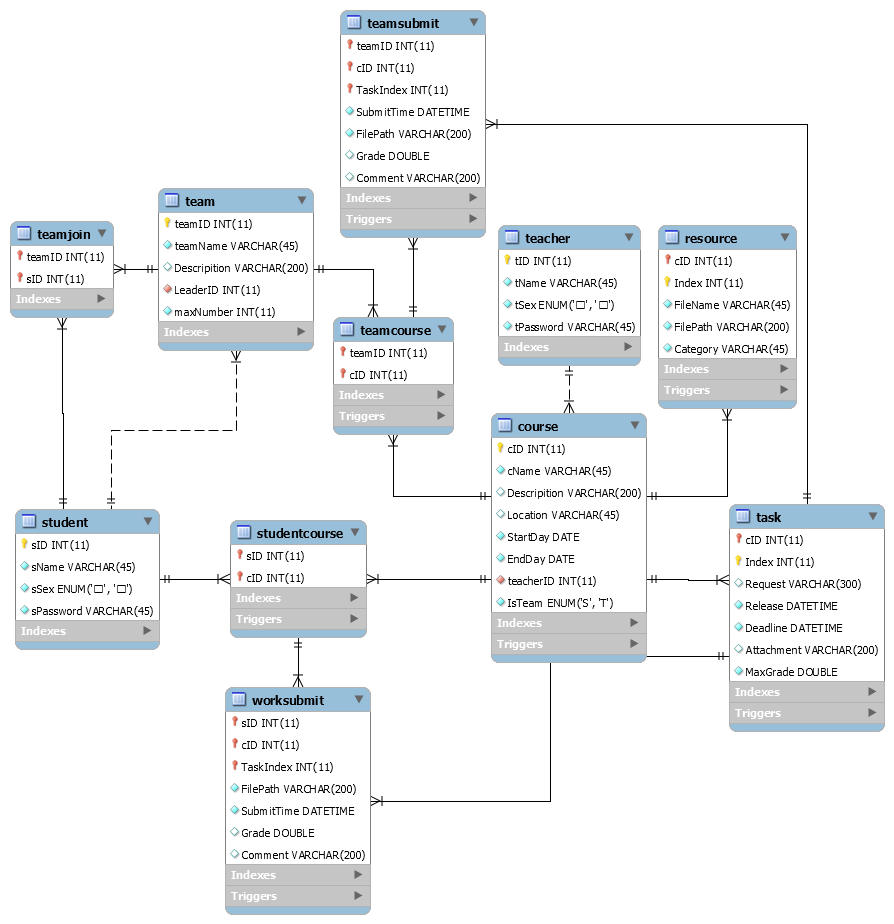
数据库2.0版本ER图



其中index属性表示作业或者资源在某一课程中示作业/资源列表的顺序（从0开始）

## SQL语句样例

（学生sID：102，团队teamID：401，课程cID：301）

1.常用的增删改语句

以worksubmit表为例，增加一条作业提交记录

INSERT INTO `coursesystem`.`worksubmit` (`sID`, `cID`, `TaskIndex`, `FilePath`, `SubmitTime`, `Grade`, `Comment`) VALUES ('101', '301, '0', 'D:\\第一章作业.doc', '2016/07/5 14:50:00', '100', '该同学作业完成的很好,没有任何错误.');

删除一条作业提交记录

DELETE FROM `coursesystem`.`worksubmit` WHERE `sID`='102' and`cID`='302' and`TaskIndex`='0';

在表中修改分数

UPDATE `coursesystem`.`worksubmit` SET `Grade`='80' WHERE `sID`='102' and`cID`='302' and`TaskIndex`='0';

查询语句：

1. 学生事务样例

1)查看报名的所有课程信息：

SELECT course.\* FROM course,studentcourse WHERE course.cID = studentcourse.cID and studentcourse.sID='102';

1. 查看一门课程的所有作业：

SELECT Task.\* FROM Task where cID='301';

1. 查看我所在的团队的全部信息：

SELECT team.\* FROM teamjoin,team WHERE teamjoin.teamID = team.teamID and teamjoin.sID = '102';

1. 查看某一团队的当前人数：

SELECT count(sID) FROM teamjoin WHEREteamID='401';

1. 查看某一作业的提交记录

SELECT worksubmit.\* FROM worksubmit WHERE worksubmit.sID='101' and worksubmit.cID='301' and worksubmit.index='0';

CREATE DATABASE `coursesystem` /\*!40100 DEFAULT CHARACTER SET utf8 \*/ ;

USE `coursesystem`;

CREATE TABLE `student` (

`sID` int(11) NOT NULL,

`sName` varchar(45) NOT NULL,

`sSex` varchar(2) NOT NULL,

`sPassword` varchar(45) NOT NULL,

PRIMARY KEY (`sID`),

UNIQUE KEY `sID\_UNIQUE` (`sID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `teacher` (  
   `tID` int(11) NOT NULL,  
   `tName` varchar(45) NOT NULL,  
   `tSex` varchar(2) NOT NULL,  
   `tPassword` varchar(45) NOT NULL,  
   PRIMARY KEY (`tID`),  
   UNIQUE KEY `tID\_UNIQUE` (`tID`)  
 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;

CREATE TABLE `course` (

`cID` int(11) NOT NULL,

`cName` varchar(45) NOT NULL,

`Descripition` varchar(200) DEFAULT NULL,

`Location` varchar(45) DEFAULT NULL,

`StartDay` date NOT NULL,

`EndDay` date NOT NULL,

`teacherID` int(11) NOT NULL,

PRIMARY KEY (`cID`),

UNIQUE KEY `cID\_UNIQUE` (`cID`),

KEY `coursetea\_idx` (`teacherID`),

CONSTRAINT `coursetea` FOREIGN KEY (`teacherID`) REFERENCES `teacher` (`tID`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `studentcourse` (

`sID` int(11) NOT NULL,

`cID` int(11) NOT NULL,

PRIMARY KEY (`sID`,`cID`),

KEY `cs\_idx` (`cID`),

CONSTRAINT `cs` FOREIGN KEY (`cID`) REFERENCES `course` (`cID`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `sc` FOREIGN KEY (`sID`) REFERENCES `student` (`sID`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `resource` (

`cID` int(11) NOT NULL,

`Index` int(11) NOT NULL,

`FileName` varchar(45) NOT NULL,

`FilePath` varchar(200) NOT NULL,

`Category` varchar(45) NOT NULL,

PRIMARY KEY (`cID`),

CONSTRAINT `course` FOREIGN KEY (`cID`) REFERENCES `course` (`cID`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `task` (

`cID` int(11) NOT NULL,

`Index` int(11) NOT NULL,

`Request` varchar(300) DEFAULT NULL,

`Release` datetime NOT NULL,

`Deadline` datetime NOT NULL,

`Attachment` varchar(200) DEFAULT NULL,

PRIMARY KEY (`cID`,`Index`),

CONSTRAINT `coursecon` FOREIGN KEY (`cID`) REFERENCES `course` (`cID`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `worksubmit` (  
   `sID` int(11) NOT NULL,  
   `cID` int(11) NOT NULL,  
   `TaskIndex` int(11) NOT NULL,  
   `FilePath` varchar(200) NOT NULL,  
   `SubmitTime` datetime NOT NULL,  
   `Grade` double DEFAULT NULL,  
   `Comment` varchar(200) DEFAULT NULL,  
   PRIMARY KEY (`sID`,`cID`,`TaskIndex`),  
   KEY `ii\_idx` (`cID`,`TaskIndex`),  
   CONSTRAINT `ii` FOREIGN KEY (`cID`, `TaskIndex`) REFERENCES `task` (`cID`, `Index`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
   CONSTRAINT `ll` FOREIGN KEY (`sID`, `cID`) REFERENCES `studentcourse` (`sID`, `cID`) ON DELETE NO ACTION ON UPDATE NO ACTION  
 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;

CREATE TABLE `team` (

`teamID` int(11) NOT NULL,

`teamName` varchar(45) NOT NULL,

`Descripition` varchar(200) DEFAULT NULL,

`LeaderID` int(11) NOT NULL,

`maxNumber` int(11) NOT NULL,

PRIMARY KEY (`teamID`),

UNIQUE KEY `teamID\_UNIQUE` (`teamID`),

KEY `teams\_idx` (`LeaderID`),

CONSTRAINT `teams` FOREIGN KEY (`LeaderID`) REFERENCES `student` (`sID`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `teamjoin` (

