[企业应用架构阅读笔记3](https://www.cnblogs.com/zlj843767688/p/13096938.html)

时间： 2020-06-11

数据源架构模式

1、表数据入口：充当数据库表访问入口的对象。一个实例处理表中所有的行。

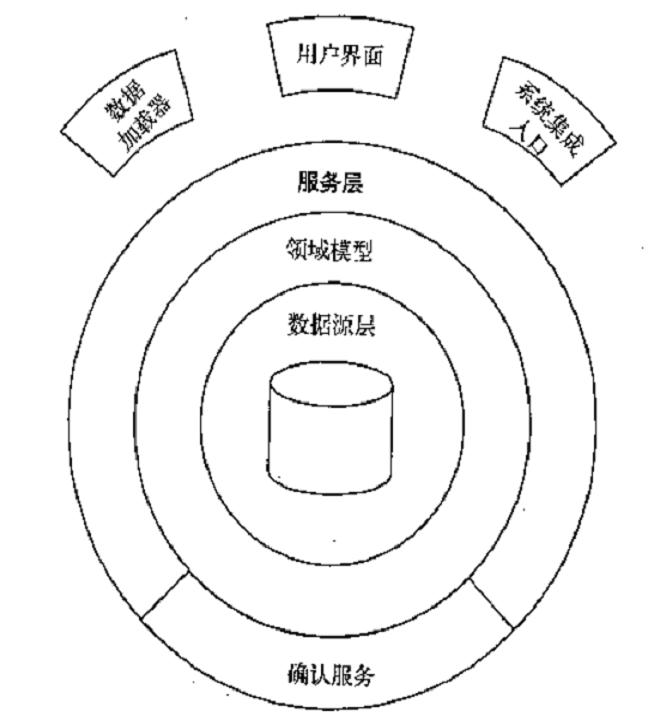
表数据入口包含了用于单个表或视图的所有SQL，如选择、插入、更新、删除等。其他代码调用它的方法来实现所有与数据库的交互。

2、行数据入口：充当数据源中单条记录入口的对象，每行一个实例。

3、活动记录：一个对象，它包装数据库表或视图中某一行，封装数据库访问，并在这些数据上增加了领域逻辑。

4、数据映射器：在保持对象和数据库彼此独立的情况下在二者之间移动数据的一个映射器层。

数据映射器是分离内存对象与数据库的一个软件层。其职责是在内存对象与数据库之间的传递数据并保持它们彼此独立。有了数据映射器，内存对象甚至不需知道数据库的存在；它们也不需要SQL接口代码，当然也不需要知道数据库方案。由于数据映射器是映射器的一种形式，因此数据映射器自身根本不为领域层所察觉。



对象—关系行为模式

1、工作单元：维护受业务影响的对象列表，并协调变化的写入和并发问题的解决。

工作单元是一个记录这些变化的对象。只要开始做一些可能会对数据库有影响的操作，就创建一个工作单元去记录这些变化。每当创建、改变或者删除一个对象时，就通知此工作单元。

工作单元的关键是在提交的时候，它决定要做什么。它打开一个事务，做所有的并发检查（使用悲观离线锁和乐观离线锁）并向数据库写入所做的修改。开发人员根本不用显式调用数据库更新方法。这样，他们就不必记录所修改的内容或者不必担心的引用完整性如何影响他们的操作顺序。

2、标识映射：通过在映射中保存每个已经加载的对象，确保每个对象只加载一次。当要访问对象的时候，通过映射来查找他们。

3、延迟加载：一个对象，它虽然不包含所需要的所有数据，但是知道怎么获取这些数据。

实现延迟加载的四种方法：延迟初始化、虚代理、值保持器、重影。

对象—关系结构模式

1、标识域：为了在内存对象和数据库行之间维护标识而在对象内存的一个数据库标识域。

数据库中通过主键来区分数据行，然而，内存对象不需要这样一个键，因此为对象系统能够保证正确的身份确认（在C++中是直接用原始内存位置）。

2、外键映射：把对象间的关联映射到表间的外键引用。

3、关联表映射：把关联保存为一个表，带有指向（有关联所连接的）表的外键。

4、依赖映射：让一个类的部分类执行数据库映射。

5、嵌入值：把一个对象映射成另一个对象表的若干字段。

6、序列化LOB：通过将多个对象序列化到一个大对象（LOB）中保存一个对象图，并存储在一个数据库字段中。

7、单表继承：将类的继承层次表示为一个单表，表中的各列代表不同类中的所有域。

8、类表继承：用每个类对应一个表来表示类的继承层次。

9、具体表继承：用每个具体类对应一个表来表示类的继承层次。

对象—关系元数据映射模式

1、元数据映射：在元数据中保持关系—对象映射的详细信息。

元数据映射使开发者可以以一种简单的表格形式来定义映射，这些映射可由通用代码来处理，从而实现读取、插入和更新数据的细节。使用元数据映射最主要的决策是如何根据运行代码来表示元数据中的信息。有两种主要的途径：代码生成和反射编程。

2、查询对象：描述一次数据库查询的对象。

3、资源库：协调领域和数据映射层，利用类似于集合的接口来访问领域对象。

WEB表现模式

1、模型—视图—控制器（MVC）：把用户界面交互分到不同的三种角色中。

2、页面控制器：在WEB站点上为特定页面或者动作处理请求的对象。

3、前端控制器：为一个WEB站点处理所有请求的控制器。

4、模板视图：通过在HTML也面中嵌入标记向HTML发送消息。

5、转换视图：一个视图，它一项一项地处理领域数据，并且把它们转换成HTML。

6、两步视图：用两个步骤来把领域数据转换成HTML。第一步，形成某种逻辑页面；第二步，把这些逻辑页面转换成HTML页面。

7、应用控制器：一个用来处理屏幕导航和应用程序流的集中控制点。