**灵派诺达**

**数据设计文档**

**(内部使用)**

**版本历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本/状态** | **作者** | **参与者** | **起止日期** | **备注** |
| 0.1发布 | 王晖 |  | 2017-06-01 | 重新设计，建立新的文档 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**SQL文档**

参见”MySqlDDL.sql”

**目录**

[0. 文档介绍 4](#_Toc419150398)

[0.1. 文档目的 4](#_Toc419150399)

[0.2. 文档范围 4](#_Toc419150400)

[0.3. 读者对象 4](#_Toc419150401)

[0.4. 文档说明 4](#_Toc419150402)

[0.5. 关系说明 4](#_Toc419150403)

[1. 数据库环境说明 5](#_Toc419150404)

[2. 数据库规范 5](#_Toc419150405)

[2.1. 表的命名规则 5](#_Toc419150406)

[2.2. 布尔值取值 5](#_Toc419150407)

[2.3. 键 5](#_Toc419150408)

[2.4. 字段/列 6](#_Toc419150409)

[2.5. 数值类型 6](#_Toc419150410)

[3. 编程注意 7](#_Toc419150411)

[4. 设计过程说明 7](#_Toc419150412)

[4.1. 第一期内容 7](#_Toc419150413)

[5. 具体设计 7](#_Toc419150414)

[5.1. 表汇总 7](#_Toc419150415)

[5.2. 用户类{1} 8](#_Toc419150416)

[5.2.1. 001用户[PLAT\_USER] 8](#_Toc419150417)

[5.3. 字典(枚举)类{2} 9](#_Toc419150418)

[5.3.1. 002字典组[PLAT\_DICTM] 9](#_Toc419150419)

[5.3.2. 003字典项[PLAT\_DICTD] 11](#_Toc419150420)

[5.4. 元数据信息{6} 12](#_Toc419150421)

[5.4.1. 004元数据模式[SA\_MD\_TABMODEL] 12](#_Toc419150422)

[5.4.2. 005元数据列描述[SA\_MD\_COLUMN] 13](#_Toc419150423)

[5.4.3. 006元数据列语义[SA\_MD\_COLSEMANTEME] 13](#_Toc419150424)

[5.4.4. 007实体表指标[SA\_MD\_TABQUOTA] 14](#_Toc419150425)

[5.4.5. 008实体列指标[SA\_MD\_COLQUOTA] 15](#_Toc419150426)

[5.4.6. 009元数据表对照[SA\_MD\_TABMAP\_REL] 15](#_Toc419150427)

[5.5. 数据导入类{1} 16](#_Toc419150428)

[5.5.1. 010数据文件/实体表对应[SA\_IMP\_TABMAP\_REL] 16](#_Toc419150429)

[5.6. 文件管理{5=3+2} 16](#_Toc419150430)

[5.6.1. 011文件记录索引[SA\_FILE\_INDEX] 16](#_Toc419150431)

[5.6.2. 012文件分类 [SA\_FILE\_CATEGORY] 17](#_Toc419150432)

[5.6.3. 013文件关系[SA\_FILE\_REL] 18](#_Toc419150433)

[5.6.4. 014反向文件关系[vSA\_FILE\_INVERSEREL] 19](#_Toc419150434)

[5.6.5. 015数据导入日志[vSA\_IMP\_LOG] 19](#_Toc419150435)

[5.7. 报告管理{2=1+1} 20](#_Toc419150436)

[5.7.1. 016报告信息[SA\_REPORT\_INFO] 20](#_Toc419150437)

[5.7.2. 017报告文件信息[vSA\_REPORT\_FILE] 21](#_Toc419150438)

[5.8. 任务管理{4=3+1} 22](#_Toc419150439)

[5.8.1. 018任务组信息[SA\_TASK\_GROUP] 22](#_Toc419150440)

[5.8.2. 019任务信息[SA\_TASK\_INFO] 23](#_Toc419150441)

[5.8.3. 020任务关系[SA\_TASK\_REL] 24](#_Toc419150442)

[5.8.4. 021任务全信息[vSA\_TASKS] 24](#_Toc419150443)

[6. 数据逻辑视图 28](#_Toc419150444)

[7. 数据库管理与维护说明 29](#_Toc419150445)

