```
import pandas as pd
from datetime import datetime, timedelta


def hours_between(start_time, end_time):
    return (end_time - start_time).total_seconds() / 3600
def has_consecutive_days(records):
    consecutive_count = 0
    for i in range(1, len(records)):
        current_date = records.iloc[i]['Pay Cycle End Date']
        prev_date = records.iloc[i - 1]['Pay Cycle End Date']
        if (current_date - prev_date).days == 1:
            consecutive_count += 1
        else:
            consecutive_count = 0
        if consecutive_count == 6:
            return True
    return False


def analyze_employee_records(file_path):
    df = pd.read_excel(file_path)
df['Pay Cycle End Date'] = pd.to_datetime(df['Pay Cycle End Date'])


    unique_employees = df['Employee Name'].unique()
    for employee_name in unique_employees:
        employee_records = df[df['Employee Name'] == employee_name].sort_values(by='Pay Cycle End Date')
        time_between_shifts = employee_records['Time Out'].shift(-1) - employee_records['Time']
        if ((time_between_shifts > pd.Timedelta(hours=1)) & (time_between_shifts < pd.Timedelta(hours=10))).any():
            print(f"{employee_name} has less than 10 hours between shifts")


        shift_duration = employee_records['Time Out'] - employee_records['Time']
        if (shift_duration > pd.Timedelta(hours=14)).any():
            print(f"{employee_name} has worked for more than 14 hours in a single shift")


        if has_consecutive_days(employee_records):
            print(f"{employee_name} has worked for 7 consecutive days")


file_path = 'Assignment_Timecard.XLSV'
analyze_employee_records(file_path)
```