

EXAMEN PARCIAL DE BASES DE DATOS PARA APOYAR LA TOMA DE DECISIONES

Nombre	Lilia Arceli Lobato Martínez	Expediente	IE706937	Carrera	Ing. Sistemas Computacionales	Fecha	04/10/2021
--------	------------------------------	------------	----------	---------	-------------------------------	-------	------------

Yo Lilia Lobato al participar en este examen me comprometo a que lo entregado sea fruto solamente de mi trabajo, como una oportunidad de demostrar lo que he aprendido hasta el día de hoy. Además, reconozco que la copia es una falta a la honestidad académica y personal. Por lo que me comprometo a no incurrir en acciones que impliquen copiar o dejar que alguien me copie, ni participar en cualquier tipo de intercambio de información por cualquier medio con los estudiantes inscritos a este curso o personas ajenas al mismo.

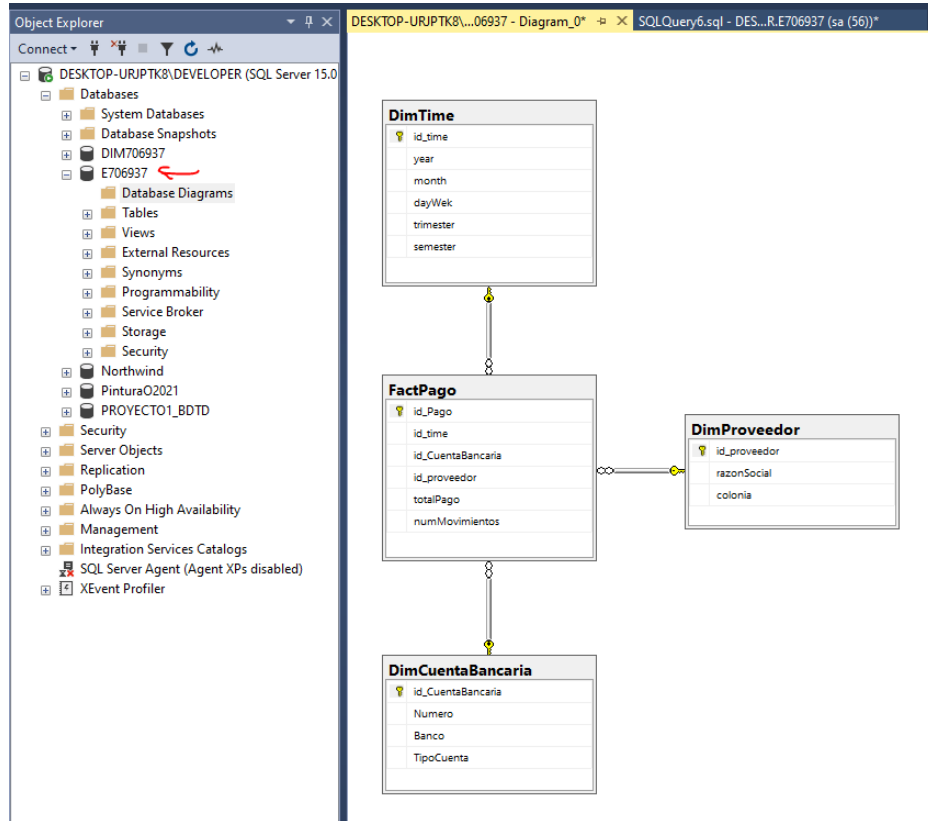
Lee todo el documento, antes de iniciar a responder. A partir del momento en el que inicias el examen tienes 2 horas para completarlo. Al subir a Canvas los entregables se registrará la hora. ¡Mucha suerte!

