

**100%** Money Back  
**Guarantee**

**Vendor:** IBM

**Exam Code:** C2090-735

**Exam Name:** DB2 9.5 SQL Procedure Developer

**Version:** Demo

---

**QUESTION NO: 1**

Given the statement shown below:

```
SELECT ROW CHANGE TOKEN FOR dept, RID_BIT (dept)
FROM dept WHERE deptno = 'A00' WITH UR
```

Which two statements are true? (Choose two.)

- A. The statement is selecting two columns from DEPT table.
- B. The statement will allow the latest ROW CHANGE TOKEN value to be returned.
- C. The statement will allow the earliest ROW CHANGE TOKEN value to be returned.
- D. The statement will return a TIMESTAMP value.
- E. The statement uses optimistic locking.

**Answer: B,E**

**Explanation:**

**QUESTION NO: 2**

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**

**Explanation:**

**QUESTION NO: 3**

Click the Exhibit button.

```
CREATE PROCEDURE testproc( IN i1 INT, INOUT i3 INT)
```

---

SPECIFIC testproc

BEGIN

SET i3 = i1;

END

CREATE PROCEDURE testproc( IN i1 INT, INOUT i2 INT, INOUT i3 INT)

SPECIFIC testp

BEGIN

SET i3 = i1 \* i2;

END

Given that the statements in the exhibits have executed successfully, which solution contains the complete set of commands that could be used to drop both procedures in the order presented?

- A. DROP PROCEDURE testp;  
DROP PROCEDURE testp;
- B. DROP PROCEDURE testp;  
DROP PROCEDURE testproc;
- C. DROP SPECIFIC PROCEDURE testproc;  
DROP PROCEDURE testproc;
- D. DROP PROCEDURE testproc(INT);

**Answer: C**

**Explanation:**

#### QUESTION NO: 4

Click the Exhibit button.

CREATE FUNCTION sum(a INT, b INT)

RETURNS INTEGER

SPECIFIC sum\_of\_2

RETURN a + b;

---

```
CREATE FUNCTION sum(a INT, b INT, c INT)
```

```
RETURNS INTEGER
```

```
SPECIFIC sum_of_3
```

```
RETURN a + b + c;
```

Given the two functions in the exhibit, what is the correct command to invoke the function which calculates the sum of two numbers from an SQL procedure?

- A. SELECT sum\_of\_2 FROM table1;
- B. SELECT sum(2,4,?);
- C. SET res\_sum = sum(2,6);
- D. CALL sum(?,?,?);

**Answer: C**

**Explanation:**

#### QUESTION NO: 5

Given the statements shown below:

```
DECLARE c_dept CURSOR WITH HOLD FOR
```

```
SELECT * FROM dept;
```

```
OPEN c_dept;
```

Which two conditions are true? (Choose two.)

- A. C\_DEPT will remain open after a ROLLBACK.
- B. C\_DEPT will remain open after a COMMIT.
- C. C\_DEPT will be returned to the caller of the routine.
- D. C\_DEPT will be positioned before the next logical row.
- E. All locks held by C\_DEPT will be released after a COMMIT.

**Answer: B,D**

**Explanation:**

#### QUESTION NO: 6

---

Given the SQL statement shown below:

```
DECLARE test CURSOR FOR  
  
SELECT hiredate  
  
FROM employee  
  
FOR UPDATE;
```

Which statement correctly describes the cursor that is created?

- A. The cursor will be considered a read-only cursor.
- B. The cursor can only be used to perform positioned updates.
- C. The cursor can only be used to perform positioned deletes.
- D. The cursor can be used to perform positioned updates and deletes.

**Answer: D**

**Explanation:**

#### **QUESTION NO: 7**

Which statement can be used to define an array of 30 names that have a maximum size of 25 characters each?

