

Oracle.1z1-909.v2022-12-22.q21

Exam Code:	1z1-909
Exam Name:	MySQL 8.0 Database Developer
Certification Provider:	Oracle
Free Question Number:	21
Version:	v2022-12-22
# of views:	695
# of Questions views:	210
https://www.freepdfdumps.com/Oracle.1z1-909.v2022-12-22.q21.html	

NEW QUESTION: 1

The employee table includes these columns:

e_id INT, e_name VARCHAR (45), dept_id INT, salary INT

You must create a stored function, getMaxSalary(), which returns the maximum salary paid for a given department id.

Which statement will create the function?

A)

```
CREATE FUNCTION getMaxSalary(v_dept_id INT) RETURNS CHAR(50)
BEGIN
    SELECT MAX(salary) INTO @m FROM employee WHERE dept_id = v_dept_id;
    RETURN m;
END
```

B)

```
CREATE FUNCTION getMaxSalary(v_dept_id INT) RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE msalary INT;
    USE schemaName;
    SELECT MAX(salary) INTO msalary FROM employee WHERE dept_id = v_dept_id;
    RETURN msalary;
END
```

C)

```
CREATE FUNCTION getMaxSalary(v_dept_id INT) RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE msalary INT;
    SELECT MAX(salary) INTO msalary FROM employee WHERE dept_id = v_dept_id;
    RETURN msalary;
END
```

D)

```
CREATE FUNCTION getMaxSalary(v_dept_id INT) RETURNS INT
DECLARE msalary INT;
BEGIN
    SELECT MAX(salary) INTO msalary FROM employee WHERE dept_id = v_dept_id;
    getMaxSalary := msalary;
    RETURN;
END
```

- A. Option D
- B. Option C
- C. Option A
- D. Option B

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 2

The collection col contains all episodes for all seasons for a TV show.

Examine this document which has an example of the details for each episode:

```
{
  "_id": "00005cbee2d1000000000000000001",
  "name": "Days Gone Bye",
  "number": 1,
  "season": 1,
  "airdate": "2010-10-31",
  "airtime": "22:00",
  "runtime": 60
}
```

Which query returns all episode names from the first season?

- A. SELECT doc->\$.name, FROM col WHERE doc->"\$.season" = "1";
- B. SELECT doc->"\$.name" FROM col WHERE doc->"\$.season" = "1";
- C. SELECT name FROM col WHERE season = 1;
- D. SELECT "S.name" FROM col WHERE "S.season" = "1";

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 3

Examine this statement and output:

```
SELECT * FROM fshop;
+-----+
|
| product
|
+-----+
| {"name": "apple", "varieties": [{"Origin": ["AF", "BY"], "VarietyName":
"Gala"},
| [{"Origin": ["PT", "ES"], "VarietyName": "RED"}, {"Origin": ["FR", "UK"], "VarietyName":
"Yel"},
| [{"Origin": ["IT", "HG"], "VarietyName":
"Jon"}]}
```

You execute this statement:

```
SELECT JSON_SEARCH(product,'one','IT') FROM fshop ;
```

What is the output?

- A. "\$.varieties[4]. origin[1]"
- B. product->varieties[3]. origin[0]"

C. product->"\$.varieties [4]. origin[1];;

D. "\$.varieties [3]. origin[0]"

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 4

Examine these statements which execute successfully:

```
CREATE TABLE `users` (
  `user_id` int(11) NOT NULL AUTO_INCREMENT,
  `loc_id` int(11) DEFAULT NULL,
  `user_name` varchar(50) NOT NULL,
  `user_static` int(11) NOT NULL DEFAULT '0'
  PRIMARY KEY (`user_id`)
) ENGINE=InnoDB AUTO_INCREMENT=4968107 DEFAULT CHARSET=latin1

CREATE TABLE `locations` (
  `loc_id` int(11) NOT NULL AUTO_INCREMENT,
  `site_id` int(11) NOT NULL,
  `loc_name` varchar(50) NOT NULL,
  `loc_shared` int(11) NOT NULL DEFAULT '0',
  `loc_mapping` char(36) NOT NULL,
  PRIMARY KEY (`loc_id`)
) ENGINE=MEMORY AUTO_INCREMENT=6835 DEFAULT CHARSET=latin1

SELECT
  loc.site_id,
  loc.loc_shared,
  usr.user_name
FROM users usr
INNER JOIN locations loc
ON usr.loc_id = loc.loc_id
WHERE
  loc.loc_mapping = 'daa9a225-8a4d-11ea-b3cf-00059a3c7a00'
```

Which two changes will improve this query performance?