`teamID` int(11) NOT NULL,

`sID` int(11) NOT NULL,

PRIMARY KEY (`teamID`,`sID`),

KEY `teamstu\_idx` (`sID`),

CONSTRAINT `teamstu` FOREIGN KEY (`sID`) REFERENCES `student` (`sID`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `teamteam` FOREIGN KEY (`teamID`) REFERENCES `team` (`teamID`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

插入数据：

INSERT INTO `coursesystem`.`student` (`sID`, `sName`, `sSex`, `sPassword`) VALUES ('101', '容志浩', '男', '12345');  
INSERT INTO `coursesystem`.`student` (`sID`, `sName`, `sSex`, `sPassword`) VALUES ('102', '樊伟富', '男', '9480');

INSERT INTO `coursesystem`.`teacher` (`tID`, `tName`, `tSex`, `tPassword`) VALUES ('201', '王海泉', '男', '12345');  
INSERT INTO `coursesystem`.`teacher` (`tID`, `tName`, `tSex`, `tPassword`) VALUES ('202', '林广艳', '女', '11111');

INSERT INTO `coursesystem`.`course` (`cID`, `cName`, `Descripition`, `Location`, `StartDay`, `EndDay`, `teacherID`) VALUES ('301', '计算机网络', '本书系统地介绍了计算机网络的基本概念、原理与技术，包括绪论、物理层、数据链路层、局域网、网络层、传输层、应用层和网络安全共8章内容，各章后附有丰富的习题.', '主北301', '2016-07-01', '2016-07-05', '201');

INSERT INTO `coursesystem`.`course` (`cID`, `cName`, `Descripition`, `Location`, `StartDay`, `EndDay`, `teacherID`) VALUES ('302', '数据结构', '数据结构是计算机科学的一门非常重要的专业基础课，内容丰富，涉及面广，我校计算机专业的本科主干基础课程，也是非计算机类本科生和研究生学习计算机的选修课。', 'J0-001', '2014-05-01', '2014-06-01', '202');

INSERT INTO `coursesystem`.`resource` (`cID`, `Index`, `FileName`, `FilePath`, `Category`) VALUES ('301', '0', '计算机网络课件第一章', 'D:\\第一章课件.ppt', '课件');

INSERT INTO `coursesystem`.`studentcourse` (`sID`, `cID`) VALUES ('101', '301');  
INSERT INTO `coursesystem`.`studentcourse` (`sID`, `cID`) VALUES ('101', '302');  
INSERT INTO `coursesystem`.`studentcourse` (`sID`, `cID`) VALUES ('102', '302');

INSERT INTO `coursesystem`.`task` (`cID`, `Index`, `Request`, `Release`, `Deadline`, `Attachment`) VALUES ('301', '0', '完成第一章作业.', '2016-07-04 14:41:00', '2016-07-06 12:00:00', 'C:\\Windows');  
INSERT INTO `coursesystem`.`task` (`cID`, `Index`, `Request`, `Release`, `Deadline`, `Attachment`) VALUES ('302', '0', '完成第二章作业.', '2016-07-01 12:34:00', '2016-07-04 13:00:00', 'C:\\Program Files');

INSERT INTO `coursesystem`.`team` (`teamID`, `teamName`, `Descripition`, `LeaderID`, `maxNumber`) VALUES ('401', '二元一次方程组', '我们喜欢上实践课', '101', '10');

INSERT INTO `coursesystem`.`team` (`teamID`, `teamName`, `Descripition`, `LeaderID`, `maxNumber`) VALUES ('402', '一元二次方程', '我们喜欢数学课', '102', '5');

INSERT INTO `coursesystem`.`teamjoin` (`teamID`, `sID`) VALUES ('401', '101');  
INSERT INTO `coursesystem`.`teamjoin` (`teamID`, `sID`) VALUES ('401', '102');  
INSERT INTO `coursesystem`.`teamjoin` (`teamID`, `sID`) VALUES ('402', '102');

INSERT INTO `coursesystem`.`worksubmit` (`sID`, `cID`, `TaskIndex`, `FilePath`, `SubmitTime`, `Grade`, `Comment`) VALUES ('101', '301', '0', 'D:\\第一章作业.doc', '2016/07/5 14:50:00', '100', '该同学作业完成的很好,没有任何错误.');

