

MIGUEL MARCELO NASCIMENTO FRANCO

Introdução a programação plsql: uma pesquisa bibliográfica com relato de experiência por meio da plataforma Oracle.

guelmi_marcelo@hotmail.com

luciano@claretiano.edu.br

Autor : Miguel Marcelo Nascimento Franco

Coautor : Prof. Luciano Bergamo

Orientador : Prof. Luciano Bergamo

RESUMO

Este artigo trata acerca de um estudo de introdução a programação PL/SQL uma pesquisa bibliográfica com relato de experiência por meio da plataforma Oracle, cujo objetivo consiste em explicar de uma maneira simples e de fácil entendimento o conceito básico de PL/SQL no âmbito profissional, principalmente, em conformidade e com base no livro escrito por Jason Prince, Oracle Database 11g SQL Domine SQL e PL/SQL no banco de dados Oracle. A motivação para esse assunto deu-se pela a relevância de ser ter o conhecimento e o interesse despertado sobre o assunto. Para tanto, o procedimento metodológico caracterizou-se por uma revisão bibliográfica à luz do referencial citado, foram realizadas pesquisas em sites confiáveis de profissionais certificados. E o resultado destas pesquisas é o entendimento sobre o tema em questão como agir e realizar algo não por não conhecer e sim pelo o conhecer. Realmente ver como a introdução a linguagem nos torna um profissional melhor e preparado para exercer o cargo que lhe é confiado.

Palavras-chave: PL/SQL. SQL. Programação.

INTRODUÇÃO A PROGRAMAÇÃO PL/SQL: UMA PESQUISA BIBLIOGRÁFICA COM RELATO DE EXPERIÊNCIA POR MEIO DA PLATAFORMA ORACLE

INTRODUÇÃO

Este artigo disserta sobre a introdução da programação PL/SQL, uma pesquisa bibliográfica com relato de experiência por meio da plataforma Oracle, com a

finalidade de ressaltar a importância do conhecimento PL/SQL para o fortalecimento no âmbito profissional com base no livro escrito por Jason Prince, Oracle Database 11g SQL Domine SQL e PL/SQL no banco de dados Oracle, traçando contorno com ênfase maior às suas peculiaridades, ingressando nos conceitos da estrutura de blocos, variáveis, lógica condicional, loops, cursores, exceções, procedures, funções, pacotes, triggers.

Justifica-se, para tanto, mostrar o fundamental entender dos conceitos básicos de PL/SQL para o autor se desenvolver, mostrar o saber, o conhecer introdutório desta faz-se ter foco, percepção no meio profissional, buscar, determinar, compreender, tornar-se um profissionalmente melhor. Objetivamente a interpretação e o esforço do aprender, alguém conseqüentemente absorvem as citações.

Para discorrer sobre o tema, é importante também conceituar estrutura de bloco que podem ser entendidas como essenciais para as instruções PL/SQL. A necessidade de compreender e aprender sobre o tema por parte de uma grande parcela de desenvolvedores, maior ênfase aos temas relacionados. De certo compreensível, posto que, nos dias atuais, não só a relevância como a atualidade dos estudos e acervos sobre o tema proposto povoa o universo das atenções ligadas as relações entre analistas e desenvolvedores.

Portanto, não se torna fácil explicar o que é o PL/SQL, mas segundo o livro de Prince (2009, p.368) costuma-se conceituá-la como “[...] permite escrever programas que contêm instruções SQL”. Assim, em suas definições, o termo PL tem uma larga abrangência do ponto de vista conceitual.

A motivação para essa realização deu-se pela a relevância de melhorar e se aprofundar do tema e o interesse despertado sobre o assunto. Para tanto, o procedimento metodológico caracterizou-se por uma revisão bibliográfica de cunho teórico descritivo.

Nesse sentido, o objetivo geral consiste em explicar de uma maneira simples e de fácil entendimento o conceito de estrutura de bloco e variáveis, seus tipos e sua lógica condicional no âmbito profissional, principalmente, no convívio das relações que

se estabelecem no conceito de lógico centralizado com base nas concepções de especialistas como Hotka (2003), Fanderuff (2003), Prince (2009), entre outros.

DESENVOLVIMENTO

Para contextualização do estudo, reportamo-nos aos conceitos de linguagem de programação procedural, estrutura de blocos e lógica condicional e seus tipos.

Independente de afirmações de cientistas e expoentes da formação, segundo Prince (2009), entende-se por PL/SQL, uma linguagem de programação procedural, conhecida como PL/SQL (Procedural Language/SQL). O PL/SQL permite escrever programas que contêm instruções SQL.

