



Data Engineering & Analytics

by: Basel Husam

Chapter 1 - What is Data Engineering?

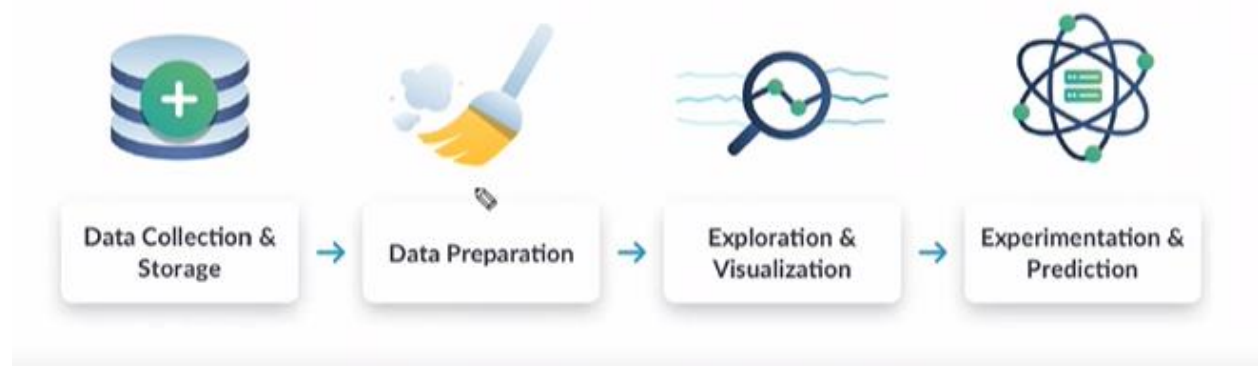
Chapter 1 topics:

1. Data Engineering and Big Data.
2. Data Engineers vs. Data Scientists.
3. Data pipelines.

❖ *Data Engineering and Big Data:*

➤ Data Workflow (The lifecycle for the data):

1. Data Collection & Storage
2. Data Preparation
3. Exploration & Visualization
4. Experimentation & Prediction



1. Data Collection & Storage:

- Before Everything, we must collect data.
- Collecting data can be from multiple resources, such as data warehouses, database, data lake, etc.

■ لازم اول اشي نعمله انه نجيب انه نجمع البيانات، وعملية التجميع ممكن تكون من اكثر من مكان يعني ممكن تكون من database عادية او data warehouse ... الخ.

2. Data Preparation:

- It's the process of cleaning the data and making it ready for analysis.
- Some of the data preparation tasks:
 - Data discovery
 - Data cleaning:
 - missing values removal
 - handling duplicates
 - delete, fix, or handle corrupted data ... etc.
 - Data transformation
 - Data validation and publishing

3. Exploration & Visualization:

- When the data are well organized and cleaned, then you explore the data and try to understand it, whether by using descriptive statistics or making statistical graphs or finding correlations, or finding differences between two datasets.

4. Experimentation & Prediction:

- The final step is building a model for making predictions.
 - Building a machine learning model allows you to make predictions for the future, or to have answers for specific assumptions.
-

➤ Data Engineers Deliver:

1. The Correct data: high-quality data.
2. In the right form: well-formatted.
3. To the right people: such as:
 - a. Data Analyst
 - b. Data Scientist
 - c. Machine Learning Engineer
4. As efficiently as possible: for example, if the data has length and width, I can calculate the size and give it to the right people instead of the length and width.

➤ A Data Engineers Responsibilities:

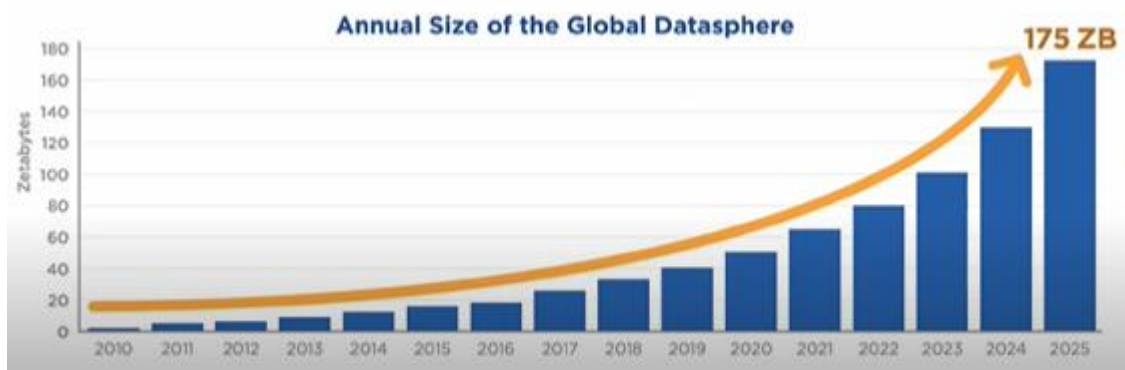
1. Ingest data from different resources:
 - تجميع البيانات من مصادر مختلفة.
2. Optimize databases for analysis:
 - الفكرة هون انه من ال database الاصلية او data lake الاصلية اقدر اطلع databases تانيين بساعدوني اني احل مشكلة معينة، مثلا من data lake فيها بيانات عن كثير اشياء، بقدر اني استخلص منها بيانات من نفس المجال وبصبو بنفس ال domain وبحطهم مثلا ب database لحالهم عشان اشتغل عليهم لحال.
3. Remove corrupted data:
 - بعض البيانات ممكن تكون مضروبة، على سبيل المثال صور ما بتفتح او text فيها رموز غريبة او مش مدعومة. هاي البيانات لازم اتخلص منها.
4. Develop, construct, test, and maintain data architectures.

➤ Data Engineers and Big Data:

- Data Engineers becomes needed more and more because of Big Data.
- Big Data:
 - is very large in volume, so you have to know how to deal with its size.
 - The traditional methods won't work anymore because of its size.

➤ Big Data Growth: امثلة ل اهم اسباب نمو البيانات الكبيرة

- Sensors and devices
- Social media
- Enterprise data
- VoIP (voice communication, multimedia sessions)



- الصورة اللي فوق بتوضحلنا قديش انه البيانات عم بزيد حجمها بشكل هائل وكثير كبير خلال السنين و انه ب 2025 حيوصل حجم البيانات تقريبا 175 Zettabyte

1 Zettabyte = 10^9 Terabyte ■

➤ The Five Vs:

- ال five Vs عبارة عن 5 خصائص لل big data و اسمهم ال five Vs لانه ال 5 خصائص بتبدأ بحرف ال V.

1. Volume (how much?)

- حجم البيانات الكبيرة بكون كبير كثير, بعصرنا الحالي ممكن يكون حجمها بال Petabyte وال Zettabyte, ليهك اول خاصية للبيانات الكبيرة انه حجمها كبير.
- ملاحظة: ممكن بيانات معينة تكون الك big data بس لغيرك لا ... كيف يعني هاد الاشئ؟
على سبيل المثال في بيانات مساحتها 10 Terabytes, انت ك طالب وجهازك الحاسوب ك جهاز طالب
حيكون عليك شبه مستحيل انه تقدر تتعامل مع هاي البيانات او تشتغل عليها او حتى تقدر تفتحها او تشوفها,
لانه حجمها عملاق بالنسبة لك, بس مثلا لو شركة جوجل اخدت هاي البيانات نفسها عشان تشتغل عليها,
حيكون سهل عليها انها تتعامل معها لانه بالنسبة الها ك شركة كبيرة ال 10 تيرابايت ولا اشئ.
لهيك نفس البيانات ممكن يختلف تصنيفها من شخص ل اخر, ومجرد ما صار صعب جدا التعامل مع البيانات
ممكن نحكي انها big data.

2. Variety (what kind?)

- الاختلاف ب انواع البيانات, ممكن تكون البيانات عبارة عن text او images او tweets او videos او audio او soundtracks ... الخ.
- حتى بنوع البيانات الواحد في اختلاف, يعني مثلا ال images هي عبارة عن صور, بس هاي الصور ممكن تكون صور عن مستشفى, طرق وشوارع, صور من اقمار صناعية (صور للفضاء) وهكذا, ف انه حتى النوع الواحد من البيانات في انواع واشكال مختلفة.

3. Velocity (how frequent?)

- قديه البيانات متكررة, و velocity معناها السرعة, يعني قديش سرعة هاي البيانات او كل قديش بتتغير البيانات, مثلا عليها سعر الاسهم بالشركات, ممكن السعر كل ثواني يتغير, ليهك سرعة البيانات كبيرة, و هاد بخلي ال analysis تكون more challenging لانه العملية بتصير اصعب

4. Veracity (how accurate?)

- قديده accurate او دقيقة هاي البيانات؟ وهل المصدر اللي اخدنا منه هاي البيانات موثوق؟
- الهدف مش بس اني اجيب اي بيانات المهم بيانات..... لأ، اذا البيانات اللي انا ماخذها مش صحيحة ف فش فائدة من هاي البيانات
- على سبيل المثال انت بدك تبني model يتنبأ ب مرض معين، بتحتاج تعرف درجة الحرارة، العمر، الجنس، الوزن، والخ من هاي المعلومات. هلا اذا اصلا البيانات اللي اجتني غلط، يعني مثلاً الممرض كان يعيبي البيانات من عنده وبحط اي اشي، ف هاي البيانات عالفاضي، التنبؤ تاع الموديل حيكون غلط لانه اصلا البيانات مش صحيحة.

5. Value (how useful?)

- هل البيانات اللي عندي مفيدة؟ هل حقدر اطلع منها value واصنع منها action واشي ملموس؟ هل لما ابني موديل من هاي البيانات حيفيدني؟ لازم نجابو على هاي الاسئلة قبل منبلش نشتغل على البيانات.
- الهدف اصلا من كل هاد الموضوع وال main goal انه نطلع value ونستفيد من هاي البيانات، ف لو ما بنستفيد منها ف هاي البيانات بتلزمناش.
- مثال على انه نطلع action من البيانات و اشي ملموس، مثلاً انت بنيت موديل يتنبأ ب مرض معين، هاد الموديل مفيد وملموس وممكن المستشفيات تصير تستعمله
كمان امثلة: موديل يتنبأ ب سعر بيت، موديل يستخلص كلام من صورة ... الخ.

Chapter 2 – Data Structures

❖ Data Structures:

➤ Structured Data:

- Easy to read and organize
- Consistent model, rows and columns
- Defined types
- Can be grouped to form relations
- Stored in relational databases
- About 20% of the data is structured
- Created and queried using SQL

- البيانات بكون سهل علي اني اقرأها وبتكون مرتبة
- بتكون البيانات منسقة على شكل سطور واعمة
- نوع البيانات مكون معروف، يعني بسهولة بنقدر نوع البيانات بكل عمود ان كان numerical او categorical
- ممكن نجمع اكثر من بيانات مع بعض ونحطهم بجدول واحد (زي ما حنشوف مثال بالصور تحت)
- يتم تخزينها ب relational databases او RDMS (Relational Database Management System)
- 20% من البيانات الموجودة بالعالم عبارة عن structured data ، ف بنعرف انه نسبة ال unstructured data اكبر بكثير
- يتم انشاؤها والتعامل معها عن طريق query ب SQL، ولهيك بكون استخلاص المعلومات من البيانات بسهولة، يعني ممكن انت تلاقى معلومة بتدور عليها عن طريق كتابة query وحدة ب SQL

Example of Structured Data:***Table 1:*****Employee table**

Index	last_name	first_name	role	team	full_time	office
0	Thien	Vivian	Data Engineer	Data Science	1	Belgium
1	Huong	Julian	Data Scientist	Data Science	1	Belgium
2	Duplantier	Norbert	Software Developer	Infrastructure	1	United Kingdom
3	McColgan	Jeff	Business Developer	Sales	1	United States
4	Sanchez	Rick	Support Agent	Customer Service	0	United States

Table 2:**Relational database**

office	address	number	city	zipcode
Belgium	Martelarenlaan	38	Leuven	3010
UK	Old Street	207	London	EC1V 9NR
USA	5th Ave	350	New York	10118

Merging the two tables:**Relational database**

Index	last_name	first_name	office	address	number	city	zipcode
0	Thien	Vivian	Belgium	Martelarenlaan	38	Leuven	3010
1	Huong	Julian	Belgium	Martelarenlaan	38	Leuven	3010
2	Duplantier	Norbert	UK	Old Street	207	London	EC1V 9NR
3	McColgan	Jeff	USA	5th Ave	350	New York	10118
4	Sanchez	Rick	USA	5th Ave	350	New York	10118

➤ Semi-structured Data:

- Relatively easy to search and organize
- Consistent model, less-rigid implementation: different observations have different size
- Different types
- Can be grouped, but needs more work
- NoSQL databases: JSON, XML, YAML

■ ال semi-structured data هي عبارة عن مكس ما بين ال structured وال unstructured data ومن الامثلة عليه ال JSON files ، هاد النوع بتكون البيانات مرتبة بطريقة معينة بس مش structured

■ بنقدر نحول ال JSON files على سبيل المثال ل structured data وتكون organized ونحطها ب relational database عن طريق tools وادوات معينة، لهيك هي can be grouped, but needs more work

Example of JSON file:

Favorite artists JSON file

```
{
  "user_1645156": {
    "last_name": "Lacroix",
    "first_name": "Hadrien",
    "favorite_artists": ["Fools in Deed", "Gojira", "Pain", "Nanowar of Steel"]},
  "user_5913764": {
    "last_name": "Billen",
    "first_name": "Sara",
    "favorite_artists": ["Tamino", "Taylor Swift"]},
  "user_8436791": {
    "last_name": "Sulmont",
    "first_name": "Lis",
    "favorite_artists": ["Arctic Monkeys", "Rihanna", "Nina Simone"]},
  ...
}
```

■ في كثير ناس او شركات بتحب تتعامل مع ال JSON/XML files لعدة اسباب، منها انه التعامل مع هاد النوع يكون flexible من ناحيو ال transforming، انه ممكن تنقل البيانات كاملة بسهولة عن طريق فلاشة او ايميل ... الخ.

Metadata: Data that provide information about other data.

- يعني على سبيل المثال الفيديو الموجود على يوتيوب يكون الـ description ف كلام اللي يكون بال description box هو عبارة عن الـ metadata والفيديو نفسه هو عبارة عن الـ data ليهيك الـ metadata is data about data.

