Solution Concept Document
Business Project

Shkrabatouskaya Vera

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 date into usable information.

Requirements

Business Requirements

ID	Description
BR-01	Calculation of revenue information (Gross, Net) on a daily, monthly, quarterly, annual basis by agency, customer, brand, promotion.
BR-02	Calculation of cost information (Gross, Net) on a daily, monthly, quarterly, annual basis by agency, customer, brand, promotion.
BR-03	Calculation of profit information (Gross, Net) on a daily, monthly, quarterly, annual basis by agency, customer, brand, promotion.
BR-04	Calculation of the Starcom advertising network business margin, expressed as a percentage.
BR-05	Calculation of employee salary information (Gross, Net) on a monthly, annual basis by agency, employee.
BR-06	Calculation of information about the number of promotions, the number of brands participating in promotions, the number of employees providing advertising services on a daily, monthly basis.
BR-07	Calculation of information about completed KPI for an advertising campaign on a daily basis.

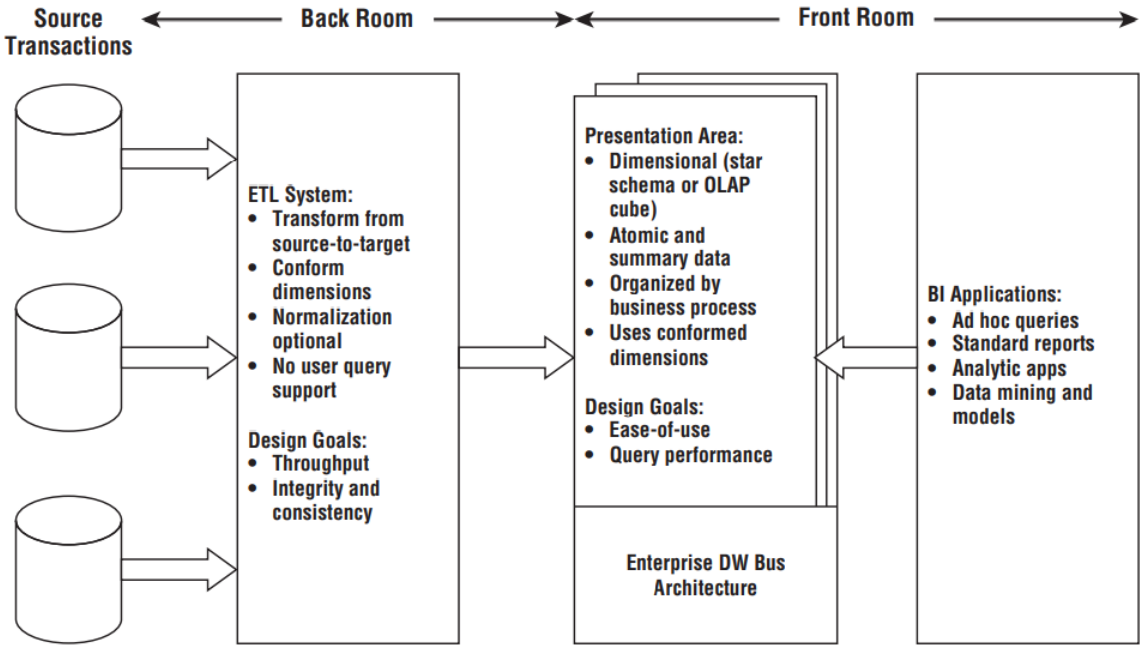
Technical Requirements

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:

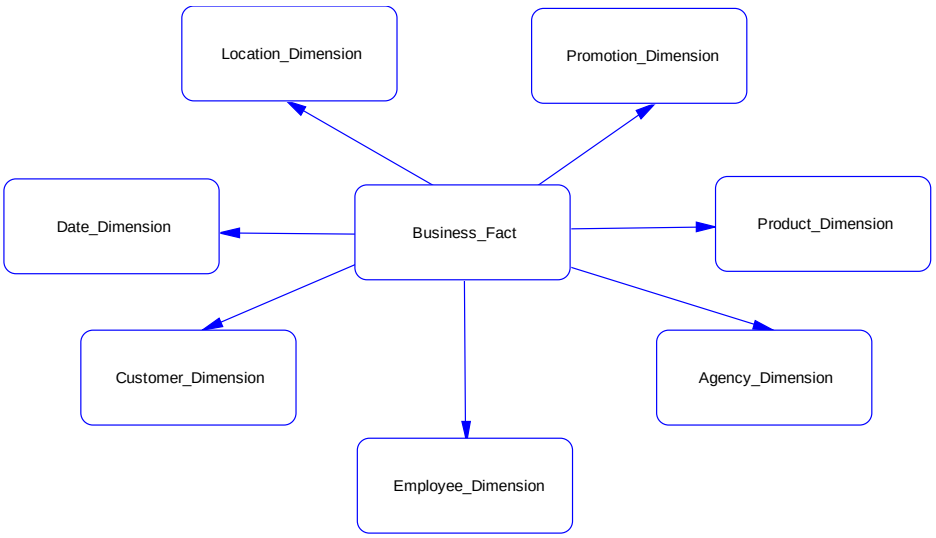
- A system will be designed and developed for Starcom DWH that can upload, analyze and import data from any data source.
- Any employee can link his data source to a Data Warehouse which will lead to quickly sharing important information within the company.

