

FIAP GRADUAÇÃO

# Tecnologia em Análise e Desenvolvimento de Sistemas

Database Application Development

PROF. MILTON

# Introdução à Linguagem PL/SQL

## Objetivos

Ao concluir esta lição, você será capaz de:

- Explicar a necessidade da linguagem PL/SQL
- Explicar as vantagens da linguagem PL/SQL
- Identificar os diferentes tipos de blocos PL/SQL
- Exibir mensagens em PL/SQL

### Objetivos

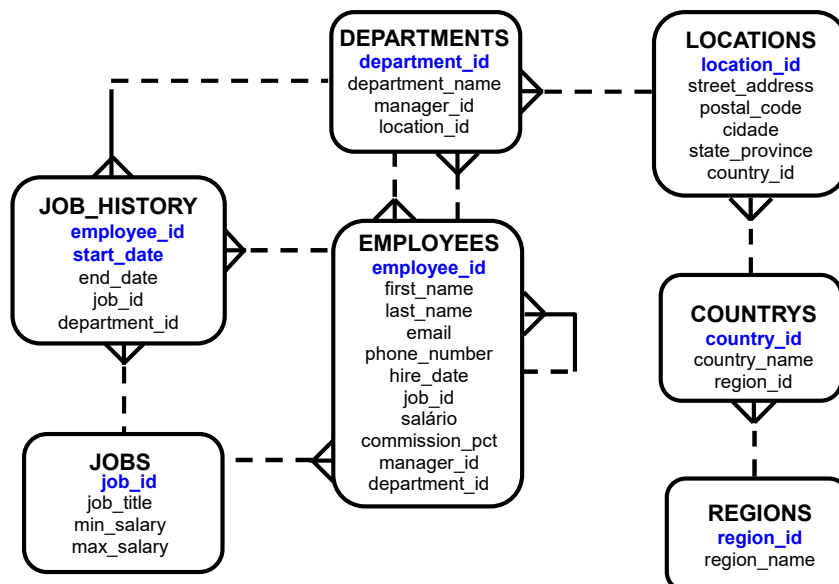
Esta lição apresenta a linguagem PL/SQL e as estruturas de programação PL/SQL. Você também aprenderá as vantagens do código PL/SQL.

# Agenda

FIAP

- Fundamentos das vantagens e da estrutura do código PL/SQL
- Examinando blocos PL/SQL
- Gerando mensagens de saída em PL/SQL

## Esquema Human Resources (HR) Usado Neste Curso



## Esquema Human Resources (HR) Usado Neste Curso

O esquema Human Resources (HR) faz parte dos Oracle Sample Schemas que podem ser instalados em um banco de dados Oracle. As sessões de exercícios deste curso utilizam dados desse esquema HR.

### Descrição das Tabelas

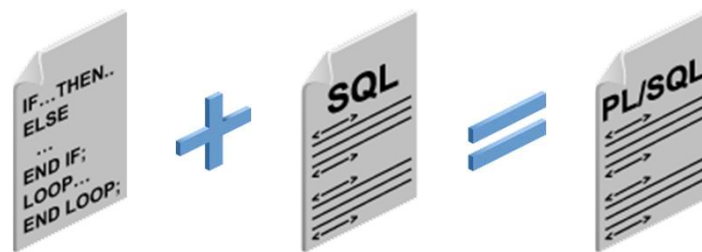
- A tabela **REGIONS** contém linhas que representam uma região, como as Américas ou a Ásia.
- A tabela **COUNTRIES** contém linhas para os países, cada um dos quais associado a uma região.
- A tabela **LOCATIONS** contém o endereço específico de determinado escritório, depósito ou local de produção de uma empresa em determinado país.
- A tabela **DEPARTMENTS** mostra detalhes dos departamentos onde os funcionários trabalham. Cada departamento pode ter um relacionamento que representa o gerente do departamento na tabela **EMPLOYEES**.
- A tabela **EMPLOYEES** contém detalhes sobre cada funcionário que trabalha em um departamento. Alguns funcionários talvez não tenham sido designados a departamento algum.
- A tabela **JOBS** contém os tipos de cargos que cada funcionário pode exercer.
- A tabela **JOB\_HISTORY** contém o histórico de cargos dos funcionários. Se um funcionário mudar de departamento mantendo o mesmo cargo ou mudar de cargo no

mesmo departamento, uma nova linha será inserida nessa tabela com as informações do cargo antigo do funcionário.