➤ Unstructured Data:

- Does not follow a model, can't be contained in rows and columns
- Difficult to search and organize
- Usually text, sound, pictures, and videos
- Usually stored in data lakes, can appear in data warehouses or databases
- Most of the data is unstructured
- Can be extremely valuable

- الـ unstructured data ما بتكون مرتبة على شكل rows and columns لانها اصلا ممكن تكون البيانات عبارة عن text او images او videos او audio files ليهيك صعب انه احنا نخليها organized

- البحث فيها صعب، ليش؟ مثلا هلا انت لما تبحث باليوتيوب عن اشي معين، هو بدولك على الاشياء اللي انت بتدور عليه بال titles وال descriptions للفيديوهات، ليهيك اذا انت كنت بتبحث عن اشي معين جوا الفيديو نفسه ف ما رح يطلعك اياه، لانه هاي العملية very complex and difficult كمان مثال ممكن نوحده نفس النقطة اللي هة الافلام، انت اذا بتدور ع مشهد معين جوا فيلم وكتبت عنه ب محرك البحث ما رح يطلعك اياه، لانه زي ما حكينا هاد الاشياء لسا عم يشتغلوا عليه و very difficult to implement.

- بنخزنها غالبا بال data lakes، الـ data lake عبارة عن مستودع تخزين واسع، ومن اسمها بحيرة البيانات يعني عبارة عن بحيرة بصب فيها اكثر من اشي، ليهيك ممكن نحكي الـ data lake مستودع تخزين بنخزن البيانات فيه وهاي البيانات بتكون من resources مختلفة.

- الـ unstructured data الها قيمة كبيرة وال value تاعتها كثير عالية، بس عملية الـ process فيها اصعب من الـ structured data ، بس بالآخر احنا بهمنا انه نشغل كل الشغل من الـ data collection ل building a model انه يكون في قيادة من هاد الموديل او هاي البيانات، مش كلشي يكون عالقاضي

- من ال concepts اللي احنا بنستخدمهم لتعامل مع ال unstructured data همة ال NLP وال image processing
- ال NLP او ال Natural Language Processing هو كيف نخلي ال computers تقدر تفهم ال voices او ال audio files زي كأنها مكتوبة ومقروءة
- ال image processing هي العملية انه نخلي ال computers تتعرف ع اشيء موجودة بالصورة على شبييل المثال اخليه يعرف الفرق بين البسة والكلب بحيث انه اعطيه صورة واخليه يتنبأ اذا الحيوان اللي بالصورة هة بسة ولا كلب، وهاد الاشئ اسمه image processing بحيث اخلي ال computer يدخل جوا الصورة ويفهمها زي كأنه انسان

Adding some Structure:

- Use AI to search and organize unstructured data
- Add information to make it semi-structured
- زي ما حكينا فوق ممكن نعمل سيرش فيهم عن طريق ال NLP وال image processing وهذول الحالتين او التقنيتين همة عبارة عن AI
- برضه زي ما حكينا فوق مثلاً الفيديوهات اذا ضفتلها description ممكن تصير semi-structured لاني انا هون بسهل عملية البحث بين هاي ال unstructured data

DATA LOCALITY: is the process of moving computation to the node where that data resides, instead of vice versa.

- يعني على سبيل المثال في بيانات مساحتها 10 Terabyte ف انت حيكون صعب عليك انك تقدر تتعامل معها من خلال جهازك البسيط، لهيك مبدأ ال data locality بحكيك انه بدل ما انت تجيب الداتا عندك وتشغل عليها وتكتب كود عليها ... خلي الكود تاعك يروح محل ما الداتا مخزنة (مش بجهازك) ويتطبق الكود عليها وهي هناك، وانت بتقدر تشوف ال results وتأخذها.

Summary:

- Structured data
- Semi-structured data
- Unstructured data
- Difference between the three
- Give examples

Questions:

Q1: Order the following:

- Exploration and visualization
- Data preparation
- Data collection and storage
- Experimentation and prediction

Q2: DE or not DE (Data Engineering or not):

- A. Optimizing the customers for analysis
- B. Ensuring corrupted, unreadable music tracks are removed and don't end up facing customers
- C. Gathering music consumptions data from desktop and mobile sources
- D. Running an experiment to identify the optimal search bar positioning in the app
- E. Based on their listening behavior, predict which songs customers are likely to enjoy
- F. Building a visualization to understand listening patterns by city

Q3: True or False:

- A. Value refers to how actionable the data is
- B. Data types refer to the variety of the data
- C. Velocity refers to how big the data is
- D. Volume has to do with how trustworthy the data is
- E. Veracity refers to how frequently the data is generated

Q4. Tell me the truth:

In 2012, IBM declared that 90% of the data in the world had been created in the past 2 years. That same year, the amount of digital data in the world first exceeded 1 zettabyte (1 billion terabytes). In 2020, we're expected to reach 44 zettabytes. This big data era led to the advent of two new roles: data engineers and data scientists, you just studied the differences between these two roles.

Let's have a quick sanity check: which of the following options is true?

- A. Data engineers intervene at the very end of the data workflow
- B. Data scientists build pipelines
- C. Data engineers need strong statistical expertise
- D. Data engineers enable data scientist

Q5: Assign the task to the data engineer or the data scientist:

- A. Provide listening sessions data so it can be analyzed with minimal preparation work
- B. Find out in which countries certain artists are popular to give them insights on where to tour
- C. Ensure that people who use the databases can't erase music videos by mistake
- D. Use Python to run an analysis on whether users prefer having the search bar on the top left or the top right of the Spotify desktop app
- E. Use Java to build a pipeline collecting album covers and storing them
- F. Identify which customers are likely to end their Spotify subscriptions, so marketing can target them and encourage them to renew

Q6: It's not true:

The main objective, when setting up data pipelines, is to improve the efficiency with which data flows from its ingestion to the final user

Most of the options below are true, but one is false, which one is it?

- A. Data pipelines ensure an efficient flow of the data through the organization
- B. Data pipelines automate data extraction
- C. Data pipelines necessarily include a transformation step
- D. ETL stands for Extract, Transform, and Load

Answers:

Q1:

- A. Data collection and storage
- B. Data preparation
- C. Exploration and visualization
- D. Experimentation and prediction

Q2:

Data Engineering tasks:

- Optimizing the customer's databases for analysis (A)
- Ensuring corrupted, unreadable music tracks are removed and don't end up facing customers. (B)
- Gathering music consumption data from desktop and mobile sources (C)

All the others are *not data engineering tasks*

Q3:

- $A + B \rightarrow \text{True}$
- $C + D + E \rightarrow \text{False}$

Q4: Data Engineers enable Data Scientist

Q5:

- $A + C + E \rightarrow \text{DE (Data Engineering)}$
- $B + D + F \rightarrow \text{DS (Data Scientist)}$

Q6: Data pipelines necessarily include a transformation step

❖ SQL:

- Structured Query Language
- Industry-standard for Relational Database Management System (RDBMS)
- Allows you to access many records at once, and group, filter or aggregate them
- Close to written English, easy to write and understand
- Data engineers use SQL to **create and maintain databases**
- Data scientists use SQL to **query (request information from) databases**

- ال SQL يعتبر ال standard للتعامل مع ال RDBMS، ف ال SQL هو الشكل العام للغة بس اللي احنا بنستخدمها بتكون تفرعات من SQL نفسه، يعني احنا لما نكتب كود SQL هو اوك يكون SQL بس الكود يكون عبارة عن MySQL او PostgreSQL ... الخ
لهيك ال SQL هو الشكل العام لهاي اللغة بس الكود اللي احنا بنكتبه يكون عبارة عن واحد من ال implementations لها
- بتخليك تقدر توصل لعدد من ال records او ممكن نسميهم ال rows بسهولة، او انك تجمع مجموعة من البيانات مع بعض، او تفلترهم بسهولة
- قريبة من اللغة الانجليزية، يعني ممكن واحد ما بفهم بالبرمجة اشي ويقرأ ال SQL statement ويعرف شو بتعمل لانه ال statements نفسهم واصفين حالهم
- الفرق بين استخدام مهندس البيانات لل SQL عن عالم البيانات هو انه مهندس البيانات يستخدم ال SQL ليصنع ال databases ويحافظ عليهم ويكونوا efficient as possible.
اما عالم البيانات يستخدم ال SQL ليستخلص بيانات من ال databases اللي عملها مهندس البيانات

Remember the employees table

index	last_name	first_name	role	team	full_time	office
0	Thien	Vivian	Data Engineer	Data Science	1	Belgium
1	Huong	Julian	Data Scientist	Data Science	1	Belgium
2	Duplantier	Norbert	Software Developer	Infrastructure	1	United Kingdom
3	McColgan	Jeff	Business Developer	Sales	1	United States
4	Sanchez	Rick	Support Agent	Customer Service	0	United States

➤ SQL for data engineers

- Data engineers use SQL to create, maintain and update tables.
- Now let's create the Employee table using SQL:

```
CREATE TABLE employees (
  employee_id INT,
  first_name VARCHAR(255),
  last_name VARCHAR(255),
  role VARCHAR(255),
  team VARCHAR(255),
  full_time BOOLEAN,
  office VARCHAR(255)
);
```

- طيب خلينا نشرح ال statements اللي فوق:
- يستخدم CREATE TABLE لحتى انشي table بعدين بكتب اسم ال table اللي انا بدي اياه وبهاي الحالة اسم ال table حيكون employees
- بعدين ال 7 اسطر اللي بعدهم مقسومين لنصين، النص اللي عالشمال بكون اسم ال column اللي انا بدي اياه، وممنوع ب اسم ال columns يكون في spaces او فراغات او special characters او اشي من هاي الاشياء، اذا بدك تكتب كلمتين بتوصلهم عن طريق ال _ underscore
- اما النص اللي عاليمين هو نوع ال data type لهاد ال column مثلا بدي ال data type يكون integer ف بكتب INT او بدي اياه string ف بكتب VARCHAR() ويط جوا القوس تاها ال maximum NO. letters او بدي اياه Boolean ف بكتب BOOLEAN
- ممكن انت تسأل ليش ناعد بكتب هاي الشغلات بالكود ب upper case، ال SQL مش case sensitive ليهك انت كتبتها Boolean او BOOLEAN مش حيفرق، بس الطريقة المتعارف عليها انه ال reserved words بال SQL اكتبهم ب upper case عشان اميزهم عن اسماء ال columns وهاي الاشياء
- ممكن حدا يسأل ليش ال Boolean ما خليته integer، هو بزيبط هاد الحكي سينس انه القيمة رح تكون اما 1 او 0، ليهك ان حطيت int او Boolean مش حيفرق، بس ال Boolean بخزن بالذاكرة اقل، ليهك انت لما يكون عندك بيانات جدا ضخمة، بكون اشي مهم انه انت تقدر تخفف مس مساحة ال database قد ما بتقدر

➤ SQL for data scientists

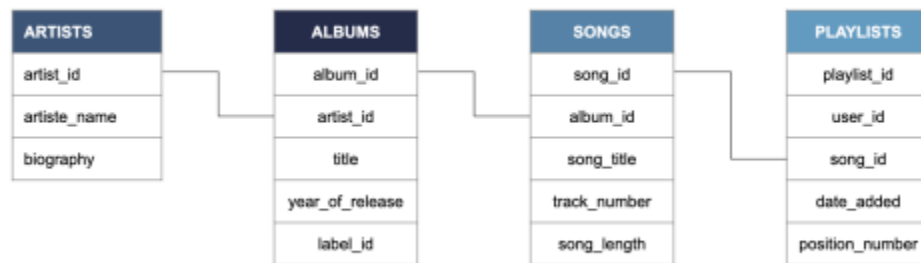
- Data scientists use SQL to query, filter, group, and aggregate data in tables.
- Now let's see an example of how to write a query to select the first and last name of the employees who have the word "Data" in their role:

```
SELECT first_name, last_name  
FROM employees  
WHERE role LIKE '%Data%'
```

- هون انا بدي احتار 2 columns اللي همة ال first name وال last name لهيك رح استخدم ال SELECT ستيتمينت، اللي هي رح تعطيني هдол ال columns
- طيب ال SQL ما بعرف هдол ال columns من وين يجيبهم، لهيك انت لازم تحكيه من اي table لازم يجيبهم، عن طريق ال FROM وبكتب بعدها اسم ال table
- اذا انا كان عندي شرط معين او بدي اطول فئة معينة او اشي بستخدم ال WHERE وبكتب بعدها اسم ال column اللي بدي اطبق عليه الشرط وهون انا مثلا بدي اطول الناس اللي ال role تاعهم مكتوب في كلمة Data، مش مهم اذا في بعدها او قبلها اشي لهيك انا رح اسخدم ال (%) percentile واذا حيطتها قبل الكلمة معناها انه مش مهم شو يكون قبلها، واذا حيطتها اخر الكلمة معناها مش مهم شو يكون بعدها وهكذا وبستخدم ال LIKE قبلها عشان تزبط عندي ال query

➤ Database schema

- Databases are made of tables
- The database schema governs how tables are related
- Schema means tables are related to each other



- ال ARTISTS, ALBUMS, SONGS, PLAYLISTS عبارة عن tables
- الخطوط التي بين ال tables هي الطريقة التي يربط ال tables بين بعض ف هي عبارة عن ال relationship between tables
- ال artist_id بال ARTISTS table عبارة عن ال primary key (مفتاح اساسي) وال artist_id بال ALBUMS table عبارة عن ال foreign key (مفتاح ثانوي)
- كل table لازم يكون فيها primary key بس مش لازم يكون فيها foreign key

➤ Several implementations (only for read):

▪ SQLite:

○ Advantages:

- It is a library, and the size of the library can easily be under 600 KB
- It acts as a complementary solution for enterprise RDMBS
- Open-source

○ Disadvantages:

- Lack of multi-user capabilities
- Can't deal with large datasets like Big Data

▪ MySQL:

○ Advantages:

- Support multi-user features
- Simple to install and use
- The ease of use and manageability makes it a great tool for websites
- One of the most popular open-source and large-scale RDBMS

○ Disadvantages:

- Lack of full-text search and slow concurrent read-writes
- Not work well with long-running SELECTs and is best suited to smaller SELECTs

▪ PostgreSQL:

○ Advantages:

- Open-source
- On top (compared to MySQL) when running long SELECTs
- The extensibility of the PostgreSQL database also makes it a perfect candidate for research and scientific projects.

○ Disadvantages:

- can be seen during frequent UPDATES, where due to no support for clustered indexes
- PostgreSQL can have a huge adverse impact on performance compared to MySQL databases.

