



**UNIVERSIDAD TECNOLÓGICA DE PUEBLA DIVISIÓN DE
TECNOLOGÍAS DE LA INFORMACIÓN-
INGENIERÍA EN DESARROLLO Y GESTIÓN DE
SOFTWARE**

MATERIA:

**ADMINISTRACIÓN
DE BASE DE
DATOS**

Recuperación - Parcial 1

Reporte BD COVID19 2021-2024

DOCENTE:

JOSÉ FRANCISCO ESPINOSA GARITA

ALUMNO:

ANDRADE REYES JUSTINO JUAN CARLOS - UTP0002150

8 ° A

FECHA: 20/02/2024

INDICE

Contenido	4
Carga de datos con LOAD DATA	4
MySQL Workbench	5
Python	6
Modelo Relacional de la base de datos	16
Conclusiones	19
Referencias	20

Introducción

El presente reporte documenta el proceso de diseño, creación y configuración de una base de datos destinada a centralizar y gestionar información relacionada con los casos de COVID-19 registrados en México durante el periodo comprendido entre los años 2020 y 2024. Este proyecto surge como una respuesta a la necesidad de contar con un repositorio estructurado y accesible que permita almacenar, organizar y analizar datos históricos sobre la pandemia, con el objetivo aplicar lo aprendido en la materia de administración de bases de datos.

La base de datos fue implementada utilizando **MySQL 9.1**, un sistema de gestión de bases de datos robusto y ampliamente utilizado, y se desarrolló y gestionó a través de **MySQL Workbench**, una herramienta gráfica que facilita el diseño, la administración y la visualización de esquemas y relaciones. Durante el desarrollo, se adoptaron principios de diseño relacional para garantizar un esquema eficiente, normalizado y escalable, que soportara el crecimiento de los datos en términos de volumen y complejidad.

Un aspecto destacado de este proyecto fue el tratamiento de los registros existentes. Para evitar redundancias y garantizar la integridad de los datos, se implementaron procesos que excluyeron los registros previamente almacenados, asegurando que únicamente se incluyeran datos nuevos o actualizados. Esta estrategia optimizó la calidad de la información y redujo posibles inconsistencias de los datos.

La base de datos incluye múltiples dimensiones de información, tales como datos demográficos de los pacientes, fechas relevantes como la de ingreso y síntomas, resultados de pruebas diagnósticas, comorbilidades y clasificaciones finales de casos. Estas características permiten que la base de datos no solo sea un almacén de información, sino también una herramienta poderosa para consultas avanzadas, generación de reportes dinámicos y análisis exploratorio de datos.

Este reporte detalla las decisiones técnicas, los pasos seguidos en la creación del esquema y los procesos de carga de datos, así como los retos enfrentados y las soluciones implementadas. El resultado es una base de datos confiable y funcional que sirve como un recurso clave para estudiar la evolución del COVID-19 y apoyar la toma de decisiones basadas en datos sólidos y actualizados.

Contenido

Carga de datos con LOAD DATA

El comando LOAD DATA en SQL se utiliza para cargar datos de un archivo externo directamente en una tabla de una base de datos. Es una herramienta eficiente y rápida para insertar grandes volúmenes de datos, evitando el uso de múltiples instrucciones INSERT.

Sintaxis básica:

```
LOAD DATA INFILE 'ruta_del_archivo'  
INTO TABLE nombre_de_la_tabla  
FIELDS TERMINATED BY 'delimitador'  
ENCLOSED BY 'carácter'  
LINES TERMINATED BY 'delimitador_de_línea'  
IGNORE n LINES  
(columnas)  
SET column = expresión;
```

Componentes principales:

1. INFILE:

- Especifica la ruta del archivo que contiene los datos a cargar. Puede ser un archivo local o en el servidor.
- Usar LOCAL (LOAD DATA LOCAL INFILE) permite cargar archivos desde el cliente en lugar del servidor.

2. INTO TABLE:

- Indica la tabla de destino donde se insertarán los datos.

3. FIELDS TERMINATED BY:

- Define el delimitador que separa los campos en cada registro, como, para archivos CSV o \t para tabulaciones.

4. ENCLOSED BY:

- Especifica un carácter opcional que encierra los datos, como comillas dobles (") o simples (').

5. LINES TERMINATED BY:

- Define cómo terminan las líneas en el archivo, como \n para nueva línea o \r\n para retorno de carro más nueva línea (usado en sistemas Windows).

6. IGNORE n LINES:

- Permite ignorar las primeras n líneas del archivo, útil si el archivo contiene encabezados.

7. Lista de columnas:

- Define las columnas específicas en las que se insertarán los valores. Esto permite cargar datos solo en columnas seleccionadas.

8. SET:

- Permite realizar transformaciones o asignaciones personalizadas a las columnas durante la carga. Por ejemplo, convertir cadenas vacías en valores NULL.

Beneficios del comando LOAD DATA:

- Rapidez: Es mucho más rápido que insertar registros de manera individual.
- Flexibilidad: Permite ajustar cómo se interpretan y transforman los datos durante la carga.
- Eficiencia: Ideal para migraciones de datos o integraciones masivas.

Precauciones:

1. Permisos: Asegúrate de que el usuario de MySQL tenga permisos para leer archivos en la ubicación especificada.
2. Formato del archivo: El archivo debe coincidir con la configuración especificada (delimitadores, formato de líneas, etc.).
3. Integridad de datos: Si los datos del archivo no cumplen con las restricciones de la tabla (tipos de datos, claves únicas, etc.), podrían generarse errores o truncamientos.

Con LOAD DATA, puedes optimizar significativamente el manejo de grandes conjuntos de datos en tus bases de datos MySQL.

EJEMPLO UTILIZADO:

MySQL Workbench

```
LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 9.1\\Uploads\\COVID192023.csv'
INTO TABLE registros_aguascalientes
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES
```

```
(FECHA_ACTUALIZACION, ID_REGISTRO, ORIGEN, SECTOR, ENTIDAD_UM, SEXO,
ENTIDAD_NAC, ENTIDAD_RES, MUNICIPIO_RES,
```

TIPO_PACIENTE, FECHA_INGRESO, FECHA_SINTOMAS, @FECHA_DEF, INTUBADO,
NEUMONIA, EDAD, NACIONALIDAD, EMBARAZO,

HABLA_LENGUA_INDIG, INDIGENA, DIABETES, EPOC, ASMA, INMUSUPR, HIPERTENSION,
OTRA_COM, CARDIOVASCULAR, OBESIDAD,

RENAL_CRONICA, TABAQUISMO, OTRO_CASO, TOMA_MUESTRA_LAB, RESULTADO_LAB,
TOMA_MUESTRA_ANTIGENO, RESULTADO_ANTIGENO,

CLASIFICACION_FINAL, MIGRANTE, PAIS_NACIONALIDAD, PAIS_ORIGEN, @UCI)

