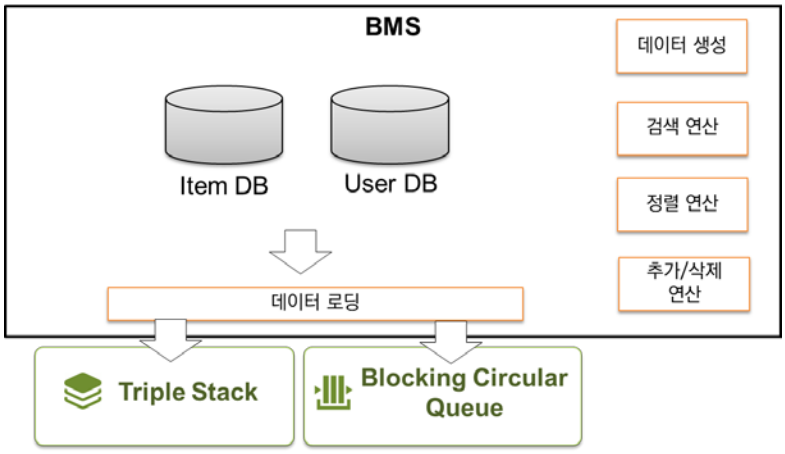
H/W 3: Advanced Stack and Queue

목표: H/W 1과 2를 통해 개발된 BMS의 시스템 위에서, 빠른 서비스를 제공하기 위한, 첨단 스택과 큐를 개발하시오. 첨부된 예제 코드 참고 *(5월 31일까지)*



1. BMS는 데이터 로딩을 통해 Triple Stack(TS)과 Blocking Circular Queue(BCQ)에 데이터를 지속적으로 추가한다. 코드 1은 TS에 대한 데이터 로딩 연산이고, 코드 2는 BCQ에 대한 데이터 로딩 연산이다. get_users(N)와 get_items(N)의 메소드를 확장 구현하시오. (20점)

코드 1. TS에 대한 데이터 로딩 연산

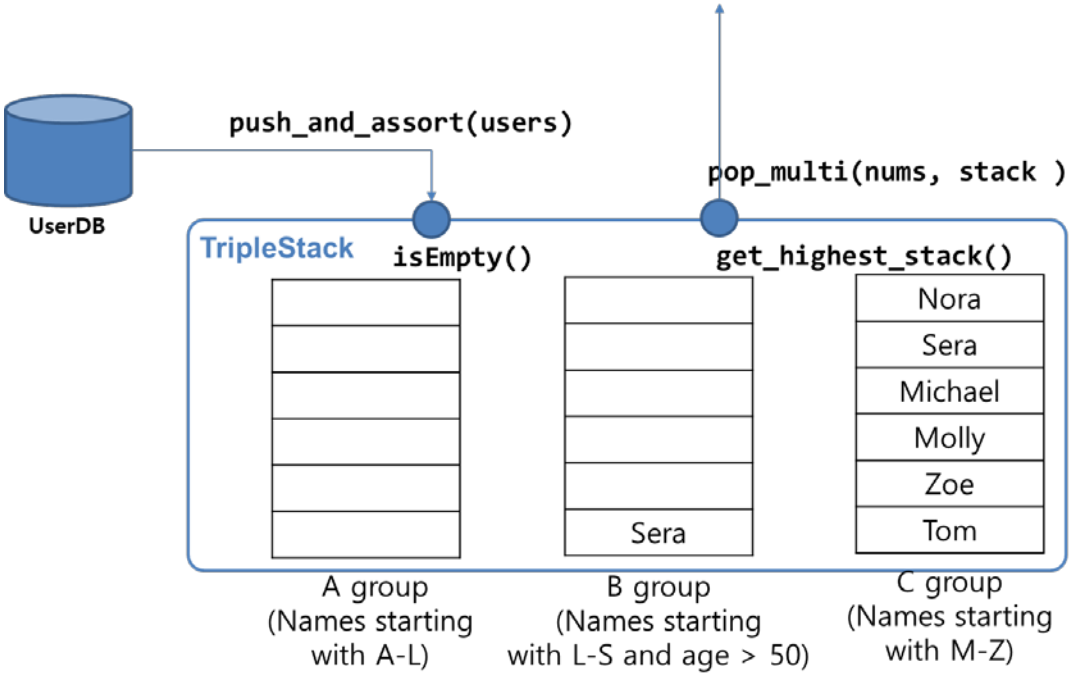
```
# TripleStack
ts = TripleStack(100)
while not ts.isEmpty():
    users = bms.get_users(100)
    ts.push_and_assort(users)
    no_stack = ts.get_highest_stack()
    ts.pop_multi(100, no_stack)
```

코드 2. BCQ 에 대한 데이터 로딩 연산

```
# Bounded Circular Queue
bcq = CustomBCQ(100)
users = bms.get_items(10000)
stages = 0
for i, user in enumerate(users):
    n_dequeued, n_enqueued = 0, 0
    while not bcq.isFull():
        bcq.enqueue(user)
        n_enqueued += 1

    while not bcq.isEmpty():
        bcq.dequeue()
        n_dequeued += 1

    print(f'stage {stages}: enqueue된 작업의 수 {n_enqueued}  dequeue된 작업의 수 {n_dequeued}')

```

코드 1과 2가 제대로 실행되기 위해서는 TS와 BCQ에 대한 class를 각각 구현해야 한다.

2. TS은 다른 유형의 데이터를 보관하는 세 개의 스택(A-C)으로 구성된다. 구체적으로, 각 스택은 사용자의 이름과 나이에 따라서 다른 데이터들을 보관한다. (40점)



3. BCQ는 enqueue 작업을 일시적으로 보류하는 기능(Blocking)을 가진 순환 큐(Circular Queue)이다. 구체적으로, BCQ는 최대 적재용량(maximum capacity)를 가지며, 리스트 기반의 순환 큐로 구성된다. 또한, BCQ는 Full 상태가 되면 25%까지 dequeue만 가능하고, 25%미미만일 때 enqueue/dequeue가 모두 가능해진다. (40점)