

# SIMPLIFY DATABASE CODE WITH utPLSQL

Philipp Salvisberg  
21<sup>st</sup> September 2022

## 2 WELCOME



### PHILIPP SALVISBERG SENIOR PRINCIPAL CONSULTANT

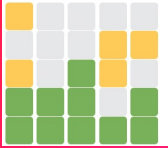
- Database centric development
- Model Driven Software Development
- Author of free SQL Developer Extensions  
PL/SQL Unwrapper, db\* CODECOP, utPLSQL,  
plscope-utils, oddgen and Bitemp Remodeler



Wordle 208 3/6



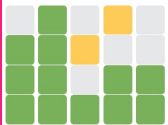
Wordle 209 5/6



Wordle 210 3/6



Wordle 211 4/6



Wordle 212 3/6



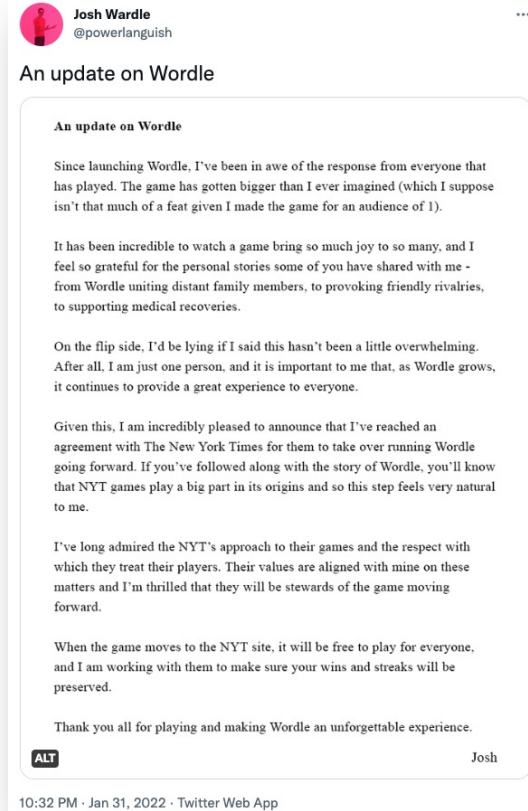
# WORDLE

## 4 INVENTOR

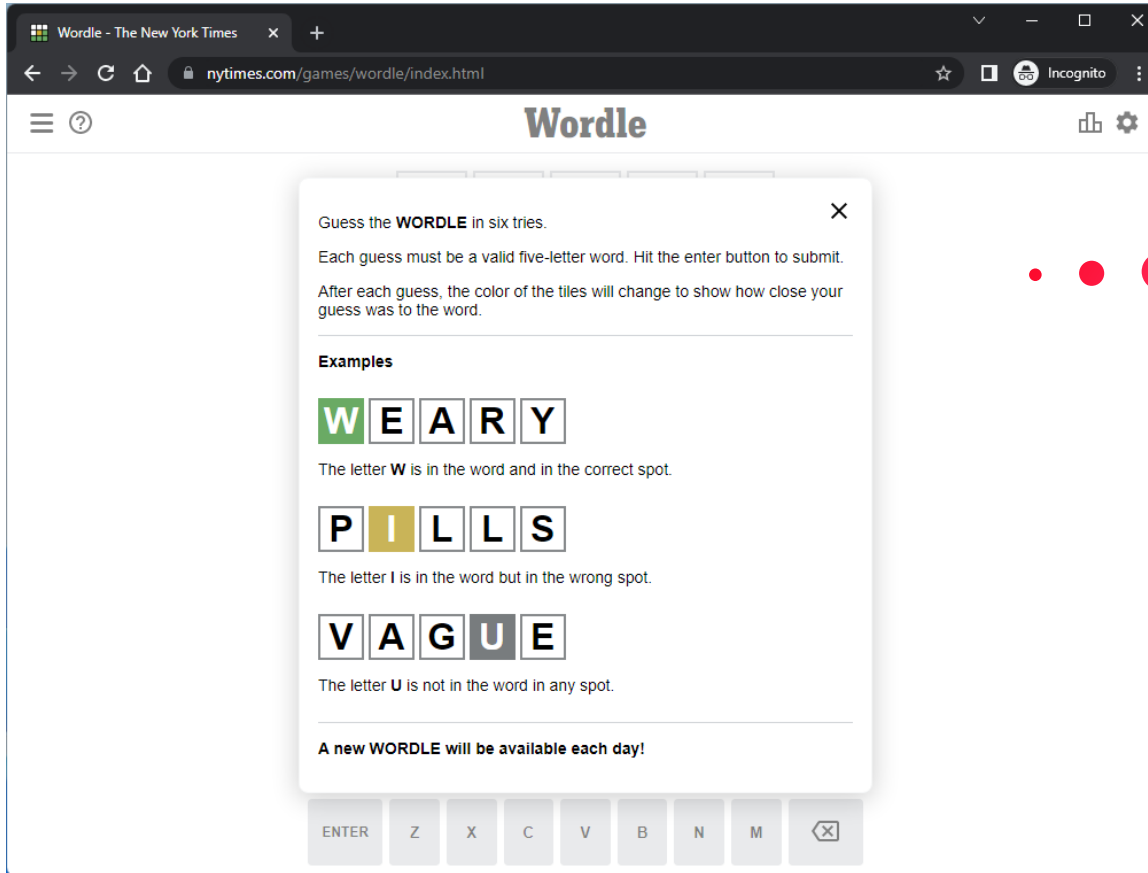


Josh Wardle  
[@powerlanguage](https://twitter.com/powerlanguage)

<https://www.powerlanguage.co.uk/wordle/>  
<https://www.nytimes.com/games/wordle/index.html>



## 5 RULES



14'855  
accepted words

2'309  
used as solution

## 6 SETTINGS

The screenshot shows the Wordle settings page in a browser. The page title is "SETTINGS" with a close button (X). The settings include:

- Hard Mode** (toggle on): Any revealed hints must be used in subsequent guesses.
- Dark Theme** (toggle off).
- High Contrast Mode** (toggle off): For improved color vision.
- Feedback** (link to [Email](#)).
- Community** (link to [Twitter](#)).
- Questions?** (link to [FAQ](#)).

