Business Template

IOWA LIQUOR SALES

Legal Notice: This document contains privileged and/or confidential information and may not be disclosed, distributed or reproduced without the prior written permission of EPAM®.

CONTENTS

1 BUSINESS DESCRIPTION 3
1.1 Business background 3
1.2 Problems because of poor data management 3
1.3 Benefits from implementing a Data Warehouse 3
2 DIMENSIONS OF A BUSINESS 3
3 LOGICAL SCHEME 5
4 DATA FLOW 5
5 FACT TABLE PARTITIONING STRATEGY 8

1 BUSINESS DESCRIPTION

1.1 BUSINESS BACKGROUND

Welcome to Iowa-based spirits retailer, offering an extensive range of alcoholic beverages to customers throughout the state, serving two regions: Northwest and Iowalakes.

The business takes pride in providing a diverse selection of premium and craft spirits at competitive prices to cater to a wide range of consumer preferences. The friendly and knowledgeable staff is always available to assist customers with their queries and provide recommendations to ensure a unique and personalized shopping experience.

Iowa Liquor Sales places a high priority on responsible sales practices, ensuring compliance with all relevant regulations to maintain a safe and enjoyable environment for our customers. Its strong focus on customer satisfaction and ethical business practices has allowed us to establish a loyal customer base and play a significant role in driving the growth of the spirits industry in Iowa.

1.2 PROBLEMS BECAUSE OF POOR DATA MANAGEMENT

Iowa Liquor Sales has been struggling with poor data management practices. They currently store their data in various paper boxes and Excel files, making it difficult to access the information they need to make informed decisions. This has resulted in missed growth opportunities and potential financial losses due to inaccurate analysis and decision-making. Additionally, the lack of secure storage has put the company at risk of data breaches and unauthorized access to sensitive information. To remain competitive in the data-driven economy, Iowa Liquor Sales needs to improve their data management practices and invest in secure and efficient digital storage solutions.

1.3 BENEFITS FROM IMPLEMENTING A DATA WAREHOUSE

Implementing a data warehouse can offer a variety of benefits, including:

- Standardized and consolidated data from various sources, resulting in accurate analysis and informed decision-making;
- Improved resource utilization through easy access to organized data;
- Identification of patterns and trends, leading to the identification of new opportunities;
- Enhanced compliance management by providing a central repository for sensitive data with access controls and audit trails;
- Improved data quality through data cleansing and integration capabilities;
- Enhanced data security through access controls, encryption, and backup and recovery procedures.

2 DIMENSIONS OF A BUSINESS

Step 1: Select the Business Process.

The particular store in Iowa follows a liquor ordering process, which is the business process under consideration.

Step 2: Declare the Grain.

In this database model, the grain indicates the level of detail at which data is stored in the fact table. For the liquor ordering process, the grain is defined as an individual liquor product ordered by a store on a specific date. It represents the lowest level of granularity in the fact table.

Step 3: Identify the Dimensions.

Dimensions are the key descriptive attributes of the fact table that provide context to the measures. In this step, we can identify the following dimensions:

Dates dimension: contains details about the dates when orders were placed, such as year, quarter, month, and day, and also the fiscal information.

Junk dimension: contains information about the transaction status, transaction type, and tax type. It has 4 statuses for transactions: pending, completed, returned, and cancelled. Five types of transactions are available: cash, card, bank transfer, PayPal, and check. Tax options include applicable, not applicable, and 12% tax on MRP.

Shippers dimension: contains data about the shippers who deliver the liquor orders, such as shipper ID, name, status, phone, and rating. Ship base rate describes the minimum shipping charge, and ship rate describes the shipping charge per liter. It also has a specified shipper region. Region history is tracked over time in current_region and historical_region fields.

Products dimension: provides information about the individual liquor products, such as item number, item description, pack size, and bottle volume. Safety stock level describes the minimum inventory quantity, and reorder point describes the inventory level that triggers a purchase order.

Stores dimension: includes details about the stores where the liquor orders were placed, such as the store number, store name, location information, contact information, and payment preferences. The opt-in flag describes the marketing opt-in status, such as promotions and newsletters. Additionally, each store may be included in a membership program that offers exclusive deals and discounts to its members.

Vendors dimension: contains data about the vendors supplying the liquor ordered, such as vendor number, vendor name, contacts, status, rating, size, and website.

