How to Gather and Import Data :

↳ Using queries to extrat data from SQL Data bases : used for Relational Databases

↳ NoSQL : ใช้ SQL ก็ได้ในหลายประเภท

- แต่ละอันก็จะใช้ tool ของตัวเอง เช่น CQL for Cassandra และ GraphQL for Neo4J

↳ API : นิยมใช้ Extracting data จาก source หลายแหล่ง

- ถูกเรียกใช้จาก App ที่ต้องการ data และต้องการเข้าถึง end point ที่มี data อยู่ โดยที่ end point ประกอบไปด้วย databases, web services, และ data market place
- ถูกใช้สำหรับ Validation data ได้ด้วย เช่น ตรวจสอบที่อยู่ รหัสไปรษณีย์ สำหรับชื่อเมืองเพื่อว่าที่อยู่นั้นตรงกันไหม

↳ Web Scraping (Screen Scraping, web Harvesting)

- เป็นการ download web page จาก parameter ที่เรากำหนดเอาไว้
- เพื่อดึงเอา text, contact information, images, video, podcasts และรายการสินค้า
- RSS feed มักใช้ในการ capture updated data จาก online forum

↳ Sensor data : Data Stream เป็นแหล่งข้อมูลที่รับจากการไหลของข้อมูลต่อเนื่องจาก source เช่น Intruments, IoT device, App และ Gps

- Data Stream และ feed ยังถูกใช้ในการดึง data จาก web site social media อีกด้วย

↳ Data Exchange

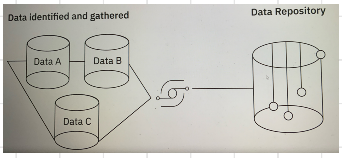
- อนุญาตให้สามารถแลกเปลี่ยนข้อมูลระหว่างผู้ให้บริการและผู้ใช้บริการ
- อำนวยความสะดวก
- มีความปลอดภัย

↳ Other sources : Numerous other data sources can be tapped into for specific data needs.

For example

- Forrester and Business Insider for marketing trends and ad spending
- Research and advisory firms such as Gartner and Forrester for strategic and operational guidance
- Agencies for user behavior data, mobile and web usage, market surveys, and demographic studies.

Importing Data:



Data type and Destination Repositories : Repository เหมาะกับประเภทข้อมูลแต่ละชนิด

- Structure Data : มี source มาจาก OLTP system, spreadsheets, online forums, sensors, network and web logs สามารถเก็บใน NoSQL ได้
- Semi-structure Data : Sources include emails, XML, zipped files, binary executables, and TCP/IP protocols สามารถเก็บได้ใน NoSQL Cluster XML และ Json นิยมใช้ในการ exchange ในโครงสร้างนี้
- Unstructure Data : Sources include web pages, social media feeds, images, videos, documents, media logs, and surveys เก็บใน NoSQL และ Data Lake

ETL และ Pipeline อำนวยความสะดวกในการใช้ Automate function ในการแปลงข้อมูล

Data Wrangling:

Raw Data ต้องผ่านการ transformation และ cleaning ซึ่งอยู่ในขั้น Data Wrangling

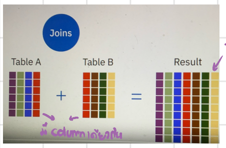
↳ Data Wrangling หรือ Data munging เกี่ยวกับ data exploration, transformation, validation และการทำ data ให้พร้อมสำหรับวิเคราะห์

Transformation :

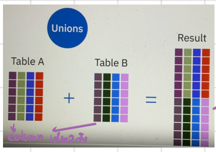
↳ Structuring Data : • เป็นการเปลี่ยนแปลง schema และ form ของมัน เพื่อให้สอดคล้องและนำไป Analyse ต่อ

- เช่น การ merge data จาก source ที่ต่างกัน เช่น Relational Database กับ web API โดยที่ต้องแปลง API เพื่อให้ใช้ได้

↳ Join & Union :



Join แล้วเพิ่ม column



Unions แล้วเพิ่ม Row

↳ Normalization : เป็นการ clean data ที่ซ้ำ และไม่สอดคล้องออกไป

และ Denormalization : ทำการรวมข้อมูลจากหลาย table ให้เป็น table เดียว เพื่อ query ไว และ Analyse ง่าย

