MYSQL

Basic Level (25 Questions)

1. **What is MySQL?**
    - **Answer**: MySQL is an open-source relational database management system (RDBMS) based on SQL (Structured Query Language). It is used to store and manage data in a tabular format.
2. **What is the difference between `CHAR` and `VARCHAR`?**
    - **Answer**: `CHAR` is a fixed-length string type, while `VARCHAR` is a variable-length string type. `CHAR` uses more space if the data is shorter than the specified length.
3. **What is a primary key in MySQL?**
    - **Answer**: A primary key is a unique identifier for each record in a table. It ensures that no two rows have the same value for the primary key column(s).
4. **What is a foreign key?**
    - **Answer**: A foreign key is a field in one table that uniquely identifies a row of another table. It is used to establish a relationship between two tables.
5. **What are the types of indexes in MySQL?**
    - **Answer**: The common types of indexes in MySQL are primary, unique, full-text, and regular indexes.
6. **How do you create a database in MySQL?**

Answer:

```
CREATE DATABASE db_name;
```

    -
7. **What is the `AUTO_INCREMENT` keyword in MySQL?**
    - **Answer**: `AUTO_INCREMENT` is used to automatically generate a unique value for a primary key field in a table whenever a new record is inserted.

8. **What are the different types of joins in MySQL?**
    - **Answer**: The types of joins in MySQL are:
        - `INNER JOIN`
        - `LEFT JOIN`
        - `RIGHT JOIN`
        - `FULL JOIN`
9. **What is a `NULL` value in MySQL?**
    - **Answer**: A `NULL` value represents the absence of a value or an unknown value in a table field.
10. **How can you retrieve all records from a table?**

Answer:

```
SELECT * FROM table_name;
```


11. **What is the purpose of the `GROUP BY` clause?**
    - **Answer**: `GROUP BY` is used to group rows that have the same values into summary rows, like calculating the sum or average.
12. **What is the difference between `DELETE` and `TRUNCATE`?**
    - **Answer**: `DELETE` removes records one by one and can be rolled back, while `TRUNCATE` removes all records at once and cannot be rolled back.
13. **What is a `LIMIT` clause?**
    - **Answer**: The `LIMIT` clause is used to specify the number of records to return in a query.
14. **What is the `DISTINCT` keyword used for?**
    - **Answer**: The `DISTINCT` keyword is used to remove duplicate values from the result set.

15. **How do you modify a column in a MySQL table?**

Answer:

```
ALTER TABLE table_name MODIFY column_name datatype;
```

16. **What is the purpose of the `ORDER BY` clause?**
    - **Answer**: `ORDER BY` is used to sort the result set in ascending or descending order based on one or more columns.
17. **What is a subquery in MySQL?**
    - **Answer**: A subquery is a query nested inside another query, usually in a `WHERE`, `HAVING`, or `FROM` clause.
18. **What is the difference between `INNER JOIN` and `LEFT JOIN`?**
    - **Answer**: `INNER JOIN` returns only matching records between two tables, while `LEFT JOIN` returns all records from the left table and matching records from the right table.
19. **How can you add a new column to an existing table?**

Answer:

```
ALTER TABLE table_name ADD column_name datatype;
```

20. **How do you find the current database in MySQL?**

Answer:

```
SELECT DATABASE();
```

21. **What is a `WHERE` clause used for?**
    - **Answer**: The `WHERE` clause is used to filter records based on specified conditions.
22. **What is the `COUNT()` function in MySQL?**

- **Answer**: The `COUNT()` function returns the number of rows that match a specified condition.

23. **How can you get the current date in MySQL?**

Answer:

```
SELECT CURDATE();
```

-

24. **What is a view in MySQL?**

    - **Answer**: A view is a virtual table created by a query, which can be used like a regular table in SELECT statements.

25. **How do you rename a table in MySQL?**

Answer:

```
RENAME TABLE old_table_name TO new_table_name;
```

-

Intermediate Level (25 Questions)

1. **What is the difference between `JOIN` and `UNION`?**

    - **Answer**: `JOIN` is used to combine rows from two or more tables based on a related column, while `UNION` combines the result sets of two or more queries, removing duplicates.

2. **What is a stored procedure in MySQL?**

    - **Answer**: A stored procedure is a set of SQL statements that can be executed as a single unit to perform a specific task in a database.