▪ Oracle SQL:

○ Features of Oracle:

- Ease of data recovery when compare to databases
- The RDMS system can easily handle large amounts of data
- Allows you to change platforms at any time
- It can be used for read-write, reporting, testing, or backups, reducing the load on the primary database
- Support for hardware and OS-specific virtualization technologies

▪ SQL Server:

○ Features of Microsoft SQL Server:

- Support tools SQL Server profiler, BI tools, SQL Server Management Studio, and Database Tuning advisor
- Offers online support and documentation
- Display errors
- An activity monitor feature with filtering and automatic refresh
- Importing and Exporting from SQL Server Management Studio

For more information about SQLite, MySQL, PostgreSQL click [here](#)

For more information about Oracle SQL, SQL Server click [here](#)

Summary:

- SQL = industry standard
- Explain how Data Engineers and Data Scientists use it differently
- Database schema
- SQL implementations

❖ Data warehouses and data lakes:

➤ Data lake:

- Stores all the raw data
- Can be petabytes (1 million GBs)
- Stores all data structures
- Cost-effective
- Difficult to analyze
- Requires an up-to-date data catalog
- Used by data scientists
- Big data, real-time analytics



■ بنخزن فيها جميع انواع البيانات (structured, unstructured)

■ البيانات بتكون موجودة وجاهزة عندي بال data lake ما بروح اجيبها من مكان لانها اوريدي موجودة عندي
ولما بدي اعدل اشي عاليبيانات بعدلها عال data lake لانها البيانات الاساسية وكلشي بكون موجود فيها، ف لما بدي اعدل اشي بعدل عليها، لانه اذا عدلت ب مكان وما عدلت فيها بصير عندي اشي اسمه inconsistency يعني تناقض، لانه بزيطش اشي يكون قيمته غير بمكان ثاني

■ صعبة بعملية التحليل، لانه بكون عندي بيانات كتيرة و بجميع الانواع، لهيك صعب تحليلها

■ بحتاج data catalog ورح نشرح عنه اكثر كمان شوي

■ عالم البيانات اللي بتعامل مع ال data lakes

■ بما انه هاد النوع بيحمل جميع انواع البيانات ف رح يتكون عندي big data

❖ Data warehouse:

- Specific data for specific use
- Relatively small
- Stores mainly structured data
- More costly to update
- Optimized for data analysis
- Also used by data analysts and business analysts (BI)
- Ad-hoc, read-only queries



- يتكون بيانات مخصصة لغرض معين، يعني كل البيانات يكون في اشي مشترك بينهم ويكونوا متشابهين
- يتكون اصغر حجم من ال data lake، بس مش المقصود انه حجمها صغير يعني 5 MB او اشي ... لا حجكها صغير مقارنة بال data lake، ويمكن يكون حجمها بالتيرابايت عادي
- بتخزن فيها بس ال structured data واللي يكونوا نفس النوع
- يكون صعب علي اناي اعدل عليها
- مخصصة لتحليل البيانات
- محلل البيانات او اللي يشتغل ب مجال ال BI (Business Intelligence) بتعاملو معها
- Ad-hoc, read-only queries معناها انه هاي البيانات قيمتها محددة وخلص، ف انت لما تحلل البيانات بتحللها عن طريق ال read فقط، يعني انت بتقدر تشوف اللي بدك اياه من البيانات بس ما بتقدر تعدل عليها، لانه زي ما حكينا فوق، اذا بدك تعدل بتعدل عال data lake نفسها
- معلومة مهمة جدا، ال data warehouses دايمًا بتكون structured data، ممكن واحد يجي يحكي لي كيف بتكون structured وهو ممكن يكون فيها صور، على سبيل المثال صورة الالبوم، باجي بقلك هي ما بتحمل الصورة نفسها، هي بتحمل index للصورة، والصورة نفسها بتكون موجودة بال data lake، وبتكون الصور كلها مرتبة ب طريقة معينة جوا ال data lake وبرضه ب مساعدة ال data catalog بتقدر نحقق هاد الاشئ

➤ Data catalog for data lakes:

- What is the source of this data?
- Where is this data used?
- Who is the owner of the data?
- How often is this data updated?
- Good practice in terms of data governance
- Ensures reproducibility
- No catalog → data swamp

- ممكن نحكي عن ال data catalog زي ال meta data انه هو data about the data
- من وين مصدر هاي البيانات؟ وين بستخدموا هاي البيانات؟ مين مالك هاي البيانات
كل كم يتم تحديث البيانات؟ كل هاي الاسئلة مهمة ولازم تكون اجوبتها موجودة بال data catalog
- لازم اتأكد انه بقدر ارجع انتج او استخدم هاي البيانات، انه مش اذا حربت معي البيانات ما اقدر اجيبها كمان مرة
- اذا ما كان في data catalog رح يتكون عندي data swamp
وال data swamp من اسمه انه مستنقع للبيانات، يعني البيانات بتكون معفشكة ومش منظمة
واشي مش كويس ابدأ انه يكون عندي data swamp، والصورة التحت رح تشرحك الاشئ



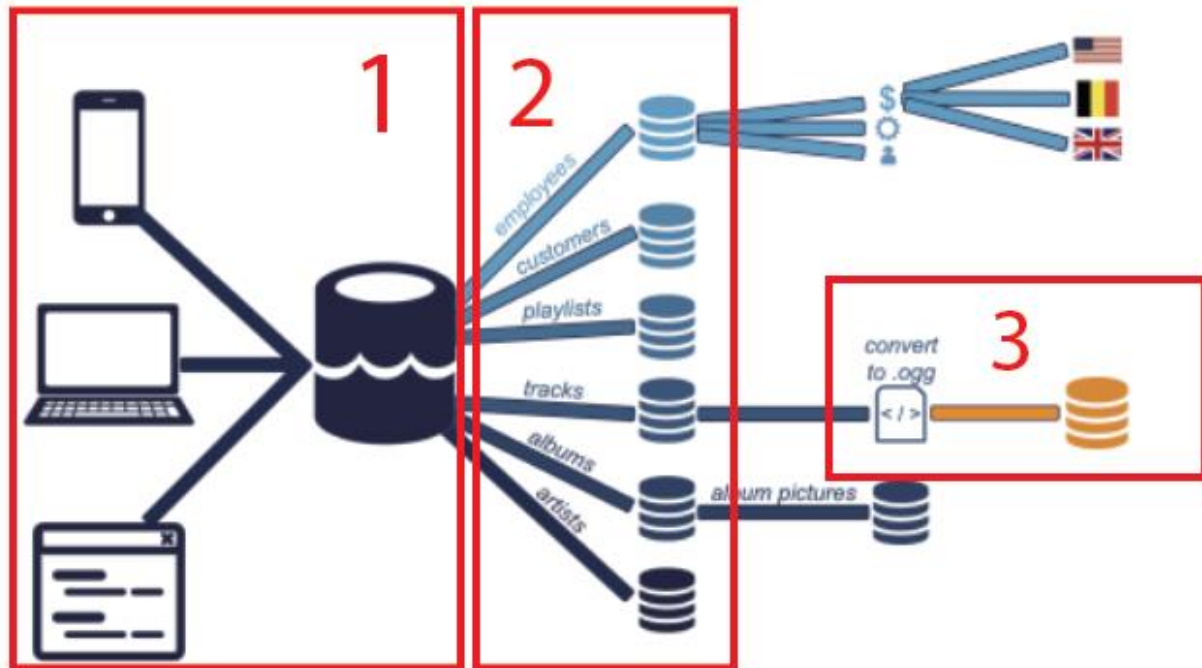
- Good practice for any data storage solution:
 - Reliability الموثوقية
 - Autonomy الحكم الذاتي
 - Scalability قابلية التوسع
 - Speed السرعة

➤ Database vs. data warehouse

- Database:
 - General term
 - Loosely defined as organized data stored and accessed on a computer
- Data warehouses are a type of database
- كلمة database لا يعني انه structured data، كلمة database يعني قاعدة البيانات بشكل عام يعني اي اشي بنخزن فيه البيانات بكون عبارة عن database والامثلة اللي اخدناها زي ال data lakes وال data warehouses هتدل عبارة عن انواع من ال databases

Summary:

- Data lakes
- Data warehouses
- Databases
- Data catalog
- Data swamp



- زي ما بنعرف انه احنا ب هاد الكورس رح نشغل و رح نفهم المادة من خلال Spotflix dataset وهلا رح نشرح هاي ال pipeline ونوضح اجزاءها
- رقم 1: بوضلي انه في بيانات عم تنلم من اكثر من مصدر وعم تتخزن ب مكان، هاد المكان هو ال data lake، لهيك زي ما حكينا انه ال data lake بكون فيها جميع انواع البيانات ومن مختلف المصادر
- رقم 2: بعد ما انا جمعت كل البيانات ب مكان واحد هلا اجا الدور اني اقسماها وارتبها، و زي ما حكينا قبل انه احنا بندرس ع spotflix dataset، و spotflix عبارة عن تطبيق ل سماع الاغاني، زيه زي spotify, apple music, anghami, sound cloud, ... etc ف هاي البيانات شو ما كانت اكيد بنقسمها ل عدة اشياء زي artist, albums, track, playlist, customer, employee ف اني اقسم البيانات تاكتي اهاي الاصناف وكل صنف اخزن بياناته ب مكان معين، هاد المكان اسمه data warehouse وزي ما حكينا انه ال data warehouse بكون مخصص ل اشي معين، وهاد اللي احنا عملناه قبل شوي
- رقم 3: على سبيل المثال الاغنية او ال track كان بصيغة flac. وانا بدو احوله ل صيغة ogg. هون ب هاي الحالة انا عملت transformation وخزنت البيانات اللي عدلت عليها ب data warehouse جديدة عشان اقدر اتعامل معها بالطريقة اللي انا بدو اياها، فالفكرة هون انا اذا بدو اعدل على اشي بخزنه ب data warehouse جديدة وبتعامل معه وهو هناك

Chapter 3 - Processing data

A general definition:

Data processing: converting raw data into meaningful information

Data processing value:

Conceptually	At Spotflix	Explain (شرح)
Remove unwanted data	No long term need for testing feature data	هون الفكرن نحذف البيانات اللي ما بنحتاجها وما بتعطينا فائدة، وكمثال عليها انه نحذف features معينة من data set تاعة spotflix
Optimize memory, process, and network costs	1. Can't afford to store and stream files this big 2. No need for a lossless format	بدي احاول اخفف من ال memory قدر الامكان ليهيك اذا كان في فايلات كبيرة مش راح اقدر اتحمل تكلفة تخزينها واذا كان في عندي بيانات مش مهمة بشيلها برضه عشان اخفف عال memory برضه
Convert data from one type to another	Convert songs from .flac to .ogg	احول البيانات من نوع ل اخر، زي اني اغير ال songs من امتداد .flac ل .ogg
Organize data	Reorganize data from the data lake to data warehouses	بدي ارتب البيانات تكون organized زي كمثال اني اطلع من ال data lake اطلع data warehouses وزي ما حكينا قبل انه ال data warehouses دائما بتكون organized

To fit into a schema/structure	Employee table example	لازم لما ادخل البيانات ادخلها ب ترتيب ال schema تاعة ال database، على سبيل المثال ال employee table ال الترتيب لازم يكون فيها last name بعدين first name بعدين role بعدين team... الخ
Increase productivity	Enable data scientists	بدي ازيد فعالية هاي البيانات واخلوها مفيدة قدر الامكان، ولما اعمل انا هيك ف انا عم بمكن ال data scientist وبساعده ب توفير بيانات مفيدة ومهمة

How Data Engineers process data:

A. Conceptually: Data manipulation, cleaning, and tidying tasks:

1. that can be automated. 2. that will always need to be done

At Spotflix: 1. Rejecting corrupt song files. 2. Deciding what happens with missing metadata

- لازم اتلاعب بالبيانات واعملها cleaning وهاد الاشئ ممكن يصير ب اكثر من طريقة: اشطب وامحي الملفات المضروبة واتخلص منها برضه اقرر شو لازم اعمل بال missing values، هلا اعييهم ولا امحيهم وذا اعييهم اعييهم ب ايش؟ ال mean ولا median ولا mode وكل هاي الاشياء لازم نقرر ها عشان تكون البيانات مفيدة

B. Conceptually: Store data in a sanely structured database

At Spotflix: Separate artists and albums tables...

- لازم اخزن البيانات بشكل سليم على شكل structured databases، على سبيل المثال افصل ال artists وال albums tables

C. Conceptually: always need to be Create views on top of the database tables
At Spotflix: ...but provide view combining them

- بدي اعمل views وال view ب عالم ال databases هو عبارة عن ال table نفسها بس الفرق اني ما بقدر اعدل على البيانات فيها، وغير هيك بقدر كام اعمل view بتضم مجموعة columns من table ومجموعة columns من table ثانية، ههاد الاشئ كثير يساعد انه نقدر نشوف البيانات بسهولة ومن دون الخوف انه نشطب او نعدل عالبيانات بالغلط

D. Conceptually: Optimizing the performance of the database
At Spotflix: Indexing

- لازم احسن من اداء قاعدة البيانات وهاد الاشئ يتم ب اكثر من طريقة منها ال indexing، ومعنى ال indexing فهو ممكن نحكي انه زي الفهرس انه كيف يكون مقسم بترتيب الاحرف الابجدية انه (أ،ب،ت،..الخ) وهون نفس الاشئ، انه اذا انا بدي ادر على اشئ معين ما ادور عليه بين كل البيانات، لا يكون عامل indexing وبصير يدور عليه بال index اللي هو فيه وهاد الاشئ بسرعه عملية البحث وبحسن اداء ال database

Batch and Stream processing:



The difference between Batch and Stream processing:

▪ Batch processing (دفعات):

- Ad-hoc or Scheduled processing
- Collection of data

- ال batch processing طريقة ال process فيه بتكون على شكل حزم شو يعني حزم اصلا؟ يعني دفعات دفعات، على سبيل المثال انت كل ساعة رح تيجيك مجموعة من البيانات اللي لازم تعملها processing ف هون انا حكيت كل ساعة حيجيك مجموعة، ف البيانات بتيجي على شكل دفعات والعملية بتكون منظمة

- من اشهر الامثلة عليها Apache Spark, Hadoop

▪ Stream processing (مستمر):

- Real-time processing
- Continuous data

- ال Stream processing طريقة ال process فيه بتكون مستمرة ومتواصلة بوتصلها جارية كيف يعني؟ يعني على سبيل المثال انه انا ممكن بعد دقيقة تيجيني شوية بيانات، وبعد كمان دقيقة تيجيني بيانات مضاعفة، ف انا هون بعمل process اول ب اول، يعني كل ما تيجيني بيانات او تيجيني معلومة، ف عالسريع بعملها processing وهاي هي العملية بال stream processing

- ومن اشهر الامثلة عليه Apache Storm, Flink

- Summary:

- What data processing is
- Why it's necessary
- What it consists in
- How we process data at Spotflix

Scheduling:

- Can apply to any task listed in data processing
 - Scheduling is the glue of your system
 - Holds each piece and organize how they work together
 - Runs tasks in a specific order and resolves all dependencies
- بتقدر تعمل اي عملية على البيانات خلال تجهيزها، على سبيل المثال data cleaning, delete corrupted data, etc
- عملية ال scheduling او الجدولة هي عبارة عن الالية او الطريق او ال process كاملة اللي رح يشتغل عليها ال system تبعك
- لما تكون ال data اللي عندي كتير كبيرة ف ما بقدر اني اتعامل معها بسهولة، مثلا اذا بدى اطلع ال mean لعمود معين ما بقدر اطلعه بسهولة، لازم اقسم ال data تاعتي ل اجزاء وبعمل الية معينة بحيث انه اعمل عملية على كل جزء واقدر بالاخر بس اخلص منهم كلهم اوصل للمعلومة اللي بدى اياها
- بنجري العمليات بترتيب معين، يعني مثلا انا ما بقدر اطلع ال mean ل عمود معين وهاد العمود فيه corrupted data او فيه null values لهيك انا لازم اتعامل مع ال corrupted او ال null values اول، بعدين اطلع ال mean ، لهيك بنسنتنتج انه اشي كتير مهم اني امشي على ترتيب معين، لازم ال system تاعنا يكون مبني على tasks الها order معين وبحل جميع المشاكل
- من اشهر الادوات او ال tools اللي بتعمل scheduling لل data تاعتك، هو Apache Airflow

Manual, time and sensor scheduling:

- Manually
Example: Manually update the employee table
- Automatically run at a specific time
Example: Update the employee table at 6 AM

- Automatically run if a specific condition is met → Sensor scheduling
Example: Update the department tables if a new employee was added

- بشكل يدوي بغير عاليات، او بشكل اوتوماتيكي ب وقت معين، او بشكل اوتوماتيكي لما يتحقق شرط معين

Batches and Streams:

- Batches:
 - Group records at intervals
 - Often Cheaper
- Streams:
 - Send individual records right away

- ال batches يكون اسهل من ال stream لانه بال batches البيانات already بتكون عندك ف انت عارف كيف بدك تتعامل معها، اما ال stream ف الداتا بتكون record by record ف انت ما بتعرف شو طبيعة البيانات اللي رح تيجيك

Scheduling tools



- Summary:
 - What scheduling is
 - Different ways to set it up
 - Difference between batches and streams
 - How Scheduling is implemented at Spotflick
 - Airflow, Luigi

Parallel Computing:

- Basis of modern data processing tools
- Necessary:
 - Mainly because of memory
 - Also for processing power
- How it works:
 - Split tasks up into several smaller subtasks
 - Distribute these subtasks over several computers

- اشي مهم واساسي في ادوات معالجة البيانات، وهي ضرورية لانه بعض العمليات بتحتاج مساحة ذاكرة كبيرة جدا وبرضه بسبب انها قوية وفعالة في عملية معالجة البيانات، وبتتم هاي العملية عن طريق تقسيم المهمة ل اجزاء اصغر وبعدين بنوزعهم وبنعملهم distribute على عدة اجهزة

- رح نشرح عن ال parallel computing او الحوسبة المتوازية باختصار، المبدأ تاعها انه اقسم تاسك معين باخد معي وقت طويل ل اكثر من جزء وكل جزء احطه ب جهاز لحال او system لحال عشان اسغل الوقت وانجز ال task بوقت اقل، هالأ مش كل ال tasks بزبط اني اعملهم parallel computing على سبيل المثال:

انا عندي 10 مليون record وبدي اطلع ال mean تاعهم، هالأ اذا بدي اقسم 10 اجزاء ل 10 اجهزة ف كل جهاز حياخد مليون، واذا بطلع لكل جهاز ال mean ف هون العملية بتصير غلط عندي، يعني اذا كل جهاز طلعت ال mean لل داتا اللي عنده وبعدين اني اطلع ال mean لل means ف هاد اشي غلط رياضيا، ليهك هاد ال task ما بزبط اعمله parallel بشكل دايركت، بس ممكن يزبط معي هاد التاسك بطريقة تانية، هي اني لكل جهاز اطلع المجموع او ال sum للبيانات تاعتهم وبعدين بجمع كل ال sums وبعدين باخد ال mean تاعهم عن طريق اني اقسم المجموع كامل تاعك كل الاجهزة على 10 مليون.

مثال ثاني: انا عندي 10 الاف بحث، بدي اطلع كل كلمة كم من مرة تكررت بكل الابحاث، ف اللي بعمله اني بعطي كل جهاز جزء من الابحاث، وكل جهاز بدي اياه يطلع كل كلمة كم من مرة تكررت، وبعدين بجمع نتائج كل الاجهزة وبعمل aggregation ل الهم، وبعدهم ب dictionary واحد بحمل النتائج كاملة والجواب اللي انا بدي اياه

- مش دائما ال parallel computing يكون الحل لكل اشي، على سبيل المثال، في بعض الاشياء او ال tasks ما بتحتاج parallel ولو عملناها parallel ف بالعكس بتصر توخذ وقت اطول، يعني اذا كان في عندي مشكلة صغيرة او task و هاد التاسك already صغير ف ما بتحتاج اني اعمله parallel computing ولو عملته ف رح يوخذ مني وقت اطول لانه لما اقسم المشكلة الصغيرة ل مشاكل اصغر وابعتها ل جهاز، ف هاد الجهاز بده يعرف كيف يتعامل معها وهاي العملية اصلا بتوخذ وقت، ف ما في داعي اني احل مشكلة صغيرة بال parallel computing
- وكم ان شغلة، انه مش دائما اذا زدت عدد الاجهزة اللي رح اعطيها subtask ف هاد اشي احسن ... لا، مش ضروري، انت لازم توخذ وتستعمل اجهزة على قد حاجتك وعلى قد ما بتحتاج ال problem ل اجهزة، ف لازم يكون عندي balance بهاد الموضوع

Benefits and risks of parallel computing:

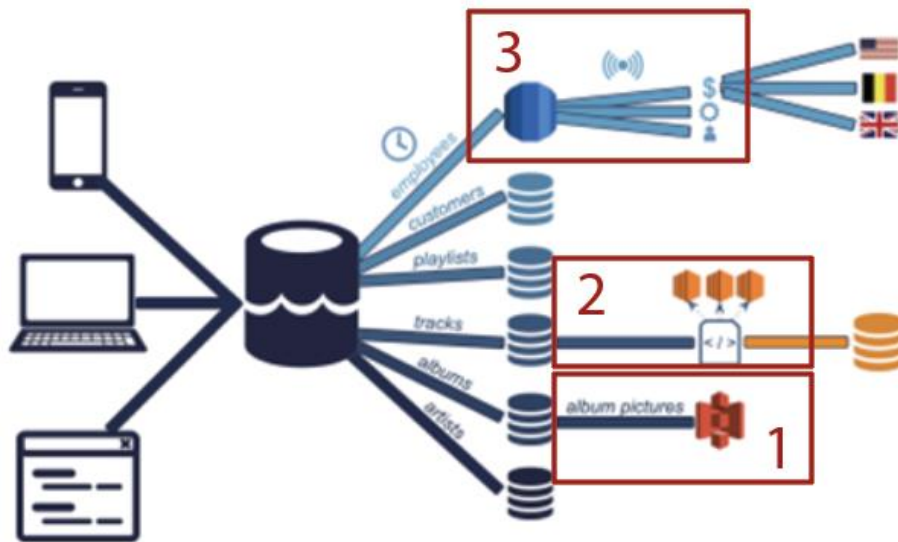
- Advantages:
 - Extra processing power
 - Reduced memory footprint
- Disadvantages:
 - Moving data incurs a cost
 - Communication time
- Summary:
 - Benefits and risks
 - How it's implemented at Spotflick

Cloud Computing:

- Servers on premises (الجهاز اللي عندك بالبيت او الشغل):
 - Bought
 - Need space
 - Electrical and maintenance cost
 - Enough power for peak moments
 - Processing power unused at quieter times
- Servers on the cloud (الموجود عالنت):
 - Rented
 - Don't need space
 - Use just the resources we need
 - When we need them
 - The closer to the user the better

Cloud Computing for data storage:





- ب رقم 1 احنا هون اخدنا صور ال albums (مش ال albums نفسهم) وخرنناهم ب AWS S3 اللي هو ال cloud file storage الخاص ب امزون

- ب رقم 2، مش احنا حكينا قبل اذا انا بدي اعمل عمليو transformation للبيانات مثلا بدي احول امتداد ال track من flac ل ogg. هاي العملية بقدر اعملها عال cloud وهون احنا استخدمنا AWS EC2 اللي هو ال Computation الخاص ب امزون

- ب رقم 3 خزننا ال employee table ب AWS RDS اللي هون Database خاص ب امزون

- Multicloud:

○ Pros (ايجابيات):

- Reducing reliance on a single vendor
- Cost-efficiencies
- Local laws requiring certain data to be physically present within the country
- Militating against disasters

- Cons (سلبيات):
 - Cloud providers try to lock in consumers
 - Incompatibility
 - Security and governance
- Summary:
 - Benefits and risks of cloud computing
 - How it is implemented at Spotflick
 - Can cite the main cloud providers
- كل القبل كان عبارة عن introduction للمادة وعن ال data engineering بشكل عام
وهلا رح نبليش تطبيق عملي عالموضوع ونمشي بال pipeline اللي حكينا عنه كامل
- **What you learned – Chapter 1:**
 - What Data Engineering is
 - How important it is
 - How data engineers differ from data scientists
 - What a data pipeline is and how it works
- **What you learned – Chapter 2:**
 - The different structured data can take
 - How fundamental SQL is
 - The difference between data lakes, data warehouses, and databases
- **What you learned – Chapter 3:**
 - How data is processed
 - How scheduling holds it all together
 - Parallel Computing
 - Cloud Computing
- **And some more:**
 - What SQL code actually looks like
 - Main tools and technologies used in data engineering
 - And some more

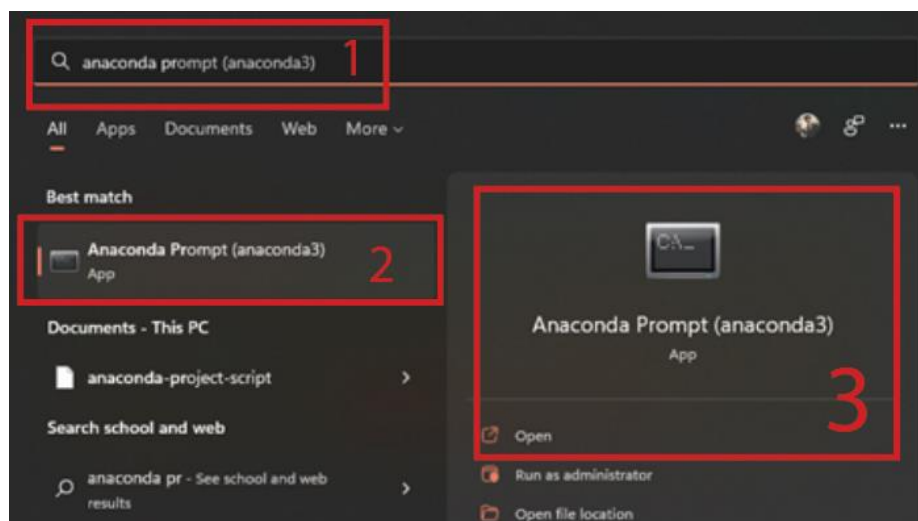
The Practical Part (الجزء العملي):

```

1 from faker import Faker
2 import csv
3
4 output = open("data.csv", "w")    # Make a file called "data"
5                                   # with csv format to write inside it
6
7 fake = Faker() #Make object from Faker library
8
9 header = ['name', 'age', 'street', 'city', 'state', 'zip', 'lng', 'lat']
10
11 mywriter = csv.writer(output) #Object for writing inside the output
12 object (data.csv)
13 mywriter.writerow(header) #Write the first row "COLUMNS NAMES"
14
15 for r in range(1000): # Making 1000 record dataframe
16     mywriter.writerow([fake.name(),
17                         fake.random_int(min=18, max=80, step=1),
18                         fake.street_address(), fake.city(),
19                         fake.state(), fake.zipcode(),
20                         fake.longitude(), fake.latitude()])
21
22 output.close() #Close the file

```

- طيب، خنبش نشرح الكود، ب اول سطرين عملنا import لل packages اللي رح نستخدمهم واللي همة: faker عشان نجيب بيانات وهمية ونتدرب ع انه كيف نتعامل معها، و csv عشان نعمل ملف ب امتداد csv ونخزن فيه هاي البيانات عشان نقدر نتعامل معه
- ال faker library ما بتكون نازلة built in من anaconda ليهيك انت لازم تنزلها، وفي كثير طرق للتنزيل ومنها:



- افتحوا ال anaconda prompt واكتبوا `pip install faker`

```

Anaconda Prompt (anaconda3)
(base) C:\Users\basel>pip install faker
Requirement already satisfied: faker in c:\users\basel\anaconda3\lib\site-packages (13.3.4)
Requirement already satisfied: python-dateutil>=2.4 in c:\users\basel\anaconda3\lib\site-packages (from faker) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\basel\anaconda3\lib\site-packages (from python-dateutil>=2.4->faker) (1.16.0)
(base) C:\Users\basel>_

