

**Oracle9i Program with PL/SQL  
1Z0-147**

**Demo Version  
From  
ITCertKeys.com  
To  
CertsBraindumps.com**

**Question 1.**

Examine this function:

```
CREATE OR REPLACE FUNCTION CALC_PLAYER_AVG
(V_ID in PLAYER_BAT_STAT.PLAYER_ID%TYPE)
RETURN NUMBER
IS
V_AVG NUMBER;
BEGIN
SELECT HITS / AT_BATS
INTO V_AVG
FROM PLAYER_BAT_STAT
WHERE PLAYER_ID = V_ID;
RETURN (V_AVG);
END;
```

Which statement will successfully invoke this function in SQL \*Plus?

- A. SELECT CALC\_PLAYER\_AVG(PLAYER\_ID) FROM PLAYER\_BAT\_STAT;
- B. EXECUTE CALC\_PLAYER\_AVG(31);
- C. CALC\_PLAYER(.RUTH.);
- D. CALC\_PLAYER\_AVG(31);
- E. START CALC\_PLAYER\_AVG(31)

**Answer: A**

**Explanation:****Incorrect Answers**

B. You can't call a function in this way, in this way you can call a procedure, because function must return a value, to call a function using EXECUTE command you should declare a bind variable using the VARIABLE command then assign the value returned from the function to this variable, in the following way:

```
SQL> VARIABLE v_get_value NUMBER
SQL> EXECUTE :v_get_value := CALC_PLAYER_AVG(31)
PL/SQL procedure successfully completed.
SQL> PRINT v_get_value
V_GET_VALUE
1
```

C. Again this way can't be use for calling a function in PL/SQL block because the function return a value

and this values must be assigned to PL/SQL variable or to bind variable. Like this

```
DECLARE
v_get_from_fn NUMBER;
BEGIN
v_get_from := CALC_PLAYER_AVG(31);
END;
/
```

D. Same as C.

E. START is use to execute a script.

**Question 2.**

Which three are true statements about dependent objects? (Choose three)

- A. Invalid objects cannot be described.

- B. An object with status of invalid cannot be a referenced object.
- C. The Oracle server automatically records dependencies among objects.
- D. All schema objects have a status that is recorded in the data dictionary.
- E. You can view whether an object is valid or invalid in the USER\_STATUS data dictionary view.
- F. You can view whether an object is valid or invalid in the USER\_OBJECTS data dictionary view.

**Answer: A, C, F**

**Question 3.**

You have created a stored procedure DELETE\_TEMP\_TABLE that uses dynamic SQL to remove a table in your schema. You have granted the EXECUTE privilege to user A on this procedure. When user A executes the DELETE\_TEMP\_TABLE procedure, under whose privileges are the operations performed by default?

- A. SYS privileges
- B. Your privileges
- C. Public privileges
- D. User A.s privileges
- E. User A cannot execute your procedure that has dynamic SQL.

**Answer: B**

**Explanation:**

When you create a procedure, it will be executed under the privileges of the creator, unless the procedure has the following statement AUTHID CURRENT\_USER. If you specify AUTHID CURRENT\_USER, the privileges of the current user are checked at run time, and external references are resolved in the schema of the current user. Like this example

```
SQL> CREATE OR REPLACE PROCEDURE delete_temp_table(v_table varchar2)
2 AUTHID CURRENT_USER
3 IS
4 BEGIN
5 EXECUTE IMMEDIATE 'DROP TABLE '||v_table;
6 END;
7 /
```

Procedure created.

If the procedure is create in this way then the EXECUTE IMMEDIATE statement will be execute under the privilege of the user who executes the procedure, but if we skip line 2 then the procedure will be executed under the privilege of the owner of the procedure.

**Incorrect Answers**

- A: SYS privilege has nothing with is.
- C: What is the public privileges? There is nothing called public privileges.
- D: This will be true if the procedure contains the AUTHID CURRENT\_USER.
- E: There is no problem in having a dynamic SQL statement in Procedure.

