

```
%mysql
use DANILO
```

FINISHED

Query executed successfully. Affected rows : 0

Took 0 sec. Last updated by anonymous at July 25 2024, 1:23:52 PM.

Pregunta 1

FINISHED

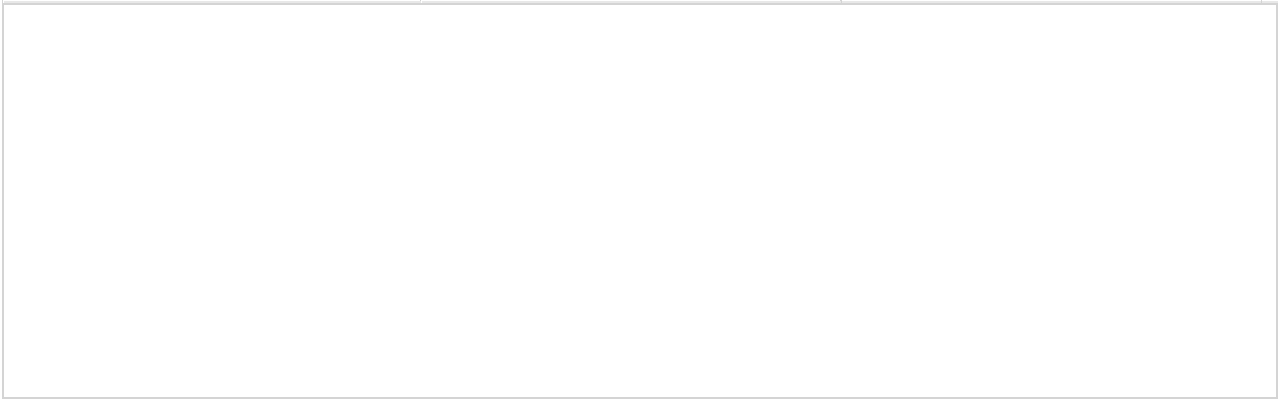
Took 7 sec. Last updated by anonymous at July 25 2024, 11:34:11 AM.

```
%mysql(saveAs = primera)
SELECT
  Z.nombre_zona AS ZonaDetencion,
  SZ.nombre_subzona AS SubzonaDetencion,
  I.presunta_infraccion AS TipoInfraccion,
  COALESCE(A.tipo_arma, 'Sin Arma') AS TipoArma, -- Muestra 'Sin Arma' si no hay tipo de ar
  COUNT(D.detenciones_id) AS TotalDetenciones
FROM
  detenciones D
JOIN
  infracciones I ON D.presunta_subinfraccion = I.presunta_subinfraccion
JOIN
  subcircuitos SC ON D.codigo_subcircuito = SC.codigo_subcircuito
JOIN
  subzonas SZ ON SC.nombre_subcircuito = SZ.nombre_subzona
JOIN
  zonas Z ON SZ.zona = Z.nombre_zona
LEFT JOIN
  armas A ON D.arma = A.arma
WHERE
  D.fecha_detencion BETWEEN '2023-01-01' AND '2024-12-31'
GROUP BY
  Z.nombre_zona, SZ.nombre_subzona, I.presunta_infraccion, A.tipo_arma
ORDER BY
  TotalDetenciones DESC
LIMIT 1000;
```

FINISHED

settings ▼

ZonaDetencion ▼	SubzonaDetencion ▼	TipoInfraccion ▼	
ZONA 5	MAR TERRITORIAL	null	1
ZONA 5	MAR TERRITORIAL	null	1



Took 0 sec. Last updated by anonymous at July 25 2024, 1:31:31 PM.

```
%python
# Cargar el DataFrame de MySQL
primeraDf = z.getAsDataFrame("primera")

# Mostrar el DataFrame
print(primeraDf)
```

FINISHED

```

  ZonaDetencion SubzonaDetencion ...      TipoArma TotalDetenciones
0      ZONA 5    MAR TERRITORIAL ...      NINGUNA              42
1      ZONA 5    MAR TERRITORIAL ...  ARMAS DE FUEGO              3

[2 rows x 5 columns]
```

