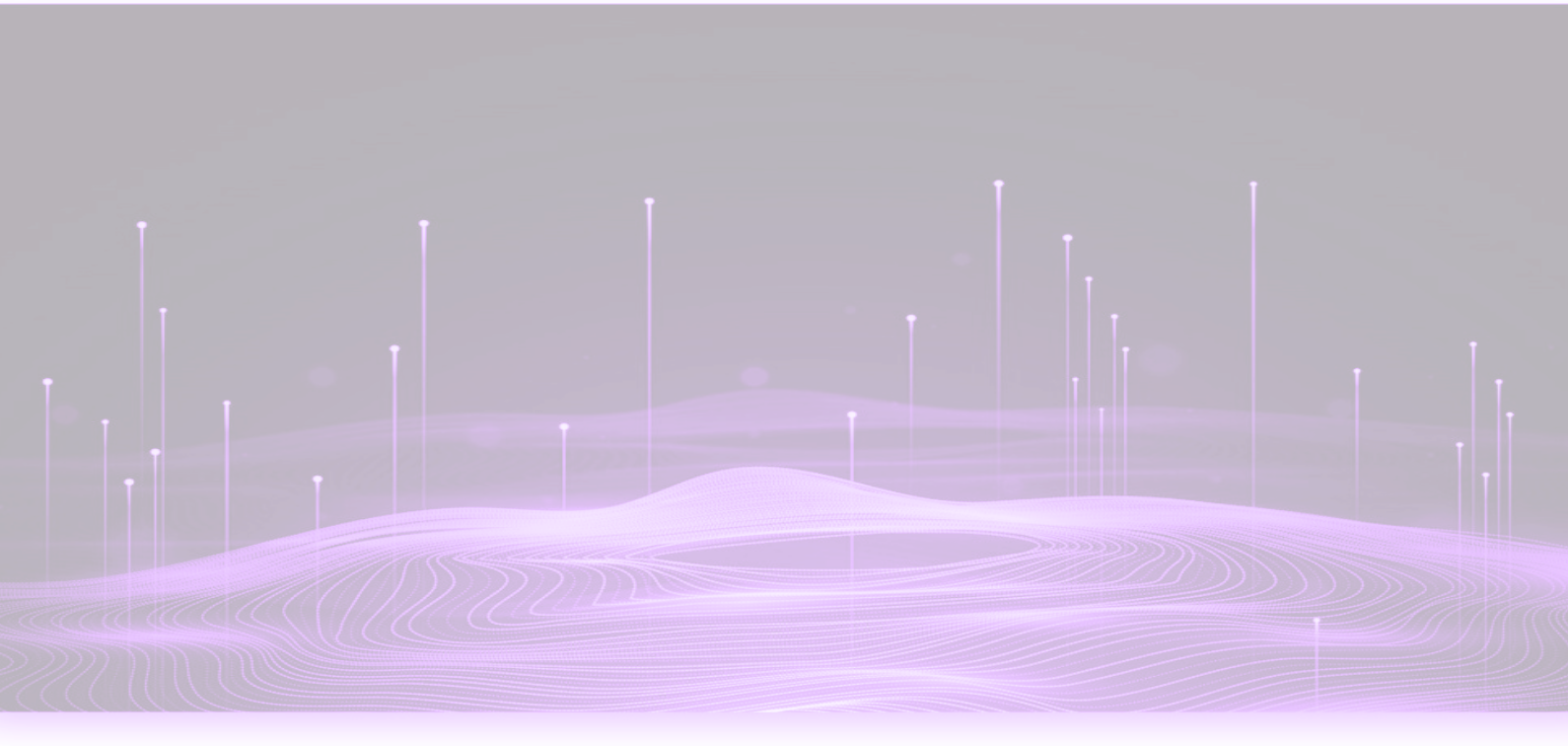
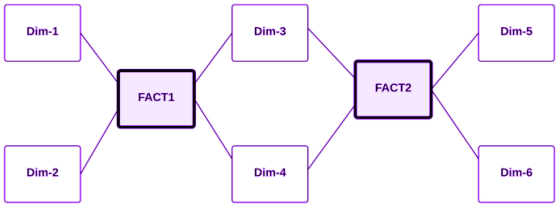
Enhancing Nubank's Data Warehouse:

multi-fact **Star Schema Design Proposal**

As Nubank introduces new products—such as life insurance, lending, and rewards—and expands into new countries, the data warehouse model must adapt to accommodate these changes. Many of these products are not related to peer-to-peer transactions and may only be available in specific regions. To support this growth, the model needs to be scalable, flexible, and easy to use for analysts, ensuring efficient data access and compliance with international regulations



What is a Multi-fact centralized star schema?