Employee dimension: includes information about the employees involved in the liquor ordering process, such as employee name, personal, contact information, and location.

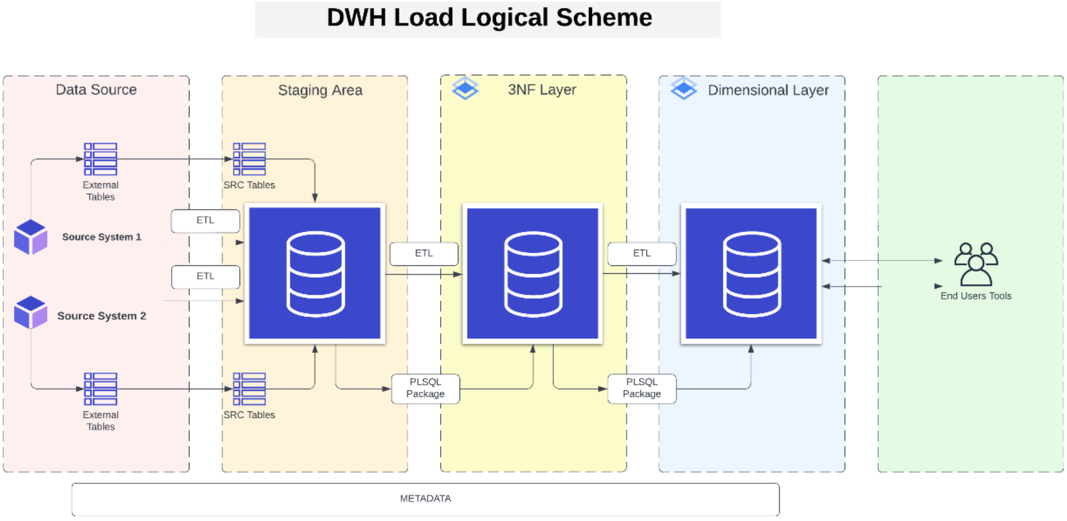
Step 4: Identify the Facts.

Facts are the numerical measurements that represent the business process. The facts are as follows:

- Quantity sold;
- Total amount;
- Volume sold in liters;
- Volume sold in gallons;
- State bottle cost;
- State bottle retail;

3 LOGICAL SCHEME

The logical scheme of the DWH load is represented down below.



