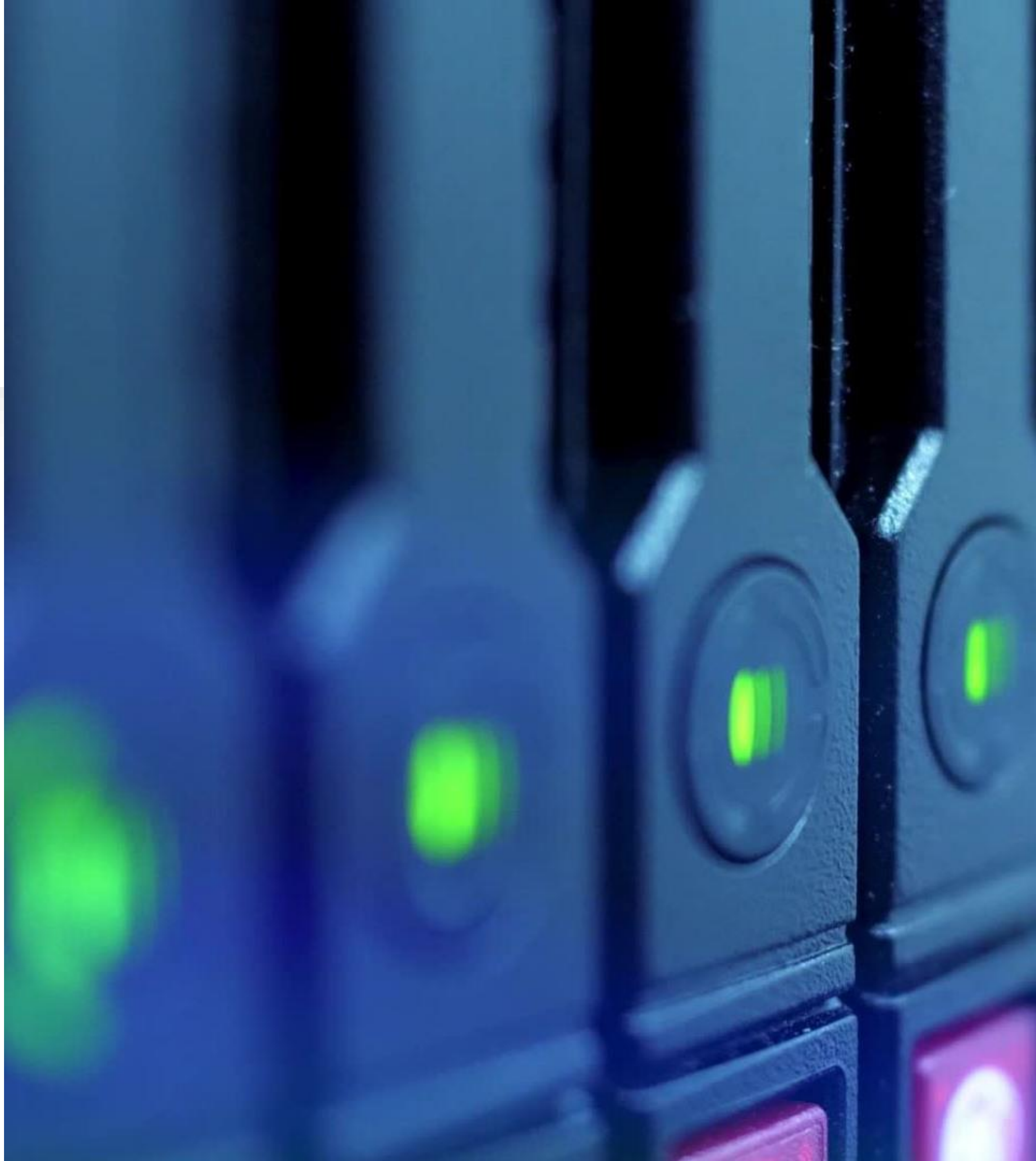
Database Connectivity
SQLite JDBC

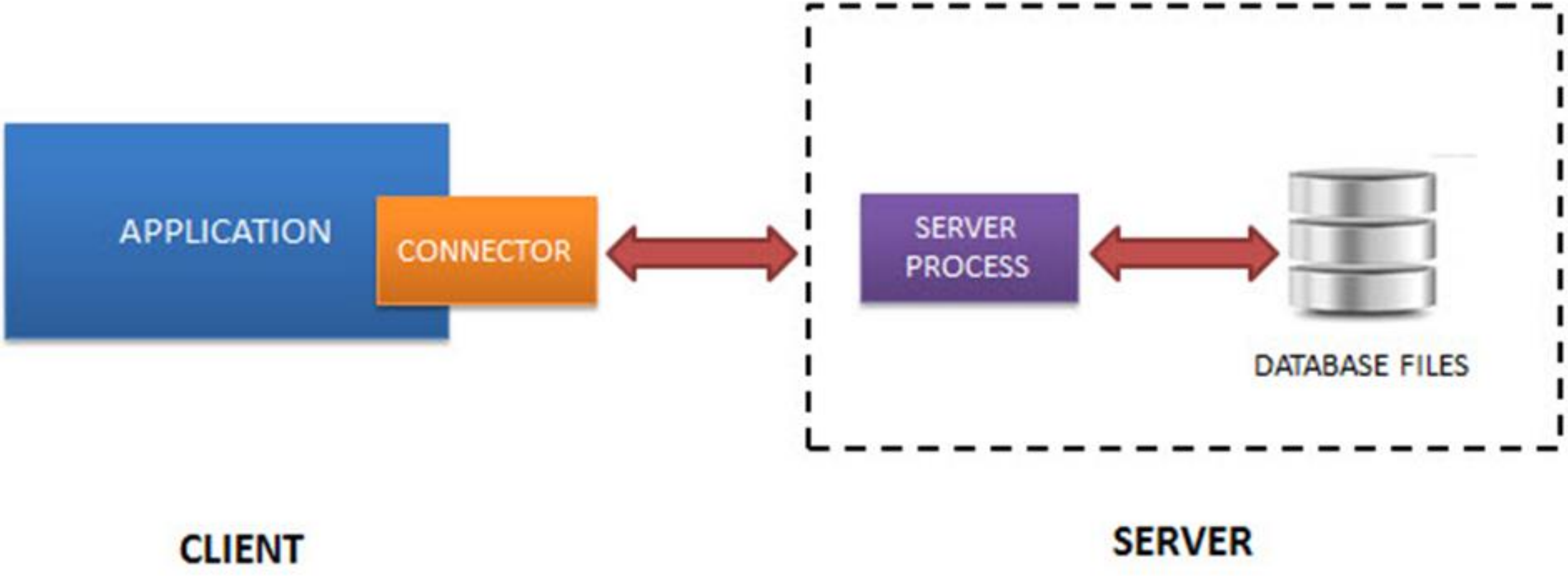
SQLite

- SQLite is a software library that provides a relational database management system
- The lite in SQLite means lightweight in terms of setup, database administration, and required resources
- SQLite has the following features: serverless, self-contained, zero-configuration, transactional



