1. Background & Scenario

Modern clinical-trial datasets come from many disparate sources—partner hospitals, regional registries, laboratories, patient devices, and more. Each source delivers data in its own shape (CSV exports, JSON APIs, Excel files, free-text reports). Analysts currently spend weeks cleaning these files by hand, delaying insights and introducing errors.

Your company—a health-tech startup focused on clinical-trial analytics—needs a **scalable, cost-efficient data pipeline** that can ingest, clean, integrate, and store this messy data in a single data warehouse for fast querying. You are the data engineer tasked with designing and implementing that pipeline.

2. Assignment Overview

Design and build a backend data pipeline that:

1. **Ingests** heterogeneous raw datasets.
2. **Cleans & normalizes** the data
3. **Loads** the transformed data into a query able data warehouse.
4. **Demonstrates**—with sample queries—how analysts can derive insights.
5. **Documents** every assumption, design choice, trade-off, and cost consideration.

3. Data Sources to Incorporate

Source	What it Covers
WHO ICTRP (International Clinical Trials Registry Platform)	Aggregates >20 national registries (EUCTR, ISRCTN, ChiCTR, CTRI, JPRN, ANZCTR, etc.). Each record is a "primary" trial plus secondary IDs.
EU Clinical Trials Register (EUCTR)	All interventional drug trials conducted in the EU/EEA since 2004. Includes protocol and result summaries.

ISRCTN Registry	Global “any intervention” registry run by BMC. Good coverage of U.K. academic studies.
EMA Clinical Data Publication	Redacted clinical study reports submitted for EU marketing approval.

4. Detailed Guidelines

1. **Data Sources**
    a. Expect multiple source types. Some of these will be straightforward and some might not
    b. Provide a short description or sample snippet in your documentation to illustrate the messiness.
2. **Ingestion & Transformation**
    a. Parse each source format.
    b. Make the pipeline idempotent (re-running should not duplicate data).
    c. Design for easy extension to new sources.
3. **Warehouse Design**
    a. Pick any warehouse technology (cloud data warehouse or relational DB).
    b. Explain how your schema supports analytic queries while remaining scalable.
4. **Insight Demonstration**
    a. Provide sample queries that an analyst could run.
    b. Examples: trial counts by recruitment status per year; top sponsors; trials involving a particular condition.
5. **Documentation & Cost**
    a. Document every assumption.
    b. Discuss resource usage: compute, storage, data-transfer costs, licensing if any.
    C. Explain how your design minimises unnecessary expense (e.g., serverless, autoscaling, columnar storage).

5. Evaluation Criteria

1. **Data Handling Quality** – Robustness of ingestion, cleaning, and error handling.
2. **Schema & Data Model** – Clarity, efficiency, and scalability of the warehouse design.
3. **Scalability & Robustness** – Ability to handle more data sources or higher volumes with minimal change.
4. **Cost Awareness** – Thoughtful, realistic cost analysis and optimization strategies.
5. **Code Quality** – Readability, modularity, configuration management, logging, and tests (if included).
6. **Documentation & Communication** – Clear explanation of design choices, assumptions, and trade-offs.
7. **Completeness & Creativity** – Fulfilment of core requirements; innovative but practical solutions.

Deliverable Format

1. You can use any and all tools available at your disposal, there are no limitations
2. This exercise is about research, ownership and clear communications (It's not about the size of the research document but the ability to guide your tech team and provide value to your customer)
3. Be as creative as you'd like, we welcome crazy ideas and risks

Once complete, please prepare to discuss your solution in a follow-up interview. We're excited to see how you approach this complex problem and turn ambiguity into a structured plan. Good luck!