



CSS326: Database Programming Laboratory

Final Mock Exam

curated by The Peanuts

Name Nonprawich I. ID 6622772422 Section..... Seat No.....

Conditions: Semi-closed Book

Directions:

1. This exam has 19 pages (including this page).
2. Write your name clearly at the top.
3. All exam queries are guaranteed to run without syntax errors. (We tested them. Probably. Maybe. Maybe not)
4. Show your work where applicable. If your answer appears without steps, you'll get zero.
5. Answers must be written in English. MySQL does not support `LANGUAGE = CHAOTIC_SIIT_SLANG`.
6. No communication is allowed. If you whisper to your friend, they may reply: "ERROR 1142 (42000): SELECT command denied to user".

For solution, [click here](#).

Part I: Multiple Choice Questions

1. Which of the following is NOT a valid MySQL aggregate function?

- a) COUNT()
- b) AVG()
- c) CONCAT()
- d) SUM()

2. What does the DELIMITER command do in MySQL?

- a) It deletes records from a table
- b) It changes the statement terminator character
- c) It creates a new database delimiter
- d) It defines the column separator in a table

3. In a stored procedure, which parameter type allows both input and output?

- a) IN
- b) OUT
- c) INOUT
- d) BOTH

- 4.** What is the primary purpose of a database trigger?
- a) To manually execute SQL statements
 - b) To automatically execute code in response to certain events on a table**
 - c) To create backup copies of data
 - d) To improve query performance
- 5.** Which keyword is used to reference the new row values in an INSERT trigger?
- a) OLD
 - b) NEW**
 - c) CURRENT
 - d) INSERTED
- 6.** What does UNION do in MySQL?
- a) Combines the result sets of two or more SELECT statements, removing duplicates**
 - b) Joins two tables based on a common column
 - c) Creates a union of all table constraints
 - d) Merges two databases into one
- 7.** Which encryption function in MySQL produces a 128-bit hash value?
- a) SHA1()
 - b) MD5()**
 - c) AES_ENCRYPT()
 - d) PASSWORD()

8. What is the main difference between a view and a table?

- a) A view stores data physically, while a table does not
- b) A view is a virtual table based on a query result
- c) A view cannot be updated
- d) There is no difference

9. Which SQL statement is used to grant privileges to a user?

- a) GIVE PRIVILEGES
- b) SET PRIVILEGES
- c) GRANT
- d) ALLOW

10. What does the ON DELETE CASCADE option do in a foreign key constraint?

- a) Prevents deletion of referenced records
- b) Automatically deletes child records when parent record is deleted
- c) Creates a backup before deletion
- d) Requires manual confirmation for deletion

11. Which join type returns all records from both tables?

- a) INNER JOIN
- b) LEFT JOIN
- c) RIGHT JOIN
- d) CROSS JOIN

12. What is SQL injection?

- a) A method to speed up database queries
- b) A security vulnerability where malicious SQL code is inserted through user input**
- c) A technique to join multiple tables
- d) A way to inject data into a database

13. Which of the following is a characteristic of a stored function in MySQL?

- a) It can return multiple values
- b) It must return a single value**
- c) It cannot accept parameters
- d) It cannot be used in SELECT statements

14. What does the GROUP_CONCAT() function do?

- a) Concatenates strings from multiple columns
- b) Concatenates values from multiple rows into a single string**
- c) Groups rows by concatenated values
- d) Creates a new group of records

15. In Discretionary Access Control (DAC), who controls access to database objects?

- a) Only the database administrator
- b) The operating system
- c) The owner of the object
- d) All users have equal control

16. Which trigger timing executes before the triggering statement?

- a) AFTER
- b) BEFORE
- c) INSTEAD OF
- d) DURING

17. What is the purpose of FLUSH PRIVILEGES in MySQL?

- a) To clear all user accounts
- b) To reload grant tables and apply privilege changes immediately
- c) To flush the database cache
- d) To remove all privileges from users

