Incremental

Spark's append mode:

- Will *change* the destination json file and add new rows. (Any changes made on a row will insert a new row)
- If exising rows that were in the destination file before are not in the new dataset to save, they will be removed / wont be in the destination file.

ETL-lib's incremental mode:

- Behavior is slightly different if reading from the raw zone or the curated zone.
- Detects changes in nested structures
- Deleted rows and incremental: A deleted row in the source side simply wont be written in the sink side. The recommended pattern is to have some sort of *is_deleted* field with the consumer handling those cases as they see fit.
- Reading from the curated zone:
    - Incremental**False**: Will overwrite the destination zone with the full extract of available data in the curated delta table.
    - Incremental **True**: Will overwrite the destination zone with only rows from the curated delta table that where either *updated* or *inserted* since the last batch.
    - Applies to: ETL.*curated.read* and ETL.*trusted.write* when passed the name of a table but not when passed a dataframe.
- Reading from the raw zone:
    - Source
        - Incremental **False**: Will take data from all files and folder
        - Incremental **True**: Will take new data only. It does this by checking the control_table.
    - Destination:
        - When using ETL.curated.*write*: Its an overwrite. The curated zone will now be populated with *only* this extract and nothing else.
        - When using ETL.curated.*merge*: Its an update. The curated zone with update existing rows from the incoming dataset only while leaving any other ones unchanged.