## Business Template

# PIZZA PLACE RESTAURANT

**Logo / Image**

CC

# 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
1.4 DATASETS DESCRIPTION ..... 3
1.5 GRAIN / DIM / FACT ..... 5
2 BUSINESS LAYER 3NF ..... 1
3 BUSINESS LAYER DIMENSIONAL MODEL ..... 13
4 LOGICAL SCHEME ..... 14
5 DATA FLOW ..... 15
6 FACT TABLE PARTITIONING STRATEGY ..... 16

# 1 BUSINESS DESCRIPTION

## 1.1 BUSINESS BACKGROUND

Pizza nowadays is one of the most iconic resemblances of Italian cuisine. There are a lot of Pizza shops in the cities all over the world because Pizzas are extremely tasty and affordable.

So Oleg and Olga decided to create their own pizza restaurant in our city. From the start they implemented a little mobile app to register there to be able to make orders on the house. Just a simple registration with a name and achieving individual customer number and free pizza as a gift.

Olga and Oleg have a lot of friends in delivery business, so they managed to connect deliveries to their restaurant straight away and more than that, they figured out a way to track couriers and customers from delivery companies. So, they can aggregate additional data on the customers and couriers, to be able to analyze it in the future.

## 1.2 PROBLEMS BECAUSE OF POOR DATA MANAGEMENT

Poor data management doesn't let to do successful business because of insufficient information about what should you do next. If you don't use instruments which can give you information for analysis and which can help you to come up with a business strategy you won't be competitive in this or that field.

## 1.3 BENEFITS FROM IMPLEMENTING A DATA WAREHOUSE

Using of data warehouse can help you with the problems described above. Implementing a data warehouse can answer you the following questions:

- What is the common order size in restaurant and via delivery?
- Which customers order more from the delivery and which order more personally in the restaurant?
- Is there any correlation between customers and deliveries they use?

Further processing data would also let you:

- Correlate sales rates with type, name, size and price of pizza.
- Which delivery is the most common between our customers.
- And many other.

## 1.4 DATASETS DESCRIPTION

The first dataset contains the following information about sales in the restaurant (pizzaplace_rest.csv).

Product Information:

Pizza_type: type of pizzas (veggie or classic).

Pizza_name: pizza's name, which customer choose (Pepperoni, Mexicana and etc.).

Size: size of pizza (S, L, XL and etc).

Sales Information:

Price: price of pizza unit.

Order ID: unique identifier of an order.

Timestamp: date and time of an order.

Customer Information:

Customer ID: unique identifier of a customer.

Customer name: first and last name of a customer.

Employee Information:

Employee ID: unique identifier of an employee.

Employee name: first and last name of an employee.

Additional Attributes:

In or out: flag to indicate if the customer ate the order in the restaurant or took it away.

The second dataset contains the following information about online sales in our restaurant vie deliveries (pizzaplace_online.csv).

Product Information:

Pizza_type: type of pizzas (veggie or classic).

Pizza_name: pizza's name, which customer choose (Pepperoni, Mexicana and etc.).

Size: size of pizza (S, L, XL and etc).

Sales Information:

Price: price of pizza unit.

Order ID: unique identifier of an order.

Timestamp: date and time of an order.

Customer Information:

Customer ID: unique identifier of a customer.

Customer name: first and last name of a customer.

Geographical information:

District: name of the district of the customer.

Address: street\avenue name and house number of the customer.

Delivery Information:

Delivery ID: unique identifier of a delivery company.

Delivery name: first and last name of a delivery company.

Courier Information:

Courier ID: unique identifier of a courier.

Courier name: first and last name of a courier.

The datasets provide a comprehensive overview of pizza restaurant sales, allowing for analysis and exploration of sales performance, customer preferences, and more within the restaurant industry.

## 1.5 GRAIN / DIM / FACT

Step 1: Selecting Business Process.

Our business process is pizza retail.

Step 2: Declare the Grain:

Based on our business process, our grain is a line in a  receipt of an order made by customer, containing pizza, date, time and price data.

Step 3: Identify the Dimensions.