Took 0 sec. Last updated by anonymous at July 25 2024, 1:31:42 PM.

```
%python
# Importar las librerías necesarias
import matplotlib.pyplot as plt
import pandas as pd

# Cargar el DataFrame de MySQL
primeraDf = z.getAsDataFrame("primera")

# Agrupar los datos por TipoArma y sumar TotalDetenciones
arma_detenciones = primeraDf.groupby('TipoArma')['TotalDetenciones'].sum().reset_index()

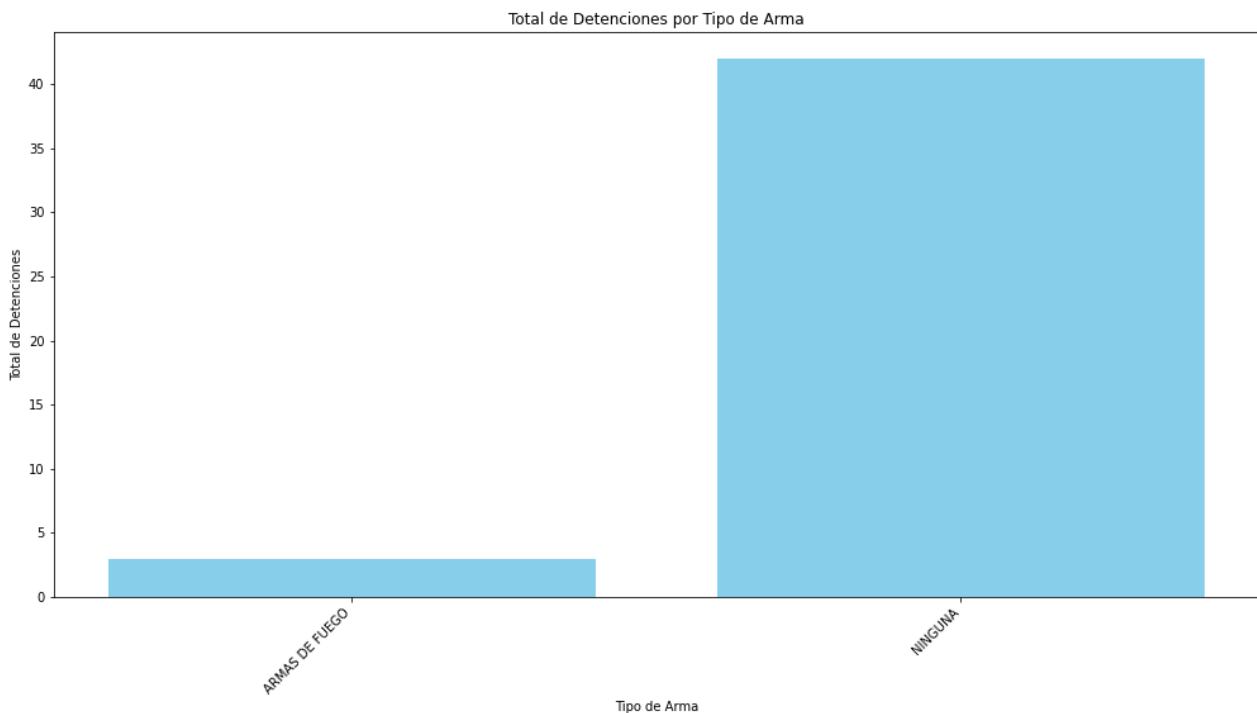
# Crear el gráfico de barras
fig, ax = plt.subplots(figsize=(14, 8))

ax.bar(arma_detenciones['TipoArma'], arma_detenciones['TotalDetenciones'], color='skyblue')

# Etiquetas y título
ax.set_title('Total de Detenciones por Tipo de Arma')
ax.set_xlabel('Tipo de Arma')
ax.set_ylabel('Total de Detenciones')

# Ajustar etiquetas y formato
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

FINISHED



Took 1 sec. Last updated by anonymous at July 25 2024, 1:31:51 PM.

Pregunta 2

FINISHED

Took 0 sec. Last updated by anonymous at July 25 2024, 11:34:29 AM.

```
%mysql(saveAs = segunda)
WITH detencionesDetalles AS (
  SELECT
    I.presunta_infraccion AS TipoInfraccion,
    L.tipo_lugar AS TipoLugarDetencion,
    D.edad AS EdadDetenido
```

FINISHED

settings

Tipos de infracción	Tipo de lugar de detención	Edad promedio	Cantidad Nacional de detenidos
null	ÁREA PRIVADA	32.2205	18
null	ÁREAS DE ACCESO PÚBLICO	31.4953	27
null	ÁREAS DEDICADAS AL COMERCIO	31.8009	8
null	CENTRO DE ADOLESCENTES INFRACTORES (CAI)	22.5970	2
null	CENTRO DE PRIVACIÓN PROVISIONAL DE LIBERTAD (CPPL)	31.0735	6

Took 1 sec. Last updated by anonymous at July 25 2024, 1:24:00 PM. (outdated)

```
%python
# Cargar el DataFrame de MySQL
segundaDf = z.getAsDataFrame("segunda")

# Mostrar el DataFrame
print(segundaDf)
```

FINISHED

	TipoInfraccion	...	RelacionVehiculosEdificios
0	NaN	...	21899.7643
1	NaN	...	21899.7643
2	NaN	...	21899.7643
3	NaN	...	21899.7643
4	NaN	...	21899.7643
5	NaN	...	21899.7643
6	NaN	...	21899.7643
7	NaN	...	21899.7643
8	NaN	...	21899.7643
9	NaN	...	21899.7643
10	NaN	...	21899.7643
11	NaN	...	21899.7643
12	NaN	...	21899.7643
13	NaN	...	21899.7643
14	NaN	...	21899.7643
15	NaN	...	21899.7643
...

Took 0 sec. Last updated by anonymous at July 25 2024, 1:24:03 PM.

