epam

RETAIL SALES OF MEDICINES

PHARMACY
HEALTH

CONTENTS

RETAIL SALES OF MEDICINES....1
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
1.4 Datasets description....4
1.5 Grain / four stage design....6
1.5.1 Business process....6
1.5.2 Grain....6
1.5.3 Dimensions....6
1.5.4 Facts....6
1.5.5 Fact table description....7
1.5.6 Dimensions descriptions....8
2 BUSINESS LAYER 3NF....16
2.1 Design process / scd....16
2.1.1 DIM_PRODUCTS normalization....16
2.1.2 DIM_CUSTOMER normalization....16
2.1.3 DIM_STORES normalization....16
2.1.4 DIM_EMPLOYEES normalization....17
2.1.5 DIM_PROMOTIONS normalization....17
2.1.6 DIM_PAYMENT_METHODS normalization....17
2.1.7 DIM_SALES_CHANNELS normalization....17
2.1.8 FCT_SALES_DD normalization....17
3 BUSINESS LAYER DIMENSIONAL MODEL....19
3.1 Denormalization....19
3.2 Dim_time_of_day....19
3.3 Other details....20
4 LOGICAL SCHEME....22
5 DATA FLOW....22
6 FACT TABLE PARTITIONING STRATEGY....25

1 BUSINESS DESCRIPTION

1.1 BUSINESS BACKGROUND

- ✓ Pharmaceuticals is a very responsible field, since usually buying medicines is rarely done for pleasure. People need practical advice when choosing a product in a pharmacy. In all regions and all ages, the need for therapeutic or preventive products arises from time to time.
- ✓ The approaches to success in this area are specific. Caring for people, analyzing their needs and problems should become the main philosophy in this business. Sales analysis in various aspects can significantly help maintain customer confidence, providing them with good medicines, valid promotions, quality consultation and service.

1.2 PROBLEMS BECAUSE OF POOR DATA MANAGEMENT

- ✓ Lack of accurate data analysis leads to erroneous conclusions about the effectiveness of certain strategies (promotions, suppliers, brands, chosen territory and premise area for a particular pharmacy and many others)
- ✓ Clear numbers, structure and calculations are a reliable basis for business ideas. What seems obvious through human observation may turn out to be wrong. Likewise, conclusions drawn from a short period may differ from information obtained through a long time
- ✓ In addition, without accurate information, it is sometimes impossible to find business weaknesses or unnecessary costs that could be avoided

1.3 BENEFITS FROM IMPLEMENTING A DATA WAREHOUSE

The data warehouse is designed specifically to solve the problems described above and can answer you a variety of useful questions:

- ➢ What are the best selling products?
- ➢ Which medications are better sold than their analogues within the same category?
- ➢ Which suppliers' prices allow you to get the most revenue?
- ➢ Which brands are more popular than others?
- ➢ Do customers prefer to order online or buy at pharmacies?
- ➢ Which promotions are useless and which ones are worth repeating and advertising?
- ➢ What is the pharmacies workload by territory?
- ➢ Which employees are better at working with customers than others?
- ➢ Is the area rented for a pharmacy justified?
- ➢ In what form is this or that medicine most in demand?

Further processing data would also let you:

- ◆ Control purchase volumes so as not to upset customers by selling medicines with a short shelf life and avoid losses due to expired products
- ◆ Save costs by renting premises with the required area for a given location
- ◆ Choose the most profitable suppliers and popular brands
- ◆ Control the amount of seasonal medications
- ◆ Effectively plan promotions and offer online clients only suitable advertising
- ◆ Manage employees appropriately to reduce queues and downtime
- ◆ Make informed decisions about business expansion, consolidation or restructuring
- ◆ Find places where the business can be improved

1.4 DATASETS DESCRIPTION

The datasets contain sales information from 2 sources, separated by sales channel:

- ✓ offline sales in the chain of 38 pharmacies spread across 5 states (USA)
- ✓ online sales through web-site or mobile application

The first dataset contains the following information about pharmacies sales in all regions

Sales Information:

Date: The date of the product sale.

Time: Time of the sale receipt registration accurate to seconds.

Cost: The cost of each particular unit.

Price: The actual selling price of the product.

Quantity Sold: The number of units sold.

Type of Payment: Cash or Card.

Product Information:

Form: Measurement units (form of sales).

Weight and Quantity: Weight and quantity of the medicine in the package.

Brand: The manufacturer of the medicine.

Class: The category of the medicine (Analgesics, Antacid...).

Subclass: Further classification of the medicine within the class (Antagonist, Blocker...).

Description: General information: purpose, action, side effects.

Employee Information:

Full name: First and last name of the employee.

Date of birth: Employee's age.

Phone number: Employee's contact.

Email: Email number of the employee.

Gender: Demographic information of the employee.

Role: Employee's position in the pharmacy.

Pharmacy Information:

Name: Pharmacy business card.

Address: Detailed location of the pharmacy with precision of the building.

Phone number: Contact phone number.

Email: Official email address of the pharmacy.

Registration date: Pharmacy opening date.

Floor space: Area occupied by the pharmacy.

Promotion:

Channel: Channel of promotion distribution(TV, Online...).

Category: Classification, by medicine's purpose ("VITAMINS AND SUPPLEMENTS")

Discount: The percentage of discount applied to the sales price.

Description: A short slogan for the promotion.

Supplier:

Name: Legal name of the medicine supplier's company.

Phone: Contact phone number.

Email: Official email address of the company.

The second dataset contains the following information about online sales of medicines:

Sales Information:

Date: The date of the online order.

Time: Time of the order registration accurate to seconds.

Cost: The cost of each particular product unit.

Price: The actual selling price of the product.

Quantity Sold: The number of units sold.

Product Information:

Form: Measurement units (form of sales).

Weight and Quantity: Weight and quantity of the medicine in the package.

Brand: The manufacturer of the medical product.

Class: The category of the medicine (Analgesics, Antacid...).

Subclass: Further classification of the product within the class (Antagonist, Blocker...).

Customer Information:

Full Name: First and last name of the customer.

Date of Birth: Customer’s age.

Phone Number: Customer’s contact specified during registration.

Email: Registration email address.

Gender: The gender for which the medication is intended (men or women).

Date of Registration: Date of registration as a user of the site.

Address: The geographical location of the customer.

Promotion:

Channel: Channel of promotion distribution(TV, Online...).

Category: Classification, by medicines purpose (“VITAMINS AND SUPPLEMENTS“)

Discount: The percentage of discount applied to the sales price.

Description: A short slogan for the action.

Supplier:

Name: Legal name of the medicine supplier's company.

Phone: Contact phone number.

Email: Official email address of the company.

The datasets provide extensive data on medicine sales by regions, brands, suppliers, time periods, payment methods, channels, categories, providing extensive opportunities for multifaceted analysis.

1.5 GRAIN / FOUR STAGE DESIGN

1.5.1 Business process

The business process is formulated as follows: products retail sales transactions, carried out in pharmacies and through an online store.

1.5.2 Grain

The grain is specified as follows:

“One row per individual product line item in a certain quantity in a sales transaction receipt”

Grain answers us the following questions:

“What product is sold in what store on which day and time under what promotional condition in which transaction to which customer by what employee through which channel and what payment type”

When formulating the next logic was applied:

- To successfully use the analytic data warehouse capabilities, each row in the fact table should be as detailed as possible and reflect a real transactions. Datasets research showed that each sales transaction (online and offline) is being recorded in receipt, each of which includes one or several products, and each product can be purchased in one or more units.
- It would be a mistake to combine all product items of one receipt in a single row, as it would violate granularity rule and make it impossible to extract data about each product sold. We can decide to provide some preliminary aggregated calculations later (like monthly statistic for instance), but the basic structure should be atomic.
- And also there is no point in further splitting several unit of the same product of one receipt into separate rows, since this will not bring any additional benefit other than counting the number of units sold, because it’s already counted in a column “quantity”.
- Thus, the granularity specified above is selected as the most optimal one.

1.5.3 Dimensions

Based on the GRAIN specified, next dimensions are determined:

1. *DIM_TIME_DAY (dates with precision of the day according to naming convention)*

2. *DIM_TIME_OF_DAY* (time of receipt registration accurate to second)

3. *DIM_PRODUCTS*

4. *DIM_SUPPLIERS*

5. *DIM_CUSTOMERS*

6. *DIM_STORES*

7. *DIM_EMPLOYEES*

8. *DIM_PROMOTIONS*

9. *DIM_PAYMENT_METHODS*

10. *DIM_SALES_CHANNELS*

1.5.4 Facts

The next 4 facts (business process performance measurements) were already found in both datasets:

1. UNIT_COST_DOLLAR_AMOUNT *: cost of each product unit (costs are variable, not static)

2. DISCOUNT_UNIT_DOLLAR_PRICE: price of the product unit after applying discount amount

3. SALES_QUANTITY: quantity of each product item sold

4. EXTENDED_SALES_DOLLAR_AMOUNT: SALES_QUANTITY multiplied by DISCOUNT_UNIT_PRICE

* We see in data sets, that products costs are not static and change quite often. We have no data about the procurement and it would be a separate business process. Thus, COST_DOLLAR_AMOUNT will be included in the fact table, because we can say that product unit cost is independent attribute.

Additionally four calculated facts will be included:

5. REGULAR_UNIT_DOLLAR_PRICE: price of the unit before promotion has been applied
6. EXTENDED_COST_DOLLAR_AMOUNT: COST_DOLLAR_AMOUNT multiplied by SALES_QUANTITY
7. EXTENDED_DISCOUNT_DOLLAR_AMOUNT: by which EXTENDED_SALES_DOLLAR_AMOUNT was reduced
8. PROFIT_DOLLAR_AMOUNT: EXTENDED_SALES_DOLLAR_AMOUNT – EXTENDED_COST_DOLLAR_AMOUNT

Reasons of including:

- Increase flexibility for further analysis
- Eliminate the possibility of users calculation errors and maintain consistency

1.5.5 Fact table description

Based on the GRAIN specified and chosen facts (measurements), the fact table looks as following.

