

# **PRA 2**

## **Carga y explotación de Datos**

**Diseño y Uso de Bases de Datos Analíticas**

**UOC**

**Universitat Oberta  
de Catalunya**

---

**Reynel López Lantigua**

**Máster Universitario en  
Ciencia de Datos**

## I. Identificación de los procesos ETL's

A la hora de diseñar los procesos de carga de un DW, no hay una única estrategia que sirva para todos los casos, pero es habitual estructurar los tratamientos en diferentes bloques para garantizar la integridad en la carga de datos y facilitar la ejecución de estos. En nuestro caso, cada uno de estos bloques se corresponde con una transformación de Pentaho Data Integration (PDI, en adelante). 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 tablas intermedias en el área intermedia (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.

Esta separación permite la ejecución de los bloques de forma consecutiva (lo más habitual) y a la vez, de forma "aislada" en el caso de que se requiera reprocesar alguno de los bloques por cualquier incidencia que se pueda producir en la ejecución. Así mismo, la evolución de los bloques para incorporar nuevos requerimientos se puede llevar a cabo de forma independiente optimizando el tiempo requerido a largo plazo.

Finalmente, la utilización de bloques de procesos permite que el mantenimiento de las ETL's será más sencillo en un futuro cuando se tengan que modificar. Normalmente, el personal responsable de la creación de las ETL's no coincide con el personal responsable de la ejecución de estas.

Consideramos el uso de un área intermedia (staging area) como una buena práctica para la extracción de los datos desde las fuentes origen y la carga al modelo multidimensional, las ventajas que se obtienen con su uso son entre otras:

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

## Bloque IN (de las fuentes a tablas intermedias)

Nombre ETL	Descripción	Orígenes de datos	Tabla de 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_SEASON_STATS_2005_2017  STG_NBA_SEASON_STATS_1950_2017
IN_WNBA_PLAYERS	Carga de los datos correspondientes a la información de las jugadoras de la WNBA	WNBA_Players_List.txt.	STG_WNBA_PLAYERES
IN_NBA_PLAYERS	Carga de los datos correspondientes estado(activo-retirado) de los jugadores de la NBA	NBA_Players_List.xls	STG_NBA_PLAYERES
IN_NBA_TEAMS	Carga de los datos correspondientes a los datos de los EQUIPOS de la NBA	NBA_Teams.xml	STG_NBA_TEAMS
IN_WNBA_TEAMS	Carga de los datos correspondientes a los datos de los EQUIPOS de la WNBA	WNBA_Teams.xls	STG_WNBA_TEAMS
IN_TEAMS_CODE	Carga de los datos correspondientes a los códigos de los EQUIPOS de la tanto de la NBA como de la WNBA.	TeamCodes.txt	STG_TEAMS_CODES
IN_STATES	Carga de los datos correspondientes a la información de los estados miembros de los Estados Unidos.	Estados_Unidos.json	STG_STATES
IN_FREE_THROWS	Carga de los datos correspondientes a los lanzamientos de tiros libres	nba_free_throws.csv	STG_FREE_THROWS

IN_PLAYERS_DATA	Carga de los datos correspondientes a la información de las jugadoras de la WNBA	NBA_players_data.json	STG_PLAYERS_DATA
-----------------	--	-----------------------	------------------

### Bloque TR (de las tablas intermedias a nuestro 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 sino ha habido errores.

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

Nombre ETL	Descripción	Orígenes de datos	Tabla de destino (Stage)
TR_DIM_TIEMPO	Carga y transformación de la dimensión TIEMPO que contiene inicialmente los datos de los años de las temporadas.	STG_WNBA_SEASON_STATS_2005_2017  STG_NBA_SEASON_STATS_1950_2017	DIM_Tiempo
TR_DIM_EQUIPOS	Carga y transformación de la dimensión EQUIPOS .	STG_TEAMS_CODES  STG_NBA_TEAMS  STG_WNBA_TEAM	DIM_Equipos
TR_DIM_Minutos	Carga y transformación de la dimensión MINUTOS .	STG_FREE_THROWS	DIM_Minutos

TR_DIM_Jugadores	Carga y transformación de la dimensión JUGADORES .	STG_WNBA_PLAYERS  STG_NBA_PLAYER  STG_PLAYERS_DATA	DIM_Jugadores
TR_DIM_Jugadas	Carga y transformación de la dimensión con datos de tipos de jugadas.	STG_FREE_THROWS	DIM_Jugadas
TR_DIM_PosicionesJuego	Carga y transformación de la dimensión con datos de las posiciones.	STG_NBA_SEASON_STATS_1950_2017	DIM_PosicionesJuego
TR_DIM_Partidos	Carga y transformación de la dimensión PARTIDOS .	STG_FREE_THROWS	DIM_Partidos
TR_DIM_Estados	Carga y transformación de la dimensión ESTADOS .	STG_STATES	DIM_Estados_EEUU
TR_DIM_Divisiones	Carga y transformación de la dimensión DIVISIONES .	STG_NBA_TEAM	DIM_Divisiones
TR_DIM_Conferencias	Carga y transformación de la dimensión con información de las conferencias de la NBA/WNBA.	STG_WNBA_TEAMS  STG_NBA_TEAMS	DIM_Conferencias

Nombre ETL	Descripción	Tablas de origen
TR_FACT_SEASONS_STATS	Carga y transformación de la tabla de hechos FACT_SEASONS_STATS.	FACT_SEASONS_STATS
TR_FACT_FREE_THROWS	Carga y transformación de la tabla de hechos FACT_FREE_THROWS.	FACT_NBA_FREE_THROWS

## II. Diseño y desarrollo de los procesos ETL's

En este apartado, se deben diseñar y desarrollar los procesos de carga identificados en el punto anterior, mediante la herramienta de diseño proporcionada: Pentaho Data Integration (PDI).

### Creación de tablas

El primer paso para la implementación de los procesos ETL consiste en la creación de las tablas. Esto se llevará a cabo una única vez, mediante scripts sobre la base de datos proporcionada, en nuestro caso, SQL Server. Se deberán crear las tablas intermedias y las tablas del modelo dimensional de la solución oficial, es decir, las dimensiones y tablas de hecho.

#### Tabla intermedia STG\_WNBA\_SEASON\_STATS\_2005\_2017

```
CREATE TABLE [dbo].[STG_WNBA_SEASON_STATS_2005_2017](
    [ID] [numeric](8 , 0) NULL,
    [SEASON] [numeric](4 , 0) NOT NULL,
    [PLAYER] [varchar](255) NULL,
    [TEAM] [varchar](100) NULL,
    [AGE] [numeric](8, 0) NULL,
    [GP] [numeric](8, 0) NULL,
    [W] [numeric](8 , 0) NULL,
    [L] [numeric](8 , 0) NULL,
    [MIN] [numeric](8 , 0) NULL,
    [PTS] [numeric](8 , 0) NULL,
    [FGM] [numeric](8 , 0) NULL,
    [FGA] [numeric](8 , 0) NULL,
    [FG%] [numeric](8 , 3) NULL,
    [_3PM] [numeric](8 , 0) NULL,
    [_3PA] [numeric](8 , 0) NULL,
    [_3P%] [numeric](8 , 3) NULL,
    [FTM] [numeric](8 , 0) NULL,
    [FTA] [numeric](8 , 0) NULL,
    [FT%] [numeric](8 , 3) NULL,
    [OREB] [numeric](8 , 0) NULL,
    [DREB] [numeric](8 , 0) NULL,
    [REB] [numeric](8 , 0) NULL,
    [AST] [numeric](8 , 0) NULL,
    [TOV] [numeric](8 , 0) NULL,
    [STL] [numeric](8 , 0) NULL,
    [BLK] [numeric](8 , 0) NULL,
    [PF] [numeric](8 , 0) NULL,
    [FP] [numeric](8 , 0) NULL,
    [DD2] [numeric](8 , 0) NULL,
    [TD3] [numeric](8 , 0) NULL,
    [+/-] [numeric](8 , 0) NULL,
)
```

## Tabla intermedia STG\_NBA\_SEASON\_STATS\_1950\_2017

```
CREATE TABLE [dbo].[STG_NBA_SEASON_STATS_1950_2017 ](
    [ID] [numeric](8 , 0) NOT NULL,
    [Year] [numeric](4 , 0) NULL,
    [Player] [varchar](255) NULL,
    [Pos] [varchar](100) NULL,
    [Age] [numeric](3 , 0) NULL,
    [Tm] [varchar](255) NULL,
    [G] [numeric](8 , 0) NULL,
    [GS] [numeric](8 , 0) NULL,
    [MP] [numeric](8 , 0) NULL,
    [PER] [numeric](8 , 2) NULL,
    [TS%] [numeric](8 , 2) NULL,
    [_3PAr] [numeric](8 , 0) NULL,
    [FTr] [numeric](8 , 2) NULL,
    [ORB%] [numeric](8 , 2) NULL,
    [DRB%] [numeric](8 , 2) NULL,
    [TRB%] [numeric](8 , 2) NULL,
    [AST%] [numeric](8 , 2) NULL,
    [STL%] [numeric](8 , 2) NULL,
    [BLK%] [numeric](8 , 2) NULL,
    [TOV%] [numeric](8 , 2) NULL,
    [USG%] [numeric](8 , 2) NULL,
    [OWS] [numeric](8 , 2) NULL,
    [DWS] [numeric](8 , 0) NULL,
    [WS] [numeric](8 , 2) NULL,
    [WSx48] [numeric](8 , 2) NULL,
    [OBPM] [numeric](8 , 2) NULL,
    [DBPM] [numeric](8 , 0) NULL,
    [BPM] [numeric](8 , 0) NULL,
    [VORP] [numeric](8 , 2) NULL,
    [FG] [numeric](8 , 0) NULL,
    [FGA] [numeric](8 , 0) NULL,
    [FG%] [numeric](8 , 2) NULL,
    [_3P] [numeric](8 , 0) NULL,
    [_3PA] [numeric](8 , 0) NULL,
    [_3P%] [numeric](8 , 2) NULL,
    [_2P] [numeric](8 , 0) NULL,
    [_2PA] [numeric](8 , 0) NULL,
    [_2P%] [numeric](8 , 2) NULL,
    [eFG%] [numeric](8 , 2) NULL,
    [FT] [numeric](8 , 0) NULL,
    [FTA] [numeric](8 , 0) NULL,
    [FT%] [numeric](8 , 2) NULL,
    [ORB] [numeric](8 , 0) NULL,
    [DRB] [numeric](8 , 0) NULL,
    [TRB] [numeric](8 , 0) NULL,
    [AST] [numeric](8 , 0) NULL,
    [STL] [numeric](8 , 0) NULL,
    [BLK] [numeric](8 , 0) NULL,
    [TOV] [numeric](8 , 0) NULL,
    [PF] [numeric](8 , 0) NULL,
    [PTS] [numeric](8 , 0) NULL,
)
```

## Tabla intermedia STG\_WNBA\_PLAYERES

```
CREATE TABLE [dbo].[STG_NBA_PLAYERES](  
    [Player] [varchar](255) NULL,  
    [Active] [numeric](4, 0) NULL,  
)
```

## Tabla intermedia STG\_NBA\_PLAYERES

```
CREATE TABLE [dbo].[STG_NBA_PLAYERES](  
    [Player] [varchar](255) NOT NULL,  
    [Historic] [varchar](255) NOT NULL,  
)
```

## Tabla intermedia STG\_NBA\_TEAMS

```
CREATE TABLE [dbo].[STG_NBA_TEAMS](  
    [Conferencia] [varchar](100) NOT NULL,  
    [División] [varchar](100) NOT NULL,  
    [Equipo] [varchar](255) NOT NULL,  
    [Ciudad] [varchar](255) NOT NULL,  
    [Estado] [varchar](255) NOT NULL,  
    [Pabellón] [varchar](255) NOT NULL,  
    [Fundado] [numeric](4, 0) NOT NULL,  
    [Patrocinio] [varchar](255) NOT NULL,  
)
```

## Tabla intermedia STG\_WNBA\_TEAMS

```
CREATE TABLE [dbo].[STG_WNBA_TEAMS](  
    [ID] [numeric](8, 0) NOT NULL,  
    [Conferencia] [varchar](100) NULL,  
    [Equipo] [varchar](255) NULL,  
    [Ciudad] [varchar](255) NULL,  
    [Estado] [varchar](255) NULL,  
)
```

## Tabla intermedia STG\_TEAMS\_CODES



```
CREATE TABLE [dbo].[STG_TEAMS_CODES](  
    [league] [varchar](50) NOT NULL,  
    [team] [varchar](255) NOT NULL,  
    [code] [varchar](50) NOT NULL,  
)
```

## Tabla intermedia STG\_STATES

```
CREATE TABLE [dbo].[STG_STATES](  
    [Estado] [varchar](255) NULL,  
    [Nombre oficial] [varchar](100) NULL,  
    [Superficie (km2)] [numeric](8 , 0) NULL,  
    [Abrev.] [varchar](50) NULL,  
    [Población (2010)] [numeric](15 , 0) NULL,  
    [Densidad de población (hab/km2)] [numeric](8 , 3) NULL,  
    [Capital] [varchar](255) NULL,  
)
```

## Tabla intermedia STG\_FREE\_THROWS

```
CREATE TABLE [dbo].[STG_FREE_THROWS](  
    [end_result] [varchar](15) NULL,  
    [game] [varchar](50) NULL,  
    [game_id] [numeric](10 , 0) NULL,  
    [period] [numeric](4, 0) NULL,  
    [play] [varchar](255) NULL,  
    [player] [varchar](255) NULL,  
    [playoffs] [varchar](50) NULL,  
    [score] [varchar](50) NULL,  
    [season] [varchar](50) NULL,  
    [shot_made] [numeric](2, 0) NULL,  
    [time] [varchar](50) NULL,  
)
```

## Tabla intermedia STG\_PLAYERS\_DATA

```
CREATE TABLE [dbo].[STG_PLAYERS_DATA](
    [id] [varchar](15) NULL,
    [birthDate] [varchar](15) NULL,
    [birthPlace] [varchar](50) NULL,
    [career_AST] [numeric](8, 2) NULL,
    [career_FG] [numeric](8, 2) NULL,
    [career_FG3] [numeric](8, 2) NULL,
    [career_FT] [numeric](8, 2) NULL,
    [career_G] [numeric](8, 2) NULL,
    [career_PER] [numeric](8, 2) NULL,
    [career PTS] [numeric](8, 2) NULL,
    [career_TRB] [numeric](8, 2) NULL,
    [career_WS] [numeric](8, 2) NULL,
    [career_eFG] [numeric](8, 2) NULL,
    [college] [varchar](255) NULL,
    [Height] [varchar](50) NULL,
    [highSchool] [varchar](255) NULL,
    [Name] [varchar](255) NULL,
    [position] [varchar](100) NULL,
    [Shoots] [varchar](50) NULL,
    [weight] [varchar](50) NULL,
)
```

## Tabla dimensión DIM\_Tiempo

```
CREATE TABLE [dbo].[DIM_Tiempo](
    [pk_date] [numeric](4, 0) NOT NULL,
    [date_year] [numeric](4, 0) NULL,
    [date_month] [numeric](4, 0) NULL,
    [date_day] [numeric](4, 0) NULL,
    [date_date] [date] NULL,
    [temporada] [varchar](50) NULL,
    CONSTRAINT [PK_DIM_Tiempo] PRIMARY KEY CLUSTERED
(
    [pk_date] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

## Tabla dimensión Minutos (DIM\_MINUTOS)

```
CREATE TABLE [dbo].[DIM_Minutos](
    [pk_minutoSegundo] [numeric](4, 0) NOT NULL,
    [minuto] [numeric] (4, 0) NULL,
    [segundo] [numeric] (4, 0) NULL,
    [minutoSegundo] [varchar] (50) NULL,
    CONSTRAINT [PK_DIM_Minutos] PRIMARY KEY CLUSTERED
    (
        [pk_minutoSegundo] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

## Tabla dimensión Jugada (DIM\_Jugadas)

```
CREATE TABLE [dbo].[DIM_Jugadas](
    [pk_jugada] [numeric](4, 0) NOT NULL,
    [codigo_jugada] [varchar] (50) NULL,
    [desc_jugada] [varchar] (100) NULL,
    CONSTRAINT [PK_DIM_Jugadas] PRIMARY KEY CLUSTERED
    (
        [pk_jugada] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

## Tabla dimensión DIM\_Equipos

```

CREATE TABLE [dbo].[DIM_Equipos](
    [pk_equipo] [numeric](4, 0) NOT NULL,
    [id_estado_EEUU] [numeric] (4, 0) NULL,
    [id_division] [numeric] (4, 0) NULL,
    [liga] [varchar] (50) NULL,
    [codigo_equipo] [varchar] (50) NULL,
    [nombre_equipo] [varchar] (555) NULL,
    [ciudad] [varchar] (255) NULL,
    [pabellon] [varchar] (255) NULL,
    [fundado] [numeric] (4, 0) NULL,
    [patrocinio] [varchar] (255) NULL,
    CONSTRAINT [PK_DIM_Equipos] PRIMARY KEY CLUSTERED
    (
        [pk_equipo] ASC
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[DIM_Equipos] WITH CHECK ADD CONSTRAINT
[FK_DIM_Equipos_DIM_Estados_EEUU] FOREIGN KEY([id_estado_EEUU])
REFERENCES [dbo].[DIM_Estados_EEUU] ([pk_estado_EEUU])
GO

ALTER TABLE [dbo].[DIM_Equipos] CHECK CONSTRAINT
[FK_DIM_Equipos_DIM_Estados_EEUU]
GO

ALTER TABLE [dbo].[DIM_Equipos] WITH CHECK ADD CONSTRAINT
[FK_DIM_Equipos_DIM_Divisiones] FOREIGN KEY([id_division])
REFERENCES [dbo].[DIM_Divisiones] ([pk_division])
GO

ALTER TABLE [dbo].[DIM_Equipos] CHECK CONSTRAINT
[FK_DIM_Equipos_DIM_Divisiones]
GO

```

## Tabla dimensión DIM\_Jugadores

```
CREATE TABLE [dbo].[DIM_Jugadores](
    [pk_jugador] [numeric](4, 0) NOT NULL,
    [liga] [varchar] (50) NULL,
    [cod_jugador] [varchar] (50) NULL,
    [nombre] [varchar] (255) NULL,
    [posicion_juego] [varchar] (100) NULL,
    [sexo] [numeric] (4, 0) NULL,
    [activo] [numeric] (4, 0) NULL,
    [altura] [varchar] (50) NULL,
    [peso] [varchar] (50) NULL,
    [shoots] [varchar] (50) NULL,
    [Universidad] [varchar] (255) NULL,
    [fecha_nacimiento] [varchar] (50) NULL,
    [ciudad_nacimiento] [varchar] (255) NULL,
    [career_AST] [numeric] (8, 2) NULL,
    [career_FG%] [numeric] (8, 2) NULL,
    [career_FG3%] [numeric] (8, 2) NULL,
    [career_FT%] [numeric] (8, 2) NULL,
    [career_G] [numeric] (8, 2) NULL,
    [career_PER] [numeric] (8, 2) NULL,
    [career_PTS] [numeric] (8, 2) NULL,
    [career_TRB] [numeric] (8, 2) NULL,
    [career_WS] [numeric] (8, 2) NULL,
    [career_eFG%] [numeric] (8, 2) NULL,
    CONSTRAINT [PK_DIM_Jugadores] PRIMARY KEY CLUSTERED
    (
        [pk_jugador] ASC
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
        IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
        ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

## Tabla dimensión DIM\_PosicionesJuego

```
CREATE TABLE [dbo].[DIM_PosicionesJuego](
    [pk_posicion_juego] [numeric](4, 0) NOT NULL,
    [codigo_posicion] [varchar] (10) NULL,
    [desc_posicion_EN] [varchar] (50) NULL,
    [desc_posicion_ES] [varchar] (50) NULL,
    [num_posicion] [numeric] (4, 0) NULL,
    CONSTRAINT [PK_DIM_PosicionesJuego] PRIMARY KEY CLUSTERED
    (
        [pk_posicion_juego] ASC
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
        IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
        ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

## Tabla dimensión DIM\_Partidos

```
CREATE TABLE [dbo].[DIM_Partidos](
    [pk_partido] [numeric](10, 0) NOT NULL,
    [partido] [varchar] (50) NULL,
    [playoffs] [varchar] (50) NULL,
    [temporada] [varchar] (50) NULL,
    [resultado] [varchar] (50) NULL,
    CONSTRAINT [PK_DIM_Partidos] PRIMARY KEY CLUSTERED
    (
        [pk_partido] ASC
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
        IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
        ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

## Tabla dimensión DIM\_Estados\_EEUU

```
CREATE TABLE [dbo].[DIM_Estados_EEUU](
    [pk_estado_EEUU] [numeric](4, 0) NOT NULL,
    [codigo] [varchar] (50) NULL,
    [nombre] [varchar] (255) NULL,
    [Nombre oficial] [varchar] (100) NULL,
    [superficie] [numeric] (8, 0) NULL,
    [poblacion] bigint NULL,
    [capital] [varchar] (100) NULL,
    [densidadPoblacion] [numeric] (8, 3) NULL,
    CONSTRAINT [PK_DIM_Estados_EEUU] PRIMARY KEY CLUSTERED
    (
        [pk_estado_EEUU] ASC
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
        IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
        ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

## Tabla dimensión DIM\_Divisiones

```
CREATE TABLE [dbo].[DIM_Divisiones](
    [pk_division] [numeric] (4, 0) NOT NULL,
    [codigo_division] [varchar] (50) NULL,
    [División] [varchar] (100) NULL,
    [id_conferencia] [numeric] (4) NOT NULL,
    CONSTRAINT [PK_DIM_Divisiones] PRIMARY KEY CLUSTERED
    (
        [pk_division] ASC
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[DIM_Divisiones] WITH CHECK ADD CONSTRAINT
[FK_DIM_Divisiones_DIM_Conferencias] FOREIGN KEY([id_conferencia])
REFERENCES [dbo].[DIM_Conferencias] ([pk_conferencia])
GO

ALTER TABLE [dbo].[DIM_Divisiones] CHECK CONSTRAINT
[FK_DIM_Divisiones_DIM_Conferencias] GO
```

## Tabla dimensión DIM\_Conferencias

```
CREATE TABLE [dbo].[DIM_Conferencias](
    [pk_conferencia] [numeric] (4, 0) NOT NULL,
    [codigo_conferencia] [varchar] (50) NULL,
    [Conferencia] [varchar] (100) NULL,
    CONSTRAINT [PK_DIM_Conferencias] PRIMARY KEY CLUSTERED
    (
        [pk_conferencia] ASC
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

## Tabla de Hechos (FACT\_FREE\_THROWS)

```
CREATE TABLE [dbo].[FACT_FREE_THROWS](
    [id_temporada] [numeric](4, 0) NOT NULL,
    [id_partido] [numeric](10, 0) NULL,
    [id_jugador] [numeric](4, 0) NULL,
    [id_jugada] [numeric](4, 0) NULL,
    [id_Minuto] [numeric](10, 0) NULL,
    [periodo_juego] [numeric](4, 0) NULL,
    [resultado_tiro_libre] [numeric](8, 2) NULL,

) ON [PRIMARY]
GO

ALTER TABLE [dbo].[FACT_FREE_THROWS] WITH CHECK ADD CONSTRAINT
[FK_FACT_FREE_THROWS_SEASON] FOREIGN KEY([id_temporada])
REFERENCES [dbo].[DIM_Tiempo] ([pk_date])
GO

ALTER TABLE [dbo].[FACT_FREE_THROWS] CHECK CONSTRAINT
[FK_FACT_FREE_THROWS_SEASON] GO

ALTER TABLE [dbo].[FACT_FREE_THROWS] WITH CHECK ADD CONSTRAINT
[FK_FACT_FREE_THROWS_GAME] FOREIGN KEY([id_partido])
REFERENCES [dbo].[DIM_Partidos] ([pk_partido])
GO

ALTER TABLE [dbo].[FACT_FREE_THROWS] CHECK CONSTRAINT
[FK_FACT_FREE_THROWS_GAME] GO

ALTER TABLE [dbo].[FACT_FREE_THROWS] WITH CHECK ADD CONSTRAINT
[FK_FACT_FREE_THROWS_PLAYER] FOREIGN KEY([id_jugador])
REFERENCES [dbo].[DIM_Jugadores] ([pk_jugador])
GO

ALTER TABLE [dbo].[FACT_FREE_THROWS] CHECK CONSTRAINT
[FK_FACT_FREE_THROWS_PLAYER] GO

ALTER TABLE [dbo].[FACT_FREE_THROWS] WITH CHECK ADD CONSTRAINT
[FK_FACT_FREE_THROWS_MOVE] FOREIGN KEY([id_jugada])
REFERENCES [dbo].[DIM_Jugadas] ([pk_jugada])
GO

ALTER TABLE [dbo].[FACT_FREE_THROWS] CHECK CONSTRAINT
[FK_FACT_FREE_THROWS_MOVE] GO

ALTER TABLE [dbo].[FACT_FREE_THROWS] WITH CHECK ADD CONSTRAINT
[FK_FACT_FREE_THROWS_MINUTE] FOREIGN KEY([id_Minuto])
REFERENCES [dbo].[DIM_Minutos] ([pk_minutoSegundo])
GO

ALTER TABLE [dbo].[FACT_FREE_THROWS] CHECK CONSTRAINT
[FK_FACT_FREE_THROWS_MINUTE] GO
```



## Tabla de Hechos (FACT\_SEASONS\_STATS)

```
CREATE TABLE [dbo].[FACT_SEASONS_STATS](
    [id_season] [numeric](4, 0) NULL,
    [id_player] [numeric](4, 0) NULL,
    [id_team] [numeric](4, 0) NULL,
    [id_position] [numeric](4, 0) NULL,
    [league] [varchar](50) NULL,
    [GP] [numeric](8, 2) NULL,
    [PTS] [numeric](8, 0) NULL,
    [MIN] [numeric](8, 2) NULL,
    [FGM] [numeric](8, 2) NULL,
    [FGA] [numeric](8, 2) NULL,
    [FG%] [numeric](8, 2) NULL,
    [_3PM] [numeric](8, 0) NULL,
    [_3PA] [numeric](8, 0) NULL,
    [_3P%] [numeric](8, 2) NULL,
    [_2PM] [numeric](8, 0) NULL,
    [_2PA] [numeric](8, 0) NULL,
    [_2P%] [numeric](8, 2) NULL,
    [FTM] [numeric](8, 2) NULL,
    [FTA] [numeric](8, 2) NULL,
    [FT%] [numeric](8, 2) NULL,
    [eFG%] [numeric](8, 2) NULL,
    [OREB] [numeric](8, 2) NULL,
    [DREB] [numeric](8, 2) NULL,
    [REB] [numeric](8, 2) NULL,
    [AST] [numeric](8, 2) NULL,
    [TOV] [numeric](8, 2) NULL,
    [STL] [numeric](8, 2) NULL,
    [BLK] [numeric](8, 2) NULL,
    [PF] [numeric](8, 2) NULL,
    [EFF] [numeric](8, 3) NULL,

) ON [PRIMARY]
GO

ALTER TABLE [dbo].[FACT_SEASONS_STATS] WITH CHECK ADD CONSTRAINT
[FK_FACT_SEASONS_STATS_SEASON] FOREIGN KEY([id_season])
REFERENCES [dbo].[DIM_Tiempo] ([pk_date])
GO

ALTER TABLE [dbo].[FACT_SEASONS_STATS] CHECK CONSTRAINT
[FK_FACT_SEASONS_STATS_SEASON] GO

ALTER TABLE [dbo].[FACT_SEASONS_STATS] WITH CHECK ADD CONSTRAINT
[FK_FACT_SEASONS_STATS_PLAYER] FOREIGN KEY([id_player])
REFERENCES [dbo].[DIM_Jugadores] ([pk_jugador])
GO

ALTER TABLE [dbo].[FACT_SEASONS_STATS] CHECK CONSTRAINT
[FK_FACT_SEASONS_STATS_PLAYER] GO

ALTER TABLE [dbo].[FACT_SEASONS_STATS] WITH CHECK ADD CONSTRAINT
[FK_FACT_SEASONS_STATS_TEAM] FOREIGN KEY([id_team])
REFERENCES [dbo].[DIM_Equipos] ([pk_equipo])
GO

ALTER TABLE [dbo].[FACT_SEASONS_STATS] CHECK CONSTRAINT
```

## Transformaciones

Una vez que tenemos implementado el modelo físico del almacén, pasaremos a diseñar los procesos ETL que permitirán poblar las tablas intermedias del área intermedia (Stagin Area) y las tablas dimensiones y de hechos del data mart que hemos diseñado.

### Variable 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: \\ucs1r1uocfiler1\CtxRedirFolders\1132412\Desktop\fuentes

La referencia a la variable de entorno durante la implementación de los procesos se realiza mediante llaves, de esta manera: {DIR\_ENT}

### Procesos ETL (Bloque IN)

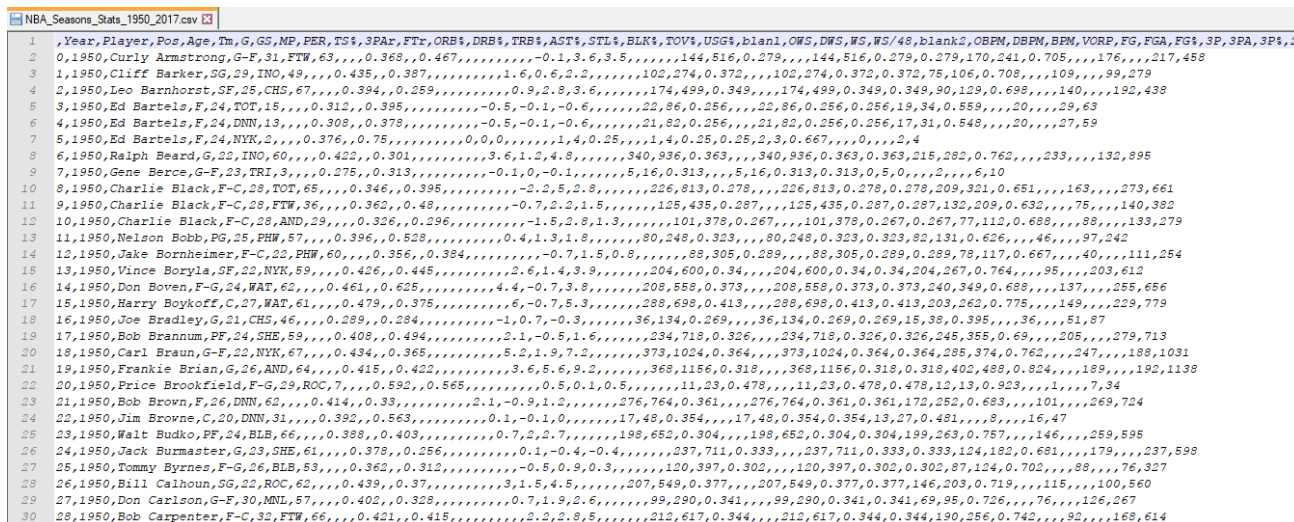
En este apartado se realizan las transformaciones necesarias para cargar en la zona intermedia (Stagin area) los datos provenientes de las fuentes de datos. Es necesario resaltar que los datos provienen de fuentes externas (no operacionales), y presentan diferentes tipos de archivos y formatos, por lo que las tareas de transformación realizadas en este apartado permitirán integrar todos estos datos en el *Data Wharehouse*. Se describirán cada una de estas transformaciones y se proporciona una guía de como implementarlas, siguiendo paso a paso las diferentes etapas del proceso, desde su lectura hasta su carga en las tablas de la base de datos.

### Transformación IN\_STATS

Esta transformación tiene como objetivo efectuar la carga de los datos de los archivos “WNBA\_Seasons\_Stats\_2005\_2017.xls” y “NBA\_Seasons\_Stats\_1950\_2017.csv” en sus respectivas tablas en la *Stagin Area*.

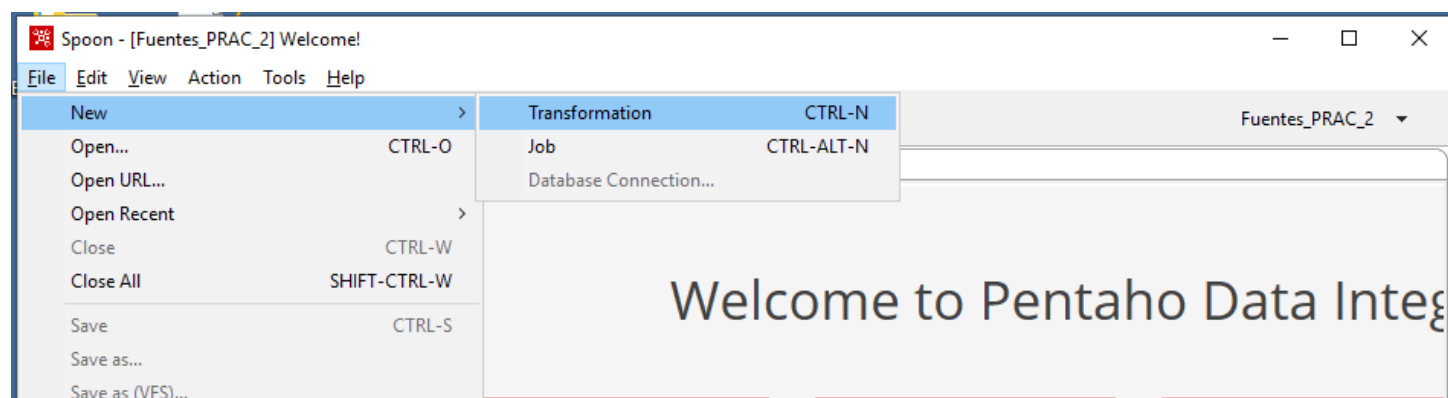
### NBA\_Seasons\_Stats

Para comenzar con estas transformaciones es útil conocer que aspecto tiene el archivo que se desea cargar, en este caso al ser un archivo csv se utiliza un editor de texto (Notepad++) para obtener esta aproximación.



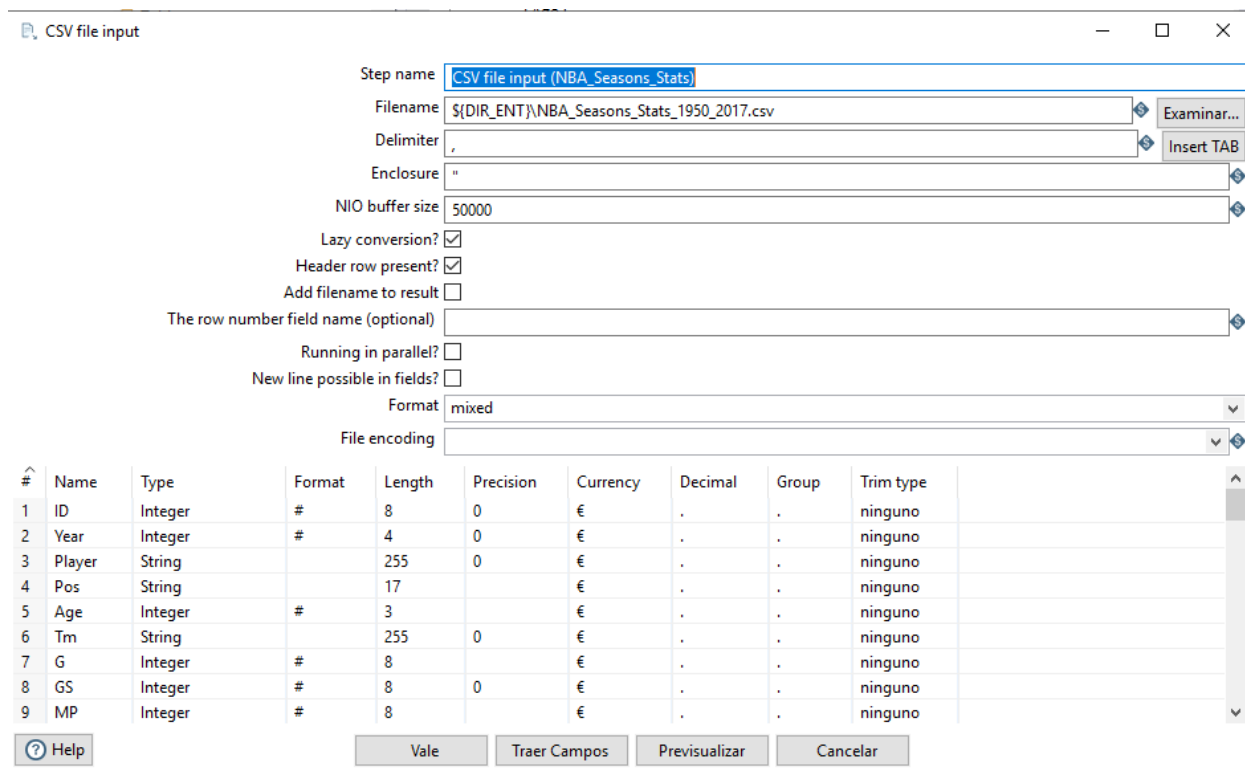
En esta figura se puede apreciar que los datos almacenados cuentan con una cabecera y están separados por coma. En este caso no será necesario realizar ninguna transformación manual a estos datos, por lo que se puede crear la transformación.

Este es el procedimiento inicial para la creación de todas las transformaciones, hay que dirigirse a la pestaña **file->New->Transformation**, una vez aquí se pulsa con el clic izquierdo y ya estaría creada la transformación.



Iniciaremos la transformación cargando los datos un “CSV INPUT” estableciendo los formatos de campo correctos. Hay que tener especial cuidado con los delimitadores de

decimales y de grupo, esto puede causar la lectura incorrecta de algunos datos, nótese que se establecen como delimitadores decimales la coma “,” y como delimitador de grupo el punto “.”.



#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type	
1	ID	Integer	#	8	0	€	.	.	ninguno	
2	Year	Integer	#	4	0	€	.	.	ninguno	
3	Player	String		255	0	€	.	.	ninguno	
4	Pos	String		17		€	.	.	ninguno	
5	Age	Integer	#	3		€	.	.	ninguno	
6	Tm	String		255	0	€	.	.	ninguno	
7	G	Integer	#	8		€	.	.	ninguno	
8	GS	Integer	#	8	0	€	.	.	ninguno	
9	MP	Integer	#	8		€	.	.	ninguno	

En la transformación “Secciona/Renombrar valores” renombraremos algunos nombres campos para hacerlos compatibles con SQL Server (los campos no pueden empezar por número y no pueden contener caracteres como “/”). También eliminaremos algunos campos que no nos interesa cargar: blanl, blank2.

Selecciona/Renombra valores
.....

Nombre paso Selecciona/Renombra valores

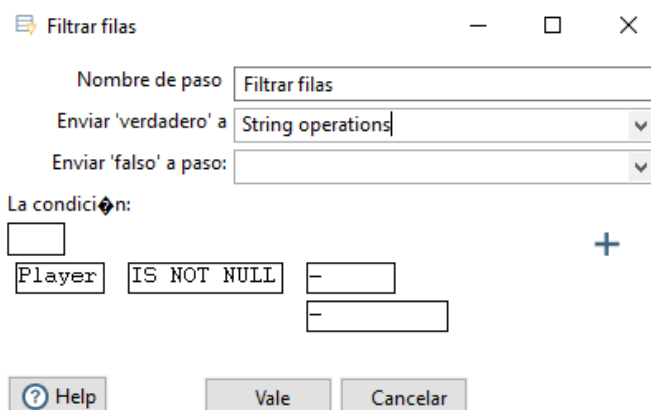
Selecciona & Modifica
Eliminar
Meta-información

Campos :

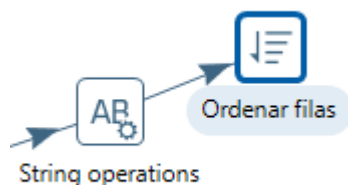
#	Nombre campo	Renombrar a	Longitud	Precisión
1	Year			
2	Player			
3	Pos			
4	Age			
5	Tm			
6	G			
7	GS			
8	MP			
9	PER			
10	TS%			
11	3PAr	_3PAr		
12	FT%			
13	ORB%			
14	DRB%			
15	TRB%			
16	AST%			
17	STL%			
18	BLK%			
19	TOV%			
20	USG%			
21	blanl			
22	OVS			
23	DWS			
24	WS			
25	WS/48	WSx48		
26	blank2			
27	OBPM			
28	DBPM			
29	BPM			
30	VORP			
31	FG			
32	FGA			
33	FG%			
34	3P	_3P		
35	3PA	_3PA		
36	3P%	_3P%		
37	2P	_2P		
38	2PA	_2PA		
39	2P%	_2P%		
40	eFG%			
41	FT			
42	FTA			

☐ Include unspecified fields, ordered by name

Filtramos los registros que no tiene informado el campo Player y los eliminamos del flujo de datos.



En el siguiente paso pasamos a mayúsculas, eliminamos espacios en blanco y ordenamos los registros por el campo Player.



Finalmente, cargamos los datos en el stage mapeando los campos del stream de datos con los campos de la tabla física.

Salida de Tabla

Nombre paso: Salida Tabla (STG\_NBA\_SEASON\_STATS\_1950\_2017)

Conexión: PRA\_2 [Editar...] [Nuevo...] [Wizard...]

Esquema destino: dbo [Examinar...]

Tabla destino: STG\_NBA\_SEASON\_STATS\_1950\_2017 [Examinar...]

Tamaño de transacción (commit): 1000

Vaciar tabla: ☐

Ignorar errores de inserción: ☐

Specify database fields: ☒

Main options | Database fields

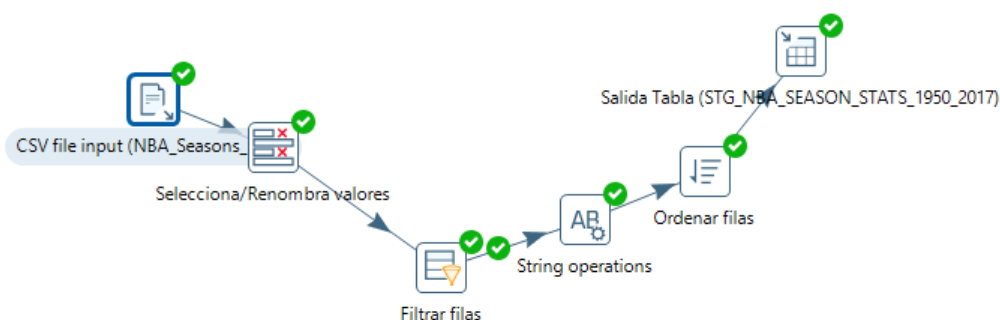
Fields to insert:

#	Table field	Stream field
1	ID	ID
2	Year	Year
3	Player	Player
4	Pos	Pos
5	Age	Age
6	Tm	Tm
7	G	G
8	GS	GS
9	MP	MP
10	PER	PER
11	TS%	TS%
12	_3PAr	_3PAr
13	FTr	FTr
14	ORB%	ORB%
15	DRB%	DRB%
16	TRB%	TRB%
17	AST%	AST%
18	STL%	STL%
19	BLK%	BLK%
20	TOV%	TOV%
21	USG%	USG%
22	OWS	OWS
23	DWS	DWS
24	WS	WS
25	WSx48	WSx48
26	OBPM	OBPM
27	DBPM	DBPM
28	BPM	BPM
29	VORP	VORP
30	FG	FG
31	FGA	FGA
32	FG%	FG%
33	_3P	_3P
34	_3PA	_3PA

[Get fields] [Enter field mapping]

[?] Help [Vale] [Cancelar] [SQL]

En la siguiente figura se muestra como quedaría la parte de la transformación IN\_STATS correspondiente a las estadísticas de la NBA.



## WNBA\_Seasons\_Stats

La otra parte de la transformación IN\_STATS\_NBA permite almacenar los datos del archivo de Excel “WNBA\_Seasons\_Stats\_2005\_2007” en su correspondiente tabla física de la base de datos.

Autoguardado ☐ WNBA\_Seasons\_Stats\_2005\_2017 - Modo de

Archivo Inicio Insertar Disposición de página Fórmulas Datos Revisión

Pegar Fuente Alineación

B4995 AVERY WARLEY-TALBERT

	A	B	C	D
1	ID	Data	Type	Stat
2	20171	2017	Number	SEASON
3	20171	1	Number	#
4	20171	TINA CHARLES	String	PLAYER
5	20171	NYL	String	TEAM
6	20171	29	Number	AGE
7	20171	34	Number	GP
8	20171	22	Number	W
9	20171	12	Number	L
10	20171	1095	Number	MIN
11	20171	671	Number	PTS
12	20171	262	Number	FGM
13	20171	593	Number	FGA
14	20171	44.2	Number	FG%



Este archivo posee la peculiaridad que su contenido más importante se encuentra normalizado dentro de la columna *Data* por lo que su tratamiento será diferente.

El primer paso es una Entrada Excel, de la que se obtienen los 4 campos que se muestran a continuación.

Entrada Excel

Nombre paso:

#	Nombre	Tipo	Longitud	Precisión	Tipo de poda	Repetir	Formato	Moneda	Decimal	Agrupamiento
1	ID	Integer	-1	-1	ninguno	N				
2	Data	String	-1	-1	ninguno	N				
3	Type	String	-1	-1	ninguno	N				
4	Stat	String	-1	-1	ninguno	N				

Obtener campos de cabecera...

Help Vale Previsualizar filas Cancelar

Luego será necesario desnormalizar la columna data para adecuar el formato de estos datos, para ello se utiliza un paso Des-Normalización de fila, en el cual se establece como campo clave *Stat*, como campo de agrupación el campo *ID*, y para obtener los campos deseados es necesario introducirlos con su respectivo Valor clave y su tipo. En la figura que se muestra a continuación se puede apreciar esta configuración.

Des-Normalizar

Nombre de paso: Des-Normalización de fila

El campo clave: Stat

Los campos que definen la agrupación:

#	Campo de agrupación
1	ID

Obtener campos

Campos resultado:

#	Nombre de campo destino	Valor de campo destino	Valor clave	Tipo	Formato	Longitud	Precis
1	SEASON	Data	SEASON	Integer			
2	#	Data	#	Integer			
3	PLAYER	Data	PLAYER	String			
4	TEAM	Data	TEAM	String			
5	AGE	Data	AGE	String			
6	GP	Data	GP	Integer			
7	W	Data	W	Integer			
8	L	Data	L	Integer			
9	MIN	Data	MIN	Integer			
10	PTS	Data	PTS	Number			
11	FGM	Data	FGM	Number			
12	FGA	Data	FGA	Integer			
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			
20	OREB	Data	OREB	Integer			
21	DREB	Data	DREB	Integer			
22	REB	Data	REB	Integer			
23	AST	Data	AST	Integer			
24	TOV	Data	TOV	Integer			
25	STL	Data	STL	Integer			
26	BLK	Data	BLK	Integer			
27	PF	Data	PF	Integer			
28	FP	Data	FP	Integer			
29	DD2	Data	DD2	Integer			
30	TD3	Data	TD3	Integer			
31	+/-	Data	+/-	Number			

Obtener campos búsqueda

Help Vale Cancelar

Como resultado de uno de los análisis anteriores se detectó que hay unos valores extraños en la variable AGE que impiden que se puedan convertir adecuadamente, por ello se utiliza un paso de Nulo si y se establece que el carácter “-” sea el sustituido.

Posteriormente se realizan las *String operations* para eliminar los espacios a los lados, en esta ocasión no es necesario transformar a mayúsculas porque los de tipo texto ya vienen con este formato.

String operations

Step name String operations 2

The fields to process:

#	In stream field	Out stream field	Trim type	Lower/Upper	Padding	Pad char	Pad Length	InitCap	Escape	Digits
1	Type		both	none	none			N	None	none
2	PLAYER		both	none	none			N	None	none
3	TEAM		both	none	none			N	None	none
4	AGE		both	none	none			N	None	none

Como paso previo a la carga en la base de datos se seleccionan los campos útiles para el modelo, se cambian los nombres de los campos que comienzan por números para que puedan ser tratados por el SQL Server y se transforman los datos del campo AGE que anteriormente no habían sido transformados.

Selección/Renombra valores

Nombre paso Selección/Renombra valores 2

Selección & Modifica					Eliminar	Meta-información
Campos :						
#	Nombre campo	Renombrar a	Longitud	Precisión		
1	SEASON					
2	ID					
3	PLAYER					
4	TEAM					
5	AGE					
6	GP					
7	W					
8	L					
9	MIN					
10	PTS					
11	FGM					
12	FGA					
13	FG%					
14	3PM	_3PM				
15	3PA	_3PA				
16	3P%	_3P%				
17	FTM					
18	FTA					
19	FT%					
20	OREB					
21	DREB					
22	REB					
23	AST					
24	TOV					
25	STL					
26	BLK					
27	PF					
28	FP					
29	DD2					
30	TD3					
31	+/-					

Obtener campos a seleccionar  
Edit Mapping

Finalmente se cargan los datos a la base de datos, el mapeo de los campos se muestra en la siguiente figura.