**Question 4.**

Examine this code:

Examine this code:

```
CREATE OR REPLACE PRODECURE add_dept(p_dept_name VARCHAR2 DEFAULT
'placeholder', p_location VARCHAR2 DEFAULT 'Boston')
```

```

IS
BEGIN
INSERT INTO departments
VALUES (dept_id_seq.NEXTVAL, p_dept_name, p_location);
END add_dept;
/

```

Which three are valid calls to the add\_dep procedure ? (Choose three)

- A. add\_dept;
- B. add\_dept( .Accounting .);
- C. add\_dept(, .New York .);
- D. add\_dept(p\_location=> .New York .);

**Answer: A, B, D**

**Explanation:**

A is correct because both of the parameter have a default values.

B is correct because here we call the procedure using position notation, and the first parameter for the procedure will have the value 'Accounting', and since the second parameter has a default value then we can skip it, and in this case it will take the default value.

D is correct because here we are calling the procedure using naming notation, the value 'New York' will go to the parameter p\_location, and the parameter p\_dept\_name will have the default value.

**Incorrect Answer**

C: You can't use this way and assume that the PL/SQL will understand that he should assign the default value for the first parameter. This is incorrect way for calling.

**Question 5.**

Which two statements about packages are true? (Choose two)

- A. Packages can be nested.
- B. You can pass parameters to packages.
- C. A package is loaded into memory each time it is invoked.
- D. The contents of packages can be shared by many applications.
- E. You can achieve information hiding by making package constructs private.

**Answer: D, E**

**Explanation:**

Actually theses are some of the advantages of the package, sharing the package among applications and hide the logic of the procedures and function that are inside the package by declaring them in the package header and write the code of these procedures and functions inside the package body.

**Incorrect Answers:**

A: Packages can not be nested

B: Parameters can't be passed to a package; parameters can be passed to procedures and functions only.

C: By the first time you call a procedure, function, or reference a global variable within the package, the whole package will be loaded into the memory and stay there, so when ever you need to reference any of the package's constructs again you will find it in the memory.

**Question 6.**

Which two programming constructs can be grouped within a package? (Choose two)

- A. Cursor
- B. Constant
- C. Trigger
- D. Sequence
- E. View

**Answer: A, B**

**Explanation:**

**Incorrect Answers**

C: Triggers are objects that we create are created on the tables.

D: Sequences can't be grouped inside the packages, but we can reference them inside the package.

E: Views are created and they are database objects, and they can't be grouped inside the packages.

**Question 7.**

Which two statements describe the state of a package variable after executing the package in which it is declared? (Choose two)

- A. It persists across transactions within a session.
- B. It persists from session to session for the same user.
- C. It does not persist across transaction within a session.
- D. It persists from user to user when the package is invoked.
- E. It does not persist from session to session for the same user.

**Answer: A, E**

**Explanation:**

You can keep track of the state of a package variable or cursor, which persists throughout the user session, from the time the user first references the variable or cursor to the time the user disconnects.

1. Initialize the variable within its declaration or within an automatic, one-time-only procedure.
2. Change the value of the variable by means of package procedures.
3. The value of the variable is released when the user disconnects.

**Incorrect Answers**

B: Each session will have its own value for the variables

C: It persists across the transactions and through the user session.

D: Each user has his own values and results, because each user has his own users.

**Question 8.**

Which code can you use to ensure that the salary is not increased by more than 10% at a time nor is it ever decreased?

