

Caso práctico:

Almacén de datos para el análisis de estadísticas deportivas de las ligas de baloncesto WNBA y NBA

Contenido

INTRODUCCIÓN	2
I. IDENTIFICACIÓN DE LOS PROCESOS ETL	2
II. DISEÑO Y DESARROLLO DE LOS PROCESOS ETL	5
a) Creación de tablas Intermedias (<i>STAGING AREA</i>)	5
b) Creación de tablas del modelo multidimensional (<i>DIMENSIONES Y HECHOS</i>)	5
c) Creación del proceso de extracción, transformación y carga (ETL)	6
Carga y Transformaciones - Bloque IN	7
Carga y Transformaciones - Bloque TR_DIM	34
Carga y Transformaciones - Bloque TR_FACT	58
III. IMPLEMENTACIÓN DE LOS PROCESOS ETL.....	64
a) Trabajo JOB_IN	64
b) Trabajo JOB_TR_DIM	66
c) Trabajo JOB_TR_FACT.....	67
d) Trabajo JOB_ETL.....	68
e) Vistas de las tablas pobladas con los datos en la BD de SQL SERVER	69
IV. DISEÑO Y CREACIÓN DEL MODELO OLAP.....	70
a) Creación del proyecto	70
b) Origen de datos	72
c) Vista del Origen de datos.....	73
d) Creación de Cubos	78
e) Dimensiones, atributos y relaciones en el Cubo	83
f) Implementación de la solución.....	89
g) Explotación de la solución	91
CONCLUSIONES	97
REFERENCIAS BIBLIOGRÁFICAS.....	98
ANEXOS	99
Script creación tablas intermedias (STG)	100
Script creación tablas modelo (DIMENSIONES)	102
Script creación tablas modelo (HECHOS)	104
Vistas de las tablas pobladas con los datos en la BD de SQL SERVER.....	105

Caso práctico:

Almacén de datos para el análisis de estadísticas deportivas de las ligas de baloncesto WNBA y NBA

INTRODUCCIÓN

Una vez tenemos la propuesta del diseño del almacén de datos, y ésta ha sido implementada en la base de datos con la que trabajamos, es el momento de centrarnos en los **procesos ETL**. Como ya sabemos, estos procesos consisten en la **extracción, transformación y carga de los datos**. En definitiva, lo que se persigue es estructurar y acomodar los datos de las fuentes de origen en el almacén de datos. En nuestro caso particular, tenemos diversas fuentes de origen de datos, con diferentes formatos (como ya se ha visto en la práctica 1), y una fuente de destino: la base de datos en SQL Server donde se aloja nuestro almacén de datos.

En esta parte del caso práctico, se debe diseñar, implementar y ejecutar los procesos de extracción, transformación y carga de los datos de las fuentes de datos proporcionadas. Tras la carga efectiva de los datos en el almacén de datos mediante los procesos ETL desarrollados, se debe implementar un cubo multidimensional para la explotación de la información como apoyo a la toma de decisiones de los usuarios potenciales que fueron presentados en el inicio de la práctica.

Los objetivos de esta parte del caso práctico son:

- I. Identificación de los procesos de ETL (extracción, transformación y carga de datos) hacia el almacén de datos.
- II. Diseño y desarrollo de los procesos ETL mediante las herramientas de diseño proporcionadas.
- III. Implementación con trabajos de los procesos ETL para su carga efectiva planificada.
- IV. Diseñar un modelo OLAP (*multidimensional on line analytical processing*) para el análisis multidimensional de la información disponible en el almacén de datos que permita dar respuesta a las siguientes cuestiones:
 - Listado de puntos anotados por conferencia, división y posición de juego durante la temporada 2015-2016 en la NBA.
 - Ranking por eficiencia (TOP 10) de las jugadoras de la WNBA durante la temporada 2012-2013.
 - Ranking (TOP 10) de jugadores de la NBA con mejor porcentaje de aciertos en tiros de 3 puntos, durante la temporada 2011-2012.
 - Listado de equipos de la NBA, ordenado por el número rebotes defensivos realizados en la temporada 2014-2015 (ordenación descendente por DREB).
 - Partido de los Boston Celtics jugado como visitante en la temporada 2013-2014 donde se fallaron más tiros libres en el segundo intento.
 - Evolución temporal desde la temporada 2006-2007 hasta la 2015-2016 del total de tiros libres realizados, acertados, fallados y tanto por ciento de acierto por brazo dominante del jugador.

I. IDENTIFICACIÓN DE LOS PROCESOS ETL

Los procesos ETL deben conceptualizarse como manipulaciones de flujos dedatos. Estos procesos deben diseñarse teniendo en cuenta distintos factores como los siguientes.

- Cómo debe cargarse de manera lógica la información, es decir, qué debecargarse primero y qué después.
- La ventana de tiempo disponible, hecho que puede condicionar lo quedebe cargar.
- Tipo de carga: inicial o incremental.

A la hora de diseñar los procesos de carga de una base de datos analítica no hay una única estrategia que sirva para todos los casos. Es habitual estructurar los procesos ETL sobre la base de las entidades de datos que se deben actualizar, ya que existen diferencias conceptuales en la actualización de una dimensión respecto a la de una tabla de hechos. La división del proceso de carga inicial en diferentes bloques de actualización facilitará el diseño de un orden de ejecución y la gestión de las dependencias. Cada uno de estos bloques de actualización se dividirá en las correspondientes etapas de extracción, transformación y carga.

Como ya se definió en el diseño de la arquitectura en la PARA 1, consideraremos el uso de una Staging Area para lo siguiente:

- Facilitar la extracción de datos (con procesos ETL) desde las fuentes de origen realizando un pretratamiento de éstas, si es necesario.
- Realizar lo que se conoce como Data Cleansing (detección, corrección y eliminación de datos erróneos).
- Mejorar la calidad de los datos.

En nuestro caso identificamos dos bloques y utilizaremos un prefijo en el nombre para identificarlos:

- **Bloque IN:** procesos de carga de los datos desde las fuentes a las tablas intermedias en el área de maniobras (staging area). Estos procesos se distinguen por el prefijo: IN_ en el nombre.
- **Bloque TR:** procesos de transformación para la carga de datos desde tablas intermedias a nuestro almacén según el modelo multidimensional diseñado. Se diferencian los procesos ETL de transformación para la carga de dimensiones de los procesos de transformación para la carga de las tablas de hecho. Estos procesos se distinguen con el prefijo TR_ en el nombre.

A continuación, se identifican los procesos que forman parte de cada uno de los bloques identificados:

Bloque IN (de las fuentes a tablas intermedias)

Los procesos del bloque IN de carga de datos al área intermedia son:

Nombre ETL	Descripción	Orígenes de datos	Tabla destino (STAGE)
IN_STATS	Carga de los datos correspondientes a las estadísticas de juego por temporadas de los jugadores de la NBA y jugadoras de la WNBA	WNBA_Seasons_Stats_2005_2017.xls NBA_Seasons_Stats_1950_2017.csv	STG_WNBA_Seasons_Stats_2005_2017 STG_NBA_Seasons_Stats_1950_2017
IN_TEAMS	Carga de los datos correspondientes a los equipos de la NBA y de la WNBA.	NBA_Teams.xml WNBA_Teams.xls TeamCodes.txt	STG_NBA_Teams STG_WNBA_Teams STG_Team_Codes
IN_FREE_THROWS	Carga de los datos correspondientes a las estadísticas de tiros libres en la NBA.	nba_free_throws.csv	STG_NBA_free_throws
IN_EEUU	Carga de los datos correspondientes a los estados de EEUU.	Estados_Unidos.json	STG_EEUU
IN_PLAYERS_NBA	Carga de los datos correspondientes de los jugadores de la NBA.	NBA_players_data.json NBA_Players_List.xls	STG_NBA_Players_data STG_NBA_Players_list
IN_PLAYERS_WNBA	Carga de los datos correspondientes de las jugadoras de la WNBA.	WNBA_Players_List.txt	STG_WNBA_Players_list

Bloque TR (de las tablas intermedias al almacén)

El bloque TR_ de procesos ETL para poblar el modelo multidimensional del almacén tiene dos partes diferenciadas: los procesos de carga y transformación de las dimensiones y los de tablas de hechos. El orden de ejecución es importante para que la carga de datos sea correcta. Las dimensiones se cargarán primero y después las tablas de hechos para no tener errores en la carga.

Los procesos del bloque TR de carga y transformación de las dimensiones son:

Nombre ETL	Descripción	Tablas Origen	Tabla destino (DIMENSION)
TR_DIM_PLAYS	Carga y transformación de la dimensión con datos de tipos de jugadas.	STG_NBA_free_throws	DIM_Jugadas
TR_DIM_CONF	Carga y transformación de la dimensión con información de las conferencias de la NBA/WNBA.	STG_NBA_Teams	DIM_Conferencias
TR_DIM_TIME	Carga y transformación de la dimensión con datos de las temporadas.	STG_NBA_Seasons_Stats_1950_2017	DIM_Tiempo
TR_DIM_MIN	Carga y transformación de la dimensión con datos de los minutos y segundos de un partido.	STG_NBA_free_throws	DIM_Minutos
TR_DIM_POS	Carga y transformación de la dimensión con información sobre la posición del juego de los jugadores.	Tabla de referencia Enunciado PRAC1	DIM_PosicionesJuego
TR_DIM_GAMES	Carga y transformación de la dimensión con información sobre los partidos.	STG_NBA_free_throws	DIM_Partidos
TR_DIM_PLAYERS	Carga y transformación de la dimensión con información personal sobre los jugadores y jugadoras.	STG_WNBA_Seasons_Stats_2005_2017 STG_NBA_Seasons_Stats_1950_2017	DIM_Jugadores
TR_DIM_EEUU	Carga y transformación de la dimensión con datos sobre los estados de EEUU.	STG_EEUU	DIM_Estados_EEUU
TR_DIM_TEAMS	Carga y transformación de la dimensión con datos sobre los equipos de la NBA y WNBA.	STG_Team_Codes	DIM_Equipos
TR_DIM_DIV	Carga y transformación de la dimensión con datos sobre las divisiones de la NBA y WNBA.	STG_NBA_Teams	DIM_Divisiones

Y el proceso del bloque de carga y transformación de la tabla de hechos:

Nombre ETL	Descripción	Tablas Origen
TR_FACT_FREE_THROWS	Carga y transformación de la tabla de hechos FACT_FREE_THROWS.	STG_NBA_free_throws
TR_FACT_SEASON_STATS	Carga y transformación de la tabla de hechos FACT_SEASON_STATS.	STG_WNBA_Seasons_Stats_2005_2017 STG_NBA_Seasons_Stats_1950_2017

Existen otras estrategias válidas que nos permitirán cargar los datos, ya sea organizando los procesos de otra forma o fusionándose en un único proceso que lleve a cabo todas las tareas. La opción de una única ETL, podría ser de aplicación en nuestro caso, aunque no es recomendable en cargas más complejas y cambiantes.

En el diseño de los procesos ETL se han tenido en cuenta las siguientes consideraciones:

- Desconocemos la ventana de tiempo disponible (y no es una condición para esta actividad), pero en el contexto de producción, es un factor muy relevante a tener en cuenta en el diseño y ejecución de los procesos para que las cargas no impacten en ningún sistema productivo.
- Son procesos ETL de cargas iniciales con el objetivo de ejecutarse para la carga inicial de datos al almacén. No son procesos ETL para cargas incrementales de datos dado que éstas tienen como el objetivo repetirse de manera periódica.
- No se han incluido los procesos específicos de control de errores, de generación de logs, de metadatos...
- Tal y como se definió en la fase de diseño, usaremos un área intermedia (staging area) para cargar los datos desde las fuentes origen antes de poblar el almacén.

- Los procesos podrán ejecutarse tantas veces como sea necesario garantizando la calidad y la integridad de los datos del almacén.

II. DISEÑO Y DESARROLLO DE LOS PROCESOS ETL

En este apartado, y teniendo en cuenta las consideraciones anteriores, vamos a diseñar e implementar los procesos de carga mediante la herramienta de diseño proporcionada: Pentaho Data Integration (PDI). Y en particular, el programa de escritorio llamado Spoon, que corresponde al entorno gráfico (IDE) de desarrollo de ETL's.

Los procesos ETL que diseñaremos en PDI consistirán la definición de trabajos y transformaciones.

a) Creación de tablas Intermedias (STAGING AREA)

El primer paso para la implementación del proceso de ETL, consiste en la creación de las tablas intermedias en la staging área. Ésta se llevará a cabo una única vez, mediante scripts sobre la base de datos proporcionada, en nuestro caso, SQL Server. Las tablas intermedias se utilizarán en los procesos IN que permitirán cargar los datos desde las fuentes de datos.

En nuestro caso, las tablas intermedias serán iguales que las de origen, y después (entre las tablas intermedias y las del modelo) se harán las transformaciones y modificaciones necesarias. Listado de tablas intermedias:

- ❖ Tabla Intermedia: STG_EEUU
- ❖ Tabla Intermedia: STG_NBA_free_throws
- ❖ Tabla Intermedia: STG_NBA_Players_data
- ❖ Tabla Intermedia: STG_NBA_Players_list
- ❖ Tabla Intermedia: STG_NBA_Seasons_Stats_1950_2017
- ❖ Tabla Intermedia: STG_NBA_Teams
- ❖ Tabla Intermedia: STG_Team_Codes
- ❖ Tabla Intermedia: STG_WNBA_Players_list
- ❖ Tabla Intermedia: STG_WNBA_Seasons_Stats_2005_2017
- ❖ Tabla Intermedia: STG_WNBA_Teams

Se adjunta el script de creación de las tablas intermedias en la sección de Anexos del presente documento.

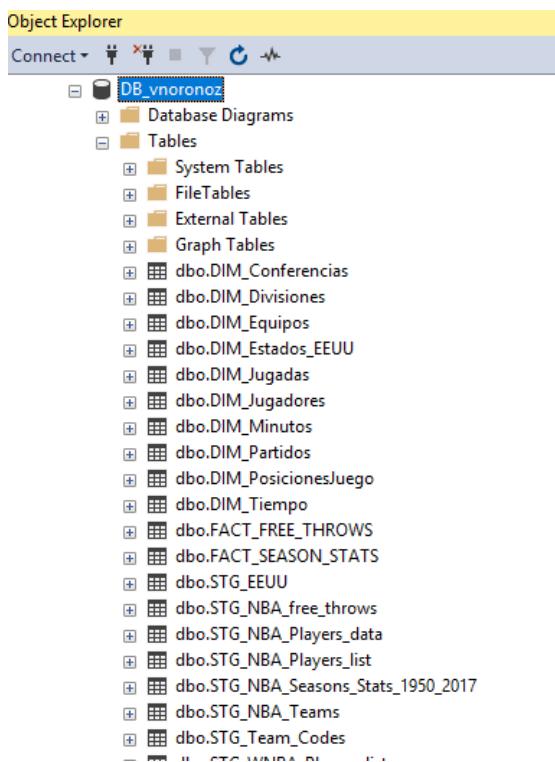
b) Creación de tablas del modelo multidimensional (DIMENSIONES Y HECHOS)

En este punto veremos la creación del modelo físico multidimensional que hemos diseñado para el almacén de egresados universitarios, compuesto por las dimensiones y las tablas de hechos. En la creación, además de atributos y métricas, se crearán también las restricciones definidas y que son propias del modelo multidimensional, las claves primarias de las dimensiones y las foráneas de las tablas de hechos. Listado de tablas de dimensiones y de hechos:

- | | |
|--|---|
| <ul style="list-style-type: none"> ○ DIMENSIONES <ul style="list-style-type: none"> ❖ DIM_Conferencias ❖ DIM_Divisiones ❖ DIM_Equipos ❖ DIM_Estados_EEUU ❖ DIM_Jugadas ❖ DIM_Jugadores ❖ DIM_Minutos ❖ DIM_Partidos ❖ DIM_PosicionesJuego ❖ DIM_Tiempo | <ul style="list-style-type: none"> ○ HECHOS <ul style="list-style-type: none"> ❖ FACT_FREE_THROWS ❖ FACT_SEASON_STATS |
|--|---|

Se adjunta el script de creación de las tablas de dimensiones y de las tablas de hechos en la sección de [Anexos](#) del presente documento.

Vista de las tablas generadas en la DB:



c) Creación del proceso de extracción, transformación y carga (ETL)

Una vez que tenemos implementado el modelo físico del almacén, pasaremos a diseñar los procesos ETL que permitirán poblar las tablas intermedias y las tablas de dimensiones y de hechos del DW que hemos diseñado.

Antes del diseño de las transformaciones definiremos en PDI las variables de entorno que usaremos en la implementación de los procesos ETL, así como la conexión a la base de datos que utilizaremos en todos ellos.

Variables de entorno

Es una buena práctica utilizar variables de entorno, para evitar introducir errores en definiciones repetitivas durante la implementación de los procesos. PDI nos permite añadir variables personalizadas y propias de nuestros desarrollos el archivo kettle.properties.

En nuestro caso utilizaremos dos variables. Una para almacenar la ruta de las fuentes de datos.

- Variable name: DIR_ENT
- Value: F:\Documents\Practica_Repository

Y otra dos para almacenar las cadenas de conexión a la base de datos.

- Variable name: STAGE
- Value: jdbc:sqlserver://UCS1R1UOCSDL01:1433;databaseName=DB_vnoronoz;integratedSecurity=false
- Variable name: BBDD
- Value: jdbc:sqlserver://UCS1R1UOCSDL01:1433;databaseName=DB_vnoronoz;integratedSecurity=false

La referencia a las variables de entorno durante la implementación de los procesos se realiza mediante llaves, de esta manera: {DIR_ENT}, {STAGE}, {BBDD}.

Existing variables		
Here is the list of defined variables:		
#	Name ^	Value
1	BBDD	jdbc:sqlserver://UCS1R1UOCSQL01:1433;databaseName= DB_vnoronoz;integratedSecurity=false
2	DIR_ENT	F:\Documents\Practica_Repository
3	STAGE	jdbc:sqlserver://UCS1R1UOCSQL01:1433;databaseName= DB_vnoronoz;integratedSecurity=false

OK Cancel

Carga y Transformaciones - Bloque IN

A continuación, se describe el desarrollo de las transformaciones para el Bloque IN del que ya hemos definido sus integrantes. En esta sección se verán en detalle los orígenes de datos, y los pasos realizados para la carga en las tablas intermedias, todo mediante PDI.

1) Transformación IN_EEUU

Inputs (1):

- Fichero formato JSON -> Estados_Unidos.json

The screenshot shows the 'File' tab of the 'JSON input' step configuration. The 'Step name' is set to 'Fichero Entrada JSON (ESTADOS UNIDOS)'. Under the 'Content' tab, the 'Source from field' section is visible, showing options like 'Source is from a previous step', 'Select field', 'Use field as file names', 'Read source as URL', and 'Do not pass field downstream'. The 'File or directory' section contains fields for 'Regular Expression' and 'Exclude Regular Expression'. Below these, the 'Selected files' table lists a single file: '\$(DIR_ENT)\Estados_Unidos.json'. The 'Fields' tab is also partially visible at the bottom.

Selección de los campos de interés de Estados_Unidos.json:

The screenshot shows the 'Fields' tab of the 'JSON input' step configuration. The 'Step name' is 'Fichero Entrada JSON (ESTADOS UNIDOS)'. The table lists seven fields with their corresponding paths and types:

#	Name	Path	Type	Format
1	Estado	\$.['Estado']	String	
2	Nombre oficial	\$.['Nombre oficial']	String	
3	Superficie (km2)	\$.['Superficie (km2)']	String	
4	Abrev.	\$.['Abrev.']	String	
5	Población (2010)	\$.['Población (2010)']	String	
6	Densidad de Población (hab/km2)	\$.['Densidad de Población (hab/km2)']	String	
7	Capital	\$.['Capital']	String	

Vista preliminar de los datos cargados:

Examine preview data

Rows of step: Fichero Entrada JSON (ESTADOS UNIDOS) (53 rows)

#	Estado	Nombre oficial	Superficie (km2)	Abrev.	Población (2010)	Densidad de Población (hab/km2)	Capital
1	Alabama	State of Alabama	135756	AL	4779736	35.21	Montgomery
2	Alaska	State of Alaska	1717854	AK	710231	0.41	Juneau
3	Arizona	State of Arizona	295254	AZ	6392017	21.65	Phoenix
4	Arkansas	State of Arkansas	137732	AR	2915918	21.17	Little Rock
5	California	State of California	423970	CA	37253956	87.87	Sacramento
6	North Carolina	State of North Carolina	139389	NC	9535483	68.41	Raleigh
7	South Carolina	State of South Carolina	82932	SC	4625364	55.77	Columbia
8	Colorado	State of Colorado	269601	CO	5029196	18.65	Denver
9	Connecticut	State of Connecticut	14357	CT	3574097	248.94	Hartford
10	North Dakota	State of North Dakota	183112	ND	672591	3.67	Bismarck
11	South Dakota	State of South Dakota	199731	SD	814180	4.08	Pierre
12	Delaware	State of Delaware	6447	DE	897934	139.28	Dover
13	Florida	State of Florida	170304	FL	18801310	110.4	Tallahassee
14	Georgia	State of Georgia	153909	GA	9687653	62.94	Atlanta
15	Hawaii	State of Hawaii	28311	HI	1360301	48.05	Honolulu
16	<null>	Moku'āina o Hawa'i	<null>	<null>	<null>	<null>	<null>
17	Idaho	State of Idaho	216446	ID	1567582	7.24	Boise
18	Illinois	State of Illinois	149998	IL	12830632	85.54	Springfield
19	Indiana	State of Indiana	94321	IN	6483802	68.74	Indianapolis
20	Iowa	State of Iowa	145743	IA	3046355	20.9	Des Moines
21	Kansas	State of Kansas	213096	KS	2853118	13.39	Topeka
22	Kentucky	Commonwealth of Kentucky	104659	KY	4339367	41.46	Frankfort
23	Louisiana	State of Louisiana	134264	LA	4533372	33.76	Baton Rouge
24	<null>	État de Louisiane	<null>	<null>	<null>	<null>	<null>
25	Maine	State of Maine	91646	ME	1328361	14.49	Augusta

Procesos (4):

- Borrar registros cuyo campo Estado sea null. **FILTER ROWS**
- Cambiar el tipo de dato de los campos Superficie, Población y Densidad de población. Son leídos como “string” pero se almacenarán como numéricos. **SELECT VALUES**
- A los campos de tipo string, como aplicación de buenas prácticas, los convertiremos a mayúsculas y quitaremos espacios en blanco del principio y del final. **STRING OPERATIONS**
- Ordenamos los datos por el campo Estado. **SORT ROWS**

Filter rows

Step name: **Filtrar Estado no sea NULL**

Send 'true' data to step: **String operations**

Send 'false' data to step:

The condition:

Estado IS NOT NULL

Step name: **Cambiar tipo de campos**

Select & Alter / Remove / Meta-data

Fields to alter the meta-data for :

#	Fieldname	Rename to	Type	Length
1	Superficie (km2)	supf	Number	
2	Población (2010)	pobl	Number	
3	Densidad de Población (hab/km2)	dens	Number	

All String operations

Step name: **String operations**

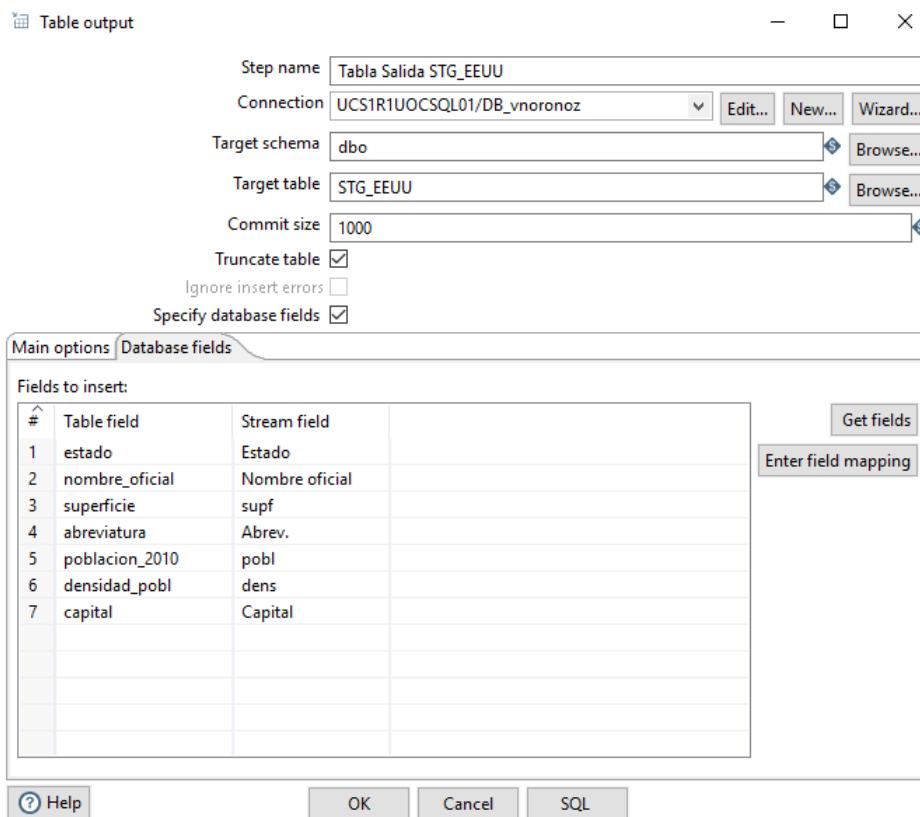
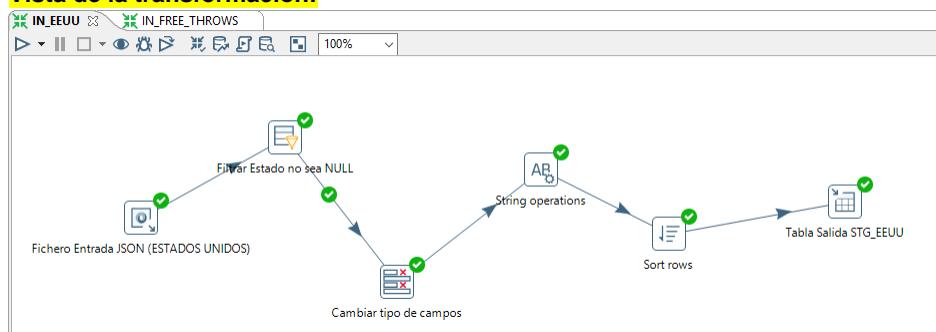
The fields to process:

#	In stream field	Out stream field	Trim type	Lower/Upper	Padding	Pad char	Pad Length	InitCap	Escape	Digits	Remove Special character
1	Estado		both	upper	none		N	None	none	none	
2	Nombre oficial		both	upper	none		N	None	none	none	
3	Abrev.		both	upper	none		N	None	none	none	
4	Capital		both	upper	none		N	None	none	none	

OK Get fields Cancel

Outputs (1):

- Tabla intermedia STG_EEUU

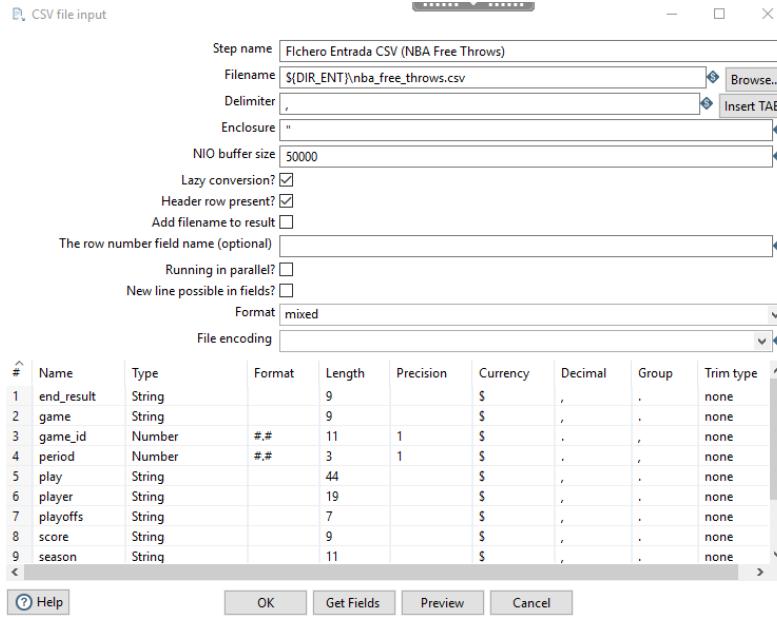
**Vista de la transformación:****Resultado de la ejecución:**

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	S
1	Fichero Entrada JSON (ESTADOS UNIDOS)	0	0	53	53	0	0	0	0	Finished	0.0s	
2	Filtrar Estado no sea NULL	0	53	50	0	0	0	0	0	Finished	0.0s	
3	Cambiar tipo de campos	0	50	50	0	0	0	0	0	Finished	0.0s	
4	String operations	0	50	50	0	0	0	0	0	Finished	0.0s	
5	Sort rows	0	50	50	0	0	0	0	0	Finished	0.1s	
6	Tabla Salida STG_EEUU	0	50	50	0	50	0	0	0	Finished	0.2s	

2) Transformación IN_FREE_THROWS

Inputs (1):

- Fichero formato CSV -> nba_free_throws.csv
Separado por comas



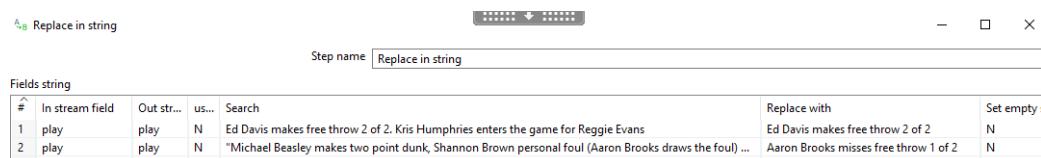
Vista preliminar de los datos cargados:

The screenshot shows a preview of the data with 1000 rows. The columns are: end_result, game, game_id, period, play, player, playoffs, score, season, shot_made, and time. The data includes various NBA players and their free throw statistics.

#	end_result	game	game_id	period	play	player	playoffs	score	season	shot_made	time
1	106 - 114	PHX - LAL	261031013	1	Andrew Bynum makes free throw 1 of 2	Andrew Bynum	regular	0 - 1	2006 - 2007	1	11:45
2	106 - 114	PHX - LAL	261031013	1	Andrew Bynum makes free throw 2 of 2	Andrew Bynum	regular	0 - 2	2006 - 2007	1	11:45
3	106 - 114	PHX - LAL	261031013	1	Andrew Bynum makes free throw 1 of 2	Andrew Bynum	regular	18 - 12	2006 - 2007	1	7:26
4	106 - 114	PHX - LAL	261031013	1	Andrew Bynum misses free throw 2 of 2	Andrew Bynum	regular	18 - 12	2006 - 2007	0	7:26
5	106 - 114	PHX - LAL	261031013	1	Shawn Marion makes free throw 1 of 1	Shawn Marion	regular	21 - 12	2006 - 2007	1	7:18
6	106 - 114	PHX - LAL	261031013	1	Amare Stoudemire makes free throw 1 of 2	Amare Stoudemire	regular	33 - 20	2006 - 2007	1	3:15
7	106 - 114	PHX - LAL	261031013	1	Amare Stoudemire makes free throw 2 of 2	Amare Stoudemire	regular	34 - 20	2006 - 2007	1	3:15
8	106 - 114	PHX - LAL	261031013	2	Leandro Barbosa misses free throw 1 of 2	Leandro Barbosa	regular	43 - 29	2006 - 2007	0	10:52
9	106 - 114	PHX - LAL	261031013	2	Leandro Barbosa makes free throw 2 of 2	Leandro Barbosa	regular	44 - 29	2006 - 2007	1	10:52
10	106 - 114	PHX - LAL	261031013	2	Lamar Odom makes free throw 1 of 2	Lamar Odom	regular	44 - 30	2006 - 2007	1	10:37
11	106 - 114	PHX - LAL	261031013	2	Lamar Odom makes free throw 2 of 2	Lamar Odom	regular	44 - 31	2006 - 2007	1	10:37
12	106 - 114	PHX - LAL	261031013	2	Lamar Odom makes free throw 1 of 1	Lamar Odom	regular	46 - 34	2006 - 2007	1	8:47
13	106 - 114	PHX - LAL	261031013	2	Smush Parker misses free throw 1 of 2	Smush Parker	regular	46 - 42	2006 - 2007	0	5:20
14	106 - 114	PHX - LAL	261031013	2	Smush Parker misses free throw 2 of 2	Smush Parker	regular	46 - 42	2006 - 2007	0	5:20
15	106 - 114	PHX - LAL	261031013	2	Vladimir Radmanovic misses free throw 1 of 2	Vladimir Radmanovic	regular	48 - 42	2006 - 2007	0	4:50
16	106 - 114	PHX - LAL	261031013	2	Vladimir Radmanovic misses free throw 2 of 2	Vladimir Radmanovic	regular	48 - 42	2006 - 2007	0	4:50
17	106 - 114	PHX - LAL	261031013	2	Maurice Evans misses free throw 1 of 2	Maurice Evans	regular	48 - 42	2006 - 2007	0	4:41
18	106 - 114	PHX - LAL	261031013	2	Maurice Evans makes free throw 2 of 2	Maurice Evans	regular	48 - 43	2006 - 2007	1	4:41
19	106 - 114	PHX - LAL	261031013	2	Leandro Barbosa misses free throw 1 of 2	Leandro Barbosa	regular	51 - 43	2006 - 2007	0	3:59
20	106 - 114	PHX - LAL	261031013	2	Leandro Barbosa makes free throw 2 of 2	Leandro Barbosa	regular	52 - 43	2006 - 2007	1	3:59
21	106 - 114	PHX - LAL	261031013	2	Shawn Marion makes free throw 1 of 2	Shawn Marion	regular	53 - 43	2006 - 2007	1	3:38
22	106 - 114	PHX - LAL	261031013	2	Shawn Marion makes free throw 2 of 2	Shawn Marion	regular	54 - 43	2006 - 2007	1	3:38
23	106 - 114	PHX - LAL	261031013	2	Amare Stoudemire misses free throw 1 of 2	Amare Stoudemire	regular	56 - 47	2006 - 2007	0	2:10
24	106 - 114	PHX - LAL	261031013	2	Amare Stoudemire misses free throw 2 of 2	Amare Stoudemire	regular	56 - 47	2006 - 2007	0	2:10
25	106 - 114	PHX - LAL	261031013	2	Andrew Bynum makes free throw 1 of 1	Andrew Bynum	regular	56 - 50	2006 - 2007	1	1:28

Procesos (2):

- Hay algunas cadenas de caracteres demasiado largas en el campo play, por lo que las recortamos a la información justa que necesitamos. **REPLACE IN STRING**
- A los campos de tipo string, como aplicación de buenas prácticas, los convertiremos a mayúsculas y quitaremos espacios en blanco del principio y del final. **STRING OPERATIONS**



String operations

Step name: String operations

The fields to process:

#	In stream field	Out stream field	Trim type	Lower/Upper	Padding	Pad char	Pad Length	InitCap	Escape	Digits	Remove Special character
1	end_result		both	upper	none			N	None	none	none
2	game		both	upper	none			N	None	none	none
3	play		both	upper	none			N	None	none	none
4	player		both	upper	none			N	None	none	none
5	playoffs		both	upper	none			N	None	none	none
6	score		both	upper	none			N	None	none	none
7	season		both	upper	none			N	None	none	none
8	time		both	upper	none			N	None	none	none

OK Get fields Cancel

Outputs (1):

- Tabla intermedia STG_NBA_free_throws

Table output

Step name: Tabla Salida STG_NBA_free_throws

Connection: UCS1R1UOCSQL01/DB_vnoronoz

Target schema: dbo

Target table: STG_NBA_free_throws

Commit size: 1000

Truncate table:

Ignore insert errors:

Specify database fields:

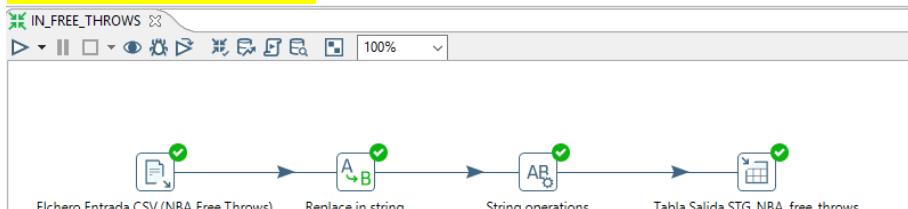
Main options Database fields

Fields to insert:

#	Table field	Stream field
1	end_result	end_result
2	game	game
3	game_id	game_id
4	period	period
5	play	play
6	player	player
7	playoffs	playoffs
8	score	score
9	season	season
10	shot_made	shot_made
11	time_	time

Get fields Enter field mapping

OK Cancel SQL

Vista de la transformación:**Resultado de la ejecución:**

Execution Results

Logging Execution History Step Metrics Performance Graph Metrics Preview data

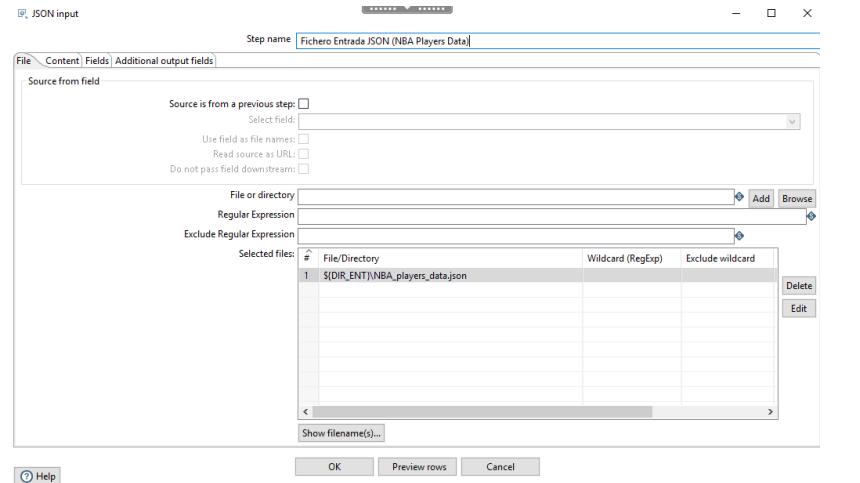
#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time
1	Fichero Entrada CSV (NBA Free Throw)	0	0	618019	618020	0	0	0	0	Finished	1mn 11s
2	Replace in string	0	618019	618019	0	0	0	0	0	Finished	1mn 12s
3	String operations	0	618019	618019	0	0	0	0	0	Finished	1mn 13s
4	Tabla Salida STG_NBA_free_throws	0	618019	618019	0	618019	0	0	0	Finished	1mn 14s

3) Transformación IN_PLAYERS_NBA

Inputs (2):

- Fichero formato JSON -> NBA_players_data.json
- Fichero formato XLS -> NBA_Players_List.xls
Hojas Players y Historic.

