Rapport de projet du cours de :

Big DATA

Présenté par :

Oumar FALL

Élève ingénieur en ing3 génie informatique 2023-2024

Aout 09, 2024

Responsable du cours : Monsieur Djibril MBOUP

Rapport de Projet: Ingestion et Traitement de Données dans Big Data

Cours : Big Data

Classe : IPSL

Auteur : Dr. Djibril MBOUP

Date : Avant le 27/07/2024

Introduction

Ce projet consiste en l'ingestion et le traitement de données dans un environnement Big Data, en utilisant Apache Sqoop pour l'importation des données et Apache Hive pour leur transformation et analyse. Les données utilisées proviennent d'une base de données MySQL appelée `retail_db`, qui contient des informations sur les ventes d'une entreprise e-commerce.

Partie I: Ingestion des données avec Apache Sqoop

1. Préparation de l'environnement

Avant de commencer l'ingestion des données, nous avons réalisé ces étapes suivantes :
- Télécharger la base de données ‘retail_db.sql’ à partir du lien fourni.
- Configurer un SGBD MySQL/MariaDB sur notre machine.
- Créer un utilisateur et une base de données dans MySQL:

  Voici les commandes :

```
  CREATE user retail_user identified by 'hadoop';
  CREATE database retail_db;
  GRANT ALL ON retail_db.* to retail_dba;
  FLUSH PRIVILEGES
```

- Charger les données dans MySQL:

```
  mysql> source /tmp/retail_db.sql;
```

- Vérifier que les tables ont bien été créées en listant les tables.

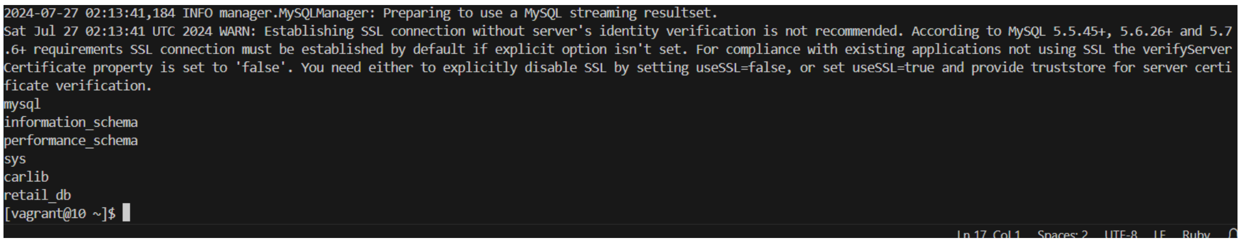
2. Importation des données dans Hive

Utilisez Apache Sqoop pour importer les tables de `retail_db` dans Hive:

```bash
sqoop import \
  --connect "jdbc:mysql://@IP_hostname:3306/retail_db" \
  --username=retail_user \
  --password=hadoop \
```

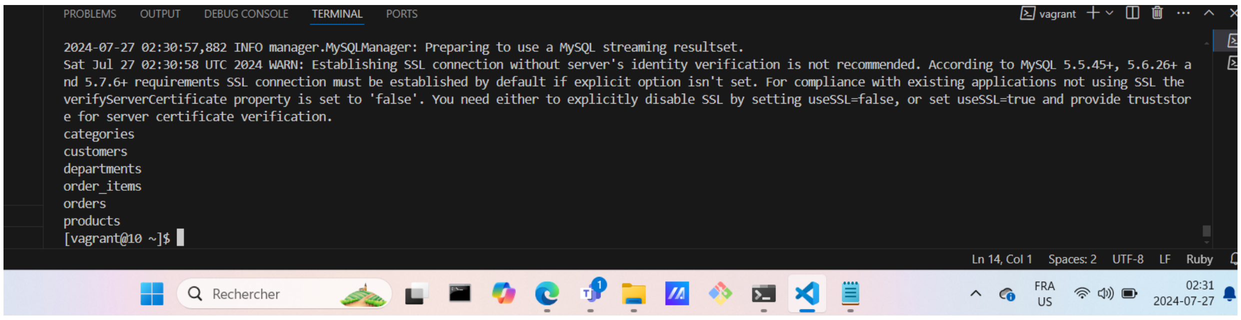
~~~~
  --table tablename \
  --as-parquetfile \
  --target-dir=/user/hive/warehouse/retail_db.db/{tablename} \
  --delete-target-dir
```
~~~~
Vérifiez que les données sont correctement ingérées dans Hive en listant les tables dans Hive:
~~~~
```bash
hive> show tables;
~~~~

Voici nos sorties de commandes :



```
2024-07-27 02:13:41,184 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
Sat Jul 27 02:13:41 UTC 2024 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ and 5.7
.6+ requirements SSL connection must be established by default if explicit option isn't set. For compliance with existing applications not using SSL the verifyServer
Certificate property is set to 'false'. You need either to explicitly disable SSL by setting useSSL=false, or set useSSL=true and provide truststore for server certi
ficate verification.
mysql
information_schema
performance_schema
sys
carlib
retail_db
[vagrant@10 ~]$
```

La liste des tables :



```
2024-07-27 02:30:57,882 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
Sat Jul 27 02:30:58 UTC 2024 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ a
nd 5.7.6+ requirements SSL connection must be established by default if explicit option isn't set. For compliance with existing applications not using SSL the
verifyServerCertificate property is set to 'false'. You need either to explicitly disable SSL by setting useSSL=false, or set useSSL=true and provide truststor
e for server certificate verification.
categories
customers
departments
order_items
orders
products
[vagrant@10 ~]$
```

Partie II: Traitement des données avec Apache Hive

1. Création des Tables dans Hive

Si les tables ne sont pas créées après l'importation, nous pouvons les créer manuellement en utilisant les scripts DDL fournis.