Annotations on the screenshot:

- A thought bubble next to "Dark Theme" says "#459 means 2022-09-21".
- A thought bubble next to "Community" says "#0 means 2021-06-19".
- A speech bubble at the bottom says "Game number".
- A small "#0" is visible at the bottom right of the page.

© 2021 The New York Times Company

after

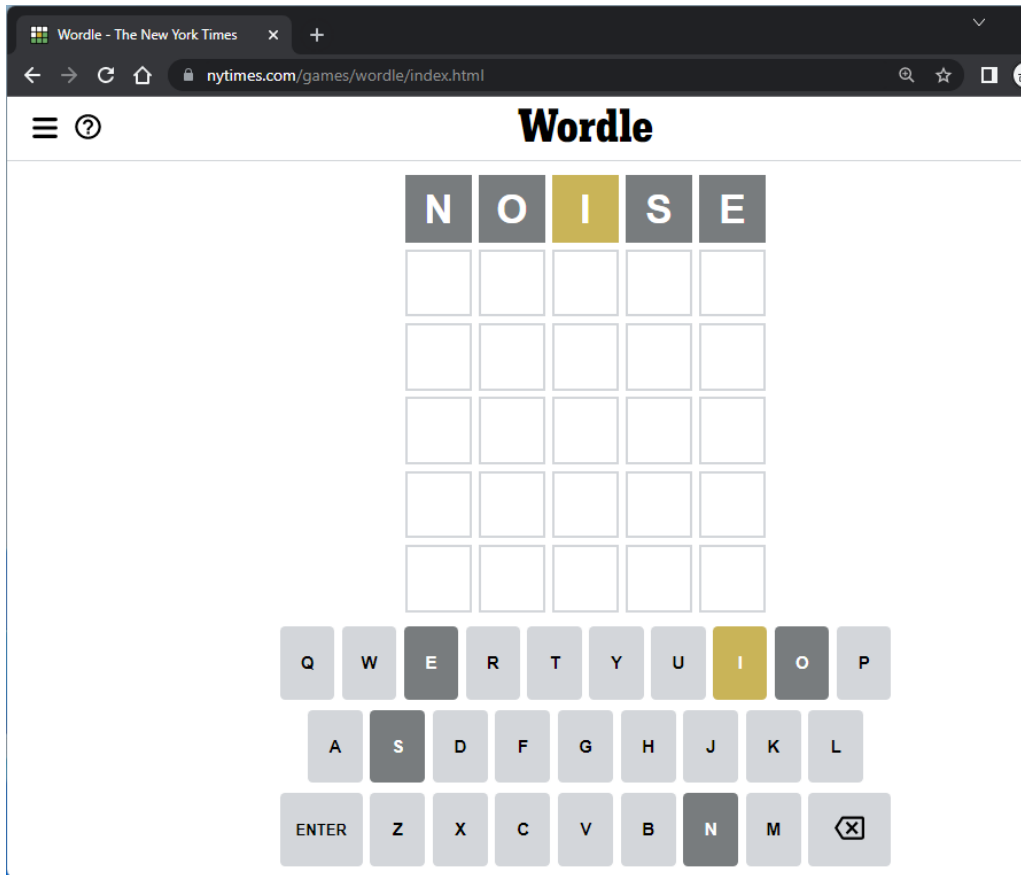
R	I	G	H	T
---	---	---	---	---

next guesses must end on "ight"

eight, fight, light,  
might, night, sight,  
tight, wight, ...

