

准备工作:建表及插入数据;

创建用户表:user

```
CREATE TABLE hw_user
(
    uid INT PRIMARY KEY AUTO_INCREMENT,
    uname VARCHAR(50) NOT NULL,
    age INT,
    loc CHAR(50)
);
```

uid	uname	age	loc

插入数据:

```
INSERT INTO hw_user VALUES (null,'李小龙',51,'北京');
INSERT INTO hw_user VALUES (null,'张家辉',52,'上海');
INSERT INTO hw_user VALUES (null,'周润发',62,'广州');
INSERT INTO hw_user VALUES (null,'梁家辉',63,'深圳');
INSERT INTO hw_user VALUES (null,'刘青云',48,'沈阳');
INSERT INTO hw_user VALUES (null,'古天乐',39,'杭州');
INSERT INTO hw_user VALUES (null,'张丰毅',28,'杭州');
INSERT INTO hw_user VALUES (null,'梁朝伟',46,'北京');
INSERT INTO hw_user VALUES (null,'陈道明',58,'北京');
INSERT INTO hw_user VALUES (null,'张国荣',47,'香港');
INSERT INTO hw_user VALUES (null,'黄渤',27,'北京');
INSERT INTO hw_user VALUES (null,'吴镇宇',38,'香港');
INSERT INTO hw_user VALUES (null,'周星驰',38,'香港');
```

创建书类型表:type

```
CREATE TABLE type
(
    tid INT PRIMARY KEY AUTO_INCREMENT,
    tname VARCHAR(50) NOT NULL
);
```

tid	tname

插入数据:

```
INSERT INTO type VALUES (NULL,'言情');
INSERT INTO type VALUES (NULL,'恐怖');
INSERT INTO type VALUES (NULL,'玄幻');
INSERT INTO type VALUES (NULL,'穿越');
INSERT INTO type VALUES (NULL,'异界');
```

```

INSERT INTO type VALUES (NULL,'校园');
INSERT INTO type VALUES (NULL,'仙侠');
INSERT INTO type VALUES (NULL,'武侠');
INSERT INTO type VALUES (NULL,'网游');
INSERT INTO type VALUES (NULL,'军事');

```

创建图书表:book

```

CREATE TABLE book
(
    bkid INT PRIMARY KEY AUTO_INCREMENT,
    bkname VARCHAR(50) NOT NULL,
    author VARCHAR(50) NOT NULL,
    tid INT NOT NULL,
    CONSTRAINT book_fk_type FOREIGN KEY (tid) REFERENCES type (tid)
);

```

bkid	bkname	author	tid

插入数据:

```

INSERT INTO book VALUES (NULL,'完美世界','辰东',3);
INSERT INTO book VALUES (NULL,'斗破苍穹','天蚕土豆',3);
INSERT INTO book VALUES (NULL,'校花的贴身高手','鱼人二代',1);
INSERT INTO book VALUES (NULL,'盗墓笔记','南派三叔',2);
INSERT INTO book VALUES (NULL,'凡人修仙传','忘语',7);
INSERT INTO book VALUES (NULL,'诛仙','萧鼎',7);
INSERT INTO book VALUES (NULL,'将夜','猫腻',3);
INSERT INTO book VALUES (NULL,'步步惊心','桐华',4);
INSERT INTO book VALUES (NULL,'琅琊榜','海宴',4);
INSERT INTO book VALUES (NULL,'异界之屠龙传奇','吴三胖子',5);
INSERT INTO book VALUES (NULL,'天龙八部','金庸',8);
INSERT INTO book VALUES (NULL,'狼牙','刘猛',10);
INSERT INTO book VALUES (NULL,'从零开始','雷云风暴',9);
INSERT INTO book VALUES (NULL,'飞升之后','黄埔奇',3);

```

```

创建评论表:comment
create table comment
(
    comid int not null auto_increment
        primary key,
    content text not null,
    bkid int null,
    uid int null,
    datetime datetime not null,
    constraint comment_fk_bkid
        foreign key (bkid) references book (bkid),
    constraint comment_fk_uid
        foreign key (uid) references hw_user (uid)
);

