**Case Study: Baggage Tracking System for Airlines**

**Problem Statement:**

Design and implement a Baggage Tracking System for Airlines using Oracle SQL and PL/SQL. The system will be used to track baggage, update baggage status, and generate tracking reports. Your task is to create the necessary database schema, populate the database with sample data, and develop PL/SQL procedures to handle baggage tracking, status updates, and report generation.

**Requirements:**

1. **Baggage Tracking**:
   * Implement the functionality to track baggage by baggage tag numbers.
   * Ensure that each baggage has attributes such as BAGGAGE\_TAG\_NUMBER, FLIGHT\_NUMBER, DESTINATION, CURRENT\_LOCATION, STATUS, and LAST\_UPDATED.
2. **Status Update**:
   * Implement the functionality to update the status of baggage.
   * Ensure that each status update record has attributes such as UPDATE\_ID, BAGGAGE\_TAG\_NUMBER, UPDATED\_BY, UPDATE\_DATE, and NEW\_STATUS.
3. **Report Generation**:
   * Implement the functionality to generate tracking reports.
   * Reports should include details such as total baggage tracked, baggage at each location, and status summaries.

**Tasks:**

1. **Design the Database Schema**:
   * Create the BaggageTracking, StatusUpdates, and TrackingReports tables with the appropriate fields and constraints.
   * Define primary keys and foreign keys to maintain data integrity.
2. **Populate the Database with Sample Data**:
   * Insert sample records into the BaggageTracking and StatusUpdates tables to facilitate testing of the system.
3. **Develop PL/SQL Procedures**:
   * Create a procedure to track baggage. The procedure should insert new baggage records.
   * Create a procedure to update the status of baggage. The procedure should insert status update records and update the current location and status of baggage.
   * Create a procedure to generate tracking reports. The procedure should aggregate data to provide insights into baggage tracking and status.

**Expected Outcomes:**

1. **BaggageTracking Table**:
   * Contains all information about the tracked baggage.
2. **StatusUpdates Table**:
   * Tracks the status updates for each baggage.
3. **TrackingReports Table**:
   * Stores the generated tracking reports.
4. **PL/SQL Procedures**:
   * Efficiently track baggage, update statuses, and generate reports, maintaining accurate records in the database.

**Deliverables:**

1. SQL scripts to create the BaggageTracking, StatusUpdates, and TrackingReports tables.
2. SQL scripts to insert sample data into the BaggageTracking and StatusUpdates tables.
3. PL/SQL scripts for the procedures to handle baggage tracking, status updates, and generate tracking reports.
4. Documentation explaining how to set up and use the system, including how to run the PL/SQL procedures.

**Database Schema:**

1. **BaggageTracking Table**:
   * **BAGGAGE\_TAG\_NUMBER**: Varchar2(50), Primary Key
   * **FLIGHT\_NUMBER**: Varchar2(50)
   * **DESTINATION**: Varchar2(50)
   * **CURRENT\_LOCATION**: Varchar2(50)
   * **STATUS**: Varchar2(50)
   * **LAST\_UPDATED**: Date
2. **StatusUpdates Table**:
   * **UPDATE\_ID**: Number, Primary Key
   * **BAGGAGE\_TAG\_NUMBER**: Varchar2(50), Foreign Key References BaggageTracking(BAGGAGE\_TAG\_NUMBER)
   * **UPDATED\_BY**: Varchar2(50)
   * **UPDATE\_DATE**: Date
   * **NEW\_STATUS**: Varchar2(50)
3. **TrackingReports Table**:
   * **REPORT\_ID**: Number, Primary Key
   * **REPORT\_DATE**: Date
   * **TOTAL\_BAGGAGE**: Number
   * **BAGGAGE\_AT\_LOCATION**: Clob
   * **STATUS\_SUMMARY**: Clob

**Case Study Task:**

* **Design**: Create the database schema as provided.
* **Implement**: Insert sample data into the BaggageTracking and StatusUpdates tables.
* **Develop**: Write PL/SQL procedures for tracking baggage, updating statuses, and generating tracking reports.
* **Test**: Test the procedures with various scenarios (e.g., tracking baggage, updating statuses, generating reports, ensuring proper updates).