DataOps.onETL

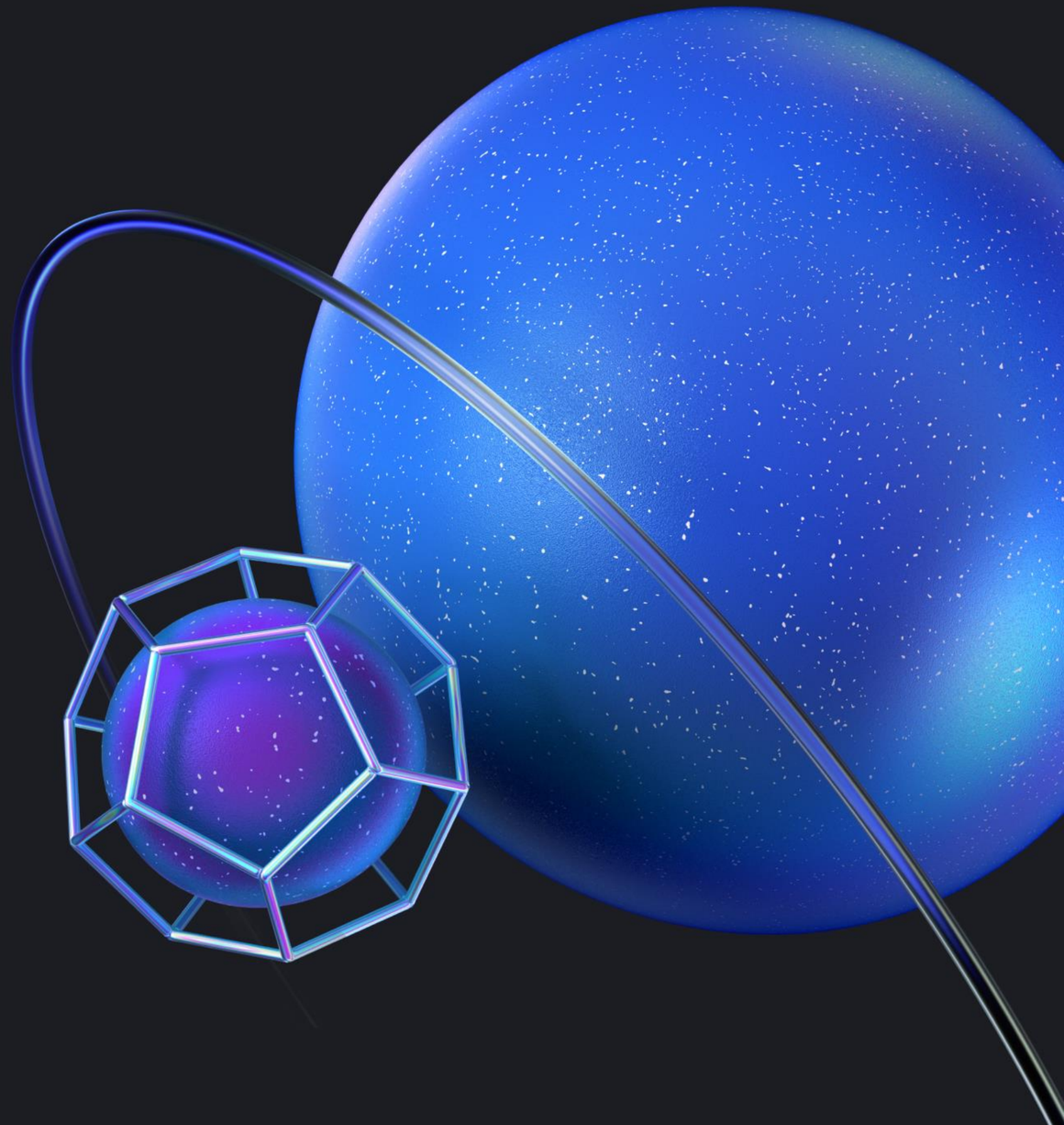
Unit#10

HWM & HWM Store.
Snapshot strategy & Incremental strategy.
Batch strategies

Саттар Гюльмамедов

Product Owner команды ETL

МТС Тета x DataOps Platform



onETL урок # 10

- HWM и HWMStore
- как используется HWM
- что такое Snapshot Strategy и Batch Snapshot Strategy
- что такое Incremental Strategy и Batch Incremental Strategy

