



# ITTEST

QUESTION & ANSWER

Guías de estudio precisos, Alta tasa de paso!



Ittest ofrece información actualizada de forma gratuita en un año!

<http://www.ittest.es/>

**Exam** : **A2090-545**

**Title** : Assessment: DB2 9.7 SQL  
Procedure Developer  
Assessment

**Version** : DEMO

1. A developer needs to create a user-defined function that will return a list of employees who work in a particular department. Which statement will successfully create a function that meets this objective?

- A. CREATE FUNCTION dept\_employees (deptno CHAR(3)) RETURNS TABLE LANGUAGE SQL READS SQL DATA RETURN SELECT empno, lastname AS l\_name, firstnme AS f\_name FROM employee WHERE employee.workdept = dept\_employees.deptno
- B. CREATE FUNCTION dept\_employees (deptno CHAR(3)) RETURNS TABLE DYNAMIC RESULT SETS 1 LANGUAGE SQL READS SQL DATA DECLARE emp\_info CURSOR WITH RETURN FOR SELECT empno, lastname AS l\_name, firstnme AS f\_name FROM employee WHERE employee.workdept = dept\_employees.deptno OPEN emp\_info; RETURN
- C. CREATE FUNCTION dept\_employees (deptno CHAR(3)) RETURNS TABLE (empno CHAR(6), l\_name VARCHAR(15), f\_name VARCHAR(12)) LANGUAGE SQL READS SQL DATA RETURN SELECT empno, lastname AS l\_name, firstnme AS f\_name FROM employee WHERE employee.workdept = dept\_employees.deptno
- D. CREATE FUNCTION dept\_employees (deptno CHAR(3)) RETURNS TABLE (empno CHAR(6), l\_name VARCHAR(15), f\_name VARCHAR(12)) DYNAMIC RESULT SETS 1 LANGUAGE SQL READS SQL DATA DECLARE emp\_info CURSOR WITH RETURN FOR

SELECT empno, lastname AS l\_name, firstnme AS f\_name FROM employee WHERE employee.workdept = dept\_employees.deptno OPEN emp\_info; RETURN

**Answer: C**

2. In the function shown below:

```
CREATE FUNCTION fcn1(deptno CHAR(3))
  RETURNS TABLE(empno CHAR(6),
                 firstnme VARCHAR(12))
  READS SQL DATA
  RETURN
  SELECT empno, firstnme
  FROM employee
  WHERE employee.workdept = fcn1.deptno;
```

- A. SELECT \* FROM TABLE(fcn1('B01'))
- B. SELECT TABLE(fcn1('B01')) FROM SYSIBM.SYSDUMMY1
- C. SELECT \* FROM fcn1('B01')
- D. SELECT fcn1('B01') FROM SYSIBM.SYSDUMMY1

**Answer: A**

3.Which statement correctly describes characteristics of external functions?

- A. External functions cannot return tables.
- B. All cursors opened within an external function should stay open until the database isquiesced.
- C. Scratchpads can be used to allocate dynamic memory required for multiple function invocations.
- D. Transactions can be terminated within external functions.

**Answer: C**

4.Click the Exhibit button.

EMPNO	FIRSTNME JOB	LASTNAME WORKDEPT
000010	CHRISTINE PRES	HAAS A00
000020	MICHAEL MANAGER	THOMPSON B01
000050	JOHN MANAGER	GEYER E01
000060	IRVING MANAGER	STERN D11
000070	EVA MANAGER	PULASKI D21
000090	EILEEN MANAGER	HENDERSON E11
000100	THEODORE MANAGER	SPENSER E21
000200	DAVID DESIGNER	BROWN D11
000210	WILLIAM DESIGNER	JONES D11
000220	JENNIFER DESIGNER	LUTZ D11
000230	JAMES CLERK	JEFFERSON D21
000240	SALVATORE CLERK	MARINO D21
000250	DANIEL CLERK	SMITH D21
000260	SYBIL CLERK	JOHNSON E21
000270	MARIA CLERK	PEREZ D21
000280	ETHEL OPERATOR	SCHNEIDER E11
000290	JOHN OPERATOR	PARKER E11
000320	RAMLAL FIELDREP	MEHTA E21
000330	WING FIELDREP	LEE E21
200170	KIYOSHI DESIGNER	YAMAMOTO D11
200220	REBA DESIGNER	JOHN D11
200240	ROBERT CLERK	MONTEVERDE D21
200330	HELENA FIELDREP	WONG E21
200340	ROY FIELDREP	ALONZO E21

