



Instituto Politécnico Nacional
Escuela Superior de Computo



Sistemas Distribuidos

Tarea extra 1

Procesamiento de tablas agregadas utilizando MapReduce

Nombre del alumno:

García Quiroz Gustavo Ivan

Grupo: 7CV4

Nombre del profesor: Guerrero Carlos Pineda

Fecha de entrega: 17/11/2025

ÍNDICE

1	Introducción	1
2	Objetivos	2
2.1	Objetivo general.....	2
2.2	Objetivos específicos	2
3	Requerimientos y materiales.....	3
4	Arquitectura y entorno de trabajo	5
5	Procedimiento	7
5.1	Creación de la máquina virtual en Azure	7
5.2	Instalación y configuración de MySQL.....	10
5.3	Creación de la base de datos y ejecución del script practica.olap.sql	13
5.4	Carga de datos a la tabla sales_data.....	15
5.5	Población de tablas de dimensiones y fact_table	18
5.6	Generación del archivo country_category_product.csv	25
5.7	Instalación y configuración de Apache Hadoop	26
5.8	Implementación del job MapReduce	31
5.9	Ejecución del job MapReduce y verificación del resultado	36
5.10	Carga del resultado a la tabla agregada country_category_product.....	39
6	Consultas OLAP (usando la tabla agregada)	41
6.1	Acumulado de sales por country.....	41
6.2	Acumulado de sales por country y category	42
6.3	Acumulado de sales por country, category y product.....	43
7	Enlace al chat de la IA GitHub Copilot	45
8	Conclusiones.....	46
9	Anexos	47

10	Referencias (formato IEEE).....	57
----	---------------------------------	----

1 Introducción

Este trabajo implementa, de extremo a extremo, un flujo de procesamiento de datos para obtener tablas agregadas mediante MapReduce sobre una VM de Azure. Se creó una máquina virtual Ubuntu en la región Canadá con el nombre TE1-2022630278; se instaló y configuró MySQL, se creó la base de datos practica.olap, se ejecutó el script practica.olap.sql y se cargaron datos desde sales_data.csv. A partir de la fact_table se generó el archivo country_category_product.csv, que sirvió como entrada para una aplicación Hadoop en Java (AggregationMapper, AggregationReducer y AggregationDriver) ejecutada en modo local, la cual acumuló las ventas por id_country, id_category e id_product. El resultado del proceso (part-r-00000) se transformó a CSV y se cargó en la tabla agregada country_category_product. Finalmente, se consultaron los cubos requeridos (ventas por country; por country y category; y por country, category y product) apoyándose exclusivamente en la tabla agregada. Todo el proceso se realizó desde Windows 11 usando SFTP únicamente cuando fue necesario transferir archivos.

En esta tarea se usó la IA generativa de GitHub copilot para guiarse mejor y completar las instrucciones más rápido. El enlace se encuentra al final de este reporte además de los resultados: tabla agregada country_category_product poblada con ventas acumuladas por país, categoría y producto.

NOTA: La plataforma de Moodle no permitió subir el reporte en buena calidad debido al límite de 2 MB por archivo. Sugiero revisar el siguiente documento de Google drive con el reporte en buena calidad:
<https://drive.google.com/file/d/1adBCfOKGC740AYOSHVSum924Mud-BB1O/view?usp=sharing>

2 Objetivos

2.1 Objetivo general

Implementar un flujo completo de aprovisionamiento, carga, agregación con MapReduce y consulta OLAP que produzca y utilice la tabla agregada country_category_product para responder cubos de ventas con eficiencia.

2.2 Objetivos específicos

- Provisionar una VM Ubuntu en Azure for Students (región Canada) con la denominación TE1-2022630278 y las especificaciones solicitadas.
- Instalar y configurar MySQL; crear la base practica.olap; ejecutar practica.olap.sql; cargar sales_data.csv.
- Poblar las tablas de dimensiones y la fact_table a partir de sales_data.
- Generar el archivo country_category_product.csv como entrada al proceso de MapReduce.
- Instalar y configurar Apache Hadoop (modo local) y Java, incluyendo variables de entorno y hadoop-env.sh.
- Desarrollar, compilar y ejecutar un job MapReduce en Java que acumule ventas por id_country, id_category e id_product.
- Convertir el resultado del job a CSV y cargarlo en la tabla agregada country_category_product.
- Ejecutar las consultas OLAP solicitadas (por country; por country y category; por country, category y product) usando únicamente la tabla agregada.
- Documentar el proceso con evidencias y capturas puntuales para cada hito técnico.

3 Requerimientos y materiales

Para ejecutar la práctica de principio a fin se utilizó una máquina virtual en Azure for Students (región Canada) con Ubuntu y el stack MySQL + Hadoop en modo local. El entorno se mantuvo deliberadamente simple para enfocarse en la generación de la tabla agregada y su consulta.

En términos de cómputo, se empleó una VM con 2 vCPU, 4 GB de RAM y disco SSD estándar, suficiente para ejecutar MySQL y un job de Hadoop MapReduce en modo standalone. El sistema operativo fue Ubuntu 22.04 LTS, lo que garantiza compatibilidad con los paquetes de MySQL y OpenJDK. Desde el equipo cliente (Windows 11), se usó SSH para administración y SFTP únicamente para transferir los archivos proporcionados por el curso (practica.olap.sql y sales_data.csv). En Azure, se expuso únicamente el puerto 22 (SSH) y el resto de la operación se realizó dentro de la VM.

Materiales utilizados (software y archivos):

- Ubuntu Server 22.04 LTS (VM Azure: TE1-2022630278, región Canada)
- MySQL Server y cliente (con soporte de LOAD DATA INFILE/SELECT INTO OUTFILE mediante secure-file-priv)
- OpenJDK (Java 17 recomendado; Java 16 aceptable si el entorno lo requiere)
- Apache Hadoop 3.4.0 en modo local (standalone)
- Herramientas de sistema: nano, wget, tar, unzip
- Cliente SFTP/SSH en Windows 11 (OpenSSH integrado)
- Archivos de la plataforma:
 - practica.olap.sql (definición de tablas: sales_data, dimensiones, fact_table y agregadas)
 - sales_data.csv (datos de ventas con encabezados)

Requisitos de cuentas y permisos:

- Acceso SSH a la VM con usuario con privilegios sudo
- Acceso a MySQL como root (o usuario con permisos de creación/carga)
- Permisos sobre el directorio de secure-file-priv de MySQL (se usó /var/lib/mysql-files)

Notas prácticas:

- secure-file-priv: se trabajó con /var/lib/mysql-files para INFILE/OUTFILE y evitar errores de permisos
- Arquitectura de la VM: x86_64 (usual en Azure); si la VM fuera ARM, se usaría el tarball aarch64 de Hadoop
- Transferencia de archivos: SFTP se utilizó solo para subir practica.olap.sql y sales_data.csv

4 Arquitectura y entorno de trabajo

La arquitectura implementada sigue un flujo lineal: los datos se ingieren en MySQL, se normalizan en dimensiones y fact_table, se exportan a CSV y se procesan con Hadoop MapReduce para obtener agregados, que se reimportan a MySQL en una tabla agregada optimizada para consultas. Todo corre en una sola VM (sin HDFS/YARN), aprovechando Hadoop en modo standalone y el sistema de archivos local.

Descripción del flujo:

- Ingesta y modelado: sales_data.csv se carga en MySQL; a partir de ella se pueblan tablas de dimensiones, fechas y fact_table
- Extracción a CSV: se genera country_category_product.csv mediante SELECT ... INTO OUTFILE
- Procesamiento MapReduce: un job en Java (Mapper, Reducer, Driver) acumula ventas por id_country, id_category e id_product
- Postproceso y carga: el resultado part-r-00000 se convierte a CSV (tabulador → coma) y se carga en la tabla agregada country_category_product
- Consultas OLAP: los cubos se ejecutan exclusivamente sobre la tabla agregada para reducir costos de cómputo y latencia

Componentes y configuración clave:

- Host de cómputo: Azure VM TE1-2022630278 (Ubuntu 22.04, 2 vCPU, 4 GB RAM, SSD estándar, región Canada)
- Base de datos: MySQL con directorio de intercambio /var/lib/mysql-files (secure-file-priv)
- Hadoop: 3.4.0 en modo local, sin HDFS; entrada/salida en el FS local (por ejemplo, prueba/input y prueba/output)
- Java: OpenJDK (JAVA_HOME definido); Hadoop configurado en hadoop-env.sh

- Cliente: Windows 11 para acceso SSH y SFTP (solo cuando es necesario copiar archivos)

Variables de entorno (referenciales):

- JAVA_HOME (por ejemplo, /usr/lib/jvm/java-17-openjdk-amd64)
- HADOOP_HOME (/home/usuario/hadoop-3.4.0)
- PATH extendido con \$JAVA_HOME/bin y \$HADOOP_HOME/bin

Consideraciones de red y seguridad:

- Puerto 22 abierto para administración (SSH/SFTP)
- Descargas de paquetes y binarios desde repositorios oficiales (apt y Apache)
- No se exponen puertos de MySQL al exterior; se opera localmente dentro de la VM

Estructura de trabajo (local):

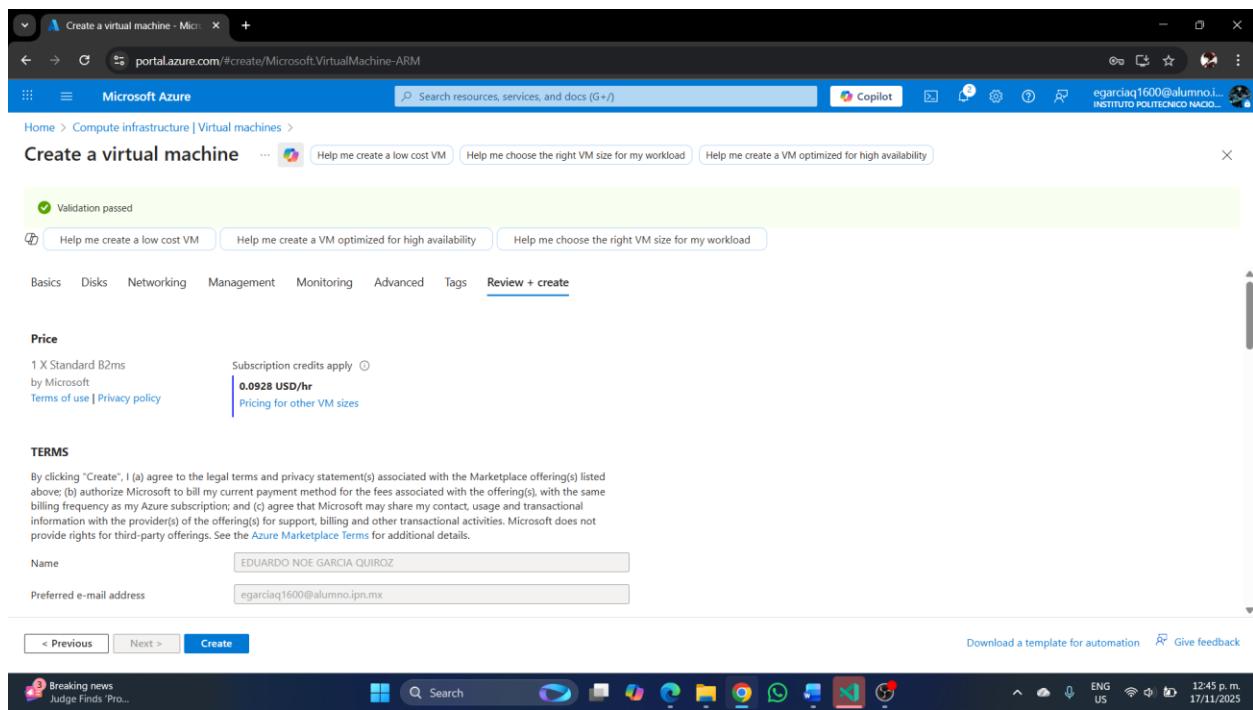
- Directorio del proyecto MapReduce (por ejemplo, ~/prueba) con fuentes Java, build y Aggregation.jar
- Directarios input y output para el job de Hadoop
- Archivos CSV intermedios y finales ubicados en /var/lib/mysql-files para facilitar INFILE/OUTFILE

5 Procedimiento

5.1 Creación de la máquina virtual en Azure

Se creó una máquina virtual en Azure for Students con Ubuntu 22.04 LTS en la región Canada. Se asignó el nombre TE1-2022630278, 2 vCPU, 4 GB de RAM y disco SSD estándar. Se habilitó únicamente el puerto 22 para SSH. Al finalizar, se verificó la IP pública desde el panel de la VM.

- Parámetros clave: Nombre TE1-2022630278, Región Canada, Tamaño Standard_B2s, Disco Standard SSD, Puerto 22 (SSH).



Create a virtual machine - Microsoft Azure

portal.azure.com/#create/Microsoft.VirtualMachine-ARM

Microsoft Azure

Search resources, services, and docs (G +)

Copilot

Help me create a low cost VM Help me choose the right VM size for my workload Help me create a VM optimized for high availability

Home > Compute infrastructure | Virtual machines >

Create a virtual machine

Validation passed

Help me create a low cost VM Help me create a VM optimized for high availability Help me choose the right VM size for my workload

Basics

Subscription	Azure for Students
Resource group	(new) TE1-2022630278-reg
Virtual machine name	TE1-2022630278
Region	Canada Central
Availability options	Availability zone
Zone options	Self-selected zone
Availability zone	1
Security type	Trusted launch virtual machines
Enable secure boot	Yes
Enable vTPM	Yes
Integrity monitoring	No
Image	Ubuntu Server 24.04 LTS - Gen2
VM architecture	x64
Size	Standard B2ms (2 vcpus, 8 GiB memory)
Enable Hibernation	No
Authentication type	Password
Username	azureuser

< Previous Next > Create Download a template for automation Give feedback

Breaking news Judge Finds 'Pro...' Search ENG US 12:45 p.m. 17/11/2025

Create a virtual machine - Microsoft Azure

portal.azure.com/#create/Microsoft.VirtualMachine-ARM

Microsoft Azure

Search resources, services, and docs (G +)

Copilot

Help me create a low cost VM Help me choose the right VM size for my workload Help me create a VM optimized for high availability

Home > Compute infrastructure | Virtual machines >

Create a virtual machine

Validation passed

Help me create a low cost VM Help me create a VM optimized for high availability Help me choose the right VM size for my workload

Disk

OS disk size	64 GiB
OS disk type	Standard SSD LRS
Use managed disks	Yes
Delete OS disk with VM	Enabled
Ephemeral OS disk	No

Networking

Virtual network	vnet-canadacentral
Subnet	snet-canadacentral-1
Public IP	(new) TE1-2022630278-ip
NIC network security group	(new) TE1-2022630278-nsg
Accelerated networking	Off
Place this virtual machine behind an existing load balancing solution?	No
Delete public IP and NIC when VM is deleted	Disabled

< Previous Next > Create Download a template for automation Give feedback

Breaking news Judge Finds 'Pro...' Search ENG US 12:45 p.m. 17/11/2025

The image consists of two screenshots of the Microsoft Azure Portal.