Exemple:
~~~~
```sql
CREATE EXTERNAL TABLE IF NOT EXISTS customers (
  customer_id int,
  customer_fname STRING,
  ...
)
~~~~

```
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS PARQUET
LOCATION 'hdfs:///user/hive/warehouse/retail_db.db/customers'
```

Dans notre cas, les tables ont été déjà crées.

2. Exercice SQL

1. Nombre total de commandes par client en 2014:

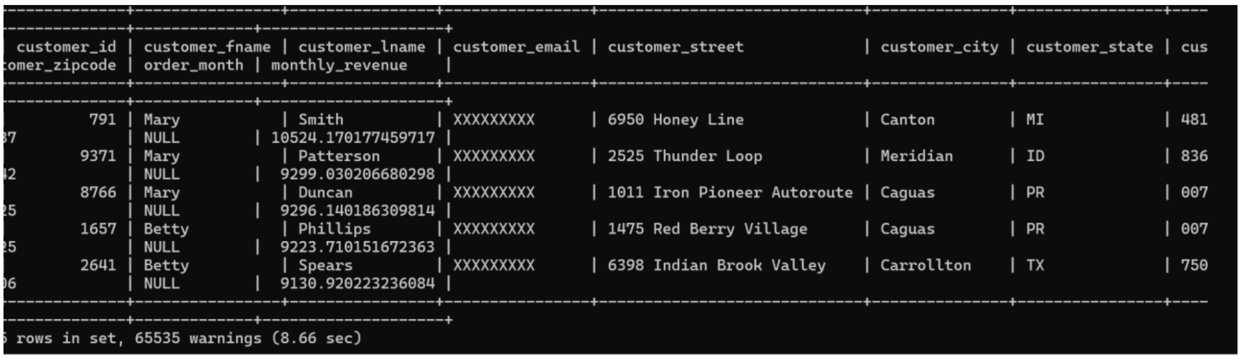
```
select order_id, count(*) as total_commande
from orders where order_status="COMPLET"
and DATE_FORMAT(FROM_UNIXTIME(order_date),'%Y')='2014'
group by order_customer_id;
```

2. Clients sans commande :
```sql
select customer_fname, customer_lname
from customers where customer_id
not in (select order_customer_id from orders)
order by customer_lname, customer_fname;;
```

3. Top 5 clients par revenue mensuel :

```
 WITH MonthlyRevenue AS (

    SELECT
        c.customer_
        id,
        c.customer_
        fname,
        c.customer_
        lname,
        c.customer_
        email,
        c.customer_
        street,
        c.customer_
        city,
        c.customer_
        state,
        c.customer_
        zipcode,
        DATE_FORMAT(FROM_UNIXTIME(o.order_date), '%Y-%m') AS order_month,
        SUM(oi.order_item_subtotal) AS monthly_revenue
    FROM
        cu
    stom
    ers c
    JOIN
        orders o ON c.customer_id =
    o.order_customer_idJOIN
        order_items oi ON o.order_id =
    oi.order_item_order_idGROUP BY
        c.customer_id, order_month
),
RankedReve
    nue AS (
    SELECT
        *,
        ROW_NUMBER() OVER (PARTITION BY order_month ORDER BY
monthly_revenueDESC) AS revenue_rank
    FROM
        MonthlyRevenue
)
SELECT
    customer
    _id,
    customer
    _fname,
    customer
    _lname,
    customer
    _email,
    customer
    _street,
    customer
    _city,
    customer
    _state,
    customer
    _zipcode,
    order_mo
    nth,
    monthly_
    revenue
FROM
    Ranked
Revenue
WHERE
    revenue_r
ank <= 5
ORDER BY
    order_month
    ASC,
    monthly_revenu
    e DESC;
```

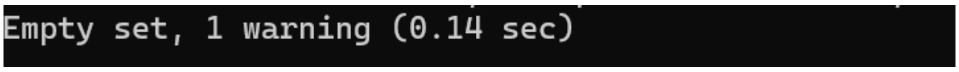


