

SELECT bookid as '图书编号',price as '价格'

FROM book

WHERE bookname='DB2数据库性能调整和优化'





SELECT bookid,bookname,author,price

FROM book

WHERE catid='c1'

ORDER BY price DESC limit 2



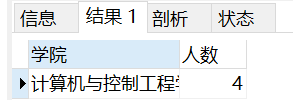


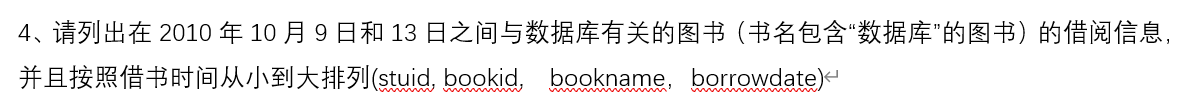
SELECT department as '学院',count(\*) as '人数'

FROM student natural join major

GROUP BY department

ORDER BY 人数 DESC limit 1



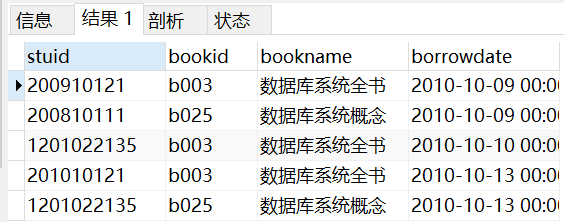


SELECT stuid,bookid,bookname,borrowdate

FROM borrow natural join book

WHERE borrowdate>= '2010-10-09 00:00:00' AND borrowdate<= '2010-10-13 23:59:59' AND bookname like '%数据库%'

ORDER BY borrowdate ASC





SELECT stuid,stuname,count(\*) AS 'borrownum'

FROM borrow natural join student

WHERE borrowdate>= '2010-10-10 00:00:00'

GROUP BY stuid

HAVING count(\*) >= all(

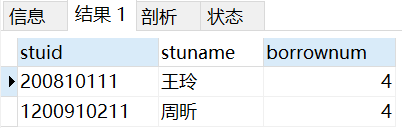
SELECT count(\*)

FROM borrow natural join student

WHERE borrowdate>= '2010-10-10 00:00:00'

GROUP BY stuid

)





SELECT bookid,bookname,author

FROM book

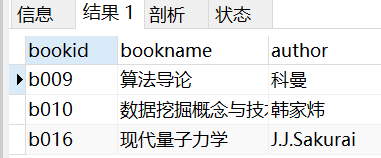
WHERE bookid not in(

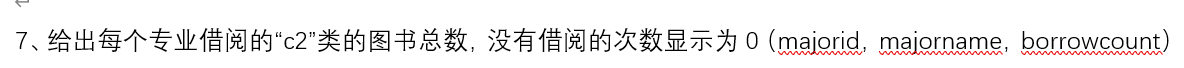
SELECT bookid

FROM borrow natural join student natural join major

WHERE degree='本科生' AND department='计算机与控制工程学院'

)





(SELECT majorid,majorname,count(\*) AS 'borrowcount'

FROM borrow NATURAL JOIN student NATURAL JOIN major NATURAL JOIN book NATURAL JOIN category

WHERE catid='c2'

GROUP BY majorid

)

UNION

(

SELECT distinct majorid,majorname,0 AS 'borrowcount'

FROM borrow NATURAL JOIN student NATURAL JOIN major NATURAL JOIN book NATURAL JOIN category

WHERE majorid NOT IN(

SELECT majorid

FROM(

SELECT majorid,majorname,count(\*) AS 'borrowcount'

FROM borrow NATURAL JOIN student NATURAL JOIN major NATURAL JOIN book NATURAL JOIN category

WHERE catid='c2'

GROUP BY majorid

) as son1

)





SELECT bookname,author

FROM book

WHERE bookid in(

SELECT bookid

FROM(

SELECT bookid,count(\*) as cnt

FROM borrow NATURAL JOIN student

WHERE degree='本科生'

GROUP BY bookid

) as son1

WHERE cnt=(

select count(distinct stuid)

FROM student

WHERE degree='本科生'

)

)

SELECT bookname,author

FROM book

WHERE not exists(

SELECT stuid FROM student WHERE degree='本科生' AND stuid not in(

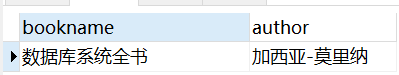
SELECT stuid

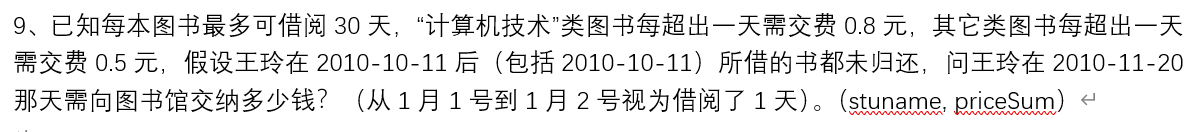
FROM student natural join borrow

WHERE borrow.bookid=book.bookid

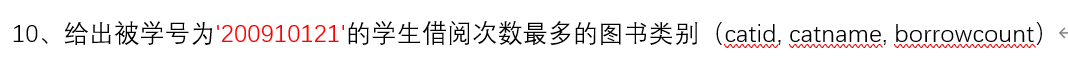
)