Para el fichero NBA_players_data.json:

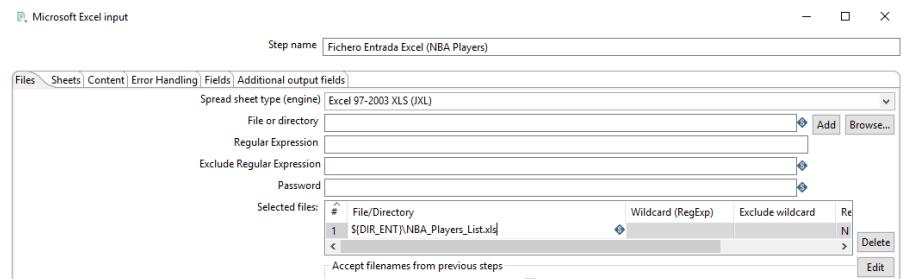


#	Name	Path	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	id	\$..id	String							none
2	birthDate	\$..birthDate	String							none
3	birthPlace	\$..birthPlace	String							none
4	State_Country	\$..State_Country	String							none
5	career_AST	\$..career_AST	String							none
6	career_FG%	\$..career_FG%	String							none
7	career_FG3%	\$..career_FG3%	String							none
8	career_FT%	\$..career_FT%	String							none
9	career_G	\$..career_G	String							none
10	career_PER	\$..career_PER	String							none
11	career PTS	\$..career PTS	String							none
12	career_TRB	\$..career_TRB	String							none
13	career_WS	\$..career_WS	String							none
14	career_eFG%	\$..career_eFG%	String							none
15	college	\$..college	String							none
16	height	\$..height	String							none
17	highSchool	\$..highSchool	String							none
18	name	\$..name	String							none
19	position	\$..position	String							none
20	shoots	\$..shoots	String							none
21	weight	\$..weight	String							none

Vista preliminar de los datos cargados:

#	id	birthDate	birthPlace	State_Country	career_AST	career_FG%	career_FG3%	career_FT%	career_G	career_PER	career PTS	career_TRB	career_WS	car
1	abdelal01	06/24/1968	Cairo	Egypt	3	502	0.0	70.1	256	13.0	3.3	48	50	
2	abdulsa01	04/07/1946	Brooklyn	New York	12	428	NULL	72.8	505	15.1	90	8.0	175	NL
3	abdulku01	04/16/1947	New York	New York	36	559	5.6	72.1	1560	24.6	246	11.2	2734	55
4	abdulma02	03/09/1969	Gulfport	Mississippi	35	442	35.4	90.5	586	15.4	146	1.9	252	47
5	abdulla01	11/03/1974	Maisons Alfort	France	11	417	23.7	70.3	236	11.4	78	3.3	35	42
6	abdush01	12/11/1976	Marietta	Georgia	25	472	29.7	81.0	830	19.0	181	7.5	712	47
7	abento01	05/06/1954	South Bend	Indiana	12	492	0.0	74.7	319	12.9	56	3.2	134	49
8	ablefo01	07/27/1932	NULL	NULL	10	0	NULL	NULL	1	-41.1	0	1.0	0	NL
9	abrampo01	02/09/1919	Etna	Pennsylvania	7	237	NULL	68.6	56	NULL	95	NULL	-19	NL
10	abrinial01	08/01/1993	Palma de Mallorca	Spain	5	387	36.8	88.0	174	8.8	53	1.4	50	52
11	ackeral01	01/21/1983	Compton	California	5	370	32.0	50.0	30	8.5	27	1.0	-2	41
12	ackerd01	09/04/1990	New York	New York	8	222	NULL	53.6	28	1.0	15	0.5	-6	NL
13	acresma01	11/15/1962	Inglewood	California	5	506	53.8	66.5	375	9.0	36	4.1	93	50
14	actonbu01	01/11/1942	NULL	NULL	5	392	NULL	65.5	23	9.5	33	2.0	1	NL
15	acyqu01	10/06/1990	Tyler	Texas	6	444	35.0	75.9	337	11.2	49	3.5	87	51
16	adamsa01	07/19/1954	Lawrence	Kansas	41	498	13.3	78.8	988	18.3	141	7.0	735	49
17	adamsdo01	11/27/1947	Atlanta	Georgia	19	402	0.0	74.1	523	10.7	88	5.6	117	40
18	adamsge01	05/15/1949	Kings Mountain	North Carolina	15	497	23.5	79.4	215	14.0	77	4.1	66	49
19	adamsjh01	06/20/1984	Inglewood	California	2	534	0.0	64.3	73	11.8	25	1.2	8	52
20	adamsjg01	05/04/1996	Baltimore	Maryland	19	345	33.8	77.8	34	7.6	32	1.8	1	45
21	adamsj01	07/08/1994	Atlanta	Georgia	6	402	38.5	60.7	32	13.1	32	0.9	4	45
22	adamsmj01	01/19/1963	Hartford	Connecticut	64	415	33.2	84.9	653	16.6	147	2.9	469	47
23	adamsmt01	07/20/1993	Rotorua	New Zealand	10	588	0.0	55.3	467	16.7	97	7.4	387	58
24	addirsa01	07/22/1964	Jersey City	New Jersey	9	449	28.2	77.2	379	10.5	58	2.1	49	46

Para el fichero NBA_Players_List.xls:



List of sheets to read			
#	Sheet name	Start row	Start column
1	Players	0	0

Fields										
#	Name	Type	Length	Precision	Trim type	Repeat	Format	Currency	Decimal	Grouping
1	player	String			none	N				

Vista preliminar de los datos cargados:

Examine preview data	
Rows of step: Fichero Entrada Excel (NBA Players) (548 rows)	
#	player
1	A
2	Adams, Steven
3	Adebayo, Bam
4	Aldridge, LaMarcus
5	Alexander, Kyle
6	Alexander-Walker, Nickeil
7	Allen, Grayson
8	Allen, Jarrett
9	Allen, Kadeem
10	Aminu, Al-Farouq
11	Anderson, Justin
12	Anderson, Kyle
13	Anderson, Ryan
14	Antetokounmpo, Giannis
15	Antetokounmpo, Kostas
16	Antetokounmpo, Thanasis
17	Anthony, Carmelo
18	Anunoby, OG
19	Arcidiacono, Ryan
20	Ariza, Trevor
21	Augustin, D.J.
22	Ayton, Deandre
23	B
24	Bacon, Dwyane
25	Bagley III, Marvin

Por una parte se ha hecho la carga de la hoja Players y de forma análoga la de la hoja Historic.

La carga de las hojas se ha hecho por separado para filtrar por activo/inactivo durante la transformación como más adelante se verá en el desarrollo del proceso.

Procesos ():

Para el fichero NBA_players_data.json (4):

- Como del json vienen todos los datos entre comillas, se leen como tipo string, entonces cambiamos a tipo fecha y tipo numérico los campos que corresponden. **SELECT VALUES**
- Se han controlado los valores nulos porque de entrada daban error al pasarlo a números. Así que todos los campos con valor “NULL” se han transformado a null de PDI y los que están en campo numérico se ha dejado el campo vacío. **NULL IF & IF FIELD VALUE IS NULL**
- A los campos de tipo string, como aplicación de buenas prácticas, los convertiremos a mayúsculas y quitaremos espacios en blanco del principio y del final. **STRING OPERATIONS**
- Ordenamos los datos por el campo Id. **SORT ROWS**

The screenshot shows the PDI transformation interface with three main steps:

- Step 1: Null if** (Step name: Null if)

#	Name	Value to turn to NULL
1	id	NULL
2	birthDate	NULL
3	birthPlace	NULL
4	State_Country	NULL
5	career_AST	NULL
6	career_FG%	NULL
7	career_FG3%	NULL
8	career_FT%	NULL
9	career_G	NULL
10	career_PER	NULL
11	career PTS	NULL
12	career_TRB	NULL
13	career_WS	NULL
14	career_eFG%	NULL
15	college	NULL
16	height	NULL
17	highSchool	NULL
18	name	NULL
19	position	NULL
20	shoots	NULL
21	weight	NULL
- Step 2: If field value is null** (Step name: If field value is null)

Replace Null for all fields			
Replace by value	Set empty string?	Mask (Date)	
	<input checked="" type="checkbox"/>		
Select fields	<input type="checkbox"/>		
Select value type	<input type="checkbox"/>		
- Step 3: Select values** (Step name: Cambiar tipo de campo)

Fields to alter the meta-data for :						
#	Fieldname	Rename to	Type	Length	Precision	Binary
1	birthDate	birthDate2	Date			
2	career_AST	career_AST2	Number			
3	career_FG%	career_FG%2	Number			
4	career_FG3%	career_FG3%2	Number			
5	career_FT%	career_FT%2	Number			
6	career_G	career_G2	Number			
7	career_PER	career_PER2	Number			
8	career PTS	career PTS2	Number			
9	career_TRB	career_TRB2	Number			
10	career_WS	career_WS2	Number			
11	career_eFG%	career_eFG%2	Number			

#	In stream field	Out stream field	Trim type	Lower/Upper	Padding	Pad char	Pad Length	InitCap	Escape	Digits	Remove Special character
1	id		both	upper	none			N	None	none	none
2	birthPlace		both	upper	none			N	None	none	none
3	State_Country		both	upper	none			N	None	none	none
4	college		both	upper	none			N	None	none	none
5	height		both	upper	none			N	None	none	none
6	highSchool		both	upper	none			N	None	none	none
7	name		both	upper	none			N	None	none	none
8	position		both	upper	none			N	None	none	none
9	shoots		both	upper	none			N	None	none	none

OK Get fields Cancel

Para el fichero NBA_Players_List.xls (8):

- Los nombre de los jugadores vienen antecedidos por un campo que guarda la letra por la que empiezan. Para eliminar ese campo, de las letras (A,B,C,D...), lo que se ha hecho es calcular la longitud del campo, y si el campo tiene longitud =1 (un solo carácter), se elimina mediante Filter Rows. **CALCULATOR + FILTER ROWS**
- Como el archivo viene estructurado de la forma: "Apellido, Nombre" y en las otras fuentes tenemos "Nombre Apellido", lo que vamos a hacer es ponerlo en esa estructura. Para ello, separaremos el campo player en apellido y nombre. **SPLIT FIELDS**
- Después concatenamos los campos separados de la forma nombre apellido y lo guardamos en un nuevo campo nombre_player. **CONCAT FIELDS**
- A los campos de tipo string, como aplicación de buenas prácticas, los convertiremos a mayúsculas y quitaremos espacios en blanco del principio y del final. **STRING OPERATIONS**
- Ordenamos los datos por el campo nombre_player. **SORT ROWS**
- Este proceso se aplica por separado a la hoja PLAYERS que contiene los jugadores en activo y a la hoja HISTORIC que contiene conjuntamente a los jugadores en activo y a los inactivos. Como se tiene información duplicada, se ha hecho la comparación de ambas tablas con la herramienta **MERGE ROWS DIFF**.
- Si el resultado está en ambas tablas, la herramienta arroja un "identical", y si solo se encuentra en la de historic arroja "new". Con estos datos podemos definir mediante valores que si el valor es identico en las dos tablas está en activo e inactivo si sucede lo contrario. **VALUE MAPPER**

Step name: Calcular longitud string

Fields:

#	New field	Calculation	Field A	Field B	Field C	Value type
1	long	Return the length of a string A player				None

Help

Filter rows

Step name: Eliminar filas con solo 1 carácter

Send 'true' data to step: Sort rows 2

Send 'false' data to step:

The condition:

NOT
long = 1 (Integer)

Split fields

Step name	separar apell nom
Field to split	player
Delimiter	,
Enclosure	

Fields

#	New field	ID	Remove ID?	Type	Length	Precision	Format
1	apellido	N		String			
2	nombre	N		String			

Concat fields

Step name	intercambiar nom ap
Target Field Name	nombre_player
Length of Target Field	0
Separator	
Enclosure	

Fields Advanced

#	Name	Type	Format	Length	Precision
1	nombre	None			
2	apellido	None			

String operations

Step name	String operations						
The fields to process:							
#	In stream field	Out stream field	Trim type	Lower/Upper	Padding	Pad char	P
1	nombre_player		both	upper	none		

Sort rows

Step name	Sort rows
Sort directory	%%java.io.tmpdir'
TMP-file prefix	out
Sort size (rows in memory)	1000000
Free memory threshold (in %)	

Compress TMP Files?

Only pass unique rows? (verifies keys only)

Fields :

#	Fieldname	Ascending	Case sensitive compare?	So
1	nombre_player	Y	N	N

Merge rows (diff)

Step name: Merge rows (diff)
 Reference rows origin: Sort rows
 Compare rows origin: Sort rows 2
 Flag fieldname: flagfield

Keys to match :		Values to compare :	
#	Key field	#	Value field
1	nombre_player	1	nombre_player

Get key fields **Get value fields**

Help **OK** **Cancel**

Value mapper

Step name: ACTIVO O NO
 Fieldname to use: flagfield
 Target field name (empty=overwrite): active
 Default upon non-matching:

#	Source value	Target value
1	new	NO
2	identical	YES

Outputs (2):

- Tabla intermedia STG_NBA_Players_data

Table output

Step name: Tabla Salida STG_NBA_Players_data
 Connection: UCS1R1UOCSQL01/DB_vnoronoz
 Target schema: dbo
 Target table: STG_NBA_Players_data
 Commit size: 1000
 Truncate table:
 Ignore insert errors:
 Specify database fields:

Main options Database fields

Fields to insert:

#	Table field	Stream field
1	id_	id
2	birthDate	birthDate2
3	birthPlace	birthPlace
4	career_AST	career_AST2
5	career_FG%	career_FG%2
6	career_FG3%	career_FG3%2
7	career_FG3%	career_FT%2
8	career_G	career_G2
9	career_PER	career_PER2
10	career PTS	career PTS2
11	career TRB	career TRB2
12	career WS	career WS2
13	career eFG%	career eFG%2
14	college	college
15	height	height

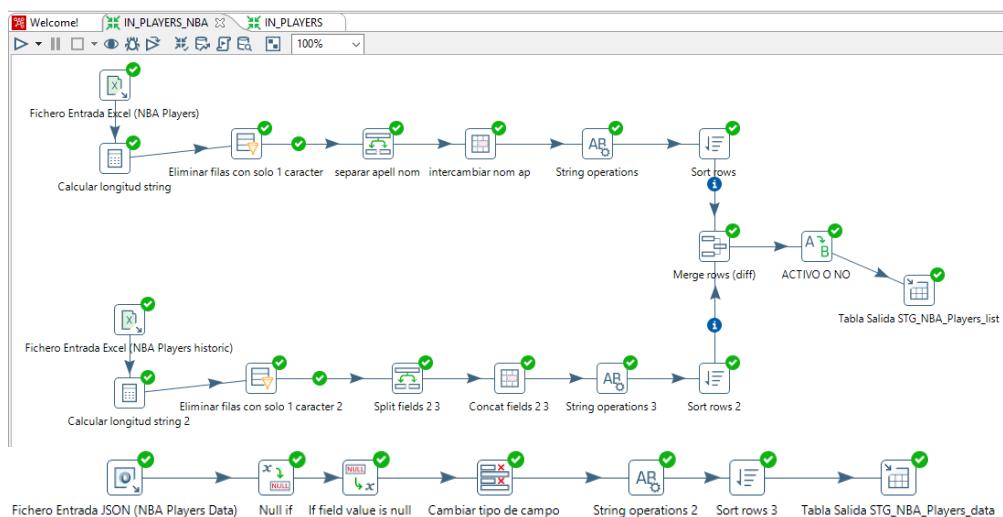
Get fields **Enter field mapping**

Help **OK** **Cancel** **SQL**

– Tabla intermedia STG_NBA_Players_list

#	Table field	Stream field
1	player	nombre_player
2	active	active

Vista de la transformación:



Resultado de la ejecución:

Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)
1 Fichero Entrada Excel (NBA Players historic)	0	0	4495	4495	0	0	0	0	Finished	5.1s	883
2 Fichero Entrada Excel (NBA Players)	0	0	548	548	0	0	0	0	Finished	3.8s	146
3 Fichero Entrada JSON (NBA Players Data)	0	0	4639	4639	0	0	0	0	Finished	5.6s	826
4 Calcular longitud string	0	548	548	0	0	0	0	0	Finished	5.1s	108
5 Calcular longitud string 2	0	4495	4495	0	0	0	0	0	Finished	5.2s	862
6 Eliminar filas con solo 1 carácter 2	0	4495	4470	0	0	0	0	0	Finished	5.3s	843
7 Split fields 2 3	0	4470	4470	0	0	0	0	0	Finished	5.4s	831
8 Eliminar filas con solo 1 carácter	0	548	525	0	0	0	0	0	Finished	5.1s	107
9 separar apell nom	0	525	525	0	0	0	0	0	Finished	5.2s	101
10 Null if	0	4639	4639	0	0	0	0	0	Finished	5.7s	809
11 If field value is null	0	4639	4639	0	0	0	0	0	Finished	6.0s	780
12 intercambiar nom ap	0	525	525	0	0	0	0	0	Finished	5.3s	98
13 Concat fields 2 3	0	4470	4470	0	0	0	0	0	Finished	5.4s	829
14 String operations 3	0	4470	4470	0	0	0	0	0	Finished	5.4s	829
15 String operations	0	525	525	0	0	0	0	0	Finished	5.4s	98
16 Sort rows 2	0	4470	4470	0	0	0	0	0	Finished	5.4s	821
17 Sort rows	0	525	525	0	0	0	0	0	Finished	5.4s	97
18 Merge rows (diff)	0	4995	4470	0	0	0	0	0	Finished	6.1s	819
19 Cambiar tipo de campo	0	4639	4639	0	0	0	0	0	Finished	6.1s	766
20 String operations 2	0	4639	4639	0	0	0	0	0	Finished	6.1s	765
21 Sort rows 3	0	4639	4639	0	0	0	0	0	Finished	6.1s	763
22 Tabla Salida STG_NBA_Players_data	0	4639	4639	0	4639	0	0	0	Finished	6.5s	714
23 ACTIVO O NO	0	4470	4470	0	0	0	0	0	Finished	6.1s	732
24 Tabla Salida STG_NBA_Players_list	0	4470	4470	0	4470	0	0	0	Finished	6.2s	725

4) Transformación IN_PLAYERS_WNBA

Inputs (1):

- Fichero formato TXT -> WNBA_Players_List.txt
Separado por TAB

The top screenshot shows the 'File or directory' field set to 'S(DIR_ENT)\WNBA_Players_List.txt' and the 'Regular Expression' field left empty. The bottom screenshot shows the 'Fields' tab with two columns: 'Player' and 'Active'.

#	Name	Type	Format	Position	Length	Precision	Currency	Decimal	Group	Null if	Default	Trim type	Repeat
1	Player	String		22		.	\$,	-	N	both	N	
2	Active	String		3		.	\$,	-	N	both	N	

Vista preliminar de los datos cargados:

Examine preview data

Rows of step: Fichero Entrada TXT (WNBA Players) (963 rows)

#	Player	Active
1	Abdi, Farhiya	NO
2	Abraham, Tajama	NO
3	Abrosimova, Svetlana	NO
4	Achonwa, Natalie	YES
5	Adair, Jessica	NO
6	Adams, Danielle	NO
7	Adams, Jordan	NO
8	Aguilar, Elisa	NO
9	Ajavon, Matee	NO
10	Akhator, Evelyn	NO
11	Alberts, Marcie	NO
12	Aldridge, Markita	NO
13	Alexander, Erin	NO
14	Alexander, Kayla	YES
15	Alhaleem, Towana	NO
16	Allemand, Julie	YES
17	Allen, Charel	NO
18	Allen, Lindsay	YES
19	Allen, Rebecca	YES
20	Alleyne, Jillian	YES
21	Alston, Ameryst	NO
22	Amachree, Mactabene	NO
23	Ambers, Monique	NO
24	Anderson, Ambrosia	NO
25	Anderson, Chantelle	NO

Procesos (4):

- Como el archivo viene estructurado de la forma: "Apellido, Nombre" y en las otras fuentes tenemos "Nombre Apellido", lo que vamos a hacer es ponerlo en esa estructura. Para ello, separaremos el campo player en apellido y nombre. **SPLIT FIELDS**
- Despues concatenamos los campos separados de la forma nombre apellido y lo guardamos en un nuevo campo nombre_player. **CONCAT FIELDS**
- A los campos de tipo string, como aplicación de buenas prácticas, los convertiremos a mayúsculas y quitaremos espacios en blanco del principio y del final. **STRING OPERATIONS**

- Ordenamos los datos por el campo nombre_player. **SORT ROWS**

Split fields

Step name: Separar nombre y apellido
 Field to split: Player
 Delimiter: ,
 Enclosure:

#	New field	ID	Remove ID?	Type	Length	Precision	Format	Gr
1	apellido	N		String				
2	nombre	N		String				

Concat fields

Step name: Unir cambiando orden
 Target Field Name: nombre_player
 Length of Target Field: 0
 Separator:
 Enclosure:

#	Name	Type	Format	Length
1	nombre	None		
2	apellido	None		

Outputs (3):

- Tabla intermedia STG_WNBA_Players_list

Table output

Step name: Tabla Salida STG_WNBA_Players_list
 Connection: UCS1R1UOCSQL01/DB_vnoronoz
 Target schema: dbo
 Target table: STG_WNBA_Players_list
 Commit size: 1000
 Truncate table:
 Ignore insert errors:
 Specify database fields:

Main options Database fields

Partition data over tables:
 Partitioning field:
 Partition data per month:
 Partition data per day:
 Use batch update for inserts:
 Is the name of the table defined in a field?
 Field that contains name of table:
 Store the tablename field:
 Return auto-generated key:
 Name of auto-generated key field:

Buttons: Help, OK, Cancel, SQL

Vista de la transformación:**Resultado de la ejecución:**

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)
1	Fichero Entrada TXT (WNBA Players)	0	0	963	964	0	1	0	0	Finished	0.2s	5,155
2	Separar nombre y apellido	0	963	963	0	0	0	0	0	Finished	0.3s	3,199
3	Unir cambiando orden	0	963	963	0	0	0	0	0	Finished	0.3s	3,038
4	String operations 2.2	0	963	963	0	0	0	0	0	Finished	0.3s	2,849
5	Sort rows 3 2	0	963	963	0	0	0	0	0	Finished	0.4s	2,495
6	Tabla Salida STG_WNBA_Players_list	0	963	963	0	963	0	0	0	Finished	0.5s	1,831

5) Transformación IN_STATS**Inputs (2):**

- Fichero formato CSV -> NBA_Seasons_Stats_1950_2017.csv
Separado por comas
- Fichero formato XLS -> WNBA_Seasons_Stats_2005_2017.xls
Hoja Stats.

Para el fichero NBA_Seasons_Stats_1950_2017.csv:

CSV file input

Step name: Fichero Entrada CSV (NBA)

Filename: \${DIR_ENT}\NBA_Seasons_Stats_1950_2017.csv

Delimiter: ,

Enclosure: "

NIO buffer size: 50000

Lazy conversion?

Header row present?

Add filename to result

The row number field name (optional):

Running in parallel?

New line possible in fields?

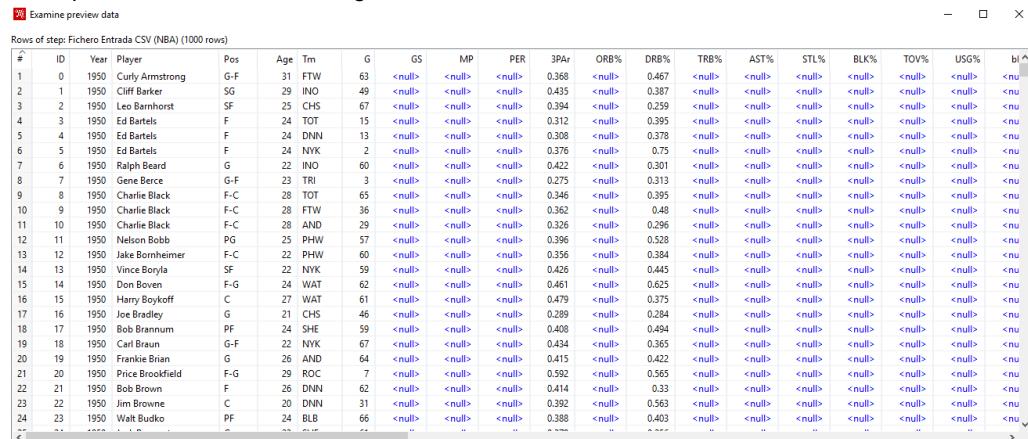
Format: mixed

File encoding:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	ID	Integer	#	15	0	\$.	,	none
2	Year	Integer	#	15	0	\$.	,	none
3	Player	String		100		\$.	,	none
4	Pos	String		100		\$.	,	none
5	Age	Integer	#	15	0	\$.	,	none
6	Tm	String		100		\$.	,	none
7	G	Integer	#	15	0	\$.	,	none
8	GS	Integer	#	15	0	\$.	,	none
9	MP	Integer	#	15	0	\$.	,	none
10	PER	Number	,###			\$.	,	none

Help OK Get Fields Preview Cancel

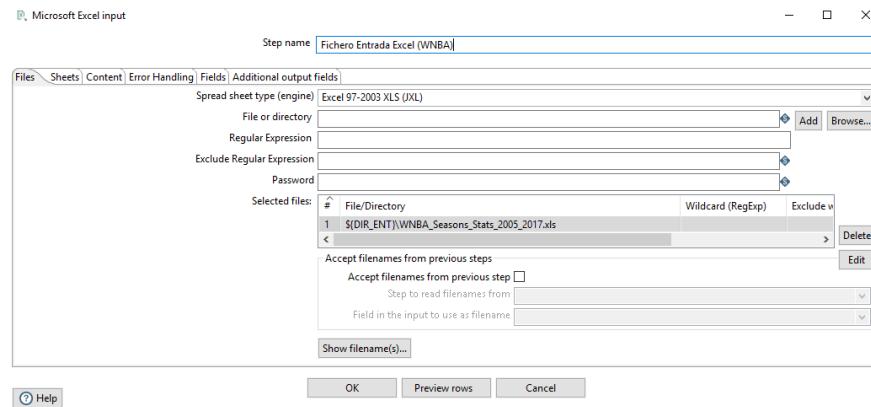
Vista preliminar de los datos cargados:



Rows of step: Fichero Entrada CSV (NBA) (1000 rows)

#	ID	Year	Player	Pos	Age	Tm	G	GS	MP	PER	3PAr	ORB%	DRB%	TRB%	AST%	STL%	BLK%	TOV%	USG%	bl
1	0	1950	Curly Armstrong	G-F	31	FTW	63	<null>	<null>	0.368	0.467	<null>	<nu>							
2	1	1950	Cliff Barker	SG	29	INO	49	<null>	<null>	0.435	0.387	<null>	<nu>							
3	2	1950	Leo Barnhorst	SF	25	CHS	67	<null>	<null>	0.394	0.259	<null>	<nu>							
4	3	1950	Ed Bartels	F	24	TOT	15	<null>	<null>	0.312	0.395	<null>	<nu>							
5	4	1950	Ed Bartels	F	24	DNN	13	<null>	<null>	0.308	0.378	<null>	<nu>							
6	5	1950	Ed Bartels	F	24	NYK	2	<null>	<null>	0.376	0.75	<null>	<nu>							
7	6	1950	Ralph Beard	G	22	INO	60	<null>	<null>	0.422	0.301	<null>	<nu>							
8	7	1950	Gene Berce	G-F	23	TRI	3	<null>	<null>	0.275	0.313	<null>	<nu>							
9	8	1950	Charlie Black	F-C	28	TOT	65	<null>	<null>	0.346	0.395	<null>	<nu>							
10	9	1950	Charlie Black	F-C	28	FTW	36	<null>	<null>	0.362	0.48	<null>	<nu>							
11	10	1950	Charlie Black	F-C	28	AND	29	<null>	<null>	0.326	0.296	<null>	<nu>							
12	11	1950	Nelson Bobb	PG	25	PHW	57	<null>	<null>	0.396	0.528	<null>	<nu>							
13	12	1950	Jake Bornheimer	F-C	22	PHW	60	<null>	<null>	0.356	0.384	<null>	<nu>							
14	13	1950	Vince Boryni	SF	22	NYK	59	<null>	<null>	0.426	0.445	<null>	<nu>							
15	14	1950	Don Boven	F-G	24	WAT	62	<null>	<null>	0.461	0.625	<null>	<nu>							
16	15	1950	Harry Boykoff	C	27	WAT	61	<null>	<null>	0.479	0.375	<null>	<nu>							
17	16	1950	Joe Bradley	G	21	CHS	46	<null>	<null>	0.289	0.284	<null>	<nu>							
18	17	1950	Bob Brannum	PF	24	SHE	59	<null>	<null>	0.408	0.494	<null>	<nu>							
19	18	1950	Carl Braun	G-F	22	NYK	67	<null>	<null>	0.434	0.365	<null>	<nu>							
20	19	1950	Frankie Brian	G	26	AND	64	<null>	<null>	0.415	0.422	<null>	<nu>							
21	20	1950	Price Brookfield	F-G	29	ROC	7	<null>	<null>	0.592	0.565	<null>	<nu>							
22	21	1950	Bob Brown	F	26	DNN	62	<null>	<null>	0.414	0.33	<null>	<nu>							
23	22	1950	Jim Browne	C	20	DNN	31	<null>	<null>	0.392	0.563	<null>	<nu>							
24	23	1950	Walt Budko	PF	24	BLB	66	<null>	<null>	0.388	0.403	<null>	<nu>							

Para el fichero WNBA_Seasons_Stats_2005_2017.xls:



Step name: Fichero Entrada Excel (WNBA)

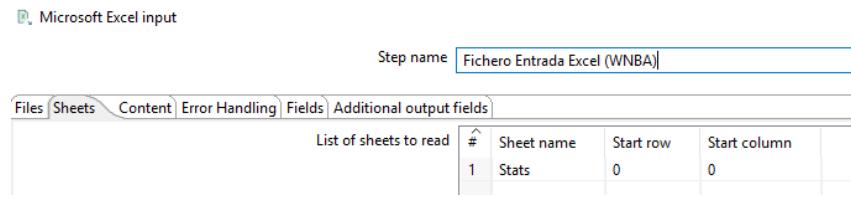
Selected files:

#	File/Directory	Wildcard (RegEx)	Exclude v
1	\$(DIR_ENT)\WNBA_Seasons_Stats_2005_2017.xls		

Accept filenames from previous steps Step to read filenames from:

Field in the input to use as filename:

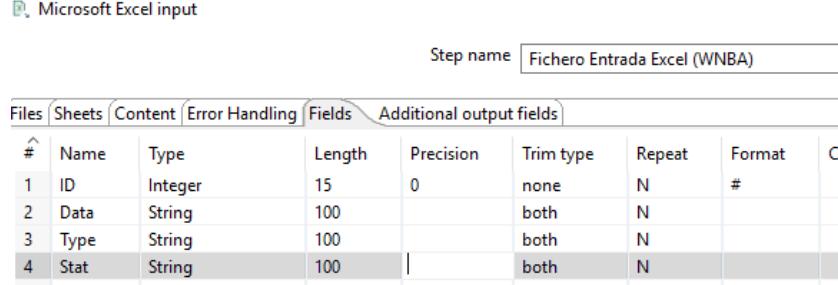
OK Preview rows Cancel Help



Step name: Fichero Entrada Excel (WNBA)

List of sheets to read:

#	Sheet name	Start row	Start column
1	Stats	0	0



#	Name	Type	Length	Precision	Trim type	Repeat	Format	C
1	ID	Integer	15	0	none	N	#	
2	Data	String	100		both	N		
3	Type	String	100		both	N		
4	Stat	String	100		both	N		

Vista preliminar de los datos cargados:

Examine preview data

Rows of step: Fichero Entrada Excel (WNBA) (1000 rows)

#	ID	Data	Type	Stat
1	20171	2017	Number	SEASON
2	20171	1	Number	#
3	20171	TINA CHARLES	String	PLAYER
4	20171	NYL	String	TEAM
5	20171	29	Number	AGE
6	20171	34	Number	GP
7	20171	22	Number	W
8	20171	12	Number	L
9	20171	1095	Number	MIN
10	20171	671	Number	PTS
11	20171	262	Number	FGM
12	20171	593	Number	FGA
13	20171	44.2	Number	FG%
14	20171	24	Number	3PM
15	20171	69	Number	3PA
16	20171	34.8	Number	3P%
17	20171	123	Number	FTM
18	20171	153	Number	FTA
19	20171	80.4	Number	FT%
20	20171	69	Number	OREB
21	20171	249	Number	DREB
22	20171	318	Number	REB
23	20171	87	Number	AST
24	20171	80	Number	TOV
25	20171	28	Number	STL

Procesos ():

Para el fichero NBA_Seasons_Stats_1950_2017.csv (4):

- Se renombran los campos para cumplir los requisitos de SQL Server y se borran dos campos que están vacíos. **SELECT VALUES**
- Contorlamos que el campo player no tenga valor null. **FILTER ROWS**
- A los campos de tipo string, como aplicación de buenas prácticas, los convertiremos a mayúsculas y quitaremos espacios en blanco del principio y del final. **STRING OPERATIONS**
- Ordenamos los datos por el campo Id. **SORT ROWS**



Step name **Renombrar/Borrar Campos**

Select & Alter Remove Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision
25	WS			
26	WS/48	WSx48		
27	blank2			
28	OBPM			
29	DBPM			
30	BPM			
31	VORP			
32	FG			
33	FGA			
34	FG%			
35	3P	_3P		
36	3PA	_3PA		
37	3P%	_3P%		
38	2P	_2P		
39	2PA	_2PA		
40	2P%	_2P%		
41	eFG%			
42	FT			

The screenshot shows the Informatica PowerCenter interface with three main steps:

- Select values**: Step name: Renombrar/Borrar Campos. It lists fields to remove: blank and blank2.
- Filtrar registros**: Step name: Filtrar registros. It filters rows where player IS NOT NULL.
- String operations**: Step name: String operations. It processes fields: player, pos, and Tm, applying upper case conversion and removing special characters.

Para el fichero WNBA_Seasons_Stats_2005_2017.xls (2):

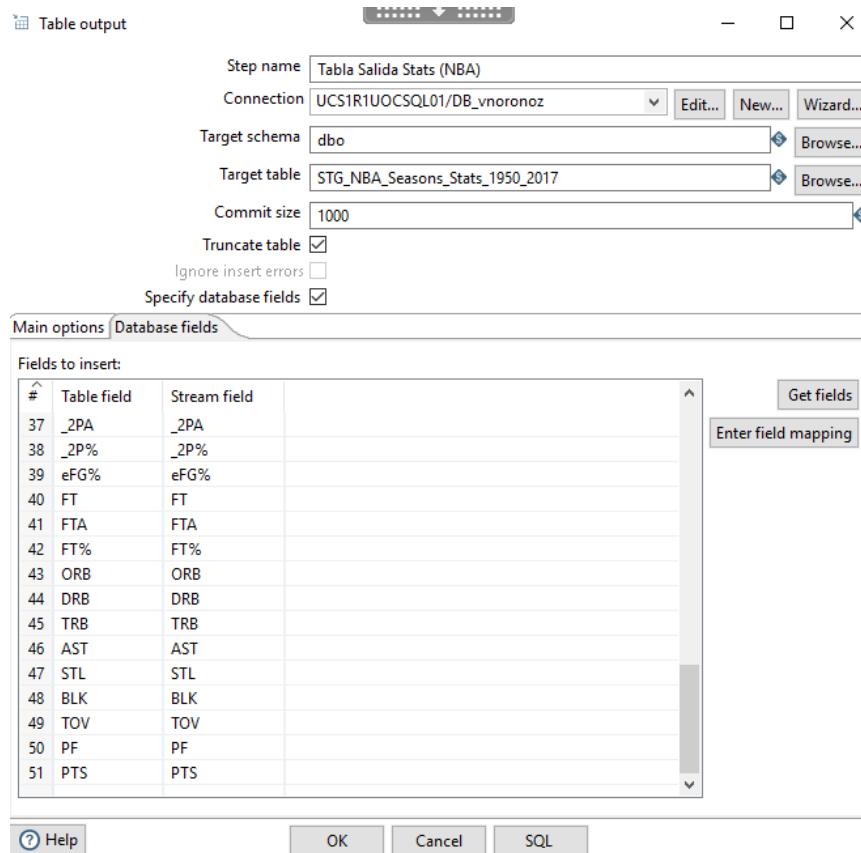
- A los campos de tipo string, como aplicación de buenas prácticas, los convertiremos a mayúsculas y quitaremos espacios en blanco del principio y del final. **STRING OPERATIONS**
- Ordenamos los datos por el campo ID. **SORT ROWS**

The screenshot shows the 'String operations' dialog box with the following settings:

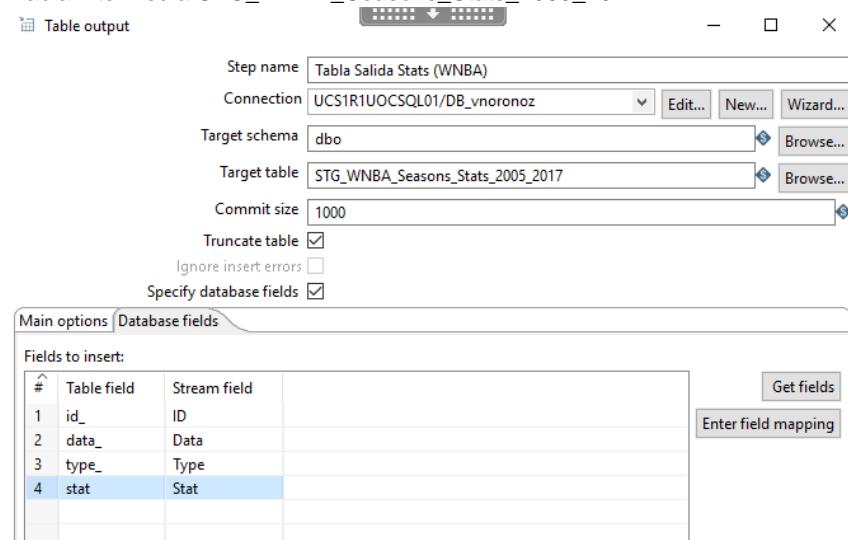
#	In stream field	Out stream field	Trim type	Lower/Upper	Padding	Pad char	Pad Length	InitCap	Escape	Digits	Remove Special character
1	Data		both	upper	none			N	None	none	none
2	Type		both	upper	none			N	None	none	none
3	Stat		both	upper	none			N	None	none	none

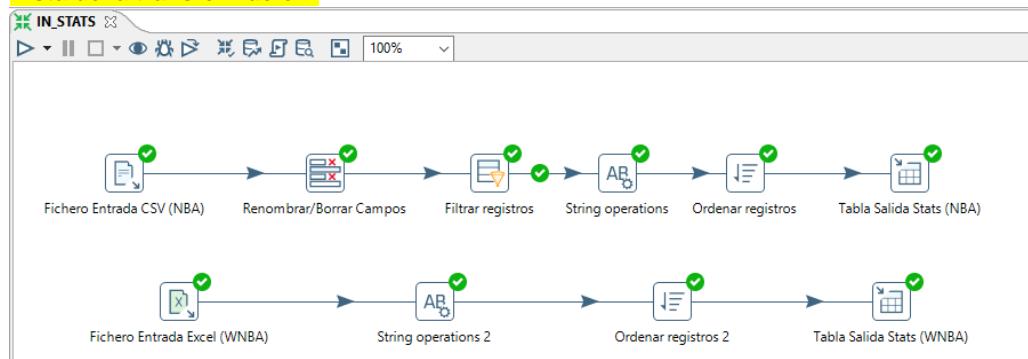
Outputs (2):

- Tabla intermedia STG_NBA_Season_Stats_1950_2017



- Tabla intermedia STG_WNBA_Seasons_Stats_2005_2017

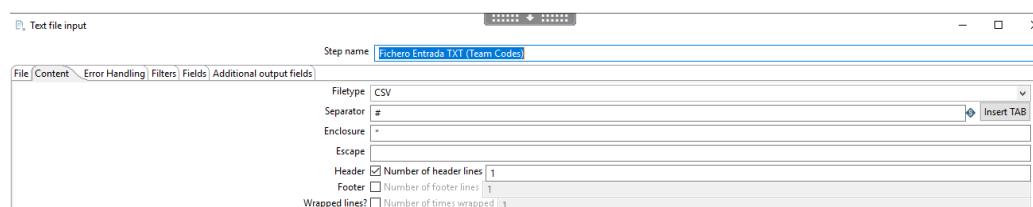
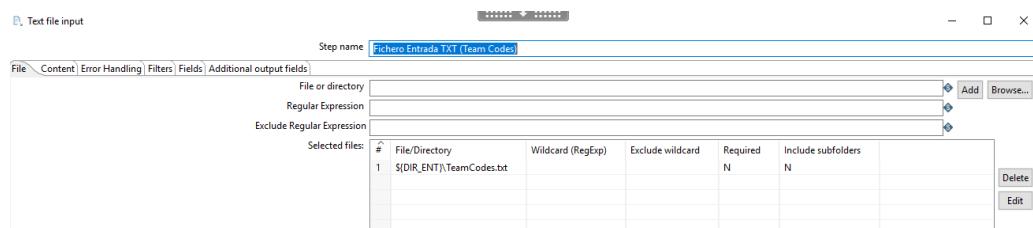


Vista de la transformación:**Resultado de la ejecución:**

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)
1	Fichero Entrada Excel (WNBA)	0	0	64635	64635	0	0	0	0	Finished	2.7s	24,082
2	Fichero Entrada CSV (NBA)	0	0	24691	24692	0	0	0	0	Finished	3.1s	8,014
3	Renombrar/Borrar Campos	0	24691	24691	0	0	0	0	0	Finished	3.1s	7,868
4	Filtrar registros	0	24691	24624	0	0	0	0	0	Finished	3.1s	7,856
5	String operations 2	0	64635	64635	0	0	0	0	0	Finished	2.7s	23,930
6	String operations	0	24624	24624	0	0	0	0	0	Finished	3.1s	7,820
7	Ordenar registros	0	24624	24624	0	0	0	0	0	Finished	4.9s	4,987
8	Tabla Salida Stats (NBA)	0	24624	24624	0	24624	0	0	0	Finished	6.2s	3,982
9	Ordenar registros 2	0	64635	64635	0	0	0	0	0	Finished	5.4s	12,036
10	Tabla Salida Stats (WNBA)	0	64635	64635	0	64635	0	0	0	Finished	5.8s	11,129

6) Transformación IN_TEAMS**Inputs (3):**

- Fichero formato TXT -> TeamCodes.txt
Separado por #
- Fichero formato XML -> NBA_Teams.xml
- Fichero formato XLS -> WNBA_Teams.xls
Hoja Teams.

