

# Solution Concept Document

## Business Project

Shkrabatouskaya Vera

[https://github.com/VeraShkrabatouskaya/DataMola\\_Data-Camping-2022](https://github.com/VeraShkrabatouskaya/DataMola_Data-Camping-2022)

### **Solution concept – Business background**

#### Overview

Data Warehouse Case Study (by Starcom)

#### Business Background

Starcom is a world-renowned media communications agency that architects connected human experiences to create value through precision marketing, content and technology solutions. With more than 5,000 employees worldwide, Starcom partners with the world's leading marketers and new establishment brands, including Airbnb, Bank of America, Kellogg Company, Kraft Heinz, Novartis, Samsung, Visa and more. Starcom is part of Publicis Media—the global media solutions group which also encompasses Zenith, Mediavest | Spark and Optimedia | Blue 449—a key division of Publicis Groupe, one of the world's leading communications groups.

Starcom harness the power of media, technology and data to create experiences people love and actions brands need. Data guides us to opportunity, helping us uncover what people want, need and expect. Technology makes personalization and relevance possible. Creativity makes it magic.

The company operates with a large amount of information and is therefore faced with the problem of data quality and lack of historical data in the right breakdowns.

Information from external and internal sources has a different structure and data is often duplicated or contradictory. The implementation of a Corporate Data Warehouse solves these problems. Data Warehouses accumulate cleansed and structured information about a company's business in a single source for external and internal users: management, employees and customers.

The problems that faced by Starcom is the agency personnel are increasingly facing a growing requirement to harness and leverage information so, it requires the ability to combine data virtually or in data warehouses with business intelligence analytic.

#### Benefit

We propose that Starcom create a data warehouse project:

- To structure this data and use it to help make time-sensitive decision, inform strategy and anticipate outcomes.
- To help it derive maximum potential from the mass of unstructured data that continues to grow each other.
- To have the potential to quickly generate a single view of all multi-channel retail information and convert big data into usable information.

## Requirements

### Business Requirements

ID	Description
BR-01	Calculation of revenue information (Gross, Net) on a daily, weekly, monthly, quarterly, annual basis by agency, department, client, brand, promotion.
BR-02	Calculation of cost information (Gross, Net) on a daily, weekly, monthly, quarterly, annual basis by agency, department, client, brand, promotion.
BR-03	Calculation of profit information (Gross, Net) on a daily, weekly, monthly, quarterly, annual basis by agency, department, client, brand, promotion.
BR-04	Calculation of employee salary information (Gross, Net) on a monthly, quarterly, annual basis.
BR-05	Calculation of information about the number of promotions by brands.
BR-06	Calculation of information on the number of clients served by the agency and the number of employees of the network agencies based on their location for the management of the head office.

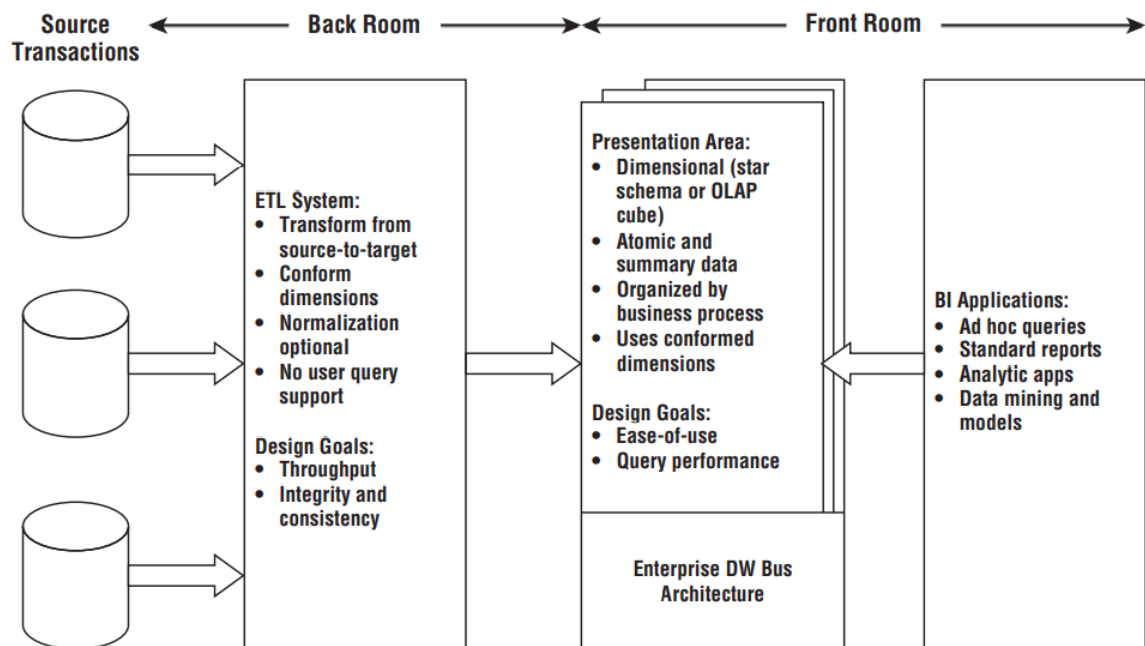
### Technical Requirements

ID	Description
TR-01	Fast, round-the-clock access to data
TR-02	Automatic support of data consistency;
TR-03	Providing the ability to work with data slices
TR-04	Ensure data integrity and reliability
TR-05	Protect large volumes of data

## Solution Sketch

### Source Tables structure

We will use the Kimball DW/BI architecture model as a basis.



We will suggest that the customer consider 2 types of DWH: star and snowflake.

