Trabajo Práctico 1
Laboratorio de Datos

1° cuatrimestre 2023

Grupo: La Scaloneta 🇦🇷🇦🇷🇦🇷⭐⭐⭐

Integrantes del grupo:

- Axel Belbrun
- Mauricio Enrich
- Giovanni Paredes

RESUMEN:

En este trabajo se nos planteó la actividad de utilizar una base de datos para responder preguntas acerca de un determinado sector de la economía en Argentina.
Como esta base de datos no estaba en condiciones óptimas para ser utilizada de forma directa, hubo que realizar un proceso de "limpieza" de la misma. Esto último incluía:

- Una reestructuración de las tablas crudas para que cumplan con normas para bases de datos.
- Correcciones de ortografía
- El establecimiento de una -convención-, para la forma de escribir los datos y los atributos (por ejemplo: Buenos Aires, BUENOS AIRES, buenos aires), y la posterior adaptación de los mismos a ésta.

Luego de este proceso, podemos utilizar esta base de datos para responder fehacientemente, a través de "consultas a la base", las preguntas que nos hicieron.

Introducción:

Dada la necesidad de saber si existe una relación entre el desarrollo de una actividad y el salario medio de quienes la realizan, se obtuvo un conjunto de datos que podrían ayudar a encontrar indicios y sacar conclusiones al respecto.
El conjunto de datos no fue obtenido del mismo 'proveedor', por lo que no existía un consenso general acerca de cómo se cargaron los mismos, generando un problema a la hora de querer analizarlos. Para resolver estos problemas usamos las herramientas y conceptos vistos en la materia para tratar con bases de datos tanto desde la parte práctica (Pandas, SQL) como desde la parte teórica (formas normales, modelado de datos). Reestructuramos las tablas de modo de asegurarnos que cumplan con la 3FN (**E.F. Codd, 1971**), basándonos en el Diagrama Entidad-Relación y en el Modelo Relacional.
Una vez 'normalizada' nuestra base de datos, realizamos una limpieza de los datos de modo que todas las tablas sean congruentes entre sí, porque si bien para nosotros es inmediato que "Buenos Aires" y "BUENOS AIRES", por ejemplo, son dos maneras de referirse al mismo objeto, para una computadora esto no es así. Este proceso fue realizado tanto con Pandas como con SQL, dado que algunas cosas eran más directas con una u otra herramienta.
Luego del proceso de normalización y limpieza de la base de datos, estuvimos en condiciones de responder la pregunta principal y algunas más (ejercicios I y J), haciendo análisis exploratorio de los datos y realizando consultas a la base a través de "consultas SQL".

Decisiones tomadas:

En la tabla de localidades censales:

- **Columna Categoría**: eliminada. Aclaraba si se trataba de una localidad simple o compuesta. No le encontramos utilidad.
- **Columna Función**: eliminada. Indicaba cabecera de departamento y capitales, contenía en su mayoría nulls.
- **Columna Fuente**: eliminada. Redundancia en sus valores (en todas las filas tenía el mismo valor).

En la tabla de padrón de operadores organicos:

- **Columna País**: eliminada. El padrón era de operadores orgánicos de Argentina, no tenía sentido aclararlo en una columna.
- **Columna País_id**: eliminada. Misma razón que la columna "País".
- **Columna Localidad**: eliminada. La mayoría de las entradas en esta columna era NULL's.

En la tabla de "diccionario_clae2":

- Se cambiaron los nombres de "clae2" y "letra" por "rubro" y "actividad" respectivamente, para que tengan nombres más representativos.

- Se le asignó la letra "Z" a la categoría "OTROS" para no generar tuplas espúreas a la hora de hacer joins.

Procesamiento de Datos:

La principal transformación realizada a los datos fue la reestructuración de los mismos, dado que no cumplían con ninguna forma normal de base de datos.
Se separaron las tablas originales en tablas más pequeñas, de modo de eliminar redundancias, relaciones dentro de relaciones, y atributos multivaluados/compuestos.
En particular:

- La tabla original “diccionario_clae2”, con primary key “clae2” tenía estas DFs:
    - DF1: clae2 -> {clae2_desc, letra}
    - DF2: letra -> letra_desc (dependencia transitiva)

La DF2 nos determinó una nueva tabla, llamada “Actividades” (luego de renombrar “letra” como “actividad_id” y “letra_desc” como “Actividad”), dado que era una dependencia funcional transitiva.

La DF1 permaneció igual, pero igualmente decidimos recomponer la tabla en otras dos más pequeñas. Una para “rubro_id” (antes “clae2”) y “actividad_id”, y otra para “rubro_id” y “rubro” (antes “clae2_desc”).
Ninguna de las DF’s de la tabla original permaneció.