Responde las preguntas dejando la evidencia solicitada en este documento

1. Crea un modelo dimensional que cumpla al menos todos los requerimientos del anexo 1.
 - 1.1. Evidencia de respuesta captura de pantalla. **[30 puntos]**



2. Implementa tu modelo dimensional en Microsoft SQL.
- 2.1. Evidencia script para crear tu base de datos, si lo hiciste con el asistente al final de la semana 3 de canvas hay una guía para generar el script. Evidencia alterna el archivo .BAK de tu base de datos. **[10 puntos]**



Script: Create_E706937.sql
Archivo: E706937.bak

```
CREATE DATABASE E706937;

-- ***** [DimCuentaBancaria]
CREATE TABLE [DimCuentaBancaria]
(
    [id_CuentaBancaria] int NOT NULL ,
    [Numero]          varchar(50) NOT NULL ,
    [Banco]           varchar(50) NOT NULL ,
    [TipoCuenta]      nvarchar(10) NOT NULL ,
    CONSTRAINT [PK_89] PRIMARY KEY CLUSTERED ([id_CuentaBancaria] ASC)
);

-- ***** [DimProveedor]
CREATE TABLE [DimProveedor]
(
    [id_proveedor] int NOT NULL ,
    [razonSocial]   varchar(250) NOT NULL ,
    [colonia]       varchar(50) NOT NULL ,
    CONSTRAINT [PK_85] PRIMARY KEY CLUSTERED ([id_proveedor] ASC)
);
GO

-- ***** [time]
CREATE TABLE [DimTime]
(
    [id_time]      bigint NOT NULL ,
    [year]         int NOT NULL ,
    [month]        int NOT NULL ,
    [dayWek]       varchar(50) NOT NULL ,
    [trimester]    int NOT NULL ,
    [semester]     int NOT NULL ,
    CONSTRAINT [PK_79] PRIMARY KEY CLUSTERED ([id_time] ASC)
);
GO

-- ***** [FactPago]
CREATE TABLE [FactPago]
(
    [id_Pago]       int NOT NULL ,
    [id_time]       bigint NOT NULL ,
    [id_CuentaBancaria] int NOT NULL ,
    [id_proveedor]  int NOT NULL ,
    [totalPago]     int NOT NULL ,
    [numMovimientos] int NOT NULL ,
    CONSTRAINT [PK_125] PRIMARY KEY CLUSTERED ([id_Pago] ASC),
    CONSTRAINT [FK_113] FOREIGN KEY ([id_time]) REFERENCES [DimTime]([id_time]),
    CONSTRAINT [FK_116] FOREIGN KEY ([id_CuentaBancaria]) REFERENCES [DimCuentaBancaria]([id_CuentaBancaria]),
    CONSTRAINT [FK_119] FOREIGN KEY ([id_proveedor]) REFERENCES [DimProveedor]([id_proveedor])
);
GO

CREATE NONCLUSTERED INDEX [fkidx_115] ON [FactPago]
( [id_time] ASC ) GO

CREATE NONCLUSTERED INDEX [fkidx_118] ON [FactPago]
( [id_CuentaBancaria] ASC ) GO

CREATE NONCLUSTERED INDEX [fkidx_121] ON [FactPago]
( [id_proveedor] ASC ) GO
```

3. Crea un procedimiento almacenado para extraer, transformar y cargar la información de la base de datos relacional descrita en el anexo 3 a la dimensional que creaste en el punto 2.
- 3.1. Evidencia de respuesta copiar y pegar el código SQL en este lugar del documento. **[20 puntos]**

```
(3 rows affected)

(20 rows affected)

(448 rows affected)

(0 rows affected)

(3 rows affected)

(20 rows affected)

(448 rows affected)

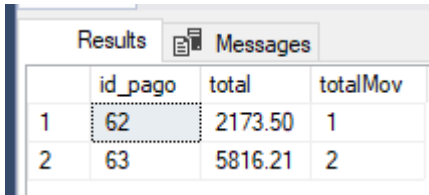
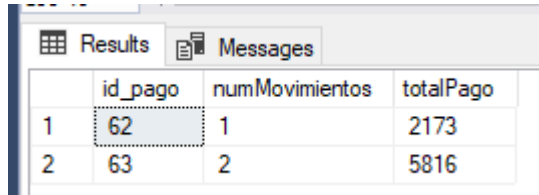
(476 rows affected)

Completion time: 2021-10-04T10:16:26.3080604-05:00
```

Script: Poblar_E706937.sql
Archivo: E706937.bak

Para comprobar que el cubo está poblado correctamente, vamos a comparar resultados tomando como base un ejemplo que se nos dio en el anexo 1.