According to grain, we will have 5 dimensions: customers, order, product, date, time.

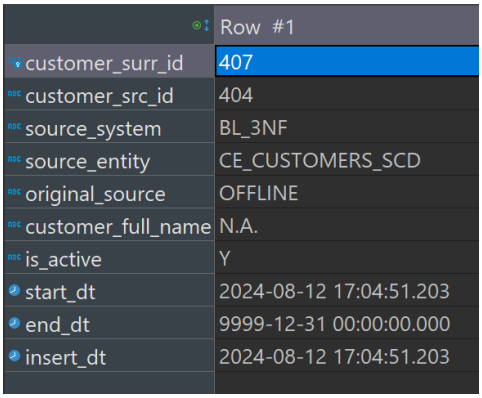
And one additional dimension to satisfy the task requirements: address.

DIM_CUSTOMERS_SCD

Dimension to store customer's data in our DWH with storing history data SCD type 2.

| Column name | Description | Data Type |
|---|---|---|
| Customer_surr_id | Unique identifier of a customer who made the order on this layer | BIGINT |
| Customer_src_id | Unique identifier of a customer who made the order on previous 3NF layer | INT |
| Source_system | Schema, data was taken from on previous 3NF layer | TEXT |
| Source_entity | Table, data was taken from on previous 3NF layer | TEXT |
| Original_source | Source of data about the customer (online/offline) | TEXT |
| Customer_full_name | Full name of a customer (first and last name) | TEXT |
| Is_active | Flag to state if this record active or history (Y/N) | TEXT |
| Start_dt | Start timestamp of when the record is active | TIMESTAMP |
| End_dt | End timestamp of when the record is active | TIMESTAMP |
| Insert_dt | Timestamp of insertion of this record in the table | TIMESTAMP |

Example with filled data



| | Row #1 |
|---|---|
| customer_surr_id | 407 |
| customer_src_id | 404 |
| source_system | BL_3NF |
| source_entity | CE_CUSTOMERS_SCD |
| original_source | OFFLINE |
| customer_full_name | N.A. |
| is_active | Y |
| start_dt | 2024-08-12 17:04:51.203 |
| end_dt | 9999-12-31 00:00:00.000 |
| insert_dt | 2024-08-12 17:04:51.203 |

DIM_ORDERS

Dimension to store order's data in our DWH.

Orders can be 2 types: in the restaurant and online via delivery apps. So there are default constraints to follow 'NOT NULL' rule.

INT's: default -1

TEXT: default 'NO'

| Column name | Description | Data Type |
|---|---|---|
| Order_surr_id | Unique identifier of order on this layer | BIGINT |
| Order_src_id | Unique identifier of order on previous 3NF layer | INT |
| Source_system | Schema, data was taken from on previous 3NF layer | TEXT |
| Source_entity | Table, data was taken from on previous 3NF layer | TEXT |
| Order_name | Name of the order | TEXT |
| Employee_src_id | Unique identifier of employee on previous 3NF layer | INT |
| Employee_full_name | Name of employee | TEXT |
| Order_type | Type of order | TEXT |
| Offline_order_type | Subtype of offline order | TEXT |
| Delivery_src_id | Unique identifier of delivery on previous 3NF layer | INT |
| Delivery_name | Name of Delivery | TEXT |
| Courier_src_id | Unique identifier of courier on previous 3NF layer | INT |
| Courier_full_name | Name of Courier | TEXT |
| Insert_dt | Timestamp of insertion of record | TIMESTAMP |
| Update_dt | Timestamp of update of record | TIMESTAMP |

