

FIAP GRADUAÇÃO

TSC

Database Application Development & Data Science
Blocos e Exceções

PROFA. PATRICIA ANGELINI profpatria.angelini@fiap.com.br

- ✓ Blocos
- ✓ Exception
- ✓ Exercícios

1. BLOCOS

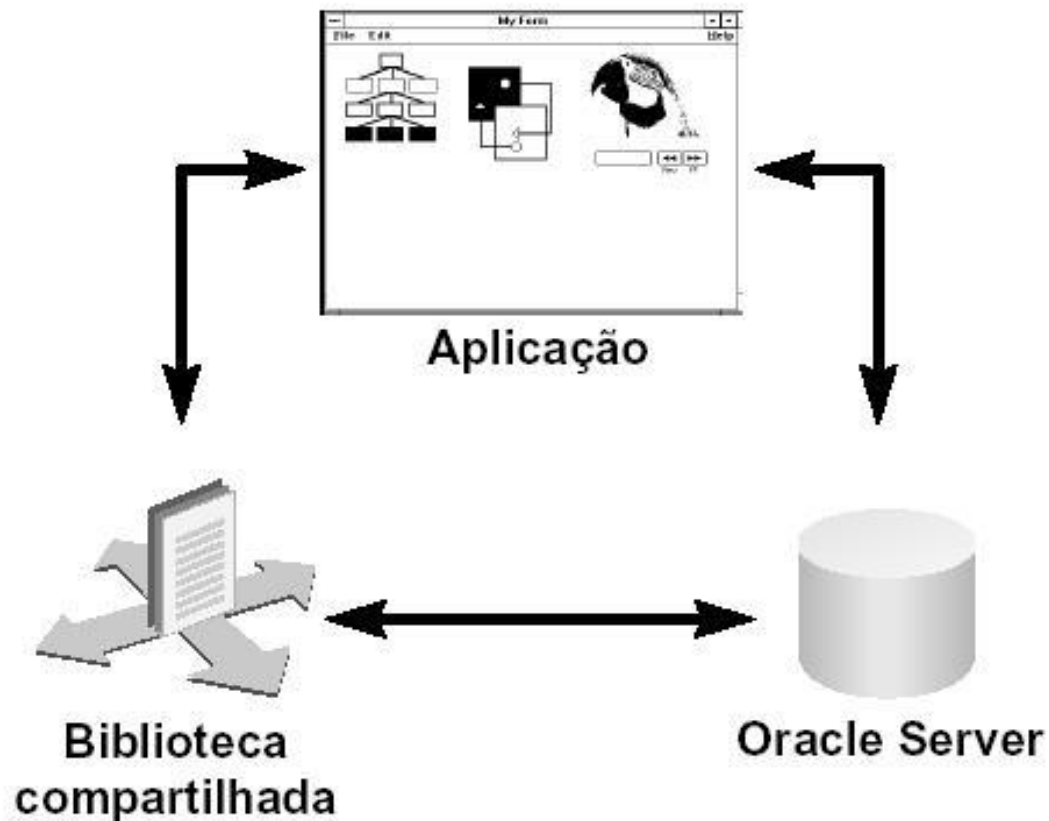
PL/SQL

- É uma extensão da linguagem SQL com recursos de design e linguagem de programação;
- As instruções DML e DQL são incluídas como unidades procedurais no código.

PL/SQL

Benefícios da Linguagem PL/SQL

Integração



ESTRUTURA DE UM BLOCO PL/SQL

- **DECLARE** – Opcional
Variáveis, cursores, exceções definidas pelo usuário
- **BEGIN** – Obrigatório
 - Instruções SQL
 - Instruções PL/SQL
- **EXCEPTION** – Opcional
Ações a serem desempenhadas quando ocorrem erros
- **END;** – Obrigatório



```
DECLARE
...
BEGIN
...
EXCEPTION
...
END;
```

TIPOS DE BLOCO

Tipos de Bloco

Anônimo

```
[DECLARE]

BEGIN
  --statements

[EXCEPTION]

END;
```

Procedimento

```
PROCEDURE name
IS

BEGIN
  --statements

[EXCEPTION]

END;
```

Função

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

END;
```


ESTRUTURA DE UM BLOCO PL/SQL

- A linguagem PL/SQL é uma linguagem estruturada em blocos (blocos lógicos).
- Um bloco consiste de até três seções: declarativa (opcional), executável (necessária) e tratamento de erros (opcional).
- Apenas as palavras chaves BEGIN e END são obrigatórias..

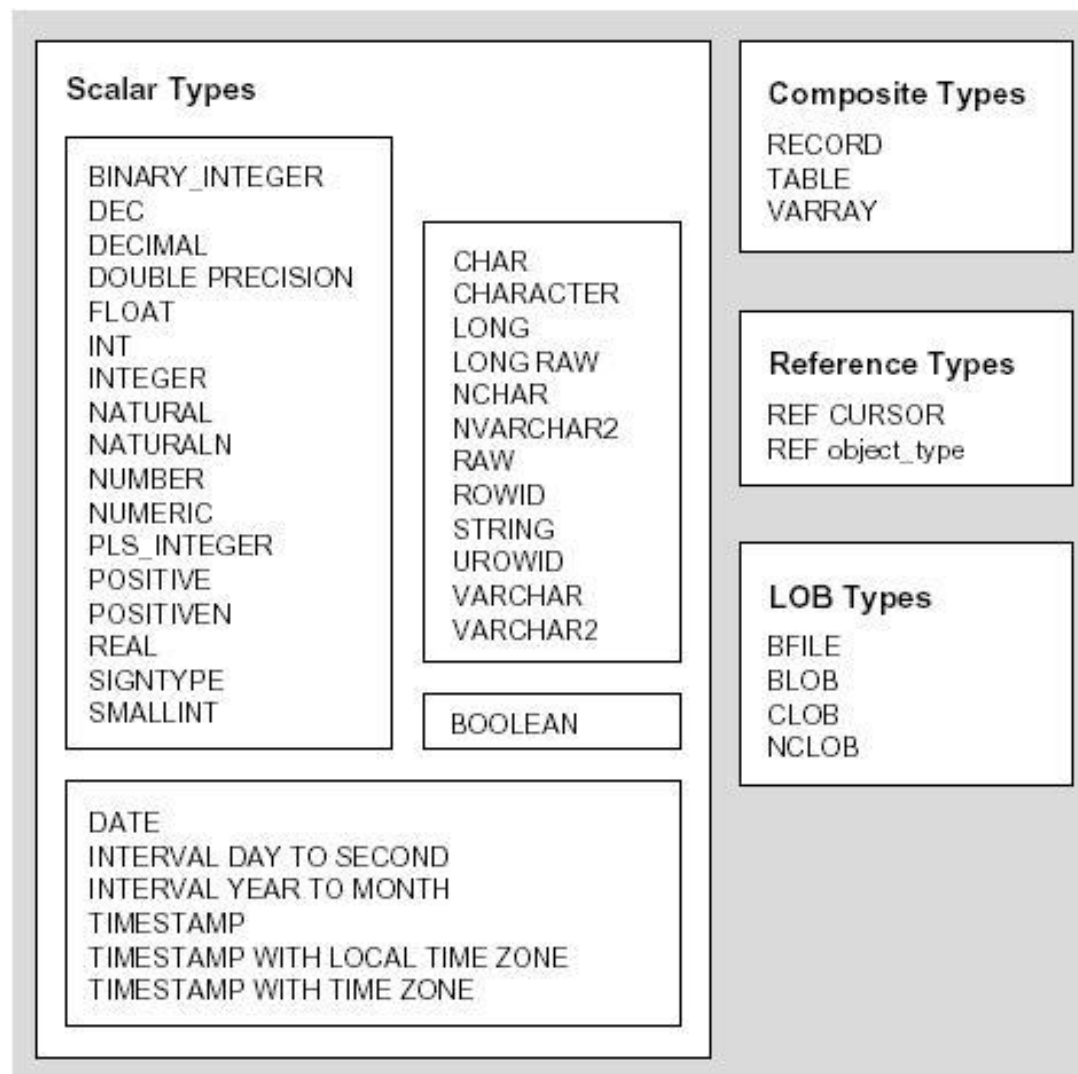
CARACTERES ESPECIAIS

<i>Símbolo</i>	<i>Caracteres</i>
;	Ponto e vírgula: fim de comando
%	Percentagem: indicador de atributo
:	Dois pontos: indicador de variável host
**	Asterisco duplo: operador exponencial
<> e !=	Operador diferente
 	Operador de concatenação
<< e >>	Delimitador de rótulo (label)
:=	Operador de atribuição
=>	Operador de associação para notação posicional
--	Indicador de comentário
/* e */	Início e final de um bloco de comentário

TIPOS DE DADOS BÁSICOS

Tipo de Dados	Descrição
CHAR	Conjunto de caracteres com tamanho fixo.
NCHAR	Conjunto de caracteres com tamanho fixo.
VARCHAR2	Conjunto de caracteres com tamanho variável.
NVARCHAR2	Conjunto de caracteres com tamanho variável.
NUMBER(p,s)	Um número com precisão (p) e escala(s). A precisão varia de 1 a 38 e a escala pode ser -84 a 127.
DATE	Valores de data de 1º janeiro de 4712 A.C. a 31 dezembro de 9999 A.D.
BLOB	Conjunto de binários (utilizado para armazenar imagens).
CLOB	Conjunto de caracteres (utilizado para armazenar arquivos).
TIMESTAMP	Ano, mês, dia, hora, minuto, segundo e segundos fracionários. O valor de segundos fracionários pode variar de 0 a 9 (precisão até um bilionésimo de segundo). O padrão é 6 (um milionésimo).

LISTA COMPLETA DOS TIPOS DE DADOS EM PL/SQL



DECLARAÇÃO DE VARIÁVEIS EM PL/SQL

Declarando Variáveis PL/SQL

Sintaxe

