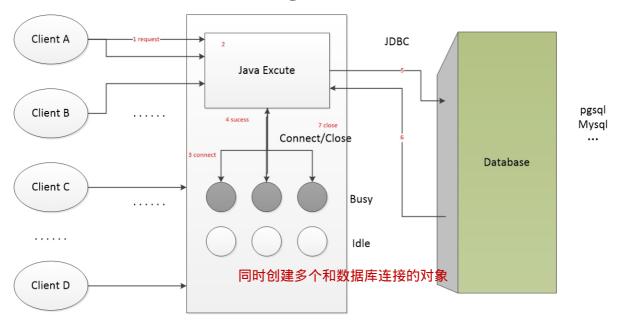# Database Connection Pooling

## Pooling

An import thought in program development, such as, Integer pool, string pool, object pool, connection pool, thread pool. Base on the thought of **reusing objects** to reduce resource consumption caused by creating and releasing objects, to achieve the goal that improving system performance.

**Tips**:

- reuse objects
- improve performance(save opening/closing time)
- a container(a pool)

## Database Connection Pooling



## Some Tests

**Test 1**: create a database connection pool and test connection statement

- use Druid pooling structure

  https://druid.apache.org/docs/latest/design/

- Client: Java-->DB: Postgresql

- set some parameters of pool, `setInitialSize`, `setMinIdele`, `setMaxActive`, `setMaxWait`

  `Step1`: Create a new Module and new java class to test `Druid` pooling, class name as you like

  `Step2`: Here we use `Maven` to manager dependencies. If there has version conflict with local configuration, please change the detail in file `pom.xml`.

`Step3` :  Import package and set connection information

```java
import com.alibaba.druid.pool.DruidDataSource;

...

dataSource.setDriverClassName("org.postgresql.Driver");
        dataSource.setUsername("User");// replace "User" by your User name
        dataSource.setPassword("pwd");
 //replace "pwd" by your Password
dataSource.setUrl("jdbc:postgresql://localhost:5432/postgres");
```

`Step4` : get connection and check the connection statement

```java
Connection connection = dataSource.getConnection();
        System.out.println(connection.getClass().getName());
```

**Test 2**: set some parameters of pool,  and discuss some inner mechanism

- inner proxy mechanism *(just 4 taking a look )
- set some parameters of pool, `initialPoolSize`, `MaxPoolSize`

```java
dataSource.setInitialSize(1);
dataSource.setMaxActive(1);
```

**Test3**: Get Statement of pool ,

- Three Number of Connection in Pool, `Busy`, `Idle`, `NumConnection`

```java
public static void poolStatus(ComboPooledDataSource dataSource) {
    try {
        System.out.println("Busy Num " +
dataSource.getNumBusyConnections());
        System.out.println("Idle Num" + dataSource.getNumIdleConnections());
        System.out.println("All Num" + dataSource.getNumConnections());
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

**Test4**: Simple Query using pooling technique

```java
// insert
String sql_insert = "......";
try {
    PreparedStatement ps_insert = connection.prepareStatement(sql_insert);
    ps_insert.executeLargeUpdate();
    poolStatus(dataSource);

}catch (Exception e){
    e.printStackTrace();
}
```