Here are 100+ MySQL interview questions with their answers, ranging from basic to more intermediate topics:

- **How do you create a new database in MySQL?**
  - **Answer:**

```
CREATE DATABASE database_name;
```

- **How do you create a new table in MySQL?**
  - **Answer:**

```
CREATE TABLE table_name (
    column1 datatype1,
    column2 datatype2,
    ...
);
```

- **How do you insert values into a table?**
  - **Answer:**

```
INSERT INTO table_name (column1, column2, ...)
VALUES (value1, value2, ...);
```

- **How do you retrieve all the columns from a table?**
  - **Answer:**

```
SELECT * FROM table_name;
```

- **How can you retrieve specific columns from a table?**
  - **Answer:**

```
SELECT column1, column2
FROM table_name;
```

- **What is the use of the WHERE clause?**
  - **Answer:** To filter records based on specific conditions.
- **How would you fetch data from a table where the age is greater than 25?**
  - **Answer:**

```
SELECT * FROM table_name WHERE age > 25;
```

- **What are the different types of SQL JOINs?**
  - **Answer:** INNER JOIN, LEFT (or LEFT OUTER) JOIN, RIGHT (or RIGHT OUTER) JOIN, and FULL (or FULL OUTER) JOIN.
- **Write a SQL query to join two tables: `students` and `courses`, assuming each student is enrolled in a course and they share a common column `course_id`.**
  - **Answer:**

```
SELECT * FROM students
INNER JOIN courses
ON students.course_id = courses.course_id;
```

- **What is the difference between the `HAVING` clause and the `WHERE` clause?**
  - **Answer:** `WHERE` filters records before aggregating in `GROUP BY`, whereas `HAVING` filters after aggregation.
- **How would you list the number of students enrolled in each course, but only display courses with more than 5 students?**
  - **Answer:**

```
SELECT course_id, COUNT(student_id) as number_of_students
FROM enrollments
GROUP BY course_id
HAVING number_of_students > 5;
```

- **What is the `LIKE` operator used for?**
  - **Answer:** To search for a specified pattern in a column.
- **Write a SQL query to find all students whose names start with 'A'.**
  - **Answer:**

```
SELECT * FROM students WHERE name LIKE 'A%';
```

- **How would you update a record in a table?**
  - **Answer:**

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE some_column = some_value;
```

- **How can you delete records from a table?**
  - **Answer:**

```
DELETE FROM table_name WHERE condition;
```

- **How do you drop a table?**
  - **Answer:**

```
DROP TABLE table_name;
```

- **What is the purpose of the `ALTER` table command?**
  - **Answer:** To modify an existing table structure, such as adding, deleting, or modifying columns.
- **How would you add a new column `email` to the `students` table?**
  - **Answer:**

```
ALTER TABLE students ADD COLUMN email VARCHAR(255);
```

- **What does the `DISTINCT` keyword do in a SQL query?**
  - **Answer:** It removes duplicate rows from the result set.
- **Write a query to find the total number of distinct courses from the `enrollments` table.**
  - **Answer:**

```
SELECT COUNT(DISTINCT course_id) FROM enrollments;
```

- **What does the `EXISTS` operator do?**
  - **Answer:** It tests for the existence of any record in a subquery.
- **Write a SQL query to find students who have enrolled in a course.**
  - **Answer:**

```
SELECT student_id
FROM students
WHERE EXISTS (SELECT 1 FROM enrollments WHERE students.student_id = enrollments.student_id);
```

- **How can you concatenate columns in MySQL?**
  - **Answer:** Using the `CONCAT()` function.
- **Write a query to get the full name of a student, given `first_name` and `last_name` columns.**
  - **Answer:**

```
SELECT CONCAT(first_name, ' ', last_name) as full_name FROM students;
```

- **How do you find the total number of rows in a table?**
  - **Answer:**

```
SELECT COUNT(*) FROM table_name;
```

- **How can you fetch the first 5 records from a table?**
  - **Answer:**

```
SELECT * FROM table_name LIMIT 5;
```

- **What is the difference between `CHAR` and `VARCHAR` data types?**
  - **Answer:** `CHAR` is fixed-length while `VARCHAR` is variable-length.
- **How can you change the data type of a column?**
  - **Answer:**

