LSI LASTEM S.r.l.
Via Ex S.P. 161 Dosso, n.9 - 20090 Settala Premenugo (MI) - Italia

Tel.: (+39) 02 95 41 41
Fax: (+39) 02 95 77 05 94
e-mail: info@lsi-lastem.it

WEB: http://www.lsi-lastem.it
CF./P. Iva: (VAT) IT-04407090150
REA: 1009921 **Reg.Imprese:** 04407090150

Gidas

Descrizione del database

Aggiornamento 03/11/2014

Cod. SWUM_00388_it

Sommario

1. Scopo del documento

Gidas è il database in formato Microsoft SQL Server, utilizzato dai programmi LSI LASTEM per memorizzare i dati acquisiti ed elaborati dagli strumenti. Questo documento ne descrive le caratteristiche tecniche.

Per informazioni sull'installazione e manutenzione del database e di SQL Server si faccia riferimento al documento *Gestione del database Gidas*.

2. Descrizione del database

2.1. Schemi

Il database contiene i seguenti schemi:

1. *Core*: contiene gli elementi principali del database (dati e loro strutture);
2. *Viewer*: contiene elementi specifici utilizzati dal programma *GidasViewer*.

2.2. Versione del database

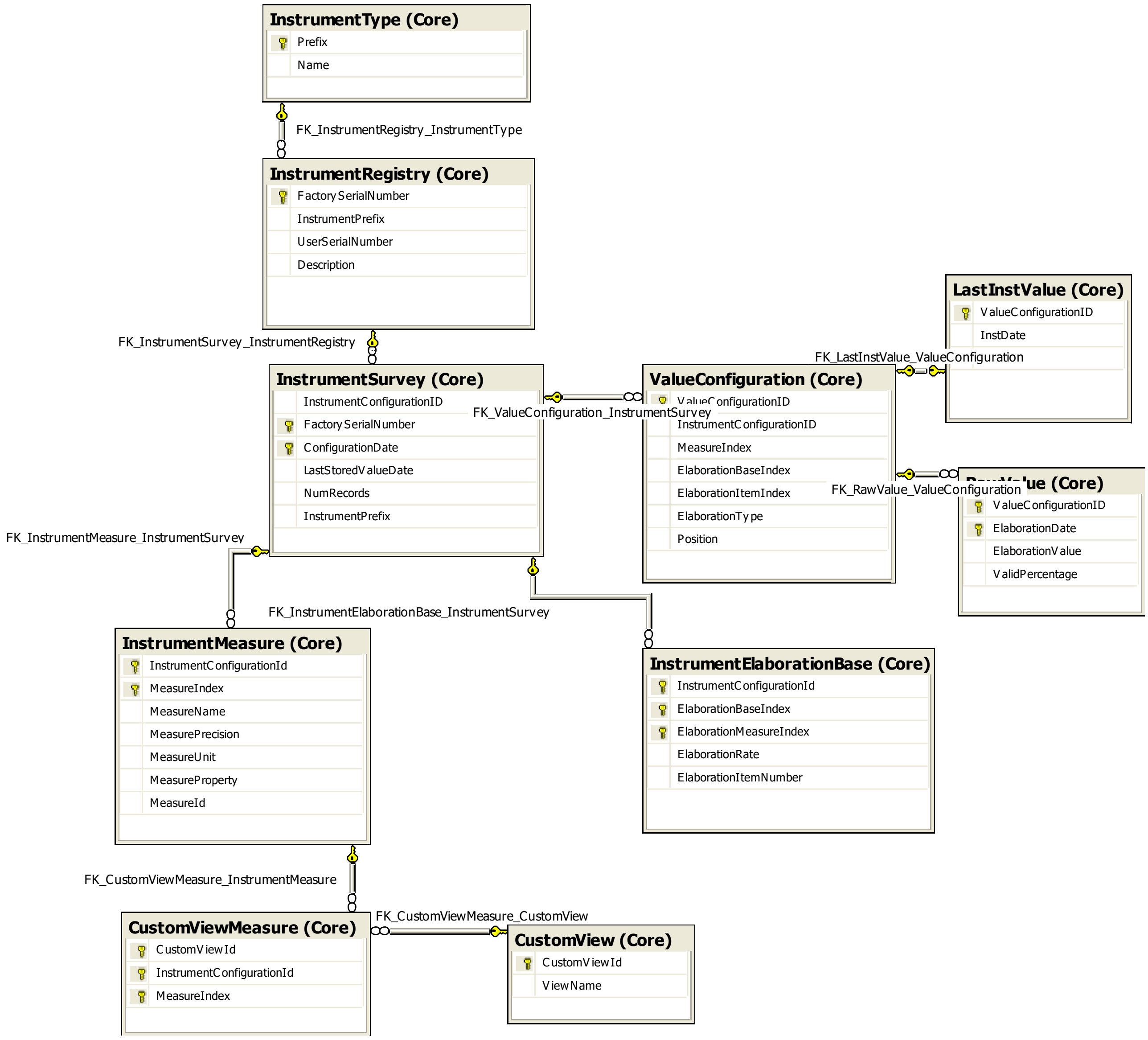
La tabella *Core.GidasVersion* contiene i campi *Version [varchar(15)]* e *ReleaseDate [datetime]* che rappresentano la versione corrente del database.

Per conoscere la versione corrente del database eseguire la stored procedure *Core.spGidasVersion_Get*.

2.3. Supporto per moduli aggiuntivi

Il database supporta l'inserimento di tabelle e procedure legati a moduli aggiuntivi (ad esempio il modulo Synop per la generazione di bollettini meteorologici in formato Synop). I moduli aggiuntivi installati nel database sono contenuti nella tabella *Core.AddIns* (§ ref. 2.5.1.3).
In genere i moduli aggiuntivi sono caratterizzati da tabelle e stored procedure inseriti in un nuovo schema (ad esempio il modulo Synop introduce lo schema *Synop*, il programma GidasViewer lo schema *Viewer*).
Per informazioni sui moduli aggiuntivi si vedano i manuali relativi.

2.4. Diagramma principale delle tabella Dati



Questo diagramma rappresenta il diagramma principale delle tabella *Dati* del database: sono rappresentate le tabelle dei dati e le loro relazioni.

Tutte le relazioni impostano l’aggiornamento e l’eliminazione dei campi collegati in cascata.

2.5. Struttura dei dati memorizzati nel database

2.5.1. Tabelle Dati

Le tabelle *Dati* contengono la struttura di memorizzazione dei dati, comune a tutti gli strumenti LSI LASTEM:

Core.InstrumentType: contiene la descrizione dei tipi di strumento supportati dal database. Campi:

- *Prefix:* identifica il tipo di strumento;
- *Name:* descrizione dello strumento.

Core.InstrumentRegistry: contiene gli strumenti inseriti nel database. Campi:

- *FactorySerialNumber*: numero di matricola dello strumento;
- *InstrumentPrefix:* identifica il tipo di strumento;
- *UserSerialNumber:* numero di matricola eventualmente inserito nello strumento dall'utente;
- *Description:* descrizione dello strumento.
- *DbDescription:* descrizione aggiuntiva inserita dal programma GidasViewer

Core.InstrumenSurvey: contiene la coppia <*matricola - rilievo*>. Per gli strumenti della famiglia E-Log il rilievo coincide con la data di configurazione dello strumento. Campi:

- *InstrumentConfigurationID*: Identity della tabella;
- *FactorySerialNumber*: numero di matricola dello strumento;
- *ConfigurationDate*: data di creazione della configurazione;
- *InstrumentPrefix*: tipo dello strumento;
- *LastStoredValueDate*: data di memorizzazione dell'ultimo dato;
- *NumRecords*: numero di record di dati memorizzati.

Core.InstrumenMeasure: contiene le misure associate a ciascun rilievo. Campi:

- *Core.InstrumentConfigurationID*: identificativo della tabella *Core.InstrumentSurvey* collegata;
- *Core.MeasureIndex*: indice della misura nella configurazione;
- *Core.MeasureName:* nome della misura;
- *Core.MeasurePrecision:* precisione della misura (numero di cifre decimali);
- *Core.MeasureUnit:* stringa che rappresenta l'unità di misura di ogni misura;
- *Core.MeasureProperty:* stringa di interi separati da virgola che rappresentano le proprietà della misura;
- *Core.MeasureId:* intero che rappresenta il tipo della misura.

Core.InstrumentElaborationBase: contiene le basi di elaborazione collegate a ciascuna misura. Per gli strumenti della famiglia E-Log esiste una sola base di elaborazione per ogni misura elaborata. Campi:

- *Core.InstrumentConfigurationID*: identificativo della tabella *Core.InstrumentSurvey* collegata.
- *Core.ElaborationMeasureIndex*: indice della misura collegata;

- *Core.ElaborationBaseIndex*: indice della base di elaborazione;
- *Core.ElaborationRate*: rate di elaborazione espresso in secondi;
- *Core.ElaborationItemNumber*: numero di elaborati associati a questa misura e a questa base di elaborazione.

NOTA
Il campo *Core.ElaborationBaseIndex* per i dati istantanei vale -1 (vedi 2.5.1.1).

Core.ValueConfiguration: contiene le informazioni necessarie per individuare il singolo dato.
Campi:

- *ValueConfigurationID*: identity della tabella;
- *InstrumentConfigurationID:* identifica il record della tabella *Core.InstrumentSurvey* della configurazione di riferimento;
- *MeasureIndex:* identifica la misura;
- *ElaborationBaseIndex:* identifica la base di elaborazione;
- *ElaborationItemIndex:* identifica l'elemento nella maschera di elaborazione (ad es: l'elemento 1 nella maschera di elaborazione *MinMaxAve* è *Max*);
- *ElaborationType:* intero che rappresenta il tipo di elaborato associato al dato (collegato alla tabella *Core.ElabTypeList*);
- *Position:* posizione del dato in un record di dati proveniente dallo strumento.

NOTA
Tutti gli indici sono in base 0. I campi *Core.ElaborationBaseIndex, Core.ElaborationItemIndex Core.Elaborationtype* per i dati istantanei valgono -1 (vedi 2.5.1.1).

Core.RawValue: contiene i dati veri e propri. Campi:

- *ValueConfigurationID:* identifica il tipo di dato nella tabella *Core.ValueConfiguration*;
- *ElaborationDate:* data di memorizzazione del dato elaborato;
- *ElaborationValue:* valore elaborato dallo strumento;
- *ValidPercentage:* percentuale di dati utilizzati per elaborare il valore.

Core.LastInstValue: contiene gli ultimi dati istantanei registrati da ogni strumento. Campi:

- *ValueConfigurationID:* identifica il tipo di dato nella tabella *Core.ValueConfiguration*;
- *InstDate:* data di memorizzazione del dato istantaneo;
- *InstValue:* ultimo valore istantaneo scaricato dallo strumento.

2.5.1.1. Memorizzazione dei dati istantanei

I dati istantanei vengono memorizzati in due tabelle:

- *Core.RawValue*
- *Core.LastInstValue*

Nella tabella *Core.LastInstValue* vengono memorizzati, per ogni misura, solo gli ultimi dati istantanei scaricati dallo strumento. Nella tabella *Core.RawValue* vengono invece memorizzati tutti i dati istantanei scaricati; in questo modo sono disponibili per ulteriori rielaborazioni. I dati istantanei memorizzati nella tabella *Core.RawValue* sono assegnati ad una base di elaborazione che presenta questi valori:

- *Core.InstrumentConfigurationID*: identificativo della tabella *Core.InstrumentSurvey* collegata;
- *Core.ElaborationMeasureIndex*: indice della misura collegata;
- *Core.ElaborationBaseIndex*: -1;
- *Core.ElaborationRate*: 1;
- *Core.ElaborationItemNumber*: 1.

Per identificare questi valori utilizzare come filtro il valore *Core.ElaborationBaseIndex*= -1.

2.5.1.2. CustomView

Una *CustomView* rappresenta una associazione di misure anche appartenenti a rilievi o strumenti diversi. Le CustomView sono definite dagli utenti e sono rappresentate da due tabelle.

Core.CustomView: rappresenta una vista generabile dall'utente. Campi:

- *CustomViewId:* identificativo della tabella;
- *ViewName*: nome associato alla tabella.

Core.CustomViewMeasure: contiene le misure associate ad una vista. Campi:

- *CustomViewId:* identificativo della tabella;
- *InstrumentConfigurationID:* identificativo della coppia strumento – rilievo di provenienza della misura;
- *MeasureIndex:* indice della misura nel rilievo.

2.5.1.3. Altre tabelle dati

Core.ElabTypeList: contiene la lista dei tipi di elaborazione utilizzati dalle misure. Campi:

- *IdElabType:* identificativo della tabella;
- *ElabTypeString:* stringa che identifica il tipo di elaborazione effettuato dallo strumento sul dato;
- *ElabTypeToDo:* stringa che identifica il tipo di rielaborazione semplice applicabile a questo tipo elaborazione.

-1	Inst (n.e.)	Ave
0	Nothing	Nothing
1	Inst	Ave
2	Min	Min
3	Ave	Ave
4	Max	Max
5	StdDev	Nothing

6	Tot	Sum
7	Duration	Sum
8	PrevDir	AvgDirection
9	RisDir	AvgDirection
10	RisVel	Ave
11	StdDevDir	Nothing
12	CalmPerc	Ave
13	ValidDataPerc	Ave
14	PSBisect	AvgDirection
15	PSPrevDir	AvgDirection
16	PSPrevVel	Ave
17	PSStdDevDir	Nothing
18	DirFreq1	Ave
19	DirFreq2	Ave
20	DirFreq3	Ave
21	DirFreq4	Ave
22	DirFreq5	Ave
23	DirFreq6	Ave
24	DirFreq7	Ave
25	DirFreq8	Ave
26	DirFreq9	Ave
27	DirFreq10	Ave
28	DirFreq11	Ave
29	DirFreq12	Ave
30	DirFreq13	Ave
31	DirFreq14	Ave
32	DirFreq15	Ave
33	DirFreq16	Ave

NOTA
Il valore `-1,Inst (n.e.)` rappresenta il valore istantaneo puro mentre il valore `1,Inst` rappresenta il valore istantaneo associato alla elaborazione.
I valori dal 14 compreso sono relativi agli strumenti Babuc ABC.

Core.RegistryLog: contiene informazioni riservate utilizzate internamente

Core.AddIns: contiene la lista degli eventuali moduli aggiuntivi configurati nel database. Campi:

- *AddInCode:* identificativo del modulo aggiuntivo;
- *Description:* descrizione del modulo aggiuntivo;
- *AttachedDate:* data di installazione del modulo aggiuntivo.

2.5.1.4. File di libreria esterni

I valori dei campi *MeasureProperty* e *MeasureId* della tabella *Core.InstrumentMeasure* si trovano in uno dei file di libreria utilizzato dalle applicazione LSI LASTEM. Il file è il seguente:

```
C:\Documents and Settings\All Users\Dati applicazioni\LSI-Lastem\Sltn2\Lib\[Culture]\MeasureAttributes.xml
```

Il contenuto del file è il seguente:

```
<?xml version="1.0" encoding="utf-8"?>
<MeasureAttributes>
  <Culture>it</Culture>
  <MeasureId>
    <Item code="0" text="Non definita" />
    <Item code="1" text="Temperatura" />
    <Item code="2" text="Umidità" />
    <Item code="3" text="Pressione" />
    <Item code="4" text="Angolo" />
    <Item code="5" text="Direzione" />
    <Item code="6" text="Velocità" />
    <Item code="7" text="Radiazione" />
    <Item code="8" text="Precipitazione" />
    <Item code="9" text="Presenza" />
    <Item code="10" text="LivelloMetrico" />
    <Item code="11" text="Illuminamento" />
    <Item code="12" text="FlussoTermico" />
    <Item code="13" text="Concentrazione" />
    <Item code="14" text="LivelloSonoro" />
    <Item code="15" text="Acidita" />
    <Item code="16" text="Spostamento" />
    <Item code="17" text="Conteggio" />
    <Item code="18" text="Asimmetria" />
    <Item code="19" text="Entalpia" />
    <Item code="20" text="Rapporto" />
    <Item code="21" text="Prodotto" />
    <Item code="22" text="Delta" />
    <Item code="23" text="Densita" />
    <Item code="24" text="Integrale" />
    <Item code="25" text="Media" />
    <Item code="26" text="Portata" />
    <Item code="27" text="Eliofania" />
    <Item code="28" text="Percorso" />
    <Item code="29" text="Frequenza" />
    <Item code="30" text="Evaporazione" />
    <Item code="31" text="Resistenza" />
    <Item code="32" text="Tensione" />
    <Item code="33" text="Corrente" />
    <Item code="34" text="WBGT" />
    <Item code="35" text="CET" />
    <Item code="36" text="Conduttanza" />
    <Item code="37" text="IntensitaLuminosa" />
    <Item code="38" text="Intensita" />
    <Item code="39" text="DR" />
    <Item code="40" text="Insoddisfatti" />
    <Item code="41" text="FattoreLuceDiurna" />
    <Item code="42" text="Indice" />
    <Item code="43" text="Livello" />
    <Item code="44" text="Disagio" />
    <Item code="45" text="WindChillIndex" />
    <Item code="46" text="TempChilling" />
    <Item code="47" text="NonSpecificata" />
    <Item code="48" text="Permittività" />
    <Item code="49" text="Albedo" />
    <Item code="50" text="Volume" />
  </MeasureId>
  <MeasureProperty>
```

```
<Item code="0" text="Non Definita" />
<Item code="1" text="Ambiente" />
<Item code="2" text="GiuntoFreddo" />
<Item code="3" text="VentilazioneForzata" />
<Item code="4" text="BulboUmido" />
<Item code="5" text="PuntoDiRugiada" />
<Item code="6" text="MediaRadiante" />
<Item code="7" text="GloboTermometro" />
<Item code="8" text="CorpoNeroPiano" />
<Item code="9" text="Superficiale" />
<Item code="10" text="Fisiologica" />
<Item code="11" text="Parete1" />
<Item code="12" text="Parete2" />
<Item code="13" text="Relativa" />
<Item code="14" text="Assoluta" />
<Item code="15" text="DellAriaUmida" />
<Item code="16" text="DiMiscelazione" />
<Item code="17" text="ParzialeDiVapore" />
<Item code="18" text="ParzialeDiVaporeSaturo" />
<Item code="19" text="Vento" />
<Item code="20" text="DellAria" />
<Item code="21" text="Atmosferica" />
<Item code="22" text="Differenziale" />
<Item code="23" text="Risultante" />
<Item code="24" text="Prevalente" />
<Item code="25" text="Globale" />
<Item code="26" text="Netta" />
<Item code="27" text="Diretta" />
<Item code="28" text="Diffusa" />
<Item code="29" text="Radiante" />
<Item code="30" text="UVA" />
<Item code="31" text="UVB" />
<Item code="32" text="PAR" />
<Item code="33" text="VIR" />
<Item code="34" text="Suolo" />
<Item code="35" text="Cl2" />
<Item code="36" text="CO" />
<Item code="37" text="CO2" />
<Item code="38" text="NH3" />
<Item code="39" text="NO" />
<Item code="40" text="NO2" />
<Item code="41" text="H2" />
<Item code="42" text="H2S" />
<Item code="43" text="HCl" />
<Item code="44" text="HCN" />
<Item code="45" text="O2" />
<Item code="46" text="O3" />
<Item code="47" text="SO2" />
<Item code="48" text="Precipitazione" />
<Item code="49" text="Bagnatura" />
<Item code="50" text="FastA" />
<Item code="51" text="FastC" />
<Item code="52" text="SlowA" />
<Item code="53" text="SlowC" />
<Item code="54" text="Batteria" />
<Item code="55" text="Planare" />
<Item code="56" text="Volume" />
<Item code="57" text="Massa" />
<Item code="58" text="Specifica" />
<Item code="59" text="Integrale" />
<Item code="60" text="Interno" />
<Item code="61" text="Esterno" />
<Item code="62" text="Ricambio" />
```

```
    <Item code="63" text="CorrentiAria" />
    <Item code="64" text="VentilazioneNaturale" />
    <Item code="65" text="Luminosa" />
    <Item code="66" text="Turbolenza" />
    <Item code="67" text="Manganina" />
    <Item code="68" text="Filo" />
    <Item code="69" text="Caviglie" />
    <Item code="70" text="Pavimento" />
    <Item code="71" text="DiffTempVertAria" />
    <Item code="72" text="TempPavimento" />
    <Item code="73" text="TempAsimmRadiante" />
    <Item code="74" text="Parziale" />
    <Item code="75" text="Totale" />
    <Item code="76" text="Impulsiva" />
    <Item code="77" text="DevStand" />
    <Item code="78" text="Calore" />
    <Item code="79" text="UV" />
    <Item code="80" text="EsposizioneUV" />
    <Item code="81" text="CH4" />
    <Item code="82" text="VOC" />
    <Item code="83" text="NMHC" />
    <Item code="84" text="Gas" />
    <Item code="85" text="Non Specificato" />
    <Item code="86" text="Neve" />
    <Item code="87" text="Verticale" />
    <Item code="88" text="Interna" />
    <Item code="89" text="Esterna" />
    <Item code="90" text="Ondacorta" />
    <Item code="91" text="Ondalunga" />
    <Item code="92" text="Riflessa" />
    <Item code="93" text="Incidente" />
  </MeasureProperty>
</MeasureAttributes>
```

Il supporto per gli strumenti Babuc ABC richiede la presenza di un file di libreria utilizzato dalle applicazione LSI LASTEM. Il file è il seguente:

```
C:\Documents and Settings\All Users\Dati applicazioni\LSI-
Lastem\Sltn2\Lib\[Culture]\BabucABCCodOpValues.xml
```