МТС Тета x DataOps Platform

Объекты манипуляции данными

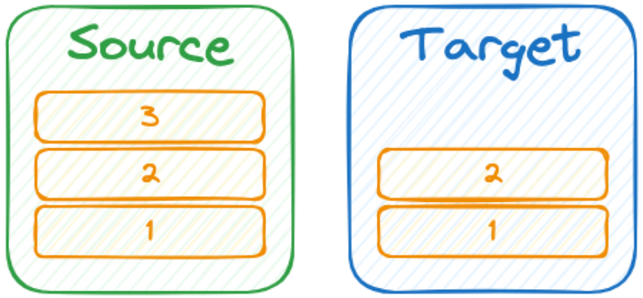
- YAMLHWMStore()
- SnapshotStrategy
- IncrementalStrategy
- SnapshotBatchStrategy
- IncrementalBatchStrategy

HWM

Стратегии чтения

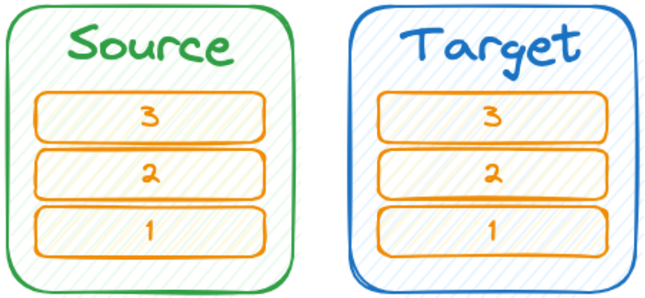
HWM

До:



hwm.value = 2

После:



hwm.value = 3

HWMStore

- YAMLHWMStore
- Horizon через etl-entities

YamlHWMStore

etl-entities

Horizon

YAMLHWMStore

Параметры

параметр	объяснение
path	Путь на локальной файловой системе, по которому будут хранится данные hwm
encoding	кодировка файла, по умолчанию "UTF-8"