18. In PHP, which method is used to establish a connection to MySQL database?

- a) `new mysql()`
- b) `new mysqli()`
- c) `connect_mysql()`
- d) `mysql_connect()`

19. What does the DETERMINISTIC keyword mean in a stored function?

- a) The function will always return the same result for the same input parameters
- b) The function determines the table structure
- c) The function can modify data
- d) The function is executed automatically

20. Which of the following is NOT a valid flow control structure in MySQL stored procedures?

- a) IF
- b) CASE
- c) WHILE
- d) FOR

Part II: SQL Programming Problems

Problem 1

Consider the `chinook` database with the following relevant tables:

```
1 Artist (ArtistId, Name)
2 Album (AlbumId, Title, ArtistId)
3 Track (TrackId, Name, AlbumId, MediaTypeId, GenreId,
4 Composer, Milliseconds, Bytes, UnitPrice)
```

Part A: Create a stored function called `GetArtistTrackCount` that takes an artist's name as input and returns the total number of tracks by that artist.

Fill in the blanks to complete the function:

```
1 DELIMITER $$

2

3 CREATE FUNCTION GetArtistTrackCount(artist_name VARCHAR(120))
4     RETURNS _____(1)_____
5     DETERMINISTIC
6 BEGIN
7     DECLARE track_count INT;

8
9     SELECT _____(2)_____
10    INTO track_count
11    FROM Artist ar
12    _____(3)_____ Album al ON ar.ArtistId = al.
13        ArtistId
14    _____(4)_____ Track t ON al.AlbumId = t.AlbumId
15    WHERE ar.Name = _____(5)_____;
16
17    RETURN track_count;
18 END $$

19 DELIMITER ;
```

(1) INT

(2) COUNT(*)

(3) JOIN

(4) JOIN

(5) artist_name

Part B: Create a stored procedure called `GetAlbumsByArtist` that accepts an artist name as an IN parameter and returns the total number of albums as an OUT parameter. The procedure should also display all album titles by that artist.

Fill in the blanks to complete the procedure:

```
1 DELIMITER $$  
2  
3 CREATE PROCEDURE GetAlbumsByArtist(  
4     ----- (1) ----- p_artist_name VARCHAR(120),  
5     ----- (2) ----- p_album_count INT  
6 )  
7 BEGIN  
8     -- Select and display all albums by the artist  
9     SELECT al.Title AS AlbumTitle  
10    FROM ----- (3) ----- ar  
11    JOIN ----- (4) ----- al ON ar.ArtistId = al.  
12        ArtistId  
13    WHERE ar.Name = p_artist_name;  
14  
15    -- Get the count of albums  
16    SELECT ----- (5) -----  
17    INTO p_album_count  
18    FROM Artist ar  
19    JOIN Album al ON ar.ArtistId = al.ArtistId  
20    WHERE ar.Name = ----- (6) -----;  
21 END $$  
22  
23 DELIMITER ;
```

(1) IN (2) OUT

(3) _____ (4) _____

(5) COUNT(*) (6) p_artist_name

Problem 2:

Consider a database with the following tables for an employee management system:

```
1 employees (
2     emp_id INT PRIMARY KEY AUTO_INCREMENT ,
3     first_name VARCHAR(50) ,
4     last_name VARCHAR(50) ,
5     email VARCHAR(100) ,
6     salary DECIMAL(10,2) ,
7     hire_date DATE
8 )
9
10 salary_audit (
11     audit_id INT PRIMARY KEY AUTO_INCREMENT ,
12     emp_id INT ,
13     old_salary DECIMAL(10,2) ,
14     new_salary DECIMAL(10,2) ,
15     change_date TIMESTAMP ,
16     change_type VARCHAR(20)
17 )
```

Part A: Create an AFTER UPDATE trigger named audit_salary_changes that automatically logs any salary changes to the salary_audit table. The trigger should record the employee ID, old salary, new salary, current timestamp, and set change_type to 'UPDATE'.