3. **What is the difference between `TRUNCATE` and `DROP`?**

    - **Answer**: `TRUNCATE` removes all records from a table but keeps the table structure intact, while `DROP` deletes the entire table, including its structure.

4. **What is the purpose of the `HAVING` clause?**

- **Answer**: `HAVING` is used to filter groups of records after an aggregation (like `COUNT()`, `SUM()`) is applied.

5. **How can you prevent NULL values from being inserted into a column?**

Answer: By specifying `NOT NULL` when creating or altering the column:

```
CREATE TABLE table_name (column_name datatype NOT NULL);
```

-

6. **What is the `EXPLAIN` command used for?**

   - **Answer**: `EXPLAIN` provides information about how MySQL executes a query, helping to optimize query performance.

7. **What is the difference between `IN` and `EXISTS`?**

   - **Answer**: `IN` checks if a value is within a list of values, while `EXISTS` checks if a subquery returns any rows.

8. **How can you create an index on multiple columns?**

Answer:

```
CREATE INDEX index_name ON table_name (column1, column2);
```

-

9. **What is a transaction in MySQL?**

   - **Answer**: A transaction is a sequence of SQL operations that are executed as a single unit of work. It ensures that the database is consistent after all operations.

10. **What is the purpose of the `COMMIT` command?**

    - **Answer**: `COMMIT` is used to save all changes made during the current transaction.

11. **What is the purpose of the `ROLLBACK` command?**

    - **Answer**: `ROLLBACK` is used to undo the changes made during the current transaction.

12. **How do you check the status of a table in MySQL?**

Answer:

```
SHOW TABLE STATUS LIKE 'table_name';
```

13. **What is normalization in a database?**
    - **Answer**: Normalization is the process of organizing the attributes of a database to minimize redundancy and dependency.
14. **What is denormalization in a database?**
    - **Answer**: Denormalization is the process of deliberately introducing redundancy into a database to improve query performance.
15. **How can you optimize a query in MySQL?**
    - **Answer**: Some optimization techniques include indexing, avoiding `SELECT *`, using `EXPLAIN` to analyze queries, and optimizing joins.
16. **What is the `UNION ALL` operator?**
    - **Answer**: `UNION ALL` combines the result sets of two or more queries and includes duplicate rows.
17. **What is a trigger in MySQL?**
    - **Answer**: A trigger is a set of SQL statements that automatically executes when an event (INSERT, UPDATE, DELETE) occurs on a table.
18. **What is the difference between `TRUNCATE` and `DELETE` regarding performance?**
    - **Answer**: `TRUNCATE` is faster than `DELETE` because it doesn't log individual row deletions and doesn't trigger any triggers.
19. **How do you check for the structure of a table?**

Answer:

```
DESCRIBE table_name;
```

20. **What is a self-join?**

- **Answer**: A self-join is a join where a table is joined with itself to retrieve related rows.

21. **How do you handle duplicates in MySQL?**
    - **Answer**: By using the `DISTINCT` keyword or the `GROUP BY` clause.
22. **What is the `BETWEEN` operator in SQL?**
    - **Answer**: The `BETWEEN` operator is used to filter the result set within a certain range, inclusive of the endpoints.
23. **What is the `DATE_FORMAT()` function in MySQL?**
    - **Answer**: `DATE_FORMAT()` is used to format a date or time value in a specific format.
24. **What is a composite key in MySQL?**
    - **Answer**: A composite key is a primary key that consists of two or more columns in a table.
25. **How can you retrieve the last inserted ID in MySQL?**

Answer:

```
SELECT LAST_INSERT_ID();
```

-

Advanced Level (25 Questions)

1. **What is a partition in MySQL?**
    - **Answer**: Partitioning divides a large table into smaller, more manageable pieces, but each partition is still treated as a single table.
2. **What is replication in MySQL?**
    - **Answer**: Replication allows data from one MySQL server (master) to be copied to other MySQL servers (slaves) to improve performance, redundancy, and fault tolerance.