~~flame, ens, ...~~

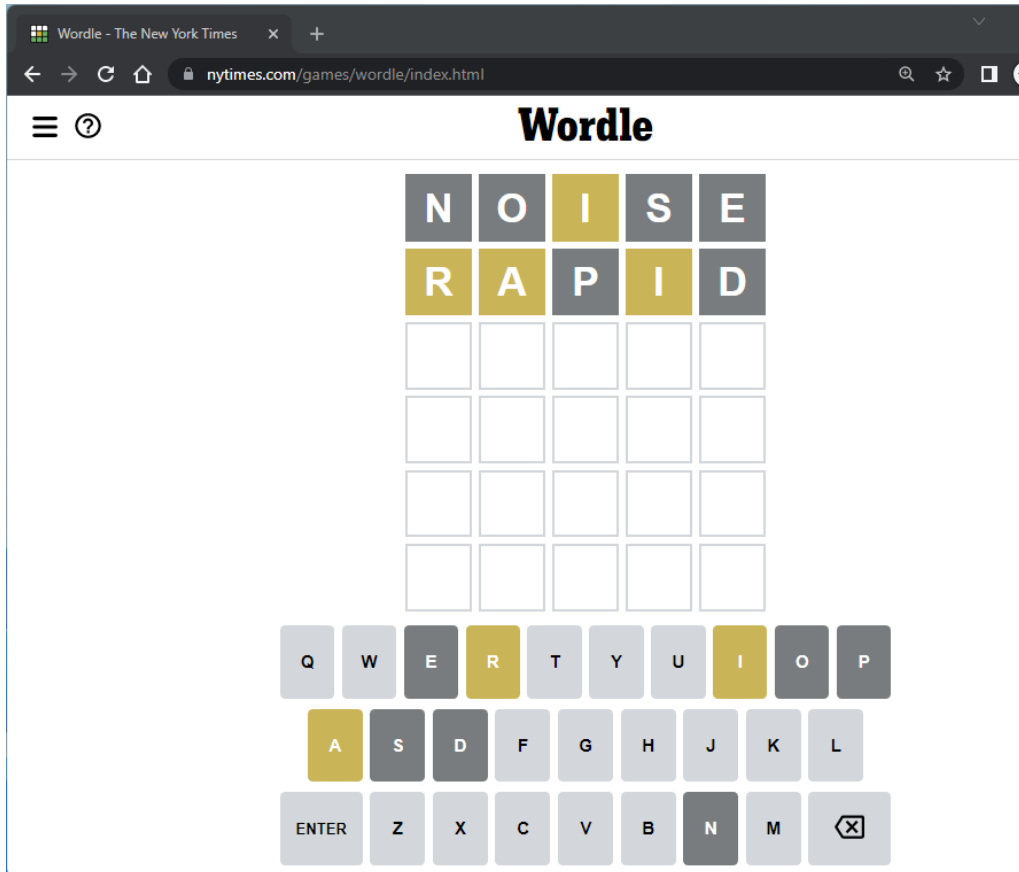
## 7 FIRST GUESS



```
select word
from words
where word not like '___i___'
and instr(word, 'i', 1, 1) > 0
and word not like '%n%'
and word not like '%o%'
and word not like '%s%'
and word not like '%e%';
```

636 rows selected.

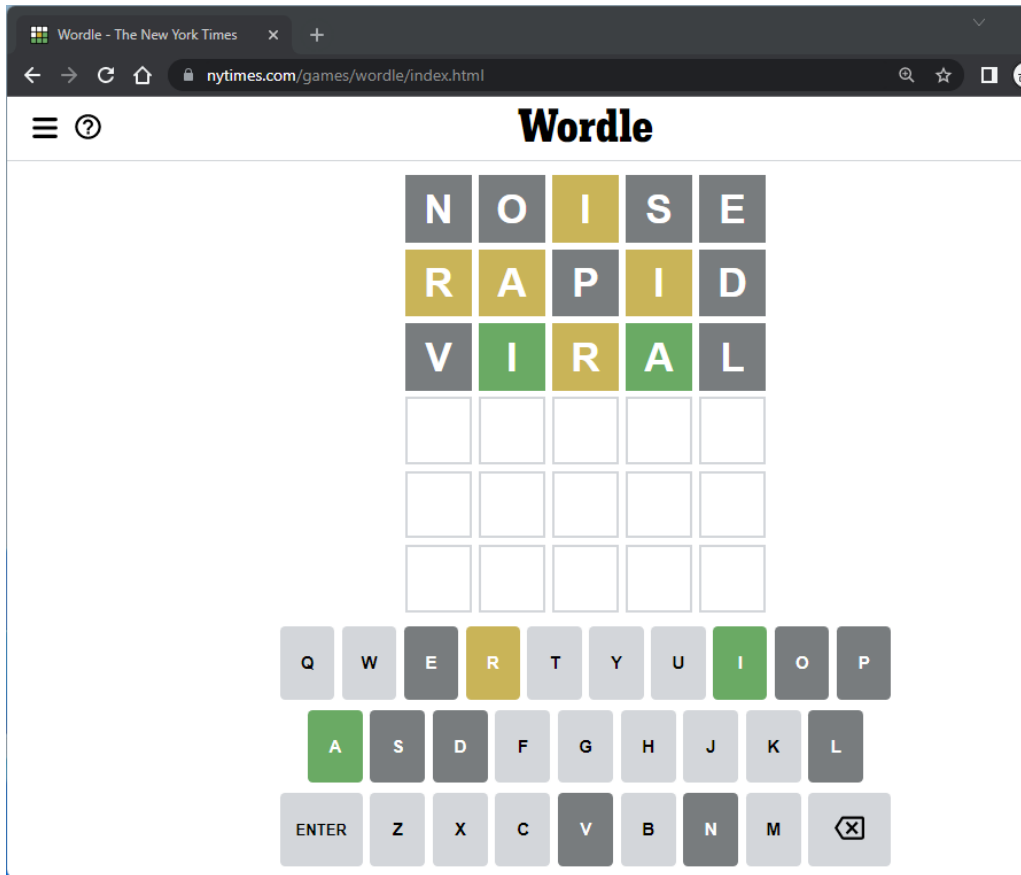
## 8 SECOND GUESS



```
select word
from words
where word not like ' _i _'
and instr(word, 'i', 1, 1) > 0
and word not like '%n%'
and word not like '%o%'
and word not like '%s%'
and word not like '%e%'
-- second guess
and word not like 'r _ _ _'
and word not like ' _a _ _'
and word not like ' _ _i _'
and instr(word, 'r', 1, 1) > 0
and instr(word, 'a', 1, 1) > 0
and word not like '%p%'
and word not like '%d%';
```

39 rows selected.

