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，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，包括聊天记录、评论、多媒体信息。
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中负责存储了产生聊天记录的两个用户，但真实的记录存储在MongoDB中，从MariaDB中找到chat\_log表中的chat\_content\_id，从而在mongoDB中检索相关的聊天记录信息
* 多媒体信息：MariaDB中存储的content\_id，通过content\_id访问MongoDB中的多媒体信息，包括图片视频等多种类型，使用GridFS分片存储

## 附录一

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

/\* DBMS name: MySQL 5.0 \*/

/\* Created on: 2019/9/9 19:28:08 \*/

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

drop table if exists Good;

drop table if exists Relationship\_5;

drop table if exists chat\_log;

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

(

good\_id int not null,

good\_name varchar(64) not null,

price decimal not null,

description text,

content\_id char(24) not null,

primary key (good\_id)

);

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

/\* Table: Relationship\_5 \*/

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

create table Relationship\_5

(

good\_id int not null,

tag\_id int not null,

primary key (good\_id, tag\_id)

);

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

/\* Table: chat\_log \*/

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

create table chat\_log

(

chat\_log\_id int not null,

user\_id2 int not null,

use\_user\_id2 int not null,

chat\_content\_id char(24) not null,

primary key (chat\_log\_id)

);

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

/\* Table: "release" \*/

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

create table "release"

(

release\_id2 int not null,

user\_id2 int not null,

good\_id int not null,

transaction\_id int,

release\_time datetime not null,

valid\_time datetime not null,

state int not null,

primary key (release\_id2)

);

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

/\* 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

(

transaction\_id int not null,

user\_id2 int not null,

create\_time datetime not null,

primary key (transaction\_id)

);

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

/\* Table: user \*/

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

create table user

(

user\_id2 int not null,

user\_name varchar(16) not null,

avatar\_id char(24) not null,

telephone char(11),

student\_id varchar(32) not null,

student\_name varchar(32) not null,

primary key (user\_id2),

key AK\_Identifier\_2 (student\_id)

);

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

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

alter table Relationship\_5 add constraint FK\_Relationship\_6 foreign key (tag\_id)

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

alter table chat\_log add constraint FK\_Relationship\_8 foreign key (user\_id2)

references user (user\_id2) on delete restrict on update restrict;

alter table chat\_log add constraint FK\_Relationship\_9 foreign key (use\_user\_id2)

references user (user\_id2) on delete restrict on update restrict;

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

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

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

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

alter table "release" add constraint FK\_Relationship\_7 foreign key (user\_id2)

references user (user\_id2) on delete restrict on update restrict;

alter table transaction add constraint FK\_Relationship\_1 foreign key (user\_id2)

references user (user\_id2) on delete restrict on update restrict;