

```

public interface EmployeeDao {
    void addEmployee(Employee e);
}

public class EmployeeDaoJdbcImpl implements EmployeeDao {
    public void addEmployee(Employee e) {...}
}

public class EmployeeDaoMongoImpl implements EmployeeDao {
    public void addEmployee(Employee e) {...}
}

```

```

public class AppService {
    EmployeeDao empDao; // loosely coupled

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

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

```

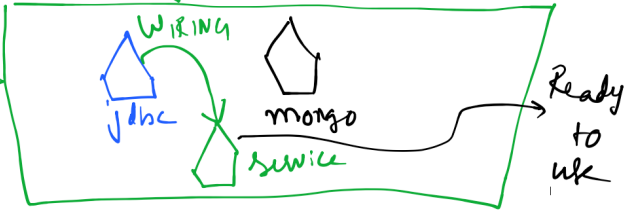
```

<beans>
  <bean id="jdbc" class="pkg.EmployeeDaoJdbcImpl" />
  <bean id="mongo" class="pkg.EmployeeDaoMongoImpl" />
  <bean id="service" class="pkg.AppService">
    <property name="empDao" ref="jdbc" />
  </bean>
</beans>

```

METADATA

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



```

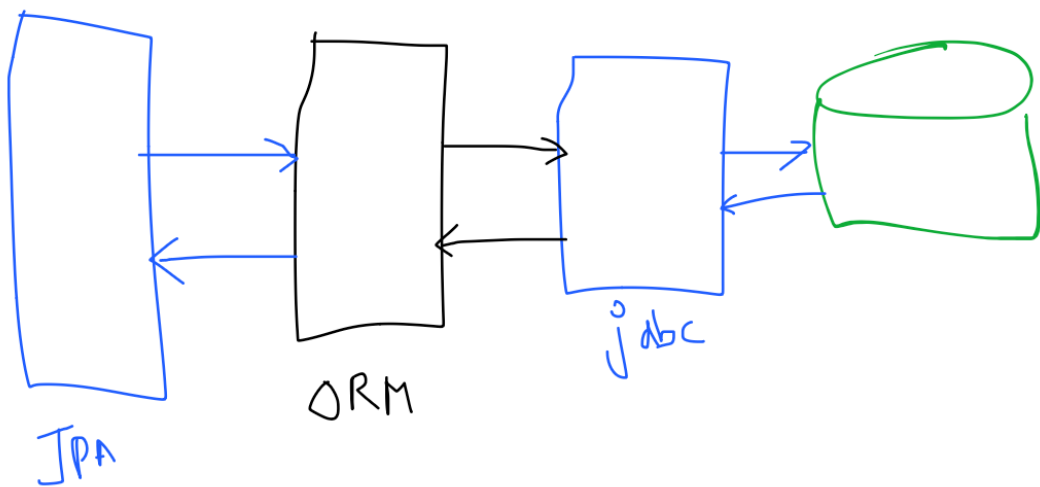
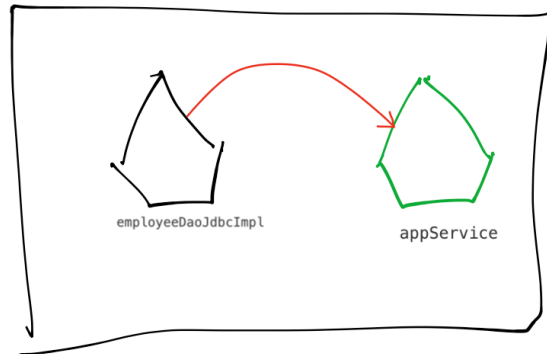
public interface EmployeeDao {
    void addEmployee(Employee e);
}

@Repository
public class EmployeeDaoJdbcImpl implements EmployeeDao {
    public void addEmployee(Employee e) {...}
}

@Service
public class AppService {
    @Autowired
    EmployeeDao empDao; // loosely coupled

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

