

itpub 第四届

参考：<http://blog.itpub.net/4227/viewspace-751984/>

SQL数据库编程大赛第四期评分及所有答题公布链接：<http://www.itpub.net/thread-1417576-1-1.html>

SQL数据库编程大赛第三期评分及所有答题公布链接：<http://www.itpub.net/thread-1415335-1-1.html>

SQL数据库编程大赛第二期评分及所有答题公布链接：<http://www.itpub.net/thread-1411980-1-1.html>

SQL数据库编程大赛第一期评分及所有答题公布链接：<http://www.itpub.net/thread-1407072-1-1.html>

第一期

现在开始“盛拓传媒杯”SQL大赛第一期答题，答题时间为：2011年2月25日-3月4日24:00结束，请大家在此时间内答题。

本期题目：5X5方格棋盘难题

在5X5的方格棋盘（如图），每行、列、斜线（斜线不仅仅包括对角线）最多可以放两个球，如何摆放才能放置最多的球，这样的摆法总共有几种？输出所有的摆法。

要求：用一句SQL实现。

输出格式：从方格棋盘第一行至第5行，每行从第一列到第5列依次输出，0表示不放球，1表示放球。例如：1001000000000000000000。一行输出一个行号和一个解，按解所在的列字符串顺序从大到小排序。

数据库平台：适用Oracle、MS SQL Sever，版本（Oracle推荐10gr2（包含）以上版本、MS SQL Sever推荐2008版本）

第一期活动评委：OO, newkid, dingjun123, bq_wang, wabjtam123, lastwinner

参与规则：

每个参赛者以附件方式贴出自己的源代码，并将阅读权限设为11~100之间的数值，答案贴出后不允许编辑帖子。编辑过的帖子做无效处理。需要修改答案请发表新的回复，并以同样的方式上传新的答案。

附件名称以SQL大赛N期(ID).txt来命名，通过rar压缩加密，推荐采取select sys_guid() from dual的方式产生密码，答题者需要在截止时间前将密码以论坛PM给论坛助理`贝贝`

pm链接：<http://www.itpub.net/pm.php?action=send&uid=93405>

活动流程：

- 1、发布本期活动题目及答题时间范围，大家可以在此期间根据以上参与规则上传自己的参赛程序文档。
- 2、结束答题，进入评审阶段，评审将在三天内对所有参赛程序进行评审，并给予点评。
- 3、公布评审结果及部分作品点评，并开始下一期竞赛。

评审规则：

- 1、由ITPUB论坛助理下载参赛者附件并解压后删除参赛者ID, 将各参赛文档重编序号发布给当期评委。
- 2、评委抽签获得序号并评审相应序号的答案。运行程序，将运行出正确答案的参赛文档选出。
- 3、所有评委独立对运行出正确答案的程序打分，把相应分数告知论坛助理。论坛助理取平均数为该答案最后得分。

注：出题者负责向评委讲解思路，并配合评委书写点评。

评审依据：

1. SQL、PL/SQL代码量(按单词)，通俗易懂，可读性(10分)（禁止在代码中包含参赛者ID、昵称等可能标识参赛者身份的信息）
 2. 解决问题的思路(10分)
 3. 解决问题的正确性和完善程度(40分)
 4. 性能要求(20分)
 5. 通用性(10分)比如输入变量
 6. 可扩展性(10分)比如扩大数据量、增加数据类别
- 具体标准可根据当期赛题进行适当调整。

特别声明：

出题者和评委需严格保守题目秘密，禁止外泄。禁止评委及同一会员用多个马甲同时参与此次活动，禁止其他一切有违比赛公平的事项。违者一经查实，将取消相关奖励以及比赛参与资格。

第二期

本期题目：

为了便于大家答题和测试,对本期题目做如下修改

1. 问题1完全列举各种旋转变形
2. 问题2将N的范围改为 $5 \leq N \leq 6$
3. 增加解题提示

本期题目：

问题1(80分)：

在一个6*6的棋盘上，放置12个球，每行、每列、每个正负45度的斜线上最多放置2个球，请用一个SQL语句（不可以用PL/SQL或T-SQL匿名块、过程或函数，也不可以用[Java](#)等外部语言）求出不“重复”的摆法的个数(剔除上下对称、左右对称、中心对称、沿中心点旋转等各种变形)

