In [1]:

```
!pip install pyarrow
```

```
Requirement already satisfied: pyarrow in /anaconda/envs/py37_default/lib/python3.7/
site-packages (4.0.0)
Requirement already satisfied: numpy>=1.16.6 in /anaconda/envs/py37_default/lib/pyth
on3.7/site-packages (from pyarrow) (1.19.2)
```

In [2]:

```
#Pandas가있는 Apache Arrow (로컬 파일 시스템)
#Pandas 데이터 프레임을 Apache Arrow Table로 변환
import numpy as np
import pandas as pd
import pyarrow as pa

df = pd.DataFrame({'one': [20, np.nan, 2.5],'two': ['january', 'february', 'march'],'three': [True,
table = pa.Table.from_pandas(df)
```

In [3]:

```
df
```

Out[3]:

	one	two	three
a	20.0	january	True
b	NaN	february	False
c	2.5	march	True

In [4]:

```
table
```

Out[4]:

```
pyarrow.Table
one: double
two: string
three: bool
__index_level_0__: string
```

In [5]:

```
df_new = table.to_pandas()
```

In [6]:

```
## pandas로 변경
from pyarrow import csv

fn = 'usedcars.csv' # 파일 이름
table = csv.read_csv(fn) # 파일 불러오기 변수
df = table.to_pandas() # pandas로 불러오기
```

In [7]:

```
df.head()
```

Out[7]:

	Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
0	7450.0	65.0	82000.0	Petrol	86	1	0	1300	3	1015
1	7250.0	74.0	130025.0	Petrol	110	1	0	1600	3	1050
2	8950.0	80.0	64000.0	Petrol	110	0	0	1600	3	1055
3	11450.0	54.0	62987.0	Petrol	110	0	0	1600	5	1080
4	NaN	42.0	38932.0	Petrol	110	1	0	1600	3	1040

In [8]:

```
## parquet 파일로 저장
import pyarrow.parquet as pq

pq.write_table(table, 'usedcars.parquet')
```

In [9]:

```
table2 = pq.read_table('usedcars.parquet')
table2
```

Out[9]:

```
pyarrow.Table
Price: double
Age: double
KM: double
FuelType: string
HP: int64
MetColor: int64
Automatic: int64
CC: int64
Doors: int64
Weight: int64
```

In [10]:

```
table2 = pq.read_table('usedcars.parquet',columns=['Age','Price'])
```

In [15]:

```
dataset = pq.ParquetDataset('usedcars.parquet')
table = dataset.read()
table
```

Out[15]:

```
pyarrow.Table
Price: double
Age: double
KM: double
FuelType: string
HP: int64
MetColor: int64
Automatic: int64
CC: int64
Doors: int64
Weight: int64
```

In [16]:

```
pdf = table.to_pandas()
```

In [17]:

```
pdf.head()
```

Out[17]:

	Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
0	7450.0	65.0	82000.0	Petrol	86	1	0	1300	3	1015
1	7250.0	74.0	130025.0	Petrol	110	1	0	1600	3	1050
2	8950.0	80.0	64000.0	Petrol	110	0	0	1600	3	1055
3	11450.0	54.0	62987.0	Petrol	110	0	0	1600	5	1080
4	NaN	42.0	38932.0	Petrol	110	1	0	1600	3	1040

In [18]:

```
pdf = pq.read_pandas('usedcars.parquet', columns=['Price']).to_pandas()
pdf
```

Out[18]:

	Price
0	7450.0
1	7250.0
2	8950.0
3	11450.0
4	NaN
...	...
1441	8750.0
1442	9950.0
1443	9250.0
1444	12450.0
1445	8500.0

1446 rows × 1 columns

In [19]:

```
## parquet 파일을 pandas로 변경하기
table = pa.Table.from_pandas(pdf, preserve_index=False)
pq.write_table(table, 'usedcars_noindex.parquet')
t = pq.read_table('usedcars_noindex.parquet')
t.to_pandas()
```

Out[19]:

	Price
0	7450.0
1	7250.0
2	8950.0
3	11450.0
4	NaN
...	...
1441	8750.0
1442	9950.0
1443	9250.0
1444	12450.0
1445	8500.0

1446 rows × 1 columns

In [20]:

```
parquet_file = pq.ParquetFile('usedcars.parquet')
parquet_file.metadata
```

Out[20]:

```
<pyarrow._parquet.FileMetaData object at 0x7f91c144ebf0>
  created_by: parquet-cpp-arrow version 4.0.0
  num_columns: 10
  num_rows: 1446
  num_row_groups: 1
  format_version: 1.0
  serialized_size: 2010
```

In [21]:

```
parquet_file.schema
```

Out[21]:

```
<pyarrow._parquet.ParquetSchema object at 0x7f91c1453f00>
required group field_id=0 schema {
  optional double field_id=1 Price;
  optional double field_id=2 Age;
  optional double field_id=3 KM;
  optional binary field_id=4 FuelType (String);
  optional int64 field_id=5 HP;
  optional int64 field_id=6 MetColor;
  optional int64 field_id=7 Automatic;
  optional int64 field_id=8 CC;
  optional int64 field_id=9 Doors;
  optional int64 field_id=10 Weight;
}
```

In [22]:

```
#Pandas는 나노초를 사용하므로 호환성을 위해 밀리 초 단위로자를 수 있습니다.
#pq.write_table(table, 'usedcars.ms1.csv',coerce_timestamps='ms')
#pq.write_table(table, 'usecars.ms2.csv',coerce_timestamps='ms', allow_truncated_timestamps=True)
```

In [23]:

```
#compression
from pyarrow import csv

fn = 'usedcars.csv'
table = csv.read_csv(fn)
df = table.to_pandas()
```

In [24]:

```
#기본적으로 Apache arrow는 다른 코덱도 허용되지만 빠른 압축을 사용합니다 (그렇게 압축되지는 않았지만

#pq.write_table(table, 'usedcars.csv.snappy', compression='snappy')
#pq.write_table(table, 'usedcars.csv.zip', compression='gzip')
#pq.write_table(table, 'usedcars.csv.brotli', compression='brotli')
#pq.write_table(table, where, compression='none')
```

In [25]:

```
#df = pd.DataFrame({ 'one' : [1, 2.5, 3],
#                    'two' : [ 'Peru' ,  'Brasil' ,  'Canada' ],
#                    'three' : [True, False, True]},
#                   index=list( 'abc' ))
#table = pa.Table.from_pandas(df)
#pq.write_to_dataset(table, root_path=' dataset_name' ,partition_cols=[ 'one' ,  'two' ])
```

In [ ]:

```

```