



**Oracle 1z0-147**

Oracle9i: Program with PL/SQL

Version 3.0

**Leading The Way**

in IT Testing And Certification Tools

**[www.testking.com](http://www.testking.com)**

## **Important Note, Please Read Carefully**

### **Study Tips**

This product will provide you questions and answers along with detailed explanations carefully compiled and written by our experts. Try to understand the concepts behind the questions instead of cramming the questions. Go through the entire document at least twice so that you make sure that you are not missing anything.

### **Further Material**

For this test TestKing also plans to provide:

\* Online Testing. Check out an Online Testing Demo at <http://www.testking.com/index.cfm?pageid=724>

### **Latest Version**

We are constantly reviewing our products. New material is added and old material is revised. Free updates are available for 90 days after the purchase. You should check your member zone at TestKing an update 3-4 days before the scheduled exam date.

Here is the procedure to get the latest version:

1. Go to [www.testking.com](http://www.testking.com)
2. Click on **Member zone/Log in**
3. The latest versions of all purchased products are downloadable from here. Just click the links.

For most updates, it is enough just to print the new questions at the end of the new version, not the whole document.

### **Feedback**

Feedback on specific questions should be send to [feedback@testking.com](mailto:feedback@testking.com). You should state: Exam number and version, question number, and login ID.

Our experts will answer your mail promptly.

### **Copyright**

Each pdf file contains a unique serial number associated with your particular name and contact information for security purposes. So if we find out that a particular pdf file is being distributed by you, TestKing reserves the right to take legal action against you according to the International Copyright Laws.

**QUESTION NO: 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****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 NO: 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**

**Incorrect answers: B, D, E**

**QUESTION NO: 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**

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 NO: 4****Examine this code:**

```
CREATE OR REPLACE PROCEDURE 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**

**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. The following table list the for passing parameters to a procedure:

**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 NO: 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**

Actually these 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 NO: 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**

**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 then inside the package.

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

**QUESTION NO: 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**

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 NO: 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**

Row triggers are the correct choice 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 let's 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 = 'KING';

```

ENAME	SAL
KING	5000

Now let's issue an update statement

```

SQL> update emp
  2 set sal = 10
  3 where ename = 'KING';

```

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 NO: 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 cacl_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 form 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**

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 NO: 10

**Which statement is valid when removing procedures?**

- A. Use a drop procedure statement to drop a standalone procedure.
- B. Use a drop procedure statement to drop a procedure that is part of a package. Then recompile the package specification.
- C. Use a drop procedure statement to drop a procedure that is part of a package. Then recompile the package body.
- D. 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

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 NO: 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?**

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

**Answer: B**

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 NO: 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?**

- 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**

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

**QUESTION NO: 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 NO: 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 NO: 15**

**Given a function CALCTAX:**

```
CREATE OR REPLACE FUNCTION calctax (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 calctax;`.
- 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**

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 NO: 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**

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 NO: 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**

**QUESTION NO: 18**

**Examine this code:**

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

```

CREATE OR REPLACE TRIGGER 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_home := 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**

- 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 NO: 19**

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

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

**Answer: C**

In the USER\_OBJECTS there is

**Incorrect Answers**

- A.** USER\_PROCEDURES lists all functions and procedures, along with associated properties. For example, ALL\_PROCEDURES 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 NO: 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**

As you can see there is nothing called BEFORE LOGON

**QUESTION NO: 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**

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

**QUESTION NO: 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 NO: 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**

**To assign a value for a public variable which is declared in the package header, all what you have to do is do user the following syntax**

**package\_name.var\_name:=value;**

**QUESTION NO: 24**

**There is a CUSTOMER table in a schema that has a public synonym CUSTOMER and you are granted all object privileges on it. You have a procedure PROCESS\_CUSTOMER that processes customer information that is in the public synonym CUSTOMER table. You have just created a new table called CUSTOMER within your schema.**

**Which statement is true?**

- A. Creating the table has no effect and procedure PROCESS\_CUSTOMER still accesses data from public synonym CUSTOMER table.
- B. If the structure of your CUSTOMER table is the same as the public synonym CUSTOMER table then the procedure PROCESS\_CUSTOMER is invalidated and gives compilation errors.

- C. If the structure of your CUSTOMER table is entirely different from the public synonym CUSTOMER table then the procedure PROCESS\_CUSTOMER successfully recompiles and accesses your CUSTOMER table.
- D. If the structure of your CUSTOMER table is the same as the public synonym CUSTOMER table then the procedure PROCESS\_CUSTOMER successfully recompiles when invoked and accesses your CUSTOMER table.

**Answer: D**

The procedure will first look in the owner of the procedure schema before looking for the public synonym.