Column name	Description	Data Type
EVENT_DT	FK: references to the DIM_TIME_DAY	DATE
TIME_OF_DAY_ID	FK: references to the DIM_TIME_OF_DAY	INT
PRODUCT_ID	FK: references to the DIM_PRODUCTS	INT
SUPPLIER_ID	FK: references to the DIM_SUPPLIERS	INT
CUSTOMER_ID	FK: references to the DIM_CUSTOMERS	INT
STORE_ID	FK: references to the DIM_STORES	INT
EMPLOYEE_ID	FK: references to the DIM_EMPLOYEES	INT
PROMO_ID	FK: references to the DIM_PROMOTIONS	INT
PAYMENT_METHOD_ID	FK: references to the DIM_PAYMENT_METHODS	INT
SALES_CHANNEL_ID	FK: references to the DIM_SALES_CHANNELS	INT
UNIT_COST_DOLLAR_AMOUNT	FACT: Cost of one unit of product	NUMERIC(8,2)
REGULAR_UNIT_DOLLAR_PRICE	FACT: Price of the unit before promotion has been applied	NUMERIC(8,2)
DISCOUNT_UNIT_DOLLAR_PRICE	FACT: Price of the product unit after promotion applied	NUMERIC(8,2)
SALES_QUANTITY	FACT: Quantity of product units sold	INT
EXTENDED_COST_DOLLAR_AMOUNT	FACT: UNIT_COST multiplied by QUANTITY	NUMERIC(8,2)
EXTENDED_DISCOUNT_DOLLAR_AMOUNT	FACT: Amount by which the whole sales amount was reduced	NUMERIC(8,2)
EXTENDED_SALES_DOLLAR_AMOUNT	FACT: Final sales amount for all units paid by customer	NUMERIC(8,2)
PROFIT_DOLLAR_AMOUNT	FACT: EXTENDED_SALES_DOLLAR_AMOUNT-EXTENDED_COST_DOLLAR_AMOUNT	NUMERIC(8,2)

Example with filled data

EVENT_DT	TIME_OF_DAY_ID	PRODUCT_ID	SUPPLIER_ID	CUSTOMER_ID	STORE_ID	EMPLOYEE_ID	PROMO_ID	PAYMENT_METHOD_ID	SALES_CHANNEL_ID	UNIT_COST_DOLLAR_AMOUNT
2022-01-01	13:00:20	567	89	4	13	66	12	1	1	12.30

REGULAR_UNIT_DOLLAR_PRICE	DISCOUNT_UNIT_DOLLAR_PRICE	SALES_QUANTITY	EXTENDED_COST_DOLLAR_AMOUNT	EXTENDED_DISCOUNT_DOLLAR_AMOUNT	EXTENDED_SALES_DOLLAR_AMOUNT	PROFIT_DOLLAR_AMOUNT
15.70	14.13	3	36.90	4.71	42.39	5.49

1.5.6 Dimensions descriptions

1. DIM_TIME_DAY* *(*dates with precision of the day according to naming convention)*

Important dimension for analytical tasks. Will be generated with different hierarchies to provide wide range of possible analysis by various periods (days, weeks, months, quarters, years and many others).

Column name	Description	Data Type
TIME_DAY_ID	Meaningful INT: combination of date'd digits (20220101)	INT
TIME_DAY_DT	Date itself (YYYY-MM-DD), PK	DATE
DAY_NAME	Name of the day (Monday, Tuesday...)	VARCHAR(9)
DAY_NUMBER_IN_WEEK	Number of week the day belongs to (1-7)	INT
DAY_NUMBER_IN_MONTH	Number of day in the month the day belongs to (1-31)	INT
DAY_NUMBER_IN_YEAR	Number of day in the month the day belongs to (1-365/366)	INT
WEEK_NUMBER_IN_YEAR	Week number to which the date belongs, in year (1-52)	INT
YEAR_OF_WEEK	The same week can span across 2 years, is used to distinguish	INT
WEEK_ENDING_DT	End of the week the date belongs to	DATE
MONTH_NUMBER	Number of month the data belongs to, in the year (1-12)	INT
MONTH_NAME	Month name the date belongs to	VARCHAR(9)
DAYS_IN_MONTH	Number of days in the month the date belongs to (29, 30, 31)	INT
MONTH_ENDING_DT	Ending date of the month the date belongs to	DATE
YEAR_MONTH_DESCR	Combined year-month the date belongs to ("2022-12")	VARCHAR(7)
QUARTER_NUMBER	Number of quarter (1-4) the date belongs to	INT
QUARTER_ENDING_DT	The ending date of the quarter	DATE
QUARTER_DESCR	Combined year-quarter description the date belongs to("2000-01")	VARCHAR(7)
YEAR_NUMBER	The year the date belongs to	INT
DAYS_IN_YEAR	Number of days in the year the date belongs to	INT
YEAR_ENDING_DT	The ending date of the year	DATE

Example with filled data

TIME_DAY_ID	TIME_DAY_DT	DAY_NAME	DAY_NUMBER_IN_WEEK	DAY_NUMBER_IN_MONTH	DAY_NUMBER_IN_YEAR	WEEK_NUMBER_IN_YEAR	YEAR_OF_WEEK	WEEK_ENDING_DT	MONTH_NUMBER
20220101	2022-01-01	Friday	5	1	1	52	2021	2022-01-03	1

MONTH_NAME	DAYS_IN_MONTH	MONTH_ENDING_DT	YEAR_MONTH_DESCR	QUARTER_NUMBER	QUARTER_ENDING_DT	QUARTER_DESCR	YEAR_NUMBER	DAYS_IN_YEAR	YEAR_ENDING_DT
January	31	2022-01-31	2022-01	1	2022-03-31	2022-01	2022	365	2022-12-31

2. DIM_SUPPLIERS

Suppliers are included as independent dimension, because, as mentioned above, each supplier delivers a lots of medicines, and the same product can be supplied by different suppliers, but at different costs. Thus, fact table contains SUPPLIER_ID and cost fact s well.

Column name	Description	Data Type
SUPPLIER_ID	Unique identifier of the supplier	INT
SUPPLIER_NAME	Name of the supplier company which supplied the product	VARCHAR(70)
SUPPLIER_PHONE_NUM	Contact phone number of supplier company	VARCHAR(20)
SUPPLIER_EMAIL	Contact email address of supplier company	VARCHAR(255)

Example with filled data

SUPPLIER_ID	SUPPLIER_NAME	SUPPLIER_PHONE	SUPPLIER_EMAIL
2582	HealthPro	693059-8976	sales_department@pharmadirect.com

3. DIM_PRODUCTS

One of the most important aspect of business process - product (particular medication or medical product), include 2 independent hierarchies:

1. PRODUCT → PRODUCT_SUBCATEGORY → PRODUCT_CATEGORY*

* Each medicine is categorized in medical subcategories and subcategories (moreover: different categories can have subcategories with the same name (e.g. category “Antacid” has sub category “Inhibitor”, and category “Proton pump inhibitor” has it’s own sub category “Inhibitor”).

2. PRODUCT → BRAND**

** Each brand produces a lot of medicines, but a specific product is represented by only one brand (that is, if there are analogues, they are considered as different products, since they are called differently, have different forms, descriptions, and so on).

Column name	Description	Data Type
PRODUCT_ID	Surrogate PK	INT
PRODUCT_NAME	Textual name of the medicine	VARCHAR(70)
PRODUCT_FORM	The form in which the medicine is sold (tablet, solution...)	VARCHAR(30)
UNIT_MASS_MEASUREMENT	Mg, ml	INT
UNIT_MASS	Weight of one unit of PRODUCT_FORM	NUMERIC(7, 2)
UNITS_PER_PACKAGE	Amount of a PRODUCT_FORM in one package	INT
PROD_SUBCATEGORY_ID	Unique identifier of the category	INT
PROD_SUBCATEGORY_NAME	Name of the sub category the product belongs to	VARCHAR(70)
PROD_SUBCATEGORY_DESCR	General description of the medicine’s subcategory	VARCHAR(250)
PROD_CATEGORY_ID	Unique identifier of the category the product belongs to	INT
PROD_CATEGORY_NAME	Name of the category the product belongs to	VARCHAR(70)
PROD_CATEGORY_DESCR	General description of the medicine’s category	VARCHAR(250)

BRAND_ID	Unique identifier of the brand, which produced the product	INT
BRAND_NAME	Name of the manufacturer which produced the product	VARCHAR(250)

Example with filled data

PRODUCT_ID	PRODUCT_NAME	PRODUCT_FORM	UNIT_MASS_MEASUREMENT	UNIT_MASS	UNITS_PER_PACKAGE
697	DOMPERIDONE	TABLET	MG	50	60

PROD_SUBCATEGORY_ID	PROD_SUBCATEGORY_NAME	PROD_SUBCATEGORY_DESCR	PROD_CATEGORY_ID	PROD_CATEGORY_NAME
890	Laxative	Fast-acting relief from constipation	38	Gastrointestinal stimulant

PROD_CATEGORY_DESCR	BRAND_ID	BRAND_NAME
Stimulates gastrointestinal activity, including motility and secretion	8	AlphaRx Pharmaceuticals

4. DIM_CUSTOMERS

People, who purchase products (offline pharmacies, online store). For non-identified customers (in offline pharmacies for instance) – default customer record with id -1 is created.