UDF definition:

```
CREATE FUNCTION getnumbemployee(p_dept
VARCHAR(3))
RETURNS TABLE (enum INT, empno VARCHAR(6),
                lastname VARCHAR(15),
                firstnme VARCHAR(12))
RETURN SELECT ROW_NUMBER() OVER (),
e.empno, e.lastname, e.firstnme
FROM employee e WHERE e.workdeq
```

Referring to the exhibit, how many rows will be returned by the SQL query shown below? SELECT \*

FROM TABLE(getnumemployee(21?) AS d)SELECT \* FROM TABLE(getnumemployee(?21?) AS d

A.0

B.1

C.7

D.10

**Answer: C**

5.Which three optional clauses can be used when creating an external function? (Choose three.)

A. SCRATCHPAD

B. NOTEPAD

C. LANGUAGE

D. EXTERNAL NAME

E. DATABASEINFO

**Answer: A,C,D**

6.Which statement is permitted within a scalar user-defined function body?

A. COMMIT

B. INSERT

C. SIGNAL

D. LOOP

**Answer: C**

7.Click the Exhibit button.

```
CREATE FUNCTION emp_birthdays (month INTEGER)
  RETURNS TABLE (l_name VARCHAR(15),
                 f_name VARCHAR(12),
                 month VARCHAR(20),
                 day INTEGER);
LANGUAGE SQL
READS SQL DATA
NO EXTERNAL ACTION
RETURN
  SELECT lastname, firstme, MONTHNAME(birthdate),
         DAY(birthdate)
  FROM employee
  WHERE MONTH(employee3.birthdate) =
emp_birthdays.month
```

A user-defined function was created using the statement shown in the exhibit. Which additional option can be added to the CREATE FUNCTION statement to tell the optimizer that the function does not always return the same results for a given argument value?

- A. NO EXTERNAL ACTION
- B. NOT FENCED
- C. NOT DETERMINISTIC
- D. STATIC DISPATCH

**Answer: C**

8. Given the following SQL:

```
CREATE PROCEDURE test_proc()
BEGIN
  DECLARE c1 CURSOR FOR
  VALUES ('NAME1','NAME2','NAME3');
  OPEN c1;
END
```

Which statement is incorrect?

- A. The procedure declaration requires the DYNAMIC RESULT SETS 1 clause in order to return a result set.
- B. The cursor declaration requires the WITH RETURN TO CLIENT clause in order to return a result set.
- C. The cursor declaration requires the WITH RETURN TO CALLER clause in order to return a result set.
- D. The cursor declaration requires the WITH RETURN clause in order to return a result set.

**Answer: A**

9. Given the following SQL:

```
CREATE PROCEDURE test_proc()
BEGIN
  DECLARE i INTEGER DEFAULT 1;
  WHILE i < 10 DO
  BEGIN
    DECLARE cur CURSOR FOR
      VALUES ('NAME1','NAME2','NAME3');
    OPEN cur;
    SET index = index + 1;
  END;
  END WHILE;
END
```

Which of the following statements is true?

- A. The procedure declaration requires the DYNAMIC RESULT SETS 1 clause in order to return a result set.
- B. The cursor declaration requires WITH RETURN TO CLIENT clause in order to return a result set.
- C. The cursor declaration requires WITH RETURN TO CALLER clause in order to return a result set.
- D. The cursor declaration requires WITH RETURN clause in order to return a result set.

**Answer: B**

10. In the stored procedure below: What will the value of the P\_ID parameter be if the procedure is invoked and a value of 2 is specified for the START\_VALUE parameter?



```

CREATE PROCEDURE proc_labels (IN start_value INT, OUT p_ID INT)
s1: BEGIN
  DECLARE v_ID INT;
  s2: BEGIN
    DECLARE v_ID INT;
    SET v_ID = start_value;
    SET s1.v_ID = v_ID + s2.v_ID;
    SET v_ID = 3;
    SET p_ID = s2.v_ID;
  END;
END s1

