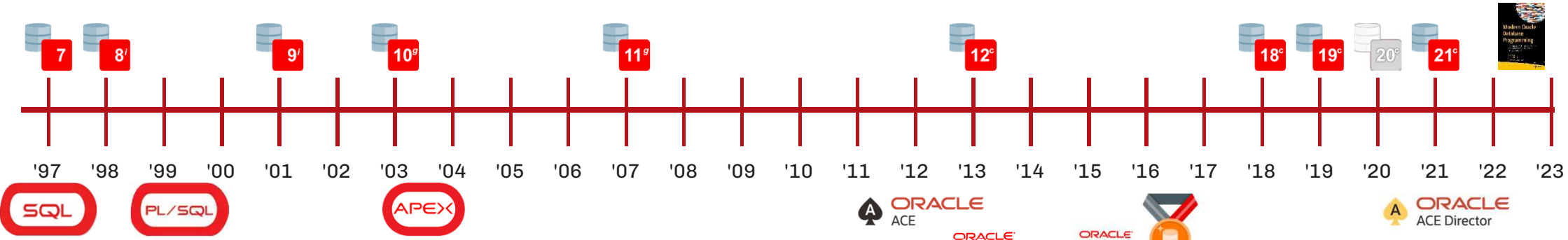
PBarel@Qualogy.com
http://blog.bar-solutions.com

About me...



bar-solutions.com blog http://blog.bar-solutions.com

~~All things ORACLE http://allthingsoracle.com~~

www.red-gate.com/simple-talk/author/patrick-barel/

~~OTECH MAGAZINE http://www.otechmag.com~~

bar-solutions.com/otechmagazine.php

Plugins for PL/SQL Developer
http://plugins.bar-solutions.com

Modern Oracle Database Programming

Level Up Your Skill Set to Oracle's Latest and Most Powerful Features in SQL, PL/SQL, and JSON

—

Alex Nuijten
Patrick Barel

Foreword by Chris Saxon

Apress®

Contact me…

@patch72

PBarel@Qualogy.com

Patrick.Barel@GMail.com

patrick@bar-solutions.com

Patrick.Barel@GMail.com

Patrick Barel

Oracle ACE Program

500+ technical experts helping peers globally

The **Oracle ACE Program** recognizes and rewards community members for their technical and community contributions to the Oracle community

3 membership tiers

Oracle ACE Director | Oracle ACE Pro | Oracle ACE Associate

For more details on Oracle ACE Program:
ace.oracle.com

Oracle ACE

Nominate
~~yourself~~ or someone you know:
ace.oracle.com/nominate

Connect: **aceprogram_ww@oracle.com** Facebook.com/OracleACEs @oracleace

Oracle Cloud Infrastructure

New Free Tier

oracle.com/cloud/free

Always Free

Services you can use for unlimited time

+

30-Day Free Trial

Free credits you can use for more services


MASH
MASH PROGRAM
Mentor and Speaker Hub
Our goal is to connect speakers with mentors to assist in preparing technical sessions and improving presentation skills
Interested? Read more and get in touch
https://mashprogram.wordpress.com

SYMPOSIUM 42

Created by the community, to support the community

Sharing of reliable knowledge
Supporting the various user groups and individuals

@sym_42

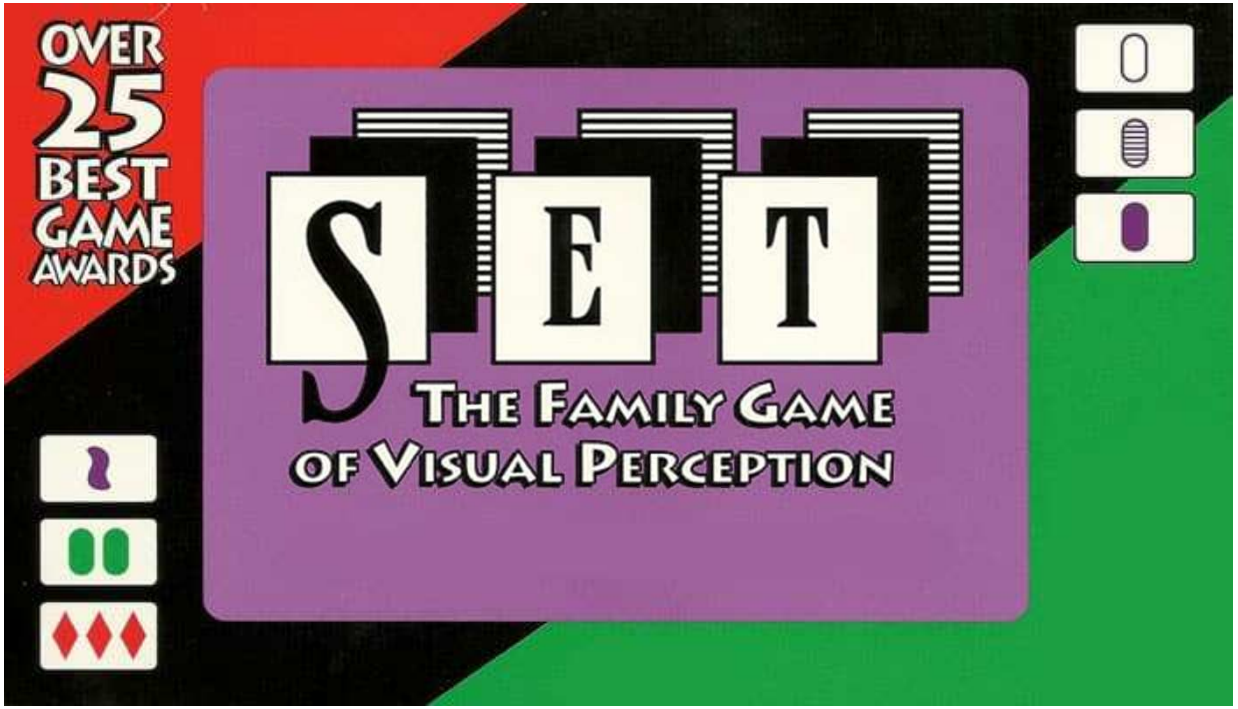
https://sym42.org/

Game-Set-Match

Patrick Barel, Qualogy

December 8, 2023

QUALOGY


OVER 25 BEST GAME AWARDS
SET
THE FAMILY GAME OF VISUAL PERCEPTION

Can you explain the game SET?

Sure! SET is a card game where players try to find sets of three cards that satisfy all of these conditions:

They have the same shape of symbols, or different shapes.
They have the same color of symbols, or different colors.
They have the same shading of symbols, or different shadings.
They have the same number of symbols, or different numbers.

How about an example?

When you look at the backside of the box there is information and a sample puzzle there.

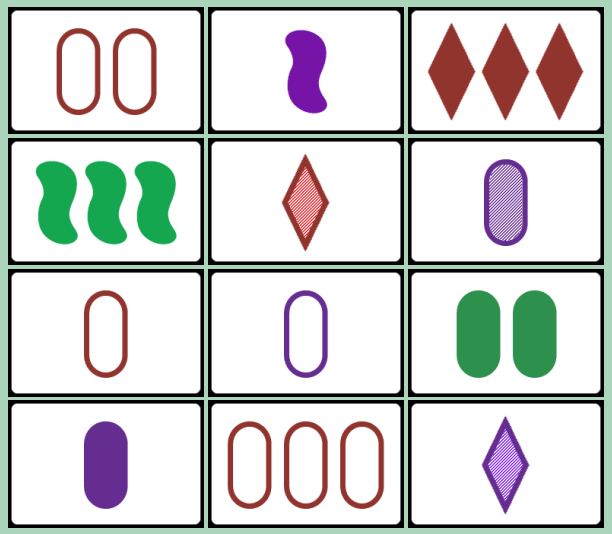
OVER 25 BEST GAME AWARDS

SET

THE FAMILY GAME OF VISUAL PERCEPTION

THE OBJECT OF THE GAME
Is to identify "sets" of three cards. Each card is unique in its four features: **number**: (1,2, or 3); **symbol**: (diamond, squiggle, or oval); **shading**: (solid, striped, or open); and **color**: (red, green, or purple).

A "set" consists of three cards on which each feature is either the same on all of the cards, or different on all of the cards.



Ages: 6 to adult
Number of Players: 1 or more

FIND THE **"SETS" IN THIS PUZZLE.** There are six "sets" in the twelve cards pictured to the left. Try to find all six. These "sets" are the ones shown in several places on the outside of this box.

SET is a board game in which any table can become the board. Package contains complete instructions for play, 81 cards, and a durable plastic carrying case.

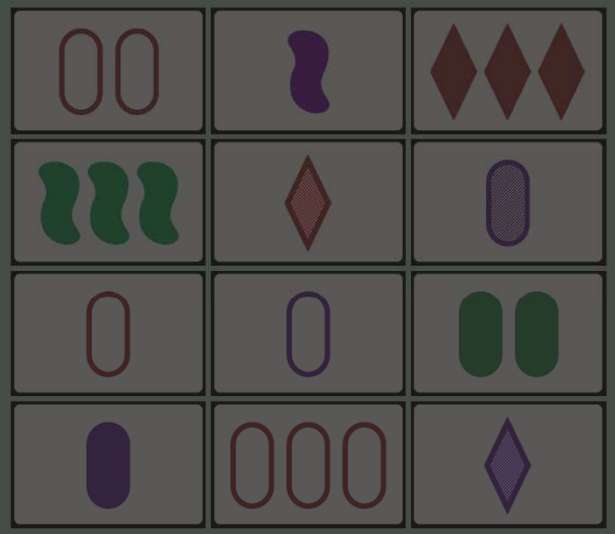
7 36396 00100 9

ISBN 0-9634691-0-X Made in U.S.A.
Package design: John Langdon, Phila. PA

SET®

THE OBJECT OF THE GAME
Is to identify "sets" of three cards. Each card is unique in its four features: **number**: (1,2, or 3); **symbol**: (diamond, squiggle, or oval); **shading**: (solid, striped, or open); and **color**: (red, green, or purple).

A "set" consists of three cards on which each feature is either the same on all of the cards, or different on all of the cards.



Ages: 6 to adult
Number of Players: 1 or more

FIND THE **"SETS" IN THIS PUZZLE.** There are six "sets" in the twelve cards pictured to the left. Try to find all six. These "sets" are the ones shown in several places on the outside of this box.

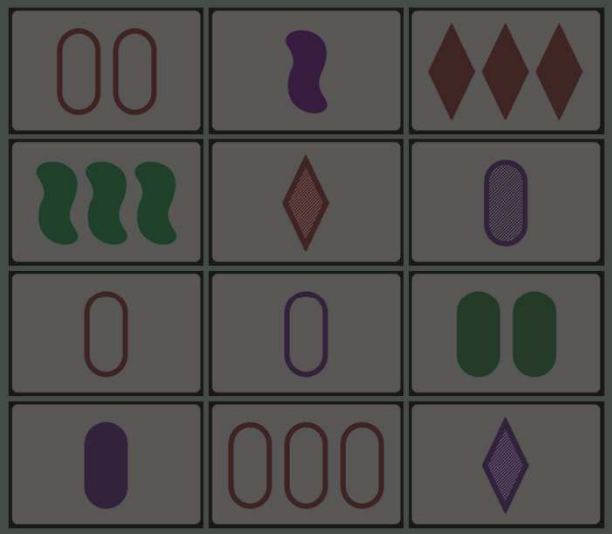
SET is a board game in which any table can become the board. Package contains complete instructions for play, 81 cards, and a durable plastic carrying case.

ISBN 0-9634691-0-X Made in U.S.A.
Package design: John Langdon, Phila. PA

7 36396 00100 9

THE OBJECT OF THE GAME
Is to identify "sets" of three cards. Each card is unique in its four features: **number**: (1,2, or 3); **symbol**: (diamond, squiggle, or oval); **shading**: (solid, striped, or open); and **color**: (red, green, or purple).

A "set" consists of three cards on which each feature is either the same on all of the cards, or different on all of the cards.



Ages: 6 to adult
Number of Players: 1 or more

FIND THE **"SETS" IN THIS PUZZLE.** There are six "sets" in the twelve cards pictured to the left. Try to find all six. These "sets" are the ones shown in several places on the outside of this box.

SET is a board game in which any table can become the board. Package contains complete instructions for play, 81 cards, and a durable plastic carrying case.

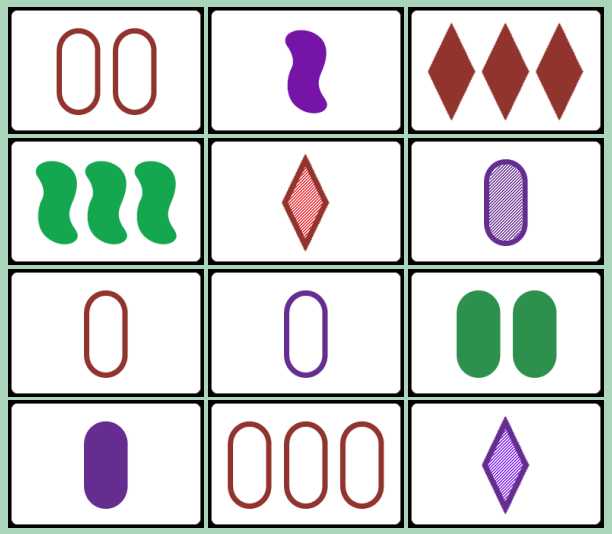
7 36396 00100 9

ISBN 0-9634691-0-X Made in U.S.A.
Package design: John Langdon, Phila. PA

SET®

THE OBJECT OF THE GAME
Is to identify "sets" of three cards. Each card is unique in its four features: **number**: (1,2, or 3); **symbol**: (diamond, squiggle, or oval); **shading**: (solid, striped, or open); and **color**: (red, green, or purple).

A "set" consists of three cards on which each feature is either the same on all of the cards, or different on all of the cards.



Ages: 6 to adult
Number of Players: 1 or more

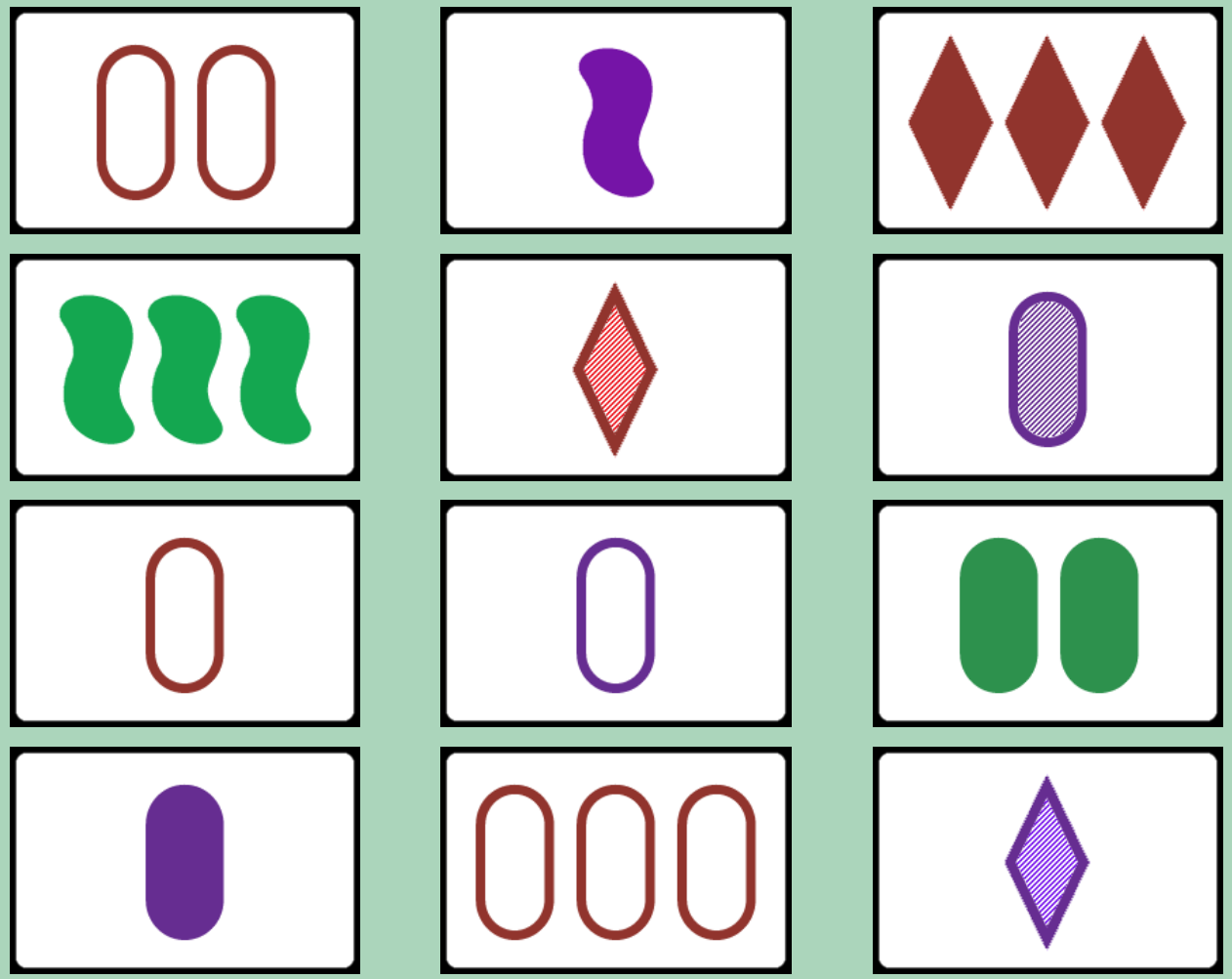
FIND THE **"SETS" IN THIS PUZZLE.** There are six "sets" in the twelve cards pictured to the left. Try to find all six. These "sets" are the ones shown in several places on the outside of this box.

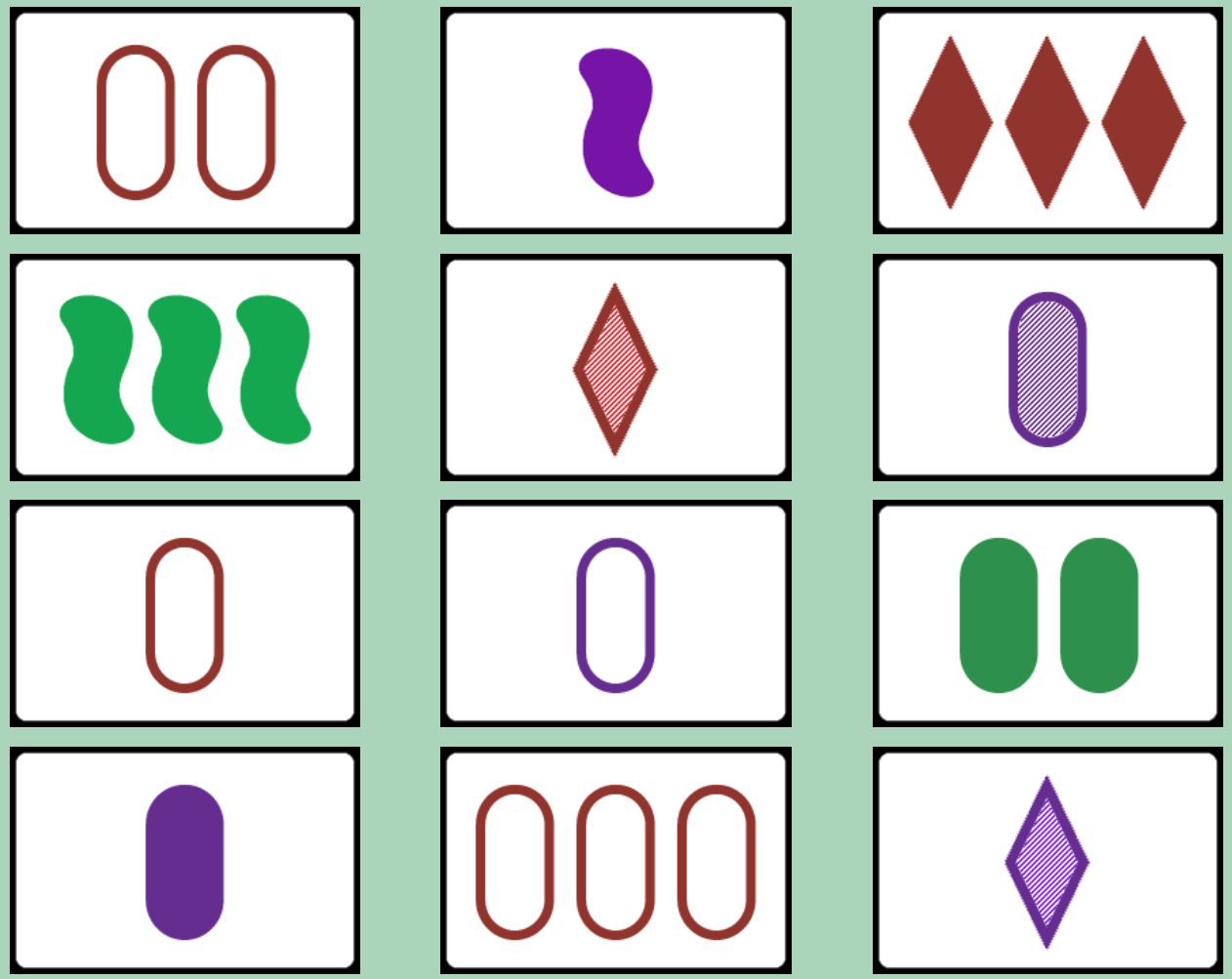
SET is a board game in which any table can become the board. Package contains complete instructions for play, 81 cards, and a durable plastic carrying case.

7 36396 00100 9

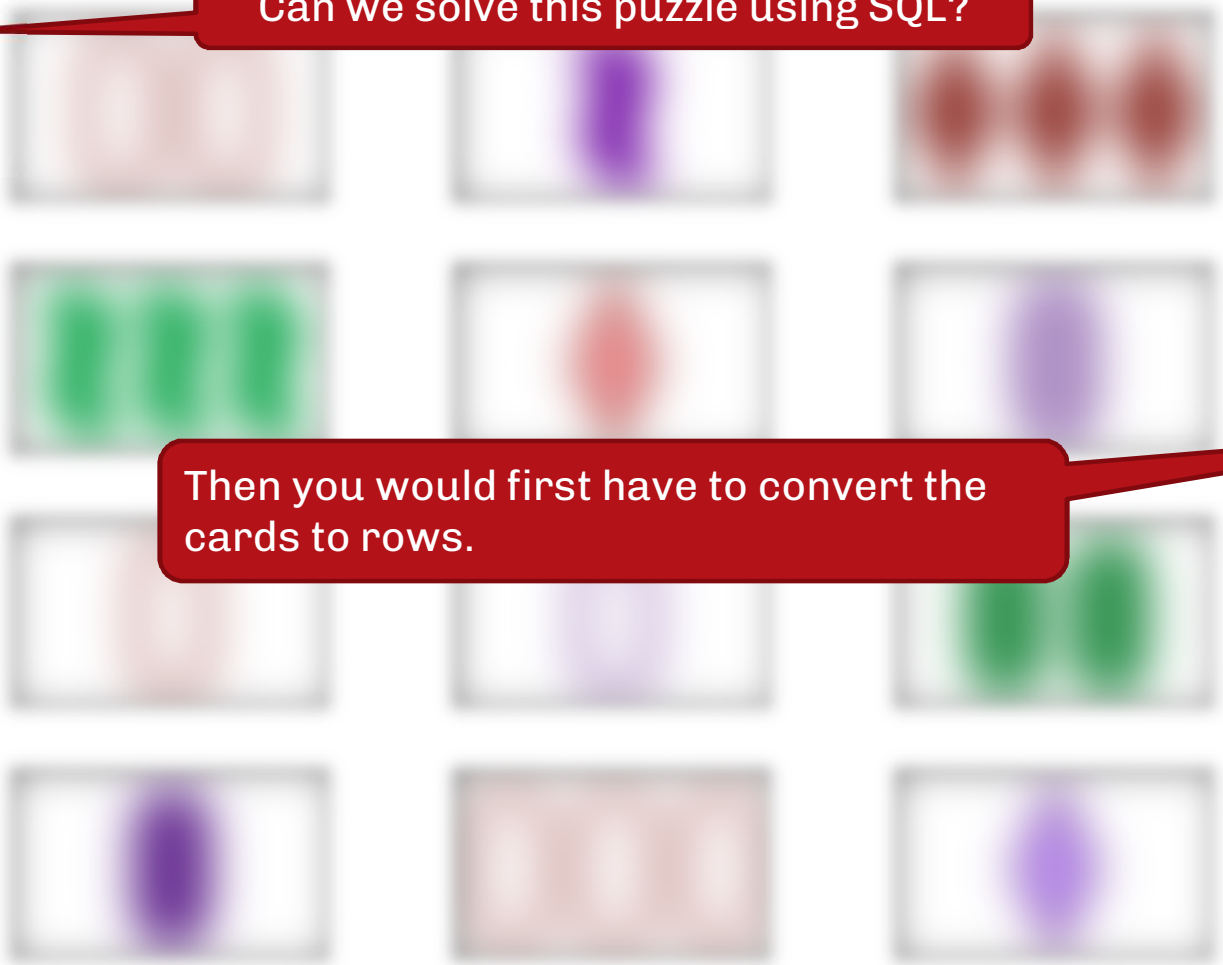
ISBN 0-9634691-0-X Made in U.S.A.
Package design: John Langdon, Phila. PA

SET®



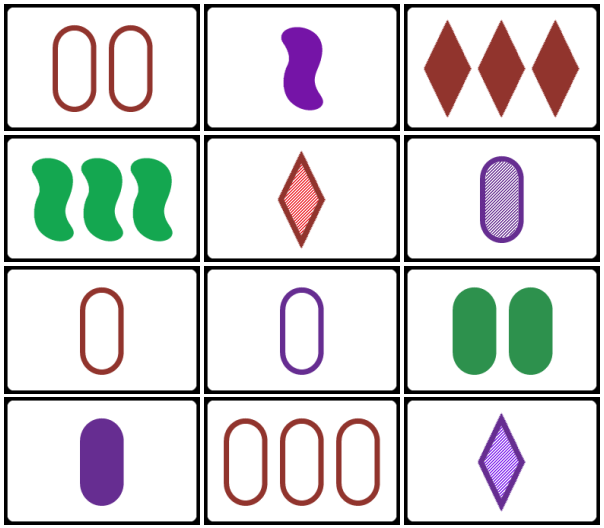


Can we solve this puzzle using SQL?

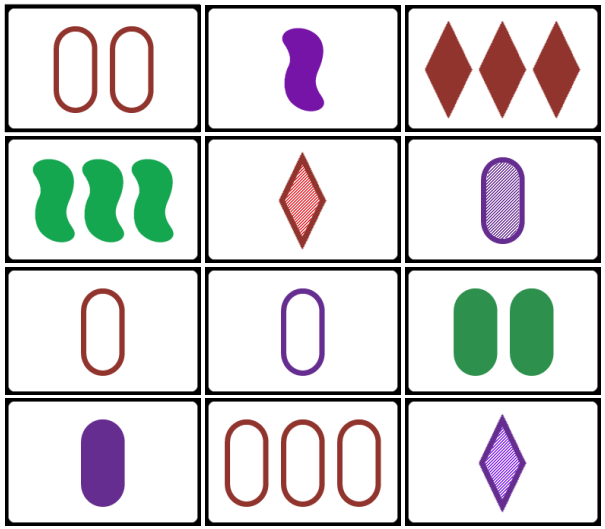


Then you would first have to convert the cards to rows.

