



Laboratório de Bases de Dados

Prof. José Fernando Rodrigues Júnior

Aula 5

Dicionário de dados e Desenvolvimento de Aplicações





Dicionário de dados Oracle

- Informações (metadados) que descrevem os objetos do banco e sua estrutura
- Dois níveis:
 - **Interno:** metadados completos, acessível apenas pelo gerenciador Oracle para que o sistema opere
 - **Externo:** visões com diferentes níveis de detalhes criadas sobre o dicionário interno, acessível aos usuários finais
- Informações:
 - Usuários
 - Tabelas, colunas e seus tipos, restrições de integridade, índices, ...
 - Estatísticas sobre as tabelas e índices – usadas pelo otimizador
 - Privilégios de uso dos objetos
 - Estruturas de armazenamento do banco



Dicionário de dados Oracle

- **SELECT * FROM DICT:** todas as tabelas e visões que constituem o dicionário
- **SELECT * FROM TAB:** todas as tabelas pertencentes a um dado usuário
- **SELECT * FROM COL:** todas as colunas pertencentes a um dado usuário
- **SELECT * FROM OBJ:** todas os objetos pertencentes a um dado usuário
- **SELECT * FROM USER_TAB_PRIVS:** privilégios concedidos a outros usuários

→ O próprio SGBD Oracle realiza várias consultas a estes dados para resolver qualquer consulta de um dado usuário. Por exemplo, a consulta:

SELECT * FROM EMPREGADO WHERE SALARIO > 1000

requer verificar: **(1)** se a tabela EMPREGADO existe; **(2)** se o usuário tem privilégio de uso desta tabela – ou se é dono; **(3)** se a coluna SALARIO existe nesta tabela

→ Como estes dados são usados frequentemente, o SGBD faz cache destas informações mantendo-as na memória após o primeiro uso



Dicionário de dados Oracle

■ Existem mais de 500 tabelas e visões no nível externo do dicionário de dados, divididas em 3 grupos, **USER_**, **ALL_**, e **DBA_**:

- **USER_** : informações referentes apenas aos objetos pertencentes ao usuário que faz a consulta
- **USER_TABLES** : dados completos de suas tabelas
- **USER_CATALOG**: suas tabelas, visões, e sinônimos
- **USER_COL_COMMENTS**: comentários sobre suas colunas
- **USER_TAB_COMMENTS**: comentários sobre suas tabelas
- **USER_INDEXES**: seus índices criados
- **USER_OBJECTS**: seus objetos
- **USER_TAB_COLUMNS**: suas colunas
- **USER_TRIGGERS**: seus gatilhos
- **USER_VIEWS**: suas visões
- **USER_USERS**: informação sobre o próprio usuário
- **USER_CONSTRAINTS** : constraints do usuário
- **USER_CONS_COLUMNS** : colunas das constraints
- ...



Comentários

- **Recurso para armazenar informação documental sobre os objetos do banco**

```
COMMENT ON object_type nome_objeto IS 'comentário';
```

➔ **Exemplo:**

```
COMMENT ON TABLE LE08CANDIDATO IS 'Lista de corruptos'
```

```
COMMENT ON COLUMN LE08CANDIDATO.NROCAND IS 'Chave da  
tabela de corruptos'
```

➔ **Documentação do esquema no próprio banco**

```
SELECT * FROM USER_TAB_COMMENTS
```

```
SELECT * FROM USER_COL_COMMENTS
```



Dicionário de dados Oracle

- **ALL_:** informações referentes aos objetos pertencentes ao usuário que faz a consulta, e sobre os objetos para os quais tem privilégio de acesso
 - **ALL_TABLES** : dados completos de todas as tabelas
 - **ALL_CATALOG:** todas as tabelas, visões, e sinônimos
 - **ALL_COL_COMMENTS:** todos os comentários sobre as colunas
 - **ALL_TAB_COMMENTS:** todos os comentários sobre as tabelas
 - **ALL_INDEXES:** todos os índices criados
 - **ALL_OBJECTS:** todos os objetos
 - **ALL_TAB_COLUMNS:** todas as colunas
 - **ALL_TRIGGERS:** todos os gatilhos
 - **ALL_VIEWS:** todas as visões
 - ...



Dicionário de dados Oracle

- **DBA_**: informações referentes a todos os objetos do banco, acessível apenas a usuários com privilégio de DBA
 - **DBA_TABLES** : dados completos de todas as tabelas do banco
 - **DBA_CATALOG**: todas as tabelas, visões, e sinônimos
 - **DBA_COL_COMMENTS**: todos os comentários sobre as colunas do banco
 - **DBA_TAB_COMMENTS**: todos os comentários sobre as tabelas do banco
 - **DBA_INDEXES**: todos os índices criados no banco
 - **DBA_OBJECTS**: todos os objetos do banco
 - **DBA_TAB_COLUMNS**: todas as colunas do banco
 - **DBA_TRIGGERS**: todos os gatilhos do banco
 - **DBA_VIEWS**: todas as visões do banco
 - ...



