01 什麼是 DATA ENGINEERING

Telung Pan
telung@mac.com

DATA 科學

線上音樂或是電影如何推薦適合你的音樂？
購物網站為什麼知道你想買什麼？

結構化資料 STRUCTURED DATA

- Structured Data can be stored in traditional database systems.
- Unstructured Data cannot be stored in terms of rows and columns.
- So it cannot be stored in traditional database systems.
- Fun Fact - Most of the Internet of Things (IoT) data is unstructured data.

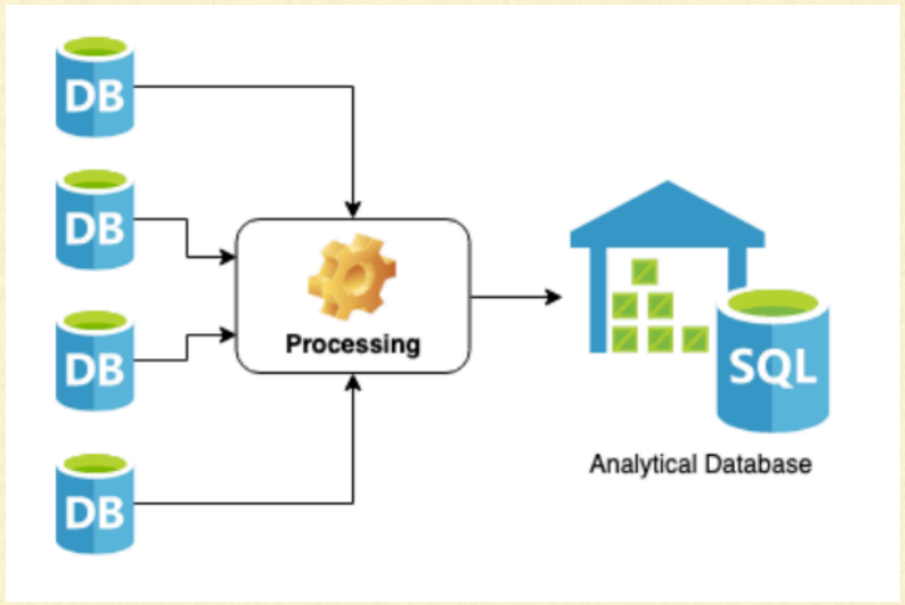
資料 & 資料科學

- Data is being produced constantly, every minute, every second with a very high speed!
- Data Science is a blend of machine learning algorithms, statistics, business intelligence and programming.
- It is helpful in discovering hidden patterns from the raw data.

資料工程師 DATA ENGINEER

- Data is scattered, not optimized for analysis
- Data engineers:
  - Gather data from different sources
  - Optimized database for analyses
  - Removed corrupt data

> *An engineer that develops, constructs, tests, and maintains architectures such as databases and large-scale processing systems.*

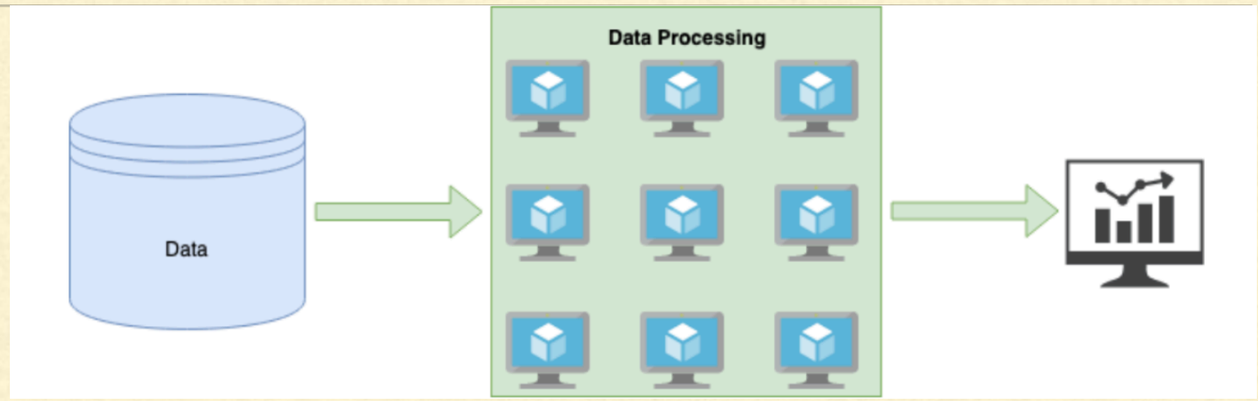


資料工程師VS 資料科學家

DATA ENGINEER	DATA SCIENTIST
Develop scalable data architecture	Mining data for patterns
Streamline data acquisition	Statistical modeling
Set up processes to bring together data	Predictive models using machine learning
Clean corrupt data	Clean outliners in data
Well versed in could technology	Monitor business process