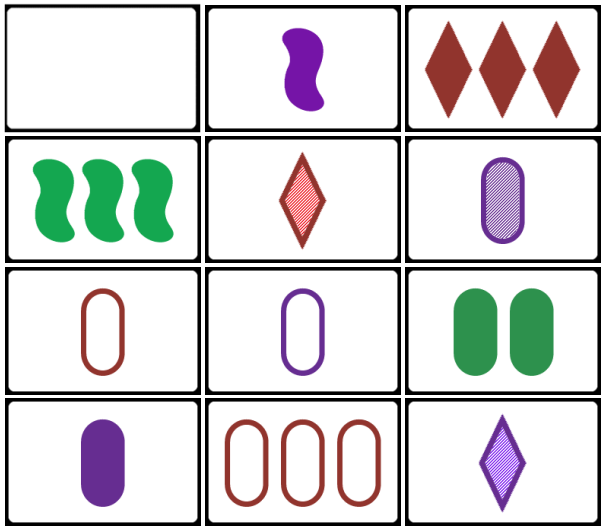
Convert Cards To Rows



Convert Cards To Rows

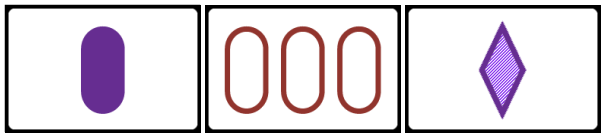


Convert Cards To Rows



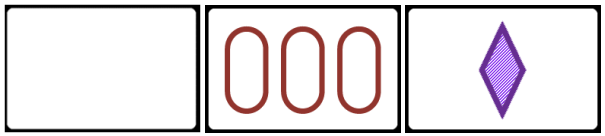
```
with cards as
 (          select 1 cardno, 'oval'     symbol, 'red'    color, 'open'    shading, 2 qty from dual
```

Convert Cards To Rows



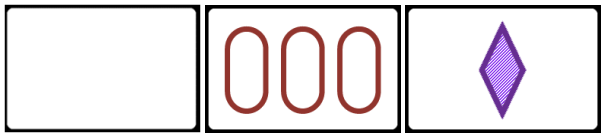
```
with cards as
 (          select 1 cardno, 'oval'     symbol, 'red'    color, 'open'    shading, 2 qty from dual
  union all select 2       , 'squiggle'       , 'purple'      , 'solid'           , 1     from dual
  union all select 3       , 'diamond'        , 'red'         , 'solid'           , 3     from dual
  union all select 4       , 'squiggle'       , 'green'       , 'solid'           , 3     from dual
  union all select 5       , 'diamond'        , 'red'         , 'striped'         , 1     from dual
  union all select 6       , 'oval'           , 'purple'      , 'striped'         , 1     from dual
  union all select 7       , 'oval'           , 'red'         , 'open'            , 1     from dual
  union all select 8       , 'oval'           , 'purple'      , 'open'            , 1     from dual
  union all select 9       , 'oval'           , 'green'       , 'solid'           , 2     from dual
```

Convert Cards To Rows



```
with cards as
 (          select 1 cardno, 'oval'     symbol, 'red'    color, 'open'    shading, 2 qty from dual
  union all select 2       , 'squiggle'       , 'purple'      , 'solid'           , 1     from dual
  union all select 3       , 'diamond'        , 'red'         , 'solid'           , 3     from dual
  union all select 4       , 'squiggle'       , 'green'       , 'solid'           , 3     from dual
  union all select 5       , 'diamond'        , 'red'         , 'striped'         , 1     from dual
  union all select 6       , 'oval'           , 'purple'      , 'striped'         , 1     from dual
  union all select 7       , 'oval'           , 'red'         , 'open'            , 1     from dual
  union all select 8       , 'oval'           , 'purple'      , 'open'            , 1     from dual
  union all select 9       , 'oval'           , 'green'       , 'solid'           , 2     from dual
                             oval               purple          solid               1
```

Convert Cards To Rows



```
with cards as
 (          select 1 cardno, 'oval'     symbol, 'red'    color, 'open'    shading, 2 qty from dual
  union all select 2       , 'squiggle'       , 'purple'      , 'solid'           , 1     from dual
  union all select 3       , 'diamond'        , 'red'         , 'solid'           , 3     from dual
  union all select 4       , 'squiggle'       , 'green'       , 'solid'           , 3     from dual
  union all select 5       , 'diamond'        , 'red'         , 'striped'         , 1     from dual
  union all select 6       , 'oval'           , 'purple'      , 'striped'         , 1     from dual
  union all select 7       , 'oval'           , 'red'         , 'open'            , 1     from dual
  union all select 8       , 'oval'           , 'purple'      , 'open'            , 1     from dual
  union all select 9       , 'oval'           , 'green'       , 'solid'           , 2     from dual
  union all select 10      , 'oval'           , 'purple'      , 'solid'           , 1     from dual
```

Convert Cards To Rows

```
with cards as
 (          select 1 cardno, 'oval'     symbol, 'red'    color, 'open'    shading, 2 qty from dual
  union all select 2       , 'squiggle'       , 'purple'      , 'solid'           , 1     from dual
  union all select 3       , 'diamond'        , 'red'         , 'solid'           , 3     from dual
  union all select 4       , 'squiggle'       , 'green'       , 'solid'           , 3     from dual
  union all select 5       , 'diamond'        , 'red'         , 'striped'         , 1     from dual
  union all select 6       , 'oval'           , 'purple'      , 'striped'         , 1     from dual
  union all select 7       , 'oval'           , 'red'         , 'open'            , 1     from dual
  union all select 8       , 'oval'           , 'purple'      , 'open'            , 1     from dual
  union all select 9       , 'oval'           , 'green'       , 'solid'           , 2     from dual
  union all select 10      , 'oval'           , 'purple'      , 'solid'           , 1     from dual
  union all select 11      , 'oval'           , 'red'         , 'open'            , 3     from dual
  union all select 12      , 'diamond'        , 'purple'      , 'striped'         , 1     from dual)
select * from cards
/
```

Now that you have the cards converted to rows, convert the rules to predicates.

Convert Rules To Predicates

Now that you have the cards converted to rows, convert the rules to predicates.

A set consists of three cards, so draw three cards from the rows

```
select *
  from       cards cards1
 cross join cards cards2
 cross join cards cards3
```

Convert Rules To Predicates

Now that you have the cards converted to rows, convert the rules to predicates.

A set consists of three cards, so draw three cards from the rows

You cannot draw the same card more than once, so remove these combinations.


```
select *
  from      cards cards1
 cross join cards cards2
 cross join cards cards3
 where cards1.cardno <> cards2.cardno
   and cards2.cardno <> cards3.cardno
   and cards3.cardno <> cards1.cardno
```

Convert Rules To Predicates

A set consists of three cards, so draw three cards from the rows

You cannot draw the same card more than once, so remove these combinations.

Combinations like 1-2-3, 2-3-1, 1-3-2 etc are the same, so we need to remove those as well


```
select *
  from       cards cards1
 cross join cards cards2
 cross join cards cards3
 where cards1.cardno <> cards2.cardno
   and cards2.cardno <> cards3.cardno
   and cards3.cardno <> cards1.cardno
```

Convert Rules To Predicates

A set consists of three cards, so draw three cards from the rows

You cannot draw the same card more than once, so remove these combinations.

Combinations like 1-2-3, 2-3-1, 1-3-2 etc are the same, so we need to remove those as well


```
select *
  from      cards cards1
 cross join cards cards2
 cross join cards cards3
 where cards1.cardno <  cards2.cardno
   and cards2.cardno <  cards3.cardno
```

Convert Rules To Predicates

```
select *
  from       cards cards1
 cross join cards cards2
 cross join cards cards3
 where cards1.cardno <  cards2.cardno
   and cards2.cardno <  cards3.cardno
```

All the same

Now that we have all combinations of three cards.

All the same

Now that we have all combinations of three cards.

We should select only those combinations where the feature is All The Same

```
(    cards1.feature =  cards2.feature
 and cards2.feature =  cards3.feature
)
```

All the same OR All different

Now that we have all combinations of three cards.

We should select only those combinations where the feature is All The Same

OR they are All Different

```
    (    cards1.feature =  cards2.feature
     and cards2.feature =  cards3.feature
    )
or  (    cards1.feature <> cards2.feature
     and cards2.feature <> cards3.feature
     and cards3.feature <> cards1.feature
    )
```

All the same OR All different

We should select only those combinations where the feature is All The Same

OR they are All Different

And this should be done for all four features.

```
    (    cards1.feature =  cards2.feature
     and cards2.feature =  cards3.feature
    )
or  (    cards1.feature <> cards2.feature
     and cards2.feature <> cards3.feature
     and cards3.feature <> cards1.feature
    )
```

All the same OR All different

And this should be done for all four features.

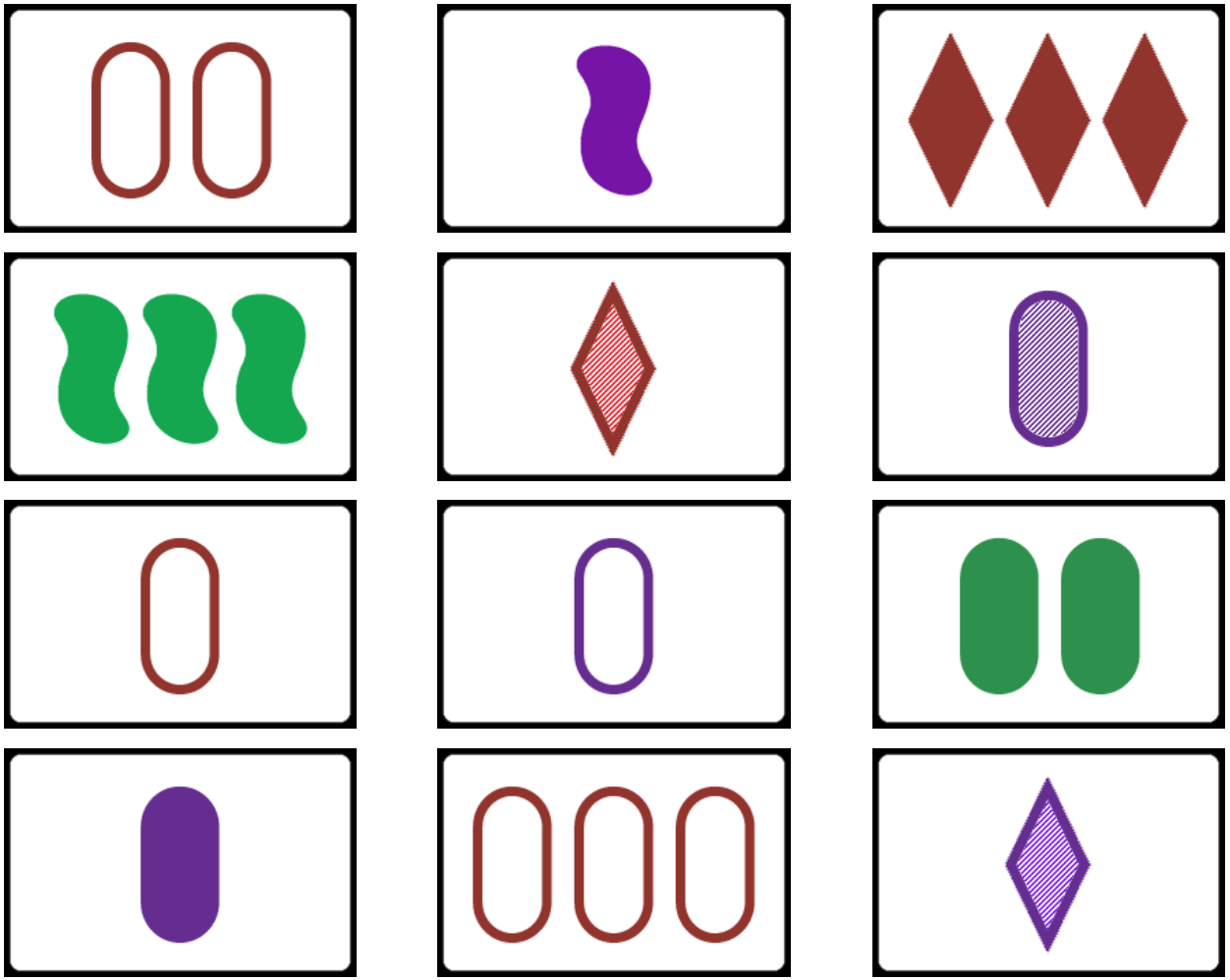
```
    (    cards1.feature =  cards2.feature
     and cards2.feature =  cards3.feature
    )
or  (    cards1.feature <> cards2.feature
     and cards2.feature <> cards3.feature
     and cards3.feature <> cards1.feature
    )
```

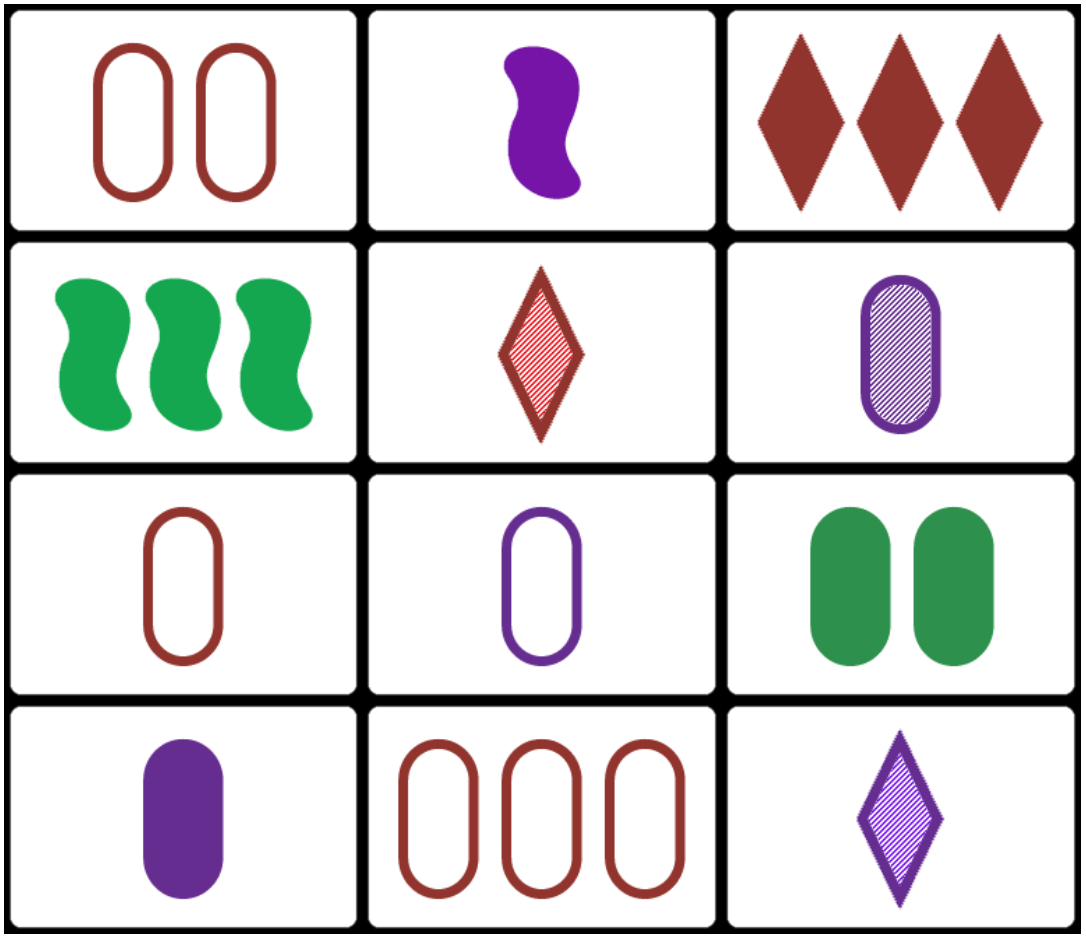
All the same OR All different

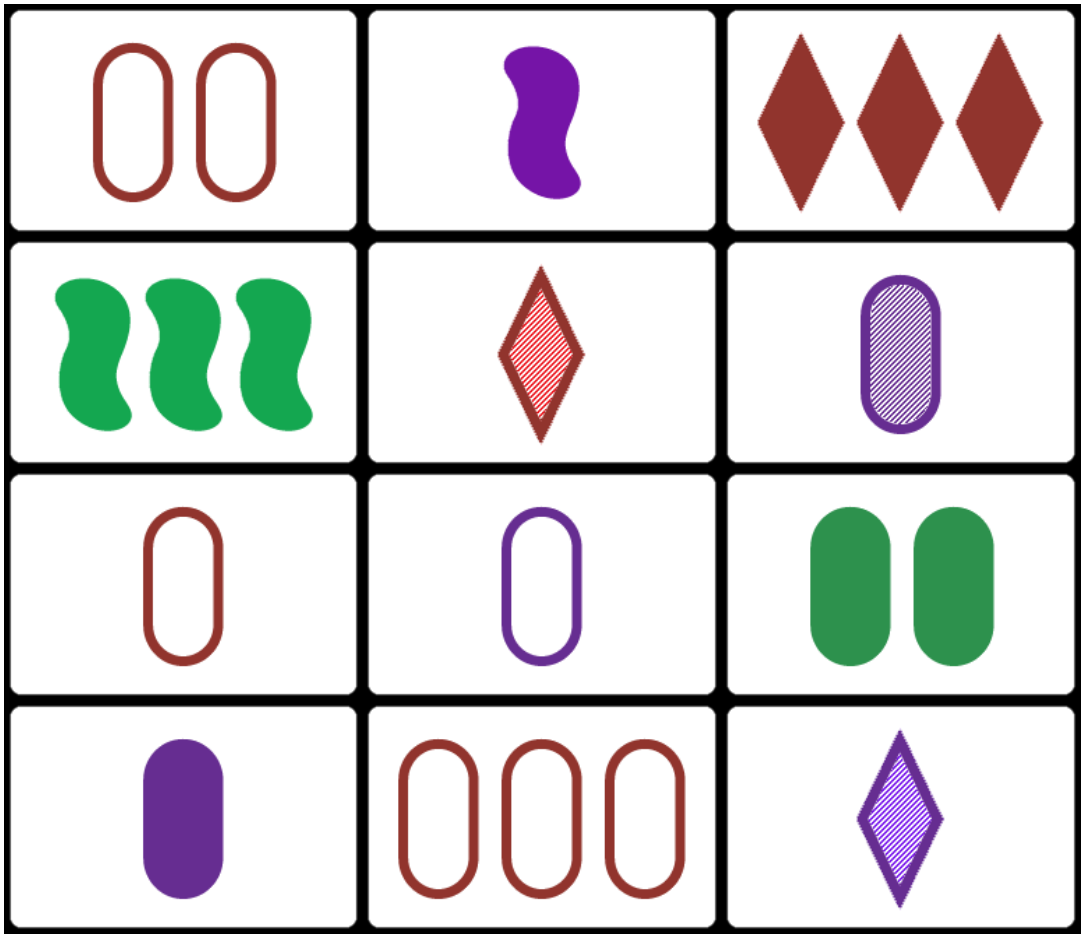
And this should be done for all four features.

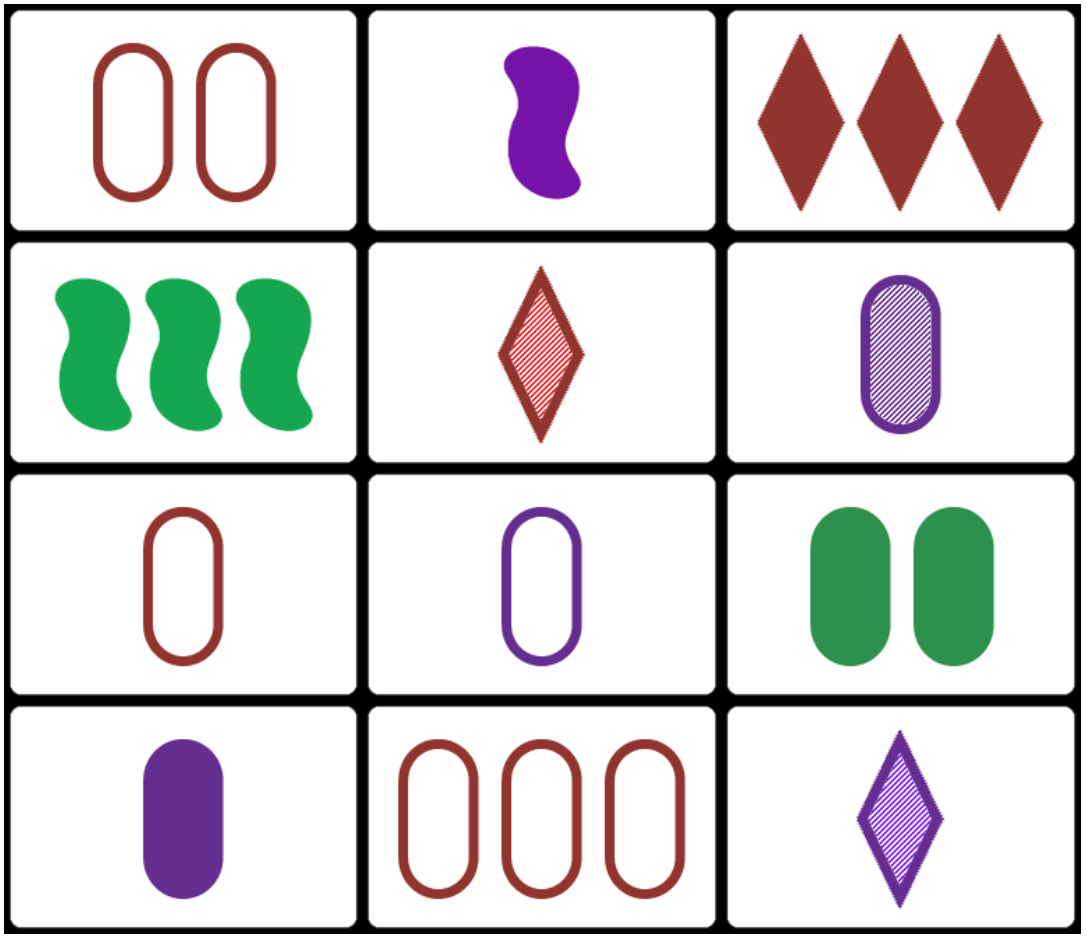
```
and (   (    cards1.texture =  cards2.texture
         and cards2.texture =  cards3.texture
        )
     or (    cards1.texture <> cards2.texture
         and cards2.texture <> cards3.texture
         and cards3.texture <> cards1.texture
        )
    )
and (   (    cards1.symbol  =  cards2.symbol
         and cards2.symbol  =  cards3.symbol
        )
     or (    cards1.symbol  <> cards2.symbol
         and cards2.symbol  <> cards3.symbol
         and cards3.symbol  <> cards1.symbol
        )
    )
```

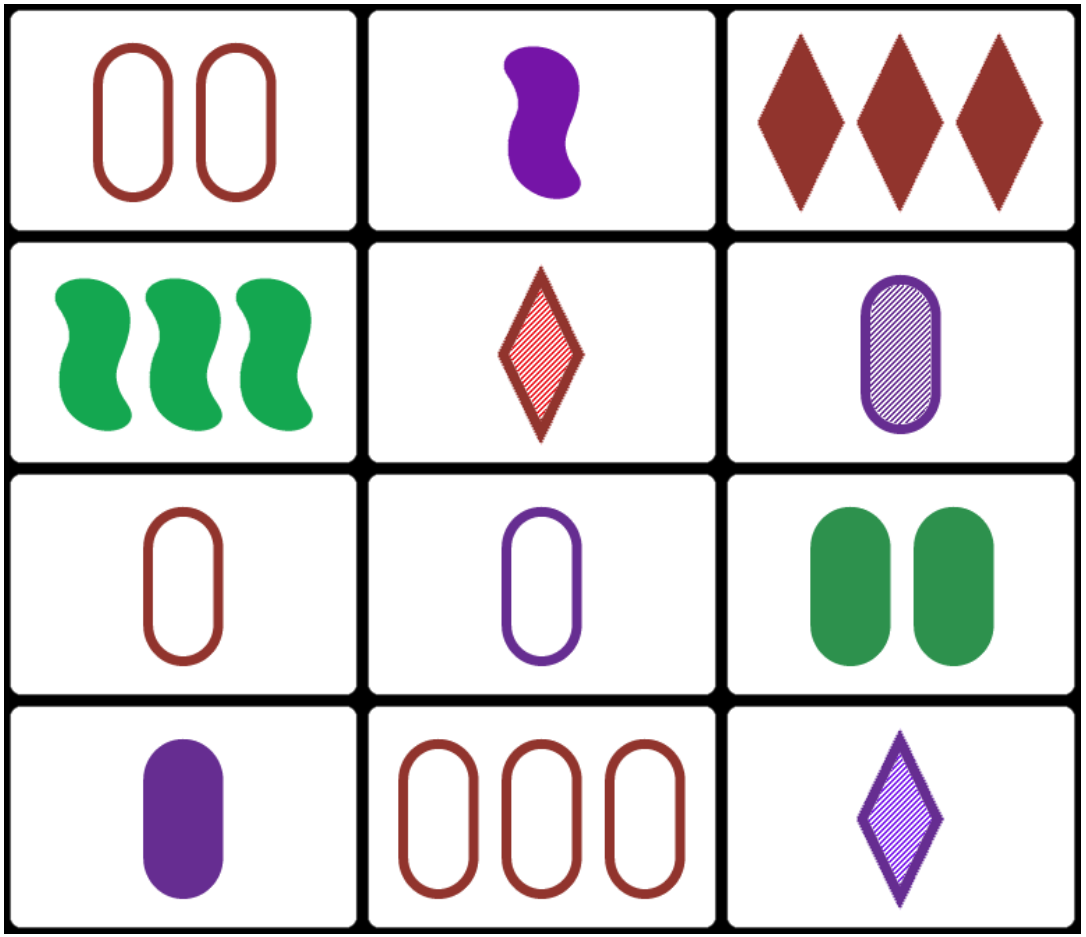
```
and (   (    cards1.shading =  cards2.shading
         and cards2.shading =  cards3.shading
        )
     or (    cards1.shading <> cards2.shading
         and cards2.shading <> cards3.shading
         and cards3.shading <> cards1.shading
        )
    )
and (   (    cards1.color   =  cards2.color
         and cards2.color   =  cards3.color
        )
     or (    cards1.color   <> cards2.color
         and cards2.color   <> cards3.color
         and cards3.color   <> cards1.color
        )
    )
```

