

# Learn PL/SQL from Mistakes

---

---

This book is about some of the PL/SQL errors. Also, give you an opportunity to learn PL/SQL from mistakes. This book employs examples and you will see a mistake in the program and the possible solution for the mistake. This book is suitable for self-learners.

---

---

## Table of Contents

<b>Oracle Error Message</b>	<b>Page No.</b>
<b>ORA-06502: PL/SQL: numeric or value error</b> .....	<b>3</b>
<b>ORA-06502: PL/SQL: numeric or value error: character string buffer too small</b> .....	<b>4</b>
<b>ORA-06502: PL/SQL: numeric or value error: number precision too large</b> .....	<b>5</b>
<b>ORA-06531: Reference to uninitialized collection</b> .....	<b>8</b>
<b>ORA-06533: Subscript beyond count</b> .....	<b>9</b>
<b>PLS-00103: Encountered the symbol ""</b> .....	<b>10</b>
<b>PLS-00103: Encountered the symbol ";"</b> .....	<b>11</b>
<b>PLS-00103: Encountered the symbol " "</b> .....	<b>19</b>
<b>PLS-00103: Encountered the symbol "="</b> .....	<b>20</b>
<b>PLS-00103: Encountered the symbol "DBMS_OUTPUT"</b> .....	<b>21</b>
<b>PLS-00103: Encountered the symbol "DECLARE"</b> .....	<b>30</b>
<b>PLS-00103: Encountered the symbol "ELSIF"</b> .....	<b>31</b>
<b>PLS-00103: Encountered the symbol "END"</b> .....	<b>32</b>
<b>PLS-00103: Encountered the symbol "end-of-file"</b> .....	<b>39</b>
<b>PLS-00103: Encountered the symbol "IF"</b> .....	<b>42</b>
<b>PLS-00103: Encountered the symbol "IS"</b> .....	<b>46</b>
<b>PLS-00103: Encountered the symbol "LOOP"</b> .....	<b>51</b>
<b>PLS-00103: Encountered the symbol "THEM"</b> .....	<b>54</b>
<b>PLS-00103: Encountered the symbol "xxxxx"</b> .....	<b>55</b>

# Learn PL/SQL from Mistakes

---

---

## Table of Contents

<b>Oracle Error Message</b>	<b>Page No.</b>
PLS-00103: PLS-00201: identifier 'xxxxx' must be declared.....	79
PLS-00113: END identifier 'xxxxx'.....	80
PLS-00201: identifier 'DBMS_OUTPUTPUT_LINE' must be declared.....	81
PLS-00201: identifier 'xxxxx' must be declared.....	82
PLS-00215: String length constraints must be in range (1 .. 32767).....	88
PLS-00216: NUMBER precision constraint must be in range (1 .. 38).....	91
PLS-00302: component 'EXTEND' must be declared.....	92
PLS-00306: wrong number or types of arguments in call to 'PUT_LINE'.....	93
PLS-00325: non-integral numeric literal xxxxx is inappropriate in this context.....	94
PLS-00330: invalid use of type name or subtype name.....	95
PLS-00355: use of pl/sql table not allowed in this context.....	98
PLS-00363: expression '1' cannot be used as an assignment target.....	99
PLS-00363: expression 'xxxxx' cannot be used as an assignment target.....	100
PLS-00382: expression is of wrong type.....	104
PLS-00573: cannot constrain scale, precision, or range of an anchored type declaration.....	107
PLS-00615: type mismatch found at 'xxxxx' between CASE operand and WHEN operands.....	108
PL/SQL Programs with Mistakes (Find the Mistakes).....	109
Solutions for PL/SQL Program Mistakes.....	127

## Learn PL/SQL from Mistakes

### Oracle Error Message: ORA-06502: PL/SQL: numeric or value error

<b>Aim of the PL/SQL Program</b>	Example program for TYPE and declared FUNCTION usage
<b>PL/SQL Code with Mistakes</b>	<pre> 1 DECLARE 2     TYPE sno_t IS TABLE OF NUMBER(1); 3     nums_v    sno_t := sno_t(1,2,3,4,5,6,7,8,9,10); 4     FUNCTION avg_fun (nums_p IN sno_t) RETURN NUMBER IS 5               i          PLS_INTEGER:=nums_p.FIRST; 6               avg_v      NUMBER:=0; 7 BEGIN 8     WHILE i IS NOT NULL LOOP 9               avg_v := avg_v + nums_p(i) / nums_p.COUNT; 10              i := nums_p.NEXT(i); 11        END LOOP; 12        RETURN avg_v; 13 END avg_fun; 14 BEGIN 15     DBMS_OUTPUT.PUT_LINE(avg_fun(nums_v)); 16 END; </pre>
<b>Oracle Error Message</b>	ORA-06502: PL/SQL: numeric or value error: number precision too large ORA-06512: at line 3 ORA-06512: at "SYS.DBMS_SQL", line 1721
<b>Mistake Line</b>	Line 2
<b>Problem Details</b>	Datatype NUMBER size is small i.e., size "1" is not sufficient.
<b>Possible Solution</b>	By increasing datatype NUMBER size from 1 to 2 or more.

## Learn PL/SQL from Mistakes

**Oracle Error Message: ORA-06502: PL/SQL: numeric or value error: character string buffer too small**

<b>Aim of the PL/SQL Program</b>	Example program for CASE statement.
<b>PL/SQL Code with Mistakes</b>	<pre> 1  DECLARE 2    code CHAR := 'CC'; 3  BEGIN 4    CASE code 5      WHEN 'H' THEN DBMS_OUTPUT.PUT_LINE('Hyderabad'); 6      WHEN 'B' THEN DBMS_OUTPUT.PUT_LINE('Bangaluru'); 7      WHEN 'C' THEN DBMS_OUTPUT.PUT_LINE('Chennai'); 8      WHEN 'D' THEN DBMS_OUTPUT.PUT_LINE('Delhi'); 9      WHEN 'P' THEN DBMS_OUTPUT.PUT_LINE('Pune'); 10     ELSE DBMS_OUTPUT.PUT_LINE('Invalid Code'); 11   END CASE; 12 END; 13 / </pre>
<b>Oracle Error Message</b>	ORA-06502: PL/SQL: numeric or value error: character string buffer too small ORA-06512: at line 2 ORA-06512: at "SYS.DBMS_SQL", line 1721
<b>Mistake Line</b>	Line 2
<b>Problem Details</b>	Here, datatype CHAR width is "1". Hence, "CC" cannot assign.
<b>Possible Solution</b>	By specifying width more than "1" or assign "C" to variable "code".

## Learn PL/SQL from Mistakes

---

**Oracle Error Message: ORA-06502: PL/SQL: numeric or value error: number precision too large**

(from Page #6 to Page #7)

## Learn PL/SQL from Mistakes

**Oracle Error Message: ORA-06502: PL/SQL: numeric or value error: number precision too large**

<b>Aim of the PL/SQL Program</b>	To display numbers from 1 to 100
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     i    NUMBER(2) := 1; 3 BEGIN 4     WHILE i &lt;= 100 LOOP 5         DBMS_OUTPUT.PUT_LINE(i); 6         i:=i+1;    (99+1 = 100 result cannot store when i=99) 7     END LOOP; 8 END; 9 /</pre>
<b>Oracle Error Message</b>	ORA-06502: PL/SQL: numeric or value error: number precision too large ORA-06512: at line 6
<b>Mistake Line</b>	Line 6
<b>Problem Details</b>	This error message is from runtime. Size of variable “i” in declare section is 2 and this size is not sufficient to store 100 value into the variable “i”.
<b>Possible Solution</b>	Increase the variable “i” size from 2 to 3 or more.


## Learn PL/SQL from Mistakes

**Oracle Error Message: ORA-06502: PL/SQL: numeric or value error: number precision too large**

<b>Aim of the PL/SQL Program</b>	To display 1 to 10 numbers along with string "PL/SQL".
<b>PL/SQL Code with Mistakes</b>	<pre> 1 DECLARE 2     n NUMBER(1,1):=1; 3 BEGIN 4     LOOP 5     IF n &gt; 10 THEN 6         EXIT; 7     END IF; 8         DBMS_OUTPUT.PUT_LINE(n  '. PL/SQL'); 9     n:=n+1; 10    END LOOP; 11    DBMS_OUTPUT.PUT_LINE('End of Program'); 12 END; 13 / </pre>
<b>Oracle Error Message</b>	ORA-06502: PL/SQL: numeric or value error: number precision too large ORA-06512: at line 2 ORA-06512: at "SYS.DBMS_SQL", line 1721
<b>Mistake Line</b>	Line 2
<b>Problem Details</b>	Datatype NUMBER width is wrong.
<b>Possible Solution</b>	By giving valid width for the datatype NUMBER.

## Learn PL/SQL from Mistakes


### Oracle Error Message: ORA-06531: Reference to uninitialized collection

<b>Aim of the PL/SQL Program</b>	Example program for TYPE usage.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     TYPE tp IS TABLE OF VARCHAR2(20) NOT NULL; 3     tp_v tp; 4 BEGIN 5     tp_v.EXTEND; 6     tp_v(1) := 'Sailaja'; 7 END; 8 /</pre> 
<b>Oracle Error Message</b>	ORA-06531: Reference to uninitialized collection ORA-06512: at line 5 ORA-06512: at "SYS.DBMS_SQL", line 1721
<b>Mistake Line</b>	Line 3
<b>Problem Details</b>	Initialization of type variable is missing.
<b>Possible Solution</b>	By initializing the type variable i.e., tp_v tp:=tp();



## Learn PL/SQL from Mistakes

### Oracle Error Message: ORA-06533: Subscript beyond count

<b>Aim of the PL/SQL Program</b>	Example program for TYPE usage.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     str VARCHAR2(20); 3     TYPE tp IS TABLE OF VARCHAR2(20) NOT NULL; 4     tp_v tp:=tp(); 5 BEGIN 6     tp_v.EXTEND; 7     tp_v(2) := 'Sailaja'; 8 END; 9 /</pre> 
<b>Oracle Error Message</b>	ORA-06533: Subscript beyond count ORA-06512: at line 7 ORA-06512: at "SYS.DBMS_SQL", line 1721
<b>Mistake Line</b>	Line 7
<b>Problem Details</b>	Wrong index specification in type variable.
<b>Possible Solution</b>	By replacing index "2" with "1" i.e., tp_v(1)

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol ""

<b>Aim of the PL/SQL Program</b>	Example program for TYPE and declared FUNCTION usage
<b>PL/SQL Code with Mistakes</b>	<pre> 1 DECLARE 2     TYPE sno_t IS TABLE OF NUMBER; 3     nums_v   sno_t := sno_t(1,2,3,4,5,6,7,8,9,10); 4     FUNCTION avg_fun (nums_p IN sno_t) RETURN NUMBER IS 5               i           PLS_INTEGER:=nums_p.FIRST; 6               avg_v      NUMBER:=0; 7 BEGIN 8     WHILE i IS NOT NULL LOOP 9               avg_v : avg_v + nums_p(i) / nums_p.COUNT; 10              i := nums_p.NEXT(i); 11        END LOOP; 12        RETURN avg_v; 13 END avg_fun; 14 BEGIN 15     DBMS_OUTPUT.PUT_LINE(avg_fun(nums_v)); 16 END; </pre>
<b>Oracle Error Message</b>	<p>ORA-06550: line 9, column 16:          PLS-00103: Encountered the symbol "" when expecting one of the following:          := . ( @ % ;          The symbol "!=" was substituted for "" to continue.</p>
<b>Mistake Line</b>	Line 9
<b>Problem Details</b>	This error message is from the parser. “=” (equal symbol) is missing in assignment operator i.e., “:=” (colon equal)
<b>Possible Solution</b>	By giving “=” equal symbol after colon (: ) symbol.

## Learn PL/SQL from Mistakes


---

**Oracle Error Message: PLS-00103: Encountered the symbol ";"**

(from Page #12 to Page #18)


## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol ";"

<b>Aim of the PL/SQL Program</b>	To display numbers from 1 to 100
<b>PL/SQL Code with Mistakes</b>	<pre style="background-color: #f0f0f0; padding: 10px;"> 1 DECLARE 2   i   NUMBER(3):=1; 3 BEGIN 4   WHILE i &lt;= 100 LOOP; 5       DBMS_OUTPUT.PUT_LINE(i); 6       i:=i+1; 7   END LOOP; 8 END;</pre> 
<b>Oracle Error Message</b>	<p>ORA-06550: line 4, column 23:          PLS-00103: Encountered the symbol ";" when expecting one of the following:          ( begin case declare exit for goto if loop mod null pragma raise return select update while          with &lt;an identifier&gt; &lt;a double-quoted delimited-identifier&gt; &lt;a bind variable&gt; &lt;&lt; continue          close current delete fetch lock insert open rollback          savepoint set sql execute commit forall merge pipe purge json_exists json_value          json_query json_object json_array          The symbol "exit" was substituted for "</p>
<b>Mistake Line</b>	Line 4
<b>Problem Details</b>	This error message is from the parser. Semicolon (;) at end of "LOOP" keyword.
<b>Possible Solution</b>	By removing semicolon (;) after "LOOP" keyword.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol ";"

<b>Aim of the PL/SQL Program</b>	To display numbers from 1 to 100
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     i    CHAR(3):=1; 3 BEGIN 4     WHILE i &lt;= 100 LOOP 5         DBMS_OUTPUT.PUT_LINE(i); 6         i:=i+1; 7     ENDLOOP; 8 END; 9 /</pre> 
<b>Oracle Error Message</b>	ORA-06550: line 8, column 4: PLS-00103: Encountered the symbol ";" when expecting one of the following: loop
<b>Mistake Line</b>	Line 7
<b>Problem Details</b>	This error message is from the parser. Space is missing in between "END" and "LOOP" keywords.
<b>Possible Solution</b>	By including at least one space between "END" and "LOOP" keywords.


## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol ";"

<b>Aim of the PL/SQL Program</b>	To display biggest of two numbers.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     n1     INT:=24; 3     n2     INT:=55; 4 BEGIN 5     IF n1 &gt; n2 THEN 6         DBMS_OUTPUT.PUT_LINE(n1    ' is BIG'); 7     ELSE 8         IF n2 &gt; n1 THEN 9             DBMS_OUTPUT.PUT_LINE(n2    ' is BIG'); 10        ELSE 11            DBMS_OUTPUT.PUT_LINE(n2    ' Both are Equal'); 12        END IF; 13 END; 14 /</pre>
<b>Oracle Error Message</b>	ORA-06550: line 13, column 4: PLS-00103: Encountered the symbol ";" when expecting one of the following: if
<b>Mistake Line</b>	Related to Line 8
<b>Problem Details</b>	This error message is from the parser. Only one "END IF" for two "IF".
<b>Possible Solution</b>	Either by replacing "ELSE" and "IF" with "ELSIF" or by adding additionally "END IF"

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol ";"

<b>Aim of the PL/SQL Program</b>	To return square of a given number.
<b>PL/SQL Code with Mistakes</b>	<pre style="background-color: #f9f9f9; padding: 10px;"> 1 CREATE or replace FUNCTION f_square (n NUMBER) return number is; 2     p NUMBER; 3 BEGIN 4     p:=n * n; 5     return p; 6 END; 7 /</pre> 
<b>Oracle Error Message</b>	<p>Errors: FUNCTION F_SQUARE  Line/Col: 1/47 PLS-00103: Encountered the symbol ";" when expecting one of the following:  begin function pragma procedure subtype type &lt;an identifier&gt;  &lt;a double-quoted delimited-identifier&gt; current cursor delete  exists prior external language</p>
<b>Mistake Line</b>	Line 1
<b>Problem Details</b>	This error message is from the parser. Semicolon (;) after "IS" keyword.
<b>Possible Solution</b>	By remove semicolon (;) after "IS" keyword.

