Dataframe transformations

The Transform class serves the purpose of making dataframe transformation operations pluggable into the library's standard functions, making clear which and when transformation occurs.

Instantiate a Transform object with a function that takes a dataframe and returns a dataframe.

```
from ETL import Config, Transform, curated


config = Config(params)

def parse_date(df):
  return (
    df
      .withColumn('created_at', to_timestamp(col('created_at').cast(TimestampType()), "yyyy-mm-dd'T'HH:mm:ss"))
      .distinct()
  )

parse_date_transform = Transform(parse_date)
curated.write(config, table='Users', transformation=parse_date_transform)
```

If the transformation function uses more parameters, passed them as keyword arguments when instantiating the Transform object:

```
def parse_date(df, config):
  ...

parse_date_transform = Transform(parse_date, config=config)
```