

“webclass”数据库设计

目录

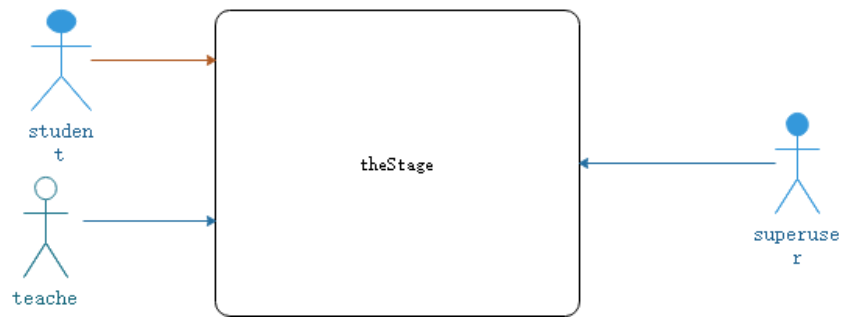
“webclass”数据库设计	1
1 UML 图	3
1.1 边界图	3
1.2 用例图	4
2 ER 图	5
2.1 图形	5
2.2 分析	5
3 逻辑结构设计	6
3.1 主要逻辑关系	6
4 数据库定义	7
4.1 birthplace	7
4.2 document	8
4.3 homework	9
4.4 question	9
4.5 score	10
4.6 student	11
4.7 subject	11
4.8 task	12
4.9 teacher	13

5 数据操作.....	13
5.1 排序	13
5.2 限制	14
5.3 聚合	14
5.4 表的连接	14
5.5 子查询.....	15
5.6 记录联合	15

1 UML 图

1.1 边界图

1.1.1 图形



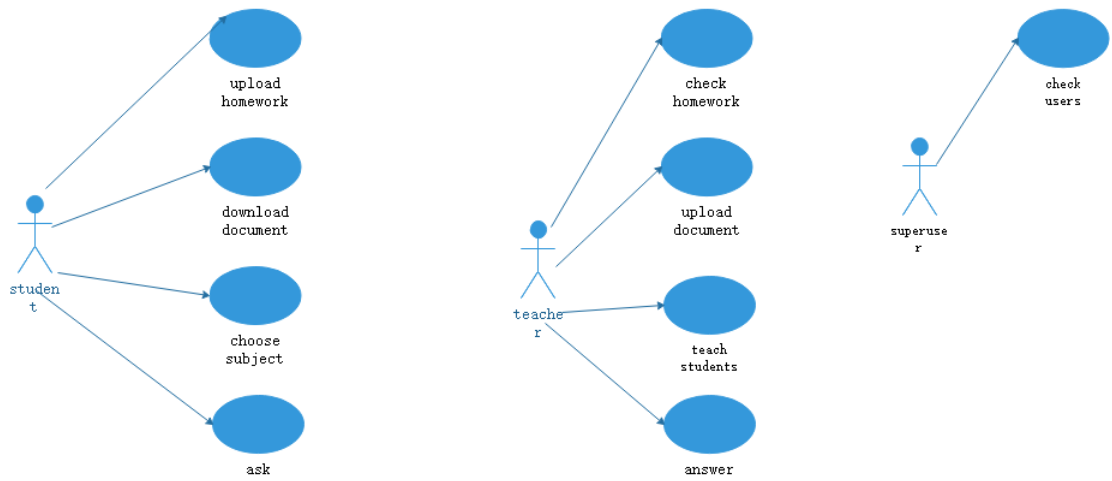
(图一)

1.1.2 分析

由题目需求可知，如图一所示，此数据库的可能访问者共有老师（teacher）、学生（student）以及管理员（superuser），分析此数据库，管理员的权限和教师的权限其实无很大区别，故在以后的数据库设计中可以将二者定义为一个实体，增加一个属性“isSuperuser”进行权限的判别。

