give me code for sql Query

Certainly! Here is an example of a SQL query that retrieves data from a table called employees:

SELECT first\_name, last\_name, email

FROM employees

WHERE department = 'Sales'

ORDER BY last\_name;

This query selects the first name, last name, and email of employees who work in the Sales department, ordered by their last names. If you have a specific query or requirement in mind, please let me know!

Convert it to c# code

Sure! Here’s an example of how to execute the SQL query using C# with ADO.NET:

using System;

using System.Data.SqlClient;

class Program

{

static void Main()

{

string connectionString = "your\_connection\_string\_here"; // Replace with your connection string

string query = "SELECT first\_name, last\_name, email FROM employees WHERE department = @department ORDER BY last\_name";

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlCommand command = new SqlCommand(query, connection);

command.Parameters.AddWithValue("@department", "Sales");

try

{

connection.Open();

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

Console.WriteLine($"Name: {reader["first\_name"]} {reader["last\_name"]}, Email: {reader["email"]}");

}

reader.Close();

}

catch (Exception ex)

{

Console.WriteLine($"An error occurred: {ex.Message}");

}

}

}

}

This code establishes a connection to the database, executes the SQL query, and prints the results. Make sure to replace "your\_connection\_string\_here" with the actual connection string for your database.

Convert it to code on vb.net

Certainly! Here’s how you can execute the SQL query using VB.NET:

Imports System.Data.SqlClient

Module Module1

Sub Main()

Dim connectionString As String = "your\_connection\_string\_here" ' Replace with your connection string

Dim query As String = "SELECT first\_name, last\_name, email FROM employees WHERE department = @department ORDER BY last\_name"

Using connection As New SqlConnection(connectionString)

Dim command As New SqlCommand(query, connection)

command.Parameters.AddWithValue("@department", "Sales")

Try

connection.Open()

Dim reader As SqlDataReader = command.ExecuteReader()

While reader.Read()

Console.WriteLine("Name: " & reader("first\_name").ToString() & " " & reader("last\_name").ToString() & ", Email: " & reader("email").ToString())

End While

reader.Close()

Catch ex As Exception

Console.WriteLine("An error occurred: " & ex.Message)

End Try

End Using

End Sub

End Module

This code performs the same functionality as the C# example, connecting to the database, executing the SQL query, and printing the results. Don't forget to replace "your\_connection\_string\_here" with your actual database connection string.

give me 3 funcs on java script

Here are three JavaScript functions demonstrating different functionalities:

**Function to add two numbers:**

`javascript

function add(a, b) {

return a + b;

}

`

**Function to check if a number is even:**

`javascript

function isEven(num) {

return num % 2 === 0;

}

`

**Function to filter an array of numbers greater than a threshold:**

`javascript

function filterGreaterThan(arr, threshold) {

return arr.filter(num => num > threshold);

}

`

These functions cover basic operations such as addition, a conditional check, and array filtering. If you need further examples or specific functionalities, feel free to ask!

Convert it to Node js

Here are the same functions implemented in Node.js. The functions will work the same way as in plain JavaScript but can be executed in a Node.js environment:

**Function to add two numbers:**

`javascript

function add(a, b) {

return a + b;

}

`

**Function to check if a number is even:**

`javascript

function isEven(num) {

return num % 2 === 0;

}

`

**Function to filter an array of numbers greater than a threshold:**

`javascript

function filterGreaterThan(arr, threshold) {

return arr.filter(num => num > threshold);

}

`

To execute these functions in a Node.js application, you could write a small script like this:

// Example usage of the functions

console.log(add(5, 10)); // Output: 15

console.log(isEven(4)); // Output: true

console.log(filterGreaterThan([1, 3, 5, 7, 9], 5)); // Output: [7, 9]

You can run this script using Node.js in your terminal. If you have any specific requirements or further questions, feel free to ask!