)





我没有写这个



SELECT catid,catname,count(\*) as 'borrowcount'

FROM borrow NATURAL JOIN book NATURAL JOIN category

WHERE stuid='200910121'

GROUP BY catid

HAVING count(\*) >= ALL(

SELECT count(\*) as 'borrowcount'

FROM borrow NATURAL JOIN book NATURAL JOIN category

WHERE stuid='200910121'

GROUP BY catid

)

1. 找出名字中包含“刘”的所有员工号和员工名称**(empno, empname)**

select empno,empname

from employee

where empname like '刘%'



1. 找出1999-01-01前参与过项目“网络布线”的员工**(empno, empname)**

select empno,empname

from employee natural join workson natural join project

where enterdate<'1999-01-01' and projectname='网络布线'



1. 统计每个部门中参与项目最少的员工

**(deptno, deptname,empno, empname, projectcount)**

select \*

from (select deptno, deptname,empno, empname, count(projectname)as projectcount from employee natural join workson natural join department natural join project group by empno) as t1

where projectcount<=all

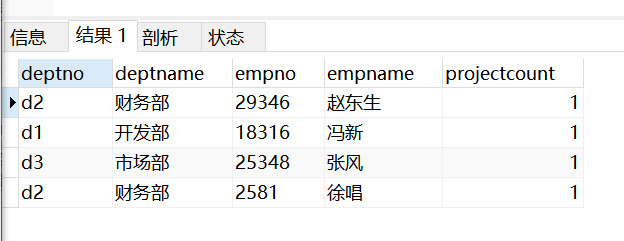
(

select projectcount

from (select deptno, deptname,empno, empname, count(projectname)as projectcount from employee natural join workson natural join department natural join project group by empno) as t2

where deptno=t1.deptno

)



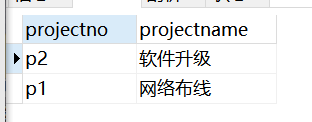
1. 找出有未安排职务(job)的员工的项目**(projectno, projectname)**

select projectno, projectname

from workson left outer join employee on workson.empno=employee.empno

natural join project

where job is null



1. 找出所在地不在“广东”且员工数不是最多的部门**(deptno,deptname,location)**

select deptno,deptname,location

from

(select deptno,deptname,location,count(empno)as num

from department natural join employee

group by deptno) as t

where location<>'广东' and num<any

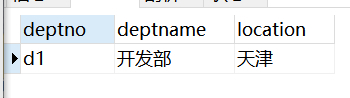
(

select count(empno)

from department natural join employee

group by deptno

)



1. 找出所有参与过两个或两个以上项目的员工**(empno, empname)**

select empno,empname

from employee natural join workson

group by empno

having count(projectno)>=2



1. 列出每个项目的员工数**(projectno, projectname, empcount)，**按照员工数由大到小排序，没有员工参与的项目，员工数显示为0，显示为NULL或不显示的不给分

本题使用一种方法且正确即可得满分，使用两种方法可加分

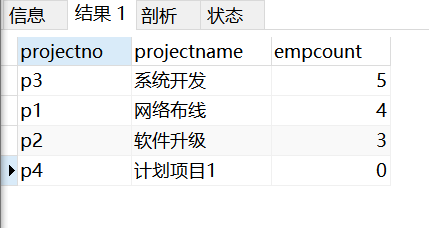
提示：方法一使用outer join；方法二使用inner join 和union

select project.projectno,projectname,count(empno) as empcount

from project left outer join workson on workson.projectno=project.projectno

group by project.projectno

order by empcount desc



1. 找出每个项目中入职时间最早的员工**(projectno, empno)**

select projectno,empname

from workson w1 natural join employee

where w1.enterdate<=all

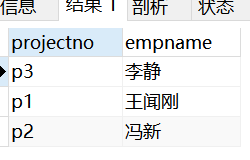
(

select enterdate

from workson

where projectno=w1.projectno

)



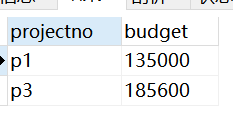
1. 找出预算超过所有项目平均预算数的项目**(projectno，budget)**

select projectno,budget

from project

where budget>all

(select avg(budget) from project)



1. 找出除了”信息技术部”外，所有部门都参与了的项目**(projectno,projectname)**

select projectno,projectname

from

(

select project.projectno,projectname,count(employee.deptno) as deptcount

from

department left outer join employee on department.deptno=employee.deptno

and department.deptname<>'信息技术部'

left outer join workson on employee.empno=workson.empno

left outer join project on project.projectno=workson.projectno

group by project.projectno

) as t1

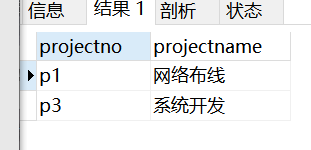
where deptcount=

(

select count(\*)

from project

)