DATA SOURCE

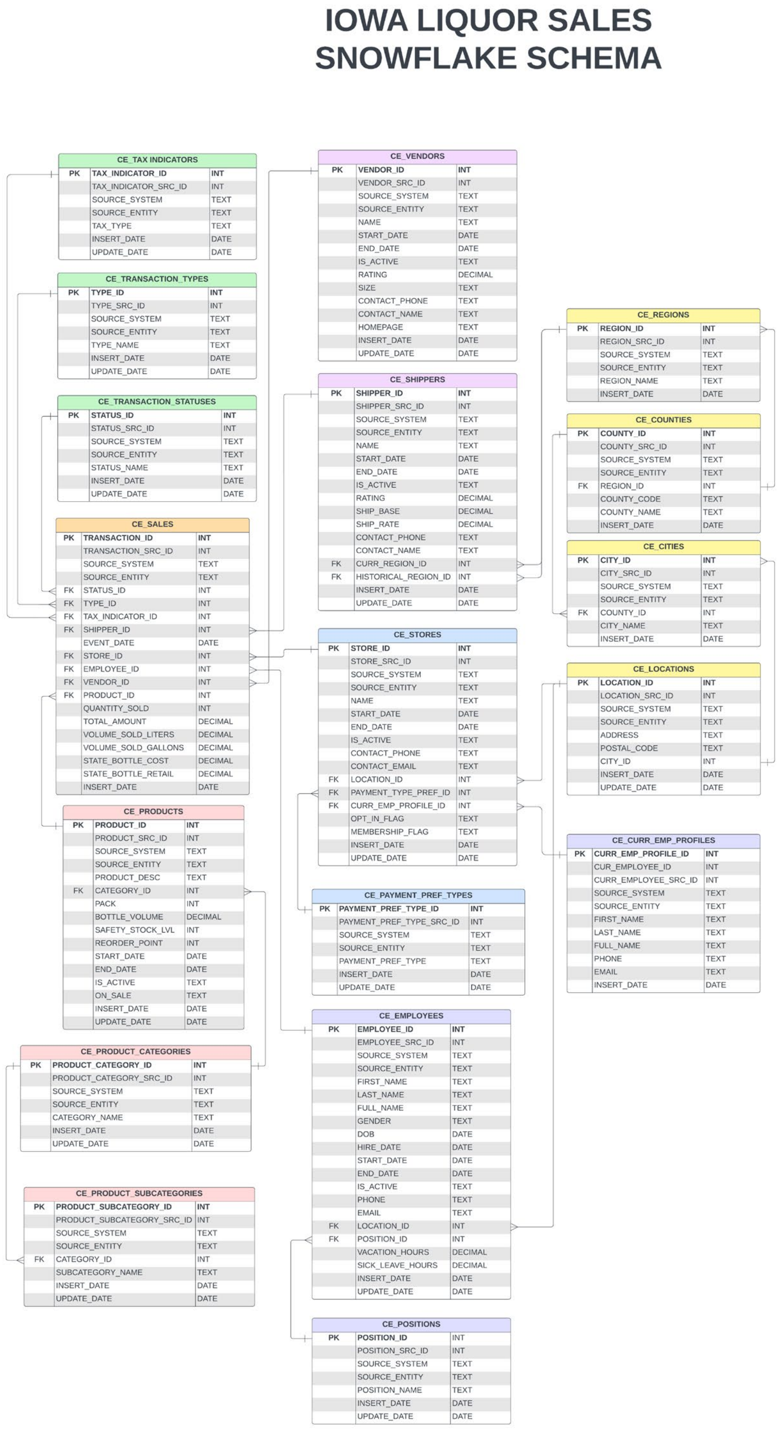
To begin the data warehouse load process, it is essential to recognize the origins of the data from where it will be extracted. These sources can vary from databases, files, applications, among other possibilities, in a typical business model. However, for the specific scenario outlined in this fictional project, the data was artificially generated and consolidated. The data was created using purchase records from Iowa State Class "E" liquor licenses, which were subsequently separated into two simulated sources: Northwest and Iowa Lakes regions.

STAGING AREA

The second stage of the data warehouse load process includes loading the data into a staging area. The staging area serves as an interim storage location where the data is stored temporarily before being loaded into the actual data warehouse. This method enables additional processing of the data, such as scrubbing and dealing with NULL values, to ensure the data's accuracy and quality before loading it into the data warehouse.

3 NF LAYER

The third step in the data warehouse loading process comprises loading the data into a 3NF layer. In this layer, the data is structured into normalized tables to eradicate redundancy and guarantee data consistency. This layer furnishes a reliable and organized view of the data, optimized for efficient querying. The 3NF layer plays a critical role in ensuring that the data is appropriately structured for analysis and reporting. Through the elimination of duplicate information and guaranteeing data consistency, the data can be easily analyzed and queried to obtain insights into customer behavior and business operations. Additionally, the 3NF relational layer's optimization for efficient querying minimizes the time and resources necessary for data analysis. *The snowflake schema of Liquor sales is represented on the left side:*



IOWA LIQUOR SALES SNOWFLAKE SCHEMA

	CE_TAX INDICATORS	
PK	**TAX_INDICATOR_ID**	**INT**
	TAX_INDICATOR_SRC_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
	TAX_TYPE	TEXT
	INSERT_DATE	DATE
	UPDATE_DATE	DATE

	CE_TRANSACTION_TYPES	
PK	**TYPE_ID**	**INT**
	TYPE_SRC_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
	TYPE_NAME	TEXT
	INSERT_DATE	DATE
	UPDATE_DATE	DATE

	CE_TRANSACTION_STATUSES	
PK	**STATUS_ID**	**INT**
	STATUS_SRC_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
	STATUS_NAME	TEXT
	INSERT_DATE	DATE
	UPDATE_DATE	DATE

