**Kelompok 3 BIE:**

Muhammad Raihan Firdaus

Anif Shofiana Durri

Rohmat Munzaini

Agung Adi

Technical Requirements Document (TRD)

Sales & Marketing Department Data Mart

Project/Initiative

February 2022

Version 0.3

*Superstore Inc.*

# Document Revisions

| **Date** | **Version Number** | **Document Changes** |
| --- | --- | --- |
| 05/02/2022 | 0.1 | Initial Draft |
| 08/08/2022 | 1.1 | Update Document |
| 10/08/2022 | 1.2 | final |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Approvals

| **Role** | **Name** | **Title** | **Signature** | **Date** |
| --- | --- | --- | --- | --- |
| User | mas permana | stack holder (User) |  | 08-Aug-2022 |
| Business Intelligence | Muhammad Raihan firdaus | BIE-6-*Student* (*Lead*) |  | 08-Aug-2022 |
| Business Intelligence | Anif Shofiana Durri | BIE-6-*Student* (Co-*Lead*) |  | 08-Aug-2022 |
| Business Intelligence | Rohmat Munzaini | BIE-6-*Student* (*Member*) |  | 08-Aug-2022 |
| Business Intelligence | Agung Adi | BIE-6-*Student* (*Member*) |  | 08-Aug-2022 |

# Introduction

## Project Overview

Proyek ini adalah project pembuatan data mart yang dapat mengintegrasikan, membersihkan, dan mengagregasi data dari berbagai data source ke dalam data mart masing-masing departemen, yang dalam proyek kali ini adalah departemen Sales dan Marketing. Proyek ini mencakup pembuatan arsitektur (technical design), infrastruktur (analytical table), dan workflow (proses etl) end-to-end dari data mart Sales & Marketing.

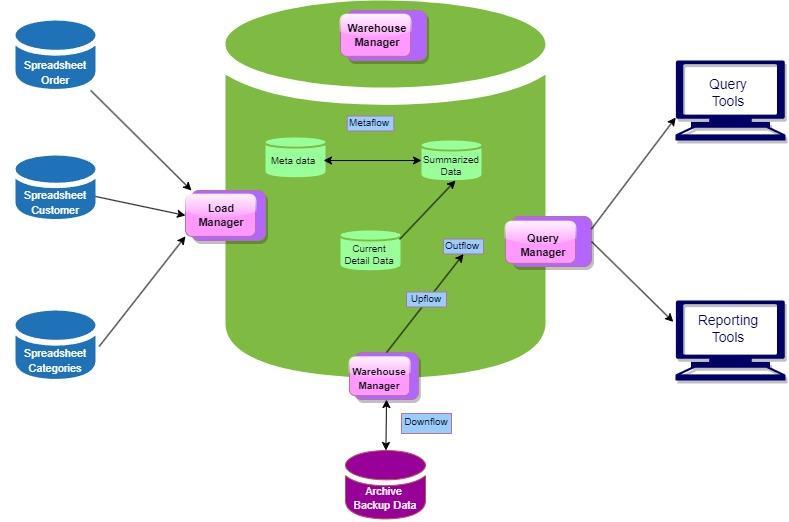
## Project Objectives

Proyek ini bertujuan untuk membuat data mart yang robust, valid, stabil, dan dapat memenuhi kebutuhan usernya. Parameter utama keberhasilan pembuatan data mart ini adalah data mart ini harus dapat digunakan sebagai data source dashboard monitoring yang akan digunakan oleh departemen Marketing dan Sales, guna menggantikan proses manual yang sebelumnya dilakukan.

## Technical Design Diagram

Technical Design Diagram data environment & pipeline Superstore, Pltd:

Data Source → Data Warehouse → Dashboard /Report



**Penjelasan bagan di atas sebagai berikut :**

1. Spreadsheet Order, Spreadsheet Customer dan Spreadsheet Categories adalah sumber data yang digunakan (data source)
2. Setelah itu dari data source dilakukan proses input data ke warehouse manager yang biasa kami sebut load manager
3. Dalam data warehouse dijalankan proses metaflow. Dalam proses ini dapat dilakukan penyeleksian data yaitu :

* Meta data
* Summarized data
* Current detail data

1. Setelah proses seleksi data selesai selanjutnya data tersebut masuk ke dalam backup data yang berfungsi membuat salinan data/arsip data jika dikemudian hari terdapat data yang hilang atau rusak
2. Kemudian yang terakhir data yang ada dalam data warehouse akan di query untuk keperluan dashboard monitoring sales & marketing

**Data Quality Metrics**

| Metrics | Completeness | Accuracy | Timeliness |
| --- | --- | --- | --- |
| Pengecekan data spreadsheet | Tidak ada status null | Pengecekan data yang terduplikasi | Pengecekan standart input data harus <5 tahun atau data terbaru |
| Hasil | * Superstore\_customer : tidak ada data status null * Superstore\_order : terdapat 200 data status null * Superstore\_categories : tidak ada data status null | * Superstore\_categories : beberapa dari product\_id memiliki banyak product\_name * Superstore\_order : tidak ada data yang redundant * Superstore\_customer : tidak ada data yang redundant | Superstore\_order sudah berisi data terbaru (<5tahun) |
| Tindakan | Menghilangkan baris yang memiliki data status null | Hanya mengambil satu product\_name dengan melakukan index partitioning menggunakan fungsi row\_number() | Tidak ada |