```

- A. 1
- B. 2
- C. 3
- D. 4

**Answer: C**

11. Which procedure demonstrates the correct use of dynamic SQL?

A. CREATE PROCEDURE update\_count1 (IN new\_count INTEGER, IN item\_code INTEGER) BEGIN DECLARE v\_dynSQL VARCHAR(200); SET v\_dynSQL = 'UPDATE stock SET quantity\_on\_hand=? WHERE item\_number=?'; PREPARE v\_stmt1 FROM v\_dynSQL; EXECUTE v\_stmt1 USING new\_count, item\_code; END

B. CREATE PROCEDURE update\_count2 (IN tab\_name VARCHAR(128), IN new\_count INTEGER, IN item\_code INTEGER) BEGIN DECLARE v\_dynSQL VARCHAR(200); SET v\_dynSQL = 'UPDATE ? SET quantity\_on\_hand=? WHERE item\_number=?'; PREPARE v\_stmt1 FROM v\_dynSQL; EXECUTE v\_stmt1 USING tab\_name, new\_count, item\_code; END

C. CREATE PROCEDURE update\_count4 (IN tab\_name VARCHAR(128), IN col\_name1 VARCHAR(128), IN col\_name2 VARCHAR(128), IN new\_count INTEGER, IN item\_code INTEGER) BEGIN

DECLARE v\_dynSQL VARCHAR(200); SET v\_dynSQL = 'UPDATE ? SET ?? WHERE ??'; PREPARE v\_stmt1 FROM v\_dynSQL; EXECUTE v\_stmt1 USING tab\_name, col\_name1, new\_count, col\_name2, item\_code; END

D. CREATE PROCEDURE update\_count5 (IN new\_count INTEGER, IN item\_code INTEGER) BEGIN DECLARE v\_dynSQL VARCHAR(200); DECLARE v\_col\_name VARCHAR(128); SET v\_col\_name =

```
'item_number'; SET v_dynSQL = 'UPDATE stock SET quantity_on_hand=? WHERE ?=?'; PREPARE v_stmt1 FROM v_dynSQL; EXECUTE v_stmt1 USING new_count, v_col_name, item_code; END
```

**Answer: A**

12.A developer wants to code the following statements in an SQL procedure:

```
SET salary = salary * 1.05;          ---- 1
DECLARE empid INTEGER;              ---- 2
DECLARE EXIT HANDLER FOR NOT FOUND  ---- 3
    SET empid = 0;
DECLARE cursor1 CURSOR FOR SELECT * FROM employee; ---- 4
```

- A. 1, 2, 3, 4
- B. 2, 4, 3, 1
- C. 3, 4, 2, 1
- D. 4, 3, 2, 1

**Answer: B**

13.What is a reason to use SQL procedures in DB2?

- A. to use different programming languages B. to reduce code sharing
- C. to move business logic rules into the database
- D. to eliminate the need for testing

**Answer: C**

14.Which SQL procedure declaration is coded correctly?

- A. CREATE PROCEDUREmyproc(IN salary DOUBLE, OUT commission DOUBLE) BEGIN DECLARE EXIT HANDLER FOR SQLEXCEPTION SET commission = 0; DECLARE my\_cur CURSOR FOR SELECT \* FROM employee; DECLARE a DOUBLE; SET a = .06 \* salary; SET commission = a; END
- B. CREATE PROCEDUREmyproc(IN salary DOUBLE, OUT commission DOUBLE) BEGIN DECLARE EXIT HANDLER FOR SQLEXCEPTION SET commission = 0; DECLARE a DOUBLE; DECLARE my\_cur CURSOR FOR SELECT \* FROM employee; SET a = .06 \* salary; SET commission = a; END

C. CREATE PROCEDURE myproc(IN salary DOUBLE, OUT commission DOUBLE) BEGIN DECLARE a DOUBLE; DECLARE EXIT HANDLER FOR SQLEXCEPTION SET commission = 0; DECLARE my\_cur CURSOR FOR SELECT \* FROM employee; SET a = .06 \* salary; SET commission = a; END

D. CREATE PROCEDURE myproc(IN salary DOUBLE, OUT commission DOUBLE) BEGIN DECLARE a DOUBLE; DECLARE my\_cur CURSOR FOR SELECT \* FROM employee; DECLARE EXIT HANDLER FOR SQLEXCEPTION

SET commission = 0; SET a = .06 \* salary; SET commission = a; END

**Answer: D**

15. The CREATE PROCEDURE statement shown below was executed against a database called MYDB. CREATE PROCEDURE myschema.proc1(IN p1 INTEGER, OUT p2 CHAR(4), OUT p3 SMALLINT) BEGIN SET p2 = 'abc'; END Which two CREATE PROCEDURE statements, when executed against the same database, will succeed? (Choose two.)

