Link definition

	Link	
PK	`FixedLinkHashKey`	Hash generated from the combination of all BKs in the link. Before inserting the relationship from a stage table, the ETL job has to confirm the relationship is unique. To do so, it compares the hashes generated from the relationships of business keys. There should be no duplicate entries that represent the same relationship or transaction.
	`LoadDate`	Same as in Hub definition.
	`RecordSource`	Same as in Hub definition.
Optional	`LastSeenDate`	Same as in Hub definition.
Optional	`DependentChildKey`	Is a reference that is not unique on its own, but rather it is unique only in combination of the main ID. This type is called *degenerate field* and affects the grain and uniqueness of the data set. When a `DependentChildKey` is defined, it has to be combined with all the `HubHash` keys to generate the `FixedLinkHashKey`.
FK	`HubHashKey` **	Foreign key to the connecting hub. There can be 2 or more connecting hubs.

Link characteristics.

- A link connects business keys.
- Capture the past, present and future relationships at the lowest possible granularity.
- Represent many-to-many relationships. A link can model 1:1, 1:M, M:1 and M:M relationships without change to the definition of the link table.
- Connect 2 or more hubs or the same hub multiple times.
- Links can be used to connect distributed data warehouses, where some parts of the model are stored in one location and other parts in other locations.
- The granularity of the links is defined by the number of hubs connected to the link. The more hubs connected to the link, the finer the granularity.