- A. ALTER TABLE emp ADD CONSTRAINT ck\_sal CHECK (sal BETWEEN sal AND sal\*1.1);
- B. CREATE OR REPLACE TRIGGER check\_sal BEFORE UPDATE OF sal ON emp FOR EACH ROW WHEN (new.sal < old.sal OR new.sal > old.sal \* 1.1) BEGIN RAISE\_APPLICATION\_ERROR ( - 20508, .Do not decrease salary not increase by more than 10% ); END;

- C. CREATE OR REPLACE TRIGGER check\_sal BEFORE UPDATE OF sal ON emp  
 WHEN (new.sal <  
 old.sal OR new.sal > old.sal \* 1.1)  
 BEGIN RAISE\_APPLICATION\_ERROR ( - 20508, .Do not decrease salary not  
 increase by more than 10% );  
 END;
- D. CREATE OR REPLACE TRIGGER check\_sal AFTER UPDATE OR sal ON emp  
 WHEN (new.sal <  
 old.sal OR -new.sal > old.sal \* 1.1)  
 BEGIN RAISE\_APPLICATION\_ERROR ( - 20508, .Do not decrease salary not  
 increase by more than 10% );  
 END;

**Answer: B**

**Explanation:**

Row triggers are the correct chose for solving the problem. A row trigger fires each time the table is affected by the triggering event. If the triggering event affects no rows, a row trigger is not executed. Row triggers are useful if the trigger action depends on data of rows that are affected or on data provided by the triggering event itself. You can create a BEFORE row trigger in order to prevent the triggering operation from succeeding if a certain condition is violated. Within a ROW trigger, reference the value of a column before and after the data change by prefixing it with the OLD and NEW qualifier.

**Incorrect Answers:**

A: Check constraint can't do this job lets take a look:

```
SQL> ALTER TABLE emp ADD
2 CONSTRAINT ck_sal CHECK (sal BETWEEN sal AND sal*1.1)
3 /
```

Table altered.

```
SQL> select ename, sal
2 from emp
3 where ename = 'CERT';
ENAME SAL
CERT 5000
```

Now let's issue an update statement

```
SQL> update emp
2 set sal = 10
3 where ename = 'CERT';
1 row updated.
```

As you can see the check constraint can't compare the old value with the new value.

D,C: You can use NEW and OLD qualifier with row level triggers, If in the CREATE TRIGGER statement you didn't say FOR EACH ROW then the trigger will be statement level trigger

**Question 9.**

Examine this code:

```
CREATE OR REPLACE PACKAGE bonus
IS
g_max_bonus NUMBER := .99;
FUNCTION calc_bonus (p_emp_id NUMBER)
RETURN NUMBER;
FUNCTION calc_salary (p_emp_id NUMBER)
RETURN NUMBER;
END;
```

```

/
CREATE OR REPLACE PACKAGE BODY bonus
IS
v_salary employees.salary%TYPE;
v_bonus employees.commission_pct%TYPE;
FUNCTION calc_bonus (p_emp_id NUMBER)
RETURN NUMBER
IS
BEGIN
SELECT salary, commission_pct
INTO v_salary, v_bonus
FROM employees
WHERE employee_id = p_emp_id;
RETURN v_bonus * v_salary;
END calc_bonus
FUNCTION calc_salary (p_emp_id NUMBER)
RETURN NUMBER
IS
BEGIN
SELECT salary, commission_pct
INTO v_salary, v_bonus
FROM employees
WHERE employees
RETURN v_bonus * v_salary + v_salary;
END calc_salary;
END bonus;
/

```

Which statement is true?

- A. You can call the BONUS.CALC\_SALARY packaged function from an INSERT command against the EMPLOYEES table.
- B. You can call the BONUS.CALC\_SALARY packaged function from a SELECT command against the EMPLOYEES table.
- C. You can call the BONUS.CALC\_SALARY packaged function from a DELETE command against the EMPLOYEES table.
- D. You can call the BONUS.CALC\_SALARY packaged function from an UPDATE command against the EMPLOYEES table.