- La tabla original “diccionario_cod_depto”, con primary key “código_departamento_indec”, tenía estas DF’s:
    - DF1: código_departamento_indec -> {nombre_departamento_indec, código_provincia_indec}
    - DF2: código_provincia_indec -> nombre_provincia_indec (dependencia transitiva)

La DF2 nos determina una nueva tabla, llamada “Provincias” (renombrando “nombre_provincia_indec” como “prov_nombre” y “código_provincia_indec” como “prov_id”), dado que era una dependencia funcional transitiva.

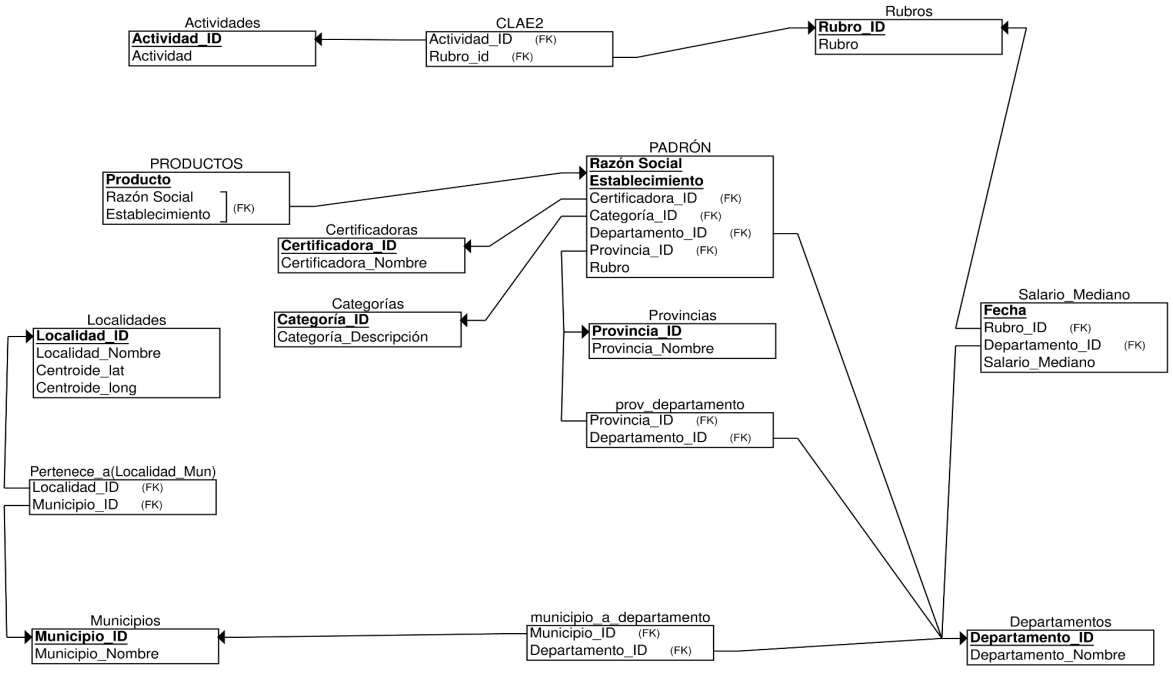
La DF1 permaneció igual, pero también decidimos recomponer la tabla en otras dos más pequeñas. Una para “departamento_id” (antes “código_departamento_indec”) y “departamento_nombre” (antes “nombre_departamento_indec”) y otra para "departamento_id” y “prov_id”.
Ninguna de las DF’s de la tabla original permaneció.

- La tabla original “localidades_censales”, con primary key “id” tenía múltiples DF’s, así que decidimos desde el inicio separar, en principio, todas las columnas de la forma “atributo_id” y “atributo_nombre” (o parecidos) en tablas nuevas, pero conservando las columnas “atributo_id” en la tabla original.
  Esto nos deja una única DF: localidad_id (antes “id”) -> {localidad_nombre, departamento_id, provincia_id, centroide_lat, centroide_lon}
- La tabla original “padron_de_operadores_organicos”, con primary key {razón_social, establecimiento}, tenía las siguientes DF’s:
    - DF1: {razón social, establecimiento} -> productos