Salida de Tabla

Nombre paso: Salida Tabla

Conexión: PRA\_2 [Editar...] [Nuevo...] [Wizard...]

Esquema destino: dbo [Examinar...]

Tabla destino: STG\_WNBA\_SEASON\_STATS\_2005\_2017 [Examinar...]

Tamaño de transacción (commit): 1000

Vaciar tabla: ☒

Ignorar errores de inserción: ☐

Specify database fields: ☒

Main options | Database fields

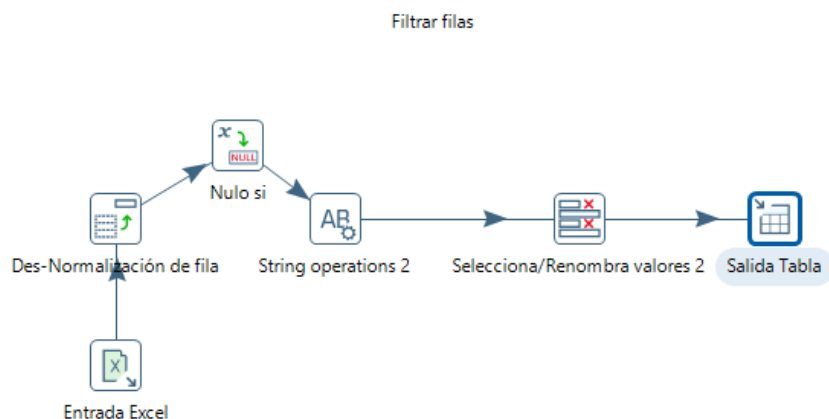
Fields to insert:

#	Table field	Stream field
1	SEASON	SEASON
2	ID	ID
3	PLAYER	PLAYER
4	TEAM	TEAM
5	AGE	AGE
6	GP	GP
7	W	W
8	L	L
9	MIN	MIN
10	PTS	PTS
11	FGM	FGM
12	FGA	FGA
13	FG%	FG%
14	_3PM	_3PM
15	_3PA	_3PA
16	_3P%	_3P%
17	FTM	FTM
18	FTA	FTA
19	FT%	FT%
20	OREB	OREB
21	DREB	DREB
22	REB	REB
23	AST	AST
24	TOV	TOV
25	STL	STL
26	BLK	BLK
27	PF	PF
28	FP	FP
29	DD2	DD2
30	TD3	TD3
31	+/-	+/-

[Get fields]

[Enter field mapping]

La estructura de esta parte de la transformación se expone a continuación.



## Transformación IN\_WNBA\_PLAYERS

De manera similar a la efectuada en pasos anteriores se comienza esta transformación visualizando el archivo original.

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

Como se puede apreciar, las columnas están separadas con un TAB y los nombres están divididos por una coma “,”. Se procede entonces a cargar estos datos con un *step* de Entrada Fichero de Texto.

En este paso se agrega el archivo “WNBA\_Players\_List.txt”, se establecen los tipos y longitudes de los campos y se le indica al asistente que el separador entre campos es el TAB. Como resultado se obtiene una tabla dividida en los campos *Player* y *Active*.

Entrada Fichero de Texto

Nombre pasoEntrada Fichero de Texto (WNBA\_Player\_List)

Fichero	Contenido	Manejo de Errores	Filtros	Campos	Additional output fields							
#	Nombre	Tipo	Formato	Posición	Longitud	Precisión	Moneda	Decimal	Grupo	Nulo si	Por defecto	
1	Player	String			255							r
2	Active	String			12							r

También se le aplica un *step* de tipo *string operation* para eliminar los espacios en blanco al final y al inicio, también se convierten a mayúsculas los registros del campo *Player*. En este caso no se ordenan los registros porque ya vienen ordenados alfabéticamente.

String operations

Step name: String operations (WNBA\_Players\_List)

The fields to process:

#	In stream field	Out stream field	Trim type	Lower/Upper	Padding	Pad char	Pad Length	InitCap	Escape	Digits
1	Active		both							
2	Player		both	upper						

Para finalizar esta transformación, se cargan los datos en la tabla STG\_WNBA\_PLAYERES.

Salida de Tabla

Nombre paso: Salida Tabla (WNBA\_Players)

Conexión: PRA\_2 [Editar...] [Nuevo...] [Wizard...]

Esquema destino: dbo [Examinar...]

Tabla destino: STG\_WNBA\_PLAYERES [Examinar...]

Tamaño de transacción (commit): 1000

Vaciar tabla: ☒

Ignorar errores de inserción: ☐

Specify database fields: ☒

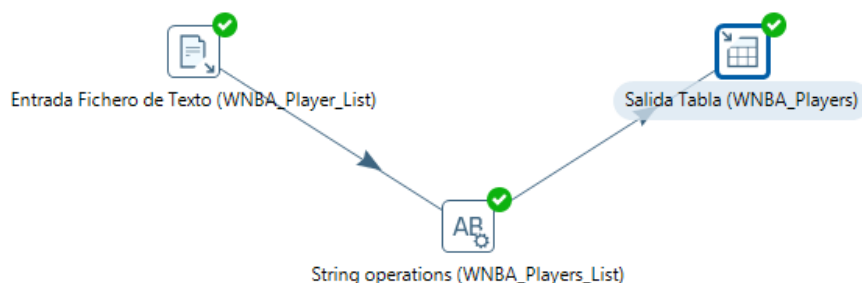
Main options Database fields

Fields to insert:

#	Table field	Stream field
1	Player	Player
2	Active	Active

[Get fields] [Enter field mapping]

Entonces la transformación IN\_WNBA\_PLAYERS en PDI quedaría de la siguiente manera.



## Transformación IN\_NBA\_PLAYERS

Tal y como se viene realizando hasta este momento, se comienza la transformación realizando la previsualización del contenido del archivo NBA\_Players\_List.xls.

	A
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

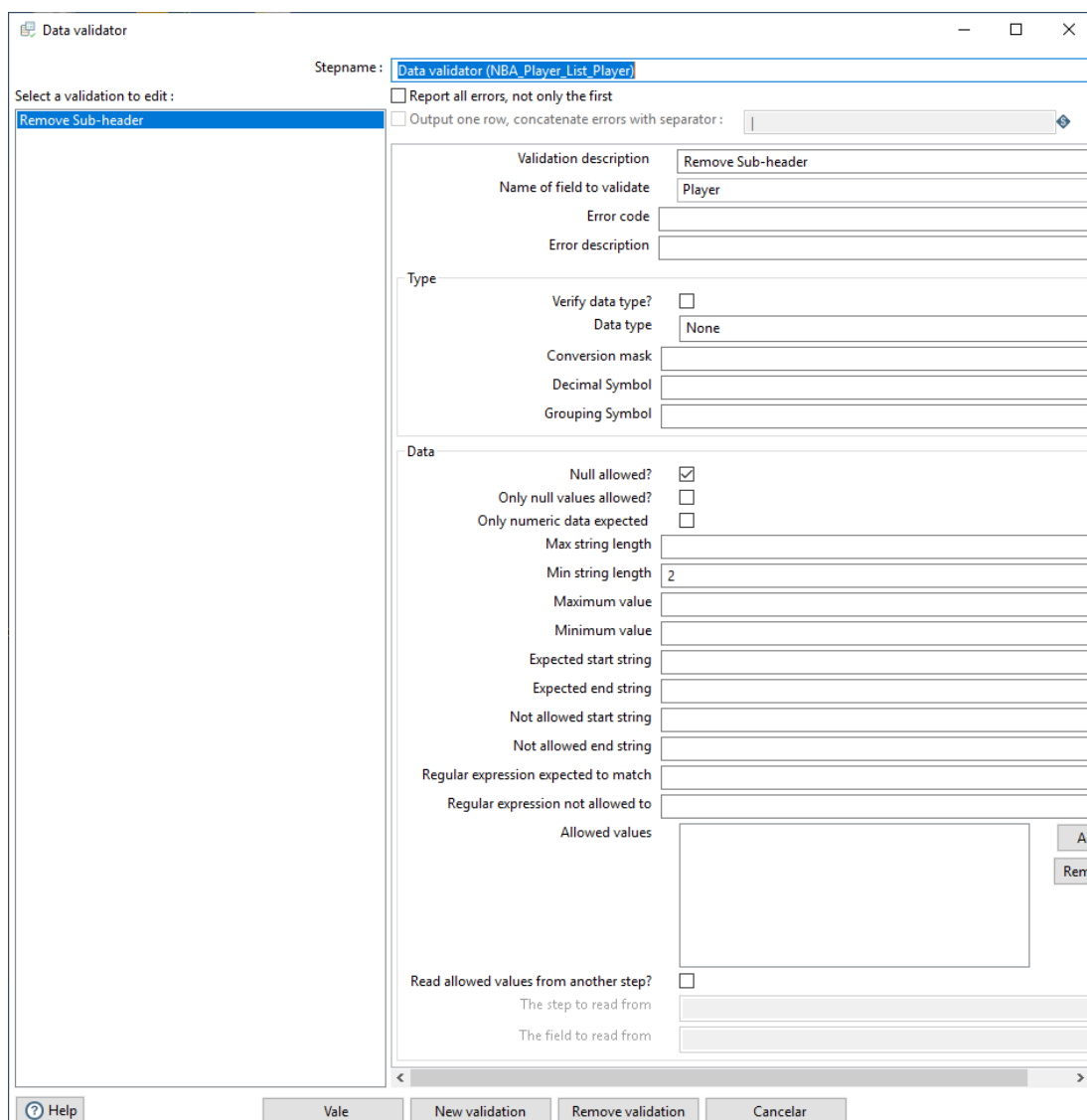
Esta transformación tiene una peculiaridad, el archivo que sirve de fuente de datos es de tipo xls y tiene dos hojas diferentes, en una los jugadores en activo y en la otra los jugadores retirados, también tiene una subcabecera indicando la inicial del nombre de cada jugador.

El primer paso para leer estos archivos es introducir dos *steps* del tipo Entrada Excel, con el objetivo de leer cada una de las hojas del archivo. Después de introducir la dirección del fichero en la pestaña Ficheros y añadirlo, es necesario seleccionar la hoja de cálculo que se desea leer. En este caso se ha seleccionado la que contiene los jugadores en activo (Player), para el caso de los retirados hay que repetir el mismo proceso pero seleccionando la hoja *Historic*. Para finalizar esta extracción solo es necesario definir el campo que albergará estos registros, esto se configura en la pestaña Campos asignándole el tipo y el tamaño adecuado (*string*, 255).



[illegible]

La segunda transformación es un validador de datos, que se encarga de eliminar las subcabeceras de las filas, para validar los datos se utiliza la propiedad *Min string length* a la cual se le establece un valor de 2 lo que excluye las subcabeceras que solo están formadas por un valor. Hay que destacar que se utilizan dos transformaciones simuladas para tratar los datos incorrectos, aunque no se realice ninguna operación con ellos.



The screenshot shows the 'Data validator' window with the following details:

- Stepname:** Data validator (NBA\_Player\_List\_Player)
- Select a validation to edit:** Remove Sub-header
- Report all errors, not only the first:** ☐
- Output one row, concatenate errors with separator:** |
- Validation description:** Remove Sub-header
- Name of field to validate:** Player
- Error code:**
- Error description:**
- Type:**
  - Verify data type?** ☐
  - Data type:** None
  - Conversion mask:**
  - Decimal Symbol:**
  - Grouping Symbol:**
- Data:**
  - Null allowed?** ☒
  - Only null values allowed?** ☐
  - Only numeric data expected?** ☐
  - Max string length:**
  - Min string length:** 2
  - Maximum value:**
  - Minimum value:**
  - Expected start string:**
  - Expected end string:**
  - Not allowed start string:**
  - Not allowed end string:**
  - Regular expression expected to match:**
  - Regular expression not allowed to:**
  - Allowed values:** (Empty list box)
  - Read allowed values from another step?** ☐
  - The step to read from:**
  - The field to read from:**

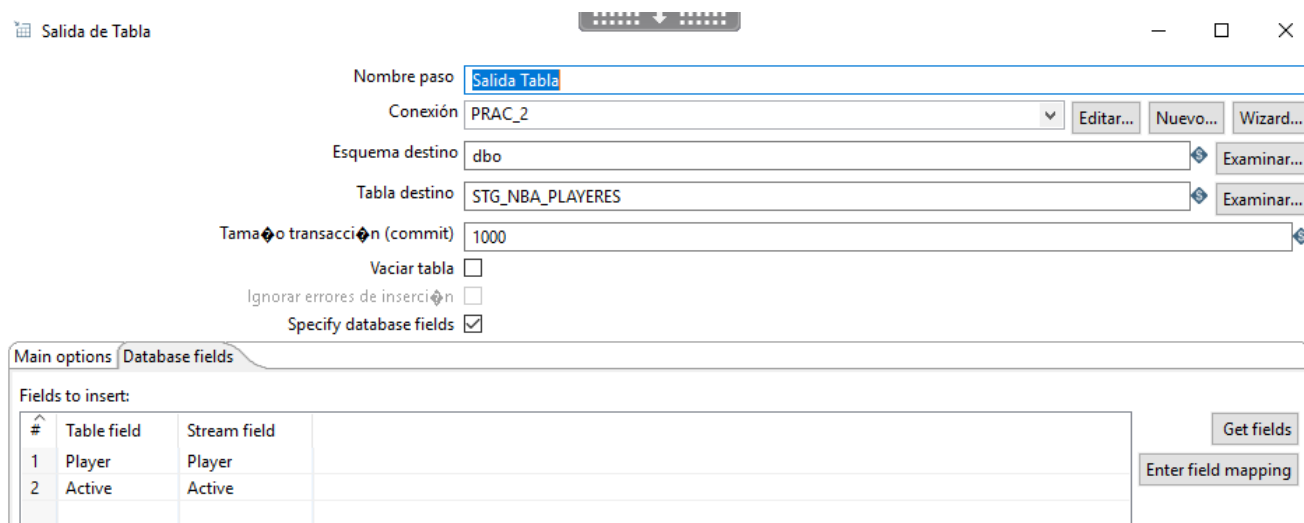
Buttons at the bottom: Help, Vale, New validation, Remove validation, Cancelar.

Para obtener la tabla en el formato deseado se agrega una columna constante a cada una de las tablas obtenidas de los validadores, de esta manera se identificarán que valores del campo *Player* está en activo (1) y cuales están retirados (0).

Para unificar los datos se utiliza una unión ordenada, en la que se irán intercalando de manera ordenada ambas tablas, sirviendo como campo clave *Players* y de esta manera se obtiene la tabla deseada con los campos que describen el nombre del jugador y si está en activo o no.

Posteriormente se eliminan los espacios a ambos lados de los *strings* y se convierten a mayúsculas, este procedimiento se realiza de la misma manera a la empleada en transformaciones anteriores.

Para finalizar se cargan los datos en la tabla física STG\_NBA\_PLAYERES utilizando una Salida Tabla y mapeando los campos *Player* y *Historic* con sus equivalentes en la tabla destino.



Salida de Tabla

Nombre paso: Salida Tabla

Conexión: PRAC\_2

Esquema destino: dbo

Tabla destino: STG\_NBA\_PLAYERES

Tamaño transacción (commit): 1000

Vaciar tabla: ☐

Ignorar errores de inserción: ☐

Specify database fields: ☒

Main options Database fields

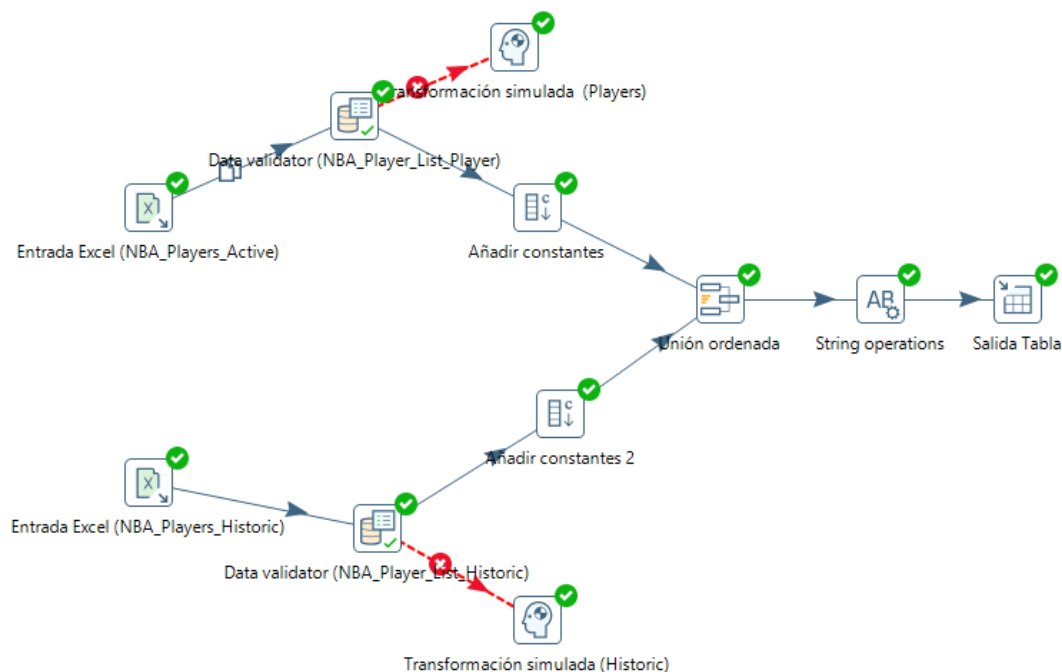
Fields to insert:

#	Table field	Stream field
1	Player	Player
2	Active	Active

Get fields

Enter field mapping

Al final esta transformación quedaría de la siguiente forma.



## Transformación IN\_TEAMS\_CODES

Para esta transformación se cuenta con un fichero de texto plano .txt, en este, están contenidos los diferentes equipos de ambas ligas y sus respectivos códigos. Para comenzar se utiliza un paso de Entrada de Fichero de texto que realizará la extracción inicial de los datos.



TeamCodes: Bloc de notes

Archivo Edición Formato Ver Ayuda

```
League#Team#Code
WNBA#Atlanta Dream#ADR
NBA#Atlanta Hawks#ATL
NBA#Boston Celtics#BOS
NBA#Brooklyn Nets#BN
NBA#Charlotte Hornets#CHA
NBA#Chicago Bulls#CHI
WNBA#Chicago Sky#CHS
NBA#Cleveland Cavaliers#CLE
WNBA#Connecticut Sun#CON
NBA#Dallas Mavericks#DAL
WNBA#Dallas Wings#DWN
```

En este paso hay que tener en consideración que el carácter separador en el archivo original es la almohadilla “#”, por lo demás la configuración no presenta complejidad alguna.

Entrada Fichero de Texto

Nombre paso: Entrada Fichero de Texto

Fichero | Contenido | Manejo de Errores | Filtros | Campos | Additional output fields

Tipo de fichero: CSV

Separador de campos: # Insert TAB

Separador de texto: "

Escape:

Cabecera: ☒ Número de líneas de cabecera: 1

Pie: ☐ Número de líneas de pie: 1

Líneas cortadas: ☐ Número de veces que se corta: 1

Paginado (impresión): ☐ Número de líneas por página: 80

Líneas en cabecera documento: 0

Comprimido (Zip): None

Eliminar filas vacías: ☒

Incluir nombre del fichero en salida: ☐ Campo con el nombre del fichero:

Número de fila en salida: ☐ Campo con el número de fila:

Número de fila por fichero: ☐

Formato: DOS

Codificación:

Length: Characters

Límite: 0

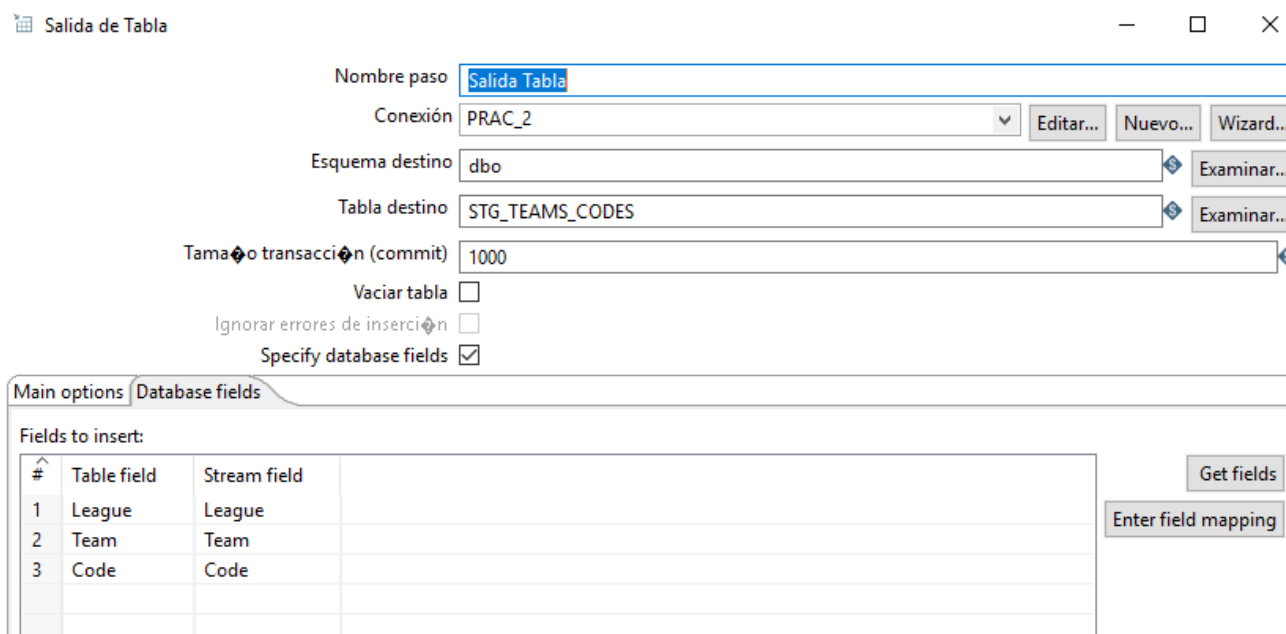
Ser flexible al leer fechas: ☒

Utilizar el formato de fecha local(e): es\_ES

Result filenames

Add filenames to result: ☒

Luego es necesario eliminar los espacios y convertir a mayúsculas los campos para posteriormente cargarlos en la tabla STG\_WNBA\_TEAMS de la base de datos.



Salida de Tabla

Nombre paso: Salida Tabla

Conexión: PRAC\_2 [Editar...] [Nuevo...] [Wizard...]

Esquema destino: dbo [Examinar...]

Tabla destino: STG\_TEAMS\_CODES [Examinar...]

Tamaño transacción (commit): 1000

Vaciar tabla: ☐

Ignorar errores de inserción: ☐

Specify database fields: ☒

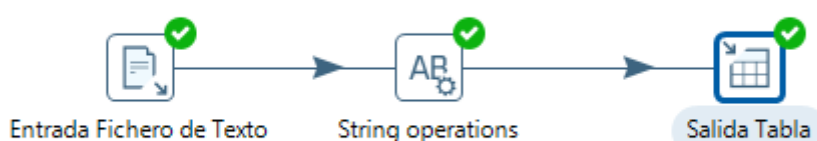
Main options | Database fields

Fields to insert:

#	Table field	Stream field
1	League	League
2	Team	Team
3	Code	Code

[Get fields] [Enter field mapping]

Tal y como se muestra en la figura, el mapeo de los campos es muy intuitivo, al ser pocos registros esta carga es relativamente sencilla. A continuación se muestra la estructura de esta transformación.



## Transformación IN\_NBA\_TEAMS

Para comenzar con esta transformación se observa el archivo original “NBA\_Teams.xml” con un editor de texto, de este análisis se detecta el XPath necesario para acceder a los datos y su estructuración. A continuación se muestra esta estructura, en la que los datos útiles están dentro de etiquetas <Data>.

```

1  <?xml version="1.0"?>
2  <NBATeams>
3    <Row>
4      <Data Type="String">Conferencia</Data>
5      <Data Type="String">Divisió</Data>
6      <Data Type="String">Equipo</Data>
7      <Data Type="String">Ciudad</Data>
8      <Data Type="String">Estado</Data>
9      <Data Type="String">Pabellón</Data>
10     <Data Type="String">Fundado</Data>
11     <Data Type="String">Patrocinio</Data>
12   </Row>
13   <Row>
14     <Data Type="String">Conferencia Oeste</Data>
15     <Data Type="String">Noroeste</Data>
16     <Data Type="String">Denver Nuggets</Data>
17     <Data Type="String">Denver</Data>
18     <Data Type="String">CO</Data>
19     <Data Type="String">Pepsi Center</Data>
20     <Data Type="Number">1967</Data>
21     <Data Type="String">Western Union</Data>
22   </Row>
23   <Row>
24     <Data Type="String">Conferencia Oeste</Data>
25     <Data Type="String">Noroeste</Data>
26     <Data Type="String">Minnesota Timberwolves</Data>
27     <Data Type="String">Minneapolis</Data>
28     <Data Type="String">MN</Data>
29     <Data Type="String">Target Center</Data>
30     <Data Type="Number">1989</Data>
31     <Data Type="String">Fitbit</Data>
32   </Row>
33   <Row>
34     <Data Type="String">Conferencia Oeste</Data>
35     <Data Type="String">Noroeste</Data>
36     <Data Type="String">Oklahoma City Thunder</Data>
37     <Data Type="String">Oklahoma City</Data>
38     <Data Type="String">OK</Data>

```

Una vez conocido el formato del archivo se procede a cargarlo a la herramienta, el paso seleccionado para esto es un Get data from xml. En este paso de manera similar a otros elementos de tipo input se introduce el archivo mediante su path, se establece el Loop XPath a “/NBATeams/\*” y se marca la pestaña para agregar un campo con el número de fila.

Get data from XML

Nombre paso: Get data from XML

File Content Fields Additional output fields

Settings

Loop XPath: /NBATeams/ Get XPath nodes

Encoding: UTF-8

Namespace aware? ☐

Ignore comments? ☐

Validate XML? ☐

Use token? ☐

Ignore empty file? ☐

Do not raise an error if no files? ☒

Limit: 0

Prune path to handle large files:

Additional fields

Include filename in output? ☐ Filename fieldname:

Rownum in output? ☒ Rownum fieldname: Rownum

Add to result filename

Add files to result filename? ☐

Help Vale Preview rows Cancelar

En el apartado de campos se establecen los nombres y los tipos de los campos deseados, y para acceder a ellos se establece un Xpath que sirve de guía para iterar sobre cada uno de los elementos del Loop XPath definido anteriormente.

Get data from XML

Nombre paso: Get data from XML

File Content Fields Additional output fields

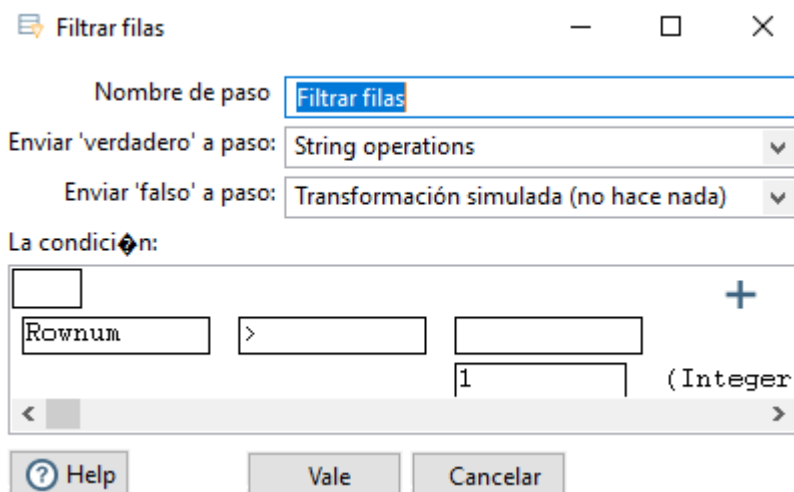
#	Name	XPath	Element	Result type	Type	Format	Length	Precision	Currency	Decimal	Group	Tr
1	Conferencia	Data[1]	Node	Value of	String							nc
2	División	Data[2]	Node	Value of	String							nc
3	Equipo	Data[3]	Node	Value of	String							nc
4	Ciudad	Data[4]	Node	Value of	String							nc
5	Estado	Data[5]	Node	Value of	String							nc
6	Pabellón	Data[6]	Node	Value of	String							nc
7	Fundado	Data[7]	Node	Value of	String							nc
8	Patrocinio	Data[8]	Node	Value of	String							nc

Get fields

Help Vale Preview rows Cancelar

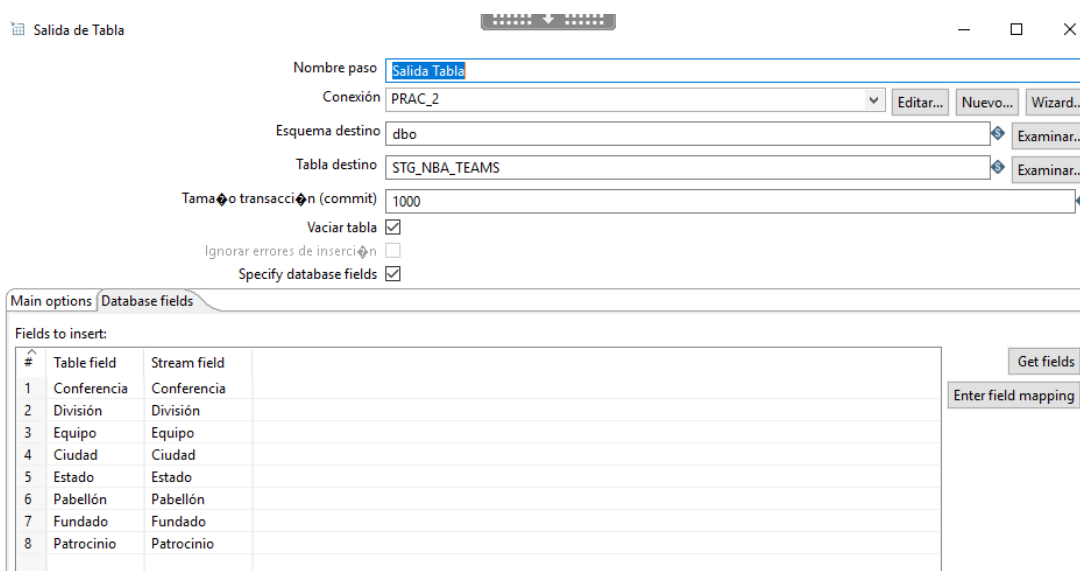


Luego de terminar la extracción se filtran las filas para eliminar la primera, que contiene los nombres de los campos, esto se realiza filtrando por la columna Rownum agregada en el paso de carga.

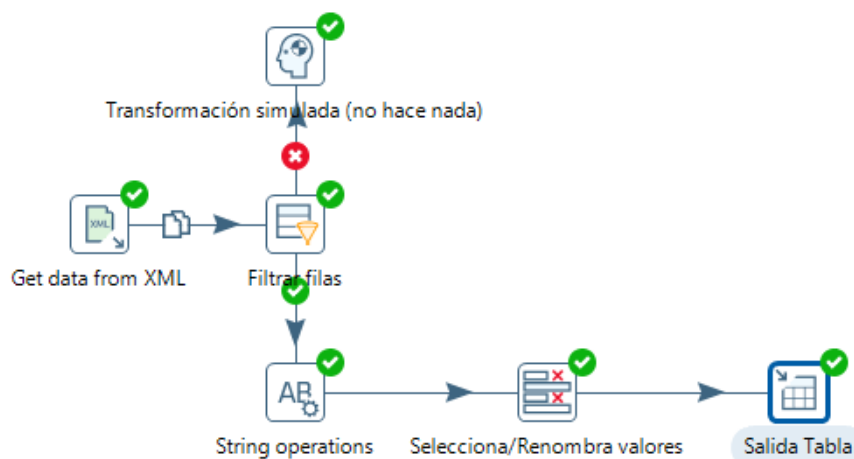


Una vez se cuenta con los datos útiles, se realizan las tareas de eliminar espacios y convertir a mayúsculas. Posteriormente se elimina la columna Rowname y se cambia el tipo al campo Fundado con un step Selecciona/Renombrar valores.

Finalmente se carga a la base de datos correspondiente con el mapeo de campos siguiente.



La transformación terminada presenta la siguiente estructura.



## Transformación IN\_WNBA\_TEAMS

Esta transformación se encarga de cargar los datos del archivo “WNBA\_Teams.xls” a la base de datos STG\_WNBA\_TEAMS. Tal y como se viene haciendo con las anteriores transformaciones, el primer paso es obtener una visión panorámica del archivo que hace de origen de datos, en este caso el archivo de Excel solo tiene una hoja de cálculo y pocos datos.

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

Teniendo en cuenta la información extraída del análisis de las fuentes de datos y conociendo la estructura del modelo físico de la tabla STG\_WNBA\_TEAMS, se procede a eliminar manualmente los indicadores a la izquierda de los nombres de los equipos. Esta tarea facilita enormemente la extracción de estos datos, además que estos valores no aportan ningún valor a la tabla final. A continuación se muestra como quedaría la tabla con la modificación.

NOTA: Independientemente de que no sea una buena práctica modificar los archivos manualmente, esta es una opción que en esta ocasión facilita enormemente la transacción.

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

El primer paso en esta transformación es una Entrada Excel (Entrada Excel (WNBA\_TEAMS)). Luego se introduce el nombre del archivo gracias a la variable de entorno DIR\_ENT ,se selecciona su única hoja de cálculo y se establecen los campos que se desea obtener. A continuación se muestra como queda esta configuración.

Entrada Excel

Nombre paso: Entrada Excel (WNBA\_TEAMS)

Ficheros	Hojas	Contenido	Manejador de Error	Campos	Additional output fields					
#	Nombre	Tipo	Longitud	Precisión	Tipo de poda	Repetir	Formato	Moneda	Decimal	Agrupamiento
1	Conferencia	String	-1	-1	ninguno	S				
2	Equipo	None	-1	-1	ninguno	N				
3	Pts	None	-1	-1	ninguno	N				
4	J	None	-1	-1	ninguno	N				
5	G	None	-1	-1	ninguno	N				
6	P	None	-1	-1	ninguno	N				
7	p.	None	-1	-1	ninguno	N				
8	c.	None	-1	-1	ninguno	N				
9	dif	None	-1	-1	ninguno	N				
10	Ciudad, Estado	None	-1	-1	ninguno	N				

Obtener campos de cabecera...

Help Vale Previsualizar filas Cancelar

El siguiente paso consiste en eliminar algunos campos de los datos, en este caso se eliminarán los campos Pts, J, G, P, p., c. y dif, se utiliza el conocido step Selecciona/renombrar valores.

Selecciona/Renombrar valores

Nombre paso: Selecciona/Renombrar valores

Selecciona & Modifica Eliminar Meta-información

Campos a eliminar:

#	Nombre campo
1	Pts
2	J
3	G
4	P
5	p.
6	c.
7	dif

A continuación se filtran las filas para eliminar los registros nulos y las cabeceras (con regex) y se procede entonces a dividir uno de los campos. En este caso se divide el campo "Ciudad, Estado" para que coincida con el formato del diseño físico, para ello se utiliza el paso Partir campo, seleccionando como separador la coma "," y eligiendo un método de trim a ambos lados para eliminar espacios indeseados.

Partir campo

Nombre paso:

Campo a partir:

Separador:

Enclosure:

Campos

#	Nuevo campo	ID	Eliminar ID?	Tipo	Longitud	Precisión	Formato	Grupo	Decimal	Moneda	Nulo Si	D
1	Ciudad		N	String								
2	Estado		N	String								

Como pasos anteriores a la carga se agrega una nueva columna que hará de identificador (ID), para ello se utiliza el paso Añadir secuencia comenzando en 1 con un salto igualmente con valor a 1, además se realizan las operaciones de los campos de tipo texto que eliminan los espacios y transforman a mayúsculas.

STG\_WNBA\_TEAMS

Obtener valor de la secuencia de la base de datos

Nombre de paso:

Nombre de valor:

Utilizar una base de datos para generar la secuencia

¿Utilizar base de datos para obtener secuencia? ☐

Conexión:

Nombre de esquema:

Nombre de secuencia:

Utilizar un contador de la transformación para generar la secuencia

¿Utilizar contador para calcular secuencia? ☒

Nombre contador (opcional):

Valor inicial:

Incremento:

Valor máximo:

Para finalizar se cargan los datos a la tabla STG\_WNBA\_TEAMS mapeando los campos de la transformación con los del modelo.

Salida de Tabla

Nombre paso: Salida Tabla

Conexión: PRAC\_2

Esquema destino: dbo

Tabla destino: STG\_WNBA\_TEAMS

Tamaño de transacción (commit): 1000

Vaciar tabla: ☒

Ignorar errores de inserción: ☐

Specify database fields: ☒

Main options Database fields

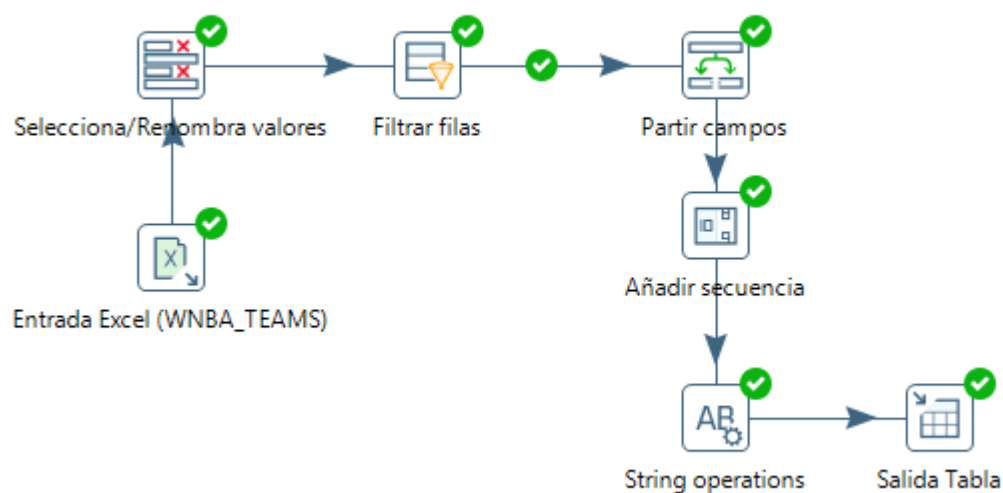
Fields to insert:

#	Table field	Stream field
1	Conferencia	Conferencia
2	Equipo	Equipo
3	Ciudad	Ciudad
4	Estado	Estado
5	ID	ID

Get fields

Enter field mapping

En la siguiente figura se muestra la estructura de esta transformación.



## Transformación IN\_STATES

Esta transformación es la encargada de extraer los datos del archivo “Estados\_Unidos.json” y cargarlos en la tabla STG\_STATES de la base de datos. Como se ha realizado en los apartados anteriores se comienza por obtener una panorámica del archivo original.



```

1  {
2    "EstadosUnidos": [
3      {
4        "Estado": "Alabama",
5        "Nombre oficial": "State of Alabama",
6        "Superficie (km2)": "135756",
7        "Abrev.": "AL",
8        "Ingreso en la Unión": "14-12-1819",
9        "Población (2010)": "4779736",
10       "Densidad de Población (hab/km2)": "35.21",
11       "Capital": "Montgomery",
12       "Ciudad con mayor Población (2006)": "Birmingham"
13     },
14     {
15       "Estado": "Alaska",
16       "Nombre oficial": "State of Alaska",
17       "Superficie (km2)": "1717854",
18       "Abrev.": "AK",
19       "Ingreso en la Unión": "03-01-1959",
20       "Población (2010)": "710231",
21       "Densidad de Población (hab/km2)": "0.41",
22       "Capital": "Juneau",
23       "Ciudad con mayor Población (2006)": "Anchorage"
24     },
25   ]
26 }
  
```

Se puede apreciar que el archivo presenta un formato aparentemente sencillo, aunque los nombres de las variables tienen espacios, lo que dificulta su lectura.

Para realizar esta transformación el primer paso será su extracción, que se implementa con un step JSON Input. Luego de introducir la dirección del archivo y seleccionar los campos que se desean extraer, la tarea más importante en este paso es modificar los nombres de los campos, más específicamente se deben evitar espacios y signos de puntuación. Para evitar estos elementos es necesario incluir el nombre entre corchetes y entre comillas simples ej: ['nombre del campo'], de esta manera no saltan errores debidos a estos caracteres mal identificados. En la siguiente figura se muestra dicha configuración.

JSON input

Nombre paso:

#	Name	Path	Type	Format	Length	Preci
1	Estado	\$.[*].['Estado']	String		255	
2	Nombre oficial	\$.[*].['Nombre oficial']	String		100	
3	Superficie (km2)	\$.[*].['Superficie (km2)']	Integer			
4	Abrev.	\$.[*].['Abrev.']	String		50	
5	Ingreso en la Unión	\$.[*].['Ingreso en la Unión']	String			
6	Población (2010)	\$.[*].['Población (2010)']	Integer		0	
7	Densidad de Población (hab/km2)	\$.[*].['Densidad de Población (hab/km2)']	Number			
8	Capital	\$.[*].['Capital']	String		255	
9	Ciudad con mayor Población (2006)	\$.[*].['Ciudad con mayor Población (2006)']	String			

Select fields

Help Vale Preview rows Cancelar

Luego se filtran las filas para que no pase ninguna con valores nulos en el campo de Estado y se eliminan los campos que no son necesarios para cargar en la base de datos.

Selección/Renombra valores

Nombre paso:

Selección & Modifica Eliminar Meta-información

Campos a eliminar:

#	Nombre campo
1	Ingreso en la Unión
2	Ciudad con mayor Población (2006)

Como último paso antes de cargar los valores a la tabla STG\_STATES se eliminan los espacios a los lados y se convierten a mayúsculas los campos de tipo texto. Entonces con una salida de tipo tabla se procede a cargar los datos, a continuación se muestra el mapeado de los campos de ambas entidades.



Salida de Tabla

Nombre paso: Salida Tabla

Conexión: PRAC\_2 [Editar...] [Nuevo...] [Wizard...]

Esquema destino: dbo [Examinar...]

Tabla destino: STG\_STATES [Examinar...]

Tamaño transacción (commit): 1000

Vaciar tabla: ☐

Ignorar errores de inserción: ☐

Specify database fields: ☒

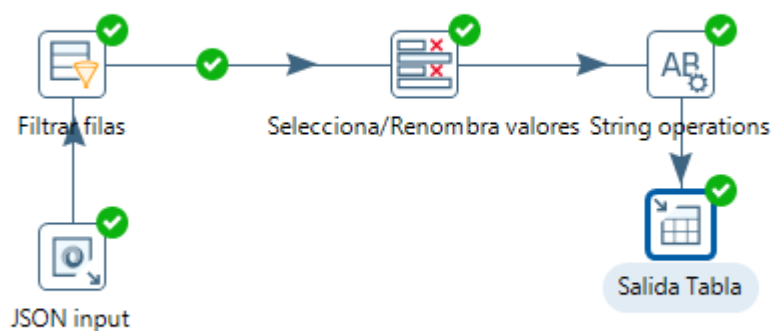
Main options Database fields

Fields to insert:

#	Table field	Stream field
1	Estado	Estado
2	Nombre ofi...	Nombre oficial
3	Superficie (...)	Superficie (k...
4	Abrev.	Abrev.
5	Población (...)	Población (20...
6	Densidad d...	Densidad de ...
7	Capital	Capital

[Get fields] [Enter field mapping]

Esta transformación quedaría estructurada de la siguiente manera.



## Transformación IN\_FREE\_THROWS

En esta transformación se cargan los datos relacionados con los tiros libres en la tabla STG\_FREE\_THROWS, es una transformación sencilla que solamente realizará la extracción del archivo csv, elimina los espacios de manera similar a los pasos anteriores y se cargan en su tabla correspondiente. A continuación se muestra como luce el archivo originalmente.

```
nba_free_throws.csv
1 end_result,game,game_id,period,play,player,playoffs,score,season,shot_made,time
2 106 - 114,PHX - LAL,261031013.0,1.0,Andrew Bynum makes free throw 1 of 2,Andrew Bynum,regular,0 - 1,2006 - 2007,1,11:45
3 106 - 114,PHX - LAL,261031013.0,1.0,Andrew Bynum makes free throw 2 of 2,Andrew Bynum,regular,0 - 2,2006 - 2007,1,11:45
4 106 - 114,PHX - LAL,261031013.0,1.0,Andrew Bynum makes free throw 1 of 2,Andrew Bynum,regular,18 - 12,2006 - 2007,1,7:26
5 106 - 114,PHX - LAL,261031013.0,1.0,Andrew Bynum misses free throw 2 of 2,Andrew Bynum,regular,18 - 12,2006 - 2007,0,7:26
6 106 - 114,PHX - LAL,261031013.0,1.0,Shawn Marion makes free throw 1 of 1,Shawn Marion,regular,21 - 12,2006 - 2007,1,7:18
7 106 - 114,PHX - LAL,261031013.0,1.0,Amare Stoudemire makes free throw 1 of 2,Amare Stoudemire,regular,33 - 20,2006 - 2007,1,3:15
8 106 - 114,PHX - LAL,261031013.0,1.0,Amare Stoudemire makes free throw 2 of 2,Amare Stoudemire,regular,34 - 20,2006 - 2007,1,3:15
9 106 - 114,PHX - LAL,261031013.0,2.0,Leandro Barbosa misses free throw 1 of 2,Leandro Barbosa,regular,43 - 29,2006 - 2007,0,10:52
10 106 - 114,PHX - LAL,261031013.0,2.0,Leandro Barbosa makes free throw 2 of 2,Leandro Barbosa,regular,44 - 29,2006 - 2007,1,10:52
11 106 - 114,PHX - LAL,261031013.0,2.0,Lamar Odom makes free throw 1 of 2,Lamar Odom,regular,44 - 30,2006 - 2007,1,10:37
12 106 - 114,PHX - LAL,261031013.0,2.0,Lamar Odom makes free throw 2 of 2,Lamar Odom,regular,44 - 31,2006 - 2007,1,10:37
13 106 - 114,PHX - LAL,261031013.0,2.0,Lamar Odom makes free throw 1 of 1,Lamar Odom,regular,46 - 34,2006 - 2007,1,8:47
14 106 - 114,PHX - LAL,261031013.0,2.0,Smush Parker misses free throw 1 of 2,Smush Parker,regular,46 - 42,2006 - 2007,0,5:20
15 106 - 114,PHX - LAL,261031013.0,2.0,Smush Parker misses free throw 2 of 2,Smush Parker,regular,46 - 42,2006 - 2007,0,5:20
16 106 - 114,PHX - LAL,261031013.0,2.0,Vladimir Radmanovic misses free throw 1 of 2,Vladimir Radmanovic,regular,48 - 42,2006 - 2007,0,4:50
```

El paso de CSV File Input no presenta mayor complejidad, prácticamente auto detecta los tipos y las dimensiones de los campos, a continuación se muestra este paso.

CSV file input

Step name

CSV file input

Filename

{DIR\_ENT}\nba\_free\_throws.csv

Examinar...