1. 找出所有参与项目“网络布线”的员工**(empno, empname)**

select empno,empname

from employee natural join workson natural join project

where projectname='网络布线'



1. 找出只参加过一个项目的员工**(empno,empname)**

select empno,empname

from (select empno,empname,count(distinct projectno) as pnum

from employee natural join workson

group by empno) as t

where pnum=1



1. 统计每个部门中参与项目最多的员工

**(deptno, deptname, empno, empname, projectcount)**

select deptno, deptname, empno, empname,projectcount

from

(select deptno, deptname, empno, empname, count(projectno)as projectcount

from workson natural join employee natural join department

group by empno)as t1

where projectcount>=all

(

select count(projectno)

from workson natural join employee natural join department

where deptno=t1.deptno

group by empno

)



1. 找出拥有最多工作为“职员”的员工的项目**(projectno, projectname,职员人数)**

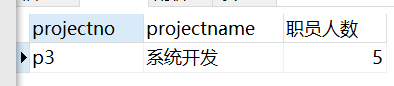
select projectno,projectname,max(empcount) as 职员人数

from

(select projectno,projectname,count(empno) as empcount

from workson natural join project

group by workson.projectno) as t



⬆⬆⬆这么写不对！！！！！！



应该写为

select projectno,projectname,empcount as 职员人数

from

(select projectno,projectname,count(empno) as empcount

from workson natural join project

where job='职员'

group by workson.projectno

) as t

where empcount>=all

(

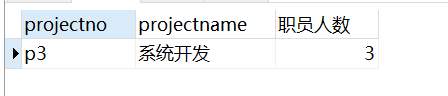
select count(empno) as empcount

from workson natural join project

where job='职员'

group by workson.projectnoa

)



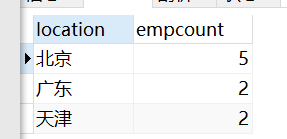
1. 统计每个地区的员工数**(location, empcount)**

select location,count(empno) as empcount

from department left outer join employee

on department.deptno=employee.deptno

group by location



1. 列出每个部门的员工数**(deptno, deptname, empcount)，**按照员工数由大到小排序，没有员工参与的部门，员工数显示为0，显示为NULL或不显示的不给分

本题使用一种方法且正确即可得满分，使用两种方法可加分

提示：方法一使用outer join；方法二使用inner join 和union

select department.deptno,deptname,count(empno) as empcount

from department left outer join employee

on employee.deptno=department.deptno

group by department.deptno

order by empcount desc



(select department.deptno,deptname,count(empno) as empcount

from department inner join employee

on employee.deptno=department.deptno

group by department.deptno

order by empcount desc)

union

(

select department.deptno,deptname,0 as 'empcount'

from department

where deptno not in

(

select deptno

from

(select deptno,count(empno)

from department natural join employee

group by deptno

having count(empno)<>0

) as t

)

)

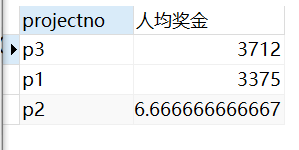


1. 假设每个项目预算的10%用来发奖金，请计算每个项目的人均奖金数**(projectno,人均奖金)**，没有员工的项目不用显示

select projectno, budget/count(empno)\*0.1 as 人均奖金

from project natural join workson

group by projectno



1. 找出1998-01-01之后参与过项目“系统开发”的员工**(empno, empname)**

select empno,empname

from employee natural join workson natural join project

where enterdate>='1998-01-02' and projectname='系统开发'



1. 找出每个项目中入职时间最晚的员工**(projectno, empno)**

select projectno,empno

from workson w

where enterdate>=all

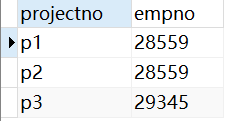
(

select enterdate

from workson

where projectno=w.projectno

)



1. 列出每个部门参与的项目数**(deptno, deptname, projectcount)，**没有参与项目的部门其项目数显示为0，显示为NULL或不显示的不给分

select department.deptno,deptname,count(distinct projectno) as projectcount

from department left outer join employee on department.deptno=employee.deptno

left outer join workson on employee.empno=workson.empno

group by department.deptno



1. 给出职工中所有男性的所有信息（empid,empname,age,sex,edpid）

select \* from Employee where sex='男'



1. 统计来自天津的“李”性职工信息，按年龄降序排序。（empid,empname,age,location）

select empid,empname,age,location from Department,Employee

where Employee.depid=Department.depid and empname like '李%'

order by age desc



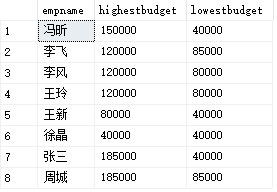
1. 给出每位职员参与项目的最高预算和最低预算（empname，highestbudget，lowestbudget）

select empname,max(budget)highestbudget,min(budget)lowestbudget

from Employee,Project,Workson

where Workson.empid=Employee.empid and Workson.proid=Project.proid

group by empname

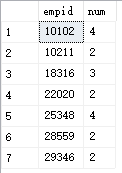


1. 给出所有项目超过一个的员工的id和参加的项目个数（empid, num）

select empid, COUNT(empid) as num from project,workson

where workson.proid=project.proid

group by empid having COUNT(empid)>1



1. 给出项目编号为“c2”且预算最多的项目。（proid，projectname，budget）

select proid,projectname,budget

from project

where budget=(

select top(1)budget from project

where catid='c2'order by budget desc

）



1. 给出参加“产品推广”项目，但不担任职位的员工的员工信息，（empid,empname，age,sex.depid）

select employee.empid,empname,age,sex,depid

from Project,Employee,Workson

where workson.proid=project.proid and projectname='产品推广' and employee.empid=workson.empid and job is NULL



