1z0-148.exam.47q

Number: 1z0-148
Passing Score: 800
Time Limit: 120 min

1z0-148

Oracle Database 12c: Advanced Pl/SQL

Exam A

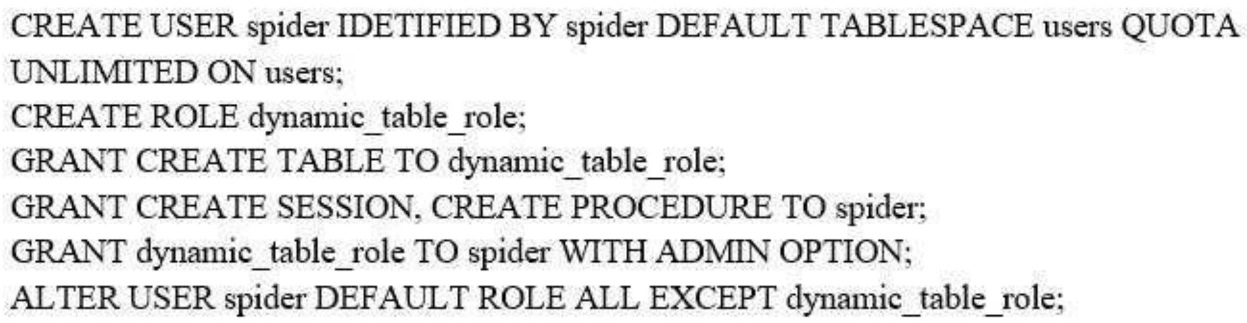
QUESTION 1
You are designing and developing a complex database application built using many dynamic SQL statements. Which option could expose your code to SQL injection attacks?

A. Using bind variables instead of directly concatenating parameters into dynamic SQL statements
B. Using automated tools to generate code
C. Not validating parameters which are concatenated into dynamic SQL statements
D. Validating parameters before concatenating them into dynamic SQL statements
E. Having excess database privileges

Correct Answer: A
Section: (none)
Explanation

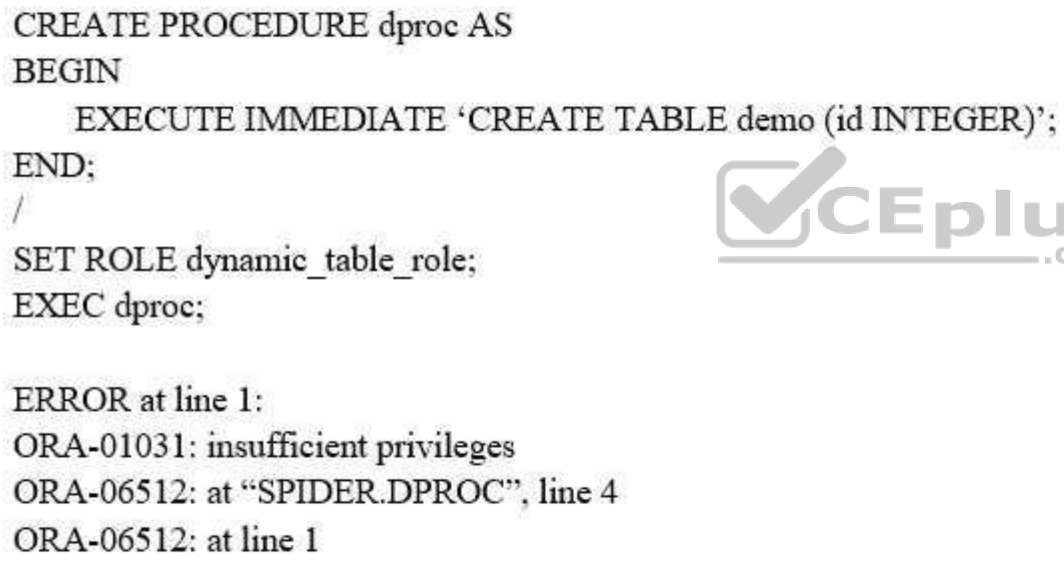
Explanation/Reference:
Reference: https://docs.oracle.com/database/121/LNPLS/dynamic.htm#LNPLS645

QUESTION 2
Examine this code executed as SYS:



```
CREATE USER spider IDETIFIED BY spider DEFAULT TABLESPACE users QUOTA
UNLIMITED ON users;
CREATE ROLE dynamic_table_role;
GRANT CREATE TABLE TO dynamic_table_role;
GRANT CREATE SESSION, CREATE PROCEDURE TO spider;
GRANT dynamic_table_role TO spider WITH ADMIN OPTION;
ALTER USER spider DEFAULT ROLE ALL EXCEPT dynamic_table_role;
```

Examine this code executed as SPIDER and the error message received upon execution:



```
CREATE PROCEDURE dproc AS
BEGIN
     EXECUTE IMMEDIATE ‘CREATE TABLE demo (id INTEGER)’;
END;
/
SET ROLE dynamic_table_role;
EXEC dproc;

ERROR at line 1:
ORA-01031: insufficient privileges
ORA-06512: at “SPIDER.DPROC”, line 4
ORA-06512: at line 1
```

What is the reason for this error?

A. The procedure needs to be granted the DYNAMIC_TABLE_ROLE role.
B. The EXECUTE IMMEDIATE clause is not supported with roles.
C. Privileges granted through roles are never in effect when running definer’s rights procedures.
D. The user SPIDER needs to be granted the CREATE TABLE privilege and the procedure needs to be granted the DYNAMIC_TABLE_ROLE.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 3

Which codes executes successfully?

```
A. CREATE PACKAGE pkg AS
      TYPE rec_typ IS RECORD (price NUMBER, inc_pct NUMBER);
      PROCEDURE calc_price (price_rec IN OUT rec_typ);
      END pkg;
      /
      CREATE PACAKGE BODY pkg AS
      PROCEDURE calc_price (price_rec IN OUT rec_typ) AS
      BEGIN
      price_rec.price  := price_rec.price + (price_rec.price * price_rec.inc_pct)/100;
      END calc_price;
      END pkg;
      /
      DECLARE
      1_rec pkg. rec_typ;
      BEGIN
      1_rec_price :=100;
      1_rec.inc_pct :=50;
      EXECUTE IMMEDIATE ‘BEGIN pkg. calc_price (:rec); END;’ USING IN OUT 1_rec;
      END;
B. CREATE PACKAGE pkg AS
      TYPE rec_typ IS RECORD (price NUMBER, inc_pct NUMBER);
      END pkg;
      /
      CREATE PROCEDURE calc_price (price_rec IN OUT pkg. rec_typ) AS
      BEGIN
      price_rec.price  := price_rec.price + (price_rec.price * price_rec.inc_pct)/100;
      END
      /
      DECLARE
```

```
    1_rec pkg.rec_typ;
    BEGIN
    EXECUTE IMMEDIATE 'BEGIN calc_price (:rec); END;' USING IN OUT 1_rec (100, 50);
    END;
C. CREATE PACKAGE pkg AS
    TYPE rec_typ IS RECORD (price NUMBER, inc_pct NUMBER);
    END pkg;
    /
    CREATE PROCEDURE calc_price (price_rec IN OUT pkg. rec_typ) AS
    BEGIN
    price_rec.price  := price_rec.price + (price_rec.price * price_rec.inc_pct)/100;
    END ;
    /
    DECLARE
    1_rec pkg. rec_typ;
    BEGIN
    1_rec_price :=100;
    1_rec.inc_pct :=50;
    EXECUTE IMMEDIATE 'BEGIN calc_price (1_rec); END;';
    END;
D. DECLARE
    TYPE rec_typ IS RECORD (price NUMBER, inc_pct NUMBER);
    1_rec rec-typ;
    PROCEDURE calc_price (price_rec IN OUT rec_typ) AS
    BEGIN
    price_rec.price := price-rec.price+ (price_rec.price * price_rec.inc_pct)/100;
    END;
    BEGIN
    1_rec_price :=100;
    1_rec.inc_pct :=50;
    EXECUTE IMMEDIATE 'BEGIN calc_price (:rec); END;' USING IN OUT 1_rec;
    END;
```

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 4
Examine this function header:

FUNCTION calc_new_sal (emp_id NUMBER) RETURN NUMBER;

You want to ensure that whenever this PL/SQL function is invoked with the same parameter value across active sessions, the result is not recomputed.

If a DML statement is modifying a table which this function depends upon, the function result must be recomputed at that point in time for all sessions calling this function.

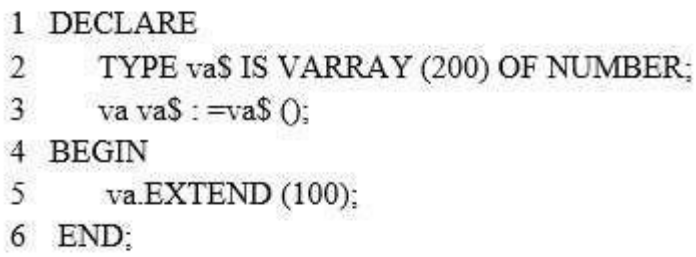
Which two actions should you perform?

A. Ensure RESULT_CACHE_MAX_SIZE is greater than 0.
B. Enable the result cache by using DBMS_RESULT_CACHE.BYPASS (FALSE).
C. Add the deterministic clause to the function definition. D. Add the RELIES_ON clause to the function definition.
E. Add the RESULT_CACHE clause to the function definition.

Correct Answer: AC
Section: (none)
Explanation

Explanation/Reference:

QUESTION 5
Examine this block:



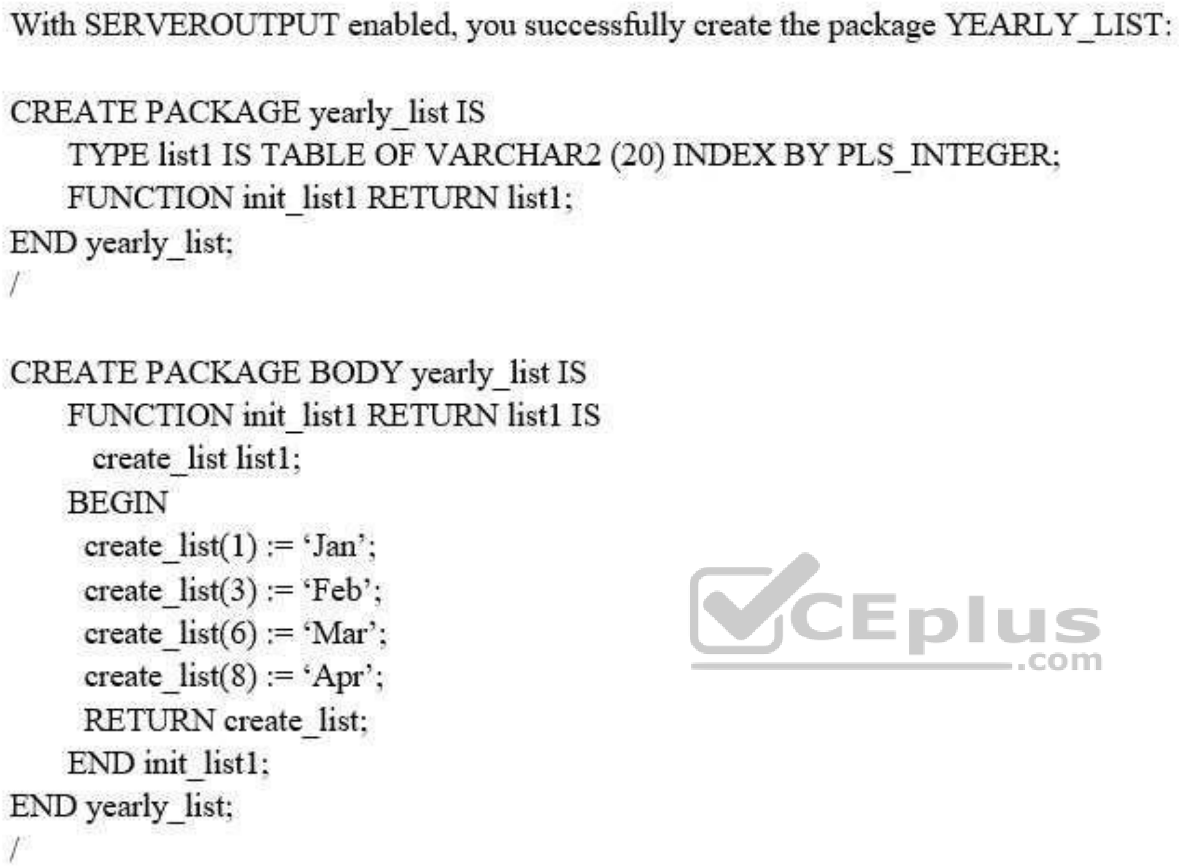
```
1  DECLARE
2      TYPE va$ IS VARRAY (200) OF NUMBER;
3      va va$ : =va$ ();
4  BEGIN
5      va.EXTEND (100);
6  END;
```

Which two will be correct after line 5?

A. va. LAST and va. LIMIT will return the same value.
B. va. LAST and va. COUNT will return the same value.
C. va. LIMIT and va. COUNT will return the same value.
D. va. LIMIT and va. NEXT (199) will return the same value.
E. va. LAST will return 200.
F. va. NEXT (199) will return NULL.

Correct Answer: AC
Section: (none)
Explanation
Explanation/Reference:

QUESTION 6

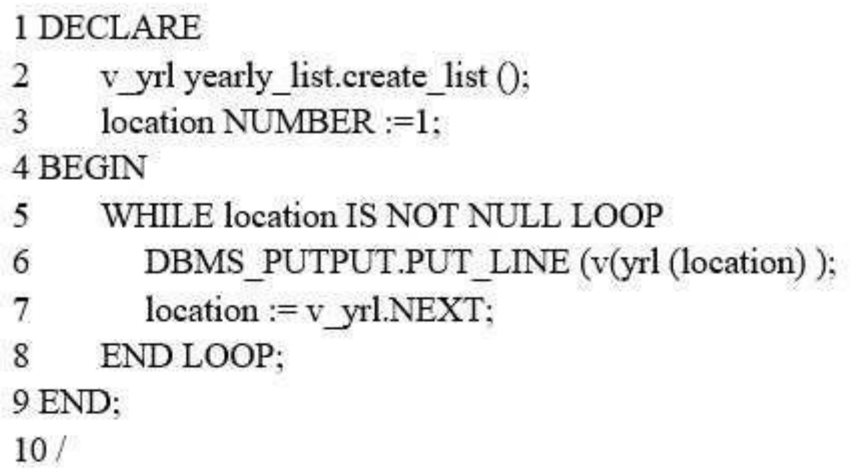


With SERVEROUTPUT enabled, you successfully create the package YEARLY_LIST:

```
CREATE PACKAGE yearly_list IS
     TYPE list1 IS TABLE OF VARCHAR2 (20) INDEX BY PLS_INTEGER;
     FUNCTION init_list1 RETURN list1;
END yearly_list;
/

CREATE PACKAGE BODY yearly_list IS
     FUNCTION init_list1 RETURN list1 IS
       create_list list1;
     BEGIN
      create_list(1) := ‘Jan’;
      create_list(3) := ‘Feb’;
      create_list(6) := ‘Mar’;
      create_list(8) := ‘Apr’;
      RETURN create_list;
     END init_list1;
END yearly_list;
/
```

Examine this code:



```
1 DECLARE
2       v_yrl yearly_list.create_list ();
3       location NUMBER :=1;
4 BEGIN
5       WHILE location IS NOT NULL LOOP
6           DBMS_PUTPUT.PUT_LINE (v(yrl (location) );
7           location := v_yrl.NEXT;
8       END LOOP;
9 END;
10 /
```

You want to display the contents of CREATE_LIST.