	CE_SALES	
PK	**TRANSACTION_ID**	**INT**
	TRANSACTION_SRC_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
FK	STATUS_ID	INT
FK	TYPE_ID	INT
FK	TAX_INDICATOR_ID	INT
FK	SHIPPER_ID	INT
	EVENT_DATE	DATE
FK	STORE_ID	INT
FK	EMPLOYEE_ID	INT
FK	VENDOR_ID	INT
FK	PRODUCT_ID	INT
	QUANTITY_SOLD	INT
	TOTAL_AMOUNT	DECIMAL
	VOLUME_SOLD_LITERS	DECIMAL
	VOLUME_SOLD_GALLONS	DECIMAL
	STATE_BOTTLE_COST	DECIMAL
	STATE_BOTTLE_RETAIL	DECIMAL
	INSERT_DATE	DATE

	CE_PRODUCTS	
PK	**PRODUCT_ID**	**INT**
	PRODUCT_SRC_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
	PRODUCT_DESC	TEXT
FK	CATEGORY_ID	INT
	PACK	INT
	BOTTLE_VOLUME	DECIMAL
	SAFETY_STOCK_LVL	INT
	REORDER_POINT	INT
	START_DATE	DATE
	END_DATE	DATE
	IS_ACTIVE	TEXT
	ON_SALE	TEXT
	INSERT_DATE	DATE
	UPDATE_DATE	DATE

	CE_PRODUCT_CATEGORIES	
PK	**PRODUCT_CATEGORY_ID**	**INT**
	PRODUCT_CATEGORY_SRC_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
	CATEGORY_NAME	TEXT
	INSERT_DATE	DATE
	UPDATE_DATE	DATE

	CE_PRODUCT_SUBCATEGORIES	
PK	**PRODUCT_SUBCATEGORY_ID**	**INT**
	PRODUCT_SUBCATEGORY_SRC_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
FK	CATEGORY_ID	INT
	SUBCATEGORY_NAME	TEXT
	INSERT_DATE	DATE
	UPDATE_DATE	DATE

	CE_VENDORS	
PK	**VENDOR_ID**	**INT**
	VENDOR_SRC_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
	NAME	TEXT
	START_DATE	DATE
	END_DATE	DATE
	IS_ACTIVE	TEXT
	RATING	DECIMAL
	SIZE	TEXT
	CONTACT_PHONE	TEXT
	CONTACT_NAME	TEXT
	HOMEPAGE	TEXT
	INSERT_DATE	DATE
	UPDATE_DATE	DATE

	CE_SHIPPERS	
PK	**SHIPPER_ID**	**INT**
	SHIPPER_SRC_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
	NAME	TEXT
	START_DATE	DATE
	END_DATE	DATE
	IS_ACTIVE	TEXT
	RATING	DECIMAL
	SHIP_BASE	DECIMAL
	SHIP_RATE	DECIMAL
	CONTACT_PHONE	TEXT
	CONTACT_NAME	TEXT
FK	CURR_REGION_ID	INT
FK	HISTORICAL_REGION_ID	INT
	INSERT_DATE	DATE
	UPDATE_DATE	DATE

	CE_STORES	
PK	**STORE_ID**	**INT**
	STORE_SRC_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
	NAME	TEXT
	START_DATE	DATE
	END_DATE	DATE
	IS_ACTIVE	TEXT
	CONTACT_PHONE	TEXT
	CONTACT_EMAIL	TEXT
FK	LOCATION_ID	INT
FK	PAYMENT_TYPE_PREF_ID	INT
FK	CURR_EMP_PROFILE_ID	INT
	OPT_IN_FLAG	TEXT
	MEMBERSHIP_FLAG	TEXT
	INSERT_DATE	DATE
	UPDATE_DATE	DATE

	CE_PAYMENT_PREF_TYPES	
PK	**PAYMENT_PREF_TYPE_ID**	**INT**
	PAYMENT_PREF_TYPE_SRC_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
	PAYMENT_PREF_TYPE	TEXT
	INSERT_DATE	DATE
	UPDATE_DATE	DATE

	CE_EMPLOYEES	
PK	**EMPLOYEE_ID**	**INT**
	EMPLOYEE_SRC_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
	FIRST_NAME	TEXT
	LAST_NAME	TEXT
	FULL_NAME	TEXT
	GENDER	TEXT
	DOB	DATE
	HIRE_DATE	DATE
	START_DATE	DATE
	END_DATE	DATE
	IS_ACTIVE	TEXT
	PHONE	TEXT
	EMAIL	TEXT
FK	LOCATION_ID	INT
FK	POSITION_ID	INT
	VACATION_HOURS	DECIMAL
	SICK_LEAVE_HOURS	DECIMAL
	INSERT_DATE	DATE
	UPDATE_DATE	DATE

