--一个题目涉及到的50个Sql语句

--（下面表的结构以给出,自己在数据库中建立表.并且添加相应的数据,数据要全面些. 其中Student表中,SId为学生的ID）

------------------------------------表结构--------------------------------------

--学生表tblStudent（编号StuId、姓名StuName、年龄StuAge、性别StuSex）

--课程表tblCourse（课程编号CourseId、课程名称CourseName、教师编号TeaId）

--成绩表tblScore（学生编号StuId、课程编号CourseId、成绩Score）

--教师表tblTeacher（教师编号TeaId、姓名TeaName）

---------------------------------------------------------------------------------

**--1、查询“001”课程比“002”课程成绩高的所有学生的学号，姓名；**

    Select StuId,StuName From tblStudent s1

        Where (Select Score From tblScore t1 Where t1.StuId=s1.stuId And t1.CourseId='001')>

            (Select Score From tblScore t2 Where t2.StuId=s1.stuId And t2.CourseId='002')

**--2、查询平均成绩大于60分的同学的学号和平均成绩；**

    Select StuId,Avg(Score) as AvgScore From tblScore

        Group By StuId

            Having Avg(Score)>60

**--3、查询所有同学的学号、姓名、选课数、总成绩；**

    Select StuId,StuName,

       (Select Count(CourseId) From tblScore t1 Where t1.StuId=s1.StuId) as SelCourses,

       (Select Sum(Score) From tblScore t2 Where t2.StuId=s1.StuId) as SumScore

            From tblStudent s1;

**--4、查询姓“李”的老师的个数；**

    Select Count(\*) From tblTeacher Where TeaName like '李%'

**--5、查询没学过“叶平”老师课的同学的学号、姓名；**

    Select StuId,StuName From tblStudent

        Where StuId Not In

            (

                Select StuID From tblScore sc

                    Inner Join tblCourse cu ON sc.CourseId=cu.CourseId

                    Inner Join tblTeacher tc ON cu.TeaId=tc.TeaId

                        Where tc.TeaName='叶平'

            )

    Select StuId,StuName From tblStudent

         Where StuId Not In

            (select sc.stuid from tblscore sc,tblcourse cu,tblteacher tc

                where  sc.courseid = cu.courseid and cu.teaid=tc.teaid

                        and tc.teaname = '叶平');

';

**--6、查询学过“001”并且也学过编号“002”课程的同学的学号、姓名；**

    Select StuId,StuName From tblStudent st

        Where (Select Count(\*) From tblScore s1 Where s1.StuId=st.StuId And s1.CourseId='001')>0

            And

            (Select Count(\*) From tblScore s2 Where s2.StuId=st.StuId And s2.CourseId='002')>0

--7、查询学过“叶平”老师所教的所有课的同学的学号、姓名；

    Select StuId,StuName From tblStudent st Where not exists

        (

            Select CourseID From tblCourse cu Inner Join tblTeacher tc On cu.TeaID=tc.TeaID

                Where tc.TeaName='叶平' And CourseID not in

                (Select CourseID From tblScore Where StuID=st.StuID)

        )

--8、查询课程编号“002”的成绩比课程编号“001”课程低的所有同学的学号、姓名；

    Select StuId,StuName From tblStudent s1

        Where (Select Score From tblScore t1 Where t1.StuId=s1.stuId And t1.CourseId='001')>

            (Select Score From tblScore t2 Where t2.StuId=s1.stuId And t2.CourseId='002')

9、查询所有课程成绩小于分的同学的学号、姓名；

    Select StuId,StuName From tblStudent st

        Where StuId Not IN

            (Select StuId From tblScore sc Where st.StuId=sc.StuId And Score>60)

--10、查询没有学全所有课的同学的学号、姓名；

    Select StuId,StuName From tblStudent st

        Where (Select Count(\*) From tblScore sc Where st.StuId=sc.StuId)<

            (Select Count(\*) From tblCourse)