### Summarize Data Plan

Expected result:

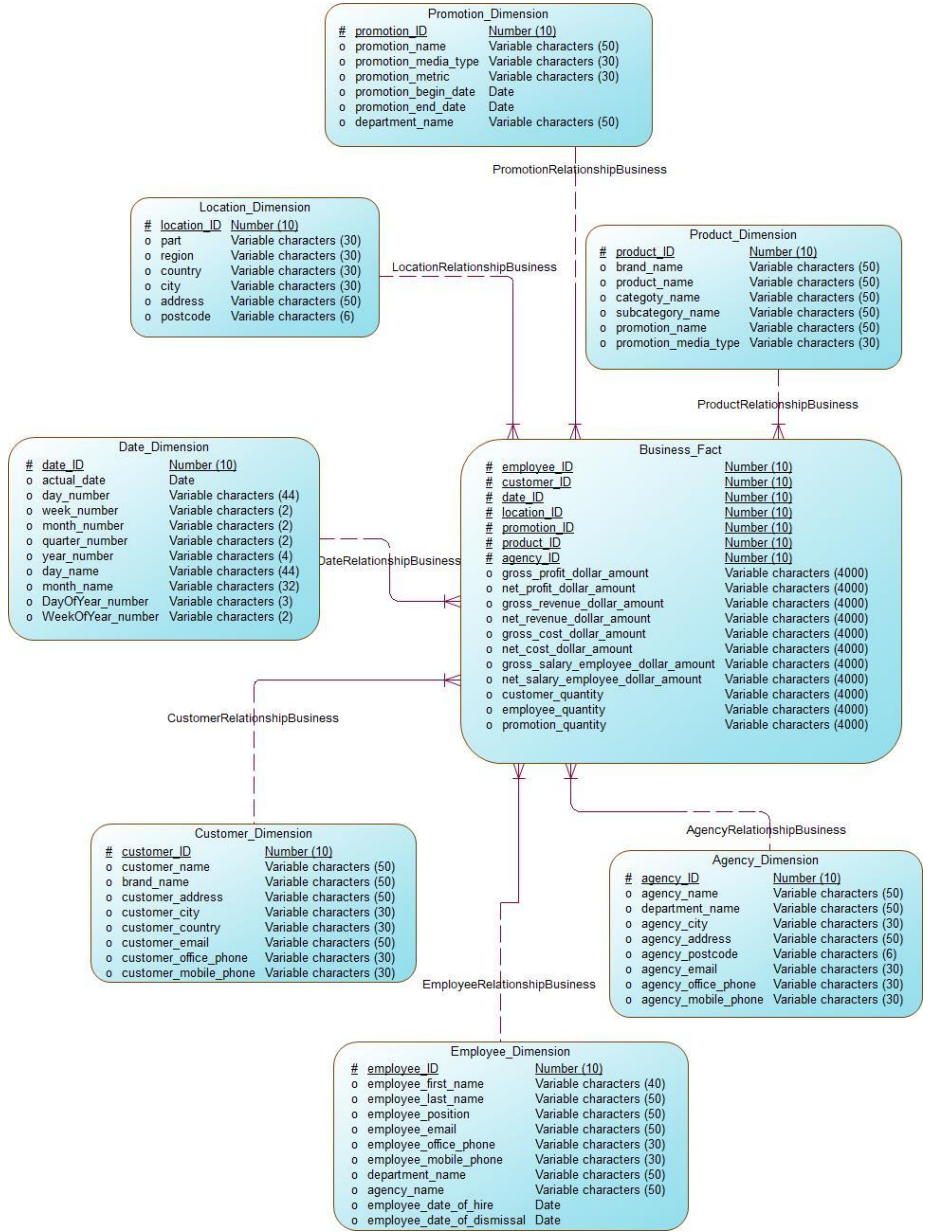
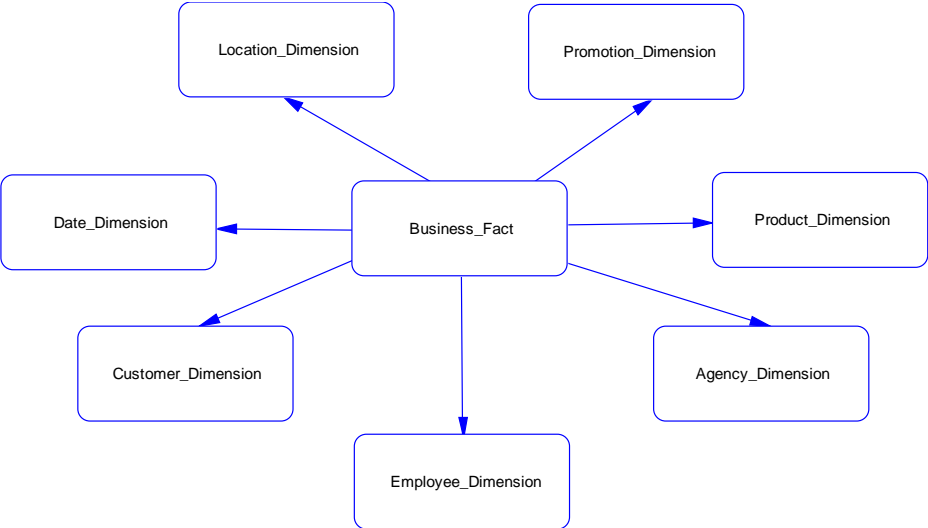
- A system will be designed and developed for Starcom DWH that can upload, analyze and import data from any data source.
- Any employee can link his data source to a Data Warehouse which will lead to quickly sharing important information within the company.

Working closely with representatives from the business and other DW/BI team members, the Data Warehouse Design Specialist developed a fact sheet with a grain of reporting for the customer, agency management and agency employees for each day.