3. **What are the different types of MySQL replication?**
    - **Answer**: The types are:
        - Statement-based replication
        - Row-based replication
        - Mixed replication
4. **What is MySQL clustering?**
    - **Answer**: MySQL Cluster is a high-availability, high-performance version of MySQL designed for distributed databases.
5. **What is the `SHOW PROCESSLIST` command used for?**
    - **Answer**: `SHOW PROCESSLIST` shows information about the threads currently being executed in MySQL.
6. **What is the purpose of the `FLUSH` command in MySQL?**
    - **Answer**: `FLUSH` is used to clear or refresh various MySQL server caches, logs, or buffers.
7. **What is a deadlock in MySQL?**
    - **Answer**: A deadlock occurs when two or more transactions are blocked, each waiting for the other to release a lock on resources.
8. **What is the `WITH ROLLUP` operator used for?**
    - **Answer**: `WITH ROLLUP` is used to generate summary reports with subtotals and grand totals for a `GROUP BY` query.
9. **What is the difference between `INNODB` and `MYISAM` storage engines?**
    - **Answer**: `INNODB` supports transactions, foreign keys, and row-level locking, while `MYISAM` is simpler and faster but does not support transactions or foreign keys.
10. **What is a full-text index in MySQL?**
    - **Answer**: A full-text index is used to perform full-text searches on text columns, making it faster to search for specific words in large texts.

11. **How do you backup and restore a MySQL database?**
    - **Answer**: Use the `mysqldump` utility for backups and the `mysql` command for restores:
        - Backup: `mysqldump -u user -p database_name > backup.sql`
        - Restore: `mysql -u user -p database_name < backup.sql`
12. **How do you handle large data loads in MySQL?**
    - **Answer**: Use techniques like batch inserts, `LOAD DATA INFILE`, disabling indexes during data load, and increasing buffer sizes.
13. **What is query optimization in MySQL?**
    - **Answer**: Query optimization involves improving the efficiency of a query by analyzing execution plans and using techniques like indexing, query rewriting, and limiting the result set.
14. **How does MySQL handle transactions?**
    - **Answer**: MySQL uses a transaction log (redo log) to manage transactions, ensuring that changes are permanent when committed and can be rolled back in case of an error.
15. **What is the purpose of the `EXPLAIN` command in MySQL?**
    - **Answer**: `EXPLAIN` shows the execution plan of a query, providing insights on indexes, table scans, and join methods used.
16. **What is the `OPTIMIZE TABLE` command used for?**
    - **Answer**: `OPTIMIZE TABLE` reclaims unused space in a table and improves performance by defragmenting the table.
17. **What are user-defined variables in MySQL?**
    - **Answer**: User-defined variables allow you to store values temporarily during the execution of a query or session. They are set using `SET` or `SELECT`.
18. **What is the difference between `LOCK TABLES` and `FLUSH TABLES`?**

- **Answer**: `LOCK TABLES` locks the specified tables, preventing others from modifying them, while `FLUSH TABLES` flushes or closes the open tables.

19. **How do you handle schema migrations in MySQL?**

    - **Answer**: Schema migrations are typically managed using migration tools like Liquibase or Flyway, or manual SQL scripts to modify table structures over time.

20. **What is the use of `AUTO_COMMIT` in MySQL?**

    - **Answer**: `AUTO_COMMIT` controls whether each individual statement is committed automatically after execution or if a transaction is explicitly committed.

21. **What is the `TRIGGERS` feature in MySQL?**

    - **Answer**: Triggers automatically execute specified actions when certain events (INSERT, UPDATE, DELETE) occur on a table.

22. **What are the benefits of using the `InnoDB` storage engine?**

    - **Answer**: `InnoDB` provides features like ACID compliance, foreign keys, row-level locking, and crash recovery, making it suitable for transactional applications.

23. **What is the role of `FOREIGN KEY` in MySQL?**

    - **Answer**: A `FOREIGN KEY` maintains referential integrity by linking columns in two tables, ensuring consistency between related data.

24. **What is the purpose of the `GROUP_CONCAT` function?**

    - **Answer**: `GROUP_CONCAT` is used to concatenate multiple rows into a single string, usually for grouping data in aggregation queries.

25. **What is `ROW_FORMAT` in MySQL?**

    - **Answer**: `ROW_FORMAT` defines how MySQL stores the data in a table, with options like `COMPACT`, `REDUNDANT`, and `DYNAMIC` for different storage optimizations.

