case study explanation

https://docs.google.com/document/d/1tK-y-dMpHeZ8IYXlwt1xV2WQZ1Fj0vo0NPa9NB0BbSE/edit?usp=sharing

Report Created by: Olaniyan Farouq Olayinka
Date: 23rd November 2024

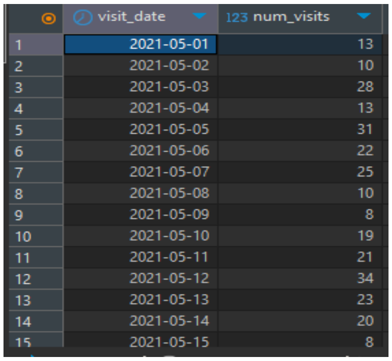
TASK 1

Task 1 requires me to write a **python script** to process the files provided to me, and also to create a single dataset to perform three (3) analysis and also to generate three (3) outputs.

- The Python script file
  Link: https://drive.google.com/file/d/1ChunWyluezCvnTNFLUq3lV_cU9rtazhe/view?usp=sharing
- The final dataset
  Link: https://drive.google.com/file/d/1je1tlyW8i5xSW2p-5lK0LJxTtlmS7oOr/view?usp=sharing
- The text file containing the queries on the final dataset used to obtain the outlined analyses in the case study pdf file
  Link: https://drive.google.com/file/d/1ZglaR6J0vIlav8S7wwfkojtXuoE19-2B/view?usp=sharing

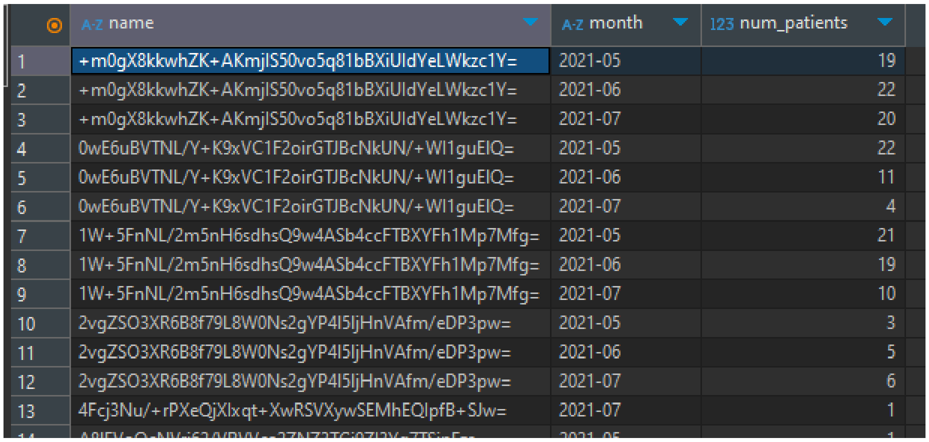
Additionally I am attaching the three (3) analyses from my Postgres Database(attaching three(3) screenshots)

- Number of hospital visits per day over the period



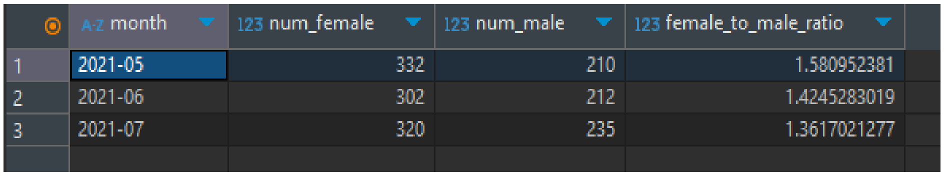
	visit_date	num_visits
1	2021-05-01	13
2	2021-05-02	10
3	2021-05-03	28
4	2021-05-04	13
5	2021-05-05	31
6	2021-05-06	22
7	2021-05-07	25
8	2021-05-08	10
9	2021-05-09	8
10	2021-05-10	19
11	2021-05-11	21
12	2021-05-12	34
13	2021-05-13	23
14	2021-05-14	20
15	2021-05-15	8