Screenshot 1: Create a virtual machine

- Header:** Create a virtual machine - Microsoft Azure | portal.azure.com
- Breadcrumbs:** Home > Compute infrastructure > Virtual machines >
- Title:** Create a virtual machine
- Validation:** Validation passed
- Buttons:** Help me create a low cost VM, Help me create a VM optimized for high availability, Help me choose the right VM size for my workload
- Management:**
 - Microsoft Defender for Cloud: None
 - System assigned managed identity: Off
 - Login with Microsoft Entra ID: Off
 - Auto-shutdown: Off
 - Enable periodic assessment: Off
 - Enable hotpatch: Off
 - Patch orchestration options: Image Default
- Monitoring:**
 - Alerts: Off
 - Boot diagnostics: On
 - Enable OS guest diagnostics: Off
 - Enable application health monitoring: Off
- Advanced:**
 - Extensions: None
- Buttons:** < Previous, Next >, Create
- Footer:** Download a template for automation, Give feedback

Screenshot 2: Overview of the created VM

- Header:** TE1-2022630278 - Microsoft Azure | portal.azure.com
- Breadcrumbs:** Home > CreateVm-canonical.ubuntu-24_04-lts-server-20251117124320 | Overview >
- Title:** TE1-2022630278
- Actions:** Help me copy this VM in any region, Manage this VM with Azure CLI
- Overview Panel:**
 - Essentials:**
 - Resource group: TE1-2022630278-reg
 - Status: Running
 - Location: Canada Central (Zone 1)
 - Subscription: Azure for Students
 - Subscription ID: 0509baab-81e3-456c-bfc3-58e571b542ab
 - Availability zone: 1
 - Tags:** Tags (edit) : Add tags
 - Properties:** Properties, Monitoring, Capabilities (7), Recommendations, Tutorials
 - Virtual machine:**
 - Computer name: TE1-2022630278
 - Operating system: Linux (ubuntu 24.04)
 - VM generation: V2
 - VM architecture: x64
 - Networking:**
 - Public IP address: 4.206.200.45 (Network interface te1-2022630278814_21)
 - Public IP address (IPv6): -
 - Private IP address: 172.16.0.4
- Footer:** 25°C Sunny, Search, Copilot, etc.

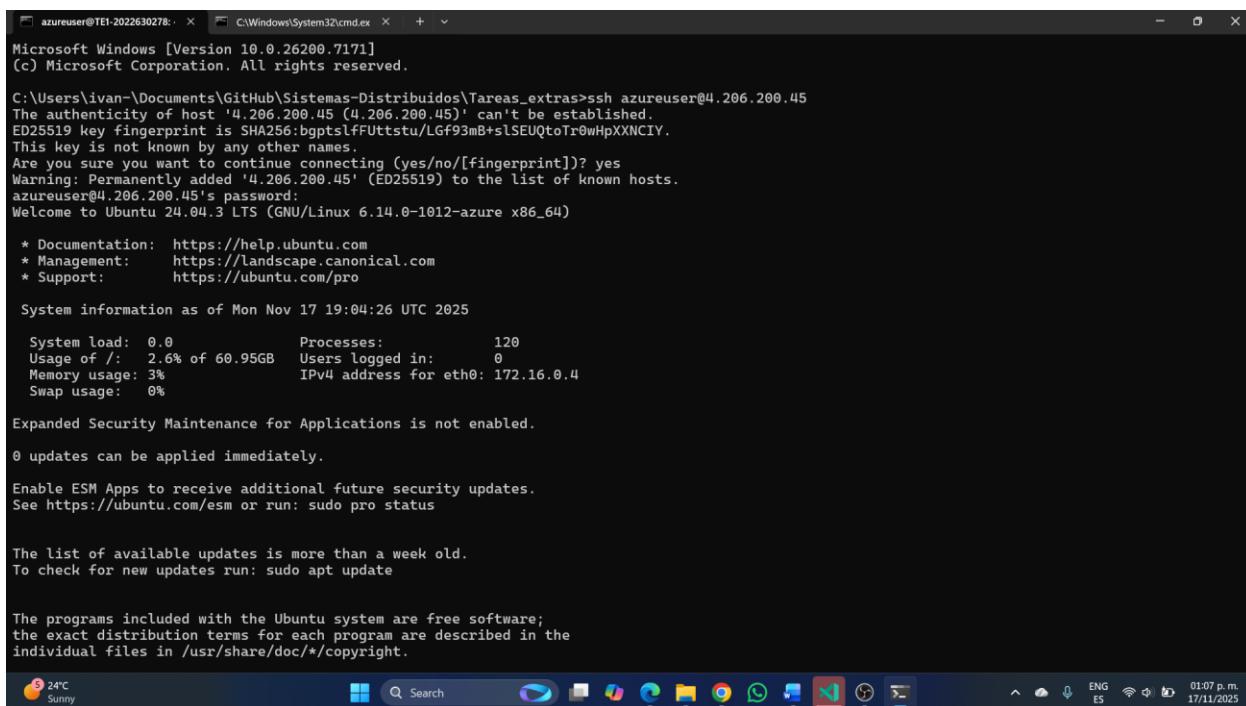
Imagen 7: Resumen de la VM creada (Overview en Azure Portal mostrando nombre TE1-2022630278, tamaño, región y IP pública).

5.2 Instalación y configuración de MySQL

Se accedió por contraseña desde Windows 11 y se actualizó el sistema. Posteriormente, se instaló MySQL Server y herramientas básicas de administración.

```
ssh azureuser@4.206.200.45
```

```
sudo apt update && sudo apt -y upgrade
```



```
azuser@TE1-2022630278: ~ C:\Windows\System32\cmd.exe + v
Microsoft Windows [Version 10.0.26200.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ivan-\Documents\GitHub\Sistemas-Distribuidos\Tareas_extras>ssh azureuser@4.206.200.45
The authenticity of host '4.206.200.45 (4.206.200.45)' can't be established.
ED25519 key fingerprint is SHA256:bgptsLffFUttstu/LGf93mB+slSEUQtoTr0whPxNNCTY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '4.206.200.45' (ED25519) to the list of known hosts.
azureuser@4.206.200.45's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1012-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Mon Nov 17 19:04:26 UTC 2025

 System load:  0.0          Processes:           120
 Usage of /:   2.6% of 60.95GB  Users logged in:     0
 Memory usage: 3%            IPv4 address for eth0: 172.16.0.4
 Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

 24°C
Sunny
  Search
  Home
  File
  Applications
  Dash
  Help
  Network
  Sound
  Volume
  Battery
  Power
  Screen
  Language
  ENG
  ES
  01:07 p.m.
  17/11/2025
```

Imagen 7.1 Inicio de sesión en la terminal.

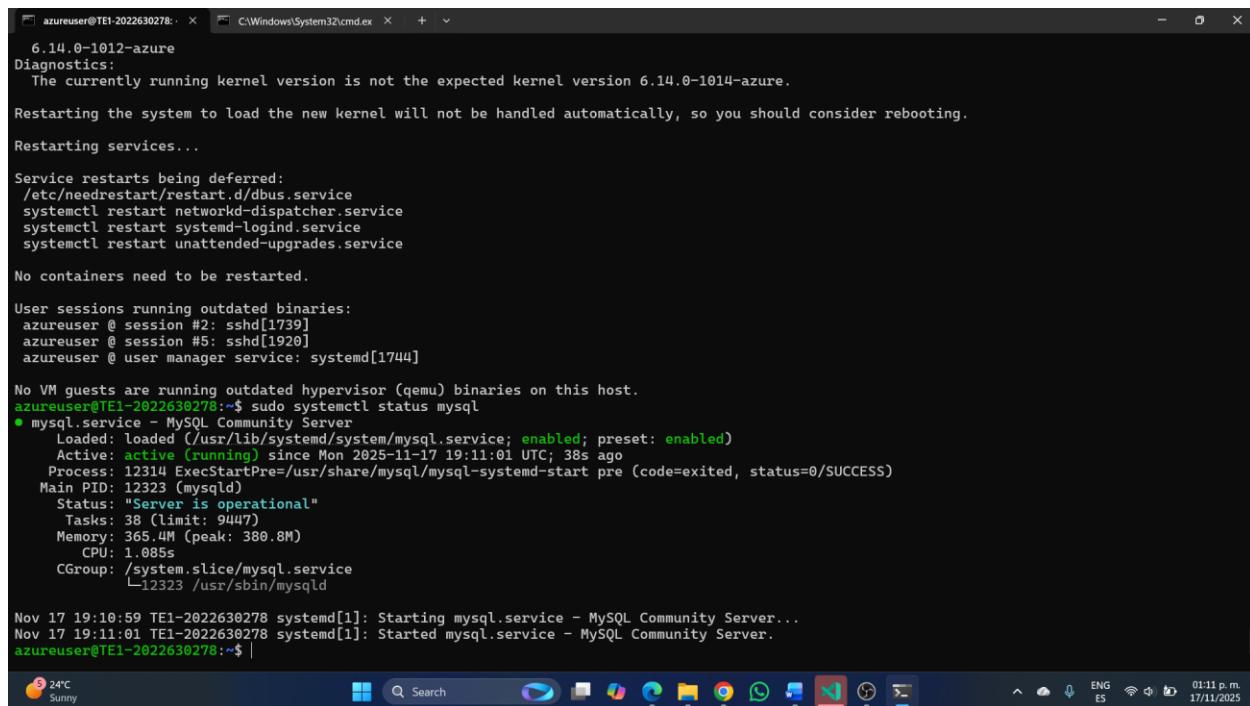
```
sudo apt -y install mysql-server mysql-client nano wget tar unzip
```

```
azuser@TE1-2022630278:~$ sudo apt -y install mysql-server mysql-client nano wget tar unzip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nano is already the newest version (7.2-2ubuntu0.1).
nano set to manually installed.
wget is already the newest version (1.21.4-1ubuntu4.1).
wget set to manually installed.
tar is already the newest version (1.35+dfsg-3build1).
tar set to manually installed.
The following additional packages will be installed:
  libcgifast-perl libcgipm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7t64 libfcgi-bin libfcgi-perl libfcgi8t64
  libhtml-parser-perl libhttp-tagger-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl
  libmecab2 libprotobuf-lite32t64 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0 mysql-common mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libbio-compress-brotli-perl libbusiness-isbn-perl libregexp-ipv6-perl libwww-perl mailx tinyca zip
The following NEW packages will be installed:
  libcgifast-perl libcgipm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7t64 libfcgi-bin libfcgi-perl libfcgi8t64
  libhtml-parser-perl libhttp-tagger-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl
  libmecab2 libprotobuf-lite32t64 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client mysql-client-8.0
  mysql-client-core-8.0 mysql-common mysql-server mysql-server-8.0 mysql-server-core-8.0 unzip
0 upgraded, 30 newly installed, 0 to remove and 0 not upgraded.
Need to get 29.8 MB of archives.
After this operation, 243 MB of additional disk space will be used.
Get:1 http://azure.archive.ubuntu.com/ubuntu/noble/main amd64 mysql-common all 5.8+1.1.0build1 [6746 B]
Get:2 http://azure.archive.ubuntu.com/ubuntu/noble-updates/main amd64 mysql-client-core-8.0 amd64 8.0.43-0ubuntu0.24.04.2 [2740 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu/noble-updates/main amd64 mysql-client-8.0 amd64 8.0.43-0ubuntu0.24.04.2 [22.4 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu/noble/main amd64 libevent-pthreads-2.1-7t64 amd64 2.1.12-stable-9ubuntu2 [7982 B]
Get:5 http://azure.archive.ubuntu.com/ubuntu/noble/main amd64 libmecab2 amd64 0.996-14ubuntu4 [201 kB]
Get:6 http://azure.archive.ubuntu.com/ubuntu/noble-updates/main amd64 libprotobuf-lite32t64 amd64 3.21.12-8.2ubuntu0.2 [238 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu/noble-updates/main amd64 mysql-server-core-8.0 amd64 8.0.43-0ubuntu0.24.04.2 [17.5 MB]
Get:8 http://azure.archive.ubuntu.com/ubuntu/noble-updates/main amd64 mysql-server-8.0 amd64 8.0.43-0ubuntu0.24.04.2 [1439 kB]
Get:9 http://azure.archive.ubuntu.com/ubuntu/noble/main amd64 libhttp-tagger-perl all 3.20-6 [11.3 kB]
Get:10 http://azure.archive.ubuntu.com/ubuntu/noble/main amd64 liburi-perl all 5.27-1 [88.0 kB]
Get:11 http://azure.archive.ubuntu.com/ubuntu/noble/main amd64 libhtml-parser-perl amd64 3.18-1build3 [85.8 kB]
Get:12 http://azure.archive.ubuntu.com/ubuntu/noble/main amd64 libcgipm-perl all 4.63-1 [185 kB]
Get:13 http://azure.archive.ubuntu.com/ubuntu/noble-updates/main amd64 libfcgi8t64 amd64 2.4.2-2.1ubuntu0.24.04.1 [27.0 kB]
```

Imagen 7.2 Instalación de MySQL.

Se verificó que el servicio MySQL quedara activo:

```
sudo systemctl status mysql
```



```
azuruser@TE1-2022630278: ~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
  Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
  Active: active (running) since Mon 2025-11-17 19:11:01 UTC; 38s ago
    Process: 12314 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
   Main PID: 12323 (mysqld)
     Status: "Server is operational"
       Tasks: 38 (limit: 9447)
      Memory: 365.4M (peak: 380.8M)
        CPU: 1.085s
       CGroup: /system.slice/mysql.service
               └─12323 /usr/sbin/mysqld

Nov 17 19:10:59 TE1-2022630278 systemd[1]: Starting mysql.service - MySQL Community Server...
Nov 17 19:11:01 TE1-2022630278 systemd[1]: Started mysql.service - MySQL Community Server.
azuruser@TE1-2022630278: ~$ |
```

Imagen 7.3 Verificación de MySQL.

