Triggers

Carga de Datos

Este proceso posterior se basa en la carga de datos la cual puede ser Full o Delta, en el primer caso se vuelcan todos los datos y en el segundo solamente los datos no almacenados en el Mart previamente.

Como una abordaje preliminar podrías preguntarte, ¿es eficiente cargar todos los registros históricos en cada ejecución del proceso?. Objetivamente es mejor una carga de datos Delta, es decir que solo vuelque las diferencias con respecto a la ultima carga, sin embargo depende de diferentes factores.

Carga de Datos

Orígenes de datos:
Determinar cuales han sido las últimas modificaciones con respecto a la última carga.

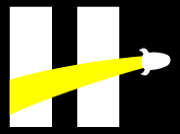
Volumen de datos:
Volumen de datos es pequeño la carga podría ser Full.
Volúmenes muy grande cargas Delta.

Velocidad de respuesta:
Para velocidades más lentas puede ser Full y rápidas Delta.

Niveles de Servicio:
Por el costo que supone la utilización de servicios en la nube.

Técnicas

A continuación se presentaran técnicas aplicables en la carga de datos a repositorios centralizados. Como un abordaje conceptual, una técnica define un marco para la realización de cierto proceso, pero la forma de implementarla (realizar el script), puede variar y ser distinta para cada profesional, con una mejor o menor performance.

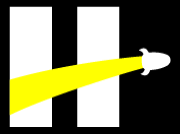


Extracciones basadas en fechas

Normalmente selecciona todas las filas donde se crearon o modificaron los campos de fecha, lo que significa muchas veces "todos los registros de ayer".

La selección puede cargar filas duplicadas y requerir intervención manual y limpieza de datos si el proceso falla por cualquier razón.

Mientras tanto, si el proceso de carga nocturna no se ejecuta y se salta un día, hay un riesgo de que los datos perdidos nunca lleguen al almacén de datos.



Comparación de diferencias completas

Mantiene una instantánea completa de los datos de ayer y los compara, registro por registro, contra los datos de hoy para encontrar qué cambió.

La buena noticia es que esta técnica es minuciosa: tiene la garantía de encontrar todos los cambios.

La mala noticia obvia es que, en muchos casos, esta técnica requiere muchos recursos.

Scrapping de registros de base de datos

El scrapping de registros, toma una instantánea de los registros de la base de datos en un momento determinado. Este es un punto en el tiempo (normalmente medianoche) y luego busca transacciones que afecten a las tablas de interés para la carga ETL. Esta es probablemente la más complicada de todas las técnicas.

Tabla de auditoria

La tabla de auditoría es una dimensión especial que se ensambla en el sistema ETL para cada tabla de hechos. La dimensión de auditoría contiene el contexto de metadatos en el momento en que se crea una fila específica de la tabla de hechos. Se podría decir se elevan metadatos a datos reales!