	CE_POSITIONS	
PK	**POSITION_ID**	INT
	POSITION_SRC_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
	POSITION_NAME	TEXT
	INSERT_DATE	DATE
	UPDATE_DATE	DATE

	CE_REGIONS	
PK	**REGION_ID**	**INT**
	REGION_SRC_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
	REGION_NAME	TEXT
	INSERT_DATE	DATE

	CE_COUNTIES	
PK	**COUNTY_ID**	**INT**
	COUNTY_SRC_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
FK	REGION_ID	INT
	COUNTY_CODE	TEXT
	COUNTY_NAME	TEXT
	INSERT_DATE	DATE

	CE_CITIES	
PK	**CITY_ID**	**INT**
	CITY_SRC_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
FK	COUNTY_ID	INT
	CITY_NAME	TEXT
	INSERT_DATE	DATE

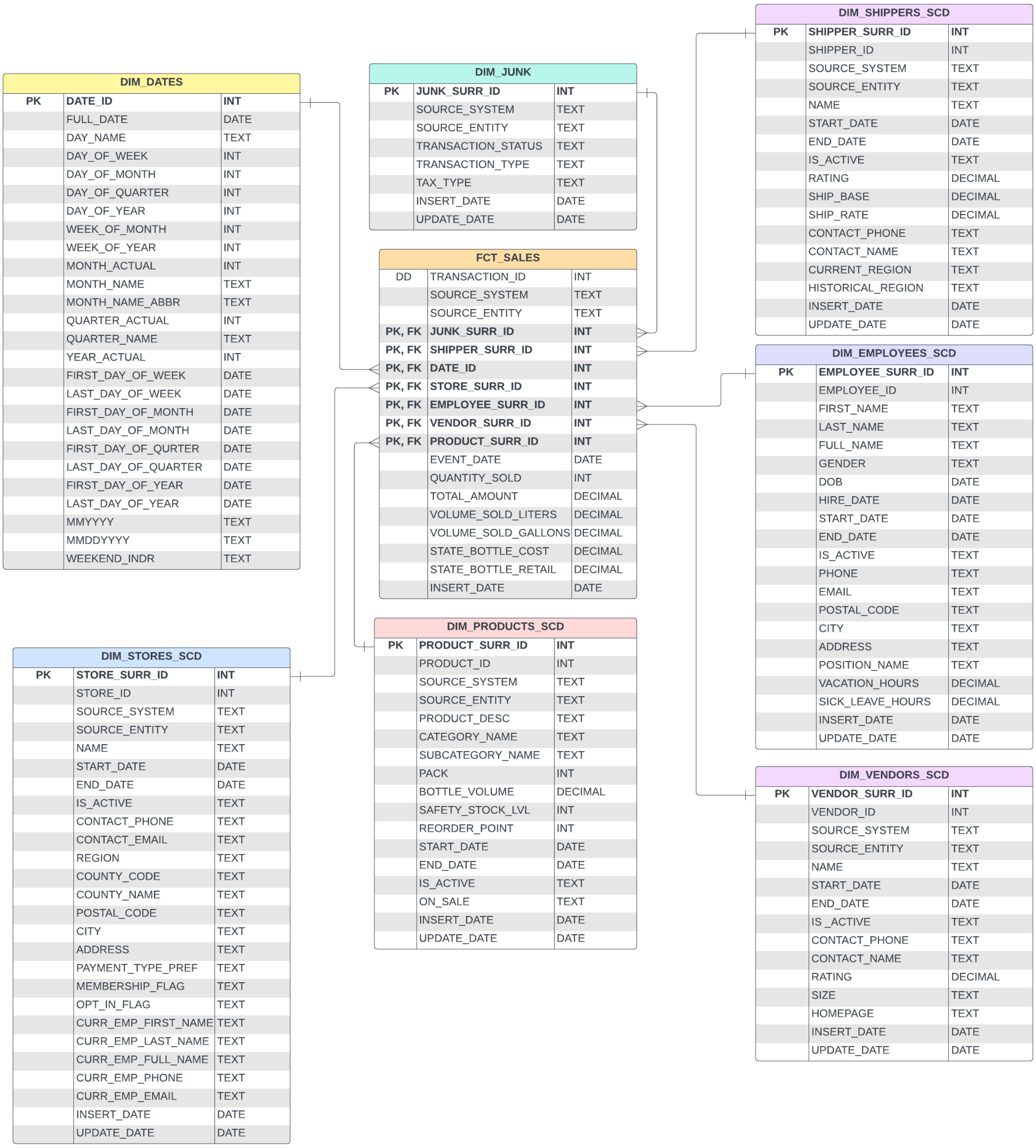
	CE_LOCATIONS	
PK	**LOCATION_ID**	**INT**
	LOCATION_SRC_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
	ADDRESS	TEXT
	POSTAL_CODE	TEXT
	CITY_ID	INT
	INSERT_DATE	DATE
	UPDATE_DATE	DATE