SET

FECHA_DEF = NULLIF(@FECHA_DEF, '0000-00-00'),

UCI = NULLIF(@UCI, ''); -- Maneja valores vacíos como NULL

Python

También con Python se pudo cargar cada archivo inclusive validando los datos para ingresar información integra, el programa ejecutado es el siguiente:

```
import pymysql
import csv
from datetime import datetime

# Configuración de conexión a MySQL
conn = pymysql.connect(
    host='127.0.0.1', # Usamos la dirección local ya que el servidor está en tu máquina
    port=3306,
    user='root',
    passwd='jjc1007M',
    db='covid19'
)

# Ruta en tu máquina local donde están los archivos
path = r"C:\Users\JusAR\Downloads\" # Cambié la ruta a la que mencionaste

# Lista de archivos a procesar (modifica los nombres de los archivos según sea necesario)
archivos = ['COVID192020.csv', 'COVID192021.csv', 'COVID192022.csv', 'COVID192023.csv'] # Cambia por los
nombres de tus archivos
#archivos = ['COVID192024.csv']

# Crear una lista para el encabezado de columnas 2020-2023
csv_header = [
    'FECHA_ACTUALIZACION', 'ID_REGISTRO', 'ORIGEN', 'SECTOR', 'ENTIDAD_UM', 'SEXO',
    'ENTIDAD_NAC', 'ENTIDAD_RES', 'MUNICIPIO_RES', 'TIPO_PACIENTE', 'FECHA_INGRESO',
    'FECHA_SINTOMAS', 'FECHA_DEF', 'INTUBADO', 'NEUMONIA', 'EDAD', 'NACIONALIDAD',
    'EMBARAZO', 'HABLA_LENGUA_INDIG', 'INDIGENA', 'DIABETES', 'EPOC', 'ASMA', 'INMUSUPR',
    'HIPERTENSION', 'OTRA_COM', 'CARDIOVASCULAR', 'OBESIDAD', 'RENAL_CRONICA', 'TABAQUISMO',
    'OTRO_CASO', 'TOMA_MUESTRA_LAB', 'RESULTADO_LAB', 'TOMA_MUESTRA_ANTIGENO',
    'RESULTADO_ANTIGENO', 'CLASIFICACION_FINAL', 'MIGRANTE', 'PAIS_NACIONALIDAD',
    'PAIS_ORIGEN', 'UCI'
]

# # Crear una lista para el encabezado de columnas 2024
```

```
# csv_header = [
# "FECHA_ACTUALIZACION", "ID_REGISTRO", "ORIGEN", "SECTOR", "ENTIDAD_UM", "SEXO",
# "ENTIDAD_NAC", "ENTIDAD_RES", "MUNICIPIO_RES", "TIPO_PACIENTE",
# "FECHA_INGRESO",
# "FECHA_SINTOMAS", "FECHA_DEF", "INTUBADO", "NEUMONIA", "EDAD", "NACIONALIDAD",
# "EMBARAZO", "HABLA_LENGUA_INDIG", "INDIGENA", "DIABETES", "EPOC", "ASMA",
# "INMUSUPR",
# "HIPERTENSION", "OTRA_COM", "CARDIOVASCULAR", "OBESIDAD", "RENAL_CRONICA",
# "TABAQUISMO",
# "OTRO_CASO", "TOMA_MUESTRA_LAB", "RESULTADO_PCR",
# "RESULTADO_PCR_COINFECCION",
# "TOMA_MUESTRA_ANTIGENO", "RESULTADO_ANTIGENO",
# "CLASIFICACION_FINAL_COVID",
# "CLASIFICACION_FINAL_FLU", "MIGRANTE", "PAIS_NACIONALIDAD", "PAIS_ORIGEN", "UCI"
# ]
```

Diccionario de reemplazos

```
replacements = {
    'M?xico': 'México',
    '9999-99-99': None, # Usar None para representar NULL
    'Canad?': 'Canadá',
    'Espa?na': 'España',
    'Estados Unidos de Am?rica': 'Estados Unidos de América',
    'Hait?': 'Haití',
    'Ir?n': 'Irán',
    'Jap?n': 'Japón',
    'Per?': 'Perú',
    'Rep?blica de Costa Rica': 'República de Costa Rica',
    'Sud?frica': 'Sudáfrica'
}
```

SQL para insertar registros

```
sql = f"""
INSERT INTO registros_puebla (
    {' '.join(csv_header)}
) VALUES ({' '.join(['%s'] * len(csv_header))})
"""
```

Procesar y filtrar los archivos

for archivo in archivos:

```
    with open(path + archivo, 'r', newline="", encoding='utf-8') as file:
        reader = csv.reader(file)
        next(reader) # Saltar el encabezado
        row_number = 0
```

with conn.cursor() as cursor:

```
    for row in reader:
        # Reemplazar valores en la fila
        row = [replacements.get(value, value) for value in row]
```

Convertir la fecha '9999-99-99' en None para tratarlo como NULL

```
    for i in range(len(row)):
        if isinstance(row[i], str) and row[i] == '9999-99-99':
            row[i] = None
```

Aplicar filtro

#01 - aguascalientes

#29 - Tlaxcala

if len(row) > 7 and (row[4] == '21' or row[6] == '21' or row[7] == '21'):

try:

```

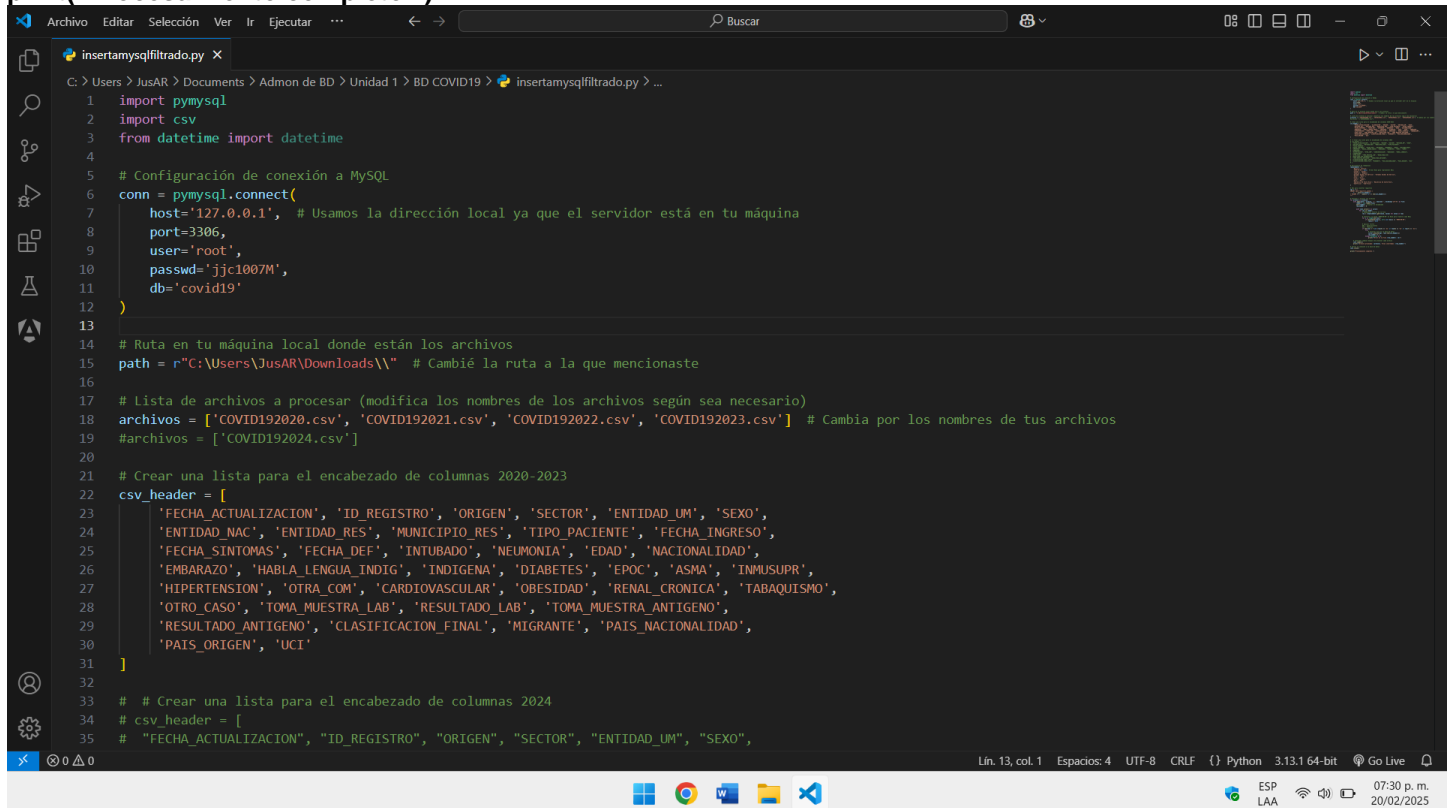
        # Insertar fila en la base de datos
        cursor.execute(sql, row[:len(csv_header)])
        row_number += 1
    except Exception as e:
        print(f"Error en la fila {row_number}: {e}")

# Confirmar cambios después de procesar cada archivo
conn.commit()
print(f"Archivo procesado: {archivo}, Filas insertadas: {row_number}")

# Cerrar la conexión a la base de datos
conn.close()

