A.N.T.

Chain of restaurants

Solution Concept



|  |  |
| --- | --- |
| **Submission Date** | *07/26/2013* |
| **Requested By** | *Kyril Bucha* |
| **Business Owner** | *Anton Tserakhau* |
| **Contact Info.** | *Anton\_Tserakhau@epam.com* |

**Contents**

[1 Overview 3](#_Toc362541140)

[1.1 Business Background 4](#_Toc362541141)

[1.2 Benefits 4](#_Toc362541142)

[2 Requirements 5](#_Toc362541143)

[2.1 Business Requirements 5](#_Toc362541144)

[2.2 Technical Requirements 5](#_Toc362541145)

[3 Solution Sketch 6](#_Toc362541146)

[3.1 Source Tables structure 6](#_Toc362541147)

[3.2 Summarize Data Plan 7](#_Toc362541148)

[4 DWH Solution Concept 8](#_Toc362541149)

[4.1 Logical Diagram 8](#_Toc362541150)

[4.2 Physical diagram 10](#_Toc362541151)

[4.3 Dimensions 10](#_Toc362541152)

[4.3.1 Dimension Types 10](#_Toc362541153)

[4.3.2 Dimension Hierarchies 11](#_Toc362541154)

[4.4 Facts 12](#_Toc362541155)

[4.4.1 Facts Aggregations 12](#_Toc362541156)

[4.5 Dataflow Diagram 13](#_Toc362541157)

[4.6 Partitioning rules 14](#_Toc362541158)

[4.7 Strategy of Parallel execution 15](#_Toc362541159)

