**触发器**

注意点：i. 注意触发时机：当需要修改数据库修改后的NEW值时使用before，其余情况均使用after

after是先完成数据的增删改、再触发，无法影响前面的增删改操作；

before是先完成触发，再进行增删改，可以修改影响即将发生的操作

ii. for each statement / for each row :根据题干描述

iii.注意referencing NEW/OLD 取决于题干对于什么数据操作

iv.一条insert语句中value值可以为多个元组

例1：对Reserves表的Insert操作进行监控，如果插入元组的外键属性在Sailors和Boats表中不存在，则回卷

答1: Create Trigger InsertUpdate------------------------创建触发器

Before Insert on Reserves--------------------------触发时机选择

Referencing New as N-------------------------------给新值起别名

For Each Row------------------------------------------对每条元组触发

When (not (exists(Select \* From Sailors Where sid = N.sid) and (exists(Select \* From Boats Where bid = N.bid)))-----condition

Rollback;-------------------------------------------------action

例2：创建触发器，对Boats表的Delete操作进行监控，如果删除元组的主键是Reserves表中的外键,回卷删除该记录的操作。

答2:Create trigger boats\_delete

Before Delete on Boats

Referencing OLD as O

For Each Row

When (exists(Select \* From Reserves Where bid = O.bid))

Rollback;

例3：监视Sailors表上的Insert操作，对每条Insert语句，判断其插入后的元组是否有年龄小于18的水手，将这样的水手自动插入到YoungSailors表中（YoungSailors表与Sailors表的模式相同）

答：Create Trigger Sailors\_Insert

After Insert On Saillors

Referencing New as N

For each Statement

Insert

Into YoungSailors(sid,sname,age,rating)

Select sid,sname,age,rating

From N

Where N.age<18-------------------------------将condition和action结合在一起

例4：对每条 Insert 语句，若插入的记录没有退房日期，则回卷该操作----------一条insert语句可以插入多个元组

答：create Trigger Check\_Insert

Before Insert On Check

Referencing New as N

For each statement

When null in ( select outDate from N)

Rollback;

**嵌入式SQL查询示例：**

问题背景：利用嵌入式SQL的C语言打印报表，内容是水手级别、该级别水手人数

EXEC SQL INCLUDE SQLCA; // 声明SQL通信区------必有

// 定义宿主变量-----根据题目要求定义

EXEC SQL BEGIN DECLARE SECTION;

float RATING;

int NUM;

EXEC SQL END DECLARE SECTION;

// 连接数据库--------必有

EXEC SQL CONNECT :uid IDENTIFIED BY :pwd;

// 使用游标读取数据-------必有

EXEC SQL DECLARE C1 CURSOR FOR

SELECT rating, COUNT(sid)

FROM Sailors

GROUP BY rating

EXEC SQL OPEN C1;

while (TRUE)

{

EXEC SQL FETCH C1 INTO :RATING, :NUM;

if (SQLCA.SQLCODE == 10)break;//控制rating

if (SQLCA.SQLCODE < 0)break;//查询出错

（循环打印数据）

……

}

EXEC SQL CLOSE C1;

**熟悉SQL中group by和having结合使用，不要总想着建临时表**

编写一个 SQL 查询，查找 Person 表中所有重复的电子邮箱(表结构：| Id | Email |）

SELECT Email

FROM Person

GROUP BY Email

Having Count(Email)>1

**利用DATEDIFF函数---比较Date类型大小**

DATEDIFF('2007-12-31','2007-12-30'); # 1

DATEDIFF('2010-12-30','2010-12-31'); # -1

**嵌套查询**

1. **查询与”张三“在同一个系学习的学生学号、姓名、系别**
2. 方法一：Select SNO,SNAME,SDEPT

FROM S

WHERE SDEPT IN (

SELECT SDEPT

FROM S

WHERE SNAME=“张三”

）；

方法二： SELECT Sno，Sname，Sdept

FROM S AS S1

WHERE EXISTS

（SELECT \*

FROM S AS S2

WHERE S2.Sdept=S1. Sdept AND S2.Sname=‘张三’）；

**查询选修了C2课程的学生姓名。**

1. SELECT Sname

FROM S

WHERE Sno IN

（ SELECT Sno

FROM SC

WHERE Cno=‘C2’);

ii) SELECT Sname

FROM S

WHERE EXISTS（

SELECT \*

FROM SC

WHERE SC.Sno=S.Sno AND Cno=‘C2’);

**查询所有未选修C2课程的学生姓名。**

SELECT Sname

FROM S

WHERE NOT EXISTS

（ SELECT \*

FROM SC

WHERE SC.Sno=S.Sno AND Cno=‘C2’);

**查询选修了全部课程的学生姓名。**

在表S中找学生，要求这个学生学了全部课程。换言之，在S表中找学生，在C中不存在一门课程，这个学生没有学。

SELECT Sname

FROM S

WHERE NOT EXISTS（

SELECT \* FROM C WHERE NOT EXISTS（SELECT \* FROM SC

WHERE SC.Sno=S.Sno AND SC.Cno=C.Cno）)；

**查询所学课程包含学生S3所学课程的学生学号**

分析：不存在这样的课程Y，学生S3选了Y，而其他学生没有选。

SELECT DISTINCT Sno

FROM SC AS X

WHERE NOT EXISTS（

SELECT \*

FROM SC AS Y

WHERE Y.Sno=‘S3’ AND NOT EXISTS

（SELECT \* FROM SC AS Z

WHERE Z.Sno=X.Sno AND Z.Cno=Y.Cno）)；

**在from里构建子表**

选择部门工资最高的员工：得到员工姓名、部门名、工资

SELECT

    b.Name AS Department,

    a.Name AS Employee,

    c.Max\_Salary AS Salary

FROM

    Employee AS a,

    Department AS b,

    (

        SELECT DepartmentId, MAX(Salary) AS Max\_Salary

        FROM Employee

        GROUP BY DepartmentId

    ) AS c

WHERE a.DepartmentId = c.DepartmentId

    AND a.DepartmentId = b.Id

    AND a.Salary = c.Max\_Salary