Il contenuto del file è il seguente:

```
<?xml version="1.0" encoding="utf-8"?>
<appSettings>
  <CodOpDataTableManager
type="LSI_Lastem.Lib2.Sltn.InfoGap.CodOpDataTableManager,LSI.Lib2.Sltn">
    <property name="CodOpDataTables">
      <item name="CodOpDataTable">
        <property name="CodOp" value="1" />
        <property name="MeasureId" value="1" />
        <property name="MeasureProperty" value="1" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="2" />
        <property name="MeasureId" value="1" />
        <property name="MeasureProperty" value="4;3" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="3" />
        <property name="MeasureId" value="1" />
        <property name="MeasureProperty" value="1" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="4" />
        <property name="MeasureId" value="1" />
        <property name="MeasureProperty" value="4; 64" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="5" />
        <property name="MeasureId" value="1" />
        <property name="MeasureProperty" value="3" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="6" />
        <property name="MeasureId" value="1" />
        <property name="MeasureProperty" value="4; 3" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="7" />
        <property name="MeasureId" value="1" />
        <property name="MeasureProperty" value="1" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="8" />
        <property name="MeasureId" value="1" />
        <property name="MeasureProperty" value="4; 64" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="9" />
        <property name="MeasureId" value="1" />
        <property name="MeasureProperty" value="1" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="10" />
        <property name="MeasureId" value="1" />
        <property name="MeasureProperty" value="7" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="11" />
        <property name="MeasureId" value="1" />
        <property name="MeasureProperty" value="7" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="12" />
        <property name="MeasureId" value="1" />
        <property name="MeasureProperty" value="8" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="13" />
        <property name="MeasureId" value="1" />
        <property name="MeasureProperty" value="9" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="14" />
        <property name="MeasureId" value="1" />
        <property name="MeasureProperty" value="1" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="15" />
        <property name="MeasureId" value="1" />
        <property name="MeasureProperty" value="10" />
      </item>
      <item name="CodOpDataTable">
```

```
  <property name="CodOp" value="16" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="1" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="17" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="7" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="18" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="7" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="19" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="8" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="20" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="9" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="21" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="1" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="22" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="9" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="23" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="1" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="24" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="9" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="25" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="1" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="26" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="9" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="27" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="1" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="28" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="9" />
```

```
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="29" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="1" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="30" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="9" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="31" />
  <property name="MeasureId" value="2" />
  <property name="MeasureProperty" value="13" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="32" />
  <property name="MeasureId" value="2" />
  <property name="MeasureProperty" value="13" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="33" />
  <property name="MeasureId" value="3" />
  <property name="MeasureProperty" value="21" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="34" />
  <property name="MeasureId" value="5" />
  <property name="MeasureProperty" value="19" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="35" />
  <property name="MeasureId" value="6" />
  <property name="MeasureProperty" value="19" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="36" />
  <property name="MeasureId" value="5" />
  <property name="MeasureProperty" value="19" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="37" />
  <property name="MeasureId" value="3" />
  <property name="MeasureProperty" value="22" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="38" />
  <property name="MeasureId" value="3" />
  <property name="MeasureProperty" value="22" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="39" />
  <property name="MeasureId" value="3" />
  <property name="MeasureProperty" value="22" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="40" />
  <property name="MeasureId" value="6" />
  <property name="MeasureProperty" value="19" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="41" />
  <property name="MeasureId" value="11" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="42" />
  <property name="MeasureId" value="11" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="43" />
  <property name="MeasureId" value="11" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="44" />
  <property name="MeasureId" value="11" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="45" />
  <property name="MeasureId" value="31" />
  <property name="MeasureProperty" value="34" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="46" />
  <property name="MeasureId" value="8" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="47" />
  <property name="MeasureId" value="7" />
  <property name="MeasureProperty" value="25" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="48" />
  <property name="MeasureId" value="7" />
  <property name="MeasureProperty" value="25" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="49" />
  <property name="MeasureId" value="7" />
  <property name="MeasureProperty" value="26" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="50" />
  <property name="MeasureId" value="7" />
  <property name="MeasureProperty" value="27" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="51" />
  <property name="MeasureId" value="7" />
  <property name="MeasureProperty" value="27" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="52" />
  <property name="MeasureId" value="7" />
  <property name="MeasureProperty" value="28" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="53" />
  <property name="MeasureId" value="11" />
  <property name="MeasureProperty" value="0" />
</item>
```

```
<item name="CodOpDataTable">
  <property name="CodOp" value="54" />
  <property name="MeasureId" value="7" />
  <property name="MeasureProperty" value="33" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="55" />
  <property name="MeasureId" value="7" />
  <property name="MeasureProperty" value="30" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="56" />
  <property name="MeasureId" value="7" />
  <property name="MeasureProperty" value="32" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="57" />
  <property name="MeasureId" value="7" />
  <property name="MeasureProperty" value="30" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="58" />
  <property name="MeasureId" value="7" />
  <property name="MeasureProperty" value="30" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="59" />
  <property name="MeasureId" value="7" />
  <property name="MeasureProperty" value="31" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="60" />
  <property name="MeasureId" value="10" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="61" />
  <property name="MeasureId" value="12" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="62" />
  <property name="MeasureId" value="10" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="63" />
  <property name="MeasureId" value="10" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="64" />
  <property name="MeasureId" value="13" />
  <property name="MeasureProperty" value="36" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="65" />
  <property name="MeasureId" value="13" />
  <property name="MeasureProperty" value="39" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="66" />
  <property name="MeasureId" value="13" />
```

```xml
    <property name="MeasureProperty" value="40" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="67" />
    <property name="MeasureId" value="13" />
    <property name="MeasureProperty" value="47" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="68" />
    <property name="MeasureId" value="13" />
    <property name="MeasureProperty" value="38" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="69" />
    <property name="MeasureId" value="13" />
    <property name="MeasureProperty" value="42" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="70" />
    <property name="MeasureId" value="13" />
    <property name="MeasureProperty" value="37" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="71" />
    <property name="MeasureId" value="13" />
    <property name="MeasureProperty" value="43" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="72" />
    <property name="MeasureId" value="13" />
    <property name="MeasureProperty" value="45" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="73" />
    <property name="MeasureId" value="9" />
    <property name="MeasureProperty" value="0" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="74" />
    <property name="MeasureId" value="13" />
    <property name="MeasureProperty" value="37" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="75" />
    <property name="MeasureId" value="13" />
    <property name="MeasureProperty" value="35" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="76" />
    <property name="MeasureId" value="13" />
    <property name="MeasureProperty" value="41" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="77" />
    <property name="MeasureId" value="1" />
    <property name="MeasureProperty" value="1" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="78" />
    <property name="MeasureId" value="9" />
    <property name="MeasureProperty" value="0" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="79" />
    <property name="MeasureId" value="9" />
    <property name="MeasureProperty" value="48" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="80" />
    <property name="MeasureId" value="9" />
    <property name="MeasureProperty" value="49" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="81" />
    <property name="MeasureId" value="3" />
    <property name="MeasureProperty" value="22" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="82" />
    <property name="MeasureId" value="3" />
    <property name="MeasureProperty" value="22" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="83" />
    <property name="MeasureId" value="3" />
    <property name="MeasureProperty" value="22" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="84" />
    <property name="MeasureId" value="13" />
    <property name="MeasureProperty" value="37" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="85" />
    <property name="MeasureId" value="1" />
    <property name="MeasureProperty" value="2" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="86" />
    <property name="MeasureId" value="32" />
    <property name="MeasureProperty" value="0" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="87" />
    <property name="MeasureId" value="32" />
    <property name="MeasureProperty" value="0" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="88" />
    <property name="MeasureId" value="32" />
    <property name="MeasureProperty" value="0" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="89" />
    <property name="MeasureId" value="32" />
    <property name="MeasureProperty" value="0" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="90" />
    <property name="MeasureId" value="32" />
    <property name="MeasureProperty" value="0" />
  </item>
  <item name="CodOpDataTable">
    <property name="CodOp" value="91" />
    <property name="MeasureId" value="32" />
    <property name="MeasureProperty" value="0" />
```

```
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="92" />
  <property name="MeasureId" value="32" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="93" />
  <property name="MeasureId" value="32" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="94" />
  <property name="MeasureId" value="32" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="95" />
  <property name="MeasureId" value="32" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="96" />
  <property name="MeasureId" value="32" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="97" />
  <property name="MeasureId" value="6" />
  <property name="MeasureProperty" value="19" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="98" />
  <property name="MeasureId" value="6" />
  <property name="MeasureProperty" value="20" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="99" />
  <property name="MeasureId" value="8" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="100" />
  <property name="MeasureId" value="29" />
  <property name="MeasureProperty" value="76" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="101" />
  <property name="MeasureId" value="6" />
  <property name="MeasureProperty" value="19" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="102" />
  <property name="MeasureId" value="8" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="103" />
  <property name="MeasureId" value="6" />
  <property name="MeasureProperty" value="20" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="104" />
  <property name="MeasureId" value="6" />
  <property name="MeasureProperty" value="19" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="105" />
  <property name="MeasureId" value="6" />
  <property name="MeasureProperty" value="20" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="106" />
  <property name="MeasureId" value="6" />
  <property name="MeasureProperty" value="20" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="107" />
  <property name="MeasureId" value="6" />
  <property name="MeasureProperty" value="20" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="108" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="5" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="109" />
  <property name="MeasureId" value="4" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="110" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="1" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="111" />
  <property name="MeasureId" value="32" />
  <property name="MeasureProperty" value="54" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="112" />
  <property name="MeasureId" value="31" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="113" />
  <property name="MeasureId" value="15" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="114" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="1" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="115" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="1" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="116" />
  <property name="MeasureId" value="13" />
  <property name="MeasureProperty" value="46" />
</item>
```

```
<item name="CodOpDataTable">
  <property name="CodOp" value="117" />
  <property name="MeasureId" value="16" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="118" />
  <property name="MeasureId" value="14" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="119" />
  <property name="MeasureId" value="14" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="120" />
  <property name="MeasureId" value="14" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="121" />
  <property name="MeasureId" value="14" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="122" />
  <property name="MeasureId" value="14" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="123" />
  <property name="MeasureId" value="14" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="124" />
  <property name="MeasureId" value="14" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="125" />
  <property name="MeasureId" value="14" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="126" />
  <property name="MeasureId" value="14" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="127" />
  <property name="MeasureId" value="14" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="128" />
  <property name="MeasureId" value="14" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="129" />
  <property name="MeasureId" value="14" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="130" />
  <property name="MeasureId" value="3" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="131" />
  <property name="MeasureId" value="6" />
  <property name="MeasureProperty" value="20" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="132" />
  <property name="MeasureId" value="38" />
  <property name="MeasureProperty" value="66" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="133" />
  <property name="MeasureId" value="13" />
  <property name="MeasureProperty" value="81" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="134" />
  <property name="MeasureId" value="13" />
  <property name="MeasureProperty" value="82" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="135" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="1" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="136" />
  <property name="MeasureId" value="15" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="137" />
  <property name="MeasureId" value="47" />
  <property name="MeasureProperty" value="85" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="138" />
  <property name="MeasureId" value="47" />
  <property name="MeasureProperty" value="85" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="139" />
  <property name="MeasureId" value="47" />
  <property name="MeasureProperty" value="85" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="140" />
  <property name="MeasureId" value="47" />
  <property name="MeasureProperty" value="85" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="141" />
  <property name="MeasureId" value="47" />
  <property name="MeasureProperty" value="85" />
</item>
<item name="CodOpDataTable">
```

```
  <property name="CodOp" value="142" />
  <property name="MeasureId" value="47" />
  <property name="MeasureProperty" value="85" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="143" />
  <property name="MeasureId" value="47" />
  <property name="MeasureProperty" value="85" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="144" />
  <property name="MeasureId" value="47" />
  <property name="MeasureProperty" value="85" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="145" />
  <property name="MeasureId" value="47" />
  <property name="MeasureProperty" value="85" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="146" />
  <property name="MeasureId" value="47" />
  <property name="MeasureProperty" value="85" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="147" />
  <property name="MeasureId" value="47" />
  <property name="MeasureProperty" value="85" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="148" />
  <property name="MeasureId" value="47" />
  <property name="MeasureProperty" value="85" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="149" />
  <property name="MeasureId" value="47" />
  <property name="MeasureProperty" value="85" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="150" />
  <property name="MeasureId" value="47" />
  <property name="MeasureProperty" value="85" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="151" />
  <property name="MeasureId" value="2" />
  <property name="MeasureProperty" value="13" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="152" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="5" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="153" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="6" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="154" />
  <property name="MeasureId" value="3" />
  <property name="MeasureProperty" value="17" />
```

```
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="155" />
  <property name="MeasureId" value="18" />
  <property name="MeasureProperty" value="29" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="156" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="29; 55" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="157" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="11" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="158" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="12" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="159" />
  <property name="MeasureId" value="6" />
  <property name="MeasureProperty" value="20" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="160" />
  <property name="MeasureId" value="17" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="161" />
  <property name="MeasureId" value="17" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="162" />
  <property name="MeasureId" value="5" />
  <property name="MeasureProperty" value="19" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="163" />
  <property name="MeasureId" value="27" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="164" />
  <property name="MeasureId" value="1" />
  <property name="MeasureProperty" value="5" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="165" />
  <property name="MeasureId" value="5" />
  <property name="MeasureProperty" value="19" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="166" />
  <property name="MeasureId" value="34" />
  <property name="MeasureProperty" value="60" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="167" />
```

```
      <property name="MeasureId" value="34" />
      <property name="MeasureProperty" value="61" />
    </item>
    <item name="CodOpDataTable">
      <property name="CodOp" value="168" />
      <property name="MeasureId" value="26" />
      <property name="MeasureProperty" value="56" />
    </item>
    <item name="CodOpDataTable">
      <property name="CodOp" value="169" />
      <property name="MeasureId" value="26" />
      <property name="MeasureProperty" value="57" />
    </item>
    <item name="CodOpDataTable">
      <property name="CodOp" value="170" />
      <property name="MeasureId" value="17" />
      <property name="MeasureProperty" value="62; 20" />
    </item>
    <item name="CodOpDataTable">
      <property name="CodOp" value="171" />
      <property name="MeasureId" value="36" />
      <property name="MeasureProperty" value="25" />
    </item>
    <item name="CodOpDataTable">
      <property name="CodOp" value="172" />
      <property name="MeasureId" value="36" />
      <property name="MeasureProperty" value="9; 20; 61" />
    </item>
    <item name="CodOpDataTable">
      <property name="CodOp" value="173" />
      <property name="MeasureId" value="36" />
      <property name="MeasureProperty" value="9; 20; 60" />
    </item>
    <item name="CodOpDataTable">
      <property name="CodOp" value="174" />
      <property name="MeasureId" value="36" />
      <property name="MeasureProperty" value="9; 11" />
    </item>
    <item name="CodOpDataTable">
      <property name="CodOp" value="175" />
      <property name="MeasureId" value="45" />
      <property name="MeasureProperty" value="0" />
    </item>
    <item name="CodOpDataTable">
      <property name="CodOp" value="176" />
      <property name="MeasureId" value="46" />
      <property name="MeasureProperty" value="0" />
    </item>
    <item name="CodOpDataTable">
      <property name="CodOp" value="177" />
      <property name="MeasureId" value="30" />
      <property name="MeasureProperty" value="34" />
    </item>
    <item name="CodOpDataTable">
      <property name="CodOp" value="178" />
      <property name="MeasureId" value="21" />
      <property name="MeasureProperty" value="0" />
    </item>
    <item name="CodOpDataTable">
      <property name="CodOp" value="179" />
      <property name="MeasureId" value="25" />
      <property name="MeasureProperty" value="0" />
    </item>
```

```
<item name="CodOpDataTable">
  <property name="CodOp" value="180" />
  <property name="MeasureId" value="25" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="181" />
  <property name="MeasureId" value="22" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="182" />
  <property name="MeasureId" value="22" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="183" />
  <property name="MeasureId" value="22" />
  <property name="MeasureProperty" value="0" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="184" />
  <property name="MeasureId" value="2" />
  <property name="MeasureProperty" value="14" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="185" />
  <property name="MeasureId" value="2" />
  <property name="MeasureProperty" value="58" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="186" />
  <property name="MeasureId" value="20" />
  <property name="MeasureProperty" value="16" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="187" />
  <property name="MeasureId" value="19" />
  <property name="MeasureProperty" value="15" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="188" />
  <property name="MeasureId" value="28" />
  <property name="MeasureProperty" value="19" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="189" />
  <property name="MeasureId" value="7" />
  <property name="MeasureProperty" value="59" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="190" />
  <property name="MeasureId" value="39" />
  <property name="MeasureProperty" value="63" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="191" />
  <property name="MeasureId" value="40" />
  <property name="MeasureProperty" value="71" />
</item>
<item name="CodOpDataTable">
  <property name="CodOp" value="192" />
  <property name="MeasureId" value="40" />
```

```
        <property name="MeasureProperty" value="72" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="193" />
        <property name="MeasureId" value="40" />
        <property name="MeasureProperty" value="73" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="194" />
        <property name="MeasureId" value="42" />
        <property name="MeasureProperty" value="79" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="195" />
        <property name="MeasureId" value="43" />
        <property name="MeasureProperty" value="80" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="196" />
        <property name="MeasureId" value="42" />
        <property name="MeasureProperty" value="78" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="197" />
        <property name="MeasureId" value="44" />
        <property name="MeasureProperty" value="78" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="198" />
        <property name="MeasureId" value="47" />
        <property name="MeasureProperty" value="85" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="199" />
        <property name="MeasureId" value="47" />
        <property name="MeasureProperty" value="85" />
      </item>
      <item name="CodOpDataTable">
        <property name="CodOp" value="999" />
        <property name="MeasureId" value="0" />
        <property name="MeasureProperty" value="0" />
      </item>
    </property>
    <property name="CodOdDbFile" value="C:\Programmi\File comuni\lsi-
lastem\CodOp\LSICodOp.mdb" />
    <property name="Version">
      <property name="NumVer" value="13" />
      <property name="Data" value="2008-05-26T00:00:00.0000000+02:00" />
    </property>
    <property name="CodOpInstrumentType" value="1" />
  </CodOpDataTableManager>
</appSettings>
```

2.5.2. Tabelle E-Log

Core.ELogConfiguration: contiene le configurazioni E-Log collegate ai dati memorizzati nel database. Campi:

- *InstrumentConfigurationID*: identificativo della configurazione (tabella *Core.InstrumentSurvey*);

- *UpdateRate*: rate di aggiornamento della configurazione;
- *CISSConfiguration*: stringa che contiene la versione in formato CISS (protocollo di comunicazione) della configurazione;
- *FirmwareVersion*: versione del firmware dello strumento.

2.6. Utenti

2.6.1. Database Users

Il database contiene tre utenti con queste caratteristiche:

LSI.Gidas.Administrator:

- ruolo: `db_owner`

LSI.Gidas.Reader:

- ruolo: `db_datareader`
- permesssi sullo schema CORE,VIEWER: SELECT, EXECUTE

LSI.Gidas.Writer:

- ruoli: `db_datareader, db_datawriter`
- permesssi sullo schema CORE,VIEWER: SELECT, EXECUTE, INSERT, ADD, UPDATE

Questi tre utenti sono collegati a tre LOGIN a livello server:

- LSI.Gidas.Reader (password=`redaer_6`)
- LSI.Gidas.Writer (password=`retirw_6`)
- LSI.Gidas.Administrator(password=`sadig_admin`)

L'utente *LSI.Gidas.Administrator* (presente dalla versione 2.2.1) viene utilizzato internamente quando è necessario aggiornare il database.

2.7. Stored procedures

2.7.1. Stored procedures che operano sulla parte dati

Il database contiene numerose *stored procedure* per la gestione dei dati contenuti.
Le stored procedure sono chiamate [*Schema*].sp[*Name*]_[*Action*] dove:

- [*Schema*]: *Core* o *Viewer*;
- [*Name*]: nome della tabella sulla quale opera la Stored Procedure;
- [*Action*]: azione compiuta dalla Stored Procedure. Con ovvio significato le azioni possibili sono:

    1. Add
    2. Remove
    3. Update
    4. Get, GetAll

5. GetBy[*id*] dove id rappresenta il filtro applicato.
6. Refresh

2.8. Viste

- *Core.vInstrumentSurvey_GetRecordCount:* conta il numero reale di record di dati elaborati associati ad ogni *Survey* (il dato contenuto nella tabella *InstrumentSurvey* potrebbe non essere aggiornato);
  ATTENZIONE: utilizzare questa vista con clausole WHERE sul campo InstrumentConfigurationID per selezionare i dati.
- *Core.vInstrumentSurvey_GetMoreRecentID:* estrae il valore *InstrumentConfigurationID* della configurazione più recente di ogni strumento;
- *vValueConfiguration_GetByMoreRecentInstrumentConfigurationID:* estrae gli indici di configurazione di ogni strumento utilizzando la sua configurazione più recente;
- *Core. vInstrumentMeasures_GetAll*: estrae i dati relativi alle misure contenute nei rilievi unificando in una sola tabella le informazioni contenute nelle tabelle *Core.InstrumentSurvey, Core.InstrumentMeasures, Core.ValueConfiguration*.
- *Core.vFlatLastInstData:* estrae tutti i dati presenti nella tabella *Core.LastInstValue* (ultimi dati istantanei della configurazione corrente di ogni strumento), unificando su una sola tabella i tre livelli delle tabelle dati *Core.InstrumentSurvey*, *Core.ValueConfiguration*, *Core.LastInstValue*;
- *Core.vFlatCoreData:* estrae tutti i dati presenti nel database, unificando su una sola tabella i tre livelli delle tabelle dati *Core.InstrumentSurvey*, *Core.ValueConfiguration*, *Core.RawValue*.
  ATTENZIONE: utilizzare questa vista con clausole WHERE per selezionare i dati.

3. Connessione al database

In questo paragrafo vengono descritte le varie stringhe di connessione da utilizzare per connettersi al database Gidas.

3.1. Stringhe di connessione al database

Utilizzo dell'autenticazione di SQL Server per leggere/memorizzare i dati:

```
Data Source=dbserver;
Initial Catalog=Gidas;
Persist Security Info=True;
User ID=LSI.Gidas.Writer;
Password=retirw_6
```

Utilizzo dell'autenticazione di SQL Server per aggiungere/rimuovere tabelle:

```
Data Source=dbserver;
Initial Catalog=Gidas;
Persist Security Info=True;
User ID=LSI.Gidas.Administrator;
Password= sadig_admin
```

Utilizzo dell'autenticazione di Windows:

```
Data Source= dbserver;
Initial Catalog=Gidas;
Integrated Security=True;
Persist Security Info=True
```

4. Memorizzazione dei dati

In questo paragrafo viene descritta la memorizzazione dei dati misurati dagli strumenti E-Log tramite i programmi *CommNetEG* e *3DOM*.

