Data Warehouse Schema: Star Schema with SCD Type 2

SCD Type 2 Strategy: We will track historical changes by adding three specific columns to any dimension we want to track. When an attribute changes (e.g., a product's condition is updated, a user moves to a new city), we will **deactivate the old row and insert a new one** with the updated information.

The three SCD columns are:

- `is_active` (BOOLEAN): `TRUE` for the current, active record; `FALSE` for old, historical records.
- `valid_from_date` (DATE): The date this version of the record became active.
- `valid_to_date` (DATE): The date this version of the record expired. For the currently active record, this will be `NULL` or a far-future date like `9999-12-31`.

Dimension Tables

1. `DIM_PRODUCT`

This dimension stores the attributes of each unique physical product configuration. It is a prime candidate for SCD Type 2, as details might be manually corrected or updated over time.

Column Name	Data Type	Description
`product_key`	`INTEGER`	**Surrogate Key (PK).** Unique ID generated by the warehouse.
`product_natural_key`	`VARCHAR`	**Business Key.** A unique hash (e.g., MD5) of brand, model, storage, etc. Used to find the product across loads.
`brand`	`VARCHAR`	e.g., 'Apple', 'Samsung'. Standardized from all sources.
`model`	`VARCHAR`	e.g., 'iPhone 13', 'Galaxy S22 Ultra'. Standardized.
`storage_gb`	`INTEGER`	e.g., 128, 256. Standardized to an integer.

`ram_gb`	`INTEGER`	e.g., 6, 8. Sourced primarily from scrapers.
`color`	`VARCHAR`	e.g., 'Midnight Green', 'Black'. Standardized.
`base_condition`	`VARCHAR`	A standardized category: 'New', 'Like New', 'Used - Good', 'Used - Fair', 'For Parts'.
`first_seen_timestamp`	`TIMESTAMP`	The first time this product configuration was seen.
`last_seen_timestamp`	`TIMESTAMP`	The last time this product configuration appeared in any source.

---

2. `DIM_USER`

Stores details about users, primarily from the mobile app. User location or device could change, making it suitable for SCD Type 2.

Column Name	Data Type	Description	SCD Type 2 Support
`user_key`	`INTEGER`	**Surrogate Key (PK).**	N/A
`app_user_id`	`VARCHAR`	**Business Key.** The user's ID from the app's production DB.	N/A
`user_city`	`VARCHAR`	The user's last known city.	**Yes** (Tracks moves)

`user_country`	`VARCHAR`	The user's last known country.	**Yes** (Tracks moves)
`last_known_device`	`VARCHAR`	e.g., 'iOS', 'Android'.	**Yes** (Tracks changes)
`first_seen_timestamp`	`TIMESTAMP`	When the user first registered or was seen.	N/A
`last_seen_timestamp`	`TIMESTAMP`	The timestamp of the user's very last activity.	N/A
`is_active`	`BOOLEAN`	`TRUE` for the user's current profile.	**SCD Column**
`valid_from_date`	`DATE`	The date this user profile became active.	**SCD Column**
`valid_to_date`	`DATE`	The date this profile was superseded (e.g., they moved).	**SCD Column**

---

3. `DIM_SOURCE`

A simple lookup table to identify the origin of an event. This is typically a **Type 1 SCD** (we just overwrite changes) as these sources are stable.

Column Name	Data Type	Description	SCD Type 2 Support
`source_key`	`INTEGER`	**Surrogate Key (PK).**	N/A
`source_name`	`VARCHAR`	'Dubizzle Scraper', 'OpenSooq Scraper', 'Internal POS', 'Mobile App', 'AI Pricing Engine', 'Manual Upload'	No (Overwrite)
`source_category`	`VARCHAR`	'Web Scraper', 'Internal Transaction', 'User Behavior', 'AI Model', 'Manual'	No (Overwrite)

---

4. `DIM_LOCATION`

Stores geographical information, primarily from listings. SCD Type 2 is less common here but could be useful if city names are corrected or districts are reclassified.

Column Name	Data Type	Description	SCD Type 2 Support
`location_key`	`INTEGER`	**Surrogate Key (PK).**	N/A
`city`	`VARCHAR`	e.g., 'Cairo', 'Giza'. Standardized.	**Yes** (Tracks changes)
`district`	`VARCHAR`	e.g., 'Maadi', '6th of October'. Standardized.	**Yes** (Tracks changes)
`country`	`VARCHAR`	e.g., 'Egypt'. Standardized.	**Yes** (Tracks changes)
`is_active`	`BOOLEAN`	`TRUE` for the current version.	**SCD Column**
`valid_from_date`	`DATE`	The date this version became active.	**SCD Column**
`valid_to_date`	`DATE`	The date this version was superseded.	**SCD Column**

---

5. `DIM_DATE`

A standard, pre-populated calendar dimension. It does not change.

Column Name	Data Type	Description	SCD Type 2 Support
`date_key`	`INTEGER`	**Surrogate Key (PK).** e.g., `20251022`	No
`full_date`	`DATE`	e.g., `2025-10-22`	No
`day_of_week_name`	`VARCHAR`	e.g., 'Wednesday'	No
`month_name`	`VARCHAR`	e.g., 'October'	No
`quarter`	`INTEGER`	e.g., 4	No

year	INTEGER	e.g., 2025	No

Fact Table

FCT_EVENTS

This is a **transactional fact table** that records every single event from every source. It is designed to be **immutable**—we only ever add new rows, never update them. This table will grow very large.

Column Name	Data Type	Description
event_id	VARCHAR	**Business Key.** A unique ID for the event, often from the source system.
event_type	VARCHAR	**Crucial.** 'LISTING_CREATED', 'SALE_COMPLETED', 'TRADE_IN_QUOTED', 'PRODUCT_VIEW', 'USER_SEARCH'
event_timestamp	TIMESTAMP	The precise, timezone-aware timestamp of when the event occurred. **The primary column for sorting/clustering.**
date_key	INTEGER	**Foreign Key to** . Links to the day the event happened.
product_key	INTEGER	**Foreign Key to** . Links to the state of the product *at the time of the event*. Can be NULL.
user_key	INTEGER	**Foreign Key to** . Links to the state of the user *at the time of the event*. Can be NULL.
location_key	INTEGER	**Foreign Key to** . Links to the state of the location *at the time of the event*. Can be NULL.
source_key	INTEGER	**Foreign Key to** . Identifies the data origin.
price	DECIMAL	The price for a listing or sale. NULL for other events.
quoted_trade_in_price	DECIMAL	The price from the AI engine. NULL for other events.

`search_term`	`VARCHAR`	The text from a user search. `NULL` for other events.
`source_listing_id`	`VARCHAR`	The original ID from the scraper (e.g., "dub-xyz-987"). Useful for debugging and joining back to raw data.