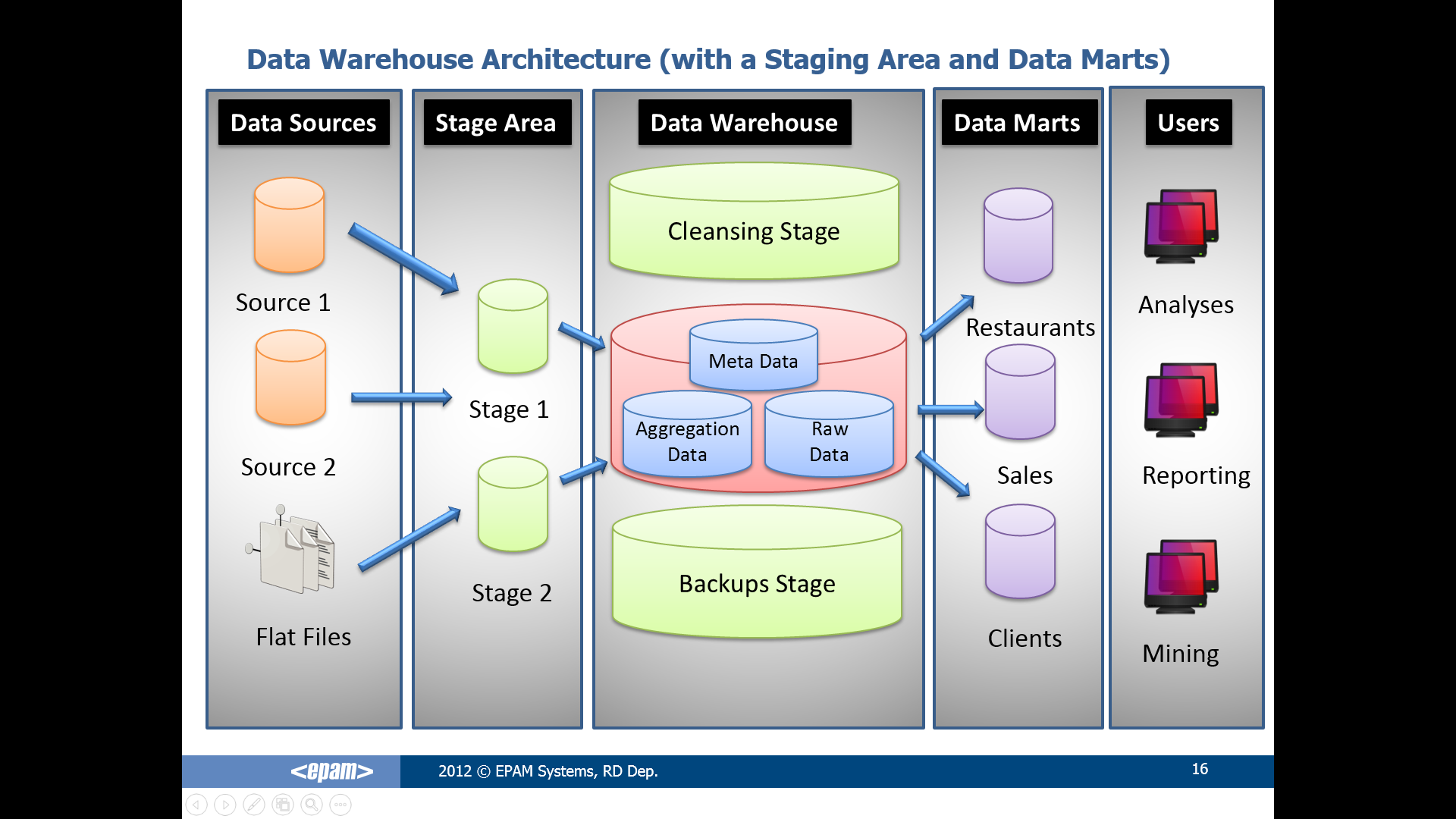
# Overview

Data Warehouse (DWH) is a database used for reporting and data analysis. It is a central repository of data which is created by integrating data from one or more disparate sources. Data warehouses store current as well as historical data and are used for creating trending reports for senior management reporting.

A data warehouse maintains a copy of information from the source transaction systems. This architectural complexity provides the opportunity to:

* Congregate data from multiple sources into a single database so a single query engine can be used to present data.
* Mitigate the problem of database isolation level lock contention in transaction processing systems caused by attempts to run large, long running, analysis queries in transaction processing databases.
* Maintain data history, even if the source transaction systems do not.
* Integrate data from multiple source systems, enabling a central view across the enterprise. This benefit is always valuable, but particularly so when the organization has grown by merger.
* Improve data quality, by providing consistent codes and descriptions, flagging or even fixing bad data.
* Present the organization's information consistently.
* Provide a single common data model for all data of interest regardless of the data's source.
* Restructure the data so that it makes sense to the business users.
* Restructure the data so that it delivers excellent query performance, even for complex analytic queries, without impacting the operational systems.
* Add value to operational business applications, notably customer relationship management (CRM) systems.

Data Warehouse architecture:



This document contains the concepts and principles of construction DWH for the restaurant chain "A.N.T.", which allows to solve business problems and responsible the specified requirements DWH.

## Business Background

Restaurant chain "A.N.T." provides catering services around the world. The chain has a lot of restaurants, so you need a powerful tool for accounting and analyze all data. Company has information about sales – time, restaurant and location, dishes.

## Benefits

Benefits of a data warehouse:

1. Storage of all data together and structured.
2. Quickly provision of necessary information.
3. Reporting on profits over the restaurant and the country / region / continent for the specified period.
4. Analysis of historical data and to provide the most profitable business solutions for specific problems:

* identifying the most profitable countries / regions / continents;
* determination of the best-selling dishes, depending on the period;
* determining the most profitable restaurants;
* determining the approximate income in future periods based on the previous one.

1. Data analysis and identification of the most problematic restaurants.
2. Data analysis and identification of discrepancies, i.e. any problems, based on the data for previous periods.

# Requirements

Requirements are divided into business requirements and technical requirements.

## Business Requirements

1. Daily/weekly/monthly information about all sales of dishes ordered in a specific country with additional information about business period.
2. Daily/weekly/monthly calculated amount and count of certain sales of dishes.
3. Daily/weekly/monthly calculated profit on certain restaurant.
4. Daily/weekly/monthly calculated count of sales all dishes on certain restaurant.

## Technical Requirements

1. The data should be stored on three servers (America, Europe and Asia).
2. The system should be updated every day at 00:00.
3. The average time receiving a report 30 seconds.
4. The statistic period is last 12 months.
5. The minimal period is one day.
6. The average count of transactions in one restaurant per day is 1000 and around 25 000 per month.
7. The system should made backup every month.
8. The system should have high information security.