**Incorrect Answers:**

**A, B, C**

**QUESTION NO: 25**

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

- A. Both the specification and body are required components of a package.
- B. The package specification is optional, but the package body is required.
- C. The package specification is required, but the package body is optional.
- D. The specification and body of the package are stored together in the database.
- E. The specification and body of the package are stored separately in the database.

**Answer: C,E**

**QUESTION NO: 26**

**When creating a function in SQL \*Plus, you receive this message:**

**“Warning: Function created with compilation errors.”**

**Which command can you issue to see the actual error message?**

- A. SHOW FUNCTION\_ERROR
- B. SHOW USER\_ERRORS
- C. SHOW ERRORS
- D. SHOW ALL\_ERRORS

**Answer: C**

**QUESTION NO: 27**

Which four triggering events can cause a trigger to fire? (Choose four)

- A. A specific error or any errors occurs.
- B. A database is shut down or started up.
- C. A specific user or any user logs on or off.
- D. A user executes a CREATE or an ALTER table statement.
- E. A user executes a SELECT statement with an ORDER BY clause.
- F. A user executes a JOIN statement that uses four or more tables.

Answer: A,B,C,D

**QUESTION NO: 28**

Examine this procedure:

```
CREATE OR REPLACE PROCEDURE ADD_PLAYER
(V_ID IN NUMBER, V_LAST_NAME VARCHAR2)
IS
    BEGIN
        INSERT INTO PLAYER (ID, LAST_NAME)
        VALUES (V_ID, V_LAST_NAME);
        COMMIT;
    END;
```

This procedure must invoke the **APD\_BAT\_STAT** procedure and pass a parameter.

Which statement, when added to the above procedure will successfully invoke the **UPD\_BAT\_STAT** procedure?

- A. EXECUTE UPD\_BAT\_STAT(V\_ID);
- B. UPD\_BAT\_STAT(V\_ID);
- C. RUN UPD\_BAT\_STAT(V\_ID);
- D. START UPD\_BAT\_STAT(V\_ID);

Answer: B

**QUESTION NO: 29**

Which statement about triggers is true?

- A. You use an application trigger to fire when a DELETE statement occurs.
- B. You use a database trigger to fire when an INSERT statement occurs.
- C. You use a system event trigger to fire when an UPDATE statement occurs.
- D. You use INSTEAD OF trigger to fire when a SELECT statement occurs.

**Answer: B**

**QUESTION NO: 30**

You want to create a PL/SQL block of code that calculates discounts on customer orders. –This code will be invoked from several places, but only within the program unit ORDERTOTAL.

**What is the most appropriate location to store the code that calculates the discounts?**

- A. A stored procedure on the server.
- B. A block of code in a PL/SQL library.
- C. A standalone procedure on the client machine.
- D. A block of code in the body of the program unit ORDERTOTAL.
- E. A local subprogram defined within the program unit ORDERTOTAL.

**Answer: E**

**QUESTION NO: 31**

**Which type of argument passes a value from a procedure to the calling environment?**

- A. VARCHAR2
- B. BOOLEAN
- C. OUT
- D. IN

**Answer: C**

**QUESTION NO: 32**

**You create a DML trigger. For the timing information, which is valid with a DML trigger?**

- A. DURING
- B. INSTEAD
- C. ON SHUTDOWN
- D. BEFORE
- E. ON STATEMENT EXECUTION

**Answer: D**

**QUESTION NO: 33**

You are about to change the arguments of the `CALC_TEAM_AVG` function.

Which dictionary view can you query to determine the names of the procedures and functions that invoke the `CALC_TEAM_AVG` function?

- A. `USER_PROC_DEPENDS`
- B. `USER_DEPENDENCIES`
- C. `USER_REFERENCES`
- D. `USER_SOURCE`

**Answer: B**

**QUESTION NO: 34**

A `CALL` statement inside the trigger body enables you to call \_\_\_\_\_.

- A. A package.
- B. A stored function.
- C. A stored procedure.
- D. Another database trigger.

**Answer: C**

**Incorrect Answers:**

- A. Package can't be called, we call a procedure inside the package.
- B. We can't call a function use `CALL` statement because function must return a value.
- D. Trigger can't be called, they are execute automatically when the trigger event occure.

**QUESTION NO: 35**

You need to remove the database trigger `BUSINESS_HOUR`.

Which command do you use to remove the trigger in the SQL \*Plus environment?