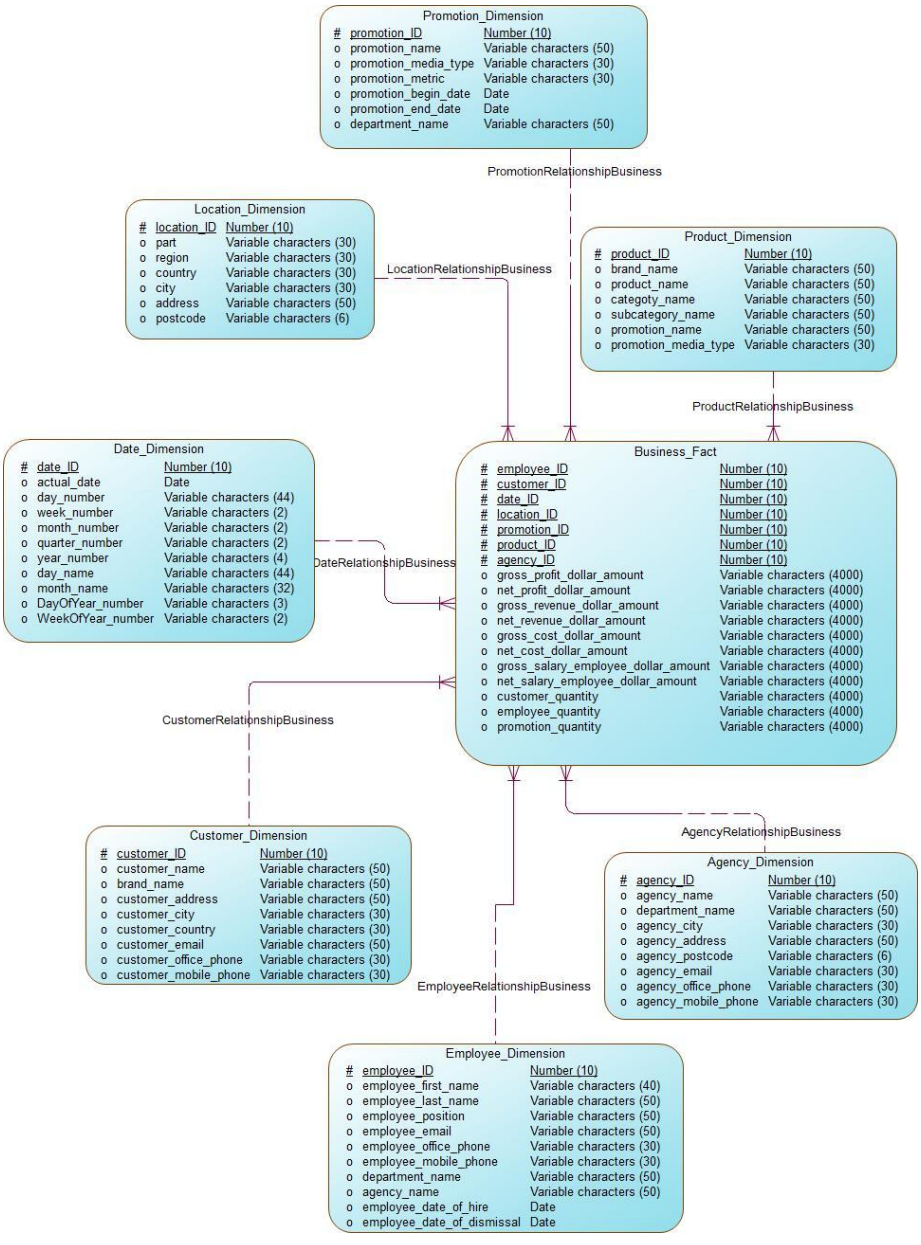
Working closely with representatives from the business and other DW/BI team members, the Data Warehouse Design Specialist developed a fact sheet with a grain of reporting for the customer, agency management and agency employees for each day.

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.

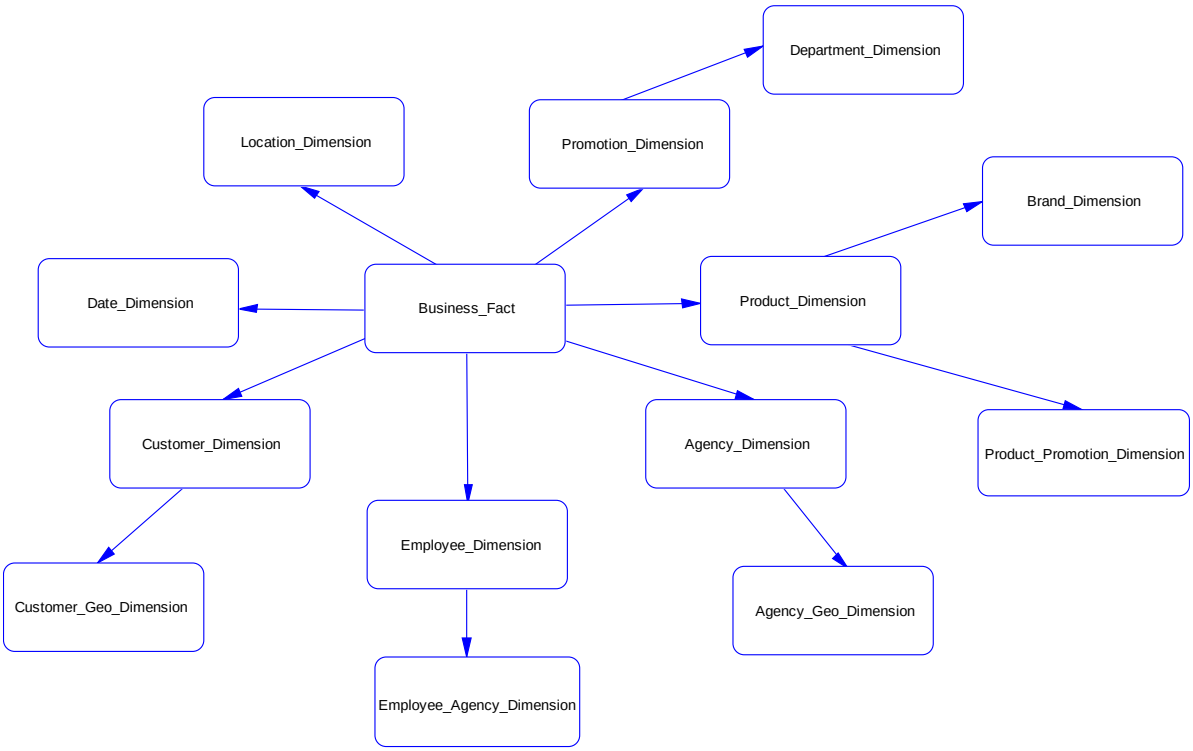
For the database, English was chosen for all locations in the agency's network. All monetary data is given in dollars.

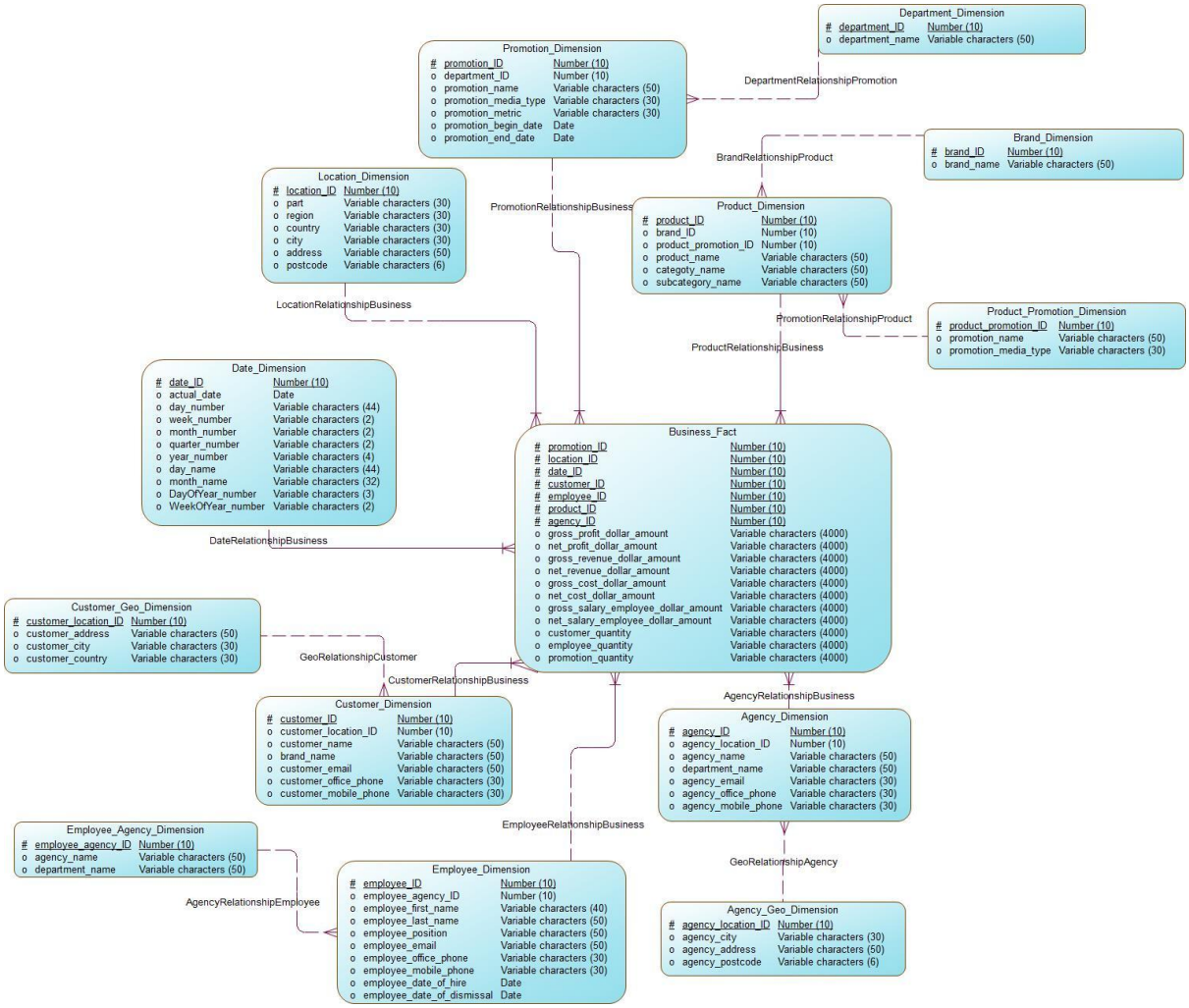
Star Scheme:


Location_Dimension
Promotion_Dimension
Date_Dimension
Business_Fact
Product_Dimension
Customer_Dimension
Agency_Dimension
Employee_Dimension