Example with filled data

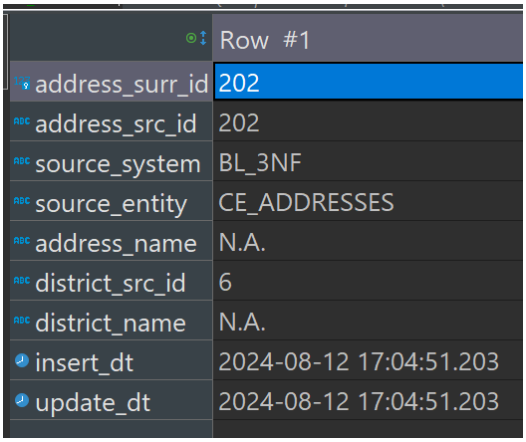
| | Row #1 |
|---|---|
| order_surr_id | 5,779 |
| order_src_id | 1 |
| source_system | BL_3NF |
| source_entity | CE_ORDERS |
| order_name | ORD101426 |
| employee_src_id | -1 |
| employee_full_name | N.A. |
| order_type | ONLINE |
| offline_order_type | N.A. |
| delivery_src_id | 39 |
| delivery_name | CLOVO |
| courier_src_id | 9 |
| courier_full_name | LARRY HERNANDEZ |
| insert_dt | 2024-08-11 11:55:20.117 |
| update_dt | 2024-08-11 11:55:20.117 |

DIM_ADDRESSES

Dimension to store addresses data in our DWH.

| Column name | Description | Data Type |
|---|---|---|
| Address_surr_id | Unique identifier of address on this layer | BIGINT |
| Address_src_id | Unique identifier of address on previous 3NF layer | INT |
| Source_system | Schema, data was taken from on previous 3NF layer | TEXT |
| Source_entity | Table, data was taken from on previous 3NF layer | TEXT |
| Address_name | Name of the address | TEXT |
| District_src_id | Unique identifier of district on previous 3NF layer | INT |
| District_name | Name of district | TEXT |
| Insert_dt | Timestamp of insertion of record | TIMESTAMP |
| Update_dt | Timestamp of update of record | TIMESTAMP |

Example with filled data



| | Row #1 |
|---|---|
| address_surr_id | 202 |
| address_src_id | 202 |
| source_system | BL_3NF |
| source_entity | CE_ADDRESSES |
| address_name | N.A. |
| district_src_id | 6 |
| district_name | N.A. |
| insert_dt | 2024-08-12 17:04:51.203 |
| update_dt | 2024-08-12 17:04:51.203 |

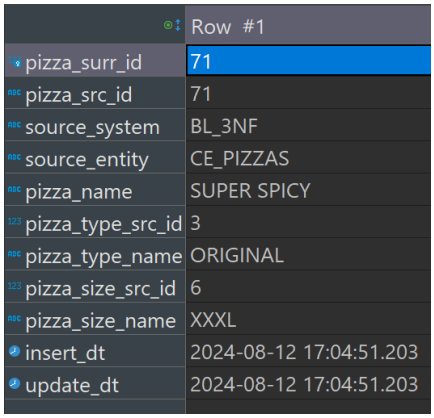
DIM_PIZZAS

Dimension to store product's data in our DWH.

Digital resemblance of a menu.

| Column name | Description | Data Type |
|---|---|---|
| Pizza_surr_id | Unique identifier of a product (pizza) | BIGINT |
| Pizza_src_id | Unique identifier of product (pizza) on previous 3NF layer | INT |
| Source_system | Schema, data was taken from on previous 3NF layer | TEXT |
| Source_entity | Table, data was taken from on previous 3NF layer | TEXT |
| Pizza_name | Name of product (pizza) as in Menu | TEXT |
| Pizza_type_src_id | Unique identifier of product (pizza) type on previous 3NF layer | INT |
| Pizza_type_mane | Type of product (pizza) | TEXT |
| Pizza_size_src_id | Unique identifier of product (pizza) size on previous 3NF layer | INT |
| Pizza_size_name | Size of product (pizza) | TEXT |
| Insert_dt | Timestamp of insertion of record | TIMESTAMP |
| Update_dt | Timestamp of update of record | TIMESTAMP |

Example with filled data



| | Row #1 |
|---|---|
| pizza_surr_id | 71 |
| pizza_src_id | 71 |
| source_system | BL_3NF |
| source_entity | CE_PIZZAS |
| pizza_name | SUPER SPICY |
| pizza_type_src_id | 3 |
| pizza_type_name | ORIGINAL |
| pizza_size_src_id | 6 |
| pizza_size_name | XXXL |
| insert_dt | 2024-08-12 17:04:51.203 |
| update_dt | 2024-08-12 17:04:51.203 |