```
%python
# Importar las librerías necesarias
import matplotlib.pyplot as plt
import pandas as pd

# Cargar el DataFrame de MySQL
segundaDf = z.getAsDataFrame("segunda")

# Crear el gráfico de barras para la Edad Promedio por Tipo de Infracción y Lugar de Detención
fig, ax = plt.subplots(figsize=(14, 8))

# Agrupar por TipoInfraccion y TipoLugarDetencion y graficar
labels = []
values = []

for key, grp in segundaDf.groupby(['TipoInfraccion', 'TipoLugarDetencion']):
    labels.append(f'{key[0]} - {key[1]}')
    values.append(grp['EdadPromedio'].values[0])

# Ordenar las etiquetas y valores
sorted_indices = sorted(range(len(labels)), key=lambda k: labels[k])
labels = [labels[i] for i in sorted_indices]
values = [values[i] for i in sorted_indices]

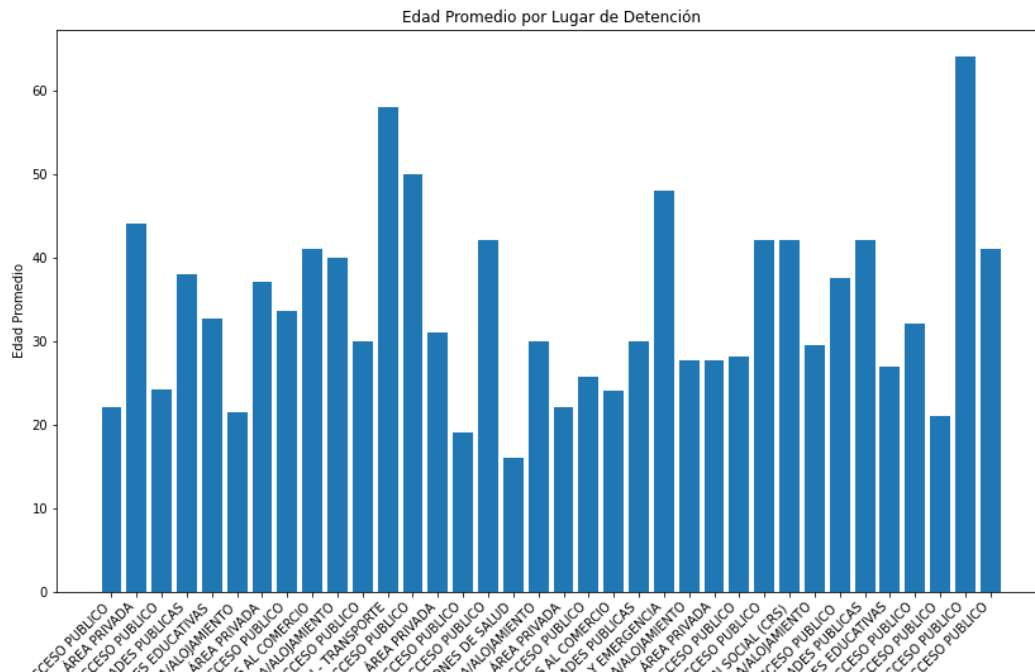
# Crear el gráfico de barras
ax.bar(labels, values)

# Etiquetas y título
ax.set_title('Edad Promedio por Lugar de Detención')
ax.set_xlabel('Tipo de Infracción - Lugar de Detención')
ax.set_ylabel('Edad Promedio')

plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

FINISHED

<stdin>:34: UserWarning: Tight layout not applied. The bottom and top margins cannot be made large enough to accommodate all Axes decorations.



