Dado o cenário proposto avaliei 2 possíveis soluções:

1. local (on-premise)
2. nuvem (Azure)

Em ambas as soluções estou considerando que a base transacional permaneça on-premise, ou seja, a empresa *Acquirer LTDA* ainda não tem interesse em migrar para a nuvem ou está em processo de migração (pensei nesse cenário por ser uma situação bastante comum). Para este exemplo criei uma base PostgreSQL no serviço de hospedagem gratuito *Railway* (https://railway.app/) para simular a base transacional da empresa

Solução 1

Código Python conectando diretamente na base transacional, as regras de negócio ficam encapsuladas numa view "*vw_profit*" que por sua vez é consumida pelo código Python

Softwares utilizados:

- Serviço *Railway* (base on-premise)
- Pgadmin (criação dos objetos e carga das tabelas)
- Visual Studio Code (código Python)
- Github

Solução 2

Em linhas gerais criar na nuvem uma base espelhada da base on-premise, a cópia dos dados seria realizada via pipeline (carga full para a tabela *contract* e carga incremental para a tabela *transaction*, e para gerenciar ambas as cargas criar uma base que contenha uma tabela *metadata* contendo quais tabelas/campos devem ser carregados e se essa carga é full ou incremental), os dados ficariam armazenados no Data Lake no formato *Parquet*

Para aplicar a regra de negócio solicitada usaríamos o DBT e nele seria criado o modelo (a mesma query da view "*vw_profit*" da 1ª solução) separando assim a camada "*raw*" da regra de negócio

Por fim linhas de código Python desenvolvidas no Jupyter Notebook similares as que fiz na 1ª solução mostrariam a informação do ganho da empresa para cada mês

Softwares utilizados:

- Serviço *Railway* (base on-premise)
- *Azure Synapse Analytics* (criação/schedule do pipeline)

- *Azure Database for PostgreSQL flexible server* (base contendo a tabela *metadata* utilizada pelo pipeline)
- *Data Lake Storage Gen2* (dados brutos no formato *Parquet*)
- *Dedicated SQL Pool* (banco)
- *DBT* (aplicação das regras de negócio sem transformar os dados *raw*)