**Tables:**

-Puzzle Table:

PuzzleID (PK) int,  
PuzzleWord varchar(50),  
PuzzleSynWords varchar(200),  
IndexesToReveal varchar(50)

-Characters Table:

PuzzleWord varchar (PK)(FK),  
indexNumber int (PK),  
letter char

-Synonym Table:

PuzzleWord varchar (PK),  
SynonymWord varchar

**Game logic association with DB:**

-enter a word for a puzzle:

-Takes input, sends that to new PHP page.  
 -Looks up to see if word is in the puzzle table in the DB.  
 -If it isn't then a new puzzle will need to be built.  
 -Need to iterate through each character and build a new puzzle to the DB  
 -Loop through each character  
 -Using the character table get a list of words that match that character  
 -Select one of the words at random, store it into a word array, then store the index number that's also in the character table in an index array, then store the corresponding  
 clue word to that word into a clue word array (The sql query will join character table and the synonym table).  
 -These arrays will be used to display the puzzle page in some other function.  
 -Using these arrays use display puzzle function, pass the three arrays.  
 -Insert into puzzle table using input word, generated synonym words, and indexes associated with them for future use if puzzle word gets entered again in the future.  
 -If it is in the database then we'll use that one.  
 -First we take the puzzleSynWords column in the puzzle table and separate the values by the commas and store into the words array.  
 -We do the same thing for the IndexesToReveal column and store those numbers into   
 the indexes array.  
 -Then we loop through each of those words and get the clues for each using the synonym table. Store each one into a clue words array.  
 -Send each of the three arrays to the function to display the puzzle page (same function that's used above if the word doesn't exist in the database).

-Guess function

-The puzzle is built, the user has entered in their guesses, then presses the guess button.  
 -PHP post with each row as a string.  
 -Put all of these strings into an array.  
 -Compare each one to corresponding position in puzzle hidden words array (correct answers)   
 -Example simplified pseudocode loop that may be used:  
 $isCorrect = true;  
 for($i = 0; $i < $inputArray's length; $i++){  
 if(guessArray[$i] != answerWords[$i]){  
 $isCorrect = false;  
 break;  
 }  
 }  
 if($isCorrect == true){  
 you win  
 }else{  
 you guess wrong, try again  
 }

Other functions people can think of how to do:  
 -Build puzzle page using the 3 resulting arrays from the first two build puzzle functions I created.