資料工程師會用到的工具

- Databases



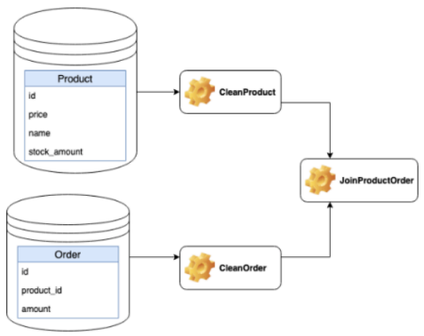
- Processing: clean, aggregate, and join data

```
df = spark.read.parquet("users.parquet")

outliers = df.filter(df["age"] > 100)

print(outliers.count())
```

- Scheduling: plan jobs with specific intervals, resolve dependency requirements of jobs



`JoinProductOrder` needs to run after `CleanProduct` and `CleanOrder`

雲端供應商

- Data processing in the cloud
  - Cover electrical and maintenance costs
  - Peaks vs. quiet moments
  - Storage, computation, databases
- AWS (32%, 2018), Azure (17%), and Google (10%)

資料庫 DATABASES

什麼是資料庫？

- Holds data
- Organizes data
- Retrieve/Search data through DBMS

> *A usually large collection of data organized especially for rapid search and retrieval.*

DATABASES AND FILE STORAGE

Databases	File systems
	
• Very organized • Functionality like search, replication, ...	• Less organized • Simple, less added functionality

結構化 與 非結構化資料

Structured: database schema

- Relational database

Semi-structured

- JSON

```
{ "key": "value"}
```

Unstructured: schemaless, more like files

- Videos, photos

SQL 與 NOSQL

SQL

- Tables
- Database schema
- Relational databases

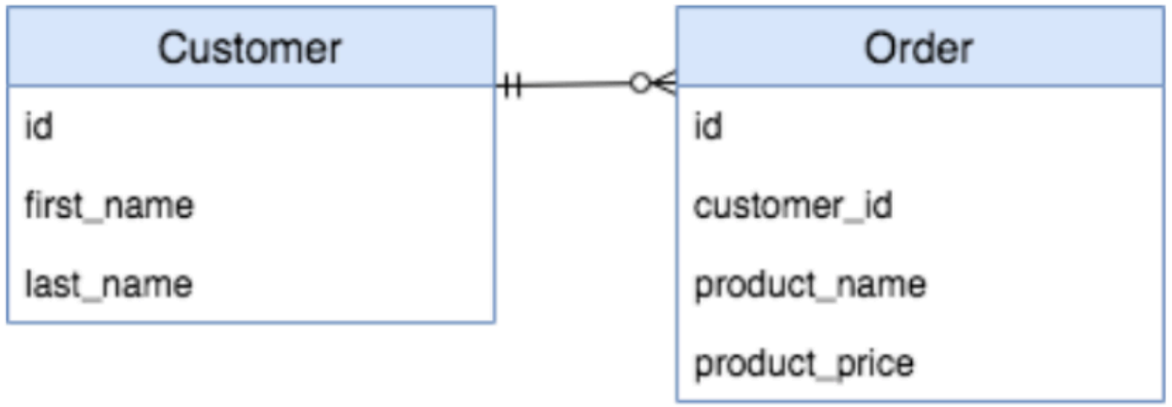
NoSQL

- Non-relational databases
- Structured or unstructured
- Key-value stores (e.g. caching)
- Document DB (e.g. JSON objects)

SQL: THE DATABASE SCHEMA

```sql
-- Create Customer Table
CREATE TABLE "Customer" (
  "id" SERIAL NOT NULL,
  "first_name" varchar,
  "last_name" varchar,
  PRIMARY KEY ("id")
);

-- Create Order Table
CREATE TABLE "Order" (
  "id" SERIAL NOT NULL,
  "customer_id" integer REFERENCES "Customer",
  "product_name" varchar,
  "product_price" integer,
  PRIMARY KEY ("id")
);
```




```sql
-- Join both tables on foreign key
SELECT * FROM "Customer"
INNER JOIN "Order"
ON "customer_id" = "Customer"."id";
```

```
 id | first_name | ... | product_price
  1 | Vincent    | ... |            10
```