Методы

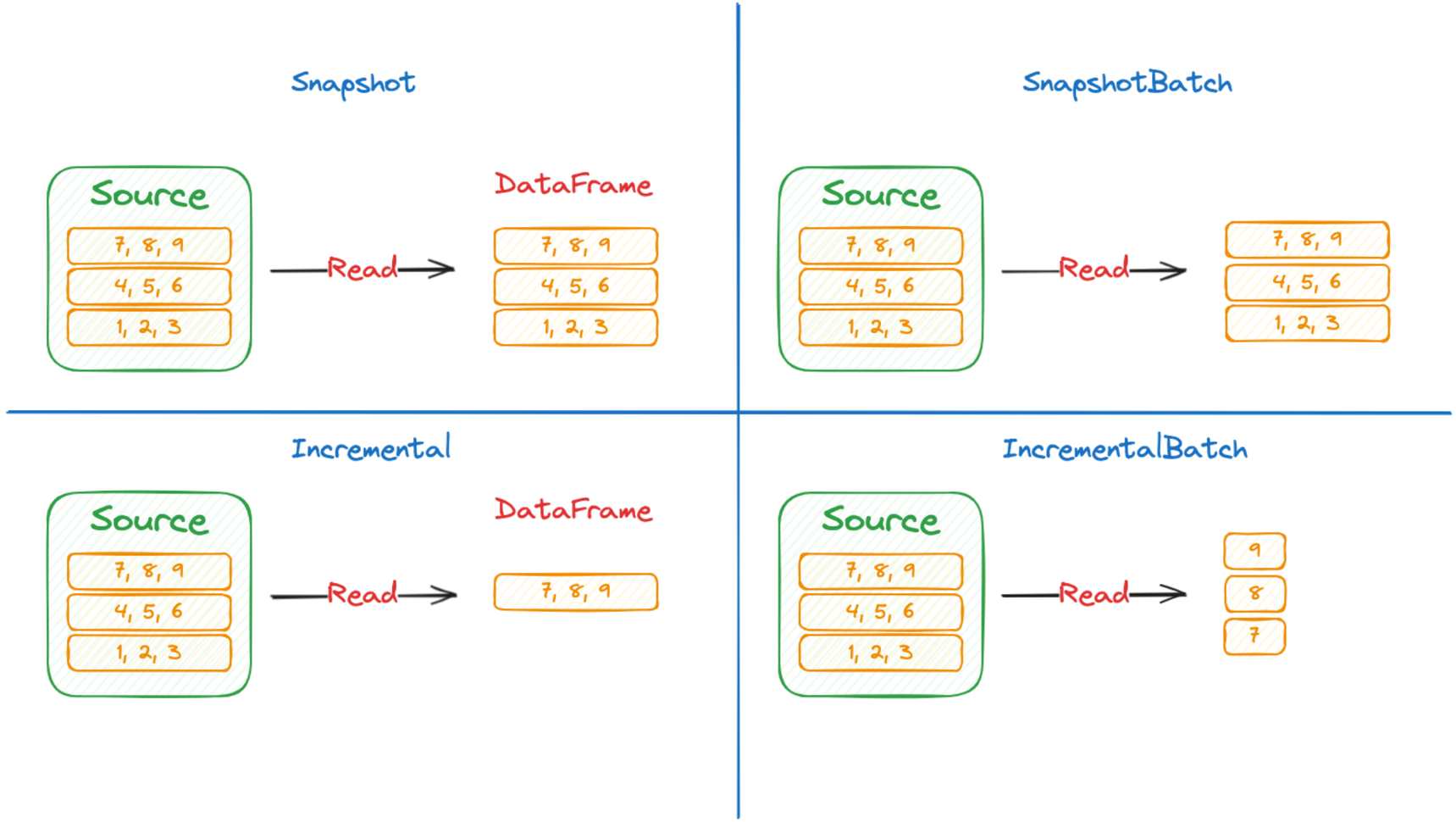
параметр	объяснение
get_hwm()	Получить текущее значение заданного hwm
set_hwm()	Установить значение заданного hwm

пример YAMLHWMStore

Python

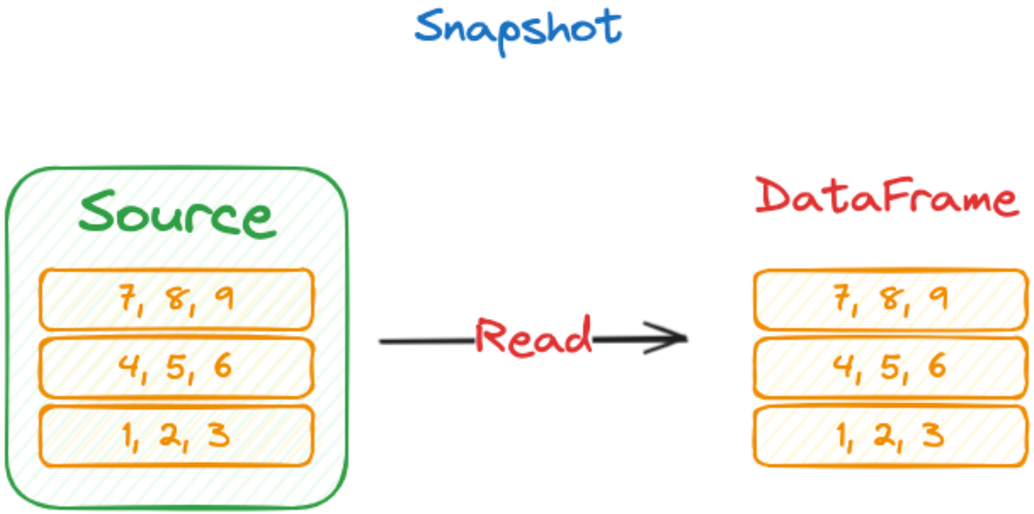
```python
from onetl.connection import Hive, Postgres
from onetl.db import DBReader
from onetl.strategy import IncrementalStrategy
from onetl.hwm.store import YAMLHWMStore

from pyspark.sql import SparkSession

maven_packages = Postgres.get_packages()
spark = (
    SparkSession.builder.appName("spark-app-name")
    .config("spark.jars.packages", ",".join(maven_packages))
    .getOrCreate()
)

postgres = Postgres(
    host="postgres.domain.com",
    user="myuser",
    password="*****",
    database="target_database",
    spark=spark,
)

hive = Hive(cluster="rnd-dwh", spark=spark)

reader = DBReader(
    connection=postgres,
    source="public.mydata",
    columns=["id", "data"],
    hwm=DBReader.AutoDetectHWM(name="some_unique_name", expression="id"),
)

writer = DBWriter(connection=hive, target="newtable")

with YAMLHWMStore():
    with IncrementalStrategy():
        df = reader.run()
        writer.run(df)
```