- A. CREATE TYPE names AS VARCHAR(25) ARRAY[30];
- B. CREATE ARRAY names[30] VARCHAR(25);
- C. CREATE TYPE names[30] VARCHAR(25);
- D. CREATE ARRAY names AS VARCHAR(25);

**Answer: A**

**Explanation:**

#### **QUESTION NO: 8**

What will be the initial value of V\_MAX in the declaration statement shown below?

```
DECLARE v_max DECIMAL(9,2);
```

- A. 0.0

- 
- B. 2
  - C. 9
  - D. NULL

**Answer: D**

**Explanation:**

#### **QUESTION NO: 9**

Which statement should be used to declare an array with at most 10 elements of type INTEGER?

- A. DECLARE sub\_total INTEGER[10];
- B. DECLARE sub\_total[10] INTEGER;
- C. CREATE TYPE sub\_total AS INTEGER[10];
- D. CREATE TYPE sub\_total[10] AS INTEGER;

**Answer: C**

**Explanation:**

#### **QUESTION NO: 10**

What are two valid DECLARE statements in an SQL procedure? (Choose two.)

- A. DECLARE var1 INTEGER;
- B. DECLARE var1 DECIMAL [9];
- C. DECLARE var1 XML;
- D. DECLARE var1 CURRENT DATE;
- E. DECLARE var1[10] INTEGER;

**Answer: A,C**

**Explanation:**

#### **QUESTION NO: 11**

Which steps must be followed to return a result set from an SQL procedure?

- A. 1. Create the procedure using the DYNAMIC RESULT SETS clause.  
2. Declare the cursor.

- 
3. Open the cursor in the SQL procedure.
  4. Close the cursor.
  5. Return to the application.
- B.** 1. Create the procedure using the DYNAMIC RESULT SETS clause.  
2. Declare the cursor using the WITH RETURN clause.  
3. Open the cursor in the SQL procedure.  
4. Return to the application.
- C.** 1. Create the procedure using the WITH RETURN clause.  
2. Declare the cursor using the DYNAMIC RESULT SETS clause.  
3. Open the cursor in the SQL procedure.  
4. Return to the application.
- D.** 1. Create the procedure using the WITH RETURN clause.  
2. Declare the cursor using the DYNAMIC RESULT SETS clause.  
3. Open the cursor in the SQL procedure.  
4. Close the cursor.

**Answer: B**

**Explanation:**

#### **QUESTION NO: 12**

Which statement can be used to declare a variable inside an SQL procedure that can be used to represent a monetary value?

- A. DECLARE v\_money MONEY;
- B. DECLARE v\_money DOUBLE;
- C. DECLARE v\_money DECIMAL(9,2);
- D. DECLARE v\_money CURRENCY;

**Answer: C**

**Explanation:**

#### **QUESTION NO: 13**

What are two valid special registers? (Choose two.)

- A. CURRENT\_CLIENT\_ACCT
- B. CURRENT\_SCHEMA
- C. CURRENT\_PATH
- D. CURRENT\_DATETIME
- E. CURRENT\_PARTITION

---

**Answer: B,C**

**Explanation:**

**QUESTION NO: 14**

Which statement will successfully create an SQL procedure that returns the name of the current month?

**A.** CREATE PROCEDURE proc.current\_month(OUT month VARCHAR(20))  
BEGIN

```
DECLARE today DATE;  
SET (today = CURRENT_DATE);  
SET month = MONTHNAME(today);  
END
```

**B.** CREATE PROCEDURE proc.current\_month(OUT month VARCHAR(20))  
BEGIN

```
DECLARE today DATE;  
SELECT (CURRENT_DATE) INTO today;  
SET month = MONTHNAME(today);  
END
```

**C.** CREATE PROCEDURE proc.current\_month(OUT month VARCHAR(20))  
BEGIN

```
DECLARE today DATE;  
VALUES (CURRENT_DATE) INTO today;  
SET month = MONTHNAME(today);  
END
```

**D.** CREATE PROCEDURE proc.current\_month(OUT month VARCHAR(20))  
BEGIN

```
SET month = MONTHNAME(SELECT (CURRENT_DATE))  
END
```

**Answer: C**

**Explanation:**

**QUESTION NO: 15**

Which statement will assign the schema names "SYSIBM", "SYSFUN", "SYSPROC", and "SYSIBMADM" to the CURRENT\_PATH special register?