4.1. Esempio: strumento tipo E-Log

Dato uno strumento di matricola *05110008* configurato con due misure (*Temperatura* e *Umidità relativa*) ognuna con le seguenti elaborazioni *Min, Ave, Max, StDev, ValidDataPerc.*, si scaricano i dati elaborati e i dati istantanei. Al primo scaricamento dati vengono aggiunti i seguenti record:

Core.InstrumentRegistry: aggiunto un record

FactorySerialNumber	InstrumentPrefix	UserSerialNumber	Description
05110008	ELog	05110008	Instrument description …

Core.InstrumentSurvey: aggiunto un record

Instrument ConfigurationID	FactorySerial Number	Configuration Date	LastStored ValueDate	NumRecords	Instrument Prefix
95	05110008	02/11/2006	07/05/2008	20568	ELog

10.20.34 11.00.00

Core.InstrumentMeasure: aggiunti due record (due misure)

Instrument ConfigurationID	Measure Index	MeasureName	Measure Precision	Measure Unit	Measure Property	MeasureId
95	1	UmiditaRel	1	%	13	2

Core.InstrumentElaborationBase: aggiunti 4 record, (due basi di elaborazione per ogni misura, dove una base di elaborazione è in realtà associata ai dati istantanei ed è identificabile dal valore *ElaborationBaseIndex*=-1);

Instrument ConfigurationID	Elaboration BaseIndex	Elaboration MeasureIndex	Elaboration Rate	Elaboration ItemNumber
95	-1	0	1	1
95	0	0	1800	5
95	-1	1	1	1
95	0	1	1800	5

Core.ValueConfiguration: aggiunti 12 record (ogni misura contiene cinque valori elaborati *Min, Ave, Max, StDev, ValidDataPerc* ai quali si aggiunge il valore istantaneo identificabile dal valore *ElaborationBaseIndex=ElaborationItemIndex=ElaborationType*=-1. Il campo *Position* per i dati istantanei non è significativo)

Value ConfigurationID	Instrument ConfigurationID	Measure Index	Elaboration BaseIndex	Elaboration ItemIndex	Elaboration Type	Position
3102	95	0	-1	-1	-1	0
3103	95	0	0	0	2	0
3104	95	0	0	1	3	1
3105	95	0	0	2	4	2
3106	95	0	0	3	5	3
3107	95	0	0	4	13	4
3108	95	1	-1	-1	-1	1
3109	95	1	0	0	2	5
3110	95	1	0	1	3	6
3111	95	1	0	2	4	7
3112	95	1	0	3	5	8
3113	95	1	0	4	13	9

Core.RawValue: se vengono scaricati dallo strumento *n* record vengono aggiunti (*n*10 + 2)* record di dati (vengono inseriti nella tabella *n* record per ogni valore di *ValueConfigurationID* inserito nella tabella sopra riportata, ad eccezione dei valori di *ValueConfigurationID* dei dati istantanei che, ad ogni scaricamento di dati, sono sempre uno per misura).

Core.LastInstValue: vengono aggiunti la prima volta due record, negli scaricamenti successivi vengono aggiornati (un valore istantaneo per ogni misura; i valori di *ValueConfigurationID* della tabella corrispondono ai dati istantanei)

Value ConfigurationID	ElaborationDate	ElaborationValue

3102	07/05/2008 12.11.12	24,5
3108	07/05/2008 12.11.12	33,3

Core.ELogConfiguration: aggiunto un record (il campo CISSConfiguration contiene la configurazione dello strumento in formato CISS del protocollo di comunicazione)

Instrument ConfigurationID	UpdateRate	CISSConfiguration	FirmwareVersion
95	1800	….	01.00.02

5. Estrazione dei dati

5.1. Lista delle misure e delle relative elaborazioni associate a un rilievo

Dato uno strumento (nel caso di esempio identificato da *FactorySerialNumber=05110008*) per ottenere la lista di tutte le misure collegate ad un rilievo identificato da *ConfigurationDate*= '`2006-11-02 10:20:34.000`' utilizzare la query seguente:

```
SELECT * FROM Core.vInstrumentMeasures_GetAll
WHERE FactorySerialNumber='05110008' AND ConfigurationDate='2006-11-02T10:20:34'
```

Il risultato è una tabella di questo tipo:

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
05110008	2006-11-02 10:20:34.000	3102	Temperatura	0	-1	1	'C	1	1
05110008	2006-11-02 10:20:34.000	3103	Temperatura	0	0	1	'C	1	1
05110008	2006-11-02 10:20:34.000	3104	Temperatura	0	1	1	'C	1	1
05110008	2006-11-02 10:20:34.000	3105	Temperatura	0	2	1	'C	1	1
05110008	2006-11-02 10:20:34.000	3106	Temperatura	0	3	1	'C	1	1
05110008	2006-11-02 10:20:34.000	3107	Temperatura	0	4	1	'C	1	1
05110008	2006-11-02 10:20:34.000	3108	UmiditaRel	1	-1	1	%	13	2
05110008	2006-11-02 10:20:34.000	3109	UmiditaRel	1	0	1	%	13	2
05110008	2006-11-02 10:20:34.000	3110	UmiditaRel	1	1	1	%	13	2
05110008	2006-11-02 10:20:34.000	3111	UmiditaRel	1	2	1	%	13	2
05110008	2006-11-02 10:20:34.000	3112	UmiditaRel	1	3	1	%	13	2
05110008	2006-11-02 10:20:34.000	3113	UmiditaRel	1	4	1	%	13	2

Dove:

(1): FactorySerialNumber;
(2): ConfigurationDate;
(3): ValueConfigurationId;
(4): MeasureName;
(5): MeasureIndex;
(6): ElaborationItemIndex;
(7): MeasurePrecision;
(8): MeasureUnit;
(9): MeasureProperty;
(10): MeasureId

5.2. Estrazione di dati elaborati

Dato uno strumento (nel caso di esempio identificato da *FactorySerialNumber=05110008*) la selezione di tutti i dati collegati alla configurazione più recente si può ottenere tramite la query seguente:

```
SELECT * FROM Core.vFlatCoreData WHERE InstrumentConfigurationID=
(
      SELECT  maxCfgId FROM Core.vInstrumentSurvey_GetMoreRecentID WHERE
      FactorySerialNumber= '05110008 '
)
```

Il risultato è una tabella di questo tipo:

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
95	05110008	2006-11-02 10:20:34.000	0	0	2	3105	2008-03-14 22:00:00.000	21,19743	100
95	05110008	2006-11-02 10:20:34.000	0	0	3	3106	2008-03-14 22:00:00.000	0,0379049	100
95	05110008	2006-11-02 10:20:34.000	1	0	0	3109	2008-03-14 22:00:00.000	35,93158	100
..	..	..	..	..	..	..	..	..	..

Dove:

(1): InstrumentConfigurationID;
(2): FactorySerialNumber;
(3): ConfigurationDate;
(4): MeasureIndex;
(5): ElaborationBaseIndex;
(6): ElaborationItemIndex;
(7): ValueConfigurationId;
(8): ElaborationDate;
(9): ElaborationValue;
(10): ValidPercentage

Per ottenere i dati di una data configurazione (identificata da *InstrumentConfigurationID=95*) utilizzare la query seguente:

```
      SELECT * FROM Core.vFlatCoreData WHERE InstrumentConfigurationID=95
```

Per limitare temporalmente i dati utilizzare la query seguente:

```
      SELECT * FROM Core.vFlatCoreData WHERE InstrumentConfigurationID=95
      AND Elaborationdate
      BETWEEN '2008-03-14T23:00:00' AND '2008-03-15T23:00:00'
```

5.3. Estrazione di dati istantanei

Per visualizzare gli ultimi dati istantanei di tutti gli strumenti utilizzare la query seguente

```
SELECT * FROM Core.vFlatLastInstData
```

Il risultato è una tabella di questo tipo:

(1)	(2)	(3)	(4)	(5)	(6)	(7)
79	05110000	2006-11-01 10:20:34.000	Temperatura	2954	2008-04-30 10:00:00.000	12,5
80	05110000	2006-11-02 10:20:34.000	Temperatura	2964	2008-04-30 10:00:00.000	12,6
95	05110008	2006-11-02 10:20:34.000	Temperatura	3102	2008-05-07 12:11:12.423	24,5
95	05110008	2006-11-02 10:20:34.000	UmiditaRel	3108	2008-05-07 12:11:12.423	33,3

Dove:

(1): InstrumentConfigurationID;
(2): FactorySerialNumber;
(3): ConfigurationDate;
(4): MeasureName;
(5): ValueConfigurationId;
(6): Date;
(7): Value;

6. Script di installazione

6.1. Creazione database - primo passo

ATTENZIONE:
le istruzioni

```
EXEC dbo.sp_dbcmptlevel @dbname=N'Gidas', @new_cmptlevel=90
GO
```

vanno commentate se si installa il database in SQL Server 2014 o superiore
Modificare la stringa #SIZE# inserendo la dimensione del database (es. 500MB)

Script

```
-----------------------------------------------------------------------------------------------------
--
-- Object:  Create Database [Gidas]
-- Date:       Script Date:
-- Remarks: Database files will be created in the same path of master db
-----------------------------------------------------------------------------------------------------
USE [master]
GO

---------------------------------------------------------------------------------------
-- Create database object in the default storage path
---------------------------------------------------------------------------------------
```

```
DECLARE @name char(5)
DECLARE @logname char(9)
DECLARE @path varchar(500)
DECLARE @dbFile varchar(500)
DECLARE @dbLogFile varchar(500)
SET @name='Gidas'
SET @logname='Gidas_log'
SELECT @path=[filename] FROM sysdatabases WHERE [name]='master'
SET @dbFile= REPLACE(@path,'\master.mdf','\Gidas.mdf')
SET @dbLogFile= REPLACE(@path,'\master.mdf','\Gidas_log.ldf')
DECLARE @queryTx nvarchar(3000)
SET @queryTx=N'CREATE DATABASE [Gidas]  ON PRIMARY (NAME = N'+ QUOTENAME(@name,'''') +
                              ',FILENAME = N' + QUOTENAME(@dbFile,'''')  +
                              ',SIZE = #SIZE#, MAXSIZE=UNLIMITED, FILEGROWTH = 10%) LOG ON ' +
                              '(NAME = N' + QUOTENAME(@logname,'''') +
                              ', FILENAME = N' + QUOTENAME(@dbLogFile,'''') +
                              ',SIZE  =  200MB  ,  MAXSIZE  =  UNLIMITED  ,  FILEGROWTH  =  100MB)  COLLATE
Latin1_General_CI_AS '
EXEC (@queryTx)
GO
------------------------------------------------------------------------------------
-- Alter database to add properties
------------------------------------------------------------------------------------
--EXEC dbo.sp_dbcmptlevel @dbname=N'Gidas', @new_cmptlevel=90
--GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [Gidas].[dbo].[sp_fulltext_database] @action = 'disable'
end
GO
ALTER DATABASE [Gidas] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [Gidas] SET ANSI_NULLS OFF
GO
ALTER DATABASE [Gidas] SET ANSI_PADDING OFF
GO
ALTER DATABASE [Gidas] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [Gidas] SET ARITHABORT OFF
GO
ALTER DATABASE [Gidas] SET AUTO_CLOSE ON
GO
ALTER DATABASE [Gidas] SET AUTO_CREATE_STATISTICS ON
GO
ALTER DATABASE [Gidas] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [Gidas] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [Gidas] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [Gidas] SET CURSOR_DEFAULT  GLOBAL
GO
ALTER DATABASE [Gidas] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [Gidas] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [Gidas] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [Gidas] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [Gidas] SET  DISABLE_BROKER
GO
ALTER DATABASE [Gidas] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [Gidas] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [Gidas] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [Gidas] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [Gidas] SET  READ_WRITE
GO
ALTER DATABASE [Gidas] SET RECOVERY SIMPLE
GO
ALTER DATABASE [Gidas] SET  MULTI_USER
GO
ALTER DATABASE [Gidas] SET PAGE_VERIFY CHECKSUM
GO
```

```
---------------------------------------------------------------------------------------
-- Create database logins
---------------------------------------------------------------------------------------
IF NOT EXISTS (SELECT * FROM syslogins WHERE [name]='LSI.Gidas.Reader')
BEGIN
       CREATE LOGIN [LSI.Gidas.Reader] WITH PASSWORD=N'redaer_6', DEFAULT_DATABASE=[Gidas],
DEFAULT_LANGUAGE=[us_english], CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF
END
GO
IF NOT EXISTS (SELECT * FROM syslogins WHERE [name]='LSI.Gidas.Writer')
BEGIN
       CREATE LOGIN [LSI.Gidas.Writer] WITH PASSWORD=N'retirw_6', DEFAULT_DATABASE=[Gidas],
DEFAULT_LANGUAGE=[us_english], CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF
END
GO
IF NOT EXISTS (SELECT * FROM syslogins WHERE [name]='LSI.Gidas.Administrator')
BEGIN
       CREATE LOGIN [LSI.Gidas.Administrator] WITH PASSWORD=N'sadig_admin',
DEFAULT_DATABASE=[Gidas], DEFAULT_LANGUAGE=[us_english], CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF
END
GO
```

