Solving a Combinatorics Challenge by Exploiting Computational Techniques Available on Relational Databases

Wei Hu

Fairfield University
wei.hu@student.fairfield.edu

Mirco Speretta

Gateway Community College
msperetta@gwcc.commnet.edu

Objective

- Create a data set used in a separate research study whose goal was to count all the possible card combinations of the SET game®

Background

About the SET game

- The **SET game** has a rich mathematical structure that involves principles of combinatorics.

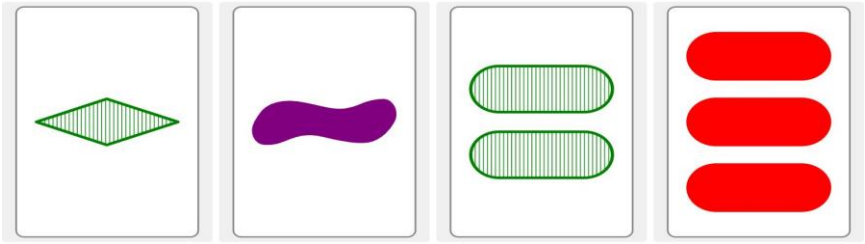
- **Constraint** of the study: every hand contains exactly four SETs, one for each possible type of combination of cards.

Background – About the SET game

Cards:

- Four attributes each card
- Three features per attribute
- 81 cards deck

Attribute	Feature
Number	{One, Two, Three}
Shading	{Solid, Striped, Open}
Color	{Red, Green, Purple}
Shape	{Oval, Squiggle, Diamond}

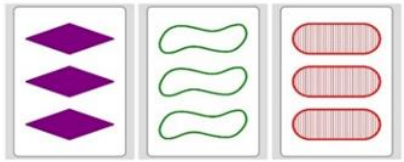


Typical cards of SET game

Background – About the SET game

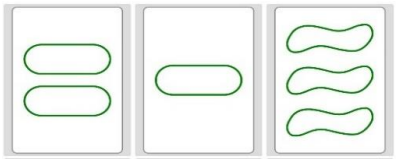
Rules:

- Three cards are called a ***SET*** if, with respect to each of the four attributes, the cards are either all the same or all different.
- The goal is to find collections of three cards satisfying its rule.



Typical SET

Three cards have different shapes, different colors, and different shading, and each card has a same number of shapes (three).



Not a SET

There are two oval cards and one squiggle card. Thus, the cards are neither all the same nor all different with respect to the shape attribute.

Background – About the SET game

SETs:

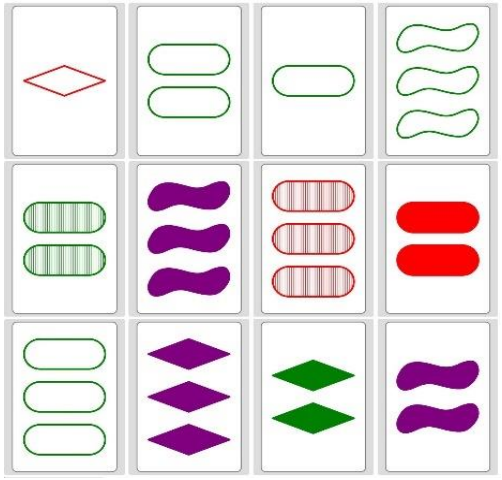
Four types of *SET*s players can identify.

- Case 1: One attribute has different features; three attributes have the same features. The number of this type of *SET* is 108.
- Case 2: Two attributes have different features; two attributes have the same features. The number of this type of *SET* is 324.
- Case 3: Three attributes have different features; one attribute has the same feature. The number of this type of *SET* is 432.
- Case 4: All four attributes have different features. The number of this type of *SET* is 216.

Background – About the SET game

Hand:

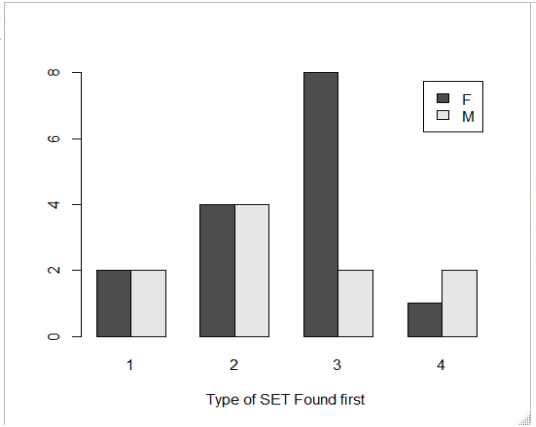
To play the game, 12 cards called one hand are dealt in front of players.