**A.** SET PATH = SYSTEM PATH

**B.** SET CURRENT\_PATH = DEFAULT

- 
- C. SET PATH = SYSTEM DEFAULT
  - D. RESET CURRENT PATH

**Answer: A**

**Explanation:**

**QUESTION NO: 16**

Given the variable declaration shown below:

```
DECLARE v_mydate DATE;
```

Which statement will assign a value to the variable V\_MYDATE?

- A. VALUES CURRENT TIME INTO v\_mydate;
- B. VALUES CURRENT TIMESTAMP INTO v\_mydate;
- C. SELECT CURRENT TIMESTAMP INTO v\_mydate FROM SYSIBM.SYSDUMMY1;
- D. SELECT CURRENT DATE INTO v\_mydate FROM SYSIBM.SYSDUMMY1;

**Answer: D**

**Explanation:**

**QUESTION NO: 17**

Given the variable declaration shown below:

```
DECLARE v_mytime TIME;
```

Which statement will assign a value to the variable named V\_MYTIME?

- A. SET v\_mytime = TIME;
- B. VALUES CURRENT TIME INTO v\_mytime;
- C. VALUES CURRENT TIMESTAMP INTO v\_mytime;
- D. SET v\_mytime = DATE;

**Answer: B**

**Explanation:**

---

**QUESTION NO: 18**

Which statement will change the value of a special register?

- A. UPDATE SPECIAL REGISTER TIME = 12:30:00
- B. UPDATE SPECIAL REGISTER SCHEMA = 'DB2ADMIN'
- C. SET CURRENT TIME = 12:30:00
- D. SET CURRENT SCHEMA = 'DB2ADMIN'

**Answer: D**

**Explanation:**

**QUESTION NO: 19**

What demonstrates the correct syntax for assigning three rows to the EMPNO, FIRSTNAME, and LASTNAME columns of a table named EMPLOYEE?

- A. INSERT INTO employee (empno, firstname, lastname)  
VALUES (100, 200, 300, 'John', 'Jane', 'Paul', 'Doe', 'Smith', 'Jones')
- B. INSERT INTO employee (empno, firstname, lastname)  
VALUES (100, 'John', 'Doe'), (200, 'Jane', 'Smith'), (300, 'Paul', 'Jones')
- C. SET (empno, firstname, lastname)  
VALUES (100, 200, 300, 'John', 'Jane', 'Paul', 'Doe', 'Smith', 'Jones')  
FOR employee
- D. SET (empno, firstname, lastname)  
VALUES (100, 'John', 'Doe'), (200, 'Jane', 'Smith'), (300, 'Paul', 'Jones')  
FOR employee

**Answer: B**

**Explanation:**

**QUESTION NO: 20**

Click the Exhibit button.

Given the following SQL procedure:

```
CREATE PROCEDURE test1 (IN someid CHAR(3), OUT status INT)
```

---

```

BEGIN
DECLARE SQLCODE INTEGER DEFAULT 0;
DECLARE count INTEGER DEFAULT 0;
SET status = 0;
IF (LENGTH(someid) < 3) THEN
RETURN -3;
GOTO exit;
ELSE
SELECT COUNT(deptno) INTO count FROM dept WHERE admrdept=someid;
IF(SQLCODE<>0) THEN
RETURN -1;
ELSE
SET status = COUNT;
RETURN 0;
END IF;
END IF;
exit:
RETURN -2;
END

```

Where the DEPT table consists of:

```

DEPTNO DEPTNAME
MGRNO ADMRDEPT

```

-----

```

A00 SPIFFY COMPUTER SERVICE DIV. 000010 A00
B01 PLANNING 000020 A00
C01 INFORMATION CENTER 000030 A00
D01 DEVELOPMENT CENTER - A00

```

---

D11 MANUFACTURING SYSTEMS 000060 D01

D21 ADMINISTRATION SYSTEMS 000070 D01

If the procedure TEST1 shown in the exhibit is called with the value 'A00' specified for the SOMEID parameter, what is the expected return code?