Стратегии чтения



Snapshot

- YAMLHWMStore
- Horizon через etl-entities



Стратегия снэпшотов

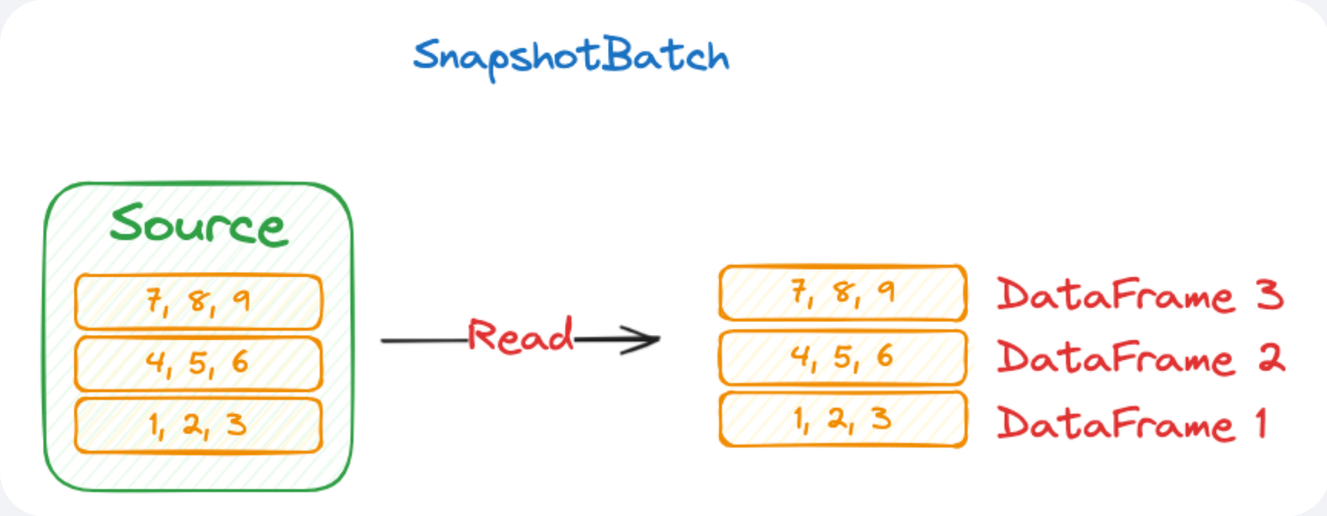
МТС Тета x DataOps Platform

Snapshot пример

```python
from onetl.connection import SFTP
from onetl.file import FileDownloader
from onetl.strategy import SnapshotStrategy

sftp = SFTP(
    host="sftp.domain.com",
    user="user",
    password="*****",
)

downloader = FileDownloader(
    connection=sftp,
    source_path="/remote",
    local_path="/local",
)

with SnapshotStrategy():
    df = downloader.run()
```

SnapshotBatch

параметр	объяснение
step	размер шага, интервал между значениями hwm, который определяет размер пакета
stop	максимальное значение hwm, которое применяется для ограничения общего объема извлекаемых данных
start	начальное значение, с которого нужно извлекать данные