Se preparó el directorio permitido por secure-file-priv para operaciones INFILE/OUTFILE:

```
sudo mkdir -p /var/lib/mysql-files
```

```
sudo chown mysql:mysql /var/lib/mysql-files
```

```
sudo chmod 750 /var/lib/mysql-files
```

The screenshot shows a Windows 11 desktop environment. In the center is a terminal window titled 'azureuser@TE1-2022630278' with the command 'C:\Windows\System32\cmd.exe'. The terminal displays the following text:

```

Restarting the system to load the new kernel will not be handled automatically, so you should consider rebooting.
Restarting services...
Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
azureuser @ session #2: sshd[1739]
azureuser @ session #5: sshd[1920]
azureuser @ user manager service: systemd[1744]

No VM guests are running outdated hypervisor (qemu) binaries on this host.

azureuser@TE1-2022630278:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
  Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
  Active: active (running) since Mon 2025-11-17 19:11:01 UTC; 38s ago
    Process: 12314 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
   Main PID: 12323 (mysqld)
     Status: "Server is operational"
      Tasks: 38 (limit: 9447)
     Memory: 365.4M (peak: 380.8M)
        CPU: 1.085s
       CGroub: /system.slice/mysql.service
              └─12323 /usr/sbin/mysqld

Nov 17 19:10:59 TE1-2022630278 systemd[1]: Starting mysql.service - MySQL Community Server...
Nov 17 19:11:01 TE1-2022630278 systemd[1]: Started mysql.service - MySQL Community Server.

azureuser@TE1-2022630278:~$ sudo mkdir -p /var/lib/mysql-files
azureuser@TE1-2022630278:~$ sudo chown mysql:mysql /var/lib/mysql-files
azureuser@TE1-2022630278:~$ sudo chmod 750 /var/lib/mysql-files
azureuser@TE1-2022630278:~$ |

```

The taskbar at the bottom shows various pinned icons including File Explorer, Edge, and File History. The system tray indicates it's 01:12 p.m. on 17/11/2025, the weather is 24°C and sunny, and the language is set to ENG ES.

Imagen 8: MySQL instalado y servicio activo (systemctl status mysql en “active (running)”).

5.3 Creación de la base de datos y ejecución del script practica.olap.sql

Se transfirió el archivo practica.olap.sql desde Windows 11 a la VM usando SFTP.

- Desde Windows 11 (PowerShell o Windows Terminal)

sftp azureuser@4.206.200.45

put practica.olap.sql

exit

```
azureuser@TE1-2022630278: ~ C:\Windows\System32\cmd.e + - x
Microsoft Windows [Version 10.0.26200.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ivan-\Documents\GitHub\Sistemas-Distribuidos\Tareas_extras>sftp azureuser@4.206.200.45
azureuser@4.206.200.45's password:
Connected to 4.206.200.45.
sftp> put practica.olap.sql
Uploading practica.olap.sql to /home/azureuser/practica.olap.sql
practica.olap.sql                                         100% 4402     36.1KB/s   00:00
sftp> put sales_data.csv
Uploading sales_data.csv to /home/azureuser/sales_data.csv
sales_data.csv                                           100% 1153KB   1.2MB/s   00:00
sftp> exit

C:\Users\ivan-\Documents\GitHub\Sistemas-Distribuidos\Tareas_extras>
```

Imagen 8.1 Copiar y pegar archivos a las VM's.

Se creó la base de datos `practica.olap` y se ejecutó el script con la definición de tablas (incluyendo dimensiones, `fact_table` y tablas agregadas).

```
sudo mysql
```

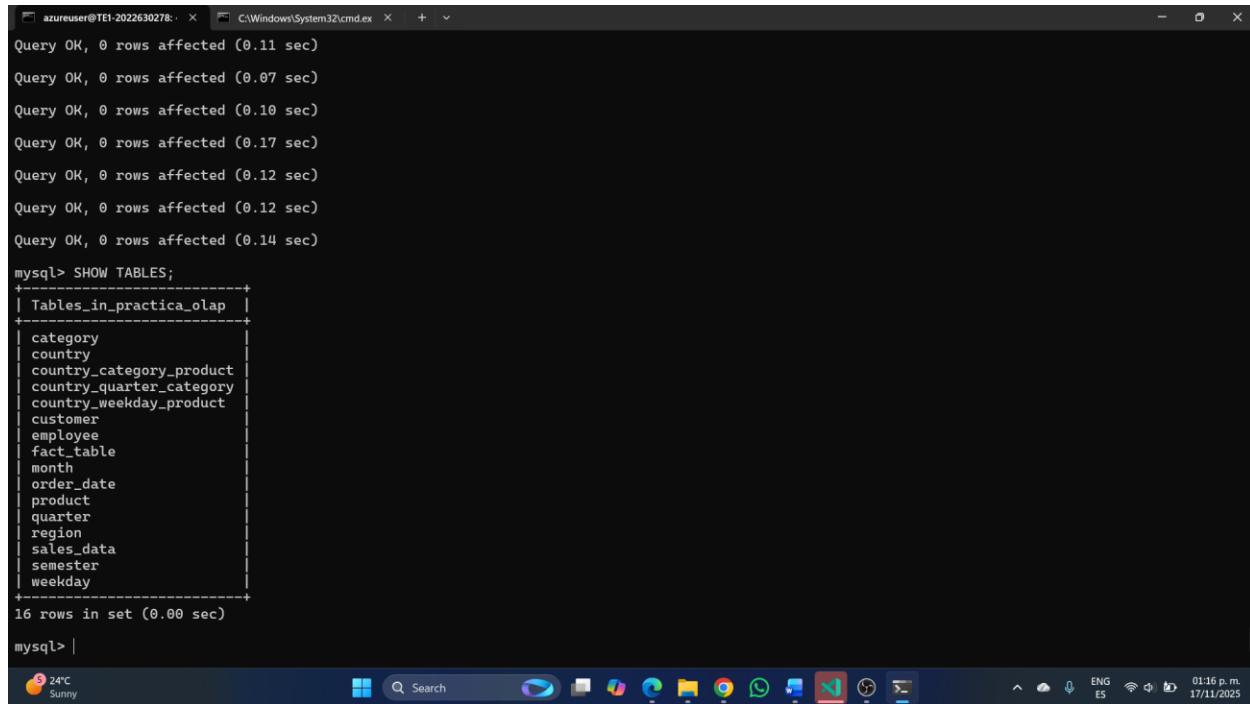
```
CREATE DATABASE practica.olap;
```

```
USE practica.olap;
```

```
SOURCE /home/azureuser/practica.olap.sql;
```

```
SHOW TABLES;
```

```
EXIT;
```



```
azureuser@TE1-2022630278: ~ C:\Windows\System32\cmd.exe + - x
Query OK, 0 rows affected (0.11 sec)
Query OK, 0 rows affected (0.07 sec)
Query OK, 0 rows affected (0.10 sec)
Query OK, 0 rows affected (0.17 sec)
Query OK, 0 rows affected (0.12 sec)
Query OK, 0 rows affected (0.12 sec)
Query OK, 0 rows affected (0.14 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_practica_olap |
+-----+
| category
| country
| country_category_product
| country_quarter_category
| country_weekday_product
| customer
| employee
| fact_table
| month
| order_date
| product
| quarter
| region
| sales_data
| semester
| weekday
+-----+
16 rows in set (0.00 sec)

mysql> |
```

Imagen 9: Evidencia de creación de la base (SHOW TABLES mostrando sales_data, dimensiones, fact_table y tablas agregadas).

5.4 Carga de datos a la tabla sales_data

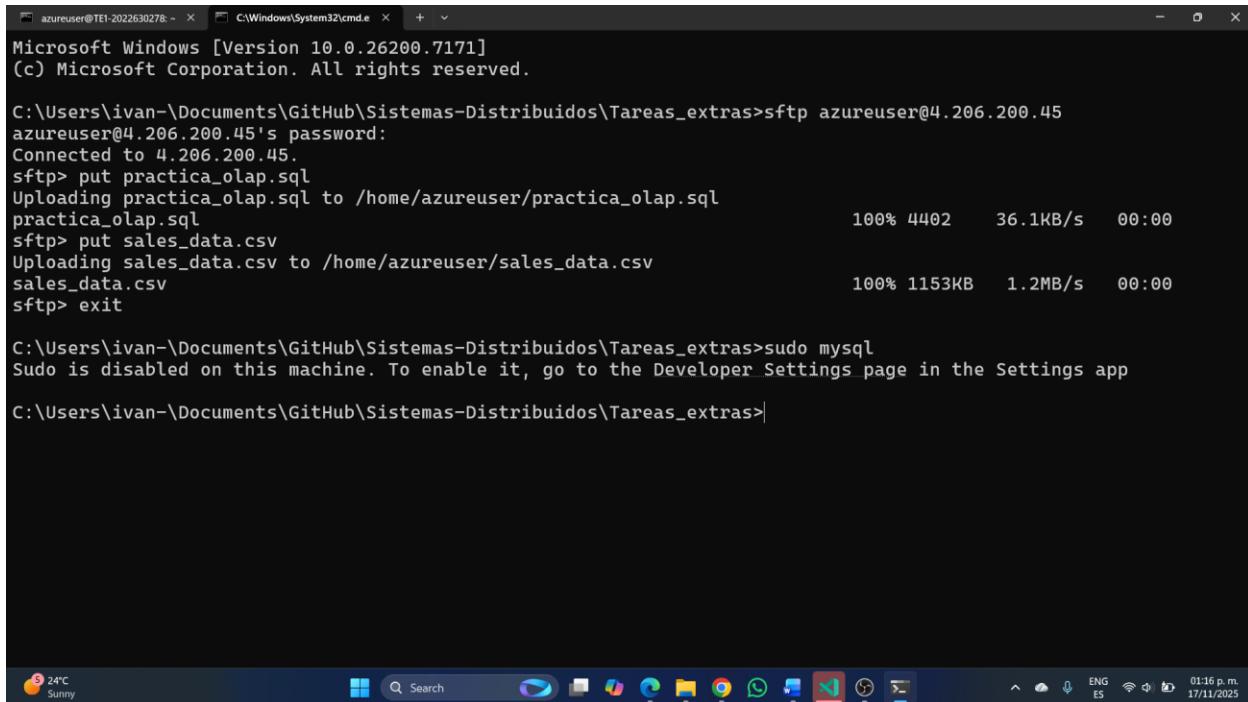
Se transfirió sales_data.csv mediante SFTP y se colocó en el directorio seguro de MySQL para la carga con LOAD DATA INFILE.

- Desde Windows 11

```
sftp azureuser@<IP_PUBLICA>
```

```
put sales_data.csv
```

```
exit
```



```
Microsoft Windows [Version 10.0.26200.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ivan-\Documents\GitHub\Sistemas-Distribuidos\Tareas_extras>sftp azureuser@4.206.200.45
azureuser@4.206.200.45's password:
Connected to 4.206.200.45.
sftp> put practica.olap.sql
Uploading practica.olap.sql to /home/azureuser/practica.olap.sql
practica.olap.sql                                         100% 4402     36.1KB/s   00:00
sftp> put sales_data.csv
Uploading sales_data.csv to /home/azureuser/sales_data.csv
sales_data.csv                                           100% 1153KB   1.2MB/s   00:00
sftp> exit

C:\Users\ivan-\Documents\GitHub\Sistemas-Distribuidos\Tareas_extras>sudo mysql
Sudo is disabled on this machine. To enable it, go to the Developer Settings page in the Settings app

C:\Users\ivan-\Documents\GitHub\Sistemas-Distribuidos\Tareas_extras>
```

Imagen 9.1: Pasar archivos a la VM.

- En la VM

```
sudo mv /home/azureuser/sales_data.csv /var/lib/mysql-files/
```

```
sudo chmod 644 /var/lib/mysql-files/sales_data.csv
```

```

azureuser@TE1-2022630278: ~ C:\Windows\System32\cmd.exe + -
Query OK, 0 rows affected (0.10 sec)
Query OK, 0 rows affected (0.17 sec)
Query OK, 0 rows affected (0.12 sec)
Query OK, 0 rows affected (0.12 sec)
Query OK, 0 rows affected (0.14 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_practica_olap |
+-----+
| category
| country
| country_category_product
| country_quarter_category
| country_weekday_product
| customer
| employee
| fact_table
| month
| order_date
| product
| quarter
| region
| sales_data
| semester
| weekday
+-----+
16 rows in set (0.00 sec)

mysql> EXIT;
Bye
azureuser@TE1-2022630278:~$ sudo mv /home/azureuser/sales_data.csv /var/lib/mysql-files/
azureuser@TE1-2022630278:~$ sudo chmod 644 /var/lib/mysql-files/sales_data.csv
azureuser@TE1-2022630278:~$ |
```

Imagen 9.2: Mostrar tablas de la VM.

Se realizó la carga a la tabla sales_data (omitiendo la línea de encabezados):

sudo mysql

USE practica.olap;

LOAD DATA INFILE '/var/lib/mysql-files/sales_data.csv'

INTO TABLE sales_data

FIELDS TERMINATED BY ',' ENCLOSED BY ""

IGNORE 1 LINES

(sales, order_date, product, customer, country, region, employee, category, weekday, month, quarter, semester);

SELECT COUNT(*) AS filas_en_sales_data FROM sales_data;

EXIT;

```

azureuser@TE1-2022630278: ~ C:\Windows\System32\cmd.exe + - x
Server version: 8.0.43-0ubuntu0.24.04.2 (Ubuntu)
Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE practica.olap;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

D BY ',' ENCLOSED BY ""
IGNORE 1 LINES
(sales, order_date, product, customer, country, region, employee, category, weekday, month, quarter, semester);
SELECT COUNT(*) AS filas_en_sales_data FDatabase changed
mysql> LOAD DATA INFILE '/var/lib/mysql-files/sales_data.csv'
-> INTO TABLE sales_data
-> FIELDS TERMINATED BY ',' ENCLOSED BY ""
-> IGNORE 1 LINES
-> (sales, order_date, product, customer, country, region, employee, category, weekday, month, quarter, semester);
ROM sales_data;
Query OK, 10000 rows affected (0.58 sec)
Records: 10000 Deleted: 0 Skipped: 0 Warnings: 0

mysql> SELECT COUNT(*) AS filas_en_sales_data FROM sales_data;
+-----+
| filas_en_sales_data |
+-----+
|          10000 |
+-----+
1 row in set (0.00 sec)

mysql> EXIT;
Bye
azureuser@TE1-2022630278:~$ |
```

Imagen 10: Resultado de la carga (SELECT COUNT(*) FROM sales_data mostrando la cantidad de filas cargadas).

5.5 Población de tablas de dimensiones y fact_table

Se poblaron las tablas de dimensiones a partir de sales_data y, posteriormente, la fact_table con los IDs correspondientes.