```
--------------+----------------+----------------+----------------+-----------------------------+----------------+-----------------+----
--------------+------------+-------------------+
 customer_id | customer_fname | customer_lname | customer_email | customer_street             | customer_city | customer_state | cus
tomer_zipcode | order_month | monthly_revenue    |
--------------+----------------+----------------+----------------+-----------------------------+----------------+-----------------+----
--------------+------------+-------------------+
        791 | Mary           | Smith          | XXXXXXXXX      | 6950 Honey Line             | Canton        | MI             | 481
37           | NULL        | 10524.170177459717 |
       9371 | Mary           | Patterson      | XXXXXXXXX      | 2525 Thunder Loop           | Meridian      | ID             | 836
42           | NULL        |  9299.030206680298 |
       8766 | Mary           | Duncan         | XXXXXXXXX      | 1011 Iron Pioneer Autoroute | Caguas        | PR             | 007
25           | NULL        |  9296.140186309814 |
       1657 | Betty          | Phillips       | XXXXXXXXX      | 1475 Red Berry Village      | Caguas        | PR             | 007
25           | NULL        |  9223.710151672363 |
       2641 | Betty          | Spears         | XXXXXXXXX      | 6398 Indian Brook Valley    | Carrollton    | TX             | 750
06           | NULL        |  9130.920223236084 |
--------------+----------------+----------------+----------------+-----------------------------+----------------+-----------------+----
--------------+------------+-------------------+
5 rows in set, 65535 warnings (8.66 sec)
```

4. Trouver toutes les commandes terminées ou fermées (completed ou closed),

puiscalculez le revenu total pour chaque jour pour chaque département. La sortie doit afficher : order_date, department_name et order_revenue

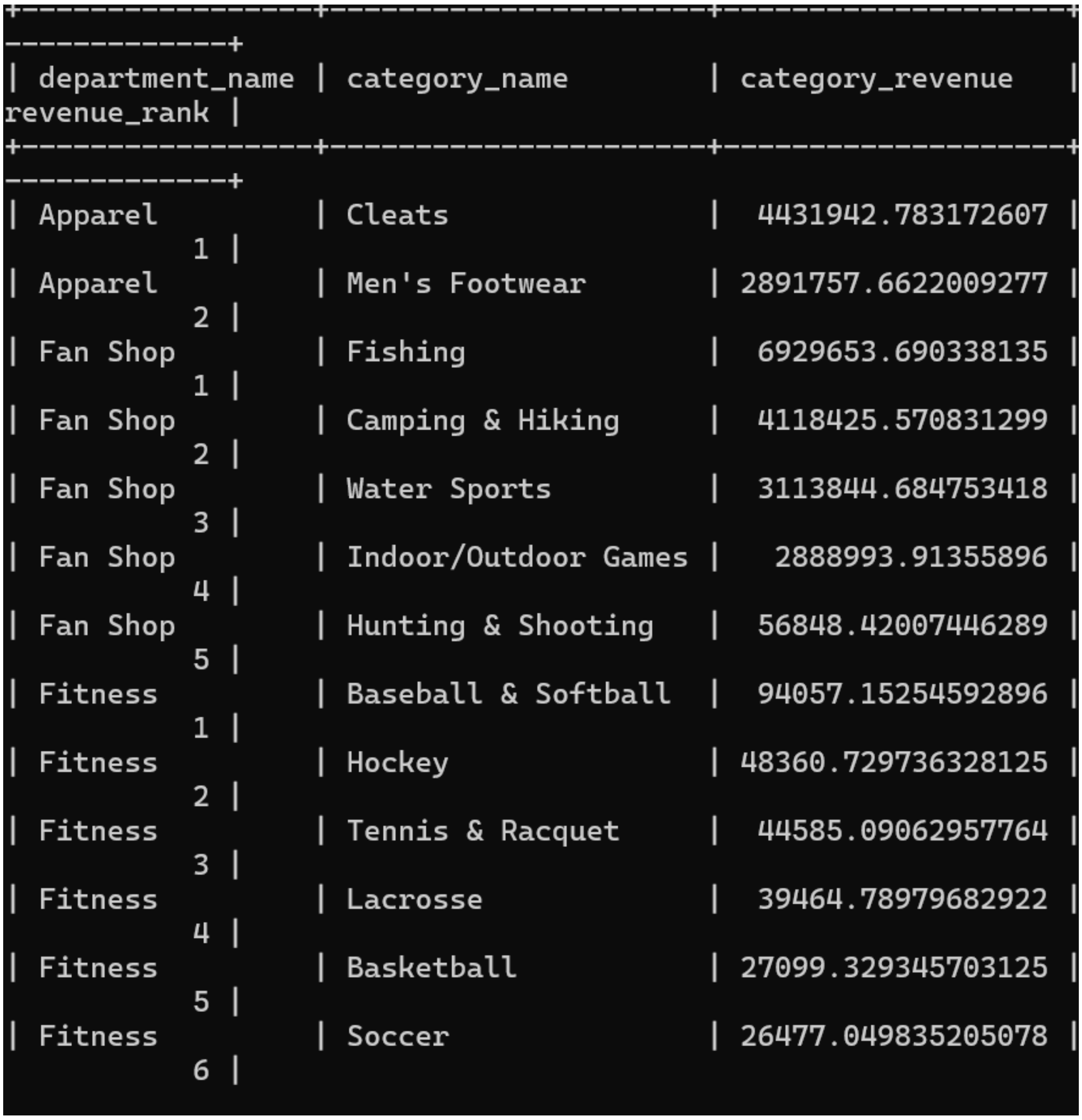
```
SELECT DATE(FROM_UNIXTIME(o.order_date)) AS order_date,
  d.department_name, SUM(oi.order_item_subtotal) AS order_revenue
FROM orders o
JOIN order_items oi ON o.order_id = oi.order_item_order_id
JOIN products p ON oi.order_item_product_id = p.product_id
JOIN categories c ON p.product_category_id = c.category_id
JOIN departments d ON c.category_department_id = d.department_id
WHERE o.order_status IN ('COMPLET', 'FERMÉ')
  GROUP BY
```