Fill in the blanks to complete the trigger:

```
1 DELIMITER $$  
2  
3 CREATE TRIGGER audit_salary_changes  
4     (1) ----- UPDATE ON employees  
5     FOR EACH ROW  
6 BEGIN  
7     IF (2) ----- <> (3) -----  
8     THEN  
9         INSERT INTO salary_audit (  
10             emp_id,  
11             old_salary,  
12             new_salary,  
13             change_date,  
14             change_type  
15         )  
16         VALUES (  
17             (4) -----,  
18             (5) -----,  
19             NEW.salary,  
20             (6) -----,  
21             'UPDATE'  
22         );  
23     END IF;  
24 END $$  
25 DELIMITER ;
```

(1) AFTER

(2) NEW.salary

(3) OLD.salary

(4) NEW.emp_id / OLD.emp_id

(5) OLD.salary

(6) NOW()

Part B: For security purposes, you need to encrypt employee email addresses in the `employees` table using AES encryption with a key derived from SHA1 hashing.

Fill in the blanks to complete the encryption process:

Step 1: Modify the email column to support encrypted data

```
1 ALTER TABLE employees  
2 MODIFY email -----(1) -----;
```

(1) VARBINARY(255) or BLOB

Step 2: Update all existing email addresses to be encrypted

```
1 UPDATE employees  
2 SET email = ----- (2) ----- (   
3           email,  
4           ----- (3) ----- ('email_key')  
5       );
```

(2) AES_ENCRYPT

(3) SHA1

Step 3: Create a view to display decrypted emails for authorized users

```
1 CREATE VIEW employee_emails AS  
2 SELECT  
3     emp_id,  
4     first_name,  
5     last_name,  
6     ----- (4) ----- (   
7           email,  
8           ----- (5) ----- ('email_key')  
9     ) AS decrypted_email,  
10    salary  
11   FROM employees;
```

(4) AES_DECRYPT

(5) SHA1

Problem 3:

Consider the spotify_2025 database with the following table:

```
1  spotify_data (
2      track_name VARCHAR(255) ,
3      artists_name VARCHAR(255) ,
4      artist_count INT ,
5      released_year INT ,
6      released_month INT ,
7      released_day INT ,
8      in_spotify_playlists INT ,
9      in_spotify_charts INT ,
10     streams INT ,
11     in_apple_playlists INT ,
12     in_apple_charts INT ,
13     bpm INT ,
14     key_ VARCHAR(2) ,
15     mode_ VARCHAR(5) ,
16     danceability_percent INT ,
17     valence_percent INT ,
18     energy_percent INT ,
19     acousticness_percent INT ,
20     liveness_percent INT ,
21     speechiness_percent INT
22 )
```

Part A: Write a query to find the top 5 artists with the highest average danceability across all their tracks. Display the artist name, number of tracks, and average danceability percentage.

Fill in the blanks to complete the query:

```
1 SELECT
2     artists_name,
3         ----- (1) ----- AS track_count,
4         ----- (2) ----- AS avg_danceability
5 FROM spotify_data
6         (3) ----- artists_name
7 ORDER BY avg_danceability ----- (4) -----
8         (5) ----- 5;
```

(1) COUNT(*)

(2) AVG(danceability_percent)

(3) GROUP BY

(4) DESC

(5) LIMIT

Part B: Create a stored procedure called `GetTrackStatsByYear` that accepts a year as input and displays comprehensive statistics for that year.