```

بطلعكم انه بلش ينزلها، انا هون طلعتي هيكلاني already منزلها

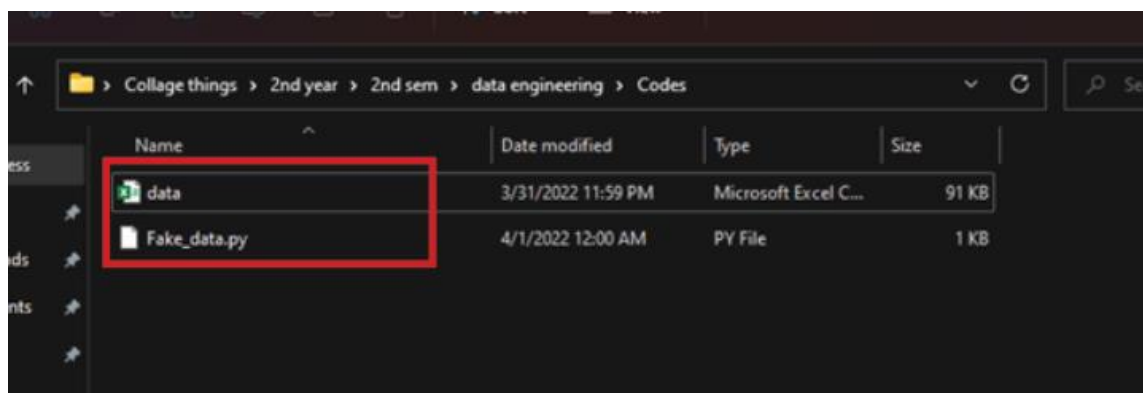
- نرجع للكود، بسطر 4 عملنا متغير او object عبارة عن file، واحنا اخدنا انه عشان نتعامل مع ال files بنستخدم open statement وبنحط جواها اسم الملف، وطريقة تعاملتي معه هون اسم الملف حيكون data.csv وطريقة تعاملتي معها حيكون write وبحطها بالكود w
- سطر 7 عملنا object من ال library faker اسمه fake
- سطر 9 عملنا list ب اسماء ال columns اللي انا بدي احطهم بال داتا تاعتي
- سطر 11 علمنا متغير او object هاد ال object شغلته انه يكتب عالفيل اللي سميناه data، كيف خليناه يعمل هيكل اشي؟ عن طريق استخدمت ال library csv واخذت منها ال method اللي اسمها writer ومن اسمها انه وظيفتها انها تكتب عالملفات، وبحط جوا الاقواس اسم الملف اللي انا بدي اكتب فيه او احط فيه البيانات
- سطر 13 علمنا اول عملية كتابة عن طريق المتغير mywriter واستخدمنا منه ال function اللي اسمه writerow ومن اسمه انه يكتب سطر كامل، واول سطر كتبناه اللي هو اسماء ال columns او المتغير اللي اسمه header او ال list اللي حطينا فيها اسماء الاعمدة، ف هيكل احنا كتبنا اول سطر باملف
- سطر 15 عملنا loop حتلّف 1000 مرة لانه انا بدي اعمل الف record او الف سطر من البيانات
- سطر 16-20 جوا ال for loop عملنا انه يكتب سطر، وهاد السطر حيكون عبارة عن بيانات عشوائية او fake من ال library نفسها، ومشينا بنفس ترتيب ال header، يعني اول اشي بال header كان name ف حطينا اول اشي name وهكذا
- ملاحظة 1: انا ممكن ب اسم ال header اكتب zip بس لما اجي بدي استخدم هاي ال library وبدي اياها تعطيني بيانات fake ف لازم اكتب اسم ال function صح، يعني بكتبه fake.zipcode()
- ملاحظة 2: في اكثر من اشي بتقدر تستفيده من ال faker library، يعني فيها اشي غير ال name وال city والخ واذا حابين تتعرفوا اكثر بتقدر تقرأوا ال [documentation](#) او بتقرأوا تلخيصها على [GitHub](#)
- اخر اشي ب سطر 22 سكرنا الملف
- على سبيل المثال انتوا سيفتوا هاد الملف تاع البايثون ب C:\python_code وبعدين عملت Run للبرنامج اللي كتبناه فوق، هلاك output عندك ما رح تلاقى، لانه ال output تبعنا ب هاد الكود مش عبارة عن print او هاي الاشياء، لأ، هو عبارة عن ملف csv file اسمه data، طيب وين بلاقيه؟ بتلاقيه بنقش الملف اللي سيفت فيه script البايثون

```

C:\Users\basel\Desktop\Collage things\2nd year\2nd sem\data engineering\Codes\Fake_data.py
1 from faker import Faker
2 import csv
3
4 output = open("data.csv", "w") # Make a file called "data"
5                                # with csv format to write inside it
6
7 fake = Faker() #Make object from Faker libraray
8

```

- هي ملف البايثون مع ملف ال data.csv اللى احنا عملناه ب نفس ال file



- الملف اذا فتحتة لازم يعطيك زي هيك:

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format. Don't

	A	B	C	D	E	F	G	H	I
1	name	age	street	city	state	zip	lng	lat	
2									
3	Lynn Steele	72	6887 Stephanie Mission	New Richardton	Arkansas	21637	-9.30071	75.0785535	
4									
5	Joshua Jackson	76	0056 Joshua Ford	Loweport	Alabama	28017	173.003379	-9.2861105	
6									
7	Kelly Swanson	48	13719 Rogers Track Apt. 113	Weberland	Tennessee	20768	-127.560336	60.5957805	
8									
9	George Pitts	29	39325 Jeffery Lodge	New Felicia	Minnesota	79086	141.909261	58.3171775	
10									
11	Melissa Taylor	51	098 Marcia Garden	Ronaldfurt	Illinois	56944	117.142448	88.0030715	
12									
13	Amanda Cameron	59	978 Miller Estates Suite 044	West Tanya	Colorado	30342	-110.107392	86.405711	
14									
15	Angela Mills	30	03580 John Curve Suite 630	New Wesley	Kentucky	22806	42.485803	-68.0597585	
16									
17	Cameron Adams	51	791 Stevenson Meadows Suite 127	Lake Lisa	Rhode Island	77919	116.356987	-11.39933	
18									
19	Jenna Brown	38	6058 Pierce Hill	Amandamouth	New York	80692	-144.049845	39.6967015	
20									

- هلاً احنا اخدنا كيف نكتب ونتعامل مع ال csv files وهلاً رح نوخذ كيف نكتب على ال JSON files

```

1 from faker import Faker
2 import json
3
4 output = open("data.json", "w")
5
6 fake = Faker()
7
8 alldata = {} #Empty dictionary
9 alldata['records'] = [] # Make empty list as a
10                        # value for the "records" key
11
12 for x in range(1000):
13     data = {"name":fake.name(),
14            "age":fake.random_int(min=18,max=80,step=1),
15            "street": fake.street_address(), "city":fake.city(),
16            "state":fake.state(), "zip":fake.zipcode(),
17            "lng": float(fake.longitude()),
18            "lat": float(fake.latitude())}
19     #filling dicionary with values
20
21     alldata['records'].append(data)
22     """append the dictionary to the empty list
23     as a value to the key "recrods" """
24
25 json.dump(alldata, output) #To make the file in JSON format

```

- اول 7 اسطر عارفين شو يعملوا، الفرق الوحيد انه اسم الملف حيكون ب امتداد json لانني انا بدي اعمل json file وبرضه ال library json بتكون نازلة already من anaconda

- سطر 8 عملنا empty dictionary لانه الهيكله تاعه ال json files عبارة عن dictionary ال json file بكون dictionary بشكل كامل، بعددين اول key بكون اسمه records وهاد ال key ال value تاعته بتكون عبارة عن list هاي ال list عبارة عن كل ال records كل record بكون عبارة عن dictionary يعني ب اختصار هي بكون { records : [dictionary, dictionary, dictionary] }

- ب سطر 9 عملنا اول key بال dictionary واسمها records زي ما حيكلنا وال value تاعته عبارة عن list فاضية وهلاً رح ابلش اضيف عليها

- عملنا for loop ب 1000 مرة لانني انه بدي اعمل الف record

- ب كل لفه عم يعمل dictionary وبسميه data وبحط فيه بيانات fake او random، بس طريقة التعباي بتكون على شكل key و value لانه بضلّه dictionary وال key بده يكون اسم الاعمدة وال value هو الاشئي ال random من ال library

- في شغلة ما شرحناها بالمثال فوق هي شغلة ال random_int، بكل بساطة بدي رقم random تكون اقل قيمة اله 18 واعلى قيمة 80 وال step بدي اياها 1، يعني بدي تعطيني قيمة نكتن عبارة 21.5 مثلاً، لأ خلص بدي اياهم integers
- سطر 21 بدي اضيف هاد ال dictionary اللي اسمه data بال list على شكل value لل records ويعمل هاي العملية ب كل لفه ، لانه كل لفه عبارة عن record واحد
- واخر اشي سطر 25 عملنا سيف لعاد الملف على شكل او ال format تاعة ال json files عن طريق dump statement ويعدين بحط جوا الاقواس ال data اللي عندي، وبحالتنا هاي اسمها alldata ويعدين بكتب وين بدي اخزنها، انه ب اي ملف وبحالتنا اسمه output اللي اصلا اسم الملف اللي فيه data.json
- هلا بعمل سيف لملف البايثون زي قبل وبعمل run للبرنامج تاعي ، برضه بلاقي الملف اللي عملته موجود ب امتداد .json. ب امكانك تفتحه عن طريق (for example) notepad → open with → right click ويعدين المفروض يعطيك اشي زي هيك

```

{"records": [{"name": "Karen Scott", "age": 38, "street": "354 Haas Brooks Apt. 316", "city": "Martinshire", "zip": "23860", "lng": 47.625563, "lat": 74.484342}, {"name": "Brandon Vargas", "age": 59, "street": "263 Choi Sprin", "city": "Walterview", "state": "Massachusetts", "zip": "25769", "lng": -87.722868, "lat": 49.8322445}, {"name": "hanson", "age": 79, "street": "827 Hart Rapids", "city": "North Richardshire", "state": "Illinois", "zip": "85", "lng": 74, "street": "456 Rowland Flats Suite 026", "city": "Bradleyhaven", "state": "Wisconsin", "zip": "18777", "lng": 21.705211, "lat": 45.6993495}, {"name": "Lori Wells", "age": 74, "street": "Harbor Apt. 808", "city": "Walkerfort", "state": "Nevada", "zip": "36732", "lng": -40.068526, "lat": -20.831}, {"name": "Parker Center Suite 528", "city": "Gordonside", "state": "Massachusetts", "zip": "17059", "lng": 110.743069, "age": 79, "street": "02625 Michele Mount Suite 990", "city": "South Charles", "state": "Tennessee", "zip": "11.664212", "lat": 11.2866205}, {"name": "Donna Jefferson", "age": 19, "street": "169 Zachary Islands", "city": "y", "state": "Maryland", "zip": "63130", "lng": -133.101282, "lat": -51.462045}, {"name": "Amber Thompson", "age": 74792, "street": "April Court Suite 085", "city": "Port Matthewfort", "state": "Nevada", "zip": "91093", "lng": 17, "name": "Hannah Chung", "age": 55, "street": "00664 Courtney Tunnel", "city": "East Jasminside", "state": "South", "zip": "2058", "lng": 31.485903, "lat": -11.5532525}, {"name": "Joseph King", "age": 67, "street": "887 Eric Viaduct", "state": "Nevada", "zip": "65943", "lng": 6.146211, "lat": 39.786421}, {"name": "Ashley Martinez", "age": 76, "street": "Port Matthewbury", "state": "Texas", "zip": "49594", "lng": 174.614928, "lat": -31.97483}, {"name": "Angel", "age": 6880, "street": "Chad Fords Apt. 160", "city": "North Julieview", "state": "Delaware", "zip": "36444", "lng": 6.7020}, {"name": "Kathryn Blake", "age": 25, "street": "133 Smith Station Suite 904", "city": "New Jameschester", "state": "104.168797", "lat": 44.847116}, {"name": "Eduardo Butler", "age": 40, "street": "0935 Kathleen Junction", "city": "Tracyport", "state": "Kentucky", "zip": "33193", "lng": -11.159245, "lat": -76.2166165}, {"name": "Leroy Coe", "age": 18, "street": "24050 Sanchez Mews Suite 723", "city": "Kennethberg", "state": "Texas", "zip": "43521", "lat": 18.

```

Read CSV file:

```

1 import csv
2 import numpy as np
3
4 with open('data.csv','r') as f:
5     myreader = csv.DictReader(f)
6     header = next(myreader)
7
8     # to find min, max, avg, No. people with age > 40, No. people with age < 20
9     counter = 0
10    age_list = []
11    for row in myreader:
12        if counter %2 ==0:
13            age_list.append(int(row['age']))
14            counter += 1
15
16    age_arr = np.array(age_list)
17
18    print("Minimum age: ", min(age_arr)) # Min
19    print("Maximum age: ", max(age_arr)) # Max
20    print("Average age: ", np.mean(age_arr)) # Avg
21
22    ageGreaterThan40 = age_arr[age_arr > 40] # Age > 40
23    print("No. people with age greater than 40: ", len(ageGreaterThan40),
24    "people")
25
26    ageLowerThan20 = age_arr[age_arr < 20] # Age < 20
27    print("No. people with age lower than 20: ", len(ageLowerThan20) , "people")
28
29 """
30     Output:
31     Minimum age:  18
32     Maximum age:  80
33     Average age:  48.898
34     No. people with age greater than 40:  625 people
35     No. people with age lower than 20:  29 people
36 """

```

- طيب، اول سطرين عملنا import لل packages اللي بنحتاجهم، وبسطر 4 قرأنا الملف (طبعا هاد الملف هو نفسه اللي احنا عملناه قبل عن طريق ال faker باكيج
- سطر 5 عملت object اسمه myreader وحطيت فيه كل بيانات الملف f (طبعا هاد الملف f هو نفس ملف ال data.csv بس انا سميتاه f (عن طريق as f) عشان اصير اوصله بسرعة زي هون، عن طريق csv.DictReader(file_name)، هالأ هاد الفنكشن بحطلي كل البيانات جوا هاد ال variable بس كل record بحطه كل dictionary

- سطر 6 بحكي انه مش ملف ال csv فوق اول اشي او عمود فيه يكون اسماء ال columns؟ لهيك انا بدي اخذ هذول الاسماء واحطهم ب variable عن طريق انا اعمل (next(myreader) ، هاد الفنكشن بعتيني اول سطر، واذا عملت كمان مرة (next(myreader) ف بعتيني ثاني سطر .. وهكذا
- طيب مثلاً علينا تاسك معين انا اطلع ال min, max, avg وعدد الاشخاص اللي عمرهم اكثر 40، وعدد الاشخاص اللي عمرهم اقل من 20 ف شو لازم اعمل:
- مبدأياً بدي احط كل هاي الاعداد جوا ليست، بس كيف؟ بدي الف على كل record يعني على كل dictionary موجود ب myreader واطول من ال value تابعة ال age واحطها جوا ليست
- ف انا بسطر 9 و 10 عملت counter وهلا بحكيلكم ليش عملته، وعملت ليست فاضية عشان احط فيها الاعداد
- بسطر 11 لفيت على كل record موجود ب myreader
- هلا الفكرة انه انا حطيت كل القيم رح يكون عندي قيمة فيها عمر وقيمة فيها اشي فاضي، هاد الاشياء الفاضية انا ما بدي اياه، وعندي اذا فتحت الملف رح تلاقى انه record فيو بيانات وال record اللي بعده فاضي (هلا هاد هيكل هو عندي، ممكن انت ما يكون عندك هاي الفراغات) ف عشان انا ما اخذ هاي الفراغات بدي اياه ينط عن كل ريكورد فردي، وانا بالكود حكتله اذا ال counter كان زوجي ف حطلي قيمة ال age جوا ال ليست، غير هيكل لا تعمل اشي، وبعدين اخر اشي جوا ال for loop بزيد ال counter بواحد
- عملت append لل ليست الفاضية اللي عندي لل value تابعة ال age اللي موجود ب هاد ال record وبعدين حولته ل int لانه القيمة لل age عبارة عن string ف انا بدي احوله ل int عشان اقدر اطول منه المعلومات اللي بدي اياها
- ب سطر 16 حولت هاي ال ليست ل array عشان اسهل عحالي بعض العمليات
- ب سطر 18 و 19 و 20 طلت ال max, min, avg بسهولة (تذكر بطول ال avg عن طريق ال mean (method
- سطر 22 عملت array فيو بس القيم اللي اعلى من 40، ببعدين طبعتهم
- سطر 25 عملت array في بس القيم اللي اقل من 20، ببعدين طبعتهم
- واخر اشي بالكود هاد عبارة عن comment فيه ال output اللي لازم يطلع معك