Para el fichero TeamCodes.txt:

Text file input

Step name: Fichero Entrada TXT (Team Codes)

#	Name	Type	Format	Position	Length	Precision	Currency	Decimal	Group	Null if	Default	Trim type	Repeat
1	League	String			4		\$.	,	-		both	N
2	Team	String			22		\$.	,	-		both	N
3	Code	String			4		\$.	,	-		both	N

Vista preliminar de los datos cargados:

Examine preview data

Rows of step: Fichero Entrada TXT (Team Codes) (42 rows)

#	League	Team	Code
1	WNBA	Atlanta Dream	ADR
2	NBA	Atlanta Hawks	ATL
3	NBA	Boston Celtics	BOS
4	NBA	Brooklyn Nets	BN
5	NBA	Charlotte Hornets	CHA
6	NBA	Chicago Bulls	CHI
7	WNBA	Chicago Sky	CHS
8	NBA	Cleveland Cavaliers	CLE
9	WNBA	Connecticut Sun	CON
10	NBA	Dallas Mavericks	DAL
11	WNBA	Dallas Wings	DWN
12	NBA	Denver Nuggets	DEN
13	NBA	Detroit Pistons	DET
14	NBA	Golden State Warriors	GS
15	NBA	Houston Rockets	HOU
16	WNBA	Indiana Fever	FVR
17	NBA	Indiana Pacers	IND
18	WNBA	Las Vegas Aces	LVA
19	NBA	Los Angeles Clippers	LAC
20	NBA	Los Angeles Lakers	LAL
21	WNBA	Los Angeles Sparks	LAS
22	NBA	Memphis Grizzlies	MEM
23	NBA	Miami Heat	MIA
24	NBA	Milwaukee Bucks	MIL
25	WNBA	Minnesota Lynx	MLY

Para el fichero NBA_Teams.xml:

Get data from XML

Step name: Get data from XML (NBA Teams)

#	File/Directory	Wildcard (RegExp)	Exclude wildcard	Required	Include subfolders
1	\$(DIR_ENT)\NBA_Teams.xml			N	N

Get data from XML

Step name: Get data from XML (NBA Teams)

Settings
Loop XPath: /NBATeams/Row
Encoding: UTF-8
Namespace aware? <input type="checkbox"/>
Ignore comments? <input type="checkbox"/>

 Get data from XML

Step name: Get data from XML (NBA Teams)

	Name	XPath	Element	Result type	Type	Format	Length
1	Conferencia	Data[1]	Node	Value of	String		
2	Division	Data[2]	Node	Value of	String		
3	Equipo	Data[3]	Node	Value of	String		
4	Ciudad	Data[4]	Node	Value of	String		
5	Estado	Data[5]	Node	Value of	String		
6	Pabellón	Data[6]	Node	Value of	String		
7	Fundado	Data[7]	Node	Value of	String		
8	Patrocinio	Data[8]	Node	Value of	String		

Vista preliminar de los datos cargados:

 Examine preview data

Rows of step: Get data from XML (NBA Teams) (31 rows)

#	Conferencia	Division	Equipo	Ciudad	Estado	Pabellón	Fundado	Patrocinio
1	Conferencia	División	Equipo	Ciudad	Estado	Pabellón	Fundado	Patrocinio
2	Conferencia Oeste	Noroeste	Denver Nuggets	Denver	CO	Pepsi Center	1967	Western Union
3	Conferencia Oeste	Noroeste	Minnesota Timberwolves	Minneapolis	MN	Target Center	1989	Fibit
4	Conferencia Oeste	Noroeste	Oklahoma City Thunder	Oklahoma City	OK	Chesapeake Energy Arena	1987	Love's
5	Conferencia Oeste	Noroeste	Portland Trail Blazers	Portland	OR	Moda Center	1970	Performance Health
6	Conferencia Oeste	Noroeste	Utah Jazz	Salt Lake City	UT	Vivint Smart Home Arena	1974	Qualtrics
7	Conferencia Oeste	Suroeste	Dallas Mavericks	Dallas	TX	American Airlines Center	1980	SMiles
8	Conferencia Oeste	Suroeste	Houston Rockets	Houston	TX	Toyota Center	1967	ROKIT Phones
9	Conferencia Oeste	Suroeste	Memphis Grizzlies	Memphis	TN	FedExForum	1995	FedEx
10	Conferencia Oeste	Suroeste	New Orleans Pelicans	Nueva Orleans	LA	Smoothie King Center	1988	Zatarain's
11	Conferencia Oeste	Suroeste	San Antonio Spurs	San Antonio	TX	AT&T Center	1967	Frost Bank
12	Conferencia Oeste	Pacífico	Golden State Warriors	San Francisco	CA	Chase Center	1946	Rakuten
13	Conferencia Oeste	Pacífico	Los Angeles Clippers	Los Ángeles	CA	Staples Center	1970	Bumble
14	Conferencia Oeste	Pacífico	Los Angeles Lakers	Los Ángeles	CA	Staples Center	1946	Wish
15	Conferencia Oeste	Pacífico	Phoenix Suns	Phoenix	AZ	Talking Stick Resort Arena	1968	PayPal
16	Conferencia Oeste	Pacífico	Sacramento Kings	Sacramento	CA	Golden 1 Center	1945	Blue Diamond Growers
17	Conferencia Este	Atlántico	Boston Celtics	Boston	MA	TD Garden	1946	General Electric
18	Conferencia Este	Atlántico	Brooklyn Nets	Nueva York	NY	Barclays Center	1967	Infor
19	Conferencia Este	Atlántico	New York Knicks	Nueva York	NY	Madison Square Garden	1946	Squarespace
20	Conferencia Este	Atlántico	Philadelphia 76ers	Filadelfia	PA	Wells Fargo Center	1939	StubHub
21	Conferencia Este	Atlántico	Toronto Raptors	Toronto	ON, Canadá	Scotiabank Arena	1995	Sun Life
22	Conferencia Este	Central	Chicago Bulls	Chicago	IL	United Center	1966	Zenni Optical
23	Conferencia Este	Central	Cleveland Cavaliers	Cleveland	OH	Rocket Mortgage FieldHouse	1970	Goodyear
24	Conferencia Este	Central	Detroit Pistons	Detroit	MI	Little Caesars Arena	1941	Flagstar Bank
25	Conferencia Este	Central	Indiana Pacers	Indianapolis	IN	Bankers Life Fieldhouse	1967	Motorola

Para el fichero WNBA_Teams.xls:

Este archivo excel venía sin formato

A	B	C	D	E	F	G	H	I	J
1	Conferencia Este								
2									
3		Equipo	Pts	J	G	P	p.	c.	dif Ciudad, Estado
4	1	Indiana Fever	6	3	3	0	216	189	27 Indianapolis, Indiana
5	2	Connecticut Sun	5	3	2	1	265	216	49 Uncasville, Connecticut
6	3	Atlanta Dream	5	3	2	1	238	221	17 Atlanta, Georgia
7	4	Washington Mystics	3	2	1	1	154	150	4 Washington D.C., Washington
8	5	New York Liberty	3	3	0	3	232	290	-58 White Plains, New York
9	6	Chicago Sky	2	2	0	2	123	145	-22 Chicago, Illinois
10									
11									
12	Conferencia Oeste								
13		Equipo	Pts	J	G	P	p.	c.	dif Ciudad, Estado
14	1	Minnesota Lynx	4	2	2	0	165	154	11 Minneapolis, Minnesota
15	2	Phoenix Mercury	4	2	2	0	169	159	10 Phoenix, Arizona
16	3	Dallas Wings	4	3	1	2	196	220	-24 Arlington, Texas
17	4	Los Angeles Sparks	3	2	1	1	167	167	0 Los Angeles, California
18	5	Seattle Storm	2	2	0	2	169	179	-10 Seattle, Washington
19	6	Las Vegas Aces	1	1	0	1	75	79	-4 Paradises, Nevada
20									
21									

Así que para poder leerlo automáticamente en Spoon, lo que se hizo manualmente es modificar la tabla de la siguiente manera, solo para darle formato, el resto se dejó como estaba:

A	B	C	D	E	F	G	H	I	J	K
1	Conferencia	Id	Equipo	Pts	J	G	p.	c.	dif	Ciudad, Estado
2	Conferencia Este	1	Indiana Fever	6	3	3	0	216	189	27 Indianapolis, Indiana
3	Conferencia Este	2	Connecticut Sun	5	3	2	1	265	216	49 Uncasville, Connecticut
4	Conferencia Este	3	Atlanta Dream	5	3	2	1	238	221	17 Atlanta, Georgia
5	Conferencia Este	4	Washington Mystics	3	2	1	1	154	150	4 Washington D.C., Washington
6	Conferencia Este	5	New York Liberty	3	3	0	3	232	290	-58 White Plains, New York
7	Conferencia Este	6	Chicago Sky	2	2	0	2	123	145	-22 Chicago, Illinois
8	Conferencia Oeste	1	Minnesota Lynx	4	2	2	0	165	154	11 Minneapolis, Minnesota
9	Conferencia Oeste	2	Phoenix Mercury	4	2	2	0	169	159	10 Phoenix, Arizona
10	Conferencia Oeste	3	Dallas Wings	4	3	1	2	196	220	-24 Arlington, Texas
11	Conferencia Oeste	4	Los Angeles Sparks	3	2	1	1	167	167	0 Los Angeles, California
12	Conferencia Oeste	5	Seattle Storm	2	2	0	2	169	179	-10 Seattle, Washington
13	Conferencia Oeste	6	Las Vegas Aces	1	1	0	1	75	79	-4 Paradises, Nevada

Microsoft Excel input

Step name Fichero Entrada EXCEL (WNBA Teams)

Files Sheets Content Error Handling Fields Additional output fields

Spread sheet type (engine) Excel 97-2003 XLS (JXL)

File or directory Add Browse...

Regular Expression

Exclude Regular Expression

Password

Selected files:

#	File/Directory	Wildcard (RegEx)	Exclude wildcard	Required
1	\$DIR_ENT\WNBA_Teams.xls		N	

Accept filenames from previous steps

Accept filenames from previous step

Microsoft Excel input

Step name Fichero Entrada EXCEL (WNBA Teams)

Files Sheets Content Error Handling Fields Additional output fields

List of sheets to read

#	Sheet name	Start row	Start column
1	Teams	0	0

Microsoft Excel input

Step name Fichero Entrada EXCEL (WNBA Teams)

Files Sheets Content Error Handling Fields Additional output fields

#	Name	Type	Length	Precision	Trim type	Repeat	Format
1	Conferencia	String			both	N	
2	Id	Number			both	N	
3	Equipo	String			both	N	
4	Pts	Number			both	N	
5	J	Number			both	N	
6	G	Number			both	N	
7	P	Number			both	N	
8	p.	Number			both	N	
9	c.	Number			both	N	
10	dif	Number			both	N	
11	Ciudad, Estado	String			both	N	

Vista preliminar de los datos cargados:

Examine preview data

Rows of step: Fichero Entrada EXCEL (WNBA Teams) (12 rows)

#	Conferencia	Id	Equipo	Pts	J	G	P	p.	c.	dif	Ciudad, Estado
1	Conferencia Este	1.0	Indiana Fever	6.0	3.0	3.0	0.0	216.0	189.0	27.0	Indianapolis, Indiana
2	Conferencia Este	2.0	Connecticut Sun	5.0	3.0	2.0	1.0	265.0	216.0	49.0	Uncasville, Connecticut
3	Conferencia Este	3.0	Atlanta Dream	5.0	3.0	2.0	1.0	238.0	221.0	17.0	Atlanta, Georgia
4	Conferencia Este	4.0	Washington Mystics	3.0	2.0	1.0	1.0	154.0	150.0	4.0	Washington D.C., Washington
5	Conferencia Este	5.0	New York Liberty	3.0	3.0	0.0	3.0	232.0	290.0	-58.0	White Plains, New York
6	Conferencia Este	6.0	Chicago Sky	2.0	2.0	0.0	2.0	123.0	145.0	-22.0	Chicago, Illinois
7	Conferencia Oeste	1.0	Minnesota Lynx	4.0	2.0	2.0	0.0	165.0	154.0	11.0	Minneapolis, Minnesota
8	Conferencia Oeste	2.0	Phoenix Mercury	4.0	2.0	2.0	0.0	169.0	159.0	10.0	Phoenix, Arizona
9	Conferencia Oeste	3.0	Dallas Wings	4.0	3.0	1.0	2.0	196.0	220.0	-24.0	Arlington, Texas
10	Conferencia Oeste	4.0	Los Angeles Sparks	3.0	2.0	1.0	1.0	167.0	167.0	0.0	Los Angeles, California
11	Conferencia Oeste	5.0	Seattle Storm	2.0	2.0	0.0	2.0	169.0	179.0	-10.0	Seattle, Washington
12	Conferencia Oeste	6.0	Las Vegas Aces	1.0	1.0	0.0	1.0	75.0	79.0	-4.0	Paradise, Nevada

Procesos (1):

Para el fichero TeamCodes.txt (2):

- A los campos de tipo string, como aplicación de buenas prácticas, los convertiremos a mayúsculas y quitaremos espacios en blanco del principio y del final. **STRING OPERATIONS**
- Ordenamos los datos por el campo Code. **SORT ROWS**

The screenshot shows the Pentaho Data Integration interface with two steps visible:

- String operations**: This step is configured to process three fields: League, Team, and Code. The configuration includes setting both trim type to upper and padding to none for all three fields. The step name is "String operations 3".
- Sort rows**: This step is configured to sort by the "Code" field in ascending order. The step name is "Ordenar por código". It also specifies a temporary file prefix of "out", a sort size of 1,000,000 rows, and a free memory threshold of 10%. The "Fields" section shows the mapping of the "Code" field from the previous step to the current step.

Para el fichero NBA_Teams.xml (3):

- Quitamos el primer registro que contiene las caeceras. **FILTER ROWS**
- Cambiar el tipo de dato del campo Fundado a Integer. **SELECT VALUES**
- A los campos de tipo string, como aplicación de buenas prácticas, los convertiremos a mayúsculas y quitaremos espacios en blanco del principio y del final. **STRING OPERATIONS**

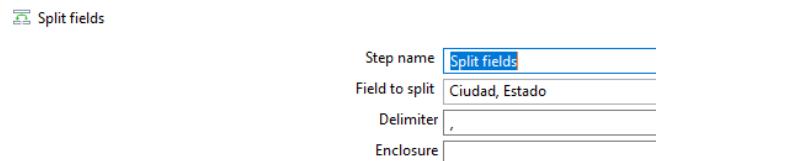
The screenshot shows the Talend Data Integration interface with three main steps:

- Filter rows**: Step name is "Filter rows". It has two dropdowns: "Send 'true' data to step" set to "Select values" and "Send 'false' data to step" set to nothing. The condition is defined as NOT Fundado = Fundado (String).
- Select values**: Step name is "Select values". It shows a table with one row: Fieldname "Fundado" is renamed to "fund" and its type is Integer.
- String operations**: Step name is "String operations 2.2". It shows a table with seven rows, each defining string processing rules for fields like Conferencia, Division, Equipo, Ciudad, Estado, Pabellon, and Patrocinio.

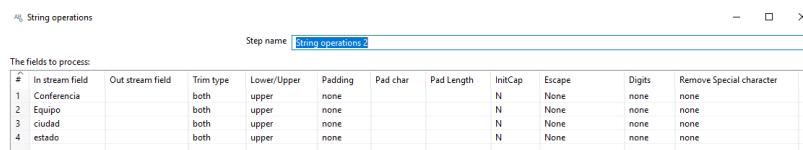
Para el fichero WNBA_Teams.xls (3):

- Borrar campos de la tabla Pts, J, G, P, p., c., dif. (Se prefiere cargar todo y después borrar por si en un futuro se usan esos datos. Solo habría que borrar ese paso de la transformación y no cambiar la carga de datos). **SELECT VALUES**
- Separar el campo Ciudad,Estado que ahora está junto en dos, uno que contenga la ciudad y otro el estado. **SPLIT FIELDS**
- A los campos de tipo string, como aplicación de buenas prácticas, los convertiremos a mayúsculas y quitaremos espacios en blanco del principio y del final. **STRING OPERATIONS**

The screenshot shows the Talend Data Integration interface with a step named "Borrar Campos" (Delete Fields) under the "Select & Alter" tab. It shows a table with seven rows, each listing a field to remove: Pts, J, G, P, p., c., and dif.



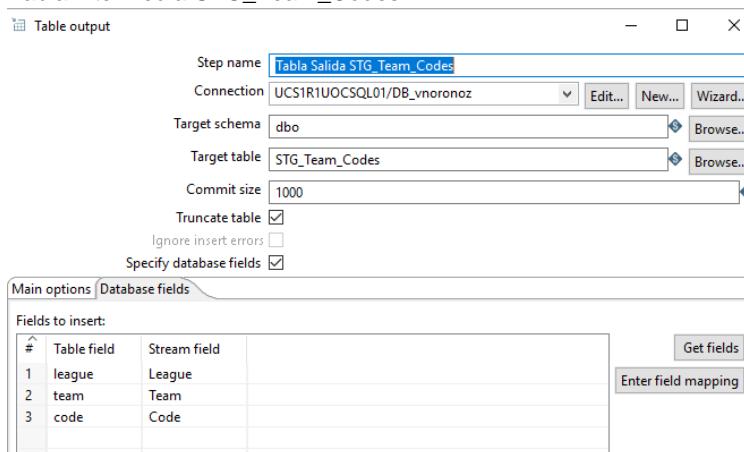
The screenshot shows the configuration for the 'Split fields' step. The 'Step name' is 'Split fields'. The 'Field to split' is 'Ciudad, Estado'. The 'Delimiter' is ',' and the 'Enclosure' is empty. Below this, a table titled 'Fields' lists two new fields: 'ciudad' and 'estado', both of type String.



The screenshot shows the configuration for the 'String operations' step. The 'Step name' is 'String operations'. The 'Fields to process' table includes four rows: 'Conferencia', 'Equipo', 'ciudad', and 'estado'. The 'Trim type' for all is 'both', 'Lower/Upper' is 'upper', and 'Padding' is 'none'. The 'Pad char' is empty, 'Pad Length' is 0, 'InitCap' is 'N', 'Escape' is 'None', 'Digits' is 'none', and 'Remove Special character' is 'none'.

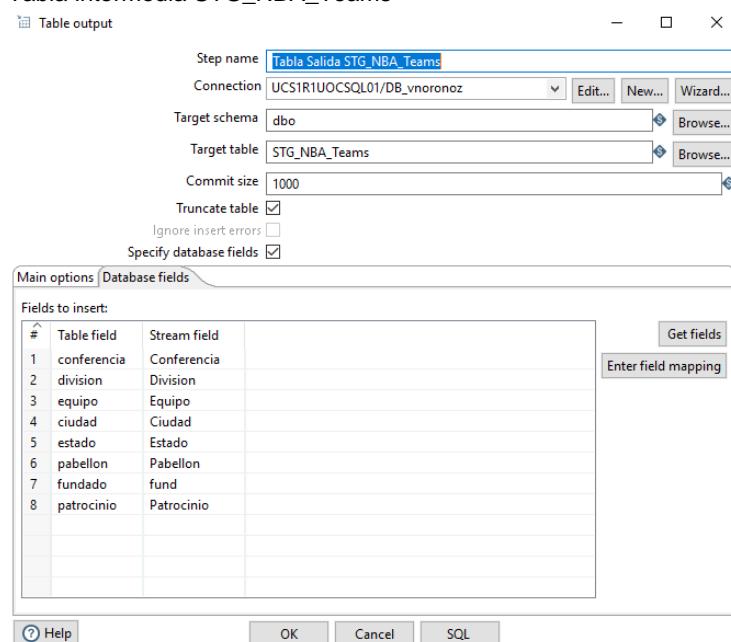
Outputs (3):

- Tabla intermedia STG_Team_Codes



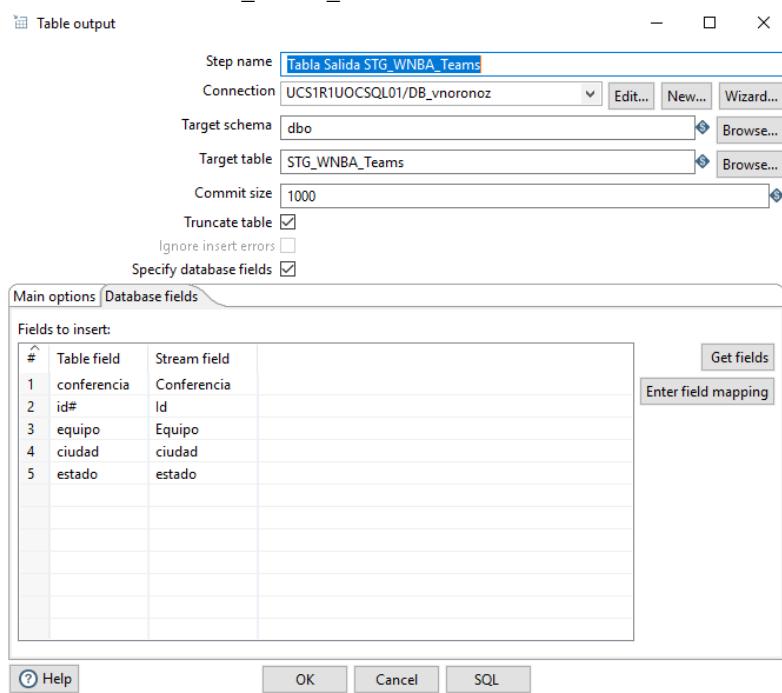
The screenshot shows the configuration for the 'Tabla Salida STG_Team_Codes' table output. The 'Step name' is 'Tabla Salida STG_Team_Codes'. The 'Connection' is 'UCS1R1UOCSQL01/DB_vnoronoz'. The 'Target schema' is 'dbo' and the 'Target table' is 'STG_Team_Codes'. The 'Commit size' is 1000. The 'Truncate table' checkbox is checked. The 'Specify database fields' checkbox is checked. The 'Main options' tab is selected, showing a table titled 'Fields to insert' with three entries: 'league' (Table field) mapped to 'League' (Stream field), 'team' mapped to 'Team', and 'code' mapped to 'Code'.

- Tabla intermedia STG_NBA_Teams

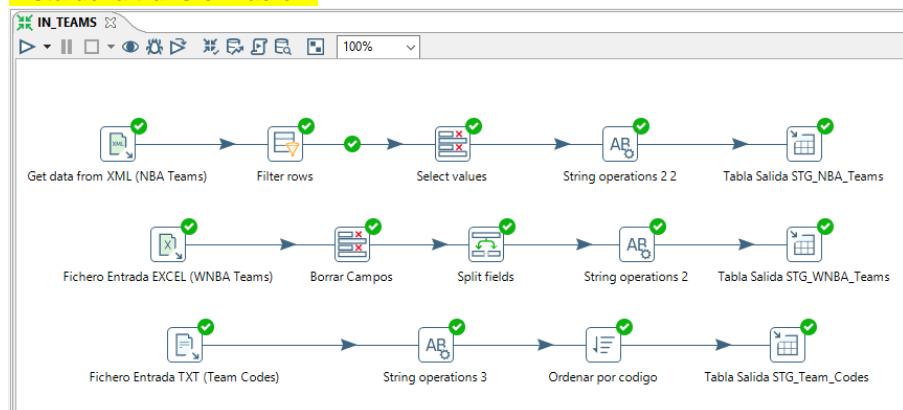


The screenshot shows the configuration for the 'Tabla Salida STG_NBA_Teams' table output. The 'Step name' is 'Tabla Salida STG_NBA_Teams'. The 'Connection' is 'UCS1R1UOCSQL01/DB_vnoronoz'. The 'Target schema' is 'dbo' and the 'Target table' is 'STG_NBA_Teams'. The 'Commit size' is 1000. The 'Truncate table' checkbox is checked. The 'Specify database fields' checkbox is checked. The 'Main options' tab is selected, showing a table titled 'Fields to insert' with eight entries: 'conferencia' (Table field) mapped to 'Conferencia' (Stream field), 'division' mapped to 'Division', 'equipo' mapped to 'Equipo', 'ciudad' mapped to 'Ciudad', 'estado' mapped to 'Estado', 'pabellon' mapped to 'Pabellon', 'fundado' mapped to 'fund', and 'patrocinio' mapped to 'Patrocinio'.

– Tabla intermedia STG_WNBA_Teams



Vista de la transformación:



Resultado de la ejecución:

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	\$
1	Fichero Entrada TXT (Team Codes)	0	0	42	43	0	1	0	0	Finished	0.0s	
2	Fichero Entrada EXCEL (WNBA Teams)	0	0	12	12	0	0	0	0	Finished	1.7s	
3	String operations 3	0	42	42	0	0	0	0	0	Finished	1.7s	
4	Borrar Campos	0	12	12	0	0	0	0	0	Finished	1.8s	
5	Get data from XML (NBA Teams)	0	0	31	31	0	0	0	0	Finished	0.1s	
6	Split fields	0	12	12	0	0	0	0	0	Finished	1.8s	
7	Filter rows	0	31	30	0	0	0	0	0	Finished	1.7s	
8	Ordenar por codigo	0	42	42	0	0	0	0	0	Finished	1.7s	
9	Select values	0	30	30	0	0	0	0	0	Finished	1.8s	
10	Tabla Salida STG_Team_Codes	0	42	42	0	42	0	0	0	Finished	1.8s	
11	String operations 2 2	0	30	30	0	0	0	0	0	Finished	1.8s	
12	String operations 2	0	12	12	0	0	0	0	0	Finished	1.8s	
13	Tabla Salida STG_WNBA_Teams	0	30	30	0	30	0	0	0	Finished	1.8s	
14	Tabla Salida STG_WNBA_Teams	0	12	12	0	12	0	0	0	Finished	1.8s	

Bloque TR

El bloque TR contiene los procesos ETL para la carga de datos al modelo multidimensional del almacén que hemos diseñado, compuesto por dimensiones y tablas de hechos, desde las tablas intermedias pobladas con los procesos del bloque IN.

Este bloque se divide a su vez en dos sub-bloques; por un lado, los procesos para la carga de dimensiones y por otro, los procesos para la carga de tablas de hechos. Esta división permite el reprocesamiento de dichos procesos en caso de error y un mejor entendimiento de la implementación de los procesos.

Carga y Transformaciones - Bloque TR_DIM

Este bloque, contiene las transformaciones para la carga inicial de las dimensiones al almacén desde las tablas intermedias STG_ del Staging Area.

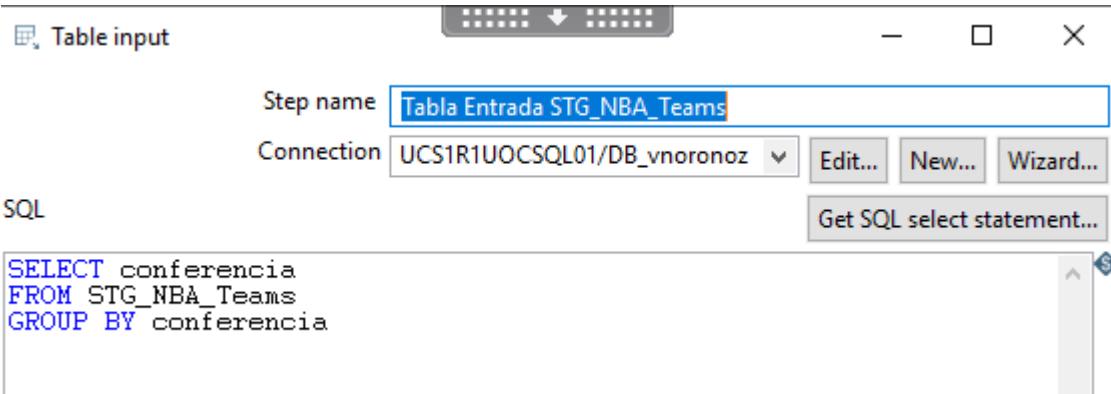
Además de la carga inicial de las dimensiones del modelo multidimensional, se tendrá en cuenta que se puedan ejecutar las transformaciones las veces que sean necesarias, por esto se incluirá como primer paso de las transformaciones del bloque TR_DIM un borrado de las tablas de dimensiones. Consistirá en la ejecución de la sentencia DDL “delete from tabla_dimension”, donde tabla dimensión es la tabla de dimensión del modelo. Este paso será común en todas las transformaciones y no se entrará en más detalle. Otro paso común en la mayoría de las transformaciones será añadir un registro “DESCONOCIDO” con pk 9999.

El otro paso común en todas las transformaciones será agregar desde PDI los valores de las claves primarias mediante Add Sequence.

1) Transformación TR_DIM_CONF

Inputs (1):

- Atributo Conferencia de STG_NBA_Teams



Procesos (5):

- Borrar dimensión. **EXECUTE SQL SCRIPT**
- Asignar códigos de Conferencia. **VALUE MAPPER**
- Añadir secuencia automática para la clave primaria. **ADD SEQUENCE**
- Generar registro para valor DESCONOCIDO y pk 9999. **GENERATE ROWS**
- Ordenar tabla por pk. **SORT ROWS**

Execute SQL script

Step name: **Borrar_DIM**
 Connection: UCS1R1UOCSQL01/DB_vnoronoz

SQL script to execute. (statements separated by ;) Question marks will be replaced by arguments.

```
DELETE FROM dbo.DIM_Conferencias
```

Line 1 Column 0

Value mapper

Step name: **Asignar codigos**
 Fieldname to use: **conferencia**
 Target field name: **codigo_conferencia**
 Default upon:

Field values:

#	Source value	Target value
1	CONFERENCIA ESTE	E
2	CONFERENCIA OESTE	O

Add sequence

Step name: **Add sequence**
 Name of value: **pk_conferencia**

Use a database to generate the sequence

Use DB to get sequence?
 Connection: UCS1R1UOCSQL01/DB_vnoronoz
 Schema name:
 Sequence name: **SEQ**

Use a transformation counter to generate the sequence

Use counter to calculate sequence?
 Counter name (optional):
 Start at value: **1**
 Increment by: **1**
 Maximum value: **999999999**

Generate rows

Step name: **Generate rows**
 Limit: **1**
 Never stop generating rows
 Interval in ms (delay): **5000**
 Current row/time field name: **now**
 Previous row/time field name: **FiveSecondsAgo**

Fields:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value
1	conferencia	String							DESCONOCIDO
2	codigo_conferencia	String							DESC
3	pk_conferencia	Integer							9999

Outputs (1):

- Tabla DIM_Conferencias

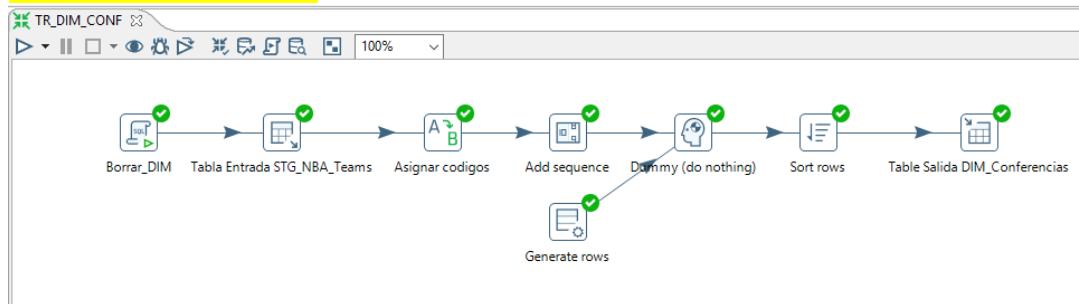
Table output

Step name	Table Salida DIM_Conferencias
Connection	UCS1R1UOCSQL01/DB_vnoronoz
Target schema	dbo
Target table	DIM_Conferencias
Commit size	1000
Truncate table	<input type="checkbox"/>
Ignore insert errors	<input type="checkbox"/>
Specify database fields	<input checked="" type="checkbox"/>

Main options Database fields

Fields to insert:

#	Table field	Stream field
1	conferencia	conferencia
2	codigo_conferencia	codigo_conferencia
3	pk_conferencia	pk_conferencia

Vista de la transformación:**Resultado de la ejecución:**

Execution Results

Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)
1 Borrar_DIM	0	0	1	0	0	0	0	0	Finished	0.0s	62
2 Tabla Entrada STG_NBA_Teams	0	0	2	2	0	0	0	0	Finished	0.1s	26
3 Asignar codigos	0	2	2	0	0	0	0	0	Finished	0.1s	26
4 Add sequence	0	2	2	0	0	0	0	0	Finished	0.1s	18
5 Generate rows	0	0	1	0	0	0	0	0	Finished	0.0s	32
6 Dummy (do nothing)	0	3	3	0	0	0	0	0	Finished	0.1s	24
7 Sort rows	0	3	3	0	0	0	0	0	Finished	0.2s	19
8 Table Salida DIM_Conferencias	0	3	3	0	3	0	0	0	Finished	0.2s	14

2) Transformación TR_DIM_DIV**Inputs (1):**

- Atributo Division de STG_NBA_Teams

Table input

Step name: Tabla Entrada STG_NBA_Teams

Connection: UCS1R1UOCSQL01/DB_vnoronoz

SQL:

```
SELECT division
FROM STG_NBA_Teams
GROUP BY division
```

Procesos (7):

- Borrar dimensión. **EXECUTE SQL SCRIPT**
- Asignar códigos de División. **VALUE MAPPER**
- Obtener valor Conferencia y su id (clave foránea). **DATABASE LOOKUP**
- Control si la clave foránea es null, darle valor 9999. **IF FIELD VALUE IS NULL**
- Añadir secuencia automática para la clave primaria. **ADD SEQUENCE**
- Generar registro para valor DESCONOCIDO y pk 9999. **GENERATE ROWS**
- Ordenar tabla por pk. **SORT ROWS**

Value mapper

Step name: Asignar códigos

Fieldname to use: division

Target field name (empty=overwrite): codigo_division

Default upon non-matching:

#	Source value	Target value
1	NOROESTE	NE
2	SUROESTE	SO
3	SURESTE	SE
4	PACÍFICO	PA
5	CENTRAL	CE
6	ATLÁNTICO	AT

OK Cancel

Database lookup

Step name: Obtener Conferencia

Connection: UCS1R1UOCSQL01/DB_vnoronoz

Lookup schema: dbo

Lookup table: STG_NBA_Teams

Enable cache?

Cache size in rows (0=cache): 0

Load all data from table

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	division	=	division	

Values to return from the lookup table:

#	Field	New name	Default	Type
1	conferencia			None

If field value is null

Step name: If field value is null

Replace Null for all fields

Replace by value:

Set empty string?