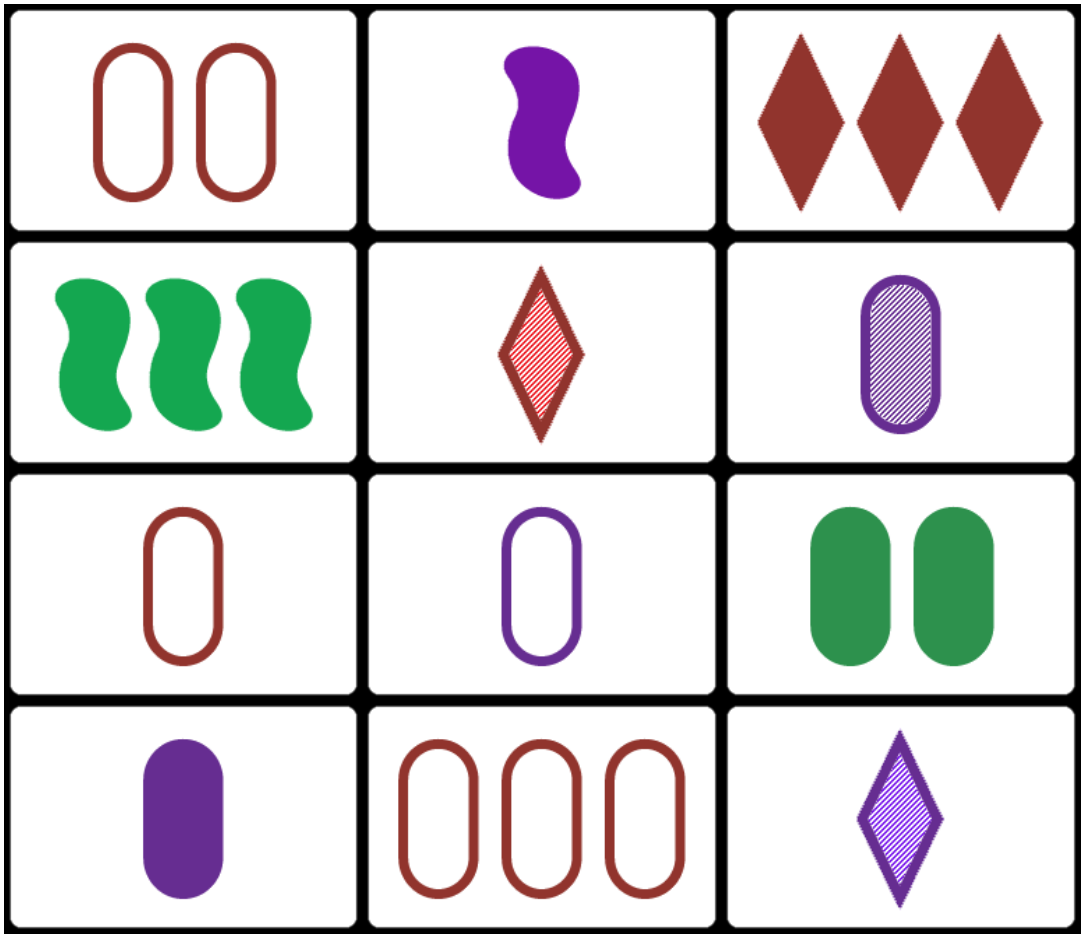


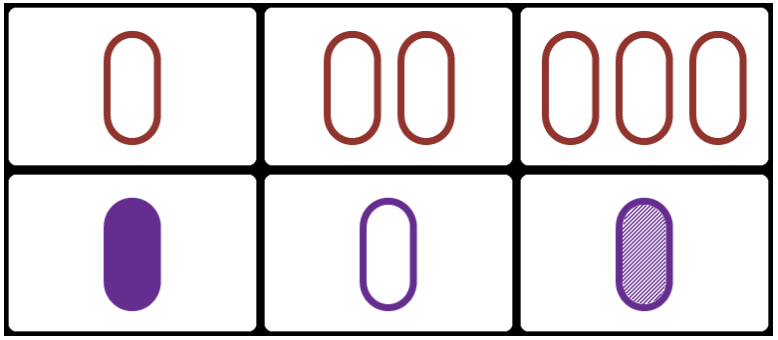


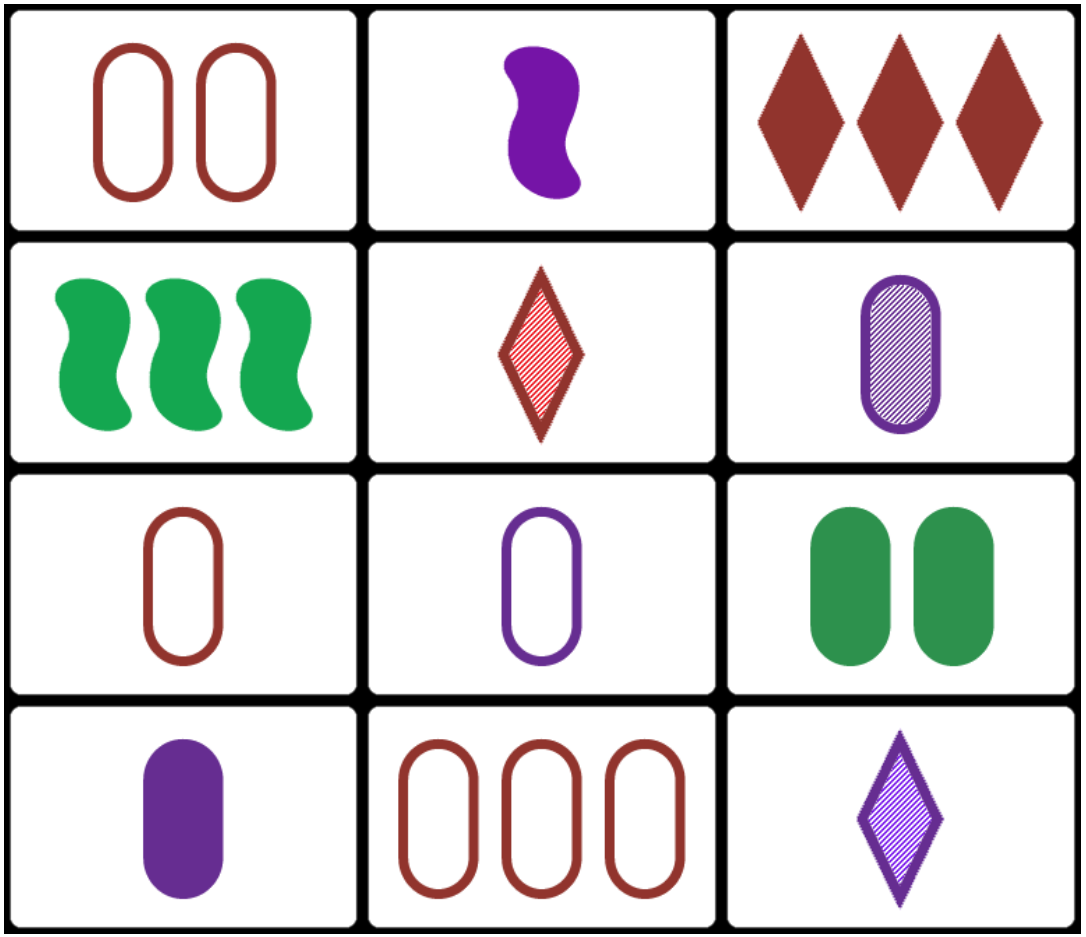


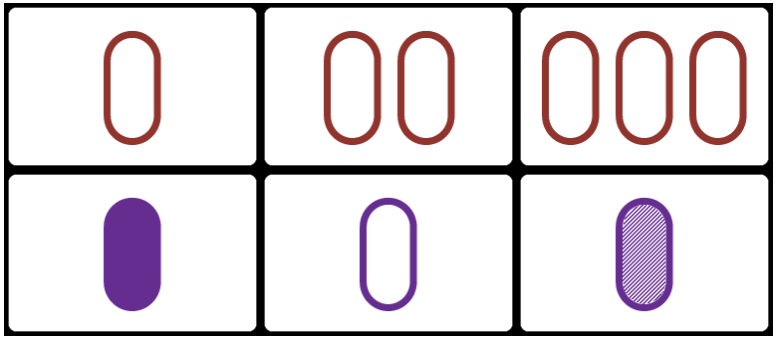


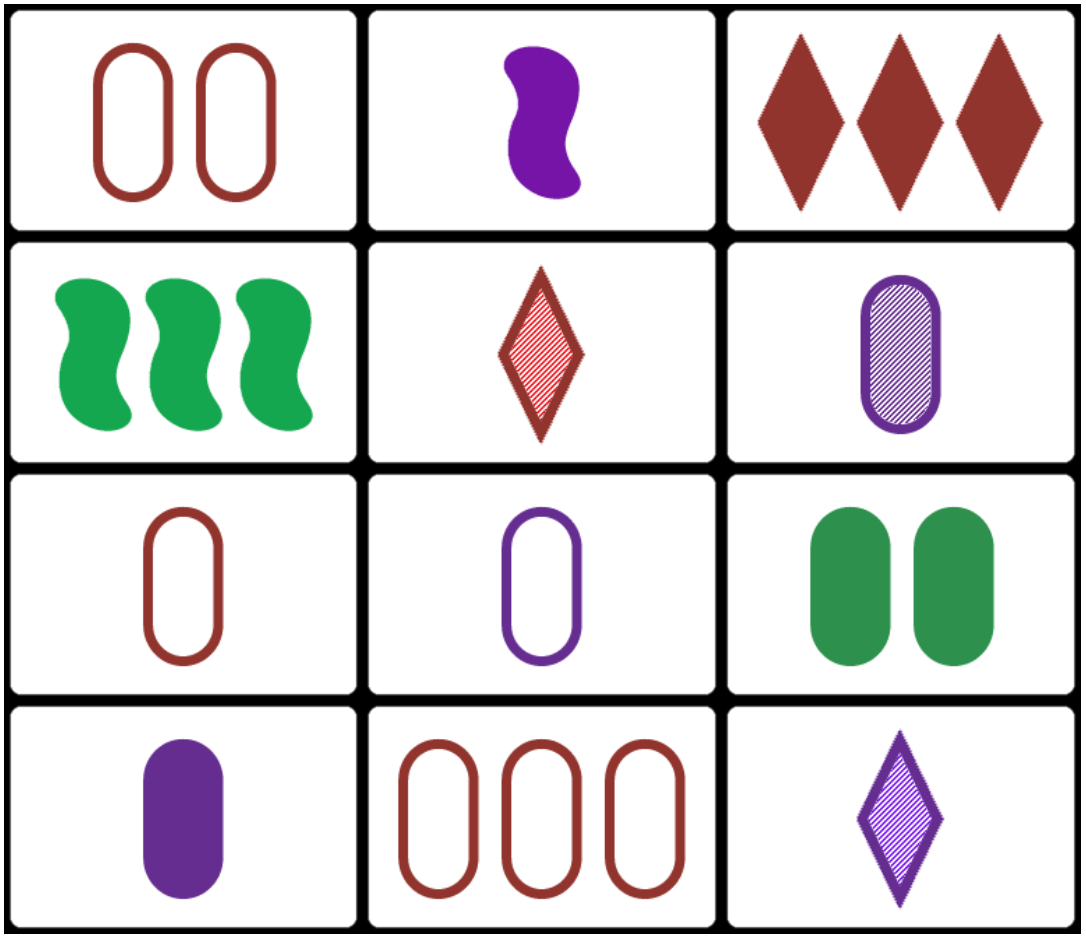


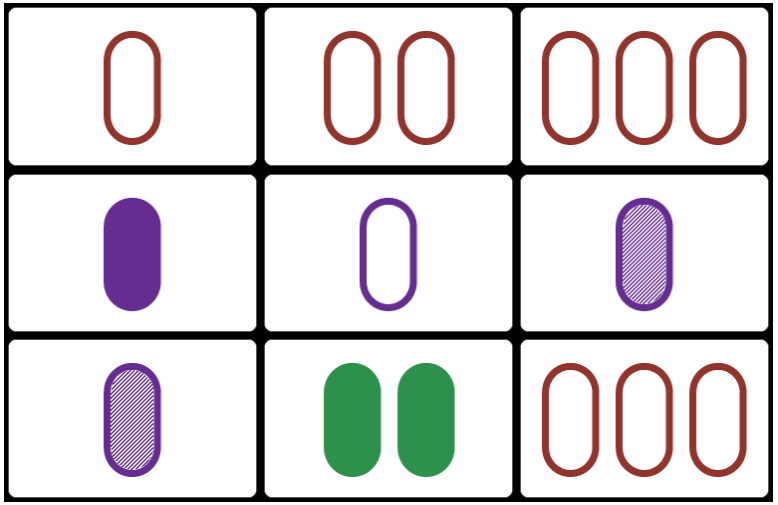


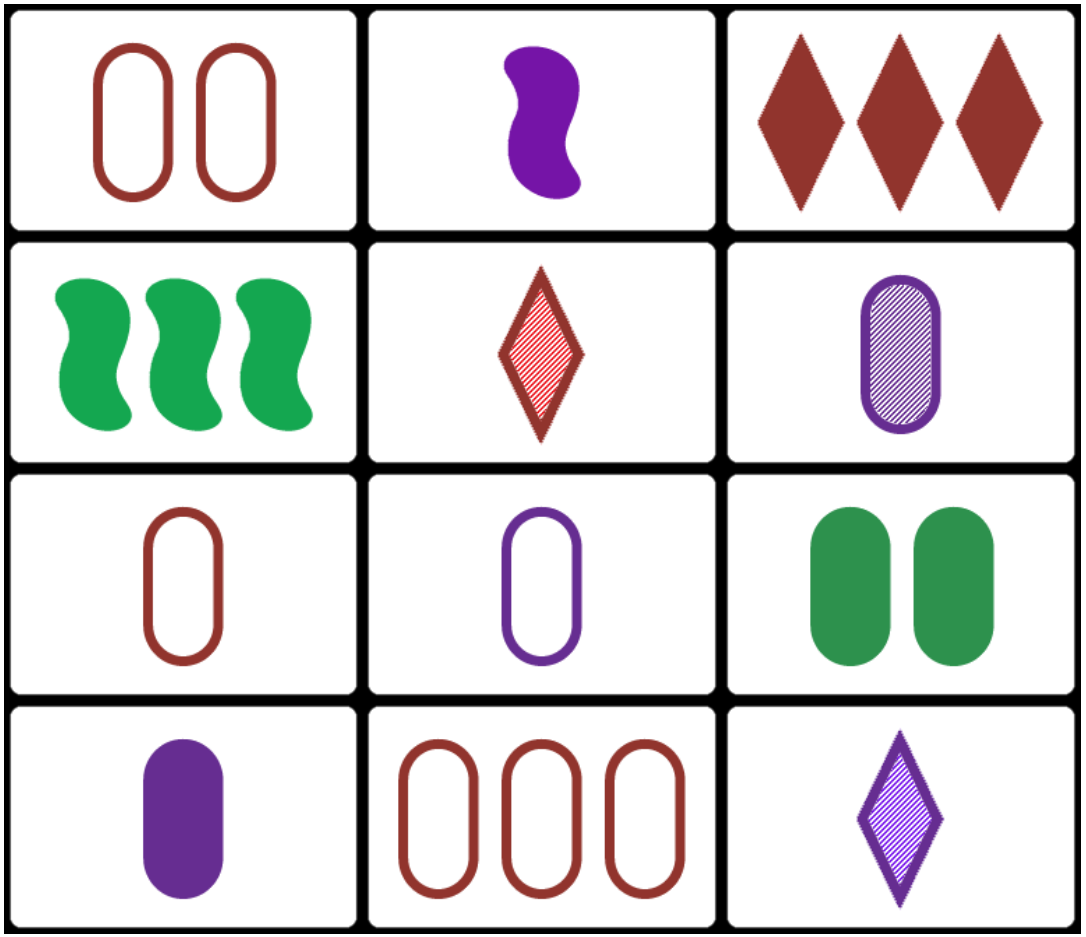


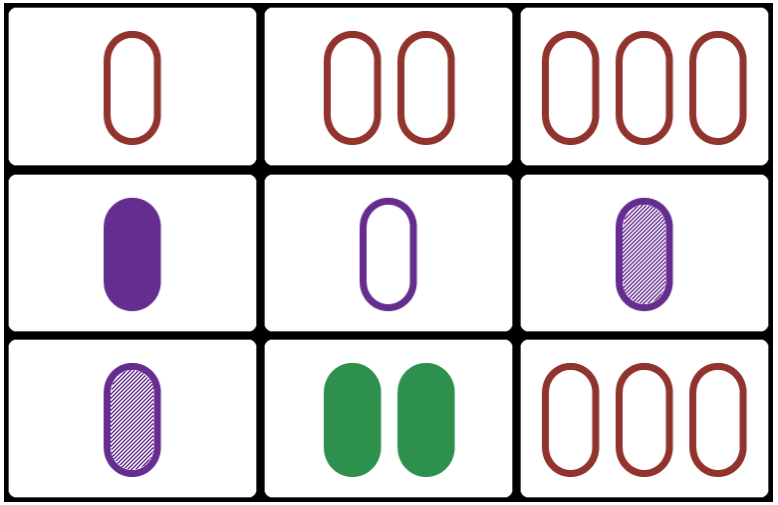


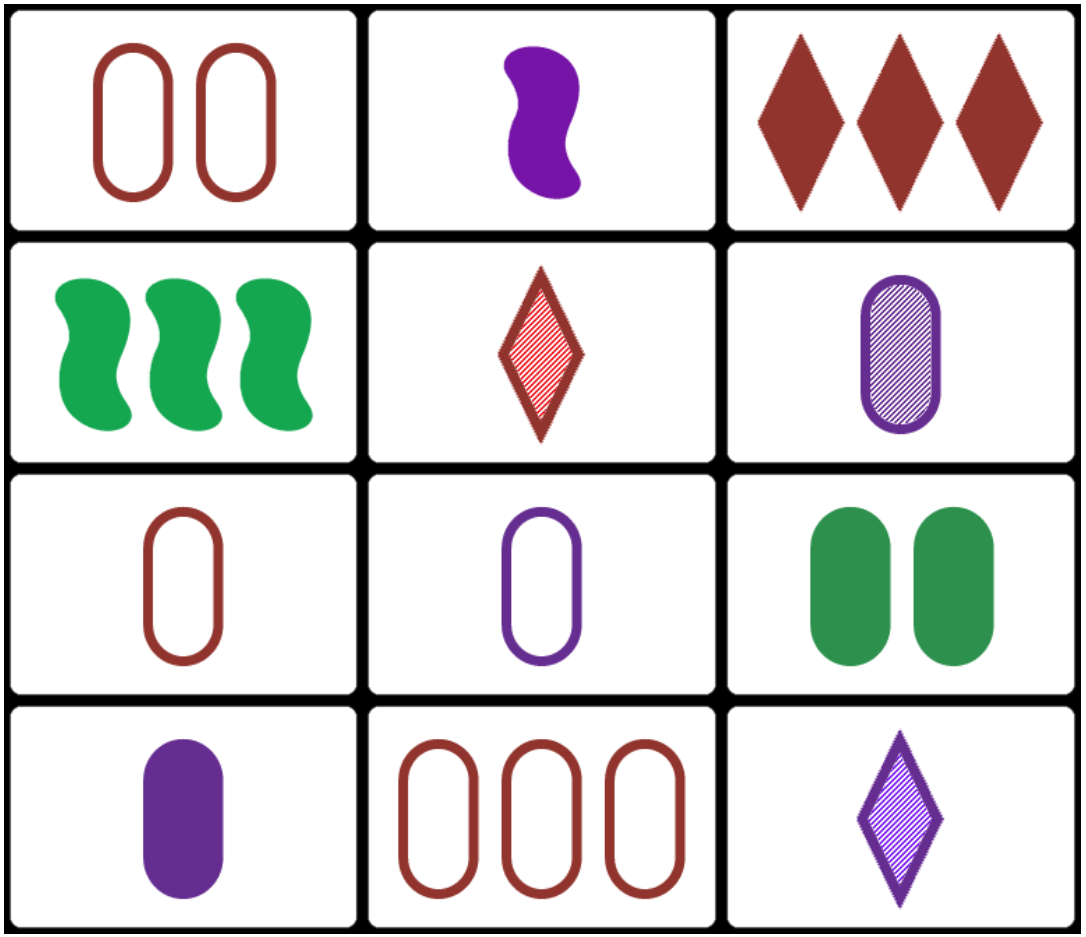


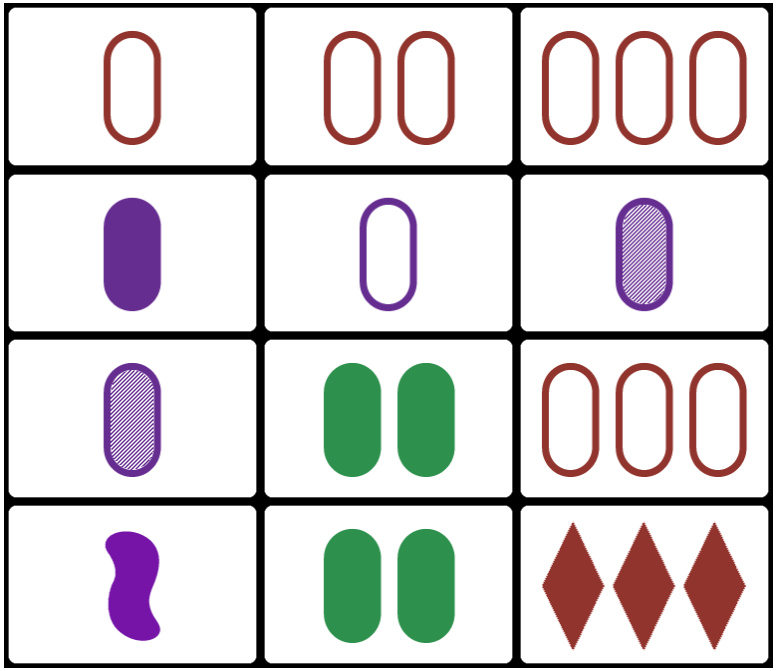


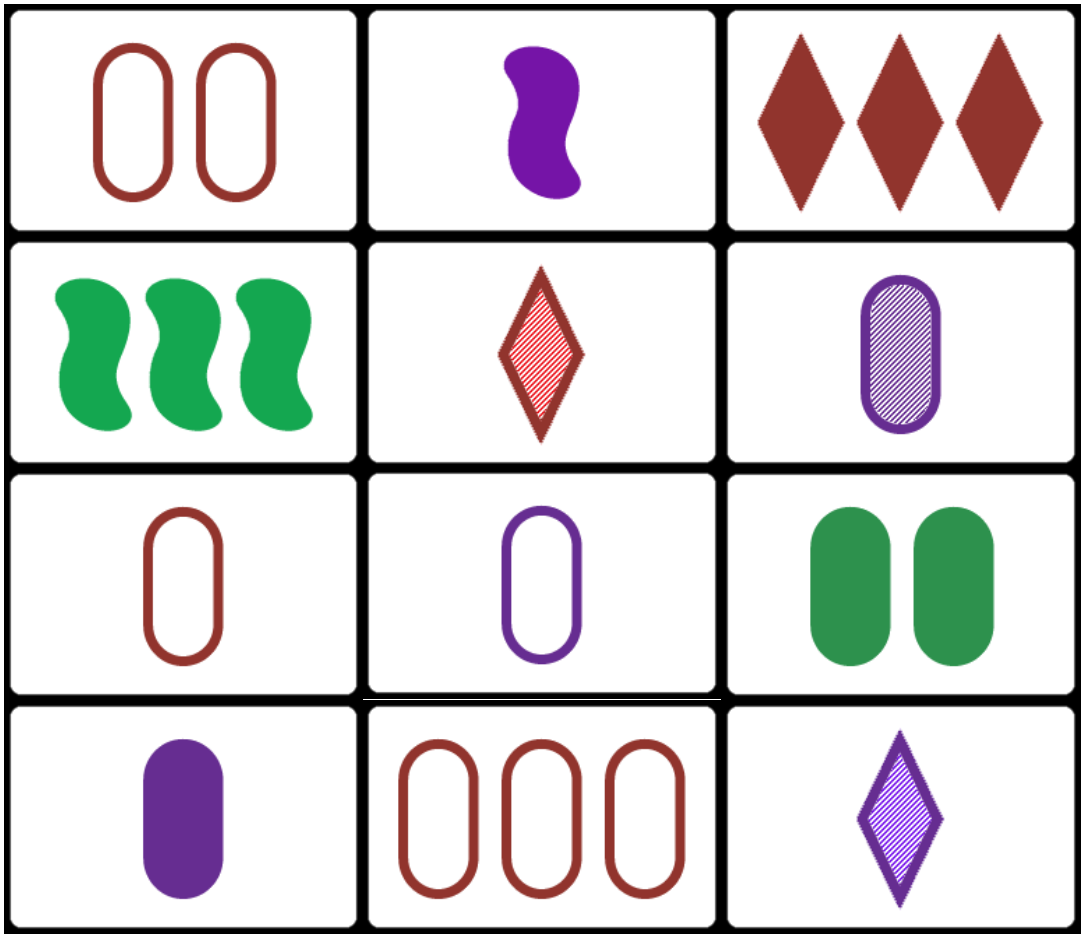


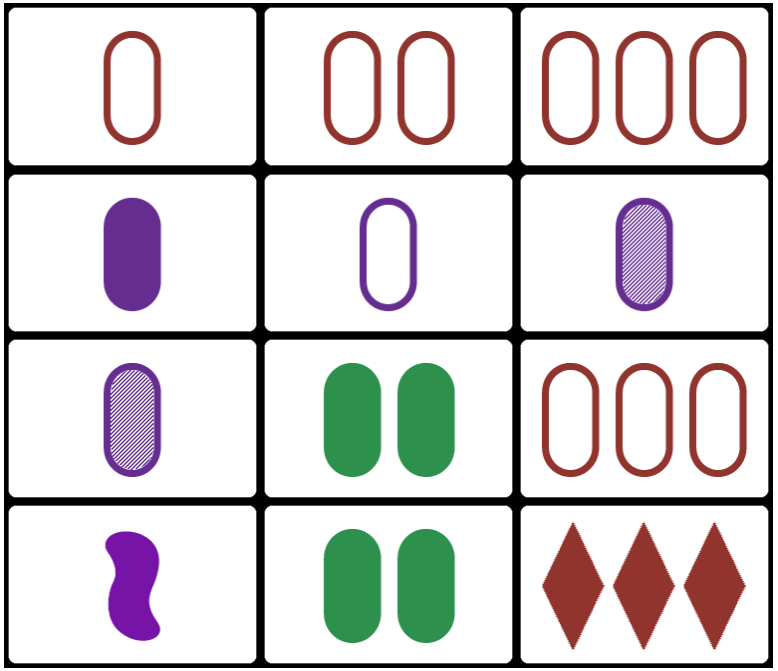


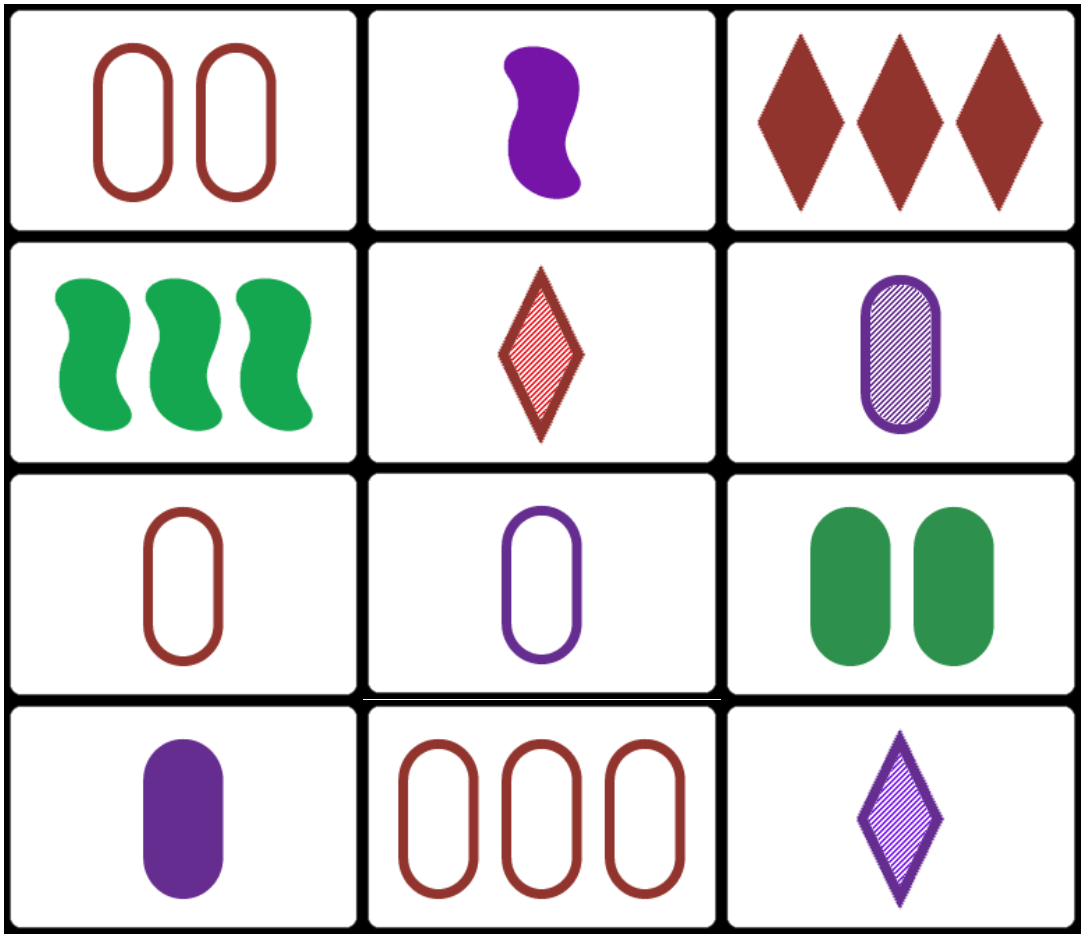


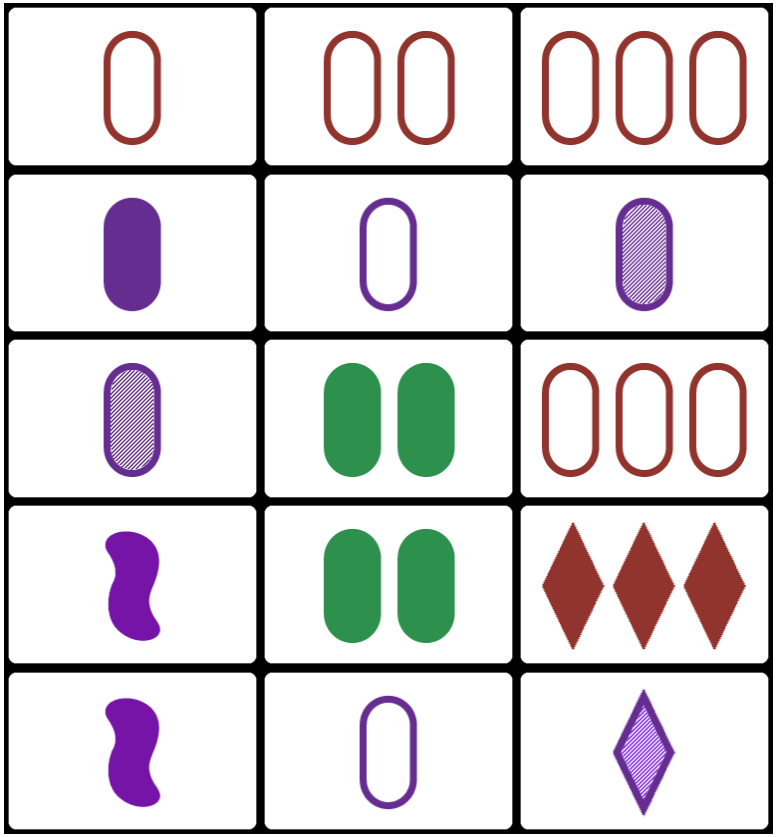


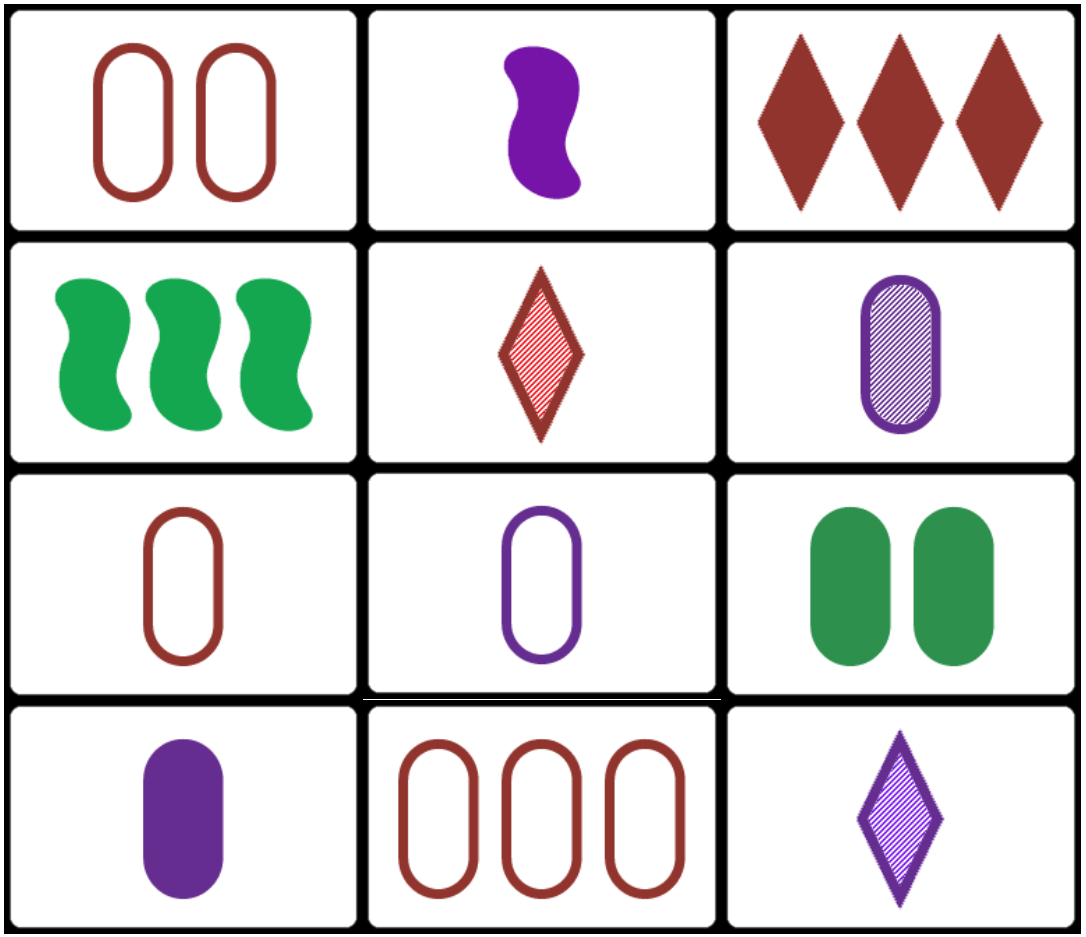


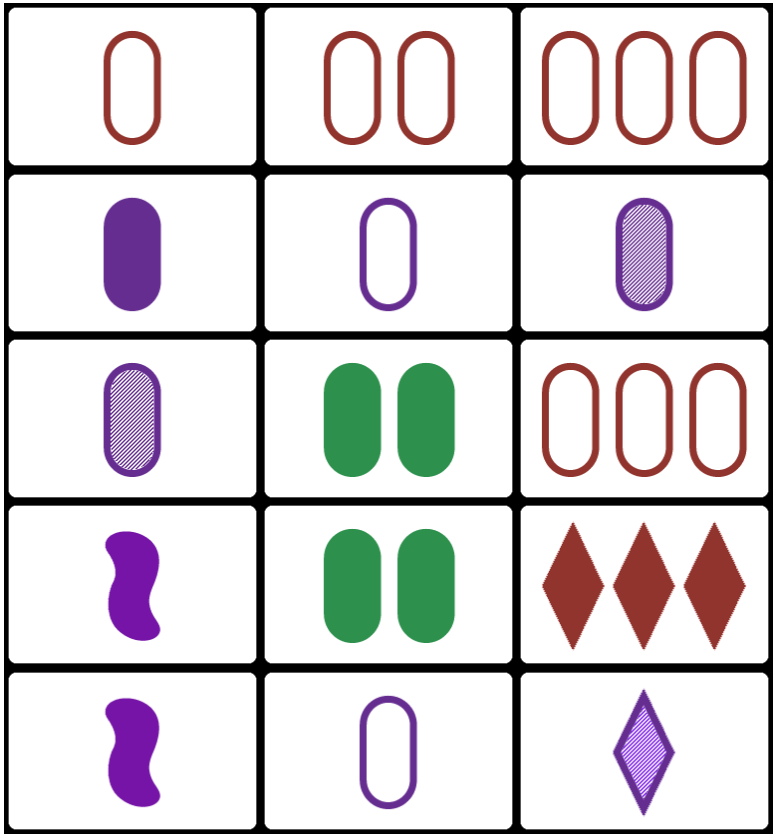


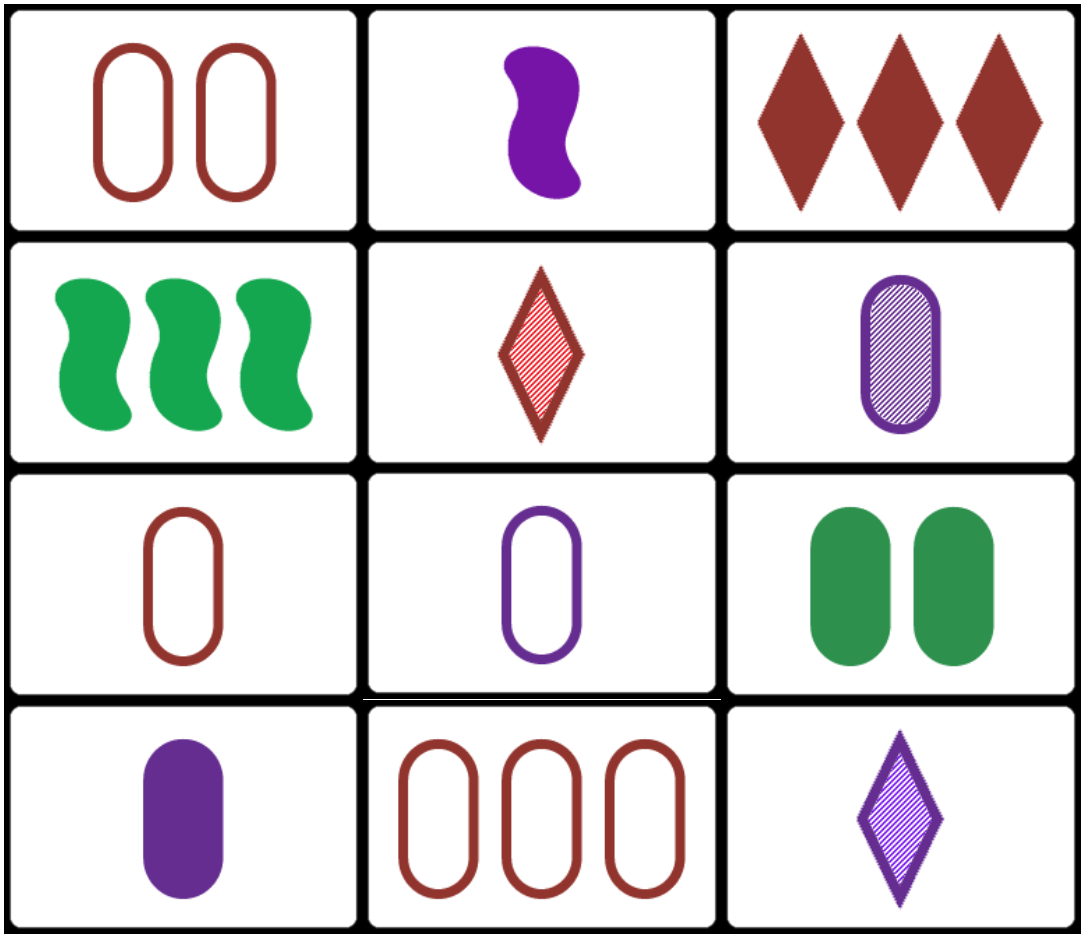


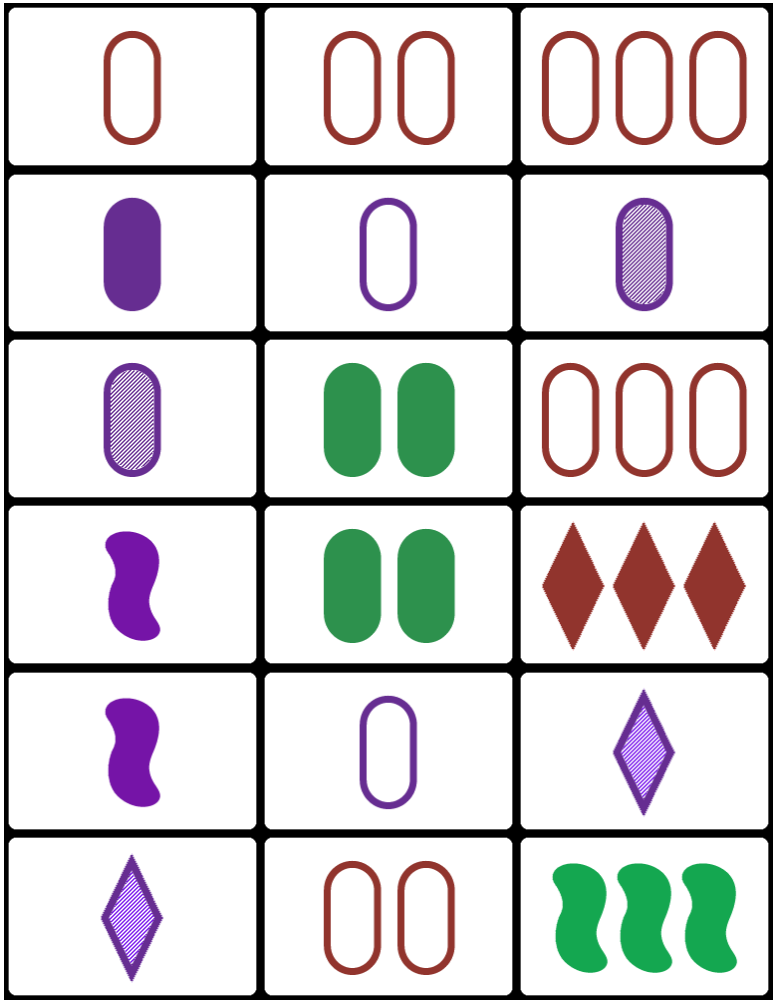








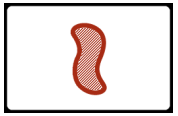




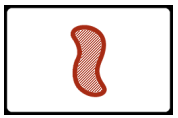