**Answer: B**

**Explanation:**

For the Oracle server to execute a SQL statement that calls a stored function, it must know the purity level of a stored functions, that is, whether the functions are free of side effects. Side effects are changes to database tables or public packaged variables (those declared in a package specification). Side effects could delay the execution of a query, yield order-dependent (therefore indeterminate) results, or require that the package state variables be maintained across user sessions. Various side effects are not allowed when a function is called from a SQL query or DML statement. Therefore, the following restrictions apply to stored functions called from SQL expressions:

- A function called from a query or DML statement may not end the current transaction, create or roll back to a savepoint, or alter the system or session
- A function called from a query statement or from a parallelized DML statement may not execute a DML statement or otherwise modify the database
- A function called from a DML statement may not read or modify the particular table being modified by that DML statement

**Question 10.**

Which statement is valid when removing procedures?

- Use a drop procedure statement to drop a standalone procedure.
- Use a drop procedure statement to drop a procedure that is part of a package. Then recompile the package specification.
- Use a drop procedure statement to drop a procedure that is part of a package. Then recompile the package body.
- For faster removal and re-creation, do not use a drop procedure statement. Instead, recompile the procedure using the alter procedure statement with the REUSE SETTINGS clause.

**Answer: A**

**Explanation:**

The DROP PROCEDURE statement is used to drop a stand alone procedure

**Incorrect Answers:**

B: You can't drop a procedure that's inside a package, you have to drop the package, and in this case the whole procedures, functions,... that are inside the packages will be dropped.

C: Same as B.

D: REUSE SETTINGS is used to prevent Oracle from dropping and reacquiring compiler switch settings. With this clause, Oracle preserves the existing settings and uses them for the recompilation.

**Question 11.**

Examine this package:

```
CREATE OR REPLACE PACKAGE BB_PACK
IS
  V_MAX_TEAM_SALARY NUMBER(12,2);
  PROCEDURE ADD_PLAYER(V_ID IN NUMBER, V_LAST_NAME VARCHAR2, V_SALARY
NUMBER);
END BB_PACK;
/
CREATE OR REPLACE PACKAGE BODY BB_PACK
IS
  PROCEDURE UPD_PLAYER_STAT
(V_ID IN NUMBER, V_AB IN NUMBER DEFAULT 4, V_HITS IN NUMBER)
IS
  BEGIN
    UPDATE PLAYER_BAT_STAT
    SET AT_BATS = AT_BATS + V_AB,
    HITS = HITS + V_HITS
    WHERE PLAYER_ID = V_ID;
    COMMIT;
  END UPD_PLAYER_STAT;
  PROCEDURE ADD_PLAYER
(V_ID IN NUMBER, V_LAST_NAME VARCHAR2, V_SALARY NUMBER)
```



```

IS
BEGIN
INSERT INTO PLAYER(ID, LAST_NAME, SALARY)
VALUES (V_ID, V_LAST_NAME, V_SALARY);
UPD_PLAYER_STAT(V_ID, 0, 0);
END ADD_PLAYER;
END BB_PACK;

```

You make a change to the body of the BB\_PACK package.

The BB\_PACK body is recompiled.

What happens if the stand alone procedure VALIDATE\_PLAYER\_STAT references this package?

- A. VALIDATE\_PLAYER\_STAT cannot recompile and must be recreated.
- B. VALIDATE\_PLAYER\_STAT is not invalidated.
- C. VALIDATE\_PLAYER\_STAT is invalidated.
- D. VALIDATE\_PLAYER\_STAT and BB\_PACK are invalidated.

**Answer: B**

**Explanation:**

You can greatly simplify dependency management with packages when referencing a package procedure or function from a stand-alone procedure or function.

- If the package body changes and the package specification does not change, the stand-alone procedure referencing a package construct remains valid.
- If the package specification changes, the outside procedure referencing a package construct is invalidated, as is the package body.

**Question 12.**

You need to create a trigger on the EMP table that monitors every row that is changed and places this information into the AUDIT\_TABLE . What type of trigger do you create?