```
ALTER TABLE table_name MODIFY column_name NEW_DATA_TYPE;
```

- **Write a SQL query to find the 3rd highest salary from a `salaries` table.**

- **Answer:**

```
SELECT DISTINCT salary
FROM salaries
ORDER BY salary DESC
LIMIT 1 OFFSET 2;
```

- **How do you create a primary key in a table?**
  - **Answer:**

```
ALTER TABLE table_name ADD PRIMARY KEY (column_name);
```

- **What is a foreign key constraint, and why is it used?**
  - **Answer:** A foreign key constraint establishes a link between two tables and ensures that records in one table correspond to records in another. It's used to maintain referential integrity in the database.
- **How can you add a foreign key constraint to an existing table?**
  - **Answer:**

```
ALTER TABLE table_name ADD FOREIGN KEY (column_name) REFERENCES other_table(other_column);
```

- **How can you retrieve the unique values from a column?**
  - **Answer:**

```
SELECT DISTINCT column_name FROM table_name;
```

- **What is the difference between an `INNER JOIN` and a `LEFT JOIN`?**
  - **Answer:** An `INNER JOIN` returns rows when there is a match in both tables, while a `LEFT JOIN` returns all rows from the left table and the matched rows from the right table. If there's no match, the result is `NULL` on the right side.
- **What is normalization, and why is it important?**
  - **Answer:** Normalization is the process of organizing a database to reduce redundancy and ensure data integrity. It divides larger tables into smaller ones and establishes relationships between them using foreign keys.
- **Describe 1NF, 2NF, and 3NF in database normalization.**
  - **Answer:**
    - **1NF (First Normal Form):** Each table has a primary key, and all attributes are atomic (no repeating groups or arrays).
    - **2NF (Second Normal Form):** All non-key attributes are fully functionally dependent on the primary key.
    - **3NF (Third Normal Form):** All attributes are functionally dependent only on the primary key.
  - **What is a subquery, and how is it different from a JOIN?**
  - **Answer:** A subquery is a query nested inside another query. A subquery can return data that will be used in the main query as a condition. A JOIN is used to combine rows from two or more tables based on a related column.
- **Write a query to find employees whose salary is above the average salary.**
  - **Answer:**

```
SELECT employee_name, salary
FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees);
```

- **What is a stored procedure in MySQL?**
  - **Answer:** A stored procedure is a precompiled group of SQL statements stored in the database. It can be invoked as needed.
- **How can you handle errors in stored procedures?**
  - **Answer:** In MySQL, you can use the `DECLARE` statement to define error handlers using `CONTINUE` or `EXIT` handlers.
- **How do you prevent SQL injection in your queries?**
  - **Answer:** Use parameterized queries or prepared statements, avoid constructing queries with string concatenation using external input, and always validate and sanitize user input.
- **What are `TRIGGERS` in MySQL?**
  - **Answer:** Triggers are automatic actions that the database can perform when a specified change occurs (like an `INSERT`, `UPDATE`, or `DELETE` operation).
- **Can you explain the difference between `CHAR_LENGTH` and `LENGTH` functions?**
  - **Answer:** `CHAR_LENGTH` returns the number of characters in a string, while `LENGTH` returns the number of bytes. For single-byte character sets, they return the same value.
- **What is the purpose of the `GROUP_CONCAT` function in MySQL?**
  - **Answer:** `GROUP_CONCAT` returns a concatenated string of aggregated data values for each group of rows in the result set.
- **Write a SQL query to concatenate all names from the `employees` table into a single string, separated by commas.**
  - **Answer:**

```
SELECT GROUP_CONCAT(employee_name) FROM employees;
```

- **How can you create an index in MySQL?**
  - **Answer:**