That looks good, but it seems like a lot of work

Can we also use SQL to generate a puzzle

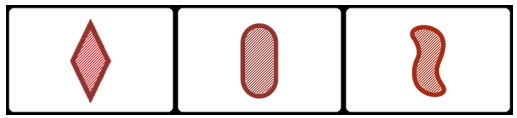
Sure, you can easily use SQL to generate rows, like we did before



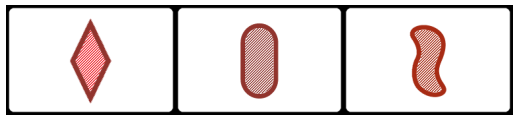
Each card has one of three shapes: diamond, oval, or squiggle



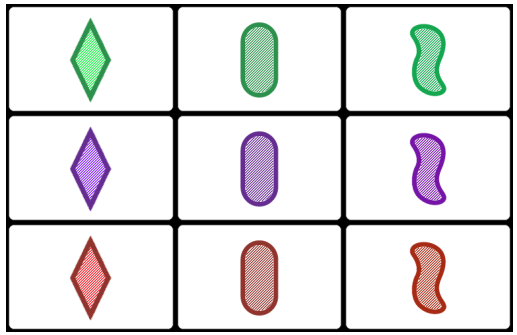
Each card has one of three shapes: diamond, oval, or squiggle



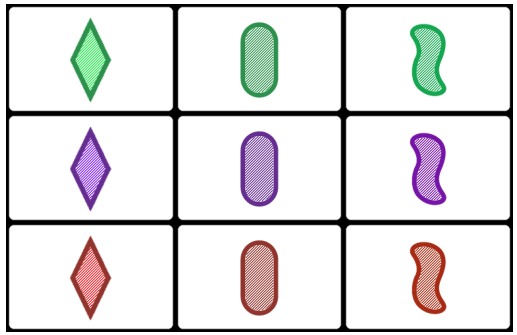
Then there are three colors: green, purple, or red



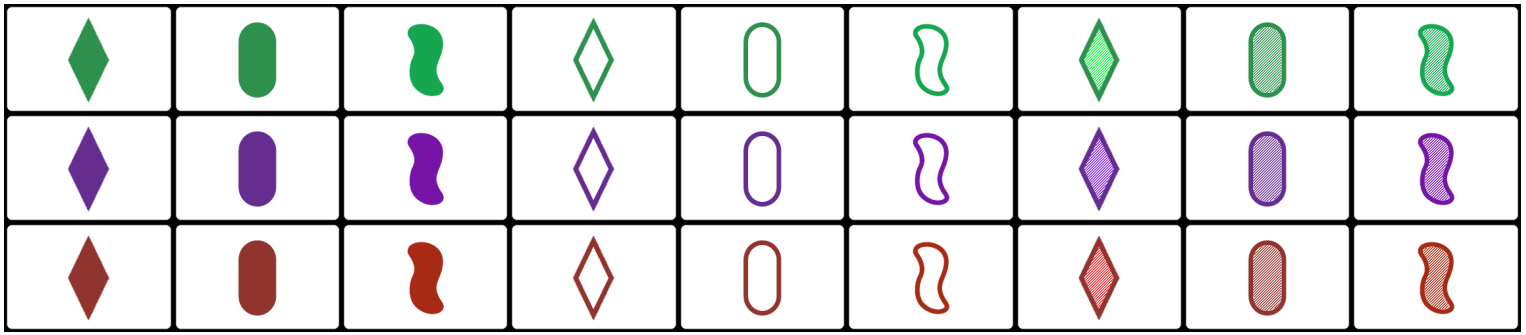
Then there are three colors: green, purple, or red



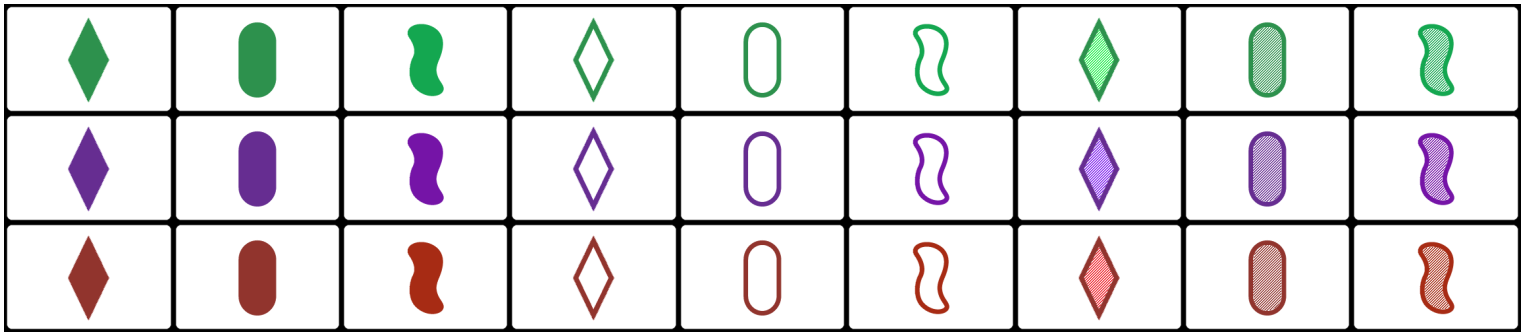
There are also three fillings: solid, open, or striped



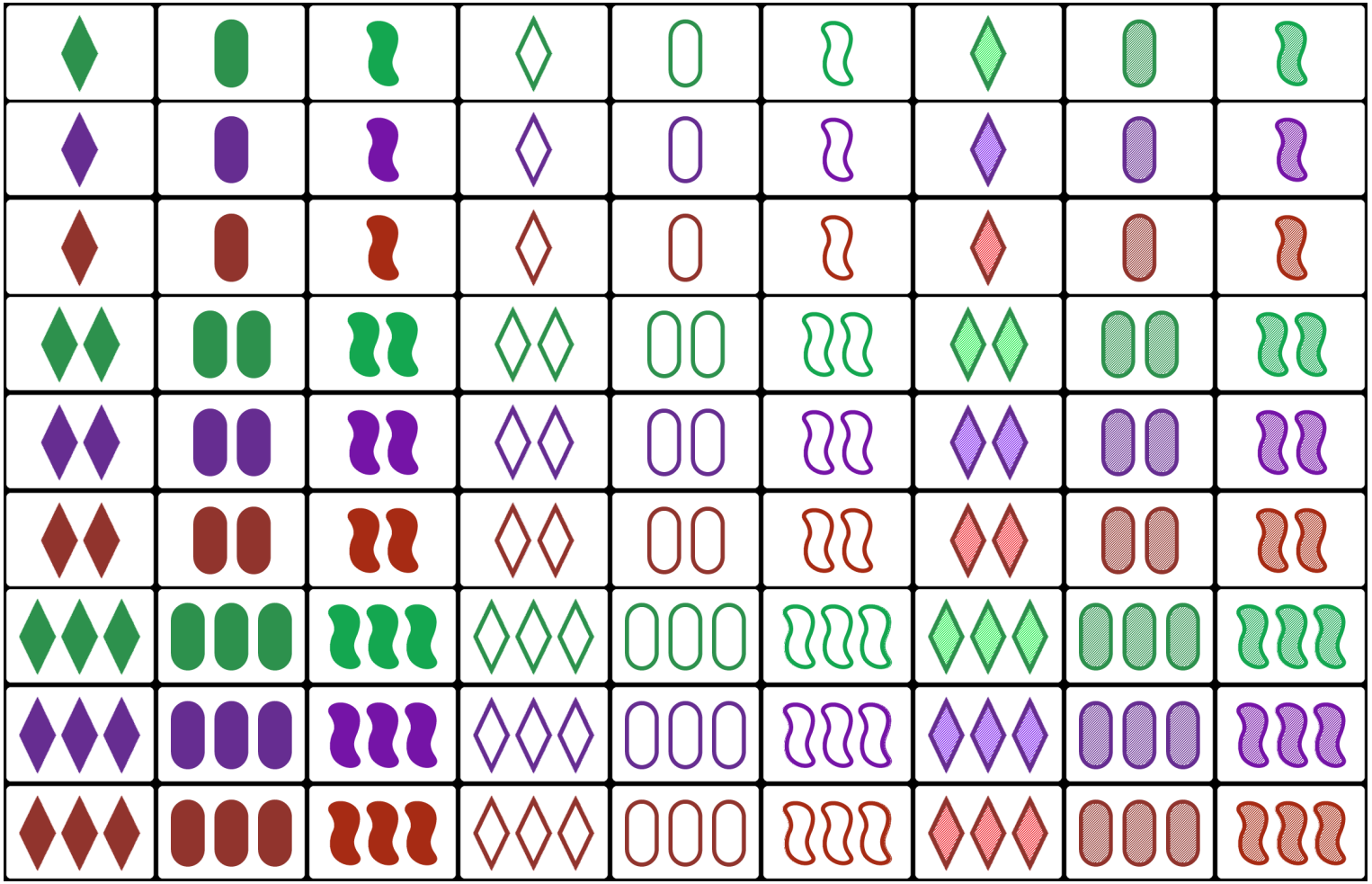
There are also three fillings: solid, open, or striped



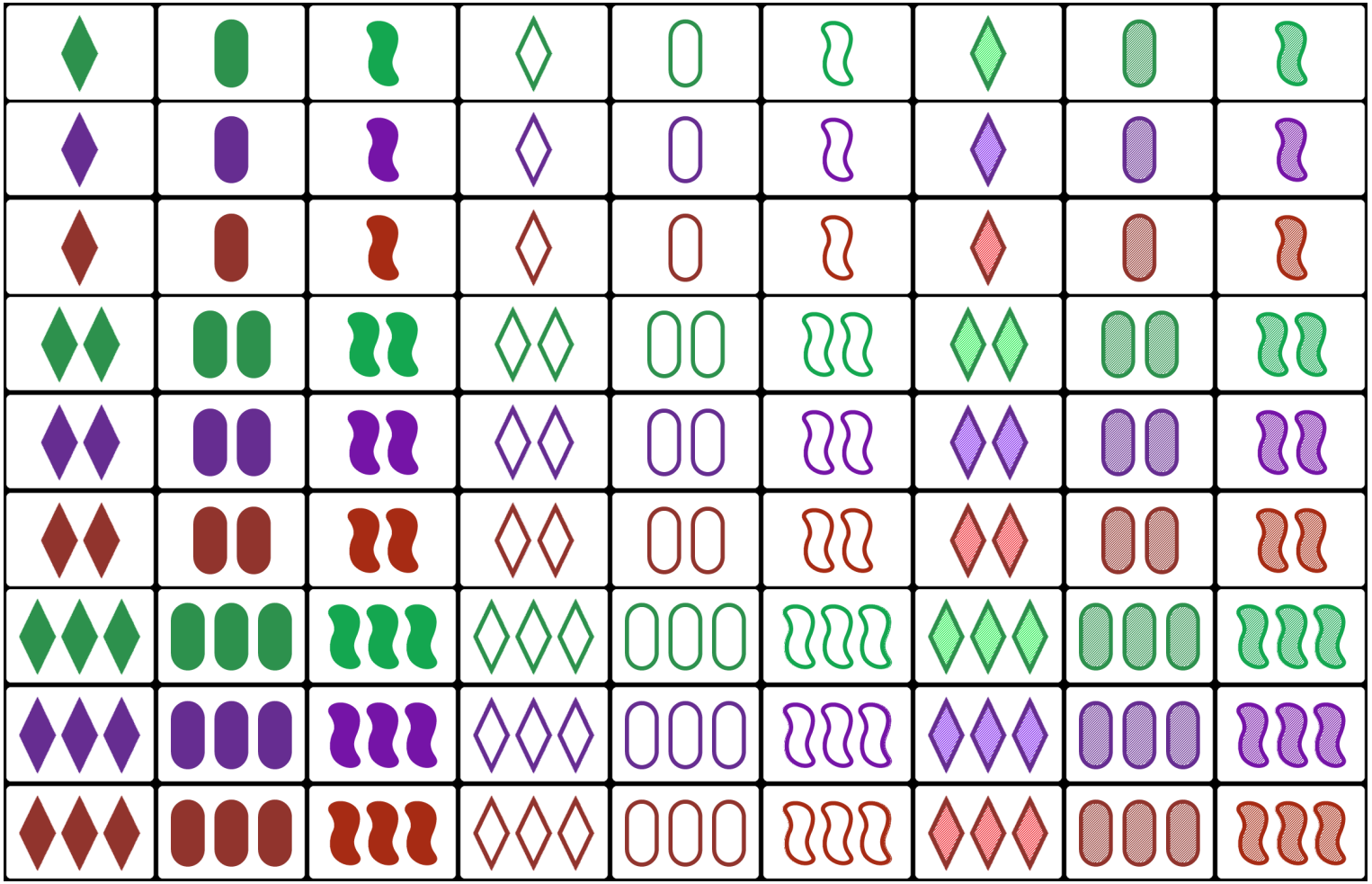
And three different counts of symbols: one, two, or three