- A. 0
- B. -1
- C. -2
- D. -3

**Answer: A**

**Explanation:**

**QUESTION NO: 21**

Click the Exhibit button.

```
CREATE TABLE ar.sales
(customer_id INTEGER,
sales_amt DECIMAL(5,2),
rct TIMESTAMP NOT NULL
IMPLICITLY HIDDEN
GENERATED ALWAYS
FOR EACH ROW ON UPDATE AS
ROW CHANGE TIMESTAMP);
INSERT INTO ar.sales (customer_id, sales_amt) VALUES(1, 20.5);
SELECT * FROM ar.sales
WHERE ROW CHANGE TIMESTAMP FOR ar.sales <= CURRENT TIMESTAMP;
```

A DB2 Command Line Processor (CLP) file contains the set of statements shown in the exhibit. What does this sequence of statements illustrate?

- 
- A. work identity
  - B. row sequencing
  - C. row change tokenizing
  - D. optimistic locking

**Answer: D**

**Explanation:**

## **QUESTION NO: 22**

Click the Exhibit button.

```
CREATE PROCEDURE proc.test(IN month VARCHAR(12), OUT I_name
VARCHAR(25))
LANGUAGE SQL
BEGIN
DECLARE b_month CHAR(12);
DECLARE at_end INTEGER DEFAULT 0;
DECLARE not_found CONDITION FOR SQLSTATE '02000';
DECLARE c1 CURSOR FOR SELECT lastname, MONTHNAME(birthdate)
FROM employee ORDER BY birthdate, lastname;
DECLARE CONTINUE HANDLER FOR not_found SET at_end = 1;
OPEN c1;
this_loop: LOOP
FETCH c1 INTO I_name, b_month;
IF at_end = 1 THEN
LEAVE this_loop;
ELSEIF UCASE(b_month) != UCASE(month) THEN
ITERATE this_loop;
```

---

```
END IF;
END LOOP;
CLOSE c1;
END
```

A developer attempted to create a procedure to determine the oldest employee celebrating a birthday in a particular month by executing the SQL statement shown in the exhibit.

Tests show the procedure does not work as planned. What are two ways to make the procedure work as intended? (Choose two.)

- A. Add the statement ELSE RETURN; before the statement END IF;.
- B. Change the statement ELSEIFUCASE(b\_month) != UCASE(month) THEN to ELSEIF UCASE(b\_month) =UCASE(month) THEN.
- C. Add the statement ELSE BREAK; before the statement END IF;.
- D. Change the statement ELSEIFUCASE(b\_month) != UCASE(month) THEN ITERATE this\_loop; to ELSEIFUCASE(b\_month) = UCASE(month) THEN LEAVE this\_loop;.
- E. Add the statement ELSE CONTINUE; before the statement END IF;.

**Answer: A,D**

**Explanation:**

### QUESTION NO: 23

Click the Exhibit button.

```
CASE rating WHEN 1 THEN UPDATE employee SET salary = salary * 1.10 WHERE
empno = v_employee_number;
WHEN 2 THEN UPDATE employee SET salary = salary * 1.05 WHERE empno =
v_employee_number;
ELSE UPDATE employee SET salary = salary * 1.03 WHERE empno =
v_employee_number;
END CASE;
```

---

Which statement is true about the CASE statement shown in the exhibit?

- A. An employee with a rating of 1 receives a 10% salary increase.
- B. An employee with a rating of 3 receives no salary increase.
- C. An employee with a rating of 2 receives a 3% salary increase.
- D. All employees will receive at least a 5% salary increase.

**Answer: A**

**Explanation:**

#### QUESTION NO: 24

Click the Exhibit button.

```
BEGIN ATOMIC DECLARE fullname CHAR(40);  
FOR vl AS SELECT firstnme, midinit, lastname FROM employee DO SET fullname =  
lastname CONCAT ',' CONCAT firstnme CONCAT ' ' CONCAT midinit;  
INSERT INTO tnames VALUES (fullname);  
END FOR END
```

Which statement correctly describes the result of the FOR loop shown in the exhibit?

