

Vendor: IBM

Exam Code: C2090-545

Exam Name: DB2 9.7 SQL Procedure Developer

Version: DEMO

QUESTION 1

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: ACD

QUESTION 2

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

- A. COMMIT
- B. INSERT
- C. SIGNAL
- D. LOOP

Answer: C

QUESTION 3

Click the Exhibit button.

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

QUESTION 4

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;

Which statement can be used to invoke the function above?

- A. SELECT * FROMTABLE(fcn1('B01'))
- B. SELECTTABLE(fcn1('B01')) FROM SYSIBM.SYSDUMMY1
- C. SELECT * FROMfcn1('B01')
- D. SELECTfcn1('B01') FROM SYSIBM.SYSDUMMY1

Answer: A

QUESTION 5

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

QUESTION 6

Click the Exhibit button.

DOUDTO CHRISTINE HAAS PRES A00 DOUDTO MICHAEL THOMPSON MANAGER B01 DOUDTO EVA PULASKI PULASKI DOUDTO EVA PULASKI PU	EMPNO	FIRSTNME	LASTNAME
PRES		JOB	WORKDEPT
PRES			
MICHAEL MANAGER B01	000010	CHRISTINE	HAAS
MANAGER B01 GEYER MANAGER E01		The second secon	10.00
DOUDSO	000020		
MANAGER E01	000000		
000060	000050		
MANAGER D11	nannen		0-3-5 (MMM)
DOUDTO	000000		
MANAGER D21	000070	State of the state	
MANAGER E11 000100 THEODORE SPENSER MANAGER E21 000200 DAVID BROWN DESIGNER D11 000210 WILLIAM JONES DESIGNER D11 000220 JENNIFER LUTZ DESIGNER D11 000230 JAMES JEFFERSON CLERK D21 000240 SALVATORE MARINO CLERK D21 000250 DANIEL SMITH CLERK D21 000260 SYBIL JOHNSON CLERK E21 000270 MARIA PEREZ CLERK D21 000270 MARIA PEREZ CLERK D21 000280 ETHEL SCHNEIDER OPERATOR E11 000290 JOHN PARKER OPERATOR E11 0003020 RAMLAL MEHTA FIELDREP E21 000330 WING LEE 200170 KIYOSHI VAMAMOTO DESIGNER D11 200220 REBA JOHN DESIGNER D11 200220 ROBERT MONTEVERDE CLERK D21 200330 HELENA WONG FIELDREP E21 200340 ROY ALONZO FIELDREP E21 200340 ROY ALONZO FIELDREP E21 UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) Iastname VARCHAR(15), firstnme VARCHAR(15), firstnme VARCHAR(15),	35031130	MANAGER	
000100 THEODORE MANAGER SPENSER E21 000200 DAVID BROWN DESIGNER D11 000210 WILLIAM JONES DESIGNER D11 000220 JENNIFER LUTZ DESIGNER D11 000230 JAMES JEFFERSON CLERK D21 000240 SALVATORE MARINO CLERK D21 000250 DANIEL SMITH CLERK D21 000260 SYBIL JOHNSON CLERK E21 000270 MARIA PEREZ CLERK D21 000280 ETHEL SCHNEIDER D21 000280 ETHEL SCHNEIDER D21 000290 JOHN PARKER D21 000301 RAMLAL MEHTA FIELDREP E21 000330 WING LEE 200170 KIYOSHI YAMAMOTO DESIGNER D11 200220 REBA JOHN DESIGNER D11 200220 REBA JOHN DESIGNER D11 200220 REBA JOHN DESIGNER D11 200330 HELENA WONG FIELDREP E21 200340 ROY ALONZO FIELDREP E21 2	000090	EILEEN	HENDERSON
MANAGER E21 000200 DAVID BROWN DESIGNER D11 000210 WILLIAM JONES DESIGNER D11 000220 JENNIFER LUTZ DESIGNER D11 000230 JAMES JEFFERSON CLERK D21 000240 SALVATORE MARINO CLERK D21 000250 DANIEL SMITH CLERK D21 000260 SYBIL JOHNSON CLERK E21 000270 MARIA PEREZ CLERK D21 000280 ETHEL SCHNEIDER OPERATOR E11 000290 JOHN PARKER OPERATOR E11 000320 RAMLAL MEHTA FIELDREP E21 000330 WING LEE 200170 KIYOSHI YAMAMOTO DESIGNER D11 200220 REBA JOHN DESIGNER D11 200220 REBA JOHN DESIGNER D11 200230 ROPERT MONTEVERDE CLERK D21 000330 FIELDREP E21 000330 FIELDREP E21 000330 FIELDREP E21 000340 ROY ALONZO FIELDREP E21 000340 ROY ALONZO FIELDREP E21 000540 ROPERT MONTEVERDE CLERK D21 00055 FIELDREP E21 00056 FIELDREP E21 00057 FIELDREP E21 00057 FIELDREP E21 00058 FIELDREP E21 00058 FIELDREP E21 00058 FIELDREP E21 00059 FIELDREP E21 00059 FIELDREP E21 00059 FIELDREP E21 0059 FIELDR	300000		