A Multi-fact Centralized Star Schema is a variation of the traditional star schema design where there are multiple fact tables, each representing different business processes (e.g., financial transactions, contracts, and rewards). These fact tables are centralized around a single set of dimension tables (e.g., customer, product, location, and time) that provide context for all the facts. This approach allows organizations to capture different aspects of their operations—such as financial transactions, product offerings, and customer rewards—while maintaining a consistent and unified view of the data.

Key Components of the Schema

- **Fact Tables**: Are the core of the star schema, each designed to capture specific business events (e.g., a transaction). They store measurable data and are connected to dimension tables through relational keys, enabling detailed analysis and reporting.
- **Dimension Tables**: Provide descriptive context for the data in fact tables. They store attributes that allow analysts to filter, group, or categorize the data. Dimensions are essential for adding meaning to the numerical data in fact tables.
- **Auxiliary Tables**: Complement the main fact and dimension tables by storing additional or specialized information that does not fit directly into the primary structures (facts or dimensions). These tables ensure data integrity and enable more granular and flexible analysis.

Benefits:

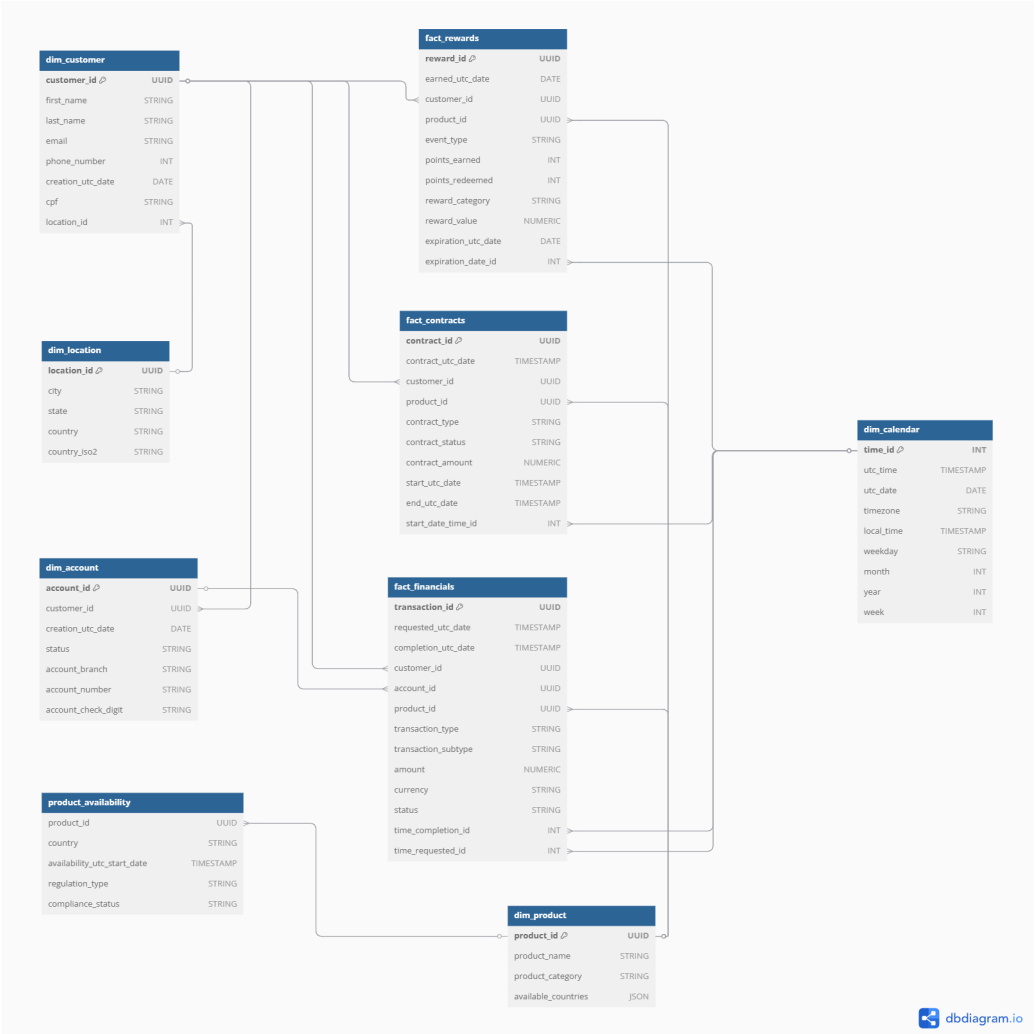
- **Simplified querying:** The denormalized structure of a star schema reduces the number of joins required to retrieve data. This makes it easier to create, understand, and update queries.
- **Faster performance:** The reduced join complexity and efficient indexing of fact and dimension tables enhances data retrieval.