```
identificador [CONSTANT] tipo de dados [NOT NULL]  
[:= | DEFAULT expr];
```

Exemplos

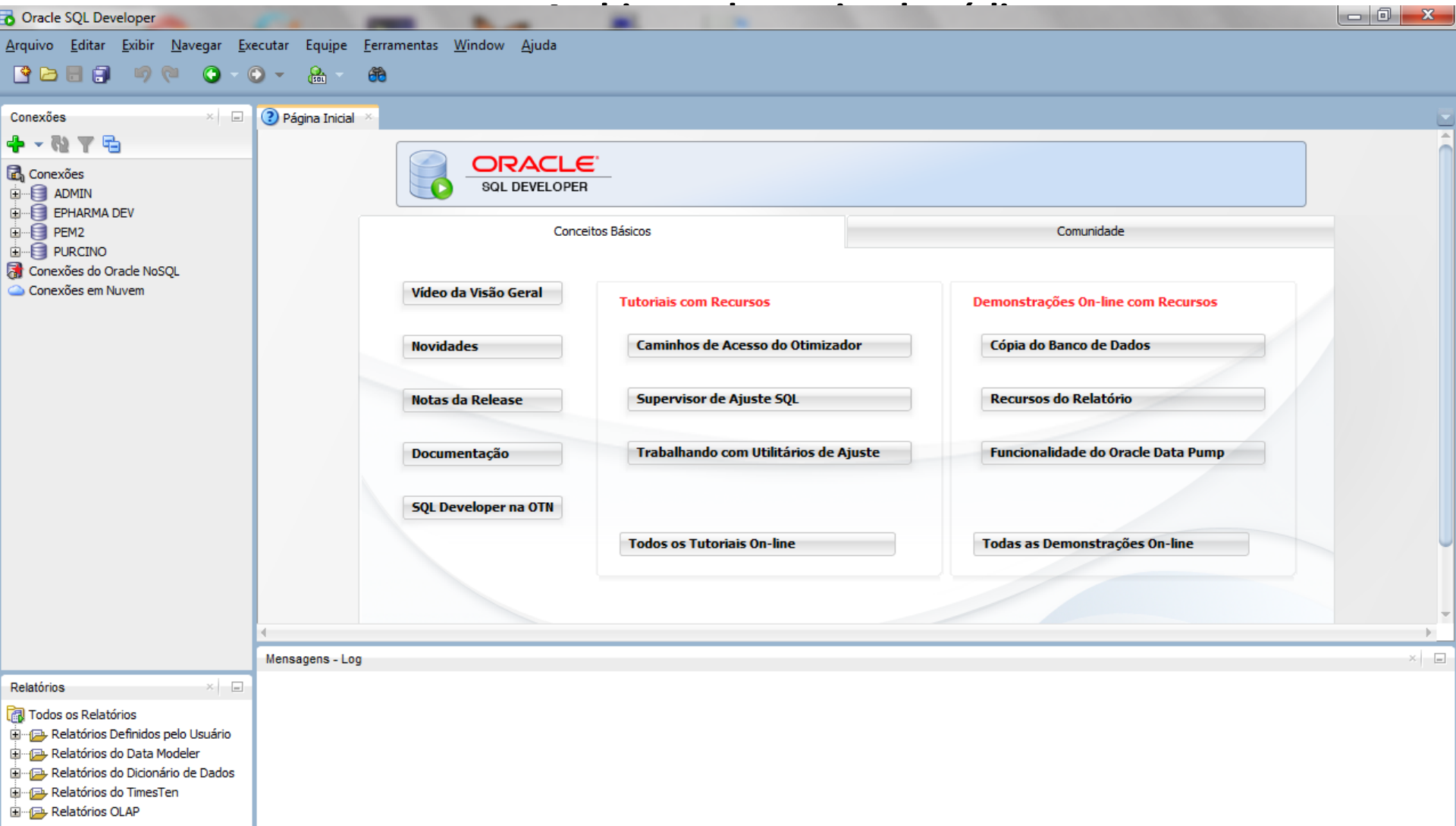
```
Declare  
  v_hiredate      DATE;  
  v_deptno        NUMBER(2) NOT NULL := 10;  
  v_location      VARCHAR2(13) := 'Atlanta';  
  c_comm          CONSTANT NUMBER := 1400;
```

Declarando Variáveis Escalares

Exemplos

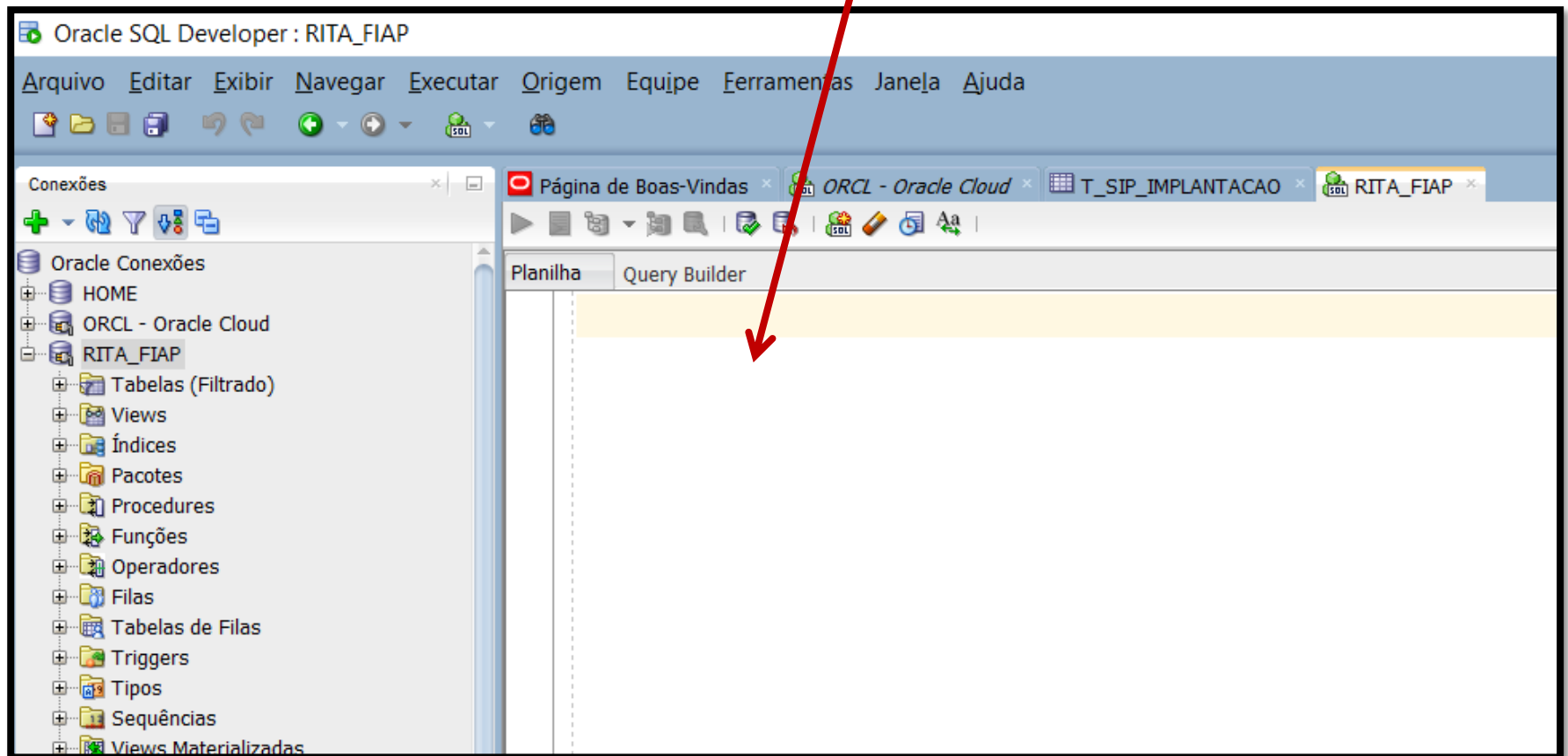
```
v_job      VARCHAR2(9);  
v_count    BINARY_INTEGER := 0;  
v_total_sal NUMBER(9,2) := 0;  
v_orderdate DATE := SYSDATE + 7;  
c_tax_rate CONSTANT NUMBER(3,2) := 8.25;  
v_valid    BOOLEAN NOT NULL := TRUE;
```

Ferramenta: SQL DEVELOPER

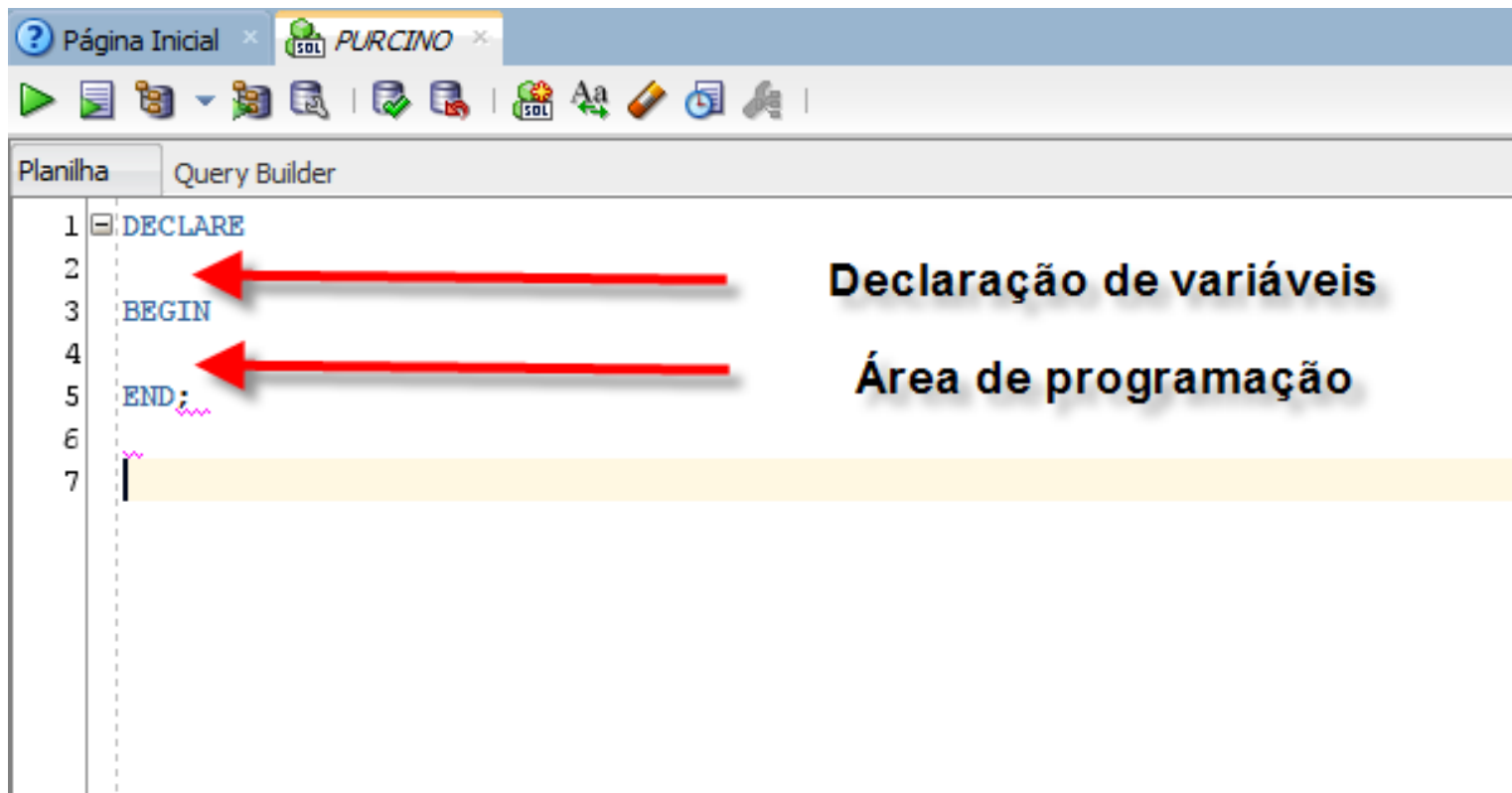


Ferramenta: SQL DEVELOPER

Ambiente de escrita de código

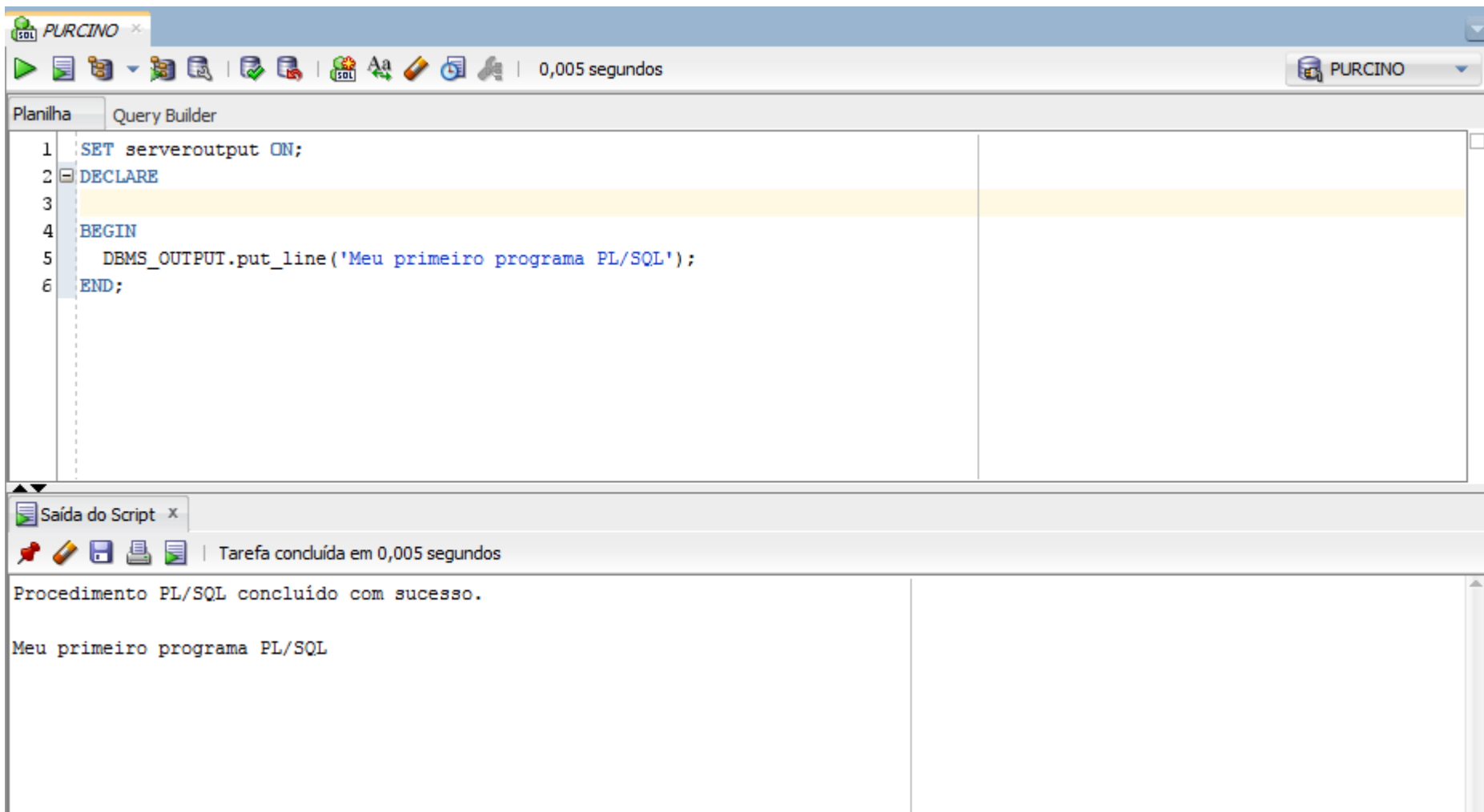


Ferramenta: SQL DEVELOPER



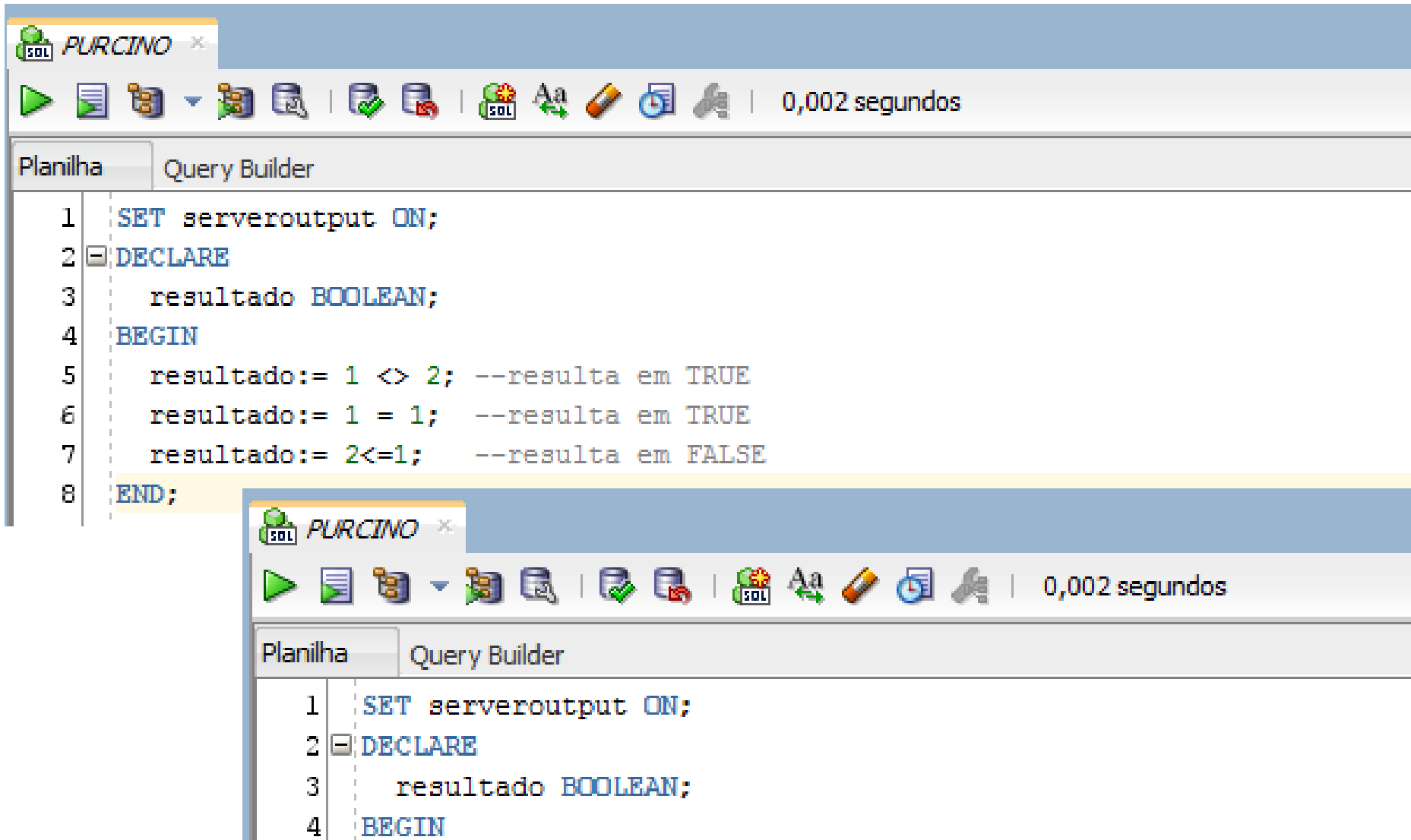
Ferramenta: SQL DEVELOPER

Meu primeiro programa em PL/SQL



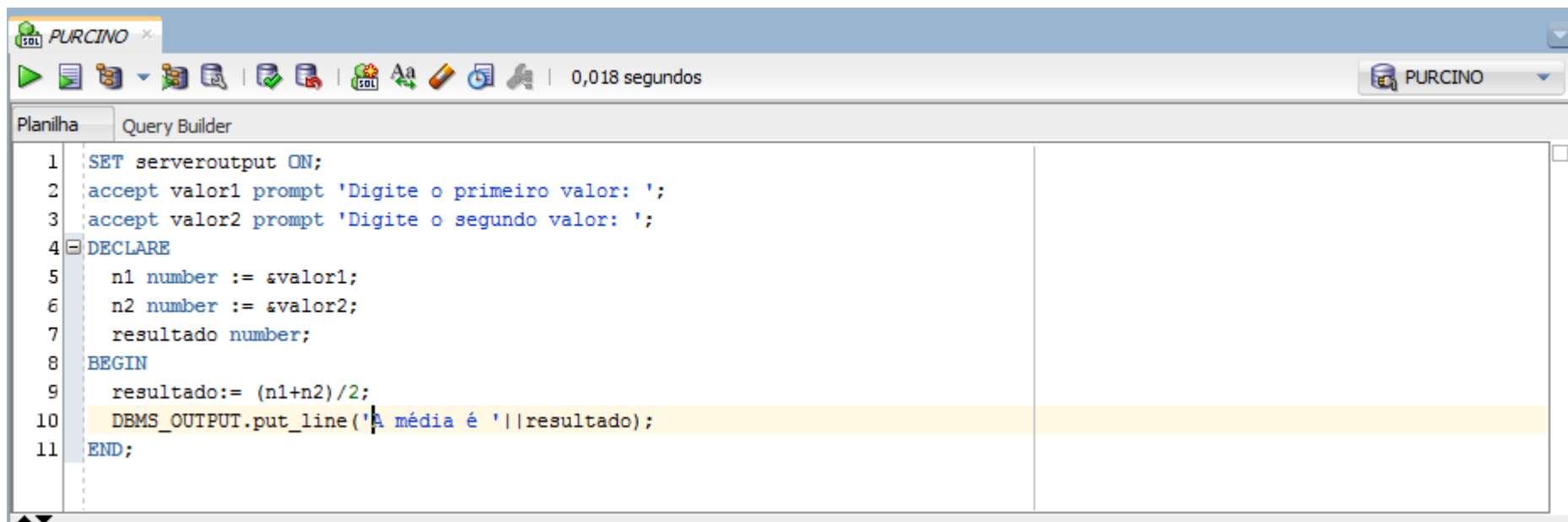
Ferramenta: SQL DEVELOPER

Programa resultado de operações lógicas



Ferramenta: SQL DEVELOPER

Um programa que calcula a média

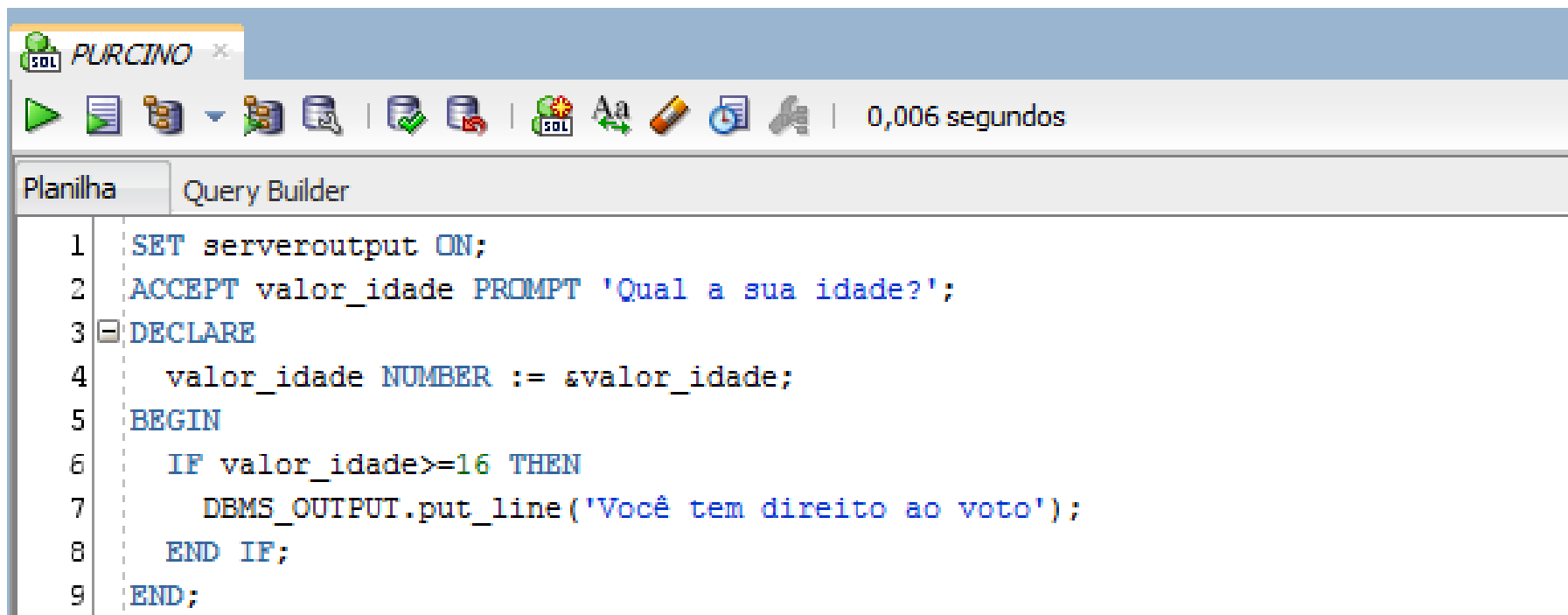


The screenshot shows the SQL Developer application window. The title bar reads 'PURCINO -x'. The toolbar contains various icons for execution, saving, and editing, along with a timer showing '0,018 segundos'. The 'Planilha' (Worksheet) tab is active, displaying a PL/SQL script in the 'Query Builder' area. The script is as follows:

```
1 SET serveroutput ON;
2 accept valor1 prompt 'Digite o primeiro valor: ';
3 accept valor2 prompt 'Digite o segundo valor: ';
4 DECLARE
5     n1 number := &valor1;
6     n2 number := &valor2;
7     resultado number;
8 BEGIN
9     resultado:= (n1+n2)/2;
10    DBMS_OUTPUT.put_line('A média é '||resultado);
11 END;
```

Ferramenta: SQL DEVELOPER

Um programa de desvio condicional simples

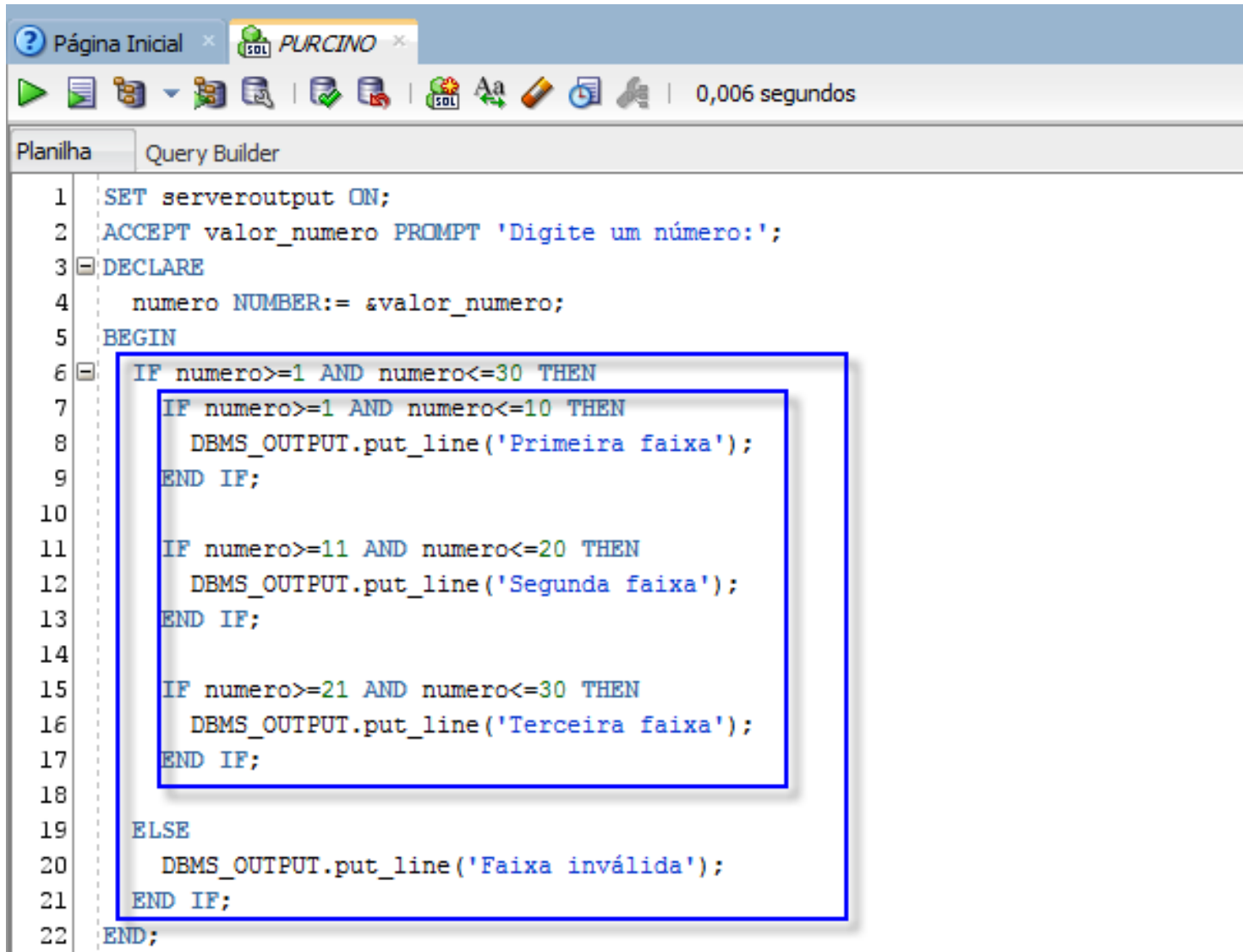


The screenshot displays the SQL Developer application window. The title bar shows the file name 'PURCINO'. The toolbar includes icons for running, saving, and other database operations, along with a timer showing '0,006 segundos'. The 'Query Builder' tab is active, showing a PL/SQL script with the following code:

```
1 SET serveroutput ON;
2 ACCEPT valor_idade PROMPT 'Qual a sua idade?';
3 DECLARE
4     valor_idade NUMBER := &valor_idade;
5 BEGIN
6     IF valor_idade >= 16 THEN
7         DBMS_OUTPUT.put_line('Você tem direito ao voto');
8     END IF;
9 END;
```

Ferramenta: SQL DEVELOPER

Programa para calculo da faixa de valores



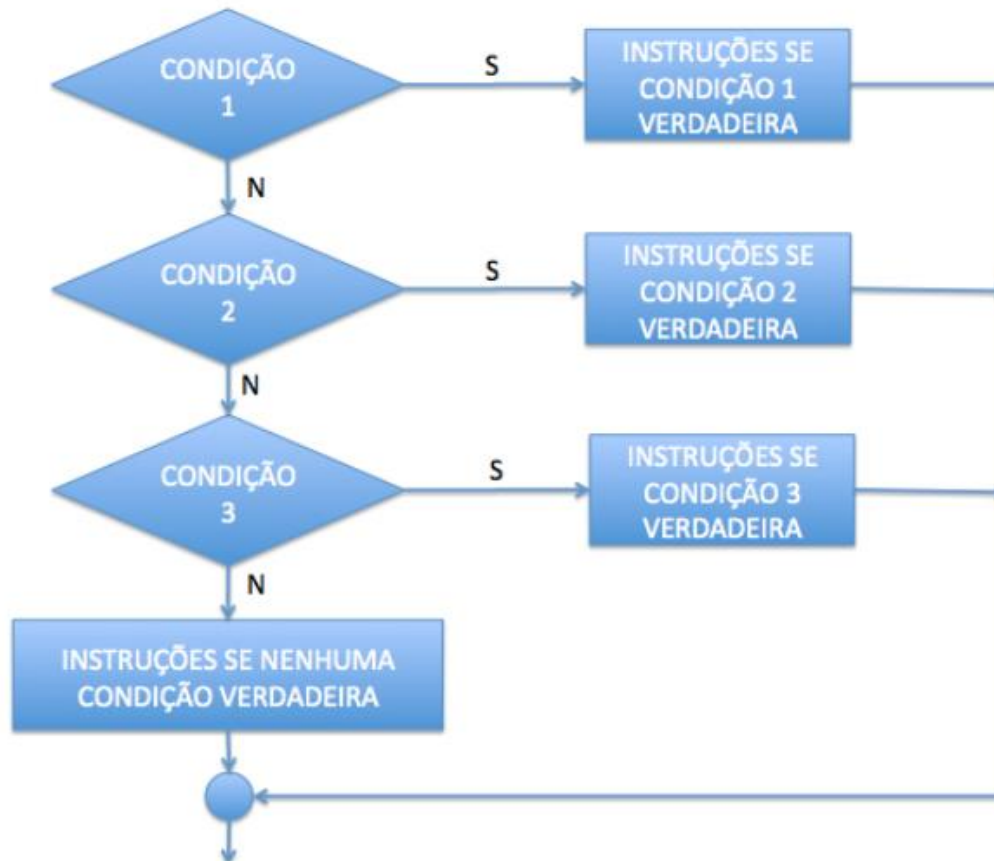
The screenshot displays the SQL Developer application window. The title bar shows 'Página Inicial' and a tab for 'PURCINO'. The toolbar includes icons for running, saving, and other database operations, along with a timer showing '0,006 segundos'. The 'Planilha' (Worksheet) tab is active, showing a SQL script. The script is a PL/SQL block that prompts the user for a number and then categorizes it into three ranges: 1-10 (Primeira faixa), 11-20 (Segunda faixa), and 21-30 (Terceira faixa). If the number is outside these ranges, it outputs 'Faixa inválida'. The script is enclosed in a blue box.

