

# Socrates 项目介绍

# 什么是Socrates?

- 三元语义模型
  - **subject predicate object**
  - 通过 subject 和 predicate 组织 object 数据的关系
  - **object** 可扩展, 支持任意存储层数据类型
- 解决关系型数据库结构固定, 不容易修改的问题
- 解决稀疏表的存储问题
- 建立于数据存储层之上, 兼容不同类型的数据库
- 规则简单
  - 元数据信息, 自描述
  - 结构开放, 可以开发第三方接口
  - 结构简单, 基于递归和引用, 容易使用
- 数据粒度小, 伸缩性好, 容易并发
- 可插入到其它体系中使用

# Socrates 使用

- 基于SQLAlchemy
  - Storage
- 基于URI, 支持自反射, 容易联接
  - Storage(uri)
- 基于元语的存储层注册机制
  - Storage.registe
  - Storage.registered\_type
- 条目ID自我管理
  - 透明、跨平台的统一序列生成器
- 基于三元语义的访问模型
  - writePredicate(name, type)
  - writeSubject(subject\_id, segments, \*\*{segments})

# Socrates元语——基础指令

- SUBJECT = 1
- PREDICATE = 2
- IS = 3
- OBJTYPE = 4
- TYPE = 5
- STRING = 6
- NAME = 7
- HAS = 8
- SET = 9
- STORAGE = 10

# Socrates元语——基础定义

- - subject is subject
- - subject has name
- - predicate is subject
- - predicate has name
- - is is predicate
- - is has name
- - is objType subject
- - objType is predicate
- - objType has name
- - objType objType subject
- - type is subject
- - type has name

# Socrates示例--KUSE 日志系统

- `storage.registe('datetime', DateTime, create=True)`
- `storage.writePredicate('kssoid', 'string')`
- `storage.writePredicate('who', 'subject')`
- `storage.writePredicate('when', 'datetime')`
- `storage.writePredicate('search', 'string')`
- `storage.writePredicate('visit snapshot', 'string')`
- `storage.writePredicate('visit url', 'string')`

# Socrates示例--KUSE 日志系统

- `storage.writeSubject(None, {"search":keywords}  
when=datetime.now, who="kssoid subject")->new subject  
id`

# Socrates已实现的技术点

- **Storage()**

- 使用SQLAlchemy URL
- 允许指定schema
- 自引导
  - 新建基础结构
  - 读取已有结构

- **registered**

- 允许运行时建立存储类型
- 可以指定存储表
- 存储定义持久化

# Socrates已实现的技术点

- `registred_type`
  - 直接存储table对象，构建查询时无需考虑schema问题
  - 查询时遍历此map
  - 允许多类型存储于同一表（即同义类型）

# Socrates讨论——正在谋求解决的问题

- 多重语义谓词：
  - 涉及数值的谓词往往要隐式支持多类型
    - avg
    - sum
    - ...
  - 日期与时间类型
    - date
    - datetime
- 解决：
  - 谓词重定义
    - sum objType integer
    - sum objType float
    - sum objType decimal

# Socrates讨论——正在谋求解决的问题

- 多重语义谓词：
  - 写入时的类型判断
- 解决：
  - 多重谓词必须显式引用
  - `predicate:objType`
  - `"predicate words":objType`
- 谓词语句化：
  - 允许引号包含的文本化谓词
- 以上需要简单的语法分析器支持
  - 逃逸字符过滤
  - 引号提取
  - 词法分析

# Socrates讨论——正在谋求解决的问题

- 匿名条目
  - KSSOID
- 解决：
  - 直接编写 predicate object 子句

# Socrates讨论——正在谋求解决的问题

- 查询
  - 需要多次select
- 解决：
  - 先使用 SQLAlchemy的查询功能
  - 积累常用基础指令集：
    - find predicate by name
    - find segment by query(predicate, object)
    - search by object
    - search by predicate
      - search by name