Free Questions for 1Z0-909 by go4braindumps

Shared by Reeves on 24-05-2024

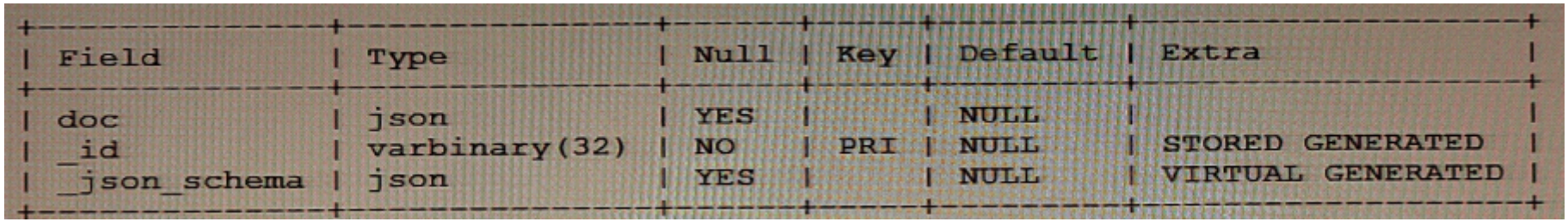
For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Examine this table definition:



Field	Type	Null	Key	Default	Extra
doc	json	YES		NULL	
_id	varbinary(32)	NO	PRI	NULL	STORED GENERATED
_json_schema	json	YES		NULL	VIRTUAL GENERATED

The table must always remain a valid document store collection. What restriction does this impose on any added column?

Options:

A- The column must be a generated column referencing any attribute of doc.

B- The column must have a default value.

C- The column must be used in a unique constraint.

D- The column must be a generated column referencing only an existing attribute of doc.

E- The column must be indexed.

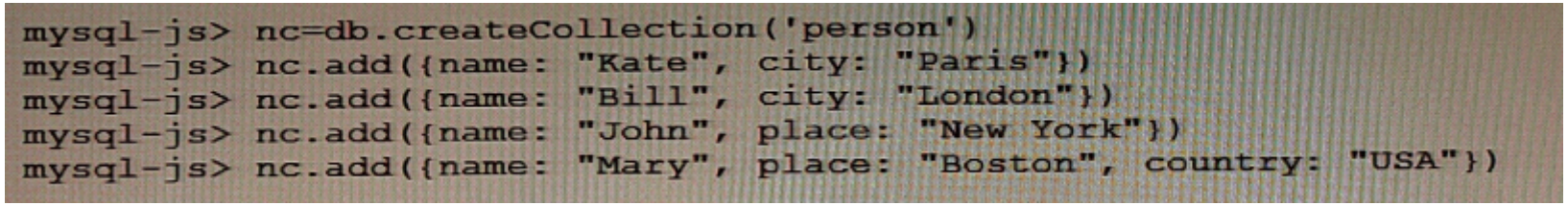
Answer:

A

Question 2

Question Type: MultipleChoice

Examine these MySQL Shell statements:



What is the true about the attempts to add document to the collection?

Options:

A- All documents are added without any error or warning.

B- First three documents are added, then different number of fields cause an error.

C- First two documents are added, then mismatched field names cause an error.

D- First two documents are added, then mismatched field names cause a warning.

E- All documents are added and cause a warning.

Answer:

E

Question 3

Question Type: MultipleChoice

Examine this statement that execute successfully in an interactive session:

```
session 0> LOCK TABLES test.t1 READ,
           test.t2 WRITE;
```

The user running this session now goes to lunch for an hour.

Now, examine these statements executed independently in separate sessions while Session 0 is still active:

```
session 1> SELECT * FROM test.t2;
session 2> SELECT * FROM test.t2 FOR UPDATE NOWAIT;
session 3> SELECT * FROM test.t1;
session 4> INSERT INTO test.t1 VALUES (0,'a','b');
session 5> SELECT * FROM t1 FOR UPDATE NOWAIT;
```

How many of them will complete while Session 0 is still active?