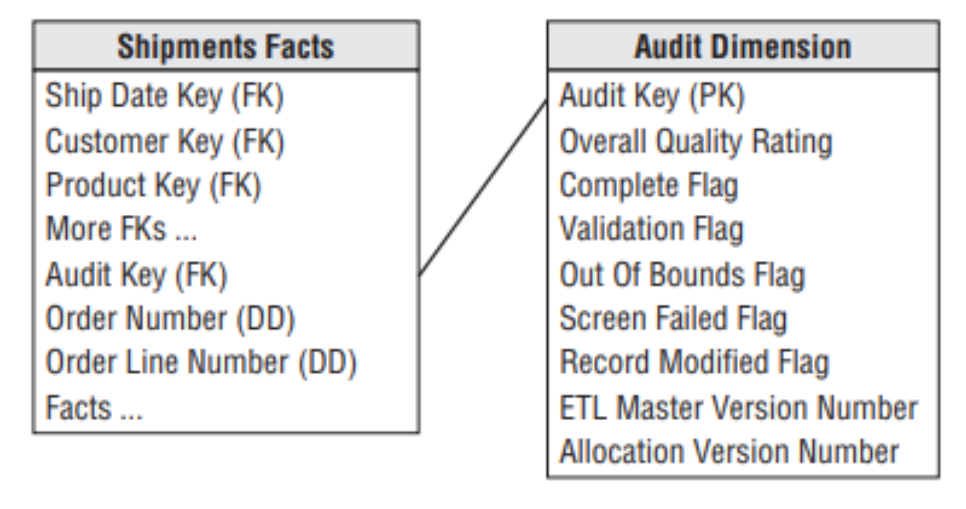
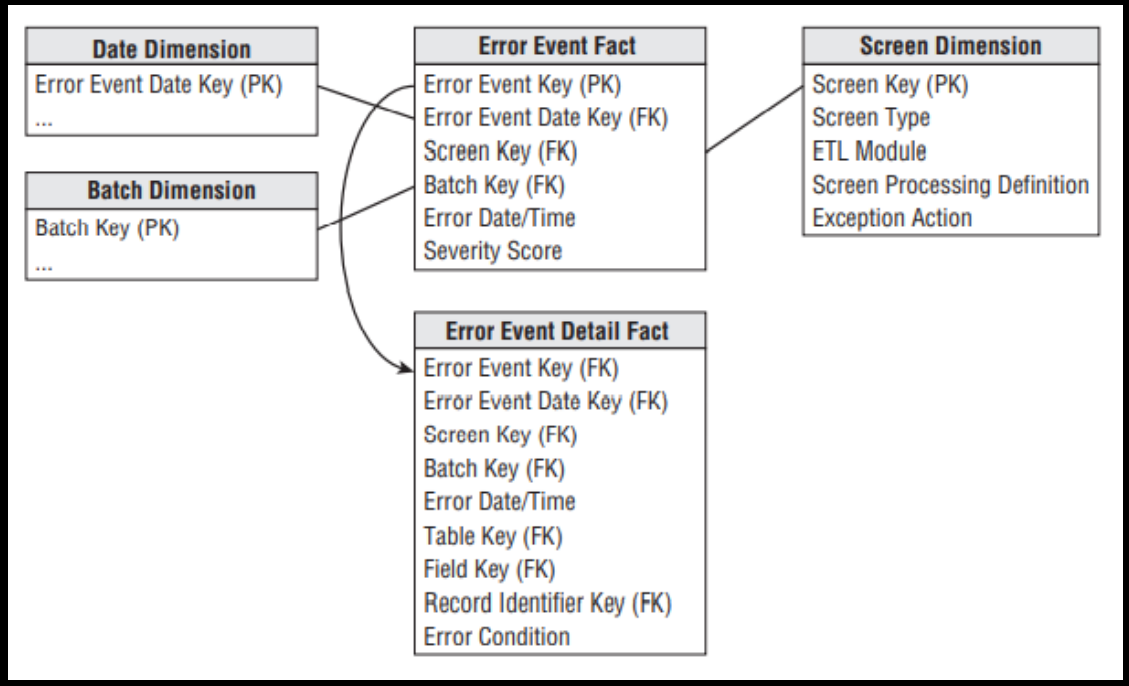
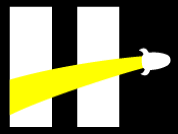


Tabla de errores

La tabla de errores es un esquema dimensional centralizado cuyo propósito es registrar cada evento de error lanzado por una pantalla de en cualquier lugar de la canalización de ETL.

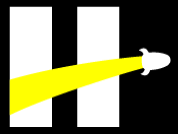




Triggers

Un disparador es un objeto con nombre dentro de una base de datos el cual se asocia con una tabla y se activa cuando ocurre en ésta un evento en particular.

La sentencia CREATE TRIGGER crea un disparador que se asocia con la tabla. También se incluyen cláusulas que especifican el momento de activación, el evento activador, y qué hacer luego de la activación.



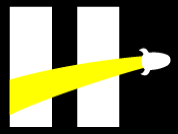
Triggers

BEFORE - AFTER indican el momento de acción del disparador.

INSERT, DELETE y UPDATE indican el evento que activará al disparador.

FOR EACH ROW, define lo que se ejecutará cada vez que el disparador se active, lo cual ocurre una vez por cada fila afectada por la sentencia activadora.

Las columnas de la tabla asociada con el disparador pueden referenciarse empleando los alias OLD - NEW. con el nombre de la columna. Esto refiere al valor que tenia ese campo.



Triggers

```
CREATE TABLE alumno (
cedulaIdentidad INT NOT NULL AUTO_INCREMENT,
nombre VARCHAR(20),
apellido VARCHAR(20),
fechaInicio DATE,
PRIMARY KEY (cedulaIdentidad)
)

CREATE alumnos_auditoria (
id_auditoria INT NOT NULL AUTO_INCREMENT,
cedulaIdentidad_auditoria INT,
nombre_auditoria VARCHAR(20),
apellido_auditoria VARCHAR(20),
fechaInicio_auditoria DATE,
usuario VARCHAR (20),
fecha DATE,
PRIMARY KEY (id_auditoria)
)

CREATE TRIGGER auditoria AFTER INSERT ON alumno
FOR EACH ROW
INSERT INTO alumnos_auditoria (cedulaIdentidad_auditoria, nombre_auditoria, apellido_auditoria, fechaInicio_auditoria, usuario, fecha)
VALUES (NEW.cedulaIdentidad_auditoria, NEW.nombre_auditoria, NEW.apellido_auditoria, NEW.fechaInicio_auditoria,CURRENT_USER,NOW())
```