Ken Page

CIS150

Assignment 4

1. 5 data types:
2. NULL – the absence of a value. Not a number, a string or anything else, but used to indicate that there is NO VALUE for the field
3. INTEGER – as is defined, a number with no decimal points, able to be used to maths or comparisons easily
4. REAL – also called floating point. Think of it as an integer with used decimal points. Also usable for maths and comparisons; like a wage 11.25/hr, 11.25 is a REAL value
5. TEXT – is a ‘string’ of text. A text value of ‘123’ cannot be used to perform math functions, but rather reads like the name of a business. Math functions will concatenate: ‘123’ + ‘456’ = ‘123456’
6. BLOB – is data stored exactly as it was provided. Can be images, videos, soundbites, etc

[source](https://www.sqlite.org/datatype3.html)

1. See attached file(s)
2. Three-valued logic is the idea that logical expressions can result in more than just true or false. In SQL this happens because of the native support for a NULL value. The third value, as far as I’ve read, is usually *unknown*, as if something is not TRUE or FALSE, the result is *unknown* until such time as more data is included so as to be able to calculate the expression with values that will result in a TRUE or FALSE.
3. OPERATORS
   1. “+” will add the values on both sides of the operator together (integers will be added, strings will be concatenated)
   2. “-“ will subtract the value from the right side of the operator from the value on the left
   3. “\*” will multiply the values on both sides together
   4. “/” will divide the value on the right side *into* the value on the left
   5. “%” will perform the division operation but return the remainder (will return 0 if there is no remeinder)
   6. “==” checks if the two values given on either side are equal to each other, returning TRUE, FALSE or NULL(unknown)
   7. “=” is the same as “==” for SQL (textbook)
   8. “!=” checks that the two values provided are NOT EQUAL, returning TRUE, FALSE or UNKNOWN
   9. “<>” is used to mathematically check that two values are equal or not, result T, F, N
   10. “>” checks for the left value to be greater than the right, result T, F, N
   11. “<” checks for the left value to be less than the right, result T, F, N
   12. “>=” checks for the left value to be greater than or equal to tie right , result T, F, N
   13. “<=” checks for the left to be less than or equal to the right , result T, F, N
   14. “!<” checks for the left to be NOT less than the right, result T, F, N
   15. “!>” checks for the left to be NOT greater than the right , result T, F, N
   16. “AND” checks for two or more conditions to have a TRUE value
   17. “BETWEEN” looks for values within a set, given the maximum and minimum
   18. “EXISTS” looks for a row in a specified table that meets certain criteria
   19. “IN” compares a value to a list of values given, resulting in a TRUE if found
   20. “NOT IN” looks to ensure a value is NOT found amongst a given list, else it will result in FALSE ([source](https://www.mssqltips.com/sqlservertip/6904/sql-not-in-operator/))
   21. “LIKE” is used to compare values using wildcard operators (\* or % for any combination of characters, ? or \_ for one character at a time) [(source)](https://www.w3schools.com/sql/sql_like.asp)
   22. “GLOB” operates like “LIKE” but is case sensitive
   23. “NOT” is used to reverse the meaning of other operators, i.e. NOT IN, NOT BETWEEN, etc.
   24. “OR” is similar to the “AND” operator but returns a TRUE if *either or any* of the conditions result as TRUE, rather than all of them
   25. “IS NULL” compares a value against a NULL, result T, F, N
   26. “IS” operates the same as “=”
   27. “IS NOT” is the same as “!=”
   28. “||” concatenates two strings together i.e. ‘123’ || ‘456’ = ‘123456’
   29. “UNIQUE” or “SELECT DISTINCT” when used only returns all different values within the search parameters, omitting any repetition of values