Which two lines need to be corrected in the PL/SQL block?

A. Line 2
B. Line 3
C. Line 5
D. Line 6
E. Line 7

Correct Answer: BD
Section: (none)
Explanation

Explanation/Reference:

QUESTION 7
Examine the following SQL statement: ALTER

SESSION SET PLSQL_OPTIMIZE_LEVEL=3;

What is the result of executing this statements?

A. The PL/SQL optimize level for some existing PL/SQL units will be changed as an immediate result.
B. The PL/SQL optimize level for subsequently complied PL/SQL units will be set to 3 and inlining will be enabled.
C. The PL/SQL optimize level for subsequently compiled PL/SQL units will be set to 3 and inlining will be disabled.
D. This statement will fail because PLSQL_OPTIMIZE_LEVEL can only be set at the system level,

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 8
Which two statements describe actions developers can take to make their application less vulnerable to security attacks?

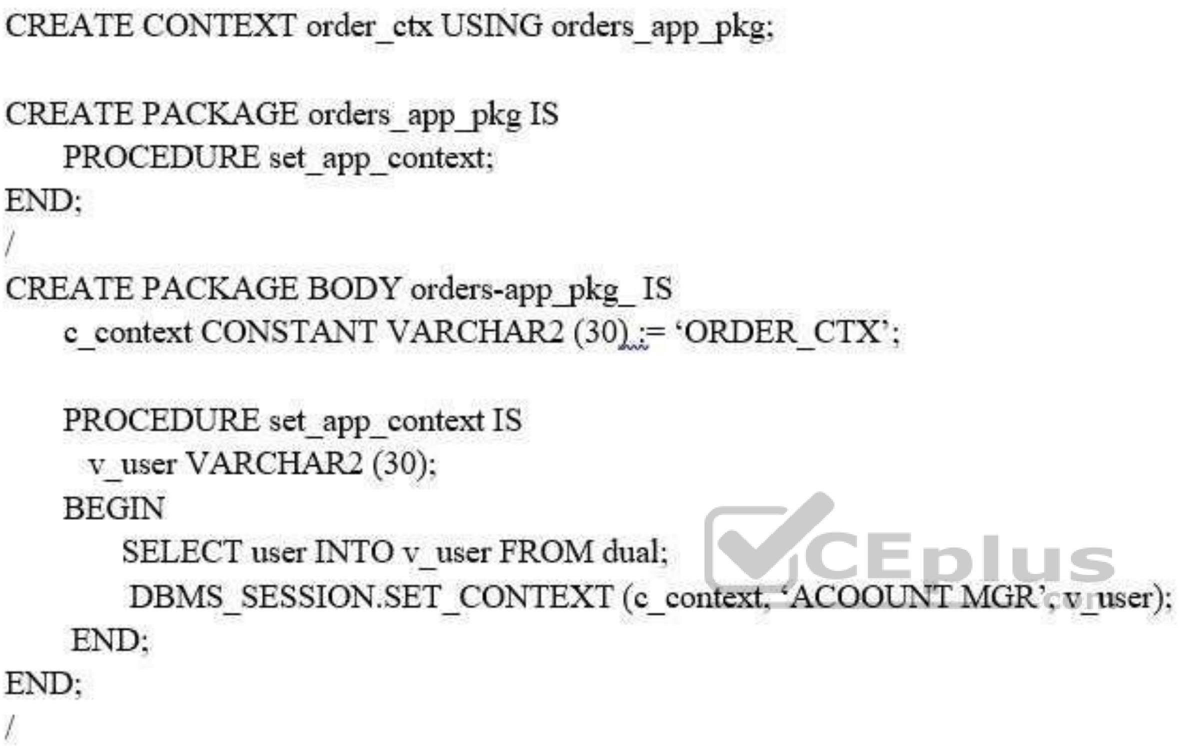
A. Include the AUTHID DEFINER clause in stored program units.
B. Do not concatenate unchecked user input into dynamically constructed SQL statements.
C. Switch from using DBMS_SQL to EXECUTE IMMEDIATE.
D. Include the AUTHID CURRENT_USER clause in stored program units.
E. Increase the amount of code that is accessible to users by default.

Correct Answer: BD
Section: (none)
Explanation

Explanation/Reference:

QUESTION 9

Examine this code:



```
CREATE CONTEXT order_ctx USING orders_app_pkg;

CREATE PACKAGE orders_app_pkg IS
     PROCEDURE set_app_context;
END;
/
CREATE PACKAGE BODY orders-app_pkg_ IS
     c_context CONSTANT VARCHAR2 (30) := ‘ORDER_CTX’;

     PROCEDURE set_app_context IS
       v_user VARCHAR2 (30);
     BEGIN
          SELECT user INTO v_user FROM dual;
           DBMS_SESSION.SET_CONTEXT (c_context, ‘ACOOUNT MGR’, v_user);
      END;
END;
/
```

What is the correct statement to get the value of attribute ACCOUNT_MGR after the procedure has been executed?

A. SELECT USERENV (‘ACCOUNT_MGR’) FROM dual;
B. SELECT SYS_CONTEXT (‘USERENV’, ‘ACCOUNT_MGR’) FROM dual;
C. SELECT SYS_CONTEXT (‘ORDER_CTX’, ‘ACCOUNT_MGR’) FROM dual;
D. SELECT SYS_CONTEXT (‘ACCOUNT_MGR’, ‘ORDER_CTX’) FROM dual;
E. SELECT USERENV (‘ORDER_CTX’) FROM dual;

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 10

Identify the two correct scenarios where a function can be optimized using the function result cache feature.

A. A function which inserts multiple records into a DEPARTMENTS table as part of one-time data setup for an HR application.
B. A function which accesses multiple tables and calculates the commission to be given to a sales representative based on the number of products sold by that representative.
C. A function which deletes all the records from an EMPLOYEES_AUDIT table based on their LOG_DATE.
D. A function which updates the SALARY of all the employees in an EMPLOYEES table by a fixed percentage based on their DESIGNATION.
E. A function which calculates the factorial of a given number without accessing any table.

Correct Answer: DE
Section: (none)
Explanation

Explanation/Reference:

QUESTION 11

Select the correct statement regarding BEQUEATH CURRENT_USER.

A. If a view references a PL/SQL function then BEQUEATH CURRENT_USER allows the function to execute with DBA privileges, regardless of the invoking user's privileges.
B. The BEQUEATH CURRENT_USER clause allows invoker's rights functions referenced in a view to execute with the privileges of the invoking user.
C. Any view calling a PL/SQL function with BEQUEATH CURRENT_USER in effect will execute with the privileges of the function owner.
D. With the BEQUEATH CURRENT_USER clause, a definer's rights function referenced in a view executes with the privileges of the view owner, not the function owner.

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

Reference: https://docs.oracle.com/database/121/DBSEG/dr_ir.htm#DBSEG558

QUESTION 12

Which tablespace is used to store the data collected by PL/Scope?

A. UNDOTBS1
B. SYSAUX C. SYSTEM
D. TEMP
E. USERS

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
Reference: https://docs.oracle.com/cd/B28359_01/appdev.111/b28424/adfns_plscope.htm#BABDGJAF

QUESTION 13
Which must be true in order to add RESULT_CACHE to a function header and have it compile successfully?

A. The IN parameters must not include BLOB, CLOB, collection or record data types.
B. The function must be created with invoker's rights or in an anonymous block.
C. The function must be declared as a pipelined table function.
D. The function must have an OUT or an IN OUT parameter.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Reference: https://docs.oracle.com/cd/E18283_01/appdev.112/e17126/subprograms.htm#insertedID11

QUESTION 14
Which two statements are true with respect to fine-grained access control?

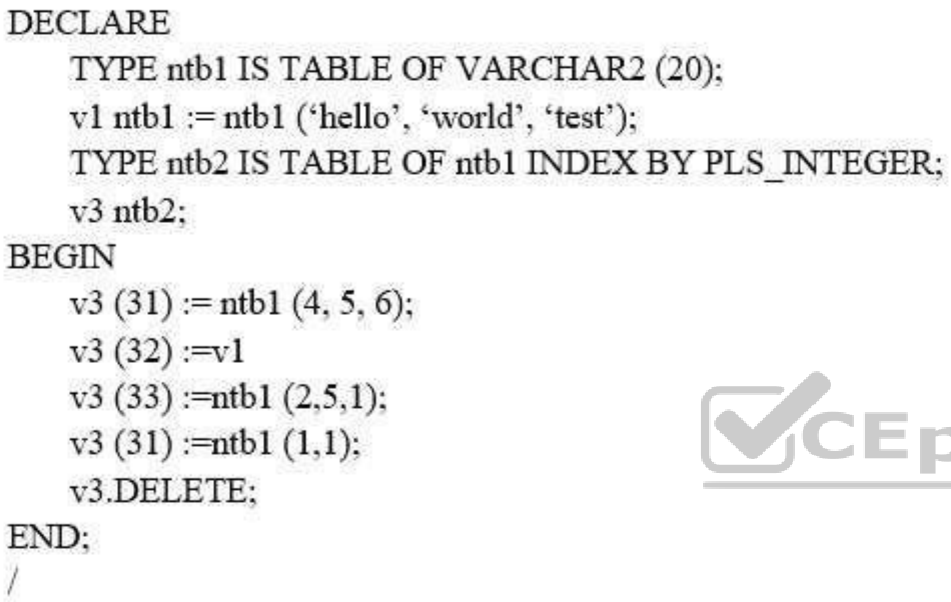
A. It is implemented by end users.
B. It can be used to implement column masking.
C. It implements security rules through functions and associates these security rules with tables, views or synonyms.
D. Separate policies are required for queries versus INSERT/UPDATE/DELETE statements.
E. The DBMS_FGA package is used to set up fine-grained access control.

Correct Answer: CD
Section: (none)

Explanation

Explanation/Reference:
Reference: https://docs.oracle.com/cd/B19306_01/server.102/b14220/security.htm

QUESTION 15



```
DECLARE
    TYPE ntb1 IS TABLE OF VARCHAR2 (20);
    v1 ntb1 := ntb1 (‘hello’, ‘world’, ‘test’);
    TYPE ntb2 IS TABLE OF ntb1 INDEX BY PLS_INTEGER;
    v3 ntb2;
BEGIN
    v3 (31) := ntb1 (4, 5, 6);
    v3 (32) :=v1
    v3 (33) :=ntb1 (2,5,1);
    v3 (31) :=ntb1 (1,1);
    v3.DELETE;
END;
/
```

Which two statements are correct about the collections before v3. DELETE is executed?

A. The values of v3(31) (2) and v3 (33) (2) are identical.
B. The value of v3 (31) (3) is 6.

C. The value of v3 (31) (1) and v3 (33) (3) are identical,
D. The value of v3 (31) (1) is “hello”.
E. The values of v3 (32) (2) and v1 (2) are identical.

Correct Answer: AD
Section: (none)
Explanation

Explanation/Reference:

QUESTION 16

Which two statements are true about the DBMS_ LOB package?

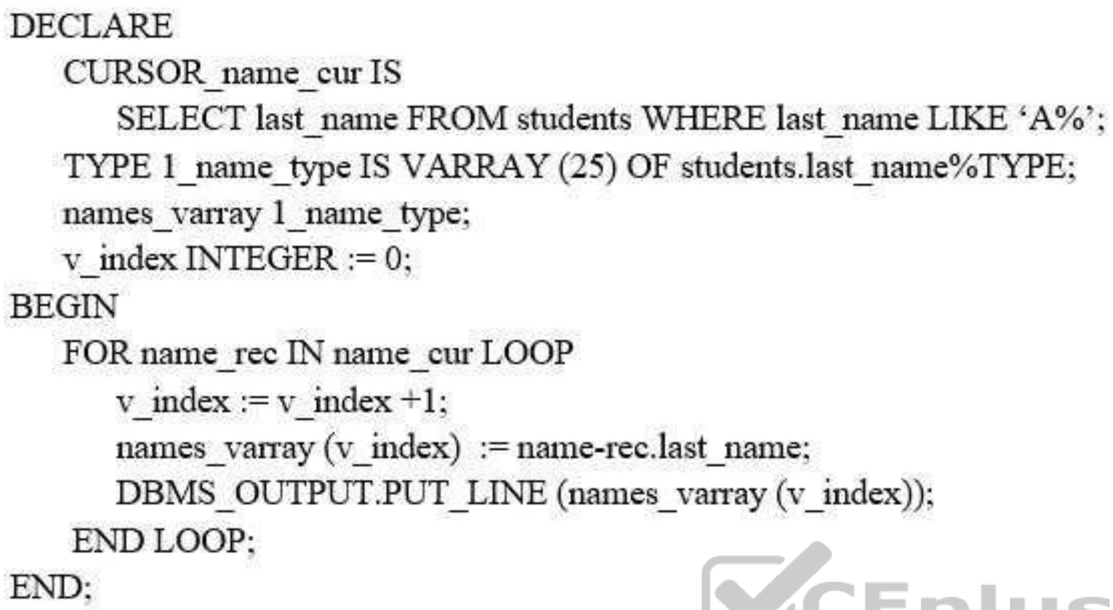
A. DBMS_LOB.COMPARE can compare parts of two LOBs.
B. DBMS_LOB.COMPARE returns the size difference of the compared LOBs.
C. DBMS_LOB.COMPARE is overloaded and can compare CLOBs with BLOBs.
D. If the destination LOB is a temporary LOB, the row must be locked before calling DBMS_LOB.CONVERTTOBLOB.
E. Before calling DBMS_LOB.CONVERTTOBLOB, both the source and destination LOB instances must exist.