- A. CREATE INDEX 1X4 ON Locations (site_id, loc_shared);
- B. CREATE INDEX 1X6 ON users (user_name);
- C. CREATE INDEX 1X3 ON locations <loc_site_id> ;
- D. CREATE INDEX 1X2 ON locations (loc_mapping) USING HASH; fh
- E. CREATE INDEX 1X7 ON users (user_name) USING HASH;
- F. CREATE INDEX 1X5 ON users (loc_id);
- G. CREATE INDEX IX1 ON locations (loc_shareci) ;

Answer: B,E ([LEAVE A REPLY](#))

NEW QUESTION: 5

How does InnoDB choose deadlock victims in MySQL?

- A. It chooses the transaction randomly.
- B. It chooses the transaction with the least accumulated CPU time.
- C. It chooses the transaction with the most modified rows.
- D. It chooses the transaction with the most accumulated CPU time.
- E. It chooses the transaction with the lowest transaction ID.

F. It chooses the transaction with the fewest modified rows.

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 6

Your session has `sqi_mode` set to default.

Examine this statement which executes successfully:

```
CREATE TABLE students (  
  std_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  firstname varchar(255) NOT NULL,  
  lastname varchar(255) NOT NULL,  
  birthdate date NOT NULL,  
  reg_date datetime NOT NULL  
) ENGINE=InnoDB AUTO_INCREMENT=10300;
```

Now examine this statement:

```
INSERT INTO students (std_id, firstname, lastname, birthdate, reg_date)  
VALUES ("NULL", "Mary", "O'Hagen", '1997-11-26', DATE());
```

Which two changes are required to the insert statement so that it inserts the correct data?

- * `std_id` = 10301
- * `firstname` = Mary
- * `lastname` = O'Hagen
- * `birthdate` = November 26, 1997
- * `reg_date` = the current date

- A. Change " NULL " to NULL.
- B. Change `date ()` to `CURRENT_TIMESTAMP ()` .
- C. Change `DATE ()` to `DAY ()`.
- D. Change "O'Hagen" to 'o'Hagen'.
- E. Change " NULL " to ' NULL ' .
- F. Change "O'Hagen" to "O'Hagen".

Answer: A,C ([LEAVE A REPLY](#))

NEW QUESTION: 7

Examine these statements issued from Session 1 which execute successfully:

```
Session 1>  
  
SET autocommit=1;  
SELECT * FROM band FOR UPDATE;
```

Now, examine these statements issued from Session 2 which execute successfully:

Session 2>

BEGIN;

UPDATE band SET song=CONCAT ("Here Comes the ", song) WHERE song LIKE ' %Sun ;

Which two are true?

- A. Session 1 does not block Session 2.

- B. Session 1 takes a shared lock on all the rows in the band table.
- C. Statements in Session 2 are committed.
- D. Session 2 does not start a transaction.
- E. Session 2 takes an exclusive lock on all the rows in the band table.
- F. Session 1 must commit before the update in Session 2 can complete.

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 8

Examine the appointments table definition which contains one million rows:

```
CREATE TABLE `appointments` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `attendant_id` int(11) NOT NULL,
  `attendant_session_id` int(11) NOT NULL,
  `start` datetime NOT NULL,
  `end` datetime NOT NULL,
  `date` date NOT NULL,
  `created_by` varchar(20) NOT NULL,
  `created_at` datetime DEFAULT CURRENT_TIMESTAMP,
  `payment` int(11) NOT NULL DEFAULT '0',
  `credit` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`),
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Now, examine this statement which executes successfully:

```
SELECT attendant_id,
       payment,
       credit
FROM appointments
WHERE attendant_session_id = 510
AND created_by = 'jsmith'
```

Which statement will improve query performance?

- A. ALTER TABLE appointments add index IX_2(attendant_session_id, created_by)
- B. ALTER TABLE appointments add index IX_1(credit,payment)
- C. ALTER TABLE appointments add index IX_4<attendant_id, payment, credit)
- D. ALTER TABLE appointments add index IX_3(attendant_id, created_by)

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 9

Examine the statement which executes successfully:

```
SET sql_mode='NO_ENGINE_SUBSTITUTION';
```

You try to create a table with a storage engine that is not available. What will happen?

- A. The server will create the table but it will be unusable until the specified storage engine is available.
- B. The server will create the table but report an error when the first attempt to insert a row is performed.
- C. The server will create the table using the default storage engine.
- D. An error occurs and the create table statement fails.

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 10

Examine the structure of the emp table:

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PR1	NULL	auto_increment
name	varchar(25)	YES		NULL	
SALARY	int(11)	YES		NULL	
email	varchar(25)	YES		NULL	

Examine the structure of the emp_vu1 view based on the emp table:

Field	Type	Null	Key	Default	Extra
name	varchar(25)	YES		NULL	
salary	int(11)	YES		NULL	

Now, examine this statement:

```
mysql> INSERT INTO emp_vu1 VALUES ('Alice',20000) ;
```

What is true about executing the statement?

- A. It returns an error because an insert operation is not allowed on views.
- B. It returns an error because the PRIMARY ACCOUNT column is not selected for the view definition.
- C. It inserts a row in the emp table.
- D. It inserts a row in the view only.

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 11

Examine this statement which executes successfully:

```
CREATE TABLE `work` (  
  `job_no` INT NOT NULL,  
  `data` JSON NOT NULL,  
  `name` VARCHAR(30) GENERATED ALWAYS AS (JSON_EXTRACT(`data`, '$.first_name'))  
  VIRTUAL,  
  PRIMARY KEY (`job_no`)  
) ENGINE = INNODB;
```

The table is populated with a range of values including jobs for Robert, John, and Katie. Now, examine this statement and output:

```
SELECT job_no, name  
FROM work  
WHERE name = 'Robert'  
Empty set (0.0007 sec)
```

Why is an empty result set returned?

- A. The json_extract() function requires a length value that matches the field length in the schema.
- B. The JSON datatype cannot be used in virtual columns.
- C. The select requires json_unquote in the where clause.
- D. Table statistics must be updated to generate values for the name column.
- E. The virtual values in the name column must be accessed using functions.

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 12

Examine these statements:

```
SET collation_connection=utf8mb4_0900_as_cs;
```

```
SELECT STRCMP(Alice', UCASE ('Alice* )) ;
```

What is displayed?

- A. 0
- B. 1
- C. NULL
- D. -1
- E. ERROR: 1267 (HY000): Illegal mix of collations

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 13

Which change will prevent negative ages to be inserted into the people table?

A. DELIMITER //

```
CREATE TRIGGER agecheck BEFORE INSERT ON people FOR EACH ROW IF NEW. age < 0  
THEN SET NEW.age =0; END IF;// DELIMITER ;
```

B. ALTER TABLE people ADD CONSTRAINT check_age CHECK (ABS(age)>=0);

C. ALTER TABLE people ADD COLUMN valid_age=ABS(check_age) GENERATED ALWAYS;

D. DELIMITER //

```
CREATE TRIGGER age check AFTER INSERT ON people FOR EACH ROW IF NEW. age < 0  
THEN SET NEW.age =0; END IF;// DELIMITER;
```

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 14

A server hosts MySQL Server and Apache Webserver supporting a PHP/PDO based application. The application must be migrated from PHP to their Java application server on another host. The MySQL instance remains on the original host.

Examine the PDO connection string used in the existing application:

```
Mysql:host=localhost;dbname=sales;unix_socket=/var/run/mysql.sock
```

Which two prevent Java from using the Unix socket?

- A. The X Dev API protocol must be enabled to use sockets in Connector/J driver.
- B. The socket can only be accessed from the local host.
- C. socket is a reserved word in Java.
- D. Java treats the socket file as insecure.
- E. The socket is not implemented in Connector/J driver.

Answer: D,E ([LEAVE A REPLY](#))

NEW QUESTION: 15

The variables c and d are declared as integer types.

Examine these initialization statements with placeholder value <p1>, which execute successfully:

Now, examine this loop which executes successfully:

```
REPEAT
  SET d = d + 1;
  SET c = c + 1;
UNTIL c > 3
END REPEAT;
```

Which loop results in the same value of d for all valid values of <p1>?

A)

```
b: LOOP
  SET d = d + 1;
  SET c = c + 1;
  IF c > 3 THEN LEAVE b; END IF;
END LOOP b;
```

B)

```
WHILE c <= 3 DO
  SET d = d + 1;
  SET c = c + 1;
END WHILE;
```

C)

```
b: LOOP
  IF c > 3 THEN LEAVE b; END IF;
  SET d = d + 1;
  SET c = c + 1;
END LOOP b;