```
CREATE INDEX index_name ON table_name(column_name);
```

- **What is the difference between a clustered and a non-clustered index?**
  - **Answer:** A clustered index determines the physical order of data in a table. A table can have only one clustered index. Non-clustered indexes, on the other hand, do not determine the physical order and a table can have multiple non-clustered indexes.
- **What are views in MySQL, and why are they used?**
  - **Answer:** A view is a virtual table based on the result-set of an SQL statement. They allow encapsulating complex queries, providing a simplified representation or hiding certain data.
- **What are transactions in MySQL?**
  - **Answer:** Transactions are a sequence of one or more SQL operations executed as a single unit. They ensure data integrity, following the ACID properties (Atomicity, Consistency, Isolation, Durability).
- **How do you start and commit a transaction in MySQL?**
  - **Answer:**

```
START TRANSACTION;
-- SQL operations
COMMIT;
```

- **What is the difference between `UNION` and `UNION ALL`?**
  - **Answer:** `UNION` returns unique records from the combined dataset, while `UNION ALL` returns all records, including duplicates.
- **What are the advantages of using stored procedures?**
  - **Answer:** They provide better performance as they are precompiled, help in modular programming, offer a security mechanism, and reduce network traffic.
- **What is the difference between `DATEDIFF` and `TIMESTAMPDIFF` in MySQL?**

- **Answer:** Both are used to find the difference between two dates, but `TIMESTAMPDIFF` allows for a more specific interval, like month or year, while `DATEDIFF` returns the difference in days.

- **How do you clone a table in MySQL?**
  - **Answer:**

```
CREATE TABLE new_table AS SELECT * FROM existing_table;
```

- **Write a SQL query to rank employees based on their salary in descending order.**
  - **Answer:**

```
SELECT employee_name, salary, RANK() OVER(ORDER BY salary DESC) AS ranking
FROM employees;
```

- **How do you remove duplicate rows in a table?**
  - **Answer:** One common way is to create a new table with the distinct rows and delete the original table:

```
CREATE TABLE new_table AS SELECT DISTINCT * FROM original_table;
DROP TABLE original_table;
RENAME TABLE new_table TO original_table;
```

- **What are the default storage engines in MySQL?**
  - **Answer:** The default storage engine was MyISAM up to MySQL 5.5, but InnoDB became the default from MySQL 5.5 onward.

- **What is a self-join, and why would you use it?**
  - **Answer:** A self-join is a join of a table to

- **What is the purpose of the `SET` data type in MySQL?**
  - **Answer:** The `SET` type is used to store a set of strings. You can store zero or more string values chosen from a list defined at table creation time.

```
CREATE TABLE t1 (colors SET('red', 'blue', 'green'));
INSERT INTO t1 (colors) VALUES ('red,blue');
```

- **How do you implement pagination in MySQL?**
  - **Answer:** Using `LIMIT` and `OFFSET`.

```
SELECT * FROM table_name LIMIT 10 OFFSET 20;  -- Skips the first 20 records and fetches the next 10.
```

- **How can you retrieve the month part from a `DATE` field in MySQL?**
  - **Answer:** Using the `MONTH()` function.

```
SELECT MONTH(date_column) FROM table_name;
```

- **How do you convert a `DATETIME` field into a Unix timestamp?**

- **Answer:** Using the `UNIX_TIMESTAMP()` function.

```
SELECT UNIX_TIMESTAMP(datetime_column) FROM table_name;
```

- **How can you perform a case-sensitive search in a column?**
  - **Answer:** Using the `BINARY` keyword.

```
SELECT * FROM table_name WHERE BINARY column_name = 'Value';
```

- **How can you transpose rows into columns, and vice versa, in a query result?**
  - **Answer:** This process is known as "Pivoting". To convert rows to columns, you use a combination of aggregate functions with `CASE` statements. For the reverse, known as "Unpivoting", you can use `UNION ALL`.