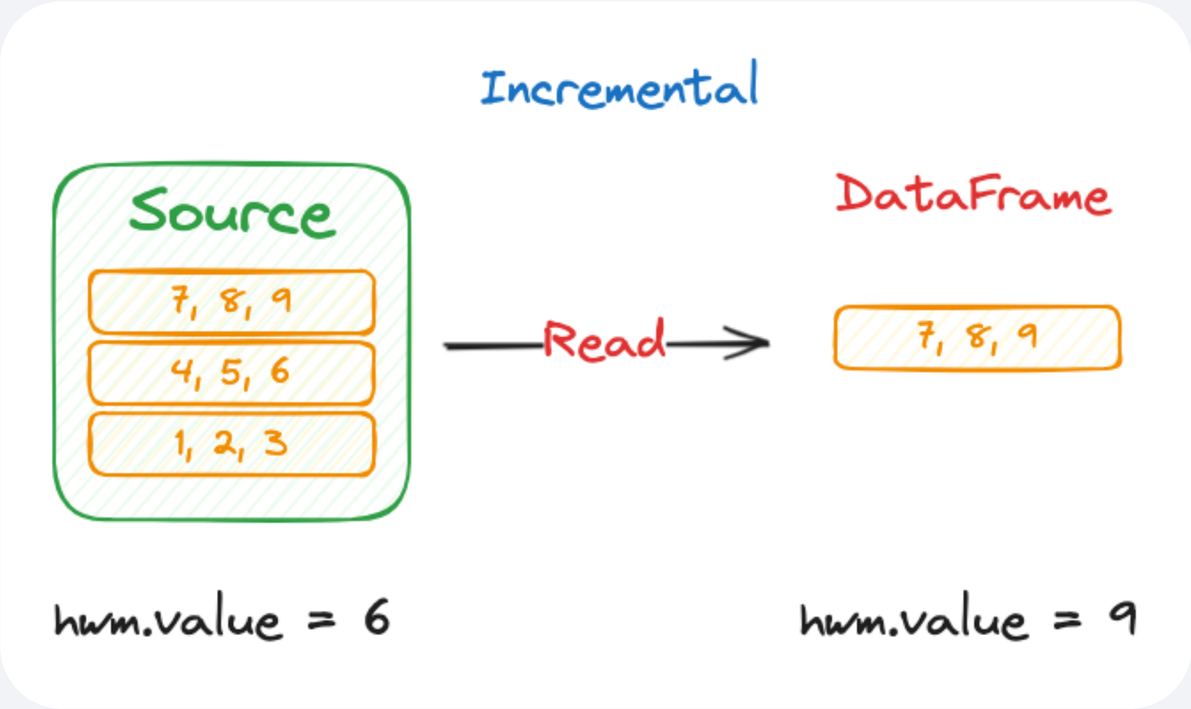
SnapshotBatch - пример

Python

```python
from onetl.connection import Postgres, Hive
from onetl.db import DBReader
from onetl.strategy import SnapshotBatchStrategy

from pyspark.sql import SparkSession

maven_packages = Postgres.get_packages()
spark = (
    SparkSession.builder.appName("spark-app-name")
    .config("spark.jars.packages", ",".join(maven_packages))
    .getOrCreate()
)

postgres = Postgres(
    host="postgres.domain.com",
    user="myuser",
    password="*****",
    database="target_database",
    spark=spark,
)

hive = Hive(cluster="rnd-dwh", spark=spark)

reader = DBReader(
    connection=postgres,
    source="public.mydata",
    columns=["id", "data"],
    hwm=DBReader.AutoDetectHWM(name="some_hwm_name", expression="id"),
)

writer = DBWriter(connection=hive, target="newtable")

with SnapshotBatchStrategy(step=100) as batches:
    for _ in batches:
        df = reader.run()
        writer.run(df)
```

Incremental

параметр	объяснение
offset	отступ от начального значения hwm (применяется в случае, если после получения последнего значения hwm были добавлены новые записи)