```

```
print("Procesamiento completo.")
```



```

insertamysqlfiltrado.py
C:\Users\JusAR\Documents\Admon de BD\Unidad 1\BD COVID19\insertamysqlfiltrado.py
1  import pymysql
2  import csv
3  from datetime import datetime
4
5  # Configuración de conexión a MySQL
6  conn = pymysql.connect(
7      host='127.0.0.1', # Usamos la dirección local ya que el servidor está en tu máquina
8      port=3306,
9      user='root',
10     passwd='jjc1007M',
11     db='covid19'
12 )
13
14 # Ruta en tu máquina local donde están los archivos
15 path = r"C:\Users\JusAR\Downloads\\" # Cambié la ruta a la que mencionaste
16
17 # Lista de archivos a procesar (modifica los nombres de los archivos según sea necesario)
18 archivos = ['COVID192020.csv', 'COVID192021.csv', 'COVID192022.csv', 'COVID192023.csv'] # Cambia por los nombres de tus archivos
19 #archivos = ['COVID192024.csv']
20
21 # Crear una lista para el encabezado de columnas 2020-2023
22 csv_header = [
23     'FECHA_ACTUALIZACION', 'ID_REGISTRO', 'ORIGEN', 'SECTOR', 'ENTIDAD_UM', 'SEXO',
24     'ENTIDAD_NAC', 'ENTIDAD_RES', 'MUNICIPIO_RES', 'TIPO_PACIENTE', 'FECHA_INGRESO',
25     'FECHA_SINTOMAS', 'FECHA_DEF', 'INTUBADO', 'NEUMONIA', 'EDAD', 'NACIONALIDAD',
26     'EMBARAZO', 'HABLA_LINGUA_INDIG', 'INDIGENA', 'DIABETES', 'EPOC', 'ASMA', 'INMUSUPR',
27     'HIPERTENSION', 'OTRA_COM', 'CARDIOVASCULAR', 'OBESIDAD', 'RENAL_CRONICA', 'TABAQUISMO',
28     'OTRO_CASO', 'TOMA_MUESTRA_LAB', 'RESULTADO_LAB', 'TOMA_MUESTRA_ANTIGENO',
29     'RESULTADO_ANTIGENO', 'CLASIFICACION_FINAL', 'MIGRANTE', 'PAIS_NACIONALIDAD',
30     'PAIS_ORIGEN', 'UCI'
31 ]
32
33 # # Crear una lista para el encabezado de columnas 2024
34 # csv_header = [
35 #     "FECHA_ACTUALIZACION", "ID_REGISTRO", "ORIGEN", "SECTOR", "ENTIDAD_UM", "SEXO",

```

Load file para registros 2024

```

LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 9.1\Uploads\COVID192024.csv'
INTO TABLE registros_aguascalientes
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES

```

```

(FECHA_ACTUALIZACION, ID_REGISTRO, ORIGEN, SECTOR, ENTIDAD_UM, SEXO, ENTIDAD_NAC,
ENTIDAD_RES, MUNICIPIO_RES,

```


TIPO_PACIENTE, FECHA_INGRESO, FECHA_SINTOMAS, @FECHA_DEF, INTUBADO, NEUMONIA, EDAD, NACIONALIDAD, EMBARAZO,

HABLA_LENGUA_INDIG, INDIGENA, DIABETES, EPOC, ASMA, INMUSUPR, HIPERTENSION, OTRA_COM, CARDIOVASCULAR, OBESIDAD,

RENAL_CRONICA, TABAQUISMO, OTRO_CASO, TOMA_MUESTRA_LAB, RESULTADO_PCR, RESULTADO_PCR_COINFECCION, TOMA_MUESTRA_ANTIGENO, RESULTADO_ANTIGENO,

CLASIFICACION_FINAL_COVID, CLASIFICACION_FINAL_FLU, MIGRANTE, PAIS_NACIONALIDAD, PAIS_ORIGEN, @UCI)

SET

FECHA_DEF = NULLIF(@FECHA_DEF, '0000-00-00'),

UCI = NULLIF(@UCI, ''); -- Maneja valores vacíos como NULL

SELECT * FROM registros_aguascalientes WHERE id_registro = 'ga7fcdd';

SELECT * FROM registros_aguascalientes WHERE resultado_pcr_coinfeccion IS NOT NULL;

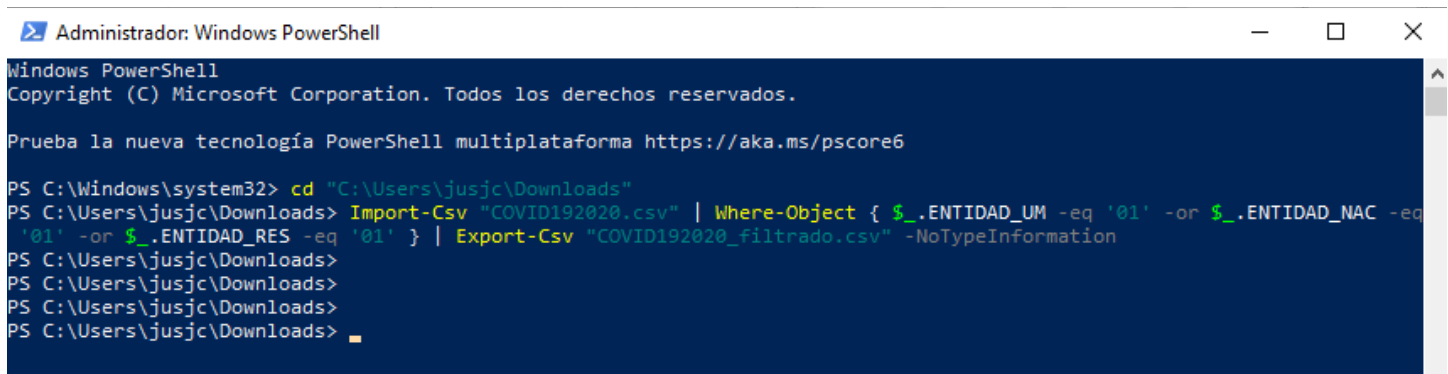
Pre tratado de los registros (filtrado y validación de datos)

Se descargó la bd en registros con formato de texto plano csv. Dicha información era demasiado pesada para poder ser trabajada y cargada, por lo que se tomó la decisión de filtrar cada archivo de registros por año para obtener solo lo deseado ya que se trabajó solo con el estado 1 – Aguascalientes.