Correct Answer: DE
Section: (none)
Explanation

Explanation/Reference:
Reference: https://docs.oracle.com/cd/E18283_01/appdev.112/e16760/d_lob.htm#insertedID2

QUESTION 17

The STUDENTS table with column LAST_NAME of data type VARCHAR2 exists in your database schema.
Examine this PL/SQL block:



```
DECLARE
    CURSOR name_cur IS
        SELECT last_name FROM students WHERE last_name LIKE 'A%';
    TYPE 1_name_type IS VARRAY (25) OF students.last_name%TYPE;
    names_varray 1_name_type;
    v_index INTEGER := 0;
BEGIN
    FOR name_rec IN name_cur LOOP
        v_index := v_index +1;
        names_varray (v_index)  := name-rec.last_name;
        DBMS_OUTPUT.PUT_LINE (names_varray (v_index));
     END LOOP;
END;
```

Which two actions must you perform for this PL/SQL block to execute successfully?

A. Replace the FOR loop with FOR name_rec IN names_varray.FIRST .. names_varray.LAST LOOP.
B. Replace the L_NAME_TYPE declaration with TYPE 1_name_type IS VARRAY (25) OF SYS_REFCURSOR;
C. Add name_rec name_cur%ROWTYPE; at the end of the DECLARE section.
D. Replace the NAMES_VARRAY declaration with names_varray 1_name_type := 1_name_type (); E. Replace the NAMES_VARRAY declaration with names_varray 1_name_type := null;

F. Add names_varray.EXTEND after the FOR …LOOP statement.

Correct Answer: EF
Section: (none)
Explanation

Explanation/Reference:

QUESTION 18

Which two blocks of code execute successfully?

```
A. DECLARE
   TYPE tab_type IS TABLE OF NUMBER;
   my_tab tab_type;
   BEGIN
   my_tab (1) :=1;
   END;
B. DECLARE
   TYPE tab_type IS TABLE OF NUMBER;
   my_tab tab_type := tab_type(2);
   BEGIN
   my_tab(1) :=55;
   END;
C. DECLARE
   TYPE tab_type IS TABLE OF NUMBER;
   my_tab tab_type;
   BEGIN
   my_tab. EXTEND (2);
   my_tab (1) := 55;
   END;
D. DECLARE
   TYPE tab_type IS TABLE OF NUMBER;
   my_tab tab_type;
   BEGIN
   my_tab := tab_type ();
   my_tab (1) := 55;
   END;
E. DECLARE
   TYPE tab_type IS TABLE OF NUMBER
   my_tab tab_type := tab_type (2, NULL, 50);
   BEGIN
   my_tab.EXTEND (3, 2);
   END;
```

Correct Answer: BD

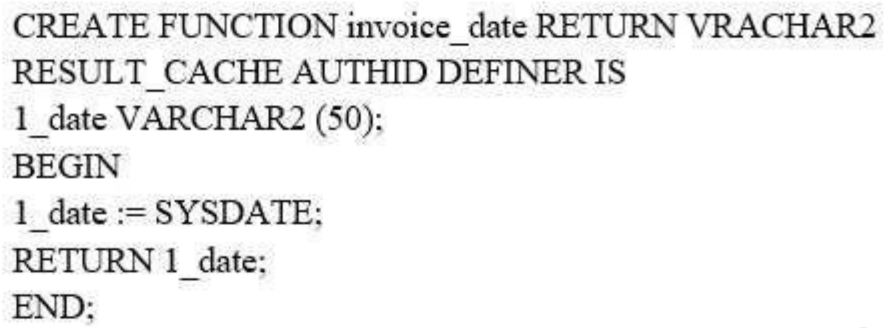
Section: (none)

Explanation

Explanation/Reference:

QUESTION 19

Examine this code:



```
CREATE FUNCTION invoice_date RETURN VRACHAR2
RESULT_CACHE AUTHID DEFINER IS
1_date VARCHAR2 (50);
BEGIN
1_date := SYSDATE;
RETURN 1_date;
END;
```

Users of this function may set different date formats in their sessions.

Which two modifications must be made to allow the use of your session’s date format when outputting the cached result of this function?

A. Change the RETURN type to DATE.
B. Change AUTHID to CURRENT_USER.
C. Use the TO_CHAR function around SYSDATE, that is, 1_date := TO_CHAR (SYSDATE).
D. Change the data type of 1_date to DATE.
E. Set NLS_DATE_FORMAT to ‘DD-MM-YY’ at the instance level.
F. Set the RESULT_CACHE_MODE parameter to FORCE.

Correct Answer: DF
Section: (none)
Explanation

Explanation/Reference:

QUESTION 20

Which statement is true about internal and external LOBs?

A. An external LOB can be loaded into an internal LOB variable using the DBMS_LOB package.
B. A NOEXIST_DIRECTORY exception can be raised when using internal and external LOBs.
C. Internal and external LOBs can be written using DBMS_LOB.
D. After an exception transfers program control outside a PL/SQL block, all references to open external LOBs are lost. E. When using DBMS_LOB.INSTR for internal and external LOBs, DBMS_LOB.OPEN should be called for each LOB.

Correct Answer: DE
Section: (none)
Explanation

Explanation/Reference:
Reference: https://docs.oracle.com/cd/E18283_01/appdev.112/e16760/d_lob.htm

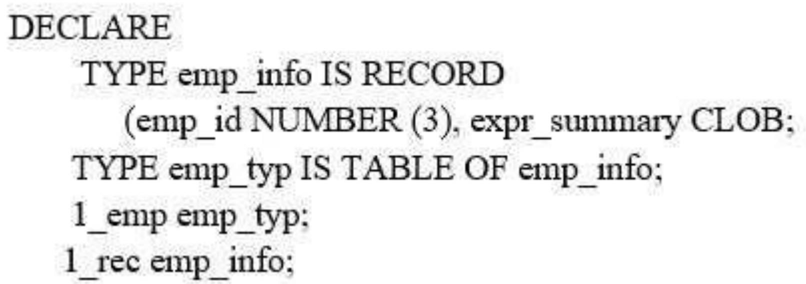
QUESTION 21
Which two statements about the PL/SQL hierarchical profiler are true?

A. Access it using the DBMS_PROFILER package.
B. Access it using the DBMS_HPROF package.
C. Profiler data is recorded in tables and published in HTML reports.
D. It is only accessible after a grant of the CREATE PROFILE privilege.
E. It helps you identify subprograms that are causing bottlenecks in application performance.

Correct Answer: BE
Section: (none)
Explanation

Explanation/Reference:
Reference: https://docs.oracle.com/cd/B28359_01/appdev.111/b28370/tuning.htm#LNPLS01214

QUESTION 22
Examine this declaration section:



```
DECLARE
     TYPE emp_info IS RECORD
        (emp_id NUMBER (3), expr_summary CLOB;
    TYPE emp_typ IS TABLE OF emp_info;
    1_emp emp_typ;
   1_rec emp_info;
```

Which two executable sections will display the message ‘Summary is null’?

A. BEGIN
1_rec := NULL;
1_emp := emp_typ (1_rec);
IF 1_emp (1).expr_summary IS EMPTY THEN
DBMS_OUTPUT.PUT_LINE (‘Summary is null’);
END IF;
END;

B. BEGIN
1_rec.emp_id :=1;
1_rec.expr_summary := NULL;
1_emp :=emp_typ (1_rec);
IF 1_emp(1).expr_summary IS NULL THEN
DBMS_OUTPUT.PUT_LINE (‘Summary is null’);
END IF;
END;

C. BEGIN
1_rec.emp_id :=1;
1_rec.expr_summary := EMPTY_CLOB ();
1_emp := emp_typ (1_rec);
IF 1_emp(1).expr_summary IS NULL THEN
DBMS_OUTPUT.PUT_LINE (‘Summary is null’);
END IF
END;

D. BEGIN
1_emp := emp_typ ();
IF NOT 1_emp. EXISTS (1) THEN

```
    DBMS_OUTPUT.PUT_LINE ('Summary is null');
    END IF
    END;
E.  BEGIN
    1_emp. EXTEND;
    IF NOT 1_emp. EXISTS (1) THEN
    DBMS_OUTPUT.PUT_LINE ('Summary is null');
    END IF
    END;
```

Correct Answer: DE

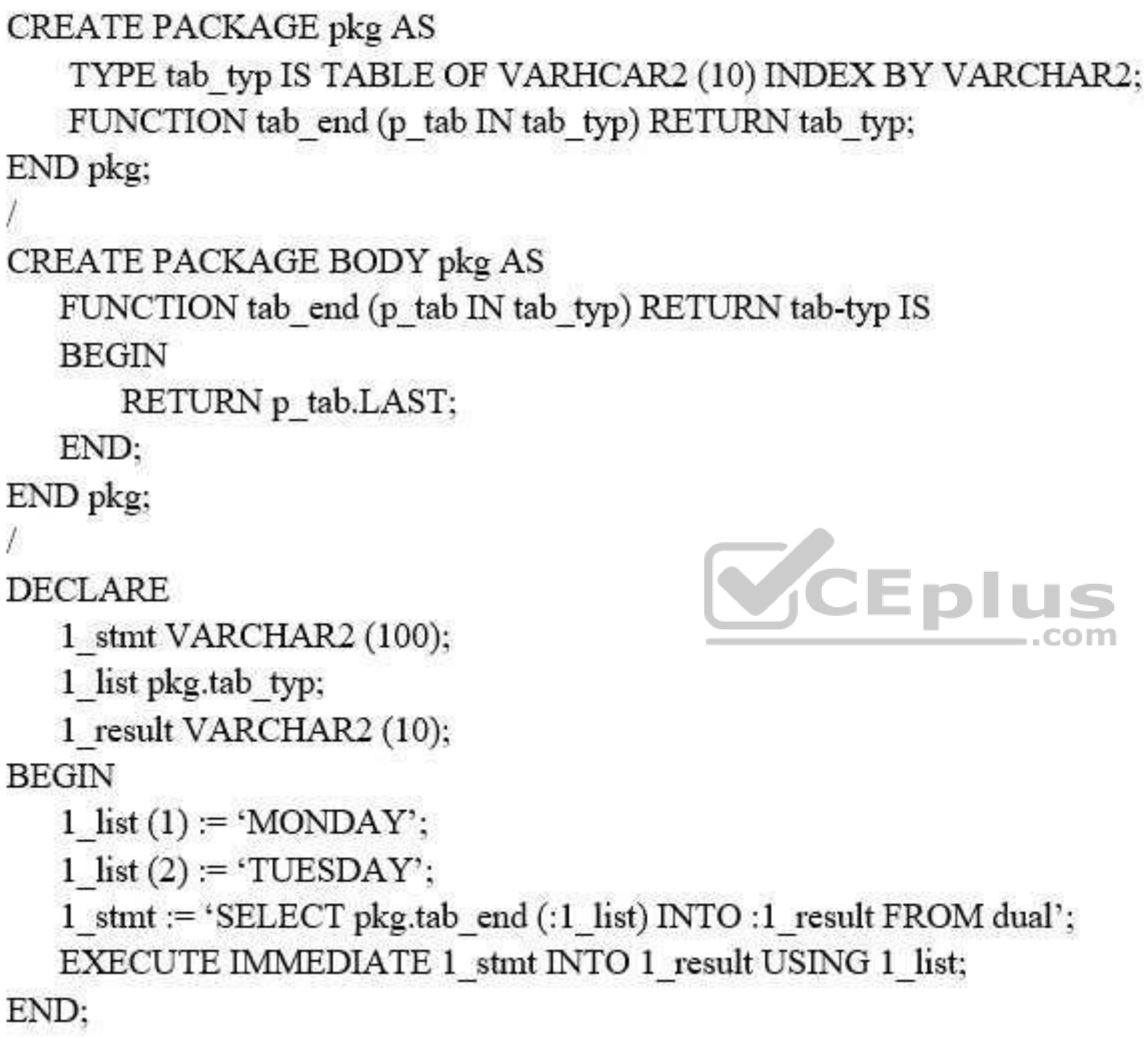
Section: (none)

Explanation

Explanation/Reference:

QUESTION 23

Examine this code:



```
CREATE PACKAGE pkg AS
     TYPE tab_typ IS TABLE OF VARHCAR2 (10) INDEX BY VARCHAR2;
     FUNCTION tab_end (p_tab IN tab_typ) RETURN tab_typ;
END pkg;
/
CREATE PACKAGE BODY pkg AS
    FUNCTION tab_end (p_tab IN tab_typ) RETURN tab-typ IS
    BEGIN
         RETURN p_tab.LAST;
    END;
END pkg;
/
DECLARE
    1_stmt VARCHAR2 (100);
    1_list pkg.tab_typ;
    1_result VARCHAR2 (10);
BEGIN
    1_list (1) := ‘MONDAY’;
    1_list (2) := ‘TUESDAY’;
    1_stmt := ‘SELECT pkg.tab_end (:1_list) INTO :1_result FROM dual’;
    EXECUTE IMMEDIATE 1_stmt INTO 1_result USING 1_list;
END;
```

Which two corrections must be applied for this anonymous block to execute successfully?

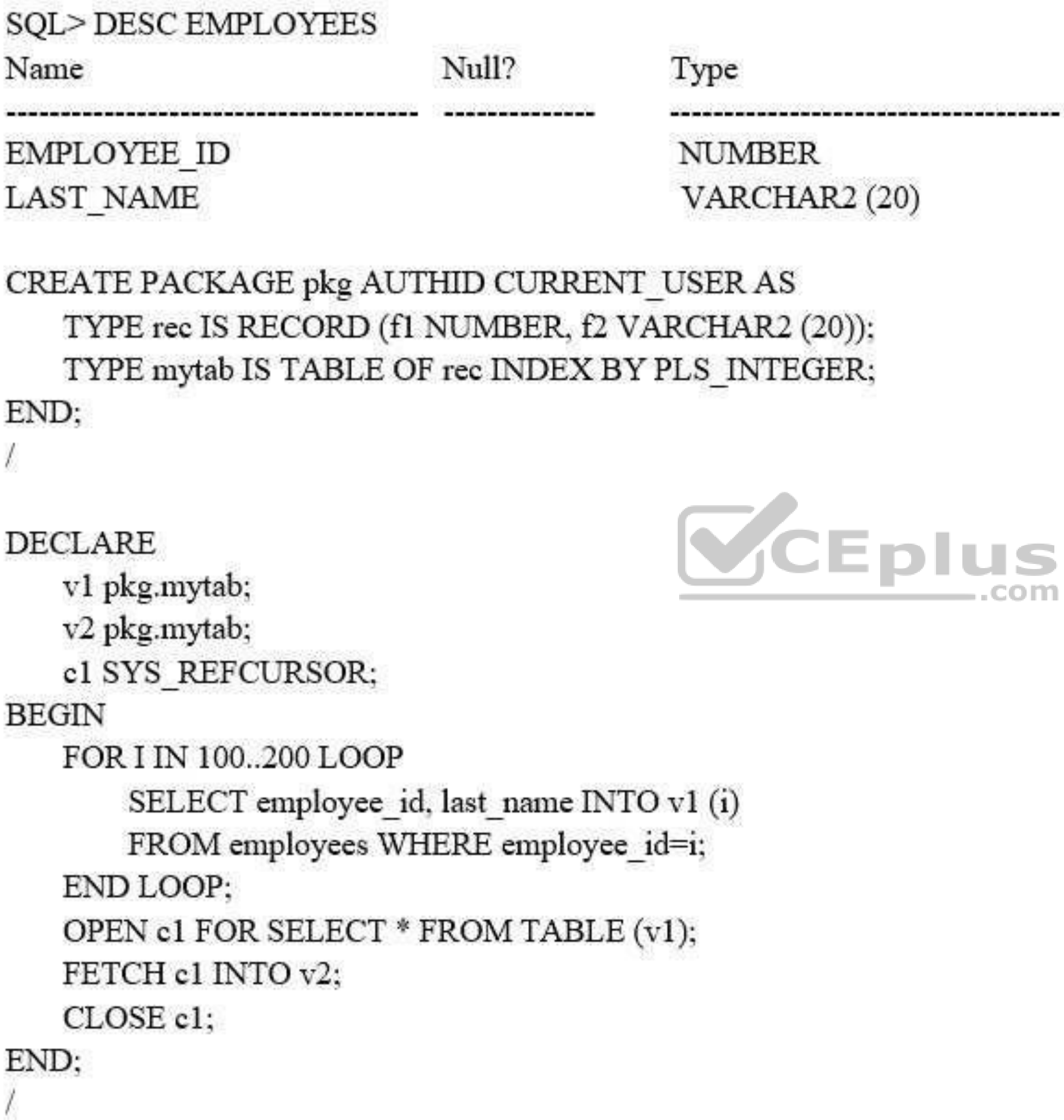
A. Change RETURN p_tab.LAST to RETURN p_tab.COUNT.
B. Declare the collection type inside the function.