如下是4*4矩阵的几种变形

原始排列

A B C D
E F G H
I J K L
M N O P

左右翻转

D C B A
H G F E
L K J I
P O N M

上下翻转

M N O P
I J K L
E F G H
A B C D

中心旋转180度

P O N M
L K J I
H G F E
D C B A

沿左上右下对角线翻转

A E I M
B F J N
C G K O
D H L P

沿左下右上对角线翻转

P L H D
O K G C
N J F B
M I E A

中心顺时针旋转90度

M I E A
N J F B
O K G C
P L H D

中心逆时针旋转90度

D H L P
C G K O

B F J N
A E I M

问题2(40分):

在一个N*N的棋盘上，每行放置M个球，每列、每个45度的斜线上最多放置M个球，其中 $5 \leq N \leq 6$, $1 \leq M \leq (N-1)$ ，现要求出每个M, N组合中最多摆放球的不同的摆法的个数（包括重复和不重复的，分别输出）。请用最多两条SQL语句得到以下结果：（以M=2, N=5为例）

两条SQL的输出格式：

[\[Copy to clipboard\]](#) [\[-\]](#)

CODE:

SQL1:

M	N	AllCnt
2	5	92

--上期的结果总数是92，您答对了吗？

SQL2:

M	N	NoReptCnt
2	5	xx

一条SQL的输出格式：

[\[Copy to clipboard\]](#) [\[-\]](#)

CODE:

M	N	AllCnt	NoReptCnt
2	5	92	xx

Oracle变量定义如下（以M=2, N=5为例）：

```
var m number;  
exec :m:=2;  
var n number;  
exec :n:=5;
```

MS SQL Server变量定义如下（以M=2, N=5为例）：

```
declare @n int,@m int;  
set @n=5;  
set @m=2;
```

扩展题的执行方式最好如下：

```
set timing on  
exec :m:=2;  
exec :n:=5;  
@\\sql2-leitaisai.sql
```

书写格式：仅包含一个查询语句(可以有子查询)和必要的注释（用/* */括起），不得包含创建表、视图、索引和插入、更新、删除等语句。

输出格式：问题1输出一个整数

问题2输出见题目说明

数据库平台：适用Oracle、MS SQL Server，版本（Oracle推荐10gr2（包含）以上版本、MS SQL Sever推荐2008版本）

各位同学答题请务必注意：

1. 思路清晰度，关键步骤的思路注释
2. SQL结构清晰
3. 问题1和问题2分别计分，2者合计为本期总得分
4. 考虑用11gR2递归with查询实现（参考资料<http://www.china-pub.com/ureader/product.asp?bookid=197199>从14页开始）

第二期活动评委：〇〇, newkid, dingjun123, bq_wang、lastwinner

第三期

现在开始“盛拓传媒杯”SQL大赛第三期答题，答题时间为：2011年3月17日-3月24日24：00结束，请大家在此时间内答题。

本期题目：

2011年度itpub数据库技术大会将于4月15日隆重举行。与会人员分布于不同城市(A,B,C,D,...)，每个城市及其会员人数保存在一张表 cities(city_name, members)

这些城市及他们之间的通路构成了一张网络交通图。有直接通路的城市及其距离保存在表routes之中，所有通路都是双向的，但每条直接通路只用一行记录来表示。

外地的会员将乘坐出租车参加大会，车费等于距离*价格(常数)。本地会员不用考虑交通费。

交通费定义:itpub要为外地会员发放往返的交通补贴，每人每公里2元（双程），在哪个城市举办能够最节省，需要花费多少钱？

问题：选择在哪个城市举行，能够使得会员的总交通费最低？列出这个城市及应发放给其他城市会员的总的交通补助。

输出格式如下：

输出3列数据,按城市名升序排列

举办城市 TOTAL 总费用
举办城市 来自城市 费用
....