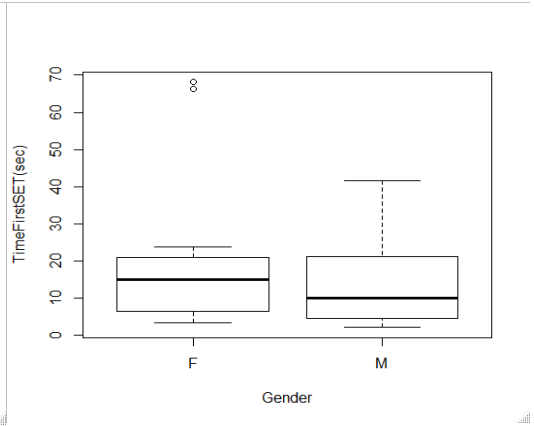


Players search for *SET*s. After all *SET*s in the hand are found the hand cards will be refreshed and dealt with another twelve random cards out of the deck.

Background – Math Circle

Dr. McSweeney and Dr. Striuli, two math professors at the University of Fairfield, noticed potential differences in the type of *SET* a student found based on gender. Middle schoolers collected a small sample data of twenty-five students and their friends and families using one hand where there were exactly four *SETs* one for each type. They had some interesting results illust





Background – Software Engineering

In order to facilitate the study and collect broader data, a student of software engineering and a professor of computer science were tasked to

1) collect data from a wider population
2) find a way to automatically generate hand combinations that have exactly four *SETs* one for each type.

Design

- We had to consider three main objects: a card, a *SET*, and a hand. Cards are the basic elements that form *SETs*, hands and deck.
- Basic four types of *SETs are generated by a Java program*.
- Constraint-based (not random) hand combinations are generated using SQL.
- Cross join, data tuple comparison, ADD function, DIV function and MOD function of SQL are used to manipulate data.
- Database was implemented on MySQL 5.7.22
- Web-based interface was implemented with Django 2.1.7 with Python 3.7.3
- Program is implemented in Java 1.8.0_181

Methodology - Card Value Definition

We used four-digit numbers to define cards. From left to right, each digit refers to one attribute. Each digit has three variables which are zero, one and two.

Attribute / *Variable*	**Number**	**Shading**	**Color**	**Shape**
	Position (from left to right)			
0	One	Open	Red	Diamond
1	Two	Striped	Green	Oval
2	Three	Solid	Purple	Squiggle

Methodology - *SET* Type Definition

We identified four types of *SET* as Type 1, Type 2, Type 3 and Type 4.

- **Type 1 *SET***: Only one attribute is the same, the other three attributes are different. The number of this type of *SETs* is 432.
- **Type 2 *SET***: Two attributes are the same, and the other two attributes are different. The number of this type of *SETs* is 324.
- **Type 3 *SET***: Three attributes are the same, only one attribute is different. The number of this type of *SETs* is 108.
- **Type 4 *SET***: None attribute is the same. The number of this type of *SETs* is 216.

Methodology - Hand Data Set Generation

Our goal is to find all possible hand card combinations that satisfy the requirements of the study:

1) Every hand contains exactly four *SETs*, one for each type.
2) A card occurrence is unique in four *SETs* of one hand.

The number of different 12-card combinations is $\binom{81}{12} = \frac{81!}{12!69!} = 70{,}724{,}320{,}184{,}700 \approx \mathbf{7.07 \times 10^{13}}$

The number of hand combinations is 432×324×108×216 = 3,265,173,504 = **3.265×10⁹** by taking *SETs* as basic components of hand.

Methodology - Hand Data Set Generation

Four Basic Table Generation (1)

We define four basic tables, one table for each type of *SET*. Potential card combinations of hand are achieved based on these four basic tables.

Table #	Table Name	*SET* Type	Number of Records
1	type1	Type 1	432
2	type2	Type 2	324
3	type3	Type 3	108
4	type4	Type 4	216

Methodology - Hand Data Set Generation

Four Basic Table Generation (2)

We generate each attribute of one card at one time using JAVA program, so we store 12 attributes of 3 cards separately into twelve columns of tables (type1 – type4).

SetId	c1a1	c1a2	c1a3	c1a4	c2a1	c2a2	c2a3	c2a4	c3a1	c3a2	c3a3	c3a4
1	0	0	0	0	0	1	1	1	0	2	2	2
2	0	0	0	0	0	1	1	2	0	2	2	1
3	0	0	0	1	0	1	1	2	0	2	2	0
4	0	0	0	1	0	1	1	0	0	2	2	2
5	0	0	0	2	0	1	1	0	0	2	2	1
6	0	0	0	2	0	1	1	1	0	2	2	0
7	0	0	0	0	0	1	2	1	0	2	1	2
8	0	0	0	0	0	1	2	2	0	2	1	1

Every SET has three cards, namely C1, C2 and C3. Each card has four attributes namely A1, A2, A3 and A4. (CnAm – Card n Attribute m, n=1,2,3; m=1,2,3,4)

Methodology - Hand Data Set Generation

Four Basic Table Validation

We run definition-based testing to validate these four tables. After a SELECT SQL script is run for each table, those records satisfy the SELECT condition are retrieved. If the number of records retrieved equals to the number of the specific type of *SET*, this table is validated.

For example, we are going to validate type1. After it was generated, we make sure it has 432 records consistent with the result known in combinatorics. After the SELECT is run, 432 records are all retrieved. Table type1 is validated. And so on.

