# SQL标准综述

# 附录：Mysql案例

## 数据库实例

## 设置字符集

## 完整的Create句法

# 附录：Sql类的设计

将Sql语句看成一个实体，显然这个实体可以根据命令的不同而分成更多的子类。

一个Select语句的基本格式

Select $WHAT From $Table Where Condition

选择的列名使用列表

选择的Table使用列表

选择的Condition同样使用列表

最主要的工作是SqlEntity的定义，一个SqlEntity既可以作为查询，也可以作为插入。

作为查询的时候，必须知道要使用哪些属性来配置，而属性的类型又是什么。

SqlEntity作为查询时，可以构造批量查询。类似于Scapy中的包批量构造那样，到头来都是笛卡尔积运算。

SqlEntity的每一个属性可以使用列表构造；使用单个的字符串或者数字初始化时，这个属性仍然被保存到列表中。后面将使用统一的循环遍历来完成。

初始化一个Entity的时候，指定以下属性：

属性名列表，类型（字符串，整数，浮点数）

# 附录：SqlAlchemy的使用

* 安装

pip3 install sqlalchemy

pip3 install mysql-connector==2.1.4 #mysql-connector

参考<http://stackoverflow.com/questions/43029672/unable-to-find-protobuf-include-directory>

* 导入

from sqlalchemy import Column, String, create\_engine

from sqlalchemy.orm import sessionmaker

from sqlalchemy.ext.declarative import declarative\_base

* 实体定义

# 创建对象的基类:

Base = declarative\_base()

#定义对象

class School(Base):

\_\_tablename\_\_ = 'school'

id = ...

name = ...

* 连接数据库

create\_engine()用来初始化数据库连接。SQLAlchemy用一个字符串表示连接信息：

'数据库类型+数据库驱动名称://用户名:口令@机器地址:端口号/数据库名'

* 插入数据

# 创建session对象:

session = DBSession()

# 创建新User对象:

new\_user = User(id='5', name='Bob')

# 添加到session:

session.add(new\_user)

# 提交即保存到数据库:

session.commit()

# 关闭session:

session.close()

* 查询数据

# 创建Session:

session = DBSession()

# 创建Query查询，filter是where条件，最后调用one()返回唯一行，如果调用all()则返回所有行:

user = session.query(User).filter(User.id=='5').one()

# 打印类型和对象的name属性:

print 'type:', type(user)

print 'name:', user.name

# 关闭Session:

session.close()

* 定义关系对应

class User(Base):

\_\_tablename\_\_ = 'user'

id = Column(String(20), primary\_key=True)

name = Column(String(20))

# 一对多:

books = relationship('Book')

class Book(Base):

\_\_tablename\_\_ = 'book'

id = Column(String(20), primary\_key=True)

name = Column(String(20))

# “多”的一方的book表是通过外键关联到user表的:

user\_id = Column(String(20), ForeignKey('user.id'))

* 例子

# 导入:

from sqlalchemy import Column, String, create\_engine

from sqlalchemy.orm import sessionmaker

from sqlalchemy.ext.declarative import declarative\_base

# 创建对象的基类:

Base = declarative\_base()

# 定义User对象:

class User(Base):

# 表的名字:

\_\_tablename\_\_ = 'user'

# 表的结构:

id = Column(String(20), primary\_key=True)

name = Column(String(20))

# 初始化数据库连接:

engine = create\_engine('mysql+mysqlconnector://root:password@localhost:3306/test')

# 创建DBSession类型:

DBSession = sessionmaker(bind=engine)