Column name	Description	Data Type
CUSTOMER_ID	Surrogate PK	INT
CUSTOMER_FIRST_NAME	First name of the customer	VARCHAR(50)
CUSTOMER_LAST_NAME	Last name of the customer	VARCHAR(60)
CUSTOMER_PHONE_NUMB	Contact phone number	VARCHAR(20)
CUSTOMER_EMAIL	Email address specified during registration on the web-site	VARCHAR(255)
CUSTOMER_GENDER	One of 2 values (male, female)	VARCHAR(6)
CUSTOMER_BIRTH_DT	Date of birth specified during registration on the web-site	DATE
ACCOUNT_REG_DT	Registration date on the web-site	DATE
CUSTOMER_ADDRESS_ID	Address unique identifier	INT
CUSTOMER_ADDRESS_DESCR	Street name (address in other words)	VARCHAR(50)
CUSTOMER_CITY_ID	Unique identifier of the city	INT
CUSTOMER_CITY_NAME	Name of the city	VARCHAR(40)
CUSTOMER_ZIP_CODE	Number that represents the postal code of the address	VARCHAR(10)
CUSTOMER_STATE_ID	Unique identifier of the sate (USA)	INT
CUSTOMER_STATE_NAME	Name of the state	VARCHAR(15)

Example with filled data

CUSTOMER_ID	CUSTOMER_FIRST_NAME	CUSTOMER_LAST_NAME	CUSTOMER_PHONE_NUM	CUSTOMER_EMAIL	CUSTOMER_GENDER
-1	N/A	N/A	N/A	N/A	N/A
1	Carter	Wilson	(458) 799-5809	deandorsey@gmail.org	male

CUSTOMER_BIRTH_DT	ACCOUNT_REG_DT	CUSTOMER_ADDRESS_ID	CUSTOMER_ADDRESS_DESCR	CUSTOMER_CITY_ID	CUSTOMER_CITY_NAME
N/A	N/A	N/A	N/A	N/A	N/A
1987-01-14	2019-05-10	7009	Ocean front walk	802	San diego

CUSTOMER_POSTAL_CODE	CUSTOMER_STATE_ID	CUSTOMER_STATE_NAME
N/A	N/A	N/A
92109	2	California

5. DIM_STORES

Chain of all existing pharmacies across all states (USA), and online-store. Online store is default store with id -1

Column name	Description	Data Type
STORE_ID	Surrogate PK	INT
STORE_NAME	Name of the store	VARCHAR(60)
STORE_ADDRESS_ID	Address unique identifier	INT
STORE_ADDRESS_DESCR	Street name (address in other words)	VARCHAR(50)
STORE_CITY_ID	Unique identifier of the city	INT
STORE_CITY_NAME	Name of the city	VARCHAR(40)
STORE_ZIP_CODE	Number that represents the postal code of the address	VARCHAR(10)
STORE_STATE_ID	Unique identifier of the sate (USA)	INT
STORE_STATE	Name of the state	VARCHAR(15)
STORE_BUILD_NUM	Building number where the customer lives	VARCHAR(10)
STORE_PHONE_NUM	Contact phone number of the pharmacy	VARCHAR(20)
STORE_EMAIL	Official email address of the pharmacy	VARCHAR(255)
OPENING_DT	The date when the pharmacy has started working	DATE
FLOOR_SPACE	Total floor area occupied by the pharmacy, square meters	NUMERIC(8, 2)

Example with filled data

STORE_ID	STORE_NAME	STORE_ADDRESS_ID	STORE_ADDRESS_DESCR	STORE_CITY_ID
-1	N/A	N/A	N/A	N/A
32	AceRelief	133	Rabey farm road	32

STORE_CITY_NAME	STORE_ZIP_CODE	STORE_STATE_ID	STORE_STATE_NAME	STORE_BUILD_NUM
N/A	N/A	N/A	N/A	N/A
Suffolk	23435	3	Virginia	75

STORE_PHONE_NUM	STORE_EMAIL	OPENING_DT	FLOOR_SPACE
(456) 478-0126	officialwebsite@gmail.com	2019-12-10	N/A
(456) 799-0123	acerelief@gmail.com	2018-05-10	52.25

6. DIM_EMPLOYEES

Employees who process transactions. Theoretically employees can move to work from one place to another one, or replace someone. We have no historical information about employee's work places, and it would be extra for this business process. Current pharmacy and employee who's made a transaction there can be retrieved directly from the fact table. For online orders the default employee is created with id -1

Column name	Description	Data Type
EMPLOYEE_ID	Surrogate PK	INT
EMPL_FIRST_NAME	First name of the employee	VARCHAR(50)
EMPL_LAST_NAME	First name of the employee	VARCHAR(60)
EMPL_BIRTH_DT	Employee's date of birth	DATE
EMPL_PHONE_NUM	Contact phone number of the employee	VARCHAR(20)
EMPL_GENDER	Gender (male or female)	VARCHAR(6)
EMPL_EMAIL	Contact email address of the employee	VARCHAR(255)
EMPL_POSITION	Role of the employee in the company	VARCHAR(50)

Example with filled data

EMPLOYEE_ID	EMPL_FIRST_NAME	EMPL_LAST_NAME	EMPL_BIRTH_DT
-1	N/A	N/A	N/A
1	Joseph	Anderson	1992-09-24

EMPL_PHONE_NUM	EMPL_GENDER	EMPL_EMAIL	EMPL_POSITION
N/A	N/A	N/A	N/A
(789) 012-3456	male	ephepokfaon@gmail.com	seller

7. DIM_PROMOTIONS

Promotion, applied for particular sale transaction, has 2 independent hierarchies:

1) PROMOTION → SUB_CATEGORY; 2) PROMOTION → CHANNEL

Column name	Description	Data Type
PROMO_ID	Surrogate PK	INT
PROMO_NAME	Description of the promotion	VARCHAR(100)
PROMO_DISCOUNT%	Number of discount (in percents)	INT
PROMO_CHANNEL_ID	Unique identifier of the promotion distribution channel	INT
PROMO_CHANNEL_NAME	Promotion distribution channel name (radio, TV...)	VARCHAR(50)
PROMO_CATEGORY_ID	Unique identifier of the promotion category	INT
PROMO_CATEGORY_NAME	Description of the promotion category	VARCHAR(50)

Example with filled data

PROMO_ID	PROMO_NAME	PROMO_DISCOUNT%
1	NONE DISCOUNT	0
56	POWERHOUSE PAIN MANAGEMENT DISCOUNTS	8

PROMO_CHANNEL_ID	PROMO_ CHANNEL_NAME	PROMO_CATEGORY_ID	PROMO_CATEGORY_NAME
1	NONE DISCOUNT	1	NONE DISCOUNT
5	RADIO	7	PAIN MANAGEMENT SOLUTIONS

8. DIM_PAYMENT_METHODS

The smallest dimension with two possible payment type for now: cash and card

Column name	Description	Data Type
PAYMENT_METHOD_ID	Surrogate PK	INT
PAYMENT_ METHOD_NAME	Description of the payment type (cash, card)	VARCHAR(4)

Example with filled data

PAYMENT_METHOD_ID	PAYMENT_METHOD_NAME
1	CASH
2	CARD

9. DIM_SALES_CHANNELS

Channel of each particular sale transaction (online through web site/mobile application or offline sales in pharmacies). Are named as SALES_CHANNELS, to avoid confusing with PROMO_CHANNELS (different channels).

Column name	Description	Data Type
SALES_CHANNEL_ID	Surrogate PK	INT
SALES_CHANNEL_NAME	Description (or name) of channel	VARCHAR(7)

Example with filled data

SALES_CHANNEL_ID	SALES_CHANNEL_NAME
1	ONLINE
2	OFFLINE

2 BUSINESS LAYER 3NF

2.1 DESIGN PROCESS / SCD

Source systems:

- ✓ SA_OFFLINE_SALES (sales data from the chain of pharmacies – offline channel type)
- ✓ SA_ONLINE_SALES (sales data through the online channel type – web site, mobile application)

Source entities:

- ✓ SRC_pharm_offline_sales
- ✓ SRC_pharm_online_sales

The GRAIN is already specified, entities and hierarchies from the source file analyzed and grouped into dimensions. It's a good basis for the further normalization. We will go step-by-step through dimensions and fact table, extract independent entities, compose final 3NF schema, establishing relationships between normalized entities, adding source triplets and technical attributes.

Why CE_PRODUCTS_SCD is selected as slowly changing dimension:

Product is the main attribute for successful business and should be analyzed carefully. The name of the product, measurement of unit mass ("mg", "ml" etc.), product form ("tablet", "capsule" etc.) or amount of units per package can change over time and it's necessary to store such history to analyze how some changes in product affect sales.

2.1.1 DIM_PRODUCTS normalization

- ◆ *CE_PRODUCTS:*

  Remain only attributes fully functionally and direct depend on the PRODUCT_ID. Split product name into attributes name, mass measurement, unit mass, units per package

- ◆ *CE_PROD_CATEGORIES*
- ◆ *CE_PROD_SUBCATEGORIES*
- ◆ *CE_SUPPLIERS*

  each supplier can deliver many products, and the same product can be delivered by many suppliers. That is why this many-to-may relationship will be implemented through the main table CE_SALES

- ◆ *CE_BRANDS*

  each product can be produced by one brand (if there is an analogue of other brand – it's named differently, different packages, descriptions and so on)

2.1.2 DIM_CUSTOMER normalization

4 entities were separated from this dimension:

- ◆ CE_CUSTOMERS
- ◆ CE_ADDRESSES
- ◆ CE_CITIES
- ◆ CE_STATES

2.1.3 DIM_STORES normalization

As we have already normalized addresses, from this dimension we extract only:

- ◆ CE_STORES

2.1.4 DIM_EMPLOYEES normalization

- CE_EMPLOYEES

Employees dimension is already normalized, and it's decided no to separate gender or employee's position attributes, as there is no keys for them and they contain 2 values each and can be left as direct employees attributes.

2.1.5 DIM_PROMOTIONS normalization

As promotion has hierarchies, 3 entities are separated. CE_PROMO_CHANNELS and CE_PROMO_CATEGORIES → independent hierarchies of promotions. Channel is promotions distribution method (TV, radio, online etc.), category is usually gradation according to medical purpose.

