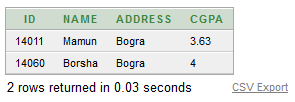
**More SQL Operation and Syntax:**

**Logical operators:** AND, OR, NOT

**AND:** The AND operator allows creating SQL statement based on two or more conditions met. It can be used in any SQL statement like select, insert, update, or delete.

Ex: SELECT \*FROM student WHERE address='Bogra' AND cgpa>3.00;

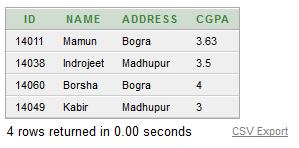
**Output:**



**OR:** The OR condition allows creating SQL statements where records are returned when any one of the conditions are met.

Ex: SELECT \*FROM student WHERE address='Bogra' OR address='Madhupur';

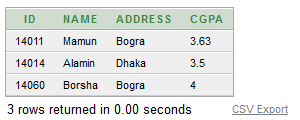
**Output:**



**NOT:** NOT operator processes all row in a table those do not satisfy the conditions.

**Ex:** SELECT \*FROM student WHERE NOT address='Madhupur';

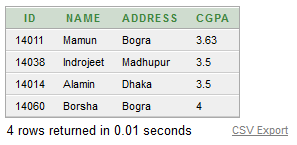
**Output:**



**Range searching:** In order to select data that is within a range of values, the BETWEEN operator is used.

**Ex:** SELECT \*FROM student WHERE cgpa BETWEEN 3.50 AND 4.00;

**Output:**



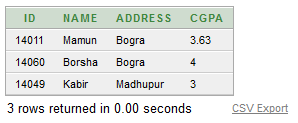
**Pattern Matching:** The LIKE predicate allows comparison of one string value with another string value, which is not identical. This is achieved by using wildcard characters.

% wildcard character allows to match any string of any length.

\_ wildcard character allows to match on a single character.

**Ex:** SELECT \*FROM student WHERE name LIKE 'B%' OR name LIKE '\_a%';

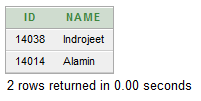
**Output:**



**View selected Columns and selected Rows:** SELECT <ColumnName1>, <ColumnName2> FROM <TableName> WHERE <Condition>;

**Ex:** SELECT id, name FROM student WHERE gpa=3.50;

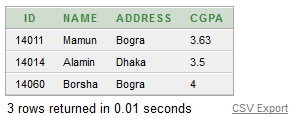
**Output:**



**IN predicate:** IN predicate helps reduce the need of use multiple OR condition.

**Ex:** SELECT \*FROM student WHERE address IN('Bogra','Dhaka');

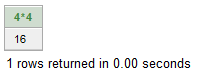
**Output:**



**Table dual:**

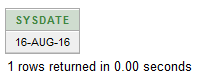
Ex: SELECT 4\*4 FROM dual;

Output:



Ex: SELECT sysdate FROM dual;

**Output:**

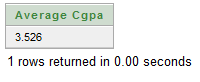


**Aggregate function:**

**AVG:** Returns an average value of ‘n’ ignoring null values in a column.

**EX:** SELECT AVG (cgpa) "Average cgpa" FROM student;

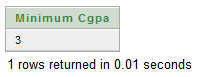
**Output:**



**MIN:** Returns minimum value;

**Ex:** SELECT MIN (cgpa) "Minimum cgpa" FROM student;

**Output:**



**MAX:** Returns maximum value;

**Ex:** SELECT MAX (cgpa) "Minimum cgpa" FROM student;

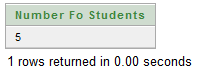
**Output:**



**COUNT:** Returns number of rows where row is not null.

**Ex:** SELECT COUNT(name) "Number fo students" FROM student;

**Output:**



**SUM:** Return summation of value n.

**Ex:** SELECT SUM(cgpa) "Total cgpa" FROM student;

**Output:**



**ABS:** Returns absolute falue of ‘n’.

**Ex:** SELECT ABS(-15) "Absolute of -15" FROM dual;

**Output:**



**POWER(m,n):** Returns m raised to the nth power. n must be integer.

**Ex:** SELECT POWER(2,4) "2^4" FROM dual;

**Output:**