Technical Level (25 Questions)

1. **Explain the concept of normalization with examples.**
    - **Answer**: Normalization is the process of organizing the attributes of a database to reduce redundancy and improve data integrity. It involves dividing a large table into smaller, related tables and using foreign keys to link them. For example, in a sales database:
        - A normalized structure might involve a `Customers` table and an `Orders` table, where the `CustomerID` in the `Orders` table links to the `Customers` table.
        - First Normal Form (1NF): Ensures that all columns contain atomic values.
        - Second Normal Form (2NF): Ensures that every non-key column is fully dependent on the primary key.
        - Third Normal Form (3NF): Ensures that non-key columns are not transitively dependent on the primary key.
2. **What is a `B-tree` index and how does it differ from a `hash` index in MySQL?**
    - **Answer**:
        - **B-tree index**: A balanced tree structure where the data is stored in sorted order. It supports range queries (e.g., `<`, `>`, `BETWEEN`), and is the default index type in MySQL.
        - **Hash index**: A direct lookup using a hash table. It is used in `MEMORY` storage engine and is fast for equality checks (`=`). However, it does not support range queries.
3. **How does MySQL handle concurrency control?**
    - **Answer**: MySQL handles concurrency control using **locking mechanisms** and **isolation levels**. It uses:
        - **Row-level locking**: InnoDB uses this to allow multiple transactions to modify different rows in the same table simultaneously.
        - **Table-level locking**: MyISAM uses this, locking the entire table for modifications.
        - **Isolation levels**: MySQL supports four isolation levels:

        - `READ UNCOMMITTED`
        - `READ COMMITTED`
        - `REPEATABLE READ` (default)
        - `SERIALIZABLE`

4. **What is the difference between a `stable` and `unstable` sort in MySQL?**
    - **Answer**: A **stable sort** preserves the order of records with equal values in the sorted column(s). In contrast, an **unstable sort** may not maintain the order of equal values after sorting.

5. **How does the `MySQL Query Optimizer` work?**
    - **Answer**: The MySQL Query Optimizer analyzes the SQL query and selects the most efficient execution plan. It takes into account factors like available indexes, table statistics, and join types. It chooses the plan based on minimizing resource usage and query execution time.

6. **What are the performance tuning techniques for MySQL?**
    - **Answer**: Some performance tuning techniques include:
        - **Index optimization**: Ensure appropriate indexes are created for frequent query columns.
        - **Query optimization**: Use `EXPLAIN` to analyze query performance and avoid `SELECT *`.
        - **Database normalization**: Minimize redundancy to reduce the size and complexity of queries.
        - **Cache optimization**: Adjust MySQL cache sizes (`key_buffer_size`, `innodb_buffer_pool_size`).
        - **Partitioning**: Partition large tables for better performance.

7. **What is the use of `MySQL Enterprise Monitor`?**
    - **Answer**: MySQL Enterprise Monitor is a tool used to monitor and optimize the performance of MySQL servers. It helps in tracking server health, query performance, and system resource usage, providing insights and recommendations for performance improvements.

8. **What are `ACID` properties in MySQL transactions?**
    - **Answer**: ACID stands for:
        - **Atomicity**: Ensures all operations in a transaction are either fully completed or fully rolled back.
        - **Consistency**: Ensures the database moves from one valid state to another after a transaction.
        - **Isolation**: Ensures that the operations of a transaction are isolated from others, preventing dirty reads or non-repeatable reads.
        - **Durability**: Ensures that once a transaction is committed, it will not be lost, even in case of a crash.
9. **What is the significance of `binlog` in MySQL replication?**
    - **Answer**: The binary log (`binlog`) records all changes to the database (e.g., inserts, updates, deletes). In MySQL replication, the master server writes changes to the `binlog`, and the slave servers read and apply these changes to maintain synchronization with the master.
10. **How does MySQL handle isolation levels in transactions?**
    - **Answer**: MySQL supports four isolation levels:
        - **READ UNCOMMITTED**: Transactions can read uncommitted changes from other transactions (dirty reads allowed).
        - **READ COMMITTED**: Transactions can only read committed changes (no dirty reads).
        - **REPEATABLE READ**: Transactions see a consistent view of the data throughout the transaction (default level).
        - **SERIALIZABLE**: The highest isolation level, ensuring full isolation but potentially lower concurrency.
11. **What is the difference between `SHARED` and `EXCLUSIVE` locks?**
    - **Answer**:
        - **SHARED lock**: Multiple transactions can hold a shared lock on the same resource (e.g., reading data without modification).
        - **EXCLUSIVE lock**: Only one transaction can hold an exclusive lock on a resource (e.g., writing data), preventing others from accessing the

resource.