# Sobre a Linguagem PL/SQL

## PL/SQL:

- Significa a extensão Procedural Language do código SQL
- É a linguagem padrão de acesso a dados da Oracle Corporation para bancos de dados relacionais
- Integra totalmente as estruturas procedurais com o código SQL



## Sobre a Linguagem PL/SQL

SQL (Structured Query Language) é a principal linguagem usada para acessar e modificar dados em bancos de dados relacionais. Existem apenas alguns comandos SQL, portanto você pode aprendê-los e usá-los facilmente.

Veja um exemplo:

```
SELECT first_name, department_id, salary FROM employees;
```

A instrução SQL anterior é simples e direta. No entanto, se você quiser alterar dados obtidos de modo condicional, enfrentará as limitações do código SQL.

Considere a instrução de uma questão um pouco modificada: Para cada funcionário recuperado, verifique o ID do departamento e o salário. Dependendo do desempenho do departamento e do salário do funcionário, você poderá fornecer bônus distintos aos funcionários.

Verificando a questão, você sabe que precisa executar a instrução SQL anterior, coletar os dados e aplicar lógica aos dados.

- Uma solução é criar uma instrução SQL para que cada departamento ofereça bônus aos funcionários daquele departamento que concede o bônus. Lembre-se de que é necessário verificar o componente salary para decidir o valor do bônus. Isso torna o processo um pouco mais complexo.
- Uma solução mais eficaz deverá incluir instruções condicionais. O código PL/SQL foi projetado para atender a esses requisitos. Ele fornece uma extensão de programação ao



código SQL existente.

# Sobre a Linguagem PL/SQL

## PL/SQL:

- Fornece uma estrutura de blocos para as unidades executáveis do código. A manutenção do código é facilitada com uma estrutura bem definida.
- Fornece estruturas procedurais, como:
  - Variáveis, constantes e tipos de dados
  - Estruturas de controle, como instruções condicionais e loops
  - Unidades de programa reutilizáveis, que são criadas uma vez e executadas várias vezes

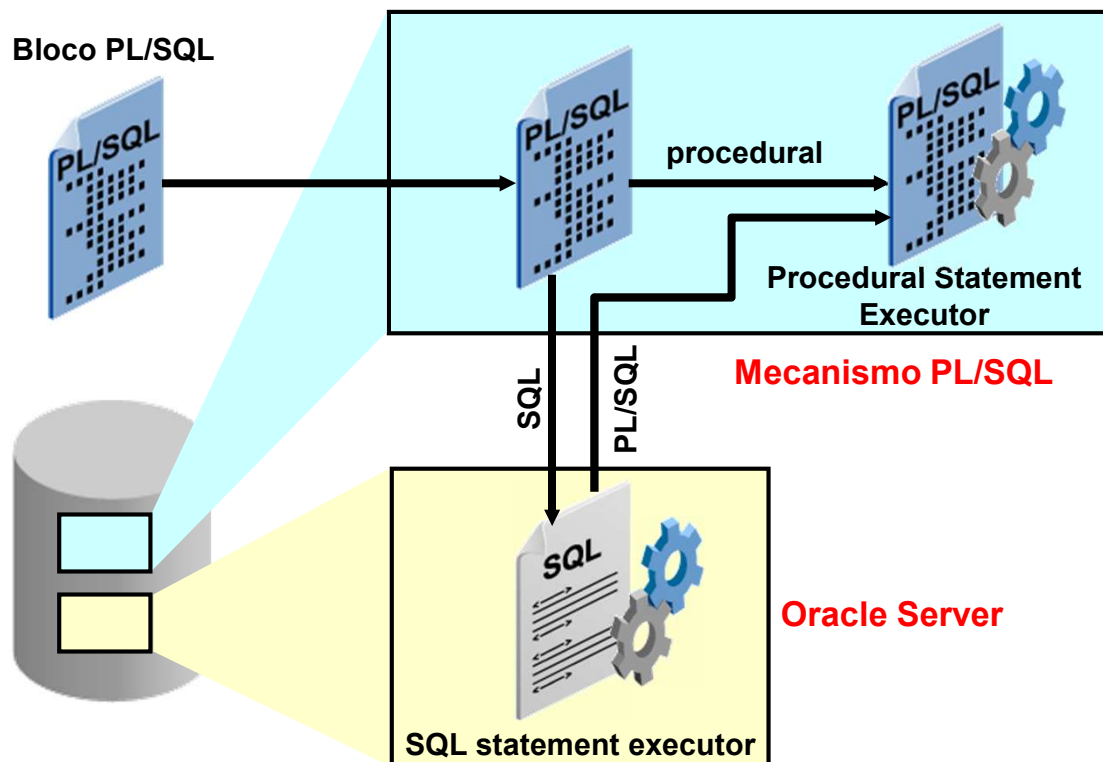
## Sobre a Linguagem PL/SQL (continuação)