6.2. Creazione database - secondo passo

```
------------------------------------------------------------------------------------------------------
-- Object:  Add objects to database [Gidas]
-- Date: Script Date: 15/11/2012
-- Version:      2.10.0
-- aggiunge la gestione della descrizione in UserConfigurationAttributes
------------------------------------------------------------------------------------------------------
USE [Gidas]

--------------------------------------------------------------------------------------
-- Create Users and Schema
--------------------------------------------------------------------------------------
GO
IF NOT EXISTS (SELECT * FROM sys.schemas WHERE name = N'Viewer')
EXEC sys.sp_executesql N'CREATE SCHEMA [Viewer] AUTHORIZATION [db_owner]'
GO
IF NOT EXISTS (SELECT * FROM ::fn_listextendedproperty(N'Description' , N'SCHEMA',N'Viewer', NULL,NULL,
NULL,NULL))
EXEC sys.sp_addextendedproperty @name=N'Description', @value=N'Schema used by GidasViewer application' ,
@level0type=N'SCHEMA',@level0name=N'Viewer'
GO
IF NOT EXISTS (SELECT * FROM sys.schemas WHERE name = N'Core')
EXEC sys.sp_executesql N'CREATE SCHEMA [Core] AUTHORIZATION [db_owner]'
GO
IF NOT EXISTS (SELECT * FROM ::fn_listextendedproperty(N'Description' , N'SCHEMA',N'Core', NULL,NULL, NULL,NULL))
EXEC sys.sp_addextendedproperty @name=N'Description', @value=N'Schema that contains data' ,
@level0type=N'SCHEMA',@level0name=N'Core'
GO
IF NOT EXISTS (SELECT * FROM sys.database_principals WHERE name = N'LSI.Gidas.Writer')
CREATE USER [LSI.Gidas.Writer] FOR LOGIN [LSI.Gidas.Writer] WITH DEFAULT_SCHEMA=[Core]
GO
IF NOT EXISTS (SELECT * FROM sys.database_principals WHERE name = N'LSI.Gidas.Reader')
CREATE USER [LSI.Gidas.Reader] FOR LOGIN [LSI.Gidas.Reader] WITH DEFAULT_SCHEMA=[Core]
GO
IF NOT EXISTS (SELECT * FROM sys.database_principals WHERE name = N'LSI.Gidas.Administrator')
CREATE USER [LSI.Gidas.Administrator] FOR LOGIN [LSI.Gidas.Administrator] WITH DEFAULT_SCHEMA=[Core]
GO
GRANT EXECUTE, INSERT, DELETE, SELECT, UPDATE ON SCHEMA :: Core TO [LSI.Gidas.Writer]
GO
GRANT EXECUTE, INSERT, DELETE, SELECT, UPDATE ON SCHEMA :: Viewer TO [LSI.Gidas.Writer]
GO
GRANT SELECT ON SCHEMA :: Core TO [LSI.Gidas.Reader]
GO
GRANT SELECT ON SCHEMA :: Viewer TO [LSI.Gidas.Reader]
GO
EXEC sp_addrolemember N'db_owner', N'LSI.Gidas.Administrator'
GO
--------------------------------------------------------------------------------------
-- Create Tables
--------------------------------------------------------------------------------------
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[GidasVersion]') AND type in (N'U'))
BEGIN
CREATE TABLE [Core].[GidasVersion](
        [Version] [varchar](15) NOT NULL,
        [ReleaseDate] [datetime] NOT NULL
) ON [PRIMARY]
END
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[InstrumentType]') AND type in
(N'U'))
BEGIN
CREATE TABLE [Core].[InstrumentType](
        [Prefix] [varchar](15) NOT NULL,
        [Name] [varchar](100) NOT NULL,
 CONSTRAINT [PK_InstrumentType_1] PRIMARY KEY CLUSTERED
(
        [Prefix] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[ELogConfiguration]') AND type in
(N'U'))
BEGIN
CREATE TABLE [Core].[ELogConfiguration](
        [InstrumentConfigurationId] [int] NOT NULL,
        [UpdateRate] [int] NOT NULL,
        [CISSConfiguration] [ntext] NOT NULL,
        [FirmwareVersion] [char](8) NOT NULL CONSTRAINT [DF_ELogConfiguration_FirmwareVersion] DEFAULT
('00.00.00'),
 CONSTRAINT [PK_ELogConfiguration] PRIMARY KEY CLUSTERED
(
        [InstrumentConfigurationId] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
END
GO
IF NOT EXISTS (SELECT * FROM ::fn_listextendedproperty(N'MS_Description' , N'SCHEMA',N'Core',
N'TABLE',N'ELogConfiguration', N'COLUMN',N'UpdateRate'))
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Rate di elaborazione in secondi' ,
@level0type=N'SCHEMA',@level0name=N'Core', @level1type=N'TABLE',@level1name=N'ELogConfiguration',
@level2type=N'COLUMN',@level2name=N'UpdateRate'
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[RegistryLog]') AND type in (N'U'))
BEGIN
CREATE TABLE [Core].[RegistryLog](
        [logId] [bigint] IDENTITY(1,1) NOT NULL,
        [logKey] [nvarchar](50) NOT NULL,
        [logKeyId] [nvarchar](50) NOT NULL,
        [logKeyNumber] [int] NOT NULL,
 CONSTRAINT [PK_RegistryLog] PRIMARY KEY CLUSTERED
(
        [logId] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[CustomView]') AND type in (N'U'))
BEGIN
CREATE TABLE [Core].[CustomView](
        [CustomViewId] [int] IDENTITY(1,1) NOT NULL,
        [ViewName] [nvarchar](50) NOT NULL,
 CONSTRAINT [PK_CustomView] PRIMARY KEY CLUSTERED
(
        [CustomViewId] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
```

```
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Viewer].[CustomFilter]') AND type in
(N'U'))
BEGIN
CREATE TABLE [Viewer].[CustomFilter](
        [CustomFilterId] [int] IDENTITY(1,1) NOT NULL,
        [ParentType] [int] NOT NULL,
        [ParentId] [int] NOT NULL,
        [FilterConfiguration] [ntext] NOT NULL,
 CONSTRAINT [PK_CustomFilter] PRIMARY KEY CLUSTERED
(
        [CustomFilterId] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
END
GO
IF NOT EXISTS (SELECT * FROM ::fn_listextendedproperty(N'MS_Description' , N'SCHEMA',N'Viewer',
N'TABLE',N'CustomFilter', N'COLUMN',N'ParentType'))
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'0-> Survey; 1 -> CustomView' ,
@level0type=N'SCHEMA',@level0name=N'Viewer', @level1type=N'TABLE',@level1name=N'CustomFilter',
@level2type=N'COLUMN',@level2name=N'ParentType'
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[ElabTypeList]') AND type in (N'U'))
BEGIN
CREATE TABLE [Core].[ElabTypeList](
        [IdElabType] [int] NOT NULL,
        [ElabTypeString] [nvarchar](15) NOT NULL,
        [ElabTypeToDo] [varchar](15) NULL,
 CONSTRAINT [PK_ElabTypeList] PRIMARY KEY CLUSTERED
(
        [IdElabType] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[InstrumentSurvey]') AND type in
(N'U'))
BEGIN
CREATE TABLE [Core].[InstrumentSurvey](
        [InstrumentConfigurationID] [int] IDENTITY(1,1) NOT NULL,
        [FactorySerialNumber] [varchar](20) NOT NULL,
        [ConfigurationDate] [datetime] NOT NULL,
        [LastStoredValueDate] [datetime] NOT NULL,
        [NumRecords] [bigint] NOT NULL CONSTRAINT [DF_InstrumentConfiguration_RecordNumber]  DEFAULT ((0)),
        [InstrumentPrefix] [varchar](15) NOT NULL,
 CONSTRAINT [PK_InstrumentConfiguration] PRIMARY KEY CLUSTERED
(
        [FactorySerialNumber] ASC,
        [ConfigurationDate] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO

IF NOT EXISTS (SELECT * FROM sys.indexes WHERE object_id = OBJECT_ID(N'[Core].[InstrumentSurvey]') AND name =
N'UN_InstrumentConfiguration_InstrumentConfigurationID')
CREATE UNIQUE NONCLUSTERED INDEX [UN_InstrumentConfiguration_InstrumentConfigurationID] ON
[Core].[InstrumentSurvey]
(
        [InstrumentConfigurationID] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
GO
IF NOT EXISTS (SELECT * FROM ::fn_listextendedproperty(N'MS_Description' , N'SCHEMA',N'Core',
N'TABLE',N'InstrumentSurvey', N'COLUMN',N'LastStoredValueDate'))
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'inizialmente si pone uguale a ConfigurationDate'
, @level0type=N'SCHEMA',@level0name=N'Core', @level1type=N'TABLE',@level1name=N'InstrumentSurvey',
@level2type=N'COLUMN',@level2name=N'LastStoredValueDate'
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[CustomViewMeasure]') AND type in
(N'U'))
BEGIN
CREATE TABLE [Core].[CustomViewMeasure](
        [CustomViewId] [int] NOT NULL,
        [InstrumentConfigurationId] [int] NOT NULL,
        [MeasureIndex] [int] NOT NULL,
 CONSTRAINT [PK_CustomViewMeasure] PRIMARY KEY CLUSTERED
(
        [CustomViewId] ASC,
        [InstrumentConfigurationId] ASC,
        [MeasureIndex] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[InstrumentRegistry]') AND type in
(N'U'))
BEGIN
CREATE TABLE [Core].[InstrumentRegistry](
        [FactorySerialNumber] [varchar](20) NOT NULL,
        [InstrumentPrefix] [varchar](15) NOT NULL,
        [UserSerialNumber] [varchar](50) NOT NULL CONSTRAINT [DF_SltnInstrument_UserSerialNumber] DEFAULT
('=FactorySerialNumber'),
        [Description] [varchar](200) NOT NULL CONSTRAINT [DF_SltnInstrument_Description] DEFAULT
('=FactorySerialNumber'),
        DbDescription [varchar] (200) NULL,
 CONSTRAINT [PK_InstrumentRegistry] PRIMARY KEY CLUSTERED
(
        [FactorySerialNumber] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO

IF NOT EXISTS (SELECT * FROM sys.indexes WHERE object_id = OBJECT_ID(N'[Core].[InstrumentRegistry]') AND name =
N'IX_InstrumentRegistry')
CREATE UNIQUE NONCLUSTERED INDEX [IX_InstrumentRegistry] ON [Core].[InstrumentRegistry]
(
        [FactorySerialNumber] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[InstrumentMeasure]') AND type in
(N'U'))
BEGIN
CREATE TABLE [Core].[InstrumentMeasure](
        [InstrumentConfigurationId] [int] NOT NULL,
        [MeasureIndex] [int] NOT NULL,
        [MeasureName] [nvarchar](50) NOT NULL CONSTRAINT [DF_InstrumentMeasure_MeasureName]  DEFAULT (N'-'),
        [MeasurePrecision] [int] NOT NULL CONSTRAINT [DF_InstrumentMeasure_MeasurePrecision]  DEFAULT ((2)),
        [MeasureUnit] [nvarchar](15) NOT NULL CONSTRAINT [DF_InstrumentMeasure_UnitMeasure]  DEFAULT (N'-'),
        [MeasureProperty] [nvarchar](20) NOT NULL CONSTRAINT [DF_InstrumentMeasure_MeasureProperty] DEFAULT
((0)),
        [MeasureId] [int] NOT NULL CONSTRAINT [DF_InstrumentMeasure_MeasureType]  DEFAULT ((0)),
 CONSTRAINT [PK_InstrumentMeasure_1] PRIMARY KEY CLUSTERED
(
        [InstrumentConfigurationId] ASC,
        [MeasureIndex] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[InstrumentElaborationBase]') AND
type in (N'U'))
BEGIN
CREATE TABLE [Core].[InstrumentElaborationBase](
        [InstrumentConfigurationId] [int] NOT NULL,
        [ElaborationBaseIndex] [int] NOT NULL,
        [ElaborationMeasureIndex] [int] NOT NULL,
        [ElaborationRate] [int] NOT NULL,
        [ElaborationItemNumber] [int] NOT NULL,
 CONSTRAINT [PK_InstrumentElaborationBase] PRIMARY KEY CLUSTERED
(
        [InstrumentConfigurationId] ASC,
        [ElaborationMeasureIndex] ASC,
        [ElaborationBaseIndex] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
IF NOT EXISTS (SELECT * FROM ::fn_listextendedproperty(N'MS_Description' , N'SCHEMA',N'Core',
N'TABLE',N'InstrumentElaborationBase', N'COLUMN',N'ElaborationRate'))
```

```
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Rate di elaborazione in secondi' ,
@level0type=N'SCHEMA',@level0name=N'Core', @level1type=N'TABLE',@level1name=N'InstrumentElaborationBase',
@level2type=N'COLUMN',@level2name=N'ElaborationRate'
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[ValueConfiguration]') AND type in
(N'U'))
BEGIN
CREATE TABLE [Core].[ValueConfiguration](
        [ValueConfigurationID] [int] IDENTITY(1,1) NOT NULL,
        [InstrumentConfigurationID] [int] NOT NULL,
        [MeasureIndex] [int] NOT NULL,
        [ElaborationBaseIndex] [int] NOT NULL,
        [ElaborationItemIndex] [int] NOT NULL,
        [ElaborationType] [int] NOT NULL CONSTRAINT [DF_ValueConfiguration_ElabType]  DEFAULT ((0)),
        [Position] [int] NOT NULL CONSTRAINT [DF_ValueConfiguration_Position]  DEFAULT ((0)),
 CONSTRAINT [PK_ValueConfiguration] PRIMARY KEY CLUSTERED
(
        [ValueConfigurationID] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[LastInstValue]') AND type in
(N'U'))
BEGIN
CREATE TABLE [Core].[LastInstValue](
        [ValueConfigurationID] [int] NOT NULL,
        [InstDate] [datetime] NOT NULL,
        [InstValue] [real] NOT NULL,
 CONSTRAINT [PK_LastInstValue] PRIMARY KEY CLUSTERED
(
        [ValueConfigurationID] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[RawValue]') AND type in (N'U'))
BEGIN
CREATE TABLE [Core].[RawValue](
        [ValueConfigurationID] [int] NOT NULL,
        [ElaborationDate] [datetime] NOT NULL,
        [ElaborationValue] [real] NOT NULL,
        [ValidPercentage] [tinyint] NOT NULL CONSTRAINT [DF_RawValue_ValidPercentage]  DEFAULT ((100)),
 CONSTRAINT [PK_RawValue] PRIMARY KEY CLUSTERED
(
        [ElaborationDate] ASC,
        [ValueConfigurationID] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO

IF NOT EXISTS (SELECT * FROM sys.indexes WHERE object_id = OBJECT_ID(N'[Core].[RawValue]') AND name =
N'NCINDX_RawValue_ElaborationDate')
CREATE NONCLUSTERED INDEX [NCINDX_RawValue_ElaborationDate] ON [Core].[RawValue]
(
        [ElaborationDate] ASC
)
INCLUDE ( [ValueConfigurationID],
[ElaborationValue]) WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
GO

--Aggiunto con la versione 2.6 solo per i nuovi database
IF NOT EXISTS (SELECT * FROM sys.indexes WHERE object_id = OBJECT_ID(N'[Core].[RawValue]') AND name =
N'NCINDX_RawValue_ElaborationeDateDESC')
CREATE NONCLUSTERED INDEX [NCINDX_RawValue_ElaborationeDateDESC] ON [Core].[RawValue]
(
        [ElaborationDate] DESC
)
INCLUDE ( [ValueConfigurationID],
[ElaborationValue]) WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
GO

IF NOT EXISTS (SELECT * FROM ::fn_listextendedproperty(N'MS_Description' , N'SCHEMA',N'Core',
N'TABLE',N'RawValue', N'COLUMN',N'ValidPercentage'))
```

```
EXEC   sys.sp_addextendedproperty   @name=N'MS_Description',   @value=N'il   tipo   va   da   0   a   255'   ,
@level0type=N'SCHEMA',@level0name=N'Core',                    @level1type=N'TABLE',@level1name=N'RawValue',
@level2type=N'COLUMN',@level2name=N'ValidPercentage'
GO

-------------------------------------------------------------------------------------
-- Create Views and Stored Procedures
-------------------------------------------------------------------------------------

IF NOT EXISTS (SELECT * FROM sys.views WHERE object_id = OBJECT_ID(N'[Core].[vInstrumentSurvey_GetRecordCount]'))
EXEC dbo.sp_executesql @statement = N'CREATE VIEW [Core].[vInstrumentSurvey_GetRecordCount]
AS
SELECT                                      COUNT(Core.RawValue.ValueConfigurationID)       AS       RecordCount,
Core.ValueConfiguration.InstrumentConfigurationID
FROM         Core.ValueConfiguration INNER JOIN
                      Core.RawValue          ON          Core.ValueConfiguration.ValueConfigurationID          =
Core.RawValue.ValueConfigurationID
WHERE     (Core.RawValue.ValueConfigurationID IN
                          (SELECT     ValueConfigurationID
                            FROM          Core.ValueConfiguration AS ValueConfiguration_1
                            WHERE      (ElaborationBaseIndex >= 0)))
GROUP BY Core.ValueConfiguration.InstrumentConfigurationID
'
GO
IF   NOT   EXISTS   (SELECT   *   FROM   ::fn_listextendedproperty(N'MS_DiagramPane1'   ,   N'SCHEMA',N'Core',
N'VIEW',N'vInstrumentSurvey_GetRecordCount', NULL,NULL))
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties =
   Begin PaneConfigurations =
      Begin PaneConfiguration = 0
         NumPanes = 4
         Configuration = "(H (1[21] 4[21] 2[23] 3) )"
      End
      Begin PaneConfiguration = 1
         NumPanes = 3
         Configuration = "(H (1 [50] 4 [25] 3))"
      End
      Begin PaneConfiguration = 2
         NumPanes = 3
         Configuration = "(H (1 [50] 2 [25] 3))"
      End
      Begin PaneConfiguration = 3
         NumPanes = 3
         Configuration = "(H (4 [30] 2 [40] 3))"
      End
      Begin PaneConfiguration = 4
         NumPanes = 2
         Configuration = "(H (1 [56] 3))"
      End
      Begin PaneConfiguration = 5
         NumPanes = 2
         Configuration = "(H (2 [66] 3))"
      End
      Begin PaneConfiguration = 6
         NumPanes = 2
         Configuration = "(H (4 [50] 3))"
      End
      Begin PaneConfiguration = 7
         NumPanes = 1
         Configuration = "(V (3))"
      End
      Begin PaneConfiguration = 8
         NumPanes = 3
         Configuration = "(H (1[56] 4[18] 2) )"
      End
      Begin PaneConfiguration = 9
         NumPanes = 2
         Configuration = "(H (1 [75] 4))"
      End
      Begin PaneConfiguration = 10
         NumPanes = 2
         Configuration = "(H (1[66] 2) )"
      End
      Begin PaneConfiguration = 11
         NumPanes = 2
         Configuration = "(H (4 [60] 2))"
      End
      Begin PaneConfiguration = 12
         NumPanes = 1
         Configuration = "(H (1) )"
      End
      Begin PaneConfiguration = 13
         NumPanes = 1
         Configuration = "(V (4))"
      End
      Begin PaneConfiguration = 14
         NumPanes = 1
         Configuration = "(V (2))"
      End
      ActivePaneConfig = 0
   End
   Begin DiagramPane = 
      Begin Origin = 
         Top = 0
         Left = 0
      End
      Begin Tables = 
         Begin Table = "RawValue (Core)"
            Begin Extent = 
               Top = 6
               Left = 286
               Bottom = 121
               Right = 469
            End
            DisplayFlags = 280
            TopColumn = 0
         End
         Begin Table = "ValueConfiguration (Core)"
            Begin Extent = 
               Top = 0
               Left = 0
               Bottom = 115
               Right = 210
            End
            DisplayFlags = 280
            TopColumn = 0
         End
      End
   End
   Begin SQLPane = 
   End
   Begin DataPane = 
      Begin ParameterDefaults = ""
      End
      Begin ColumnWidths = 9
         Width = 284
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
      End
   End
   Begin CriteriaPane = 
      Begin ColumnWidths = 12
         Column = 2970
         Alias = 900
         Table = 2370
         Output = 720
         Append = 1400
         NewValue = 1170
         SortType = 1350
         SortOrder = 1410
         GroupBy = 1350
         Filter = 1350
         Or = 1350
         Or = 1350
         Or = 1350
      End
   End
End
' , @level0type=N'SCHEMA',@level0name=N'Core', @level1type=N'VIEW',@level1name=N'vInstrumentSurvey_GetRecordCount'
GO
IF NOT EXISTS (SELECT * FROM ::fn_listextendedproperty(N'MS_DiagramPaneCount' , N'SCHEMA',N'Core', 
N'VIEW',N'vInstrumentSurvey_GetRecordCount', NULL,NULL))
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount', @value=1 , 
@level0type=N'SCHEMA',@level0name=N'Core', @level1type=N'VIEW',@level1name=N'vInstrumentSurvey_GetRecordCount'
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = 
OBJECT_ID(N'[Core].[spInstrumentSurvey_UpdateNumrecords]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
-- =============================================
-- Author:              Stefano Giarola
-- Create date: 7/5/2008
-- Description:  Updates instrument survey record numbers
-- =============================================
CREATE PROCEDURE [Core].[spInstrumentSurvey_UpdateNumrecords]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationID int
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
```

```
	-- interfering with SELECT statements.
	SET NOCOUNT ON;
	DECLARE @RecordCount int

	SELECT  @RecordCount= count(Core.RawValue.ValueCOnfigurationID) FROM Core.RawValue
	WHERE Core.RawValue.ValueCOnfigurationID IN
	(
		select ValueConfigurationID
		FROM Core.ValueCOnfiguration
		WHERE ElaborationBaseIndex >=0 AND InstrumentConfigurationID=@InstrumentConfigurationID
	)

	UPDATE Core.InstrumentSurvey SET NumRecords=@RecordCount WHERE
InstrumentConfigurationID=@InstrumentConfigurationID
	Return @RecordCount
END
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spRawValue_Add]') AND type in
(N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'

-- =============================================
-- Author:		Stefano Giarola
-- Create date: 27/10/2006
-- Description:	Add a single row of data, checking primarykey violation
-- Return:		1=added, -1=error, 0=already in the table
-- =============================================
CREATE PROCEDURE [Core].[spRawValue_Add]
	-- Add the parameters for the stored procedure here
	@ValueConfigurationID int,
	@ElaborationDate datetime,
	@ElaborationValue real,
	@ValidPercentage tinyint,
	@RowCount int output
AS
BEGIN
	-- SET NOCOUNT ON added to prevent extra result sets from
	-- interfering with SELECT statements.
	SET NOCOUNT ON;

    -- Insert statements for procedure here
	BEGIN TRY
		INSERT INTO Core.RawValue
		(
			ValueConfigurationID,
			ElaborationDate,
			ElaborationValue,
			ValidPercentage
		)
		VALUES
		(
			@ValueConfigurationID,
			@ElaborationDate,
			@ElaborationValue,
			@ValidPercentage
		)
		SET @RowCount=''1''

	END TRY
	BEGIN CATCH
		IF (ERROR_NUMBER() = 2627) BEGIN
			SET @RowCount=''0''
		END
		ELSE BEGIN
			SET @RowCount=''-1''
		END
	END CATCH

END
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.views WHERE object_id = OBJECT_ID(N'[Core].[vFlatCoreData]'))
EXEC dbo.sp_executesql @statement = N'CREATE VIEW [Core].[vFlatCoreData]
AS
SELECT		Core.InstrumentSurvey.InstrumentConfigurationID, Core.InstrumentSurvey.FactorySerialNumber,
Core.InstrumentSurvey.ConfigurationDate,
			Core.ValueConfiguration.MeasureIndex, Core.ValueConfiguration.ElaborationBaseIndex,
Core.ValueConfiguration.ElaborationItemIndex,
                      Core.ValueConfiguration.ValueConfigurationID,                 Core.RawValue.ElaborationDate,
Core.RawValue.ElaborationValue,
                      Core.RawValue.ValidPercentage
FROM         Core.ValueConfiguration INNER JOIN
                      Core.RawValue           ON           Core.ValueConfiguration.ValueConfigurationID           =
Core.RawValue.ValueConfigurationID INNER JOIN
                      Core.InstrumentSurvey      ON      Core.ValueConfiguration.InstrumentConfigurationID      =
Core.InstrumentSurvey.InstrumentConfigurationID
'
GO
IF    NOT    EXISTS    (SELECT    *    FROM    ::fn_listextendedproperty(N'MS_DiagramPane1'    ,    N'SCHEMA',N'Core',
N'VIEW',N'vFlatCoreData', NULL,NULL))
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties =
   Begin PaneConfigurations =
      Begin PaneConfiguration = 0
         NumPanes = 4
         Configuration = "(H (1[38] 4[17] 2[16] 3) )"
      End
      Begin PaneConfiguration = 1
         NumPanes = 3
         Configuration = "(H (1 [50] 4 [25] 3))"
      End
      Begin PaneConfiguration = 2
         NumPanes = 3
         Configuration = "(H (1 [50] 2 [25] 3))"
      End
      Begin PaneConfiguration = 3
         NumPanes = 3
         Configuration = "(H (4 [30] 2 [40] 3))"
      End
      Begin PaneConfiguration = 4
         NumPanes = 2
         Configuration = "(H (1 [56] 3))"
      End
      Begin PaneConfiguration = 5
         NumPanes = 2
         Configuration = "(H (2 [66] 3))"
      End
      Begin PaneConfiguration = 6
         NumPanes = 2
         Configuration = "(H (4 [50] 3))"
      End
      Begin PaneConfiguration = 7
         NumPanes = 1
         Configuration = "(V (3))"
      End
      Begin PaneConfiguration = 8
         NumPanes = 3
         Configuration = "(H (1[56] 4[18] 2) )"
      End
      Begin PaneConfiguration = 9
         NumPanes = 2
         Configuration = "(H (1 [75] 4))"
      End
      Begin PaneConfiguration = 10
         NumPanes = 2
         Configuration = "(H (1[66] 2) )"
      End
      Begin PaneConfiguration = 11
         NumPanes = 2
         Configuration = "(H (4 [60] 2))"
      End
      Begin PaneConfiguration = 12
         NumPanes = 1
         Configuration = "(H (1) )"
      End
      Begin PaneConfiguration = 13
         NumPanes = 1
         Configuration = "(V (4))"
      End
      Begin PaneConfiguration = 14
         NumPanes = 1
         Configuration = "(V (2))"
      End
      ActivePaneConfig = 0
   End
   Begin DiagramPane =
      Begin Origin =
         Top = 0
         Left = 0
      End
      Begin Tables =
         Begin Table = "ValueConfiguration (Core)"
            Begin Extent =
               Top = 18
               Left = 248
               Bottom = 133
               Right = 458
            End
            DisplayFlags = 280
            TopColumn = 0
         End
         Begin Table = "RawValue (Core)"
            Begin Extent =
               Top = 18
               Left = 630
               Bottom = 133
               Right = 813
            End
            DisplayFlags = 280
            TopColumn = 0
         End
         Begin Table = "InstrumentSurvey (Core)"
            Begin Extent =
               Top = 6
               Left = 38
               Bottom = 121
               Right = 248
            End
            DisplayFlags = 280
            TopColumn = 0
         End
      End
   End
   Begin SQLPane =
   End
   Begin DataPane =
      Begin ParameterDefaults = ""
      End
      Begin ColumnWidths = 9
         Width = 284
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
      End
   End
   Begin CriteriaPane =
      Begin ColumnWidths = 11
         Column = 1440
         Alias = 900
         Table = 1170
         Output = 720
         Append = 1400
         NewValue = 1170
         SortType = 1350
         SortOrder = 1410
         GroupBy = 1350
         Filter = 1350
         Or = 1350
         Or = 1350
         Or = 1350
      End
   End
End
' , @level0type=N'SCHEMA',@level0name=N'Core', @level1type=N'VIEW',@level1name=N'vFlatCoreData'
GO
IF NOT EXISTS (SELECT * FROM ::fn_listextendedproperty(N'MS_DiagramPaneCount' , N'SCHEMA',N'Core', N'VIEW',N'vFlatCoreData', NULL,NULL))
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount', @value=1 , @level0type=N'SCHEMA',@level0name=N'Core', @level1type=N'VIEW',@level1name=N'vFlatCoreData'
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spInstrumentRegistry_Remove]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'-- =============================================
-- Author:              Stefano Giarola
-- Create date: 17/4/2008
-- Description:  Remove a selected InstrumentRegistry
-- =============================================
CREATE PROCEDURE [Core].[spInstrumentRegistry_Remove]
        -- Add the parameters for the stored procedure here
        @FactorySerialNumber varchar(20)
AS
BEGIN
        --impostare per avere il valore di ritorno
        --SET NOCOUNT ON;

    -- Insert statements for procedure here
        DELETE FROM Core.InstrumentRegistry WHERE FactorySerialNumber=@FactorySerialNumber
END
```

```
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spInstrumentRegistry_Add]') AND
type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
-- =============================================
-- Author:              Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Insert a new Instrument if not already present
-- =============================================
CREATE PROCEDURE [Core].[spInstrumentRegistry_Add]
        -- Add the parameters for the stored procedure here
                @FactorySerialNumber varchar(20),
                @InstrumentPrefix varchar(15),
                @UserSerialNumber varchar(50),
                @Description varchar(200)
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Prima dell''inserimento verifica se esiste già il record
        SELECT FactorySerialNumber  FROM Core.InstrumentRegistry  WHERE FactorySerialNumber=@FactorySerialNumber

    -- Insert statements for procedure here
        IF (@@ROWCOUNT <= 0)
        BEGIN
                INSERT INTO Core.InstrumentRegistry
                (
                        FactorySerialNumber,
                        InstrumentPrefix,
                        UserSerialNumber,
                        Description
                )
                VALUES
                (
                        @FactorySerialNumber,
                        @InstrumentPrefix,
                        @UserSerialNumber,
                        @Description
                )
        END
END


'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spInstrumentRegistry_GetAll]') AND
type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'



-- =============================================
-- Author:              Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Load all instruments
-- =============================================
CREATE PROCEDURE [Core].[spInstrumentRegistry_GetAll]
        -- Add the parameters for the stored procedure here
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

SELECT   Core.InstrumentRegistry.InstrumentPrefix, Core.InstrumentRegistry.FactorySerialNumber,
        Core.InstrumentRegistry.UserSerialNumber,                            Core.InstrumentRegistry.[Description],
Core.InstrumentRegistry.DbDescription,
        Core.InstrumentType.Name AS InstrumentName
FROM     Core.InstrumentRegistry INNER JOIN
        Core.InstrumentType ON Core.InstrumentRegistry.InstrumentPrefix = Core.InstrumentType.Prefix
END'

END
GO
```