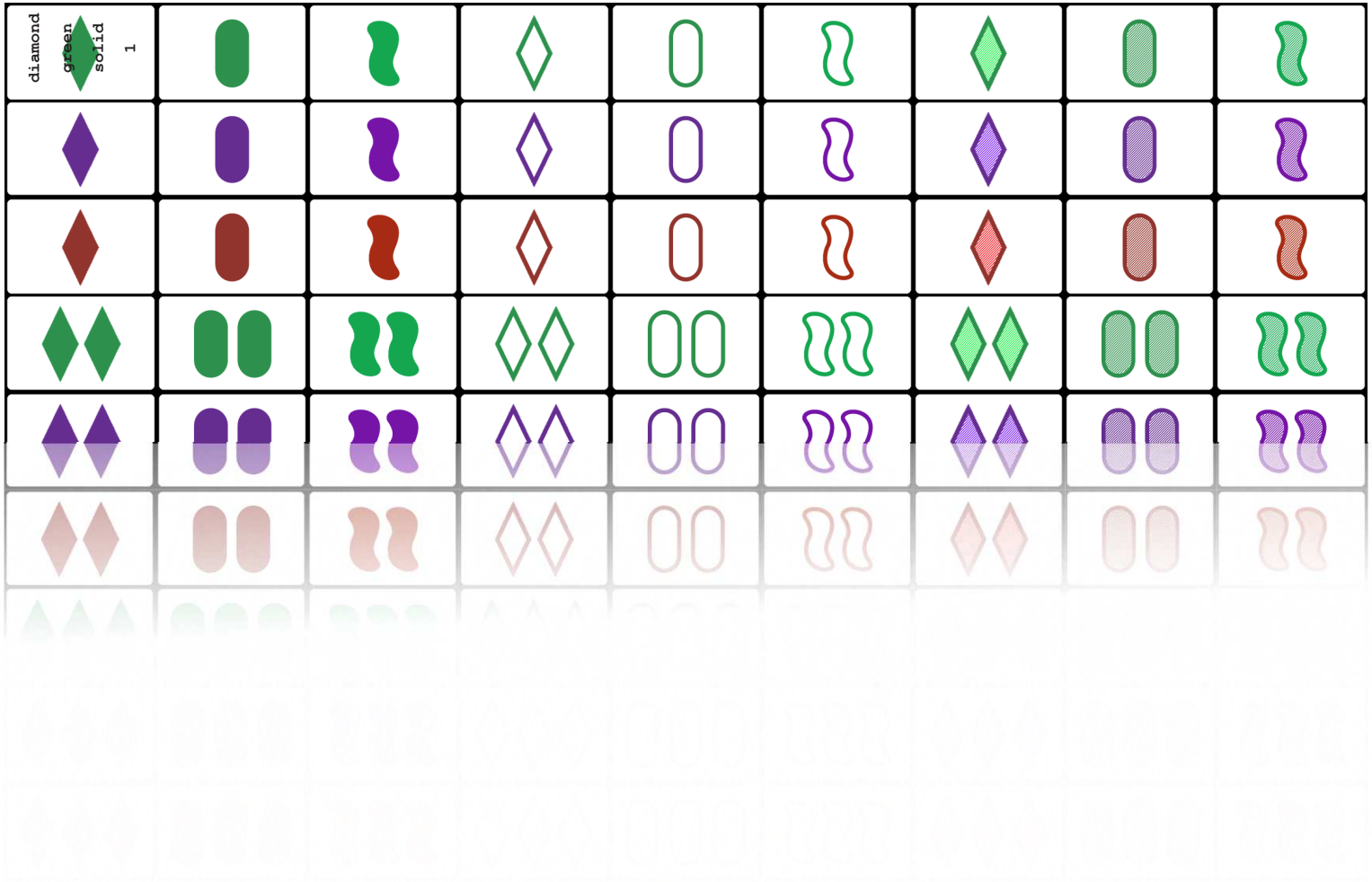
And three different counts of symbols: one, two, or three



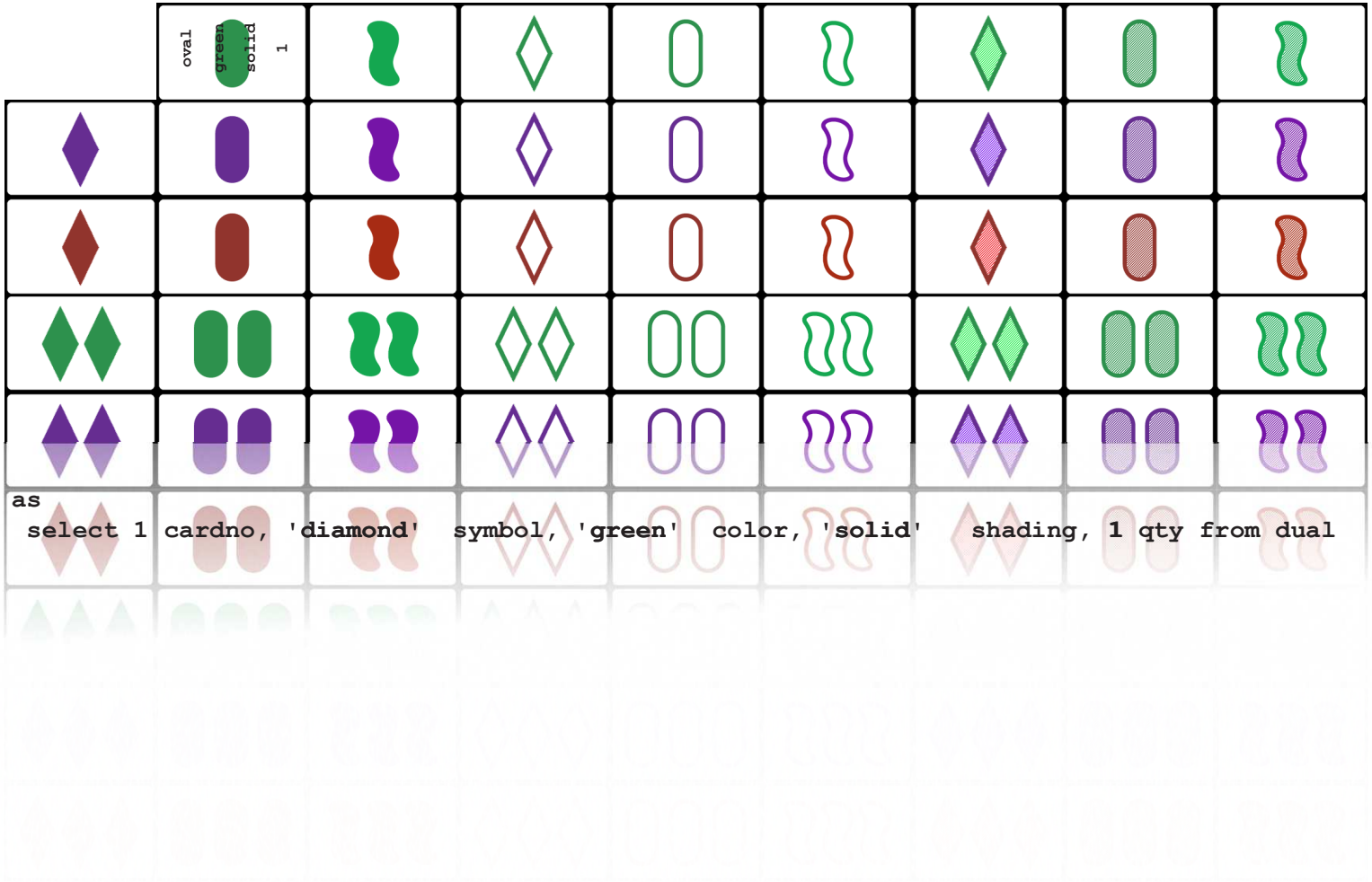
And three different counts of symbols: one, two, or three



Then we convert these cards to rows.

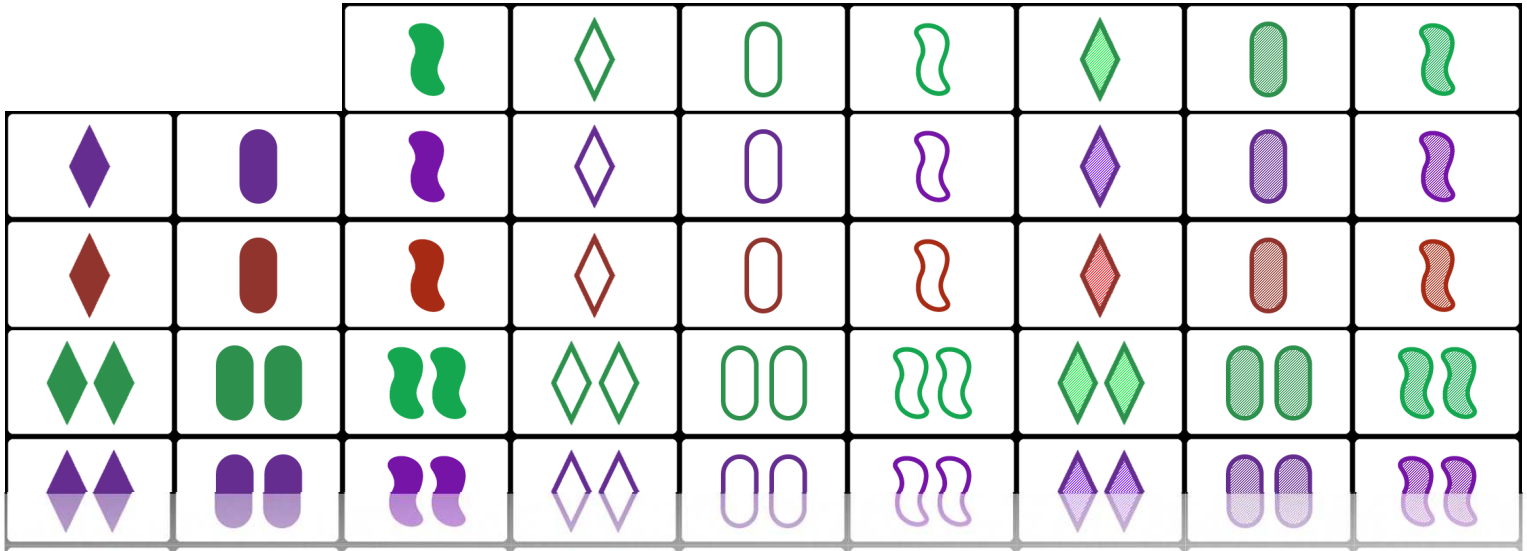


Then we convert these cards to rows.




```
with cards as
 (
   select 1 cardno, 'diamond' symbol, 'green' color, 'solid' shading, 1 qty from dual
```

Then we convert these cards to rows.



```
with cards as
 (          select 1 cardno, 'diamond'  symbol, 'green'  color, 'solid'   shading, 1 qty from dual
  union all select 2       , 'oval'            , 'green'      , 'solid'           , 1     from dual
```

```
with symbols  as
  (          select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual
  )
```

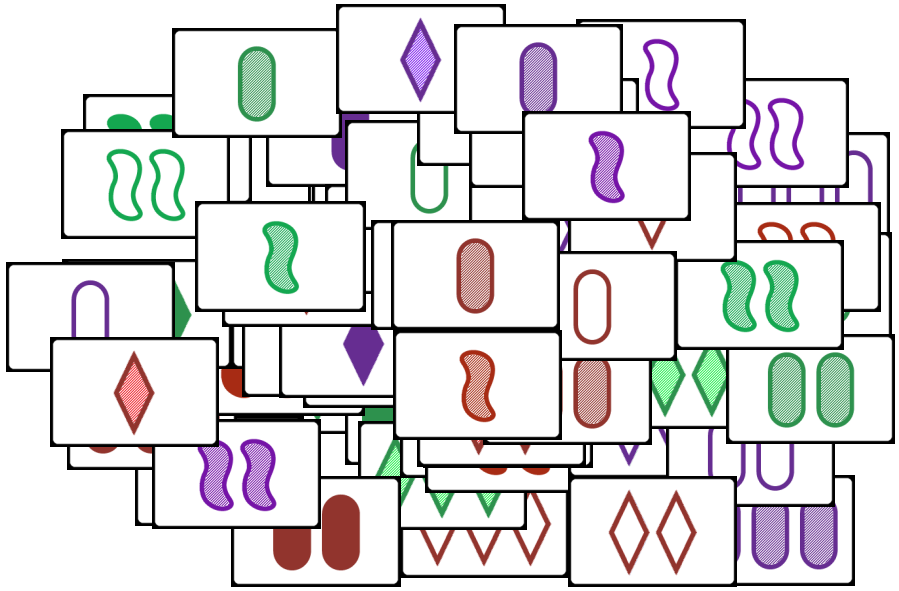
```
with symbols  as
  (          select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual
  )
,    colors   as
 (           select 'green'    color    from dual
   union all select 'purple'            from dual
   union all select 'red'               from dual
  )
```

```
with symbols  as
  (          select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual
  )
,    colors   as
 (           select 'green'    color    from dual
   union all select 'purple'            from dual
   union all select 'red'               from dual
  )
,    shadings as
 (           select 'filled'   shading  from dual
   union all select 'striped'           from dual
   union all select 'open'              from dual
  )
```

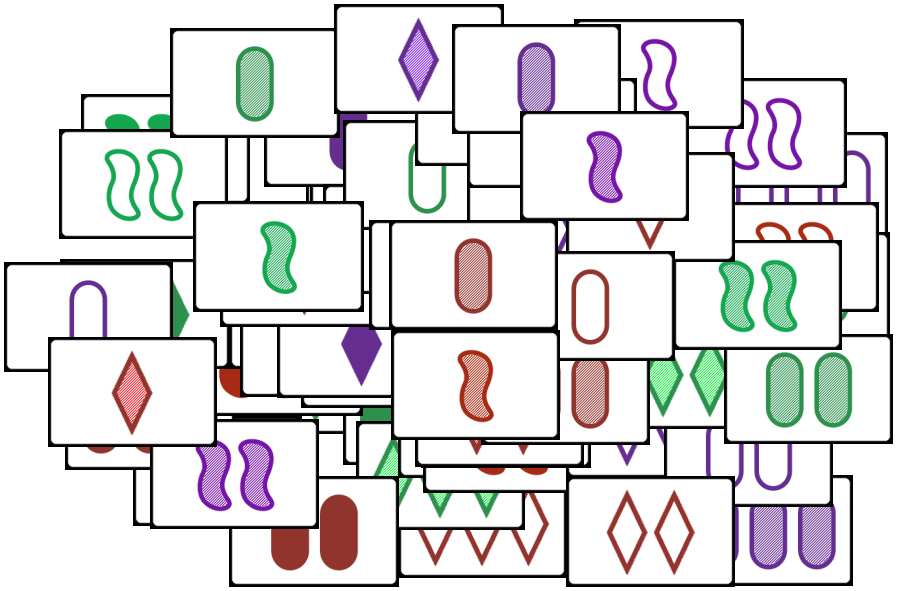
```
with symbols  as
  (          select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual
  )
,    colors   as
 (           select 'green'    color    from dual
   union all select 'purple'            from dual
   union all select 'red'               from dual
  )
,    shadings as
 (           select 'filled'   shading  from dual
   union all select 'striped'           from dual
   union all select 'open'              from dual
  )
,    numbers  as
 (           select 1          quantity from dual
   union all select 2                   from dual
   union all select 3                   from dual
  )
```

```
with symbols  as
  (          select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual
  )
,    colors   as
 (           select 'green'    color    from dual
   union all select 'purple'            from dual
   union all select 'red'               from dual
  )
,    shadings as
 (           select 'filled'   shading  from dual
   union all select 'striped'           from dual
   union all select 'open'              from dual
  )
,    numbers  as
 (           select 1          quantity from dual
   union all select 2                   from dual
   union all select 3                   from dual
  )
,    deck     as
  (select symbol, color, shading, quantity
   from       symbols
   cross join colors
   cross join shadings
   cross join numbers
  )
```

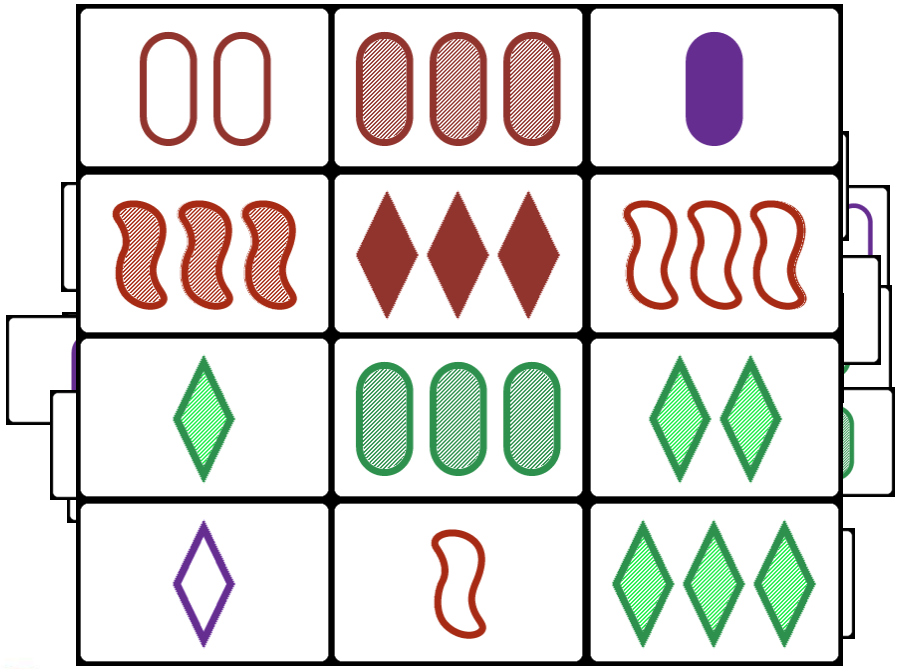
```
with symbols  as
  (          select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual
  )
,    colors   as
 (           select 'green'    color    from dual
   union all select 'purple'            from dual
   union all select 'red'               from dual
  )
,    shadings as
 (           select 'filled'   shading  from dual
   union all select 'striped'           from dual
   union all select 'open'              from dual
  )
,    numbers  as
 (           select 1          quantity from dual
   union all select 2                   from dual
   union all select 3                   from dual
  )
,    deck     as
  (select symbol, color, shading, quantity
   from       symbols
   cross join colors
   cross join shadings
   cross join numbers
  )
select *
from   deck
```



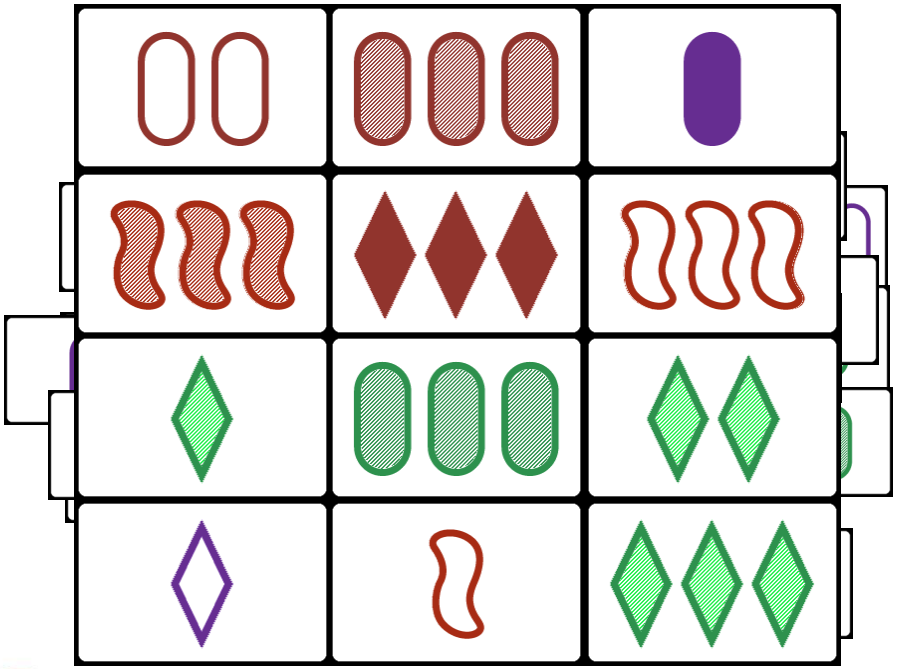
```
with symbols  as
  (          select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual
  )
,    colors   as
 (           select 'green'    color    from dual
   union all select 'purple'            from dual
   union all select 'red'               from dual
  )
,    shadings as
 (           select 'filled'   shading  from dual
   union all select 'striped'           from dual
   union all select 'open'              from dual
  )
,    numbers  as
 (           select 1          quantity from dual
   union all select 2                   from dual
   union all select 3                   from dual
  )
,    deck     as
  (select symbol, color, shading, quantity
   from       symbols
   cross join colors
   cross join shadings
   cross join numbers
  )
select *
from   deck
```



```oracle
with symbols  as
  (          select 'oval'     symbol   from dual
   union all select 'squiggle'         from dual
   union all select 'diamond'          from dual
  )
,    colors   as
 (           select 'green'    color    from dual
   union all select 'purple'           from dual
   union all select 'red'              from dual
  )
,    shadings as
 (           select 'filled'   shading  from dual
   union all select 'striped'          from dual
   union all select 'open'             from dual
  )
,    numbers  as
 (           select 1          quantity from dual
   union all select 2                   from dual
   union all select 3                   from dual
  )
,    deck     as
  (select symbol, color, shading, quantity
   from       symbols
   cross join colors
   cross join shadings
   cross join numbers
  )
select *
from   (select *
        from   deck
        order  by dbms_random.value
       )
where  rownum <= 12
```



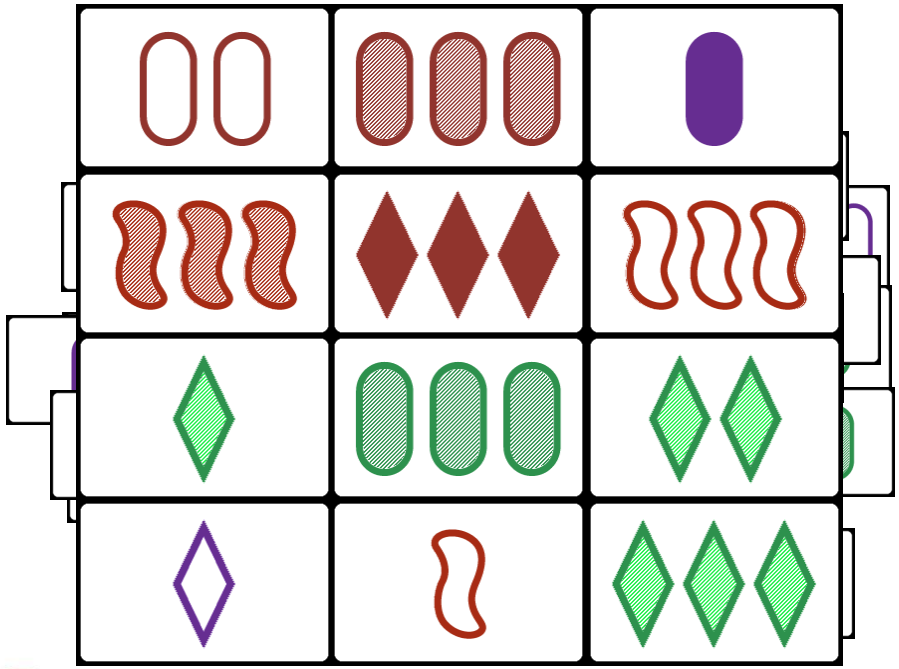
```oracle
with symbols  as
  (          select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual
  )
,    colors   as
 (           select 'green'    color    from dual
   union all select 'purple'            from dual
   union all select 'red'               from dual
  )
,    shadings as
 (           select 'filled'   shading  from dual
   union all select 'striped'           from dual
   union all select 'open'              from dual
  )
,    numbers  as
 (           select 1          quantity from dual
   union all select 2                   from dual
   union all select 3                   from dual
  )
,    deck     as
  (select symbol, color, shading, quantity
   from       symbols
   cross join colors
   cross join shadings
   cross join numbers
  )
select *
from   (select *
        from   deck
        order  by dbms_random.value
       )
where  rownum <= 12
```



```
with symbols  as
  (           select 'oval'     symbol    from dual
   union all select 'squiggle'           from dual
   union all select 'diamond'            from dual
  )
,    colors   as
 (            select 'green'    color     from dual
   union all select 'purple'             from dual
   union all select 'red'                from dual
  )
,    shadings as
 (            select 'filled'   shading   from dual
   union all select 'striped'            from dual
   union all select 'open'               from dual
  )
,    numbers  as
 (            select 1          quantity from dual
   union all select 2                    from dual
   union all select 3                    from dual
  )
,    deck     as
  (select symbol, color, shading, quantity
   from       symbols
   cross join colors
   cross join shadings
   cross join numbers
  )
,    puzzle   as
  (select rownum as cardno, symbol, color
        , shading          , quantity
   from   (select *
           from   deck
           order  by dbms_random.value
          )
```

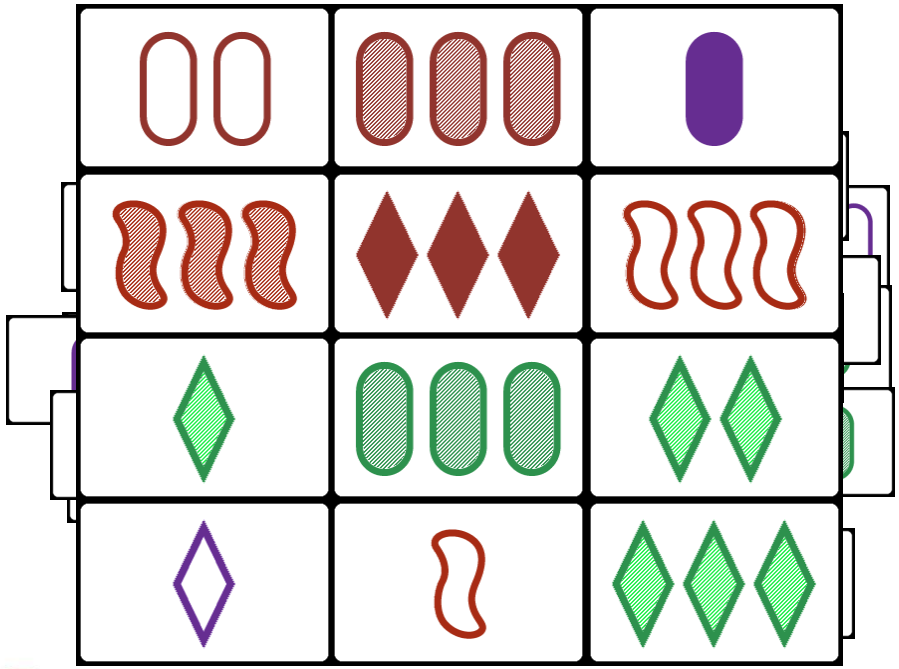

```
with symbols  as
  (           select 'oval'     symbol    from dual
   union all select 'squiggle'           from dual
   union all select 'diamond'            from dual
  )
,    colors   as
 (            select 'green'    color     from dual
   union all select 'purple'             from dual
   union all select 'red'                from dual
  )
,    shadings as
 (            select 'filled'   shading   from dual
   union all select 'striped'            from dual
   union all select 'open'               from dual
  )
,    numbers  as
 (            select 1          quantity from dual
   union all select 2                    from dual
   union all select 3                    from dual
  )
,    deck     as
  (select symbol, color, shading, quantity
   from       symbols
   cross join colors
   cross join shadings
   cross join numbers
  )
,    puzzle   as
  (select rownum as cardno, symbol, color
        , shading          , quantity
   from   (select *
           from   deck
           order  by dbms_random.value
          )
```


```
  )
,    solution as
  (select cards1.symbol   as symbol1
        , cards1.color    as color1
        , cards1.shading  as shading1
        , cards1.quantity as quantity1
        , cards2.symbol   as symbol2
        , cards2.color    as color2
        , cards2.shading  as shading2
        , cards2.quantity as quantity2
        , cards3.symbol   as symbol3
        , cards3.color    as color3
        , cards3.shading  as shading3
        , cards3.quantity as quantity3
   from   puzzle cards1
   join   puzzle cards2
     on   cards1.cardno < cards2.cardno
   join   puzzle cards3
     on   cards2.cardno < cards3.cardno
   where  1=1
   and    (   (    cards1.symbol  =  cards2.symbol
              and cards2.symbol  =  cards3.symbol
              )
           or (    cards1.symbol  <> cards2.symbol
              and cards2.symbol  <> cards3.symbol
              and cards3.symbol  <> cards1.symbol
              )
           )
   and    (   (    cards1.color  =  cards2.color
              and cards2.color  =  cards3.color
              )
           or (    cards1.color  <> cards2.color
              and cards2.color  <> cards3.color
              and cards3.color  <> cards1.color
              )
           )
   and    (   (    cards1.shading  =  cards2.shading
              and cards2.shading  =  cards3.shading
              )
           or (    cards1.shading  <> cards2.shading
              and cards2.shading  <> cards3.shading
              and cards3.shading  <> cards1.shading
              )
           )
   and    (   (    cards1.quantity  =  cards2.quantity
              and cards2.quantity  =  cards3.quantity
              )
           or (    cards1.quantity  <> cards2.quantity
              and cards2.quantity  <> cards3.quantity
              and cards3.quantity  <> cards1.quantity
              )
           )
  )
select 'puzzle'   as what
     .                symbol,          color
     .                shading,         quantity
     . null       as symbol,  null as color
     . null       as shading, null as quantity
     . null       as symbol,  null as color
     . null       as shading, null as quantity
from   puzzle
union  all
select 'solution' as what
     .                symbol1,         color1
     .                shading1,        quantity1
     .                symbol2,         color2
     .                shading2,        quantity2
     .                symbol3,         color3
     .                shading3,        quantity3
from   solution
```