Delimiter

,

Insert TAB

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	end_result	String		9		€	,	.	ninguno
2	game	String		9		€	,	.	ninguno
3	game_id	Integer	#	10	0	€	.	.	ninguno
4	period	Integer	#	4	0	€	.	.	ninguno
5	play	String		44		€	,	.	ninguno
6	player	String		19		€	,	.	ninguno
7	playoffs	String		7		€	,	.	ninguno
8	score	String		9		€	,	.	ninguno
9	season	String		11		€	,	.	ninguno

Help

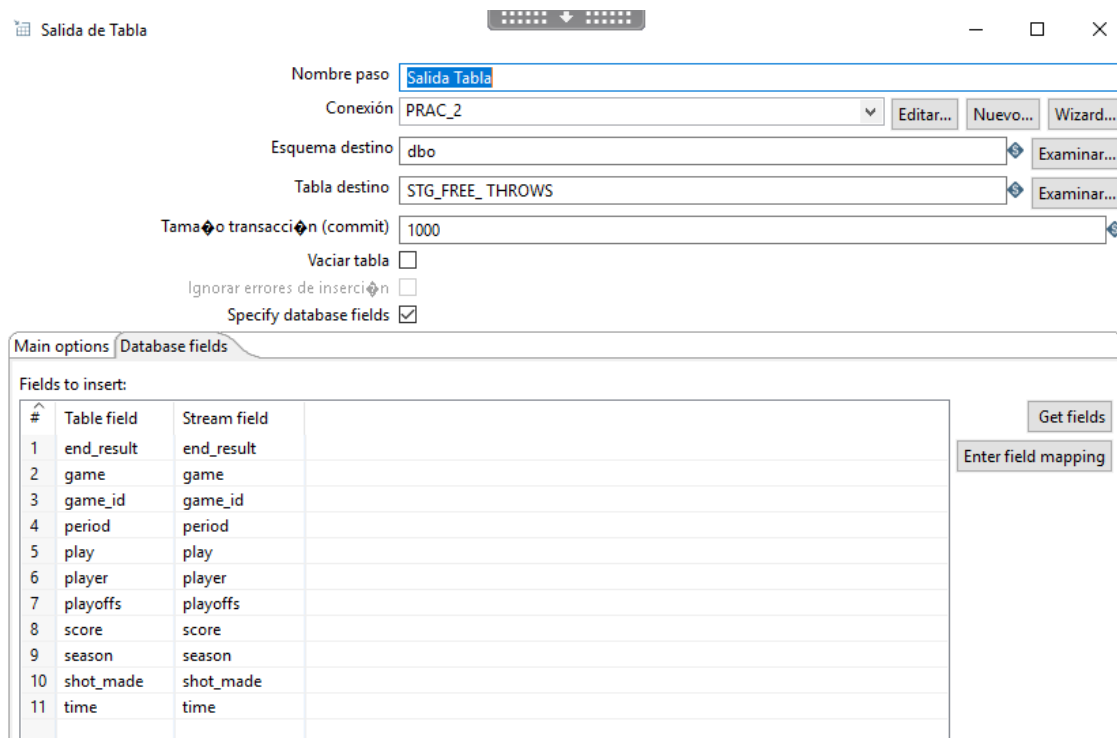
Vale

Traer Campos

Previsualizar

Cancelar

Después de realizar las string operations ya los datos están listos para ser cargados en la base de datos, la configuración del mapeo de los campos se muestra a continuación.



Salida de Tabla

Nombre paso: Salida Tabla

Conexión: PRAC\_2 [Editar... Nuevo... Wizard...]

Esquema destino: dbo [Examinar...]

Tabla destino: STG\_FREE\_THROWS [Examinar...]

Tamaño transacción (commit): 1000

Vaciar tabla: ☐

Ignorar errores de inserción: ☐

Specify database fields: ☒

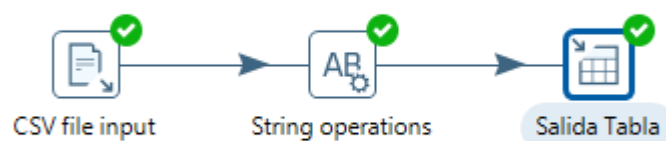
Main options | Database fields

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]

La transformación en su conjunto presenta la siguiente estructura.



## Transformación IN\_PLAYERS\_DATA

En esta transformación se utiliza como fuente de datos el fichero “NBA\_players\_data.json” el cual será procesado y cargado en la tabla STG\_PLAYERS\_DATA de la base de datos. Tal y como se ha venido implementando el primer paso es visualizar el archivo en su formato original para obtener una panorámica de su formato y estructura.

```

1  {
2    "players":
3    [
4      {"id": "abdulal01", "birthDate": "06/24/1968", "birthDate_str": "24/06/1968", "birthPlace": "Cairo", "State_Country": "Egypt", "career_AST": "3", "career_FG1": "502", "career_FG2": "12", "career_FG3": "426", "career_FG4": "559", "career_FG5": "44", "career_FG6": "492", "career_FG7": "237", "career_FG8": "370", "career_FG9": "222", "career_FG10": "501"},
5      {"id": "abdulke01", "birthDate": "04/07/1946", "birthDate_str": "07/04/1946", "birthPlace": "Brooklyn", "State_Country": "New York", "career_AST": "12", "career_FG1": "426", "career_FG2": "559", "career_FG3": "44", "career_FG4": "492", "career_FG5": "237", "career_FG6": "370", "career_FG7": "222", "career_FG8": "501"},
6      {"id": "abdulke02", "birthDate": "04/16/1947", "birthDate_str": "16/04/1947", "birthPlace": "New York", "State_Country": "New York", "career_AST": "36", "career_FG1": "559", "career_FG2": "44", "career_FG3": "492", "career_FG4": "237", "career_FG5": "370", "career_FG6": "222", "career_FG7": "501"},
7      {"id": "abdulma02", "birthDate": "03/09/1969", "birthDate_str": "09/03/1969", "birthPlace": "Gulfport", "State_Country": "Mississippi", "career_AST": "35", "career_FG1": "44", "career_FG2": "492", "career_FG3": "237", "career_FG4": "370", "career_FG5": "222", "career_FG6": "501"},
8      {"id": "abdulth01", "birthDate": "11/03/1974", "birthDate_str": "03/11/1974", "birthPlace": "Maisons Alfort", "State_Country": "France", "career_AST": "11", "career_FG1": "44", "career_FG2": "492", "career_FG3": "237", "career_FG4": "370", "career_FG5": "222", "career_FG6": "501"},
9      {"id": "abdurth01", "birthDate": "12/11/1976", "birthDate_str": "11/12/1976", "birthPlace": "Marietta", "State_Country": "Georgia", "career_AST": "25", "career_FG1": "472", "career_FG2": "492", "career_FG3": "237", "career_FG4": "370", "career_FG5": "222", "career_FG6": "501"},
10     {"id": "abernth01", "birthDate": "05/06/1954", "birthDate_str": "06/05/1954", "birthPlace": "South Bend", "State_Country": "Indiana", "career_AST": "12", "career_FG1": "492", "career_FG2": "237", "career_FG3": "370", "career_FG4": "222", "career_FG5": "501"},
11     {"id": "ablefo01", "birthDate": "07/27/1932", "birthDate_str": "27/07/1932", "birthPlace": "NULL", "State_Country": "NULL", "career_AST": "10", "career_FG1": "0", "career_FG2": "237", "career_FG3": "370", "career_FG4": "222", "career_FG5": "501"},
12     {"id": "abramjo01", "birthDate": "02/09/1919", "birthDate_str": "09/02/1919", "birthPlace": "Etna", "State_Country": "Pennsylvania", "career_AST": "7", "career_FG1": "237", "career_FG2": "370", "career_FG3": "222", "career_FG4": "501"},
13     {"id": "abramal01", "birthDate": "08/01/1993", "birthDate_str": "01/08/1993", "birthPlace": "Palma de Mallorca", "State_Country": "Spain", "career_AST": "5", "career_FG1": "237", "career_FG2": "370", "career_FG3": "222", "career_FG4": "501"},
14     {"id": "ackeral01", "birthDate": "01/21/1983", "birthDate_str": "21/01/1983", "birthPlace": "Compton", "State_Country": "California", "career_AST": "5", "career_FG1": "370", "career_FG2": "222", "career_FG3": "501"},
15     {"id": "ackerdo01", "birthDate": "09/04/1930", "birthDate_str": "04/09/1930", "birthPlace": "New York", "State_Country": "New York", "career_AST": "8", "career_FG1": "222", "career_FG2": "501"},
16     {"id": "ackresma01", "birthDate": "11/15/1962", "birthDate_str": "15/11/1962", "birthPlace": "Inglewood", "State_Country": "California", "career_AST": "5", "career_FG1": "501"},
17     {"id": "actonbu01", "birthDate": "01/11/1942", "birthDate_str": "11/01/1942", "birthPlace": "NULL", "State_Country": "NULL", "career_AST": "5", "career_FG1": "392", "career_FG2": "501"}
18   ]
19 }

```

Se puede apreciar en esta imagen que el archivo cuenta con valores nulos, establecidos con la cadena de caracteres “NULL”, esto en JSON es un problema, ya que detecta los valores nulos con la cadena “null”, por lo cual será necesario trabajar con estos valores para lograr una correcta transformación.

Para realizar la extracción del archivo se utiliza un paso JSON input, luego de cargar la dirección del archivo se seleccionan los campos que se desean extraer, inicialmente se marcan todos de tipo string, esto viene determinado por el hecho que al tener valores marcados como “NULL”, los valores numéricos lanzarán un error al intentar convertir. Estos valores serán adecuados al formato correcto para su tratamiento.

JSON input


Nombre paso
JSON input

#	Name	Path	Type	Format	Length	Precision	Currency	Decimal	Group
1	id	\$.[*].id	String						
2	birthDate	\$.[*].birthDate	String						
3	birthDate_str	\$.[*].birthDate_str	String						
4	birthPlace	\$.[*].birthPlace	String						
5	State_Country	\$.[*].State_Country	String						
6	career_AST	\$.[*].career_AST	String					.	,
7	career_FG%	\$.[*].career_FG%	String					.	,
8	career_FG3%	\$.[*].career_FG3%	String					.	,
9	career_FT%	\$.[*].career_FT%	String					.	,
10	career_G	\$.[*].career_G	String					.	,
11	career_PER	\$.[*].career_PER	String					.	,
12	career_PTS	\$.[*].career_PTS	String					.	,
13	career_TRB	\$.[*].career_TRB	String					.	,
14	career_WS	\$.[*].career_WS	String					.	,
15	career_eFG%	\$.[*].career_eFG%	String					.	,
16	college	\$.[*].college	String						
17	draft_pick	\$.[*].draft_pick	String						
18	draft_round	\$.[*].draft_round	String						
19	draft_team	\$.[*].draft_team	String						
20	draft_year	\$.[*].draft_year	String						
21	height	\$.[*].height	String						
22	highSchool	\$.[*].highSchool	String						
23	name	\$.[*].name	String						
24	position	\$.[*].position	String						
25	shoots	\$.[*].shoots	String						
26	weight	\$.[*].weight	String						

Select fields

? Help
Vale
Preview rows
Cancelar

Para tratar los valores nulos adecuadamente es necesario identificarlos, para esto se utiliza un paso Nulo si, que marca como nulo los valores de la tabla que coincidan con la cadena "NULL".

 Nulo si

Nombre de paso :

Campos:

#	Nombre	Valor a convertir en NULO
1	id	NULL
2	birthDate	NULL
3	birthDate_str	NULL
4	birthPlace	NULL
5	State_Country	NULL
6	career_AST	NULL
7	career_FG%	NULL
8	career_FG3%	NULL
9	career_FT%	NULL
10	career_G	NULL
11	career_PER	NULL
12	career PTS	NULL
13	career_TRB	NULL
14	career_WS	NULL
15	career_eFG%	NULL
16	college	NULL
17	draft_pick	NULL
18	draft_round	NULL
19	draft_team	NULL
20	draft_year	NULL
21	height	NULL
22	highSchool	NULL
23	name	NULL
24	position	NULL
25	shoots	NULL
26	weight	NULL

Una vez identificados correctamente los valores nulos, se procede a transformar los campos a su correcto tipo, en la figura que se muestra a continuación se muestran las conversiones pertinentes utilizando un paso de Seleccionar/Renombrar. También se seleccionaron los campos necesarios para cargar en el modelo, descartando entre otros los campos relacionados con *the draft*.

Selecciona/Renombra valores



— □ ×

Nombre paso Selecciona/Renombra valores

Selecciona & Modifica Eliminar Meta-información

Campos a modificar meta información:

#	Nombre campo	Renombrar a	Tipo	Longitud	Precisión	Binary to Normal?	Format	Date F
1	career_AST		Number			N		N
2	career_FG%		Number			N	#,##	N
3	career_FG3%		Number			N		N
4	career_FT%		Number			N		N
5	career_G		Number			N		N
6	career_PER		Number			N		N
7	career PTS		Number			N		N
8	career_TRB		Number			N		N
9	career_WS		Number			N		N
10	career_eFG%		Number			N		N

Posteriormente se aplican las operaciones de los *strings* de eliminar los espacios y se procede a la carga de los datos a la base de datos.

Salida de Tabla



— □ ×

Nombre paso Salida Tabla

Conexión PRAC\_2

Editar... Nuevo... Wizard...

Esquema destino dbo

Examinar...

Tabla destino STG\_PLAYERS\_DATA

Examinar...

Tamaño transacción (commit) 1000

Vaciar tabla ☒

Ignorar errores de inserción ☐

Specify database fields ☒

Main options Database fields

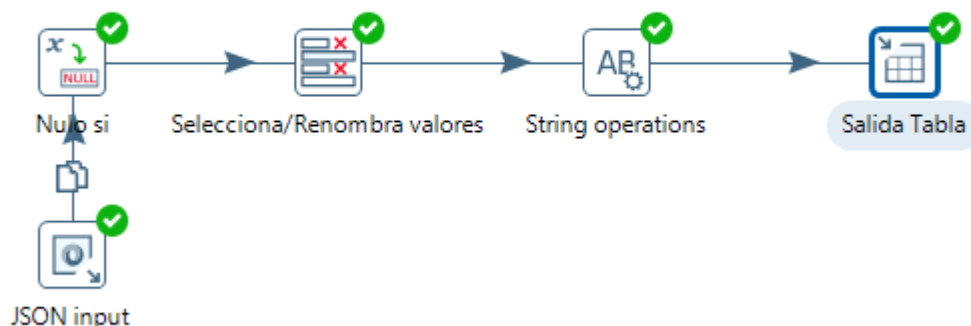
Fields to insert:

#	Table field	Stream field
1	id	id
2	birthDate	birthDate
3	birthPlace	birthPlace
4	career_AST	career_AST
5	career_FG%	career_FG%
6	career_FG3%	career_FG3%
7	career_FT%	career_FT%
8	career_G	career_G
9	career_PER	career_PER
10	career PTS	career PTS
11	career_TRB	career_TRB
12	career_WS	career_WS
13	career_eFG%	career_eFG%
14	college	college
15	height	height
16	highSchool	highSchool
17	name	name
18	position	position
19	shoots	shoots
20	weight	weight

Get fields

Enter field mapping

La estructura final de esta transformación se muestra en la siguiente figura.



## Procesos ETL (Bloque TR)

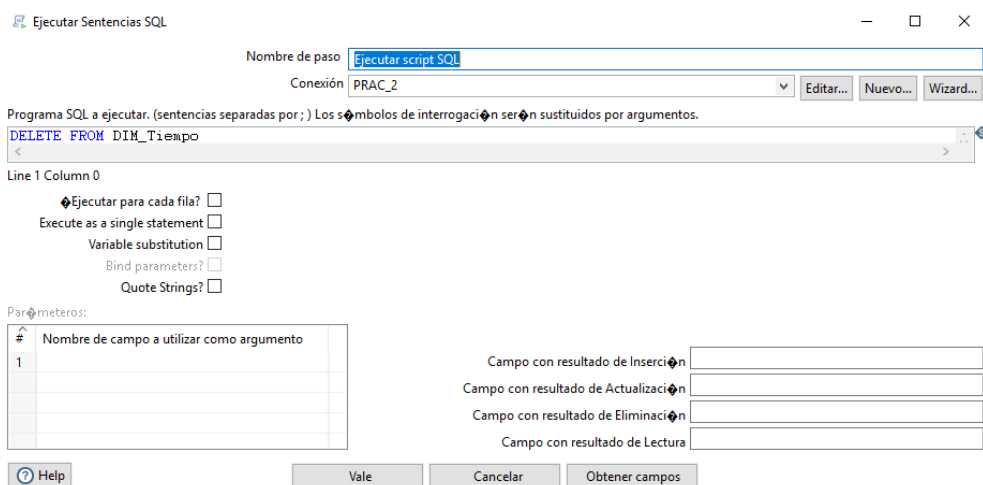
El bloque TR contiene los procesos ETL para la carga de datos al modelo multidimensional del almacén que hemos diseñado, compuesto por dimensiones y tablas de hechos, desde las tablas intermedias pobladas con los procesos del bloque IN.

### Transformaciones bloque TR\_DIM\_ Tiempo

Este bloque contiene las transformaciones necesarias para realizar la carga inicial de la dimensión Tiempo en el almacén, esta carga se realiza desde las tablas intermedias “STG\_WNBA\_SEASON\_STATS\_2005\_2017” y “STG\_NBA\_SEASON\_STATS\_1950\_2017” de las que se tomarán solamente una porción de los datos.

El primer paso de cada una de estas transformaciones es la eliminación del contenido de la tabla, esto facilitará el trabajo de la carga inicial, ya que estas tareas pueden necesitar ser ejecutadas a modo de prueba en múltiples ocasiones. Esta tarea se implementa gracias a un paso Ejecutar Sentencias SQL, al cual se le pasará una query **DELETE FROM “Nombre de la Tabla”**, a continuación se muestra este paso y para futuras implementaciones de procedimientos similares solamente se mencionarán.





Ejecutar Sentencias SQL

Nombre de paso:

Conexión:

Programa SQL a ejecutar. (sentencias separadas por ; ) Los símbolos de interrogación serán sustituidos por argumentos.

Line 1 Column 0

☒ Ejecutar para cada fila? ☐

☐ Execute as a single statement

☐ Variable substitution

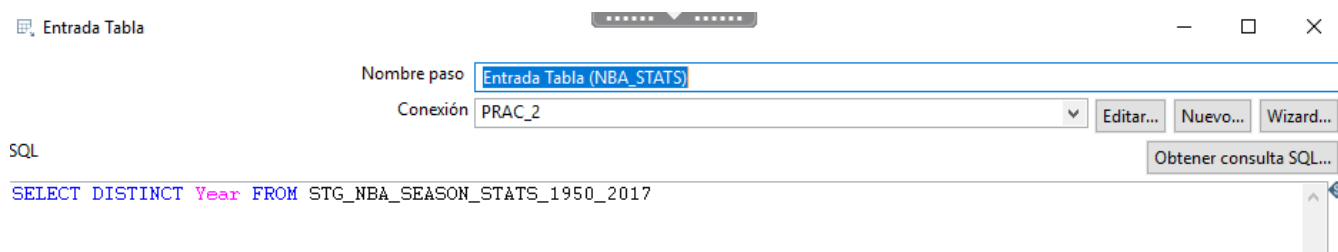
☐ Bind parameters?

☐ Quote Strings?

Parámetros:

#	Nombre de campo a utilizar como argumento
1	

A continuación del paso de borrado es necesario leer las diferentes tablas intermedias para obtener los valores de los diferentes años de los que se tienen estadísticas. Como las fuentes son dos tablas intermedias se colocan dos pasos Entrada Tabla, uno para cada una de las tablas, para el caso de la tabla STG\_NBA\_SEASON\_STATS\_1950\_2017 la sentencia SQL que devuelve los valores únicos de los años almacenados en el campo Year, es la siguiente.



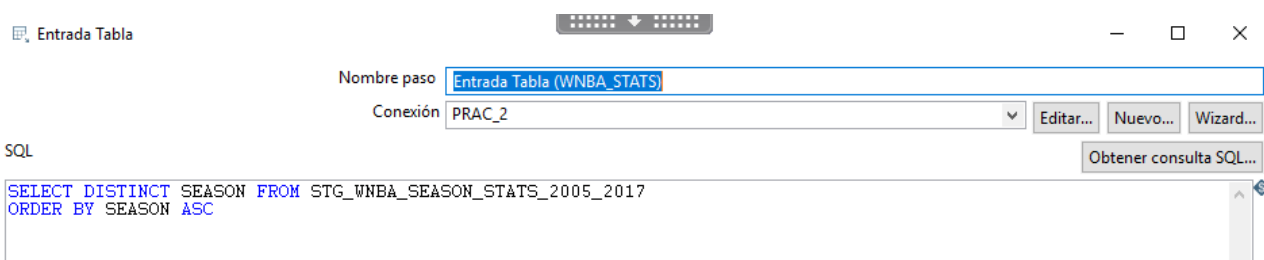
Entrada Tabla

Nombre paso:

Conexión:

SQL

En el caso de la tabla STW\_WNBA\_SEASON\_STATS2005\_2017 la sentencia se modifica parcialmente, para devolver los valores almacenados en el campo SEASON ordenados ascendentemente.



Entrada Tabla

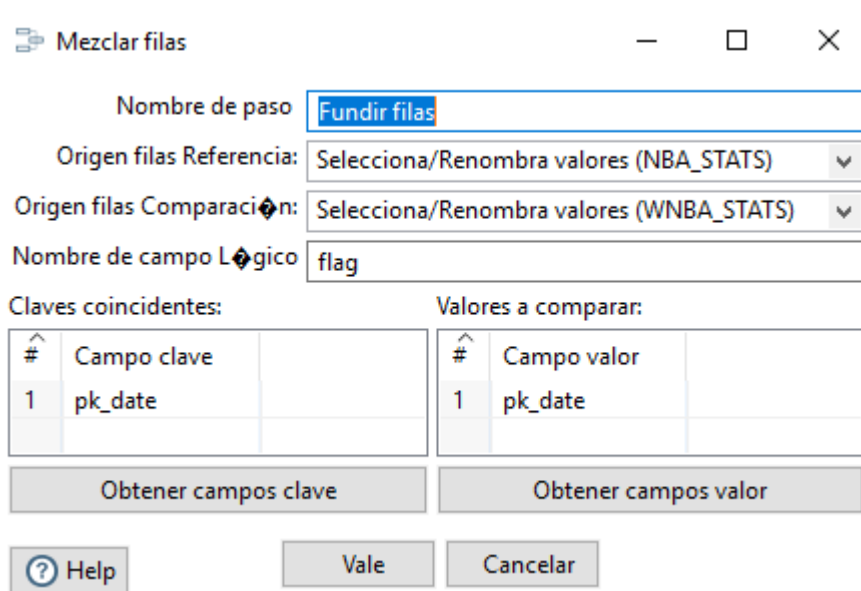
Nombre paso:

Conexión:

SQL

Luego se añaden dos pasos para renombrar los campos *SEASON* Y *Year* a **pk\_date**, tarea necesaria para unificarlos en pasos posteriores, para ello se utilizan los pasos Renombrar/Seleccionar.

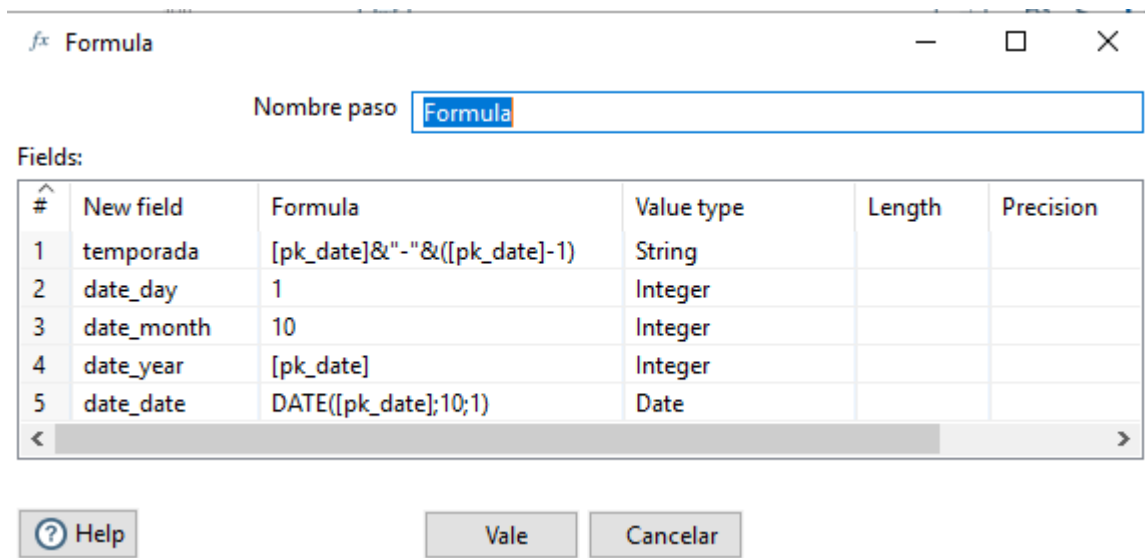
Una vez estos campos tienen el mismo nombre, se utiliza un campo fundir filas para buscar las apariciones de los años en ambos campos, este paso añade una columna llamada *flag* que indica si el valor estaba en ambos *streams* o solo en uno. Como resultado de este paso se obtiene una columna con los valores de todos los años de los que se tiene estadísticas, tanto de la NBA como de la WNBA almacenado en el campo **pk\_date** que será la clave primaria que se cargará en la tabla de dimensiones.



Según el análisis de esta dimensión, se conoce que cuenta con varios campos más (*date\_year*, *date\_month*, *date\_day*, *date\_date* y *temporada*). Algunos de estos se añaden para futuras implementaciones, por lo que serán creados como campos constantes, a continuación se describen sus valores:

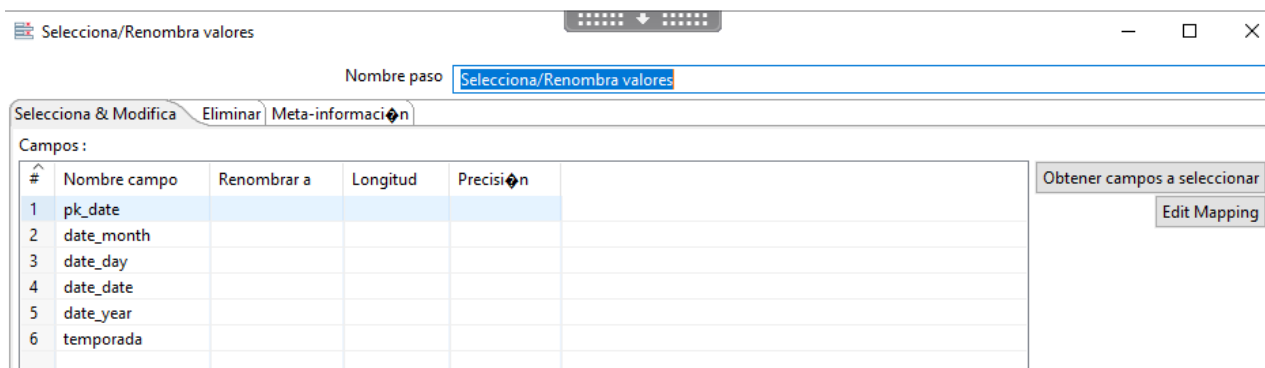
- *date\_year* : El mismo valor que el campo *pk\_date*
- *date\_month*: Por defecto octubre (10)
- *date\_day*: Por defecto el día 1.
- *date\_date*: Tendrá el formato (*date\_year-date\_month-date\_day*)
- *temporada*: Tendrá el formato “Año en curso – Año anterior”.

Para crear todos estos campos se utiliza el step Fórmula, en el que se establecen los valores de los nuevos campos y su tipo. Este paso es de vital importancia ya que facilita enormemente la transformación de campos y su creación a partir de otros, su configuración se muestra en la siguiente figura.



#	New field	Formula	Value type	Length	Precision
1	temporada	[pk_date]&"-"&([pk_date]-1)	String		
2	date_day	1	Integer		
3	date_month	10	Integer		
4	date_year	[pk_date]	Integer		
5	date_date	DATE([pk_date];10;1)	Date		

Como último paso antes de cargar los datos a la tabla se limpia el stream de datos seleccionando solamente los campos que componen la dimensión, descartando así los que fueron creados de manera auxiliar para construir los definitivos.



#	Nombre campo	Renombrar a	Longitud	Precisión
1	pk_date			
2	date_month			
3	date_day			
4	date_date			
5	date_year			
6	temporada			

Finalmente se agrega un paso Salida de Tabla para cargar los datos a la tabla DIM\_Tiempo, el mapeado de los campos es muy sencillo ya que a lo largo de la transformación se fueron adecuando los campos con el objetivo que hacer coincidir los nombres de ambas entidades.

Salida de Tabla

Nombre paso: Salida Tabla

Conexión: PRAC\_2

Esquema destino: dbo

Tabla destino: DIM\_Tiempo

Tamaño de transacción (commit): 1000

Vaciar tabla: ☐

Ignorar errores de inserción: ☐

Specify database fields: ☒

Main options Database fields

Fields to insert:

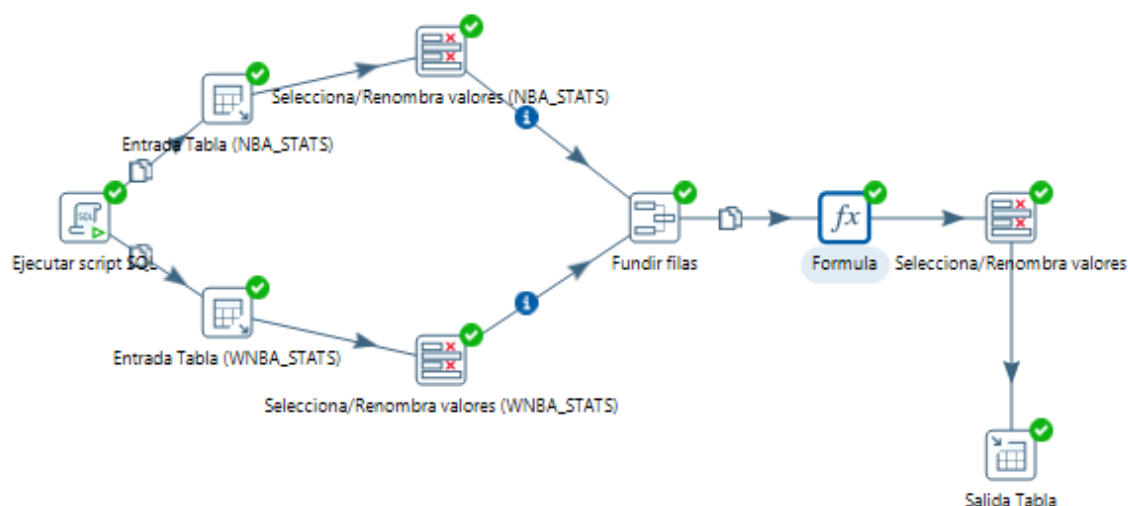
#	Table field	Stream field
1	pk_date	pk_date
2	date_month	date_month
3	date_day	date_day
4	date_date	date_date
5	date_year	date_year
6	temporada	temporada

Get fields

Enter field mapping

A modo de resumen, en esta transformación se realizaron los siguientes pasos.

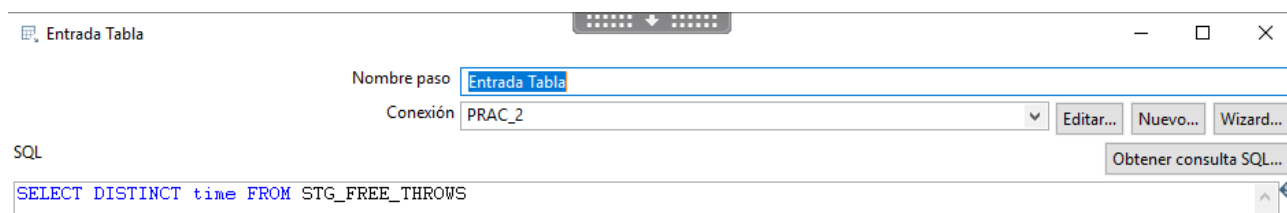
- Eliminación del contenido de la tabla dimensión.
- Lectura de las tablas STG\_ para obtener los datos.
- Renombramiento de los campos.
- Integrar los datos de ambos flujos en uno.
- Añadir campos mediante fórmulas.
- Seleccionar los campos definitivos y desechar los auxiliares.
- Carga de los datos al modelo multidimensional.



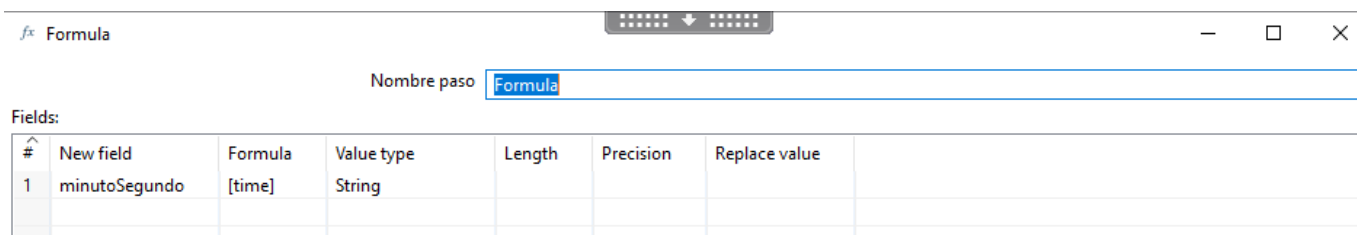
## Transformaciones bloque TR\_DIM\_Minutos

Este bloque contiene las transformaciones necesarias para realizar la carga inicial de la dimensión Minutos en el almacén, esta carga se realiza desde la tabla intermedia “STG\_FREE\_THROWS” ya que es la única en la que se definen estas métricas temporales.

Inicialmente se realizan los pasos de eliminación del contenido existente y lectura de la tabla STG\_FREE\_THROWS de la cual se seleccionan solamente las entradas únicas el campo time, que será la base para la creación de los campos Minuto, Segundo, minutoSegundo que se almacenarán en la DIM\_Minuto.



Una vez cargados estos datos se realiza una copia del campo time mediante un step Fórmula, esta copia será el campo minutoSegundo.



#	New field	Formula	Value type	Length	Precision	Replace value
1	minutoSegundo	[time]	String			

Posteriormente se divide el campo time en dos utilizando como separador el “:”, para separar el valor de los minutos y los segundos, el resultado de esta partición son los campos necesarios para cargar en la base de datos.

Partir campo

Nombre paso:

Campo a partir:

Separador:

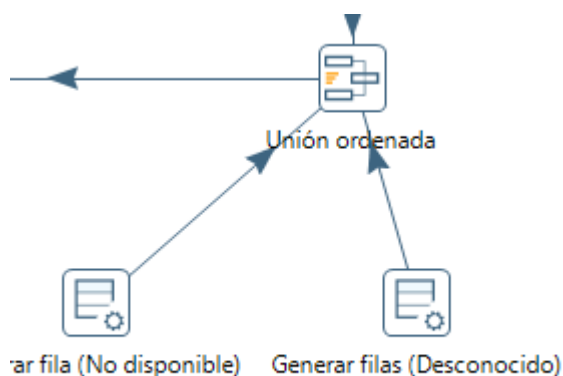
Enclosure:

Campos

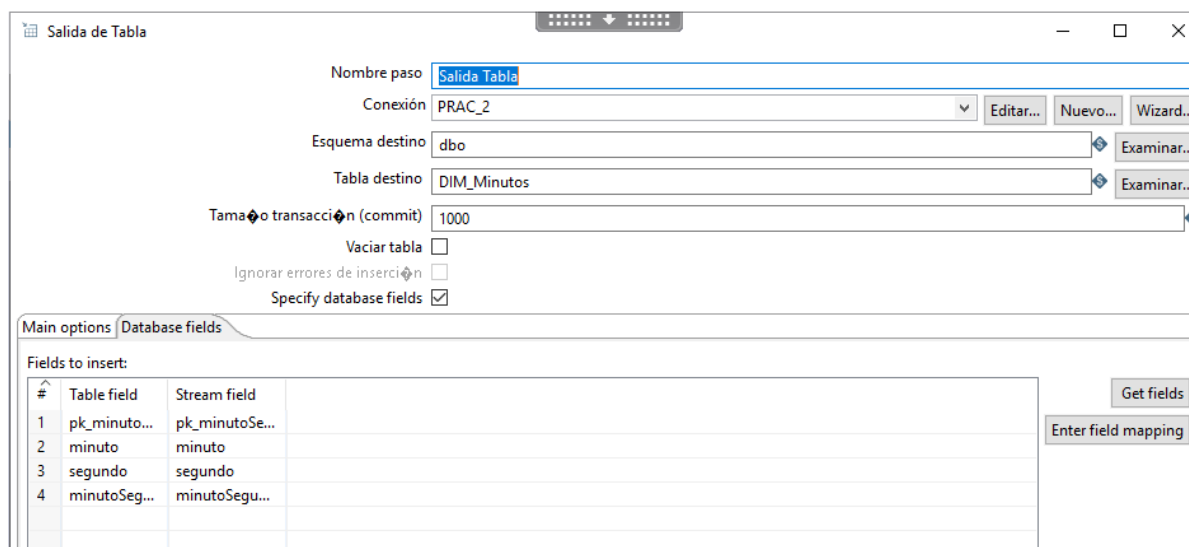
#	Nuevo campo	ID	Eliminar ID?	Tipo	Longitud	Precisión	Formato	Grupo	Decimal	Moneda	Nulo Si	Defecto
1	minuto		N	Integer								
2	segundo		N	Integer								

Posteriormente se añade la secuencia que hace de pk\_minuto y se seleccionan los campos para almacenar en la tabla.

Luego añaden las columnas especiales “desconocido” y “no disponible” y se unen los flujos de datos de manera ordenada.



Para finalizar se cargan los datos a la tabla DIM\_Minutos.



Salida de Tabla

Nombre paso:

Conexión:

Esquema destino:

Tabla destino:

Tamaño de transacción (commit):

Vaciado de tabla: ☐

Ignorar errores de inserción: ☐

Specify database fields: ☒

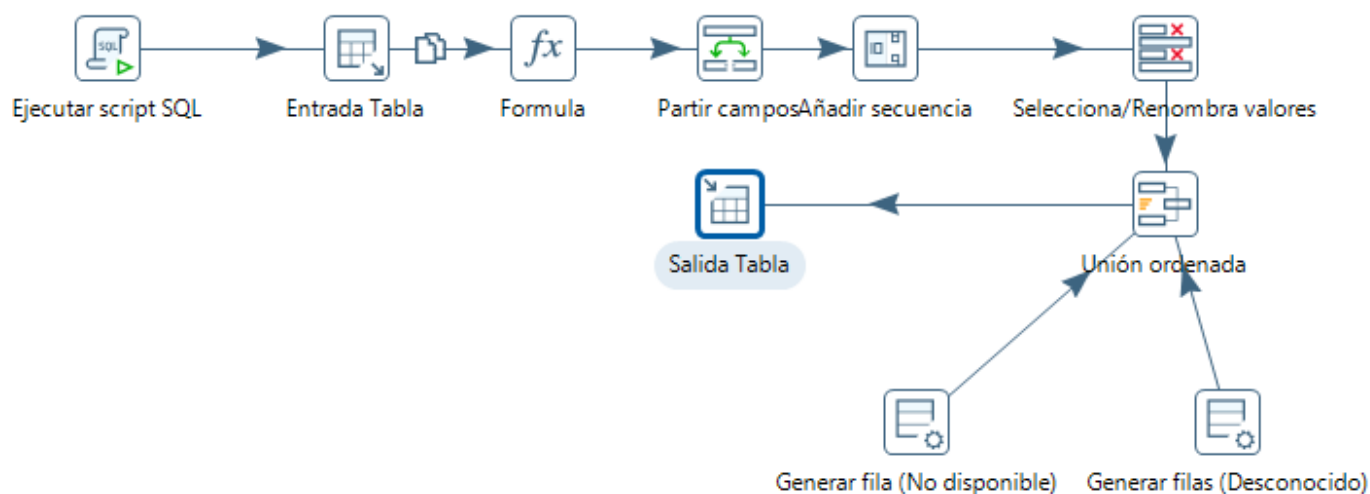
Main options Database fields

Fields to insert:

#	Table field	Stream field
1	pk_minuto...	pk_minutoSe...
2	minuto	minuto
3	segundo	segundo
4	minutoSeg...	minutoSegu...

A modo de resumen, en esta transformación se realizaron los siguientes pasos.

- Eliminación del contenido de la tabla dimensión.
- Lectura de las tablas STG\_FREE\_THOWS para obtener los datos del campo time.
- Creación del campo minutoSegundo a partir de una copia del campo time.
- Partición del campo time en los campos minuto y segundo.
- Agregación de la secuencia pk\_minutos.
- Selección y renombramiento de campo.
- Creación de los registros especiales.
- Unión de los flujos de datos.
- Carga de los datos al modelo multidimensional.

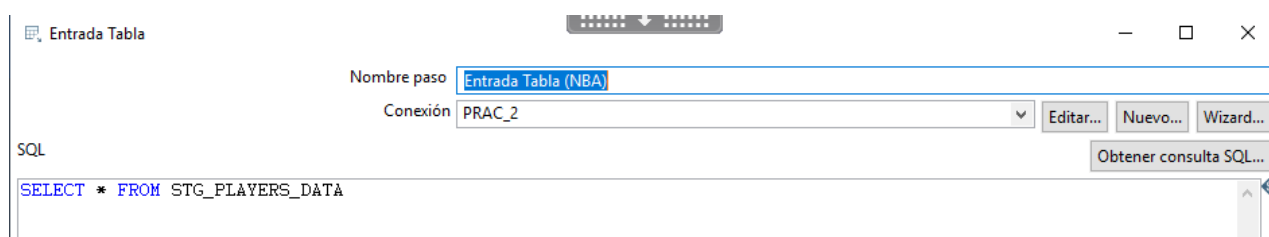


## Transformaciones bloque TR\_DIM\_Jugadores

Este bloque contiene las transformaciones necesarias para realizar la carga inicial de la dimensión Jugadores en el almacén, esta carga se realiza desde las tablas intermedias “STG\_PLAYERS\_DATA” “STG\_WNBA\_PLAYERS” de las que se tomarán los datos relacionados con las estadísticas de los jugadores de ambas ligas.

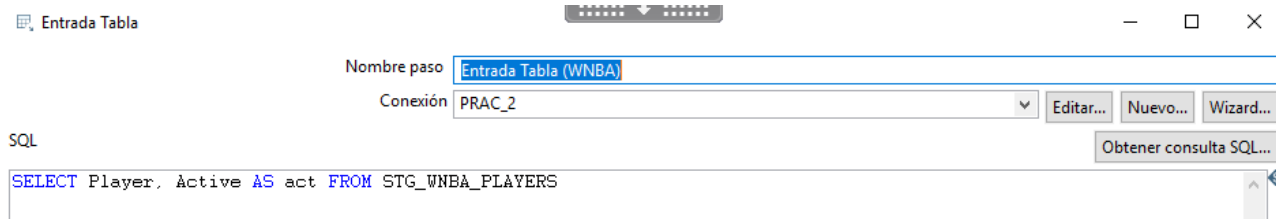
Inicialmente se ejecuta el *script* que elimina el contenido de la tabla, y posteriormente se cargan en dos flujos diferentes los datos de las tablas del *Stagin Area*.

Uno de los flujos para los datos de los jugadores de la NBA.



Y el otro para las jugadoras de la WNBA.

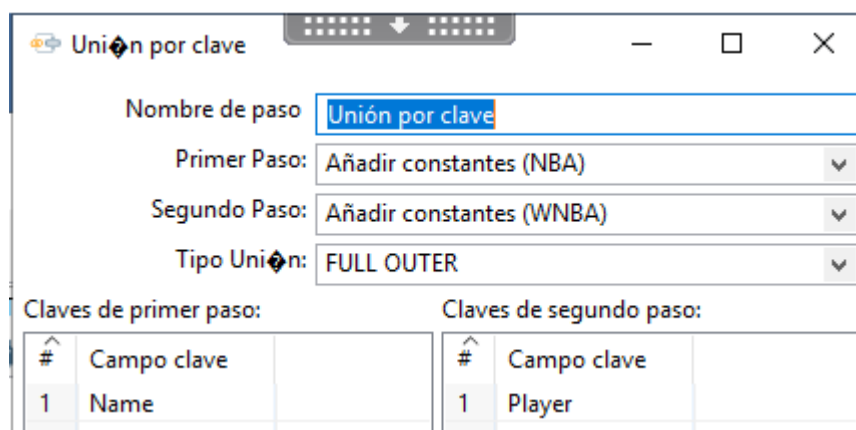




Luego se añaden campos constantes en cada uno de los flujos, correspondientes a los campos liga y sexo, teniendo los siguientes valores:

- Flujo superior NBA: liga = NBA, sexo = 1
- Flujo inferior WNBA: liga = WNBA, sexo = 0

Posteriormente se unen ambos flujos de manera ordenada, como resultado de este paso se obtiene una tabla en la que las estadísticas de las mujeres están en su mayoría ausentes. Se conoce que los nombres en ambos flujos son diferentes, y lo que se desea es unirlos en un mismo flujo, por eso se utiliza la opción FULL OUTER para la unión, y los campos Name y Player para hacer las veces de claves.



Luego se realiza un mapeo de valores para convertir los valores de “YES” y “No” del campo activo a la codificación 1 -> “YES” 0->”NO”.

Para rellenar el campo active correspondiente a los datos de la NBA, se incluye un campo Búsqueda en la Base de Datos que busca en la tabla STG\_NBA\_PLAYERS los datos correspondientes a su estado, en este paso comparando los nombres se obtiene su estado. A continuación se muestra esta configuración.

Búsqueda de valor en base de datos

Nombre paso: Búsqueda en base de datos

Conexión: PRAC\_2 Editar... Nuevo... Wizard...

Esquema de búsqueda: dbo Examinar...

Tabla de búsqueda: STG\_NBA\_PLAYERS Examinar...

☐ Habilitar cache?

Tamaño de cache en filas (0=todas): 0

☐ Load all data from table

La clave(s) para realizar búsqueda de valor(es):

#	Campo de tabla	Comparador	Campo1	Campo2
1	Player	=	Name	

Valores a devolver de la tabla de búsqueda:

#	Campo	Nuevo nombre	Defecto	Tipo
1	Active	NBA_active		Integer

☐ No procesar la fila si la búsqueda falla

☐ Producir error si se obtienen múltiples

Ordenar por:

Help Vale Cancelar Obtener Campos Obtener Campos Búsqueda

Como se había comentado antes, los nombres de los jugadores y jugadoras estaban en campos secundarios separados como resultado de la unión excluyente, para unificarlos es necesario incluir un paso Concatenar campos, en el que utilizando los campos Name y Player se conforma el campo nombre.

Concat fields

Nombre paso: Concat fields (Nombre)

Target Field Name: nombre

Length of Target Field: 0

Separator:  Insert TAB

Enclosure:

Fields Advanced

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim Type	Null
1	Name	String							ninguno	
2	Player	String							ninguno	

De manera similar sucede con los campos Active, Sexo y liga por lo que será necesario unificarlos con sus respectivos equivalentes con steps Concatenar campos. A continuación se propone la configuración de estos pasos.

## Active

Concat fields

Nombre paso: Concat fields (Active)

Target Field Name: activo

Length of Target Field: 0

Separator: Insert TAB

Enclosure:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim Type	Null
1	WNBA_Active	Integer							ninguno	
2	NBA_active	Integer							ninguno	

## Sexo

Concat fields

Nombre paso: Concat fields (Sexo)

Target Field Name: sexo

Length of Target Field: 0

Separator: Insert TAB

Enclosure: "

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim Type	Null
1	sexo_NBA	Integer							ninguno	
2	sexo_WNBA	Integer							ninguno	

## Liga

Concat fields

Nombre paso: Concat fields (Liga)

Target Field Name: liga

Length of Target Field: 0

Separator: Insert TAB

Enclosure:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim Type	Null
1	liga_NBA	String							ninguno	
2	liga_WNBA	String							ninguno	

Posteriormente se añade la secuencia que compone el campo pk\_jugador y se seleccionan y renombran los campos del flujo de datos.

Selecciona/Renombra valores

Nombre paso: Selecciona/Renombra valores

Selecciona & Modifica | Eliminar | Meta-información

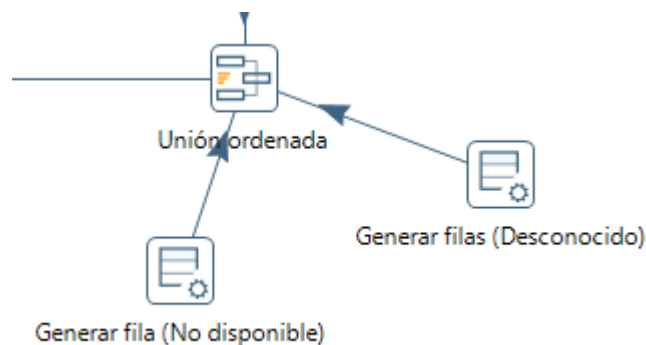
Campos:

#	Nombre campo	Renombrar a	Longitud	Precisión
1	pk_jugador			
2	liga			
3	id	cod_jugador		
4	nombre			
5	position	posicion_juego		
6	sexo			
7	activo			
8	Height	altura		
9	weight	peso		
10	Shoots	shoots		
11	college	Universidad		
12	birthDate	fecha_nacimiento		
13	birthPlace	ciudad_nacimiento		
14	career_AST			
15	career_FG%			
16	career_FG3%			
17	career_FT%			
18	career_G			
19	career_PER			
20	career_PTS			
21	career_TRB			
22	career_WS			
23	career_eFG%			

Obtener campos a seleccionar

Edit Mapping

Como pasos previos a la carga se añaden los registros especiales para gestionar los casos “desconocido” y “no disponible” y se ordenan con el resto del flujo.