```
Empty set, 1 warning (0.14 sec)
```

1. 5. Trouver le rank de chaque catégorie par revenue obtenue dans chaque départementà partir de toutes les transactions. Affichez les résultats par department_name et classez-les par ordre croissant.

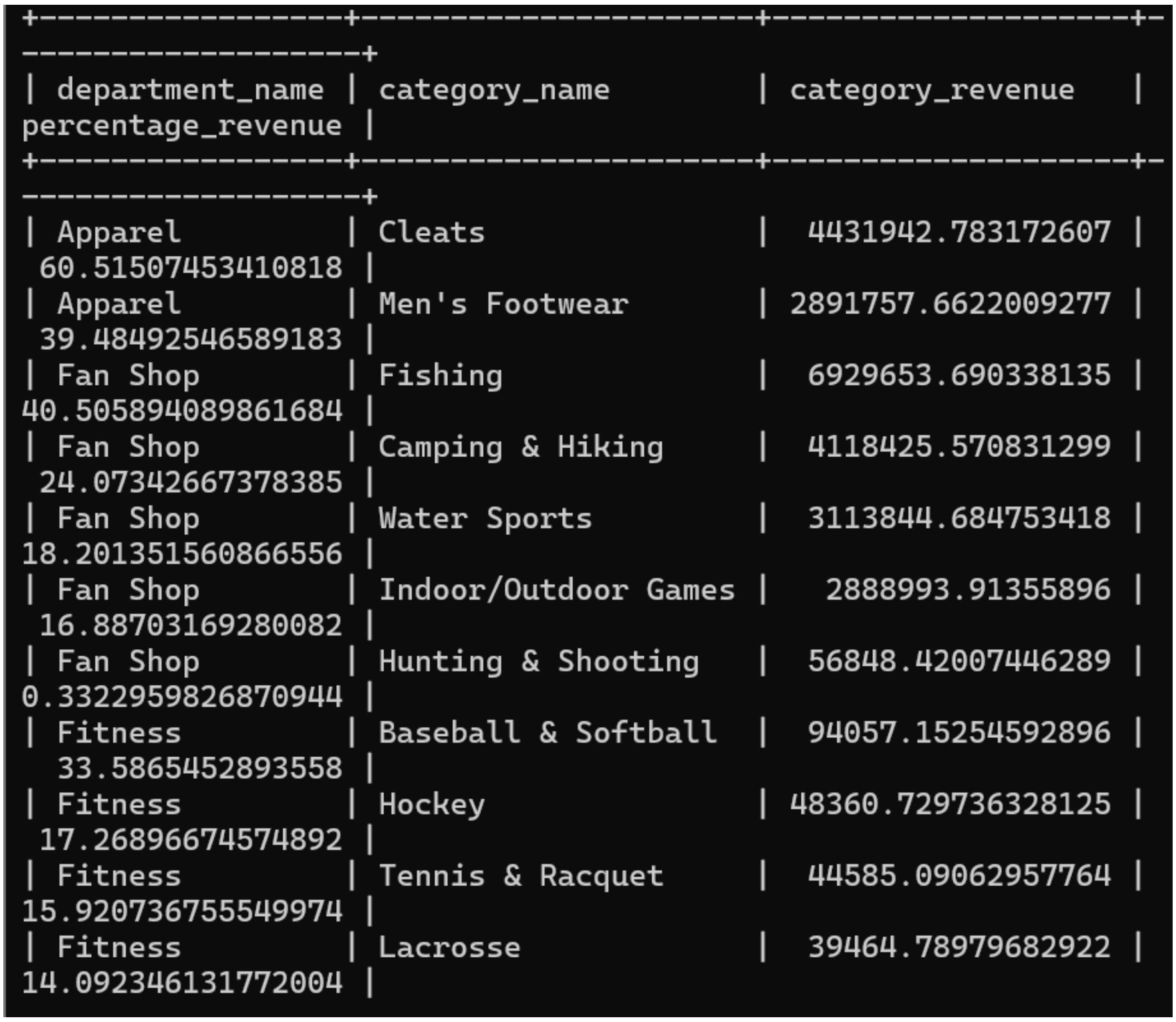
```
WITH CategoryRevenue AS (SELECT d.department_name, c.category_name,
SUM(oi.order_item_subtotal) AS category_revenue FROM order_items oi JOIN products p ON
oi.order_item_product_id = p.product_id JOIN categories c ON p.product_category_id =
c.category_id JOIN departments d ON c.category_department_id = d.department_id GROUP
BY d.department_name, c.category_name), RankedCategoryRevenue AS (SELECT
department_name, category_name, category_revenue, RANK() OVER (PARTITION BY
department_name ORDER BY category_revenue DESC) AS revenue_rank FROM
CategoryRevenue) SELECT department_name, category_name, category_revenue,
revenue_rank FROM RankedCategoryRevenue ORDER BY department_name ASC,
revenue_rank ASC;
```



```
+-----------------+---------------------+-------------------+
--------------+
| department_name | category_name       | category_revenue  |
revenue_rank |
+-----------------+---------------------+-------------------+
--------------+
| Apparel         | Cleats              |  4431942.783172607 |
           1 |
| Apparel         | Men's Footwear      | 2891757.6622009277 |
           2 |
| Fan Shop        | Fishing             |  6929653.690338135 |
           1 |
| Fan Shop        | Camping & Hiking    |  4118425.570831299 |
           2 |
| Fan Shop        | Water Sports        |  3113844.684753418 |
           3 |
| Fan Shop        | Indoor/Outdoor Games |   2888993.91355896 |
           4 |
| Fan Shop        | Hunting & Shooting  |  56848.42007446289 |
           5 |
| Fitness         | Baseball & Softball |  94057.15254592896 |
           1 |
| Fitness         | Hockey              | 48360.729736328125 |
           2 |
| Fitness         | Tennis & Racquet    |  44585.09062957764 |
           3 |
| Fitness         | Lacrosse            |  39464.78979682922 |
           4 |
| Fitness         | Basketball          | 27099.329345703125 |
           5 |
| Fitness         | Soccer              | 26477.049835205078 |
           6 |
```

6. Afficher le pourcentage de chaque catégorie par revenue dans chaque département.Afficher les résultats par department_name et pourcentage par ordre décroissant.

```
WITH DepartmentTotalRevenue AS (SELECT d.department_name,
SUM(oi.order_item_subtotal) AS total_revenue FROM order_items oi JOIN
products p ON oi.order_item_product_id = p.product_id JOIN categories c ON
p.product_category_id = c.category_id JOIN departments d ON
c.category_department_id = d.department_id GROUP BY d.department_name),
CategoryRevenue AS (SELECT d.department_name, c.category_name,
SUM(oi.order_item_subtotal) AS category_revenue FROM order_items oi JOIN
products p ON oi.order_item_product_id = p.product_id JOIN categories c ON
p.product_category_id = c.category_id JOIN departments d ON
c.category_department_id = d.department_id GROUP BY d.department_name,
c.category_name) SELECT cr.department_name, cr.category_name,
cr.category_revenue, (cr.category_revenue / dtr.total_revenue) * 100 AS
percentage_revenue FROM CategoryRevenue cr JOIN DepartmentTotalRevenue dtr
ON cr.department_name = dtr.department_name ORDER BY cr.department_name
ASC, percentage_revenue DESC;
```



