# 5.2-2 怎样管理和利用数据（SQL应用）

1、 INSERT INTO Goods(Name，Storage，Price) VALUES(’Keyboard’，3000，90.00)的作用是\_\_\_\_\_。

(A)添加数据到一行中的所有列；

(B)插人默认值；

(C)添加数据到一行中的部分列；

(D)插入多个行；

**2**. 已知如下关系：学生Student(S#, Sname)，课程Course(C#, Cname, T#)，选课SC(S#, C#, Score)，教师T(T#, Tname)。其中S#为学号，Sname为学生命名，C#为课号，Cname为课程名，T#为教师编号，Score为成绩，Tname为教师名。

/\*下列题目考查学生阅读SQL语句的能力

(1)阅读SQL语句

SELECT S# FROM SC WHERE C# = ‘C01’ OR C# =‘C02’；

关于其查询结果，下列说法正确的是\_\_\_\_\_ 。

(A) 既学过C01号课程，又学过C02号课程的所有学生的学号；

(B) 或者学过C01号课程，或者学过C02号课程的所有学生的学号；

(C) 既未学过C01号课程，又未学过C02号课程的所有学生的学号；

(D) 或者未学过C01号课程，或者未学过C02号课程的所有学生的学号；

(E) 以上说法都不正确；

(2\*)阅读SQL语句

SELECT S# FROM SC WHERE C# <> ‘C01’ OR C# <> ‘C02’；

关于其查询结果，下列说法正确的是\_\_\_\_\_ 。

(A) 既学过C01号课程，又学过C02号课程的所有学生的学号；

(B) 或者学过C01号课程，或者学过C02号课程的所有学生的学号；

(C) 既未学过C01号课程，又未学过C02号课程的所有学生的学号；

(D) 或者未学过C01号课程，或者未学过C02号课程的所有学生的学号；

(E) 以上说法都不正确；

(3)阅读SQL语句

SELECT Sname FROM Student, SC, Course WHERE Student.S#=SC.S# AND

Course.C#=SC.C# AND T#= ‘T01’ AND Score<60；

关于其查询结果，下列说法正确的是\_\_\_\_\_ 。

(A) 学习T01号教师讲授课程不及格的所有学生的姓名；

(B) 学习T01号教师讲授课程及格的所有学生的姓名；

(C) 学习非T01号教师讲授课程不及格的所有学生的姓名；

(D) 课程不及格的所有学生的姓名；

(E) 以上说法都不正确；

(10)阅读SQL语句

SELECT AVG(Score) FROM Course, SC WHERE Course.C#=SC.C#；

关于其查询结果，下列说法正确的是\_\_\_\_\_ 。

(A) 按每个学生统计其所学课程的平均成绩；

(B) 按每门课程统计平均成绩；

(C) 计算所有学生所有课程的平均成绩；

(D) 前述(A)和(B)；

(E) 以上说法都不正确；

3**、**现有关系数据库如下： 学生（学号，姓名，性别，专业、奖学金），课程（课程号，课程名，学分），选课（学号，课程号，分数）， 用SQL语言实现下列(1)—(4)小题：

(1) 检索没有获得奖学金、同时至少有一门课程成绩在95分以上的学生信息，包括学号、姓名和专业，正确的SQL语句是\_\_\_\_\_ 。

(A)SELECT 学号，姓名，专业  FROM 学生，课程，选课

     WHERE 学生.学号=选课.学号 AND课程.课程号=选课.课程号 AND 奖学金<=0 AND 分数>95；

(B)SELECT 学号，姓名，专业 FROM 学生，选课

 WHERE 学生.学号=选课.学号  课程.课程号=选课.课程号  奖学金<=0  分数>95；

(C)SELECT 学号，姓名，专业  FROM 学生，课程，选课

 WHERE 学生.学号=选课.学号 OR 课程.课程号=选课.课程号 AND 奖学金<=0 AND 分数>95；

(D)SELECT 学号，姓名，专业  FROM 学生，课程，选课

      WHERE 奖学金<=0 AND 分数>95；

4、如下表所示，用SQL语言实现下列操作。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 学生 | | | | |  | | 表 课程 | | | | | | | |
| 学号 | 姓名 | | 年级 | 专业 |  | | | 课号 | 课名 | 开课教师 | | | 总学时 | 学分 |
| 890237 | 陈莉 | | 89 | 软件 |  | | | 1001 | 程序设计 | 王一唯 | | | 80 | 4 |
| 902783 | 李玉刚 | | 90 | 应用 |  | | | 1002 | 汇编语言 | 刘锋 | | | 80 | 4 |
| 903829 | 王磊 | | 90 | 软件 |  | | | 2001 | 数据库 | 徐伟 | | | 60 | 3 |
| 918327 | 刘玉 | | 91 | 应用 |  | | | 2002 | 人工智能 | 张再生 | | | 60 | 3 |
| 表 必修课 | | | | | |  | | 表 选课 | | | | | | |
| 课号 | | 必修专业 | | | |  | | 学号 | | | 课号 | 成绩 | | |
| 1001 | | 软件 | | | |  | | 890237 | | | 1001 | 85 | | |
| 1001 | | 应用 | | | |  | | 890237 | | | 1002 | 78 | | |
| 1002 | | 软件 | | | |  | | 890237 | | | 2002 | 75 | | |
| 2001 | | 软件 | | | |  | | 902783 | | | 1001 | 72 | | |
| 2001 | | 应用 | | | |  | | 902783 | | | 2001 |  | | |
| 2002 | | 应用 | | | |  | | 903829 | | | 1001 | 82 | | |
|  | |  | | | |  | | 903829 | | | 1002 | 83 | | |
|  | |  | | | |  | | 918327 | | | 1001 | 87 | | |