- Dentro de MySQL (USE practica.olap;)

INSERT INTO region (region)

SELECT DISTINCT region FROM sales_data WHERE region IS NOT NULL;

INSERT INTO country (country, id_region)

SELECT DISTINCT sd.country, r.id_region

FROM sales_data sd

JOIN region r ON r.region = sd.region;

```
INSERT INTO customer (customer)  
SELECT DISTINCT customer FROM sales_data WHERE customer IS NOT NULL;
```

```
INSERT INTO employee (employee)  
SELECT DISTINCT employee FROM sales_data WHERE employee IS NOT NULL;
```

```
INSERT INTO semester (semester)  
SELECT DISTINCT semester FROM sales_data WHERE semester IS NOT NULL;
```

```
INSERT INTO quarter (quarter, id_semester)  
SELECT DISTINCT sd.quarter, s.id_semester  
FROM sales_data sd  
JOIN semester s ON s.semester = sd.semester;
```

```
INSERT INTO month (month, id_quarter)  
SELECT DISTINCT sd.month, q.id_quarter  
FROM sales_data sd  
JOIN quarter q ON q.quarter = sd.quarter;
```

```
INSERT INTO weekday (weekday)  
SELECT DISTINCT weekday FROM sales_data WHERE weekday IS NOT NULL;
```

```
INSERT INTO category (category)
SELECT DISTINCT category FROM sales_data WHERE category IS NOT NULL;
```

```
INSERT INTO product (product, id_category)
SELECT DISTINCT sd.product, c.id_category
FROM sales_data sd
JOIN category c ON c.category = sd.category;
```

```
INSERT INTO order_date (order_date, id_weekday, id_month)
SELECT DISTINCT sd.order_date, w.id_weekday, m.id_month
FROM sales_data sd
JOIN weekday w ON w.weekday = sd.weekday
JOIN month m ON m.month = sd.month;
```

```
INSERT INTO fact_table (sales, id_order_date, id_product, id_customer, id_country,
id_employee)
```

```
SELECT
sd.sales,
od.id_order_date,
p.id_product,
cu.id_customer,
co.id_country,
e.id_employee
```

```
FROM sales_data sd  
JOIN order_date od ON od.order_date = sd.order_date  
JOIN product p ON p.product = sd.product  
JOIN customer cu ON cu.customer = sd.customer  
JOIN country co ON co.country = sd.country  
JOIN employee e ON e.employee = sd.employee;
```

Se verificó el llenado básico:

```
SELECT  
(SELECT COUNT(*) FROM product) AS productos,  
(SELECT COUNT(*) FROM category) AS categorias,  
(SELECT COUNT(*) FROM country) AS paises,  
(SELECT COUNT(*) FROM fact_table) AS filas_fact;
```

```
azureuser@TE1-2022630278:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.43-Ubuntu0.24.04.2 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE practica_olap;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> INSERT INTO region (region)
-> SELECT DISTINCT region FROM sales_data WHERE region IS NOT NULL;
Query OK, 4 rows affected (0.03 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> INSERT INTO country (country, id_region)
-> SELECT DISTINCT sd.country, r.id_region
-> FROM sales_data sd
-> JOIN region r ON r.region = sd.region;
r;

INSERT INTO month (month, id_quarter)
SELECT DISTINCT sd.month, q.id_quarter
FROM sales_data sd
JOIN quarter q ON q.quarter = sd.quarter;

INSERT INTO weekday (weekday)
SELECT DISTINCT weekday FROM sales_data WHERE weekday IS NOT NULL;

INSERT INTO category (category)

```

```
azureuser@TE1-2022630278:~$ C:\Windows\System32\cmd.exe + v
INSERT INTO category (category)
SELECT DISTINCT category FROM sales_data WHERE category IS NOT NULL;

INSERT INTO product (product, id_category)
SELECT DISTINCT sd.product, c.id_category
FROM sales_data sd
JOIN category c ON c.category = sd.category;

INSERT INTO order_date (order_date, id_weekday, id_month)
SELECT DISTINCT sd.order_date, w.id_weekday, m.id_month
FROM sales_data sd
JOIN weekday w ON w.weekday = sd.weekday
JOIN month m ON m.month = sd.month;

INSERT INTO fact_table (sales, id_order_date, id_product, id_customer, id_country, id_employee)
SELECT
sd.sales,
od.id_order_date,
p.id_product,
cu.id_customer,
co.id_country,
e.id_employee
FROM sales_data sd
JOIN order_date od ON od.order_date = sd.order_date
JOIN product p ON p.product = sd.product
JOIN customer cu ON cu.customer = sd.customer
JOIN country co ON co.country = sd.country
JOIN employee e ON e.employee = sd.employee;
Se verificó el llenado básico:
SQL
SELECT
(SELECT COUNT(*) FROM product) AS productos,
(SELECT COUNT(*) FROM category) AS categorias,
(SELECT COUNT(*) FROM country) AS paises,
(SELECT COUNT(*) FROM weekday) AS dias_semana
Query OK, 16 rows affected (0.03 sec)
Records: 16 Duplicates: 0 Warnings: 0

mysql>
```

```
azureuser@TE1-2022630278: ~ C:\Windows\System32\cmd.exe + - x
mysql>
mysql> INSERT INTO customer (customer)
-> SELECT DISTINCT customer FROM sales_data WHERE customer IS NOT NULL;
Query OK, 15 rows affected (0.03 sec)
Records: 15 Duplicates: 0 Warnings: 0

mysql>
mysql> INSERT INTO employee (employee)
-> SELECT DISTINCT employee FROM sales_data WHERE employee IS NOT NULL;
T COUNT(*) FROM fact_table) AS filas_fact;
Query OK, 10 rows affected (0.03 sec)
Records: 10 Duplicates: 0 Warnings: 0

mysql>
mysql> INSERT INTO semester (semester)
-> SELECT DISTINCT semester FROM sales_data WHERE semester IS NOT NULL;
Query OK, 2 rows affected (0.03 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql>
mysql> INSERT INTO quarter (quarter, id_semester)
-> SELECT DISTINCT sd.quarter, s.id_semester
-> FROM sales_data sd
-> JOIN semester s ON s.semester = sd.semester;
Query OK, 4 rows affected (0.04 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql>
mysql> INSERT INTO month (month, id_quarter)
-> SELECT DISTINCT sd.month, q.id_quarter
-> FROM sales_data sd
-> JOIN quarter q ON q.quarter = sd.quarter;
Query OK, 12 rows affected (0.03 sec)

Tomorrow's high
To break record
01:25 p.m.
ENG
17/11/2025
```

```
azureuser@TE1-2022630278: ~ C:\Windows\System32\cmd.exe + - x
mysql>
mysql> INSERT INTO weekday (weekday)
-> SELECT DISTINCT weekday FROM sales_data WHERE weekday IS NOT NULL;
Query OK, 7 rows affected (0.03 sec)
Records: 7 Duplicates: 0 Warnings: 0

mysql>
mysql> INSERT INTO category (category)
-> SELECT DISTINCT category FROM sales_data WHERE category IS NOT NULL;
Query OK, 24 rows affected (0.03 sec)
Records: 24 Duplicates: 0 Warnings: 0

mysql>
mysql> INSERT INTO product (product, id_category)
-> SELECT DISTINCT sd.product, c.id_category
-> FROM sales_data sd
-> JOIN category c ON c.category = sd.category;
Query OK, 51 rows affected (0.03 sec)
Records: 51 Duplicates: 0 Warnings: 0

mysql>
mysql> INSERT INTO order_date (order_date, id_weekday, id_month)
-> SELECT DISTINCT sd.order_date, w.id_weekday, m.id_month
-> FROM sales_data sd
-> JOIN weekday w ON w.weekday = sd.weekday
-> JOIN month m ON m.month = sd.month;
Query OK, 1571 rows affected (0.15 sec)
Records: 1571 Duplicates: 0 Warnings: 0

mysql>
mysql> INSERT INTO fact_table (sales, id_order_date, id_product, id_customer, id_country, id_employee)
-> SELECT
-> sd.sales,
-> od.id_order_date,
```

```
azuser@TE1-2022630278: ~ C:\Windows\System32\cmd.exe + x
-> SELECT DISTINCT sd.product, c.id_category
-> FROM sales_data sd
-> JOIN category c ON c.category = sd.category;
Query OK, 51 rows affected (0.03 sec)
Records: 51 Duplicates: 0 Warnings: 0

mysql>
mysql> INSERT INTO order_date (order_date, id_weekday, id_month)
-> SELECT DISTINCT sd.order_date, w.id_weekday, m.id_month
-> FROM sales_data sd
-> JOIN weekday w ON w.weekday = sd.weekday
-> JOIN month m ON m.month = sd.month;
Query OK, 1571 rows affected (0.15 sec)
Records: 1571 Duplicates: 0 Warnings: 0

mysql>
mysql> INSERT INTO fact_table (sales, id_order_date, id_product, id_customer, id_country, id_employee)
-> SELECT
->     sd.sales,
->     od.id_order_date,
->     p.id_product,
->     cu.id_customer,
->     co.id_country,
->     e.id_employee
-> FROM sales_data sd
-> JOIN order_date od ON od.order_date = sd.order_date
-> JOIN product p ON p.product = sd.product
-> JOIN customer cu ON cu.customer = sd.customer
-> JOIN country co ON co.country = sd.country
-> JOIN employee e ON e.employee = sd.employee;
Query OK, 100000 rows affected (0.96 sec)
Records: 100000 Duplicates: 0 Warnings: 0
```

Imagen 11: Conteos de dimensiones y fact_table (SELECT con totales de productos, categorías, países y filas_fact).

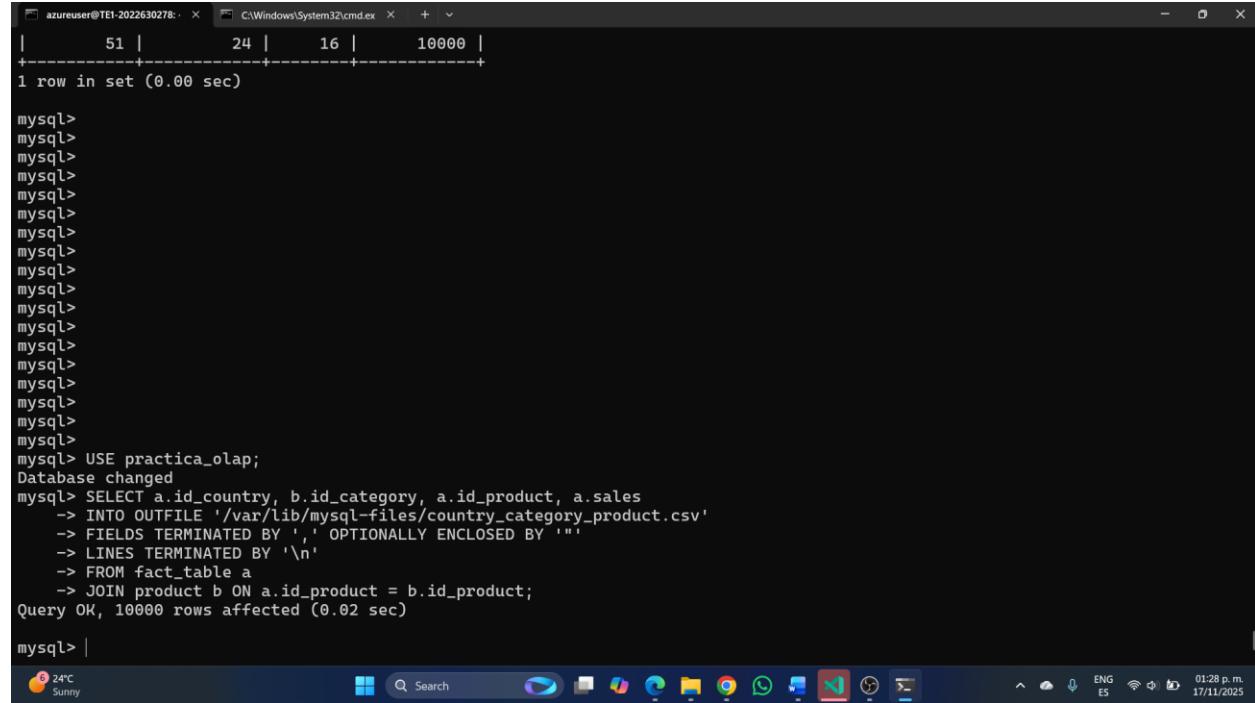
5.6 Generación del archivo country_category_product.csv

Se generó el archivo CSV de entrada para Hadoop mediante SELECT ... INTO OUTFILE en el directorio seguro de MySQL y se copió al HOME del usuario para facilitar su uso con Hadoop.

- En MySQL

USE practica.olap;

```
SELECT a.id_country, b.id_category, a.id_product, a.sales
INTO OUTFILE '/var/lib/mysql-files/country_category_product.csv'
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY ""
LINES TERMINATED BY '\n'
FROM fact_table a
JOIN product b ON a.id_product = b.id_product;
```



```
azuser@TE1-2022630278: ~ C:\Windows\System32\cmd.exe
+-----+
|      51 |      24 |      16 |      10000 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
mysql> USE practica.olap;
Database changed
mysql> SELECT a.id_country, b.id_category, a.id_product, a.sales
-> INTO OUTFILE '/var/lib/mysql-files/country_category_product.csv'
-> FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY ""
-> LINES TERMINATED BY '\n'
-> FROM fact_table a
-> JOIN product b ON a.id_product = b.id_product;
Query OK, 10000 rows affected (0.02 sec)

mysql> |
```

Imagen 11.1: Consulta a la base de datos.

- En la VM:

```
cp /var/lib/mysql-files/country_category_product.csv /home/azureuser/
```

```
ls -lh /var/lib/mysql-files/country_category_product.csv
```

```
head -n 5 /home/azureuser/country_category_product.csv
```

```
azureuser@TE1-2022630278:~$ sudo cp /var/lib/mysql-files/country_category_product.csv /home/azureuser/
azureuser@TE1-2022630278:~$ sudo chown azureuser:azureuser /home/azureuser/country_category_product.csv
azureuser@TE1-2022630278:~$ ls -lh /home/azureuser/country_category_product.csv
-rw-r-- 1 azureuser azureuser 144K Nov 17 19:37 /home/azureuser/country_category_product.csv
azureuser@TE1-2022630278:~$ head -n 5 /home/azureuser/country_category_product.csv
1,1,1,354.54
13,1,1,963.59
3,1,1,32.70
8,1,1,559.87
10,1,1,781.53
azureuser@TE1-2022630278:~$
azureuser@TE1-2022630278:~$ |
```

The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The user has run several commands to copy a CSV file from the MySQL files directory to their home folder, change its ownership to 'azureuser:azureuser', list its details with '-lh', and then display its first five lines with 'head -n 5'. The terminal shows the resulting file structure and the contents of the first five lines of the CSV.

Imagen 12: Evidencia del archivo country_category_product.csv (ls -lh y primeras líneas con head).

5.7 Instalación y configuración de Apache Hadoop

Se verificó la arquitectura de la VM y se descargó la distribución adecuada de Hadoop 3.4.0. Si la arquitectura fue x86_64, se utilizó el tarball genérico; si fue ARM, se usó el aarch64 indicado en la guía.

```
uname -m
```

```
cd /home/azureuser
```

```
azureuser@TE1-2022630278:~$ head -n 5 /home/azureuser/country_category_product.csv
1,1,1,354.54
13,1,1,963.59
3,1,1,32.70
8,1,1,559.87
10,1,1,781.53
azureuser@TE1-2022630278:~$
azureuser@TE1-2022630278:~$ uname -m
x86_64
azureuser@TE1-2022630278:~$ cd /home/azureuser
azureuser@TE1-2022630278:~$ |
```

Imagen 12.1: Arquitectura x86_64.