C. Declare the collection type at the schema level instead of the package.
D. Define the function as stand-alone instead of in a package body.
E. Change the INDEX BY clause from VARCHAR2 to PLS_INTEGER.
F. Modify the function return type to return a scalar, VARCHAR2.

Correct Answer: DE
Section: (none)
Explanation

Explanation/Reference:

QUESTION 24
Examine this code:

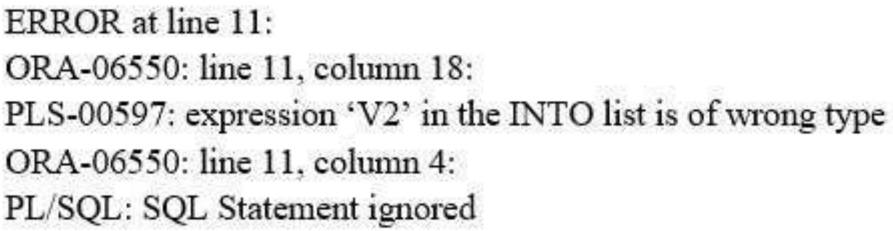


```
SQL> DESC EMPLOYEES
Name                                    Null?           Type
--------------------------------------- --------------  ------------------------------------
EMPLOYEE_ID                                              NUMBER
LAST_NAME                                                VARCHAR2 (20)

CREATE PACKAGE pkg AUTHID CURRENT_USER AS
    TYPE rec IS RECORD (f1 NUMBER, f2 VARCHAR2 (20));
    TYPE mytab IS TABLE OF rec INDEX BY PLS_INTEGER;
END;
/

DECLARE
    v1 pkg.mytab;
    v2 pkg.mytab;
    c1 SYS_REFCURSOR;
BEGIN
    FOR I IN 100..200 LOOP
        SELECT employee_id, last_name INTO v1 (i)
        FROM employees WHERE employee_id=i;
    END LOOP;
    OPEN c1 FOR SELECT * FROM TABLE (v1);
    FETCH c1 INTO v2;
    CLOSE c1;
END;
/
```

The anonymous block fails this error stack:



```
ERROR at line 11:
ORA-06550: line 11, column 18:
PLS-00597: expression 'V2' in the INTO list is of wrong type
ORA-06550: line 11, column 4:
PL/SQL: SQL Statement ignored
```

Which two changes, when separately applied, would prevent these errors from occurring?

A. Define v2 as employees%ROWTYPE.
B. Initialize v1 and v2 with appropriate constructor functions.
C. Define v2 as pkg. rec.
D. Nothing because using the function TABLE (V1) is prohibited.
E. Define v1 as employees%ROWTYPE.

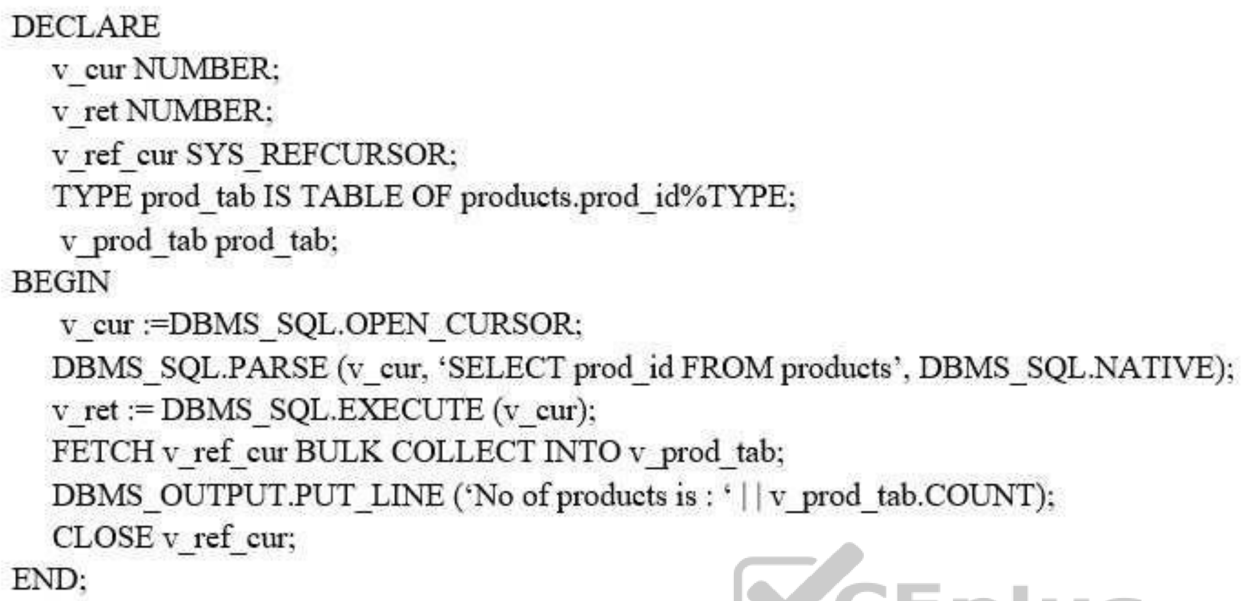
Correct Answer: BE
Section: (none)
Explanation

Explanation/Reference:

QUESTION 25

A products TABLE exists with a PROD_ID column.

Examine this PL/SQL block:



```
DECLARE
    v_cur NUMBER;
    v_ret NUMBER;
    v_ref_cur SYS_REFCURSOR;
    TYPE prod_tab IS TABLE OF products.prod_id%TYPE;
     v_prod_tab prod_tab;
BEGIN
     v_cur :=DBMS_SQL.OPEN_CURSOR;
    DBMS_SQL.PARSE (v_cur, ‘SELECT prod_id FROM products’, DBMS_SQL.NATIVE);
    v_ret := DBMS_SQL.EXECUTE (v_cur);
    FETCH v_ref_cur BULK COLLECT INTO v_prod_tab;
    DBMS_OUTPUT.PUT_LINE (‘No of products is : ‘ | | v_prod_tab.COUNT);
    CLOSE v_ref_cur;
END;
```

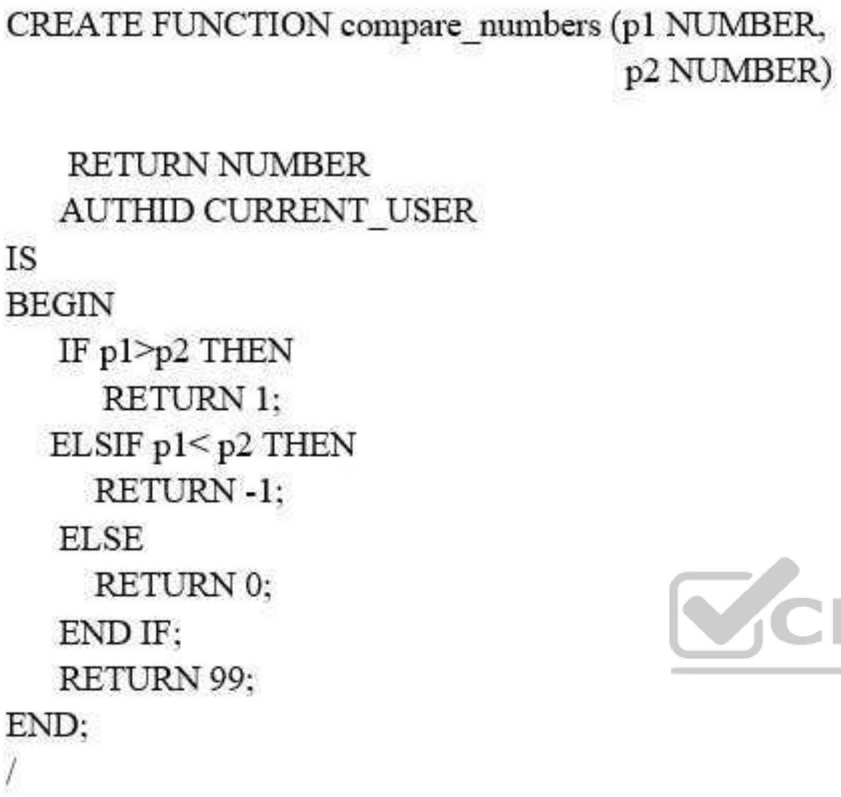
Which statement is true?

A. It executes successfully only if v_ref_cur := DBMS_SQL.TO_REFCURSOR (V_CUR); is added before the FETCH statement.
B. It executes successfully.
C. It executes successfully only if v_ref_cur: = DBMS_SQL.TO_CURSOR_NUMBER (v_cur); is added before the FETCH statement.
D. It executes successfully only if the FETCH statement is replaced by DBMS_SQL.RETURN_RESULT (v_ref_cur);
E. It executes successfully only if the FETCH statement is replaced by DBMS_SQL.FETCH_ROWS (v_cur);

Correct Answer: CD
Section: (none)
Explanation

Explanation/Reference:

QUESTION 26
Examine this PL/SQL function:



```
CREATE FUNCTION compare_numbers (p1 NUMBER,
                                  p2 NUMBER)

   RETURN NUMBER
   AUTHID CURRENT_USER
IS
BEGIN
   IF p1>p2 THEN
      RETURN 1;
  ELSIF p1< p2 THEN
      RETURN -1;
   ELSE
      RETURN 0;
   END IF;
   RETURN 99;
END;
/
```

What happens when the function is created with PLSQL_WARNINGS set to ‘ENABLE: ALL’?

A. There are no compilation warnings or errors.
B. It fails compilation.
C. An information compilation warning is generated.
D. A performance compilation warning is generated.
E. A severe compilation warning is generated.

Correct Answer: E
Section: (none)
Explanation

Explanation/Reference:

QUESTION 27

In your schema, the DEPARTMENTS table contains the columns DEPARTMENT_ID and DEPARTMENT_NAME.

You want to display the department name for existing department id 10.

With SERVEROUTPUT enabled, which two blocks of code will give the required output?

```
A. DECLARE
      TYPE dept_cur IS REF CURSOR;
      cv1 dept_cur;
      v_dept_name departments. department_name%TYPE;
      BEGIN
      OPEN cv1 FOR SELECT department_name FROM departments WHERE department_id=10;
      IF cv1 IS NOT NULL THEN
      FETCH cv1 INTO v_dept_name;
      DBMS_OUTPUT.PUT_LINE (v_dept_name);
      END IF
      CLOSE cv1;
      END;
B.  DECLARE
      TYPE dept_cur IS REF CURSOR RETURN departments%ROWTYPE;
      cv1 dept_cur;
      v_dept_name departments.department_name%TYPE;
      BEGIN
      OPEN cv1 FOR SELECT * FROM departments WHERE department_id=10;
      FETCH cv1. department_name INTO v_dept_name;
      DBMS_OUTPUT.PUT_LINE (v_dept_name);
      CLOSE cv1;
      END;
C.  DECLARE
      TYPE names_t IS TABLE OF SYS_REFCURSOR INDEX BY PLS_INTEGER;
      cv1 names_t;
      v_dept_name departments.department_name%TYPE;
      BEGIN
      OPEN cv1 FOR SELECT department_name FROM departments WHERE department_id=10;
      FETCH cv1 INTO v_dept_name;
```

```
DBMS_OUTPUT.PUT_LINE (v_dept_name);
CLOSE cv1;
END;
```

D. DECLAREcv1 SYS_REFCURSOR;

```
v_dept_name departments.department_name%TYPE;
BEGIN
EXECUTE IMMEDIATE ‘BEGIN OPEN: cv1 FOR
SELECT department_name FROM departmnets WHERE department_id=10: END;’
USING IN cv1;
FETCH cv1 INTO v_dept_name;
DBMS_OUTPUT.PUT_LINE (v_dept_name);
CLOSE cv1;
END;
```

Correct Answer: CD
Section: (none)
Explanation

Explanation/Reference:

QUESTION 28

Which two statements are correct for collecting data about identifiers in PL/SQL source code?