例如，假设你求出在A城市最省，总的补助需要发放10000元，其中给B城市的会员要发8000，给C城市的会员要发2000

输出

- 1. A TOTAL 10000
- 2. A B 8000
- 3. A C 2000

复制代码

如果存在在多个城市办会交通费用相同，则都列出,按城市名升序排列

例如，假设你求出在A,B城市最省，需要发10000元，如果在A办，B城市的会员8000，C城市的会员2000，如果在B办，A城市的会员8000，C城市的会员2000，

输出

- 1. A TOTAL 10000
- 2. A B 8000
- 3. A C 2000
- 4. B TOTAL 10000
- 5. B A 8000
- 6. B C 2000

复制代码

表结构和样例数据如下，样例数据供参考，评委将用多组数据验证你的解答。表结构不能变动，不按此表结构答题的不得分。

- 1. CREATE TABLE cities (
- 2. city_name VARCHAR2(10) PRIMARY KEY
- 3. ,members NUMBER
- 4.);
- 5. INSERT INTO cities
- 6. (
- 7. SELECT 'A',20 FROM dual union
- 8. SELECT 'B',31 FROM dual union
- 9. SELECT 'C',23 FROM dual union
- 10. SELECT 'D',42 FROM dual union
- 11. SELECT 'E',25 FROM dual union
- 12. SELECT 'F',29 FROM dual union
- 13. SELECT 'G',32 FROM dual union
- 14. SELECT 'H',45 FROM dual union
- 15. SELECT 'I',50 FROM dual union

```

16. SELECT 'J' ,32 FROM dual union
17. SELECT 'K' ,22 FROM dual
18. );
19. commit;
20.
21. CREATE TABLE routes (
22.     city1    VARCHAR2(10)
23.     ,city2    VARCHAR2(10)
24.     ,distance NUMBER
25.     ,PRIMARY KEY (city1,city2)
26.     ,CONSTRAINT check_cities CHECK (city1<city2) ---- 为了防止同一条通路被插入两遍
27. );
28. INSERT INTO routes
29. (
30. SELECT 'A' , 'B' , 71 FROM dual union
31. SELECT 'A' , 'C' , 80 FROM dual union
32. SELECT 'A' , 'D' , 80 FROM dual union
33. SELECT 'A' , 'E' , 66 FROM dual union
34. SELECT 'A' , 'F' , 86 FROM dual union
35. SELECT 'C' , 'G' , 112 FROM dual union
36. SELECT 'D' , 'G' , 60 FROM dual union
37. SELECT 'G' , 'H' , 102 FROM dual union
38. SELECT 'G' , 'I' , 91 FROM dual union
39. SELECT 'G' , 'J' , 133 FROM dual union
40. SELECT 'G' , 'K' , 127 FROM dual union
41. SELECT 'C' , 'D' , 79 FROM dual union
42. SELECT 'C' , 'H' , 155 FROM dual union
43. SELECT 'C' , 'E' , 119 FROM dual union
44. SELECT 'B' , 'D' , 52 FROM dual union
45. SELECT 'D' , 'I' , 44 FROM dual union
46. SELECT 'B' , 'L' , 41 FROM dual union
47. SELECT 'J' , 'L' , 201 FROM dual union
48. SELECT 'B' , 'F' , 79 FROM dual union
49. SELECT 'H' , 'K' , 38 FROM dual union
50. SELECT 'J' , 'K' , 81 FROM dual
51. );
52. commit;

```

复制代码

数据库平台：适用[Oracle](#)、MS SQL Server，版本（Oracle推荐10gr2（包含）以上版本、MS SQL Sever推荐2008版本）

第三期活动评委：OO, newkid, dingjun123, bq_wang、lastwinner

第四期

现在开始“盛拓传媒杯”SQL大赛最后一期 第四期答题，答题时间为：2011年3月25日-4月1日24：00结束，请大家在此时间内答题。

本期题目：

某项房产资源服务按天数、人数收费。收费标准与那一天的用户数有关，对于来自同一个房产经纪公司的用户，某天的总用户数越多则折扣越多。

```

1. CREATE TABLE services ( ----- 各项服务
2.     service_id  NUMBER PRIMARY KEY
3.     ,service_name VARCHAR2(20)
4. );
5. CREATE TABLE companies ( ----- 使用服务的公司
6.     company_id  NUMBER PRIMARY KEY
7.     ,company_name VARCHAR2(40)
8. );
9.