Read JSON:

```

1 import json
2
3 with open('data.json','r') as f:
4     data = json.load(f)
5
6 # print(type(data)) # Dict
7 # print(data) # All the records
8 print("first records \n" , data['records'][0], "\n")
9
10 # find min, max, avg
11
12 age_list = [int(record['age']) for record in data['records']]
13
14 print("Minimum age: ",min(age_list))
15 print("Maximum age: ",max(age_list))
16 print("Average age: ", sum(age_list) / len(age_list))
17
18 """Output:
19     first records
20     {'name': 'Karen Scott', 'age': 38, 'street': '354 Haas Brooks Apt. 316',
21 'city': 'Martinshire', 'state': 'Colorado', 'zip': '52135', 'lng': 116.733657,
22 'lat': 17.106047}
23
24     Minimum age:  18
25     Maximum age:  80
26     Average age:  48.988
27 """

```

- هلاً بدنا نعمل نفس الاشئ بس من ملف json ، اول 3 اسطر بنعرف شو بعملوا
- السطر 4 حطيت كل البيانات اللي بالملف بال data variable عن طريق الفنكشن json.load
- ال type لل data variable هو dictionary
- لو بدى اطبع ال data variable ف هو رح يطبعلي كل الملف
- اذا بدى اطول اول record ف بطوله عن طريق data['records'][0] زي ب سطر 8، لانه البيانات او ال records هي عبارة عن value لل key اللي اسمه records ف عشان اوصل لل values تاعونه بعمل data['records'] ، ف هلا ال value عبارة عن ليست ، ف اذا بدى اطول اول record بعمل [0] لانه اول record هو عبارة عن اول عنصر ب هاي الليست، لانه ال value تاعة ال records عبارة عن list وكل item او عنصر فيها هو عبارة عن dictionary وكل dictionary هو عبارة عن record (ركزوا بالكلام اللي قبل عشان تفهمو كويس)
- اخر اشئ حطيت الاعداد كلهم ب ليست عن طريق اني انا اخذ ال value تاعة ال age واحولها ل int وال ف عكس ال records اللي بال data variable ، بعدين طلعتا القيم اللي هم min,max,avg زي ما عملنا قبل

Lab 1 (Assignment):

1. Convert the JSON FILE (data1.json) to CSV FILE (data1.csv)
2. Convert the CSV FILE (data2.csv) to JSON FILE (data2.json)
3. Load the data1 and data2 to df1, df2 dataframes respectively, using numpy as pandas libraries
4. print the number of the records in df1, and df2
5. print the average of the salary attribute in the df2
6. Normalize the age attribute in df2 using z-score normalization, and replace the original one with the normalized one.

- Download [data1.json](#)
- Download [data2.csv](#)

The Solution:

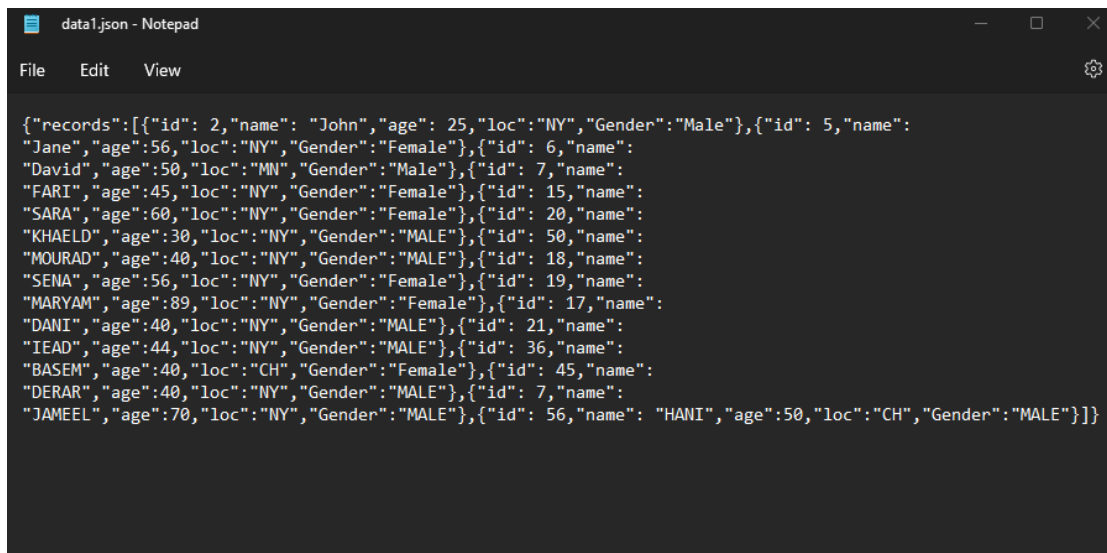
```
1 # 1) Convert the JSON FILE (data1.json) to CSV File (data1.csv)
2
3 import json, csv #import json and csv libraries
4
5 with open("data1.json", 'r') as json_file_1:
6     data = json.load(json_file_1)
7
8 records = data['records'] #Taking the value for the 'records' element
9
10 with open("data1.csv", 'w') as csv_file_1:
11     writer = csv.writer(csv_file_1) # object for writing inside the file
12     record = records[0] #Taking the first element "first record"
13     writer.writerow(record.keys()) #and write the keys for that element, instead
14 of writing (['id', 'name', 'age', etc])
15     for record in records: #looping inside all the records record by record
16         writer.writerow(record.values()) #writing every record
17
18
19
20 # 2) Convert the CSV FILE (data2.csv) to JSON FILE (data2.json)
21
22 with open('data2.csv', 'r') as csv_file_2:
23     csv_reader = csv.DictReader(csv_file_2)
```



```
24     all_data = {} # null dictionary for making the json file
25     all_data['records'] = [row for row in csv_reader]
26
27 with open("data2.json", 'w') as json_file_2: # open the file, 'w' for writing and
28     creating one if doesn't exist
29     json.dump(all_data, json_file_2) # put all the data inside the json file, using
30     json.dump(all_the_data, the_file)
31
32
33
34 # 3) Load the data1 and data2 to df1, df2 dataframes respectively, using numpy as
35     panda libraries
36
37 import numpy as np
38 import pandas as pd
39
40 df1 = pd.read_csv('data1.csv') # load file1 into pandas dataframe
41 df2 = pd.read_csv('data2.csv') # load file2 into pandas dataframe
42
43
44
45 # 4) Print the number of the records in df1, and df2.
46
47 print("Number of records in df1:", df1.shape[0])
48 print("Number of records in df2:", df2.shape[0])
49
50
51
52 # 5) Print the average salary attribute in the df2
53
54 salary_mean = df2['salary'].mean() # calculate the mean of the salary attribute
55 print("The average of the salary:", salary_mean)
56
57
58
59 # 6) Normalize the age attribute in df2 using z-score normalization, and replace
60     the original one with the normalized one.
61
62 def z_score(attribute):
63     """ Calculate the z-score while z-score = (value - mean) / std
64         and return the values """
65     mean = np.mean(attribute)
66     std = np.std(attribute)
67     return [(value - mean) / std for value in attribute] #return a list with the
68     normalized values
69
70 df2['age'] = z_score(df2['age']) # replace the original with the normalized
```

- التاسك 1: احوال الملف من json ل csv:

- سطر 3 علمنا import لل packages اللي بنحتاجهم بعمليو التحويل، واللي همة json, csv
- سطر 5 فتحنا الملف ال json عن طريق with open وكتبنا اسم الملف اللي هو data1.json وبعدين حطيت نوع القراءة r لانه انا بس بدي اقرأ الملف وما بدي اعدل عليه، بعدين اخر اشي حطيت as json_file_1 يعني اعطيت الملف اسم عشان استخدمه انا بالكود (زي لما اعمل import لل pandas (as pd
- سطر 6 حطينا كل بيانات ملف ال json جوا متغير اسمه data عن طريق ال json.load(json_file_1)
- سطر 8 اخدنا كل البيانات عن طريق ال data['records'] وحطيتهم ب متغير هلاً اذا احنا بنتطلع عال json file اللي عنا بتلاحظوا انه ال structure تاعه كالاتي:
- اول اشي واكبر اشي اللي هو dictionary كبير يعني الملف كله عبارة عن dictionary واحد وهاد ال dictionary اله key واحد اللي هو records وال value لهاد ال key اللي همة ال records يعني اللي همة البيانات
- زي ما حكينا ال value لل key اللي موجود بال dictionary الكبير هو عبارة عن ليست لهيك هاي الليست هي ثاني اكبر اشي ب ملف ال json، طيب هاي الليست شو فيها؟
- بيجي دور ثالث الاشلي اللي هو dictionary صغير، هلاً هاي الليست فيها items كل item هو عبارة عن dictionary وهاد ال dictionary الصغير يمثل ال record
- نرجع للموضوع، هلاً لما انا اخذ ال value لل key اللي اسمه records انا هيكون اخذت ليست فيها dictionary كل dictionary هو عبارة عن record والصورة اللي تحت رح توضحك هاد الاشلي



```
data1.json - Notepad
File Edit View
{"records":[{"id": 2,"name": "John","age": 25,"loc":"NY","Gender":"Male"}, {"id": 5,"name": "Jane","age":56,"loc": "NY","Gender": "Female"}, {"id": 6,"name": "David","age":50,"loc": "MN","Gender": "Male"}, {"id": 7,"name": "FARI","age":45,"loc": "NY","Gender": "Female"}, {"id": 15,"name": "SARA","age":60,"loc": "NY","Gender": "Female"}, {"id": 20,"name": "KHAELD","age":30,"loc": "NY","Gender": "MALE"}, {"id": 50,"name": "MOURAD","age":40,"loc": "NY","Gender": "MALE"}, {"id": 18,"name": "SENA","age":56,"loc": "NY","Gender": "Female"}, {"id": 19,"name": "MARYAM","age":89,"loc": "NY","Gender": "Female"}, {"id": 17,"name": "DANI","age":40,"loc": "NY","Gender": "MALE"}, {"id": 21,"name": "IEAD","age":44,"loc": "NY","Gender": "MALE"}, {"id": 36,"name": "BASEM","age":40,"loc": "CH","Gender": "Female"}, {"id": 45,"name": "DERAR","age":40,"loc": "NY","Gender": "MALE"}, {"id": 7,"name": "JAMEEL","age":70,"loc": "NY","Gender": "MALE"}, {"id": 56,"name": "HANI","age":50,"loc": "CH","Gender": "MALE"}]}
```

- سطر 10 قرأنا ملف ال csv عن طريق ('data1.csv','w') with open و w لانه احنا بدنا نعمل write عالملف واذا ما كان في ملف بهاد الاسم يعمل create لواحد دايركت as csv_file_1 زي ما حكينا قبل هي تمسية للملف عشان اسمتخدمه بالكود
- سطر 11 هو عبارة عن object وظيفته انه يكتبلي علف ال csv عن طريق اني اطول من ال package csv فنكشن writer واحط جواه اسم الملف اللي بدي اكتب عليه (لاحظ كتبت اسم الملف عن طريق الاسم اللي سميت اياه فوق (اللي هو csv_file_1) ، ف اذا بدي اكتب عالملف لازم اكتب عليه عن طريق ال object اللي اسمه writer
- سطر 12 اخدنا اول عنصر من الليست الكبيرة، واول عنصر هو عبارة عن dictionary ف زي كاني انا اخدت record واحد من هاي البيانات وحطيتها بتمغير اسمه record، وهلا بتعرفوا ليش انا عملت هيك
- سطر 13 بدي اطبع او اكتب اول سطر ع ملف ال csv ودايما اول سطر يكون ال headers اللي همة اسماء ال columns لهيك انا عندي طريقتين اني اكتب ال header الاولى اني انا اكتب اسماء ال columns ب ايدي جوا ليست يعني زي هيك: writer.writerow(['id','name','age',....]) او الطريقة الثانية واللي هي الاسهل، مش انا اخدت record واحد فوق وسميت المتغير اصلا record؟ هلا هاد ال record ال keys تاعته هي نفسها اسماء ال columns اذا بترجع تطلع عالصورة فوق بتشوف انه ال keys تاعه اي record هي عبارة عن اسماء ال headers لهيك بقدر بسهولة اني احط ال record.keys() وطبعا اذا بدي اكتب ع ملف ال csv بستخدم ال object اللي انا عملته قبل شوي وبطول منه الفنكشن writerow وبحط جوا الاقواس ليست بالاسماء او الاشياء اللي بدي اعبي فيها السطر
- سطر 15 عملت for loop بتلف عكل عنصر بالليست (وكل عنصر هو عبارة عن dictionary وكل dictionary هو عبارة عن record) لهيك عملت for record in records وبعدين جواها بدي اياه تطبعلي كل سطر، يعني بدي اياه تعبيلي الاسطر بال values تاعة كل dictionary انت بتلف عليه، لانه انا ما بدي اضيف ال dictionary زي ما هو او ما بدي اضيف ال keys تاعة ال dictionary انا بس بدي اضيف ال values تاعة ال dictionary، واذا بتطلع الصورة فوق بتشوف انه احنا بهد ما كتبنا ال header اللي همة اسماء ال columns ف بعدين كل الاسطر اللي بنحتاجها تحتهم هي عبارة عن values

- التاسك 2: تحويل ملف ال csv ل json:

- سطر 22 قرأت الملف و r عشان بدش اعدل عليه واعطيته اسم csv_file_2
- سميت متغير اسمه csv_reader هاد المتغير فيه كل record على شكل dictionary عن طريق ال csv.DictReader(csv_file_2) من اسمها DictReader يعني بتقرأ ال records على شكل dictionaries

- سطر 24 عملنا dictionary فاضي بلانها احنا حكينا فوق انه ملف ال json هو عبارة عن dictionary كبير
- بسطر 25 لما عمل `all_data['records']` ف انا زي هيك كأني حطيت key جوا هاد ال dictionary والقيمة او ال value تاعة هاد ال key اللي هي لست فيها dictionaries لما عملنا هيك `[row for row in csv_reader]` هاي اسمها ال list comprehension انها انا عملت ليست كبيرة حطيت فيها كل مرة row هاي ال row هي عبارة عن dictionary اللي هي ال record بحد ذاته
- سطر 27 فتحت او عملت ملف `data2.json` من نوع w يعني write عشان بدي اطبع عليه واعطيته اسم برضه `json_file_2`
- هلا ب سطر 29 ف انا هون بعمل انه بنقل كل هاي البيانات (اذا بتلاحظوا انا كل بيانات ملف ال csv حطيتها جوا متغير اسمه `all_data` لهيك عشان انا بس احط هاي ال data ب ملف json بستخدم `json.dump(all_data, json_file_2)` هاي فنكشن بتساعدني حط ال data اللي عندي على شكل ال json او بنفسه ال format تاعة ال json files

- تاسك 3: بدي اقرأ الملفات على شكل `dataframes`:

- سطر 37 و 38 عملنا `import pandas و numpy`
- سطر 40 و 41 قرأنا الملفات اللي عملناها على شكل csv عن طريق `pd.read_csv` و هاد فنكشن بال pandas بقرألي الملفات اللي من نوع csv وبخط جوا الاقواس اسم ال file اللي بدي اقرأه (بشرط انه يكون موجود بنفس مكان ملف البايثون اللي بشتغل عليه)

- تاسك 4: بده اطبع عدد ال records بكل `dataframe`:

- بستخدم ال `df.shape` والناتج او ال output لهاد الفنكشن هو عبارة عن tuple محطوط فيه (rows, columns) بس انا بس بدي اطول ال rows اللي همة نفسهم ال records لهيك بدي اطول اول عنصر من هاد الناتج، وبطول اول عنصر عن طريق اني اختار ال index 0 اللي هي بتمثللي عدد ال records

- تاسك 5: اطبع ال avg ل عمود ال salary اللي بال `df2`:

- بقدر اطبع ال average او ال mean عن طريق اني اكتب اسم ال dataframe واطول ال column اللي بدي اطول ال mean تاعه بعدين بحت `mean()`. وبس

- تاسك 6: اعمل normalization لل column اللي اسمه age اللي بال `df2` باستخدام ال z-score وارجع اعيد كتابة العمود بالقيم الجديدة اللي معموللها normalization وجاهزة

- في قانون ال z-score الي هو لكل نقطة بطرح منها ال mean للعمود بعدين بقسمها على ال std تاع العمود كامل والقيمة الناتجة هي القيمة بعد ما نعمللها normalization

- سطر 62 عملت فنكشن اسمه z-score رح يدخل فيه attribute اللي بدي اعملها normalization
- سطر 63 و 64 عبارة عن multi line comment
- سطر 65 بطلع ال mean للعمود عن طريق np.mean(attribute) وبخزنها بمتغير
- سطر 66 بعمل نفس الاشئ لل std
- سطر 67 بعمل ليست هاي الليست القيم رح تكون فيها عبارة عن كل قيمة بال attribute الاصلية وبطرح منها ال mean اللي طلعت فوق وبعدين بقسمها عال std، كل القيم رح تدخل بهاد القانون، بالتالي الناتج او ال output او ال return من الفنكشن عبارة عن ليست معمول للقيم اللي داخلة فيه normalization عن طريق ال z-score
- سطر 70 بدي اعمل replace للعمود الاصلي عن طريق اني اعمل
`df2['age'] = z_score(df2['age'])` ف هون انا بعمل overwrite لعمود ال age بالناتج الطالع من
 الفنكشن z_score اللي عملته
- بتقدر تشوف الكود برضه [من هون مع شوية كومنترات](#)

Downloading ubuntu on windows:

1. Download Virtual Box from [here](#)
2. Download the Ubuntu image from [here](#)
3. Connect the Ubuntu image to the virtual box, watch [this video](#) to learn more

Downloading Airflow on Ubuntu:

- Follow the instructions [here](#)

Importing CSV:

```

1 import pandas as pd
2 pd.options.display.float_format = '{:.2f}'.format
3 pd.set_option('display.width',75)
4 pd.set_option('display.max_columns',20)
5
6 landtemps = pd.read_csv("./data/landtempssample.csv",
7                           names= ['stationid','year',
8                                   'month','avgtemp',
9                                   'latitude','longitude',
10                                  'elevation','station',
11                                  'countryid','country'],
12                           skiprows=1,
13                           parse_dates = [['month','year']],
14                           low_memory=False)
15
16 type(landtemps) # pandas dataframe
17
18 # show enough data to get a sense of how the import went
19 landtemps.head(7) # first 7 rows
20 landtemps.dtypes # show the datatypes for every column
21 landtemps.shape # show the shape for the dataframe
22
23 # fix the columns name for the data
24 landtemps.rename(columns={"month_year":"measuredate"}, inplace=True) # rename the
25 column name
26 landtemps.dtypes # show datatypes for every column
27 landtemps.avgtemp.describe() # show basic information about the column
28 (count,mean, std,min,max...)
29 landtemps.isnull().sum() # show how many null values in every column
30
31 # remove rows with missing values
32 landtemps.dropna(subset=['avgtemp'],inplace=True) #dropping the the record which
33 their avgtemp value is null
34 landtemps.shape

```

- أول 4 اسطر فوق هي عبارة عن تعديل شوية تعديلا بطريقة عرض البيانات ، وهي ابدأ مش ضرورية، يعني بقدر انت تشوف وتستخدم البيانات من دون ما تلعب بهاي الاعدادات، بس خنشرهم يعني:

○ سطر 2 هي عبارة عن اني انا اعرض الارقام ال float بس خانتين من بعد الفاصلة، عن طريق اني انا اعمل {:.2f} وال 2 بعد. بتعني انه بدني اياك تعرض اول خانتين من ال float number وال f هي عبارة عن انه هاد الرقم حيكون float

○ سطر 3 هو اني انا اغير ال width لل column، وهي شغلة مش ضرورية صراحة، لانه ال IDE اللي احنا بنستخدمهم همة بعملو autodetect لل width وبحطوه، و by default يكون 80

- سطر 4 احكيه كم من column تعرضي لما انا بدي اشوف ال head مثلا، مثلا انا عندي dataset فيها 100 عمود، حيكون صعب انه ال IDE يعرضلي اياهم كلهم، لهيك مثلا انا بحددله انا ال maximum columns يكون 20، ف هو بظهرلي 20 عمود وبخبي الباقي
- سطر 6 عملنا متغير سميناه landtemps ، وهاد المتغيره حيكون عبارة عن dataframe وحققنا ال داتا تاغتنا عن طريق ال read_csv لانه الملف بصيغة csv، بعدين بحط ال path ل مكان وجود البيانات (انا هون عملت ./data/landtempssample.csv. لانه البيانات موجودة جوا فايل مسميه data وهاد الفايل موجود بنفس مكان ملف البايتون)
- ال names بحط فيها اسماء ال columns اللي انا بدي اياهم بال داتا اللي عندي، ولان شرط اساسي انه يكون عدد الاسماء نفس عدد الاعمدة وب نفس الترتيب (ملاحظة: بحط اسماء الاعمدة الجديدة جوا ليست)
- ال skiprows من اسمها بتعمل skip للاسطر من فوق، وانا حطيت 1 لاني ما بدي اياه يقرألي اول سطر اللي هو ال headers لاني انا اوريدي حطيت اسماء لل columns مني عن طريق ال names method
- ال parse_dates هي بتدمجلي اكثر من عمود الهم علاقة بالوقت (ان وجد، يعني اذا كان في همود واحد رضه مش مشكلة) ويتحوللي ال data type لهاد العمود ل time، يعني بصير افقدر اتعامل معاه بطريقة more flexible ملاحظات مهمة:
- اذا كان عندي اكثر من عمود الهم دخل بالوقت ف بحطهم ب nested list يعني زي ما عملت انا فوق `[[‘month’,‘year’]]`
- اذا كان عندي اكثر من عمود وحطيتهم ب هاي ال statement ف ال output او النتاج حيكون عبارة عن column واحد، يعني بدمجلي العمودين مع بعض
- هون انا عندي بس ال month وال year بس ما عندي ال day لهيك ال pandas رح تعمل ال day يكون 1 by default
- ال low_memory الفكرة تاغتها انه اذا انا عندي داتا كثير كبيرة ف pandas by default يكون حاططها True، انه اذا الادوات كبيرة ف بدي اياك تستخدم مساحة قليلة من ال memory عن طريق انه يقسم ال data ل patches
- بس اذا كانت قيمتها False ف بحمل كل البيانات ويمكن يسحب كمية كبيرة من ال memory من الجهاز تاغك واللي ممكن يسبب ب بطء بالجهاز
- سطر 16: بعطيني ال type لل landtemps واللي هو pandas dataframe
- سطر 19: بعطيني اول 7 records ، ولو ما حطيت رقم ف رح يعطيني اول 5 تلقائيا
- سطر 20: بعطيني ال data types لكل column بال dataframe
- سطر 21: بعطيني حجم ال dataframe اللي عندي على شكل (rows, columns) وال output من ال .shape. عبارة عن tuple

- سطر 24: بدي اعمل rename لل column الناتج من دمج ال year وال month، لما دمجهم حط اسمهم month_year ف انا بدي اغيره عن طريق ال rename. وكتب columns عشان احدد انه انا بدي اغير بالاعمدة، وبحط ال value لل columns اللي هو dict ال key عبارة عن الاسم القديم وال value عبارة عن الاسم الجديد، بعدين اخر اشي عملنا inplace=True لانه لما ما عملت True رح يتغير اسم ال column بس انه بشكل مؤقت، يعني لو رجعت بدي اشوف ال head ف رح تلاقى انه ما تغير اسم العمود لاني انا ما عملت inplace=True ، ف هاي وظيفتها انه انا اذا بدي اغير اشي عال dataframe بشكل دائم لازم اعمل inplace=True ولو ما حطيت هاي ال inplace ف هي بتكون محطوبة ثمنتها by default انها False
- سطر 27: هون انا بعمل شوية descriptive analysis للعمود ال avgtemp عن طريق ال describe(). هاي بتعطيني معلومات اساسية عن هاد ال column زي ال min, max, avg, 25%,50%,75%,count,mean,std ولو ما حطيت انه انا بدي بس لهاد ال column يعني landtemps.describe() ف رح يطلعلي هاي المعلومات لكل ال columns. (ملاحظة: لما ادي اطول همود معين بناديه ب اسمه اللي انا سميتة فوق جوا ال names مش ب اسم ال column جوا ملف ال csv نفسه)
- سطر 29: هاد بعمل sum للعدد ال null values لكل column بال dataframe وال output سيكون زي هيك: فهون بحكي لي انه عمود ال avgtemp فيو 14446 قيمة فاضية وال country فيو 5 والباقي صفار ، يعني كلهم ملايين او ما فيهم null_values
- سطر 32: dropna بتعملي drop يعني بتحذف ال records اللي فيهم null values، بس هون احنا حطينا subset القيمة فيها ['avgtemp'] يعني بدي اياك تشطب ال records اللي فيهم null values بالعمود avgtemp، يعني اذا عمود ال country فيه null value ب record معين بس ما في null value بهمود ال avgtemp ف مش رح يحذف هاد ال record لانه انا محدله بتحذف ال record بحال بس اذا ال avgtemp فيه null value، بعدين عملت inplace عشان اثبت هاد الاشي واخليه دائم مش مؤقت
- سطر 34: برجع بطبع ال shape لهادي ال dataframe وكيد رح ينقص عدد ال records عن قبل لاني عملت drop لل null values الموجودات ب عمود ال avgtemp، بس عدد ال columns ثابت
- بتقدر تنزل هاي ال dataset عشان تشتغل عليها براحتك [من هون](#)

```
Out[46]:
measuredate      0
stationid         0
avgtemp          14446
latitude          0
longitude         0
elevation         0
station          0
countryid         0
country           5
dtype: int64
```

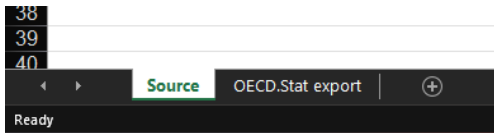

Importing Excel:

```
1 import pandas as pd
2 pd.options.display.float_format = '{:.0f}'.format
3 pd.set_option('display.width',85)
4 pd.set_option('display.max_columns',20)
5
6 # import the land temperature data
7 percapitaGDP = pd.read_excel("./data/GDPpercapita.xlsx",
8                               sheet_name = "OECD.Stat export",
9                               skiprows = 4,
10                               skipfooter = 1,
11                               usecols = "A, C:T")
12
13 percapitaGDP.head()
14 percapitaGDP.info()
15
16 # rename the Year column to metro
17 percapitaGDP.rename(columns={'Year':"metro"}, inplace=True)
18 percapitaGDP.metro.str.startswith(' ').any() # returns boolean value
19 percapitaGDP.metro.str.endswith(' ').any() # return boolean value
20 percapitaGDP.metro = percapitaGDP.metro.str.strip()
21
22 # convert the data columns to numeric
23 for col in percapitaGDP.columns[1:]:
24     percapitaGDP[col] = pd.to_numeric(percapitaGDP[col], errors='coerce')
25     percapitaGDP.rename(columns={col:"pcGDP" + col}, inplace= True)
26
27
28 percapitaGDP.head()
29 percapitaGDP.dtypes
30 percapitaGDP.describe()
31
32 # remove rows where all of the per capita GDP values are missing
33 percapitaGDP.dropna(subset = percapitaGDP.columns[1:], how = 'all', inplace=True)
34 percapitaGDP.describe()
35 percapitaGDP.head()
36 percapitaGDP.shape
37
38 # set an index using the metropolitan area column
39 percapitaGDP.metro.count()
40 percapitaGDP.metro.nunique()
41 percapitaGDP.set_index('metro',inplace=True)
42 percapitaGDP.head()
43 percapitaGDP.loc['AUS02: Greater Melbourne']
```

- نبيلش، ب اول 4 اسطر بنعرف احنا شو بعملوا، بس الفرق انه بالسطر الثاني خط انه ما يطلعلي ولا خانة اذا كان الرقم float عن طريق ال {:.0f}

- ب سطر 7 قرأنا ال dataset عن طريق pd.read_excel لانه الملف بصيغة excel اللي هي .xlsx

