# SQL基础练习 （请写出建表语句和测试结果截图）

**请根据以下描述建立数据表，并插入测试数据，如下：**

1.学生表 Student(SId,Sname,Sage,Ssex)

--SId 学生编号,Sname 学生姓名,Sage 出生年月,Ssex 学生性别

2.课程表 Course(CId,Cname,TId)

--CId --课程编号,Cname 课程名称,TId 教师编号

3.教师表 Teacher(TId,Tname)

--TId 教师编号,Tname 教师姓名

4.成绩表 SC(SId,CId,score)

--SId 学生编号,CId 课程编号,score 分数

测试数据

学生表 Student

**create** **table** Student(**SId** varchar(10),Sname varchar(10),Sage datetime,Ssex varchar(10));

**insert** **into** Student **values**('01' , '赵雷' , '1990-01-01' , '男');

**insert** **into** Student **values**('02' , '钱电' , '1990-12-21' , '男');

**insert** **into** Student **values**('03' , '孙风' , '1990-05-20' , '男');

**insert** **into** Student **values**('04' , '李云' , '1990-08-06' , '男');

**insert** **into** Student **values**('05' , '周梅' , '1991-12-01' , '女');

**insert** **into** Student **values**('06' , '吴兰' , '1992-03-01' , '女');

**insert** **into** Student **values**('07' , '郑竹' , '1989-07-01' , '女');

**insert** **into** Student **values**('09' , '张三' , '2017-12-20' , '女');

**insert** **into** Student **values**('10' , '李四' , '2017-12-25' , '女');

**insert** **into** Student **values**('11' , '李四' , '2017-12-30' , '女');

**insert** **into** Student **values**('12' , '赵六' , '2017-01-01' , '女');

**insert** **into** Student **values**('13' , '孙七' , '2018-01-01' , '女');

科目表 Course

**create** **table** Course(CId varchar(10),Cname **nvarchar**(10),TId varchar(10));

**insert** **into** Course **values**('01' , '语文' , '02');

**insert** **into** Course **values**('02' , '数学' , '01');

**insert** **into** Course **values**('03' , '英语' , '03');

教师表 Teacher

**create** **table** Teacher(TId varchar(10),Tname varchar(10));

**insert** **into** Teacher **values**('01' , '张三');

**insert** **into** Teacher **values**('02' , '李四');

**insert** **into** Teacher **values**('03' , '王五');

成绩表 SC

**create** **table** SC(**SId** varchar(10),CId varchar(10),score decimal(18,1));

**insert** **into** SC **values**('01' , '01' , 80);

**insert** **into** SC **values**('01' , '02' , 90);

**insert** **into** SC **values**('01' , '03' , 99);

**insert** **into** SC **values**('02' , '01' , 70);

**insert** **into** SC **values**('02' , '02' , 60);

**insert** **into** SC **values**('02' , '03' , 80);

**insert** **into** SC **values**('03' , '01' , 80);

**insert** **into** SC **values**('03' , '02' , 80);

**insert** **into** SC **values**('03' , '03' , 80);

**insert** **into** SC **values**('04' , '01' , 50);

**insert** **into** SC **values**('04' , '02' , 30);

**insert** **into** SC **values**('04' , '03' , 20);

**insert** **into** SC **values**('05' , '01' , 76);

**insert** **into** SC **values**('05' , '02' , 87);

**insert** **into** SC **values**('06' , '01' , 31);

**insert** **into** SC **values**('06' , '03' , 34);

**insert** **into** SC **values**('07' , '02' , 89);

**insert** **into** SC **values**('07' , '03' , 98);

**使用上述数据表，练习以下题目，如下：**

1. 查询平均成绩大于等于 60 分的同学的学生编号和学生姓名和平均成绩。