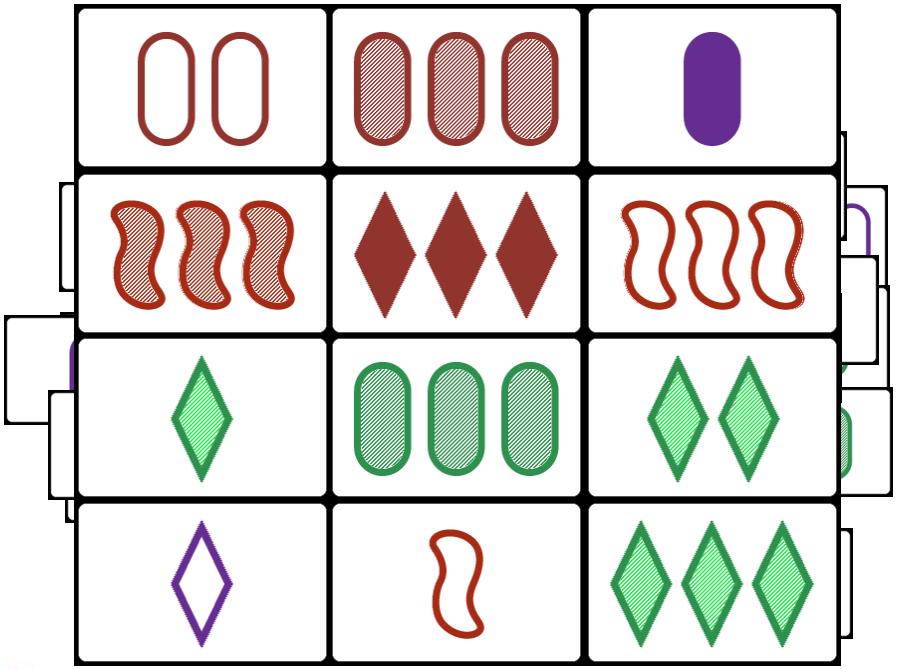
```
  )
,    puzzle   as
  (select rownum as cardno, symbol, color
        , shading          , quantity
   from   (select *
           from   deck
           order  by dbms_random.value
          )
   where  rownum <= 12
  )
,    solution as
  (select cards1.symbol   as symbol1
        , cards1.color    as color1
        , cards1.shading  as shading1
        , cards1.quantity as quantity1
        , cards2.symbol   as symbol2
        , cards2.color    as color2
        , cards2.shading  as shading2
        , cards2.quantity as quantity2
        , cards3.symbol   as symbol3
        , cards3.color    as color3
        , cards3.shading  as shading3
        , cards3.quantity as quantity3
   from   puzzle cards1
   join   puzzle cards2
     on   cards1.cardno < cards2.cardno
   join   puzzle cards3
     on   cards2.cardno < cards3.cardno
   where  1=1
   and    (   (    cards1.symbol   =  cards2.symbol
                 and cards2.symbol   =  cards3.symbol
               )
            or (    cards1.symbol   <> cards2.symbol
                 and cards2.symbol   <> cards3.symbol
                 and cards3.symbol   <> cards1.symbol
               )
```

```
with symbols  as
  (         select 'oval'     symbol   from dual
  union all select 'squiggle'          from dual
  union all select 'diamond'           from dual
  )
,   colors   as
  (         select 'green'    color    from dual
  union all select 'purple'            from dual
  union all select 'red'               from dual
  )
,   shadings as
  (         select 'filled'   shading  from dual
  union all select 'striped'           from dual
  union all select 'open'              from dual
  )
,   numbers  as
  (         select 1          quantity from dual
  union all select 2                   from dual
  union all select 3                   from dual
  )
,   deck     as
  (select symbol, color, shading, quantity
  from    symbols
  cross join colors
  cross join shadings
  cross join numbers
  )
,   puzzle   as
  (select rownum as cardno, symbol, color
        , shading          , quantity
   from   (select *
           from   deck
           order  by dbms_random.value
          )
   where  rownum <= 12
  )
,   solution as
  (select cards1.symbol   as symbol1
        , cards1.color    as color1
        , cards1.shading  as shading1
        , cards1.quantity as quantity1
        , cards2.symbol   as symbol2
        , cards2.color    as color2
        , cards2.shading  as shading2
        , cards2.quantity as quantity2
        , cards3.symbol   as symbol3
        , cards3.color    as color3
        , cards3.shading  as shading3
        , cards3.quantity as quantity3
   from   puzzle cards1
   join   puzzle cards2
     on   cards1.cardno < cards2.cardno
   join   puzzle cards3
     on   cards2.cardno < cards3.cardno
   where  1=1
   and    (   (    cards1.symbol   =  cards2.symbol
                 and cards2.symbol   =  cards3.symbol
               )
            or (    cards1.symbol   <> cards2.symbol
                 and cards2.symbol   <> cards3.symbol
                 and cards3.symbol   <> cards1.symbol
               )
   and    (   (    cards1.color    =  cards2.color
                 and cards2.color    =  cards3.color
               )
            or (    cards1.color    <> cards2.color
                 and cards2.color    <> cards3.color
                 and cards3.color    <> cards1.color
               )
   and    (   (    cards1.shading  =  cards2.shading
                 and cards2.shading  =  cards3.shading
               )
            or (    cards1.shading  <> cards2.shading
                 and cards2.shading  <> cards3.shading
                 and cards3.shading  <> cards1.shading
               )
   and    (   (    cards1.quantity =  cards2.quantity
                 and cards2.quantity =  cards3.quantity
               )
            or (    cards1.quantity <> cards2.quantity
                 and cards2.quantity <> cards3.quantity
                 and cards3.quantity <> cards1.quantity
               )
   and    (   (    cards1.color    =  cards2.color
                 and cards2.color    =  cards3.color
               )
            or (    cards1.color    <> cards2.color
                 and cards2.color    <> cards3.color
                 and cards3.color    <> cards1.color
               )
   and    (   (    cards1.shading  =  cards2.shading
                 and cards2.shading  =  cards3.shading
               )
            or (    cards1.shading  <> cards2.shading
                 and cards2.shading  <> cards3.shading
                 and cards3.shading  <> cards1.shading
               )
   and    (   (    cards1.quantity =  cards2.quantity
                 and cards2.quantity =  cards3.quantity
               )
            or (    cards1.quantity <> cards2.quantity
                 and cards2.quantity <> cards3.quantity
                 and cards3.quantity <> cards1.quantity
               )
   and    (   (    cards1.quantity =  cards2.quantity
                 and cards2.quantity =  cards3.quantity
               )
            or (    cards1.quantity <> cards2.quantity
                 and cards2.quantity <> cards3.quantity
                 and cards3.quantity <> cards1.quantity
               )
  )
select 'puzzle'   as what
     .              symbol,          color
     .              shading,         quantity
     . null       as symbol,  null as color
     . null       as shading, null as quantity
     . null       as symbol,  null as color
     . null       as shading, null as quantity
from   puzzle
union  all
select 'solution' as what
     .              symbol1,         color1
     .              shading1,        quantity1
     .              symbol2,         color2
     .              shading2,        quantity2
     .              symbol3,         color3
     .              shading3,        quantity3
from   solution
```



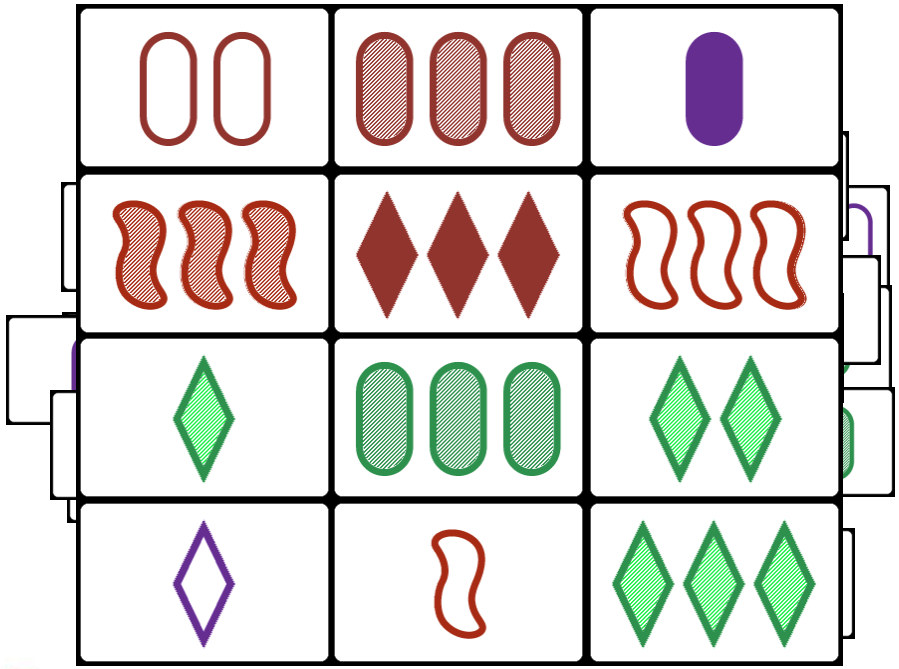
```
  on   cards2.cardno < cards3.cardno
where  1=1
and    (   (    cards1.symbol   =  cards2.symbol
                and cards2.symbol   =  cards3.symbol
            )
         or (    cards1.symbol  <> cards2.symbol
                and cards2.symbol  <> cards3.symbol
                and cards3.symbol  <> cards1.symbol
            )
       )
and    (   (    cards1.color   =  cards2.color
                and cards2.color   =  cards3.color
            )
         or (    cards1.color  <> cards2.color
                and cards2.color  <> cards3.color
                and cards3.color  <> cards1.color
            )
       )
and    (   (    cards1.shading   =  cards2.shading
                and cards2.shading   =  cards3.shading
            )
         or (    cards1.shading  <> cards2.shading
                and cards2.shading  <> cards3.shading
                and cards3.shading  <> cards1.shading
            )
       )
and    (   (    cards1.quantity   =  cards2.quantity
                and cards2.quantity   =  cards3.quantity
            )
         or (    cards1.quantity  <> cards2.quantity
                and cards2.quantity  <> cards3.quantity
                and cards3.quantity  <> cards1.quantity
            )
       )
  )
select 'puzzle'   as what
```

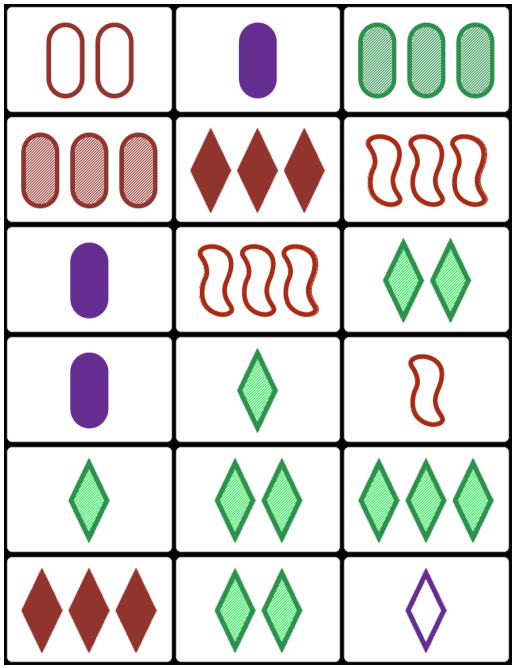
```
with symbols  as
  (         select 'oval'     symbol   from dual
  union all select 'squiggle'          from dual
  union all select 'diamond'           from dual
  )
,   colors   as
  (         select 'green'    color    from dual
  union all select 'purple'            from dual
  union all select 'red'               from dual
  )
,   shadings as
  (         select 'filled'   shading  from dual
  union all select 'striped'           from dual
  union all select 'open'              from dual
  )
,   numbers  as
  (         select 1          quantity from dual
  union all select 2                   from dual
  union all select 3                   from dual
  )
,   deck     as
  (select symbol, color, shading, quantity
   from     symbols
   cross join colors
   cross join shadings
   cross join numbers
  )
,   puzzle   as
  (select rownum as cardno, symbol, color
        , shading         , quantity
   from   (select *
           from   deck
           order  by dbms_random.value
          )
   where  rownum <= 12
  )
,   solution as
  (select cards1.symbol   as symbol1
        , cards1.color    as color1
        , cards1.shading  as shading1
        , cards1.quantity as quantity1
        , cards2.symbol   as symbol2
        , cards2.color    as color2
        , cards2.shading  as shading2
        , cards2.quantity as quantity2
        , cards3.symbol   as symbol3
        , cards3.color    as color3
        , cards3.shading  as shading3
        , cards3.quantity as quantity3
   from   puzzle cards1
   join   puzzle cards2
     on   cards1.cardno < cards2.cardno
   join   puzzle cards3
     on   cards2.cardno < cards3.cardno
   where  1=1
   and    (   (    cards1.symbol   =  cards2.symbol
                  and cards2.symbol   =  cards3.symbol
              )
           or (    cards1.symbol  <> cards2.symbol
                  and cards2.symbol  <> cards3.symbol
                  and cards3.symbol  <> cards1.symbol
              )
          )
   and    (   (    cards1.color   =  cards2.color
                  and cards2.color   =  cards3.color
              )
           or (    cards1.color  <> cards2.color
                  and cards2.color  <> cards3.color
                  and cards3.color  <> cards1.color
              )
          )
   and    (   (    cards1.shading   =  cards2.shading
                  and cards2.shading   =  cards3.shading
              )
           or (    cards1.shading  <> cards2.shading
                  and cards2.shading  <> cards3.shading
                  and cards3.shading  <> cards1.shading
              )
          )
   and    (   (    cards1.quantity   =  cards2.quantity
                  and cards2.quantity   =  cards3.quantity
              )
           or (    cards1.quantity  <> cards2.quantity
                  and cards2.quantity  <> cards3.quantity
                  and cards3.quantity  <> cards1.quantity
              )
          )
  )
select 'puzzle'   as what
     ,                symbol,          color
     ,                shading,         quantity
     , null        as symbol,  null as color
     , null        as shading, null as quantity
     , null        as symbol,  null as color
     , null        as shading, null as quantity
from   puzzle
union  all
select 'solution' as what
     ,                symbol1,         color1
     ,                shading1,        quantity1
     ,                symbol2,         color2
     ,                shading2,        quantity2
     ,                symbol3,         color3
     ,                shading3,        quantity3
from   solution
```

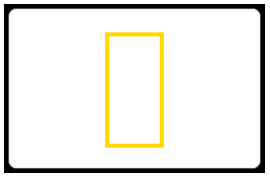


```
  )
select 'puzzle'    as what
     ,                  symbol,           color
     ,                  shading,          quantity
     , null        as symbol,   null as color
     , null        as shading,  null as quantity
     , null        as symbol,   null as color
     , null        as shading,  null as quantity
from   puzzle
union  all
select 'solution' as what
     ,                  symbol1,          color1
     ,                  shading1,         quantity1
     ,                  symbol2,          color2
     ,                  shading2,         quantity2
     ,                  symbol3,          color3
     ,                  shading3,         quantity3
from   solution
```

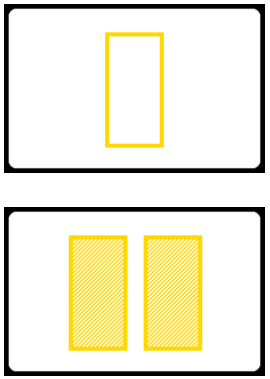
```
with symbols  as
  (         select 'oval'     symbol   from dual
  union all select 'squiggle'          from dual
  union all select 'diamond'           from dual
  )
,   colors   as
  (         select 'green'    color    from dual
  union all select 'purple'            from dual
  union all select 'red'               from dual
  )
,   shadings as
  (         select 'filled'   shading  from dual
  union all select 'striped'           from dual
  union all select 'open'              from dual
  )
,   numbers  as
  (         select 1          quantity from dual
  union all select 2                   from dual
  union all select 3                   from dual
  )
,   deck     as
  (select symbol, color, shading, quantity
  from    symbols
  cross join colors
  cross join shadings
  cross join numbers
  )
,   puzzle   as
  (select rownum as cardno, symbol, color
        , shading        , quantity
  from   (select *
          from   deck
          order  by dbms_random.value
          )
  where  rownum <= 12
  )
,   solution as
  (select cards1.symbol   as symbol1
        , cards1.color    as color1
        , cards1.shading  as shading1
        , cards1.quantity as quantity1
        , cards2.symbol   as symbol2
        , cards2.color    as color2
        , cards2.shading  as shading2
        , cards2.quantity as quantity2
        , cards3.symbol   as symbol3
        , cards3.color    as color3
        , cards3.shading  as shading3
        , cards3.quantity as quantity3
  from   puzzle cards1
  join   puzzle cards2
    on   cards1.cardno < cards2.cardno
  join   puzzle cards3
    on   cards2.cardno < cards3.cardno
  where  1=1
  and    (   (    cards1.symbol  =  cards2.symbol
             and cards2.symbol  =  cards3.symbol
             )
          or (    cards1.symbol  <> cards2.symbol
             and cards2.symbol  <> cards3.symbol
             and cards3.symbol  <> cards1.symbol
             )
          )
  and    (   (    cards1.color  =  cards2.color
             and cards2.color  =  cards3.color
             )
          or (    cards1.color  <> cards2.color
             and cards2.color  <> cards3.color
             and cards3.color  <> cards1.color
             )
          )
  and    (   (    cards1.shading  =  cards2.shading
             and cards2.shading  =  cards3.shading
             )
          or (    cards1.shading  <> cards2.shading
             and cards2.shading  <> cards3.shading
             and cards3.shading  <> cards1.shading
             )
          )
  and    (   (    cards1.quantity  =  cards2.quantity
             and cards2.quantity  =  cards3.quantity
             )
          or (    cards1.quantity  <> cards2.quantity
             and cards2.quantity  <> cards3.quantity
             and cards3.quantity  <> cards1.quantity
             )
          )
  )
select 'puzzle'   as what
     ,                  symbol,           color
     ,                  shading,          quantity
     , null        as symbol,   null as color
     , null        as shading,  null as quantity
     , null        as symbol,   null as color
     , null        as shading,  null as quantity
from   puzzle
union  all
select 'solution' as what
     ,                  symbol1,          color1
     ,                  shading1,         quantity1
     ,                  symbol2,          color2
     ,                  shading2,         quantity2
     ,                  symbol3,          color3
     ,                  shading3,         quantity3
from   solution
```



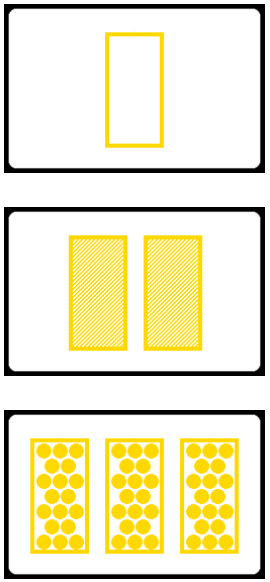




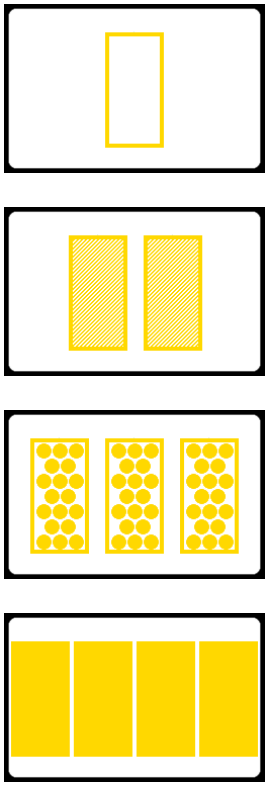
```
with symbols as
  (          select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual
   union all select 'square'            from dual -- fourth symbol
  )
```



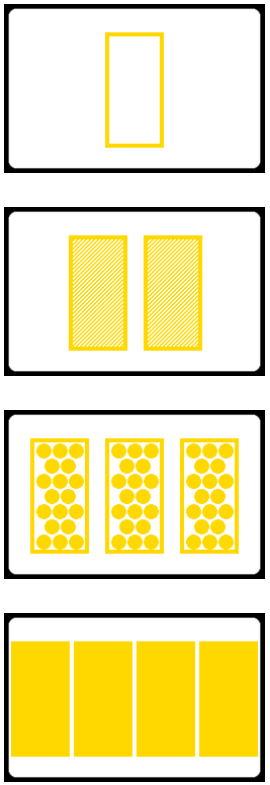
```
with symbols as
  (          select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual
   union all select 'square'            from dual -- fourth symbol
  )
,    colors   as
 (           select 'green'    color    from dual
   union all select 'purple'            from dual
   union all select 'red'               from dual
   union all select 'yellow'            from dual -- fourth color
  )
```



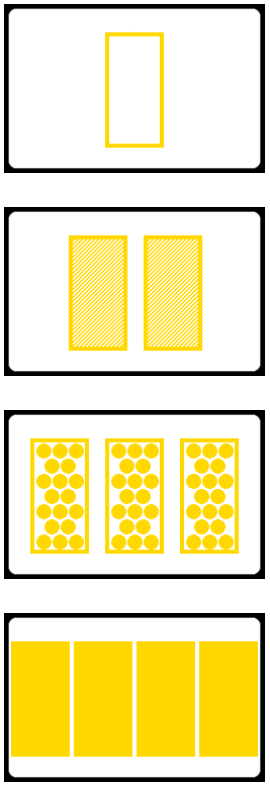
```
with symbols as
  (          select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual
   union all select 'square'            from dual -- fourth symbol
  )
,    colors   as
 (           select 'green'    color    from dual
   union all select 'purple'            from dual
   union all select 'red'               from dual
   union all select 'yellow'            from dual -- fourth color
  )
,    shadings as
 (           select 'filled'   shading  from dual
   union all select 'striped'           from dual
   union all select 'open'              from dual
   union all select 'dotted'            from dual -- fourth shading
  )
```



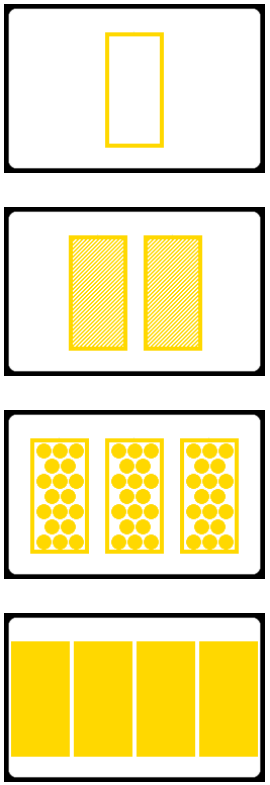
```
with symbols as
  (          select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual
   union all select 'square'            from dual -- fourth symbol
  )
,    colors   as
 (           select 'green'    color    from dual
   union all select 'purple'            from dual
   union all select 'red'               from dual
   union all select 'yellow'            from dual -- fourth color
  )
,    shadings as
 (           select 'filled'   shading  from dual
   union all select 'striped'           from dual
   union all select 'open'              from dual
   union all select 'dotted'            from dual -- fourth shading
  )
,    numbers  as
 (           select 1          quantity from dual
   union all select 2                   from dual
   union all select 3                   from dual
   union all select 4                   from dual -- fourth quantity
  )
```



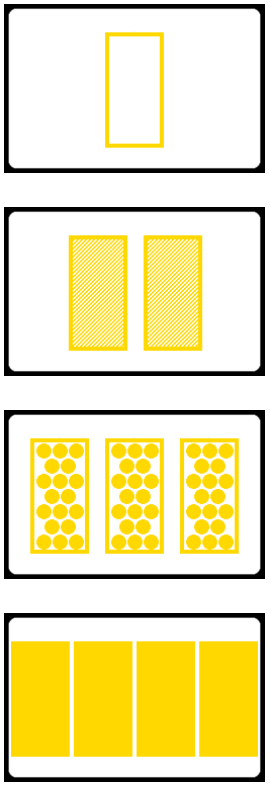
```
with symbols as
  (          select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual
   union all select 'square'            from dual -- fourth symbol
  )
,    colors   as
 (          select 'green'    color    from dual
   union all select 'purple'            from dual
   union all select 'red'               from dual
   union all select 'yellow'            from dual -- fourth color
  )
,    shadings as
 (          select 'filled'   shading  from dual
   union all select 'striped'           from dual
   union all select 'open'              from dual
   union all select 'dotted'            from dual -- fourth shading
  )
,    numbers  as
 (          select 1          quantity from dual
   union all select 2                   from dual
   union all select 3                   from dual
   union all select 4                   from dual -- fourth quantity
  )
,    deck     as
  (select symbol, color, shading, quantity
   from       symbols
   cross join colors
   cross join shadings
   cross join numbers
  )
...
```



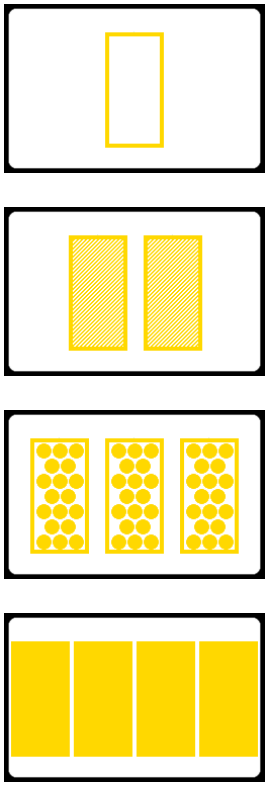
```
    (    cards1.feature =  cards2.feature
     and cards2.feature =  cards3.feature
     and cards3.feature =  cards4.feature -- fourth card
    )
or
    (    cards1.feature <> cards2.feature
     and cards1.feature <> cards3.feature
     and cards1.feature <> cards4.feature -- fourth card
     and cards2.feature <> cards3.feature
     and cards2.feature <> cards4.feature -- fourth card
     and cards3.feature <> cards4.feature -- fourth card
    )
```



