SPRING FRAMEWORK GURU

SQL Beginner to Guru

MySQL Edition

Introduction to the SQL WHERE Clause

The SQL Where Clause

- If no WHERE clause is given, the SQL statement will apply to all records in the table.
  - If you are SELECT-ing data, all rows are returned.
  - If you are UPDATE-ing data, all rows are updated.
  - If you are DELETE-ing data, all rows are deleted.
- The WHERE clause allows us to tell MySQL the criteria we want to use for which rows are returned.
- The WHERE clause is a very versatile clause and important to master.

Where Clause Examples

- Simple WHERE Clause - `SELECT * FROM employees WHERE first_name = ‘Elvis';`
- This statement will return all records in the employees table where the first_name column is equal to Elvis.
  - Must match case EXACTLY - i.e. records with first_name of ‘elvis’ would not be returned.
  - Records with leading and trailing spaces will not be returned.
  - Functions can be used to address matching on case, or remove leading / trailing spaces.

The AND Logical Operator

- The AND operator can be used to add additional criteria.
- With AND all criteria must be satisfied.
- Example: **`SELECT * FROM employees WHERE first_name = 'Elvis' AND last_name = ‘Rossi';`**
- This statement will return all rows from the employee table where the first_name column is Elvis and last_name is Rossi.
- Records matching one, but not the other will NOT be returned.

The OR Logical Operator

- The OR operator can be used to add additional criteria.
- With OR one side of the joined criteria must be satisfied.
- Example: **SELECT * FROM employees WHERE first_name = 'Elvis' OR last_name = ‘Rossi';**
- This statement will return all rows from the employee table where the first_name column is Elvis or where last_name is Rossi.
- All records matching either one will be returned.

WHERE Clause with IN Keyword

- The **IN** keyword allows you to specify a list of values to match on.
- Can be used to reduce the number of OR clauses.
- Example: `SELECT * FROM employees WHERE first_name IN ('Elvis', 'Mary', 'Uri', ‘Moss');`
- The above will return all rows from the database where the first_name column is equal to one of the given values.
- Works like the equals statement - must match case and not have leading or trailing spaces.

WHERE Clause with NOT IN Keyword

- The NOT IN keyword excludes any records being returned where that value is found in that given list.
- Can be used to reduce the number of OR clauses.
- Example: `SELECT * FROM employees WHERE first_name IN ('Elvis', 'Mary', 'Uri', ‘Moss');`
- The above will return all rows from the database where the first_name column is NOT equal to one of the given values.
- Works like the equals statement - must match case and not have leading or trailing spaces.

NULL Values

- A NULL value is where a value has not been assigned to the column.
  - NULL means nothing. A 'space' is still considered a value.
- Keyword IS NOT NULL will return all rows where the column has an assigned value.
- Keyword IS NULL will return all rows where the column DOES NOT have an assigned value.
- Equal operator (=) will return NULL if both values are NULL.
- NULL Safe Equal operator will return 1 (character 1) if both values are NULL.

