Wordle Solver Challenge in SQL

The challenge is to code the best algorithm to solve all Wordle words with as few guesses as possible, as quickly as possible.

In this repo you will find a few SQL Scripts:

WordleScript.sql

This script will build the following tables:

```
select * from WordleAnswers
select * from WordlePlays
select * from WordleWords
select * from WordleAlgoResults
```

And the following procs:

```
wordle_PlayWord
wordle_GetOptimalNextWord_sample
```

Below are notes on each of the above:

```
select * from WordleAnswers
```

- This contains every word that will be used by Wordle (2,315), but I changed the dates so that it doesn't spoil the actual Wordle game. These dates are randomly assigned, one word per date.

```
select * from WordlePlays
```

- This will contain all your guesses, for every game played. For example, if you took three guesses to land on the correct word for the date below, this would return three rows:
  `select * from WordlePlays where WordDate = '2027-02-13'and Player = 'Alberto'`

```
select * from WordleWords
```

- Contains all the possible answers, as well as some helpful information about each word, like the number of distinct characters and the number of each letter A, B, C… etc. in the word.

```
select * from WordleAlgoResults
```

- When you execute the wordle_PlayWords.sql script (see below), it will store results here.

wordle_PlayAllWords.sql

This is a script that will do the following:

- Loop through all possible rows in WordleAnswers
- Starting with your chosen Starting Word, it will call your algorithm to ask “what word should I guess next”

- Once you solve the word for that date, it will reset to your starting word and begin the process again on the next word until all words are solved.

The challenge:

Write a stored procedure containing your algorithm. The proc should be called:

wordle_GetOptimalNextWord_*YourNameHere*

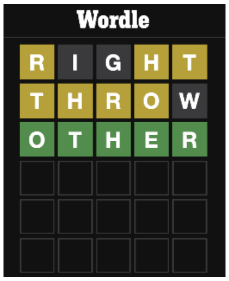
This proc will accept the following parameters:

- `@Player char(25)`
- `@WordDate date`
- `@NextGuess char(5) output`
- `@IsDebugRun bit = 0`

Note that the NextGuess is an output parameter that you will use to return the next word that you would like to play (must be in the table WordleWords).

A sample version of this procedure is included in the wordle_GetOptimalWord_sample. It uses the answers to return the ideal answer every time. Obviously, that is not allowed in your real submission, but it is here so that if you run the wordle_PlayAllWords script without touching it, it will succeed and insert a row into the WordleAlgoResults table.

An example play will give you a better idea how it will work:



Wordle

R I G H T

T H R O W

O T H E R

How to read the above if you have never played Wordle:

1. The first word played is RIGHT and we learned that R, H and T are in the word, but not in those locations (yellow). Also, letters I and G are not in the word at all (gray).
2. We then played THROW in the next word and learned that THRO are in the word, but not in those places. Also W is not in the word.
3. We played OTHER and learned that all of the letters are in their correct spot, so this is the answer (green).

   Below is how that same game would look in the SQL version:

```
declare @next_guess char(5), @l1 bit, @l2 bit, @l3 bit, @l4 bit, @l5 bit

--clear the answers for this date in case we have previously played it
```

```
delete WordlePlays where WordDate = '2027-02-13'and Player = 'Alberto'

--play the word RIGHT
exec wordle_PlayWord 'Alberto', '2027-02-13', 'RIGHT', @L1 output, @L2 output, @L3
output, @L4 output, @L5 output , @IsDebugRun = 1

select @l1 as L1, @l2 as L2, @l3 as L3, @l4 as L4, @l5 as L5

returns the following:
L1      L2      L3      L4      L5
0       NULL    NULL    0       0

Note that NULL, 0, 1 correspond to Gray, Yellow, Green respectively.
```

Now that we have played the word RIGHT, we call your algo proc like this:

```
declare @next_guess char(5)
exec wordle_GetOptimalNextWord_alberto 'Alberto', '2027-02-13', @NextGuess =
@next_guess output, @IsDebugRun = 1
select @next_guess
```

Your proc returns the word: THROW

```
--play the word
exec wordle_PlayWord 'Alberto', '2027-02-13', 'THROW', @L1 output, @L2 output, @L3
output, @L4 output, @L5 output , @IsDebugRun = 1
select @l1 as L1, @l2 as L2, @l3 as L3, @l4 as L4, @l5 as L5
       returns the following:
       L1      L2      L3      L4      L5
       0       0       0       0       NULL

That tells us THRO are in the word, but not in those places.  And there is no W.
```

We call your proc again now that we have played two words, it should know more:

```
declare @next_guess char(5)
exec wordle_GetOptimalNextWord_alberto 'Alberto', '2027-02-13', @NextGuess =
@next_guess output, @IsDebugRun = 1
select @next_guess
```

Your proc now returns this word: OTHER

```
--we play the word
exec wordle_PlayWord 'Alberto', '2027-02-13', 'OTHER', @L1 output, @L2 output, @L3
output, @L4 output, @L5 output , @IsDebugRun = 1
select @l1 as L1, @l2 as L2, @l3 as L3, @l4 as L4, @l5 as L5


returns the following:
L1      L2      L3      L4      L5
1       1       1       1       1
```

Since all characters L1-L5 now return as 1 (green), we know this is the answer and we can move on to the next word to solve.

What specifically do you need to do to submit your entry?

1 – edit the wordle_PlayAllWords.sql script to add your selected start word, replacing the word ‘Hello’ in the sample I provided.

```
	declare @starting_word char(5) = 'HELLO'
```

2 – write your algo proc: wordle_GetOptimalNextWord_yournamehere

3 – edit the wordle_PlayAllWords.sql to call your algo proc in a loop instead of the _sample proc.

4 – message me on LinkedIn with two things: your starting word and a link I can use to download your wordle_GetOptimalNextWord_yournamehere.sql solution (github, dropbox, etc). Your subject should include “Wordle” so I know to read it.

What version of SQL Server?

The solution will be run on SQL Server 2019, so you should try your solution on that version if possible.

How do I know if my solution is good?

If you followed all the rules and your solution averages less than 4 guesses per word and can solve all 2,315 words in under 90 seconds or so (machine power is a factor), that is a solid solution. If your solution is below 3.7 guesses per word, that is very impressive!

Rules:

- **Every word you play, including your starting word, must be in the WordleWords list.**
- **Your solution must implement what Wordle calls “hard mode”. What that means is that in every word you suggest you have to use all of the clues provided by prior words played. For example:**
    - **If your start word is Hello and the H comes back with a 1 (meaning that the answer starts with H, so it is in the right spot), then every word you suggest after that must start with H.**
    - **If your start word is HELLO and the E comes back with a 0 (meaning it is in the word, but not in the second spot), then every word you suggest must have an E in it, but not in the second spot.**
- **Your solution proc can only reference two tables: WordleWords and WordlePlays.**
- **Do not use loops or cursors in your proc.**