

HIVE - BUILT-IN OPERATORS

This chapter explains the built-in operators of Hive. There are four types of operators in Hive:

- Relational Operators
- Arithmetic Operators
- Logical Operators
- Complex Operators

Relational Operators

These operators are used to compare two operands. The following table describes the relational operators available in Hive:

Operator	Operand	Description
A = B	all primitive types	TRUE if expression A is equivalent to expression B otherwise FALSE.
A != B	all primitive types	TRUE if expression A is not equivalent to expression B otherwise FALSE.
A < B	all primitive types	TRUE if expression A is less than expression B otherwise FALSE.
A <= B	all primitive types	TRUE if expression A is less than or equal to expression B otherwise FALSE.
A > B	all primitive types	TRUE if expression A is greater than expression B otherwise FALSE.
A >= B	all primitive types	TRUE if expression A is greater than or equal to expression B otherwise FALSE.
A IS NULL	all types	TRUE if expression A evaluates to NULL otherwise FALSE.
A IS NOT NULL	all types	FALSE if expression A evaluates to NULL otherwise TRUE.
A LIKE B	Strings	TRUE if string pattern A matches to B otherwise FALSE.
A RLIKE B	Strings	NULL if A or B is NULL, TRUE if any substring of A matches the Java regular expression B , otherwise FALSE.
A REGEXP B	Strings	Same as RLIKE.

Example

Let us assume the **employee** table is composed of fields named Id, Name, Salary, Designation, and Dept as shown below. Generate a query to retrieve the employee details whose Id is 1205.

```
+-----+-----+-----+-----+-----+
| Id   | Name           | Salary | Designation           | Dept |
+-----+-----+-----+-----+-----+
|1201 | Gopal          | 45000  | Technical manager    | TP   |
|1202 | Manisha        | 45000  | Proofreader           | PR   |
|1203 | Masthanvali    | 40000  | Technical writer     | TP   |
|1204 | Krian          | 40000  | Hr Admin              | HR   |
|1205 | Kranthi        | 30000  | Op Admin              | Admin|
+-----+-----+-----+-----+-----+
```

The following query is executed to retrieve the employee details using the above table:

```
hive> SELECT * FROM employee WHERE Id=1205;
```

On successful execution of query, you get to see the following response:

```
+-----+-----+-----+-----+-----+
| ID   | Name      | Salary  | Designation          | Dept  |
+-----+-----+-----+-----+-----+
|1205 | Kranthi   | 30000   | Op Admin             | Admin |
+-----+-----+-----+-----+-----+
```

The following query is executed to retrieve the employee details whose salary is more than or equal to Rs 40000.

```
hive> SELECT * FROM employee WHERE Salary>=40000;
```

On successful execution of query, you get to see the following response:

```
+-----+-----+-----+-----+-----+
| ID   | Name      | Salary  | Designation          | Dept  |
+-----+-----+-----+-----+-----+
|1201 | Gopal     | 45000   | Technical manager   | TP    |
|1202 | Manisha   | 45000   | Proofreader         | PR    |
|1203 | Masthanvali | 40000   | Technical writer    | TP    |
|1204 | Krian     | 40000   | Hr Admin            | HR    |
+-----+-----+-----+-----+-----+
```

Arithmetic Operators

These operators support various common arithmetic operations on the operands. All of them return number types. The following table describes the arithmetic operators available in Hive:

Operators	Operand	Description
A + B	all number types	Gives the result of adding A and B.
A - B	all number types	Gives the result of subtracting B from A.
A * B	all number types	Gives the result of multiplying A and B.
A / B	all number types	Gives the result of dividing B from A.
A % B	all number types	Gives the remainder resulting from dividing A by B.
A & B	all number types	Gives the result of bitwise AND of A and B.
A B	all number types	Gives the result of bitwise OR of A and B.
A ^ B	all number types	Gives the result of bitwise XOR of A and B.
~A	all number types	Gives the result of bitwise NOT of A.

Example

The following query adds two numbers, 20 and 30.

```
hive> SELECT 20+30 ADD FROM temp;
```

On successful execution of the query, you get to see the following response:

```
+-----+
|  ADD  |
+-----+
|  50   |
+-----+
```

Logical Operators

The operators are logical expressions. All of them return either TRUE or FALSE.

Operators	Operands	Description
A AND B	boolean	TRUE if both A and B are TRUE, otherwise FALSE.
A && B	boolean	Same as A AND B.
A OR B	boolean	TRUE if either A or B or both are TRUE, otherwise FALSE.
A B	boolean	Same as A OR B.
NOT A	boolean	TRUE if A is FALSE, otherwise FALSE.
!A	boolean	Same as NOT A.

Example

The following query is used to retrieve employee details whose Department is TP and Salary is more than Rs 40000.

```
hive> SELECT * FROM employee WHERE Salary>40000 && Dept=TP;
```

On successful execution of the query, you get to see the following response:

```
+-----+-----+-----+-----+-----+
| ID   | Name       | Salary   | Designation       | Dept   |
+-----+-----+-----+-----+-----+
|1201 | Gopal     | 45000   | Technical manager | TP     |
+-----+-----+-----+-----+-----+
```

Complex Operators

These operators provide an expression to access the elements of Complex Types.

Operator	Operand	Description
A[n]	A is an Array and n is an int	It returns the nth element in the array A. The first element has index 0.
M[key]	M is a Map<K, V> and key has type K	It returns the value corresponding to the key in the map.
S.x	S is a struct	It returns the x field of S.