***SOAL***

* 1. **Load data from CSV to Hive**

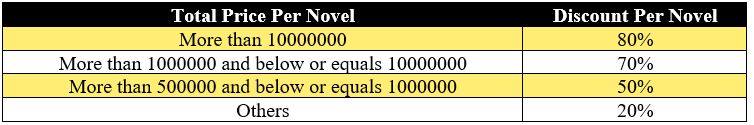
Given the file “**Novel**.**csv**” and “**Writer**.**csv**”, you were asked to load the data from **Comma-Separated Values** (**CSV**) file to **Hive** for data integration.

1. **Load data from MySQL to Hive**

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

1. **Query Analysis**

From the data in **Hive**, you need to gain some sales insight in **Roman Store**, below are some statements you need to answer using **Hive** / **Impala** query:

* 1. Showthe **number of books sold** for **five oldest novels** writtenby **Raymund Shmyr.**
  2. Show **top 3 writers** who wrote the most books and was published **5 years ago.**
  3. Show **top 3 female customers** that do the **most transaction**.
  4. Show **male staff** who have **sold** **books** **more than the average**.
  5. Show **novel** and its **total price after discount per transaction** for novel that is published **after 1999** and the novel name must contain **at least 3 words**. The **discount per novel** will be based from its **total price** with the following condition:

***JAWABAN***

1. Proses pembuatan Databases

CREATE DATABASE RomanNovelStore

CREATE EXTERNAL TABLE Novel

(

NovelID VARCHAR (255),

NovelName VARCHAR (255),

WriterID VARCHAR (255),

PublishDate TIMESTAMP,

Price INT

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ';'

STORED AS TEXTFILE

TBLPROPERTIES ("skip.header.line.count"="1")

CREATE EXTERNAL TABLE Writer

(

WriterID VARCHAR (255),

WriterName VARCHAR (255)

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ';'

STORED AS TEXTFILE

TBLPROPERTIES ("skip.header.line.count"="1")

Proses memasukan data .CSV ke HIVE

(Bagian terminal)

(base) [cloudera@quickstart ~]$ hadoop ls -copyFromLocal Desktop/data

Error: Could not find or load main class ls

(base) [cloudera@quickstart ~]$ hadoop fs -copyFromLocal Desktop/data

(base) [cloudera@quickstart ~]$ hadoop fs -ls

Found 3 items

drwxrwxrwx - cloudera cloudera 0 2020-10-19 10:00 Data

drwxr-xr-x - cloudera cloudera 0 2020-10-20 01:48 data

drwxr-xr-x - cloudera cloudera 0 2020-02-07 20:15 tmp

(base) [cloudera@quickstart ~]$ hadoop fs -chmod 777 /user/cloudera/data

(base) [cloudera@quickstart ~]$ hadoop fs -ls

Found 3 items

drwxrwxrwx - cloudera cloudera 0 2020-10-19 10:00 Data

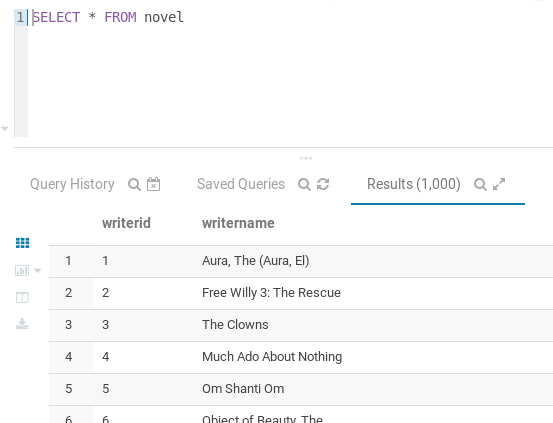
drwxrwxrwx - cloudera cloudera 0 2020-10-20 01:48 data

drwxr-xr-x - cloudera cloudera 0 2020-02-07 20:15 tmp

(Bagian Hive)

