🧩 PostgreSQL Framework for Customer Identity Resolution (CIR)

Tận dụng sức mạnh ACID & Logic của PGSQL

🤔 Why PostgreSQL as the Core of CIR?

- CIR yêu cầu xử lý **Logic phức tạp**, **Transactional** & **Đảm bảo Data Integrity**.
- PostgreSQL, với các tính năng DB truyền thống + extension hiện đại, là nền tảng lý tưởng cho vai trò xử lý cốt lõi này.
- Kết hợp sức mạnh của Stored Procedures, Trigger và ACID.

🚀 8-Step Framework for CIR on PostgreSQL

Bước 1: Raw Data Ingestion

- **Mô tả:** Thu thập **data** khách hàng từ các **sources** (web, **app**, **CRM**...).
- **Thực thi:** Đưa **data** này vào một **staging table** (ví dụ: `raw_profiles`) trong **PostgreSQL**.
- **Công cụ:** Có thể dùng **Data Firehose** hoặc **Apache Kafka** làm lớp trung gian để **streaming data**.

Bước 2: Initiate Resolution

PROF

- **Mô tả:** Bắt đầu quy trình xử lý **CIR** cho **data** mới.
- **Thực thi:** Có hai cơ chế chính:
  - **Real-time: Trigger** (`AFTER INSERT/UPDATE`) trên **staging table** tự động gọi **Stored Procedure** xử lý cho từng bản ghi mới hoặc cập nhật.
  - **Batch/Schedule:** Một lịch trình hàng ngày (ví dụ: chạy bằng **Python script** hoặc **Lambda** lúc 2AM) quét các bản ghi còn lại trong **staging table** và gọi **Stored Procedure** để xử lý theo lô lớn. Cơ chế này có thể tạm vô hiệu hóa **trigger real-time** trong lúc chạy **batch**.
- (Tùy chọn) Sử dụng **status table** (`identity_resolution_status`) để theo dõi trạng thái và thời gian chạy của quy trình, giúp **trigger/schedule** phối hợp.
- **Ưu điểm PG: Trigger** mạnh mẽ, **Stored Procedure** tập trung **logic**.

Bước 3: Select Data for Processing

- **Mô tả:** Xác định bản ghi nào trong **staging** cần được **resolve**.

- **Thực thi:** Bên trong **Stored Procedure** xử lý **CIR**, các bản ghi từ **bảng staging** (`raw_profiles`) chưa được đánh dấu là đã xử lý sẽ được chọn ra bằng **SQL query**.
- Quá trình chọn có thể theo lô nhỏ (ví dụ: 100-1000 bản ghi) khi kích hoạt **real-time** hoặc theo lô lớn/toàn bộ khi chạy **batch**.
- **Ưu điểm PG: SQL query** linh hoạt, hỗ trợ `LIMIT`/`OFFSET` cho xử lý **batch** có kiểm soát.

Bước 4: Load Existing Context & Rules

- **Mô tả:** Lấy thông tin và các **rules** cần thiết từ **database** để thực hiện **resolve**.
- **Thực thi: Stored Procedure** truy vấn các bảng chính của hệ thống **CIR** trong **PostgreSQL** để lấy thông tin:
  - Các **master profiles** hiện có (`master_profiles`).
  - Các **profile links** giữa **master** và **alias**.
  - Cấu hình các **attributes** quan trọng và **rules** so khớp (`profile_attributes_config`).
- **Ưu điểm PG: JOIN** hiệu quả giữa các **relational tables**.

Bước 5: Execute Resolution Logic

- **Mô tả:** Áp dụng các **rules** phức tạp để so khớp (**matching**), liên kết (**linking**) và gộp (**merging**) **profiles**.
- **Thực thi:** Xảy ra hoàn toàn bên trong **Stored Procedure** trong **PostgreSQL**.
  - Sử dụng **procedural logic** (**PL/pgSQL**).
  - Áp dụng **fuzzy matching**, **conditional rules**.
  - Tích hợp **pgvector** cho so khớp dựa trên **embedding**.
- **Ưu điểm PG:** Tập trung **logic phức tạp**, **performance** cao cho các phép toán trong **DB**, hỗ trợ **extension**.

PROF

Bước 6: Persist Resolved State

- **Mô tả:** Ghi lại trạng thái **profile** đã **resolve** vào **database**.
- **Thực thi: Stored Procedure** cập nhật hoặc chèn mới các bản ghi vào bảng **`master_profiles`** và chèn các liên kết mới vào bảng **`profile_links`** (liên kết **raw profile** với **master profile**).
- **Ưu điểm PG:** Toàn bộ được thực hiện trong một **Transaction ACID** duy nhất, đảm bảo **data** luôn **consistent** và không bị mất mát/sai lệch do **race condition**.

Bước 7: Finalize Source Data

- **Mô tả:** Cập nhật trạng thái của **raw data** ban đầu.
- **Thực thi: Stored Procedure** đánh dấu các bản ghi tương ứng trong **bảng staging** (`raw_profiles`) là đã xử lý.

- **Ưu điểm PG:** Là một phần của **transaction** Bước 6, đảm bảo bản ghi chỉ được đánh dấu khi kết quả **resolve** đã được lưu thành công.

Bước 8: Expose Resolved Data

- **Mô tả:** Làm cho **data master profile** đã xử lý sẵn sàng cho các **applications** và **analytics**.
- **Thực thi: Data** nằm trực tiếp trong các bảng **PostgreSQL** (`master_profiles`, `profile_links`).
- **Ưu điểm PG:** Truy vấn trực tiếp bằng **SQL** tiêu chuẩn từ các công cụ **BI**, **applications** khác để xây dựng **Single Customer 360 View** và báo cáo.

Conclusion

- PostgreSQL cung cấp nền tảng vững chắc cho **core logic processing** và **accurate data management** trong **CIR**.
- Tận dụng **Stored Procedures & Triggers** cho **robust processing flow**.
- Đảm bảo **ACID** cho **critical data integrity**.
- Kết hợp tốt với các **external systems** (**Stream**, **Search Engine** như OpenSearch cho lớp **UI/Analytics**).