INSERT INTO `coursesystem`.`worksubmit` (`sID`, `cID`, `TaskIndex`, `FilePath`, `SubmitTime`, `Grade`, `Comment`) VALUES ('101', '302', '0', 'C:\\第二章作业.doc', '2016/07/3 13:03:22', '80', '该同学第3道习题没有理解,回答的不对.');

INSERT INTO `coursesystem`.`worksubmit` (`sID`, `cID`, `TaskIndex`, `FilePath`, `SubmitTime`, `Grade`, `Comment`) VALUES ('102', '302', '0', 'E:\\第二次作业.doc', '2016/07/02 13:0:0', '90', '该同学作业完成还可以,有小错误.');

ALTER TABLE `coursesystem`.`student`

CHANGE COLUMN `sSex` `sSex` ENUM('男', '女') NOT NULL ;

ALTER TABLE `coursesystem`.`teacher`

CHANGE COLUMN `tSex` `tSex` ENUM('男', '女') NOT NULL ;

USE `coursesystem`;

DELIMITER $$

DROP TRIGGER IF EXISTS coursesystem.course\_AFTER\_INSERT$$

USE `coursesystem`$$

CREATE DEFINER = CURRENT\_USER TRIGGER `coursesystem`.`course\_AFTER\_INSERT` AFTER INSERT ON `course` FOR EACH ROW

BEGIN

if (new.StartDay>=new.EndDay) Then

delete from course where cID = new.cID;

End if ;

END$$

DELIMITER ;

USE `coursesystem`;

DELIMITER $$

DROP TRIGGER IF EXISTS coursesystem.task\_BEFORE\_INSERT$$

USE `coursesystem`$$

CREATE DEFINER=`root`@`localhost` TRIGGER `coursesystem`.`task\_BEFORE\_INSERT` BEFORE INSERT ON `task` FOR EACH ROW

BEGIN

if ((select count(cID) from task where cID = new.cID) = 0) then

set new.`Index`=0;

else

set new.`Index`=(select max(`Index`) from task where cID=new.cID)+1;

end if;

END$$

DELIMITER ;

ALTER TABLE `coursesystem`.`resource`

DROP PRIMARY KEY,

ADD PRIMARY KEY (`cID`, `Index`);

USE `coursesystem`;

DELIMITER $$

DROP TRIGGER IF EXISTS coursesystem.resource\_BEFORE\_INSERT$$

USE `coursesystem`$$

CREATE DEFINER = CURRENT\_USER TRIGGER `coursesystem`.`resource\_BEFORE\_INSERT` BEFORE INSERT ON `resource` FOR EACH ROW

BEGIN

if ((select count(cID) from resource where cID = new.cID) = 0) then

set new.`Index`=0;

else

set new.`Index`=(select max(`Index`) from resource where cID=new.cID)+1;

end if;

END$$

DELIMITER ;

ALTER TABLE `coursesystem`.`task`

ADD COLUMN `MaxGrade` DOUBLE NOT NULL AFTER `Attachment`;

UPDATE `coursesystem`.`task` SET `MaxGrade`='10.0' WHERE `cID`='301' and`Index`='0';

UPDATE `coursesystem`.`task` SET `MaxGrade`='5.0' WHERE `cID`='301' and`Index`='1';

UPDATE `coursesystem`.`task` SET `MaxGrade`='8.0' WHERE `cID`='302' and`Index`='0';

USE `coursesystem`;

DELIMITER $$

DROP TRIGGER IF EXISTS coursesystem.worksubmit\_AFTER\_INSERT$$

USE `coursesystem`$$

CREATE DEFINER=`root`@`localhost` TRIGGER `coursesystem`.`worksubmit\_AFTER\_INSERT` AFTER INSERT ON `worksubmit` FOR EACH ROW

BEGIN

if (new.grade>(select MaxGrade from task where task.cID=new.cID and task.`Index`=new.`TaskIndex` )) then

delete from worksubmit where worksubmit.sID=new.sID and worksubmit.cID=new.cID and worksubmit.`TaskIndex`=new.`TaskIndex`;

end if;

END$$

DELIMITER ;

CREATE TABLE `coursesystem`.`teamcourse` (

`teamID` INT NOT NULL,

`cID` INT NOT NULL,

PRIMARY KEY (`teamID`, `cID`),

INDEX `courser\_idx` (`cID` ASC),

CONSTRAINT `teamr`

FOREIGN KEY (`teamID`)