- A. FOR EACH ROW trigger on the EMP table.
- B. Statement-level trigger on the EMP table.
- C. FOR EACH ROW trigger on the AUDIT\_TABLE table.
- D. Statement-level trigger on the AUDIT\_TABLE table.
- E. FOR EACH ROW statement-level trigger on the EMP table.

**Answer: A**

**Explanation:**

FOR EACH ROW trigger on the updated table(emp) should be create to record each update row in the AUDIT\_TABLE.

**Question 13.**

Which statements are true? (Choose all that apply)

- A. If errors occur during the compilation of a trigger, the trigger is still created.
- B. If errors occur during the compilation of a trigger you can go into SQL \*Plus and query the USER\_TRIGGERS data dictionary view to see the compilation errors.
- C. If errors occur during the compilation of a trigger you can use the SHOW ERRORS command within iSQL \*Plus to see the compilation errors.
- D. If errors occur during the compilation of a trigger you can go into SQL \*Plus and

query the USER\_ERRORS data dictionary view to see compilation errors.

**Answer: A, C, D**

**Question 14.**

Which two dictionary views track dependencies? (Choose two)

- A. USER\_SOURCE
- B. UTL\_DEPTREE
- C. USER\_OBJECTS
- D. DEPTREE\_TEMPTAB
- E. USER\_DEPENDENCIES
- F. DBA\_DEPENDENT\_OBJECTS

**Answer: D, E**

**Question 15.**

Given a function CALCTAX :

```
CREATE OR REPLACE FUNCTION calc tax (sal NUMBER) RETURN NUMBER
IS
BEGIN
RETURN (sal * 0.05);
END;
```

If you want to run the above function from the SQL \*Plus prompt, which statement is true?

- A. You need to execute the command CALCTAX(1000); .
- B. You need to execute the command EXECUTE FUNCTION calc tax; .
- C. You need to create a SQL \*Plus environment variable X and issue the command :X := CALCTAX(1000); .
- D. You need to create a SQL \*Plus environment variable X and issue the command EXECUTE :X := CALCTAX;
- E. You need to create a SQL \*Plus environment variable X and issue the command EXECUTE :X := CALCTAX(1000);

**Answer: E**

**Explanation:**

When you call a function from SQL\*PLUS you need to assign the returned value a bind variable, and you need the EXECUTE command to execute the function.

**Question 16.**

What happens during the execute phase with dynamic SQL for INSERT, UPDATE, and DELETE operations?

- A. The rows are selected and ordered.
- B. The validity of the SQL statement is established.
- C. An area of memory is established to process the SQL statement.
- D. The SQL statement is run and the number of rows processed is returned.
- E. The area of memory established to process the SQL statement is released.

**Answer: D**

**Explanation:**

All SQL statements have to go through various stages. Some stages may be skipped.

1. Parse Every SQL statement must be parsed. Parsing the statement includes checking the statement's syntax and validating the statement, ensuring that all references to objects are correct, and ensuring that the relevant privileges to those objects exist.
2. Bind After parsing, the Oracle server knows the meaning of the Oracle statement but still may not have enough information to execute the statement. The Oracle server may need values for any bind variable in the statement. The process of obtaining these values is called binding variables.
3. Execute At this point, the Oracle server has all necessary information and resources, and the statement is executed.
4. Fetch In the fetch stage, rows are selected and ordered (if requested by the query), and each successive fetch retrieves another row of the result, until the last row has been fetched. You can fetch queries, but not the DML statements.

### Question 17

What part of a database trigger determines the number of times the trigger body executes?