--11、查询至少有一门课与学号为“1001”的同学所学相同的同学的学号和姓名；

    ------运用连接查询

    Select DistInct st.StuId,StuName From tblStudent st

        Inner Join tblScore sc ON st.StuId=sc.StuId

            Where sc.CourseId IN (Select CourseId From tblScore Where StuId='1001')

    ------嵌套子查询

    Select StuId,StuName From tblStudent

        Where StuId In

        (

            Select Distinct StuId From tblScore Where CourseId In (Select CourseId From tblScore Where StuId='1001')

        )

--12、查询至少学过学号为“1001”同学所有课程的其他同学学号和姓名；

    Select StuId,StuName From tblStudent

        Where StuId In

        (

            Select Distinct StuId From tblScore Where CourseId Not In (Select CourseId From tblScore Where StuId='1001')

--13、把“SC”表中“叶平”老师教的课的成绩都更改为此课程的平均成绩； (从子查询中获取父查询中的表名，这样也行？？？？)

    --创建测试表

    Select \* Into Sc From tblScore

    go

    Update Sc Set Score=(Select Avg(Score) From tblScore s1 Where s1.CourseId=sc.CourseId)

        Where CourseId IN

            (Select CourseId From tblCourse cs INNER JOIN tblTeacher tc ON cs.TeaID=tc.TeaID WHERE TeaName ='叶平')

--14、查询和“1002”号的同学学习的课程完全相同的其他同学学号和姓名；

    Select StuID,StuName From tblStudent st

        Where StuId <> '1002'

            And

            Not Exists(Select \* From tblScore sc Where sc.StuId=st.StuId And CourseId Not In (Select CourseId From tblScore Where StuId='1002'))

            And

            Not Exists(Select \* From tblScore Where StuId='1002' And CourseId Not In (Select CourseId From tblScore sc Where sc.StuId=st.StuId))

--15、删除学习“叶平”老师课的SC表记录；

    Delete From tblScore Where CourseId IN

        (Select CourseId From tblCourse cs INNER JOIN tblTeacher tc ON cs.TeaId=tc.TeaId Where tc.TeaName='叶平')

--16、向SC表中插入一些记录，这些记录要求符合以下条件：没有上过编号“003”课程的同学学号、''号课的平均成绩；

    Insert Into tblScore (StuId,CourseId,Score)

        Select StuId,'002',(Select Avg(Score) From tblScore Where CourseId='002') From tblScore Where

            StuId Not In (Select StuId From tblScore Where CourseId='003')

--17、按平均成绩从高到低显示所有学生的“数据库”、“企业管理”、“英语”三门的课程成绩，按如下形式显示： 学生ID,,数据库,企业管理,英语,有效课程数,有效平均分

    Select StuId

        ,数据库=(Select Score From tblScore sc Inner Join tblCourse cs On sc.CourseId=cs.CourseId Where CourseName='数据库' And sc.StuID=st.StuId)

        ,企业管理=(Select Score From tblScore sc Inner Join tblCourse cs On sc.CourseId=cs.CourseId Where CourseName='企业管理' And sc.StuID=st.StuId)

        ,英语=(Select Score From tblScore sc Inner Join tblCourse cs On sc.CourseId=cs.CourseId Where CourseName='英语' And sc.StuID=st.StuId)

        ,有效课程数=(Select Count(Score) From tblScore sc Inner Join tblCourse cs On sc.CourseId=cs.CourseId Where (CourseName='数据库' or CourseName='企业管理' or CourseName='英语') And sc.StuID=st.StuId)

        ,有效平均分=(Select Avg(Score) From tblScore sc Inner Join tblCourse cs On sc.CourseId=cs.CourseId Where (CourseName='数据库' or CourseName='企业管理' or CourseName='英语') And sc.StuID=st.StuId)

        From tblStudent st

        Order by 有效平均分 Desc

--18、查询各科成绩最高和最低的分：以如下形式显示：课程ID，最高分，最低分

    Select CourseId as 课程ID, 最高分=(Select Max(Score) From tblScore sc Where sc.CourseId=cs.CourseId ),

        最低分=(Select Min(Score) From tblScore sc Where sc.CourseId=cs.CourseId )

        From tblCourse cs

--19、按各科平均成绩从低到高和及格率的百分数从高到低顺序 (百分数后如何格式化为两位小数??)

