1. ***Unable to acquire JDBC Connection [HikariPool-1 - Connection is not available, request timed out after \*\*\*\*\*ms (total=2, active=2, idle=0, waiting=1)] [n/a]***

The error message you're seeing:

Unable to acquire JDBC Connection [HikariPool-1 - Connection is not available, request timed out after \*\*\*\*\*ms (total=2, active=2, idle=0, waiting=1)]

means that **HikariCP** (your JDBC connection pool) couldn't provide a database connection within the configured timeout.

**🔍 Root Cause:**

* **Connection Pool Exhaustion**: All 2 available connections (total=2) are in use (active=2), none are idle (idle=0), and one thread is waiting for a connection (waiting=1).
* **Timeout**: The thread waited for the configured time (\*\*\*\*\*ms) and gave up because no connection became available.

**🔧 Common Solutions:**

**1. Increase Maximum Pool Size**

If your application legitimately needs more concurrent connections:

spring.datasource.hikari.maximum-pool-size=10

Or in Java:

HikariDataSource ds = new HikariDataSource();

ds.setMaximumPoolSize(10);

**2. Investigate Connection Leaks**

Connections might not be getting closed properly:

* Ensure connection.close() is always called (use try-with-resources in Java).
* Enable leak detection to log where connections are being held too long:

spring.datasource.hikari.leak-detection-threshold=30000 # 30 seconds

**3. Optimize Query/Transaction Time**

Long-running queries or transactions can tie up connections:

* Profile SQL queries (use APM or database logs).
* Reduce transaction scopes and avoid unnecessary DB calls.

**4. Add Monitoring & Logging**

Use Hikari’s pool metrics or logs to understand usage patterns:

spring.datasource.hikari.pool-name=HikariPool-1

logging.level.com.zaxxer.hikari=DEBUG

**5. Check Database Connection Limits**

The DB itself might limit connections. Verify:

* Max connections on the database server.
* No firewall/DB issues preventing new connections.

**🧪 Temporary Workaround**

If you're running a test or local development, increase pool size for now to avoid blocking:

spring.datasource.hikari.maximum-pool-size=10

spring.datasource.hikari.connection-timeout=30000

If you share your config (Spring Boot YAML/properties, connection logic, or full stack trace), I can help fine-tune the fix.