Took 2 sec. Last updated by anonymous at July 25 2024, 1:24:14 PM.

```
%python
# Crear el gráfico de barras para la Edad Promedio por Tipo de Infracción
fig, ax = plt.subplots(figsize=(10, 6))

# Agrupar por TipoInfraccion y calcular la Edad Promedio
infraccion_group = segundaDf.groupby('TipoInfraccion')['EdadPromedio'].mean().reset_index()

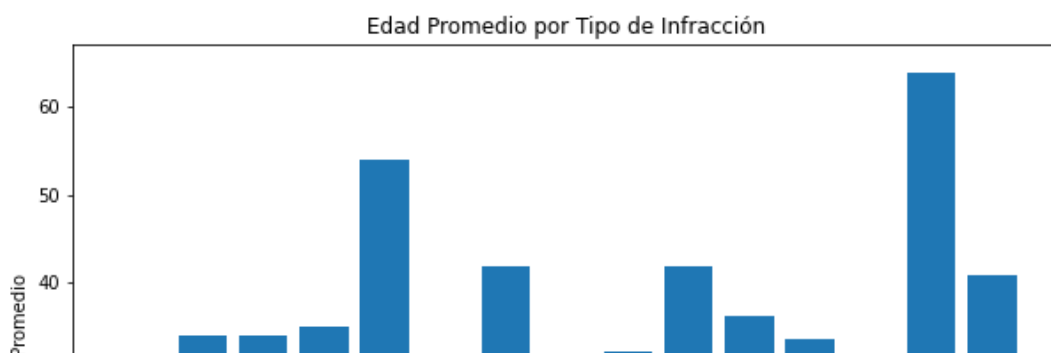
# Crear el gráfico de barras
ax.bar(infraccion_group['TipoInfraccion'], infraccion_group['EdadPromedio'])

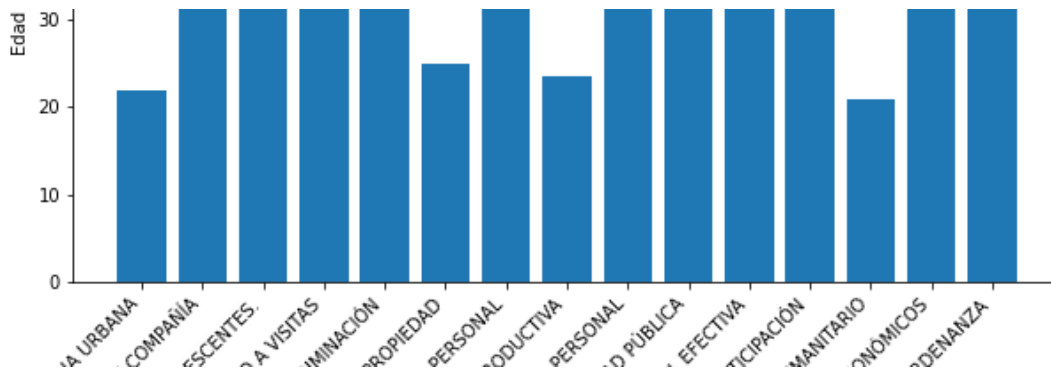
# Etiquetas y título
ax.set_title('Edad Promedio por Tipo de Infracción')
ax.set_xlabel('Tipo de Infracción')
ax.set_ylabel('Edad Promedio')

plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

FINISHED

<stdin>:17: UserWarning: Tight layout not applied. The bottom and top margins cannot be made large enough to accommodate all Axes decorations.





Took 2 sec. Last updated by anonymous at July 25 2024, 11:34:36 AM.

Pregunta3

FINISHED

Took 0 sec. Last updated by anonymous at July 25 2024, 11:34:36 AM.

```
%mysql(saveAs = tercera)
WITH DetencionesPorArmaYLugar AS (
  SELECT
    A.tipo_armas AS TipoArma,
    L.lugar AS LugarDetencion,
    EXTRACT(YEAR FROM D.fecha_detencion) AS AñoDetencion,
    COUNT(D.detenciones_id) AS TotalDetenciones
  FROM
    detenciones D
  JOIN
    armas A ON D.armas = A.armas
  JOIN
    lugares L ON D.lugar = L.lugar
  WHERE
    D.fecha_detencion BETWEEN '2023-01-01' AND '2023-12-31'
  GROUP BY
    A.tipo_armas, L.lugar, EXTRACT(YEAR FROM D.fecha_detencion)
),
TotalDetencionesPorLugar AS (
  SELECT
    LugarDetencion,
    AñoDetencion,
    SUM(TotalDetenciones) AS TotalPorLugar
  FROM
    DetencionesPorArmaYLugar
  GROUP BY
    LugarDetencion, AñoDetencion
),
DetencionesConPorcentaje AS (
  SELECT
    dpa.TipoArma,
    dpa.LugarDetencion,
    dpa.AñoDetencion,
    dpa.TotalDetenciones,
    tdl.TotalPorLugar,
    (dpa.TotalDetenciones / tdl.TotalPorLugar) * 100 AS Porcentaje
  FROM
    DetencionesPorArmaYLugar dpa
  JOIN
    TotalDetencionesPorLugar tdl
  ON
    dpa.LugarDetencion = tdl.LugarDetencion
    AND dpa.AñoDetencion = tdl.AñoDetencion
)
SELECT
  TipoArma,
  LugarDetencion,
  AñoDetencion,
  TotalDetenciones,
  TotalPorLugar,
```

FINISHED

settings ▼

TipoArma	LugarDetencion	AñoDetencion	TotalDetenciones
NINGUNA	AEROPUERTO	2023	293
ARMAS DE FUEGO	AEROPUERTO	2023	2
ARMAS BLANCAS (OBJET	AEROPUERTO	2023	1
NINGUNA	AEROPUERTO CARGUERAS	2023	7
NINGUNA	AEROPUERTO CURIER	2023	1
NINGUNA	AGENCIA DE REGULACIÓN Y CONTROL	2023	4

Took 1 sec. Last updated by anonymous at July 25 2024, 11:34:37 AM. (outdated)