DIM_DATE

Dimension to store date data in our DWH.

| Column name | Description | Data Type |
|---|---|---|
| Date_id | Unique identifier of a date | DATE |
| Date_year | Year | INT |
| Date_month | Month | INT |
| Date_monthname | Name of month | TEXT |
| Date_monthday | Day of month | INT |
| Date_yearday | Day of year | INT |
| Date_weekdayname | Day of week | TEXT |
| Date_calendarweek | Calendar week | INT |
| Date_formatteddate | Formatted date | TEXT |
| Date_quarter | Quarter date | TEXT |
| Date_yearquarter | Year quarter | TEXT |
| Date_yearmonth | Month of year | TEXT |
| Date_yearcalendarweek | Calendar week of year | TEXT |
| Date_weekend | Weekend | TEXT |
| Date_georgianholiday | National holidays of Georgia | TEXT |
| Date_period | Period | TEXT |
| Date_cwstart | CW start | DATE |
| Date_cwend | CW end | DATE |
| Date_monthstart | Start of month | DATE |
| Date_monthend | End of month | TIMESTAMP |

Example with filled data

| | Row #1 |
|---|---|
| date_id | 2025-12-31 |
| date_year | 2,025 |
| date_month | 12 |
| date_monthname | December |
| date_monthday | 31 |
| date_yearday | 365 |
| date_weekdayname | Wednesday |
| date_calendarweek | 1 |
| date_formatteddate | 31. 12. 2025 |
| date_quarter | Q4 |
| date_yearquarter | 2025/Q4 |
| date_yearmonth | 2025/12 |
| date_yearcalendarweek | 2026/01 |
| date_weekend | Weekday |
| date_georgianholiday | No holiday |
| date_period | New Year Season |
| date_cwstart | 2025-12-29 |
| date_cwend | 2026-01-04 |
| date_monthstart | 2025-12-01 |
| date_monthend | 2025-12-31 00:00:00.000 |

DIM_TIME

Dimension to store date time in our DWH.

This dimension is made for comfortable reporting by period in time.

Because this business requires reports also based on time, not only on date.

| Column name | Description | Data Type |
|---|---|---|
| Time_id | Unique identifier of a time | TEXT |
| Time_hourofday | Hour of day | INT |
| Time_quarterhour | Quarter hour | TEXT |
| Time_minuteofday | Minute of day | INT |
| Time_daytimename | Name of daytime | TEXT |
| Time_daynightname | Name of nighttime | TEXT |

Example with filled data

| | Row #1 |
|---|---|
| time_id | 00:00 |
| time_hourofday | 0 |
| time_quarterhour | 00:00 – 00:14 |
| time_minuteofday | 0 |
| time_daytimename | Night |
| time_daynightname | Night |

Step 4: Identify Facts.

FCT_SALES

As far as our process is selling pizza lets see what data should be present in our fact table.

| Column name | Description | Data Type |
|---|---|---|
| Customer_surr_id | FK to DIM_CUSTOMERS | BIGINT |
| Order_surr_id | FK to DIM_ORDERS | BIGINT |
| Pizza_surr_id | FK to DIM_PRODUCTS | BIGINT |
| Event_time | FK to DIM_TIMES | TEXT |
| Address_surr_id | FK to DIM_ADDRESSES | BIGINT |
| Event_dt | FK to DIM_DATES | DATE |
| Quantity | Quantity of same pizza in order | INT |
| Price | Price of the product in order | DECIMAL (6,2) |
| Fct_cost_order | Calculated price of same pizzas | DECIMAL (10,2) |
| Insert_dt | Timestamp of insertion of record | TIMESTAMP |
| Update_dt | Timestamp of update of record | TIMESTAMP |