    Select 课程ID,平均分,及格率 From

        (Select CourseId as 课程ID, 平均分=(Select Avg(Score) From tblScore sc Where sc.CourseId=cs.CourseId ),

            及格率=Convert(varchar(10),((Select Count(\*) From tblScore sc Where sc.CourseId=cs.CourseId And sc.Score>=60)\*10000/(Select Count(\*) From tblScore sc Where sc.CourseId=cs.CourseId))/100)+'%'

            From tblScore cs) as tmp

        Group by 课程ID,平均分,及格率

            Order by 平均分, Convert(float,substring(及格率,1,len(及格率)-1)) Desc

--20、查询如下课程平均成绩和及格率的百分数(用"1行"显示): 企业管理（001），马克思（002），OO&UML （003），数据库（004）

    Select 课程ID=sc.CourseId,课程名称=cs.CourseName,平均成绩=Avg(Score)

        ,及格率 =Convert(varchar(10),((Select Count(Score) From tblScore Where CourseId=sc.CourseId And Score>=60)\*10000/Count(Score))/100.0)+'%'

        From tblScore sc

        Inner Join tblCourse cs ON sc.CourseId=cs.CourseId

        Where sc.CourseId like '00[1234]'

        Group By sc.CourseId,cs.CourseName

--21、查询不同老师所教不同课程平均分从高到低显示

    Select 课程ID=CourseId,课程名称=CourseName,授课教师=TeaName,平均成绩=(Select Avg(Score) From tblScore Where CourseId=cs.CourseId)

        From tblCourse cs

        Inner Join tblTeacher tc ON cs.TeaId=tc.TeaId

        Order by 平均成绩 Desc

--22、查询如下课程成绩第 3 名到第 6 名的学生成绩单：企业管理（001），马克思（002），UML （003），数据库（004） 格式：[学生ID],[学生姓名],企业管理,马克思,UML,数据库,平均成绩

    Select \* From

        (

            Select Top 6 学生ID=StuId,学生姓名=StuName

                ,企业管理=(Select Score From tblScore sc Inner Join tblCourse cs On sc.CourseId=cs.CourseId Where CourseName='企业管理' And sc.StuID=st.StuId)

                ,马克思=(Select Score From tblScore sc Inner Join tblCourse cs On sc.CourseId=cs.CourseId Where CourseName='马克思' And sc.StuID=st.StuId)

                ,UML=(Select Score From tblScore sc Inner Join tblCourse cs On sc.CourseId=cs.CourseId Where CourseName='UML' And sc.StuID=st.StuId)

                ,数据库=(Select Score From tblScore sc Inner Join tblCourse cs On sc.CourseId=cs.CourseId Where CourseName='数据库' And sc.StuID=st.StuId)

                ,平均成绩=(Select Avg(Score) From tblScore sc Inner Join tblCourse cs On sc.CourseId=cs.CourseId Where (CourseName='数据库' or CourseName='企业管理' or CourseName='UML'or CourseName='马克思') And sc.StuID=st.StuId)

                ,排名=Row\_Number() Over(Order by(Select Avg(Score) From tblScore sc Inner Join tblCourse cs On sc.CourseId=cs.CourseId Where (CourseName='数据库' or CourseName='企业管理' or CourseName='UML'or CourseName='马克思') And sc.StuID=st.StuId) DESC)

                From tblStudent st

                Order by 排名

        ) as tmp

        Where 排名 between 3 And 6

--23、统计列印各科成绩,各分数段人数:课程ID,课程名称,[100-85],[85-70],[70-60],[ <60]