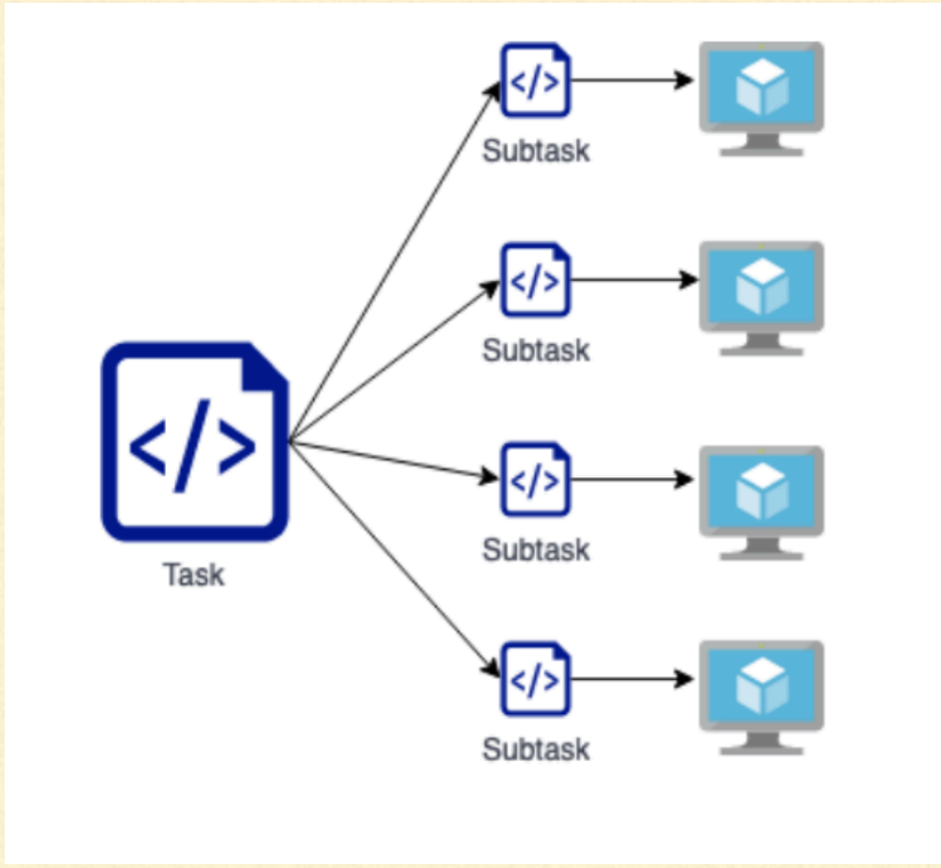
練習

- https://rextester.com/l/postgresql_online_compiler
- Create tables, query, join query

平行運算 PARALLEL COMPUTING

IDEA BEHIND PARALLEL COMPUTING

- Split task into subtasks
- Distribute subtasks over several computers
- Work together to finish task



FROM TASK TO SUBTASKS

```
# Function to apply a function over multiple cores
@print_timing
def parallel_apply(apply_func, groups, nb_cores):
    with Pool(nb_cores) as p:
        results = p.map(apply_func, groups)
    return pd.concat(results)

# Parallel apply using 1 core
parallel_apply(take_mean_age, athlete_events.groupby('Year'), 1)

# Parallel apply using 2 cores
parallel_apply(take_mean_age, athlete_events.groupby('Year'), 2)

# Parallel apply using 4 cores
parallel_apply(take_mean_age, athlete_events.groupby('Year'), 4)
```

APACHE AIRFLOW

- pip install apache-airflow (3 mins)
- airflow initdb
- Python3
  - from airflow import DAG #DAG object
  - from airflow.operators.bash_operator import BashOperator
  - from datetime import datetime, timedelta

資料處理

What all properties do we need to know before acquiring data?

Select the right answer

- [ ] not all data is completely clean
- [ ] fact checking
- [ ] where to find data
- [x] all of the above

資料的種類

- Primary and Secondary: 主要、次要
  - Primary data is the data that you collect or generate.
  - Secondary data is created by other researchers.
- Qualitative and Quantitative: 質性、量性
  - Qualitative refers to text, images, video, sound recordings, observations, etc.
  - Quantitative refers to numerical data.