Para dicho proceso, se utilizó el comando siguiente para filtrar la información y crear archivos más ligeros:

```
cd "C:\Users\jusjc\Downloads"
```

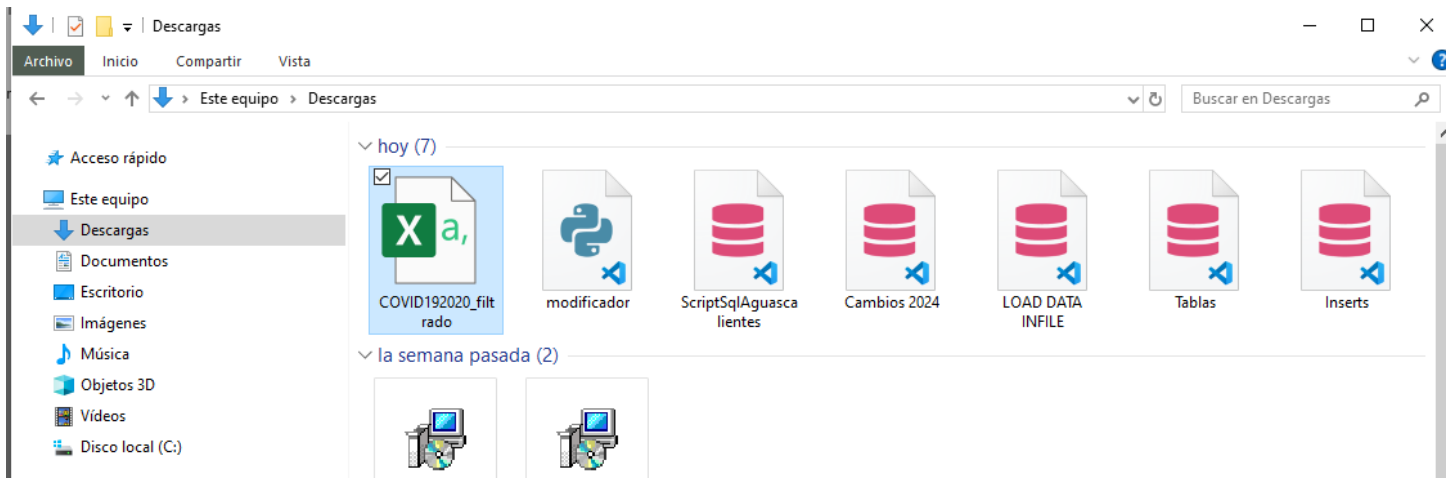
```
Import-Csv "COVID192020.csv" | Where-Object { $_.ENTIDAD_UM -eq '01' -or $_.ENTIDAD_NAC -eq '01' -or $_.ENTIDAD_RES -eq '01' } | Export-Csv "COVID192020_filtrado.csv" -NoTypeInfo
```



```
Administrador: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Windows\system32> cd "C:\Users\jusjc\Downloads"
PS C:\Users\jusjc\Downloads> Import-Csv "COVID192020.csv" | Where-Object { $_.ENTIDAD_UM -eq '01' -or $_.ENTIDAD_NAC -eq '01' -or $_.ENTIDAD_RES -eq '01' } | Export-Csv "COVID192020_filtrado.csv" -NoTypeInfo
PS C:\Users\jusjc\Downloads>
PS C:\Users\jusjc\Downloads>
PS C:\Users\jusjc\Downloads>
PS C:\Users\jusjc\Downloads> .
```



Creación de tablas

-- Tabla origen

```
CREATE TABLE origen (
    clave INT PRIMARY KEY,
    descripcion VARCHAR(50) NOT NULL
);
```

-- Tabla sector

```
CREATE TABLE sector (
    clave INT PRIMARY KEY,
    descripcion VARCHAR(50) NOT NULL
);
```

-- Tabla sexo

```
CREATE TABLE sexo (
    clave INT PRIMARY KEY,
    descripcion VARCHAR(50) NOT NULL
);
```

-- Tabla tipo_paciente

```
CREATE TABLE tipo_paciente (
    clave INT PRIMARY KEY,
```

```
descripcion VARCHAR(50) NOT NULL
```

```
);
```

-- Tabla si_no

```
CREATE TABLE si_no (
    clave INT PRIMARY KEY,
    descripcion VARCHAR(50) NOT NULL
);
```

-- Tabla nacionalidad

```
CREATE TABLE nacionalidad (
    clave INT PRIMARY KEY,
    descripcion VARCHAR(50) NOT NULL
);
```

-- Tabla resultado_pcr

```
CREATE TABLE resultado_pcr (
    clave INT PRIMARY KEY,
    descripcion VARCHAR(255) NOT NULL
);
```

```

-- Tabla resultado_antigeno
CREATE TABLE resultado_antigeno (
    clave INT PRIMARY KEY,
    descripcion VARCHAR(255) NOT NULL
);

-- Tabla clasificacion_final_covid
CREATE TABLE clasificacion_final_covid (
    clave INT PRIMARY KEY,
    clasificacion VARCHAR(255) NOT NULL,
    descripcion TEXT NOT NULL
);

-- Tabla clasificacion_final_flu
CREATE TABLE clasificacion_final_flu (
    clave INT PRIMARY KEY,
    clasificacion VARCHAR(255) NOT NULL,
    descripcion TEXT NOT NULL
);

-- Tabla entidades
CREATE TABLE entidades (
    clave_entidad CHAR(2) PRIMARY KEY,
    entidad_federativa VARCHAR(255) NOT NULL,
    abreviatura VARCHAR(4) NOT NULL
);

-- Tabla municipios
CREATE TABLE municipios (
    clave_entidad CHAR(2) NOT NULL,
    clave_municipio CHAR(3) NOT NULL,
    municipio VARCHAR(255) NOT NULL,
    PRIMARY KEY (clave_municipio),
    FOREIGN KEY (clave_entidad) REFERENCES entidades(clave_entidad)
);

-- Tabla aguascalientes con claves foráneas
CREATE TABLE registros_aguascalientes (
    fecha_actualizacion DATE,
    id_registro VARCHAR(50),
    origen INT,
    sector INT,
    entidad_um CHAR(2),
    sexo INT,
    entidad_nac CHAR(2),
    entidad_res CHAR(2),
    municipio_res CHAR(3),
    tipo_paciente INT,
    fecha_ingreso DATE,
    fecha_sintomas DATE,
    fecha_def DATE DEFAULT NULL,
    intubado INT,
    neumonia INT,
    edad INT,
    nacionalidad INT,
    embarazo INT,
    habla_lengua_indig INT,
    indigena INT,
    diabetes INT,
    epoc INT,
    asma INT,
    inmusupr INT,
    hipertension INT,

```

```

otra_com INT,
cardiovascular INT,
obesidad INT,
renal_cronica INT,
tabaquismo INT,
otro_caso INT,
toma_muestra_lab INT,
resultado_lab INT,
toma_muestra_antigeno INT,
resultado_antigeno INT,
clasificacion_final INT,
migrante INT,
pais_nacionalidad VARCHAR(100),
pais_origen VARCHAR(100),
uci INT,

```

```

FOREIGN KEY (origen) REFERENCES
origen(clave),

FOREIGN KEY (sector) REFERENCES
sector(clave),

FOREIGN KEY (sexo) REFERENCES
sexo(clave),

FOREIGN KEY (entidad_um) REFERENCES
entidades(clave_entidad),

FOREIGN KEY (entidad_nac) REFERENCES
entidades(clave_entidad),

FOREIGN KEY (entidad_res) REFERENCES
entidades(clave_entidad),

FOREIGN KEY (tipo_paciente) REFERENCES
tipo_paciente(clave),

FOREIGN KEY (nacionalidad) REFERENCES
nacionalidad(clave),

FOREIGN KEY (clasificacion_final)
REFERENCES clasificacion_final_covid(clave)
);

```

Inserción de datos

Catálogo

-- Insertar datos en la tabla origen

```

INSERT INTO origen (clave, descripcion) VALUES
(1, 'USMER'),
(2, 'FUERA DE USMER'),
(99, 'NO ESPECIFICADO');

```

-- Insertar datos en la tabla sector

```

INSERT INTO sector (clave, descripcion) VALUES
(1, 'CRUZ ROJA'),
(2, 'DIF'),
(3, 'ESTATAL'),
(4, 'IMSS'),
(5, 'IMSS-BIENESTAR'),
(6, 'ISSSTE'),
(7, 'MUNICIPAL'),
(8, 'PEMEX'),
(9, 'PRIVADA'),

```

```

(10, 'SEDENA'),
(11, 'SEMAR'),
(12, 'SSA'),
(13, 'UNIVERSITARIO'),
(14, 'CIJ'),
(15, 'IMSS Bienestar OPD'),
(99, 'NO ESPECIFICADO');

```

-- Insertar datos en la tabla sexo

```

INSERT INTO sexo (clave, descripcion) VALUES
(1, 'MUJER'),
(2, 'HOMBRE'),
(99, 'NO ESPECIFICADO');