A linguagem PL/SQL define uma estrutura de blocos para a criação de códigos. A manutenção e a depuração do código são facilitadas com tal estrutura porque é possível entender facilmente o fluxo e a execução da unidade de programa.

A linguagem PL/SQL dispõe de modernos recursos de engenharia de software, como encapsulamento de dados, tratamento de exceções, ocultamento de informações e orientação a objetos. Ela fornece a programação mais avançada ao conjunto de ferramentas e ao Oracle Server. O código PL/SQL fornece todas estruturas procedurais disponíveis em qualquer 3GL (linguagem de terceira geração).

## Arquitetura de Runtime de PL/SQL

FIAP



9

### Arquitetura de Runtime de PL/SQL

O diagrama mostra um bloco PL/SQL que está sendo executado pelo mecanismo PL/SQL. Esse mecanismo reside:

- No banco de dados Oracle para executar subprogramas armazenados
- No Oracle Forms Client, quando você executa aplicações cliente/servidor, ou no Oracle Application Server, quando você usa o Oracle Forms Services para executar o Forms na Web

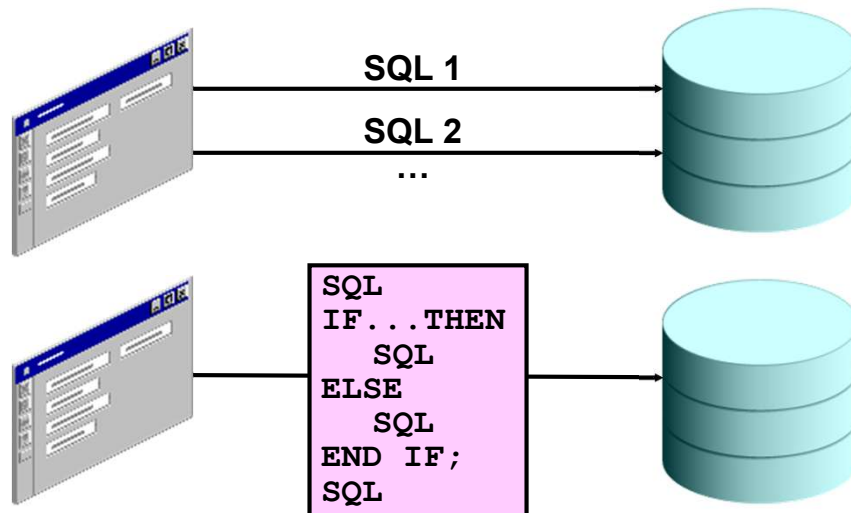
Independentemente do ambiente de runtime PL/SQL, a arquitetura básica permanece a mesma. Portanto, todas as instruções PL/SQL são processadas no Procedural Statement Executor, e todas as instruções SQL devem ser enviadas ao SQL Statement Executor para serem processadas pelo Oracle Server. O ambiente SQL também pode chamar o ambiente PL/SQL. Por exemplo, o ambiente PL/SQL é chamado quando uma função PL/SQL é usada em uma instrução `SELECT`.