    Select 课程ID=CourseId, 课程名称=CourseName

        ,[100-85]=(Select Count(\*) From tblScore sc Where CourseId=cs.CourseId And Score between 85 And 100)

        ,[85-70]=(Select Count(\*) From tblScore sc Where CourseId=cs.CourseId And Score between 70 And 84)

        ,[70-60]=(Select Count(\*) From tblScore sc Where CourseId=cs.CourseId And Score between 60 And 69)

        ,[<60]=(Select Count(\*) From tblScore sc Where CourseId=cs.CourseId And Score <60)

        From tblCourse cs

--24、查询学生平均成绩及其名次

    Select 学号=st.StuId, 姓名=StuName,平均成绩=sc.AvgScore,名次=(Dense\_Rank() Over(Order by sc.AvgScore Desc)) From tblStudent st

        Inner Join (Select StuId,Avg(Score) as AvgScore From tblScore Group by StuId) as sc On sc.StuId=st.StuId

        Order by 学号

--25、查询各科成绩前三名的记录:(不考虑成绩并列情况)

    Select 学号=StuId,课程号=CourseId,分数=Score

        From

        (Select Row\_Number() Over(order by CourseId,Score Desc) as i,\* From tblScore) as tmp    --得到一个临时的排名表,其中i表示编号

            Where i In

            (

                Select Top 3 i From (Select Row\_Number() Over(order by CourseId,Score Desc) as i,\* From tblScore) as t1 Where t1.CourseId=tmp.CourseId

            )

--26、查询每门课程被选修的学生数

    Select 课程ID=CourseId,选修人数=(Select Count(\*) From (Select Distinct StuId From tblScore Where CourseId=cs.CourseId) as tmp)

        From tblCourse cs

--27、查询出只选修了一门课程的全部学生的学号和姓名

    Select 学号=StuId,姓名=StuName

        From tblStudent st

        Where (Select Count(\*) From (Select Distinct CourseId From tblScore Where StuId=st.StuId) as tmp)=1

--28、查询男生、女生人数

    Select 男生人数=(select Count(\*) From tblStudent Where StuSex='男'),

            女生人数=(select Count(\*) From tblStudent Where StuSex='女')

--29、查询姓“张”的学生名单

    Select \* From tblStudent Where StuName like '张%'

--30、查询同名同性学生名单，并统计同名人数

    Select Distinct 学生姓名=StuName,同名人数=(Select Count(\*) From tblStudent s2 Where s2.StuName=st.StuName) From tblStudent st

        Where (Select Count(\*) From tblStudent s2 Where s2.StuName=st.StuName)>=2

--31、1981年出生的学生名单(注：Student表中Sage列的类型是datetime)

    Select \* From tblStudent Where Year(Sage)=1981

--32、查询每门课程的平均成绩，结果按平均成绩升序排列，平均成绩相同时，按课程号降序排列

    Select 课程ID=CourseId,课程名称=CourseName,平均成绩=(Select Avg(Score) From tblScore Where CourseId=cs.CourseId)

        From tblCourse cs

        Order by 平均成绩,CourseId Desc

--33、查询平均成绩大于85的所有学生的学号、姓名和平均成绩

    Select 学号=StuId,姓名=StuName,平均成绩=(Select Avg(Score) From tblScore Where StuId=st.StuId) From tblStudent st

        Where (Select Avg(Score) From tblScore Where StuId=st.StuId)>85

--34、查询课程名称为“数据库”，且分数低于60的学生姓名和分数

    Select 姓名=StuName,分数=Score From tblScore sc

        Inner Join tblStudent st On sc.StuId=st.StuId

        Inner Join tblCourse cs On sc.CourseId=cs.CourseId

        Where CourseName='数据库' And Score<60

--35、查询所有学生的选课情况；

    Select 学号=StuId,选课数=(Select Count(\*) From (Select Distinct CourseId From tblScore Where StuId=st.StuId) as tmp)

