

# eICU Análisis de datos(MDX)



Álvaro González Plaza

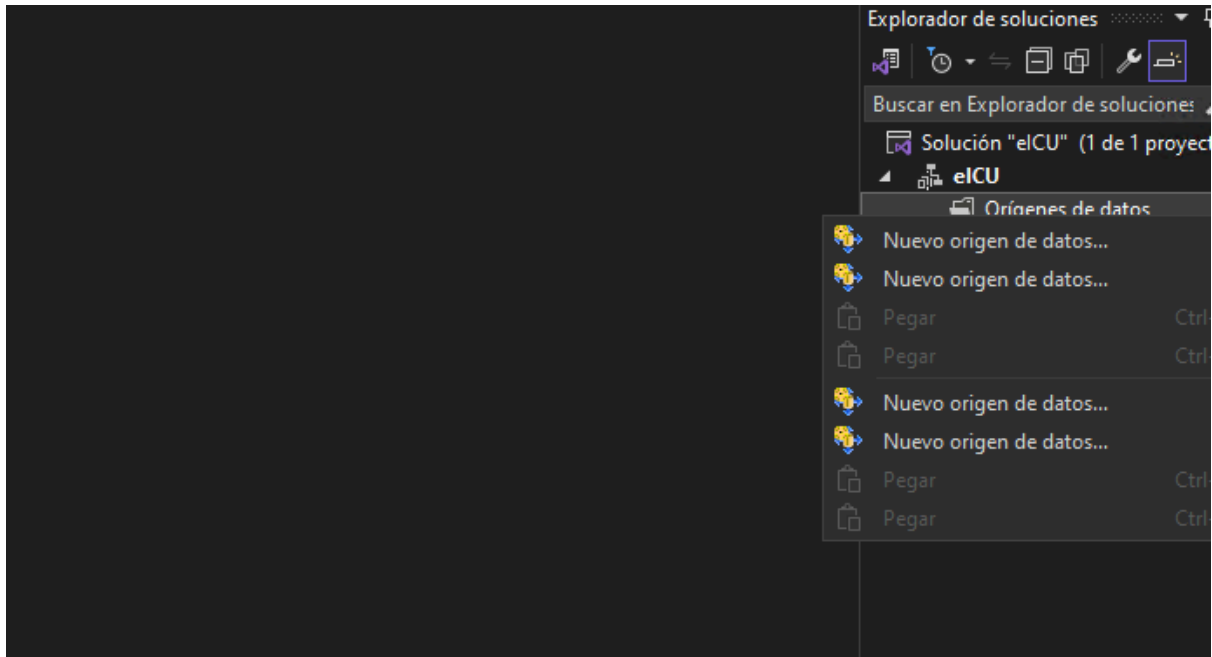
# ÍNDICE

<b>PROCEDIMIENTO</b>	<b>3</b>
DIMENSIONES Y JERARQUÍAS	9
DESPLIEGUE EN SQL SERVER	17
<b>CONSULTAS</b>	<b>18</b>
<b>PROBLEMAS ENCONTRADOS</b>	<b>26</b>
<b>INSTRUCCIONES PARA EJECUCIÓN</b>	<b>27</b>

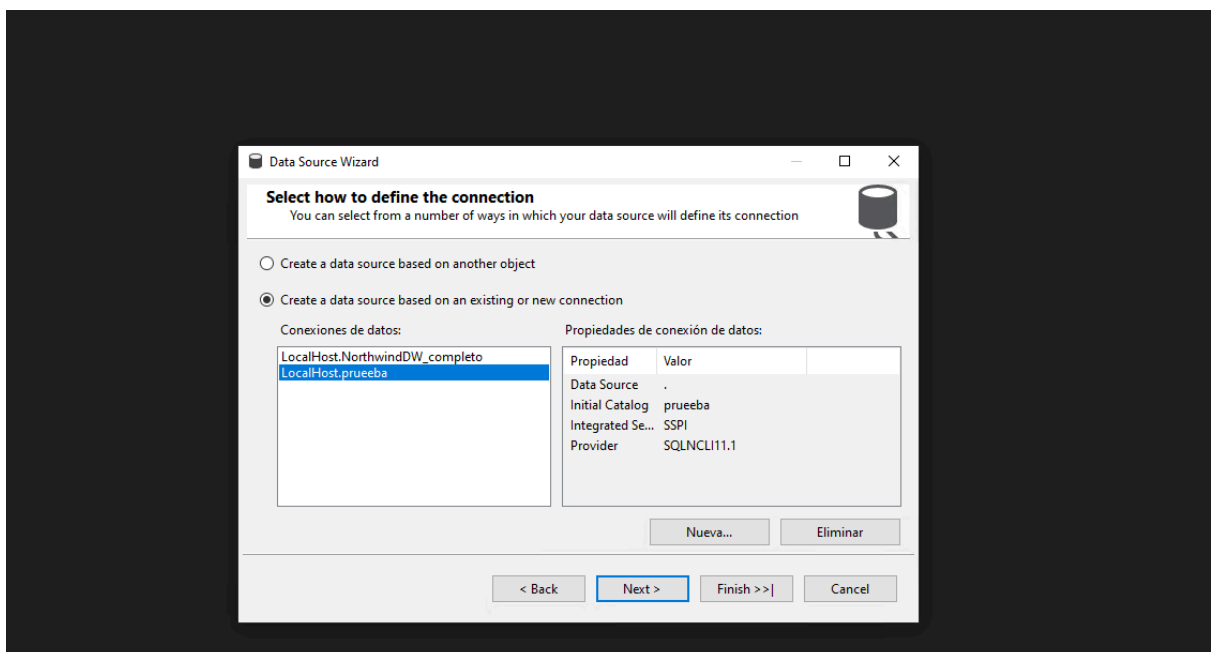
# PROCEDIMIENTO

Una vez que los datos han sido cargados en nuestro almacén, debemos crear un nuevo proyecto de **analysis services**, en mi caso he utilizado visual studio.

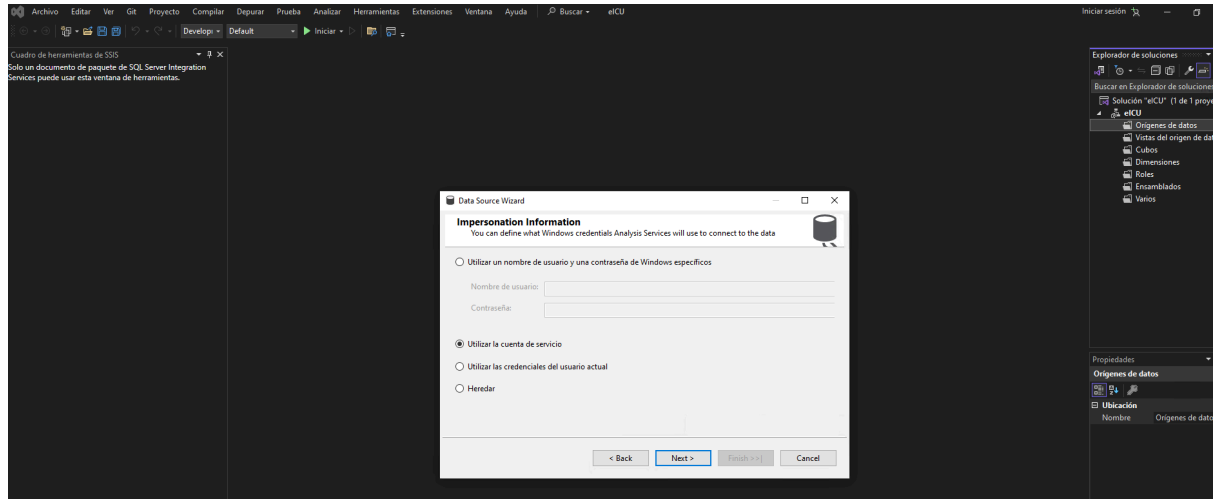
Una vez creado creamos un nuevo origen de datos en nuestro proyecto tal y como podemos observar en la imagen:



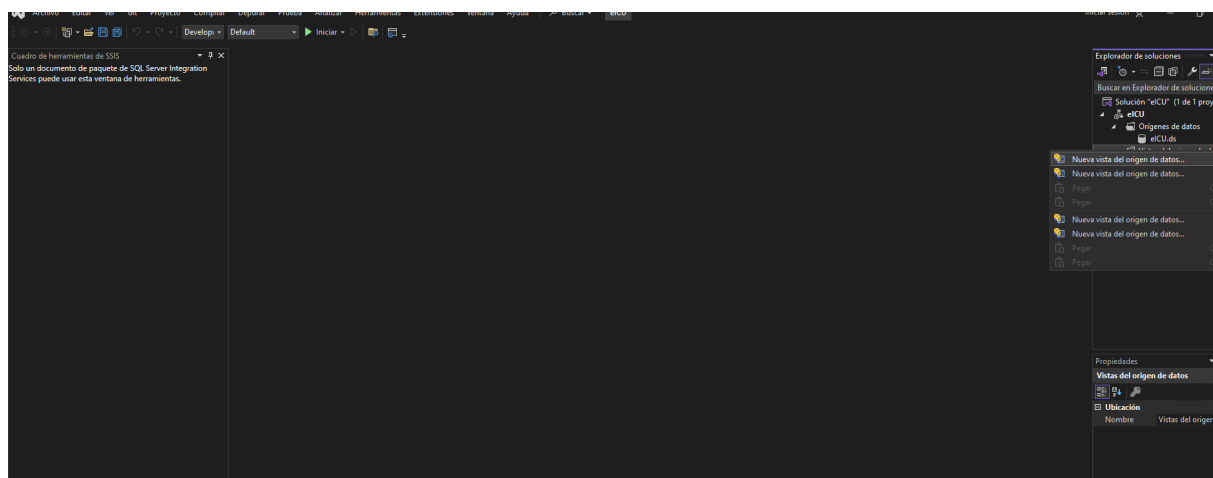
A continuación creamos una nueva conexión con la fuente de datos:



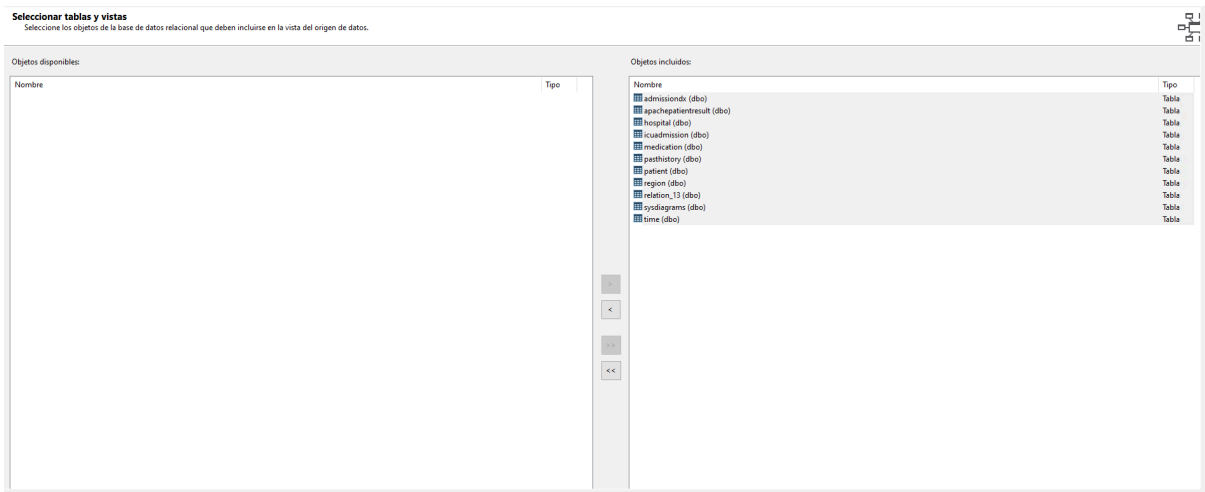
Y en este apartado marcamos la opción de utilizar la cuenta de servicio. Aunque es posible que más adelante a medida que vayamos trabajando nos pueda aparecer un error relacionado con la conexión, lo único que tendremos que hacer es editar la conexión, marcando conexión a sql e introduciendo las credenciales pertinentes.



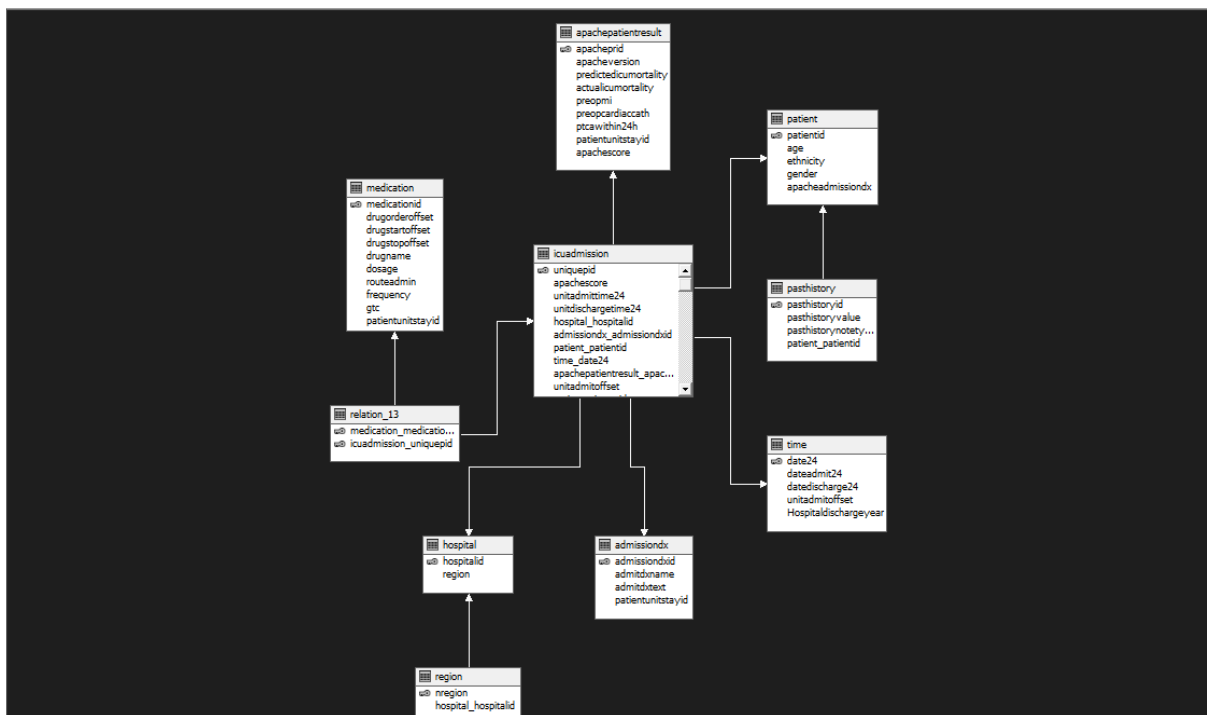
Una vez creado el origen de datos, crearemos una nueva vista de origen de datos tal y como se puede observar en la imagen:



Se nos pedirá que indiquemos cuáles son las tablas y vistas que deseamos utilizar de entre los disponibles, aunque en este caso utilizaremos todas las tablas:



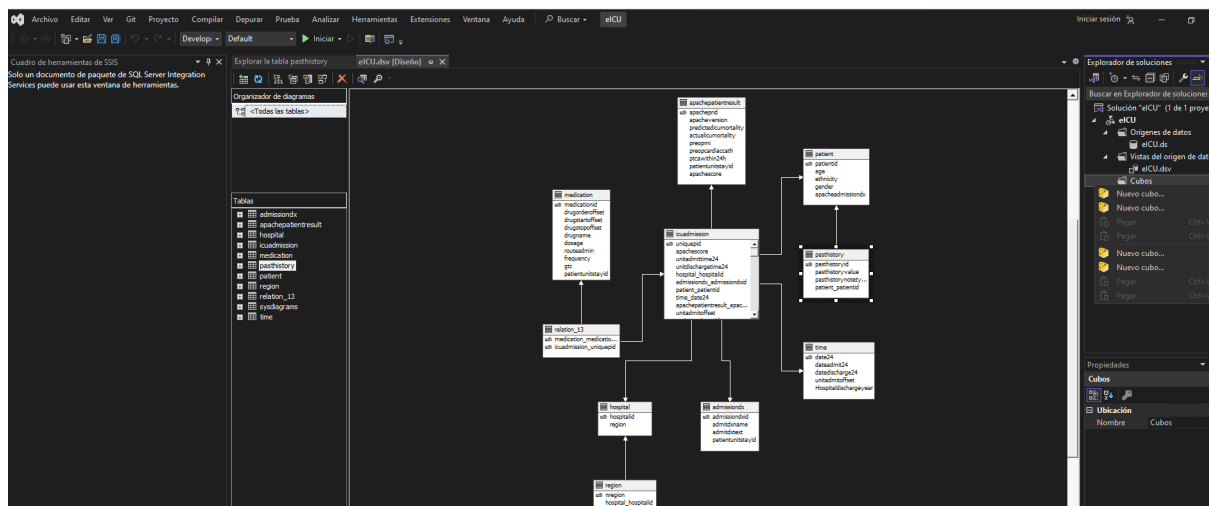
La **vista del origen de datos** se vería así, donde podemos observar el hecho y todas sus dimensiones alrededor:



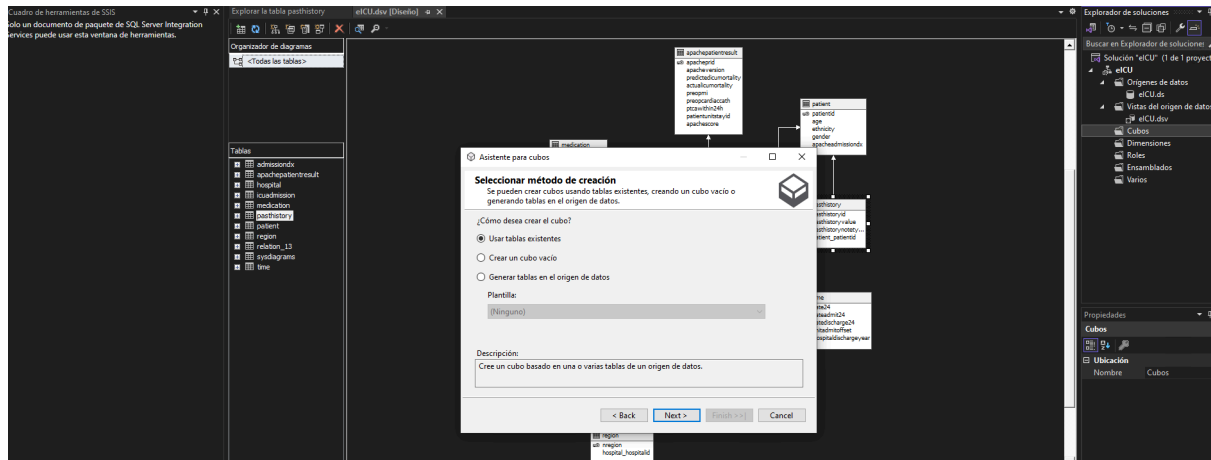
De **manera preventiva** exploraré los datos de cualquiera de las tablas para comprobar que efectivamente los datos están cargados:

Explorar la tabla pasthistory - eICU.dsv [Diseño]			
Tabla			
pasthistoryid	pasthistoryvalue	pasthistorynotetype	patient_patientid
848035	Performed	Comprehensive Progress	157427
848036	insulin dependent diabetes	Comprehensive Progress	157427
848467	Performed	Comprehensive Progress	238463
848468	CHF	Comprehensive Progress	238463
851377	No Health Problems	Comprehensive Progress	174956
851418	No Health Problems	Comprehensive Progress	232447
852162	Performed	Comprehensive Progress	162842
852163	COPD - no limitations	Comprehensive Progress	162842
852164	hypertension requiring treatment	Comprehensive Progress	162842
852165	renal insufficiency - creatinine 1-2	Comprehensive Progress	162842
852166	atrial fibrillation - intermittent	Comprehensive Progress	162842
857319	Performed	Comprehensive Progress	205061
857320	MI - within 2 years	Comprehensive Progress	205061
857321	hypertension requiring treatment	Comprehensive Progress	205061
857322	peripheral vascular disease	Comprehensive Progress	205061
857323	peptic ulcer disease with h/o GI ...	Comprehensive Progress	205061
863231	No Health Problems	Comprehensive Progress	217838
864745	Performed	Comprehensive Progress	223787
864746	hypertension requiring treatment	Comprehensive Progress	223787
864747	MI - date unknown	Comprehensive Progress	223787
864748	COPD - moderate	Comprehensive Progress	223787
864749	prostate	Comprehensive Progress	223787
865684	Performed	Comprehensive Progress	186085
865685	multiple myeloma	Comprehensive Progress	186085
869457	Performed	Comprehensive Progress	203970
869458	hypertension requiring treatment	Comprehensive Progress	203970
871734	Performed	Comprehensive Progress	173458
871735	CHF - class IV	Comprehensive Progress	173458
871736	insulin dependent diabetes	Comprehensive Progress	173458

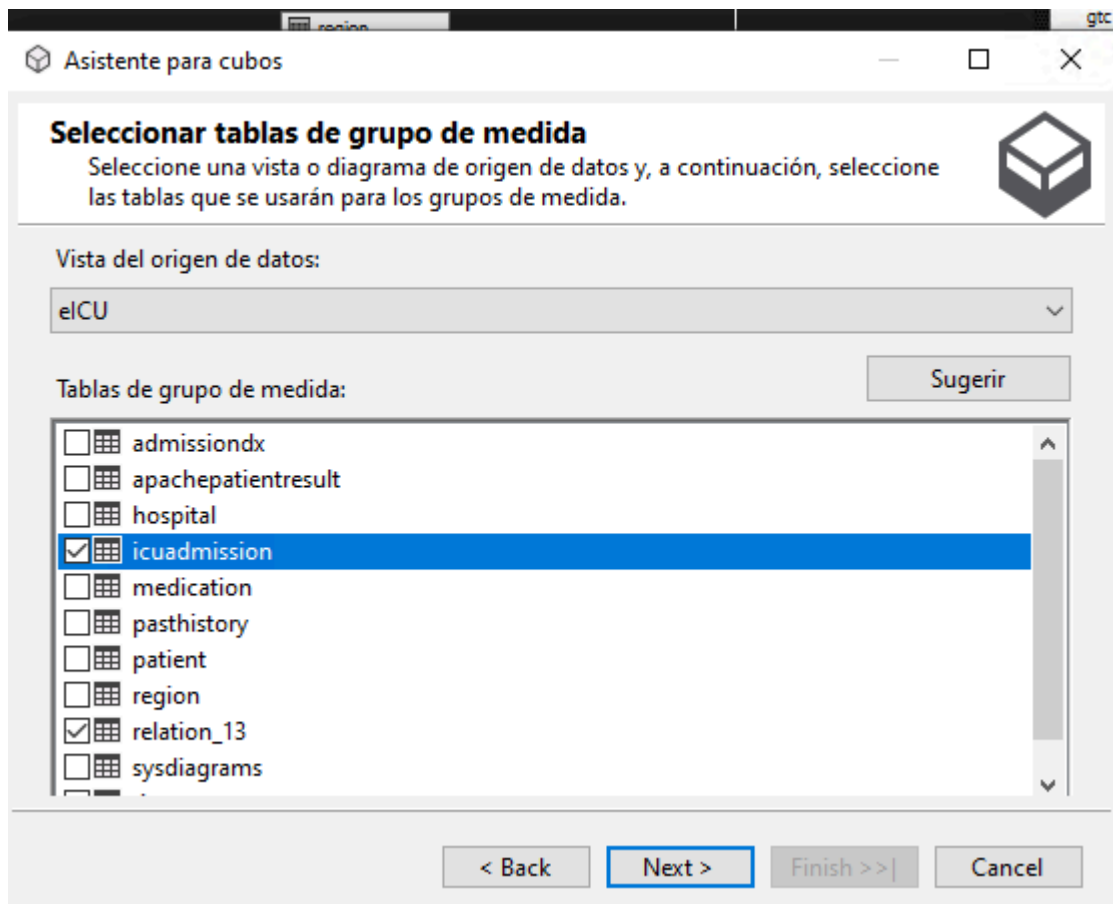
Ahora crearemos el cubo tal y como se puede observar en la imagen:



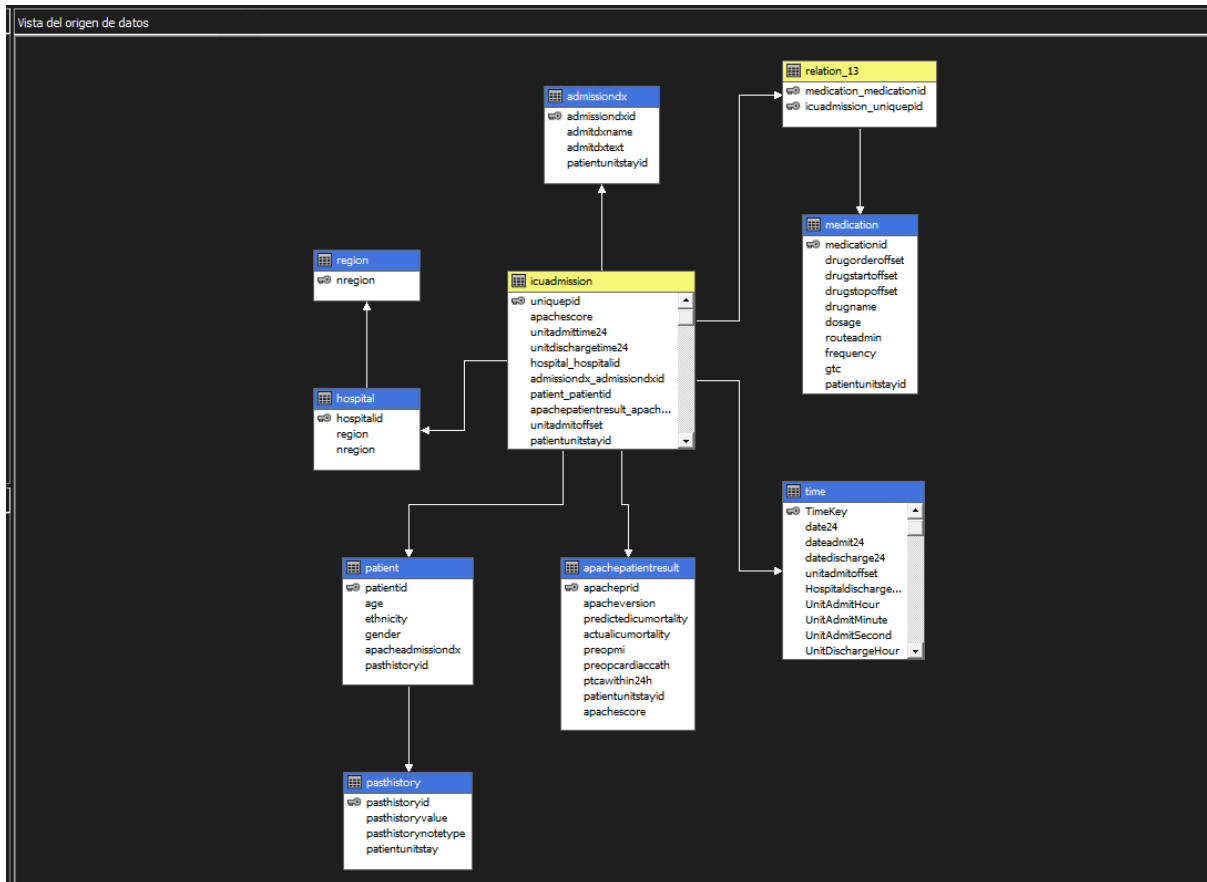
Se nos pedirá inicialmente que seleccionemos el método de creación, en este caso voy a seleccionar **utilizando las tablas ya existentes**:



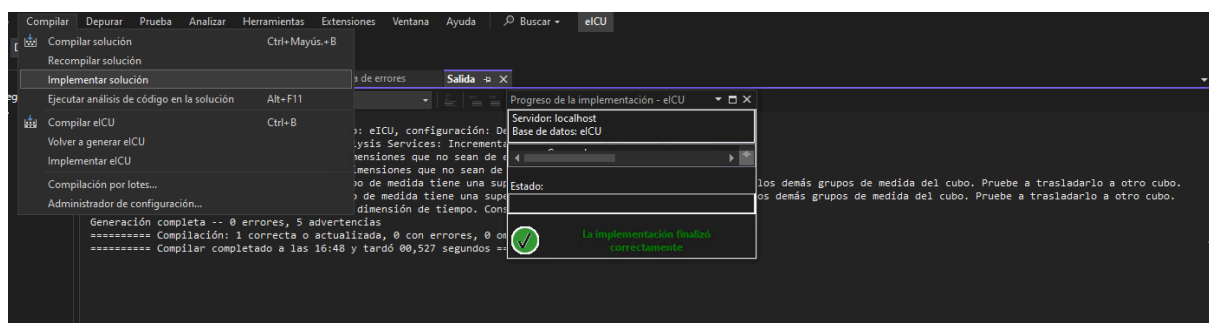
Ahora seleccionamos la tabla de grupo de medida que en nuestro caso es el hecho **icuadmission** y además por motivos ya descritos en la sección [problemas encontrados](#) he añadido la tabla **relation1\_3** correspondiente a la relación M:M entre **icuadmission** y **medication**:



Así luciría la vista del origen de datos tras crear el cubo y seleccionar la tabla de medidas:



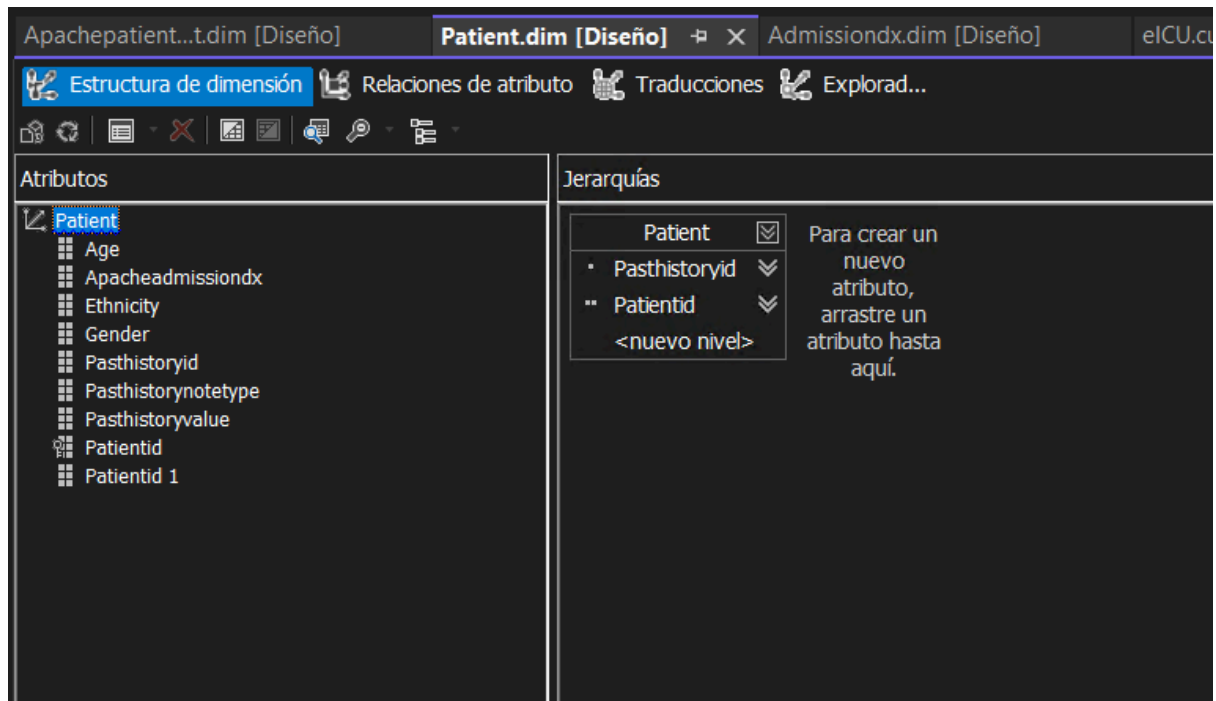
Ahora en la sección **compilar**, implementaremos y compilaremos nuestra solución para obtener el resultado de todo este proceso:



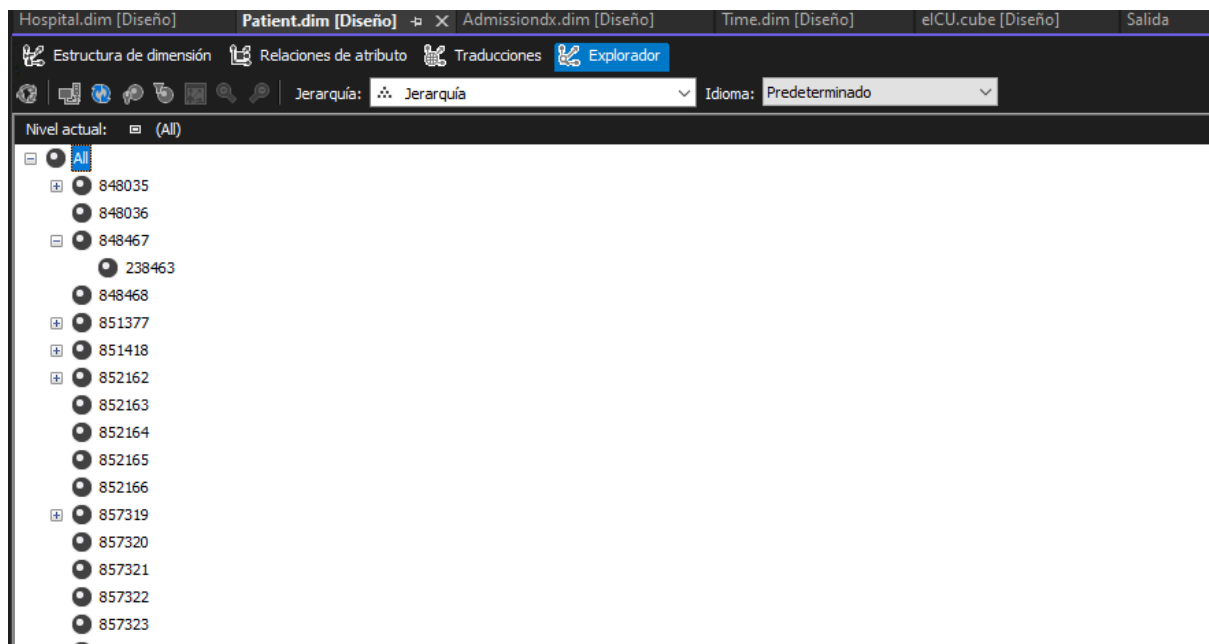


## DIMENSIONES Y JERARQUÍAS

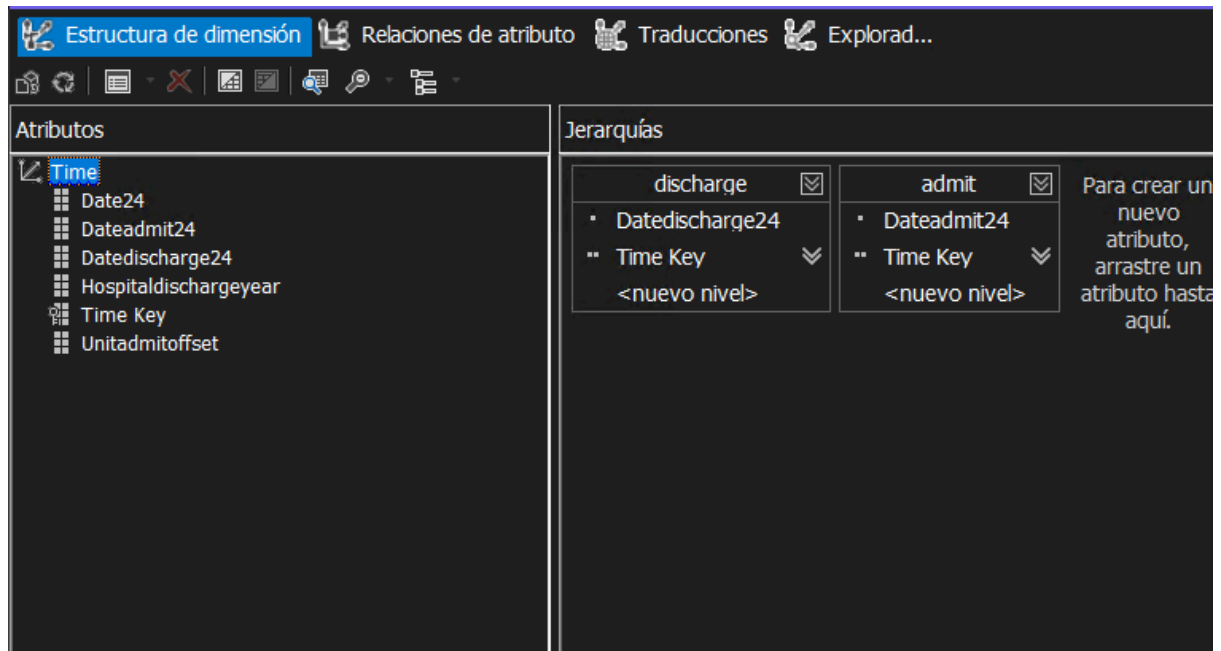
Comenzamos a crear las jerarquías dentro de cada dimensión, en este caso será la dimensión **patient** y la jerarquía de nombre **Patient**:



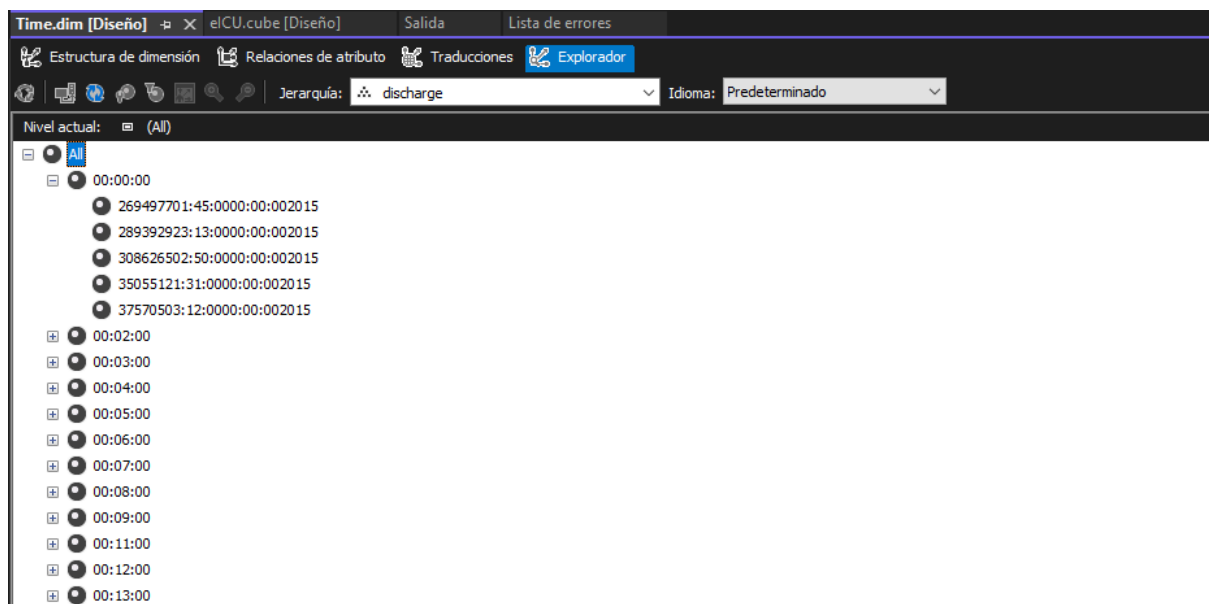
Y finalmente si comprobamos en el **explorador**, luciría así:



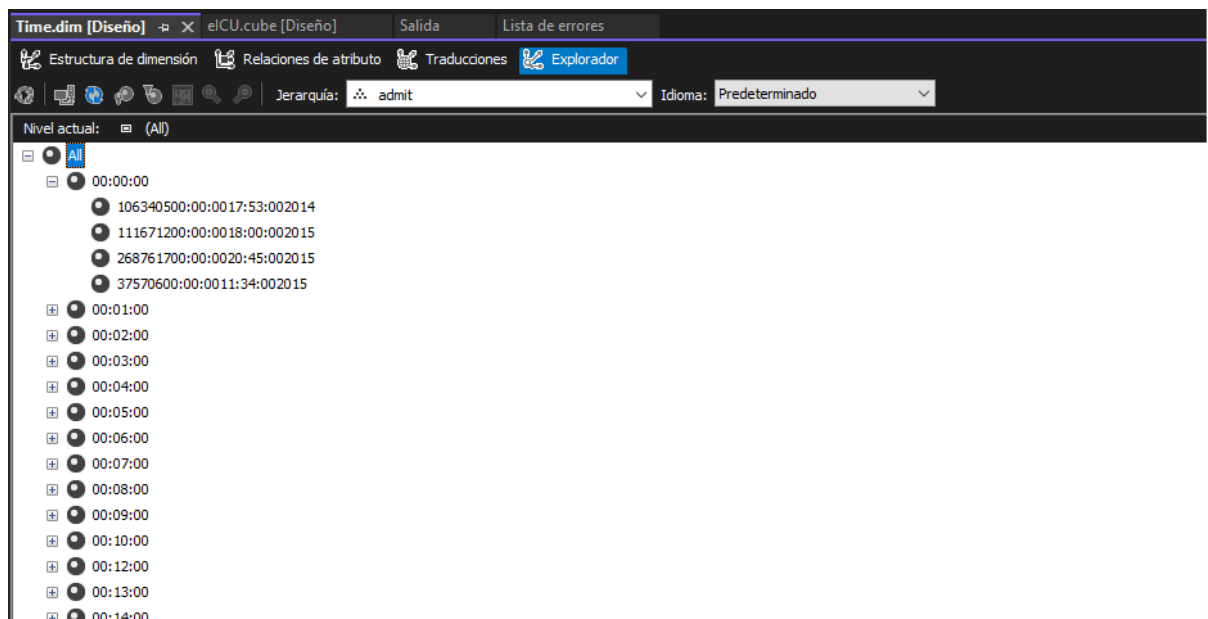
Procedemos ahora con la dimensión **tiempo** y en este caso crearé dos jerarquías, una para admisión y otra para el alta de pacientes:



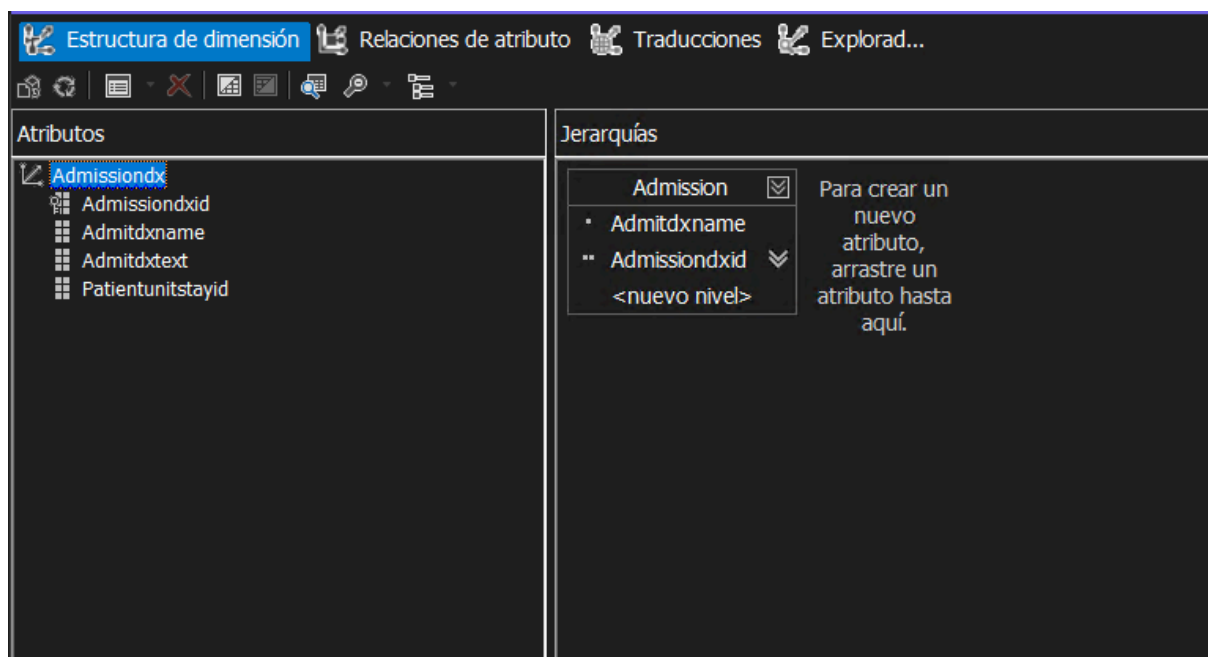
Finalmente en el explorador luciría así para la jerarquía **discharge**:



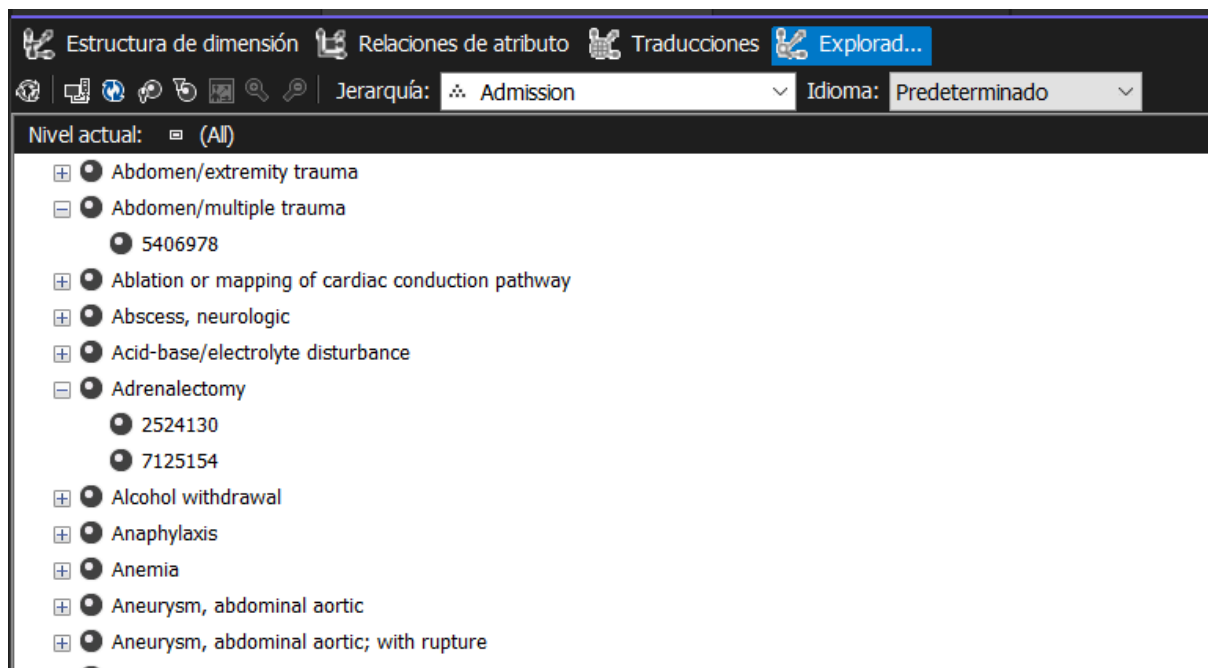
Y así para la jerarquía **admit**:



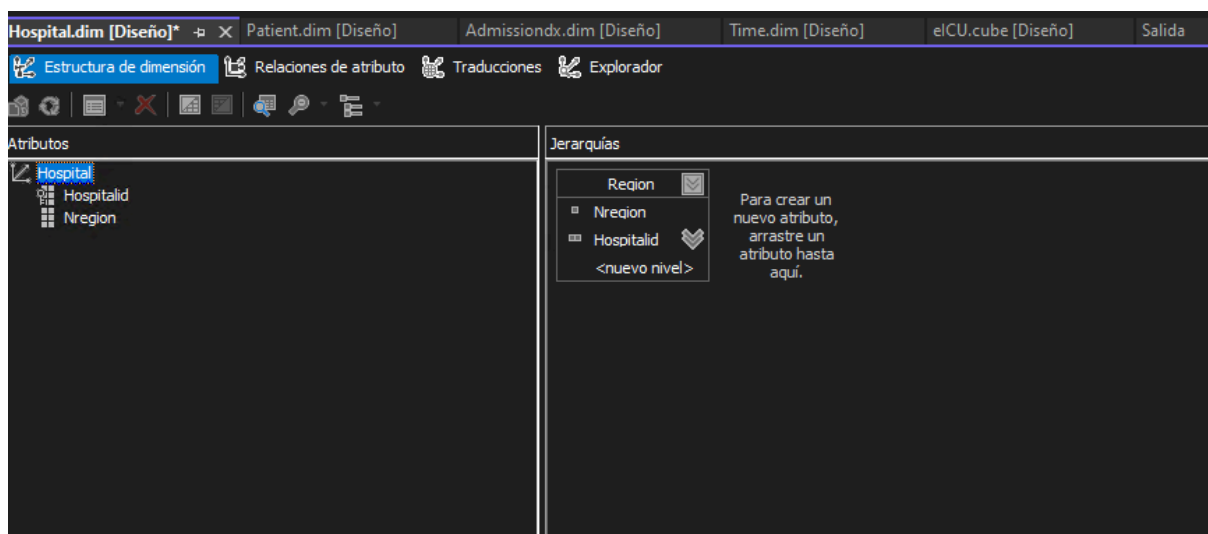
En cuanto a la dimensión **admissiondx** he creado la siguiente jerarquía llamada **admission**:



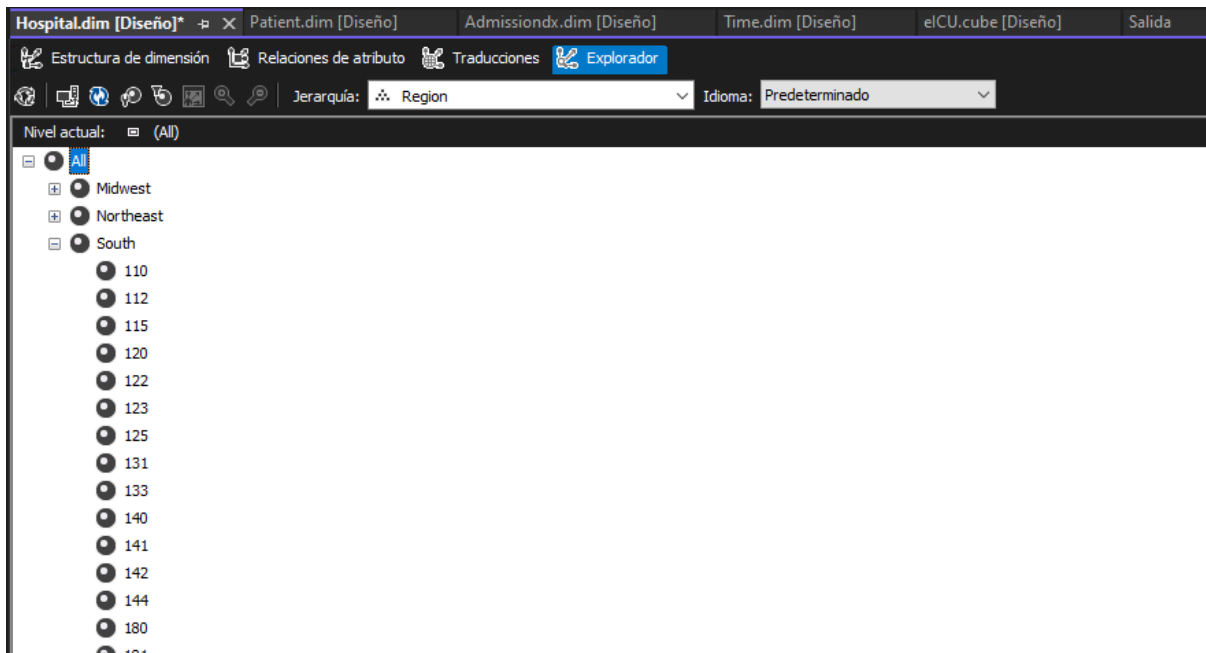
Dando a lugar a este resultado:



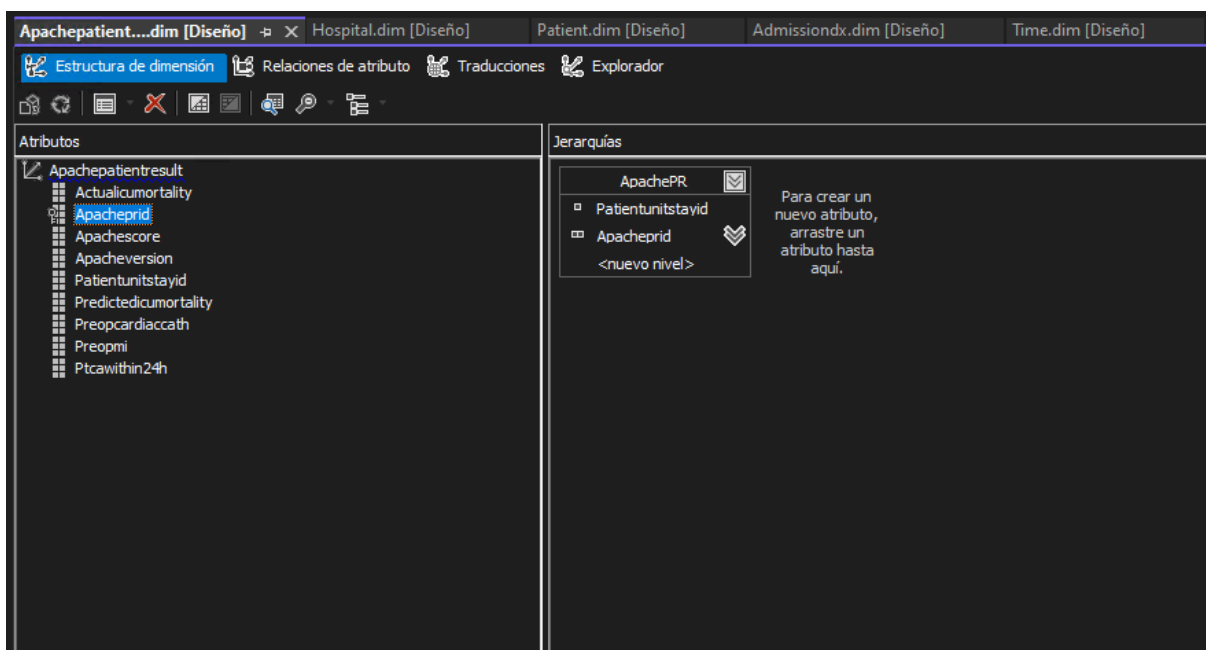
Para la dimensión **hospital**, he creado la siguiente jerarquía llamada **Region**:



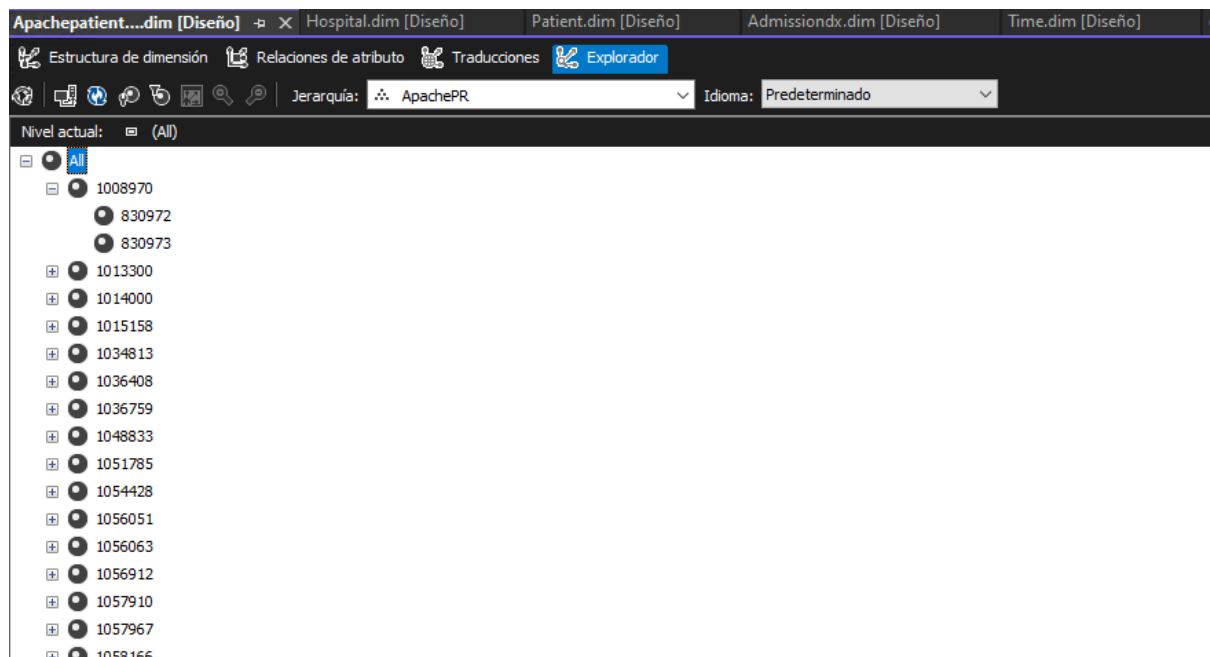
Y si la observamos desde el **explorador** luciría de esta forma:



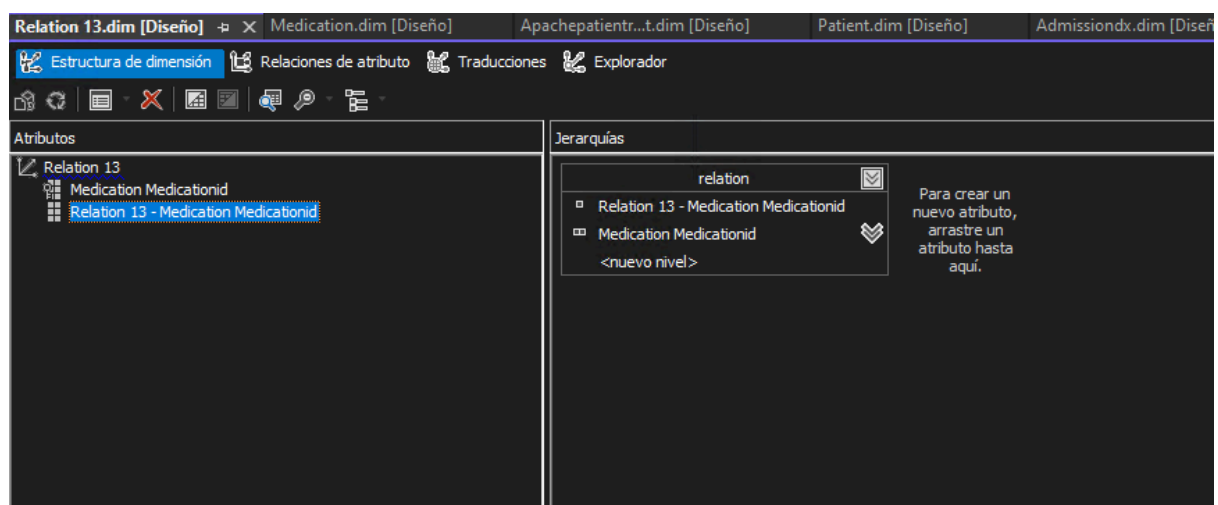
Para la dimensión **Apachepatientresult** la jerarquía sería **ApachePR**:



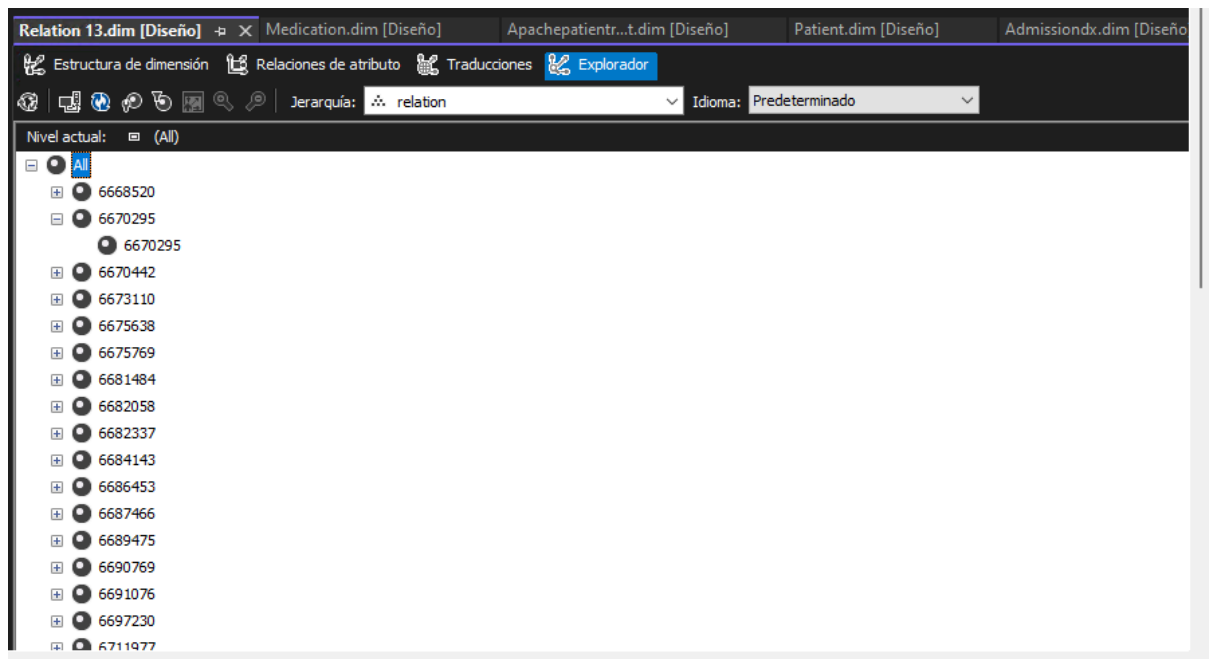
Y que finalmente quedaría tal y como podemos observar en la siguiente imagen:



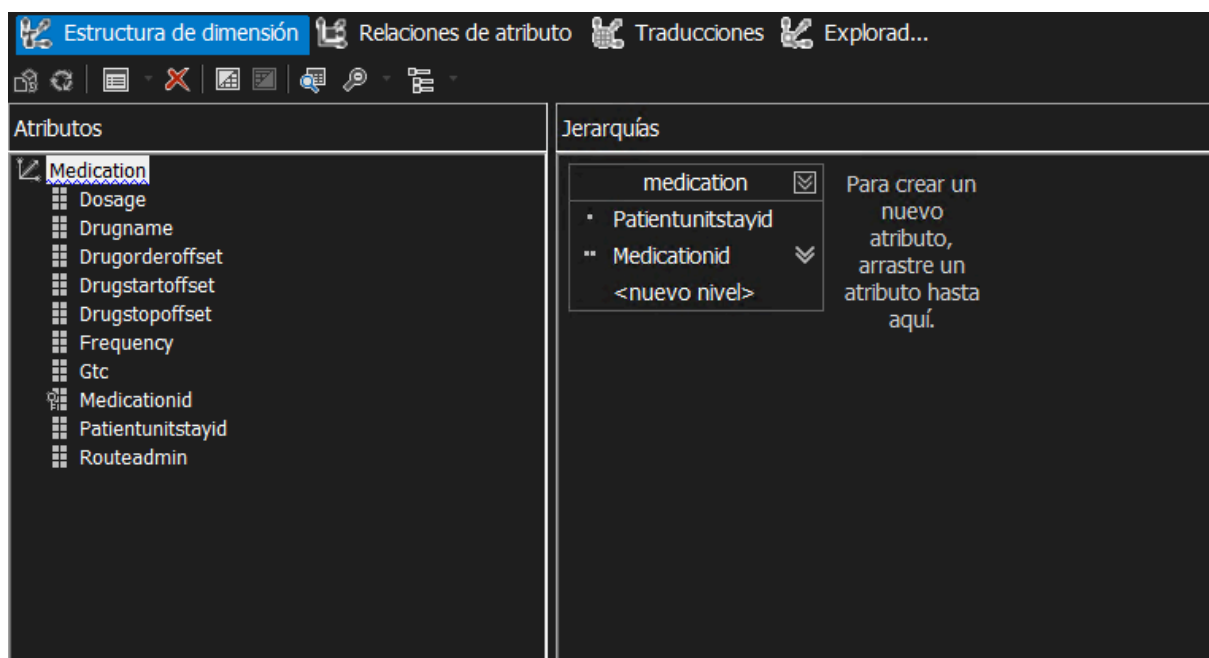
Para la tabla **relation\_13** correspondiente a la relación **M:M**:



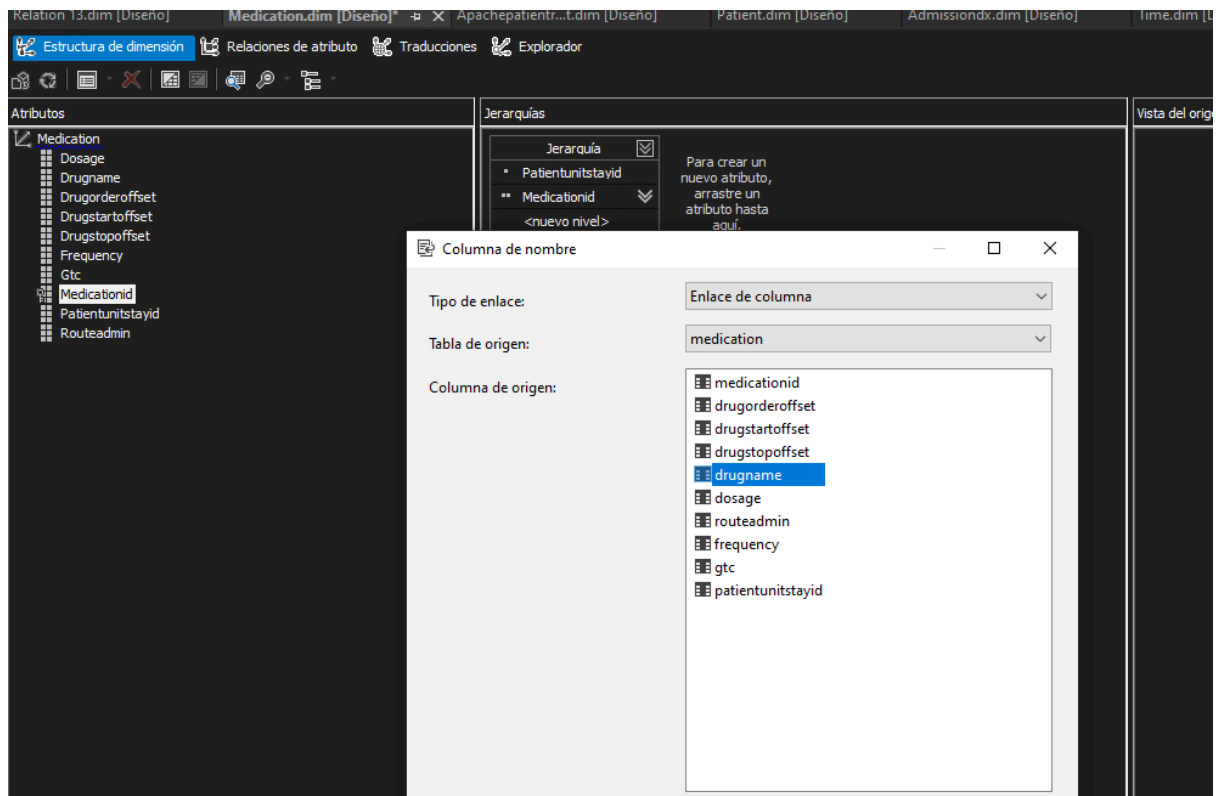
Luciría de la siguiente manera, de forma que cada **medicationid** en la tabla **medicament** está relacionado con ese mismo **id** en el hecho:



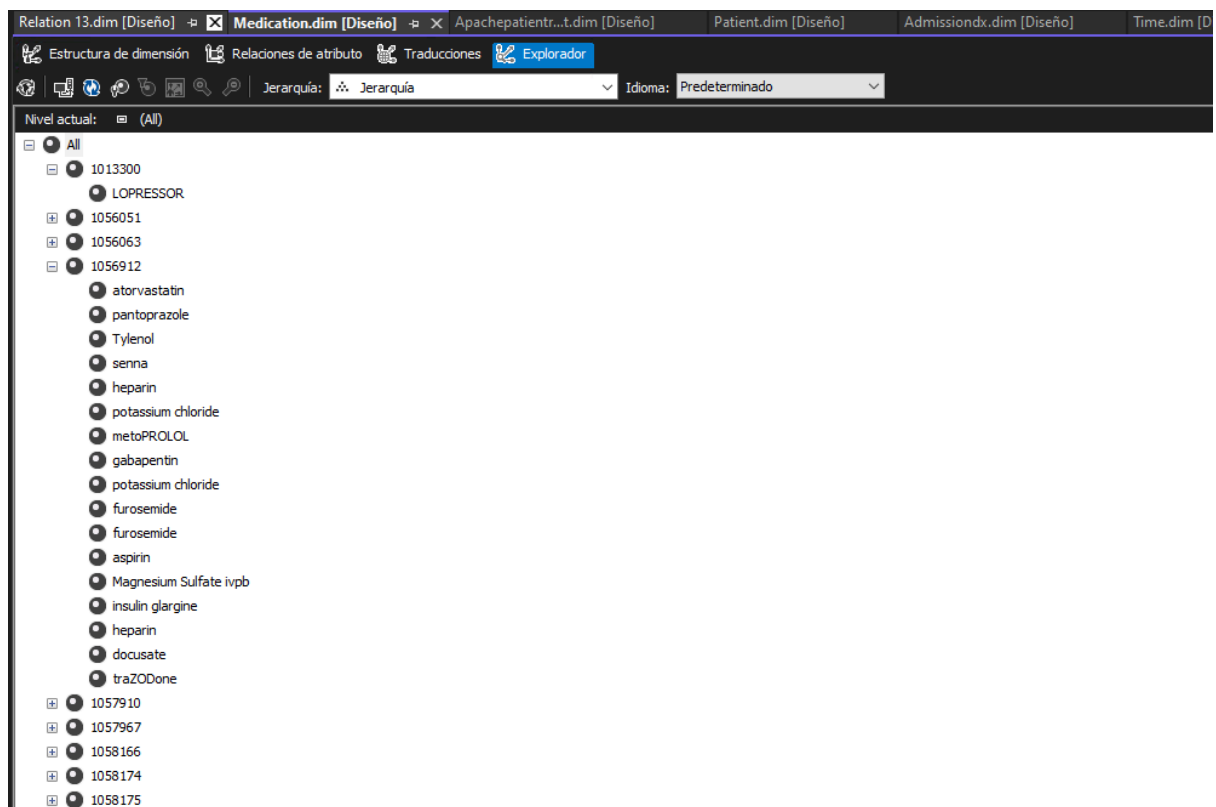
Para la dimensión **medication**, la jerarquía luciría tal y como se puede observar en la imagen con el nombre **medication**:



Sobre **medicationid**, en sus propiedades marcaremos **drugname** como su **namecolumn**, para que de esta manera quede mucho más visible:



De esta manera:

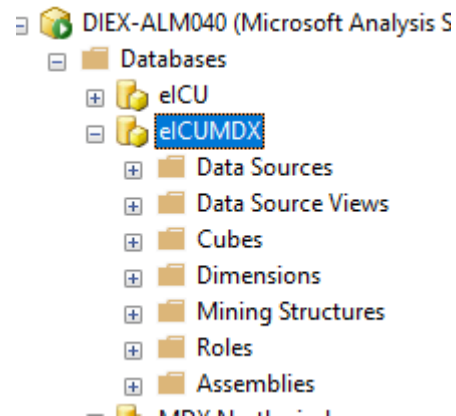




## DESPLIEGUE EN SQL SERVER

Una vez realizado todo lo anterior, procesaremos el resultado y volveremos a implementar la solución tal y como explicamos previamente, de esta manera, nos aseguramos de que **todos los cambios se hayan guardado** y no haya errores en nuestro proyecto.

A continuación, iremos a **sql server** y nos conectaremos mediante **analysis services** para obtener nuestras opciones disponibles:



En este caso, mi cubo es **eICUMDX** por lo que ya está listo para recibir consultas.

# CONSULTAS

## Recuento de Ingresos en UCI por Hospital en Midwest con Manejo de Valores Nulos:

En esta consulta se obtiene el recuento de admisiones en UCI por hospital en la región del Midwest para el año 2015, asegurándonos también de reemplazar los valores nulos con 0, proporcionando un recuento claro y consistente para cada hospital en dicha región.

```
WITH MEMBER [Measures].[RecuentoAdmisionesPorHospitalEnMidwest] AS If( IsEmpty([Measures].[Recuento Icuadmission]), 0,
[Measures].[Recuento Icuadmission] ) SELECT ([Hospital],[Region],[Midwest]) * {[Hospital],[HospitalID].MEMBERS} ON COLUMNS,
[Measures].[RecuentoAdmisionesPorHospitalEnMidwest] ON ROWS FROM [eICU] WHERE ([Time].[Hospitaldischargeyear].&[2015]) --
Filtrado por el año 2015
```

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'eICU' cube is expanded, showing dimensions like Region, Hospital, and Patient. The central pane shows the MDX query:   
`WITH MEMBER [Measures].[RecuentoAdmisionesPorHospitalEnMidwest] AS If( IsEmpty([Measures].[Recuento Icuadmission]), 0, [Measures].[Recuento Icuadmission] ) SELECT ([Hospital],[Region],[Midwest]) * {[Hospital],[HospitalID].MEMBERS} ON COLUMNS, [Measures].[RecuentoAdmisionesPorHospitalEnMidwest] ON ROWS FROM [eICU] WHERE ([Time].[Hospitaldischargeyear].&[2015]) -- Filtrado por el año 2015`   
 The bottom pane shows the query results in a table with 16 columns. The first column is 'RecuentoAdmisionesPorHospitalEnMidwest' with a value of 267. The subsequent 15 columns represent individual hospitals, with values ranging from 0 to 7. The table is filtered for the year 2015.

	Midwest	Midwest	Midwest	Midwest	Midwest	Midwest	Midwest	Midwest	Midwest	Midwest	Midwest	Midwest	Midwest	Midwest	Midwest	Midwest
RecuentoAdmisionesPorHospitalEnMidwest	267	4	5	1	7	3	5	5	3	3	7	3	4	5	5	0

## Promedio de la puntuación Apache (ApacheScore) por diagnóstico cardiovascular

Esta consulta calcula el puntaje APACHE promedio para pacientes con un diagnóstico de ingreso relacionado con enfermedades cardiovasculares.

```
WITH MEMBER [Measures].[Avg_ApacheScore] AS AVG([Patient].[PatientID].MEMBERS, [Measures].[ApacheScore]) SELECT
{[Measures].[Avg_ApacheScore]} ON COLUMNS FROM [eICU] WHERE ([Admissiondx].[Admitdxname].&[Cardiovascular])
```

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The central pane shows the following MDX query:

```
WITH MEMBER [Measures].[Avg_ApacheScore] AS
  AVG([Patient].[PatientID].MEMBERS, [Measures].[ApacheScore])
SELECT
  {[Measures].[Avg_ApacheScore]} ON COLUMNS
FROM
  [eICU]
WHERE
  ([Admissiondx].[Admitdxname].&[Cardiovascular])
```

The left pane shows the 'eICU' cube structure with dimensions like Measures, Admission, and Patient. The bottom pane, titled 'Results', shows a single value for the measure 'Avg\_ApacheScore':

Avg_ApacheScore
55.5

## Cantidad de admisiones por hospital y año

Esta consulta calcula la cantidad de admisiones en UCI, diferenciadas por hospital y año de alta hospitalaria. Utilizamos el recuento de admisiones combinándolo con hospital y año, además de excluir filas vacías para mostrar solo los datos relevantes.