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spInstrumentRegistry_GetByPrefix]')
AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'



-- =============================================
-- Author:              Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Load all instruments of a defined type
-- =============================================
CREATE PROCEDURE [Core].[spInstrumentRegistry_GetByPrefix]
        -- Add the parameters for the stored procedure here
        @prefix varchar(15)
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert
SELECT   Core.InstrumentRegistry.InstrumentPrefix, Core.InstrumentRegistry.FactorySerialNumber,
        Core.InstrumentRegistry.UserSerialNumber,                           Core.InstrumentRegistry.[Description],
Core.InstrumentRegistry.DbDescription,
        Core.InstrumentType.Name AS InstrumentName
FROM     Core.InstrumentRegistry INNER JOIN
        Core.InstrumentType ON Core.InstrumentRegistry.InstrumentPrefix = Core.InstrumentType.Prefix
WHERE   Core.InstrumentRegistry.InstrumentPrefix = @prefix
END'

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spInstrumentRegistry_GetByFSN]')
AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
-- =============================================
-- Author:              Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Load an instrument
-- =============================================
CREATE PROCEDURE [Core].[spInstrumentRegistry_GetByFSN]
        -- Add the parameters for the stored procedure here
        @fMatr varchar(20)
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
SELECT  Core.InstrumentRegistry.InstrumentPrefix, Core.InstrumentRegistry.FactorySerialNumber,
        Core.InstrumentRegistry.UserSerialNumber,                           Core.InstrumentRegistry.[Description],
Core.InstrumentRegistry.DbDescription,
        Core.InstrumentType.Name AS InstrumentName
FROM     Core.InstrumentRegistry INNER JOIN
        Core.InstrumentType ON Core.InstrumentRegistry.InstrumentPrefix = Core.InstrumentType.Prefix
WHERE FactorySerialNumber=@fMatr
END
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF       NOT       EXISTS       (SELECT       *       FROM       sys.objects       WHERE       object_id       =
OBJECT_ID(N'[Core].[spInstrumentMeasure_GetByConfigurationID]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'


-- =============================================
-- Author:              Stefano Giarola
-- Create date: 4/02/2008
-- Description:  Load measures of an InstrumentSurvey
-- =============================================
CREATE PROCEDURE [Core].[spInstrumentMeasure_GetByConfigurationID]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationId int
AS
```

```
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT  InstrumentConfigurationId, MeasureIndex,MeasureName, MeasurePrecision, MeasureUnit,
                        MeasureProperty, MeasureId
        FROM Core.InstrumentMeasure
        WHERE InstrumentConfigurationId=@InstrumentConfigurationId
END
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spInstrumentMeasure_Add]') AND type
in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'



-- =============================================
-- Author:              Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Add an InstrumentMeasure
-- =============================================
CREATE PROCEDURE [Core].[spInstrumentMeasure_Add]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationId int,
        @MeasureIndex int,
        @MeasureName nvarchar(50),
        @MeasurePrecision int,
        @MeasureUnit nvarchar(15),
        @MeasureProperty nvarchar(20),
        @MeasureId int,
        @NewId int output
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        INSERT INTO Core.InstrumentMeasure
        (
                InstrumentConfigurationId,
                MeasureName,
                MeasurePrecision,
                MeasureUnit,
                MeasureProperty,
                MeasureId,
                MeasureIndex
        )
        VALUES
        (
                @InstrumentConfigurationId,
                @MeasureName,
                @MeasurePrecision,
                @MeasureUnit,
                @MeasureProperty,
                @MeasureId,
                @MeasureIndex
        )

        --recupera l''id
        SET @NewId= SCOPE_IDENTITY()

END
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.views WHERE object_id = OBJECT_ID(N'[Core].[vFlatLastInstData]'))
EXEC dbo.sp_executesql @statement = N'
CREATE VIEW [Core].[vFlatLastInstData]
AS
SELECT
Core.InstrumentSurvey.InstrumentConfigurationID,                        Core.InstrumentSurvey.FactorySerialNumber,
Core.InstrumentSurvey.ConfigurationDate,
Core.InstrumentMeasure.MeasureName,
Core.ValueConfiguration.ValueConfigurationID,
Core.LastInstValue.InstDate, Core.LastInstValue.InstValue
FROM
```

```
Core.ValueConfiguration
INNER JOIN Core.LastInstValue ON Core.ValueConfiguration.ValueConfigurationID =
Core.LastInstValue.ValueConfigurationID
INNER JOIN Core.InstrumentSurvey ON Core.ValueConfiguration.InstrumentConfigurationID =
Core.InstrumentSurvey.InstrumentConfigurationID
INNER JOIN Core.InstrumentMeasure ON Core.ValueConfiguration.InstrumentConfigurationID =
Core.InstrumentMeasure.InstrumentConfigurationID
                                                                            AND
Core.ValueConfiguration.MeasureIndex = Core.InstrumentMeasure.MeasureIndex
'
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.views WHERE object_id = OBJECT_ID(N'[Core].[vInstrumentMeasures_GetAll]'))
EXEC dbo.sp_executesql @statement = N'CREATE VIEW [Core].[vInstrumentMeasures_GetAll]
AS
SELECT Core.InstrumentSurvey.FactorySerialNumber, Core.InstrumentSurvey.ConfigurationDate,
Core.ValueConfiguration.ValueConfigurationID,
                      Core.InstrumentMeasure.MeasureName, Core.InstrumentMeasure.MeasureIndex,
Core.ValueConfiguration.ElaborationItemIndex,
                      Core.InstrumentMeasure.MeasurePrecision, Core.InstrumentMeasure.MeasureUnit,
Core.InstrumentMeasure.MeasureProperty,
                      Core.InstrumentMeasure.MeasureId
FROM Core.InstrumentMeasure INNER JOIN
                      Core.InstrumentSurvey ON Core.InstrumentMeasure.InstrumentConfigurationId =
Core.InstrumentSurvey.InstrumentConfigurationID INNER JOIN
                      Core.ValueConfiguration ON Core.InstrumentMeasure.InstrumentConfigurationId =
Core.ValueConfiguration.InstrumentConfigurationID AND
                      Core.InstrumentMeasure.MeasureIndex = Core.ValueConfiguration.MeasureIndex
'
GO
IF NOT EXISTS (SELECT * FROM ::fn_listextendedproperty(N'MS_DiagramPane1' , N'SCHEMA',N'Core',
N'VIEW',N'vInstrumentMeasures_GetAll', NULL,NULL))
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties =
   Begin PaneConfigurations =
      Begin PaneConfiguration = 0
         NumPanes = 4
         Configuration = "(H (1[26] 4[27] 2[16] 3) )"
      End
      Begin PaneConfiguration = 1
         NumPanes = 3
         Configuration = "(H (1 [50] 4 [25] 3))"
      End
      Begin PaneConfiguration = 2
         NumPanes = 3
         Configuration = "(H (1 [50] 2 [25] 3))"
      End
      Begin PaneConfiguration = 3
         NumPanes = 3
         Configuration = "(H (4 [30] 2 [40] 3))"
      End
      Begin PaneConfiguration = 4
         NumPanes = 2
         Configuration = "(H (1 [56] 3))"
      End
      Begin PaneConfiguration = 5
         NumPanes = 2
         Configuration = "(H (2 [66] 3))"
      End
      Begin PaneConfiguration = 6
         NumPanes = 2
         Configuration = "(H (4 [50] 3))"
      End
      Begin PaneConfiguration = 7
         NumPanes = 1
         Configuration = "(V (3))"
      End
      Begin PaneConfiguration = 8
         NumPanes = 3
         Configuration = "(H (1[56] 4[18] 2) )"
      End
      Begin PaneConfiguration = 9
         NumPanes = 2
         Configuration = "(H (1 [75] 4))"
      End
      Begin PaneConfiguration = 10
         NumPanes = 2
         Configuration = "(H (1[66] 2) )"
      End
      Begin PaneConfiguration = 11
         NumPanes = 2
         Configuration = "(H (4 [60] 2))"
      End
      Begin PaneConfiguration = 12
         NumPanes = 1
         Configuration = "(H (1) )"
      End
      Begin PaneConfiguration = 13
         NumPanes = 1
         Configuration = "(V (4))"
      End
      Begin PaneConfiguration = 14
         NumPanes = 1
         Configuration = "(V (2))"
      End
      ActivePaneConfig = 0
   End
   Begin DiagramPane =
      Begin Origin =
         Top = 0
         Left = 0
      End
      Begin Tables =
         Begin Table = "InstrumentMeasure (Core)"
            Begin Extent =
               Top = 6
               Left = 264
               Bottom = 121
               Right = 473
            End
            DisplayFlags = 280
            TopColumn = 0
         End
         Begin Table = "InstrumentSurvey (Core)"
            Begin Extent =
               Top = 8
               Left = 0
               Bottom = 123
               Right = 210
            End
            DisplayFlags = 280
            TopColumn = 0
         End
         Begin Table = "ValueConfiguration (Core)"
            Begin Extent =
               Top = 6
               Left = 533
               Bottom = 121
               Right = 743
            End
            DisplayFlags = 280
            TopColumn = 3
         End
      End
   End
   Begin SQLPane =
   End
   Begin DataPane =
      Begin ParameterDefaults = ""
      End
      Begin ColumnWidths = 12
         Width = 284
         Width = 1500
         Width = 2670
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
      End
   End
   Begin CriteriaPane =
      Begin ColumnWidths = 11
         Column = 1440
         Alias = 900
         Table = 1170
         Output = 720
         Append = 1400
         NewValue = 1170
         SortType = 1350
         SortOrder = 1410
         GroupBy = 1350
         Filter = 1350
         Or = 1350
         Or = 1350
         Or = 1350
      End
   End
End
' , @level0type=N'SCHEMA',@level0name=N'Core', @level1type=N'VIEW',@level1name=N'vInstrumentMeasures_GetAll'
GO
IF   NOT   EXISTS   (SELECT   *   FROM   ::fn_listextendedproperty(N'MS_DiagramPaneCount'   ,   N'SCHEMA',N'Core',
N'VIEW',N'vInstrumentMeasures_GetAll', NULL,NULL))
```

```
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount', @value=1 ,
@level0type=N'SCHEMA',@level0name=N'Core', @level1type=N'VIEW',@level1name=N'vInstrumentMeasures_GetAll'
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spGidasVersion_Get]') AND type in
(N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'-- =============================================
-- Author:              Stefano Giarola
-- Create date: 27/10/2005
-- Description:  Get database version
-- =============================================
CREATE PROCEDURE [Core].[spGidasVersion_Get]
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT Version, ReleaseDate FROM Core.GidasVersion
END
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spInstrumentType_GetAll]') AND type
in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'



-- =============================================
-- Author:              Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Get all instrument types
-- =============================================
CREATE PROCEDURE [Core].[spInstrumentType_GetAll]
        -- Add the parameters for the stored procedure here
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT Prefix, [Name]
        FROM Core.InstrumentType
END
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[Core].[spInstrumentElaborationBase_GetByConfigurationID]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'


-- =============================================
-- Author:              Stefano Giarola
-- Create date: 4/02/2008
-- Description:  Get all elaboration bases of a defined Survey
-- =============================================
CREATE PROCEDURE [Core].[spInstrumentElaborationBase_GetByConfigurationID]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationId int
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT  ElaborationBaseIndex, ElaborationRate,ElaborationMeasureIndex,ElaborationItemNumber
        FROM Core.InstrumentElaborationBase
        WHERE InstrumentConfigurationId=@InstrumentConfigurationId
END
```

```
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spInstrumentElaborationBase_Add]')
AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'



-- =============================================
-- Author:              Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Add a new ElaborationBase
-- =============================================
CREATE PROCEDURE [Core].[spInstrumentElaborationBase_Add]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationId int,
        @ElaborationBaseIndex int,
        @ElaborationRate int,
        @ElaborationMeasureIndex int,
        @ElaborationItemNumber int

AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        INSERT INTO Core.InstrumentElaborationBase
        (
                InstrumentConfigurationId,
                ElaborationBaseIndex,
                ElaborationRate,
                ElaborationMeasureIndex,
                ElaborationItemNumber
        )
        VALUES
        (
                @InstrumentConfigurationId,
                @ElaborationBaseIndex,
                @ElaborationRate,
                @ElaborationMeasureIndex,
                @ElaborationItemNumber
        )

END
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[Core].[spElogConfiguration_GetByConfigurationID]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'


-- =============================================
-- Author:              Stefano Giarola
-- Create date: 4/02/2008
-- Description:  Get extended properties of an E-Log configuration
-- =============================================
CREATE PROCEDURE [Core].[spElogConfiguration_GetByConfigurationID]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationId int
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT  UpdateRate, CISSConfiguration, FirmwareVersion
        FROM Core.ELogConfiguration
        WHERE InstrumentConfigurationId=@InstrumentConfigurationId
END
```

```
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spELogConfiguration_Add]') AND type
in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'-- =============================================
-- Author:              Stefano Giarola
-- Create date: 27/1072006
-- Description:  Insert ELog configuration
-- =============================================
CREATE PROCEDURE [Core].[spELogConfiguration_Add]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationId int,
        @UpdateRate int,
        @FirmwareVersion char(8),
        @CISSConfiguration ntext
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;
        INSERT     INTO     Core.ELogConfiguration     (InstrumentConfigurationId,     UpdateRate,FirmwareVersion,
CISSConfiguration)
        VALUES (@InstrumentConfigurationId, @UpdateRate,@FirmwareVersion, @CISSConfiguration)

END









'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spInstrumentSurvey_Remove]') AND
type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'-- =============================================
-- Author:              Stefano Giarola
-- Create date: 17/4/2008
-- Description:  Remove a selected InstrumentSurvey
-- =============================================
CREATE PROCEDURE [Core].[spInstrumentSurvey_Remove]
        -- Add the parameters for the stored procedure here
        @FactorySerialNumber varchar(20),
        @ConfigurationDate datetime
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        --SET NOCOUNT ON;
    -- Insert statements for procedure here
        DELETE FROM Core.InstrumentSurvey WHERE FactorySerialNumber=@FactorySerialNumber
        AND ConfigurationDate=@ConfigurationDate
END

'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.views WHERE object_id = OBJECT_ID(N'[Core].[vInstrumentSurvey_GetMoreRecentID]'))
EXEC dbo.sp_executesql @statement = N'CREATE VIEW [Core].[vInstrumentSurvey_GetMoreRecentID]
AS
SELECT     FactorySerialNumber, MAX(InstrumentConfigurationID) AS maxCfgId
FROM         Core.InstrumentSurvey
GROUP BY FactorySerialNumber
'
GO
IF    NOT    EXISTS    (SELECT    *    FROM    ::fn_listextendedproperty(N'MS_DiagramPane1'    ,    N'SCHEMA',N'Core',
N'VIEW',N'vInstrumentSurvey_GetMoreRecentID', NULL,NULL))
```

```
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties =
   Begin PaneConfigurations =
      Begin PaneConfiguration = 0
         NumPanes = 4
         Configuration = "(H (1[40] 4[20] 2[20] 3) )"
      End
      Begin PaneConfiguration = 1
         NumPanes = 3
         Configuration = "(H (1 [50] 4 [25] 3))"
      End
      Begin PaneConfiguration = 2
         NumPanes = 3
         Configuration = "(H (1 [50] 2 [25] 3))"
      End
      Begin PaneConfiguration = 3
         NumPanes = 3
         Configuration = "(H (4 [30] 2 [40] 3))"
      End
      Begin PaneConfiguration = 4
         NumPanes = 2
         Configuration = "(H (1 [56] 3))"
      End
      Begin PaneConfiguration = 5
         NumPanes = 2
         Configuration = "(H (2 [66] 3))"
      End
      Begin PaneConfiguration = 6
         NumPanes = 2
         Configuration = "(H (4 [50] 3))"
      End
      Begin PaneConfiguration = 7
         NumPanes = 1
         Configuration = "(V (3))"
      End
      Begin PaneConfiguration = 8
         NumPanes = 3
         Configuration = "(H (1[56] 4[18] 2) )"
      End
      Begin PaneConfiguration = 9
         NumPanes = 2
         Configuration = "(H (1 [75] 4))"
      End
      Begin PaneConfiguration = 10
         NumPanes = 2
         Configuration = "(H (1[66] 2) )"
      End
      Begin PaneConfiguration = 11
         NumPanes = 2
         Configuration = "(H (4 [60] 2))"
      End
      Begin PaneConfiguration = 12
         NumPanes = 1
         Configuration = "(H (1) )"
      End
      Begin PaneConfiguration = 13
         NumPanes = 1
         Configuration = "(V (4))"
      End
      Begin PaneConfiguration = 14
         NumPanes = 1
         Configuration = "(V (2))"
      End
      ActivePaneConfig = 0
   End
   Begin DiagramPane =
      Begin Origin =
         Top = 0
         Left = 0
      End
      Begin Tables =
         Begin Table = "InstrumentSurvey (Core)"
            Begin Extent =
               Top = 30
               Left = 134
               Bottom = 145
               Right = 430
            End
            DisplayFlags = 280
            TopColumn = 0
         End
      End
   End
   Begin SQLPane =
   End
   Begin DataPane =
      Begin ParameterDefaults = ""
      End
      Begin ColumnWidths = 9
         Width = 284
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
      End
   End
   Begin CriteriaPane =
      Begin ColumnWidths = 12
         Column = 2550
         Alias = 900
         Table = 1170
         Output = 720
         Append = 1400
         NewValue = 1170
         SortType = 1350
         SortOrder = 1410
         GroupBy = 1350
         Filter = 1350
         Or = 1350
         Or = 1350
         Or = 1350
      End
   End
End
' , @level0type=N'SCHEMA',@level0name=N'Core',
@level1type=N'VIEW',@level1name=N'vInstrumentSurvey_GetMoreRecentID'
GO
IF NOT EXISTS (SELECT * FROM ::fn_listextendedproperty(N'MS_DiagramPaneCount' , N'SCHEMA',N'Core',
N'VIEW',N'vInstrumentSurvey_GetMoreRecentID', NULL,NULL))
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount', @value=1 ,
@level0type=N'SCHEMA',@level0name=N'Core', @level1type=N'VIEW',@level1name=N'vInstrumentSurvey_GetMoreRecentID'
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[Core].[spLastInstValue_RemoveBySerialNumber]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
-- =============================================
-- Author:              Giarola Stefano
-- Create date: 4/5/2008
-- Description:  Remove all instvalue linked to an instrument
-- =============================================
CREATE PROCEDURE [Core].[spLastInstValue_RemoveBySerialNumber]
        @FactorySerialNumber varchar(20)

AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

        DELETE
        FROM  Core.LastInstValue
        WHERE Core.LastInstValue.ValueConfigurationID IN
        (
                Select Core.LastInstValue.ValueConfigurationID
                FROM
                        Core.InstrumentSurvey INNER JOIN
           Core.ValueConfiguration ON Core.InstrumentSurvey.InstrumentConfigurationID =
Core.ValueConfiguration.InstrumentConfigurationID INNER JOIN
           Core.LastInstValue ON Core.ValueConfiguration.ValueConfigurationID =
Core.LastInstValue.ValueConfigurationID
                WHERE
                        Core.InstrumentSurvey.FactorySerialNumber = @FactorySerialNumber
        )
END
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spInstrumentSurvey_Add]') AND type
in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'



-- =============================================
-- Author:              Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Add a new Survey
-- =============================================
CREATE PROCEDURE [Core].[spInstrumentSurvey_Add]
        -- Add the parameters for the stored procedure here
        @FactorySerialNumber varchar(20),
        @ConfigurationDate datetime,
        @InstrumentPrefix varchar(15),
        @LastStoredValueDate datetime,
        @NewId int output
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        INSERT           INTO           [Core].InstrumentSurvey           (FactorySerialNumber,ConfigurationDate,
InstrumentPrefix,LastStoredValueDate)
        VALUES(@FactorySerialNumber,@ConfigurationDate, @InstrumentPrefix,@LastStoredValueDate)

        --recupera l''id
        SET @NewId= SCOPE_IDENTITY()

END
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spInstrumentSurvey_GetAll]') AND
type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'



-- =============================================
-- Author:              Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Get all instrument surveys
-- =============================================
CREATE PROCEDURE [Core].[spInstrumentSurvey_GetAll]
        -- Add the parameters for the stored procedure here
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT
InstrumentConfigurationID,FactorySerialNumber,ConfigurationDate,InstrumentPrefix,LastStoredValueDate,NumRecords
        FROM Core.InstrumentSurvey
END
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spInstrumentSurvey_GetByFSN]') AND
type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'


-- =============================================
-- Author:              Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Get all surveys of a defined instrument
-- =============================================
CREATE PROCEDURE [Core].[spInstrumentSurvey_GetByFSN]
        -- Add the parameters for the stored procedure here
        @FactorySerialNumber char(20)

AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT
InstrumentConfigurationID,FactorySerialNumber,ConfigurationDate,InstrumentPrefix,LastStoredValueDate,NumRecords
        FROM Core.InstrumentSurvey
        WHERE FactorySerialNumber=@FactorySerialNumber
END
```

```
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spInstrumentSurvey_GetByID]') AND
type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'


-- =============================================
-- Author:               Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Get an intrument survey
-- =============================================
CREATE PROCEDURE [Core].[spInstrumentSurvey_GetByID]
        -- Add the parameters for the stored procedure here
        @FactorySerialNumber char(20),
        @ConfigurationDate datetime
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT
InstrumentConfigurationID,FactorySerialNumber,ConfigurationDate,InstrumentPrefix,LastStoredValueDate,NumRecords
        FROM Core.InstrumentSurvey
        WHERE FactorySerialNumber=@FactorySerialNumber AND ConfigurationDate=@ConfigurationDate
END



'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[Core].[spInstrumentSurvey_GetMoreRecentByID]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'