**select** student.\*,t1.avgscore

**from** student **inner** **JOIN**(

**select** sc.SId ,**AVG**(sc.score)**as** avgscore

**from** sc

**GROUP** **BY** sc.SId

**HAVING** **AVG**(sc.score)>=60)**as** t1 **on** student.SId=t1.SId

1. 查询在 SC 表存在成绩的学生信息。

**select** **DISTINCT** student.\*

**from** student ,sc

**where** student.SId=sc.SId

1. 查询「李」姓老师的数量。

**select** **count**(\*)

**from** teacher

**where** teacher.Tname **like** '李%

1. 查询学过「张三」老师授课的同学的信息。

**select** student.\*

**from** teacher ,course ,student,sc

**where** teacher.Tname='张三'

**and** teacher.TId=course.TId

**and** course.CId=sc.CId

**and** sc.SId=student.SId

1. 查询没有学全所有课程的同学的信息。

* 解法1

**select** student.\*

**from** sc ,student

**where** sc.SId=student.SId

**GROUP** **BY** sc.SId

**Having** **count**(\*)<(**select** **count**(\*) **from** course)

**但这种解法得出来的结果不包括什么课都没选的同学。**

* 解法2

**select** **DISTINCT** student.\*

**from**

(**select** student.SId,course.CId

**from** student,course ) **as** t1 **LEFT** **JOIN** (**SELECT** sc.SId,sc.CId **from** sc)**as** t2 **on** t1.SId=t2.SId **and** t1.CId=t2.CId,student

**where** t2.SId **is** null

**and** t1.SId=student.SId

**利用笛卡尔积可以把什么课都没选的同学查询出来**

1. 查询至少有一门课与学号为" 01 "的同学所学相同的同学的信息。

select DISTINCT student.\*

from sc ,student

where sc.CId in (select CId from sc where sc.SId='01')

and sc.SId=student.SId

1. 查询和" 01 "号的同学学习的课程 完全相同的其他同学的信息。

**select** **DISTINCT** student.\*

**from** (

**select** student.SId,t.CId

**from** student ,(**select** sc.CId **from** sc **where** sc.SId='01') **as** t) **as** t1 **LEFT** **JOIN** sc **on** t1.SId=sc.SId **and** t1.CId=sc.CId,student

**where** sc.SId **is** null

**and** t1.SId=student.SId

1. 查询没学过"张三"老师讲授的任一门课程的学生姓名。

**select** \*

**from** student

**where** student.SId **not** **in**

(

**select** student.SId

**from** student **left** **join** sc **on** student.SId=sc.SId

**where** **EXISTS**

(**select** \*

**from** teacher ,course

**where** teacher.Tname='张三'

**and** teacher.TId=course.TId

**and** course.CId=sc.CId))

1. 查询两门及其以上不及格课程的同学的学号，姓名及其平均成绩。

**select** student.SId,student.Sname,**avg**(sc.score)

**from** student ,sc

**where** student.SId=sc.SId

**and** sc.score<60

**GROUP** **BY** sc.SId

**HAVING** **count**(\*)>=2

1. 统计各科成绩各分数段人数：课程编号，课程名称，[100-85]，[85-70]，[70-60]，[60-0] 及所占百分比。

select course.CId,course.Cname,t1.\*

from course LEFT JOIN (

select sc.CId,CONCAT(sum(case when sc.score>=85 and sc.score<=100 then 1 else 0 end )/count(\*)\*100,'%') as '[85-100]',

CONCAT(sum(case when sc.score>=70 and sc.score<85 then 1 else 0 end )/count(\*)\*100,'%') as '[70-85)',

CONCAT(sum(case when sc.score>=60 and sc.score<70 then 1 else 0 end )/count(\*)\*100,'%') as '[60-70)',

CONCAT(sum(case when sc.score>=0 and sc.score<60 then 1 else 0 end )/count(\*)\*100,'%') as '[0-60)'

from sc

GROUP BY sc.CId) as t1 on course.CId=t1.CId

1. 求每门课程的学生人数。

**select** sc.CId,**count**(\*) **as** 学生人数

**from** sc

**GROUP** **BY** sc.CId

1. 查询不同课程成绩相同的学生的学生编号、课程编号、学生成绩。

**select** \*

**from** sc **as** t1

**where** **exists**(**select** \* **from** sc **as** t2 **where** t1.SId=t2.SId **and** t1.CId!=t2.CId **and** t1.score =t2.score )