```
-- Pivoting:
SELECT
    SUM(CASE WHEN column = 'value1' THEN 1 ELSE 0 END) AS 'Value1',
    SUM(CASE WHEN column = 'value2' THEN 1 ELSE 0 END) AS 'Value2'
FROM table_name;

-- Unpivoting:
SELECT 'Value1' AS 'Column', Value1 AS 'Value' FROM table_name
UNION ALL
SELECT 'Value2' AS 'Column', Value2 AS 'Value' FROM table_name;
```

- **How can you get a list of all indexes in a database?**
  - **Answer:**

```
SHOW INDEXES FROM table_name IN database_name;
```

- **How can you optimize a MySQL query?**
  - **Answer:** Some methods include using `EXPLAIN` to analyze the query plan, indexing appropriate columns, avoiding the use of wildcard characters at the start of a `LIKE` query, and avoiding the use of `SELECT *`.
- **What is the difference between `MyISAM` and `InnoDB`?**
  - **Answer:** Major differences include:
    - `InnoDB` supports ACID-compliant transactions, whereas `MyISAM` does not.
    - `InnoDB` supports foreign key constraints, while `MyISAM` does not.
    - `MyISAM` typically offers better read performance, while `InnoDB` offers better write performance.
  - **How can you lock a table explicitly?**
  - **Answer:**

```
LOCK TABLES table_name READ|WRITE; --Lock for reading/writing
UNLOCK TABLES; --To release the lock
```

- **How do you get the second highest value from a table column?**
  - **Answer:**

```
SELECT MAX(column_name) FROM table_name WHERE column_name < (SELECT MAX(column_name) FROM table_name);
```

- **What is a correlated subquery?**
  - **Answer:** A correlated subquery is a subquery that references columns from the outer query. It's executed once for each row processed by the outer query.

```
SELECT column_name
FROM table_name t1
WHERE some_value = (SELECT MAX(column_name) FROM table_name t2 WHERE t1.id = t2.id);
```

- **How can you increase the performance of a MySQL database?**
  - **Answer:** Optimize queries using `EXPLAIN`, use indexes wisely, normalize database schema, consider hardware upgrades, and configure database parameters appropriately in `my.cnf` or `my.ini`.
- **How do you backup and restore a MySQL database?**
  - **Answer:**

```
mysqldump -u username -p database_name > backup.sql
```

To restore:

```
mysql -u username -p database_name < backup.sql
```

- **What are the different types of MySQL collations?**
  - **Answer:** Collations specify the rules for string comparison. There are various types like `utf8_general_ci`, `utf8mb4_unicode_ci`, and `latin1_general_ci`.
- **How do you find the total number of rows affected by a query?**
  - **Answer:**

```
SELECT ROW_COUNT();
```

- **Explain the difference between `CHAR` and `VARCHAR` data types.**
  - **Answer:** `CHAR` has a fixed length, while `VARCHAR` has a variable length. For `CHAR`, unused spaces are filled with blank spaces, whereas `VARCHAR` only uses the required storage plus one or two extra bytes for the length.
- **How can you change the data type of a column in MySQL?**
  - **Answer:**

```
ALTER TABLE table_name MODIFY column_name NEW_DATA_TYPE;
```

- **How can you measure the size of a MySQL database?**
  - **Answer:**

```
SELECT table_schema AS "Database", ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) AS "Size (MB)"
FROM information_schema.TABLES
GROUP BY table_schema;
```

- **How can you delete all records from a table without deleting the table?**
  - **Answer:**

```
TRUNCATE TABLE table_name;
```

- **How can you prevent a query from displaying duplicate rows?**
  - **Answer:**

```
SELECT DISTINCT column_name FROM table_name;
```

- **How do you combine results from multiple SQL queries and return a single table?**
  - **Answer:** You can use the `UNION` or `UNION ALL` operator, depending on whether or not you want duplicate records.
- **How can you convert a string to upper-case in MySQL?**
  - **Answer:**

```
SELECT UPPER(column_name) FROM table_name;
```

- **How can you remove leading and trailing whitespace from a string in MySQL?**
  - **Answer:**

```
SELECT TRIM(column_name) FROM table_name;
```

- **Explain the purpose of `information_schema` in MySQL.**
  - **Answer:** `information_schema` is a meta-database that provides detailed information about all other databases, tables, columns, indexes, constraints, etc. present in the MySQL server.
- **How can you ensure that a field value is unique across the table, other than using the `PRIMARY KEY` constraint?**
  - **Answer:** Use the `UNIQUE` constraint on the desired column.

