**ANEXO 3:**

**MANUAL DE OPERACIÓN**

**BUSINESS INTELLIGENCE PARA ANALIZAR**

**LA DEMANDA DE ENERGÍA ELÉCTRICA**

**POR:**

**HUGO GONZÁLEZ OLAYA**



**TRABAJO REALIZADO COMO REQUISITO PARCIAL PARA**

**OPTAR AL TÍTULO DE INGENIERO DE SISTEMAS**

**ASESOR:**

**ROBERTO CARLOS GUEVARA CALUME**

**CORPORACIÓN UNIVERSITARIA REMINGTON**

**FACULTAD DE INGENIERÍAS**

**MEDELLÍN - ANTIOQUIA**

**2020**

**DECLARACIÓN SOBRE DERECHOS DE AUTOR**

Las ideas expuestas en el presente trabajo de grado son de responsabilidad del autor. Los derechos de la información desarrollada durante el proceso de investigación pertenecen a la Corporación Universitaria Remington de Colombia y al creador de este trabajo. El material y código desarrollado en este trabajo es de uso libre. El autor no se hace responsable por su uso.

**AUTOR**

Hugo de Jesús González Olaya, Ingeniero Químico, MCSE: Microsoft certified solutions expert business intelligence, estudiante de Ingeniería de Sistemas en la universidad Uniremington.

[hugo14.gonzalez@gmail.com](mailto:hugo14.gonzalez@gmail.com)

[ing\_hgonzalez@hotmail.com](mailto:ing_hgonzalez@hotmail.com)

**CÓDIGO**

El código puede ser descargado del siguiente link. Solo fue subido una muestra de datos, puesto que el volumen de datos de demanda y pérdida de energía eléctrica de la empresa XM es alrededor de 2,000 millones de filas en 20 años. La muestra son unos 1,373 archivos con un tamaño de 300 MB, esta muestra contiene una fracción de los datos.

<https://github.com/hugo14gonzalez/BI_Demanda>

**CONTENIDO**

[1 INTRODUCCIÓN MANUAL DE OPERACIÓN 1](#_Toc43741373)

[2 ARTEFACTOS DEL SISTEMA 2](#_Toc43741374)

[2.1 LISTA DE DOCUMENTOS 2](#_Toc43741375)

[2.2 INSTALACIÓN DEL SISTEMA 4](#_Toc43741376)

[2.3 AMBIENTES 4](#_Toc43741377)

[3 NAVEGAR POR EL SITIO WEB DEL SISTEMA 5](#_Toc43741378)

[4 CONSULTAR REPORTES DESDE SQL REPORTING SERVICES 6](#_Toc43741379)

[4.1 SEGURIDAD EN REPORTES 6](#_Toc43741380)

[4.2 CONSULTAR REPORTES DEL SISTEMA 6](#_Toc43741381)

[4.3 CONSULTAR REPORTES DE AUDITORÍA 7](#_Toc43741382)

[4.4 EXPORTAR REPORTE 7](#_Toc43741383)

[4.5 IMPRIMIR REPORTE 7](#_Toc43741384)

[5 CARGA MANUAL DE DATOS 8](#_Toc43741385)

[5.1 CARGAR DÍAS FESTIVOS 8](#_Toc43741386)

[5.2 CARGAR DIMENSIÓN FECHA 8](#_Toc43741387)

[5.3 DIMENSIONES QUE NO TIENEN ETL 8](#_Toc43741388)

[6 CARGA DE LA BODEGA DE DATOS 9](#_Toc43741389)

[6.1 CONFIGURACIÓN DE ETLS 9](#_Toc43741390)

[6.2 FECHA DE EJECUCIÓN Y RANGO DE FECHAS 9](#_Toc43741391)

[6.3 JOBS SQL SERVER 9](#_Toc43741392)

[6.3.1 Agenda de los jobs 9](#_Toc43741393)

[6.3.2 Notificaciones cuando falla un job 10](#_Toc43741394)

[6.3.3 Credencial y cuenta proxy 10](#_Toc43741395)

[6.3.4 Permisos en objetos de base de datos 11](#_Toc43741396)

[6.4 EJECUCIÓN DE ETLs PARA CARGA DE DATOS 11](#_Toc43741397)

[6.4.1 Ejecución de Job SQL 11](#_Toc43741398)

[6.4.2 Revisión de Jobs 11](#_Toc43741399)

[7 RECARGA DE DATOS 12](#_Toc43741400)

[7.1 TABLAS DE CONFIGURACIÓN 12](#_Toc43741401)

[7.2 RECARGA DE DATOS 13](#_Toc43741402)

[7.2.1 Recarga de dimensiones 13](#_Toc43741403)

[7.2.2 Recarga de métricas 14](#_Toc43741404)

[7.2.3 Procesar dimensiones de cubos 17](#_Toc43741405)

[7.2.4 Procesar particiones del cubo 17](#_Toc43741406)

[7.2.5 Procesar todos los cubos 18](#_Toc43741407)

[7.3 ASPECTO A MEJORAR PARA LA RECARGA DE DATOS 18](#_Toc43741408)

[8 AUDITORÍA O BITÁCORA DE EJECUCIÓN DE PROCESOS 19](#_Toc43741409)

[8.1 TABLAS DE BITÁCORA 19](#_Toc43741410)

[8.2 ARCHIVOS DE LOG 22](#_Toc43741411)

[8.3 REVISIÓN DE JOBS 23](#_Toc43741412)

[8.3.1 Historial de Jobs 23](#_Toc43741413)

[8.3.2 Notificación por Correo electrónico 23](#_Toc43741414)

[8.4 REPORTES DE AUDITORÍA 23](#_Toc43741415)

[8.5 REPORTES PROPIOS DE SQL DE EJECUCIÓN DE ETL 24](#_Toc43741416)

[8.6 LOG SQL SERVER 25](#_Toc43741417)

[9 ADMINISTRACIÓN DE CUBOS 26](#_Toc43741418)

[9.1 SEGURIDAD 26](#_Toc43741419)

[9.2 CAMBIOS DISEÑO DE DIMENSIONES Y PROCESAMIENTO 26](#_Toc43741420)

[10 CONFIGURACIÓN DE MODELOS DEL SISTEMA 27](#_Toc43741421)

[10.1 MODELOS 27](#_Toc43741422)

[10.2 METADATOS 27](#_Toc43741423)

[10.2.1 Diccionario de datos 27](#_Toc43741424)

[10.2.2 Consulta de los objetos de base de dato data warehouse 27](#_Toc43741425)

[10.3 TABLAS DE UTILIDADES 28](#_Toc43741426)

[10.3.1 [Utility].[ChangeDateForDim] 29](#_Toc43741427)

[10.3.2 [Utility].[CubePartition] 30](#_Toc43741428)

[10.3.3 [Utility].[CubePartitionDetail] 31](#_Toc43741429)

[10.3.4 [Utility].[CubeSystem] 31](#_Toc43741430)

[10.3.5 [Utility].[MetricSystem] 32](#_Toc43741431)

[10.3.6 [Utility].[ModelLoad] 32](#_Toc43741432)

[10.3.7 [Utility].[ModelParameter] 32](#_Toc43741433)

[10.3.8 [Utility].[ModelSchedule] 33](#_Toc43741434)

[10.3.9 [Utility].[ModelSystem] 33](#_Toc43741435)

[10.3.10 [Utility].[ModelVersion] 33](#_Toc43741436)

[10.3.11 [Utility].[ModuleModel] 34](#_Toc43741437)

[10.3.12 [Utility].[ModuleProgramming] 34](#_Toc43741438)

[10.3.13 [Utility].[ModuleSystem] 36](#_Toc43741439)

[10.3.14 [Utility].[ParameterKeyValue] 36](#_Toc43741440)

[10.3.15 [Utility].[QuerySQL] 36](#_Toc43741441)

[10.3.16 [Utility].[ScheduleSystem] 36](#_Toc43741442)

[10.3.17 [Utility].[SpecialDate] 37](#_Toc43741443)

[10.3.18 [Utility].[Subscription] 37](#_Toc43741444)

[10.3.19 [Utility].[UserAgent] 37](#_Toc43741445)

[10.3.20 [Utility].[UserDomain] 37](#_Toc43741446)

[11 FUENTES DE DATOS 38](#_Toc43741447)

[11.1 CONFIGURACIÓN ARCHIVOS PLANOS 38](#_Toc43741448)

[11.2 FORMATOS DE ARCHIVOS 38](#_Toc43741449)

[11.2.1 Archivo dimensión agente 38](#_Toc43741450)

[11.2.2 Archivo dimensión compañía 39](#_Toc43741451)

[11.2.3 Archivo dimensión geografía 39](#_Toc43741452)

[11.2.4 Archivo de movimiento demanda y pérdidas de energía 40](#_Toc43741453)

[12 QUÉ HACER EN CASO DE FALLOS 41](#_Toc43741454)

[12.1 FALLOS AL EJECUTAR EL JOB FALLA 41](#_Toc43741455)

[12.2 FALLO POR CONEXIÓN DE RED 41](#_Toc43741456)

[12.3 CLAVES DUPLICADAS 41](#_Toc43741457)

[12.4 FALLA AL PROCESAR LAS DIMENSIONES 41](#_Toc43741458)

[12.5 FALLO AL PROCESAR EL CUBO 42](#_Toc43741459)

**INDICE DE FIGURAS**

[Figura 1. Sitio web del sistema 5](#_Toc43741296)

[Figura 2. Administración de carpetas de reportes 6](#_Toc43741297)

[Figura 3. Exportar e imprimir reporte 7](#_Toc43741298)

[Figura 4. Job SQL para ejecutar tareas 9](#_Toc43741299)

[Figura 5. Operador SQL para recibir notificaciones 10](#_Toc43741300)

[Figura 6. Credencial SQL y cuenta proxy SQL 10](#_Toc43741301)

[Figura 7. Consulta de métricas 15](#_Toc43741302)

[Figura 8. Consulta de modelo 16](#_Toc43741303)

[Figura 9. Log en archivo planos de ejecución de ETLs 23](#_Toc43741304)

[Figura 10. Reporte de bítacora del sistema 24](#_Toc43741305)

[Figura 11. Reporte propio de SQL para consultar ejecución de ETLs 25](#_Toc43741306)

[Figura 12. Parámetro procesamiento de dimensiones en forma completa o actualización 26](#_Toc43741307)

[Figura 13. Tablas de utilidades 29](#_Toc43741308)

**INDICE DE TABLAS**

[Tabla 1. Artefactos del sistema 2](#_Toc43741460)

[Tabla 2. Ambientes 4](#_Toc43741461)

[Tabla 3. Tablas de bitácora 19](#_Toc43741462)

[Tabla 4. Estados de ejecución de procesos 20](#_Toc43741463)

[Tabla 5. Tipos de errores de ejecución de procesos 20](#_Toc43741464)

[Tabla 6. Formato archivo para la dimensión agente 38](#_Toc43741465)

[Tabla 7. Formato archivo para la dimensión compañía 39](#_Toc43741466)

[Tabla 8. Formato archivo para la dimensión geografía 39](#_Toc43741467)

[Tabla 9. Formato archivo para la tabla de hechos demanda y pérdidas 40](#_Toc43741468)

# INTRODUCCIÓN MANUAL DE OPERACIÓN

El objetivo de este documento es proporcionar al operador del sistema elementos para la operación del sistema Demanda BI. El sistema tiene dos aspectos:

La parte operativa para que los datos estén disponibles, en esta parte no hay intervención por parte de los usuarios finales que consumen la información sino el personal administrativo en cargado de validar que los procesos de carga de datos y servidores estén en operación en conjunto con los analistas del sistema encargados de validar que las cargas hayan sido realizadas en forma satisfactoria y resolver las inconsistencias y calidad de datos.