```

-- Insertar datos en la tabla tipo_paciente

```

INSERT INTO tipo_paciente (clave, descripcion) VALUES
(1, 'AMBULATORIO'),

```

```

(2, 'HOSPITALIZADO'),
(99, 'NO ESPECIFICADO');

-- Insertar datos en la tabla si_no
INSERT INTO si_no (clave, descripcion) VALUES
(1, 'SI'),
(2, 'NO'),
(97, 'NO APLICA'),
(98, 'SE IGNORA'),
(99, 'NO ESPECIFICADO');

-- Insertar datos en la tabla nacionalidad
INSERT INTO nacionalidad (clave, descripcion) VALUES
(1, 'MEXICANA'),
(2, 'EXTRANJERA'),
(99, 'NO ESPECIFICADO');

-- Insertar datos en la tabla resultado_pcr
INSERT INTO resultado_pcr (clave, descripcion) VALUES
(1, 'INFLUENZA AH1N1 PMD'),
(2, 'INFLUENZA A H1'),
(3, 'INFLUENZA A H3'),
(4, 'INFLUENZA B'),
(5, 'NEGATIVO'),
(6, 'MUESTRA NO ADECUADA'),
(7, 'ADENOVIRUS'),
(8, 'PARAINFLUENZA 1'),
(9, 'PARAINFLUENZA 2'),
(10, 'PARAINFLUENZA 3'),
(11, 'VIRUS SINCICIAL RESPIRATORIO'),
(13, 'INFLUENZA A NO SUBTIPIFICADA'),
(14, 'INFLUENZA A H5'),
(17, 'MUESTRA RECHAZADA'),
(20, 'VIRUS SINCICIAL RESPIRATORIO A'),
(21, 'VIRUS SINCICIAL RESPIRATORIO B'),
(22, 'CORONA 229E'),
(23, 'CORONA OC43'),
(24, 'CORONA SARS'),
(25, 'CORONA NL63'),

```

```

(26, 'CORONA HKU1'),
(27, 'MUESTRA QUE NO AMPLIFICO'),
(28, 'ENTEROVIRUS/RHINOVIRUS'),
(29, 'METAPNEUMOVIRUS'),
(30, 'MUESTRA SIN AISLAMIENTO'),
(32, 'PARAINFLUENZA 4'),
(33, 'MUESTRA SIN CELULAS'),
(34, 'SARS-CoV-2'),
(35, 'MERS-CoV'),
(36, 'SARS-CoV'),
(37, 'BOCAVIRUS'),
(41, 'MUESTRA NO RECIBIDA'),
(997, 'NO APLICA (CASO SIN MUESTRA)'),
(998, 'SIN COINFECCIÓN'),
(999, 'PENDIENTE');

-- Insertar datos en la tabla resultado_antigeno
INSERT INTO resultado_antigeno (clave, descripcion) VALUES
(1, 'INFLUENZA AH1N1 PMD'),
(2, 'INFLUENZA A H1'),
(3, 'INFLUENZA A H3'),
(4, 'INFLUENZA B'),
(5, 'NEGATIVO'),
(6, 'MUESTRA NO ADECUADA'),
(7, 'ADENOVIRUS'),
(8, 'PARAINFLUENZA 1'),
(9, 'PARAINFLUENZA 2'),
(10, 'PARAINFLUENZA 3'),
(11, 'VIRUS SINCICIAL RESPIRATORIO'),
(13, 'INFLUENZA A NO SUBTIPIFICADA'),
(14, 'INFLUENZA A H5'),
(17, 'MUESTRA RECHAZADA'),
(20, 'VIRUS SINCICIAL RESPIRATORIO A'),
(21, 'VIRUS SINCICIAL RESPIRATORIO B'),
(22, 'CORONA 229E'),
(23, 'CORONA OC43'),
(24, 'CORONA SARS'),
(25, 'CORONA NL63'),
(26, 'CORONA HKU1'),

```

(27, 'MUESTRA QUE NO AMPLIFICO'),
 (28, 'ENTEROV//RHINOVIRUS'),
 (29, 'METAPNEUMOVIRUS'),
 (30, 'MUESTRA SIN AISLAMIENTO'),
 (32, 'PARAINFLUENZA 4'),
 (33, 'MUESTRA SIN CELULAS'),
 (34, 'SARS-CoV-2'),
 (35, 'MERS-CoV'),
 (36, 'SARS-CoV'),
 (37, 'BOCAVIRUS'),
 (41, 'MUESTRA NO RECIBIDA'),
 (997, 'NO APLICA (CASO SIN MUESTRA)'),
 (998, 'SIN COINFECCIÓN'),
 (999, 'PENDIENTE');

-- Insertar datos en la tabla clasificacion_final_covid

INSERT INTO clasificacion_final_covid (clave, clasificacion, descripcion) VALUES

(1, 'CASO DE COVID-19 CONFIRMADO POR ASOCIACIÓN CLÍNICA EPIDEMIOLÓGICA', 'Confirmado por asociación aplica cuando el caso informó ser contacto de un positivo a COVID-19 (y este se encuentra registrado en el SISVER) y: Al caso no se le tomo muestra ni de PCR ni de antígeno o la muestra resultó no válida.'),

(2, 'CASO DE COVID-19 CONFIRMADO POR COMITÉ DE DICTAMINACIÓN', 'Confirmado por dictaminación solo aplica para defunciones bajo las siguientes condiciones: Al caso no se le tomo muestra ni de PCR ni de antígeno o si se tomó muestra, pero la muestra resultó no válida.'),

(3, 'CASO DE SARS-COV-2 CONFIRMADO', 'Confirmado aplica cuando: El caso tiene muestra de laboratorio PCR o prueba antigénica y resultó positiva a SARS-CoV-2, sin importar si el caso tienen asociación clínica epidemiológica.'),

(4, 'INVÁLIDO POR LABORATORIO', 'Inválido aplica cuando el caso no tiene asociación clínico epidemiológica, ni dictaminación a COVID-19. Se le tomó muestra de laboratorio PCR y esta resultó no válida.'),

(5, 'NO REALIZADO POR LABORATORIO', 'No realizado aplica cuando el caso no tiene asociación clínico epidemiológica, ni dictaminación a COVID-19 y se le tomó muestra de laboratorio PCR y esta no se procesó.'),

(6, 'CASO SOSPECHOSO', 'Sospechoso aplica cuando: El caso no tiene asociación clínico epidemiológica, ni dictaminación a COVID-19 y no se le tomó muestra, o se le tomó muestra de laboratorio y está pendiente de resultado, sin importar otra condición.'),

(7, 'NEGATIVO A SARS-COV-2', 'Negativo aplica cuando el caso: 1. Se le tomo muestra de laboratorio PCR y ésta resultó: negativa a SARS-CoV-2 o positiva a cualquier otro virus respiratorio (Influenza, VSR, Bocavirus, otros) sin importar que este caso tenga asociación clínico epidemiológica o dictaminación a COVID-19. 2. Se le tomó muestra antigénica que resultó negativa a SARS-CoV-2 y al caso no se le tomó muestra de laboratorio PCR ni se le confirmo por

asociación epidemiológica o por dictaminación clínica epidemiológica.');

-- Insertar datos en la tabla clasificacion_final_flu

INSERT INTO clasificacion_final_flu (clave, clasificacion, descripcion) VALUES

(3, 'CASO DE INFLUENZA CONFIRMADO', 'Confirmado aplica cuando el caso tiene muestra de laboratorio PCR y resultó positiva a Influenza.'),

(4, 'INVÁLIDO POR LABORATORIO', 'Inválido aplica cuando al caso se le tomó muestra de laboratorio PCR y esta resultó no válida.'),

(5, 'NO REALIZADO POR LABORATORIO', 'No realizado aplica cuando al caso se le tomó muestra de laboratorio PCR y esta no se procesó.'),