1. 给出工号为“10102”的员工每类项目的参加总数，若没有参加过某类项目，则参加项目总数显示为0（catid，proNum）

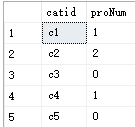
select category.catid,count(workson.proid)as'proNum'

from category left outer join project on project.catid=category.catid

left outer join workson on workson.proid=project.proid

and workson.empid='10102'

group by category.catid



1. 给出没有参与“软件类”项目女性职工的信息（empid,empname, age,sex,depid)

SELECT e.empid, e.empname, age, sex, depid

FROM Employee e

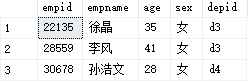
WHERE sex='女'and not exists(

SELECT \*

from project p, workson w, Category c

WHERE e.empid = w.empid and p.proid = w.proid and c.catid=p.catid and c.catname='软件类'

)



1. 给出有30岁以上男性员工的省份名称和该省男员工最大年龄，结果按最大年龄升序排序（location，maxAge）

select location,maxAge

from(

SELECT max(age) as maxAge,d.location

FROM Employee e, Department d

WHERE e.depid = d.depid and e.sex = '男'

GROUP BY d.location

)p

where maxAge>30

order by maxAge asc

C:\Users\gjw\AppData\Local\Temp\WeChat Files\30cfc61dae84e214c436cfc4a02948d.png

1. 给出在广州工作的、参加“'产品推广'”项目的职员id、姓名及他们参加的项目个数（empid, empname, procnt）

select emp.empid,emp.empname,procnt

from Workson,Department,Project,

(select Employee.empid,empname,Employee.depid,count(proid)as procnt

from employee,Workson

where Employee.empid=Workson.empid

group by Employee.empid,empname,Employee.depid

)emp

where Department.location='广州' and Department.depid=Emp.depid and Workson.proid=project.proid

and projectname='产品推广' and Workson.empid=Emp.empid

group by emp.empid,emp.empname,procnt





SELECT bookname,price,catid

FROM book

WHERE (catid='c2' OR catid='c3') AND price >=30 AND price <=60





SELECT stuid,count(bookid) as num

FROM borrow

WHERE borrowdate>='2010-10-09 00:00:00'

GROUP BY stuid

HAVING count(\*)>=ALL(

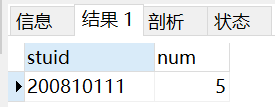
SELECT count(bookid)

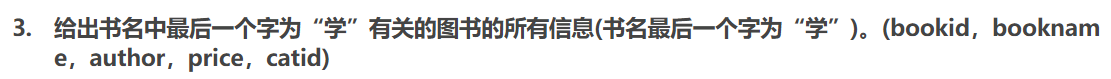
FROM borrow

WHERE borrowdate>='2010-10-09 00:00:00'

GROUP BY stuid

)





SELECT bookid,bookname,author,price,catid

FROM book

WHERE bookname like '%学'





SELECT bookid,bookname,author,price,catid

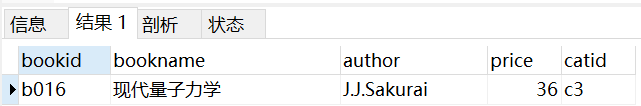
FROM book NATURAL JOIN borrow

WHERE borrowdate>= ALL(

SELECT borrowdate

FROM borrow

)





SELECT stuname,MAX(price) as highestprice,MIN(price) as lowestprice

FROM borrow NATURAL JOIN book NATURAL JOIN student

WHERE stuname like '王%'

GROUP BY stuname



Select student.stuname,max(price) as highestprice,min(price) as lowestprice

From student,borrow,book

Where student.stuid=borrow.stuid and borrow.bookid=book.bookid and stuname like ’王%’

Group by stuname



SELECT stuid,stuname

FROM student

WHERE not exists(

SELECT stuid FROM category where catid not in(

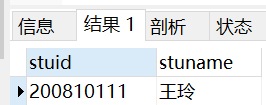
SELECT catid

FROM book NATURAL JOIN borrow

WHERE borrow.stuid=student.stuid

)

)

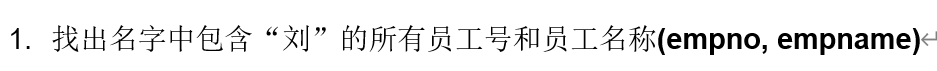


Select stuid,stuname

From student

Where

Book natural join category

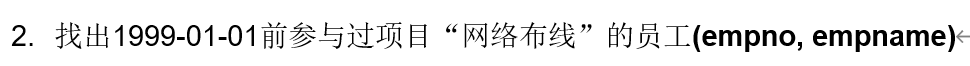


SELECT empno,empname

FROM employee

WHERE empname like '%刘%'