Mask (Date):

Select fields

Select value type

Value types

#	Type	Replace by value	Conversion
1			

Fields

#	Field	Replace by value	Conversion n
1	id_conferencia	9999	

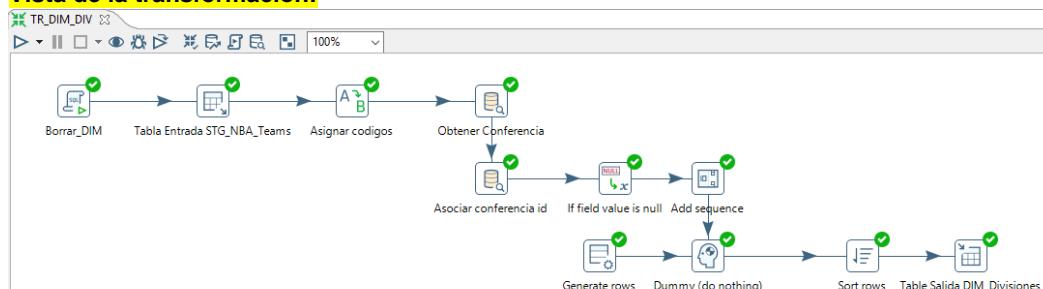
Outputs (1):

- Tabla DIM_Divisiones

Table output

Step name: Table Salida DIM_Divisiones
 Connection: UCS1R1UOCSQL01/DB_vnoronoz
 Target schema: dbo
 Target table: DIM_Divisiones
 Commit size: 1000
 Truncate table:
 Ignore insert errors:
 Specify database fields:

Fields to insert:		
#	Table field	Stream field
1	division	division
2	codigo_division	codigo_division
3	id_conferencia	id_conferencia
4	pk_division	pk_division

Vista de la transformación:**Resultado de la ejecución:**

Execution Results

Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)
1 Borrar_DIM	0	0	1	0	0	0	0	0	Finished	0.0s	62
2 Tabla Entrada STG_NBA_Teams	0	0	6	6	0	0	0	0	Finished	0.0s	375
3 Asignar codigos	0	6	6	0	0	0	0	0	Finished	0.0s	140
4 Obtener Conferencia	0	6	6	6	0	0	0	0	Finished	0.1s	80
5 Asociar conferencia id	0	6	6	6	0	0	0	0	Finished	0.1s	69
6 Generate rows	0	0	1	0	0	0	0	0	Finished	0.0s	62
7 If field value is null	0	6	6	0	0	0	0	0	Finished	0.1s	51
8 Add sequence	0	6	6	0	0	0	0	0	Finished	0.1s	40
9 Dummy (do nothing)	0	7	7	0	0	0	0	0	Finished	0.2s	39
10 Sort rows	0	7	7	0	0	0	0	0	Finished	0.2s	33
11 Table Salida DIM_Divisiones	0	7	7	0	7	0	0	0	Finished	0.3s	26

3) Transformación TR_DIM_EEUU

Inputs (1):

- Tabla STG_EEUU

Table input

Step name: Tabla Entrada STG_EEUU

Connection: UCS1R1UOCSQL01/DB_vnoronoz

SQL:

```
SELECT *
FROM STG_EEUU
```

Procesos (4):

- Borrar dimensión. **EXECUTE SQL SCRIPT**
- Añadir secuencia automática para la clave primaria. **ADD SEQUENCE**
- Generar registro para valor DESCONOCIDO y pk 9999. **GENERATE ROWS**
- Ordenar tabla por pk. **SORT ROWS**

Outputs (1):

- Tabla DIM_Estados_EEUU

Table output

Step name: Table Salida DIM_Estados_EEUU

Connection: UCS1R1UOCSQL01/DB_vnoronoz

Target schema: dbo

Target table: DIM_Estados_EEUU

Commit size: 1000

Truncate table:

Ignore insert errors:

Specify database fields:

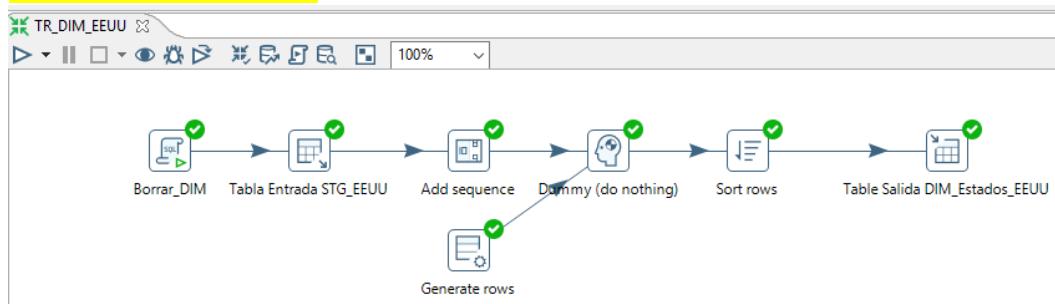
Main options tab:

#	Table field	Stream field
1	nombre	estado
2	nombre_oficial	nombre_oficial
3	superficie	superficie
4	codigo	abreviatura
5	poblacion	poblacion_2010
6	densidadPoblacion	densidad_pobl
7	capital	capital
8	pk_estado_EEU	pk_estado_EEU

Database fields tab:

#	Table field	Stream field
1	nombre	estado
2	nombre_oficial	nombre_oficial
3	superficie	superficie
4	codigo	abreviatura
5	poblacion	poblacion_2010
6	densidadPoblacion	densidad_pobl
7	capital	capital
8	pk_estado_EEU	pk_estado_EEU

Vista de la transformación:



Resultado de la ejecución:

The screenshot shows the 'Execution Results' view in Talend Studio. The table displays various steps and their performance metrics. The columns include Stepname, Copynr, Read, Written, Input, Output, Updated, Rejected, Errors, Active, Time, and Speed (r/s). The steps listed are: 1. Borrar_DIM, 2. Tabla Entrada STG_EEUU, 3. Generate rows, 4. Add sequence, 5. Dummy (do nothing), 6. Sort rows, and 7. Table Salida DIM_Estados_EEUU.

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)
1	Borrar_DIM	0	0	1	0	0	0	0	0	Finished	0.0s	91
2	Tabla Entrada STG_EEUU	0	0	50	50	0	0	0	0	Finished	0.0s	1,786
3	Generate rows	0	0	1	0	0	0	0	0	Finished	0.0s	91
4	Add sequence	0	50	50	0	0	0	0	0	Finished	0.1s	1,000
5	Dummy (do nothing)	0	51	51	0	0	0	0	0	Finished	0.1s	895
6	Sort rows	0	51	51	0	0	0	0	0	Finished	0.1s	567
7	Table Salida DIM_Estados_EEUU	0	51	51	0	51	0	0	0	Finished	0.2s	256

4) Transformación TR_DIM_TEAMS

Inputs (1):

- Tabla STG_Team_Codes

Table input

Step name: Tabla Entrada STG_Team_Codes

Connection: UCS1R1UOCSQL01/DB_vnoronoz

SQL:

```
SELECT *
FROM STG_Team_Codes
```

Procesos (12):

- Borrar dimensión. **EXECUTE SQL SCRIPT**
- Asociar datos con equipos NBA. **DATABASE LOOKUP**
- Asociar datos con equipos WNBA. **DATABASE LOOKUP**
- Unir datos leídos de equipos. **CONCAT FIELDS**
- Asociar datos con estados NBA. **DATABASE LOOKUP**
- Asociar datos con estados WNBA. **DATABASE LOOKUP**
- Unir datos leídos de estados. **CONCAT FIELDS**
- Asociar datos división. **DATABASE LOOKUP**
- Control si la clave foránea es null, darle valor 9999. **IF FIELD VALUE IS NULL**
- Añadir secuencia automática para la clave primaria. **ADD SEQUENCE**
- Generar registro para valor DESCONOCIDO y pk 9999. **GENERATE ROWS**
- Ordenar tabla por pk. **SORT ROWS**

Database lookup

Step name **Datos NBA**

Connection **UCS1R1UOCSQL01/DB_vnoronoz**

Lookup schema **dbo**

Lookup table **STG_NBA_Teams**

Enable cache?

Cache size in rows (0=cache) **0**

Load all data from table

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	equipo	=	team	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	division	division_nba		None
2	ciudad	ciudad_nba		None
3	estado	cod_estado_nba		None
4	pabellon	pabellon_nba		None
5	fundado	fundado_nba		None
6	patrocinio	patrocinio_nba		None

Database lookup

Step name **Datos WNBA**

Connection **UCS1R1UOCSQL01/DB_vnoronoz**

Lookup schema **dbo**

Lookup table **STG_WNBA_Teams**

Enable cache?

Cache size in rows (0=cache) **0**

Load all data from table

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	equipo	=	team	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	ciudad	ciudad_wnba		None
2	estado	estado_wnba		None

Database lookup

Step name **Datos estados 1**

Connection **UCS1R1UOCSQL01/DB_vnoronoz**

Lookup schema **dbo**

Lookup table **DIM_Estados_EEUU**

Enable cache?

Cache size in rows (0=cache) **0**

Load all data from table

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	nombre	=	estado_wnba	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	pk_estado_EEU	id_estado		None

Database lookup

Step name **Datos estados 2**

Connection **UCS1R1UOCSQL01/DB_vnoronoz**

Lookup schema **dbo**

Lookup table **DIM_Estados_EEUU**

Enable cache?

Cache size in rows (0=cache) **0**

Load all data from table

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	codigo	=	cod_estado_nba	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	pk_estado_EEU	id_estado		None

Database lookup

Step name **Datos division**

Connection **UCS1R1UOCSQL01/DB_vnoronoz**

Lookup schema **dbo**

Lookup table **DIM_Divisiones**

Enable cache?

Cache size in rows (0=cache) **0**

Load all data from table

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	division	=	division_nba	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	pk_division	id_division		None

Outputs (1):

- Tabla DIM_Equipos

Table output

Step name **Table Salida DIM_Equipos**

Connection **UCS1R1UOCSQL01/DB_vnoronoz**

Target schema **dbo**

Target table **DIM_Equipos**

Commit size **1000**

Truncate table

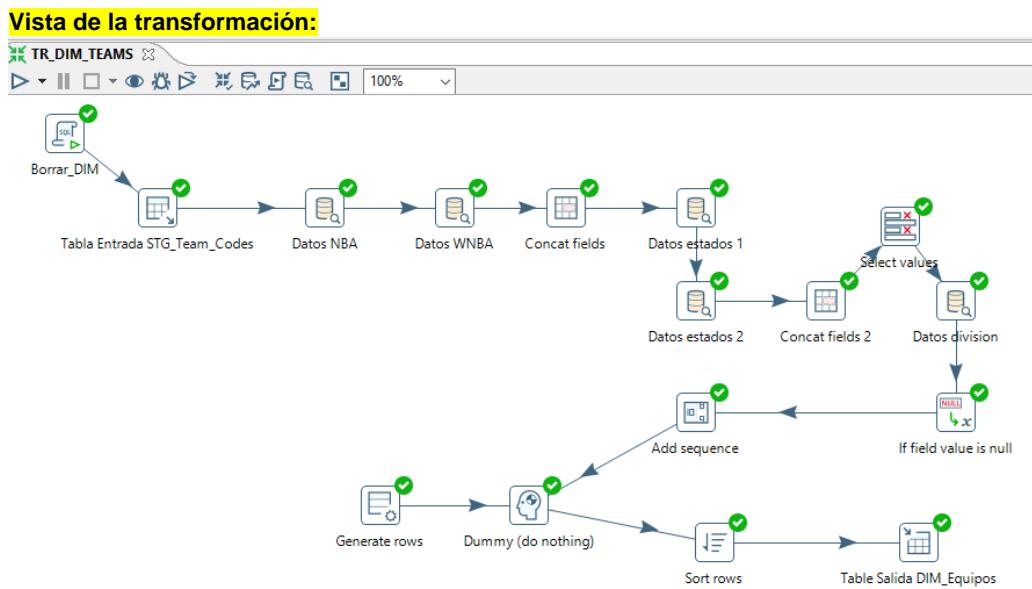
Ignore insert errors

Specify database fields

Main options Database fields

Fields to insert:

#	Table field	Stream field
1	liga	league
2	nombre_equipo	team
3	codigo_equipo	code
4	pabellon	pabellon_nba
5	fundado	fundado_nba
6	patrocinio	patrocinio_nba
7	ciudad	ciudad_ok
8	id_estado_EEU	id_estado_ok
9	id_division	id_division
10	pk_equipo	pk_equipo



Resultado de la ejecución:

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)
1	Borrar_DIM	0	0	1	0	0	0	0	0	Finished	0.0s	62
2	Tabla Entrada STG_Team_Codes	0	0	42	42	0	0	0	0	Finished	0.0s	1,050
3	Datos NBA	0	42	42	30	0	0	0	0	Finished	0.1s	438
4	Datos WNBA	0	42	42	12	0	0	0	0	Finished	0.3s	128
5	Concat fields	0	42	42	0	0	0	0	0	Finished	0.3s	127
6	Datos estados 1	0	42	42	12	0	0	0	0	Finished	0.3s	126
7	Datos estados 2	0	42	42	29	0	0	0	0	Finished	0.4s	115
8	Generate rows	0	0	1	0	0	0	0	0	Finished	0.0s	48
9	Concat fields 2	0	42	42	0	0	0	0	0	Finished	0.4s	110
10	Select values	0	42	42	0	0	0	0	0	Finished	0.4s	106
11	Datos division	0	42	42	30	0	0	0	0	Finished	0.4s	98
12	If field value is null	0	42	42	0	0	0	0	0	Finished	0.4s	95
13	Add sequence	0	42	42	0	0	0	0	0	Finished	0.5s	92
14	Dummy (do nothing)	0	43	43	0	0	0	0	0	Finished	0.5s	91
15	Sort rows	0	43	43	0	0	0	0	0	Finished	0.5s	85
16	Table Salida DIM_Equipos	0	43	43	0	43	0	0	0	Finished	0.6s	76

5) Transformación TR_DIM_POS

Inputs (0):

Procesos (2):

- Borrar dimensión. **EXECUTE SQL SCRIPT**
- Asociar datos de forma manual de las posiciones de juego. **DATA GRID**

Data grid

Step name: Data grid

Meta	Data																														
	<table border="1"> <thead> <tr> <th>#</th> <th>Name</th> <th>Type</th> <th>Format</th> <th>L</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>pk_posicion</td> <td>Integer</td> <td></td> <td></td> </tr> <tr> <td>2</td> <td>codigo_posicion</td> <td>String</td> <td></td> <td></td> </tr> <tr> <td>3</td> <td>desc_posicion_EN</td> <td>String</td> <td></td> <td></td> </tr> <tr> <td>4</td> <td>desc_posicion_ES</td> <td>String</td> <td></td> <td></td> </tr> <tr> <td>5</td> <td>num_posicion</td> <td>Integer</td> <td></td> <td></td> </tr> </tbody> </table>	#	Name	Type	Format	L	1	pk_posicion	Integer			2	codigo_posicion	String			3	desc_posicion_EN	String			4	desc_posicion_ES	String			5	num_posicion	Integer		
#	Name	Type	Format	L																											
1	pk_posicion	Integer																													
2	codigo_posicion	String																													
3	desc_posicion_EN	String																													
4	desc_posicion_ES	String																													
5	num_posicion	Integer																													

Data grid

Step name **Data grid**

Meta	Data				
#	pk_posicion	codigo_posicion	desc_posicion_EN	desc_posicion_ES	num_posicion
1	1	PG	POINT GUARD	BASE	1
2	2	SG	SHOOTING GUARD	ESCOLTA	2
3	3	SF	SMALL FORWARD	ALERO	3
4	4	PF	POWER FORWARD	ALA-PIVOT	4
5	5	C	CENTER	PIVOT	5
6	9999	DESC		DESCONOCIDO	9999

Outputs (1):

- Tabla DIM_PosicionesJuego

Table output

Step name **Table Salida DIM_PosicionesJuego**

Connection **UCS1R1UOCSQL01/DB_vnoronoz**

Target schema **dbo**

Target table **DIM_PosicionesJuego**

Commit size **1000**

Truncate table

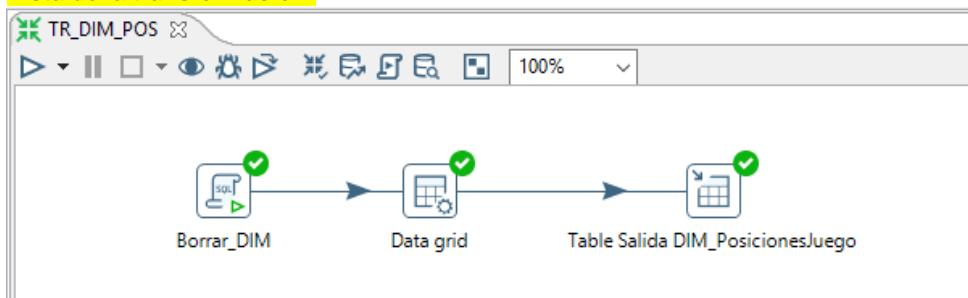
Ignore insert errors

Specify database fields

Main options Database fields

Fields to insert:

#	Table field	Stream field
1	codigo_posicion	codigo_posicion
2	desc_posicion_EN	desc_posicion_EN
3	desc_posicion_ES	desc_posicion_ES
4	num_posicion	num_posicion
5	pk_posicion_juego	pk_posicion

Vista de la transformación:**Resultado de la ejecución:**

Execution Results

Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)
Borrar_DIM	0	0	1	0	0	0	0	0	Finished	0.0s	53
Data grid	0	0	6	0	0	0	0	0	Finished	0.0s	333
Table Salida DIM_PosicionesJuego	0	6	6	0	6	0	0	0	Finished	0.1s	62

6) Transformación TR_DIM_PLAYS

Inputs (1):

- Tabla STG_NBA_free_throws

The screenshot shows the configuration for a 'Table input' step. The 'Step name' is 'Tabla Entrada STG_NBA_free_throws'. The 'Connection' is set to 'UCS1R1UOCSQL01/DB_vnoronoz'. The 'SQL' section contains the query: `SELECT * FROM STG_NBA_free_throws`. There are buttons for 'Edit...', 'New...', and 'Wizard...'.

Procesos (6):

- Borrar dimensión. **EXECUTE SQL SCRIPT**
- Asignar código de jugada made si acertó missed si falló. **VALUE MAPPER**
- Ordenar los registros y quedarnos con valores únicos. **SORT ROWS + UNIQUE ROWS**
- Añadir secuencia automática para la clave primaria. **ADD SEQUENCE**
- Generar registro para valor DESCONOCIDO y pk 9999. **GENERATE ROWS**
- Ordenar tabla por pk. **SORT ROWS**

The screenshot shows the configuration for a 'Value mapper' step. The 'Step name' is 'Value mapper'. The 'Fieldname to use' is 'shot_made' and the 'Target field name' is 'codigo'. The 'Default upon non-matching' field is empty. Below, under 'Field values:', there is a table:

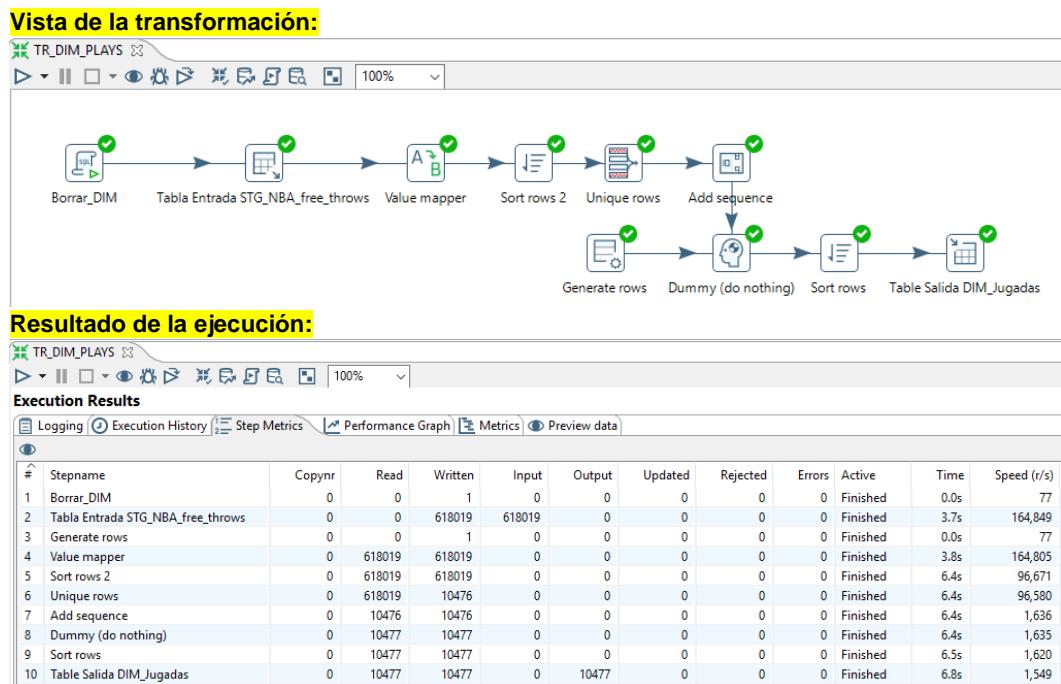
#	Source value	Target value
1	0	MISSED
2	1	MADE

Outputs (1):

- Tabla DIM_Jugadas

The screenshot shows the configuration for a 'Table output' step. The 'Step name' is 'Table Salida DIM_Jugadas'. The 'Connection' is 'UCS1R1UOCSQL01/DB_vnoronoz'. The 'Target schema' is 'dbo' and the 'Target table' is 'DIM_Jugadas'. The 'Commit size' is '1000'. Under 'Main options', 'Specify database fields' is checked. In the 'Database fields' tab, there is a table titled 'Fields to insert':

#	Table field	Stream field
1	dec_jugada	play
2	codigo_jugada	codigo
3	pk_jugada	pk_jugada



7) Transformación TR_DIM_GAMES

Inputs (1):

- Tabla STG_NBA_free_throws

Table input

Step name: Tabla Entrada STG_NBA_free_throws

Connection: UCS1R1UOCSQL01/DB_vnoronoz

SQL:

```
SELECT *
FROM STG_NBA_free_throws
```

Procesos (5):

- Borrar dimensión. **EXECUTE SQL SCRIPT**
- Ordenar los registros y quedarnos con valores únicos de juego, temporada y playoff. **SORT ROWS + UNIQUE ROWS**
- Añadir secuencia automática para la clave primaria. **ADD SEQUENCE**
- Generar registro para valor DESCONOCIDO y pk 9999. **GENERATE ROWS**
- Ordenar tabla por pk. **SORT ROWS**

Sort rows

Step name: Sort rows
 Sort directory: %%java.io.tmpdir%%
 TMP-file prefix: out
 Sort size (rows in memory): 1000000
 Free memory threshold (in %):
 Compress TMP Files?
 Only pass unique rows? (verifies keys only)

Fields :

#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	season	Y	N	N	0	N
2	game	Y	N	N	0	N

Unique rows

Step name: Unique rows
 Settings:
 Add counter to output? Counter field:
 Redirect duplicate row Error description:

Fields to compare on (no entries means: compare complete row)

#	Fieldname	Ignore case
1	game	N
2	season	N
3	playoffs	N

Outputs (1):

- Tabla DIM_Partidos

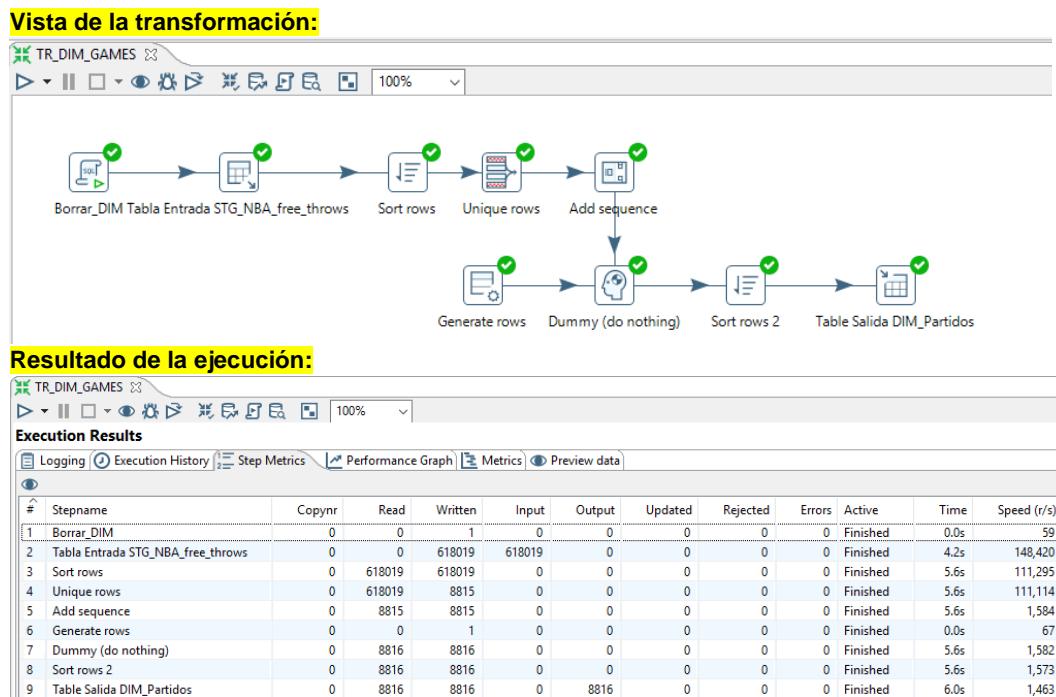
Table output

Step name: Table Salida DIM_Partidos
 Connection: UCS1R1UOCSQL01/DB_vnoronoz
 Target schema: dbo
 Target table: DIM_Partidos
 Commit size: 1000
 Truncate table:
 Ignore insert errors:
 Specify database fields:

Main options Database fields

Fields to insert:

#	Table field	Stream field
1	resultado	end_result
2	partido	game
3	playoffs	playoffs
4	temporada	season
5	pk_partido	pk_partido



8) Transformación TR_DIM_MIN

Inputs (1):

- Atributo time_ de la Tabla STG_NBA_free_throws

Table input

Step name: Tabla Entrada STG_NBA_free_throws

Connection: UCS1R1UOCSQL01/DB_vnoronoz

SQL:

```
SELECT time_
FROM STG_NBA_free_throws
```

Procesos (4):

- Borrar dimensión. **EXECUTE SQL SCRIPT**
- Separar minutos y segundos. **SPLIT FIELDS**
- Añadir secuencia automática para la clave primaria. **ADD SEQUENCE**
- Ordenar tabla por pk. **SORT ROWS**

Split fields

Step name: Separar minutos y segundos

Field to split: time_

Delimiter: :

Enclosure:

#	New field	ID	Remove ID?	Type	Length	Precision	Format	Group
1	minutos	N		Integer				
2	segundos	N		Integer				

OK Cancel

Outputs (1):

- Tabla DIM_Minutos

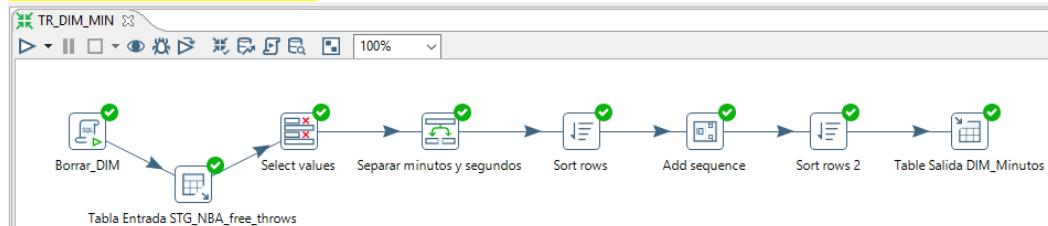
Table output

Step name	Table Salida DIM_Minutos
Connection	UCS1R1UOCSQL01/DB_vnoronoz
Target schema	dbo
Target table	DIM_Minutos
Commit size	1000
Truncate table	<input type="checkbox"/>
Ignore insert errors	<input type="checkbox"/>
Specify database fields	<input checked="" type="checkbox"/>

Main options Database fields

Fields to insert:

#	Table field	Stream field
1	minuto	minutos
2	segundo	segundos
3	minutoSegundo	tiempo
4	pk_minutoSegundo	pk_minutoSegundo

Vista de la transformación:**Resultado de la ejecución:**

Execution Results

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)
1	Borrar_DIM	0	0	1	0	0	0	0	0	Finished	0.0s	167
2	Tabla Entrada STG_NBA_free_throws	0	0	534	534	0	0	0	0	Finished	0.2s	2,282
3	Select values	0	534	534	0	0	0	0	0	Finished	0.2s	2,225
4	Separar minutos y segundos	0	534	534	0	0	0	0	0	Finished	0.2s	2,188
5	Sort rows	0	534	534	0	0	0	0	0	Finished	0.3s	1,928
6	Add sequence	0	534	534	0	0	0	0	0	Finished	0.3s	1,907
7	Sort rows 2	0	534	534	0	0	0	0	0	Finished	0.3s	1,712
8	Table Salida DIM_Minutos	0	534	534	0	534	0	0	0	Finished	0.4s	1,487

9) Transformación TR_DIM_PLAYERS**Inputs (2):**

- Tabla STG_NBA_Seasons_Stats_1950_2017
- Tabla STG_WNBA_Seasons_Stats_2005_2017

Table input

Step name: Tabla Entrada STG_NBA_Seasons_Stats
 Connection: UCS1R1UOCSQL01/DB_vnoronoz

SQL

```
SELECT player
FROM STG_NBA_Seasons_Stats_1950_2017
GROUP BY player
ORDER BY player ASC
```

Table input

Step name: Tabla Entrada STG_WNBA_Stats
 Connection: UCS1R1UOCSQL01/DB_vnoronoz

SQL

```
SELECT id,data,stat
FROM STG_WNBA_Seasons_Stats_2005_2017
```

Procesos ():

Para la tabla STG_NBA_Seasons_Stats_1950_2017 (5):

- Quitar asterisco del final de los nombres. **REPLACE IN STRING**
- Quedarnos con los valores únicos de los nombres de los jugadores. **SORT ROWS + UNIQUE ROWS**
- Asociar datos de si está activo o no. **DATABASE LOOKUP**
- Asociar datos personales. **DATABASE LOOKUP**
- Ingresar genero Masculino y liga NBA manualmente. **ADD CONSTANT**

Database lookup

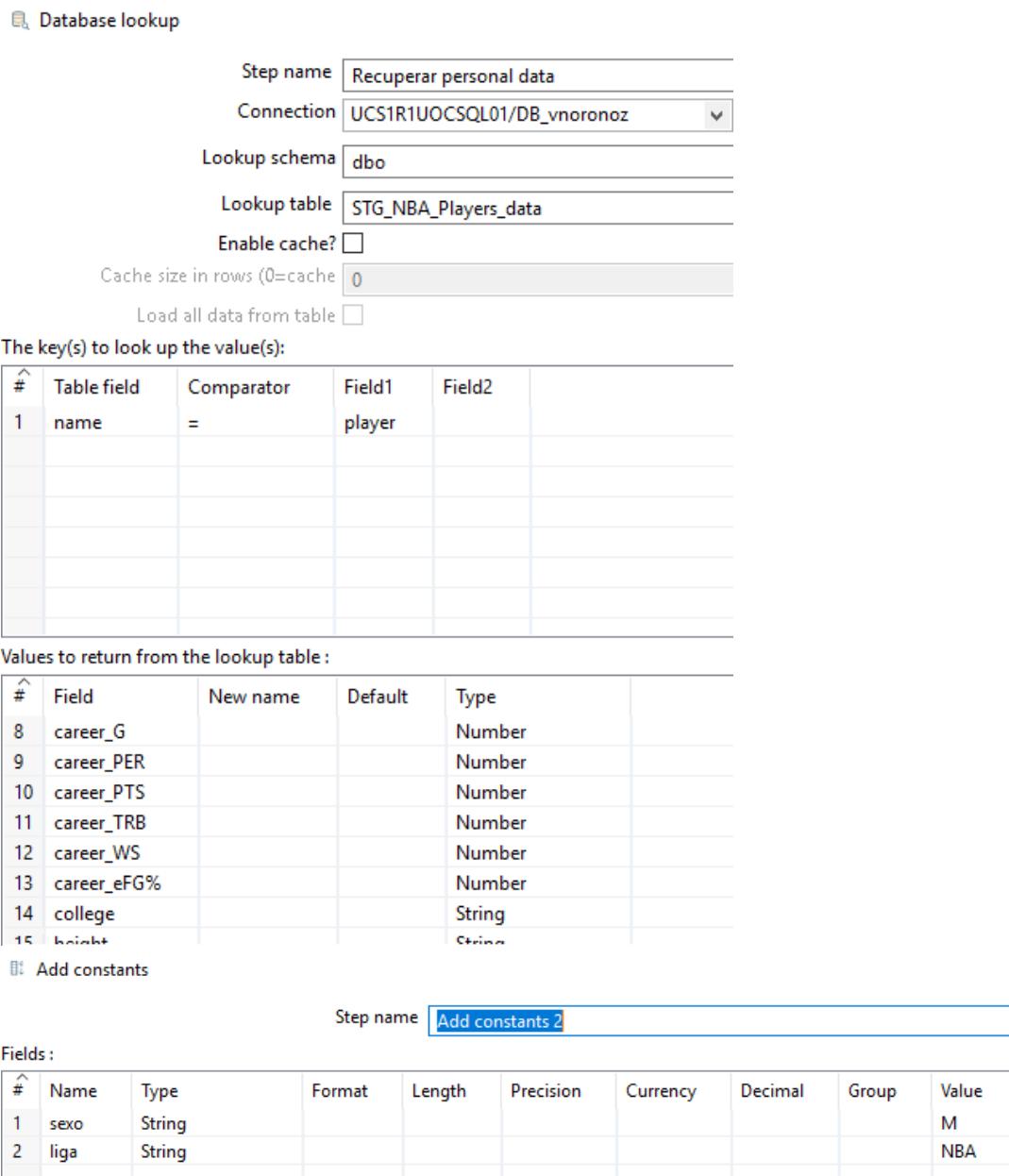
Step name: Recuperar active 22
 Connection: UCS1R1UOCSQL01
 Lookup schema: dbo
 Lookup table: STG_NBA_Players_list
 Enable cache?
 Cache size in rows (0=cache): 0
 Load all data from table

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	player	=	player	

Values to return from the lookup table:

#	Field	New name	Default	Type
1	active			String

Database lookup

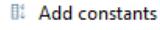
Step name: Recuperar personal data
 Connection: UCS1R1UOCSQL01/DB_vnoronoz
 Lookup schema: dbo
 Lookup table: STG_NBA_Players_data
 Enable cache?
 Cache size in rows (0=cache) 0
 Load all data from table

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	name	=	player	

Values to return from the lookup table :

#	Field	New name	Default	Type
8	career_G			Number
9	career_PER			Number
10	career PTS			Number
11	career TRB			Number
12	career WS			Number
13	career eFG%			Number
14	college			String
15	height			String

 Add constants

Step name: **Add constants 2**

Fields :

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value
1	sexo	String							M
2	liga	String							NBA

Para la tabla STG_WNBA_Seasons_Stats_2005_2017 (6):

- Quitar valor “-” de campo Age. **VALUE MAPPER**
- Desnormalizar datos. **ROW DENORMALISER**
- Quitar registros que tengan player null. **FILTER ROWS**
- Quedarnos con los valores únicos de los nombres de las jugadoras. **SORT ROWS + UNIQUE ROWS**
- Asociar si la jugadora esta en activo o no. **DATABASE LOOKUP**
- Ingresar genero Femenino y liga WNBA manualmente. **ADD CONSTANT**

Row denormaliser

Step name: **Row denormaliser**

The key field: **stat**

The fields that make up the grouping:

#	Group field
1	id_

Target fields:

#	Target fieldname	Value fieldname	Key value	Type	Format	Length
1	season	data_	SEASON	Integer		
2	num#	data_	#	Integer		
3	player	data_	PLAYER	String		
4	team	data_	TEAM	String		
5	age	data_	AGE	Integer		
6	GP	data_	GP	Number		
7	W	data_	W	Number		
8	L	data_	L	Number		
9	MIN	data_	MIN	Number		
10	PTS	data_	PTS	Number		
11	FGM	data_	FGM	Number		
12	FGA	data_	FGA	Number		
13	FG%	data_	FG%	Number		
14	_3PM	data_	3PM	Number		
15	_3PA	data_	3PA	Number		
16	_3P%	data_	3P%	Number		
17	FTM	data_	FTM	Number		
18	FTA	data_	FTA	Number		
19	FT%	data_	FT%	Number		

Filter rows

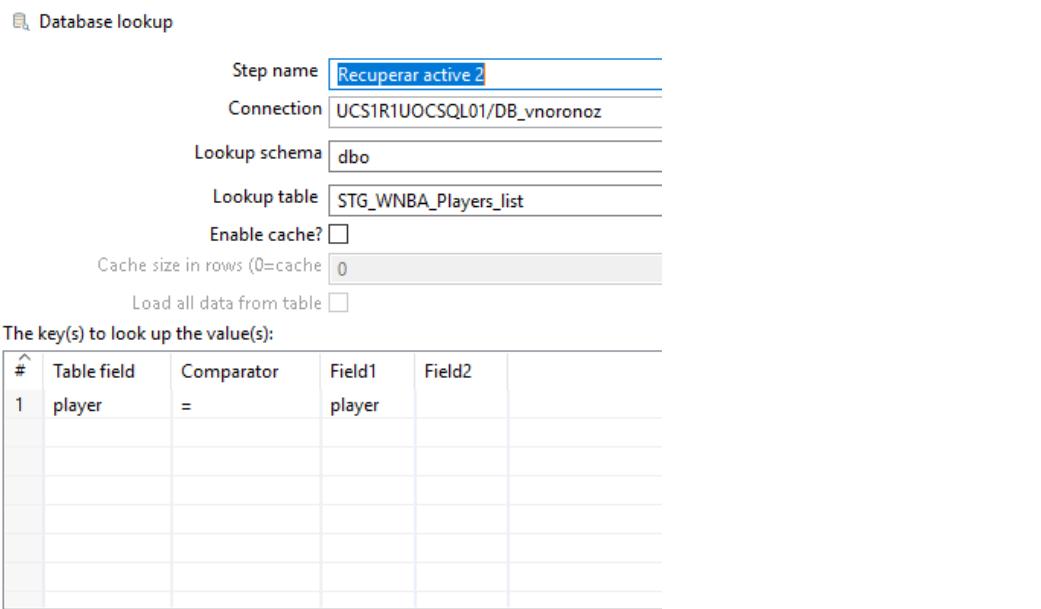
Step name: **Filter rows**

Send 'true' data to step: **Sort rows 2**

Send 'false' data to step: **[]**

The condition:

[] **player** **IS NOT NULL** []
 []

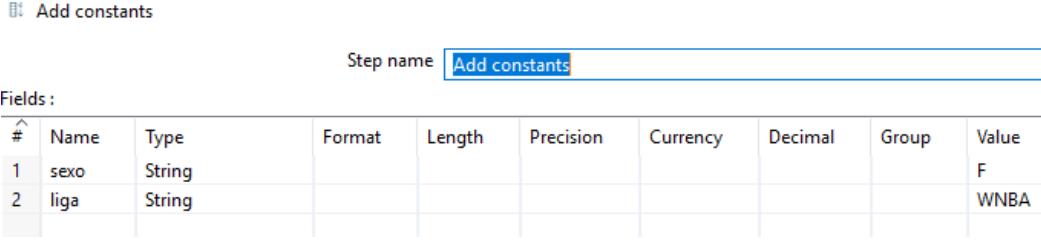


The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	player	=	player	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	active			String



Fields :

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value
1	sexo	String							F
2	liga	String							WNBA

Para ambos:

- Unir datos de jugadores NBA y WNBA para hacer solo una carga. **MULTIWAY MERGE**
- Unir datos de ambas tablas para unificar campos (las veces que sean necesarias). **CONCAT FIELDS**
- Asociar datos división. **DATABASE LOOKUP**
- Añadir secuencia automática para la clave primaria. **ADD SEQUENCE**
- Ordenar tabla por pk. **SORT ROWS**





Concat fields

Step name **Concat fields**

Target Field Name **jugadores**

Length of Target Field **0**

Separator

Enclosure

Fields		Advanced	
#	Name	Type	Format
1	player	String	
2	player_1	String	

Outputs (1):