Promotion_Dimension
promotion_ID Number (10)
o promotion_name Variable characters (50)
o promotion_media_type Variable characters (30)
o promotion_metric Variable characters (30)
o promotion_begin_date Date
o promotion_end_date Date
o department_name Variable characters (50)
PromotionRelationshipBusiness
Location_Dimension
location_ID Number (10)
o part Variable characters (30)
o region Variable characters (30)
o country Variable characters (30)
o city Variable characters (30)
o address Variable characters (50)
o postcode Variable characters (6)
LocationRelationshipBusiness
Product_Dimension
product_ID Number (10)
o brand_name Variable characters (50)
o product_name Variable characters (50)
o categoty_name Variable characters (50)
o subcategory_name Variable characters (50)
o promotion_name Variable characters (50)
o promotion_media_type Variable characters (30)
ProductRelationshipBusiness
Date_Dimension
date_ID Number (10)
o actual_date Date
o day_number Variable characters (44)
o week_number Variable characters (2)
o month_number Variable characters (2)
o quarter_number Variable characters (2)
o year_number Variable characters (4)
o day_name Variable characters (44)
o month_name Variable characters (32)
o DayOfYear_number Variable characters (3)
o WeekOfYear_number Variable characters (2)
DateRelationshipBusiness
Business_Fact
employee_ID Number (10)
customer_ID Number (10)
date_ID Number (10)
location_ID Number (10)
promotion_ID Number (10)
product_ID Number (10)
agency_ID Number (10)
o gross_profit_dollar_amount Variable characters (4000)
o net_profit_dollar_amount Variable characters (4000)
o gross_revenue_dollar_amount Variable characters (4000)
o net_revenue_dollar_amount Variable characters (4000)
o gross_cost_dollar_amount Variable characters (4000)
o net_cost_dollar_amount Variable characters (4000)
o gross_salary_employee_dollar_amount Variable characters (4000)
o net_salary_employee_dollar_amount Variable characters (4000)
o customer_quantity Variable characters (4000)
o employee_quantity Variable characters (4000)
o promotion_quantity Variable characters (4000)
CustomerRelationshipBusiness
AgencyRelationshipBusiness
Customer_Dimension
customer_ID Number (10)
o customer_name Variable characters (50)
o brand_name Variable characters (50)
o customer_address Variable characters (50)
o customer_city Variable characters (30)
o customer_country Variable characters (30)
o customer_email Variable characters (50)
o customer_office_phone Variable characters (30)
o customer_mobile_phone Variable characters (30)
Agency_Dimension
agency_ID Number (10)
o agency_name Variable characters (50)
o department_name Variable characters (50)
o agency_city Variable characters (30)
o agency_address Variable characters (50)
o agency_postcode Variable characters (6)
o agency_email Variable characters (30)
o agency_office_phone Variable characters (30)
o agency_mobile_phone Variable characters (30)
EmployeeRelationshipBusiness
Employee_Dimension
employee_ID Number (10)
o employee_first_name Variable characters (40)
o employee_last_name Variable characters (50)
o employee_position Variable characters (50)
o employee_email Variable characters (50)
o employee_office_phone Variable characters (30)
o employee_mobile_phone Variable characters (30)
o department_name Variable characters (50)
o agency_name Variable characters (50)
o employee_date_of_hire Date
o employee_date_of_dismissal Date

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.

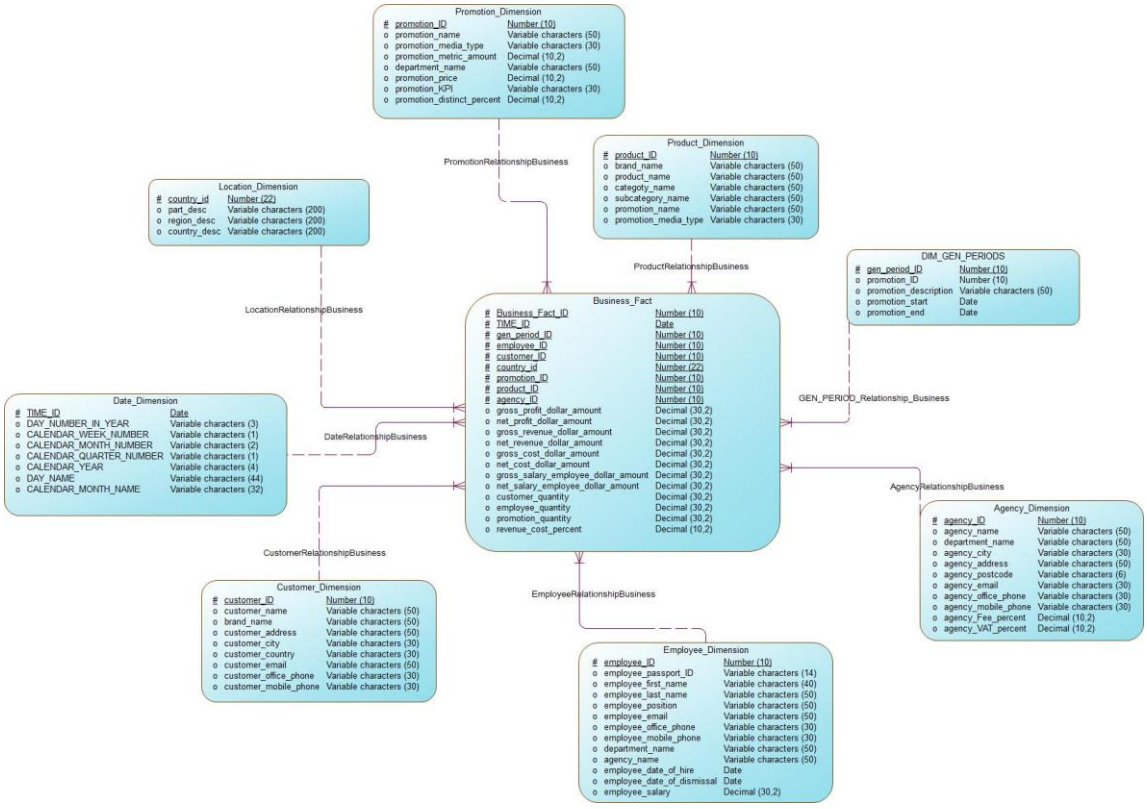
The star schema has fewer links between the dimension table and the fact table compared to the snowflake schema, which has many links, which explains the lower query complexity. Because measurements in a star type scheme are linked through a central fact table, it has clear connection paths, which means 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 categoty_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-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
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 MONTH-QUARTER-YEAR