```
+-----------------+----------------------+--------------------+-
-------------------+
| department_name | category_name        | category_revenue   |
percentage_revenue |
+-----------------+----------------------+--------------------+-
-------------------+
| Apparel         | Cleats               |  4431942.783172607 |
 60.51507453410818 |
| Apparel         | Men's Footwear       | 2891757.6622009277 |
 39.48492546589183 |
| Fan Shop        | Fishing              |  6929653.690338135 |
40.505894089861684 |
| Fan Shop        | Camping & Hiking     |  4118425.570831299 |
 24.07342667378385 |
| Fan Shop        | Water Sports         |  3113844.684753418 |
18.201351560866556 |
| Fan Shop        | Indoor/Outdoor Games |   2888993.91355896 |
 16.88703169280082 |
| Fan Shop        | Hunting & Shooting   |  56848.42007446289 |
0.3322959826870944 |
| Fitness         | Baseball & Softball  |  94057.15254592896 |
  33.5865452893558 |
| Fitness         | Hockey               | 48360.729736328125 |
 17.26896674574892 |
| Fitness         | Tennis & Racquet     |  44585.09062957764 |
15.920736755549974 |
| Fitness         | Lacrosse             |  39464.78979682922 |
14.092346131772004 |
```

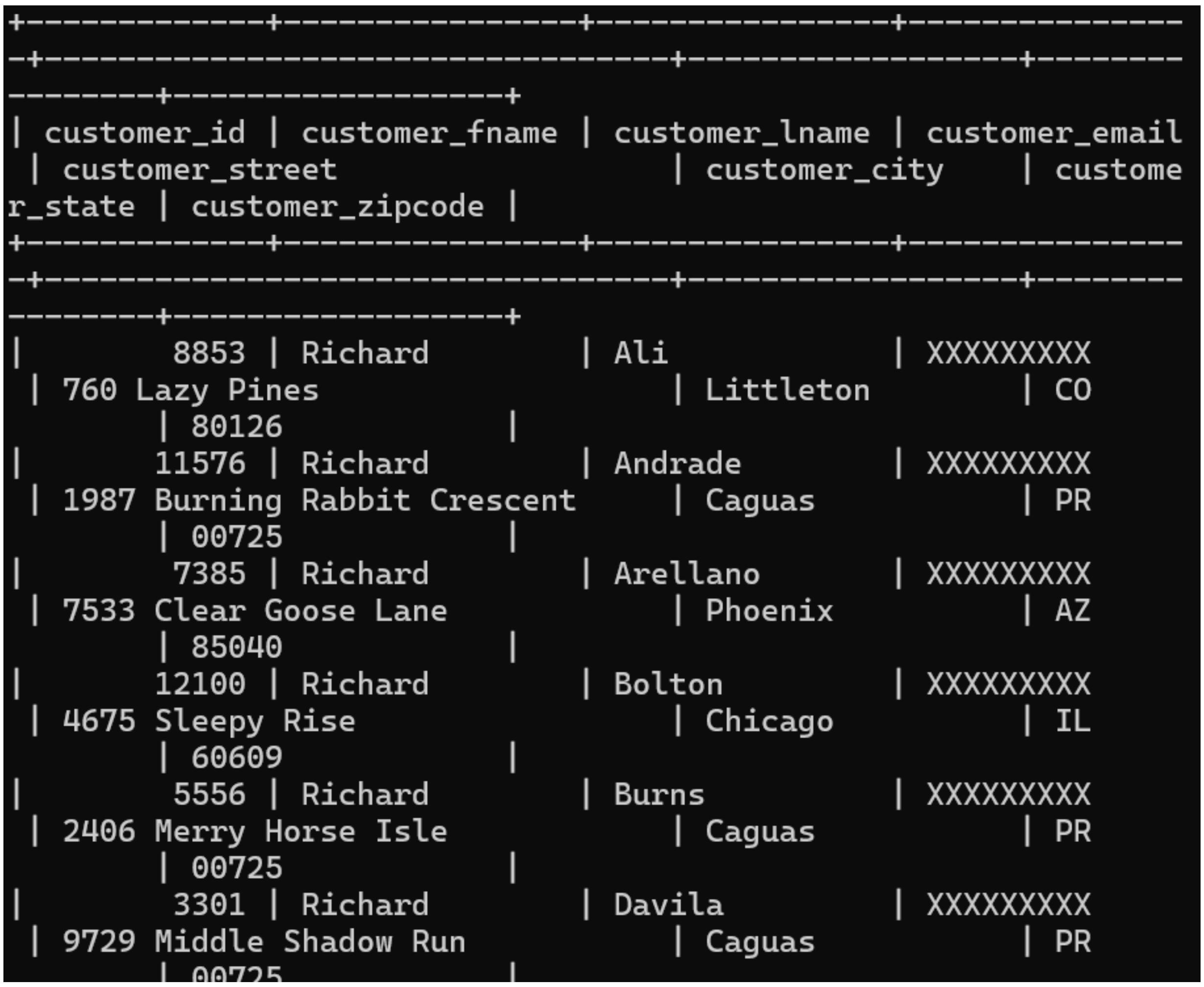
7. Afficher tous les clients qui ont passé une commande d'un montant supérieur à 200 $

```
select customer_fname, customer_lname
from customers c, orders o, order_items ot
where c.customer_id=o.order_customer_id and ot.order_item_order_id=o.order_id
and ot.order_item_subtotal>200;
```

8) Afficher les clients de la "customers" dont les noms customer_fname commence par "Rich"