- Tabla DIM_Jugadores

Table output

Step name **Table Salida DIM_Jugadores**

Connection **UCS1R1UOCSQL01/DB_vnoronoz**

Target schema **dbo**

Target table **DIM_Jugadores**

Commit size **1000**

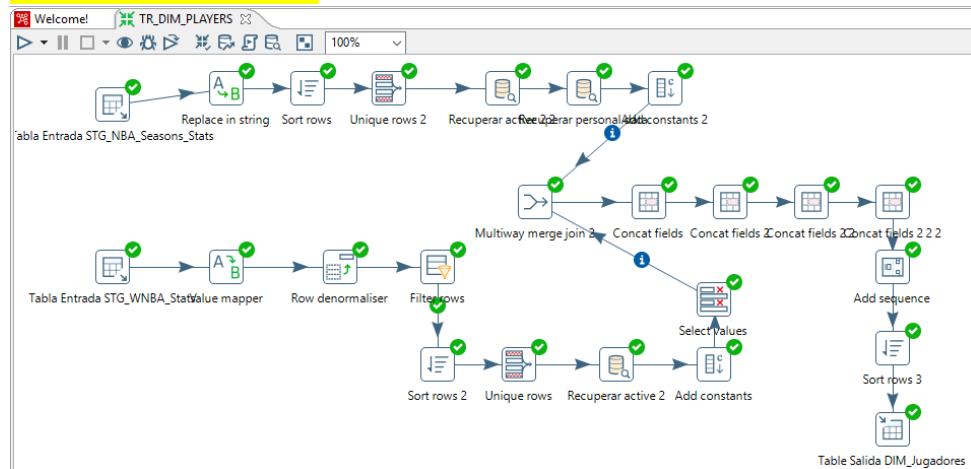
Truncate table

Ignore insert errors

Specify database fields

Main options		Database fields	
Fields to insert:			
#	Table field	Stream field	
8	career_FT%	career_FT%	
9	career_G	career_G	
10	career_PER	career_PER	
11	career PTS	career PTS	
12	career TRB	career TRB	
13	career WS	career WS	
14	career_eFG%	career_eFG%	
15	universidad	college	
16	altura	height	
17	posicion_juego	position	
18	shoots	shoots	
19	peso	weight	
20	sexo	sexs	
21	liga	ligas	
22	activo	activs	
23	pk_jugador	pk_jugador	

Vista de la transformación:



Resultado de la ejecución:

Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)
1 Tabla Entrada STG_WNBA_Stats	0	0	64635	64635	0	0	0	0	Finished	0.6s	114,398
2 Tabla Entrada STG_NBA_Seasons_Stats	0	0	3921	3921	0	0	0	0	Finished	0.3s	13,428
3 Value mapper	0	64635	64635	0	0	0	0	0	Finished	0.6s	114,805
4 Replace in string	0	3921	3921	0	0	0	0	0	Finished	0.3s	12,527
5 Row denormaliser	0	64635	2086	0	0	0	0	0	Finished	0.6s	113,395
6 Filter rows	0	2086	2085	0	0	0	0	0	Finished	0.6s	3,615
7 Sort rows	0	3921	3921	0	0	0	0	0	Finished	0.4s	10,318
8 Unique rows 2	0	3921	3919	0	0	0	0	0	Finished	0.5s	8,084
9 Sort rows 2	0	2085	2085	0	0	0	0	0	Finished	0.6s	3,516
10 Unique rows	0	2085	573	0	0	0	0	0	Finished	0.6s	3,469
11 Recuperar active 2	0	573	573	572	0	0	0	0	Finished	1.2s	465
12 Recuperar active 2 2	0	3919	3919	3703	0	0	0	0	Finished	5.8s	676
13 Recuperar personal data	0	3919	3919	3878	0	0	0	0	Finished	6.2s	632
14 Add constants	0	573	573	0	0	0	0	0	Finished	1.2s	464
15 Add constants 2	0	3919	3919	0	0	0	0	0	Finished	6.2s	632
16 Select values	0	573	573	0	0	0	0	0	Finished	1.2s	462
17 Multiway merge join 2	0	4492	4492	0	0	0	0	0	Finished	6.5s	690
18 Concat fields	0	4492	4492	0	0	0	0	0	Finished	6.5s	690
19 Concat fields 2	0	4492	4492	0	0	0	0	0	Finished	6.5s	689
20 Concat fields 2 2	0	4492	4492	0	0	0	0	0	Finished	6.5s	689
21 Concat fields 2 2 2	0	4492	4492	0	0	0	0	0	Finished	6.5s	688
22 Add sequence	0	4492	4492	0	0	0	0	0	Finished	6.5s	688
23 Sort rows 3	0	4492	4492	0	0	0	0	0	Finished	6.6s	684
24 Table Salida DIM_Jugadores	0	4492	4492	0	4492	0	0	0	Finished	6.9s	655

10) Transformación TR_DIM_TIME

Inputs ():

- Atributo year_ de la tabla STG_NBA_Season_Stats_1950_2017

Table input

Step name: Tabla Entrada STG_NBA_Season_Stats

Connection: UCS1R1UOCSQL01/DB_vnoronoz

SQL:

```
SELECT year_
FROM STG_NBA_Seasons_Stats_1950_2017
GROUP BY year_
ORDER BY year_ ASC
```

Procesos (4):

- Borrar dimensión. **EXECUTE SQL SCRIPT**
- Asociar temporada al año. **VALUE MAPPER**
- Añadir secuencia automática para la clave primaria. **ADD SEQUENCE**
- Ordenar tabla por pk. **SORT ROWS**

A B Value mapper

Step name : **Value mapper**

Fieldname to use : **year_a**

Target field name (empty=overwrite) : **temporada**

Default upon non-matching :

Field values:

#	Source value	Target value
1	1950	1949 - 1950
2	1951	1950 - 1951
3	1952	1951 - 1952
4	1953	1952 - 1953
5	1954	1953 - 1954
6	1955	1954 - 1955
7	1956	1955 - 1956
8	1957	1956 - 1957
9	1958	1957 - 1958
10	1959	1958 - 1959
11	1960	1959 - 1960
12	1961	1960 - 1961
13	1962	1961 - 1962
14	1963	1962 - 1963
15	1964	1963 - 1964
16	1965	1964 - 1965
17	1966	1965 - 1966
18	1967	1966 - 1967
19	1968	1967 - 1968
20	1969	1968 - 1969
21	1970	1969 - 1970
22	1971	1970 - 1971
23	1972	1971 - 1972

Outputs (1):

- Tabla DIM_Tiempo

Table output

Step name : **Table Salida DIM_Tiempo 2**

Connection : **UCS1R1UOCSQL01/DB_vnor**

Target schema : **dbo**

Target table : **DIM_Tiempo**

Commit size : **1000**

Truncate table :

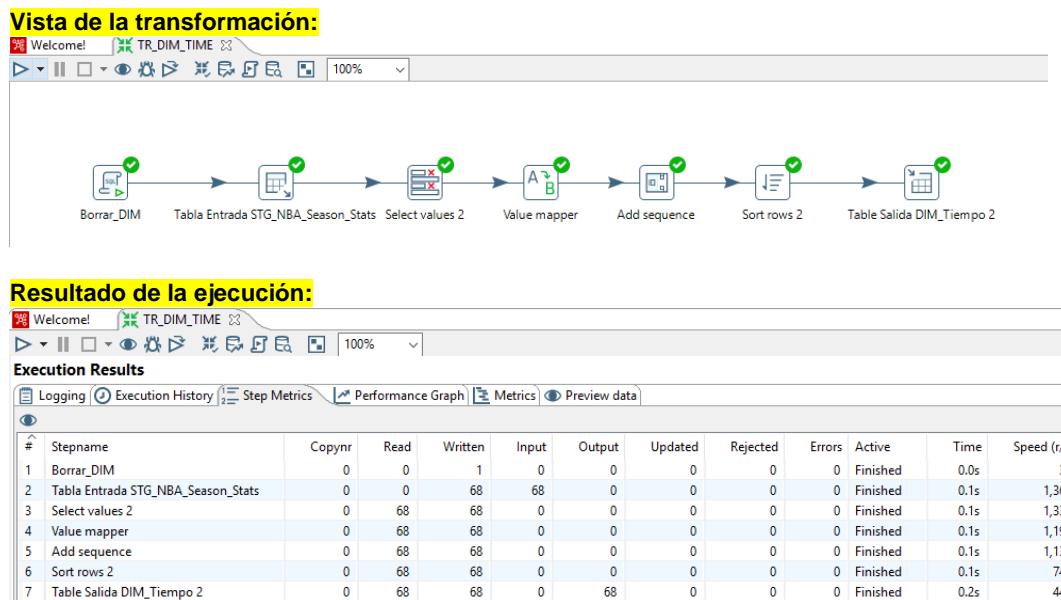
Ignore insert errors :

Specify database fields :

Main options Database fields

Fields to insert:

#	Table field	Stream field
1	date_year	year_a
2	temporada	temporada
3	pk_date	pk_date



Carga y Transformaciones - Bloque TR_FACT

Este bloque, contiene las transformaciones para la carga inicial de las tablas de hecho al almacén desde las tablas intermedias STG_ del Staging Area.

Con la implementación y ejecución de los procesos de carga de dimensiones tendremos pobladas de datos nuestro modelo dimensional y podemos ahora pasar a poblar el modelo de hechos, haciendo referencia a las dimensiones disponibles, mediante sus claves foráneas.

1) Transformación TR_FACT_FREE_THROWS

Inputs (1):

- Tabla STG_NBA_free_throws

Table input

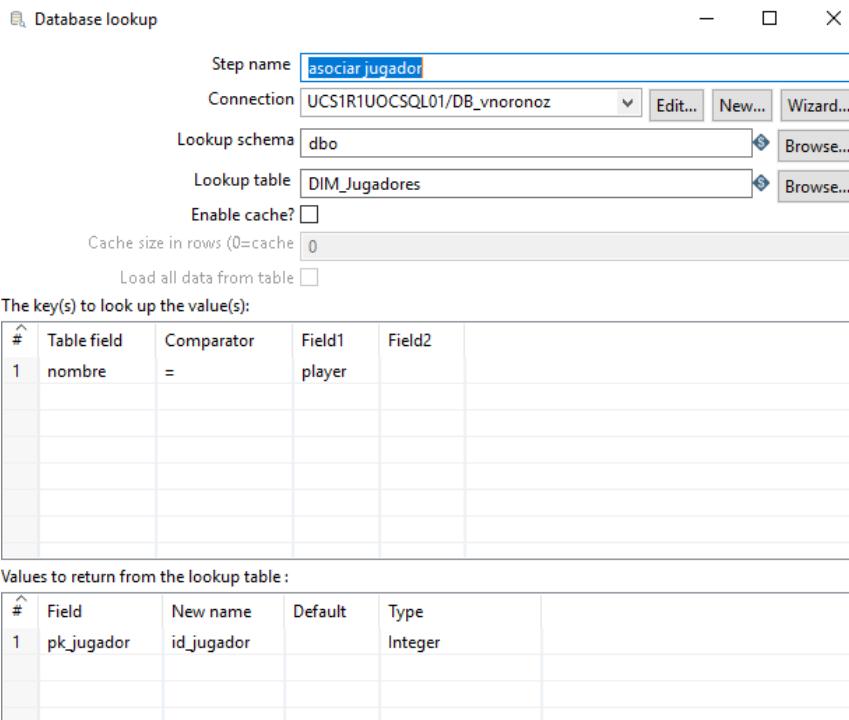
Step name: Tabla Entrada STG_NBA_free_throws

Connection: UCS1R1UOCSQL01/DB_vnoronoz

SQL: `SELECT *
FROM STG_NBA_free_throws`

Procesos (6):

- Asociar datos jugador. **DATABASE LOOKUP**
- Asociar datos temporada. **DATABASE LOOKUP**
- Asociar datos partido. **DATABASE LOOKUP**
- Asociar datos jugada. **DATABASE LOOKUP**
- Asociar datos minuto. **DATABASE LOOKUP**
- Control si alguna clave foránea es null, darle valor 9999. **IF FIELD VALUE IS NULL**



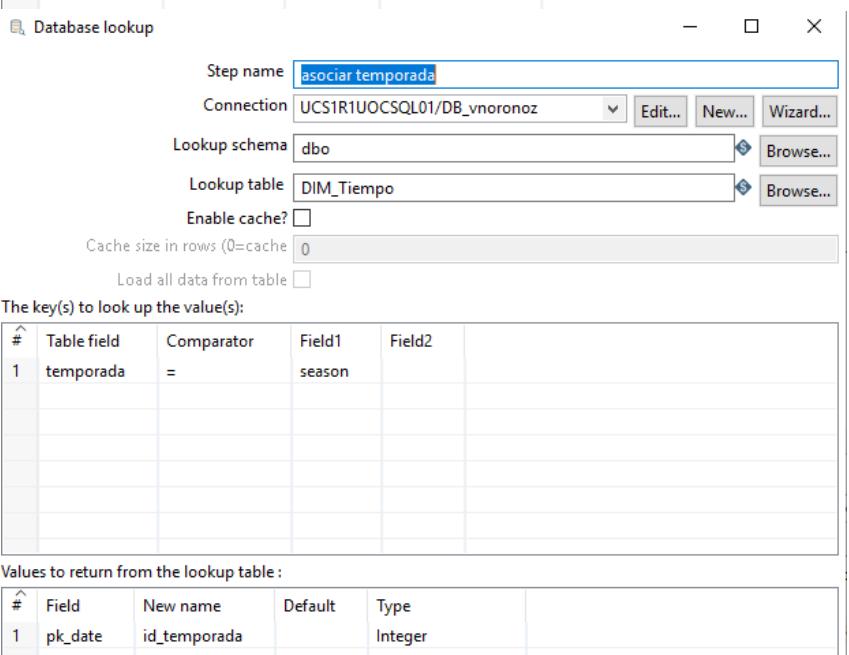
The screenshot shows the configuration for a 'Database lookup' step named 'asociar jugador'. The connection is set to 'UCS1R1UOCSQL01/DB_vnoronoz'. The lookup schema is 'dbo' and the lookup table is 'DIM_Jugadores'. The 'Enable cache?' checkbox is unchecked. The 'Cache size in rows (0=cache)' field contains '0'. The 'Load all data from table' checkbox is unchecked.

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	nombre	=	player	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	pk_jugador	id_jugador		Integer



The screenshot shows the configuration for a 'Database lookup' step named 'asociar temporada'. The connection is set to 'UCS1R1UOCSQL01/DB_vnoronoz'. The lookup schema is 'dbo' and the lookup table is 'DIM_Tiempo'. The 'Enable cache?' checkbox is unchecked. The 'Cache size in rows (0=cache)' field contains '0'. The 'Load all data from table' checkbox is unchecked.

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	temporada	=	season	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	pk_date	id_temporada		Integer

Database lookup

Step name: **asociar partido**

Connection: UCS1R1UOCSQL01/DB_vnoronoz

Lookup schema: **dbo**

Lookup table: **DIM_Partidos**

Enable cache?

Cache size in rows (0=cache) **0**

Load all data from table

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	resultado	=	end_result	
2	partido	=	game	
3	temporada	=	season	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	pk_partido	id_partido		Integer

Database lookup

Step name: **asociar jugada**

Connection: UCS1R1UOCSQL01/DB_vnoronoz

Lookup schema: **dbo**

Lookup table: **DIM_Jugadas**

Enable cache?

Cache size in rows (0=cache) **0**

Load all data from table

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	dec_jugada	=	play	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	pk_jugada	id_jugada		Integer

Database lookup

Step name: **asociar minuto**

Connection: UCS1R1UOCSQL01/DB_vnoronoz

Lookup schema: dbo

Lookup table: DIM_Minutos

Enable cache?

Cache size in rows (0=cache): 0

Load all data from table

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	minutoSegundo	=	time_	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	pk_minutoSegundo	id_minutoSegundo		Integer

If field value is null

Step name: **If field value is null**

Replace Null for all fields

Replace by value

Set empty string?

Mask (Date)

Select fields

Select value type

value types

#	Type	Replace by value	Conversion mask (Date)	Set empty string?

Fields

#	Field	Replace by value	Conversion mask (Date)	Set empty string?
1	id_jugada	9999999		N
2	id_partido	9999		N

Outputs (1):

- Tabla FACT_FREE_THROWS

Table output

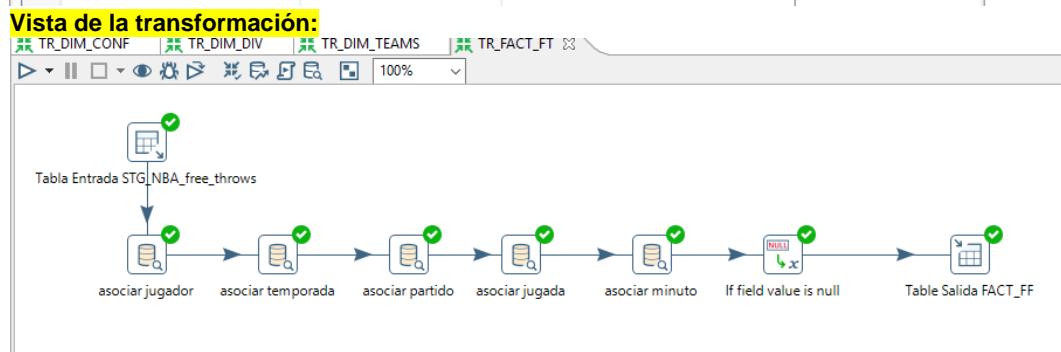
Step name: Table Salida FACT_FF
 Connection: UCS1R1UOCSQL01/DB_vnoronoz
 Target schema: dbo
 Target table: FACT_FREE_THROWS
 Commit size: 1000
 Truncate table:
 Ignore insert errors:
 Specify database fields:

Main options Database fields

Fields to insert:

#	Table field	Stream field
1	periodo_juego	period
2	resultado_tiro_libre	shot_made
3	id_jugador	id_jugador
4	id_temporada	id_temporada
5	id_partido	id_partido
6	id_jugada	id_jugada
7	id_minuto	id_minutoSegundo

Get fields Enter field mapping



Resultado de la ejecución:

Execution Results

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active
1	Tabla Entrada STG_NBA_free_throws	0	0	618019	618019	0	0	0	0	Finished
2	asociar jugador	0	618019	618019	605987	0	0	0	0	Finished
3	asociar temporada	0	618019	618019	618019	0	0	0	0	Finished
4	asociar partido	0	618019	618019	422234	0	0	0	0	Finished
5	asociar jugada	0	618019	618019	618019	0	0	0	0	Finished
6	asociar minuto	0	618019	618019	618019	0	0	0	0	Finished
7	If field value is null	0	618019	618019	0	0	0	0	0	Finished
8	Table Salida FACT_FF	0	618019	618019	0	618019	0	0	0	Finished

2) Transformación TR_FACT_SEASON_STATS

Inputs (2):

- Tabla STG_NBA_Seasons_Stats_1950_2017
- Tabla STG_WNBA_Seasons_Stats_2005_2017

Procesos ():

Para la tabla STG_NBA_Seasons_Stats_1950_2017 (9):

- Quitar asterisco del final de los nombres. **REPLACE IN STRING**
- Separar las posiciones de juego y qudarnos con la primera como la posicion principal. **SPLIT FIELDS**
- Ingresar liga NBA manualmente. **ADD CONSTANT**
- Cálculo estadística EFF. **CALULATOR**
- Asociar datos posición de juego. **DATABASE LOOKUP**
- Asociar datos equipo. **DATABASE LOOKUP**
- Asociar datos temporada. **DATABASE LOOKUP**
- Asociar datos jugador. **DATABASE LOOKUP**
- Control si alguna clave foránea es null, darle valor 9999. **IF FIELD VALUE IS NULL**

Para la tabla STG_WNBA_Seasons_Stats_2005_2017 (9):

- Quitar valor “-” de campo Age. **VALUE MAPPER**
- Desnormalizar datos. **ROW DENORMALISER**
- Quitar registros que tengan player null. **FILTER ROWS**
- Ingresar liga WNBA manualmente. **ADD CONSTANT**
- Cálculo estadísticas 2P, 2PA, 2P%, eFG%, EFF. **CALULATOR**
- Asociar datos equipo. **DATABASE LOOKUP**
- Asociar datos temporada. **DATABASE LOOKUP**
- Asociar datos jugador. **DATABASE LOOKUP**
- Control si alguna clave foránea es null, darle valor 9999. **IF FIELD VALUE IS NULL**

Calculator

Step name

Throw an error on non existing files

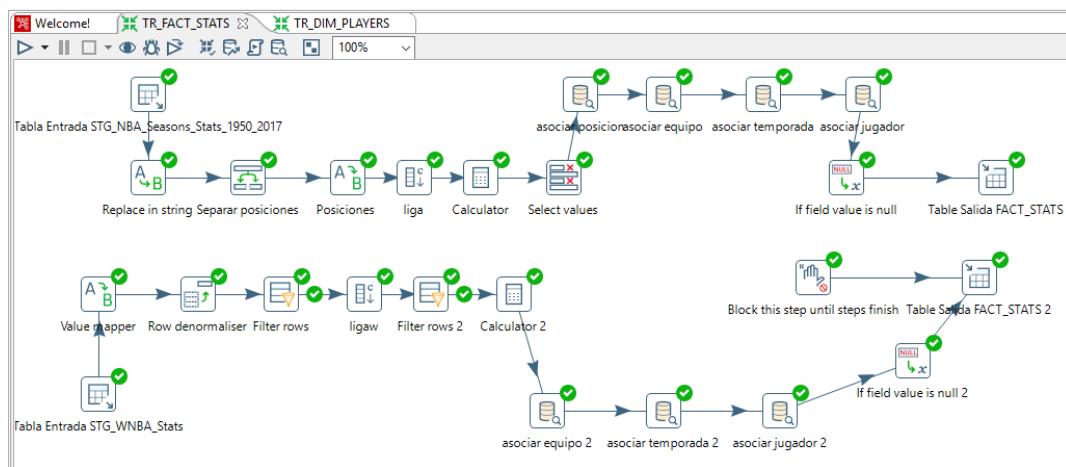
Fields:

#	New field	Calculation	Field A	Field B	Field C	Value type	Length	Precision	Remove
1	2P	A - B	FGM	_3PM		Integer			N
2	2PA	A - B	FGA	_3PA		Integer			N
3	2P%	A / B	2P	2PA		Number		3	N
4	efg1	A + B	FGM	FTM		Integer			N
5	efg2	A + B	FGA	FTA		Integer			N
6	eFG%	A / B	efg1	efg2		Number		3	N
7	int1	A + B + C	PTS	REB	AST	Integer			N
8	int2	A + B + C	int1	STL	BLK	Integer			N
9	missfg	A - B	FGA	FGM		Integer			N
10	missft	A - B	FTA	FTM		Integer			N
11	int3	A + B + C	missfg	missft	TOV	Integer			N
12	eff	A - B	int2	int3		Number		3	N

Outputs (1):

- Tabla FACT_SEASON_STATS

Vista de la transformación:



Resultado de la ejecución:

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)
1	Tabla Entrada STG_WNBA_Stats	0	0	64635	64635	0	0	0	0	Finished	1.3s	50,814
2	Tabla Entrada STG_NBA_Seasons_Stats_1950_2017	0	0	24624	24624	0	0	0	0	Finished	1.3s	19,358
3	Block this step until steps finish	0	0	0	0	0	0	0	0	Finished	21.3s	0
4	Replace in string	0	24624	24624	0	0	0	0	0	Finished	1.3s	19,683
5	Value mapper	0	64635	64635	0	0	0	0	0	Finished	1.2s	51,916
6	Separar posiciones	0	24624	24624	0	0	0	0	0	Finished	1.3s	19,527
7	Posiciones	0	24624	24624	0	0	0	0	0	Finished	1.3s	19,450
8	liga	0	24624	24624	0	0	0	0	0	Finished	1.3s	19,374
9	Row denormaliser	0	64635	2086	0	0	0	0	0	Finished	1.3s	51,625
10	Filter rows	0	2086	2085	0	0	0	0	0	Finished	1.3s	1,661
11	ligaw	0	2085	2085	0	0	0	0	0	Finished	1.3s	1,644
12	Calculator	0	24624	24624	0	0	0	0	0	Finished	4.2s	5,916
13	Select values	0	24624	24624	0	0	0	0	0	Finished	9.6s	2,559
14	asociar posicion	0	24624	24624	24624	0	0	0	0	Finished	16.8s	1,464
15	asociar equipo	0	24624	24624	15441	0	0	0	0	Finished	16.9s	1,457
16	Filter rows 2	0	2085	2007	0	0	0	0	0	Finished	1.3s	1,629
17	Calculator 2	0	2007	2007	0	0	0	0	0	Finished	1.3s	1,552
18	asociar equipo 2	0	2007	2007	1613	0	0	0	0	Finished	2.0s	1,015
19	asociar temporada 2	0	2007	2007	2007	0	0	0	0	Finished	2.5s	809
20	asociar jugador 2	0	2007	2007	0	0	0	0	0	Finished	3.4s	586
21	asociar temporada	0	24624	24624	24624	0	0	0	0	Finished	17.0s	1,449
22	If field value is null 2	0	2007	2007	0	0	0	0	0	Finished	3.4s	585
23	asociar jugador	0	24624	24624	24624	0	0	0	0	Finished	21.2s	1,161
24	Table Salida FACT_STATS 2	0	2007	2007	0	2007	0	0	0	Finished	21.3s	94
25	If field value is null	0	24624	24624	0	0	0	0	0	Finished	21.2s	1,161
26	Table Salida FACT_STATS	0	24624	24624	0	24624	0	0	0	Finished	21.3s	1,158

III. IMPLEMENTACIÓN DE LOS PROCESOS ETL

Teniendo en cuenta los bloques de procesos implementados:

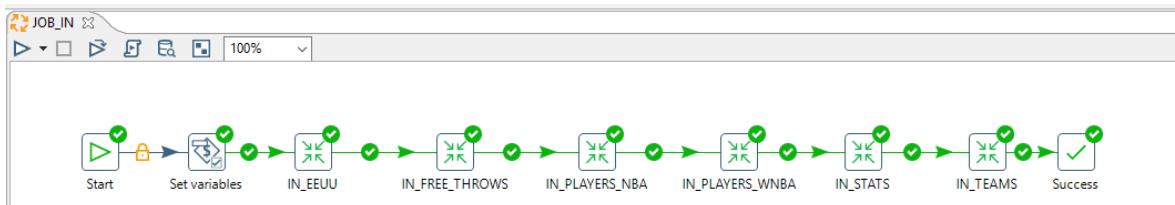
- Bloque IN_: Procesos ETL de transformación y carga al área intermedia.
- Bloque TR_DIM: Procesos ETL de transformación y carga de dimensiones.
- Bloque TR_FACT: Procesos ETL de transformación y carga de hechos.

Vamos a diseñar los trabajos (jobs) mediante PDI que van a permitir la ejecución secuencial de todos los procesos ETL incluidos en cada bloque definido. Cada trabajo contiene como pasos cada una de las transformaciones implementadas en el apartado anterior de Diseño de ETL.

a) Trabajo JOB_IN

El trabajo (job) JOB_IN procesa todas las transformaciones del Bloque IN_ para la carga de datos desde las fuentes de datos proporcionadas al área intermedia (staging area).

El diseño completo del trabajo (job) JOB_IN es el siguiente:



Los pasos incluidos en el trabajo JOB_IN son:

- ✓ **Start:** Componente General de diseño de trabajos, que marca el Inicio del trabajo.
- ✓ **Set variables:** Componente General de diseño de trabajos, que permite la definición de variables de entorno. En nuestro caso definimos las variables de entorno DIR_ENT, con el directorio donde se encuentran las fuentes proporcionadas y BBDD, con la cadena de conexión a la base de datos.
- ✓ **IN_EEUU:** Ejecución de la transformación IN_EEUU.
- ✓ **IN_FREE_THROWS:** Ejecución de la transformación IN_FREE_THROWS.
- ✓ **IN_PLAYERS_NBA:** Ejecución de la transformación IN_PLAYERS_NBA.
- ✓ **IN_PLAYERS_WNBA:** Ejecución de la transformación IN_PLAYERS_WNBA.
- ✓ **IN_STATS:** Ejecución de la transformación IN_STATS.
- ✓ **IN_TEAMS:** Ejecución de la transformación IN_TEAMS.
- ✓ **Success:** Componente General de diseño de trabajos, que marca la Finalización del trabajo.

El resultado de la ejecución de la transformación completa es el siguiente:

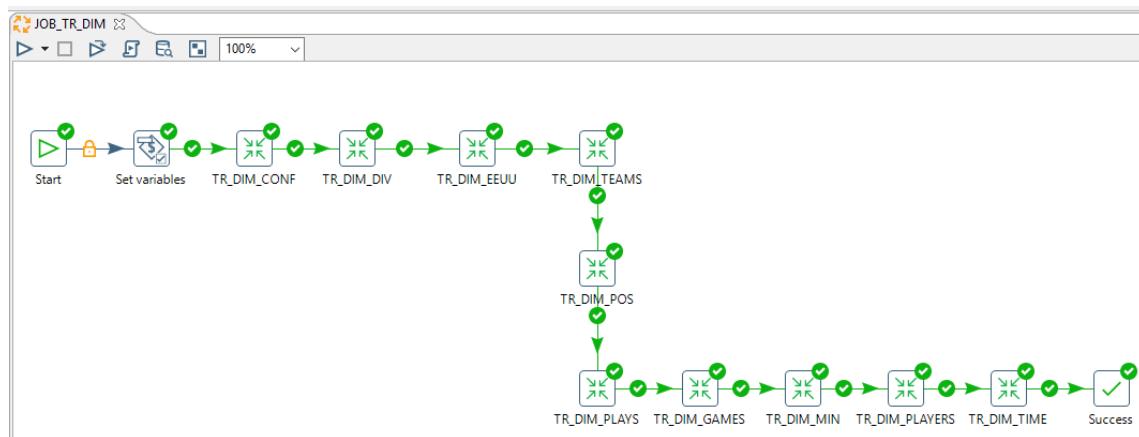
Execution Results			
Job / Job Entry	Comment	Result	Reason
Job: JOB_IN			
Job: JOB_IN	Start of job execution	start	
Start	Start of job execution	start	
Start	Job execution finished	Success	
Set variables	Start of job execution		Followed unconditional link
Set variables	Job execution finished	Success	
IN_EEUU	Start of job execution		Followed link after success
IN_EEUU	Job execution finished	Success	
IN_FREE_THROWS	Start of job execution		Followed link after success
IN_FREE_THROWS	Job execution finished	Success	
IN_PLAYERS_NBA	Start of job execution		Followed link after success
IN_PLAYERS_NBA	Job execution finished	Success	
IN_PLAYERS_WNBA	Start of job execution		Followed link after success
IN_PLAYERS_WNBA	Job execution finished	Success	
IN_STATS	Start of job execution		Followed link after success
IN_STATS	Job execution finished	Success	
IN_TEAMS	Start of job execution		Followed link after success
IN_TEAMS	Job execution finished	Success	
Success	Start of job execution		Followed link after success
Success	Job execution finished	Success	
Job: JOB_IN	Job execution finished	finished	

Se observa el procesamiento con éxito de todos los pasos del JOB_IN, correspondiente a la ejecución de todas las transformaciones que están incluidas trabajo.

b) Trabajo JOB_TR_DIM

El trabajo (job) JOB_TR_DIM procesa todas las transformaciones del bloque TR_DIM para la carga de datos desde las tablas intermedias a las tablas de dimensiones del almacén.

El diseño del trabajo (job) JOB_TR_DIM es la siguiente:



Los pasos incluidos en el trabajo JOB_TR_DIM son:

- ✓ **Start:** Componente General de diseño de trabajos, que marca el Inicio del trabajo.
- ✓ **Set variables:** Componente General de diseño de trabajos, que permite la definición de variables de entorno. En nuestro caso definimos las variables de entorno DIR_ENT, con el directorio donde se encuentran las fuentes proporcionadas y BBDD, con la cadena de conexión a la base de datos.
- ✓ **TR_DIM_CONF:** Ejecución de transformación para la carga de la dimensión DIM_Conferencias.
- ✓ **TR_DIM_DIV:** Ejecución de transformación para la carga de la dimensión DIM_Divisiones.
- ✓ **TR_DIM_EEUU:** Ejecución de transformación para la carga de la dimensión DIM_Estados_EEUU.
- ✓ **TR_DIM_TEAMS:** Ejecución de transformación para la carga de la dimensión DIM_Equipos.
- ✓ **TR_DIM_POS:** Ejecución de transformación para la carga de la dimensión DIM_PosicionesJuego.
- ✓ **TR_DIM_PLAYS:** Ejecución de transformación para la carga de la dimensión DIM_Jugadas.
- ✓ **TR_DIM_GAMES:** Ejecución de transformación para la carga de la dimensión DIM_Partidos.
- ✓ **TR_DIM_MIN:** Ejecución de transformación para la carga de la dimensión DIM_Minutos.
- ✓ **TR_DIM_PLAYERS:** Ejecución de transformación para la carga de la dimensión DIM_Jugadores.
- ✓ **TR_DIM_TIME:** Ejecución de transformación para la carga de la dimensión DIM_Tiempo.
- ✓ **Success:** Componente General de diseño de trabajos, que marca la Finalización del trabajo.

El resultado de la ejecución de la transformación completa es el siguiente:

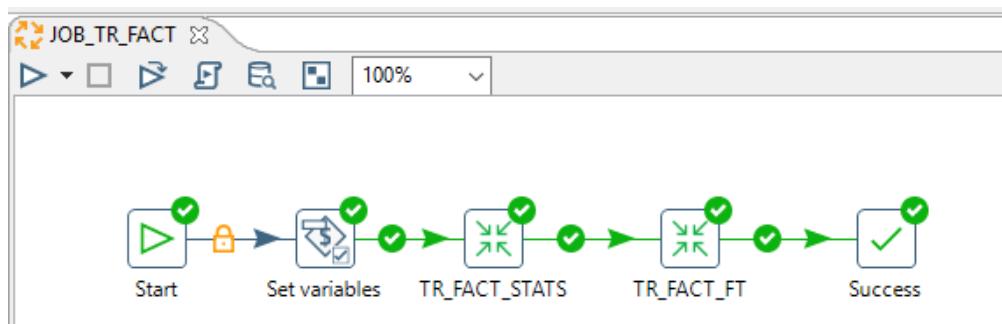
Execution Results			
Job / Job Entry	Comment	Result	Reason
Start	Job execution finished	Success	
Set variables	Start of job execution	Success	Followed unconditional link
Set variables	Job execution finished	Success	
TR_DIM_CONF	Start of job execution	Success	Followed link after success
TR_DIM_CONF	Job execution finished	Success	
TR_DIM_DIV	Start of job execution	Success	Followed link after success
TR_DIM_DIV	Job execution finished	Success	
TR_DIM_EEUU	Start of job execution	Success	Followed link after success
TR_DIM_EEUU	Job execution finished	Success	
TR_DIM_TEAMS	Start of job execution	Success	Followed link after success
TR_DIM_TEAMS	Job execution finished	Success	
TR_DIM_POS	Start of job execution	Success	Followed link after success
TR_DIM_POS	Job execution finished	Success	
TR_DIM_PLAYS	Start of job execution	Success	Followed link after success
TR_DIM_PLAYS	Job execution finished	Success	
TR_DIM_GAMES	Start of job execution	Success	Followed link after success
TR_DIM_GAMES	Job execution finished	Success	
TR_DIM_MIN	Start of job execution	Success	Followed link after success
TR_DIM_MIN	Job execution finished	Success	
TR_DIM_PLAYERS	Start of job execution	Success	Followed link after success
TR_DIM_PLAYERS	Job execution finished	Success	
TR_DIM_TIME	Start of job execution	Success	Followed link after success
TR_DIM_TIME	Job execution finished	Success	
Success	Start of job execution	Success	Followed link after success
Success	Job execution finished	Success	
Job: JOB_TR_DIM	Job execution finished	Success	finished

Se observa del resultado de la ejecución del trabajo, que todos los pasos han sido procesados con éxito. Con la finalización de este proceso, se realizan las cargas de todas las dimensiones del modelo multidimensional del almacén.

c) Trabajo JOB TR_FACT

El trabajo (job) JOB_TR_FACT procesa todas las transformaciones del bloque TR_FACT para la carga de datos desde las tablas intermedias a las tablas de hechos del almacén.

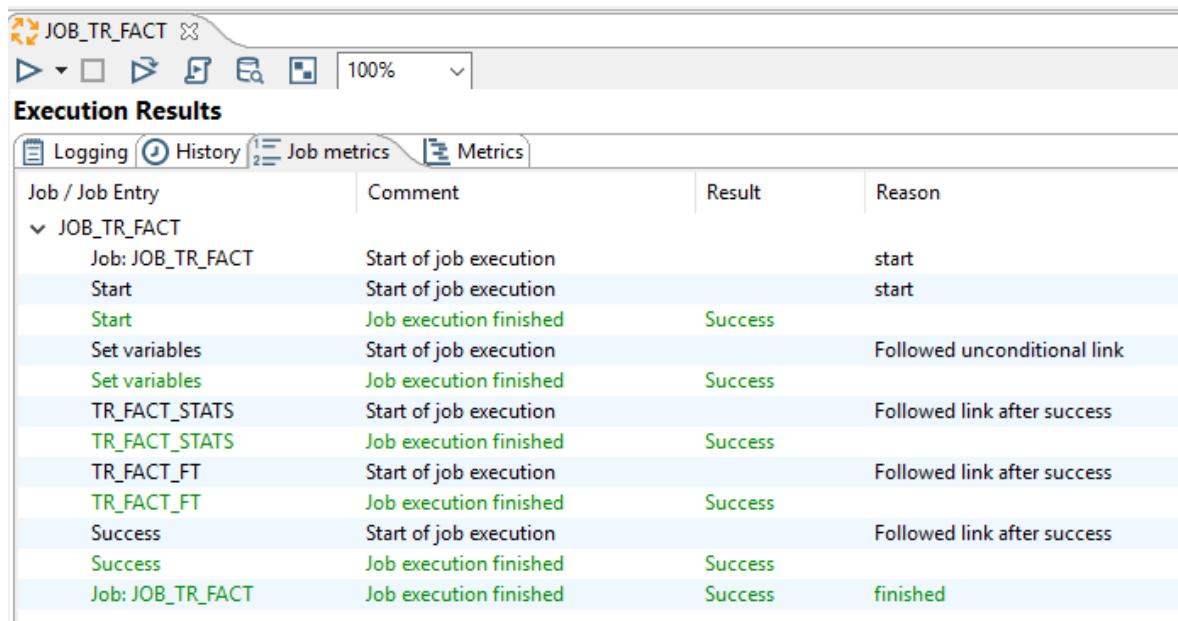
El diseño del trabajo (job) JOB_TR_FACT es la siguiente:



Los pasos incluidos en el trabajo JOB_TR_FACT son:

- ✓ **Start:** Componente General de diseño de trabajos, que marca el Inicio del mismo.
- ✓ **Set variables:** Componente General de diseño de trabajos, que permite la definición de variables de entorno. En nuestro caso definimos las variables de entorno DIR_ENT, con el directorio donde se encuentran las fuentes proporcionadas y BBDD, con la cadena de conexión a la base de datos.
- ✓ **TR_FACT_FREE_THROWS:** Ejecución de la transformación para la carga de la tabla de hechos FACT_FREE_THROWS.
- ✓ **TR_FACT_SEASON_STATS:** Ejecución de la transformación para la carga de la tabla de hechos FACT_SEASON_STATS.
- ✓ **Success:** Componente General de diseño de trabajos, que marca la Finalización del trabajo.

