DB Connection Pool

DB Connection

DB에 연결하기 위해서는 App Server와 DB의 연결이 필요

Java는 JDBC를 통해서 연결 Next는 mysql 모듈로 연결

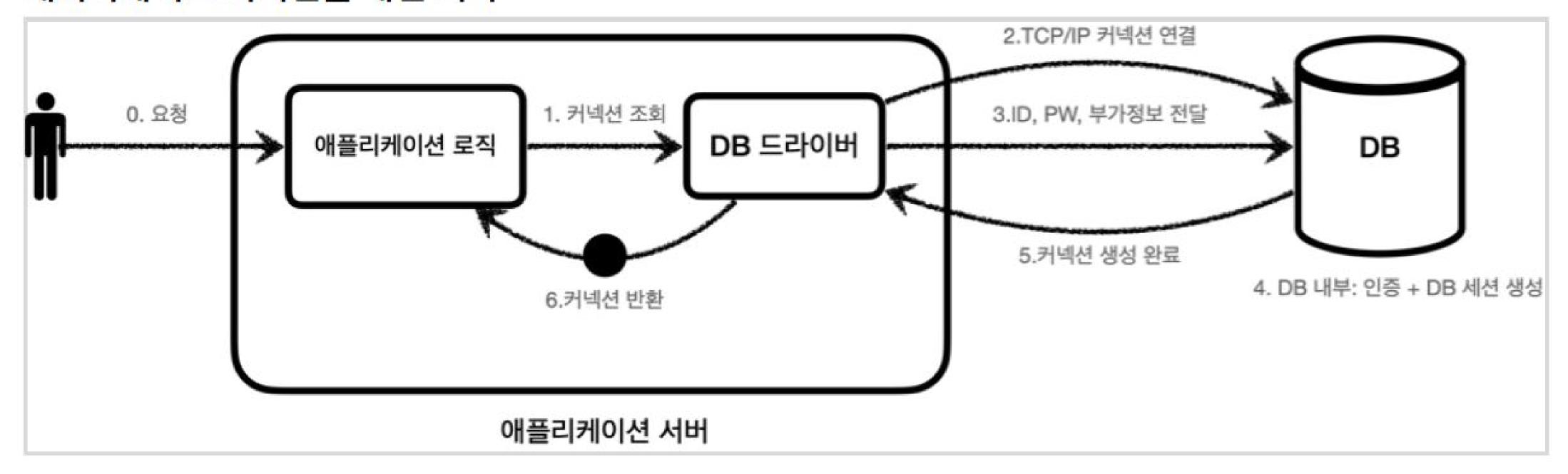
DB Connection

- 2Tier
 - 자바 프로그램(ex: JSP)이 직접 데이터베이스 서버에 접근

- 3Tier