WHILE c < 3 DO
  SET d = d + 1;
  SET c = c + 1;
END WHILE;
```

A. Option C

B. Option A

C. Option D

D. Option B

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 16

Examine these statements and output:

```
mysql> SET AUTOCOMMIT=on;
Query OK, 0 rows affected (0.01 sec)

mysql> UPDATE emp
  -> SET salary=24000
  -> WHERE id=101;
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EMP values (102,'John',13000,'jj',10);
Query OK, 1 row affected (0.00 sec)

mysql> SET AUTOCOMMIT=off;
Query OK, 0 rows affected (0.01 sec)
```

Now, examine this command:

Mysql> ROLLBACK;

What is true about the effect of the command?

- A. It undoes the insert command.
- B. It returns an error because there is no active transaction.
- C. It has no effect.
- D. It undoes both insert and update commands.
- E. It undoes the update command.

Answer: A ([LEAVE A REPLY](#))

Valid 1z1-909 Dumps shared by Actual4test.com for Helping Passing 1z1-909 Exam!
Actual4test.com now offer the **newest 1z1-909 exam dumps**, the Actual4test.com 1z1-909 exam **questions have been updated** and **answers have been corrected** get the **newest** Actual4test.com 1z1-909 dumps with Test Engine here:

https://www.actual4test.com/1z1-909_examcollection.html (66 Q&As Dumps, **30%OFF**)

Special Discount: **Freepdfdumps**)

NEW QUESTION: 17

Examine this statement which has executed successfully:

```
CREATE TABLE `film_text` (
  `film_id` smallint NOT NULL,
  `title` varchar(255) NOT NULL,
  `description` text,
  PRIMARY KEY (`film_id`),
  FULLTEXT KEY `description_idx` (`description`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

film_text contains millions of rows.

Now, examine this statement:

```
SELECT title
FROM film_text
WHERE description LIKE "Scientist%";
```

- A. Execution performance can be improved by adding an index on column description.
- B. Execution performance can be improved by, using a composite index with column description as the leftmost prefix column description.
- C. No index will improve statement performance.
- D. Execution performance can be improved by using like instead of RLIKE.
- E. The statement takes advantage of index description_idx.

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 18

Examine these statements issued from Session 1 which execute successfully:

```
Session 1> SET transaction_isolation = 'SERIALIZABLE';
Session 1> START TRANSACTION;
Session 1> SELECT * FROM world.city WHERE name='Roma' AND CountryCode='ITA';
```

Now, examine this statement issued from Session 2:

```
Session 2> UPDATE world.city SET population=2660000 WHERE name='Roma' AND
CountryCode='ITA';
```

What is the outcome of the update statement in Session 2?

- A. The statement will wait for the transaction in Session 1 to finish.
- B. The row will be updated immediately.
- C. The transaction in Session 1 will be rolled back automatically.
- D. A deadlock will occur.

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 19

Examine the Test.php script which is numbered for clarity, and its output:

```
1. <?PHP
2. $link = mysqli_connect("localhost", "username", "password", "schema");
3. $sql = "SELECT actor_id, first_name, last_name FROM actor";
4. $result = $link->query($sql);
5. echo $result->num_rows;
6. ?>
```

PHP Fatal error: Uncaught Error: call to underlined function mysqli_connect () in Test.php:2 Which action will fix this error?

- A. Replace line 2 with Slink = mysql.connect {"localhost: 3306n, "username", "pas "schema") ;
- B. Enable the mysqli extension in the php.ini file.
- C. Install the PHP executable in the path used by the MySQL installation.
- D. Replace line 2 With: Slink = mysql_xdevapi\getSession ("mysqlx://username:password@localhost:3306","schema");

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 20

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?

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

Answer: A,D ([LEAVE A REPLY](#))

NEW QUESTION: 21

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?

- A. 4
- B. 2
- C. 5
- D. 1
- E. 0
- F. 3

Answer: A ([LEAVE A REPLY](#))

Valid 1z1-909 Dumps shared by Actual4test.com for Helping Passing 1z1-909 Exam!
Actual4test.com now offer the **newest 1z1-909 exam dumps**, the Actual4test.com 1z1-909 exam **questions have been updated** and **answers have been corrected** get the **newest** Actual4test.com 1z1-909 dumps with Test Engine here:

https://www.actual4test.com/1z1-909_examcollection.html (**66** Q&As Dumps, **30%OFF**

Special Discount: Freepdfdumps)