- CE_PROMOTIONS
- CE_PROMO_CATEGORIES
- CE_PROMO_CHANNELS

2.1.6 DIM_PAYMENT_METHODS normalization

This dimension is already normalized, just create table in model based on it

- CE_PAYMENT_METHODS

2.1.7 DIM_SALES_CHANNELS normalization

This dimension is already normalized, just create table in model based on it

- CE_SALES_CHANNELS

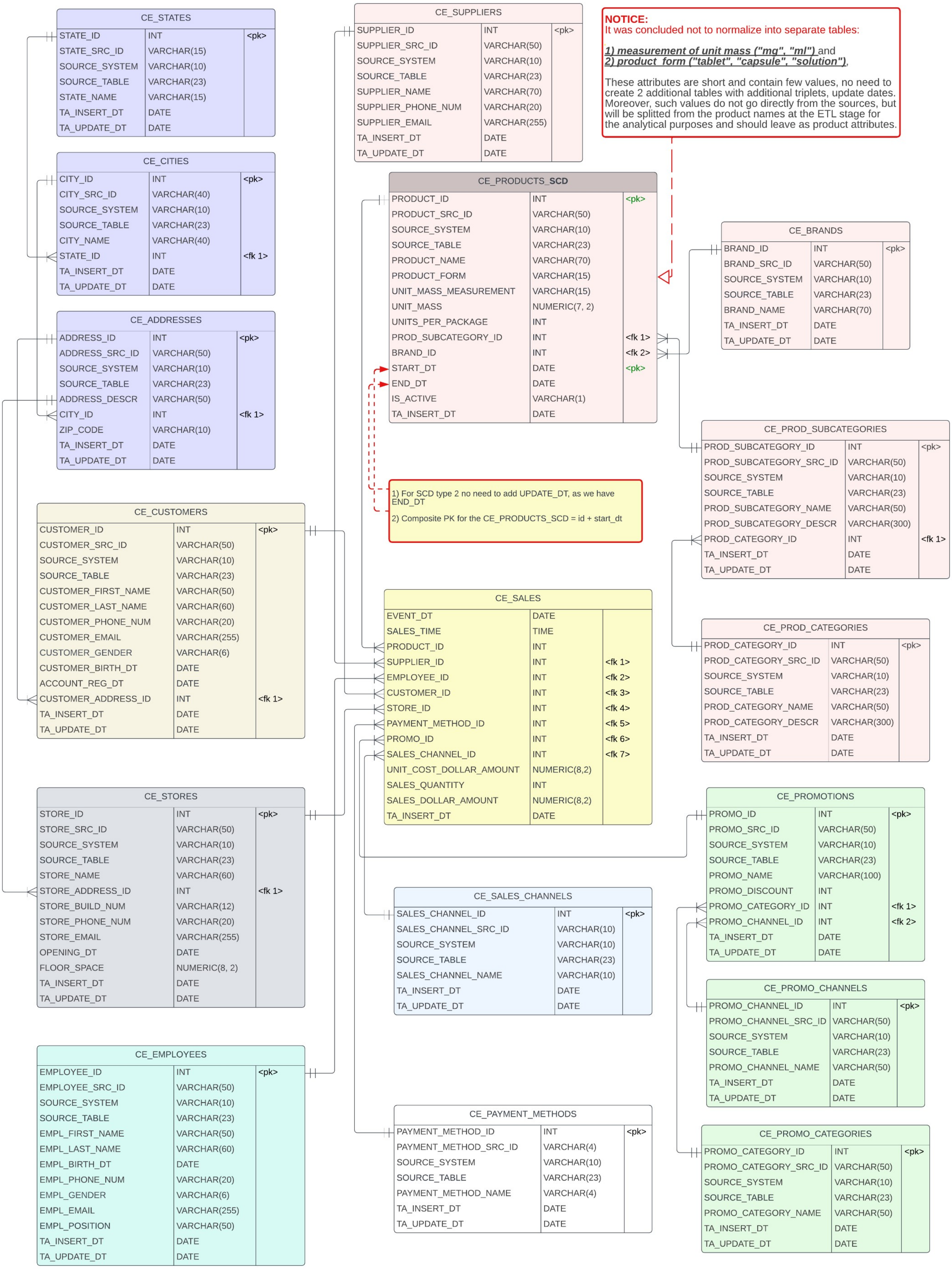
2.1.8 FCT_SALES_DD normalization

Into the CE_SALES go almost all attributes, that are present in fact table, except of the 4 additional calculated facts (extended amounts, profit and so on). They will occur at the BL_DM layer. In the BL_3NF only initial facts are stored.

At the last stage, all attributes related to the source triplet, insertion and update dates are added in the appropriate order.

PK for the CE_PRODUCTS_SCD at BL_3NF = PRODUCT_ID + START_DT

The next page contains BL_3NF.png



CE_STATES

STATE_ID	INT	<pk>
STATE_SRC_ID	VARCHAR(15)	
SOURCE_SYSTEM	VARCHAR(10)	
SOURCE_TABLE	VARCHAR(23)	
STATE_NAME	VARCHAR(15)	
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

CE_CITIES

CITY_ID	INT	<pk>
CITY_SRC_ID	VARCHAR(40)	
SOURCE_SYSTEM	VARCHAR(10)	
SOURCE_TABLE	VARCHAR(23)	
CITY_NAME	VARCHAR(40)	
STATE_ID	INT	<fk 1>
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

CE_ADDRESSES

ADDRESS_ID	INT	<pk>
ADDRESS_SRC_ID	VARCHAR(50)	
SOURCE_SYSTEM	VARCHAR(10)	
SOURCE_TABLE	VARCHAR(23)	
ADDRESS_DESCR	VARCHAR(50)	
CITY_ID	INT	<fk 1>
ZIP_CODE	VARCHAR(10)	
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

CE_CUSTOMERS

CUSTOMER_ID	INT	<pk>
CUSTOMER_SRC_ID	VARCHAR(50)	
SOURCE_SYSTEM	VARCHAR(10)	
SOURCE_TABLE	VARCHAR(23)	
CUSTOMER_FIRST_NAME	VARCHAR(50)	
CUSTOMER_LAST_NAME	VARCHAR(60)	
CUSTOMER_PHONE_NUM	VARCHAR(20)	
CUSTOMER_EMAIL	VARCHAR(255)	
CUSTOMER_GENDER	VARCHAR(6)	
CUSTOMER_BIRTH_DT	DATE	
ACCOUNT_REG_DT	DATE	
CUSTOMER_ADDRESS_ID	INT	<fk 1>
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

CE_STORES

STORE_ID	INT	<pk>
STORE_SRC_ID	VARCHAR(50)	
SOURCE_SYSTEM	VARCHAR(10)	
SOURCE_TABLE	VARCHAR(23)	
STORE_NAME	VARCHAR(60)	
STORE_ADDRESS_ID	INT	<fk 1>
STORE_BUILD_NUM	VARCHAR(12)	
STORE_PHONE_NUM	VARCHAR(20)	
STORE_EMAIL	VARCHAR(255)	
OPENING_DT	DATE	
FLOOR_SPACE	NUMERIC(8, 2)	
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

CE_EMPLOYEES

EMPLOYEE_ID	INT	<pk>
EMPLOYEE_SRC_ID	VARCHAR(50)	
SOURCE_SYSTEM	VARCHAR(10)	
SOURCE_TABLE	VARCHAR(23)	
EMPL_FIRST_NAME	VARCHAR(50)	
EMPL_LAST_NAME	VARCHAR(60)	
EMPL_BIRTH_DT	DATE	
EMPL_PHONE_NUM	VARCHAR(20)	
EMPL_GENDER	VARCHAR(6)	
EMPL_EMAIL	VARCHAR(255)	
EMPL_POSITION	VARCHAR(50)	
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

CE_SUPPLIERS

SUPPLIER_ID	INT	<pk>
SUPPLIER_SRC_ID	VARCHAR(50)	
SOURCE_SYSTEM	VARCHAR(10)	
SOURCE_TABLE	VARCHAR(23)	
SUPPLIER_NAME	VARCHAR(70)	
SUPPLIER_PHONE_NUM	VARCHAR(20)	
SUPPLIER_EMAIL	VARCHAR(255)	
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

NOTICE:
It was concluded not to normalize into separate tables:

1) measurement of unit mass ("mg", "ml") and
2) product_form ("tablet", "capsule", "solution"),

These attributes are short and contain few values, no need to create 2 additional tables with additional triplets, update dates. Moreover, such values do not go directly from the sources, but will be splitted from the product names at the ETL stage for the analytical purposes and should leave as product attributes.