```

```

10. CREATE TABLE service_users ( -----公司下的用户（房产经纪）
11.     user_id    NUMBER PRIMARY KEY
12.     ,company_id NUMBER NOT NULL REFERENCES companies(company_id)
13.     ,user_name  VARCHAR2(20)
14. );
15. CREATE TABLE service_usage (
16. ----- 使用情况，起止日为闭合区间（首尾包含），如果只用一天，则start_date=end_date
17. ----- 该表为大表，数据较多
18.     usage_id    NUMBER PRIMARY KEY
19.     ,user_id     NUMBER NOT NULL REFERENCES service_users(user_id)
20.     ,service_id  NUMBER NOT NULL REFERENCES services(service_id)
21.     ,start_date  DATE   NOT NULL ----- 数据只含日期，不含时间
22.     ,end_date    DATE   NOT NULL ----- 数据只含日期，不含时间
23.     ,CONSTRAINT check_dates CHECK (end_date>=start_date)
24. );
25. CREATE TABLE service_rates ( -----收费标准，每个公司各不相同，按人数段计费
26.     service_id   NUMBER NOT NULL REFERENCES services(service_id)
27.     ,company_id  NUMBER NOT NULL REFERENCES companies(company_id)
28.     ,category_id NUMBER NOT NULL ----- 1,2,3,4表示1档，2档，3档，4档
29.     ,user_count_min NUMBER NOT NULL ----- 该档人数下限，1档从1开始
30.     ,user_count_max NUMBER NOT NULL ----- 该档人数上限，同样是闭合区间。最后一档不妨设为99999999
31.     ,rate        NUMBER NOT NULL ----- 该档的费率，乘以天数再乘以人数就是总费用
32.     ,PRIMARY KEY (company_id,service_id,category_id)
33. );
34. 输入：
35. 起止日期(yyyymmdd字符串), 闭合区间（首尾包含），如果只计算一天，则p_start_date=p_end_date
36. 绑定变量：
37. VAR p_start_date VARCHAR2(8);
38. VAR p_end_date VARCHAR2(8);
39. 输出：费用报表
40. company_id,company_name,service_id,service_name,fee1,fee2,fee3,fee4,total_fee
41. 其中:fee1,fee2,fee3,fee4分别表示1档，2档，3档，4档的费用,total_fee是上述4项的总和。若存在某档无数据，则总和为NULL。
42. 按照company_id,service_id排序(升序)。

```

复制代码

测试数据：

```

1. INSERT INTO services VALUES(1,'Service A');
2. INSERT INTO services VALUES(2,'Service B');
3. INSERT INTO services VALUES(3,'Service C');
4. INSERT INTO companies VALUES(1,'company A');
5. INSERT INTO companies VALUES(2,'company B');
6. INSERT INTO companies VALUES(3,'company C');
7. INSERT INTO service_users VALUES(101,1,'User 1 Comp A');
8. INSERT INTO service_users VALUES(102,1,'User 2 Comp A');
9. INSERT INTO service_users VALUES(103,1,'User 3 Comp A');
10. INSERT INTO service_users VALUES(104,1,'User 4 Comp A');
11. INSERT INTO service_users VALUES(201,2,'User 1 Comp B');
12. INSERT INTO service_users VALUES(202,2,'User 2 Comp B');
13. INSERT INTO service_users VALUES(203,2,'User 3 Comp B');
14. INSERT INTO service_users VALUES(204,2,'User 4 Comp B');
15. INSERT INTO service_users VALUES(301,3,'User 1 Comp C');
16. INSERT INTO service_users VALUES(302,3,'User 2 Comp C');
17. INSERT INTO service_users VALUES(303,3,'User 3 Comp C');
18. insert into service_rates values (1,1,1,1,1,20);
19. insert into service_rates values (1,1,2,2,2,15);
20. insert into service_rates values (1,1,3,3,3,10);
21. insert into service_rates values (1,1,4,4,99999999,5);
22. insert into service_rates values (2,1,1,1,1,30);