Para finalizar se cargan los datos al modelo, el mapeo de los valores ha sido simplificado gracias al paso de renombrar, a continuación se muestra esta configuración.

Salida de Tabla

Nombre paso: Salida Tabla

Conexión: PRAC\_2 [Editar... Nuevo... Wizard...]

Esquema destino: dbo [Examinar...]

Tabla destino: DIM\_Jugadores [Examinar...]

Tamaño de transacción (commit): 1000

Vaciar tabla: ☐

Ignorar errores de inserción: ☐

Specify database fields: ☒

Main options Database fields

Fields to insert:

#	Table field	Stream field
1	pk_jugador	pk_jugador
2	liga	liga
3	cod_jugador	cod_jugador
4	nombre	nombre
5	posicion_ju...	posicion_jue...
6	sexo	sexo
7	activo	activo
8	altura	altura
9	peso	peso
10	shoots	shoots
11	Universidad	Universidad
12	fecha_naci...	fecha_nacimi...
13	ciudad_nac...	ciudad_naci...
14	career_AST	career_AST
15	career_FG%	career_FG%
16	career_FG3%	career_FG3%
17	career_FT%	career_FT%
18	career_G	career_G
19	career_PER	career_PER
20	career_PTS	career_PTS
21	career_TRB	career_TRB
22	career_WS	career_WS
23	career_eFG%	career_eFG%

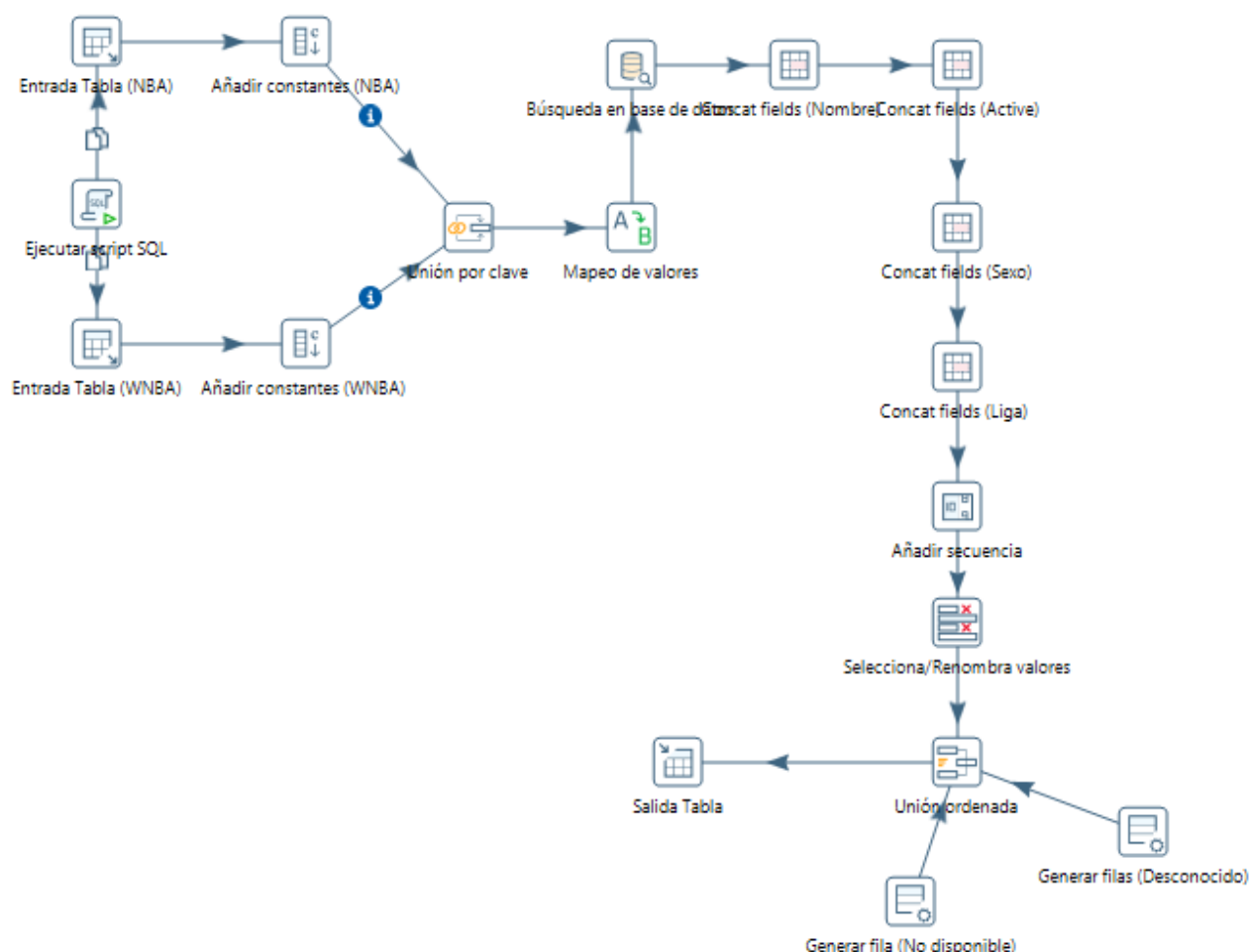
[Get fields]

[Enter field mapping]

A modo de resumen, en esta transformación se realizaron los siguientes pasos.

- Eliminación del contenido de la tabla dimensión.
- Lectura de las tablas STG\_PLAYERS\_DATA y STG\_WNBA\_PLAYERS para obtener los datos.
- Creación de campos sexo y liga con constantes.
- Ordenación por clave de ambos flujos.
- Mapeo de valores para codificar el campo active de las jugadoras.
- Búsqueda en la Base de datos para obtener el valor de active para los jugadores.
- Concatenar los campos secundarios de los flujos y unificarlos en el campo definitivo.
- Añadir secuencia de clave primaria.
- Seleccionar y renombrar valores.
- Creación de registros especiales.

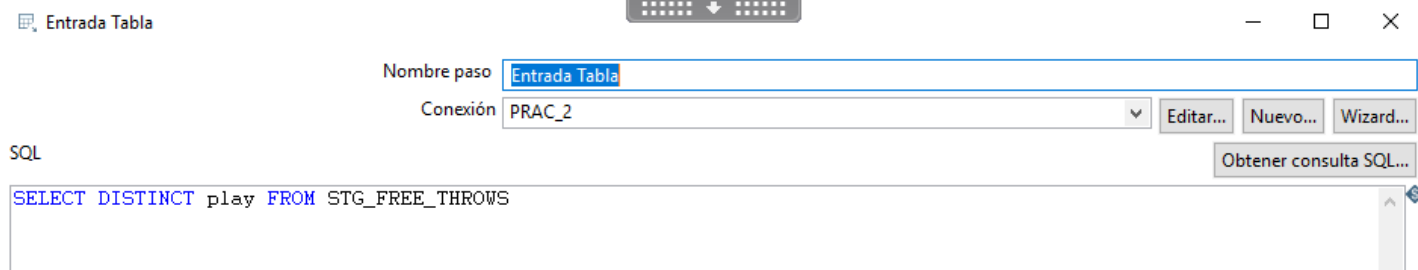
- Unión de todos los flujos de datos.
- Carga de los datos al modelo multidimensional.



## Transformaciones bloque TR\_DIM\_Jugadas

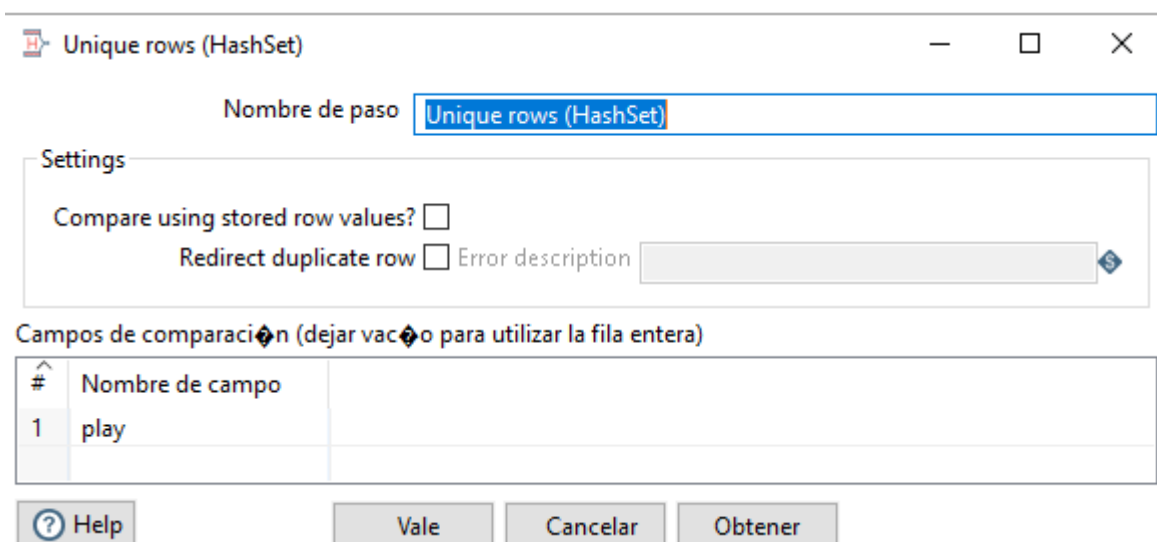
Este bloque contiene las transformaciones necesarias para realizar la carga inicial de la dimensión Jugadas en el almacén, esta carga se realiza desde la tabla intermedia “STG\_FREE\_THROWS” ya que solo se contemplarán este tipo de jugadas.

Inicialmente se eliminarán los anteriores registros de la tabla DIM\_Jugadas, ejecutando el *script*, y se leen los datos de la tabla del área intermedia, solamente se tomarán los datos referentes al campo play, que describe el contenido de la jugada en sí.

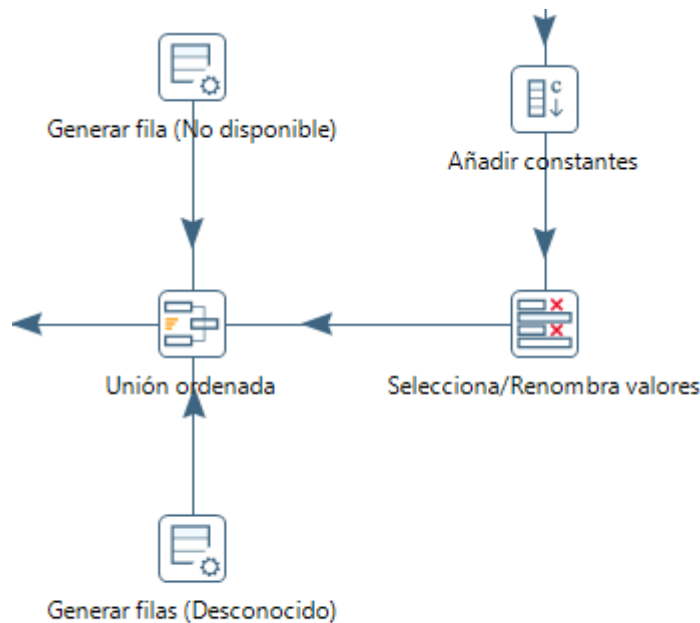


Luego con un campo *replace string* y utilizando expresiones regulares se elimina una parte del contenido de cada registro, quedando prácticamente el final de cada entrada. Este final es el que contiene la parte importante de la descripción del tiro libre, y es lo que servirá de campo desc\_jugada.

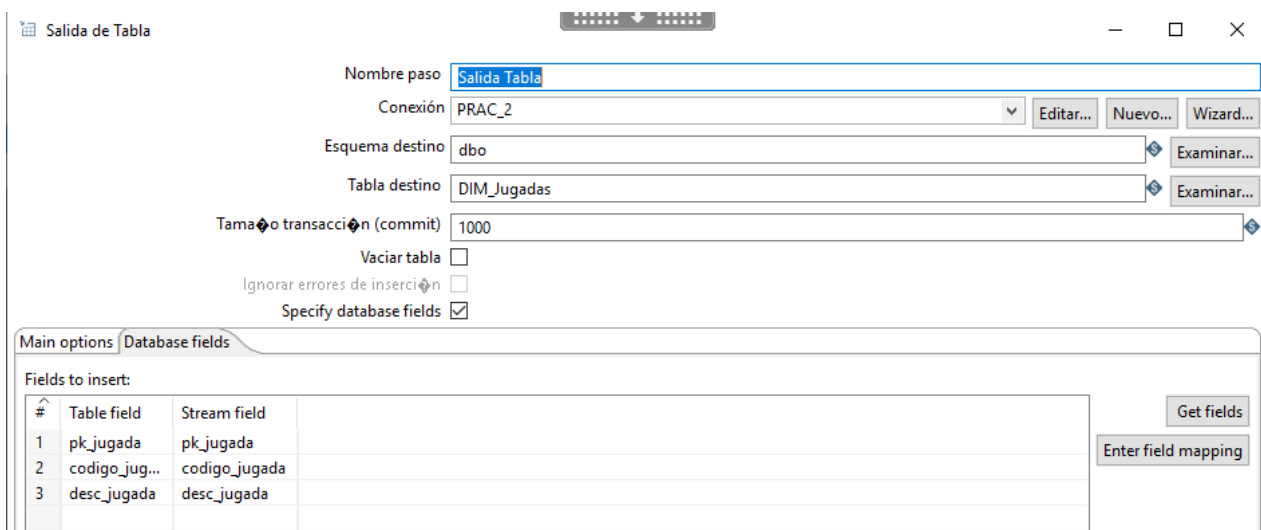
Posteriormente se utiliza un paso que selecciona solamente las filas únicas del campo play ya filtrado y limpio.



Posteriormente se repite el proceso de creación de la secuencia de la clave primaria, la selección de los campos necesarios para cargar a la tabla, la creación de los registros especiales y la unión de todos los flujos. También se añade un campo con el tipo de jugada, que en este caso solamente se contemplan los tiros libres.



Finalmente se cargan los datos a la tabla.



Salida de Tabla

Nombre paso: Salida Tabla

Conexión: PRAC\_2 [Editar...] [Nuevo...] [Wizard...]

Esquema destino: dbo [Examinar...]

Tabla destino: DIM\_Jugadas [Examinar...]

Tamaño de transacción (commit): 1000

Vaciar tabla: ☐

Ignorar errores de inserción: ☐

Specify database fields: ☒

Main options | Database fields

Fields to insert:

#	Table field	Stream field
1	pk_jugada	pk_jugada
2	codigo_jug...	codigo_jugada
3	desc_jugada	desc_jugada

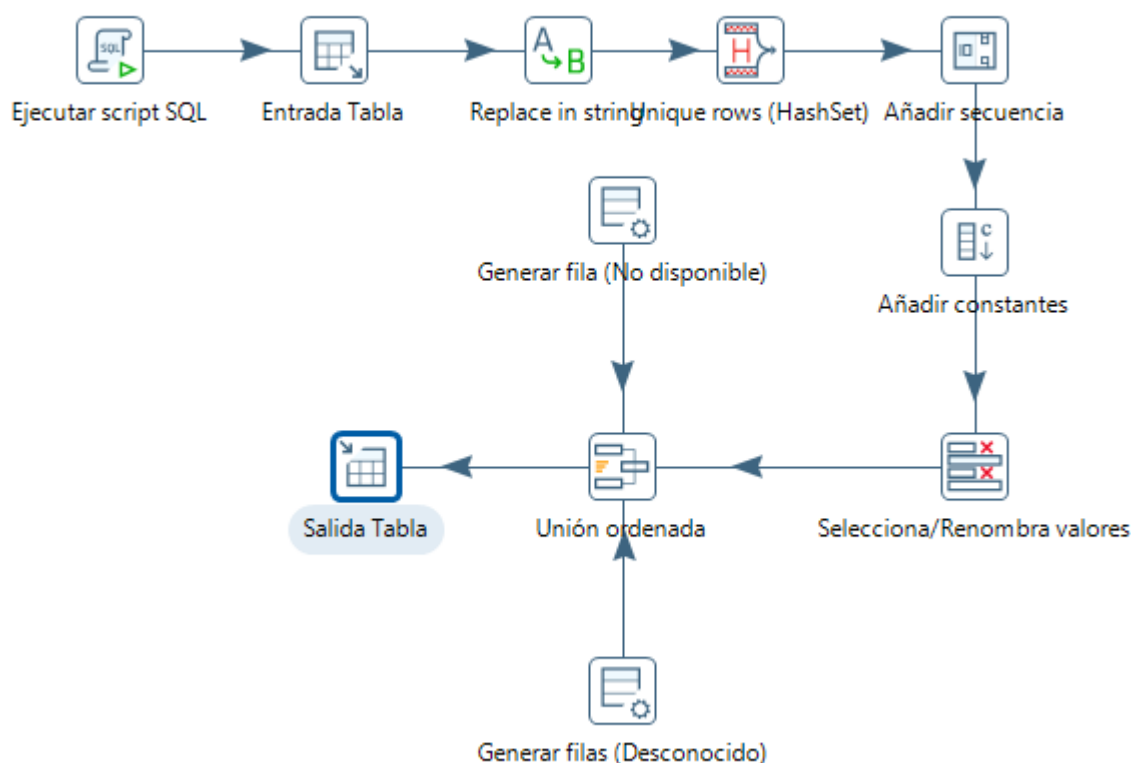
[Get fields] [Enter field mapping]

A modo de resumen, en esta transformación se realizaron los siguientes pasos.

- Eliminación del contenido de la tabla dimensión.
- Lectura de las tablas STG\_FREE\_THROWS para obtener el campo play.
- Sustitución y filtrado de las descripciones.
- Selección de filas únicas filtradas por el campo play.
- Creación de la clave primaria.



- Anadir constantes de tipo de jugada.
- Selección y renombramiento de campos.
- Creación de los registros especiales.
- Unión de los flujos de datos.
- Carga de los datos al modelo multidimensional.



## Transformaciones bloque TR\_DIM\_PosicionesJuego

Este bloque contiene las transformaciones necesarias para realizar la carga inicial de la dimensión Posiciones de Juego en el almacén, esta carga se realiza desde la tabla intermedia “STG\_NBA\_SEASON\_STATS\_1950\_2017” de las que se tomarán solamente el campo pos.

Tal y como se viene realizando hasta el momento, el primer paso es eliminar el contenido de la tabla DIM\_PosicionesJuego, para proceder entonces a cargar los registros únicos del campo posición de la tabla del área intermedia.

Entrada Tabla

Nombre paso: Entrada Tabla

Conexión: PRAC\_2

SQL: `SELECT DISTINCT Pos FROM STG_NBA_SEASON_STATS_1950_2017`

Esta transformación no es muy compleja en el sentido de tener muchas fuentes de datos o realizar complicadas fórmulas pero si es necesario aclarar que se separan los datos en dos flujos distintos, uno para crear las descripciones en español y otro para las descripciones en ingles.

Antes de separar los flujos se parte el campo pos tomando como delimitador el carácter "-", ya que hay casos en los que un jugador puede desempeñar más de una posición. Como resultado de esta partición se obtienen campos con los valores de la posición 1 y 2, nótese que algunas posiciones no cuentan con posición 2 (las posiciones básicas) por lo que su valor quedará nulo.

Partir campo

Nombre paso

Partir campos

Campo a partir

Pos

Separador

-

Enclosure

Campos

#	Nuevo campo	ID	Eliminar ID?	Tipo	Longitud	Precisión	Formato	Grupo	Decimal	Moneda	NuloSi	Defecto
1	Pos_1		N	String								
2	Pos_2		N	String								

<

>

Help

Vale

Cancelar

Utilizando una descripción proporcionada en el enunciado del caso de estudio se utiliza un mapeo de valores para establecer una relación entre las posiciones básicas y su número en baloncesto.

**Mapeo de valores**

Nombre de paso : Mapeo de valores (Pos)

Nombre de campo origen : Pos\_1

Nombre de campo destino (vacío=sobreescribir) num\_pos

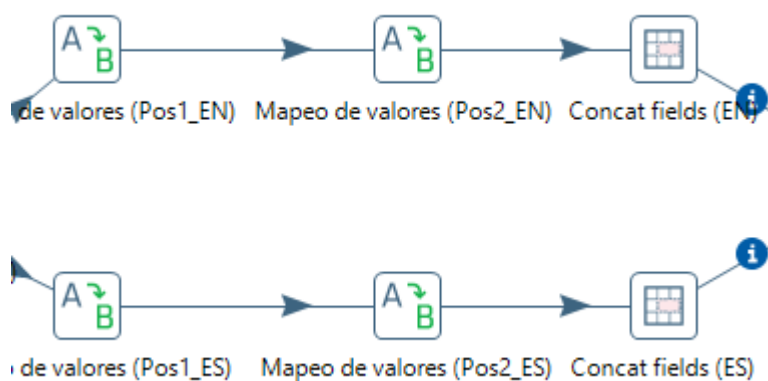
Default upon non-matching :

Valores de campo:

#	Valor origen	Valor destino
1	PG	1
2	SG	2
3	SF	3
4	PF	4
5	C	5
6	F	4
7	G	2

? Help Vale Cancelar

Es entonces cuando se separan ambos flujos por la parte de arriba se crearán las descripciones en inglés y por abajo en español.



En ambos flujos ocurre exactamente lo mismo, se realiza un mapeo de las descripciones proporcionadas en el enunciado en ambos idiomas y se aplica este mapeo a cada uno de los campos de posición. Como resultado de este segmento se obtiene 4 campos con descripciones de las posiciones. Para ejemplificar se muestra a continuación el mapeo de

valores para la posición 1 en inglés y para la posición 2 en español, ya que los restantes pasos son iguales solo que cambiando el campo sobre el que se aplican.

### Posición 1 en Ingles

**Mapeo de valores**

Nombre de paso : Mapeo de valores (Pos1\_EN)

Nombre de campo origen : Pos\_1

Nombre de campo destino (vacío=sobreescribir) : desc\_pos\_1\_EN

Default upon non-matching :

Valores de campo:

#	Valor origen	Valor destino
1	PG	Point Guard
2	SG	Shooting Guard
3	SF	Small Forward
4	PF	Power Fordward
5	C	Center
6	F	Power Fordward
7	G	Shooting Guard

### Posición 2 en español

**Mapeo de valores**

Nombre de paso : Mapeo de valores (Pos2\_ES)

Nombre de campo origen : Pos\_2

Nombre de campo destino (vacío=sobreescribir) : desc\_pos\_2\_ES

Default upon non-matching :

Valores de campo:

#	Valor origen	Valor destino
1	PG	Base
2	SG	Escolta
3	SF	Alero
4	PF	Ala-pívot
5	C	Pívot
6	F	Ala-pívot
7	G	Escolta

Para obtener la descripción definitiva se concatenan estos 2 campos por idioma.

Concat fields

Nombre paso: Concat fields (EN)

Target Field Name: desc\_posicion\_EN

Length of Target Field: 0

Separator: /

Enclosure: "

Insert TAB

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim Type	Null
1	desc_pos_1_EN	String							ninguno	
2	desc_pos_2_EN	String							ninguno	

Para unir los flujos se utiliza una unión por clave seleccionando los campos originales POS\_1 y POS\_2.

Unión por clave

Nombre de paso: Unión por clave

Primer Paso: Concat fields (EN)

Segundo Paso: Concat fields (ES)

Tipo Unión: INNER

Claves de primer paso:

#	Campo clave
1	Pos_1
2	Pos_2

Claves de segundo paso:

#	Campo clave
1	Pos_1
2	Pos_2

Estos mismos campos se concatenan para formar el campo Pos final, al concatenarlos, los valores de posición 2 que eran nulos se sustituirán por valores vacíos.

Concat fields

Nombre paso: Concat fields

Target Field Name: Pos

Length of Target Field: 0

Separator: Insert TAB

Enclosure:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim Type	Nu
1	Pos_1	None							ninguno	
2	Pos_2	None							ninguno	

Posteriormente se cambia el tipo al campo num\_pos con un paso Renombrar.

Selecciona/Renombrar valores

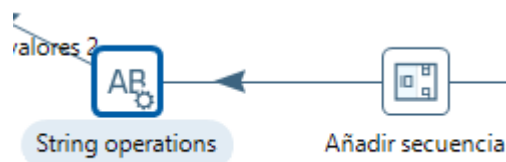
Nombre paso: Selecciona/Renombrar valores

Selecciona & Modifica | Eliminar | Meta-información

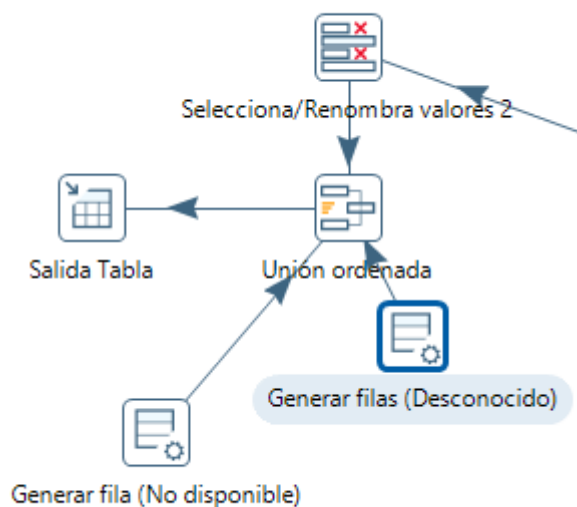
Campos a modificar meta información:

#	Nombre campo	Renombrar a	Tipo	Longitud	Precisión	Binary to Normal?	Format	Date F
1	num_pos		Integer			N		N

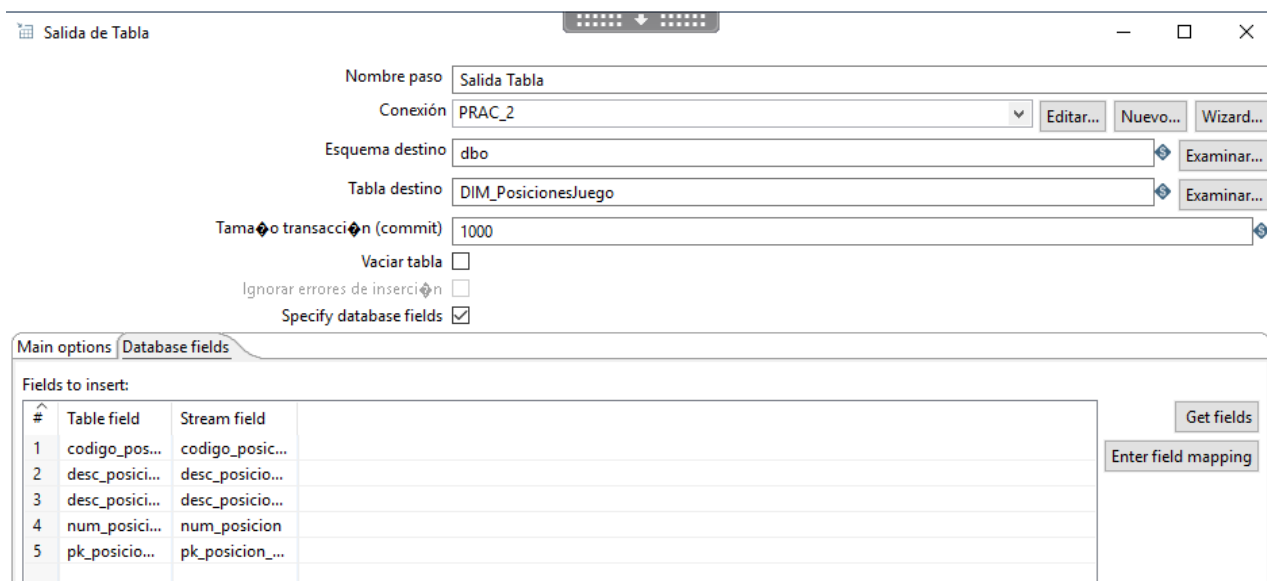
Se añade la secuencia de la clave primaria y se añade un campo de *strings operations* para convertir a mayúsculas los campos de tipo texto y eliminar espacioa en los laterales.



Por último se seleccionan los campos que formarán parte del modelo, se crean los registros especiales para los casos “no disponible” y “desconocido” y se unen los flujos de manera ordenada.



Y para finalizar la carga en la base de datos de este flujo resultante.



Nombre paso: Salida Tabla

Conexión: PRAC\_2 [Editar...] [Nuevo...] [Wizard...]

Esquema destino: dbo [Examinar...]

Tabla destino: DIM\_PosicionesJuego [Examinar...]

Tamaño transacción (commit): 1000

Vaciar tabla: ☐

Ignorar errores de inserción: ☐

Specify database fields: ☒

Main options | Database fields

Fields to insert:

#	Table field	Stream field
1	codigo_pos...	codigo_posic...
2	desc_posici...	desc_posicio...
3	desc_posici...	desc_posicio...
4	num_posici...	num_posicion
5	pk_posicio...	pk_posicion_...

[Get fields] [Enter field mapping]

A modo de resumen, en esta transformación se realizaron los siguientes pasos.

- Eliminación del contenido de la tabla dimensión.
- Lectura de las tablas STG\_NBA\_SEASON\_STATS\_1950\_2017 para obtener los datos del campo pos.
- Se parte el campo pos en dos.

- Se mapean los números de posición.
- Se separa el flujo en dos.
- En cada flujo se mapean las descripciones.
- Se concatenan las descripciones.
- Se unen los flujos con las descripciones definitivas creadas.
- Se concatenan las posiciones a su estado original.
- Se cambia el tipo a la variable pos.
- Se añade la secuencia de la clave primaria.
- Se convierten a mayúsculas y se eliminan los espacios.
- Se crean los registros especiales
- Se unen todos los flujos de datos.
- Carga de los datos al modelo multidimensional.

## Transformaciones bloque TR\_DIM\_ Partidos

Este bloque contiene las transformaciones necesarias para realizar la carga inicial de la dimensión Partidos en el almacén, esta carga se realiza desde la tabla intermedia “STG\_FREE\_TRHOWS”.

El primer paso es eliminar el contenido de la DIM\_Partidos, luego se lee de la tabla del *Stagin Area* los datos de los partidos.



Entrada Tabla

Nombre paso: Entrada Tabla

Conexión: PRAC\_2

Editar... Nuevo... Wizard...

Obtener consulta SQL...

SQL

```
SELECT game_id, playoffs, game, season, score
FROM STG_FREE_THROWS
ORDER BY game_id
```

Luego se filtran el flujo de datos para que solo muestre las columnas únicas y se seleccionan los datos necesarios para el modelo con un paso Seleccionar/Renombrar.



Selecciona/Renombra valores

Nombre paso: Selecciona/Renombra valores

Selecciona & Modifica | Eliminar | Meta-información

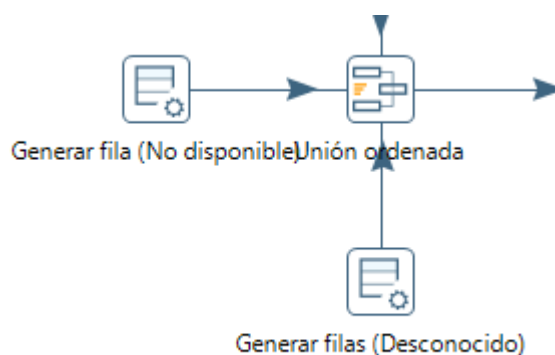
Campos:

#	Nombre campo	Renombrar a	Longitud	Precisión
1	game_id	pk_partido		
2	game	partido		
3	playoffs			
4	season	temporada		
5	score	resultado		

Obtener campos a seleccionar

Edit Mapping

Para finalizar se crean los registros especiales que soportan los casos “desconocido” y “no disponible” y se unen todos los flujos de datos.



Y se cargan los datos a la tabla.

Salida de Tabla

Nombre paso: Salida Tabla

Conexión: PRAC\_2

Esquema destino: dbo

Tabla destino: DIM\_Partidos

Tamaño transacción (commit): 1000

Vaciar tabla: ☐

Ignorar errores de inserción: ☐

Specify database fields: ☒

Main options | Database fields

Fields to insert:

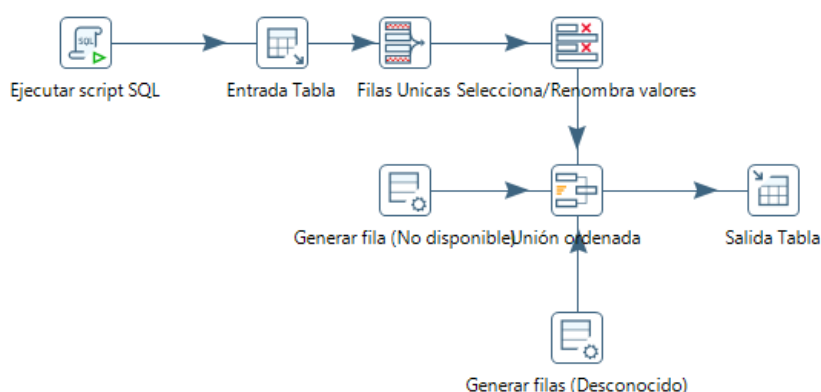
#	Table field	Stream field
1	pk_partido	pk_partido
2	partido	partido
3	playoffs	playoffs
4	temporada	temporada
5	resultado	resultado

Get fields

Enter field mapping

A modo de resumen, en esta transformación se realizaron los siguientes pasos.

- Eliminación del contenido de la tabla dimensión.
- Lectura de las tablas STG\_FREE\_THROWS para obtener los datos.
- Selección de filas únicas.
- Selección y renombramiento de campos.
- Creación de registros especiales.
- Unión de todos los flujos.
- Carga de los datos al modelo multidimensional.



## Transformaciones bloque TR\_DIM\_Estados

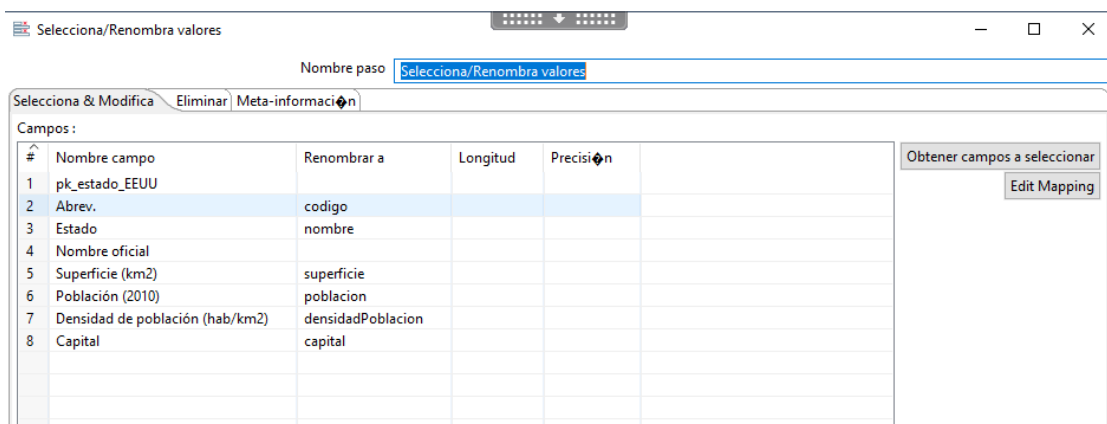
Este bloque contiene las transformaciones necesarias para realizar la carga inicial de la dimensión Estados en el almacén, esta carga se realiza desde la tabla intermedia “STG\_STATE” .

Como siempre en este tipo de transformaciones el primer paso es eliminar el contenido de la tabla y luego se leen los datos desde la tabla del *Stagin Area*.



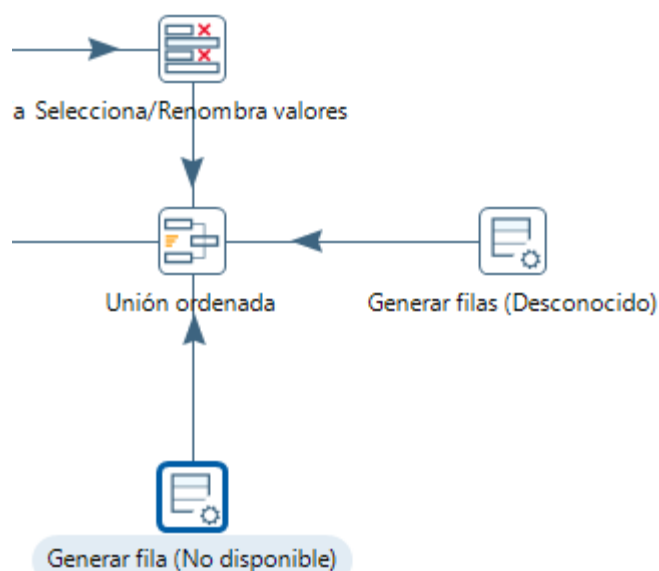
En esta transformación los datos están bastante bien estructurados y prácticamente pueden ser cargados directamente a la tabla de la base de datos, solamente será necesario añadir la secuencia de la clave primaria seleccionar los campos y crear los registros especiales.

## Selección y renombramiento de campos.

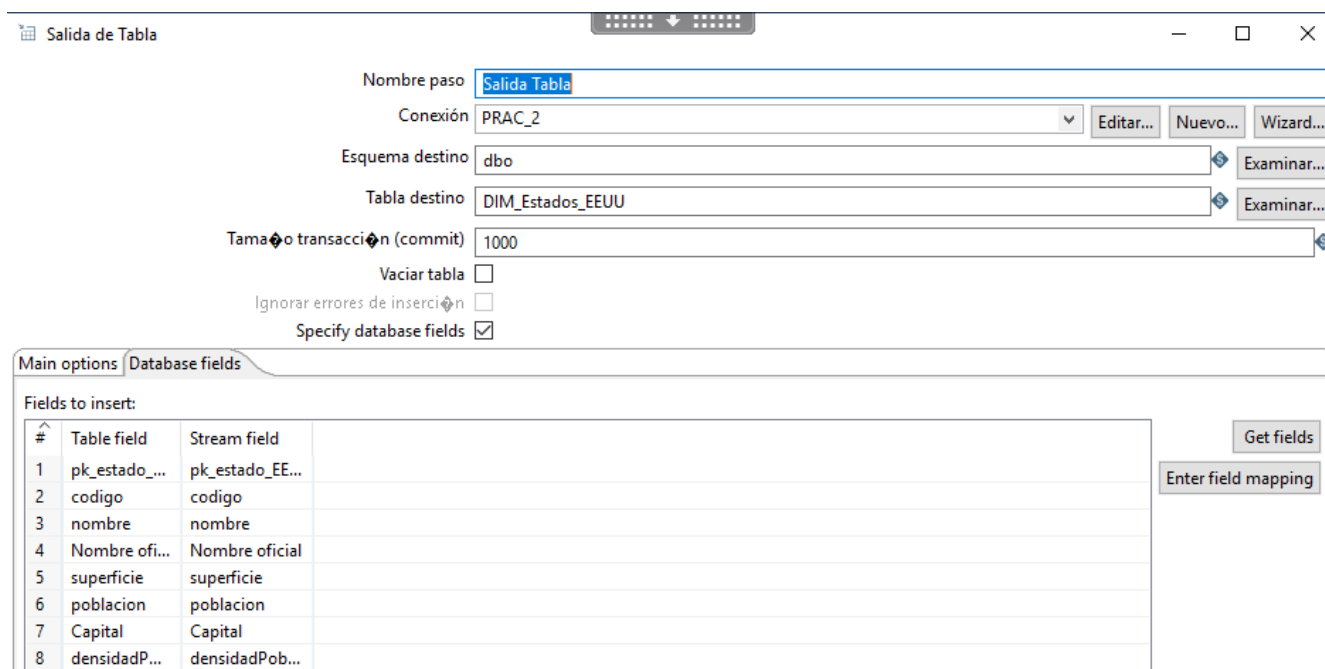


#	Nombre campo	Renombrar a	Longitud	Precisión
1	pk_estado_EEUU			
2	Abrev.	codigo		
3	Estado	nombre		
4	Nombre oficial			
5	Superficie (km2)	superficie		
6	Población (2010)	poblacion		
7	Densidad de población (hab/km2)	densidadPoblacion		
8	Capital	capital		

## Creación de registros especiales y unión ordenada.



Finalmente se cargan los datos a la tabla DIM\_Estados\_EEUU



Salida de Tabla

Nombre paso: Salida Tabla

Conexión: PRAC\_2

Esquema destino: dbo

Tabla destino: DIM\_Estados\_EEUU

Tamaño de transacción (commit): 1000

Vaciar tabla: ☐

Ignorar errores de inserción: ☐

Specify database fields: ☒

Main options | Database fields

Fields to insert:

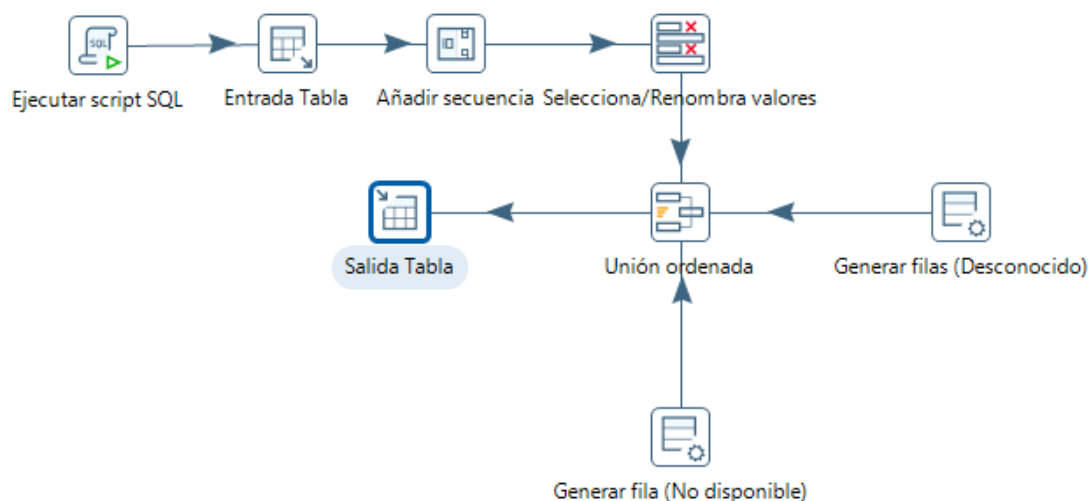
#	Table field	Stream field
1	pk_estado_...	pk_estado_EE...
2	codigo	codigo
3	nombre	nombre
4	Nombre ofi...	Nombre oficial
5	superficie	superficie
6	poblacion	poblacion
7	Capital	Capital
8	densidadP...	densidadPob...

Get fields

Enter field mapping

A modo de resumen, en esta transformación se realizaron los siguientes pasos.

- Eliminación del contenido de la tabla dimensión.
- Lectura de las tablas STG\_STATES para obtener los datos.
- Añadir secuencia de clave primaria.
- Seleccionar y renombrar los campos.
- Creación de registros especiales.
- Unión de todos los flujos.
- Carga de los datos al modelo multidimensional.



## Transformaciones bloque TR\_DIM\_Conferencias

Este bloque contiene las transformaciones necesarias para realizar la carga inicial de la dimensión Conferencias en el almacén, esta carga se realiza desde la tabla intermedia “STG\_NBA\_TEAMS”.

Inicialmente se elimina el contenido de la tabla DIM\_Conferencias, y se leen los datos de la tabla Intermedia, solamente se selecciona el campo conferencia y en la petición SQL se seleccionan los valores únicos.

Entrada Tabla

Nombre paso

Entrada Tabla 2

Conexión

PRAC\_2

Editar...

Nuevo...

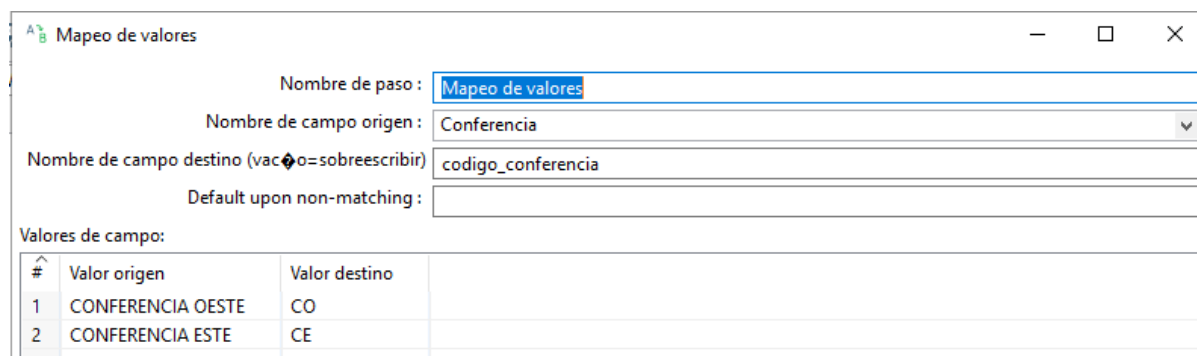
Wizard...

SQL

SELECT DISTINCT Conferencia FROM STG\_NBA\_TEAMS

Obtener consulta SQL...

Luego se mapean los valores de la siguiente manera.



Nombre de paso: Mapeo de valores

Nombre de campo origen: Conferencia

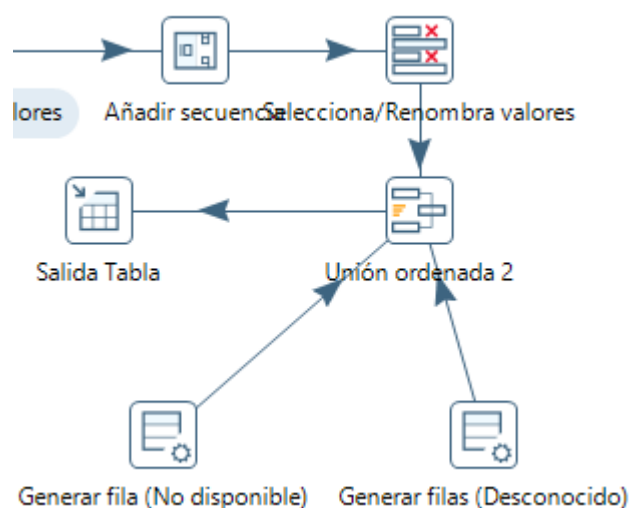
Nombre de campo destino (vacío=sobreescribir): codigo\_conferencia

Default upon non-matching:

Valores de campo:

#	Valor origen	Valor destino
1	CONFERENCIA OESTE	CO
2	CONFERENCIA ESTE	CE

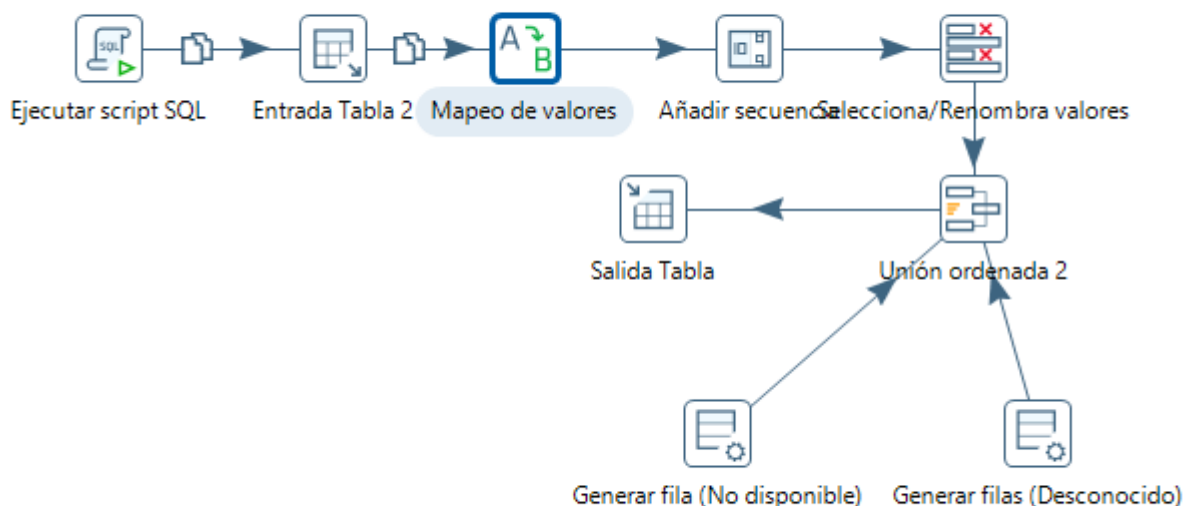
Se añade la secuencia de la clave primaria, se seleccionan los campos y se unen con los registros especiales creados. Estos pasos son configurados de la misma manera que en las anteriores transformaciones.