O mecanismo PL/SQL é uma máquina virtual residente na memória que processa as instruções m-code PL/SQL. Quando o mecanismo PL/SQL encontra uma instrução SQL, ocorre uma alternância de contexto a fim de especificar a instrução SQL para os processos do Oracle Server. O mecanismo PL/SQL aguarda o término da instrução SQL e o retorno dos resultados para depois continuar processando as instruções subsequentes do bloco PL/SQL. O mecanismo PL/SQL do Oracle Forms é executado no cliente para a implementação de cliente/servidor e no

servidor de aplicações para a implementação do Forms Services. Em ambos os casos, as instruções SQL geralmente são enviadas pela rede para processamento em um Oracle Server.

## Vantagens da Linguagem PL/SQL

- Integração de estruturas procedurais com o código SQL
- Melhor desempenho



### Vantagens da Linguagem PL/SQL

**Integração de estruturas procedurais com o código SQL:** A maior vantagem da linguagem PL/SQL é a integração das estruturas procedurais com o código SQL. SQL é uma linguagem não procedural. Quando você executa um comando SQL, o seu comando avisa o servidor de banco de dados sobre *o que* fazer. No entanto, você não pode especificar *como* fazer. O código PL/SQL integra instruções de controle e instruções condicionais com SQL, proporcionando maior controle sobre as instruções SQL e sua execução. Anteriormente nesta seção, você viu um exemplo da necessidade dessa integração.

**Melhor desempenho:** Sem o código PL/SQL, não seria possível combinar logicamente as instruções SQL como uma unidade. Caso você tenha projetado uma aplicação que contenha muitos forms, poderá haver, em cada form, vários forms diferentes com campos. Quando um form submeter dados, você deverá executar várias instruções SQL. As instruções SQL são enviadas ao banco de dados uma de cada vez. Isso resulta em vários percursos de ida e volta pela rede e uma chamada para o banco de dados de cada instrução SQL, aumentando, dessa forma, o tráfego na rede e reduzindo o desempenho (especialmente em um modelo cliente/servidor).

Com o código PL/SQL, você pode combinar todas essas instruções SQL em uma só unidade de programa. A aplicação pode enviar o bloco inteiro para o banco de dados, em vez de enviar as instruções SQL uma de cada vez. Isso reduz drasticamente o número de chamadas do banco de dados. Como é ilustrado no slide, se a aplicação for predominantemente SQL, você poderá usar blocos PL/SQL para agrupar as instruções SQL antes de enviá-las ao servidor de banco de dados.

Oracle para execução.

# Vantagens da Linguagem PL/SQL

- Desenvolvimento de programas em módulos
- Integração com ferramentas Oracle
- Portabilidade
- Tratamento de exceções

## Vantagens da Linguagem PL/SQL (continuação)

**Desenvolvimento de programas em módulos:** A unidade básica em todos os programas PL/SQL é o bloco. Os blocos podem estar em sequência ou aninhados em outros blocos. O desenvolvimento de programas em módulos tem as seguintes vantagens:

- Agrupar logicamente em blocos as instruções relacionadas.
- Aninhar blocos dentro de blocos maiores para criar programas avançados.
- Fragmentar a aplicação em módulos menores. Se estiver projetando uma aplicação complexa, o código PL/SQL permitirá que você fragmente a aplicação em módulos menores, gerenciáveis e logicamente relacionados.
- Manter e depurar o código com facilidade.

No código PL/SQL, a modularização é implementada através de procedures, funções e pacotes, que são abordados na lição “Apresentando Procedures e Funções Armazenados”.

**Integração com ferramentas:** O mecanismo PL/SQL é integrado às ferramentas Oracle, como Oracle Forms e Oracle Reports. Quando você usa essas ferramentas, o mecanismo PL/SQL disponível localmente processa as instruções procedurais e apenas as instruções SQL são passadas para o banco de dados.

## **Vantagens da Linguagem PL/SQL (continuação)**

**Portabilidade:** Os programas PL/SQL podem ser executados em qualquer lugar onde seja executado um Oracle Server, independentemente do sistema operacional e da plataforma. Não é necessário personalizá-los para cada novo ambiente. Você pode criar pacotes de programas portáteis e bibliotecas que podem ser reutilizadas em diferentes ambientes.

**Tratamento de exceções:** O código PL/SQL permite tratar exceções com eficácia. Você pode definir blocos separados para lidar com as exceções. Você aprenderá mais sobre o tratamento de exceções na lição “Tratamento de Exceções”.