1.2 用例图

1.2.1 图形



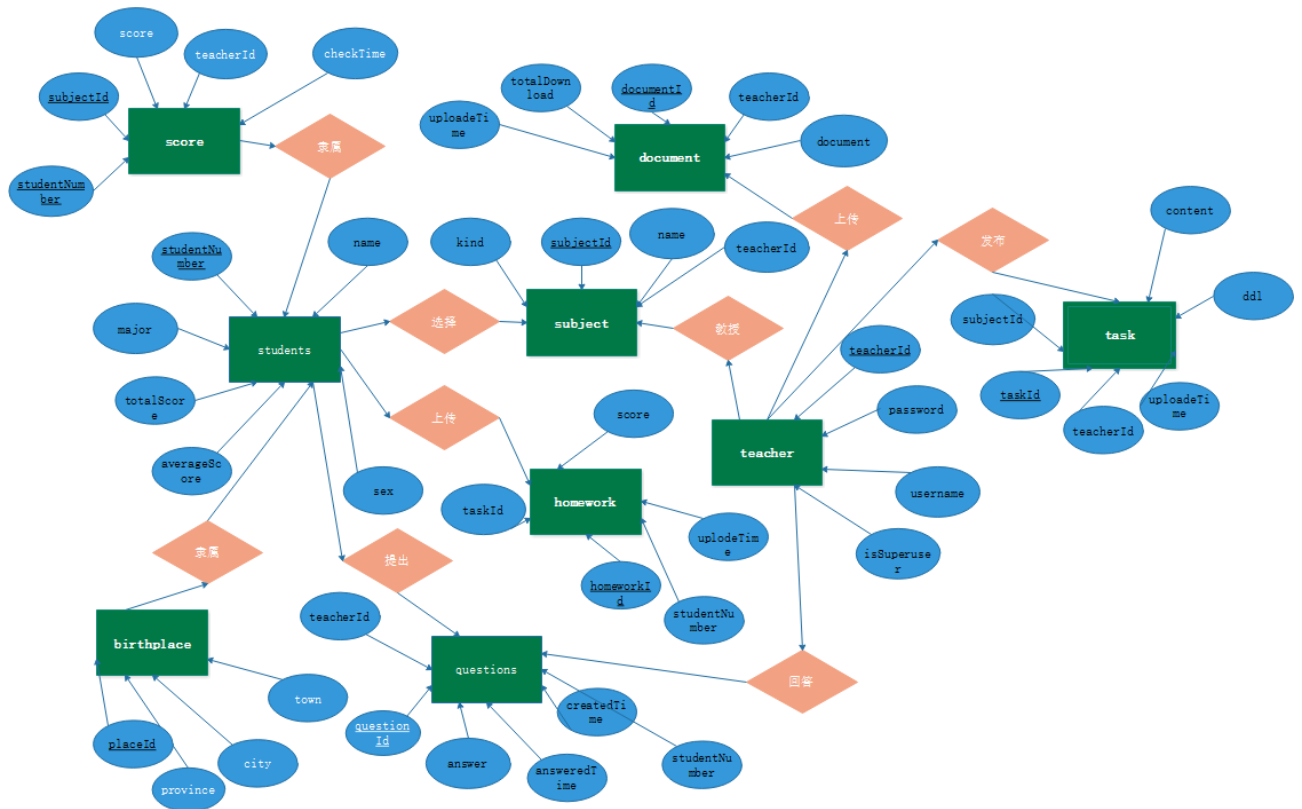
(图二)

1.2.2 分析

分析上述用例，可以初步发现实体共有 students、subjects、homework、teacher、question 和 task，具体见图二。

2 ER 图

2.1 图形



2.2 分析

随着对数据库的不断分析，对实体间的关系的不断改善，以及按照一对多、多对多关系和范式要求的不断完善，最终设计出来上述 ER 图，其中比初步分析多出来的实体有 birthplace、score 和 document。各实体间的关系以及各实体的属性、主键如图三所示

3 逻辑结构设计

3.1 主要逻辑关系

3.1 student and score (1:n):

student(studentNumber, major, totalScore, averageScore, name, sex)

score(studentNumber, subjectId, score, teacher, checkTime)

3.2 student and subject (m:n):

student(studentNumber, major, totalScore, averageScore, name, sex)

subject(kind, subjectId, name, teacherId)

student_choose_subject(studentNumber, subjectId)

3.3 student and homework (1:n):

student(studentNumber, major, totalScore, averageScore, name, sex)

homework(score, uploadeTime, studentNumber, homeworkId, taskId)

3.4 student and birthplace (m:n):

student(studentNumber, major, totalScore, averageScore, name, sex)

birthplace(province, city, town, placeId)

student_in_birthplace(studentNumber, placeId)

3.5 student and question (1:n):

student(studentNumber, major, totalScore, averageScore, name, sex)

question(questionId, answer, answerId, answerTime, createTime, studentNumber)

3.6 subject and teacher (1:n):

subject(kind, subjectId, name, teacherId)

teacher(teacherId, password, username, isSuperuser)

3.7 teacher and document (1:n):

teacher(teacherId, password, username, isSuperuser)

document(uploadTime, totalDownload, documentId, teacherId, document)

3.8. teacher and question (1:n):

teacher(teacherId, password, username, isSuperuser)

question(questionId, answer, answerId, answerTime, createTime, studentNumber)

3.9. teacher and task (1:n):

teacher(teacherId, password, username, isSuperuser)

task(subjectId, taskId, teacherId, uploadTime, ddl, content)