Id_pago	Total de de movimientos	Total de pago
62	1	2173.50
63	2	5816.21

Pintuta02021	E706937																								
 <table><tr><th></th><th>id_pago</th><th>total</th><th>totalMov</th></tr><tr><td>1</td><td>62</td><td>2173.50</td><td>1</td></tr><tr><td>2</td><td>63</td><td>5816.21</td><td>2</td></tr></table>		id_pago	total	totalMov	1	62	2173.50	1	2	63	5816.21	2	 <table><tr><th></th><th>id_pago</th><th>numMovimientos</th><th>totalPago</th></tr><tr><td>1</td><td>62</td><td>1</td><td>2173</td></tr><tr><td>2</td><td>63</td><td>2</td><td>5816</td></tr></table>		id_pago	numMovimientos	totalPago	1	62	1	2173	2	63	2	5816
	id_pago	total	totalMov																						
1	62	2173.50	1																						
2	63	5816.21	2																						
	id_pago	numMovimientos	totalPago																						
1	62	1	2173																						
2	63	2	5816																						
<pre>select p.id_pago, sum(pd.Total) as total, count(pd.Id_pago_d) as totalMov from [PinturaO2021].[dbo].[Pago] p inner join [PinturaO2021].[dbo].[Pago_d] pd on (pd.Id_pago = p.Id_pago) where p.id_pago = 62 or p.id_pago = 63 group by p.Id_pago</pre>	<pre>select id_pago, numMovimientos, totalPago from [E706937].[dbo].[FactPago] where id_pago = 62 or id_pago = 63</pre>																								

```
USE E706937
GO

CREATE PROCEDURE POBLAR_EX
AS
BEGIN

-- Borro las tablas
DELETE [E706937].[dbo].[DimCuentaBancaria];
DELETE [E706937].[dbo].[DimProveedor];
DELETE [E706937].[dbo].[DimTime];
DELETE [E706937].[dbo].[FactPago];

-- ***** [DimCuentaBancaria] 3
INSERT INTO [E706937].[dbo].[DimCuentaBancaria]
select id_CuentaBancaria, Numero, Banco, tc.Descripcion
from [PinturaO2021].[dbo].[CuentaBancaria] c