```
%python
# Cargar el DataFrame de MySQL
terceraDf = z.getAsDataFrame("tercera")

# Mostrar el DataFrame
print(terceraDf)
```

FINISHED

	TipoArma	LugarDetencion	...	TotalPorLugar	Porcentaje
0	NINGUNA	AEROPUERTO	...	296	98.99
1	ARMAS DE FUEGO	AEROPUERTO	...	296	0.68
2	ARMAS BLANCAS (OBJET	AEROPUERTO	...	296	0.34
3	NINGUNA	AEROPUERTO CARGUERAS	...	7	100.00
4	NINGUNA	AEROPUERTO CURIER	...	1	100.00
..
325	NINGUNA	VÍA PÚBLICA	...	48300	88.34
326	ARMAS DE FUEGO	VÍA PÚBLICA	...	48300	8.53
327	ARMAS BLANCAS (OBJET	VÍA PÚBLICA	...	48300	3.13
328	NINGUNA	VULCANIZADORA	...	4	50.00
329	ARMAS DE FUEGO	VULCANIZADORA	...	4	50.00

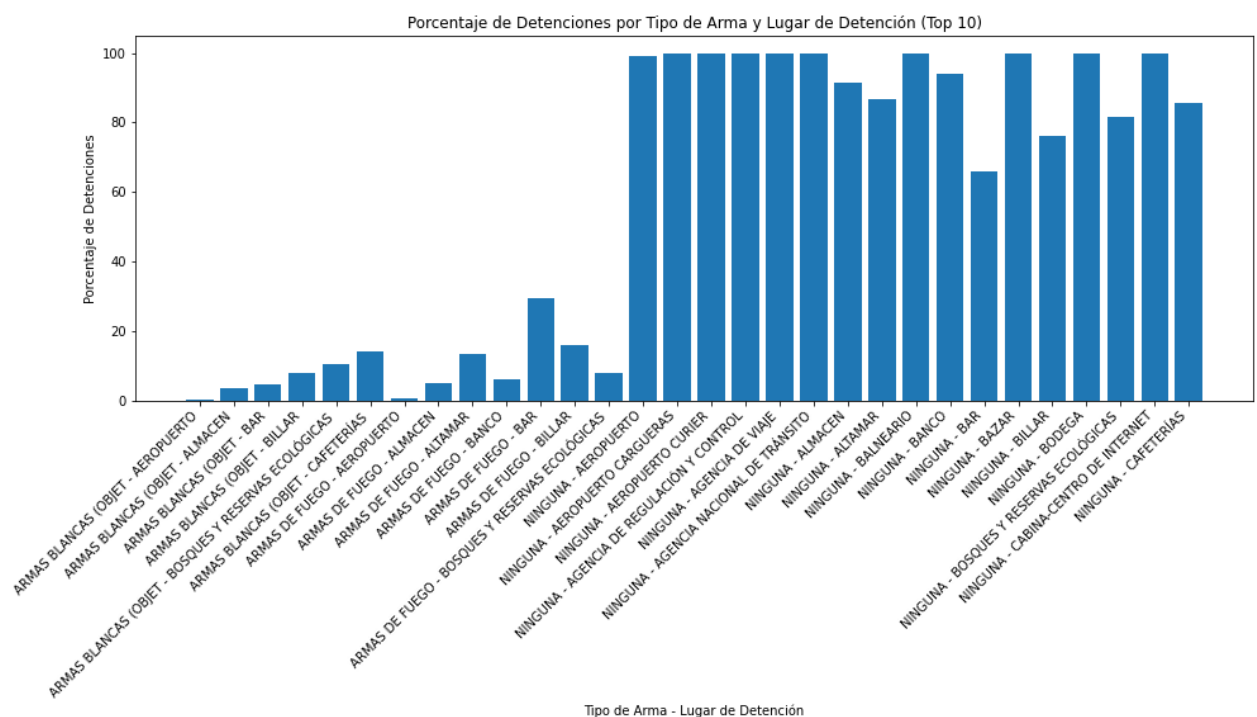
[330 rows x 6 columns]

Took 0 sec. Last updated by anonymous at July 25 2024, 11:34:37 AM.