Dari tabel di atas hal hal yang dilakukan untuk data quality metrik ada 3 hal yaitu :

1. Completeness

Pada tahap completenes dilakukan pengecekan data null values. Hasil yang didapatkan adalah data spreadsheet superstore categories no null values, spreadsheet superstore customers no null values dan spreadsheet superstore order terdapat 200 null values. Kemudian dari hasil data null values pada superstore order dihilangkan,

1. Accuracy

Pada tahap ini dilakukan pengecekan data yang redundan. Hasil pengecekan data tersebut didapatkan pada spreadsheet superstore categories terdapat beberapa dari product\_id yang memiliki banyak product\_name, kemudian pada spreadsheet superstore customers dan order tidak ditemukan data yang redundan. Dari hasil yang didapat diambil satu product\_name dengan melakukan index partitioning menggunakan fungsi row\_number().

1. Timelines

Pada tahap ini dilakukan pengecekan standard data dimana data yang diinput adalah data 5 tahun terakhir atau data terbaru. Dalam proses pengecekan tersebut tidak ditemukan data yang melebihi 5 tahun.

**Show Case Data Quality Metrics:**

1. **Null values pada tabel Superstore\_Order**

Kami melakukan checking null values pada tabel Superstore\_Order dengan query sebagai berikut:

select

sum(case when order\_date is null then 1 else 0 end) as order\_date\_check

,sum(case when order\_id is null then 1 else 0 end) as order\_id\_check

,sum(case when customer\_id is null then 1 else 0 end) as customer\_id\_check

,sum(case when ship\_date is null then 1 else 0 end) as ship\_datecheck

,sum(case when ship\_mode is null then 1 else 0 end) as ship\_modecheck

,sum(case when customer\_id is null then 1 else 0 end) as customer\_id\_check

,sum(case when country is null then 1 else 0 end) as country\_check

,sum(case when city is null then 1 else 0 end) as city\_check

,sum(case when state is null then 1 else 0 end) as state\_check

,sum(case when postal\_code is null then 1 else 0 end) as postal\_code\_check

,sum(case when region is null then 1 else 0 end) as region\_check

,sum(case when product\_id is null then 1 else 0 end) as product\_id\_check

,sum(case when sales is null then 1 else 0 end) as sales\_check

,sum(case when quantity is null then 1 else 0 end) as quantity\_check

,sum(case when profit is null then 1 else 0 end) as profit\_check

,sum(case when discount is null then 1 else 0 end) as discount\_check

from `test1.superstore\_order` a

Hasilnya: 

Hasilnya, terdapat 200 null values pada table Superstore\_Order. Selanjutnya, handling yang kami lakukan adalah dengan mengambil data Superstore\_Order yang tidak mengandung null values.

select \* from `test1.superstore\_order`

where

1=1

and order\_date is not null

1. **Data Redundant pada Superstore\_Order**

Kami melakukan duplicated value testing untuk memastikan tidak ada data berulang pada tabel. Query yang kami gunakan adalah sebagai berikut:

select

count(product\_id) as product\_id

,count(distinct product\_id) as distinct\_product\_id

from `test1.superstore\_categories`

Hasilnya, kami menemukan data redundant (duplicated values) pada tabel Superstore\_Orders. Hal ini terlihat dari jumlah row product\_id tidak sama ketika dilakukan perhitungan count dan count distinct.



Lebih jauh lagi, kami menemukan bahwa product\_name tidak unique terhadap product\_id. Ada beberapa product\_id yang memiliki lebih dari satu product\_name. Jika tidak dibersihkan, hal ini akan menyebabkan redundancy (keberulangan) data saat dijoin dengan table Superstore\_Order.

with superstore\_categories\_cleaned as

(SELECT product\_id

,sub\_category

,category

,params as product\_name

,row\_number() over(partition by product\_id) as row\_num

from (select product\_id

, replace(category, 'Office Supply', 'Office Supplies') as category

, sub\_category

, product\_name

, product\_id ||' '|| product\_name as params

FROM `test1.superstore\_categories` )

)

select

product\_id

,category

,sub\_category

,product\_name

from superstore\_categories\_cleaned

where

1=1

and category != 'Category'

and row\_num = 1

untuk mengatasinya, kami melakukan concatenate product\_id dan product\_name yang kemudian akan dianggap sebagai product\_name (agar bersifat unique) dan menggunakan function row\_numbering & partitioning. Kami hanya mengambil value yang memiliki row\_number = 1 untuk memastikan semua data yang didapatkan adalah unique dan tidak redundant.