```
SELECT customer_fname
FROM customers
```

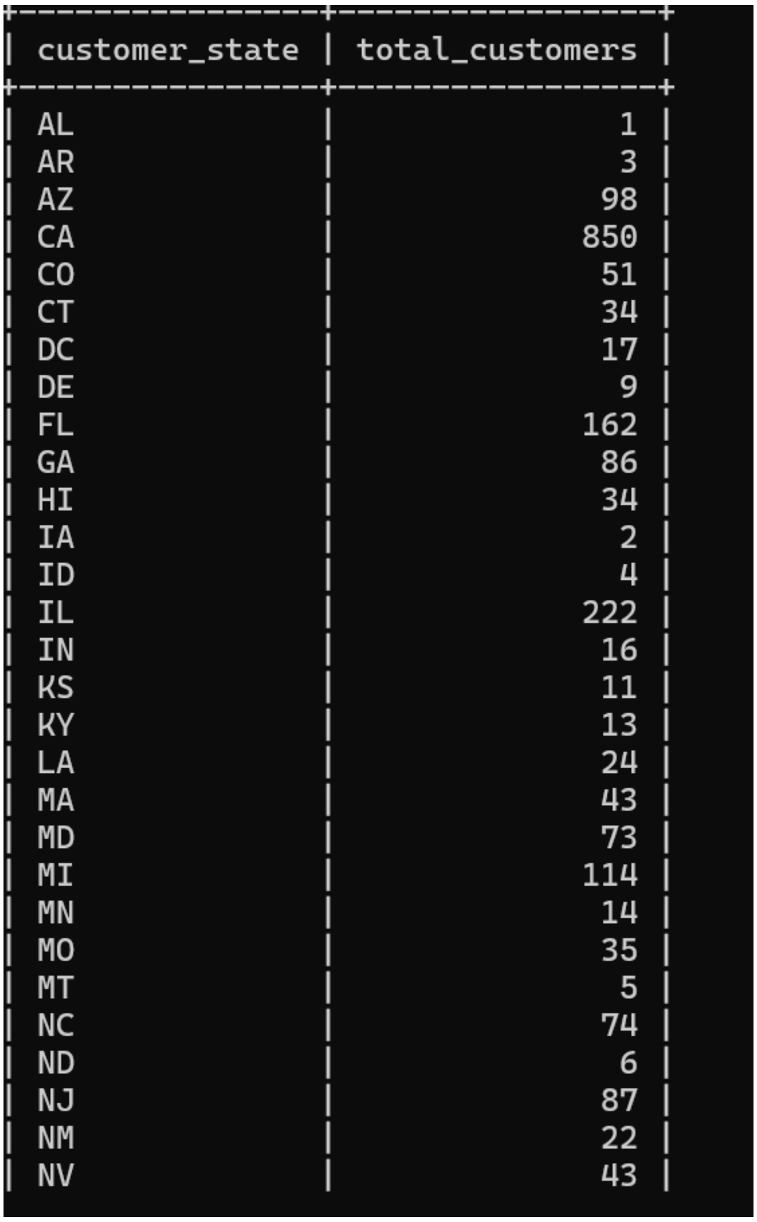
```
WHERE customer_fname like "Rich%" ;
```



```
+-------------+----------------+----------------+---------------
-+-----------------------------+------------------+--------
--------+------------------+
| customer_id | customer_fname | customer_lname | customer_email
 | customer_street             | customer_city    | custome
r_state | customer_zipcode |
+-------------+----------------+----------------+---------------
-+-----------------------------+------------------+--------
--------+------------------+
|        8853 | Richard        | Ali            | XXXXXXXXX
 | 760 Lazy Pines              | Littleton        | CO
        | 80126            |
|       11576 | Richard        | Andrade        | XXXXXXXXX
 | 1987 Burning Rabbit Crescent     | Caguas           | PR
        | 00725            |
|        7385 | Richard        | Arellano       | XXXXXXXXX
 | 7533 Clear Goose Lane       | Phoenix          | AZ
        | 85040            |
|       12100 | Richard        | Bolton         | XXXXXXXXX
 | 4675 Sleepy Rise            | Chicago          | IL
        | 60609            |
|        5556 | Richard        | Burns          | XXXXXXXXX
 | 2406 Merry Horse Isle       | Caguas           | PR
        | 00725            |
|        3301 | Richard        | Davila         | XXXXXXXXX
 | 9729 Middle Shadow Run      | Caguas           | PR
        | 00725            |
```

9) Fournir le nombre total de clients dans chaque état (state) dont le prénom commence par « M ».

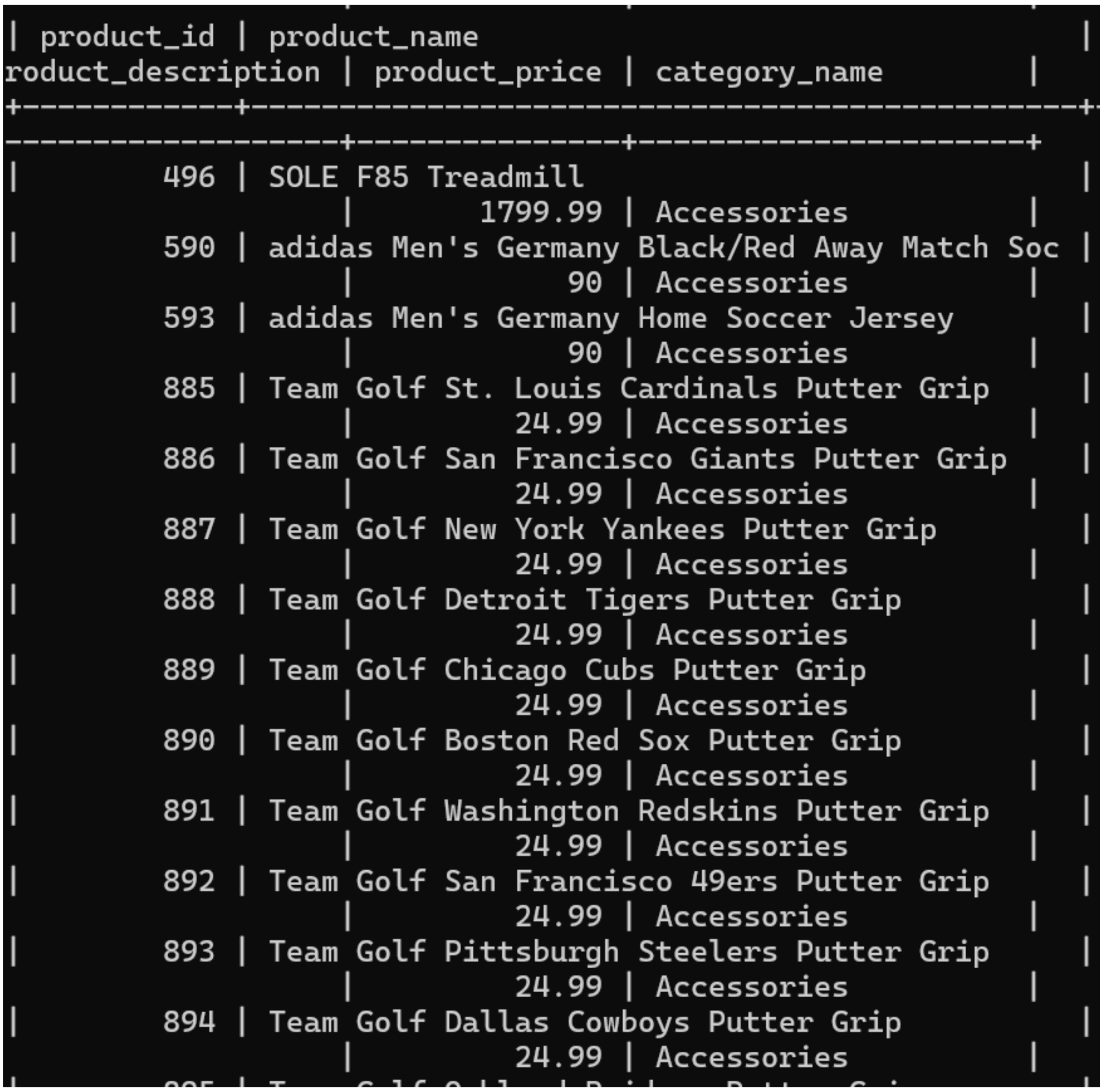
```
select customer_state, count(*) as total_client
from customers
where customer_fname like "M%"
group by customer_state ;
```