-- =============================================
-- Author:               Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Get more recent survey of a defined instrument
-- =============================================
CREATE PROCEDURE [Core].[spInstrumentSurvey_GetMoreRecentByID]
        -- Add the parameters for the stored procedure here
        @FactorySerialNumber char(20)
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT TOP(1)
InstrumentConfigurationID,FactorySerialNumber,ConfigurationDate,InstrumentPrefix,LastStoredValueDate,NumRecords
        FROM Core.InstrumentSurvey
        WHERE FactorySerialNumber=@FactorySerialNumber
        ORDER BY ConfigurationDate DESC
END



'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spInstrumentSurvey_Refresh]') AND
type in (N'P', N'PC'))
```

```
BEGIN
EXEC dbo.sp_executesql @statement = N'
-- =============================================
-- Author:              Stefano Giarola
-- Create date: 2/11/2006
-- Description:  Refresh instrumentsurvey table of a defined instrument
-- =============================================
CREATE PROCEDURE [Core].[spInstrumentSurvey_Refresh]
        -- Add the parameters for the stored procedure here
        @FactorySerialNumber  char(20),
        @ConfigurationDate datetime,
        @LastStoredValueDate datetime,
        @NumRecords bigint
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        UPDATE Core.InstrumentSurvey
                SET LastStoredValueDate=@LastStoredValueDate, NumRecords=@NumRecords
                WHERE FactorySerialNumber=@FactorySerialNumber AND ConfigurationDate=@ConfigurationDate
END

'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spRegistryLog_Add]') AND type in
(N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'-- =============================================
-- Author:              Stefano Giarola
-- Create date: 15/04/2008
-- Description:  Add a record in the Elogregistry table
-- =============================================
CREATE PROCEDURE [Core].[spRegistryLog_Add]
        -- Add the parameters for the stored procedure here
        @logKey  nvarchar(50),
        @logKeyId nvarchar(50),
        @logKeyNumber int,
        @NewId int output
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;
        INSERT INTO Core.RegistryLog (logKey, logKeyId, logKeyNumber)
        VALUES (@logKey, @logKeyId, @logKeyNumber)

        --recupera l''id
        SET @NewId= SCOPE_IDENTITY()

END

'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spRegistryLog_GetByKey]') AND type
in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'-- =============================================
-- Author:              Stefano Giarola
-- Create date: 15/04/2008
-- Description:  Get registrylog
-- =============================================
CREATE PROCEDURE [Core].[spRegistryLog_GetByKey]
        @logKey nvarchar(50)
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT logId, logKey, logKeyId, logKeyNumber FROM Core.RegistryLog WHERE logKey=@logKey
END
'
END
GO
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spRegistryLog_Remove]') AND type in
(N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'-- =============================================
-- Author:              Stefano Giarola
-- Create date: 17/4/2008
-- Description:  Remove a selected LOG
-- =============================================
CREATE PROCEDURE [Core].[spRegistryLog_Remove]
        -- Add the parameters for the stored procedure here
        @logKey varchar(50)
AS
BEGIN
        --impostare per avere il valore di ritorno
        --SET NOCOUNT ON;

    -- Insert statements for procedure here
        DELETE FROM Core.RegistryLog WHERE logKey=@logKey
END

'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spCustomView_GetAll]') AND type in
(N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'


-- =============================================
-- Author:              Stefano Giarola
-- Create date: 18/04/2008
-- Description:  Get all records from CustomView
-- =============================================
CREATE PROCEDURE [Core].[spCustomView_GetAll]
        -- Add the parameters for the stored procedure here
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT CustomViewId, ViewName
        FROM Core.CustomView
END






'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spCustomView_Add]') AND type in
(N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'



-- =============================================
-- Author:              Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Insert new CustomView
-- =============================================
CREATE PROCEDURE [Core].[spCustomView_Add]
        -- Add the parameters for the stored procedure here
        @ViewName nvarchar(50),
        @NewId int output
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        INSERT INTO [Core].CustomView (ViewName)
        VALUES(@ViewName)
        --recupera l''id
        SET @NewId= SCOPE_IDENTITY()

END







'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spCustomView_Remove]') AND type in
(N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'



-- =============================================
-- Author:              Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Remove a CustomView
-- =============================================
CREATE PROCEDURE [Core].[spCustomView_Remove]
        -- Add the parameters for the stored procedure here
        @CustomViewId int
AS
BEGIN

    -- Insert statements for procedure here
        DELETE  FROM [Core].CustomView WHERE CustomViewId=@CustomViewId

END







'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spCustomView_Update]') AND type in
(N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
-- =============================================
-- Author:              Stefano Giarola
-- Create date: 20/4/2008
-- Description:  Update a CustomView
-- =============================================
CREATE PROCEDURE [Core].[spCustomView_Update]
        -- Add the parameters for the stored procedure here
        @ViewName nvarchar(50),
        @CustomViewId int
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        UPDATE [Core].CustomView SET ViewName=@ViewName
        WHERE CustomViewId=@CustomViewId

END

'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spCustomViewMeasure_Remove]') AND
type in (N'P', N'PC'))
BEGIN
```

```
EXEC dbo.sp_executesql @statement = N'



-- =============================================
-- Author:              Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Remove all measure realted to a  CustomView
-- =============================================
CREATE PROCEDURE [Core].[spCustomViewMeasure_Remove]
        -- Add the parameters for the stored procedure here
        @CustomViewId int
AS
BEGIN

    -- Insert statements for procedure here
        DELETE  FROM [Core].CustomViewMeasure WHERE CustomViewId=@CustomViewId

END







'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spCustomViewMeasure_Add]') AND type
in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'



-- =============================================
-- Author:              Stefano Giarola
-- Create date: 20/4/2008
-- Description:  Insert new CustomViewMeasure
-- =============================================
CREATE PROCEDURE [Core].[spCustomViewMeasure_Add]
        -- Add the parameters for the stored procedure here
        @CustomViewId int,
        @InstrumentConfigurationId int,
        @MeasureIndex int
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        INSERT INTO [Core].CustomViewMeasure (CustomViewId, InstrumentConfigurationId, MeasureIndex)
        VALUES(@CustomViewId, @InstrumentConfigurationId, @MeasureIndex)

END







'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spCustomViewMeasure_GetAll]') AND
type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'



-- =============================================
-- Author:              Stefano Giarola
-- Create date: 18/04/2008
-- Description:  get all records from CustomViewMeasure
-- =============================================
CREATE PROCEDURE [Core].[spCustomViewMeasure_GetAll]
        -- Add the parameters for the stored procedure here
```

```
AS
BEGIN
	-- SET NOCOUNT ON added to prevent extra result sets from
	-- interfering with SELECT statements.
	SET NOCOUNT ON;

    -- Insert statements for procedure here
	SELECT CustomViewId, InstrumentConfigurationId, MeasureIndex
	FROM Core.CustomViewMeasure
END




'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spCustomViewMeasure_GetByCustomViewId]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'



-- =============================================
-- Author:		Stefano Giarola
-- Create date: 18/04/2008
-- Description:	get all cutosmview related records from CustomViewMeasure
-- =============================================
CREATE PROCEDURE [Core].[spCustomViewMeasure_GetByCustomViewId]
	-- Add the parameters for the stored procedure here
	@CustomViewId int
AS
BEGIN
	-- SET NOCOUNT ON added to prevent extra result sets from
	-- interfering with SELECT statements.
	SET NOCOUNT ON;

    -- Insert statements for procedure here
	SELECT CustomViewId, InstrumentConfigurationId, MeasureIndex
	FROM Core.CustomViewMeasure
	WHERE CustomViewId=@CustomViewId
END




'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Viewer].[spCustomFilter_Add]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'-- =============================================
-- Author:		Stefano Giarola
-- Create date: 25/04/2008
-- Description:	Insert a new custom filter
-- =============================================
CREATE PROCEDURE [Viewer].[spCustomFilter_Add]
	-- Add the parameters for the stored procedure here
	@ParentType int,
	@ParentId int,
	@FilterConfiguration ntext,
	@NewId int output
AS
BEGIN
	-- SET NOCOUNT ON added to prevent extra result sets from
	-- interfering with SELECT statements.
	SET NOCOUNT ON;
	INSERT INTO Viewer.CustomFilter (ParentType, ParentId, FilterConfiguration)
	VALUES (@ParentType, @ParentId,  @FilterConfiguration)

	--recupera l''id
	SET @NewId= SCOPE_IDENTITY()
END
```

```
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Viewer].[spCustomFilter_GetAll]') AND type
in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'



-- =============================================
-- Author:               Stefano Giarola
-- Create date: 25/04/2008
-- Description:  get all records from CustomFilter
-- =============================================
CREATE PROCEDURE [Viewer].[spCustomFilter_GetAll]
        -- Add the parameters for the stored procedure here
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT CustomFilterId, ParentType,ParentId, FilterConfiguration
        FROM Viewer.CustomFilter
END






'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Viewer].[spCustomFilter_Remove]') AND type
in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'



-- =============================================
-- Author:               Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Remove a CustomFilter
-- =============================================
CREATE PROCEDURE [Viewer].[spCustomFilter_Remove]
        -- Add the parameters for the stored procedure here
        @CustomFilterId int
AS
BEGIN

    -- Insert statements for procedure here
        DELETE  FROM Viewer.CustomFilter WHERE CustomFilterId=@CustomFilterId

END








'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Viewer].[spCustomFilter_Update]') AND type
in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
```

```
-- =============================================
-- Author:              Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Update a CustomFilter
-- =============================================
CREATE PROCEDURE [Viewer].[spCustomFilter_Update]
        -- Add the parameters for the stored procedure here
        @CustomFilterId int,
        @ParentType int,
        @ParentId int,
        @FilterConfiguration ntext
AS
BEGIN

    -- Insert statements for procedure here
        UPDATE Viewer.CustomFilter SET ParentType=@ParentType,
                        ParentId=@ParentId, FilterConfiguration=@FilterConfiguration
        WHERE CustomFilterId=@CustomFilterId
END







'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF       NOT       EXISTS       (SELECT       *       FROM       sys.objects       WHERE       object_id       =
OBJECT_ID(N'[Viewer].[spCustomFilter_RemoveByParentId]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'



-- =============================================
-- Author:              Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Remove a CustomFilter by ParentId
-- =============================================
CREATE PROCEDURE [Viewer].[spCustomFilter_RemoveByParentId]
        -- Add the parameters for the stored procedure here
        @ParentId int
AS
BEGIN

    -- Insert statements for procedure here
        DELETE  FROM Viewer.CustomFilter WHERE ParentId=@ParentId

END







'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF       NOT       EXISTS       (SELECT       *       FROM       sys.objects       WHERE       object_id       =
OBJECT_ID(N'[Core].[spValueConfiguration_GetMoreRecent]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
-- =============================================================
-- Author:              Stefano Giarola
-- Create date: 16/11/2006
-- Description:
-- =============================================================
CREATE PROCEDURE [Core].[spValueConfiguration_GetMoreRecent]
        -- Add the parameters for the stored procedure here
        @OldInstCfgId int
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;
        --verifica la presenza di una configurazione più recente
        declare @maxId int
        SELECT @maxId=MAX(Core.InstrumentConfiguration.InstrumentConfigurationID)
        FROM   Core.InstrumentConfiguration INNER JOIN
                    Core.InstrumentConfiguration AS InstrumentConfiguration_1 ON
                    Core.InstrumentConfiguration.FactorySerialNumber                              =
InstrumentConfiguration_1.FactorySerialNumber
        WHERE InstrumentConfiguration_1.InstrumentConfigurationID=@OldInstCfgId

        --se ha trovato esegue la query per recuperare tutte le nuove configurazioni
        IF(@maxId > @OldInstCfgId) BEGIN
                SELECT  ValueConfigurationID,  InstrumentConfigurationID,    MeasureIndex,  ElaborationBaseIndex,
ElaborationItemIndex, ElaborationType, Position
                FROM Core.ValueConfiguration
                WHERE InstrumentConfigurationID=@maxId
        END

END



'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spValueConfiguration_GetAll]') AND
type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
-- =============================================
-- Author:              Stefano Giarola
-- Create date: 26/10/2006
-- Description:
-- =============================================
CREATE PROCEDURE [Core].[spValueConfiguration_GetAll]
        -- Add the parameters for the stored procedure here
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT
ValueConfigurationID,InstrumentConfigurationID,MeasureIndex,ElaborationBaseIndex,ElaborationItemIndex,
                    ElaborationType, Position
        FROM Core.ValueConfiguration
END



'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spValueConfiguration_Add]') AND
type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'-- =============================================
-- Author:              Stefano Giarola
-- Create date: 27/10/2006
-- Description:
-- =============================================
CREATE PROCEDURE [Core].[spValueConfiguration_Add]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationID int,
        @MeasureIndex int,
        @ElaborationBaseIndex int,
        @ElaborationItemIndex int,
        @ElaborationType int,
        @Position int,
        @NewId int output
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        INSERT INTO Core.ValueConfiguration
        (
                InstrumentConfigurationID,
                MeasureIndex,
                ElaborationBaseIndex,
                ElaborationItemIndex,
                ElaborationType,
                Position
        )
        VALUES
        (
                @InstrumentConfigurationID,
                @MeasureIndex,
                @ElaborationBaseIndex,
                @ElaborationItemIndex,
                @ElaborationType,
                @Position
        )

        --recupera l''id
        SET @NewId= SCOPE_IDENTITY()
END



'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[Core].[spValueConfiguration_GetByConfigurationID]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
-- =============================================
-- Author:              Stefano Giarola
-- Create date: 26/10/2006
-- Description:
-- =============================================
CREATE PROCEDURE [Core].[spValueConfiguration_GetByConfigurationID]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationID int = -1
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT
        ValueConfigurationID,InstrumentConfigurationID,MeasureIndex,ElaborationBaseIndex,ElaborationItemIndex,
                        ElaborationType, Position
        FROM Core.ValueConfiguration
        WHERE InstrumentConfigurationID=@InstrumentConfigurationID
END



'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[Core].[spValueConfiguration_GetSingleMoreRecent]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'-- ============================================
-- Author:              Stefano Giarola
-- Create date: 16/11/2006
-- Description:
-- =============================================
CREATE PROCEDURE [Core].[spValueConfiguration_GetSingleMoreRecent]
        -- Add the parameters for the stored procedure here
        @OldInstCfgId int,
        @MeasureIndex int,
        @ElaborationBaseIndex int,
        @ElaborationItemIndex int

AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    --Verifica se esiste una nuova configurazione più recente
        declare @maxId int
        SELECT @maxId=MAX(Core.InstrumentConfiguration.InstrumentConfigurationID)
        FROM   Core.InstrumentConfiguration INNER JOIN
                    Core.InstrumentConfiguration AS InstrumentConfiguration_1 ON
                    Core.InstrumentConfiguration.FactorySerialNumber                                 =
InstrumentConfiguration_1.FactorySerialNumber
        WHERE InstrumentConfiguration_1.InstrumentConfigurationID=@OldInstCfgId

        --se ha trovato una nuova configurazione prosegue
        IF(@maxId > @OldInstCfgId) BEGIN

                --ricava la configurazione aggiornata
                SELECT  ValueConfigurationID,  InstrumentConfigurationID,  MeasureIndex,  ElaborationBaseIndex,
ElaborationItemIndex, ElaborationType, Position
                FROM Core.ValueConfiguration
                WHERE InstrumentConfigurationID=@maxId
                AND MeasureIndex=@MeasureIndex
                AND ElaborationBaseIndex=@ElaborationBaseIndex
                AND ElaborationItemIndex=@ElaborationItemIndex
        END
END



'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spLastInstValue_Add]') AND type in
(N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
-- =============================================
-- Author:              Stefano Giarola
-- Create date: 27/10/2006
-- Description:
-- =============================================
CREATE PROCEDURE [Core].[spLastInstValue_Add]
        -- Add the parameters for the stored procedure here
        @ValueConfigurationID int,
        @InstDate datetime,
        @InstValue real
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON

        DECLARE @rc int
        SELECT          @rc=Count(ValueCOnfigurationID)          FROM          Core.LastInstValue          WHERE
ValueConfigurationID=@ValueConfigurationID

        IF(@rc > 0)
                -- aggiorna il record
                UPDATE Core.LastInstValue SET InstDate=@InstDate, InstValue=@InstValue
                WHERE ValueConfigurationID=@ValueConfigurationID

        ELSE
                -- inserisce il nuovo recor
                INSERT INTO Core.LastInstValue
                (
                        ValueConfigurationID,
                        InstDate,
                        InstValue
                )
                VALUES
                (
                        @ValueConfigurationID,
                        @InstDate,
                        @InstValue
                )
END
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spElabTypeList_Get]') AND type in
(N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'-- =============================================
-- Author:              Stefano Giarola
-- Create date: 27/10/2005
-- Description:  Get ElabType list
-- =============================================
CREATE PROCEDURE [Core].[spElabTypeList_Get]
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT IdElabType, ElabTypeString, ElabTypeToDo FROM Core.ElabTypeList
END


'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF       NOT       EXISTS       (SELECT       *       FROM       sys.views       WHERE       object_id       =
OBJECT_ID(N'[Core].[vValueConfiguration_GetByMoreRecentInstrumentConfigurationID]'))
EXEC dbo.sp_executesql @statement = N'/*WHERE      (Core.vInstrumentSurvey_GetMoreRecentID.FactorySerialNumber =
''05110008'')*/
CREATE VIEW [Core].[vValueConfiguration_GetByMoreRecentInstrumentConfigurationID]
AS
SELECT                                          Core.vInstrumentSurvey_GetMoreRecentID.FactorySerialNumber,
Core.ValueConfiguration.ValueConfigurationID,
                      Core.ValueConfiguration.MeasureIndex,          Core.ValueConfiguration.ElaborationBaseIndex,
Core.ValueConfiguration.ElaborationItemIndex,
                      Core.ValueConfiguration.ElaborationType
FROM         Core.vInstrumentSurvey_GetMoreRecentID INNER JOIN
                      Core.ValueConfiguration      ON      Core.vInstrumentSurvey_GetMoreRecentID.maxCfgId      =
Core.ValueConfiguration.InstrumentConfigurationID
'
GO
IF    NOT    EXISTS    (SELECT    *    FROM    ::fn_listextendedproperty(N'MS_DiagramPane1'    ,    N'SCHEMA',N'Core',
N'VIEW',N'vValueConfiguration_GetByMoreRecentInstrumentConfigurationID', NULL,NULL))
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties =
   Begin PaneConfigurations =
      Begin PaneConfiguration = 0
         NumPanes = 4
         Configuration = "(H (1[40] 4[13] 2[20] 3) )"
      End
      Begin PaneConfiguration = 1
         NumPanes = 3
         Configuration = "(H (1 [50] 4 [25] 3))"
      End
      Begin PaneConfiguration = 2
         NumPanes = 3
         Configuration = "(H (1 [50] 2 [25] 3))"
      End
      Begin PaneConfiguration = 3
         NumPanes = 3
         Configuration = "(H (4 [30] 2 [40] 3))"
      End
      Begin PaneConfiguration = 4
         NumPanes = 2
         Configuration = "(H (1 [56] 3))"
      End
      Begin PaneConfiguration = 5
         NumPanes = 2
         Configuration = "(H (2 [66] 3))"
      End
      Begin PaneConfiguration = 6
         NumPanes = 2
         Configuration = "(H (4 [50] 3))"
      End
      Begin PaneConfiguration = 7
         NumPanes = 1
         Configuration = "(V (3))"
      End
      Begin PaneConfiguration = 8
         NumPanes = 3
         Configuration = "(H (1[56] 4[18] 2) )"
      End
      Begin PaneConfiguration = 9
         NumPanes = 2
         Configuration = "(H (1 [75] 4))"
      End
      Begin PaneConfiguration = 10
         NumPanes = 2
         Configuration = "(H (1[66] 2) )"
      End
      Begin PaneConfiguration = 11
         NumPanes = 2
         Configuration = "(H (4 [60] 2))"
      End
      Begin PaneConfiguration = 12
         NumPanes = 1
         Configuration = "(H (1) )"
      End
      Begin PaneConfiguration = 13
         NumPanes = 1
         Configuration = "(V (4))"
      End
      Begin PaneConfiguration = 14
         NumPanes = 1
         Configuration = "(V (2))"
      End
      ActivePaneConfig = 0
   End
   Begin DiagramPane =
      Begin Origin =
         Top = 0
         Left = 0
      End
      Begin Tables =
         Begin Table = "vInstrumentSurvey_GetMoreRecentID (Core)"
            Begin Extent =
               Top = 6
               Left = 38
               Bottom = 91
               Right = 219
            End
            DisplayFlags = 280
            TopColumn = 0
         End
         Begin Table = "ValueConfiguration (Core)"
            Begin Extent =
               Top = 6
               Left = 257
               Bottom = 121
               Right = 467
            End
            DisplayFlags = 280
            TopColumn = 0
         End
      End
   End
   Begin SQLPane =
   End
   Begin DataPane =
      Begin ParameterDefaults = ""
      End
      Begin ColumnWidths = 9
         Width = 284
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
      End
   End
   Begin CriteriaPane =
      Begin ColumnWidths = 11
         Column = 1440
         Alias = 900
         Table = 1170
         Output = 720
         Append = 1400
         NewValue = 1170
         SortType = 1350
         SortOrder = 1410
         GroupBy = 1350
         Filter = 1350
         Or = 1350
         Or = 1350
         Or = 1350
      End
   End
End
'                                    ,                                    @level0type=N'SCHEMA',@level0name=N'Core',
@level1type=N'VIEW',@level1name=N'vValueConfiguration_GetByMoreRecentInstrumentConfigurationID'
GO

-------------------------------------------------------------------------------------
-- Relazioni
------------------------------------------------------------------------------------- */
IF NOT EXISTS (SELECT * FROM ::fn_listextendedproperty(N'MS_DiagramPaneCount' , N'SCHEMA',N'Core',
N'VIEW',N'vValueConfiguration_GetByMoreRecentInstrumentConfigurationID', NULL,NULL))
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount', @value=1 ,
@level0type=N'SCHEMA',@level0name=N'Core',
@level1type=N'VIEW',@level1name=N'vValueConfiguration_GetByMoreRecentInstrumentConfigurationID'
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[Core].[FK_InstrumentSurvey_InstrumentRegistry]') AND parent_object_id =
OBJECT_ID(N'[Core].[InstrumentSurvey]'))
ALTER TABLE [Core].[InstrumentSurvey] WITH CHECK ADD CONSTRAINT [FK_InstrumentSurvey_InstrumentRegistry] FOREIGN
KEY([FactorySerialNumber])
REFERENCES [Core].[InstrumentRegistry] ([FactorySerialNumber])
```