La otra parte está formada por los usuarios finales los cuales utilizan la aplicación web para consultar la información para la toma de decisiones o consultas los reportes o generan sus propias consultas desde herramientas como Excel.

La parte de usuario final es más simple porque la aplicación es pequeña y fácil de comprender. La parte operativa es a la cual será dedicado más espacio.

# ARTEFACTOS DEL SISTEMA

## LISTA DE DOCUMENTOS

La Tabla 1 presenta las rutas de los artefactos construidos en el proyecto

Tabla 1. Artefactos del sistema

|  |  |
| --- | --- |
| **Archivo** | **Descripción** |
| **\DemandaBI\Doc\Proyecto\** | |
| Presentacion\_DemandaBI.pptx | Presentación proyecto de grado demanda BI |
| ProyectoGrado\_DemandaBI.docx | Documento proyecto de grado |
| Anexo1\_Solicitud\_Datos\_XM.docx | Anexo 1 solicitud de datos XM |
| Anexo2\_Manual\_Instalacion.docx | Anexo 2 manual de instalación |
| Anexo3\_Manual\_Operacion.docx | Anexo 3 manual de operación |
| Anexo4\_Herramientas\_Software.docx | Anexo 4 herramientas de software utilizadas en el proyecto |
| **\DemandaBI\Doc\Formatos\** | |
| DatosAutoriza.pdf | Solicitud de datos para el proyecto de grado a la empresa XM |
| DatosAutoriza\_Firma.pdf | Solicitud de datos para el proyecto de grado a la empresa XM |
| Solicitud\_Datos\_XM.docx | Solicitud de datos para el proyecto de grado a la empresa XM |
| **\DemandaBI\Doc\Manual\** | |
| DicDatos\_DemandaDW.xlsm | Diccionario de datos del sistema |
| DicDatos\_DemandaXM.xlsm | Diccionario de datos con las tablas del BI de XM utilizadas para generar archivos planos para el sistema |
| DisenoOLAP\_DemandaBI.xlsm | Diseño de la base de datos multidimensional |
| MatrizCRUD\_DemandaDW.xlsm | Matriz CRUD con permisos para el sistema |
| **\DemandaBI\Resources\Images\** | |
| \*.\* | Archivos de imágenes utilizadas en el sistema |
| **\DemandaBI\Setup\** | |
| Cubos | Instalador de base de datos multidimensional SQL Server Analysis Services |
| DemandaWeb | Instalador de sitio Web del sistema |
| ETL | Instaladores de ETLs |
| Report | Instaladores de reportes SQL Reporting Services |
| **\DemandaBI\Src\** | |
| Auditoria\_RS | Proyecto de reportes de auditoria creados en SSDT para VS 2017 |
| Demanda\_OLAP | Proyecto de cubos creados en SSDT para VS 2017 |
| DemandaBI\_Datos\_SSIS | Proyecto de ETLs para extraer los datos de XM creados en SSDT para VS 2017 |
| DemandaBI\_RS | Proyecto de reportes del sistema creados en SSDT para VS 2017 |
| DemandaBI\_SSIS | Proyecto de ETLs del sistema creados en SSDT para VS 2017 |
| DemandaWeb | Proyecto sitio web del sistema creado con VS 2019 |
| Script | Proyecto de base de datos creado con SSMS |
| **\DemandaBI\Test\** | |
| \*.\* | Esta carpeta contiene casos de pruebas y una muestra con archivos para cargar el sistema. Solo fue subido una muestra de datos, puesto que el volumen de datos de demanda y pérdida de energía eléctrica de la empresa XM es alrededor de 2,000 millones de filas en 20 años. La muestra son unos 1,373 archivos con un tamaño total de 300 MB, esta muestra contiene una fracción de los datos. La muestra contiene datos para los años 2016 a 2019, solos los meses 2019-08 y 2019-09 contiene todos los datos, pero los demás archivos es una parte de los datos. |

## INSTALACIÓN DEL SISTEMA

Para instalar el sistema siga los pasos descritos en el manual de instalación, en la lista de documentos indica donde puede encontrar este documento.

## AMBIENTES

La Tabla 2 muestra los equipos donde están instalados las diferentes partes del sistema.

Tabla 2. Ambientes

|  |  |  |
| --- | --- | --- |
| **Ambiente** | **Servidor** | **Base datos o Carpetas** |
| Motor SQL Server (DW), Producción | MiServidor | DemandaDW |
| SQL Server Integration Server, Producción | MiServidor | Demanda |
| SQL Analisys Services, Producción | MiServidor | Demanda\_OLAP |
| SQL Reporting Services, Producción | MiServidor | DemandaBI |
| URL Sitio Web | MiServidor | DemandaWeb |

# NAVEGAR POR EL SITIO WEB DEL SISTEMA

La Figura 1 muestra el sitio web de la aplicación es solo de consulta, en el sitio puede consultar los mismos reportes desplegados en servidor de reportes, desde el sitio también puede exportar e imprimir los reportes. Para abrir utilizar la aplicación haga lo siguiente:

1. Abra un browser y escriba la dirección:

<http://localhost/DemandaWeb>



Figura 1. Sitio web del sistema

Fuente: Autor

1. El menú de inicio es un dashboar con indicadores del sistema, estos indicadores no tienen parámetros y los datos presentados son para última fecha con datos, por tanto, la información es actualizado en forma periódica cuando actualice los datos del sistema.
2. Los reportes están agrupados en los menús Demanda y Bitácora. En estos menús puede navegar por los diferentes menús de la aplicación.

# CONSULTAR REPORTES DESDE SQL REPORTING SERVICES

El sistema tiene dos grupos de reportes los propios del sistema y de auditoria de procesos, los reportes fueron creados en SQL Reporting Services. Estos reportes pueden ser consultados desde el servidor de reportes o en la aplicación web.

## SEGURIDAD EN REPORTES

Para consultar los reportes debe tener permisos en el servidor de reportes, para ello debe asignar permisos en las carpetas de reportes, para asignar permisos haga lo siguiente:

1. Abra un browser e ingrese al sitio de reportes
2. En la carpeta DemandaBI, haga clic en el botón tres puntos y luego clic en **Administrar**. La Figura 2 muestra el menú desplegable desde el cual puede administrar la carpeta.

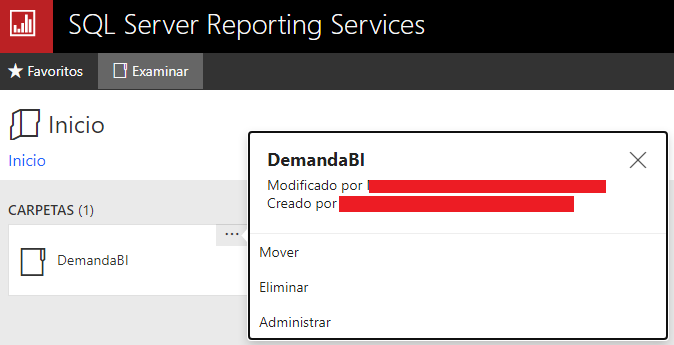


Figura 2. Administración de carpetas de reportes

Fuente: Autor

1. A mano izquierda haga clic en **Seguridad**
2. Configure permisos para los usuarios que puedan consultar los reportes.

## CONSULTAR REPORTES DEL SISTEMA

Para consultar los reportes haga lo siguiente:

1. En el sitio de reportes abra la carpeta DemandaBI
2. Haga clic en uno de los 4 reportes del sistema:

* DashBoardBI
* DemandaComercialAgente
* DemandaDepartamento
* DemandaRealComercialPerdida

1. Establezca los parámetros y presione el botón “**Ver informe**”

## CONSULTAR REPORTES DE AUDITORÍA

Para consultar los reportes haga lo siguiente:

1. En el sitio de reportes abra la carpeta DemandaBI\Auditoria
2. Haga clic en uno de los 5 reportes de auditoría
3. Establezca los parámetros y presione el botón “**Ver informe**”

## EXPORTAR REPORTE

La Figura 3 muestra el botón **Exportar** para exportar el reporte a formatos como: Word, Excel, PowerPoint, PDF, etc. Este botón lo encuentra en la barra de navegación del reporte.

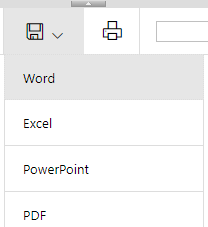


Figura 3. Exportar e imprimir reporte

Fuente: Autor

## IMPRIMIR REPORTE

En la Figura 3, puede ver el botón **Imprimir** para imprimir el reporte. Este lo botón lo encuentra en la barra de navegación del reporte. Otra alternativa consiste en exportar el reporte e imprimir desde la herramienta con la cual abre el archivo.

# CARGA MANUAL DE DATOS

Algunas tablas no tienen una ETL para carga, y requieren ser alimentadas en forma periódica, por ejemplo, la taba de días festivos.