- A. `DROP TRIGGER business_hour;`
- B. `DELETE TRIGGER business_hour;`
- C. `REMOVE TRIGGER business_hour;`
- D. `ALTER TRIGGER business_hour REMOVE;`
- E. `DELETE FROM USER_TRIGGERS`  
`WHERE TRIGGER_NAME = 'BUSINESS_HOUR';`

**Answer: A**

**QUESTION NO: 36**

**How can you migrate from a LONG to a LOB data type for a column?**

- A. Use the DBMS\_MANAGE\_LOB.MIGRATE procedure.
- B. Use the UTL\_MANAGE\_LOB.MIGRATE procedure.
- C. Use the DBMS\_LOB.MIGRATE procedure.
- D. Use the ALTER TABLE command.
- E. You cannot migrate from a LONG to a LOB data type for a column.

**Answer: D**

**QUESTION NO: 37**

**Examine this procedure:**

```
CREATE OR REPLACE PROCEDURE INSERT_TEAM
(V_ID in NUMBER, V_CITY in VARCHAR2 DEFAULT 'AUSTIN', V_NAME in
VARCHAR2)
IS
BEGIN
INSERT INTO TEAM (id, city, name)
VALUES (v_id, v_city, v_name);
COMMIT;
END
```

**Which two statements will successfully invoke this procedure in SQL \*Plus? (Choose two)**

- A. EXECUTE INSERT\_TEAM;
- B. EXECUTE INSERT\_TEAM(3, V\_NAME=>'LONGHORNS', V\_CITY=>'AUSTIN');
- C. EXECUTE INSERT\_TEAM(3, 'AUSTIN', 'LONGHORNS');
- D. EXECUTE INSERT\_TEAM (V\_ID := V\_NAME := 'LONGHORNS', V\_CITY := 'AUSTIN');
- E. EXECUTE INSERT\_TEAM (3, 'LONGHORNS');

**Answer: B,C**

**QUESTION NO: 38**

**To be callable from a SQL expression, a user-defined function must do what?**

- A. Be stored only in the database.
- B. Have both IN and OUT parameters.
- C. Use the positional notation for parameters.
- D. Return a BOOLEAN or VARCHAR2 data type.

**Answer: A**

**QUESTION NO: 39**

**Which two describe a stored procedure? (Choose two)**

- A. A stored procedure is typically written in SQL.
- B. A stored procedure is a named PL/SQL block that can accept parameters.
- C. A stored procedure is a type of PL/SQL subprogram that performs an action.
- D. A stored procedure has three parts: the specification, the body, and the exception handler part.
- E. The executable section of a stored procedure contains statements that assigns values, control execution, and return values to the calling environment.

**Answer: B,C**

A procedure is a named PL/SQL block that can accept parameters (sometimes referred to as arguments), and be invoked. Generally speaking, you use a procedure to perform an action. A procedure has a header, a declaration section, an executable section, and an optional exception-handling section.

A procedure can be compiled and stored in the database as a schema object.

Procedures promote reusability and maintainability. When validated, they can be used in any number of applications. If the requirements change, only the procedure needs to be updated.

**QUESTION NO: 40**

**Examine this code:**

```
CREATE OR REPLACE PROCEDURE add_dept
  (p_name departments.department_name%TYPE DEFAULT 'unknown',
  p_loc departments.location_id%TYPE DEFAULT 1700)
IS
BEGIN
  INSERT INTO departments(department_id, department_name,
  loclation_id)
  VALUES(dept_seq.NEXTVAL,p_name, p_loc);
END add_dept;
/
```

You created the add\_dept procedure above, and you now invoke the procedure in SQL \*Plus. Which four are valid invocations? (Choose four)

- A. EXECUTE add\_dept(p\_loc=>2500)
- B. EXECUTE add\_dept('Education', 2500)
- C. EXECUTE add\_dept('2500', p\_loc =>2500)
- D. EXECUTE add\_dept(p\_name=>'Education', 2500)
- E. EXECUTE add\_dept(p\_loc=>2500, p\_name=>'Education')

Answer: A,B,C,E

**QUESTION NO: 41**

Which three are valid ways to minimize dependency failure? (Choose three)

- A. Querying with the SELECT \* notification.
- B. Declaring variables with the %TYPE attribute.
- C. Specifying schema names when referencing objects.
- D. Declaring records by using the %ROWTYPE attribute.
- E. Specifying package.procedure notation while executing procedures.

Answer: A,B,D

**QUESTION NO: 42**

Which two does the INSTEAD OF clause in a trigger identify? (Choose two)

- A. The view associated with the trigger.
- B. The table associated with the trigger.
- C. The event associated with the trigger.
- D. The package associated with the trigger.
- E. The statement level or for each row association to the trigger.

