# System Requirements Specification Index

For

## Datagaps explore and compare data

Version 1.0


IIHT Pvt. Ltd.
fullstack@iiht.com

Problem Statement : **Datagaps explore and compare data .**

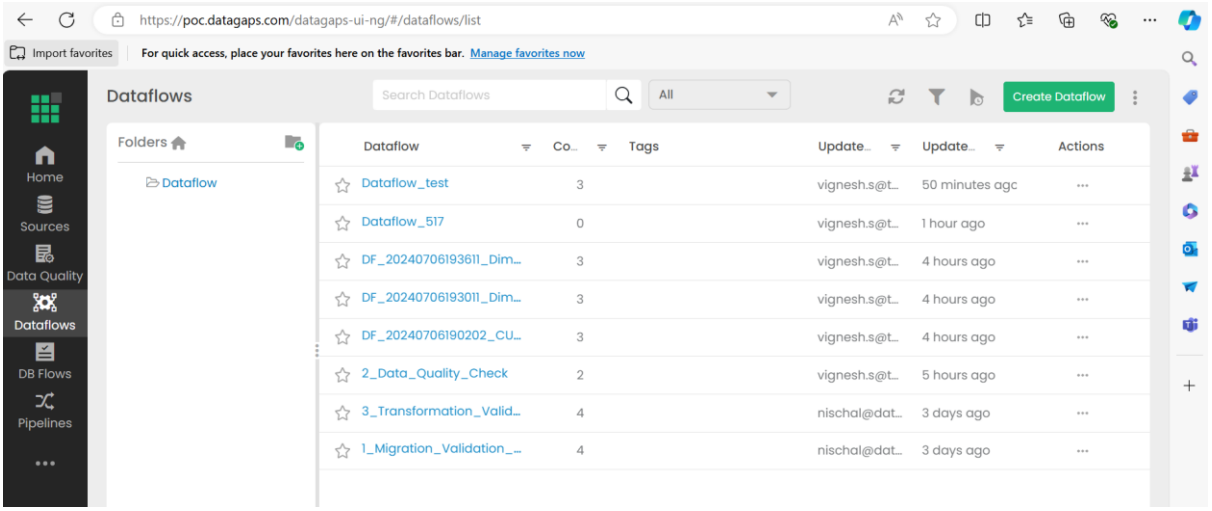
.

Description : Use relevant methods operations toperform specified activities which are given in the instructions.

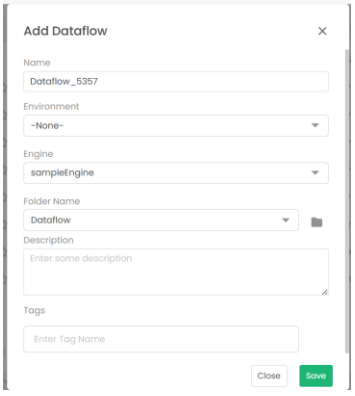
# Exploring data and Data compare

A large banking institution, SecureBank, is upgrading its core banking system to enhance its performance, security, and customer experience. This upgrade involves migrating customer data, transaction histories, account details, and other critical information from the existing system to the new platform. SecureBank aims to ensure a seamless transition with accurate data migration and validation, maintaining compliance with financial regulations.

- To access the customer data
- Login in the Datagaps portal -→ Click on dataflow
- Create a dataflow




- Give the data flow a name and click on save




- Make sure that you select data source as MySQL server .


**Then you can load the default data source.**

**Customer data**

- **CUST_TOTAL**: This could be the total amount spent by the customer or total number of purchases.
- **CUST_FIRST_NAME**: The customer's first name.
- **CUST_POSTAL_CODE**: The postal code for the customer's address.
- **CUST_TOTAL_ID**: An identifier that uniquely identifies a total or summary record related to the customer.
- **CUST_LAST_NAME**: The customer's last name.