(6, 'CASO SOSPECHOSO', 'Sospechoso aplica cuando el caso no se le tomó muestra PCR, o se le tomó muestra de laboratorio PCR y está pendiente de resultado.'),

(7, 'NEGATIVO A INFLUENZA', 'Negativo aplica cuando al caso se le tomo muestra de laboratorio PCR y ésta resultó negativa a influenza o positiva a cualquier otro virus respiratorio.');

-- Inserción con clave_entidad como cadena

INSERT INTO entidades (clave_entidad, entidad_federativa, abreviatura) VALUES

('01', 'AGUASCALIENTES', 'AS'),

('02', 'BAJA CALIFORNIA', 'BC'),

('03', 'BAJA CALIFORNIA SUR', 'BS'),

('04', 'CAMPECHE', 'CC'),

('05', 'COAHUILA DE ZARAGOZA', 'CL'),

('06', 'COLIMA', 'CM'),

('07', 'CHIAPAS', 'CS'),

('08', 'CHIHUAHUA', 'CH'),

('09', 'CIUDAD DE MÉXICO', 'DF'),

('10', 'DURANGO', 'DG'),

('11', 'GUANAJUATO', 'GT'),

('12', 'GUERRERO', 'GR'),

('13', 'HIDALGO', 'HG'),

('14', 'JALISCO', 'JC'),

('15', 'MÉXICO', 'MC'),

('16', 'MICHOACÁN DE OCAMPO', 'MN'),

('17', 'MORELOS', 'MS'),

('18', 'NAYARIT', 'NT'),

('19', 'NUEVO LEÓN', 'NL'),

('20', 'OAXACA', 'OC'),

('21', 'PUEBLA', 'PL'),	
('22', 'QUERÉTARO', 'QT'),	-- Inserción de municipios con clave_entidad como cadena
('23', 'QUINTANA ROO', 'QR'),	INSERT INTO municipios (clave_entidad, clave_municipio, municipio)
('24', 'SAN LUIS POTOSÍ', 'SP'),	VALUES
('25', 'SINALOA', 'SL'),	('01', '001', 'AGUASCALIENTES'),
('26', 'SONORA', 'SR'),	('01', '002', 'ASIENTOS'),
('27', 'TABASCO', 'TC'),	('01', '003', 'CALVILLO'),
('28', 'TAMAULIPAS', 'TS'),	('01', '004', 'COSÍO'),
('29', 'TLAXCALA', 'TL'),	('01', '005', 'JESÚS MARÍA'),
('30', 'VERACRUZ DE IGNACIO DE LA LLAVE', 'VZ'),	('01', '006', 'PABELLÓN DE ARTEAGA'),
('31', 'YUCATÁN', 'YN'),	('01', '007', 'RINCÓN DE ROMOS'),
('32', 'ZACATECAS', 'ZS'),	('01', '008', 'SAN JOSÉ DE GRACIA'),
('36', 'ESTADOS UNIDOS MEXICANOS', 'EUM'),	('01', '009', 'TEPEZALÁ'),
('97', 'NO APLICA', 'NA'),	('01', '010', 'EL LLANO'),
('98', 'SE IGNORA', 'SI'),	('01', '011', 'SAN FRANCISCO DE LOS ROMO'),
('99', 'NO ESPECIFICADO', 'NE');	('01', '999', 'NO ESPECIFICADO');

Cambios y actualizaciones a considerar para 2024

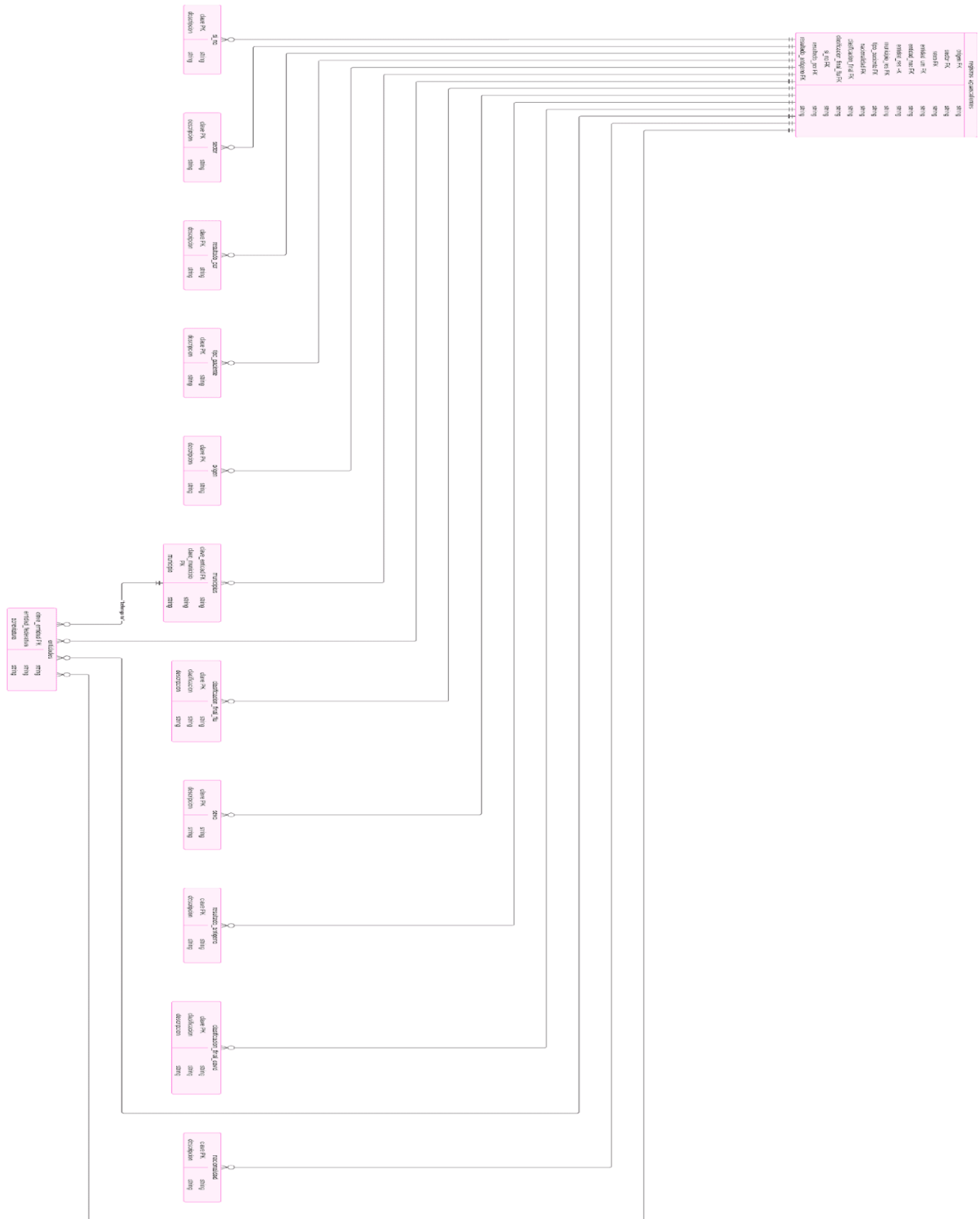
```
ALTER TABLE registros_aguascalientes CHANGE COLUMN RESULTADO_LAB resultado_pcr INT;
```

```
ALTER TABLE registros_aguascalientes CHANGE COLUMN CLASIFICACION_FINAL
clasificacion_final_covid INT;
```

```
ALTER TABLE registros_aguascalientes ADD COLUMN resultado_pcr_coinfeccion INT NULL AFTER
resultado_pcr;
```

```
ALTER TABLE registros_aguascalientes ADD COLUMN clasificacion_final_flu INT NULL AFTER
clasificacion_final_covid;
```