1. 查询每门功成绩最好的前两名。

**select** \*

**from** sc **as** t1

**where** (**select** **count**(\*) **from** sc **as** t2 **where** t1.CId=t2.CId **and** t2.score >t1.score)<2

**ORDER** **BY** t1.CId

1. 检索至少选修两门课程的学生学号。

**select** **DISTINCT** t1.SId

**from** sc **as** t1

**where** (**select** **count**(\* )**from** sc **where** t1.SId=sc.SId)>=3

1. 查询选修了全部课程的学生信息。

**select** student.\*

**from** sc ,student

**where** sc.SId=student.SId

**GROUP** **BY** sc.SId

**HAVING** **count**(\*) = (**select** **DISTINCT** **count**(\*) **from** course )

# 存储过程及函数练习

1. 创建一个存储过程，以学生编号为参数，输出该学生的年龄。

**create** or replace **procedure** p\_sxt1(v\_empno in emp.empno%type, v\_sal **out** emp.sal%type) **is**

**begin**

**select** sal **into** v\_sal **from** emp **where** empno = v\_empno;

**end**;

--（1）执行

**declare**

v\_empno emp.empno%type := 7369;

v\_sal emp.sal%type;

**begin**

p\_sxt1(v\_empno,v\_sal);

dbms\_output.put\_line(v\_empno || ' 员工的工资为：' || v\_sal);

**end**;

创建一个存储过程，以员工号为参数，输出该员工的工资

--Mysql方法：

DELIMITER //

CREATE PROCEDURE myproc(IN id varchar(10),OUT Sage datetime)

BEGIN

SELECT Student .Sage INTO Sage FROM Student where Sid = id;

END

//

DELIMITER ;

调用：

SET @p\_id = ‘01’;

SET @p\_datetime = now();

CALL myproc(@p\_id, @p\_datetime);

SELECT @p\_datetime;

1. 创建一个存储过程，以课程编号为参数，输出不及格的学生信息。

**create** or replace **procedure** p\_sxt3(v\_empno in emp.empno%type, v\_year **out** number) **is**

**begin**

**select** round((sysdate - hiredate)/365,1) **into** v\_year **from** emp **where** empno = v\_empno;

**end**;

--（3）执行

**declare**

 v\_empno emp.empno%type := 7369;

v\_year number;

**begin**

 p\_sxt3(v\_empno,v\_year);

 dbms\_output.put\_line(v\_empno || ' 工作年限为 ' || v\_year || '年');

**end**;

创建一个存储过程，以员工号为参数，返回该员工的工作年限（以参数形式返回）。

1. 创建一个函数，以课程编号为参数，返回该课程的平均成绩。

**create** or replace **function** f\_sxt6(v\_deptno emp.deptno%type) **return** emp.sal%type **is**

vr\_sal emp.sal%type;

**begin**

**select** avg(sal) **into** vr\_sal **from** emp **where** deptno = v\_deptno;

**return** vr\_sal;

**end**;

--（6）执行

**select** f\_sxt6(20) 部门平均工资 **from** dual;

创建一个函数，以部门号为参数，返回该部门的平均工资。

--Mysql方法：

1. DELIMITER //
2. **CREATE FUNCTION GetAvgScoreByID(cid VARCHAR(10))**
3. **RETURNS**decimal(18,1)
4. **BEGIN**
5. **RETURN(SELECT avg(score) FROM course WHERE CId=cid);**
6. **END//**
7. DELIMITER ;

调用:

SET @cid =’01’;

SELECT **GetAvgScoreByID(**@cid**);**

# 触发器练习

1. 创建一个行级触发器CASCADE\_DEL\_UPD，当删除课程表中某门课程时，成绩表中该门课程的成绩也一并删除。

--Mysql方法：

DELIMITER ||

CREATE TRIGGER demo BEFORE DELETE

ON Course FOR EACH ROW

BEGIN

If deleting then

delete from SC where CId = OLD.CId

END

||

DELIMITER ;

create or replace trigger CASCADE\_DEL\_UPD

before update of deptno or delete

on dept

for each row

declare

begin

if deleting then

delete from emp where deptno=:old.deptno;

end if;

end;