CE_PRODUCTS_SCD

PRODUCT_ID	INT	<pk>
PRODUCT_SRC_ID	VARCHAR(50)	
SOURCE_SYSTEM	VARCHAR(10)	
SOURCE_TABLE	VARCHAR(23)	
PRODUCT_NAME	VARCHAR(70)	
PRODUCT_FORM	VARCHAR(15)	
UNIT_MASS_MEASUREMENT	VARCHAR(15)	
UNIT_MASS	NUMERIC(7, 2)	
UNITS_PER_PACKAGE	INT	
PROD_SUBCATEGORY_ID	INT	<fk 1>
BRAND_ID	INT	<fk 2>
START_DT	DATE	<pk>
END_DT	DATE	
IS_ACTIVE	VARCHAR(1)	
TA_INSERT_DT	DATE	

1) For SCD type 2 no need to add UPDATE_DT, as we have END_DT

2) Composite PK for the CE_PRODUCTS_SCD = id + start_dt

CE_BRANDS

BRAND_ID	INT	<pk>
BRAND_SRC_ID	VARCHAR(50)	
SOURCE_SYSTEM	VARCHAR(10)	
SOURCE_TABLE	VARCHAR(23)	
BRAND_NAME	VARCHAR(70)	
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

CE_PROD_SUBCATEGORIES

PROD_SUBCATEGORY_ID	INT	<pk>
PROD_SUBCATEGORY_SRC_ID	VARCHAR(50)	
SOURCE_SYSTEM	VARCHAR(10)	
SOURCE_TABLE	VARCHAR(23)	
PROD_SUBCATEGORY_NAME	VARCHAR(50)	
PROD_SUBCATEGORY_DESCR	VARCHAR(300)	
PROD_CATEGORY_ID	INT	<fk 1>
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

CE_SALES

EVENT_DT	DATE	
SALES_TIME	TIME	
PRODUCT_ID	INT	
SUPPLIER_ID	INT	<fk 1>
EMPLOYEE_ID	INT	<fk 2>
CUSTOMER_ID	INT	<fk 3>
STORE_ID	INT	<fk 4>
PAYMENT_METHOD_ID	INT	<fk 5>
PROMO_ID	INT	<fk 6>
SALES_CHANNEL_ID	INT	<fk 7>
UNIT_COST_DOLLAR_AMOUNT	NUMERIC(8,2)	
SALES_QUANTITY	INT	
SALES_DOLLAR_AMOUNT	NUMERIC(8,2)	
TA_INSERT_DT	DATE	

CE_PROD_CATEGORIES

PROD_CATEGORY_ID	INT	<pk>
PROD_CATEGORY_SRC_ID	VARCHAR(50)	
SOURCE_SYSTEM	VARCHAR(10)	
SOURCE_TABLE	VARCHAR(23)	
PROD_CATEGORY_NAME	VARCHAR(50)	
PROD_CATEGORY_DESCR	VARCHAR(300)	
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

CE_PROMOTIONS

PROMO_ID	INT	<pk>
PROMO_SRC_ID	VARCHAR(50)	
SOURCE_SYSTEM	VARCHAR(10)	
SOURCE_TABLE	VARCHAR(23)	
PROMO_NAME	VARCHAR(100)	
PROMO_DISCOUNT	INT	
PROMO_CATEGORY_ID	INT	<fk 1>
PROMO_CHANNEL_ID	INT	<fk 2>
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

CE_SALES_CHANNELS

SALES_CHANNEL_ID	INT	<pk>
SALES_CHANNEL_SRC_ID	VARCHAR(10)	
SOURCE_SYSTEM	VARCHAR(10)	
SOURCE_TABLE	VARCHAR(23)	
SALES_CHANNEL_NAME	VARCHAR(10)	
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

CE_PROMO_CHANNELS

PROMO_CHANNEL_ID	INT	<pk>
PROMO_CHANNEL_SRC_ID	VARCHAR(50)	
SOURCE_SYSTEM	VARCHAR(10)	
SOURCE_TABLE	VARCHAR(23)	
PROMO_CHANNEL_NAME	VARCHAR(50)	
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

CE_PAYMENT_METHODS

PAYMENT_METHOD_ID	INT	<pk>
PAYMENT_METHOD_SRC_ID	VARCHAR(4)	
SOURCE_SYSTEM	VARCHAR(10)	
SOURCE_TABLE	VARCHAR(23)	
PAYMENT_METHOD_NAME	VARCHAR(4)	
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

CE_PROMO_CATEGORIES

PROMO_CATEGORY_ID	INT	<pk>
PROMO_CATEGORY_SRC_ID	VARCHAR(50)	
SOURCE_SYSTEM	VARCHAR(10)	
SOURCE_TABLE	VARCHAR(23)	
PROMO_CATEGORY_NAME	VARCHAR(50)	
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

3 BUSINESS LAYER DIMENSIONAL MODEL

3.1 DENORMALIZATION

We have already described all dimensions, then normalized them up to 3NF. At this point we are going to denormalize it to the appropriate dimensional level. The BL_DM schema will contain

10 dimensions and 1 fact table (with 8 metrics).

- ✓ CE_PROD_CATEGORIES, CE_PROD_SUBCATEGORIES, CE_BRANDS and CE_PRODUCTS_SCD will be joined by ids and combined into:

  1. DIM_PRODUCTS_SCD

- ✓ CE_SUPPLIERS is independent entity as mentioned above and it can't be denormalized further. Thus, it will be a separate:

  2. DIM_ SUPPLIERS

- ✓ CE_ADDRESSES, CE_CITIES, AND CE_STATES will be joined and included as address attributes into both:

  3. DIM_CUSTOMERS

  4. DIM_STORES

- ✓ CE_EMPLOYEES cannot be denormalized further and will be:

  5. DIM_EMPLOYEES

- ✓ CE_SALES_CHANNELS will be loaded into:

  6. DIM_SALES_CHANNELS

- ✓ CE_PROMO_CATEGORIES and CE_PROMO_CHANNELS will be joined with CE_PROMOTIONS (as 2 separate hierarchies, not one!) and go into:

  7. DIM_PROMOTIONS

- ✓ CE_PAYMENT_METHODS just goes to:

  8. DIM_PAYMENT_METHODS

- ✓ Generated by SQL script:

  9. DIM_TIME_DAY (dates)

  10. DIM_TIME_OF_DAY (times)

Additional calculated metrics will be added to the FCT_SALES_DD (see BL_DM diagram)

3.2 DIM_TIME_OF_DAY

Reasons for including a separate times dimension (HH:MM:SS → exact time of the sales receipt registration):

✓ The GRAIN is defined as one product line on the sales receipt. The uniqueness of the GRAIN (each row in the FCT_SALES_DD) is maintained either by receipt number, either by exact time(HH:MM:SS). Because the same purchase (same product, same quantity, same customer, same pharmacy... can be made several times during one day). Also one customer can make an order online, than come back and repeat the same order in the same quantity. Our grain is determined by receipt.

✓ It was decided not to store receipt number attribute as a degenerated dimension since it doesn't provide much benefit

✓ But exact time is needed for analytical tasks, listed in the chapter 1 (analyzing online shopping behavior by time of day for marketing strategies, analyzing pharmacy workload by hours)

✓ When attribute is often used for grouping, it's better to separate it into dimension

10. DIM_TIME_OF_DAY

Column name	Description	Data Type
TIME_OF_DAY_SURR_ID	Generated by sequence INT	INT
TIME_OF_DAY	PK: Standard time format: HH:MM:SS (12:00:15)	TIME
HOUR_24	Hour in day in 24-hour format (0-23)	INT
HOUR_12	Hour in day in 12-hour format with abbreviation AM/PM (“01 AM”, “11 PM”)	VARCHAR(5)
MINUTE_OF_HOUR	Minutes number in the hour (0-59)	INT
SECOND_OF_HOUR	Seconds number in the hour(0-59)	INT

Example with filled data

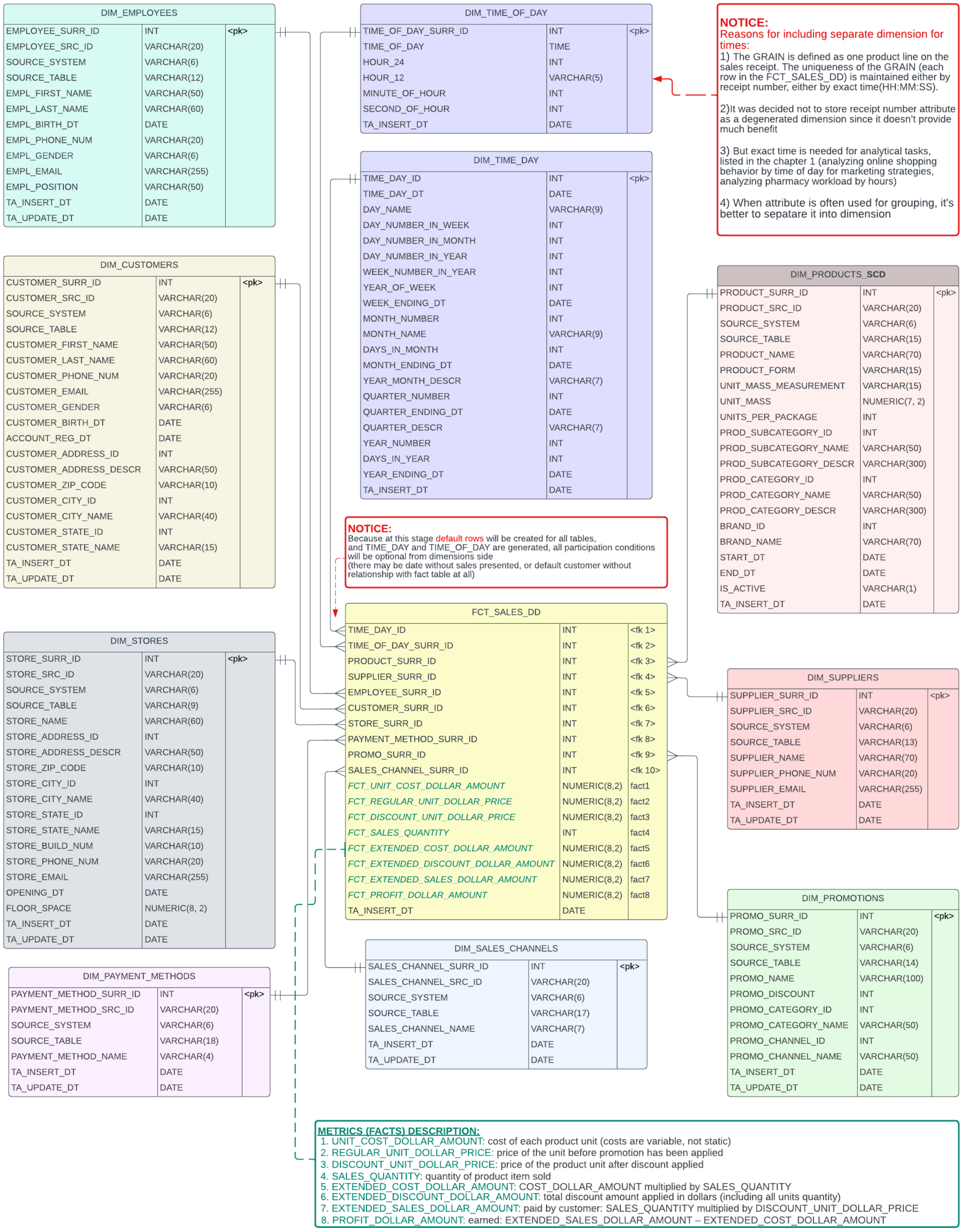
TIME_OF_DAY_SURR_ID	TIME_OF_DAY	HOUR_24	HOUR_12	MINUTE_OF_HOUR	SECOND_OF_HOUR
467	13:28:14	13	01 PM	28	14

