Zoom Car Data Processing Pipeline

- **Dataset Details** - There will be two separate locations in the gcp storage bucket where daily files of **car_booking** and **customers** will arrive.
    - Dataset Names:
        - zoom_car_bookings_yyyymmdd.json
        - zoom_car_customers_yyyymmdd.json

    Sample Data:

    zoom_car_bookings_yyyymmdd.json write a python script to mock this data or use chatgpt to generate many such files.


```
[
  {
    "booking_id": "B001",
    "customer_id": "C001",
    "car_id": "CAR123",
    "booking_date": "2024-07-20",
    "start_time": "2024-07-21T10:00:00Z",
    "end_time": "2024-07-21T18:00:00Z",
    "total_amount": 150.75,
    "status": "completed"
  },
  {
    "booking_id": "B002",
    "customer_id": "C002",
    "car_id": "CAR456",
    "booking_date": "2024-07-20",
    "start_time": "2024-07-21T12:00:00Z",
    "end_time": "2024-07-21T16:00:00Z",
    "total_amount": 80.50,
    "status": "cancelled"
  }
]
```


    zoom_car_customers_yyyymmdd.json write a python script to mock this data or use chatgpt to generate many such files.

```
[
  {
    "customer_id": "C001",
    "name": "John Doe",
    "email": "john.doe@example.com",
    "phone_number": "1234567890",
    "signup_date": "2024-01-15",
    "status": "active"
  },
  {
    "customer_id": "C002",
    "name": "Jane Smith",
    "email": "jane.smith@example.com",
    "phone_number": "0987654321",
    "signup_date": "2023-12-22",
    "status": "inactive"
  }
]
```


- **PySpark Notebooks**

Create two separate PySpark notebooks to process the bookings and customers datasets.

- **Notebook 1: Process Zoom Car Bookings**
  - Read JSON file for the current date.
  - Perform data cleaning and validation:
  - Remove records with null values in critical fields (booking_id, customer_id, car_id, booking_date).
  - Validate date formats.
  - Ensure status is one of the predefined statuses (e.g., completed, cancelled, pending).
  - Load cleaned data into the staging_bookings_delta table.

- **Notebook 2: Process Zoom Car Customers**
  - Read JSON file for the current date.
  - Perform data cleaning and validation:

        - Remove records with null values in critical fields (customer_id, name, email).
        - Validate email formats.
        - Ensure status is one of the predefined statuses (e.g., active, inactive).
        - Load cleaned data into the staging_customers_delta table.

    - **Parameterized PySpark Notebooks**
        - Both notebooks should accept the current date as a parameter and read the corresponding file:

- **Apply Transformations**

    - **Bookings Data Transformations:**
        - Parse start_time and end_time into separate date and time columns.
        - Calculate the total duration of each booking.

    - **Customers Data Transformations:**
        - Normalize phone numbers to a standard format.
        - Calculate customer tenure from signup_date.

- **Merging Data in Target Delta Table**
    - Create a third notebook to read the staged data and perform merge operations.
    - **Merge Conditions:**
        - Update: If booking_id or customer_id exists in the target table, update the existing records.
        - Insert: If booking_id or customer_id does not exist, insert new records.
        - Delete: If the status of a booking is cancelled, delete the record from the target table.

- **Databricks Workflow**
    - Create a Databricks job to automate this workflow:
        - Step 1: Trigger Process Zoom Car Bookings notebook.
        - Step 2: Trigger Process Zoom Car Customers notebook.
        - Step 3: Trigger Merge Data notebook.

  Manually trigger this workflow daily by passing the current date parameter.

- **Deliverables**
    - Datasets: Provide sample JSON files.
    - Notebooks: Provide PySpark notebooks for bookings, customers, and merging data.
    - Job Flow JSON: Provide the JSON configuration for the Databricks job workflow.