DAVID	000100		
DESIGNER D11 000210 WILLIAM JONES DESIGNER D11 000220 JENNIFER LUTZ DESIGNER D11 000230 JAMES JEFFERSON CLERK D21 000240 SALVATORE MARINO CLERK D21 000250 DANIEL SMITH CLERK D21 000260 SYBIL JOHNSON CLERK E21 000270 MARIA PEREZ CLERK D21 000270 MARIA PEREZ CLERK D21 000280 ETHEL SCHNEIDER OPERATOR E11 000290 JOHN PARKER OPERATOR E11 000320 RAMLAL MEHTA FIELDREP E21 000330 WING LEE 200170 KIYOSHI YAMAMOTO DESIGNER D11 200220 REBA JOHN DESIGNER D11 200220 REBA JOHN DESIGNER D11 200230 ROPERT MONTEVERDE CLERK D21 200330 HELENA WONG FIELDREP E21 200340 ROY ALONZO FIELDREP E21 200340 ROY ALONZO FIELDREP E21 200540 ROY ALONZO FIELDREP E21 20055 FIELDREP E21 20065 FIELDREP E21 20076 ROY ALONZO FIELDREP E21 20077 CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) Iastname VARCHAR(15), firstnme VARCHAR(15), firstnme VARCHAR(15)		Control of the contro	
ODD ODD ODD	000200		
DESIGNER D11 000220 JENNIFER LUTZ DESIGNER D11 000230 JAMES JEFFERSON CLERK D21 000240 SALVATORE MARINO CLERK D21 000250 DANIEL SMITH CLERK D21 000260 SYBIL JOHNSON CLERK D21 000260 SYBIL SCHNEIDER 000270 MARIA PEREZ CLERK D21 000280 ETHEL SCHNEIDER 0PERATOR E11 000290 JOHN PARKER 0PERATOR E11 000320 RAMLAL MEHTA FIELDREP E21 000330 WING LEE 200170 KIYOSHI YAMAMOTO DESIGNER D11 200220 REBA JOHN DESIGNER D11 200220 REBA JOHN DESIGNER D11 200230 HELENA WONG FIELDREP E21 200330 HELENA WONG FIELDREP E21 200340 ROY ALONZO FIELDREP E21 UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) Iastname VARCHAR(15), firstnme VARCHAR(15), firstnme VARCHAR(15)	000040	CONCOUNT OF THE PROPERTY OF	
000220 JENNIFER DESIGNER LUTZ DESIGNER D11 000230 JAMES JEFFERSON CLERK D21 000240 SALVATORE MARINO CLERK D21 000250 DANIEL SMITH CLERK D21 000260 SYBIL JOHNSON CLERK E21 000270 MARIA PEREZ CLERK D21 000280 ETHEL SCHNEIDER OPERATOR E11 000290 JOHN PARKER OPERATOR E11 000290 JOHN PARKER OPERATOR E11 000301 JOHN PARKER OPERATOR E11 000320 RAMLAL MEHTA FIELDREP E21 E21 000330 WING LEE LEE 200170 KIYOSHI YAMAMOTO DESIGNER D11 DESIGNER D11 200220 REBA JOHN DESIGNER D11 D21 200340 ROBERT MONTEVERDE CLERK D21 D21 200340 ROY ALONZO FIELDREP E21 200340 ROY ALONZO FIELDREP E21 200340 ROY ALONZO FIELDREP E21 200540 ROBERT GARCHAR(15), firstnime VARCHAR(15), firstnime VARCHAR(15), firstnime VARCHAR(15),	000210		
DESIGNER D11 000230 JAMES JEFFERSON CLERK D21 000240 SALVATORE MARINO CLERK D21 000250 DANIEL SMITH CLERK D21 000260 SYBIL JOHNSON CLERK E21 000270 MARIA PEREZ CLERK D21 000280 ETHEL SCHNEIDER OPERATOR E11 000290 JOHN PARKER OPERATOR E11 000320 RAMLAL MEHTA FIELDREP E21 000330 WING LEE 200170 KIYOSHI YAMAMOTO DESIGNER D11 200220 REBA JOHN DESIGNER D11 200220 REBA JOHN DESIGNER D11 200240 ROBERT MONTEVERDE CLERK D21 200330 HELENA WONG FIELDREP E21 200340 ROY ALONZO FIELDREP E21 UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) Iastname VARCHAR(15), firstnme VARCHAR(15), firstnme VARCHAR(15)	nnnoon		