```
%python
# Importar las librerías necesarias
import matplotlib.pyplot as plt
import pandas as pd

# Limitar el DataFrame a los primeros 10 registros
limitedDf = terceraDf.head(30)
```

FINISHED



Took 3 sec. Last updated by anonymous at July 25 2024, 11:34:40 AM.

Pregunta 4

FINISHED

Took 0 sec. Last updated by anonymous at July 25 2024, 11:34:40 AM.

```
%mysql(saveAs = cuarta)
SELECT
  D.arma AS TipoArma,
  L.lugar AS Lugar,
  L.tipo_lugar AS TipoLugar,
  COUNT(D.detenciones_id) AS TotalDetenciones,
```

FINISHED

settings ▼

TipoArma ▼	Lugar ▼	TipoLugar ▼	TotalDetenciones ▼	En
NINGUNA	VÍA PÚBLICA	ÁREAS DE ACCESO PUBLICO	45812	202

Took 3 sec. Last updated by anonymous at July 25 2024, 11:34:44 AM. (outdated)

```
%python
# Cargar el DataFrame de MySQL
cuartaDf = z.getAsDataFrame("cuarta")

# Mostrar el DataFrame
print(cuartaDf)
```

FINISHED

	TipoArma	...	Movilizaciones
0	NINGUNA	...	A PIE,ACÉMILAS,AUTOMOVIL,BICICLETA,BOTE,BUS,CA...
1	NINGUNA	...	A PIE,ACÉMILAS,AUTOMOVIL,BICICLETA,BUS,CAMIONE...
2	NINGUNA	...	A PIE,ACÉMILAS,AUTOMOVIL,BICICLETA,BUS,CAMIONE...
3	REVÓLVER	...	A PIE,ACÉMILAS,AUTOMOVIL,BICICLETA,BUS,CAMIONE...
4	PISTOLA	...	A PIE,ACÉMILAS,AUTOMOVIL,BICICLETA,BOTE,BUS,CA...
..
606	SUBAMETRALLADORA	...	NO APLICA
607	TIJERAS	...	A PIE
608	TIJERAS	...	A PIE
609	TIJERAS	...	A PIE
610	TIJERAS	...	A PIE

[611 rows x 9 columns]

Took 1 sec. Last updated by anonymous at July 25 2024, 11:34:45 AM.

```
%python
import matplotlib.pyplot as plt

# Cargar el DataFrame de MySQL
cuartaDf = z.getAsDataFrame("cuarta")

# Limitar el DataFrame a los primeros 20 registros
limitedDf = cuartaDf.head(20)

# Extraer y contar las nacionalidades
nacionalidades = limitedDf[['Nacionalidades', 'TotalDetenciones']].copy()
nacionalidades = nacionalidades.dropna()
nacionalidades['Nacionalidades'] = nacionalidades['Nacionalidades'].apply(lambda x: x.split(','))

# Expandir la lista de nacionalidades en filas separadas
expanded_nacionalidades = nacionalidades.explode('Nacionalidades')

# Agrupar por nacionalidad y sumar el total de detenciones
total_detenciones_por_nacionalidad = expanded_nacionalidades.groupby('Nacionalidades')['TotalDetenciones'].sum()

# Ordenar por total de detenciones y tomar los primeros 20
top_20_nacionalidades = total_detenciones_por_nacionalidad.sort_values(by='TotalDetenciones', ascending=False).head(20)