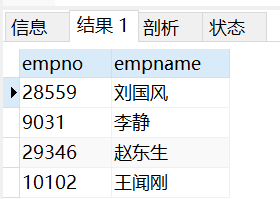


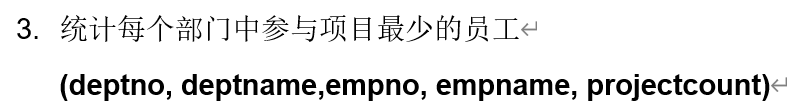


SELECT empno,empname

FROM workson natural join project natural join department natural join employee

WHERE enterdate<'1999-01-01 00:00:00' AND projectname='网络布线'





SELECT d1.deptno,d1.deptname,empno,empname,count(projectno) as projectcount

FROM workson natural join employee natural join department d1

GROUP BY empno

HAVING count(projectno)<=ALL(

SELECT count(projectno)

FROM workson natural join department d2 natural join employee

WHERE d1.deptno=d2.deptno

GROUP BY empno

)





SELECT projectno,projectname

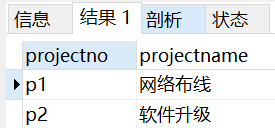
FROM(

SELECT distinct projectno

FROM workson

WHERE job IS NULL

) as son1 NATURAL JOIN project





SELECT deptno,deptname,location

FROM department natural join(

SELECT deptno

FROM employee

GROUP BY deptno

HAVING count(\*)< ANY (

SELECT count(\*)

FROM employee

GROUP BY deptno

)

) as son1

WHERE location<>'广东'



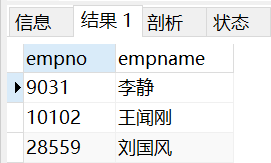


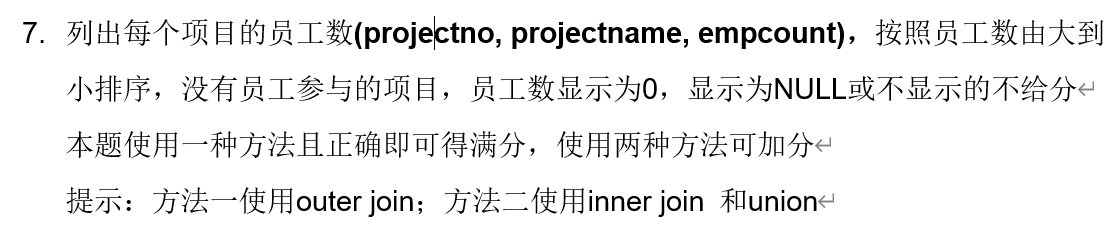
SELECT empno,empname

FROM employee natural join workson

GROUP by empno

HAVING count(projectno)>=2





SELECT project.projectno,projectname,count(empno)as empcount

FROM workson RIGHT OUTER JOIN project on workson.projectno=project.projectno

GROUP BY project.projectno

ORDER BY empcount desc

(

SELECT projectno,projectname,count(empno) as 'empcount'

FROM workson natural join project

GROUP BY projectno

ORDER BY empcount DESC)

union

(

SELECT projectno,projectname,0 as 'empcount'

FROM project

WHERE projectno not in(

SELECT projectno

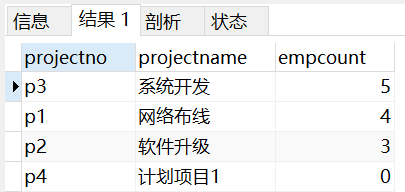
FROM workson natural join project

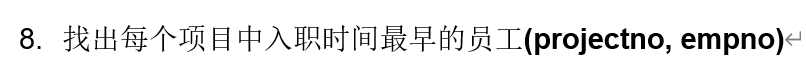
GROUP BY projectno

HAVING count(empno)>0

)

)





SELECT projectno,empno

FROM workson w1 natural join employee

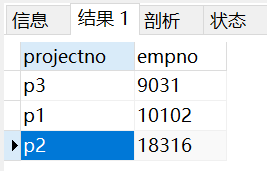
WHERE w1.enterdate<=ALL(

SELECT enterdate

FROM workson w2

WHERE w1.projectno=w2.projectno

)





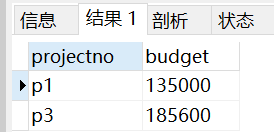
SELECT projectno,budget

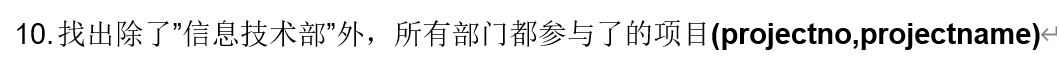
FROM project

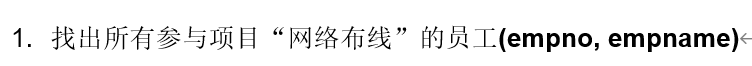
WHERE budget>(

SELECT AVG(budget)