(1)定义“课程”表：课程（课号，课名，开课教师，总学时，学分），并使属性“课号”不允许空值，正确的SQL语句是\_\_\_\_\_ 。

(A)CREATE TABLE 课程(课号 char(4) not null，课名 char(10)，开课教师 char(10)，总学时 　char(3)，学分　smallint)；

(B)CREATE 课程(课号 char(4) not null，课名 char(10)，开课教师 char(10)，总学时 char(3)，学分smallint)；

(C)CREATE TABLE (课号 char(4) not null，课名 char(10)，开课教师 char(10)，总学时 char(3)，学分　smallint)；

(D)以上都不对；

(2) 向“课程”表中插入一条记录的数据。课号-2003，课名-操作系统，开课教师-刘卫，总学时-60，学分-4，正确的SQL语句是\_\_\_\_\_ 。

(A) INSERT 课程 VALUES (‘2003’，‘操作系统’，‘刘卫’，‘60’ ，4)；

(B) INSERT INTO 课程 VALUES (‘2003’，‘操作系统’，‘刘卫’，‘60’ ，4)；

(C) INSERT INTO 课程　( ‘2003’，‘操作系统’，‘刘卫’，‘60’ ，4)；

(D) INSERT INTO 课程 VALUES (‘2003’，操作系统，‘刘卫’，60 ，4)；

(3) 求既学过“1001”号课，又学过“2002”号课的所有学生的学号。正确的SQL语句是\_\_\_\_\_ 。

(A) SELECT DISTINCT 学号　FROM 选课　WHERE 课号=‘1001’ AND 课号=‘2002’ ；

(B) SELECT 　DISTINCT 学号　FROM 选课　WHERE 课号＝‘1001’ AND

学号 IN　(FROM 选课　WHERE 课号＝‘2002’ )；

(C) SELECT DISTINCT 学号　FROM 选课　WHERE 课号＝‘1001’ AND

学号　(SELECT DISTINCT 学号　FROM 选课 WHERE 课号＝‘2002’ )；

(D) SELECT DISTINCT 学号　FROM 选课　WHERE 课号＝‘1001’ AND

学号 IN　(SELECT DISTINCT 学号　FROM 选课　WHERE 课号＝‘2002’ )；

(4) 列出各门课的平均成绩、最高成绩、最低成绩和选课人数。正确的SQL语句是\_\_\_\_\_ 。

(A) SELECT 课号，AVG(成绩)，MAX(成绩)，MIN(成绩)，COUNT(学号)

FROM 选课　GROUP BY 课号；

(B) SELECT 课号，AVG(成绩)，MAX(成绩)，MIN(成绩)，COUNT(学号)

FROM 选课　ORDER BY 课号；

(C) SELECT 课号，AVG(成绩)，MAX(成绩)，MIN(成绩)，COUNT(学号)

FROM 选课　WHERE 课号；

(D) SELECT 课号，AVG(成绩)，MAX(成绩)，MIN(成绩)，COUNT(学号)

FROM 选课　HAVING 课号；

(5)列出最少选修了三门课程的学生姓名。正确的SQL语句是\_\_\_\_\_ 。

(A) SELECT 姓名　FROM 学生　WHERE 学号 IN

(SELECT 学号　FROM 选课　GROUP BY 学号　HAVING COUNT(\*) >= 3)；

(B) SELECT 姓名　FROM 学生　WHERE 学号 IN

(SELECT 学号　FROM 选课　GROUP BY 学号　COUNT(\*) >= 3)；

(C) SELECT 姓名　FROM 学生　WHERE 学号

(SELECT 学号　FROM 选课　GROUP BY 学号　HAVING COUNT(\*) >= 3)；

(D) SELECT 姓名　FROM 学生　WHERE 学号 IN

(SELECT 学号　FROM 选课　ORDER BY 学号　HAVING COUNT(\*) >= 3)；

19、按原始递归的定义，h是由f和g递归地构造出来的。假设已知h(n) = n!，请给出构造h的f和g的函数。正确的是\_\_\_\_\_。

(A) f()是常数为1的函数；g(x1,x2) = x1\* x2。

(B) f()是常数为1的函数；g(x1,x2) = x1\* (x2+1)。

(C) f()是常数为1的函数；g(x1,x2) = (x1+1)\*(x2+1)。

(D) f()是常数为1的函数；g(x1) = n \* (x1)。

答案：B

21、已知f(x)=5，g(x1,x2,x3)=x1, 其中x,x1,x2,x3均为自然数，新函数h可递归的构造如下：h(0,x) = f(x), 且h(S(n), x) = g(h(n,x),n,x)，请按递归式进行计算下列式子，正确的是\_\_\_\_\_。

(A) h(1，x) = 5；

(B) h(2，x) = 5+x；

(C) h(3，x) = 5+2x；

(D) h(4，x) = 5+3x ；

(E)上述都不正确。

答案：A