DOD230	000220		
CLERK D21	000230		
CLERK D21 000250 DANIEL SMITH CLERK D21 000260 SYBIL JOHNSON CLERK E21 000270 MARIA PEREZ CLERK D21 000280 ETHEL SCHNEIDER OPERATOR E11 000290 JOHN PARKER OPERATOR E11 000320 RAMLAL MEHTA FIELDREP E21 000330 WING LEE 200170 KIYOSHI YAMAMOTO DESIGNER D11 200220 REBA JOHN DESIGNER D11 200220 REBA JOHN DESIGNER D11 200230 HELENA WONG FIELDREP E21 200330 HELENA WONG FIELDREP E21 000330 ROY ALONZO FIELDREP E21 000340 ROY ALONZO FIELDREP E21 00055 GARCHAR(15) FIRSTNAME VARCHAR(15), FIRSTNAME VARCHAR(15), FIRSTNAME VARCHAR(15), FIRSTNAME VARCHAR(15), FIRSTNAME VARCHAR(15),		The state of the s	
000250 DANIEL CLERK SMITH D21 000260 SYBIL CLERK JOHNSON CLERK E21 000270 MARIA MARIA PEREZ D21 000280 ETHEL OPERATOR E11 000290 JOHN PARKER OPERATOR 0PERATOR E11 000320 RAMLAL FIELDREP MEHTA FIELDREP 200170 KIYOSHI DESIGNER YAMAMOTO DESIGNER 200220 REBA D21 JOHN DESIGNER 200240 ROBERT CLERK D21 200330 HELENA FIELDREP E21 200340 ROY FIELDREP E21 UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) Iastname VARCHAR(15), firstnme VARCHAR(15),	000240		
CLERK D21 000260 SYBIL JOHNSON CLERK E21 000270 MARIA PEREZ CLERK D21 000280 ETHEL SCHNEIDER OPERATOR E11 000290 JOHN PARKER OPERATOR E11 000320 RAMLAL MEHTA FIELDREP E21 000330 WING LEE 200170 KIYOSHI YAMAMOTO DESIGNER D11 200220 REBA JOHN DESIGNER D11 200240 ROBERT MONTEVERDE CLERK D21 200330 HELENA WONG FIELDREP E21 200340 ROY ALONZO FIELDREP E21 UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) lastname VARCHAR(15), firstnme VARCHAR(15),		CLERK	D21
000260 SYBIL CLERK JOHNSON E21 000270 MARIA PEREZ CLERK 000280 ETHEL ETHEL OPERATOR SCHNEIDER E11 000290 JOHN OPERATOR E11 000320 RAMLAL FIELDREP E21 000330 WING WING LEE 200170 KIYOSHI DESIGNER YAMAMOTO DESIGNER 200220 REBA DESIGNER JOHN DESIGNER 200240 ROBERT CLERK D21 200330 HELENA FIELDREP WONG FIELDREP 200340 ROY FIELDREP E21 UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) Iastname VARCHAR(15), firstnme VARCHAR(15))	000250	DANIEL	SMITH
CLERK E21 000270 MARIA PEREZ CLERK D21 000280 ETHEL SCHNEIDER OPERATOR E11 000290 JOHN PARKER OPERATOR E11 000320 RAMLAL MEHTA FIELDREP E21 000330 WING LEE 200170 KIYOSHI YAMAMOTO DESIGNER D11 200220 REBA JOHN DESIGNER D11 200240 ROBERT MONTEVERDE CLERK D21 200330 HELENA WONG FIELDREP E21 200340 ROY ALONZO FIELDREP E21 UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) Iastname VARCHAR(15), firstnme VARCHAR(15)			
000270	000260		
CLERK D21 000280 ETHEL SCHNEIDER	000070		
ODD ODD ODD ODD	000270		
OPERATOR E11 000290 JOHN PARKER OPERATOR E11 000320 RAMLAL MEHTA FIELDREP E21 000330 WING LEE 200170 KIYOSHI YAMAMOTO DESIGNER D11 200220 REBA JOHN DESIGNER D11 200240 ROBERT MONTEVERDE CLERK D21 200330 HELENA WONG FIELDREP E21 200340 ROY ALONZO FIELDREP E21 UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) lastname VARCHAR(15), firstnme VARCHAR(15)	nnnnen		
000290 JOHN PARKER	000200		
OPERATOR E11 000320 RAMLAL MEHTA FIELDREP E21 000330 WING LEE 200170 KIYOSHI YAMAMOTO DESIGNER D11 200220 REBA JOHN DESIGNER D11 200240 ROBERT MONTEVERDE CLERK D21 200330 HELENA WONG FIELDREP E21 200340 ROY ALONZO FIELDREP E21 UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) lastname VARCHAR(15), firstnme VARCHAR(15)	000290		The state of the s