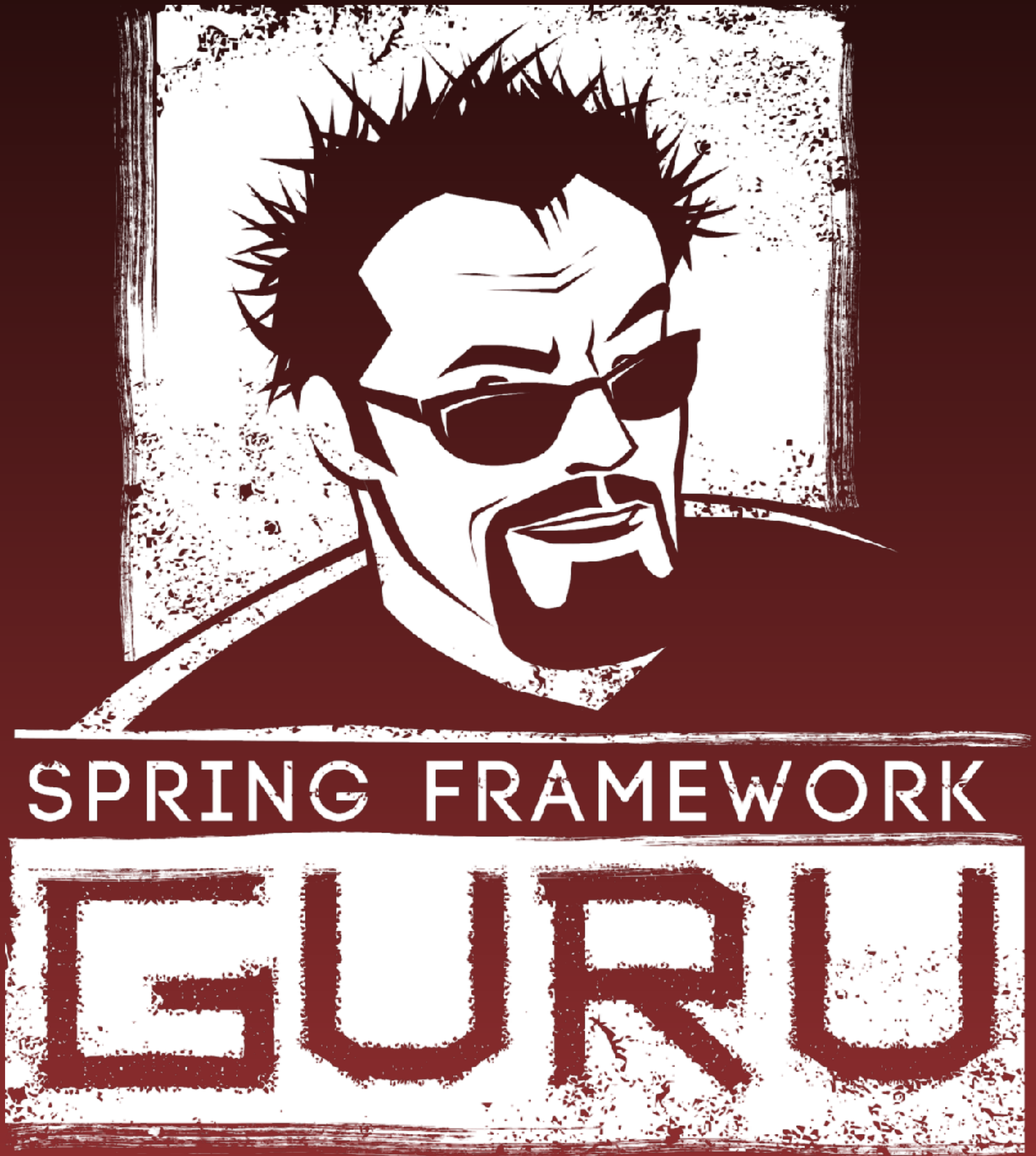
Comparison Operators

Operator	Description
=	**Equals** - Returns true if both values match exactly. Null if both values are NULL.
<=>	**Null Safe Equal Operator** - Like equals operator, but returns 1 if both columns are NULL.
<> or !=	**Not Equal Operator** - Returns true where values are not equal.
>	**Greater Than Operator** - Returns true where left side is greater than right side.
>=	**Greater Than or Equal To Operator** - Returns true where left side is greater than or equal to right side.
<	**Less Than Operator** - Returns true when left side is less than right side.
<=	**Less Than Or Equal To Operator** - Returns true where left side is less than or equal to right side.

Comparison Operators

- Matching Wildcards used in Like / Not Like:
  - % - Wildcard for many characters. ie F% will match any string starting with F
  - _ (underscore) - Wildcard for ONE character. _d% will match on any string with 'd' as second character.

Operator	Description
LIKE	**Like** - Used to match string values using wildcard values.
NOT LIKE	**Not Like** - Used to exclude string values matching wildcard values.
BETWEEN	**Between** - Match records where column is between two given values.
NOT BETWEEN	**Not Between** - Exclude records where column is between two given values.


SPRING FRAMEWORK
GURU