- x86_64 (típico en Azure)

wget <https://dlcdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz>

tar -xzf hadoop-3.4.0.tar.gz

```

azureuser@TE1-2022630278:~$ 
azureuser@TE1-2022630278:~$ uname -m
x86_64
azureuser@TE1-2022630278:~$ cd /home/azureuser
azureuser@TE1-2022630278:~$ wget https://dlcdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz
tar -xzf hadoop-3.4.0.tar.gz
--2025-11-17 19:42:50-- https://dlcdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response...
200 OK
Length: 965537117 (921M) [application/x-gzip]
Saving to: 'hadoop-3.4.0.tar.gz'

hadoop-3.4.0.tar.gz          100%[=====] 920.81M   231MB/s   in 4.0s

2025-11-17 19:43:29 (231 MB/s) - `hadoop-3.4.0.tar.gz' saved [965537117/965537117]

azureuser@TE1-2022630278:~$ 
azureuser@TE1-2022630278:~$ |

```

Imagen 12. 2: Instalar hadoop

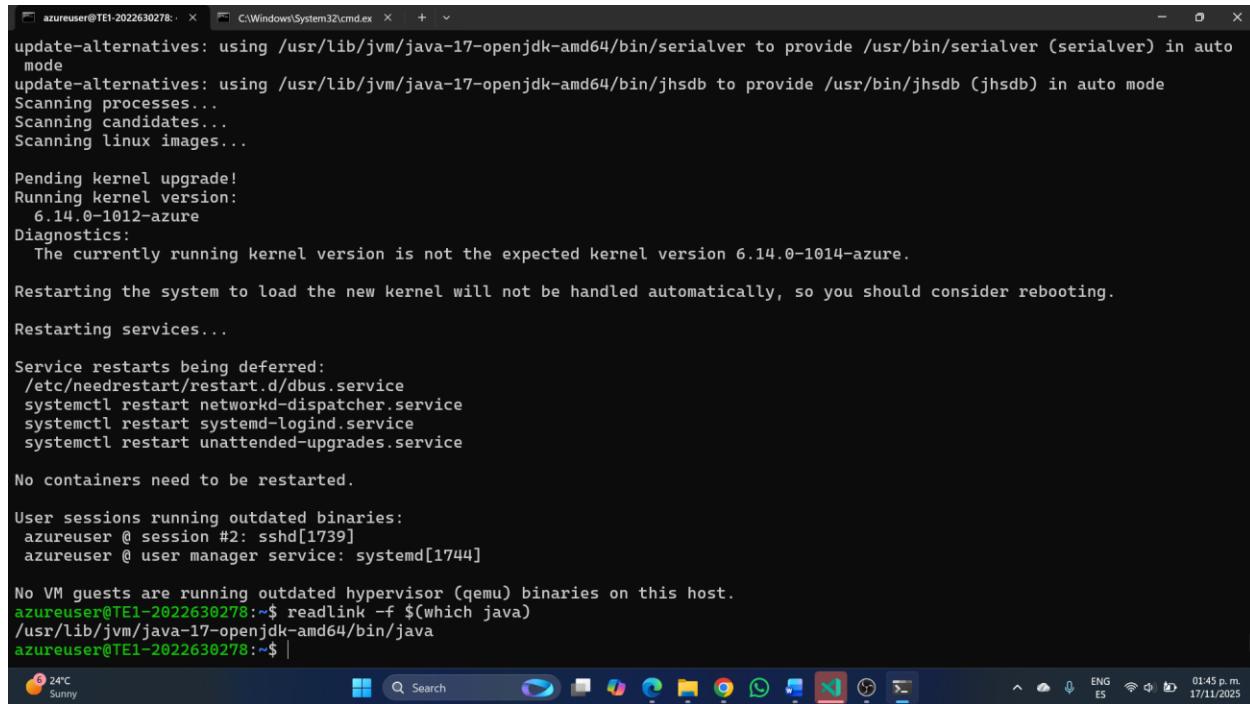
Se instaló Java según lo requerido por Hadoop (OpenJDK 16 o, en su defecto, 17), y se configuró JAVA_HOME en `hadoop-env.sh`.

```

sudo apt -y install openjdk-16-jdk-headless || sudo apt -y install openjdk-17-jdk-headless

readlink -f $(which java)

```



```
azureuser@TE1-2022630278: ~ C:\Windows\System32\cmd.exe + - x
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/serialver to provide /usr/bin/serialver (serialver) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jhsdb to provide /usr/bin/jhsdb (jhsdb) in auto mode
Scanning processes...
Scanning candidates...
Scanning linux images...

Pending kernel upgrade!
Running kernel version:
  6.14.0-1012-azure
Diagnostics:
  The currently running kernel version is not the expected kernel version 6.14.0-1014-azure.

Restarting the system to load the new kernel will not be handled automatically, so you should consider rebooting.

Restarting services...

Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
azureuser @ session #2: sshd[1739]
azureuser @ user manager service: systemd[1744]

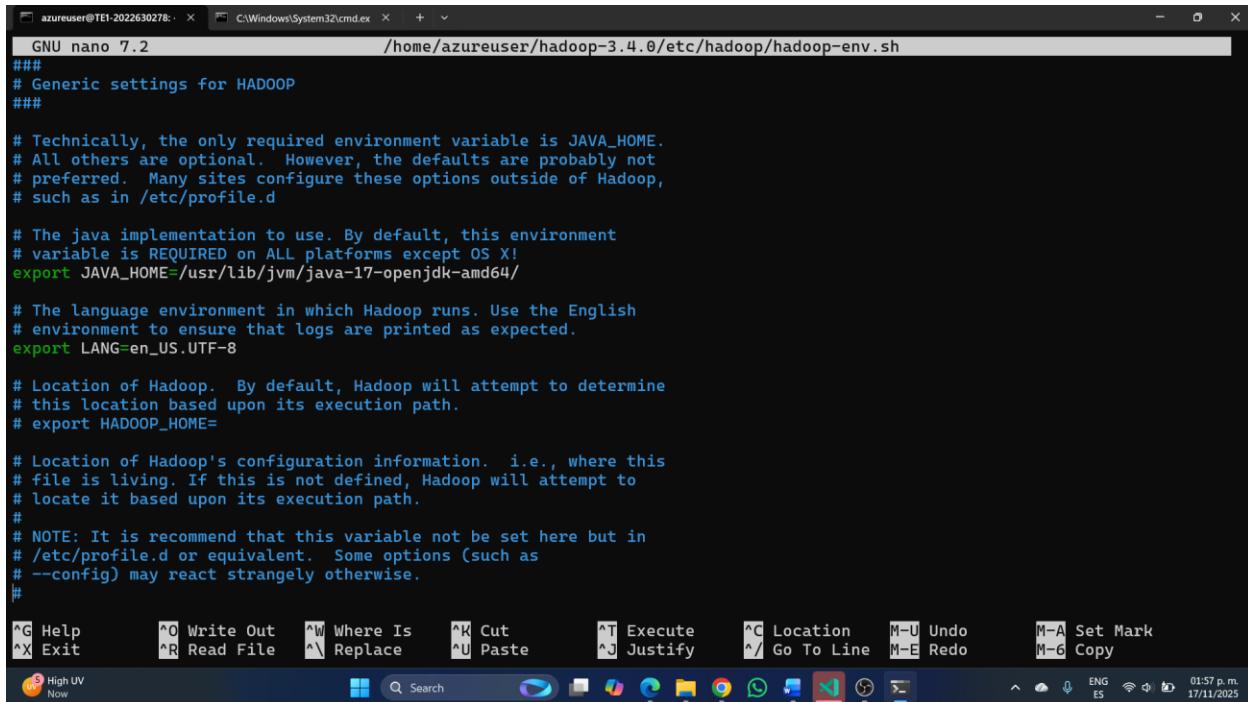
No VM guests are running outdated hypervisor (qemu) binaries on this host.
azureuser@TE1-2022630278:~$ readlink -f $(which java)
/usr/lib/jvm/java-17-openjdk-amd64/bin/java
azureuser@TE1-2022630278:~$ |
```

Imagen 12.3: Mostrar ubicación de java

Ejemplo típico: /usr/lib/jvm/java-17-openjdk-amd64/bin/java

nano /home/azureuser/hadoop-3.4.0/etc/hadoop/hadoop-env.sh

Se actualizó la línea: export JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64

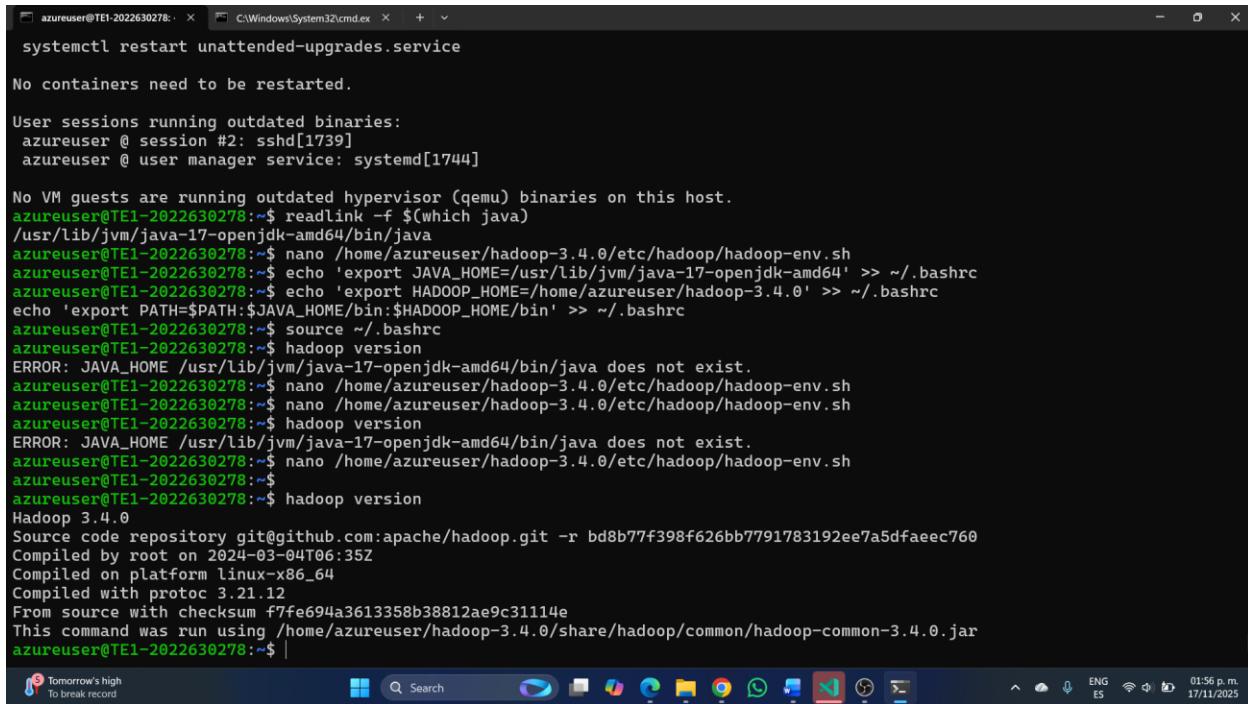


```
GNU nano 7.2          /home/azureuser/hadoop-3.4.0/etc/hadoop/hadoop-env.sh
###
# Generic settings for HADOOP
###
# Technically, the only required environment variable is JAVA_HOME.
# All others are optional. However, the defaults are probably not
# preferred. Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d
#
# The java implementation to use. By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
export JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64/
#
# The language environment in which Hadoop runs. Use the English
# environment to ensure that logs are printed as expected.
export LANG=en_US.UTF-8
#
# Location of Hadoop. By default, Hadoop will attempt to determine
# this location based upon its execution path.
# export HADOOP_HOME=
#
# Location of Hadoop's configuration information. i.e., where this
# file is living. If this is not defined, Hadoop will attempt to
# locate it based upon its execution path.
#
# NOTE: It is recommend that this variable not be set here but in
# /etc/profile.d or equivalent. Some options (such as
# --config) may react strangely otherwise.
#
^G Help      ^O Write Out   ^W Where Is    ^K Cut        ^T Execute     ^C Location    M-U Undo
^X Exit      ^R Read File   ^A Replace     ^U Paste       ^J Justify     ^V Go To Line  M-E Redo
                                         M-A Set Mark
                                         M-G Copy
High UV Now
Search
```

Imagen 12.4: Editar JAVA_HOME

Se exportaron variables de entorno para la sesión y para futuras sesiones:

```
echo 'export JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64' >> ~/.bashrc
echo 'export HADOOP_HOME=/home/azureuser/hadoop-3.4.0' >> ~/.bashrc
echo 'export PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin' >> ~/.bashrc
source ~/.bashrc
hadoop version
```



```
azureuser@TE1-2022630278:~$ systemctl restart unattended-upgrades.service
No containers need to be restarted.

User sessions running outdated binaries:
azureuser @ session #2: sshd[1739]
azureuser @ user manager service: systemd[1744]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
azureuser@TE1-2022630278:~$ readlink -f $(which java)
/usr/lib/jvm/java-17-openjdk-amd64/bin/java
azureuser@TE1-2022630278:~$ nano /home/azureuser/hadoop-3.4.0/etc/hadoop/hadoop-env.sh
azureuser@TE1-2022630278:~$ echo 'export JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64' >> ~/.bashrc
azureuser@TE1-2022630278:~$ echo 'export HADOOP_HOME=/home/azureuser/hadoop-3.4.0' >> ~/.bashrc
echo 'export PATH=$JAVA_HOME/bin:$HADOOP_HOME/bin' >> ~/.bashrc
azureuser@TE1-2022630278:~$ source ~/.bashrc
azureuser@TE1-2022630278:~$ hadoop version
ERROR: JAVA_HOME /usr/lib/jvm/java-17-openjdk-amd64/bin/java does not exist.
azureuser@TE1-2022630278:~$ nano /home/azureuser/hadoop-3.4.0/etc/hadoop/hadoop-env.sh
azureuser@TE1-2022630278:~$ nano /home/azureuser/hadoop-3.4.0/etc/hadoop/hadoop-env.sh
azureuser@TE1-2022630278:~$ hadoop version
ERROR: JAVA_HOME /usr/lib/jvm/java-17-openjdk-amd64/bin/java does not exist.
azureuser@TE1-2022630278:~$ nano /home/azureuser/hadoop-3.4.0/etc/hadoop/hadoop-env.sh
azureuser@TE1-2022630278:~$ hadoop version
Hadoop 3.4.0
Source code repository git@github.com:apache/hadoop.git -r bd8b77f398f626bb7791783192ee7a5dfaeecc760
Compiled by root on 2024-03-04T06:35Z
Compiled on platform linux-x86_64
Compiled with protoc 3.21.12
From source with checksum f7fe694a3613358b38812ae9c31114e
This command was run using /home/azureuser/hadoop-3.4.0/share/hadoop/common/hadoop-common-3.4.0.jar
azureuser@TE1-2022630278:~$ |
```

