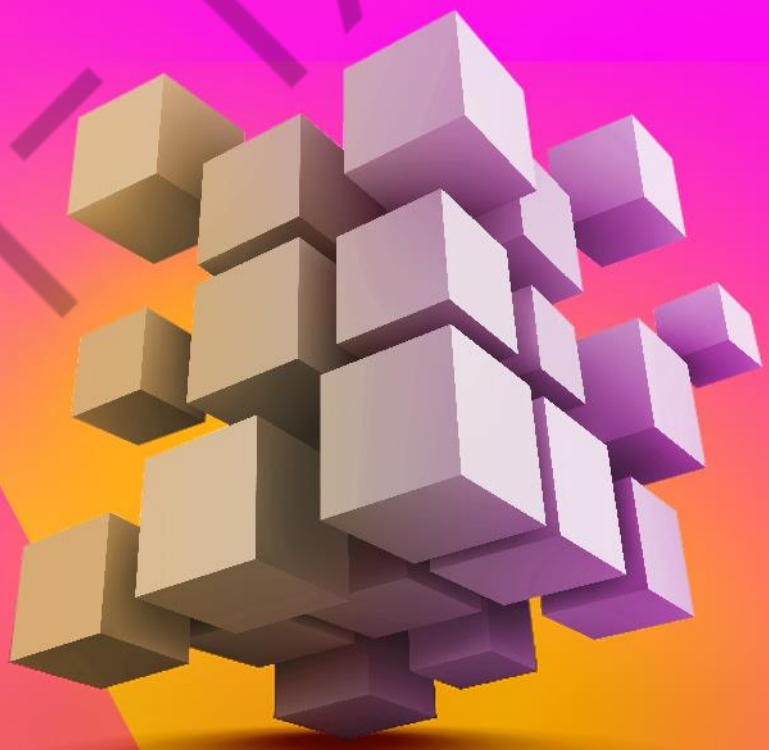
FIAP ON

PROGRAMAÇÃO EM BANCO E DADOS

OBJETOS DO *BANCO DE DADOS*

MILTON GOYA



5

LISTA DE QUADROS

LISTA DE CÓDIGOS-FONTE

SUMÁRIO

5 OBJETOS DO BANCO DE DADOS

5.1 Introdução aos objetos compilados do banco de dados

5.2 Dicionário de dados

Para a Oracle (2016), um dicionário de dados é um conjunto de metadados, que, por sua vez, são dados sobre os dados. O dicionário de dados fornece uma descrição do banco de dados. Nele, temos informações sobre a sua estrutura física, sua estrutura lógica e seu conteúdo. O dicionário de dados é criado durante o processo de criação do banco de dados e consiste em uma série de tabelas e *views*.

O dicionário de dados contém informações sobre os objetos do banco de dados, como, por exemplo, tabelas, sinônimos, índices e restrições. Normalmente, o usuário só pode ler as informações dentro do dicionário de dados. Qualquer manutenção ao dicionário de dados é feita automaticamente pelo servidor Oracle.

Para Feuerstein et al. (2014), existem três tipos de *views* que permitem a consulta do dicionário de dados. Elas são prefixadas pelas palavras USER_, ALL_ e DBA_.

As *views* USER_ fornecem informações sobre os objetos de propriedade do usuário conectado.

As *views* ALL_ fornecem informações sobre os objetos aos quais o usuário tem acesso.

As *views* DBA_ são de acesso exclusivo dos administradores do banco de dados e fornecem informações de todos os objetos do banco de dados.

Vejamos um exemplo simples:

```
SELECT DISTINCT object_type
  FROM user_objects;
```

Código-fonte 5.1 – Exemplo consulta a *view* USER_
Fonte: Elaborado pelo autor (2020)

No exemplo, serão listados todos os tipos de objetos criados pelo usuário conectado no momento. Note que usamos o comando DISTINCT para eliminar a duplicidade dos objetos nessa listagem.