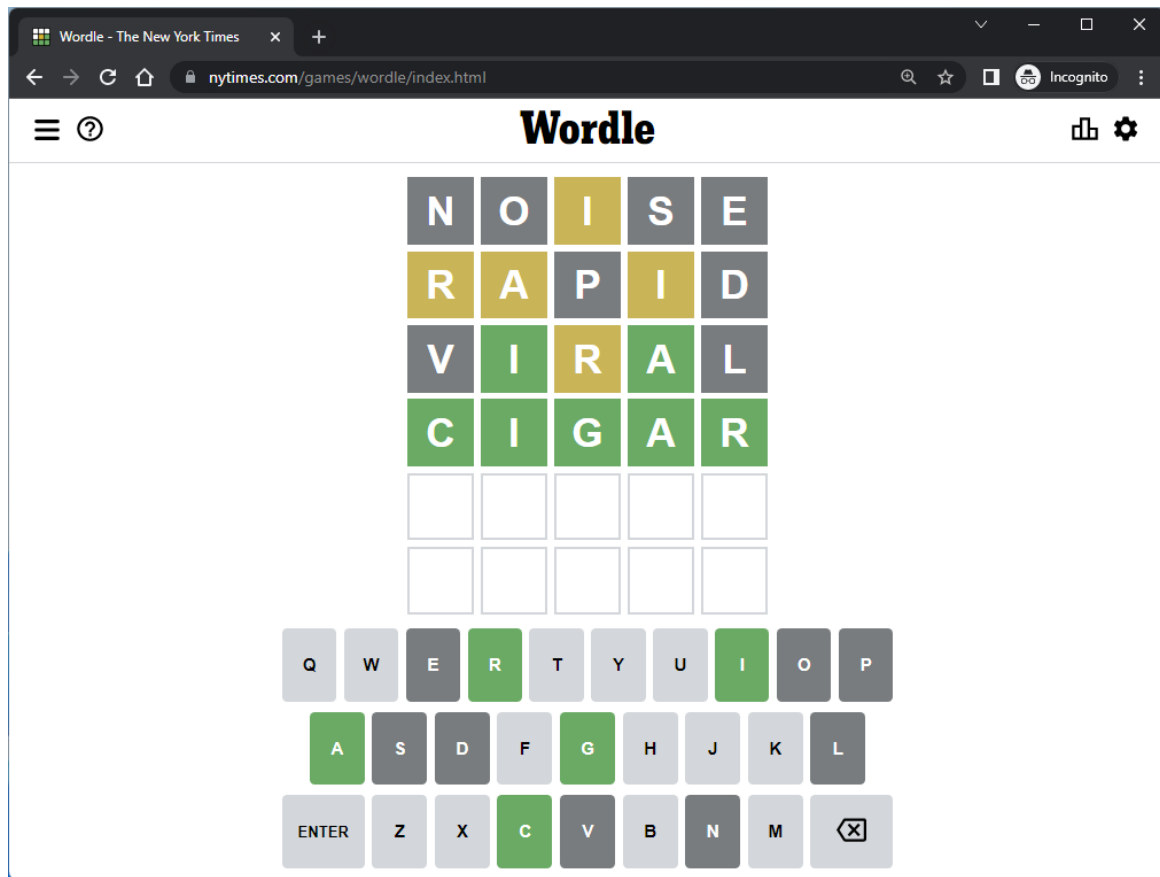
## 9 THIRD GUESS



```
select word
from words
where word not like '__i__'
  and instr(word, 'i', 1, 1) > 0
  and word not like '%n%'
  and word not like '%o%'
  and word not like '%s%'
  and word not like '%e%'
-- second guess
and word not like 'r____'
and word not like '_a____'
and word not like '____i__'
and instr(word, 'r', 1, 1) > 0
and instr(word, 'a', 1, 1) > 0
and word not like '%p%'
and word not like '%d%'
-- third guess
and word like '_i_a_'
and word not like '____r__'
and word not like '%v%'
and word not like '%l%';
```

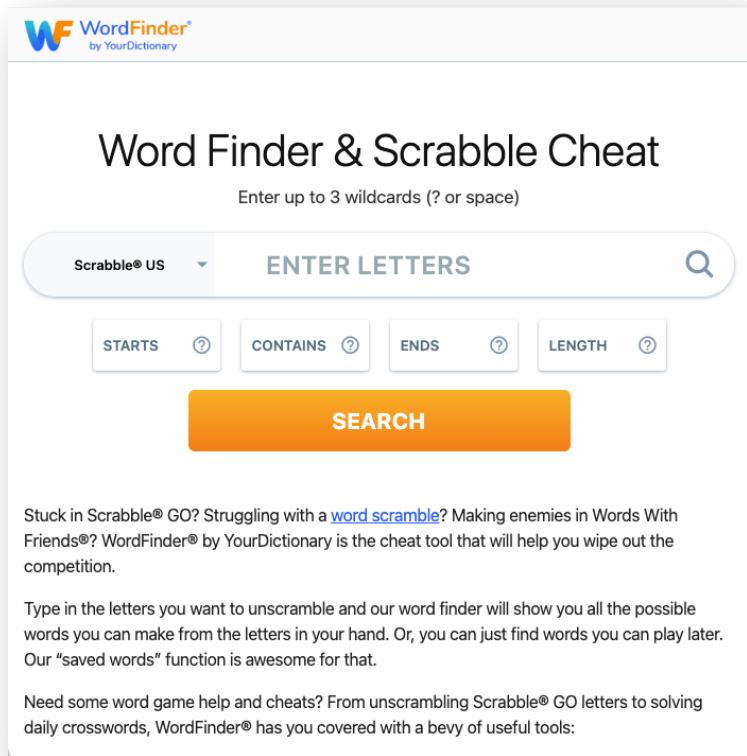
3 rows selected.