Modelo Relacional de la base de datos



Capturas de pantalla de herramientas utilizadas

```
MySQL 9.1 Command Line Client

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 covid19;
Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_covid19 |
+-----+
| clasificacion_final_covid |
| clasificacion_final_flu |
| entidades |
| municipios |
| nacionalidad |
| origen |
| registros_aguascalientes |
| resultado_antigeno |
| resultado_pcr |
| sector |
| sexo |
| si_no |
| tipo_paciente |
+-----+
13 rows in set (0.00 sec)

mysql>
```

```
MySQL 9.1 Command Line Client

+-----+
| tipo_paciente |
+-----+
13 rows in set (0.00 sec)

mysql> describe registros_aguascalientes;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| fecha_actualizacion | date | YES | | NULL | |
| id_registro | varchar(50) | YES | | NULL | |
| origen | int | YES | MUL | NULL | |
| sector | int | YES | MUL | NULL | |
| entidad_um | char(2) | YES | MUL | NULL | |
| sexo | int | YES | MUL | NULL | |
| entidad_nac | char(2) | YES | MUL | NULL | |
| entidad_res | char(2) | YES | MUL | NULL | |
| municipio_res | char(3) | YES | | NULL | |
| tipo_paciente | int | YES | MUL | NULL | |
| fecha_ingreso | date | YES | | NULL | |
| fecha_sintomas | date | YES | | NULL | |
| fecha_def | date | YES | | NULL | |
| intubado | int | YES | | NULL | |
| neumonia | int | YES | | NULL | |
| edad | int | YES | | NULL | |
| nacionalidad | int | YES | MUL | NULL | |
| embarazo | int | YES | | NULL | |
| habla_lengua_indig | int | YES | | NULL | |
| indigena | int | YES | | NULL | |
| diabetes | int | YES | | NULL | |
| epoc | int | YES | | NULL | |
+-----+
```

MySQL Workbench

Local instance MySQL91 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHEMAS

covid19

Tables

- clasificacion_final_covid
- clasificacion_final_flu
- entidades
- municipios
- nacionalidad
- origen
- registros_aguascalientes
- resultado_antigeno
- resultado_pcr
- sector
- sexo
- si_no
- tipo_paciente

Views

Stored Procedures

Administration Schemas

Information

Table: registros_aguascalientes

Columns:

- fecha_actualizacion date
- id_registro int
- origen int
- sector int
- entidad_um char(2)
- sexo char(2)
- entidad_nac char(2)
- entidad_res char(2)
- municipio_res char(2)

Script: registros_aguascalientes

Limit to 1000 rows

1 SELECT * FROM covid19.registros_aguascalientes;

Result Grid

fecha_actualizacion	id_registro	origen	sector	entidad_um	sexo	entidad_nac	entidad_res	municipio_res	tipo_paciente	fecha_ingreso	fecha_sintomas	fecha_def	intubado
2021-10-31	z1da1e	1	12	01	2	01	01	001	1	2020-03-09	2020-03-05	NULL	97
2021-10-31	z552ac	1	12	01	1	09	01	001	1	2020-06-02	2020-05-30	NULL	97
2021-10-31	z2z6fd	1	12	01	2	14	32	019	1	2020-12-24	2020-12-24	NULL	97
2021-10-31	z166d5	1	12	01	1	01	01	001	2	2020-04-17	2020-04-14	NULL	2
2021-10-31	z51aa0	1	12	01	1	01	01	001	1	2020-10-19	2020-10-11	NULL	97
2021-10-31	z1b0d1	1	12	01	1	01	01	003	1	2020-04-23	2020-04-21	NULL	97

Output

Action Output

#	Time	Action	Message	Duration / Fetch
10	12:01:55	SELECT * FROM registros_aguascalientes WHERE fecha_ingreso LIKE '2024%' LIMIT 1000	1000 row(s) returned	1.172 sec / 0.016 sec
11	12:03:54	SELECT * FROM registros_aguascalientes WHERE id_registro = 'ga7cdd' LIMIT 0, 1000	2 row(s) returned	1.015 sec / 0.000 sec
12	12:07:35	SELECT * FROM registros_aguascalientes WHERE fecha_ingreso LIKE '2024%' LIMIT 1000	1000 row(s) returned	1.171 sec / 0.016 sec
13	12:08:02	SELECT * FROM registros_aguascalientes WHERE fecha_actualizacion LIKE '2024%' LIMIT 1000	1000 row(s) returned	1.063 sec / 0.015 sec
14	12:12:47	SELECT * FROM registros_aguascalientes WHERE resultado_pcr_coinfeccion IS NOT NULL	1000 row(s) returned	0.922 sec / 0.000 sec
15	13:07:39	SELECT * FROM covid19.registros_aguascalientes LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.015 sec

Archivo Editar Selección Ver ...

Buscar

Tablas.sql Inserts.sql LOAD DATA INFILE.sql Cambios 2024.sql

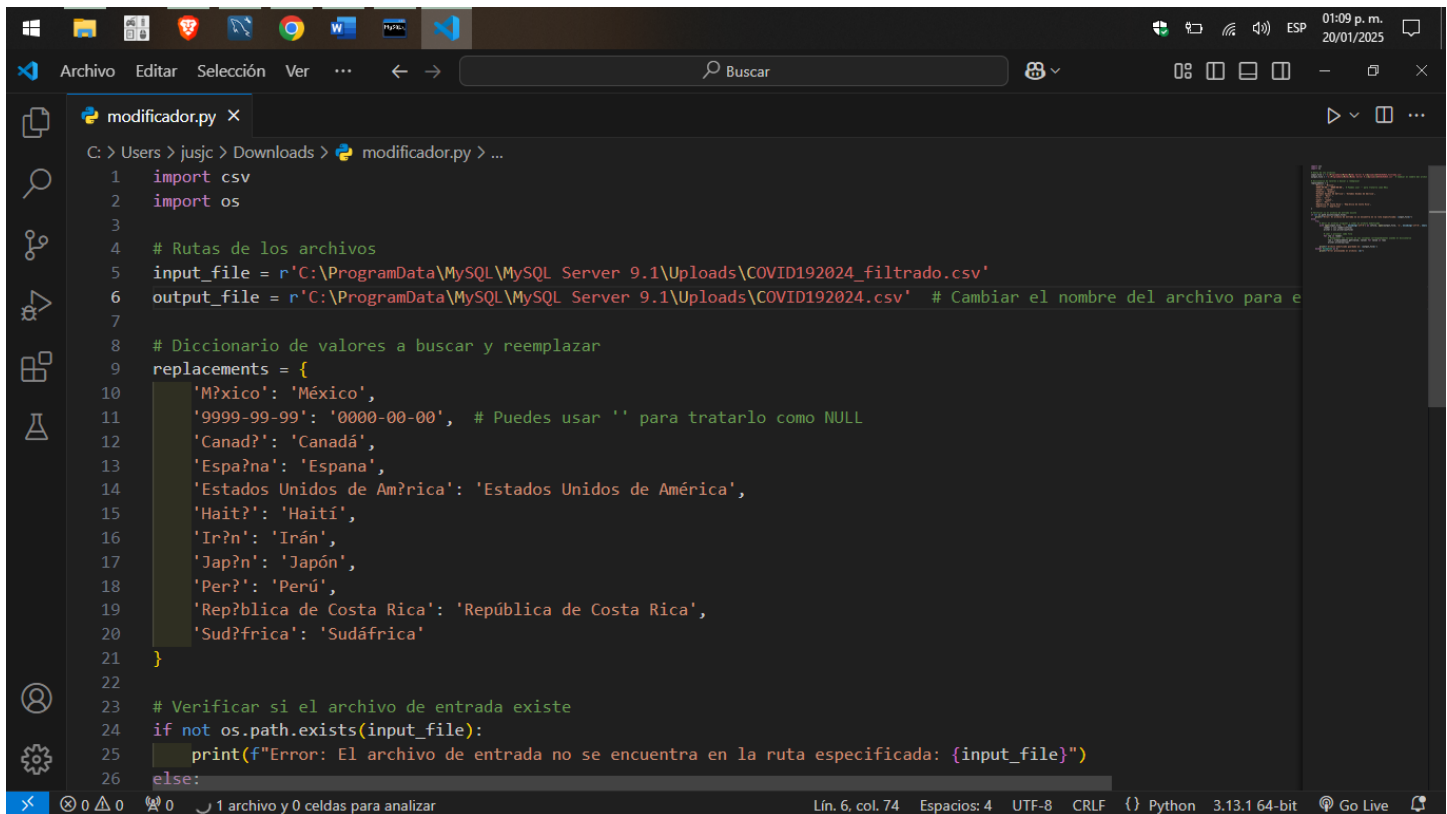
C: > Users > jusjc > Downloads > Cambios 2024.sql

```

1 ALTER TABLE registros_aguascalientes CHANGE COLUMN RESULTADO_LAB resultado_pcr INT;
2
3 ALTER TABLE registros_aguascalientes CHANGE COLUMN CLASIFICACION_FINAL clasificacion_final_covid INT;
4
5 ALTER TABLE registros_aguascalientes ADD COLUMN resultado_pcr_coinfeccion INT NULL AFTER resultado_pcr;
6
7 ALTER TABLE registros_aguascalientes ADD COLUMN clasificacion_final_flu INT NULL AFTER clasificacion_final_covid;
8
9 LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 9.1\Uploads\COVID192024.csv'
10 INTO TABLE registros_aguascalientes
11 FIELDS TERMINATED BY ','
12 ENCLOSED BY '"'
13 LINES TERMINATED BY '\n'
14 IGNORE 1 LINES
15 (FECHA_ACTUALIZACION, ID_REGISTRO, ORIGEN, SECTOR, ENTIDAD_UM, SEXO, ENTIDAD_NAC, ENTIDAD_RES, MUNICIPIO_RES,
16 TIPO_PACIENTE, FECHA_INGRESO, FECHA_SINTOMAS, @FECHA_DEF, INTUBADO, NEUMONIA, EDAD, NACIONALIDAD, EMBARAZO,
17 HABLA_LENGUA_INDIG, INDIGENA, DIABETES, EPOC, ASMA, INMUSUPR, HIPERTENSION, OTRA_COM, CARDIOVASCULAR, OBESIDAD,
18 RENAL_CRONICA, TABAQUISMO, OTRO_CASO, TOMA_MUESTRA_LAB, RESULTADO_PCR, RESULTADO_PCR_COINFECCION, TOMA_MUESTRA_ANTIGE
19 CLASIFICACION_FINAL_COVID, CLASIFICACION_FINAL_FLU, MIGRANTE, PAIS_NACIONALIDAD, PAIS_ORIGEN, @UCI)
20 SET
21 FECHA_DEF = NULLIF(@FECHA_DEF, '0000-00-00'),
22 UCI = NULLIF(@UCI, ''); -- Maneja valores vacios como NULL
23
24 SELECT * FROM registros_aguascalientes WHERE id_registro = 'ga7cdd';
25
26 SELECT * FROM registros_aguascalientes WHERE resultado_pcr_coinfeccion IS NOT NULL;