        From tblStudent st

    Select distinct 姓名=StuName,选修课程=CourseName From tblScore sc

        Inner Join tblStudent st On sc.StuId=st.StuId

        Inner Join tblCourse cs On sc.CourseId=cs.CourseId

--36、查询任何一门课程成绩在70分以上的姓名、课程名称和分数；

    Select 姓名=StuName,课程名称=CourseName,分数=Score From tblScore sc

        Inner Join tblStudent st On sc.StuId=st.StuId

        Inner Join tblCourse cs On sc.CourseId=cs.CourseId

        Where Score>=70

--37、查询不及格的课程，并按课程号从大到小排列

    Select \* From tblScore Where Score<60 order by CourseId Desc

--38、查询课程编号为003且课程成绩在80分以上的学生的学号和姓名；

    Select StuId,StuName From tblStudent

        Where StuId in

            (Select StuId From tblScore Where CourseId='003' And Score>=80)

--39、求选了课程的学生人数

    Select 选了课程的学生人数=Count(\*) From tblStudent st Where StuId IN (Select StuID From tblScore)

--40、查询选修“叶平”老师所授课程的学生中，成绩最高的学生姓名及其成绩

Select CourseId,CourseName

    ,该科最高学生=(Select StuName From tblStudent Where StuId in (Select Top 1 StuID From tblScore Where CourseId=cs.CourseId Order by Score Desc))

    ,成绩=(Select Top 1 Score From tblScore Where CourseId=cs.CourseId Order by Score Desc)

    From tblCourse cs Inner Join tblTeacher tc ON cs.TeaId=tc.TeaId

    Where TeaName='叶平'

--41、查询各个课程及相应的选修人数

    Select 课程ID=CourseId,选修人数=(Select Count(\*) From (Select Distinct StuId From tblScore Where CourseId=cs.CourseId) as tmp)

        From tblCourse cs

--42、查询不同课程成绩相同的学生的学号、课程号、学生成绩

    Select 学号=StuId, 课程号=CourseId, 成绩=Score From tblScore sc

        Where Exists (Select \* From tblScore Where Score=sc.Score And StuId=sc.StuId And CourseId <>sc.CourseId)

        Order by 学号,成绩

--43、查询每门功成绩最好的前两名

    Select 课程号=CourseId,

        第1名=(Select Top 1 StuId From tblScore Where CourseId=cs.CourseId Order by Score DESC),

        第2名=(Select Top 1 StuID From (Select Top 2 StuId,Score From tblScore Where CourseId=cs.CourseId Order by Score DESC) as tmp Order by Score)

        From tblCourse cs

--44、统计每门课程的学生选修人数（超过10人的课程才统计）。要求输出课程号和选修人数，查询结果按人数降序排列，若人数相同，按课程号升序排列

    Select 课程ID=CourseId,选修人数=(Select Count(\*) From (Select Distinct StuId From tblScore Where CourseId=cs.CourseId) as tmp)

        From tblCourse cs

        Where (Select Count(\*) From (Select Distinct StuId From tblScore Where CourseId=cs.CourseId) as tmp)>=10

        Order by 选修人数 DESC, 课程ID

--45、检索至少选修两门课程的学生学号

    Select StuId from tblScore Group by Stuid having Count(\*)>=2    --没有重复课程数据时可用此方法

    --有重复课程时用此方法(如补考)

    Select StuId from tblStudent st Where

        (Select Count(\*) From (Select Distinct CourseId From tblScore Where StuId=st.StuId) as tmp)>=2

--46、查询全部学生都选修的课程的课程号和课程名

    Select CourseId,CourseName From tblCourse cs

        Where Not Exists

        (

            Select \* From tblStudent Where StuId Not In                 --没学过本课程的学生是否存在

            (Select StuId From tblScore Where CourseId=cs.CourseId)

        )

