

8bd6f37a-7163-4d75-9372-e17ec4cbf0e3

July 13, 2024

Una empresa de comercio electrónico, Store 1, recientemente comenzó a recopilar datos sobre sus clientes. El objetivo final de Store 1 es comprender mejor el comportamiento de sus clientes y tomar decisiones basadas en datos para mejorar su experiencia online.

Como parte del equipo de análisis, tu primera tarea es evaluar la calidad de una muestra de datos recopilados y prepararla para futuros análisis.

## 1 Cuestionario

Store 1 tiene como objetivo garantizar la coherencia en la recopilación de datos. Como parte de esta iniciativa, se debe evaluar la calidad de los datos recopilados sobre los usuarios y las usuarias. Te han pedido que revises los datos recopilados y propongamos cambios. A continuación verás datos sobre un usuario o una usuaria en particular; revisa los datos e identifica cualquier posible problema.

```
[ ]: user_id = '32415'
      user_name = ' mike_reed '
      user_age = 32.0
      fav_categories = ['ELECTRONICS', 'SPORT', 'BOOKS']
```

### Opciones:

1. El tipo de datos para `user_id` debe cambiarse de una cadena a un número entero.
2. La variable `user_name` contiene una cadena que tiene espacios innecesarios y un guion bajo entre el nombre y el apellido.
3. El tipo de datos de `user_age` es incorrecto.
4. La lista `fav_categories` contiene cadenas en mayúsculas. En su lugar, deberíamos convertir los valores de la lista a minúsculas.

Escribe en la celda Markdown a continuación el número de las opciones que has identificado como problemas. Si has identificado varios problemas, sepáralos con comas. Por ejemplo, si piensas que los números 1 y 3 son incorrectos, escribe 1, 3.

**Escribe tu respuesta y explica tu argumentación:**

## 2 Ejercicio 1

Vamos a implementar los cambios que identificamos. Primero, necesitamos corregir los problemas de la variable `user_name`. Como vimos, tiene espacios innecesarios y un guion bajo como separador

entre el nombre y el apellido; tu objetivo es eliminar los espacios y luego reemplazar el guion bajo con el espacio.

```
[4]: user_name = ' mike_reed '  
user_name = user_name.strip(' ') # eliminar los espacios en la cadena original  
user_name = user_name.replace('_', ' ') # reemplazar el guion bajo con el espacio  
  
print(user_name)
```

mike reed

### 3 Ejercicio 2

Luego, debemos dividir el `user_name` (nombre de usuario o usuaria) actualizado en dos subcadenas para obtener una lista que contenga dos valores: la cadena para el nombre y la cadena para el apellido.

```
[5]: user_name = 'mike reed'  
name_split = user_name.split(' ') # divide aquí el string user_name  
  
print(name_split)
```

['mike', 'reed']

### 4 Ejercicio 3

¡Genial! Ahora debemos trabajar con la variable `user_age`. Como ya mencionamos, esta tiene un tipo de datos incorrecto. Arreglemos este problema transformando el tipo de datos y mostrando el resultado final.

```
[7]: user_age = 32.0  
user_age = int(user_age) # cambia el tipo de datos para la edad de un usuario o  
    ↪ usuaria  
  
print(user_age)
```

32

### 5 Ejercicio 4

Como sabemos, los datos no siempre son perfectos. Debemos considerar escenarios en los que el valor de `user_age` no se pueda convertir en un número entero. Para evitar que nuestro sistema se bloquee, debemos tomar medidas con anticipación.

