Hive Query Language Manipulation

Hive provides a CLI to write Hive queries using Hive Query Language (HiveQL). Generally HQL syntax is similar to the[SQL](https://www.guru99.com/sql.html)syntax that most data analysts are familiar with.

Hive's SQL-inspired language separates the user from the complexity of Map Reduce programming. It reuses familiar concepts from the relational database world, such as tables, rows, columns and schema, to ease learning.

**Built-in operators**

Hive provides Built-in operators for Data operations to be implemented on the tables present inside Hive warehouse.

These operators are used for mathematical operations on operands, and it will return specific value as per the logic applied.

Types of Built-in Operators in HIVE are:

* Relational Operators
* Arithmetic Operators
* Logical Operators
* Operators on Complex types
* Complex type Constructors

**Relational Operators:**

We use Relational operators for relationship comparisons between two operands.

* Operators such as equals, Not equals, less than, greater than …etc
* The operand types are all number types in these Operators.

|  |  |  |
| --- | --- | --- |
| X = Y | TRUE   if expression X is equivalent to expression Y   Otherwise FALSE. | It takes all primitive types |

**Arithmetic Operators**:

* Arithmetic operations such as addition, subtraction, multiplication and division between operands we use these Operators.
* The operand types all are number types in these Operators

**Sample Example:**

**2 + 3 gives result 5.**

In this example, '+' is theoperator and 2 and 3 are operands. The return value is 5

**Logical Operators:**

We use Logical operators for performing Logical operations on operands

* Logical operations such as AND, OR, NOT between operands we use these Operators.
* The operand types all are BOOLEAN type in these Operators

|  |  |  |
| --- | --- | --- |
| X AND Y | TRUE if both X and Y are TRUE, otherwise FALSE. | Boolean types only |

**Operators on Complex types:**

These are operators which will provide a different mechanism to access elements in complex types.

|  |  |  |
| --- | --- | --- |
| **Operators** | **Operands** | **Description** |
| A[n] | A is an Array and n is an integer type | It will return nth element in the array A. The first element has index of 0 |