```
ALTER TABLE table_name ADD UNIQUE (column_name);
```

- **How can you count the total number of tables in a database?**
  - **Answer:**

```
SELECT COUNT(*) FROM information_schema.tables WHERE table_schema = 'your_database_name';
```

- **How can you find all the tables that have a specific column name in a database?**
  - **Answer:**

```
SELECT table_name
FROM information_schema.columns
WHERE column_name = 'desired_column' AND table_schema = 'your_database_name';
```

- **How can you replace a specific string in a field?**
  - **Answer:**

```
UPDATE table_name SET column_name = REPLACE(column_name, 'old_string', 'new_string');
```

- **What is the difference between `NOW()` and `CURDATE()` functions in MySQL?**
  - **Answer:** `NOW()` returns the current date and time, while `CURDATE()` returns only the current date.

These questions cover a range of advanced topics and should help in assessing the depth of knowledge of individuals familiar with MySQL.

89. Explain the `WITH` clause and provide an example.

- **Answer:** The `WITH` clause, also known as Common Table Expressions (CTE), provides a temporary result set that you can reference within a `SELECT`, `INSERT`, `UPDATE`, or `DELETE` statement. It's useful for breaking down complex queries.

```
WITH CTE_Name AS (
    SELECT column1, column2
    FROM table_name
    WHERE condition
)
SELECT * FROM CTE_Name;
```

90. What is a self-join and why would you use it?

- **Answer:** A self-join is a join where a table is joined with itself. It's useful for finding relationships within the same table.

```
SELECT A.column_name, B.column_name
FROM table_name A, table_name B
WHERE A.column_name = B.column_name;
```

91. What are the different types of subqueries? Explain with examples.

- **Answer:** There are three types:
- Scalar subquery: Returns a single value.

```
SELECT column_name
FROM table_name
WHERE another_column = (SELECT MAX(column_name) FROM table_name);
```

- Row subquery: Returns a single row.

```
SELECT column1, column2
FROM table_name
WHERE (column1, column2) = (SELECT column1, column2 FROM another_table WHERE condition);
```

- Table subquery: Returns a table.

```
SELECT column_name
FROM (
  SELECT column_name FROM table_name WHERE condition
) AS subquery_name;
```

92. How can you update data in one table based on data in another table?

- **Answer:**

```
UPDATE table1
SET table1.column_name = table2.column_name
FROM table2
WHERE table1.another_column = table2.another_column;
```

93. How can you retrieve a random row from a table?

- **Answer:**

```
SELECT column_name FROM table_name ORDER BY RAND() LIMIT 1;
```

94. What's the difference between `INNER JOIN` and `OUTER JOIN`?

- **Answer:** `INNER JOIN` returns rows when there's a match in both tables. `OUTER JOIN` returns all rows from one table and the matching rows from the other table, filling with NULL if no match is found.

95. How can you clone a table, including both data and schema?

- **Answer:**

```
CREATE TABLE new_table AS SELECT * FROM original_table;
```

96. How do you insert multiple rows in a single SQL query?

- **Answer:**

```
INSERT INTO table_name (column1, column2)
VALUES (value1a, value2a),
       (value1b, value2b),
       ...;
```

97. Explain partitions in MySQL. How do you create them?

- **Answer:** Partitioning divides a table into smaller, more manageable pieces, yet still being treated as a single table. It can improve performance and assist in organizing large datasets.

```
CREATE TABLE table_name (
   column_name1 INT,
   column_name2 DATE
)
PARTITION BY RANGE(YEAR(column_name2)) (
   PARTITION p0 VALUES LESS THAN (1991),
   PARTITION p1 VALUES LESS THAN (1995),
   PARTITION p2 VALUES LESS THAN (1999)
);
```

98. What is the `GROUP_CONCAT` function and provide an example.

- **Answer:** It's used to concatenate values from multiple rows into a single string.

```
SELECT group_column, GROUP_CONCAT(value_column)
FROM table_name
GROUP BY group_column;
```

99. How can you prevent SQL injection in your queries?

- **Answer:** Using parameterized queries or prepared statements. In PHP, for instance, you'd use PDO or MySQLi to bind parameters.

100. How do you show the current SQL mode, and how can you change it?

- **Answer:**

```
SELECT @@sql_mode;  -- To show
SET sql_mode = 'modes';  -- To change
```

101. What is a transaction and how would you use it in MySQL?

- **Answer:** Transactions group a set of tasks into a single execution unit. If one task fails, all fail. Useful for maintaining data integrity.