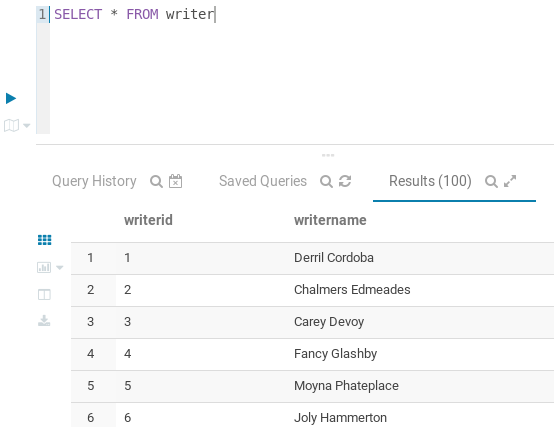
LOAD DATA INPATH '/user/cloudera/data/Novel.csv' INTO TABLE novel





LOAD DATA INPATH '/user/cloudera/data/Writer.csv' INTO TABLE writer





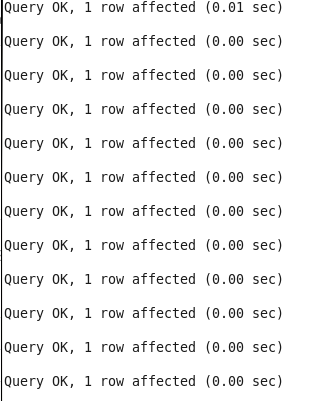
1. Proses pembuatan Databases di MySQL (Terminal)

(base) [cloudera@quickstart ~]$ mysql -u root -p

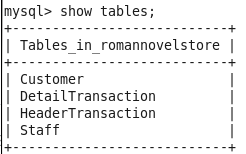
mysql> create database romannovelstore;

mysql> use romannovelstore;

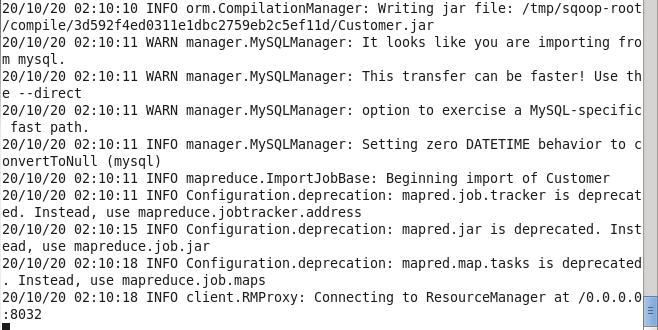
mysql> source Desktop/data/create+insert.sql;

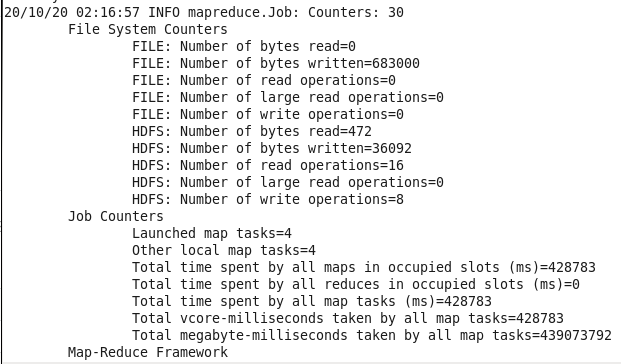


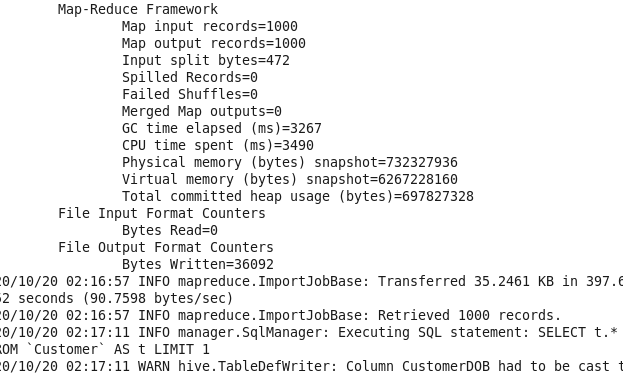
mysql> show tables;



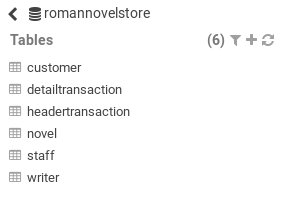
(base) [cloudera@quickstart ~]$ sudo sqoop import-all-tables --connect jdbc:mysql://quickstar:3306/romannovelstore --username=root -P --hive-import --hive-database=romannovelstore











1. Query :
2. Showthe **number of books sold** for **five oldest novels** writtenby **Raymund Shmyr.**

SELECT Nov.novelname,Nov.publishdate

FROM novel Nov

JOIN writer Wri

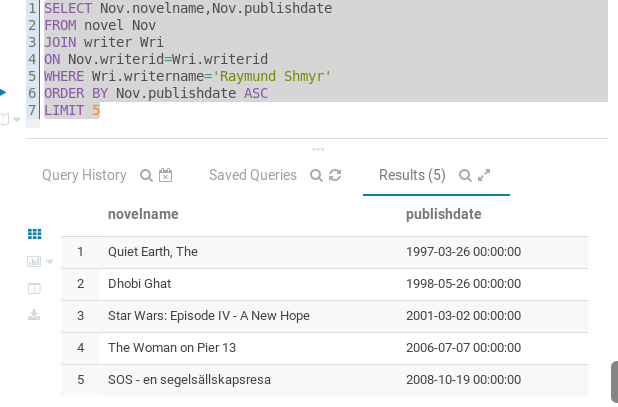
ON Nov.writerid=Wri.writerid

WHERE Wri.writername='Raymund Shmyr'

ORDER BY Nov.publishdate ASC

LIMIT 5

Proof :



1. Show **top 3** writers who wrote the most books and was published **5 years ago.**

SELECT Wri.writername, count(Nov.writerid) AS BookSell

FROM novel Nov

JOIN writer Wri

ON Nov.writerid=Wri.writerid

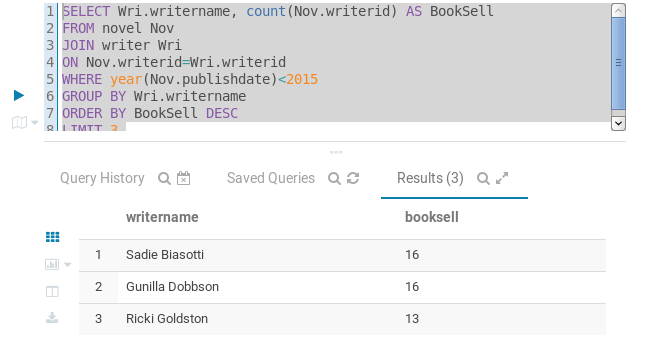
WHERE year(Nov.publishdate)<2015

GROUP BY Wri.writername

ORDER BY BookSell DESC

LIMIT 3

Proof:



1. Show **top 3 female customers** that do the **most transaction**.

SELECT Cus.customername, count(Hea.customerid) AS BookBuy

FROM customer Cus

JOIN headertransaction Hea

ON Cus.customerid=Hea.customerid

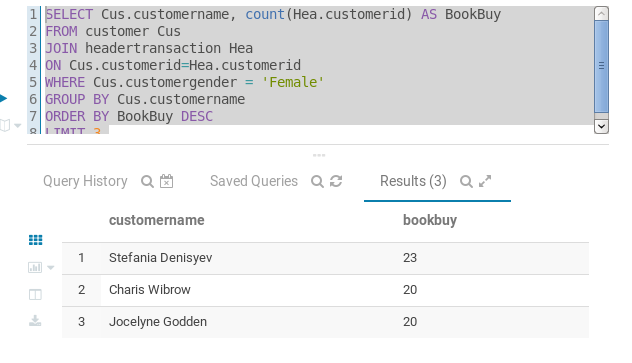
WHERE Cus.customergender = 'Female'

GROUP BY Cus.customername

ORDER BY BookBuy DESC

LIMIT 3

Proof:



1. Show **male staff** who have **sold** **books** **more than the average**.

SELECT Sta.staffname, count(Hea.staffid) AS StaffBookSell

FROM staff Sta

JOIN headertransaction Hea

ON Sta.staffid=Hea.staffid

WHERE Sta.staffgender = 'Male'

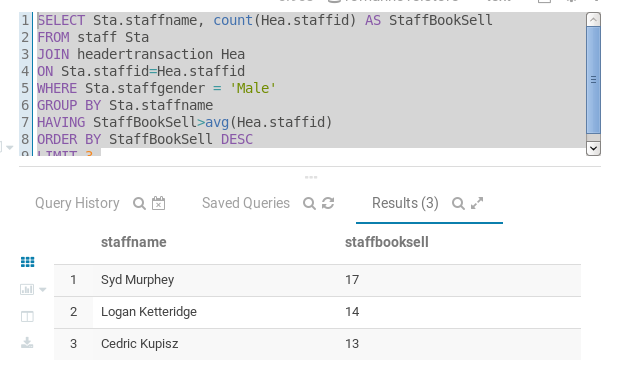
GROUP BY Sta.staffname

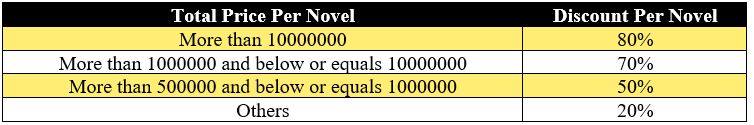
HAVING StaffBookSell>avg(Hea.staffid)

ORDER BY StaffBookSell DESC

LIMIT 3

Proof:



1. Show **novel** and its **total price after discount per transaction** for novel that is published **after 1999** and the novel name must contain **at least 3 words**. The **discount per novel** will be based from its **total price** with the following condition:

(Tanpa Sort)

SELECT Nov.novelname,

(

CASE

WHEN Nov.price >10000000 THEN Nov.price\*80/100

WHEN Nov.price >1000000 AND Nov.price <=10000000 THEN Nov.price\*70/100

WHEN Nov.price >500000 AND Nov.price <=1000000 THEN Nov.price\*50/100

ELSE Nov.price\*20/100

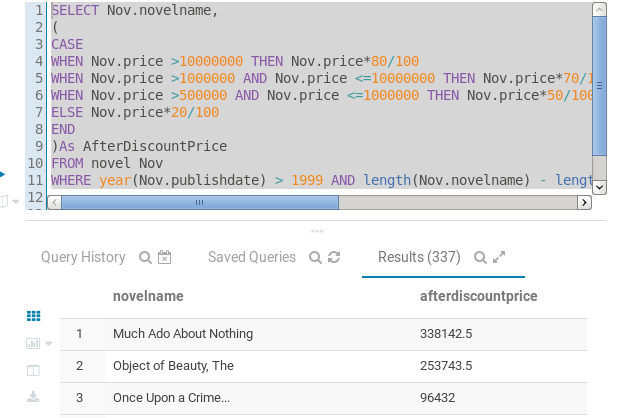
END

)As AfterDiscountPrice

FROM novel Nov

WHERE year(Nov.publishdate) > 1999 AND length(Nov.novelname) - length(regexp\_replace(Nov.novelname,' ','')) + 1 >3

Proof:



(Dengan Sort (Price))

SELECT Nov.novelname,

(

CASE

WHEN Nov.price >10000000 THEN Nov.price\*80/100

WHEN Nov.price >1000000 AND Nov.price <=10000000 THEN Nov.price\*70/100

WHEN Nov.price >500000 AND Nov.price <=1000000 THEN Nov.price\*50/100

ELSE Nov.price\*20/100

END

)As AfterDiscountPrice

FROM novel Nov

WHERE year(Nov.publishdate) > 1999 AND length(Nov.novelname) - length(regexp\_replace(Nov.novelname,' ','')) + 1 >3

ORDER BY AfterDiscountPrice DESC

Proof:



(Dengan Sort (Nama))

SELECT Nov.novelname,

(

CASE

WHEN Nov.price >10000000 THEN Nov.price\*80/100

WHEN Nov.price >1000000 AND Nov.price <=10000000 THEN Nov.price\*70/100

WHEN Nov.price >500000 AND Nov.price <=1000000 THEN Nov.price\*50/100

ELSE Nov.price\*20/100

END

)As AfterDiscountPrice

FROM novel Nov

WHERE year(Nov.publishdate) > 1999 AND length(Nov.novelname) - length(regexp\_replace(Nov.novelname,' ','')) + 1 >3

ORDER BY Nov.novelname ASC

Proof:

