**Data Vault Concepts & Best Practices**

**1. Hub / Link / Satellite Rules**

**Hubs**

* Store unique business keys from source systems.
* Each hub row represents a single distinct business entity.
* No descriptive or transactional data stored in hubs.
* Surrogate key is created via hashing of business key(s).
* Load timestamp and record source included.
* Immutable after load; no updates, only inserts.

**Links**

* Represent relationships between hubs (many-to-many).
* Business keys from connected hubs combined and hashed to create a link key.
* May contain relationship-specific metadata (e.g., role, status).
* Load timestamp and record source included.
* Satellites attached for descriptive data about relationships.

**Satellites**

* Store descriptive, time-variant attributes related to hubs or links.
* Include load timestamp, record source, and possibly end date (for soft deletes).
* Can be split by granularities or business domains.
* Use hash diff keys to detect changes efficiently.
* Support historization of attribute changes.

**2. Hash Keys & Collision Testing**

* Hash keys are fixed-length surrogate keys generated from business key values using hashing algorithms like MD5, SHA-1, or SHA-256.
* Enable consistent and scalable integration of multiple source systems.
* Greatly reduce key storage size and simplify joins.
* Collision (two different inputs producing the same hash) is rare but must be tested during development.
* Collision Testing:
  + Generate hashes for large samples of data.
  + Verify uniqueness.
  + Tools/scripts can automate collision detection.
* Best practices include concatenating keys with delimiters before hashing.

**3. PIT (Point-In-Time) Tables**

* PIT tables provide a simplified snapshot of related satellite data as of a certain point in time.
* Useful for performance optimization when querying historical data.
* Pre-join satellites to hubs or links with the latest available satellite record per point in time.
* Can be physical tables or views, updated regularly via ETL.
* Reduce complex joins and speed up queries.

**4. Bridge Tables**

* Handle complex many-to-many relationships beyond links, especially for multi-valued attributes.
* Aggregate or filter link data for more meaningful queries.
* Help in creating drill-across capabilities in star schemas.
* Bridge tables sit between business vault and data marts or reporting layers.
* Example: A customer may have multiple contact methods — a bridge table can represent this.

**5. Raw Vault → Business Vault Pipeline**

* **Raw Vault**:
  + Stores unmodified, raw source data.
  + Maintains full historization and auditability.
  + Contains hubs, links, and satellites as-is from source.
* **Business Vault**:
  + Applies business rules and derives new attributes.
  + Adds conformed dimensions, calculated fields.
  + Creates PIT and bridge tables.
  + Enables easier and faster reporting.
* Pipeline Steps:
  + Extract and load data into Raw Vault.
  + Use automation macros to generate consistent keys and audit columns.
  + Transform data applying business logic in Business Vault.
  + Publish or expose data to star schemas or marts.

**6. Data Vault Automation Macros**

* Macros/scripts automate repetitive tasks in ETL processes.
* Common automated tasks include:
  + Generating hash keys from business keys.
  + Detecting changes via hash diffs.
  + Managing audit columns (load timestamp, record source).
  + Creating and updating PIT and bridge tables.
* Automation improves consistency, reduces errors, and speeds development.
* Can be implemented in SQL, Python, or integrated ETL tool scripting.

**7. Zero-Copy Star Schema Publishing**

* Star schemas are optimized for query performance but can duplicate data.
* Zero-copy approach builds star schemas as virtual views directly on Data Vault tables without physical copying.
* Saves storage space and maintenance effort.
* Offers near real-time data availability.
* Requires efficient database engine and indexing to maintain performance.

**8. Audit Columns**

* Essential for data governance, lineage, and troubleshooting.
* Common audit fields:
  + **Load Date / Timestamp:** When record was loaded.
  + **Record Source:** Originating system or file.
  + **Load End Date:** Indicates when a satellite record is superseded or deleted (soft delete).
  + **Batch or Run ID:** Identifies the ETL job run.
* Audit data helps validate data quality and trace historical changes.

**Additional Tips for Studying Data Vault**

* Understand the separation of concerns among hubs, links, and satellites.
* Practice modeling sample business scenarios using Data Vault principles.
* Explore hashing algorithms and practice creating hash keys.
* Study how PIT and bridge tables improve performance.
* Familiarize yourself with automation strategies to make Data Vault pipelines efficient.
* Review case studies or whitepapers on zero-copy star schemas.
* Learn about audit data requirements in enterprise environments.