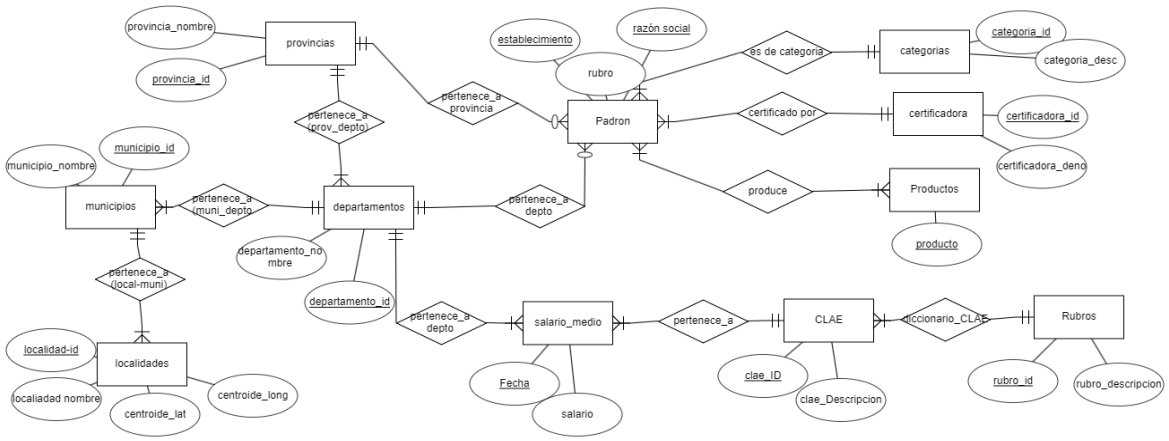
- Otras DF's que caían en el mismo criterio de "atributo_id" y "atributo_nombre", las cuáles fueron tratadas igual que en tablas anteriores.
  La DF1 nos mostraba un problema en la columna de "productos", ya que era un atributo multivaluado, por lo cual hubo que separar esto en una nueva relación "productos" en la cual, para cada producto del conjunto de productos asociado a una tupla, se generaba una nueva tupla (razón, establecimiento, producto).

- La tabla "w_median_depto_priv_clae2" no presentaba anomalías, por lo que no sufrió más que cambios de nombre, y su única DF era:
  - DF1: {fecha, departamento_id, provincia_id, rubro_id} -> salario_mediano

Realizamos un proceso "inverso" al que sugería el orden de los ejercicios, y primero creamos el Modelo Relacional a partir de las tablas que nos quedaron como resultado de normalizar la base:



Haciendo esto, luego realizar el DER resultó más sencillo, dado que no era del todo intuitivo, y es que ya estamos iniciando el trabajo con tablas existentes, y no estamos modelando desde 0. Pasamos entonces al desarrollo del DER. Nosotros sabíamos que había 2 tablas principales en nuestra base de datos, salario_mediano y padrón. Como no había una relación directa entre estas tablas, pensamos que la mejor decisión seria hacer el DER de cada uno por separado y gracias a eso luego quedaría más clara la relación entre estos 2. Una vez hecho esto pudimos ver que estas se relacionaban mediante la tabla departamento, lo cual tiene sentido ya que salario_mediano habla de los salarios del sector privado por actividad y departamento desde 2014, mientras que padrón trata sobre la producción de distintos establecimientos donde el departamento en el que se encuentran tiene gran relevancia ya que segun el departamento en el que se encuentren habrá distintos establecimientos productores, por lo que es razonable comparar el salario mediano por departamento con la producción por departamento. El único inconveniente que encontramos fue que en padrón los rubros principales son la fruticultura, agricultura, horticultura, etc. que no representan la totalidad de las actividades nacionales (cosa que sí hace salario_mediano), lo tuvimos en cuenta a la hora de comparar los salarios:



Cómo limpiamos los datos:

1) Comenzando por localidades censales eliminamos las columnas que vamos a trasladar a otras tablas y los NULLs:

-Eliminamos primero las columnas "función" y "fuente" usando drop en pandas ya que estas no aportan nada.

-Para las filas que puedan tener NULLs usamos dropna.

- Luego ponemos en mayúsculas las columnas municipio_nombre y localidad_nombre

- Por último eliminamos las tildes de departamento_nombre

2) Para poder leer la tabla “padrón” tuvimos que usar encoding 1252 ya que no leía en utf-8. Y debido al problema de ingresar al nombre de la columna razón social lo renombramos a razon_social (sin tilde) con “rename”.
3) En la tabla “diccionario_cod_depto” hacemos algo parecido al punto 1, que es poner todos los nombres en mayúsculas y quitarle las tildes, así hay congruencia entre ambas tablas. Luego renombramos 'codigo_departamento_indec' como departamento_id y 'nombre_departamento_indec' como departamento_nombre para que tengan
nombres más cortos y declarativos
4) En la tabla dicc_clae2 arreglamos el problema del NULL en la actividad 'Otros' asignándole la letra 'Z', ya que antes no tenía asignado nada.
Luego renombramos 'clae2_desc' como 'rubro' y 'clae2' como 'rubro_id' para que tengan nombres más declarativos, hacemos lo mismo con 'letra' y 'letra_desc' renombrándolos como 'actividad_id' y 'actividad'.
5) En la tabla “cod_departamento” cambiamos “'id_provincia_indec' y 'nombre_provincia_indec' renombrándolos como 'prov_id' y 'prov_nombre' respectivamente. La misma idea la aplicamos con departamentos, renombrando codigo_departamento_indec como departamento_id y nombre_departamento_indec como departamento_nombre.
6) En la tabla de salarios Eliminamos los registros que tengan NULLS en el departamento y/o salario mediano<0.