	CE_CURR_EMP_PROFILES	
PK	**CURR_EMP_PROFILE_ID**	**INT**
	CUR_EMPLOYEE_ID	INT
	CURR_EMPLOYEE_SRC_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
	FIRST_NAME	TEXT
	LAST_NAME	TEXT
	FULL_NAME	TEXT
	PHONE	TEXT
	EMAIL	TEXT
	INSERT_DATE	DATE

DIMENSIONAL LAYER

The ultimate stage is to load the data into the dimensional layer, which organizes data into a star schema configuration. Fact tables contain quantitative or measurement data, while dimension tables hold descriptive data. The dimensional layer is constructed to optimize analytical querying, allowing users to examine data along several dimensions and hierarchies to acquire insights and make informed choices. *The star schema of the Liquor sales:*

IOWA LIQUOR SALES
STAR SCHEMA



DIM_DATES		
PK	DATE_ID	INT
	FULL_DATE	DATE
	DAY_NAME	TEXT
	DAY_OF_WEEK	INT
	DAY_OF_MONTH	INT
	DAY_OF_QUARTER	INT
	DAY_OF_YEAR	INT
	WEEK_OF_MONTH	INT
	WEEK_OF_YEAR	INT
	MONTH_ACTUAL	INT
	MONTH_NAME	TEXT
	MONTH_NAME_ABBR	TEXT
	QUARTER_ACTUAL	INT
	QUARTER_NAME	TEXT
	YEAR_ACTUAL	INT
	FIRST_DAY_OF_WEEK	DATE
	LAST_DAY_OF_WEEK	DATE
	FIRST_DAY_OF_MONTH	DATE
	LAST_DAY_OF_MONTH	DATE
	FIRST_DAY_OF_QURTER	DATE
	LAST_DAY_OF_QUARTER	DATE
	FIRST_DAY_OF_YEAR	DATE
	LAST_DAY_OF_YEAR	DATE
	MMYYYY	TEXT
	MMDDYYYY	TEXT
	WEEKEND_INDR	TEXT

DIM_STORES_SCD		
PK	STORE_SURR_ID	INT
	STORE_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
	NAME	TEXT
	START_DATE	DATE
	END_DATE	DATE
	IS_ACTIVE	TEXT
	CONTACT_PHONE	TEXT
	CONTACT_EMAIL	TEXT
	REGION	TEXT
	COUNTY_CODE	TEXT
	COUNTY_NAME	TEXT
	POSTAL_CODE	TEXT
	CITY	TEXT
	ADDRESS	TEXT
	PAYMENT_TYPE_PREF	TEXT
	MEMBERSHIP_FLAG	TEXT
	OPT_IN_FLAG	TEXT
	CURR_EMP_FIRST_NAME	TEXT
	CURR_EMP_LAST_NAME	TEXT
	CURR_EMP_FULL_NAME	TEXT
	CURR_EMP_PHONE	TEXT
	CURR_EMP_EMAIL	TEXT
	INSERT_DATE	DATE
	UPDATE_DATE	DATE

DIM_JUNK		
PK	JUNK_SURR_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
	TRANSACTION_STATUS	TEXT
	TRANSACTION_TYPE	TEXT
	TAX_TYPE	TEXT
	INSERT_DATE	DATE
	UPDATE_DATE	DATE