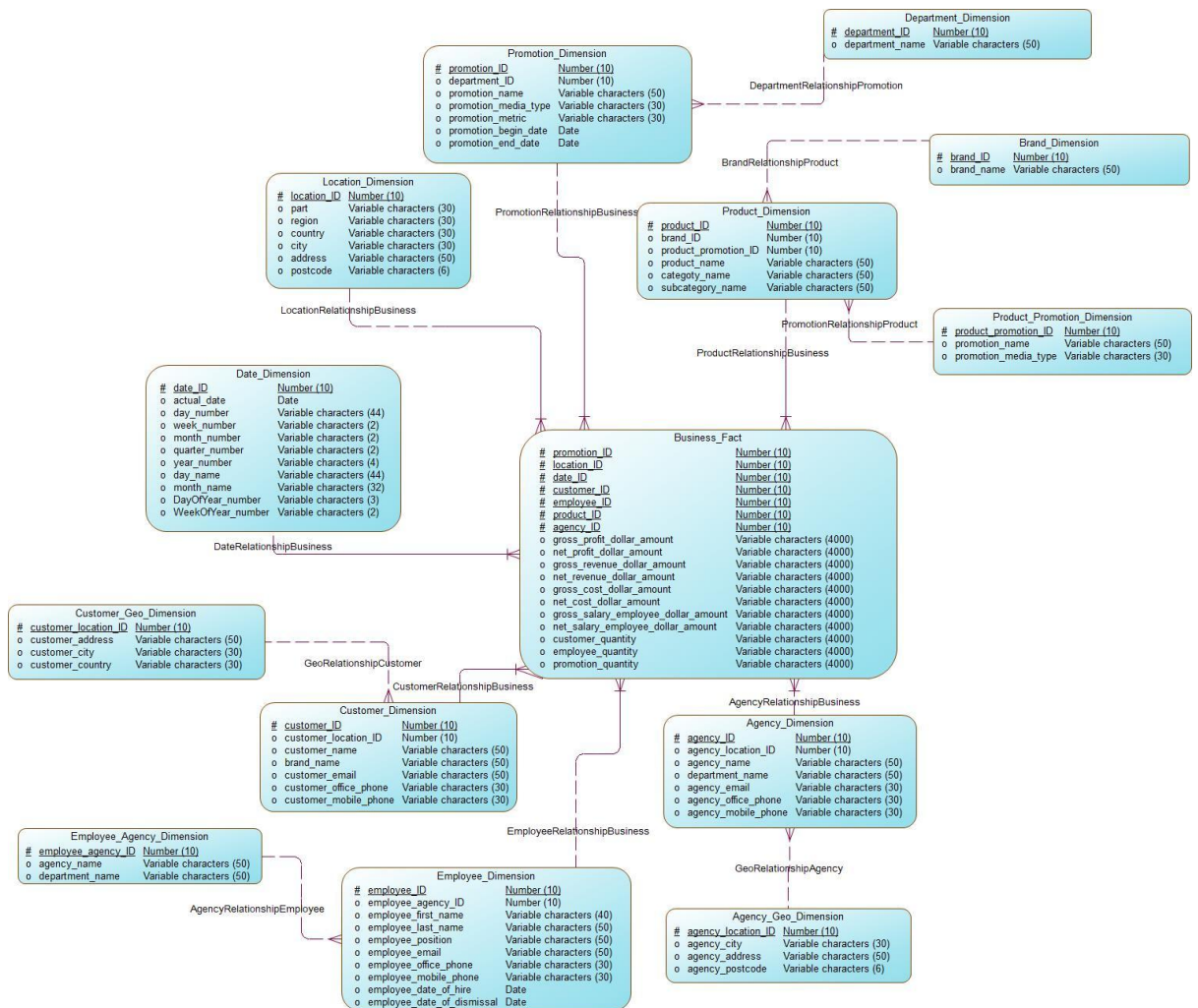
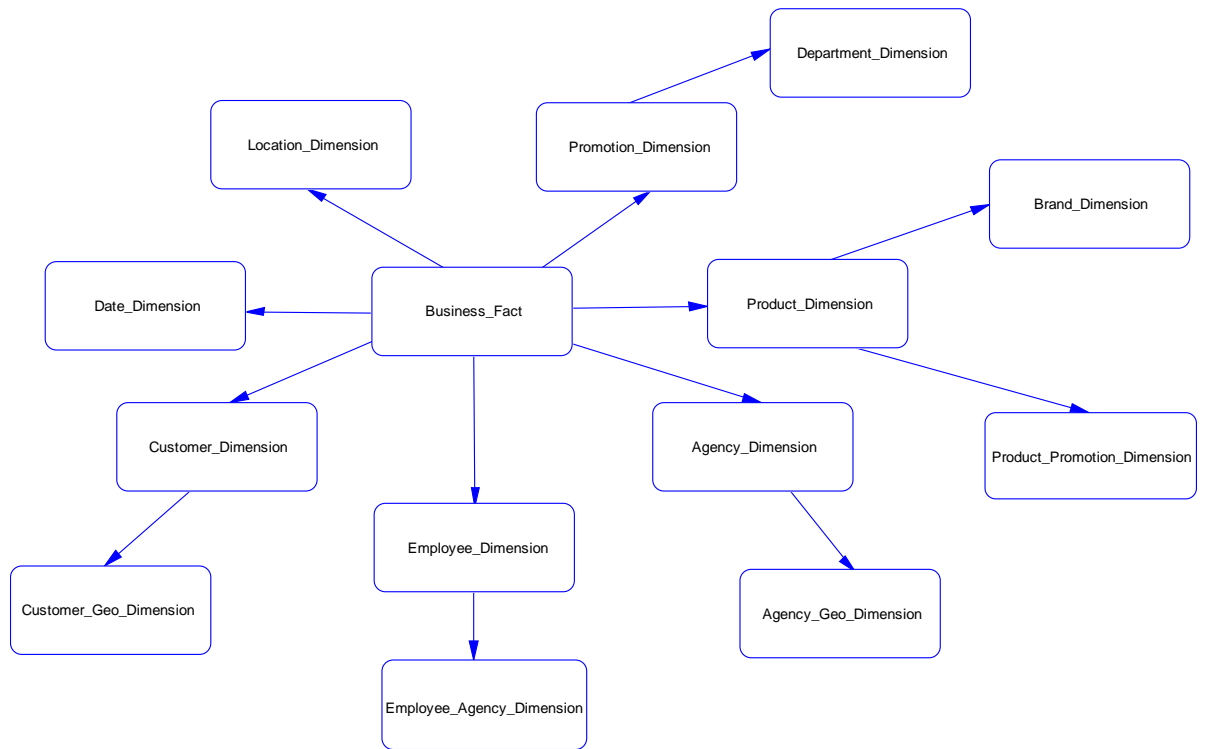
Based on the business problem, we needed to build a data warehouse to calculate gross/net cost and profit, employee salaries, number of clients, employees and promotion project. All fact table metrics go through a data aggregation procedure.

For the database, English was chosen for all locations in the agency's network. All monetary data is given in dollars.

Star Scheme:



### Snowflake Scheme:



As we can see, the main difference between the two relational database models is normalization. The dimension tables in the star schema are not normalized, which means that the business model will use relatively more space to store the dimension tables, and more space means more redundant records, which will eventually lead to inconsistency. The snowflake scheme, on the other hand, minimizes data redundancy because the measurement tables are normalized, which explains the much smaller number of redundant records. The business hierarchy and its dimensions are preserved through referential integrity, which means that links can be updated independently in the data stores.