A modo de resumen, en esta transformación se realizaron los siguientes pasos.

- Eliminación del contenido de la tabla dimensión.
- Lectura de las tablas STG\_NBA\_STATS para obtener los datos.
- Mapeo de valores.
- Añadir secuencia de la clave primarias.
- Renombrar y seleccionar los campos que irán a la tabla.
- Crear los registros especiales.

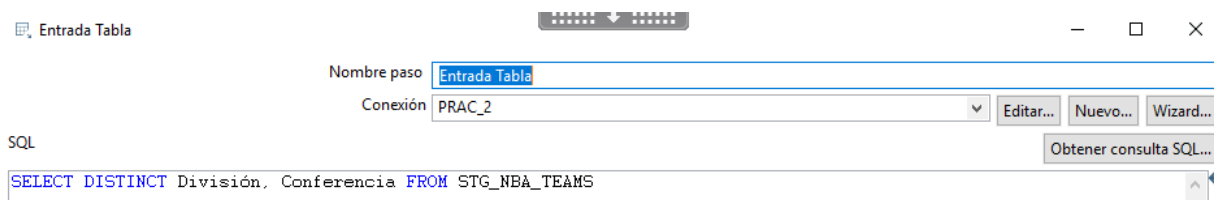
- Unir los diferentes flujos.
- Carga de los datos al modelo multidimensional.



## Transformaciones bloque TR\_DIM\_Divisiones

Este bloque contiene las transformaciones necesarias para realizar la carga inicial de la dimensión Divisiones en el almacén, esta carga se realiza desde la tabla intermedia “STG\_NBA\_TEAMS”

Inicialmente se eliminan los registros existentes en la tabla, se leen los datos de la tabla del área intermedia. Solamente se carga el campo División.



Posteriormente para crear el campo `id_conferencia` se realiza una búsqueda en la base de datos en la tabla `DIM_Conferencias` y mapeando los campos `Conferencia` del flujo con los de la tabla y se obtiene el valor de `pk_conferencia` y se almacena como `id_conferencia`.

Búsqueda de valor en base de datos (No responde)

Nombre paso: Búsqueda en base de datos

Conexión: PRAC\_2 [Editar...] [Nuevo...]

Esquema de búsqueda: dbo

Tabla de búsqueda: DIM\_Conferencias

Habilitar cache? ☐

Tamaño de cache en filas (0=todas): 0

Load all data from table ☐

La clave(s) para realizar búsqueda de valor(es):

#	Campo de tabla	Comparador	Campo1	Campo2
1	Conferencia	=	Conferencia	

Valores a devolver de la tabla de búsqueda:

#	Campo	Nuevo nombre	Defecto	Tipo
1	pk_conferencia	id_conferencia		Integer

Posteriormente se realiza un mapeo de valores para asignar a cada conferencia un código.



**Mapeo de valores**

Nombre de paso :

Nombre de campo origen :

Nombre de campo destino (vacío=sobrescribir) :

Default upon non-matching :

Valores de campo:

#	Valor origen	Valor destino
1	ATLÁNTICO	ATL
2	CENTRAL	CNT
3	NOROESTE	NOW
4	PACÍFICO	PAC
5	SURESTE	SUE
6	SUROESTE	SUW

Se genera la secuencia que hace de clave primaria.

**Obtener valor de la secuencia de la base de datos**

Nombre de paso :

Nombre de valor :

Utilizar una base de datos para generar la secuencia

¿Utilizar base datos para obtener secuencia? ☐

Conexión :

Nombre de esquema :

Nombre de secuencia :

Utilizar un contador de la transformación para generar la secuencia

¿Utilizar contador para calcular secuencia? ☒

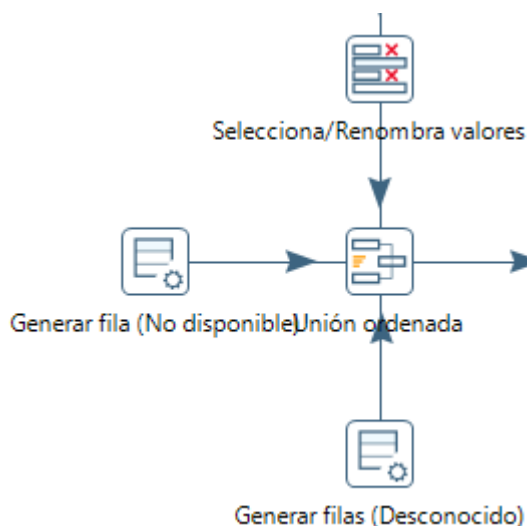
Nombre contador (opcional) :

Valor inicial :

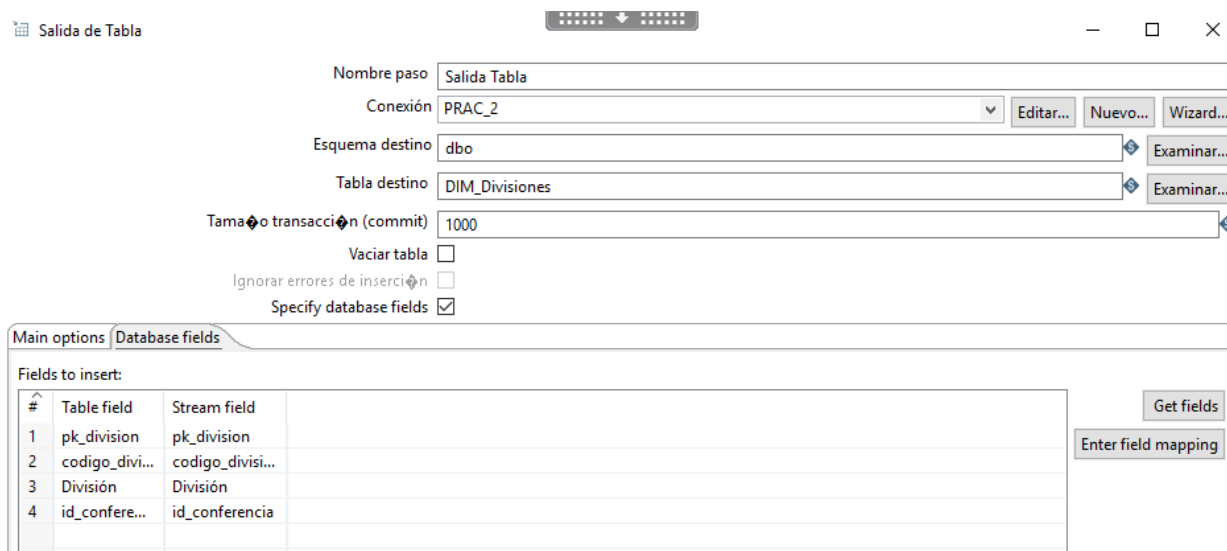
Incremento :

Valor máximo :

Como último segmento antes de la carga se aplican los pasos que seleccionan los campos, crean los registros especiales y unen los flujos.



Finalmente se cargan los datos a la tabla.



Salida de Tabla

Nombre paso: Salida Tabla

Conexión: PRAC\_2 [Editar...] [Nuevo...] [Wizard...]

Esquema destino: dbo [Examinar...]

Tabla destino: DIM\_Divisiones [Examinar...]

Tamaño de transacción (commit): 1000

Vaciar tabla: ☐

Ignorar errores de inserción: ☐

Specify database fields: ☒

Main options Database fields

Fields to insert:

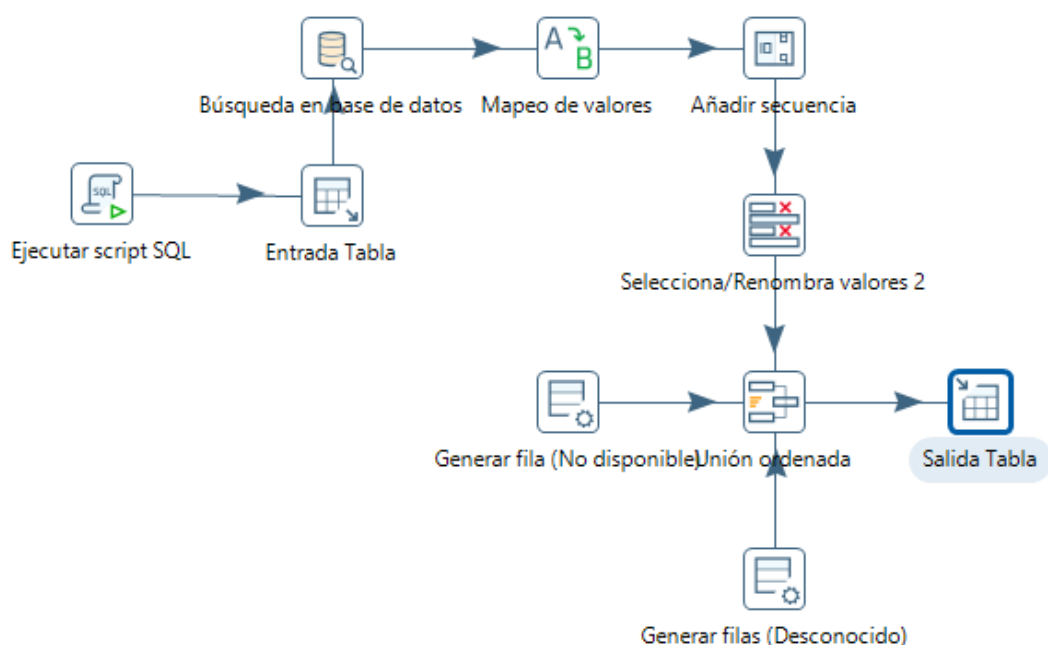
#	Table field	Stream field
1	pk_division	pk_division
2	codigo_divi...	codigo_divisi...
3	División	División
4	id_confere...	id_conferencia

[Get fields] [Enter field mapping]

A modo de resumen, en esta transformación se realizaron los siguientes pasos.

- Eliminación del contenido de la tabla dimensión.
- Lectura de las tablas STG\_NBA\_TEAMS para obtener los datos.

- Búsqueda de id\_conferencia.
- Mapeo de valores.
- Añadir secuencia de clave primaria.
- Seleccionar y renombrar los campos.
- Crear los registros especiales.
- Unión de los diferentes flujos de datos.
- Carga de los datos al modelo multidimensional.



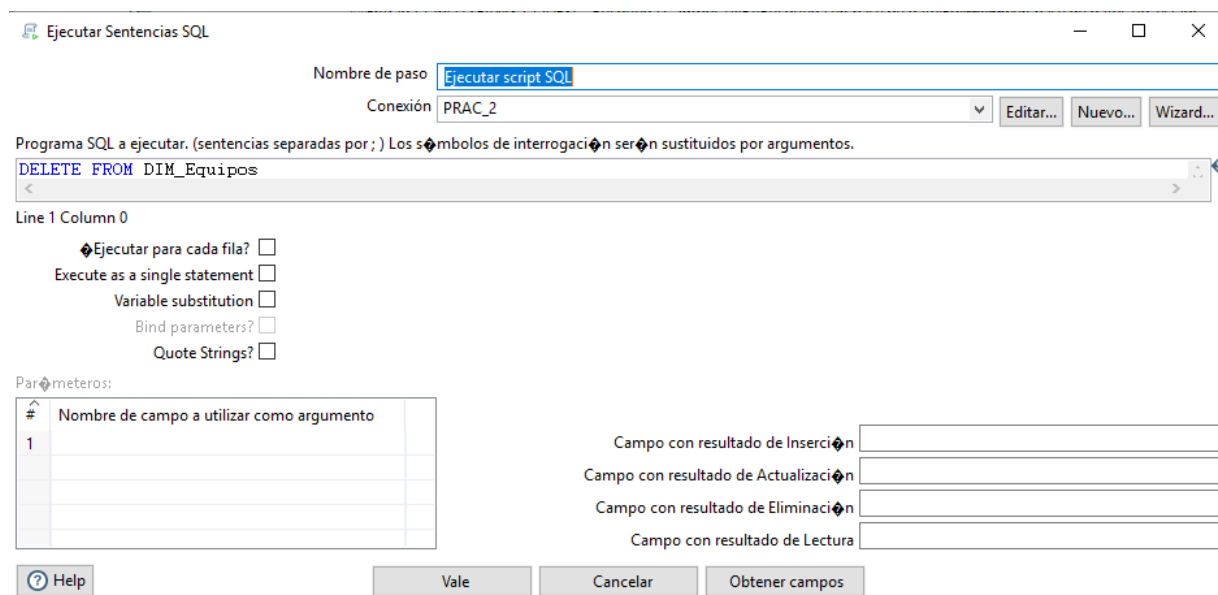
## Transformaciones bloque TR\_DIM\_Equipos

Este bloque contiene las transformaciones necesarias para realizar la carga inicial de la dimensión Equipos en el almacén, esta carga se realiza desde las tablas intermedias “STG\_TEAMS\_CODES”, “STG\_NBA\_TEAMS” y “STG\_WNBA\_TEAMS”, tendrá como destino la tabla DIM\_Equipos.

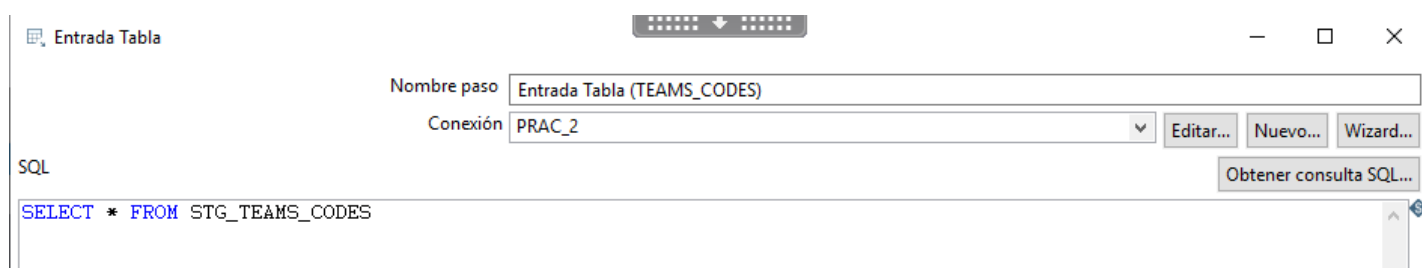
La estrategia tomada en esta transformación es cargar inicialmente todos los equipos desde la tabla que contiene el mapeo de los códigos y la liga, y luego mediante pasos de

Búsquedas en las Bases de Datos ir rellenando los campos necesarios para cada uno de estos equipos.

Los primeros dos pasos son los mismos que en el paso anterior, el *script* que elimina el contenido de la tabla DIM\_Equipos y el paso Entrada tabla que lee de la tabla del *Stagin Area*.



En el caso de la lectura de la tabla se selecciona todo su contenido, ya que se ha comprobado de análisis anteriores que no cuenta con suplicados y que se desea utilizar todos sus registros.



Luego se utilizan tres pasos de búsqueda en base de datos, los dos primeros para extraer la información de cada uno de los equipos de la NBA y la WNBA y el tercero para obtener el identificador del estado al que pertenecen. Como se puede apreciar en la siguiente figura se utilizan los campos "Equipo" de la tabla del *Stagin* y el campo team del flujo de datos para obtener las relaciones y entonces se obtienen los campos Ciudad, Pabellón, Fundado, Patrocinio, División y Estado.

Búsqueda de valor en base de datos

Nombre paso:

Conexión:

Esquema de búsqueda:

Tabla de búsqueda:

☒ Habilitar cache? ☐

Tamaño de cache en filas (0=todas):

☐ Load all data from table

La clave(s) para realizar búsqueda de valor(es):

#	Campo de tabla	Comparador	Campo1	Campo2
1	Equipo	=	team	

Valores a devolver de la tabla de búsqueda:

#	Campo	Nuevo nombre	Defecto	Tipo
1	Ciudad	ciudad_NBA		String
2	Pabellón	pabellon		String
3	Fundado	fundado		Integer
4	Patrocinio	patrocinio		String
5	División			String
6	Estado	cod_estado		String

Para el caso de la WNBA se utiliza la misma analogía para matchear los equipos, pero en esta tabla no aparecen los mismos campos que en la de la NBA, solo se obtienen los datos de los campos Ciudad y Estado, con lo cual muchos de estos atributos tendrán valor nulo en el *stream* de datos.

Búsqueda de valor en base de datos

Nombre paso:

Conexión:

Esquema de búsqueda:

Tabla de búsqueda:

☒ Habilitar cache?

Tamaño de cache en filas (0=todas):

☐ Load all data from table

La clave(s) para realizar búsqueda de valor(es):

#	Campo de tabla	Comparador	Campo1	Campo2
1	Equipo	=	team	

Valores a devolver de la tabla de búsqueda:

#	Campo	Nuevo nombre	Defecto	Tipo
1	Ciudad	ciudad_WNBA		String
2	Estado			String

☐ No procesar la fila si la búsqueda falla

☐ Producir error si se obtienen múltiples

Ordenar por:

La tercera de las búsquedas utiliza el campo código del flujo y el campo cod\_estado de la DIM\_Estados para obtener el campo id\_estado, a continuación se muestra este paso.

Búsqueda de valor en base de datos

Nombre paso: Búsqueda en base de datos (Estado)

Conexión: PRAC\_2 [Editar...] [Nuevo...] [Wizard...]

Esquema de búsqueda: dbo [Examinar...]

Tabla de búsqueda: DIM\_Estados\_EEUU [Examinar...]

Habilitar cache? ☐

Tamaño de cache en filas (0=todas): 0

Load all data from table ☐

La clave(s) para realizar búsqueda de valor(es):

#	Campo de tabla	Comparador	Campo1	Campo2
1	codigo	=	cod_estado	

Valores a devolver de la tabla de búsqueda:

#	Campo	Nuevo nombre	Defecto	Tipo
1	nombre	Estado		String

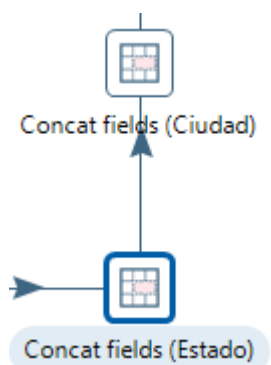
No procesar la fila si la búsqueda falla ☐

Producir error si se obtienen múltiples ☐

Ordenar por:

[?] Help [Vale] [Cancelar] [Obtener Campos] [Obtener Campos Búsqueda]

Como resultado de las búsquedas anteriores (NBA y WNBA) se han generado campos secundarios, que contienen la información de ambas ligas, de cada paso de búsqueda se genera un campo con el mismo nombre que el campo que se devuelve de la tabla en la que se realiza la búsqueda, por ese motivo en este punto de la transformación se cuenta con dos campos para las ciudades y dos campos que contiene los estados. Resaltar que estos campos son excluyentes entre sí, al provenir de tablas diferentes, y los registros que para unos tienen algún valor almacenados para el otro no lo tienen (null), con lo cual ejecutando pasos de Concatenar campos se obtiene como resultado un campo totalmente cubierto con todos los registros.



Posteriormente se realiza una nueva búsqueda en la tabla de la dimensión divisiones para obtener el campo id\_división, se conoce que para el caso de las estadísticas de la WNBA no se cuenta con este registro, con lo cual sus registros quedarán cubiertos a null.

Búsqueda de valor en base de datos

Nombre paso: Búsqueda en base de datos (Division)

Conexión: PRAC\_2 [Editar...] [Nuevo...] [Wizard...]

Esquema de búsqueda: dbo [Examinar...]

Tabla de búsqueda: DIM\_Divisiones [Examinar...]

Habilitar cache? ☐

Tamaño de cache en filas (0=todas): 0

Load all data from table ☐

La clave(s) para realizar búsqueda de valor(es):

#	Campo de tabla	Comparador	Campo1	Campo2
1	División	=	División	

Valores a devolver de la tabla de búsqueda:

#	Campo	Nuevo nombre	Defecto	Tipo
1	pk_division	id_division		None

No procesar la fila si la búsqueda falla ☐

Producir error si se obtienen múltiples ☐

Ordenar por:

[?] Help [Vale] [Cancelar] [Obtener Campos] [Obtener Campos Búsqueda]

Siguiendo una lógica similar a la anterior se obtiene el campo id\_estado, realizan una Búsqueda en la base de datos de la dimensión estados.



Búsqueda de valor en base de datos

Nombre paso:

Conexión:

Esquema de búsqueda:

Tabla de búsqueda:

☐ Habilitar cache?

Tamaño de cache en filas (0=todas):

☐ Load all data from table

La clave(s) para realizar búsqueda de valor(es):

#	Campo de tabla	Comparador	Campo1	Campo2
1	nombre	=	ref_estado	

Valores a devolver de la tabla de búsqueda:

#	Campo	Nuevo nombre	Defecto	Tipo
1	pk_estado_EEUU	id_estado_EEUU		Integer

☐ No procesar la fila si la búsqueda falla

☐ Producir error si se obtienen múltiples

Ordenar por:

Posteriormente y de forma similar a transformaciones anteriores se añade la secuencia que hace de pk\_equipo. Se seleccionan los campos necesarios para cargar a la base de datos y se renombran si es necesario quedando así ordenados y listos para cargar en la base de datos.

Selecciona/Renombra valores

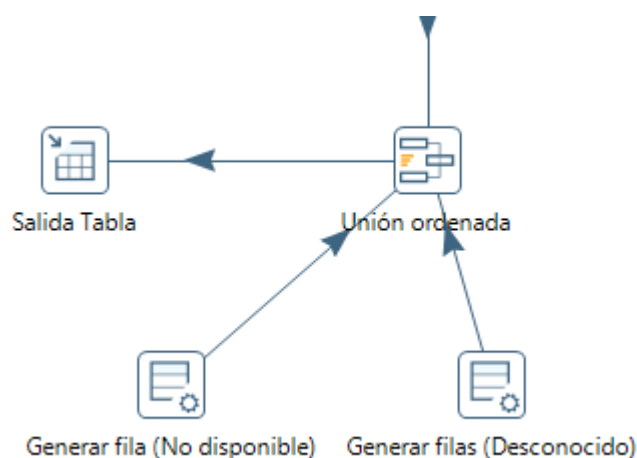
Nombre paso:

Selecciona & Modifica | Eliminar | Meta-información

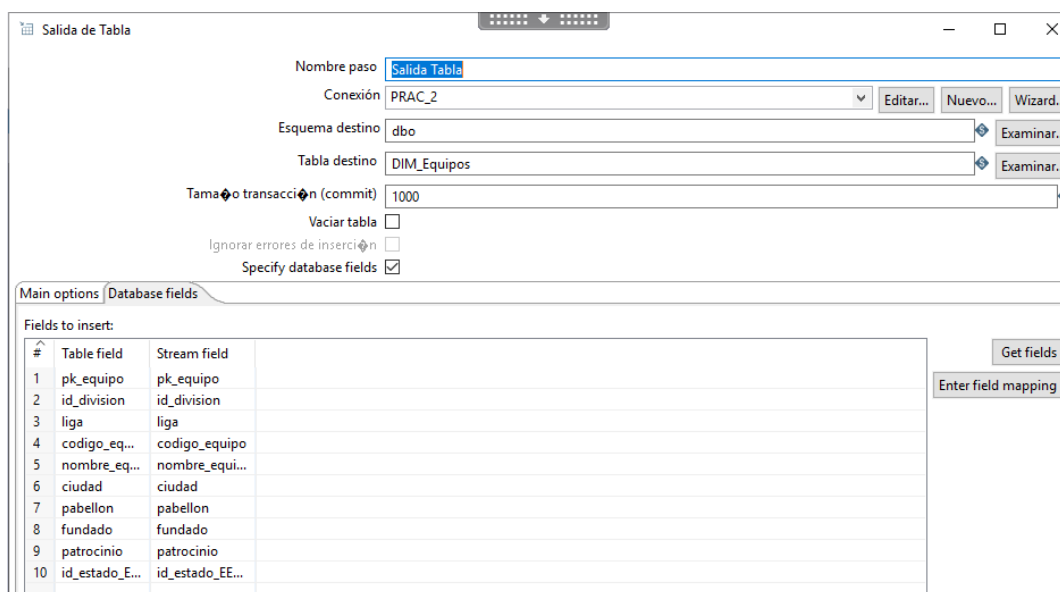
Campos:

#	Nombre campo	Renombrar a	Longitud	Precisión
1	pk_equipo			
2	id_estado_EEUU			
3	id_division			
4	league	liga		
5	code	codigo_equipo		
6	team	nombre_equipo		
7	ciudad			
8	pabellon			
9	fundado			
10	patrocinio			

Y para finalizar la transformación, se añaden las columnas para cubrir los casos especiales de los valores “desconocidos” y “no disponibles”.



Gracias a la limpieza de los campos y su renombramiento el mapeo en la Salida de Tabla es muy sencillo, quedando los datos almacenados correctamente en la tabla DIM\_Equipos.



Nombre paso: Salida Tabla

Conexión: PRAC\_2

Esquema destino: dbo

Tabla destino: DIM\_Equipos

Tamaño transacción (commit): 1000

Vaciar tabla: ☐

Ignorar errores de inserción: ☐

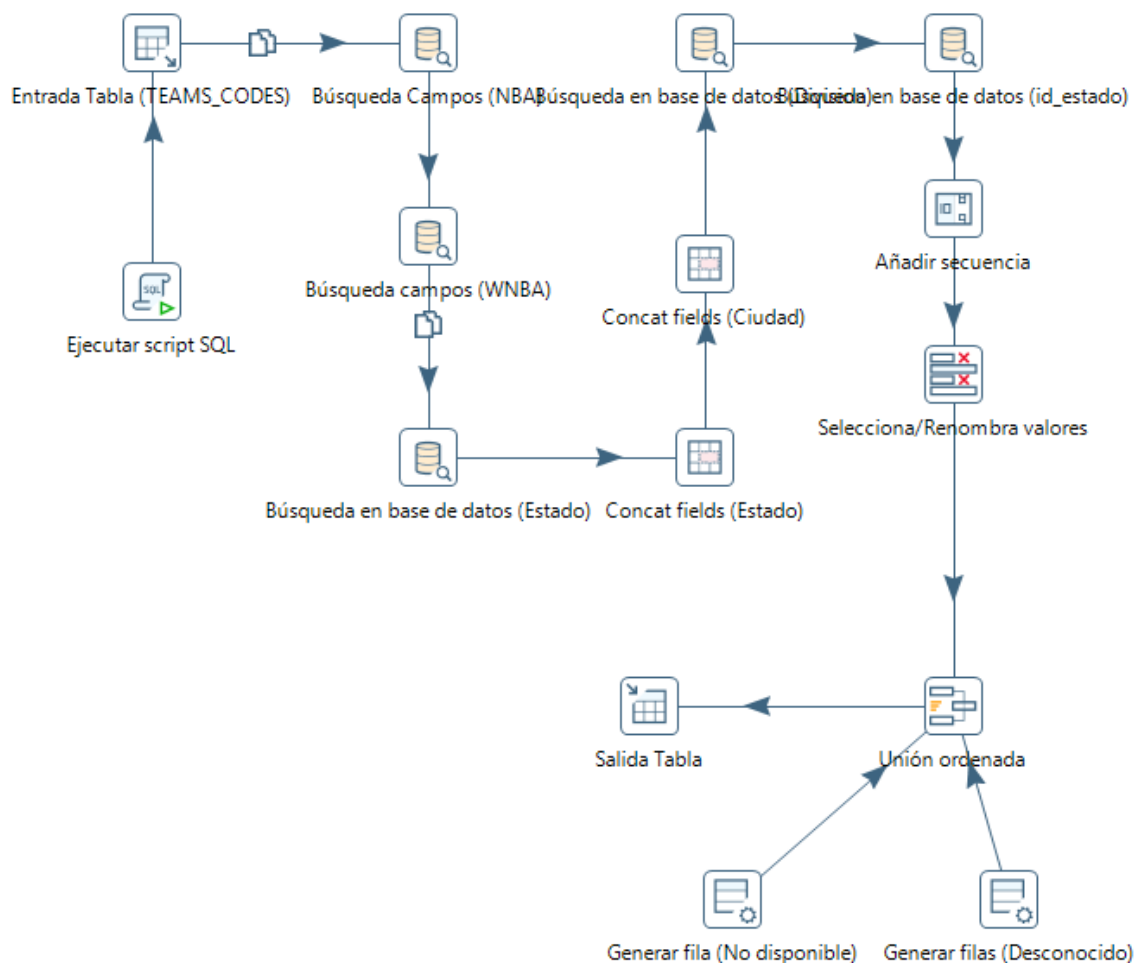
Specify database fields: ☒

#	Table field	Stream field
1	pk_equipo	pk_equipo
2	id_division	id_division
3	liga	liga
4	codigo_eq...	codigo_equipo
5	nombre_eq...	nombre_equi...
6	ciudad	ciudad
7	pabellon	pabellon
8	fundado	fundado
9	patrocinio	patrocinio
10	id_estado_E...	id_estado_EE...

A modo de resumen, en esta transformación se realizaron los siguientes pasos.

- Eliminación del contenido de la tabla dimensión.
- Lectura de la tabla STG\_TEAMS\_CODES para obtener los datos.
- Búsqueda en tabla para obtener los datos de los equipos de la NBA.
- Búsqueda en tabla para obtener los datos de los equipos de la WNBA.
- Búsqueda en tabla DIM\_Estados para obtener el valor de los estados a los que pertenecen los equipos.

- Dos campos de concatenación (Estado y Ciudad).
- Búsqueda en la tabla DIM\_División del campo id\_division.
- Búsqueda en la tabla DIM\_Estado del campo id\_estado.
- Generación de clave pk\_equipo.
- Creación de las filas especiales.
- Unión de los flujos.
- Carga de los datos al modelo multidimensional.



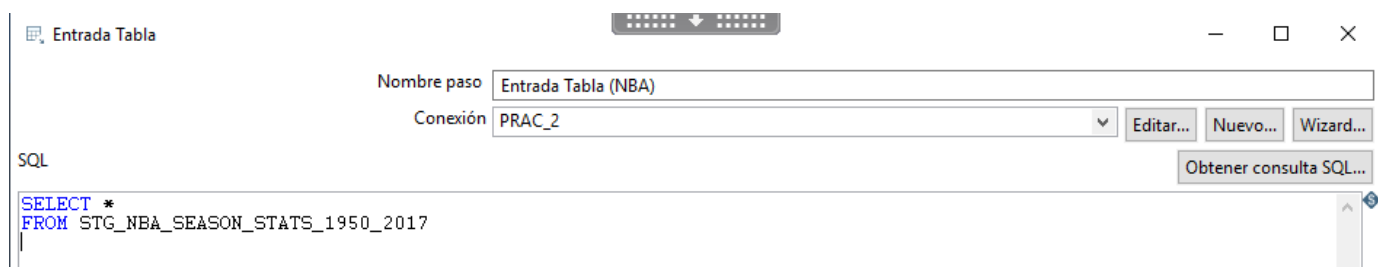
## Transformaciones TR\_FACT\_SEASONS\_STATS

En esta transformación se carga la tabla de hechos FACT\_SEASONS\_STATS del modelo multidimensional. EL origen de los datos serán las tablas de dimensiones definidas anteriormente y el procedimiento a seguir se define a continuación.

Inicialmente se ejecuta el script que elimina el contenido anterior de la tabla, y luego se crean dos flujos de datos, uno con los datos de la NBA y otro con los datos de la WNBA. Para facilitar la comprensión de este segmento se tratarán por separado.

### NBA

El primer paso en este flujo es leer la tabla STG\_NBA\_SEASON\_STATS\_1950\_2017 de la que inicialmente se obtiene todo el contenido.



Luego se agrega un paso Seleccionar/Renombrar para cambiar los nombres de algunas de las variables y que queden unificados a como se quieren en el modelo, esta tarea facilitará el desarrollo de los pasos posteriores.

Selecciona/Renombra valores

Nombre paso: Selecciona/Renombra valores (NBA)

Selecciona & Modifica | Eliminar | Meta-información

Campos:

#	Nombre campo	Renombrar a	Longitud	Precisión
1	Year	SEASON		
2	Player			
3	Tm	Team		
4	Pos	Position		
5	G	GP		
6	PTS			
7	MP	MIN		
8	FG	FGM		
9	FGA			
10	FG%			
11	_3P	_3PM		
12	_3PA			
13	_3P%			
14	_2P	_2PM		
15	_2PA			
16	_2P%			
17	FT	FTM		
18	FTA			
19	FT%			
20	eFG%			
21	ORB	OREB		
22	DRB	DREB		
23	TRB	REB		
24	AST			
25	TOV			
26	STL			
27	BLK			
28	PF			

Obtener campos a seleccionar

Edit Mapping

Luego mediante un paso Fórmula se agrega el campo EFF, la fórmula fue extraída del enunciado de la actividad.

fx Formula

Nombre paso: Formula (NBA)

Fields:

#	New field	Formula	Value type	Length	Precision	Replace value
1	EFF	((PTS) + (REB) + (AST) + (STL) + (BLK) - ((FGA) - (FGM)) - ((FTA) - (FTM)) - (TOV))	Number			

Luego se le cambia el tipo a algunas variables para lograr compatibilidad con el tipo de las variables en el modelo y se ordenan las filas, este es el último paso del segmento NBA.

Selecciona/Renombra valores

Nombre paso: Selecciona/Renombra (NBA\_ORDER)

Selecciona & Modifica | Eliminar | Meta-información

Campos a modificar meta información:

#	Nombre campo	Renombrar a	Tipo	Longitud	Precisión	Binary to Normal?	Format	Date Format
1	_2PM		Integer			N		N
2	_3PM		Integer			N		N
3	_3PA		Integer			N		N

## WNBA

Igualmente en este paso la primera tarea es leer la tabla STG\_WNBA\_SEASON\_STATS del área intermedia.

Entrada Tabla

Nombre paso: Entrada Tabla (WNBA)

Conexión: PRAC\_2

SQL

```
SELECT *
FROM STG_WNBA_SEASON_STATS_2005_2017
ORDER BY PLAYER
```

Obtener consulta SQL...

Luego se reordenan los campos y se agregan algunos que se pueden calcular a partir de otros existentes, esta tarea mediante el paso Fórmula.

Fórmula

Nombre paso: Fórmula (WNBA)

Fields:

#	New field	Formula	Value type	Length	Precision	Replace value
1	_2PM	((FGM) - [_3PM])	Integer			
2	_2PA	((FGA) - [_3PA])	Integer			
3	EFF	((PTS) + [REB] + [AST] + [STL] + [BLK] - ((FGA) - [FGM]) - ((FTA) - [FTM]) - [TOV])	Number			
4	Position	"Desconocido"	String			

Luego se calculan otros campos a partir de estos últimos.

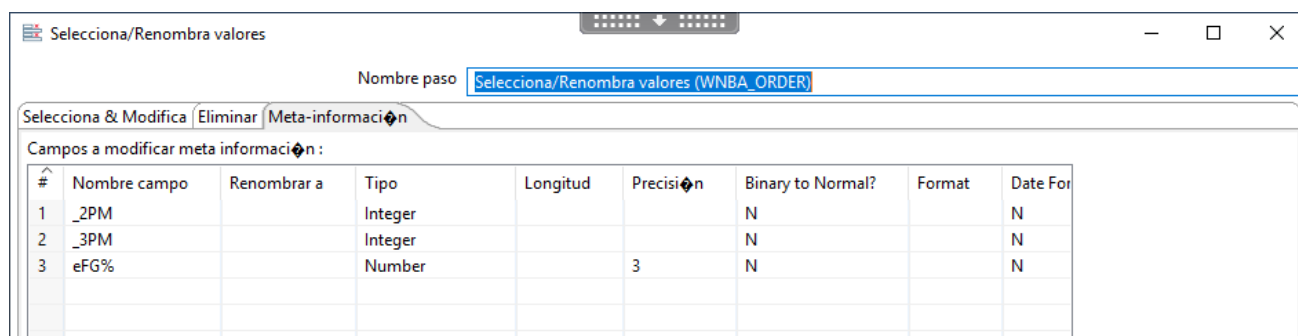
Fórmula

Nombre paso: Fórmula (Calculated Fields)

Fields:

#	New field	Formula	Value type	Length	Precision	Replace value
1	_2P%	((_2PA) - [_2PM])	Number			
2	eFG%	IF( ((FGA) + [FTA]) > 0; (((FGM) + [FTM]) / ((FGA) + [FTA])); 0)	Number			

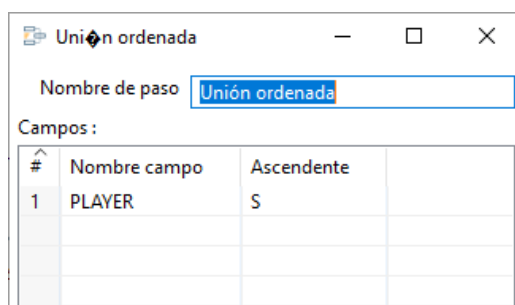
Al igual que en el flujo de la NBA se modifica el tipo de algunas variables y se ordenan los campos.



#	Nombre campo	Renombrar a	Tipo	Longitud	Precisión	Binary to Normal?	Format	Date For
1	_2PM		Integer			N		N
2	_3PM		Integer			N		N
3	eFG%		Number		3	N		N

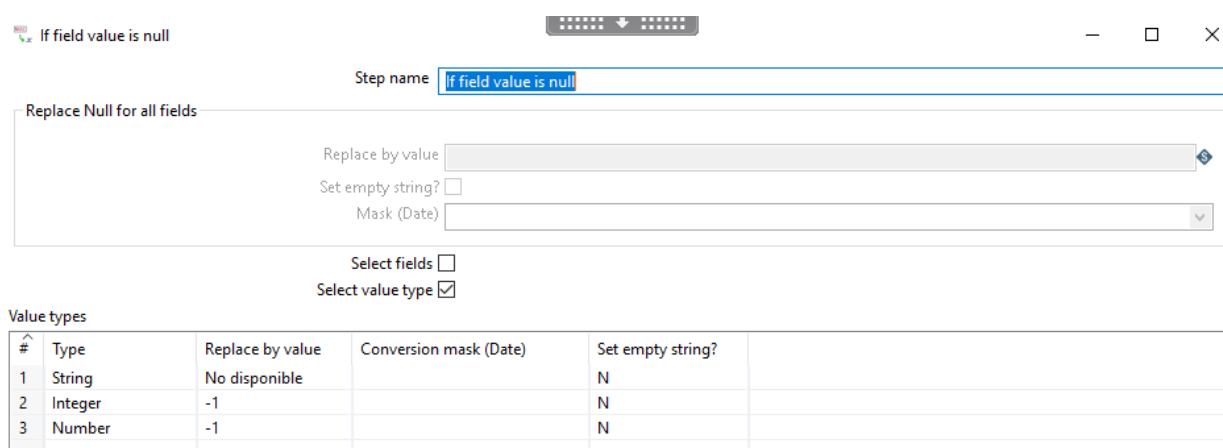
## Unión de ambos flujos de datos

Una vez se cuenta con ambos flujos listos es tiempo de unirlos, para ello se usa una unión ordenada tal y como se muestra a continuación.



#	Nombre campo	Ascendente
1	PLAYER	S

Luego se sustituyen los valores nulos de todo el flujo de datos según el tipo.



Step name: If field value is null

Replace Null for all fields

Replace by value: [Field Selection]

Set empty string? ☐

Mask (Date): [Date Mask]

Select fields ☐

Select value type ☒

#	Type	Replace by value	Conversion mask (Date)	Set empty string?
1	String	No disponible		N
2	Integer	-1		N
3	Number	-1		N

En este punto ya están listos los campos para referenciar las diferentes dimensiones que componen el hecho, para ello se utilizan las búsquedas en bases de datos para obtener los valores de los campos id\_season, id\_player, id\_team y id\_posición, a continuación se muestran las configuraciones de estas búsquedas.

## id\_season

Búsqueda de valor en base de datos

Nombre paso:

Conexión:

Esquema de búsqueda:

Tabla de búsqueda:

☒ Habilitar cache?

Tamaño de cache en filas (0=todas):

☒ Load all data from table

La clave(s) para realizar búsqueda de valor(es):

#	Campo de tabla	Comparador	Campo1	Campo2
1	date_year	=	SEASON	

Valores a devolver de la tabla de búsqueda:

#	Campo	Nuevo nombre	Defecto	Tipo
1	pk_date	id_season	0	Integer

## id\_player



## Búsqueda de valor en base de datos

Nombre paso

Conexión

[Editar...](#)

[Nuevo...](#)

[Wizard...](#)

Esquema de búsqueda



[Examinar...](#)

Tabla de búsqueda



[Examinar...](#)

☒ Habilitar cache?

Tamaño de cache en filas (0=todas)

☒ Load all data from table

La clave(s) para realizar búsqueda de valor(es):

#	Campo de tabla	Comparador	Campo1	Campo2
1	nombre	=	Player	

Valores a devolver de la tabla de búsqueda :

#	Campo	Nuevo nombre	Defecto	Tipo
1	pk_jugador	id_player	0	Integer
2	liga	league	Desconocido	String

id\_team

Búsqueda de valor en base de datos

Nombre paso:

Conexión:

Esquema de búsqueda:

Tabla de búsqueda:

☒ Habilitar cache?

Tamaño de cache en filas (0=todas):

☒ Load all data from table

La clave(s) para realizar búsqueda de valor(es):

#	Campo de tabla	Comparador	Campo1	Campo2
1	codigo_equipo	=	Team	

Valores a devolver de la tabla de búsqueda:

#	Campo	Nuevo nombre	Defecto	Tipo
1	pk_equipo	id_team	0	Integer

## Id\_posición

Búsqueda de valor en base de datos

Nombre paso:

Conexión:

Esquema de búsqueda:

Tabla de búsqueda:

☒ Habilitar cache?

Tamaño de cache en filas (0=todas):

☒ Load all data from table

La clave(s) para realizar búsqueda de valor(es):

#	Campo de tabla	Comparador	Campo1	Campo2
1	codigo_posicion	=	Position	

Valores a devolver de la tabla de búsqueda:

#	Campo	Nuevo nombre	Defecto	Tipo
1	pk_posicion_juego	id_position	0	Integer

Finalmente se cargan los datos en la tabla `FACT_SEASON_STATS`, usando el mapeo siguiente.

Salida de Tabla

Nombre paso: `Salida Tabla`

Conexión: `PRAC_2` [Editar...] [Nuevo...] [Wizard...]

Esquema destino: `dbo` [Examinar...]

Tabla destino: `FACT_SEASONS_STATS` [Examinar...]

Tamaño transacción (commit): `1000`

Vaciar tabla: ☐

Ignorar errores de inserción: ☐

Specify database fields: ☒

Main options | Database fields

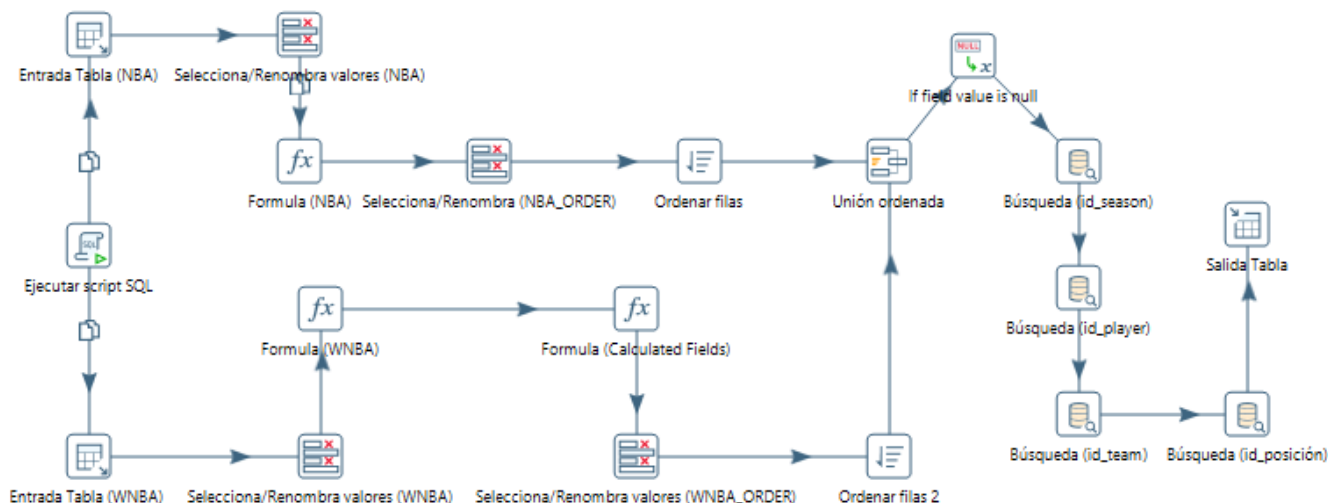
Fields to insert:

#	Table field	Stream field
1	GP	GP
2	PTS	PTS
3	MIN	MIN
4	FGM	FGM
5	FGA	FGA
6	FG%	FG%
7	_3PM	_3PM
8	_3PA	_3PA
9	_3P%	_3P%
10	_2PM	_2PM
11	_2PA	_2PA
12	_2P%	_2P%
13	FTM	FTM
14	FTA	FTA
15	FT%	FT%
16	eFG%	eFG%
17	OREB	OREB
18	DREB	DREB
19	REB	REB
20	AST	AST
21	TOV	TOV
22	STL	STL
23	BLK	BLK
24	PF	PF
25	EFF	EFF
26	id_season	id_season
27	id_player	id_player
28	id_team	id_team
29	id_position	id_position
30	league	league

[Get fields]

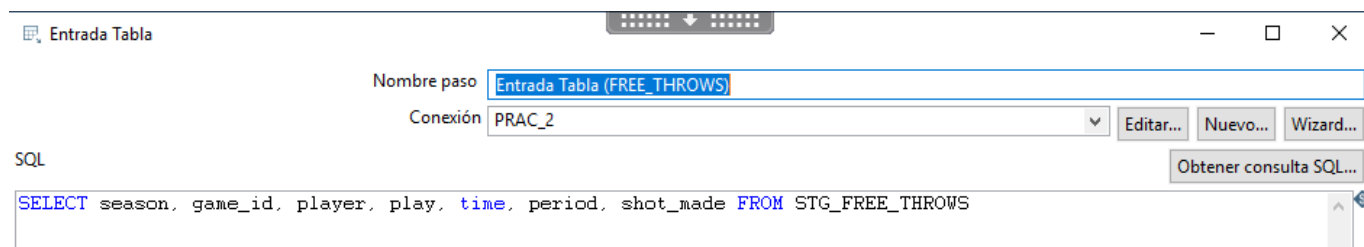
[Enter field mapping]

La transformación en su conjunto luce de la siguiente manera.



## Transformaciones del bloque TR\_FACT\_FREE\_THROWS

En este apartado se muestran las transformaciones para cargar la tabla de hechos FACT\_FREE\_THROWS, para ello se cargan los datos de varios atributos de la tabla STG\_FREE\_THROWS luego eso sí, de eliminar el contenido de la tabla con la ejecución del script SQL.



Posteriormente se utiliza un *replace string* similar al usado para conformar la dimensión Jugadas y se tratan los valores nulos.

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?
1	String	No disponible		N
2	Integer	-1		N
3	Number	-1		N

Gracias a que en esta tabla del *Stagin Area* los datos tienen buena calidad ahora solamente es necesario referenciar estos datos a cada una de las dimensiones que componen en hecho, y así obtener los diferentes identificadores.

## Id\_temporda

Búsqueda de valor en base de datos

Nombre paso: **Búsqueda en base de datos (Temporada)**

Conexión: **PRAC\_2**

Esquema de búsqueda: **dbo**

Tabla de búsqueda: **DIM\_Tiempo**

Habilitar cache? ☒

Tamaño de cache en filas (0=todas): **0**

Load all data from table ☒

La clave(s) para realizar búsqueda de valor(es):

#	Campo de tabla	Comparador	Campo1	Campo2
1	temporada	=	season	

Valores a devolver de la tabla de búsqueda:

#	Campo	Nuevo nombre	Defecto	Tipo
1	pk_date	id_temporada	0	Integer

## id\_partidos

Búsqueda de valor en base de datos

Nombre paso: Búsqueda en base de datos (Partido)

Conexión: PRAC\_2 Editar... Nuevo... Wizard...

Esquema de búsqueda: dbo Examinar...

Tabla de búsqueda: DIM\_Partidos Examinar...