1. 文档介绍
   1. 文档目的

描述平台的数据设计，为平台建造提供核心信息架构。

平台的代码设计与实现必须以此文档为依据。

* 1. 文档范围

包括底层平台的

1-数据库（目前为MySql）

2-基本元数据——不存在数据库中的内容

* 1. 读者对象

·平台开发组人员、测试人员、及产品相关人员；

·基于数据平台进行开发的人员；

* 1. 文档说明

·**红色粗体字体**部分为需要注意的内容或最新版本修改的内容；

·**蓝色粗体字体**部分为需要完善和补充的内容；

·在表格中：黄色底的字段为主键；

·在表格中：淡蓝绿底的字段为外键或其值从其他表或元数据得到；

·在表格中：粉色底的字段为冗余字段；

* 1. 关系说明

·本设计基于平台的数据库设计，参见(简单用户模块及权限设计(DB).docx)；

1. 基本元数据

这部分数据是业务说明，可以扩展。

在程序中用枚举的方式使用，若是这样，则鼓励采用配置文件的方式。

也可以把这类数据存储在数据库中。

* 1. 所有者或参与者

是对参与系统的各相关主体的一个顶层描述。

此元数据由两部分组成ownerType和ownerId。[此命名规则参考3.5]

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **数据类型** |
| ownerType | 所有者或参与者类型 | 字符串，是整型字符串，见下面的说明。  此字段可能会对应到某一本表。 |
| ownerId | 所有者Id | 可能是相应表中的Id |

* + 1. ownerType

|  |  |
| --- | --- |
| **释名** | 所有者或参与者类型，目前有三类：系统、人员、组织。 |
| **取值** | 取5位，分三段：  第一段：第一位——大分类，目前有1=系统；2=人员；3=组织  第二段：第二三位——小分类，使用者自定义  第三段：第四五位——具体编号，使用者自定义  每位可以取1~9|a~z|A~Z，共52种状态； |
| **意义**  **规则** | **【系统】1xxxx：**  指IT系统。可以是子系统，外围的业务系统，数据导入或抓取中的原系统等等。目前有——   * 100xx：我们自己的系统，如10001=灵派诺达平台 * 101xx：外部数据源 |
| **【用户】2xxxx：**  自然人信息   * 200xx：我们自己的人员，如20000=灵牌诺达注册用户，，如20011=无状态App前端；20012=无状态Pc前端； |
| **【组织】3xxxx：**  组织部门信息等，法人信息等   * 300xx：我们自己的组织，如30000=灵牌诺达注册用户自发组成的用户组；30001管理人员所组成的组 |

* + 1. ownerId

|  |  |
| --- | --- |
| **释名** | 所有者或参与者的Id，很可能是某一类（人或组织）表的Id |
| **取值** | 32位字符串，UUID |

* + 1. 说明

所有者元数据也可以存储在关系数据库中，也鼓励采用此标准的项目把这类信息数据库化。从本项目来说，此元数据的离散值不多，所以这部分不全部数据库化。

若数据库化ownerType代表一个表，ownerType的值是这个表在此元数据内的编码。ownerId是表的Id。ownerType的三段内容，可以出现在表中的某个字段中。

* + 1. 使用

在使用过程中可以是创建者creatorType，createrId，也可是所有者ownerType，ownerId。若ownerType是固定的，那么也可以省略，请参考plat\_User表。

1. 数据库环境说明

希望满足跨数据库平台的要求，但目前只对MySql数据库。

若以Oracle为基准，表空间统一为platform，数据库用户统一为platform，口令platform，权限限制为资源访问（如：记录的增、删、改、查等）和应用级管理（建表、改表、删表、建索引等）。