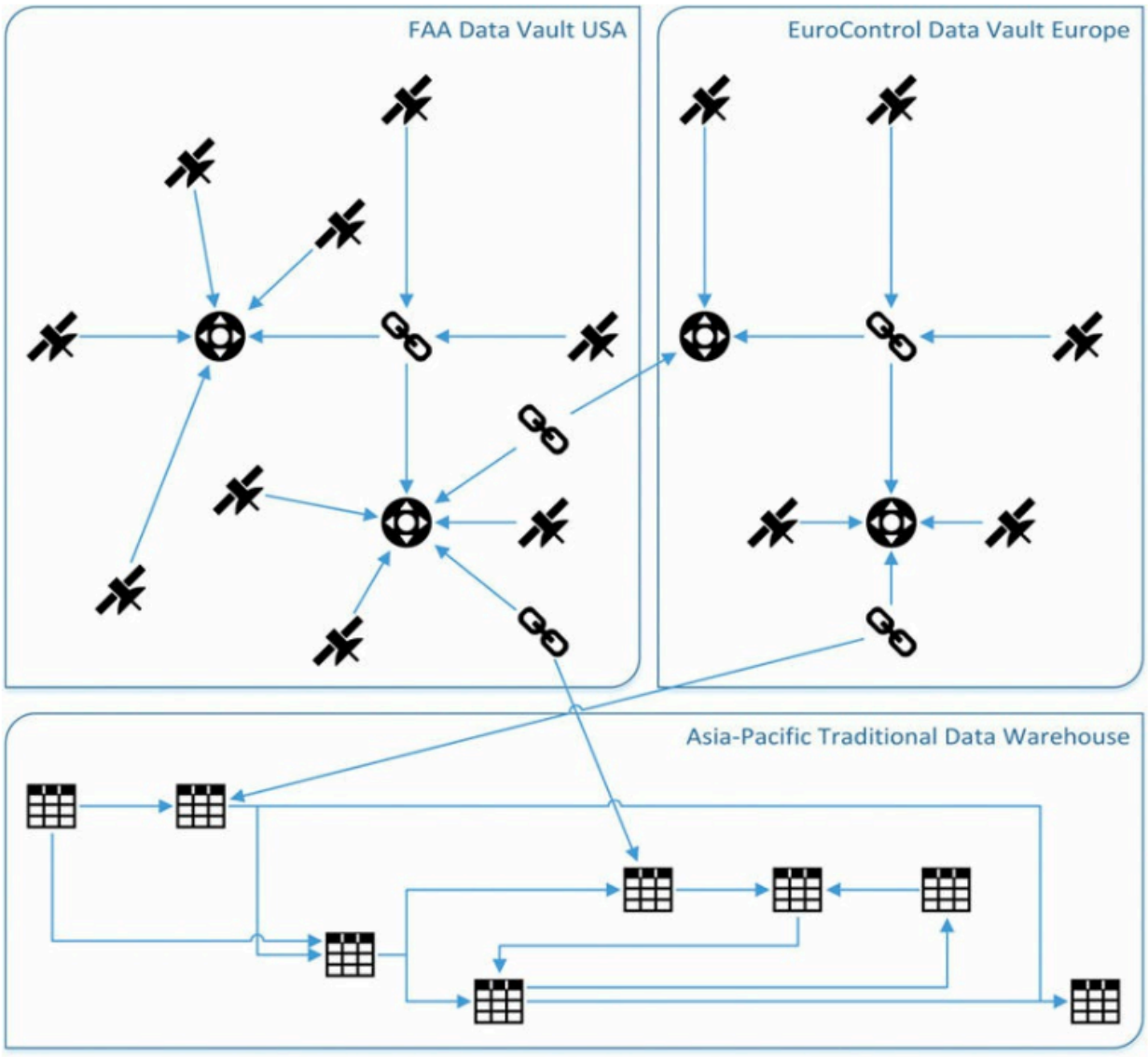

FIGURE 4.18 Distributed data warehouse connected by Data Vault links.

Degenerate field

Rules that apply to degenerate fields:

- Cannot stand on their own (like hubs).
- Have no business meaning.
- Dependent on another context in order to be valid.
- Give meaning and uniqueness to additional relationship information.
- Have no "descriptors" of their own.

Examples:

- The sides of a cassette tape (side A, side B).
- Page number in a book.
- Timestamp of an email.

Refactoring a Link (granularity change)

- If the business requirements changed and now we need either extra hubs or less hubs connected to my link, **we cannot edit the structure of the link**.
- The recommended procedure is "close" the link (meaning no more data ingested into it) and create a new link the same hubs connected to it and including the new changes.
- For querying purposes, we will need to `UNION` both links (legacy and new) and assign a `unknown` value to the missing values in the keys of added/removed hubs.

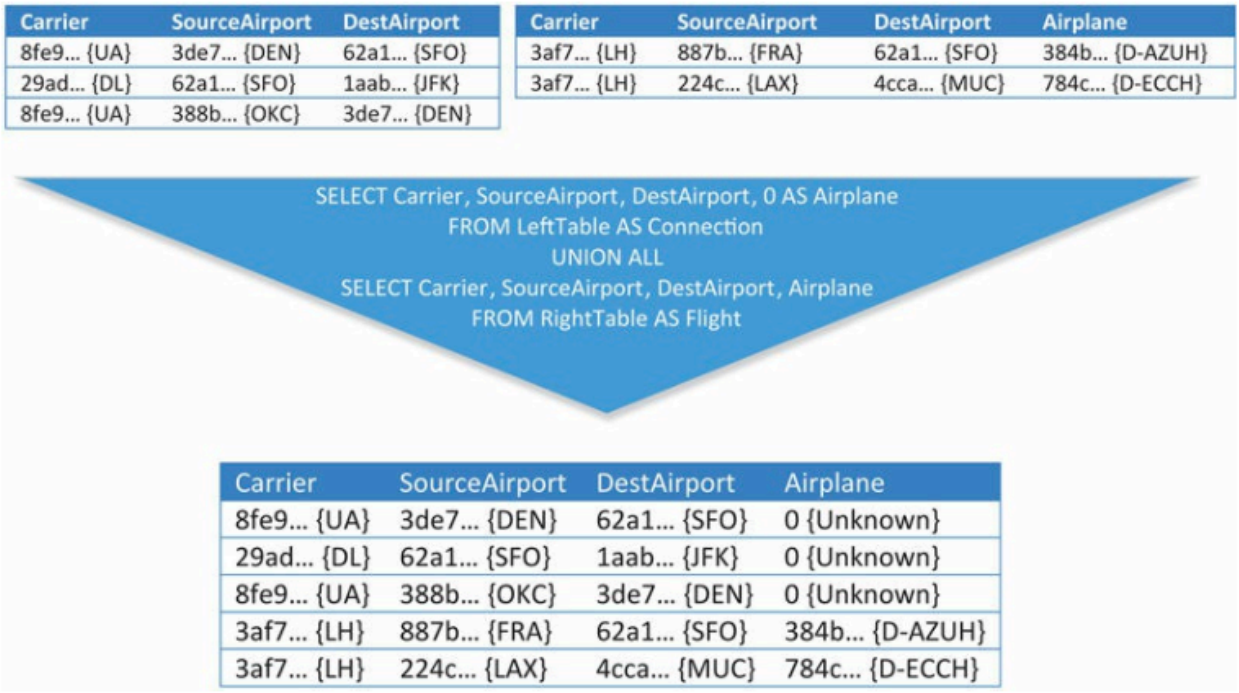

FIGURE 4.20 Merging historic data (upper left) with new data (upper right) from individual Data Vault links.