## CARGAR DÍAS FESTIVOS

Antes de iniciar cada año, debe adicionar datos a la tabla de días festivos mediante un script para insertar datos, el siguiente es un ejemplo:

IF NOT EXISTS(SELECT \* FROM [Utility].[SpecialDate] WHERE [DateId] = '20190101')

BEGIN

INSERT INTO [Utility].[SpecialDate] ([DateId], [Type]) VALUES('20190101', 'Festivo');

END

## CARGAR DIMENSIÓN FECHA

En forma automática los paquetes llenan los datos de la tabla: [DW].[DimFecha], para adicionar datos en forma manual o cuando requiera actualizar datos cuando cambiar la tabla de días festivos, ejecute el siguiente script. Debe establecer las fechas de inicio y fin:

Declare @dateStart date, @dateEnd date;

Set @dateStart = convert(date, '2019-01-01', 120);

set @dateEnd = convert(date, '2020-01-01', 120);

EXECUTE [DW].[spLoadDimDate] @dateStart, @dateEnd, 1;

## DIMENSIONES QUE NO TIENEN ETL

Algunas tablas son pobladas al inicio del sistema, su número de filas es pequeño y el cambio es casi ninguno, el sistema cuenta con procedimientos almacenados (SP) denominados: spInferirDimNombreDimension para adicionar filas, pero es probable que requiera realizar modificaciones a los datos, puesto que estos SP, utilizan datos comodines. En estos casos es preferible insertas datos en forma manual a estas dimensiones. Las siguientes son las tablas que no tienen una ETL de carga:

* DimMercado

Las dimensiones de fecha son cargadas por SP, en las ETL:

* DimFecha: Poblada con el SP: [DW].[spLoadDimDate]
* DimTiempo: Poblada al inicio, no requiere cambios.
* DimPeriodo: Poblada al inicio, no requiere cambios.

# CARGA DE LA BODEGA DE DATOS

## CONFIGURACIÓN DE ETLS

Los parámetros de ejecución son almacenados en las ETLs, los cuales son establecidos en la instalación, para cambiar los parámetros consulte el manual de instalación.

## FECHA DE EJECUCIÓN Y RANGO DE FECHAS

La fecha de proceso es el día en que debe ser ejecutado una ETL, mientras que el rango de fechas define el conjunto de datos a ser procesados en el momento de ejecución. Como regla general las dimensiones (Dim) son cargadas todos los días, pero las tablas de hechos o movimiento (Fact) deben tener una agenda(s) de ejecución y rango de datos a ser cargados. Consulte la explicación anterior de las tablas de utilidades de agendas: [Utility].[ScheduleSystem] y la relación entre agendas y modelos: [Utility].[ModelSchedule].

## JOBS SQL SERVER

### Agenda de los jobs

La Figura 4 muestra el job de carga diaria, debe ser habilitados y modificada la agenda de ejecución según las necesidades del negocio. En ambiente de pruebas y desarrollo debe estar deshabilitado.



Figura 4. Job SQL para ejecutar tareas

Fuente: Autor

### Notificaciones cuando falla un job

Las notificaciones llegan solo cuando falla un job, pero puede modificar esta configuración para que sean enviadas también cuando termine en forma exitosa. La Figura 5 muestra el operador SQL Server Agent: OperatorDemandaDW. En este operador debe establecer los buzones de correo.

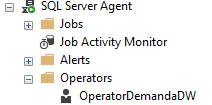


Figura 5. Operador SQL para recibir notificaciones

Fuente: Autor

### Credencial y cuenta proxy

La Figura 6 muestra la credencial y cuenta proxy. Para la ejecución de Jobs es utilizada una cuenta proxy, mapeada a un usuario de Windows, si la contraseña cambia debe ser rescrita en el servidor SQL. A esta cuenta son establecido permisos en carpetas de red y en objetos de base de datos:

|  |  |
| --- | --- |
| Credencial: CreDemandaDW | Cuenta proxy: PrxDemandaDW |

Figura 6. Credencial SQL y cuenta proxy SQL

Fuente: Autor

### Permisos en objetos de base de datos

En las bases de datos de los sistemas fuentes, en la bodega de datos y en el servidor de cubos, fueron creados roles y asignados permisos para un correcto funcionamiento. En la documentación encuentra archivos Excel con las matrices CRUD de permisos. Para mantener la documentación actualizada, estos archivos deben ser modificados cuando necesite hacer cambios en la seguridad:

\DemandaBI\Doc\Manual\MatrizCRUD\_DemandaDW.xlsm

## EJECUCIÓN DE ETLs PARA CARGA DE DATOS

### Ejecución de Job SQL

El job “**JobDemandaBI\_CargaDiaria**”, corre en forma automática todos los días, y ejecuta una sola ETL: “**pkgFull**”, la cual internamente ejecuta el SP: [Utility].[spModuleProgramming\_Sheduling], que llena la tabla de programación: [Utility].[ModuleProgramming], con los modelos que deben ser ejecutados y el rango de fechas a ser procesados. Este job también puede ser ejecutado en forma manual, en caso que necesite procesar todo en otra hora del día, pero si necesita recargar solo una parte es preferible configurar solo lo requerido. Consulte el capítulo de recarga de datos.

### Revisión de Jobs

Consulte el capítulo de auditoría o bitácora para más detalle.

# RECARGA DE DATOS

Para recargar datos es necesario adicionar filas a la tabla: [Utility].[ModuleProgramming], y ejecutar de nuevo la el job de carga diaria, pero este job adiciona la programación del día a día y ejecuta todo. Vamos a ver diferentes formas de recargar de datos, puede utilizar una combinación de estas según las necesidades.

Otra alternativa puede ser ejecutar las ETLs en forma manual o procesar las dimensiones y particiones de los cubos desde SSMS, o mediante scripts XMLA.

## TABLAS DE CONFIGURACIÓN

Como hemos vistos hay un grupo de tablas con metadatos del sistema, estas tablas deben ser actualizadas cuando cambie el sistema, para un correcto funcionamiento. Consulte los capítulos anteriores. Las siguientes consultas son útiles para consultar la programación actual o el estado de la programación que está siendo ejecutada:

/\* Programacion en forma detalla \*/

SELECT MP.[Priority], MP.[ModelId],MO.[AppName] As [Model],MP.[SequenceId],MP.[ModuleId],MD.[AppName] As [Module]

,MP.[ModelProgId],MOP.[AppName] As [ModelProg],MP.[DateVersion],MP.[DateStart],MP.[DateEnd],MP.[Version],MP.[VersionId]

,MP.[StateId],MP.[DateUpdate]

FROM [Utility].[ModuleProgramming] MP

INNER JOIN [Utility].[ModuleSystem] MD ON MD.[ModuleId] = MP.[ModuleId]

INNER JOIN [Utility].[ModelSystem] MO ON MO.[ModelId] = MP.[ModelId]

INNER JOIN [Utility].[ModelSystem] MOP ON MOP.[ModelId] = MP.[ModelProgId]

--WHERE MP.[StateId] <> 20

ORDER BY MP.[Priority],MP.[ModelId],MP.[SequenceId],MP.[DateStart];

/\* Programacion en forma compacta \*/

SELECT [Priority],[SequenceId],[ModuleId],[AppNameModule],[TypeModule],[TypeModel],[StateId]

FROM (

SELECT MP.[Priority],MP.[SequenceId],MP.[ModuleId],MD.[AppName] As [AppNameModule],MP.[StateId]

, MD.[TypeId] [TypeModule], MO.[TypeId] [TypeModel]

, ROW\_NUMBER() OVER (PARTITION BY MP.[ModuleId]

ORDER BY MP.[ModuleId], CASE WHEN MO.[TypeId] <> 'Cubo' THEN MP.[Priority] END ASC

, CASE WHEN MO.[TypeId] = 'Cubo' THEN MP.[Priority] END DESC, MP.[SequenceId]) [Fila]

FROM [Utility].[ModuleProgramming] MP

INNER JOIN [Utility].[ModuleSystem] MD ON MD.[ModuleId] = MP.[ModuleId]

INNER JOIN [Utility].[ModelSystem] MO ON MO.[ModelId] = MP.[ModelProgId]

) T

WHERE [Fila] = 1

ORDER BY [Priority],[SequenceId];

## RECARGA DE DATOS

### Recarga de dimensiones

1. Establecer que necesita cargar: dimensión, tabla de hechos, procesar dimensiones o procesar una partición de un cubo. Según lo requerido las estrategias son diferentes.
2. Si lo que necesita es cargar una dimensión, puede ejecutar la(S) ETLs que cargan las dimensiones y luego procesar las dimensiones. Para determinar dependencias, verifique la cadena de claves foráneas de una tabla. La siguiente consulta muestra el orden en el cual deben ser cargadas las dimensiones.

SELECT ML.[ModelId],MO.[AppName] As [Modelo],ML.[SequenceId] As [Secuencia],ML.[ModuleId],MD.[AppName] As [Modulo]

,ML.[ModelProgId],MOP.[AppName] As [ModeloProgramar]

FROM [Utility].[ModelLoad] ML

INNER JOIN [Utility].[ModelSystem] MO ON MO.[ModelId] = ML.[ModelId]

INNER JOIN [Utility].[ModelSystem] MOP ON MOP.[ModelId] = ML.[ModelProgId]

INNER JOIN [Utility].[ModuleSystem] MD ON MD.[ModuleId] = ML.[ModuleId]

WHERE ML.[ModelId] IN ('DW\_DDim')

ORDER BY ML.[ModelId], ML.[SequenceId];

1. Las siguientes instrucciones programan la recarga de todas las dimensiones, luego de ejecutar puede consultar las filas adicionadas a la tabla de programación y borrar las dimensiones que no requiera procesar, note que la última actividad es procesar todas las dimensiones de los cubos (DW\_CDim) y que truncamos la tabla de programación antes de iniciar.