3.3 OTHER DETAILS

- NO need to add extra column UPDATE_DT for DIM_TIME_DAY and DIM_TIME_OF_DAY, because all values are constant and can’t be changed. Also there are no source triplets for them, because these tables are generated
- Although at the BL_3NF layer all surrogate id’s are created as integer sequences, by naming convention we must specify the data type of SRC_ID attribute at the BL_DM layer as VARCHAR as well
- When joining hierarchies, no need to put source triplets for all entities, but only for the general one (for product only in DIM_PRODUCTS), because related categories and subcategories can be extracted by joining with products at the BL_3NF layer
- INT data type is used instead of BIGINT to save space, because the maximum integer 2,147,483,647 has been calculated and analyzed to be large enough to store all dimensions keys for the given business process. Even if we need to extend dates dimension up to 9999 year
- All table names of the level BL_3NF are already specified. So, we know exact lengths of all source tables for the BL_DM layer. The precise VARCHAR length is used for all source_table references to save space. For instance: LEN(“CE_EMPLOYEES”) = 12. Use VARCHAR(12) type for the source_table attribute in the DIM_EMPLOYEES
- For DIM_TIME_DAY the PK isn’t surrogate, and not a combination like YYYYMMDD, but just DATE itself

The next page shows the BL_DM diagram with metrics description.

BL_DM diagram



DIM_EMPLOYEES		
EMPLOYEE_SURR_ID	INT	<pk>
EMPLOYEE_SRC_ID	VARCHAR(20)	
SOURCE_SYSTEM	VARCHAR(6)	
SOURCE_TABLE	VARCHAR(12)	
EMPL_FIRST_NAME	VARCHAR(50)	
EMPL_LAST_NAME	VARCHAR(60)	
EMPL_BIRTH_DT	DATE	
EMPL_PHONE_NUM	VARCHAR(20)	
EMPL_GENDER	VARCHAR(6)	
EMPL_EMAIL	VARCHAR(255)	
EMPL_POSITION	VARCHAR(50)	
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

DIM_TIME_OF_DAY		
TIME_OF_DAY_SURR_ID	INT	<pk>
TIME_OF_DAY	TIME	
HOUR_24	INT	
HOUR_12	VARCHAR(5)	
MINUTE_OF_HOUR	INT	
SECOND_OF_HOUR	INT	
TA_INSERT_DT	DATE	

NOTICE:
Reasons for including separate dimension for times:

1) The GRAIN is defined as one product line on the sales receipt. The uniqueness of the GRAIN (each row in the FCT_SALES_DD) is maintained either by receipt number, either by exact time(HH:MM:SS).

2)It was decided not to store receipt number attribute as a degenerated dimension since it doesn't provide much benefit

3) But exact time is needed for analytical tasks, listed in the chapter 1 (analyzing online shopping behavior by time of day for marketing strategies, analyzing pharmacy workload by hours)

4) When attribute is often used for grouping, it's better to sepatare it into dimension

DIM_TIME_DAY		
TIME_DAY_ID	INT	<pk>
TIME_DAY_DT	DATE	
DAY_NAME	VARCHAR(9)	
DAY_NUMBER_IN_WEEK	INT	
DAY_NUMBER_IN_MONTH	INT	
DAY_NUMBER_IN_YEAR	INT	
WEEK_NUMBER_IN_YEAR	INT	
YEAR_OF_WEEK	INT	
WEEK_ENDING_DT	DATE	
MONTH_NUMBER	INT	
MONTH_NAME	VARCHAR(9)	
DAYS_IN_MONTH	INT	
MONTH_ENDING_DT	DATE	
YEAR_MONTH_DESCR	VARCHAR(7)	
QUARTER_NUMBER	INT	
QUARTER_ENDING_DT	DATE	
QUARTER_DESCR	VARCHAR(7)	
YEAR_NUMBER	INT	
DAYS_IN_YEAR	INT	
YEAR_ENDING_DT	DATE	
TA_INSERT_DT	DATE	

DIM_CUSTOMERS		
CUSTOMER_SURR_ID	INT	<pk>
CUSTOMER_SRC_ID	VARCHAR(20)	
SOURCE_SYSTEM	VARCHAR(6)	
SOURCE_TABLE	VARCHAR(12)	
CUSTOMER_FIRST_NAME	VARCHAR(50)	
CUSTOMER_LAST_NAME	VARCHAR(60)	
CUSTOMER_PHONE_NUM	VARCHAR(20)	
CUSTOMER_EMAIL	VARCHAR(255)	
CUSTOMER_GENDER	VARCHAR(6)	
CUSTOMER_BIRTH_DT	DATE	
ACCOUNT_REG_DT	DATE	
CUSTOMER_ADDRESS_ID	INT	
CUSTOMER_ADDRESS_DESCR	VARCHAR(50)	
CUSTOMER_ZIP_CODE	VARCHAR(10)	
CUSTOMER_CITY_ID	INT	
CUSTOMER_CITY_NAME	VARCHAR(40)	
CUSTOMER_STATE_ID	INT	
CUSTOMER_STATE_NAME	VARCHAR(15)	
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

DIM_PRODUCTS_SCD		
PRODUCT_SURR_ID	INT	<pk>
PRODUCT_SRC_ID	VARCHAR(20)	
SOURCE_SYSTEM	VARCHAR(6)	
SOURCE_TABLE	VARCHAR(15)	
PRODUCT_NAME	VARCHAR(70)	
PRODUCT_FORM	VARCHAR(15)	
UNIT_MASS_MEASUREMENT	VARCHAR(15)	
UNIT_MASS	NUMERIC(7, 2)	
UNITS_PER_PACKAGE	INT	
PROD_SUBCATEGORY_ID	INT	
PROD_SUBCATEGORY_NAME	VARCHAR(50)	
PROD_SUBCATEGORY_DESCR	VARCHAR(300)	
PROD_CATEGORY_ID	INT	
PROD_CATEGORY_NAME	VARCHAR(50)	
PROD_CATEGORY_DESCR	VARCHAR(300)	
BRAND_ID	INT	
BRAND_NAME	VARCHAR(70)	
START_DT	DATE	
END_DT	DATE	
IS_ACTIVE	VARCHAR(1)	
TA_INSERT_DT	DATE	

NOTICE:
Because at this stage default rows will be created for all tables,
and TIME_DAY and TIME_OF_DAY are generated, all participation conditions will be optional from dimensions side
(there may be date without sales presented, or default customer without relationship with fact table at all)

FCT_SALES_DD		
TIME_DAY_ID	INT	<fk 1>
TIME_OF_DAY_SURR_ID	INT	<fk 2>
PRODUCT_SURR_ID	INT	<fk 3>
SUPPLIER_SURR_ID	INT	<fk 4>
EMPLOYEE_SURR_ID	INT	<fk 5>
CUSTOMER_SURR_ID	INT	<fk 6>
STORE_SURR_ID	INT	<fk 7>
PAYMENT_METHOD_SURR_ID	INT	<fk 8>
PROMO_SURR_ID	INT	<fk 9>
SALES_CHANNEL_SURR_ID	INT	<fk 10>
FCT_UNIT_COST_DOLLAR_AMOUNT	NUMERIC(8,2)	fact1
FCT_REGULAR_UNIT_DOLLAR_PRICE	NUMERIC(8,2)	fact2
FCT_DISCOUNT_UNIT_DOLLAR_PRICE	NUMERIC(8,2)	fact3
FCT_SALES_QUANTITY	INT	fact4
FCT_EXTENDED_COST_DOLLAR_AMOUNT	NUMERIC(8,2)	fact5
FCT_EXTENDED_DISCOUNT_DOLLAR_AMOUNT	NUMERIC(8,2)	fact6
FCT_EXTENDED_SALES_DOLLAR_AMOUNT	NUMERIC(8,2)	fact7
FCT_PROFIT_DOLLAR_AMOUNT	NUMERIC(8,2)	fact8
TA_INSERT_DT	DATE	

DIM_STORES		
STORE_SURR_ID	INT	<pk>
STORE_SRC_ID	VARCHAR(20)	
SOURCE_SYSTEM	VARCHAR(6)	
SOURCE_TABLE	VARCHAR(9)	
STORE_NAME	VARCHAR(60)	
STORE_ADDRESS_ID	INT	
STORE_ADDRESS_DESCR	VARCHAR(50)	
STORE_ZIP_CODE	VARCHAR(10)	
STORE_CITY_ID	INT	
STORE_CITY_NAME	VARCHAR(40)	
STORE_STATE_ID	INT	
STORE_STATE_NAME	VARCHAR(15)	
STORE_BUILD_NUM	VARCHAR(10)	
STORE_PHONE_NUM	VARCHAR(20)	
STORE_EMAIL	VARCHAR(255)	
OPENING_DT	DATE	
FLOOR_SPACE	NUMERIC(8, 2)	
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