Outro exemplo simples:

```
SELECT DISTINCT object_type
  FROM all_objects;
```

Código-fonte 5.2 – Exemplo consulta a *view* ALL_
Fonte: Elaborado pelo autor (2020)

No exemplo, serão listados todos os tipos de objetos aos quais o usuário conectado no momento tem acesso. Você perceberá que a diferença entre a quantidade de objetos criados pelo usuário e a quantidade de objetos a que o usuário tem acesso é notável. Nessa consulta, também usamos o comando DISTINCT para eliminar a duplicidade dos objetos na listagem.

5.3 Objetos PL/SQL armazenados

Para Dillon et al. (2013), os objetos PL/SQL armazenados são procedimentos, gatilhos, funções e pacotes. Todos esses objetos são armazenados no dicionário de dados com o seu código-fonte e a sua compilação. Toda vez que um objeto PL/SQL armazenado é chamado por uma seção, ele é lido a partir do dicionário de dados ou, caso já tenha sido executado anteriormente e esteja disponível, a partir da memória CACHE.

Até o momento, trabalhamos com blocos PL/SQL anônimos. Blocos PL/SQL anônimos são programas em PL/SQL não nomeados, cujo código não fica armazenado no banco de dados. A desvantagem de trabalhar com blocos anônimos é que eles precisam ser compilados toda vez que serão executados e não podem ser chamados por outras aplicações. Se quiser reexecutar um bloco PL/SQL anônimo, é necessário executar o SCRIPT com o bloco anônimo novamente.

Ao usarmos blocos PL/SQL nomeados, passamos a ter uma série de vantagens. Por exemplo, procedimentos e funções são armazenados no banco de dados em formato compilado, só haverá necessidade de recompilar o código, caso ocorram modificações no código ou em seus objetos dependentes.

Outra vantagem de armazenarmos os procedimentos e funções no banco de dados em formato compilado de forma nomeada é que outras aplicações e/ou usuários podem executá-los, desde que possuam os privilégios e autorizações para

tanto. Além disso, podemos passar parâmetros para os programas e, no caso das funções, podemos obter um retorno.

Existem várias *views* com informações sobre os objetos armazenados, destacamos algumas delas:

Nome da *view*	Descrição
USER_ARGUMENTS	Os argumentos (parâmetros) em todos os procedimentos e funções do seu esquema.
USER_DEPENDENCIES	As dependências para os objetos que você possui. Essa visão é usada principalmente pelo banco de dados Oracle para invalidar o status dos objetos de banco de dados quando um objeto de que eles dependem muda.
USER_ERRORS	O conjunto de erros recentes de compilação para todos os objetos armazenados que você possui. Essa exibição é acessada pelo comando SHOW ERRORS do SQL.
USER_OBJECTS	Os objetos que você possui podem, por exemplo, usar essa *view* para ver se um objeto está marcado como INVALID, encontrar todos os pacotes que têm EMP em seus nomes e assim por diante.
USER_PROCEDURES	Informações sobre os seus procedimentos armazenados.
USER_SOURCE	O código-fonte do texto para todos os objetos que você possui. Muito útil para analisar o código-fonte por meio do uso de comandos SQL.
USER_TRIGGERS e USER_TRIGGER_COLS	Informações sobre os seus gatilhos e as colunas identificadas neles. Inclui o código-fonte e uma descrição dos eventos.

Quadro 5.1 – Lista das exceções predefinidas
Fonte: ORACLE (2016)

5.3.1 Exibindo informações sobre os objetos armazenados

Para a Oracle (2016), cada linha da *view* USER_OBJECTS contém informações dos seus objetos do banco de dados. As colunas mais utilizadas são:

Nome da Coluna	Descrição
OBJECT_NAME	Nome do objeto.
OBJECT_TYPE	Tipo do objeto, alguns dos valores que essa coluna pode conter são: PACKAGE, FUNCTION ou TRIGGER
STATUS	Status do objeto, pode conter os valores VALID ou INVALID
LAST_DDL_TIME	Contém a data e hora da última alteração deste objeto.

Quadro 5.2 – Lista das colunas mais utilizadas da *view* USER_OBJECTS
Fonte: ORACLE (2016)

Vejamos alguns exemplos de consultas usando USER_OBJECTS.

```
SELECT object_name
  FROM user_objects
 WHERE object_type = 'TABLE'
 ORDER BY object_name
/
```

Código-fonte 5.3 – Lista o nome de todas as tabelas do usuário conectadas ao banco de dados
Fonte: ORACLE (2016).