```

comid	content	bkid	uid	datetime

插入数据:

```

INSERT INTO comment VALUES (NULL,'这完美世界本书写的太好了.',1,3,'2000-01-01 19:00:00');
INSERT INTO comment VALUES (NULL,'天蚕土豆我要给你生土豆.',2,11,'2011-06-04 21:03:10');
INSERT INTO comment VALUES (NULL,'天蚕土豆我要给你生土豆+1.',2,11,'2011-06-04 21:03:11');
INSERT INTO comment VALUES (NULL,'天蚕土豆我要给你生土豆+2.',2,11,'2011-06-04 21:03:12');
INSERT INTO comment VALUES (NULL,'天蚕土豆我要给你生土豆+3.',2,11,'2011-06-04 21:03:13');
INSERT INTO comment VALUES (NULL,'你拿什么给他生?.',2,13,'2011-06-04 22:05:23');
INSERT INTO comment VALUES (NULL,'强推飞升之后!.',14,3,'2013-09-14 02:15:33');
INSERT INTO comment VALUES (NULL,'这是我看过的印象很深刻的军事小说!.',12,8,'2016-03-14 12:25:03');
INSERT INTO comment VALUES (NULL,'写的不错,在追!.',7,7,'2016-03-14 12:25:03');
INSERT INTO comment VALUES (NULL,'确实写的不错,我也在追!.',7,6,'2016-03-14 12:26:13');
INSERT INTO comment VALUES (NULL,'哎这风云无忌怎么这么傻!.',14,9,'2013-03-17 14:36:19');
INSERT INTO comment VALUES (NULL,'其实我还是比较喜欢迟伤的!.',14,10,'2018-05-22 00:36:00');
INSERT INTO comment VALUES (NULL,'迟伤三扣谢恩师的时候都快给我看哭了!.',14,5,'2012-12-22 03:46:03');
INSERT INTO comment VALUES (NULL,'还生土豆,小白文一个!.',2,5,'2011-07-22 13:41:13');
INSERT INTO comment VALUES (NULL,'萧炎个烧饼!.',2,5,'2011-07-22 13:42:30');
INSERT INTO comment VALUES (NULL,'不亏是金老师,经典就是经典!.',11,5,'2009-12-12 20:12:31');
INSERT INTO comment VALUES (NULL,'诛仙真是经典啊!.',6,2,'2012-12-12 12:12:12');

```

创建收藏表:u_coll_bk

```
create table u_coll_bkid
```

```
(
    uid int null,
    bkid int null,
    constraint u_coll_bkid_fk_uid
        foreign key (uid) references hw_user (uid),
    constraint u_coll_bkid_fk_bkid
        foreign key (bkid) references book (bkid),
    constraint uk_ucb unique(uid,bkid)
);
```

uid	bkid

插入数据:

```
INSERT INTO u_coll_bkid VALUES (1,2);
INSERT INTO u_coll_bkid VALUES (1,5);
INSERT INTO u_coll_bkid VALUES (1,9);
INSERT INTO u_coll_bkid VALUES (2,6);
INSERT INTO u_coll_bkid VALUES (2,9);
INSERT INTO u_coll_bkid VALUES (2,14);
INSERT INTO u_coll_bkid VALUES (3,1);
INSERT INTO u_coll_bkid VALUES (4,5);
INSERT INTO u_coll_bkid VALUES (5,14);
INSERT INTO u_coll_bkid VALUES (5,2);
INSERT INTO u_coll_bkid VALUES (5,3);
INSERT INTO u_coll_bkid VALUES (6,7);
INSERT INTO u_coll_bkid VALUES (7,7);
INSERT INTO u_coll_bkid VALUES (9,14);
INSERT INTO u_coll_bkid VALUES (13,14);
INSERT INTO u_coll_bkid VALUES (2,11);
INSERT INTO u_coll_bkid VALUES (2,13);
INSERT INTO u_coll_bkid VALUES (4,14);
INSERT INTO u_coll_bkid VALUES (7,1);
INSERT INTO u_coll_bkid VALUES (7,2);
INSERT INTO u_coll_bkid VALUES (7,3);
INSERT INTO u_coll_bkid VALUES (7,5);
INSERT INTO u_coll_bkid VALUES (7,9);
INSERT INTO u_coll_bkid VALUES (7,11);
INSERT INTO u_coll_bkid VALUES (7,14);
```

题目:

1,查询收藏了两本书以上的用户的姓名与居住的城市.

答案:

李小龙 北京

张家辉 上海

刘青云 沈阳

张丰毅 杭州

2,查询年龄在三十到六十之间的用户的姓名,评论过的书籍名称

答案:

张家辉 诛仙
刘青云 飞升之后
刘青云 斗破苍穹
刘青云 天龙八部
古天乐 将夜
梁朝伟 狼牙
陈道明 飞升之后
张国荣 飞升之后
周星驰 斗破苍穹

3,每个用户在2016年之前的评论数量

答案:

张家辉 1
周润发 2
刘青云 4
陈道明 1
黄渤 4
周星驰 1

4,查询刘青云评论过的书籍id及评论的次数

答案:

书籍id	评论次数
2	2
11	1
14	1

5,查询收藏了书籍飞升之后的用户且年龄大于用户平均年龄5岁的用户的详细信息.

