

PassLeader

PassLeader

> Contact Us Login / Register Search...

HOME

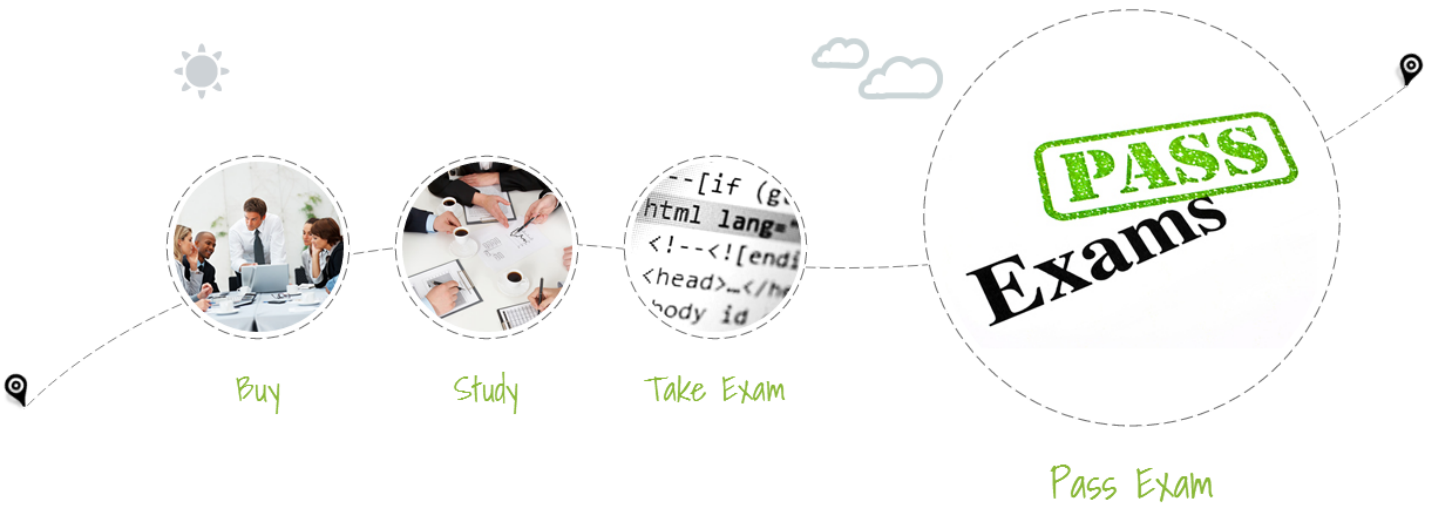
ALL VENDORS

★ GUARANTEE

? FAQ

TESTIMONIALS

CART (1)



Try **PDF Demo** before you buy

We're not the only ones **happy** about PassLeader Practice Material ...

63159+ customers in 100+ countries use PassLeader Test Engine. Meet our customers.

VOREED

GetCustom

JET ORANGE

iCompany

Paradoxx

iMessenger



<http://www.passleader.top/>

Latest Exam Guide & Learning Materials

Exam : 1z1-144 日本語 (JPN)

Title : Oracle Database 11g:
Program with PL/SQL

Vendor : Oracle

Version : DEMO

NO.1 どの2つのステートメントは内部的に定義またはユーザー定義PL7SQL例外の取り扱いについて真ですか。(2つを選んでください。)

- A. エラーが発生するたびに例外ハンドラを追加します。
- B. 例外ハンドラはトランザクションをコミットする必要があります。
- C. 可能な限りの代わりに、例外ハンドラにするととき他の人を使用して、指定された例外を処理します。
- D. あなたの PL/SQL ブロックに例外ハンドラを追加する代わりに、彼らが発生する可能性があるすべての点でエラーをチェックします。