 - 자바 프로그램과 데이터베이스 서버 사이에 미들웨어 계층이 존재

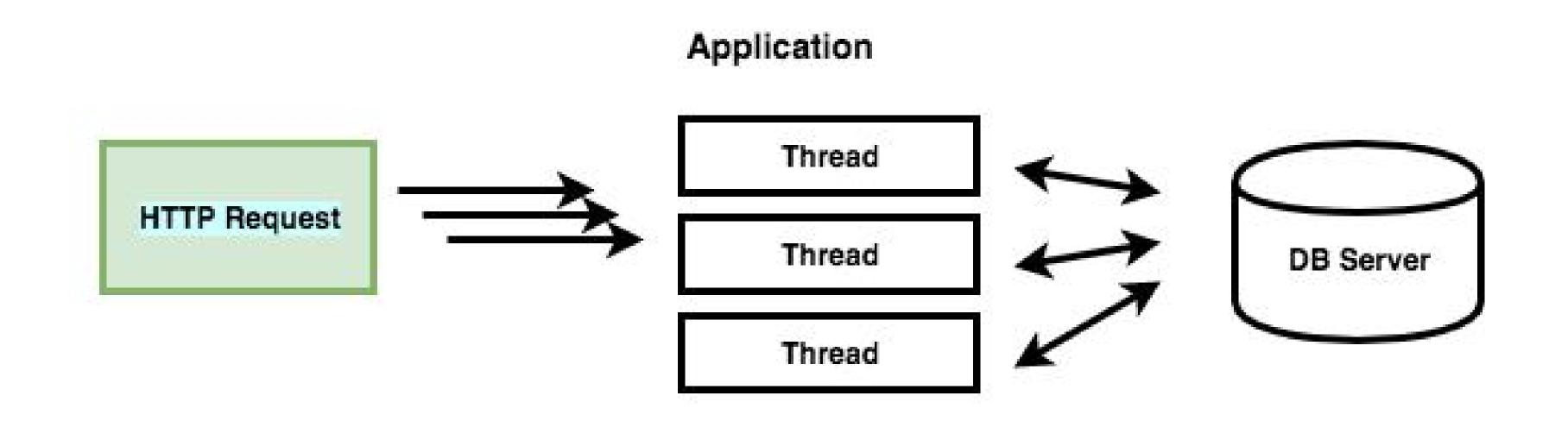
일반적인 DB 연결

데이터베이스 커넥션을 매번 획득



요청 발생 시 Connection을 생성해 연결

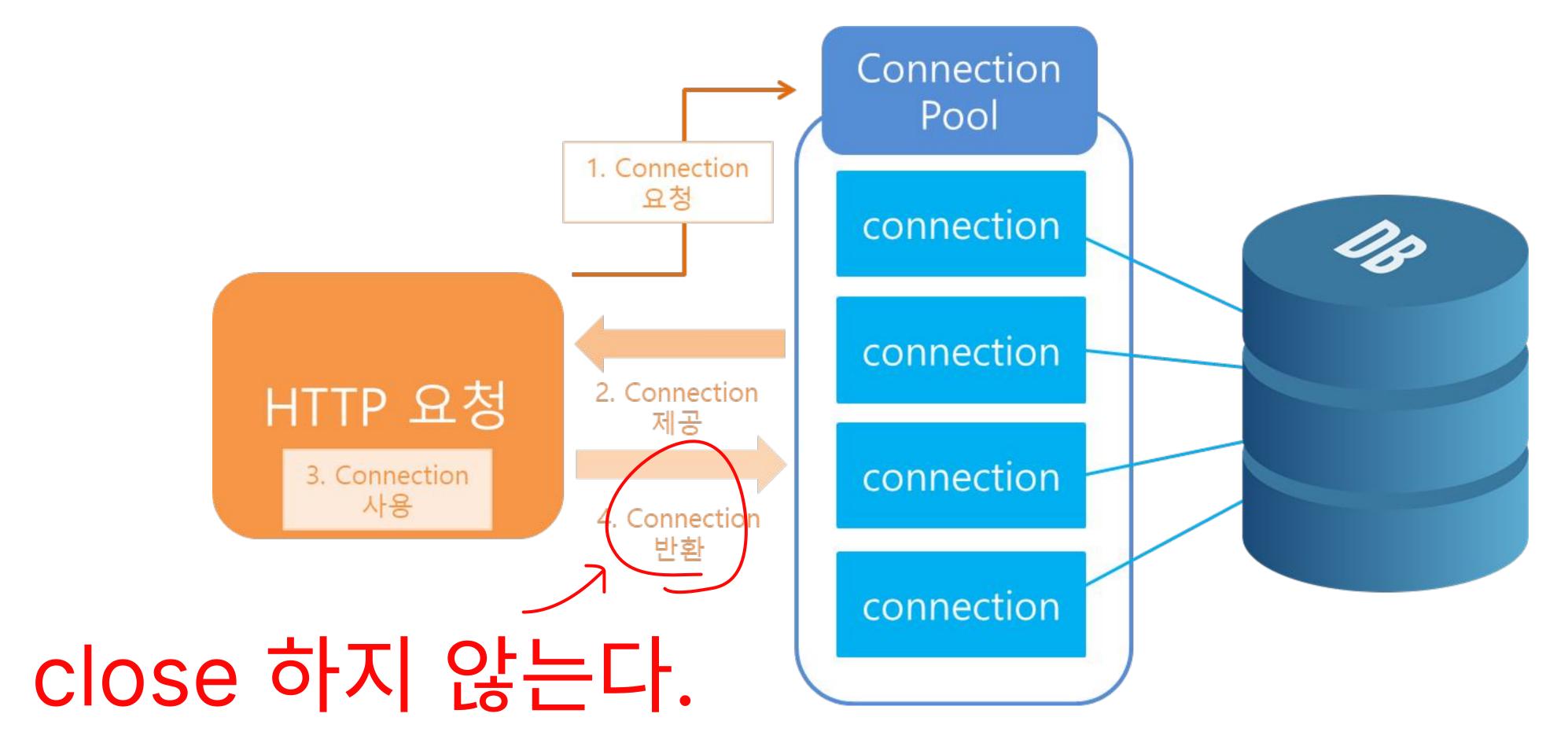
일반적인 DB 연결



요청 시 마다 3wh가 이루어진다? 작업 완료 시 마다 close가 이루어진다?