DIM_SUPPLIERS		
SUPPLIER_SURR_ID	INT	<pk>
SUPPLIER_SRC_ID	VARCHAR(20)	
SOURCE_SYSTEM	VARCHAR(6)	
SOURCE_TABLE	VARCHAR(13)	
SUPPLIER_NAME	VARCHAR(70)	
SUPPLIER_PHONE_NUM	VARCHAR(20)	
SUPPLIER_EMAIL	VARCHAR(255)	
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

DIM_PROMOTIONS		
PROMO_SURR_ID	INT	<pk>
PROMO_SRC_ID	VARCHAR(20)	
SOURCE_SYSTEM	VARCHAR(6)	
SOURCE_TABLE	VARCHAR(14)	
PROMO_NAME	VARCHAR(100)	
PROMO_DISCOUNT	INT	
PROMO_CATEGORY_ID	INT	
PROMO_CATEGORY_NAME	VARCHAR(50)	
PROMO_CHANNEL_ID	INT	
PROMO_CHANNEL_NAME	VARCHAR(50)	
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

DIM_SALES_CHANNELS		
SALES_CHANNEL_SURR_ID	INT	<pk>
SALES_CHANNEL_SRC_ID	VARCHAR(20)	
SOURCE_SYSTEM	VARCHAR(6)	
SOURCE_TABLE	VARCHAR(17)	
SALES_CHANNEL_NAME	VARCHAR(7)	
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

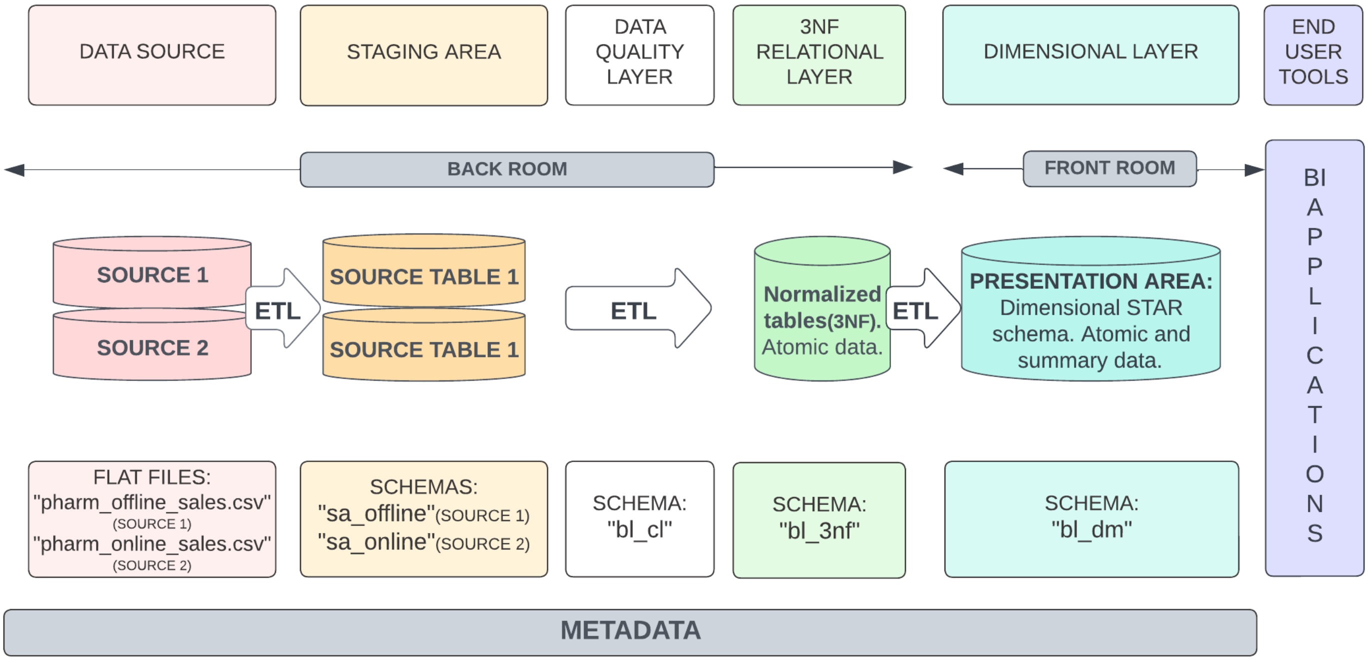
DIM_PAYMENT_METHODS		
PAYMENT_METHOD_SURR_ID	INT	<pk>
PAYMENT_METHOD_SRC_ID	VARCHAR(20)	
SOURCE_SYSTEM	VARCHAR(6)	
SOURCE_TABLE	VARCHAR(18)	
PAYMENT_METHOD_NAME	VARCHAR(4)	
TA_INSERT_DT	DATE	
TA_UPDATE_DT	DATE	

METRICS (FACTS) DESCRIPTION:

1. UNIT_COST_DOLLAR_AMOUNT: cost of each product unit (costs are variable, not static)
2. REGULAR_UNIT_DOLLAR_PRICE: price of the unit before promotion has been applied
3. DISCOUNT_UNIT_DOLLAR_PRICE: price of the product unit after discount applied
4. SALES_QUANTITY: quantity of product item sold
5. EXTENDED_COST_DOLLAR_AMOUNT: COST_DOLLAR_AMOUNT multiplied by SALES_QUANTITY
6. EXTENDED_DISCOUNT_DOLLAR_AMOUNT: total discount amount applied in dollars (including all units quantity)
7. EXTENDED_SALES_DOLLAR_AMOUNT: paid by customer: SALES_QUANTITY multiplied by DISCOUNT_UNIT_DOLLAR_PRICE
8. PROFIT_DOLLAR_AMOUNT: earned: EXTENDED_SALES_DOLLAR_AMOUNT – EXTENDED_COST_DOLLAR_AMOUNT

4 LOGICAL SCHEME

Note: *"Data quality"* is specified as a separate layer, because it covers many ETL data quality operations that were prformed: validations, transformations, cleansing, deduplication, and others.

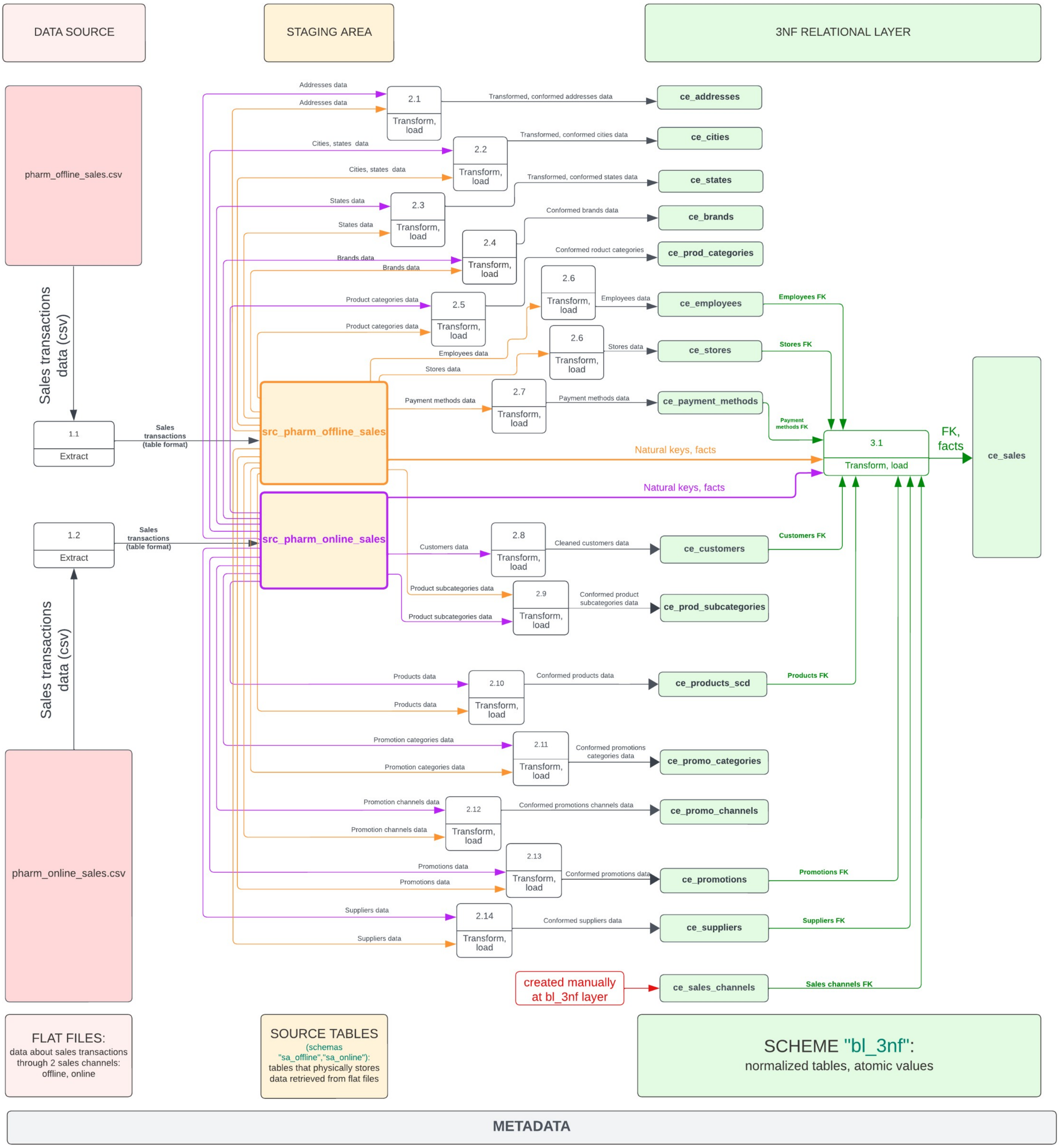

DATA SOURCE
STAGING AREA
DATA QUALITY LAYER
3NF RELATIONAL LAYER
DIMENSIONAL LAYER
END USER TOOLS
BACK ROOM
FRONT ROOM
BI APPLICATIONS
SOURCE 1
SOURCE 2
ETL
SOURCE TABLE 1
SOURCE TABLE 1
ETL
Normalized tables(3NF). Atomic data.
ETL
PRESENTATION AREA: Dimensional STAR schema. Atomic and summary data.
FLAT FILES: "pharm_offline_sales.csv" (SOURCE 1) "pharm_online_sales.csv" (SOURCE 2)
SCHEMAS: "sa_offline"(SOURCE 1) "sa_online"(SOURCE 2)
SCHEMA: "bl_cl"
SCHEMA: "bl_3nf"
SCHEMA: "bl_dm"
METADATA