Answer: C,D

NO.2 図表を見て、実行する予定のコードのブロックを検査します。どのステートメントはコードのブロックに関して真ですか。

```
CREATE OR REPLACE FUNCTION df1t RETURN NUMBER IS
  cnt NUMBER := 0;
BEGIN
  cnt := cnt + 1;
  RETURN 45;
END df1t;

CREATE OR REPLACE PROCEDURE p(i IN NUMBER DEFAULT df1t()) IS
BEGIN
  DBMS_OUTPUT.PUT_LINE(1);
END p;

DECLARE
  cnt NUMBER := df1t();
BEGIN
  FOR j IN 1..3 LOOP
    p();
  END LOOP;
  DBMS_OUTPUT.PUT_LINE('cnt: '||cnt);
  p();
  DBMS_OUTPUT.PUT_LINE('cnt: '||cnt);
END;
```

- A. すべてのブロックが正常に実行され、無名ブロックは 1 2 3 cnt: 45 45 cnt: 45 を表示されます。
- B. すべてのブロックが正常に実行され、無名ブロックは 1 2 3 cut: 0 45 cart: 1 を表示されま
- C. ライン 2 の関数呼び出しが有効ではありませんので、無名ブロックはエラーが発生しま
- D. ライン 1 の関数呼び出しが有効ではありませんので、プログラムの作成がエラーにな

Answer: A

NO.3 どのステートメントは正しく関数およびプログラムの機能について説明しますか。

(該当するものをすべて選択してください。)

- A. プロシージャは値のない return 文を含めることができます。
- B. 関数は単一の return 節を使用して複数の値を返すことができます。
- C. プロシージャは SQL 式の一部として、または PL/SQL 文として実行することができます。
- D. 関数は呼び出し元の環境から転送されたゼロ以上のパラメータを含めることができます。

Answer: A

Reference: http://docs.oracle.com/cd/B19306_01/appdev.102/b14261/subprograms.htm (using the return statement)

NO.4 PL/SQL コードの次のスニペットを検査してください:

```

DECLARE
  emp_job      employees.job_id%TYPE := 'ST_CLERK';
  emp_salary   employees.salary%TYPE := 3000;
  my_record    employees%ROWTYPE;
  CURSOR c1 (job VARCHAR2, max_wage NUMBER) IS
    SELECT * FROM employees
      WHERE job_id = job
        AND salary > max_wage;
BEGIN
  .....
```

EMPLOYEES 表の表の説明のための図表を見ます。EMPLOYEES 表には 200 行があります。

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER (6)
FIRST_NAME		VARCHAR2 (20)
LAST_NAME	NOT NULL	VARCHAR2 (25)
EMAIL	NOT NULL	VARCHAR2 (25)
PHONE_NUMBER		VARCHAR2 (20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2 (10)
SALARY		NUMBER (8, 2)
COMMISSION_PCT		NUMBER (2, 2)
MANAGER_ID		NUMBER (6)
DEPARTMENT_ID		NUMBER (4)

「ST_CLERK」として JOB_ID を持つ従業員と 3000 より大きい給料から成る結果をフェッチするためのカーソルを開く OPEN 文を識別してください。

- A. OPEN c1 (NULL, 3000);
- B. OPEN c1 (emp_job, 3000);
- C. OPEN c1 (3000, emp_salary);
- D. OPEN c1 ('ST_CLERK', 3000)

E. OPEN c1 (EMP_job, emp_salary);

Answer: D

NO.5 どの2つのステートメントは PL/SQL パッケージのコンポーネントについて正しいですか。(2つを選んでください。)

- A. パッケージには仕様部と本体の両方を持っていなければなりません。
- B. パッケージ本体はパッケージ仕様なしで存在することができます。
- C. パッケージ仕様はパッケージ本体なしで存在することができます。
- D. パッケージ化されたパブリック変数がセッションで最初に呼び出されるとき、パッケージ全体がメモリーにロードされます。

Answer: C,D

NO.6 次の表を作成し、次のコードを実行します:

```
SQL>CREATE TABLE emp_temp (deptno NUMBER(2), job VARCHAR2(18));
SQL>DECLARE
    TYPE NumList IS TABLE OF NUMBER;
    depts NumList := NumList(10, 20, 30);
BEGIN
    INSERT INTO emp_temp VALUES(10, 'Clerk');
    INSERT INTO emp_temp VALUES(20, 'Bookkeeper');
    INSERT INTO emp_temp VALUES(30, 'Analyst');
    FORALL j IN depts.FIRST..depts.LAST
        UPDATE emp_temp SET job = job || ' (Senior)'
            WHERE deptno = depts(j);
    EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE ('Problem in the FORALL statement. ');
    COMMIT;
END;
/
```