A. CREATE PROCEDURE myschema.proc1(IN p1 CHAR(4), OUT p2 INTEGER) BEGIN SET p2 = 123; END

B. CREATE PROCEDURE myschema.proc1(IN p1 INTEGER, OUT p2 CHAR(4), OUT p3 CHAR(4)) BEGIN SET p2 = 'abc'; END

C. CREATE PROCEDURE myschema.proc1(IN p1 CHAR(4), OUT p2 INTEGER, OUT p3 SMALLINT) BEGIN SET p2 = 123; END

D. CREATE PROCEDURE otherschema.proc1(IN p1 CHAR(4), OUT p2 CHAR(4), OUT p3 CHAR(4)) BEGIN SET p2 = 'abc'; END

E. CREATE PROCEDURE myschema.proc1(IN p1 NUMBER, OUT p2 NUMBER, OUT p3 NUMBER) BEGIN SET p2 = 'abc'; END

**Answer: A,D**

16. If this statement was used to create an SQL procedure named P\_EXIT:

```
CREATE PROCEDURE p_exit (IN p_in INTEGER, OUT p_out INTEGER)
BEGIN
  IF (p_in < 0) THEN
    SIGNAL SQLSTATE '70001' SET MESSAGE_TEXT = 'Custom SQL State';
  ELSEIF (p_in < 10) THEN
    RETURN 10;
  ELSE
    SET p_out = 20;
  END IF;
  RETURN 99;
END
```

- A. 10
- B. 20
- C. 99
- D. NULL

**Answer: C**

17.The CREATE OR REPLACE PROCEDURE statement is similar semantically to which of the following combined statements?

- A. DROP and CREATE PROCEDURE
- B. ALTER and CREATE PROCEDURE
- C. UPDATE and CREATE PROCEDURE
- D. DROP and ALTER PROCEDURE

**Answer: A**

18.Given the two SQL procedures shown below: What is the expected output if procedure S1 is invoked with the value 1 provided for parameter V1?

```
CREATE PROCEDURE s2(IN p1 INT, OUT p2 INT)
BEGIN
  SET p2 = p1 + 3;
END
```

```
CREATE PROCEDURE s1(IN v1 INT, OUT v2 INT)
BEGIN
  DECLARE v3 INT;
  SET v3 = v1 + 1;
  CALL s2(v3,v2);
  SET v2 = v2 + 2;
END
```

- A. NULL
- B. 2
- C. 5

D. 7

**Answer: D**

19.Which CREATE PROCEDURE statement option should be used if you plan on issuing a DECLARE GLOBAL TEMPORARY TABLE statement from within the SQL procedure body?

A. CONTAINS SQL

B. READS SQL DATA

C. MODIFIES SQL DATA

D. LANGUAGE SQL

**Answer: C**

20.Click on the Exhibit button.

```
CREATE TYPE phonenumbers AS VARCHAR(12)
ARRAY[1000]

CREATE PROCEDURE find_customers( IN numbers_in
phonenumbers,
                                IN area_code
                                CHAR(3),
                                OUT numbers_out
phonenumbers)
BEGIN
  DECLARE i, j, max INTEGER;
  SET i = 1;
  SET j = 1;
  SET numbers_out = NULL;
  SET max = CARDINALITY(numbers_in);

  WHILE i <= max DO
    IF SUBSTR(numbers_in[i], 1, 3) = area_ccdc THEN
      SET numbers_out[j] = numbers_in[i];
      SET j = j + 1;
    END IF;
    SET i = i + 1;
  END WHILE;
END
```

Referring to the exhibit, what is the OUT\_PHONENUMBERS result set when the SQL procedure is invoked by the call statement shown below? CALL

find\_customers(ARRAY[?16-305-3745? ?05-416-4565? ?16-305-3746? '905-414-4566'], ?16??)

- A. [?16-305-3745??05-416-4565? ?16-305-3746']
- B. [?16-305-3745??05-416-4565? ?16-305-3746? '905-414-4566']
- C. [?05-416-4565? '905-414-4566']
- D. [?16-305-3745? ?16-305-3746[?16-305-3745? ?16-305-3746?

**Answer: D**