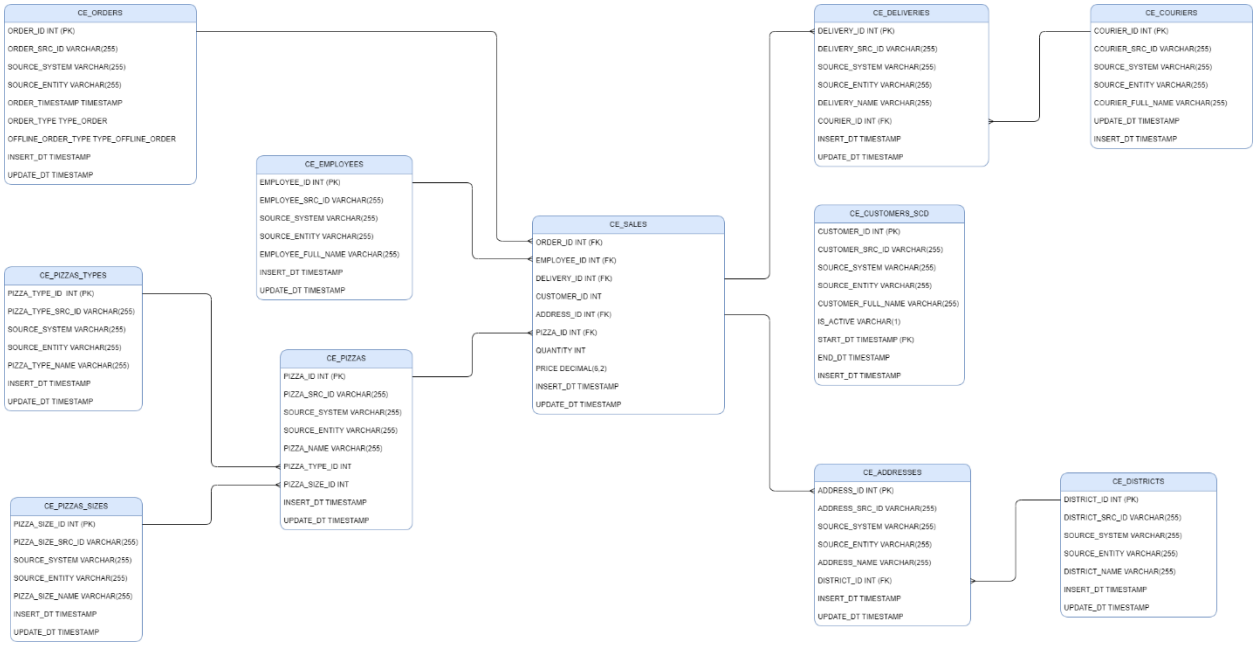
Example with filled data

| | Row #1 |
|---|---|
| customer_surr_id | 407 |
| order_surr_id | 409,400 |
| pizza_surr_id | 70 |
| event_time | N.A. |
| address_surr_id | -1 |
| event_dt | 1900-01-01 |
| quantity | 0 |
| price | 0 |
| fct_cost_order | 0 |
| insert_dt | 2024-08-12 17:04:51.203 |
| update_dt | 2024-08-12 17:04:51.203 |