inner join [PinturaO2021].[dbo].[TipoDeCuenta] tc on (c.Id_CuentaTipo=tc.Id_CuentaTipo);

-- ***** [DimProveedor] 20
INSERT INTO [E706937].[dbo].[DimProveedor]
select Id_Proveedor, RazonSocial, Colonia
from [PinturaO2021].[dbo].[Proveedor];

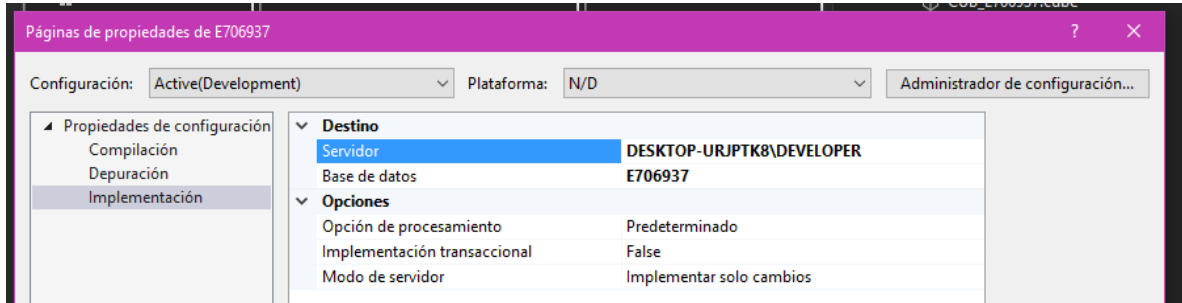
-- ***** [DimTime] 448
INSERT INTO [E706937].[dbo].[DimTime]
SELECT
DISTINCT
CAST (
SUBSTRING( CONVERT ( varchar, Fecha, 126),1,4)+ --año YYYY-MM-DD
SUBSTRING( CONVERT ( varchar, Fecha, 126),6,2)+ --mes
SUBSTRING( CONVERT ( varchar, Fecha, 126),9,2) --día
AS bigint) time_id,
DATEPART(yy, Fecha) as year,
DATEPART(MM, Fecha) as month,
DATENAME(weekday, Fecha) as DayOfWeek,
((DATEPART(quarter, Fecha))) as trimester,
((DATEPART(quarter, Fecha)-1)/2)+1 as semester
FROM [PinturaO2021].[dbo].[Pago];

-- ***** [FactPago] 476
with mov as (
select Id_pago, count(Id_pago_d) as movimientos
from PinturaO2021.dbo.Pago_d
group by Id_pago
)
INSERT INTO [E706937].[dbo].[FactPago]
SELECT DISTINCT
P.Id_pago,
CAST (
SUBSTRING( CONVERT ( varchar, P.Fecha, 126),1,4)+ --año YYYY-MM-DD
SUBSTRING( CONVERT ( varchar, P.Fecha, 126),6,2)+ --mes
SUBSTRING( CONVERT ( varchar, P.Fecha, 126),9,2) --día
AS bigint) time_id,
c.id_CuentaBancaria,
p.Id_Proveedor,
p.Total, --TotalPago
mov.movimientos --cantidadMovimientos
FROM [PinturaO2021].[dbo].[CuentaBancaria] C
INNER JOIN [PinturaO2021].[dbo].[Pago] P ON (P.Id_CuentaBancaria=C.id_CuentaBancaria)
INNER JOIN [PinturaO2021].[dbo].[Pago_d] PD ON (PD.Id_pago=P.Id_pago)
JOIN mov ON (mov.Id_pago = p.Id_pago);

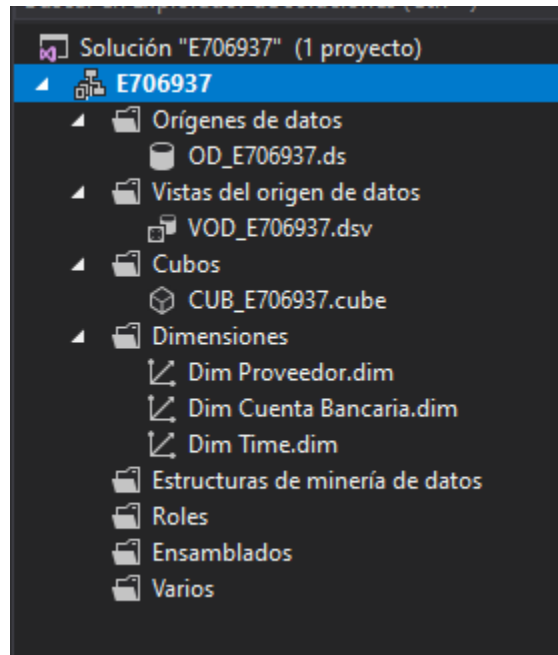
END
```

4. Crea un cubo utilizando Microsoft Visual Studio (SSDT).
4.1. Evidencia de respuesta, 4 capturas de pantalla a tu lección del procedimiento. **[5 puntos]**

