



First International NoCOUG SQL Challenge

BE IT KNOWN BY THESE PRESENTS that the great Wizard of Odds at Hogwash School of Es-Cue-El needs your help in solving the riddle of the ancient jade icosahedron found in the secret chamber of mystery. A great tournament has been organized, and all practitioners of the ancient arts of Es-Cue-El have been invited to demonstrate their prowess.

Unsolvable Riddle

An ancient 20-sided die (icosahedron) has been discovered in the secret chamber of mystery at Hogwash School of Es-Cue-El. A mysterious symbol is inscribed on each face of the die. The great Wizard of Odds has discovered that each symbol represents a number. The great wizard has also discovered that the die is biased: that is, it is more probable that certain numbers will be displayed than others if the die were used in a game of chance. The great wizard has recorded this information in tabular fashion as described below.

Name	Null?	Type
-----	-----	----
FACE_ID	NOT NULL	INT
FACE_VALUE	NOT NULL	INT
PROBABILITY	NOT NULL	REAL

The great wizard now implores you to create an Es-Cue-El spell that displays the probabilities of obtaining various sums when the die is thrown N times in succession in a game of chance.*

Big Prizes

The *August Order of the Wooden Pretzel* will be conferred on the winner, in keeping with the celebrated pronouncement of another great wizard that “*some people can perform seeming miracles with straight Es-Cue-El, but the statements end up looking like pretzels created by somebody who is experimenting with hallucinogens.*” As if that singular honor were not enough, a marvelous collection of oracular tomes will be bestowed upon the champion. May the best wizard win!

RULES: The winner will receive his or her choice of six books from the Apress book catalog. Due to shipping costs and limitations for certain parts of the world, electronic copies may be substituted. Prizes may be awarded to runners-up at the discretion of the organizers. Submissions should be emailed to SQLchallenge@nocoug.org. Contestants may use any database technology at their disposal, but the submitted solutions should be compatible with at least one of the following database technologies: Oracle 11g for Windows, SQL Server 2008, and DB2 9.5 for Windows. The competition will be judged by Dan Tow, author of *SQL Tuning*, published by O’Reilly Media, and Iggy Fernandez, author of *Beginning Oracle Database 11g Administration*, published by Apress. Judging criteria include correctness, originality, efficiency, portability, and readability. The judges’ decisions are final. The competition will close at a time determined by the organizers. The judges and organizers reserve the right to publish and comment on any of the submissions with due credit to the originators. More information about the problem and additional rules can be found at www.nocoug.org.

* N is a “substitution variable” or “bind variable.”



Backgrounder

Here are SQL commands that can be used to create the table and populate it with sample data. In the interest of space, we consider a cube instead of an icosahedron. Notice that the die is biased and that the faces are not numbered conventionally.

```
CREATE TABLE die(  
    face_id INT NOT NULL,  
    face_value INT NOT NULL,  
    probability REAL NOT NULL,  
    CONSTRAINT pk_die PRIMARY KEY (face_id));  
INSERT INTO die VALUES (1, 1, 1/6 + 1/12);  
INSERT INTO die VALUES (2, 3, 1/6 + 1/12);  
INSERT INTO die VALUES (3, 4, 1/6 + 1/12);  
INSERT INTO die VALUES (4, 5, 1/6 - 1/12);  
INSERT INTO die VALUES (5, 6, 1/6 - 1/12);  
INSERT INTO die VALUES (6, 8, 1/6 - 1/12);
```

Consider the problem of computing the probabilities of obtaining various sums in N throws of the die. Here is the solution for $N = 2$; it requires a two-way Cartesian join. The solution for $N = 3$ would require a three-way Cartesian join, and so on.

```
SELECT    SUM,  
          SUM (probability) AS probability  
FROM      (SELECT d1.face_value + d2.face_value AS SUM,  
                d1.probability * d2.probability AS probability  
           FROM die d1 CROSS JOIN die d2) temp  
GROUP BY SUM  
ORDER BY SUM;  
-----  
SUM      PROBABILITY  
-----  
2        0.0625000000  
4        0.1250000000  
5        0.1250000000  
6        0.1041666667  
7        0.1666666667  
8        0.1041666667  
9        0.1250000000  
10       0.0486111111  
11       0.0555555555  
12       0.0486111111  
13       0.0138888889  
14       0.0138888889  
16       0.0069444444
```

If the die is unbiased and numbered conventionally—beginning with 1—the following results are obtained. Explicit formulas can be found in such cases; refer to mathworld.wolfram.com/Dice.html.

```
-----  
SUM      PROBABILITY  
-----  
2        0.0277777778  
3        0.0555555556  
4        0.0833333334  
5        0.1111111112  
6        0.1388888889  
7        0.1666666667  
8        0.1388888889  
9        0.1111111112  
10       0.0833333334  
11       0.0555555556  
12       0.0277777778
```

As the above analysis shows, it is not hard to solve the problem for particular values of N . The challenge is to compute the probabilities without hard-coding the value of N .

Judges' Statement



Solutions that use procedural loops to multiply probabilities are not eligible. Points will be awarded in the following categories:

- **Avoidance of non-SQL extensions:** Avoidance of non-SQL extensions such as Oracle SQL*Plus. For example, an Oracle SQL*Plus script that first generates an SQL statement and then executes it would lose points in this category.
- **Inclusion of commentary:** Inclusion of commentary or other documentation that clearly explains the logic of the solution—especially explaining any use of uncommon features or features that are specific to only your version of SQL, or that are features of the SQL standard that are not implemented in all common vendor implementations. Links to online documentation of unusual features used are a help here. It is our wish that a broad range of readers at differing levels of ability, and specializing in different vendors' implementations, should be able to learn useful lessons from the solutions offered. Commentary will lose points for leaving out a discussion of limits. If you think your approach runs into limits—logically or from a performance perspective—document those limits for highest points. Don't count on the judges not noticing the limits—it's better to mention them yourself!
- **Inclusion of test results:** Inclusion of documentation that demonstrates actual tests of the SQL solution against a range of actual data and inputs, clear enough and comprehensive enough to demonstrate correctness and the degree of efficiency and scalability over the ranges tested. A perfect score in this category does not imply that the results are good; for example, you could get a perfect documentation score showing well-documented results demonstrating poor performance and scaling, and even actual errors, at high N. Points will be deducted if the judges believe that testing at higher N than you demonstrate would have led to outright errors, though points will not be deducted for documented tests that are cut short after long runs for practical reasons.
- **Elegance, readability, and clarity:** Elegance, readability, and clarity of the solution, not counting the comments and other documentation, which may be as long as needed to thoroughly explain the logic and to document the tests.
- **Ability to scale logically to large N:** This reflects how well the logic could handle scaling in the absence of resource constraints.
- **Demonstrated ability to scale performance to large N:** This reflects how well the solution handles scaling in the presence of resource constraints, and how well this scaling has been demonstrated in any attached test results.
- **Avoidance of nonstandard SQL features:** Avoidance of SQL features that are implemented only by one particular database vendor, are not part of the SQL standard, or are part of the SQL standard that is not widely and correctly implemented.