--47、查询没学过“叶平”老师讲授的任一门课程的学生姓名

    Select StuId,StuName From tblStudent

        Where StuId Not In

            (

                Select StuID From tblScore sc

                    Inner Join tblCourse cu ON sc.CourseId=cu.CourseId

                    Inner Join tblTeacher tc ON cu.TeaId=tc.TeaId

                        Where tc.TeaName='叶平'

            )

--48、查询两门以上不及格课程的同学的学号及其平均成绩

    Select StuID as 学号,Avg(Score) as 平均成绩 From tblScore sc

        Where (Select Count(\*) From tblScore s1 Where s1.StuId=sc.StuId And Score<60)>=2

        Group By StuId

--49、检索“004”课程分数小于60，按分数降序排列的同学学号 (ok)

    Select StuID,Score From tblScore Where CourseId='004' And Score<60 Order by Score Desc

--50、删除“002”同学的“001”课程的成绩

    Delete From SC Where StuId='1002' And CourseId='001'

----------------------SC为删除数据临时表

Select \* INTO SC From tblScore

Select \* from sc Where stuId='1018'

Insert Sc(Stuid,courseId,Score) Select StuID,'009',74 From tblStudent

/\*

Navicat MySQL Data Transfer

Source Server : mysql

Source Server Version : 50541

Source Host : localhost:3306

Source Database : myschool

Target Server Type : MYSQL

Target Server Version : 50541

File Encoding : 65001

Date: 2018-04-10 17:27:20

\*/

SET FOREIGN\_KEY\_CHECKS=0;

-- ----------------------------

-- Table structure for `tblcourse`

-- ----------------------------

DROP TABLE IF EXISTS `tblcourse`;

CREATE TABLE `tblcourse` (

`CourseId` varchar(3) NOT NULL,

`CourseName` varchar(20) NOT NULL,

`TeaId` varchar(3) NOT NULL,

PRIMARY KEY (`CourseId`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- ----------------------------

-- Records of tblcourse

-- ----------------------------

INSERT INTO `tblcourse` VALUES ('001', '企业管理', '002');

INSERT INTO `tblcourse` VALUES ('002', '马克思', '008');

INSERT INTO `tblcourse` VALUES ('003', 'UML', '006');

INSERT INTO `tblcourse` VALUES ('004', '数据库', '007');

INSERT INTO `tblcourse` VALUES ('005', '逻辑电路', '006');

INSERT INTO `tblcourse` VALUES ('006', '英语', '003');

INSERT INTO `tblcourse` VALUES ('007', '电子电路', '005');

INSERT INTO `tblcourse` VALUES ('008', '毛泽东思想概论', '004');

INSERT INTO `tblcourse` VALUES ('009', '西方哲学史', '012');

INSERT INTO `tblcourse` VALUES ('010', '线性代数', '017');

INSERT INTO `tblcourse` VALUES ('011', '计算机基础', '013');

INSERT INTO `tblcourse` VALUES ('012', 'AUTO CAD制图', '015');

INSERT INTO `tblcourse` VALUES ('013', '平面设计', '011');

INSERT INTO `tblcourse` VALUES ('014', 'Flash动漫', '001');

INSERT INTO `tblcourse` VALUES ('015', 'Java开发', '009');

INSERT INTO `tblcourse` VALUES ('016', 'C#基础', '002');

INSERT INTO `tblcourse` VALUES ('017', 'Oracl数据库原理', '010');

-- ----------------------------

-- Table structure for `tblscore`

-- ----------------------------

DROP TABLE IF EXISTS `tblscore`;

CREATE TABLE `tblscore` (

`StuId` varchar(5) NOT NULL,

`CourseId` varchar(3) NOT NULL,

`Score` float DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- ----------------------------

-- Records of tblscore

-- ----------------------------

INSERT INTO `tblscore` VALUES ('1001', '003', '90');

INSERT INTO `tblscore` VALUES ('1001', '002', '87');

INSERT INTO `tblscore` VALUES ('1001', '001', '96');

INSERT INTO `tblscore` VALUES ('1001', '010', '85');