- A. Trigger type
- B. Trigger body
- C. Trigger event
- D. Trigger timing

**Answer: A**

### Explanation:

Part	Description	Possible
trigger timing	When the trigger fires in relation to the triggering event	BEFORE AFTER
Triggering event	Which data manipulation operation on the table or view causes the trigger to fire	INSERT UPDATE DELETE
Trigger type	How many times the trigger body executes	Statement Row
Trigger body	What action the trigger performs	Complete PL/SQL block

### Question 18.

Examine this code:

```
CREATE OR REPLACE FUNCTION gen_email_name
(p_first_name VARCHAR2, p_last_name VARCHAR2, p_id NUMBER)
RETURN VARCHAR2
is
v_email_name VARCHAR2(19);
BEGIN
v_email_name := SUBSTR(p_first_name, 1, 1) ||
SUBSTR(p_last_name, 1, 7) ||
'@Oracle.com';
UPDATE employees
SET email = v_email_name
WHERE employee_id = p_id;
RETURN v_email_name;
END;
You run this SELECT statement:
SELECT first_name, last_name
gen_email_name(first_name, last_name, 108) EMAIL
```

FROM employees;

What occurs?

- A. Employee 108 has his email name updated based on the return result of the function.
- B. The statement fails because functions called from SQL expressions cannot perform DML.
- C. The statement fails because the functions does not contain code to end the transaction.
- D. The SQL statement executes successfully, because UPDATE and DELETE statements are ignoring in stored functions called from SQL expressions.
- E. The SQL statement executes successfully and control is passed to the calling environment.

**Answer: B**

**Explanation:**

- When called from a SELECT statement or a parallelized UPDATE or DELETE statement, the function cannot modify any database tables
- When called from an UPDATE, or DELETE statement, the function cannot query or modify any database tables modified by that statement.
- When called from a SELECT, INSERT, UPDATE, or DELETE statement, the function cannot execute SQL transaction control statements (such as COMMIT), session control statements (such as SET ROLE), or system control statements (such as ALTER SYSTEM). Also, it cannot execute DDL statements (such as CREATE) because they are followed by an automatic commit.
- The function cannot call another subprogram that breaks one of the above restrictions.

**Question 19.**

Which table should you query to determine when your procedure was last compiled?

- A. USER\_PROCEEDURES
- B. USER\_PROCS
- C. USER\_OBJECTS
- D. USER\_PLSQL\_UNITS

**Answer: C**

**Explanation:**

In the USER\_OBJECTS there is

**Incorrect Answers**

- A. USER\_PROCEEDURES lists all functions and procedures, along with associated properties. For example, ALL\_PROCEEDURES indicates whether or not a function is pipelined, parallel enabled or an aggregate function. If a function is pipelined or an aggregate function, the associated implementation type (if any) is also identified. It doesn't have when the object was last compiled.
- B. There is nothing called USER\_PROCS.
- D. There is nothing called USER\_PLSQL\_UNITS

**Question 20.**

Examine this code:

```
CREATE OR REPLACE TRIGGER secure_emp
BEFORE LOGON ON employees
BEGIN
IF (TO_CHAR(SYSDATE, 'DY') IN ('SAT', 'SUN')) OR
(TO_CHAR(SYSDATE, 'HH24:MI')
```

```

NOT BETWEEN '08:00' AND '18:00')
THEN RAISE_APPLICATION_ERROR (-20500, 'You may
insert into the EMPLOYEES table only during
business hours. ');
END IF;
END;
/

```

What type of trigger is it?

- A. DML trigger
- B. INSTEAD OF trigger
- C. Application trigger
- D. System event trigger
- E. This is an invalid trigger.

**Answer: E**

**Explanation:**

As you can see there is nothing called BEFORE LOGON

**Question 21.**

Examine this package:

```

CREATE OR REPLACE PACKAGE discounts
IS
g_id NUMBER := 7829;
discount_rate NUMBER := 0.00;
PROCEDURE display_price (p_price NUMBER);
END discounts;
/
CREATE OR REPLACE PACKAGE BODY discounts
IS
PROCEDURE display_price (p_price NUMBER)
IS
BEGIN
DBMS_OUTPUT.PUT_LINE('Discounted '||
TO_CHAR(p_price*NVL(discount_rate, 1)));
END display_price;
BEGIN
discount_rate := 0.10;
END discounts;
/

```

Which statement is true?