☒ Habilitar cache?

Tamaño de cache en filas (0=todas): 0

☒ Load all data from table

La clave(s) para realizar búsqueda de valor(es):

#	Campo de tabla	Comparador	Campo1	Campo2
1	pk_partido	=	game_id	

Valores a devolver de la tabla de búsqueda:

#	Campo	Nuevo nombre	Defecto	Tipo
1	pk_partido	id_partido	0	Integer

## id\_jugador

Búsqueda de valor en base de datos

Nombre paso: Búsqueda en base de datos (Jugador)

Conexión: PRAC\_2 Editar... Nuevo... Wizard...

Esquema de búsqueda: dbo Examinar...

Tabla de búsqueda: DIM\_Jugadores Examinar...

☒ Habilitar cache?

Tamaño de cache en filas (0=todas): 0

☒ Load all data from table

La clave(s) para realizar búsqueda de valor(es):

#	Campo de tabla	Comparador	Campo1	Campo2
1	nombre	=	player	

Valores a devolver de la tabla de búsqueda:

#	Campo	Nuevo nombre	Defecto	Tipo
1	pk_jugador	id_jugador	0	Integer

## id\_minutos

Búsqueda de valor en base de datos

Nombre paso:

Conexión:

Esquema de búsqueda:

Tabla de búsqueda:

☒ Habilitar cache?

Tamaño de cache en filas (0=todas):

☒ Load all data from table

La clave(s) para realizar búsqueda de valor(es):

#	Campo de tabla	Comparador	Campo1	Campo2
1	minutoSegundo	=	time	

Valores a devolver de la tabla de búsqueda:

#	Campo	Nuevo nombre	Defecto	Tipo
1	pk_minutoSegundo	id_minuto	0	Integer

## id\_jugada

Búsqueda de valor en base de datos

Nombre paso:

Conexión:

Esquema de búsqueda:

Tabla de búsqueda:

☒ Habilitar cache?

Tamaño de cache en filas (0=todas):

☒ Load all data from table

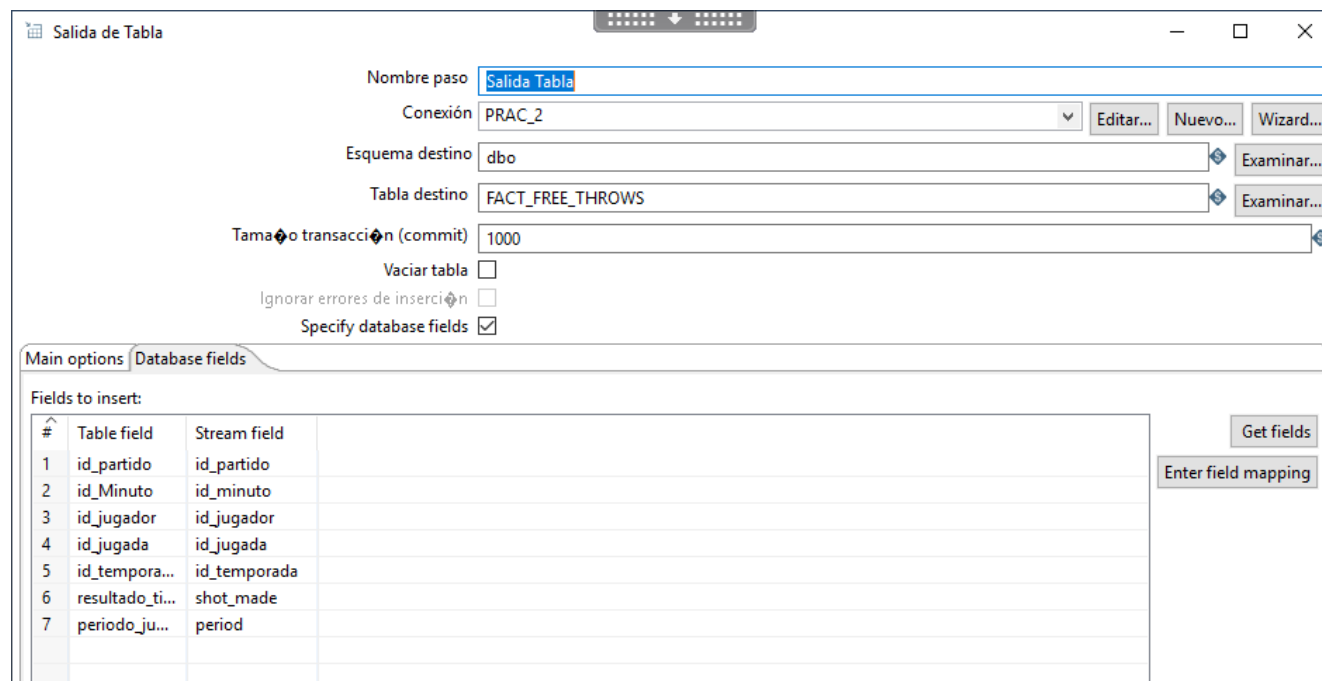
La clave(s) para realizar búsqueda de valor(es):

#	Campo de tabla	Comparador	Campo1	Campo2
1	desc_jugada	=	play	

Valores a devolver de la tabla de búsqueda:

#	Campo	Nuevo nombre	Defecto	Tipo
1	pk_jugada	id_jugada	0	Integer

Finalmetne se cargan los datos a la tabla correspondiente.



Salida de Tabla

Nombre paso: Salida Tabla

Conexión: PRAC\_2

Esquema destino: dbo

Tabla destino: FACT\_FREE\_THROWS

Tamaño transacción (commit): 1000

Vaciar tabla: ☐

Ignorar errores de inserción: ☐

Specify database fields: ☒

Main options Database fields

Fields to insert:

#	Table field	Stream field
1	id_partido	id_partido
2	id_Minuto	id_minuto
3	id_jugador	id_jugador
4	id_jugada	id_jugada
5	id_tempora...	id_temporada
6	resultado_ti...	shot_made
7	periodo_ju...	period

Get fields

Enter field mapping

### III. Implementación de trabajos con procesos ET

En este apartado se implementan los trabajos (*JOBS*) en la herramienta PDI, que permiten la ejecución de manera secuencial de todos los procesos ETL incluido en cada uno de los bloques (IN, TR). También se mostrará su rendimiento y un análisis de la volumetría de los elementos cargados.

#### Trabajo JOB\_IN

Este *job* se encarga de procesar todas las transformaciones del bloque IN, estas transformaciones a s vez son las encargadas de poblar las tablas del *Stagin Area* con los datos originales de las fuentes.

Los pasos que componen este *job* son los siguientes:

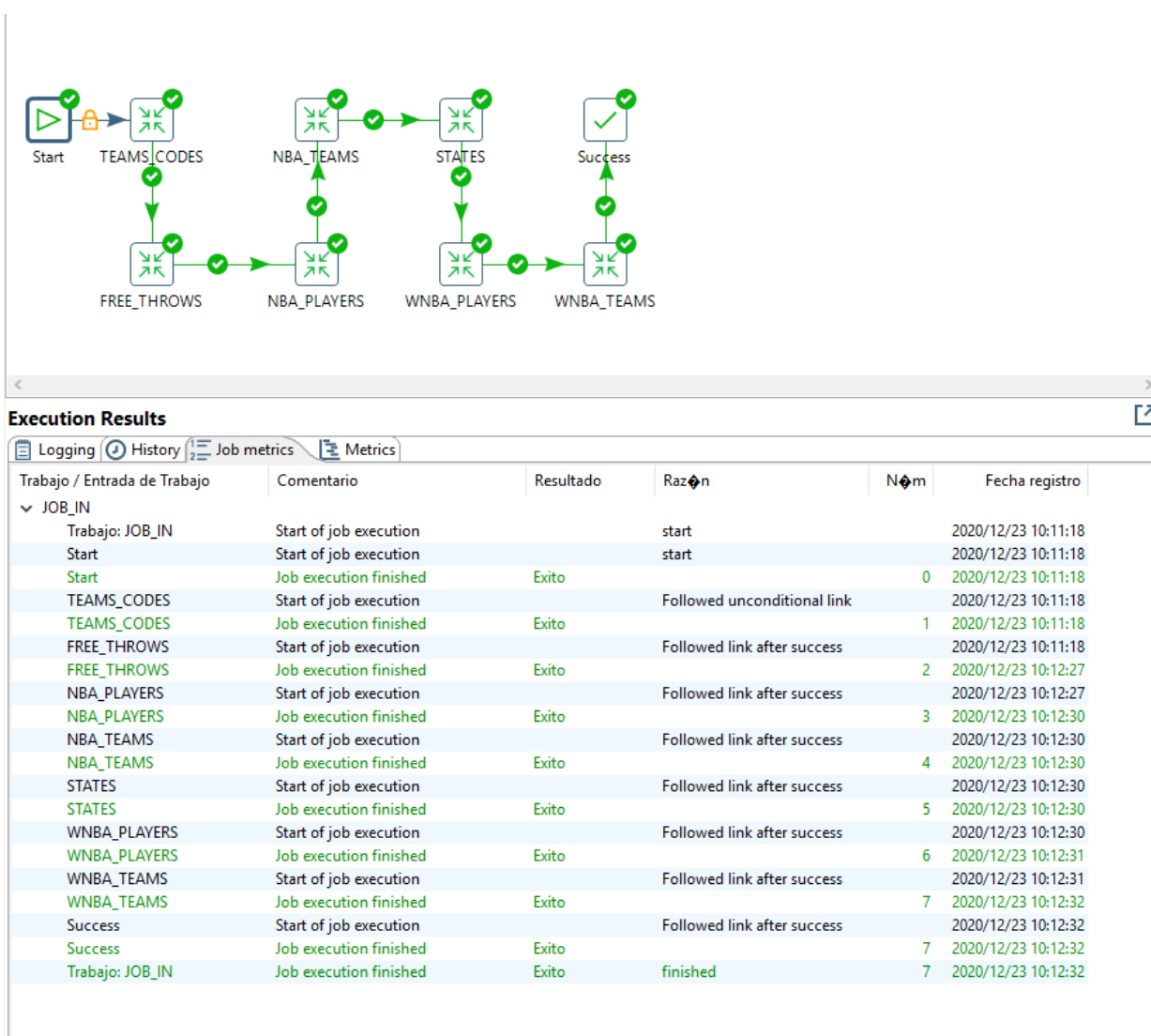
- Start: Componente general que determina el inicio del trabajo.
- TEAMS\_CODES: Ejecución de la transformación IN\_TEAMS\_CODES
- FREE\_THROWS: Ejecución de la transformación IN\_FREE\_THROWS
- NBA\_PLAYERS: Ejecución de la transformación IN\_NBA\_PLAYERS



- NBA\_TEAMS: Ejecución de la transformación IN\_NBA\_TEAMS
- STATES: Ejecución de la transformación IN\_STATES
- WNBA\_PLAYERS: Ejecución de la transformación IN\_WNBA\_PLAYERS
- WNBA\_TEMAS: Ejecución de la transformación IN\_WNBA\_TEMA
- Success: Componente general que determina la finalización del *job*.

## Ejecución

El resultado de la ejecución exitosa del *job* se muestra en la siguiente figura.



## Análisis de la volumetría

En el análisis de la volumetría se muestra la cantidad de registros almacenados en cada tabla.

Tabla	Número de registros
STG_FREE_THROWS	618 019
STG_NBA_PLAYERS	4 470
STG_NBA_SEASON_STATS	24 624
STG_NBA_TEAMS	30
STG_PLAYERS_DATA	4 639
STG_STATES	50
STG_TEAMS_CODES	42
STG_WNBA_PLAYERS	963
STG_WNBA_TEAMS	12
STG_WNBA_SEASON_STATS_2005_2017	2 085

## Trabajo JOB\_TR

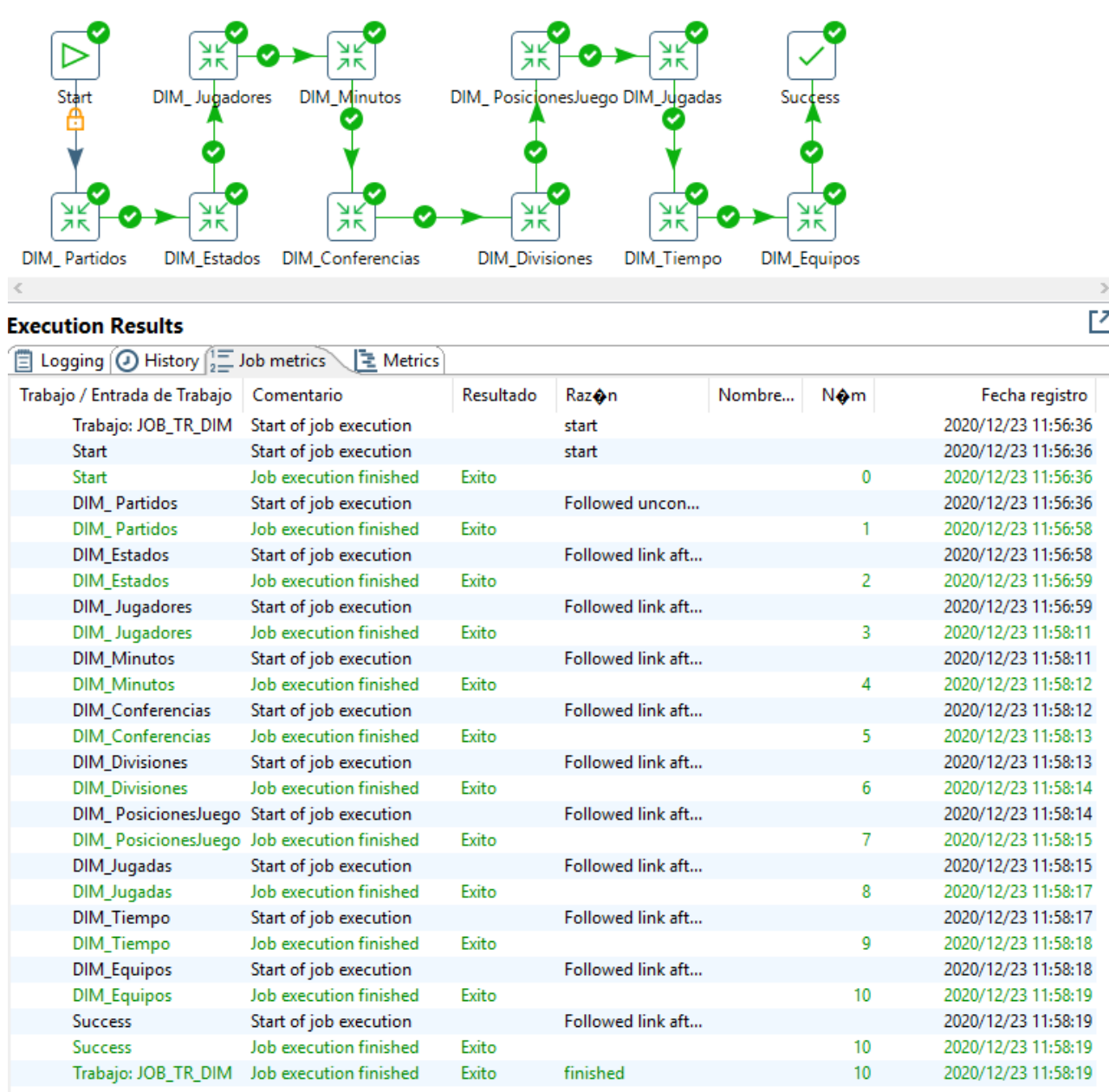
Este *job* se encarga de procesar todas las transformaciones del bloque TR encargadas de poblar las tablas de dimensiones del modelo multidimensional.

Los pasos que componen este *job* son los siguientes:

- Start: Componente general que determina el inicio del trabajo.
- DIM\_Partidos: Ejecución de la transformación TR\_DIM\_Partidos
- DIM\_Estados: Ejecución de la transformación TR\_DIM\_Estados
- DIM\_Jugadores: Ejecución de la transformación TR\_DIM\_Jugadores
- DIM\_Minutos: Ejecución de la transformación TR\_DIM\_Minutos
- DIM\_Conferencias: Ejecución de la transformación TR\_DIM\_Conferencias
- DIM\_Divisiones: Ejecución de la transformación TR\_DIM\_Divisiones.
- DIM\_PosicionesJuego: Ejecución de la transformación TR\_DIM\_PosicionesJuego
- DIM\_Jugadas: Ejecución de la transformación TR\_DIM\_Jugadas
- DIM\_Tiempo: Ejecución de la transformación TR\_DIM\_Tiempo
- DIM\_Equipos: Ejecución de la transformación TR\_DIM\_Equipos
- Success: Componente general que determina la finalización del *job*.

## Ejecución

El resultado de la ejecución exitosa del *job* se muestra en la siguiente figura.



## Análisis de la volumetría

En el análisis de la volumetría se muestra la cantidad de registros almacenados en cada tabla.

Tabla	Número de registros
DIM_Estados_EEUU	52
DIM_Jugadas	26
DIM_Jugadores	5 604
DIM_Minutos	536
DIM_Partidos	12 876
DIM_PosicionesJuego	25
DIM_Tiempo	70
DIM_Conferencias	4
DIM_Divisiones	8
DIM_Equipos	44

## Trabajo JOB\_FACT

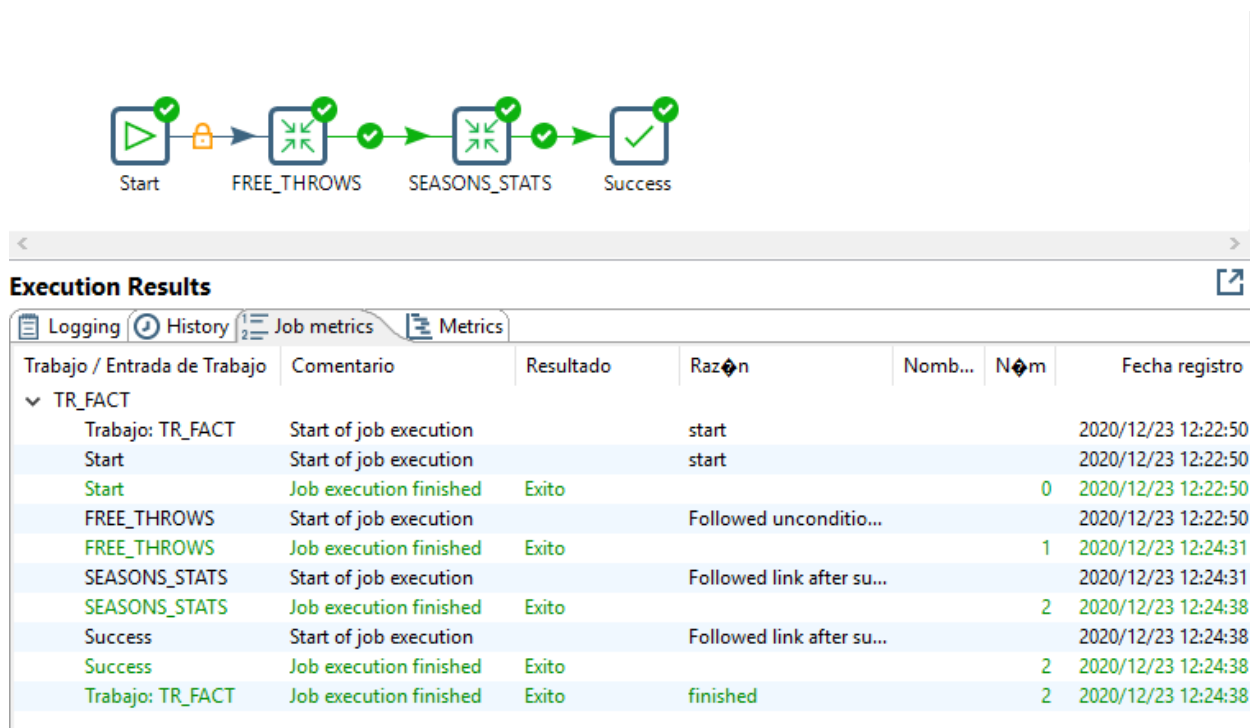
Este *job* se encarga de procesar todas las transformaciones del bloque TR que tienen la tarea de poblar las tablas de hechos del modelo multidimensional.

Los pasos que componen este *job* son los siguientes:

- Start: Componente general que determina el inicio del trabajo.
- FREE\_THROWS: Ejecución de la transformación TR\_FREE\_THROWS.
- SEASONS\_STATS: Ejecución de la transformación TR\_SEASONS\_STATS.
- Success: Componente general que determina la finalización del *job*.

## Ejecución

El resultado de la ejecución exitosa del *job* se muestra en la siguiente figura.



## Análisis de la volumetría

En el análisis de la volumetría se muestra la cantidad de registros almacenados en cada tabla.

Tabla	Número de registros
FACT_FREE_THROWS	618 019
FACT_SEASON_STATS	26 706

## Trabajo JOB\_ETL

Este trabajo procesa todos los trabajos con los procesos ETL relacionados con la construcción del almacén de datos para el análisis de estadísticas deportivas de las ligas de baloncesto WNBA y NBA



Los pasos incluidos en el trabajo JOB\_ETL son:

- Start: Componente general que determina el inicio del trabajo.
- JOB\_IN: Ejecución del trabajo JOB\_IN.
- JOB\_TR\_DIM: Ejecución del trabajo JOB\_TR\_DIM.
- JOB\_TR\_FACT: Ejecución del trabajo JOB\_TR\_FACT.
- Success: Componente general que determina la finalización del *job*.

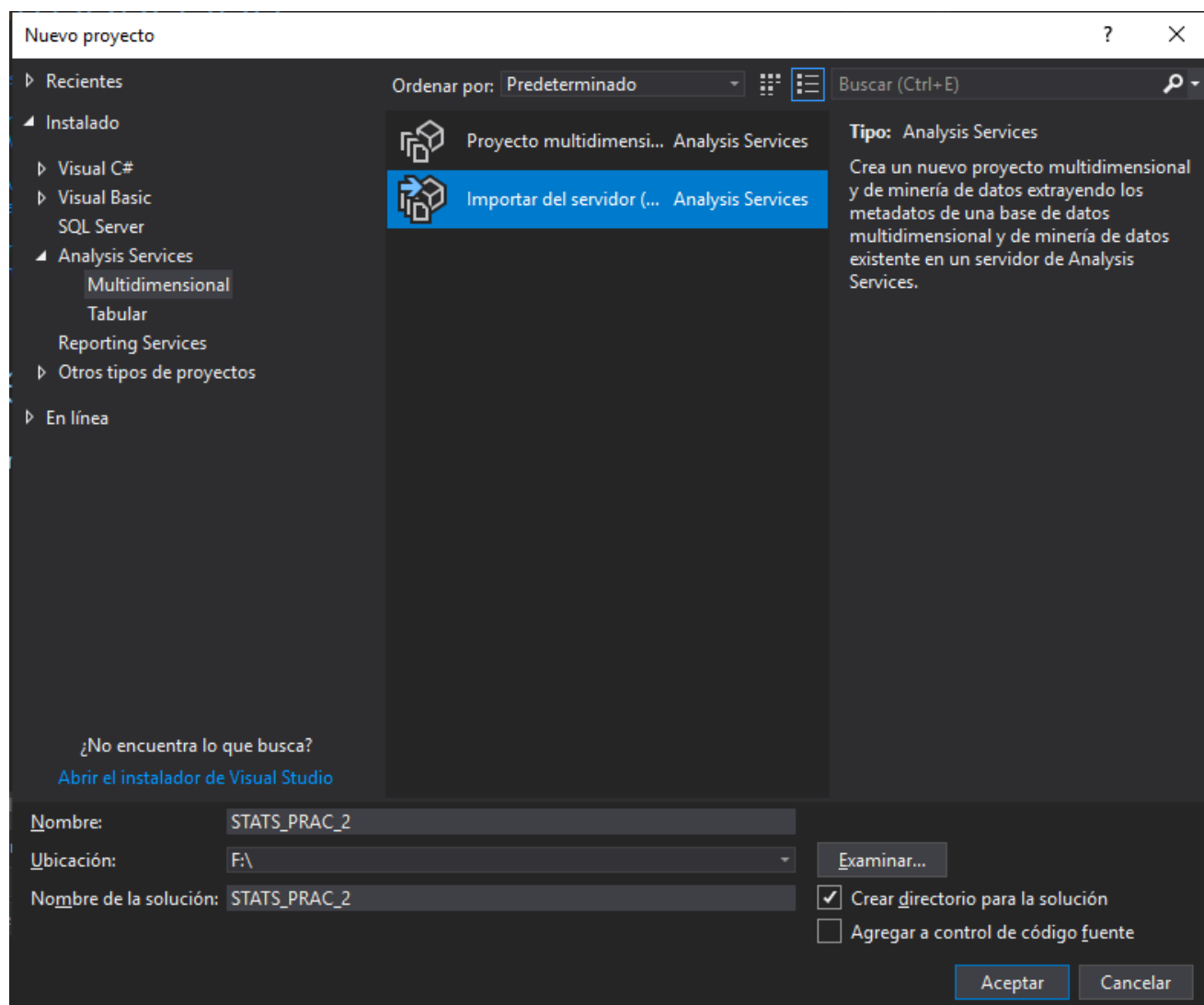
La ejecución completa y con éxito del JOB\_ETL supondrá la ejecución de los procesos ETL para la carga de los datos necesarios para construir el DW para el análisis de las estadísticas de la NBA y la WNBA desde la fuente de datos hasta las tablas del modelo multidimensional.

## IV. Diseño de modelo OLAP

En este apartado se procede a la creación del cubo multidimensional, pasando desde su creación hasta su explotación con varios ejemplos.

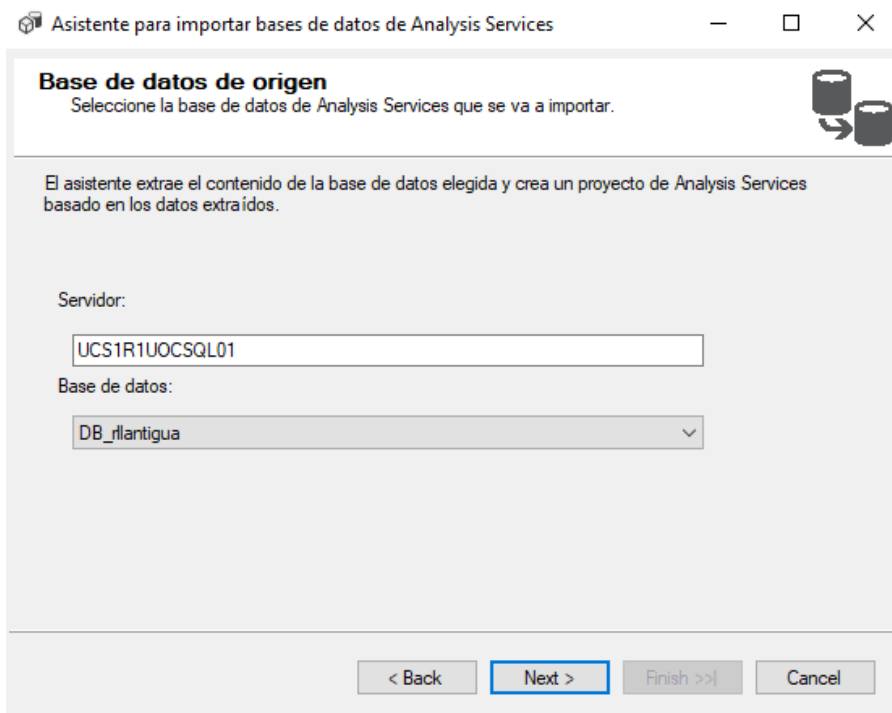
### Creación del proyecto

Una vez iniciado el Visual Studio, se selecciona la opción nuevo proyecto, este debe ser del tipo proyecto multidimensional y de minería de datos, luego se selecciona “Importar del Servidor”. En la figura que se muestra a continuación se ejemplifica esta acción, el nombre del proyecto es STATS\_PRAC\_2.

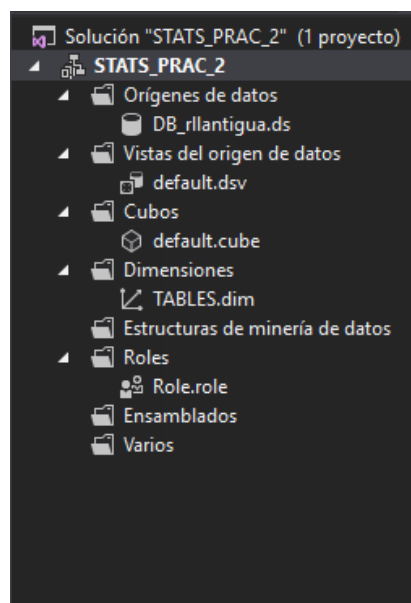


Luego de pulsar aceptar aparece un asistente en el que hay que introducir el nombre del servidor y de la base de datos y se pulsa en “Next”.



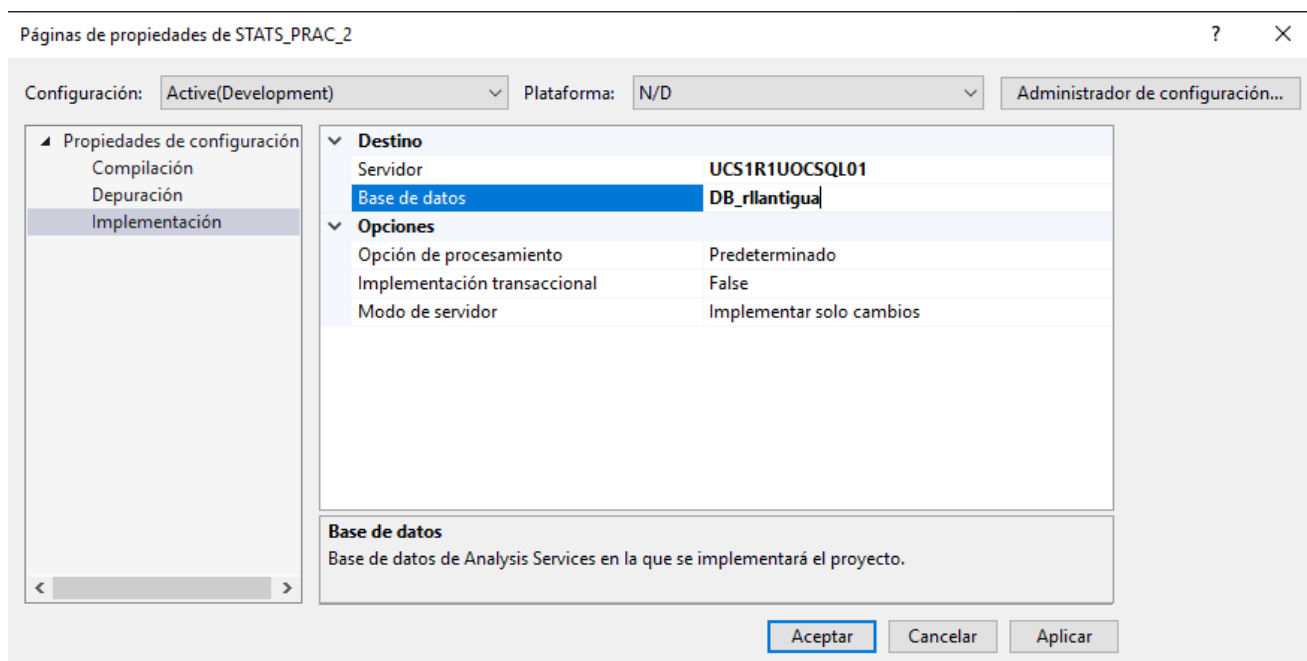


Luego de ejecutar este asistente se importa automáticamente la vista del origen de datos de nuestra base de datos junto a otros elementos por defecto.



## Destino de los datos

Es necesario también configurar el destino de los datos, para ello hay que pulsar con el clic derecho sobre el nombre del proyecto en el explorador de soluciones y dirigirse a la pestaña de propiedades. Una vez en propiedades es necesario seleccionar la parte de implementación y establecer el nombre del servidor y de la base de datos personal.



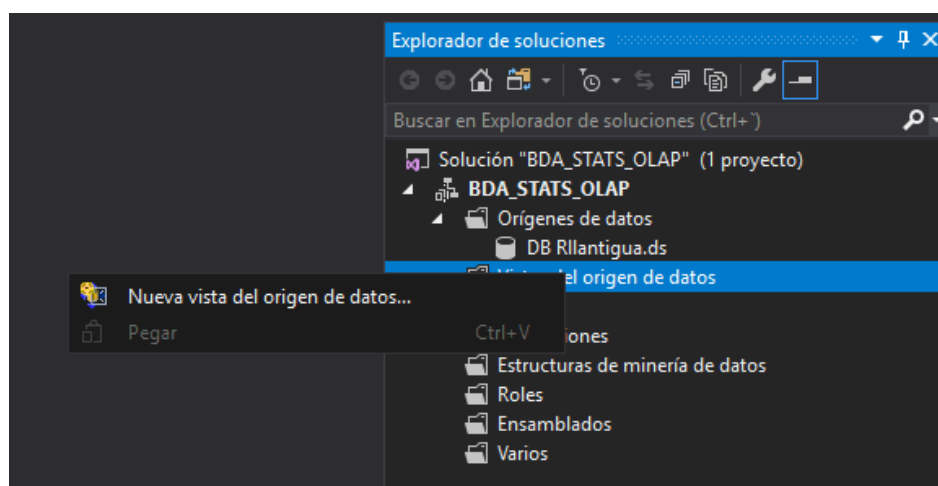
## Vista del origen de datos

Una vez se cuenta con la conexión es necesario crear las vistas del origen de datos, lo que define la estructura del cubo MOLAP y su alcance.

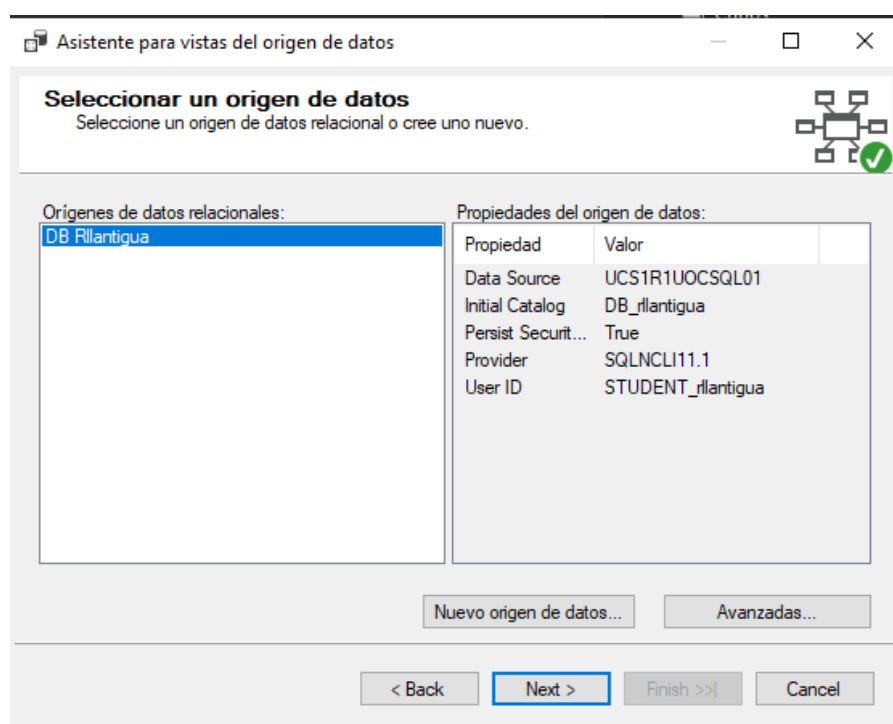
Para la implementación del caso en cuestión se definen 2 vistas de orígenes de datos:

- Vista SEASON\_STATS:
- Viasta FREE\_THROWS:

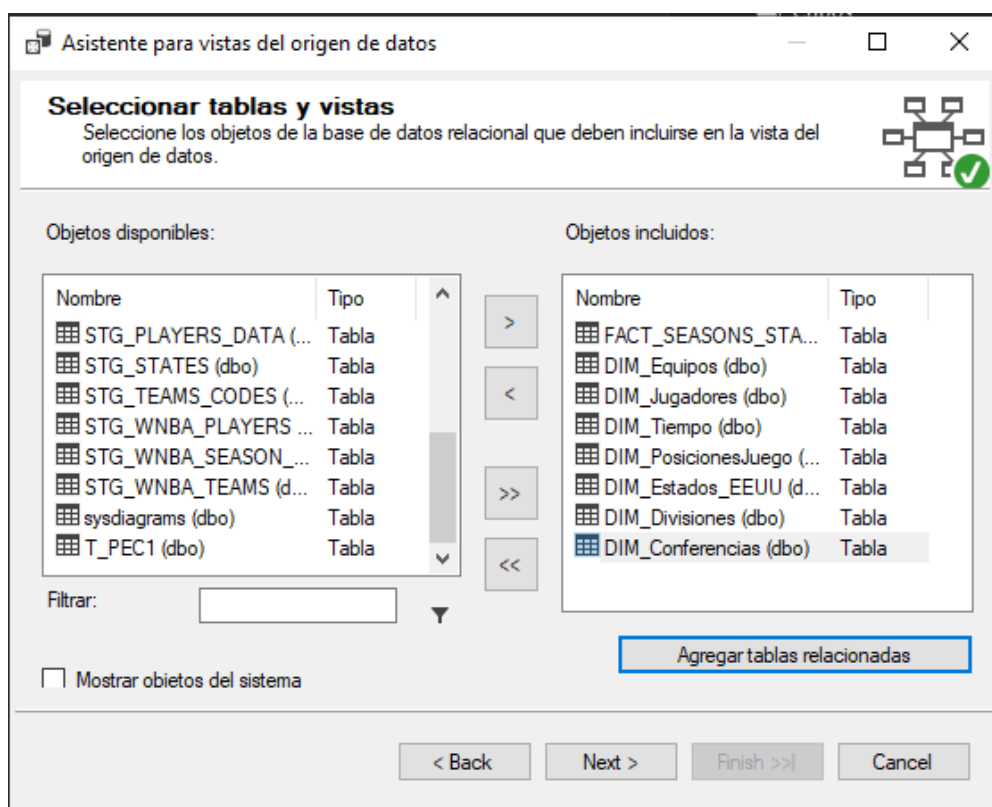
Para la creación es necesario navegar nuevamente por el “Explorador de soluciones” hasta el apartado correspondiente “Vista del origen de datos”, se pulsa clic derecho y se selecciona “Nueva vista del origen de datos..”.

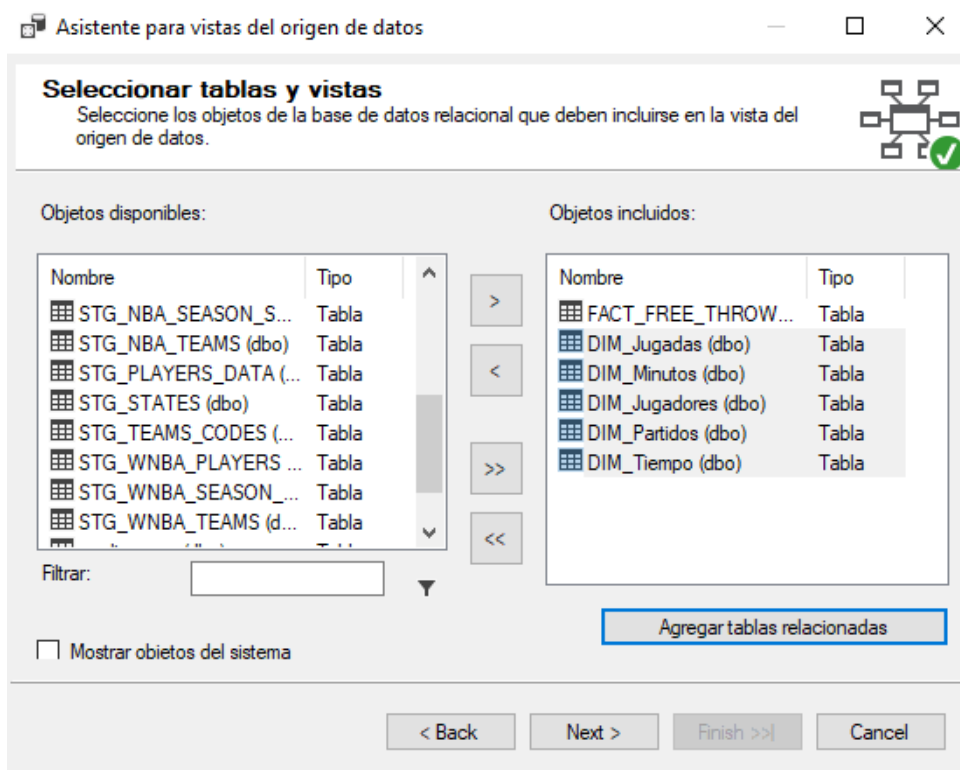


En la primera de las pestañas del asistente se selecciona el origen de datos creado anteriormente y se pulsa “Next”.



Entonces el asistente mostrará las tablas que componen el “Origen de los datos”, y de estas será necesario seleccionar las necesarias para cada una de las vistas. La manera más fácil de seleccionar las tablas relacionadas con la Vista que se desea crear es seleccionando la tabla de hechos y pulsando “Agregar tablas relacionadas”, por ejemplo en el caso de la tabla FACTS\_SEASON\_STATS al pulsar una vez se cargan tablas relacionadas a ella por una clave foránea (DIM\_Equipos, DIM\_Jugadores, DIM\_Tiempo, DIM\_PosicionesJuego) y siguiendo ese proceso se obtienen cada una de las tablas que componen la vista. El último paso es dotar a cada una de las vistas con un nombre, en este caso se les otorga el mismo nombre con el que se nombraron en la definición anterior. A continuación se muestran las tablas seleccionadas de cada una de las vistas, primero las pertenecientes a SEASON\_STATS y luego las de FREE\_THROWS.

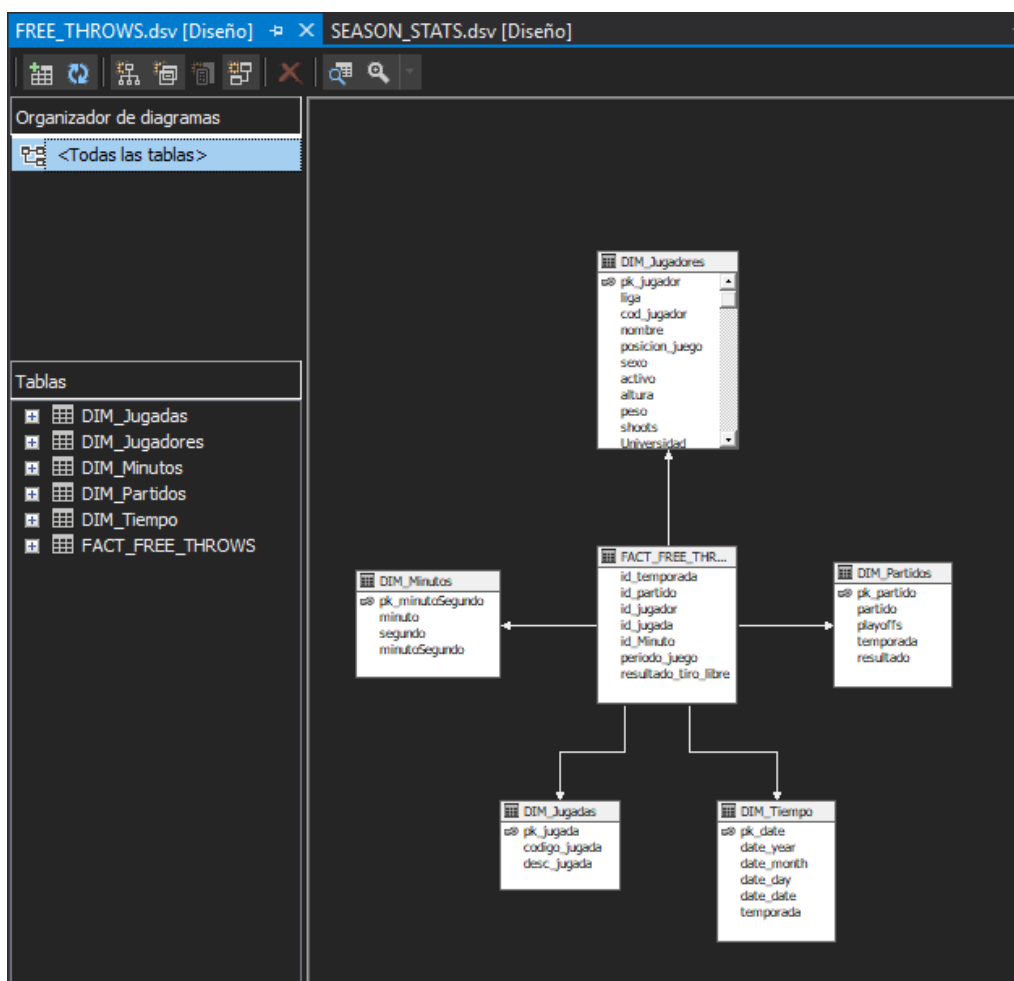




Una vez creadas las vistas se puede comprobar que están correctamente conformadas pulsando doble clic sobre su nombre en el “Explorador de soluciones”. Al hacer esto aparece en el área de trabajo del Visual Studio un diagrama en el que se muestran las relaciones entre las diferentes tablas de la vista, a continuación se muestra como luce la vista SEASON\_STATS.



Por su parte la vista FREE\_THROWS luce así:

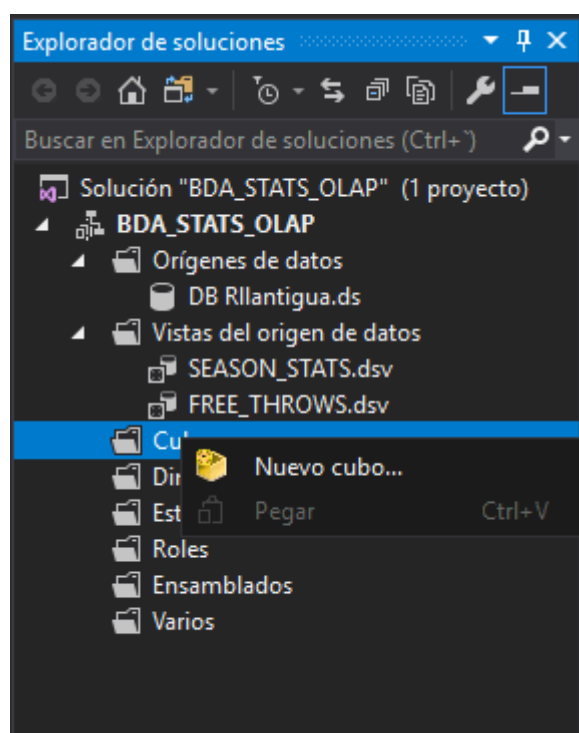


## Creación de cubos

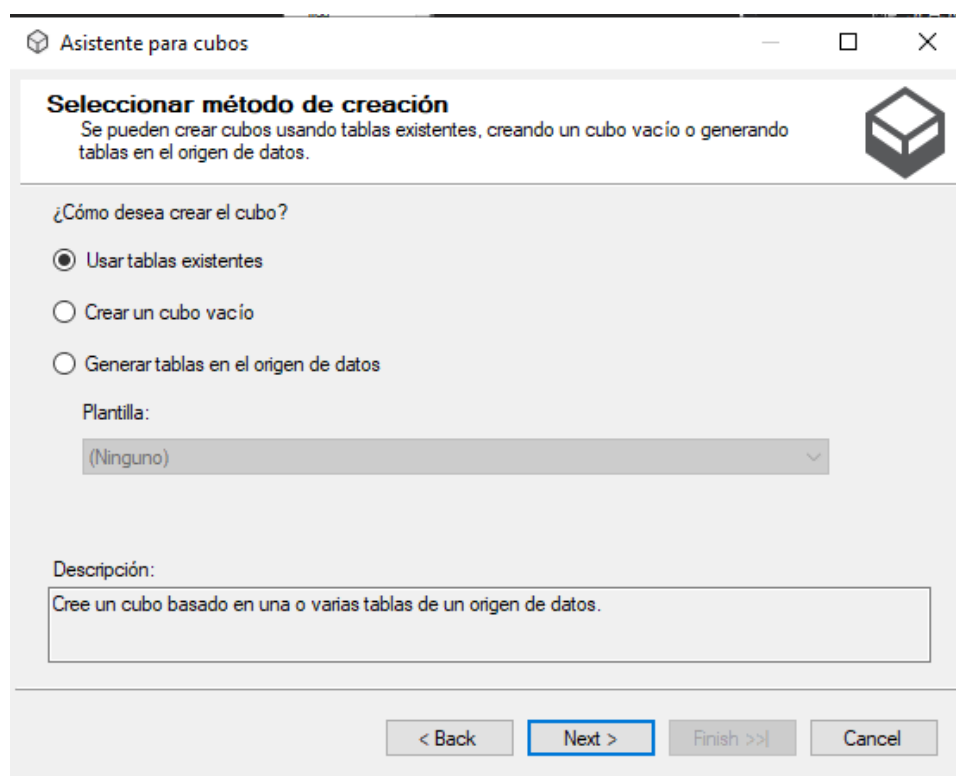
Con las vistas ya creadas es posible pasar a la creación de los diferentes cubos con sus correspondientes modelos multidimensionales. Será necesario crear un cubo para cada una de las vistas.

