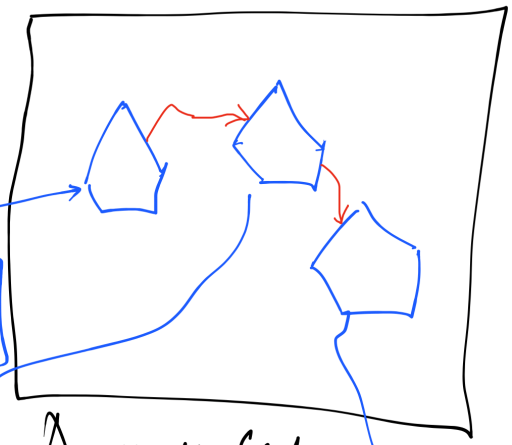
```java
public class ProductDaoJdbcImpl implements ProductDao {

    @Override
    public void addProduct(Product product) throws PersistenceException {
        String SQL = "INSERT INTO products (name, price, category, quantity) VALUES (?, ?, ? ,?)";
        Connection con = null;
        try {
            con = DBUtil.getConnection(); //latency
            PreparedStatement ps = con.prepareStatement(SQL);
            ps.setString(1, product.getName());
            ps.setDouble(2, product.getPrice());
            ps.setString(3, product.getCategory());
            ps.setInt(4, product.getQuantity());
            ps.executeUpdate();
        } catch (SQLException e) {
            throw new PersistenceException("unable to add product", e);
        } finally {
            DBUtil.closeConnection(con);
        }
    }

    @Override
    public List<Product> getProducts() throws FetchException {
        String SQL = "SELECT id, name, price, category, quantity FROM products";
        List<Product> products = new ArrayList<>();
        Connection con = null;
        try {
            con = DBUtil.getConnection();
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery(SQL);
            while(rs.next()) {
                products.add(Product.builder()
                        .id(rs.getInt("id"))
                        .name(rs.getString("name"))
                        .price(rs.getDouble("price"))
                        .quantity(rs.getInt("quantity"))
                        .category(rs.getString("category")).build());

                // products.add(new Product(rs.getInt("id"),
                // rs.getString("name"), rs.getDouble("price"),
                // rs.getInt("quantity"), rs.getString("category"));

            }
        } catch (SQLException e) {
            throw new FetchException("unable to get products", e);
        } finally {
            DBUtil.closeConnection(con);
        }
        return products;
    }

    @Override
    public Product getProduct(int id) throws FetchException {
        String SQL = "SELECT id, name, price, category, quantity FROM products WHERE id = ?";

        Connection con = null;
        try {
            con = DBUtil.getConnection();
        } catch (SQLException e) {
            throw new FetchException("unable to get product by id " + id, e);
        } finally {
            DBUtil.closeConnection(con);
        }
```

```java
public interface ProductDao extends JpaRepository<Product, Integer> {

}
```

```java
@SpringBootApplication
public class OrderappApplication {

    public static void main(String[] args) {
        SpringApplication.run(OrderappApplication.class, args);
    }

}
```