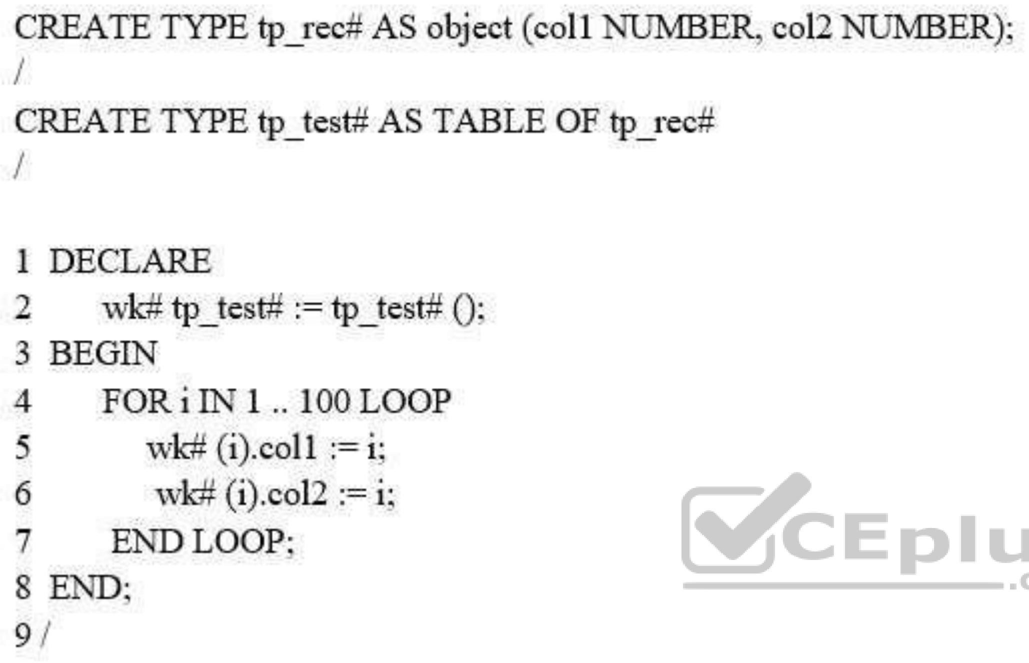
A. CREATE < function/Procedure> PLSCOPE_SETTINGS = ‘IDENTIFIERS: ALL’ AS …
B. ALTER SYSTEM SET PLSCOPE_SETTINGS = ‘IDENTIFIERS: NONE’
C. ALTER SESSION SET PLSCOPE_SETTINGS = ‘IDENTIFIERS: NONE’
D. ALTER SESSION SET PLSCOPE_SETTINGS = ‘IDENTIFIERS: ALL’
E. ALTER <function/Procedure> COMPILE PLSCOPE_SETTINGS = ‘IDENTIFIERS: ALL’

Correct Answer: AD
Section: (none)
Explanation

Explanation/Reference:

QUESTION 29

Examine these statements:



```
CREATE TYPE tp_rec# AS object (col1 NUMBER, col2 NUMBER);
/
CREATE TYPE tp_test# AS TABLE OF tp_rec#
/

1  DECLARE
2       wk# tp_test# := tp_test# ();
3  BEGIN
4       FOR i IN 1 .. 100 LOOP
5           wk# (i).col1 := i;
6            wk# (i).col2 := i;
7        END LOOP;
8  END;
9 /
```

Which two corrections will allow this anonymous block to execute successfully?

A. Add wk# .NEXT; before the 7th line.

B. Add i PLS_INTEGER; before the 3rd line.

C. Add wk#. EXTEND (1); before the 5th line.

D. Change line #2 to wk# tp_test# := tp_test# (tp_rec# ());

E. Replace lines 5 and 6 with wk# (i) := tp_rec# (i, i);

Correct Answer: CE

Section: (none)

Explanation

Explanation/Reference:

QUESTION 30

Select a valid reason for using VARRAYS.

A. When the amount of data to be held in the collection is widely variable.

B. As a column in a table when you want to retrieve the collection data for certain rows by ranges of values.

C. When you want to delete elements from the middle of the collection.

D. As a column in a table when you want to store no more than 10 elements in each row's collection.

Correct Answer: B

Section: (none)

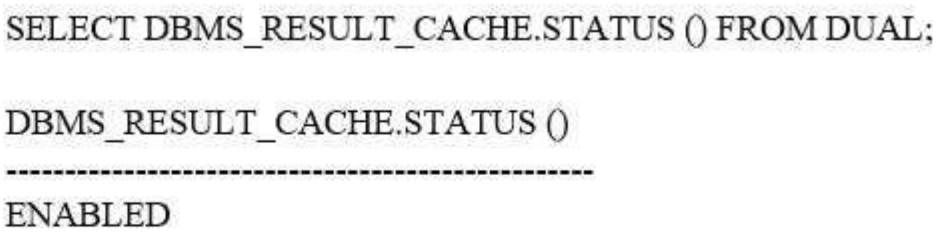
Explanation

Explanation/Reference:

Reference https://www.go4expert.com/articles/oracle-nested-tables-varrays-t20494/

QUESTION 31

Examine this query executed as SYS and its output:



```
SELECT DBMS_RESULT_CACHE.STATUS () FROM DUAL;

DBMS_RESULT_CACHE.STATUS ()
---------------------------------------------------
ENABLED
```

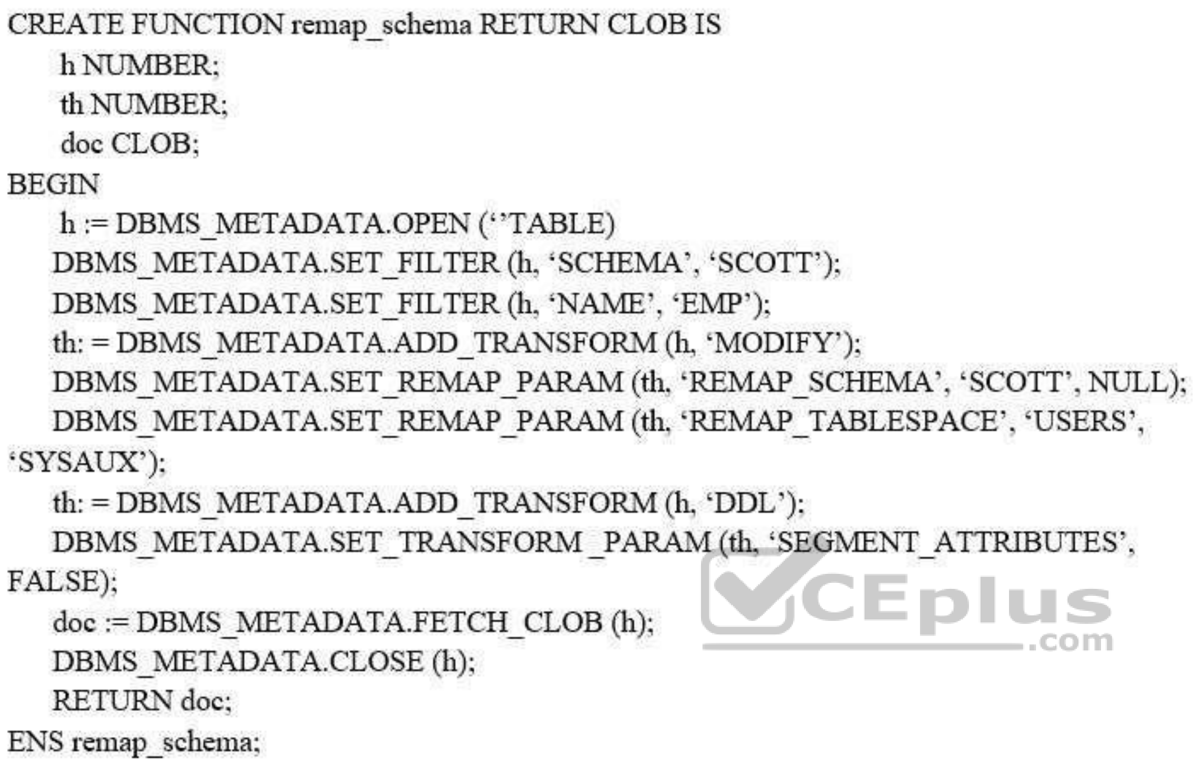
Which two observations are true based on the output?

A. The client-side result cache and the server-side result cache are enabled.
B. All distinct query results are cached for the duration of a SYS user session.
C. Repetitive SQL queries and PL/SQL function results are cached and automatically used from the cache across all SYS user sessions.
D. The result cache exists but which SQL queries are cached depends on the value of the RESULT_CACHE_MODE parameter.
E. Repetitive SQL queries executed on permanent non-dictionary objects may have faster response times.

Correct Answer: CD
Section: (none)
Explanation

Explanation/Reference:

QUESTION 32
Examine this function:



```
CREATE FUNCTION remap_schema RETURN CLOB IS
      h NUMBER;
      th NUMBER;
      doc CLOB;
BEGIN
      h := DBMS_METADATA.OPEN (‘’TABLE)
     DBMS_METADATA.SET_FILTER (h, ‘SCHEMA’, ‘SCOTT’);
     DBMS_METADATA.SET_FILTER (h, ‘NAME’, ‘EMP’);
     th: = DBMS_METADATA.ADD_TRANSFORM (h, ‘MODIFY’);
     DBMS_METADATA.SET_REMAP_PARAM (th, ‘REMAP_SCHEMA’, ‘SCOTT’, NULL);
     DBMS_METADATA.SET_REMAP_PARAM (th, ‘REMAP_TABLESPACE’, ‘USERS’,
‘SYSAUX’);
     th: = DBMS_METADATA.ADD_TRANSFORM (h, ‘DDL’);
     DBMS_METADATA.SET_TRANSFORM _PARAM (th, ‘SEGMENT_ATTRIBUTES’,
FALSE);
     doc := DBMS_METADATA.FETCH_CLOB (h);
     DBMS_METADATA.CLOSE (h);
     RETURN doc;
ENS remap_schema;
```

Execute the query:
SELECT remap_schema FROM dual;

Which is the correct output from the query?

A. CREATE TABLE “EMP” (“EMPNO” NUMBER (4,0), “ENAME” VARCHAR2 (10), “JOB” VARCHAR2 (9), “MGR” NUMBER (4,0), “HIREDATE”
DATE, “SAL” NUMBER (7,2) , “COMM” NUMBER (7,2), “DEPTNO” NUMBER (2,0),
CONSTRAINT “PK_EMP” PRIMARY KEY (“EMPNO”)
USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
STORAGE (INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2417483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CHACHE DEFAULT CELL_FLASH_CACHE DEFAULT)

```
TABLESPACE “USERS” ENABLE,
CONSTRAINT “FK_DEPTNO” FOREIGN KEY (“DEPTNO”)
REFERENCES “DEPT” (“DEPTNO”) ENABLE
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE (INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE “USERS”
```

B.

```
CREATE TABLE “EMP” (“EMPNO” NUMBER (4, 0), “ENAME” VARCHAR2 (10), “JOB” VARCHAR2 (9), “MGR” NUMBER (4, 0), “HIREDATE” DATE, “SAL”
NUMBER (7, 2), “COMM” NUMBER (7, 2), “DEPTNO” NUMBER (2, 0),
CONSTRAINT “PK_EMP” PRIMARY KEY (“EMPNO”)
USING INDEX ENABLE,
CONSTRAINT “FK_DEPTNO” FOREIGN KEY (“DEPTNO”)
REFERENCES “DEPT” (“DEPTNO”) ENABLE)
```

C.

```
CREATE TABLE “SCOTT”. “EMP” (“EMPNO” NUMBER (4, 0), “ENAME” VARCHAR2 (10), “JOB” VARCHAR2 (9), “MGR” NUMBER (4, 0), “HIREDATE”
DATE, “SAL” NUMBER (7, 2), “COMM” NUMBER (7, 2), “DEPTNO” NUMBER (2, 0),
CONSTRAINT “PK_EMP” PRIMARY KEY (“EMPNO”)
USING INDEX ENABLE,
CONSTRAINT “FK_DEPTNO” FOREIGN KEY (“DEPTNO”)
REFERENCES “DEPT” (“DEPTNO”) ENABLE)
```

D.

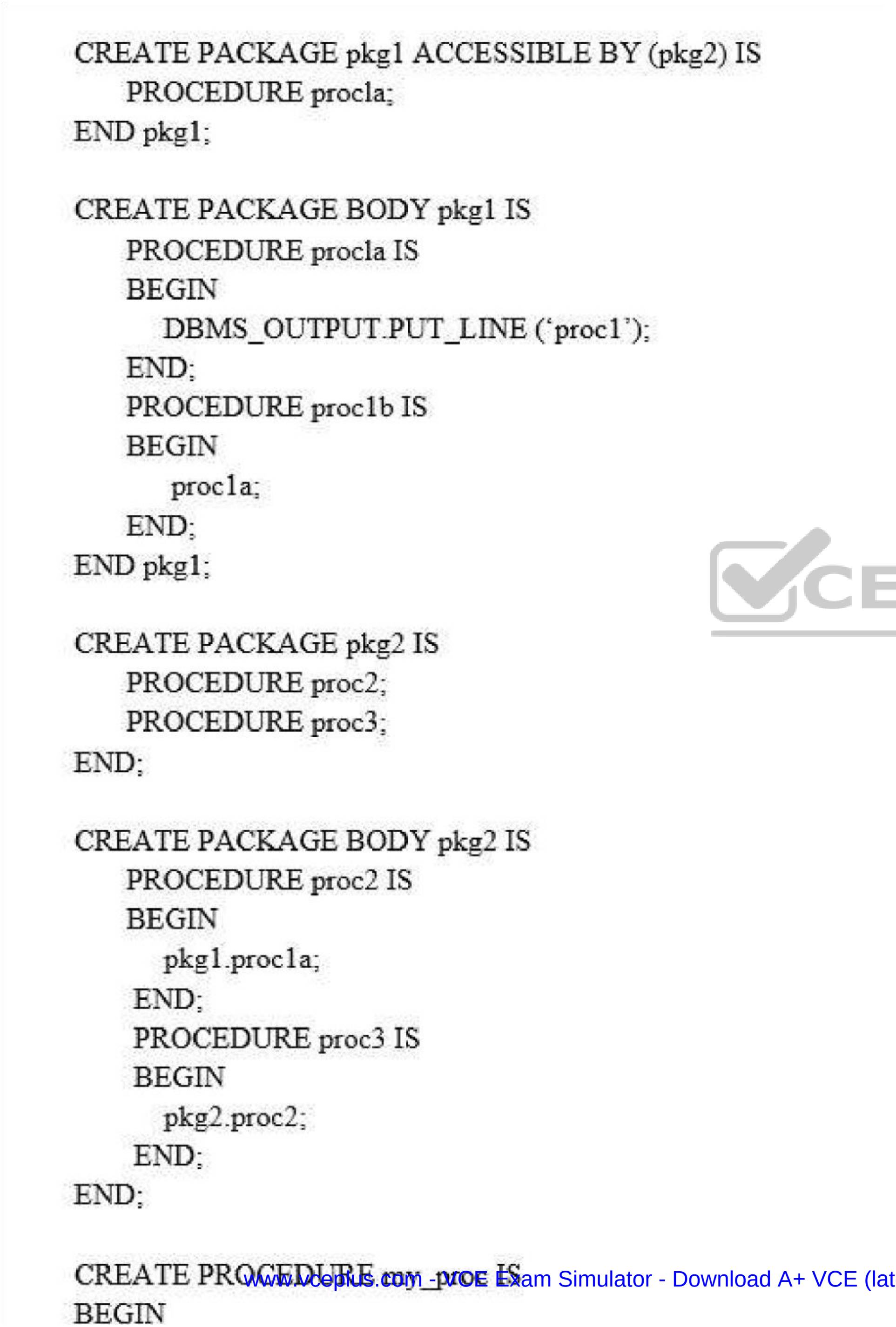
```
CREATE TABLE “EMP” (“EMPNO” NUMBER (4,0), “ENAME” VARCHAR2 (10), “JOB” VARCHAR2 (9), “MGR” NUMBER (4,0), “HIREDATE” DATE, “SAL”
NUMBER (7, 2) , “COMM” NUMBER (7, 2), “DEPTNO” NUMBER (2,0),
CONSTRAINT “PK_EMP” PRIMARY KEY (“EMPNO”)
USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
STORAGE (INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2417483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CHACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE “SYSAUX” ENABLE,
CONSTRAINT “FK_DEPTNO” FOREIGN KEY (“DEPTNO”)
REFERENCES “DEPT” (“DEPTNO”) ENABLE
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE (INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
```

BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE “SYSAUX”

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 33
Examine these program units:



```
CREATE PACKAGE pkg1 ACCESSIBLE BY (pkg2) IS
      PROCEDURE procla;
END pkg1;

CREATE PACKAGE BODY pkg1 IS
      PROCEDURE procla IS
      BEGIN
          DBMS_OUTPUT.PUT_LINE ('proc1');
      END;
      PROCEDURE proc1b IS
      BEGIN
           proc1a;
      END;
END pkg1;

CREATE PACKAGE pkg2 IS
      PROCEDURE proc2;
      PROCEDURE proc3;
END;

CREATE PACKAGE BODY pkg2 IS
      PROCEDURE proc2 IS
      BEGIN
          pkg1.proc1a;
       END;
       PROCEDURE proc3 IS
       BEGIN
          pkg2.proc2;
       END;
END;

CREATE PROCEDURE my_proc IS
BEGIN
```

Which two blocks will execute successfully?