↳ Cleaning : แก้ไข data ที่ไม่ปกติ ทำให้ data มีความน่าเชื่อถือ และแม่นยำในการ Analyse

- Detecting issues and errors : ใช้ script และ tools ในการหา error ใน data set
- Validating against rules and constrains:
- Data Profiling : ทำให้รู้ data structure, content, และ relationship ภายใน data ทำให้รู้คุณภาพ และความผิดปกติ
- Visualizing the Data : ทำ Visualize จาก math และสถิติ ทำให้รู้ ความผิดปกติ และปัญหาคุณภาพ เช่น blank Value, ค่าซ้ำ, ค่าผิดปกติ

Cleaning :

technique ที่ใช้ clean dataset ขึ้นกับสถานการณ์ และปัญหาที่เจอ

↳ Missing Value : ค่าไม่ครบอาจมี bias

- วิธี : Filter out record ด้วย missing value, การหาข้อมูลเพิ่มเติม, ประมาณค่าจากสถิติ แล้วใส่ไป

↳ Duplicate Data : ต้องลบ เพราะอาจเกิดเบี่ยงเบนผลการวิเคราะห์

↳ Addressing Irrelevant Data : Data ที่ไม่สอดคล้องกับ context หรือ purpose ของการ Analyse ให้เอาออก เช่น ลบเบอร์ออกจาก Data ที่จะวิเคราะห์สุขภาพ

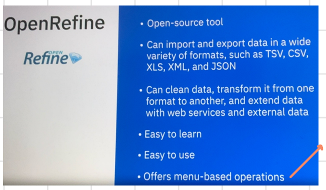
↳ Data Type Conversion and Standardization : ทำให้ data อยู่ใน format เดียวกัน เพื่อความแม่นยำในการวิเคราะห์ เช่น แปลง string เป็นตัวพิมพ์เล็ก หรือ แปลงวันที่ให้อยู่ในรูปแบบเดียวกัน

↳ Fixing Syntax Errors and Typos (พิมพ์ผิด) : การจัดการที่ไม่ถูกต้อง หรือพิมพ์ผิดต้องแก้ไขให้ถูก เพื่อความสอดคล้องและแม่นยำ

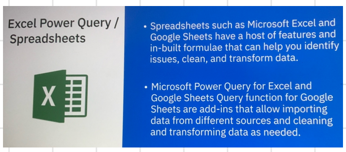
↳ Handling Outlier : บางค่าอาจผิดสังเกต เพราะต่างจากค่าอื่น เราควรสังเกตและตรวจสอบว่าค่านั้นผิดหรือไม่ ถ้าถูกอาจทำให้ Analyse แตกต่างไป ต้องตรวจดูได้

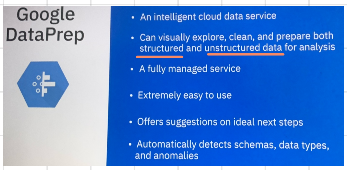
tools for Data Wrangling

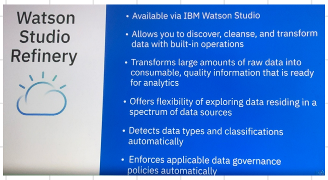
- Excel Power Query / Spreadsheets
- OpenRefine
- Google DataPrep
- Watson Studio Refinery
- Trifacta Wrangler
- Python
- R