### Components

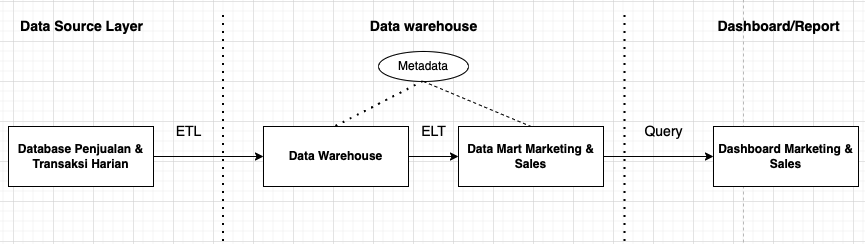
* *Connection & Location*

Data yang digunakan pada proyek ini berasal dari spreadsheet data superstore yang disimpan dalam google drive dan telah dimuat dalam BigQuery. Data superstore yang digunakan merupakan data internal perusahaan yang nantinya akan diinput setiap harinya oleh divisi sales dan marketing. Data location pada proyek ini adalah asia-southeast1

* *Server & Storage*

Proyek ini menggunakan cloud storage atau media penyimpanan berbasis internet. Data yang disimpan di cloud, akan disimpan di server penyedia layanan dan bisa diakses dengan menggunakan koneksi internet.

* *Logical Data Flow*



**Flow Explanation:**

1. Data penjualan dan transaksi harian di ekstraksi dari source system
2. Kemudian data tersebut ditransformasikan sebelum di load ke dalam data warehouse. Dalam proses ini dilakukan pembersihan data untuk data yang tidak konsisten dan bernilai null, memasang aturan penginputan data, menghilangkan data yang terduplikasi, mengurutkan data sesuai dengan jenisnya
3. Data yang sudah melewati proses cleaning tadi selanjutnya akan di load masuk ke dalam data warehouse dan siap untuk digunakan
4. Selanjutnya data yang terdapat dalam data warehouse akan dilakukan proses *extraction, load,* dan *transform* dari Data Warehouse menuju Data Mart masing-masing departemen, yang dalam hal ini adalah Data Mart departemen Marketing & Sales
5. Data Mart Marketing & Sales akan di query dan digunakan untuk membuat Dashboard Monitoring dan divisi Marketing & Sales

### Attachment & Link

Berikut adalah daftar link yang digunakan dalam proyek ini :

[link bigquery](https://console.cloud.google.com/bigquery?project=raihanfirdaus111&visibility=&queryText=&ws=!1m14!1m4!4m3!1sraihanfirdaus111!2scurrent_detail_data!3scurrent_detail_data_fix!1m4!4m3!1sraihanfirdaus111!2ssummarized_data!3ssegmentasi_customer!1m3!3m2!1sraihanfirdaus111!2scurrent_detail_data)

## Technical Specification

### Application Server

Data yang digunakan pada proyek ini berasal dari spreadsheet data superstore yang disimpan dalam google drive dan telah dimuat dalam BigQuery. Data superstore yang digunakan merupakan data internal perusahaan yang nantinya akan diinput setiap harinya oleh divisi sales dan marketing. Data location pada proyek ini adalah asia-southeast1

Application server pada project ini ada di BigQuery server: asia-southeast1

### Database Server

Proyek ini menggunakan cloud storage atau media penyimpanan berbasis internet. Data yang disimpan di cloud, akan disimpan di server penyedia layanan dan bisa diakses dengan menggunakan koneksi internet.

