# SQL-Solutions and Services:

**Table: Sol. CatalogServices**

CatalogID | ServiceID, default CatalogOrder| default ServiceOrder, Name(T120), Desc(T512), HelpURL(1024)

; CatalogID start from 101 and ServiceID from 1001

; default ServiceOrder is a Sort Order within Solution or within Services for a solution

; All fields are required

; donot use Autonumber for Sol module

; prefix all objects with ‘Sol.’

**Table: Sol. CatalogServicesRelation**

CatalogID , int

ServiceID, int

ServiceCustomization nVarChar(1024)

**Table: Sol.CatalogServicesOrder ; each user can choose their own order**

UserID, CatalogID | ServiceID, CatalogOrder|ServiceOrder

; ServiceOrder is Order within Solution or within Services for a solution

**Table: Sol.HideCatalogServices**

UserID

CatalogID | ServiceID

**Test Tables: ;Data used for testing**

Table: Test.UserID.ServiceID

UserID

ServiceID

# Methods

GetCatalogs(UserID) ; JSON array

GetServices(UserID, CatalogID) ; JSON array

HideCatalogServices(UserID, CatalogID | ServiceID, HideFlag|UnhideFlag,) ;True to Hide/false to unhide

UpdateCatalogServicesOrder (UserID, CatalogID | ServiceID, New Order)

**Deliverables:**

**DB with Tables and programmable object**

**Test Tables and Test Programmable object**

**Prefix Test tables and SP\_ etc with ‘t’**

**Tips/Keep in view these limits:**

Max Solutions/Catalogs 99

Max Services per Solutions/Catalogs 99

Total Services 9999

User will be presented Catalog array and order or Service array and order in list view for user to manually change the order

Learn to self-Join, inner Join

Minimize Join where possible

Create view if to be frequently used but do create indexes on View and binding.

I will provide data in excel sheets create test data in tables

Develop code so that all methods could be test by passing a one or more coma separated parameters with EXEC SP\_ <coma separated Parameter list>

String parameters will be within single quotes.

Do not preserve transaction for Rollback unless makes sense

Always include Exceptions where needed

Consider implementing Triggers where transactions can be customized; ex: send event to another SP\_ etc.