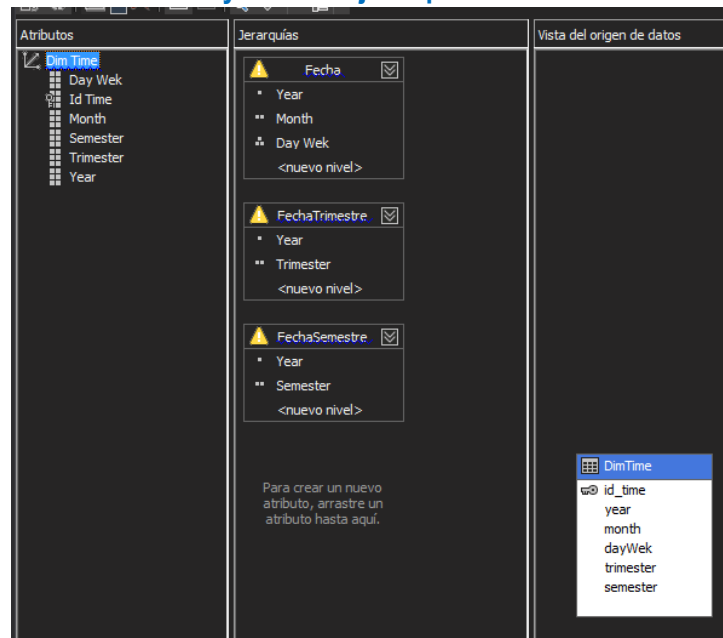
Comenzamos conectándonos a nuestro servidor donde creamos la base



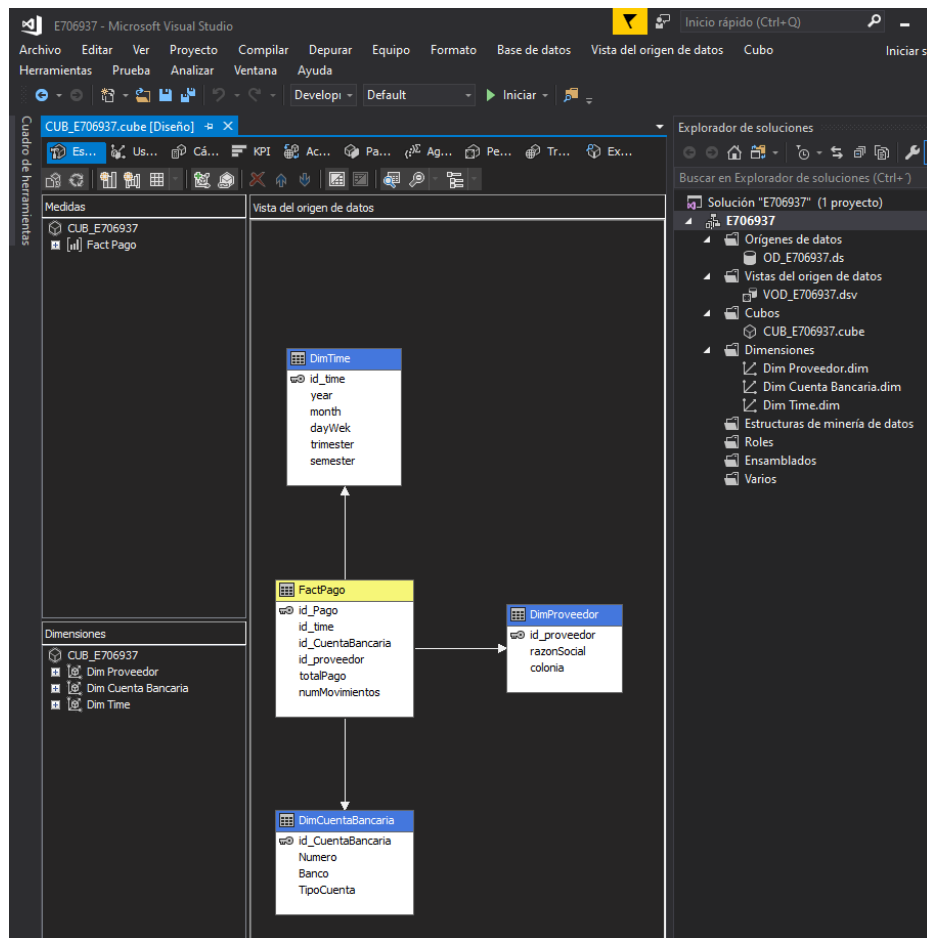
Tenemos que cargar el origen de datos y crear una vista de origen. Tras eso, podemos crear el cubo