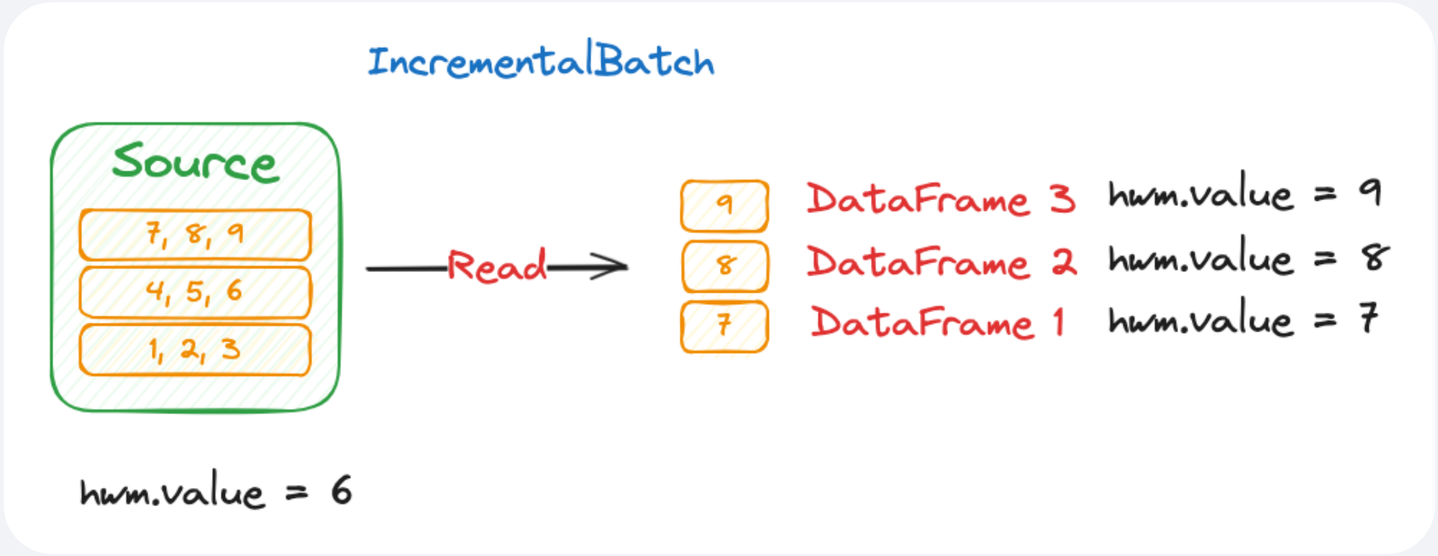
Incremental - пример

Python

```python
from onetl.connection import Postgres
from onetl.db import DBReader
from onetl.strategy import IncrementalStrategy

from pyspark.sql import SparkSession

maven_packages = Postgres.get_packages()
spark = (
    SparkSession.builder.appName("spark-app-name")
    .config("spark.jars.packages", ",".join(maven_packages))
    .getOrCreate()
)

postgres = Postgres(
    host="postgres.domain.com",
    user="myuser",
    password="*****",
    database="target_database",
    spark=spark,
)

reader = DBReader(
    connection=postgres,
    source="public.mydata",
    columns=["id", "data"],
    hwm=DBReader.AutoDetectHWM(name="some_hwm_name", expression="id"),
)

writer = DBWriter(connection=hive, target="newtable")

with IncrementalStrategy():
    df = reader.run()
    writer.run(df)
```

IncrementalBatch

параметр	объяснение
step	размер шага, интервал между значениями hwm, который определяет размер пакета
stop	максимальное значение hwm, которое применяется для ограничения общего объема извлекаемых данных
offset	отступ от начального значения hwm (применяется в случае, если после получения последнего значения hwm были добавлены новые записи)



