MySQL SQL Engine: Behind the Scenes

1. Using Statement (Dynamic SQL)

Example:

```
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("SELECT * FROM users WHERE id = 101");
```

Behind the Scenes Steps:

1. Full query sent to MySQL

2. Parsing (Tokenization)

3. Validation (table, column check)

4. Query Optimization (index, plan)

5. Execution Plan is generated every time

6. Query is executed and results returned

2. Using PreparedStatement

Example:

```
PreparedStatement pst = conn.prepareStatement("SELECT * FROM users WHERE id = ?");
pst.setInt(1, 101);
ResultSet rs = pst.executeQuery();
```

Behind the Scenes Steps:

First Time Only:

1. Parse SQL with '?'

2. Validate syntax

3. Optimize and store execution plan

Every Time:

4. Bind parameters

5. Use cached execution plan

6. Execute and return results

MySQL SQL Engine: Behind the Scenes

3. Statement vs PreparedStatement Comparison

Statement:

- Full SQL is parsed every time
- Slower for repeated queries
- Vulnerable to SQL Injection

PreparedStatement:

- SQL parsed once with placeholders
- Execution plan reused
- Secure and efficient

4. Real-World Analogy

Statement: Ordering a fully custom pizza each time.

PreparedStatement: Using a pre-defined recipe and changing just the toppings.