Fill in the blanks to complete the procedure:

```
1 DELIMITER $$  
2  
3 CREATE PROCEDURE GetTrackStatsByYear(  
4     IN p_year INT  
5 )  
6 BEGIN  
7     -- Display comprehensive statistics for the given year  
8     SELECT  
9         p_year AS year,  
10        _____(1)_____ AS total_tracks,  
11        _____(2)_____ (bpm) AS avg_bpm,  
12        _____(3)_____ (energy_percent) AS avg_energy  
13        ,  
14        _____(4)_____ (streams) AS total_streams  
15     FROM spotify_data  
16     WHERE released_year = _____(5)_____;  
17  
18     -- Find the most common key for the year  
19     SELECT key_ AS most_common_key, _____(6)_____  
20         AS count  
21     FROM spotify_data  
22     WHERE released_year = p_year  
23         _____(7)_____ key_  
24     ORDER BY count DESC  
25         _____(8)_____ 1;  
26 END $$  
27  
28 DELIMITER ;
```

(1) COUNT(*)

(2) AVG

(3) AVG

(4) SUM

(5) p-year

(6) COUNT(*)

(7) GROUP BY

(8) LIMIT

Part III: PHP-MySQL Integration

Problem 4

Consider the following products table in a database:

product_id	product_name	category	price
1	Laptop	Electronics	45000
2	Mouse	Electronics	500
3	Desk	Furniture	8000
4	Chair	Furniture	3500
5	Monitor	Electronics	12000

The following PHP code connects to the database and executes a query:

```
1 <?php
2 $mysqli = new mysqli('localhost', 'root', 'root', 'shop_db');
3
4 $q = SELECT category, COUNT(*) as item_count, AVG(price) as
5     avg_price
6     FROM products
7     GROUP BY category
8     ORDER BY avg_price DESC ;
9
10 $result = $mysqli->query($q);
11 $data = array();
12
13 while($row = $result->fetch_array()) {
14     $data[] = $row;
15 }
16
17 $mysqli->close();
?>
```

Question: What will be stored in the `$data` array after the code executes?
Write the contents of the array. Show your work by explaining how you calculated the values.

Answer:

`$data[0]['category'] = Electronics`

`$data[0]['item_count'] = 3`

`$data[0]['avg_price'] = 19166.67`

`$data[1]['category'] = Furniture`

`$data[1]['item_count'] = 2`

`$data[1]['avg_price'] = 5750`

Show your calculations:

$$\frac{45000 + 500 + 12000}{3} \approx 19166.67$$

$$\frac{8000 + 3500}{2} = 5750$$

Problem 5

Consider the following two tables in a database:
students table:

student_id	name	major
101	Alice	CS
102	Bob	EE
103	Carol	CS
104	David	ME

grades table:

	student_id	course	grade
Alice	101	CSS326	85
	101	CSS324	90
Bob	102	CSS326	78
	103	CSS326	92
Carol	103	CSS324	88
	103	CSS322	95

The following PHP code executes a query:

```

1 <?php
2 $mysqli = new mysqli('localhost', 'user', 'pass', 'uni_db');
3
4 $q = "SELECT s.name, s.major, COUNT(g.course) as course_count
5      ,
6          AVG(g.grade) as avg_grade
7      FROM students s
8      LEFT JOIN grades g ON s.student_id = g.student_id
9      GROUP BY s.student_id
10     HAVING COUNT(g.course) >= 2
11     ORDER BY avg_grade DESC ;
12
13 $result = $mysqli->query($q);
14 $students = array();
15
16 while($row = $result->fetch_array()) {
17     $students[] = array(
18         'name' => $row['name'],
19         'major' => $row['major'],

```

```

19     'courses' => $row['course_count'],
20     'average' => $row['avg_grade']
21   );
22 }
23
24 $mysqli->close();
25 ?>

```

Question: What will be stored in the `$students` array after the code executes? Fill in the table below and explain your reasoning.

Answer:

Index	name	major	courses	average
<code>\$students[0]</code>	Carol	CS	3	91.66
<code>\$students[1]</code>	Alice	CS	2	87.5
<code>\$students[2]</code>	N/A	N/A	N/A	N/A

Note: If there are fewer than 3 results, write “N/A” in the unused rows.

Explanation (show which students qualify for the result and why):

Above!