SQL Cheat Sheet: SELECT, WHERE, DISTINCT

This cheat sheet captures the essential commands and concepts covered in Lesson 1B, focusing on `SELECT`, `WHERE`, and `DISTINCT` statements to navigate and retrieve data from databases effectively.

SELECT Statement

- Retrieves data from one or more columns of a table.
- Basic syntax: `SELECT column_name(s) FROM table_name;`
- To select all columns from a table, use the asterisk (``): `SELECT  FROM table_name;`

WHERE Clause

- Filters records that fulfill a specified condition.
- Used to narrow down the results returned from a `SELECT` statement.
- Basic syntax: `SELECT column_name(s) FROM table_name WHERE condition;`
- Conditions can utilize operators such as `=`, `<`, `>`, `<=`, `>=`, `!=`, `BETWEEN`, `LIKE`, and `IN`.

DISTINCT Keyword

- Eliminates duplicate rows from the result set and returns only unique values.
- Particularly useful for counting or listing different values in a column.
- Basic syntax: `SELECT DISTINCT column_name FROM table_name;`

Operators in WHERE Clauses

- Equality: `=` (e.g., `WHERE age = 30`)
- Inequality: `!=` or `<>` (e.g., `WHERE age != 30`)
- Less Than/Greater Than: `<` or `>` (e.g., `WHERE age > 18`)
- Less Than or Equal to/Greater Than or Equal to: `<=` or `>=` (e.g., `WHERE age >= 65`)
- BETWEEN: Used for range (inclusive) (e.g., `WHERE age BETWEEN 18 AND 65`)
- LIKE: Used for pattern matching (e.g., `WHERE first_name LIKE 'Jo%'`)
- IN: Used to specify multiple possible values (e.g., `WHERE age IN (21, 22, 23)`)

Combining Conditions

- AND: Used to combine two or more conditions, all of which must be true (e.g., `WHERE age > 18 AND age < 65`)
- OR: Used when at least one of the conditions must be true (e.g., `WHERE age < 18 OR age > 65`)
- NOT: Used to exclude records that fulfill a condition (e.g., `WHERE NOT age = 30`)

Tips

- VERY IMPORTANT: Single quotes (`' '`) are used in SQL to specify string (text) values. Whenever you are comparing or assigning a value to a column that is of a string data type (such as `VARCHAR`, `CHAR`, `TEXT`, etc.), enclose the value in single quotes.
- Example: `SELECT  FROM movies WHERE title = 'Inception';`
- Numeric values and boolean values do not require single quotes. However, dates and times often do, depending on the database system.
- Incorrect use of single quotes can lead to errors or unexpected behavior in your queries. Always ensure that the data type of the column matches the format of the value you're using.
- If you need to use a single quote within a string value (e.g., for names like O'Brien), you can escape it by typing two single quotes (`''`). For example: `SELECT  FROM authors WHERE last_name = 'O''Brien';`
- Always match the data type of the column to the condition value (e.g., use single quotes around strings).
- Use parentheses `()` to group conditions when combining `AND` and `OR` for clarity and to control the order of evaluation.
- Practice writing queries with different conditions to become comfortable with filtering and selecting data effectively.

Keep this cheat sheet handy as you work through SQL queries involving `SELECT`, `WHERE`, and `DISTINCT` statements to guide your practice and reinforce your learning.