**Prompt**:

You are a domain-specific \*\*workflow\*\* generator which takes a \*\*user query\*\* as input and generate a \*\*workflow\* as output. The domain is for analyzing yield based on test data collected from a semiconductor chip production line.

A \*\*workflow\*\* contains a sequence of \*\*Steps\*\*. When you generate the \*\*workflow\*\*, please try to use \*\*Steps\*\* that are as simple and specific as possible. Also, before the \*\*workflow\*\*, please generate a \*\*thought\*\* process behind the workflow and an \*\*output message\*\* after the workflow.

Please output in the following format:

\*\*Thought\*\*: your thought here.  
\*\*Workflow\*\*:  
Step: The task description: The step description.

\*\*Output Message\*\*: your output message here.

Below is a description about the domain of semiconductor test data analytics I am interested in.

\*\*Description\*\*:

<Insert Scope Description>

### Example 1

User Query:  
Show the overall yield performance of the latest lot.

\*\*Thought\*\*:  
The objective is to display the overall yield performance for the latest lot. The workflow should start by identifying and extracting the relevant test data for the latest lot. Once the data is isolated, wafer-level yields need to be calculated (if not pre-calculated) and then aggregated to obtain the lot-level yield. A comparison against the target yield can also be incorporated to provide context. Finally, visualizing the overall performance will help in quickly assessing the yield status.

\*\*Workflow\*\*:

Step 1: \*\*Identify Latest Lot\*\*: Query the test data repository to find the most recent lot.  
Step 2: \*\*Extract Wafer Test Data\*\*: Retrieve all wafer-level test records associated with the latest lot.  
Step 3: \*\*Calculate Wafer-Level Yields\*\*: For each wafer, calculate yield by determining the ratio of passing dies to total dies.  
Step 4: \*\*Aggregate to Lot-Level Yield\*\*: Combine wafer-level yields to compute the overall lot yield.  
Step 5: \*\*Compare Against Target\*\*: Check if the computed lot yield meets or exceeds the predefined target yield (e.g., 95%).  
Step 6: \*\*Visualize Yield Performance\*\*: Generate a summary report or chart displaying the overall lot yield and the status relative to the target.

\*\*Output Message\*\*:  
The overall yield performance for the latest lot has been computed and visualized. The calculated yield is now available along with a comparison to the target yield, providing clear insight into the production quality.

### Example 2  
User Query:  
List weekly wafer-level yield values.

**\*\*Thought\*\***:  
To list weekly wafer-level yield values, we need to segment the test data by week and then compute the yield for each wafer within those weekly intervals. This involves identifying the appropriate weekly timeframes, extracting wafer-level test data for each week, calculating yields by determining the ratio of passing dies to total dies, and finally organizing these computed values in a list or table. This structured approach ensures we capture the variations and trends on a weekly basis.

**\*\*Workflow\*\***:  
Step 1: \*\***Define Weekly Timeframes\*\***: Determine the start and end dates for each week in the test data period.  
Step 2: \*\***Retrieve Wafer Test Data\*\***: Query the test data repository to extract wafer-level test records for each defined weekly interval.  
Step 3: \*\***Calculate Wafer Yields\*\***: For every wafer in each week, compute the yield as the ratio of passing dies to total dies.  
Step 4: \*\***Organize Weekly Yield Values\*\***: Aggregate the computed yield values into a list or table, grouped by week.  
Step 5: \*\***Visualize or Report Data\*\***: Optionally generate a visual summary (e.g., chart or report) to clearly display the weekly wafer-level yields.

**\*\*Output Message\*\***:  
The weekly wafer-level yield values have been successfully computed and organized. Each week’s data is now available for review, enabling further analysis of yield trends over time.

### Example 3  
User Query:  
Identify trends in wafer-level yield over the last quarter.

**\*\*Thought\*\***:  
The goal is to analyze wafer-level yield trends over the last quarter. This involves extracting test data for the last three months, computing wafer-level yields for each testing event, and then aggregating these yields over time (e.g., weekly or monthly) to observe trends. The workflow should include data extraction, calculation of yields, trend analysis (including statistical evaluation or visual pattern recognition), and visualization of the trend for easy interpretation.

**\*\*Workflow\*\***:  
Step 1: \*\***Extract Quarterly Data\*\***: Query the test data repository to retrieve all wafer-level test records from the last quarter.  
Step 2: \*\***Segment Data by Timeframe\*\***: Divide the extracted data into regular intervals (e.g., weekly or monthly) to facilitate trend analysis.  
Step 3: \*\***Calculate Wafer-Level Yields\*\***: For each wafer in every time segment, compute the yield as the ratio of passing dies to total dies.  
Step 4: \*\***Aggregate Yields Over Time\*\***: Organize the computed yields by time segment to generate a trend dataset for the quarter.  
Step 5: \*\***Perform Trend Analysis\*\***: Apply statistical methods (e.g., moving average or regression analysis) to identify upward, downward, or stable trends in the wafer yields.  
Step 6: \*\***Visualize Trends\*\***: Create charts or graphs that display the yield trends over the quarter, highlighting any significant changes or patterns.  
Step 7: \*\***Generate Analysis Report\*\***: Compile the findings and visualizations into a report for further review and decision-making.

**\*\*Output Message\*\***:  
The wafer-level yield trends over the last quarter have been identified and analyzed. The workflow has computed yields, aggregated them over time, and produced visual trend analyses to clearly illustrate performance changes across the quarter.  
  
### Example 4  
User Query:  
Compare yield performance between two consecutive lots.  
  
**\*\*Thought\*\***:  
To compare yield performance between two consecutive lots, the workflow must first identify and extract the test data for the two lots. After isolating the relevant wafer-level test records for each lot, the next step is to calculate individual wafer yields and then aggregate these values to obtain a lot-level yield for each. Finally, a direct comparison is made between the two computed yields to highlight any differences or trends, potentially including visual representations to aid interpretation.

**\*\*Workflow\*\***:  
Step 1: \*\***Identify Consecutive Lots\*\***: Query the test data repository to locate and select the two most recent, consecutive lots.  
Step 2: \*\***Extract Wafer Test Data\*\***: Retrieve wafer-level test records corresponding to each of the two identified lots.  
Step 3: \*\***Calculate Wafer-Level Yields\*\***: For each wafer in both lots, compute the yield as the ratio of passing dies to total dies.  
Step 4: \*\***Aggregate to Lot-Level Yields\*\*:** Combine the wafer yields within each lot to determine the overall yield performance for each lot.  
Step 5: \*\***Compare Yields\*\*:** Directly compare the two lot-level yields, highlighting the differences and any deviations from target yield thresholds.  
Step 6: \*\***Visualize Comparison\*\*:** Generate comparative charts or graphs to visually display the yield performance of the two consecutive lots.  
Step 7: \*\***Report Findings\*\*:** Compile the calculated data and visualizations into a clear, concise report outlining the differences in yield performance.

**\*\*Output Message\*\***:  
The yield performance between the two consecutive lots has been successfully compared. The workflow has calculated and visualized the lot-level yields, enabling an effective side-by-side evaluation of performance differences.  
  
User Query:

**<Insert query here>**