TRUNCATE TABLE [Utility].[ModuleProgramming];

EXECUTE [Utility].[spModuleProgramming\_Reload] 0, NULL, NULL, 'DW\_DDim', 'DW\_DFull', 0, 0;

1. Revise la programación, si es necesario borre filas para que solo cargue lo que necesita.

SELECT MP.[ProgrammingId], MP.[Priority], MP.[ModelId],MP.[SequenceId], MP.[ModuleId], MD.[AppName] As [Modelo], MP.[ModelProgId]

FROM [Utility].[ModuleProgramming] MP

INNER JOIN [Utility].[ModuleSystem] MD ON MD.[ModuleId] = MP.[ModuleId]

ORDER BY MP.[Priority],MP.[ModelId],MP.[SequenceId];

/\* Para no cargar una dimension, remueva el módulo de la programación \*/

DELETE FROM [Utility].[ModuleProgramming]

WHERE [ModelId] = 'DW\_DDim' AND [ModuleId] IN (#, #, ...);

1. Luego ejecute el job: JobDemandaBI\_CargaDiaria, este job lanza la ETL: pkgFull.dtsx.

### Recarga de métricas

En el siguiente ejemplo supongamos que necesitamos recargar la métrica: “Demanda real” para los meses de junio y julo de 2016, pero no recordamos el nombre exacto de la métrica, las siguientes instrucciones nos ayudan a realizar la programación:

1. Buscar los códigos de los modelos asociados a las métricas utilizando la documentación o las siguientes consultas, puede adicionar filtros por nombres de tablas de hechos, nombres de campos, etc.

SELECT DISTINCT MO.[ModelId],MO.[AppName] As [Modelo],ME.[MetricId],ME.[AppName] As [Metrica]

,ME.[CalculationType] As [TipoMetrica],ME.[UnityId] As [UnidadId],ME.[DatabaseId] As [BaseDatosId]

,ME.[Table] As [Tabla],ME.[Field] As [Campo],ME.[QueryId],ME.[DateStart],ME.[DateEnd]

,ME.[Formula],ME.[Description] As [Descripcion]

FROM [Utility].[ModelSystem] MO

INNER JOIN [Utility].[MetricSystem] ME ON ME.[ModelId] = MO.[ModelId]

ORDER BY MO.[AppName], ME.[AppName];

SELECT \* FROM [Utility].[ModelSystem]

WHERE [TypeId] = 'DW'

ORDER BY [AppName];

1. Aplicar filtro para establecer el código de los modelos. La Figura 7 muestra el resultado de la consulta de la tabla de métricas.

/\* Consultamos los que inicie con Dem\_ \*/

SELECT [MetricId], [AppName] As [Metrica], [ModelId], [Table] As [Tabla]

FROM [Utility].[MetricSystem] WHERE [AppName] LIKE 'Dem\_%';

/\* Buscamos codigo de modelo o nombre de tabla de hechos asociada \*/

SELECT [MetricId], [AppName] As [Metrica], [ModelId], [Table] As [Tabla]

FROM [Utility].[MetricSystem] WHERE [AppName] = 'Dem\_Demanda real';



Figura 7. Consulta de métricas

Fuente: Autor

1. Pero el código de modelo encontrado tiene que ver con cubos, pero necesitamos el código del modelo parar cargar el data warehouse. Con una de las siguientes consultas podemos encontrar el código. La Figura 8 muestra el resultado de la consulta de la tabla de modelos del sistema.

/\* Con el modelo del cubo encontrar el modelo asociado \*/

SELECT DISTINCT [ModelId] FROM [Utility].[ModelLoad]

WHERE [ModelProgId] = 'DW\_CDemaP' AND [ModelId] <> 'DW\_DFull';

/\* Utilizando el nombre de la tabla fact \*/

SELECT [ModelId] FROM [Utility].[ModelSystem]

WHERE [TypeId] = 'DW' AND [AppName] = 'FactDemandaPerdida';



Figura 8. Consulta de modelo

Fuente: Autor

1. Antes de programar la recarga, borramos las filas de tabla de programación y luego ejecutamos el SP que adiciona filas a la tabla de programación con los modelos seleccionados (los datos también pueden ser insertados en forma manual). Para el ejemplo el modelo a cargar es: **DW\_DDemaP** para los meses de junio y julo de 2016. Note que adicionamos el código del modelo completo 'DW\_DFull', el cual es utilizado para establecer prioridades y que el rango de fecha no va hasta el último día si no un día más.

TRUNCATE TABLE [Utility].[ModuleProgramming];

EXECUTE [Utility].[spModuleProgramming\_Reload] 0, '2016-06-01', '2016-08-01', 'DW\_DDemaP', 'DW\_DFull', 0, 0;

**NOTA**: puede cargar otros modelos en el mismo rango de fecha, para ello, ingreses los códigos de modelos separados por coma.

1. Revise la programación, si es necesario borre filas para que solo cargue lo que necesita.

SELECT MP.[ProgrammingId], MP.[Priority], MP.[ModelId],MP.[SequenceId], MP.[ModuleId], MD.[AppName] As [Modelo], MP.[ModelProgId]

FROM [Utility].[ModuleProgramming] MP

INNER JOIN [Utility].[ModuleSystem] MD ON MD.[ModuleId] = MP.[ModuleId]

ORDER BY MP.[Priority],MP.[ModelId],MP.[SequenceId];

1. El SP anterior adiciona filas a la tabla de programación, pero si no quiere alguna de las actividades la puede borrar, por ejemplo, las siguientes instrucciones remueven las actividades de procesar dimensiones y de crear particiones.

SELECT \* FROM [Utility].[ModuleProgramming] WHERE [ModelProgId] = 'DW\_CDim' OR [ModuleId] = 4;

DELETE FROM [Utility].[ModuleProgramming] WHERE [ModelProgId] = 'DW\_CDim';

DELETE FROM [Utility].[ModuleProgramming] WHERE [ModuleId] = 4;

1. Luego ejecute el job: JobDemandaBI\_CargaDiaria, este job lanza la ETL: pkgFull.dtsx.

### Procesar dimensiones de cubos

1. Adicione una fila a la tabla de programación, por ejemplo, con la siguiente instrucción. Como las dimensiones no requieren fecha, puede escribir cualquier rango de fechas.

TRUNCATE TABLE [Utility].[ModuleProgramming];

EXECUTE [Utility].[spModuleProgramming\_Reload] 0, '2018-01-01', '2018-02-01', 'DW\_CDim', 'DW\_DFull', 1, 0;

1. Revise la programación.

SELECT \* FROM [Utility].[ModuleProgramming] ORDER BY [Priority],[ModelId],[SequenceId];

1. Luego ejecute el job: JobDemandaBI\_CargaDiaria, este job lanza la ETL: pkgFull.dtsx.

### Procesar particiones del cubo

1. Debe buscar los códigos de los modelos para los cuales procesar sus particiones, pero los códigos que debe buscar no son los asociados a los cubos, si no los asociados al data Warehouse

SELECT DISTINCT MO.[ModelId],MO.[AppName] As [Modelo],ME.[MetricId],ME.[AppName] As [Metrica]

,ME.[CalculationType] As [TipoMetrica],ME.[UnityId] As [UnidadId],ME.[DatabaseId] As [BaseDatosId]

,ME.[Table] As [Tabla],ME.[Field] As [Campo],ME.[QueryId],ME.[DateStart],ME.[DateEnd]

,ME.[Formula],ME.[Description] As [Descripcion]

FROM [Utility].[ModelSystem] MO

INNER JOIN [Utility].[MetricSystem] ME ON ME.[ModelId] = MO.[ModelId]

ORDER BY MO.[AppName], ME.[AppName];

SELECT \* FROM [Utility].[ModelSystem]

WHERE [TypeId] = 'DW'

ORDER BY [AppName];

1. Adicione filas a la tabla de programación, note el parámetro 6, es uno (1), lo que indica que solo programe el procesamiento de cubos. Cuando el sistema crea la programación la fecha inicio y fin, pueden ser diferentes a las que usted programó, esto depende de la configuración de particiones para cada modelo (anual, mensual, …). Puede programar varios modelos, separándolos por coma:

TRUNCATE TABLE [Utility].[ModuleProgramming];

EXECUTE [Utility].[spModuleProgramming\_Reload] 0, '2016-06-01', '2016-08-01', 'DW\_DDemaP', 'DW\_DFull', 1, 0;

1. Revise la programación, si es necesario modifíquela.

SELECT \* FROM [Utility].[ModuleProgramming] ORDER BY [Priority],[ModelId],[SequenceId];

1. Luego ejecute el job: JobDemandaBI\_CargaDiaria, este job lanza la ETL: pkgFull.dtsx.

### Procesar todos los cubos

1. Adicione filas a la tabla de programación, en este caso utilizamos el modelo full el cual contiene todos los modelos e indicamos que solo queremos procesar dimensiones y cubos

EXECUTE [Utility].[spModuleProgramming\_Reload] 0, '2016-01-01', '2017-01-01', 'DW\_DFull', 'DW\_DFull', 1, 0;

1. Revise la programación, si es necesario modifíquela.

SELECT \* FROM [Utility].[ModuleProgramming] ORDER BY [Priority],[ModelId],[SequenceId];

1. Luego ejecute el job: JobDemandaBI\_CargaDiaria, este job lanza la ETL: pkgFull.dtsx.

## ASPECTO A MEJORAR PARA LA RECARGA DE DATOS

La ejecución de la ETL presenta la desventaja de cargar todas las tablas para casos donde solo es requerido recargar una parte del sistema, para estos casos es recomendable crear una ETL para recarga de datos (pkgReload.dtsx) y un job para ejecutar esta ETL (JobDemandaBI\_Recarga).

# AUDITORÍA O BITÁCORA DE EJECUCIÓN DE PROCESOS

El seguimiento a la ejecución de procesos puede ser realizada de varias formas, a continuación, será explicado cada una de las formas de consultar el estado del sistema:

* Tablas de bitácora
* Archivos de log
* Revisión de Jobs
* Reportes de auditoría
* Reportes propios de SQL de ejecución de ETL
* Logs SQL Server
* Estado de suscripciones de reportes

## TABLAS DE BITÁCORA

La Tabla 3 lista las tablas de bitácora para el seguimiento de procesos, estas tablas están en el esquema Audit de la base de datos del DW.