El resultado de la ejecución de la transformación completa es el siguiente:



The screenshot shows the 'Execution Results' window for the 'JOB_TR_FACT' job. The window has tabs at the top: Logging, History, Job metrics (selected), and Metrics. The main area displays a table of execution details:

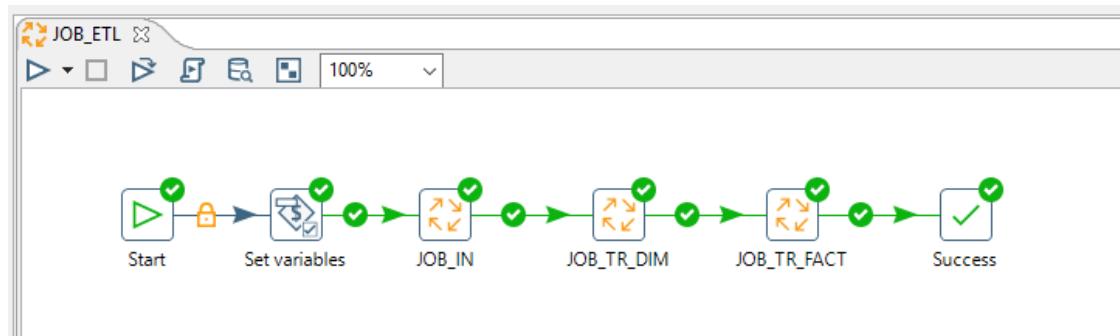
Job / Job Entry	Comment	Result	Reason
Job: JOB_TR_FACT			
Job: JOB_TR_FACT	Start of job execution	start	
Start	Start of job execution	start	
Start	Job execution finished	Success	
Set variables	Start of job execution		Followed unconditional link
Set variables	Job execution finished	Success	
TR_FACT_STATS	Start of job execution		Followed link after success
TR_FACT_STATS	Job execution finished	Success	
TR_FACT_FT	Start of job execution		Followed link after success
TR_FACT_FT	Job execution finished	Success	
Success	Start of job execution		Followed link after success
Success	Job execution finished	Success	
Job: JOB_TR_FACT	Job execution finished	Success	finished

Se observa del resultado de la ejecución del trabajo, que todos los pasos se procesan con éxito. Con la finalización de este proceso, se realizarán las cargas de todas las tablas de hechos del modelo multidimensional al almacén.

d) Trabajo JOB_ETL

El trabajo (job) JOB_ETL procesa todos los trabajos con todos los procesos de extracción, transformación y carga del almacén de personas egresadas.

El diseño del trabajo (job) JOB_ETL es la siguiente:



Los pasos incluidos en el trabajo JOB_ETL son:

- ✓ **Start:** Componente General de diseño de trabajos, que marca el Inicio del mismo.
- ✓ **Set variables:** Definición de las variables de entorno, DIR_ENT y BBDD
- ✓ **Job_IN:** Ejecución de trabajo del Bloque IN, para la carga al área intermedia.
- ✓ **Job_TR_DIM:** Ejecución de trabajo del Bloque TR_DIM, para la carga de dimensiones.
- ✓ **Job_TR_FACT:** Ejecución de trabajo del Bloque TR_FACT, para la carga de tablas de hechos.
- ✓ **Success:** Componente General de diseño de trabajos, que marca la Finalización del trabajo.

El resultado de la ejecución de la transformación completa es el siguiente:

The screenshot shows the 'Execution Results' window for the 'JOB_ETL' job. The window has tabs at the top: Logging, History, Job metrics, and Metrics. The History tab is selected. It displays a table of execution events:

Job / Job Entry	Comment	Result	Reason	Filename
Job: JOB_ETL				
Job: JOB_ETL	Start of job execution	start		
Start	Start of job execution	start		
Start	Job execution finished	Success		
Set variables	Start of job execution		Followed unconditional link	
Set variables	Job execution finished	Success		
JOB_IN	Start of job execution		Followed link after success	
Job: JOB_IN				
JOB_IN	Job execution finished	Success		
JOB_TR_DIM	Start of job execution		Followed link after success	
Job: JOB_TR_DIM				
JOB_TR_DIM	Job execution finished	Success		
JOB_TR_FACT	Start of job execution		Followed link after success	
Job: JOB_TR_FACT				
JOB_TR_FACT	Job execution finished	Success		
Success	Start of job execution		Followed link after success	
Success	Job execution finished	Success		
Job: JOB_ETL	Job execution finished	Success	finished	

La ejecución completa y con éxito del JOB_ETL supone la ejecución de los procesos ETL para la carga de los datos al almacén integrado de datos de la NBA y WNBA desde las fuentes origen hasta el modelo multidimensional que se ha diseñado.

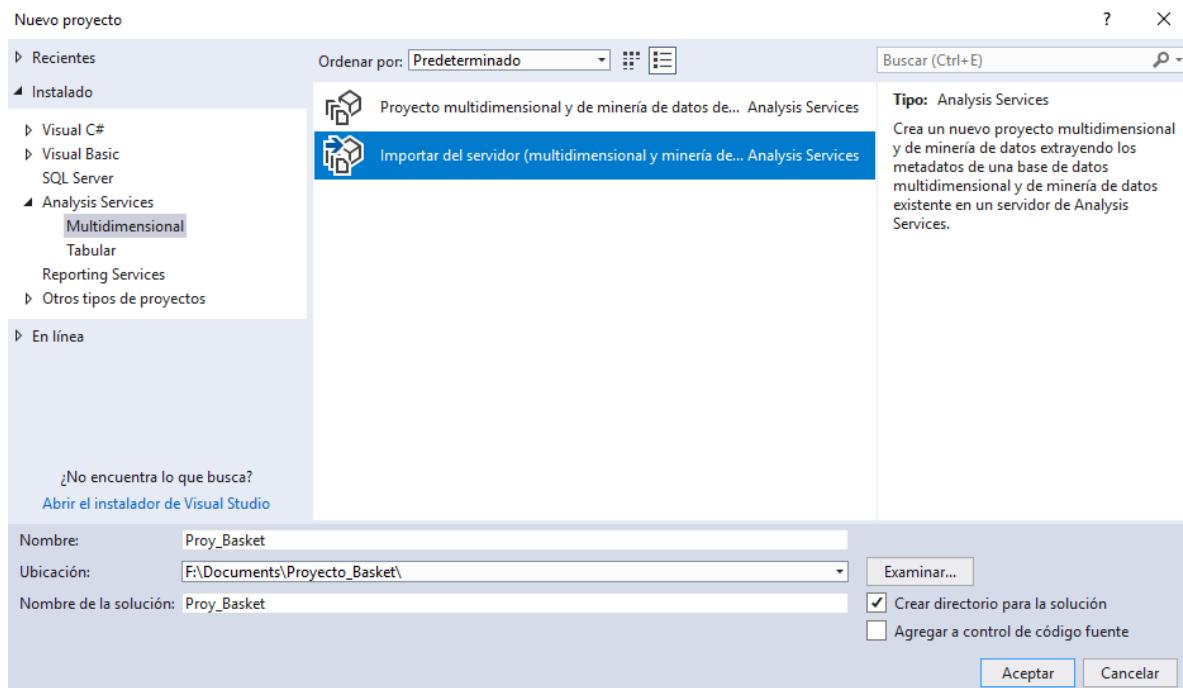
e) Vistas de las tablas pobladas con los datos en la BD de SQL SERVER

En la sección de Anexos del presente documento se puede revisar la efectividad de la carga de datos a la base de datos proporcionada mediante los procesos descritos en las secciones anteriores.

IV. DISEÑO Y CREACIÓN DEL MODELO OLAP

a) Creación del proyecto

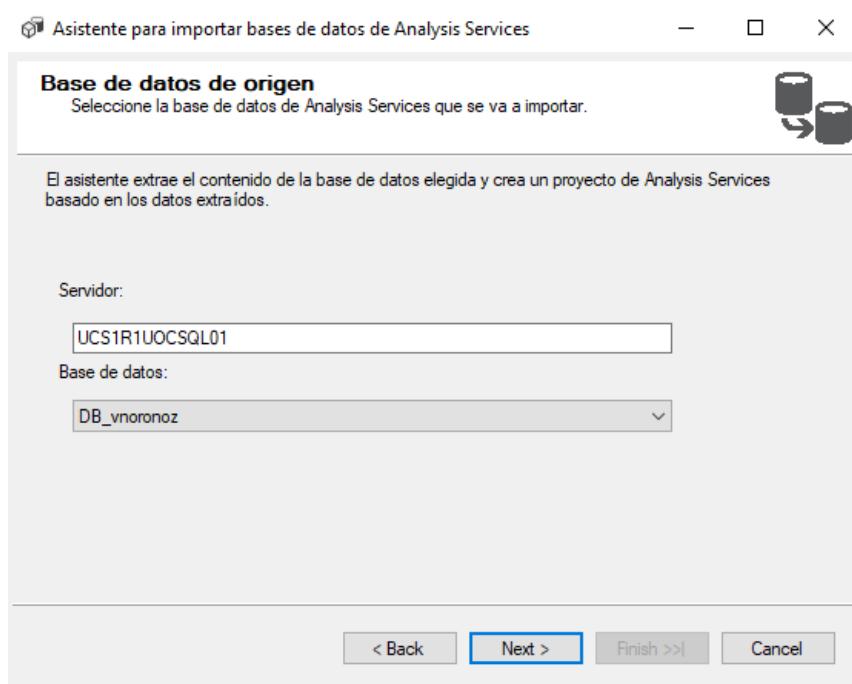
Iniciamos Visual Studio y seleccionamos nuevo proyecto. El tipo de proyecto que nos interesa esta vez es el “proyecto multidimensional y de minería de datos” que tanto permite crear cubos como proyectos de minería de datos. Seleccionamos Analysis Services -> Multidimensional -> Importar del servidor.



Para nuestro caso, creamos el proyecto de nombre “**Proy_BASKET**”.

Configuración base de datos de origen

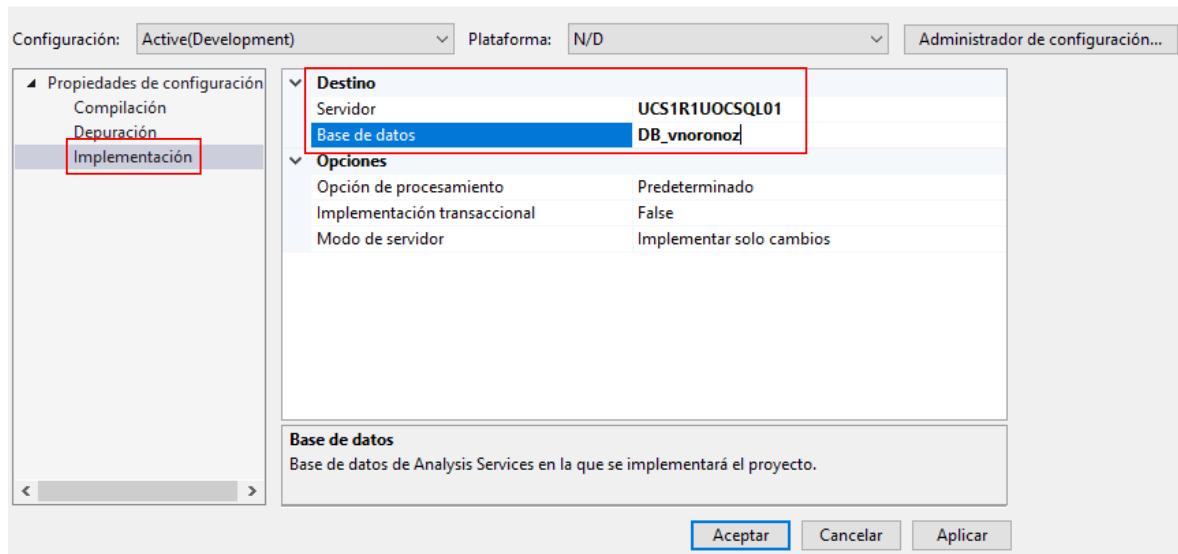
Una vez hecho esto, importamos la estructura de la base de datos de Analysis Services, seleccionando como Origen de la base de datos nuestra base de datos (en mi caso DB_vnoronoz) que se encuentra en el servidor MS SQL server UCS1R1UOCSQL01.



Configuración destino de datos

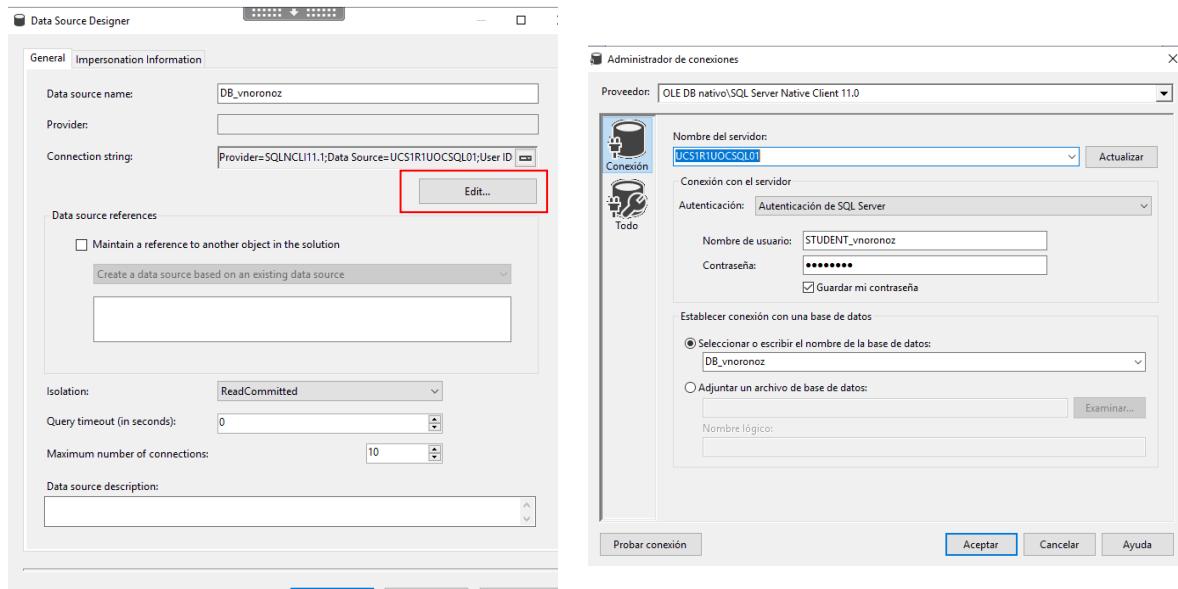
Aunque hemos importado en el paso anterior la base de datos, en el actual proyecto debemos indicarle que lo guarde en el servidor. Para realizar esta configuración, vamos a las propiedades del proyecto en el explorador de soluciones.

En propiedades vamos al apartado Implementación y cambiamos el valor de Servidor por el nombre del MS SQL server UCS1R1UOCSQL01 y en Base de datos configuramos la nuestra, DB_vnoronoz.



b) Origen de datos

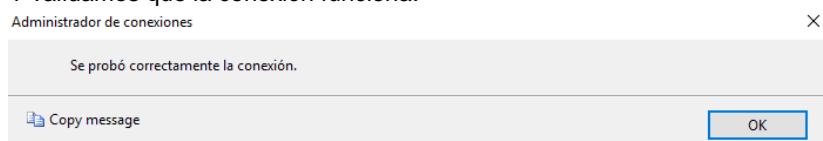
Una vez creado el proyecto, tenemos su estructura en el lateral. Esta estructura está vacía y debemos completarla para poder crear la estructura MOLAP. El primer paso consiste en definir el origen de los datos. Para poder configurar que el origen de los datos sea nuestra base de datos del MS SQL server, debemos realizar doble click sobre el nombre de la base de datos en el Explorador de Soluciones. Con esta acción se abrirá una nueva ventana con título Data Source Designer:



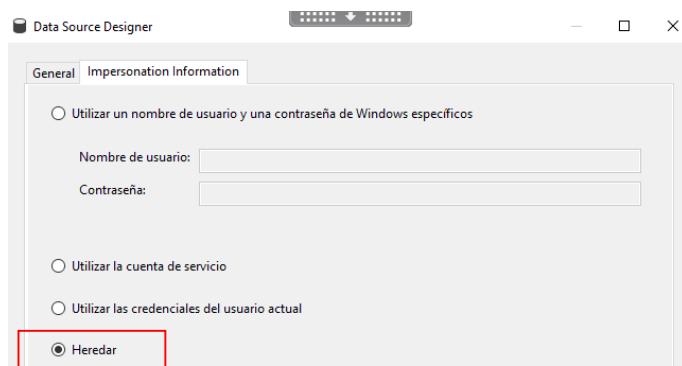
Pulsando sobre el botón Edit... se abrirá el administrador de conexiones donde rellenaremos los parámetros válidos para nuestra conexión:

- Nombre del servidor: **UCS1R1UOCSQL01**
- Tipo de autenticación: **Autenticación de SQL Server**
- Login y password: **STUDENT_vnoronoz** y la contraseña establecida en MS SQL Server. Activamos la casilla *Guardar mi contraseña* para no tener que introducirla en cada acción.
- Nombre de la base de datos: **DB_vnoronoz**

Y validamos que la conexión funciona:



Una vez editada la conexión de datos, pulsamos sobre *Impersonation Information* y nos aseguramos que la opción seleccionada es **Heredar**.



c) Vista del Origen de datos

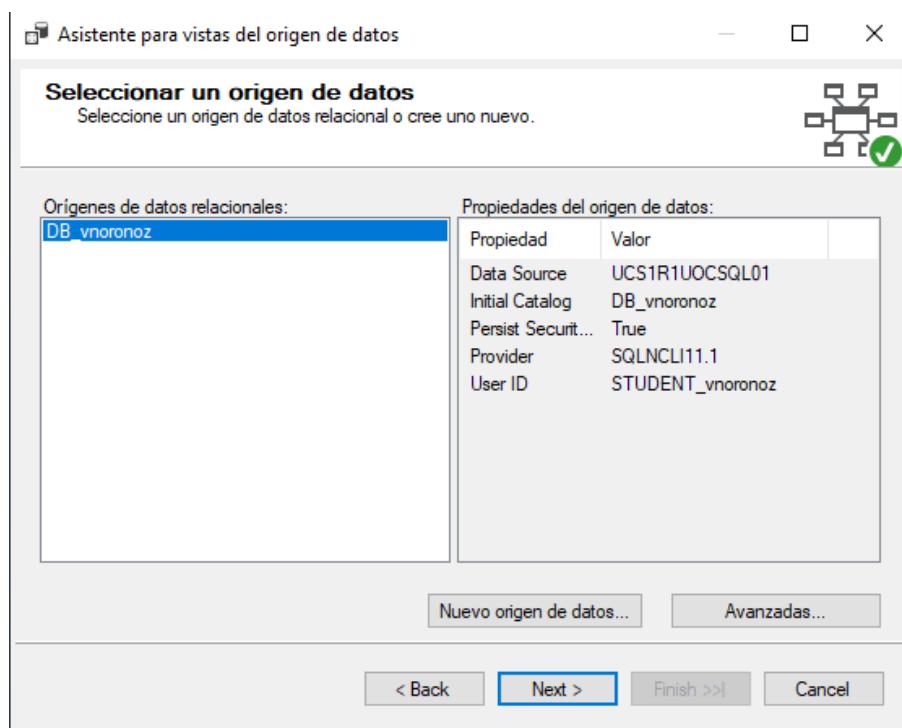
Una vez creada la fuente de origen, pasamos al siguiente paso que consiste en crear las vistas del origen de datos. Este paso suele ser muy importante en las estructuras MOLAP dado que definen el alcance de la misma.

En nuestro caso vamos a definir 2 vistas de orígenes de datos:

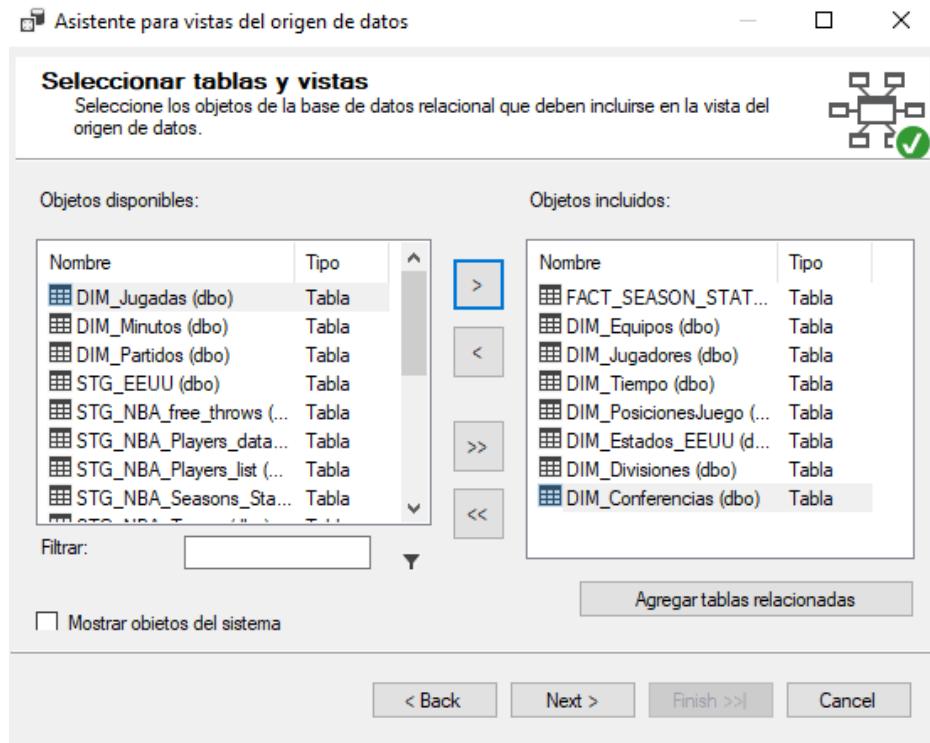
- **View_SS:** nos dará acceso a las tablas para diseñar el cubo que se usará para realizar un seguimiento por temporadas de las estadísticas de juego de los jugadores y jugadoras, para evaluar su evolución en el tiempo y analizar su rendimiento, según el equipo y posición en la que juegan.
- **View_FT:** nos dará acceso a las tablas para diseñar el cubo que se usará para estudiar detalladamente los tiros libres realizados durante las temporadas y analizar su impacto sobre el resultado final de los partidos.

Para crear una nueva vista, podemos hacerlo utilizando menú Proyecto -> Nueva vista del origen de datos o en el Explorador de Soluciones, pulsando botón derecho sobre Vistas del origen de datos -> Nueva vista del origen de datos

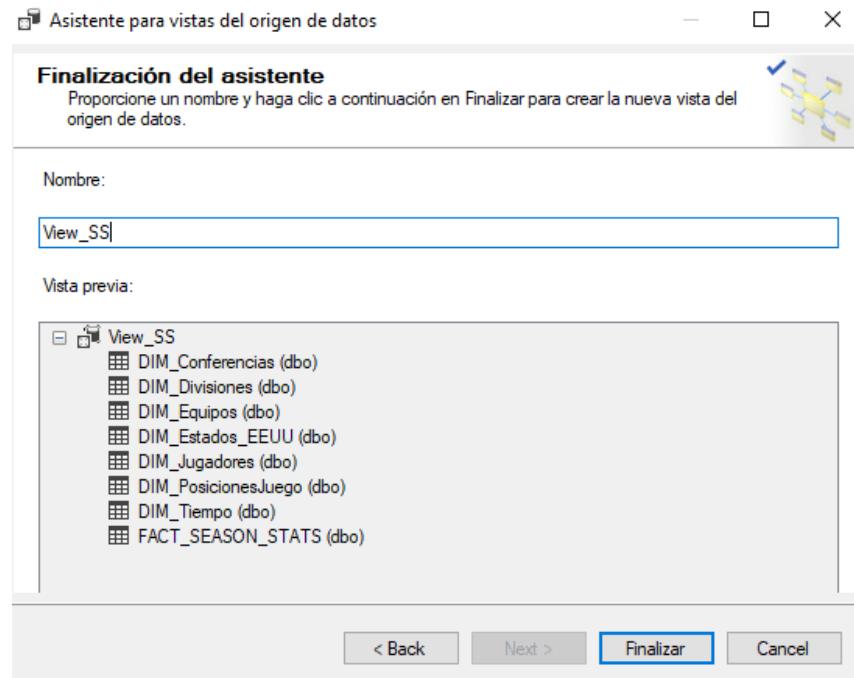
Esto nos abre un Asistente para vistas del origen de datos, donde en la primera pantalla, seleccionamos el origen de datos creado anteriormente, DB_vnoronoz:



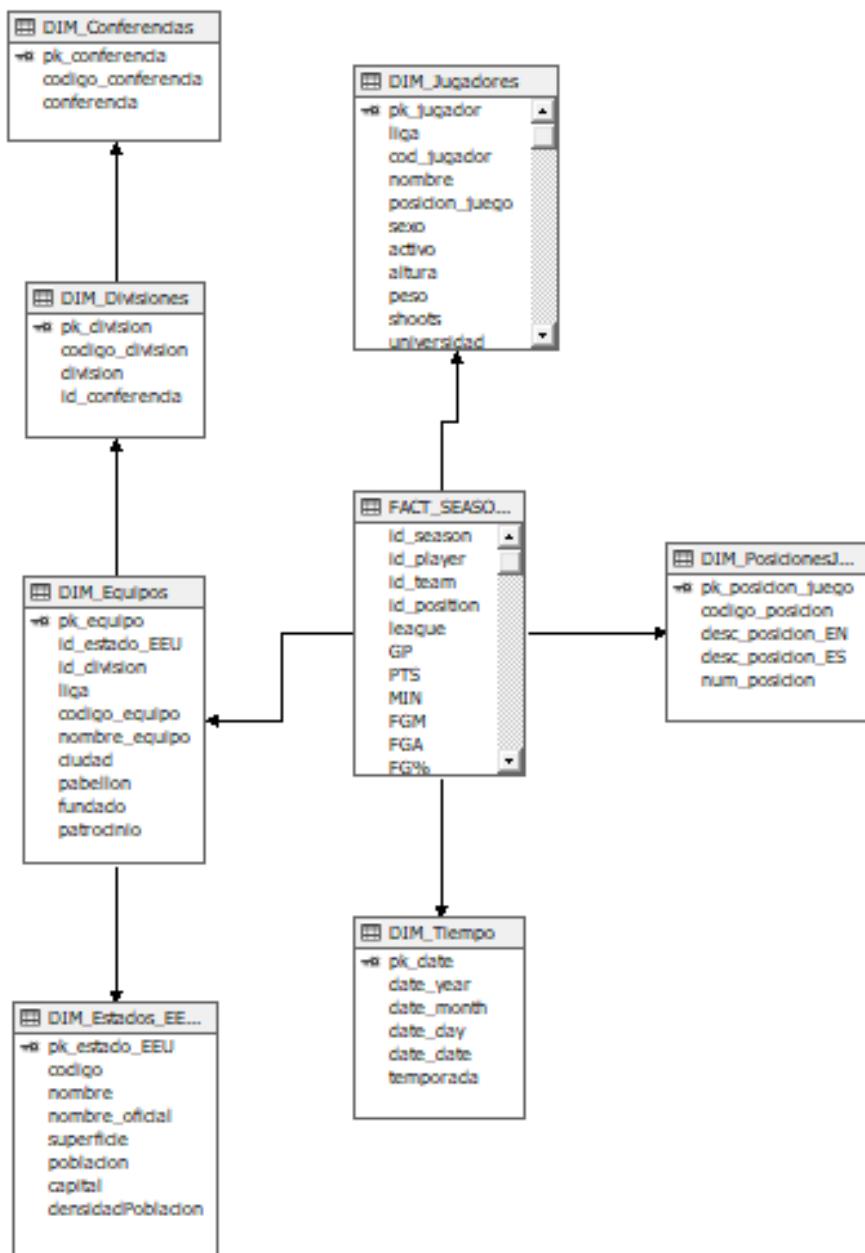
En la siguiente pantalla seleccionamos las tablas involucradas en el primer caso View_SS que sirven para analizar las estadísticas de los jugadores y jugadoras desde diferentes perspectivas: la tabla de hechos FACT_SEASON_STATS y todas las dimensiones asociadas DIM_Equipos, DIM_Estadis_EEUU, DIM_Conferencias, DIM_Divisiones, DIM_Jugadores, DIM_Tiempo y DIM_PosicionesJuego.



En la pantalla final revisamos que las tablas elegidas sean las correctas y asignamos un nombre a la vista de orígenes de datos, para este caso View_SS.



Vista de View_SS:



Se procede de manera análoga para la creación de la vista View_FT que sirve para analizar las estadísticas del juego (los tiros libres) desde diferentes perspectivas. Se seleccionan la tabla de hechos FACT_FREE_THROWS y sus tablas de dimensiones asociadas DIM_Jugadas, DIM_Minutos, DIM_Jugadores, DIM_Partidos y DIM_Tiempo.

Asistente para vistas del origen de datos

Seleccionar tablas y vistas
Seleccione los objetos de la base de datos relacional que deben incluirse en la vista del origen de datos.

Objetos disponibles:

Nombre	Tipo
STG_NBA_Players_data...	Tabla
STG_NBA_Players_list (...	Tabla
STG_NBA_Seasons_Sta...	Tabla
STG_NBA_Teams (dbo)	Tabla
STG_Team_Codes (dbo)	Tabla
STG_WNBA_Players_list...	Tabla
STG_WNBA_Seasons_...	Tabla
STG_WNBA_Teams (dbo)	Tabla

Filtrar: _____

Objetos incluidos:

Nombre	Tipo
FACT_FREE_THROW...	Tabla
DIM_Jugadas (dbo)	Tabla
DIM_Minutos (dbo)	Tabla
DIM_Jugadores (dbo)	Tabla
DIM_Partidos (dbo)	Tabla
DIM_Tiempo (dbo)	Tabla

Mostrar objetos del sistema

Agregar tablas relacionadas

< Back Next > Finish >> Cancel

Asistente para vistas del origen de datos

Finalización del asistente
Proporcione un nombre y haga clic a continuación en Finalizar para crear la nueva vista del origen de datos.

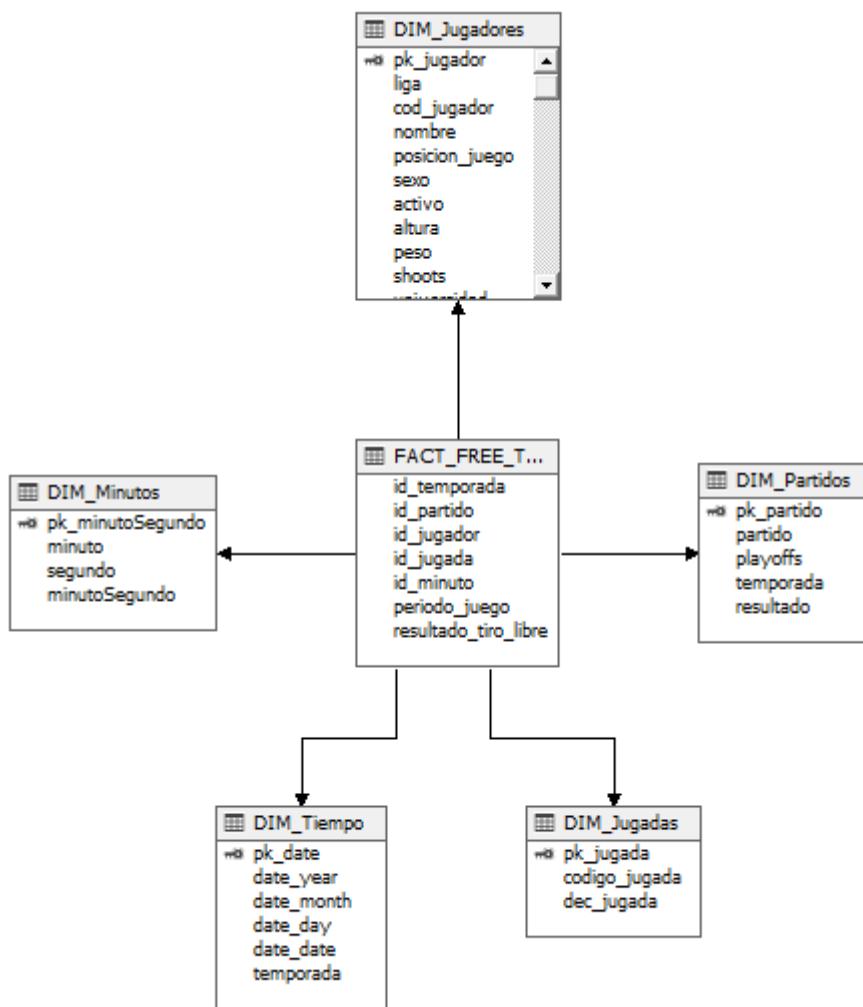
Nombre: **View_FT**

Vista previa:

- View_FT
 - FACT_FREE_THROWS (dbo)
 - DIM_Jugadas (dbo)
 - DIM_Minutos (dbo)
 - DIM_Jugadores (dbo)
 - DIM_Partidos (dbo)
 - DIM_Tiempo (dbo)

< Back Next > Finalizar Cancel

Vista de View_FT:



En el Explorador de Soluciones podemos ver las dos vistas creadas:

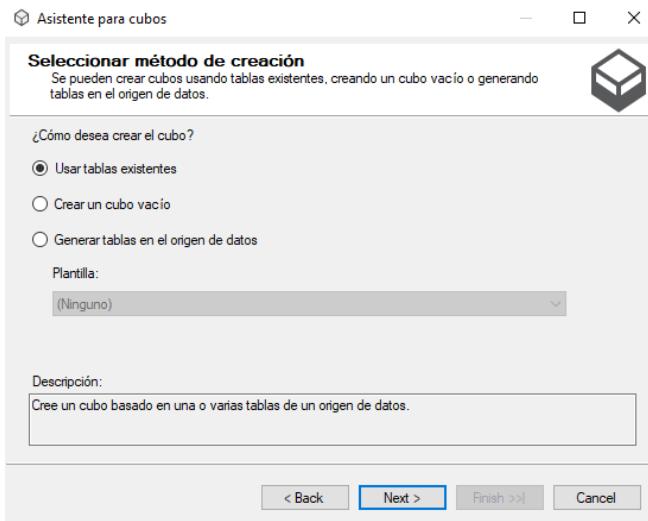
- ◀ **Vistas del origen de datos**
 - default.csv
 - View_FT.csv
 - View_SS.csv

d) Creación de Cubos

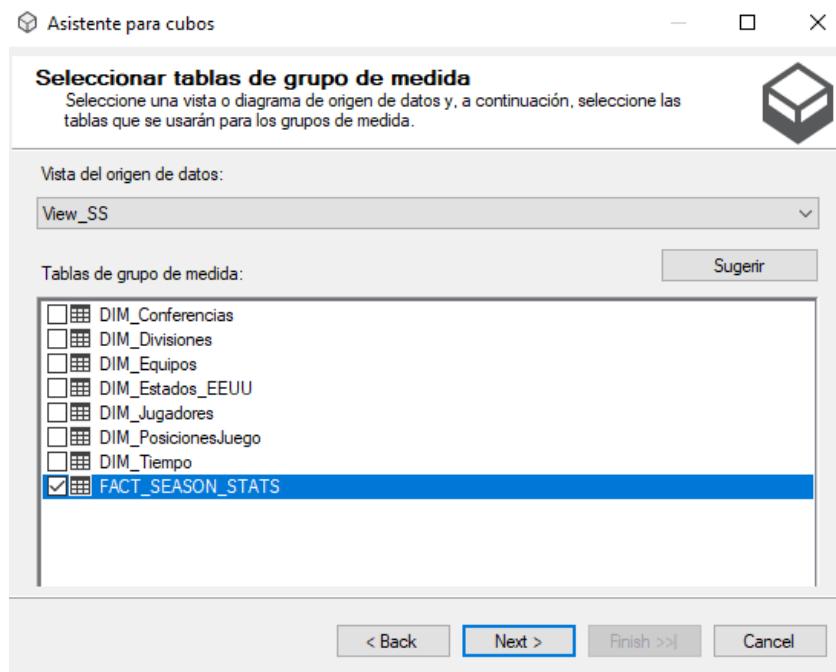
Ahora ya está todo preparado para crear los cubos con los modelos multidimensionales correspondientes. Vamos a crear 2 cubos, uno para cada vista de origen de datos:

- **Cube_SS:** que se corresponderá con la vista View_SS.
- **Cube_FT:** que se corresponderá con la vista View_FT.

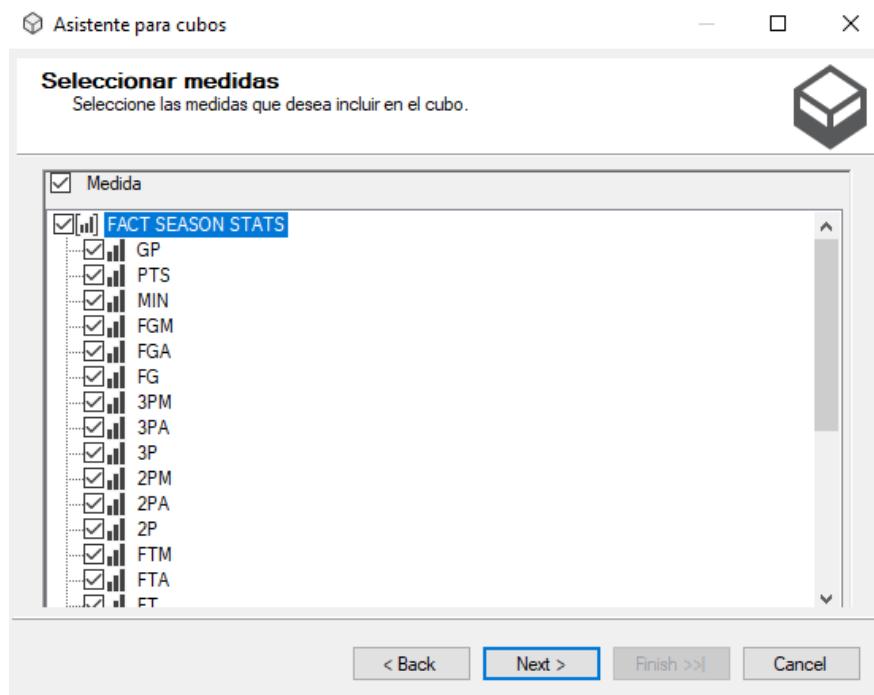
Para crear un nuevo cubo, podemos hacerlo utilizando menú Proyecto -> Nuevo cubo o en el Explorador de Soluciones, pulsando botón derecho sobre Cubos -> Nuevo cubo. Esto nos abre un Asistente para cubos, donde seleccionamos el método de creación. En este caso como queremos crear un cubo sobre nuestros datos, seleccionamos Usar tablas existentes.