A. BEGIN
My_proc;
END;

B. BEGIN
pkg2.proc3;
END;

C. BEGIN
pkg2.proc2;
END;

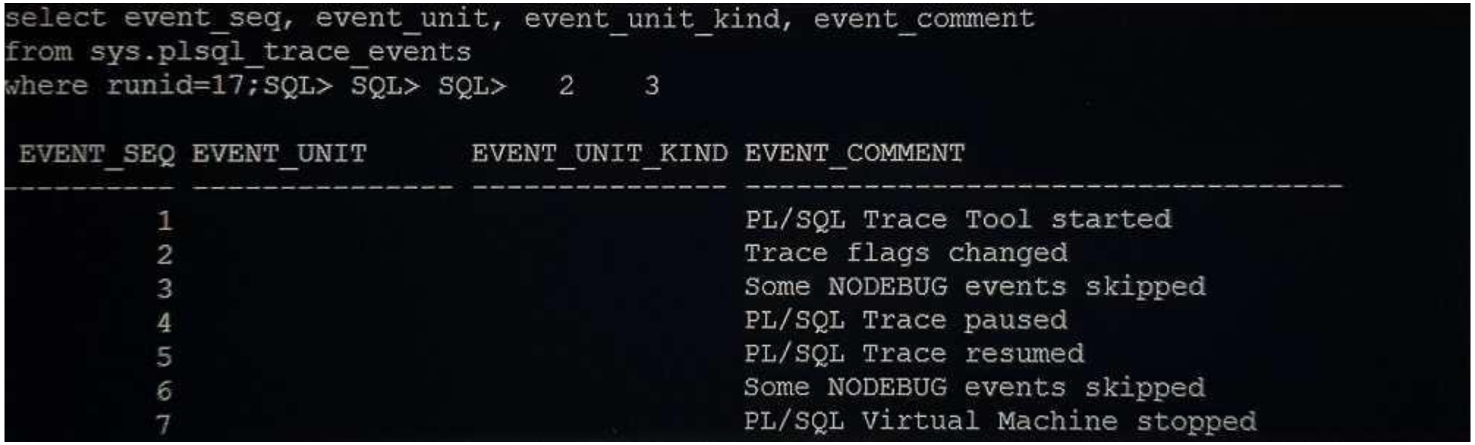
D. BEGIN
pkg1.proc1a;
END;

E. BEGIN
pkg1.proc1b;
END;

Correct Answer: BD
Section: (none)
Explanation

Explanation/Reference:

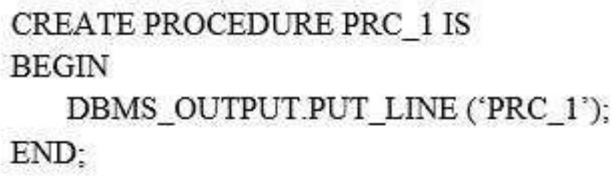
QUESTION 34
Refer to the Exhibit.



```
select event_seq, event_unit, event_unit_kind, event_comment
from sys.plsql_trace_events
where runid=17;SQL> SQL> SQL>   2    3

 EVENT_SEQ EVENT_UNIT        EVENT_UNIT_KIND EVENT_COMMENT
---------- ---------------- --------------- -----------------------------------
         1                                  PL/SQL Trace Tool started
         2                                  Trace flags changed
         3                                  Some NODEBUG events skipped
         4                                  PL/SQL Trace paused
         5                                  PL/SQL Trace resumed
         6                                  Some NODEBUG events skipped
         7                                  PL/SQL Virtual Machine stopped
```

Examine this procedure created in a session where PLSQL_OPTIMIZE_LEVEL =2:



```
CREATE PROCEDURE PRC_1 IS
BEGIN
    DBMS_OUTPUT.PUT_LINE ('PRC_1');
END;
```

PL/SQL tracing in enabled in a user session using this command:

EXEC DBMS_TRACE.SET_PLSQL_TRACE (DBMS_TRACE.TRACE_ENABLED_LINES)

The procedure is executed using this command:

EXEC PRC_1

Examine the exhibit for the content of the PLSQL_TRACE_EVENTS table.

Why is tracing excluded from the PLSQL_TRACE_EVENTS table?

A. DBMS_TRACE.TRACE_ENABLED_LINES traces only exceptions in subprograms.
B. PRC_1 is not compiled with debugging information.
C. Tracing is not enabled with the TRACE_ENABLED_CALLS option.

D. PRC_1 is compiled with the default AUTHID DEFINER clause.
E. Tracing will be enabled only for the second execution of PRC_1.

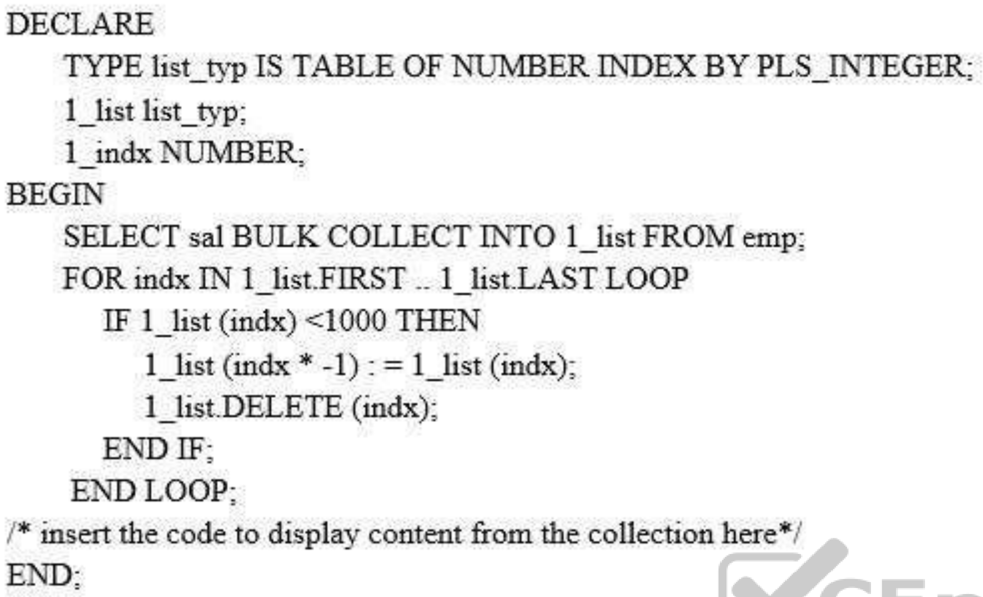
Correct Answer: BE
Section: (none)
Explanation

Explanation/Reference:

QUESTION 35
Examine the structure of the EMP table:

```
Name                 Null?          Type
-------------------  -------------- ----------------------
EMPNO                NOT NULL       NUMBER (4)
ENAME                               VARCHAR2 (10)
SAL                                 NUMBER (7, 2)
```

Examine this code:



```
DECLARE
    TYPE list_typ IS TABLE OF NUMBER INDEX BY PLS_INTEGER;
    1_list list_typ;
    1_indx NUMBER;
BEGIN
    SELECT sal BULK COLLECT INTO 1_list FROM emp;
    FOR indx IN 1_list.FIRST .. 1_list.LAST LOOP
        IF 1_list (indx) <1000 THEN
            1_list (indx * -1) : = 1_list (indx);
            1_list.DELETE (indx);
        END IF;
     END LOOP;
/* insert the code to display content from the collection here*/
END;
```

Which code should be inserted to display the collection contents?

A. 1_indx := 1_list.FIRST;
WHILE (1_indx IS NOT NULL) LOOP
DBMS_OUTPUT.PUT_LINE (1_indx || ‘ ’ || 1_list (1_indx));
1_indx := 1_emp.NEXT (1_indx);
END LOOP;

B. FOR indx IN 1_list. COUNT .. -1 LOOP
DBMS_OUTPUT.PUT_LINE (indx | | ‘ ’ | | 1_list (indx));
END LOOP;

C. FOR indx IN -1 .. 1_list.LIMIT LOOP
DBMS_OUTPUT.PUT_LINE (indx | | ‘ ’ | | 1_list (indx));
END LOOP;

D. FOR indx IN 1_list.FIRST . . 1_list.LAST LOOP
DBMS_OUTPUT.PUT_LINE (indx | | ‘ ’ | | 1_list (indx));
END LOOP;

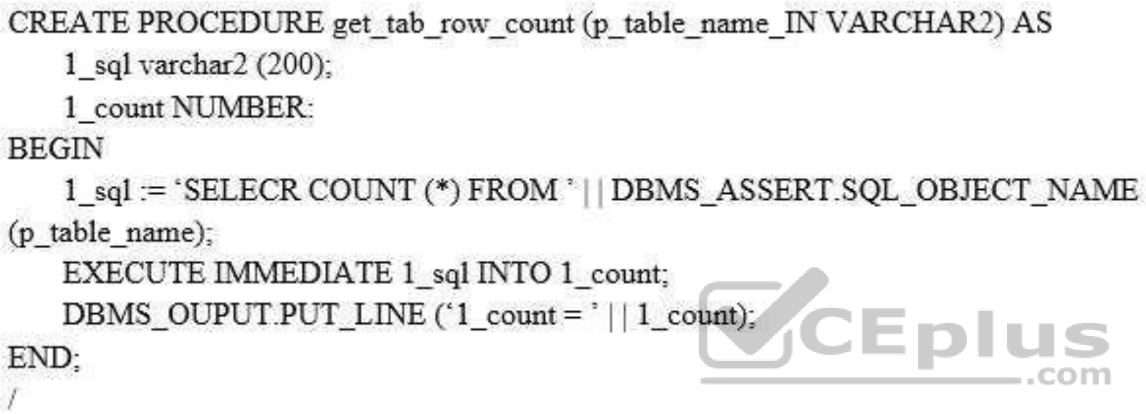
Correct Answer: C

Section: (none)
Explanation

Explanation/Reference:

QUESTION 36

You are logged on to the SCOTT schema and the schema has EMP and DEPT tables already created:
Examine this PL/SQL procedure:



```
CREATE PROCEDURE get_tab_row_count (p_table_name_IN VARCHAR2) AS
        1_sql varchar2 (200);
        1_count NUMBER:
BEGIN
        1_sql := 'SELECR COUNT (*) FROM ' || DBMS_ASSERT.SQL_OBJECT_NAME
(p_table_name);
        EXECUTE IMMEDIATE 1_sql INTO 1_count;
        DBMS_OUPUT.PUT_LINE ('1_count = ' || 1_count);
END;
/
```

Which PL/SQL block will raise an exception?

A. EXEC get_tab_row_count (‘emp’);
B. EXEC get_tab_row_count (‘SCOTT.EMP’);
C. EXEC get_tab_row_count (‘ “EMP” ’);
D. EXEC get_tab_row_count (‘DEPT’);
E. EXEC get_tab_row_count (‘DEPT, EMP’)

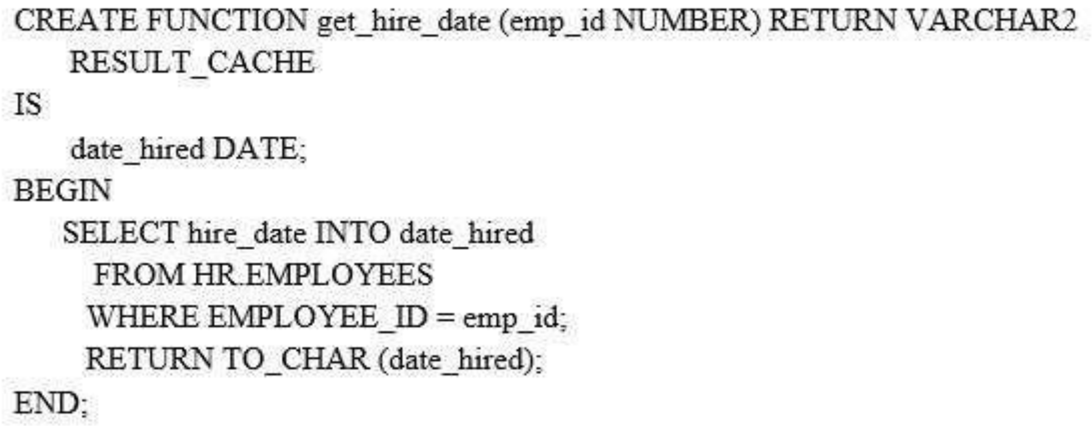
Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 37

This result cache is enabled for the database instance.

