B.原文的翻译

SQLAlchem​​y的主要特点

一些一目了然的主要特点：

**没有必需的ORM**

SQLAlchem​​y的包括两个不同的组成部分，被称为核心和ORM。核心本身是一个全功能的SQL抽象的工具包，这个核心在多种DBAPI实现和行为上，以及允许通过Python表示的SQL语言上提供了一个光滑的抽象层。一个模式的代表性系统，既可以发出DDL语句类似于反射现有模式和类型系统，又可以让Python类型映射数据库类型，完善了制度。对象关系映射器则是一个可选包，它是建立在核心。许多应用程序都严格建立在核心，使用SQL表达系统提供了数据库交互简洁和详细的控制。

**成熟的，高性能的架构**

经历了70的持续发展，剖析和重构催生了一个高性能和准确的，覆盖了测试的，部署在数以千计环境中的工具包。在其第二或第三个完整的迭代几乎每一个主要组成部分，在几年前SQLAlchem​​y0.6版本大致是从0.4版本两倍的速度，和版本0.7和0.8仍在不断改善。其原生的执行速度已经和类似的工具相媲美，先进的ORM的功能，比如其工作单位，在内存中的集合，通过连接或二级子选择集合预先装载的竞争力，以及其他优化允许的SQLAlchem​​y的ORM发出更少的都比以前的版本更加高效。

**DBA认证**

内置符合什么数据库管理员的需求，包括使用手工优化报表的换出生成的SQL和为所有文字值绑定参数完整的用法，使用单元工作模式完成充分的事务化和批量的数据库写入。所有的对象关系方案的设计都是围绕适当的完整性引用，和外键作为其使用的一个组成部分。

**非自用的**

SQLAlchem​​y没有将最高的价值放在数据库和应用程序架构应用上。不像许多工具，它从来没有“生成”模式（不与发行用户定义的DDL混淆），或依赖于任何形式的命名约定。 SQLAlchem​​y的支持最广泛的数据库和合理可行的框架设计。

**工作单元**

工作单元的体系，是SQLAlchem​​y的的对象关系映射（ORM）的核心部分，组织挂起插入/更新/删除等操作到队列并把他们刷新放在一个批次。要做到这一点，他将队列中修改过的项目进行了拓扑排序，以彰显行间依赖关系，以及冗余的语句组合在一起被进一步的批处理。这将产生最大的效率和交易的安全性，并最大限度地减少死锁的可能性。“工作单元”模式和Java领先的关系对象模型Hibernate很相似。

**基于函数查询的构造**

基于函数的查询构造允许SQL子句通过Python函数和表达式来构建。整个范围，包括布尔表达式，运算符，函数表的别名，可选的子查询，插入/更新/删除语句，级联更新，选择和EXISTS语句，UNION语句，内部和外部连接，绑定参数，以及表达式中的文字文本的自由混合。构造的表达式可编译具体到任意数量的数据库供应商实现（如PostgreSQL的或Oracle）组成，由一个“语言”和“编译”的实现提供的组合决定。

**模块化和可扩展**

SQLAlchem​​y的不同部分可以被独立的应用于其他部分。就像连接池，SQL语句编译和事务服务元素可以彼此独立地使用，并且还可以通过各种插件分扩展。一个集成的事件系统允许自定义代码在五十多个交互点进行注射，包括核心语句执行，模式生成和反射，连接池操作中，对象关系的配置，持久化操作，属性突变事件和事务阶段。新的SQL表达式元素和自定义数据库类型可以构建成并实现无缝集成。

**独立的映射和类的设计**

在ORM标准化了一个“声明”构型系统，它允许用户定义的构造类的内嵌它们映射的表元数据，就像对象关系工具提供的那样。然而，该系统是完全可选的 - 在其核心，在ORM考虑到了用户定义的类，内联的表元数据，并且两个的映射是完全分开的。通过使用映射器的功能，任意一个Python类可以被映射到数据库表或图。映射类在各种缓存系统仍保留串行化的用法（pickle模块）。

**支持的平台**

通过SQLAlchem​​y的最新版本3.x的Python的支持2.5。其他支持的平台包括Jython和Pypy。

**支持的数据库**

SQLAlchem​​y支持的语言包括SQLite的和PostgreSQL，MySQL和Oracle，MS-SQL，Firebird，Sybase和其他，其中大部分支持多DBAPIs语言。其他语言发型为外部项目。相应的DB-API 2.0实现（或有时几个可用的一种）是需要使用每个特定的数据库。查看当前DBAPI支持