```
1 SET serveroutput ON;
2 ACCEPT valor_numero PROMPT 'Digite um número: ';
3 DECLARE
4     numero NUMBER := &valor_numero;
5 BEGIN
6     IF numero >= 1 AND numero <= 30 THEN
7         IF numero >= 1 AND numero <= 10 THEN
8             DBMS_OUTPUT.put_line('Primeira faixa');
9         END IF;
10
11        IF numero >= 11 AND numero <= 20 THEN
12            DBMS_OUTPUT.put_line('Segunda faixa');
13        END IF;
14
15        IF numero >= 21 AND numero <= 30 THEN
16            DBMS_OUTPUT.put_line('Terceira faixa');
17        END IF;
18
19    ELSE
20        DBMS_OUTPUT.put_line('Faixa inválida');
21    END IF;
22 END;
```

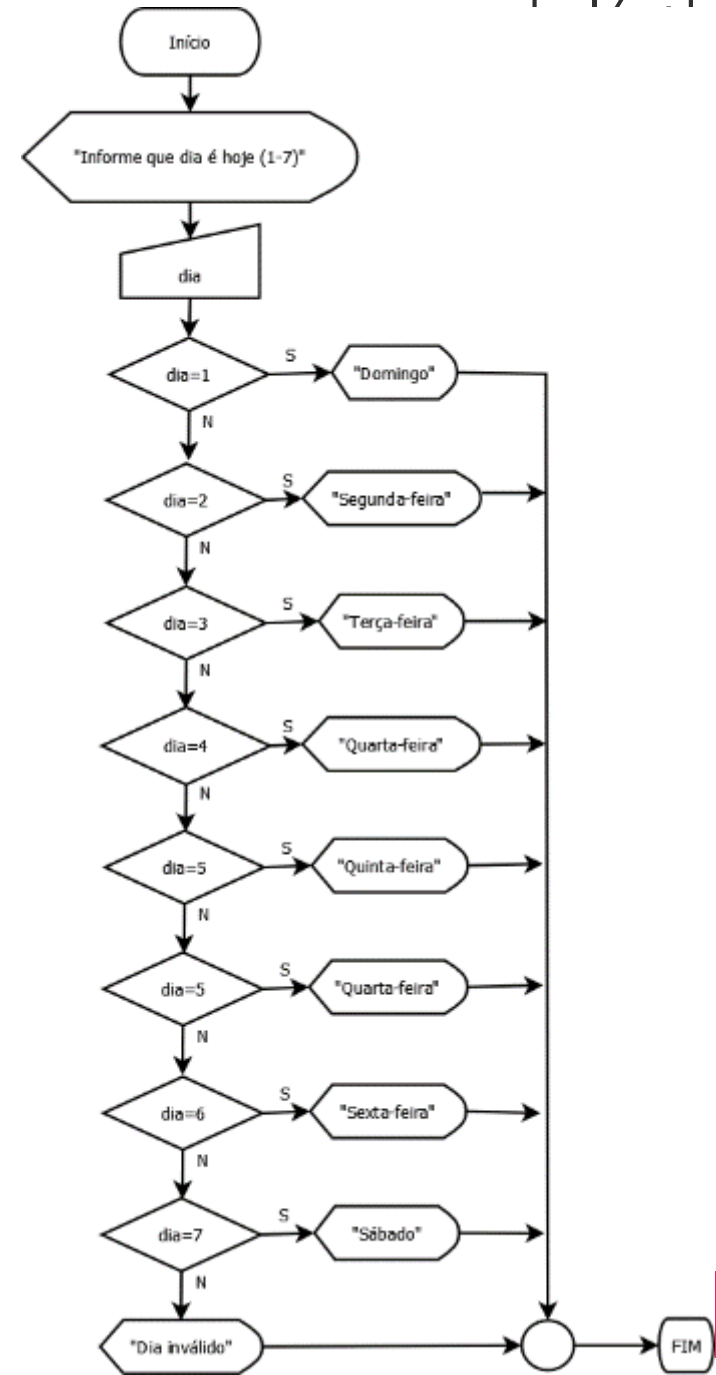
2. ESTRUTURA DECISÃO SELEÇÃO

Estrutura de Decisão por Seleção – o case / when

- A estrutura de decisão por seleção é usada quando temos um grande número de alternativas.

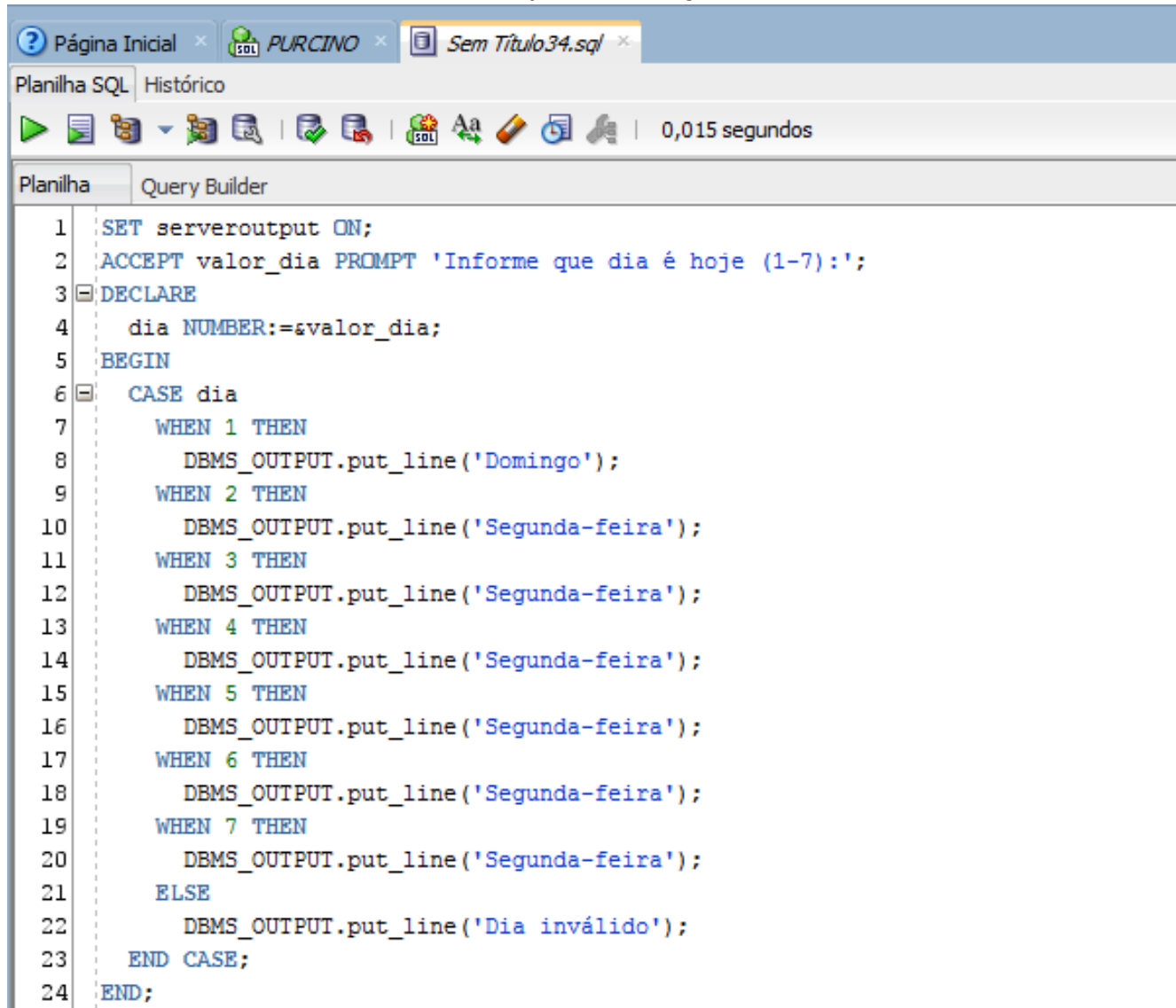


EXEMPLO USANDO DIAGRAMA DE BLOCOS



Ferramenta: SQL DEVELOPER

Programa com estrutura de decisão por seleção

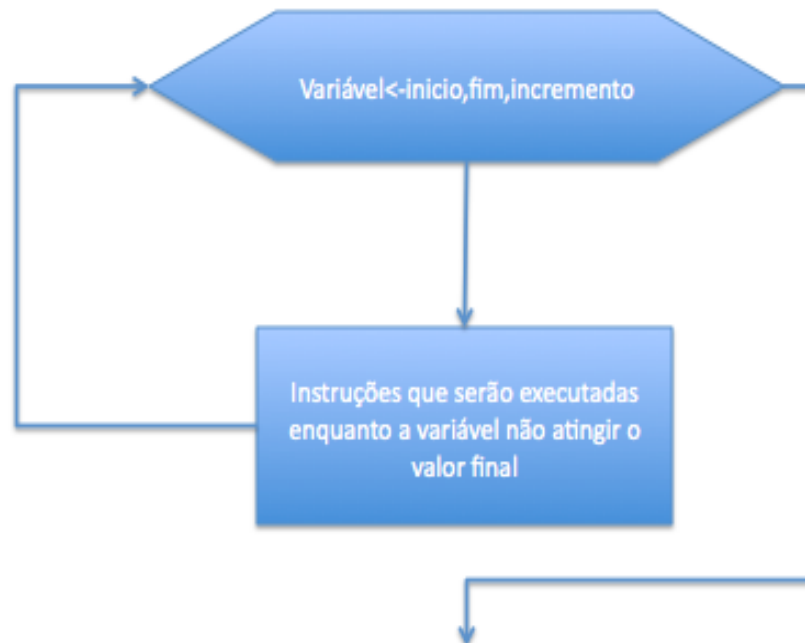


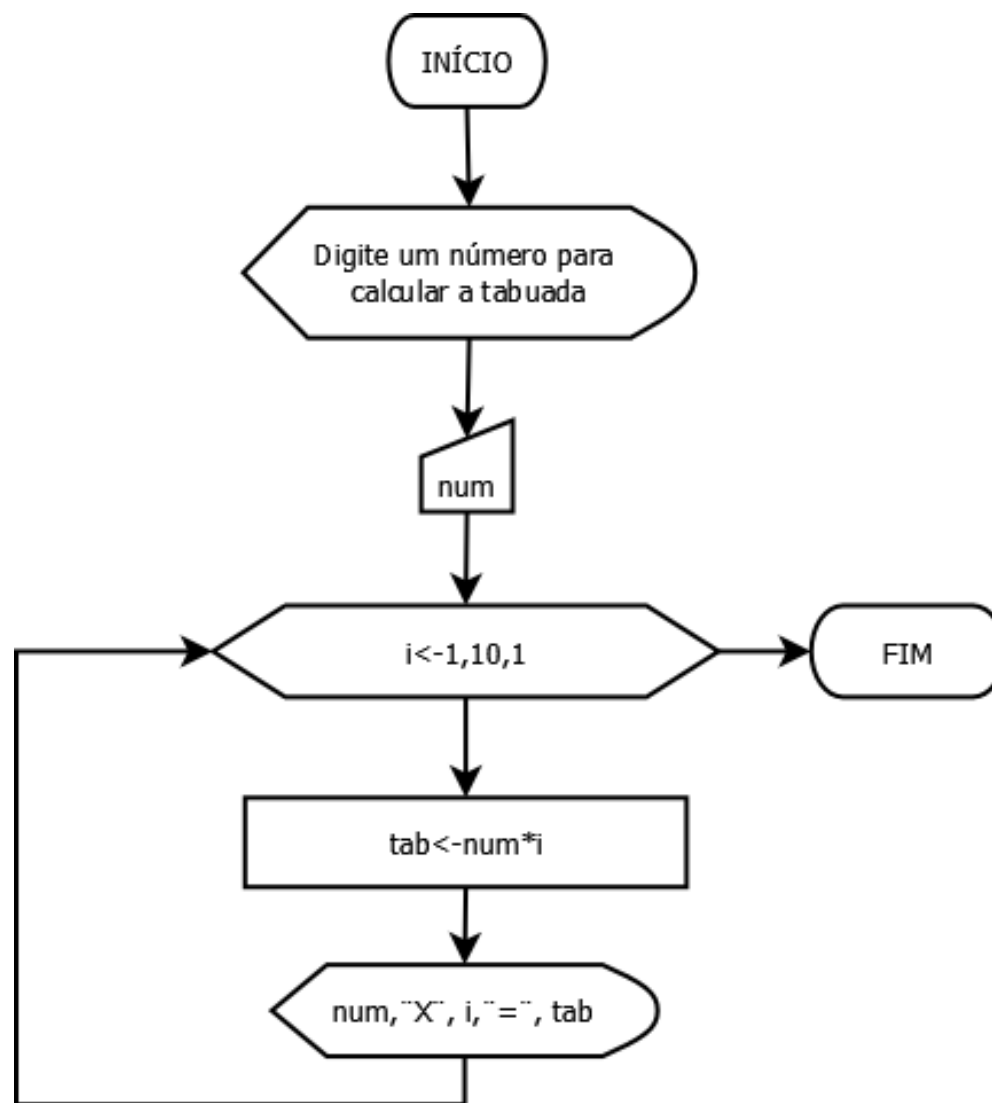
The screenshot displays the SQL Developer application window. The title bar shows three tabs: 'Página Inicial', 'PURCINO', and 'Sem Título34.sql'. Below the tabs is a toolbar with various icons for file operations, execution, and editing. The main workspace is divided into two panes: 'Planilha SQL' (active) and 'Histórico'. The 'Planilha SQL' pane shows a PL/SQL script with the following code:

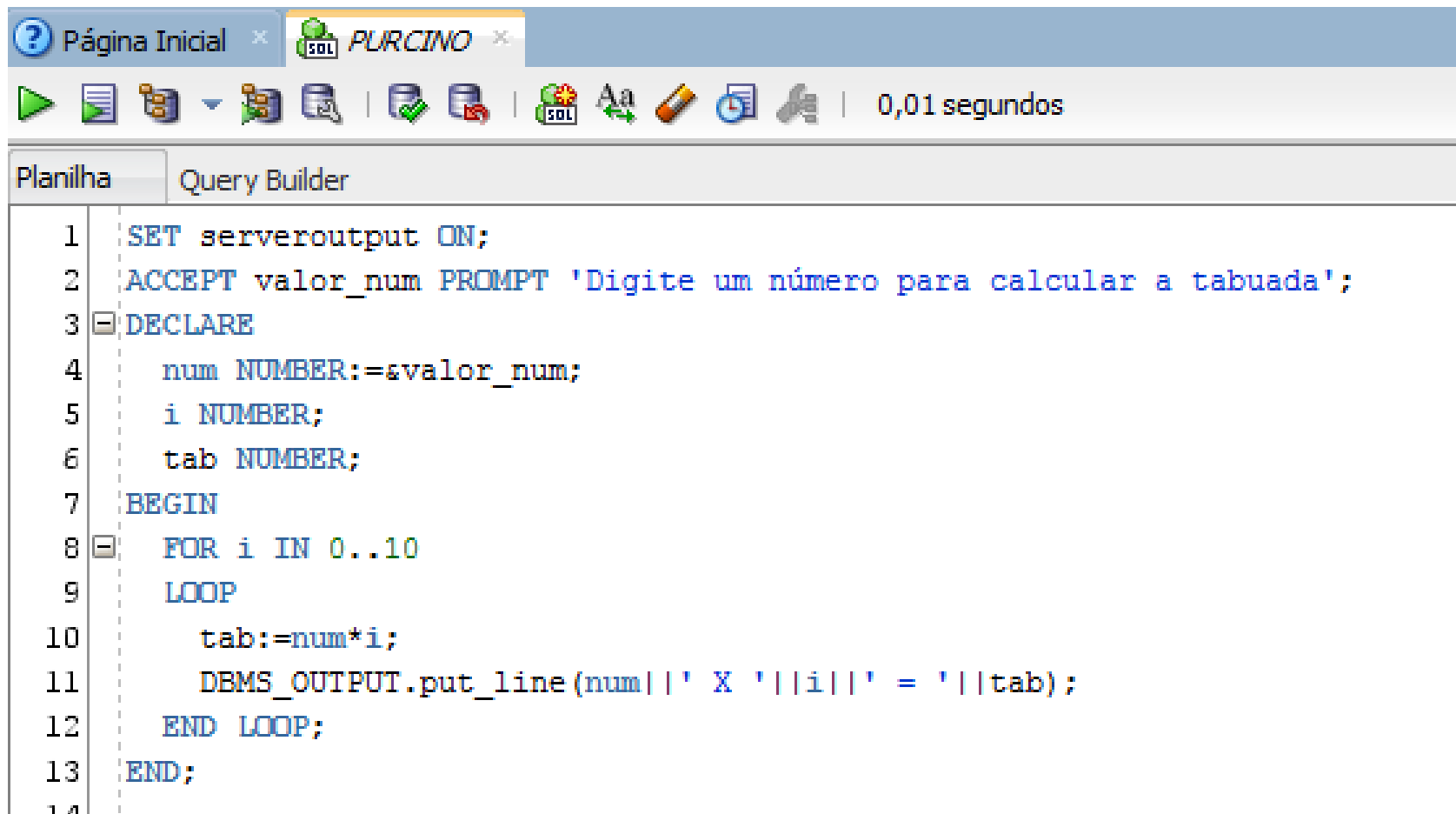
```
1 SET serveroutput ON;
2 ACCEPT valor_dia PROMPT 'Informe que dia é hoje (1-7):';
3 DECLARE
4     dia NUMBER:=&valor_dia;
5 BEGIN
6     CASE dia
7         WHEN 1 THEN
8             DBMS_OUTPUT.put_line('Domingo');
9         WHEN 2 THEN
10            DBMS_OUTPUT.put_line('Segunda-feira');
11        WHEN 3 THEN
12            DBMS_OUTPUT.put_line('Segunda-feira');
13        WHEN 4 THEN
14            DBMS_OUTPUT.put_line('Segunda-feira');
15        WHEN 5 THEN
16            DBMS_OUTPUT.put_line('Segunda-feira');
17        WHEN 6 THEN
18            DBMS_OUTPUT.put_line('Segunda-feira');
19        WHEN 7 THEN
20            DBMS_OUTPUT.put_line('Segunda-feira');
21        ELSE
22            DBMS_OUTPUT.put_line('Dia inválido');
23    END CASE;
24 END;
```

3. ESTRUTURAS REPETIÇÃO

- Esta é uma estrutura de repetição iterativa, ou seja, que não depende de nenhuma ação do usuário, por isso é denominada incondicional.
- Esta estrutura tem a quantidade de vezes limitada e controlada por uma variável contadora. Essa variável contadora recebe um valor inicial, um valor final e qual o passo, que significa qual o valor que será incrementado nesta variável na próxima execução do bloco. O passo define se o limite máximo será atingido de 1 em 1, de 2 em 2, de 3 em 3 e assim por diante.



















The screenshot shows a SQL IDE window with a tab labeled 'PURCINO'. The toolbar includes icons for running, saving, and other database operations, along with a timer showing '0,01 segundos'. The main editor area is titled 'Query Builder' and contains the following PL/SQL code:

```
1  SET serveroutput ON;
2  ACCEPT valor_num PROMPT 'Digite um número para calcular a tabuada';
3  DECLARE
4      num NUMBER:=&valor_num;
5      i NUMBER;
6      tab NUMBER;
7  BEGIN
8      FOR i IN 0..10
9      LOOP
10         tab:=num*i;
11         DBMS_OUTPUT.put_line(num||' X '||i||' = '||tab);
12     END LOOP;
13 END;
```

- Podemos obter os dados de uma tabela utilizando loops e a pseudocoluna ROWNUM:

Planilha		Query Builder	
1		CREATE VIEW VIEW_LIVROS AS	
2		SELECT ROWNUM AS LINHA, TX_TITULO FROM T_AU5_LIVRO;	
3			
5			
6		SELECT * FROM VIEW_LIVROS WHERE LINHA = 6;	
7			

Planilha SQL Histórico

            | 0,016 segundos

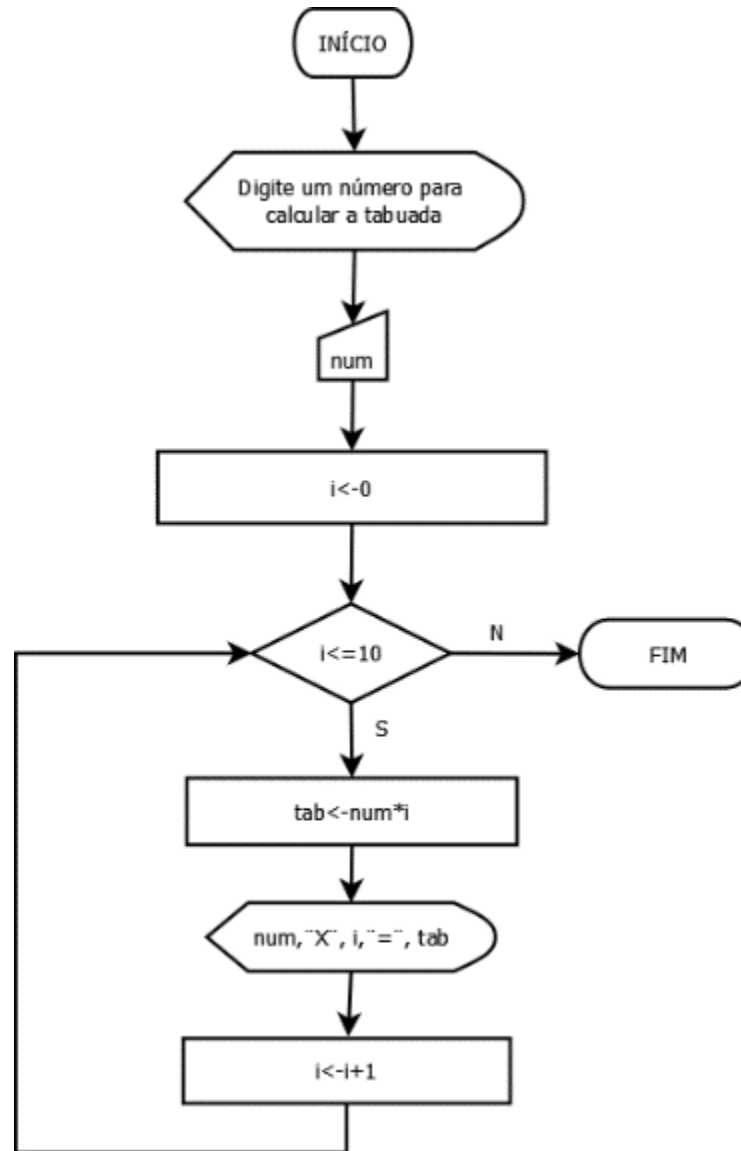
Planilha Query Builder

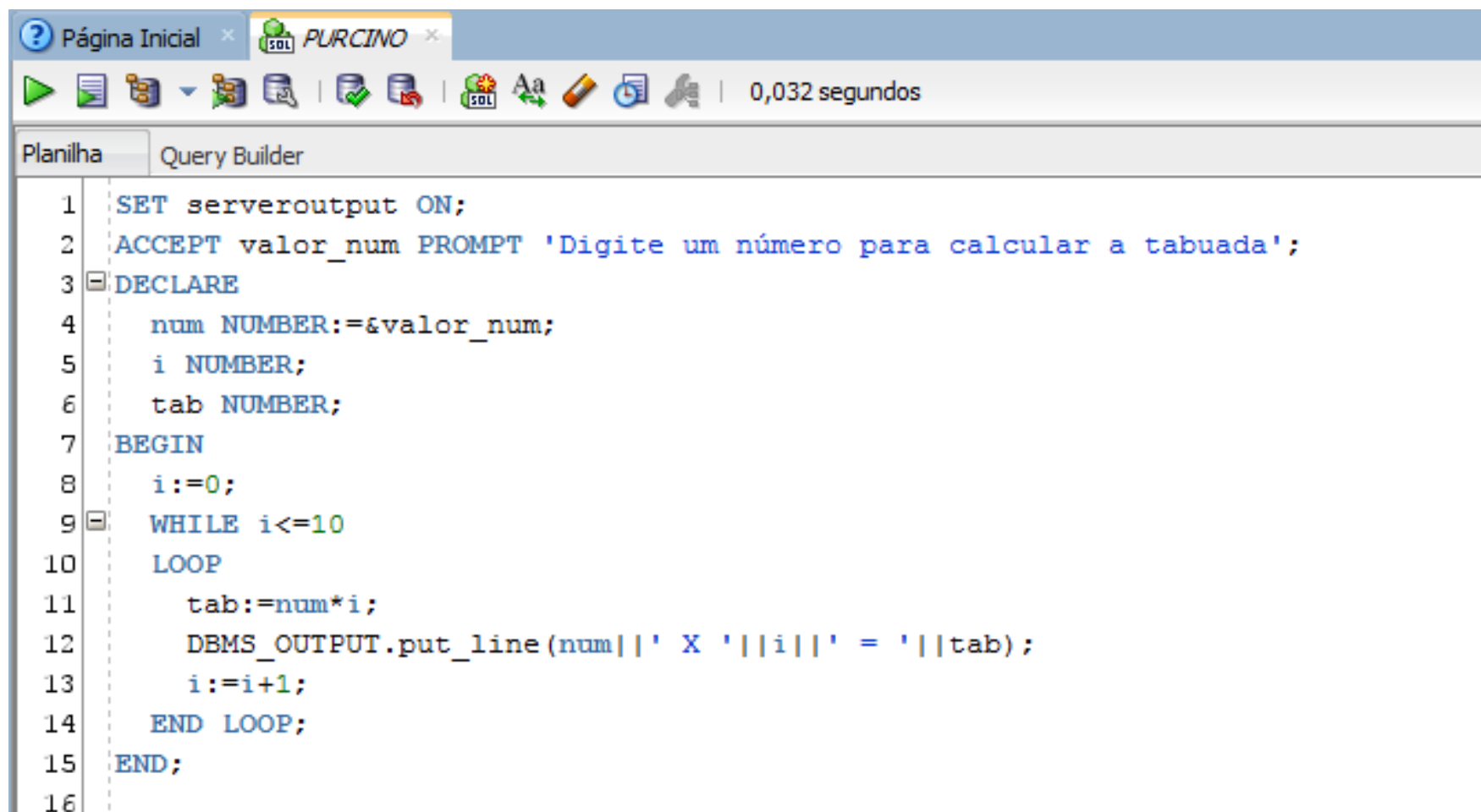
```
1 DECLARE
2     n NUMBER;
3     total NUMBER;
4     titulo VARCHAR(50);
5 BEGIN
6     SELECT COUNT(*) into total FROM VIEW_LIVROS;
7     FOR n IN 1..total
8     LOOP
9         SELECT tx_titulo INTO titulo FROM VIEW_LIVROS WHERE LINHA = n;
10        DBMS_OUTPUT.put_line(titulo);
11    END LOOP;
12 END;
```

- Esta é uma estrutura de repetição que pode ser iterativa ou interativa, ou seja, pode ou não depender de uma ação do usuário.
- Esta estrutura tem a quantidade de vezes ilimitada e controlada por uma expressão lógica. Sempre que esta expressão lógica for verdadeira o bloco de código será executado.




```
WHILE condições  
LOOP  
    {...instruções...}  
END LOOP;
```





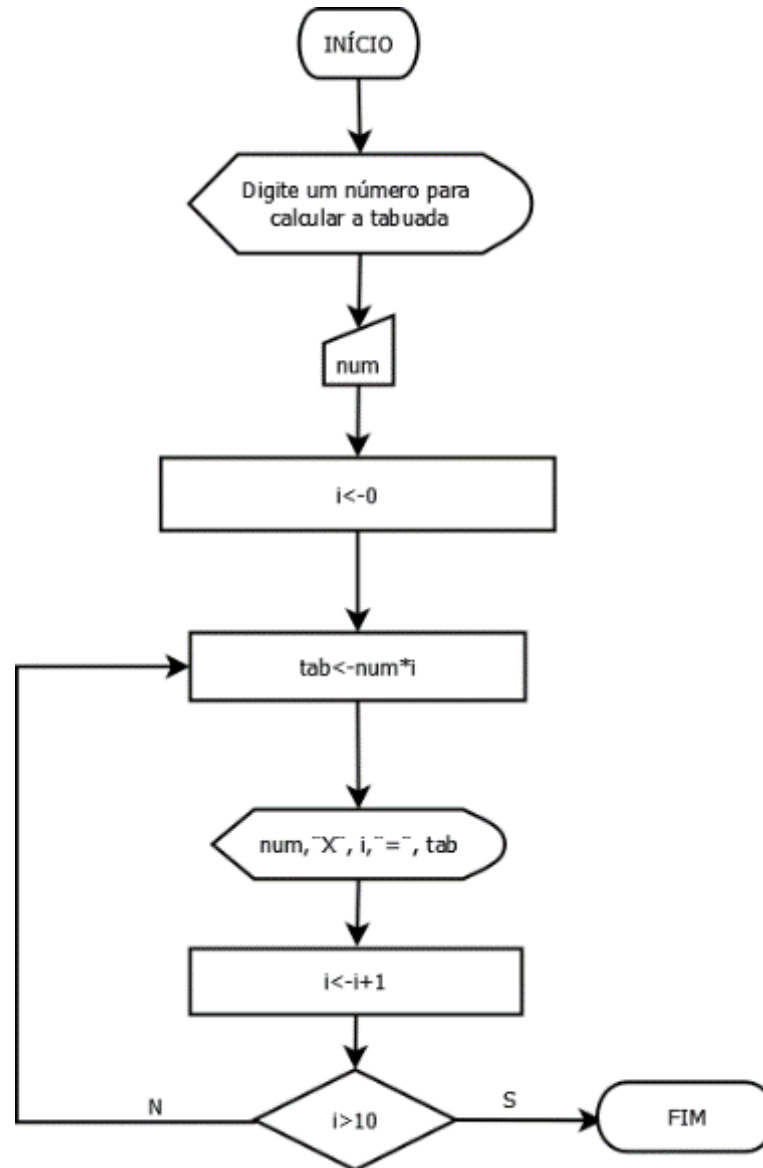
The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for running, saving, and other database operations, along with a timer showing 0,032 segundos. The main window is titled 'Planilha' and 'Query Builder'. The script being edited is as follows:

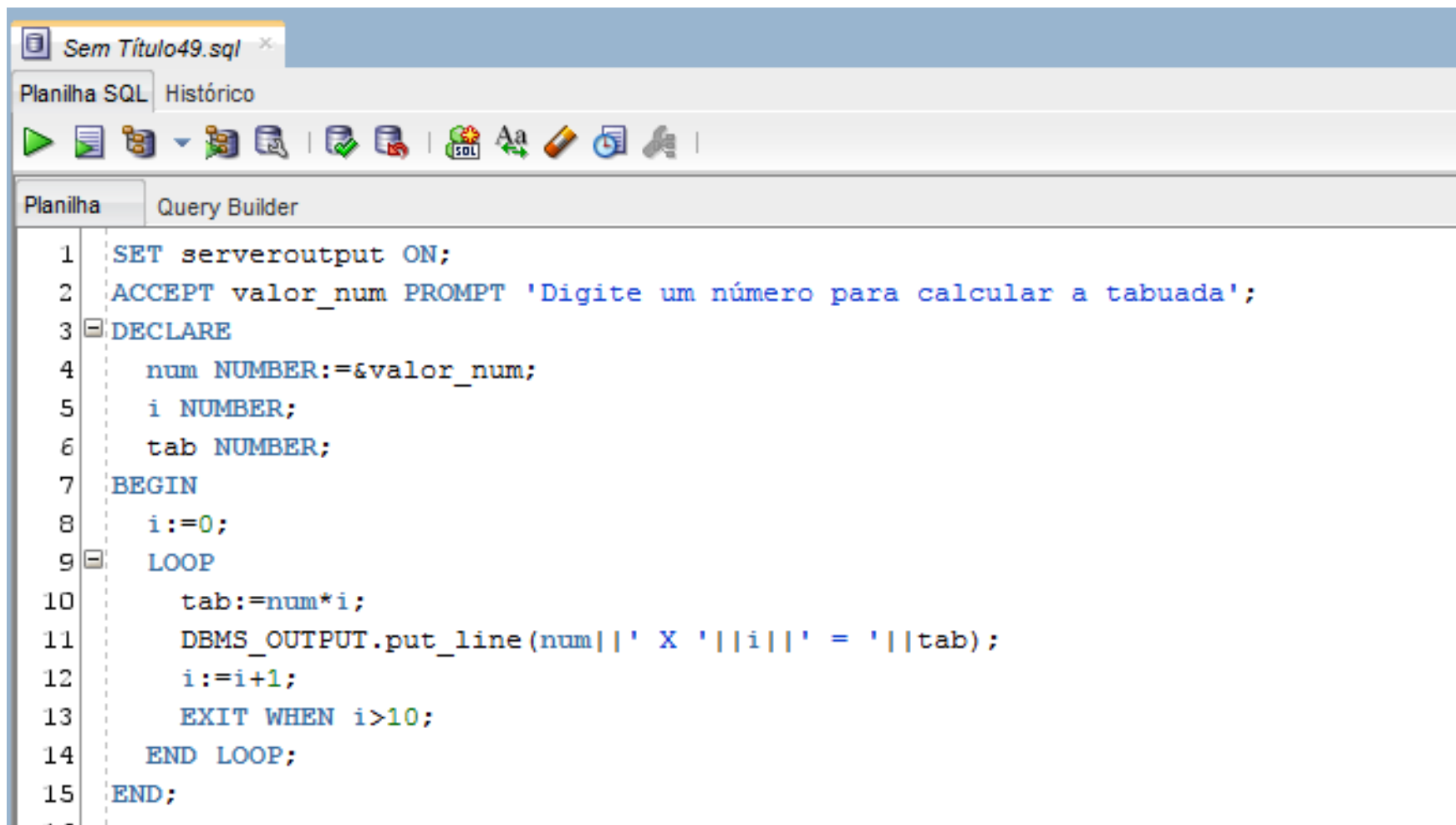
```
1 SET serveroutput ON;
2 ACCEPT valor_num PROMPT 'Digite um número para calcular a tabuada';
3 DECLARE
4     num NUMBER:=&valor_num;
5     i NUMBER;
6     tab NUMBER;
7 BEGIN
8     i:=0;
9     WHILE i<=10
10    LOOP
11        tab:=num*i;
12        DBMS_OUTPUT.put_line(num||' X '||i||' = '||tab);
13        i:=i+1;
14    END LOOP;
15 END;
```

- Esta é uma estrutura de repetição que pode ser iterativa ou interativa, ou seja, pode ou não depender de uma ação do usuário.
- Esta estrutura executa ao menos uma vez o bloco de código e valida uma expressão lógica. Sempre que esta expressão lógica for verdadeira o bloco de código será repetido.



```
LOOP
    {...instruções...}
    EXIT [ WHEN condição ];
END LOOP;
```





The screenshot shows a SQL IDE window titled "Sem Título49.sql". The interface includes a toolbar with icons for running, saving, and editing. The main editor area displays a PL/SQL script with line numbers 1 through 15. The script sets the server output to ON, prompts the user for a number, and then uses a loop to calculate and display a multiplication table for that number.