- Number of patients attended to per doctor per month showing their names



	name	month	num_patients
1	+m0gX8kkwhZK+AKmjIS50vo5q81bBXiUIdYeLWkzc1Y=	2021-05	19
2	+m0gX8kkwhZK+AKmjIS50vo5q81bBXiUIdYeLWkzc1Y=	2021-06	22
3	+m0gX8kkwhZK+AKmjIS50vo5q81bBXiUIdYeLWkzc1Y=	2021-07	20
4	0wE6uBVTNL/Y+K9xVC1F2oirGTJBcNkUN/+WI1guEIQ=	2021-05	22
5	0wE6uBVTNL/Y+K9xVC1F2oirGTJBcNkUN/+WI1guEIQ=	2021-06	11
6	0wE6uBVTNL/Y+K9xVC1F2oirGTJBcNkUN/+WI1guEIQ=	2021-07	4
7	1W+5FnNL/2m5nH6sdhsQ9w4ASb4ccFTBXYFh1Mp7Mfg=	2021-05	21
8	1W+5FnNL/2m5nH6sdhsQ9w4ASb4ccFTBXYFh1Mp7Mfg=	2021-06	19
9	1W+5FnNL/2m5nH6sdhsQ9w4ASb4ccFTBXYFh1Mp7Mfg=	2021-07	10
10	2vgZSO3XR6B8f79L8W0Ns2gYP4I5IjHnVAfm/eDP3pw=	2021-05	3
11	2vgZSO3XR6B8f79L8W0Ns2gYP4I5IjHnVAfm/eDP3pw=	2021-06	5
12	2vgZSO3XR6B8f79L8W0Ns2gYP4I5IjHnVAfm/eDP3pw=	2021-07	6
13	4Fcj3Nu/+rPXeQjXlxqt+XwRSVXywSEMhEQIpfB+SJw=	2021-07	1
14	[illegible]	2021-05	1

- The ratio of female to male patient visits per month

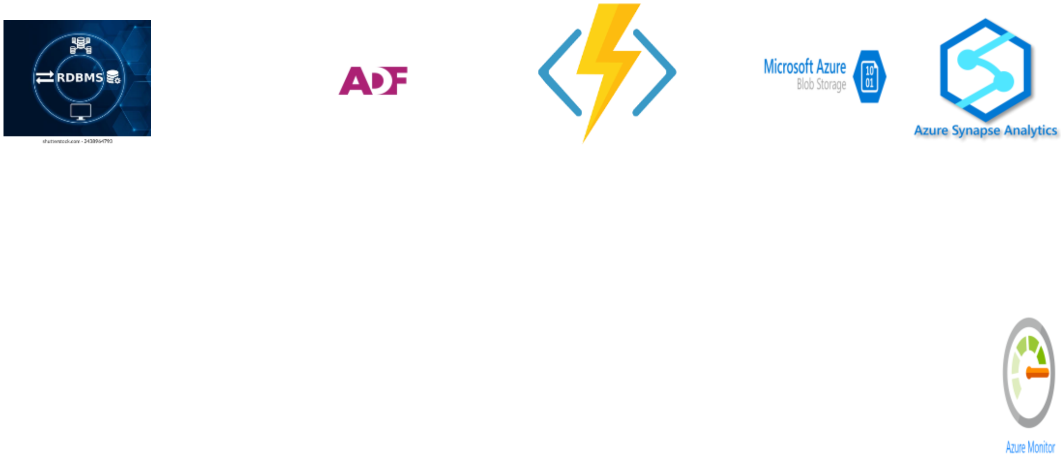


	month	num_female	num_male	female_to_male_ratio
1	2021-05	332	210	1.580952381
2	2021-06	302	212	1.4245283019
3	2021-07	320	235	1.3617021277

TASK 2

Task 2 requires me to outline a high-level **architecture** to deploy and schedule the python script to run once a day with my **design** reading the data from an **RDBMS** and writing the resulting dataset to a **Data Warehouse**. The **architecture** would be based on **Azure cloud platform**.