- A. FULLNAME is set to the last name of the employee, followed by a comma, the firstname, a blank space, and the middle initial. Only the last value for FULLNAME is inserted into table TNAMES.
- B. FULLNAME is set to the last name of the employee, followed by a comma, the firstname, a blank space, and the middle initial. Only the first value for FULLNAME is inserted into table TNAMES.
- C. FULLNAME is set to the last name of the employee, followed by a comma, the firstname, a blank space, and the middle initial for each row. Each value for FULLNAME is inserted into table TNAMES in alphabetical order.
- D. FULLNAME is set to the last name of the employee, followed by a comma, the firstname, a blank space, and the middle initial for each row. Each value for FULLNAME is inserted into table TNAMES.

**Answer: D**

**Explanation:**

---

**QUESTION NO: 25**

How is the FOR statement distinct from other conditional statements?

- A. FOR statements are evaluated at the completion of each iteration of the loop.
- B. FOR statements are evaluated before each iteration of the loop.
- C. FOR statements have a terminating condition clause.
- D. FOR statements are used to iterate over rows in a defined read-only result set.

**Answer: D**

**Explanation:**

**QUESTION NO: 26**

Which two statements describe a CASE statement? (Choose two.)

- A. CASE statements are used to enter into some logic based on a literal value.
- B. CASE statements are used to enter into some logic based on the value of an expression.
- C. CASE statements are used to return control to the beginning of an expression.
- D. CASE statements are used to enter into some condition and loop until the condition is met.
- E. CASE statements are used to iterate into some logic based on a literal value.

**Answer: A,B**

**Explanation:**

**QUESTION NO: 27**

Click the Exhibit button.

Table INFO1 contains the following data:

PERSON\_NAME AGE HOBBY DOB

-----

John Smith 45 Reading 05/01/1963

---

John Brown 5 Skiing 07/01/1982

Sally John 50 Reading 04/21/1958

Rob Drake 15 VideoGames 02/01/1993

Betty Hoop 20 Shopping 04/11/1988

Bill Lee 29 Sailing 12/01/1978

Mary Pope 48 Shopping 04/21/1960

Procedure MOVE\_DATA() is created as follows:

```
CREATE PROCEDURE move_data() BEGIN
```

```
FOR v1 AS SELECT person_name, age, hobby FROM info1 DO IF v1.age < 30 THEN
```

```
INSERT INTO info2 VALUES(v1.person_name, v1.hobby );
```

```
END IF;
```

```
END FOR;
```

```
END
```

How many rows will be in table INFO2 after the procedure MOVE\_DATA shown in the exhibit is executed?

- A. 0
- B. 4
- C. 5
- D. 7

**Answer: B**

**Explanation:**

**QUESTION NO: 28**

Click the Exhibit button.

A table named DEPT is created using the following DDL:

---

```
CREATE TABLE dept (id INT NOT NULL, name CHAR(9), salary DEC(7,2), comm  
DEC(7,2), PRIMARY KEY (id))
```

Immediately after the table is created, the following CREATE PROCEDURE statement is issued:

```
CREATE PROCEDURE test5 ()  
BEGIN NOT ATOMIC DECLARE v1 VARCHAR(10)  
INSERT INTO dept ( id, name) VALUES ( 10, 'Sales')  
INSERT INTO dept ( id, name) VALUES ( 20, 'Service')  
INSERT INTO dept ( id, name) VALUES ( 30, 'Service')  
SIGNAL SQLSTATE '70000'  
INSERT INTO dept ( id, name) VALUES ( 40, 'Service')  
END
```

The procedure TEST5 shown in the exhibit was invoked. How many rows will be added to the DEPT table?

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

**Answer: C**

**Explanation:**

**QUESTION NO: 29**

Click the Exhibit button.

```
CREATE PROCEDURE updatetest ( IN p_empname VARCHAR(30),  
IN p_empno CHAR(6),
```

---

```

OUT p_sqlstate CHAR(5),
OUT p_sqlcode INT )

BEGIN

DECLARE SQLSTATE CHAR(5) DEFAULT '00000';

DECLARE SQLCODE INT DEFAULT 0;

DECLARE EXIT HANDLER FOR SQLEXCEPTION

SELECT SQLSTATE, SQLCODE INTO p_sqlstate, p_sqlcode FROM

sysibm.sysdummy1;

VALUES (SQLSTATE, SQLCODE) INTO p_sqlstate, p_sqlcode;

UPDATE employee SET empname = p_empname WHERE empno = p_empno;

END

```

If the procedure shown in the exhibit is invoked, and the UPDATE statement returns an SQL0100W "No row was found for FETCH, UPDATE or DELETE; or the result of a query is an empty table. SQLSTATE '02000'", which two situations will be true? (Choose two.)