No campo organizacional, conforme esse autor, você pode usar PL/SQL para adicionar lógica de negócio em um aplicativo de banco de dados. Essa lógica de negócio centralizada pode ser executada por qualquer programa que possa acessar o banco de dados (PRINCE, 2009).

Assim, em suas definições, a estrutura de bloco tem uma larga abrangência do ponto de vista conceitual. Entretanto, numa primeira definição, conforme Prince (2009, p.368):

No entanto, conforme Prince (2009):

“os programas em PL/SQL são divididos em estruturas conhecidas como blocos, com cada bloco contendo instruções PL/SQL e SQL. Um bloco PL/SQL tem a seguinte estrutura: declare instruções_de_declaração begin instruções_executáveisexception instruções_de_tratamento_de_exceção end;” Prince (2009).

Ainda de acordo com Prince (2009):

“instruções_de_declaração declaram as variáveis usadas no restante do bloco PL/SQL. Os blocos DECLARE são opcionais, instruções_executáveis são instruções que serão executadas, as quais podem incluir loops, lógica condicional etc, instruções_de_tratamento_de_exceção são instruções que tratam de todos os erros de execução que possam ocorrer quando o bloco é executado. Os blocos EXCEPTION são opcionais” Prince (2009).

Portanto, realizar tarefas corretas, obedecer a padrões da linguagem, mostrando-lhe que ser competente é buscar fortalecer a base do conhecimento adquirido. Assim, tal conhecimento torna-se melhor, buscar ideais para crescer profissionalmente, faz a vontade de se querer ter a ânsia de aprender. As noções aqui apresentadas e ilustradas fazem o querer bem, de se aprender.

Portanto, Contextualizando a esse respeito, pode-se destacar que nas palavras de Hotka (2003, p. 11), que “[...] o SQL é um conjunto de comandos que todos os programas e usuários devem usar para se comunicar com o banco de dados Oracle”.

No entanto, conforme Hotka (2009):

“O SQL é uma linguagem não procedural que processa de uma só vez todo o conjunto de registro em vez de processar registros individuais. O Oracle usa o otimizador para determinar a melhor maneira de acessar os dados; ele possui dois otimizadores: o original que é um otimizador baseado em regras que toma as decisões com base em 19 regras (que incluem o uso de índices – se houver –, a ordem dos itens na cláusula where e assim por diante) e o otimizador baseado em custo que toma as decisões com base nas estatísticas obtidas pelo comando ANALYZE do Oracle” Hotka (2003).

Nestes casos, precisa haver um consenso no bom procedimento adotado, onde possa atender a um conjunto de regras consideradas válidas.

Em se tratando da PL/SQL, segundo Prince (2009), toda instrução é terminada por um ponto-e-vírgula (;) e um bloco PL/SQL é terminado com caractere de barra normal (/).

Considerando agora as variáveis e seus tipos, segundo Prince (2009), as variáveis são declaradas dentro de bloco DECLARE. Uma declaração de variável tem um nome e tipo. Os tipos PL/SQL são semelhantes aos tipos de coluna de banco de dados. É possível especificar o tipo de uma variável usando a palavra-chave % type, que diz ao PL/SQL para que use o mesmo tipo da coluna especificada em uma tabela.

Se tratando de lógica condicional usam-se palavras-chave IF, THEN, ELSE, ELSIF e END IF para executar a lógica, também são possíveis incorporar uma instrução IF dentro de outra instrução IF, a lógica condicional flui de acordo com a sua

condição, se a condição for verdadeira então os comandos dentro da condição são executados, se uma determinada condição for falsa, mas a outra condição é verdadeira então a condição falsa é executada, caso a condição não seja verdadeira ou falsa então outra condição é executada (PRINCE, 2009).

A lógica condicional e sua estrutura estar presente no dia a dia sejam desenvolvendo ou em decisões.

Entretanto, falando um pouco de loops para Prince (2009):

“usa-se um loop para executar instruções zero ou mais vezes. Existem três tipos de loops na PL/SQL, loops simples, loops while, loops for. Os loops simples são executados até que você determine o loop explicitamente, os loops while são executados até que ocorra uma condição especificada, os loops for são executados um número predeterminado de vezes” Prince (2009).