# Solution Sketch

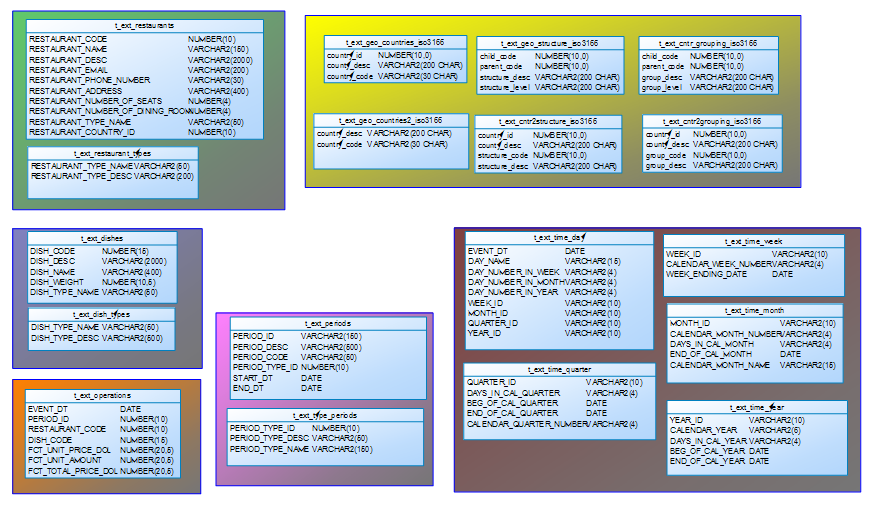
## Source Tables structure

For the correct creating of a system of analysis, at first, it is necessary to determine structure and content source data. Our system will receive data from other tables that were structured based on data received from the files.

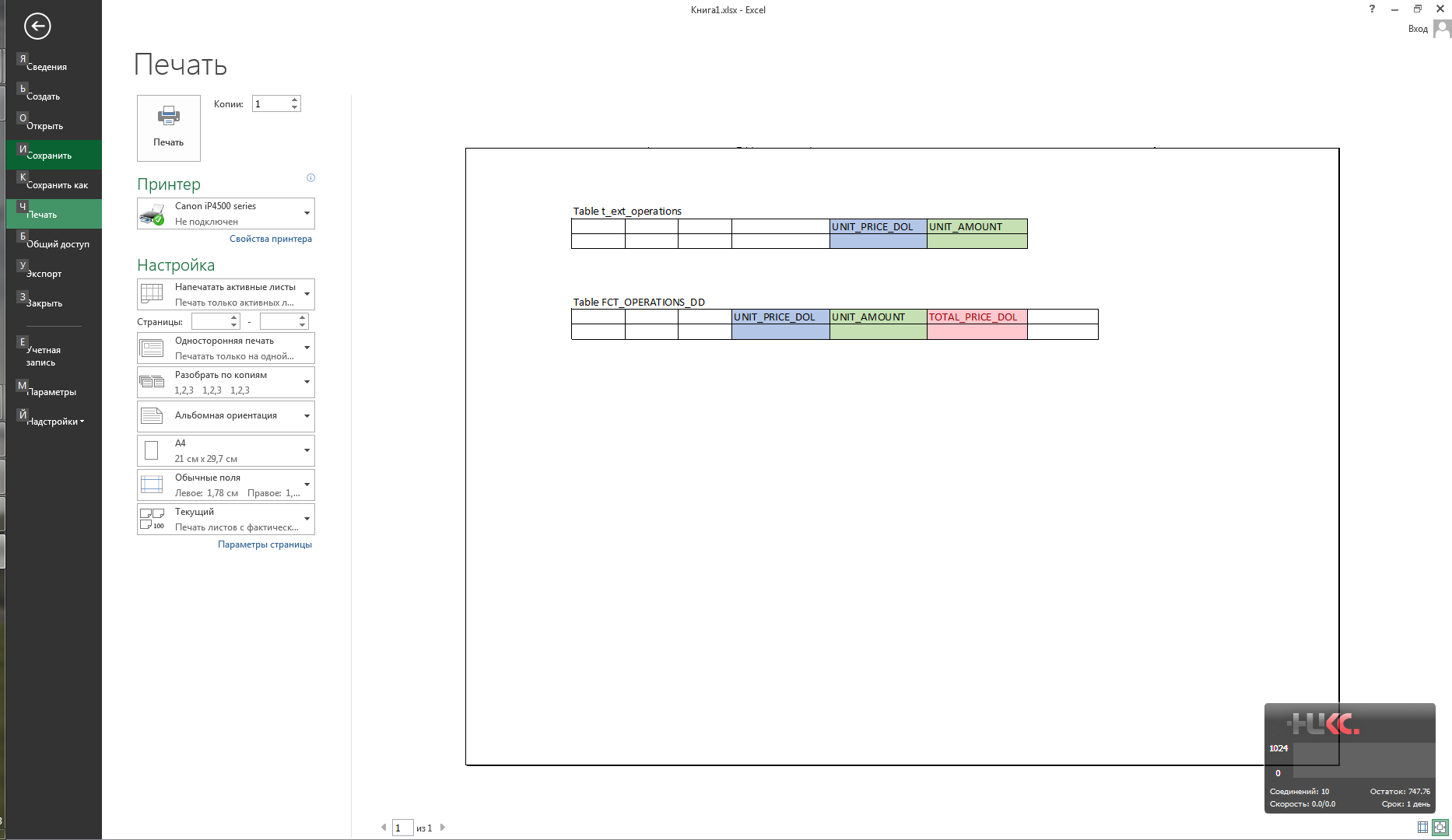
External files carry the following information:

* Information about restaurants (green);
* Information about dishes (blue);
* Information about operations-transactions (orange);
* Information about periods (magenta);
* Information about locations (yellow);
* Information about time (burgundy).

Logical and physical structure of source:



## Summarize Data Plan



Source

**TOTAL\_PRICE\_DOL**

=

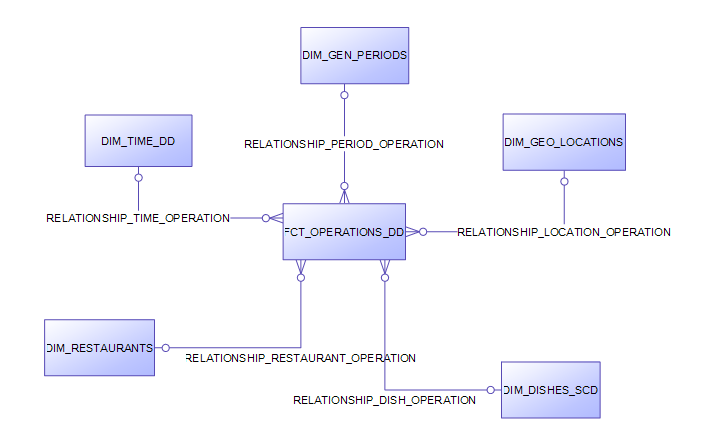
**UNIT\_PRICE\_DOL** x **UNIT\_AMOUNT**

It is additive measure.

# DWH Solution Concept

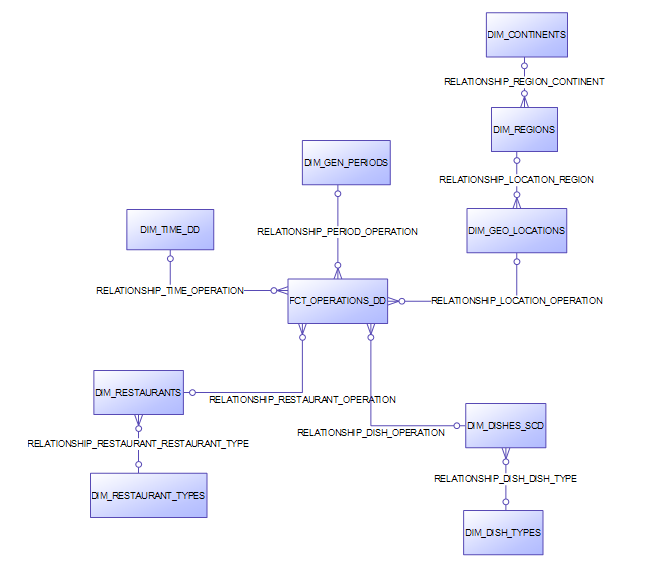
## Logical Diagram

Star-Scheme logical diagram:



|  |  |  |
| --- | --- | --- |
| Table name | Table type | Description |
| DIM\_TIME\_DD | Dimension | Information about time – list of event with graduation by days |
| DIM\_GEN\_PERIOD | Dimension | Information about periods – list of specified intervals |
| DIM\_GEO\_LOCATIONS | Dimension | Information about locations – list of all possible divisions with graduation by countries |
| DIM\_DISHES\_SCD | Dimension | Information about dishes – list of all dishes and their attributes |
| DIM\_RESTAURANTS | Dimension | Information about restaurants – list of all restaurants and their attributes |
| FCT\_OPERATIONS\_DD | Fact | Information about operations-transactions – list of all sales dishes in every restaurants by day |

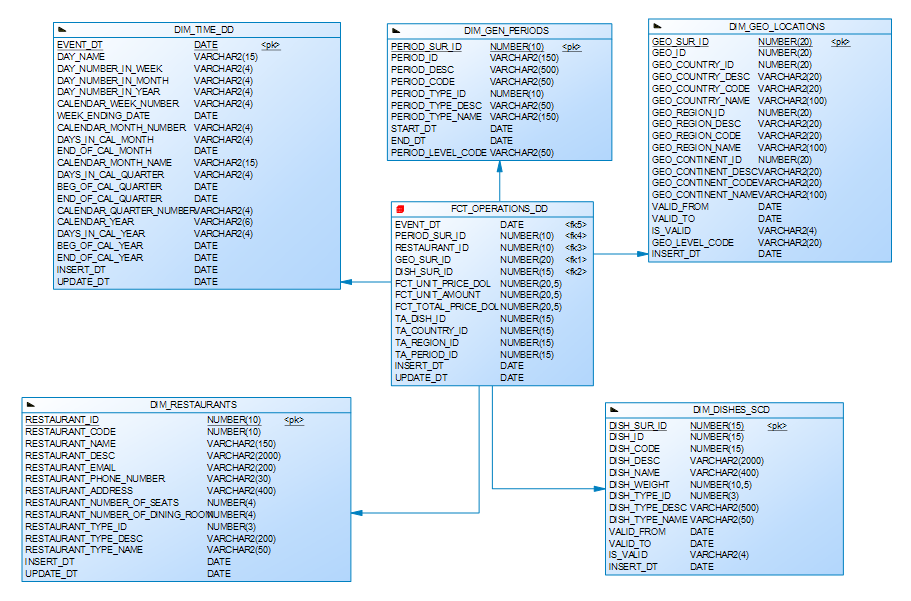
Snowflake logical diagram:



|  |  |  |
| --- | --- | --- |
| Table name | Table type | Description |
| DIM\_TIME\_DD | Dimension | Information about time – list of event with graduation by days |
| DIM\_GEN\_PERIOD | Dimension | Information about periods – list of specified intervals |
| DIM\_GEO\_LOCATIONS | Dimension | Information about locations – list of all possible divisions with graduation by countries |
| DIM\_REGIONS | Dimension | Information about regions – list of all regions and their codes |
| DIM\_CONTINENTS | Dimension | Information about continents – list of all continents and their codes |
| DIM\_DISHES\_SCD | Dimension | Information about dishes – list of all dishes and their attributes |
| DIM\_DISH\_TYPES | Dimension | Information about dish types – list of all possible dish types and their descriptions |
| DIM\_RESTAURANTS | Dimension | Information about restaurants – list of all restaurants and their attributes |
| DIM\_RESTAURANT\_TYPES | Dimension | Information about restaurant types – list of all possible restaurant types and their descriptions |
| FCT\_OPERATIONS\_DD | Fact | Information about operations-transactions – list of all sales dishes in every restaurants by day |

## Physical diagram

Star-Scheme physical diagram:



## Dimensions

A dimension table is a table in a star schema of a data warehouse. A dimension table stores attributes, or dimensions, that describe the objects in a fact table.

In data warehousing, a dimension is a collection of reference information about a measurable event. These events are known as facts and are stored in a fact table. Dimensions categorize and describe data warehouse facts and measures in ways that support meaningful answers to business questions. They form the very core of dimensional modeling.

### Dimension Types

| NAME | TYPE | SIZE | MERGED DIMENSIONS | DESCRIPTION |
| --- | --- | --- | --- | --- |
| DIM\_TIME\_DD | SCD1 | BIG | DW.T\_DAYS  DW.LC\_DAYS  DW.T\_WEEKS  DW.T\_MONTHS  DW.LC\_MONTHS  DW.T\_QUARTERS  DW.T\_YEARS  DW.T\_LOCALIZATIONS | Dimension table with list of time parameters. |
| DIM\_GEO\_LOCATIONS | SCD2 | SMALL | DW.T\_COUNTRIES  DW.LC\_COUNTRIES  DW.T\_CNTR\_GROUPS  DW.LC\_CNTR\_GROUPS  DW.T\_GEO\_REGIONS  DW.LC\_GEO\_REGIONS  DW.T\_GEO\_OBJECTS  DW.T\_GEO\_TYPES  DW.T\_GEO\_OBJECT\_LINKS  DW.T\_LOCALIZATIONS  DW.T\_GEO\_ACTION | Dimension table with full geo information - countries, regions, subregions and some other classifications. |
| DIM\_GEN\_PERIODS | SCD1 | SMALL | DW.GEN\_PERIODS  DW.LC\_GEN\_PERIODS | Dimension table with information for different periods. |
| DIM\_DISHES\_SCD | SCD2 | SMALL | DW.T\_DISHES  DW.LC\_DISHES  DW.T\_DISH\_TYPES  DW.LC\_DISH\_TYPES | Dimension with information about dishes and dish types. It has the history of changing names and attributes. |
| DIM\_RESTAURANTS | SCD1 | SMALL | DW.T\_RESTAURANTS  DW.LC\_ RESTAURANTS  DW.T\_ RESTAURANTS\_TYPE  DW.LC\_ RESTAURANT\_TYPES | Dimension with information about restaurants and restaurant types. |

### Dimension Hierarchies

**DIM\_TIME\_DD:**

Hierarchy DAY-WEEK-MONTH-QUARTER-YEAR

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | LEVEL\_CODE | LEVEL\_DESC | LEVEL\_NATURAL\_KEY |
| DAY | DAY | Store all days at the calendar period | DAY\_ID |
| WEEK | WEEK | Store all weeks at the calendar period | WEEK\_ID |
| MONTH | MONTH | Store all months at the calendar period | MONTH\_ID |
| QUARTER | QUARTER | Store all quarters at the calendar period | QUARTER\_ID |
| YEAR | YEAR | Store years at the calendar period | YEAR\_ID |

**DIM\_GEO\_LOCATIONS:**

Hierarchy COUNTRY –REGION -- CONTINET

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | LEVEL\_CODE | LEVEL\_DESC | LEVEL\_NATURAL\_KEY |
| COUNTRIES | COUNTRY | Store all country name in region | GEO\_COUNTRY\_ID |
| REGIONS | REGION | Store all regions name in continent | GEO\_REGION\_ID |
| CONTINENTS | CONTINENT | Store all continents name | GEO\_CONTINENT \_ID |

**DIM\_DISHES\_SCD:**

Hierarchy DISH – DISH TYPE

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | LEVEL\_CODE | LEVEL\_DESC | LEVEL\_NATURAL\_KEY |
| DISH | DISH | Store all dishes in type | DISH\_ID |
| DISH\_TYPE | DISH\_TYPE | Store all dish types | DISH\_TYPE\_ID |

## Facts

A fact table is the central table in a star schema of a data warehouse. A fact table stores quantitative information for analysis.

A fact table works with dimension tables. A fact table holds the data to be analyzed, and a dimension table stores data about the ways in which the data in the fact table can be analyzed. Thus, the fact table consists of two types of columns. The foreign keys column allows joins with dimension tables, and the measures columns contain the data that is being analyzed.

