

OPERATORS IN DBMS

SQL Operators

Operator is a reserved word or a character used primarily in an SQL statement WHERE clause to perform operations, such as comparisons and arithmetic operations. In another word operators are used inside an expression or condition to specify particular operations. SQL operator can be divided into main six categories :

1. Arithmetic Operators
2. Relational Operators
3. Logical Operators
4. Range searching Operators
5. Set Searching Operators
6. Character Operators

Arithmetic Operators:

Arithmetic operators can perform arithmetical operation on numeric operands involved. Arithmetic operators such as addition(+), subtraction(-), multiplication(*) and division(/). The + and - operators can also be used in date arithmetic.

Operator	Description
+	Addition - Adds values on either side of the operator
-	Subtraction - Subtracts right hand operand from left hand operand
*	Multiplication - Multiplies values on either side of the operator
/	Division - Divides left hand operand by right hand operand
%	Modulus - Divides left hand operand by right hand operand and returns remainder

Example:

Run SQL Command Line

```
SQL> Select 5+3 from dual;
```

```
5+3
```

```
-----
```

```
8
```

```
SQL> Select 5*3 from dual;
```

```
5*3
```

```
-----
```

```
15
```

```
SQL> Select 15/2 from dual;
```

```
15/2
```

```
-----
```

```
7.5
```

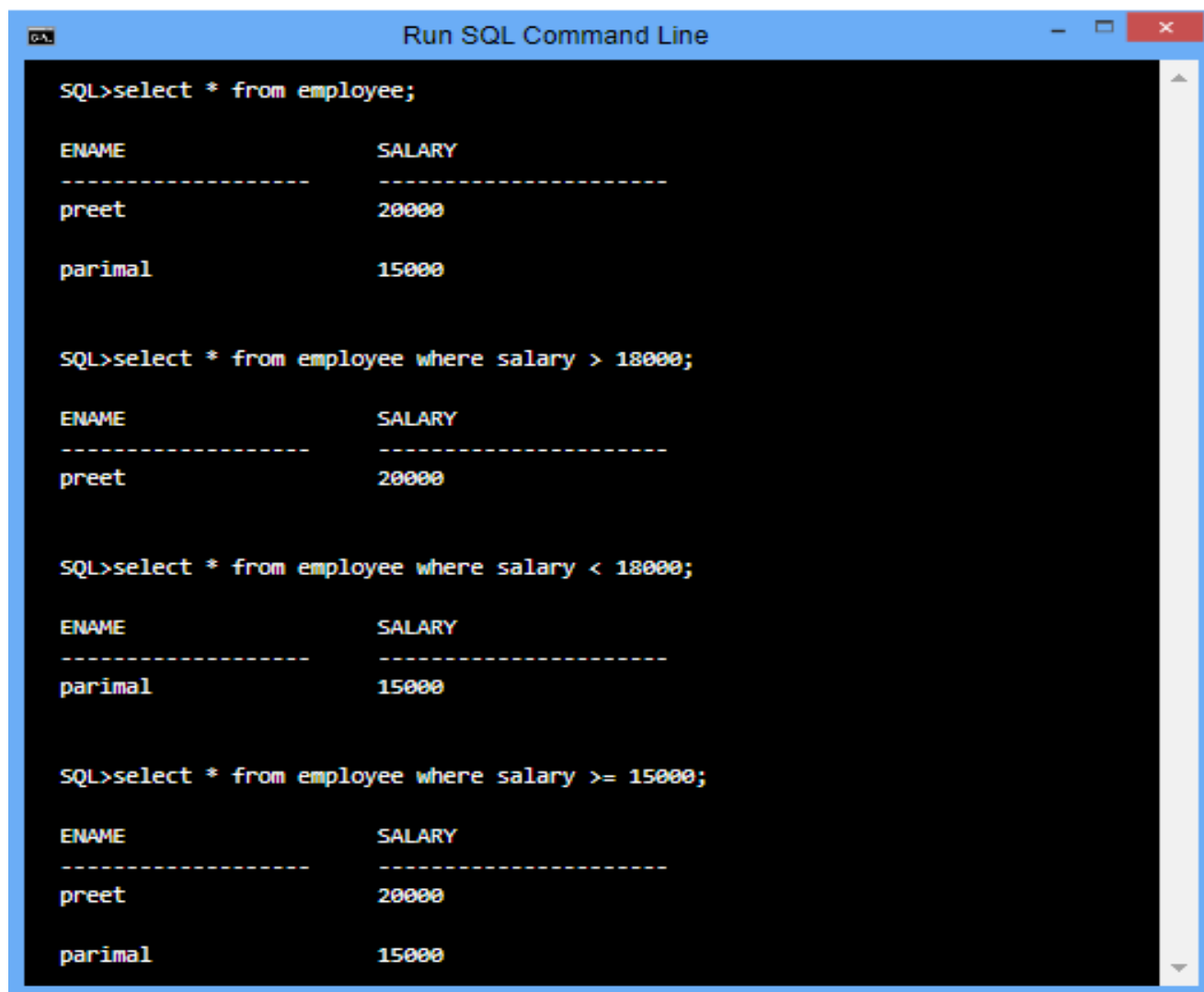
Relational Operators:

Relational operators are used in condition to compare one expression with other. The Relation operators are =, <, >, >=, <=, !=. These relation operators compare expressions and returns

one of three values : True, False, Unknown. An unknown is returned when comparison is performed with a null value. An unknown is treated like as false.

Operator	Description
=	Equals
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
!= or <>	Not Equals

Example:



```
Run SQL Command Line

SQL>select * from employee;

ENAME          SALARY
-----
preet          20000
parimal        15000

SQL>select * from employee where salary > 18000;

ENAME          SALARY
-----
preet          20000

SQL>select * from employee where salary < 18000;

ENAME          SALARY
-----
parimal        15000

SQL>select * from employee where salary >= 15000;

ENAME          SALARY
-----
preet          20000
parimal        15000
```

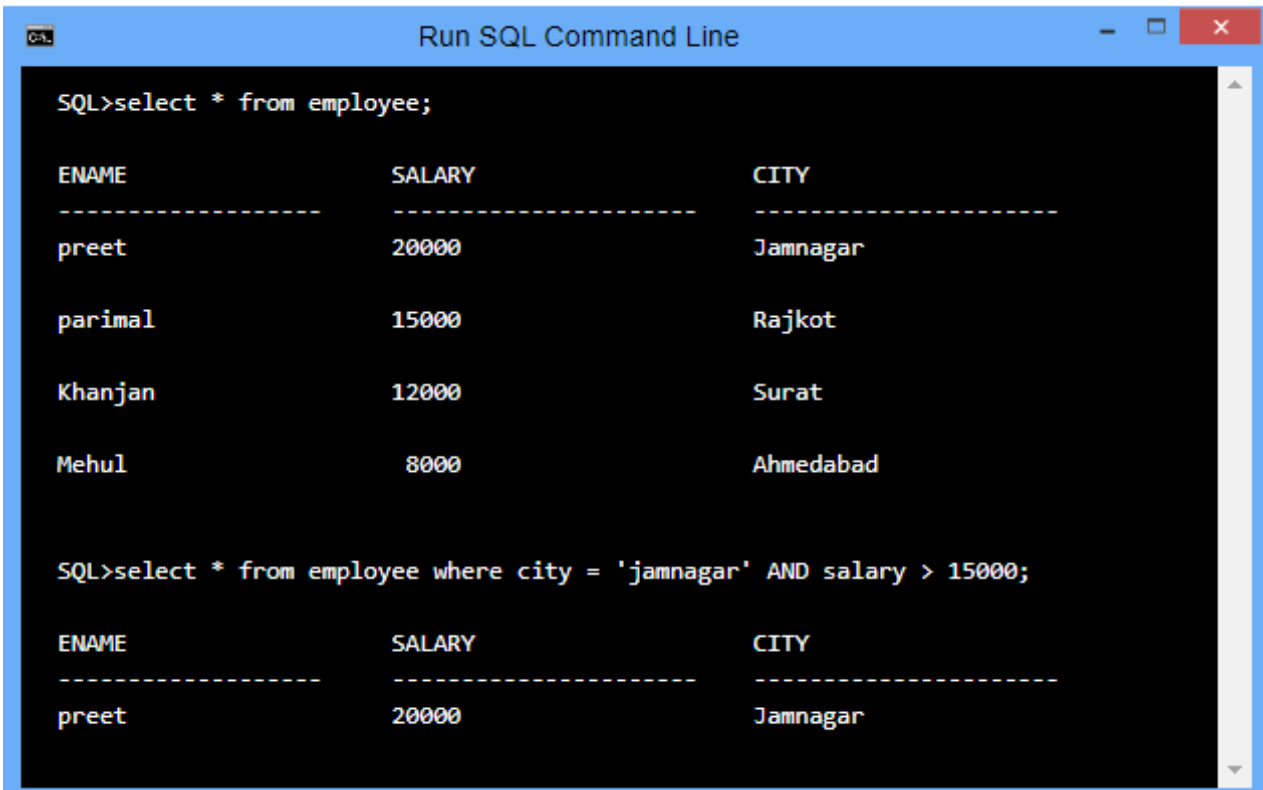
Logical Operators:

SQL provides three logical operators : AND, OR and NOT. These operators are explained below:

(i) AND Operator:

The AND Operator allows creating an SQL statement based on two or more conditions being met. It can be used in any valid SQL statement such as select, insert, or delete.

Example:



```
SQL>select * from employee;
```

ENAME	SALARY	CITY
preet	20000	Jamnagar
parimal	15000	Rajkot
Khanjan	12000	Surat
Mehul	8000	Ahmedabad

```
SQL>select * from employee where city = 'jamnagar' AND salary > 15000;
```

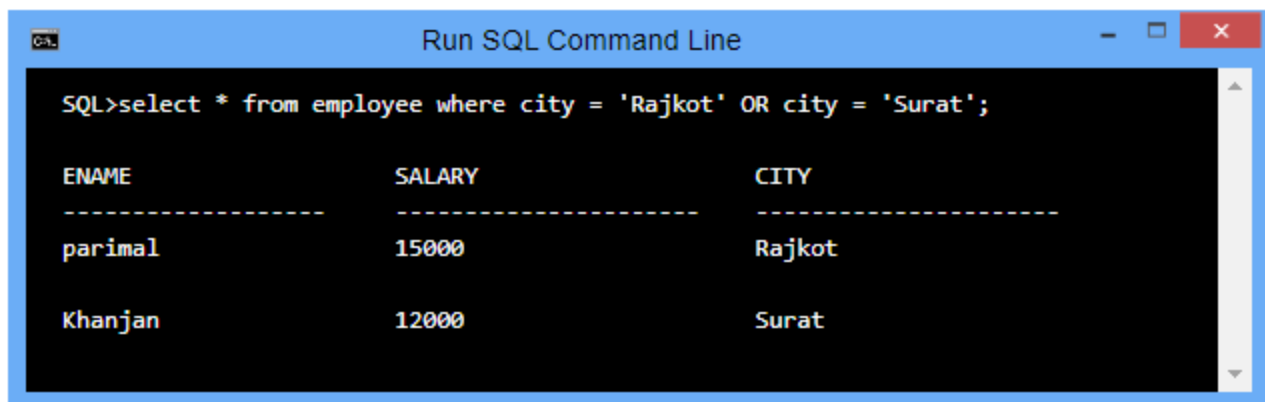
ENAME	SALARY	CITY
preet	20000	Jamnagar

(ii) OR Operator:

The OR condition allows creating an SQL statement where records are returned when any one of the conditions are met

It is used to combine two or more conditions in WHERE and HAVING clause.

Example:



```
SQL>select * from employee where city = 'Rajkot' OR city = 'Surat';
```

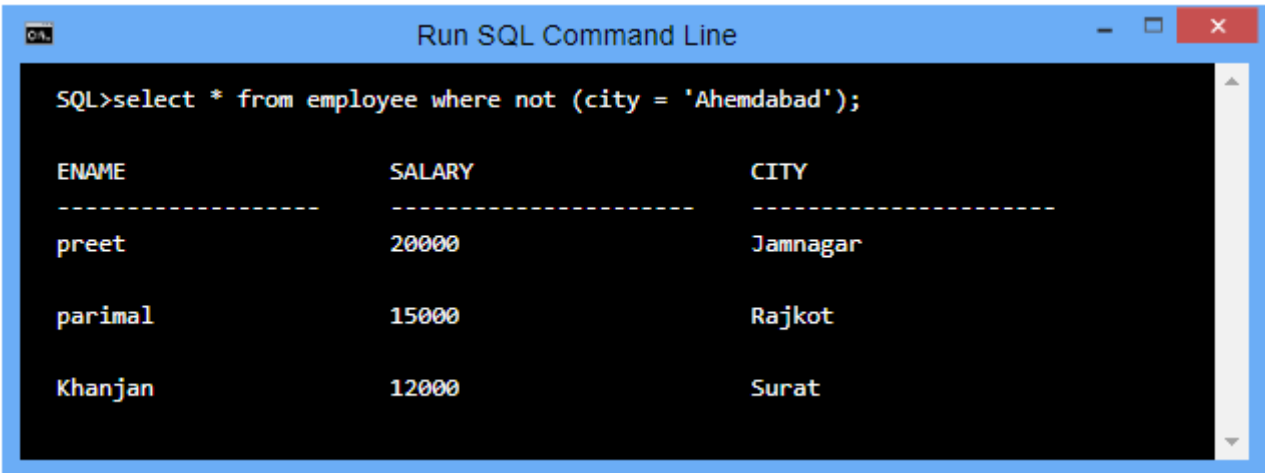
ENAME	SALARY	CITY
parimal	15000	Rajkot
Khanjan	12000	Surat

(iii) NOT Operator:

The Oracle engine will process all rows in a table and display only those records that do not satisfy the condition specified

It is used to negate the result of any condition or group of conditions.

Example:



```
SQL>select * from employee where not (city = 'Ahemdabad');
```

ENAME	SALARY	CITY
preet	20000	Jamnagar
parimal	15000	Rajkot
Khanjan	12000	Surat

Range Searching Operator:

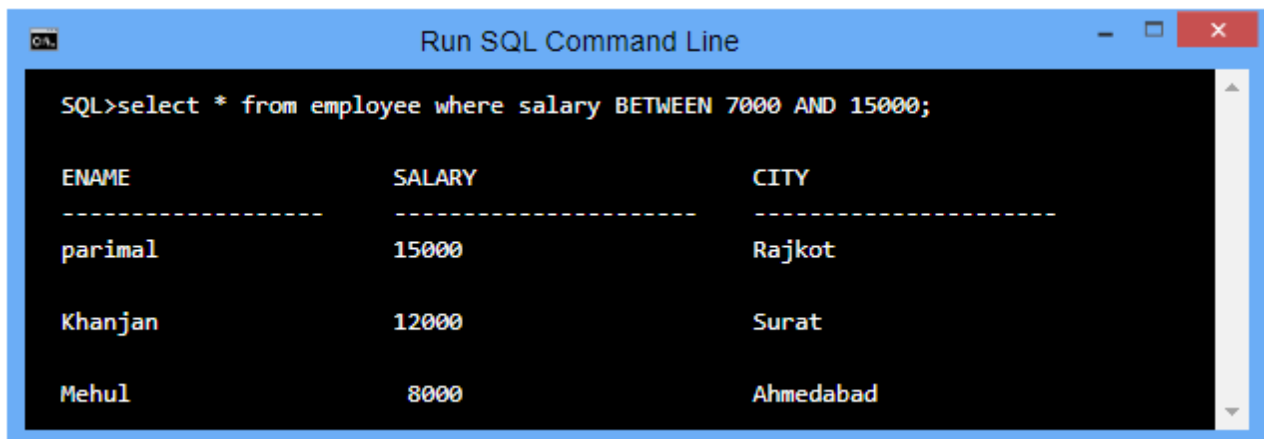
- ❖ Between operator is used to check between two values or specific range.

Syntax:

```
... ColumnName BETWEEN lower limit AND upper limit;
```

- ❖ Selects the rows that contain values within a specified lower and upper limit.
- ❖ The lower and upper limit values must be linked with the keyword AND.
- ❖ The lower and upper limits are inclusive in range.