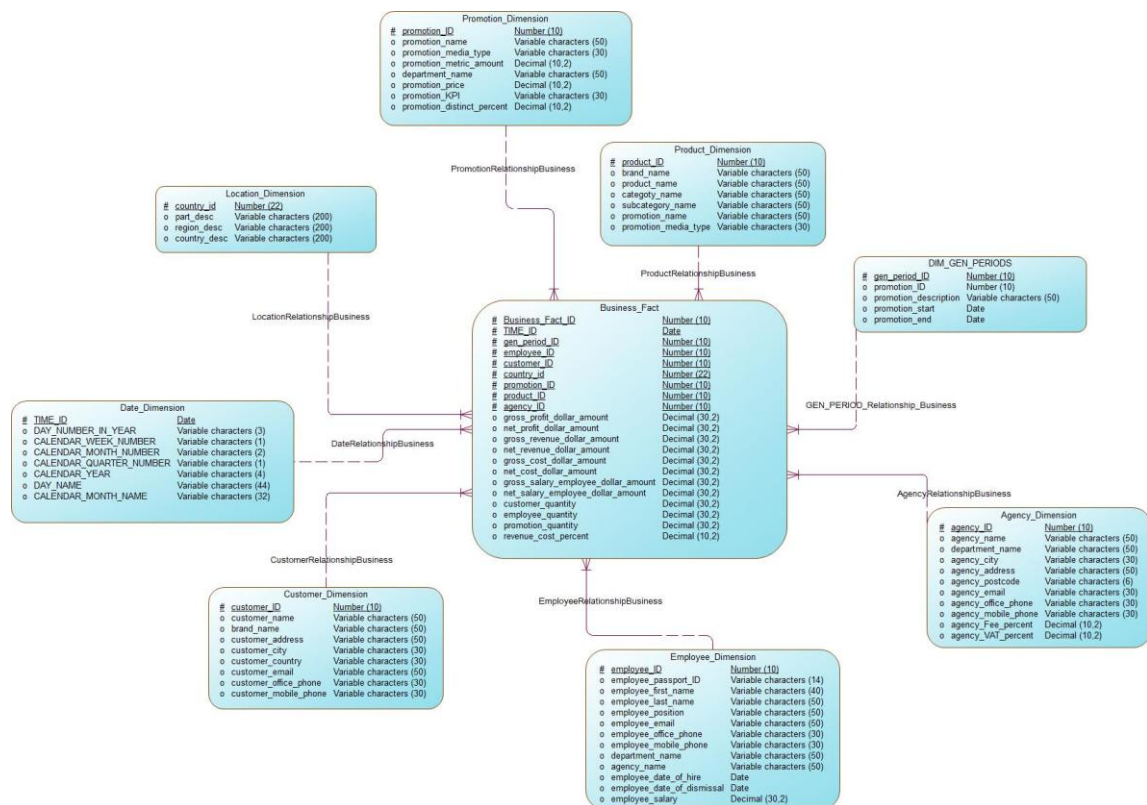
The star schema has fewer links between the dimension table and the fact table compared to the snowflake schema, which has many links, which explains the lower query complexity. Because measurements in a star type scheme are linked through a central fact table, it has clear connection paths, which means fast query response time, and fast response time means better performance. The snowflake scheme uses more connections, so the query response time increases, resulting in more complex queries, which ultimately reduces performance.

The choice between these two models must be made based on the customer's business objectives.

Based on the results of the business analysis, the customer approved the Star data warehouse model design.

### History of changes

Name	Date	Reason for change	Version
Vera Shkrabatouskaya	7/25/2022	Creation of a STAR data warehouse model.	1.0 draft 1
Vera Shkrabatouskaya	7/26/2022	Changes after approval. Added details to the Business Model.	1.0 approved



Summary table to describe all future STAR Dimensions:

Name	Type	Size	DW – Merged Dimensions	Descriptions
Location_Dimension	SCD1	SMALL	country_id part_desc region_desc country_desc	This table contains information on all countries, parts of the world, regions.
Date_Dimension	SCD1	BIG	TIME_ID DAY_NUMBER_IN_YEAR CALENDAR_WEEK_NUMBER CALENDAR_MONTH_NUMBER CALENDAR_QUARTER_NUMBER CALENDAR_YEAR DAY_NAME CALENDAR_MONTH_NAME	This table contains information about day, week, month, quarter and year numbers, day and month names and the current date TIME_ID.
Customer_Dimension	SCD1	SMALL	customer_ID customer_name brand_name customer_address customer_city customer_country customer_email customer_office_phone customer_mobile_phone	This table contains detailed information about the agency's clients (including name, client brands, office locations, emails and phone numbers).
Employee_Dimension	SCD2	BIG	employee_ID employee_passport_ID employee_first_name employee_last_name employee_position employee_email employee_office_phone employee_mobile_phone department_name agency_name employee_date_of_hire employee_date_of_dismissal employee_salary	This table provides information on agency staff (including heads of offices, as well as to the length of time the employee has been with the agency).
Agency_Dimension	SCD1	SMALL	agency_ID agency_name department_name agency_city agency_address agency_postcode agency_email agency_office_phone agency_mobile_phone agency_Fee_percent agency_VAT_percent	This table contains detailed information about the agencies (including agency name, department, office location, email, telephone numbers, fee information and the VAT the agencies work with).

DIM_GEN_PERIODS	SCD2	BIG	gen_period_ID promotion_ID promotion_description promotion_start promotion_end	This table allows facts to be grouped according to the duration of the advertising campaign.
Product_Dimension	SCD1	SMALL	product_ID brand_name product_name category_name subcategory_name promotion_name promotion_media_type	This table contains detailed information about the products advertised (including categories and subcategories, as well as the name of the advertising campaign and type of purchase method).
Promotion_Dimension	SCD1	BIG	promotion_ID promotion_name promotion_media_type promotion_metric_amount department_name promotion_price promotion_KPI promotion_distinct_percent	This table provides information on advertising projects (including information on prices, discounts, number of metrics and KPIs).

