|  |  |
| --- | --- |
| **Practicum Case** |  |
| COMP6579  Big Data Processing |
| **Computer Science** | **E201-COMP6579-DD01-03** |
| ***Valid on*** *Odd Semester Year 2020/2021* | **Revision 00** |

## Learning Outcomes

* Understand Big Data Architecture Layers
* Understand Big Data Analytics and Visualizations

## Topic

* Session 03 - Big Data Management System I

## Subtopics

* Data Ingestion using Sqoop
* Querying Relational Data

## Soal

*Case*

**Ramen Shop**

**Ramen Shop** is a shop located in Jakarta. Since the sales are going high, they need to improve the store sales more quickly. To do that they intended to do some **analysis** from **different** **kinds** of data they have.

From the sales business process, the data can be analyzed to gain sales insight. The data is stored in **Comma-Separated** **Values (CSV)** file and **MySQL** dump file and the data schema is drawn using **Entity Relationship Diagram (ERD)** below:

*A screenshot of a social media post

Description automatically generated*

**Figure 1. Ramen Shop ERD**

You were given the task to gain some insight from the **sales** data using **Hadoop** tools. Below is the task you must do:

# **Load data from CSV to Hive**

Given the file “**MsCustomer**.**csv**” and “**MsRamen**.**csv**”, you were asked to load the data from **Comma-Separated Values** (**CSV**) file to **Hive** for data integration.

# **Load data from MySQL to Hive**

Given the file “**create+insert.sql**” that consists of the data about **brand**, **country**, **ramenstyle,** and **transaction**. You need to load the data to **MySQL** database, then **ingest** the data from **MySQL** database to **Hive** for data integration.

# **Query Analysis**

From the data in **Hive**, you need to gain some sales insight about the data. Below are some statements you need to answer using **Hive** / **Impala** query:

* 1. Show **top 5** **customer** which bought the **most ramen in 2020**
  2. Show **ramen** which **quantity** have been **sold greater than 100** and ramen styleis **'Cup'**
  3. Show **country** that distributed **more than 20 ramen** and **ramen** **price is** **above 30000**
  4. Show **ramen** that has been **bought** **more than the averag**e ramen transaction
  5. Show **ramen**, **price level**, and **total ramen sold** where ramen brand is **established after 2012** and **total ramen sold** is **above 50.** The price level will be determined with the following below table:

|  |  |
| --- | --- |
| Ramen Price | Price Level |
| < 15000 | Cheap |
| >= 15000 and < 30000 | Moderate |
| >= 30000 | Expensive |

**Please ask your teaching assistant if there are any related questions.**

**--1**

**SELECT customername,sum(quantity) FROM mscustomer**

**JOIN transactionheader ON cast(mscustomer.customerid AS INT) = transactionheader.customerid**

**JOIN transactiondetail ON transactionheader.transactionid = transactiondetail.transactionid**

**WHERE year(transactiondate) = 2020**

**GROUP BY customername**

**ORDER BY sum(quantity) DESC**

**LIMIT 5**

**--2**

**SELECT a.ramenname,sum(c.quantity) as SoldQuantity**

**FROM msramen a**

**JOIN msramenstyle b**

**ON b.ramenstyleid=a.ramenstyleid**

**AND b.ramenstylename='Cup'**

**JOIN transactiondetail c**

**ON c.ramenid=a.ramenid**

**GROUP BY a.ramenname**

**HAVING sum(c.quantity)>100**

**--3**

**SELECT countryname,count(msramen.ramenid) FROM mscountry**

**JOIN msramen ON cast(mscountry.countryid AS INT)= cast(msramen.countryid AS INT)**

**WHERE msramen.ramenprice > 30000**

**GROUP BY countryname**

**HAVING count(msramen.ramenid) > 20**

**--4**

**SELECT a.ramenname,sum(b.quantity) as QuantityRamen**

**FROM**

**(**

**SELECT a.ramenid,a.ramenname,sum(b.quantity) as QuantityRamen**

**FROM msramen a**

**JOIN transactiondetail b**

**ON a.ramenid = b.ramenid**

**GROUP BY a.ramenid,a.ramenname**

**) a**

**JOIN transactiondetail b**

**ON a.ramenid=b.ramenid**

**GROUP BY a.ramenname**

**HAVING**

**sum(b.quantity)>avg(a.QuantityRamen)**

**--5**

**SELECT a.ramenname,(**

**CASE**

**WHEN a.ramenprice>=30000 THEN 'Expensive'**

**WHEN a.ramenprice>=15000 THEN 'Moderate'**

**ELSE 'CHEAP'**

**END**

**) as pricelevel**

**,sum(c.quantity) as totalramensold**

**FROM msramen a**

**JOIN msbrand b**

**ON cast(a.brandid AS INT)=cast(b.brandid AS INT)**

**AND b.establishedyear>2012**

**JOIN transactiondetail c**

**on c.ramenid=a.ramenid**

**GROUP BY a.ramenname,pricelevel**

**HAVING sum(c.quantity) > 50**