Dicionário de dados Oracle

Exemplo, dados de um objeto específico:

```
SELECT OBJECT_NAME, OBJECT_TYPE, CREATED, STATUS  
FROM OBJ  
WHERE OBJECT_NAME = 'LE08CANDIDATO'
```




Dicionário de dados Oracle

Exemplo, privilégios concedidos aos objetos de um usuário, não incluindo privilégios aos seus próprios objetos:

```
SELECT TABLE_NAME, GRANTEE, PRIVILEGE  
FROM USER_TAB_PRIVS
```



Conectando-se ao Oracle

- A conexão com o Oracle se dá por meio de uma ***string de conexão***
- Uma string de conexão é um parâmetro que indica de maneira estruturada, **quais são os parâmetros necessários para se conectar a um banco de dados**
- Para cada banco de dados, há uma **string de conexão diferente** cuja sintaxe varia de acordo com a linguagem de programação



Conectando-se ao Oracle

- Em **PostgreSQL**, por exemplo, a string deve ter no mínimo: usuário, senha, host, porta, e nome do banco
- Em **Oracle**, a string de conexão deve possuir no mínimo: usuário, senha, host, porta, e nome do serviço (instância do banco). Exemplo em Java:

```
DriverManager.getConnection("jdbc:oracle:thin:@192.168.183.15:1521:orcl",  
                             "teste", "teste");
```

➔ Mais exemplos em: <http://www.connectionstrings.com/>



Conectando-se ao Oracle

- Em Java, a conexão completa se dá da seguinte maneira:

```
Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
/*requer ojdbc14.jar*/
```

```
Connection connection = DriverManager.getConnection(  
    "jdbc:oracle:thin:@192.168.183.15:1521:orcl",  
    "usuario",  
    "senha");
```



Conectando-se ao Oracle

- A instância do objeto **Connection** passa a ser o **canal de comunicação** com o banco por meio da criação de instâncias de objetos **Statement**. Como no exemplo de seleção de dados:

```
Statement stmt = connection.createStatement();  
ResultSet rs = stmt.executeQuery("SELECT * FROM LE08CANDIDATO");  
    while (rs.next()) {  
        System.out.println(  
            rs.getString("NROCAND") + "-"  
            + rs.getString("NOME") + "-"  
            + rs.getString("IDADE")  
        );  
    }
```



Conectando-se ao Oracle

- Exemplo de inserção de dados:

```
PreparedStatement pstmt = connection.prepareStatement("INSERT INTO  
LE03ZONA VALUES(100, 50)");  
  
try{  
    pstmt.executeUpdate();  
} catch (SQLException e) {
```

- Exemplo completo em Tidia->Repositorio



Recuperação estruturada de metadados



Recuperação estruturada de metadados

- O Oracle armazena os metadados fornecidos por meio das instruções DDL usadas na definição dos objetos
- No entanto, **as próprias instruções DDL não são armazenadas**
- Estas instruções podem ser necessárias, caso:
 - deseje-se exportar o esquema;
 - as instruções DDL tenham sido perdidas;
 - as instruções DDL não tenham sido escritas, pois o esquema foi definido via interface gráfica.
- Ainda, pode ser necessário usar os dados em um modelo não relacional, como o XML



Recuperação estruturada de metadados

- Para este propósito o Oracle oferece o pacote de funções ***dbms_metadata***, uma API que lê o dicionário de dados e **reconstrói** os comandos usados na definição do esquema
- O pacote permite a geração de comandos **DDL** e também a exportação do esquema segundo o modelo de dados semi-estruturado **XML**; seus principais comandos são:
 - **GET_DDL**
 - **GET_XML**
 - **SET_TRANSFORM_PARAM**: permite formatar a saída de dados



Recuperação estruturada de metadados

- Sintax básica:

```
dbms_metadata.get_ddl(  
    'tipo_objeto',  
    'nome_objeto',  
    'nome_esquema');
```

➔ Os parâmetros são sensíveis ao caso!

- Exemplo:

```
SELECT  
dbms_metadata.get_ddl('TABLE', 'LE08CANDIDATO', 'TESTE')  
FROM DUAL;
```



Recuperação estruturada de metadados

- A geração padrão inclui informações de segmentos e de armazenamento, nem sempre úteis; para uma saída de dados mais amigável, pode-se parametrizar o pacote dbms_metadata da seguinte maneira:

```
/*omitir metadados de armazenamento*/
```

```
EXEC
```

```
dbms_metadata.set_transform_param(dbms_metadata.session_transform,'STORAGE',false);
```

```
/*omitir metadados de segmentação*/
```

```
EXEC
```

```
dbms_metadata.set_transform_param(dbms_metadata.session_transform,'SEGMENT_ATTRIBUTES',false);
```

```
/*incluir ';' após cada chamada a get_ddl*/
```

```
EXEC
```

```
dbms_metadata.set_transform_param(dbms_metadata.session_transform,'SQLTERMINATOR',true);
```

Recuperação estruturada de metadados

- A geração sempre útil para o pacote dbms

/*omitir me

EXEC

dbms_metadata.
transform,

/*omitir me

EXEC

dbms_metadata.
transform,

/*incluir

EXEC

dbms_metadata.set_transform_param(dbms_metadata.session_transform,'SQLTERMINATOR',true);

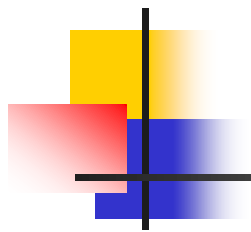
Em Java, estes comandos podem ser executados com um preparedstatement contendo o código “begin ... end;”



ORACLE

■ Referências:

- *<http://www.mathcs.emory.edu/~cheung/Courses/377/Others/tutorial.pdf>*
- *http://docs.oracle.com/cd/B19306_01/appdev.102/b14258/d_metada.htm#BGBHHHBG*
- *http://www.optimaldba.com/papers/DBMS_METADATA_han dout.pdf*



PRÁTICA 5