FCT_SALES		
DD	TRANSACTION_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
PK, FK	JUNK_SURR_ID	INT
PK, FK	SHIPPER_SURR_ID	INT
PK, FK	DATE_ID	INT
PK, FK	STORE_SURR_ID	INT
PK, FK	EMPLOYEE_SURR_ID	INT
PK, FK	VENDOR_SURR_ID	INT
PK, FK	PRODUCT_SURR_ID	INT
	EVENT_DATE	DATE
	QUANTITY_SOLD	INT
	TOTAL_AMOUNT	DECIMAL
	VOLUME_SOLD_LITERS	DECIMAL
	VOLUME_SOLD_GALLONS	DECIMAL
	STATE_BOTTLE_COST	DECIMAL
	STATE_BOTTLE_RETAIL	DECIMAL
	INSERT_DATE	DATE

DIM_PRODUCTS_SCD		
PK	PRODUCT_SURR_ID	INT
	PRODUCT_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
	PRODUCT_DESC	TEXT
	CATEGORY_NAME	TEXT
	SUBCATEGORY_NAME	TEXT
	PACK	INT
	BOTTLE_VOLUME	DECIMAL
	SAFETY_STOCK_LVL	INT
	REORDER_POINT	INT
	START_DATE	DATE
	END_DATE	DATE
	IS_ACTIVE	TEXT
	ON_SALE	TEXT
	INSERT_DATE	DATE
	UPDATE_DATE	DATE

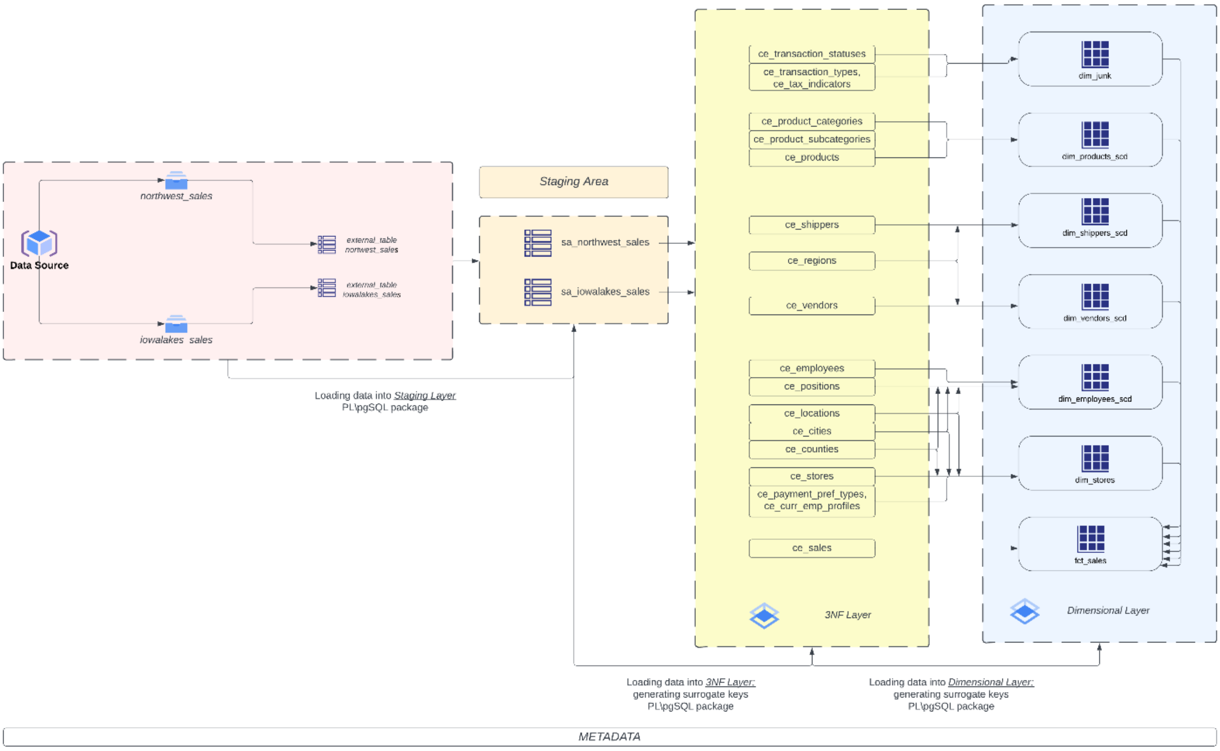
DIM_SHIPPERS_SCD		
PK	SHIPPER_SURR_ID	INT
	SHIPPER_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
	NAME	TEXT
	START_DATE	DATE
	END_DATE	DATE
	IS_ACTIVE	TEXT
	RATING	DECIMAL
	SHIP_BASE	DECIMAL
	SHIP_RATE	DECIMAL
	CONTACT_PHONE	TEXT
	CONTACT_NAME	TEXT
	CURRENT_REGION	TEXT
	HISTORICAL_REGION	TEXT
	INSERT_DATE	DATE
	UPDATE_DATE	DATE