O código PL/SQL compartilha o mesmo sistema de tipo de dados do código SQL (com algumas extensões) e utiliza a mesma sintaxe de expressões.



# Estrutura dos Blocos PL/SQL

- DECLARE (opcional)
  - Variáveis, cursores e exceções definidas pelo usuário
- BEGIN (obrigatório)
  - Instruções SQL
  - Instruções PL/SQL
- EXCEPTION (opcional)
  - Ações a serem executadas quando ocorrerem exceções
- END; (obrigatório)



## Estrutura dos Blocos PL/SQL

O slide mostra um bloco PL/SQL básico. Um bloco PL/SQL consiste em quatro seções:

- **Declarativa (opcional):** A seção declarativa começa com a palavra-chave DECLARE e termina quando a seção executável começa.
- **Begin (obrigatório):** A seção executável começa com a palavra-chave BEGIN. Esta seção precisa ter, no mínimo, uma instrução. No entanto, a seção executável de um bloco PL/SQL pode conter inúmeros blocos PL/SQL.
- **Tratamento de exceções (opcional):** A seção de exceção é aninhada dentro da seção executável. Essa seção começa com a palavra-chave EXCEPTION.
- **End (obrigatório):** Todos os blocos PL/SQL devem terminar com uma instrução END. Observe que END é seguido de um ponto e vírgula.

## Estrutura dos Blocos PL/SQL (continuação)

Em um bloco PL/SQL, as palavras-chave `DECLARE`, `BEGIN` e `EXCEPTION` não são seguidas de ponto e vírgula. No entanto, a palavra-chave `END`, todas as instruções SQL e as instruções PL/SQL devem ser seguidas de um ponto e vírgula.

Seção	Descrição	Inclusão
Declarativa ( <code>DECLARE</code> )	Contém declarações das variáveis, das constantes, dos cursores e das exceções definidas pelo usuário que são referenciadas nas seções executável e de exceção	Opcional
Executável ( <code>BEGIN ...</code> <code>END</code> )	Contém instruções SQL para obter dados do banco de dados; contém instruções PL/SQL para manipular dados no bloco	Obrigatória
Exceção ( <code>EXCEPTION</code> )	Especifica as ações a serem realizadas quando ocorrem erros ou condições anormais na seção executável	Opcional

# Agenda

FIAP

- Fundamentos das vantagens e da estrutura do código PL/SQL
- Examinando blocos PL/SQL
- Gerando mensagens de saída em PL/SQL

# Tipos de Blocos

## Procedure

```
PROCEDURE name
IS
BEGIN
    --statements
[EXCEPTION]
END ;
```

## Função

```
FUNCTION name
RETURN datatype
IS
BEGIN
    --statements
    RETURN value;
[EXCEPTION]
END ;
```

## Anônimo

```
[DECLARE]
BEGIN
    --statements
[EXCEPTION]
END ;
```

## Tipos de Blocos

Um programa PL/SQL inclui um ou mais blocos. Esses blocos podem estar totalmente separados ou aninhados em outro bloco.

Existem três tipos de blocos que formam um programa PL/SQL:

- Procedures
- Funções
- Blocos anônimos

**Procedures:** São objetos nomeados que contêm instruções SQL e/ou PL/SQL.

**Funções:** São objetos nomeados que contêm instruções SQL e/ou PL/SQL. Diferentemente de um procedure, uma função retorna um valor de um tipo de dados especificado.

### Blocos anônimos

Blocos anônimos são blocos não nomeados. Eles são declarados em linha no momento da aplicação em que eles deverão ser executados, e serão compilados cada vez que a aplicação for executada. Esses blocos não são armazenados no banco de dados. Eles são passados para o mecanismo PL/SQL para execução durante o runtime. Os triggers dos componentes do Oracle Developer consistem nesses blocos.

Se quiser executar novamente o mesmo bloco, você terá de recriá-lo. Você não poderá chamar o bloco que criou anteriormente porque os blocos são anônimos e não existem mais depois que são

executados.