**注意：各项目可以有自己的用户名，如海南项目——platform\_hn。**

1. 数据库规范
   1. 符号规则

·元域名——不同元数据的名称。

·分割符——只有下划线“\_”。

·必选——用中括号括起的元域名和分割符，如[元域名\_]。

·可选——用大括号括起的元域名和分割符，如{\_元域名}。

* 1. 数值类型

所有用数值类型表示状态分类的字段，0都作为空值处理，这样避免程序中没有赋值的数值变量对数据库的影响。

* 1. 布尔值取值

·用数值类型表示布尔值，数据库种类型为“NUBMER”，java中类型为int；

·0（空）、1（是）、2（否）；说明：**由于java中数值int的默认值为0，为便于编程采用本条规则；**

·布尔字段包括包括字段：

1. **isValidate**—是否有效：1（有效）、2（无效），与delTag意思是相近的。

**b)hasChild**—是否有子节点：1（是）、2（不是）；

* 1. 表的命名规则

·前缀：不同类型的表有自己的前缀，前缀小写。

·功能名：各单词大写字母开头中间无空格。

·后缀：全部大写——目前只有

* 与权限相关的表用后缀“ORG”标明；
* 与关联关系相关的表用后缀“REL”标明。

表名=[前缀\_][业务表名\_]{后缀}，如：分析平台部分的表以“sa”命名，如sa\_MdTable(元数据-表)；

·表名最长为100个字符(由程序判断)，表tableName字段长度的限制。

* 1. 字段/列命名规则

·全部小写：在Oracle中对象名称大小写不敏感，除非新建时用””框起来。

·按照Java变量的命名规则，英文单词首字母大写，第一个英文单词首字母不大些。

·是判断类型的用is+单词，并且单词首字母大写，如isValidate，isUnit

·为了避免一些数据库系统的保留字，type，name，level均不作为表名或列明（包括其它一切用户对象的名称）

·常用的命名及规范：

a)isValidateNUMBER //是否生效，1有效；2无效，默认值1

c)descnVARCHAR2(2000) //说明字段

* 1. 键

·主键用id命名，UUID或长整型，若为长整型，则从1开始编号，若为UUID则长度为VARCHAR2(32)。

·父结点用pId命名；

·主子关系，子表到主表的外键用mId；

·其他外键命名：用功能缩写Id，如uId或userId：用户ID，ddId：字典项ID；

* 1. 通用字段说明
     1. 树型数据库

树结构通用规范，所有的树都要按这个建立。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段中文名** | **字段名** | **数据类型** | **空否** | **约束条件** | **说明** |
| 主键 | id | VARCHAR | 非空 | 主键PK | 从PLAT\_SERIALNO生成 |
| 父结点ID | pId | VARCHAR | 非空 | 默认值为0 | 用户组表中的ID(外键)  若是第一级用户组，此值为0 |
| 是否有子结点 | hasChild | NUMBER | 非空 | 默认值2无子 | 1有结点,2无子结点 |
| 字典项层数 | levels | NUMBER | 非空 | 默认值1无子 | 从1开始 |
| 排序 | sort | NUMBER | 非空 | 默认值0 | 数值大者靠前，从0-99999 |
| 是否生效 | isValidate | NUMBER | 非空 | 默认值1有效 | 1有效；2无效 |

* + 1. 通用类型

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段中文名** | **字段名** | **数据类型** | **约束条件** | **说明** |
| 主键 | id | VARCHAR | 主键PK | 从PLAT\_SERIALNO生成  或32位UUID |
| 是否生效 | isValidate | NUMBER | 默认值1有效 | 1有效；2无效 |
| 排序 | sort | NUMBER | 默认值0 | 数值大者靠前，从0-99999 |
| 创建时间 | cTime | TIMESTAMP(4) | 默认值当前时间 | 创建这条记录的时间 |
| 最后修改时间 | lmTime | TIMESTAMP(4) |  | 记录修改时间 |

* 1. 表中字段建议

建议性字段，星号越多建议性越强，最高三星。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **中文名** | **字段名** | **数据类型** | **建议性** | **说明** |
| 主键 | id | VARCHAR | ★★★ | 建议采用UUID，若是系统自己用的表，也可以没有id主键，而用联合主键 |
| 是否生效 | isValidate | NUMBER | ★★ | 建议任何与真实实体相关的表都有这个字段，此字段与isDel字段在某些情况下可通用，但意义相反 |
| 创建者Id | creatorId | NUMBER | ★★ | 创建者的类型都是20000，所以省略掉了[参考1.1基所有者或参与者] |
| 创建时间 | cTime | TIMESTAMP(4) | ★★★ | 自动生成，当前时间 |
| 最后修改时间 | lmTime | NUMBER | ★ | 与时间维度相关性高的可以有这个字段 |

这里的字段，时建议在任何表中都出现的

1. 设计过程说明
   1. 第一期内容

只包括：用户、用户组、字典、模块、权限

1. 具体设计
   1. 本项目的基础元数据
      1. 所有者或参与者

**业务意义：**

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **ownerType** | **ownerId** | **说明** |
| **100** | 10000 | 灵派数据 | 自己的主系统 |
| **101** | 10100 | 微信、微博、QQ等 | 外部社交系统 |
| **200** | 20000 | 000000 | 系统超级管理员 |
| **201** | 20000 | plat\_User中的Id | 系统用户表plat\_User |
| **202** | 20011 | 设备IMEI | 无状态App设备编号 |
| **203** | 20012 | SessionId | 无状态Pc设备J2EE规范的SessionId |