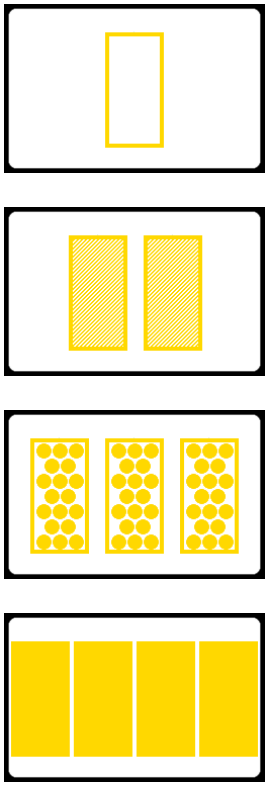
```
with symbols as
  (          select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual
   union all select 'square'            from dual -- fourth symbol
  )
,    colors   as
 (          select 'green'    color    from dual
   union all select 'purple'            from dual
   union all select 'red'               from dual
   union all select 'yellow'            from dual -- fourth color
  )
,    shadings as
 (          select 'filled'   shading  from dual
   union all select 'striped'           from dual
   union all select 'open'              from dual
   union all select 'dotted'            from dual -- fourth shading
  )
,    numbers  as
 (          select 1          quantity from dual
   union all select 2                   from dual
   union all select 3                   from dual
   union all select 4                   from dual -- fourth quantity
  )
,    deck     as
  (select symbol, color, shading, quantity
   from       symbols
   cross join colors
   cross join shadings
   cross join numbers
  )
,    puzzle   as
  (select rownum as cardno, symbol, color
        , shading         , quantity
```



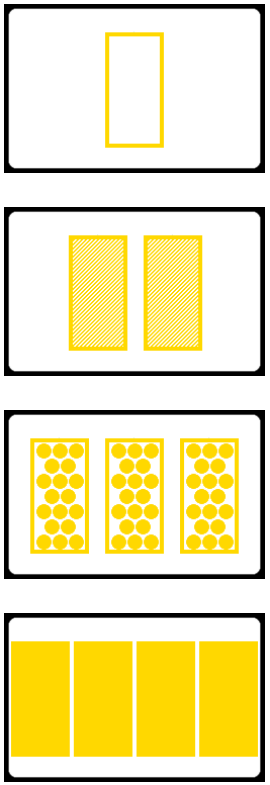
```
  ,
,    puzzle   as
  (select rownum as cardno, symbol, color
        , shading         , quantity
   from   (select *
           from   deck
           order  by dbms_random.value
          )
   -- increase this number to increase the number of sets present
   where  rownum <= 20
  )
,    solution as
  (select cards1.symbol   as symbol1
        , cards1.color    as color1
        , cards1.shading  as shading1
        , cards1.quantity as quantity1
        , cards2.symbol   as symbol2
        , cards2.color    as color2
        , cards2.shading  as shading2
        , cards2.quantity as quantity2
        , cards3.symbol   as symbol3
        , cards3.color    as color3
        , cards3.shading  as shading3
        , cards3.quantity as quantity3
        , cards4.symbol   as symbol4
        , cards4.color    as color4
        , cards4.shading  as shading4
        , cards4.quantity as quantity4
   from   puzzle cards1
   join   puzzle cards2
     on   cards1.cardno < cards2.cardno
   join   puzzle cards3
     on   cards2.cardno < cards3.cardno
   join   puzzle cards4
     on   cards3.cardno < cards4.cardno
   where  1=1
```



```
  on    cards1.cardno < cards2.cardno
join    puzzle cards3
  on    cards2.cardno < cards3.cardno
join    puzzle cards4
  on    cards3.cardno < cards4.cardno
where   1=1
and     (   (    cards1.symbol   =  cards2.symbol
             and cards2.symbol   =  cards3.symbol
             and cards3.symbol   =  cards4.symbol
            )
         or (    cards1.symbol   <> cards2.symbol
             and cards1.symbol   <> cards3.symbol
             and cards1.symbol   <> cards4.symbol
             and cards2.symbol   <> cards3.symbol
             and cards2.symbol   <> cards4.symbol
             and cards3.symbol   <> cards4.symbol
            )
        )
and     (   (    cards1.color    =  cards2.color
             and cards2.color    =  cards3.color
             and cards3.color    =  cards4.color
            )
         or (    cards1.color    <> cards2.color
             and cards1.color    <> cards3.color
             and cards1.color    <> cards4.color
             and cards2.color    <> cards3.color
             and cards2.color    <> cards4.color
             and cards3.color    <> cards4.color
            )
        )
and     (   (    cards1.shading  =  cards2.shading
             and cards2.shading  =  cards3.shading
             and cards3.shading  =  cards4.shading
            )
         or (    cards1.shading  <> cards2.shading
```



```
                    and cards1.color    <> cards4.color
                    and cards2.color    <> cards3.color
                    and cards2.color    <> cards4.color
                    and cards3.color    <> cards4.color
                  )
            )
    and     (   (   cards1.shading  =  cards2.shading
                    and cards2.shading  =  cards3.shading
                    and cards3.shading  =  cards4.shading
                  )
                or (   cards1.shading  <> cards2.shading
                    and cards1.shading  <> cards3.shading
                    and cards1.shading  <> cards4.shading
                    and cards2.shading  <> cards3.shading
                    and cards2.shading  <> cards4.shading
                    and cards3.shading  <> cards4.shading
                  )
            )
    and     (   (   cards1.quantity =  cards2.quantity
                    and cards2.quantity =  cards3.quantity
                    and cards3.quantity =  cards4.quantity
                  )
                or (   cards1.quantity <> cards2.quantity
                    and cards1.quantity <> cards3.quantity
                    and cards1.quantity <> cards4.quantity
                    and cards2.quantity <> cards3.quantity
                    and cards2.quantity <> cards4.quantity
                    and cards3.quantity <> cards4.quantity
                  )
            )
  )
select 'puzzle'       as what
     , puzzle.symbol  as symbol1,  puzzle.color    as color1
     , puzzle.shading as shading1, puzzle.quantity as quantity1
     , null           as symbol2,  null            as color2
```



```
                )
            )
  )
select 'puzzle'        as what
     , puzzle.symbol   as symbol1,  puzzle.color    as color1
     , puzzle.shading  as shading1, puzzle.quantity as quantity1
     , null            as symbol2,  null            as color2
     , null            as shading2, null            as quantity2
     , null            as symbol3,  null            as color3
     , null            as shading3, null            as quantity3
     , null            as symbol4,  null            as color4
     , null            as shading4, null            as quantity4
from   puzzle
union  all
select 'solution'      as what
     ,                    symbol1,                     color1
     ,                    shading1,                    quantity1
     ,                    symbol2,                     color2
     ,                    shading2,                    quantity2
     ,                    symbol3,                     color3
     ,                    shading3,                    quantity3
     ,                    symbol4,                     color4
     ,                    shading4,                    quantity4
from   solution
```

```
    (    cards1.feature =  cards2.feature
     and cards1.feature =  cards3.feature
    )
or  (    cards1.feature <> cards2.feature
     and cards2.feature <> cards3.feature
     and cards1.feature <> cards3.feature
    )
```

```
and (   (    cards1.texture =  cards2.texture
         and cards2.texture =  cards3.texture
        )
     or (    cards1.texture <> cards2.texture
         and cards2.texture <> cards3.texture
         and cards3.texture <> cards1.texture
        )
    )
and (   (    cards1.symbol  =  cards2.symbol
         and cards2.symbol  =  cards3.symbol
        )
     or (    cards1.symbol  <> cards2.symbol
         and cards2.symbol  <> cards3.symbol
         and cards3.symbol  <> cards1.symbol
        )
    )
```

```
and (   (    cards1.shading =  cards2.shading
         and cards2.shading =  cards3.shading
        )
     or (    cards1.shading <> cards2.shading
         and cards2.shading <> cards3.shading
         and cards3.shading <> cards1.shading
        )
    )
and (   (    cards1.color   =  cards2.color
         and cards2.color   =  cards3.color
        )
     or (    cards1.color   <> cards2.color
         and cards2.color   <> cards3.color
         and cards3.color   <> cards1.color
        )
    )
```

```
and (   (    cards1.qty      =  cards2.qty
         and cards2.qty      =  cards3.qty
        )
     or (    cards1.qty      <> cards2.qty
         and cards2.qty      <> cards3.qty
         and cards3.qty      <> cards1.qty
        )
    )
and (   (    cards1.symbol   =  cards2.symbol
         and cards2.symbol   =  cards3.symbol
        )
     or (    cards1.symbol   <> cards2.symbol
         and cards2.symbol   <> cards3.symbol
         and cards3.symbol   <> cards1.symbol
        )
    )
```

```
and (   (    cards1.shading =  cards2.shading
         and cards2.shading =  cards3.shading
        )
     or (    cards1.shading <> cards2.shading
         and cards2.shading <> cards3.shading
         and cards3.shading <> cards1.shading
        )
    )
and (   (    cards1.color   =  cards2.color
         and cards2.color   =  cards3.color
        )
     or (    cards1.color   <> cards2.color
         and cards2.color   <> cards3.color
         and cards3.color   <> cards1.color
        )
    )
```

```
and (   (    cards1.qty      =  cards2.qty
         and cards2.qty      =  cards3.qty
        )
     or (    cards1.qty      <> cards2.qty
         and cards2.qty      <> cards3.qty
         and cards3.qty      <> cards1.qty
        )
    )
and (   (    cards1.symbol   =  cards2.symbol
         and cards2.symbol   =  cards3.symbol
        )
     or (    cards1.symbol   <> cards2.symbol
         and cards2.symbol   <> cards3.symbol
         and cards3.symbol   <> cards1.symbol
        )
    )
```

```
and (   (    cards1.shading =  cards2.shading
         and cards2.shading =  cards3.shading
        )
     or (    cards1.shading <> cards2.shading
         and cards2.shading <> cards3.shading
         and cards3.shading <> cards1.shading
        )
    )
and (   (    cards1.color   =  cards2.color
         and cards2.color   =  cards3.color
        )
     or (    cards1.color   <> cards2.color
         and cards2.color   <> cards3.color
         and cards3.color   <> cards1.color
        )
    )
```

```
    (    cards1.feature =  cards2.feature
     and cards1.feature =  cards3.feature
    )
or  (    cards1.feature <> cards2.feature
     and cards2.feature <> cards3.feature
     and cards1.feature <> cards3.feature
    )
```

```
select
from
where
```

{

Table

```
select
from
where
```

{

Table

```
select
from
where
```

```
sqlmacro( table  )
```

Table

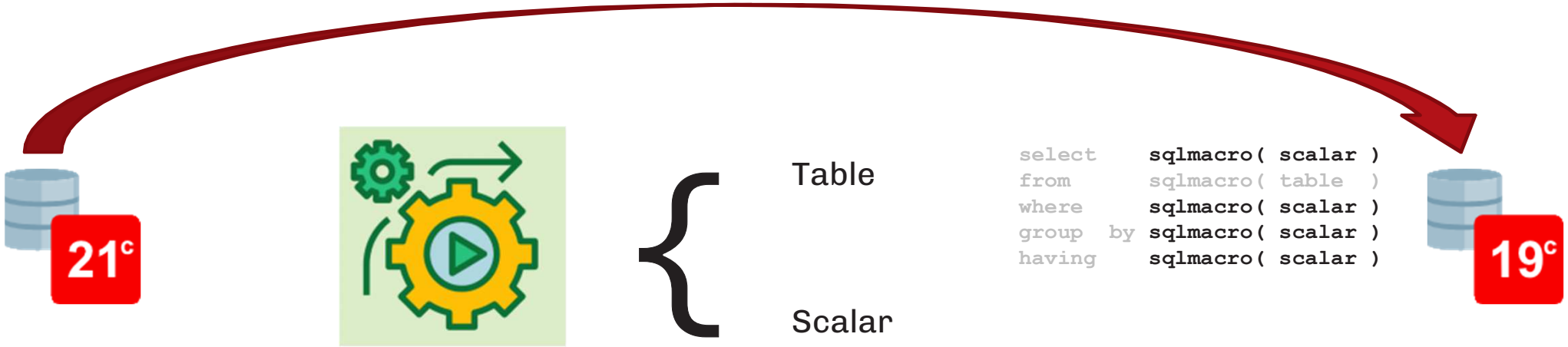
Scalar

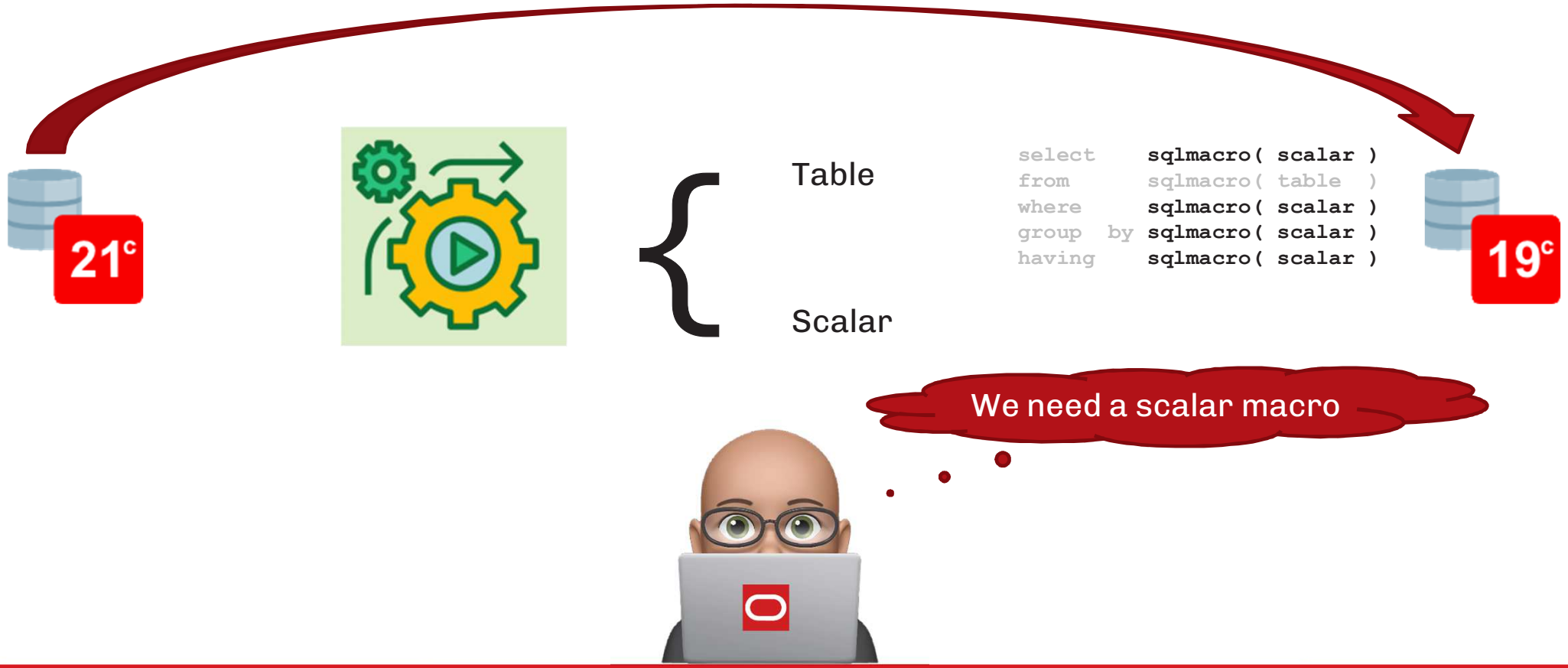
```
select
from        sqlmacro( table  )
where
```

Table

Scalar

```
select    sqlmacro( scalar )
from      sqlmacro( table  )
where     sqlmacro( scalar )
```


21c
Table
Scalar
select sqlmacro(scalar)
from sqlmacro(table)
where sqlmacro(scalar)
group by sqlmacro(scalar)
having sqlmacro(scalar)
19c


21c
Table
Scalar
select sqlmacro(scalar)
from sqlmacro(table)
where sqlmacro(scalar)
group by sqlmacro(scalar)
having sqlmacro(scalar)
19c
We need a scalar macro

```
create or replace function <function_name>
( <parameters> )
return varchar2 sql_macro( scalar )
```

```
create or replace function gameofset_predicate
( <parameters> )
return varchar2 sql_macro( scalar )
```

```
create or replace function gameofset_predicate
( col_in in dbms_tf.columns_t )
return varchar2 sql_macro( scalar )
```

```
create or replace function gameofset_predicate
( col_in in dbms_tf.columns_t )
return varchar2 sql_macro( scalar ) is
  lsql varchar2( 32767 );
begin
  lsql := '

              (     cards1.feature            = cards2.feature
                and cards2.feature            = cards3.feature
              )
           or (     cards1.feature           <> cards2.feature
                and cards2.feature           <> cards3.feature
                and cards3.feature           <> cards1.feature
              )



)';
return lsql;

end;
/
```

```
create or replace function gameofset_predicate
( col_in in dbms_tf.columns_t )
return varchar2 sql_macro( scalar ) is
  lsql varchar2( 32767 );
begin
  lsql := '

              (     c1.feature                = c2.feature
                and c2.feature                = c3.feature
              )
           or (     c1.feature               <> c2.feature
                and c2.feature               <> c3.feature
                and c3.feature               <> c1.feature
              )



)';
return lsql;

end;
/
```

```
create or replace function gameofset_predicate
( col_in in dbms_tf.columns_t )
return varchar2 sql_macro( scalar ) is
  lsql varchar2( 32767 );
begin
  lsql := '

             (     c1.' || col_in( 1 ) || '  = c2.' || col_in( 1 ) || '
               and c2.' || col_in( 1 ) || '  = c3.' || col_in( 1 ) || '
             )
          or (     c1.' || col_in( 1 ) || ' <> c2.' || col_in( 1 ) || '
               and c2.' || col_in( 1 ) || ' <> c3.' || col_in( 1 ) || '
               and c3.' || col_in( 1 ) || ' <> c1.' || col_in( 1 ) || '
             )



)';
return lsql;

end;
/
```

```
create or replace function gameofset_predicate
( col_in in dbms_tf.columns_t )
return varchar2 sql_macro( scalar ) is
  lsql varchar2( 32767 );
begin
  lsql := '
( case
    when (    (     c1.' || col_in( 1 ) || '  = c2.' || col_in( 1 ) || '
                and c2.' || col_in( 1 ) || '  = c3.' || col_in( 1 ) || '
              )
           or (     c1.' || col_in( 1 ) || ' <> c2.' || col_in( 1 ) || '
                and c2.' || col_in( 1 ) || ' <> c3.' || col_in( 1 ) || '
                and c3.' || col_in( 1 ) || ' <> c1.' || col_in( 1 ) || '
              )
         ) then 1
    else 0
  end
)';
return lsql;

end;
/
```

```
where  1=1
and    (   (     cards1.symbol  =  cards2.symbol
             and cards2.symbol  =  cards3.symbol
            )
        or (     cards1.symbol  <> cards2.symbol
             and cards2.symbol  <> cards3.symbol
             and cards3.symbol  <> cards1.symbol
            )
       )
and    (   (     cards1.color  =  cards2.color
             and cards2.color  =  cards3.color
           )
        or (     cards1.color  <> cards2.color
             and cards2.color  <> cards3.color
             and cards3.color  <> cards1.color
            )
        )
and    (   (     cards1.shading  =  cards2.shading
             and cards2.shading  =  cards3.shading
            )
        or (     cards1.shading  <> cards2.shading
             and cards2.shading  <> cards3.shading
             and cards3.shading  <> cards1.shading
            )
        )
and    (   (     cards1.quantity  =  cards2.quantity
             and cards2.quantity  =  cards3.quantity
            )
        or (     cards1.quantity  <> cards2.quantity
             and cards2.quantity  <> cards3.quantity
             and cards3.quantity  <> cards1.quantity
            )
        )
```

```
where  1=1
and    gameofset_predicate ( columns ( symbol    ) ) = 1
and    gameofset_predicate ( columns ( color     ) ) = 1
and    gameofset_predicate ( columns ( shading   ) ) = 1
and    gameofset_predicate ( columns ( quantity  ) ) = 1
```

```
          ,
   where  rownum <= 12
  )
,    solution as
  (select c1.symbol   as symbol1
        , c1.color    as color1
        , c1.shading  as shading1
        , c1`.quantity as quantity1
        , c2.symbol   as symbol2
        , c2.color    as color2
        , c2.shading  as shading2
        , c2.quantity as quantity2
        , c3.symbol   as symbol3
        , c3.color    as color3
        , c3.shading  as shading3
        , c3.quantity as quantity3
   from   puzzle c1
   join   puzzle c2
     on   c1.cardno < c2.cardno
   join   puzzle c3
     on   c2.cardno < c3.cardno
   where  1=1
   and    gameofset_predicate( columns( symbol   ) ) = 1
   and    gameofset_predicate( columns( color    ) ) = 1
   and    gameofset_predicate( columns( shading  ) ) = 1
   and    gameofset_predicate( columns( quantity ) ) = 1
  )
select 'puzzle'   as what
     ,               symbol,          color
     ,               shading,         quantity
     , null       as symbol,  null as color
     , null       as shading, null as quantity
     , null       as symbol,  null as color
     , null       as shading, null as quantity
from   puzzle
union  all
```

```
          ,
   where  rownum <= 12
  )
select 'puzzle'   as what
     ,               symbol,          color
     ,               shading,         quantity
     , null       as symbol,  null as color
     , null       as shading, null as quantity
     , null       as symbol,  null as color
     , null       as shading, null as quantity
from   puzzle
union  all
select 'solution' as what
     , c1.symbol   as symbol1
     , c1.color    as color1
     , c1.shading  as shading1
     , c1.quantity as quantity1
     , c2.symbol   as symbol2
     , c2.color    as color2
     , c2.shading  as shading2
     , c2.quantity as quantity2
     , c3.symbol   as symbol3
     , c3.color    as color3
     , c3.shading  as shading3
     , c3.quantity as quantity3
from  puzzle c1
join   puzzle c2
  on   c1.cardno < c2.cardno
join   puzzle c3
  on   c2.cardno < c3.cardno
where  1=1
and    gameofset_predicate( columns( symbol   ) ) = 1
and    gameofset_predicate( columns( color    ) ) = 1
and    gameofset_predicate( columns( shading  ) ) = 1
and    gameofset_predicate( columns( quantity ) ) = 1
/
```

```
with symbols as
  (          select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual

  )
,    colors   as
 (           select 'green'    color    from dual
   union all select 'purple'            from dual
   union all select 'red'               from dual

  )
,    shadings as
 (           select 'filled'   shading  from dual
   union all select 'striped'           from dual
   union all select 'open'              from dual

  )
,    numbers  as
 (           select 1          quantity from dual
   union all select 2                   from dual
   union all select 3                   from dual

  )
,    deck     as
  (select symbol, color, shading, quantity
   from       symbols
   cross join colors
   cross join shadings
   cross join numbers
  )
,    puzzle   as
  (select rownum as cardno, symbol, color
```

```
with symbols as
  (          select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual
   union all select 'square'            from dual -- fourth symbol
  )
,    colors   as
 (           select 'green'    color    from dual
   union all select 'purple'            from dual
   union all select 'red'               from dual
   union all select 'yellow'            from dual -- fourth color
  )
,    shadings as
 (           select 'filled'   shading  from dual
   union all select 'striped'           from dual
   union all select 'open'              from dual
   union all select 'dotted'            from dual -- fourth shading
  )
,    numbers  as
 (           select 1          quantity from dual
   union all select 2                   from dual
   union all select 3                   from dual
   union all select 4                   from dual -- fourth quantity
  )
,    deck     as
  (select symbol, color, shading, quantity
   from       symbols
   cross join colors
   cross join shadings
   cross join numbers
  )
,    puzzle   as
  (select rownum as cardno, symbol, color
```

```
,    puzzle   as
  (select rownum as cardno, symbol, color
        , shading         , quantity
   from   (select *
           from   deck
           order  by dbms_random.value
          )
   -- increase this number to increase the number of sets present
   where  rownum <= 12
  )
select 'puzzle'   as what
     , symbol     as symbol1 ,  color    as color1
     , shading    as shading1, quantity as quantity1
     , null       as symbol2 ,  null as color2
     , null       as shading2, null as quantity2
     , null       as symbol3 ,  null as color3
     , null       as shading3, null as quantity3


from   puzzle
union  all
select 'solution' as what
     , c1.symbol  as symbol1 , c1.color    as color1
     , c1.shading as shading1, c1.quantity as quantity1
     , c2.symbol  as symbol2 , c2.color    as color2
     , c2.shading as shading2, c2.quantity as quantity2
     , c3.symbol  as symbol3 , c3.color    as color3
     , c3.shading as shading3, c3.quantity as quantity3


from   puzzle c1
join   puzzle c2
  on   c1.cardno < c2.cardno
join   puzzle c3
  on   c2.cardno < c3.cardno
```

```
,    puzzle   as
  (select rownum as cardno, symbol, color
        , shading         , quantity
   from   (select *
           from   deck
           order  by dbms_random.value
          )
   -- increase this number to increase the number of sets present
   where  rownum <= 20
  )
select 'puzzle'   as what
     , symbol     as symbol1 ,  color    as color1
     , shading    as shading1, quantity as quantity1
     , null       as symbol2 ,  null as color2
     , null       as shading2, null as quantity2
     , null       as symbol3 ,  null as color3
     , null       as shading3, null as quantity3
     , null       as symbol4 ,  null as color4
     , null       as shading4, null as quantity4
from   puzzle
union  all
select 'solution' as what
     , c1.symbol  as symbol1 , c1.color    as color1
     , c1.shading as shading1, c1.quantity as quantity1
     , c2.symbol  as symbol2 , c2.color    as color2
     , c2.shading as shading2, c2.quantity as quantity2
     , c3.symbol  as symbol3 , c3.color    as color3
     , c3.shading as shading3, c3.quantity as quantity3
     , c4.symbol  as symbol4 , c4.color    as color4
     , c4.shading as shading4, c4.quantity as quantity4
from   puzzle c1
join   puzzle c2
  on   c1.cardno < c2.cardno
join   puzzle c3
  on   c2.cardno < c3.cardno
join   puzzle c4
```

```
     , null       as shading4, null as quantity4
from   puzzle
union  all
select 'solution' as what
     , c1.symbol  as symbol1 , c1.color    as color1
     , c1.shading as shading1, c1.quantity as quantity1
     , c2.symbol  as symbol2 , c2.color    as color2
     , c2.shading as shading2, c2.quantity as quantity2
     , c3.symbol  as symbol3 , c3.color    as color3
     , c3.shading as shading3, c3.quantity as quantity3
     , c4.symbol  as symbol4 , c4.color    as color4
     , c4.shading as shading4, c4.quantity as quantity4
from   puzzle c1
join   puzzle c2
  on   c1.cardno < c2.cardno
join   puzzle c3
  on   c2.cardno < c3.cardno
join   puzzle c4
  on   c3.cardno < c4.cardno
where  1=1
and    gameofset_predicate( columns( symbol   ) ) = 1
and    gameofset_predicate( columns( color    ) ) = 1
and    gameofset_predicate( columns( shading  ) ) = 1
and    gameofset_predicate( columns( quantity ) ) = 1
/
```

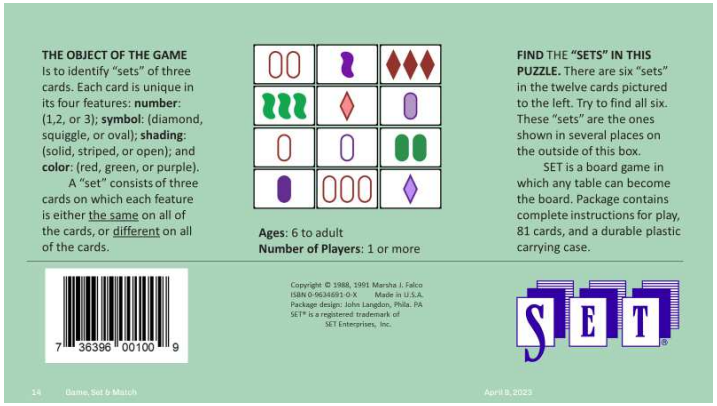
```
create or replace function gameofset_predicate
( col_in in dbms_tf.columns_t )
return varchar2 sql_macro( scalar ) is
  lsql varchar2(32767);
begin
  lsql := '
( case
    when (    (     c1.' || col_in( 1 ) || '  = c2.' || col_in( 1 ) || '
                and c2.' || col_in( 1 ) || '  = c3.' || col_in( 1 ) || '

             )
           or (     c1.' || col_in( 1 ) || ' <> c2.' || col_in( 1 ) || '
                and c1.' || col_in( 1 ) || ' <> c3.' || col_in( 1 ) || '

                and c2.' || col_in( 1 ) || ' <> c3.' || col_in( 1 ) || '


             )
         ) then 1
    else 0
  end
)';
  return lsql;
end;
/
```

```
create or replace function gameofset_predicate
( col_in in dbms_tf.columns_t )
return varchar2 sql_macro( scalar ) is
  lsql varchar2(32767);
begin
  lsql := '
( case
    when (    (     c1.' || col_in( 1 ) || '  = c2.' || col_in( 1 ) || '
                and c2.' || col_in( 1 ) || '  = c3.' || col_in( 1 ) || '
                and c3.' || col_in( 1 ) || '  = c4.' || col_in( 1 ) || '
              )
           or (     c1.' || col_in( 1 ) || ' <> c2.' || col_in( 1 ) || '
                and c1.' || col_in( 1 ) || ' <> c3.' || col_in( 1 ) || '
                and c1.' || col_in( 1 ) || ' <> c4.' || col_in( 1 ) || '
                and c2.' || col_in( 1 ) || ' <> c3.' || col_in( 1 ) || '
                and c2.' || col_in( 1 ) || ' <> c4.' || col_in( 1 ) || '
                and c3.' || col_in( 1 ) || ' <> c4.' || col_in( 1 ) || '
              )
         ) then 1
    else 0
  end
)';
  return lsql;
end;
/
```



THE OBJECT OF THE GAME
Is to identify "sets" of three cards. Each card is unique in its four features: **number**: (1,2, or 3); **symbol**: (diamond, squiggle, or oval); **shading**: (solid, striped, or open); and **color**: (red, green, or purple).