Hay que cargarle los valores al cubo y crear las jerarquias



Al final tenemos un cubo con conexión



5. Crea un cubo sin conexión utilizando código MDX.
- 5.1. Evidencia de respuesta copiar y pegar el código MDX en este lugar del documento. [20 puntos]

The screenshot displays the Microsoft SQL Server Enterprise Developer interface. The main window shows the MDX code for creating a cube named [CUBO_NC_E706937]. The code is as follows:

```
CREATE GLOBAL cube [CUBO_NC_E706937]
storage 'C:\Users\1234\source\repos\CUB_NC_E706937'

FROM [CUB_E706937](
    MEASURE [CUB_E706937].[Num Movimientos],
    MEASURE [CUB_E706937].[Total Pago],
    MEASURE [CUB_E706937].[Recuento Fact Pago],

    DIMENSION [CUB_E706937].[Dim Proveedor],
    DIMENSION [CUB_E706937].[Dim Cuenta Bancaria],
    DIMENSION [CUB_E706937].[Dim Time]
)
```

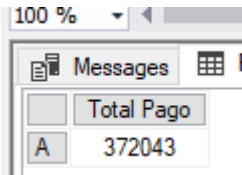
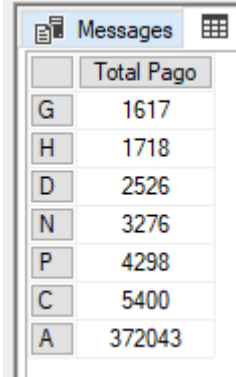
Below the code editor, a file explorer window is open, showing the directory structure of the project. The file explorer shows the following files and folders:

Nombre	Fecha de modificación	Tipo	Tamaño
CUBO706937	19/09/2021 08:12 p. m.	Carpeta de archivos	
E706937	04/10/2021 10:21 a. m.	Carpeta de archivos	
MD706937	19/09/2021 07:51 p. m.	Carpeta de archivos	
CUB_NC_706937.cub	03/10/2021 11:38 p. m.	Archivo CUB	3,644 KB
CUB_NC_E706937.cub	04/10/2021 10:42 a. m.	Archivo CUB	760 KB

Script: MDX_Cube.mdx
Archivo: CUB_NC_E706937.cub

6. Responde tres requerimientos de tu elección del anexo 2 con código MDX.
6.1. Evidencia de respuesta copiar y pegar el código MDX en este lugar del documento. [15 puntos]