Summary table to describe all future STAR Dimensions Hierarchies:

**Date Dimension:**

**Hierarchy DAY-WEEK-MONTH-YEAR**

Name	LEVEL_CODE	LEVEL_DESC	LEVEL_NATURAL_KEY
DAY	DAY_NUMBER_IN_YEAR	Store the number of the day in the year	DAY_NUMBER_IN_YEAR
WEEK	CALENDAR_WEEK_NUMBER	Store week number in month	CALENDAR_WEEK_NUMBER
MONTH	CALENDAR_MONTH_NUMBER	Store the number of the month in the year	CALENDAR_MONTH_NUMBER
YEAR	CALENDAR_YEAR	Store all years	CALENDAR_YEAR

**Hierarchy WEEK-MONTH-QUARTER-YEAR**

Name	LEVEL_CODE	LEVEL_DESC	LEVEL_NATURAL_KEY
WEEK	CALENDAR_WEEK_NUMBER	Store week number in month	CALENDAR_WEEK_NUMBER
MONTH	CALENDAR_MONTH_NUMBER	Store the number of the month in the year	CALENDAR_MONTH_NUMBER
QUARTER	CALENDAR_QUARTER_NUMBER	Store the number of quarters in the year	CALENDAR_QUARTER_NUMBER
YEAR	CALENDAR_YEAR	Store all years	CALENDAR_YEAR



**Hierarchy DAY-MONTH-QUARTER-YEAR**

Name	LEVEL_CODE	LEVEL_DESC	LEVEL_NATURAL_KEY
DAY	DAY_NUMBER_IN_YEAR	Store the number of the day in the year	DAY_NUMBER_IN_YEAR
MONTH	CALENDAR_MONTH_NUMBER	Store the number of the month in the year	CALENDAR_MONTH_NUMBER
QUARTER	CALENDAR_QUARTER_NUMBER	Store the number of quarters in the year	CALENDAR_QUARTER_NUMBER
YEAR	CALENDAR_YEAR	Store all years	CALENDAR_YEAR

If you change the hierarchy in dates from lower to higher, the data will not be uploaded correctly.  
We therefore recommend using the classic hierarchical order.

**Location Dimension:****Hierarchy PART-REGION-COUNTRY**

Name	LEVEL_CODE	LEVEL_DESC	LEVEL_NATURAL_KEY
PART	part_desc	Store all parts of the world	part_desc
REGION	region_desc	Save all regions of the parts of the world	region_desc
COUNTRY	county_desc	Store all countries for each region.	county_desc

**DIM GEN PERIODS:****Hierarchy PROMOTION\_ID-PROMOTION\_START-PROMOTION\_END-PROMOTION\_DESCRIPTION**

Name	LEVEL_CODE	LEVEL_DESC	LEVEL_NATURAL_KEY
PROMOTION_ID	gen_period_ID	Store all ID numbers of promotional projects by NK promotion_ID	promotion_ID
PROMOTION_START	promotion_start	Store all start dates of promotional projects	promotion_start
PROMOTION_END	promotion_end	Store all end dates of promotional projects	promotion_end
PROMOTION_DESCRIPTION	promotion_description	Store all descriptions of promotional projects	promotion_description

**Employee Dimension:**

**Hierarchy** EMPLOYEE\_ID–EMPLOYEE\_FIRST\_NAME–EMPLOYEE\_LAST\_NAME–  
EMPLOYEE\_POSITION–EMPLOYEE\_START–EMPLOYEE\_END

Name	LEVEL_CODE	LEVEL_DESC	LEVEL_NATURAL_KEY
EMPLOYEE_ID	employee_ID	Store all employee ID numbers by NK passport_ID	employee_passport_ID
EMPLOYEE_FIRST_NAME	employee_first_name	Store all first names of employees	employee_first_name
EMPLOYEE_LAST_NAME	employee_last_name	Store all last names of employees	employee_last_name
EMPLOYEE_POSITION	employee_position	Store all employee positions in the agency	employee_position
EMPLOYEE_START	employee_date_of_hire	Store all the dates the employee started working at the agency	employee_date_of_hire
EMPLOYEE_END	employee_date_of_dismissal	Store all dates of termination of the employee's work at the agency	employee_date_of_dismissal

Summary table to describe all future STAR Fact Table Aggregations:

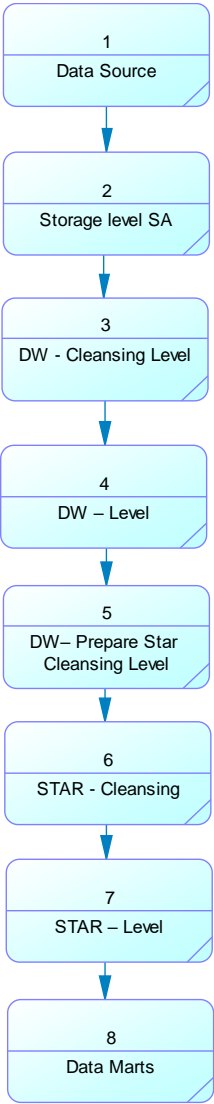
Name	Code	Table Name	Additive	Descriptions
Counts the sum of profit in dollars (GROSS)	gross_profit_dollar_amount	Business_Fact	+	Calculate the sum of profit in dollars (GROSS) in the selected period.
Counts the sum of profit in dollars (NET)	net_profit_dollar_amount	Business_Fact	+	Calculate the sum of profit in dollars (NET) in the selected period.
Counts the sum of revenue in dollars (GROSS)	gross_revenue_dollar_amount	Business_Fact	+	Calculate the sum of revenue in dollars (GROSS) in the selected period.
Counts the sum of revenue in dollars (NET)	net_revenue_dollar_amount	Business_Fact	+	Calculate the sum of revenue in dollars (NET) in the selected period.

Counts the sum of cost in dollars (GROSS)	gross_cost_dollar_amount	Business_Fact	+	Calculate the sum of cost in dollars (GROSS) in the selected period.
Counts the sum of cost in dollars (NET)	net_cost_dollar_amount	Business_Fact	+	Calculate the sum of cost in dollars (NET) in the selected period.
Counts the sum of salary in dollars (GROSS)	gross_salary_employee_dollar_amount	Business_Fact	+	Calculate the sum of salary in dollars (GROSS) in the selected period.
Counts the sum of salary in dollars (NET)	net_salary_employee_dollar_amount	Business_Fact	+	Calculate the sum of salary in dollars (NET) in the selected period.
Counts number of Customers	customer_quantity	Business_Fact	+	Calculate the total amount of Customers in the selected period.
Counts number of Employee	employee_quantity	Business_Fact	+	Calculate the total amount of Employee in the selected period.
Counts number of Promotions	promotion_quantity	Business_Fact	+	Calculate the total amount of Promotions in the selected period.
Ratio of revenues to costs in percent	revenue_cost_percent	Business_Fact	-	Calculate the ratio of revenues to costs in percent.

