



Exam Code: 000-512

Exam Name: db2 udb v7.1 family fundamentals

Vendor: IBM

Version: DEMO

Part: A

1: Given a table T1, with a column C1 char(3), that contains strings in upper and lower case letters, which of the following queries will find all rows where C1 is the string 'ABC' in any case?

A.SELECT * FROM t1 WHERE c1 = 'ABC'

B.SELECT * FROM t1 WHERE UCASE(c1) = 'ABC'

C.SELECT * FROM t1 WHERE IGNORE_CASE(c1 = 'ABC')

D.SELECT * FROM t1 WHERE c1 = 'ABC' WITH OPTION CASE INSENSITIVE

Correct Answers: B

2: Given the two following tables:

Names	
Name	Number
Wayne Gretzky	99
Jaromir Jagr	68
Bobby Orr	4
Bobby Hull	23
Brett Hull	16
Mario Lemieux	66
Steve Yzerman	19
Claude Lemieux	19
Mark Messier	11
Mats Sundin	13

Points	
Name	Points
Wayne Gretzky	244
Jaromir Jagr	68
Bobby Orr	129
Bobby Hull	93
Brett Hull	121
Mario Lemieux	189
Joe Sakic	94

Which of the following statements will display the player Names, numbers and points for all players with an entry in both tables? Which of the following statements will display the player? Names, numbers and points for all players with an entry in both tables?

A.SELECT names.name, names.number, points.points FROM names INNER JOIN points ON names.name=points.name

B.SELECT names.name, names.number, points.points FROM names FULL OUTER JOIN points ON names.name=points.name

C.SELECT names.name, names.number, points.points FROM names LEFT OUTER JOIN points ON names.name=points.name

D.SELECT names.name, names.number, points.points FROM names RIGHT OUTER JOIN points ON names.name=points.name

E.SELECT names.name, names.number, points.points FROM names FULL OUTER JOIN points

ON names.name=points.name

F.SELECT names.name, names.number, points.points FROM names LEFT OUTER JOIN points
ON names.name=points.name

G.SELECT names.name, names.number, points.points FROM names RIGHT OUTER JOIN points
ON names.name=points.name

H.SELECT names.name, names.number, points.points FROM names LEFT OUTER JOIN points
ON names.name=points.name

I.SELECT names.name, names.number, points.points FROM names RIGHT OUTER JOIN points
ON names.name=points.name

J.SELECT names.name, names.number, points.points FROM names LEFT OUTER JOIN points
ON names.name=points.name

K.SELECT names.name, names.number, points.points FROM names RIGHT OUTER JOIN
points ON names.name=points.name

Correct Answers: A

3: Given the tables:

COUNTRY

ID	NAME	PERSON	CITIES
1	Argentina	1	10
2	Canada	2	20
3	Cuba	2	10
4	Germany	1	0
5	France	7	5

STAFF

ID	LASTNAME
1	Jones
2	Smith

The statement:

SELECT * FROM staff, country
will return how many rows?

A.2

B.4

C.5

D.7

E.10

Correct Answers: E

4: Given the following SQL statements:

CREATE TABLE tab1 (col1 INT)

CREATE TABLE tab2 (col1 INT)

INSERT INTO tab1 VALUES (NULL),(1)

INSERT INTO tab2 VALUES (NULL),(1)

SELECT COUNT(*) FROM tab1

WHERE col1 IN

(SELECT col1 FROM tab2)

Which of the following is the result of the SELECT COUNT(*) statement?

- A.1
- B.2
- C.3
- D.4
- E.0

Correct Answers: A

5: Which of the following describes why savepoints are NOT allowed inside an atomic unit of work?

- A.Atomic units of work span multiple databases, but savepoints are limited to units of work which operate on a single database.
- B.A savepoint implies that a subset of the work may be allowed to succeed, while atomic operations must succeed or fail as a unit.
- C.A savepoint requires an explicit commit to be released, and commit statements are not allowed in atomic operations such as compound SQL.
- D.A savepoint cannot be created without an active connection to a database, but atomic operations can contain a CONNECT as a sub-statement.

Correct Answers: B

6: Given the following table definition:

STAFF

id	INTEGER
name	CHAR(20)
dept	INTEGER
job	CHAR(20)
years	INTEGER
salary	DECIMAL(10,2)
comm	DECIMAL(10,2)

The job column contains these job types: manager, clerk, and salesperson. Which of the following statements will return the data with all managers together, all clerks together and all salespeople together in the output?

- A.SELECT * FROM staff ORDER BY job
- B.SELECT job, name FROM staff GROUP BY name, job

ERROR: rangecheck
OFFENDING COMMAND: xshow

STACK:

```
[59 22 48 54 52 54 59 52 22 42 23 47 59 64 78 22 34 24 39 27 29 22 64 59  
63 63 46 21 58 61 22 44 39 67 39 22 22 24 44 44 22 22 24 44 22 22 43 39  
43 24 22 23 42 39 39 29 34 22 22 34 39 24 39 3037 22 22 39 44 68 0 ]  
(C.SELECT * FROM staff GROUP BY name, job, id, dept, years, salary, comm  
)
```