

Introtallent

Training | Analytics | Consulting

Helping People Decode Analytics for Business

Query
Language

SELECT Statement, LIMIT, DISTINCT

We are going to use country table from world database. This database and table already exists in MySQL.

SELECT: The SELECT statement allows you to get the data from tables or views.

*/*Show all data from country table*/*

```
SELECT * From world.country;
```

Select * extracts all columns from the table

*/*Show only continent and population*/*

```
SELECT Continent, Population From world.country;
```

LIMIT: Limit constrains the number of returned rows.

*/*Show only 10 records from country table*/*

```
SELECT * From world.country LIMIT 10;
```

DISTINCT: Distinct keyword removes duplicate entries and returns the unique values.

*/*Show unique continents from country table*/*

```
SELECT DISTINCT Continent From world.country;
```

Filtering Data (Where Clause, AND, OR, NOT Operator)

WHERE Clause: The WHERE clause is used to filter data based on given criteria

*/*Show all data of Asia continent from country table*/*

```
SELECT * from world.country  
WHERE Continent='Asia';
```

AND Operator: The AND operator displays a record if all the conditions separated by AND are TRUE.

OR Operator: The OR operator displays a record if any of the conditions separated by OR is TRUE.

NOT Operator: The NOT operator displays a record if the condition(s) is NOT TRUE.

*/*Show all data of Asia continent from country table*/*

```
SELECT * from world.country WHERE Continent='Asia' AND LifeExpectancy<60;
```

*/*Show all data of Asia and Africa continent from country table*/*

```
SELECT * from world.country WHERE Continent='Asia' OR Continent='Africa'
```

*/*Show all data except Asia continent from country table*/*

```
SELECT * from world.country WHERE NOT Continent='Asia';
```

IN, NOT IN , BETWEEN, NOT BETWEEN Operator

IN: The IN operator can be used as a replacement of multiple OR conditions.

*/*Show all data of Asia and Africa continent from country table*/*

```
SELECT * from world.country WHERE Continent IN('Asia','Africa'); /*Using IN */
```

```
SELECT * from world.country WHERE Continent='Asia' OR Continent='Africa'; /*Using OR */
```

NOT IN: The NOT IN operator displays all records excluding the ones given in the criteria.

*/*Show all data of without Asia continent from country table*/*

```
SELECT * from world.country WHERE Continent NOT IN ('Asia', 'Africa');
```

BETWEEN: The BETWEEN operator can be used as a replacement of AND conditions. It considers both the values i.e. begin and end in the range.

*/*Show all data of having population between 1 Lac and 2 Lac*/*

```
SELECT * from world.country WHERE Population BETWEEN 100000 AND 200000;
```

NOT BETWEEN: This can be used to eliminate the data for a given range.

*/*Show all data of excluding population range between 1 Lac and 2 Lac*/*

```
SELECT * from world.country WHERE Population NOT BETWEEN 100000 AND 200000;
```

Wildcard Character and LIKE Operator

- A wildcard character is used to substitute any other character(s) in a string.
- Wildcard characters are used with the **LIKE** operator.
- The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

There are two wildcards used in conjunction with the LIKE operator:

- ✓ **Percentage (%)** : The percent sign represents any number of character
- ✓ **Underscore (_)** : The underscore represents a single character. So to represent 3 characters we have to use ____ (3 underscores without space).

*/*Show all data for continents name starting with A*/*

```
SELECT * from world.country WHERE Continent LIKE 'A%';
```

*/*Show all country names starting with I and has 4 character in total*/*

```
SELECT Name from world.country WHERE NAME LIKE 'I_____'; /*It's I followed by 3 underscores*/
```

Note: You must use LIKE with wildcard character. = sign cannot be used with wildcard character.

ORDER BY (for sorting data)

- The ORDER BY keyword is used to sort data in Ascending or Descending order.
- Sorting can be applied on single or multiple columns based on requirement.
- The ORDER BY keyword sorts the records in ascending order by default.
- To sort the records in descending order, use the DESC keyword.

*/*Show all data from country table and sort the data by continent in ascending order*/*

```
SELECT * from world.country ORDER BY Continent ASC;
```

```
SELECT * from world.country ORDER BY Continent; /*By default sorting is Ascending hence it works even without ASC*/
```

*/*Show all data from country table and sort the data by continent in ascending order and population in descending order*/*

```
SELECT * from world.country ORDER BY Continent ASC, Population DESC;
```

*/*Show Continent and Population from country table and sort the data by continent in ascending order and population in descending order*/*

```
SELECT Continent, Population from world.country ORDER BY Continent ASC, Population DESC;
```

*/*Using sequence of column name given in the query i.e 1 for first column, 2 for second and so on...*/*

```
SELECT Continent, Population from world.country ORDER BY 1 ASC, 2 DESC;
```

GROUP BY (for Aggregating data)

The GROUP BY statement can be used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group the result-set by one or more columns.

*/*Show continent wise population*/*

```
SELECT Continent, SUM(Population) from world.country Group BY Continent;
```

*/*Show continent wise population and sort in descending order of population*/*

```
SELECT Continent, SUM(Population) from world.country Group BY Continent ORDER BY Population DESC;
```

*/*Show continent wise and region wise population and sort in ascending order of continent*/*

```
SELECT Continent,Region, SUM(Population) from world.country Group BY Continent, Region ORDER BY Continent;
```

HAVING Clause

- The HAVING clause is used in the SELECT statement to specify filter conditions for a group of rows or aggregates.
- The HAVING clause is often used with the GROUP BY clause to filter groups based on a specified condition. If the GROUP BY clause is omitted, the HAVING clause behaves like the WHERE clause.
- HAVING clause applies a filter condition to each group of rows, while the WHERE clause applies the filter condition to each individual row.

*/*Show continent wise population*/*

/(1)*/*

```
SELECT Continent, SUM(Population) from world.country Group BY Continent  
HAVING Continent IN ('Asia', 'Africa');
```

/(2)*/*

```
SELECT Continent, SUM(Population) from world.country  
WHERE Continent IN ('Asia', 'Africa') Group BY Continent;
```

*/*Show continent wise population and sort in descending order of population (population range between 1 Lac and 2 Lac)*/*

```
SELECT Continent, SUM(Population) from world.country Group BY Continent ORDER BY Population  
HAVING Population BETWEEN 100000 and 200000;
```


ALIAS

There are two kinds of aliases:

- 1) Column alias
- 2) Table alias

Alias for columns:

Sometimes the names of columns are so technical that make the query's output very difficult to understand. To give a column a descriptive name, you use a column alias.

```
SELECT Name as Country_Name, SUM(Population) as Total_Population from world.country;
```

Alias for tables:

You can use an alias to give a table a different name. You assign a table an alias by using the AS keyword as the following syntax:

```
SELECT a.Name , SUM(a.Population) as Total_Population from world.country a;
```

We would love to hear back!

 **80737 99421**
861 856 9998

 **info@introtallent.com**

 **www.introtallent.com**



Office Address:
Introtallent Pvt Ltd.
#12, Anu Arcade, 3rd Floor
CMH Road, Indiranagar,
Bangalore – 560038