Methodology - Hand Data Set Generation

Attribute Concatenation

After type1, type2, type3 and type4 four basic tables are validated, we concatenate four attributes into one four-digit number per card. We stored new records into another four tables type1_concat, type2_concat, type3_concat and tyep4_concat.

SetId	card1	card2	card3
1	0	111	222
2	0	112	221
3	1	112	220
4	1	110	222
5	2	110	221
6	2	111	220
7	0	121	212
8	0	122	211

Methodology - Hand Data Set Generation

Cross-join & De-duplicate

- Cross-join of four type of *SETs* is to get all possible hand card combinations.
- After every step of cross joins, the records where cards' occurrence is more than once are removed.

id	card1	card2	card3	card4	card5	card6
1	0	1000	2000	0	1111	2222
14	111	1111	2111	0	1111	2222
27	222	1222	2222	0	1111	2222
28	0	100	200	0	1111	2222
41	1011	1111	1211	0	1111	2222
54	2022	2122	2222	0	1111	2222
55	0	10	20	0	1111	2222
68	1101	1111	1121	0	1111	2222

The left table illustrates the records where cards' occurrence is more than once after type3 and type4 are cross joined.

Methodology - Hand Data Set Generation

Hand Combination Validation (3-card groups)

- The records of hand are composed by 12 cards within that every 3-card combination, one card from each type could form a *SET*.
- There are 4 ($\binom{4}{3}$) groups, 27 (3^3) 3-card combinations each group that could make up new SETs. We take 3-card combination highlighted for following illustration.

Group	Potential *SET*s (3-card combinations)		
Type 3 + Type 4 + Type 2	card1+card4+card7	card1+card4+card8	card1+card4+card9
	card1+card5+card7	card1+card5+card8	card1+card5+card9
	card1+card6+card7	card1+card6+card8	card1+card6+card9
	card2+card4+card7	card2+card4+card8	card2+card4+card9
	card2+card5+card7	card2+card5+card8	card2+card5+card9
	card2+card6+card7	card2+card6+card8	card2+card6+card9
	card3+card4+card7	card3+card4+card8	card3+card4+card9
	card3+card5+card7	card3+card5+card8	card3+card5+card9
	card3+card6+card7	card3+card6+card8	card3+card6+card9

Methodology - Hand Data Set Generation

Hand Combination Validation (3-card combination validation)

We take these 2 hand records before validation shown below for example:

card1	card2	card3	card4	card5	card6	card7	card8	card9	card10	card11	card12	id
1	1001	2001	222	1111	2000	110	1010	2210	2012	2101	2220	1
1	1001	2001	0	1111	2222	100	112	121	20	1210	2100	2

Hand (id = 1)	A1	A2	A3	A4	Hand (id = 2)	A1	A2	A3	A4
Card 1	0	0	0	1	**Card 1**	0	0	0	1
Card 4	0	2	2	2	**Card 4**	0	0	0	0
Card 7	0	1	1	0	**Card 7**	0	1	0	0
SUM	0	3	3	3	**SUM**	0	1	0	1
Remainder (SUM%3)	0	0	0	0	**Remainder (SUM%3)**	0	1	0	1
	SET					***Non-SET***			

If remainders (sum % 3) of four attributes are all zero, these 3 cards form a *SET* or else they don't.

.

Methodology - Hand Data Set Generation

Hand Combination Validation (Attribute check of 3-card)

In order to validate 3-card combination, we need to check each attribute of them. If any attribute of 3 cards is neither all same nor all different, the 3-card combination is not a *SET*. Let's take color attribute of three cards to explain this algorithm illustrated as table below.

Color Attribute of 3 cards	**All different**	**All same**	**Neither (Non-SET)**
Attribute value of 3 cards	0+1+2	All 0, 1 or 2	0+0+1, 0+0+2, 1+1+2, 1+1+0, 2+2+0, 2+2+1
Sum of attribute value	3	0, 3 or 6	1, 2, 4 or 5
Remainder (Sum%3)	0	0	1 or 2

Methodology - Hand Data Set Generation

Hand Combination Validation (SQL script for validation)

we use the WHERE clause in SELECT statement as ***WHERE not (((card1+card4+card7) div 1000) mod 3) + (((card1+card4+card7) div 100) mod 10 mod 3) +(((card1+card4+card7) div 10) mod 10 mod 3) + ((card1+card4+card7) mod 10 mod 3) = 0*** to validate the 3-card is not a *SET*; and so do the rest 3-card combinations.

Finally, we got **269,635,392** hand combinations satisfying our requirements and so far the hand data set is generated.

Conclusions

- In this study we explored the feasibility of generating automatically data required to support an experimental study based on users' feedbacks.
- The amount of data to be generated presented a challenge: it was not possible to count mathematically the number of card combinations (i.e., hands).
- We solved this problem by implementing a computational design into a relational database server.
- We believe that this approach can be used in many other scenarios in which the creation of data generation is required. Experimental studies based on users' feedbacks should particularly benefit from this approach.