### Facts Aggregations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | CODE | TABLE NAME | ADDITIVE | DESCRIPTION |
| Amount of dishes sold | FCT\_UNIT\_AMOUNT | FCT\_OPERATIONS\_DD | + | Calculating amount of dishes sold |
| Total price of dishes sold | FCT\_TOTAL\_PRICE\_DOL | FCT\_OPERATIONS\_DD | + | Calculating total price of dishes sold |

## Dataflow Diagram



## Partitioning rules

To good maintenance, table FCT\_OPERATIONS\_DD should be partitioned in several tablespaces. It’s should be range partitions by EVENT\_DT column by month:

CREATE TABLE "U\_DW".FCT\_OPERATIONS\_DD

(

EVENT\_DT DATE,

PERIOD\_SUR\_ID NUMBER(10),

RESTAURANT\_ID NUMBER(10),

GEO\_SUR\_ID NUMBER(20),

DISH\_SUR\_ID NUMBER(15),

FCT\_UNIT\_PRICE\_DOL NUMBER(20,5) NOT NULL,

FCT\_UNIT\_AMOUNT NUMBER(20,5) NOT NULL,

FCT\_TOTAL\_PRICE\_DOL NUMBER(20,5) NOT NULL,

TA\_DISH\_ID NUMBER(15),

TA\_COUNTRY\_ID NUMBER(15),

TA\_REGION\_ID NUMBER(15),

TA\_PERIOD\_ID NUMBER(15),

INSERT\_DT DATE NOT NULL,

UPDATE\_DT DATE

)

PARTITION BY RANGE

(EVENT\_DT)

(

PARTITION

PART\_MONTH\_1

VALUES LESS THAN (TO\_DATE('01-FEB-2012','DD-MON-RR'))

TABLESPACE TS\_DATA\_MONTH\_1

NOCOMPRESS,

PARTITION

PART\_MONTH\_2

VALUES LESS THAN (TO\_DATE('01-MAR-2012','DD-MON-RR'))

TABLESPACE TS\_DATA\_MONTH\_2

NOCOMPRESS,

PARTITION

PART\_MONTH\_3

VALUES LESS THAN (TO\_DATE('01-APR-2012','DD-MON-RR'))

TABLESPACE TS\_DATA\_MONTH\_3

NOCOMPRESS,

PARTITION

PART\_MONTH\_4

VALUES LESS THAN (TO\_DATE('01-MAY-2012','DD-MON-RR'))

TABLESPACE TS\_DATA\_MONTH\_4

NOCOMPRESS,

PARTITION

PART\_MONTH\_5

VALUES LESS THAN (TO\_DATE('01-JUN-2012','DD-MON-RR'))

TABLESPACE TS\_DATA\_MONTH\_5

NOCOMPRESS,

PARTITION

PART\_MONTH\_6

VALUES LESS THAN (TO\_DATE('01-JUL-2012','DD-MON-RR'))

TABLESPACE TS\_DATA\_MONTH\_6

NOCOMPRESS,

PARTITION

PART\_MONTH\_7

VALUES LESS THAN (TO\_DATE('01-AUG-2012','DD-MON-RR'))

TABLESPACE TS\_DATA\_MONTH\_7

NOCOMPRESS,

PARTITION

PART\_MONTH\_8

VALUES LESS THAN (TO\_DATE('01-SEP-2012','DD-MON-RR'))

TABLESPACE TS\_DATA\_MONTH\_8

NOCOMPRESS,

PARTITION

PART\_MONTH\_9

VALUES LESS THAN (TO\_DATE('01-OCT-2012','DD-MON-RR'))

TABLESPACE TS\_DATA\_MONTH\_9

NOCOMPRESS,

PARTITION

PART\_MONTH\_10

VALUES LESS THAN (TO\_DATE('01-NOV-2012','DD-MON-RR'))

TABLESPACE TS\_DATA\_MONTH\_10

NOCOMPRESS,

PARTITION

PART\_MONTH\_11

VALUES LESS THAN (TO\_DATE('01-DEC-2012','DD-MON-RR'))

TABLESPACE TS\_DATA\_MONTH\_11

NOCOMPRESS,

PARTITION

PART\_MONTH\_12

VALUES LESS THAN (TO\_DATE('01-JAN-2013','DD-MON-RR'))

TABLESPACE TS\_DATA\_MONTH\_12

NOCOMPRESS

);

## Strategy of Parallel execution

1. Parallel query should be used to perform SELECT operation to prevent a long waiting of a response from the database
2. Using parallel in DDL operations (create tables as select, create index, alter index rebuild) saving a lot of time during updating structure of our DWH and star levels because in this case OS using more hardware resources to execute it, but not DB.
3. Using parallel in DML operations (update, insert etc.) can help to save a lot of time to update FACTs and DIMMENSIONs tables.