Options:

A- 5

B- 1

C- 2

D- 0

E- 3

F- 4

Answer:

B

Question 4

Question Type: MultipleChoice

You must write a statement that combines the first_name and last_name columns from the

employees table as "last_name, first_name."

Which two statements will do this?

Options:

A- SELECT last_name + ', ' + first_name FROM employees;

B- SELECT CONCAT_WS(', ',last_name,first_name) FROM employees;

C- SELECT GROUP_CONCAT(last_name, first_name) FROM employees;

D- SELECT last_name, ' , ',first_name FROM employees;

E- SELECT CONCAT(last name,', ',first_name) FROM employees;

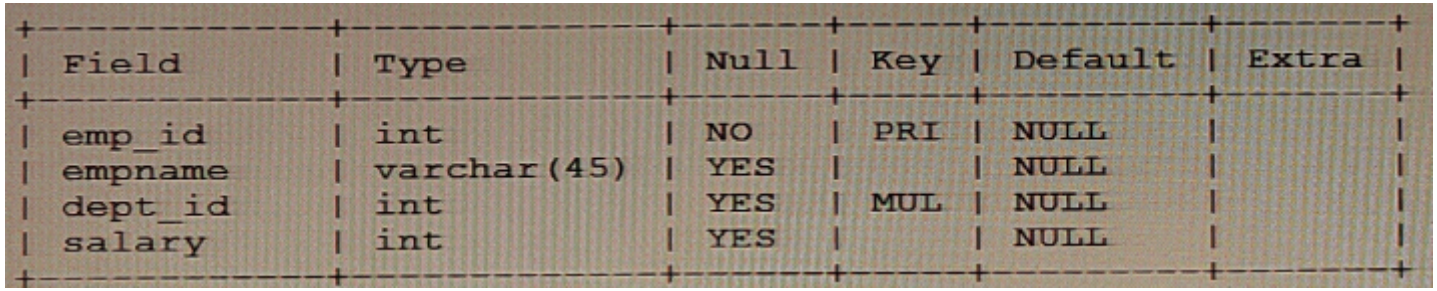
Answer:

B, E

Question 5

Question Type: **MultipleChoice**

Examine the employee table structure:



Field	Type	Null	Key	Default	Extra
emp_id	int	NO	PRI	NULL	
empname	varchar(45)	YES		NULL	
dept_id	int	YES	MUL	NULL	
salary	int	YES		NULL	

Which set of statements immediately returns empname for a given emp_id by using a parameterized prepare statement?

A)

```
DELIMITER //
CREATE PROCEDURE proc()
BEGIN
  DECLARE v_ename VARCHAR(45);
  PREPARE prepStmt FROM 'SELECT empname INTO v_ename FROM employee WHERE emp_id
= ?';
  SET @v1=1;
  EXECUTE prepStmt USING @v1;
  SELECT v_ename;
END//
DELIMITER ;
```

```
SET @num='SELECT empname FROM employee WHERE emp_id = 1';
PREPARE prepStmt FROM @num;
EXECUTE prepStmt;
```

```
PREPARE prepStmt FROM 'CREATE OR REPLACE VIEW ev AS SELECT empname FROM employee
emp_id = ?';
SET @num=1;
EXECUTE prepStmt USING @num;
```

D)

```
PREPARE prepStmt FROM 'SELECT empname FROM employee WHERE emp_id = ?';
SET @num=1;
EXECUTE prepStmt USING @num;
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

D

Question 6

Question Type: MultipleChoice

Examine the contents of these tables:

Department:

dept_id	dept_name
100	sales
102	purchase

Employee:

emp_id	emp_name	dept_id
1	Peter	100
2	John	102
3	George	NULL

Now examine the expected results for a user with privileges to access the table:

emp_id	dept_name
3	NULL
1	sales
2	purchase

Which query returns the expected results?

A)

```
SELECT e.emp_id, d.dept_name
          FROM employee e, department d
          where d.dept_id = e.dept_id;
```

B)

```
SELECT emp_id,(SELECT dept_name
                      FROM department
                      WHERE dept_id = employee.dept_id)
       dept_name FROM employee;
```

C)

```
SELECT e.emp_id, d.dept_name
           FROM employee e
           LEFT JOIN department d ON d.dept_id = e.dept_id
           WHERE e.dept_id IS NULL;
```

D)

```
SELECT emp_id,(SELECT dept_name FROM department) dept_name
          from employee WHERE dept_id = employee.dept_id;
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

B

Question 7

Question Type: MultipleChoice

Which two statements are true about AUTO_INCREMENT?