很多情况下100和200的业务意义是相同的。

**表中的使用**

|  |  |  |  |
| --- | --- | --- | --- |
| **表** | **ownerType**  **对应的表字段** | **ownerId**  **对应的表字段** | **说明** |
| **plat\_DictD** | creatorType | creatorId | **用户：**用于说明是哪个用户创建的。用到的业务意义200/201/202/203  **系统：**只是默认值，用到的业务意义100  **组织：**无意义，若要体现字典是用于哪个组织，在权限中处理。 |
| **plat\_DictM** |
| **plat\_ThirdUser** | thirdSysType | thirdSysId | **系统：**用到业务意义101 |
| **plat\_UserLogin** | sysType | sysId | **系统：**目前仅支持自己的系统，用到业务意义100 |
| **log\_Visit** | fromSysType | fromSysId | **系统：**从那个系统访问的，默认值业务意义100 |
| visitSysType | visitSysId | **系统：**访问那个系统，默认值业务意义100 |
| visitorType | visitorId | **用户：**请求用户，用到的业务意义200/201/202/203 |

* 1. 表汇总

下表按功能对表进行了划分，在具体介绍表的结构时也采用这一划分。

目前平台表分为如下：

|  |  |  |
| --- | --- | --- |
| **分类** | **表名** | **功能说明** |
| **字典类{2}** | D001-plat\_DictM | 字典组 |
| D002-plat\_DictD | 字典项 |
| **用户类{4}** | U001-plat\_User | 用户表 |
| U002-plat\_UserExt | 用户扩展信息表 |
| U003-plat\_ThirdUser | 第三方登录 |
| U004-plat\_UserLogin | 已登录用户信息 |
| **日志类{1}** | L001-log\_Visit | 访问日志 |

* 1. 字典(枚举)类{2}

字典结构由字典组合字典项构成。

字典组是一类字典项的统称，字典项采用排序树进行存储。

如“行政区划”，由行政单位组成的树，这棵树存储在字典项表中，而字典组表中存储“行政区划”这个统称及一些其他信息。

另外，一级字典项，也用树来存储，如“性别”，其存储在字典项中就是单级树。

* + 1. D001字典组[plat\_DictM]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **表名** | **plat\_DictM（字典组）** | | | | |
| **意义** | **字段名** | **数据类型** | **空否** | **约束** | **说明** |
| 字典组ID | id | VARCHAR2(32) | 非空 | 主键PK | 从PLAT\_SERIALNO生成 |
| 创建者类型 | creatorType | VARCHAR2(5) | 非空 | 默认值  10000 | 20000=平台  20011=无状态App前端；  20012=无状态Pc前端； |
| 创建者ID | creatorId | VARCHAR2(32) | 非空 | 默认值  灵派数据 | 当cT=20011：IMEI；  当cT=20012：SessionId；  当cT=20000：用户Id，且id=000000的为系统本身创建 |
| 字典组名称 | dmName | VARCHAR2(200) | 非空 |  |  |
| 名称拼音 | nPy | VARCHAR2(800) | 可空 |  |  |
| 排序 | sort | NUMBER | 非空 | 默认值0 | 数值大者靠前，从0-99998 |
| 是否生效 | isValidate | NUMBER | 非空 | 默认值1 | 1有效；2无效 |
| 类型 | mType | NUMBER | 非空 | 默认值3 | 1系统保留；2、系统；3定义； |
| 引用描述 | mRef | VARCHAR2(400) | 可空 |  | 可以是：  1-所引用的字典组：用“#+MID”标明的用“，”号分开的字典项ID，如“#3,#12”表明此字典引用了id为3和12的字典组；  2-所参考的文件名：标准文件名称，如《国家XX标准 GB-23212》;  3-所参考的文件（存储在系统中）：用“$+FID”标明的用“，”号分开的文件ID  若包括多项以上内容，则各项间用“；”隔开。  如：#3,#45;《国家XX标准 GB-23212》,《国家XX标准GB-23》;#45;$2034,$2342 |
| 创建时间 | cTime | TIMESTAMP(4) | 非空 |  |  |
| 最后修改时间 | lmTime | TIMESTAMP(4) | 非空 |  |  |
| 说明 | descn | VARCHAR2(500) | 可空 |  |  |
| **补充说明** | 1-通过此表中的记录管理不同的字典项，资源与字典项的对应见表：PLAT\_RESOURCE\_SCHEMA(注意要进行必要的修改)  2-关于mType的说明：  mType=1，系统保留，只能修改名称和排序；不能向其内部加入新的字典项，其下属字典项的dType必须是1；（在本系统中，这应该是汇总得到的）  mType=2，系统，只能修改名称和排序；可以向其下增加字典项，其下属字典项的dType是2或3；  mType=3，用户定义的字典组，可以修改和删除，并向其下加字典项，其下属字典项的dType必须3或4；(用户自己定义的)  3-修改了如下内容，要在本此平台整理中修改过来：  **4-creatorType,creatorId与mType:**  **当mType=1时，creatorType值必须为20000, creatorId必须为000000**  **当mType=2时，creatorType值必须为20000, creatorId可为其他值**  **当【creatorType=20000, creatorId=000000】或【creatorType=10000, creatorId=灵派数据】**  **时，mType值必须为1，若不为1也要按1处理**  **另：当creatorType=20000, creatorId=000000时，默认的系统是自己的系统，即等于creatorType=10000, creatorId=灵派诺达**  **5-若要给字典组加入图标，请参看文件的处理** | | | | |
| **索引** | 1. id，主键 2. sortIdx(sort)，排序的索引 3. bizIdx001(ownerType, ownerId, dmName) 唯一索引，表明相同用户下，不能有同名字典组 | | | | |