Examine this code for a PL/SQL function:



```
CREATE FUNCTION get_hire_date (emp_id NUMBER) RETURN VARCHAR2
      RESULT_CACHE
IS
      date_hired DATE;
BEGIN
     SELECT hire_date INTO date_hired
       FROM HR.EMPLOYEES
      WHERE EMPLOYEE_ID = emp_id;
      RETURN TO_CHAR (date_hired);
END;
```

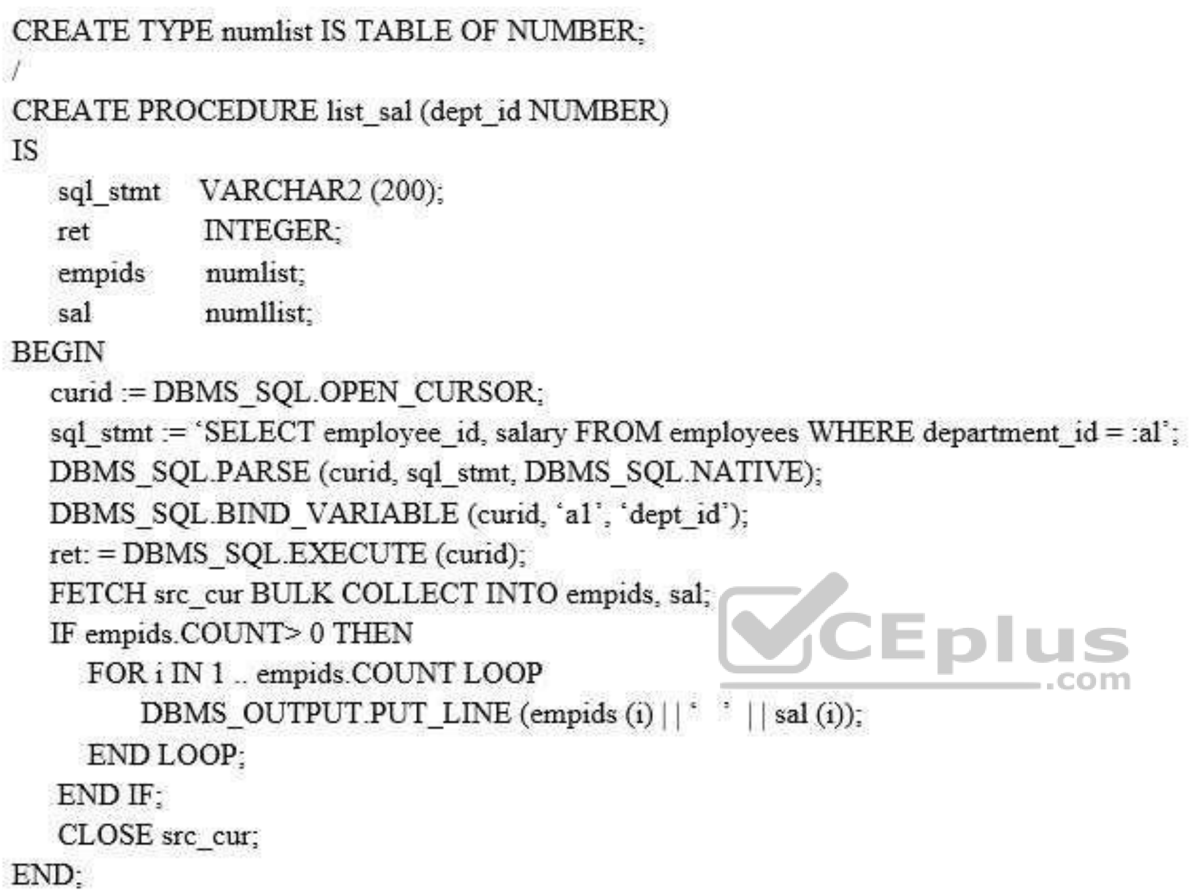
Which two actions would ensure that the same result will be consistently returned for any session when the same input value is passed to the function?

A. Add a parameter, fmt, and change the RETURN statement to: RETURN TO_CHAR (date_hired, fmt);
B. Set the RESULT_CACHE_MODE parameter to FORCE.
C. Increase the value for the RESULT_CACHE_MAX_SIZE parameter.
D. Change the return type of GET_HIRE_DATE to DATE and have each session invoke the TO_CHAR function.
E. Set the RESULT_CACHE_MAX_RESULT parameter to 0.

Correct Answer: BD
Section: (none)
Explanation

Explanation/Reference:

QUESTION 38
Examine the incomplete code:



```
CREATE TYPE numlist IS TABLE OF NUMBER;
/
CREATE PROCEDURE list_sal (dept_id NUMBER)
IS
     sql_stmt    VARCHAR2 (200);
     ret         INTEGER;
     empids      numlist;
     sal         numllist;
BEGIN
    curid := DBMS_SQL.OPEN_CURSOR;
    sql_stmt := ‘SELECT employee_id, salary FROM employees WHERE department_id = :al’;
    DBMS_SQL.PARSE (curid, sql_stmt, DBMS_SQL.NATIVE);
    DBMS_SQL.BIND_VARIABLE (curid, ‘a1’, ‘dept_id’);
    ret: = DBMS_SQL.EXECUTE (curid);
    FETCH src_cur BULK COLLECT INTO empids, sal;
    IF empids.COUNT> 0 THEN
       FOR i IN 1 .. empids.COUNT LOOP
            DBMS_OUTPUT.PUT_LINE (empids (i) || ‘  ’ || sal (i));
       END LOOP;
     END IF;
     CLOSE src_cur;
END;
```

Which three lines of code must be added for it to successfully compile?

A. curid := DBMS_SQL.TO_CURSOR_NUMBER (src_cur);
B. src_cur := DBMS_SQL.TO_REFCURSOR (curid);
C. src_cur= NUMBER;
D. curid NUMBER;
E. curid SYS_FEFCURSOR;
F. src_cur SYS_REFCURSOR;

Correct Answer: BCF

Section: (none)
Explanation

Explanation/Reference:

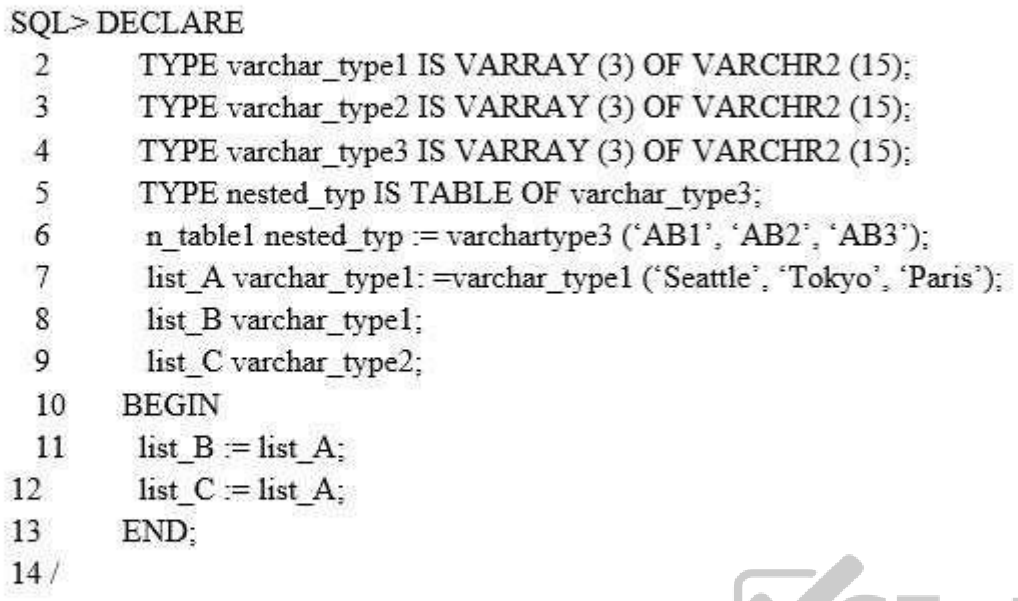
QUESTION 39
Which two statements are correct in Oracle Database 12c?

A. For native compilation, PLSQL_OPTIMIZE_LEWVEL should be set to 2.
B. Native compilation is the default compilation method
C. Native compilation should be used during development.
D. Natively compiles code is stored in the SYSTEM tablespace.
E. To change a PL/SQL object from interpreted to native code, set the PLSQL_CODE_TYPE to NATIVE and recompile it.

Correct Answer: DE
Section: (none)
Explanation

Explanation/Reference:
Reference: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0ahUKEwiW9-2j66rYAhUBORQKHaKOAnsQFggtMAE&url=http%3A%2F%2Fwww.oracle.com%2Ftechnetwork%2Fdatabase%2Ffeatures%2Fplsql%2Fncomp-faq087606.html&usg=AOvVaw3H2JhdwNaDzp-Jly5-wtTk

QUESTION 40
You execute this PL/SQL block:



```
SQL> DECLARE
  2        TYPE varchar_type1 IS VARRAY (3) OF VARCHR2 (15);
  3        TYPE varchar_type2 IS VARRAY (3) OF VARCHR2 (15);
  4        TYPE varchar_type3 IS VARRAY (3) OF VARCHR2 (15);
  5        TYPE nested_typ IS TABLE OF varchar_type3;
  6         n_table1 nested_typ := varchartype3 (‘AB1’, ‘AB2’, ‘AB3’);
  7         list_A varchar_type1: =varchar_type1 (‘Seattle’, ‘Tokyo’, ‘Paris’);
  8         list_B varchar_type1;
  9         list_C varchar_type2;
  10     BEGIN
  11       list_B := list_A;
 12        list_C := list_A;
 13      END;
 14 /
```

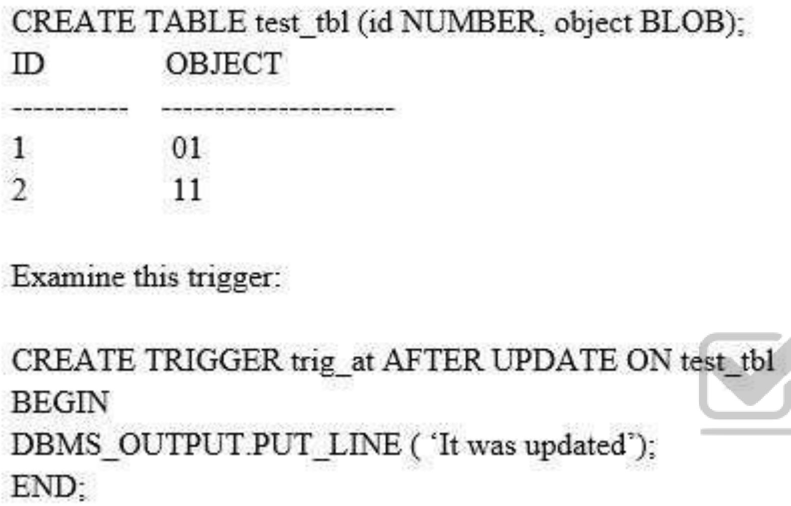
What will be the outcome?

A. It will fail compile because of errors at lines 11 and 12.
B. It will fail compile because of errors at lines 6 and 12.
C. It will fail compile because of error at line 7.D. It will fail compile and execute successfully.
E. It will fail compile because of errors at lines 5 and 6.

Correct Answer: C
Section: (none)

Explanation
Explanation/Reference:

QUESTION 41
Examine the test_tbl table and its contents:

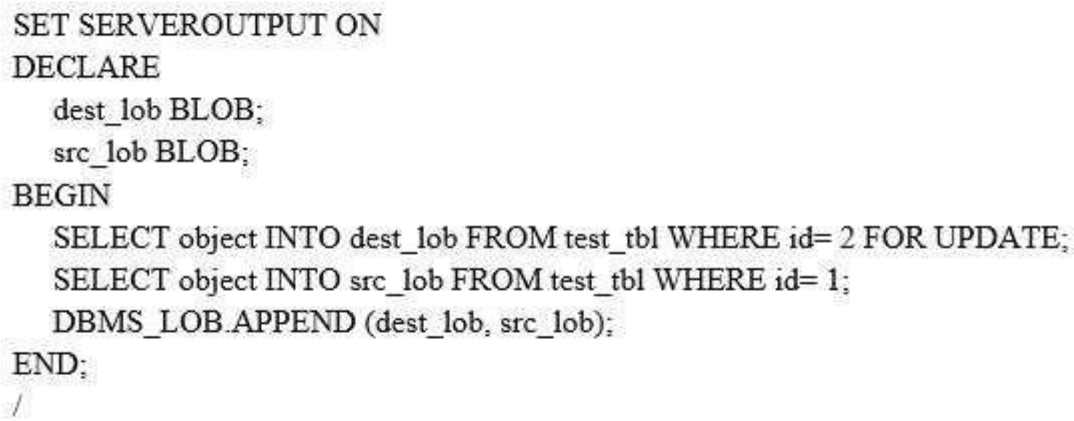


```
CREATE TABLE test_tbl (id NUMBER, object BLOB);
ID          OBJECT
----------- ----------------------
1            01
2            11
```

Examine this trigger:

```
CREATE TRIGGER trig_at AFTER UPDATE ON test_tbl
BEGIN
DBMS_OUTPUT.PUT_LINE ( 'It was updated');
END;
```

Examine this code:



```
SET SERVEROUTPUT ON
DECLARE
    dest_lob BLOB;
    src_lob BLOB;
BEGIN
    SELECT object INTO dest_lob FROM test_tbl WHERE id= 2 FOR UPDATE;
    SELECT object INTO src_lob FROM test_tbl WHERE id= 1;
    DBMS_LOB.APPEND (dest_lob, src_lob);
END;
/
```

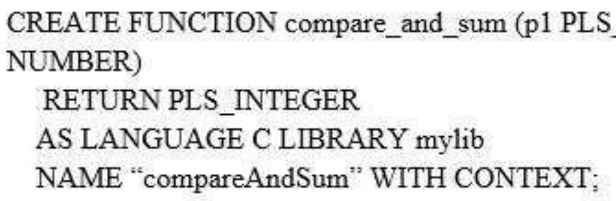
What is the outcome of this anonymous PL/SQL block?

A. “It was updated” is displayed.
B. Successful completion without printing “It was updated”.
C. A NO_DATA_FOUND exception is thrown.
D. ORA-06502: PL/SQL: numeric or value error: invalid LOB locator specified
E. ORA-22920: row containing the LOB value is not locked

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 42
Examine this external function declaration:



```
CREATE FUNCTION compare_and_sum (p1 PLS_INTEGER, p2 IN PLS_INTEGER, p3 IN OUT
NUMBER)
 RETURN PLS_INTEGER
 AS LANGUAGE C LIBRARY mylib
 NAME "compareAndSum" WITH CONTEXT;
```

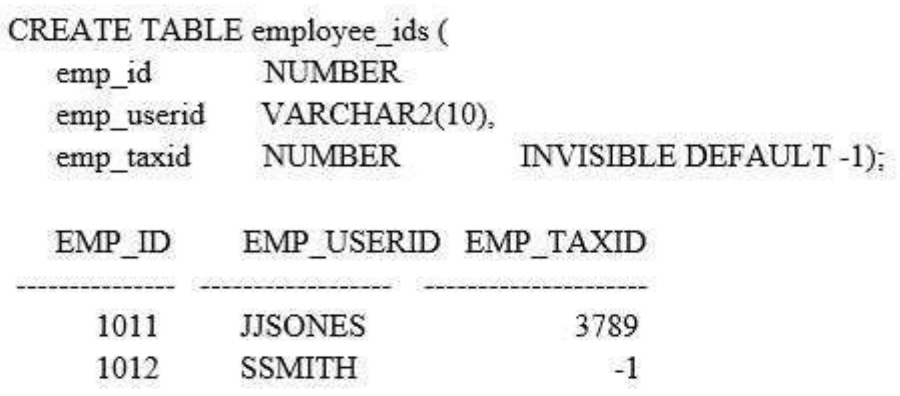
Which C function does it publish?