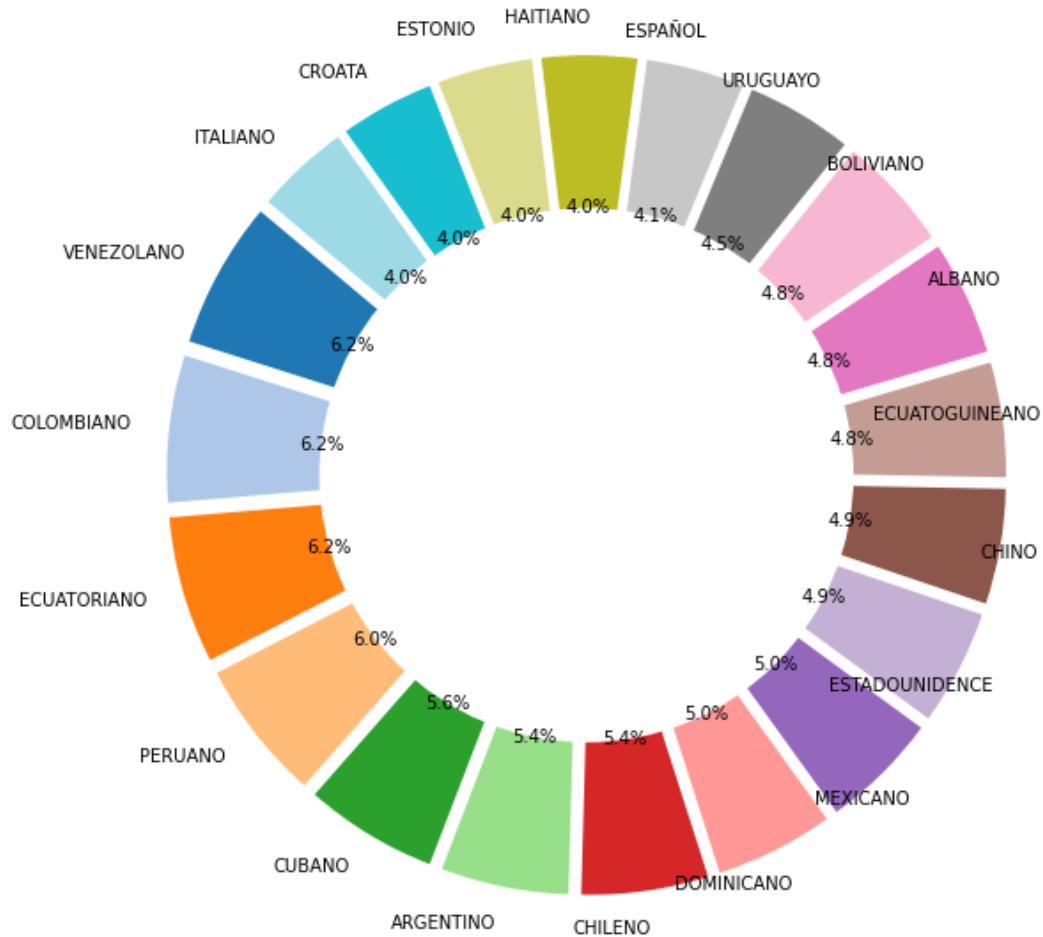
# Crear el gráfico de tarta
fig, ax = plt.subplots(figsize=(10, 8))

# Tarta con porcentaje y colores predefinidos
wedges, texts, autotexts = ax.pie(
    top_20_nacionalidades['TotalDetenciones'],
    labels=top_20_nacionalidades['Nacionalidades'],
    autopct='%1.1f%%',
    startangle=140,
    colors=plt.get_cmap('tab20').colors[:20], # Colores predefinidos para múltiples categorías
    explode=[0.1] * len(top_20_nacionalidades) # Resalta todas las partes un poco
)

# Ajustar el tamaño y la separación de las etiquetas
for text in texts:
    text.set_fontsize(10)
    text.set_color('black')
    text.set_horizontalalignment('right') # Alinear horizontalmente
for autotext in autotexts:
    autotext.set_fontsize(10)
    autotext.set_color('black')
```

FINISHED

Distribución de Detenciones por Nacionalidad (Primeros 20 Datos)



Took 4 sec. Last updated by anonymous at July 25 2024, 11:34:49 AM.

Pregunta 5

FINISHED

Took 0 sec. Last updated by anonymous at July 25 2024, 11:34:49 AM.

```
%mysql(saveAs = quinta)
WITH DetencionesDetalles AS (
  SELECT
    I.presunta_infraccion AS TipoInfraccion,
    L.tipo_lugar AS TipoLugarDetencion,
    D.edad AS EdadDetenido,
    D.estado_civil AS EstadoCivilDetenido,
    D.nacionalidad AS NacionalidadDetenido,
    V.num_vehiculos AS NumeroVehiculos,
    E.num_edificios_admin AS NumeroEdificios
  FROM
    detenciones D
```

FINISHED

 settings ▼

TipoInfraccion	TipoLugarDetencion	EdadPromedio	CantidadNacionalidad	EdadPromedio
null	ÁREAS DE ACCESO PUBLICO	31.4953	27	
null	ÁREAS DEDICADAS AL COMERCIO	31.8009	8	
null	CENTRO DE ADOLESCENTES INFRACTORES (CAI)	22.5970	2	
null	CENTRO DE PRIVACIÓN PROVISIONAL DE LIBERTAD (CPPL)	31.0735	6	

Took 3 sec. Last updated by anonymous at July 25 2024, 1:14:25 PM. (outdated)

```
%python
# Cargar el DataFrame de MySQL
quintaDf = z.getAsDataFrame("quinta")
```

FINISHED

	TipoInfraccion	...	RelacionVehiculosEdificios
0	NaN	...	21899.7643
1	NaN	...	21899.7643
2	NaN	...	21899.7643
3	NaN	...	21899.7643
4	NaN	...	21899.7643
5	NaN	...	21899.7643
6	NaN	...	21899.7643
7	NaN	...	21899.7643
8	NaN	...	21899.7643
9	NaN	...	21899.7643
10	NaN	...	21899.7643
11	NaN	...	21899.7643
12	NaN	...	21899.7643
13	NaN	...	21899.7643
14	NaN	...	21899.7643
15	NaN	...	21899.7643
...	21899.7643

Took 9 sec. Last updated by anonymous at July 25 2024, 1:14:34 PM.