Answer: A,C

**QUESTION NO: 43**

Examine this package:

```
CREATE OR REPLACE PACKAGE manage_emps
IS
    tax_rate CONSTANT NUMBER(5,2) := .28;
```

```

v_id NUMBER;
PROCEDURE insert_emp (p_deptno NUMBER, p_sal NUMBER);
PROCEDURE delete_emp;
PROCEDURE update_emp;
FUNCTION calc_tax (p_sal NUMBER)
    RETURN NUMBER;
END manage_emps;
/
CREATE OR REPLACE PACKAGE BODY manage_emps
IS
    PROCEDURE update_sal
        (p_raise_amt NUMBER)
    IS
    BEGIN
        UPDATE emp
            SET sal = (sal * p_raise_amt) + sal
            WHERE empno = v_id;
        END;
    PROCEDURE insert_emp
        (p_deptno NUMBER, p_sal NUMBER)
    IS
    BEGIN
        INSERT INTO emp(empno, deptno, sal)
            VALYES(v_id, p_deptno, p_sal);
        END insert_emp;
    PROCEDURE delete_emp
    IS
    BEGIN
        DELETE FROM emp
            WHERE empno = v_id;
        END delete_emp;
    PROCEDURE update_emp
    IS
        v_sal NUMBER(10, 2);
        v_raise NUMBER(10, 2);
    BEGIN
        SELECT sal
            INTO v_sal
            FROM emp
            WHERE empno = v_id;
        IF v_sal < 500 THEN
            v_raise := .05;
        ELSIF v_sal < 1000 THEN
            v_raise := .07;
        ELSE
            v_raise := .04;
        END IF;
        update_sal(v_raise);
    END update_emp;

```

```
FUNCTION calc_tax
  (p_sal NUMBER)
  RETURN NUMBER
  IS
  BEGIN
    RETURN p_sal * tax_rate;
  END calc_tax;
END manage_emps;
/
```

**What is the name of the private procedure in this package?**

- A. CALC\_TAX
- B. INSERT\_EMP
- C. UPDATE\_SAL
- D. DELETE\_EMP
- E. UPDATE\_EMP
- F. MANAGE\_EMPS

**Answer: C**

**QUESTION NO: 44**

**What can you do with the DBMS\_LOB package?**

- A. Use the DBMS\_LOB.WRITE procedure to write data to a BFILE.
- B. Use the DBMS\_LOB.BFILENAME function to locate an external BFILE.
- C. Use the DBMS\_LOB.FILEEXISTS function to find the location of a BFILE.
- D. Use the DBMS\_LOB.FILECLOSE procedure to close the file being accessed.

**Answer: D**

**See next page**

APPEND	Append the contents of the source LOB to the destination LOB
COPY	Copy all or part of the source LOB to the destination LOB
ERASE	Erase all or part of a LOB
LOADFROMFILE	Load BFILE data into an internal LOB
TRIM	Trim the LOB value to a specified shorter length
WRITE	Write data to the LOB from a specified offset
GETLENGTH	Get the length of the LOB value
INSTR	Return the matching position of the <i>n</i> th occurrence of the pattern in the LOB
READ	Read data from the LOB starting at the specified offset
SUBSTR	Return part of the LOB value starting at the specified offset
FILECLOSE	Close the file
FILECLOSEALL	Close all previously opened files
FILEEXISTS	Check if the file exists on the server
FILEGETNAME	Get the directory alias and file name
FILEISOPEN	Check if the file was opened using the input BFILE locators
FILEOPEN	Open a file

**Incorrect Answers:**

**A.** DBMS\_LOB.WRITE is used to write to Internal LOBs.

The internal LOB is stored inside the Oracle server. A BLOB, NCLOB, or CLOB can be one of the following:

- An attribute of a user-defined type
- A column in a table
- A bind or host variable
- A PL/SQL variable, parameter, or result

Internal LOBs can take advantage of Oracle features such as:

- Concurrency mechanisms
- Redo logging and recovery mechanisms
- Transactions with commit or rollbacks

**B.** BFILENAME is a built-in function that initializes a BFILE column to point to an external file. Use the BFILENAME function as part of an INSERT statement to initialize a BFILE column by associating it with a physical file in the server file system. You can use the UPDATE statement to change the reference target of the BFILE. A BFILE can be initialized to NULL and updated later by using the BFILENAME function.

**C.** DBMS\_LOB.FILEEXISTS function to find if the file exists on the server

**QUESTION NO: 45**

**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
  V_PLAYER_AVG    NUMBER(4,3);
  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;
    VALIDATE_PLAYER_STAT(V_ID);
  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 .333 to the V\_PLAYER\_AVG variable from a procedure outside the package?**

- A. V\_PLAYER\_AVG := .333;
- B. BB\_PACK.UPD\_PLAYER\_STAT.V\_PLAYER\_AVG := .333;
- C. BB\_PACK.V\_PLAYER\_AVG := .333;
- D. This variable cannot be assigned a value from outside of the package.