```
START TRANSACTION;
INSERT INTO table_name1 ...;
INSERT INTO table_name2 ...;
COMMIT;  -- Or ROLLBACK;
```

102. What are the differences between `VARCHAR` and `TEXT` data types?

- **Answer:** While both are used to store strings, `VARCHAR` can store up to 65,535 characters and you can specify its max length, while `TEXT` can store up to 65,535 characters without specifying max length. `VARCHAR` can have a default value, but `TEXT` cannot.

103. How do you find and fix broken foreign key constraints?

- **Answer:** Identify them using a `LEFT JOIN` to find orphaned records, and either delete these records or update them to restore referential integrity.

104. How do you use `FULLTEXT` indexing in MySQL?

- **Answer:** `FULLTEXT` indexes are used for full-text searches. You can create one with:

```
CREATE FULLTEXT INDEX index_name ON table_name(column_name);
```

Then you'd search with:

```
SELECT * FROM table_name WHERE MATCH(column_name) AGAINST('search term');
```

105. How can you check for index fragmentation on a table and defragment it?

- **Answer:** You can check fragmentation using `SHOW TABLE STATUS LIKE 'table_name';` and optimize (defragment) using `OPTIMIZE TABLE table_name;`.

106. How can you convert character sets in columns?

- **Answer:**

```
ALTER TABLE table_name MODIFY column_name COLUMN_TYPE CHARACTER SET charset_name;
```

107. How do you schedule a recurring SQL script execution in MySQL?

- **Answer:** Using MySQL's Event Scheduler. First, ensure the scheduler is on with `SHOW VARIABLES LIKE 'event_scheduler';`, then create your scheduled event.

108. What are MySQL stored procedures and how do you use them?

- **Answer:** Stored procedures are SQL codes saved in the database to be reused. Created using `CREATE PROCEDURE`, and called via `CALL procedure_name()`.

109. How would you monitor the performance of your MySQL database in real-time?

- **Answer:** Tools like `SHOW PROCESSLIST`, Performance Schema, MySQL Enterprise Monitor, and third-party tools like Percona Monitoring and Management.

110. How can you run SQL script from the command line without entering the MySQL console?

- **Answer:** Use:

```
mysql -u username -p database_name < script.sql
```

111. What is the `EXPLAIN` keyword in MySQL?

- **Answer:** `EXPLAIN` provides a query execution plan, showing how MySQL will execute the query, which can be vital for optimization.

112. How do you enforce a column to not accept NULL values?

- **Answer:** By adding the `NOT NULL` constraint during table creation or modification.

113. How do you rename a database in MySQL?

- **Answer:** MySQL does not have a straightforward command to rename a database. Instead, one common approach is to dump the database, create a new one with the desired name, and then restore the dumped database into the new one.

114. How can you reset the auto-increment value of a column?

- **Answer:**

```
ALTER TABLE table_name AUTO_INCREMENT = value;
```

115. How can you handle time zones in MySQL?

- **Answer:** MySQL provides the `CONVERT_TZ()` function to convert datetime values across time zones. Additionally, `SET time_zone = timezone;` sets the time zone for the current session.

116. How do you retrieve only a specified number of characters from a string column?

- **Answer:**

```
SELECT LEFT(column_name, number_of_chars) FROM table_name;
```

117. What are views in MySQL and why are they used?

- **Answer:** Views are virtual tables based on the result set of an SQL statement. They encapsulate the SQL statement and present data in a simplified manner, ensuring data abstraction, protection, and to represent a subset of the data.

118. How do you find the second highest value in a column?

- **Answer:**

```
SELECT MAX(column_name)
FROM table_name
WHERE column_name NOT IN (SELECT MAX(column_name) FROM table_name);
```

These questions should serve well for interviews at product-based companies that expect a deep understanding of MySQL.