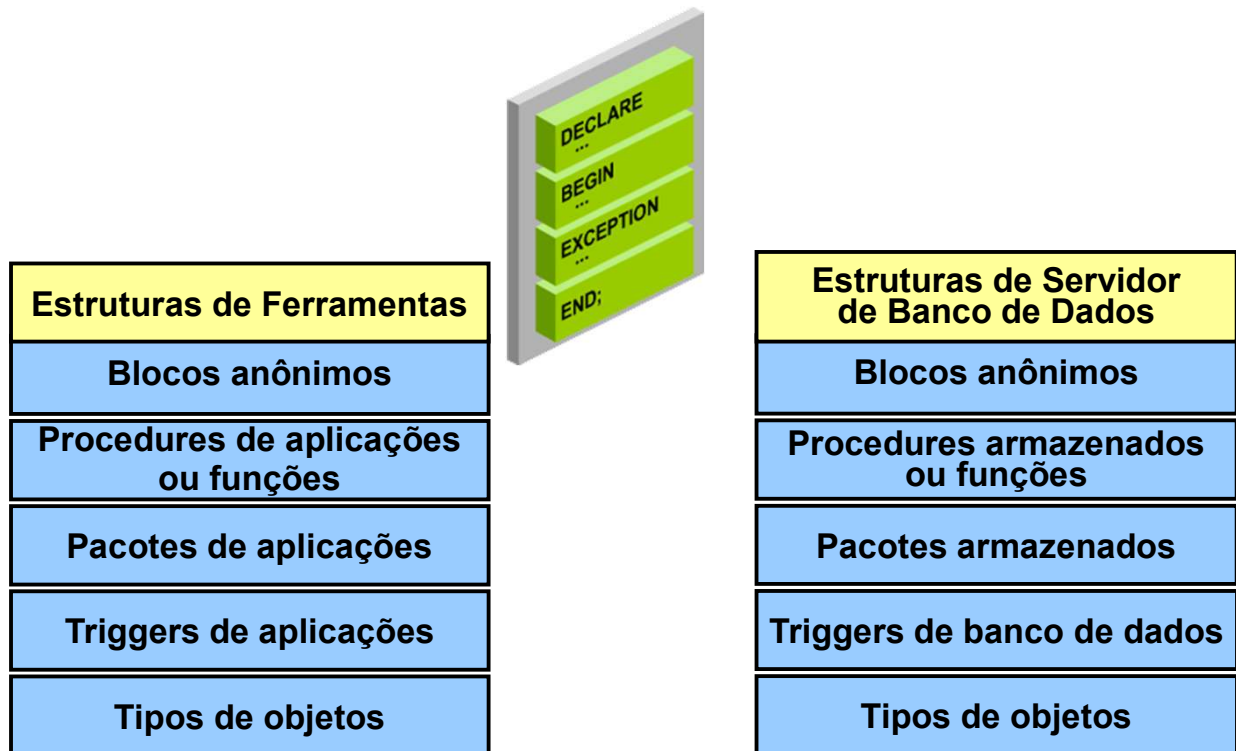
## **Tipos de Blocos (continuação)**

### **Subprogramas**

Os subprogramas complementam os blocos anônimos. Eles são blocos PL/SQL nomeados armazenados no banco de dados. Como eles são nomeados e armazenados, você poderá chamá-los sempre que desejar, dependendo da sua aplicação. Você pode declará-los como procedures ou funções. Em geral, o procedure é usado para realizar uma ação e a função para calcular e retornar um valor.

Os subprogramas podem ser armazenados no nível do servidor ou da aplicação. Usando os componentes do Oracle Developer (Forms, Reports), você pode declarar procedures e funções como parte da aplicação (um form ou relatório) e chamá-los a partir de outros procedures, funções e triggers dentro da mesma aplicação, sempre que necessário.

# Estruturas de Programa



## Estruturas de Programa

A tabela a seguir resume várias estruturas de programa PL/SQL que usam blocos PL/SQL básicos. As estruturas de programa estão disponíveis com base no ambiente em que são executadas.

