

L U M B R E

```
if hasattr(self, 'filepath'):
    self.filepath = self.filepath

def export(self):
    """Export the scene to a file"""
    path.dirname(self.filepath)
    # Export in the viewport
    bpy.context.selected_objects = []
    # Viewport selection
    on_setting = False:
    = [1 for i in bpy.context.scene.objects]
    # Deselect all
    bpy.ops.object.select_all(action='DESELECT')
    # Export list
    # MESH:
    use_path_join(folder_path, "{}.obj".format(item.name))
    use_filepath_file_path, use_selection=True,
    use_forward=self.axis_forward_setting,
    use_axis_up=self.axis_up_setting,
    use_animation=self.use_animation_setting,
    use_animation_modifiers=self.use_animation_modifiers_setting,
    use_mesh_modifiers=self.use_mesh_modifiers_setting,
    use_edges=self.use_edges_setting,
    use_smooth_groups=self.use_smooth_groups_setting,
    use_smooth_groups_bitflags=self.use_smooth_groups_bitflags_setting,
    use_normals=self.use_normals_setting,
    use_uv=self.use_uv_setting,
    use_material=self.use_material_setting,
```

Datos incrementales e idempotencia

Grado: 6

Materia: Ingeniería de Datos · Ingeniería de Datos

11/10/2026

Antes de empezar

Reprocesar todo desde cero es sencillo pero insostenible a escala. La carga incremental trae solo el delta, y la idempotencia garantiza que relanzarla por un fallo no doble ni pise datos: las dos cualidades que separan un script de un pipeline maduro.

QUÉ APRENDERÁS

- Diseñar cargas que procesan solo lo nuevo o modificado y que, al repetirse, no duplican ni corrompen, garantizando pipelines reproducibles y económicos.

Datos incrementales e idempotencia

Solo el delta, y sin duplicar

Procesar solo lo que cambió ahorra cómputo y tiempo; procesarlo sin duplicar aunque el job se relance a medio vuelo es lo que deja dormir al ingeniero. Juntas, ambas cualidades hacen el pipeline confiable y barato.

VIDEO

Captura de cambios y procesamiento incremental

Vídeo que explica cómo sincronizar destinos moviendo solo lo que cambió en el origen.



Escanea para verlo en Lumbre

<https://www.youtube.com/watch?v=PzPsmERP2Go>

Carga completa frente a incremental

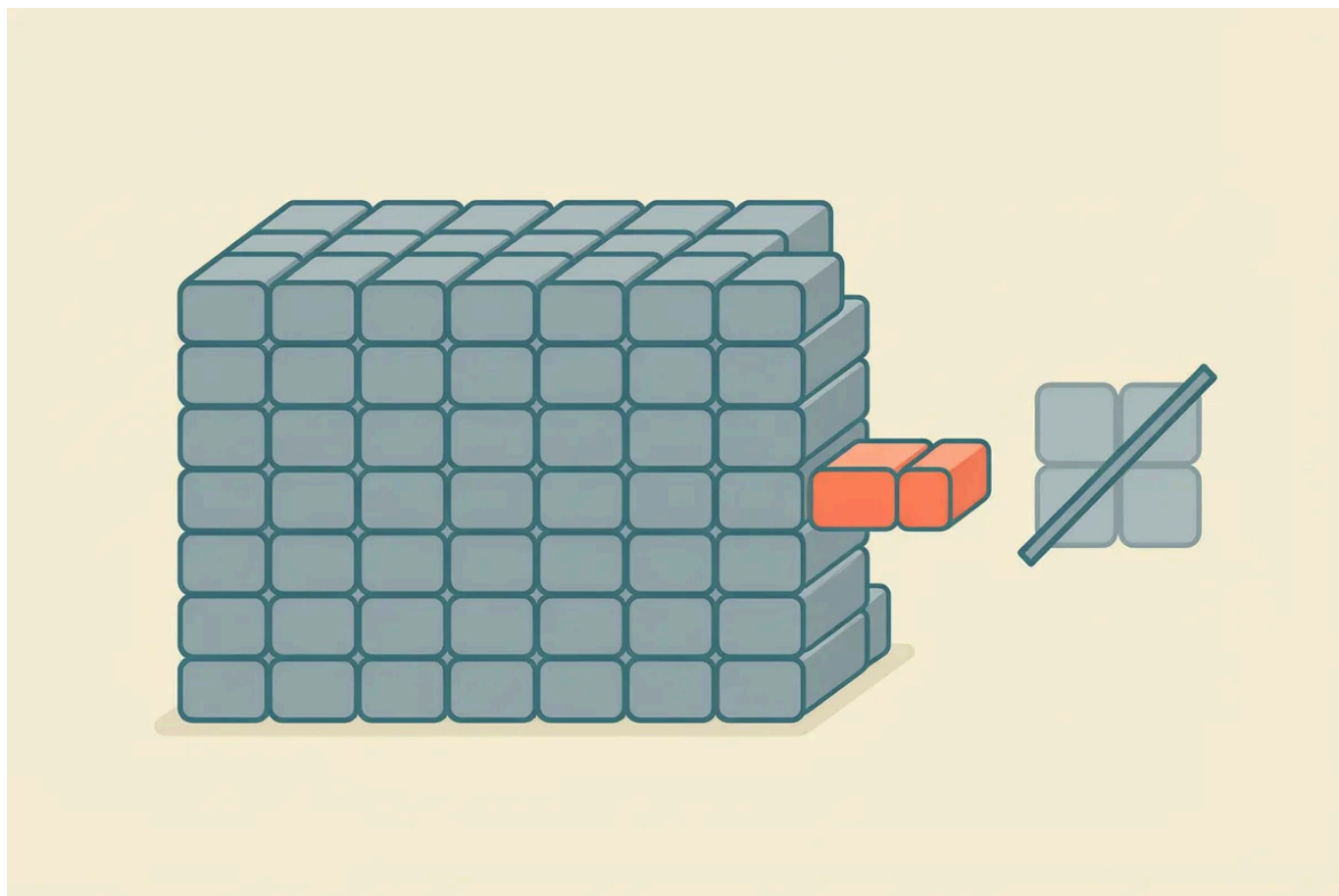
La carga completa vuelca todo cada vez: simple y auto-correctible, pero cara y lenta al crecer. La incremental trae solo las filas nuevas o modificadas, escalando con el cambio y no con el histórico.