O exemplo fornece uma listagem com os nomes de todos os objetos do tipo TABLE pertencentes ao usuário que está executando a consulta. Para obter uma listagem com o nome de todos os objetos do tipo TABLE a que o usuário que está executando a consulta tem acesso, basta substituir o nome da *view* USER_OBJECTS por ALL_OBJECTS.

```
SELECT object_type, object_name
  FROM user_objects
 WHERE status = 'INVALID'
 ORDER BY object_type, object_name
/
```

Código-fonte 5.4 – Lista o nome de todos os objetos do usuário conectados ao banco de dados cujo STATUS for INVALID
Fonte: ORACLE (2016)

Esse exemplo apresenta uma listagem com os nomes de todos os objetos com o STATUS de INVALID pertencentes ao usuário que está executando a consulta. O STATUS de um programa PL/SQL armazenado é definido como INVALID quando um objeto do qual é dependente é alterado. Os programas com o STATUS de INVALID devem ser recompilados. A recompilação pode ser feita manualmente ou automaticamente. A recompilação automática será feita na próxima vez que o banco de dados tentar executar o programa.

```
SELECT object_type, object_name,
       last_ddl_time
  FROM user_objects
 WHERE last_ddl_time >= TRUNC (SYSDATE)
 ORDER BY object_type, object_name
/
```

Código-fonte 5.5 – Lista todos os objetos do usuário conectados ao banco de dados que foram alterados hoje
Fonte: ORACLE (2016)

Esse exemplo mostra uma listagem com o tipo de objeto, nome do objeto e data da última alteração de todos os objetos pertencentes ao usuário que está executando a consulta e que tenham sido alterados hoje.

5.3.2 Exibindo informações sobre o código-fonte

Para a Oracle (2016), o código-fonte de todos os programas PL/SQL que você compilou com sucesso estão disponíveis por meio da *view* USER_SOURCE. As colunas mais utilizadas são:

Nome da Coluna	Descrição
NAME	Nome do objeto.
TYPE	Tipo do objeto. O tipo de objeto pode variar de fonte de programa JAVA até fonte de programa de uma TRIGGER
LINE	Número da linha do código fonte
TEXT	Texto do código-fonte

Quadro 5.3 – Lista das colunas mais utilizadas da *view* USER_SOURCE
Fonte: ORACLE (2016)

A USER_SOURCE pode ser consultada para obtermos várias informações significativas, por exemplo, você pode usá-la para:

- Descobrir quais programas usam determinado valor literal que precise ser alterado.
- Verificar se os padrões de nomes de variáveis da empresa estão sendo obedecidos.
- Encontrar todos os programas PL/SQL que chamem um determinado procedimento e/ou função e/ou pacote.

Vejamos um exemplo:

```
SELECT name, line, text
  FROM user_source
 WHERE UPPER (text)
  LIKE '%DEPT%'
 ORDER BY name, line
/
```

Código-fonte 5.6 – Lista todos os programas do usuário conectados ao banco de dados que usam a tabela DEPT
Fonte: ORACLE (2016)

Esse exemplo fornece uma listagem com o nome do programa, número da linha e o texto do código-fonte que trabalha com a tabela DEPT. Como é uma pesquisa em um texto, o exemplo também pode retornar linhas comentadas que possuam a palavra DEPT ou variáveis que tenham a palavra DEPT em seu nome.

5.3.3 Exibindo informações sobre procedimentos e funções

Para a Oracle (2016), as informações sobre os procedimentos e funções criadas pelo usuário podem ser obtidas por meio da *view* USER_PROCEDURES. As colunas mais utilizadas são:

Nome da Coluna	Descrição
OBJECT_NAME	Nome do objeto.
PROCEDURE_NAME	Nome do procedimento ou função
AUTHID	Mostra se um procedimento ou função é executada com os privilégios de execução de outro programa (DEFINER) ou se é executado com os privilégios do usuário que executa o procedimento ou função (CURRENT_USER).
OBJECT_TYPE	Tipo do objeto.