Bloco de Programa	Descrição	Disponibilidade
Blocos anônimos	Blocos PL/SQL não nomeados incorporados a uma aplicação ou executados interativamente	Todos os ambientes PL/SQL
Functions e procedures de aplicações	Blocos PL/SQL nomeados armazenados em uma aplicação do Oracle Forms Developer ou uma biblioteca compartilhada podem aceitar parâmetros e podem ser chamados várias vezes pelo nome	Componentes de ferramentas do Oracle Developer (por exemplo, Oracle Forms Developer, Oracle Reports)
procedures ou funções armazenados	Blocos PL/SQL nomeados armazenados no servidor Oracle podem aceitar parâmetros e podem ser chamados várias vezes pelo nome	Servidor Oracle ou ferramentas do Oracle Developer
Pacotes (de aplicações ou armazenados)	Módulos PL/SQL nomeados que agrupam procedures, funções e identificadores relacionados	Servidor Oracle e componentes de ferramentas do Oracle Developer (por exemplo, Oracle Forms Developer)

## Examinando um Bloco Anônimo

Um bloco anônimo no espaço de trabalho do SQL Developer:



```
SQL_Wsheet1 History
DECLARE
  v_fname VARCHAR2(20);
BEGIN
  SELECT first_name INTO v_fname FROM employees
  WHERE employee_id=100;
END;
```

19

### Examinando um Bloco Anônimo

Para criar um bloco anônimo usando o SQL Developer, informe o bloco no espaço de trabalho (como mostrado no slide).

#### Exemplo

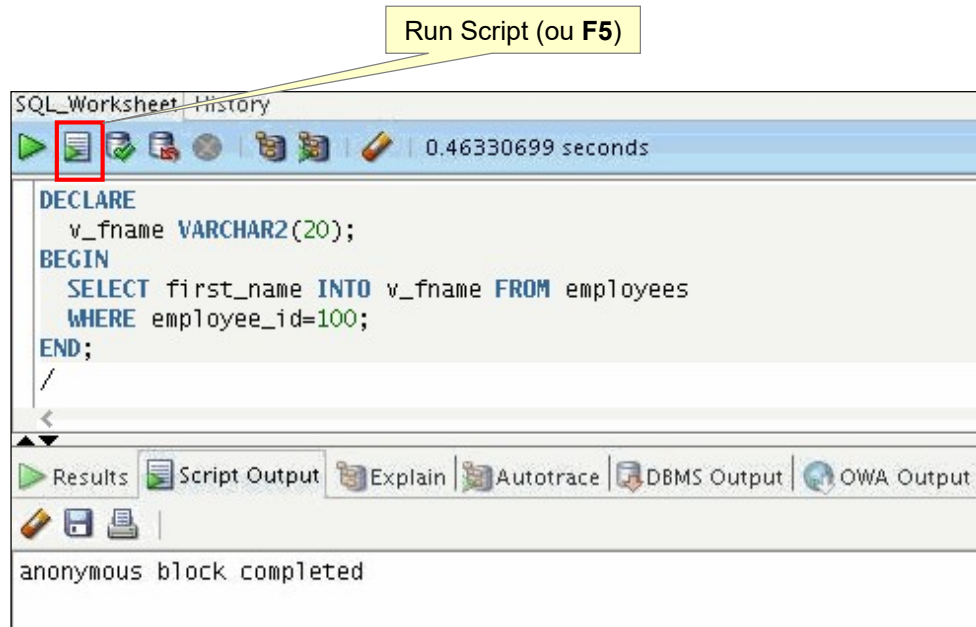
O bloco do exemplo tem a seção declarativa e a executável. Você não precisa prestar atenção à sintaxe das instruções contidas no bloco; você aprenderá a sintaxe posteriormente neste curso.

O bloco anônimo obtém o nome (`first_name`) do funcionário cujo id (`employee_id`) é 100 e armazena-o em uma variável chamada `v_fname`.



## Executando um Bloco Anônimo

Clique no botão Run Script para executar o bloco anônimo:



20

### Executando um Bloco Anônimo

Para executar um bloco anônimo, clique no botão Run Script (ou pressione F5).

**Observação:** A mensagem “anonymous block completed” é exibida na janela Script Output após o bloco ser executado.

# Agenda

FIAP

- Fundamentos das vantagens e da estrutura do código PL/SQL
- Examinando blocos PL/SQL
- Gerando mensagens de saída em PL/SQL

## Habilitando a Saída de um Bloco PL/SQL FIAP

1. Para habilitar a saída no SQL Developer, execute o seguinte comando antes de executar o bloco PL/SQL:

```
SET SERVEROUTPUT ON
```

2. Utilize um pacote Oracle predefinido e seu procedure no bloco anônimo:

```
- DBMS_OUTPUT.PUT_LINE
```

```
DBMS_OUTPUT.PUT_LINE(' The First Name of the  
Employee is ' || v_fname);  
...
```

22

### Habilitando a Saída de um Bloco PL/SQL

No exemplo mostrado no slide anterior, foi armazenado um valor na variável `v_fname`. No entanto, o valor não foi impresso.