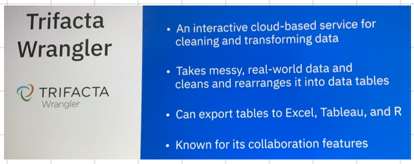


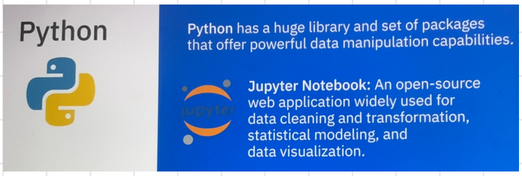
คือเราไม่จำเป็นต้องจำคำสั่ง

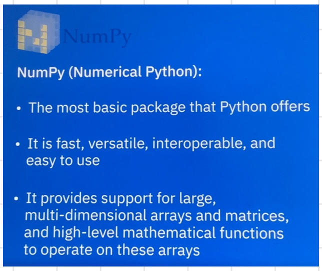


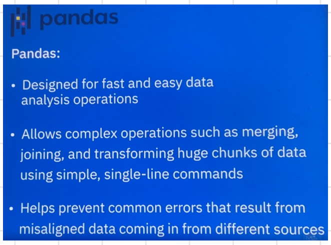


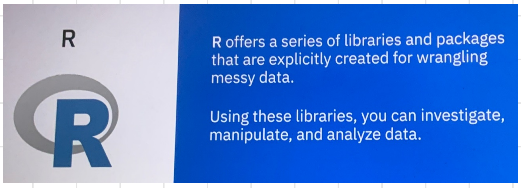


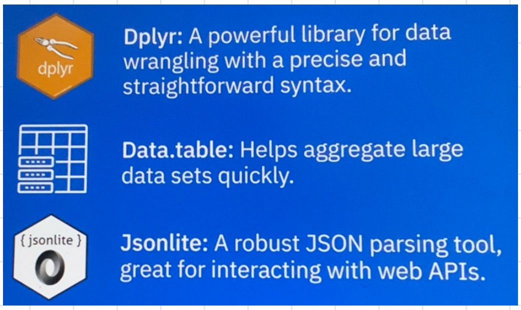












Performance Tuning and Troubleshooting: เน้นที่ Data Pipeline และ Data bases

↳ Perfomance Threats in Data Pipeline:

- Scalability : Data Pipeline ต้อง handle การเพิ่มขึ้นของปริมาณ data และประสิทธิภาพของ workload เมื่อ data เพิ่ม ความสามารถของ pipeline ที่จะขยายได้จะส่งผลต่อการ maintain performance
- Application Failures : Failure ใน pipeline เช่น error ใน processing สามารถรบกวน data flow ส่งผลต่อประสิทธิภาพได้
- Scheduling Issues : งานอาจไม่เริ่มตามเวลาที่กำหนด ส่งผลต่อ work flow ได้
- Tool Incompatibilities : ความไม่เข้ากันของ tools ใน pipe line อาจทำให้มันไร้ประสิทธิภาพได้ เพราะ pipeline ใช้หลาย tools ในนั้น

↳ Performance Metrics for Data Pipeline:

- Latency : เวลาทั้งหมดที่ใช้ตอบสนองต่อคำสั่ง request ของ pipeline
- Failure : ส่งผลต่อความน่าเชื่อถือ
- Resource Utilization : CPU, memory, network ต้องจัดการทรัพยากรให้เหมาะสม
- Traffic : จัดปริมาณผู้ใช้

↳ Troubleshooting Data Pipeline Performance Issues:

- Information Collection : รวบรวมข้อมูลที่เกี่ยวข้อง ดูว่ามันส่งผลกับปัญหาด้าน performance อย่างไร
- Version and Deployment Check : ดู version software ว่าส่งผลกระทบต่อ performance ไหม
- Log and Metrics Analysis : ตรวจสอบ log เพื่อแยกปัญหาที่เกี่ยวกับ infrastructure, data หรือ software
- Reproduction in Test Environment: การ Analyse ให้เจอปัญหาที่แท้จริง ลองสร้างเหตุการณ์จำลองในสภาพแวดล้อม เพื่อทดสอบวิธีแก้ไข

↳ Database Optimization Practise:

- Capacity Planing : Plan กำหนด Resource ของ hardware และ software ที่จะใช้ในอนาคตและปัจจุบัน
- Indexing : สร้าง Index เพื่อความเร็วในการดึงข้อมูลที่ดีขึ้น
- Partitioning : แบ่งตารางใหญ่ออกเป็นส่วนๆ เพื่อปรับปรุงประสิทธิภาพ Query และ data management
- Normalization : ออกแบบ database ไม่ให้ complex ช่วยให้ Query ดีขึ้นและ data ครบถ้วนขึ้น

↳ Monitoring and Alerting System:

- Database Monitoring tools: Capture performance metrics เพื่อติดตามแนวโน้มประสิทธิภาพและระบุปัญหา
- Application Performance Management (APM) Tools: ตรวจสอบ performance App เช่น เวลาตอบสนอง และการใช้ทรัพยากร เพื่อระบุปัญหาและเพิ่มประสิทธิภาพ Resource
- Query Performance Mornitoring Tools: รวบรวมสถิติเกี่ยวกับการ Query และการใช้ Resource เพื่อเพิ่มประสิทธิภาพ Resource
- Job-level Runtime Monitoring: แบ่งกระบวนการที่ใช้เวลานานออกเป็นขั้นตอนเล็กๆ สามารถตรวจสอบความสมบูรณ์และเวลาในการดำเนินการ ช่วยให้เพิ่มความประสิทธิภาพของ Pipeline