Tabla 3. Tablas de bitácora

|  |  |
| --- | --- |
| **Tabla** | **Descripción** |
| [Audit].[Bitacora] | Bitácora de ejecución de procesos del sistema. |
| [Audit].[BitacoraFile] | Información de auditoría de archivos. Los archivos son tomados de PathSource, pasan luego a PathTarget, si está cifrado pasa a PathEncrypted y el archivos descifrado pasa a PathWithoutProcessing, si no está cifrado pasa a PathWithoutProcessing, si el procesamiento es exitoso pasa a PathProcessed y opcionalmente a PathProcessed2, pero si falla pasa a PathError. |
| [Audit].[BitacoraDetail] | información de ejecución de procesos. |
| [Audit].[BitacoraStatistic] | Estadísticas de ejecución como tiempo de ejecución, filas procesadas y filas con error. |
| [Audit].[BitacoraState] | Estados para las tablas de bitácora. |
| [Audit].[BitacoraTable] | Información de ejecución de procesos detallando las tablas involucradas y 5 valores de las columnas origen principales. |
| [Audit].[BitacoraType] | Tipos de inconsistencias de bitácora. |

La Tabla 4, lista los estados de ejecución de procesos, los cuales pueden ser consultados en la tabla de estados de bitácora.

SELECT \* FROM [Audit].[BitacoraState] ORDER BY [StateId];

Tabla 4. Estados de ejecución de procesos

|  |  |  |  |
| --- | --- | --- | --- |
| **StateId** | **Name** | **StateId** | **Name** |
| 0 | Exitoso | 8 | Generado |
| 1 | Error | 9 | Definitivo |
| 2 | Iniciado | 10 | Publicado |
| 3 | Cancelado | 11 | Existente |
| 4 | Abierto | 12 | Eliminado |
| 5 | Cerrado | 13 | Sin fuente de datos |
| 6 | Activo | 20 | Programado para ejecutar |
| 7 | Inactivo | 21 | Reprogramar ejecución |

La Tabla 5, lista los tipos de errores de procesos, los cuales pueden ser consultados en la tabla de tipos de bitácora

SELECT \* FROM [Audit].[BitacoraType] ORDER BY [TypeId];

Tabla 5. Tipos de errores de ejecución de procesos

|  |  |  |  |
| --- | --- | --- | --- |
| **TypeId** | **Name** | **Group** | **Description** |
| 1 | General | General | General |
| 2 | Conversion | General | Conversión de tipo de datos |
| 3 | Datos ausente | General | Datos no suministrados |
| 4 | Repetido | General | Datos repetidos |
| 5 | Clave FK | General | Clave foránea no encontrada |
| 6 | Inferir FK | General | Clave foránea inferida adicionada |
| 7 | Valor no permitido | General | Valor no permitido |
| 8 | Operacion no valida | General | Operación no valida |
| 9 | Dim tipo 2 DateStart | General | Dim tipo 2 fecha inicio igual, fila actualizada sin insertar versión |
| 10 | Informacion | Informacion | Información |
| 11 | Calidad | General | Calidad de datos |
| 12 | Particion | Particion | Operaciones CRUD con partición de cubos o tablas |
| 13 | Mapeo | Mapeo | Mapeo de código |
| 14 | ErrorFuente | General | Error o ajuste en la fuente de datos |

Puede consultar las tablas de bitácora en forma directa, usar funciones que retornan datos o ejecutar reportes creados con Reporting Services:

/\* ===================================

DATOS AUDITORIA

=================================== \*/

SELECT \* FROM [Audit].[Bitacora];

SELECT \* FROM [Audit].[BitacoraFile];

SELECT \* FROM [Audit].[BitacoraDetail];

SELECT \* FROM [Audit].[BitacoraStatistic];

SELECT \* FROM [Audit].[BitacoraState] ORDER BY [StateId];

SELECT \* FROM [Audit].[BitacoraType] ORDER BY [TypeId];

SELECT \* FROM [Audit].[BitacoraTable];

/\* ===================================

FUNCIONES PARA CONSULTAR AUDITORIA

=================================== \*/

* [Audit].[GetBitacoraDetailTree]
* [Audit].[GetBitacoraDetailTree\_ByDate]
* [Audit].[GetBitacoraFileTree]
* [Audit].[GetBitacoraFileTree\_ByDate]
* [Audit].[GetBitacoraStatisticTree]
* [Audit].[GetBitacoraStatisticTree\_ByDate]
* [Audit].[GetBitacoraTableTree]
* [Audit].[GetBitacoraTableTree\_ByDate]
* [Audit].[GetBitacoraTree]
* [Audit].[GetBitacoraTree\_ByDate]

SELECT \* FROM [Audit].[GetBitacoraTree]((SELECT Max([BitacoraId]) FROM [Audit].[Bitacora]), 1);

SELECT \* FROM [Audit].[GetBitacoraStatisticTree]((SELECT Max([BitacoraId]) FROM [Audit].[Bitacora]), 1);

SELECT \* FROM [Audit].[GetBitacoraTableTree]((SELECT Max([BitacoraId]) FROM [Audit].[Bitacora]), 1);

SELECT \* FROM [Audit].[GetBitacoraDetailTree]((SELECT Max([BitacoraId]) FROM [Audit].[Bitacora]), 1)

SELECT \* FROM [Audit].[GetBitacoraDetailTree](41, 1)

SELECT \* FROM [Audit].[GetBitacoraFileTree]((SELECT Max([BitacoraId]) FROM [Audit].[Bitacora]), 1);

## ARCHIVOS DE LOG

La Figura 9 muestra el log en archivo plano de ejecución, cada paquete deja un archivo en una ruta configurada en el parámetro de la ETL: FolderLog.

El nombre de cada archivo log, tiene como formato: NombreETL.csv, por ejemplo: pkgFull.csv. Este es un archivo separado por comas que puede ser abierto en un block de notas o en Excel. Para revisarlo vaya hasta el final del archivo y busque, de abajo hacia arriba, las líneas ente PackageStart y PackageEnd, si existe algún mensaje de error.

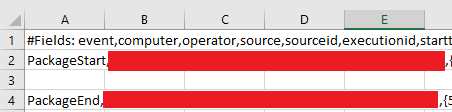


Figura 9. Log en archivo planos de ejecución de ETLs

Fuente: Autor

## REVISIÓN DE JOBS

### Historial de Jobs

Los Jobs cuentan con un historial de ejecución, el cual informa el estado de ejecución y las causas de error, esta es otra forma de auditoría. Para consultar el historial, de haga clic secundario en el Job y seleccione “**Ver Historial**”. La desventaja de este método es que el mensaje, puede que no aporte mucho.

### Notificación por Correo electrónico

Como fue mencionado los Jobs SQL, envían correo a los buzones de correo definidos en el operador: OperatorDemandaDW.

## REPORTES DE AUDITORÍA

Otra forma de consultar las tablas de bitácora es mediante reportes Reporting Services, los reportes se encuentran en el servidor de reportes en la carpeta:

DemandaBI\Auditoria

* Audit\_Bitacora
* Audit\_BitacoraArchivo
* Audit\_BitacoraDetalle
* Audit\_BitacoraEstadisticas
* Audit\_BitacoraTabla

La Figura 10 muestra el reporte de bitácora del sistema, en la carpeta de reportes puede consultar todos los reportes creados para el proyecto.

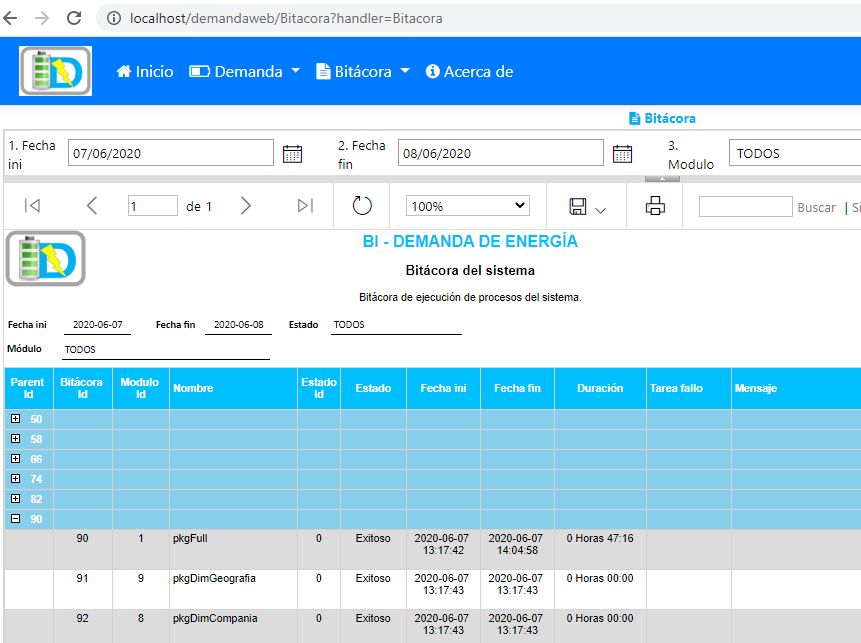


Figura 10. Reporte de bítacora del sistema

Fuente: Autor

## REPORTES PROPIOS DE SQL DE EJECUCIÓN DE ETL

SQL contiene reportes para revisar el estado de ejecución de ETLs. Por ejemplo, para revisar el estado de ejecución de la ETL pkgFull.dtsx haga lo siguiente:

1. En SSMS, expanda los nodos “**Integration services catalogs**”, **SSISDB**, Demanda, **Projects**, DemandaBI\_SSIS, **Packages**.
2. Haga clic secundario en la ETL: pkgFull.dtsx, seleccione el comando **Reports**, “**Standard reports**”, y haga clic en “**All executions**” La Figura 11 muestra el reporte de ejecución de ETL, el reporte tiene links para navegar a los detalles de ejecución, también puede exportar o imprimir el reporte.

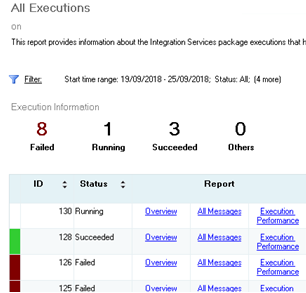


Figura 11. Reporte propio de SQL para consultar ejecución de ETLs

Fuente: Autor

## LOG SQL SERVER

Otro lugar donde puede revisar errores es el log de SQL Server el cual lo encuentra en SSMS, nodo **Management** (**Administración**), “**SQL Server Logs**” (Registros de SQL Server). Consulte el log de SQL Server para errores generales a nivel de todo el motor de base de datos.

# ADMINISTRACIÓN DE CUBOS

## SEGURIDAD

Para el usuario poder consultar los cubos debe tener permisos suficientes, los permisos son manejados a través de roles.