INSERT INTO `tblscore` VALUES ('1002', '003', '70');

INSERT INTO `tblscore` VALUES ('1002', '002', '87');

INSERT INTO `tblscore` VALUES ('1002', '001', '42');

INSERT INTO `tblscore` VALUES ('1002', '010', '65');

INSERT INTO `tblscore` VALUES ('1003', '006', '78');

INSERT INTO `tblscore` VALUES ('1003', '003', '70');

INSERT INTO `tblscore` VALUES ('1003', '005', '70');

INSERT INTO `tblscore` VALUES ('1003', '001', '32');

INSERT INTO `tblscore` VALUES ('1003', '010', '85');

INSERT INTO `tblscore` VALUES ('1003', '011', '21');

INSERT INTO `tblscore` VALUES ('1004', '007', '90');

INSERT INTO `tblscore` VALUES ('1004', '002', '87');

INSERT INTO `tblscore` VALUES ('1005', '001', '23');

INSERT INTO `tblscore` VALUES ('1006', '015', '85');

INSERT INTO `tblscore` VALUES ('1006', '006', '46');

INSERT INTO `tblscore` VALUES ('1006', '003', '59');

INSERT INTO `tblscore` VALUES ('1006', '004', '70');

INSERT INTO `tblscore` VALUES ('1006', '001', '99');

INSERT INTO `tblscore` VALUES ('1007', '011', '85');

INSERT INTO `tblscore` VALUES ('1007', '006', '84');

INSERT INTO `tblscore` VALUES ('1007', '003', '72');

INSERT INTO `tblscore` VALUES ('1007', '002', '87');

INSERT INTO `tblscore` VALUES ('1008', '001', '94');

INSERT INTO `tblscore` VALUES ('1008', '012', '85');

INSERT INTO `tblscore` VALUES ('1008', '006', '32');

INSERT INTO `tblscore` VALUES ('1009', '003', '90');

INSERT INTO `tblscore` VALUES ('1009', '002', '82');

INSERT INTO `tblscore` VALUES ('1009', '001', '96');

INSERT INTO `tblscore` VALUES ('1009', '010', '82');

INSERT INTO `tblscore` VALUES ('1009', '008', '92');

INSERT INTO `tblscore` VALUES ('1010', '003', '90');

INSERT INTO `tblscore` VALUES ('1010', '002', '87');

INSERT INTO `tblscore` VALUES ('1010', '001', '96');

INSERT INTO `tblscore` VALUES ('1011', '009', '24');

INSERT INTO `tblscore` VALUES ('1011', '009', '25');

INSERT INTO `tblscore` VALUES ('1012', '003', '30');

INSERT INTO `tblscore` VALUES ('1013', '002', '37');

INSERT INTO `tblscore` VALUES ('1013', '001', '16');

INSERT INTO `tblscore` VALUES ('1013', '007', '55');

INSERT INTO `tblscore` VALUES ('1013', '006', '42');

INSERT INTO `tblscore` VALUES ('1013', '012', '34');

INSERT INTO `tblscore` VALUES ('1000', '004', '16');

INSERT INTO `tblscore` VALUES ('1002', '004', '55');

INSERT INTO `tblscore` VALUES ('1004', '004', '42');

INSERT INTO `tblscore` VALUES ('1008', '004', '34');

INSERT INTO `tblscore` VALUES ('1013', '016', '86');

INSERT INTO `tblscore` VALUES ('1013', '016', '44');

INSERT INTO `tblscore` VALUES ('1000', '014', '75');

INSERT INTO `tblscore` VALUES ('1002', '016', '100');

INSERT INTO `tblscore` VALUES ('1004', '001', '83');

INSERT INTO `tblscore` VALUES ('1008', '013', '97');

-- ----------------------------

-- Table structure for `tblstudent`

-- ----------------------------

DROP TABLE IF EXISTS `tblstudent`;

