# 2. Morpheus Prebuilt Filters: 내장 필터 및 결측치 처리

## 프로젝트 개요

Morpheus가 제공하는 내장 스테이지를 활용하여 데이터 전처리 작업을 수행했습니다. 특히, FilterStage와 결측치(Nulls) 처리 기능을 사용하여 데이터 품질을 향상시키고, 특정 조건에 맞는 메시지만을 선별적으로 통과시키는 방법을 실습했습니다.

## ⚙️ 사용 기술 및 역할

* **주요 기술:** Morpheus FilterStage, DropNullColumnsStage (파일 내용 기반 추론)
* **핵심 역할:**
  1. **결측치 처리:** 데이터셋에서 Null 값이 너무 많은 컬럼을 자동으로 제거하는 로직을 파이프라인에 통합 (노트북 파일명 추론).
  2. **조건부 필터링:** FilterStage를 사용하여 데이터프레임(페이로드)의 내용에 따라 메시지를 통과시키거나 드롭(Drop)하는 조건부 로직을 구현했습니다.

## 🛠️ 주요 구현 내용 (Exercise-drop-nulls 기반)

1. **데이터 클렌징:** 로그 데이터에서 error\_message 컬럼의 결측치(Null)를 확인하고, Null이 아닌 유효한 에러 메시지를 가진 행만 필터링하는 로직을 적용했습니다.
2. **Filter Function:** FilterStage는 사용자 정의 함수(filter\_fn)를 받아 메시지 페이로드의 cuDF 데이터를 검사하도록 구성했습니다.
   * **예시 로직:** error\_message 컬럼이 Null이 아닌 경우(not message.get\_data()['error\_message'].isnull().all())에만 메시지를 통과시킵니다.
3. **파이프라인 통합:** FilterStage를 DeserializeStage와 InMemorySinkStage 사이에 배치하여, 데이터 흐름 중간에 실시간으로 데이터 품질 관리 및 선별 작업이 이루어지도록 했습니다.

## 📌 주요 코드 로직 (추상화)

from morpheus.stages.preprocess.filter\_stage import FilterStage  
  
# 필터링 로직 정의: 'error\_message'가 존재하는 행만 남김  
def filter\_errors(message: MessageMeta) -> bool:  
 # MessageMeta의 페이로드(DataFrame)에 접근하여 조건 확인  
 df = message.get\_data()  
 # 'error\_message' 컬럼에 Null이 아닌 값이 하나라도 있는지 확인  
 return not df['error\_message'].isnull().all()  
  
# 파이프라인에 FilterStage 추가  
pipeline.add\_stage(  
 FilterStage(  
 config,   
 filter\_fn=filter\_errors # 정의한 함수를 필터로 적용  
 )  
)