## CAMBIOS DISEÑO DE DIMENSIONES Y PROCESAMIENTO

Las dimensiones son procesadas por una de las ETLs, en modo de procesamiento actualización, si cambia el diseño de dimensiones o hay cambios poco comunes, por ejemplo, cambios en relaciones rígidas, el proceso falla, para solucionar debe reprocesar la dimensión manualmente en forma completa. Otra alternativa es cambiar, en forma temporal, el parámetro de ETLs “IsProcessFull\_pkgCubeDimProcess” que define el modo de procesamiento de dimensiones. La Figura 12 muestra este parámetro para establecer la forma como procesar las dimensiones del cubo.



Figura 12. Parámetro procesamiento de dimensiones en forma completa o actualización

Fuente: Autor

# CONFIGURACIÓN DE MODELOS DEL SISTEMA

## MODELOS

Para un BI son creado modelos para representar el sistema en forma lógica. Para nuestro caso los modelos son representadas por Fact(s) que contienen las métricas del negocio y sus dimensiones relacionadas. Un modelo puede ser una tabla de hechos, un grupo de medidas de un cubo, etc.

## METADATOS

La descripción de los campos de cada tabla puede ser consultada en la metadata del motor de base de datos, en el diccionario de datos o el archivo de metadatos de modelos. También fueron creado tablas personalizadas de utilidades para la administración del sistema, las cuales serán descritas adelante.

### Diccionario de datos

\DemandaBI\Doc\Analisis\DicDatos\_DemandaDW.xlsm

### Consulta de los objetos de base de dato data warehouse

La siguiente consulta, es similar al diccionario de datos y muestra metadata de tablas:

SELECT CASE o.type WHEN 'U' THEN 'Tabla' WHEN 'V' THEN 'Vista' ELSE o.type END [Type],

QUOTENAME(SCHEMA\_NAME(O.schema\_id)) + '.' + QUOTENAME(O.name) As [Name],

SCHEMA\_NAME(o.schema\_id) As [Esquema], O.name As [Table], C.column\_id As [ColumnaId], c.name As [Field],

CASE WHEN T.name IN ('varbinary', 'nvarchar', 'binary', 'char', 'nvarchar', 'nchar', 'sysname') THEN

T.name + ' (' + CASE WHEN C.max\_length = - 1 THEN 'max' ELSE CAST(C.max\_length AS nvarchar) END + ')'

WHEN T.name IN ('numeric', 'real', 'money', 'float', 'decimal') THEN T.name + ' (' + CAST(C.precision AS nvarchar) + ', ' + CAST(C.scale AS nvarchar) + ')'

ELSE T.name END + CASE WHEN c.is\_identity = 1 THEN ' (IDENTITY)' ELSE '' END AS [TipoDato],

CASE WHEN ic.column\_id IS NULL THEN 'NO' ELSE 'SI' END [EsPK],

CASE WHEN C.is\_nullable = 0 THEN 'SI' ELSE 'NO' END AS [Requerido],

CASE WHEN C.is\_computed = 0 THEN 'NO' ELSE 'SI' END AS [EsComputado],

ISNULL(exT.value, '') As [TablaDescripcion],

ISNULL(ex.value, '') As [Description],

ISNULL(CC.definition, '') [FormulaColCalculada],

LEFT(SCHEMA\_NAME(o.schema\_id), 3) + '\_' + OBJECT\_NAME(c.object\_id) AS [Hoja]

FROM sys.columns c

INNER JOIN sys.objects o on c.object\_id = o.object\_id

INNER JOIN sys.types T ON C.system\_type\_id = T.system\_type\_id AND T.system\_type\_id = T.user\_type\_id

LEFT JOIN sys.extended\_properties exT

ON exT.major\_id = c.object\_id AND exT.minor\_id = 0 AND exT.name = 'MS\_Description'

LEFT JOIN sys.extended\_properties ex

ON ex.major\_id = c.object\_id AND ex.minor\_id = c.column\_id AND ex.name = 'MS\_Description'

LEFT JOIN sys.indexes i ON i.object\_id = o.object\_id AND i.is\_primary\_key = 1

LEFT JOIN sys.index\_columns ic on ic.object\_id = o.object\_id AND ic.index\_id = i.index\_id AND ic.column\_id = c.column\_id

LEFT JOIN sys.computed\_columns CC on cc.object\_id = c.object\_id AND cc.column\_id = c.column\_id AND cc.is\_computed = c.is\_computed AND cc.is\_computed = 1

WHERE o.type IN ('U', 'V')

-- AND SCHEMA\_NAME(o.schema\_id) IN ('Utility') AND OBJECT\_NAME(c.object\_id) IN ('SpecialDate')

ORDER BY O.type, SCHEMA\_NAME(O.schema\_id), O.name, c.column\_id;

## TABLAS DE UTILIDADES

Para la administración del sistema, fueron creadas tablas de base de datos en el esquema: [Utility]. La Figura 13 muestra las tablas de utilidades que serán descritas a continuación.

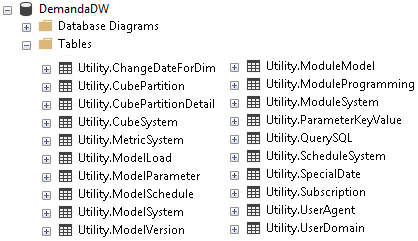


Figura 13. Tablas de utilidades

Fuente: Autor

### [Utility].[ChangeDateForDim]

Almacena fecha de cambio para dimensiones tipo 2, utilizada cuando los datos origen no manejan vigencia o la fecha de cambio es nula.

En un Data Warehouse existe el concepto de dimensiones tipo 1 y tipo 2. Las dimensiones tipo 1 no guardan historia, y si hay un cambio los datos son sobrescritos, por ejemplo, DimGeografia. Las dimensiones tipo 2 guardan historia de cambios de algunos o todos sus campos, tienen campos adicionales para el manejo de vigencia: Activa (indica cual es la vigencia activa), FechaIni y FechaFin.

Cuando la fuente de datos no contiene la vigencia de cambios de las tablas maestras, es necesario identificar la fecha de inicio de vigencia de cada fila de datos y la fecha a partir de la cual marcar un nuevo cambio para definir las vigencias. Para estos casos puede configurar el manejo de vigencias para dimensiones tipo 2 en la tabla: [Utility].[ChangeDateForDim]. Para nuestro sistema no aplica, puesto que la fuente de datos incluye vigencia para las tablas de dimensiones:

SELECT \* FROM [Utility].[ChangeDateForDim];

Para cada tabla hay dos filas. Cuando la ETL ejecuta el sp de carga de dimensión (spLoadDimNombre):

* Campo [UseCurrentDate] = 0. En el rango de fechas [1990-01-01, hasta la fecha de instalación + 2 meses], el inicio de vigencia de cada fila nueva es una fecha mínima = 1900-01-01. Para cambios tipo2 es utilizada la fecha de ejecución como cierre de vigencia de la fila anterior e inicio de vigencia de la nueva fila.
* Campo [UseCurrentDate] = 1. En el rango de fechas [Fecha de instalación + 2 meses, sin fecha fin (siempre nulo)], el inicio de vigencia de cada fila nueva es la fecha de ejecución menos el valor del campo [DaysToDecrease] (generalmente un día). Para cambios tipo2 es utilizada la fecha de ejecución como cierre de vigencia de la fila anterior e inicio de vigencia de la nueva fila.

Para adicionar filas a esta tabla debe cerrar la vigencia de la fila anterior [FechaFin], con la fecha deseada e iniciar una nueva fila con [FechaIni] igual a esta misma fecha y [FechaFin] = NULL.

### [Utility].[CubePartition]

Almacena la forma de partir los datos de grupos de medidas de los cubos de datos. En esta tabla debe indicar la consulta de selección de datos (campo: [QueryId]), el lugar en disco donde serán almacenado los datos (Campo: [StoragePath]). El campo [Period], indica la forma de partir los datos, en este proyecto son utilizado los valores: Ninguna, Año y Mes.

Valores permitidos para el campo [Period]:

* Ninguna: Indica que el grupo de medidas no tiene particiones.
* Dia
* Semana
* Mes
* Bimestre
* Trimestre
* Semestre
* Año

Consulta tipo de partición en el cubo de modelos

SELECT CU.[CubeId],CU.[AppName] As [Cubo],MO.[ModelId],MO.[AppName] As [Modelo]

,CP.[Partitioned],CP.[Period],CP.[PrefixPartition],CP.[StoragePath]

FROM [Utility].[CubeSystem] CU

INNER JOIN [Utility].[CubePartition] CP ON CP.[CubeId] = CU.[CubeId]

INNER JOIN [Utility].[ModelSystem] MO ON MO.[ModelId] = CP.[ModelId]

ORDER BY CU.[Name], MO.[Name];

### [Utility].[CubePartitionDetail]

Almacena los rangos de fecha de las particiones de los grupos de medidas de modelos. Esta tabla. Los datos de esta tabla son utilizados por las ETL de creación y procesamiento de particiones. Solo modifique los datos en forma manual si tiene problemas con las particiones existentes. Para llenar los datos las ETLs utilizan el SP: [Utility].[spOLAP\_PartitionDefine]. Por ejemplo, puede usar este SP, para adelantar las particiones del siguiente año para un modelo:

EXECUTE [Utility].[spOLAP\_PartitionDefine] 0, 'DemandaBI', 'DW\_CDemaP', '2000-01-01', '2020-01-01', 0;

Consulta particiones de cubos

SELECT CU.[CubeId],CU.[AppName] As [Cubo],MO.[ModelId],MO.[AppName] As [Modelo]

,CP.[Partitioned],CP.[Period],CP.[PrefixPartition]

,PD.[PartitionId],PD.[PartitionName],PD.[DateStart],PD.[DateEnd],PD.[Create]

FROM [Utility].[CubeSystem] CU

INNER JOIN [Utility].[CubePartition] CP ON CP.[CubeId] = CU.[CubeId]

INNER JOIN [Utility].[ModelSystem] MO ON MO.[ModelId] = CP.[ModelId]

INNER JOIN [Utility].[CubePartitionDetail] PD ON PD.[CubeId] = CU.[CubeId] AND PD.[ModelId] = CP.[ModelId]

ORDER BY CU.[Name], MO.[Name], PD.[PartitionId];

### [Utility].[CubeSystem]

Almacena los cubos de datos del sistema. Un cubo está formado por modelos, los cuales contiene métricas. En este sistema solo hay un único cubo.