000320 RAMLAL MEHTA FIELDREP E21 000330 WING LEE 200170 KIYOSHI YAMAMOTO DESIGNER D11 200220 REBA JOHN DESIGNER D11 200240 ROBERT MONTEVERDE CLERK D21 200330 HELENA WONG FIELDREP E21 200340 ROY ALONZO FIELDREP E21 UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) lastname VARCHAR(15), firstnme VARCHAR(15)			
000330 WING LEE 200170 KIYOSHI YAMAMOTO DESIGNER D11 200220 REBA JOHN DESIGNER D11 200240 ROBERT MONTEVERDE CLERK D21 200330 HELENA WONG FIELDREP E21 200340 ROY ALONZO FIELDREP E21 UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) lastname VARCHAR(15), firstnme VARCHAR(15))	000320		
200170 KIYOSHI YAMAMOTO		FIELDREP	E21
DESIGNER D11 200220 REBA JOHN DESIGNER D11 200240 ROBERT MONTEVERDE CLERK D21 200330 HELENA WONG FIELDREP E21 200340 ROY ALONZO FIELDREP E21 UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) lastname VARCHAR(15), firstnme VARCHAR(12))	000330	WING	LEE
200220 REBA DOHN DESIGNER D11 200240 ROBERT MONTEVERDE CLERK D21 200330 HELENA WONG FIELDREP E21 200340 ROY ALONZO FIELDREP E21 UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) lastname VARCHAR(15), firstnme VARCHAR(12))	200170	KIYOSHI	YAMAMOTO
DESIGNER D11 200240 ROBERT MONTEVERDE CLERK D21 200330 HELENA WONG FIELDREP E21 200340 ROY ALONZO FIELDREP E21 UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) lastname VARCHAR(15), firstnme VARCHAR(12))		DESIGNER	
200240 ROBERT CLERK MONTEVERDE 200330 HELENA WONG FIELDREP E21 200340 ROY ALONZO FIELDREP E21 UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) lastname VARCHAR(15), firstnme VARCHAR(12))	200220		
CLERK D21 200330 HELENA WONG FIELDREP E21 200340 ROY ALONZO FIELDREP E21 UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) liastname VARCHAR(15), firstnme VARCHAR(12))			1 May 1 7 7
200330 HELENA WONG	200240		
FIELDREP E21 200340 ROY ALONZO FIELDREP E21 UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) lastname VARCHAR(15), firstnme VARCHAR(12))	200220		
200340 ROY ALONZO FIELDREP E21 UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) lastname VARCHAR(15), firstnme VARCHAR(12))	200330		
FIELDREP E21 UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) lastname VARCHAR(15), firstnme VARCHAR(12))	200340		
UDF definition: CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) lastname VARCHAR(15), firstnme VARCHAR(12))	200040		
CREATE FUNCTION getnumbemployee(p_dept VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) lastname VARCHAR(15), firstnme VARCHAR(12))		TO THE PARTY OF TH	
VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) Iastname VARCHAR(15), firstnme VARCHAR(12))	UDF defi	nition:	
VARCHAR(3)) RETURNS TABLE (enum INT, empno VARCHAR(6) Iastname VARCHAR(15), firstnme VARCHAR(12))	CREATE	FUNCTION getru	mhemplovee/n dent
RETURNS TABLE (enum INT, empno VARCHAR(6) lastname VARCHAR(15), firstnme VARCHAR(12))			mocmpojec(p_dept
lastname VARCHAR(15), firstnme VARCHAR(12))			INT, empno VARCHAR(6).
firstnme VARCHAR(12))			
		N SELECT ROW	NUMBER() OVER (),
e empno, e lastname, e firstnine FROM employee e WHERE e workd			