A. OCINumber * compareAndSum (OCIExtProcContext *ctx, OCINumber *p1, OCINumber *p2, OCINumber *p3);
B. OCINumber compareAndSum (OCIExtProcContext *ctx, OCINumber p1, OCINumber p2, OCINumber *p3);
C. int compareAndSum (OCIExtProcContext *ctx, int p1, int p2, OCINumber *p3);
D. int compareAndSum (OCIExtProcContext *ctx, int p1, int p2, OCINumber p3);
E. int compareAndSum (OCIExtProcContext *ctx, int p1, int p2, int p3);
F. int compareAndSum (OCIExtProcContext *ctx, int p1, int p2, int* p3);
G. OCINumber compareAndSum (OCIExtProcContext *ctx, OCINumber p1, OCINumber p2, OCINumber p3);

Correct Answer: F
Section: (none)
Explanation

Explanation/Reference:

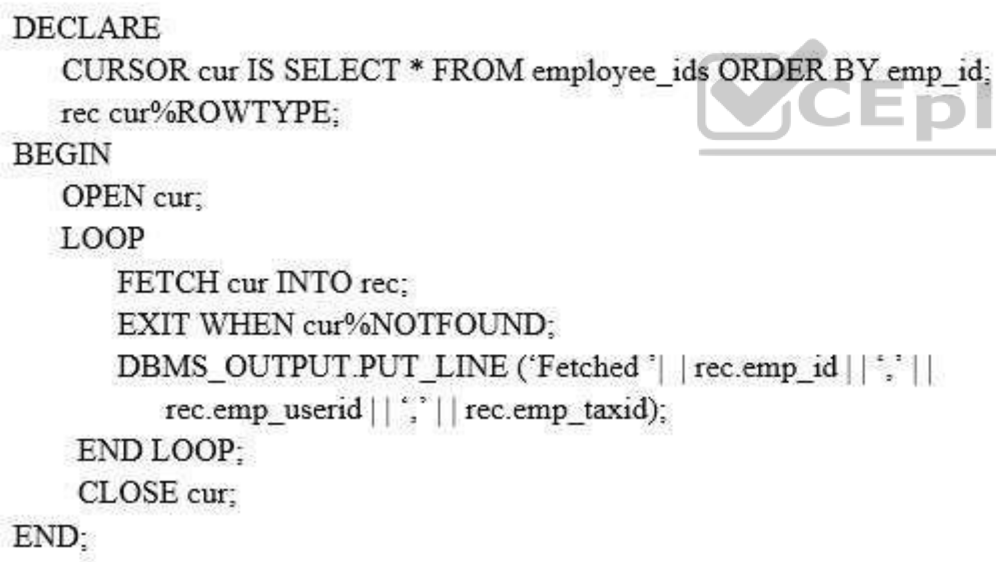
QUESTION 43
Examine the EMPLOYEE_IDS table its data:



```
CREATE TABLE employee_ids (
    emp_id          NUMBER
    emp_userid      VARCHAR2(10),
    emp_taxid       NUMBER          INVISIBLE DEFAULT -1);

    EMP_ID      EMP_USERID  EMP_TAXID
 ----------- ------------ --------------
      1011      JJSONES              3789
      1012      SSMITH                 -1
```

Examine this PL/SQL block:



```
DECLARE
    CURSOR cur IS SELECT * FROM employee_ids ORDER BY emp_id;
    rec cur%ROWTYPE;
BEGIN
    OPEN cur;
    LOOP
        FETCH cur INTO rec;
        EXIT WHEN cur%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE ('Fetched '| | rec.emp_id | | ',' | |
            rec.emp_userid | | ',' | | rec.emp_taxid);
     END LOOP;
     CLOSE cur;
END;
```

What is the result of executing this PL/SQL block with SERVEROUTPUT enabled?

A. It executes successfully and outputs:

Fetched: 1011, JJONES, 3789
Fetched: 1012, SSMITH, -1

B. Compilation fails saying EMP_TAXID must be declared.
C. An exception is thrown at runtime saying EMP_TAXID is not visible.
D. It executes successfully and outputs:
Fetched: 1011, JJONES,
Fetched: 1012, SSMITH,
E. It executes successfully and outputs:Fetched: 1011, JJONES, -1
Fetched: 1012, SSMITH, -1

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 44

Examine this code:

```
CREATE TYPE list_typ IS TABLE OF NUMBER;
/
DECLARE
l_list list_typ := list_typ ();
```

Which two executable sections will display the message TRUE?

A. BEGIN
```
IF l_list.LIMIT IS NOT NULL THEN
DBMS_OUTPUT.PUT_LINE ('TRUE');
END IF;
END;
```
B. BEGIN
```
l_list.EXTEND;
IF l_list.PRIOR (1_list.FIRST) IS NULL THEN
DBMS_OUTPUT.PUT_LINE ('TRUE');
END IF;
END;
```
C. BEGIN

l_list.EXTEND;
IF l_list IS EMPTY THEN
DBMS_OUTPUT.PUT_LINE (‘TRUE’);
END IF;
END;

D. BEGIN
IF l_list.FIRST IS NULL THEN
DBMS_OUTPUT.PUT_LINE (‘TRUE’);
END IF;
END;

E. BEGIN
IF l_list.FIRST =1 THEN
DBMS_OUTPUT.PUT_LINE (‘TRUE’);
END IF;
END;

Correct Answer: BE
Section: (none)
Explanation

Explanation/Reference:

QUESTION 45
In which situation will cached results become invalid?

A. When the memory allocated to the server result cache is increased using the RESULT_CACHE_MAX_SIZE initialization parameter
B. When a session executes an insert, update, or delete statement on a table or view that is queried by the result-cached function
C. When a session on this database instance invokes the function with the same parameter values
D. When the RESULT_CACHE_MODE parameter is set to FORCE.
E. When a new session is opened to invoke the function which is already cached

Correct Answer: CD
Section: (none)
Explanation

Explanation/Reference:
Reference: https://docs.oracle.com/database/121/TGDBA/tune_result_cache.htm

QUESTION 46

Which PRAGMA statement may enable associated PL/SQL functions to run more efficiently when called from SQL?

A. PRAGMA SERIALLY_REUSABLE;

B. PRAGMA UDF;

C. PRAGMA INLINE (‘<function_name>’, ‘YES’);

D. PRAGMA AUTONOMOUS_TRANSACTION;

Correct Answer: B

Section: (none)

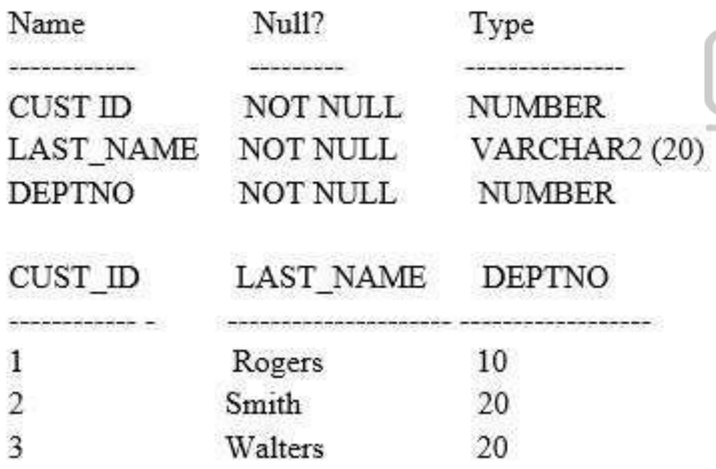
Explanation

Explanation/Reference:

Reference: https://mwidlake.wordpress.com/2015/11/04/pragma-udf-speeding-up-your-plsql-functions-called-from-sql/

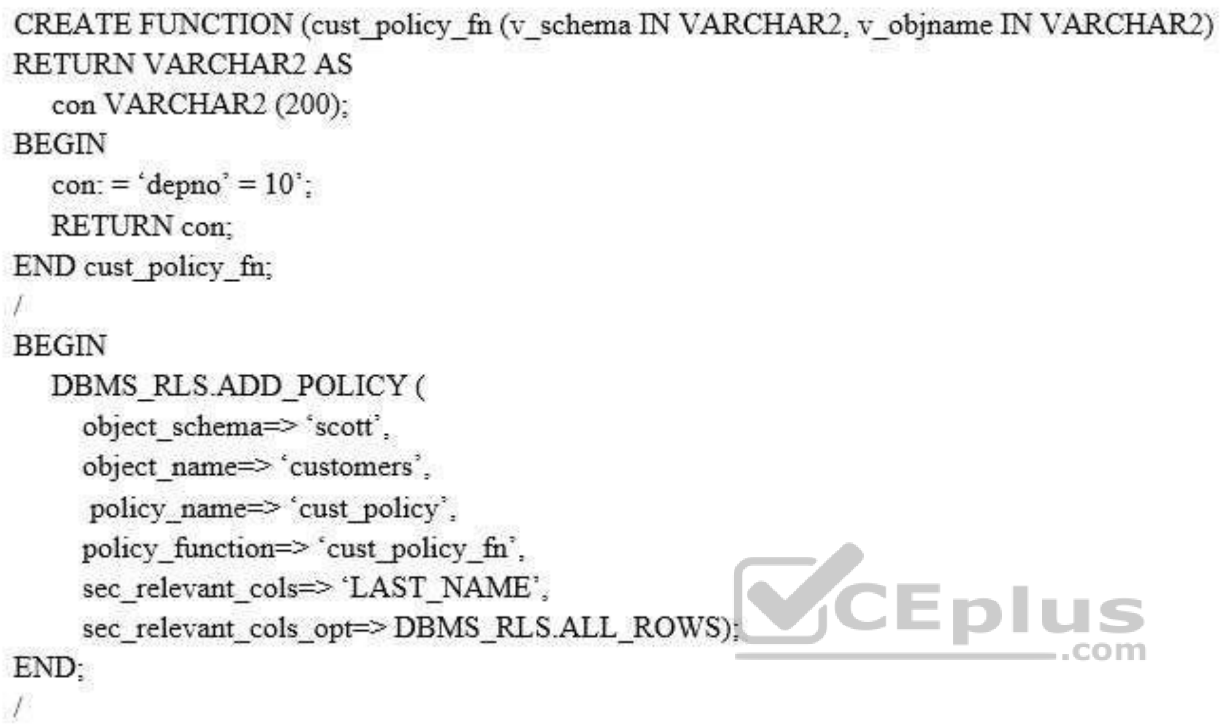
QUESTION 47

Examine the structure and data in the CUSTOMERS table:



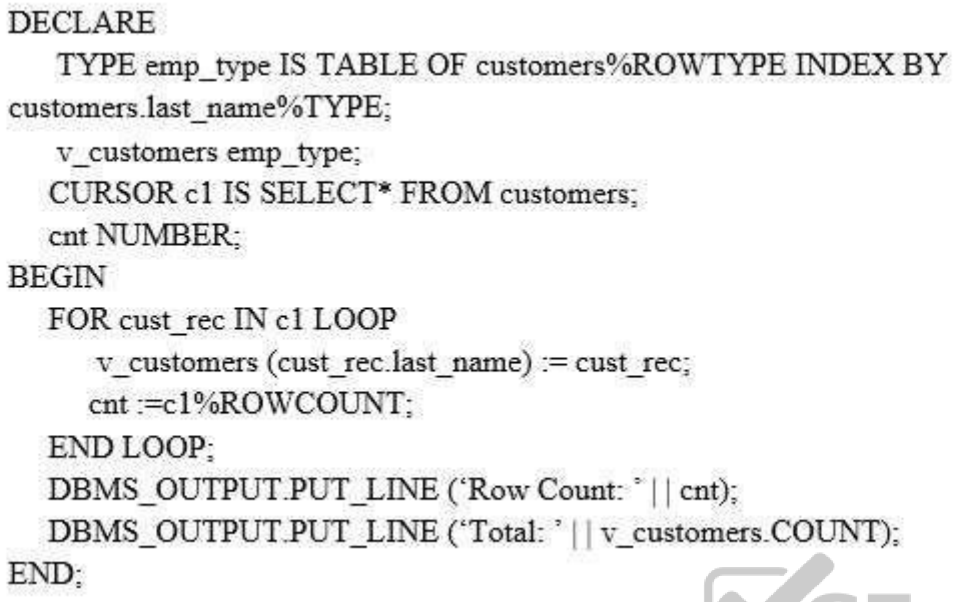
```
Name            Null?         Type
------------    ---------     ---------------
CUST ID         NOT NULL      NUMBER
LAST_NAME       NOT NULL      VARCHAR2 (20)
DEPTNO          NOT NULL      NUMBER

CUST_ID         LAST_NAME     DEPTNO
------------ -  --------------------- ------------------
1               Rogers        10
2               Smith         20
3               Walters       20
```



```
CREATE FUNCTION (cust_policy_fn (v_schema IN VARCHAR2, v_objname IN VARCHAR2)
RETURN VARCHAR2 AS
    con VARCHAR2 (200);
BEGIN
    con: = ‘depno’ = 10’;
    RETURN con;
END cust_policy_fn;
/
BEGIN
    DBMS_RLS.ADD_POLICY (
        object_schema=> ‘scott’,
        object_name=> ‘customers’,
         policy_name=> ‘cust_policy’,
        policy_function=> ‘cust_policy_fn’,
        sec_relevant_cols=> ‘LAST_NAME’,
        sec_relevant_cols_opt=> DBMS_RLS.ALL_ROWS);
END;
/
```

Examine this code:



```
DECLARE
    TYPE emp_type IS TABLE OF customers%ROWTYPE INDEX BY
customers.last_name%TYPE;
    v_customers emp_type;
   CURSOR c1 IS SELECT* FROM customers;
   cnt NUMBER;
BEGIN
   FOR cust_rec IN c1 LOOP
       v_customers (cust_rec.last_name) := cust_rec;
      cnt :=c1%ROWCOUNT;
   END LOOP;
   DBMS_OUTPUT.PUT_LINE (‘Row Count: ’ || cnt);
   DBMS_OUTPUT.PUT_LINE (‘Total: ’ || v_customers.COUNT);
END;
```

What is the outcome on execution with SERVEROUTPUT enabled?

A. It throws a NO_DATA_FOUND exception.
B. It displays 3 for both the row count and the total.
C. It throws a PL/SQL: numeric or value error: NULL index table key value exception.
D. It displays 3 for the row count and 2 for the total.

Correct Answer: B
Section: (none)

Explanation

Explanation/Reference: