**Ingenio InterView - Problem Solution Approach**

**Environment :** SQL Server,Visual Studio 2012

1. Created Data Base Ingentio
2. Created Data Table Category which is the DataSet and Input to the Application

CategoryID ParentCategoryID Name KeyWords

100 -1 Business Money

101 100 Accounting Taxes

102 100 Taxation NULL

103 101 Corporate Tax NULL

109 101 Small Business Tax NULL

200 -1 Tutoring Teaching

201 200 Computer NULL

202 201 Operating System NULL

**Problem – 1**

1. GetCategoryInfo by CategoryID,If the Category ID is not there query the record against the ParentCategoryID
2. Create SP which returns the CategoryInfo by passing Category ID as Input Parameter

USE [Ingenio\_DB]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER PROCEDURE [dbo].[usp\_GetCategoryInfoByCategoryID]

(

@CategoryID INT

)

AS BEGIN

;WITH CTE AS (

SELECT CategoryId, ParentCategoryID, Name, Keywords

FROM Category

WHERE CategoryId = @CategoryID

UNION ALL

SELECT c1.CategoryId, c1.ParentCategoryId, c1.Name, c1.Keywords

FROM Category AS c1

INNER JOIN CTE AS c2 ON c1.CategoryId = c2.ParentCategoryId

WHERE c1.Keywords IS NOT NULL

)

SELECT \*

FROM CTE

END

Query the Category table to get the Category Info where given CategoryID

SELECT CategoryId, ParentCategoryID, Name, Keywords

FROM Category

WHERE CategoryId = @CategoryID

If the CategoryID is not there query the Category table against with PArentCategory

SELECT c1.CategoryId, c1.ParentCategoryId, c1.Name, c1.Keywords

FROM Category AS c1

INNER JOIN CTE AS c2 ON c1.CategoryId = c2.ParentCategoryId

**WHERE c1.Keywords IS NOT NULL**

Recursive CTE is used to union CTE you divide it into two sections joined by a  union all where the first section is the category section, and is only called once, while the second section do the recursion and is called repeatedly until where key word from previous level is not NULL

Results

----Input 201

EXEC dbo.usp\_GetCategoryInfoByCategoryID 201

--CategoryId ParentCategoryID Name Keywords

--201 200 Computer NULL

--200 -1 Tutoring Teaching

----Input 202

EXEC dbo.usp\_GetCategoryInfoByCategoryID 202

--Output

--CategoryId ParentCategoryID Name Keywords

--202 201 Operating System NULL

**Problem 2**

GetNameoftheCategoryByCategorylevel by passing Category level

1. Select the Category Root and retrieve the next level category terminate the recursive call if the level is reached

CREATE PROCEDURE dbo.usp\_GetNameoftheCategoryByCategorylevel

(

@catlevel INT

)

AS BEGIN

;WITH CTE AS (

SELECT CategoryId, ParentCategoryId, Name, Keywords, level = 1

FROM Category

WHERE ParentCategoryId = -1

UNION ALL

SELECT c1.CategoryId, c1.ParentCategoryId,

c1.Name, c1.Keywords, level = c2.level + 1

FROM Category AS c1

INNER JOIN CTE AS c2 ON c1.ParentCategoryId = c2.CategoryId

WHERE c2.level < @catlevel

)

SELECT CategoryId

FROM CTE

WHERE level = @catlevel

END

---Input 2

EXEC dbo.usp\_GetNameoftheCategoryByCategorylevel 2

--Output

--CategoryId

--201

--101

--102

**User Interface - Console Application**

**Created two Static Methods to call SP**

public static void GetCategoryInfoByCategoryID(int categoryID)

{

string connetionString = null;

SqlConnection connection;

SqlDataAdapter adapter;

SqlCommand command = new SqlCommand();

SqlParameter param;

DataSet ds = new DataSet();

connetionString = "Data Source=local;Initial Catalog=Ingenio\_DB;User ID=sa;Password=compaq123";

connection = new SqlConnection(connetionString);

connection.Open();

command.Connection = connection;

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "usp\_GetCategoryInfoByCategoryID";

param = new SqlParameter("@CategoryID", categoryID);

param.Direction = ParameterDirection.Input;

param.DbType = DbType.Int32;

command.Parameters.Add(param);

adapter = new SqlDataAdapter(command);

adapter.Fill(ds);

for (int i = 0; i <= ds.Tables[0].Rows.Count - 1; i++)

{

Console.WriteLine(ds.Tables[0].Rows[i][0].ToString());

}

connection.Close();

}

public static void GetNameoftheCategoryByCategorylevel(int categorylevel)

{

string connetionString = null;

SqlConnection connection;

SqlDataAdapter adapter;

SqlCommand command = new SqlCommand();

SqlParameter param;

DataSet ds = new DataSet();

connetionString = "Data Source=local;Initial Catalog=Ingenio\_DB;User ID=sa;Password=compaq123";

connection = new SqlConnection(connetionString);

connection.Open();

command.Connection = connection;

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "usp\_GetNameoftheCategoryByCategorylevel";

param = new SqlParameter("@level", categorylevel);

param.Direction = ParameterDirection.Input;

param.DbType = DbType.Int32;

command.Parameters.Add(param);

adapter = new SqlDataAdapter(command);

adapter.Fill(ds);

for (int i = 0; i <= ds.Tables[0].Rows.Count - 1; i++)

{

Console.WriteLine(ds.Tables[0].Rows[i][0].ToString());

}

connection.Close();

}

**Next level of Implementation**

1. Implement Repository and UnitofWork Pattern.
2. Write Unit Test using Moq Framework