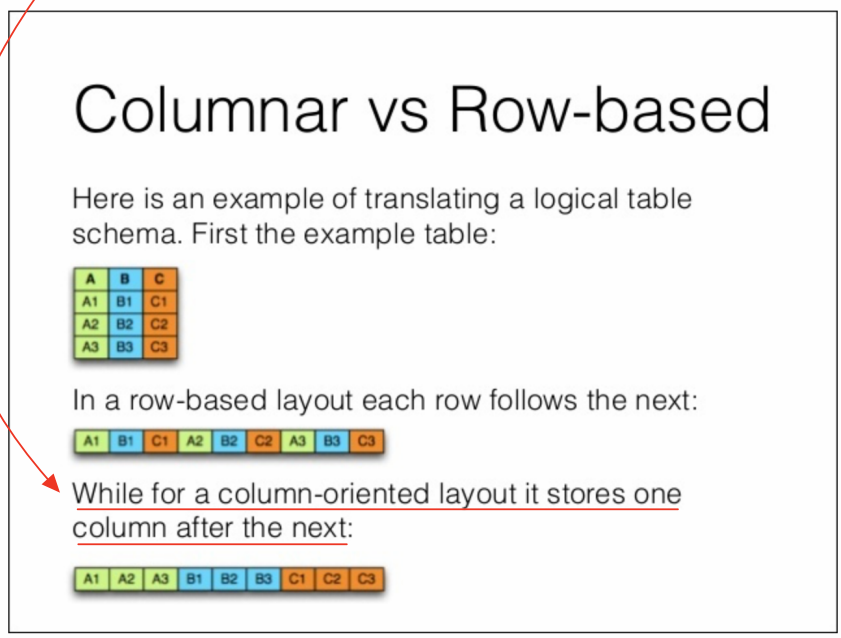
Arrows는 Apache에서 만든 Python 도구로 Multi Thread로 파일을 읽어들인다고 한다. 상식적으로 여러 Thread에서 I/O를 병렬적으로 진행하면 오히려 느려질 것 같은데, 아무튼 빠르다. Columnar 저장 방식을 활용하고 기본 포맷으로 Parquet를 사용한다. 여러 포스팅들에서 다루고 있지만, row-based 방식에 비해 columnar는 컴퓨터가 다루기 쉬운 형태로 저장한다.



Parquet는 사실상의 빅데이터 표준으로 자리 잡았고, 속도와 압축률이 무척 뛰어나다. 간단히, Arrow와 Pandas의 읽기 성능을 비교해보면 아래와 같다. 둘다 데이터 분석에 활용하기 위한 Dataframe 형식으로 Load하였고, Arrow는 Parquet형식으로 읽어 dataframe으로 변환하지만, pandas와 arrow는 메모리를 공유하는 형식으로 합의(?)되어 있기 때문에 속도 저하는 없다고. Data는 아래 사이트의 '5m Sales Record'로 압축풀면 600MB 크기의 텍스트 파일이다.
eforexcel.com/wp/downloads-18-sample-csv-files-data-sets-for-testing-sales/

```
import time
from pyarrow import csv
import pandas as pd

# Apache Arrow
start = time.time()
pyarrow_table = csv.read_csv('my_data2.csv')
df = pyarrow_table.to_pandas()
duration = time.time() - start
duration

-> 1.995872974395752

# Pandas
start = time.time()
pd.read_csv('my_data2.csv')
duration = time.time() - start
duration

-> 5.622146129608154
```

별다른 조작없이도 거의 1/3 수준으로 읽기 속도가 줄어들었다. 속도는 빨라지지만 메모리는 동일하기 소모하는데, 옵션으로 읽어들일 컬럼의 타입을 지정하면 일부 줄어들 수 있다고 한다. 저장에 있어서는 pyarrow는 parquet 형식만 지원하고 있는데, csv에 비해서는 월등히 높은 저장 속도를 보인다. 아래 블로그 참고하면 테스트 결과를 알 수 있다.