Kelsey Berta

CIT 255

Where Clause

To improve the results found from data tables you can use WHERE clauses. The WHERE clauses have many options from strings to numbers and dates to filter. The goal with a where clause is to get the most reliable and efficient results, meaning the most specific. WHERE clauses can include one or more filter conditions. The filters use operators in MySQL. These compare two expressions and include equal '=', less than '<’, greater than ‘>’, and not equal too ‘<>’ ‘! =’. If the data is TRUE, then the result set will include that data.

Along with operators, MySQL also uses conditions. One filter expression is the *and* condition which means all filtering conditions must be true to be in the result set. The *or* condition has results with either of filters being true. If some data does not have either filter included it would be considered false and not show up in the results. If there are three or more filters, parenthesis should be used to help make the query readable for both the person reading the code and the database server. Along with AND and OR there is a NOT operator. This would include anything but what is filtered. For example, WHERE NOT department = art, would include every department, but art.

When creating a condition, you need at least one expression and one operator. The operator including, AND, OR, NOT, =, >, etc. The expression including, numbers, a string, a column in a table, a built in function or a subquery. When creating a filter, you can get many different results. You could be looking for a set of values, specific values, or partial matchings. One type of condition is called equality conditions and the is ‘column = expression’. The next type of condition is inequality conditions which would be, ‘column <> expression’ or ‘column != expression’. The third type of condition is range conditions, these would include symbols such as ‘<’, ‘>’, ‘<=’, ‘>=’. The fourth type of condition is the between condition which would result in a range of values. You could do a range or numbers or dates or even strings and could look like, WHERE customer\_id BETWEEN 20 AND 50, another example would be WHERE first\_name BETWEEN ‘Be’ AND ‘Ce’. Another type of filter you could do is matching. With an example like names this condition is good for strings and matching partials. An example would be WHERE left(last\_name, 2 = ‘E’ this means with the selected columns from this table show the results where the last name has the letter ‘E’ in the second position starting from the left side.

The WHERE clause has many different ways to be used and can be used with a combination of different operators and expressions. The last area to get into is on NULL. NULL is an when there is not a value to be given. NULL can be given to death dates if someone has not died or simply given where there is no data to be found. When something is not applicable in the column a null value can be used. When there is no value as of now, NULL can be used. Something important is something can never equal NULL. Two NULLs are never equal to each other, this is because with the values being unknown, a future value added may be different than the other value that is added in the future. To find the NULL results in the WHERE clause, an IS operator is added. For example, WHERE customer\_id IS NULL; and this would give you a result set customer\_id without a value assigned.