Name	LEVEL_CODE	LEVEL_DESC	LEVEL_NATURAL_KEY
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:

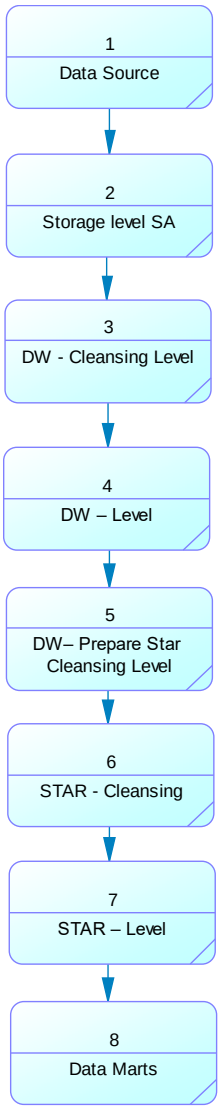
Name	Code	Table Name	Additive	Descriptions
Counts the sum of profit in dollars (GROSS)	gross_profit_dollar_amount	Business_Fact	+	Calculate the sum of profit in dollars (GROSS) in the selected period.
Counts the sum of profit in dollars (NET)	net_profit_dollar_amount	Business_Fact	+	Calculate the sum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

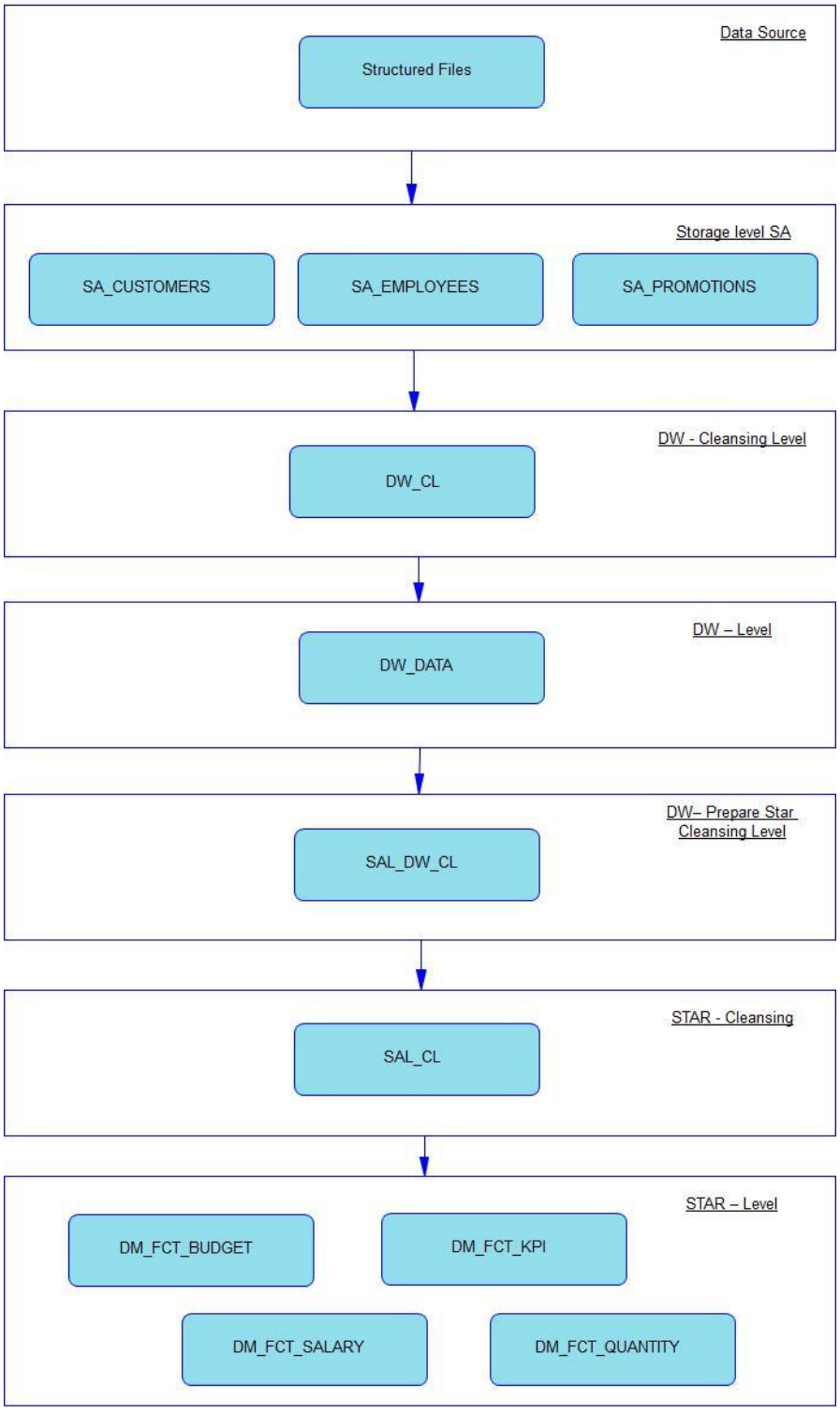
Name Conversation table

Level Type	Object Name	Tablespace	Desctiption
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