どのステートメントは上記のコードの結果について真ですか。

- A. 正常に実行し、すべての行が更新されます。
- B. エラーになりますが、挿入された行と最初の行への更新を保存します。
- C. エラーになりますが、挿入された行を保存します。しかし、行は更新されません。
- D. エラーになり、すべてのデータ操作言語 (DML) 文はロールバックされます。

Answer: A

NO.7 次の PL/SQL コードを検査してください:

```
DECLARE
  CURSOR c_emp_cursor IS
    SELECT employee_id, last_name FROM employees
    WHERE department_id =30;
BEGIN
  FOR emp_record IN c_emp_cursor
  LOOP
    DEMS_OUTPUT.PUT_LINE( emp_record.employee_id||' ' ||emp_record.last_name);
  END LOOP;
END;
```

サーバー出力はセッションのためにオンになっています。どのステートメントはコードの実行に関して真ですか。

- A. コードが正常に実行され、所望の出力を提供します。
- B. EMP_RECORD 変数が宣言されていないため、コードはエラーを生成します。
- C. カーソルが FOR ループの前に開かれていないため、コードはエラーを生成します。
- D. ループが終了したときに句を持っていないため、コードはエラーを生成します。

Answer: A

NO.8 PL/SQL コードを検査するために図表を見ます。

```

DECLARE
  jobid employees.job_id%TYPE;
  empid employees.employee_id%TYPE := 115;
  sal employees.salary%TYPE;
  sal_raise NUMBER(3,2);
BEGIN
  SELECT job_id, salary INTO jobid, sal from employees
  WHERE employee_id = empid;
  CASE
    WHEN jobid = 'PU_CLERK' THEN
      IF sal < 3000 THEN sal_raise := .12;
      ELSE sal_raise := .09;
      END IF;
    WHEN jobid = 'SH_CLERK' THEN
      IF sal < 4000 THEN sal_raise := .11;
      ELSE sal_raise := .08;
      END IF;
    WHEN jobid = 'ST_CLERK' THEN
      IF sal < 3500 THEN sal_raise := .10;
      ELSE sal_raise := .07;
      END IF;
    ELSE
      BEGIN
        DBMS_OUTPUT.PUT_LINE('No raise for this job: ' || jobid);
      END;
  END CASE;
  UPDATE employees SET salary = salary + salary * sal_raise
  WHERE employee_id = empid;
  COMMIT;
END;

```

SERVEROUTPUT はセッションのためにオンになっています。

どのステートメントはコードの実行に関して真ですか。

- A. 見当違い else 節という原因で、実行が失敗しました。
- B. EMPLOYEE_ID115 との従業員が存在しない場合でも、実行が成功します
- C. EMPLOYEE_ID たちとの従業員が見つからない場合、実行が低下し、例外をスローします。
- D. EMPLOYEE_ID115 との従業員が見つからない場合、実行が成功しますが、それは誤った出力が表示されます。

Answer: C

NO.9 行が顧客テーブルから削除され、それがテーブルに残っている行数を表示しているたびに、起動するトリガーを作成します。

トリガーは上記の要件のために作成されようについて、どの 2 つのステートメントは正しいですか。(2 つを選んでください。)

- A. それはトリガーの後でなければなりません。
- B. それはトリガーの前でなければなりません。
- C. それは行レベルのトリガーでなければなりません。
- D. それは文レベルのトリガーでなければなりません。
- E. それはトリガーの後または前でできます。

Answer: A,C

NO.10 どの 2 つのタスクがプロシージャとしての代わりに機能として作成する必要がありますか。(2 つを選んでください。)

- A. コードの PL7SQL ブロックの参照ホストまたはバインド変数
- B. 呼び出し元の環境に複数の値を計算し、返すタスク
- C. 呼び出し環境に戻す必要がある値を計算するタスク
- D. Oracle サーバー内の複雑なデータ分析を処理することによってではなく、アプリケーションにデータを取得して、データの独立性を増加させる SQL で実行されるタスク

Answer: A,C