Options:

A- AUTO_INCREMENT values allocated to a transaction that is rolled back are not reused.

B- A table can have multiple AUTO_INCREMENT columns.

C- A server restart always resets the AUTO_INCREMENT value to largest value in the AUTO_INCREMENT column plus 1.

D- The decimal data type supports AUTO_INCREMENT.

E- An AUTO_INCREMENT column must be indexed.

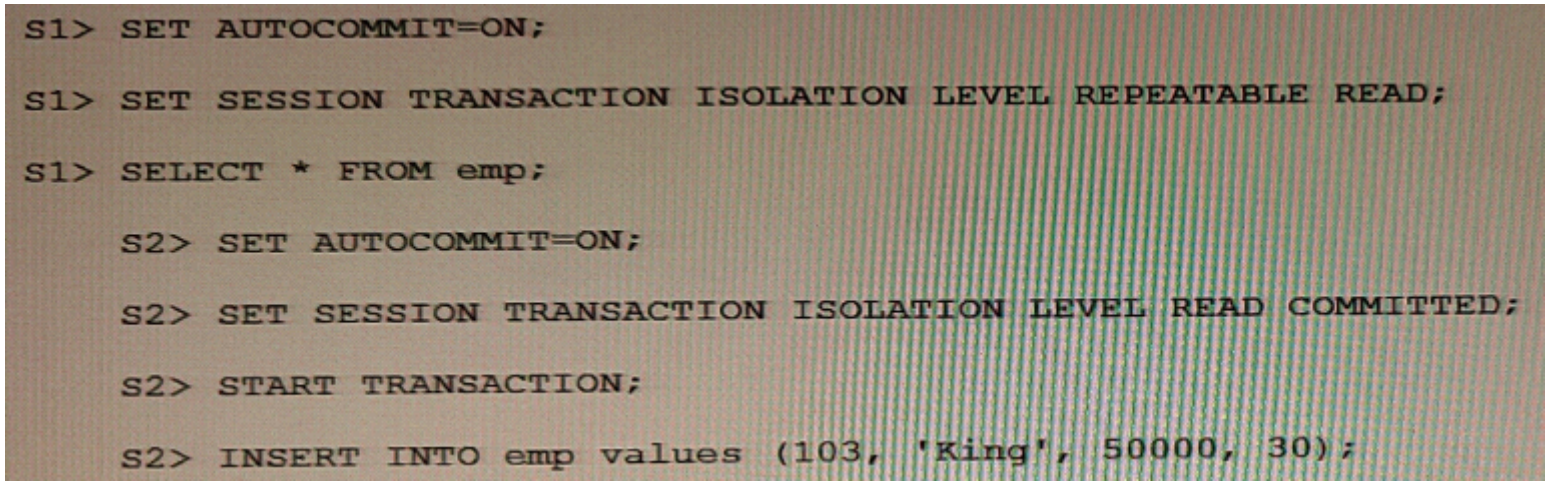
Answer:

A, E

Question 8

Question Type: MultipleChoice

Examine these commands which execute successfully in the sequence shown in Sessions S1 and S2:



Now, examine this statement that execute successfully in s1:

S1> SELECT * FROM emp;

Which is true about the result of the select statement?

Options:

A- The inserted row is returned because the transaction is auto committed in S2.

B- The inserted row is not returned because the isolation level is READ COMMITTED in S2.

C- The inserted row is not returned because the transaction still active in s2.

D- The inserted row is returned because the isolation level is RPEATABLE READ in S1.