- **Intuitive analysis:** Users can easily understand relationships and hierarchies among dimensions.
- **Easier business insights and reporting:** Star schemas simplify the process of pulling business reports.
- **Scalability:** Star schemas scale well when working with large volumes of data.

Overview of our proposal

This approach ensures seamless expansion into new products and regions. The multi-fact star schema allows for easy integration of new business processes by adding fact tables or expanding dimensions. As Nubank introduces products like life insurance, lending, and rewards, the model adapts without disrupting existing workflows.

New dimensions, such as product categories, customer behaviors, or region-specific data, can be added effortlessly. This adaptability ensures the model scales with Nubank's growth, avoiding complex restructuring while accommodating evolving data needs.



Fact tables

fact_financials: It stores financial transactions events made by customers, including details such as transaction type, amount, currency and status. This table is essential for analyzing customers' financial behavior and managing operations such as transfers and payments (PIX and not-PIX).

Field Name	Data Type	Description	Possible Values	Notes
`transaction_id`	UUID	Unique identifier for each transaction	-	Primary Key
`requested_utc_time`	TIMESTAMP	The timestamp of when the transaction was requested in UTC	-	-
`completion_utc_time`	TIMESTAMP	The timestamp of when the transaction was completed in UTC	-	-
`customer_id`	UUID	Reference to the customer table (`dim_customer`)	UUID of customer	Foreign Key to `dim_customer`
`account_id`	UUID	Reference to the account table (`dim_account`)	UUID of account	Foreign Key to `dim_account`
`product_id`	UUID	Reference to the product table (`dim_product`)	UUID of product	Foreign Key to `dim_product`
`transaction_type`	STRING	Type of the transaction	"PIX_IN", "PIX_OUT", "TRANSFER_IN", "TRANSFER_OUT"	-
`amount`	NUMERIC	Amount of the transaction	Positive decimal value	-
`currency`	STRING	Currency used in the transaction	"USD", "BRL", "EUR", etc.	-
`status`	STRING	Status of the transaction	"failed", "completed"	-
`time_completion_id`	INT	Reference to the time dimension (`dim_calendar`)	Time ID from `dim_calendar`	Foreign Key to `dim_calendar`
`time_requested_id`	INT	Reference to the time dimension (`dim_calendar`	Time ID from `dim_calendar`	Foreign Key to `dim_calendar`

fact_contracts: Stores information about contracts signed by customers, such as loans or insurance. It includes details such as contract type, status, amount, and start and end dates. This table is key to managing and monitoring customer financial agreements.