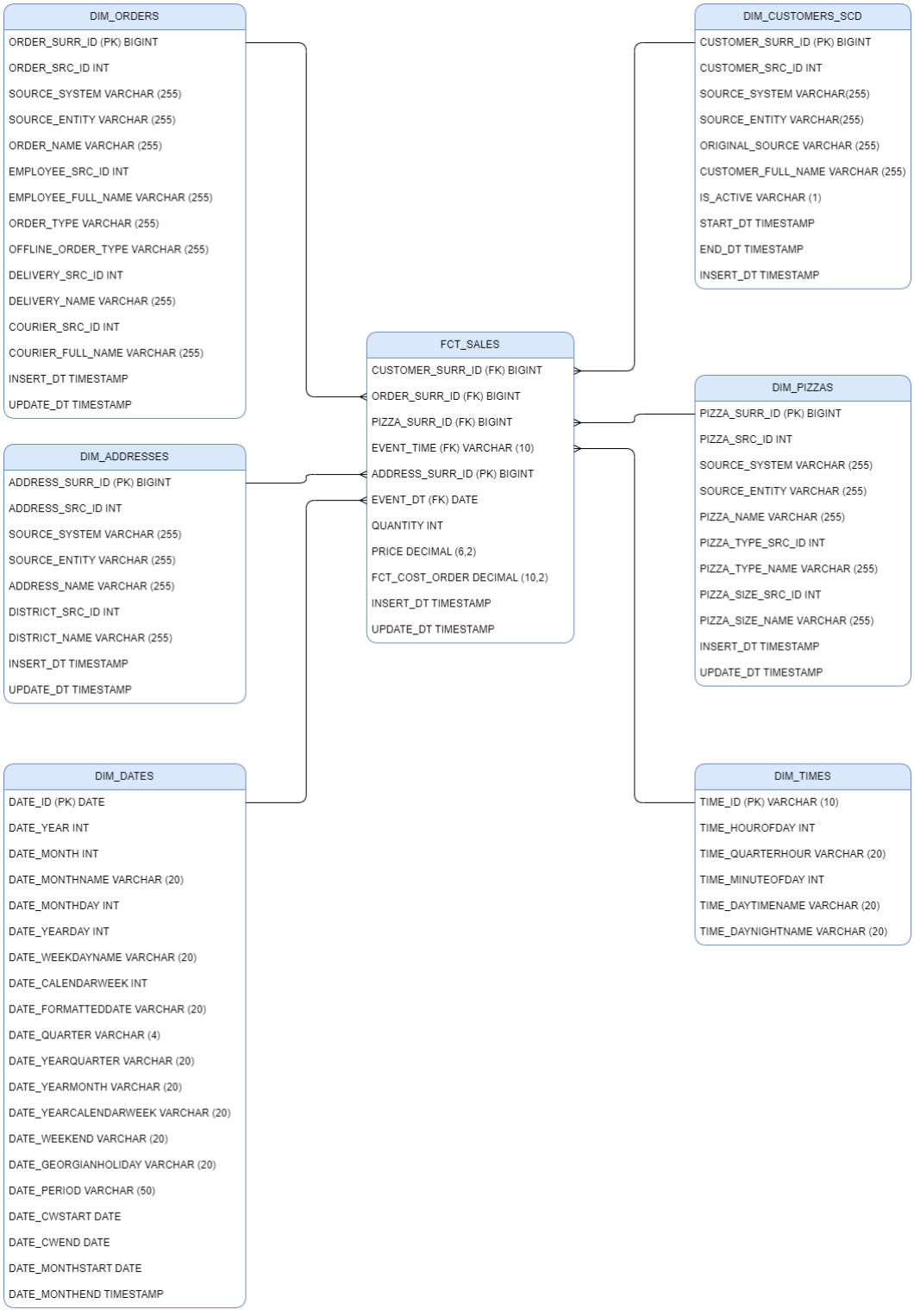
# 2 BUSINESS LAYER 3NF

Here we have performed 3NF schema layer conversion of our sources.

SCD’s to track history data here are: CE_CUSTOMERS_SCD.