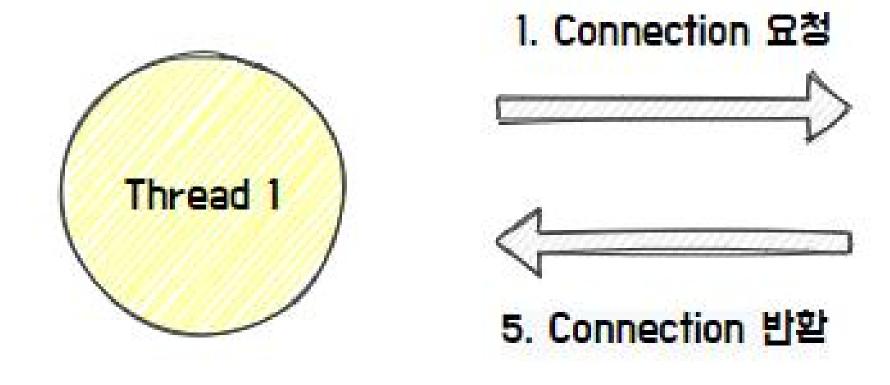
⇒ 비용 소모, 비효율

Connection Pool?



min :max 개수 설정 가능

Connection Pool 동작과정



Connection Pool

Connection 1

Connection 3

Connection 2

Connection 4

Connection N

왜 ? Why?