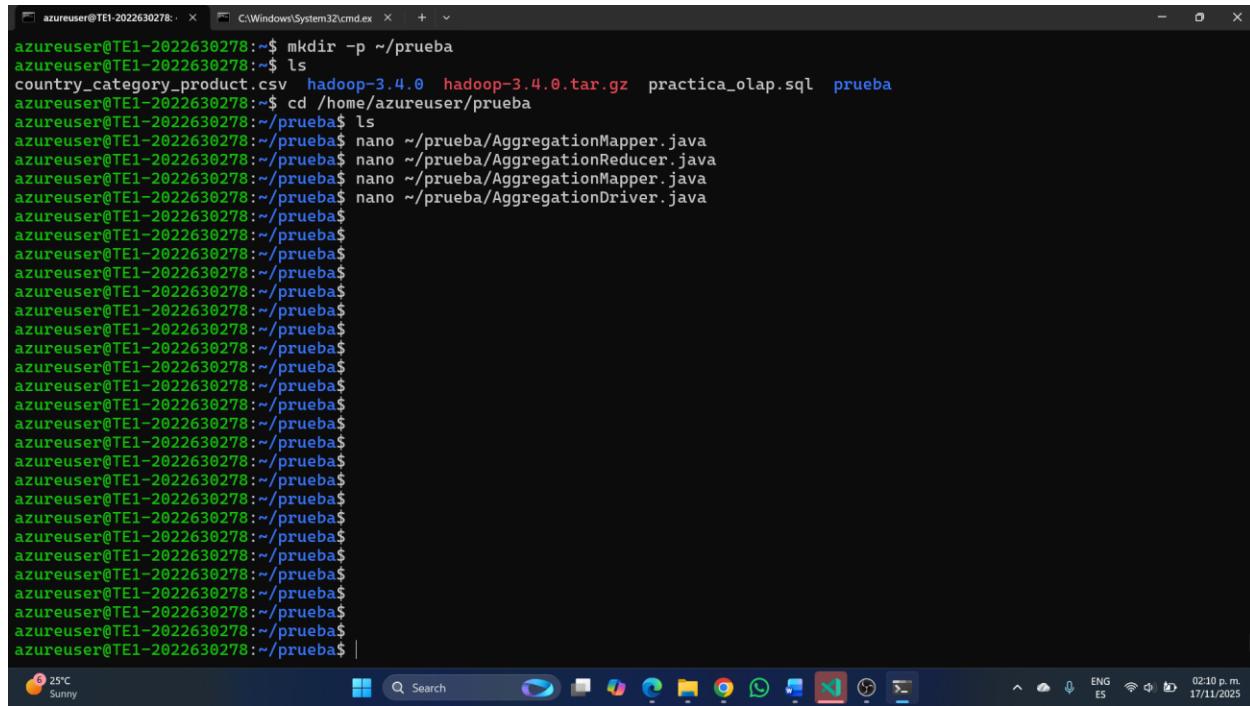
Imagen 13: Configuración de Hadoop y Java (salida de hadoop version y valor de JAVA_HOME; captura de hadoop-env.sh con export JAVA_HOME).

5.8 Implementación del job MapReduce

Se creó el directorio de trabajo y se implementaron las clases Java del Mapper, Reducer y Driver, conforme a lo visto en clase. Se compiló y se construyó el JAR de la aplicación.

mkdir -p /home/azureuser/prueba

cd /home/azureuser/prueba



```
azureuser@TE1-2022630278:~$ mkdir -p ~/prueba
azureuser@TE1-2022630278:~$ ls
country_category_product.csv  hadoop-3.4.0  hadoop-3.4.0.tar.gz  practica_olap.sql  prueba
azureuser@TE1-2022630278:~$ cd /home/azureuser/prueba
azureuser@TE1-2022630278:~/prueba$ ls
azureuser@TE1-2022630278:~/prueba$ nano ~/prueba/AggregationMapper.java
azureuser@TE1-2022630278:~/prueba$ nano ~/prueba/AggregationReducer.java
azureuser@TE1-2022630278:~/prueba$ nano ~/prueba/AggregationMapper.java
azureuser@TE1-2022630278:~/prueba$ nano ~/prueba/AggregationDriver.java
azureuser@TE1-2022630278:~/prueba$ 
```

Imagen 13.1: Creación de archivos java

AggregationMapper.java

The image shows two side-by-side terminal windows on a Windows desktop. Both windows have the title 'azuser@TE1-2022630278: ~' and the path 'C:\Windows\System32\cmd.exe'. The top window has the command 'GNU nano 7.2' and the file path '/home/azureuser/prueba/AggregationMapper.java'. The bottom window also has the command 'GNU nano 7.2' and the same file path. Both windows display the same Java code for an AggregationMapper class. The code includes imports for Java and Hadoop classes, a class definition with a map method, and a try-catch block for parsing the sales string into a double. The nano editor interface is visible at the bottom of each window, showing various keyboard shortcuts.

```

import java.io.IOException;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

// Entrada CSV: id_country,id_category,id_product,sales
public class AggregationMapper extends Mapper<LongWritable, Text, Text, DoubleWritable> {

    private final Text outKey = new Text();
    private final DoubleWritable outVal = new DoubleWritable();

    @Override
    protected void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
        String line = value.toString().trim();
        if (line.isEmpty()) return;

        String[] parts = line.split(",", -1);
        if (parts.length != 4) return;

        String idCountry = parts[0].trim();
        String idCategory = parts[1].trim();
        String idProduct = parts[2].trim();
        String salesStr = parts[3].trim();

        if (idCountry.isEmpty() || idCategory.isEmpty() || idProduct.isEmpty() || salesStr.isEmpty()) return;

        double sales;
        try {

```

Imagen 13.1: AggregationMapper.java

AggregationReducer.java

The screenshot shows a terminal window titled "azureuser@TE1-2022630278: ~" with a sub-titled "C:\Windows\System32\cmd.exe". The main pane displays Java code for an AggregationReducer class. The code imports necessary classes from java.io, org.apache.hadoop.io, and org.apache.hadoop.mapreduce. It defines a class AggregationReducer that extends Reducer<Text, DoubleWritable, Text, DoubleWritable>. The reduce method takes a Text key and an Iterable<DoubleWritable> values, and writes the sum of the values back to the context. The nano editor status bar at the bottom indicates "[Wrote 19 lines]". The terminal window also shows a weather icon for 25°C and a date/time stamp of 02:08 p.m. 17/11/2025.

```
GNU nano 7.2                               /home/azureuser/prueba/AggregationReducer.java
import java.io.IOException;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class AggregationReducer extends Reducer<Text, DoubleWritable, Text, DoubleWritable> {
    private final DoubleWritable result = new DoubleWritable();

    @Override
    protected void reduce(Text key, Iterable<DoubleWritable> values, Context context)
        throws IOException, InterruptedException {
        double sum = 0.0;
        for (DoubleWritable v : values) {
            sum += v.get();
        }
        result.set(sum);
        context.write(key, result);
    }
}
```

Imagen 13. 2: AggregationReducer.java

AggregationDriver.java

The screenshot shows a Windows terminal window titled "azuser@TE1-2022630278: ~" with the command "C:\Windows\System32\cmd.exe". The nano editor is open with the file "/home/azureuser/prueba/AggregationDriver.java". The code is a Java program for Hadoop aggregation. It includes imports for org.apache.hadoop.conf.Configuration, org.apache.hadoop.fs.Path, org.apache.hadoop.io.DoubleWritable, org.apache.hadoop.io.Text, org.apache.hadoop.mapreduce.Job, org.apache.hadoop.mapreduce.lib.input.TextInputFormat, and org.apache.hadoop.mapreduce.lib.output.TextOutputFormat. The main class is AggregationDriver, which has a main method that checks for two arguments, sets up a job configuration, and defines map and reduce classes. It also sets map and reduce output classes and adds an input path. The nano editor status bar indicates "[Wrote 37 lines]". The taskbar at the bottom shows various application icons.

```
GNU nano 7.2 /home/azureuser/prueba/AggregationDriver.java
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class AggregationDriver {
    public static void main(String[] args) throws Exception {
        if (args.length != 2) {
            System.out.println("Uso: AggregationDriver <input_dir> <output_dir>");
            System.exit(1);
        }

        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Country-Category-Product Aggregation");

        job.setJarByClass(AggregationDriver.class);
        job.setMapperClass(AggregationMapper.class);
        job.setReducerClass(AggregationReducer.class);

        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(DoubleWritable.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(DoubleWritable.class);

        TextInputFormat.addInputPath(job, new Path(args[0]));
        TextOutputFormat.setOutputPath(job, new Path(args[1]));

        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);

        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

This screenshot is identical to the one above, showing the same Java code for AggregationDriver.java in the nano editor. The code is identical, and the terminal environment is the same, indicating a copy-and-paste operation or a very similar setup.

```
GNU nano 7.2 /home/azureuser/prueba/AggregationDriver.java
public static void main(String[] args) throws Exception {
    if (args.length != 2) {
        System.out.println("Uso: AggregationDriver <input_dir> <output_dir>");
        System.exit(1);
    }

    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "Country-Category-Product Aggregation");

    job.setJarByClass(AggregationDriver.class);
    job.setMapperClass(AggregationMapper.class);
    job.setReducerClass(AggregationReducer.class);

    job.setMapOutputKeyClass(Text.class);
    job.setMapOutputValueClass(DoubleWritable.class);

    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(DoubleWritable.class);

    TextInputFormat.addInputPath(job, new Path(args[0]));
    TextOutputFormat.setOutputPath(job, new Path(args[1]));

    job.setInputFormatClass(TextInputFormat.class);
    job.setOutputFormatClass(TextOutputFormat.class);

    System.exit(job.waitForCompletion(true) ? 0 : 1);
}
```

Imagen 13.3: AggregationDriver.java

mkdir -p build

javac -classpath "\$(hadoop classpath)" -d build *.java

```
jar -cvf Aggregation.jar -C build .
```

```
ls -lh Aggregation.jar
```

The screenshot shows a Windows Command Prompt window titled "azureuser@TE1-2022630278: ~ /prueba\$". The command history at the top of the window shows multiple instances of the command "jar -cvf Aggregation.jar -C build .". The user then runs "javac -classpath \$(hadoop classpath) -d build *.java", followed by "jar -cvf Aggregation.jar -C build .". The output indicates that a manifest was added and several classes were added with their respective sizes and compression ratios. Finally, "ls -lh Aggregation.jar" is run, showing the file's permissions as "-rw-rw-r--" and its size as 3.5K. The system tray at the bottom right shows the date as 17/11/2025 and the time as 20:11.

Imagen 14: Compilación y empaquetado exitosos (salida de javac/jar y listado mostrando Aggregation.jar).

5.9 Ejecución del job MapReduce y verificación del resultado

Se preparó el directorio de entrada con el CSV generado desde MySQL y se ejecutó el job en modo local (standalone). Se visualizó la salida en part-r-00000 y se normalizó el separador a coma para su posterior carga.

```
cd /home/azureuser/prueba
```

```
mkdir -p input
```

```
cp /home/azureuser/country_category_product.csv input/
```

```
rm -rf output
```

```
hadoop jar Aggregation.jar AggregationDriver input output
```

```
azureuser@TE1-2822630278:~/prueba$ cd /home/azureuser/prueba
azureuser@TE1-2822630278:~/prueba$ mkdir -p input
azureuser@TE1-2822630278:~/prueba$ cp /home/azureuser/country_category_product.csv input/
azureuser@TE1-2822630278:~/prueba$ rm -rf output
azureuser@TE1-2822630278:~/prueba$ hadoop jar Aggregation.jar AggregationDriver input output
2025-11-17 20:22:18,598 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-11-17 20:22:18,718 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-11-17 20:22:18,719 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2025-11-17 20:22:18,792 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2025-11-17 20:22:18,870 INFO input.FileInputFormat: Total input files to process : 1
2025-11-17 20:22:18,890 INFO mapreduce.JobSubmitter: number of splits:1
2025-11-17 20:22:19,087 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local2009366042_0001
2025-11-17 20:22:19,087 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-11-17 20:22:19,243 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2025-11-17 20:22:19,244 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2025-11-17 20:22:19,244 INFO mapreduce.Job: Running job: job_local2009366042_0001
2025-11-17 20:22:19,252 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2025-11-17 20:22:19,253 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-11-17 20:22:19,253 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-11-17 20:22:19,254 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2025-11-17 20:22:19,286 INFO mapred.LocalJobRunner: Waiting for map tasks
2025-11-17 20:22:19,286 INFO mapred.LocalJobRunner: Starting task: attempt_local2009366042_0001_m_000000_0
2025-11-17 20:22:19,306 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2025-11-17 20:22:19,306 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-11-17 20:22:19,306 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-11-17 20:22:19,332 INFO mapred.Task: Using ResourceCalculatorProcessTree : []
2025-11-17 20:22:19,336 INFO mapred.MapTask: Processing split: file:/home/azureuser/prueba/input/country_category_product.csv
```

ls -lh output

head -n 10 output/part-r-00000

Se convirtió el tabulador a coma:

sed -i 's/\t/,/g' output/part-r-00000

head -n 10 output/part-r-00000

```

Spilled Records=20000
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=8
Total committed heap usage (bytes)=434110464
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=147286
File Output Format Counters
Bytes Written=16825
azureuser@TE1-2022630278:~/prueba$ ls -lh output
total 20k
-rw-r--r-- 1 azureuser azureuser 0 Nov 17 20:22 _SUCCESS
-rw-r--r-- 1 azureuser azureuser 17K Nov 17 20:22 part-r-00000
azureuser@TE1-2022630278:~/prueba$ head -n 10 output/part-r-00000
1,1,1 7187.64
1,1,11 5791.529999999999
1,10,17 4492.640000000001
1,10,32 7046.429999999999
1,11,19 4471.83
1,11,43 6772.27
1,12,20 7562.839999999999
1,12,30 3522.750000000005
1,13,21 6034.699999999999
1,13,38 9237.73
azureuser@TE1-2022630278:~/prueba$ |

```

Imagen 15: Ejecución del job (salida con counters/resumen y existencia de output/part-r-00000).