- A. P\_SQLSTATE will be set to '00000' and P\_SQLCODE will be set to 0.
- B. P\_SQLSTATE will be set to '02000' and P\_SQLCODE will be set to 100.
- C. P\_SQLSTATE will be set to '00000' and P\_SQLCODE will be set to 100.
- D. The caller will receive an SQLCODE of 0.
- E. The caller will receive an SQLCODE of 100

**Answer: A,E**

**Explanation:**

### QUESTION NO: 30

You want to see the explain plan for all SQL statements used in your SQL procedures. Which two statements will allow you to obtain the explain plan? (Choose two.)

- A. Copy SQL statements from the SQL procedure and paste into the Control Center, Command Editor, or Data Studio explain facility.
- B. Use the DB2SET command to globally set DB2\_SQLROUTINE\_PREPOPTS="EXPLAIN YES EXPLSNAP YES" .
- C. Call SYSPROC. SET\_ROUTINE\_OPTS('EXPLAIN YES EXPLSNAP YES') to setexplain

---

parameters for a session and compile the SQL procedure. Run the db2expl or db2exfmt tool to capture the explain plan.

**D.** Call SYSPROC. REBIND\_ROUTINE\_PACKAGE and specify 'EXPLAIN YES EXPLSNAP YES' bind option. Run the db2expl or db2exfmt tool to capture explain plan.

**E.** Call SYSPROC. SET\_ROUTINE\_OPTS('EXPLAIN YES EXPLSNAP YES') to set explain parameters for a session and compile the SQL procedure.

**Answer: A,C**

**Explanation:**

### **QUESTION NO: 31**

What is a reason to use SQL procedures in DB2?

**A.** to use different programming languages

**B.** to reduce code sharing

**C.** to reduce network traffic

**D.** to eliminate the need for testing

**Answer: C**

**Explanation:**

### **QUESTION NO: 32**

Which three statements are true about SQL procedures? (Choose three.)

**A.** SQL procedures reside outside of the database for easy portability.

**B.** SQL procedures can be called from triggers.

**C.** SQL procedures may return multiple result sets to the caller or to a client application.

**D.** SQL procedures support recursion.

**E.** SQL procedures support the PRINT statement.

**Answer: B,C,D**

**Explanation:**

### **QUESTION NO: 33**

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_code 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[416-305-3745,
905-414-4565,
416-305-3746,
'905-414-4566'],
416,?)
```

- A. [416-305-3745,905-414-4565, 416-305-3746']
- B. [416-305-3745,905-414-4565, 416-305-3746, '905-414-4566']

---

C. [905-414-4565, '905-414-4566']

D. [416-305-3745, 416-305-3746]

**Answer: D**

**Explanation:**

**QUESTION NO: 34**

Click the Exhibit button.

```
CREATE PROCEDURE procmain ()
```

```
DYNAMIC RESULT SETS 2
```

```
BEGIN
```

```
CALL procA();
```

```
END
```

```
CREATE PROCEDURE procA ()
```

```
DYNAMIC RESULT SETS 0
```

```
BEGIN
```

```
CALL procB();
```

```
END
```

```
CREATE PROCEDURE procB ()
```

```
BEGIN
```

```
DECLARE c_1 CURSOR WITH RETURN TO CALLER FOR
```

```
SELECT * FROM sysibm.sysdummy1;
```

```
DECLARE c_2 CURSOR WITH RETURN TO CLIENT FOR
```

```
SELECT * FROM sysibm.sysdummy1;
```

```
OPEN c_1;
```

```
OPEN c_2;
```

---

END

If the three procedures were built as shown in the exhibit, what will take place if an application invokes SQL procedure PROCMAIN?

- A. 0 result sets are returned.
- B. 1 result set is returned.
- C. 1 result set is returned with a warning.
- D. 2 result sets are returned with a warning.

**Answer: B**

**Explanation:**

**QUESTION NO: 35**

Click the Exhibit button.

