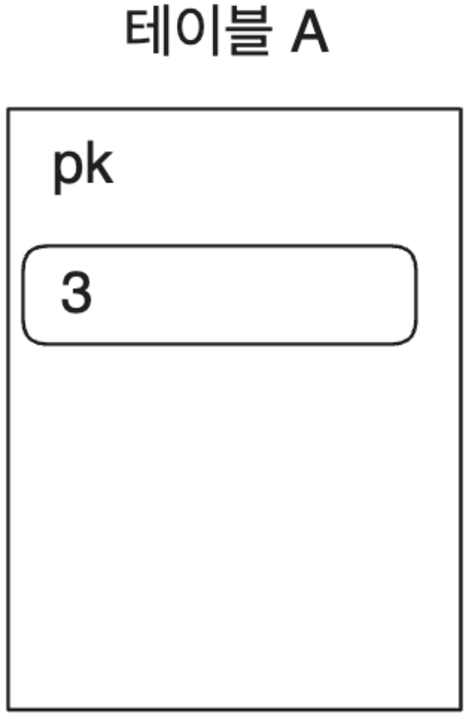
MySql의
트랜잭션, 락, 인덱스, 격리수준

면접을 대비하며 공부했던 내용 정리

락

트랜잭션에서 하나의 쿼리만 사용하면 필요 없나?

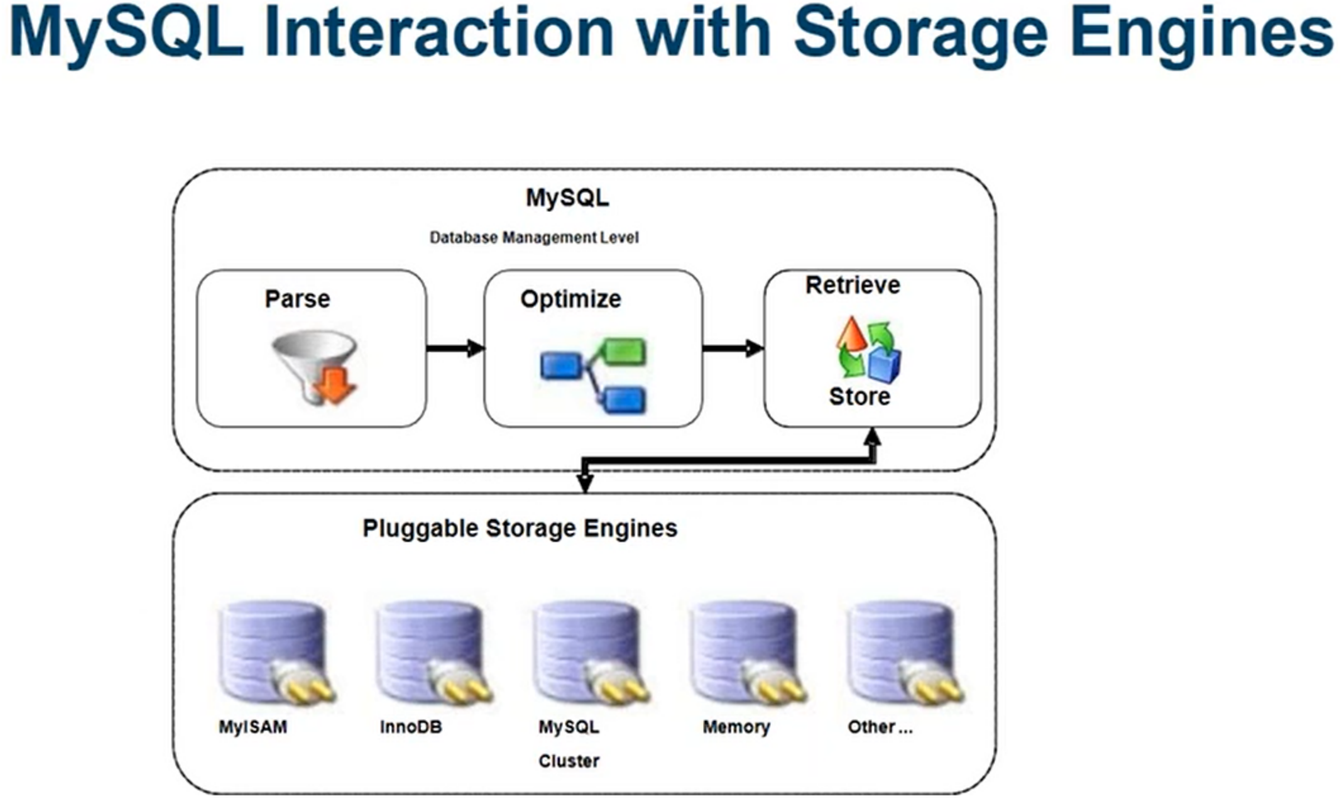
- 트랜잭션을 가진 경우, 모두 실패
- 트랜잭션 없이 실행할 경우, 2,3만 삽입




```
INSERT INTO 테이블 A
VALUES (1), (2), (3);
```