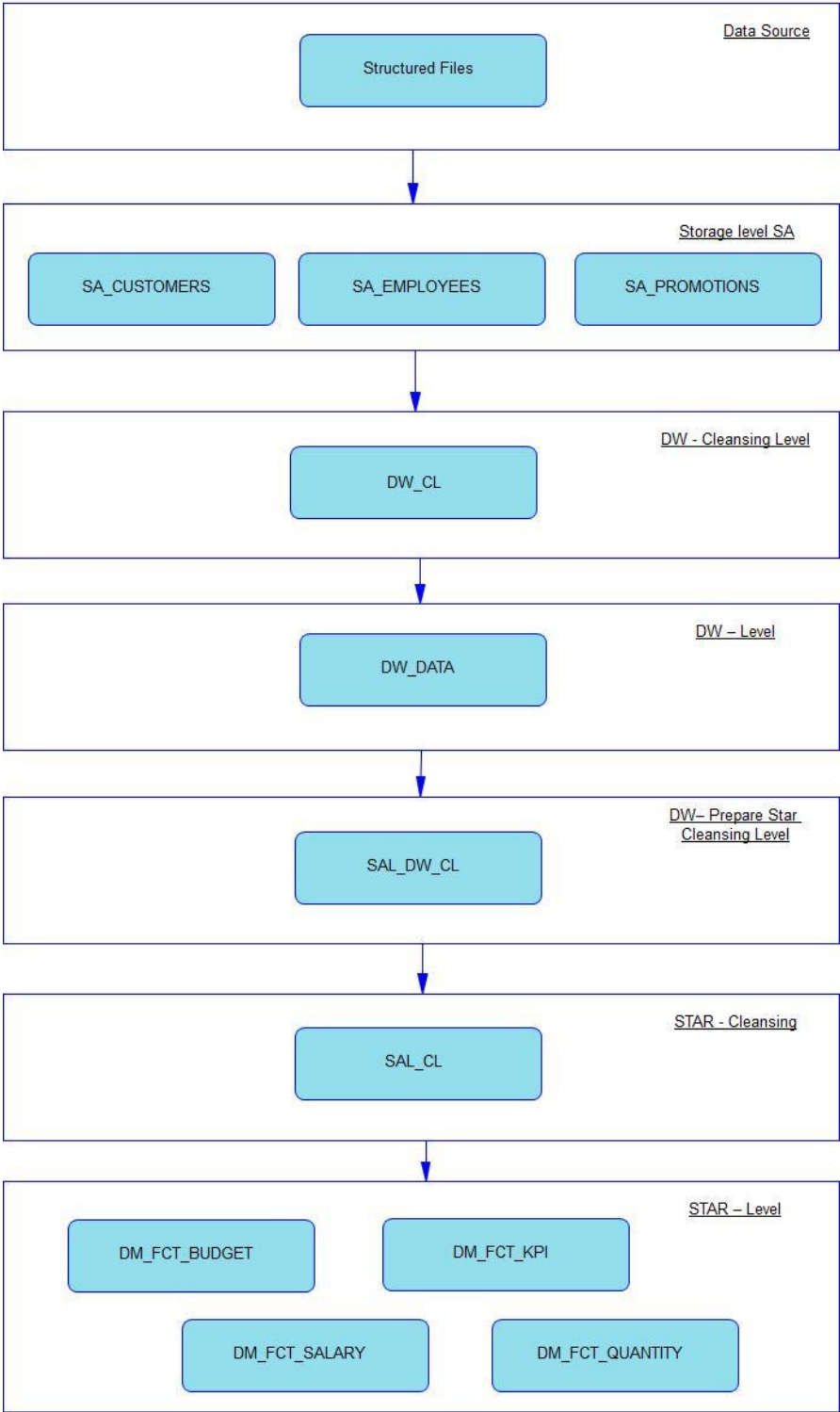
## Name Conversation table

Level Type	Object Name	Tablespace	Description
Storage level SA_*	SA_CUSTOMERS	ts_sa_customers_data_01	Loading from structured files. Contains Customer information.
	SA_EMPLOYEES	ts_sa_employees_data_01	Loading from structured files. Contains Employee information.
	SA_PROMOTIONS	ts_sa_promotions_data_01	Loading from structured files. Contains Promotion information.
DW - Cleansing Level	DW_CL	ts_dw_cl	Loading from a scene-level system. Contains information about preparation for subsequent use (cleaning).
DW – Level	DW_DATA	ts_dw_data_01	Loading data from cleansing tables. Contains clean information tending to the 3rd normal form ready to prepare a star schema.
DW– Prepare Star Cleansing Level	SAL_DW_CL	ts_dw_str_cls	Loading data from DW system. Contains views merging objects from DW level.
STAR - Cleansing	SAL_CL	ts_sal_cl	Loading data from DW_CL system. Contains views from previous level but clean any redundancy.
STAR – Level	DM_FCT_KPI	ts_sa_fct_kpi_01	Store information about fact KPI of ad promotions.
	DM_FCT_BUDGET	ts_sa_fct_budget_01	Store information about fact budget.
	DM_FCT_SALARY	ts_sa_fct_salary_01	Store information about fact salary of employees.
	DM_FCT_QUANTITY	ts_sa_fct_quantity_01	Store information about fact quantity of agency customers, employees and promotions.

