**Questions:**

1. Given the table structure provided in the assignment, how would you handle indexing to ensure efficient querying?
    - Use Composite Keys when querying e.g. **id** and **submission_id**
    - Create additional indexes on frequently queried table attributes

2. The assignment mentions transforming data to fit a specific PostgreSQL table structure. How would you handle a situation where the Excel data doesn't neatly fit into the provided schema?
    - Depending on missing data either:
        i. Compute e.g. Compute cme_completion_date from date_submitted for instance in Test Data scenario 3 where the location is *Kilifi*
        ii. Fill will NULL unless the key is a primary key

3. When deploying this Dockerized Flask application, how would you ensure that the PostgreSQL database maintains data persistence?
    - Define a volume in the docker configuration file and mount the volume when running the app

4. How would you implement and monitor a cron job for this assignment to ensure the data processing happens as expected?

5. If the cron job failed for some reason (e.g., the Excel file was not accessible), how would you handle error logging and notifications?
    - Logging events (errors) for troubleshooting
    - Error Handling

6. Given the potential for new datasets in the future, how would you design the system to easily accommodate new or altered data fields?
    - Use flexible data types e.g. TEXT
    - Store attribute records in a separate entity then add/modify details without interfering with the main tables

7. If stakeholders wanted real-time analytics on the number of completed CMEs and drill curriculum topics, how would you modify or extend your current solution?
    - Add a column to include totals of topics done based on data fetched from the *mentor_checklist/cme_grp/cme_total* and *mentor_checklist/drills_grp/drills_total* columns in the input file