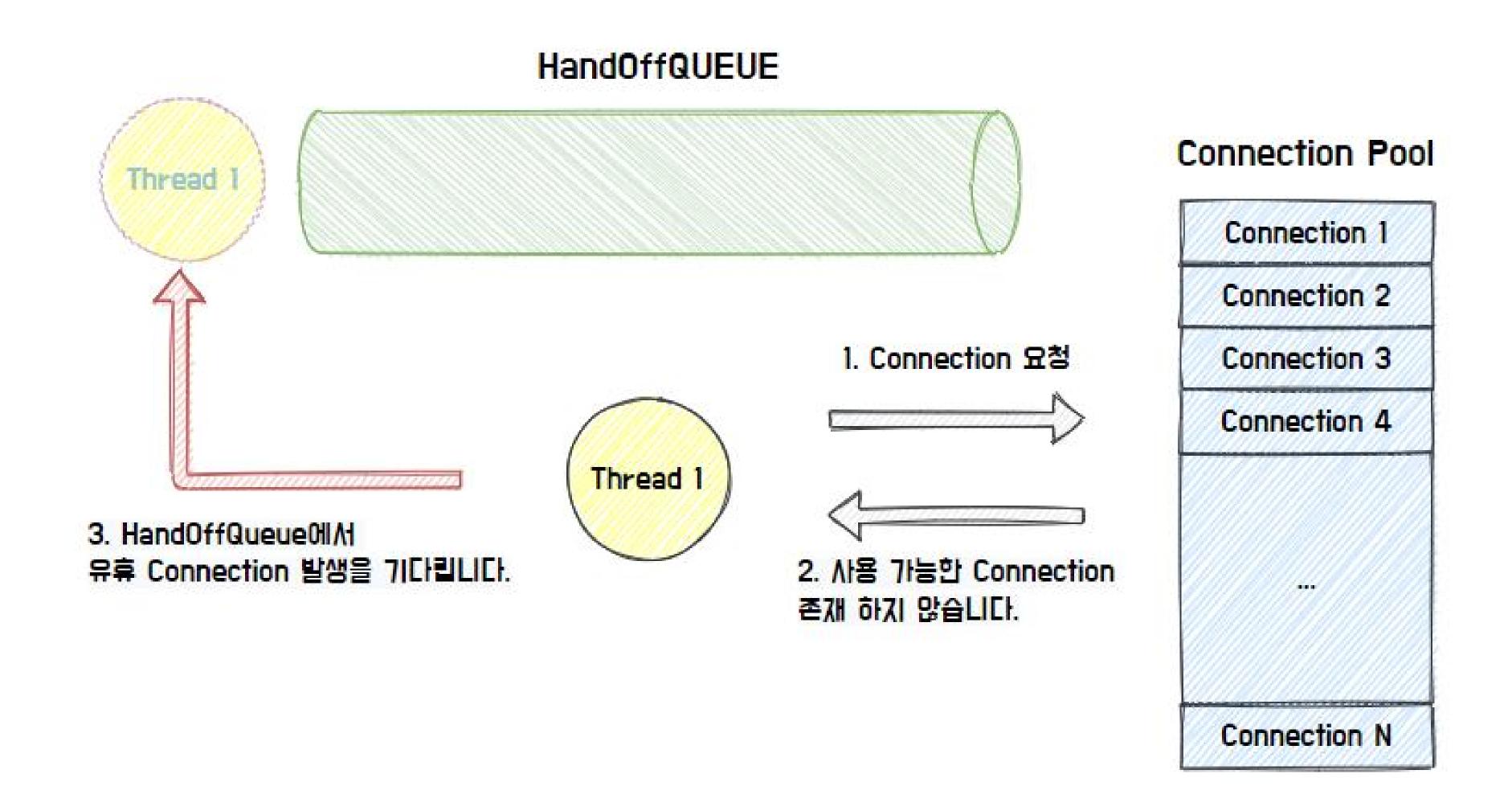
2. 이전 사용했던 Connection 정보 존재 여부 확인

3. 이전 사용했던 Connection 목록 중 사용 가능한 Connection 존재 여부 확인

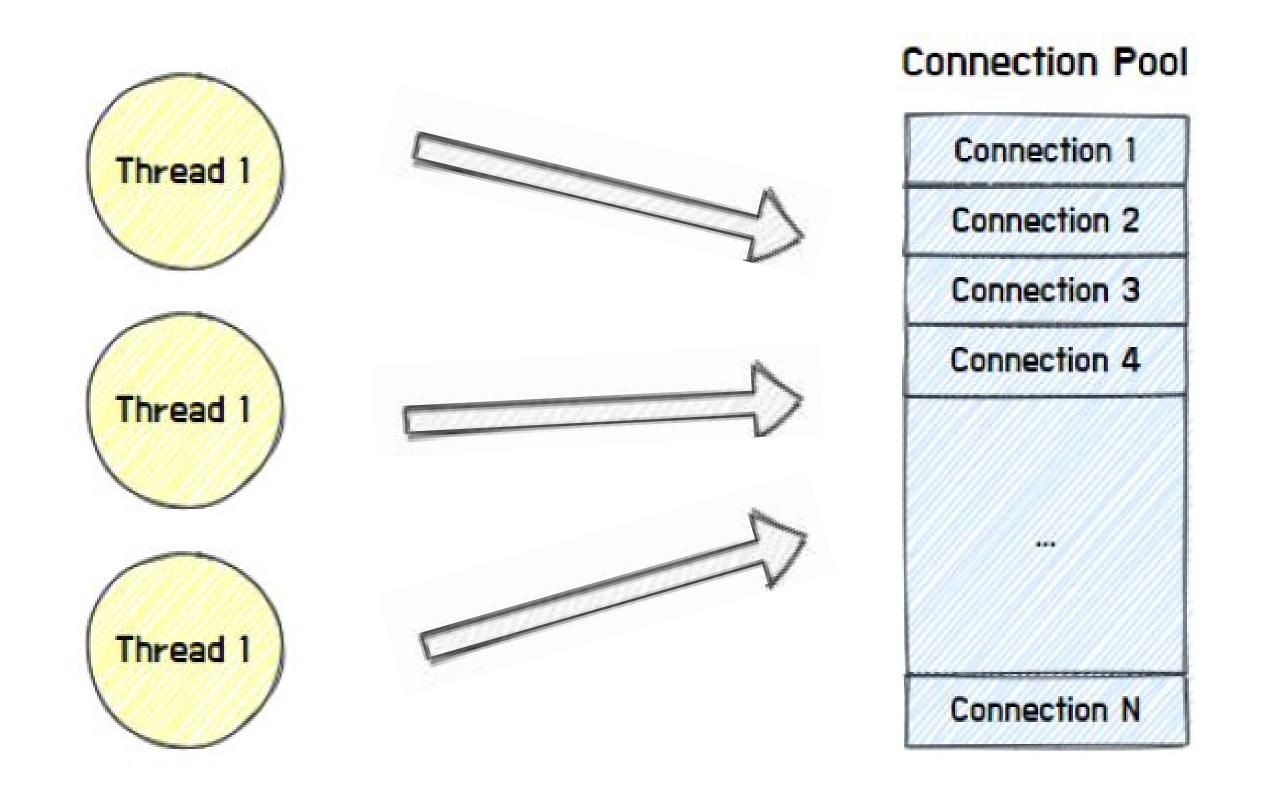
4. 전체 Connection 목록 중 사용 가능한 Connection 존재 여부 확인

(2, 3, 4 순서대로 유휴 Connection() 존 재하면 다음 과정을 생략하고 반환합니다.)