```



```

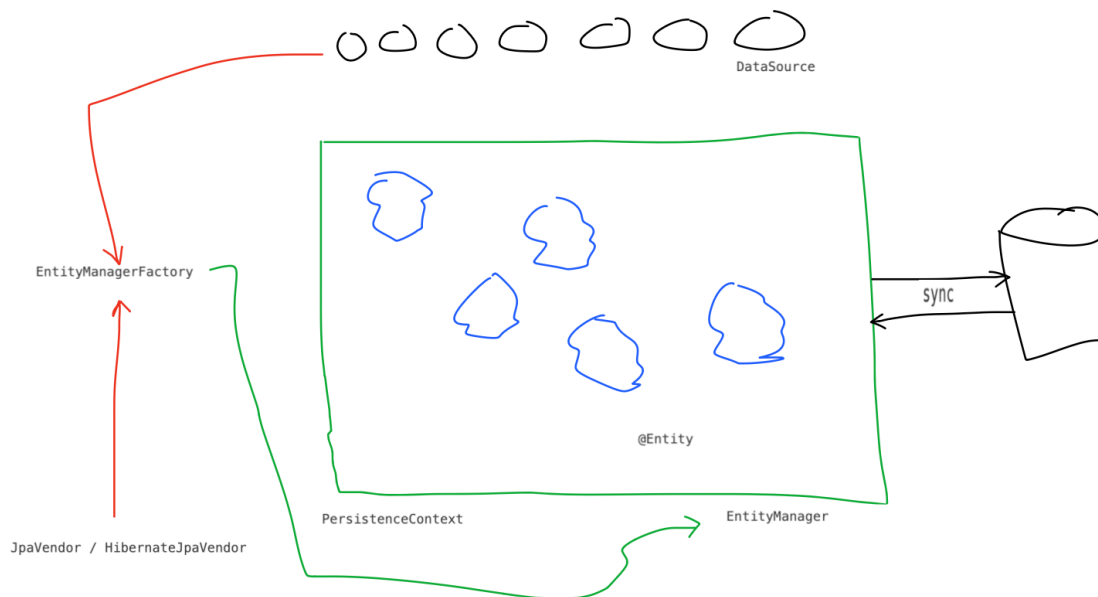
@Override
public void addProduct(Product product) throws DaoException {
    Connection con = null;
    try {
        String SQL = "INSERT INTO products (id, name, price, quantity) VALUES(?, ?, ?, ?)";
        con = DBUtil.getConnection();
        PreparedStatement ps = con.prepareStatement(SQL);
        ps.setString(1, product.getName());
        ps.setDouble(2, product.getPrice());
        ps.setInt(3, product.getQuantity());
        ps.executeUpdate();
    } catch (SQLException ex) {
        if(ex.getErrorCode() == 1062) {
            throw new DaoException("Product with given id: " + product.getId() + " already exists!");
        } else if(ex.getErrorCode() == 1054) {
            throw new DaoException("Unable to add Product!!!", ex);
        }
    } finally {
        DBUtil.closeConnection(con);
    }
}

```

```

public void addProduct(Product product) {
    em.persist(product);
}

```



SQL

```

mysql> select * from customers;
+-----+-----+-----+
| email | fname | lname |
+-----+-----+-----+
| ana@visa.com | Anna | Ben |
| Uma@visa.com | Uma | Thurman |
+-----+-----+-----+

```

1) Get all customers
select * from customers

2) Get first name and lastname customers
select fname, lname from customers

JPQL

```

@Entity
@Table(name="customers")
public class Customer {
    @Id
    private String email; // PRIMARY KEY

    @Column(name="FNAME", length = 100)
    private String firstName;

    @Column(name="LNAME", length = 100)
    private String lastName;
}

```

1) Get all customers
from Customer

2) Get first name and lastname customers
select firstName, lastName from Customer

```
mysql> select * from customers;
```

email	fname	lname
ana@visa.com	Anna	Ben
Uma@visa.com	Uma	Thurman

```
mysql> select * from products;
```

id	name	price	quantity
1	iPhone 16	980000	100
2	Macbook Pro	2350000	100
3	Wacom	5400	100
4	Reynold	50	100
5	Tata Play	4500	200
6	Logitech Mouse	890	700
7	LG AC	52000	100

orders

oid	customer_fk	order_date	total
533	uma@visa.com	12-SEP-2024 4:56:00	98000
534	ana@visa.com	14-SEP-2024 10:30:12	8111.00
535	uma@visa.com	20-SEP-2024 4:20:00	8000.00

line_items

Item_id	order_fk	Product_fk	Qty	amount
899	533	5	3	12000.00
900	533	1	1	85000.00
901	534	1	1	88000.00

