM, STATE, and USER#. The table lists various sessions, including those for SYS, SYSTEM, and other users. Each row provides details about the session's ID, user, machine, process, program, state, and user number.

SID	USERNAME	MACHINE	PROCESS	PROGRAM	STATE	USER#
1	SYS	136229451809	56_32	oracle@136229451809 (SYS)	WAITING	0
2		136229451809	74	oracle@136229451809 (SYS)	WAITING	0
3		136229451809	108	oracle@136229451809 (TT02)	WAITING	0
4	SYSTEM	mdwp@136229451809	12190	node@mdwp@136229451809 (SYS V1-V5)	WAITING	9
5	SYSTEM	mdwp@136229451809	12190	node@mdwp@136229451809 (SYS V1-V5)	WAITING	9
6	SYSTEM	mdwp@136229451809	12190	node@mdwp@136229451809 (SYS V1-V5)	WAITING	9
7	SYSTEM	mdwp@136229451809	12190	node@mdwp@136229451809 (SYS V1-V5)	WAITING	9
8		136229451809	884	oracle@136229451809 (SYS)	WAITING	0
41		136229451809	50_50	oracle@136229451809 (SYS)	WAITING	0
42		136229451809	76	oracle@136229451809 (SYS)	WAITING	0
43		136229451809	110	oracle@136229451809 (TT02)	WAITING	0
44	SYSTEM	mdwp@136229451809	12190	node@mdwp@136229451809 (SYS V1-V5)	WAITING	9
45	SYSTEM	mdwp@136229451809	12190	node@mdwp@136229451809 (SYS V1-V5)	WAITING	9
46	SYSTEM	mdwp@136229451809	12190	node@mdwp@136229451809 (SYS V1-V5)	WAITING	9
47	SYSTEM	mdwp@136229451809	12190	node@mdwp@136229451809 (SYS V1-V5)	WAITING	9
48		136229451809	886	oracle@136229451809 (SYS)	WAITING	0
81		136229451809	28	oracle@136229451809 (SYS)	WAITING	0
82		136229451809	54	oracle@136229451809 (SYS)	WAITING	0
83		136229451809	78	oracle@136229451809 (SYS)	WAITING	0
84		136229451809	112	oracle@136229451809 (TT02)	WAITING	0
86	SYSTEM	mdwp@136229451809	12190	node@mdwp@136229451809 (SYS V1-V5)	WAITING	9

Figura 11: Listagem de todas as sessões ativas e inativas no sistema, com hipótese de escolher quais as propriedades a serem apresentadas



The screenshot shows a web application titled "pythia-oracle" with a dark header. The header contains navigation links for "Configuration", "Storage", and "Security". On the right, it says "Last updated at: 10:54:22". Below the header is a table with 14 columns: USERNAME, USER_ID, ACCOUNT_STATUS, LOCK_DATE, EXPIRY_DATE, DEFAULT_TABLESPACE, TEMPORARY_TABLESPACE, LOCAL_TEMP_TABLESPACE, CREATED, PROFILE, AUTHENTICATION_TYPE, COMMON, LAST_LOGIN, ORACLE_MAINTAINED, and DEFAULT_COLLATION. The table contains one row for the "SYSTEM" user.

USERNAME	USER_ID	ACCOUNT_STATUS	LOCK_DATE	EXPIRY_DATE	DEFAULT_TABLESPACE	TEMPORARY_TABLESPACE	LOCAL_TEMP_TABLESPACE	CREATED	PROFILE	AUTHENTICATION_TYPE	COMMON	LAST_LOGIN	ORACLE_MAINTAINED	DEFAULT_COLLATION
SYSTEM	9	OPEN		2020-06-17T13:39:25.000Z	SYSTEM	TEMP	TEMP	2017-01-24T13:53:20.000Z	DEFAULT	PASSWORD	YES	2019-12-23T05:54:22.000Z	Y	USING_NLS_COMP

Figura 12: No exemplo anterior, cada sessão tem um link no utilizador correspondente, quando clica nesse link o utilizador é redirecionado para ver os dados desse utilizador

5 Conclusão

Através de uma arquitetura baseada na recolha e preenchimento de dados diretamente na base de dados, através da execução automática de um *procedure* para a atualização dos dados históricos, e da criação de views para a consulta estatística do mesmos é garantida a qualidade e correta atualização do histórico. Deste modo, permitiu-se também desenvolver um servidor de dados muito simples, pouco dispendioso de implementar e em termos computacionais, que reduz a carga criada pelo seu uso na base de dados. A interface também ela é gerada do lado do cliente e por isso também barata em termos de servidor.