Connection Pool 동작과정

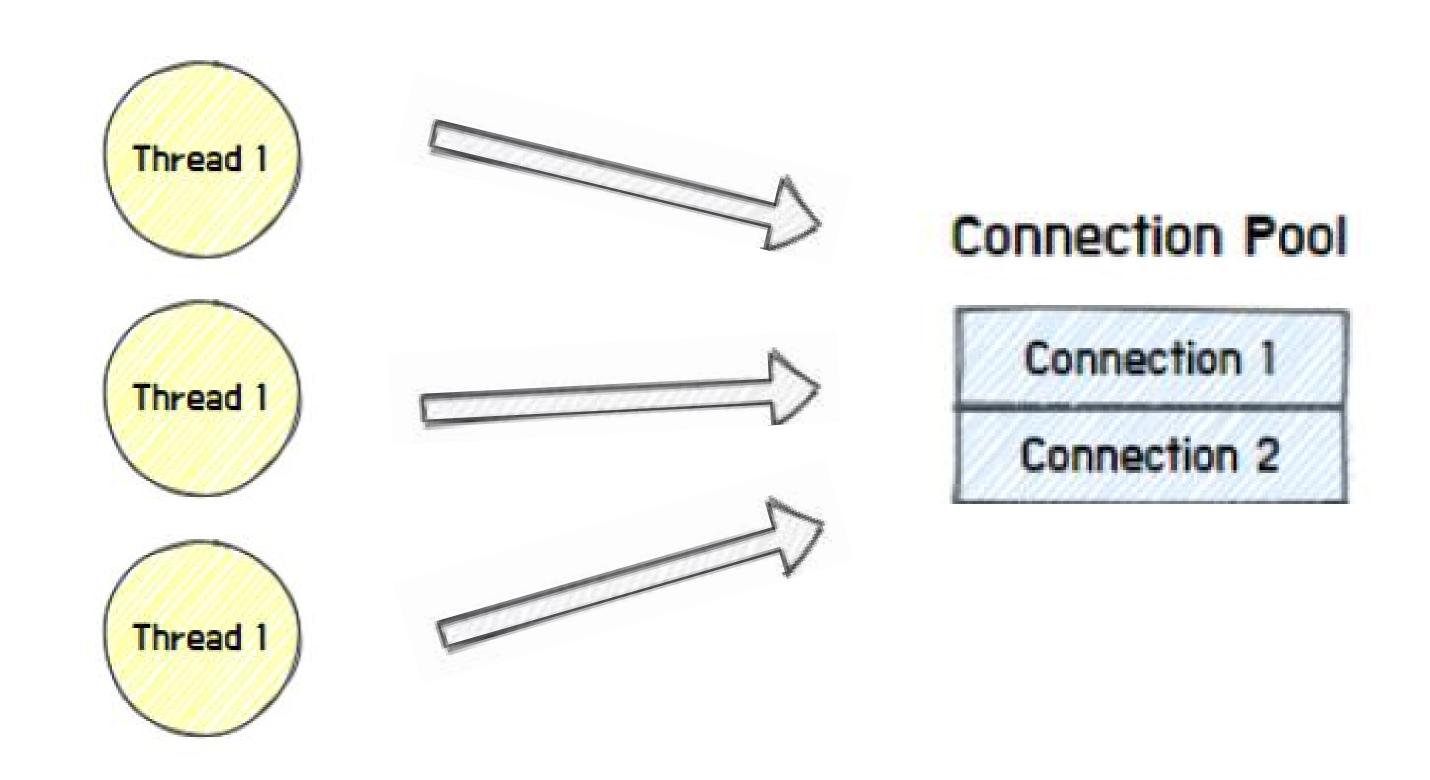


Thread와 Connection 개수의 상관관계



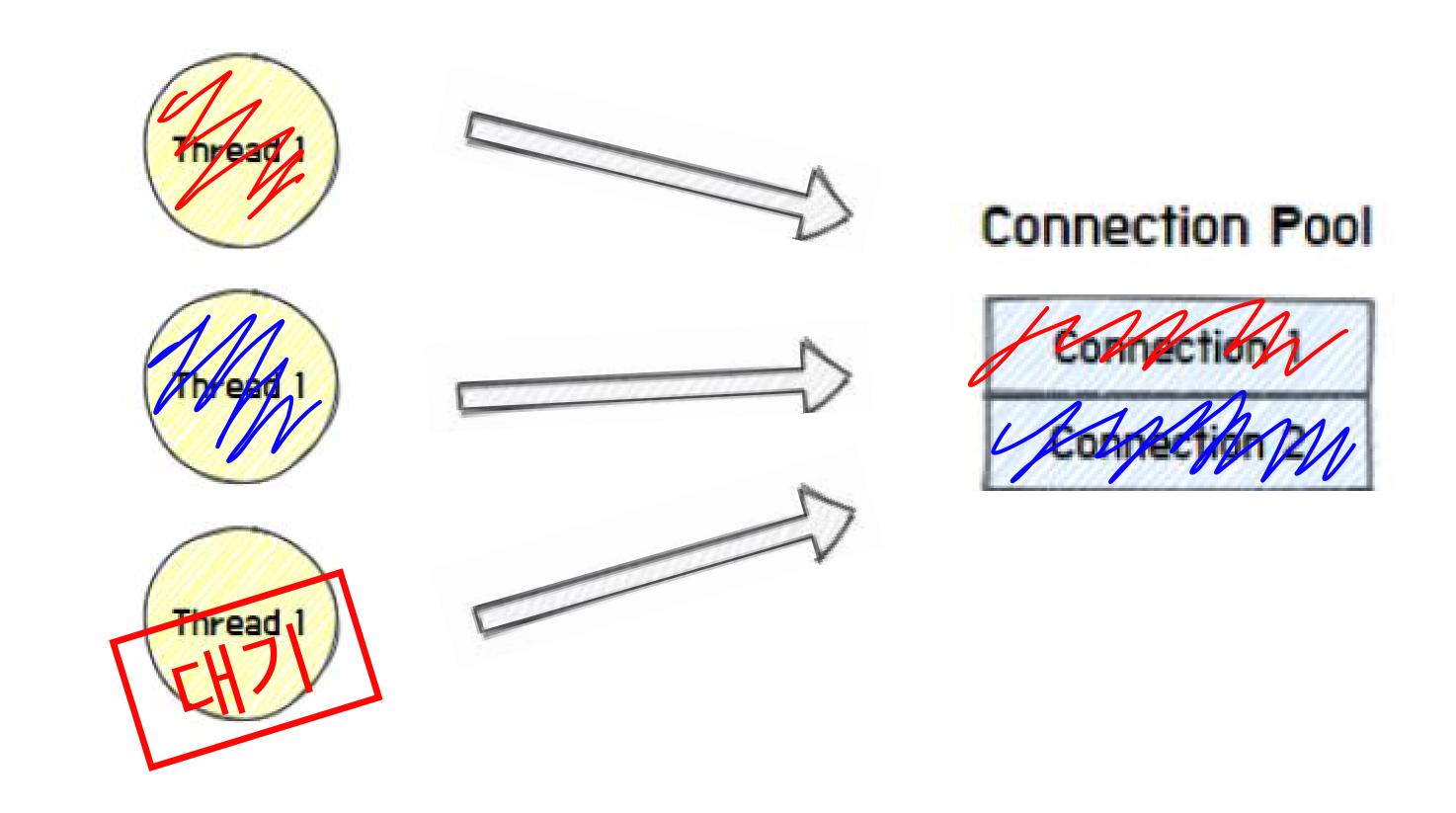
Connection pool 뭐가 문제일까?

Thread와 Connection 개수의 상관관계



Connection pool 뭐가, 문제일까?

Thread와 Connection 개수의 상관관계



필요 Connection이 1개라고 가정

ex_) Thr 8개, Task 필요 Connection 2개

$$8*(2-1)+1=97$$

Dead Lock을 피하기 위해

| Thread-1 | Thread-2 | Thread-3 | Thread-4 | Thread-5 | Thread-6 | Thread-7 | Thread-8 |
|------------|------------|------------|------------|------------|------------|------------|------------|
| Connection |
| | Connection | | | | | | |

replica

block

절대적인 것은 없다.

Context Switching

Time out