Cómo detectar el delta

- Por marca temporal de creación o actualización en el origen.
- Por un número de versión o registro de cambios creciente.
- Por captura de cambios de datos que registra inserciones, modificaciones y borrados.

Borrados y la trampa

Lo incremental se olvida de los borrados si solo miras altas y cambios. Sin una señal de supresión, una fila eliminada en el origen sobrevive para siempre en el destino: el borrado lógico suele ser la salida honesta.



La carga incremental solo mueve el delta resaltado; la raya recuerda que repetir el lote no debe duplicar nada.

Qué es idempotencia

Una operación idempotente produce el mismo resultado aplicarla una vez que cien. En cargas, relanzar el mismo lote no debe duplicar ni degradar: el destino queda como si el job hubiera corrido una sola vez y bien.

Cómo lograrla

Se escribe por fusión sobre una clave natural, reemplazando particiones completas o con un merge que actualiza o inserta. La clave está en que cada fila tenga un identificador determinista que la haga única.

Reproducibilidad

Un pipeline reproducible entrega el mismo resultado si se relanza sobre los mismos insumos, aunque lleve días parado. Exige anclar versión de código, datos de referencia y no depender del reloj del ejecutor.

Coste del backfill

Corregir el pasado con cargas incrementales es delicado: hay que reprocesar las ventanas afectadas en orden y de forma idempotente, o acabarás con histórico inconsistente entre lo viejo y lo nuevo.

Errores frecuentes

- Reprocesar todo desde cero cada día y descubrir el coste cuando el histórico ya no cabe.
- Detectar solo altas y cambios, sin señal de supresión, y dejar filas borradas vivas en el destino.
- Relanzar el lote a medias sobre una escritura no idempotente y duplicar lo ya cargado.

Ejemplo resuelto: sincronizar ventas sin duplicar

Un job nocturno trae ventas de un ERP y a mitad de ventana falla a veces. El diseño seguro:

1. Definir una clave natural determinista que haga única cada fila de venta.
2. Detectar el delta por marca temporal y, además, por registro de cambios que capture borrados.
3. Escribir por fusión sobre la clave, o reemplazar la partición de la ventana completa.
4. Propagar los borrados con una señal de supresión y borrado lógico en el destino.
5. Verificar recuentos contra el origen en cada lote para detectar huecos temprano.

¿Para qué sirve en la realidad?

Toda empresa que alimenta un panel desde sistemas transaccionales necesita mover cambios, no historia completa. De ello dependen el coste de cómputo, la frescura del dato y dormir tranquilo sabiendo que un reintento no duplica ventas.

1. Un job incremental termina a medias por un fallo y vuelve a lanzarse sobre la misma ventana de fecha. La propiedad que evita que las filas de esa ventana se dupliquen es:

- ☐ la carga completa
- ☐ la idempotencia de la escritura
- ☐ la paginación de la API
- ☐ la compresión del archivo

2. Detectar el delta solo por filas nuevas basta para reflejar correctamente los borrados del origen.

- ☐ Verdadero
- ☐ Falso

ACTIVIDAD INTERACTIVA DISPONIBLE EN LUMBRE

3. Incremental



4. Ordena el diseño de una carga incremental fiable:

Elementos:

- definir una clave natural determinista por fila
- detectar qué filas cambiaron desde la última vez
- aplicar el delta por fusión idempotente
- contrastar recuentos contra el origen

Clave	. . .
Delta	. . .
Fusion	. . .
Verificacion	. . .

Idempotencia <https://en.wikipedia.org/wiki/Idempotence> Definición formal de idempotencia y su sentido en operaciones que se repiten.

5. Une cada necesidad de la carga con su técnica.

5.	traer solo lo nuevo o modificado	-----
5.	captar también los borrados	-----
5.	repetir el lote sin duplicar	-----
5.	corregir el pasado en orden	-----

Opciones: delta por marca temporal · captura de cambios · fusión sobre clave · backfill idempotente

Solo el delta, y sin duplicar

Incremental e idempotencia

Completa

simple y autocorrectiva

cara al crecer

Delta

marca temporal

registro de cambios

ojo con borrados

Idempotencia

fusión por clave

reemplazo de partición

Garantía

reproducibile

verificar recuentos

El mapa mental interactivo es interactivo en la versión online.



Replicación incremental con CDC <https://airbyte.com/tutorials/incremental-change-data-capture-cdc-replication> Guía práctica de replicación incremental y captura de cambios en conectores reales.

6. Tus ventas diarias corregían ayer una fila de hace tres meses. Una carga incremental por marca temporal, ¿la recogería? ¿Qué necesitas añadir al diseño para no perder correcciones tardías?

Referencia curricular

Grado:	6
Materia:	Ingeniería de Datos · Ingeniería de Datos

Versión interactiva completa (con actividades, videos y autoevaluación):



<https://univ.lumbre.lat/object/cdia-ingd-004-datos-incrementales-e-idempotencia>

Material educativo generado desde Lumbre · lumbre.lat