Field Name	Data Type	Description	Possible Values	Notes
`contract_id`	UUID	Unique identifier for each contract	-	Primary Key
`contract_utc_date`	TIMESTAMP	The timestamp of the contract creation in UTC	-	-
`customer_id`	UUID	Reference to the customer table (`dim_customer`)	UUID of customer	Foreign Key to `dim_customer`
`product_id`	UUID	Reference to the product table (`dim_product`)	UUID of product	Foreign Key to `dim_product`
`contract_type`	STRING	Type of the contract	"LOAN", "INSURANCE"	-
`contract_status`	STRING	Current status of the contract	"ACTIVE", "CLOSED", "PENDING"	-
`contract_amount`	NUMERIC	Amount of the contract	Positive decimal value	-
`start_utc_date`	TIMESTAMP	Start date of the contract	-	-
`end_utc_date`	TIMESTAMP	End date of the contract	-	-
`start_date_time_id`	INT	Reference to the start date time (`dim_calendar`)	Time ID from `dim_calendar`	Foreign Key to `dim_calendar`

fact_rewards: It records rewards earned or redeemed by customers, linking these events to specific products and the customers who performed them. This table is essential for managing and analyzing Nubank's rewards program, allowing detailed tracking of how customers interact with the rewards offered, such as cashback, travel, shopping, entertainment and more.

Field Name	Data Type	Description	Possible Values	Notes
`reward_id`	UUID	Unique identifier for each reward event	-	Primary Key
`earned_utc_date`	DATE	Date of when the reward was earned	-	-
`customer_id`	UUID	Reference to the customer table (`dim_customer`)	UUID of customer	Foreign Key to `dim_customer`
`product_id`	UUID	Reference to the product table (`dim_product`)	UUID of product	Foreign Key to `dim_product`
`event_type`	STRING	Type of the event	"POINTS_EARNED", "REWARD_REDEEMED"	-
`points_earned`	INT	Points earned during the event	Positive integer	-
`points_redeemed`	INT	Points redeemed during the event	Positive integer	-
`reward_category`	STRING	Category of the reward	"TRAVEL MILES", "CASHBACK"	-
`reward_value`	NUMERIC	Value of the reward in monetary terms	Positive decimal value	-
`expiration_utc_date`	DATE	Expiration date of the reward	-	-
`expiration_date_id`	INT	Reference to the expiration date time (`dim_calendar`)	Time ID from `dim_calendar`	Foreign Key to `dim_calendar`

Dimension tables

We can propose more dimensional tables related to our new products: rewards and contracts.

dim_customer: It contains detailed information about customers, such as name, email, phone number and CPF. This dimension provides the necessary context for analyzing customer behavior and preferences.

Field Name	Data Type	Description	Possible Values	Notes
`customer_id`	UUID	Unique identifier for each customer	-	Primary Key
`first_name`	STRING	Customer's first name	-	-
`last_name`	STRING	Customer's last name	-	-
`email`	STRING	Customer's email address	-	-
`phone_number`	INT	Customer's phone number	-	-
`creation_utc_date`	DATE	Date when the customer was created	-	-
`cpf`	STRING	Customer's CPF (Brazilian Taxpayer ID)	-	-
`location_id`	INT	Reference to the location table (`dim_location`)	UUID of location	Foreign Key to `dim_location`

dim_account: Stores account-specific data for each customer, such as account details, status, branch and account creation information. This dimension provides critical context for any financial data related to specific accounts in the system.

Field Name	Data Type	Description	Possible Values	Notes
`account_id`	UUID	Unique identifier for each account	-	Primary Key
`customer_id`	UUID	Reference to the customer table (`dim_customer`)	UUID of customer	Foreign Key to `dim_customer`
`creation_utc_date`	DATE	Date when the account was created	-	-
`status`	STRING	Status of the account	"ACTIVE", "CLOSED"	-
`account_branch`	STRING	Account branch code	-	-
`account_number`	STRING	Account number	-	-
`account_check_digit`	STRING	Check digit of the account number	-	-

dim_product: Stores data on products offered by Nubank, such as credit cards, loans, insurance and rewards. It includes information such as product name, category and countries where it is available. This dimension is essential for analyzing product performance.

Field Name	Data Type	Description	Possible Values	Notes
`product_id`	UUID	Unique identifier for each product	-	Primary Key
`product_name`	STRING	Name of the product	-	-
`product_category`	STRING	Category of the product	"insurance", "loan", "reward"	-
`available_countries`	JSON	List of countries where the product is available	-	-

dim_location: Records geographic information related to customers, such as city, state and country. This dimension allows you to analyze location-based data, such as regional distribution of customers or products.