REFERENCES `coursesystem`.`team` (`teamID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `courser`

FOREIGN KEY (`cID`)

REFERENCES `coursesystem`.`course` (`cID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

ALTER TABLE `coursesystem`.`course`

ADD COLUMN `IsTeam` ENUM('S', 'T') NOT NULL AFTER `teacherID`;

INSERT INTO `coursesystem`.`course` (`cID`, `cName`, `Descripition`, `Location`, `StartDay`, `EndDay`, `teacherID`, `IsTeam`) VALUES ('304', '生产实践', '同学们实践吧~', '工训317', '2016-07-02', '2016-07-12', '202', 'T');

USE `coursesystem`;

DELIMITER $$

DROP TRIGGER IF EXISTS coursesystem.studentcourse\_AFTER\_INSERT$$

USE `coursesystem`$$

CREATE DEFINER=`root`@`localhost` TRIGGER `coursesystem`.`studentcourse\_AFTER\_INSERT` AFTER INSERT ON `studentcourse` FOR EACH ROW

BEGIN

if ((select course.IsTeam from course where new.cID=course.cID)='T' )then

delete from studentcourse where studentcourse.cID=new.cID and studentcourse.sID=new.sID;

end if;

END$$

DELIMITER ;

INSERT INTO `coursesystem`.`studentcourse` (`sID`, `cID`) VALUES ('101', '303');

USE `coursesystem`;

DELIMITER $$

DROP TRIGGER IF EXISTS coursesystem.teamcourse\_AFTER\_INSERT$$

USE `coursesystem`$$

CREATE DEFINER = CURRENT\_USER TRIGGER `coursesystem`.`teamcourse\_AFTER\_INSERT` AFTER INSERT ON `teamcourse` FOR EACH ROW

BEGIN

if ((select course.IsTeam from course where new.cID=course.cID)='S' )then

delete from teamcourse where teamcourse.cID=new.cID and teamcourse.teamID=new.teamID;

end if;

END$$

DELIMITER ;

INSERT INTO `coursesystem`.`teamcourse` (`teamID`, `cID`) VALUES ('401', '304');

CREATE TABLE `coursesystem`.`teamsubmit` (

`teamID` INT NOT NULL,

`cID` INT NOT NULL,

`TaskIndex` INT NOT NULL,

`SubmitTime` DATETIME NOT NULL,

`FilePath` VARCHAR(200) NOT NULL,

`Grade` DOUBLE NULL,

`Comment` VARCHAR(200) NULL,

PRIMARY KEY (`teamID`, `cID`, `TaskIndex`),

INDEX `task\_idx` (`cID` ASC, `TaskIndex` ASC),

CONSTRAINT `teamcourse`

FOREIGN KEY (`teamID` , `cID`)

REFERENCES `coursesystem`.`teamcourse` (`teamID` , `cID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `task`

FOREIGN KEY (`cID` , `TaskIndex`)

REFERENCES `coursesystem`.`task` (`cID` , `Index`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

USE `coursesystem`;

DELIMITER $$

DROP TRIGGER IF EXISTS coursesystem.teamsubmit\_AFTER\_INSERT$$

USE `coursesystem`$$

CREATE DEFINER = CURRENT\_USER TRIGGER `coursesystem`.`teamsubmit\_AFTER\_INSERT` AFTER INSERT ON `teamsubmit` FOR EACH ROW

BEGIN

if (new.grade>(select MaxGrade from task where task.cID=new.cID and task.`Index`=new.`TaskIndex` )) then

delete from teamsubmit where teamsubmit.teamID=new.teamID and teamsubmit.cID=new.cID and teamsubmit.`TaskIndex`=new.`TaskIndex`;

end if;

END$$

DELIMITER ;

INSERT INTO `coursesystem`.`task` (`cID`, `Index`, `Request`, `Release`, `Deadline`, `MaxGrade`) VALUES ('304', '0', '完成用户故事', '2016-07-01 18:00:00', '2016-07-02 15:00:00', '10.0');

INSERT INTO `coursesystem`.`teamsubmit` (`teamID`, `cID`, `TaskIndex`, `SubmitTime`, `FilePath`, `Grade`, `Comment`) VALUES ('401', '304', '0', '2016-07-02', 'c\\work.docx', '8', '用户故事不够完善');