## Learn PL/SQL from Mistakes


### Oracle Error Message: PLS-00103: Encountered the symbol ";"

<b>Aim of the PL/SQL Program</b>	Example program for TYPE and declared FUNCTION usage
<b>PL/SQL Code with Mistakes</b>	<pre> 1 DECLARE 2     TYPE sno_t IS TABLE OF NUMBER; 3     nums_v    sno_t := sno_t(1,2,3,4,5,6,7,8,9,10); 4     FUNCTION avg_fun (nums_p IN sno_t) RETURN NUMBER IS 5               i          PLS_INTEGER:=nums_p.FIRST; 6               avg_v      NUMBER:=0; 7 BEGIN 8     WHILE i IS NOT NULL LOOP; ← 9               avg_v := avg_v + nums_p(i) / nums_p.COUNT; 10              i := nums_p.NEXT(i); 11        END LOOP; 12        RETURN avg_v; 13 END avg_fun; 14 BEGIN 15     DBMS_OUTPUT.PUT_LINE(avg_fun(nums_v)); 16 END; </pre>
<b>Oracle Error Message</b>	<p>ORA-06550: line 8, column 29:          PLS-00103: Encountered the symbol ";" when expecting one of the following:          ( begin case declare exit for goto if loop mod null pragma raise return select update while          with &lt;an identifier&gt; ..... json_query json_object json_array          The symbol "exit" was substituted for "</p>
<b>Mistake Line</b>	Line 8
<b>Problem Details</b>	Semicolon (;) given after "LOOP" keyword.
<b>Possible Solution</b>	By removing semicolon (;) after the keyword "LOOP"




## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol ";"

<b>Aim of the PL/SQL Program</b>	Example program for CASE statement.
<b>PL/SQL Code with Mistakes</b>	<pre> 1  DECLARE 2    code CHAR:='C'; 3  BEGIN 4    CASE code 5      WHEN 'H' THEN DBMS_OUTPUT.PUT_LINE('Hyderabad'); 6      WHEN 'B' THEN DBMS_OUTPUT.PUT_LINE('Bangaluru'); 7      WHEN 'C' THEN DBMS_OUTPUT.PUT_LINE('Chennai'); 8      WHEN 'D' THEN DBMS_OUTPUT.PUT_LINE('Delhi'); 9      WHEN 'P' THEN DBMS_OUTPUT.PUT_LINE('Pune'); 10     ELSE DBMS_OUTPUT.PUT_LINE('Invalid Code'); 11   END CASE; 12 END; 13 / </pre> 
<b>Oracle Error Message</b>	<p>ORA-06550: line 6, column 53:          PLS-00103: Encountered the symbol ";" when expecting one of the following:          ), * &amp; = - + &lt; / &gt; at in is mod remainder not rem =&gt;          &lt;an exponent (**)&gt; &lt;&gt; or != or ~= &gt;= &lt;= &lt;&gt; and or default          like like2 like4 likec as between from using    multiset          member submultiset          The symbol ")" was substituted for ";" to continue.</p>
<b>Mistake Line</b>	Line 6
<b>Problem Details</b>	This error message is from the parser. Closing bracket is missing.
<b>Possible Solution</b>	By giving closing bracket for the statement line 6.


## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol ";"

<b>Aim of the PL/SQL Program</b>	Example program for CASE statement.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     code CHAR:='C'; 3 BEGIN 4     CASE code 5         WHEN 'H' THEN DBMS_OUTPUT.PUT_LINE('Hyderabad'); 6         WHEN 'B' THEN DBMS_OUTPUT.PUT_LINE('Bangaluru'); 7         WHEN 'C' THEN DBMS_OUTPUT.PUT_LINE('Chennai'); 8         WHEN 'D' THEN DBMS_OUTPUT.PUT_LINE('Delhi'); 9         WHEN 'P' THEN DBMS_OUTPUT.PUT_LINE('Pune'); 10        ELSE DBMS_OUTPUT.PUT_LINE('Invalid Code'); 11    ENDCASE; 12 END; 13 /</pre> 
<b>Oracle Error Message</b>	ORA-06550: line 12, column 4: PLS-00103: Encountered the symbol ";" when expecting one of the following: case
<b>Mistake Line</b>	Line 11
<b>Problem Details</b>	This error message is from the parser. Space is missing in between END and CASE.
<b>Possible Solution</b>	By adding at least one space between END and CASE.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "|"

<b>Aim of the PL/SQL Program</b>	To swap two numbers and display two number after swapping.
<b>PL/SQL Code with Mistakes</b>	<pre> 1 DECLARE 2     v1  INT:=1; 3     v2  INT:=2; 4     PROCEDURE swap(p1 IN OUT INT, p2 IN OUT INT) 5     IS 6         l  INT; 7     BEGIN 8         l:=p1; 9         p1:=p2; 10        p2:=l; 11    END swap; 12 BEGIN 13     swap(v1, v2); 14     DBMS_OUTPUT.PUT_LINE('v1 value = ' v1); 15     DBMS_OUTPUT.PUT_LINE('v2 value = '  v2); 16 END;</pre> <div style="text-align: right; margin-right: 50px;">  </div>
<b>Oracle Error Message</b>	ORA-06550: line 14, column 38: PLS-00103: Encountered the symbol " " when expecting one of the following: ) , * & = - + < / > at in.....between from using    multiset member submultiset The symbol " ", was inserted before " " to continue.
<b>Mistake Line</b>	Line 14
<b>Problem Details</b>	This error message is from the parser. One pipe ( ) symbol is missing in concatenation operator i.e., two pipe (  ) symbols.
<b>Possible Solution</b>	By adding pipe ( ) symbol.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "="

<b>Aim of the PL/SQL Program</b>	To display numbers from 1 to 100
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     i    NUMBER(3):=1; 3 BEGIN 4     WHILE i &lt;= 100 LOOP 5         DBMS_OUTPUT.PUT_LINE(i); 6         =&gt; i=i+1; 7     END LOOP; 8 END; 9 /</pre>
<b>Oracle Error Message</b>	ORA-06550: line 6, column 8: PLS-00103: Encountered the symbol "=" when expecting one of the following: := . ( @ % ; The symbol ":= was inserted before "=" to continue.
<b>Mistake Line</b>	Line 6
<b>Problem Details</b>	This error message is from the parser. Colon (:) is missing in expression.
<b>Possible Solution</b>	By giving colon (:) in expression as ":= " (colon equal) is the assignment operator in PL/SQL.

## Learn PL/SQL from Mistakes

---

**Oracle Error Message: PLS-00103: Encountered the symbol "DBMS\_OUTPUT"**

(from Page #22 to Page #29)

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "DBMS\_OUTPUT"

<b>Aim of the PL/SQL Program</b>	To display numbers from 1 to 100
<b>PL/SQL Code with Mistakes</b>	<pre style="background-color: #f0f0f0; padding: 10px;"> 1 DECLARE 2     i    NUMBER(3) := 1; 3 BEGIN 4     WHILE i &lt;= 100 ← 5         DBMS_OUTPUT.PUT_LINE(i); 6         i:=i+1; 7     END LOOP; 8 END; 9 /</pre>
<b>Oracle Error Message</b>	<p>ORA-06550: line 5, column 7:          PLS-00103: Encountered the symbol "DBMS_OUTPUT" when expecting one of the following:          * &amp; - + / at loop mod remainder rem &lt;an exponent (**)&gt; and or             multiset          The symbol "*" was substituted for "DBMS_OUTPUT" to continue.</p>
<b>Mistake Line</b>	Line 4
<b>Problem Details</b>	This error message is from the parser. "LOOP" keyword is missing.
<b>Possible Solution</b>	By including the LOOP keyword.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "DBMS\_OUTPUT"

<b>Aim of the PL/SQL Program</b>	To display biggest of two numbers.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     n1  INT:=24; 3     n2  INT:=55; 4 BEGIN 5     IF n1 &gt; n2 THEN 6         DBMS_OUTPUT.PUT_LINE(n1    ' is BIG'); 7     ELSIF n2 &gt; n1 8         DBMS_OUTPUT.PUT_LINE(n2    ' is BIG'); 9     ELSE 10        DBMS_OUTPUT.PUT_LINE(n2    ' Both are Equal'); 11    END IF; 12 END; 13 /</pre>
<b>Oracle Error Message</b>	ORA-06550: line 8, column 7: PLS-00103: Encountered the symbol "DBMS_OUTPUT" when expecting one of the following: . ( * @ % & - + / at mod remainder rem then <an exponent (**)> and or    multiset The symbol "." was substituted for "DBMS_OUTPUT" to continue.
<b>Mistake Line</b>	Line 7
<b>Problem Details</b>	This error message is from the parser. "THEN" keyword is missing for "ELSIF"
<b>Possible Solution</b>	By adding "THEN" keyword for "ELSIF"

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "DBMS\_OUTPUT"

<b>Aim of the PL/SQL Program</b>	To display numbers from 1 to 100
<b>PL/SQL Code with Mistakes</b>	<pre>1 BEGIN 2     FOR i IN 1..10 3         DBMS_OUTPUT.PUT_LINE(i*i); 4     END LOOP; 5 END; 6 /</pre>
<b>Oracle Error Message</b>	<p>ORA-06550: line 3, column 8: PLS-00103: Encountered the symbol "DBMS_OUTPUT" when expecting one of the following: * &amp; - + / at loop mod remainder rem &lt;an exponent (**)&gt;    multiset</p> <p>The symbol "*" was substituted for "DBMS_OUTPUT" to continue.</p>
<b>Mistake Line</b>	Line 2
<b>Problem Details</b>	This error message is from the parser. "LOOP" keyword is missing.
<b>Possible Solution</b>	By adding "LOOP" keyword.



## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "DBMS\_OUTPUT"

<b>Aim of the PL/SQL Program</b>	To swap two numbers and display two number after swapping.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     v1  INT:=1; 3     v2  INT:=2; 4     PROCEDURE swap(p1 IN OUT INT, p2 IN OUT INT) 5     IS 6         l  INT; 7     BEGIN 8         l:=p1; 9         p1:=p2; 10        p2:=l; 11    END swap; 12 BEGIN 13    swap(v1, v2) 14    DBMS_OUTPUT.PUT_LINE('v1 value = '  v1); 15    DBMS_OUTPUT.PUT_LINE('v2 value = '  v2); 16 END;</pre>
<b>Oracle Error Message</b>	ORA-06550: line 14, column 4: PLS-00103: Encountered the symbol "DBMS_OUTPUT" when expecting one of the following:
<b>Mistake Line</b>	Line 13
<b>Problem Details</b>	This error message is from the parser. Semicolon (;) is missing.
<b>Possible Solution</b>	By giving semicolon (;).

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "DBMS\_OUTPUT"

<b>Aim of the PL/SQL Program</b>	To display 1 to 10 numbers along with string "PL/SQL".
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     n NUMBER:=1; 3 BEGIN 4     IF n &gt; 10 THEN 5         EXIT; 6     END IF; 7     DBMS_OUTPUT.PUT_LINE(n  '. PL/SQL'); 8     n:=n+1; 9     END LOOP; 10    DBMS_OUTPUT.PUT_LINE('End of Program'); 11 END; 12 /</pre>
<b>Oracle Error Message</b>	ORA-06550: line 10, column 5: PLS-00103: Encountered the symbol "DBMS_OUTPUT"
<b>Mistake Line</b>	Before Line 4
<b>Problem Details</b>	This error message is from the parser. LOOP is missing for END LOOP
<b>Possible Solution</b>	By including LOOP keyword before "IF" i.e., line 4

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "DBMS\_OUTPUT"

<b>Aim of the PL/SQL Program</b>	Example program for CASE statement.
<b>PL/SQL Code with Mistakes</b>	<pre> 1  DECLARE 2      code CHAR:='C'; 3  BEGIN 4      CASE code 5          WHEN 'H' THEN DBMS_OUTPUT.PUT_LINE('Hyderabad'); 6          WHEN 'B' THEN DBMS_OUTPUT.PUT_LINE('Bangaluru'); 7          WHEN 'C' THEN DBMS_OUTPUT.PUT_LINE('Chennai'); 8          WHEN 'D' THEN DBMS_OUTPUT.PUT_LINE('Delhi'); 9          WHEN 'P' DBMS_OUTPUT.PUT_LINE('Pune'); 10         ELSE DBMS_OUTPUT.PUT_LINE('Invalid Code'); 11     END CASE; 12 END; 13 / </pre>
<b>Oracle Error Message</b>	<p>ORA-06550: line 9, column 16:          PLS-00103: Encountered the symbol "DBMS_OUTPUT" when expecting one of the following: * &amp; = - + &lt;/&gt; at in is mod remainder not rem then &lt;an exponent (**)&gt; &lt;&gt; or != or ~= &gt;= &lt;= &lt;&gt; and or like like2 like4 likec between    multiset member submultiset          The symbol "*" was substituted for "DBMS_OUTPUT" to continue.</p>
<b>Mistake Line</b>	Line 9
<b>Problem Details</b>	This error message is from the parser. THEN keyword is missing for WHEN.
<b>Possible Solution</b>	By adding THEN keyword for WHEN condition in line 9.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "DBMS\_OUTPUT"

<b>Aim of the PL/SQL Program</b>	Example program for CASE statement.
<b>PL/SQL Code with Mistakes</b>	<pre> 1  DECLARE 2      code CHAR:='C'; 3  BEGIN 4      CASE 5          WHEN code = 'H' THEN DBMS_OUTPUT.PUT_LINE('Hyderabad'); 6          WHEN code = 'B' THEN DBMS_OUTPUT.PUT_LINE('Bangaluru'); 7          WHEN code = 'C' THEN DBMS_OUTPUT.PUT_LINE('Chennai'); 8          WHEN code = 'D' THEN DBMS_OUTPUT.PUT_LINE('Delhi'); 9          WHEN code = 'P' DBMS_OUTPUT.PUT_LINE('Pune'); 10         ELSE DBMS_OUTPUT.PUT_LINE('Invalid Code'); 11     END CASE; 12 END; 13 / </pre>
<b>Oracle Error Message</b>	<p>ORA-06550: line 9, column 23:          PLS-00103: Encountered the symbol "DBMS_OUTPUT" when expecting one of the following:              * &amp; - + / at mod remainder rem then &lt;an exponent (**)&gt; and or                 multiset          The symbol "*" was substituted for "DBMS_OUTPUT" to continue.</p>
<b>Mistake Line</b>	Line 9
<b>Problem Details</b>	This error message is from the parser. THEN keyword is missing for WHEN.
<b>Possible Solution</b>	By adding "THEN" keyword for "WHEN" condition in line 9.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "DBMS\_OUTPUT"

<b>Aim of the PL/SQL Program</b>	Example program for CASE statement.
<b>PL/SQL Code with Mistakes</b>	<pre> 1 DECLARE 2     code CHAR:='C'; 3 BEGIN 4     CASE 5         WHEN code = 'H' THEN DBMS_OUTPUT.PUT_LINE('Hyderabad'); 6         WHEN code = 'B' THEN DBMS_OUTPUT.PUT_LINE('Bangaluru'); 7         WHEN code = 'C' THEN DBMS_OUTPUT.PUT_LINE('Chennai'); 8         WHEN code = 'D' THEN DBMS_OUTPUT.PUT_LINE('Delhi'); 9         WHEN code = 'P' THEN DBMS_OUTPUT.PUT_LINE('Pune'); 10        END CASE; 11    EXCEPTION 12        WHEN CASE_NOT_FOUND ← 13            DBMS_OUTPUT.PUT_LINE('Invalid Code'); 14 END; 15 / </pre>
<b>Oracle Error Message</b>	ORA-06550: line 13, column 8: PLS-00103: Encountered the symbol "DBMS_OUTPUT" when expecting one of the following: . then or The symbol "then" was substituted for "DBMS_OUTPUT" to continue.
<b>Mistake Line</b>	Line 12
<b>Problem Details</b>	This error message is from the parser. THEN keyword is missing for WHEN statement in EXCEPTION section.
<b>Possible Solution</b>	By adding THEN keyword at end of WHEN statement in EXCEPTION section.


## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "DECLARE"

<b>Aim of the PL/SQL Program</b>	To return square of a given number.
<b>PL/SQL Code with Mistakes</b>	<pre> 1 CREATE or replace FUNCTION f_square (n NUMBER) return number is 2 DECLARE ← 3     p NUMBER; 4 BEGIN 5     p:=n * n; 6     return p; 7 END; 8 / </pre>
<b>Oracle Error Message</b>	<p>Errors: FUNCTION F_SQUARE  Line/Col: 2/1 PLS-00103: Encountered the symbol "DECLARE" when expecting one of the following:</p> <ul style="list-style-type: none"> <li>begin function pragma procedure subtype type &lt;an identifier&gt;</li> <li>&lt;a double-quoted delimited-identifier&gt; current cursor delete</li> <li>exists prior external language</li> </ul>
<b>Mistake Line</b>	Line 2
<b>Problem Details</b>	This error message is from the parser. "DECLARE" keyword after "IS" keyword.
<b>Possible Solution</b>	By removing "DECLARE" keyword after "IS" keyword.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "ELSIF"

<b>Aim of the PL/SQL Program</b>	To display biggest of two numbers.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     n1   INT:=24; 3     n2   INT:=55; 4 BEGIN 5     IF n1 &gt; n2 THEN 6         DBMS_OUTPUT.PUT_LINE(n1    ' is BIG') 7     ELSIF n2 &gt; n1 THEN 8         DBMS_OUTPUT.PUT_LINE(n2    ' is BIG'); 9     ELSE 10        DBMS_OUTPUT.PUT_LINE(n2    ' Both are Equal'); 11    END IF; 12 END; 13 /</pre> 
<b>Oracle Error Message</b>	ORA-06550: line 7, column 4: PLS-00103: Encountered the symbol "ELSIF" when expecting one of the following: := . ( % ;
<b>Mistake Line</b>	Line 6
<b>Problem Details</b>	This error message is from the parser. The statement termination character i.e., semicolon (;) is missing.
<b>Possible Solution</b>	By including semicolon (;) at end of DBMS_OUTPUT statement.

## Learn PL/SQL from Mistakes

---


**Oracle Error Message: PLS-00103: Encountered the symbol "END"**

(from Page #33 to Page #38)



## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "END"

<b>Aim of the PL/SQL Program</b>	To display biggest of two numbers.
<b>PL/SQL Code with Mistakes</b>	<pre> 1  DECLARE 2      n1   INT:=24; 3      n2   INT:=55; 4  BEGIN 5      IF n1 &gt; n2 THEN 6          DBMS_OUTPUT.PUT_LINE(n1    ' is BIG'); 7      ELSIF n2 &gt; n1 THEN 8          DBMS_OUTPUT.PUT_LINE(n2    ' is BIG'); 9      ELSE 10         DBMS_OUTPUT.PUT_LINE(n2    ' Both are Equal') 11     END IF; 12 END; 13 / </pre> 
<b>Oracle Error Message</b>	ORA-06550: line 11, column 4: PLS-00103: Encountered the symbol "END" when expecting one of the following: := . ( % ; The symbol ";" was substituted for "END" to continue.
<b>Mistake Line</b>	Line 10
<b>Problem Details</b>	This error message is from the parser. The statement termination character i.e., semicolon (;) is missing.
<b>Possible Solution</b>	By including semicolon (;) at end of DBMS_OUTPUT statement.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "END"

<b>Aim of the PL/SQL Program</b>	<p>To display 1 to 10 numbers along with string "PL/SQL".</p> <pre style="background-color: #f0f0f0; padding: 10px;"> 1 DECLARE 2     n NUMBER:=1; 3 BEGIN 4     LOOP 5         IF n &gt; 10 THEN 6             EXIT; 7         END IF; 8         DBMS_OUTPUT.PUT_LINE(n  '. PL/SQL'); 9         n:=n+1 ← 10        END LOOP; 11        DBMS_OUTPUT.PUT_LINE('End of Program'); 12    END; 13 /</pre>
<b>PL/SQL Code with Mistakes</b>	
<b>Oracle Error Message</b>	<p>ORA-06550: line 10, column 5:          PLS-00103: Encountered the symbol "END" when expecting one of the following:          * &amp; = - + ; &lt; / &gt; at in is mod remainder not rem          &lt;an exponent (**)&gt; &lt;&gt; or != or ~= &gt;= &lt;= &lt;&gt; and or like like2          like4 likec between    member submultiset          The symbol ";" was substituted for "END" to continue.</p>
<b>Mistake Line</b>	Line 9
<b>Problem Details</b>	This error message is from the parser. Semicolon (;) is missing for the expression "n:=n+1"
<b>Possible Solution</b>	By adding semicolon (;) at end of expression in line 9 i.e., n:=n+1;

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "END"

<b>Aim of the PL/SQL Program</b>	To display 1 to 10 numbers along with string "PL/SQL".
<b>PL/SQL Code with Mistakes</b>	<pre> 1 DECLARE 2     n NUMBER:=1; 3 BEGIN 4     LOOP 5     IF n &gt; 10 THEN 6         EXIT ← 7     END IF; 8         DBMS_OUTPUT.PUT_LINE(n  '. PL/SQL'); 9     n:=n+1; 10    END LOOP; 11    DBMS_OUTPUT.PUT_LINE('End of Program'); 12 END; 13 / </pre>
<b>Oracle Error Message</b>	ORA-06550: line 7, column 2: PLS-00103: Encountered the symbol "END" when expecting one of the following: ; when <an identifier> <a double-quoted delimited-identifier> The symbol ";" was substituted for "END" to continue.
<b>Mistake Line</b>	Line 6
<b>Problem Details</b>	This error message is from the parser. Semicolon (;) is missing for the statement "EXIT"
<b>Possible Solution</b>	By ending the statement "EXIT" with semicolon (;) i.e., EXIT;

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "END"

<b>Aim of the PL/SQL Program</b>	To display 1 to 10 numbers along with string "PL/SQL".
<b>PL/SQL Code with Mistakes</b>	<pre> 1  DECLARE 2      n NUMBER:=1; 3  BEGIN 4      LOOP 5          DBMS_OUTPUT.PUT_LINE(n  '. PL/SQL'); 6          n:=n+1; 7          EXIT WHEN n &gt; 10 ← 8      END LOOP; 9      DBMS_OUTPUT.PUT_LINE('End of Program'); 10 END; 11 / </pre>
<b>Oracle Error Message</b>	<p>ORA-06550: line 8, column 5:          PLS-00103: Encountered the symbol "END" when expecting one of the following:          * &amp; - + ; / at mod remainder rem &lt;an exponent (**)&gt; and or             multiset          The symbol ";" was substituted for "END" to continue.</p>
<b>Mistake Line</b>	Line 7
<b>Problem Details</b>	This error message is from the parser. Semicolon (;) is missing.
<b>Possible Solution</b>	By adding semicolon (;) at end of the statement in line 7.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "END"

<b>Aim of the PL/SQL Program</b>	Example program for nested FOR loop.
<b>PL/SQL Code with Mistakes</b>	<pre>1 BEGIN 2   &lt;&lt;outer_loop&gt;&gt; 3   FOR i IN 1..3 LOOP 4     &lt;&lt;inner_loop&gt;&gt; 5     FOR i IN 1..3 LOOP 6       IF outer_loop.i = 2 THEN 7         DBMS_OUTPUT.PUT_LINE ('outer: '    TO_CHAR(outer_loop.i) 8              ' inner: '    TO_CHAR(inner_loop.i)); 9       END IF; 10      END LOOP inner_loop ← 11    END LOOP outer_loop; 12  END;</pre>
<b>Oracle Error Message</b>	ORA-06550: line 11, column 3: PLS-00103: Encountered the symbol "END" when expecting one of the following: ; The symbol ";" was substituted for "END" to continue.
<b>Mistake Line</b>	Line 10
<b>Problem Details</b>	This error message is from the parser. Semicolon (;) is missing.
<b>Possible Solution</b>	By adding semicolon (;) at end of line 10.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "END"

<b>Aim of the PL/SQL Program</b>	Example program user defined function creation.
<b>PL/SQL Code with Mistakes</b>	<pre style="background-color: #f0f0f0; padding: 10px;"> 1 CREATE OR REPLACE FUNCTION system_time RETURN varchar2 IS 2   time_v VARCHAR2(20); 3 BEGIN 4   time_v := TO_CHAR(SYSDATE, 'HH:MI:SS A.M. '); 5   RETURN time_v; 6 EXCEPTION 7   WHEN OTHERS THEN 8     RETURN SQLERRM ← 9 END; 10 /</pre>
<b>Oracle Error Message</b>	<p>Errors: FUNCTION SYSTEM_TIME  Line/Col: 9/1 PLS-00103: Encountered the symbol "END" when expecting one of the following:</p> <p>. ( * @ % &amp; = - + ; &lt; / &gt; at in is mod remainder not rem  &lt;an exponent (**)&gt; &lt;&gt; or != or ~= &gt;= &lt;= &lt;&gt; and or like like2  like4 likec between    multiset member submultiset  The symbol ";" was substituted for "END" to continue.</p>
<b>Mistake Line</b>	Line 8
<b>Problem Details</b>	This error message is from the parser. Semicolon (;) is missing.
<b>Possible Solution</b>	By adding semicolon (;) at end of the statement in line 8.

## Learn PL/SQL from Mistakes

---

**Oracle Error Message: PLS-00103: Encountered the symbol "end-of-file"**

(from Page #40 to Page #41)

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "end-of-file"

<b>Aim of the PL/SQL Program</b>	To display numbers from 1 to 100
<b>PL/SQL Code with Mistakes</b>	<pre> 1 DECLARE 2     i    NUMBER(3) := 1; 3 BEGIN 4     WHILE i &lt;= 100 LOOP 5         DBMS_OUTPUT.PUT_LINE(i); 6         i:=i+1; 7     END LOOP; 8 EN; ← 9 / </pre>
<b>Oracle Error Message</b>	<p>ORA-06550: line 8, column 3:          PLS-00103: Encountered the symbol "end-of-file" when expecting one of the following:          ( begin case declare end exception exit for goto if loop mod null pragma raise return          select update while with &lt;an identifier&gt; &lt;a double-quoted delimited-identifier&gt; &lt;a bind          variable&gt; &lt;&lt; continue close current delete fetch lock insert open rollback savepoint set sql          execute commit forall merge pipe purge json_exists json_value json_query json_object          json_array</p>
<b>Mistake Line</b>	Line 8
<b>Problem Details</b>	This error message is from the parser. Spelling mistake in “END” keyword.
<b>Possible Solution</b>	By correcting the spelling mistake i.e., by replacing “EN” with “END”.



## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "end-of-file"

<b>Aim of the PL/SQL Program</b>	To display numbers from 1 to 100
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     i    NUMBER(3) :=1; 3 BEGIN 4     WHILE i &lt;= 100 LOOP 5         DBMS_OUTPUT.PUT_LINE(i); 6         i:=i+1; 7     END LOOP; 8 END ← 9 /</pre>
<b>Oracle Error Message</b>	ORA-06550: line 8, column 3: PLS-00103: Encountered the symbol "end-of-file" when expecting one of the following: ; <an identifier> <a double-quoted delimited-identifier> The symbol ";" was substituted for "end-of-file" to continue.
<b>Mistake Line</b>	Line 8
<b>Problem Details</b>	This error message is from the parser. Statement termination character i.e., semicolon (;) is missing.
<b>Possible Solution</b>	By giving semicolon (;) at the end of "END" keyword.

## Learn PL/SQL from Mistakes

---

**Oracle Error Message: PLS-00103: Encountered the symbol "IF"**

(from Page #43 to Page #45)

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "IF"

<b>Aim of the PL/SQL Program</b>	To display biggest of two numbers.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     n1     INT:=24; 3     n2     INT:=55; 4 BEGIN 5     IF n1 &gt; n2 THEN 6         DBMS_OUTPUT.PUT_LINE(n1    ' is BIG'); 7     ELSIF n2 &gt; n1 THEN 8         DBMS_OUTPUT.PUT_LINE(n2    ' is BIG'); 9     ELSE 10        DBMS_OUTPUT.PUT_LINE(n2    ' Both are Equal'); 11    END IF; 12    END IF; 13 END; 14 /</pre>
<b>Oracle Error Message</b>	ORA-06550: line 12, column 8: PLS-00103: Encountered the symbol "IF" when expecting one of the following: ; <an identifier> <a double-quoted delimited-identifier>
<b>Mistake Line</b>	Line 12
<b>Problem Details</b>	This error message is from the parser. An additional "END IF" without "IF".
<b>Possible Solution</b>	By removing additional "END IF".