Mesmo com estas preocupações em mente, o monitor é capaz de servir e apresentar de modo fácil, intuitivo e útil todos os dados importante em termos da monitorização da evolução e performance da base de dados, no que diz respeito a métricas como CPU, memória, tablespaces, datafiles, utilizadores e sessões.

6 Bibliografia

Referências

- [1] Monitoring and Tuning the Database,
https://docs.oracle.com/cd/B19306_01/server.102/b14196/montune.htm#ADMQS010
- [2] Oracle Database 12c: The Complete Reference
Bob Bryla, Kevin Loney. McGraw-Hill Education; 1 edition (October 2, 2013)
- [3] Discussão sobre *Oracle Database*,
<https://community.oracle.com/>
- [4] Documentação da framework React,
<https://reactjs.org/docs>

7 Anexos

7.1 Criação de Tablespace

```
create tablespace pythia_oracle_tables datafile
'\u01\app\oracle\oradata\orcl12\orcl\pythia_oracle_01.dbf'
size 100M;

alter database default tablespace pythia_oracle_tables;

select PROPERTY_VALUE
from DATABASE_PROPERTIES
where PROPERTY_NAME = 'DEFAULT_PERMANENT_TABLESPACE';
```

7.2 Criação do utilizador Monitor

```
CREATE USER monitor IDENTIFIED BY 123456
    default tablespace pythia_oracle_tables
    quota unlimited on pythia_oracle_tables;

GRANT CONNECT TO monitor;

GRANT CREATE SESSION TO monitor;

-- as sysdba
GRANT SELECT ON V_$OSSTAT TO monitor;
GRANT SELECT ON V_$SGA TO monitor;
GRANT SELECT ON V_$SESSION TO monitor;
GRANT SELECT ON V_$PROCESS TO monitor;
GRANT SELECT ON DBA_DATA_FILES TO monitor;
GRANT SELECT ON DBA_TABLESPACES TO monitor;
GRANT SELECT ON DBA_USERS TO monitor;
```

7.3 Criação de tabelas

```
-- CPU
create table cpu_history (
    stat_name varchar2(50),
    value number,
    query_date TIMESTAMP
);

-- Memory
create table memory_history (
    "TOTAL (MB)" number,
    "USED (MB)" number,
    query_date TIMESTAMP
);

-- Datafiles
create table datafiles_history (
    file_name varchar2(150),
    tablespace_name varchar2(150),
    bytes number,
```

```

        maxbytes number,
        status varchar2(50),
        autoextensible varchar2(3),
        query_date TIMESTAMP
    );

-- Tablespaces
create table tablespaces_history (
    tablespace_name varchar2(150),
    block_size number,
    initial_extent number,
    next_extent number,
    min_extents number,
    max_extents number,
    min_extlen number,
    status varchar2(50),
    contents varchar2(50),
    query_date TIMESTAMP
);

-- Users
create table users (
    username varchar2(150),
    user_id number,
    account_status varchar2(45),
    default_tablespace varchar2(150)
);

-- Sessions
create table sessions (
    sid number,
    username varchar2(150),
    status varchar2(45),
    program varchar2(45),
    "type" varchar2(45)
);

```

7.4 Criação das views do histórico do CPU

```

create or replace view cpu_minutely as
    select
        STAT_NAME,
        avg(VALUE) as avg_value,
        TO_CHAR(query_date, 'YYYY/MM/DD HH24:MI') as query_date
    from cpu_history
    group by STAT_NAME, TO_CHAR(query_date, 'YYYY/MM/DD HH24:MI')
    order by query_date desc;

create or replace view cpu_hourly as
    select
        STAT_NAME,
        avg(VALUE) as avg_value,
        TO_CHAR(query_date, 'YYYY/MM/DD HH24') as query_date
    from cpu_history

```

```

        group by STAT_NAME, TO_CHAR(query_date, 'YYYY/MM/DD HH24')
        order by query_date desc;

create or replace view cpu_daily as
select
    STAT_NAME,
    avg(VALUE) as avg_value,
    TO_CHAR(query_date, 'YYYY/MM/DD') as query_date
from cpu_history
group by STAT_NAME, TO_CHAR(query_date, 'YYYY/MM/DD')
order by query_date desc;

create or replace view cpu_monthly as
select
    STAT_NAME,
    avg(VALUE) as avg_value,
    TO_CHAR(query_date, 'YYYY/MM') as query_date
from cpu_history
group by STAT_NAME, TO_CHAR(query_date, 'YYYY/MM')
order by query_date desc;

create or replace view cpu_yearly as
select
    STAT_NAME,
    avg(VALUE) as avg_value,
    TO_CHAR(query_date, 'YYYY') as query_date
from cpu_history
group by STAT_NAME, TO_CHAR(query_date, 'YYYY')
order by query_date desc;

