Como aplicar **idempotência nos pipelines de dados**

- O que é idempotência?
- Qual a importância em pipelines de dados?
- Aplicação em um caso real

O que é **idempotência?**

Em pipelines de dados, a idempotência é um conceito chave para garantir que a **execução de uma operação repetida não cause efeitos indesejados**.

Ou seja, ao processar dados várias vezes, o resultado final será o mesmo, independentemente de quantas vezes a operação seja realizada.

idempotência é importante porque...

- Evita **duplicação de dados**.
- Garante a **consistência do processo**.
- **Reduz o risco de erros**, especialmente em **ambientes distribuídos** ou quando ocorrem falhas e **reinícios de processos**.

Caso real

Como apliquei idempotência em um pipeline de dados

Cenário:

Fui alocado em um projeto em andamento que tinha como objetivo consumir dados de API’s para desenvolver dashboards analíticos.

Existiam vários desafios, principalmente em relação a performance de execução e consistência das informações. Além disso, caso ocorresse uma falha no consumo, os dados eram perdidos.

Resolvi esses problemas com algumas técnicas e metodologias, entre elas a **idempotência**.

Caso real

Como apliquei idempotência em um pipeline de dados

1º Passo: Defini uma chave primária adequada

Realizei análises e identifiquei que a chave primária não atendia todas as necessidades do negócio.
Foi necessário criar uma nova, pois a inconsistência da chave primária criava dados duplicados durante o processo de inserção dos dados.

Caso real

Como apliquei idempotência em um pipeline de dados

2º Passo: Criei mecanismos de controles

Defini funções que, durante o pipeline de dados, geravam logs e mantinham o status do processo executado, guardando informações da chave primária para recuperação de dados (seja via banco de dados ou endpoint da API).

Caso real

Como apliquei idempotência em um pipeline de dados

3º Passo: Ajustei o método de consumo do endpoint

Criei um método para tornar o ‘schema’ dos dados flexíveis, evitando uma série de validações das informações recebidas do consumo da API. Dessa forma, foram mantidos os dados padronizados e necessários para as análises dos dashboards.

Com esses ajustes conseguimos...

- **Reduzir significativamente o tempo de execução do pipeline: de 6hrs para 1 hr em cargas completas e cerca de 3 minutos para cargas incrementais** (considerando uma taxa média de 8.5 mi de registros)

- **Resolver o problema de inconsistência dos dados**

Você tem alguma pergunta?

Envie para mim que estarei a disposição para ajudar!

NATHÃ CORREIA
SR. DATA ENGINEER