FROM project)







SELECT empno,empname

FROM employee

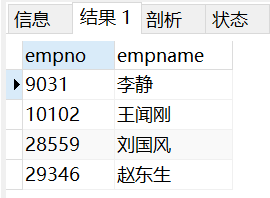
WHERE empno in(

SELECT empno

FROM workson natural join project

WHERE projectname='网络布线'

)





SELECT empno,empname

FROM workson NATURAL JOIN employee

GROUP BY empno

HAVING count(projectno)=1





SELECT projectno,projectname,count(\*) as '职员人数'

FROM(

SELECT \*

FROM workson NATURAL JOIN project

WHERE job='职员'

) as son1

GROUP BY projectno

HAVING count(\*)>= ALL(

SELECT count(\*)

FROM(

SELECT \*

FROM workson NATURAL JOIN project

WHERE job='职员'

) as son2

GROUP BY projectno

)

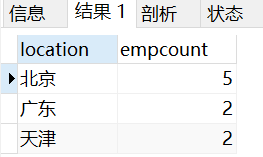


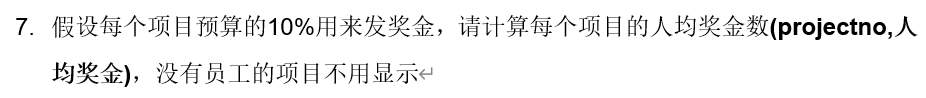


SELECT location,count(\*) AS empcount

FROM employee NATURAL JOIN department

GROUP BY location





SELECT projectno, budget/count(\*) as '人均奖金'

FROM project NATURAL JOIN workson

GROUP BY projectno





SELECT empno,empname

FROM workson NATURAL JOIN employee NATURAL JOIN project

WHERE enterdate>'1998-01-01 23:59:59' AND projectname='系统开发'





SELECT projectno,empno

FROM workson w1

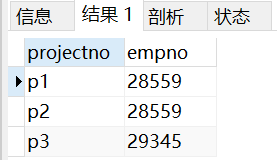
WHERE w1.enterdate>= ALL(

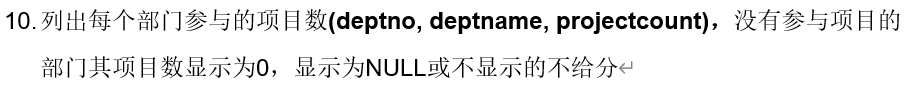
SELECT enterdate

FROM workson w2

WHERE w1.projectno=w2.projectno

)





SELECT department.deptno,deptname,count(projectno) as projectcount

FROM department

LEFT OUTER JOIN employee ON department.deptno=employee.deptno

LEFT OUTER JOIN workson ON workson.empno=employee.empno

GROUP BY deptno



1. 给出项目名称为“软件升级”的预算（projectname, budget）

select projectname, budget

from project

where projectname = '软件升级'



1. 给出姓王的所有员工编号和姓名，并按照员工编号排序（empid, empname）

select empid, empname

from employee

where empname like '王%'

order by empid



1. 统计每个项目参与的员工人数。（proid, projectname, count）

select project.proid, project.projectname,count(\*) count

from workson,project

where workson.proid=project.proid

group by project.proid,project.projectname



1. 给出部门人数超过两人的部门信息（不包括两人）。（depid, depname）

select department.depid,depname

from department,employee

where employee.depid = department.depid

group by department.depid,depname

having count(\*)>2



1. 查找没有参加过项目p6的雇员信息。（empid, empname, age, sex, depid）

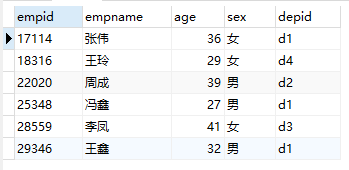
select empid, empname, age, sex, depid

from employee

where empid not in

(select employee.empid from employee,workson where employee.empid = workson.empid and proid = 'p6'

)

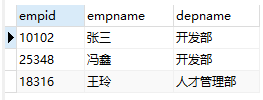


1. 找出Workson表中职业为空的员工姓名，编号和所在部门名称（empid, empname, depname）

select distinct workson.empid,empname,depname

from workson,employee,department

where workson.empid = Employee.empid and employee.depid = Department.depid and job is null



1. 给出所有女员工的信息及参与项目的个数，如果没有参与项目，显示0。（empid, empname, procount）

select Employee.empid, empname,count(proid) procount

from Employee left outer join Workson

on Employee.empid = Workson.empid

where sex = '女'

group by Employee.empid, empname



1. 给出参与的项目数最少的员工信息（empid, empname, number）

（注：可能存在多个项目参与人数数量相同的情况）

select e1.empid, e1.empname,count(p1.proid) as number

from Employee e1, Project p1, Workson w1

where e1.empid = w1.empid and p1.proid = w1.proid

group by e1.empid, e1.empname

having count(p1.proid) =

( select count(p.proid)

from Employee e, Project p, Workson w

where e.empid = w.empid and p.proid = w.proid

group by e.empid, e.empname

order by count(p.proid) asc limit 1

)