Database server pada project ini ada di BigQuery server: asia-southeast1

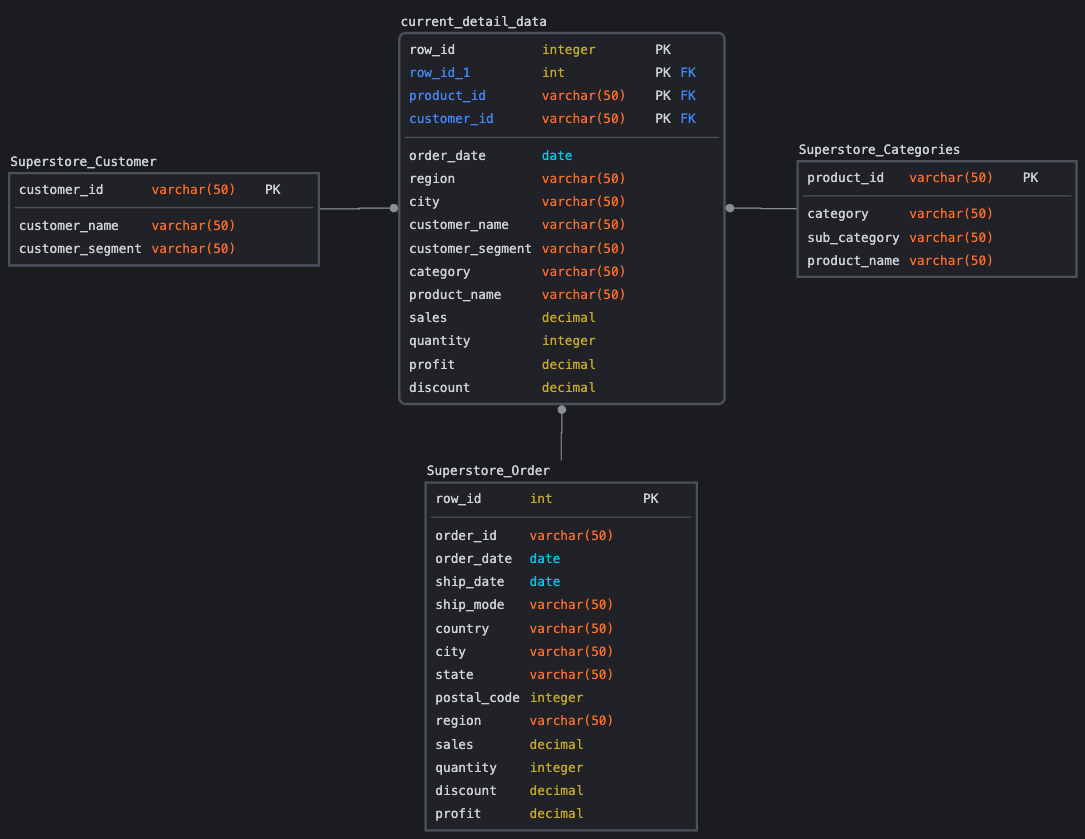
### Server Request

User yang mendapatkan granted ke server adalah sebagai berikut:

| **No** | **Jabatan** | **Department** | **Akses** | **User Name** |
| --- | --- | --- | --- | --- |
| 1 | Head Departement Pusat | Sales & Marketing | full | juragan99 |
| 2 | PIC Departement Pusat | Sales & Marketing | full | esteler77 |
| 3 | Manager per Region | Sales & Marketing | full (hanya region terkait) | TBA |
| 4 | Branch Manager per Kota | Sales & Marketing | full (hanya region terkait) | TBA |
| 5 | BI Analyst | Business Intelligence | Full | Ronaldo7 |
| 6 | BI Engineer | Business Intelligence | Full | Degea1 |

### Database Request

ERD (*entity relationship diagram*) dari dataset project data mart superstore ini dapat dilihat pada gambar berikut :

****

Raw dataset dari project ini adalah:

* Superstore\_Order

Merupakan table fact yang berisi transaksi harian toko superstore. Memiliki Primary key row\_id,

* Superstore\_Categories

Merupakan dim table yang berisi informasi tentang category, subcategory, dan nama produk dari Superstore.

* Superstore\_Customer

Merupakan dim table yang berisi informasi tentang nama dan segmen dari konsumen Superstore.

* Current\_Detail\_Data

Merupakan tabel hasil agregasi dari fact Superstore\_Order dan joined dengan dim Superstore\_Customer & dim Superstore\_Categories

**Current Detail Data**

Current\_detail\_data merupakan table aggregated dari raw table superstore. Semua table di Summarized Data akan diquery dari table ini. Selain itu, table ini juga berguna apabila nantinya user (khususnya BI Analyst) ingin melakukan query sendiri ke data source.

**Query :**

**SELECT**

**order\_date**

**,region**

**,city**

**,category**

**,product\_id**

**,product\_name**

**,customer\_id**

**,customer\_name**

**,sum(sales) as total\_sales**

**,sum(quantity) as total\_qty**

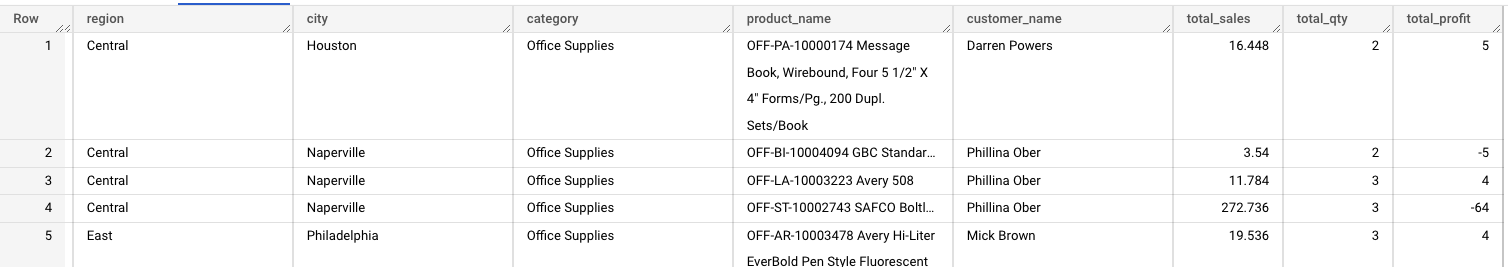
**,sum(profit) as total\_profit**

**from `raw\_table.raw\_table\_clean`**

**group by 1,2,3,4,5,6,7,8**

**order by 1,2,3,4,5,6,7,8**

**Table :**

**

**Summarized Data**

Summarized data merupakan data aggregated yang dibuat khusus sesuai dengan permintaan user. Detail metrics yang ingin ditrack dengan query yang dibuat dapat dilihat pada table berikut:

| **No** | **Merics Name** | **Description** | **Data** | **Formula** |
| --- | --- | --- | --- | --- |
| 1 | Daily reporting | Aggregasi total sales, Aggregasi sales bersih per harian | Dataset: Superstore  Table:  \*Superstore\_order  \*Superstore\_category  \*superstore\_customer | *lihat section Technical Dependencies (a)* |
| 2 | Monthly reporting | Aggregasi total sales, Aggregasi sales bersih per bulan | *lihat section Technical Dependencies (b)* |
| 3 | Quarterly reporting | Aggregasi total sales, Aggregasi sales bersih per 3 bulanan | *lihat section Technical Dependencies (c)* |
| 4 | Yearly reporting | Aggregasi total sales, Aggregasi sales bersih per tahun | *lihat section Technical Dependencies (d)* |
| 5 | Actual vs Target | Ratio pencapaian penjualan actual vs budget ( dengan asumsi target tahunan = sales tahun lalu + (sales tahun lalu \* 10%) | *lihat section Technical Dependencies (e)* |
| 6 | Current Month vs Last Month Growth | % growth bulan ini vs bulan lalu | *lihat section Technical Dependencies (f)* |
| 7 | Current Quarter vs Last Quarter Growth | % growth 3-bulan ini vs 3-bulan lalu | *lihat section Technical Dependencies (g)* |
| 8 | Current Year vs Last Year Growth | % growth tahun ini vs tahun lalu | *lihat section Technical Dependencies (h)* |
| 9 | Loss reporting | Laporan kerugian per produk dan per toko cabang (monthly, quarterly dan yearly) dalam | *lihat section Technical Dependencies (i)* |
| 10 | Segmentasi Customer | Laporan segmentasi customer yang berbelanja: Bronze < USD 200 , USD 200 < Silver < USD 500 , Gold > USD 500 /3 bulan. | *lihat section Technical Dependencies (j)* |
| 11 | Segmentasi Product | Laporan segmentasi produk dengan penjualan : 3nd Product < 5 , 5 < 2nd Product < 10 , 1st Product > 10 /bulan secara per toko dan per regional. | *lihat section Technical Dependencies (k)* |
| 12 | Segmentasi Toko Cabang | Kemampuan segmentasi toko cabang dengan omzet : Kategori I < USD 1000 , USD 1000 < Kategori II < USD 2000 , Kategori III > USD 2000 /bulan | *lihat section Technical Dependencies (l)* |

### Access Request

Detail user yang dapat mengakses database ini dapat dilihat pada tabel berikut.

| **No** | **Jabatan** | **Department** | **Akses** | **User Name** |
| --- | --- | --- | --- | --- |
| 1 | Head Departement Pusat | Sales & Marketing | full | juragan99 |
| 2 | PIC Departement Pusat | Sales & Marketing | full | esteler77 |
| 3 | Manager per Region | Sales & Marketing | full (hanya region terkait) | TBA |
| 4 | Branch Manager per Kota | Sales & Marketing | full (hanya region terkait) | TBA |
| 5 | BI Analyst | Business Intelligence | Full | Ronaldo7 |
| 6 | BI Engineer | Business Intelligence | Full | Degea1 |

## Resources Requirement

### Department & Staff

Tim yang akan terlibat dalam proyek ini :

1. Semua Manager di masing-masing region
2. Semua Branch Manager di masing-masing kota
3. PIC dari Departemen Marketing & Sales yang bertugas melakukan konsolidasi dan pembuatan report di pusat
4. Head Department Marketing & Sales
5. BI Analyst
6. BI Engineer

### Technical Dependencies

1. Daily reporting

SELECT

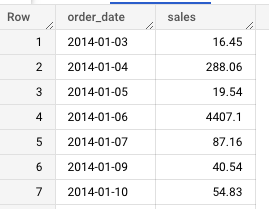
order\_date

,round(sum(total\_sales),2) as sales

from `current\_detail\_data.current\_detail\_data\_fix`

group by 1

order by 1



1. Monthly reporting

SELECT

extract(month from order\_date) as month\_

,extract(year from order\_date) as year\_

,round(sum(total\_sales),2) as sales

from `current\_detail\_data.current\_detail\_data\_fix`

group by 1,2

order by 2,1



1. Quarterly reporting

SELECT

extract(quarter from order\_date) as quarter\_

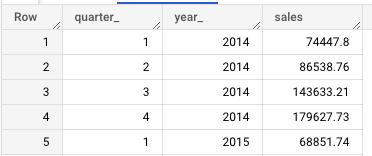
,extract(year from order\_date) as year\_