用户登录：

(“sID”:”x”, “type”:”1”) 学生

(“sID”:”x”, “sPassword”:”x”, “key”:”x”)

(“sID”:”x”, “type”:”2”) 教师

(“sID”:”x”, “sPassword”:”x”, “key”:”x”)

(“sID”:”x”, “type”:”3”) 教务

没做

学生部分：

1. 查看自己课程列表信息

(“sID”:”x”, “key”:”x”)

(“cID”:”301”, “cName”:”计算机网络”, “Descripition”:”x”, “Location”:”主北301”, “StartDay”:”2016-07-01”, “EndDay”:”2016-07-05”, “tID”:”201”)

1. 查看个人已提交作业成绩及评论

(“cID”:”x”, “TaskIndex”:”x”, “sID”:”x”, “nowtime”:”x”, “key”:”x”)

(“Request”:”x”, “Release”:’x’, ‘Deadline’:’x’, ‘MaxGrade’:’x’, ‘FilePath’:’x’, ‘SubmitTime’:’x’, ‘Grade’:’x’, ‘Comment’:’x’)

1. 查看布置的个人作业及要求

(‘cID’:’x’, ‘TaskIndex’:’x’, ‘sID’:’x’, ‘nowtime’:’x’, “key”:”x”,”Deadline”:”x”)

当Deadline>nowtime时返回,

(‘TaskIndex’:’x’, ‘Request’:’x’, ‘Release’:’x’, ‘Deadline’:’x’, ‘Attachment’:’x’, ‘MaxGrade’:’x’)

1. 提交个人作业

(‘cID’:’x’, ‘TaskIndex’:’x’, ‘sID’:’x’, ‘nowtime’:’x’, ‘FilePath’:’x’, “key”:”x”)

无返回，更新数据库

1. 提交团队作业

(“teamID”:”x”, “sID’:”x”, “TaskIndex”:”x”, “cID”:”x”, “nowtime”:”x”, “FilePath”:”x”, “key”:”x”)

6.查看团队已提交作业成绩及评论

(“cID”:”x”, “TaskIndex”:”x”, “teamID”:”x”, “nowtime”:”x”, “key”:”x”)

(“Request”:”x”, “Release”:’x’, ‘Deadline’:’x’, ‘MaxGrade’:’x’, ‘FilePath’:’x’, ‘SubmitTime’:’x’, ‘Grade’:’x’, ‘Comment’:’x’)

7.查看布置的团队作业及要求

(‘cID’:’x’, ‘TaskIndex’:’x’, ‘teamID’:’x’, ‘nowtime’:’x’, “key”:”x”)

(‘TaskIndex’:’x’, ‘Request’:’x’, ‘Release’:’x’, ‘Deadline’:’x’, ‘Attachment’:’x’, ‘MaxGrade’:’x’)

1. 查看和下载课程资源

(“cID”:”x”, “resource.`Index`”:”x”, “key”:”x”)

(“cID”:”x”, “resource.`Index`”:”x”, “resource.Filename”:”x”, “resource.Filepath”:”x”, “Category”:”x”)

1. 查看自己的团队列表和信息

(“sID”:”x”, “key”:”x”)

(“teamID”:”x”, “teamName”:”x”, “team.Descripition”:”x”, “LeaderID”:”x”, “maxNumber”:”x”)

1. 完善自己的团队信息

(“sID”:”x”, “key”:”x”, “teamName”:”x”, “team.Descripition”:”x”, “maxNumber”:”x”, “key”:”x”)

无返回，更新数据库

1. 查看所有的团队列表

(“key”:”x”)

(“teamID”:”x”, “teamName”:”x”, “team.Descripition”:”x”, “LeaderID”:”x”, “maxNumber”:”x”)

1. 申请感兴趣的团队

(“sID”:”x”, “teamID”:”x” )

在下面的count<maxNumber时可以申请，无返回，更新数据库

这包括要求一个团队的当前人数

(“teamID”:”x”)

(“count”:”x”)

1. 申请团队课程

(“teamID”:”x”, “cID”:”x”, “IsTeam”:”x”,”sID”:”x”)

当IsTeam=’T’时，可以申请，无返回，更新数据库