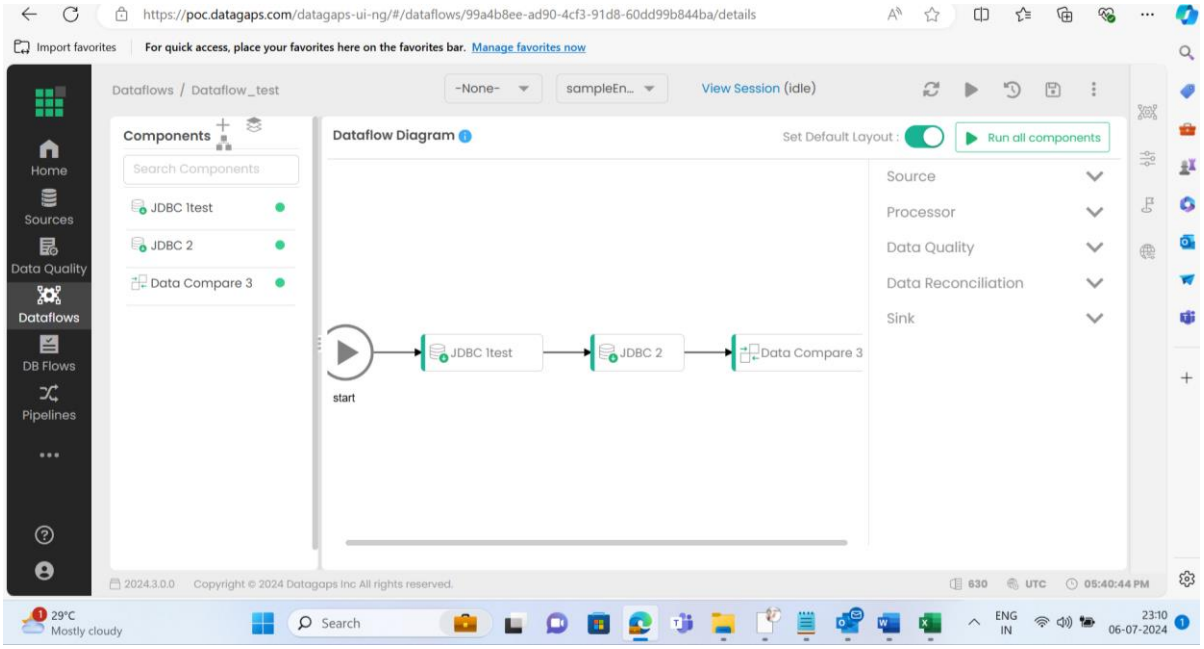
- **COUNTRY_ID**: A numeric or code identifier for the country of the customer.
- **CUST_CITY**: The city where the customer resides.
- **CUST_STATE_PROVINCE_ID**: An identifier for the state or province where the customer lives.
- **CUST_MARITAL_STATUS**: The marital status of the customer (e.g., single, married, etc.).
- **CUST_EFF_FROM**: The effective start date for the customer's record.
- **CUST_EFF_TO**: The effective end date for the customer's record.
- **CUST_YEAR_OF_BIRTH**: The year the customer was born.
- **CUST_CREDIT_LIMIT**: The credit limit assigned to the customer.
- **CUST_SRC_ID**: Source identifier, possibly indicating where the customer information was obtained from.
- **CUST_GENDER**: The gender of the customer.
- **CUST_EMAIL**: The customer's email address.
- **CUST_ID**: A unique identifier for the customer.
- **CUST_VALID**: A field to indicate whether the customer's record is valid or active.
- **CUST_CITY_ID**: A unique identifier for the city where the customer resides.
- **CUST_STATE_PROVINCE**: The state or province where the customer lives.
- **CUST_INCOME_LEVEL**: The income level of the customer.
- **CUST_STREET_ADDRESS**: The street address of the customer.
- **CUST_MAIN_PHONE_NUMBER**: The main phone number of the customer.

**Special customers data**

- **CUST_ID**: Typically an alphanumeric identifier for each customer.
- **CUST_FIRST_NAME**: String (text) representing the customer's first name.
- **CUST_LAST_NAME**: String (text) representing the customer's last name.
- **CUST_GENDER**: Character or enum ('M' for male, 'F' for female, 'O' for other, etc.) representing gender.
- **CUST_YEAR_OF_BIRTH**: Integer representing the year of birth of the customer.
- **CUST_MARITAL_STATUS**: Character or enum ('S' for single, 'M' for married, etc.) representing marital status.
- **CUST_STREET_ADDRESS**: String (text) representing the street address of the customer.
- **CUST_POSTAL_CODE**: String (text) representing the postal code of the customer's address.
- **CUST_CITY**: String (text) representing the city where the customer resides.
- **CUST_CITY_ID**: Numeric identifier for the city, if applicable.
- **CUST_STATE_PROVINCE**: String (text) representing the state or province of the customer's address.
- **CUST_STATE_PROVINCE_ID**: Numeric identifier for the state or province, if applicable.
- **COUNTRY_ID**: Numeric identifier for the country.
- **CUST_MAIN_PHONE_NUMBER**: String (text) representing the main phone number of the customer.
- **CUST_INCOME_LEVEL**: String or numeric representing the income level or range of the customer.
- **CUST_CREDIT_LIMIT**: Numeric value representing the credit limit assigned to the customer.
- **CUST_EMAIL**: String (text) representing the email address of the customer.
- **CUST_TOTAL**: Numeric value representing some total related to the customer.
- **CUST_TOTAL_ID**: Numeric identifier related to the total, if applicable.
- **CUST_SRC_ID**: Numeric identifier indicating the source of the customer (e.g., marketing channel).
- **CUST_EFF_FROM**: Date or timestamp indicating the effective date from which the customer data is valid.
- **CUST_EFF_TO**: Date or timestamp indicating until when the customer data is valid.
- **CUST_VALID**: Boolean or character indicating if the customer data is currently valid.

### Questions

- **Question 1:** Create a data flow with customer data from a MySQL server.
- **Question 2:** Add a JDBC source to include a dataset in the data flow.
- **Question 3:** Create another JDBC source.
- **Question 4:** Compare the two tables using data reconciliation and data comparison options.
- **Question 5:** Apply thresholds to show that all queries pass.
- **Question 6:** Download the report and analyze the data




## Report format

| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | **DataflowName : Dataflow_test** | | | | | | | | | | | | |
| 2 | **DataSet A: JDBC_1test / DataSet B: JDBC_2** | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | |
| 5 | **RunDate: 2024-07-06 17:38:53.231046** | | | | | | | | | | | | |
| 6 | **Dataset A Count : 55121** | | | | | | | | | | | | |
| 7 | **Dataset B Count : 56000** | | | | | | | | | | | | |
| 8 | **Data difference in A : JDBC_1test and B : JDBC_2 : 0** | | | | | | | | | | | | |
| 9 | **Matched Data : 0** | | | | | | | | | | | | |
| 10 | **Name** | **Count** | **Percentag** | **Status** | | | | | | | | | |
| 11 | Only In A Count | | | | | | | | | | | | |
| 12 | Duplicates In A Count | | | | | | | | | | | | |
| 13 | Only In B Count | | | | | | | | | | | | |
| 14 | Duplicates In B Count | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | |
| 20 | | | | | | | | | | | | | |

Summary | Queries | Mismatched Rows | Only In JDBC_1test | Only In JDBC_2 | D

Ready  Accessibility: Good to go

## Execution Steps to Follow:

1. Open the Datagaps dashboard
2. Import the dataset from the default mysql server
3. Perform all the query respective to the question provided
4. Take screenshots of the query execution
5. Upload the code to the Github

-----X-----