* + 1. D002字典项[plat\_DictD]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **表名** | **plat\_DictD（字典项）** | | | | | |
| **意义** | **字段名** | **数据类型** | **空否** | **约束条件** | **说明** |
| 字典项ID | id | NUMBER | 非空 | 主键PK | 从PLAT\_SERIALNO生成 |
| 主表字典组ID | mId | NUMBER | 非空 | 默认值为0，无意义 | 字典组表中的ID(外键) |
| 父结点ID | pId | NUMBER | 非空 | 默认值为0，无意义 | 字典明细表的ID(外键)  若是明细表中的第一级字典项，此值为0 |
| 创建者类型 | creatorType | VARCHAR2(5) | 非空 | 默认值  10000 | 20000=平台  20011=无状态App前端；  20012=无状态Pc前端； |
| 创建者ID | creatorId | VARCHAR2(32) | 非空 | 默认值  灵派数据 | 当cT=20011：IMEI；  当cT=20012：SessionId；  当cT=20000：用户Id，且id=000000的为系统本身创建 |
| 排序 | sort | NUMBER | 非空 | 默认值0 | 数值大者靠前，从0-999999 |
| 是否生效 | isValidate | NUMBER | 非空 | 默认值1有效 | 1有效；2无效 |
| 字典项名称 | ddName | VARCHAR2(200) | 非空 |  |  |
| 名称拼音 | nPy | VARCHAR2(800) | 可空 |  |  |
| 别名或简称 | aliasName | VARCHAR2(200) | 可空 |  | 类似于省的简称，如川 |
| 别名拼音 | anPy | VARCHAR2(800) | 可空 |  |  |
| 业务编码 | bCode | VARCHAR2(50) | 非空 |  | 在新增时，若业务编码为空，则默认把业务编码设置为字典项的ID  在同一字典组内不能有相同的业务编码。  若是引用字典项，则其业务编码为“所引用字典组ID+’$’+所引用字典项的业务编码”，引用字典项此值不能改变 |
| 字典项类型 | dType | NUMBER | 非空 | 默认值3定义 | 1系统保留；2系统；3定义；4强联引用：随关联的删除；  5弱联引用：只随关联修改 |
| 引用描述 | dRef | VARCHAR2(4000) | 可空 | 默认值为0，无意义 | 若dType=4|5，则此为引用的：字典组ID:字典项ID |
| 创建时间 | cTime | TIMESTAMP(4) | 非空 |  |  |
| 最后修改时间 | lmTime | TIMESTAMP(4) | 非空 |  |  |
| 说明 | descn | VARCHAR2(2000) | 可空 |  |  |
| **补充说明** | 1-业务编码不能为空，在新增时，若业务编码为空，则默认把业务编码设置为字典项的ID  2-关于dType的说明：  dType=1，系统保留，只能修改名称和排序；也不能扩充其结构；  dType=2，导入系统，只能修改名称和排序；可以向其下增加字典项，其下属字典项的dType是2或3；  dType=3，用户定义的字典项，可以修改和删除，并向其下加字典项，其下属字典项的dType必须3；  dType=4，引用自其他字典项，其下属字典项的dType是3或4；  3-修改了如下内容，要在本此平台整理中修改过来：  a)字典项名称，从name改为ddName  **注意：**  **1- level从0开始**  **2-由于本次是导入字典项，所以可能会是这样的情况：**  **·只有1层结点**  **·isValidate总是生效；**  **·别名为空，没有实际意义**  **·字典性类型，都选取为2/3这个在上面已经解释；**  **·业务编码：这个需要复杂的分析才能导入，目前和主键相同**  **3-创建者继承自主标plat\_DictM**  **4-若要给字典组加入图标，请参看文件的处理** | | | | | |
| **索引** | 1. id，主键 2. sortIdx(mId, pId, sort)，排序的索引 3. bizIdx001(mId, bCode)，此为隐含的唯一索引，同一字典组下面不能有相同的业务编码 | | | | | |