4 数据库定义

4.1 birthplace

4.1.1 创建

```
-----+
| birthplace | CREATE TABLE `birthplace` (
|   `placeId` int(11) NOT NULL,
|   `province` varchar(10) CHARACTER SET utf8 NOT NULL,
|   `city` varchar(10) CHARACTER SET utf8 NOT NULL,
|   `town` varchar(10) CHARACTER SET utf8 NOT NULL,
|   PRIMARY KEY (`placeId`)
| ) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+-----+
-----+
-----+
-----+
```

4.1.2 描述

Field	Type	Null	Key	Default	Extra
placeId	int(11)	NO	PRI	NULL	
province	varchar(10)	NO		NULL	
city	varchar(10)	NO		NULL	
town	varchar(10)	NO		NULL	

4.2 document

4.2.1 创建

```
+-----+
| document | CREATE TABLE `document` (
|   `uploadTime` datetime NOT NULL,
|   `totalDownload` int(11) NOT NULL,
|   `documentId` int(11) NOT NULL,
|   `teacherId` int(11) NOT NULL,
|   `document` varchar(100) CHARACTER SET utf8 NOT NULL COMMENT 'url',
|   PRIMARY KEY (`documentId`),
|   KEY `teacherId` (`teacherId`)
| ) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+-----+
```

4.2.2 描述

Field	Type	Null	Key	Default	Extra
uploadTime	datetime	NO		NULL	
totalDownload	int(11)	NO		NULL	
documentId	int(11)	NO	PRI	NULL	
teacherId	int(11)	NO	MUL	NULL	
document	varchar(100)	NO		NULL	

4.3 homework

4.3.1 创建

```
-----+
| homework | CREATE TABLE `homework` (
|   `score` float NOT NULL,
|   `taskId` int(11) NOT NULL,
|   `uploadTime` datetime NOT NULL,
|   `studentNumber` int(11) NOT NULL,
|   `homeworkId` int(11) NOT NULL,
|   PRIMARY KEY (`homeworkId`),
|   KEY `studentNumber` (`studentNumber`)
| ) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+-----+
```

4.3.2 描述

Field	Type	Null	Key	Default	Extra
score	float	NO		NULL	
taskId	int(11)	NO		NULL	
uploadTime	datetime	NO		NULL	
studentNumber	int(11)	NO	MUL	NULL	
homeworkId	int(11)	NO	PRI	NULL	

4.4 question

4.4.1 创建

```
-----+
| question | CREATE TABLE `question` (
|   `teacherId` int(11) NOT NULL,
|   `questionId` int(11) NOT NULL,
|   `answer` text CHARACTER SET utf8 NOT NULL,
|   `answeredTime` datetime NOT NULL,
|   `createTime` datetime NOT NULL,
|   `studentNumber` int(11) NOT NULL,
|   PRIMARY KEY (`questionId`)
| ) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+-----+
```

4.4.2 描述

Field	Type	Null	Key	Default	Extra
teacherId	int(11)	NO	PRI	NULL	
questionId	int(11)	NO		NULL	
answer	text	NO		NULL	
answeredTime	datetime	NO		NULL	
createTime	datetime	NO		NULL	
studentNumber	int(11)	NO		NULL	

4.5 score

4.5.1 创建

```
score | CREATE TABLE `score` (  
  `studentNumber` int(11) NOT NULL,  
  `subjectId` int(11) NOT NULL,  
  `score` float NOT NULL,  
  `teacherId` int(11) NOT NULL,  
  `checkTime` datetime NOT NULL,  
  PRIMARY KEY (`studentNumber`, `subjectId`),  
  KEY `studentNumber` (`studentNumber`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
```

4.5.2 描述

Field	Type	Null	Key	Default	Extra
studentNumber	int(11)	NO	PRI	NULL	
subjectId	int(11)	NO	PRI	NULL	
score	float	NO		NULL	
teacherId	int(11)	NO		NULL	
checkTime	datetime	NO		NULL	

4.6 student

4.6.1 创建

```
+-----+
| student | CREATE TABLE `student` (
|   `studentNumber` int(11) NOT NULL,
|   `major` varchar(20) CHARACTER SET utf8 NOT NULL,
|   `totalScore` float NOT NULL,
|   `averageScore` float NOT NULL,
|   `name` varchar(10) CHARACTER SET utf8 NOT NULL,
|   `sex` tinyint(1) NOT NULL COMMENT '1为男, 0为女',
|   PRIMARY KEY (`studentNumber`)
| ) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+-----+
```

4.6.2 描述

Field	Type	Null	Key	Default	Extra
studentNumber	int(11)	NO	PRI	NULL	
major	varchar(20)	NO		NULL	
totalScore	float	NO		NULL	
averageScore	float	NO		NULL	
name	varchar(10)	NO		NULL	
sex	tinyint(1)	NO		NULL	