```
1 SET serveroutput ON;
2 ACCEPT valor_num PROMPT 'Digite um número para calcular a tabuada';
3 DECLARE
4     num NUMBER:=&valor_num;
5     i NUMBER;
6     tab NUMBER;
7 BEGIN
8     i:=0;
9     LOOP
10         tab:=num*i;
11         DBMS_OUTPUT.put_line(num||' X '||i||' = '||tab);
12         i:=i+1;
13         EXIT WHEN i>10;
14     END LOOP;
15 END;
```

EXERCÍCIOS

- Escreva um algoritmo que peça um número e verifique se ele é par
- Escreva um algoritmo que pergunte em que país a pessoa nasceu (Brasil, Mexico, Canadá) e diga se é America do Sul, Central ou Norte
- O jogo do PIM era uma brincadeira conhecida do Silvio Santos em seu programa de auditório que consistia em pedir a alguém que recite uma sequência numérica iniciada em 1 e caso o número seja múltiplo de quatro deveria substituí-lo pela palavra PIM. Faça um programa que escreva na tela uma sequência de 1 a 30 substituindo os múltiplos de quatro pela palavra PIM.
- Escreva um algoritmo que solicite o saldo de uma conta corrente e qual o valor da prestação do carro. O algoritmo deve escrever sucessivas vezes o novo saldo até que seja insuficiente para pagar no seguinte formato de exemplo:

```
Qual o saldo da conta corrente? 1650,00
Qual o valor da prestação do carro? 400,00
```

```
Ao final do mês 1 você terá 1250,00
Ao final do mês 2 você terá 850,00
Ao final do mês 3 você terá 450,00
Ao final do mês 4 você terá 50,00
A partir do mês 5 você não conseguirá mais pagar a prestação
```

4. EXCEÇÕES

EXCEPTIONS

- Exceções são todos os erros ou imprevistos que possam ocorrer durante a execução de um bloco PL/SQL.
- Quando ocorre um erro, o SGBDR Oracle abandona a área de comandos(aborta a execução) e procura em uma área de exceções(EXCEPTION), o tratamento para a falha.
- Erros inesperados ocorridos em tempo real de execução são geralmente decorrentes de uma falha na codificação do programa fonte, falha de hardware, leitura inconsistente de um determinado conteúdo entre outras origens
- Como não é possível antecipar a todos os erros, o planejamento para capturá-los são imprescindíveis em aplicações críticas onde não se permite interrupções

EXCEPTIONS

- EM PL/SQL, uma condição de erro é chamada de exception (exceção)
- Uma exception pode ser internamente definida ou definida por uma determinada regra na fase de concepção do código fonte
- Como se realiza o tratamento da exceção ?
 - Através de comandos implementados no código PL/SQL

RECUPERANDO OS CÓDIGOS DE ERRO DO ORACLE

- Para recuperar qualquer código de erro e sua respectiva mensagem relativo no acesso ao banco de dados Oracle, pode-se usar as funções internas SQLCODE e SQLERRM
- SQLCODE – Retorna o código do erro Oracle
- SQLERRM – Retorna o código e mensagem de erro Oracle
- O tamanho máximo da mensagem a ser exibida é de 512 caracteres incluindo o código de erro Oracle
- Se não ocorrer erro a mensagem exibida é: **ORA-0000: normal, successful completion.**

BLOCO DE EXCEÇÃO Estrutura de Bloco PL/SQL

- **DECLARE** – Opcional
Variáveis, cursores, exceções definidas pelo usuário
- **BEGIN** – Obrigatório
 - Instruções SQL
 - Instruções PL/SQL
- **EXCEPTION** – Opcional
Ações a serem desempenhadas quando ocorrem erros
- **END;** – Obrigatório

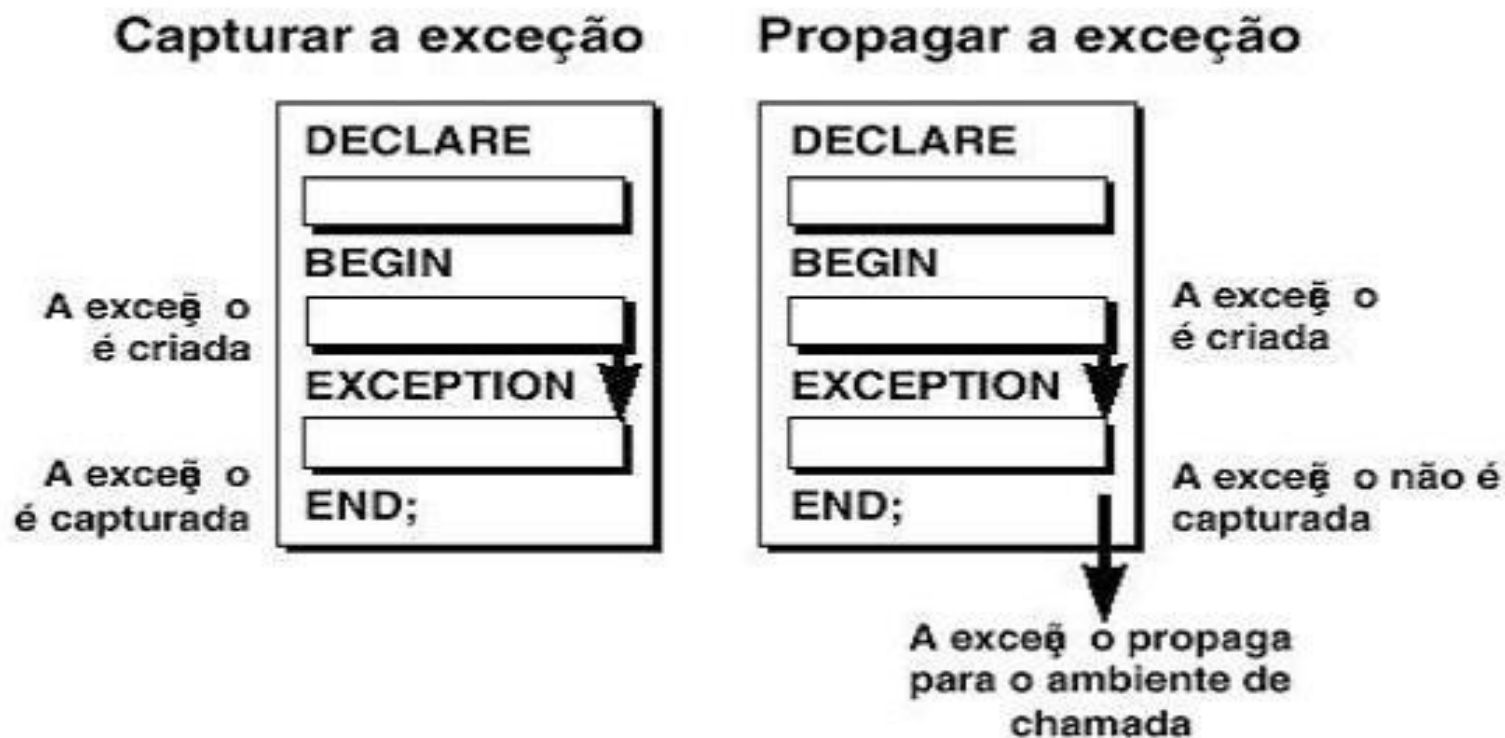
```

DECLARE
  ○ ○ ○
BEGIN
  ○ ○ ○
EXCEPTION
  ○ ○ ○
END;
```

Seção	Descrição	Inclusão
Declarativa	Contém todas as variáveis, constantes, cursores e exceções definidas pelo usuário que são referenciadas nas seções executável e declarativa	Opcional
Executável	Contém instruções SQL para manipular dados no banco de dados e instruções PL/SQL para manipular dados no bloco	Obrigatória
Tratamento de exceção	Especifica as ações a desempenhar quando erros e condições anormais surgem na seção executável	Opcional

PROPAGAÇÃO DAS EXCEÇÕES

Tratando Exceções



EXCEÇÕES DE SISTEMA E EXCEÇÕES DE USUÁRIO

- *EM PL/SQL, uma condição de erro é chamada de exception (exceção)*
- *Uma exception pode ser internamente definida ou definida por uma determinada regra na fase de concepção do código fonte*
- *Ex: WHEN_NO_DATA_FOUND é uma exceção pré-definida em PL/SQL internamente. CLIENTE_SEM_CREDITO pode ser uma exceção definida pela regra de negócio do projeto CARLOCA*

COMANDO RAISE

- *Quando um erro ocorre uma exceção é acionada, isto é, a execução normal é interrompida e o controle é transferido para a área de tratamento de exceções, que é uma parte de seu bloco PL/SQL*
- *A chamada de uma exceção implícita é feita automaticamente pelo PL/SQL. Já a chamada de uma exceção definida é chamada pelo comando RAISE*
- *Exemplo de chamada de uma execução definida pela regra de negócio:
RAISE CLIENTE_SEM_CREDITO*

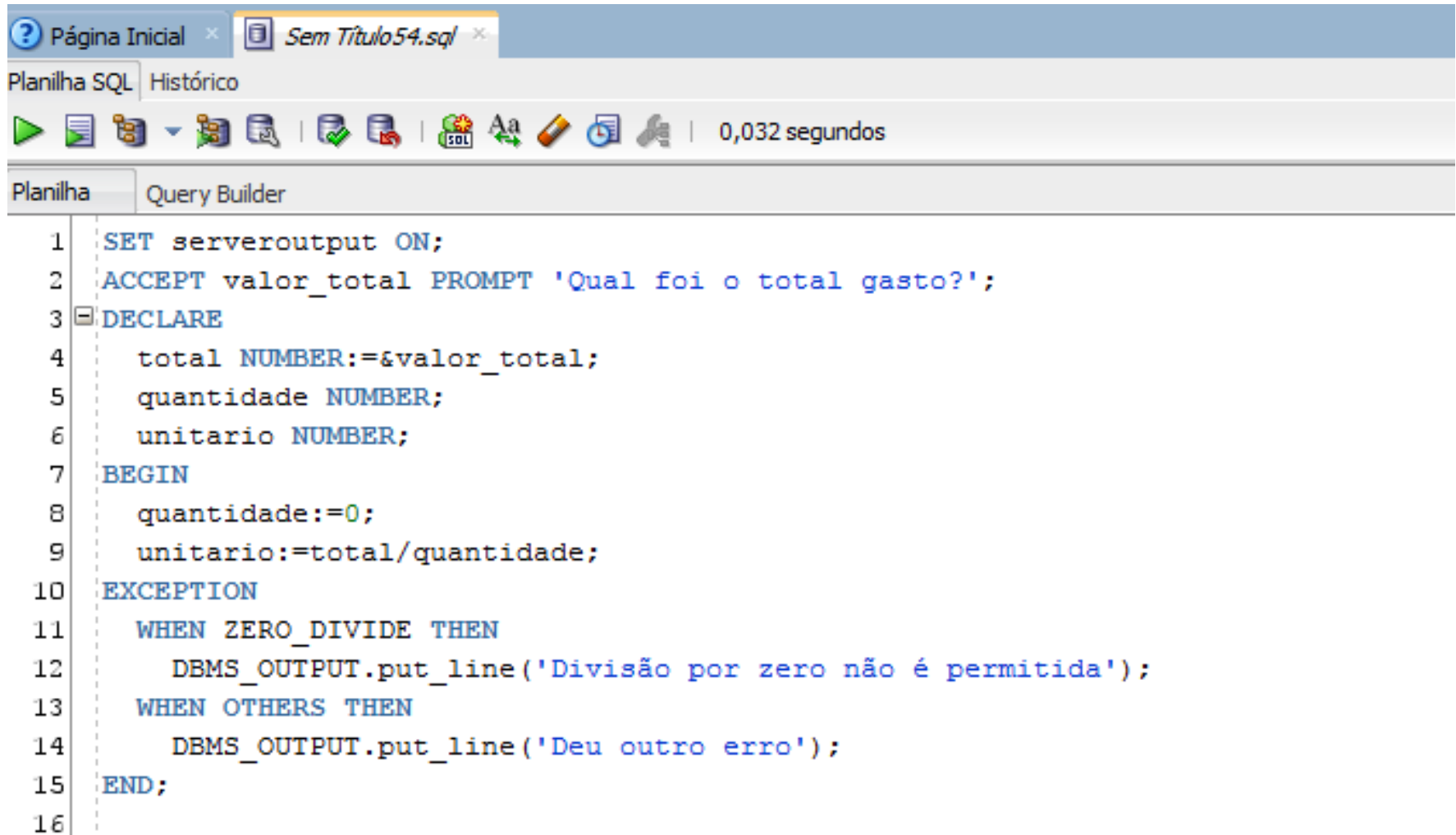
EXCEÇÕES PRÉ-DEFINIDAS

Exceção Predefinida

Sintaxe

```
BEGIN
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    statement1;
    statement2;
  WHEN TOO_MANY_ROWS THEN
    statement1;
  WHEN OTHERS THEN
    statement1;
    statement2;
    statement3;
END;
```

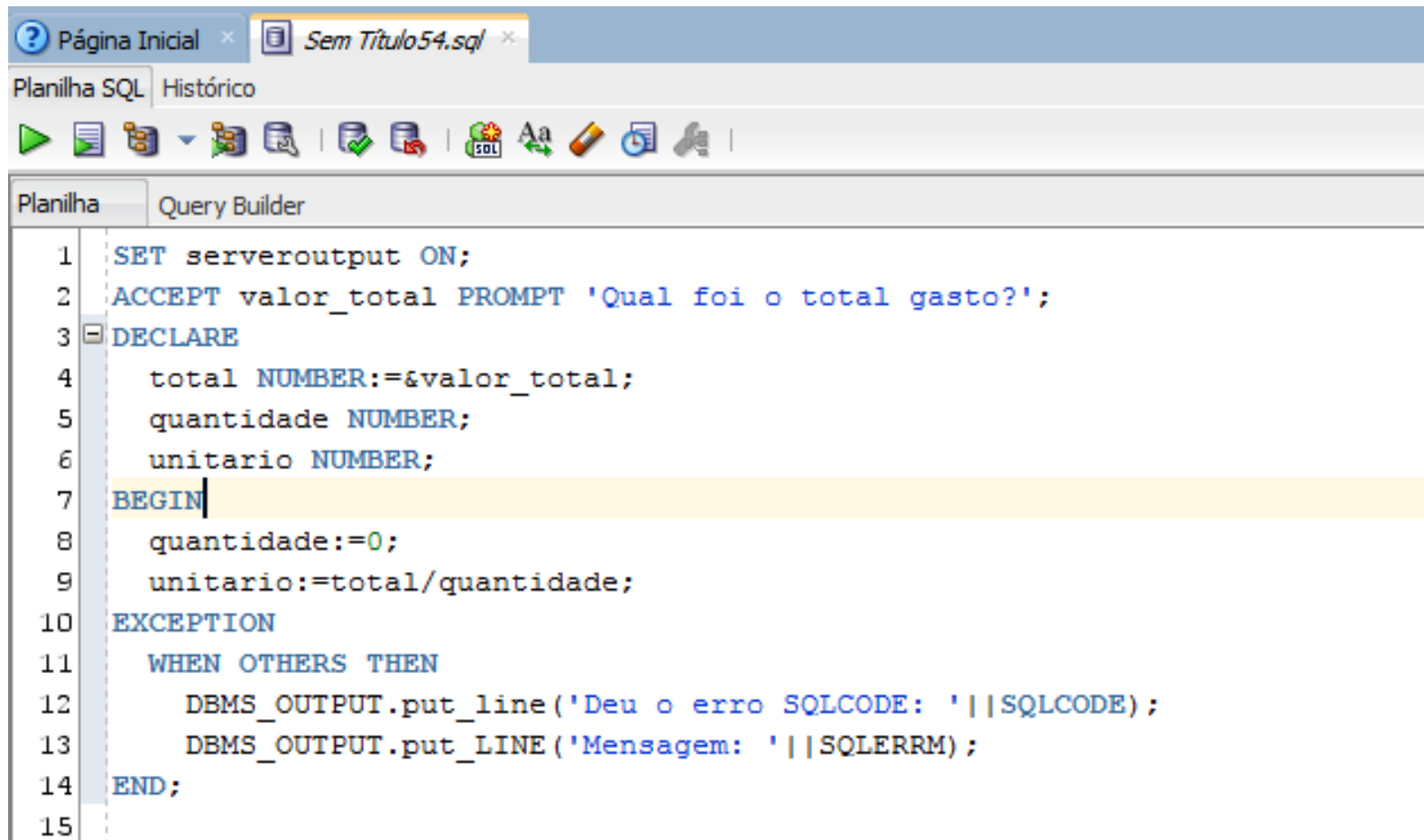
EXEMPLO



The screenshot shows a SQL IDE window with a tab titled "Sem Título54.sql". The interface includes a toolbar with icons for running queries, saving, and other database operations. The main editor area displays a PL/SQL script. The script starts with "SET serveroutput ON;" and "ACCEPT valor_total PROMPT 'Qual foi o total gasto?;". It then declares three variables: "total", "quantidade", and "unitario", all of type "NUMBER". The "BEGIN" block sets "quantidade" to 0 and "unitario" to "total/quantidade;". An "EXCEPTION" block handles two cases: "WHEN ZERO_DIVIDE THEN" which outputs "Divisão por zero não é permitida"; and "WHEN OTHERS THEN" which outputs "Deu outro erro;". The script ends with "END;". The IDE shows a execution time of 0,032 segundos.