```
+----------------+-----------------+
| customer_state | total_customers |
+----------------+-----------------+
| AL             |               1 |
| AR             |               3 |
| AZ             |              98 |
| CA             |             850 |
| CO             |              51 |
| CT             |              34 |
| DC             |              17 |
| DE             |               9 |
| FL             |             162 |
| GA             |              86 |
| HI             |              34 |
| IA             |               2 |
| ID             |               4 |
| IL             |             222 |
| IN             |              16 |
| KS             |              11 |
| KY             |              13 |
| LA             |              24 |
| MA             |              43 |
| MD             |              73 |
| MI             |             114 |
| MN             |              14 |
| MO             |              35 |
| MT             |               5 |
| NC             |              74 |
| ND             |               6 |
| NJ             |              87 |
| NM             |              22 |
| NV             |              43 |
```

10. Trouver le produit le plus cher dans chaque catégorie

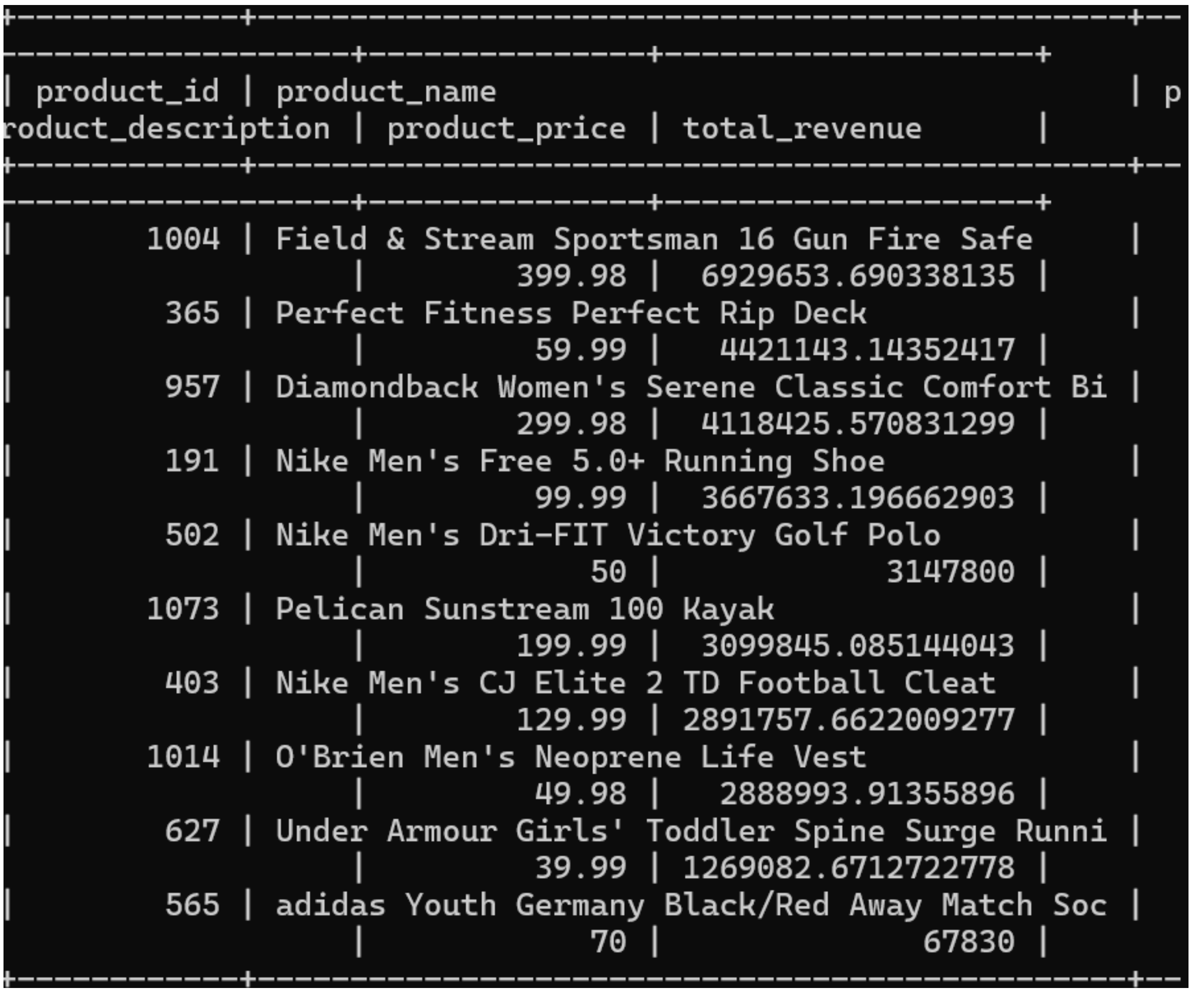
```
WITH MaxPricePerCategory AS (SELECT product_category_id, MAX(product_price) AS
max_price FROM products GROUP BY product_category_id) SELECT p.product_id,
p.product_name, p.product_description, p.product_price, c.category_name FROM products p
JOIN MaxPricePerCategory mpc ON p.product_category_id = mpc.product_category_id AND
p.product_price = mpc.max_price JOIN categories c ON p.product_category_id = c.category_id
ORDER BY c.category_name ASC;
```



```
| product_id | product_name                                      |
roduct_description | product_price | category_name        |
+------------+---------------------------------------------------+-
-------------------+---------------+----------------------+
|        496 | SOLE F85 Treadmill                                |
                   |       1799.99 | Accessories          |
|        590 | adidas Men's Germany Black/Red Away Match Soc |
                   |            90 | Accessories          |
|        593 | adidas Men's Germany Home Soccer Jersey           |
                   |            90 | Accessories          |
|        885 | Team Golf St. Louis Cardinals Putter Grip         |
                   |         24.99 | Accessories          |
|        886 | Team Golf San Francisco Giants Putter Grip        |
                   |         24.99 | Accessories          |
|        887 | Team Golf New York Yankees Putter Grip            |
                   |         24.99 | Accessories          |
|        888 | Team Golf Detroit Tigers Putter Grip              |
                   |         24.99 | Accessories          |
|        889 | Team Golf Chicago Cubs Putter Grip                |
                   |         24.99 | Accessories          |
|        890 | Team Golf Boston Red Sox Putter Grip              |
                   |         24.99 | Accessories          |
|        891 | Team Golf Washington Redskins Putter Grip         |
                   |         24.99 | Accessories          |
|        892 | Team Golf San Francisco 49ers Putter Grip         |
                   |         24.99 | Accessories          |
|        893 | Team Golf Pittsburgh Steelers Putter Grip         |
                   |         24.99 | Accessories          |
|        894 | Team Golf Dallas Cowboys Putter Grip              |
                   |         24.99 | Accessories          |
```

11. Trouvez les 10 meilleurs produits qui ont généré les revenus les plus élevés

```
SELECT p.product_id, p.product_name, p.product_description, p.product_price,
SUM(oi.order_item_subtotal) AS total_revenue FROM products p JOIN order_items oi ON
p.product_id = oi.order_item_product_id GROUP BY p.product_id, p.product_name,
p.product_description, p.product_price ORDER BY total_revenue DESC LIMIT 10;
```



```
+------------+------------------------------------------------+--
------------------+---------------+---------------------+
| product_id | product_name                                   | p
roduct_description | product_price | total_revenue       |
+------------+------------------------------------------------+--
------------------+---------------+---------------------+
|       1004 | Field & Stream Sportsman 16 Gun Fire Safe      |
                   |        399.98 |   6929653.690338135 |
|        365 | Perfect Fitness Perfect Rip Deck               |
                   |         59.99 |    4421143.14352417 |
|        957 | Diamondback Women's Serene Classic Comfort Bi |
                   |        299.98 |   4118425.570831299 |
|        191 | Nike Men's Free 5.0+ Running Shoe              |
                   |         99.99 |   3667633.196662903 |
|        502 | Nike Men's Dri-FIT Victory Golf Polo           |
                   |            50 |             3147800 |
|       1073 | Pelican Sunstream 100 Kayak                    |
                   |        199.99 |   3099845.085144043 |
|        403 | Nike Men's CJ Elite 2 TD Football Cleat        |
                   |        129.99 |  2891757.6622009277 |
|       1014 | O'Brien Men's Neoprene Life Vest               |
                   |         49.98 |    2888993.91355896 |
|        627 | Under Armour Girls' Toddler Spine Surge Runni |
                   |         39.99 |  1269082.6712722778 |
|        565 | adidas Youth Germany Black/Red Away Match Soc |
                   |            70 |               67830 |
+------------+------------------------------------------------+--
```

Conclusion

Ce projet nous a permis de mettre en œuvre l'ingestion de données depuis une base MySQL vers un environnement Hadoop via Sqoop, ainsi que la transformation et l'analyse de ces données avec Hive. Les résultats obtenus à partir des requêtes SQL permettent d'extraire des informations précieuses sur les ventes de l'entreprise.