4.7 subject

4.7.1 创建

```
+-----+
| subject | CREATE TABLE `subject` (
|   `subjectId` int(11) NOT NULL,
|   `kind` varchar(10) CHARACTER SET utf8 NOT NULL,
|   `name` varchar(20) CHARACTER SET utf8 NOT NULL,
|   `teacherId` int(11) NOT NULL,
|   PRIMARY KEY (`subjectId`,`teacherId`)
| ) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+-----+
```

4.7.2 描述

Field	Type	Null	Key	Default	Extra
subjectId	int(11)	NO	PRI	NULL	
kind	varchar(10)	NO		NULL	
name	varchar(20)	NO		NULL	
teacherId	int(11)	NO	PRI	NULL	

4.8 task

4.8.1 创建

```
task | CREATE TABLE `task` (  
  `subjectId` int(11) DEFAULT NULL,  
  `taskId` int(11) NOT NULL,  
  `teacherId` int(11) DEFAULT NULL,  
  `uploadTime` datetime DEFAULT NULL,  
  `content` text,  
  `ddl` datetime DEFAULT NULL,  
  PRIMARY KEY (`taskId`),  
  KEY `teacherId` (`teacherId`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
```

4.8.2 描述

Field	Type	Null	Key	Default	Extra
subjectId	int(11)	YES		NULL	
taskId	int(11)	NO	PRI	NULL	
teacherId	int(11)	YES	MUL	NULL	
uploadTime	datetime	YES		NULL	
content	text	YES		NULL	
ddl	datetime	YES		NULL	

4.9 teacher

4.9.1 创建

```
-----+
| teacher | CREATE TABLE `teacher` (
|   `teacherId` int(11) NOT NULL,
|   `password` varchar(16) CHARACTER SET utf8 NOT NULL,
|   `username` varchar(16) CHARACTER SET utf8 NOT NULL,
|   `isSuperuser` tinyint(1) NOT NULL,
|   PRIMARY KEY (`teacherId`)
| ) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+-----+
```

4.9.2 描述

Field	Type	Null	Key	Default	Extra
teacherId	int(11)	NO	PRI	NULL	
password	varchar(16)	NO		NULL	
username	varchar(16)	NO		NULL	
isSuperuser	tinyint(1)	NO		NULL	

5 数据操作

5.1 排序

```
mysql> select * from teacher;
```

teacherId	password	username	isSuperuser
1020160318	0318	王亚静	0
1020161300	xiaokeai	小可爱	1
1020161302	shinian	施念	1
1020161305	shiniana	施念	1
1020161308	nian	方心	0
1020171309	shi	静静	0
1120161309	shinian191498	施念	1

```
7 rows in set (0.00 sec)
```

5.2 限制

5.2.1 限制导致失败

```
mysql> INSERT INTO `task` (`subjectId`, `taskId`, `teacherId`, `uploadTime`, `content`, `ddl`) VALUES ('1010032', '0000001', '1020161309', '2018-04-12 00:00:00', 'it\'s just a test!', '2018-04-25 10:25:38')
-> ;
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`weibclass`.`task`, CONSTRAINT `task_ibfk_1` FOREIGN KEY (`teacherId`) REFERENCES `teacher` (`teacherId`))
mysql>
```

5.2.2 成功

```
mysql> INSERT INTO `task` (`subjectId`, `taskId`, `teacherId`, `uploadTime`, `content`, `ddl`) VALUES ('1010032', '0000001', '1020161302', '2018-04-12 00:00:00', 'it\'s just a test!', '2018-04-25 10:25:38')
-> ;
Query OK, 1 row affected (0.01 sec)
```

5.3 聚合

```
mysql> select isSuperuser, count(1) from teacher group by isSuperuser;
```

isSuperuser	count(1)
0	3
1	4

5.4 表的连接

5.4.1 左连接

```
mysql> select teacher.teacherId, username, isSuperuser from teacher left join task on teacher.teacherId=task.teacherId;
```

teacherId	username	isSuperuser
1020160318	王亚静	0
1020161300	小可爱	1
1020161302	施念	1
1020161305	施念	1
1020161308	方心	0
1020171309	静静	0
1120161309	施念	1

5.4.2 右连接

```
mysql> select task.teacherId,content from teacher right join task on teacher.teacherId=task.teacherId;
```

teacherId	content
1020161308	bingo!
54666666	It's just a test!
1120161303	加油!
1120161304	加油!

5.5 子查询

```
mysql> select * from teacher teacherId where teacherId in (select teacherId from task);
```

teacherId	password	username	isSuperuser
1020161308	nian	方心	0

5.6 记录联合

```
mysql> select teacherId from teacher  
-> union all  
-> select teacherId from task;
```

teacherId
1020160318
1020161300
1020161302
1020161305
1020161308
1020171309
1120161309
54666666
1020161308
1120161303
1120161304