```
1 SET serveroutput ON;
2 ACCEPT valor_total PROMPT 'Qual foi o total gasto;
3 DECLARE
4     total NUMBER:=&valor_total;
5     quantidade NUMBER;
6     unitario NUMBER;
7 BEGIN
8     quantidade:=0;
9     unitario:=total/quantidade;
10 EXCEPTION
11     WHEN ZERO_DIVIDE THEN
12         DBMS_OUTPUT.put_line('Divisão por zero não é permitida');
13     WHEN OTHERS THEN
14         DBMS_OUTPUT.put_line('Deu outro erro');
15 END;
16
```

EXEMPLO CAPTURANDO O CÓDIGO DE ERRO



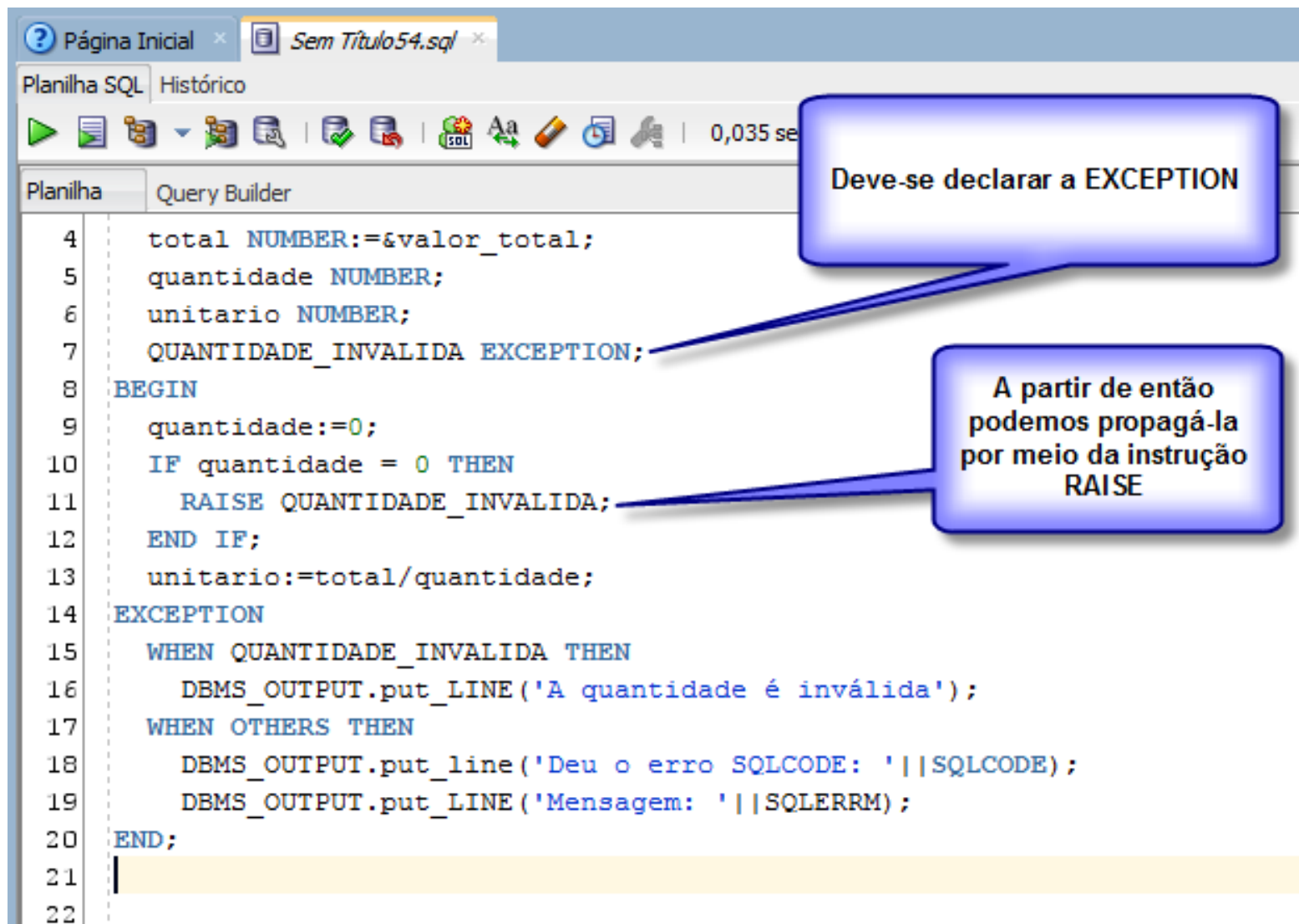
The screenshot shows a SQL IDE window with a tab titled "Sem Título54.sql". The interface includes a toolbar with various icons for execution and editing. The main editor area is titled "Planilha" and "Query Builder". It contains a PL/SQL script with the following code:

```
1 SET serveroutput ON;
2 ACCEPT valor_total PROMPT 'Qual foi o total gasto?';
3 DECLARE
4     total NUMBER:=&valor_total;
5     quantidade NUMBER;
6     unitario NUMBER;
7 BEGIN
8     quantidade:=0;
9     unitario:=total/quantidade;
10 EXCEPTION
11     WHEN OTHERS THEN
12         DBMS_OUTPUT.put_line('Deu o erro SQLCODE: '||SQLCODE);
13         DBMS_OUTPUT.put_LINE('Mensagem: '||SQLERRM);
14 END;
```

ERROS NOMEADOS PADRÃO

Exception	Oracle Error	SQLCODE Value
ACCESS_INTO_NULL	ORA-06530	-6530
CASE_NOT_FOUND	ORA-06592	-6592
COLLECTION_IS_NULL	ORA-06531	-6531
CURSOR_ALREADY_OPEN	ORA-06511	-6511
DUP_VAL_ON_INDEX	ORA-00001	-1
INVALID_CURSOR	ORA-01001	-1001
INVALID_NUMBER	ORA-01722	-1722
LOGIN_DENIED	ORA-01017	-1017
NO_DATA_FOUND	ORA-01403	+100
NOT_LOGGED_ON	ORA-01012	-1012
PROGRAM_ERROR	ORA-06501	-6501
ROWTYPE_MISMATCH	ORA-06504	-6504
SELF_IS_NULL	ORA-30625	-30625
STORAGE_ERROR	ORA-06500	-6500
SUBSCRIPT_BEYOND_COUNT	ORA-06533	-6533
SUBSCRIPT_OUTSIDE_LIMIT	ORA-06532	-6532
SYS_INVALID_ROWID	ORA-01410	-1410
TIMEOUT_ON_RESOURCE	ORA-00051	-51
TOO_MANY_ROWS	ORA-01422	-1422
VALUE_ERROR	ORA-06502	-6502
ZERO_DIVIDE	ORA-01476	-1476

DISPARANDO ERROS EXPLÍCITOS



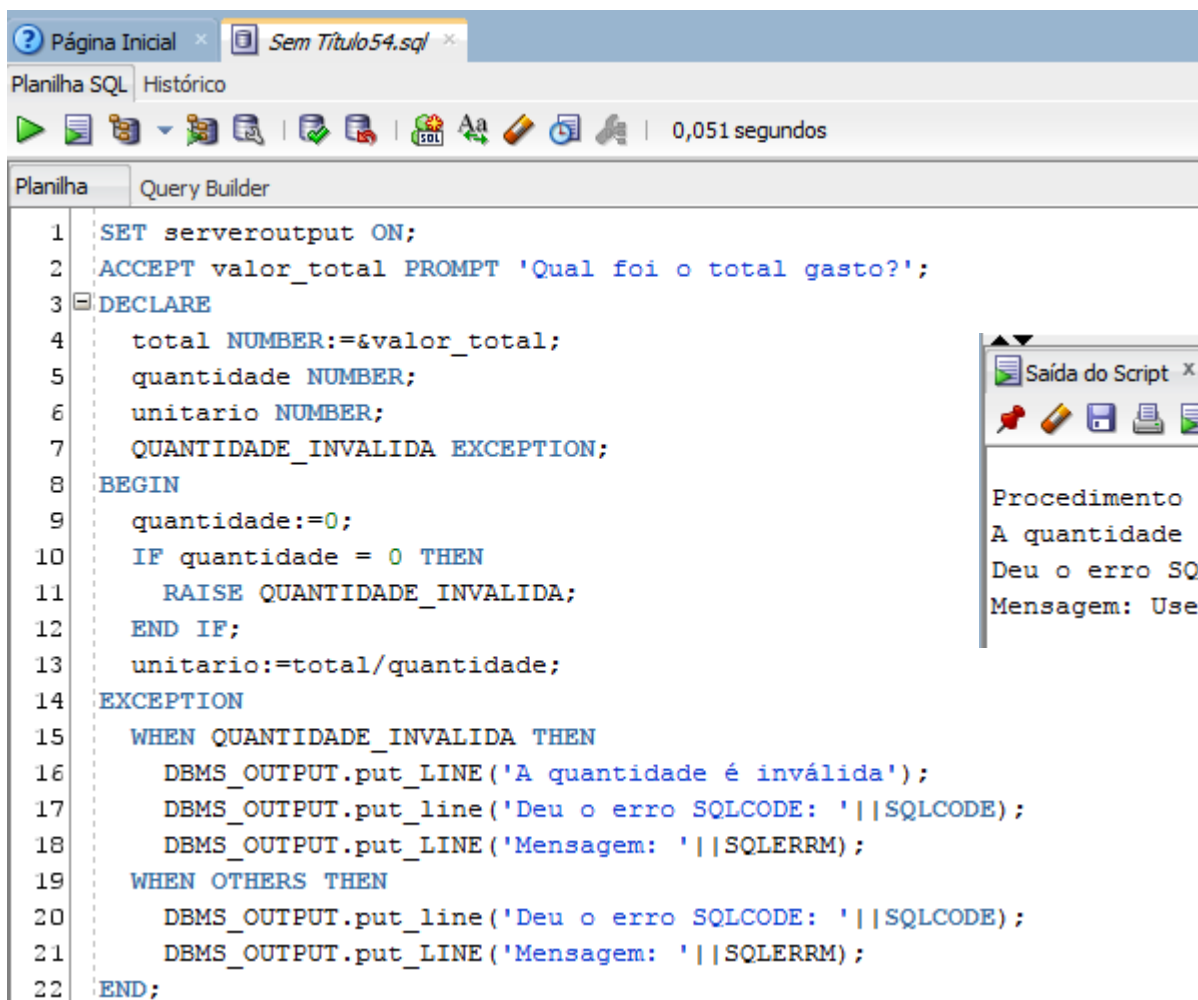
The screenshot shows a SQL IDE window titled 'Sem Título54.sql'. The interface includes a toolbar with icons for running, saving, and editing, and a status bar indicating '0,035 se'. The main editor displays a PL/SQL script with the following code:

```
4  total NUMBER:=&valor_total;
5  quantidade NUMBER;
6  unitario NUMBER;
7  QUANTIDADE_INVALIDA EXCEPTION;
8  BEGIN
9  quantidade:=0;
10 IF quantidade = 0 THEN
11     RAISE QUANTIDADE_INVALIDA;
12 END IF;
13 unitario:=total/quantidade;
14 EXCEPTION
15     WHEN QUANTIDADE_INVALIDA THEN
16         DBMS_OUTPUT.put_LINE('A quantidade é inválida');
17     WHEN OTHERS THEN
18         DBMS_OUTPUT.put_line('Deu o erro SQLCODE: '||SQLCODE);
19         DBMS_OUTPUT.put_LINE('Mensagem: '||SQLERRM);
20 END;
```

Two callout boxes provide additional information:

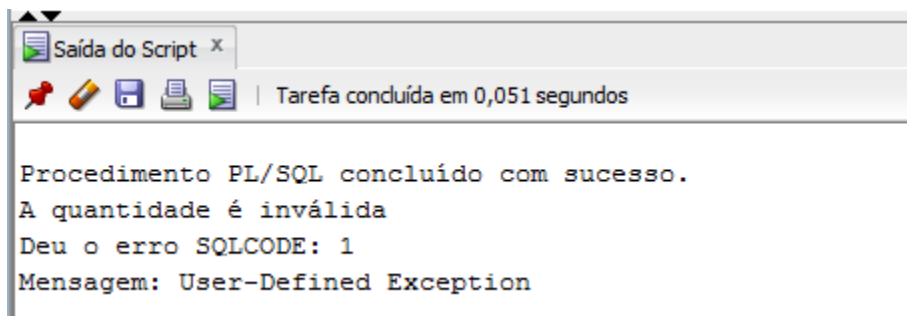
- Deve-se declarar a EXCEPTION**: Points to line 7, where the exception is declared.
- A partir de então podemos propagá-la por meio da instrução RAISE**: Points to line 11, where the exception is raised.

CÓDIGOS DE ERRO VS EXCEÇÕES DE USUÁRIO



The screenshot shows a SQL IDE window titled 'Sem Título54.sql'. The interface includes a toolbar with icons for running, saving, and editing, and a status bar indicating '0,051 segundos'. The script is written in PL/SQL and demonstrates a user-defined exception. It starts with 'SET serveroutput ON;' and 'ACCEPT valor_total PROMPT 'Qual foi o total gasto?';'. The 'DECLARE' section defines a variable 'total' of type 'NUMBER', a variable 'quantidade' of type 'NUMBER', a variable 'unitario' of type 'NUMBER', and a user-defined exception 'QUANTIDADE_INVALIDA EXCEPTION;'. The 'BEGIN' block sets 'quantidade:=0;', and includes an 'IF' statement: 'IF quantidade = 0 THEN RAISE QUANTIDADE_INVALIDA; END IF;'. It then calculates 'unitario:=total/quantidade;'. The 'EXCEPTION' section has two handlers: 'WHEN QUANTIDADE_INVALIDA THEN' which outputs 'A quantidade é inválida', the error code 'Deu o erro SQLCODE: '||SQLCODE'', and the message 'Deu o erro SQLERRM: '||SQLERRM''; and 'WHEN OTHERS THEN' which outputs the error code and message. The script ends with 'END;'.

```
1 SET serveroutput ON;
2 ACCEPT valor_total PROMPT 'Qual foi o total gasto?';
3 DECLARE
4     total NUMBER:=&valor_total;
5     quantidade NUMBER;
6     unitario NUMBER;
7     QUANTIDADE_INVALIDA EXCEPTION;
8 BEGIN
9     quantidade:=0;
10    IF quantidade = 0 THEN
11        RAISE QUANTIDADE_INVALIDA;
12    END IF;
13    unitario:=total/quantidade;
14 EXCEPTION
15    WHEN QUANTIDADE_INVALIDA THEN
16        DBMS_OUTPUT.put_LINE('A quantidade é inválida');
17        DBMS_OUTPUT.put_line('Deu o erro SQLCODE: '||SQLCODE);
18        DBMS_OUTPUT.put_LINE('Mensagem: '||SQLERRM);
19    WHEN OTHERS THEN
20        DBMS_OUTPUT.put_line('Deu o erro SQLCODE: '||SQLCODE);
21        DBMS_OUTPUT.put_LINE('Mensagem: '||SQLERRM);
22 END;
```



The screenshot shows the 'Saída do Script' window, which displays the output of the PL/SQL script. The status bar indicates 'Tarefa concluída em 0,051 segundos'. The output text is: 'Procedimento PL/SQL concluído com sucesso. A quantidade é inválida Deu o erro SQLCODE: 1 Mensagem: User-Defined Exception'.