Escribe un código que intente convertir la variable `user_age` en un número entero y asigna el valor transformado a `user_age_int`. Si el intento falla, mostramos un mensaje pidiendo al usuario o la usuaria que proporcione su edad como un valor numérico con el mensaje: **Please provide your age as a numerical value.** (Proporcione su edad como un valor numérico.)

```
[19]: user_age = 'treinta y dos' # aquí está la variable que almacena la edad como un
      ↪ string.

      # escribe un código que intente transformar user_age en un entero y si falla,
      ↪ imprime el mensaje especificado

      try:
          uses_age_int = int(user_age)
          print(uses_age_int)
      except ValueError:
          print('Please provide your age as a numerical value')
```

Please provide your age as a numerical value

## 6 Ejercicio 5

Finalmente, considera que todas las categorías favoritas se almacenan en mayúsculas. Para llenar una nueva lista llamada `fav_categories_low` con las mismas categorías, pero en minúsculas, itera los valores en la lista `fav_categories`, modifícalos y agrega los nuevos valores a la lista `fav_categories_low`. Como siempre, muestra el resultado final.

```
[14]: fav_categories = ['ELECTRONICS', 'SPORT', 'BOOKS']
      fav_categories_low = []

      # escribe tu código aquí
      for category in fav_categories:
          fav_categories_low.append(category.lower())

      # no elimines la siguiente declaración print
      print(fav_categories_low)
```

['electronics', 'sport', 'books']

## 7 Ejercicio 6

Hemos obtenido información adicional sobre los hábitos de gasto de nuestros usuarios y usuarias, incluido el importe gastado en cada una de sus categorías favoritas. La gerencia está interesada en las siguientes métricas:

- Importe total gastado por el usuario o la usuaria.
- Importe mínimo gastado.
- Importe máximo gastado.

Vamos a calcular estos valores y mostrarlos en la pantalla:

```
[20]: fav_categories_low = ['electronics', 'sport', 'books']
      spendings_per_category = [894, 213, 173]
```

```
total_amount = sum(spendings_per_category) # escribe tu código aquí
max_amount = max(spendings_per_category) # escribe tu código aquí
min_amount = min(spendings_per_category) # escribe tu código aquí

# no elimines la siguiente declaración print
print(total_amount)
print(max_amount)
print(min_amount)
```

1280

894

173

## 8 Ejercicio 7

La empresa quiere ofrecer descuentos a sus clientes leales. Los clientes y las clientas que realizan compras por un importe total mayor a \$1500 se consideran leales y recibirán un descuento.

Nuestro objetivo es crear un bucle **while** que compruebe el importe total gastado y se detenga al alcanzarlo. Para simular nuevas compras, la variable **new\_purchase** genera un número entre 30 y 80 en cada iteración del bucle. Esto representa el importe de dinero gastado en una nueva compra y es lo que hay que sumar al total.

Una vez que se alcance el importe objetivo y se termine el bucle **while**, se mostrará la cantidad final.

```
[32]: from random import randint

total_amount_spent = 1280
target_amount = 1500

while total_amount_spent <= target_amount: # escribe tu código aquí
    new_purchase = randint(30, 80) # generamos un número aleatorio de 30 a 80
    total_amount_spent += new_purchase # escribe tu código aquí

print(total_amount_spent)
```

1565

## 9 Ejercicio 8

Ahora tenemos toda la información sobre un cliente o una clienta de la forma que queremos que sea. La gerencia de una empresa nos pidió proponer una forma de resumir toda la información sobre un usuario o una usuaria. Tu objetivo es crear una cadena formateada que utilice información de las variables **user\_id**, **user\_name** y **user\_age**.

Esta es la cadena final que queremos crear: **User 32415 is mike who is 32 years old.** (El usuario 32415 es Mike, quien tiene 32 años).

```
[39]: user_id = '32415'
user_name = ['mike', 'reed']
user_age = 32

full_name = f"{user_name[0]} {user_name[1]}"

user_info = f"User {user_id} is {full_name} who is {user_age} years old" #
↳ escribe tu código aquí

# no elimines la siguiente declaración print
print(user_info)
```

User 32415 is mike reed who is 32 years old

Como sabes, las empresas recopilan y almacenan datos de una forma particular. Store 1 desea almacenar toda la información sobre sus clientes y clientas en una tabla.

user_id	user_name	user_age	purchase_category	spending_per_category
'32415'	'mike', 'reed'	32	'electronics', 'sport', 'books'	894, 213, 173
'31980'	'kate', 'morgan'	24	'clothes', 'shoes'	439, 390

En términos técnicos, una tabla es simplemente una lista anidada que contiene una sublista para cada usuario o usuaria.