* 1. 用户类{4}
     1. U001用户[plat\_User]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **表名** | **plat\_User（用户）** | | | | |
| **意义** | **字段名** | **数据类型** | **空否** | **约束条件** | **说明** |
| 用户ID | id | VARCHAR2(32) | 非空 | 主键PK | UUID |
| 登录名 | loginName | VARCHAR2(15) | 非空 | 唯一索引 | 可以自动生成 |
| 密码 | password | VARCHAR2(30) | 非空 |  | 经过H5加密的 |
| 密码加盐 | salt | VARCHAR2(12) | 非空 |  | 12位UUID码 |
| 是否生效 | isValidate | NUMBER | 非空 | 默认值1有效 | 0-未激活；1有效；2无效； |
| 创建时间 | cTime | TIMESTAMP(4) | 非空 |  | 创建时的系统时间 |
| 最后修改 | lmTime | TIMESTAMP(4) | 非空 |  | 每次更新的时间 |
| **补充说明** | 此用户是用户信息的主要部分。 | | | | |
| **索引** | 1-id，主键  2- loginNameIdx(loginName)，保证登录名唯一 | | | | |

* + 1. U002用户扩展[plat\_UserExt]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **表名** | **plat\_UserExt（用户扩展信息）** | | | | | |
| **意义** | **字段名** | **数据类型** | **空否** | **约束条件** | **说明** |
| 用户ID | id | VARCHAR2(32) | 非空 | 外PK | 指向plat\_User表 |
| 实名 | realName | VARCHAR2(100) | 可空 |  |  |
| 用户主手机号 | mainPhoneNum | VARCHAR2(100) | 可空 |  | 经过H5加密的 |
| 公开字段信息 | pubFields | VARCHAR2(1000) | 可空 |  | 公开字段信息，字段间用逗号隔开 |
| 创建时间 | cTime | TIMESTAMP(4) | 非空 |  | 创建时的系统时间 |
| 最后修改 | lmTime | TIMESTAMP(4) | 非空 |  | 每次更新的时间 |
| **补充说明** | 用户的扩展信息  1-id既是本表的主键也是这个表的外键，指向plat\_User表；  2-本表还可以扩展，如生日，QQ号，星座等等。目前就是上面这些 | | | | | |
| **索引** | 1-id，主键  2-login\_name，IDX\_USER\_LNAME，保证登录名唯一 | | | | | |

* + 1. U003第三方登录[plat\_ThirdUser]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **表名** | **plat\_ThirdUser（用户第三方登录）** | | | | |
| **意义** | **字段名** | **数据类型** | **空否** | **约束条件** | **说明** |
| 第三方登录Id | id | VARCHAR2(32) | 非空 | 主键 | 指向plat\_User表 |
| 用户Id | userId | VARCHAR2(32) | 非空 |  | 在我们系统中的用户Id |
| 三方中用户标识 | thirdUserId | VARCHAR2(100) | 非空 |  | 第三方中的用户唯一标识 |
| 三方系统类型 | thirdSysType | VARCHAR2(5) | 非空 | 默认值10100 | 外部社交系统 |
| 三方系统代号 | thirdSysId | VARCHAR2(32) | 非空 |  | 微信，微博，QQ等 |
| 三方用户数据 | thirdUserInfo | text | 非空 |  | 第三方用户数据，以Json格式存储 |
| 三方用户登录次数 | thirdLoginCount | Integer | 非空 |  | 三方登录次数 |
| 创建时间 | cTime | TIMESTAMP(4) | 非空 |  | 创建时的系统时间 |
| 最后修改 | lmTime | TIMESTAMP(4) | 非空 |  | 每次更新的时间 |
| **补充说明** | 采用第三方登录，第三方登录信息记录在这个表中。  **1-登录次数：**说明用户用这个第三方账号已经登录了本系统几次了（注意，这里不关心是通过那个设备实现的第三方登录，如用QQ账号在App或在PC上登录过，那三方登录次数至少是2） | | | | |
| **索引** | 1-id，主键  2-login\_name，IDX\_USER\_LNAME，保证登录名唯一 | | | | |

* + 1. U004用户登录情况[plat\_UserLogin]

记录用户最近一次登录的情况，是客户端状态的一种保持，为实现App的自动登录。

注意，这个模型不支持在同一设备上，同一App登录两次

自动登录功能在PC上不能实现 ，因为目前无法实现和App一样的类IMEI的信息。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **表名** | **plat\_UserLogin（用户登录情况）** | | | | | |
| **意义** | **字段名** | **数据类型** | **空否** | **约束条件** | **说明** | |
| 本表Id | id | VARCHAR2(32) | 非空 | 主键 | 本表Id | |
| 应用分类 | sysType | VARCHAR2(5) | 非空 | 默认值10000 | 自己的系统 | 所登录的系统 |
| 应用标识 | sysId | VARCHAR2(32) | 非空 | 默认值灵派数据 | 系统标识 |
| 用户Id | userId | VARCHAR2(32) | 非空 |  | 用户的id | |
| 客户端类型 | deviceType | INTEGER(2) | 非空 |  | 设备分类：1=手机；2=设备；3=PC，默认1 | |
| 设备编号 | deviceId | VARCHAR2(32) | 非空 |  | 若是PC，则这个值是SessionId | |
| 登录状态 | status | text | 非空 |  | 状态：1-登录；2-注销；3-踢出 | |
| 过期时间 | expire | Long | 非空 |  | 过期时间，默认为7天 | |
| 最后修改 | lmTime | TIMESTAMP(4) | 非空 |  | 每次更新的时间 | |
| **补充说明** | 1-这个表中status=2实际是不存在的，用户注销后，这条记录就被删除了；  2-踢出状态是为有消息处理的框架准备的；  3-任何有效访问都会更新lmTime；  4-要有一个后台线程，清理这个表中的过期数据；清理逻辑为curTime-lmTime>expire，单位是毫秒  5-sysType,sysId,userId，这里了一个关系，即那个系统的用户，这个关系目前还没有在单独的关系表中记录。 | | | | | |
| **索引** | 1-id，主键  2-bizIdx001(sysType, userId,pcdType,deviceId) ，唯一索引  3-userIdIdx(sysType, userId) ，便于找到同一用户的登录情况  4-bizIdx002(sysType, userId,pcdType) ，便于找到用户在同类设备上的登录情况 | | | | | |

* 1. 访问日志{1}

用户以怎样的方式，访问了那些系统。

* + 1. L001访问者访问日志记录 [log\_Visit]