**Answer: D**

**QUESTION NO: 46**

**Examine this code:**

```
CREATE OR REPLACE PACKAGE comm_package
```

```

IS
  g_comm NUMBER := 10;
  PROCEDURE reset_comm(p_comm IN NUMBER);
END comm_package;
/

```

**User Jones executes the following code at 9:01am:**

```
EXECUTE comm_package.g_comm := 15
```

**User Smith executes the following code at 9:05am:**

```
EXECUTE comm_package.g_comm := 20
```

**Which statement is true?**

- A. g\_comm has a value of 15 at 9:06am for Smith.
- B. g\_comm has a value of 15 at 9:06am for Jones.
- C. g\_comm has a value of 20 at 9:06am for both Jones and Smith.
- D. g\_comm has a value of 15 at 9:03 am for both Jones and Smith.
- E. g\_comm has a value of 10 at 9:06am for both Jones and Smith.
- F. g\_comm has a value of 10 at 9:03am for both Jones and Smith

**Answer: B**

**QUESTION NO: 47**

**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;

```

**Which statement removes the function?**

- A. DROP FUNCTION gen\_email\_name;
- B. REMOVE gen\_email\_name;

- C. DELETE gen\_email\_name;
- D. \*\*\*MISSING\*\*\*

**Answer: A**

**QUESTION NO: 48**

**Examine this procedure:**

```
CREATE OR REPLACE PROCEDURE UPD_BAT_STAT
(V_ID IN NUMBER DEFAULT 10, V_AB IN NUMBER DEFAULT 4)
IS
BEGIN
    UPDATE PLAYER_BAT_STAT
    SET AT_BATS = AT_BATS + V_AB
    WHERE PLAYER_ID = V_ID;
    COMMIT;
END;
```

**Which two statements will successfully invoke this procedure in SQL \*Plus? (Choose two)**

- A. EXECUTE UPD\_BAT\_STAT;
- B. EXECUTE UPD\_BAT\_STAT(V\_AB=>10, V\_ID=>31);
- C. EXECUTE UPD\_BAT\_STAT(31, 'FOUR', 'TWO');
- D. UPD\_BAT\_STAT(V\_AB=>10, V\_ID=>31);
- E. RUN UPD\_BAT\_STAT;

**Answer: A,B**

**QUESTION NO: 49**

**Examine this code:**

```
CREATE OR REPLACE PROCEDURE audit_action
(p_who VARCHAR2)
AS
BEGIN
    INSERT INTO audit(schema_user) VALUES(p_who);
END audit_action;
/
```

```
CREATE OR REPLACE TRIGGER watch_it
AFTER LOGON ON DATABASE
CALL audit_action(ora_login_user)
```

/

**What does this trigger do?**

- A. The trigger records an audit trail when a user makes changes to the database.
- B. The trigger marks the user as logged on to the database before an audit statement is issued.
- C. The trigger invoked the procedure audit\_action each time a user logs on to his/her schema and adds the username to the audit table.
- D. The trigger invokes the procedure audit\_action each time a user logs on to the database and adds the username to the audit table.

**Answer: D**

**QUESTION NO: 50**

**Which view displays indirect dependencies, indenting each dependency?**

- A. DEPTREE
- B. IDEPTREE
- C. INDENT\_TREE
- D. I\_DEPT\_TREE

**Answer: B**

**QUESTION NO: 51**

**The OLD and NEW qualifiers can be used in which type of trigger?**

- A. Row level DML trigger
- B. Row level system trigger
- C. Statement level DML trigger
- D. Row level application trigger
- E. Statement level system trigger
- F. Statement level application trigger

**Answer: A**

**QUESTION NO: 52**

**Which statement is true?**

- A. Stored functions can be called from the SELECT and WHERE clauses only.
- B. Stored functions do not permit calculations that involve database links in a distributed environment.
- C. Stored functions cannot manipulate new types of data, such as longitude and latitude.
- D. Stored functions can increase the efficiency of queries by performing functions in the query rather than in the application.

**Answer: D**

**QUESTION NO: 53**

**Examine the trigger:**

```
CREATE OR REPLACE TRIGGER Emp_count
AFTER DELETE ON Emp_tab
FOR EACH ROW
DECLARE
  n INTEGER;
BEGIN
  SELECT COUNT(*)
  INTO n
  FROM Emp_tab;
  DBMS_OUTPUT.PUT_LINE(' There are now ' || a ||
    ' employees,');
END;
```

**This trigger results in an error after this SQL statement is entered:**