Example:



```
SQL>select * from employee where salary BETWEEN 7000 AND 15000;
```

ENAME	SALARY	CITY
parimal	15000	Rajkot
Khanjan	12000	Surat
Mehul	8000	Ahmedabad

Set Searching Operator:

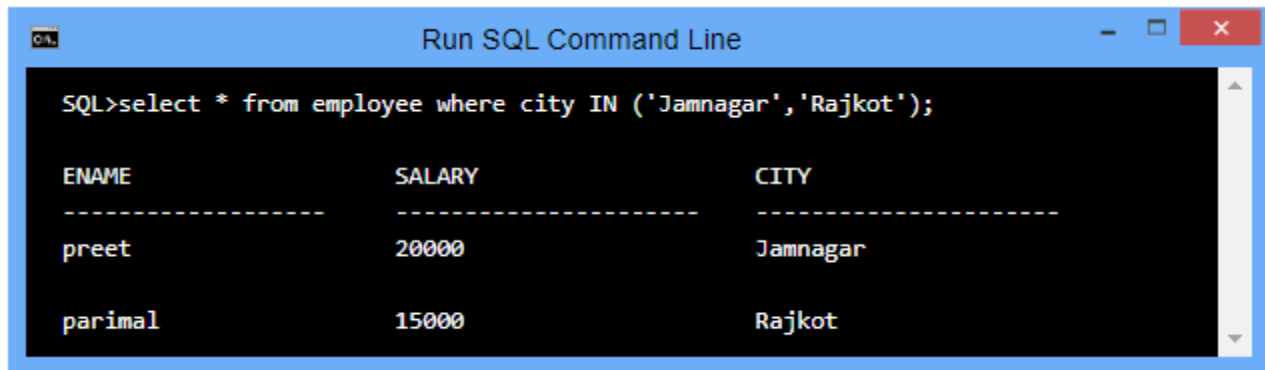
- ❖ The IN operator can be used to select rows that match one of the values in a list.

Syntax:

```
... ColumnName IN ( value1, value2, ..., valueN );
```

- ❖ Select rows that contain any value given in a set.
- ❖ This is similar to '='. But, '=' compares single value to another single value, while IN compares single value to a list (set) of values provided with IN predicate.
- ❖ Can be used when there is a need to use multiple OR conditions.
- ❖ Can be used with numerical, Character as well as date data type.

Example:



```
SQL>select * from employee where city IN ('Jamnagar','Rajkot');
```

ENAME	SALARY	CITY
preet	20000	Jamnagar
parimal	15000	Rajkot

Character Operator:

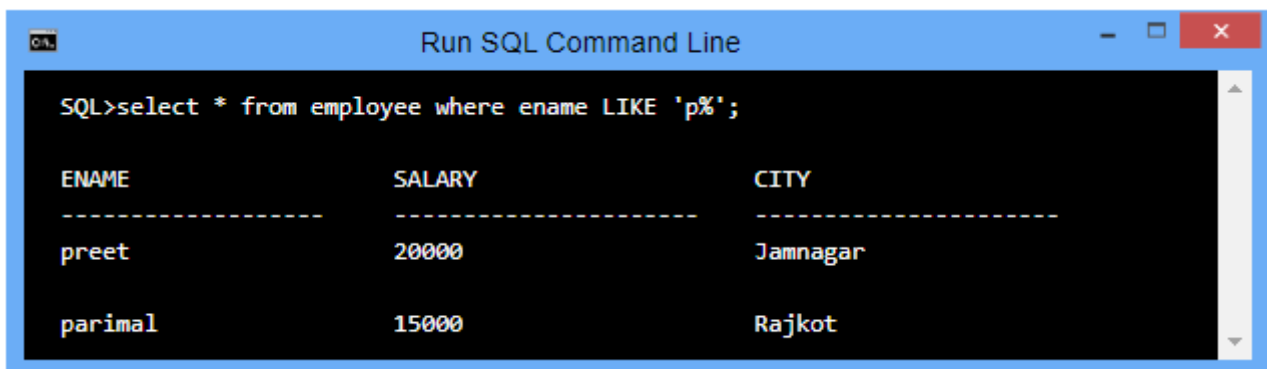
- ❖ Character operators can be used to manipulate the way character strings are represented - both in the output of data as well as in providing conditions. The LIKE operator is used for character operator.

Syntax:

```
... ColumnName like Pattern;
```

- ❖ The like operator is used with special character % and _(underscore).
- ❖ This is similar to '='. BUT, the '=' operator compares for exact matching. While LIKE compares for pattern similarity.

Example:



```
SQL>select * from employee where ename LIKE 'p%';
```

ENAME	SALARY	CITY
preet	20000	Jamnagar
parimal	15000	Rajkot