```

WRONG_REDUCE=0
File Input Format Counters
Bytes Read=147286
File Output Format Counters
Bytes Written=16825
azureuser@TE1-2022630278:~/prueba$ ls -lh output
total 20k
-rw-r--r-- 1 azureuser azureuser 0 Nov 17 20:22 _SUCCESS
-rw-r--r-- 1 azureuser azureuser 17K Nov 17 20:22 part-r-00000
azureuser@TE1-2022630278:~/prueba$ head -n 10 output/part-r-00000
1,1,1 7187.64
1,1,11 5791.529999999999
1,10,17 4492.640000000001
1,10,32 7046.429999999999
1,11,19 4471.83
1,11,43 6772.27
1,12,20 7562.839999999999
1,12,30 3522.750000000005
1,13,21 6034.699999999999
1,13,38 9237.73
azureuser@TE1-2022630278:~/prueba$ sed -i 's/\t/,/g' output/part-r-00000
azureuser@TE1-2022630278:~/prueba$ head -n 10 output/part-r-00000
1,1,1,7187.64
1,1,11,5791.529999999999
1,10,17,4492.640000000001
1,10,32,7046.429999999999
1,11,19,4471.83
1,11,43,6772.27
1,12,20,7562.839999999999
1,12,30,3522.750000000005
1,13,21,6034.699999999999
1,13,38,9237.73
azureuser@TE1-2022630278:~/prueba$ |

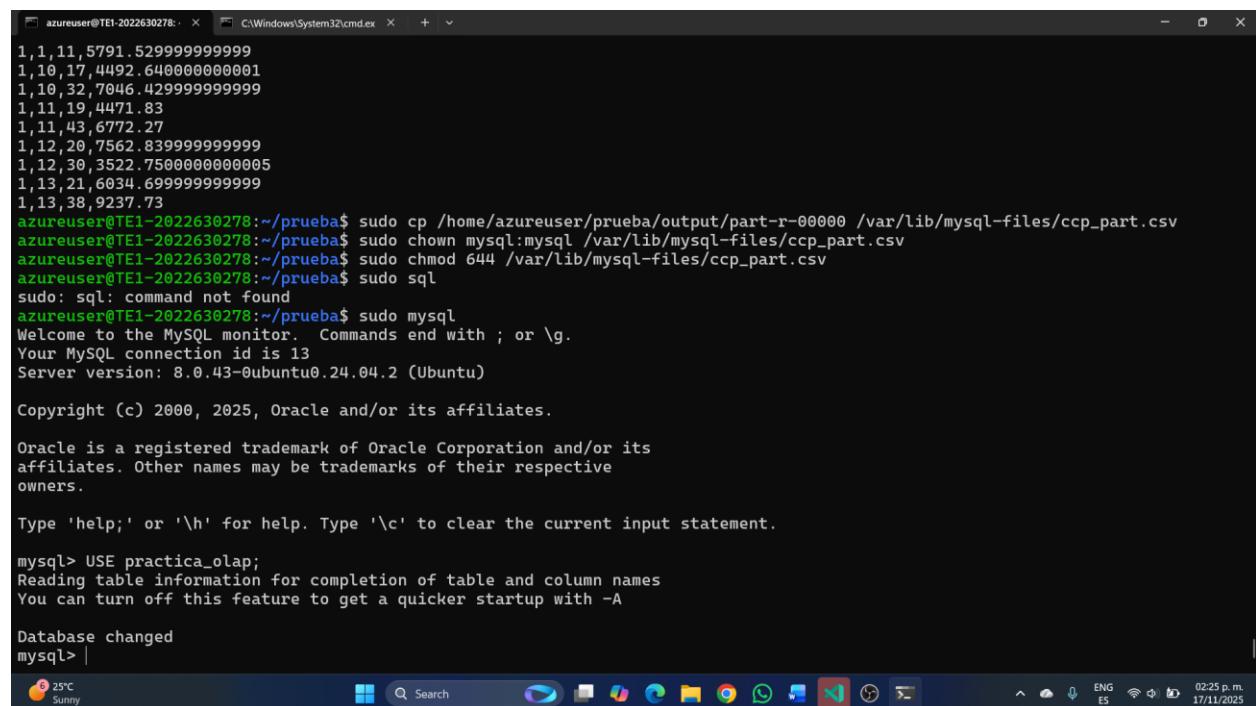
```

Imagen 16: Contenido de part-r-00000 antes y después de sed (dos capturas mostrando el cambio de tabulador a coma).

5.10 Carga del resultado a la tabla agregada country_category_product

Se colocó el archivo final en el directorio seguro de MySQL y se cargó a la tabla agregada. Se verificó la cantidad de filas.

```
sudo cp /home/azureuser/prueba/output/part-r-00000 /var/lib/mysql-files/ccp_part.csv  
sudo chown mysql:mysql /var/lib/mysql-files/ccp_part.csv  
sudo chmod 644 /var/lib/mysql-files/ccp_part.csv
```



The screenshot shows a Windows terminal window titled 'C:\Windows\System32\cmd.exe'. It displays the command-line steps to copy a CSV file from a user directory to the MySQL data directory, change ownership, and set permissions. It then shows the MySQL command-line interface (mysql) being opened and the 'practica.olap' database being selected. Finally, it shows the execution of a LOAD DATA INFILE command to load the data into the 'country_category_product' table.

```
1,1,11,5791.529999999999  
1,10,17,4492.640000000001  
1,10,32,7046.429999999999  
1,11,19,4471.83  
1,11,43,6772.27  
1,12,20,7562.839999999999  
1,12,30,3522.750000000005  
1,13,21,6034.699999999999  
1,13,38,9237.73  
azureuser@TE1-2022630278:~/prueba$ sudo cp /home/azureuser/prueba/output/part-r-00000 /var/lib/mysql-files/ccp_part.csv  
azureuser@TE1-2022630278:~/prueba$ sudo chown mysql:mysql /var/lib/mysql-files/ccp_part.csv  
azureuser@TE1-2022630278:~/prueba$ sudo chmod 644 /var/lib/mysql-files/ccp_part.csv  
azureuser@TE1-2022630278:~/prueba$ sudo sql  
sudo: sql: command not found  
azureuser@TE1-2022630278:~/prueba$ sudo mysql  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 13  
Server version: 8.0.43-0ubuntu0.24.04.2 (Ubuntu)  
  
Copyright (c) 2000, 2025, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> USE practica.olap;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
mysql> |  
25°C Sunny 02:25 p.m. ENG ES 17/11/2025
```

Imagen 16: Carga del resultado a la tabla agregada

- En MySQL

```
USE practica.olap;
```

```
TRUNCATE TABLE country_category_product;
```

```
LOAD DATA INFILE '/var/lib/mysql-files/ccp_part.csv'
```

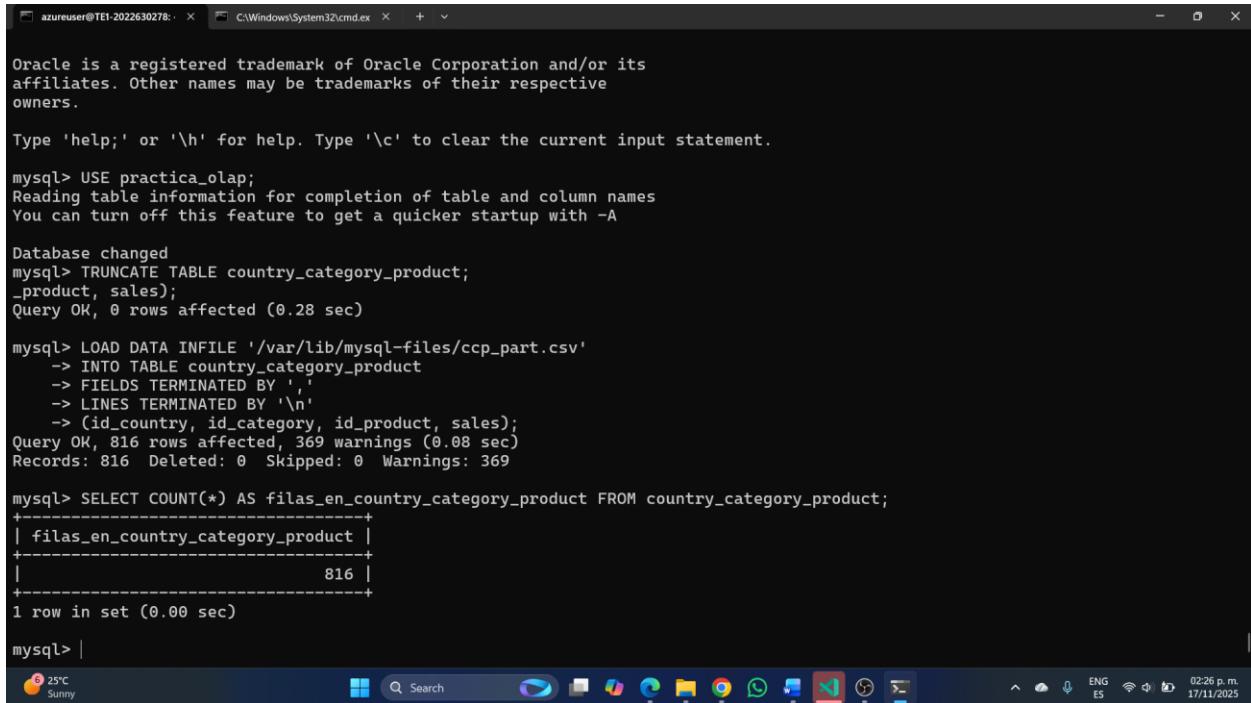
```
INTO TABLE country_category_product
```

```
FIELDS TERMINATED BY ','
```

LINES TERMINATED BY '\n'

(id_country, id_category, id_product, sales);

```
SELECT      COUNT(*)      AS      filas_en_country_category_product      FROM
country_category_product;
```



```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE practica.olap;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> TRUNCATE TABLE country_category_product;
 _product, sales);
Query OK, 0 rows affected (0.28 sec)

mysql> LOAD DATA INFILE '/var/lib/mysql-files/ccp_part.csv'
    -> INTO TABLE country_category_product
    -> FIELDS TERMINATED BY ','
    -> LINES TERMINATED BY '\n'
    -> (id_country, id_category, id_product, sales);
Query OK, 816 rows affected, 369 warnings (0.08 sec)
Records: 816  Deleted: 0  Skipped: 0  Warnings: 369

mysql> SELECT COUNT(*) AS filas_en_country_category_product FROM country_category_product;
+-----+
| filas_en_country_category_product |
+-----+
|          816 |
+-----+
1 row in set (0.00 sec)

mysql> |
```

Imagen 17: Carga a la tabla agregada y verificación (SELECT COUNT(*) FROM country_category_product mostrando el total de filas).

6 Consultas OLAP (usando la tabla agregada)

En esta sección se ejecutan los tres cubos solicitados, exclusivamente sobre la tabla agregada country_category_product. Cada consulta muestra la suma de ventas (sales) y se acompaña de su evidencia. Toma la captura de la “primera pantalla” de resultados en MySQL para cada caso.

Sugerencia: para que la salida sea legible, ordena por el total y limita las primeras filas si tu terminal imprime demasiadas líneas. Se usa ROUND(...,2) para mostrar dos decimales.

6.1 Acumulado de sales por country

Consulta que suma las ventas por país, aprovechando el join entre la tabla agregada y la dimensión country.

```
SELECT c.country,  
       ROUND(SUM(ccp.sales), 2) AS total_sales  
  FROM country_category_product ccp  
 JOIN country c  
    ON c.id_country = ccp.id_country  
 GROUP BY c.country  
 ORDER BY total_sales DESC;
```

```

| azureuser@TE1-2022630278: ~ | C:\Windows\System32\cmd.exe | + - x
+-----+
1 row in set (0.00 sec)

mysql> SELECT c.country,
->           ROUND(SUM(ccp.sales), 2) AS total_sales
->     FROM country_category_product ccp
->   JOIN country c
->     ON c.id_country = ccp.id_country
->   GROUP BY c.country
->   ORDER BY total_sales DESC;
+-----+-----+
| country | total_sales |
+-----+-----+
| South Korea | 343001.89 |
| Egypt | 330573.77 |
| Japan | 327858.27 |
| China | 323593.10 |
| Mexico | 320266.32 |
| Kenya | 317248.97 |
| USA | 317208.35 |
| Germany | 311998.46 |
| Spain | 309176.01 |
| India | 307337.33 |
| South Africa | 304864.92 |
| Canada | 303541.82 |
| Italy | 301304.05 |
| Brazil | 300279.73 |
| France | 289453.69 |
| Nigeria | 281136.94 |
+-----+-----+
16 rows in set (0.00 sec)

mysql> |

```

0 25°C Sunny 02:27 p.m.
ENG ES 17/11/2025

Imagen 18. Resultado de la consulta de acumulado de ventas por país (primera pantalla).

6.2 Acumulado de sales por country y category

Consulta que suma las ventas por país y categoría, uniendo la tabla agregada con las dimensiones country y category.

```

SELECT c.country,
       cat.category,
       ROUND(SUM(ccp.sales), 2) AS total_sales
  FROM country_category_product ccp
 JOIN country c
   ON c.id_country = ccp.id_country
 JOIN category cat
   ON cat.id_category = ccp.id_category

```

```

GROUP BY c.country, cat.category
ORDER BY c.country ASC, total_sales DESC;

```

```

mysql> SELECT c.country,
->         cat.category,
->         ROUND(SUM(ccp.sales), 2) AS total_sales
->     FROM country_category_product ccp
->     JOIN country c
->       ON c.id_country = ccp.id_country
->     JOIN category cat
->       ON cat.id_category = ccp.id_category
->   GROUP BY c.country, cat.category
-> ORDER BY c.country ASC, total_sales DESC;
+-----+-----+-----+
| country | category | total_sales |
+-----+-----+-----+
| Brazil  | Mobile Devices | 29729.86 |
| Brazil  | Audio Equipment | 18667.11  |
| Brazil  | Computers      | 18065.85  |
| Brazil  | Outdoor Gear    | 16445.10  |
| Brazil  | Storage Devices | 16362.15  |
| Brazil  | Books          | 14778.31  |
| Brazil  | Home Appliances | 14632.82  |
| Brazil  | Wearables      | 14229.26  |
| Brazil  | Computer Accessories | 13150.81 |
| Brazil  | Fitness Equipment | 12146.79  |
| Brazil  | Musical Instruments | 11985.49 |
| Brazil  | Footwear        | 11520.12  |
| Brazil  | Monitors         | 11286.43  |
| Brazil  | Pet Supplies    | 11149.34  |
| Brazil  | Art              | 10795.02  |
| Brazil  | Office Furniture | 10677.60  |
| Brazil  | School Supplies | 10476.29  |
| Brazil  | Kitchen Appliances | 10416.94 |
| Brazil  | Home Decor       | 9979.56  |
+-----+-----+-----+

```

Imagen 19. Resultado de la consulta de acumulado de ventas por país y categoría (primera pantalla).

6.3 Acumulado de sales por country, category y product

Consulta que desglosa las ventas hasta nivel de producto, combinando las tres dimensiones con la tabla agregada.