5 DATA FLOW

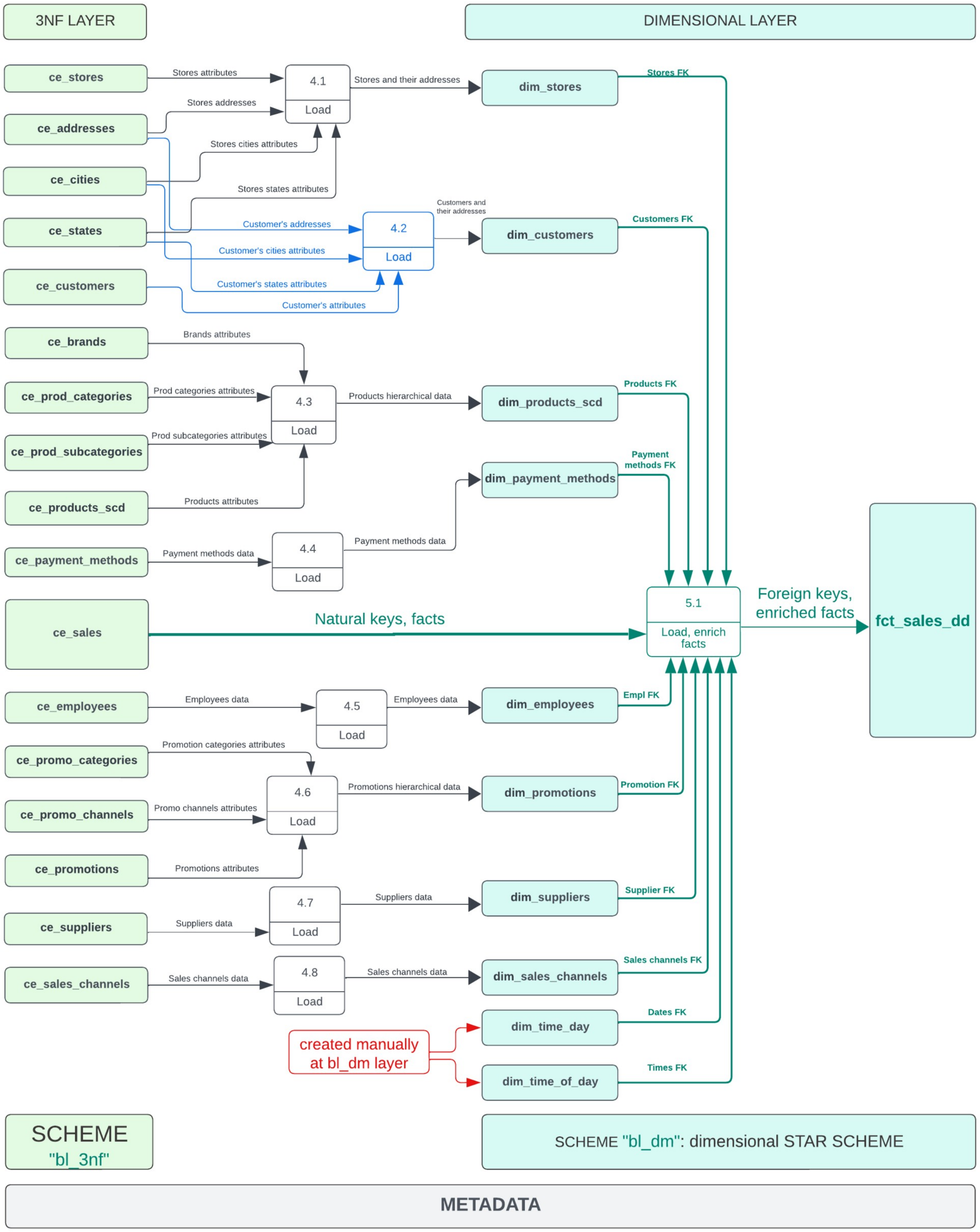
Notes:

1. For readability DFD is split into 2 parts:
   - data flow from sources to bl_3nf
   - data from from bl_3f to bl_dm layer
2. Entities that occur at the layer (not come from sources, but created) marked with comments

DFD Part 1


DATA SOURCE
STAGING AREA
3NF RELATIONAL LAYER
pharm_offline_sales.csv
Sales transactions data (csv)
1.1
Extract
Sales transactions (table format)
1.2
Extract
Sales transactions (table format)
Sales transactions data (csv)
pharm_online_sales.csv
src_pharm_offline_sales
src_pharm_online_sales
Addresses data
Addresses data
2.1
Transform, load
Transformed, conformed addresses data
ce_addresses
Cities, states data
Cities, states data
2.2
Transform, load
Transformed, conformed cities data
ce_cities
Transformed, conformed states data
ce_states
States data
States data
2.3
Transform, load
Conformed brands data
ce_brands
Brands data
Brands data
2.4
Transform, load
Conformed roduct categories
ce_prod_categories
Product categories data
Product categories data
2.5
Transform, load
2.6
Transform, load
Employees data
ce_employees
Employees data
Stores data
2.6
Transform, load
Stores data
ce_stores
Payment methods data
2.7
Transform, load
Payment methods data
ce_payment_methods
Employees FK
Stores FK
Payment methods FK
Natural keys, facts
Natural keys, facts
3.1
Transform, load
FK, facts
ce_sales
Customers data
2.8
Transform, load
Cleaned customers data
ce_customers
Customers FK
Product subcategories data
Product subcategories data
2.9
Transform, load
Conformed product subcategories data
ce_prod_subcategories
Products data
Products data
2.10
Transform, load
Conformed products data
ce_products_scd
Products FK
Promotion categories data
Promotion categories data
2.11
Transform, load
Conformed promotions categories data
ce_promo_categories
Promotion channels data
Promotion channels data
2.12
Transform, load
Conformed promotions channels data
ce_promo_channels
Promotions data
Promotions data
2.13
Transform, load
Conformed promotions data
ce_promotions
Promotions FK
Suppliers data
Suppliers data
2.14
Transform, load
Conformed suppliers data
ce_suppliers
Suppliers FK
created manually at bl_3nf layer
ce_sales_channels
Sales channels FK
FLAT FILES:
data about sales transactions through 2 sales channels: offline, online
SOURCE TABLES
(schemas "sa_offline","sa_online"):
tables that physically stores data retrieved from flat files
SCHEME "bl_3nf":
normalized tables, atomic values
METADATA

DFD Part 2


3NF LAYER
DIMENSIONAL LAYER
ce_stores
ce_addresses
ce_cities
ce_states
ce_customers
ce_brands
ce_prod_categories
ce_prod_subcategories
ce_products_scd
ce_payment_methods
ce_sales
ce_employees
ce_promo_categories
ce_promo_channels
ce_promotions
ce_suppliers
ce_sales_channels
Stores attributes
Stores addresses
Stores cities attributes
Stores states attributes
4.1
Load
Stores and their addresses
dim_stores
Stores FK
Customer's addresses
Customer's cities attributes
Customer's states attributes
Customer's attributes
4.2
Load
Customers and their addresses
dim_customers
Customers FK
Brands attributes
Prod categories attributes
Prod subcategories attributes
Products attributes
4.3
Load
Products hierarchical data
dim_products_scd
Products FK
Payment methods data
4.4
Load
Payment methods data
dim_payment_methods
Payment methods FK
Natural keys, facts
5.1
Load, enrich facts
Foreign keys, enriched facts
fct_sales_dd
Employees data
4.5
Load
Employees data
dim_employees
Empl FK
Promotion categories attributes
Promo channels attributes
Promotions attributes
4.6
Load
Promotions hierarchical data
dim_promotions
Promotion FK
Suppliers data
4.7
Load
Suppliers data
dim_suppliers
Supplier FK
Sales channels data
4.8
Load
Sales channels data
dim_sales_channels
Sales channels FK
created manually at bl_dm layer
dim_time_day
Dates FK
dim_time_of_day
Times FK
SCHEME "bl_3nf"
SCHEME "bl_dm": dimensional STAR SCHEME
METADATA

6 FACT TABLE PARTITIONING STRATEGY

Partitions key

Key: dates. Because it will be used for analysis in most cases when extracting data. Almost all analytics will be closely tied with dates periods (sales by dates, profits, comparison of sales across different attributes and by dates, comparisons with previous periods and other).

Partitions period

Partitions: monthly. Because incremental loading will be performed each month at bl_dm layer, it will be convenient to “open” each time new partition. And the majority of analytical queries will be related to aggregated by months sales. If for instance all queries retrieved the entire data from sales table, the partitions would be useless at all.

Key range values

Range values: integers. Not dates itself. Because fact table will contain reference to the dim_time_day as meaningful combination of date’s digits (20220101). And integer is softer than dates when calculating. Moreover, these combinations (20200201, 20230203...) are ranged well ant suit partition’s goal.

Partitions implementation

- Partitions are created dynamically in procedure that loads data into fct_sales_dd:
- Each time data will be loaded, the rolling period will be taken to refresh some data (for instance if during previous 2 months that were already loaded some transactions occur, or not all transactions were covered accurately)
- The procedure will consider all month starting from the rolling period and ending with the end of current month.
- The table for each month will be created(if not exists), than partition DETACHed (if already exists), CHECK constraints created(if not exists), tables filled with appropriate monthly data, and then ATTACHED again.