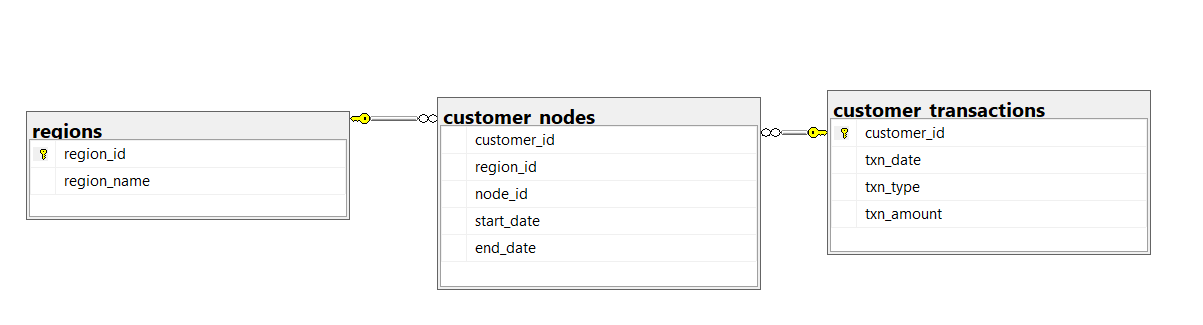
**C# ADO ASSESSMENT**

**CASE STUDY 2**

The basic schema would consist of 3 entities

* Regions
* Customer nodes
* Customer transactions



Driver Code

using DBConnect;

using System.Data.SqlClient;

using System.Text;

using System.Configuration;

class Program:ConfigurationSection

{

public static void Main(string[] args)

{

DBDisconnectedAssign dbd = new DBDisconnectedAssign();

dbd.OpenConn();

dbd.NodesPerRegion();

dbd.CustomerPerRegion();

dbd.AverageAmount();

dbd.ClosingBalance();

dbd.IncreasedClosingBalance();

}

}

Stub Code

1. Display the number of nodes per region

2. Display the number of customers allocated to each region

3. Display the total count and average amount of deposits for all the customers

4. Display the closing balance for each customer at the end of the month

5. Display the number of customers who have increased their closing balance compared to the

previous month.

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DBConnect

{

internal class DBDisconnectedAssign

{

SqlConnection conn;

SqlDataAdapter da;

DataSet ds;

public void OpenConn()

{

string cnnstr = "data source=LAPTOP-BMALL679\\SQLEXPRESS;Initial catalog=DataBank;integrated security=SSPI";

conn = new SqlConnection(cnnstr);

try

{

conn.Open();

Console.WriteLine("Connection Established");

}

catch (SqlException ex)

{

Console.WriteLine("Connection not established");

}

}

string query1 = "select r.region\_name, count( distinct node\_id) node\_counts from customer\_nodes c inner join regions r on c.region\_id=r.region\_id group by r.region\_name";

public void NodesPerRegion()

{

da = new SqlDataAdapter(query1, conn);

ds = new DataSet();

da.Fill(ds, "sd");

//Console.WriteLine("Region Name" + " " + "Node Counts");

Console.WriteLine("Displaying the number of nodes per region");

foreach (DataRow dr in ds.Tables["sd"].Rows)

{

Console.WriteLine(dr[0].ToString() + ":" + dr[1] + " nodes");

}

conn.Close();

}

string query2 = "select r.region\_name, count(distinct c.customer\_id) customer\_counts from customer\_nodes c inner join regions r on c.region\_id=r.region\_id group by r.region\_name";

public void CustomerPerRegion()

{

da = new SqlDataAdapter(query2, conn);

ds = new DataSet();

da.Fill(ds, "sd");

Console.WriteLine("Displaying the number of customers allocated to each region");

foreach (DataRow dr in ds.Tables["sd"].Rows)

{

Console.WriteLine(dr[0].ToString() + ":" + dr[1] + " customers");

}

conn.Close();

}

string query3 = "select count(\*) Total\_count , AVG(txn\_amount) Average\_amount from customer\_transactions where txn\_type='deposit'";

public void AverageAmount()

{

da = new SqlDataAdapter(query3, conn);

ds = new DataSet();

da.Fill(ds, "sd");

Console.WriteLine("Displaying the total count and average amount of deposits for all the customers");

foreach (DataRow dr in ds.Tables["sd"].Rows)

{

Console.WriteLine("Total Count : " + dr[0].ToString() + "\n" +"Average Amount : " + dr[1]);

}

conn.Close();

}

string query4 = "WITH CTE as (SELECT customer\_id, DATEPART(MONTH,txn\_date) as month,SUM(CASE WHEN txn\_type ='deposit' then txn\_amount else 0 end) as deposit, SUM(CASE WHEN txn\_type ='purchase' then -txn\_amount else 0 end) as purchase , SUM(CASE WHEN txn\_type ='withdrawal' then -txn\_amount else 0 end) as withdrawal from customer\_transactions GROUP BY customer\_id,DATEPART(MONTH,txn\_date)),CTE\_2 AS (SELECT customer\_id,month,(deposit +purchase +withdrawal) as total from CTE) SELECT customer\_id, month, SUM(total) OVER (PARTITION BY customer\_id ORDER BY customer\_id,month ROWS BETWEEN UNBOUNDED PRECEDING AND current ROW) AS balance FROM CTE\_2";

public void ClosingBalance()

{

da = new SqlDataAdapter(query4, conn);

ds = new DataSet();

da.Fill(ds, "sd");

Console.WriteLine("Displaying the closing balance for each customer at the end of the month");

Console.WriteLine("Customer\_id, Month, Balance");

foreach (DataRow dr in ds.Tables["sd"].Rows)

{

Console.WriteLine(dr[0] + " " + dr[1] + " " + dr[2]);

}

conn.Close();

}

string query5 = "WITH CTE as (SELECT customer\_id, DATEPART(MONTH,txn\_date) as month, SUM(CASE WHEN txn\_type ='deposit' then txn\_amount else 0 end) as deposit,SUM(CASE WHEN txn\_type ='purchase' then -txn\_amount else 0 end) as purchase ,SUM(CASE WHEN txn\_type ='withdrawal' then -txn\_amount else 0 end) as withdrawal from customer\_transactions GROUP BY customer\_id,DATEPART(MONTH,txn\_date)),CTE\_2 AS (SELECT customer\_id,month,(deposit +purchase +withdrawal) as total from CTE) SELECT count(\*) AS change\_in\_balance FROM CTE\_2";

public void IncreasedClosingBalance()

{

da = new SqlDataAdapter(query5, conn);

ds = new DataSet();

da.Fill(ds, "sd");

Console.WriteLine("Displaying the number of customers who have increased their closing balance compared to the previous month.");

foreach (DataRow dr in ds.Tables["sd"].Rows)

{

Console.WriteLine(dr[0]);

}

conn.Close();

}

}

}