SELECT [CubeId],[Name],[AppName],[Database],[Description]

FROM [Utility].[CubeSystem]

ORDER BY [Name];

### [Utility].[MetricSystem]

Almacena métricas o variables del sistema, las cuales están asociadas a un grupo de medidas (modelo). Los datos de esta tabla pueden ser utilizados como metadatos para consultar las variables del sistema. Consulta métricas

SELECT DISTINCT MO.[ModelId],MO.[AppName] As [Modelo],ME.[MetricId],ME.[AppName] As [Metrica]

,ME.[CalculationType] As [TipoMetrica],ME.[UnityId] As [UnidadId],ME.[DatabaseId] As [BaseDatosId]

,ME.[Table] As [Tabla],ME.[Field] As [Campo],ME.[QueryId],ME.[DateStart],ME.[DateEnd]

,ME.[Formula],ME.[Description] As [Descripcion]

FROM [Utility].[ModelSystem] MO

INNER JOIN [Utility].[MetricSystem] ME ON ME.[ModelId] = MO.[ModelId]

ORDER BY MO.[AppName], ME.[AppName];

### [Utility].[ModelLoad]

Almacena el orden de actividades a ser ejecutados para cargar los modelos del sistema. El orden de proceso es muy importante, por ejemplo, cargar tablas de dimensiones, luego tablas de hechos, procesar dimensiones de cubos y por último procesar particiones de cubos. Los datos de esta tabla son utilizados para la recarga de modelos. Consulta secuencia de pasos para recargar modelos

SELECT ML.[ModelId],MO.[AppName] As [Modelo],ML.[SequenceId] As [Secuencia],ML.[ModuleId],MD.[AppName] As [Modulo]

,ML.[ModelProgId],MOP.[AppName] As [ModeloProgramar]

FROM [Utility].[ModelLoad] ML

INNER JOIN [Utility].[ModelSystem] MO ON MO.[ModelId] = ML.[ModelId]

INNER JOIN [Utility].[ModelSystem] MOP ON MOP.[ModelId] = ML.[ModelProgId]

INNER JOIN [Utility].[ModuleSystem] MD ON MD.[ModuleId] = ML.[ModuleId]

ORDER BY MO.[TypeId] DESC,ML.[ModelId], ML.[SequenceId];

### [Utility].[ModelParameter]

Almacena parámetros para la ejecución de modelos.

SELECT [ModelId],[ParameterId],[Value],[Description]

FROM [Utility].[ModelParameter];

### [Utility].[ModelSchedule]

Almacena la(s) agenda(s) o día de ejecución y el rango de fechas (consultas de carga, procesamiento particiones, etc) para procesar modelos. Esta tabla define dos elementos, el día(s) de ejecución de los modelos (tablas de hechos, procesamiento de cubos, etc) y el rango de fechas, por ejemplo, cargar el día anterior, procesar la partición del cubo para el mes actual, etc. Consulta agendas (días de ejecución) y rango de fecha para procesar modelos

SELECT MO.[ModelId],MO.[AppName] As [Modelo],MS.[ScheduleId],S.[Name] [Agenda],S.[Description] As [DescripcionAgenda]

,MS.[UseVersion],MS.[PeriodType],MS.[PeriodQuantity],MS.[PeriodTypeBefore],MS.[PeriodsBefore],MS.[Enabled],MS.[Description] As [Descripcion]

,Convert(date, [Utility].[GetSchedule\_NextExecution](MS.[ScheduleId], GetDate(), 1)) [SiguienteEjecucion]

FROM [Utility].[ModelSystem] MO

INNER JOIN [Utility].[ModelSchedule] MS ON MS.[ModelId] = MO.[ModelId]

INNER JOIN [Utility].[ScheduleSystem] S ON S.[ScheduleId] = MS.[ScheduleId]

ORDER BY MO.[TypeId] DESC,MO.[ModelId],MS.[ScheduleId];

### [Utility].[ModelSystem]

Almacena los modelos del sistema. Un modelo es como un grupo de métricas el cual está formada por Fact(s) y sus dimensiones relacionadas.

SELECT [ModelId],[Name],[AppName],[TypeId],[UseDatesProcess],[UseVersion],[ReloadPeriodType],[Description]

FROM [Utility].[ModelSystem]

ORDER BY [TypeId] DESC,[ModelId];

### [Utility].[ModelVersion]

Almacena código de versión de los datos de modelos. Las versiones son creadas por ajustes o cambios en los datos con el tiempo. Para este proyecto no es utilizada esta tabla, puesto que las fuentes de datos de movimiento no manejan versión. Consulta versión de modelos

SELECT MV.[DateVersion],MV.[ModelId],MO.[Name],MV.[DateStart],MV.[DateEnd]

,MV.[Version],MV.[VersionId],MV.[StateId]

,MV.[CreateBitacoraId],MV.[CreateDate],MV.[UpdateBitacoraId],MV.[UpdateDate]

FROM [Utility].[ModelVersion] MV

INNER JOIN [Utility].[ModelSystem] MO ON MO.[ModelId] = MV.[ModelId]

ORDER BY MV.[DateVersion], MO.[Name];

### [Utility].[ModuleModel]

Almacena la secuencia de módulos (ejemplo: ETL, servicio) a ser ejecutados para procesar los modelos del sistema. Los modelos son agrupaciones lógicas, mientras que los módulos son como aplicaciones. Consulta secuencia de módulos para procesar modelos

SELECT MM.[ModelId],MO.[AppName] As [Modelo],MM.[SequenceId],MM.[ModuleId],MD.[AppName] As [Modulo]

FROM [Utility].[ModuleModel] MM

INNER JOIN [Utility].[ModelSystem] MO ON MO.[ModelId] = MM.[ModelId]

INNER JOIN [Utility].[ModuleSystem] MD ON MD.[ModuleId] = MM.[ModuleId]

ORDER BY MO.[TypeId] DESC,MO.[AppName],MM.[SequenceId];

### [Utility].[ModuleProgramming]

Almacena programación de ejecución de módulos, por ejemplo, la ejecución de ETLs para cargar un data warehouse. Consultar programación

SELECT MP.[Priority], MP.[ModelId],MO.[AppName] As [Model],MP.[SequenceId],MP.[ModuleId],MD.[AppName] As [Module]

,MP.[ModelProgId],MOP.[AppName] As [ModelProg],MP.[DateVersion],MP.[DateStart],MP.[DateEnd],MP.[Version],MP.[VersionId]

,MP.[StateId],MP.[DateUpdate]

FROM [Utility].[ModuleProgramming] MP

INNER JOIN [Utility].[ModuleSystem] MD ON MD.[ModuleId] = MP.[ModuleId]

INNER JOIN [Utility].[ModelSystem] MO ON MO.[ModelId] = MP.[ModelId]

INNER JOIN [Utility].[ModelSystem] MOP ON MOP.[ModelId] = MP.[ModelProgId]

--WHERE MP.[StateId] <> 20

ORDER BY MP.[Priority],MP.[ModelId],MP.[SequenceId],MP.[DateStart];

SELECT [Priority],[SequenceId],[ModuleId],[AppNameModule],[TypeModule],[TypeModel],[StateId]

FROM (

SELECT MP.[Priority],MP.[SequenceId],MP.[ModuleId],MD.[AppName] As [AppNameModule],MP.[StateId]

, MD.[TypeId] [TypeModule], MO.[TypeId] [TypeModel]

, ROW\_NUMBER() OVER (PARTITION BY MP.[ModuleId]

ORDER BY MP.[ModuleId], CASE WHEN MO.[TypeId] <> 'Cubo' THEN MP.[Priority] END ASC

, CASE WHEN MO.[TypeId] = 'Cubo' THEN MP.[Priority] END DESC, MP.[SequenceId]) [Fila]

FROM [Utility].[ModuleProgramming] MP

INNER JOIN [Utility].[ModuleSystem] MD ON MD.[ModuleId] = MP.[ModuleId]

INNER JOIN [Utility].[ModelSystem] MO ON MO.[ModelId] = MP.[ModelProgId]

) T

WHERE [Fila] = 1

ORDER BY [Priority],[SequenceId];

La ETL que ejecuta todo: pkgFull, al inicio ejecuta el siguiente SP, el cual utiliza la configuración de las agendas para llenar la tabla de programación:

EXECUTE [Utility].[spModuleProgramming\_Sheduling] 0, 'DW\_DFull';

La ETL de recarga: pkgReload, utiliza el siguiente SP para consultar la programación:

EXECUTE [Utility].[spModuleProgramming\_Reload\_Query];

Las ETLs que utilizan fechas de programación, utilizan el siguiente SP para consultar la programación:

EXECUTE [Utility].[spModuleProgramming\_Query] NULL, 'pkgFactDemandaPerdida', 'ETL';

Para recargar datos, puede adicionar datos a esta tabla y luego ejecutar las ETLs respectivas, pero es preferible utilizar el siguiente SP el cual adiciona filas y secuencia de ejecución en la forma apropiada:

EXECUTE [Utility].[spModuleProgramming\_Reload] 0, '2019-09-01', '2019-10-01', 'DW\_DDemaP', 'DW\_DFull', 0, 0;

### [Utility].[ModuleSystem]

Almacena módulos del sistema, los módulos hacen parte de aplicaciones o programas de cómputo, por ejemplo, una ETL para carga de datos. Para este sistema los módulos son ETLs.

SELECT [ModuleId],[Name],[AppName],[TypeId],[ProcessTypeId],[Description]

FROM [Utility].[ModuleSystem]

ORDER BY [TypeId],[ProcessTypeId] DESC,[ModuleId];

### [Utility].[ParameterKeyValue]

Almacena parámetros de ejecución de procesos como jobs y ETLs. Consulte el SP: [Utility].[spJob\_StartJob\_ParameterKeyValue]. Al inicio del proyecto no es utilizada esta tabla, pero puede ser útil, para definir parámetros de ejecución de procesos.

### [Utility].[QuerySQL]

Almacena consultas de selección de datos. Las ETLs utilizan esta tabla para definir la consulta de selección de datos.