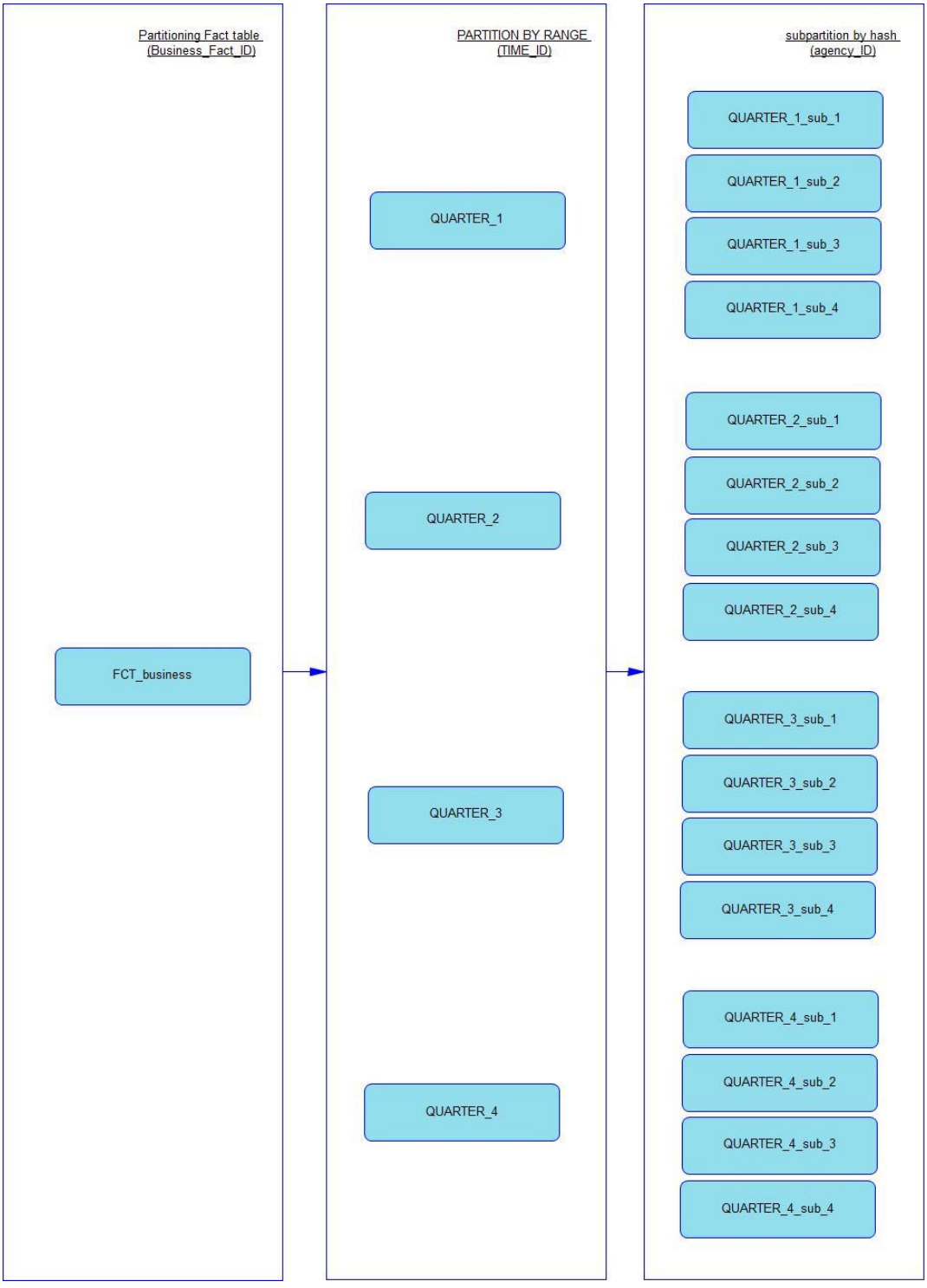


Partitioning Fact table.

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.

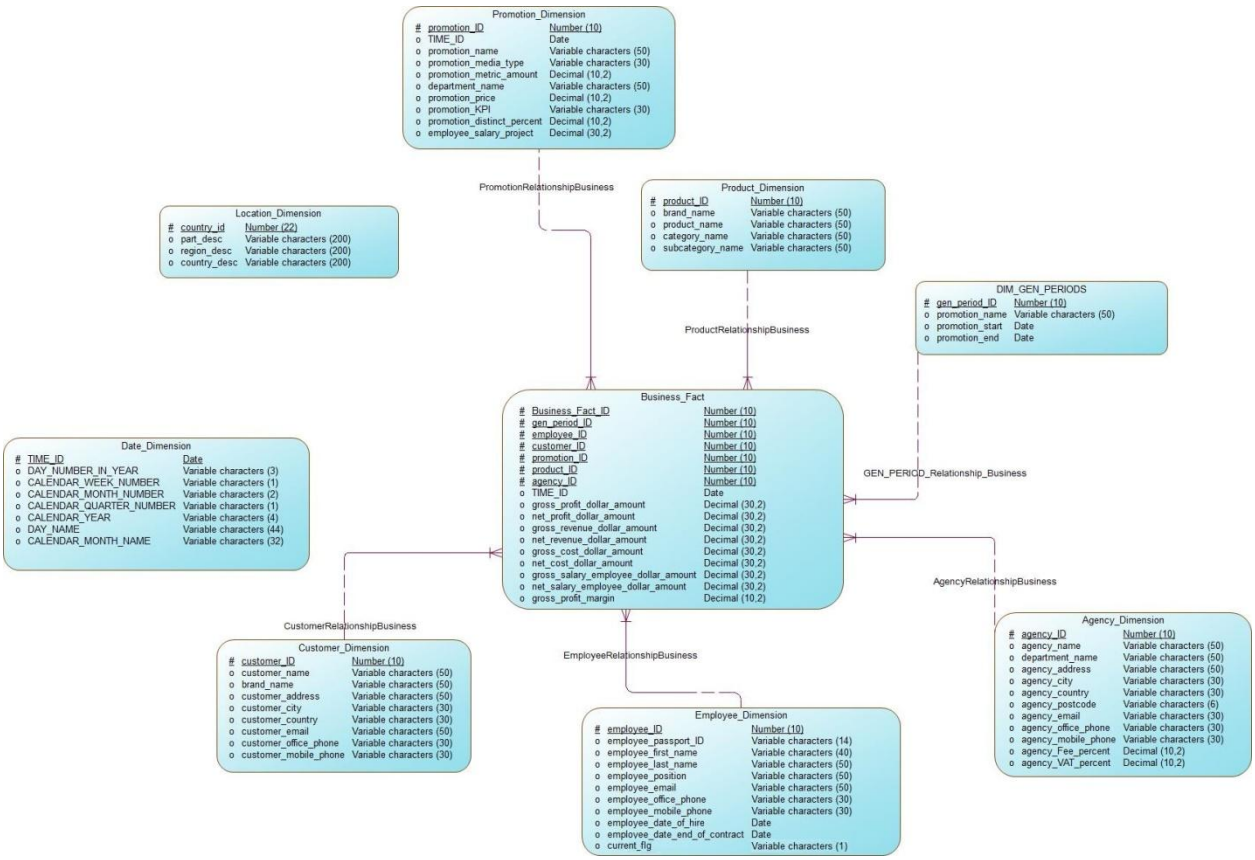
Parallelism is the idea of breaking down a task so that, instead of one process doing all of the work in a query, many processes do part of the work at the same time.

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 current_flg field was added to Employee_Dimension. 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


Promotion_Dimension
promotion_ID Number (10)
o TIME_ID Date
o promotion_name Variable characters (50)
o promotion_media_type Variable characters (30)
o promotion_metric_amount Decimal (10,2)
o department_name Variable characters (50)
o promotion_price Decimal (10,2)
o promotion_KPI Variable characters (30)
o promotion_distinct_percent Decimal (10,2)
o employee_salary_project Decimal (30,2)
PromotionRelationshipBusiness
Location_Dimension
country_id Number (22)
o part_desc Variable characters (200)
o region_desc Variable characters (200)
o country_desc Variable characters (200)
Product_Dimension
product_ID Number (10)
o brand_name Variable characters (50)
o product_name Variable characters (50)
o category_name Variable characters (50)
o subcategory_name Variable characters (50)
ProductRelationshipBusiness
DIM_GEN_PERIODS
gen_period_ID Number (10)
o promotion_name Variable characters (50)
o promotion_start Date
o promotion_end Date
Business_Fact
Business_Fact_ID Number (10)
gen_period_ID Number (10)
employee_ID Number (10)
customer_ID Number (10)
promotion_ID Number (10)
product_ID Number (10)
agency_ID Number (10)
o TIME_ID Date
o gross_profit_dollar_amount Decimal (30,2)
o net_profit_dollar_amount Decimal (30,2)
o gross_revenue_dollar_amount Decimal (30,2)
o net_revenue_dollar_amount Decimal (30,2)
o gross_cost_dollar_amount Decimal (30,2)
o net_cost_dollar_amount Decimal (30,2)
o gross_salary_employee_dollar_amount Decimal (30,2)
o net_salary_employee_dollar_amount Decimal (30,2)
o gross_profit_margin Decimal (10,2)
GEN_PERIOD_Relationship_Business
Date_Dimension
TIME_ID Date
o DAY_NUMBER_IN_YEAR Variable characters (3)
o CALENDAR_WEEK_NUMBER Variable characters (1)
o CALENDAR_MONTH_NUMBER Variable characters (2)
o CALENDAR_QUARTER_NUMBER Variable characters (1)
o CALENDAR_YEAR Variable characters (4)
o DAY_NAME Variable characters (44)
o CALENDAR_MONTH_NAME Variable characters (32)
AgencyRelationshipBusiness
CustomerRelationshipBusiness
Customer_Dimension
customer_ID Number (10)
o customer_name Variable characters (50)
o brand_name Variable characters (50)
o customer_address Variable characters (50)
o customer_city Variable characters (30)
o customer_country Variable characters (30)
o customer_email Variable characters (50)
o customer_office_phone Variable characters (30)
o customer_mobile_phone Variable characters (30)
EmployeeRelationshipBusiness
Employee_Dimension
employee_ID Number (10)
o employee_passport_ID Variable characters (14)
o employee_first_name Variable characters (40)
o employee_last_name Variable characters (50)
o employee_position Variable characters (50)
o employee_email Variable characters (50)
o employee_office_phone Variable characters (30)
o employee_mobile_phone Variable characters (30)
o employee_date_of_hire Date
o employee_date_end_of_contract Date
o current_flg Variable characters (1)
Agency_Dimension
agency_ID Number (10)
o agency_name Variable characters (50)
o department_name Variable characters (50)
o agency_address Variable characters (50)
o agency_city Variable characters (30)
o agency_country Variable characters (30)
o agency_postcode Variable characters (6)
o agency_email Variable characters (30)
o agency_office_phone Variable characters (30)
o agency_mobile_phone Variable characters (30)
o agency_Fee_percent Decimal (10,2)
o agency_VAT_percent Decimal (10,2)

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.