Entender a sintaxe da PL/SQL é de grande importância, seus conceitos e estrutura facilitam na hora de codificar, então para Prince, um loop simples ele segue a seguinte sintaxe, loop instruções end loop, para terminar o loop, use uma instrução exit ou exit when. A instrução exit termina um loop imediatamente, a instrução exit when termina o loop quando ocorre uma condição especificada. A instrução exit when pode aparecer em qualquer lugar do código do loop. A sintaxe de um loop while é while condição loop instruções end loop. Um loop for é executado um número predeterminado de vezes, podendo determinar o número de vezes especificando o limite inferior e superior de uma variável de loop. Então, a variável de loop é incrementada (ou decrementada) em cada passagem do loop Prince (2009).

E cursores? Segundo Fanderuff (2003, p.236): “[...] cursores são áreas compostas de linhas e colunas em memórias que servem para armazenar o resultado de seleção que retorna 0 ou mais linhas. No PL/SQL os cursores podem ser de dois tipos, cursores explícitos e cursores implícitos”.

Para Damaris (2003), cursores explícitos são utilizados para a execução de consultas que possam retornar nenhum ou mais de uma linha. Nesse caso, o cursor deve ser explicitamente declarado na área de declaração (Declare). O nome do cursor não pode ser igual ao da tabela.

Embora considerando o entendimento acima dos cursores explícitos. Mesmo assim, “[...] O primeiro passo é declarar as variáveis que serão usadas para armazenar os valores da coluna. Essas variáveis devem ser compatíveis com os tipos de colunas. O passo dois é declara o curso. Uma declaração de cursor consiste em um nome atribuído a ele e a consulta que você deseja executar. O passo três é abrir o cursor. Abre-se o cursor usando a instrução OPEN, que deve ser colocada na seção executável do bloco. O passo quatro é buscar as linhas do cursor, usa-se a instrução FETCH. A instrução fetch lê os valores de coluna nas variáveis declaradas no passo um. E o quinto passo é fechar o cursor. Usa-se a instrução CLOSE. O fechamento do cursor libera recursos do sistema”. (PRINCE, 2009, p.375).

Concluindo o entendimento dos cursores pode-se usar nas palavras de Prince (2009, p.377) um “[...] loop FOR para acessar as linhas de um cursor. Quando faz isso, não precisa abrir e fechar o cursor explicitamente, o loop FOR faz isso automaticamente”.

Em se tratando de exceções, segundo Prince (2009), as exceções são usadas para tratar de erros em tempo de execução no código PL/SQL. Que contém um bloco exception. A exceção ZERO_DIVIDE é lançada quando é feita a tentativa de dividir um número por zero. Quando uma exceção é lançada, o controle do programa passa para o bloco EXCEPTION e a clausula WHEN é examinada para encontrar uma exceção correspondente, então, o código dentro da cláusula correspondente é executado. A exceção zero_divide é lançada no bloco Begin e o controle do programa passa então para o bloco exception, uma exceção correspondente é encontrada na cláusula when e o código dentro da cláusula é executado. Se nenhuma exceção correspondente é encontrada, a exceção é propagada para o bloco externo.

Portanto, para concluir a contextualização sobre as exceções, pode se usar a exceção OTHERS para tratar de todas as exceções. Como o others corresponde a todas as exceções, deve listá-la depois de todas as exceções específicas em seu bloco EXCEPTION. Se listar others em outro lugar qualquer, o banco de dados retornará o erro PLS-00370 (PRINCE, 2009).

Falando de procedures, Prince (2009), acrescenta que uma procedure contém um grupo de instruções SQL e PL/SQL. As procedures permitem centralizar

sua lógica do negócio no banco de dados e podem ser usadas por qualquer programa que acesse o banco de dados. Para criar uma procedure, usa-se a instrução `create procedure`, `or replace` significa que a procedure deve substituir uma procedure existente, um parâmetro IN deve ser configurado com o valor quando a procedure é executada. O valor de um parâmetro IN não pode ser alterado no corpo da procedure, OUT, significa que o parâmetro é configurado com um valor no corpo da procedure, IN OUT, significa que o parâmetro pode ter um valor quando a procedure é executada e o valor pode ser alterado no corpo. Executa-se uma procedure usando a instrução `call`, excluir uma procedure usando `drop procedure`.

Reportando-se a importância de centralizar a regra de negócio no banco de dados, iremos abordar agora a importância das funções para Prince (2009), uma função é semelhante a um procedure, exceto que uma função deve retornar um valor. Juntas, as procedures armazenadas e as funções às vezes são referidas como subprogramas armazenados, pois são pequenos programas. Cria uma função usando a instrução `create function`, é possível chamar uma função usando uma instrução `select`, usa-se `drop function` para excluir uma função.