```
CREATE PROCEDURE procB (INOUT p_parm1 INT, INOUT p_parm2 CHAR(5) )
```

```
BEGIN DECLARE v_1 INT DEFAULT 0;
```

```
DECLARE v_2 CHAR(5) DEFAULT '12345';
```

```
SET p_parm1 = v_1;
```

```
SET p_parm2 = v_2;
```

```
END@
```

```
CREATE PROCEDURE procA (IN p_parm1 INT, INOUT p_parm2 INT)
```

```
BEGIN DECLARE v_1 INT DEFAULT 0;
```

```
DECLARE v_2 INT DEFAULT 0;
```

```
SET v_1 = p_parm1;
```

```
CALL procB(v_1, v_2);
```

```
SET p_parm2 = v_2;
```

```
END@ CALL procA(1, 1)@
```

---

The file myscript.sql (shown in the exhibit) is executed from the CLP using the command:

```
db2 -td@ -vf myscript.sql
```

What is the expected outcome?

- A. SQL procedures PROC A and PROC B will be created, but the CALL command will fail.
- B. SQL procedure PROC A will not be created.
- C. SQL procedures PROC A and PROC B will be created and the CALL command will succeed.
- D. SQL procedure PROC B will not be created.

**Answer: B**

**Explanation:**

### QUESTION NO: 36

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**

**Explanation:**

#### **QUESTION NO: 37**

Which two DB2 privileges are required to issue a CREATE PROCEDURE statement, assuming the SCHEMA name used for the SQL procedure already exists? (Choose two.)

- A. EXECUTE privilege on any dependent SQL procedures
- B. BINDADD privilege on the database
- C. BIND privilege on the database
- D. SECADM authority on the database
- E. CREATEIN privilege on the schema

**Answer: B,E**

**Explanation:**

#### **QUESTION NO: 38**

Which two statements are true? (Choose two.)

- A. SQL procedures can contain static and/or dynamic SQL statements.
- B. Static or dynamic SQL execution is not associated with a package.
- C. The SQL procedure is always associated with a package that contains access paths of SQL statements in the procedure.
- D. It is necessary for an end-user to have object level privileges if that user has execute privileges on an associated package and the SQL procedure.
- E. SQL procedures can have COMMIT or ROLLBACK within atomic compound statements.

---

**Answer: A,C**

**Explanation:**

**QUESTION NO: 39**

Given the procedure shown below:

```
CREATE PROCEDURE proc ()  
BEGIN ATOMIC  
INSERT INTO mytable (col1) VALUES ('a');  
INSERT INTO mytable (col1) VALUES ('b');  
SIGNAL SQLSTATE '70000';  
INSERT INTO mytable (col1) VALUES ('c');  
END
```

How many rows will be inserted in the table?

- A. 0
- B. 1
- C. 2
- D. 3

**Answer: A**

**Explanation:**

**QUESTION NO: 40**

What is the correct order for declarations in a compound statement defined by BEGIN and END?

- A. variable, cursor, handler, condition
- B. variable, condition, cursor, handler
- C. variable, condition, handler, cursor
- D. variable, cursor, condition, handler

**Answer: B**

**Explanation:**

---

**QUESTION NO: 41**

Which SQL procedure declaration is valid?

**A.** CREATE PROCEDURE myproc(IN multp INTEGER, OUT p\_code INTEGER)  
BEGIN  
DECLARE SQLSTATE CHAR(5);  
DECLARE a INTEGER;  
DECLARE c\_duplicate CONDITION FOR SQLSTATE '23505';  
DECLARE my\_cur CURSOR FOR  
SELECT \* FROM employee;  
DECLARE EXIT HANDLER FOR SQLEXCEPTION  
SET p\_code = 0;  
SET a = a \* multp;  
DECLARE b INTEGER;  
b = a \*1.7;  
SET p\_code = b;  
END

**B.** CREATE PROCEDURE myproc(IN multp INTEGER, OUT p\_code INTEGER)  
BEGIN  
DECLARE EXIT HANDLER FOR SQLEXCEPTION  
SET p\_code = 0;  
DECLARE SQLSTATE CHAR(5);  
DECLARE a INTEGER;  
DECLARE c\_duplicate CONDITION FOR SQLSTATE '23505';  
DECLARE my\_cur CURSOR FOR  
SELECT \* FROM employee;  
SET a = a \* multp;  
SET p\_code = a;  
END

**C.** CREATE PROCEDURE myproc(IN multp INTEGER, OUT p\_code INTEGER)  
BEGIN  
DECLARE SQLSTATE CHAR(5);  
DECLARE a INTEGER;  
DECLARE my\_cur CURSOR FOR  
SELECT \* FROM employee;  
DECLARE c\_duplicate CONDITION FOR SQLSTATE '23505';  
DECLARE EXIT HANDLER FOR SQLEXCEPTION  
SET p\_code = 0;  
SET a = a \* multp;  
SET p\_code = a;  
END

**D.** CREATE PROCEDURE myproc(IN multp INTEGER, OUT p\_code INTEGER)

---