락

- MySql은 특이하게
  - MySql 수준 락과
  - 스토리지 엔진 락으로 나뉨



락

- MySql은 특이하게
  - MySql 수준 락과
  - 스토리지 엔진 락으로 나뉨

- MySql 락
  - 글로벌 락
  - 테이블 락
  - 네임드 락
  - 메타데이터 락
- InnoDB 스토리지 엔진 락
  - 레코드 락
  - 갭 락
  - 넥스트 키 락
  - 자동 증가 락

MySql 수준 락

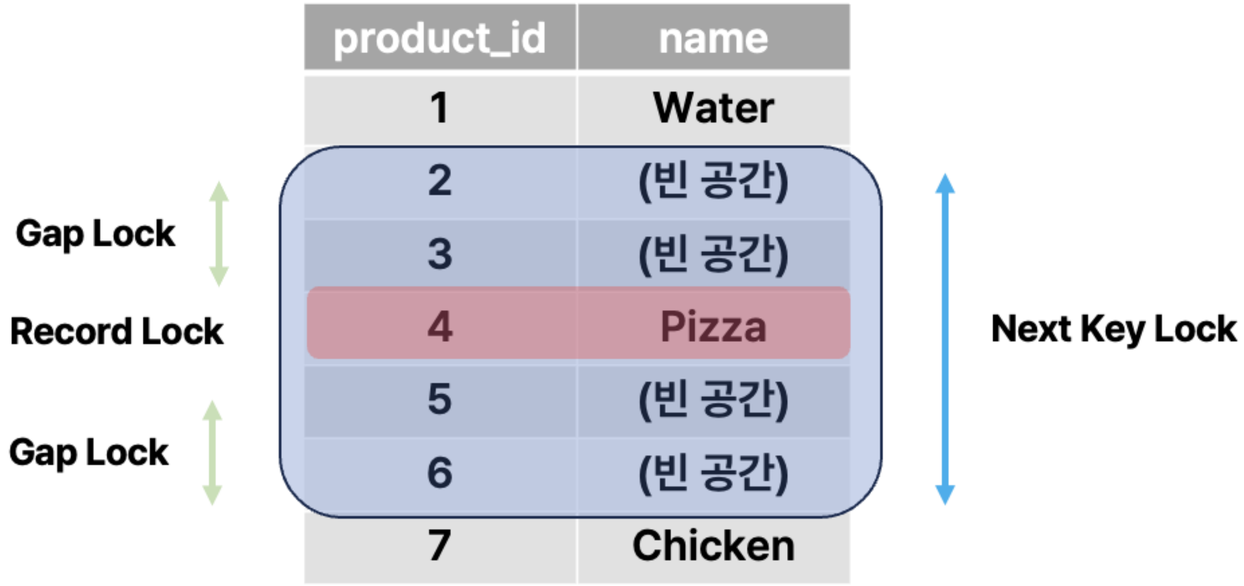
- 글로벌 락
- 테이블 락
- 네임드 락
- 메타데이터 락

```
-- 잠금 획득을 최대 2초간 대기한다.
SELECT GET_LOCK("mylock", 2);

-- 잠금 설정 확인
SELECT IS_FREE_LOCK("mylock");

-- 락 해제
SELECT RELEASE_LOCK("mylock");

-- 모든 함수에서 락을 획득하거나 해제하면 1을, 아니면 NULL이나 0 반환
```

InnoDB 스토리지 엔진 잠금

InnoDB에서는 인덱스에 락을 건다.

- 인덱스에 락을 건다는 것이 매우 중요한 특징



인덱스에 락을 거는게 뭐 어때서?

인덱스 설계가 매우 중요

```
-- 인덱스는 first_name 컬럼에만 걸려있다.

SELECT COUNT(*) FROM employees WHERE first_name="Tom";
-- 256
SELECT COUNT(*) FROM employees WHERE last_name="Kim";
-- 1
```

```
UPDATE employees SET hire_date=NOW() WHERE first_name="Tom" AND last_name="Kim";
```

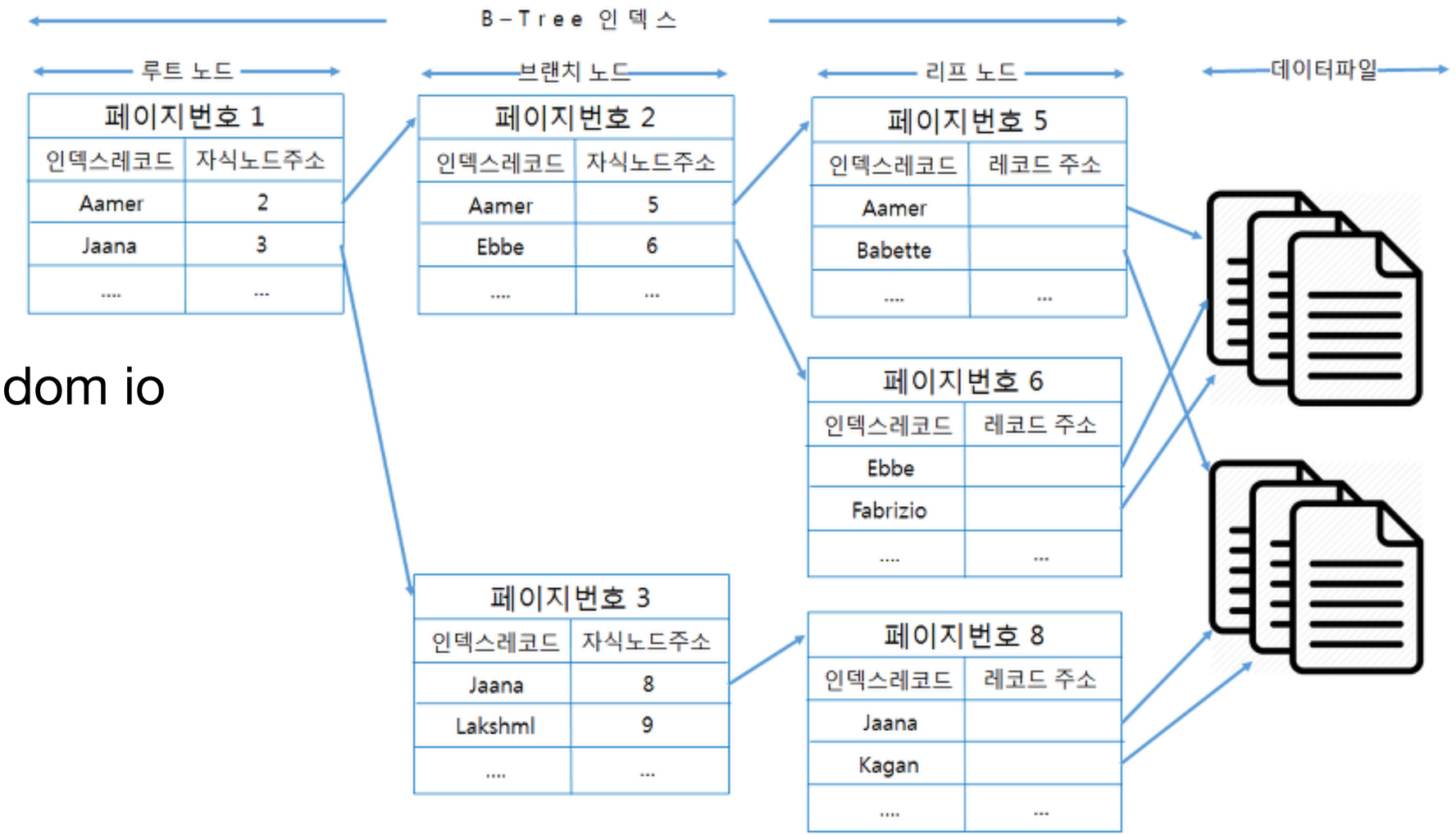
256개의 컬럼에 모두 락이 걸린다.

