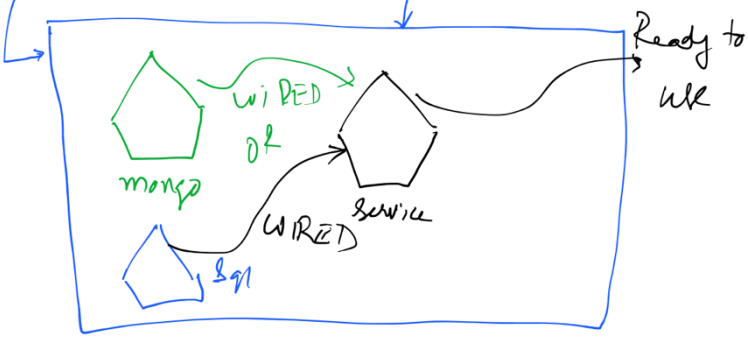
```java
interface EmployeeDao {
    addEmployee(Employee e);
}

public class EmployeeDaoMongoImpl implements EmployeeDao {
    addEmployee(...)
}

public class EmployeeDaoSQLImpl implements EmployeeDao {
    addemployee(...)
}

public class AppService {
    EmployeeDao empDao; // loosly coupled

    public void setDao(EmployeeDao edao) {
        this.empDao = edao;
    }

    insert(Employee e) {
        this.empDao.addEmployee(e);
    }
}
```

```xml
<beans>
    <bean id="mongo" class="pkg.EmployeeDaoMongoImpl" />
    <bean id="sql" class="pkg.EmployeeDaoSQLImpl" />
    <bean id="service" class="pkg.AppService">
        <property name="dao" ref="sql" />
    </bean>
</beans>
```

setDao( )

mongo

```
ApplicationContext ctx = new ApplicationContext("beans.xml");
```




```java
@Autowired
private EmployeeDao employeeDao;
```

OR

```java
@Repository
public class EmployeeDaoMongoImpl implements  EmployeeDao {
```

```java
@Repository
public class EmployeeDaoSqlImpl implements  EmployeeDao {
```

ERROR

BService

```java
@Autowired
@Qualifier("employeeDaoSqlImpl")
private EmployeeDao employeeDao;
```

```java
@Repository
public class EmployeeDaoMongoImpl implements  EmployeeDao {
```

```java
@Repository
public class EmployeeDaoSqlImpl implements  EmployeeDao {
```

AService

```java
@Autowired
@Qualifier("employeeDaoMongoImpl")
private EmployeeDao employeeDao; /,
```

JDBC

```java
@Override
public void addProduct(Product product) throws PersistenceException {
    String SQL = "INSERT INTO products(id, name, price, quantity) VALUES(0, ? ,? ,?)";
    Connection con = null;
    try {
        con = DBUtil.getConnection();
        PreparedStatement ps = con.prepareStatement(SQL);
        ps.setString(1, product.getName());
        ps.setDouble(2, product.getPrice());
        ps.setInt(3, product.getQuantity());
        ps.executeUpdate();
    } catch (SQLException e) {
        // log
        throw  new PersistenceException("unable to add product", e);
    } finally {
        DBUtil.closeConnection(con);
    }

}
```

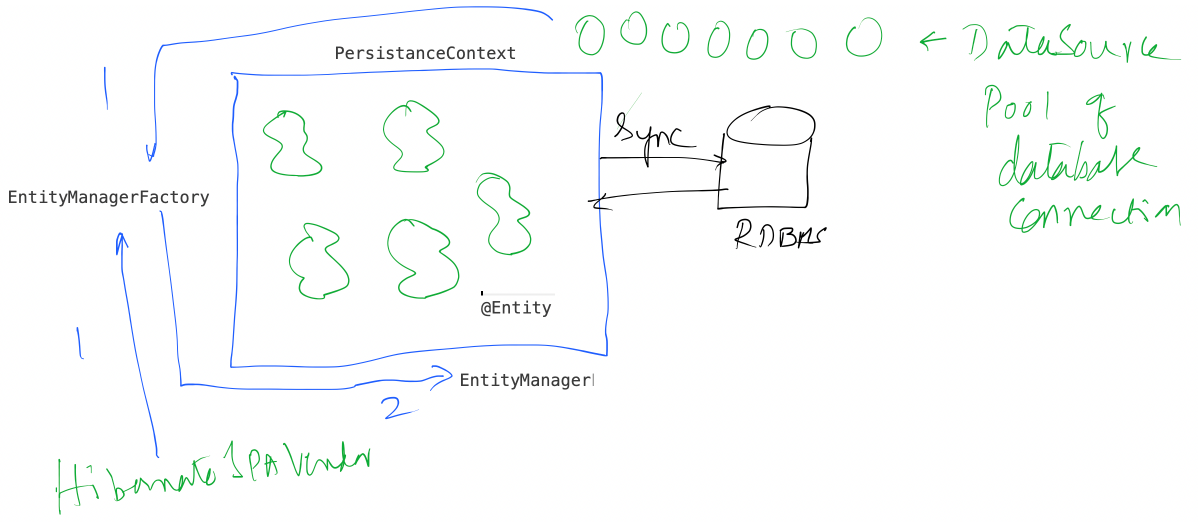
```java
@Override
public List<Product> getProducts() throws DaoException {
    String SQL = "SELECT id, name, price, quantity FROM products";
    Connection con = null;
    List<Product> products = new ArrayList<>();
    try {
        con = DBUtil.getConnection();
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(SQL);
        while(rs.next()) {
            products.add(new Product(
                    rs.getInt("id"),
                    rs.getString("name"),
                    rs.getDouble("price"),
                    rs.getInt("quantity")
            ));
        }
    } catch (SQLException e) {
        // log
        throw  new DaoException("unable to get products", e);
    } finally {
        DBUtil.closeConnection(con);
    }
```

Contacts

ORM

```java
public void addProduct(Product product) {
    em.save(product);
}
```

```java
public List<Product> getProducts() {
    TypedQuery<Product> query = em.createQuery(Product.class);
    return em.getQueryResult();
}
```



```
mysql> select * from customers;
+-----------------+-------------+
| email           | first_name |
+-----------------+-------------+
| raj@adobe.com   | Raj         |
| rita@adobe.com | Rita        |
+-----------------+-------------+
```

```
mysql> select * from products;
+----+----------------+--------+------+-----+
| id | name           | price  | qty  | ver |
+----+----------------+--------+------+-----+
|  1 | iPhone 14      |  89000 |  100 |   0 |
|  2 | Wacom          |   8000 |  100 |   0 |
|  3 | Sony Bravia    | 189000 |  100 |   0 |
|  4 | Logitech mouse |   1200 |  100 |   0 |
+----+----------------+--------+------+-----+
```

```
mysql> select * from orders;
+-----+----------------------------+--------+----------------+
| oid | order_date                 | total  | customer_fk    |
+-----+----------------------------+--------+----------------+
|   1 | 2023-11-22 12:26:06.460000 | 367000 | raj@adobe.com |
+-----+----------------------------+--------+----------------+
```

```
mysql> select * from items;
+---------+--------+----------+------------+----------+
| item_id | amount | quantity | product_fk | order_fk |
+---------+--------+----------+------------+----------+
|       1 | 189000 |        1 |          3 |        1 |
|       2 | 178000 |        2 |          1 |        1 |
+---------+--------+----------+------------+----------+
```