Pré requisitos 1º parte:

1. Utilizar uma IDE compatível com python. Para esta atividade foi utilizado o VSC;
2. Ter python instalado;
3. Ter instalado as bibliotecas: pandas, numpy, unicodedata, re, sqlalchemy;
4. Ter SQLite instalado;
5. No arquivo testFirstPartETL.py mudar para o devido caminho da sua máquina:
    a. inputPath – Caminho onde estão os arquivos json da Base A;
    b. outputPath – Caminho onde será salvo a base para BI, ficando assim:
        i. sqlite:///<seu_caminho_para_salvar>/pd_database.sqlite3

Observações:

O código está comentado para facilitar o entendimento do que está acontecendo. Foi feito apenas em um arquivo para manter a ideia de ser simples, mas pensando em reaproveitamento de código as funções ali criadas podem ficar em um arquivo separado, contendo apenas funções para tratamento de dados, por exemplo.

Decidi não utilizar os arquivos student_follow_subject.json e subjects.json pois com as informações que eu possuía caso eu juntasse todas as bases eu iria gerar uma falsa informação de acesso relacionada a um conteúdo. Utilizando as outras cinco bases eu consegui gerar outras informações que podem ser relevantes para a análise de perfil e segmentação dos alunos.

Em um cenário real eu entraria em contato com o time parceiro (BI, CRM, Projetos, DS) para validar os campos que realmente são necessários para eles. Caso a entrega fosse para DS, eu aplicaria técnicas de feature engineering (pex. one hot enconding, feature hashing encode, missing data solutions, etc).

Resposta para a pergunta: "Como você estruturaria a solução para que a base analítica seja mantida atualizada? "

O código que eu enviei poderia ser utilizado em um SageMaker sendo iniciado por uma chamada via Lambda (AWS) ou AirFlow (Free), com as devidas alterações de output (salvar no S3, Redshift ou outros)

O ETL também poderia ser feito pelo Glue (AWS), fazendo as alterações no código necessárias para funcionar na ferramenta.