12. **How can you optimize a large `JOIN` query in MySQL?**
    - **Answer**: To optimize large `JOIN` queries:
        - Use indexes on columns used in `JOIN` conditions.
        - Ensure the query only retrieves necessary columns (avoid `SELECT *`).
        - Use `EXPLAIN` to analyze the execution plan.
        - Consider breaking large queries into smaller subqueries.
        - Use proper `JOIN` types (e.g., `INNER JOIN` vs. `LEFT JOIN`).
13. **What is the difference between `HAVING` and `WHERE` clauses in MySQL?**
    - **Answer**:
        - **WHERE** filters rows before any aggregation (applies before `GROUP BY`).
        - **HAVING** filters rows after the aggregation (applies after `GROUP BY`).
14. **Explain the role of `Foreign Key Constraints` and how to handle violations in MySQL.**
    - **Answer**: Foreign key constraints ensure referential integrity between tables. If a record in a child table references a non-existent record in the parent table, a foreign key violation occurs. To handle violations, you can specify actions like `ON DELETE CASCADE`, `ON UPDATE RESTRICT`, or `ON DELETE SET NULL`.
15. **What is the difference between `LOAD DATA INFILE` and `INSERT` for bulk data loading?**
    - **Answer**
        - **LOAD DATA INFILE**: A highly efficient way to load large datasets from a file into a table.
        - **INSERT**: Typically slower for bulk inserts because it processes each row individually.

16. **What are `User-Defined Functions` (UDFs) in MySQL?**

    - **Answer**: UDFs are custom functions written in C or C++ that extend the functionality of MySQL. They allow users to create their own SQL functions to be used in queries.

17. **How do you analyze a query's performance using `EXPLAIN` output?**

    - **Answer**: The `EXPLAIN` statement provides a query execution plan, showing how MySQL executes a query, including:
        - Which indexes are used.
        - How tables are joined.
        - The order of table access.
        - The number of rows examined at each step.
    - Use this information to identify bottlenecks and optimize the query.

18. **What is a `deadlock victim` in MySQL, and how do you handle it?**

    - **Answer**: A deadlock occurs when two transactions are blocked, each waiting for the other to release a lock. MySQL automatically selects one of the transactions as the "deadlock victim" and rolls it back. Handling it involves ensuring that transactions acquire locks in a consistent order and using smaller transactions to reduce lock contention.

19. **How does MySQL's `InnoDB` handle locking and concurrency?**

    - **Answer**: InnoDB uses **row-level locking** for high concurrency, allowing multiple transactions to update different rows of the same table simultaneously. It uses a **locking mechanism** to prevent conflicts, including **read locks** and **write locks**.

20. **What is the `sql_mode` in MySQL, and how does it affect query execution?**

    - **Answer**: `sql_mode` controls the behavior of MySQL, including data validation, error reporting, and handling of edge cases. Examples of modes include `STRICT_TRANS_TABLES`, `NO_ZERO_DATE`, and `ANSI_QUOTES`. It can be configured to make MySQL more compliant with standards or to modify how errors are handled.

21. **What is the purpose of the `CROSS JOIN` in MySQL?**

- **Answer**: A `CROSS JOIN` produces a Cartesian product of two tables, returning all combinations of rows from both tables. It is often used in scenarios where all possible combinations of records are needed.

22. **How do you handle database security in MySQL?**

- **Answer**:
  - Use **strong passwords** for MySQL user accounts.
  - Grant the minimum necessary privileges using **GRANT** and **REVOKE** statements.
  - Encrypt sensitive data using SSL or AES functions.
  - Keep MySQL up to date with the latest security patches.

23. **What are some best practices for database design in MySQL?**

- **Answer**: Best practices include:
  - Normalize the database to reduce redundancy.
  - Use appropriate data types for fields.
  - Create indexes on columns frequently used in queries.
  - Ensure proper use of foreign keys to maintain referential integrity.
  - Regularly backup the database.

24. **What is the `SELECT FOR UPDATE` statement used for in MySQL?**

- **Answer**: `SELECT FOR UPDATE` is used to lock the selected rows for update in a transaction, preventing other transactions from modifying those rows until the current transaction is committed or rolled back.

25. **What is the role of the `Query Cache` in MySQL, and how can you configure it?**

- **Answer**: The **Query Cache** stores the results of SELECT queries to avoid recomputing them. It can be configured using variables like `query_cache_size`, `query_cache_type`, and `query_cache_limit`. It is disabled by default in MySQL 5.7 and later versions.

IDM TECHPARK

GUIDE`S FOR PERFECT CAREER PATHWAY