Multi-Specialty Hospital Chain - ETL Process and Database Design (PostgreSQL)

Overview:
This document outlines the process for managing customer data in a multi-specialty hospital chain, where patients from various countries are categorized and managed based on their country of origin. The system handles billions of customer records daily and splits them into country-specific tables while performing validations and transformations.

The process is built using PostgreSQL and covers:
- Database schema creation
- Data extraction, transformation, and loading (ETL)
- Handling large datasets
- Edge case handling (duplicate customers, missing fields)
- Derived columns for customer age and days since last consultation

Requirements:
1. Split customer data by country and store it in corresponding country-specific tables.
2. Handle large datasets with billions of rows processed daily.
3. Perform transformations such as:
   - Calculating age based on the customer's date of birth.
   - Calculating the number of days since last consultation.
4. Ensure the latest consultation record is considered if a customer has visited multiple times from different locations.
5. Validate that required fields are populated.

Database Schema:

Staging Table:
The Staging_Customers table is used to load raw data before processing. This is where the initial data from files is loaded.

```
CREATE TABLE Staging_Customers (
    customer_name VARCHAR(255) NOT NULL,
    customer_id VARCHAR(18) NOT NULL,
    open_date DATE NOT NULL,
    last_consulted_date DATE,
    vaccination_type CHAR(5),
    doctor_name VARCHAR(255),
    state CHAR(5),
    country CHAR(5),
    post_code INT,
    date_of_birth DATE,
    is_active CHAR(1),
    PRIMARY KEY (customer_id)
);
```

Country-Specific Table:
Each country will have its own table. Below is an example of how the structure of the Table_India looks:

```
CREATE TABLE Table_India (
    customer_name VARCHAR(255),
    customer_id VARCHAR(18),
    open_date DATE,
    last_consulted_date DATE,
    vaccination_type CHAR(5),
    doctor_name VARCHAR(255),
    state CHAR(5),
    dob DATE,
    is_active CHAR(1),
    PRIMARY KEY (customer_id)
);
```

Derived Columns (Age and Days Since Last Consulted):
To calculate derived columns like age and days_since_last_consulted, use a VIEW in PostgreSQL.

```
CREATE VIEW Table_India_Derived AS
SELECT
    customer_name,
    customer_id,
    open_date,
    last_consulted_date,
    vaccination_type,
    doctor_name,
    state,
    dob,
    is_active,
    EXTRACT(YEAR FROM AGE(dob)) AS age,
    EXTRACT(DAY FROM NOW() - last_consulted_date) AS days_since_last_consulted
FROM Table_India;
```

ETL Process:
The source data will be provided as CSV files. Use PostgreSQL's COPY command to load the data into the staging table. I have created a dummy file by the name of Staging_Customers_Dummy_Data.csv which is used in this scenario.

```
COPY Staging_Customers(customer_name, customer_id, open_date, last_consulted_date,
vaccination_type,
    doctor_name, state, country, post_code, date_of_birth, is_active)
FROM '~/Documents/Incubyte/Staging_Customers_Dummy_Data' DELIMITER '|' CSV HEADER;
```

Transform Data:

Data transformations, such as calculating age and days since last consulted, will be handled through VIEW as described above.

Load Data into Country-Specific Tables:
Move the data from the Staging_Customers table to the appropriate country-specific table. Example for India:

```
INSERT INTO Table_India (customer_name, customer_id, open_date, last_consulted_date,
vaccination_type,
   doctor_name, state, dob, is_active)
SELECT customer_name, customer_id, open_date, last_consulted_date, vaccination_type,
   doctor_name, state, date_of_birth, is_active
FROM Staging_Customers
WHERE country = 'IND';
```

Edge Case Handling:
Duplicate Customers:
If a customer has multiple consultation records, use the latest consultation date. Delete old records and insert the latest ones:

```
DELETE FROM Table_India
WHERE customer_id IN (
    SELECT customer_id
    FROM Staging_Customers
    WHERE country = 'IND'
    AND last_consulted_date < (SELECT MAX(last_consulted_date) FROM Staging_Customers
WHERE customer_id = Staging_Customers.customer_id)
);

INSERT INTO Table_India (customer_name, customer_id, open_date, last_consulted_date,
vaccination_type,
    doctor_name, state, dob, is_active)
SELECT customer_name, customer_id, open_date, last_consulted_date, vaccination_type,
    doctor_name, state, date_of_birth, is_active
FROM Staging_Customers
WHERE country = 'IND';
```

Handling Large Data Volumes:
- Indexes: Create indexes to optimize data retrieval for large datasets.

```
  CREATE INDEX idx_customer_id ON Staging_Customers(customer_id);
  CREATE INDEX idx_country ON Staging_Customers(country);
```

- Batch Inserts: Load data in smaller batches to avoid memory issues.

- Partitioning: Use table partitioning to improve query performance, for example, partitioning Table_India by state.

```
  CREATE TABLE Table_India_Partitioned (
      customer_name VARCHAR(255),
      customer_id VARCHAR(18),
      open_date DATE,
      last_consulted_date DATE,
      vaccination_type CHAR(5),
      doctor_name VARCHAR(255),
      state CHAR(5),
      dob DATE,
      is_active CHAR(1),
      PRIMARY KEY (customer_id, state)
  ) PARTITION BY LIST(state);

  CREATE TABLE Table_India_State_SA PARTITION OF Table_India_Partitioned FOR VALUES IN
('SA');
```

```
CREATE TABLE Table_India_State_VIC PARTITION OF Table_India_Partitioned FOR VALUES
IN ('VIC');
```

Validations:
Data Integrity:
Ensure mandatory fields like customer_name, customer_id, and open_date are populated.

```
SELECT * FROM Staging_Customers
WHERE customer_name IS NULL OR customer_id IS NULL OR open_date IS NULL;
```

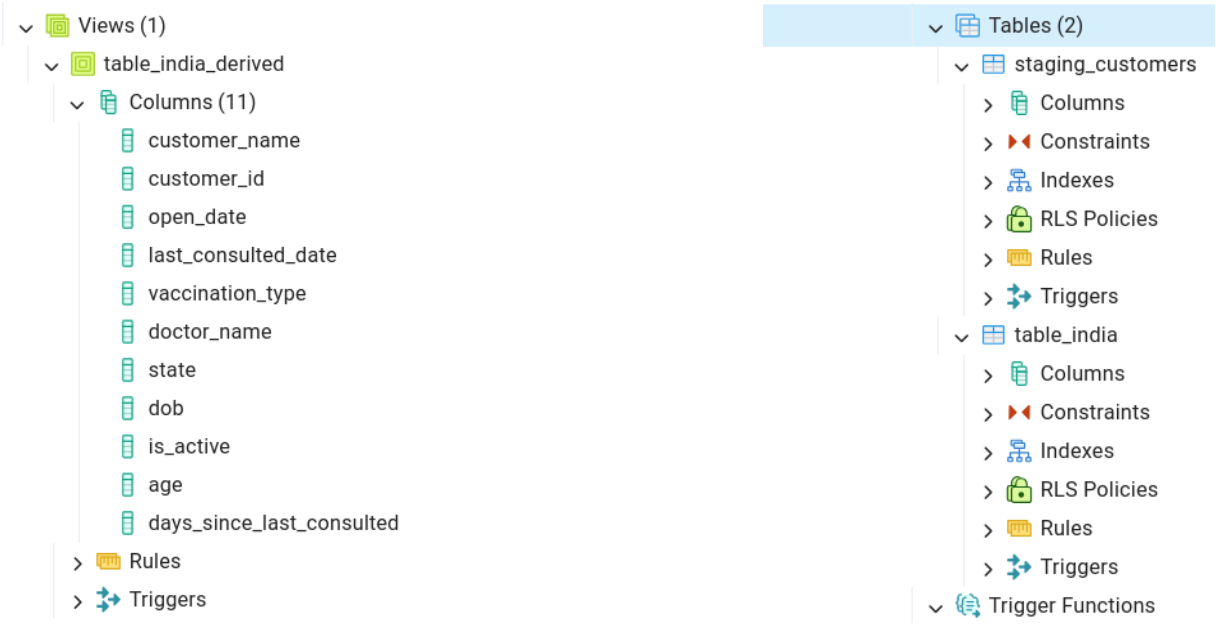
Date Validations:
Check for invalid dates in fields like open_date.

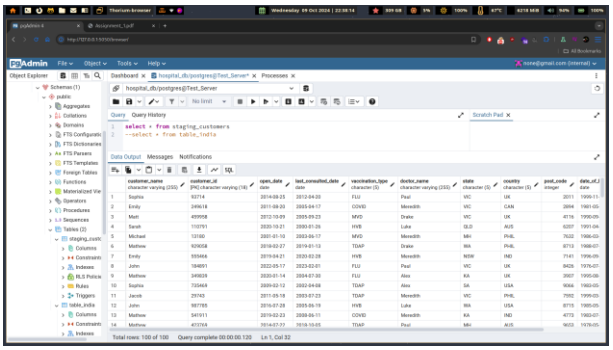
```
SELECT * FROM Staging_Customers WHERE open_date > NOW();
```

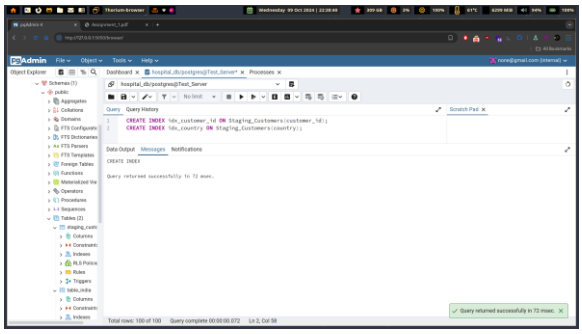
Active Status Validation:
Ensure the is_active field contains valid data ('A' or 'I').

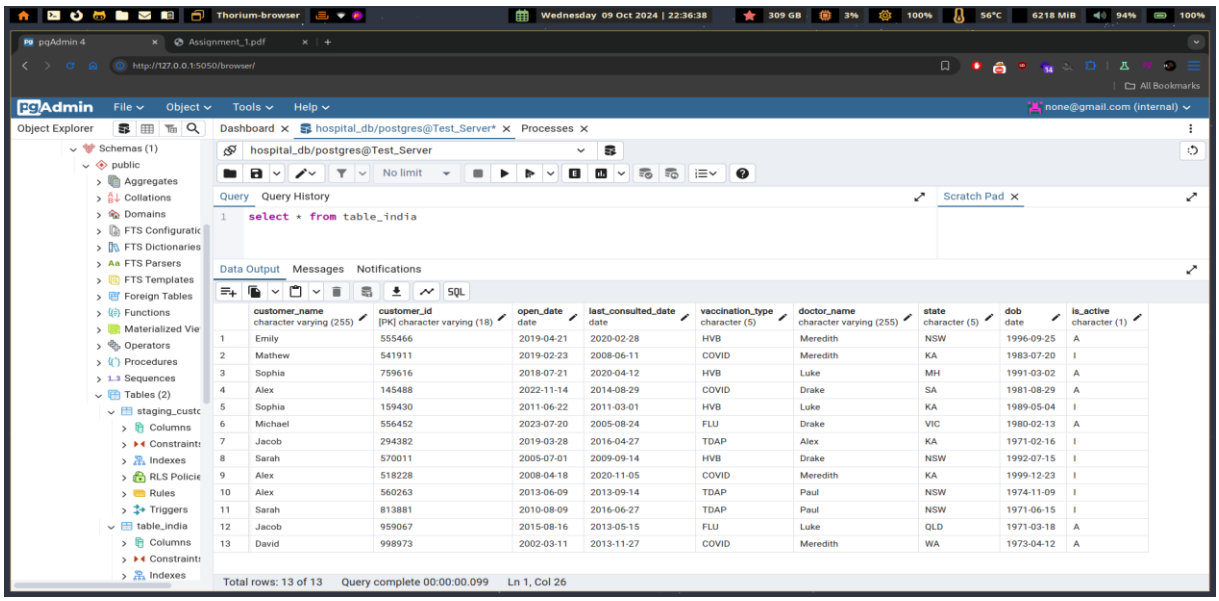
```
SELECT * FROM Staging_Customers WHERE is_active NOT IN ('A', 'I');
```

Appendix:


Views (1)
table_india_derived
Columns (11)
customer_name
customer_id
open_date
last_consulted_date
vaccination_type
doctor_name
state
dob
is_active
age
days_since_last_consulted
Rules
Triggers
Tables (2)
staging_customers
Columns
Constraints
Indexes
RLS Policies
Rules
Triggers
table_india
Columns
Constraints
Indexes
RLS Policies
Rules
Triggers
Trigger Functions






pgAdmin 4
Assignment_1.pdf
http://127.0.0.1:5050/browser/
pgAdmin
File
Object
Tools
Help
none@gmail.com (internal)
Object Explorer
Dashboard
hospital_db/postgres@Test_Server*
Processes
hospital_db/postgres@Test_Server
No limit
Query
Query History
Scratch Pad
select * from table_india
Data Output
Messages
Notifications
SQL
Schemas (1)
public
Aggregates
Collations
Domains
FTS Configuration
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Vie
Operators
Procedures
Sequences
Tables (2)
staging_custo
table_india
Columns
Constraints
Indexes
RLS Policie
Rules
Triggers
customer_name character varying (255)
customer_id [PK] character varying (18)
open_date date
last_consulted_date date
vaccination_type character (5)
doctor_name character varying (255)
state character (5)
dob date
is_active character (1)
1 Emily 555466 2019-04-21 2020-02-28 HVB Meredith NSW 1996-09-25 A
2 Mathew 541911 2019-02-23 2008-06-11 COVID Meredith KA 1983-07-20 I
3 Sophia 759616 2018-07-21 2020-04-12 HVB Luke MH 1991-03-02 A
4 Alex 145488 2022-11-14 2014-08-29 COVID Drake SA 1981-08-29 A
5 Sophia 159430 2011-06-22 2011-03-01 HVB Luke KA 1989-05-04 I
6 Michael 556452 2023-07-20 2005-08-24 FLU Drake VIC 1980-02-13 A
7 Jacob 294382 2019-03-28 2016-04-27 TDAP Alex KA 1971-02-16 I
8 Sarah 570011 2005-07-01 2009-09-14 HVB Drake NSW 1992-07-15 I
9 Alex 518228 2008-04-18 2020-11-05 COVID Meredith KA 1999-12-23 I
10 Alex 560263 2013-06-09 2013-09-14 TDAP Paul NSW 1974-11-09 I
11 Sarah 813881 2010-08-09 2016-06-27 TDAP Paul NSW 1971-06-15 I
12 Jacob 959067 2015-08-16 2013-05-15 FLU Luke QLD 1971-03-18 A
13 David 998973 2002-03-11 2013-11-27 COVID Meredith WA 1973-04-12 A
Total rows: 13 of 13
Query complete 00:00:00.099
Ln 1, Col 26