```

BEGIN
DECLARE SQLSTATE CHAR(5);
DECLARE a INTEGER;
DECLARE c_duplicate CONDITION FOR SQLSTATE '23505';
DECLARE my_cur CURSOR FOR
SELECT * FROM employee;
DECLARE EXIT HANDLER FOR SQLEXCEPTION
SET p_code = 0;
SET a = a * multp;
SET p_code = a;
END

```

**Answer: D**

**Explanation:**

### QUESTION NO: 42

Which two procedures demonstrate the correct use of dynamic SQL? (Choose two.)

- 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\_count3 (IN new\_count INTEGER, IN item\_code INTEGER)  
BEGIN  
DECLARE v\_dynSQL VARCHAR(200);  
SET v\_dynSQL = 'UPDATE stock SET quantity\_on\_hand=' || CHAR(new\_count) || ' WHERE item\_number=' || CHAR(item\_code);  
EXECUTE IMMEDIATE v\_dynSQL;

---

```

END
D. 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
E. 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,C**

**Explanation:**

### QUESTION NO: 43

Click the Exhibit button.

```

CREATE PROCEDURE proc_labels (IN start_value INT, OUT p_ID INT) s1: BEGIN
DECLARE v_ID INT;
s12: BEGIN DECLARE v_ID INT;
SET v_ID = start_value;
SET s1.v_ID = v_ID + s12.v_ID;
SET v_ID = 3;
SET p_ID = s12.v_ID;
END;

```

To Read the [Whole Q&As](#), please purchase the [Complete Version](#) from [Our website](#).

# Trying our product !

- ★ **100%** Guaranteed Success
- ★ **100%** Money Back Guarantee
- ★ **365 Days** Free Update
- ★ **Instant Download** After Purchase
- ★ **24x7** Customer Support
- ★ Average **99.9%** Success Rate
- ★ More than **69,000** Satisfied Customers Worldwide
- ★ Multi-Platform capabilities - **Windows, Mac, Android, iPhone, iPod, iPad, Kindle**

## Need Help

Please provide as much detail as possible so we can best assist you.

To update a previously submitted ticket:



 <b>One Year Free Update</b> <p>Free update is available within One Year after your purchase. After One Year, you will get 50% discounts for updating. And we are proud to boast a 24/7 efficient Customer Support system via Email.</p>	 <b>Money Back Guarantee</b> <p>To ensure that you are spending on quality products, we provide 100% money back guarantee for 30 days from the date of purchase.</p>	 <b>Security &amp; Privacy</b> <p>We respect customer privacy. We use McAfee's security service to provide you with utmost security for your personal information &amp; peace of mind.</p>
---	---	--

## [Guarantee & Policy](#) | [Privacy & Policy](#) | [Terms & Conditions](#)

Any charges made through this site will appear as Global Simulators Limited.

All trademarks are the property of their respective owners.

Copyright © 2004-2015, All Rights Reserved.