```
Saída do Script x
Tarefa concluída em 0,051 segundos

Procedimento PL/SQL concluído com sucesso.
A quantidade é inválida
Deu o erro SQLCODE: 1
Mensagem: User-Defined Exception
```

CRIANDO NOVOS CÓDIGOS DE ERRO

The screenshot displays the Oracle SQL Developer environment. The main window shows a script named 'Sem Título54.sql' with the following PL/SQL code:

```

1 SET serveroutput ON;
2 ACCEPT valor_total PROMPT 'Qual foi o total gasto?';
3 DECLARE
4     total NUMBER:=&valor_total;
5     quantidade NUMBER;
6     unitario NUMBER;
7     QUANTIDADE_INVALIDA EXCEPTION;
8 BEGIN
9     quantidade:=0;
10    IF quantidade = 0 THEN
11        RAISE_APPLICATION_ERROR(-20101, 'Não é possível dividir por zero');
12    END IF;
13    unitario:=total/quantidade;
14 EXCEPTION
15    WHEN OTHERS THEN
16        DBMS_OUTPUT.put_line('Deu o erro SQLCODE: '||SQLCODE);
17        DBMS_OUTPUT.put_line('Mensagem: '||SQLERRM);
18 END;
19

```

The 'Saída do Script' (Script Output) window shows the execution results:

```

Procedimento PL/SQL concluído com sucesso.

Deu o erro SQLCODE: -20101
Mensagem: ORA-20101: Não é possível dividir por zero

```


EXERCÍCIOS

EXERCÍCIOS

- Elabore um algoritmo que leia dois números e imprima qual é maior, qual é menor; Se eles forem iguais lance uma exceção e termine o programa;
- Elabore um algoritmo que leia as 5 primeiras letras do alfabeto (ou A, ou B, ou C, ou D) e o e imprima um nome de fruta que comece com essa letra. Caso seja digitado um uma outra letra, deverá ser lançada uma exceção;
- Para vários tributos, a base de cálculo é o salário mínimo. Elabore um algoritmo que leia o valor do salário mínimo e o valor do salário de uma pessoa. Calcular e imprimir quantos salários mínimos essa pessoa ganha; Lance tratamentos de exceção condizentes;

5. VARRAY

VETORES

- Um vetor ou array é uma coleção de variáveis do mesmo tipo que compartilham um mesmo nome, onde cada valor individual é referenciado por um índice.

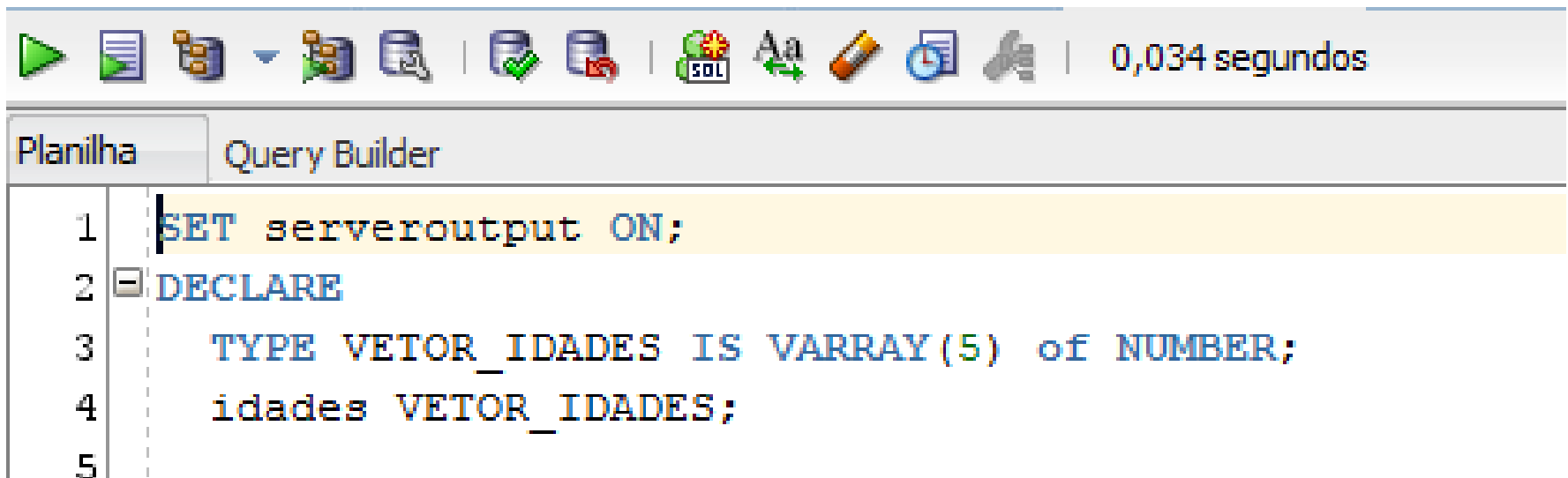
idades

21	1	18	68	43
1	2	3	4	5

O vetor idade tem tamanho 5
Cada elemento é acessado por seu índice. Por exemplo
vetor[2] vale 1 enquanto vetor[5] vale 43

DECLARANDO UM VETOR EM PL/SQL

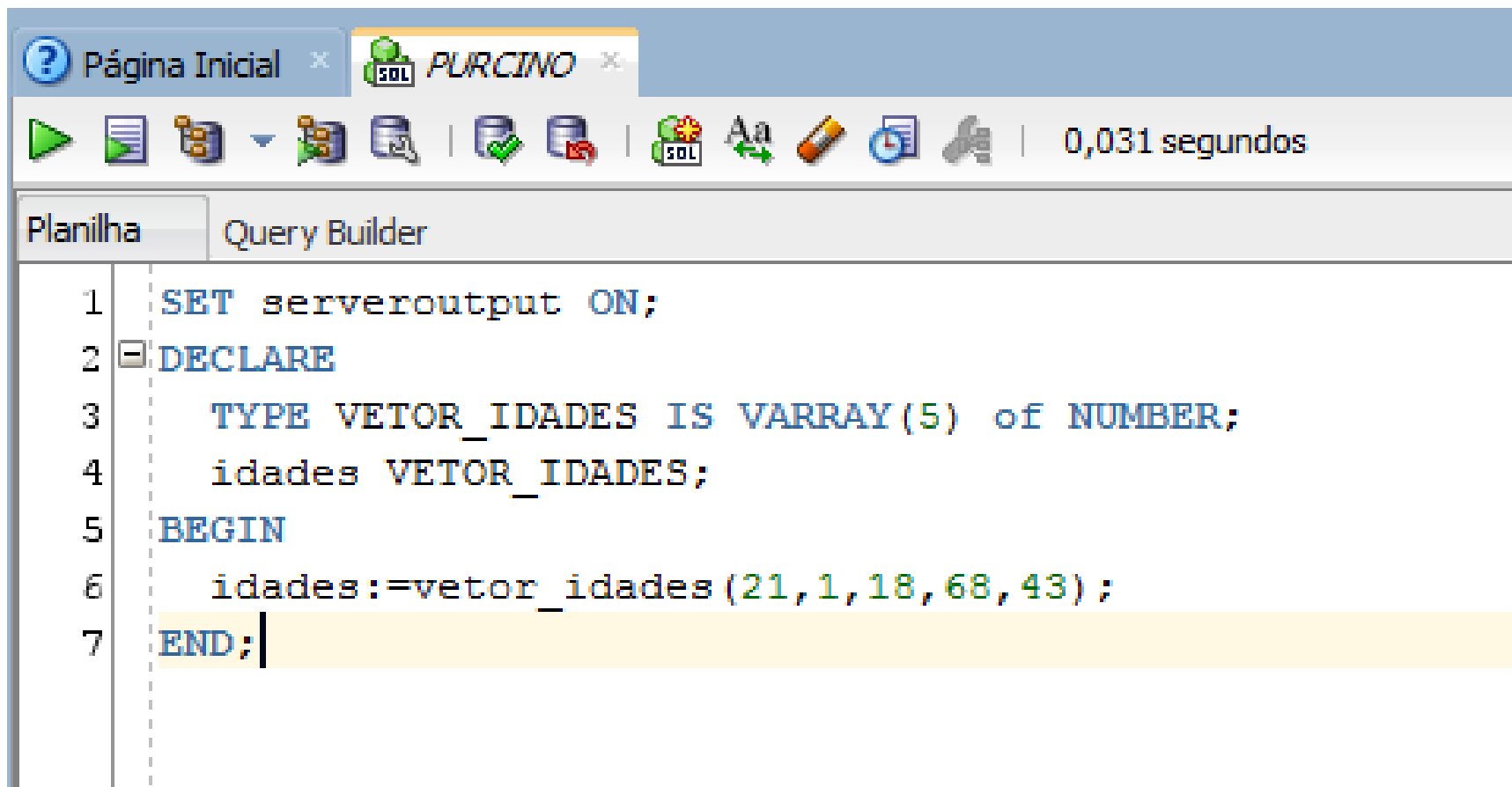
```
TYPE NOME_TIPO IS VARRAY(tamanho) of TIPO;  
nome_vetor NOME_TIPO;
```



The screenshot shows a SQL IDE interface. At the top, there is a toolbar with various icons for execution, saving, and editing. Below the toolbar, there are two tabs: 'Planilha' and 'Query Builder'. The 'Query Builder' tab is active, displaying a list of SQL statements. The first statement is 'SET serveroutput ON;'. The second statement is 'DECLARE', followed by an indented line 'TYPE VETOR_IDADES IS VARRAY(5) of NUMBER;', and another indented line 'idades VETOR_IDADES;'. The line numbers 1 through 5 are visible on the left side of the code editor.

```
1 SET serveroutput ON;  
2 DECLARE  
3     TYPE VETOR_IDADES IS VARRAY(5) of NUMBER;  
4     idades VETOR_IDADES;  
5
```

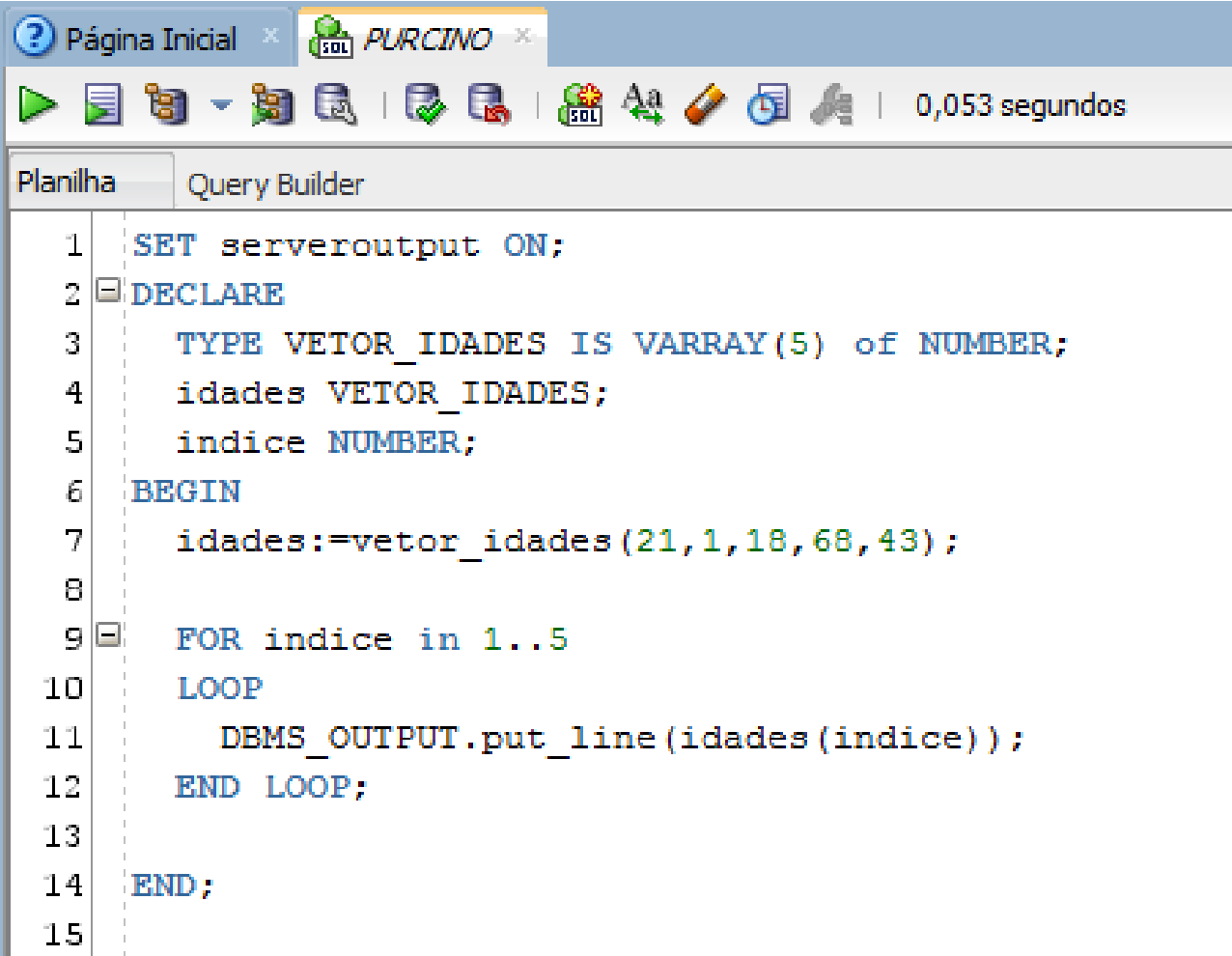
PREENCHENDO OS DADOS DE UM VETOR EM PL/SQL



The screenshot shows a web-based PL/SQL IDE interface. At the top, there are two tabs: 'Página Inicial' and 'PURCINO'. Below the tabs is a toolbar with various icons for execution, formatting, and editing. A status bar on the right indicates a execution time of '0,031 segundos'. The main area is a code editor with a line number margin on the left. The code being entered is a PL/SQL block that declares a vector and populates it with specific values.

```
1 SET serveroutput ON;
2 DECLARE
3     TYPE VETOR_IDADES IS VARRAY(5) of NUMBER;
4     idades VETOR_IDADES;
5 BEGIN
6     idades:=vetor_idades (21,1,18,68,43);
7 END;
```

ACESSANDO UM VETOR EM PL/SQL



The screenshot shows the PL/SQL Query Builder interface. The top bar includes a toolbar with icons for execution, saving, and other functions, along with a timer showing 0,053 segundos. The main area displays a PL/SQL script with line numbers 1 through 15. The script declares a vector and iterates through its elements.

```
1 SET serveroutput ON;
2 DECLARE
3     TYPE VETOR_IDADES IS VARRAY(5) of NUMBER;
4     idades VETOR_IDADES;
5     indice NUMBER;
6 BEGIN
7     idades:=vetor_idades(21,1,18,68,43);
8
9     FOR indice in 1..5
10     LOOP
11         DBMS_OUTPUT.put_line(idades(indice));
12     END LOOP;
13
14 END;
```

EXERCÍCIOS

EXERCÍCIOS

- Declare um vetor com as temperaturas de 7 pessoas diferentes e escreva na tela a temperatura de todas as pessoas junto com um diagnóstico se ela está com febre ou não.

REFERÊNCIAS



- OLIVEIRA, Jayr Figueiredo de; MANZANO, José Augusto N. G. **Algoritmos: Lógica para Desenvolvimento de Programação de Computadores**. 23ª Edição. São Paulo: Érica, 2010.
- MIZRAHI, Victorine Viviane. **Treinamento em Linguagem C**. 2ª Edição. São Paulo: Pearson, 2008.
- ELMASRI, R.; NAVATHE, S.B. **Sistemas de Banco de Dados: Fundamentos e Aplicações**. Pearson, 2005.
- SILBERSCHATZ, A; KORTH, H. F.; SUDARSHAN, S. **Sistema de Banco de Dados**. Campus, 2006.

Copyright © 2023 Profa. Patrícia Angelini

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).