↳ Maintenance Routines:

- Preventive Maintenance: บำรุงรักษาเป็นประจำ เพื่อระบุและแก้ไขปัญหาก่อนจะเกิด ก่อนจะส่งผลอันตรายต่อระบบ
- Time-Based Maintenance: Maintenance ตามเวลาที่กำหนด
- Condition-Based Maintenance: ประสิทธิภาพลดลงตาม conditioning ไม่ต้องมีการหยุดทำงาน และประหยัดต้นทุนเพื่อสภาพอื่น

Governance and Compliance:

↳ Data Governance: คือ collection ของ principles แนวทาง, practice, และ process ในการ maintain security, privacy, และ Integrity ตลอด data life cycle

↳ Types of Data ที่ต้องการการดูแล: • privacy data, sensitive data, ข้อมูลที่ต้องการ track ต้องมีการดูแลรักษาเป็นพิเศษ

↳ Compliance: การปฏิบัติตามกฎระเบียบ (Regulation) เช่น กฎหมายที่เกี่ยวข้องกับการใช้ข้อมูล ในการเก็บ, ประมวลผล, และการส่ง

In this lesson, you have learned:

Data Governance is a collection of principles, practices, and processes that help maintain the security, privacy, and integrity of data through its lifecycle.

Personal Information and Sensitive Personal Information, that is, data that can be traced back to an individual or can be used to identify or cause harm to an individual, needs to be protected through governance regulations.

General Data Protection Regulation, or GDPR, is one such regulation that protects the personal data and privacy of EU citizens for transactions that occur within EU member states.
Regulations, such as HIPAA (Health Insurance Portability and Accountability Act) for Healthcare, PCI DSS (Payment Card Industry Data Security Standard) for retail, and SOX (Sarbanes Oxley) for financial data are some of the industry-specific regulations.

Compliance covers the processes and procedures through which an organization adheres to regulations and conducts its operations in a legal and ethical manner.

Compliance requires organizations to maintain an auditable trail of personal data through its lifecycle, which includes acquisition, processing, storage, sharing, retention, and disposal of data.

Tools and technologies play a critical role in the implementation of a governance framework, offering features such as:

- Authentication and Access Control.
- Encryption and Data Masking.
- Hosting options that comply with requirements and restrictions for international data transfers.
- Monitoring and Alerting functionalities.
- Data erasure tools that ensure deleted data cannot be retrieved.

Module 4

In this lesson, you have learned:

Data Engineering is reported to be one of the top ten jobs experiencing tremendous growth in the U.S. today. It is also reported to be one of the fastest growing tech occupations with year-over-year growth of around 50%.

Currently, the demand for skilled data engineers far outweighs the supply, which means companies are willing to pay a premium to hire skilled data engineers.

Data engineering roles in organizations tend to break the specialization up into Data Architecture, Database Design and Architecture, Data Platforms, Data Pipelines and ETL, Data Warehouses, and Big Data.

Regardless of the niche you choose to specialize in, knowledge of operating systems, languages, databases, and infrastructure components, is essential.

To work your way up from a Junior Data Engineer to a Lead or Principal Data Engineer, you need to continually advance your technical, functional, and soft skills from a foundational level to an expert level. You need to not only expand your skills in your niche area, but also into other areas of data engineering at the same time.

Big Data Engineers and Machine Learning Engineers are some of the emerging roles in this field and they require specialized skills in addition to basic data engineering.

There are several paths you can consider in order to gain entry into the data engineering field.

- An academic degree in Computer Science or engineering qualifies you for an entry-level job.
- If you are not a graduate, or a graduate in a non-relate stream, you can earn professional certifications from online multi-course specializations offered by learning platforms such as Coursera, edX, and Udacity.
- If you have a coding background, or you are an IT Support Specialist, a Software Tester, a Programmer, or a data professional such as a Statistician, Data Analyst, or BI Analyst, you can upskill with the help of online courses to become a Data Engineer.