```
DELETE FROM Emp_tab WHERE Empno = 7499;
```

**How do you correct the error?**

- A. Change the trigger type to a BEFORE DELETE.
- B. Take out the COUNT function because it is not allowed in a trigger.
- C. Remove the DBMS\_OUTPUT statement because it is not allowed in a trigger.
- D. Change the trigger to a statement-level trigger by removing FOR EACH ROW.

**Answer: D**

**QUESTION NO: 54**

**What is true about stored procedures?**

- A. A stored procedure uses the DELCLARE keyword in the procedure specification to declare formal parameters.

- B. A stored procedure is named PL/SQL block with at least one parameter declaration in the procedure specification.
- C. A stored procedure must have at least one executable statement in the procedure body.
- D. A stored procedure uses the DECLARE keyword in the procedure body to declare formal parameters.

**Answer: C**

**QUESTION NO: 55**

**Examine this code:**

```
CREATE OR REPLACE PROCEDURE insert_dept
  (p_location_id NUMBER)
IS
  v_dept_id NUMBER(4);
BEGIN
  INSERT INTO departments
  VALUES (5, 'Education', 150, p_location_id);

  SELECT department_id
  INTO v_dept_id
  FROM employees
  WHERE employee_id=99999;
END insert_dept;
/
```

```
CREATE OR REPLACE PROCEDURE insert_location
  (p_location_id NUMBER,
   p_city VARCHAR2)
IS
BEGIN
  INSERT INTO locations(location_id, city)
  VALUES (p_location_id, p_city);

  insert_dept(p_location_id);
END insert_location;
/
```

**You just created the departments, the locations, and the employees table. You did not insert any rows. Next you created both procedures.**

**You now invoke the insert\_location procedure using the following command:**

```
EXECUTE insert_location (19, 'San Francisco')
```

**What is the result in this EXECUTE command?**

- A. The locations, departments, and employees tables are empty.
- B. The departments table has one row.  
The locations and the employees tables are empty.
- C. The location table has one row.  
The departments and the employees tables are empty.
- D. The locations table and the departments table both have one row.  
The employees table is empty.

**Answer: A**

**QUESTION NO: 56**

**The creation of which four database objects will cause a DDL trigger to fire? (Choose four)**

- A. Index
- B. Cluster
- C. Package
- D. Function
- E. Synonyms
- F. Dimensions
- G. Database links

**Answer: A,,D,C,E**

**QUESTION NO: 57**

**Which two program declarations are correct for a stored program unit? (Choose two)**

- A. CREATE OR REPLACE FUNCTION tax\_amt  
(p\_id NUMBER)  
RETURN NUMBER
- B. CREATE OR REPLACE PROCEDURE tax\_amt  
(p\_id NUMBER)  
RETURN NUMBER
- C. CREATE OR REPLACE PROCEDURE tax\_amt  
(p\_id NUMBER, p\_amount OUT NUMBER)
- D. CREATE OR REPLACE FUNCTION tax\_amt  
(p\_id NUMBER)  
RETURN NUMBER(10,2)

E. CREATE OR REPLACE PROCEDURE tax\_amt  
(p\_id NUMBER, p\_amount OUT NUMBER(10, 2))

**Answer: A,C**

**QUESTION NO: 58**

**You need to implement a virtual private database (vpd). In order to have the vpd functionality, a trigger is required to fire when every user initiates a session in the database. What type of trigger needs to be created?**

- A. DML trigger
- B. System event trigger
- C. INSTEAD OF trigger
- D. Application trigger

**Answer: B**

**QUESTION NO: 59**

**You have a row level BEFORE UPDATE trigger on the EMP table. This trigger contains a SELECT statement on the EMP table to ensure that the new salary value falls within the minimum and maximum salary for a given job title.**

**What happens when you try to update a salary value in the EMP table?**

- A. The trigger fires successfully.
- B. The trigger fails because it needs to be a row level AFTER UPDATE trigger.
- C. The trigger fails because a SELECT statement on the table being updated is not allowed.
- D. The trigger fails because you cannot use the minimum and maximum functions in a BEFORE UPDATE trigger.

**Answer: C**

**QUESTION NO: 60**

**Examine this code:**

```
CREATE OR REPLACE STORED FUNCTION get_sal  
(p_raise_amt NUMBER, p_employee_id employees.employee_id%TYPE)  
RETURN NUMBER
```