Continuando o conceito de subprogramas armazenados, reforçando os estudos, abordo então sobre a importância de agrupar as procedures e funções em pacotes. Os pacotes, nas palavras de Prince (2009) permitem:

“encapsular funcionalidades relacionadas em uma unidade independente. Modularizando o seu código PL/SQL por meio de pacotes, é possível construir suas próprias bibliotecas de códigos que outros programadores podem reutilizar. O banco de dados Oracle vem com a biblioteca de pacotes pré-configurados, os quais permitem acessar arquivos externos, gerenciar o banco de dados, gerar códigos HTML e muito mais, para ver todos os pacotes, você deve consultar o manual Oracle Database PL/SQL Packages And Types Reference da Oracle Corporation” Prince (2009).

Contudo, para Puga, França e Goya (2013), pacotes são blocos PL/SQL que podem agrupar procedimentos, funções, exceções, estruturas de dados, cursores ou declarações de variáveis globais. A especificação do pacote (`package specification`) é o conjunto de declarações dele. Deve conter a declaração das variáveis, cursores,

exceções, cabeçalho dos procedimentos e funções que serão acessíveis ao ambiente externo ao pacote.

Observa-se agora o conceito que pode ser utilizado para a implementação de regras de negócio. Os Gatilhos, “[...] ou triggers, são blocos PL/SQL acionados ou disparados automaticamente, antes ou depois da ocorrência de um evento, que pode ser uma operação DML para inserção, alteração ou exclusão de dados ou um evento associado ao banco, como efetuar logon de um usuário. Os triggers podem ser utilizados para geração de valores calculados, manutenção da integridade referencial dos dados, replicação de dados ou manutenção de registro histórico de alteração ocorridas no banco de dados, entre outras”.

Evidentemente, para o melhor entendimento dessa contextualização que será abordada logo a seguir segundo Prince (2009), a instrução, *before* significa que o trigger é disparado antes que o evento de disparo seja executado, *after* significa que o trigger é disparado depois que o evento de disparo seja executado, *for each row* significa que o trigger é um trigger em nível de linha, ou seja, o código contido dentro do corpo da trigger é executado para cada linha quando a trigger dispara.

Há uma curiosa semelhança entre o SQL e o PL/SQL e a efetiva importância que se dá a linguagem no campo prático, ou seja, no desenvolvimento no dia a dia. Sabe-se, porém, que tomar decisões inteligentes e coerentes em uma determinada situação no desenvolvimento do algoritmo é fundamental para preservar boas práticas da linguagem em si.

CONSIDERAÇÕES FINAIS

O presente artigo tratou dos conceitos de linguagem de programação procedural, estrutura de blocos e lógica condicional e seus tipos com base nos estudos e experiências dos escritores Prince, Fanderuff e Hotka entre outros.

O tema aqui descrito é vastíssimo, e por isso não comporta solução única e global, tendo em vista conceitos abordados da linguagem PL/SQL vai além do conteúdo abordado, entretanto esse é o alicerce para passos mais longos.

A introdução é um pré-requisito fundamental para o desenvolvimento do meio, entretanto, se prender e reter o que foi proposto, tornar o meio adquirido mínimo. Tendo em vista ampliar e complementar o estudo faz-se necessário ouvir e orientar sobre o tema em questão. Sobretudo, o constante crescimento e aperfeiçoamento da tecnologia fazem-se ter atenção em dobro a respeito da importância que se tem sobre o aprimoramento de todo esse material.

De forma explícita, este estudo proporcionou uma reflexão sobre os caminhos que a linguagem PL/SQL tem na condução profissional, nesse sentido, não é tarefa das mais difíceis, considerando que o objetivo maior é dominar os conceitos aqui abordados. Dessa forma, este estudo não se esgota, visto que há muito que se relata sobre o tema, propiciando pesquisas futuras.

REFERÊNCIAS

- HOTKA, DAN. Aprendendo Oracle 9i. Makron Books. São Paulo: Pearson Education do Brasil, 2003; 11p.
- FANDERUFF, DAMARIS. Dominando o Oracle 9i. Makron Books. São Paulo: Pearson Education do Brasil, 2003; 236p.
- PRINCE. Oracle Database 11g SQL Domine SQL e PL/SQL no Banco de Dados Oracle. Artmed Editora S.A. São Paulo: Bookman Editora S.A, 2009; 393p.
- PUGA, FRANÇA, GOYA. Banco de Dados Implementação em SQL, PL/SQL e Oracle 11g. São Paulo: Pearson Education do Brasil, 2013; 33p.