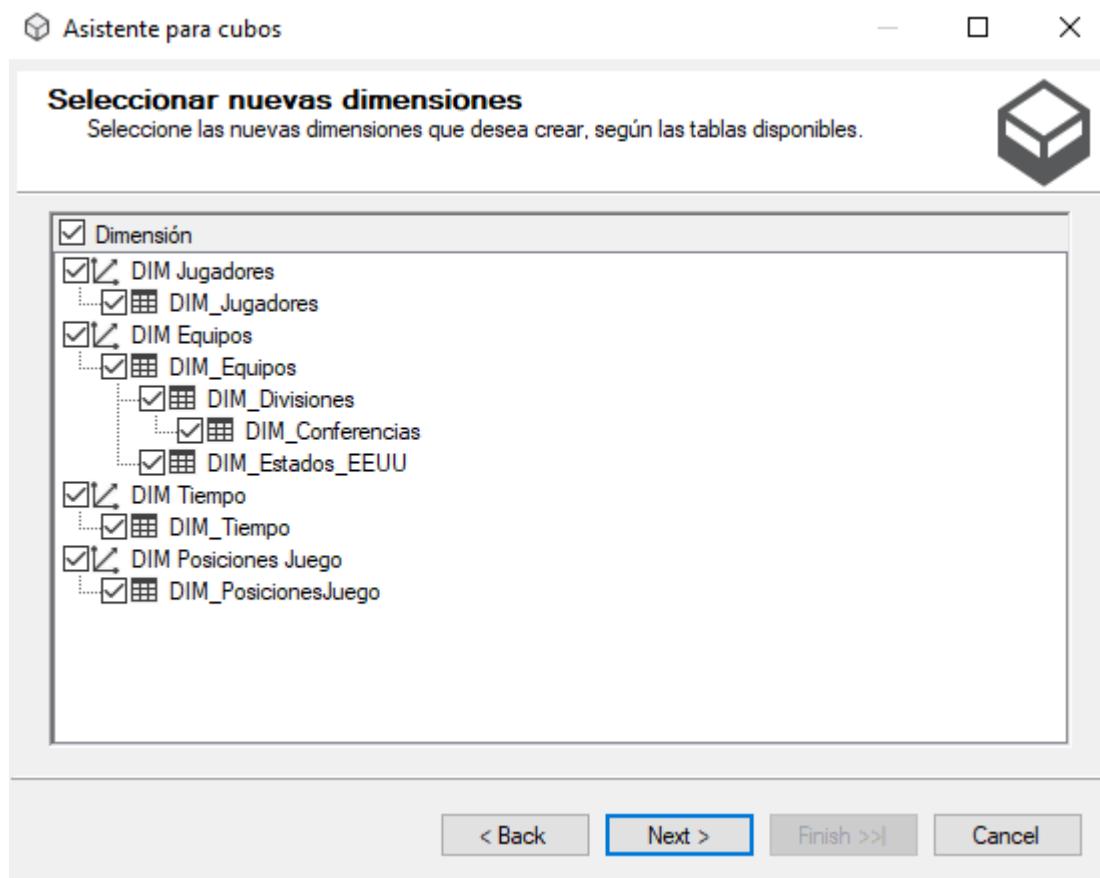
Seleccionamos la vista de origen de datos base para la creación de este cubo. En este caso, View_SS. Igualmente seleccionamos las tablas de medidas (hechos) a utilizar en el cubo. Para este caso será FACT_SEASON_STATS.



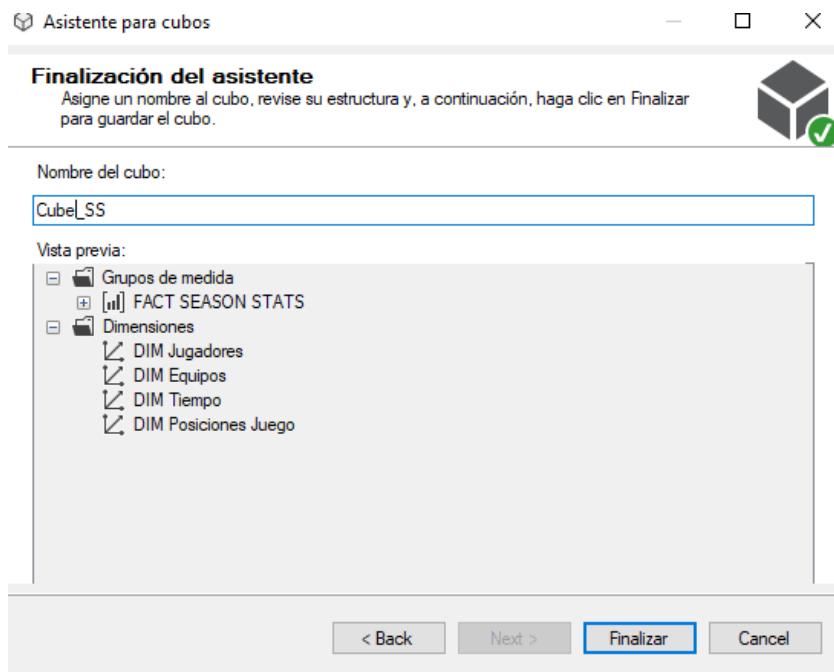
Seleccionamos las métricas de la tabla de hechos:



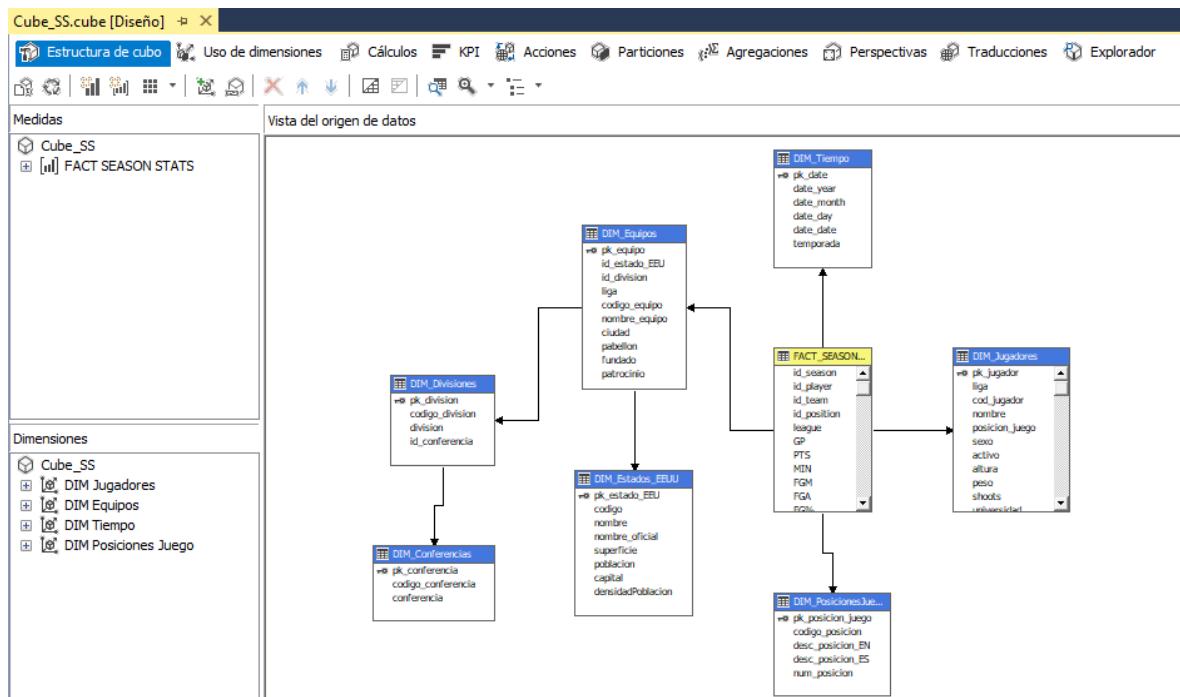
Y después seleccionamos las dimensiones asociadas:



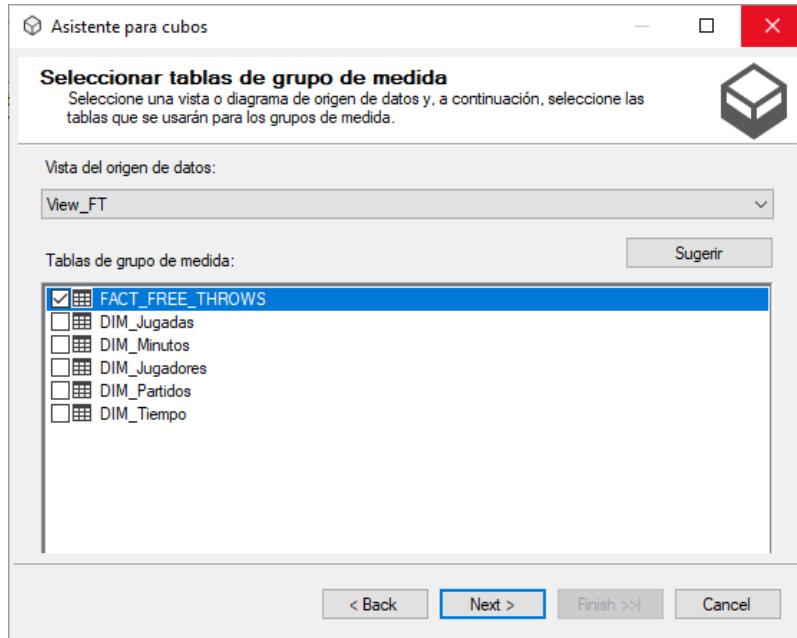
Por último, revisamos un resumen del cubo que se va a crear y le asignamos nombre, a este Cube_SS.



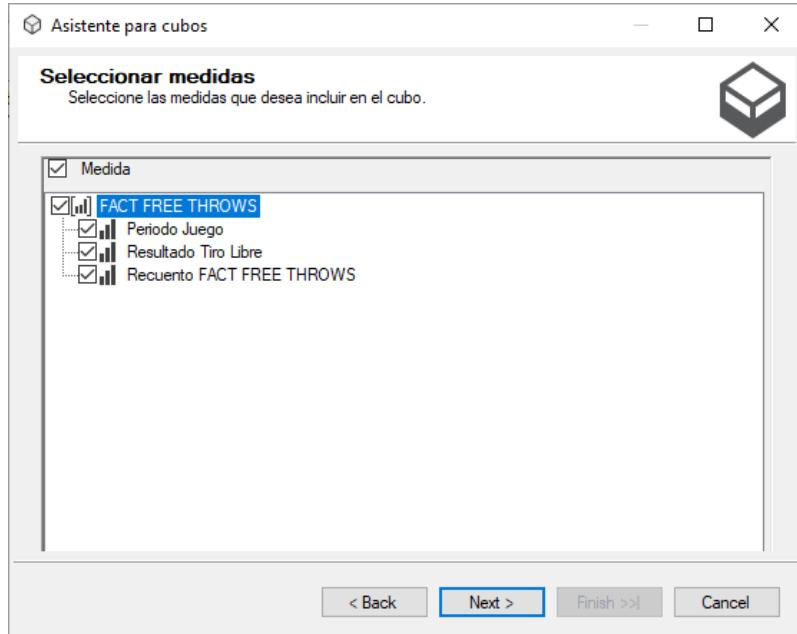
Vista de Cube_SS:



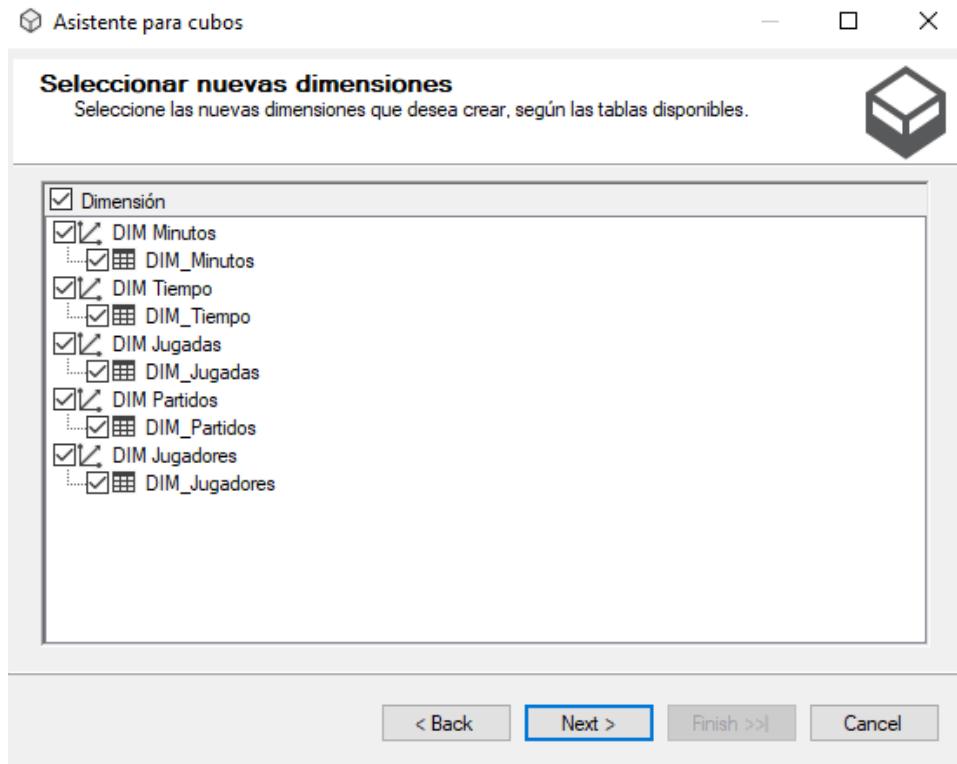
Seguimos ahora con el caso del segundo cubo Cube_FT. Seleccionamos la vista de origen de datos base para la creación de este cubo. En este caso, View_FT. Igualmente seleccionamos las tablas de medidas (hechos) a utilizar en el cubo. Para este caso será FACT_FREE_THROWS.



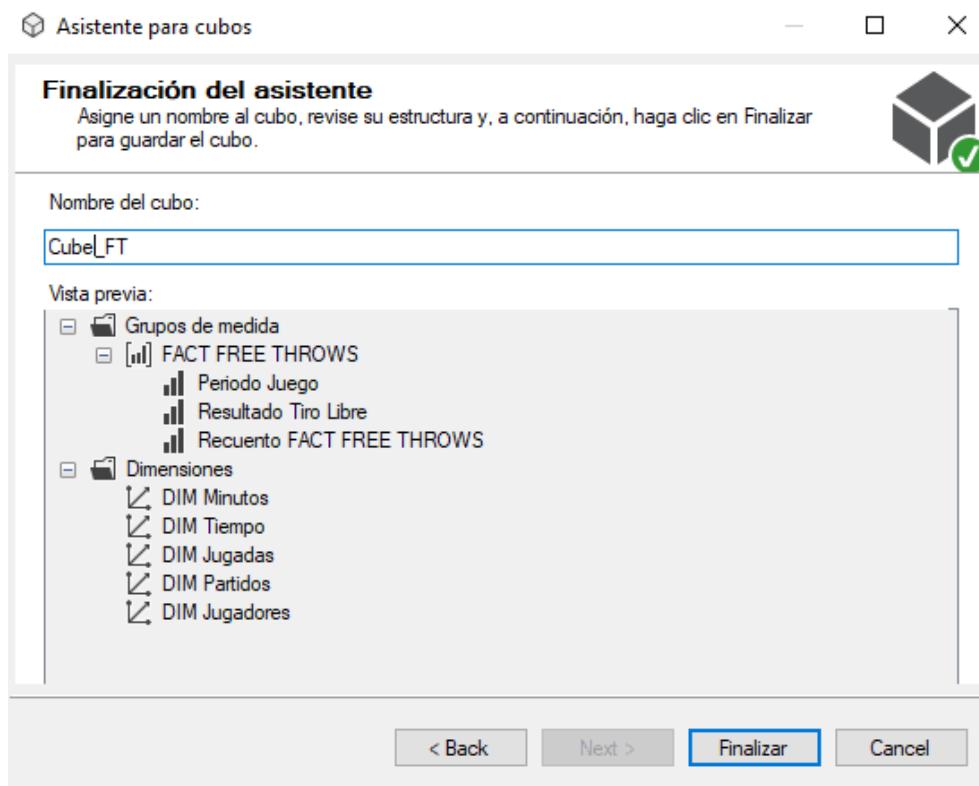
Seleccionamos las métricas de la tabla de hechos:



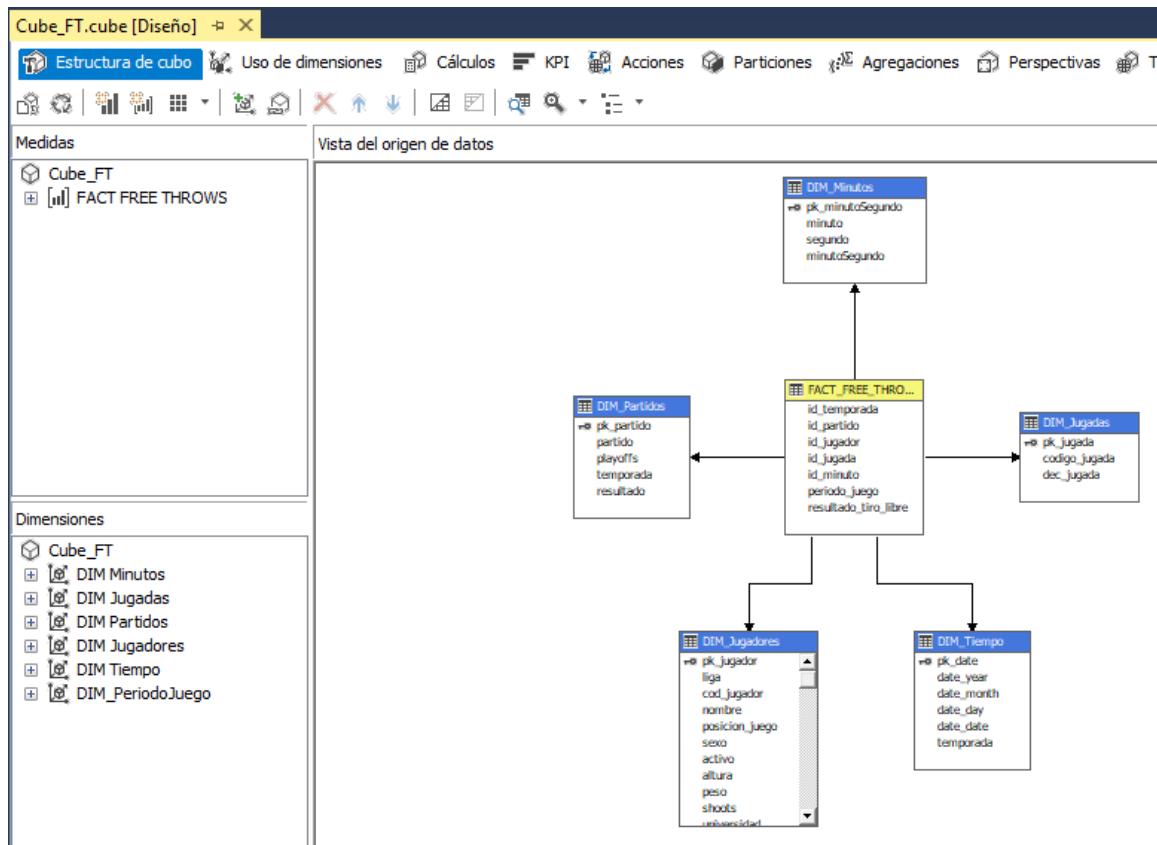
Y después seleccionamos las dimensiones asociadas:



Por último, revisamos un resumen del cubo que se va a crear y le asignamos nombre, a este Cube_FT.



Vista de Cube_FT:

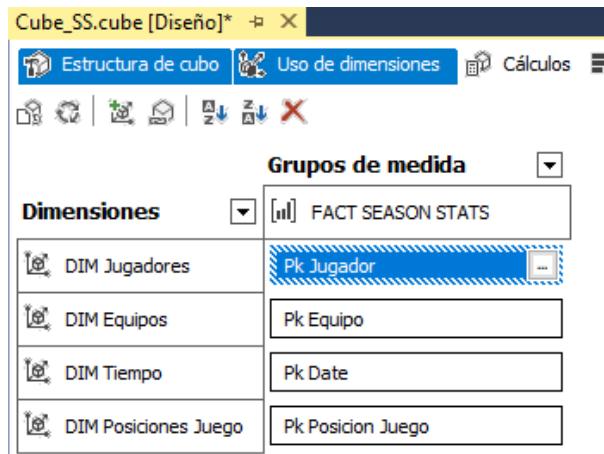


e) Dimensiones, atributos y relaciones en el Cubo

Una vez definidos los cubos, vamos a definir los atributos de las dimensiones que harán parte de los análisis requeridos.

CUBE_SS

Primeramente revisemos que ya están bien definidas las relaciones entre las dimensiones y la tabla de medida según las claves foráneas.



Después, para cada dimensión añadimos los atributos que vamos a necesitar en el análisis:

DIM Posiciones Juego.dim [Diseño]* -> **Cube_SS.cube [Diseño]***

Estructura de dimensión | Relaciones de atributo | Traducciones | Explorador

Atributos | Jerarquías | Vista del origen de datos

Atributos

- DIM Posiciones Juego
 - Desc Posición ES
 - Pk Posición Juego

Jerarquías

Para crear un nuevo atributo, arrastra un atributo hasta aquí.

Vista del origen de datos

DIM_PosicionesJuego

- pk_posicion_juego
- codigo_posicion
- desc_posicion_EN
- desc_posicion_ES
- num_posicion

DIM Tiempo.dim [Diseño]* -> **Cube_SS.cube [Diseño]***

Estructura de dimensión | Relaciones de atributo | Traducciones | Explorador

Atributos | Jerarquías | Vista del origen de datos

Atributos

- DIM Tiempo
 - Date Date
 - Date Year
 - Pk Date
 - Temporada

Jerarquías

Para crear un nuevo atributo, arrastra un atributo hasta aquí.

Vista del origen de datos

DIM_Tiempo

- pk_date
- date_year
- date_month
- date_day
- date_date
- temporada

DIM Equipos.dim [Diseño]* -> **Cube_SS.cube [Diseño]***

Estructura de dimensión | Relaciones de atributo | Traducciones | Explorador

Atributos | Jerarquías | Vista del origen de datos

Atributos

- DIM Equipos
 - Conferencia
 - División
 - Liga
 - Nombre
 - Nombre Equipo
 - Pk Conferencia
 - Pk División
 - Pk Equipo
 - Pk Estado EEU

Jerarquías

Para crear un nuevo atributo, arrastra un atributo hasta aquí.

Vista del origen de datos

DIM_Equipos

- pk_equipo
- id_estado_EEU
- id_division
- liga
- codigo_equipo
- nombre_equipo
- club
- pabellon
- fundado
- patrocinio

DIM_Conferencias

- pk_conferencia
- codigo_conferencia
- conferencia

DIM_Divisiones

- pk_division
- codigo_division
- division
- id_conferencia

DIM_Estados_EEU

- pk_estado_EEU
- codigo
- nombre
- nombre_oficial
- superficie
- poblacion
- capital
- densidadPoblacion

CUBE_FT

Para este cubo, en primer lugar, vamos a añadir la dimensión degenerada DIM_PeriodosJuego. Esta dimensión, no tiene una tabla física en la base de datos, pero sí será una dimensión lógica en nuestros cubos. Para crearla vamos al asistente de Dimensiones y en la primera pantalla indicamos que la crearemos a partir de tablas existentes.

Seleccionamos la vista View_FT y la tabla de hechos FACT_FREE_THROW, que es la tabla asociada a la nueva dimensión e indicamos la columna a partir de la cual la crearemos. A continuación seleccionaremos los atributos.

Asistente para dimensiones

Seleccionar los atributos de la dimensión

Especifique los atributos de dimensión y seleccione Habilitar exploración para mostrarlos como jerarquías.



Atributos disponibles:

Nombre del atributo	Habilitar exploración	Tipo de atributo
Periodo Juego	<input checked="" type="checkbox"/>	Normal
Resultado Tiro Libre	<input type="checkbox"/>	Normal
Pk Minuto Segundo	<input checked="" type="checkbox"/>	Normal
Minuto	<input type="checkbox"/>	Normal
Segundo	<input type="checkbox"/>	Normal
Minuto Segundo	<input type="checkbox"/>	Normal
Pk Date	<input checked="" type="checkbox"/>	Normal
Date Year	<input type="checkbox"/>	Normal
Date Month	<input type="checkbox"/>	Normal
Date Day	<input type="checkbox"/>	Normal
Date Date	<input type="checkbox"/>	Normal

< Back Next > Finish >> Cancel

Por ultimo vemos un resumen de la nueva dimensión y le asignamos nombre, en este caso, DIM_PeriodosJuego.

Asistente para dimensiones

Finalización del asistente

Escriba un nombre para la nueva dimensión, compruebe la estructura de la dimensión y luego haga clic en Finalizar para guardar la dimensión.



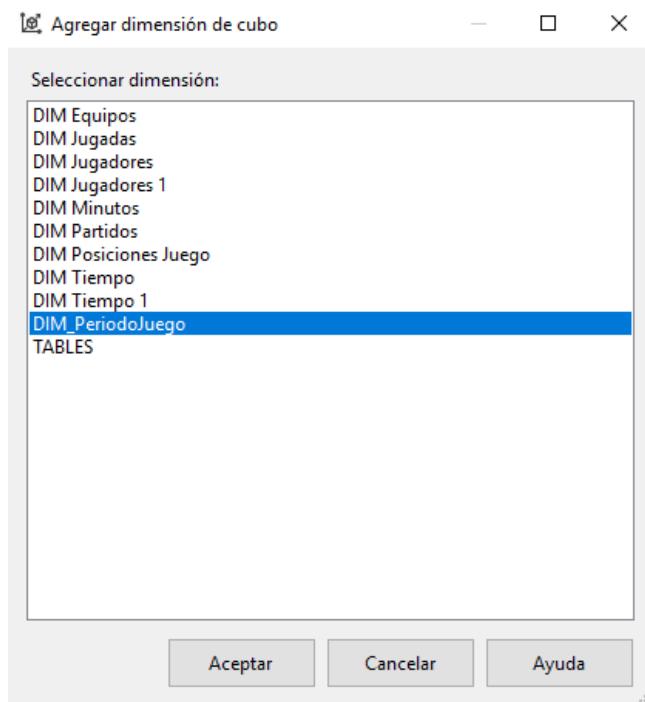
Nombre:

Vista previa:

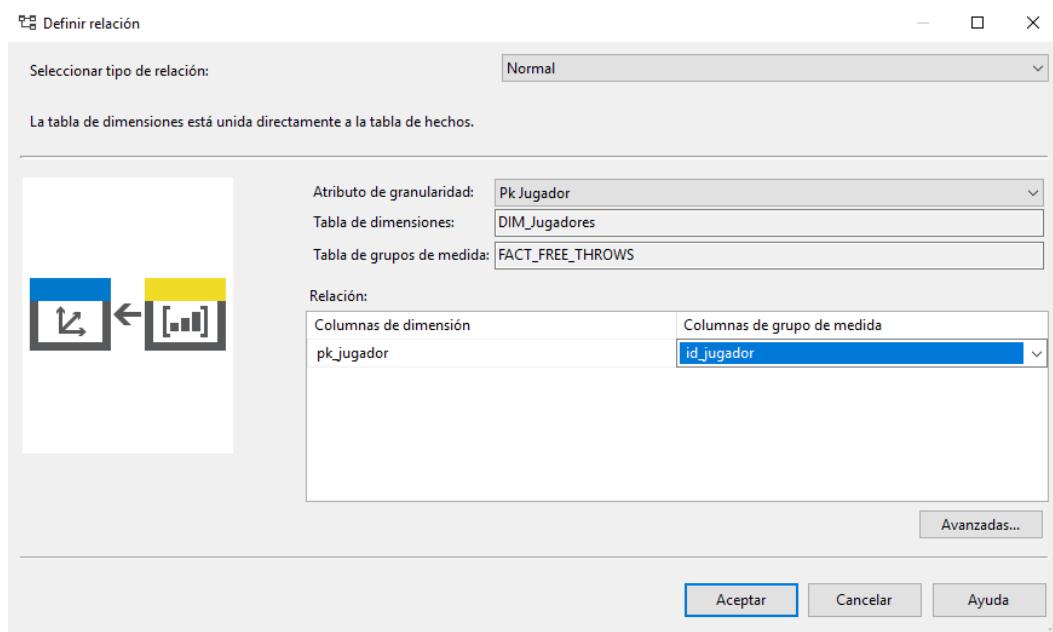
- DIM_PeriodoJuego
 - Atributos
 - Periodo Juego
 - Pk Minuto Segundo
 - Pk Date
 - Pk Jugada
 - Pk Partido
 - Pk Jugador

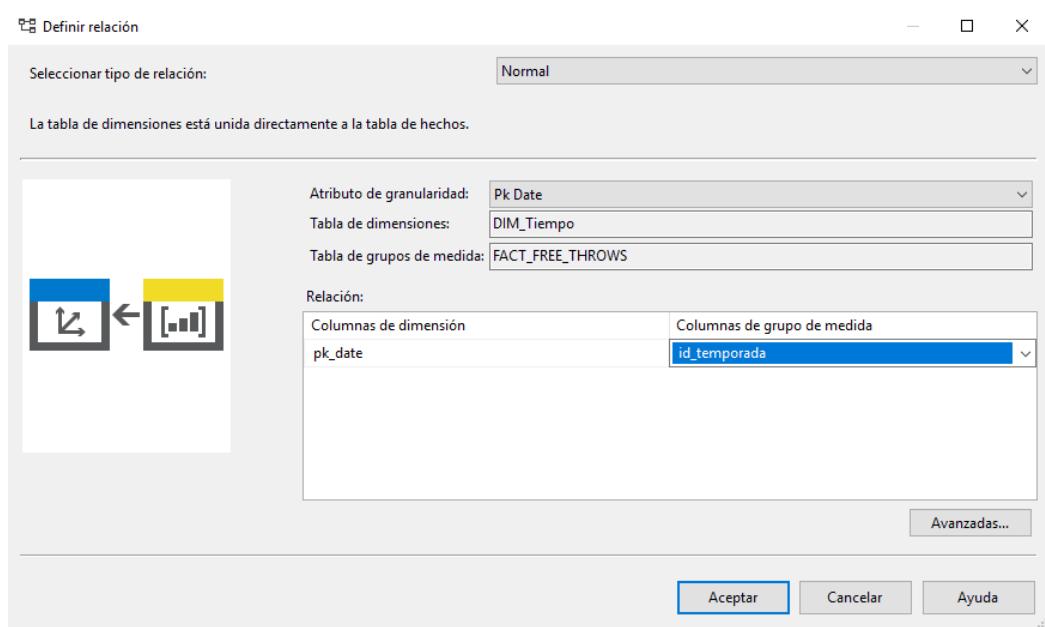
< Back Next > Finalizar Cancel

El ultimo paso es agregar la nueva dimension a nuestro cubo_Cube_FT.



Al crear este cubo se han creado las dimensiones duplicadas DIM_Jugadores 1 y DIM_Tiempo 1. Debemos borrar estas dimensiones creadas por el sistema y seleccionar las que ya tenemos para este cubo. Agregamos las dimensiones que tienen que ser DIM_Jugadores y DIM_Tiempo y definimos las relaciones entre las dimensiones y las medidas del cubo.





Ahora vemos en el diseño de nuestro cubo que tenemos bien asignadas las dimensiones junto con sus claves para la medida de FACT_FREE_THROWS.

Por ultimo agregamos los atributos de las dimensiones que vamos a usar en este caso.

The screenshot shows two separate windows of the Analysis Studio interface, both titled "Cube_FT.cube [Diseño]".

DIM Partidos.dim [Diseño]

- Atributos:** Shows the dimension structure with a tree view under "DIM Partidos" containing "Partido", "Pk Partido", "Resultado", and "Temporada".
- Jerarquías:** A placeholder area with the instruction: "Para crear un nuevo atributo, arrastre un atributo hasta aquí."
- Vista del origen de datos:** Shows the table "DIM_Partidos" with columns: pk_partido, partido, playoffs, temporada, and resultado.

DIM Jugadas.dim [Diseño]

- Atributos:** Shows the dimension structure with a tree view under "DIM Jugadas" containing "Codigo Jugada", "Dec Jugada", and "Pk Jugada".
- Jerarquías:** A placeholder area with the instruction: "Para crear un nuevo atributo, arrastre un atributo hasta aquí."
- Vista del origen de datos:** Shows the table "DIM_Jugadas" with columns: pk_jugada, codigo_jugada, and dec_jugada.

f) Implementación de la solución

Para poder hacer explotaciones con los datos en los cubos que hemos definido, tenemos que implementar la solución, que consistirá en la compilación de todos los componentes. Para ello iremos al menú principal, elegiremos de la opción Compilar > Implementación.

Después de la compilación, nos aparece una pantalla con el resultado del despliegue de la solución. Vamos a corregir el mensaje:

Database [DB_vnoronoz] : La base de datos no tiene dimensión de tiempo. Considere la posibilidad de crearla.

Para que la BD tenga dimensión tiempo y la reconozca el programa, vamos a nuestra dimensión DIM_Tiempo y especificamos que su tipo sea Time. Y en su atributo "Date Year" indicamos que es de tipo Years.

The screenshot shows two windows side-by-side:

- Left Window (DIM Tiempo Dimension):**
 - WriteEnabled: False
 - Type:** Time (highlighted with a red box)
- Right Window (Date Year DimensionAttribute):**
 - TokenizationBehavior: TokenizationNone
 - UserEditFlag: 0
 - VertipaqCompressionHint: VertipaqAutomatic
 - Name:** Date Year
 - Type:** Years (highlighted with a red box)
 - Usage: Regular

Ahora, cuando procesemos la solución no nos dará el mensaje de que no hay dimensión tiempo asociada a la BD. Resultado de la implementación y procesamiento del cubo CUBE_SS:

The screenshot shows the "Progreso del proceso" (Processing Progress) window:

- Comando:**
 - Procesando Base de datos, 'DB_vnoronoz' completados.
 - Procesando Cubo, 'Cube_SS' completados.
 - Hora de inicio: 24/12/2020 18:32:13; Hora de finalización: 24/12/2020 18:32:15; Duración: 0:00:01
 - Procesando Grupo de medida, 'FACT SEASON STATS' completados.
 - Procesando Dimensión, 'DIM Equipos' completados.
 - Procesando Dimensión, 'DIM Jugadores' completados.
 - Procesando Dimensión, 'DIM Posiciones Juego' completados.
 - Procesando Dimensión, 'DIM Tiempo' completados.
 - Procesando Dimensión, 'TABLES' completados.
 - Hora de inicio: 24/12/2020 18:32:13; Hora de finalización: 24/12/2020 18:32:15; Duración: 0:00:01
 - Procesando Cubo, 'default' completados.
 - Hora de inicio: 24/12/2020 18:32:13; Hora de finalización: 24/12/2020 18:32:14; Duración: 0:00:01
 - Procesando Grupo de medida, 'TABLES' completados.
- Estado:** Proceso finalizado correctamente.

The screenshot shows the "Progreso de la implementación - Proy_Basket" (Implementation Progress - Proy_Basket) window:

- Comando:** (empty)
- Estado:** La implementación finalizó correctamente.

g) Explotación de la solución

Una vez diseñados y desplegados los cubos ya estamos en disposición de explotarlo para responder a las preguntas planteadas en el enunciado. La forma de acceder al visor OLAP es entrando en la pestaña 'Explorador' en cada cubo creado.

VISTA DEL EXPLORADOR DEL CUBE_SS CON EL QUE REALIZAREMOS EL ANÁLISIS DE ESTADÍSTICAS

The screenshot shows the Microsoft Analysis Services Explorer window for the 'Cube_SS' cube. The top menu bar includes options like Estructura de cubo, Uso de dimensiones, Cálculos, KPI, Acciones, Particiones, Agregaciones, Perspectivas, Traducciones, and Explorador. The ribbon below has tabs for Editar como texto, Importar..., MDX, and other tools. The main workspace is divided into several panes:

- Left pane:** Shows the cube structure with 'Cube_SS' expanded, revealing 'Metadatos', 'Buscar modelo', 'Grupo de medida:', and a list of dimensions: 'Todos', 'Cube_SS' (Measures, KPI), 'DIM Equipos', 'DIM Jugadores', 'DIM Posiciones Juego', and 'DIM Tiempo'. Below this is a 'Miembros calculados' pane.
- Top right pane:** A grid for defining dimensions with columns: Dimensión, Jerarquía, Operador, Expresión de filtro, and Parámetros. A placeholder row says '<Seleccionar dimensión>'.
- Bottom center pane:** A large area with the placeholder text 'Arrastre niveles o medidas aquí para agregarlos a la consulta.'

Listado de puntos anotados por conferencia, división y posición de juego durante la temporada 2015-2016 en la NBA.

Para ver el informe, abriremos el cubo **Cube_SS** que hemos diseñado y nos posicionaremos en la pestaña Explorador, que es la que nos permite crear los informes utilizando las medidas y dimensiones disponibles.

En la zona de filtros arrastraremos la Temporada de DIM_Tiempo, seleccionando el valor “2015 - 2016” y la Liga de DIM_Equipos, filtrando por el valor “NBA”. Arrastramos la posición de juego de DIM_PosicionesJuego y la Conferencia y División de DIM_Equipos al área de trabajo. Finalmente seleccionamos la medida PTS:

Dimensión	Jerarquía	Operador	Expresión de filtro
DIM Tiempo	Temporal	Igual	{ 2015 - 2016 }
DIM Equipos	Liga	Igual	{ NBA }
<Seleccionar dimensión>			

Conferencia	Division	Desc Posicion ES	PTS
CONFERENCIA ESTE	ATLÁNTICO	ALA-PIVOT	4207
CONFERENCIA ESTE	ATLÁNTICO	ALERÓ	3917
CONFERENCIA ESTE	ATLÁNTICO	BASE	5649
CONFERENCIA ESTE	ATLÁNTICO	ESCOLTA	7136
CONFERENCIA ESTE	ATLÁNTICO	PIVOT	4170
CONFERENCIA ESTE	CENTRAL	ALA-PIVOT	6919
CONFERENCIA ESTE	CENTRAL	ALERÓ	8987
CONFERENCIA ESTE	CENTRAL	BASE	9609
CONFERENCIA ESTE	CENTRAL	ESCOLTA	8320
CONFERENCIA ESTE	CENTRAL	PIVOT	7915
CONFERENCIA ESTE	CENTRAL	Unknown	5666
CONFERENCIA ESTE	SURESTE	ALA-PIVOT	6292
CONFERENCIA ESTE	SURESTE	ALERÓ	3866
CONFERENCIA ESTE	SURESTE	BASE	4731
CONFERENCIA ESTE	SURESTE	ESCOLTA	5105
CONFERENCIA ESTE	SURESTE	PIVOT	5012
CONFERENCIA ESTE	SURESTE	Unknown	2805
CONFERENCIA OESTE	NOROESTE	ALA-PIVOT	4460
CONFERENCIA OESTE	NOROESTE	ALERÓ	7807
CONFERENCIA OESTE	NOROESTE	BASE	6799
CONFERENCIA OESTE	NOROESTE	ESCOLTA	8827
CONFERENCIA OESTE	NOROESTE	PIVOT	6520
CONFERENCIA OESTE	NOROESTE	Unknown	2917
CONFERENCIA OESTE	PACÍFICO	ALA-PIVOT	3376
CONFERENCIA OESTE	PACÍFICO	ALERÓ	5001
CONFERENCIA OESTE	PACÍFICO	BASE	5075
CONFERENCIA OESTE	PACÍFICO	ESCOLTA	7060
CONFERENCIA OESTE	PACÍFICO	PIVOT	4779
CONFERENCIA OESTE	SUROESTE	ALA-PIVOT	5387
CONFERENCIA OESTE	SUROESTE	ALERÓ	5710
CONFERENCIA OESTE	SUROESTE	BASE	5749
CONFERENCIA OESTE	SUROESTE	ESCOLTA	4867
CONFERENCIA OESTE	SUROESTE	PIVOT	3538
CONFERENCIA OESTE	SUROESTE	Unknown	2810

Ranking por eficiencia (TOP 10) de las jugadoras de la WNBA durante la temporada 2012-2013.