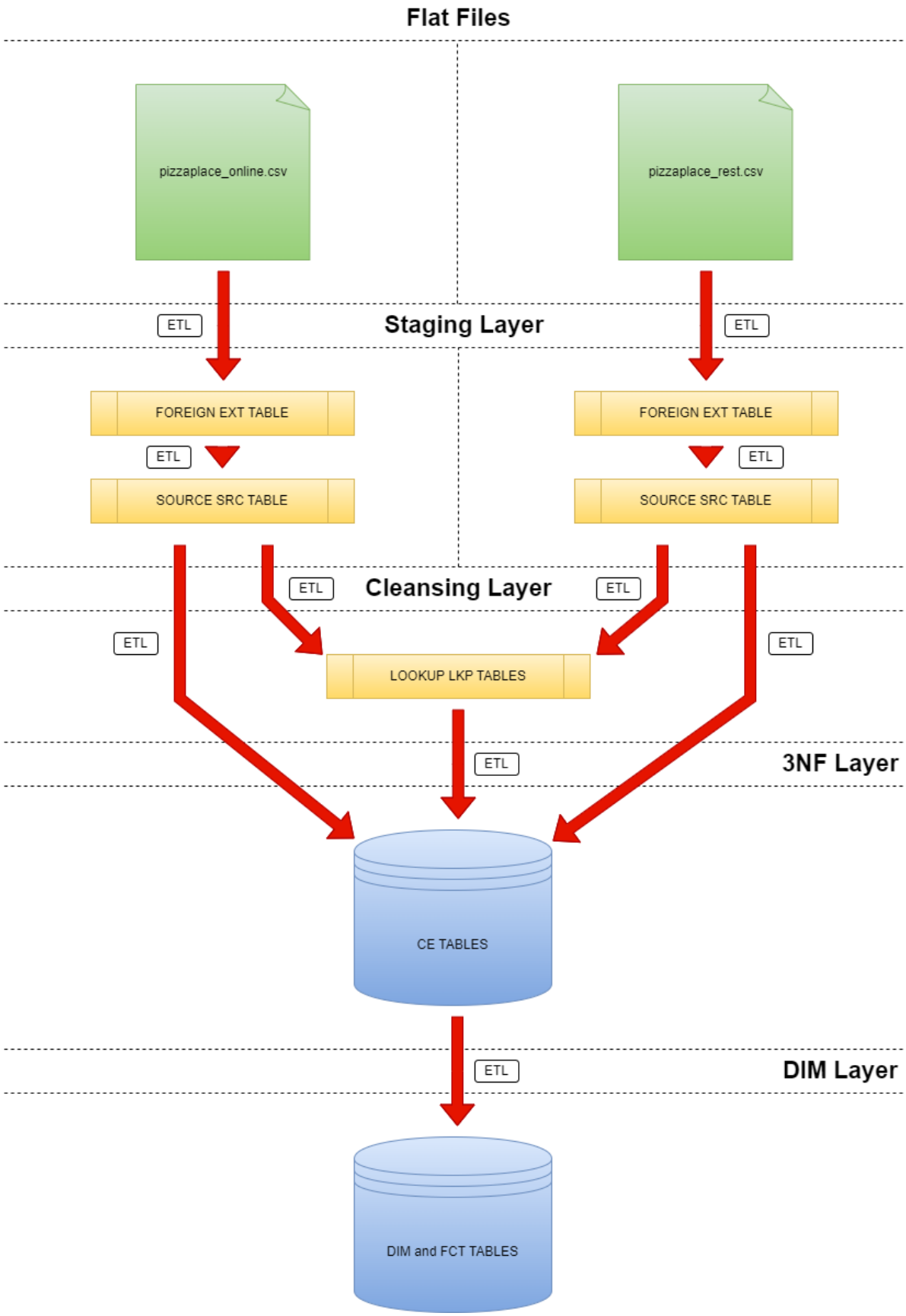
# 3 BUSINESS LAYER DIMENSIONAL MODEL



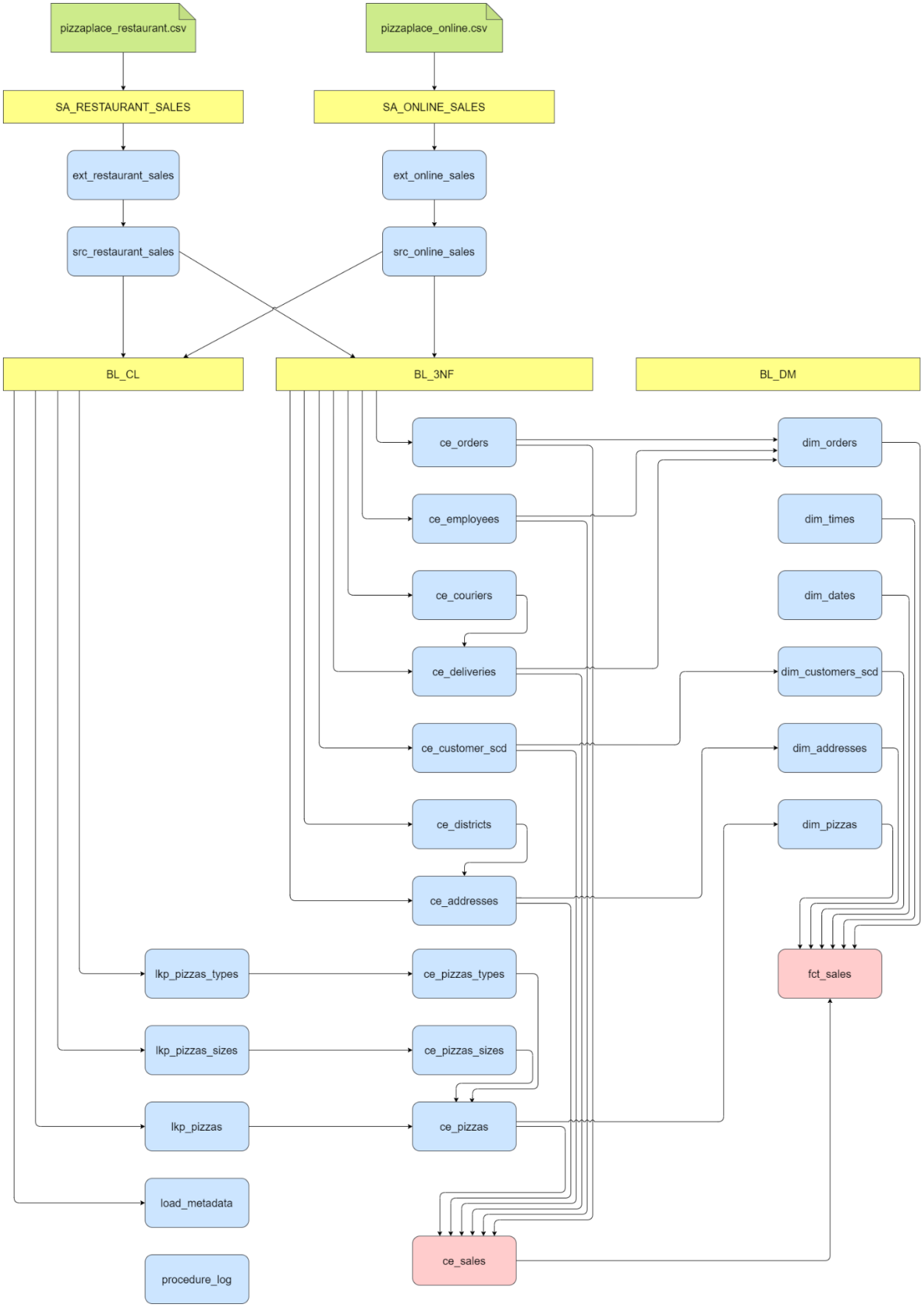

Metrics, presented in this model:

- QUANTITY – quantity of product sold,
- PRICE – price of a product piece,
- FCT_COST_ORDER – cost of a product sold (Price * Quantity).

# 4 LOGICAL SCHEME



# 5 DATA FLOW


pizzaplace_restaurant.csv
pizzaplace_online.csv
SA_RESTAURANT_SALES
SA_ONLINE_SALES
ext_restaurant_sales
ext_online_sales
src_restaurant_sales
src_online_sales
BL_CL
BL_3NF
BL_DM
ce_orders
ce_employees
ce_couriers
ce_deliveries
ce_customer_scd
ce_districts
ce_addresses
ce_pizzas_types
ce_pizzas_sizes
ce_pizzas
ce_sales
dim_orders
dim_times
dim_dates
dim_customers_scd
dim_addresses
dim_pizzas
fct_sales
lkp_pizzas_types
lkp_pizzas_sizes
lkp_pizzas
load_metadata
procedure_log

# 6 FACT TABLE PARTITIONING STRATEGY

Partitions for fact table are created based on window_partitions_procedure.sql.

The logic is: partitions created on the fly while loading new data, that has dates, partitions for which are not yet created.

Partitions are created for interval of 2 months.