1. 给出不同类别项目参与总人数的详细信息，按照参与总人数降序排序，参与总人数相同的按照catid升序排序（catid, catname, participantcount）

select category.catid, catname, count(\*) participantcount

from category, project, workson

where category.catid = project.catid and project.proid=workson.proid

group by category.catid, catname

order by participantcount desc, catid asc



10、给出参与了除c5之外的所有类别的项目的职员信息（empid, empname）

select empid, empname from employee where not exists(

select catid from category where catid not in(

select catid from workson, project

where employee.empid=workson.empid and workson.proid=project.proid

) and catid != 'c5'

)



NATURAL JOIN、INNER JOIN和OUTER JOIN是关系型数据库中用于连接表格的不同类型的操作。

NATURAL JOIN（自然连接）：

NATURAL JOIN是一种基于两个表之间的共同列自动进行连接的操作。它会自动查找两个表中具有相同列名的列，并将它们进行匹配，然后返回匹配的结果集。NATURAL JOIN省略了ON子句，并且只返回那些在两个表中都存在的匹配行。

示例：

SELECT \*

FROM 表A

NATURAL JOIN 表B;

NATURAL JOIN将根据两个表之间的共同列自动进行连接，不需要显式指定连接条件。

INNER JOIN（内连接）：

INNER JOIN是根据指定的连接条件将两个表进行连接，并返回满足连接条件的行。它只返回那些在连接条件下存在匹配的行。

示例：

SELECT \*

FROM 表A

INNER JOIN 表B

ON 表A.列 = 表B.列;

INNER JOIN使用ON子句来指定连接条件，只返回满足连接条件的行。

OUTER JOIN（外连接）：

OUTER JOIN是一种连接操作，它返回满足连接条件的行以及不满足连接条件但至少存在于一个表中的行。OUTER JOIN可以进一步分为左外连接、右外连接和全外连接。

左外连接（LEFT OUTER JOIN）：返回左表中的所有行以及满足连接条件的右表中的匹配行。如果右表中没有匹配的行，则右表的列值为NULL。

右外连接（RIGHT OUTER JOIN）：返回右表中的所有行以及满足连接条件的左表中的匹配行。如果左表中没有匹配的行，则左表的列值为NULL。

全外连接（FULL OUTER JOIN）：返回左表和右表中的所有行，并将它们根据连接条件进行匹配。如果某个表中没有匹配的行，则对应表的列值为NULL。

示例：

SELECT \*

FROM 表A

LEFT OUTER JOIN 表B

ON 表A.列 = 表B.列;

上述示例中使用的是左外连接，可以根据需要选择左、右或全外连接。

总结：

NATURAL JOIN自动根据共同列进行连接，省略了连接条件。

INNER JOIN根据指定的连接条件返回满足条件的行。

OUTER JOIN返回满足连接条件以及不满足连接条件的行，分为左外连接、右外连接和全外连接。

数据库查询中，如果要求说，空值显示为0，查询语句应该怎么写

在数据库查询中，如果要求将空值显示为0，可以使用COALESCE函数或ISNULL函数来实现。下面是两种常用的查询语句示例：

使用COALESCE函数：

SELECT COALESCE(column\_name, 0) AS column\_alias

FROM table\_name;

在上面的查询语句中，将要查询的列名替换为column\_name，将要查询的表名替换为table\_name。COALESCE(column\_name, 0)表示如果column\_name列的值为空，则显示为0。你可以使用AS关键字为结果设置别名，将其替换为column\_alias。

使用ISNULL函数：

SELECT ISNULL(column\_name, 0) AS column\_alias

FROM table\_name;

在上述查询语句中，ISNULL(column\_name, 0)表示如果column\_name列的值为空，则显示为0。同样，你可以使用AS关键字为结果设置别名。

数据库查询中，如果要求说，空值显示为0，并且涉及到多表查询，查询语句应该怎么写

在涉及到多表查询时，如果要求将空值显示为0，你可以在查询语句中使用COALESCE函数或ISNULL函数，并结合适当的JOIN语句来连接多个表。以下是一个示例查询语句：

SELECT COALESCE(t1.column\_name, 0) AS column\_alias

FROM table1 AS t1

JOIN table2 AS t2 ON t1.id = t2.id;

在上面的查询语句中，假设你有两个表table1和table2，你需要根据实际情况调整表名和列名，并使用适当的连接条件（例如，t1.id = t2.id）。COALESCE(t1.column\_name, 0)表示如果table1中的column\_name列的值为空，则显示为0。你可以使用AS关键字为结果设置别名，将其替换为column\_alias。

请注意，根据你的实际需求和数据库结构，可能需要使用不同的JOIN类型（如INNER JOIN、LEFT JOIN等）和其他查询条件来构建更复杂的查询语句。

LEFT JOIN和RIGHT JOIN是两种常见的表连接操作，它们在连接多个表时具有不同的行为和结果。

LEFT JOIN:

LEFT JOIN 返回左边表（左侧表）的所有行，以及满足连接条件的右边表（右侧表）的匹配行。如果右边表中没有与左边表匹配的行，则结果中右侧表的列将显示为NULL。