```

7.5 Criação das views do histórico da Memória

```

create or replace view memory_state as
select T as "TOTAL (MB)", U as "USED (MB)"
from dual
inner join (
    select sum(pga_max_mem)/1024/1024 "U", 'X' "DUMMY"
    from v$process
) "A" on dual.DUMMY = "A".DUMMY
inner join (
    select sum(value)/1024/1024 "T", 'X' "DUMMY"
    from v$sga
) "B" on dual.DUMMY = "B".DUMMY
;

create or replace view memory_minutely as
select
    avg("TOTAL (MB)") as "AVERAGE TOTAL (MB)",
    avg("USED (MB)") as "AVERAGE USED (MB)",
    TO_CHAR(query_date, 'YYYY/MM/DD HH24:MI') as query_date
from memory_history
group by TO_CHAR(query_date, 'YYYY/MM/DD HH24:MI')
order by query_date desc;

```

```

create or replace view memory_hourly as
select
    avg("TOTAL (MB)") as "AVERAGE TOTAL (MB)",
    avg("USED (MB)") as "AVERAGE USED (MB)",
    TO_CHAR(query_date, 'YYYY/MM/DD HH24') as query_date
from memory_history
group by TO_CHAR(query_date, 'YYYY/MM/DD HH24')
order by query_date desc;

```

```

create or replace view memory_daily as
select
    avg("TOTAL (MB)") as "AVERAGE TOTAL (MB)",
    avg("USED (MB)") as "AVERAGE USED (MB)",
    TO_CHAR(query_date, 'YYYY/MM/DD') as query_date
from memory_history
group by TO_CHAR(query_date, 'YYYY/MM/DD')
order by query_date desc;

```

```

create or replace view memory_monthly as
select
    avg("TOTAL (MB)") as "AVERAGE TOTAL (MB)",
    avg("USED (MB)") as "AVERAGE USED (MB)",
    TO_CHAR(query_date, 'YYYY/MM') as query_date
from memory_history
group by TO_CHAR(query_date, 'YYYY/MM')
order by query_date desc;

```

```

create or replace view memory_yearly as
select
    avg("TOTAL (MB)") as "AVERAGE TOTAL (MB)",
    avg("USED (MB)") as "AVERAGE USED (MB)",
    TO_CHAR(query_date, 'YYYY') as query_date
from memory_history
group by TO_CHAR(query_date, 'YYYY')
order by query_date desc;

```

7.6 Criação das views do histórico dos Datafiles

```

create or replace view datafiles_minutely as
select
    FILE_NAME,
    avg(BYTES) as avg_bytes,
    avg(MAXBYTES) as avg_maxbytes,
    TO_CHAR(query_date, 'YYYY/MM/DD HH24:MI') as query_date
from datafiles_history
group by FILE_NAME, TO_CHAR(query_date, 'YYYY/MM/DD HH24:MI')
order by query_date desc;

```

```

create or replace view datafiles_hourly as
select
    FILE_NAME,
    avg(BYTES) as avg_bytes,
    avg(MAXBYTES) as avg_maxbytes,
    TO_CHAR(query_date, 'YYYY/MM/DD HH24') as query_date

```

```

from datafiles_history
group by FILE_NAME, TO_CHAR(query_date, 'YYYY/MM/DD HH24')
order by query_date desc;

create or replace view datafiles_daily as
select
    FILE_NAME,
    avg(BYTES) as avg_bytes,
    avg(MAXBYTES) as avg_maxbytes,
    TO_CHAR(query_date, 'YYYY/MM/DD') as query_date
from datafiles_history
group by FILE_NAME, TO_CHAR(query_date, 'YYYY/MM/DD')
order by query_date desc;

create or replace view datafiles_monthly as
select
    FILE_NAME,
    avg(BYTES) as avg_bytes,
    avg(MAXBYTES) as avg_maxbytes,
    TO_CHAR(query_date, 'YYYY/MM') as query_date
from datafiles_history
group by FILE_NAME, TO_CHAR(query_date, 'YYYY/MM')
order by query_date desc;

create or replace view datafiles_yearly as
select
    FILE_NAME,
    avg(BYTES) as avg_bytes,
    avg(MAXBYTES) as avg_maxbytes,
    TO_CHAR(query_date, 'YYYY') as query_date
from datafiles_history
group by FILE_NAME, TO_CHAR(query_date, 'YYYY')
order by query_date desc;