```

Lín. 12, col. 17 Espacios: 2 UTF-8 CRLF MS SQL Go Live



```
C: > Users > jusjc > Downloads > modificador.py > ...
1 import csv
2 import os
3
4 # Rutas de los archivos
5 input_file = r'C:\ProgramData\MySQL\MySQL Server 9.1\Uploads\COVID192024_filtrado.csv'
6 output_file = r'C:\ProgramData\MySQL\MySQL Server 9.1\Uploads\COVID192024.csv' # Cambiar el nombre del archivo para e
7
8 # Diccionario de valores a buscar y reemplazar
9 replacements = {
10     'M?xico': 'México',
11     '9999-99-99': '0000-00-00', # Puedes usar '' para tratarlo como NULL
12     'Canad?': 'Canadá',
13     'Espa?na': 'España',
14     'Estados Unidos de Am?rica': 'Estados Unidos de América',
15     'Hait?': 'Haití',
16     'Ir?n': 'Irán',
17     'Jap?n': 'Japón',
18     'Per?': 'Perú',
19     'Rep?blica de Costa Rica': 'República de Costa Rica',
20     'Sud?frica': 'Sudáfrica'
21 }
22
23 # Verificar si el archivo de entrada existe
24 if not os.path.exists(input_file):
25     print(f"Error: El archivo de entrada no se encuentra en la ruta especificada: {input_file}")
26 else:
```

Conclusiones

La creación de la base de datos de COVID-19 para el periodo de 2020 a 2024 fue un proceso meticuloso y bien estructurado que permitió consolidar y gestionar la información relacionada con los casos reportados durante esa pandemia. La base de datos fue diseñada utilizando MySQL 9.1 y gestionada a través de MySQL Workbench, lo que permitió realizar una implementación robusta y eficiente.

A lo largo de este proceso, se tomaron en cuenta varios aspectos clave:

Estructuración de la Base de Datos: Se diseñó un esquema adecuado para almacenar información crítica, como detalles de los pacientes, resultados de pruebas, fechas clave, condiciones de salud y otros aspectos epidemiológicos. Esto permitió contar con una estructura de datos organizada y accesible para análisis posteriores.

Importación y Filtrado de Datos: Para cargar los datos históricos de COVID-19, se utilizó el comando LOAD DATA INFILE, lo que facilitó la importación masiva de los registros desde archivos CSV. Además, se implementaron scripts en PowerShell para filtrar y limpiar los datos antes de cargarlos en la base de datos, asegurando que solo los registros relevantes fueran importados.

Optimización del Proceso de Carga de Datos: Se tomaron precauciones para manejar registros con valores nulos y se realizaron modificaciones a las tablas para incorporar nuevos campos, lo que mejoró la integridad de los datos y permitió un análisis más detallado. Estas modificaciones fueron necesarias para adaptar la estructura de la base de datos a las necesidades del análisis, como la inclusión de campos relacionados con resultados de pruebas adicionales y nuevas clasificaciones.

Uso de Herramientas de Gestión: MySQL Workbench se destacó como una herramienta eficiente para la administración de la base de datos, permitiendo la creación de consultas, la visualización de tablas y la gestión de registros de forma intuitiva y fácil de usar. Su interfaz gráfica facilitó la interacción con el servidor MySQL, permitiendo a los usuarios realizar tareas de mantenimiento y gestión de datos sin necesidad de usar comandos complejos de consola.

Integración y Acceso a los Datos: La base de datos es accesible para consultas y análisis detallados, permitiendo a los investigadores y analistas realizar estudios de tendencias, como la distribución de casos por entidad o la evolución de la pandemia en diferentes regiones. Esto constituye un recurso valioso para la toma de decisiones y la implementación de medidas sanitarias.

En conclusión, la implementación de esta base de datos no solo ha permitido almacenar de manera eficiente grandes volúmenes de información sobre los casos de COVID-19, sino que también ha brindado herramientas adecuadas para realizar un seguimiento detallado de la pandemia a lo largo del tiempo. El uso de MySQL y MySQL Workbench ha proporcionado la flexibilidad necesaria para adaptar la base de datos a medida que surgen nuevas necesidades, garantizando que los datos estén siempre actualizados y sean fácilmente accesibles para los análisis.

Referencias

- Gobierno de México. (n.d.). Datos abiertos. Secretaría de Salud. <https://www.gob.mx/salud/documentos/datos-abiertos-152127>
- Gobierno de México. (2020). *Actualizaciones en la presentación de información referente a casos de COVID-19 en México*. Secretaría de Salud. <https://www.gob.mx/salud/documentos/datos-abiertos-152127>
- Goodine, M. [Mike Goodine]. (2019, 18 de abril). *LOAD DATA INFILE with Workbench for newbies* [Mensaje en un foro en línea]. Recuperado de <https://forums.mysql.com/read.php?152,674208,674208>
- Material visto en classroom