## 10 FOURTH GUESS



# COMMON WORDLE HELPER

## 12 WORD FINDER BY YourDictionary




The screenshot shows the WordFinder & Scrabble Cheat interface. At the top left is the logo 'WF WordFinder by YourDictionary'. The main heading is 'Word Finder & Scrabble Cheat' with a subtitle 'Enter up to 3 wildcards (? or space)'. Below this is a search bar with a dropdown menu set to 'Scrabble® US' and a placeholder 'ENTER LETTERS'. To the right of the search bar is a magnifying glass icon. Below the search bar are four filter buttons: 'STARTS', 'CONTAINS', 'ENDS', and 'LENGTH', each with a question mark icon. Below these filters is a large orange 'SEARCH' button. At the bottom, there is a paragraph of text: 'Stuck in Scrabble® GO? Struggling with a [word scramble](#)? Making enemies in Words With Friends®? WordFinder® by YourDictionary is the cheat tool that will help you wipe out the competition. Type in the letters you want to unscramble and our word finder will show you all the possible words you can make from the letters in your hand. Or, you can just find words you can play later. Our "saved words" function is awesome for that. Need some word game help and cheats? From unscrambling Scrabble® GO letters to solving daily crosswords, WordFinder® has you covered with a bevy of useful tools:'

Limited Search  
Features

Source: <https://wordfinder.yourdictionary.com/>

# 13 THE FREE DICTIONARY WORD FINDER

The Free Dictionary Word Finder 

Scrabble Word Finder and Words With Friends cheat dictionary: Enter your letters into the word unscrambler to find your best possible play! Every word solver search provides options for Scrabble, Words With Friends, [WordHub](#), Wordle, and crossword help. Plus: Boost your vocabulary in [Spelling Bee](#) and [Hangman](#)!

[Correct all you're your grammar errors instantly. Try it now.](#)

### Wordle Word Finder

If you've made a guess in Wordle, type that word below using **?** after **yellow** letters and **!** after **green** ones. If you need a good suggestion for a starting word, leave the "Word #1" box blank and press "Submit." Watch [\(video\)](#)

For example:

R	A	Z	E	D
---	---	---	---	---

Type: r?a?zed!

S	P	R	A	D
---	---	---	---	---

Type: splr?a?d!

Word #1

Word #2

Word #3

Word #4

Word #5

Excellent Search Features

Different Word List

Source: <https://www.thefreedictionary.com/Word-Finder.htm#Wordle>

# ORACLE DATABASE WORDLE HELPER

# 15 DATA MODEL

68'179 rows

LETTER_IN_WORDS		
PF *	LETTER	VARCHAR2 (1 CHAR)
PF *	WORD	VARCHAR2 (5 CHAR)
*	OCCURRENCES	INTEGER
LETTER_IN_WORDS_PK (WORD, LETTER)		
LETTER_IN_WORDS_UK (LETTER, WORD)		
LETTER_IN_WORDS_LETTERS_FK (LETTER)		
LETTER_IN_WORDS_WORDS_FK (WORD)		

26 rows

LETTERS		
P *	LETTER	VARCHAR2 (1 CHAR)
*	OCCURRENCES	INTEGER
	IS_VOWEL	case when letter in ('a', 'e', 'i', 'o', 'u') then 1 else 0 end
LETTERS_PK (LETTER)		

14'855 rows

WORDS		
P *	WORD	VARCHAR2 (5 CHAR)
*	DISTINCT_LETTERS	INTEGER
*	OCCURRENCES	INTEGER
U	GAME_ID	INTEGER
U	GAME_DATE	DATE
WORDS_PK (WORD)		
WORDS_UK1 (GAME_ID)		
WORDS_UK2 (GAME_DATE)		

	LETTER	WORD	OCCURRENCES
1	a	cigar	1
2	c	cigar	1
3	g	cigar	1
4	i	cigar	1
5	r	cigar	1

	LETTER	OCCURRENCES	IS_VOWEL
1	a	5990	1
2	c	2028	0
3	g	1644	0
4	i	3759	1
5	r	4158	0

	WORD	DISTINCT_LETTERS	OCCURRENCES	GAME_ID	GAME_DATE
1	cigar	5	17579	0	19.06.2021 00:00:00

## 16 FIRST GUESS

```
SQL> set pagesize 1000
SQL> set linesize 250
SQL> exec wordle.set_hard_mode
```

PL/SQL procedure successfully completed.

```
SQL> exec wordle.set_ansiconsole
```

PL/SQL procedure successfully completed.

```
SQL> select * from wordle.play(0, 'noise');
```

Result Sequence

-----  
N O I S E

suggestions:

trail  
tapir  
grail  
tidal  
plait  
rapid  
diary  
flair  
frail  
acrid

14 rows selected.

```
SQL> █
```

```
SQL> set pagesize 1000
SQL> set linesize 250
SQL> exec wordle.set_hard_mode
```

PL/SQL procedure successfully completed.

```
SQL> exec wordle.set_ansiconsole(false)
```

PL/SQL procedure successfully completed.

```
SQL> select * from wordle.play(0, 'noise');
```