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "IF"

<b>Aim of the PL/SQL Program</b>	To display biggest of two numbers.
<b>PL/SQL Code with Mistakes</b>	<pre> 1 DECLARE 2     n1     INT:=24; 3     n2     INT:=55; 4 BEING ← 5     IF n1 &gt; n2 THEN 6         DBMS_OUTPUT.PUT_LINE(n1    ' is BIG'); 7     ELSIF n2 &gt; n1 THEN 8         DBMS_OUTPUT.PUT_LINE(n2    ' is BIG'); 9     ELSE 10        DBMS_OUTPUT.PUT_LINE(n2    ' Both are Equal'); 11    END IF; 12 END; 13 / </pre>
<b>Oracle Error Message</b>	ORA-06550: line 5, column 4: PLS-00103: Encountered the symbol "IF" when expecting one of the following: constant exception <an identifier> <a double-quoted delimited-identifier> table columns long double ref char time timestamp interval date binary national character nchar
<b>Mistake Line</b>	Line 4
<b>Problem Details</b>	This error message is from the parser. Spelling mistake in “BEGIN” keyword.
<b>Possible Solution</b>	By correcting the spelling mistake i.e., by replacing “BEING” with “BEGIN”.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "IF"

<b>Aim of the PL/SQL Program</b>	To display 1 to 10 numbers along with string "PL/SQL".
<b>PL/SQL Code with Mistakes</b>	<pre> 1  DECLARE 2      n NUMBER:=1; 3  LOOP 4      IF n &gt; 10 THEN 5          EXIT; 6      END IF; 7      DBMS_OUTPUT.PUT_LINE(n  '. PL/SQL'); 8      n:=n+1; 9  END LOOP; 10 DBMS_OUTPUT.PUT_LINE('End of Program'); 11 END; 12 / </pre>
<b>Oracle Error Message</b>	<p>ORA-06550: line 4, column 2:          PLS-00103: Encountered the symbol "IF" when expecting one of the following:          constant exception &lt;an identifier&gt;          &lt;a double-quoted delimited-identifier&gt; table columns long          double ref char time timestamp interval date binary national          character nchar</p>
<b>Mistake Line</b>	Before Line 3
<b>Problem Details</b>	This error message is from the parser. BEGIN keyword is missing.
<b>Possible Solution</b>	By adding BEGIN keyword before "LOOP" keyword

## Learn PL/SQL from Mistakes


---

**Oracle Error Message: PLS-00103: Encountered the symbol "IS"**

(from Page #47 to Page #50)


## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "IS"

<b>Aim of the PL/SQL Program</b>	To return square of a given number.
<b>PL/SQL Code with Mistakes</b>	<pre>1 CREATE or replace FUNCTION f_square (n NUMBER) is 2     p NUMBER; 3 BEGIN 4     p:=n * n; 5     return p; 6 END; 7 /</pre> 
<b>Oracle Error Message</b>	Errors: FUNCTION F_SQUARE Line/Col: 1/31 PLS-00103: Encountered the symbol "IS" when expecting one of the following: return
<b>Mistake Line</b>	Line 1
<b>Problem Details</b>	This error message is from the parser. RETURN <datatype> is missing.
<b>Possible Solution</b>	By adding RETURN NUMBER before "IS" keyword.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "IS"

<b>Aim of the PL/SQL Program</b>	Example program for TYPE and declared FUNCTION usage
<b>PL/SQL Code with Mistakes</b>	<pre> 1 DECLARE 2     TYPE sno_t IS TABLE OF NUMBER; 3     nums_v    sno_t := sno_t(1,2,3,4,5,6,7,8,9,10) 4     FUNCTION avg_fun (nums_p IN sno_t) RETURN NUMBER IS 5         i      PLS_INTEGER:=nums_p.FIRST; 6         avg_v   NUMBER:=0; 7 BEGIN 8     WHILE i IS NOT NULL LOOP 9         avg_v := avg_v + nums_p(i) / nums_p.COUNT; 10        i := nums_p.NEXT(i); 11    END LOOP; 12    RETURN avg_v; 13 END avg_fun; 14 BEGIN 15     DBMS_OUTPUT.PUT_LINE(avg_fun(nums_v)); 16 END; </pre> 
<b>Oracle Error Message</b>	<p>ORA-06550: line 4, column 4:          PLS-00103: Encountered the symbol "FUNCTION" when expecting one of the following: . ( * % &amp; = - + ; &lt;/&gt; at in is mod remainder not rem          &lt;an exponent (**)&gt; &lt;&gt; or != or ~= &gt;= &lt;= &lt;&gt; and or like like2 like4 likec between    multiset member submultiset The symbol ";" was substituted for "FUNCTION" to continue.</p>
<b>Mistake Line</b>	Line 3
<b>Problem Details</b>	This error message is from the parser. Semicolon (;) is missing.
<b>Possible Solution</b>	By adding semicolon (;) at end of "num_v" declaration.




## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "IS"

<b>Aim of the PL/SQL Program</b>	Example program for TYPE and declared FUNCTION usage
<b>PL/SQL Code with Mistakes</b>	<pre> 1 DECLARE 2     TYPE sno_t IS TABLE OF NUMBER(2); 3     nums_v     sno_t := sno_t(1,2,3,4,5,6,7,8,9,10); 4     FUNCTION avg_fun (nums_p IN sno_t) RETURN NUMBER IS 5         i           PLS_INTEGER:=nums_p.FIRST; 6         avg_v       NUMBER:=0; 7 BEGIN 8     WHILE i IS NOT NULL LOOP 9         avg_v := avg_v + nums_p(i) / nums_p.COUNT; 10        i := nums_p.NEXT(i); 11        ENDL00P; 12        RETURN avg_v; 13 END avg_fun; 14 BEGIN 15     DBMS_OUTPUT.PUT_LINE(avg_fun(nums_v)); 16 END; </pre>
<b>Oracle Error Message</b>	ORA-06550: line 13, column 5: PLS-00103: Encountered the symbol "AVG_FUN" when expecting one of the following: loop The symbol "loop" was substituted for "AVG_FUN" to continue.
<b>Mistake Line</b>	Line 11
<b>Problem Details</b>	This error message is from the parser. Space is missing in END LOOP.
<b>Possible Solution</b>	By giving at least one space between "END" and "LOOP" keywords.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "IS"

<b>Aim of the PL/SQL Program</b>	Example program for TYPE and declared FUNCTION usage
<b>PL/SQL Code with Mistakes</b>	<pre style="background-color: #f0f0f0; padding: 10px;"> 1  DECLARE 2      TYPE sno_t IS TABLE OF NUMBER(2); 3      nums_v    sno_t := sno_t(1,2,3,4,5,6,7,8,9,10); 4      FUNCTION avg_fun (nums_p IN sno_t) IS 5                 i          PLS_INTEGER:=nums_p.FIRST; 6                 avg_v      NUMBER:=0; 7  BEGIN 8      WHILE i IS NOT NULL LOOP 9                 avg_v := avg_v + nums_p(i) / nums_p.COUNT; 10                i := nums_p.NEXT(i); 11         END LOOP; 12         RETURN avg_v; 13     END avg_fun; 14     BEGIN 15         DBMS_OUTPUT.PUT_LINE(avg_fun(nums_v)); 16     END;</pre> 
<b>Oracle Error Message</b>	ORA-06550: line 4, column 41: PLS-00103: Encountered the symbol "IS" when expecting one of the following: return
<b>Mistake Line</b>	Line 4
<b>Problem Details</b>	This error message is from the parser. RETURN <datatype> is missing.
<b>Possible Solution</b>	By including RETURN NUMBER before "IS" keyword

## Learn PL/SQL from Mistakes

---

**Oracle Error Message: PLS-00103: Encountered the symbol "LOOP"**

(from Page #52 to Page #53)

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "LOOP"

<b>Aim of the PL/SQL Program</b>	To display 1 to 10 numbers along with string "PL/SQL".
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     n NUMBER:=1; 3 BEGIN 4     LOOP 5         IF n &gt; 10 THEN 6             EXIT; 7             DBMS_OUTPUT.PUT_LINE(n  '. PL/SQL'); 8             n:=n+1; 9         END LOOP; 10    DBMS_OUTPUT.PUT_LINE('End of Program'); 11 END; 12 /</pre>
<b>Oracle Error Message</b>	ORA-06550: line 9, column 9: PLS-00103: Encountered the symbol "LOOP" when expecting one of the following: if
<b>Mistake Line</b>	Before Line 9
<b>Problem Details</b>	This error message is from the parser. "END IF;" is missing for IF statement.
<b>Possible Solution</b>	By adding "END IF;" keyword before "END LOOP;" i.e., line 9

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "LOOP"

<b>Aim of the PL/SQL Program</b>	Example program for nested FOR loop.
<b>PL/SQL Code with Mistakes</b>	<pre>1 BEGIN 2   &lt;&lt;outer_loop&gt;&gt; 3   FOR i IN 1..3 LOOP 4     &lt;&lt;inner_loop&gt;&gt; 5     FOR i IN 1..3 LOOP 6       IF outer_loop.i = 2 THEN 7         DBMS_OUTPUT.PUT_LINE ('outer: '    TO_CHAR(outer_loop.i) 8              ' inner: '    TO_CHAR(inner_loop.i)); 9         ENDIF; 10      END LOOP inner_loop; 11    END LOOP outer_loop; 12  END;</pre>
<b>Oracle Error Message</b>	ORA-06550: line 10, column 9: PLS-00103: Encountered the symbol "LOOP" when expecting one of the following: if
<b>Mistake Line</b>	Line 9
<b>Problem Details</b>	This error message is from the parser. Space is missing in between END and IF.
<b>Possible Solution</b>	By giving at least one space between END and IF.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "THEM"

<b>Aim of the PL/SQL Program</b>	To display biggest of two numbers.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     n1  INT:=24; 3     n2  INT:=55; 4 BEGIN 5     IF n1 &gt; n2 THEM 6         DBMS_OUTPUT.PUT_LINE(n1    ' is BIG'); 7     ELIF n2 &gt; n1 THEN 8         DBMS_OUTPUT.PUT_LINE(n2    ' is BIG'); 9     ELSE 10        DBMS_OUTPUT.PUT_LINE(n1    ' and '    n2    ' are Equal'); 11    END IF; 12 END; 13 /</pre>
<b>Oracle Error Message</b>	ORA-06550: line 5, column 15: PLS-00103: Encountered the symbol "THEM" when expecting one of the following: . ( * @ % & - + / at mod remainder rem then <an exponent (**)> and or    multiset
<b>Mistake Line</b>	Line 5
<b>Problem Details</b>	This error message is from the parser. Spelling mistake in "THEM" keyword.
<b>Possible Solution</b>	By replacing "THEM" with "THEN".

## Learn PL/SQL from Mistakes


---

**Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"**

(from Page #56 to Page #78)

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	To display numbers from 1 to 100
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     i    NUMBER(3):=1; 3 BEGIN 4     WHILE i &lt;= 100 LOOP 5         DBMS_OUTPUT.PUT_LINE(i) 6         i:=i+1; 7     END LOOP; 8 END; 9 / 10</pre> 
<b>Oracle Error Message</b>	ORA-06550: line 6, column 7: PLS-00103: Encountered the symbol "I" when expecting one of the following: := . ( % ; The symbol "." was substituted for "I" to continue.
<b>Mistake Line</b>	Line 5
<b>Problem Details</b>	This error message is from the parser. Statement termination character i.e., semicolon (;) is missing.
<b>Possible Solution</b>	By giving semicolon (;) at the end of DBMS_OUTPUT.PUT_LINE(i) statement.



## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	To display numbers from 1 to 100
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     i     NUMBER(3):=1; 3 BEING ← 4     WHILE i &lt;= 100 LOOP 5         DBMS_OUTPUT.PUT_LINE(i); 6         i:=i+1; 7     END LOOP; 8 END; 9 /</pre>
<b>Oracle Error Message</b>	ORA-06550: line 4, column 10: PLS-00103: Encountered the symbol "I" when expecting one of the following: := . ( @ % ; not null range default character
<b>Mistake Line</b>	Line 3
<b>Problem Details</b>	This error message is from the parser. Spelling mistake in "BEGIN" keyword.
<b>Possible Solution</b>	By correcting the spelling mistake i.e., by replacing "BEING" with "BEGIN".

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	To display numbers from 1 to 100
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     i    NUMBER(3):=1; 3 BEGIN 4     WHIE i &lt;= 100 LOOP 5         ↑ DBMS_OUTPUT.PUT_LINE(i); 6         i:=i+1; 7     END LOOP; 8 END; 9 /</pre>
<b>Oracle Error Message</b>	ORA-06550: line 4, column 9: PLS-00103: Encountered the symbol "I" when expecting one of the following: := . ( @ % ;
<b>Mistake Line</b>	Line 4
<b>Problem Details</b>	This error message is from the parser. Spelling mistake in “WHIE” keyword.
<b>Possible Solution</b>	By correcting the spelling mistake i.e., by replacing “WHIE” with “WHILE”.


## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	To display biggest of two numbers.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     n1  INT:=24; 3     n2  INT:=55; 4 BEGIN 5     IF n1 &gt; n2 THEN 6         DBMS_OUTPUT.PUT_LINE(n1    ' is BIG'); 7     ELSEIF n2 &gt; n1 THEN 8         DBMS_OUTPUT.PUT_LINE(n2    ' is BIG'); 9     ELSE 10        DBMS_OUTPUT.PUT_LINE(n2    ' Both are Equal'); 11    END IF; 12 END; 13 /</pre>
<b>Oracle Error Message</b>	ORA-06550: line 7, column 11: PLS-00103: Encountered the symbol "N2" when expecting one of the following: := . ( @ % ;
<b>Mistake Line</b>	Line 7
<b>Problem Details</b>	This error message is from the parser. Spelling mistake in keyword "ELSEIF".
<b>Possible Solution</b>	By replacing "ELSEIF" with "ELSIF".


## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	To display numbers from 1 to 100
<b>PL/SQL Code with Mistakes</b>	<pre style="background-color: #f0f0f0; padding: 5px;"> 1 BEGIN 2     DBMS_OUTPUT.PUT_LINE(TO_CHAR(SYSDATE, 'FMSSSSStH)); 3 END; 4 /</pre> 
<b>Oracle Error Message</b>	<p>ORA-06550: line 2, column 42:          PLS-00103: Encountered the symbol "FMSSSSStH));          END;" when expecting one of the following:</p> <p>( - + case mod new not null &lt;an identifier&gt;          &lt;a double-quoted delimited-identifier&gt; &lt;a bind variable&gt;          continue avg count current exists max min prior sql stddev          sum variance execute forall merge time timestamp interval          date &lt;a string literal with character set specification&gt;          &lt;a number&gt; &lt;a single-quoted SQL string&gt; pipe          &lt;an alternatively-quoted string literal with cha</p>
<b>Mistake Line</b>	Line 2
<b>Problem Details</b>	This error message is from the parser. Missing single quote at end of FMSSSSStH.
<b>Possible Solution</b>	By adding single quote at end of FMSSSSStH.


## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	To display today date
<b>PL/SQL Code with Mistakes</b>	<pre>1 BEGIN 2     DBMS_OUTPUT.PUT_LINE('Today's Date is '    SYSDATE); 3 END; 4 /</pre> 
<b>Oracle Error Message</b>	ORA-06550: line 2, column 32: PLS-00103: Encountered the symbol "S" when expecting one of the following: ) , * & = - + < / > at in is mod remainder not rem => <an exponent (**)> <> or != or ~= >=<=<> and or default like like2 like4 likec as between from using    multiset member submultiset
<b>Mistake Line</b>	Line 2
<b>Problem Details</b>	This error message is from the parser. Missing additional single quote.
<b>Possible Solution</b>	By adding additional single quote.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	To return square of a given number.
<b>PL/SQL Code with Mistakes</b>	<pre style="background-color: #f0f0f0; padding: 10px;"> 1 CREATE or replace FUNCTION f_square (n NUMBER(20)) return number is 2     p NUMBER; 3 BEGIN 4     p:=n * n; 5     return p; 6 END; 7 /</pre> 
<b>Oracle Error Message</b>	<p>Errors: FUNCTION F_SQUARE  Line/Col: 1/29 PLS-00103: Encountered the symbol "(" when expecting one of the following:  := . ) , @ % default character  The symbol "!=" was substituted for "(" to continue.</p>
<b>Mistake Line</b>	Line 1
<b>Problem Details</b>	This error message is from the parser. Datatype size specification in parameter.
<b>Possible Solution</b>	By removing size i.e., 20 in NUMBER parameter.


## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	To swap two numbers and display two number after swapping.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     v1  INT:=1; 3     v2  INT:=2; 4     PROCEDURE swap(p1 IN OUT INT, p2 IN OUT INT) 5     IS 6         l  INT; 7     BEGIN 8         l:=p1; 9         p1:=p2; 10        p2:=l; 11    END swap 12 BEGIN 13     swap(v1, v2); 14     DBMS_OUTPUT.PUT_LINE('v1 value = '  v1); 15     DBMS_OUTPUT.PUT_LINE('v2 value = '  v2); 16 END;</pre>
<b>Oracle Error Message</b>	ORA-06550: line 12, column 1: PLS-00103: Encountered the symbol "BEGIN" when expecting one of the following: ; The symbol ";" was substituted for "BEGIN" to continue.
<b>Mistake Line</b>	Line 11
<b>Problem Details</b>	This error message is from the parser. Semicolon (;) is missing.
<b>Possible Solution</b>	By adding semicolon (;) at end of line 11.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	To swap two numbers and display two number after swapping.
<b>PL/SQL Code with Mistakes</b>	<pre> 1  DECLARE 2     v1  INT:=1; 3     v2  INT:=2; 4     PROCEDURE swap(p1 IN OUT INT, p2 IN OUT INT) 5     IS 6         l  INT; 7     BEGIN 8         l:=p1; 9         p1:=p2; 10        p2:=l; 11    END swap; 12 BEGIN 13     swap(v1, v2); 14     DBMS_OUTPUT.PUT_LINE('v1 value = '  v1); 15     DBMS_OUTPUT.PUT_LINE('v2 value = '  v2); 16 END;</pre> 
<b>Oracle Error Message</b>	ORA-06550: line 10, column 10: PLS-00103: Encountered the symbol "P2" when expecting one of the following: . ( * @ % & = - + ; < / > at in is mod remainder not rem..... between    multiset member submultiset The symbol ";" was substituted for "P2" to continue.
<b>Mistake Line</b>	Line 9
<b>Problem Details</b>	This error message is from the parser. Semicolon (;) is missing.
<b>Possible Solution</b>	By adding semicolon (;) at end of line 9.



## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	To swap two numbers and display two number after swapping.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     v1  INT:=1; 3     v2  INT:=2; 4     PROCEDURE swap(p1 IN OUT INT, p2 IN OUT INT) 5     IS 6     DECLARE ← 7         l INT; 8     BEGIN 9         l:=p1; 10        p1:=p2; 11        p2:=l; 12    END swap; 13 BEGIN 14     swap(v1, v2); 15     DBMS_OUTPUT.PUT_LINE('v1 value = '  v1); 16     DBMS_OUTPUT.PUT_LINE('v2 value = '  v2); 17 END;</pre>
<b>Oracle Error Message</b>	ORA-06550: line 6, column 4: PLS-00103: Encountered the symbol "DECLARE" when expecting .....current cursor delete exists prior external language
<b>Mistake Line</b>	Line 6
<b>Problem Details</b>	This error message is from the parser. DECLARE keyword is wrong inside procedure declaration.
<b>Possible Solution</b>	By removing DECLARE keyword after "IS" keyword.


## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	To swap two numbers and display two number after swapping.
<b>PL/SQL Code with Mistakes</b>	<pre> 1 DECLARE 2   v1  INT:=1; 3   v2  INT:=2 ← 4   PROCEDURE swap(p1 IN OUT INT, p2 IN OUT INT) 5   IS 6     l  INT; 7   BEGIN 8     l:=p1; 9     p1:=p2; 10    p2:=l; 11  END swap; 12 BEGIN 13   swap(v1, v2); 14   DBMS_OUTPUT.PUT_LINE('v1 value = '  v1); 15   DBMS_OUTPUT.PUT_LINE('v2 value = '  v2); 16 END;</pre>
<b>Oracle Error Message</b>	ORA-06550: line 4, column 4: PLS-00103: Encountered the symbol "PROCEDURE" when expecting one of the following: * & = - + ; < / > at in is mod remainder not rem <an exponent (**)> <> or != or ~= >= <= <> and or like like2.....The symbol ";" was substituted for "PROCEDURE" to continue.
<b>Mistake Line</b>	Line 3
<b>Problem Details</b>	This error message is from the parser. Semicolon (;) is missing at end of variable declaration.
<b>Possible Solution</b>	By giving semicolon (;) at end of "v2" variable declaration.


## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	To swap two numbers and display two number after swapping.
<b>PL/SQL Code with Mistakes</b>	<pre> 1  DECLARE 2    v1  INT:=1; 3    v2  INT:=2; 4    PROCEDURE swap(p1 IN OUT INT, p2 NUMBER IN OUT INT) 5    IS 6      l  INT; 7    BEGIN 8      l:=p1; 9      p1:=p2; 10     p2:=l; 11  END swap; 12 BEGIN 13   swap(v1, v2); 14   DBMS_OUTPUT.PUT_LINE('v1 value = '  v1); 15   DBMS_OUTPUT.PUT_LINE('v2 value = '  v2); 16 END;</pre> 
<b>Oracle Error Message</b>	ORA-06550: line 4, column 44: PLS-00103: Encountered the symbol "IN" when expecting one of the following: := .) , @ % default character
<b>Mistake Line</b>	Line 4
<b>Problem Details</b>	This error message is from the parser. NUMBER datatype before "IN OUT"
<b>Possible Solution</b>	By removing NUMBER datatype before "IN OUT" in 2 <sup>nd</sup> parameter.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	To swap two numbers and display two number after swapping.
<b>PL/SQL Code with Mistakes</b>	<pre> 1  DECLARE 2    v1  INT:=1; 3    v2  INT:=2; 4    PROCEDURE swap(p1 IN OUT INT, p2 IN OUT INT) 5        l INT; 6    BEGIN 7        l:=p1; 8        p1:=p2; 9        p2:=l; 10   END swap; 11  BEGIN 12    swap(v1, v2); 13    DBMS_OUTPUT.PUT_LINE('v1 value = '  v1); 14    DBMS_OUTPUT.PUT_LINE('v2 value = '  v2); 15  END;</pre> 
<b>Oracle Error Message</b>	ORA-06550: line 5, column 10: PLS-00103: Encountered the symbol "L" when expecting one of the following: ; is with default authid as cluster order using external deterministic parallel_enable pipelined result_cache accessible rewrite
<b>Mistake Line</b>	Line 4
<b>Problem Details</b>	This error message is from the parser. "IS" keyword is missing.
<b>Possible Solution</b>	By adding "IS" keyword at end of procedure parameters.


## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	Example program for TYPE and declared FUNCTION usage
<b>PL/SQL Code with Mistakes</b>	<pre> 1 DECLARE 2     TYPE sno_t IS TABLE OF NUMBER(2); 3     nums_v    sno_t := sno_t(1,2,3,4,5,6,7,8,9,10); 4     FUNCTION avg_fun (nums_p IN sno_t) RETURN NUMBER IS 5               i          PLS_INTEGER:=nums_p.FIRST; 6               avg_v      NUMBER:=0; 7 BEGIN 8     WHILE i IS NOT NULL 9               avg_v := avg_v + nums_p(i) / nums_p.COUNT; 10              i := nums_p.NEXT(i); 11    END LOOP; 12    RETURN avg_v; 13 END avg_fun; 14 BEGIN 15    DBMS_OUTPUT.PUT_LINE(avg_fun(nums_v)); 16 END; </pre>
<b>Oracle Error Message</b>	ORA-06550: line 9, column 10: PLS-00103: Encountered the symbol "AVG_V" when expecting one of the following: loop and or The symbol "loop" was substituted for "AVG_V" to continue.
<b>Mistake Line</b>	Line 8
<b>Problem Details</b>	This error message is from the parser. "LOOP" keyword is missing.
<b>Possible Solution</b>	By adding "LOOP" keyword.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	To display 1 to 10 numbers along with string "PL/SQL".
<b>PL/SQL Code with Mistakes</b>	<pre style="background-color: #f0f0f0; padding: 10px;"> 1  DECLARE 2      n NUMBER:=1; 3  BEGIN 4      LOOP 5          IF n &gt; 10 THEN 6              EXIT; 7          END IF; 8              DBMS_OUTPUT.PUT_LINE(n  '. PL/SQL'); 9              n:=n+1; 10         END LOOP; 11         DBMS_OUTPUT.PUT_LINE('End of Program); 12     END; 13 /</pre> 
<b>Oracle Error Message</b>	<p>ORA-06550: line 11, column 26:          PLS-00103: Encountered the symbol "End of Program);          END;" when expecting one of the following:          ( ) - + case mod new not null &lt;an identifier&gt; &lt;a double-quoted delimited-identifier&gt;          .....second timezone_hour timezone_minute timezone_region timezone_abbr time          timestamp interval date</p>
<b>Mistake Line</b>	Line 11
<b>Problem Details</b>	This error message is from the parser. For closing the string in parameter of DBMS_OUTPUT.PUT_LINE, single quote is missing.
<b>Possible Solution</b>	By including single quote.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	To display 1 to 10 numbers along with string "PL/SQL".
<b>PL/SQL Code with Mistakes</b>	<pre>1 BEGIN 2   n NUMBER:=1; 3   LOOP 4     IF n &gt; 10 THEN 5       EXIT; 6     END IF; 7     DBMS_OUTPUT.PUT_LINE(n  '. PL/SQL'); 8     n:=n+1; 9   END LOOP; 10  DBMS_OUTPUT.PUT_LINE('End of Program'); 11 END; 12 /</pre>
<b>Oracle Error Message</b>	ORA-06550: line 2, column 7: PLS-00103: Encountered the symbol "NUMBER" when expecting one of the following: := . ( @ % ; The symbol "." was substituted for "NUMBER" to continue.
<b>Mistake Line</b>	Line 2
<b>Problem Details</b>	This error message is from the parser. DECLARE section is missing for variable declaration i.e., "n NUMBER:=1;"
<b>Possible Solution</b>	By keeping variable declaration i.e., "n NUMBER:=1;" in DECLARE section.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	Example program for CASE statement.
<b>PL/SQL Code with Mistakes</b>	<pre> 1  DECLARE 2      code CHAR:='C'; 3  BEGIN 4      CASE code 5          WHEN 'H' THEN DBMS_OUTPUT.PUT_LINE('Hyderabad'); 6          WHEN 'B' THEN DBMS_OUTPUT.PUT_LINE('Bangaluru'); 7          WHEN C' THEN DBMS_OUTPUT.PUT_LINE('Chennai'); 8          WHEN 'D' THEN DBMS_OUTPUT.PUT_LINE('Delhi'); 9          WHEN 'P' THEN DBMS_OUTPUT.PUT_LINE('Pune'); 10         ELSE DBMS_OUTPUT.PUT_LINE('Invalid Code'); 11     END CASE; 12 END; 13 / </pre>
<b>Oracle Error Message</b>	<p>ORA-06550: line 7, column 13:          PLS-00103: Encountered the symbol " THEN DBMS_OUTPUT.PUT_LINE(" when expecting one of the following:          . (* @ % &amp; = - + &lt; / &gt; ' at in is mod remainder not rem then          &lt;an exponent (**)&gt; &lt;&gt; or != or ~= &gt;= &lt;= &lt;&gt; and or like like2          like4 likec between    multiset member submultiset</p>
<b>Mistake Line</b>	Line 7
<b>Problem Details</b>	This error message is from the parser. Single quote is missing.
<b>Possible Solution</b>	By giving single quote in line 7.



## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	Example program for CASE statement.
<b>PL/SQL Code with Mistakes</b>	<pre> 1  DECLARE 2      code CHAR:='C'; 3  BEGIN 4      CASE code 5          WHEN 'H' THEN DBMS_OUTPUT.PUT_LINE('Hyderabad'); 6          WHEN 'B' THEN DBMS_OUTPUT.PUT_LINE('Bangaluru'); 7          WHEN 'C' THEN DBMS_OUTPUT.PUT_LINE('Chennai'); 8          WHEN 'D' THEN DBMS_OUTPUT.PUT_LINE('Delhi'); 9          ELSE 'P' THEN DBMS_OUTPUT.PUT_LINE('Pune'); 10         ELSE DBMS_OUTPUT.PUT_LINE('Invalid Code'); 11     END CASE; 12 END; 13 / </pre>
<b>Oracle Error Message</b>	<p>ORA-06550: line 9, column 12:          PLS-00103: Encountered the symbol "P" when expecting one of the following:          ( begin case declare exit for goto if loop mod null pragma          raise return select update while with .....merge pipe purge          json_exists json_value json_query json_object json_array          The symbol "if" was substituted for "P"</p>
<b>Mistake Line</b>	Line 9
<b>Problem Details</b>	This error message is from the parser. ELSE is wrong.
<b>Possible Solution</b>	By replacing “ELSE” keyword with “WHEN” keyword.


## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	Example program for nested FOR loop.
<b>PL/SQL Code with Mistakes</b>	<pre> 1 BEGIN 2   &lt;&lt;outer_loop&gt;&gt; 3   FOR i IN 1..3 LOOP 4     &lt;&lt;inner_loop &lt;img alt="red arrow pointing to the missing &gt;&gt; symbol" data-bbox="508 358 548 388"/&gt; 5     FOR i IN 1..3 LOOP 6       IF outer_loop.i = 2 THEN 7         DBMS_OUTPUT.PUT_LINE ('outer: '    TO_CHAR(outer_loop.i) 8              ' inner: '    TO_CHAR(inner_loop.i)); 9       END IF; 10      END LOOP inner_loop; 11     END LOOP outer_loop; 12  END;</pre>
<b>Oracle Error Message</b>	<p>ORA-06550: line 5, column 5:          PLS-00103: Encountered the symbol "FOR" when expecting one of the following:          &gt;&gt;          The symbol "&gt;&gt;" was substituted for "FOR" to continue.</p>
<b>Mistake Line</b>	Line 4
<b>Problem Details</b>	This error message is from the parser. ">>" is missing for label "inner_loop".
<b>Possible Solution</b>	By adding ">>" for label "inner_loop" in line 4.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	Example program user defined function creation.
<b>PL/SQL Code with Mistakes</b>	<pre style="background-color: #f9f9f9; padding: 10px;"> 1 CREATE OR REPLACE FUNCTION system_time RETURNS varchar2 IS 2   time_v VARCHAR2(20); 3 BEGIN 4   time_v := TO_CHAR(SYSDATE, 'HH:MI:SS A.M. '); 5   RETURN time_v; 6 EXCEPTION 7   WHEN OTHERS THEN 8     RETURN SQLERRM; 9 END; 10 /</pre> 
<b>Oracle Error Message</b>	<p>Errors: FUNCTION SYSTEM_TIME            Line/Col: 1/22 PLS-00103: Encountered the symbol "RETURNS" when expecting one of the following:            (return compress compiled wrapped</p>
<b>Mistake Line</b>	Line 1
<b>Problem Details</b>	This error message is from the parser. Spelling mistake in RETURN keyword.
<b>Possible Solution</b>	By correcting the spelling mistake i.e., replacing "RETURNS" with "RETURN".


## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	Example program user defined function creation.
<b>PL/SQL Code with Mistakes</b>	<pre> 1 CREATE OR REPLACE FUNCTION system_time RETURN varchar2 IS 2     time_v VARCHAR2(20); 3 BEGIN 4     time v := TO CHAR(SYSDATE, 'HH:MI:SS A.M. '); 5     RETURNS time_v; 6 EXCEPTION 7     WHEN OTHERS THEN 8         RETURN SQLERRM; 9 END; 10 / </pre>
<b>Oracle Error Message</b>	<p>Errors: FUNCTION SYSTEM_TIME  Line/Col: 5/12 PLS-00103: Encountered the symbol "TIME_V" when expecting one of the following:  := . ( @ % ;  The symbol "!=" was substituted for "TIME_V" to continue.</p>
<b>Mistake Line</b>	Line 5
<b>Problem Details</b>	This error message is from the parser. Spelling mistake in RETURN keyword.
<b>Possible Solution</b>	By correcting the spelling mistake i.e., replacing “RETURNS” with “RETURN”.


## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	Example program user defined function creation.
<b>PL/SQL Code with Mistakes</b>	<pre> 1 CREATE OR REPLACE FUNCTION system_time() RETURN varchar2 IS 2     time_v VARCHAR2(20); 3 BEGIN 4     time_v := TO_CHAR(SYSDATE, 'HH:MI:SS A.M. '); 5     RETURN time_v; 6 EXCEPTION 7     WHEN OTHERS THEN 8         RETURN SQLERRM; 9 END; 10 / </pre> 
<b>Oracle Error Message</b>	Errors: FUNCTION SYSTEM_TIME Line/Col: 1/22 PLS-00103: Encountered the symbol ")" when expecting one of the following: <an identifier> <a double-quoted delimited-identifier> current delete exists prior
<b>Mistake Line</b>	Line 1
<b>Problem Details</b>	This error message is from the parser. Empty brackets along with function name is wrong.
<b>Possible Solution</b>	By removing empty brackets along with function name i.e., system_time

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00103: Encountered the symbol "xxxxx"

<b>Aim of the PL/SQL Program</b>	Example program user defined function creation.
<b>PL/SQL Code with Mistakes</b>	<pre style="background-color: #f0f0f0; padding: 10px;"> 1 CREATE OR REPLACE FUNCTION system_time RETURN varchar2(20) IS 2     time_v VARCHAR2(20); 3 BEGIN 4     time_v := TO_CHAR(SYSDATE, 'HH:MI:SS A.M. '); 5     RETURN time_v; 6 EXCEPTION 7     WHEN OTHERS THEN 8         RETURN SQLERRM; 9 END; 10 /</pre> <div style="text-align: right; margin-top: 10px;">  </div>
<b>Oracle Error Message</b>	<p>Errors: FUNCTION SYSTEM_TIME  Line/Col: 1/37 PLS-00103: Encountered the symbol "(" when expecting one of the following:  . @ % ; is default authid as cluster order using external  character deterministic parallel_enable pipelined aggregate  result_cache accessible rewrite</p>
<b>Mistake Line</b>	Line 1
<b>Problem Details</b>	This error message is from the parser. Width given for RETURN datatype i.e., RETURN VARCHAR2(20).
<b>Possible Solution</b>	By removing width from RETURN datatype i.e., RETURN VARCHAR2

## Learn PL/SQL from Mistakes

**Oracle Error Message: PLS-00103: PLS-00201: identifier 'xxxxx' must be declared**

<b>Aim of the PL/SQL Program</b>	Example program for nested FOR loop.
<b>PL/SQL Code with Mistakes</b>	<pre>1 BEGIN 2  &lt;&lt;out_loop&gt;&gt; 3  FOR i IN 1..3 LOOP 4    &lt;&lt;inner_loop&gt;&gt; 5    FOR i IN 1..3 LOOP 6      IF outer_loop.i = 2 THEN 7        DBMS_OUTPUT.PUT_LINE ('outer: '    TO_CHAR(outer_loop.i) 8             ' inner: '    TO_CHAR(inner_loop.i)); 9      END IF; 10   END LOOP inner_loop; 11  END LOOP outer_loop; 12 END;</pre>
<b>Oracle Error Message</b>	ORA-06550: line 6, column 10: PLS-00201: identifier 'OUTER_LOOP.I' must be declared
<b>Mistake Line</b>	Line 2
<b>Problem Details</b>	Mistake in label name.
<b>Possible Solution</b>	By replacing label name “out_loop” with “outer_loop” in line 2.

## Learn PL/SQL from Mistakes


### Oracle Error Message: PLS-00113: END identifier 'xxxxx'

<b>Aim of the PL/SQL Program</b>	To swap two numbers and display two number after swapping.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     v1  INT:=1; 3     v2  INT:=2; 4     PROCEDURE swap(p1 IN OUT INT, p2 IN OUT INT) 5     IS 6         l INT; 7     BEGIN 8         l:=p1; 9         p1:=p2; 10        p2:=l; 11    END swa; ← 12 BEGIN 13     swap(v1, v2); 14     DBMS_OUTPUT.PUT_LINE('v1 value = '  v1); 15     DBMS_OUTPUT.PUT_LINE('v2 value = '  v2); 16 END;</pre>
<b>Oracle Error Message</b>	ORA-06550: line 11, column 8: PLS-00113: END identifier 'SWA' must match 'SWAP' at line 4, column 14
<b>Mistake Line</b>	Line 11
<b>Problem Details</b>	Spelling mistake in procedure name following the “END” keyword.
<b>Possible Solution</b>	By replacing “swa” with “swap” after “END” keyword.



## Learn PL/SQL from Mistakes

**Oracle Error Message: PLS-00201: identifier 'DBMS\_OUTPUTPUT\_LINE' must be declared**

<b>Aim of the PL/SQL Program</b>	To display numbers from 1 to 100
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     i    NUMBER(3):=1; 3 BEGIN 4     WHILE i &lt;= 100 LOOP 5         DBMS_OUTPUTPUT_LINE(i); 6         i:=i+1; 7     END LOOP; 8 END; 9 /</pre> 
<b>Oracle Error Message</b>	ORA-06550: line 5, column 7: PLS-00201: identifier 'DBMS_OUTPUTPUT_LINE' must be declared
<b>Mistake Line</b>	Line 5
<b>Problem Details</b>	Dot (.) is missing in between package name and procedure name.
<b>Possible Solution</b>	By including dot (.) in between DBMS_OUTPUT and PUT_LINE.

## Learn PL/SQL from Mistakes

---

**Oracle Error Message: PLS-00201: identifier 'xxxxx' must be declared**

(from Page #83 to Page #87)


## Learn PL/SQL from Mistakes

**Oracle Error Message:** PLS-00201: identifier 'xxxxx' must be declared

<b>Aim of the PL/SQL Program</b>	To display numbers from 1 to 100
<b>PL/SQL Code with Mistakes</b>	<pre>1  DECLAR ← 2    i    NUMBER(3):=1; 3  BEGIN 4    WHILE i &lt;= 100 LOOP 5        DBMS_OUTPUT.PUT_LINE(i); 6        i:=i+1; 7    END LOOP; 8  END; 9  /</pre>
<b>Oracle Error Message</b>	Invalid statement Unsupported Command ORA-06550: line 2, column 10: PLS-00201: identifier 'T' must be declared
<b>Mistake Line</b>	Line 1
<b>Problem Details</b>	Spelling mistake in “DECLARE” keyword.
<b>Possible Solution</b>	By correcting the spelling mistake i.e., by replacing “DECLAR” with “DECLARE”.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00201: identifier 'xxxxx' must be declared

<b>Aim of the PL/SQL Program</b>	To display numbers from 1 to 100
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     i    NUMBER(2):=1; 3 BEGIN 4     WHILE i &lt;= 100 LOOP 5         DBMS_OUTPUT.PUT_LINE(i); 6         i:=j+1; 7     END LOOP; 8 END; 9 /</pre> 
<b>Oracle Error Message</b>	ORA-06550: line 6, column 10: PLS-00201: identifier 'J' must be declared
<b>Mistake Line</b>	Line 6
<b>Problem Details</b>	The declaration of variable “j” is missing.
<b>Possible Solution</b>	By replacing “j” with “i” as “j” is not required for this logic.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00201: identifier 'xxxxx' must be declared

<b>Aim of the PL/SQL Program</b>	To swap two numbers and display two number after swapping.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     v1  INT:=1; 3     v2  INT:=2; 4     PROCEDURE swap(p1 IN OUT INT, p2 IN OUT INT) 5     IS 6         l  INT; 7     BEGIN 8         l:=p1; 9         p1:=p2; 10        p2:=l; 11    END swap; 12 BEGIN 13     swa(v1, v2); 14     DBMS_OUTPUT.PUT_LINE('v1 value = '  v1); 15     DBMS_OUTPUT.PUT_LINE('v2 value = '  v2); 16 END;</pre>
<b>Oracle Error Message</b>	ORA-06550: line 13, column 4: PLS-00201: identifier 'SWA' must be declared
<b>Mistake Line</b>	Line 13
<b>Problem Details</b>	Spelling mistake in procedure name.
<b>Possible Solution</b>	By replacing “swa” with “swap”

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00201: identifier 'xxxxx' must be declared

<b>Aim of the PL/SQL Program</b>	Example program for TYPE and declared FUNCTION usage
<b>PL/SQL Code with Mistakes</b>	<pre> 1  DECLARE 2      TYPE sno_t IS TABLE OF NUMBER(2); 3      nums_v    sno_t := sno t(1,2,3,4,5,6,7,8,9,10); 4      FUNCTION avg_fun (nums_v IN sno_t) RETURN NUMBER IS 5                 i          PLS_INTEGER:=nums_p.FIRST; 6                 avg_v      NUMBER:=0; 7  BEGIN 8      WHILE i IS NOT NULL LOOP 9                 avg_v := avg_v + nums_p(i) / nums_p.COUNT; 10                i := nums_p.NEXT(i); 11         END LOOP; 12         RETURN avg_v; 13     END avg_fun; 14     BEGIN 15         DBMS_OUTPUT.PUT_LINE(avg_fun(nums_v)); 16     END;</pre>
<b>Oracle Error Message</b>	ORA-06550: line 5, column 35: PLS-00201: identifier 'NUMS_P.FIRST' must be declared
<b>Mistake Line</b>	Line 4
<b>Problem Details</b>	Parameter name is wrong.
<b>Possible Solution</b>	By replacing “num_v” with “num_p”.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00201: identifier 'xxxxx' must be declared

<b>Aim of the PL/SQL Program</b>	To display 1 to 10 numbers along with string "PL/SQL".
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     n NUM:=1; 3 BEGIN 4     LOOP 5     IF n &gt; 10 THEN 6         EXIT; 7     END IF; 8         DBMS_OUTPUT.PUT_LINE(n  '. PL/SQL'); 9     n:=n+1; 10    END LOOP; 11    DBMS_OUTPUT.PUT_LINE('End of Program'); 12 END; 13 /</pre>
<b>Oracle Error Message</b>	ORA-06550: line 2, column 7: PLS-00201: identifier 'NUM' must be declared
<b>Mistake Line</b>	Line 2
<b>Problem Details</b>	This error message is from the parser. Wrong datatype i.e., NUM.
<b>Possible Solution</b>	By replacing NUM with NUMBER datatype.

## Learn PL/SQL from Mistakes

---


**Oracle Error Message: PLS-00215: String length constraints must be in range (1 .. 32767)**

(from Page #89 to Page #90)



## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00215: String length constraints must be in range (1 .. 32767)

<b>Aim of the PL/SQL Program</b>	Example program for TYPE usage.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     str VARCHAR2(20); 3     TYPE tp IS TABLE OF VARCHAR2 NOT NULL; 4     tp_v tp:=tp(); 5 BEGIN 6     tp_v.EXTEND; 7     tp_v(1) := 'Sailaja'; 8 END; 9 /</pre> 
<b>Oracle Error Message</b>	ORA-06550: line 3, column 24: PLS-00215: String length constraints must be in range (1 .. 32767)
<b>Mistake Line</b>	Line 3
<b>Problem Details</b>	Size specification is missing with VARCHAR2 datatype in TYPE declaration.
<b>Possible Solution</b>	By specifying the size with VARCHAR2 in TYPE declaration. For example, TYPE tp IS TABLE OF VARCHAR2(20) NOT NULL.

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00215: String length constraints must be in range (1 .. 32767)

<b>Aim of the PL/SQL Program</b>	Example program user defined function creation.
<b>PL/SQL Code with Mistakes</b>	<pre> 1 CREATE OR REPLACE FUNCTION system_time RETURN varchar2 IS 2   time_v VARCHAR2; ← 3 BEGIN 4   time_v := TO_CHAR(SYSDATE, 'HH:MI:SS A.M. '); 5   RETURN time_v; 6 EXCEPTION 7   WHEN OTHERS THEN 8     RETURN SQLERRM; 9 END; 10 / </pre>
<b>Oracle Error Message</b>	Errors: FUNCTION SYSTEM_TIME Line/Col: 2/12 PLS-00215: String length constraints must be in range (1 .. 32767)
<b>Mistake Line</b>	Line 2
<b>Problem Details</b>	Datatype VARCHAR2 width is missing for the variable “time_v”.
<b>Possible Solution</b>	By specifying the valid width i.e., time_v VARCHAR2(20);

## Learn PL/SQL from Mistakes

**Oracle Error Message: PLS-00216: NUMBER precision constraint must be in range (1 .. 38)**

<b>Aim of the PL/SQL Program</b>	To display 1 to 10 numbers along with string "PL/SQL".
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     n NUMBER(1111111111) := 1; 3 BEGIN 4     LOOP 5         IF n &gt; 10 THEN 6             EXIT; 7         END IF; 8         DBMS_OUTPUT.PUT_LINE(n    '. PL/SQL'); 9         n := n + 1; 10        END LOOP; 11        DBMS_OUTPUT.PUT_LINE('End of Program'); 12    END; 13 /</pre>
<b>Oracle Error Message</b>	ORA-06550: line 2, column 14: PLS-00216: NUMBER precision constraint must be in range (1 .. 38)
<b>Mistake Line</b>	Line 2
<b>Problem Details</b>	Datatype NUMBER width is wrong.
<b>Possible Solution</b>	By giving valid width for the datatype NUMBER.


## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00302: component 'EXTEND' must be declared

<b>Aim of the PL/SQL Program</b>	To display numbers from 1 to 100
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     TYPE tp IS TABLE OF VARCHAR2(20) NOT NULL; 3     tp_v tp:=tp(); 4 BEGIN 5     tp.EXTEND; ← 6     tp_v(1) := 'Sailaja'; 7 END; 8 /</pre>
<b>Oracle Error Message</b>	ORA-06550: line 5, column 7: PLS-00302: component 'EXTEND' must be declared
<b>Mistake Line</b>	Line 5
<b>Problem Details</b>	Type variable is wrong with EXTEND method.
<b>Possible Solution</b>	By giving correct type variable with EXTEND method i.e., tp_v.EXTEND

## Learn PL/SQL from Mistakes

**Oracle Error Message: PLS-00306: wrong number or types of arguments in call to 'PUT\_LINE'**

<b>Aim of the PL/SQL Program</b>	To display biggest of two numbers.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     n1     INT:=24; 3     n2     INT:=55; 4 BEGIN 5     IF n1 &gt; n2 THEN 6         DBMS_OUTPUT.PUT_LINE(n1    ' is BIG'); 7     ELIF n2 &gt; n1 THEN 8         DBMS_OUTPUT.PUT_LINE(n2    ' is BIG'); 9     ELSE 10        DBMS_OUTPUT.PUT_LINE(n2, ' Both are Equal'); 11    END IF; 12 END; 13 /</pre> 
<b>Oracle Error Message</b>	ORA-06550: line 10, column 7: PLS-00306: wrong number or types of arguments in call to 'PUT_LINE'
<b>Mistake Line</b>	Line 10
<b>Problem Details</b>	More than one argument in PUT_LINE procedure of DBMS_OUTPUT Package.
<b>Possible Solution</b>	By replacing “,” with pipe (  ) symbol.

## Learn PL/SQL from Mistakes

**Oracle Error Message: PLS-00325: non-integral numeric literal xxxxx is inappropriate in this context**

<b>Aim of the PL/SQL Program</b>	To display 1 to 10 numbers along with string "PL/SQL".
<b>PL/SQL Code with Mistakes</b>	<pre> 1  DECLARE 2      n NUMBER(11111111111):=1; 3  BEGIN 4      LOOP 5          IF n &gt; 10 THEN 6              EXIT; 7          END IF; 8              DBMS_OUTPUT.PUT_LINE(n  '. PL/SQL'); 9          n:=n+1; 10         END LOOP; 11         DBMS_OUTPUT.PUT_LINE('End of Program'); 12     END; 13 / </pre>
<b>Oracle Error Message</b>	ORA-06550: line 2, column 14: PLS-00325: non-integral numeric literal 11111111111 is inappropriate in this context
<b>Mistake Line</b>	Line 2
<b>Problem Details</b>	Datatype NUMBER width is wrong.
<b>Possible Solution</b>	By giving valid width for the datatype NUMBER.