A "set" consists of three cards on which each feature is either the same on all of the cards, or different on all of the cards.

Ages: 6 to adult
Number of Players: 1 or more

FIND THE **"SETS" IN THIS PUZZLE.** There are six "sets" in the twelve cards pictured to the left. Try to find all six. These "sets" are the ones shown in several places on the outside of this box.

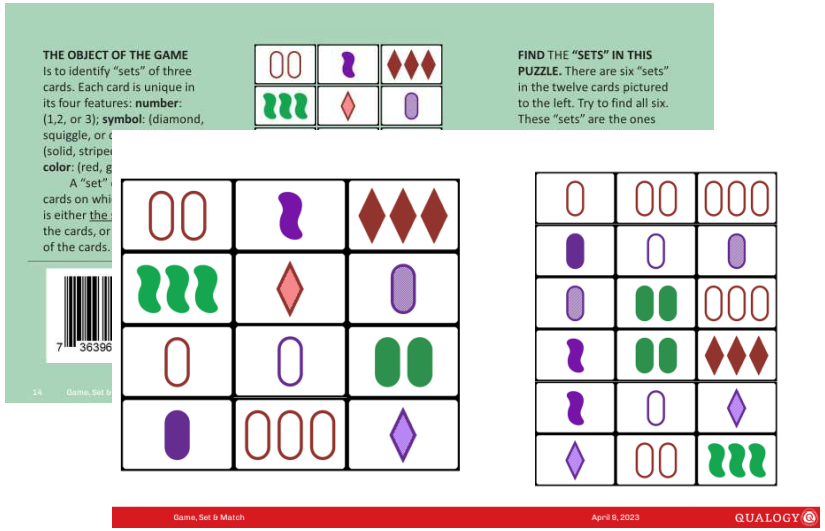
SET is a board game in which any table can become the board. Package contains complete instructions for play, 81 cards, and a durable plastic carrying case.

7 36396 00100 9

Copyright © 1988, 1991 Marsha J. Falco
ISBN 0-9634691-0-X Made in U.S.A.
Package design: John Langdon, Phila. PA
SET™ is a registered trademark of
SET Enterprises, Inc.

SET®

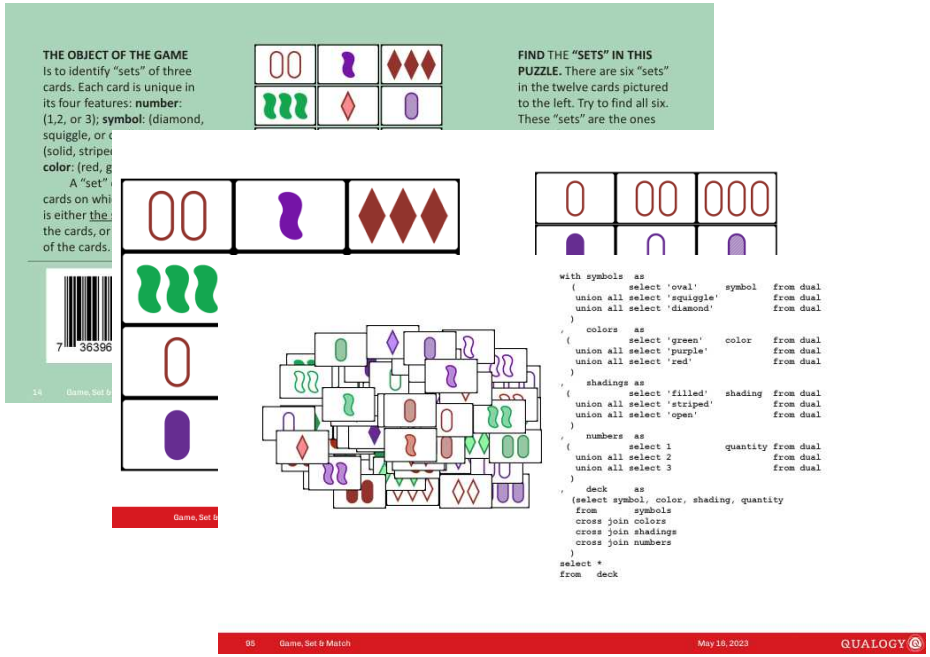
14 Game, Set & Match April 9, 2023



THE OBJECT OF THE GAME Is to identify "sets" of three cards. Each card is unique in its four features: **number**: (1,2, or 3); **symbol**: (diamond, squiggle, or
(solid, stripe
color: (red, g
A "set"
cards on whi
is either the
the cards, or
of the cards.

FIND THE **"SETS" IN THIS PUZZLE.** There are six "sets" in the twelve cards pictured to the left. Try to find all six. These "sets" are the ones

Game, Set & Match April 9, 2023 QUALOGY

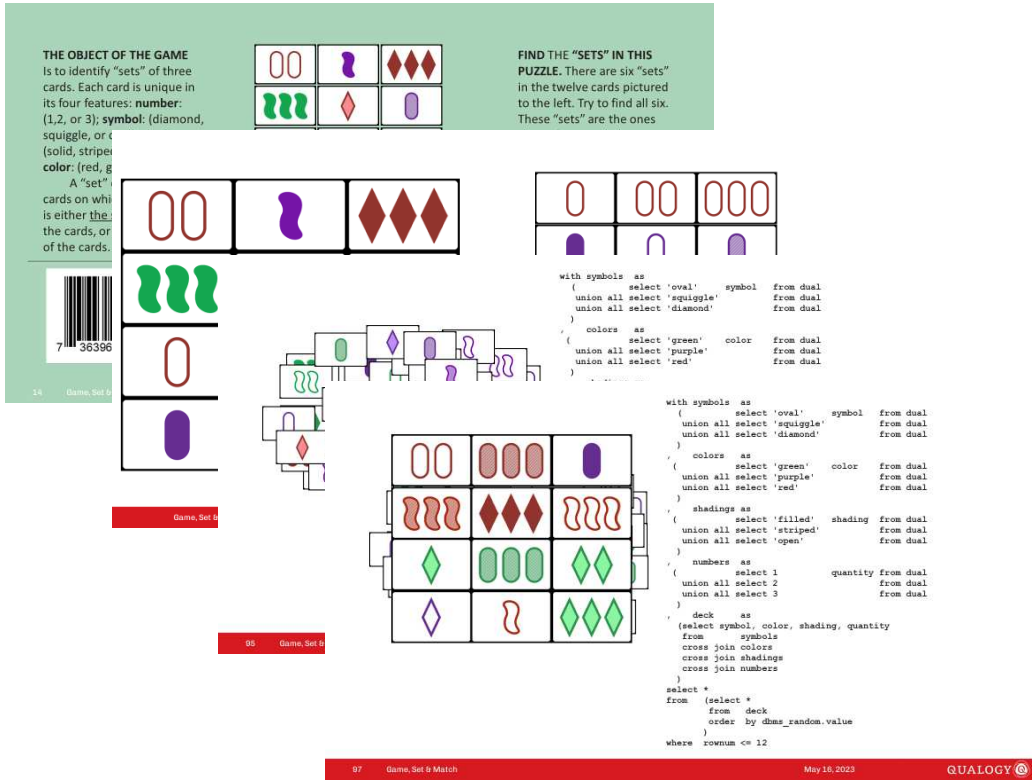


THE OBJECT OF THE GAME Is to identify "sets" of three cards. Each card is unique in its four features: **number**: (1,2, or 3); **symbol**: (diamond, squiggle, or
(solid, stripe
color: (red, g
A "set"
cards on whi
is either the
the cards, or
of the cards.

FIND THE **"SETS" IN THIS PUZZLE.** There are six "sets" in the twelve cards pictured to the left. Try to find all six. These "sets" are the ones

```
with symbols  as
  (                 select 'oval'      symbol   from dual
   union all select 'squiggle'                  from dual
   union all select 'diamond'                   from dual
  )
,    colors   as
  (                 select 'green'     color    from dual
   union all select 'purple'                    from dual
   union all select 'red'                       from dual
  )
,    shadings as
  (                 select 'filled'    shading  from dual
   union all select 'striped'                   from dual
   union all select 'open'                      from dual
  )
,    numbers  as
  (                 select 1           quantity from dual
   union all select 2                           from dual
   union all select 3                           from dual
  )
,    deck     as
  (select symbol, color, shading, quantity
   from       symbols
   cross join colors
   cross join shadings
   cross join numbers
  )
select *
from   deck
```

95 Game, Set & Match May 16, 2023 QUALOGY



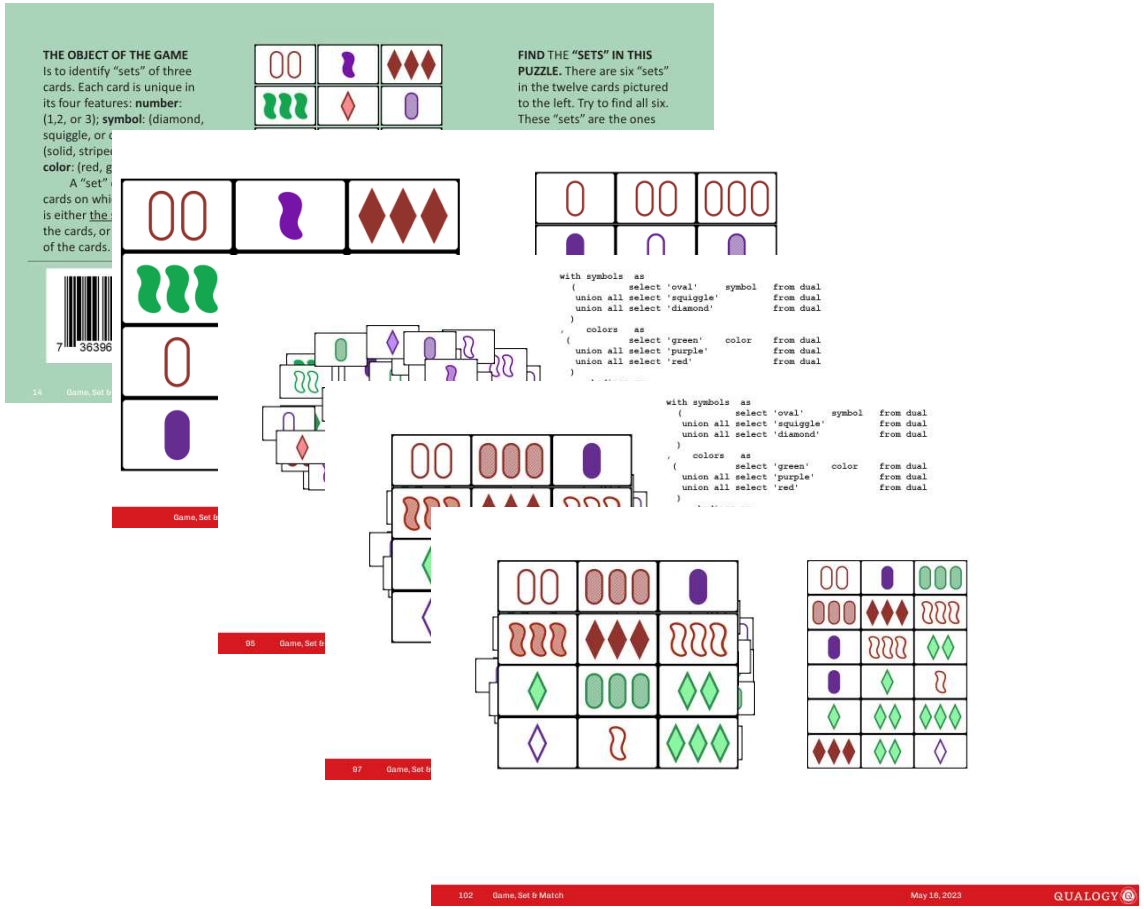
THE OBJECT OF THE GAME
Is to identify "sets" of three cards. Each card is unique in its four features: **number**: (1,2, or 3); **symbol**: (diamond, squiggle, or
(solid, stripe
color: (red, g
A "set"
cards on whi
is either the
the cards, or
of the cards.

FIND THE **"SETS" IN THIS PUZZLE.** There are six "sets" in the twelve cards pictured to the left. Try to find all six. These "sets" are the ones

```
with symbols  as
  (           select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual
  )
,    colors   as
 (            select 'green'    color    from dual
   union all select 'purple'            from dual
   union all select 'red'               from dual
  )
```

```
with symbols  as
  (           select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual
  )
,    colors   as
 (            select 'green'    color    from dual
   union all select 'purple'            from dual
   union all select 'red'               from dual
  )
,    shadings as
 (            select 'filled'   shading  from dual
   union all select 'striped'           from dual
   union all select 'open'              from dual
  )
,    numbers  as
 (            select 1          quantity from dual
   union all select 2                    from dual
   union all select 3                    from dual
  )
,    deck     as
  (select symbol, color, shading, quantity
   from        symbols
   cross join colors
   cross join shadings
   cross join numbers
  )
select *
from   (select *
        from   deck
        order  by dbms_random.value
       )
where  rownum <= 12
```

97 Game, Set & Match May 16, 2023 QUALOGY



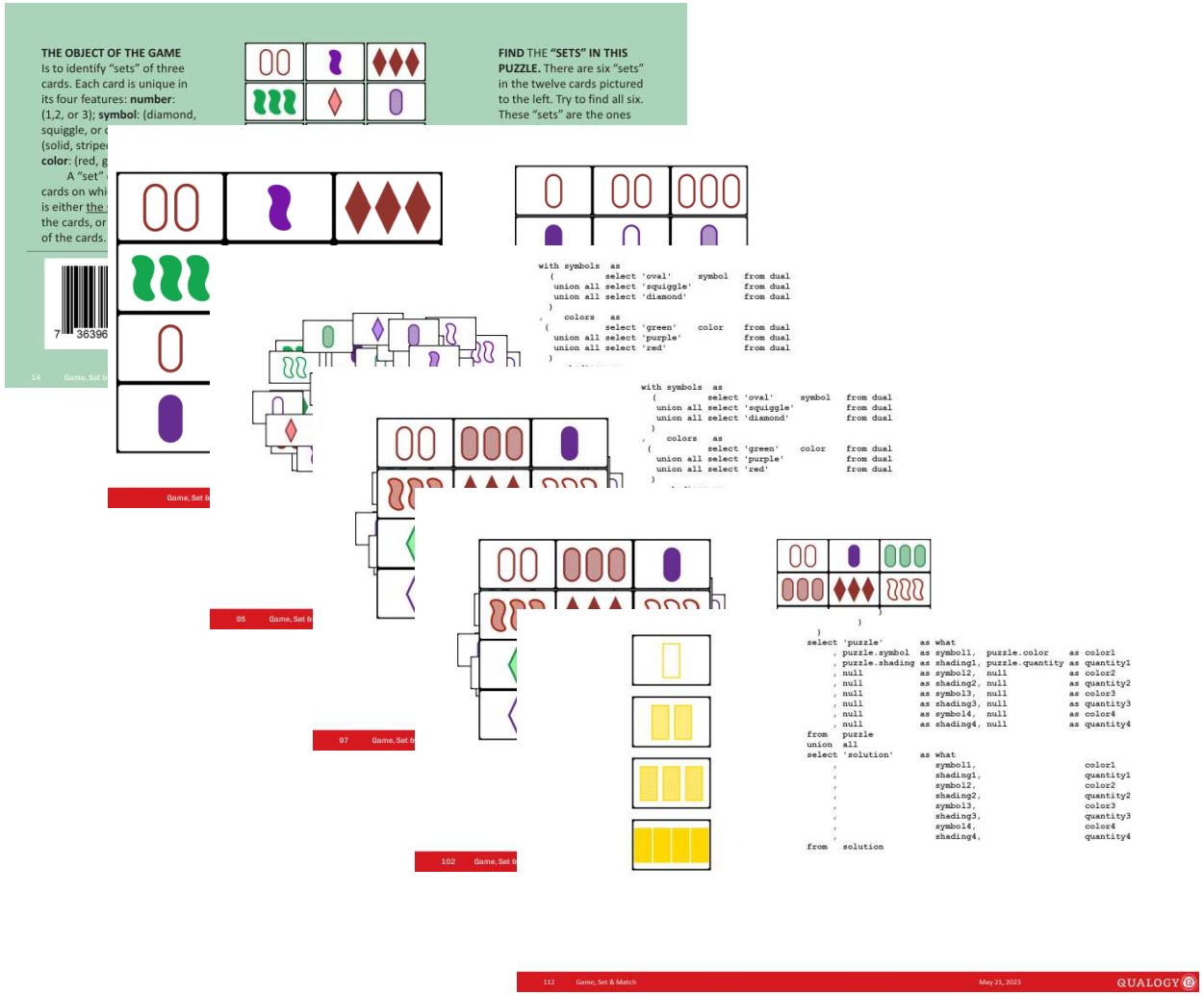
THE OBJECT OF THE GAME Is to identify "sets" of three cards. Each card is unique in its four features: **number**: (1,2, or 3); **symbol**: (diamond, squiggle, or
(solid, stripe
color: (red, g
A "set"
cards on whi
is either the
the cards, or
of the cards.

FIND THE **"SETS" IN THIS PUZZLE.** There are six "sets" in the twelve cards pictured to the left. Try to find all six. These "sets" are the ones

```
with symbols  as
  (                 select 'oval'      symbol   from dual
   union all select 'squiggle'                  from dual
   union all select 'diamond'                   from dual
  )
,    colors   as
  (                 select 'green'     color    from dual
   union all select 'purple'                    from dual
   union all select 'red'                       from dual
  )
```

```
with symbols  as
  (                 select 'oval'      symbol   from dual
   union all select 'squiggle'                  from dual
   union all select 'diamond'                   from dual
  )
,    colors   as
  (                 select 'green'     color    from dual
   union all select 'purple'                    from dual
   union all select 'red'                       from dual
  )
```

102 Game, Set & Match May 18, 2023 QUALOGY



THE OBJECT OF THE GAME Is to identify "sets" of three cards. Each card is unique in its four features: **number**: (1,2, or 3); **symbol**: (diamond, squiggle, or
(solid, stripe
color: (red, g
A "set"
cards on whi
is either the
the cards, or
of the cards.

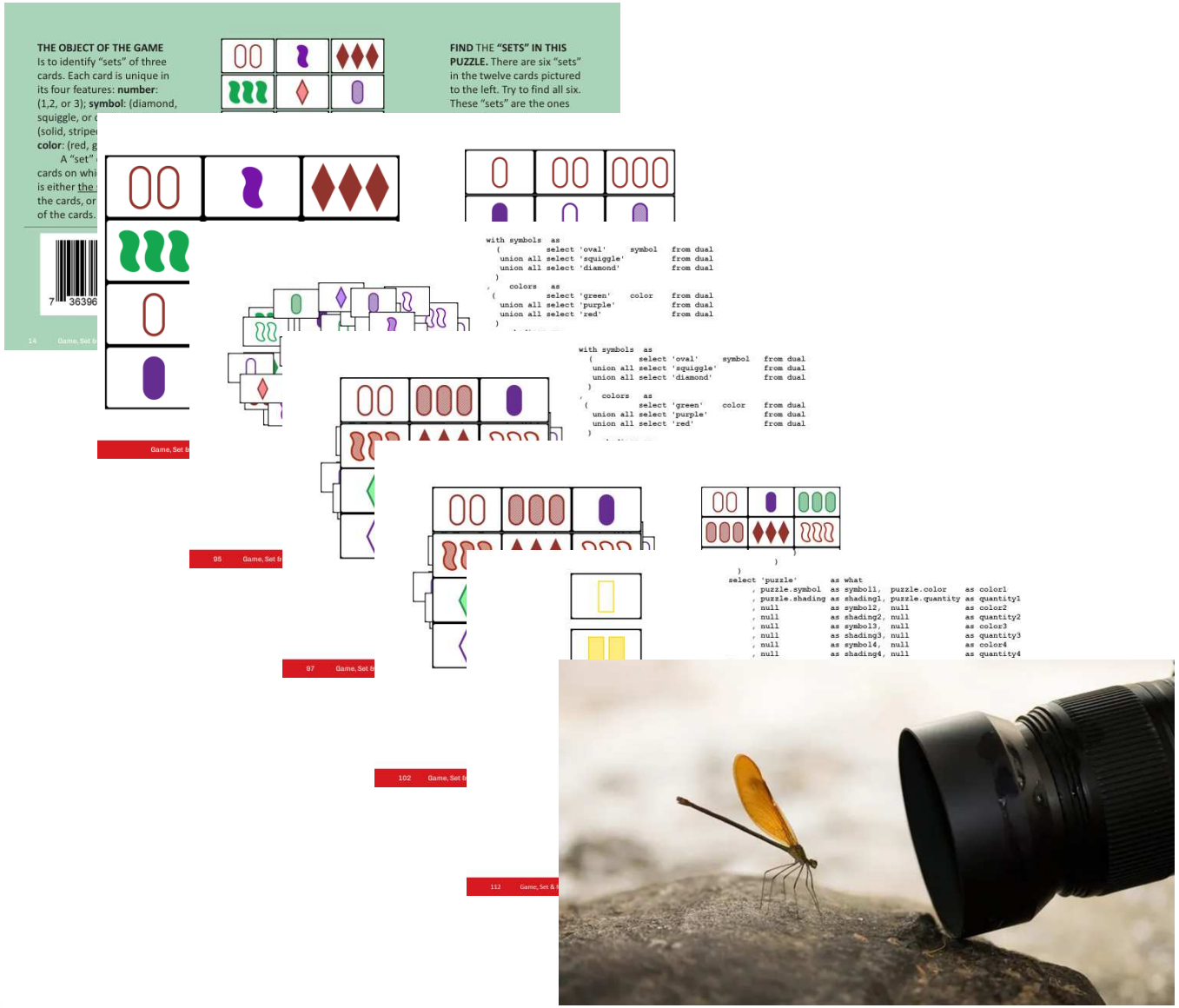
FIND THE **"SETS" IN THIS PUZZLE.** There are six "sets" in the twelve cards pictured to the left. Try to find all six. These "sets" are the ones

```
with symbols  as
  (           select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual
  )
,    colors   as
  (           select 'green'    color    from dual
   union all select 'purple'            from dual
   union all select 'red'               from dual
  )
```

```
with symbols  as
  (           select 'oval'     symbol   from dual
   union all select 'squiggle'          from dual
   union all select 'diamond'           from dual
  )
,    colors   as
  (           select 'green'    color    from dual
   union all select 'purple'            from dual
   union all select 'red'               from dual
  )
```

```
        )
  )
select 'puzzle'          as what
     , puzzle.symbol     as symbol1,  puzzle.color     as color1
     , puzzle.shading    as shading1, puzzle.quantity  as quantity1
     , null              as symbol2,  null             as color2
     , null              as shading2, null             as quantity2
     , null              as symbol3,  null             as color3
     , null              as shading3, null             as quantity3
     , null              as symbol4,  null             as color4
     , null              as shading4, null             as quantity4
from   puzzle
union  all
select 'solution'        as what
     ,                      symbol1,                    color1
     ,                      shading1,                   quantity1
     ,                      symbol2,                    color2
     ,                      shading2,                   quantity2
     ,                      symbol3,                    color3
     ,                      shading3,                   quantity3
     ,                      symbol4,                    color4
     ,                      shading4,                   quantity4
from   solution
```

112 Game, Set & Match May 21, 2023 QUALOGY



THE OBJECT OF THE GAME Is to identify "sets" of three cards. Each card is unique in its four features: **number**: (1,2, or 3); **symbol**: (diamond, squiggle, or
(solid, stripe
color: (red, g
A "set"
cards on whi
is either the
the cards, or
of the cards.

FIND THE **"SETS" IN THIS PUZZLE.** There are six "sets" in the twelve cards pictured to the left. Try to find all six. These "sets" are the ones

```
with symbols  as
  (             select 'oval'     symbol   from dual
   union all select 'squiggle'            from dual
   union all select 'diamond'             from dual
  )
,    colors   as
  (             select 'green'    color    from dual
   union all select 'purple'              from dual
   union all select 'red'                 from dual
  )
```

```
with symbols  as
  (             select 'oval'     symbol   from dual
   union all select 'squiggle'            from dual
   union all select 'diamond'             from dual
  )
,    colors   as
  (             select 'green'    color    from dual
   union all select 'purple'              from dual
   union all select 'red'                 from dual
  )
```

```
  )
)
select 'puzzle'         as what
     , puzzle.symbol    as symbol1,  puzzle.color     as color1
     , puzzle.shading   as shading1, puzzle.quantity  as quantity1
     , null             as symbol2,  null             as color2
     , null             as shading2, null             as quantity2
     , null             as symbol3,  null             as color3
     , null             as shading3, null             as quantity3
     , null             as symbol4,  null             as color4
     , null             as shading4, null             as quantity4
```

Q&A