### Cubo SEASON\_STATS

Inicialmente se crea el cubo que permite realizar el análisis de las estadísticas, para crear el nuevo cubo se sigue un procedimiento similar al utilizado con los elementos anteriores, clic derecho en el elemento “Cubos”, y pulsar sobre “Nuevo cubo...”

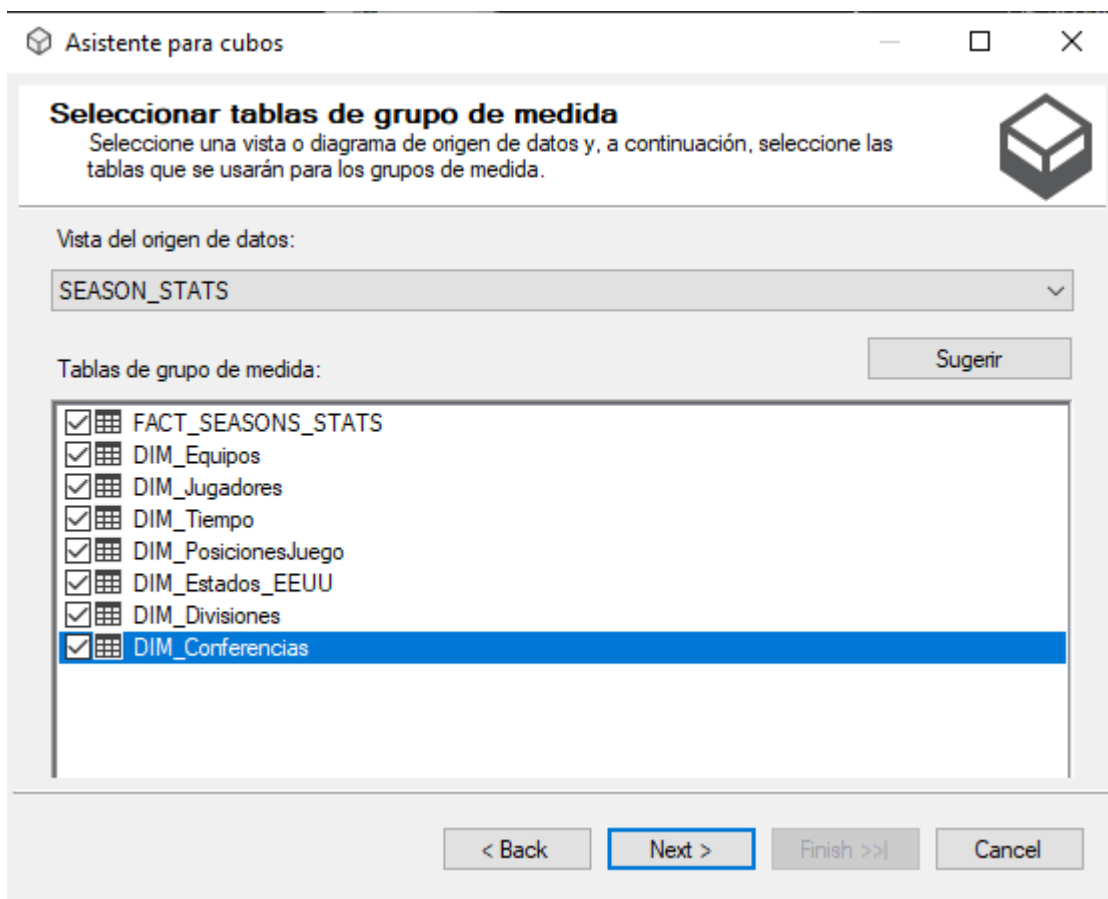


Se abre entonces un “Asistente para cubos”, y en la primera pestaña donde solicita un método de creación se selecciona “Usar tablas existentes” y se pulsa “Next”

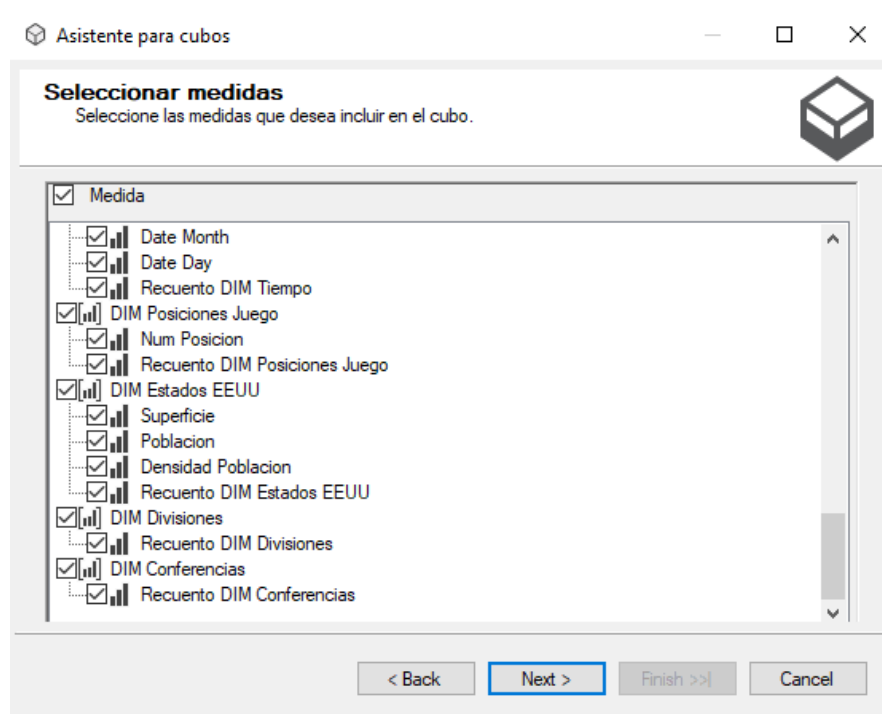




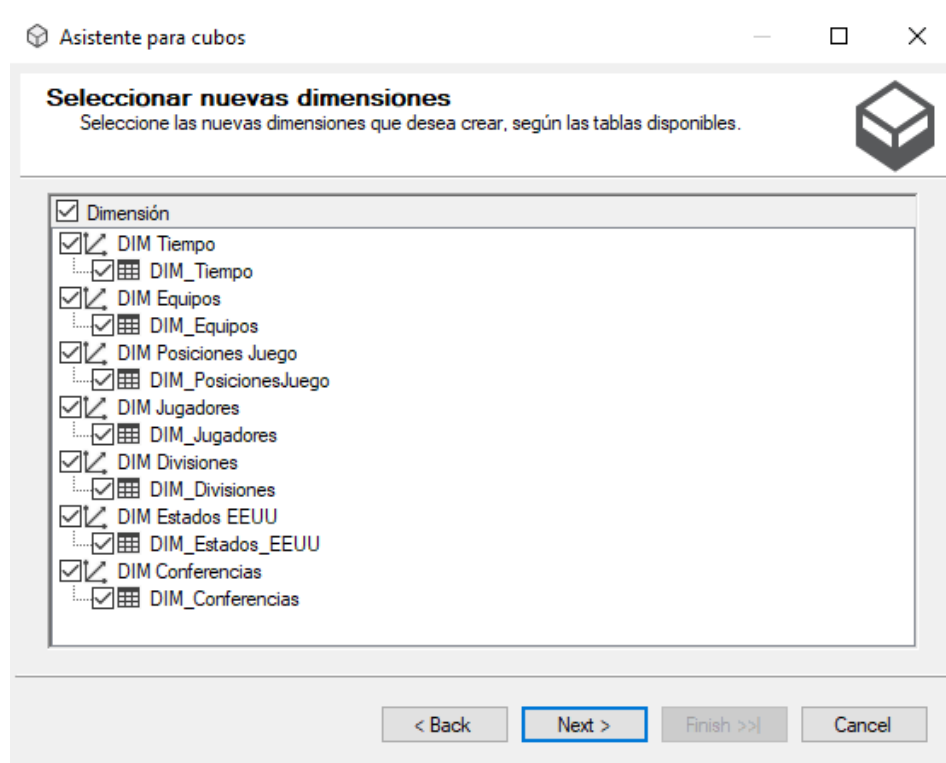
El asistente solicita entonces la elección de una vista del origen de datos, en este caso se selecciona la vista SEASON\_STATS. También es necesario elegir las tablas que componen el cubo.



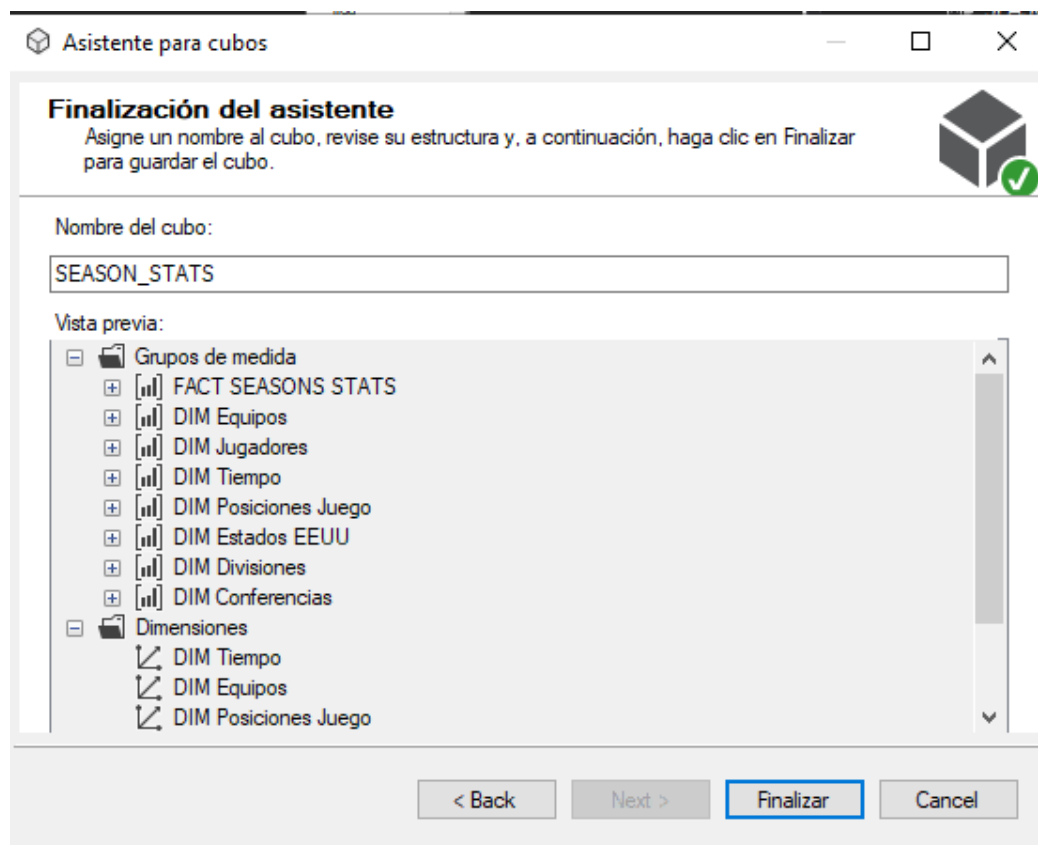
A continuación se seleccionan las métricas, en este caso se seleccionan todas las métricas que existen en las diferentes dimensiones, aunque inicialmente no todas sean necesarias, pueden ser de ayuda en futuras implementaciones.



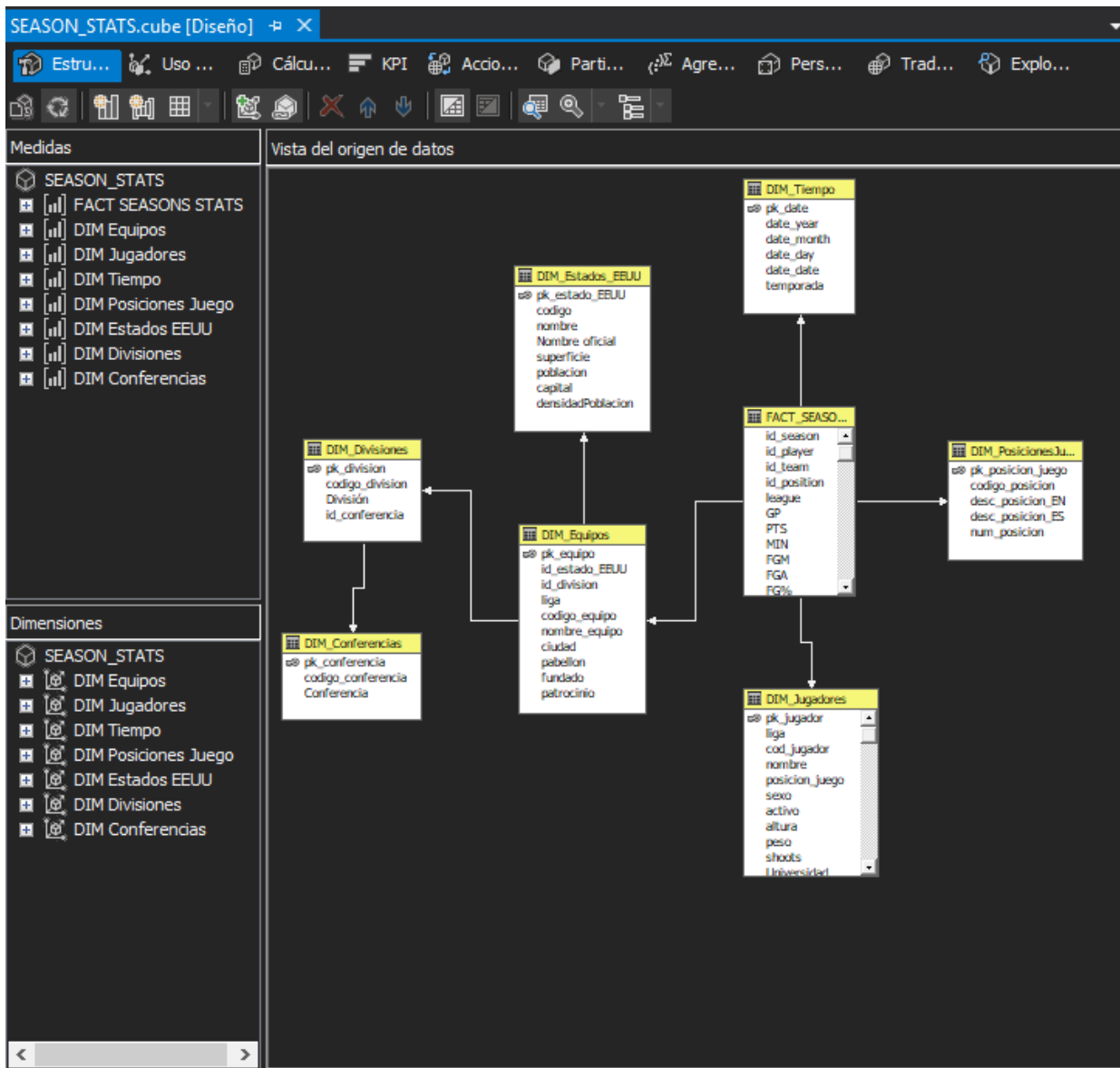
Igualmente para las dimensiones.



Luego de este paso el asistente muestra un resumen de la selección de medidas y dimensiones seleccionadas para la creación del cubo. Se pulsa en finalizar y el cubo estaría listo.



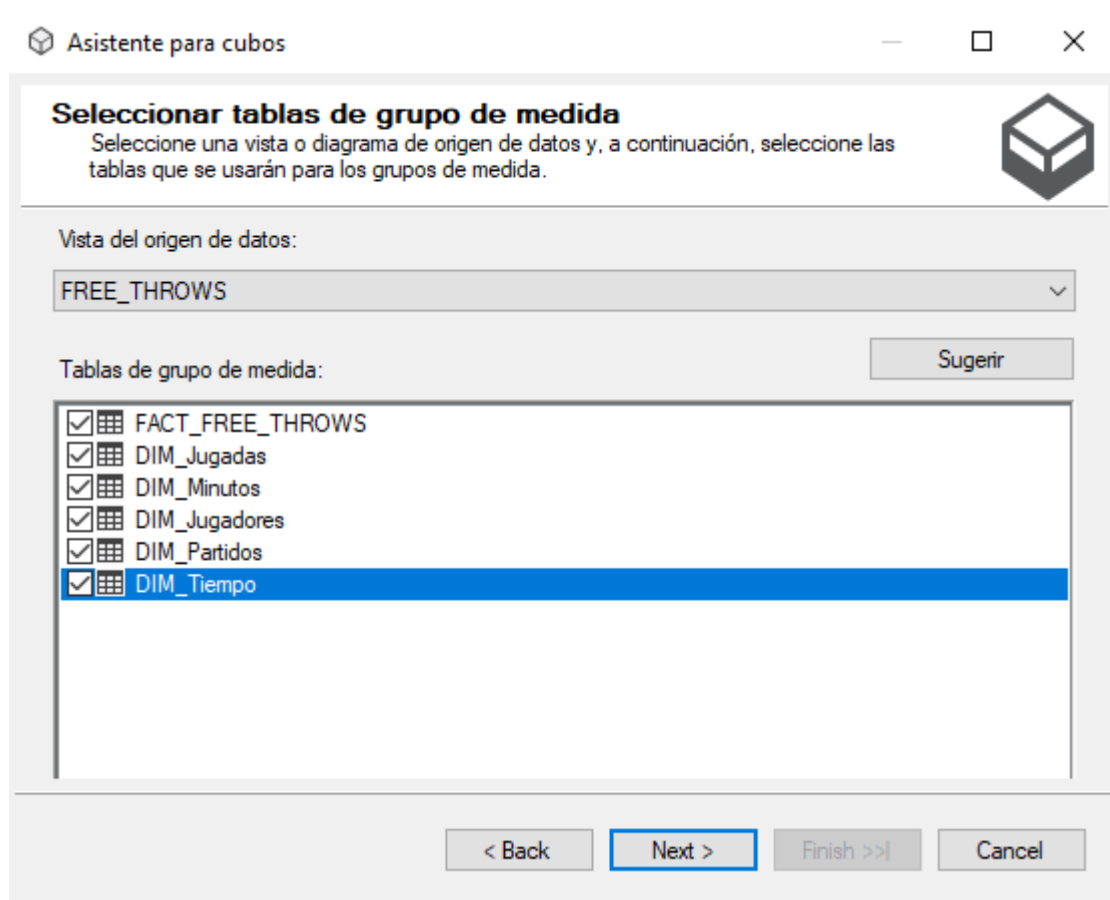
Para comprobar el diseño final del cubo se puede pulsar sobre su nombre en el explorador de soluciones y aparece en el área de trabajo un diagrama con su estructura, también aparecen a la izquierda del diagrama, sus medidas y dimensiones.



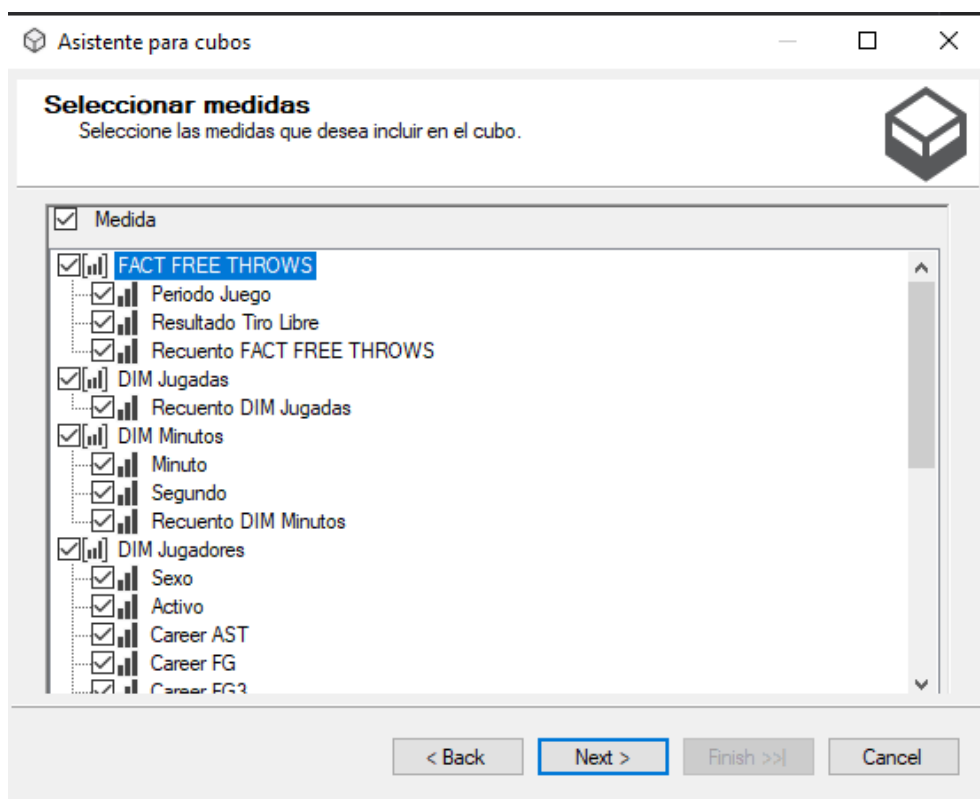
## Cubo FREE\_THROWS

Para la creación del cubo FREE\_THROWS se seguirá exactamente el mismo procedimiento que para crear el cubo anterior, a continuación se muestran las configuraciones de cada uno de los pasos.

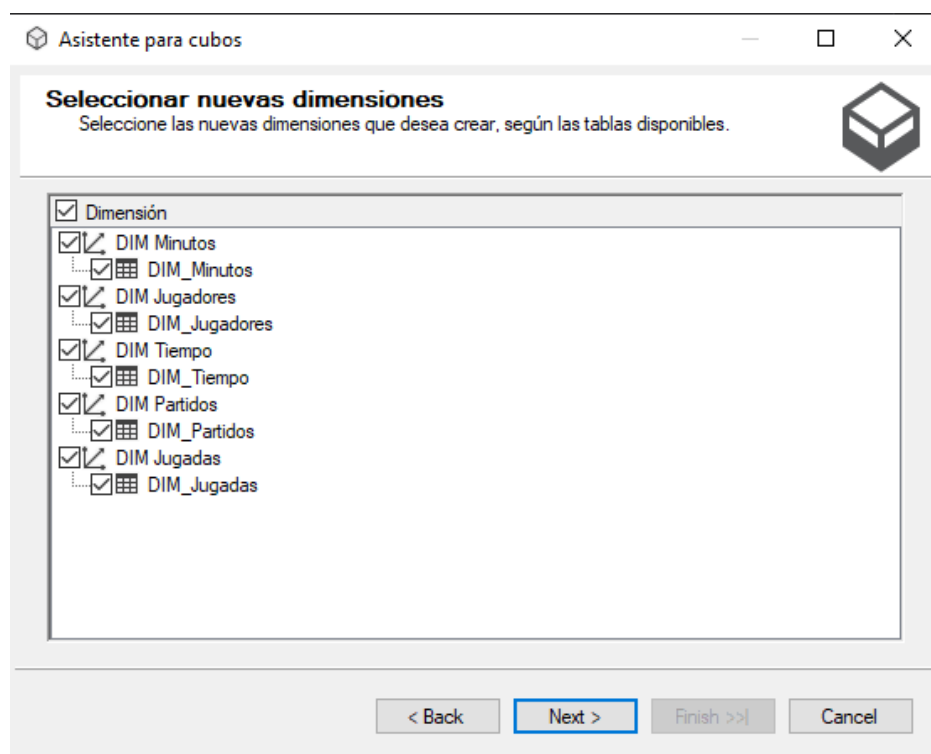
Luego de pulsar clic derecho y seleccionar “Nuevo cubo” se selecciona nuevamente la opción “Usar tablas existentes”. Entonces en la selección de las vistas es necesario seleccionar la vista FREE\_THROWS, donde se seleccionan también los grupos de medidas.



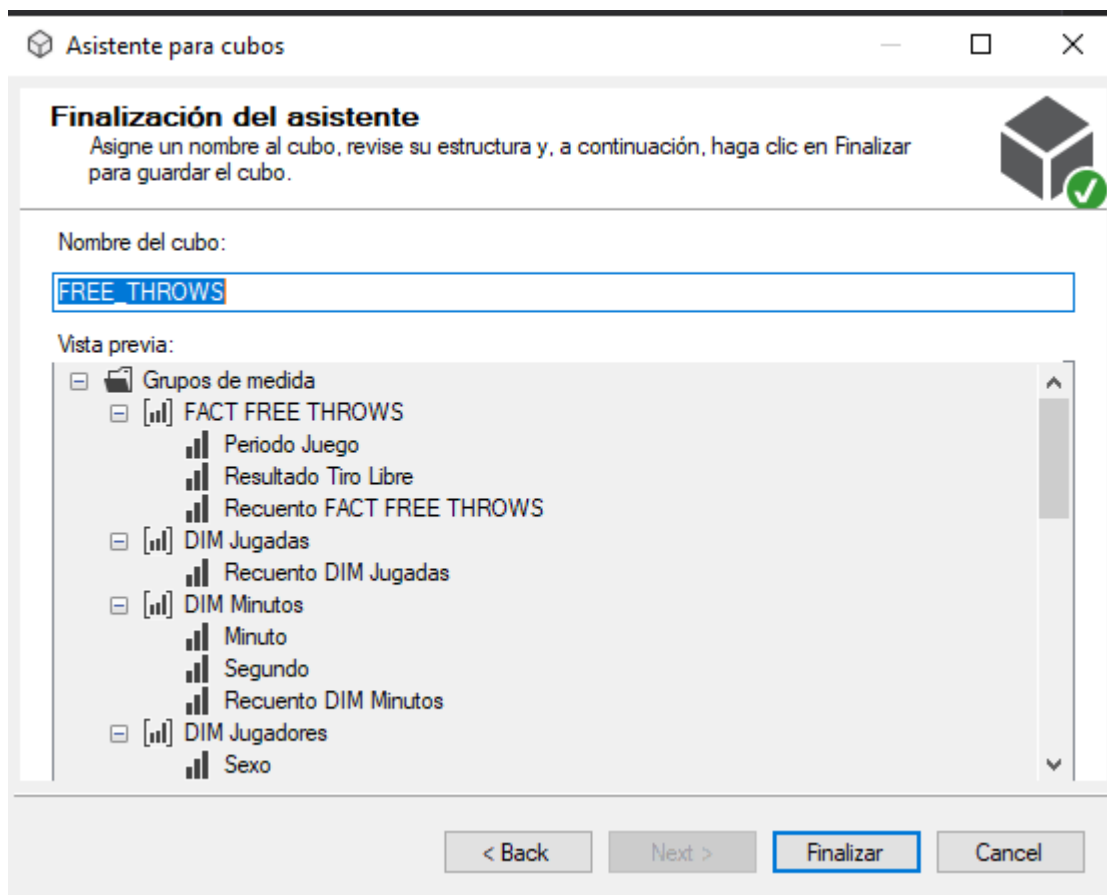
Posteriormente se seleccionan las medidas, al igual que en el cubo SEASON\_STATS se seleccionan todas las disponibles.



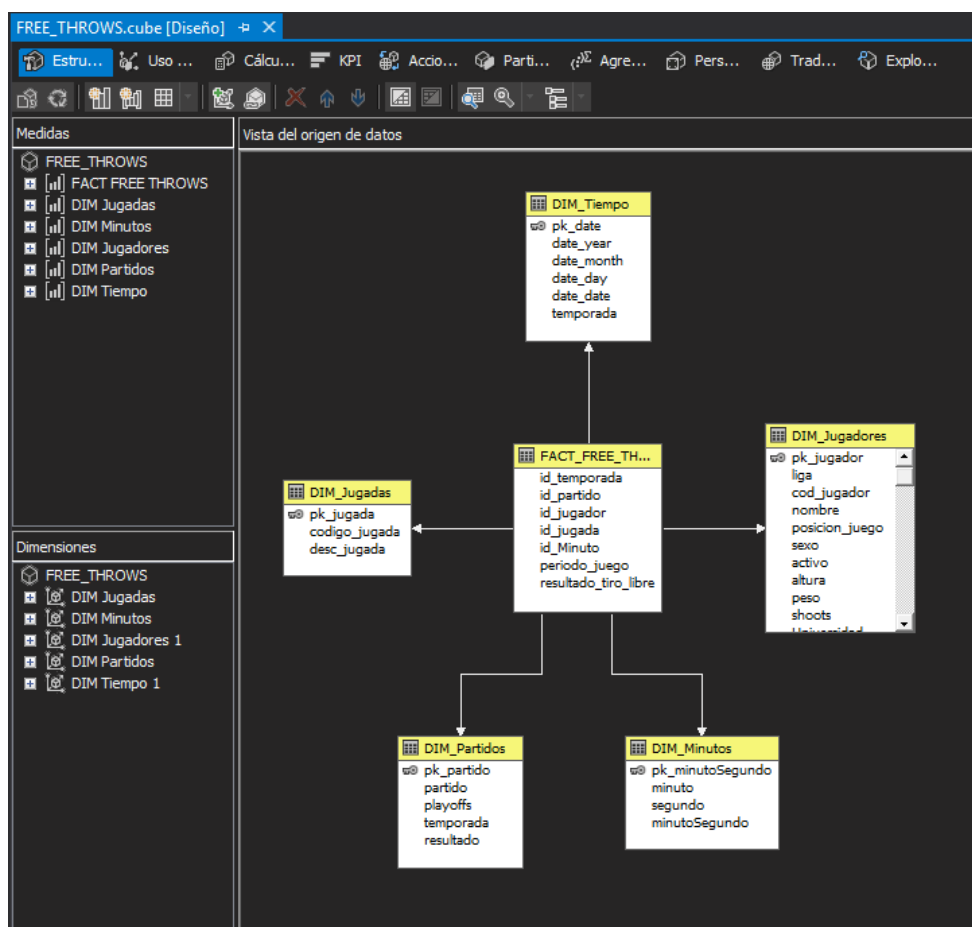
Se procede igual con las dimensiones.



El resumen del proceso de creación del cubo FREE\_THROWS se muestra al final del asistente. Se comprueba que todo esté correctamente configurado y se pulsa “Finalizar”.



Para comprobar su diseño se pulsa dos veces sobre su nombre en el explorador de soluciones y se obtiene el diseño del cubo FREE\_THROWS.

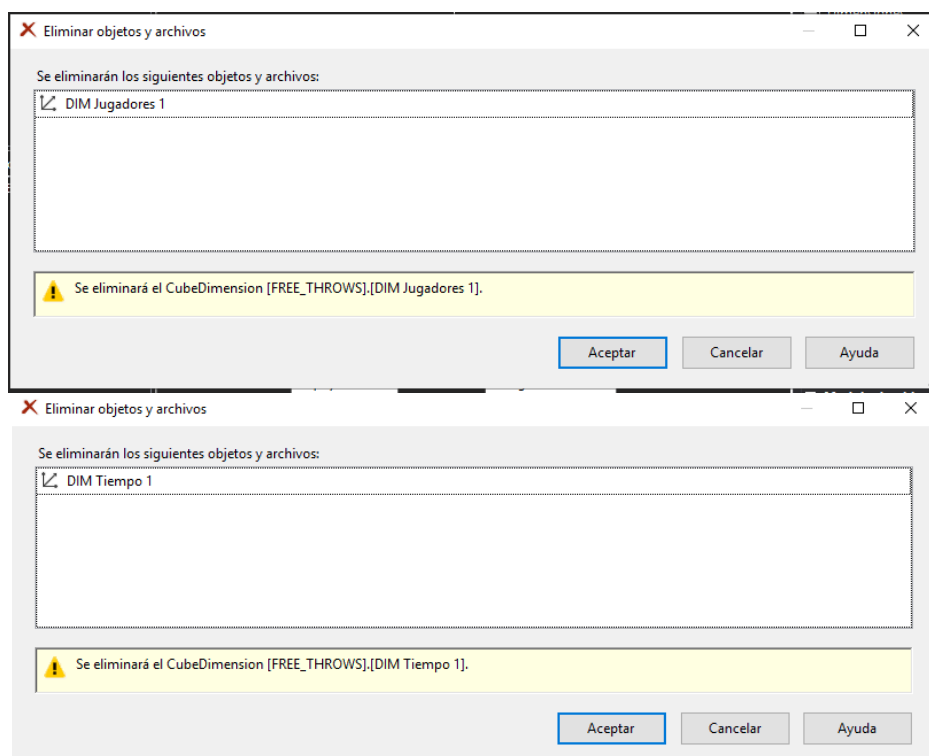


Luego de observar el resultado de la creación del cubo, se observa una anomalía, el sistema ha creado un duplicado a las dimensiones DIM\_Jugadores y DIM\_Tiempo, este comportamiento debe ser corregido porque queremos mantener la integridad de los datos y utilizar las mismas dimensiones en ambos cubos.

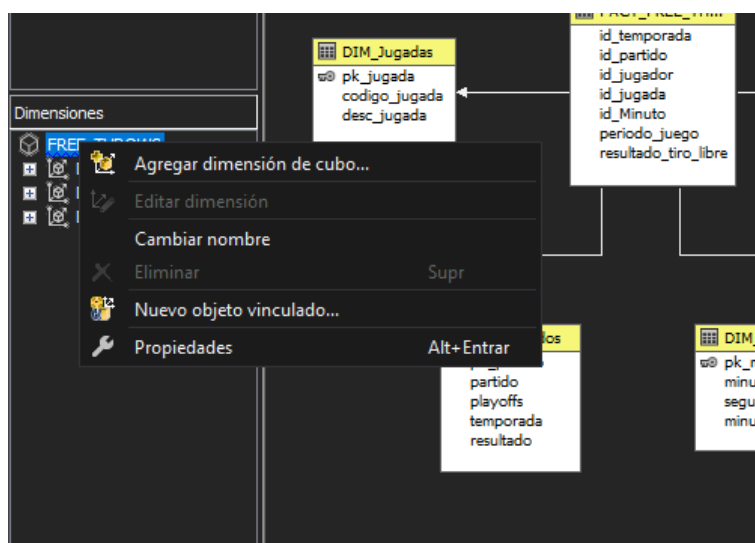


Primeramente es necesario eliminar estas dimensiones, esto puede realizarse desde el mismo "Explorador de soluciones", seleccionando los duplicados y pulsando la tecla Delete.

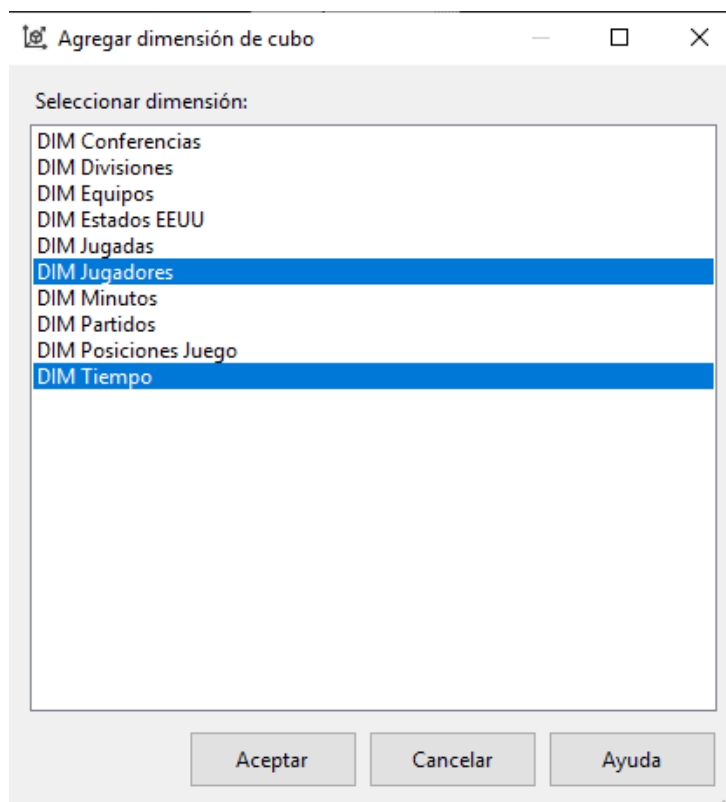




En este punto, el nuevo cubo **FREE\_THROWS** carece de estas dimensiones, por lo que es necesario agregarlas, se utilizan las mismas que ya están creadas del proceso de creación del cubo anterior. Para agregarlas es necesario desplazarse hasta la pestaña Estructura del diseñador del cubo y en el menú de “Dimensiones” pulsar clic derecho y seleccionar “Agregar dimensión de cubo”.



De la lista que aparece se seleccionan las dimensiones que se desean agregar y se pulsa aceptar.



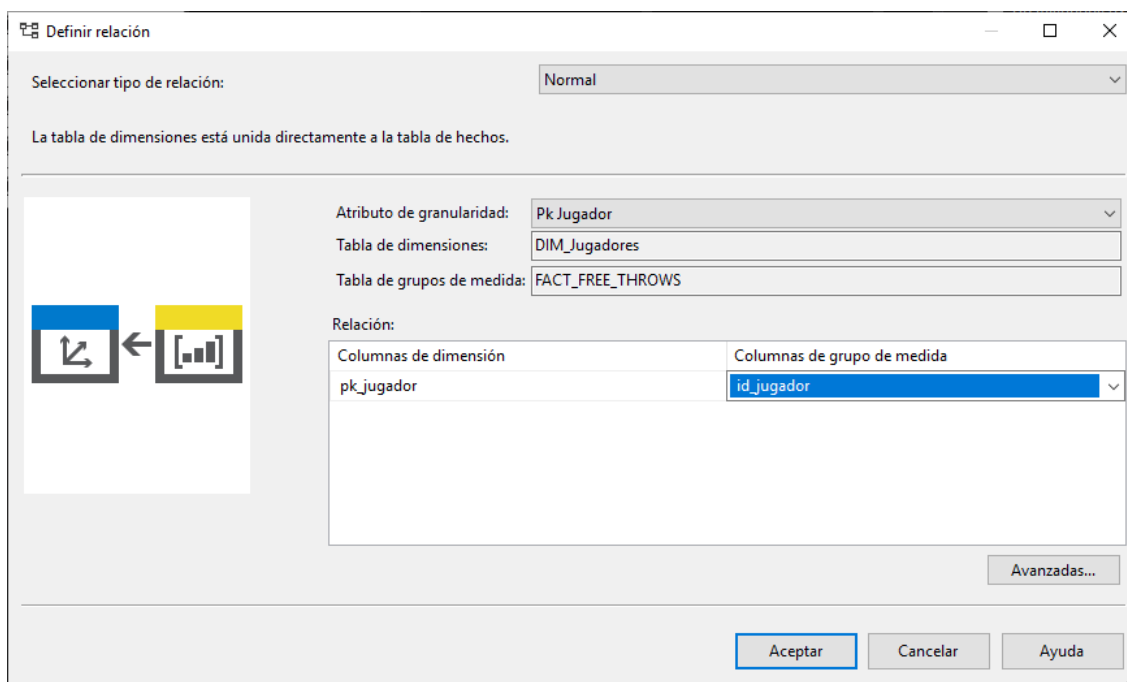
Para finalizar la creación del cubo FREE\_THROWS es necesario definir las relaciones entre las métricas y las dimensiones, esto porque se han eliminado y añadido las nuevas Dimensiones. Este paso se realiza en la pestaña “uso de dimensiones” del diseñador del cubo.



Se puede observar que para las nuevas Dimensiones no existe relación entre las dimensiones y las medidas, para realizar esta tarea hay que hacer clic en los puntos suspensivos al final de cuadro de texto del Grupo de medida para cada dimensión, de esta manera aparece el asistente que permite establecer la relación.

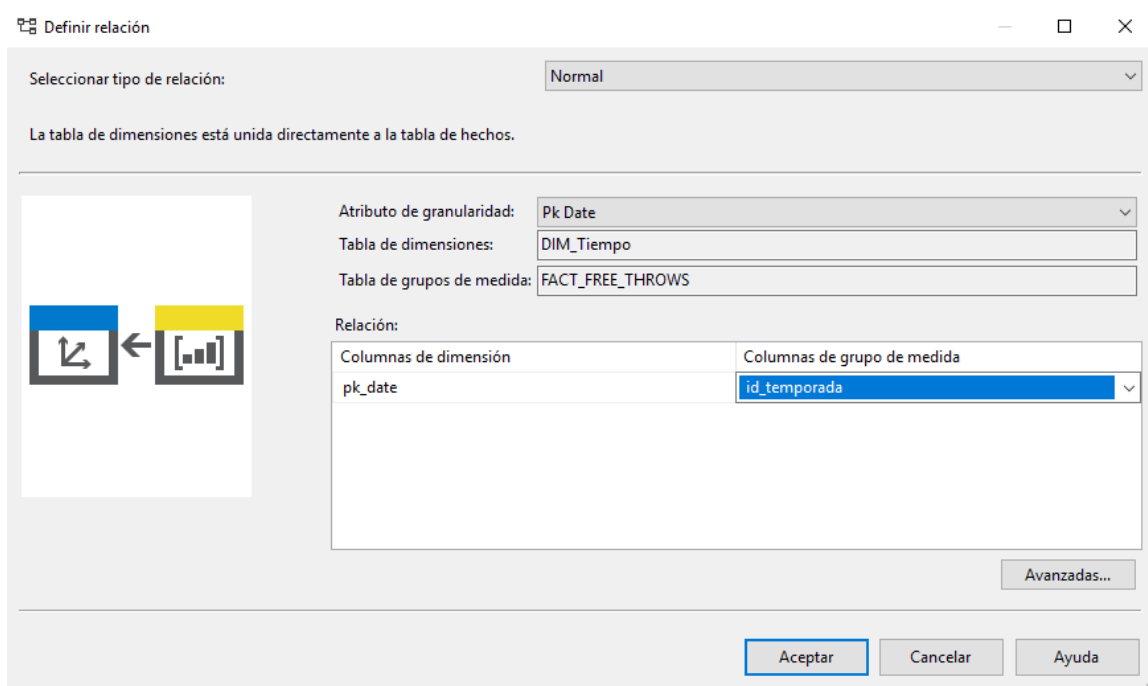
Se establecen los siguientes parámetros de configuración para la dimensión Jugador.

- Tipo de relación: Normal
- Atributo de granularidad: pk\_jugador
- Tabla de dimensiones: DIM\_Jugadores
- Tabla de grupo de medida: FACT\_FREE\_THROWS
- Columnas de dimensión: pk\_jugador
- Clave principal de la dimensión: id\_jugador

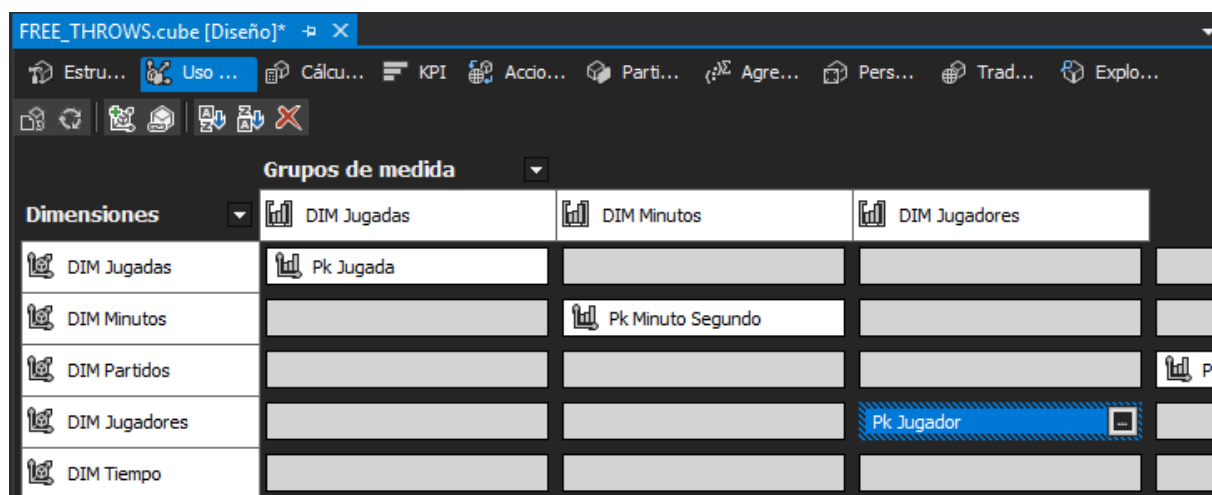


Se establecen los siguientes parámetros de configuración para la dimensión Jugador.

- Tipo de relación: Normal
- Atributo de granularidad: pk\_date
- Tabla de dimensiones: DIM\_Tiempo
- Tabla de grupo de medida: FACT\_FREE\_THROWS
- Columnas de dimensión: pk\_date
- Clave principal de la dimensión: id\_temporada



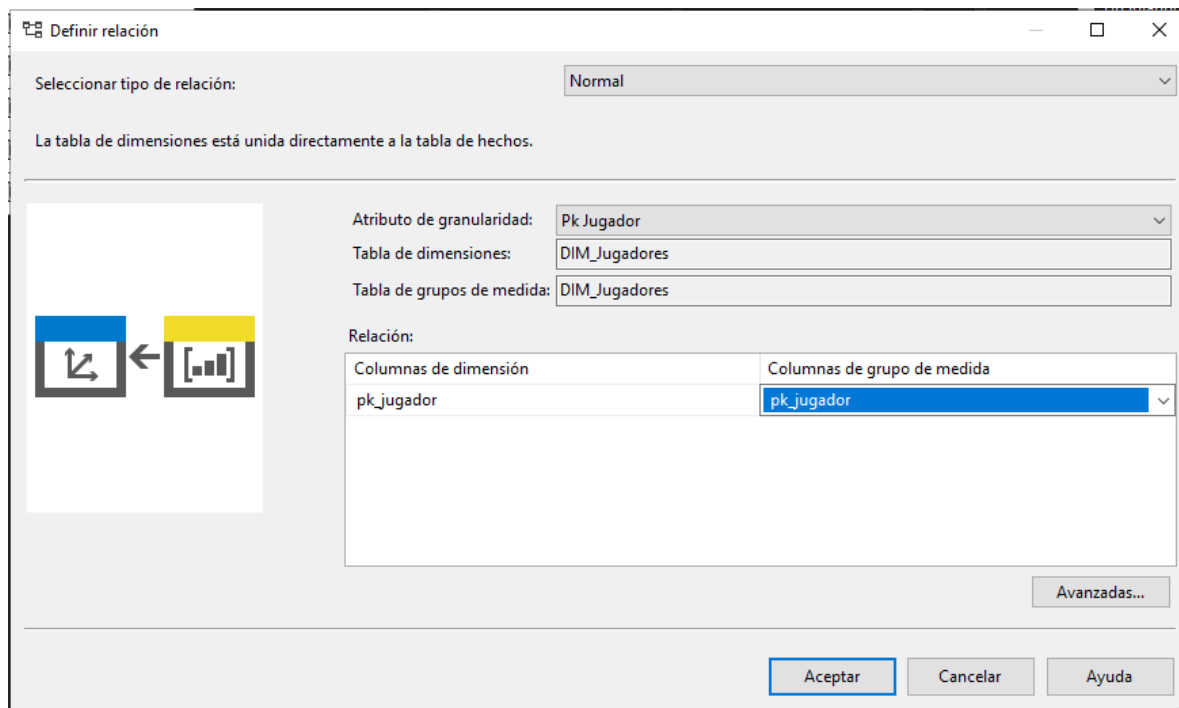
Hasta este punto, se han establecido las relaciones de las nuevas dimensiones con el grupo de medidas, es necesario realizar el mismo procedimiento con las Medidas a las que hacen referencia. En la misma pestaña “Uso de dimensiones” y siguiendo un procedimiento similar, se identifica el cruce entre la nueva dimensión y su Medida.