O bloco PL/SQL não possui funcionalidade incorporada de entrada ou saída. Dessa forma, é necessário usar pacotes Oracle predefinidos para entrada e saída. Para gerar a saída, faça o seguinte:

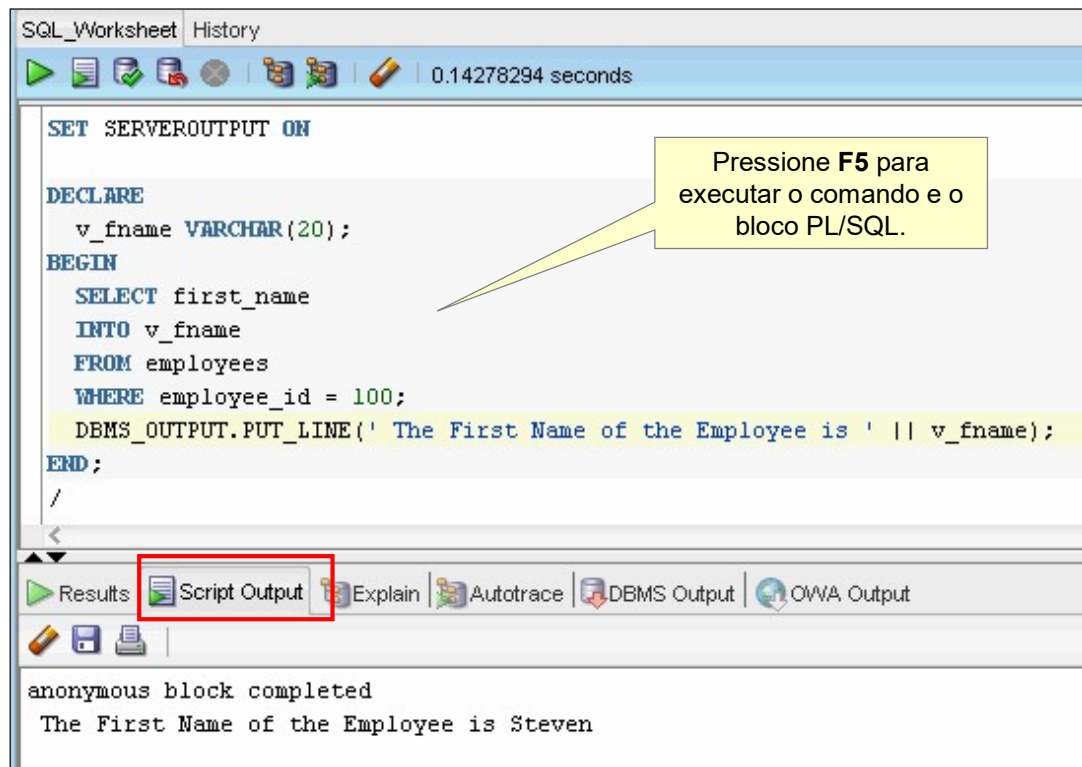
1. Execute este comando:

```
SET SERVEROUTPUT ON
```

**Observação:** Para habilitar a saída no SQL\*Plus, execute explicitamente o comando `SET SERVEROUTPUT ON`.

2. No bloco PL/SQL, use o procedure `PUT_LINE` do pacote `DBMS_OUTPUT` para exibir a saída. Especifique o valor a ser impresso como um argumento para esse procedure (como mostrado no slide). Em seguida, o procedure exibe o argumento.

## Exibindo a Saída de um Bloco PL/SQL



23

### Exibindo a Saída de um Bloco PL/SQL

Pressione F5 (ou clique no ícone Run Script) para exibir a saída do bloco PL/SQL. Ao fazer isso, você:

1. Executa o comando SET SERVEROUTPUT ON
2. Executa o bloco PL/SQL anônimo

A saída é exibida na tab Script Output.