SELECT [QueryId],[Name],[Type],[DatabaseId],[QuerySQL],[Parameters],[Description]

FROM [Utility].[QuerySQL]

ORDER BY [QueryId];

El campo [Type], indica el tipo de consulta de selección de datos: F: Función, S: Procedimiento almacenado, T: Texto (consulta a tabla o vista), X: XML. Con los datos de esta tabla es creada la consulta por medio del SP: [Utility].[spGetQuerySQL].

### [Utility].[ScheduleSystem]

Almacena agenda de ejecución de procesos, por ejemplo, la fecha de ejecución de una ETL. El diseño es similar a las agendas de los Jobs SQL Server.

SELECT S.[ScheduleId],S.[Name] As [Agenda],S.[Period],S.[PeriodHowMany],S.[DayPeriod],S.[DayHowMany]

,S.[Monday],S.[Tuesday],S.[Wednesday],S.[Thursday],S.[Fryday],S.[Saturday],S.[Sunday]

,S.[OccursDayOfMonth],S.[DayOccurs],S.[OccursThe],S.[OccursTheDay],S.[Description]

FROM [Utility].[ScheduleSystem] S

ORDER BY S.[ScheduleId];

Estos son ejemplos de agendas, creadas al momento de instalar, pero puede adicionar más agendas y asociarlas a los modelos mediante la tabla: [Utility].[ModelSystem], con estos datos la ETL se encarga en forma automática de crear la programación diaria de lo que debe ser ejecutado y el rango de datos a ser procesados:

* Cada 1 día, en el día una vez a las 6:00 am
* Cada 1 semana de lunes a viernes; en el día una vez a las 6:00:00 am
* Cada 1 semana de sábado a domingo; en el día una vez a las 6:00:00 am
* Cada 1 mes el primer domingo; en el día una vez a las 6:00:00 am
* Cada 1 mes el segundo sábado; en el día una vez a las 6:00:00 am
* Cada 1 mes el día 10; en el día una vez a las 6:00:00 am
* Cada 3 mes el primer sábado; en el día una vez a las 6:00:00 pm

### [Utility].[SpecialDate]

Almacena días festivos y fechas especiales. Es utilizada para definir el campo: [DiaTipo] de la tabla: [DW].[DimFecha]. Adelante ampliamos el tema.

### [Utility].[Subscription]

Almacena datos para configurar suscripciones de reportes Reporting Services

SELECT [SKId], [FileName], [Path], [RenderFormat], [WriteMode], [FileExtension], [UserName], [Password], [FileShareAccount], [DateName]

FROM [Utility].[Subscription];

### [Utility].[UserAgent]

Almacena usuarios asociados a un agente. La información puede ser utilizada para definir permisos por usuario. Al momento esta tabla no es utilizada.

### [Utility].[UserDomain]

Almacena usuario de dominio del sistema operativo al cual está asociado un usuario del sistema. Al momento esta tabla no es utilizada.

# FUENTES DE DATOS

La fuente de datos para el sistema son archivos planos. Los archivos deben seguir las siguientes normas.

## CONFIGURACIÓN ARCHIVOS PLANOS

Los archivos planos son configurados en las ETLs en los parámetros de configuración, para mayor detalle consulte el manual de instalación.

## FORMATOS DE ARCHIVOS

Los datos serán cargados con archivos planos tipo CSV, con las siguientes características:

* Las columnas deben ser separadas con coma (,).
* La primera fila debe contener el nombre de las columnas
* Las fechas deben estar en formato: yyy-MM-dd, si contiene horas deben ser en formato yyyy-MM-dd HH:mm:ss
* El separador decimal debe ser punto (.)
* No es obligatorio que las cadenas estén dentro de comillas dobles, solo es obligatorio si el texto contiene caracteres como coma (,) o comillas simples (‘).
* Si hay un valor nulo, el campo debe estar vacío
* No puede ser alterado el orden y número de columnas, si un dato no existe deben estar las comas
* Entre las filas no debe existir filas en blanco

### Archivo dimensión agente

La Tabla 6 describe las columnas del archivo para la dimensión agente

Tabla 6. Formato archivo para la dimensión agente

|  |  |  |  |
| --- | --- | --- | --- |
| **Columna** | **Tipo datos** | **Descripción** | **Ejemplo** |
| AgenteId | Texto (20) | Código de agente | Ung0022 |
| AgenteMemId | Texto (20) | Código de agente en el MEM | ADCG |
| Nombre | Texto (100) | Nombre del agente | AGUAS DE LA CABAÑA S.A. E.S.P. |
| Actividad | Texto (20) | Actividad económica | GENERACIÓN |
| CompaniaId | Texto (20) | Código de compañía | 0 |
| Activo | 0 o 1 | Indica si está activo o no | 1 |
| FechaIni | Fecha formato: YYYY-MM-DD | Fecha inicio vigencia | 2008-03-20 |
| FechaFin | Fecha formato: YYYY-MM-DD | Fecha fin de vigencia |  |

### Archivo dimensión compañía

La Tabla 7 describe las columnas del archivo para la dimensión agente

Tabla 7. Formato archivo para la dimensión compañía

|  |  |  |  |
| --- | --- | --- | --- |
| **Columna** | **Tipo datos** | **Descripción** | **Ejemplo** |
| CompaniaId | Texto (20) | Código de compañía | Cia0001 |
| Nombre | Texto (100) | Nombre | CENTRAL HIDROELECTRICA DE BETANIA S.A. E.S.P. |
| Sigla | Texto (100) | Siglas | CHB |
| Nit | Texto (20) | Nit | 8600638758 |
| TipoPropiedad | Texto (50) | Tipo de propiedad | PRIVADA |
| Activa | 0 o 1 | Indica si está activa o no | 1 |
| FechaIni | Fecha formato: YYYY-MM-DD | Fecha inicio vigencia | 1995-07-20 |
| FechaFin | Fecha formato: YYYY-MM-DD | Fecha fin de vigencia | 2007-09-01 |

### Archivo dimensión geografía

La Tabla 8 describe las columnas del archivo para la dimensión agente

Tabla 8. Formato archivo para la dimensión geografía

|  |  |  |  |
| --- | --- | --- | --- |
| **Columna** | **Tipo datos** | **Descripción** | **Ejemplo** |
| PaisId | Texto (10) | Código país | Pai0001 |
| Pais | Texto (150) | Nombre país | COLOMBIA |
| DepartamentoId | Texto (10) | Código departamento | Dep0001 |
| Departamento | Texto (150) | Nombre departamento | AMAZONAS |
| MunicipioId | Texto (10) | Código municipio | Mpi0002 |
| Municipio | Texto (150) | Nombre municipio | EL ENCANTO |
| AreaId | Texto (10) | Código área geográfica | Are0128 |
| Area | Texto (150) | Nombre área geográfica | AREA SUROCCIDENTAL |
| SubAreaId | Texto (10) | Código subárea geográfica | Are0026 |
| SubArea | Texto (150) | Nombre subárea geográfica | SUBAREA CAQUETA |

### Archivo de movimiento demanda y pérdidas de energía

La Tabla 9 describe las columnas del archivo para la tabla de hechos demanda y pérdidas

Tabla 9. Formato archivo para la tabla de hechos demanda y pérdidas

|  |  |  |  |
| --- | --- | --- | --- |
| **Columna** | **Tipo datos** | **Descripción** | **Ejemplo** |
| Fecha | Fecha formato: YYYY-MM-DD | Fecha | 2012-01-01 |
| Periodo | Entero | Periodo | 1 |
| AgenteMemDisId | Texto (20) | Código MEM de agente distribuidor | ASTN |
| Mercado | Texto (50) | Tipo de mercado | NO REGULADO |
| PaisId | Texto (10) | Código país | Pai0001 |
| DepartamentoId | Texto (10) | Código departamento | Dep0016 |
| MunicipioId | Texto (10) | Código municipio | Mpi0610 |
| DemandaReal | Numérico | Demanda real en MWh | 1859.28 |
| PerdidaEnergia | Numérico | Pérdidas de energía MWh | 34.81 |

# QUÉ HACER EN CASO DE FALLOS

Cuando falla la ejecución de un paquete, revise la causa, solucione las inconsistencias, luego ejecute de nuevo el Job de carga diaria o realice una recarga parcial. Si el problema persiste consulte con el analista de sistemas. Algunas causas de fallos pueden ser:

## FALLOS AL EJECUTAR EL JOB FALLA

Revise la cuenta con la cual suben los servicios del Server Agent, es recomendable usar la misma cuenta de dominio que la utilizada para subir los servicios SQL Server. Revise el switche para correr paquetes en modo de 32-bits, puesto que algunas conexiones OLE-DB como Excel y Oracle no son soportadas en servidores de 64 bits. Revise la cuenta con la cual está conectado en SSMS cuando ejecuta el Job en forma manual, es posible que requiera permisos para ejecutar Jobs o que no haya modificado el password de los paquetes.

## FALLO POR CONEXIÓN DE RED

Si el paquete falla porque no encuentra la ruta origen o destino realice un ping para verificar conexión, si el ping falla consulte con plataforma.

## CLAVES DUPLICADAS

Si falla por claves duplicadas es posible que esté tratando de subir dos veces el mismo registro. Este tipo de errores es más complejo de solucionar, puesto que pueden ser causado por reglas de negocio o cambios en los sistemas fuente.

## FALLA AL PROCESAR LAS DIMENSIONES

La cuenta proxy está asociada a un login de Windows esta cuenta debe ser administradora del servidor Analysis Services o debe tener permisos para procesar objetos. Si el fallo no es por seguridad es probable que se presenten fallos al procesar dimensiones por cambios en datos configurados con relaciones tipo rígido, el paquete que procesa dimensiones está configurado para procesar en modo “Actualizar” para solucionar el problema procese de nuevo todas las dimensiones en forma “Completa” y luego ejecute el job desde el paso en el cual se presentó el fallo.

## FALLO AL PROCESAR EL CUBO

El error puede ser similar al anterior. Los cubos son procesados en modo Completo, es probable que el error sea solucionado al procesar en forma manual y luego ejecutar de nuevo el job en el paso que se presentó el fallo. El paquete tiene configurado la consulta de selección de datos, nombres de grupos de archivo, nombres de particiones, etc. Si cambia el estándar de nombres es probable que el proceso falle o que tenga datos duplicados. En este caso la solución puede requerir cambios en las consultas utilizadas en el procesamiento de cubos.