Quadro 5.4 – Lista das colunas mais utilizadas da *view* USER_PROCEDURES
Fonte: ORACLE (2016)

Vejamos um exemplo de uso:

```
SELECT   object_name
       , procedure_name
    FROM user_procedures
   WHERE authid = 'CURRENT_USER'
ORDER BY object_name, procedure_name
/
```

Código-fonte 5.7 – Lista todos os procedimentos e funções do usuário conectados ao banco de dados que são executados com os privilégios do próprio usuário
Fonte: ORACLE (2016)

Esse exemplo apresenta uma listagem com o nome do objeto e nomes das funções e procedimentos que serão executados usando os privilégios de execução do usuário que executa o programa. Os privilégios do executor do programa são usados em tempo de execução para resolver referências a objetos de banco de dados, como, por exemplo, tabelas.

5.3.4 Exibindo informações sobre gatilhos (triggers)

Para a Oracle (2016), as informações sobre os gatilhos (*TRIGGERS*) podem ser obtidas por meio da *view* USER_TRIGGERS. As colunas mais utilizadas são:

Nome da Coluna	Descrição
TRIGGER_NAME	Nome do gatilho
TRIGGER_TYPE	Texto indicando se o gatilho é do tipo AFTER ou BEFORE e se é do tipo ROW ou STATEMENT.
TRIGGERING_EVENT	Indica qual tipo de operação irá disparar o gatilho. Pode assumir os valores INSERT ou UPDATE ou DELETE ou INSERT OR UPDATE ou DELETE OR UPDATE
TABLE_NAME	O nome da tabela em que o gatilho está definido
STATUS	O status do gatilho, pode ser ENABLE ou DISABLE
WHEN_CLAUSE	Uma cláusula opcional que pode ser usada para evitar a execução desnecessária do corpo do gatilho
TRIGGER_BODY	O código executado quando o gatilho dispara

Quadro 5.5 – Lista das colunas mais utilizadas da *view* USER_TRIGGERS
Fonte: ORACLE (2016)

Vejamos um exemplo de uso:

```
SELECT *
  FROM user_triggers
 WHERE status = 'DISABLED'
/
```

Código-fonte 5.8 – Lista todos os gatilhos do usuário conectados ao banco de dados que estão desabilitados
Fonte: ORACLE (2016)

Esse exemplo fornece uma listagem com informações sobre todos os gatilhos desabilitados criados pelo usuário conectado ao banco de dados.

```
SELECT *
  FROM user_triggers
 WHERE table_name = 'EMP'
   AND trigger_type LIKE '%EACH ROW'
/
```

Código-fonte 5.9 – Lista todos os gatilhos do usuário conectados ao banco de dados que estão associados à tabela EMP
Fonte: ORACLE (2016)

Esse exemplo mostra uma listagem com informações sobre todos os gatilhos associados à tabela EMP que são executados em nível de linha.

```
SELECT *
  FROM user_triggers
 WHERE triggering_event LIKE '%UPDATE%'
/
```

Código-fonte 5.10 – Lista todos os gatilhos do usuário conectados ao banco de dados que são acionados por um comando de atualização
Fonte: ORACLE (2016)

Esse exemplo apresenta uma listagem com informações sobre todos os gatilhos que são acionados quando uma operação de atualização é executada.

Exemplo, tabelas.

CONCLUSÃO

Com a necessidade de criação de tantos elementos ou objetos de bancos de dados diferentes, é natural que se fique um tanto perdido sobre quais estão compondo a solução. Conhecer o dicionário de dados é como extrair as informações certas de tais elementos, é extremamente importante para se manter a organização da solução e, eventualmente, recuperar e reaproveitar algum elemento de um projeto antigo para um novo.

REFERÊNCIAS

DILLON, Sean; BECK, Christopher; KYTE, Thomas; KALLMAN, Joel; ROGERS, Howard. **Beginning Oracle Programming**. Apress, 2013.

FEUERSTEIN, Steven; PRIBYL, Bill. **Oracle Pl/SqlProgramming**. O'Reilly Media, 2014.

ORACLE. **Oracle Database:** PL/SQL Language Reference 12c Release 2 (12.2) B28370-05. Oracle Press, 2016.

PUGA, Sandra; FRANÇA, Edson; GOYA, Milton. **Banco de dados**. São Paulo: Pearson, 2015.