○ حطيت ال path بالمكان اللي فيه الداتا



○ سطر 8، احنا بنعرف انه بال excel file ممكن يكون في اكثر من sheet لهيك انا بختار ال sheet اللي بدي اطول منها البيانات، ويكتب اسمها بالزبط زي مهو بالملف، وبجالتنا اسم ال sheet هو OECD.Stat export

	A	B	C	D	E
2	Dataset: Metropolitan areas				
3		Variables	GDP per cap		
4		Unit	US Dollar		
5		Year	2001	2002	2003
6	Metropolitan areas				
7	AUS: Australia				
8	AUS01: Greater Sydney		43313	44008	45424

○ سطر 9 بحكيه انا بدي اعمل سكيب ل اول 4 اسطر، ليش؟ لانه اول 4 اسطر عبرة عن اسطر ما الها دخل الداتا، او انه الها دخل بس هي مش عبارة عن بيانات، وانا بدي استخدم البيانات واخدها، بديش اشي ثاني

■ بتلاحظوا انه ما في رقم 1 بال rows؟ ليش؟
لانه ممكن ب بعض المرات يكون في اسطر مخفية، وهي هون مبين انه السطر 1 مخفي، طيب كيف نشوفه؟ اكبس right click على رقم 2 اللي هو السطر الثاني، وبعدين في يكون خيار unhide اكبس عليه وببينك السطر المخفي

705	USA167: Weber		34592	34997	35587	357
706	USA169: Cass		44597	46856	49043	491
707	USA170: Renton (AR)		41088	44687	45296	477
708	Data extracted on 05 May 2020 10:55 UTC (GMT) from OECD.Stat					
709						
710						
711						
712						

○ سطر 10 من اسمها skip footer ، مرات برضه يكون في اسطر اخر الملف ما محتاجها لهيك بعملها سكيب، والسكيب يكون بطريقة عكسية، انه اذا حطيت سكيب 2 يعني رح يعمل سكيب عن اخر سطرين وهكذا، زي هون بالصورة بتلاحظوا انه هاد الاسطر اللي بالاخير انا ما بستفيد منه

#	A	B	C	D	E	F	G	H	I	J	K
1	#NAME?	Sorry, the query is too large to fit into the Excel cell. You will not be able to update your table with the Stat Populator.									
2	Dataset: Metropolitan areas	POP per 1000	Year	US Dollar							
3	Year	2001	2002	2003	2004	2005	2006	2007	2008	2009	
4	Metropolitan areas										
5	AUS: Australia										
6	AUS01: Greater Sydney	43313	44008	45424	45837	45423	45547	45880	45225	45900	
7	AUS02: Greater Melbourne	40122	40894	40664	40188	41424	41599	42514	40915	41464	
8	AUS03: Greater Brisbane	37580	37564	39050	40782	42976	44475	44635	46192	43007	
9	AUS04: Greater Perth	42713	47311	48718	51020	50278	50142	52651	53899	53616	
10	AUS05: Greater Adelaide	36505	37194	37634	37999	37804	38151	39049	38502	39538	
11	AUS06: Gold Coast										
12	AUS07: Canberra	41465	44028	44814	45675	46024	46578	46699	48919	51358	
13	AUS08: Newcastle										
14	AUS10: Wollongong										
15	AUS11: Sunshine Coast										
16	AUS14: Geelong										
17	AUT: Austria										
18	AT001: Vienna	52504	53172	52675	53496	53686	55215	56330	56900	54930	
19	AT002: Graz	40299	40154	40119	41713	40535	40256	50898	51934	48945	
20	AT003: Linz	47110	47031	47942	48029	49670	51364	52841	54883	52092	
21	AT004: Salzburg	44493	44494	44498	46968	47819	48254	48198	49998	49687	
22	AT005: Innsbruck	49065	49467	49820	50425	52055	53372	54573	54470	52388	
23	AT006: Klagenfurt	38819	39580	40049	41224	42746	43514	40746	45932	43198	
24	BEL: Belgium										
25	BE001: Brussels	98117	86383	86383	88411	89696	76413	71928	71526	70835	
26	BE002: Antwerp	50822	51662	51561	53763	56093	56609	58313	58391	54150	
27	BE003: Gent	44247	46470	46881	46814	46460	46460	45884	45064	46961	
28	BE004: Charleroi	29320	29106	29627	30548	31005	31519	32357	33037	31262	
29	BE005: Liege	31944	32090	32678	33749	34440	35263	36078	36009	35542	
30	CAN: Canada										
31	CAN01: Toronto									43804	
32	CAN02: Montreal									34085	

- سطر 11 usecols اختصار ل use columns
بملف الاكسيل في اعمدة بتكون فوق فوق مسمية
ب A, B, C, الخ، مرات في يكون اعمدة انا ما
بحتاجها، او اعمدة فاضية، ف انا ما بدي
استخدمها، ليهيك انا من خلال usecols بقدر اسمي
الاعمدة اللي بدي اياها، زي هون مثلا بالداتا سيت
اللي عنا، العمود B هو عبارة عن عمود فاضي،
ف انا ما بدي اياه وما بدي احطه بال
dataframe اللي بدي اشتغل عليها ليهيك كتبت
usecols = "A, C:T" انه اختار لي العمود A،
بعدين ادل ب رينج ودخلني الاعمدة من C ل T

- سطر 13 بعطيني اول 5 records بال داتاسيت اللي عندي

- سطر 14 بتعطيني info(). معلومات عن كل عمود زي عدد ال none null values وال datatype
لهاد العمود

- اذا بتلاحظوا عمود ال year تحته اسماء مدن ف ليهيك مش منطقي انه يكون اسم العمود year ليهيك احنا
بدنا نغيره عن طريق ال rename. وبحط اسم العمود القديم بعدين الجديد as a dictionary وبعدين
بعمل inplace=True عشان اثبت هاد التغيير واخليه تغيير دائم مش مؤقت

- هلا بعد مغيرنا اسم العمود من year ل metro بنلاحظ انه في شوائب كتير بالاسم ك string زي انه
في فراغات قبل الاسم او بعده، هاد كله لازم نتخلص منه وعملية التخلص منه اسمها data cleaning،
طيب، هلا كيف بدنا نشيل الفراغات القبل واللي بعد، اول اشي بدنا نتأكد اصلا اذا في فراغات قبل وبعد،
كيف؟ عن طريق اني انادي ال dataframe بعدين انادي اسم العمود اللي هو metro بعدين بنادي str.
عشان اخذ خصائص ال string واقدر اوصل للي بدي اياه، بعدين في method بال str اللي هي
(startswith() وبتحط جوا الاقواس ايش الاشياء اللي انت بتدور عليه اول الاسم (بحالتنا رح يكون فراغ
او space " ") بعدين بحط any(). عشان انا بدي اعرف انه اذا في بس اسم واحد في ب قبله فراغ، ف
لو كان بس اسم واحد ب قبله فراغ حيعطيني True ولو كانوا كلهم برضه حيعطيني True ولو ما كان
ولا وحدة فيهم بتطبق عليها الشرط وقتها حيعطيني False، ولازم نعرف انه ال methods اللي همة
(startswith() و endswith() هتدول برجعولي Boolean value يعني بعطوني True او False
ف بسطر 18 و 19 طبقت هاد الاشياء، ورح يرجعولي التنتين True يعني في spaces ببعض ال
records قبلها وبعدها، ليهيك انا بعد ما عرفت هاي المعلومة، وقتها باجي اتخلص منهم

- بدي اعمل overwrite للقيمة الاصلية بالقيمة الجديدة اللي انا زبطتها، عن طريق اني انا اعطي value
لهاد العمود وهاي ال value هي نفي العمود بس بعد ما عملته clean
وكيف بعمله clean؟ بنادي العمود من الداتاسيت وبعدين بنادي ال str. عشان اقدر اوصل لل فنكشن
اللي رح يعمللي cleaning واللي هو strip() هاد الفنكشن بنادي به بعد ال str. وبشيللي الفراغات اللي قبل
الكلمة وبعدها، هو by default رح يكون ال قيمة اللي بده يشيلها قبل وبعد ال space بس مثلا لو انت
بدك اشي ثاني مثلا * بتقدر تحطها جوا الاقواس زي هيك strip("*"). وبشيل كل ال * اللي قبل الاسم
وبعده

- هلاً لما عملنا `percapitaGDP.info()` بنلاحظ انه ال `dtype` لل `columns` عباره عن `object` لهيك انا لازم اغيره لانه لازم يكون عبارة عن ارقام او `integers` او `float` وفي عندي كمان مشكلة انه اسماء الاعمدة عندي بال `dataframe` عبارة عن `integers` وهاد اشئ ممنوع بالبايثون، لازم يكون اسماء الاعمدة `string` لانه لو مثلاً بدي انادي العمود 2012 زي هيك `percapitaGDP[2012]` ف هو بهاي الحالة حيفكر انها `index` مش اسم عمود، لهيك لازم نغيره، واحنا رح نغيره بطريقة انه نلف عالاسم واحد واحد ونضيف قبله كلمة `pcGDP`
- طيب، بسطر 23 لفينا عالعمدة كلهم ماعدا اول واحد، لانها اول واحد جاهز وما بدي اغير فيه اشئ وال `values` تاعته مزبوطات والاسم تاعه برضه مزبوط، لهيك بلف عليهم كلهم ماعدا اول واحد عن طريق `for col in percapitaGDP.columns[1:]` انه لفل عكل الاعمدة دون اول واحد
- بعدين بلف ع كل عمود بال `dataframe` وبحول محتوياته ل `numeric` عن طريق `pd.to_numeric` وبحط جوا اسم العمود من ال `dataframe` وبعدين نوع ال `error` حيكون `coerce`، هاي شو معناها؟ معناها هاي معناها انه اذا فيه `value` انت ما عرفت تحولها ل `numeric` ف حولها ل `NaN` يعني `NaN` `value`، لانه اذا بتلاحظوا في بالبيانات ال `value` بتكون عبارة عن ف هاي قيمة ما بزبط انه يحولها ل `numeric` لهيك بحولها ل `NaN`
- بعدين برضه جوا ال `for loop` بغير اسم العمود عن طريق اني اضيف كلمة `pcGDP` قبل كل اسم `column` وبعدين بعمل `inplace=True`
- سطر 28 ل 30 بنعرف شو بعملوا
- ب سطر 33 بدي احذف ال `records` اللي فيهم `null values` بحيث انه كل ال `columns` تكون القيمة تاعته `null`، يعني.....

		Unit US Dollar																	
	Year	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
5	Metropolitan areas																		
6	AUS: Australia																		
7	AUS01: Greater Sydney	43313	44008	45424	45837	45423	45547	45880	45225	45900	45672	46535	47350	47225	48510	50075	50519	50578	49860
8	AUS02: Greater Melbourne	40125	40894	41602	42188	41484	41589	42316	40975	41384	40943	41165	41264	41157	42114	42928	42671	43025	42671
9	AUS03: Greater Brisbane	37580	37564	39080	40762	42976	44475	44635	46192	43507	42774	44166	43764	43379	43754	44388	45723	46876	46640
10	AUS04: Greater Perth	45713	47371	48719	51020	52278	50142	52521	53899	53616	55111	57118	57670	56153	57555	58544	60302	60424	70398
11	AUS05: Greater Adelaide	36505	37194	37634	37399	37604	38151	39049	38502	39538	39309	39223	39812	39855	40306	40295	39737	40115	39921
12	AUS06: Gold Coast																		
13	AUS07: Canberra	41465	44028	44814	45675	46024	48578	49689	48919	51358	51364	52801	54049	53206	54503	56421	55979	56301	55971
14	AUS08: Newcastle																		
15	AUS10: Wollongong																		
16	AUS11: Sunshine Coast																		
17	AUS14: Geelong																		
18	AUT: Austria																		
19	AT001: Vienna	52504	53172	52675	53486	53686	55218	56330	56000	54930	55335	55794	55150	54468	53992	53885	54575	54426	54890
20	AT002: Graz	45259	45124	46119	47713	48535	49295	50895	51204	48945	49218	50628	51060	50349	50492	50190	51332	52441	53155
21	AT003: Linz	47110	47031	47942	48529	49870	51364	52841	54883	52092	52903	54111	53941	53994	53792	54330	54387	55553	56340
22	AT004: Salzburg	53464	54095	54493	55855	57859	60224	63079	65998	69627	67114	67714	64752	65953	67784	68893	67748	69662	69998
23	AT005: Innsbruck	49065	49467	49820	50425	52055	53372	54573	54470	52388	52617	53621	54087	54019	53602	54067	54289	54996	55176
24	AT006: Klagenfurt	38875	39592	40049	41224	42746	43514	45746	45932	43198	44284	45911	45361	44742	45004	44729	44691	45797	47109
25	BEL: Belgium																		

اذا بتلاحظوا هون، الاعمدة من 2001 ل 2018 ب بعض ال `records` كلها بتكون فاضية، هون كانت عبارة عن "...." بس قبل شوي غيرناها ل `NaN` لهيك في بعض الاسطر حيكون كل الاعمدة فيها تاعة التواريخ `null values`، هون بدي احذفهم، انا كيف خليته يشطب بس ال `records` اللي فاضي فيهم كل الاعمدة؟ عن طريق `'all'` `how` (ملاحظة انه انا ما اخدت اول عمود لانه هاد عالاكيد كله رح يكون معبأ، لانه هاد بمثابة `index` وبعد شوي حنشوف كيف حنحطه ك `index`) واخر اشئ بعمل `inplace=True` عشان اثبت حذف الاسطر واحذفهم بشكل كامل

- من سطر 34 ل 36 بنعرف شو بعملوا

- سطر 39 ال count(). بتعطيني كم من value موجودة بهاد العمود
- سطر 40 ال nunique(). بتعطيني عدد ال unique values بهاد العمود
ولو كان عدد ال unique values يساوي عدد ال count بهاد العمود ف بزبط معي اني احطه ك index لانه unique وما بتكرر
- ب سطر 41 بدي احط عمود ال metro ك index عن طريق
percapita.set_index('metro',inplace=True) بحط جوا الاقواس اسم العمود اللي بدي اخليه index وما بنسى اعمل inplace=True عشان اثبت هاد الاشئ واخليه دائم
- ب سطر 42 بعرضلي اول 5 اسطر
- ب سطر 43 بطول ال record اللي ال index فيه اسمه "AUS02: Greater Melbourne" ب استخدام ال .loc. وهاي بحط جواها اسم ال index اللي بدي اطوله، والنتائج او ال output لهاي الجملة
حيكون عبارة عن record واحد كامل
- بتقدر تنزل الداتا عندك [من هون برضه](#)

اخخخ ادعولي ادعولي يجماعة