```

7.7 Criação das views do histórico dos Tablespaces

```

create or replace view tablespaces_minutely as
select
    TABLESPACE_NAME,
    avg(BLOCK_SIZE) as avg_block_size,
    avg(INITIAL_EXTENT) as avg_initial_extent,
    avg(NEXT_EXTENT) as avg_next_extent,
    avg(MIN_EXTENTS) as avg_min_extents,
    avg(MAX_EXTENTS) as avg_max_extents,
    avg(MIN_EXTLEN) as avg_min_extlen,
    TO_CHAR(query_date, 'YYYY/MM/DD HH24:MI') as query_date
from tablespaces_history
group by TABLESPACE_NAME, TO_CHAR(query_date, 'YYYY/MM/DD HH24:MI')
order by query_date desc;

create or replace view tablespaces_hourly as
select
    TABLESPACE_NAME,
    avg(BLOCK_SIZE) as avg_block_size,

```

```

        avg(INITIAL_EXTENT) as avg_initial_extent,
        avg(NEXT_EXTENT) as avg_next_extent,
        avg(MIN_EXTENTS) as avg_min_extents,
        avg(MAX_EXTENTS) as avg_max_extents,
        avg(MIN_EXTLEN) as avg_min_extlen,
        TO_CHAR(query_date, 'YYYY/MM/DD HH24') as query_date
    from tablespaces_history
    group by TABLESPACE_NAME, TO_CHAR(query_date, 'YYYY/MM/DD HH24')
    order by query_date desc;

create or replace view tablespaces_daily as
select
    TABLESPACE_NAME,
    avg(BLOCK_SIZE) as avg_block_size,
    avg(INITIAL_EXTENT) as avg_initial_extent,
    avg(NEXT_EXTENT) as avg_next_extent,
    avg(MIN_EXTENTS) as avg_min_extents,
    avg(MAX_EXTENTS) as avg_max_extents,
    avg(MIN_EXTLEN) as avg_min_extlen,
    TO_CHAR(query_date, 'YYYY/MM/DD') as query_date
from tablespaces_history
group by TABLESPACE_NAME, TO_CHAR(query_date, 'YYYY/MM/DD')
order by query_date desc;

create or replace view tablespaces_monthly as
select
    TABLESPACE_NAME,
    avg(BLOCK_SIZE) as avg_block_size,
    avg(INITIAL_EXTENT) as avg_initial_extent,
    avg(NEXT_EXTENT) as avg_next_extent,
    avg(MIN_EXTENTS) as avg_min_extents,
    avg(MAX_EXTENTS) as avg_max_extents,
    avg(MIN_EXTLEN) as avg_min_extlen,
    TO_CHAR(query_date, 'YYYY/MM') as query_date
from tablespaces_history
group by TABLESPACE_NAME, TO_CHAR(query_date, 'YYYY/MM')
order by query_date desc;

create or replace view tablespaces_yearly as
select
    TABLESPACE_NAME,
    avg(BLOCK_SIZE) as avg_block_size,
    avg(INITIAL_EXTENT) as avg_initial_extent,
    avg(NEXT_EXTENT) as avg_next_extent,
    avg(MIN_EXTENTS) as avg_min_extents,
    avg(MAX_EXTENTS) as avg_max_extents,
    avg(MIN_EXTLEN) as avg_min_extlen,
    TO_CHAR(query_date, 'YYYY') as query_date
from tablespaces_history
group by TABLESPACE_NAME, TO_CHAR(query_date, 'YYYY')
order by query_date desc;

```

7.8 Criação do job de atualização do histórico

7.8.1 Procedure para atualização

```
create or replace procedure update_values
is
begin
    insert into cpu_history
    select STAT_NAME, VALUE, CURRENT_TIMESTAMP
    from V$OSSTAT;

    insert into memory_history
    select "TOTAL (MB)", "USED (MB)", CURRENT_TIMESTAMP
    from memory_state;

    insert into datafiles_history
    select
        FILE_NAME, TABLESPACE_NAME, BYTES, MAXBYTES,
        STATUS, AUTOEXTENSIBLE, CURRENT_TIMESTAMP
    from DBA_DATA_FILES;

    insert into tablespaces_history
    select
        TABLESPACE_NAME, BLOCK_SIZE, INITIAL_EXTENT, NEXT_EXTENT,
        MIN_EXTENTS, MAX_EXTENTS, MIN_EXTLEN, STATUS, CONTENTS,
        CURRENT_TIMESTAMP
    from DBA_TABLESPACES;
end;
```

7.8.2 Agendamento do procedure update_values procedure para ser executado a cada minuto

```
BEGIN
    DBMS_SCHEDULER.CREATE_JOB (
        job_name          => 'update_values_job',
        job_type           => 'STORED_PROCEDURE',
        job_action          => 'update_values',
        start_date          => CURRENT_TIMESTAMP,
        repeat_interval     => 'FREQ=MINUTELY;',
        enabled             => true
    );
END;
```