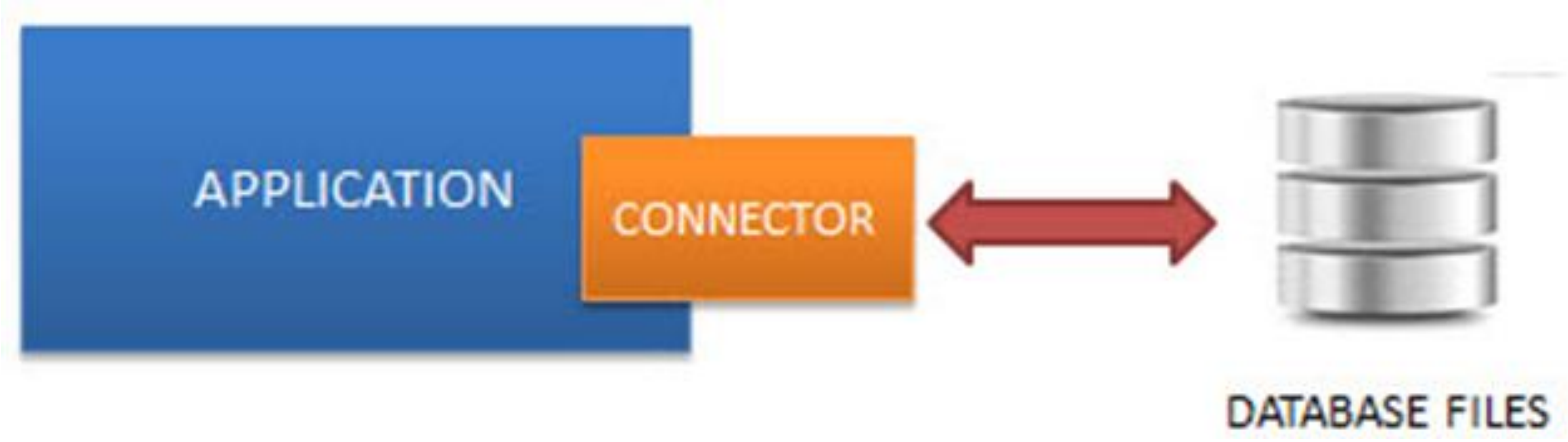
DBMS

- A relational DBMS such as MySQL, PostgreSQL, etc., requires a separate server process to operate



Serverless

- SQLite is a serverless DBMS. It does NOT require a server to run.
- SQLite database is integrated with the application that accesses the database



Self-Contained

SQLite is self-contained which means it requires minimal support from the operating system or external library

This makes SQLite usable in any environment especially in embedded devices like iPhones, Android phones, game consoles, handheld media players, etc

Zero-configuration

Because of its serverless architecture, you don’t need to “install” SQLite before using it

There is no server process that needs to be configured, started, and stopped

In addition, SQLite does not use any configuration files

Transactional

All transactions in SQLite are fully ACID-compliant which means all queries and changes are Atomic, Consistent, Isolated, and Durable

This means all changes within a transaction will either take place completely or not at all, even when an unexpected situation like application crash, power failure, or operating system crash occurs

Using the SQLite tool

SQLiteBrowser

https://sqlitebrowser.org/

SQLite Java Tutorial

https://www.sqlitetutorial.net/sqlite-java/sqlite-jdbc-driver/

Using SQLite JDBC to connect your database to Java code in VS Code

1. Download the SQLite JDBC driver: visit https://bitbucket.org/xerial/sqlite-jdbc/downloads/ and download the appropriate JAR file for your SQLite version.
2. Create a Java project: Create a new folder for your Java project or open an existing Java project in VS Code.
3. Inside your project folder, create a lib folder (if it doesn't exist) to store external libraries. Place the downloaded SQLite JDBC driver JAR file in this folder.
4. Copy your database file (e.g students.db) into the root directory of your project.

Using SQLite JDBC to connect your database to Java code in VS Code

5. Write Java code: Create a new Java file or open an existing Java file in VS Code, and write your Java code to connect to the SQLite database. Make sure to import the necessary classes, such as java.sql.*, to work with JDBC.

Using SQLite JDBC : Example code

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import java.sql.ResultSet;
import java.sql.SQLException;
public class Connect {
    public static void connect() throws ClassNotFoundException {
        Connection connection = null;
        Statement statement = null;
        ResultSet resultSet = null;
        try {
            // db parameters
            String url = "jdbc:sqlite:students.db";
            // create a connection to the database
            connection = DriverManager.getConnection(url);
            System.out.println("Connection to SQLite has been established.");
            // Create a statement object to execute SQL queries
} catch (SQLException e)  {
            System.out.println("Error Message!");
            System.out.println(e.getMessage());
        }
} }
```