Cantidad de admisiones por hospital y año  
 SELECT {[Measures].[Recuento Icuadmission]} ON COLUMNS, NONEMPTY(  
 {[Hospital].[Hospitalid].MEMBERS} \* {[Time].[Hospitaldischargeyear].MEMBERS} ) ON ROWS FROM [eICU]

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The central pane displays the following MDX query:

```

SELECT
  ([Measures].[Recuento Icuadmission]) ON COLUMNS,
  NONEMPTY(
    {[Hospital].[Hospitalid].MEMBERS} *
    {[Time].[Hospitaldischargeyear].MEMBERS}
  ) ON ROWS
FROM [eICU]
  
```

The left pane shows the 'eICU' cube structure with measures like 'Recuento Icuadmission' and dimensions like 'Hospital' and 'Time'. The bottom pane shows the query results in a table format:

Messages		Results
Recuento Icuadmission		
All	All	1371
All	2014	633
All	2015	738
56	All	6
56	2014	2
56	2015	4
58	All	9
58	2014	4
58	2015	5
59	All	8
59	2014	7
59	2015	1
60	All	9
60	2014	2
60	2015	7
61	All	6
61	2014	3
61	2015	3

**suma total de pacientes admitidos entre 10:00:00 y 22:00:00 por cada hospital**

Calcula la suma total de pacientes admitidos en la UCI entre las 10:00 y las 22:00, desglosada por cada hospital. Utiliza la medida de recuento de admisiones y aplica un filtro en el rango de tiempo especificado para incluir únicamente los datos dentro de ese intervalo horario.

```
SELECT {[Measures].[Recuento Icuadmission]} ON COLUMNS, NONEMPTY ( {[Hospital].[hospitalid].MEMBERS} ) ON ROWS FROM [eICU]
WHERE ( [Time].[Dateadmit24].[Dateadmit24].&[10:00:00]:[Time].[Dateadmit24].[Dateadmit24].&[22:00:00] )
```

MDXQuery11.mdx ~ ALM040.alumnos)\*    MDXQuery8.mdx ~ ALM040.alumnos)\*    SQLQuery11.sql ~ ALM040.alumnos (50)    MDXQuery7.mdx ~ ALM040.alumnos)\*    MDXQuery6.mdx ~ ALM040.alumnos)\*

Cube: eICU

Metadata Functions  
Search Model  
Measure Group:  
<All>

- Patientunitstayd
- Recuento Icuadmission
- Relation 13
- AverageAge
- KPIs
- Admissiondx
- Apachepatientresult
- Hospital
- Hospitalid
- Nregion
- Region
- Medication
- Dosage
- Drugname
- Drugorderoffset
- Drugstartoffset
- Drugstopoffset
- Frequency
- Gtc
- Medicationid
- Patientunitstayd
- Routeadmn
- Jerarquia
- Members
- Patientunitstayd
- Medicationid
- Patient
- Relation 13
- Time
- Date24
- Dateadm24
- Members
- Dateadm24
- Datedischge24
- Hospitaldischgeyear
- Time Key
- Unitadmitoffset
- admt
- Members
- Dateadm24
- Time Key
- discharge

```
SELECT
{([Measures].[Recuento Icuadmission]) ON COLUMNS,
NONEMPTY(
{([Hospital].[hospitalid],MEMBERS)
} ON ROWS
FROM [eICU]
WHERE (
[Time].[Dateadm24].[Dateadm24].[&[10:00:00]:[Time].[Dateadm24].[Dateadm24].[&[22:00:00]]
)
```

90 %

Messages Results

Recuento Icuadmission	
All	656
56	3
58	1
59	5
60	5
61	3
63	5
66	5
67	4
68	4
69	4
71	2
73	4
79	8
92	7
95	4
108	7
110	4

## Promedio de Apachescore por Género y Región

En esta consulta se calcula el promedio del ApacheScore agrupado por género y región hospitalaria. Utiliza un miembro calculado para determinar la media y presenta los resultados combinando género y región, además de ir excluyendo las filas vacías.

```
WITH MEMBER [Measures].[AvgApachescore] AS
    AVG(
        NONEMPTY(
            {[Patient].[Gender].[Gender].MEMBERS} *
            {[Hospital].[Region].[Nregion].MEMBERS}
        ), [Measures].[Apachescore]
    )
SELECT {[Measures].[AvgApachescore]} ON COLUMNS,
    NONEMPTY(
        {[Patient].[Gender].[Gender].MEMBERS} *
        {[Hospital].[Region].[Nregion].MEMBERS}
    ) ON ROWS FROM [eICU]
```

The screenshot shows the SQL Server Enterprise Manager interface. The central pane displays the MDX query for calculating the average ApacheScore by gender and region. The left pane shows the cube structure for 'eICU', including measures like 'Apachescore' and dimensions like 'Patient', 'Hospital', and 'Time'. The bottom pane shows the query results as a table.

		AvgApachescore
	South	8947.125
Female	Midwest	7677.75
Female	Northeast	8542.375
Female	South	7489.125
Female	West	8442.875
Male	Midwest	6837.125
Male	Northeast	8654.875
Male	South	7497.125
Male	West	7487.625

## Recuento de Admisiones en la UCI por Diagnóstico de Admisión para el Año 2015

Esta consulta muestra el recuento de admisiones en la UCI agrupadas por el diagnóstico en el año 2015. Utilizamos `NONEMPTY` para incluir únicamente los diagnósticos con admisiones registradas en ese año.

```
SELECT ([Measures].[Recuento Icuadmission]) ON COLUMNS, NONEMPTY( [Patient].[Apacheadmissiondx].MEMBERS, ([Measures].[Recuento Icuadmission], [Time].[Hospitaldischargeyear].&[2015]) ) ON ROWS FROM [eICU] WHERE ([Time].[Hospitaldischargeyear].&[2015])
```

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The top pane shows the MDX query: `SELECT ([Measures].[Recuento Icuadmission]) ON COLUMNS, NONEMPTY( [Patient].[Apacheadmissiondx].MEMBERS, ([Measures].[Recuento Icuadmission], [Time].[Hospitaldischargeyear].&[2015]) ) ON ROWS FROM [eICU] WHERE ([Time].[Hospitaldischargeyear].&[2015])`. The bottom pane shows the results in a table with two columns: 'Recuento Icuadmission' and 'Patient'. The table lists various medical conditions and their corresponding counts for the year 2015.

Patient	Recuento Icuadmission
All	738
Abdomen only trauma	1
Abdomen/multiple trauma	1
Ablation or mapping of cardiac conduction pathway	2
Acid-base/electrolyte disturbance	10
Adrenalectomy	1
Alcohol withdrawal	3
Anaphylaxis	1
Anemia	1
Aneurysm, abdominal aortic; with rupture	1
Aneurysm, dissecting aortic	1
Aneurysm, thoracic aortic	1
Angina, unstable (angina interferes w/quality of life or meds are tolerated poorly)	9
Aortic and Mitral valve replacement	2
Aortic valve replacement (isolated)	5
Appendectomy	2
ARDS-adult respiratory distress syndrome, non-cardiogenic pulmonary edema	3
Arrest, respiratory (without cardiac arrest)	2
Asthma	2

## Diagnóstico de Admisión Menos Frecuente por Etnia

Esta consulta identifica el diagnóstico de admisión menos frecuente para cada grupo étnico, junto con el recuento de admisiones en la UCI asociadas. Utiliza la función 'BOTTOMCOUNT' para determinar el diagnóstico menos recurrente y aplica 'NONEMPTY' para filtrar únicamente los grupos étnicos que tienen datos.

```
WITH MEMBER [Measures].[LessFrequentAdmissionDx] AS BOTTOMCOUNT( FILTER(
[AdmissionDx].[Admission].[Admitdxname].MEMBERS, NOT ISEMPTY([Measures].[Recuento Icuadmission]) ), 1, [Measures].[Recuento Icuadmission] ).Item(0).Name
SELECT {[Measures].[Recuento Icuadmission], [Measures].[LessFrequentAdmissionDx]} ON COLUMNS, NONEMPTY(
[Patient].[Ethnicity].MEMBERS ) ON ROWS
```

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. On the left, the 'Cube Explorer' pane shows the 'eICU' cube with various dimensions and measures. The 'Functions' pane lists various MDX functions, including 'BOTTOMCOUNT'. The main window shows the following MDX query:

```
WITH MEMBER [Measures].[LessFrequentAdmissionDx] AS
  BOTTOMCOUNT(
    FILTER(
      [AdmissionDx].[Admission].[Admitdxname].MEMBERS,
      NOT ISEMPTY([Measures].[Recuento Icuadmission])
    ),
    1,
    [Measures].[Recuento Icuadmission]
  ).Item(0).Name -- Tomamos el diagnóstico menos frecuente en el contexto de cada etnia
SELECT
  {[Measures].[Recuento Icuadmission], [Measures].[LessFrequentAdmissionDx]} ON COLUMNS,
  NONEMPTY(
    [Patient].[Ethnicity].MEMBERS
  ) ON ROWS
FROM [eICU]
```

Below the query, the 'Results' pane shows a table with two columns: 'Recuento Icuadmission' and 'LessFrequentAdmissionDx'. The table contains data for various ethnic groups:

	Recuento Icuadmission	LessFrequentAdmissionDx
All	1371	Sepsis, unknown
African American	17	Yes
Asian	127	Cardiovascular
Caucasian	18	Yes
Hispanic	1084	Sepsis, unknown
Native American	68	Yes
Other/Unknown	6	No
	51	Yes



## Edad Media y Porcentaje de Pacientes por Género

Calcula la edad media y el porcentaje de pacientes por género dentro de un rango de edades(20 a 50 años):Para ello, filtra las edades válidas dentro del rango y calcula la media de edad para cada género, luego, determina el porcentaje de pacientes de cada género con respecto al total de pacientes.

```
WITH
MEMBER [Measures].[EdadValida] AS
    IIF(
        IsNumeric([Patient].[Age].CURRENTMEMBER.Name)
        AND Val([Patient].[Age].CURRENTMEMBER.Name) >= 20
        AND Val([Patient].[Age].CURRENTMEMBER.Name) <= 50,
        Val([Patient].[Age].CURRENTMEMBER.Name),
        NULL
    )
MEMBER [Measures].[MediaEdadPorGenero] AS
    AVG(
        EXISTING [Patient].[PatientID].MEMBERS,
        [Measures].[EdadValida]
    )
MEMBER [Measures].[RecuentoPacientesPorGenero] AS
    COUNT(EXISTING [Patient].[PatientID].MEMBERS)
MEMBER [Measures].[RecuentoTotalPacientes] AS
    COUNT([Patient].[PatientID].MEMBERS)
MEMBER [Measures].[PorcentajePacientesPorGenero] AS
    IIF(
        [Measures].[RecuentoTotalPacientes] = 0,
        0,
        [Measures].[RecuentoPacientesPorGenero] / [Measures].[RecuentoTotalPacientes] * 100
    )
SELECT
{
    [Measures].[MediaEdadPorGenero],
    [Measures].[PorcentajePacientesPorGenero]
} ON COLUMNS,
{([Patient].[Gender].MEMBERS)} ON ROWS
FROM [eICU]
```

MDXQuery21.mdx...-ALM040\alumnos)\* MDXQuery20.mdx...-ALM040\alumnos)\* SQLQuery21.sql...-M040\alumnos (62)) MDXQuery19.mdx...-ALM040\alumnos)\* SQLQuery20.sql...-M040\alumnos)\*