```

SELECT c.country,
       cat.category,
       p.product,
       ROUND(SUM(ccp.sales), 2) AS total_sales
FROM country_category_product ccp
JOIN country c

```

```

ON c.id_country = ccp.id_country
JOIN category cat
ON cat.id_category = ccp.id_category
JOIN product p
ON p.id_product = ccp.id_product
GROUP BY c.country, cat.category, p.product
ORDER BY c.country ASC, cat.category ASC, total_sales DESC;

```

```

mysql> SELECT c.country,
->         cat.category,
->         p.product,
->         ROUND(SUM(ccp.sales), 2) AS total_sales
->     FROM country_category_product ccp
->     JOIN country c
->     ON c.id_country = ccp.id_country
->     JOIN category cat
->     ON cat.id_category = ccp.id_category
->     JOIN product p
->     ON p.id_product = ccp.id_product
->     GROUP BY c.country, cat.category, p.product
->     ORDER BY c.country ASC, cat.category ASC, total_sales DESC;
+-----+-----+-----+-----+
| country | category | product | total_sales |
+-----+-----+-----+-----+
| Brazil | Art      | Landscape Painting | 6189.27 |
| Brazil | Art      | Sculpture          | 4665.75 |
| Brazil | Audio Equipment | Bluetooth Speaker | 10545.97 |
| Brazil | Audio Equipment | Noise Cancelling Headphones | 8121.14 |
| Brazil | Books    | Science Fiction Novel | 7945.72 |
| Brazil | Books    | Biography Book       | 6832.59 |
| Brazil | Computer Accessories | USB Keyboard | 9574.01 |
| Brazil | Computer Accessories | Wireless Mouse | 3576.80 |
| Brazil | Computers | Notebook Professional | 5990.04 |
| Brazil | Computers | Desktop Beta        | 4582.72 |
| Brazil | Computers | Notebook Basic       | 3863.20 |
| Brazil | Computers | Desktop Alpha        | 3629.89 |
| Brazil | Electronics | Virtual Reality Headset | 4852.41 |
| Brazil | Electronics | Drone with Camera     | 2474.75 |
| Brazil | Fashion Accessories | Wristwatch | 3790.02 |
| Brazil | Fashion Accessories | Sunglasses | 2556.23 |
+-----+-----+-----+-----+

```

Imagen 20. Resultado de la consulta de acumulado de ventas por país, categoría y producto (primera pantalla).

7 Enlace al chat de la IA GitHub Copilot

El enlace al chat de la IA GitHub Copilot que usamos para el desarrollo de esta práctica es el siguiente:

- <https://github.com/copilot/share/881c0286-0a44-8ca7-8811-a00604304183>

8 Conclusiones

La práctica demostró un flujo completo y reproducible de generación de tablas agregadas apoyado en MySQL y Hadoop MapReduce, desplegado en una sola VM de Azure. Se verificó que, para consultas analíticas recurrentes, operar sobre una tabla agregada reduce el volumen de datos a escanear y simplifica las uniones, mejorando la latencia y el consumo de recursos frente a consultar directamente la fact_table.

La integración entre MySQL y Hadoop funcionó de forma fluida usando archivos intermedios CSV, y la ejecución en modo local de Hadoop fue suficiente para los objetivos del laboratorio, evitando la complejidad de HDFS/YARN. La etapa crítica consistió en cuidar el secure-file-priv (para INFILE/OUTFILE) y la correcta configuración del JAVA_HOME en hadoop-env.sh; ambas resultaron determinantes para evitar errores de permisos y de clase en tiempo de ejecución.

En términos de aprendizaje, se consolidaron:

- El diseño dimensional básico (dimensiones, fechas, fact_table) y su relación con tablas agregadas para OLAP.
- La implementación de un job MapReduce en Java (Mapper/Reducer/Driver) para agregaciones por claves compuestas.

9 Anexos

- **AggregationMapper.java**

```
import java.io.IOException;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

// Entrada CSV: id_country,id_category,id_product,sales
public class AggregationMapper extends Mapper<LongWritable, Text, Text,
DoubleWritable> {

    private final Text outKey = new Text();
    private final DoubleWritable outVal = new DoubleWritable();

    @Override
    protected void map(LongWritable key, Text value, Context context) throws
IOException, InterruptedException {
        String line = value.toString().trim();
        if (line.isEmpty()) return;

        String[] parts = line.split(", ", -1);
        if (parts.length != 4) return;

        String idCountry = parts[0].trim();
        String idCategory = parts[1].trim();
        String idProduct = parts[2].trim();
```

```

String salesStr = parts[3].trim();

if (idCountry.isEmpty() || idCategory.isEmpty() || idProduct.isEmpty() ||
salesStr.isEmpty()) return;

double sales;
try {
    sales = Double.parseDouble(salesStr);
} catch (NumberFormatException e) {
    return;
}

outKey.set(idCountry + "," + idCategory + "," + idProduct);
outVal.set(sales);
context.write(outKey, outVal);
}
}

```

- **AggregationReducer.java**

```

import java.io.IOException;

import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class AggregationReducer extends Reducer<Text, DoubleWritable, Text,
DoubleWritable> {

    private final DoubleWritable result = new DoubleWritable();

    @Override

```

```

protected void reduce(Text key, Iterable<DoubleWritable> values, Context context)
    throws IOException, InterruptedException {
    double sum = 0.0;
    for (DoubleWritable v : values) {
        sum += v.get();
    }
    result.set(sum);
    context.write(key, result);
}

```

- **AggregationDriver.java**

```

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

```

```
public class AggregationDriver {
```

```

    public static void main(String[] args) throws Exception {
        if (args.length != 2) {
            System.err.println("Uso: AggregationDriver <input_dir> <output_dir>");
            System.exit(1);
    }

```

```
    Configuration conf = new Configuration();
```

```

Job job = Job.getInstance(conf, "Country-Category-Product Aggregation");

job.setJarByClass(AggregationDriver.class);

job.setMapperClass(AggregationMapper.class);
job.setReducerClass(AggregationReducer.class);

job.setMapOutputKeyClass(Text.class);
job.setMapOutputValueClass(DoubleWritable.class);

job.setOutputKeyClass(Text.class);
job.setOutputValueClass(DoubleWritable.class);

TextInputFormat.addInputPath(job, new Path(args[0]));
TextOutputFormat.setOutputPath(job, new Path(args[1]));

job.setInputFormatClass(TextInputFormat.class);
job.setOutputFormatClass(TextOutputFormat.class);

System.exit(job.waitForCompletion(true) ? 0 : 1);
}

}

• olap_workflow.sql

-- Asume: CREATE DATABASE practica.olap; y SOURCE practica.olap.sql ya ejecutados

USE practica.olap;

```

```
-- Carga del CSV a sales_data (archivo colocado en /var/lib/mysql-files)
LOAD DATA INFILE '/var/lib/mysql-files/sales_data.csv'
INTO TABLE sales_data
FIELDS TERMINATED BY ',' ENCLOSED BY ""
IGNORE 1 LINES
(sales, order_date, product, customer, country, region, employee, category, weekday,
month, quarter, semester);
```

-- Dimensiones

```
INSERT INTO region (region)
SELECT DISTINCT region FROM sales_data WHERE region IS NOT NULL;
```

```
INSERT INTO country (country, id_region)
SELECT DISTINCT sd.country, r.id_region
FROM sales_data sd
JOIN region r ON r.region = sd.region;
```

```
INSERT INTO customer (customer)
```

```
SELECT DISTINCT customer FROM sales_data WHERE customer IS NOT NULL;
```

```
INSERT INTO employee (employee)
```

```
SELECT DISTINCT employee FROM sales_data WHERE employee IS NOT NULL;
```

```
INSERT INTO semester (semester)
```

```
SELECT DISTINCT semester FROM sales_data WHERE semester IS NOT NULL;
```

```
INSERT INTO quarter (quarter, id_semester)
```

```
SELECT DISTINCT sd.quarter, s.id_semester  
FROM sales_data sd  
JOIN semester s ON s.semester = sd.semester;
```

```
INSERT INTO month (month, id_quarter)  
SELECT DISTINCT sd.month, q.id_quarter  
FROM sales_data sd  
JOIN quarter q ON q.quarter = sd.quarter;
```

```
INSERT INTO weekday (weekday)  
SELECT DISTINCT weekday FROM sales_data WHERE weekday IS NOT NULL;
```

```
INSERT INTO category (category)  
SELECT DISTINCT category FROM sales_data WHERE category IS NOT NULL;
```

```
INSERT INTO product (product, id_category)  
SELECT DISTINCT sd.product, c.id_category  
FROM sales_data sd  
JOIN category c ON c.category = sd.category;
```

```
INSERT INTO order_date (order_date, id_weekday, id_month)  
SELECT DISTINCT sd.order_date, w.id_weekday, m.id_month  
FROM sales_data sd  
JOIN weekday w ON w.weekday = sd.weekday  
JOIN month m ON m.month = sd.month;
```

-- Fact table

```

INSERT INTO fact_table (sales, id_order_date, id_product, id_customer, id_country,
id_employee)

SELECT
    sd.sales,
    od.id_order_date,
    p.id_product,
    cu.id_customer,
    co.id_country,
    e.id_employee

FROM sales_data sd
JOIN order_date od ON od.order_date = sd.order_date
JOIN product p ON p.product = sd.product
JOIN customer cu ON cu.customer = sd.customer
JOIN country co ON co.country = sd.country
JOIN employee e ON e.employee = sd.employee;

```

```

-- Exportación a CSV para MapReduce
SELECT a.id_country, b.id_category, a.id_product, a.sales
INTO OUTFILE '/var/lib/mysql-files/country_category_product.csv'
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY ""
LINES TERMINATED BY '\n'
FROM fact_table a
JOIN product b ON a.id_product = b.id_product;

```

```

-- Importación del resultado de MapReduce a la tabla agregada
-- (asume que /var/lib/mysql-files/ccp_part.csv ya existe y usa comas)
TRUNCATE TABLE country_category_product;

```

```

LOAD DATA INFILE '/var/lib/mysql-files/ccp_part.csv'
INTO TABLE country_category_product
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
(id_country, id_category, id_product, sales);

-- Consultas de cubos
-- 1) Ventas por country
SELECT c.country, ROUND(SUM(ccp.sales), 2) AS total_sales
FROM country_category_product ccp
JOIN country c ON c.id_country = ccp.id_country
GROUP BY c.country
ORDER BY total_sales DESC;

-- 2) Ventas por country y category
SELECT c.country, cat.category, ROUND(SUM(ccp.sales), 2) AS total_sales
FROM country_category_product ccp
JOIN country c ON c.id_country = ccp.id_country
JOIN category cat ON cat.id_category = ccp.id_category
GROUP BY c.country, cat.category
ORDER BY c.country, total_sales DESC;

-- 3) Ventas por country, category y product
SELECT c.country, cat.category, p.product, ROUND(SUM(ccp.sales), 2) AS total_sales
FROM country_category_product ccp
JOIN country c ON c.id_country = ccp.id_country
JOIN category cat ON cat.id_category = ccp.id_category

```

```
JOIN product p ON p.id_product = ccp.id_product  
GROUP BY c.country, cat.category, p.product  
ORDER BY c.country, cat.category, total_sales DESC;
```

- **hadoop-env-snippet.sh**

```
# Ruta ejemplo en Ubuntu 22.04 con OpenJDK 17  
export JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64
```

- **commands.sh**

```
# Conexión y transferencia
```

```
ssh azureuser@<IP_PUBLICA>  
sftp azureuser@<IP_PUBLICA>  
put practica_olap.sql  
put sales_data.csv  
exit
```

```
# Paquetes
```

```
sudo apt update && sudo apt -y upgrade  
sudo apt -y install mysql-server mysql-client openjdk-17-jdk-headless wget tar nano
```

```
# MySQL secure-file-priv
```

```
sudo mkdir -p /var/lib/mysql-files  
sudo chown mysql:mysql /var/lib/mysql-files  
sudo chmod 750 /var/lib/mysql-files
```

```
# Hadoop 3.4.0 (x86_64)
```

```
cd ~
```

```
wget https://dlcdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz  
tar -xzf hadoop-3.4.0.tar.gz
```

```
# Variables de entorno

echo 'export JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64' >> ~/.bashrc
echo 'export HADOOP_HOME=$HOME/hadoop-3.4.0' >> ~/.bashrc
echo 'export PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin' >> ~/.bashrc
source ~/.bashrc

# Compilación y ejecución MapReduce

mkdir -p ~/prueba
cd ~/prueba
javac -classpath "$(hadoop classpath)" -d build *.java
jar -cvf Aggregation.jar -C build .
mkdir -p input
cp ~/country_category_product.csv input/
rm -rf output
hadoop jar Aggregation.jar AggregationDriver input output
sed -i 's/\t/,/g' output/part-r-00000
sudo cp output/part-r-00000 /var/lib/mysql-files/ccp_part.csv
```

10 Referencias (formato IEEE)

- [1] Oracle and/or its affiliates, “MySQL 8.0 Reference Manual,” MySQL Documentation. [En línea]. Disponible en: <https://dev.mysql.com/doc/refman/8.0/en/> (Accedido: 17-nov-2025).
- [2] Apache Software Foundation, “Apache Hadoop 3.4 – Documentation,” Apache Hadoop. [En línea]. Disponible en: <https://hadoop.apache.org/docs/r3.4.0/> (Accedido: 17-nov-2025).
- [3] Eclipse Adoptium, “Temurin 17 (OpenJDK 17) Documentation,” Adoptium. [En línea]. Disponible en: <https://adoptium.net/temurin/releases/?version=17> (Accedido: 17-nov-2025).
- [4] Canonical Ltd., “Ubuntu 22.04 LTS Documentation,” Ubuntu. [En línea]. Disponible en: <https://help.ubuntu.com/> (Accedido: 17-nov-2025).
- [5] Microsoft, “Create a Linux virtual machine in the Azure portal,” Microsoft Learn. [En línea]. Disponible en: <https://learn.microsoft.com/azure/virtual-machines/linux/quick-create-portal> (Accedido: 17-nov-2025).
- [6] Apache Software Foundation, “MapReduce Tutorial,” Apache Hadoop. [En línea]. Disponible en: <https://hadoop.apache.org/docs/r3.4.0/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html> (Accedido: 17-nov-2025).
- [7] Oracle and/or its affiliates, “MySQL Server System Variables – secure_file_priv,” MySQL Documentation. [En línea]. Disponible en: https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_secure_file_priv (Accedido: 17-nov-2025).
- [8] The GNU Project, “sed, a stream editor,” GNU sed Manual. [En línea]. Disponible en: <https://www.gnu.org/software/sed/manual/> (Accedido: 17-nov-2025).

Siquieres, puedo entregarte estas secciones en un archivo Markdown listo para imprimir o integrarlas al documento completo con portada e índice.