,round(sum(total\_sales),2) as sales

from `current\_detail\_data.current\_detail\_data\_fix`

group by 1,2

order by 2,1



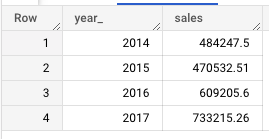
1. Yearly reporting

select \* from `test1.superstore\_order`

where

1=1

and order\_date is not null



1. Actual vs Target

Select

year

,round(sales,2) as sales

,round(LAG(sales) over (partition by 1 order by year) \*1.1,2) as target

from (

SELECT

extract(year from order\_date) as year

,sum(total\_sales) as sales

from `current\_detail\_data.current\_detail\_data\_fix`

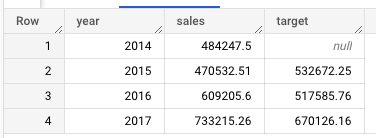
group by 1

)

where

1=1

and year is not null



1. Current Month vs Last Month Growth

Select

month

,year

,round(sales,2) as sales

,growth as current\_month\_growth

,LAG(growth) over (partition by 1 order by year, month) as prev\_month\_growth

from (

Select

month

,year

,sales

,round((sales - (LAG(sales) over (partition by 1 order by year, month)) )/(LAG(sales) over (partition by 1 order by year, month)),2)as growth

from (

SELECT

extract(month from order\_date) as month

,extract(year from order\_date) as year

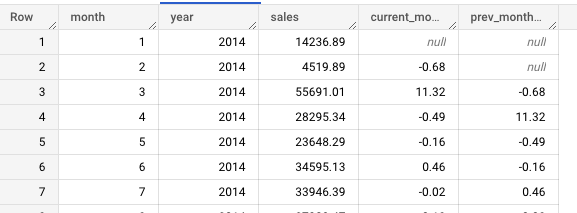
,sum(total\_sales) as sales

from `current\_detail\_data.current\_detail\_data\_fix`

group by 1,2

) p ) k

order by 2,1



1. Current Quarter vs Last Quarter Growth

Select

quarter

,year

,round(sales, 2) as sales

,growth as current\_quarter\_growth

, LAG(growth) over (partition by 1 order by year, quarter) as prev\_quarter\_growth

from (

Select

quarter

,year

,sales

,round((sales - (LAG(sales) over (partition by 1 order by year, quarter)) )/(LAG(sales) over (partition by 1 order by year, quarter)),2)as growth

from (

SELECT

extract(quarter from order\_date) as quarter

,extract(year from order\_date) as year

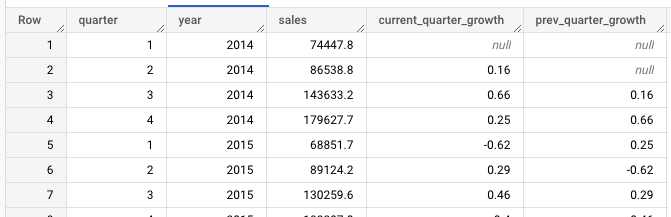
,round(sum(total\_sales),1) as sales

from `current\_detail\_data.current\_detail\_data\_fix`

group by 1,2

) p ) k

order by 2,1



1. Current Year vs Last Year Growth

Select

year

,round(sales, 2) as sales

,growth as current\_year\_growth

, LAG(growth) over (partition by 1 order by year) as prev\_year\_growth

from (

Select

year

,sales

,round((sales - (LAG(sales) over (partition by 1 order by year)) )/(LAG(sales) over (partition by 1 order by year)),2)as growth

from (

SELECT

extract(year from order\_date) as year

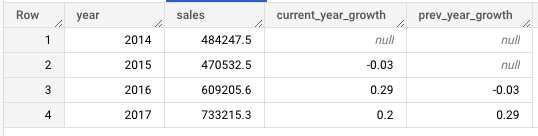
,round(sum(total\_sales),1) as sales

from `current\_detail\_data.current\_detail\_data\_fix`

group by 1

) p ) k

order by 1



1. Loss reporting

\***Yearly**

**select**

**extract(year from order\_date) as year**

**,category**

**,round(sum(total\_profit),1) as loss**

**from `current\_detail\_data.current\_detail\_data\_fix`**

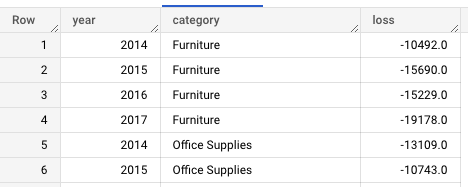
**where**

**1=1**

**and total\_profit <0**

**group by 1,2**

**order by 2,1**

****

**\*Monthly**

select

extract(month from order\_date) as month

,extract(year from order\_date) as year

,category

,round(sum(total\_profit),1) as loss

from `current\_detail\_data.current\_detail\_data\_fix`

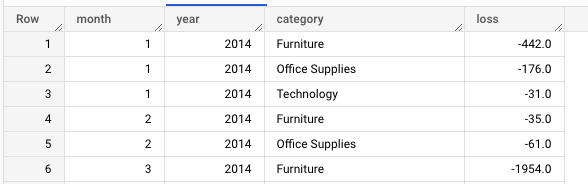
where

1=1

and total\_profit <0

group by 1,2,3

order by 2,1,3



**\*Quarterly**

select

extract(quarter from order\_date) as quarter

,extract(year from order\_date) as year

,category

,round(sum(total\_profit),1) as loss

from `current\_detail\_data.current\_detail\_data\_fix`

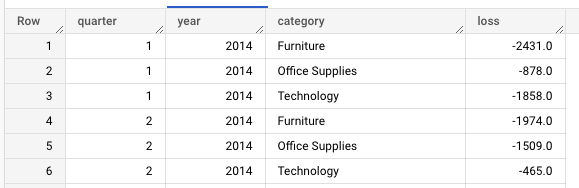
where

1=1

and total\_profit <0

group by 1,2,3

order by 2,1,3



1. Segmentasi Customer

select

quarter

, year

, count(case when purchase <= 200 then customer end) as Bronze

, count(case when purchase > 200 and purchase <= 500 then customer end) as Silver