**Data Warehouse Architecture diagram**



**DataFlow Diagram to describe refresh process of Business STAR**

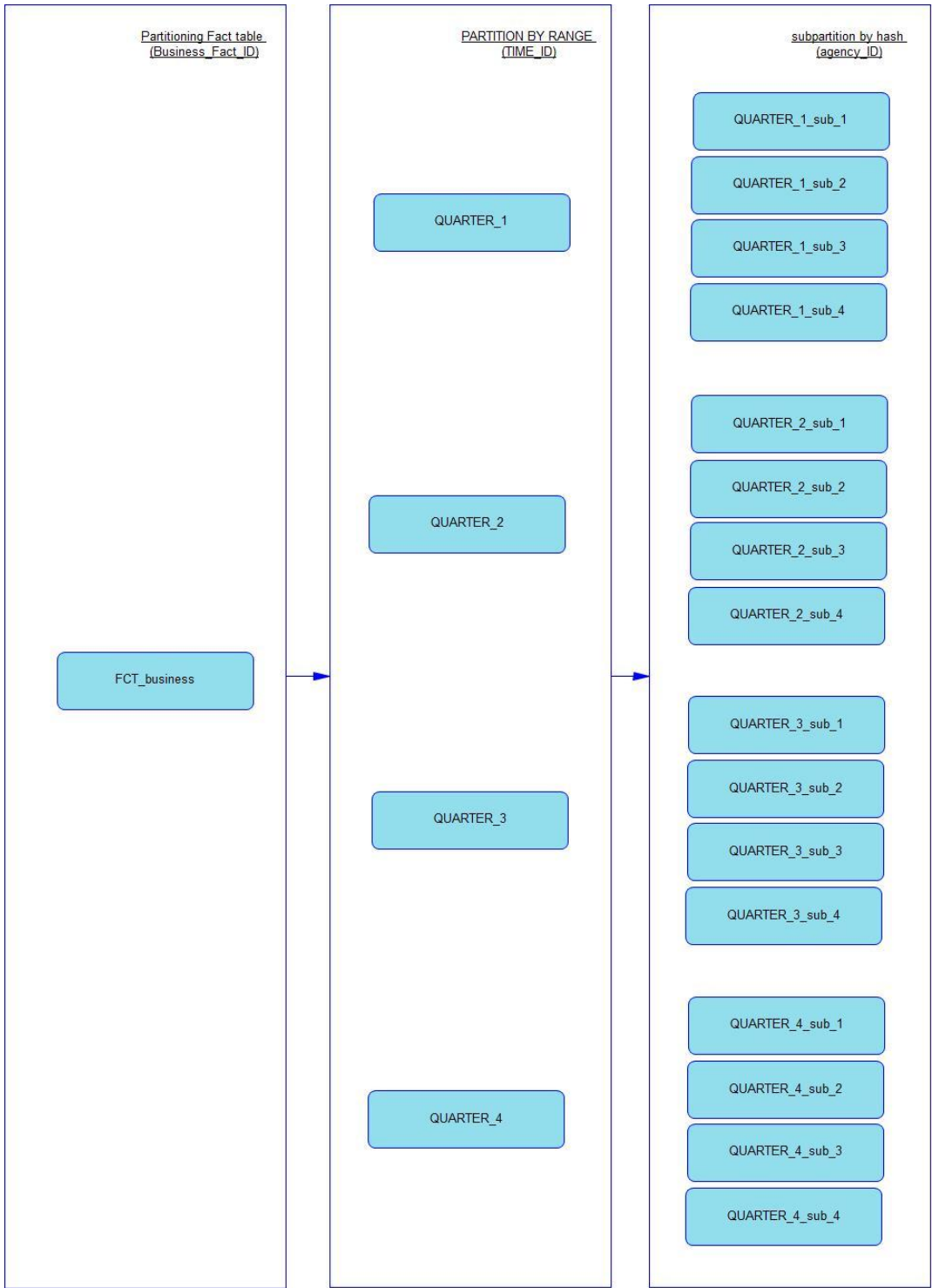


**Partitioning Fact table.**

Partitioning is done to enhance performance and facilitate easy management of data. Partitioning also helps in balancing the various requirements of the system. It optimizes the hardware performance and simplifies the management of data warehouse by partitioning fact table into multiple separate partitions.

In terms of the usability of assessing data from FACT\_TABLE, it is suggested to use data partitioning by quarters in the context of network agencies.

**Partitioning scheme:**



## **Strategy of Parallel refreshing Fact Tables and Dimension Tables.**

Databases today contain a large amount of information. However, finding, updating and presenting the right information in a timely fashion can be a challenge because of the vast quantity of data involved.

Parallel execution is the capability that addresses this challenge. Using parallel execution (also called parallelism), terabytes of data can be processed in minutes, not hours or days, simply by using multiple processes to accomplish a single task. This dramatically reduces response time for data-intensive operations on large databases typically associated with decision support systems (DSS) and data warehouses.

Parallelism is the idea of breaking down a task so that, instead of one process doing all of the work in a query, many processes do part of the work at the same time.

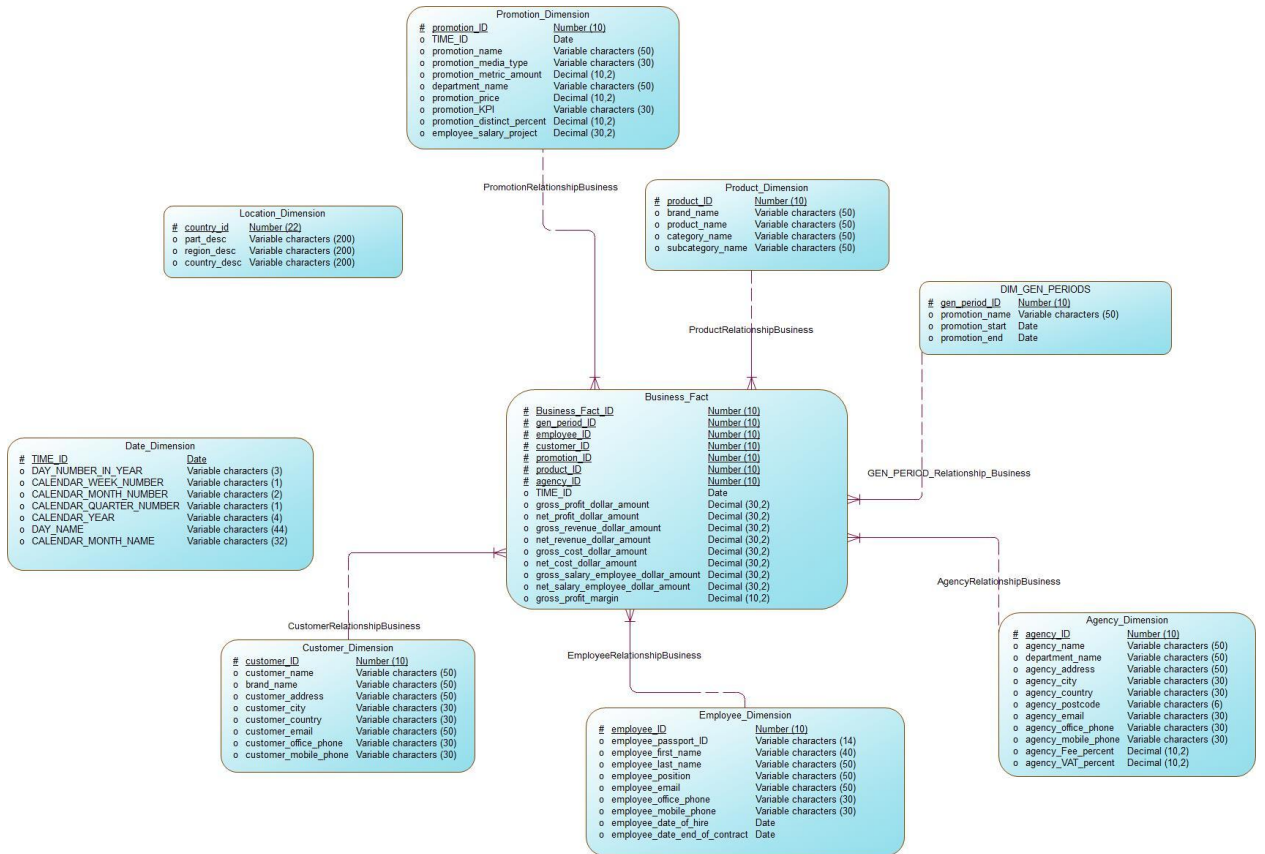
When it becomes necessary to make changes to Fact Tables and Dimension Tables in the data warehouse, we can use parallel queries and parallel subqueries in SELECT statements. We can also parallelize the query portions of DDL statements and DML statements (INSERT, UPDATE, and DELETE). Parallel execution can accelerate large DML operations and is especially useful in data warehouse environments where large summary or history tables must be maintained.