Pulsando sobre los puntos suspensivos se accede a un panel de configuración similar a los anteriores y los parámetros de configuración son los siguientes:

- Tipo de relación: Normal
- Atributo de granularidad: pk Jugador

- Tabla de dimensiones: DIM\_Jugadores
- Tabla de grupo de medida: DIM\_Jugadores
- Columnas de dimensión: pk\_jugador
- Clave principal de la dimensión: pk\_jugador



Definir relación

Seleccionar tipo de relación: Normal

La tabla de dimensiones está unida directamente a la tabla de hechos.

Atributo de granularidad: Pk Jugador

Tabla de dimensiones: DIM\_Jugadores

Tabla de grupos de medida: DIM\_Jugadores

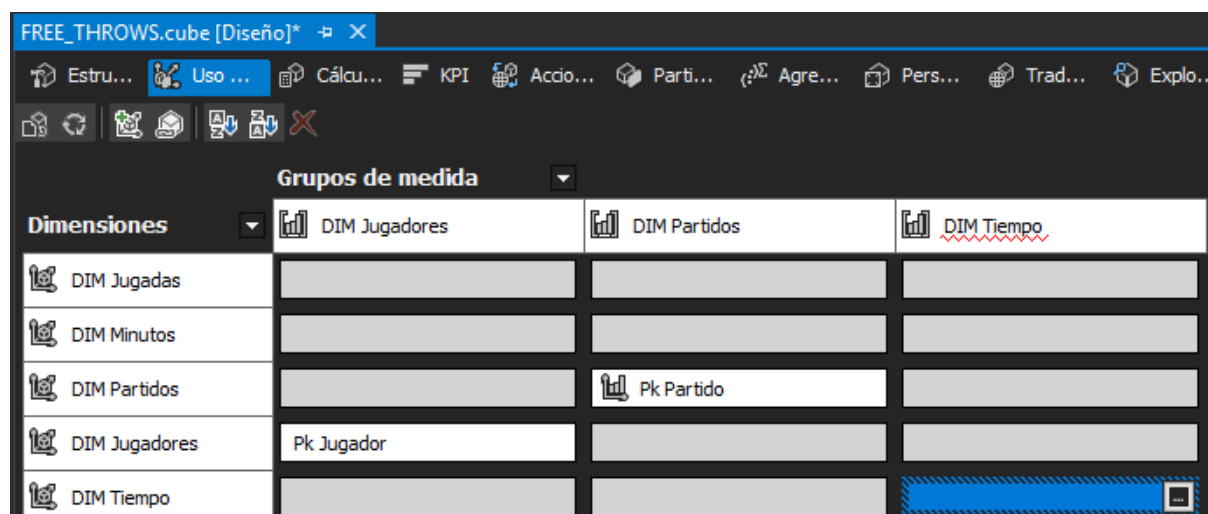
Relación:

Columnas de dimensión	Columnas de grupo de medida
pk_jugador	pk_jugador

Avanzadas...

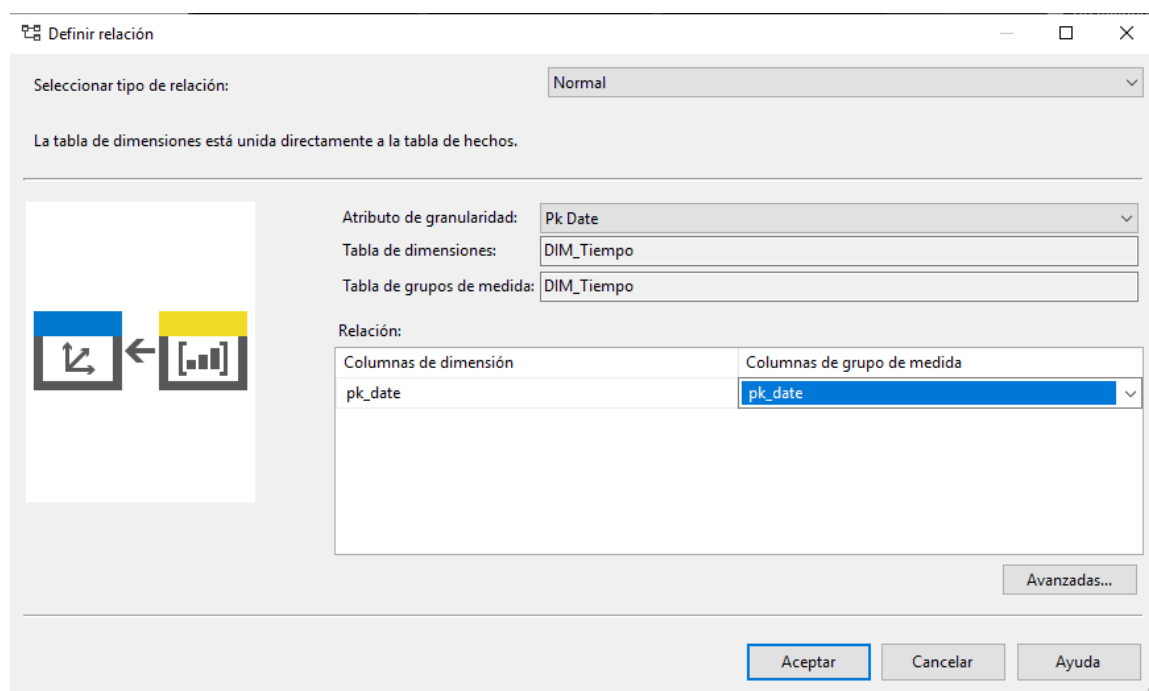
Aceptar Cancelar Ayuda

De manera análoga es necesario realizar el mismo proceso para la DIM\_Tiempo. Primeramente se identifica el cruce entre la dimensión y su tabla correspondiente.



Posteriormente se configura con los siguientes parámetros y se pulsa “Aceptar”.

- Tipo de relación: Normal
- Atributo de granularidad: pk Date
- Tabla de dimensiones: DIM\_Tiempo
- Tabla de grupo de medida: DIM\_Tiempo
- Columnas de dimensión: pk\_date
- Clave principal de la dimensión: pk\_date



De esta manera quedan definidas las relaciones entre el cubo FREE\_THROWS y sus nuevas medidas, quedando correctamente configurado.

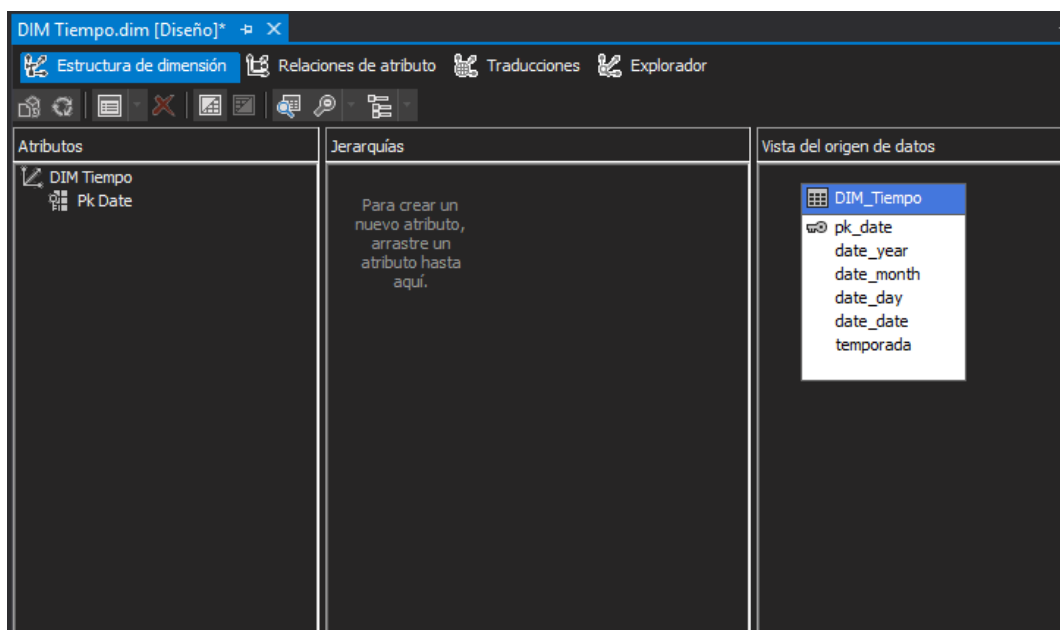
## Jerarquías y Dimensiones

Una vez definidas las relaciones de los cubos, es necesario trabajar con las dimensiones disponibles, con el objetivo de definir sus atributos y jerarquías. En este apartado se definirán estas jerarquías sobre los atributos de cada una de las dimensiones.

### DIM\_Tiempo

La manera de establecer estas relaciones es desde el editor de dimensiones, para acceder a el es suficiente con hacer doble clic sobre la dimensión que se desea editar.

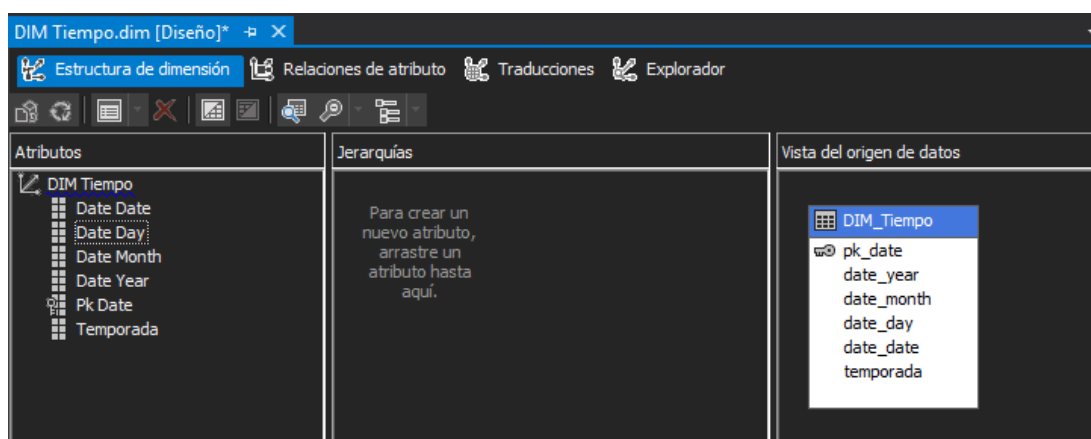
Inicialmente el editor luce de la siguiente manera.



Se observa que en la sección de atributos no están reflejados ninguno de los presentes en la pestaña de la “Vista del origen de datos”, por lo que hay que arrastrarlos desde la vista hasta la pestaña de Atributos. Para el ejemplo que nos ocupa y de manera análoga en todas las dimensiones se utilizarán todos los atributos de cada una de las dimensiones, previendo futuras necesidades.

En el caso de la jerarquía de esta dimensión aunque está bien definida por los atributos año, mes y días, no se implementa porque estos datos están rellenos con valores por defecto, con lo cual si fuese necesario su uso en un futuro sería necesario modificar esta dimensión. Igualmente, si se desease crear las jerarquías solamente será necesario arrastrar los elementos en su respectivo orden jerárquico a la pestaña “Jerarquías”.

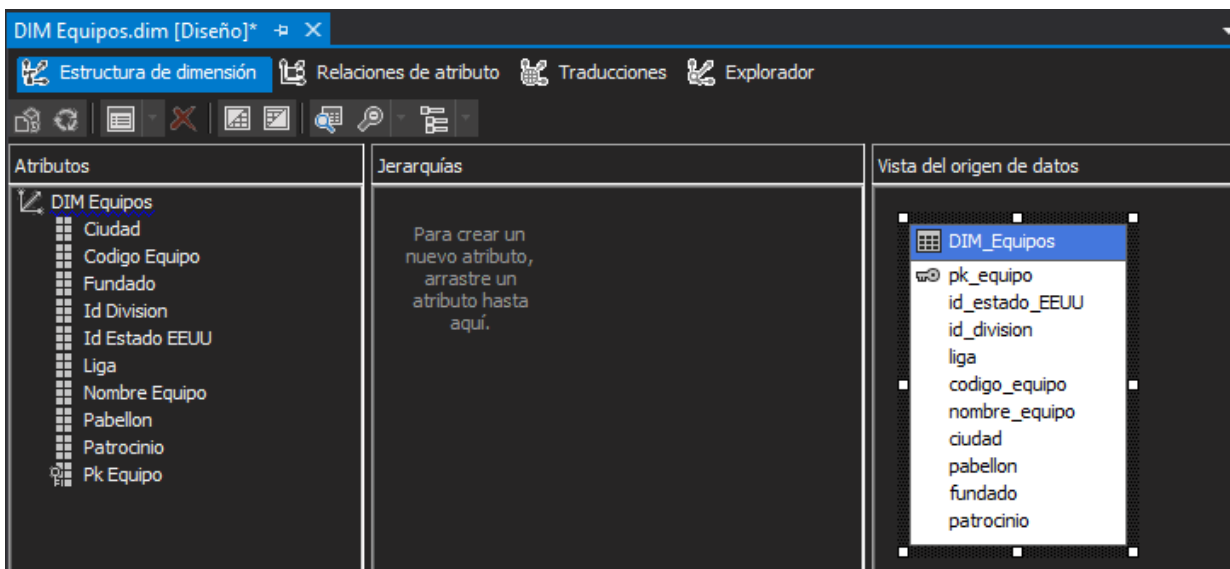
Luego de agregar las medidas a la dimensión, esa queda estructurada de la siguiente manera.



## DIM\_Equipo

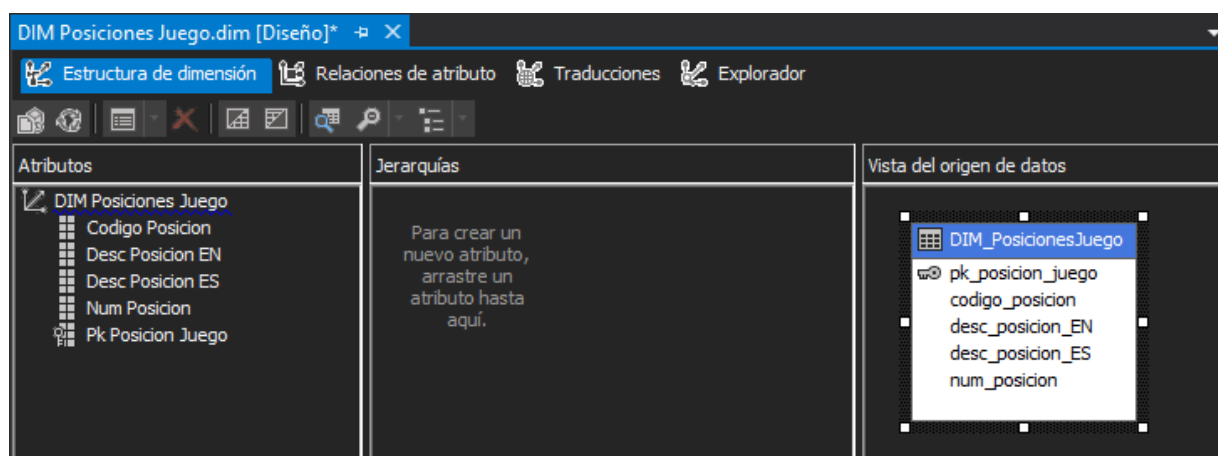
Para definir la dimensión DIM\_Tiempo se agregan todos los atributos presentes en la vista a la pestaña “Atributos”, en este caso no existe jerarquía. La dimensión queda de la siguiente manera.





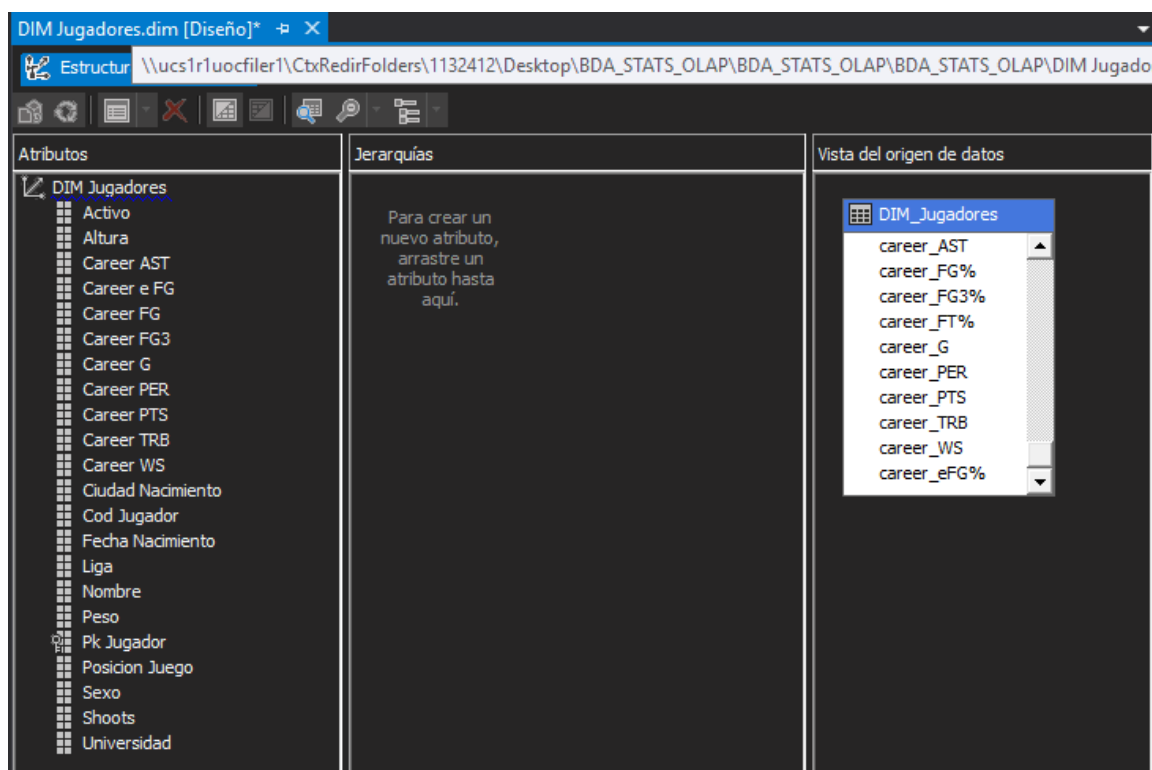
## DIM\_PosicionesJuego

Los atributos de esta dimensión también serán los mismos que los de la vista, y tampoco posee jerarquías.



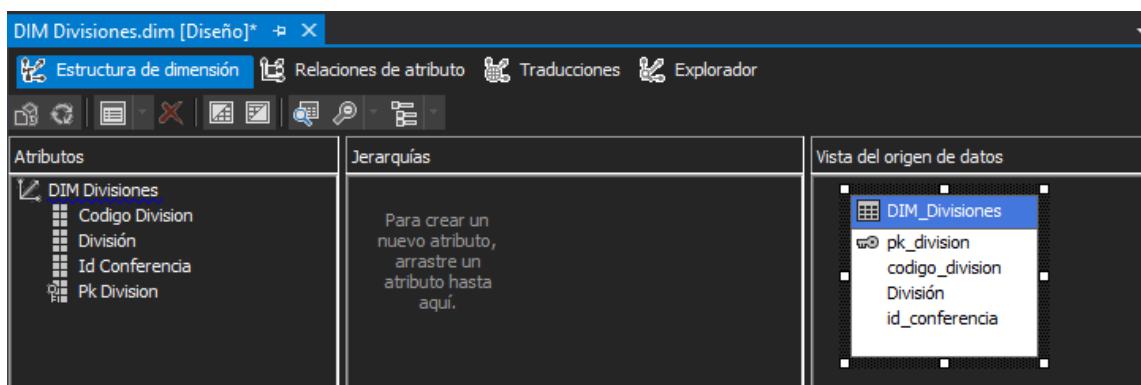
## DIM\_Jugadores

En esta dimensión se cargan todos los atributos, que serán de vital importancia a la hora de evaluar el desempeño de estos, y sobre todo para compararlos en el análisis del cubo.



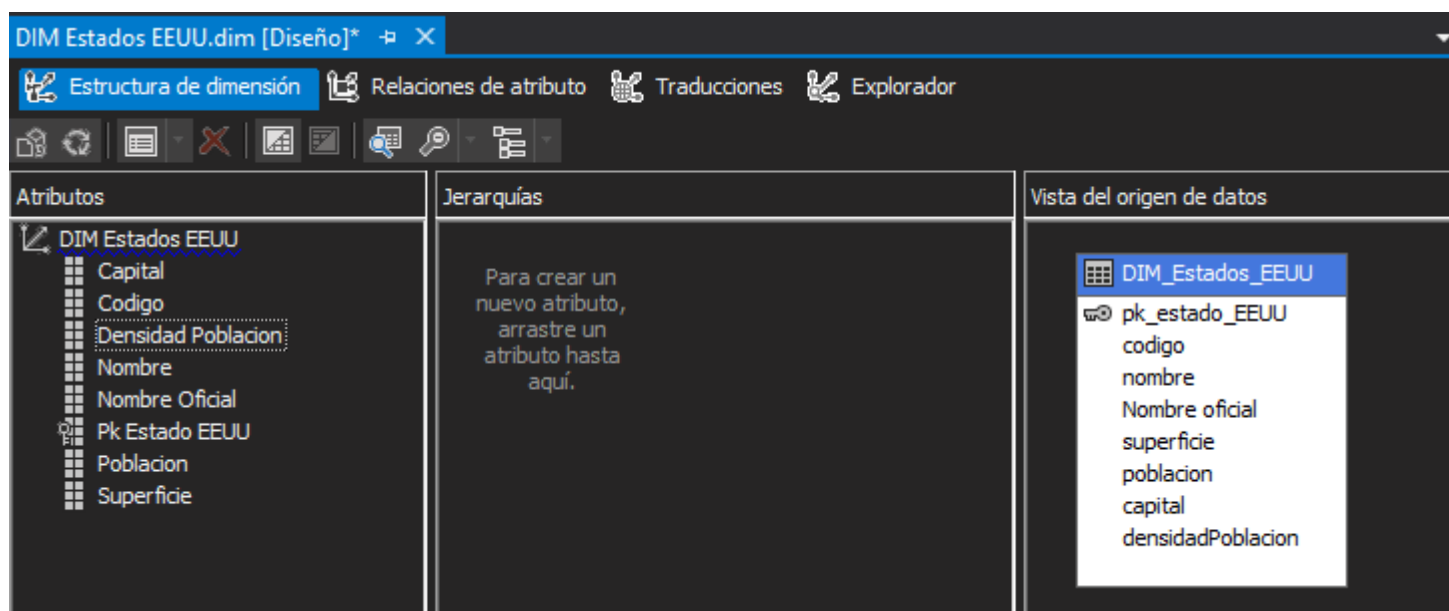
## DIM\_Divisiones

En el caso de esta Dimensión solo es necesario agregar el código y la división en si, porque el id\_conferencia ya lo toma por defecto. Tampoco presenta jerarquía.



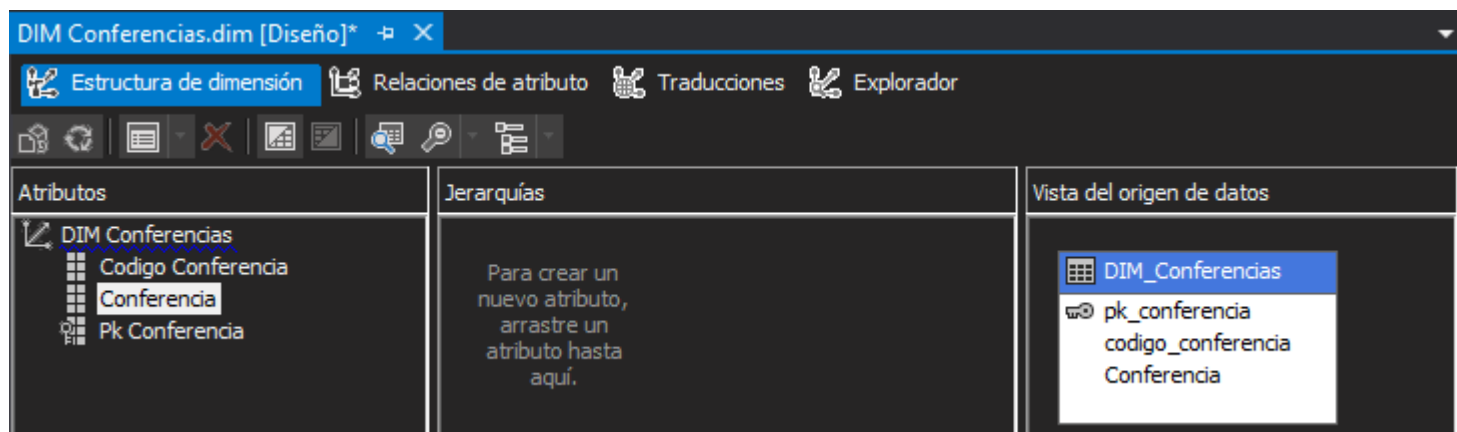
## DIM\_Estados\_EEUU

En esta dimensión se cargan los datos que describen cada uno de los estados, y estos datos no presentan jerarquías.



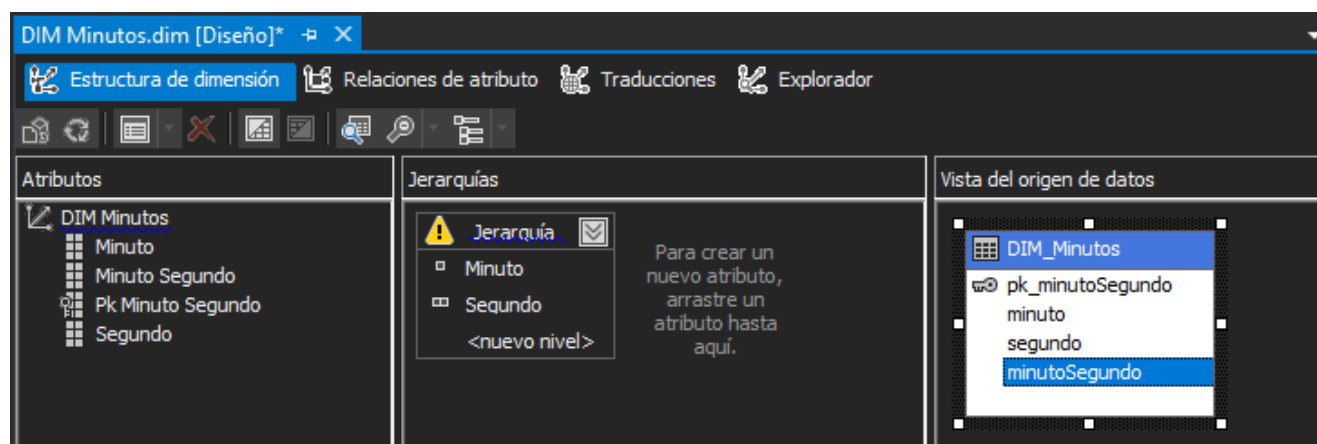
## DIM\_Conferencias

Esta es una de las dimensiones más pequeñas, solamente contiene el código y el nombre de la conferencia, estos datos serán cargados como atributos en el cubo.



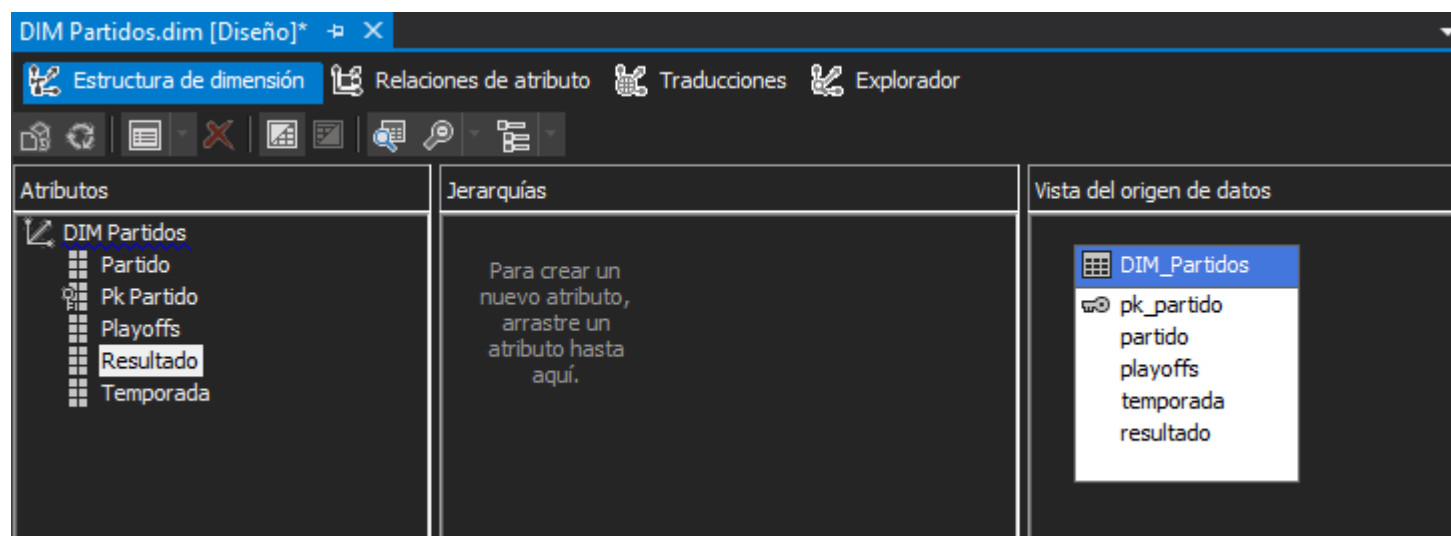
## DIM\_Minuto

Esta dimensión básicamente contiene los datos del tiempo del partido en el que fueron realizados los tiros libres. Presenta una jerarquía temporal entre los minutos y los segundos.



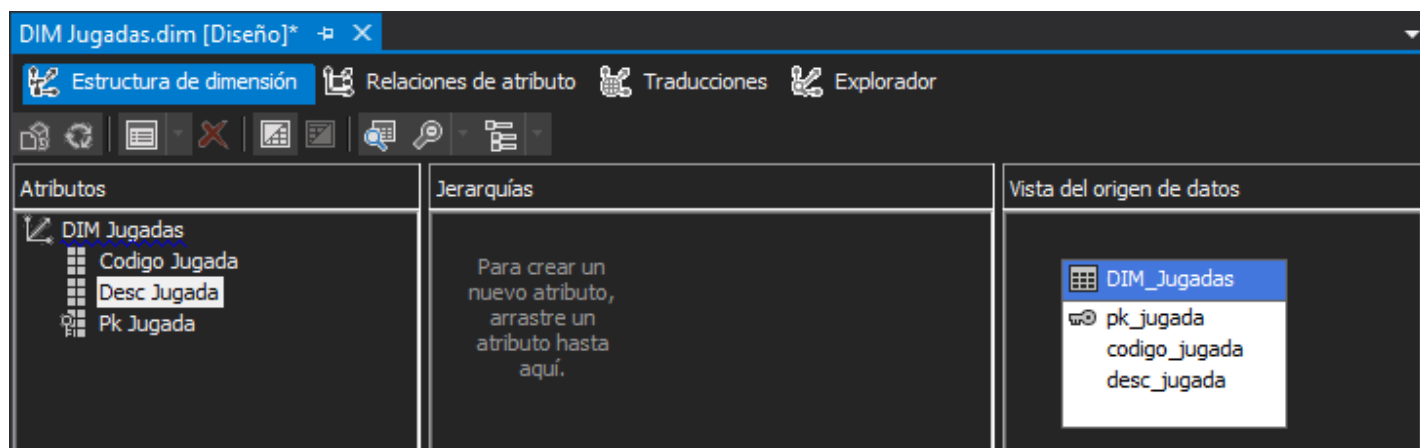
## DIM\_Partido

La dimensión igualmente contará con los mismos atributos que la vista del origen e datos.



## DIM\_Jugadas

La dimensión Jugadas cuenta con muy pocos datos, pero se creó para que en un futuro fuesen incrementándose los tipos de jugadas, de momento solo contiene los tiros libres. No presenta ninguna jerarquía entre sus atributos.



## Explotación de cubos

En este apartado se realizarán 6 explotaciones de los cubos. En cada una se irán recorriendo varias métricas y dimensiones de ambos cubos y su configuración se podrá ver en las capturas adjuntas, proponiendo en la parte superior el código MDX y en la inferior el resultado.

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

Esta consulta se generó completamente con los filtros del explorador, a continuación se muestran estos filtros y el resultado de la consulta.

Dimensión	Jerarquía	Operador	Expresión de filtro
DIM Tiempo	Temporada	Igual	{ 2015 - 2016 }
DIM Equipos	Liga	Igual	{ NBA }
<Seleccionar dimensi...			

Conferencia	División	Desc Posicion ES	PTS
CONFERENCIA ESTE	ATLÁNTICO	ALA PÍVOT	4617
CONFERENCIA ESTE	ATLÁNTICO	ALERO	2638
CONFERENCIA ESTE	ATLÁNTICO	BASE	3272
CONFERENCIA ESTE	ATLÁNTICO	ESCOLTA	5199
CONFERENCIA ESTE	ATLÁNTICO	PÍVOT	3852
CONFERENCIA ESTE	CENTRAL	ALA PÍVOT	8634
CONFERENCIA ESTE	CENTRAL	ALERO	6482
CONFERENCIA ESTE	CENTRAL	BASE	7641
CONFERENCIA ESTE	CENTRAL	DESCONOCIDO	5606
CONFERENCIA ESTE	CENTRAL	ESCOLTA	7284
CONFERENCIA ESTE	CENTRAL	PÍVOT	4677
CONFERENCIA ESTE	SURESTE	ALA PÍVOT	3769
CONFERENCIA ESTE	SURESTE	ALERO	4165
CONFERENCIA ESTE	SURESTE	BASE	4222
CONFERENCIA ESTE	SURESTE	DESCONOCIDO	2514
CONFERENCIA ESTE	SURESTE	ESCOLTA	5662
CONFERENCIA ESTE	SURESTE	PÍVOT	5064
CONFERENCIA OESTE	NOROESTE	ALA PÍVOT	4738
CONFERENCIA OESTE	NOROESTE	ALERO	5638
CONFERENCIA OESTE	NOROESTE	BASE	6505
CONFERENCIA OESTE	NOROESTE	DESCONOCIDO	2624
CONFERENCIA OESTE	NOROESTE	ESCOLTA	4800
CONFERENCIA OESTE	NOROESTE	PÍVOT	4439
CONFERENCIA OESTE	PACÍFICO	ALA PÍVOT	5832
CONFERENCIA OESTE	PACÍFICO	ALERO	3853
CONFERENCIA OESTE	PACÍFICO	BASE	4775
CONFERENCIA OESTE	PACÍFICO	ESCOLTA	5959
CONFERENCIA OESTE	PACÍFICO	PÍVOT	3654

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

Para resolver esta pregunta si que fue necesario recurrir al uso de MDX, sobre todo para seleccionar cuantos elementos mostrar en dependencia de la métrica EFF. Como se muestra en la siguiente figura se hace uso de la función TOPCOUNT() para obtener el resultado deseado.

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

Nombre	EFF
CANDACE PARKER	746
TINA CHARLES	712
TAMIKA CATCHINGS	710
MAYA MOORE	656
SOPHIA YOUNG-MALC...	634
CAPPIE PONDEXTER	594
NNEKA OGWUMIKE	589
KRISTI TOLIVER	575
KARA LAWSON	570
SYLVIA FOWLES	567

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

De manera similar a la aplicada en la pregunta 2, para generar esta consulta se hizo uso de la función TOPCOUNT() filtrando los valores del nombre del jugador por la métrica 3P, que representa el porcentaje de acierto de tiros de 3 puntos. Nótese que se obtienen resultados 1, lo que puede indicar que estos jugadores siempre que lanzan aciertan, pero este valor perfecto debe estar fundamentado en que realizó muy pocos intentos, pero en esta ocasión dejaremos estos valores tal y como se reciben de la consulta.

```
SELECT TOPCOUNT([DIM Jugadores].[Nombre].[Nombre].ALLMEMBERS, 10,
[Measures].[3P]) ON ROWS,
[Measures].[3P] ON COLUMNS
FROM [SEASON_STATS]
WHERE ( [DIM Tiempo].[Temporada].&[2011 - 2012], [DIM Jugadores].[Liga].&[NBA] )
```

Nombre	3P
JASON COLLINS	1
KURT THOMAS	1
SEAN MARKS	1
MAURICE AUSTIN	0,75
TONY BATTIE	0,67
STEVE NOVAK	0,56
AL HORFORD	0,5
BEN WALLACE	0,5
DERRICK BROWN	0,5
HILTON ARMSTRONG	0,5

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

En esta consulta solamente es necesario ordenar los valores de los rebotes por equipos en orden descendente, en esta ocasión no se tratan los valores “Desconocidos”, para reflejar que pueden aparecer en múltiples ocasiones en el top de las listas si estos tienen muchas apariciones, en posteriores apartados si que se eliminan de los resultados.



```
SELECT
ORDER( [DIM Equipos].[Nombre Equipo].[Nombre Equipo].ALLMEMBERS,
[Measures].[DREB], DESC ) ON ROWS,
[Measures].[DREB] ON COLUMNS

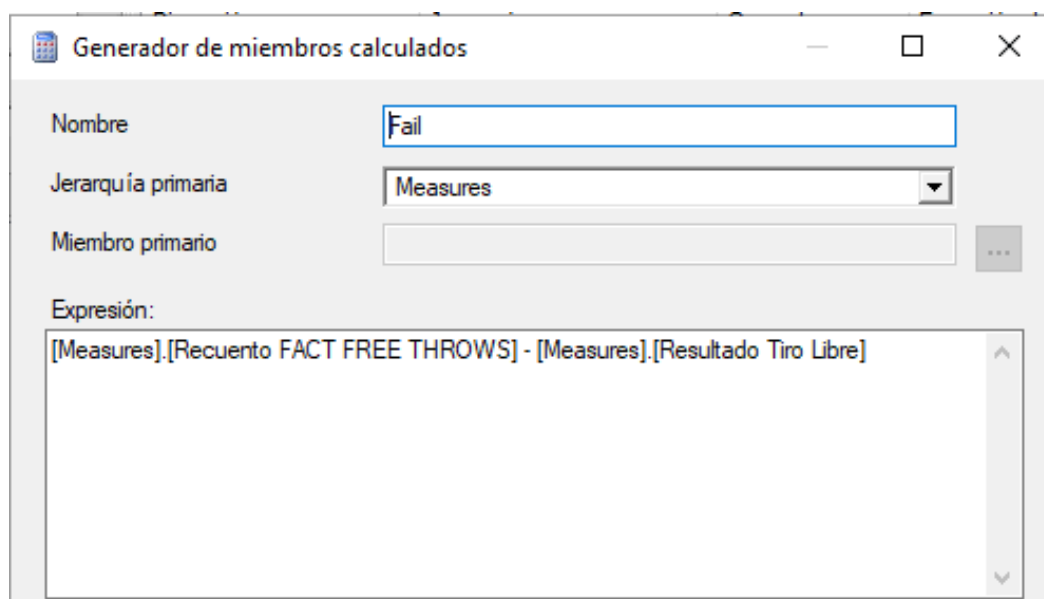
FROM [SEASON_STATS]

WHERE
( [DIM Tiempo].[Temporada].&[2014 - 2015],
[DIM Jugadores].[Liga].&[NBA] )
```

Nombre Equipo	DREB
DESCONOCIDO	21484
PORTLAND TRAIL BLAZERS	2745
OKLAHOMA CITY THUNDER	2715
HOUSTON ROCKETS	2694
PHOENIX MERCURY	2601
INDIANA PACERS	2598
LOS ANGELES CLIPPERS	2569
MINNESOTA TIMBERWOLVES	2546
DALLAS MAVERICKS	2514
DETROIT PISTONS	2482
ATLANTA HAWKS	2464
CHARLOTTE HORNETS	2464
DENVER NUGGETS	2442
LOS ANGELES LAKERS	2431
CHICAGO BULLS	2424
ORLANDO MAGIC	2405
MEMPHIS GRIZZLIES	2352
MIAMI HEAT	2342
BOSTON CELTICS	2254
WASHINGTON MYSTICS	2222
CLEVELAND CAVALIERS	2098

## 5. Partido de los Boston Celtics jugado como visitante en la temporada 2013-2014 donde se fallaron más tiros libres en el segundo intento.

Para resolver esta consulta es necesario crear un campo calculado, cuyo contenido se muestra en la siguiente figura. Este campo toma los valores totales del campo y le resta los aciertos, con lo que devuelve la cantidad de fallos en estos tiros libres.



Para configurarlo en orden descendente y filtrarlo para que solamente presentase los segundos tiros libres, se buscó la similitud de la frase “2 of 2”, que es la que indica que se hace referencia a los segundos tiros libres. De manera similar y usando la propiedad ‘Member\_Caption’ se compara la cadena “- BOS” para filtrar los partidos que juegan los Celtics como visitante.

El resto de la consulta se ha generado con el diseñador, a continuación se adjunta la consulta y el resultado de esta en la que muestra que el partido con mayor cantidad de fallos fue en el que se enfrentaron a los *Toronto Raptors* con 15 fallos.

```
WITH
  MEMBER [Measures].[Fail] AS [Measures].[Recuento FACT FREE THROWS] - [Measures].[
    Resultado Tiro Libre]

SELECT
  NON EMPTY { [Measures].[Resultado Tiro Libre], [Measures].[Fail] } ON COLUMNS,
  NON EMPTY { ORDER( ([DIM Partidos].[Partido].[Partido].ALLMEMBERS ), [Measures].[Fail],
    DESC ) } DIMENSION PROPERTIES MEMBER_CAPTION, MEMBER_UNIQUE_NAME ON ROWS

FROM ( SELECT ( Filter( [DIM Jugadas].[Desc Jugada].[Desc Jugada].ALLMEMBERS, Instr( [DIM
  Jugadas].[Desc Jugada].currentmember.Properties( 'Member_Caption' ), '2 of 2' ) > 0 ) ) ON
  COLUMNS FROM ( SELECT ( Filter( [DIM Partidos].[Partido].[Partido].ALLMEMBERS, Instr( [DIM
  Partidos].[Partido].currentmember.Properties( 'Member_Caption' ), '- BOS' ) > 0 ) ) ON
  COLUMNS FROM ( SELECT ( { [DIM Tiempo].[Temporada].[2013 - 2014] } ) ON COLUMNS
  FROM [FREE_THROWS]))))

WHERE ( [DIM Tiempo].[Temporada].[2013 - 2014] )
```

Partido	Resultado Tiro ...	Fail
TOR - BOS	31	15

## 6. 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 solucionar este análisis temporal el primer paso que hay que realizar es establecer las fechas en los filtros. Para ello se utiliza la DIM Tiempo, con jerarquía Temporada y los valores desde la temporada 2006-2007 hasta la temporada 2015-2016.

También se establece un filtro para evitar los datos de los jugadores que no tienen registrada una mano hábil. En la figura que se muestra a continuación se muestra como se excluyen los valores “DESCONOCIDO” y “NO DISPONIBLE”.

Dimensión	Jerarquía	Operador	Expresión de fil
DIM Tiempo	Temporada	Igual	{ 2015 - 2016,
DIM Jugadores	Shoots	No igual a	
<Seleccionar dimensi...			

Shoots	Temporada	FTA	FTM	Fail	Porcie
DESCONO...	2006 - 2007	12561	9312	3249	74,134
DESCONO...	2007 - 2008	11429	8703	2726	76,148
DESCONO...	2008 - 2009	12477	9355	3122	74,977
DESCONO...	2009 - 2010	11306	8675	2631	76,729
DESCONO...	2010 - 2011	9504	7242	2262	76,199
DESCONO...	2011 - 2012	8581	6536	2045	76,168
DESCONO...	2012 - 2013	7947	6019	1928	75,739
DESCONO...	2013 - 2014	8402	6508	1894	77,457
DESCONO...	2014 - 2015	8075	6223	1852	77,065


  

☐ All  
☒ DESCONOCIDO  
☐ Left  
☐ Left Right  
☒ NO DISPONIBLE  
☐ Right

Aceptar Cancelar

De los campos necesarios para resolver la query, la tabla de hechos ya cuenta con dos de ellos, el campo FTA que muestra los intentos de tiros libres, y el campo FTM que muestra los aciertos. Ahora para calcular los fallos y el porcentaje de aciertos es necesario crear dos campos calculados.

En el primero de estos campos "Fail", se restan los valores de los campo FTA – FTM y este valor sería el valor de la cantidad de fallos.


**Generador de miembros calculados**

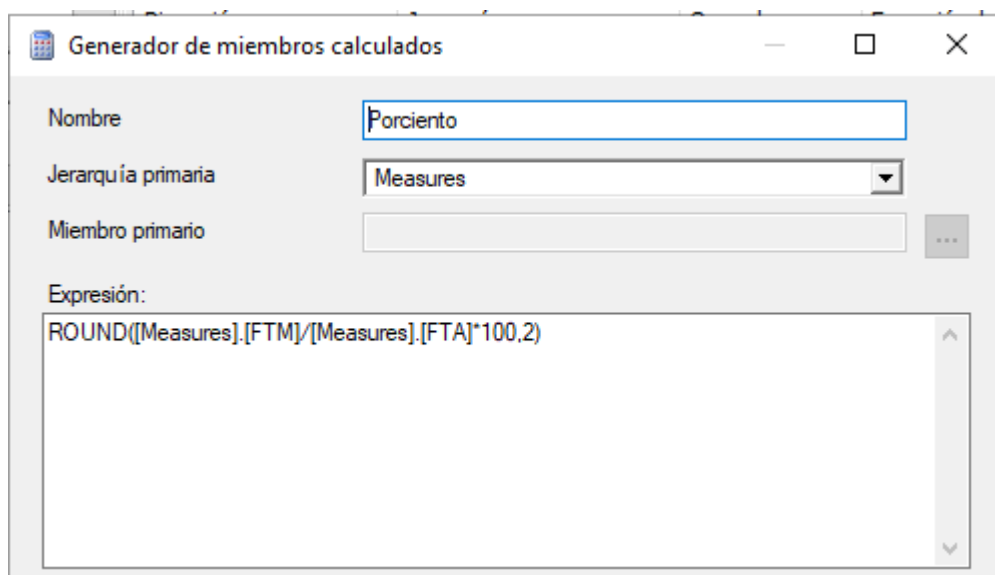
Nombre

Jerarquía primaria

Miembro primario

Expresión:

Para el segundo simplemente se aplica la fórmula  $\text{aciertos/intentos} * 100$  o lo que sería lo mismo  $\text{FTM7FTA} * 100$ .



Generador de miembros calculados

Nombre:

Jerarquía primaria:

Miembro primario:

Expresión:

```
ROUND([Measures].[FTM]/[Measures].[FTA]*100,2)
```

Una vez incluidos los campos y las métricas en el área de trabajo y al ejecutarse la query se obtienen los siguientes resultados, de los que se puede apreciar un cambio en el comportamiento del porcentaje de aciertos, al principio del rango los mejores eran los lanzadores zurdos y luego pasaron a ser los derechos, además que se aprecia que los lanzadores llamados “ambidiestros” no presentan muy buenos resultados comparados con los que tienen un brazo de lanzamiento dominante.

Temporada	Shoots	FTA	FTM	Fail	Porciento
2006 - 2007	Left	5628	4332	1296	76,97
2006 - 2007	Right	56242	41850	14392	74,41
2007 - 2008	Left	4833	3747	1086	77,53
2007 - 2008	Right	56766	42690	14076	75,2
2008 - 2009	Left	4916	3811	1105	77,52
2008 - 2009	Right	54269	40923	13346	75,41
2009 - 2010	Left	4520	3341	1179	73,92
2009 - 2010	Right	54194	41990	12204	77,48
2010 - 2011	Left	5130	3866	1264	75,36
2010 - 2011	Right	54136	41213	12923	76,13
2011 - 2012	Left	5717	4200	1517	73,47
2011 - 2012	Right	53329	40948	12381	76,78
2012 - 2013	Left	4420	3288	1132	74,39
2012 - 2013	Left Ri...	192	106	86	55,21
2012 - 2013	Right	39474	29821	9653	75,55
2013 - 2014	Left	5690	4154	1536	73,01
2013 - 2014	Left Ri...	291	177	114	60,82
2013 - 2014	Right	47839	36222	11617	75,72
2014 - 2015	Left	6401	4596	1805	71,8
2014 - 2015	Left Ri...	339	235	104	69,32
2014 - 2015	Right	50795	38729	12066	76,25
2015 - 2016	Left	5824	4145	1679	71,17
2015 - 2016	Left Ri...	248	159	89	64,11
2015 - 2016	Right	49621	37547	12074	75,67