语法: LEFT JOIN或LEFT OUTER JOIN

RIGHT JOIN:

RIGHT JOIN 返回右边表（右侧表）的所有行，以及满足连接条件的左边表（左侧表）的匹配行。如果左边表中没有与右边表匹配的行，则结果中左侧表的列将显示为NULL。

语法: RIGHT JOIN或RIGHT OUTER JOIN

使用示例：

假设我们有两个表：Customers和Orders。Customers表包含客户信息，Orders表包含订单信息。每个客户可以有多个订单，但并非每个客户都有订单。

我们可以使用LEFT JOIN和RIGHT JOIN来连接这两个表，以获取不同的结果：

使用 LEFT JOIN：

SELECT Customers.CustomerName, Orders.OrderID

FROM Customers

LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID;

上述查询会返回所有客户的姓名（来自Customers表）以及他们的订单ID（来自Orders表）。如果某个客户没有订单，对应的订单ID将显示为NULL。

使用 RIGHT JOIN：

SELECT Customers.CustomerName, Orders.OrderID

FROM Customers

RIGHT JOIN Orders ON Customers.CustomerID = Orders.CustomerID;

上述查询会返回所有订单的订单ID（来自Orders表）以及对应客户的姓名（来自Customers表）。如果某个订单没有对应的客户，客户姓名将显示为NULL。

请注意，LEFT JOIN和RIGHT JOIN可以根据你的需求和表之间的关系选择使用，以确保你获取到想要的结果。

"not exists"是一个逻辑运算符，用于在逻辑表达式中表示不存在的情况。它通常用于描述一个集合中不存在满足某个条件的元素。

以下是几个使用"not exists"的示例：

在数据库查询中使用：

假设有一个名为"Employees"的表，其中包含员工的信息。我们想要查找没有分配到任何项目的员工。可以使用"not exists"来实现：

SELECT \*

FROM Employees

WHERE NOT EXISTS (

SELECT \*

FROM Projects

WHERE Projects.EmployeeID = Employees.EmployeeID

)

上述查询将返回在"Projects"表中没有对应记录的所有员工。

在数据库查询语句中，"not exists"用于检查一个查询结果是否为空，以此判断是否存在满足特定条件的记录。它通常与子查询结合使用，子查询用于检索符合条件的数据，而"not exists"用于判断子查询的结果是否为空。

下面是一个具体的示例，假设我们有两个表：Employees（员工表）和 Projects（项目表）。我们想要找出没有被分配到任何项目的员工。

Employees 表结构：

EmployeeID | EmployeeName

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

1 | John

2 | Sarah

3 | Michael

Projects 表结构：

ProjectID | EmployeeID | ProjectName

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

1 | 1 | Project A

2 | 2 | Project B

3 | 3 | Project C

使用"not exists"的查询语句如下：

SELECT \*

FROM Employees e

WHERE NOT EXISTS (

SELECT \*

FROM Projects p

WHERE p.EmployeeID = e.EmployeeID

)

在上述查询中，我们通过子查询选择了Projects表中与Employees表关联的记录。然后，"not exists"子句检查子查询的结果是否为空。如果子查询为空，即没有匹配的项目记录，那么该员工会被包含在结果集中。

对于上述示例，查询的结果将是：

EmployeeID | EmployeeName

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

1 | John

2 | Sarah

3 | Michael

因为没有任何一个员工被分配到项目。

请注意，"not exists"是一种使用子查询的方法，它与其他方式（如使用LEFT JOIN和IS NULL）相比可能具有不同的执行效率和语义。在实际使用时，可以根据具体情况选择最合适的查询方式。

HAVING语句是在SQL中用于在对查询结果进行分组后进行筛选的条件表达式。它通常与GROUP BY子句一起使用，用于对分组后的结果进行过滤。

以下是HAVING语句的用法和示例：

假设我们有一个名为"Orders"的表，其中包含了顾客下的订单信息，包括订单号（OrderID）、顾客ID（CustomerID）和订单金额（Amount）。我们想要找到订单金额大于1000的顾客，并且只包括那些订单数量超过3个的顾客。

使用HAVING语句的查询示例：

SELECT CustomerID, COUNT(OrderID) as OrderCount

FROM Orders

GROUP BY CustomerID

HAVING Amount > 1000 AND COUNT(OrderID) > 3

在上述查询中，我们首先使用GROUP BY子句将订单按照顾客ID进行分组。然后，使用HAVING语句对分组后的结果进行筛选。在HAVING语句中，我们设置了两个条件：Amount > 1000表示订单金额大于1000，COUNT(OrderID) > 3表示订单数量大于3。只有满足这两个条件的顾客才会包含在查询结果中。

请注意，HAVING语句是在分组后对结果进行筛选，而WHERE子句是在分组前对原始数据进行筛选。HAVING语句仅用于对分组后的结果进行条件过滤。

此外，HAVING语句可以使用各种比较运算符（如>、<、=、<>）和逻辑运算符（如AND、OR）来构建复杂的条件表达式。你可以根据具体的需求来灵活运用HAVING语句进行数据筛选和分组分析。