- A. The value of DISCOUNT\_RATE always remains 0.00 in a session.
- B. The value of DISCOUNT\_RATE is set to 0.10 each time the package is invoked in a session.
- C. The value of DISCOUNT\_RATE is set to 1.00 each time the procedure DISPLAY\_PRICE is invoked.
- D. The value of DISCOUNT\_RATE is set to 0.10 when the package is invoked for the first time in a session.

**Answer: D**

**Explanation:**

A one-time-only procedure is executed only once, when the package is first invoked within the user session

**Question 22.**

Examine this code:

```
CREATE OR REPLACE TRIGGER update_emp
AFTER UPDATE ON emp
BEGIN
INSERT INTO audit_table (who, dated)
VALUES (USER, SYSDATE);
END;
```

You issue an UPDATE command in the EMP table that results in changing 10 rows.

How many rows are inserted into the AUDIT\_TABLE ?

- A. 1
- B. 10
- C. None
- D. A value equal to the number of rows in the EMP table.

**Answer: A**

**Question 23.**

Examine this package:

```
CREATE OR REPLACE PACKAGE BB_PACK
IS
V_MAX_TEAM_SALARY NUMBER(12,2);
PROCEDURE ADD_PLAYER(V_ID IN NUMBER, V_LAST_NAME VARCHAR2,
V_SALARY NUMBER);
END BB_PACK;
/
CREATE OR REPLACE PACKAGE BODY BB_PACK
IS
PROCEDURE UPD_PLAYER_STAT
(V_ID IN NUMBER, V_AB IN NUMBER DEFAULT 4, V_HITS IN NUMBER)
IS
BEGIN
UPDATE PLAYER_BAT_STAT
SET AT_BATS = AT_BATS + V_AB,
HITS = HITS + V_HITS
WHERE PLAYER_ID = V_ID)
COMMIT;
END UPD_PLAYER_STAT;
PROCEDURE ADD_PLAYER
(V_ID IN NUMBER, V_LAST_NAME VARCHAR2, V_SALARY NUMBER)
IS
BEGIN
INSERT INTO PLAYER(ID, LAST_NAME, SALARY)
VALUES (V_ID, V_LAST_NAME, V_SALARY);
UPD_PLAYER_STAT(V_ID, 0.0);
END ADD_PLAYER;
END BB_PACK;
```

Which statement will successfully assign \$75,000,000 to the V\_MAX\_TEAM\_SALARY variable from within a stand-alone procedure?

- A. V\_MAX\_TEAM\_SALARY := 7500000;
- B. BB\_PACK.ADD\_PLAYER.V\_MAX\_TEAM\_SALARY := 75000000;
- C. BB\_PACK.V\_MAX\_TEAM\_SALARY := 75000000;
- D. This variable cannot be assigned a value from outside the package.

**Answer: C**

**Explanation:**

To assign a value for a public variable which is declared in the package header, all what you have to do is do use the following syntax package\_name.var\_name:=value;

**Question 24.**

Which of the following oracle supplied package is used to Enable HTTP callouts from PL/SQL and SQL to access data on the Internet?

- A. DBMS\_DDL
- B. UTL\_HTTP
- C. UTL\_SMTP
- D. UTL\_URL

**Answer: B**

**Explanation:**

“UTL\_HTTP” enables HTTP callouts from PL/SQL and SQL to access data on the Internet or to call Oracle Web Server Cartridges.

**Question 25.**

The DBMS\_DDL package provides access from within PL/SQL to

- A. One DDL
- B. Two DDL
- C. Three DDL
- D. Four DDL

**Answer: B**

**Explanation:**

The DBMS\_DDL package provides access from within PL/SQL to two DDL.