CREATE TABLE `tblstudent` (

`StuId` varchar(5) NOT NULL,

`StuName` varchar(10) NOT NULL,

`StuAge` int(11) DEFAULT NULL,

`StuSex` char(1) NOT NULL,

PRIMARY KEY (`StuId`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- ----------------------------

-- Records of tblstudent

-- ----------------------------

INSERT INTO `tblstudent` VALUES ('1000', '张无忌', '18', '男');

INSERT INTO `tblstudent` VALUES ('1001', '周芷若', '19', '女');

INSERT INTO `tblstudent` VALUES ('1002', '杨过', '19', '男');

INSERT INTO `tblstudent` VALUES ('1003', '赵敏', '18', '女');

INSERT INTO `tblstudent` VALUES ('1004', '小龙女', '17', '女');

INSERT INTO `tblstudent` VALUES ('1005', '张三丰', '18', '男');

INSERT INTO `tblstudent` VALUES ('1006', '令狐冲', '19', '男');

INSERT INTO `tblstudent` VALUES ('1007', '任盈盈', '20', '女');

INSERT INTO `tblstudent` VALUES ('1008', '岳灵珊', '19', '女');

INSERT INTO `tblstudent` VALUES ('1009', '韦小宝', '18', '男');

INSERT INTO `tblstudent` VALUES ('1010', '康敏', '17', '女');

INSERT INTO `tblstudent` VALUES ('1011', '萧峰', '19', '男');

INSERT INTO `tblstudent` VALUES ('1012', '黄蓉', '18', '女');

INSERT INTO `tblstudent` VALUES ('1013', '郭靖', '19', '男');

INSERT INTO `tblstudent` VALUES ('1014', '周伯通', '19', '男');

INSERT INTO `tblstudent` VALUES ('1015', '瑛姑', '20', '女');

INSERT INTO `tblstudent` VALUES ('1016', '李秋水', '21', '女');

INSERT INTO `tblstudent` VALUES ('1017', '黄药师', '18', '男');

INSERT INTO `tblstudent` VALUES ('1018', '李莫愁', '18', '女');

INSERT INTO `tblstudent` VALUES ('1019', '冯默风', '17', '男');

INSERT INTO `tblstudent` VALUES ('1020', '王重阳', '17', '男');

INSERT INTO `tblstudent` VALUES ('1021', '郭襄', '18', '女');

-- ----------------------------

-- Table structure for `tblteacher`

-- ----------------------------

DROP TABLE IF EXISTS `tblteacher`;

CREATE TABLE `tblteacher` (

`TeaId` varchar(3) NOT NULL,

`TeaName` varchar(10) NOT NULL,

PRIMARY KEY (`TeaId`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- ----------------------------

-- Records of tblteacher

-- ----------------------------

INSERT INTO `tblteacher` VALUES ('001', '姚明');

INSERT INTO `tblteacher` VALUES ('002', '叶平');

INSERT INTO `tblteacher` VALUES ('003', '叶开');

INSERT INTO `tblteacher` VALUES ('004', '孟星魂');

INSERT INTO `tblteacher` VALUES ('005', '独孤求败');

INSERT INTO `tblteacher` VALUES ('006', '裘千仞');

INSERT INTO `tblteacher` VALUES ('007', '裘千尺');

INSERT INTO `tblteacher` VALUES ('008', '赵志敬');

INSERT INTO `tblteacher` VALUES ('009', '阿紫');

INSERT INTO `tblteacher` VALUES ('010', '郭芙蓉');

INSERT INTO `tblteacher` VALUES ('011', '佟湘玉');

INSERT INTO `tblteacher` VALUES ('012', '白展堂');

INSERT INTO `tblteacher` VALUES ('013', '吕轻侯');

INSERT INTO `tblteacher` VALUES ('014', '李大嘴');

INSERT INTO `tblteacher` VALUES ('015', '花无缺');

INSERT INTO `tblteacher` VALUES ('016', '金不换');

INSERT INTO `tblteacher` VALUES ('017', '乔丹');