Diagram:



Text:

On-Prem RDBMS
|
[Secure Connection: VPN Gateway / ExpressRoute]
|
Azure Data Factory (Orchestrator)
|
Azure Function (Python Script)
|
Azure Storage (Blob) -> Azure Synapse Analytics (Data Warehouse)
|
Azure Monitor (Error Handling & Monitoring)

Architecture Overview

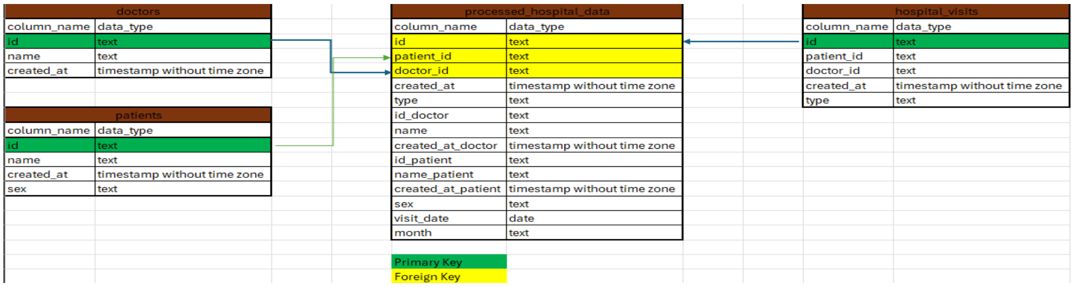
1. **On-Prem RDBMS**: I would connect the on-premises database to Azure using **Azure VPN Gateway** or **ExpressRoute** for secure data transfer.
2. **Azure Services**:
   - **Azure Data Factory (ADF)**: I would use **ADF** for **orchestrating** the **ETL** workflow, including triggering the Python script and scheduling its execution.
   - **Azure Storage (Blob)**: Then store intermediate data files (e.g., CSVs) for backup or staging data.
   - **Azure Functions**: Deploy the **Python script** as an **Azure Function** for scalable and serverless execution.
   - **Azure SQL Database** (Optional for staging): Although this is optional, I would temporarily stage data before writing to the final Data Warehouse.
   - **Azure Synapse Analytics** (Data Warehouse): Finally, I store the final processed dataset for analysis and reporting.
3. **Security and Monitoring**:
   - **Azure Key Vault**: This here securely stores database credentials and other sensitive information.
   - **Azure Monitor**: I would use this to track the pipeline execution and logs for debugging and monitoring.

Workflow

For clarity, I would break down the process into stages for better understanding

1. **Data Extraction**: As I stated earlier, I would use **Azure Data Factory** to connect to the on-prem RDBMS via Integration **Runtime** for secure and seamless access. Afterwards I can extract raw data into **Blob Storage** or pass it directly to the **Azure Function**.
2. **Data Processing**: **Azure Function** runs the **Python script**, processes the data, and writes the output to Blob Storage or Azure SQL Database.
3. **Data Loading**: **Azure Data Factory** would also be used to move the processed dataset from Blob Storage or Azure SQL Database to Azure Synapse Analytics using the “COPY INTO” command.
4. **Scheduling**: Azure Data Factory has a built-in scheduler that would be used to trigger the entire workflow once daily.
5. **Error Handling and Monitoring**: A crucial part, this helps see the progress and to know what stops the process flow. To achieve this, I would configure **Azure Monitor** for logging and alerting in case of failures during the ETL process.

WAREHOUSE DESIGN(Addition)



doctors	
column_name	data_type
id	text
name	text
created_at	timestamp without time zone

patients	
column_name	data_type
id	text
name	text
created_at	timestamp without time zone
sex	text

processed_hospital_data	
column_name	data_type
id	text
patient_id	text
doctor_id	text
created_at	timestamp without time zone
type	text
id_doctor	text
name	text
created_at_doctor	timestamp without time zone
id_patient	text
name_patient	text
created_at_patient	timestamp without time zone
sex	text
visit_date	date
month	text

hospital_visits	
column_name	data_type
id	text
patient_id	text
doctor_id	text
created_at	timestamp without time zone
type	text

🟩 Primary Key
🟨 Foreign Key