答案:

9	陈道明	58	北京
3	周润发	62	广州

作业目标及能力提升

练习多表查询,连接查询,子查询,锻炼逻辑能力.

思路:

一:查询收藏了两本书以上的用户的姓名与居住的城市.

```
select 用户的姓名:hw_user.username 居住的城市:hw_user.loc
from 因为有一个条件是收藏了两本书以上,所以一定会查一下收藏表:u_coll_bk
      自然还有用户表:hw_user
where 在查询收藏表时,有条件:收藏了两本书以上==>说明uid出现了两次
group by 在确定用户收藏数量时,需要根据uid分组
```

- (1)因为要判断uid出现次数,所以在查询收藏表时,根据u_coll_bk.uid分组,并查出分组后的各组数量
- (2)查出各组数量后,取出数量大于2的数据.
- (3)前两步的结果会生成一个结果集,起一个别名:result_u_coll_bk,这个结果集中就包含了收藏了两本书以上的用户的uid.
- (4)从result_u_coll_bk结果集中查出uid属性,这时候就又生成了一个结果集.
- (5)查询hw_user表的username,loc两列,条件是hw_user.uid的属性出现在第四步生成的结果集里了.

二:查询年龄在三十到六十之间的用户的姓名,评论过的书籍名称

```
select 用户姓名:hw_user.username 评论过的书籍名称:book.bkname
from 用户表 评论表 书籍表
where 用户表:age between 30 and 60
      要查询该用户评论过的书籍,所以要去评论表中查询符合该用户uid的书籍的bkid:
      hw_user.uid=comment.uid
      再根据得到的书籍的bkid去书籍表中查询名称
```

- (1)先查出符合年龄条件的结果集,查列uid,username两列(uid用于后面的查询中,username最后用来显示)起别名为:result_age
- (2)再从结果集result_age中查询出uid,这时候生成的结果集可以看成是一个集合(多行单列)
- (3) 查询评论表的bkid列和uid列,条件是uid的值存在第(2)步生成的结果集中.(用关键字IN),由因为评论中会有同一个用户评论多次同一个书籍的情况,所以要用distinct关键字去重.
- (4) 这时候就基本查出了每个符合条件的数据,但是还需要找到要求显示的内容(现在只找到了各种id,还需要根据id找名字和书名)
- (5)给第(3)步查出的结果集起一个别名rbu
- (6)使用连接查询查列:hw_user.username,book.bkname两列,查询表rbu,book,hw_user
- (7)rbu inner join book on 条件为bkid相同
- (8) inner join hw_user 条件为uid相同

三:每个用户的姓名,及其在2016年之前的评论数量

```
select username,要评论数量,所以需要在查询comment表时根据用户分组,分组后取和
from hw_user ,comment
where 在查评论时,datetime>'2016-01-01 00:00:00'
group uid
```

- (1)查询表comment满足时间条件的数据,并根据uid分组,查出uid列和分组后每组的数量(别名cc)

(2)给第(1)步生成的结果集起一个别名, rdt

(3)使用连接查询,查询hw_user.uname 和rdt.cc两列,查询rdt和hw_user两个表,条件是hw_user.uid=rdt.uid;

四,查询刘青云评论过的书籍id及评论的次数

```
select 书籍id,评论的次数(需要根据书籍id分组)
from hw_user(需要确定刘青云的uid),comment
where uname='刘青云'得到uid后,去comment表中查询comment.uid=刘青云的uid
group bkid
```

(1)在表hw_user中查询刘青云的uid,会生成一个单行单列的结果集,这个结果集就可以直接用于where判断条件

(2)在表comment中查询符合刘青云uid的数据,并且按照bkid分组,查询出bkid,和分组后的每个bkid组的数量(聚合函数count(bkid) 别名:评论次数)

五,查询收藏了书籍飞升之后的用户且年龄大于用户平均年龄5岁的用户的详细信息.

```
select 查询用户的详细信息(*);
from hw_user
where 条件有两个,
      收藏了飞升之后
      and
      年龄大于平均年龄
```

(1)首先需要获得飞升之后的书籍id

(2)通过该书籍id,去u_coll_bk收藏表中,查询出收藏了该书籍的用户uid

(3)第(2)步会生成一个结果集(单列多行,可看成集合),所以可以使用关键字IN

(4)查询用户中uid与第(2)步的结果集匹配的数据,使用关键字IN

(5)在第(4)步的条件后,加上and 查询年龄大于平均年龄的数据

(6)查询表hw_user 列sum(age)/count(*)即可查出平均年龄,为一个单行单列的结果集,可直接用于where条件

(7) and age > 第(6)步生成的结果集+5