At the customer's request, the following changes were made to the data warehouse in order to optimize costs.

### History of changes

Name	Date	Reason for change	Version
Vera Shkrabatouskaya	7/25/2022	Creation of a STAR data warehouse model.	1.0 draft 1
Vera Shkrabatouskaya	7/26/2022	Changes after approval. Added details to the Business Model.	1.0 approved
Vera Shkrabatouskaya	8/3/2022	Changes after approval. The agency_country field was added to the Agency_Dimension table of the business model. The employee_salary field was renamed to employee_salary_project in Employee_Dimension.	1.1 approved
Vera Shkrabatouskaya	8/11/2022	Changes after approval. The Date_Dimension and Location_Dimension tables can be used as references, but will not be used in the data warehouse. The employee_salary_project field was moved from Employee_Dimension to Promotion_Dimension of the business model. The employee_date_of_dismissal field was renamed to employee_date_end_of_contract in Employee_Dimension. The department_name field and the agency_name field were removed from Employee_Dimension. The promotion_ID field has been renamed to promotion_name in DIM_GEN_PERIODS. Also, the promotion_description field in DIM_GEN_PERIODS has been removed. The promotion_name and promotion_media_type field in Product_Dimension have been removed. The TIME_ID field has been added to the Promotion_Dimension and Business_Fact. The customer_quantity, employee_quantity, and promotion_quantity parameters will not be calculated in this Business_Fact table. The revenue_cost_percent parameter has been replaced with gross_profit_margin due to the change of the repository's goal to calculate the margin indicator.	1.2 approved



## **Evolution of the Data Warehouse**

As the data warehouse is a living IT system, sources and targets might change. Those changes must be maintained and tracked through the lifespan of the system without overwriting or deleting the old ETL process flow information. To build and keep a level of trust about the information in the warehouse, the process flow of each individual record in the warehouse can be reconstructed at any point in time in the future in an ideal case.

### **Extraction Description**

We need to load your data warehouse regularly so that it can serve its purpose of facilitating business analysis. To do this, data from one or more operational systems needs to be extracted and copied into the data warehouse. The challenge in data warehouse environments is to integrate, rearrange and consolidate large volumes of data over many systems, thereby providing a new unified information base for business intelligence.

The process of extracting data from source systems and bringing it into the data warehouse is commonly called ETL, which stands for extraction, transformation, and loading.

The methodology and tasks of ETL have been well known for many years, and are not necessarily unique to data warehouse environments: a wide variety of proprietary applications and database systems are the IT backbone of any enterprise. Data has to be shared between applications or systems, trying to integrate them, giving at least two applications the same picture of the world. This data sharing was mostly addressed by mechanisms similar to what we now call ETL.

During extraction, the desired data is identified and extracted from many different sources, including database systems and applications. Very often, it is not possible to identify the specific subset of interest, therefore more data than necessary has to be extracted, so the identification of the relevant data will be done at a later point in time. Depending on the source system's capabilities (for example, operating system resources), some transformations may take place during this extraction process. The size of the extracted data varies from hundreds of kilobytes up to gigabytes, depending on the source system and the business situation. The same is true for the time delta between two (logically) identical extractions: the time span may vary between days/hours and minutes to near real-time.

The method for extracting data from a data warehouse is highly dependent on the original system as well as the business needs of the target data warehouse environment.

In our business model, we will use full extraction as the logical extraction method and offline extraction as the physical extraction method.

### **Full Extraction**

The data is extracted completely from the source system. Because this extraction reflects all the data currently available on the source system, there's no need to keep track of changes to the data source since the last successful extraction. The source data will be provided as-is and no additional logical information (for example, timestamps) is necessary on the source site.

## Offline Extraction

The data is not extracted directly from the source system but is staged explicitly outside the original source system. The data already has an existing structure (for example, redo logs, archive logs or transportable tablespaces) or was created by an extraction routine.

## **Transportation Description**

After data is extracted, it has to be physically transported to the target system or to an intermediate system for further processing. Depending on the chosen way of transportation, some transformations can be done during this process, too.

Transportation is the operation of moving data from one system to another system. In a data warehouse environment, the most common requirements for transportation are in moving data from:

- A source system to a staging database or a data warehouse database
- A staging database to a data warehouse
- A data warehouse to a data mart

There are three basic choices for transporting data in warehouses:

- Transportation Using Flat Files
- Transportation Through Distributed Operations
- Transportation Using Transportable Tablespaces

For our business model, we suggest considering Transportation Using Flat Files.

## Transportation Using Flat Files

The most common method for transporting data is by the transfer of flat files, using mechanisms such as FTP or other remote file system access protocols. Data is unloaded or exported from the source system into flat files and is then transported to the target platform using FTP or similar mechanisms.

Because source systems and data warehouses often use different operating systems and database systems, using flat files is often the simplest way to exchange data between heterogeneous systems with minimal transformations. However, even when transporting data between homogeneous systems, flat files are often the most efficient and most easy-to-manage mechanism for data transfer.