谁什么时间，在什么地点，以什么设备，通过什么方式访问了系统中的那个对象。基本上客户分为如下几类信息：

1、访问者信息：主要是id

2、从那个系统的什么功能访问的：fromSys，fromModel

3、访问那个系统的什么功能：visitSys,vistModel

4、访问的具体接口功能：url，主要是http协议；方法get/post/……；访问参数param

5、接口功能功能的执行情况：开始时间技术时间，返回值等

6、用户访问是通过什么样的设备访问的

7、用户访问时的地点是什么

**其中有些信息在某些情况下，会无法收集，这个情况碰到具体了再处理或扩展。**

**若这套日志方式每个系统自己维护自己的，则fromSys和visitSys无用。**

**希望通过这个表的设计，能够记录：**

**A)数据访问接口的日志；**

**B)网站访问的日志**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **表名** | | | **log\_Visit（访问日志）** | | | | |  |  |
| **意义** | | | **字段名** | **数据类型** | **空否** | **约束** | **说明** | **数据访问** | **页面收集** |
| 访问Id | | | Id | VARCHAR2(32) | 非空 | 主键PK | UUID | 需要 | 需要 |
| 服务系统 | 应用类型 | | servSysType | VARCHAR2(5) | 非空 |  | 处理请求的系统，默认自己访问自己10000::灵派数据。  在数据交换中，此值应该是系统表中的数据 | 外围系统 | 外围系统 |
| 应用标识 | | servSysId | VARCHAR2(32) | 非空 |  | 外围系统 | 外围系统 |
| 访问类型 | | | visitType | Integer(1) | 非空 |  | 访问类型：1=页面访问html；2=数据访问；3=数据记录 | 需要 | 需要 |
| 客户端  系统 | 系统类型 | | visitSysType | VARCHAR2(5) | 可空 |  | 那个系统访问的  这里默认的是自己的系统：10000::灵派数据 | 本系统 | 本系统或空 |
| 系统标识 | | visitSysId | VARCHAR2(32) | 可空 |  | 本系统 | 本系统或空 |
| 用何模块 | | visitModuleId | VARCHAR2(400) | 可空 |  | 这个一般无用，可以是访问的具体接口的分类 | 模块名称表示 | 模块名称表示 |
| 访问者 | 访问者类型 | | visitorType | VARCHAR2(5) | 非空 | 默认值  10000 | 20000=平台；  20011=无状态App前端；  20012=无状态Pc前端； | 需要 | 需要 |
| 访问者ID | | visitorId | VARCHAR2(32) | 非空 | 默认值  灵派数据 | 当vT=20011：IMEI；  当vT=20012：SessionId；  当vT=20000：用户Id； | 需要 | 需要 |
| 客户端  信息 | IP | | clientIp | VARCHAR2(40) | 可空 |  | 客户端Ip，可接受IPv6 | 若App应该可以收集 | 无法收集 |
| 地理位置信息 | | pointInfo | VARCHAR2(500) | 可空 |  | GPS坐标，Json格式。  通常通过App手机，PC无法收集 | 需要 | 需要 |
| 设备  信息 | 分类 | deviceType | INTEGER(1) | 非空 |  | 0=服务器1=手机2=设备3=PC | 需要 | 需要 |
| 名称 | deviceId | VARCHAR2(32) | 可空 |  | 设备编号：IMEI或PC标识（mac地址或CPU号） | 需要 | 需要 |
| 型号 | deviceClass | VARCHAR2(100) | 可空 |  | 设备型号：手机型号+品牌，如：HUAWEI, hc-10 | 需要 | 需要 |
| 屏幕 | screenSize | VARCHAR2(100) | 可空 |  | 屏幕大小 | 需要 | 需要 |
| 浏览  器 | 名称 | exploreName | VARCHAR2(100) | 可空 |  | 若是App，则无浏览器，则此字段为空。 | App:无需 | 需要 |
| 版本 | exploreVer | VARCHAR2(100) | 可空 |  | App:无需 | 需要 |
| 操作  系统 | 名称 | OSName | VARCHAR2(100) | 可空 |  | 操作系统：1-对APP：Android|IOS版本；2-对PC:是PC操作系统版本 | 需要 | 若能收集 |
| 版本 | OSVer | VARCHAR2(100) | 可空 |  | 需要 | 若能收集 |
| 访问参数 | 访问名称 | | apiName | VARCHAR2(100) | 非空 |  | 当是数据访问：访问Api名称，是服务器端的信息  当是html收集：是业务Title，是客户端信息  若是动作信息：则是客户端信息，是动作的表示，如播放 | 需要：Api名称 | 页面Title |
| 导入URL | | fromUrl | VARCHAR2(400) | 可控 |  | 从那个URL导入的 | 无需 | 需要 |
| 访问URL | | reqUrl | VARCHAR2(400) | 非空 |  | 访问URL：可以是请求数据的URL，也可以直接是html | 需要 | 需要 |
| HTTP方法 | | method | VARCHAR2(10) | 非空 |  | http协议方法：get/post/del/update；对visitType=1 | 需要 | 需要 |
| 请求参数 | | reqParam | Text | 非空 |  | 访问参数，若参数为空，这里是空串 | 需要 | 需要 |
| 访问对象类型 | | objType | VARCHAR2(100) | 非空 |  | 是数据访问中，要访问的数据的类型，可能就是表名 | 需要 | 页面分类 |
| 访问对象Id | | objId | VARCHAR2(32) | 非空 |  | 数据的ID | 可能需要 | 无需 |
| 处理情况 | 开始时间 | | beginTime | Long | 非空 |  | 接口开始时间，或访问时间 | 需要：开始时间 | 需要：访问时间 |
| 结束时间 | | endTime | Long | 非空 |  | 接口执行完时间 | 需要 | 无需 |
| 处理状态 | | dealFlag | integer(1) | 非空 |  | 1成功2失败 | 需要 | 都是成功 |
| **补充说明** | | | 1-这个表由于塞入了很多字段，所以是稀疏的  2-今后看看是否需要把这个表拆开  3-在某些情况下，fromSys和visitSys是同一个系统，如我们自己的系统；  4-接口的返回值不显示存储在这个表中，返回值可能很大，若需要会存储在硬盘文件中 | | | | | | |
| **索引** | | | 1-id，主键 | | | | | | |

1. 数据逻辑视图

说明：这里只是第一阶段功能主要部分，关于语义部分，应该还需要设计一些存储结构。