Cube: eICU

Metadatos Functions

Search Model

Measure Group: <All>

Hospitalid

Region

Region

Members

Region

Hospitalid

Medication

Dosage

Members

Dosage

Drugname

Drugorderoffset

Drugstartoffset

Drugstopoffset

Frequency

Gtc

Medicationid

Patientunitstayid

Routeadmin

Jerarquia

Patient

Age

Apacheadmissiondx

Ethnicity

Gender

Pasthistoryid

Pasthistorynotetype

Pasthistoryvalue

Patientid

Patientid 1

Patient

Members

Pasthistoryid

Patientid

Relation 13

Time

Date24

```
-- Calcula la media de edad para cada género
MEMBER [Measures].[MediaEdadPorGenero] AS
    AVG(
        EXISTING [Patient].[PatientID].MEMBERS, -- Aplica el filtro para las edades válidas en el contexto del género
        [Measures].[EdadValida]
    )

-- Recuento de pacientes por género
MEMBER [Measures].[RecuentoPacientesPorGenero] AS
    COUNT(EXISTING [Patient].[PatientID].MEMBERS)

-- Recuento total de pacientes
MEMBER [Measures].[RecuentoTotalPacientes] AS
    COUNT([Patient].[PatientID].MEMBERS)

-- Porcentaje de pacientes por género respecto al total de pacientes
MEMBER [Measures].[PorcentajePacientesPorGenero] AS
    IIF(
        [Measures].[RecuentoTotalPacientes] = 0,
        0,
        [Measures].[RecuentoPacientesPorGenero] / [Measures].[RecuentoTotalPacientes] * 100
    )

SELECT
{
    [Measures].[MediaEdadPorGenero],
    [Measures].[PorcentajePacientesPorGenero]
} ON COLUMNS,
{([Patient].[Gender].MEMBERS)} ON ROWS
FROM [eICU]
```

68 %

	MediaEdadPorGenero	PorcentajePacientesPorGenero
All	36.5844444444444	100
	40	0.18348623853211
Female	35.4709677419355	39.4954128440367
Male	37.1598639455782	60.4128440366972

# PROBLEMAS ENCONTRADOS

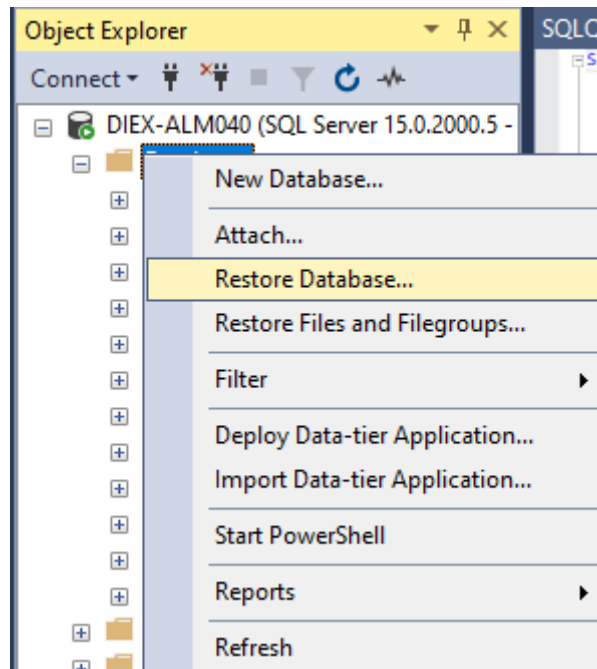
Durante el desarrollo y ejecución del proyecto, se presentaron los siguientes inconvenientes:

**1. ID Autogenerado para la Dimensión de Tiempo:** existía la necesidad de añadir un identificador único autogenerado para la dimensión de tiempo. Por lo que lo implementé para garantizar que cada entrada en la tabla de tiempo fuera única y pudiera ser referenciada correctamente en las consultas. Anteriormente había utilizado una clave inválida por lo que a la hora de querer hacer el cubo era imposible seguir. Finalmente opté por crear ese atributo autogenerado y además crear como clave primaria una combinación de varios atributos de la tabla tiempo ya que al no existir originalmente una relación clave-atributo, no tenía manera de relacionar tablas al momento de realizar el proceso de ETL en mi almacén. De esta manera me aseguraba al 100% de que cada fila fuera única y estuviera referenciada entre sí para hacer posible de nuevo la carga del almacén y más tarde, realizar consultas.

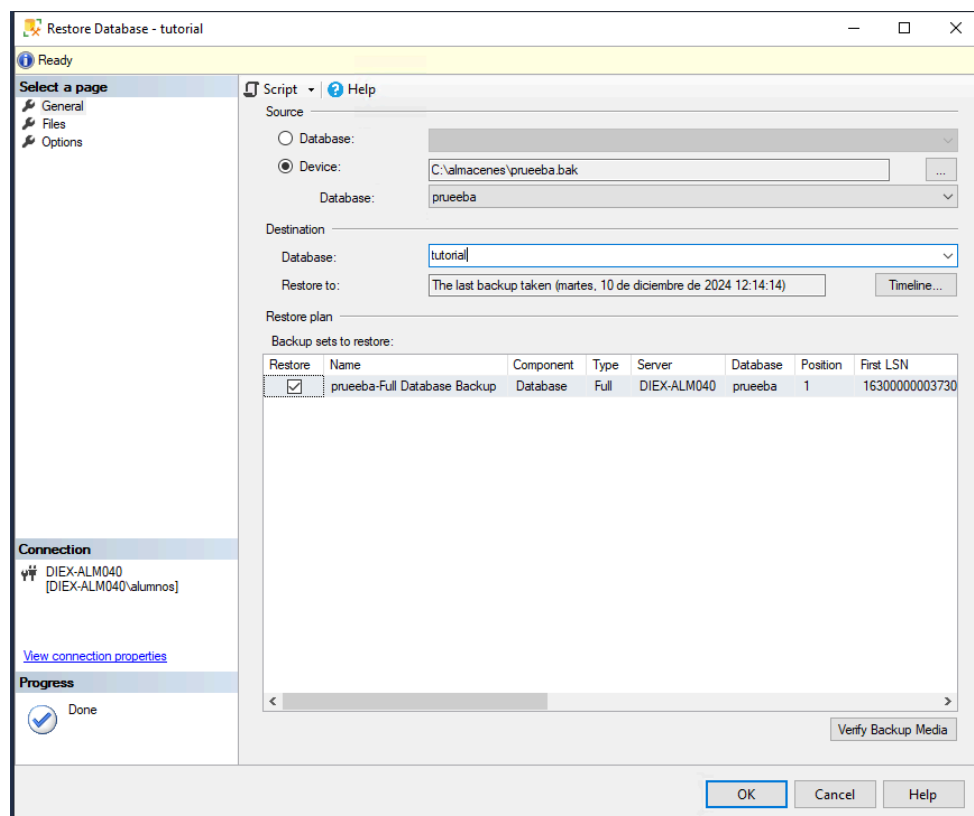
**2. Eliminación de Valores erróneos relacionados con la Dimensión 'Patient':** la dimensión de **patient** contenía valores no presentes en la tabla **icudmission** que afectaban al proceso de creación del cubo en forma de error, además de que posteriormente los resultados de las consultas generarían conteos inexactos. Para resolver este problema, analice el proceso de carga y me di cuenta de que lo estaba realizando de manera incorrecta, ya que al realizar la carga de **icudmission** realizaba varias búsquedas para cargar los atributos relacionados con las dimensiones pero al existir en la **BD original** un atributo **patientid(patientunitstay)**, lo estaba usando directamente desde ahí y así conseguir ahorrarme una búsqueda ya que se trataría del mismo atributo. En realidad este era el problema, ya que los datos que se cargan en patient al realizar la carga con ciertos filtros y los que existen en la base de datos original **son distintos**.

# INSTRUCCIONES PARA EJECUCIÓN

1. Click derecho en database y **restore database**:



2. en device seleccionamos el **.bak** proporcionado en la entrega:



### 3. Ya tendremos cargado el almacén en sql:

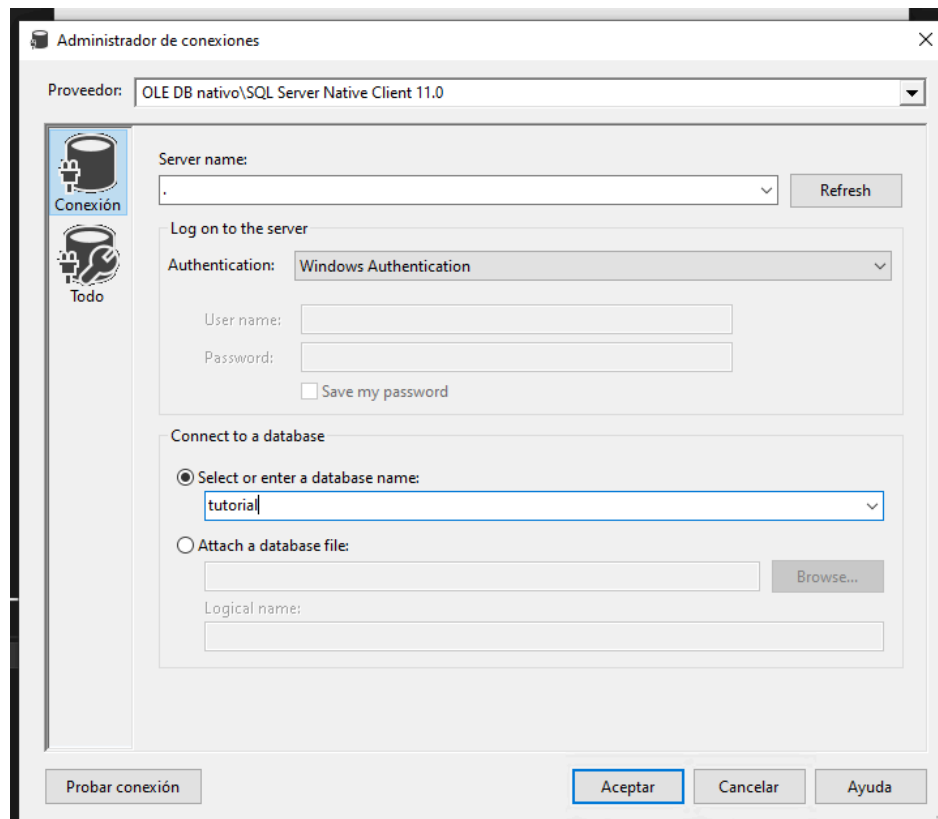
The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure for 'DIEX-ALM040 (SQL Server 15.0.2000)'. The 'tutorial' database is expanded, showing tables like 'dbo.patient' and 'dbo.pasthistory'. The central pane shows a SQL query in 'SQLQuery1.sql':

```
SELECT TOP (1000) [patientid]
, [age]
, [ethnicity]
, [gender]
, [apacheadmissiondx]
, [pasthistoryid]
FROM [tutorial].[dbo].[patient]
```

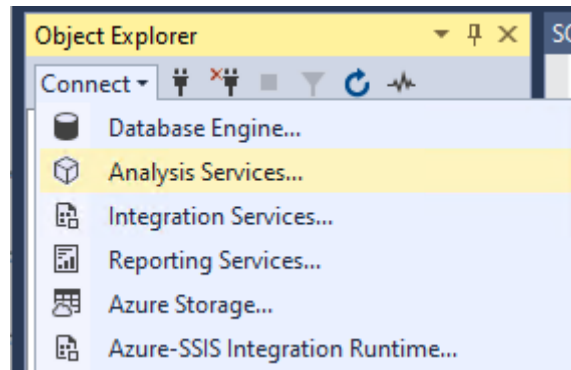
Below the query, the 'Results' tab displays a table with 15 rows of data. The columns are: patientid, age, ethnicity, gender, apacheadmissiondx, and pasthistoryid.

	patientid	age	ethnicity	gender	apacheadmissiondx	pasthistoryid
1	141765	87	Caucasian	Female	Rhythm disturbance (atrial, supraventricular)	990803
2	143870	76	Caucasian	Male	Enderectomy, carotid	970059
3	144815	34	Caucasian	Female	Overdose, other toxin, poison or drug	1180401
4	145427	61	Caucasian	Male	GI perforation/rupture, surgery for	1194998
5	147307	55	Caucasian	Female	Enderectomy, carotid	896652
6	147784	60	Hispanic	Female	Coma/change in level of consciousness (for hepat...	1261966
7	148611	28	Caucasian	Male	Overdose, other toxin, poison or drug	1150985
8	149713	> 89	Caucasian	Female	Infarction, acute myocardial (MI)	1252461
9	151179	59	Caucasian	Female	Sepsis, cutaneous/soft tissue	1137517
10	151867	44	Caucasian	Male	GI perforation/rupture, surgery for	1245288
11	151900	66	Caucasian	Female	Sepsis, pulmonary	915968
12	152954	41	Caucasian	Female	Respiratory - medical, other	1157344
13	153972	63	Caucasian	Male	Bleeding, lower GI	910690
14	155961	57	Caucasian	Female	Knee replacement, total (non-traumatic)	963402
15	156308	87	Caucasian	Male	Sepsis, pulmonary	899913

4. Descomprimos el zip de analysis services, entramos en el .sln y cambiamos la conexión al origen de datos con el nombre que hayamos puesto en sql:



5. Vamos a compilar -> implementar solución y ya aparecerá el cubo en sql server al conectarnos mediante analysis services:



6. Vamos a new query y listo para recibir consultas:

