**Case Study: Building a Data Vault Raw Vault, PIT Table, Star View, and Hash Validation**

**Background**

A retail company wants to modernize its data warehouse by implementing a Data Vault 2.0 model. The initial focus is on tracking customer purchases to support historical analysis and improve reporting agility.

**Objective**

* Build a **Raw Vault** model to store source data for Customers, Products, and Purchases.
* Generate **Point-In-Time (PIT) tables** to simplify querying historical satellite data.
* Publish **zero-copy star schema views** for reporting.
* Validate **hash key uniqueness** and collision risks in the data.

**Data Sources**

* **Customer Source System**: Contains customer ID, name, address, and contact details.
* **Product Source System**: Product SKU, name, category, price.
* **Sales Transactions System**: Records purchase transactions including customer ID, product SKU, purchase date, quantity, and sales amount.

**Steps**

**Step 1: Design Raw Vault Model**

* **Hubs**:
  + Hub\_Customer with business key: Customer ID
  + Hub\_Product with business key: Product SKU
* **Link**:
  + Link\_Purchase representing purchase transactions connecting Hub\_Customer and Hub\_Product.
* **Satellites**:
  + Sat\_Customer\_Details for customer descriptive attributes (name, address, contact).
  + Sat\_Product\_Details for product descriptive attributes (name, category, price).
  + Sat\_Purchase\_Details for transactional attributes (purchase date, quantity, sales amount).

**Step 2: Load Raw Vault Data**

* Extract source data and load into Raw Vault hubs, links, and satellites.
* Generate hash keys by hashing business keys for hubs and concatenated hub keys for links.
* Populate audit columns: load date, record source, batch ID.
* Maintain full historization by inserting satellite records on each change.

**Step 3: Generate PIT Tables**

* Create PIT\_Customer table combining Hub\_Customer with latest Sat\_Customer\_Details records as of specific dates.
* Create PIT\_Product similarly for products.
* Create PIT\_Purchase that joins Link\_Purchase with related PIT customer and product data for quick historical snapshots.

**Step 4: Publish Zero-Copy Star Schema Views**

* Build **Dimension Views**:
  + Dim\_Customer combining hub and latest satellite attributes from PIT\_Customer.
  + Dim\_Product combining hub and satellite data from PIT\_Product.
* Build **Fact View**:
  + Fact\_Purchase combining Link\_Purchase, Dim\_Customer, and Dim\_Product with Sat\_Purchase\_Details via PIT\_Purchase.
* Ensure no physical duplication; use SQL views referencing Raw Vault and PIT tables.

**Step 5: Validate Hash Keys and Collision Testing**

* Run scripts to detect any duplicate hash keys in hubs and links.
* Test with large datasets for collision risks.
* If collisions are found, adjust hash key generation strategy (e.g., include delimiters or change hashing algorithm).
* Document results and confirm hash key uniqueness.

**Deliverables**

* SQL scripts or ETL workflows to load Raw Vault components.
* PIT table creation scripts and incremental update logic.
* Star schema views definitions.
* Hash validation scripts and collision testing report.
* Sample queries demonstrating historical data retrieval using PIT tables and star views.

**Expected Benefits**

* Full historization with auditability for regulatory compliance.
* Simplified and performant historical queries using PIT tables.
* Storage efficient star schema views without data duplication.
* Reliable key management with hash collision safety.