Result Sequence

-----  
-N- -O- (I) -S- -E-

suggestions:

trail  
tapir  
grail  
tidal  
plait  
rapid  
diary  
flair  
frail  
acrid

14 rows selected.

```
SQL> █
```

## 17 SECOND GUESS

```
SQL> select * from wordle.play(0, 'noise', 'rapid');
```

Result Sequence

N	O	I	S	E
R	A	P	I	D

suggestions:

viral  
cigar  
vicar  
tiara  
circa  
liart  
urali  
hilar  
airth  
libra

15 rows selected.

```
SQL> █
```

```
SQL> select * from wordle.play(0, 'noise', 'rapid');
```

Result Sequence

-N- -O- (I) -S- -E-  
(R) (A) -P- (I) -D-

suggestions:

viral  
cigar  
vicar  
tiara  
circa  
liart  
urali  
hilar  
airth  
libra

15 rows selected.

```
SQL> █
```

## 18 THIRD GUESS

```
SQL> select * from wordle.play(0, 'noise', 'rapid', 'viral');
```

Result Sequence

N	O	I	S	E
R	A	P	I	D
V	I	R	A	L

suggestions:

cigar  
cimar  
titar

9 rows selected.

```
SQL> █
```

```
SQL> select * from wordle.play(0, 'noise', 'rapid', 'viral');
```

Result Sequence

-N-	-O-	(I)	-S-	-E-
(R)	(A)	-P-	(I)	-D-
-V-	.I.	(R)	.A.	-L-

suggestions:

cigar  
cimar  
titar

9 rows selected.

```
SQL> █
```

## 19 FOURTH GUESS

```
SQL> select * from wordle.play(0, 'noise', 'rapid', 'viral', 'cigar');
```

Result Sequence

N	O	I	S	E
R	A	P	I	D
V	I	R	A	L
C	I	G	A	R

Bravo! You completed Wordle 0 4/6

6 rows selected.

```
SQL> █
```

```
SQL> select * from wordle.play(0, 'noise', 'rapid', 'viral', 'cigar');
```

Result Sequence

-N- -O- (I) -S- -E-  
(R) (A) -P- (I) -D-  
-V- .I. (R) .A. -L-  
.C. .I. .G. .A. .R.

Bravo! You completed Wordle 0 4/6

6 rows selected.

```
SQL> █
```

# utPLSQL

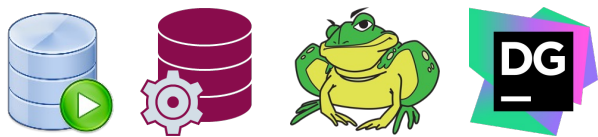
## Core Testing Framework

- Schema in the database
- No repository
- Annotation based tests



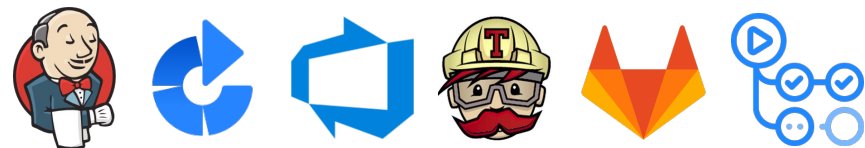
## Development

- Realtime Reporter
- Code Coverage, Code Templates, etc.



## Test Automation

- Command Line Client
- Maven Plugin
- Various Reporters



## 22 TEST DECLARATION

create or replace package test\_suite as

**--%suite**

**--%test**

procedure test\_case;

end test\_suite;

--%displayname(<description>)

**--%test(<description>)**

--%tags(<tag>[,...])

--%throws(<exception>[,...])

--%beforeall

--%afterall

--%beforeeach

--%aftereach

--%beforetest([...])

--%aftertest([...])

--%rollback(manual)

--%disabled(<reason>)

**--%suite(<description>)**

--%suitepath(<path>)

--%tags(<tag>[,...])

--%displayame(<description>)

--%beforeall([...])

--%afterall([...])

--%beforeeach([...])

--%aftereach([...])

--%rollback(manual)

--%disabled(<reason>)

--%context

--%endcontext

## 23 TEST IMPLEMENTATION

```
create or replace package body test_suite as
  procedure test_case is
    c_actual sys_refcursor;
    c_expected sys_refcursor;
  begin
    -- arrange
    wordle.set_ansiconsole(false);
    -- act (solution is cigar)
    open c_actual for select column_value from wordle.play(0, 'noise', 'rapid', 'viral') where rownum <= 3;
    -- assert
    open c_expected for
      select '-N- -O- (I) -S- -E-' as column_value from dual union all
      select '(R) (A) -P- (I) -D-' from dual;
    ut.expect(c_actual).to_equal(c_expected).unordered;
  end test_case;
end test_suite;
```

### Matcher:

be\_between, be\_empty, be\_false, be\_greater\_than,  
be\_greater\_or\_equal, be\_less\_or\_equal, be\_less\_than,  
be\_like, be\_not\_null, be\_null, be\_true, contain, **equal**,  
have\_count, match, be\_within, be\_within\_pct, ...