, count(case when purchase > 500 then customer end) as Gold

from (

select

extract(quarter from order\_date) as quarter

,extract(year from order\_date) as year

,customer\_id as customer

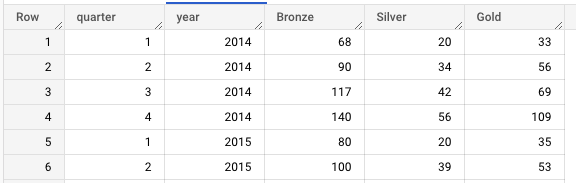
,sum(total\_sales) as purchase

from `current\_detail\_data.current\_detail\_data\_fix`

group by 2,1,3 ) p

group by 1,2

order by 2,1



1. Segmentasi Product

select

month

,year

,region

,city as toko\_cabang

,count(case when sales\_qty <= 5 then product\_name end) as third\_product

,count(case when sales\_qty > 5 and sales\_qty <= 10 then product\_name end) as second\_product

,count(case when sales\_qty > 10 then product\_name end) as first\_product

from (

select

extract(month from order\_date) as month

,extract(year from order\_date) as year

,region

,city

,product\_name

,sum(total\_qty) as sales\_qty

from `current\_detail\_data.current\_detail\_data\_fix`

group by 2,1,3,4,5

)p

group by 1,2,3,4

order by 2,1



1. Segmentasi Toko Cabang

select

quarter

, year

, count(case when omzet <= 1000 then toko\_cabang end) as category1

, count(case when omzet > 1000 and omzet <= 2000 then toko\_cabang end) as category2

, count(case when omzet > 2000 then toko\_cabang end) as category3

from (

select

extract(quarter from order\_date) as quarter

,extract(year from order\_date) as year

,city as toko\_cabang

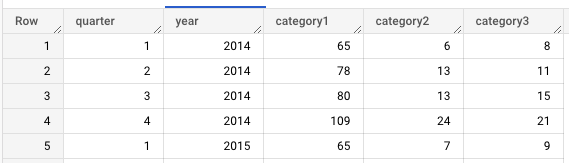
,sum(total\_sales) as omzet

from `current\_detail\_data.current\_detail\_data\_fix`

group by 2,1,3 ) p

group by 1,2

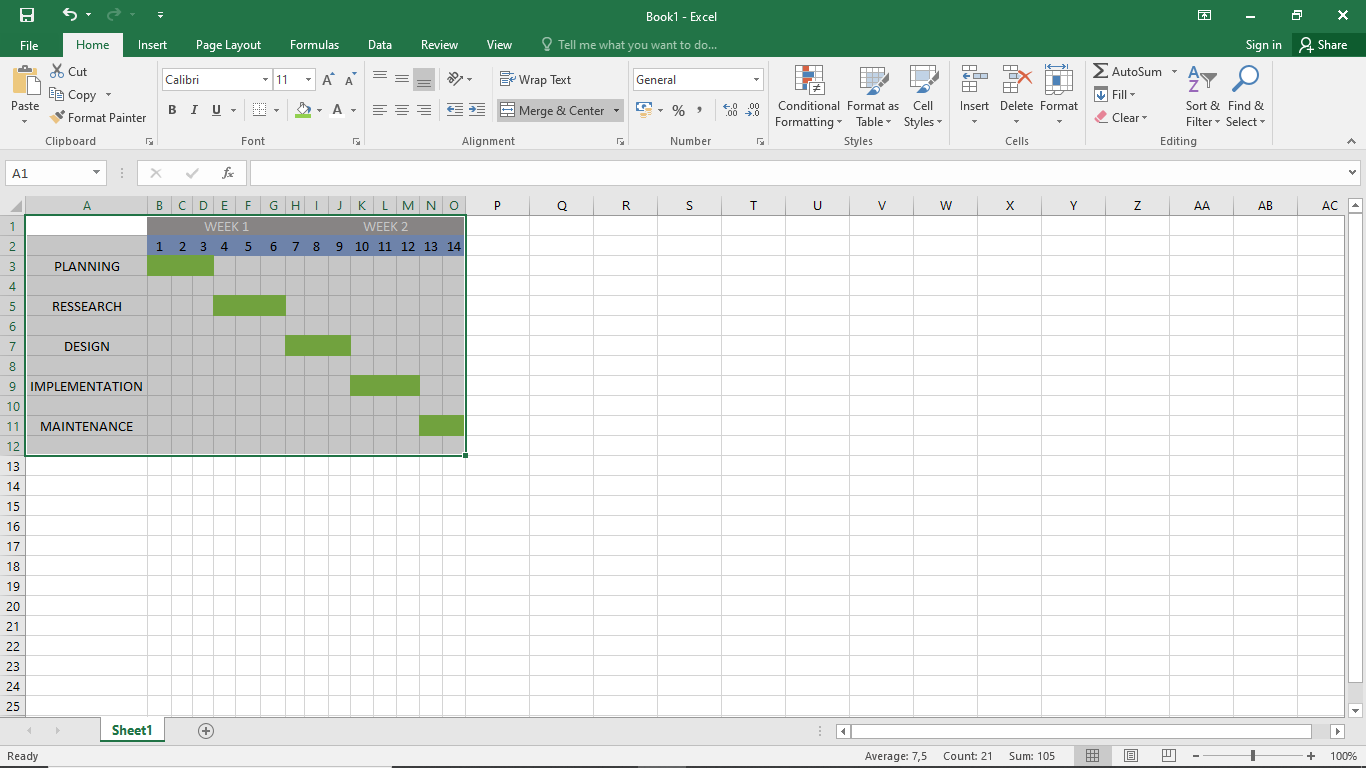
order by 2,1



### Non-Technical Dependencies

1. Proses ETL dilakukan setiap hari pukul 01.00 AM JKT Time, sehingga data yang ditampilkan delay 1 hari dari tanggal sebenarnya.
2. Apabila ingin mendownload ke spreadsheet (xlsx atau csv) dapat dilakukan langsung di BigQuery.

## Timeline & Deadlines



**\*Planning**

1. Perencanaan sumberdaya dan timeline project

Perencanaan awal sumberdaya yang akan digunakan, scope dari project, stakeholder yang akan terkait, serta konsep yang akan dipropose.

1. Sync up dengan semua stakeholder yang terlibat

Diskusi dengan stakeholder-stakeholder yang terkait dengan project ini untuk memastikan semua pihak align dengan planning yang sudah dibuat.

**\*Research**

1. Discuss validasi kebutuhan user

Brainstorming dan diskusi dengan user yang akan membuat dan memakai dashboard Sales & Marketing untuk memastikan data warehouse & data mart yang dibuat dapat mengakomodasi semua kebutuhan user.

1. Technical comparison

Setelah mendapatkan semua gambaran dari user yang akan memakainya, selanjutnya melakukan technical learning dan comparison dari previous project dan senior/lead di BI untuk mendapatkan cara handling dan approach terbaik dari project yang akan dikerjakan

**\*Design**

1. Membuat technical design dan infrastruktur

Membuat blueprint technical design dan infrastruktur yang sudah didiskusikan sebelumnya.

1. Membuat design data flow dan pipeline

Membuat blueprint data flow dan pipeline yang sudah didiskusikan sebelumnya.

1. Membuat design data mart

Membuat blueprint data mart yang sudah didiskusikan sebelumnya.

**\*Implementation**

1. Proses eksekusi planning sesuai dengan technical design

**\*Maintenance**

1. Monitoring etl/elt

Melakukan monitoring etl/elt untuk memastikan flow data yang dibuat sudah benar dan bekerja dengan semestinya.

1. Validasi data output yang dihasilkan

Memastikan data hasil elt/etl yang dilakukan valid dan sesuai dengan yang diharapkan

## Assumptions

Untuk menghemat load data, data yang ada di data mart hanya mencakup 5 tahun terakhir. jika membutuhkan data dengan timeframe lebih lama lagi, silahkan hubungi BI Engineer masing-masing departemen.

## Project Cost

Proyeksi biaya untuk project ini secara internal tidak ada, hanya saja harus membayar biaya subscription Google Bigquery per bulannya.

# Appendices

## List of Acronyms

* ERD : Entity Relationship Diagram
* ELT : Extract Load Transform
* ETL : Extract Transform Load
* BI : Business Intelligence
* PIC : Person in Charge
* BRD: Business Requirement Document
* TRD: Technical Requirement Document

## Glossary of Terms

* Pipeline : Elemen Pemrosesan Data
* Data Flow : Aliran dari sebuah proses informasi
* Data Warehouse : Sistem komputer untuk mengarsipkan dan menganalisa data
* Data Mart : Bagian (subset) data dari warehouse
* Big Query : Cloud data warehouse
* Extract : Proses mengambil dan memilih data dari satu sumber atau lebih
* Transform : Mengubah bentuk data menjadi bentuk yang sesuai kebutuhan
* Load : Memasukkan data ke dalam data warehouse

## Related Documents

* [Business Requirement Document](https://docs.google.com/document/d/1mb4Ii0dJw2qps9OZIXjo6o7vxERaEff_/edit?usp=sharing&ouid=116558742065848056539&rtpof=true&sd=true)
* Spreadsheet Categories <https://docs.google.com/spreadsheets/d/1nsW1y1xP5zM5lmKINMFLFeDu6KNELKbWJ8dY3eYeJAE/edit#gid=767970689>
* Spreadsheet Customer <https://docs.google.com/spreadsheets/d/1Vq6go54q06VcX0Aw_SmJS_WtPlC0hJ0eslrraTZXVu8/edit#gid=1379107214>
* Spreadsheet Order <https://docs.google.com/spreadsheets/d/1tBZtLja4DTHlvqemYlHa9yc5ShFczM4FlTcrD-6bcQE/edit#gid=633794958>