```
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [Core].[InstrumentSurvey] CHECK CONSTRAINT [FK_InstrumentSurvey_InstrumentRegistry]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[Core].[FK_CustomViewMeasure_CustomView]') AND parent_object_id = OBJECT_ID(N'[Core].[CustomViewMeasure]'))
ALTER TABLE [Core].[CustomViewMeasure] WITH CHECK ADD CONSTRAINT [FK_CustomViewMeasure_CustomView] FOREIGN KEY([CustomViewId])
REFERENCES [Core].[CustomView] ([CustomViewId])
ON DELETE CASCADE
GO
ALTER TABLE [Core].[CustomViewMeasure] CHECK CONSTRAINT [FK_CustomViewMeasure_CustomView]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[Core].[FK_CustomViewMeasure_InstrumentMeasure]') AND parent_object_id = OBJECT_ID(N'[Core].[CustomViewMeasure]'))
ALTER TABLE [Core].[CustomViewMeasure] WITH CHECK ADD CONSTRAINT [FK_CustomViewMeasure_InstrumentMeasure] FOREIGN KEY([InstrumentConfigurationId], [MeasureIndex])
REFERENCES [Core].[InstrumentMeasure] ([InstrumentConfigurationId], [MeasureIndex])
ON DELETE CASCADE
GO
ALTER TABLE [Core].[CustomViewMeasure] CHECK CONSTRAINT [FK_CustomViewMeasure_InstrumentMeasure]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[Core].[FK_InstrumentRegistry_InstrumentType]') AND parent_object_id = OBJECT_ID(N'[Core].[InstrumentRegistry]'))
ALTER TABLE [Core].[InstrumentRegistry] WITH CHECK ADD CONSTRAINT [FK_InstrumentRegistry_InstrumentType] FOREIGN KEY([InstrumentPrefix])
REFERENCES [Core].[InstrumentType] ([Prefix])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [Core].[InstrumentRegistry] CHECK CONSTRAINT [FK_InstrumentRegistry_InstrumentType]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[Core].[FK_InstrumentMeasure_InstrumentSurvey]') AND parent_object_id = OBJECT_ID(N'[Core].[InstrumentMeasure]'))
ALTER TABLE [Core].[InstrumentMeasure] WITH CHECK ADD CONSTRAINT [FK_InstrumentMeasure_InstrumentSurvey] FOREIGN KEY([InstrumentConfigurationId])
REFERENCES [Core].[InstrumentSurvey] ([InstrumentConfigurationID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [Core].[InstrumentMeasure] CHECK CONSTRAINT [FK_InstrumentMeasure_InstrumentSurvey]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[Core].[FK_InstrumentElaborationBase_InstrumentSurvey]') AND parent_object_id = OBJECT_ID(N'[Core].[InstrumentElaborationBase]'))
ALTER TABLE [Core].[InstrumentElaborationBase] WITH CHECK ADD CONSTRAINT [FK_InstrumentElaborationBase_InstrumentSurvey] FOREIGN KEY([InstrumentConfigurationId])
REFERENCES [Core].[InstrumentSurvey] ([InstrumentConfigurationID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [Core].[InstrumentElaborationBase] CHECK CONSTRAINT [FK_InstrumentElaborationBase_InstrumentSurvey]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[Core].[FK_ValueConfiguration_InstrumentSurvey]') AND parent_object_id = OBJECT_ID(N'[Core].[ValueConfiguration]'))
ALTER TABLE [Core].[ValueConfiguration] WITH CHECK ADD CONSTRAINT [FK_ValueConfiguration_InstrumentSurvey] FOREIGN KEY([InstrumentConfigurationID])
REFERENCES [Core].[InstrumentSurvey] ([InstrumentConfigurationID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [Core].[ValueConfiguration] CHECK CONSTRAINT [FK_ValueConfiguration_InstrumentSurvey]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[Core].[FK_LastInstValue_ValueConfiguration]') AND parent_object_id = OBJECT_ID(N'[Core].[LastInstValue]'))
ALTER TABLE [Core].[LastInstValue] WITH CHECK ADD CONSTRAINT [FK_LastInstValue_ValueConfiguration] FOREIGN KEY([ValueConfigurationID])
REFERENCES [Core].[ValueConfiguration] ([ValueConfigurationID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [Core].[LastInstValue] CHECK CONSTRAINT [FK_LastInstValue_ValueConfiguration]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[Core].[FK_RawValue_ValueConfiguration]') AND parent_object_id = OBJECT_ID(N'[Core].[RawValue]'))
ALTER TABLE [Core].[RawValue] WITH CHECK ADD CONSTRAINT [FK_RawValue_ValueConfiguration] FOREIGN KEY([ValueConfigurationID])
REFERENCES [Core].[ValueConfiguration] ([ValueConfigurationID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [Core].[RawValue] CHECK CONSTRAINT [FK_RawValue_ValueConfiguration]
GO
```

```
IF NOT EXISTS (SELECT * FROM sys.check_constraints WHERE object_id =
OBJECT_ID(N'[Core].[CK_RawValue_ValidPercentage]') AND parent_object_id = OBJECT_ID(N'[Core].[RawValue]'))
ALTER TABLE [Core].[RawValue] WITH CHECK ADD CONSTRAINT [CK_RawValue_ValidPercentage] CHECK
(([ValidPercentage]>=(0) AND [ValidPercentage]<=(100)))
GO
ALTER TABLE [Core].[RawValue] CHECK CONSTRAINT [CK_RawValue_ValidPercentage]
GO
IF NOT EXISTS (SELECT * FROM ::fn_listextendedproperty(N'MS_Description' , N'SCHEMA',N'Core',
N'TABLE',N'RawValue', N'CONSTRAINT',N'CK_RawValue_ValidPercentage'))
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'valori ammessi 0-100' ,
@level0type=N'SCHEMA',@level0name=N'Core', @level1type=N'TABLE',@level1name=N'RawValue',
@level2type=N'CONSTRAINT',@level2name=N'CK_RawValue_ValidPercentage'
GO

-------------------------------------------------------------------------------------
-- Add AddIns table (versione 2.1.0.0)
------------------------------------------------------------------------------------ */
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[AddIns]') AND type in (N'U'))
BEGIN
CREATE TABLE [Core].[AddIns](
        [AddInCode] [nvarchar](20) NOT NULL,
        [Description] [nvarchar](200) NOT NULL,
        [AttachedDate] [datetime] NOT NULL
 CONSTRAINT [PK_AddIn] PRIMARY KEY CLUSTERED
(
        [AddInCode] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
END
GO

IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spAddIn_Add]') AND type in (N'P',
N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [Core].[spAddIn_Add]
        -- Add the parameters for the stored procedure here
        @AddInCode nvarchar(20),
        @Description nvarchar(200),
        @AttachedDate datetime
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        INSERT INTO [Core].[AddIns] (AddInCode,[Description],AttachedDate)
        VALUES(@AddInCode,@Description,@AttachedDate)
END'
END

-------------------------------------------------------------------------------------
-- Add backup/restore sp (versione 2.3.0.0)
------------------------------------------------------------------------------------ */

IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[Core].[spInstrumentSurvey_GetFastDataRange]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [Core].[spInstrumentSurvey_GetFastDataRange]
        -- Add the parameters for the stored procedure here
        @elabID int,
        @instID int
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        Select Min(ElaborationDate) as mindate, Max(ElaborationDate) as maxdate from Core.RawValue
        WHERE ValueConfigurationID=@elabID or  ValueConfigurationID=@instID
END'
END
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spInstrumentSurvey_GetDataRange]')
AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [Core].[spInstrumentSurvey_GetDataRange]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationID int
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;
    -- Insert statements for procedure here
        Select Min(ElaborationDate) as mindate, Max(ElaborationDate) as maxdate from Core.RawValue
        WHERE ValueConfigurationID in
        (
                SELECT  Core.ValueConfiguration.ValueConfigurationID
                FROM    Core.ValueConfiguration
                where   Core.ValueConfiguration.InstrumentConfigurationID=@InstrumentConfigurationID
        )
END'
END
GO
--------------------------------------------------------------------------------------
-- Add survey support (versione 2.4.0.0)
------------------------------------------------------------------------------------- */
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[UserSurvey]') AND type in (N'U'))
BEGIN
CREATE TABLE [Core].[UserSurvey](
        [InstrumentConfigurationId] [int] NOT NULL,
        [FactorySerialNumber] [nvarchar](20) NOT NULL,
        [DtStart] [datetime] NOT NULL,
        [DtEnd] [datetime] NOT NULL,
        [Description] [nvarchar](200) NOT NULL,
        [SurveyNumber] [int] NOT NULL,
 CONSTRAINT [PK_UserSurvey_1] PRIMARY KEY CLUSTERED
(
        [InstrumentConfigurationId] ASC,
        [DtStart] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS  = ON) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[Core].[FK_UserSurvey_InstrumentSurvey]') AND parent_object_id = OBJECT_ID(N'[Core].[UserSurvey]'))
ALTER TABLE [Core].[UserSurvey] WITH CHECK ADD CONSTRAINT [FK_UserSurvey_InstrumentSurvey] FOREIGN
KEY([InstrumentConfigurationId])
REFERENCES [Core].[InstrumentSurvey] ([InstrumentConfigurationID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
IF  EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[Core].[FK_UserSurvey_InstrumentSurvey]')
AND parent_object_id = OBJECT_ID(N'[Core].[UserSurvey]'))
ALTER TABLE [Core].[UserSurvey] CHECK CONSTRAINT [FK_UserSurvey_InstrumentSurvey]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spUserSurvey_Remove]') AND type in
(N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
-- =============================================
-- Author:               Stefano Giarola
-- Create date: 7/4/2010
-- Description:  Remove a selected UserSurvey
-- =============================================
CREATE PROCEDURE [Core].[spUserSurvey_Remove]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationId int,
        @DtStart datetime
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        --SET NOCOUNT ON;
    -- Insert statements for procedure here
        DELETE FROM Core.UserSurvey
        WHERE InstrumentConfigurationId=@InstrumentConfigurationId AND DtStart=@DtStart
END'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spUserSurvey_GetByFSN]') AND type
in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
```

```
-- =============================================
-- Author:              Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Get all user surveys of a defined instrument
-- =============================================
CREATE PROCEDURE [Core].[spUserSurvey_GetByFSN]
        -- Add the parameters for the stored procedure here
        @FactorySerialNumber char(20)

AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT InstrumentConfigurationID,FactorySerialNumber, DtStart, DtEnd, [Description], SurveyNumber
        FROM Core.UserSurvey
        WHERE FactorySerialNumber=@FactorySerialNumber
END
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spUserSurvey_Add]') AND type in
(N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
-- =============================================
-- Author:              Stefano Giarola
-- Create date: 7/4/2010
-- Description:  Add a new UserSurvey
-- =============================================
CREATE PROCEDURE [Core].[spUserSurvey_Add]
        -- Add the parameters for the stored procedure here
        @FactorySerialNumber varchar(20),
        @InstrumentConfigurationId int,
        @DtStart datetime,
        @DtEnd datetime,
        @Description nvarchar(200),
        @SurveyNumber int
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        INSERT           INTO           Core.UserSurvey           (FactorySerialNumber,InstrumentConfigurationId,
DtStart,DtEnd,[Description],SurveyNumber)
        VALUES(@FactorySerialNumber,@InstrumentConfigurationId, @DtStart,@DtEnd,@Description, @SurveyNumber)

END
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spUserSurvey_AddWithCheck]') AND
type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
-- =============================================
-- Author:              Stefano Giarola
-- Create date: 08/04/2010
-- Description:  Add a single row of User survey, checking primarykey violation
-- Return:              1=added, -1=error, 0=already in the table
-- =============================================
CREATE PROCEDURE [Core].[spUserSurvey_AddWithCheck]
        -- Add the parameters for the stored procedure here
        @FactorySerialNumber varchar(20),
        @InstrumentConfigurationId int,
        @DtStart datetime,
        @DtEnd datetime,
        @Description nvarchar(200),
        @SurveyNumber int,
        @RowCount int output
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        BEGIN TRY
```

```
                INSERT INTO Core.UserSurvey
                (
                        FactorySerialNumber,
                        InstrumentConfigurationId,
                        DtStart,
                        DtEnd,
                        [Description],
                        SurveyNumber
                )
                VALUES
                (
                        @FactorySerialNumber,
                        @InstrumentConfigurationId,
                        @DtStart,
                        @DtEnd,
                        @Description,
                        @SurveyNumber
                )
                SET @RowCount=''1''

        END TRY
        BEGIN CATCH
                IF (ERROR_NUMBER() = 2627) BEGIN
                        SET @RowCount=''0''
                END
                ELSE BEGIN
                        SET @RowCount=''-1''
                END
        END CATCH

END
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spUserSurvey_GetByConfigurationID]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
-- =============================================
-- Author:              Stefano Giarola
-- Create date: 26/10/2006
-- Description:  Get an intrument survey
-- =============================================
CREATE PROCEDURE [Core].[spUserSurvey_GetByConfigurationID]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationId int
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT InstrumentConfigurationID,FactorySerialNumber,DtStart, DtEnd, [Description], SurveyNumber
        FROM Core.UserSurvey
        WHERE InstrumentConfigurationId=@InstrumentConfigurationId
END
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spUserSurvey_GetAll]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
-- =============================================
-- Author:              Stefano Giarola
-- Create date: 7/4/2010
-- Description:  Get all user surveys
-- =============================================
CREATE PROCEDURE [Core].[spUserSurvey_GetAll]
        -- Add the parameters for the stored procedure here
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT InstrumentConfigurationId, FactorySerialNumber, DtStart, DtEnd, [Description], SurveyNumber
        FROM Core.UserSurvey
END
'
```

```
END
GO

------------------------------------------------------------------------------------
-- Add UserConfigurationAttributes support (version 2.5.0.0)
------------------------------------------------------------------------------------ */
/****** Object:  Table [Core].[UserConfigurationAttributes]    Script Date: 05/27/2010 17:43:40 ******/
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[UserConfigurationAttributes]') AND
type in (N'U'))
BEGIN
CREATE TABLE [Core].[UserConfigurationAttributes](
        [InstrumentConfigurationID] [int] NOT NULL,
        [Name] [nvarchar](100) NOT NULL,
        [Description] [nvarchar](200) NOT NULL,
        [Code] [nvarchar](50) NOT NULL
) ON [PRIMARY]
ALTER      TABLE      [Core].[UserConfigurationAttributes]           WITH      CHECK      ADD           CONSTRAINT
[FK_UserConfigurationAttributes_InstrumentSurvey] FOREIGN KEY([InstrumentConfigurationID])
REFERENCES [Core].[InstrumentSurvey] ([InstrumentConfigurationID])
ON UPDATE CASCADE
ON DELETE CASCADE
ALTER             TABLE             [Core].[UserConfigurationAttributes]             CHECK             CONSTRAINT
[FK_UserConfigurationAttributes_InstrumentSurvey]
END

IF       NOT       EXISTS       (SELECT       *       FROM       sys.objects       WHERE       object_id       =
OBJECT_ID(N'[Core].[spUserConfigurationAttributes_Add]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
-- =============================================
-- Author:              Stefano Giarola
-- Create date: 28/4/2010
-- Description:  Add a new spUserConfigurationAttributes
-- =============================================
CREATE PROCEDURE [Core].[spUserConfigurationAttributes_Add]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationId int,
        @Name nvarchar(100),
        @Description nvarchar(200),
        @Code nvarchar(25)
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        INSERT INTO Core.UserConfigurationAttributes (InstrumentConfigurationID, Name, [Description],Code)
        VALUES(@InstrumentConfigurationID, @Name, @Description,@Code)

END'
END

IF       NOT       EXISTS       (SELECT       *       FROM       sys.objects       WHERE       object_id       =
OBJECT_ID(N'[Core].[spUserConfigurationAttributes_GetAll]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
-- =============================================
-- Author:              Stefano Giarola
-- Create date: 28/4/2010
-- Description:  Get all UserConfigurationAttributes
-- =============================================
CREATE PROCEDURE [Core].[spUserConfigurationAttributes_GetAll]
        -- Add the parameters for the stored procedure here
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT InstrumentConfigurationID, Name, [Description],Code FROM Core.UserConfigurationAttributes

END'
END

IF       NOT       EXISTS       (SELECT       *       FROM       sys.objects       WHERE       object_id       =
OBJECT_ID(N'[Core].[spUserConfigurationAttributes_GetByConfigurationID]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
-- =============================================
-- Author:              Stefano Giarola
-- Create date: 28/4/2010
-- Description:  Get UserConfigurationAttributes by ConfigurationID
-- =============================================
CREATE PROCEDURE [Core].[spUserConfigurationAttributes_GetByConfigurationID]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationID int
AS
BEGIN
```

```
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT InstrumentConfigurationID, Name, [Description],Code FROM Core.UserConfigurationAttributes
        WHERE InstrumentConfigurationID=@InstrumentConfigurationID

END'
END
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[Core].[spUserConfigurationAttributes_Update]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
-- =============================================
-- Author:              Stefano Giarola
-- Create date: 28/04/2010
-- Description:  Update a UserConfigurationAttributes
-- =============================================
CREATE PROCEDURE [Core].[spUserConfigurationAttributes_Update]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationID int,
        @Name nvarchar(100),
        @Description nvarchar(200),
        @Code nvarchar(25)

AS
BEGIN

    -- Insert statements for procedure here
        UPDATE Core.UserConfigurationAttributes SET Name=@Name, [Description]=@Description, Code=@Code
        WHERE InstrumentConfigurationID=@InstrumentConfigurationID
END'
END
GO

-------------------------------------------------------------------------------------
-- Add SurveyInstInfo support (version 2.6.0.0 corretto)
------------------------------------------------------------------------------------- */
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[InstrumentSurveyInstInfo]') AND
type in (N'U'))
BEGIN
CREATE TABLE [Core].[InstrumentSurveyInstInfo](
        [InstrumentConfigurationID] [int] NOT NULL,
        [FirstInstDate] [datetime] NOT NULL,
        [LastInstDate] [datetime] NOT NULL,
        [NumIntRecords] [bigint] NOT NULL,
 CONSTRAINT [PK_InstrumentSurveyInstInfo] PRIMARY KEY CLUSTERED
(
        [InstrumentConfigurationID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
ALTER TABLE [Core].[InstrumentSurveyInstInfo] WITH CHECK ADD CONSTRAINT
[FK_InstrumentSurveyInstInfo_InstrumentSurvey] FOREIGN KEY([InstrumentConfigurationID])
REFERENCES [Core].[InstrumentSurvey] ([InstrumentConfigurationID])
ON UPDATE CASCADE
ON DELETE CASCADE
ALTER TABLE [Core].[InstrumentSurveyInstInfo] CHECK CONSTRAINT [FK_InstrumentSurveyInstInfo_InstrumentSurvey]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[Core].[spInstrumentSurveyInstInfo_Refresh]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [Core].[spInstrumentSurveyInstInfo_Refresh]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationID int,
        @InstDate datetime,
        @NumIntRecords bigint

AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON

        DECLARE @rc int
        declare @dt1 datetime
        declare @dt2 datetime
        declare @nr bigint
        SELECT  @rc=InstrumentConfigurationID,  @dt1=FirstInstDate,  @dt2=LastInstDate,  @nr=NumIntRecords  FROM
Core.InstrumentSurveyInstInfo WHERE InstrumentConfigurationID=@InstrumentConfigurationID

        IF(@rc > 0)
                BEGIN
                        -- aggiorna il record scegliendo i valori giusti
                        IF(@dt1 > @InstDate)
                                SET @dt1=@InstDate

                        IF(@dt2 < @InstDate)
                                SET @dt2=@InstDate

                        UPDATE    Core.InstrumentSurveyInstInfo    SET    FirstInstDate=@dt1    ,LastInstDate=@dt2
,NumIntRecords= @nr +@NumIntRecords
                        WHERE InstrumentConfigurationID=@InstrumentConfigurationID
                END
        ELSE
                -- inserisce il nuovo record
                INSERT INTO Core.InstrumentSurveyInstInfo
                (
                        InstrumentConfigurationID,
                        FirstInstDate,
                        LastInstDate,
                        NumIntRecords
                )
                VALUES
                (
                        @InstrumentConfigurationID,
                        @InstDate,
                        @InstDate,
                        @NumIntRecords
                )
END'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF  NOT  EXISTS  (SELECT  *  FROM  sys.objects  WHERE  object_id  =  OBJECT_ID(N'[Core].[spInstrumentSurveyInstInfo
_GetByConfigurationID]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [Core].[spInstrumentSurveyInstInfo _GetByConfigurationID]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationID int

AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT      InstrumentConfigurationID,      FirstInstDate,      LastInstDate,      NumIntRecords      FROM
Core.InstrumentSurveyInstInfo
        WHERE InstrumentConfigurationID=@InstrumentConfigurationID
END'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spLastInstValue26_Add]') AND type
in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [Core].[spLastInstValue26_Add]
        -- Add the parameters for the stored procedure here
        @ValueConfigurationID int,
        @InstDate datetime,
        @InstValue real
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON

        DECLARE @rc int
        SELECT          @rc=Count(ValueCOnfigurationID)          FROM          Core.LastInstValue          WHERE
ValueConfigurationID=@ValueConfigurationID
        IF(@rc > 0)
                begin
                        -- aggiorna il record
                        UPDATE Core.LastInstValue SET InstDate=@InstDate, InstValue=@InstValue
                        WHERE ValueConfigurationID=@ValueConfigurationID
                end
        ELSE
                BEGIN
```