Extended options for refcursor, object  
type, JSON, nested table and varray:

- include(<items>)
- exclude(<items>)
- **unordered**
- join\_by(<items>)

## 24 TEST RUN

```
set serveroutput on size unlimited
exec ut.run('test_suite')
```

List of ...  
schema  
[schema.]**package**[.procedure]  
[schema]:suitepath[.context][.procedure]

```
test_suite
  test_case [.024 sec] (FAILED - 1)
```

Failures:

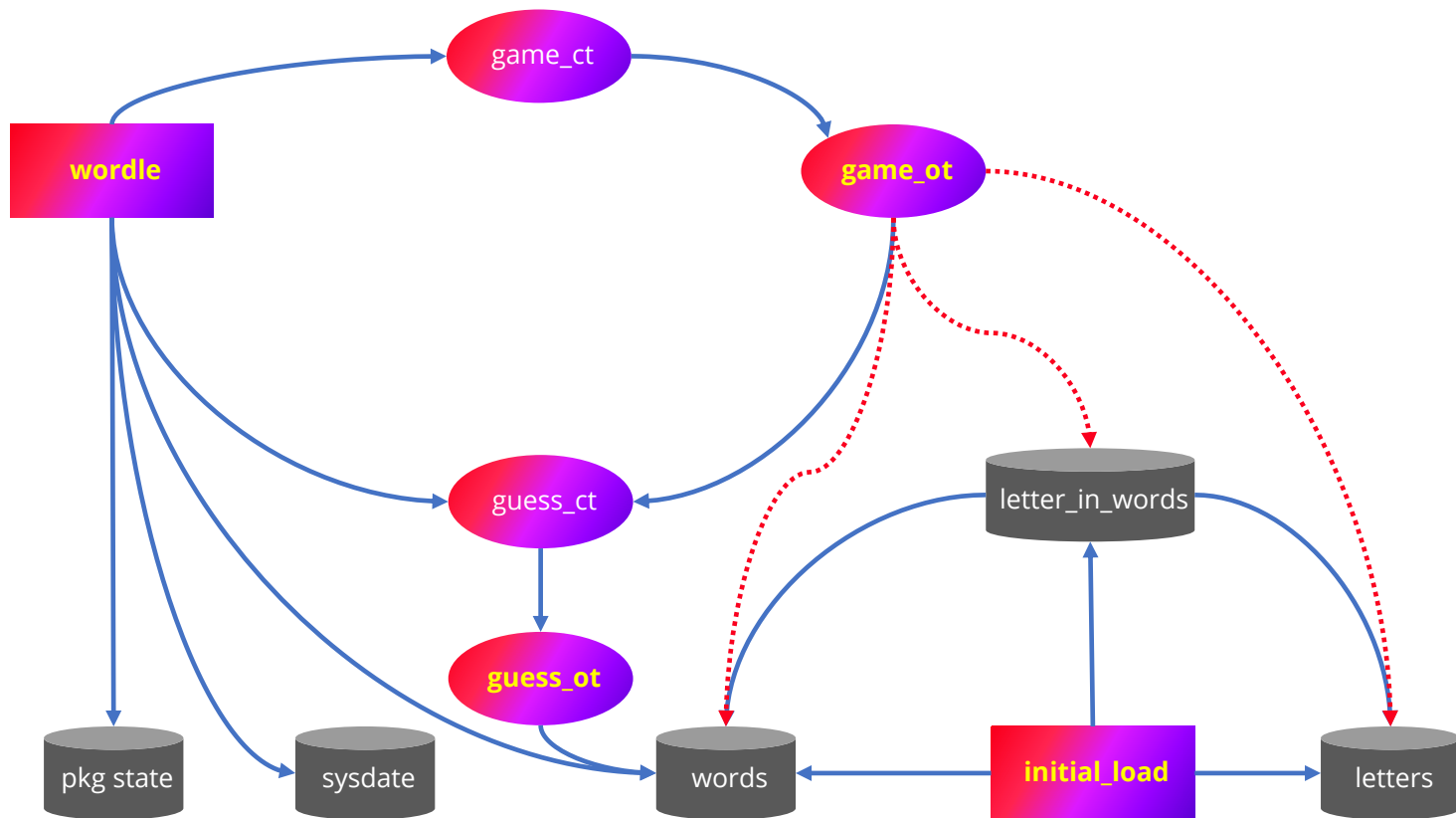
```
1) test_case
  Actual: refcursor [ count = 3 ] was expected to equal: refcursor [ count = 2 ]
  Diff:
  Rows: [ 1 differences ]
  Extra:    <COLUMN_VALUE>-V- .I. (R) .A. -L-</COLUMN_VALUE>
  at "WH.TEST_SUITE.TEST_CASE", line 14 ut.expect(c_actual).to_equal(c_expected).unordered;
```

```
Finished in .026307 seconds
1 tests, 1 failed, 0 errored, 0 disabled, 0 warning(s)
```

Optionally extended by ...  
, a\_tags => 'includeTag, -excludeTag[, ...]'

# HOW TO SIMPLIFY CODE WITH utPLSQL

## 26 BE AWARE OF DEPENDENCIES AND THEIR STATES



## 27 API TEST

```
procedure set_hard_mode is
  c_actual sys_refcursor;
  c_expected sys_refcursor;
begin
  -- act
  wordle.set_hard_mode(true);
  -- assert (solution is banal)
  open c_actual for
    select text
      from (select rownum as row_num, column_value as text
            from wordle.play(201, 'abcde', 'annal', 'noise'))
      where row_num < 7 or row_num = 8;
  open c_expected for
    select 'reduced input due to the following errors:' as text from dual union all
    select '- abcde is not in word list.' from dual union all
    select '- noise does not contain letter A (2 times).' from dual union all
    select '- noise's letter #3 is not a N.' from dual union all
    select '- noise's letter #4 is not a A.' from dual union all
    select '- noise's letter #5 is not a L.' from dual union all
    select '(A) -N- .N. .A. .L.' from dual;
  ut.expect(c_actual).to_equal(c_expected);
end;
```

Relies on  
existing data and  
--%beforeeach

What do we  
want to test?