```

IS
v_salary NUMBER;
v_raise  NUMBER(8,2);
BEGIN
  SELECT salary
     INTO   v_salary
     FROM   employees
     WHERE  employee_id = p_employee_id;
  v_raise := p_raise_amt * v_salary;
  RETURN v_raise;
END;

```

**Which statement is true?**

- A. This statement creates a stored procedure named get\_sal.
- B. This statement returns a raise amount based on an employee id.
- C. This statement creates a stored function named get\_sal with a status of invalid.
- D. This statement creates a stored function named get\_sal.
- E. This statement fails.

**Answer: E**

**QUESTION NO: 61**

**You need to disable all triggers on the EMPLOYEES table.**

**Which command accomplishes this?**

- A. None of these commands; you cannot disable multiple triggers on a table in one command.
- B. ALTER TRIGGERS ON TABLE employees DISABLE;
- C. ALTER employees DISABLE ALL TRIGGERS;
- D. ALTER TABLE employees DISABLE ALL TRIGGERS;

**Answer: D**

**QUESTION NO: 62**

**An internal LOB is \_\_\_\_.**

- A. A table.
- B. A column that is a primary key.
- C. Stored in the database.
- D. A file stored outside of the database, with an internal pointer to it from a database column.

**Answer: C**

**QUESTION NO: 63**

**Examine this code:**

```
CREATE OR REPLACE FUNCTION calc_sal(p_salary NUMBER)
RETURN NUMBER
IS
    v_raise NUMBER(4,2) DEFAULT 1.08;
BEGIN
    RETURN v_raise * p_salary;
END calc_sal;
/
```

**Which statement accurately call the stored function CALC\_SAL? (Choose two)**

- A. UPDATE employees (calc\_sal(salary))  
SET salary = salary \* calc\_sal(salary);
- B. INSERT calc\_sal(salary) INTO employees  
WHERE department\_id = 60;
- C. DELETE FROM employees(calc\_sal(salary))  
WHERE calc\_sal(salary) > 1000;
- D. SELECT salary, calc\_sal(salary)  
FROM employees  
WHERE department\_id = 60;
- E. SELECT last\_name, salary, calc\_sal(salary)  
FROM employees ORDER BY  
calc\_sal(salary);

**Answer: D,E**

**QUESTION NO: 64**

**This statement fails when executed:**

```
CREATE OR REPLACE TRIGGER CALC_TEAM_AVG
AFTER INSERT ON PLAYER
BEGIN

INSERT INTO PLAYER_BATSTAT (PLAYER_ID, SEASON_YEAR,AT_BATS,HITS)
VALUES (:NEW.ID, 1997, 0,0);
END;
```

**To which type must you convert the trigger to correct the error?**

- A. Row
- B. Statement
- C. ORACLE FORM trigger
- D. Before

**Answer: A**

**QUESTION NO: 65**

**Examine this code:**

```
CREATE OR REPLACE PROCEDURE audit_emp
  (p_id IN emp_empno%TYPE)
IS
  v_id NUMBER;
  PROCEDURE log_exec
  IS
  BEGIN
    INSERT INTO log_table (user_id, log_delete)
      VALUES (USER, SYSDATE);
  END log_exec;
  v_name VARCHAR2(20);
BEGIN
  DELETE FROM emp
    WHERE empno = p_id;
  log_exec;
  SELECT ename, empno
    INTO v_name, v_id
    FROM emp
    WHERE empno = p_id;
END audit_emp;
```

**Why does this code cause an error when compiled?**

- A. An statement is not allowed in a subprogram declaration.
- B. Procedure LOG\_EXEC should be declared before any identifiers.
- C. Variable v\_name should be declared before declaring the LOG\_EXEC procedure.
- D. The LOG\_EXEC procedure should be invoked as EXECUTE log\_exec with the AUDIT\_EMP procedure.

**Answer: C**

**QUESTION NO: 66****Examine this code:**

```

CREATE OR REPLACE PACKAGE metric_converter
IS
  c_height CONSTRAINT NUMBER := 2.54;
  c_weight CONSTRAINT NUMBER := .454;
  FUNCTION calc_height (p_height_in_inches NUMBER)
    RETURN NUMBER;
  FUNCTION calc_weight (p_weight_in_pounds NUMBER)
    RETURN NUMBER;
END;
/

CREATE OR REPLACE PACKAGE BODY metric_converter
IS
  FUNCTION calc_height (p_height_in_inches NUMBER)
    RETURN NUMBER
  IS
  BEGIN
    RETURN p_height_in_inches * c_height;
  END calc_height;
  FUNCTION calc_weight (p_weight_in_pounds NUMBER)
    RETURN NUMBER
  IS
  BEGIN
    RETURN p_weight_in_pounds * c_weight
  END calc_weight
END metric_converter;
/

