Week 6

Dingyun Hu

In [1]:
```
import pandas as pd
import time
```

In [2]:
```
import dask.dataframe as dd
```

In [3]:
```
file_path = '/Users/oliverhu/Desktop/DataGlacier Intern/week 6/2019-Nov.csv'
```

In [4]:
```
# Using pandas
start_time = time.time()
df_pandas = pd.read_csv(file_path)
pandas_time = time.time() - start_time
```

In [5]:
```
pandas_time
```

Out[5]:
```
796.856516122818
```

In [9]:
```
# Using dask
start_time = time.time()
df = dd.read_csv(file_path)
dask_time = time.time() - start_time
```

In [10]:
```
dask_time
```

Out[10]:
```
39.38685607910156
```

In [11]:
```
# Basic validation
df.columns = df.columns.str.replace('[^a-zA-Z0-9]', '_', regex=True)
```

In [12]:
```
df.info()
```

```
<class 'dask.dataframe.core.DataFrame'>
Columns: 9 entries, event_time to user_session
dtypes: object(5), float64(1), int64(3)
```

In [13]:
```
df.head()
```

Out[13]:

	event_time	event_type	product_id	category_id	category_code	brand	price	user_id	user_session
0	2019-11-01 00:00:00 UTC	view	1003461	2053013555631882655	electronics.smartphone	xiaomi	489.07	520088904	4d3b30da-a5e4-49df-b1a8-ba5943f1dd33
1	2019-11-01 00:00:00 UTC	view	5000088	2053013566100866035	appliances.sewing_machine	janome	293.65	530496790	8e5f4f83-366c-4f70-860e-ca7417414283
2	2019-11-01 00:00:01 UTC	view	17302664	2053013553853497655	NaN	creed	28.31	561587266	755422e7-9040-477b-9bd2-6a6e8fd97387
3	2019-11-01 00:00:01 UTC	view	3601530	2053013563810775923	appliances.kitchen.washer	lg	712.87	518085591	3bfb58cd-7892-48cc-8020-2f17e6de6e7f
4	2019-11-01 00:00:01 UTC	view	1004775	2053013555631882655	electronics.smartphone	xiaomi	183.27	558856683	313628f1-68b8-460d-84f6-cec7a8796ef2

In [16]:
```
config = {
    'columns': ['event_time', 'event_type', 'product_id','category_id','category_code','brand','price','user_id','us
    'separator': '|',
}
```

In [17]:
```
import yaml

with open('config.yaml', 'w') as yaml_file:
    yaml.dump(config, yaml_file, default_flow_style=False)
```

In [18]:
```
with open('config.yaml', 'r') as yaml_file:
    loaded_config = yaml.safe_load(yaml_file)

expected_columns = loaded_config['columns']
separator = loaded_config['separator']
```

In [20]:
```
import yaml

with open("config.yaml", "r") as yaml_file:
    config = yaml.safe_load(yaml_file)

expected_columns = config['columns']
if set(expected_columns) == set(df.columns):
    print("Columns match the YAML configuration.")
else:
    print("Columns do not match the YAML configuration.")
```

```
Columns match the YAML configuration.
```