ETL database tools perform several critical business functions:

- Reconcile varying data formats to move data from a legacy system into modern technology, which often cannot support the legacy format
- Sync data from external ecosystem partners, like suppliers, vendors, and customers
- Consolidate data from various overlapping systems acquired via merger and/or acquisition
- Combine transactional data from a data store so it can be read and understood by business users

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 company. 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.

Loading and Transformation Description

Data transformations are often the most complex and, in terms of processing time, the most costly part of the extraction, transformation, and loading (ETL) process. They can range from simple data conversions to extremely complex data scrubbing techniques. Many, if not all, data transformations can occur within an Oracle database, although transformations are often implemented outside of the database (for example, on flat files) as well.

Data transformation is the T in ETL. Data Transformation means that data in one format is processed, either inside or outside the data store and persisted in the new required format. This transformation can take several forms, from column storage types, file types, or even encoding types. There is no "single database" solution for company, so data reside in different locations and formats making data transformation necessary to ensure data from one application or database is available to other applications and databases.

Data is oxygen for the modern company, and ETL has enabled a lot of value. To answer many of the business' analytical questions, it's essential to integrate these information silos and leverage existing IT assets to create more flexible, agile enterprise systems. One way to do this is through data transformation.

In the ETL process, Transform is the process of converting the extracted data from its previous form into the form it needs to be in so that it can be placed into another database. Transformation occurs by using rules or lookup tables or by combining the data with other data or potentially code.

Transformation Flow

From an architectural perspective, data can be transformed in the following ways:

- Multistage Data Transformation
- Pipelined Data Transformation
- Staging Area

Multistage Data Transformation

When using Oracle Database as a transformation engine, a common strategy is to implement each transformation as a separate SQL operation and to create a separate, temporary staging table to store the incremental results for each step. This load-then-transform strategy also provides a natural checkpointing scheme to the entire transformation process, which enables the process to be more easily monitored and restarted. However, a disadvantage to multistaging is that the space and time requirements increase.

It may also be possible to combine many simple logical transformations into a single SQL statement or single PL/SQL procedure. Doing so may provide better performance than performing each step independently, but it may also introduce difficulties in modifying, adding, or dropping individual transformations, as well as recovering from failed transformations.

Pipelined Data Transformation

The ETL process flow can be changed dramatically and the database becomes an integral part of the ETL solution.

The new functionality renders some of the former necessary process steps obsolete while some others can be remodeled to enhance the data flow and the data transformation to become more scalable and non-interruptive. The task shifts from serial transform-then-load process (with most of the tasks done outside the database) or load-then-transform process, to an enhanced transform-while-loading.

It is important to understand that the database offers toolkit functionality rather than trying to address a one-size-fits-all solution. The underlying database has to enable the most appropriate ETL process flow for a specific customer need, and not dictate or constrain it from a technical perspective.

Staging Area

The overall speed of your load is determined by how quickly the raw data can be read from the staging area and written to the target table in the database. It is highly recommended that you stage your raw data across as many physical disks as possible to ensure the reading of the raw data is not a bottleneck during the load.

For the Starcom business model, we will use - Multistage Data Transformation.

Lambda vs Kappa Architecture

ETL is the process of extracting, transforming, and loading data from various sources into data warehouse systems. It used to be done in batches, but modern architectures are evolving and ETL now happens in near real-time whenever possible. Besides near real-time capabilities, a modern data infrastructure should be flexible, scalable, automated, elastic, extensible, as well as being able to support updates and new applications.

In this context, Kappa and Lambda architectures can handle both real-time (streaming data) and static (batch) datasets, but they do so in very different ways.

Lambda Architecture

Lambda architecture comprises of Batch Layer, Speed Layer (also known as Stream layer) and Serving Layer. When data feeds the system, the Batch Layer and the Speed Layer receive it simultaneously. This starts the processing, and then, the batch layer corrects the stream data. However, each layer has a specific task to complete.

This means that Lambda architecture separates the processing of real-time data from batch data. The Batch Layer can take its time to process tons of data that take a lot of computation time while the Speed Layer computes in real-time and performs incremental updates to the batch layer results. The Serving Layer takes the outputs of both and uses this data to solve pending queries.

Kappa Architecture

The Kappa Architecture is event-based and only has the Streaming Layer and the Serving Layer, so every kind of data that needs to be processed will be handled by a single technology stack. This paradigm solves the redundancy in Lambda architectures, avoiding the maintenance of two different code bases to feed the layers, as well as the multi-layered structure.

The idea behind doing everything in the Streaming Layer is replaying data: this means handling continuous data reprocessing and real-time data using only one processing engine, thus avoiding maintaining different code for different layers. Kappa may appear simpler, but what happens when data is out of order and needs to be rearranged, a field needs to be added, or items need to be cross-referenced? These challenges are solved easier by batch-processing and Kappa architectures were not designed to handle this kind of operation, although solutions to those problems have emerged with time.

When starting a big data project, a comprehensive analysis helps decide which architecture to use, remembering that migration from one to the other is not simple and sometimes an impossible mission. So, choosing the right architecture from the beginning is key.

For the Starcom business model, it is proposed to choose the Lambda architecture.