CREATE OR REPLACE FUNCTION calc_height (p_height_in_inches NUMBER)
  RETURN NUMBER
  IS
  BEGIN
    RETURN p_height_in_inches * metric_converter.c_height;
  END calc_height;
/

```

**Which statement is true?**

- A. If you remove the package specification, then the package body and the stand alone stored function CALC\_HEIGHT are removed.
- B. If you remove the package body, then the package specification and the stand alone stored function CALC\_HEIGHT are removed.
- C. If you remove the package specification, then the package body is removed.
- D. If you remove the package body, then the package specification is removed.

- E. If you remove the stand alone stored function CALC\_HEIGHT, then the METRIC\_CONVERTER package body and the package specification are removed.
- F. The stand alone function CALC\_HEIGHT cannot be created because its name is used in a packaged function.

**Answer: C**

**QUESTION NO: 67**

**What is a condition predicate in a DML trigger?**

- A. A conditional predicate allows you to specify a WHEN-LOGGING-ON condition in the trigger body.
- B. A conditional predicate means you use the NEW and OLD qualifiers in the trigger body as a condition.
- C. A conditional predicate allows you to combine several DBM triggering events into one in the trigger body.
- D. A conditional predicate allows you to specify a SHUTDOWN or STARTUP condition in the trigger body.

**Answer: C**

**QUESTION NO: 68**

**Examine this package specification:**

```
CREATE OR REPLACE PACKAGE concat_all
IS
  v_string VARCHAR2(100);
  PROCEDURE combine (p_num_val NUMBER);
  PROCEDURE combine (p_date_val DATE);
  PROCEDURE combine (p_char_val VARCHAR2, p_num_val NUMBER);
END concat_all;
/
```

**Which overloaded COMBINE procedure declaration can be added to this package specification?**

- A. PROCEDURE combine;
- B. PROCEDURE combine (p\_no NUMBER);
- C. PROCEDURE combine (p\_val\_1 VARCHAR2, p\_val\_2 NUMBER);
- D. PROCEDURE concat\_all (p\_num\_val VARCHAR2, p\_char\_val NUMBER);

**Answer: A**

**QUESTION NO: 69**

**Local procedure A calls remote procedure B. Procedure B was compiled at 8 A.M. Procedure A was modified and recompiled at 9 A.M. Remote procedure B was later modified and recompiled at 11 A.M.**

**The dependency mode is set to `TIMESTAMP`.**

**What happens when procedure A is invoked at 1 P.M.?**

- A. There is no affect on procedure A and it runs successfully.
- B. Procedure B is invalidated and recompiles when invoked.
- C. Procedure A is invalidated and recompiles for the first time it is invoked.
- D. Procedure A is invalidated and recompiles for the second time it is invoked.

**Answer: D**

When the local procedure is invoked, at run time the Oracle server compares the two time stamps of the referenced remote procedure. If the time stamps are equal (indicating that the remote procedure has not recompiled), the Oracle server executes the local procedure. If the time stamps are not equal (indicating that the remote procedure has recompiled), the Oracle server invalidates the local procedure and returns a runtime error.

If the local procedure, which is now tagged as invalid, is invoked a second time, the Oracle server recompiles it before executing, in accordance with the automatic local dependency mechanism. So if a local procedure returns a run-time error the first time that it is invoked, indicating that the remote procedure's time stamp has changed, you should develop a strategy to re-invoke the local procedure.

**Incorrect Answers:**

**A, B, C**

**QUESTION NO: 70**

**Under which two circumstances do you design database triggers? (Choose two)**

- A. To duplicate the functionality of other triggers.
- B. To replicate built-in constraints in the Oracle server such as primary key and foreign key.
- C. To guarantee that when a specific operation is performed, related actions are performed.
- D. For centralized, global operations that should be fired for the triggering statement, regardless of which user or application issues the statement.

**Answer: C,D**

**QUESTION NO: 71**

**Examine this procedure:**

```
CREATE OR REPLACE PROCEDURE DELETE_PLAYER
(V_ID IN NUMBER)
IS
BEGIN
    DELETE FROM PLAYER
    WHERE ID = V_ID;
EXCEPTION
    WHEN STATS_EXISTS_EXCEPTION
    THEN DBMS_OUTPUT.PUT_LINE
    ('Cannot delete this player, child records exist in PLAYER_BAT_STAT
    table');
END;
```

**What prevents this procedure from being created successfully?**

- A. A comma has been left after the STATS\_EXISTS\_EXCEPTION exception.
- B. The STATS\_EXISTS\_EXCEPTION has not been declared as a number.
- C. The STATS\_EXISTS\_EXCEPTION has not been declared as an exception.
- D. Only predefined exceptions are allowed in the EXCEPTION section.

**Answer: C**