DIM_EMPLOYEES_SCD		
PK	EMPLOYEE_SURR_ID	INT
	EMPLOYEE_ID	INT
	FIRST_NAME	TEXT
	LAST_NAME	TEXT
	FULL_NAME	TEXT
	GENDER	TEXT
	DOB	DATE
	HIRE_DATE	DATE
	START_DATE	DATE
	END_DATE	DATE
	IS_ACTIVE	TEXT
	PHONE	TEXT
	EMAIL	TEXT
	POSTAL_CODE	TEXT
	CITY	TEXT
	ADDRESS	TEXT
	POSITION_NAME	TEXT
	VACATION_HOURS	DECIMAL
	SICK_LEAVE_HOURS	DECIMAL
	INSERT_DATE	DATE
	UPDATE_DATE	DATE

DIM_VENDORS_SCD		
PK	VENDOR_SURR_ID	INT
	VENDOR_ID	INT
	SOURCE_SYSTEM	TEXT
	SOURCE_ENTITY	TEXT
	NAME	TEXT
	START_DATE	DATE
	END_DATE	DATE
	IS _ACTIVE	TEXT
	CONTACT_PHONE	TEXT
	CONTACT_NAME	TEXT
	RATING	DECIMAL
	SIZE	TEXT
	HOMEPAGE	TEXT
	INSERT_DATE	DATE
	UPDATE_DATE	DATE

4 DATA FLOW

The data flow of the DWH is represented down below.



To initiate the DWH load, the first step is to acquire the files from the data source and create foreign tables in the database, which represent external data sources. Foreign tables allow the manipulation and access of external data sources as if they were regular database tables. This ensures easy integration of the source files into the data warehouse.

The second step involves loading the tables to a staging area, while the third step entails loading the data into the 3NF layer. This step involves generating unique surrogate keys for each record in the data warehouse, ensuring that the data is clean and accurate and all tables are represented in the 3NF model.

Finally, the data is loaded into the dimensional layer, requiring the creation of fact and dimension tables and generating a different set of surrogate keys. The use of PL/pgSQL packages is integral to the data flow, encapsulating and organizing relevant procedures, functions, variables, and other constructs into a single schema object stored in the database, reusable across multiple applications and sessions.

Overall, the data flow of a DWH load comprises a series of steps that guarantee the data is consistent, accurate, complete, and can be effortlessly analyzed and reported by end-users.

5 FACT TABLE PARTITIONING STRATEGY

When it comes to partitioning a fact table, a number of factors can come into play. Factors like the size of the table, available hardware resources, data distribution, and query patterns can all impact the partitioning strategy. Depending on the situation, some factors may hold more weight than others.

For instance, if the queries that are expected to be performed involve a specific time range or dimension, partitioning the fact table based on those parameters may be the most optimal choice. However, if the data distribution is uneven, a more sophisticated partitioning strategy may be necessary to ensure efficient processing.

In our specific case, the dataset is not particularly large and doesn't require significant hardware resources. While query patterns and data distribution still need to be considered, the business requirements regarding analytics are somewhat unclear. As such, it's challenging to tailor the partitioning to specific aspects like region.

However, one key factor to consider is the data distribution. If the data is evenly distributed, a simple partitioning strategy based on time or another dimension may be sufficient. On the other hand, if the data is skewed or has uneven distribution, a more complex partitioning strategy may be necessary.

In our scenario, we will implement table partitioning using a range of dates (years) as the primary partition key, with further subpartitions based on months within each year. This method offers several benefits such as improved performance through parallel processing, enhanced data retrieval efficiency, and better organization. Furthermore, this strategy enables efficient incremental loading of data into the fact table, reducing the data volume that needs to be loaded during each update.