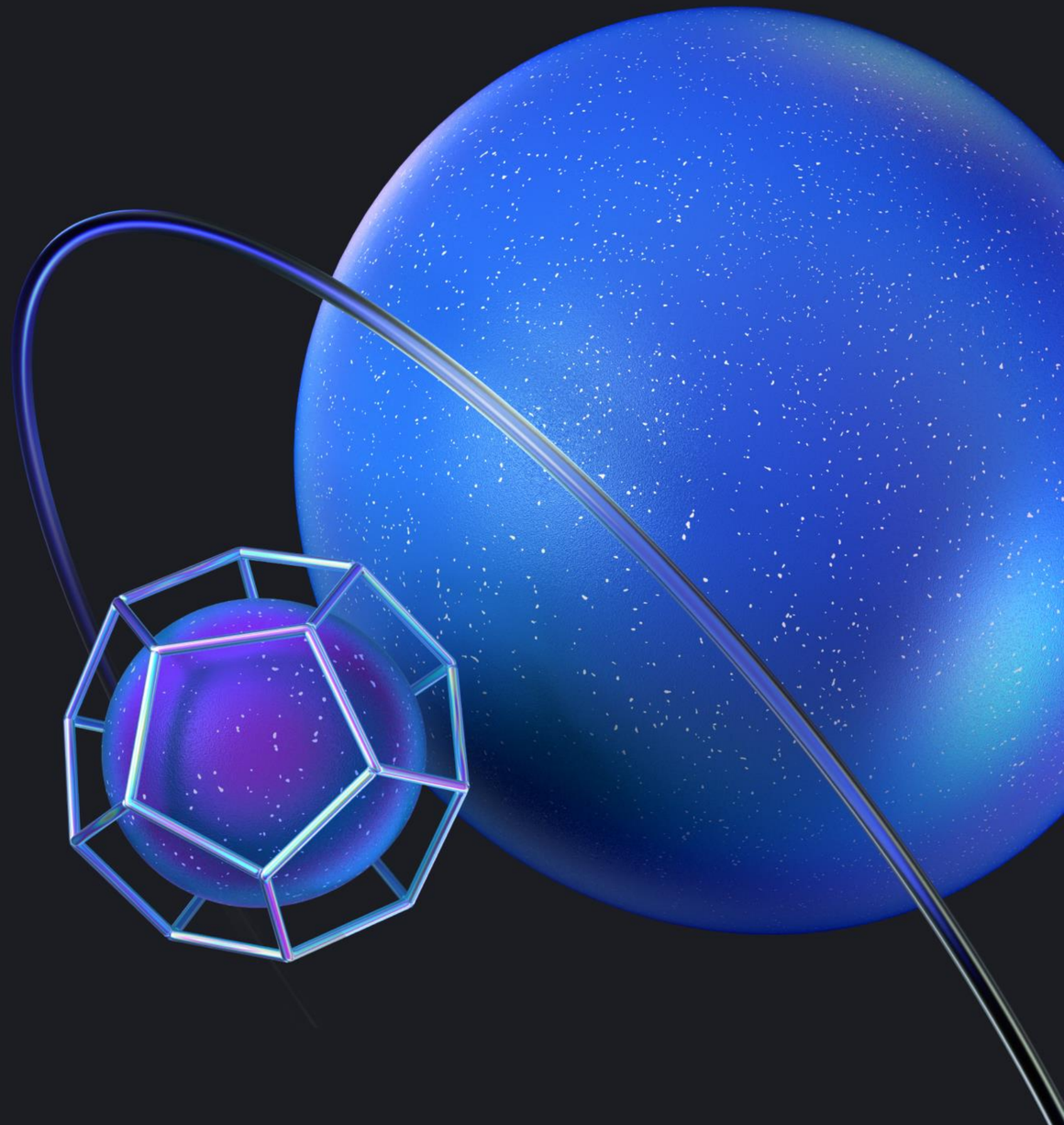
МТС Тета x DataOps Platform

IncrementalBatch - пример

Python

```python
from onetl.connection import Postgres, Hive
from onetl.db import DBReader
from onetl.strategy import IncrementalBatchStrategy

from pyspark.sql import SparkSession

maven_packages = Postgres.get_packages()
spark = (
    SparkSession.builder.appName("spark-app-name")
    .config("spark.jars.packages", ",".join(maven_packages))
    .getOrCreate()
)

postgres = Postgres(
    host="postgres.domain.com",
    user="myuser",
    password="*****",
    database="target_database",
    spark=spark,
)

hive = Hive(cluster="rnd-dwh", spark=spark)

reader = DBReader(
    connection=postgres,
    source="public.mydata",
    columns=["id", "data"],
    hwm=DBReader.AutoDetectHWM(name="some_hwm_name", expression="id"),
)

writer = DBWriter(connection=hive, target="newtable")

with IncrementalBatchStrategy(step=100) as batches:
    for _ in batches:
        df = reader.run()
        writer.run(df)
```

Спасибо!



onetools@mts.ru

https://t.me/c/1511728757/5

МТС Тета x DataOps Platform