## Learn PL/SQL from Mistakes

---

**Oracle Error Message:** PLS-00330: invalid use of type name or subtype name

(from Page #96 to Page #97)

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00330: invalid use of type name or subtype name

<b>Aim of the PL/SQL Program</b>	Example program for TYPE usage.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     TYPE tp IS TABLE OF VARCHAR2(20) NOT NULL; 3     tp_v tp:=tp; 4 BEGIN 5     tp_v.EXTEND; 6     tp_v(1) := 'Sailaja'; 7 END; 8 /</pre>
<b>Oracle Error Message</b>	ORA-06550: line 3, column 13: PLS-00330: invalid use of type name or subtype name
<b>Mistake Line</b>	Line 3
<b>Problem Details</b>	Invalid initialization of type variable.
<b>Possible Solution</b>	By correcting initialization of type variable i.e., tp_v tp:=tp();



## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00330: invalid use of type name or subtype name

<b>Aim of the PL/SQL Program</b>	Example program for TYPE usage.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     TYPE tp IS TABLE OF VARCHAR2(20) NOT NULL; 3     tp_v tp:=tp(); 4 BEGIN 5     tp_v.EXTEND; 6     tp(1) := 'sailaja'; 7 END; 8 /</pre>
<b>Oracle Error Message</b>	ORA-06550: line 6, column 4: PLS-00330: invalid use of type name or subtype name
<b>Mistake Line</b>	Line 6
<b>Problem Details</b>	Type name is wrong
<b>Possible Solution</b>	By giving correct type name i.e., tp_v(1)

## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00355: use of pl/sql table not allowed in this context

<b>Aim of the PL/SQL Program</b>	Example program for TYPE usage.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     TYPE tp IS TABLE OF VARCHAR2(20) NOT NULL; 3     tp_v tp:=tp(); 4 BEGIN 5     tp_v().EXTEND; 6     tp_v(1) := 'Sailaja'; 7 END; 8 /</pre>
<b>Oracle Error Message</b>	ORA-06550: line 5, column 4: PLS-00355: use of pl/sql table not allowed in this context
<b>Mistake Line</b>	Line 5
<b>Problem Details</b>	Type variable usage with EXTEND method is wrong.
<b>Possible Solution</b>	By correcting type variable usage with EXTEND method i.e., tp_v.EXTEND;

## Learn PL/SQL from Mistakes

**Oracle Error Message: PLS-00363: expression '1' cannot be used as an assignment target**

<b>Aim of the PL/SQL Program</b>	To swap two numbers and display two number after swapping.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     v1  INT:=1; 3     v2  INT:=2; 4     PROCEDURE swap(p1 IN OUT INT, p2 IN OUT INT) 5     IS 6         l  INT; 7     BEGIN 8         l:=p1; 9         p1:=p2; 10        p2:=l; 11    END swap; 12 BEGIN 13    swap(1, 2); 14    DBMS_OUTPUT.PUT_LINE('v1 value = '  v1); 15    DBMS_OUTPUT.PUT_LINE('v2 value = '  v2); 16 END;</pre>
<b>Oracle Error Message</b>	ORA-06550: line 13, column 9: PLS-00363: expression '1' cannot be used as an assignment target
<b>Mistake Line</b>	Line 13
<b>Problem Details</b>	Values passed directly for the IN OUT parameters.
<b>Possible Solution</b>	By referring values instead of passing directly i.e., swap (v1, v2)

## Learn PL/SQL from Mistakes


---

**Oracle Error Message:** PLS-00363: expression 'xxxxx' cannot be used as an assignment target

(from Page #101 to Page #103)

## Learn PL/SQL from Mistakes

**Oracle Error Message: PLS-00363: expression 'xxxxx' cannot be used as an assignment target**

<b>Aim of the PL/SQL Program</b>	To display numbers from 1 to 10
<b>PL/SQL Code with Mistakes</b>	<pre>1 BEGIN 2     FOR i IN 1..10 LOOP 3         DBMS_OUTPUT.PUT_LINE(i); 4         i:=i+1; 5     END LOOP; 6 END; 7 /</pre> 
<b>Oracle Error Message</b>	ORA-06550: line 4, column 8: PLS-00363: expression 'I' cannot be used as an assignment target
<b>Mistake Line</b>	Line 4
<b>Problem Details</b>	FOR variable “i” used for increment.
<b>Possible Solution</b>	By removing the expression “i:=i+1”.


## Learn PL/SQL from Mistakes

**Oracle Error Message: PLS-00363: expression 'xxxxx' cannot be used as an assignment target**

<b>Aim of the PL/SQL Program</b>	To return square of a given number.
<b>PL/SQL Code with Mistakes</b>	<pre>1 CREATE or replace FUNCTION f_square (n NUMBER) return number is 2     p NUMBER; 3 BEGIN 4     n:=20; ← 5     p:=n * n; 6     return p; 7 END; 8 /</pre>
<b>Oracle Error Message</b>	Errors: FUNCTION F_SQUARE Line/Col: 4/5 PL/SQL: Statement ignored Line/Col: 4/5 PLS-00363: expression 'N' cannot be used as an assignment target
<b>Mistake Line</b>	Line 4
<b>Problem Details</b>	Value assigned to “IN” parameter.
<b>Possible Solution</b>	By passing value instead of assigning to “n” parameter i.e., remove the 4 <sup>th</sup> line.

## Learn PL/SQL from Mistakes

**Oracle Error Message: PLS-00363: expression 'xxxxx' cannot be used as an assignment target**

<b>Aim of the PL/SQL Program</b>	To swap two numbers and display two number after swapping.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     v1  INT:=1; 3     v2  INT:=2; 4     PROCEDURE swap(p1 IN OUT INT, p2 IN INT) 5     IS 6         l INT; 7     BEGIN 8         l:=p1; 9         p1:=p2; 10        p2:=l; 11    END swap; 12 BEGIN 13     swap(v1, v2); 14     DBMS_OUTPUT.PUT_LINE('v1 value = '  v1); 15     DBMS_OUTPUT.PUT_LINE('v2 value = '  v2); 16 END;</pre> 
<b>Oracle Error Message</b>	ORA-06550: line 10, column 10: PLS-00363: expression 'P2' cannot be used as an assignment target
<b>Mistake Line</b>	Line 4
<b>Problem Details</b>	Parameter type is wrong.
<b>Possible Solution</b>	By replacing “IN” with “IN OUT”.

## Learn PL/SQL from Mistakes

---


**Oracle Error Message: PLS-00382: expression is of wrong type**

(from Page #105 to Page #106)



## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00382: expression is of wrong type

<b>Aim of the PL/SQL Program</b>	Example program for TYPE usage.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     str VARCHAR2(20); 3     TYPE tp IS TABLE OF VARCHAR2(20) NOT NULL; 4     tp_v tp:=tp(); 5 BEGIN 6     tp_v.EXTEND; 7     tp_v := 'Sailaja'; 8 END; 9 /</pre> 
<b>Oracle Error Message</b>	ORA-06550: line 7, column 12: PLS-00382: expression is of wrong type
<b>Mistake Line</b>	Line 7
<b>Problem Details</b>	Index is missing in type variable.
<b>Possible Solution</b>	By giving index in type variable i.e., tp_v(1)


## Learn PL/SQL from Mistakes

### Oracle Error Message: PLS-00382: expression is of wrong type

<b>Aim of the PL/SQL Program</b>	Example program for CASE statement.
<b>PL/SQL Code with Mistakes</b>	<pre> 1  DECLARE 2      code CHAR:='C'; 3  BEGIN 4      CASE ← 5          WHEN 'H' THEN DBMS_OUTPUT.PUT_LINE('Hyderabad'); 6          WHEN 'B' THEN DBMS_OUTPUT.PUT_LINE('Bangaluru'); 7          WHEN 'C' THEN DBMS_OUTPUT.PUT_LINE('Chennai'); 8          WHEN 'D' THEN DBMS_OUTPUT.PUT_LINE('Delhi'); 9          WHEN 'P' THEN DBMS_OUTPUT.PUT_LINE('Pune'); 10         ELSE DBMS_OUTPUT.PUT_LINE('Invalid Code'); 11     END CASE; 12 END; 13 / </pre>
<b>Oracle Error Message</b>	ORA-06550: line 5, column 12: PLS-00382: expression is of wrong type
<b>Mistake Line</b>	Line 4
<b>Problem Details</b>	Variable is missing with “CASE” statement.
<b>Possible Solution</b>	By attaching variable “code” with “CASE” statement i.e., CASE code

## Learn PL/SQL from Mistakes

**Oracle Error Message: PLS-00573: cannot constrain scale, precision, or range of an anchored type declaration**

<b>Aim of the PL/SQL Program</b>	Example program for subtype usage.
<b>PL/SQL Code with Mistakes</b>	<pre>1 DECLARE 2     str1 VARCHAR2(10); 3     SUBTYPE str2_st IS str1%TYPE(20); 4     str3 str2_st; 5 BEGIN 6     str3:='Sailaja'; 7 END; 8 /</pre> 
<b>Oracle Error Message</b>	ORA-06550: line 3, column 23: PLS-00573: cannot constrain scale, precision, or range of an anchored type declaration
<b>Mistake Line</b>	Line 3
<b>Problem Details</b>	Size specification for TYPE.
<b>Possible Solution</b>	By removing size including brackets after TYPE i.e., 20

## Learn PL/SQL from Mistakes

**Oracle Error Message:** PLS-00615: type mismatch found at 'xxxxx' between CASE operand and WHEN operands

<b>Aim of the PL/SQL Program</b>	Example program for CASE statement.
<b>PL/SQL Code with Mistakes</b>	<pre> 1  DECLARE 2      code CHAR:='C'; 3  BEGIN 4      CASE code 5          WHEN code = 'H' THEN DBMS_OUTPUT.PUT_LINE('Hyderabad'); 6          WHEN code = 'B' THEN DBMS_OUTPUT.PUT_LINE('Bangaluru'); 7          WHEN code = 'C' THEN DBMS_OUTPUT.PUT_LINE('Chennai'); 8          WHEN code = 'D' THEN DBMS_OUTPUT.PUT_LINE('Delhi'); 9          WHEN code = 'P' THEN DBMS_OUTPUT.PUT_LINE('Pune'); 10         ELSE DBMS_OUTPUT.PUT_LINE('Invalid Code'); 11     END CASE; 12 END; 13 / </pre>
<b>Oracle Error Message</b>	ORA-06550: line 4, column 9: PLS-00615: type mismatch found at 'CODE' between CASE operand and WHEN operands
<b>Mistake Line</b>	Line 4
<b>Problem Details</b>	Variable is not required along with “CASE” keyword.
<b>Possible Solution</b>	By removing variable “code” with “CASE” keyword in line 4.

# Learn PL/SQL from Mistakes

---

## PL/SQL Programs with Mistakes (Find the Mistakes)

### Program #1:-

```
BEGIN
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(SYSDATE, 'FMSSSSsth));
END;
/
```

### Program #2:-

```
BEGIN
  DBMS_OUTPUT.PUT_LINE('Today's Date is ' || SYSDATE);
END;
/
```

## Learn PL/SQL from Mistakes

---

### Program #3:-

```
CREATE OR REPLACE PACKAGE global_pkg
IS
  CURSOR ref_c IS SELECT line FROM user_source;
  TYPE ref_t IS REF CURSOR ref_c%ROWTYPE;
END global_pkg;
/
```

### Program #4:-

```
CREATE OR REPLACE PROCEDURE lowest_proc
(num1_p NUMBER, num2_p NUMBER, num3_p NUMBER)
DECLARE
  l_lowest NUMBER;
BEGIN
  SELECT LEAST(num1_p, num2_p, num3_p) INTO l_lowest FROM dual;
  DBMS_OUTPUT.PUT_LINE(l_lowest);
END;
/
```

## Learn PL/SQL from Mistakes

---

### Program #5:-

```
DECLARE
  i INT;
BEGIN
  FOR i IN 1..10 LOOP
    DBMS_OUTPUT.PUT_LINE(i*i);
    i:=i+1;
  END LOOP;
END;
/
```

### Program #6:-

```
DECLARE
  l_clob CLOB(4000);
BEGIN
  IF l_clob IS NULL THEN
    DBMS_OUTPUT.PUT_LINE('CLOB is NULL');
  ELSIF LENGTH(l_clob) = 0 THEN
    DBMS_OUTPUT.PUT_LINE('CLOB has a length of ZERO');
  ELSE
    DBMS_OUTPUT.PUT_LINE('CLOB has Blank Spaces');
```

## Learn PL/SQL from Mistakes

---

```
END IF;  
END;  
/
```

### Program #7:-

```
DECLARE  
  str1 VARCHAR2(10);  
  SUBTYPE str2_st IS str1%TYPE(20);  
  str3 str2_st;  
BEGIN  
  str3:='Sailaja';  
END;  
/
```

### Program #8:-

```
BEGIN  
  DBMS_SQL.EXEC_DDL_STATEMENT('CREATE TABLE employees(empno NUMBER));  
END;
```



## Learn PL/SQL from Mistakes

---

### Program #9:-

```
DECLARE
  str VARCHAR2(20);
  TYPE tp IS TABLE OF VARCHAR2(20) NOT NULL;
  tp_v tp:=tp();
BEGIN
  tp_v(1) := 'Oracle';
END;
/
```

### Program #10:-

```
DECLARE
  cur_v PLS_INTEGER;
  flag PLS_INTEGER;
  str VARCHAR2(200);
BEGIN
  str := 'UPDATE emp SET sal = sal + 1000 WHERE deptno = 20';
  cur_v := DBMS_SQL.OPEN_CURSOR();
  DBMS_SQL.PARSE(str, DBMS_SQL.NATIVE);
  flag := DBMS_SQL.EXECUTE(cur_v);
  IF flag = 0 THEN
```

## Learn PL/SQL from Mistakes

---

```
DBMS_OUTPUT.PUT_LINE('Nothing Updated');
ELSIF flag = 1 THEN
  DBMS_OUTPUT.PUT_LINE('One Record Updated');
ELSE
  DBMS_OUTPUT.PUT_LINE('More than One Record Updated');
END IF;
END;
/
```

### Program #11:-

```
DECLARE
  TYPE student_ty IS RECORD(name VARCHAR2(100), phone NUMBER);
  TYPE student_marks_ty is RECORD(subject VARCHAR2(100), marks NUMBER, student_details
student_ty);
  student_rec  student_marks_ty;

BEGIN
  student_rec.student_details.name := 'Sailaja Pasupuleti';
  student_rec.marks := 90;

  IF student_rec.phone IS NULL THEN
    DBMS_OUTPUT.PUT_LINE('Student Contact Number NOT FOUND');
```

## Learn PL/SQL from Mistakes

---

```
END IF;  
END;  
/
```

### Program #12:-

```
CREATE OR REPLACE TRIGGER emp  
  BEFORE UPDATE ON emp  
  BEFORE INSERT ON emp  
  BEFORE DELETE ON emp  
DECLARE  
BEGIN  
  IF TO_CHAR(SYSDATE, 'DY', 'NLS_DATE_LANGUAGE=ENGLISH') IN ('SAT', 'SUN') THEN  
    RAISE_APPLICATION_ERROR(-20000, 'Customers maintained Monday-Friday only.');
```

```
  END IF;  
END;  
/
```