Store 1 ha creado una tabla de este tipo para sus usuarios y usuarias. Se almacena en la variable `users`. Cada sublista contiene el ID del usuario o la usuaria, nombre y apellido, edad, categorías favoritas y el importe gastado en cada categoría.

## 10 Ejercicio 9

Para calcular los ingresos de la empresa, sigue estos pasos.

1. Utiliza `for` para iterar sobre la lista `users`.
2. Extrae la lista de gastos de cada usuario o usuaria y suma los valores.
3. Actualiza el valor de los ingresos con el total de cada usuario o usuaria.

Así obtendrás los ingresos totales de la empresa que mostrarás en la pantalla al final.

```
[43]: users = [
    # este es el inicio de la primera sublista
    ['32415', ['mike', 'reed'], 32, ['electronics', 'sport', 'books'],
    [894, 213, 173]
], # este es el final de la primera sublista

    # este es el inicio de la segunda sublista
    ['31980', ['kate', 'morgan'], 24, ['clothes', 'shoes'],
    [439, 390]]
```

```

    ] # este es el final de la segunda sublista
]

revenue = 0

for user in users:
    spendings_list = user[4] # extrae la lista de gastos de cada usuario o
    ↪ usuaria y suma los valores
    total_spendings = sum(spendings_list) # suma los gastos de todas las
    ↪ categorías para obtener el total de un usuario o una usuaria en particular
    revenue += total_spendings # actualiza los ingresos

# no elimines la siguiente declaración print
print(revenue)

```

2109

## 11 Ejercicio 10

Recorre la lista de usuarios y usuarias que te hemos proporcionado y muestra los nombres de la clientela menor de 30 años.

```

[45]: users = [
    ['32415', ['mike', 'reed'], 32, ['electronics', 'sport', 'books'],
    [894, 213, 173]],
    ['31980', ['kate', 'morgan'], 24, ['clothes', 'books'], [439,
    390]],
    ['32156', ['john', 'doe'], 37, ['electronics', 'home', 'food'],
    [459, 120, 99]],
    ['32761', ['samantha', 'smith'], 29, ['clothes', 'electronics',
    'beauty'], [299, 679, 85]],
    ['32984', ['david', 'white'], 41, ['books', 'home', 'sport'], [234,
    329, 243]],
    ['33001', ['emily', 'brown'], 26, ['beauty', 'home', 'food'], [213,
    659, 79]],
    ['33767', ['maria', 'garcia'], 33, ['clothes', 'food', 'beauty'],
    [499, 189, 63]],
    ['33912', ['jose', 'martinez'], 22, ['sport', 'electronics', 'home'
    ], [259, 549, 109]],
    ['34009', ['lisa', 'wilson'], 35, ['home', 'books', 'clothes'],
    [329, 189, 329]],
    ['34278', ['james', 'lee'], 28, ['beauty', 'clothes', 'electronics'
    ], [189, 299, 579]],
]

# escribe tu código aquí

```

```

for user in users:
    if user[2] < 30:
        print(user)

```

```

['31980', ['kate', 'morgan'], 24, ['clothes', 'books'], [439, 390]]
['32761', ['samantha', 'smith'], 29, ['clothes', 'electronics', 'beauty'], [299, 679, 85]]
['33001', ['emily', 'brown'], 26, ['beauty', 'home', 'food'], [213, 659, 79]]
['33912', ['jose', 'martinez'], 22, ['sport', 'electronics', 'home'], [259, 549, 109]]
['34278', ['james', 'lee'], 28, ['beauty', 'clothes', 'electronics'], [189, 299, 579]]

