

**HENRY**



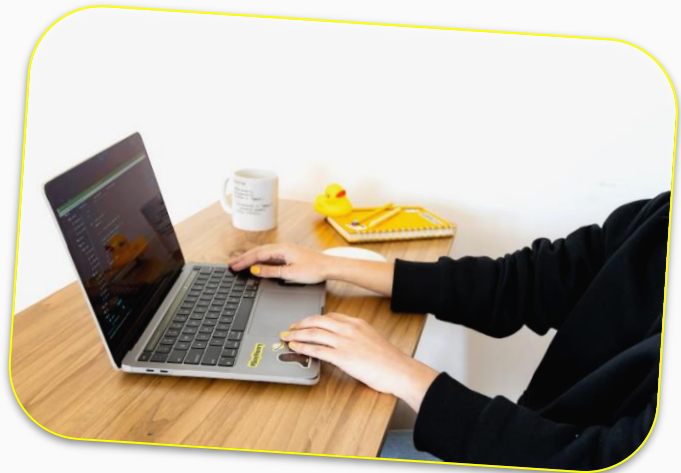
# Carga de Datos Triggers

Data Science





# Agenda



- **Carga de datos**
- **Técnicas para la carga de datos**
- **Tabla de auditoría**
- **Triggers**



# **OBJETIVOS DE LA CLASE**

*Al finalizar esta lecture estarás en la capacidad de...*

- **Inferir** la diferencia entre carga inicial y carga incremental
- **Entender** el concepto de Triggers

**Carga de datos**



# ¿Qué es?

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.





# Proceso para la carga de datos

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

**Volumen de datos:**

Si el volumen de datos es pequeño, la carga podría ser **Full**.  
Volúmenes muy grande las cargas son **Delta**.

**Velocidad de respuesta:**

En entornos donde la velocidad sea irrelevante, una carga Full sería válida y más fácil de desarrollar

**Niveles de servicio:**

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

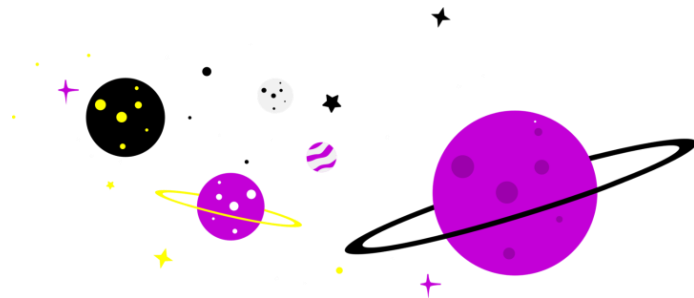


# **Técnicas para la carga de datos**



# 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"**.







# Extracciones basadas en fechas

LOADING ...



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.



# 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 auditoría

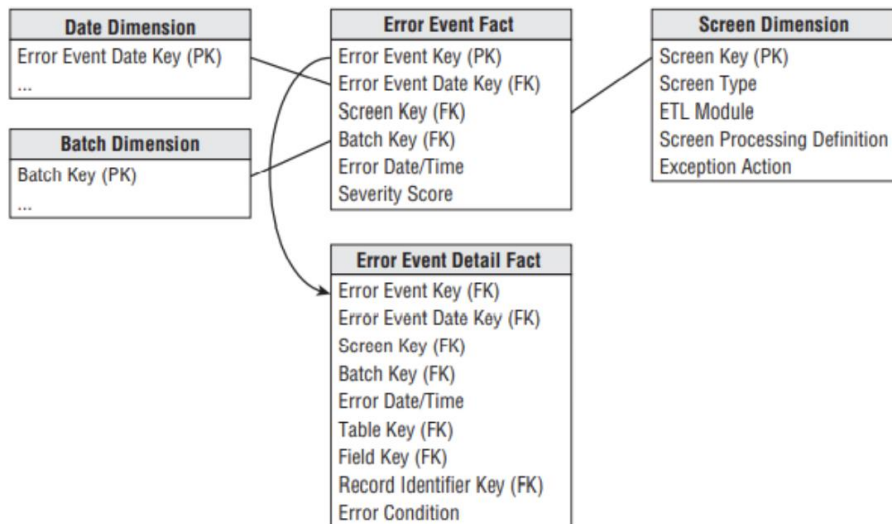
La **tabla de auditoría** es una dimensión especial que se ensambla en el sistema ETL para cada tabla de hechos. La dimensión de auditoría contiene el contexto de metadatos en el momento en que se crea una fila específica de la tabla de hechos. En resumen, se elevan **metadatos a datos reales**.

Shipments Facts
Ship Date Key (FK)
Customer Key (FK)
Product Key (FK)
More FKs ...
Audit Key (FK)
Order Number (DD)
Order Line Number (DD)
Facts ...

Audit Dimension
Audit Key (PK)
Overall Quality Rating
Complete Flag
Validation Flag
Out Of Bounds Flag
Screen Failed Flag
Record Modified Flag
ETL Master Version Number
Allocation Version Number



# 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



## ¿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**.





# ¿Triggers?



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.



**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 tenía ese campo.





```
CREATE TABLE alumno (  
cedulaIdentidad INT NOT NULL AUTO_INCREMENT,  
nombre VARCHAR(20),  
apellido VARCHAR(20),  
fechaInicio DATE,  
PRIMARY KEY (cedulaIdentidad)  
);  
  
CREATE TABLE alumno_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,  
        NEW.nombre,  
        NEW.apellido,  
        NEW.fechaInicio,  
        CURRENT_USER,  
        NOW());
```

# HENRY