### Program #13:-

```
CREATE TABLE emp(ename VARCHAR2(20), sal NUMBER);  
/
```

## Learn PL/SQL from Mistakes

---

```
BEGIN
  INSERT INTO emp VALUES('scott', 9000);
  INSERT INTO emp VALUES('smith', 6000);
  INSERT INTO emp VALUES('Jones', 9000);
  INSERT INTO emp VALUES('James', 5000);
  INSERT INTO emp VALUES('Adams', 6000);
  COMMIT;
END;
/
```

```
CREATE OR REPLACE PACKAGE emp_pkg
IS
  PROCEDURE emp_list;
  FUNCTION emp_total RETURN NUMBER;
END emp_pkg;
/
```

```
CREATE OR REPLACE PACKAGE BODY emp_pkg
IS
  PROCEDURE emp_list
  IS
  BEGIN
```

## Learn PL/SQL from Mistakes

---

```
FOR e IN (SELECT * FROM emp) LOOP
    DBMS_OUTPUT.PUT_LINE(e.ename,e.sal);
END LOOP;
END emp_list;
```

```
FUNCTION emp_total
IS
    ctr INT;
BEGIN
    SELECT COUNT(1) INTO ctr FROM emp;
    RETURN ctr;
END emp_total;
END emp_pkg;
/
```

### Program #14:-

```
DECLARE
    n1 INT:=&no1;
    n2 INT:=&no2;
BEGIN
    IF n1 > n2 THEN
```

## Learn PL/SQL from Mistakes

---

```
    DBMS_OUTPUT.PUT_LINE(n1 || ' is BIG');
ELSEIF n2 > n1 THEN
    DBMS_OUTPUT.PUT_LINE(n2 || ' is BIG');
ELSE
    DBMS_OUTPUT.PUT_LINE(n2 || ' Both are Equal');
ENDIF;
END;
/
```

### Program #15:-

```
CREATE OR REPLACE FUNCTION system_time RETURN varchar2(20) IS
DECLARE
    time_v VARCHAR2(20);
BEGIN
    time_v := TO_CHAR(SYSDATE, 'HH:MI:SS A.M. ');
    RETURN time_v;
EXCEPTION
    WHEN OTHERS THEN
        RETURN SQLERRM;
END;
/
```

## Learn PL/SQL from Mistakes

---

### Program #16:-

```
DECLARE
    empno_v  INTEGER(4);
    ename_v  emp.ename%TYPE;
    job_v    emp.job%TYPE;
    salary_v emp.sal%TYPE;
BEGIN
    empno_v := &Employee_Number;
    SELECT ename, job, sal FROM emp WHERE empno = empno_v;
    DBMS_OUTPUT.PUT_LINE('Employee Name : ' || ename);
    DBMS_OUTPUT.PUT_LINE('Designation  : ' || job);
    DBMS_OUTPUT.PUT_LINE('Salary      : ' || sal);
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(SQLERRM);
    WHEN VALUE_ERROR THEN
        DBMS_OUTPUT.PUT_LINE('Datatype or Width Mismatch');
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Employee Not Found');

END;
```

/

## Learn PL/SQL from Mistakes

---

### Program #17:-

```
CREATE FUNCTION f_square (n NUMBER)
  DECLARE
    p  NUMBER;
  BEGIN
    p=n * n;
    n=p;
  END;
/
```

### Program #18:-

```
DECLARE
  v1 INT:=1;
  v2 INT:=2;
CREATE OR REPLACE PROCEDURE swap(p1 OUT INT, IN OUT INT)
IS
  l INT;
BEGIN
  l:=p1;
  p1:=p2;
  p2:=l;
```



## Learn PL/SQL from Mistakes

---

```
END swap;
BEGIN
  swap(v1, v2);
  DBMS_OUTPUT.PUT_LINE('v1 value = ' || v1);
  DBMS_OUTPUT.PUT_LINE('v2 value = ' || v2);
END;
/
```

### Program #19:-

```
CREATE OR REPLACE TRIGGER emp_trig BEFORE DELETE emp FOR EACH ROW
BEGIN
  IF TO_CHAR(SYSDATE, 'D') IN (1, 7) THEN
    RAISE APPLICATION_ERROR(-20000, 'You cannot Delete Records from Emp Table Today');
  END IF;
END;
/
```

## Learn PL/SQL from Mistakes

---

### Program #20:-

```
DECLARE
  CURSOR emp_cur IS SELECT * FROM emp;
  emp_rec emp_cur%ROWTYPE;
BEGIN
  OPEN emp_cur;
  FETCH emp_cur INTO emp_rec;
  WHILE emp_cur%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE(emp_rec.ename, emp_rec.sal, emp_rec.job);
    FETCH emp_cur INTO emp_rec;
  END LOOP;
  CLOSE emp_rec;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE(SQLERRM);
  WHEN INVALID_CURSOR THEN
    DBMS_OUTPUT.PUT_LINE('Invalid Cursor Name');
END;
```

/

## Learn PL/SQL from Mistakes

---

### Program #21:-

```
BEGIN
  DBMS_OUTPUT.PUT_LINE(RPAD('Name',10) || 'Salary');
  DBMS_OUTPUT.PUT_LINE(RPAD('---',10) || '----');
DECLARE
  CURSOR emp_cur IS SELECT * FROM emp;
  emp_rec emp_cur%TYPE;
BEGIN
  OPEN emp_cur;
  LOOP
    FETCH emp_cur INTO emp_rec;
    IF emp_cur%NOTFOUND THEN
      EXIT;
    END IF;
    IF MOD(emp_cur%ROWTYPE, 2) = 0 THEN
      DBMS_OUTPUT.PUT_LINE(RPAD(emp_rec.ename,10), emp_rec.sal);
    END IF;
  END LOOP;
  CLOSE emp_rec;
EXCEPTION
  WHEN INVALID_CURSOR THEN
    DBMS_OUTPUT.PUT_LINE('Invalid Cursor Name');
```

## Learn PL/SQL from Mistakes

---

```
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE(SQLERRM);
END;
END;
/
```

### Program #22:-

```
DECLARE
    TYPE sno_t IS TABLE OF NUMBER;
    nums_v sno_t := sno_t(1,2,3,4,5,6,7,8,9,10);
    FUNCTION avg_fun (nums_p IN sno_t) RETURN NUMBER IS
        i PLS_INTEGER:=nums_p.FIRST;
        avg_v NUMBER:=0;
BEGIN
    WHILE i IS NOT NULL LOOP
        avg_v := avg_v + nums_p / nums_p.COUNT;
        i := nums_p.NEXT(i);
    END LOOP;
END avg_fun;
BEGIN
    DBMS_OUTPUT.PUT_LINE(avg_fun);
END;
```

## Learn PL/SQL from Mistakes

---

### Program #23:-

```
CREATE OR REPLACE TRIGGER emp_trg BEFORE INSERT ON emp
IS
  ctr INT:=0;
BEGIN
  SELECT COUNT(1) INTO ctr FROM emp WHERE empno = :NEW.empno;
  IF ctr != 0 THEN
    RAISE_APPLICATION_ERROR(20500, 'Employee Number already used');
  END IF;
END;
/
```

### Program #24:-

```
CREATE OR REPLECE PROCADURE display
IS
  n NUMBER;
  CURSOR emp_cur IS SELECT ename, sal, job FROM emp;
  TYPE emp_tab IS TABLE OF emp_cur%ROWTYPE;
  v_emp_tab emp_tab%ROWTYPE;
BEGIN
  FOR v_emp_cur IN emp_cur
```

## Learn PL/SQL from Mistakes

---

LOOP

n:=n+1;

v\_emp\_tab(n).ename:=v\_emp\_cur.ename;

v\_emp\_tab(n):=v\_emp\_cur.sal;

v\_emp\_tab(n).job:=v\_emp\_cur.job;

FOR n IN v\_emp\_tab.FIRST..v\_emp\_tab.LAST LOOP

DBMS\_OUTPUT.PUTLINE(RPAD(v\_emp\_tab(i).ename,10) || v\_emp\_tab(i).sal,v\_emp\_tab(i).job);

END LOOP;

END display;

# Learn PL/SQL from Mistakes

---

## Solutions for PL/SQL Program Mistakes

### Program #1:-

1) A single quote is missing in 'FMSSSSStH'

### Program #2:-

1) Additional single quote is required to display like Today's string.

```
BEGIN
  DBMS_OUTPUT.PUT_LINE('Today"s Date is ' || SYSDATE);
END;
/
```

### Program #3:-

1) RETURN keyword should be there in the TYPE definition

## Learn PL/SQL from Mistakes

---

```
CREATE OR REPLACE PACKAGE global_pkg
IS
    CURSOR ref_c IS SELECT line FROM user_source;
    TYPE ref_t IS REF CURSOR RETURN ref_c%ROWTYPE;
END global_pkg;
/
```

### Program #4:-

1) Replace "DECLARE" with "IS"

### Program #5:-

- 1) We should NOT declare "i" variable as "i" is implicitly declaring by FOR
- 2) We should NOT give i:=i+1 as increment is implicit in FOR

```
BEGIN
    FOR i IN 1..10 LOOP
        DBMS_OUTPUT.PUT_LINE(i*i);
    END LOOP;
END;
/
```



## Learn PL/SQL from Mistakes

---

### Program #6:-

1) Remove size 4000 with CLOB in DECLARE Section

### Program #7:-

1) Remove size in %TYPE

### Program #8:-

1) Replace DBMS\_SQL with DBMS\_UTILITY

### Program #9:-

1) Add the line tp\_v.EXTEND;

```
DECLARE
str VARCHAR2(20);
TYPE tp IS TABLE OF VARCHAR2(20) NOT NULL;
tp_v tp:=tp();
```

## Learn PL/SQL from Mistakes

---

```
BEGIN
  tp_v.EXTEND;
  tp_v(1) := 'Oracle';
END;
/
```

### Program #10:-

1) cur\_v as a 1st parameter to DBMS\_SQL.PARSE

```
DECLARE
  cur_v PLS_INTEGER;
  flag PLS_INTEGER;
  str VARCHAR2(200);
BEGIN
  str := 'UPDATE emp SET sal = sal + 1000 WHERE deptno = 20';
  cur_v := DBMS_SQL.OPEN_CURSOR();
  DBMS_SQL.PARSE(cur_v, str, DBMS_SQL.NATIVE);
  flag := DBMS_SQL.EXECUTE(cur_v);
  IF flag = 0 THEN
    DBMS_OUTPUT.PUT_LINE('Nothing Updated');
  ELSIF flag = 1 THEN
```

## Learn PL/SQL from Mistakes

---

```
    DBMS_OUTPUT.PUT_LINE('One Record Updated');  
ELSE  
    DBMS_OUTPUT.PUT_LINE('More than One Record Updated');  
END IF;  
END;  
/
```

### Program #11:-

1) In IF condition, mistake in "student\_rec.phone". It should be like this "student\_rec.student\_rec.phone"

### Program #12:-

1) We need to replace the following

```
BEFORE UPDATE ON emp  
BEFORE INSERT ON emp  
BEFORE DELETE ON emp
```

with BEFORE UPDATE OR INSERT OR DELETE ON emp

## Learn PL/SQL from Mistakes

---

```
CREATE OR REPLACE TRIGGER emp
  BEFORE UPDATE OR INSERT OR DELETE ON emp
DECLARE
BEGIN
  IF TO_CHAR(SYSDATE, 'DY', 'NLS_DATE_LANGUAGE=ENGLISH') IN ('SAT', 'SUN') THEN
    RAISE_APPLICATION_ERROR(-20000, 'Customers maintained Monday-Friday only.');
```

END IF;

```
END;
/
```

### Program #13:-

- 1) Replace "," with || (pipe) in DBMS\_OUTPUT.PUT\_LINE of the Procedure emp\_list
- 2) RETURN NUMBER is required in Function definition

### Program #14:-

- 1) Replace "ELSEIF" with "ELSIF"
- 2) Replace "ENDIF" with "END IF"

## Learn PL/SQL from Mistakes

---

### Program #15:-

- 1) Remove size in RETURN datatype i.e., replace VARCHAR2(20) with VARCHAR2
- 2) Remove DECLARE keyword

### Program #16:-

- 1) WHEN OTHERS THEN - should be at end
- 2) INTO Clause is mandatory in PL/SQL SELECT Statement
- 3) Replace ename, job, sal with ename\_v, job\_v, salary\_v respectively

### Program #17:-

- 1) RETURN type is missing
- 2) IS keyword is missing
- 3) remove DECLARE keyword
- 4) := is the assignment operator
- 5) should not assign a value to IN type argument (for "n" parameter)

## Learn PL/SQL from Mistakes

---

### Program #18:-

- 1) Remove "CREATE OR REPLACE" before the "PROCEDURE" keyword
- 2) Procedure parameter type should be "IN OUT", so replace "IN" with "IN OUT" for "p1" and "p2" parameters

### Program #19:-

- 1) "RAISE APPLICATION\_ERROR" is a single work, underscore between RAISE and APPLICATION words
- 2) ON keyword is required before the Table Name "emp" in Trigger definition, i.e., ON emp

### Program #20:-

- 1) Pipe Symbol "|" in place of "," in DBMS\_OUTPUT.PUT\_LINE
- 2) Cursor name "emp\_cur" instead of record name "emp\_rec" in CLOSE statement
- 3) WHEN OTHERS THEN should be at end

## Learn PL/SQL from Mistakes

---

### Program #21:-

- 1) emp\_cur is ROWTYPE in the DECLARE Part
- 2) Replace "," with Pipe Symbol "|" in DBMS\_OUTPUT.PUT\_LINE(RPAD(emp\_rec.ename,10), emp\_rec.sal);
- 3) Replace ROWTYPE with ROWCOUNT in MOD(emp\_cur%ROWCOUNT, 2)
- 4) Give cursor name with CLOSE

### Program #22:-

- 1) nums\_p(i) in place of nums\_p in the expression avg\_v := avg\_v + nums\_p / nums\_p.COUNT;  
i.e., avg\_v := avg\_v + nums\_p(i) / nums\_p.COUNT;
- 2) RETURN avg\_v; statement is required after END LOOP in the FUNCTION definition
- 3) avg\_fun(nums\_v) in place of avg\_fun in DBMS\_OUTPUT.PUT\_LINE

## Learn PL/SQL from Mistakes

---

### Program #23:-

- 1) Add FOR EACH ROW to the Trigger Definition
- 2) Replace "IS" with "DECLARE"
- 3) First parameter value in RAISE\_APPLICATION\_ERROR should be in the range -20000 to -20999

### Program #24:-

- 1) Spelling mistakes in REPLACE and PROCEDURE
- 2) Remove %ROWTYPE in v\_emp\_tab variable declaration
- 3) Initialize "n" with ZERO
- 4) Replace "i" with "n" in DBMS\_OUTPUT\_LINE
- 5) Replace "PUTLINE" with "PUT\_LINE"



## Learn PL/SQL from Mistakes

---

- 6) Replace "," with "||" in DBMS\_OUTPUT.PUT\_LINE
- 7) Give END LOOP for 2nd FOR ..... LOOP
- 8) Give column name "sal" along with v\_emp\_tab(n) i.e., replace v\_emp\_tab(n) with v\_emp\_tab(n).sal



Thank you !!!!!

Please feel free to send your comments to [sailajamail@yahoo.com](mailto:sailajamail@yahoo.com)