1) SpringContainer



ApplicationContext

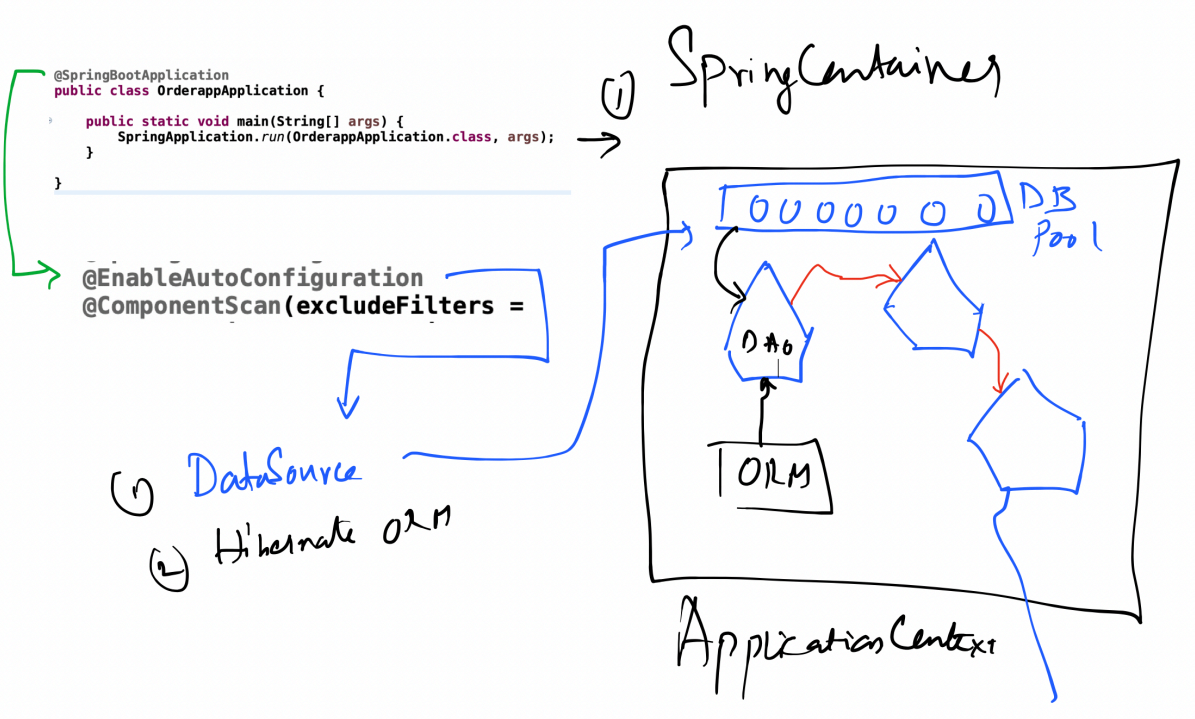
```java
@EnableAutoConfiguration
@ComponentScan(excludeFilters =
```

```java
public interface ProductDao extends JpaRepository<Product, Integer> {

}
```

class

```java
@Service
public class OrderService {
    @Autowired
    private ProductDao productDao;
```

```java
@Component
public class ProductClient implements CommandLineRunner {
    @Autowired
    private OrderService service;
```



meeting_room

ID	NAME	LOCATION	CAPACITY
21	CHANDRA	15	NOIDA
22	SIR MV	40	BENGALURU

employees

ID	NAME	DESIGNATION	EMAIL
18	A	MANAGER	a@adobe.com
61	B	TEAM LEAD	b@adobe.com

bookings

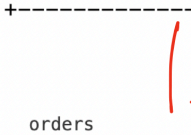
ID	booked_by	booked_date	training	room
800	b@adobe.com	19-JUL-23	2	22
801	a@adobe.com	20-JUL-23	1	21
802	a@adobe.com	21-JUL-23	2	21

trainings

ID	NAME	TRAINER
1	JAVA	T
2	WEB	R
3	C++	X

```
mysql> select * from customers;
+----------------+----------+--------+
| email          | fname    | lname  |
+----------------+----------+--------+
| raj@adobe.com  | Raj      | Kumar  |
| sam@adobe.com  | Samanta  | Ruth   |
+----------------+----------+--------+
```

```
+----+----------+----------------+--------+----------+
| id | category | name           | price  | quantity |
+----+----------+----------------+--------+----------+
|  1 | tv       | Sony Bravia    | 210000 |      100 |
|  2 | mobile   | iPhone 14      | 120000 |      100 |
|  3 | mobile   | OnePlus Nord   |  98000 |      100 |
|  4 | computer | Wacom          |   4500 |      200 |
|  5 | computer | Logitech Mouse |    980 |      200 |
|  6 | tv       | Tata Bing      |   9800 |      100 |
+----+----------+----------------+--------+----------+
```



orders

ORDER_ID	ORDER_DATE	TOTAL	customer_fk
123	19-JUL-23	89011	sam@adobe.com
124	20-JUL-23	600	raj@adobe.com
125	20-JUL-23	9800	sam@adobe.com

line_items

ITEM_ID	PRODUCT_FK	ORDER_FK	QTY	AMOUNT
422	4	123	2	9000.00
423	6	123	1	8900.00
425	4	124	1	4500.00