

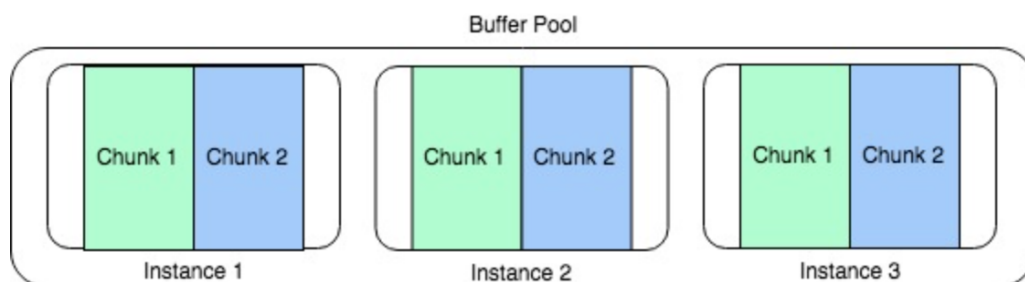
InnoDB Buffer Pool Optimization

목적: 이 강의에서는 InnoDB Buffer Pool의 설정 값을 변경하여 MySQL의 성능을 최적화하는 방법을 배웁니다.

포인트: Buffer Pool의 역할과 관련 설정 변수들을 이용한 최적화 기법을 이해하는게 중요합니다.

▼ Understanding InnoDB Buffer Pool

역할과 중요성: 테이블 데이터와 인덱스 데이터를 메모리에서 캐싱하고, 변경된 데이터를 디스크에 플러시하는 역할.



<https://www.percona.com/blog/chunk-change-innodb-buffer-pool-resizing/>

- **Buffer Pool 은 Instance 로 나뉘져있음:**
 - Instance 단위로 Flush List 와 LRU 리스트가 나뉘져있음.
 - Instance 의 수가 많을수록 동시성 수준의 성능 증가
- **Chunk 는 Buffer Pool 이 resizing 될 때 비용을 최소화 하기 위해서 설계됨:**

▼ Buffer Pool Optimization Strategies

1. `innodb_buffer_pool_size` 조절하기:

- `innodb_buffer_pool_size` 는 OS 메모리의 70-80% 정도로 할당해주는 것이 좋음.
- 메모리 크기가 8GB 미만이라면 OS 메모리의 50%만 쓰는 것을 권장
- 기본 값: 128MB

2. Buffer Pool Instances 조절하기

- `innodb_buffer_pool_size` 사이즈에 맞게 `innodb_buffer_pool_instance` 의 수도 조절 해서 동시성 성능 향상
- 메모리가 40GB 밑이라면 `innodb_buffer_pool_instance` 는 기본 값으로 사용. (기본 값: 8)
- 40GB 보다 많아진다면 5GB 마다 `innodb_buffer_pool_instance` 의 수를 하나씩 증가

3. Read-Ahead 트리거링 조절:

- **Read-Ahead 전략:** 테이블이나 인덱스의 연속된 데이터 블록을 미리 비동기로 가져오는 최적화 방식.
- `innodb_read_ahead_threshold` **설정:** 이 값보다 많은 연속된 페이지를 읽는다면 Read-Ahead 가 활성화 된다. (기본 값: 56)
- `innodb_random_read_ahead` **설정:** Buffer Pool 에 이 값보다 많은 연속된 페이지가 있다면 Read-Ahead 가 활성화 된다. (기본 값: 비활성화)