Referring to the exhibit, how many rows will be returned by the SQL query shown below? SELECT * FROM TABLE(getnumemployee(21?) AS dSELECT * FROM TABLE(getnumemployee(?21?) ASd

- A. 0
- B. 1
- C. 7

D. 10

Answer: C

QUESTION 7

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

QUESTION 8

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

QUESTION 9

In the stored procedure below:

```
CREATE PROCEDURE proc_labels (IN start_value INT, OUT p_ID INT)
sl1: BEGIN

DECLARE v_ID INT;
sl2: BEGIN

DECLARE v_ID INT;
SET v_ID = start_value;
SET sl1.v_ID = v_ID + sl2.v_ID;
SET v_ID = 3;
SET p_ID = sl2.v_ID;
END;
END;
```

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?

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

Answer: C

QUESTION 10

Which procedure demonstrates the correct use of dynamic SQL?

```
A. CREATE PROCEDURE update count1 (INnew 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;
B. CREATE PROCEDURE update count2 (INtab 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 (INtab name VARCHAR(128), IN col name1
   VARCHAR(128), IN col name2 VARCHAR(128), IN
   new_countINTEGER, IN item_code INTEGER)
   BEGIN
   DECLARE v_dynSQL VARCHAR(200);
   SET v_dynSQL = 'UPDATE ? SET ?=? WHERE ?=?';
   PREPARE v stmt1 FROM v dvnSQL:
   EXECUTE v_stmt1 USING tab_name, col_name1, new_count, col_name2, item_code; END
```

D. CREATE PROCEDURE update count5 (INnew 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

QUESTION 11

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 SET empid = 0;	3
DECLARE cursor1 CURSOR FOR SELECT * FROM employee;	4

What order must these statements be coded in?

A. 1, 2, 3, 4

B. 2, 4, 3, 1

C. 3, 4, 2, 1

D. 4, 3, 2, 1

Answer: B

QUESTION 12

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 FUNCTIONdept_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 FUNCTIONdept_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 I_name, firstnme AS f_name FROM employee

WHERE employee.workdept = dept_employees.deptno

OPEN emp_info;

RETURN

C. CREATE FUNCTIONdept employees (deptno CHAR(3))

RETURNS TABLE (empno CHAR(6),

I_nameVARCHAR(15),

f_nameVARCHAR(12))

LANGUAGE SQL

READS SQL DATA

RETURN

SELECT empno, lastname AS I_name, firstnme AS f_name FROM employee

WHERE employee.workdept = dept_employees.deptno

D. CREATE FUNCTIONdept_employees (deptno CHAR(3))

RETURNS TABLE (empno CHAR(6),

I_nameVARCHAR(15),

f_nameVARCHAR(12))

DYNAMIC RESULT SETS 1

LANGUAGE SQL

READS SQL DATA

DECLARE emp_info CURSOR WITH RETURN FOR

SELECT empno, lastname AS I_name, firstnme AS f_name FROM employee

WHERE employee.workdept = dept_employees.deptno

OPEN emp_info;

RETURN

Answer: C

Thank You for Trying Our Product

Braindump2go Certification Exam Features:

- ★ More than 99,900 Satisfied Customers Worldwide.
- ★ Average 99.9% Success Rate.
- ★ Free Update to match latest and real exam scenarios.
- ★ Instant Download Access! No Setup required.
- ★ Questions & Answers are downloadable in PDF format and VCE test engine format.



- ★ Multi-Platform capabilities Windows, Laptop, Mac, Android, iPhone, iPod, iPad.
- ★ 100% Guaranteed Success or 100% Money Back Guarantee.
- ★ Fast, helpful support 24x7.

View list of all certification exams: http://www.braindump2go.com/all-products.html

























10% Discount Coupon Code: BDNT2014