**שאלה 1 סעף ב' 3**

**תכנון פתרון כאשר מקבלים נתונים ע"י stream(זרימה)**

**הפתרון שאני מציעה הוא כזה:**

ראשית, יש מנגנון שמקבל את הנתונים ברגע שהם מגיעים. לכל נתון שמגיע, אזהה לאיזו שעה הוא שייך בהתבסס על חותם הזמן שלו (כלומר, אם הנתון הגיע בשעה מסוימת, אסווג אותו לאותה שעה).

אשתמש בשני "מילונים" (או טבלה) כדי לשמור מידע לכל שעה:

**מילון אחד (sums):** שומר את הסכום המצטבר של כל הערכים שהתקבלו בשעה מסוימת.

**מילון שני (counts):** סופר כמה נתונים התקבלו באותה שעה.

לדוגמה, אם בשעה "2025-04-04 13" התקבלו ערכים שונים, המפתח במילונים יהיה השעה הזו והערך יהיה הסכום או הספירה של הנתונים עד כה.

בכל פעם שנתון חדש מגיע, אזהה את השעה אליה הוא שייך, אם כבר יש רשומה כזאת , אעדכן את הסכום והמונה. אם זו שעה חדשה, אצור רשומה חדשה המאותחלת בסכום לערך שהתקבל ובמונה ל-1.

לאחר עדכון הנתונים בכל שעה, אפשר לחשב בכל רגע נתון את הממוצע לכל שעה.

**יש לשים לב:**

* סנכרוןאם הנתונים מגיעים מכמה מקורות שונים- חותמות הזמן צריכות להיות מסונכרנות כך שהחלוקה לשעות תהיה נכונה ואחידה.
* במערכות בהן קיימת דרישה לדאטה קריטית, להוסיף מנגנוני גיבוי ושמירה על המידע כדי לא לאבד נתונים במקרה של תקלה.

פונקציה לדוגמא:

* def validate\_stream\_item(timestamp, value):  
   *"""  
   Checks the validity of a data stream item.  
   Args:  
   timestamp (str): String representation of the timestamp.  
   value (any): The value associated with the timestamp.  
    
   Returns:  
   bool: True if the item is valid, False otherwise.  
   """* try:  
   pd.to\_datetime(timestamp)  
   if not isinstance(value, (int, float)):  
   return False  
    
   return True  
    
   except ValueError as e:  
   raise ValueError(e)

def update\_hourly\_averages\_stream(stream):  
 *"""  
 Updates hourly averages in real-time from a data stream.  
 args:  
 stream (iterable): An iterable of tuples, where each tuple contains (timestamp, value).  
  
 returns:  
 None  
 """* sums = {}  
 counts = {}  
  
 for timestamp, value in stream:  
 if validate\_stream\_item(timestamp, value):  
 timestamp = pd.to\_datetime(timestamp)  
 hour = timestamp.floor('h')  
  
 if hour not in sums:  
 sums[hour] = 0  
 counts[hour] = 0  
  
 sums[hour] += value  
 counts[hour] += 1  
  
 average = sums[hour] / counts[hour]  
 print(f"שעה: {hour}, ממוצע: {average}")  
 else:  
 print(f"Skipping invalid item: timestamp='{timestamp }', value='{value}'")