# DIVIDE & CONQUER

## 29 REDUCE DEPENDENCIES (1)

```
procedure pattern_of_guess_annal_for_solution_banal is
  l_actual varchar2(5 char);
begin
  -- act
  l_actual := util.pattern(in_solution => 'banal', in_guess => 'annal');

  -- assert
  ut.expect(l_actual).to_equal('10222');
end;
```

## 30 REDUCE DEPENDENCIES (2)

```
procedure abcde_is_an_unknown_word is
  o_guess guess_ot;
begin
  -- act
  o_guess := guess_ot(
    in_word          => 'abcde',
    in_solution      => 'zzzzz',
    in_previous_guess => null,
    in_hard_mode     => 0
  );

  -- assert
  ut.expect(o_guess.errors.count).to_equal(1);
  ut.expect(o_guess.errors(1)).to_equal('abcde is not in word list.');
```

end;

## 31 REDUCE DEPENDENCIES (3)

```
procedure abcde_is_a_known_word is
  o_guess guess_ot;
begin
  -- arrange
  insert into words (word) values ('abcde');

  -- act
  o_guess := guess_ot(
    in_word          => 'abcde',
    in_solution      => 'zzzzz',
    in_previous_guess => null,
    in_hard_mode     => 0
  );

  -- assert
  ut.expect(o_guess.errors.count).to_equal(0);
end;
```

## 32 REDUCE DEPENDENCIES (4)

```
procedure revealed_hints_must_be_reused_in_hard_mode is
  o_first_guess  guess_ot;
  o_second_guess guess_ot;
begin
  -- arrange
  o_first_guess := guess_ot(in_word => 'annal', in_solution => 'banal',
                           in_previous_guess => null, in_hard_mode => 1);

  -- act
  o_second_guess := guess_ot(in_word => 'noise', in_solution => 'banal',
                           in_previous_guess => o_first_guess, in_hard_mode => 1);

  -- assert
  ut.expect(anydata.convertcollection(o_second_guess.errors)).to_equal(
    anydata.convertcollection(
      text_ct(
        'noise does not contain letter A (2 times).',
        'noise's letter #3 is not a N.',
        'noise's letter #4 is not a A.',
        'noise's letter #5 is not a L.'
      )
    )
  ).unordered;
end;
```

## 33 API TEST – REFACTORED

```
procedure set_hard_mode is
  l_actual integer;
begin
  -- act
  wordle.set_hard_mode(true);

  -- assert (solution is banal)
  select count(*) into l_actual
    from wordle.play(201, 'annal', 'noise')
    where column_value like '%errors%';
  ut.expect(l_actual).to_be_greater_or_equal(1);
end;
```

## 34 WHAT ABOUT TEST DOUBLES?



- Dummies
- Stubs
- Spies
- Mocks
- Fakes

Requires usually  
a dedicated  
test schema

Missing  
frameworks

Good option for  
3<sup>rd</sup> party system  
access

Source: Heidi Moneymaker, <https://www.instagram.com/p/BIIm4tCBzyl/>

# 35 DEMO

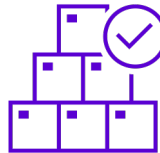
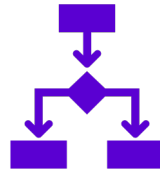
Description	Time [s]
✓ install	21.687
✓ initial_load	21.687
✓ load all tables when empty	5.109
✓ delete data in all tables	6.644
✓ delete and load all tables	9.934
✓ wordle	0.985
✓ internal	0.407
✓ util	
✓ guess_ot	
✓ game_ot	
✓ api	
✓ wordle	
✓ configuration	
✓ enable ANSI console and start play	
✓ reduce suggestions to one without showing query	
✓ show query to retrieve suggestions	
✓ force reuse of known letters in hard mode	
✓ play	
✓ produce XML document with statistics	
✓ play Wordle 213 first attempt	
✓ play Wordle 213 second attempt	
✓ play Wordle 213 third attempt	
✓ play Wordle 213 forth attempt	
✓ play Wordle 213 fifth attempt and solved	
✓ fetatures	
✓ consider wrong positions in suggestions	
✓ consider number of letters in suggestions	
✓ bug fixes	
✓ consider wrong positions in suggestions for repeated letters	
✓ consider occurrences of repeated letters	

```
133 --
134 -- bulkplay
135 --
136
137 procedure bulkplay is
138   l_actual clob;
139   begin
140     -- arrange
141     wordle.set_hard_mode(true);
142
143     -- act
144     l_actual := wordle.bulkplay(in_from_game_id => 867, in_to_game_id => 869).getstringval();
145
146     -- assert
147     ut.expect(l_actual).to_match(
148       a_pattern => '^<bulkplay>.*<solved_games_percent>66.67</.*10 rows only',
149       a_modifiers => 'n'
150     );
151   end bulkplay;
```

# CORE MESSAGES

## 37 BY WRITING A utPLSQL TEST YOU ...

- Describe how to use a piece of software
- Reduce dependencies to make testing
  - simpler
  - faster
  - deterministic
- Simplify code



**TOGETHER WE ARE  
#1 PARTNER FOR BUSINESSES TO  
HARNESS THE POWER OF DATA  
FOR A SMARTER LIFE**