**Databricks Assignment**

**Question 1:**

**Please find the CSV files (Employee, Department, Country)**

**Questions:**

• Create 3 folders as source\_to\_bronze, bronze\_to\_silver, silver\_to\_gold. DONE

• Create 4 notebooks in this respective order. DONE

• 2 Notebooks named in source\_to\_bronze as utils (add all common functions in this notebook) and employee\_source\_to\_bronze (driver notebook) DONE

• 1 Notebook in bronze to silver as employee\_bronze\_to\_silver DONE

• 1 Notebook in silver to gold as employee\_silver\_to\_gold DONE

• Read the 3 datasets as Dataframe in **employee\_source\_to\_bronze**, call utils notebook in this notebook,

• and write to a location in DBFS,

as /source\_to\_bronze/file\_name.csv (employee, department\_df, country\_df) as CSV format.

• In **employee\_bronze\_to\_silver**, call utils notebook in this notebook.

Read the file located in DBFS location source\_to\_bronze with as data frame different read methods using custom schema.

• convert the Camel case of the columns to the snake case using UDF.

• Add the **load\_date** column with the current date.

• The primary key is EmployeeID, the Database name is Employee\_info, Table name is dim\_employee.

• write the DF as a delta table to the location /silver/db\_name/table\_name.

• In gold notebook **employee\_silver\_to\_gold**, call utils notebook in this notebook  
 Read the table stored in a silver layer as DataFrame and select the columns based on the following requirements.

• Requirements:

• Find the salary of each department in descending order.

• Find the number of employees in each department located in each country.

• List the department names along with their corresponding country names.

• What is the average age of employees in each department?

• Add the at\_load\_date column to data frames.

• Write the df to dbfs location /gold/employee/table\_name(fact\_employee) with overwrite and replace where condition on at\_load\_date.

**Question 2:**

Api: https://reqres.in/api/users?page=2  
 drop "page”, "per\_page", "total", "total\_pages" and complete block of support.

• Fetch the data from the given API by passing the parameter as a page and retrieving the data till the data is empty.

• Read the data frame with a custom schema

• Flatten the dataframe

• Derive a new column from email as site\_address with values(reqres.in)

• Add load\_date with the current date

• Write the data frame to location in DBFS as /db\_name /table\_name with

Db\_name as site\_info and table\_name as person\_info with delta format and overwrite mode.