Análisis de datos:

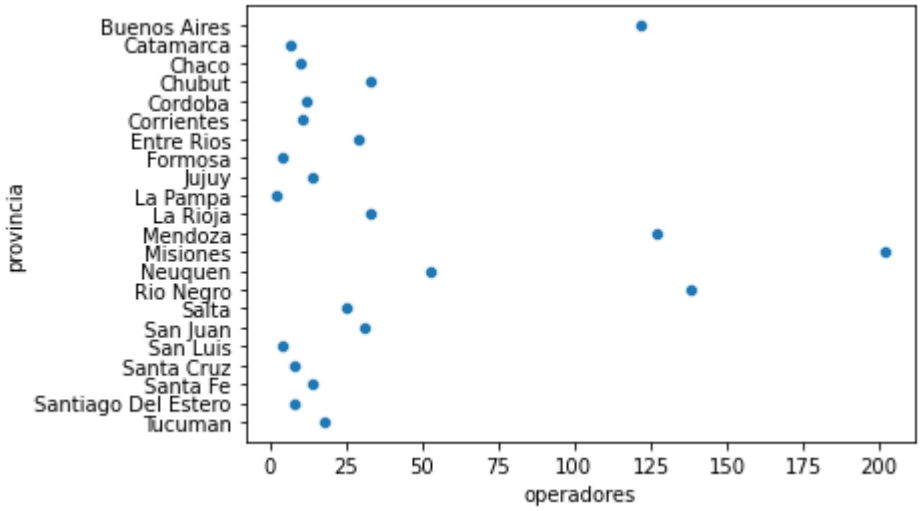
Para realizar este apartado, asumimos que todas las actividades que aparecen en el padrón pertenecen al rubro correspondiente a rubro_id = 1 (Agricultura, ganadería, caza y servicios relacionados)

Ejercicio i)

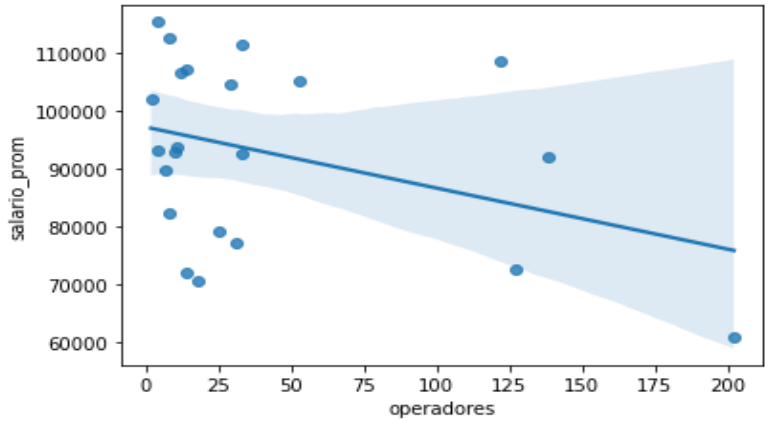
1. La única provincia sin operadores es CABA.
2. Existen varios departamentos sin operadores, en total son 333 y están representados en el data frame “departamentos_sin_operadores”
3. La actividad con más operadores es “agricultura”.
4. El salario promedio de la agricultura en 2022 fue de $195844 (Diciembre 2022)
5. El promedio anual de los salarios en Argentina es de $60902, con un desvío de $62433. Pero estos números no son realmente representativos ya que el monto es altamente variable, y habría que hacer algún cálculo previo para poder hacer una comparativa más realista, por ejemplo, teniendo los datos de la inflación mes a mes desde 2014 hasta 2023.

Ejercicio j)

1. Cantidad de operadores por provincia:



2. - Gráficos en código.
3. Si bien no puede verse a qué provincia corresponde cada punto (no encontramos manera de hacerlo prolijamente), puede verse con la recta de regresión lineal que cuanto menos emprendimientos certificados mejor salario.



4.

Conclusiones:

Dados los gráficos que obtuvimos y el análisis sobre los datos al responder las preguntas, podría concluirse que existe una relación inversamente proporcional -no muy fuerte- entre la cantidad de productores por provincia y el salario

Decimos que -no muy fuerte- porque los datos están muy dispersos, y podría estar sucediendo que al haber menos competencia la ganancia es mayor.

Intuitivamente se creería que al estar más desarrollada la actividad (más operadores), esta se desarrollaría y crearía más ganancias. Pero un análisis de ese tipo se nos escapa totalmente.

Notamos, como muchas veces comentaron los docentes, que la mayoría del trabajo fue la limpieza y reestructuración de los datos. Las preguntas concretas pudieron responderse rápidamente una vez "acomodada" la base de datos.