```
%python
# Importar las librerías necesarias
import matplotlib.pyplot as plt
import pandas as pd

# Cargar el DataFrame de MySQL
quintaDf = z.getAsDataFrame("quinta")

# Limitar el DataFrame a los primeros 30 registros
limitedDf = quintaDf.head(30)

# Crear etiquetas combinando TipoInfraccion y TipoLugarDetencion
labels = [f'{row["TipoInfraccion"]} - {row["TipoLugarDetencion"]}' for index, row in limitedDf]

# Crear el gráfico de líneas para la Edad Promedio
fig, ax = plt.subplots(figsize=(16, 10))

# Graficar los puntos
ax.plot(labels, limitedDf['EdadPromedio'], marker='o', linestyle='-', color='coral', alpha=0.7)

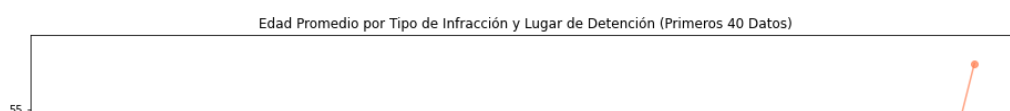
# Etiquetas y título
ax.set_title('Edad Promedio por Tipo de Infracción y Lugar de Detención (Primeros 40 Datos)')
ax.set_xlabel('Tipo de Infracción - Lugar de Detención')
ax.set_ylabel('Edad Promedio')

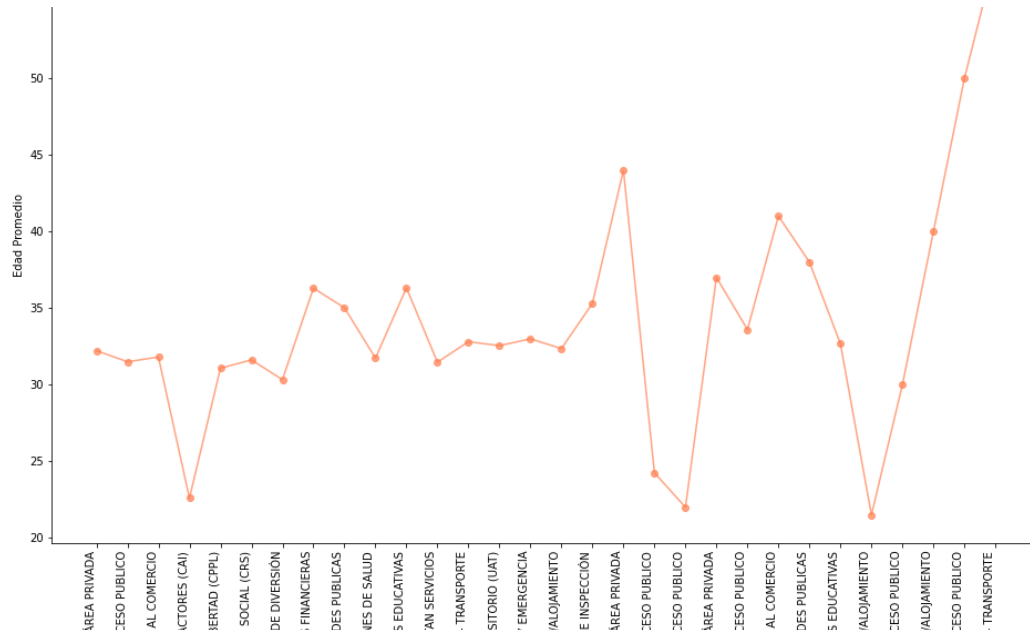
# Rotar etiquetas para mejor legibilidad
plt.xticks(rotation=90, ha='right')

# Ajustar el diseño
plt.tight_layout()
plt.show()
```

FINISHED

<stdin>:30: UserWarning: Tight layout not applied. The bottom and top margins cannot be made large enough to accommodate all Axes decorations.





Took 2 sec. Last updated by anonymous at July 25 2024, 1:14:36 PM. (outdated)

```
%python
```

FINISHED

Took 0 sec. Last updated by anonymous at July 25 2024, 11:34:56 AM.