```
                                -- inserisce il nuovo recor
                                INSERT INTO Core.LastInstValue
                                (
                                        ValueConfigurationID,
                                        InstDate,
                                        InstValue
                                )
                                VALUES
                                (
                                        @ValueConfigurationID,
                                        @InstDate,
                                        @InstValue
                                )
                        END
END'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
EXEC dbo.sp_executesql @statement = N'
ALTER PROCEDURE [Core].[spLastInstValue_Add]
        -- Add the parameters for the stored procedure here
        @ValueConfigurationID int,
        @InstDate datetime,
        @InstValue real
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON

        DECLARE @rc int
        SELECT          @rc=Count(ValueCOnfigurationID)          FROM          Core.LastInstValue          WHERE
ValueConfigurationID=@ValueConfigurationID
        IF(@rc > 0)
                begin
                        -- aggiorna il record
                        UPDATE Core.LastInstValue SET InstDate=@InstDate, InstValue=@InstValue
                        WHERE ValueConfigurationID=@ValueConfigurationID
                end
        ELSE
                BEGIN
                        -- inserisce il nuovo recor
                        INSERT INTO Core.LastInstValue
                        (
                                ValueConfigurationID,
                                InstDate,
                                InstValue
                        )
                        VALUES
                        (
                                @ValueConfigurationID,
                                @InstDate,
                                @InstValue
                        )
                END

        DECLARE @cfgid int
        declare @dt1 datetime
        declare @dt2 datetime
        declare @nr bigint
        set @rc=-1

        SELECT         @cfgid=InstrumentConfigurationID         FROM         Core.ValueConfiguration         WHERE
ValueConfigurationID=@ValueConfigurationID
        SELECT  @rc=InstrumentConfigurationID,  @dt1=FirstInstDate,  @dt2=LastInstDate,  @nr=NumIntRecords  FROM
Core.InstrumentSurveyInstInfo WHERE InstrumentConfigurationID=@cfgid

        IF(@rc >= 0)
                BEGIN
                        -- aggiorna il record scegliendo i valori giusti
                        IF(@dt1 > @InstDate)
                                SET @dt1=@InstDate

                        IF(@dt2 < @InstDate)
                                SET @dt2=@InstDate

                        UPDATE    Core.InstrumentSurveyInstInfo    SET    FirstInstDate=@dt1    ,LastInstDate=@dt2
,NumIntRecords= @nr +1
                        WHERE InstrumentConfigurationID=@cfgid
                END
        ELSE
                -- inserisce il nuovo record
                INSERT INTO Core.InstrumentSurveyInstInfo
                (
                        InstrumentConfigurationID,
                        FirstInstDate,
                        LastInstDate,
                        NumIntRecords
                )
                VALUES
                (
                        @cfgid,
                        @InstDate,
                        @InstDate,
                        1
                )
END'

EXEC dbo.sp_executesql @statement = N'
ALTER PROCEDURE [Core].[spInstrumentSurvey_Add]
        -- Add the parameters for the stored procedure here
        @FactorySerialNumber varchar(20),
        @ConfigurationDate datetime,
        @InstrumentPrefix varchar(15),
        @LastStoredValueDate datetime,
        @NewId int output
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        INSERT INTO [Core].InstrumentSurvey (FactorySerialNumber,ConfigurationDate,
InstrumentPrefix,LastStoredValueDate)
        VALUES(@FactorySerialNumber,@ConfigurationDate, @InstrumentPrefix,@LastStoredValueDate)

        --recupera l''id
        SET @NewId= SCOPE_IDENTITY()

        --aggiunge il nuovo record nella nuova tabella
        INSERT INTO Core.InstrumentSurveyInstInfo (InstrumentConfigurationID, FirstInstDate, LastInstDate,
NumIntRecords)
        VALUES (@NewId,''2100-01-01'',''1900-01-01'',''0'')

END'
GO
--------------------------------------------------------------------------------------
-- CreateAddInsFrom20602  (versione 2.7)
-- Aggiunge il supporto ai Gruppi
-------------------------------------------------------------------------------------- */
/****** Object:  Table [Viewer].[CustomGroup] ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Viewer].[CustomGroup]') AND type in
(N'U'))
BEGIN
CREATE TABLE [Viewer].[CustomGroup](
        [IdGroup] [int] IDENTITY(1,1) NOT NULL,
        [Name] [nvarchar](100) NOT NULL,
 CONSTRAINT [PK_Group] PRIMARY KEY CLUSTERED
(
        [IdGroup] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS  = ON) ON [PRIMARY]
) ON [PRIMARY]
END
GO

/****** Object:  Table [Viewer].[CustomGroupRegistry]    Script Date: 04/13/2011 14:02:33 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Viewer].[CustomGroupRegistry]') AND type
in (N'U'))
BEGIN
CREATE TABLE [Viewer].[CustomGroupRegistry](
        [Id] [int] IDENTITY(1,1) NOT NULL,
        [IdGroup] [int] NOT NULL,
        [FactorySerialNumber] [varchar](20) NOT NULL,
 CONSTRAINT [PK_GroupRegistry] PRIMARY KEY CLUSTERED
(
        [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS  = ON) ON [PRIMARY]
) ON [PRIMARY]
ALTER TABLE [Viewer].[CustomGroupRegistry] WITH CHECK ADD CONSTRAINT [FK_CustomGroupRegistry_CustomGroup]
FOREIGN KEY([IdGroup])
REFERENCES [Viewer].[CustomGroup] ([IdGroup])
ON UPDATE CASCADE
ON DELETE CASCADE
ALTER TABLE [Viewer].[CustomGroupRegistry] CHECK CONSTRAINT [FK_CustomGroupRegistry_CustomGroup]
END
GO
```

```
/****** Object:  StoredProcedure [Viewer].[spCustomGroup_GetAll]    Script Date: 04/13/2011 14:02:32 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Viewer].[spCustomGroup_GetAll]') AND type
in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [Viewer].[spCustomGroup_GetAll]
        -- Add the parameters for the stored procedure here
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT IdGroup, Name
        FROM Viewer.CustomGroup
END'
END
GO

/****** Object:  StoredProcedure [Viewer].[spCustomGroup_RemoveAll]    Script Date: 04/13/2011 14:02:32 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Viewer].[spCustomGroup_RemoveAll]') AND
type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [Viewer].[spCustomGroup_RemoveAll]
        -- Add the parameters for the stored procedure here
AS
BEGIN

    -- Insert statements for procedure here
        DELETE  FROM [Viewer].CustomGroup

END'
END
GO

/****** Object:  StoredProcedure [Viewer].[spCustomGroup_Add]    Script Date: 04/13/2011 14:02:32 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Viewer].[spCustomGroup_Add]') AND type in
(N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [Viewer].[spCustomGroup_Add]
        -- Add the parameters for the stored procedure here
        @Name nvarchar(100),
        @NewId int output
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        INSERT INTO [Viewer].CustomGroup (Name)
        VALUES(@Name)

        --recupera id
        SET @NewId= SCOPE_IDENTITY()

END'
END
GO

/****** Object:  StoredProcedure [Viewer].[spCustomGroupRegistry_Add]    Script Date: 04/13/2011 14:02:32 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Viewer].[spCustomGroupRegistry_Add]') AND
type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [Viewer].[spCustomGroupRegistry_Add]
        -- Add the parameters for the stored procedure here
        @IdGroup int,
        @FactorySerialNumber varchar(20),
        @NewId int output
```

```
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        INSERT INTO [Viewer].CustomGroupRegistry (IdGroup, FactorySerialNumber)
        VALUES(@IdGroup,@FactorySerialNumber)

        --recupera id
        SET @NewId= SCOPE_IDENTITY()

END'
END
GO

/****** Object:  StoredProcedure [Viewer].[spCustomGroup_GetFactoryMatrsFromIdGroup]    Script Date: 04/13/2011
14:02:32 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[Viewer].[spCustomGroup_GetFactoryMatrsFromIdGroup]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [Viewer].[spCustomGroup_GetFactoryMatrsFromIdGroup]
        -- Add the parameters for the stored procedure here
        @IdGroup int
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT FactorySerialNumber
        FROM Viewer.CustomGroupRegistry
        WHERE IdGroup=@IdGroup
END'
END
GO

-------------------------------------------------------------------------------------
-- Add fixed data
-- (2.4.0) aggiunti i record di supporto per RLog e RComm
-- (2.11.0) modificate le elaborazioni per RisDir e RisVel
------------------------------------------------------------------------------------- */
INSERT INTO [Gidas].[Core].[InstrumentType] ([Prefix] ,[Name]) VALUES ('BabucABC' ,'Babuc ABC')
INSERT INTO [Gidas].[Core].[InstrumentType] ([Prefix] ,[Name]) VALUES ('BabucM' ,'Babuc A/M')
INSERT INTO [Gidas].[Core].[InstrumentType] ([Prefix] ,[Name]) VALUES ('ELog' ,'E-Log')
INSERT INTO [Gidas].[Core].[InstrumentType] ([Prefix] ,[Name]) VALUES ('RLog' ,'R-Log')
INSERT INTO [Gidas].[Core].[InstrumentType] ([Prefix] ,[Name]) VALUES ('RComm' ,'R-Comm')
INSERT INTO [Gidas].[Core].[ElabTypeList] ([IdElabType],[ElabTypeString],[ElabTypeToDo]) VALUES (-1,'Inst
(n.e.)','Ave')
INSERT INTO [Gidas].[Core].[ElabTypeList] ([IdElabType],[ElabTypeString],[ElabTypeToDo]) VALUES
(0,'Nothing','Nothing')
INSERT INTO [Gidas].[Core].[ElabTypeList] ([IdElabType],[ElabTypeString],[ElabTypeToDo]) VALUES (1,'Inst','Ave')
INSERT INTO [Gidas].[Core].[ElabTypeList] ([IdElabType],[ElabTypeString],[ElabTypeToDo]) VALUES (2,'Min','Min')
INSERT INTO [Gidas].[Core].[ElabTypeList] ([IdElabType],[ElabTypeString],[ElabTypeToDo]) VALUES (3,'Ave','Ave')
INSERT INTO [Gidas].[Core].[ElabTypeList] ([IdElabType],[ElabTypeString],[ElabTypeToDo]) VALUES (4,'Max','Max')
INSERT INTO [Gidas].[Core].[ElabTypeList] ([IdElabType],[ElabTypeString],[ElabTypeToDo]) VALUES
(5,'StDev','Nothing')
INSERT INTO [Gidas].[Core].[ElabTypeList] ([IdElabType],[ElabTypeString],[ElabTypeToDo]) VALUES (6,'Tot','Sum')
INSERT INTO [Gidas].[Core].[ElabTypeList] ([IdElabType],[ElabTypeString],[ElabTypeToDo]) VALUES
(7,'Duration','Sum')
INSERT INTO [Gidas].[Core].[ElabTypeList] ([IdElabType],[ElabTypeString],[ElabTypeToDo]) VALUES
(8,'PrevDir','AvgDirection')
INSERT INTO [Gidas].[Core].[ElabTypeList] ([IdElabType],[ElabTypeString],[ElabTypeToDo]) VALUES
(9,'RisDir','VectDirection')
INSERT INTO [Gidas].[Core].[ElabTypeList] ([IdElabType],[ElabTypeString],[ElabTypeToDo]) VALUES
(10,'RisVel','VectVel')
INSERT INTO [Gidas].[Core].[ElabTypeList] ([IdElabType],[ElabTypeString],[ElabTypeToDo]) VALUES
(11,'StdDevDir','Nothing')
INSERT INTO [Gidas].[Core].[ElabTypeList] ([IdElabType],[ElabTypeString],[ElabTypeToDo]) VALUES
(12,'CalmPerc','Ave')
INSERT INTO [Gidas].[Core].[ElabTypeList] ([IdElabType],[ElabTypeString],[ElabTypeToDo]) VALUES
(13,'ValidDataPerc','Ave')
INSERT INTO Core.AddIns (AddInCode, [Description], AttachedDate) VALUES ('AdminUser', 'Administrator Gidas
User',GETDATE())
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('14','PSBisect','AvgDirection')
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('15','PSPrevDir','AvgDirection')
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('16','PSPrevVel','Ave')
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('17','PSStdDevDir','Nothing')
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('18','DirFreq1','Ave')
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('19','DirFreq2','Ave')
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('20','DirFreq3','Ave')
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('21','DirFreq4','Ave')
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('22','DirFreq5','Ave')
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('23','DirFreq6','Ave')
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('24','DirFreq7','Ave')
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('25','DirFreq8','Ave')
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('26','DirFreq9','Ave')
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('27','DirFreq10','Ave')
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('28','DirFreq11','Ave')
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('29','DirFreq12','Ave')
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('30','DirFreq13','Ave')
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('31','DirFreq14','Ave')
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('32','DirFreq15','Ave')
INSERT INTO Core.ElabTypeList (IdElabType,ElabTypeString, ElabTypeToDo) VALUES ('33','DirFreq16','Ave')

--------------------------------------------------------------------------------------
-- (2.6.2) aggiunge il supporto a S-Log
------------------------------------------------------------------------------------- */
INSERT INTO [Gidas].[Core].[InstrumentType] ([Prefix] ,[Name]) VALUES ('SLog' ,'S-Log')


--------------------------------------------------------------------------------------
-- (2.9) aggiunge il supporto alle descrizioni
------------------------------------------------------------------------------------- */

/*inserisce la nuov sp che gestisce la configurazione */
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spInstrumentRegistry_Add291]') AND
type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [Core].[spInstrumentRegistry_Add291]
        -- Add the parameters for the stored procedure here
                @FactorySerialNumber varchar(20),
                @InstrumentPrefix varchar(15),
                @UserSerialNumber varchar(50),
                @Description varchar(200),
                @DbDescription varchar(200)
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Prima dell''inserimento verifica se esiste già il record
        SELECT FactorySerialNumber  FROM Core.InstrumentRegistry  WHERE FactorySerialNumber=@FactorySerialNumber

    -- Insert statements for procedure here
        IF (@@ROWCOUNT <= 0)
        BEGIN
                INSERT INTO Core.InstrumentRegistry
                (
                        FactorySerialNumber,
                        InstrumentPrefix,
                        UserSerialNumber,
                        Description,
                        DbDescription
                )
                VALUES
                (
                        @FactorySerialNumber,
                        @InstrumentPrefix,
                        @UserSerialNumber,
                        @Description,
                        @DbDescription
                )
        END
END'
END

IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[Core].[spInstrumentRegistry_ChangeDescription]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [Core].[spInstrumentRegistry_ChangeDescription]
        -- Add the parameters for the stored procedure here
        @FactorySerialNumber varchar(20),
        @DbDescription varchar(200)
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

        UPDATE Core.InstrumentRegistry SET  DbDescription=@DbDescription
        WHERE FactorySerialNumber=@FactorySerialNumber
END'
END

IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[Core].[spInstrumentRegistry_ChangeDescriptionAndSerialNumber]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [Core].[spInstrumentRegistry_ChangeDescriptionAndSerialNumber]
```

```
        -- Add the parameters for the stored procedure here
        @FactorySerialNumber varchar(20),
        @Description varchar(200),
        @UserSerialNumber varchar(50)
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

        UPDATE Core.InstrumentRegistry SET  [Description]=@Description, UserSerialNumber=   @UserSerialNumber

        WHERE FactorySerialNumber=@FactorySerialNumber
END'
END

---------------------------------------------------------------------------------------
-- (2.10) aggiunge la gestione del DbDescription per le UserConfigurationAttributes
------------------------------------------------------------------------------------ */
/* aggiunge il nuovo campo alla tabella */
if not exists (select * from syscolumns where id=object_id('Core.UserConfigurationAttributes') and
name='DbDescription')
BEGIN
EXEC dbo.sp_executesql @statement = N'
ALTER TABLE Core.UserConfigurationAttributes ADD DbDescription varchar(200)'
END

/* modifica le sp che agiscono sulla tabella */
IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spUserConfigurationAttributes_GetAll]')
AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
ALTER PROCEDURE [Core].[spUserConfigurationAttributes_GetAll]
        -- Add the parameters for the stored procedure here
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT InstrumentConfigurationID, Name, [Description],Code, DbDescription FROM
Core.UserConfigurationAttributes

END'
END

IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spUserConfigurationAttributes_Add]')
AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
ALTER PROCEDURE [Core].[spUserConfigurationAttributes_Add]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationId int,
        @Name nvarchar(100),
        @Description nvarchar(200),
        @Code nvarchar(25),
        @DbDescription nvarchar(200)=null
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

                -- per i writer che non la supportano
        if @DbDescription IS NULL
        BEGIN
                INSERT INTO Core.UserConfigurationAttributes (InstrumentConfigurationID, Name,
[Description],Code)
                VALUES(@InstrumentConfigurationID, @Name, @Description,@Code)
                RETURN
        END

     -- procedura aggiornata alla versione 2.9.3 del db
        INSERT INTO Core.UserConfigurationAttributes (InstrumentConfigurationID, Name,
[Description],Code,DbDescription)
        VALUES(@InstrumentConfigurationID, @Name, @Description,@Code,@DbDescription)

END
'
END

IF EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[Core].[spUserConfigurationAttributes_GetByConfigurationID]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
ALTER PROCEDURE [Core].[spUserConfigurationAttributes_GetByConfigurationID]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationID int
AS
```

```
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SELECT InstrumentConfigurationID, Name, [Description],Code, DbDescription FROM
Core.UserConfigurationAttributes
        WHERE InstrumentConfigurationID=@InstrumentConfigurationID

END
'
END

IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[Core].[spUserConfigurationAttributes_Update]')
AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
ALTER PROCEDURE [Core].[spUserConfigurationAttributes_Update]
        -- Add the parameters for the stored procedure here
        @InstrumentConfigurationID int,
        @Name nvarchar(100),
        @Description nvarchar(200),
        @Code nvarchar(25),
        @DbDescription nvarchar(200)=null

AS
BEGIN

        -- per i writer che non la supportano
        if @DbDescription IS NULL
        BEGIN
                UPDATE Core.UserConfigurationAttributes SET Name=@Name, [Description]=@Description, Code=@Code
                WHERE InstrumentConfigurationID=@InstrumentConfigurationID
                RETURN
        END

     -- procedura aggiornata alla versione 2.9.3 del db
        UPDATE Core.UserConfigurationAttributes SET Name=@Name, [Description]=@Description, Code=@Code,
DbDescription=@DbDescription
        WHERE InstrumentConfigurationID=@InstrumentConfigurationID
END'
END

--------------------------------------------------------------------------------------
-- Versione installazione
------------------------------------------------------------------------------------- */
INSERT INTO [Gidas].[Core].[GidasVersion] ([Version],[ReleaseDate]) VALUES('2.11.0' ,'2013-03-04T00:00:00')
INSERT INTO Core.AddIns (AddInCode, [Description],  AttachedDate) VALUES ('CreatedVers_2.11.0', 'Database creation
script v2.11.0',GETDATE())
GO
```

6.3. Utenti e Login:

```
--------------------------------------------------------------------------------------
-- Create database logins
--------------------------------------------------------------------------------------
IF NOT EXISTS (SELECT * FROM syslogins WHERE [name]='LSI.Gidas.Reader')
BEGIN
        CREATE LOGIN [LSI.Gidas.Reader] WITH PASSWORD=N'redaer_6', DEFAULT_DATABASE=[Gidas],
DEFAULT_LANGUAGE=[us_english], CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF
END
GO
IF NOT EXISTS (SELECT * FROM syslogins WHERE [name]='LSI.Gidas.Writer')
BEGIN
        CREATE LOGIN [LSI.Gidas.Writer] WITH PASSWORD=N'retirw_6', DEFAULT_DATABASE=[Gidas],
DEFAULT_LANGUAGE=[us_english], CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF
END
GO
IF NOT EXISTS (SELECT * FROM syslogins WHERE [name]='LSI.Gidas.Administrator')
BEGIN
        CREATE LOGIN [LSI.Gidas.Administrator] WITH PASSWORD=N'sadig_admin',
DEFAULT_DATABASE=[Gidas], DEFAULT_LANGUAGE=[us_english], CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF
END
GO

--------------------------------------------------------------------------------------
-- Delete and recreate Database Users
--------------------------------------------------------------------------------------
USE [Gidas]

IF EXISTS (SELECT * FROM sys.database_principals WHERE name = N'LSI.Gidas.Writer')
```

```
       DROP USER [LSI.Gidas.Writer]
GO
IF EXISTS (SELECT * FROM sys.database_principals WHERE name = N'LSI.Gidas.Reader')
       DROP USER [LSI.Gidas.Reader]
GO
IF EXISTS (SELECT * FROM sys.database_principals WHERE name = N'LSI.Gidas.Administrator')
       DROP USER [LSI.Gidas.Administrator]
GO

IF NOT EXISTS (SELECT * FROM sys.database_principals WHERE name = N'LSI.Gidas.Writer')
CREATE USER [LSI.Gidas.Writer] FOR LOGIN [LSI.Gidas.Writer] WITH DEFAULT_SCHEMA=[Core]
GO
IF NOT EXISTS (SELECT * FROM sys.database_principals WHERE name = N'LSI.Gidas.Reader')
CREATE USER [LSI.Gidas.Reader] FOR LOGIN [LSI.Gidas.Reader] WITH DEFAULT_SCHEMA=[Core]
GO
IF NOT EXISTS (SELECT * FROM sys.database_principals WHERE name = N'LSI.Gidas.Administrator')
CREATE USER [LSI.Gidas.Administrator] FOR LOGIN [LSI.Gidas.Administrator] WITH DEFAULT_SCHEMA=[Core]
GO
GRANT EXECUTE, INSERT, DELETE, SELECT, UPDATE ON SCHEMA :: Core TO [LSI.Gidas.Writer]
GO
GRANT EXECUTE, INSERT, DELETE, SELECT, UPDATE ON SCHEMA :: Viewer TO [LSI.Gidas.Writer]
GO
GRANT SELECT ON SCHEMA :: Core TO [LSI.Gidas.Reader]
GO
GRANT SELECT ON SCHEMA :: Viewer TO [LSI.Gidas.Reader]
GO
EXEC sp_addrolemember N'db_owner', N'LSI.Gidas.Administrator'
GO
```