```

```

23. insert into service_rates values (2,1,2,2,2,25);
24. insert into service_rates values (2,1,3,3,3,20);
25. insert into service_rates values (2,1,4,4,99999999,15);
26. insert into service_rates values (1,2,1,1,1,22);
27. insert into service_rates values (1,2,2,2,2,17);
28. insert into service_rates values (1,2,3,3,3,12);
29. insert into service_rates values (1,2,4,4,99999999,8);
30. insert into service_rates values (2,2,1,1,1,28);
31. insert into service_rates values (2,2,2,2,2,21);
32. insert into service_rates values (2,2,3,3,3,17);
33. insert into service_rates values (2,2,4,4,99999999,12);
34. INSERT INTO service_usage VALUES(1,101,1,DATE '2010-1-5',DATE '2010-1-10');
35. INSERT INTO service_usage VALUES(2,102,1,DATE '2010-1-3',DATE '2010-1-11');
36. INSERT INTO service_usage VALUES(3,103,1,DATE '2010-1-7',DATE '2010-1-10');
37. INSERT INTO service_usage VALUES(4,104,1,DATE '2010-1-10',DATE '2010-1-26');
38. INSERT INTO service_usage VALUES(11,101,2,DATE '2010-1-1',DATE '2010-1-10');
39. INSERT INTO service_usage VALUES(12,102,2,DATE '2010-1-2',DATE '2010-1-8');
40. INSERT INTO service_usage VALUES(13,103,2,DATE '2010-1-7',DATE '2010-1-10');
41. INSERT INTO service_usage VALUES(14,104,2,DATE '2010-1-5',DATE '2010-1-13');
42. INSERT INTO service_usage VALUES(21,201,1,DATE '2010-1-2',DATE '2010-1-21');
43. INSERT INTO service_usage VALUES(22,202,1,DATE '2010-1-1',DATE '2010-1-17');
44. INSERT INTO service_usage VALUES(23,203,1,DATE '2010-1-7',DATE '2010-1-13');
45. INSERT INTO service_usage VALUES(24,204,1,DATE '2010-1-3',DATE '2010-1-30');
46. INSERT INTO service_usage VALUES(31,201,2,DATE '2010-1-1',DATE '2010-1-2');
47. INSERT INTO service_usage VALUES(32,202,2,DATE '2010-1-2',DATE '2010-1-3');
48. INSERT INTO service_usage VALUES(33,203,2,DATE '2010-1-4',DATE '2010-1-10');
49. INSERT INTO service_usage VALUES(34,204,2,DATE '2010-1-11',DATE '2010-1-13');
50. commit;

```

复制代码

答题注意：

1. 选手必须按照题目给定的表结构答题，否则不得分
2. 应提供总体思路 and 关键步骤的汉字注释
3. 并提供样例数据对如下输入变量的输出结果，以便验证

```

1. VAR p_start_date VARCHAR2(8);
2. VAR p_end_date VARCHAR2(8);
3. EXEC p_start_date := '20100104';
4. EXEC p_end_date := '20100130';

```

复制代码

4. 要求用一个SQL查询语句完成（可以包括子查询，但不能包括ddl和dml语句）

第四期活动评委：OO, newkid, dingjun123, bq_wang, lastwinner

参与规则：

每个参赛者以附件方式贴出自己的源代码，并将阅读权限设为11~100之间的数值，答案贴出后不允许编辑帖子。编辑过的帖子做无效处理。需要修改答案请发表新的回复，并以同样的方式上传新的答案。

附件名称以SQL大赛第N期(ID).txt来命名，通过rar压缩加密，推荐采取select sys_guid() from dual的方式产生密码，答题者需要在截止时间前将密码以论坛PM给论坛助理~贝贝~

pm链接：<http://www.itpub.net/pm.php?action=send&uid=93405>

此次活动也非常欢迎新注册会员参与，但因新注册会员三天内均在临时用户组，故不能上传附件，所以请参与此次活动的新会员，把答题压缩包和解压密码直接发邮件给我即可：wangbei@it168.com，我会在收到后第一时间给予您回复的。

特别申明：在答题期间不准在论坛公开讨论与此题有关的解题思路！

活动流程：

- 1、发布本期活动题目及答题时间范围，大家可以在此期间根据以上参与规则上传自己的参赛程序文档。
- 2、结束答题，进入评审阶段，评审将在**一周内**对所有参赛程序进行评审，评审结束后将公布评审及作品点评。

评审规则：

- 1、由ITPUB论坛助理下载参赛者附件并解压后删除参赛者ID, 将各参赛文档编入代号，并发给当期评委。
- 2、评委抽签获得序号并评审相应序号的答案。运行程序，将运行出正确答案的参赛文档选出。
- 3、所有评委独立对运行出正确答案的程序打分，把相应分数告知论坛助理。论坛助理取平均数为该答案最后得分。

注：出题者负责向评委讲解思路，并配合评委书写点评。

第三期的答题结果，预计于4月6日公布，评委们在此期间会抓紧时间对大家的答题进行评审和点评给予建议！现在请大家继续支持第四期的活动，参与答题吧，无论语句写的好坏，参与其中必然得到最多收获，期待更多的puber参与到活动中！