Nombre del banco en columnas, año en filas y el cruce el total pagado.																																																																
<div> <div>Messages</div> <div>Results</div> <table> <tr> <th></th><th></th><th>Banamex</th><th>BBVA</th><th>Santander</th></tr> <tr><td>Total Pago</td><td>2009</td><td>(null)</td><td>188220</td><td>202658</td></tr> <tr><td>Total Pago</td><td>2010</td><td>(null)</td><td>250599</td><td>215203</td></tr> <tr><td>Total Pago</td><td>2011</td><td>(null)</td><td>192814</td><td>328222</td></tr> <tr><td>Total Pago</td><td>2012</td><td>(null)</td><td>211414</td><td>292320</td></tr> <tr><td>Total Pago</td><td>2013</td><td>(null)</td><td>145896</td><td>170951</td></tr> <tr><td>Total Pago</td><td>2014</td><td>(null)</td><td>101904</td><td>171659</td></tr> <tr><td>Total Pago</td><td>2015</td><td>(null)</td><td>272562</td><td>160232</td></tr> <tr><td>Total Pago</td><td>2016</td><td>(null)</td><td>334285</td><td>240565</td></tr> <tr><td>Total Pago</td><td>2017</td><td>(null)</td><td>162625</td><td>199061</td></tr> <tr><td>Total Pago</td><td>2018</td><td>(null)</td><td>127309</td><td>356096</td></tr> <tr><td>Total Pago</td><td>2019</td><td>(null)</td><td>141328</td><td>151927</td></tr> </table> </div>							Banamex	BBVA	Santander	Total Pago	2009	(null)	188220	202658	Total Pago	2010	(null)	250599	215203	Total Pago	2011	(null)	192814	328222	Total Pago	2012	(null)	211414	292320	Total Pago	2013	(null)	145896	170951	Total Pago	2014	(null)	101904	171659	Total Pago	2015	(null)	272562	160232	Total Pago	2016	(null)	334285	240565	Total Pago	2017	(null)	162625	199061	Total Pago	2018	(null)	127309	356096	Total Pago	2019	(null)	141328	151927
		Banamex	BBVA	Santander																																																												
Total Pago	2009	(null)	188220	202658																																																												
Total Pago	2010	(null)	250599	215203																																																												
Total Pago	2011	(null)	192814	328222																																																												
Total Pago	2012	(null)	211414	292320																																																												
Total Pago	2013	(null)	145896	170951																																																												
Total Pago	2014	(null)	101904	171659																																																												
Total Pago	2015	(null)	272562	160232																																																												
Total Pago	2016	(null)	334285	240565																																																												
Total Pago	2017	(null)	162625	199061																																																												
Total Pago	2018	(null)	127309	356096																																																												
Total Pago	2019	(null)	141328	151927																																																												
<pre>select [Dim Cuenta Bancaria].[Banco].[Banco] on columns, non empty ([Measures].[Total Pago], [Dim Time].[Year].[Year]) on rows from [CUB_E706937]</pre>																																																																

Razón social de los proveedores que representan el 20% del total de pagos en el año 2019. Razón social en filas y total pagado en columnas.	
	<pre>select [Measures].[Total Pago] on columns, non empty toppercnt([Dim Proveedor].[Razon Social].[Razon Social], 20, [Measures].[Total Pago]) on rows from [CUB_E706937] where [Dim Time].[Year].[2009]</pre>
	<pre>select [Measures].[Total Pago] on columns, non empty bottompercent([Dim Proveedor].[Razon Social].[Razon Social], 20, [Measures].[Total Pago]) on rows from [CUB_E706937] where [Dim Time].[Year].[2009]</pre>

Porcentaje del total que representa el pago a cada proveedor. Razón social en filas. Total pagado y porcentaje que representa en columnas.

	Total Pago	% del Total
All	4617850	100.00 %
A	4373573	94.71 %
P	47152	1.02 %
D	45831	.99 %
J	24733	.54 %
C	22301	.48 %
N	20495	.44 %
F	17441	.38 %
E	13911	.30 %
H	11171	.24 %
G	7991	.17 %
Q	7624	.17 %
Y	6169	.13 %
I	5957	.13 %
X	5470	.12 %
O	4076	.09 %
Z	1925	.04 %
K	1455	.03 %
M	575	.01 %

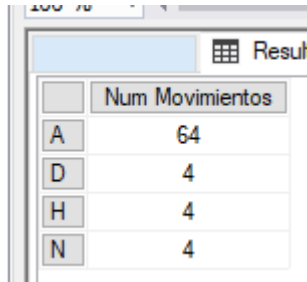
```
WITH
    MEMBER [Measures].[% del Total] as
        [Measures].[Total
Pago]/
        ([Dim
Proveedor].[Razon Social].[All],
        [Measures].[Total
Pago]),
        format_string=" #.00
%"
SELECT
    {[Measures].[Total Pago],
    [Measures].[% del Total]}
ON COLUMNS,
NON EMPTY ORDER(
    [Dim Proveedor].[Razon
Social].MEMBERS,
    [Measures].[% del Total],
    BDESC)
ON ROWS
FROM [CUB_E706937];
```

Top 3 de movimientos en el pago a proveedores. En filas la razón social de los proveedores y en columnas el total de movimientos.

	Num Movimientos
A	666
P	28
D	27

```
SELECT
non empty
    [Measures].[Num Movimientos]
ON COLUMNS,
NON EMPTY topcount(
    [Dim Proveedor].[Razon
Social].[Razon Social]
    ,3, [Measures].[Num Movimientos]
)ON ROWS
FROM [CUB_E706937]
```

Únicamente los lunes. Mostrar la razón social de los proveedores en filas y el total de movimientos en columnas, solo de aquellos que tengas más de tres movimientos.



The screenshot shows a PivotTable with the following data:

	Num Movimientos
A	64
D	4
H	4
N	4

```
SELECT  
non empty  
    [Measures].[Num Movimientos]  
ON COLUMNS,  
NON EMPTY  
    FILTER(  
        [Dim Proveedor].[Razon  
Social].[Razon Social],  
        [Measures].[Num  
Movimientos] > 3)  
ON ROWS  
FROM [CUB_E706937]  
WHERE [Dim Time].[Day Wek].[Monday]
```

Sube a Canvas los siguientes 4 entregables por separado:

- 1) Este documento, en formato de Ms Word o PDF, con las respuestas a cada una de las preguntas **[100 puntos menos si no se sube a Canvas o se sube y no puede abrirse el archivo]**
- 2) El respaldo de tu base de datos dimensional poblada. Archivo .BAK o script para crear la base de datos. **[20 puntos menos si no se sube a Canvas o se sube y no puede abrirse el archivo]**
- 3) El archivo que contiene tu procedimiento almacenado (SP). Archivo .sql o .txt **[20 puntos menos si no se sube a Canvas o se sube y no puede abrirse el archivo]**
- 4) Tu cubo sin conexión. Archivo .CUB **[20 puntos menos si no se sube a Canvas o se sube y no puede abrirse el archivo]**

Anexo 1. Requerimientos mínimos para el cubo

1. El **total de pago** por cuenta bancaria. Se requiere el nombre del banco y el total pagado.
2. El **total de pago** por año, mes y día de la semana. Se requiere mostrar la **razón social del proveedor** y el **total de pago** por día de la semana.
3. El **total de pago** por **tipo de cuenta**. Los datos a mostrar son el **tipo de cuenta** y el **total de pago**.
4. Se requiere saber el **semestre y trimestre** en el que se ha hecho el pago a cada proveedor. Los datos a mostrar son el **año, semestre, el total de pago y la colonia del proveedor**.
5. Se requiere saber el número de movimientos por cada pago. Los datos a mostrar son el **Id_pago**, el total de movimientos y el total de pago. Si tomamos como ejemplo el pago con Id_pago = 62 y Id_pago = 63, la salida mostrada sería

Id_pago	Total de de movimientos	Total de pago
62	1	2173.50
63	2	5816.21

6. Dado un identificador de pago, mostrar la razón social del proveedor, el nombre del banco con el que se le hizo el pago, el tipo de cuenta, el total de movimientos y el total de pago.

Anexo 2. Preguntas de análisis con MDX

- a) Nombre del banco en columnas, año en filas y el cruce el total pagado.
- b) Razón social de los proveedores que representan el 20% del total de pagos en el año 2019. Razón social en filas y total pagado en columnas.
- c) Porcentaje del total que representa el pago a cada proveedor. Razón social en filas. Total pagado y porcentaje que representa en columnas.
- d) Top 3 de movimientos en el pago a proveedores. En filas la razón social de los proveedores y en columnas el total de movimientos.
- e) Únicamente los lunes. Mostrar la razón social de los proveedores en filas y el total de movimientos en columnas, solo de aquellos que tengas más de tres movimientos.

Anexo 3. Diagrama relacional de PinturaO2021 a utilizar para crear el modelo dimensional