Para ver el informe, abriremos el cubo **Cube_SS** que hemos diseñado y realizamos la siguiente consulta MDX cambiando el modo de explotación con el botón 'Design Mode'.

```
SELECT TOPCOUNT( [DIM Jugadores] . [Nombre] . [Nombre] .ALLMEMBERS, 10, [Measures] . [EFF] ) ON ROWS,
[Measures] . [EFF] ON COLUMNS
FROM [Cube_SS]
WHERE ( [DIM Jugadores] . [Liga] .&[WNBA], [DIM Tiempo] . [Temporada] .&[2012 - 2013] )
```

```
SELECT TOPCOUNT([DIM Jugadores].[Nombre].[Nombre].ALLMEMBERS, 10,[Measures].[EFF]) ON ROWS,
[Measures].[EFF] ON COLUMNS
FROM [Cube_SS]
WHERE ([DIM Jugadores].[Liga].&[WNBA], [DIM Tiempo].[Temporada].&[2012 - 2013] )
```

Nombre	EFF
SYLVIA FOWLES	745
MAYA MOORE	737
CANDACE PARKER	713
ERIKA DE SOUZA	687
NNEKA OGWUMIKE	645
DIANA TAURASI	633
ANGEL MCCOUGHTRY	620
TAMIKA CATCHINGS	584
LINDSAY WHALEN	565
ELENA DELLE DONNE	554

Ranking (TOP 10) de jugadores de la NBA con mejor porcentaje de aciertos en tiros de 3 puntos, durante la temporada 2011-2012.

Para ver el informe, abriremos el cubo **Cube_SS** que hemos diseñado y realizamos la siguiente consulta MDX:

```
WITH MEMBER [Measures].[Perc3] AS
([3PM]/[3PA])
SELECT TOPCOUNT([DIM Jugadores].[Nombre].[Nombre].ALLMEMBERS, 10,[Measures].[Perc3]) ON ROWS,
[Measures].[Perc3] ON COLUMNS
FROM [Cube_SS]
WHERE ( [DIM Jugadores].[Liga].&[NBA], [DIM Tiempo].[Temporada].&[2011 - 2012] )
```

```
WITH MEMBER [Measures].[Perc3] AS
([3PM]/[3PA])
SELECT TOPCOUNT([DIM Jugadores].[Nombre].[Nombre].ALLMEMBERS, 10,[Measures].[Perc3]) ON ROWS,
[Measures].[Perc3] ON COLUMNS
FROM [Cube_SS]
WHERE ( [DIM Jugadores].[Liga].&[NBA], [DIM Tiempo].[Temporada].&[2011 - 2012] )
```

Nombre	Perc3
D.J. WHITE	1
JEFF FOSTER	1
JOHAN PETRO	1
SHAUN LIVINGSTON	0,6666666666666667
MIKE JAMES	0,6
D.J. KENNEDY	0,5
TREVOR BOOKER	0,5
STEVE NOVAK	0,471631205673759
BRAD MILLER	0,4666666666666667
ERSAN ILYASOVA	0,455357142857143

Listado de equipos de la NBA, ordenado por el número rebotes defensivos realizados en la temporada 2014-2015 (ordenación descendente por DREB).

Para ver el informe, abriremos el cubo **Cube_SS** que hemos diseñado y realizamos la siguiente consulta MDX:

```
SELECT ORDER ([DIM Equipos].[Nombre Equipo].[Nombre Equipo].ALLMEMBERS, [Measures].[DREB], DESC) ON ROWS ,
[Measures].[DREB] ON COLUMNS
FROM [Cube_SS]
WHERE ( [DIM Tiempo].[Temporada].&[2014 - 2015], [DIM Equipos].[Liga].&[NBA] )
```

```
SELECT ORDER ([DIM Equipos].[Nombre Equipo].[Nombre Equipo].ALLMEMBERS, [Measures].[DREB], DESC) ON ROWS ,
[Measures].[DREB] ON COLUMNS
FROM [Cube_SS]
WHERE ( [DIM Tiempo].[Temporada].&[2014 - 2015], [DIM Equipos].[Liga].&[NBA] )
```

Nombre Equipo	DREB
CHICAGO BULLS	3779
INDIANA PACERS	3593
ATLANTA HAWKS	3371
MINNESOTA TIMBERWOLVES	3298
PORTLAND TRAIL BLAZERS	2881
OKLAHOMA CITY THUNDER	2844
SACRAMENTO KINGS	2728
LOS ANGELES CLIPPERS	2711
BOSTON CELTICS	2685
DENVER NUGGETS	2653
LOS ANGELES LAKERS	2647
MEMPHIS GRIZZLIES	2634
DETROIT PISTONS	2630
HOUSTON ROCKETS	2624
CLEVELAND CAVALIERS	2612
DALLAS MAVERICKS	2608
ORLANDO MAGIC	2607
MILWAUKEE BUCKS	2574

Evolución temporal desde la temporada 2006-2007 hasta la 2015-2016 del total de tiros libres realizados, acertados, fallados y tanto por ciento de acierto por brazo dominante del jugador.

Para ver el informe, abriremos el cubo **Cube_SS** que hemos diseñado y realizamos la siguiente consulta MDX:

Dimensión	Jerarquía	Operador	Expresión de filtro	Parán
DIM Tiempo	Temporal	Intervalo (Inclusivo)	2006 - 2007 : 2015 - 2016	<input type="checkbox"/>
DIM Jugadores	Shoots	Igual	{LEFT, RIGHT}	<input type="checkbox"/>
<Seleccionar dimensión>				<input type="checkbox"/>

Shoots	Temporada	FTM	FTA	FT
LEFT	2006 - 2007	3883	5014	30,228
LEFT	2007 - 2008	3910	5050	27,777
LEFT	2008 - 2009	3526	4772	24,039
LEFT	2009 - 2010	3881	5158	20,327
LEFT	2010 - 2011	4429	6077	32,29
LEFT	2011 - 2012	3376	4527	33,356
LEFT	2012 - 2013	4538	6215	34,245
LEFT	2013 - 2014	4691	6548	30,285
LEFT	2014 - 2015	5031	7063	44,682
LEFT	2015 - 2016	4664	6345	38,051
RIGHT	2006 - 2007	46416	61679	327,249
RIGHT	2007 - 2008	45673	60899	371,957
RIGHT	2008 - 2009	47545	61400	378,47
RIGHT	2009 - 2010	45926	60476	374,048
RIGHT	2010 - 2011	47454	61770	393,09
RIGHT	2011 - 2012	31425	41691	339,978

CONCLUSIONES

- En el presente documento se documenta la segunda parte del caso práctico "*Almacén de datos para el análisis de estadísticas deportivas de las ligas de baloncesto WNBA y NBA*" consistente en la la carga y explotación de datos.
- Se han identificado los procesos de extracción, transformación y carga de datos (ETL) necesarios para la carga efectiva de los orígenes de datos en el almacén.
- Se han diseñado y desarrollado los procesos ETL mediante el software Pentaho Data Integration (PDI).
- Los procesos ETL han sido implementados mediante trabajos-jobs y transformaciones. Éstos han sido divididos en dos bloques: Bloque IN para la carga desde los orígenes a la staging area y Bloque TR para la carga desde la staging area al almacén. Este bloque a su vez ha sido diferenciado en la carga de dimensiones y en carga de hechos.
- Se ha diseñado un modelo OLAP para el análisis de la información que da respuesta a los requerimientos establecidos de entrada:
 - Realizar un seguimiento por temporadas de las estadísticas de juego de los jugadores y jugadoras, para evaluar su evolución en el tiempo y analizar su rendimiento, según el equipo y posición en la que juegan.
 - Estudiar detalladamente los tiros libres realizados durante las temporadas y analizar su impacto sobre el resultado final de los partidos.
- Se presenta una solución funcional que podría ser mejorada en posteriores revisiones. Algunas de las cuestiones que sería de interés optimizar son:
 - Revisar en más detalle los valores en los orígenes de datos para realizar un proceso ETL más optimizado.
 - Pensar cómo mejorar tiempos de ejecución de los procesos.
 - Mejorar soluciones con respecto a valores desconocidos investigando, recopilando información extra y completando los orígenes de datos para que la relación entre claves foráneas sea más exacta.
 - Mejorar la carga con respecto a las posiciones de juego, habilitando más combinaciones.

REFERENCIAS BIBLIOGRÁFICAS

- Abelló Gamazo, A. , Curto Díaz, J. , Samos Jiménez, J. , Vidal Gil, J. y Díaz Arias, D.** (2020). *La construcción de la factoría de información corporativa*. PID_00270641. FUOC
- Abelló Gamazo, A. , Llorach Rius, C. , Ruiz Marqués, V.** (2020). *Diseño multidimensional y explotación de datos*. PID_00270639. FUOC
- J. Conesa, J. Curto.** *El conocimiento imprescindible*. Disponible en:
<http://reader.digitalbooks.pro/content/preview/books/43005/book/OEBPS/chapter02.xhtml>
- Díaz, D.** (2020). Caso práctico: *Almacén de datos para el análisis de estadísticas deportivas de las ligas de baloncesto WNBA y NBA*. PID_00273959. UOC
- Díaz, D.** (2020). Caso práctico: *Almacén de datos para el análisis de estadísticas deportivas de las ligas de baloncesto WNBA y NBA: PRA1- Análisis y diseño del data warehouse*. PID_00277332. UOC
- Díaz, D.** (2020). Caso práctico: *Almacén de datos para el análisis de estadísticas deportivas de las ligas de baloncesto WNBA y NBA: PRA2- Carga y explotación de datos*. PID_00277329. UOC
- Sevilla Marchena, N.** (2019-2020). Caso Práctico: *Sistema Integrado de Egresados*. UOC
- Kimball, R.** (2013). *The Data Warehouse Toolkit* (3.^a ed.). Nueva York: John Wiley & Sons Inc.
- Inmon W. H.** (1996). *Building the Data Warehouse* (2.^a ed.). John Wiley & Sons Inc.
- NBA Advanced Stats. Disponible en: <https://www.nba.com/stats/>
- WNBA Advanced Stats. Disponible en: <https://www.wnba.com/stats/>
- NBA Players stats since 1950. Disponible en: <https://www.kaggle.com/drgilermo/nba-players-stats>
- NBA Free Throws. Disponible en: <https://www.kaggle.com/sebastianmantey/nba-free-throws>
- NBA Players data. Disponible en: <https://data.world/search?context=community&page=5&q=NBA&type=all>

ANEXOS

Script creación tablas intermedias (STG)

```
-----
-- Create table STG_WNBA_Players_list
--

CREATE TABLE dbo.STG_WNBA_Players_list (
    [player]      [NVARCHAR](255) NULL,
    [active]       [NVARCHAR](255) NULL
);

-----
-- Create table STG_NBA_Players_list
--

CREATE TABLE dbo.STG_NBA_Players_list (
    [player]      [NVARCHAR](255) NULL,
    [active]       [NVARCHAR](255) NULL
);

-----
-- Create table STG_NBA_Players_data
--

CREATE TABLE dbo.STG_NBA_Players_data (
    [id_]          [NVARCHAR](50) NULL,
    [birthDate]    [DATE] NULL,
    [birthPlace]   [NVARCHAR](255) NULL,
    [career_AST]   [FLOAT] NULL,
    [career_FG%]   [FLOAT] NULL,
    [career_FG3%]  [FLOAT] NULL,
    [career_FT%]   [FLOAT] NULL,
    [career_G]     [FLOAT] NULL,
    [career_PER]   [FLOAT] NULL,
    [career_PTS]   [FLOAT] NULL,
    [career_TRB]   [FLOAT] NULL,
    [career_WS]    [FLOAT] NULL,
    [career_eFG%]  [FLOAT] NULL,
    [college]      [NVARCHAR](255) NULL,
    [height]       [NVARCHAR](50) NULL,
    [highSchool]   [NVARCHAR](255) NULL,
    [name]         [NVARCHAR](255) NULL,
    [position]     [NVARCHAR](100) NULL,
    [shoots]        [NVARCHAR](50) NULL,
    [weight]        [NVARCHAR](50) NULL
);

-----
-- Create table STG_Team_Codes
--

CREATE TABLE dbo.STG_Team_Codes (
    [league]       [NVARCHAR] (50) NULL,
    [team]         [NVARCHAR] (255) NULL,
    [code]         [NVARCHAR] (50) NULL
);

-----
-- Create table STG_EEUU
--

CREATE TABLE dbo.STG_EEUU (
    [estado]        [NVARCHAR](255) NULL,
    [nombre_oficial] [NVARCHAR](100) NULL,
    [superficie]    [FLOAT] NULL,
    [abreviatura]   [NVARCHAR](50) NULL,
    [poblacion_2010] [BIGINT] NULL,
    [densidad_pobl] [FLOAT] NULL,
    [capital]       [NVARCHAR](100) NULL
);

-----
-- Create table STG_NBA_free_throws
--

CREATE TABLE dbo.STG_NBA_free_throws (
    [end_result]   [NVARCHAR](50) NULL,
    [game]          [NVARCHAR](50) NULL,
    [game_id]       [FLOAT] NULL,
    [period]        [FLOAT] NULL,
    [play]          [NVARCHAR](255) NULL,
    [player]        [NVARCHAR](255) NULL,
    [playoffs]      [NVARCHAR](50) NULL,
    [score]          [NVARCHAR](50) NULL,
    [season]        [NVARCHAR](50) NULL,
    [shot_made]    [INT] NULL,
    [time_]         [NVARCHAR](50) NULL
);

-----
-- Create table STG_WNBA_Teams
--


```

```

CREATE TABLE dbo.STG_WNBA_Teams (
    [id#]          [INT] NULL,
    [conferencia]  [NVARCHAR](100) NULL,
    [equipo]        [NVARCHAR](255) NULL,
    [ciudad]        [NVARCHAR](255) NULL,
    [estado]        [NVARCHAR](255) NULL
);

-- Create table STG_NBA_Teams
-- 

CREATE TABLE dbo.STG_NBA_Teams (
    [conferencia]  [NVARCHAR](100) NULL,
    [division]     [NVARCHAR](100) NULL,
    [equipo]        [NVARCHAR](255) NULL,
    [ciudad]        [NVARCHAR](255) NULL,
    [estado]        [NVARCHAR](255) NULL,
    [pabellon]      [NVARCHAR](255) NULL,
    [fundado]       [INT] NULL,
    [patrocinio]   [NVARCHAR](255) NULL
);

-- Create table STG_NBA_Seasons_Stats_1950_2017
-- 

CREATE TABLE dbo.STG_NBA_Seasons_Stats_1950_2017 (
    [id#]          [INT] NULL,
    [year_]         [INT] NULL,
    [player]        [NVARCHAR](255) NULL,
    [pos]           [NVARCHAR](100) NULL,
    [age]           [INT] NULL,
    [Tm]            [NVARCHAR](255) NULL,
    [G]             [FLOAT] NULL,
    [GS]            [FLOAT] NULL,
    [MP]            [FLOAT] NULL,
    [PER]           [FLOAT] NULL,
    [TS%]           [FLOAT] NULL,
    [_3PAr]         [FLOAT] NULL,
    [FTr]           [FLOAT] NULL,
    [ORB%]          [FLOAT] NULL,
    [DRB%]          [FLOAT] NULL,
    [TRB%]          [FLOAT] NULL,
    [AST%]          [FLOAT] NULL,
    [STL%]          [FLOAT] NULL,
    [BLK%]          [FLOAT] NULL,
    [TOV%]          [FLOAT] NULL,
    [USG%]          [FLOAT] NULL,
    [OWS]           [FLOAT] NULL,
    [DWS]           [FLOAT] NULL,
    [WS]             [FLOAT] NULL,
    [WSx48]          [FLOAT] NULL,
    [OBPM]          [FLOAT] NULL,
    [DBPM]          [FLOAT] NULL,
    [BPM]           [FLOAT] NULL,
    [VORP]          [FLOAT] NULL,
    [FG]             [FLOAT] NULL,
    [FGA]            [FLOAT] NULL,
    [FG%]            [FLOAT] NULL,
    [_3P]            [FLOAT] NULL,
    [_3PA]           [FLOAT] NULL,
    [_3P%]           [FLOAT] NULL,
    [_2P]            [FLOAT] NULL,
    [_2PA]           [FLOAT] NULL,
    [_2P%]           [FLOAT] NULL,
    [eFG%]          [FLOAT] NULL,
    [FT]             [FLOAT] NULL,
    [FTA]            [FLOAT] NULL,
    [FT%]            [FLOAT] NULL,
    [ORB]            [FLOAT] NULL,
    [DRB]            [FLOAT] NULL,
    [TRB]            [FLOAT] NULL,
    [AST]            [FLOAT] NULL,
    [STL]            [FLOAT] NULL,
    [BLK]            [FLOAT] NULL,
    [TOV]            [FLOAT] NULL,
    [PF]             [FLOAT] NULL,
    [PTS]            [FLOAT] NULL
);

-- Create table STG_WNBA_Seasons_Stats_2005_2017
-- 

CREATE TABLE dbo.STG_WNBA_Seasons_Stats_2005_2017 (
    [id_]          [INT] NULL,
    [data_]        [NVARCHAR](255) NULL,
    [type_]        [NVARCHAR](255) NULL,
    [stat]          [NVARCHAR](255) NULL
);

```

Script creación tablas modelo (DIMENSIONES)

```
-----
-- DIMENSIONES
-- Create table DIM_Conferencias
-- Create table DIM_Divisiones
-- Create table DIM_Equipos
-- Create table DIM_Estados_EEUU
-- Create table DIM_Jugadas
-- Create table DIM_Jugadores
-----
```

```
CREATE TABLE dbo.DIM_Conferencias (
    [pk_conferencia]          [INT] NOT NULL,
    [codigo_conferencia]       [NVARCHAR](50) NULL,
    [conferencia]              [NVARCHAR](100) NULL,
    CONSTRAINT PK_DIM_Conferencias PRIMARY KEY (pk_conferencia)
);
```

```
-- Create table DIM_Divisiones
-- Create table DIM_Equipos
-- Create table DIM_Estados_EEUU
-- Create table DIM_Jugadas
-- Create table DIM_Jugadores
-----
```

```
CREATE TABLE dbo.DIM_Divisiones (
    [pk_division]          [INT] NOT NULL,
    [codigo_division]       [NVARCHAR](50) NULL,
    [division]              [NVARCHAR](100) NULL,
    [id_conferencia]        [INT] NULL,
    CONSTRAINT PK_DIM_Divisiones PRIMARY KEY (pk_division),
    CONSTRAINT FK_DIM_Divisiones_Conf FOREIGN KEY (id_conferencia) REFERENCES dbo.DIM_Conferencias (pk_conferencia)
);
```

```
-- Create table DIM_Equipos
-- Create table DIM_Estados_EEUU
-- Create table DIM_Jugadas
-- Create table DIM_Jugadores
-----
```

```
CREATE TABLE dbo.DIM_Equipos (
    [pk_equipo]          [INT] NOT NULL,
    [id_estado_EEU]       [INT] NULL,
    [id_division]         [INT] NULL,
    [liga]                [NVARCHAR](50) NULL,
    [codigo_equipo]       [NVARCHAR](50) NULL,
    [nombre_equipo]       [NVARCHAR](255) NULL,
    [ciudad]              [NVARCHAR](255) NULL,
    [pabellon]             [NVARCHAR](255) NULL,
    [fundado]              [INT] NULL,
    [patrocinio]           [NVARCHAR](255) NULL,
    CONSTRAINT PK_DIM_Equipos PRIMARY KEY (pk_equipo),
    CONSTRAINT FK_DIM_Equipos_Est FOREIGN KEY (id_estado_EEU) REFERENCES dbo.DIM_Estados_EEUU (pk_estado_EEU),
    CONSTRAINT FK_DIM_Equipos_Div FOREIGN KEY (id_division) REFERENCES dbo.DIM_Divisiones (pk_division)
);
```

```
-- Create table DIM_Estados_EEUU
-- Create table DIM_Jugadas
-- Create table DIM_Jugadores
-----
```

```
CREATE TABLE dbo.DIM_Estados_EEUU (
    [pk_estado_EEU]       [INT] NOT NULL,
    [codigo]               [NVARCHAR](50) NULL,
    [nombre]               [NVARCHAR](255) NULL,
    [nombre_oficial]       [NVARCHAR](100) NULL,
    [superficie]            [FLOAT] NULL,
    [poblacion]             [BIGINT] NULL,
    [capital]               [NVARCHAR](100) NULL,
    [densidadPoblacion]     [FLOAT] NULL,
    CONSTRAINT PK_DIM_Estados_EEUU PRIMARY KEY (pk_estado_EEU)
);
```

```
-- Create table DIM_Jugadas
-- Create table DIM_Jugadores
-----
```

```
CREATE TABLE dbo.DIM_Jugadas (
    [pk_jugada]          [INT] NOT NULL,
    [codigo_jugada]       [NVARCHAR](50) NULL,
    [dec_jugada]           [NVARCHAR](255) NULL,
    CONSTRAINT PK_DIM_Jugadas PRIMARY KEY (pk_jugada)
);
```

```
-- Create table DIM_Jugadores
-----
```

```
CREATE TABLE dbo.DIM_Jugadores (
    [pk_jugador]          [INT] NOT NULL,
    [liga]                 [NVARCHAR](50) NULL,
    [cod_jugador]          [NVARCHAR](50) NULL,
    [nombre]               [NVARCHAR](255) NULL,
    [posicion_juego]       [NVARCHAR](100) NULL,
    [sexo]                  [NVARCHAR](50) NULL,
    [activo]                [NVARCHAR](50) NULL,
    [altura]                [NVARCHAR](50) NULL,
    [peso]                  [NVARCHAR](50) NULL,
```

```

[shoots]           [NVARCHAR](50) NULL,
[universidad]      [NVARCHAR](255) NULL,
[fecha_nacimiento] [DATE] NULL,
[ciudad_nacimiento] [NVARCHAR](255) NULL,
[career_AST]        [FLOAT] NULL,
[career_FG%]        [FLOAT] NULL,
[career_FG3%]       [FLOAT] NULL,
[career_FT%]        [FLOAT] NULL,
[career_G]          [FLOAT] NULL,
[career_PER]         [FLOAT] NULL,
[career_PTS]         [FLOAT] NULL,
[career_TRB]         [FLOAT] NULL,
[career_WS]          [FLOAT] NULL,
[career_eFG%]        [FLOAT] NULL,
CONSTRAINT PK_DIM_Jugadores PRIMARY KEY (pk_jugador)
);

-- Create table DIM_Minutos
--

CREATE TABLE dbo.DIM_Minutos (
    [pk_minutoSegundo] [INT] NOT NULL,
    [minuto]            [INT] NULL,
    [segundo]           [INT] NULL,
    [minutoSegundo]     [NVARCHAR](50) NULL,
    CONSTRAINT PK_DIM_Minutos PRIMARY KEY (pk_minutoSegundo)
);

-- Create table DIM_Partidos
--

CREATE TABLE dbo.DIM_Partidos (
    [pk_partido]        [INT] NOT NULL,
    [partido]           [NVARCHAR](50) NULL,
    [playoffs]          [NVARCHAR](50) NULL,
    [temporada]         [NVARCHAR](50) NULL,
    [resultado]         [NVARCHAR](50) NULL,
    CONSTRAINT PK_DIM_Partidos PRIMARY KEY (pk_partido)
);

-- Create table DIM_PosicionesJuego
--

CREATE TABLE dbo.DIM_PosicionesJuego (
    [pk_posicion_juego] [INT] NOT NULL,
    [codigo_posicion]   [NVARCHAR](10) NULL,
    [desc_posicion_EN]  [NVARCHAR](50) NULL,
    [desc_posicion_ES]  [NVARCHAR](50) NULL,
    [num_posicion]       [INT] NULL,
    CONSTRAINT PK_DIM_PosicionesJuego PRIMARY KEY (pk_posicion_juego)
);

-- Create table DIM_Tiempo
--

CREATE TABLE dbo.DIM_Tiempo (
    [pk_date]           [INT] NOT NULL,
    [date_year]          [INT] NULL,
    [date_month]         [INT] NULL DEFAULT 10,
    [date_day]           [INT] NULL DEFAULT 1,
    [date_date]          [DATE] NULL,
    [temporada]          [NVARCHAR](50) NULL,
    CONSTRAINT PK_DIM_Tiempo PRIMARY KEY (pk_date)
);

```

Script creación tablas modelo (HECHOS)

```
-----
-- HECHOS
--



-----



-- Create table FACT_FREE_THROWS
--



CREATE TABLE dbo.FACT_FREE_THROWS (
    [id_temporada] [INT] NULL,
    [id_partido] [INT] NULL,
    [id_jugador] [INT] NULL,
    [id_jugada] [INT] NULL,
    [id_minuto] [INT] NULL,
    [periodo_juego] [INT] NULL,
    [resultado_tiro_libre] [FLOAT] NULL,
    CONSTRAINT FK_FACT_FT_Temp FOREIGN KEY (id_temporada) REFERENCES dbo.DIM_Tiempo (pk_date),
    CONSTRAINT FK_FACT_FT_Part FOREIGN KEY (id_partido) REFERENCES dbo.DIM_Partidos (pk_partido),
    CONSTRAINT FK_FACT_FT_Jug FOREIGN KEY (id_jugador) REFERENCES dbo.DIM_Jugadores (pk_jugador),
    CONSTRAINT FK_FACT_FT_Play FOREIGN KEY (id_jugada) REFERENCES dbo.DIM_Jugadas (pk_jugada),
    CONSTRAINT FK_FACT_FT_Min FOREIGN KEY (id_minuto) REFERENCES dbo.DIM_Minutos (pk_minutoSegundo)
);
-----



-- Create table FACT_SEASON_STATS
--



CREATE TABLE dbo.FACT_SEASON_STATS (
    [id_season] [INT] NULL,
    [id_player] [INT] NULL,
    [id_team] [INT] NULL,
    [id_position] [INT] NULL,
    [league] [NCHAR](4) NULL,
    [GP] [FLOAT] NULL,
    [PTS] [FLOAT] NULL,
    [MIN] [FLOAT] NULL,
    [FGM] [FLOAT] NULL,
    [FGA] [FLOAT] NULL,
    [FG%] [FLOAT] NULL,
    [3PM] [FLOAT] NULL,
    [3PA] [FLOAT] NULL,
    [3P%] [FLOAT] NULL,
    [2PM] [FLOAT] NULL,
    [2PA] [FLOAT] NULL,
    [2P%] [FLOAT] NULL,
    [FTM] [FLOAT] NULL,
    [FTA] [FLOAT] NULL,
    [FT%] [FLOAT] NULL,
    [eFG%] [FLOAT] NULL,
    [OREB] [FLOAT] NULL,
    [DREB] [FLOAT] NULL,
    [REB] [FLOAT] NULL,
    [AST] [FLOAT] NULL,
    [TOV] [FLOAT] NULL,
    [STL] [FLOAT] NULL,
    [BLK] [FLOAT] NULL,
    [PF] [FLOAT] NULL,
    [EFF] [FLOAT] NULL,
    CONSTRAINT FK_FACT_SS_Sea FOREIGN KEY (id_season) REFERENCES dbo.DIM_Tiempo (pk_date),
    CONSTRAINT FK_FACT_SS_Jug FOREIGN KEY (id_player) REFERENCES dbo.DIM_Jugadores (pk_jugador),
    CONSTRAINT FK_FACT_SS_Equip FOREIGN KEY (id_team) REFERENCES dbo.DIM_Equipos (pk_equipo),
    CONSTRAINT FK_FACT_SS_Pos FOREIGN KEY (id_position) REFERENCES dbo.DIM_PosicionesJuego (pk_posicion_juego)
);
```

Vistas de las tablas pobladas con los datos en la BD de SQL SERVER**DIM_Conferencias**

	pk_conferencia	codigo_conferencia	conferencia
1	1	E	CONFERENCIA ESTE
2	2	O	CONFERENCIA OESTE
3	9999	DESC	DESCONOCIDO

DIM_Divisiones

	pk_division	codigo_division	division	id_conferencia
1	1	AT	ATLÁNTICO	1
2	2	CE	CENTRAL	1
3	3	NE	NOROESTE	2
4	4	PA	PACÍFICO	2
5	5	SE	SURESTE	1
6	6	SO	SUROESTE	2
7	9999	DESCONOCIDO	DESCONOCIDO	9999

DIM_Equipos

	pk_equipo	id_estado_EEU	id_division	liga	codigo_equipo	nombre_equipo	ciudad	pabellon	fundado	patrocinio
1	1	10	9999	WNBA	ADR	ATLANTA DREAM	ATLANTA	NULL	NULL	NULL
2	2	10	5	NBA	ATL	ATLANTA HAWKS	ATLANTA	STATE FARM ARENA	1946	SHARE
3	3	32	1	NBA	BN	BROOKLYN NETS	NUEVA YORK	BARCLAYS CENTER	1967	INFOR
4	4	21	1	NBA	BOS	BOSTON CELTICS	BOSTON	TD GARDEN	1946	GENER
5	5	33	5	NBA	CHA	CHARLOTTE HORNETS	CHARLOTTE	SPECTRUM CENTER	1988	LENDIN
6	6	13	2	NBA	CHI	CHICAGO BULLS	CHICAGO	UNITED CENTER	1966	ZENN
7	7	13	9999	WNBA	CHS	CHICAGO SKY	CHICAGO	NULL	NULL	NULL
8	8	35	2	NBA	CLE	CLEVELAND CAVALIERS	CLEVELAND	ROCKET MORTGAGE FIELDHOUSE	1970	GOODY
9	9	7	9999	WNBA	CON	CONNECTICUT SUN	UNCASVILLE	NULL	NULL	NULL

DIM_Estados_EEUU

	pk_estado_EEU	codigo	nombre	nombre_oficial	superficie	poblacion	capital	densidadPoblacion
1	1	AL	ALABAMA	STATE OF ALABAMA	135756	4779736	MONTGOMERY	35,21
2	2	AK	ALASKA	STATE OF ALASKA	1717854	710231	JUNEAU	0,41
3	3	AZ	ARIZONA	STATE OF ARIZONA	295254	6392017	PHOENIX	21,65
4	4	AR	ARKANSAS	STATE OF ARKANSAS	137732	2915918	LITTLE ROCK	21,17
5	5	CA	CALIFORNIA	STATE OF CALIFORNIA	423970	37253956	SACRAMENTO	87,87
6	6	CO	COLORADO	STATE OF COLORADO	269601	5029196	DENVER	18,65
7	7	CT	CONNECTICUT	STATE OF CONNECTICUT	14357	3574097	HARTFORD	248,94
8	8	DE	DELAWARE	STATE OF DELAWARE	6447	897934	DOVER	139,28
9	9	FL	FLORIDA	STATE OF FLORIDA	170304	18801310	TALLAHASSEE	110,4

DIM_Jugadas

	pk_jugada	codigo_jugada	dec_jugada
1	1	MADE	A.J. PRICE MAKES FREE THROW
2	2	MADE	A.J. PRICE MAKES FREE THROW 1 OF 1
3	3	MADE	A.J. PRICE MAKES FREE THROW 1 OF 2
4	4	MADE	A.J. PRICE MAKES FREE THROW 1 OF 3
5	5	MADE	A.J. PRICE MAKES FREE THROW 2 OF 2
6	6	MADE	A.J. PRICE MAKES FREE THROW 2 OF 3
7	7	MADE	A.J. PRICE MAKES FREE THROW 3 OF 3
8	8	MADE	A.J. PRICE MAKES TECHNICAL FREE THROW
9	9	MISSSED	A.J. PRICE MISSES FREE THROW 1 OF 1
10	10	MISSSED	A.J. PRICE MISSES FREE THROW 1 OF 2

DIM_Jugadores

		Results	Messages	pk_jugador	liga	cod_jugador	nombre	posicion_juego	sexo	activo	altura	peso	shoots	universidad	fecha_nacimiento
4	4	NBA	BRAMLAJ01	A.J. BRAMLETT	CENTER	M	NO	6-10	227LB	RIGHT	UNIVERSITY OF ARIZO...	1977-01-10			
5	5	NBA	ENGLIAJ01	A.J. ENGLISH	SHOOTING GUARD	M	NO	6-3	175LB	RIGHT	VIRGINIA UNION UNIVE...	1967-07-11			
6	6	NBA	GUYTOAJ01	A.J. GUYTON	POINT GUARD	M	NO	6-1	180LB	RIGHT	INDIANA UNIVERSITY	1978-02-12			
7	7	NBA	HAMMOAJ01	A.J. HAMMONS	CENTER	M	NULL	7-0	260LB	RIGHT	PURDUE UNIVERSITY	1992-08-27			
8	8	NBA	PRICEAJ01	A.J. PRICE	SHOOTING GUARD A...	M	NULL	6-2	181LB	RIGHT	UNIVERSITY OF CONN...	1986-10-07			
9	9	NBA	WYNDEAJ01	A.J. WYNDER	POINT GUARD	M	NO	6-2	180LB	RIGHT	UNIVERSITY OF MASSA...	1964-09-11			
10	10	NBA	HOLTAW01	A.W. HOLT	SMALL FORWARD	M	NULL	6-7	210LB	RIGHT	JACKSON STATE UNIV...	1946-08-26			
11	11	NBA	BROOKAA01	AARON BROOKS	POINT GUARD	M	NO	6-0	161LB	RIGHT	UNIVERSITY OF OREG...	1985-01-14			
12	12	NBA	GORDOAA01	AARON GORDON	POWER FORWARD ...	M	YES	6-9	220LB	RIGHT	UNIVERSITY OF ARIZO...	1995-09-16			

DIM_Minutos

	pk_minutoSegundo	minuto	segundo	minutoSegundo
1	1	0	0	0:00
2	2	0	1	0:01
3	3	0	2	0:02
4	4	0	3	0:03
5	5	0	4	0:04
6	6	0	5	0:05
7	7	0	6	0:06
8	8	0	7	0:07
9	9	0	8	0:08
10	10	0	9	0:09

DIM_Partidos

	pk_partido	partido	playoffs	temporada	resultado
1	1	ATL - BOS	REGULAR	2006 - 2007	82 - 76
2	2	ATL - CHA	REGULAR	2006 - 2007	85 - 104
3	3	ATL - CHI	REGULAR	2006 - 2007	86 - 94
4	4	ATL - CLE	REGULAR	2006 - 2007	104 - 95
5	5	ATL - DAL	REGULAR	2006 - 2007	87 - 110
6	6	ATL - DEN	REGULAR	2006 - 2007	98 - 96
7	7	ATL - DET	REGULAR	2006 - 2007	98 - 103
8	8	ATL - GS	REGULAR	2006 - 2007	106 - 105
9	9	ATL - HOU	REGULAR	2006 - 2007	68 - 94
10	10	ATL - IND	REGULAR	2006 - 2007	72 - 91

DIM_PosicionesJuego

	pk_posicion_juego	codigo_posicion	desc_posicion_EN	desc_posicion_ES	num_posicion
1	1	PG	POINT GUARD	BASE	1
2	2	SG	SHOOTING GUARD	ESCOLTA	2
3	3	SF	SMALL FORDWARD	ALERÓ	3
4	4	PF	POWER FORDWARD	ALA-PIVOT	4
5	5	C	CENTER	PIVOT	5
6	9999	DESC	NULL	DESCONOCIDO	9999

DIM_Tiempo

	pk_date	date_year	date_month	date_day	date_date	temporada
57	57	2006	10	1	NULL	2005 - 2006
58	58	2007	10	1	NULL	2006 - 2007
59	59	2008	10	1	NULL	2007 - 2008
60	60	2009	10	1	NULL	2008 - 2009
61	61	2010	10	1	NULL	2009 - 2010
62	62	2011	10	1	NULL	2010 - 2011
63	63	2012	10	1	NULL	2011 - 2012
64	64	2013	10	1	NULL	2012 - 2013
65	65	2014	10	1	NULL	2013 - 2014
66	66	2015	10	1	NULL	2014 - 2015

FACT_FREE_THROWS

	id_temporada	id_partido	id_jugador	id_jugada	id_minuto	periodo_juego	resultado_tiro_libre
1	58	21	117	280	170	4	1
2	58	21	117	282	170	4	1
3	58	353	3876	9151	476	1	1
4	58	353	3876	9158	476	1	0
5	58	353	3869	9134	430	1	1
6	58	353	3869	9135	430	1	1
7	58	353	3467	8081	285	1	1
8	58	353	3467	8082	285	1	1
9	58	353	3997	9383	277	1	1
10	58	353	2907	6625	170	1	1

FACT_SEASON_STATS

Results	Messages	id_season	id_player	id_team	id_position	league	GP	PTS	MIN	FGM	FGA	FG%	_3PM	_3PA	_3P%	_2PM	_2PA	_2P%	FTM	FT%
16452		51	1336	2	1	NBA	8	19	74	6	15	0.4	4	9	0.444	2	6	0.333	3	3
16453		51	2386	37	2	NBA	62	495	1281	161	346	0.465	66	154	0.429	95	192	0.495	107	11
16454		51	4212	4	5	NBA	82	541	1505	219	459	0.477	1	8	0.125	218	451	0.483	102	15
16455		51	2394	17	3	NBA	24	64	130	23	70	0.329	2	12	0.167	21	58	0.362	16	24
16456		51	343	8	5	NBA	3	2	8	1	3	0.333	0	0	NULL	1	3	0.333	0	0
16457		51	903	6	2	NBA	48	370	862	145	350	0.414	31	89	0.348	114	261	0.437	49	82
16458		51	2927	18	4	NBA	1	0	3	0	3	0	0	0	NULL	0	3	0	0	0
16459		51	4246	17	1	NBA	82	733	1691	271	561	0.483	35	93	0.376	236	468	0.504	156	19
16460		51	3100	9999	1	NBA	82	11...	3155	459	10...	0.445	77	212	0.363	382	819	0.466	195	25