SJTU-JJ数据库设计说明

1. 数据库设计采用Maria关系型数据库和Mongo非关系型数据库的结合。其中Maria关系型数据库中存储结构化的数据，例如用户、信息发布、商品、标签、交易、申诉等等，包含了主要的全部数据存储；Mongo非关系型数据库中存储非结构化数据，包括三个部分：聊天记录、评论、图片。
2. Maria关系型数据库

如图所示：



图1. CDM



图2. PDM

SQL建表语句见附录1。

1. Mongo非关系型数据库

* 聊天记录：MariaDB中负责存储了两个用户之间存在聊天记录，但真实的记录存储在MongoDB中，从MariaDB中找到chat\_log表中的mongo\_chat\_id，从而在mongoDB中检索实际的聊天记录信息
* 评论：MariaDB中只存储了一个用户对一次交易的一级评论，其他基于此一级评论下的所有评论信息都存储在MongoDB中，同样使用comment表中的mongo\_comment\_id检索其他的所有评论。
* 图片：MariaDB中存储的所有图片都是他们的url，包括用户头像，评论图片，商品的图片等等，实际的图片信息存储在MongoDB，通过MariaDB拿到图片url之后访问MongoDB获取图片。

## 附录一

/\*==============================================================\*/

/\* DBMS name: MySQL 5.0 \*/

/\* Created on: 2019/7/2 10:57:32 \*/

/\*==============================================================\*/

drop table if exists Good;

drop table if exists Relationship\_5;

drop table if exists chat\_log;

drop table if exists comment;

drop table if exists comment\_picture;

drop table if exists complaint;

drop table if exists evaluate;

drop table if exists good\_picture;

drop table if exists "release";

drop table if exists tag;

drop table if exists transaction;

drop table if exists user;

/\*==============================================================\*/

/\* Table: Good \*/

/\*==============================================================\*/

create table Good

(

release\_id int not null,

good\_name varchar(64) not null,

good\_description varchar(1024) not null,

primary key (release\_id)

);

/\*==============================================================\*/

/\* Table: Relationship\_5 \*/

/\*==============================================================\*/

create table Relationship\_5

(

release\_id int not null,

tag\_id int not null,

primary key (release\_id, tag\_id)

);

/\*==============================================================\*/

/\* Table: chat\_log \*/

/\*==============================================================\*/

create table chat\_log

(

chat\_log\_id int not null,

user\_id int not null,

use\_user\_id int not null,

mongo\_chat\_id varchar(1024) not null,

primary key (chat\_log\_id)

);

/\*==============================================================\*/

/\* Table: comment \*/

/\*==============================================================\*/

create table comment

(

comment\_id varchar(1024) not null,

user\_id int not null,

release\_id int not null,

content varchar(1024) not null,

mongo\_comment\_id varchar(1024),

primary key (comment\_id)

);

/\*==============================================================\*/

/\* Table: comment\_picture \*/

/\*==============================================================\*/

create table comment\_picture

(

comment\_picture\_id int not null,

comment\_id varchar(1024) not null,

comment\_picture\_url varchar(1024) not null,

primary key (comment\_picture\_id)

);

/\*==============================================================\*/

/\* Table: complaint \*/

/\*==============================================================\*/

create table complaint

(

complaint\_id int not null,

user\_id int,

use\_user\_id int,

content varchar(1024) not null,

primary key (complaint\_id)

);

/\*==============================================================\*/

/\* Table: evaluate \*/

/\*==============================================================\*/

create table evaluate

(

evaluate\_id int not null,

user\_id int not null,

score int not null,

primary key (evaluate\_id)

);

/\*==============================================================\*/

/\* Table: good\_picture \*/

/\*==============================================================\*/

create table good\_picture

(

good\_picture\_id int not null,

release\_id int,

good\_picture\_url varchar(1024) not null,

primary key (good\_picture\_id)

);

/\*==============================================================\*/

/\* Table: "release" \*/

/\*==============================================================\*/

create table "release"

(

release\_id int not null,

user\_id int not null,

Goo\_release\_id int,

tra\_release\_id int,

is\_saled bool not null,

release\_time datetime not null,

valid\_time datetime not null,

is\_appointed bool not null,

is\_finished bool not null,

primary key (release\_id)

);

/\*==============================================================\*/

/\* Table: tag \*/

/\*==============================================================\*/

create table tag

(

tag\_id int not null,

tag\_name varchar(16) not null,

primary key (tag\_id)

);

/\*==============================================================\*/

/\* Table: transaction \*/

/\*==============================================================\*/

create table transaction

(

release\_id int not null,

user\_id int not null,

create\_time datetime not null,

primary key (release\_id)

);

/\*==============================================================\*/

/\* Table: user \*/

/\*==============================================================\*/

create table user

(

user\_id int not null,

user\_name varchar(16) not null,

password varchar(16) not null,

portrait\_url varchar(1024) not null,

telephone char(11) not null,

primary key (user\_id)

);

alter table Good add constraint FK\_Relationship\_6 foreign key (release\_id)

references "release" (release\_id) on delete restrict on update restrict;

alter table Relationship\_5 add constraint FK\_Relationship\_7 foreign key (release\_id)

references Good (release\_id) on delete restrict on update restrict;

alter table Relationship\_5 add constraint FK\_Relationship\_8 foreign key (tag\_id)

references tag (tag\_id) on delete restrict on update restrict;

alter table chat\_log add constraint FK\_Relationship\_14 foreign key (user\_id)

references user (user\_id) on delete restrict on update restrict;

alter table chat\_log add constraint FK\_Relationship\_15 foreign key (use\_user\_id)

references user (user\_id) on delete restrict on update restrict;

alter table comment add constraint FK\_Relationship\_12 foreign key (user\_id)

references user (user\_id) on delete restrict on update restrict;

alter table comment add constraint FK\_Relationship\_13 foreign key (release\_id)

references transaction (release\_id) on delete restrict on update restrict;

alter table comment\_picture add constraint FK\_Relationship\_11 foreign key (comment\_id)

references comment (comment\_id) on delete restrict on update restrict;

alter table complaint add constraint FK\_Relationship\_17 foreign key (user\_id)

references user (user\_id) on delete restrict on update restrict;

alter table complaint add constraint FK\_Relationship\_18 foreign key (use\_user\_id)

references user (user\_id) on delete restrict on update restrict;

alter table evaluate add constraint FK\_Relationship\_16 foreign key (user\_id)

references user (user\_id) on delete restrict on update restrict;

alter table good\_picture add constraint FK\_Relationship\_10 foreign key (release\_id)

references Good (release\_id) on delete restrict on update restrict;

alter table "release" add constraint FK\_Relationship\_4 foreign key (tra\_release\_id)

references transaction (release\_id) on delete restrict on update restrict;

alter table "release" add constraint FK\_Relationship\_5 foreign key (Goo\_release\_id)

references Good (release\_id) on delete restrict on update restrict;

alter table "release" add constraint FK\_Relationship\_9 foreign key (user\_id)

references user (user\_id) on delete restrict on update restrict;

alter table transaction add constraint FK\_Relationship\_1 foreign key (user\_id)

references user (user\_id) on delete restrict on update restrict;

alter table transaction add constraint FK\_Relationship\_3 foreign key (release\_id)

references "release" (release\_id) on delete restrict on update restrict;