```

## 12 Ejercicio 11

Juntemos las tareas 9 y 10 e imprimamos los nombres de los usuarios y las usuarias que tengan menos de 30 años y un gasto total superior a 1000 dólares.

```

[60]: users = [
    ['32415', ['mike', 'reed'], 32, ['electronics', 'sport', 'books'],
     [894, 213, 173]],
    ['31980', ['kate', 'morgan'], 24, ['clothes', 'books'], [439, 390]],
    ['32156', ['john', 'doe'], 37, ['electronics', 'home', 'food'],
     [459, 120, 99]],
    ['32761', ['samantha', 'smith'], 29, ['clothes', 'electronics', 'beauty'],
     [299, 679, 85]],
    ['32984', ['david', 'white'], 41, ['books', 'home', 'sport'], [234, 329, 243]],
    ['33001', ['emily', 'brown'], 26, ['beauty', 'home', 'food'], [213, 659, 79]],
    ['33767', ['maria', 'garcia'], 33, ['clothes', 'food', 'beauty'],
     [499, 189, 63]],
    ['33912', ['jose', 'martinez'], 22, ['sport', 'electronics', 'home'],
     [259, 549, 109]],
    ['34009', ['lisa', 'wilson'], 35, ['home', 'books', 'clothes'],
     [329, 189, 329]],
    ['34278', ['james', 'lee'], 28, ['beauty', 'clothes', 'electronics'],
     [189, 299, 579]],
]

for user in users:
    age = user[2]
    spendings_list = user[4]
    total_spendings = sum(spendings_list)

```

```

if age < 30 and total_spending > 1000:
    print(user)

# escribe tu código aquí

```

```

['32761', ['samantha', 'smith'], 29, ['clothes', 'electronics', 'beauty'], [299, 679, 85]]
['34278', ['james', 'lee'], 28, ['beauty', 'clothes', 'electronics'], [189, 299, 579]]

```

## 13 Ejercicio 12

Ahora vamos a mostrar el nombre y la edad de todos los usuarios y todas las usuarias que han comprado ropa. Imprime el nombre y la edad en la misma declaración print.

```

[64]: users = [
    ['32415', ['mike', 'reed'], 32, ['electronics', 'sport', 'books'],
     [894, 213, 173]],
    ['31980', ['kate', 'morgan'], 24, ['clothes', 'books'], [439, 390]],
    ['32156', ['john', 'doe'], 37, ['electronics', 'home', 'food'],
     [459, 120, 99]],
    ['32761', ['samantha', 'smith'], 29, ['clothes', 'electronics', 'beauty'],
     [299, 679, 85]],
    ['32984', ['david', 'white'], 41, ['books', 'home', 'sport'], [234, 329, 243]],
    ['33001', ['emily', 'brown'], 26, ['beauty', 'home', 'food'], [213, 659, 79]],
    ['33767', ['maria', 'garcia'], 33, ['clothes', 'food', 'beauty'],
     [499, 189, 63]],
    ['33912', ['jose', 'martinez'], 22, ['sport', 'electronics', 'home'],
     [259, 549, 109]],
    ['34009', ['lisa', 'wilson'], 35, ['home', 'books', 'clothes'],
     [329, 189, 329]],
    ['34278', ['james', 'lee'], 28, ['beauty', 'clothes', 'electronics'],
     [189, 299, 579]],
]

```

*#escribe tu código aquí*

```

for user in users:
    age = user[2]
    categories = user[3]
    if 'clothes' in categories:
        print(user)

```



```
['31980', ['kate', 'morgan'], 24, ['clothes', 'books'], [439, 390]]  
['32761', ['samantha', 'smith'], 29, ['clothes', 'electronics', 'beauty'], [299,  
679, 85]]  
['33767', ['maria', 'garcia'], 33, ['clothes', 'food', 'beauty'], [499, 189,  
63]]  
['34009', ['lisa', 'wilson'], 35, ['home', 'books', 'clothes'], [329, 189, 329]]  
['34278', ['james', 'lee'], 28, ['beauty', 'clothes', 'electronics'], [189, 299,  
579]]
```

#Escribe aquí cualquier comentario o reflexión final