Field Name	Data Type	Description	Possible Values	Notes
`id`	UUID	Unique identifier for each location	-	Primary Key
`city`	STRING	City of the location	Text	-
`state`	STRING	State of the location	Text	-
`country`	STRING	Country of the location	Text	-
`country_iso2`	STRING	ISO 2 code of the country	"BR", "US", "IN", etc.	-

dim_calendar: Provides a time frame for analyzing events and transactions. It includes details such as dates, time zones, days of the week and months. This dimension is essential for temporal analysis, such as monthly or yearly trends.

Field Name	Data Type	Description	Possible Values	Notes
`time_id`	INT	Unique identifier for each time entry	-	Primary Key
`utc_time`	TIMESTAMP	UTC time of the record	-	-
`utc_date`	DATE	UTC date of the record	-	-
`timezone`	STRING	Timezone of the location	"America/Sao_Paulo", etc.	-
`local_time`	TIMESTAMP	Local time converted from UTC	-	-
`weekday`	STRING	Weekday (e.g., "Monday", "Tuesday")	-	-
`month`	INT	Month (e.g., "1" for January)	-	-
`year`	INT	Year	-	-
`week`	INT	Week number in the year	-	-

Auxiliary tables

product_availability: It records product availability in different countries, along with regulatory and compliance information. This table is key to managing product expansion and ensuring legal compliance in each region.

Field Name	Data Type	Description	Possible Values	Notes
`product_id`	UUID	Reference to the product table (`dim_product`)	UUID of product	Foreign Key to `dim_product`
`country`	STRING	Country where the product is available	-	-
`availability_utc_start_date`	TIMESTAMP	Start date of product availability	-	-
`regulation_type`	STRING	Regulation type the product complies with	"GDPR", "LGPD"	-
`compliance_status`	STRING	Product's compliance status	"compliant", "non-compliant"	-

Good practices added in this strategy

1. **Denormalized model:** The star schema uses controlled denormalization in dimension tables (e.g., dim_location, dim_customer) to simplify queries and improve performance. By consolidating related attributes (e.g., city, state, country), it reduces the need for complex joins, allowing analysts to access frequently used data directly. Fact tables remain normalized, ensuring a balance between performance and data integrity.
2. **Good Nomenclature:** Using clear and consistent naming conventions for field names and data types across the schema helps in maintaining clarity and enhances the readability of the data model. Each field is named descriptively so it is immediately understandable to anyone who interacts with the data.
3. **Fact Grouping by Product Type:** Grouping fact tables by product type (e.g., financial transactions, rewards, and contracts) ensures that data related to different products is logically separated and easier to manage. This method enhances data granularity and allows for specialized analysis per product line
4. **Consolidation of Time Dimensions into a Single Calendar Table:** We moved from multiple time-related tables (e.g., d_month, d_year, d_week…) to a single dim_calendar

table. This simplifies queries by reducing the need for multiple joins, ensures consistent time analysis across fact tables, and improves scalability and maintainability by centralizing all time attributes into one table.

5. **Optimizing for Scalability:** The proposed model allows for easy scaling as new products, business processes, or data sources are added. New fact tables can be introduced without major changes to the existing schema, ensuring that the model evolves with the business. Whether it's adding a new product or introducing new regulatory requirements, the model is flexible enough to support future growth and new business use cases.
6. **Support for Regulatory Compliance:** The new model integrates regulatory compliance tracking, such as GDPR and LGPD compliance, via the product_availability table. This enables the business to easily manage and report on products' compliance status based on the regions they are offered, which is essential for meeting international data protection regulations.
7. **Local Time Support:** The dim_calendar table includes both utc_time and local_time fields, along with the timezone attribute. This approach centralizes time zone conversions, allowing analysts to seamlessly work with data in their local time zones without redundancy or manual calculations. It enhances usability, ensures consistency across time-based data, and supports global operations by standardizing time-related information.