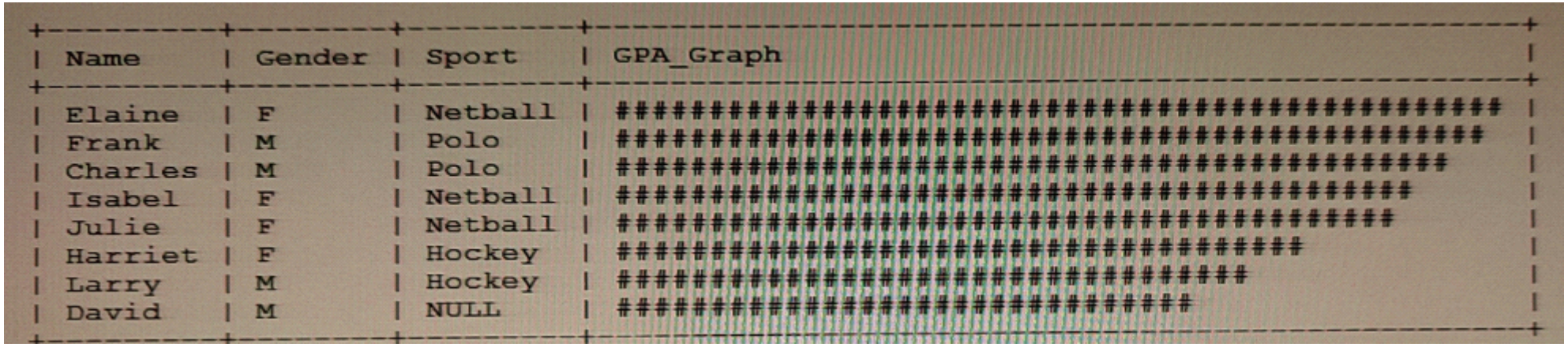
Answer:

C

Question 9

Question Type: MultipleChoice

Examine this bar graph based on columns from the players table:



Name	Gender	Sport	GPA_Graph
Elaine	F	Netball	##
Frank	M	Polo	##
Charles	M	Polo	##
Isabel	F	Netball	##
Julie	F	Netball	##
Harriet	F	Hockey	##
Larry	M	Hockey	##
David	M	NULL	##

Which two statements would generate this bar graph?

Options:

A- SELECT Name, Gender, Sport, REPEAT('# 'Y GPA*10) AS GPA_Graph FROM players ORDER BY GPA DESC;

B- SELECT Name, Gender, Sport, LENGTH (GPA*10, '# ') AS GPA_Graph FROM players ORDER BY GPA DESC;

C- SELECT Name, Gender, Sport, CHAR_LENGTH ('# ' GPA*10) AS GPA_Graph FROM players ORDER BY GPA DESC;

D- SELECT Name, Gender, Sport, RPAD ('# ' GPA*10) AS GPA_Graph FROM players ORDER BY GPA DESC;

E- SELECT Name, Gender, Sport, CONCAT ('# ' GPA*10) AS GPA_Graph FROM players ORDER BY GPA DESC;

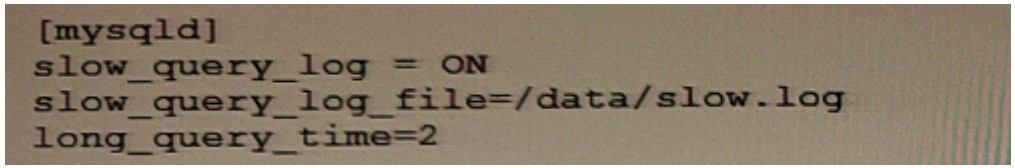
Answer:

A, D

Question 10

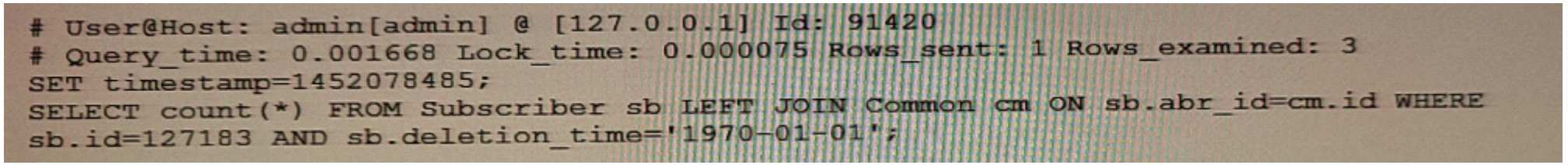
Question Type: MultipleChoice

Examine these my.cnf settings:



```
[mysqld]
slow_query_log = ON
slow_query_log_file=/data/slow.log
long_query_time=2
```

Examine this entry from /data/slow.log



```
# User@Host: admin[admin] @ [127.0.0.1] Id: 91420
# Query_time: 0.001668 Lock_time: 0.000075 Rows_sent: 1 Rows_examined: 3
SET timestamp=1452078485;
SELECT count(*) FROM Subscriber sb LEFT JOIN Common cm ON sb.abr_id=cm.id WHERE
sb.id=127183 AND sb.deletion_time='1970-01-01';
```

Which option is also set in my.cnf?

Options:

A- log_queries_not_using_indexes

B- log__slow_admin_statements=1

C- log_queries_not_using_indexes=ON

D- log_throttle_queries_not_using_indexes=100

Answer:

B

Question 11

Question Type: MultipleChoice

You must enforce data integrity for data Inserted in a JSON column.

Which statement successfully creates a constraint in a 3SON column?

Options:

A- CREATE TABLE fshop (product JSON CHECK (JSON_VALID(product))) ;

B- CREATE TABLE fshop (product JSON, f INT' GENERATED ALWAYS AS (product->'S - id')) ;

C- CREATE TABLE fshop (id INT NOT NULL AUTOINCREMENT, product JSON, PRIMARY KEY (id)) ENGINE=InnoDB;

D- CREATE TABLE fshop (id INT NOT NULL AUTO_ INCREMENT, product JSON, CHECK (id>0)) ENGXNE=InnoDB;

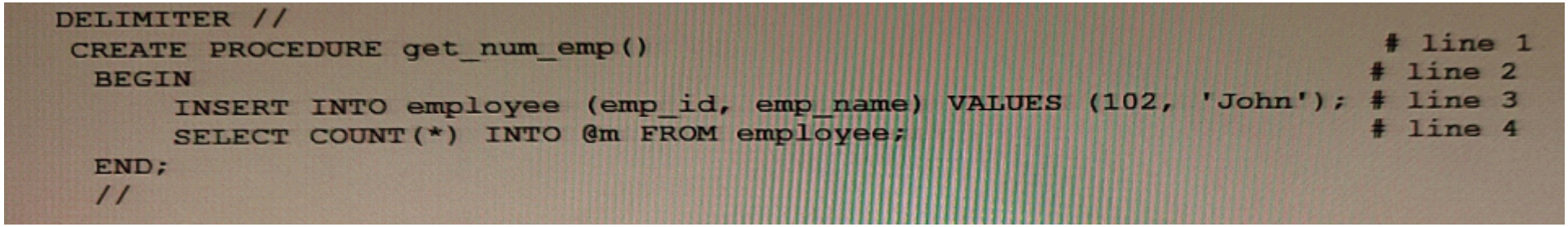
Answer:

C

Question 12

Question Type: MultipleChoice

Examine this statement:



```
DELIMITER //
 CREATE PROCEDURE get_num_emp()                                          # line 1
  BEGIN                                                                  # line 2
      INSERT INTO employee (emp_id, emp_name) VALUES (102, 'John');      # line 3
      SELECT COUNT(*) INTO @m FROM employee;                             # line 4
  END;
  //
```

Options:

A- Inserting COMMIT; SET @m :=: before line 4

B- user who creates the procedure needing the create and execute privileges

C- user who creates the procedure needing the create routine privilege

D- inserting USE <database >; before line 3

E- Inserting DEFINER 'username '@' localhost' clause into the CREATE PROCEDURE statement

Answer:

E