만약 인덱스가 없다면?

인덱스 - B-Tree

B-Tree

- balanced tree
- 정렬되어 있다.
- 하나 진행시 마다 random io

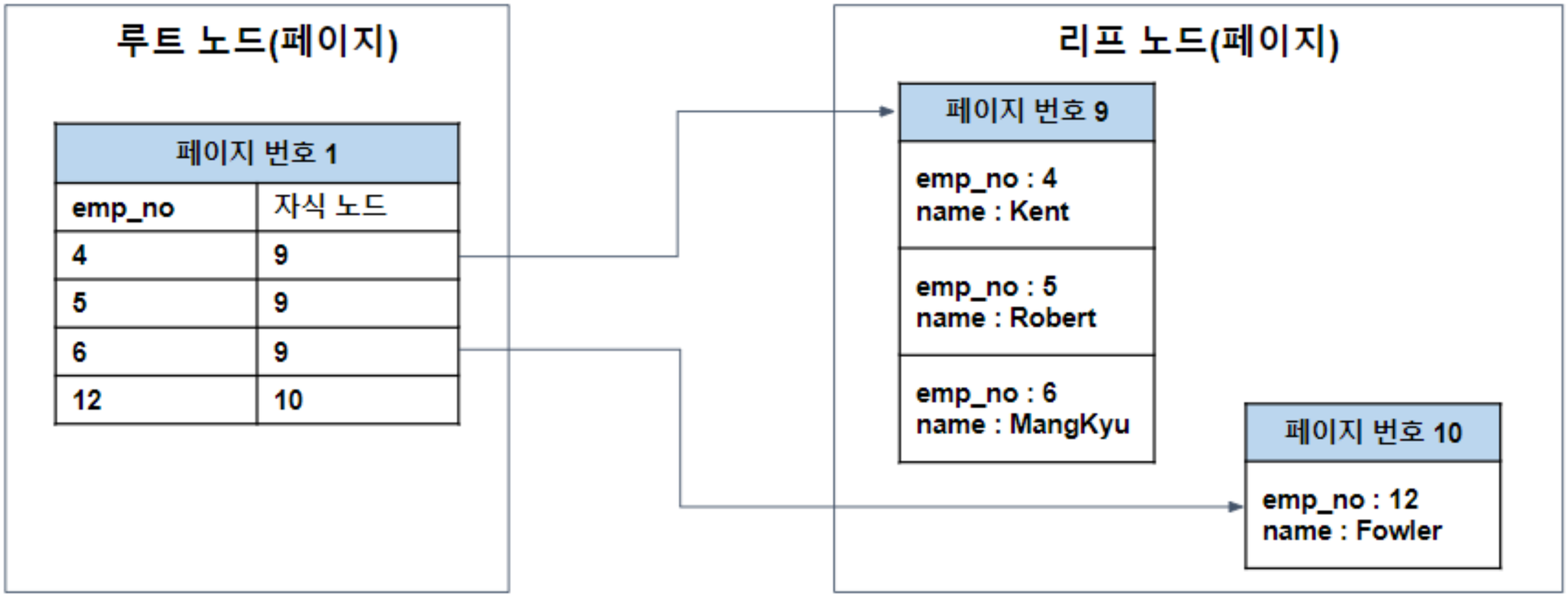


인덱스가 걸려도 풀스캔?

- 20%가 넘어가면 옵티마이저는 테이블 풀 스캔을 한다.
- 카디널리티를 계산해서, 예상 레코드 수를 책정
- 예를 들어 1000개의 레코드, 10개의 유니크 값

인덱스 - 클러스터링 인덱스

클러스터링 인덱스


루트 노드(페이지)
페이지 번호 1
emp_no
자식 노드
4
9
5
9
6
9
12
10
리프 노드(페이지)
페이지 번호 9
emp_no : 4
name : Kent
emp_no : 5
name : Robert
emp_no : 6
name : MangKyu
페이지 번호 10
emp_no : 12
name : Fowler

인덱스 값과 트리 자식 수

운영체제 시간에 배운것과 비슷하다.

- 자식수가 줄어들면, depth가 깊어진다.

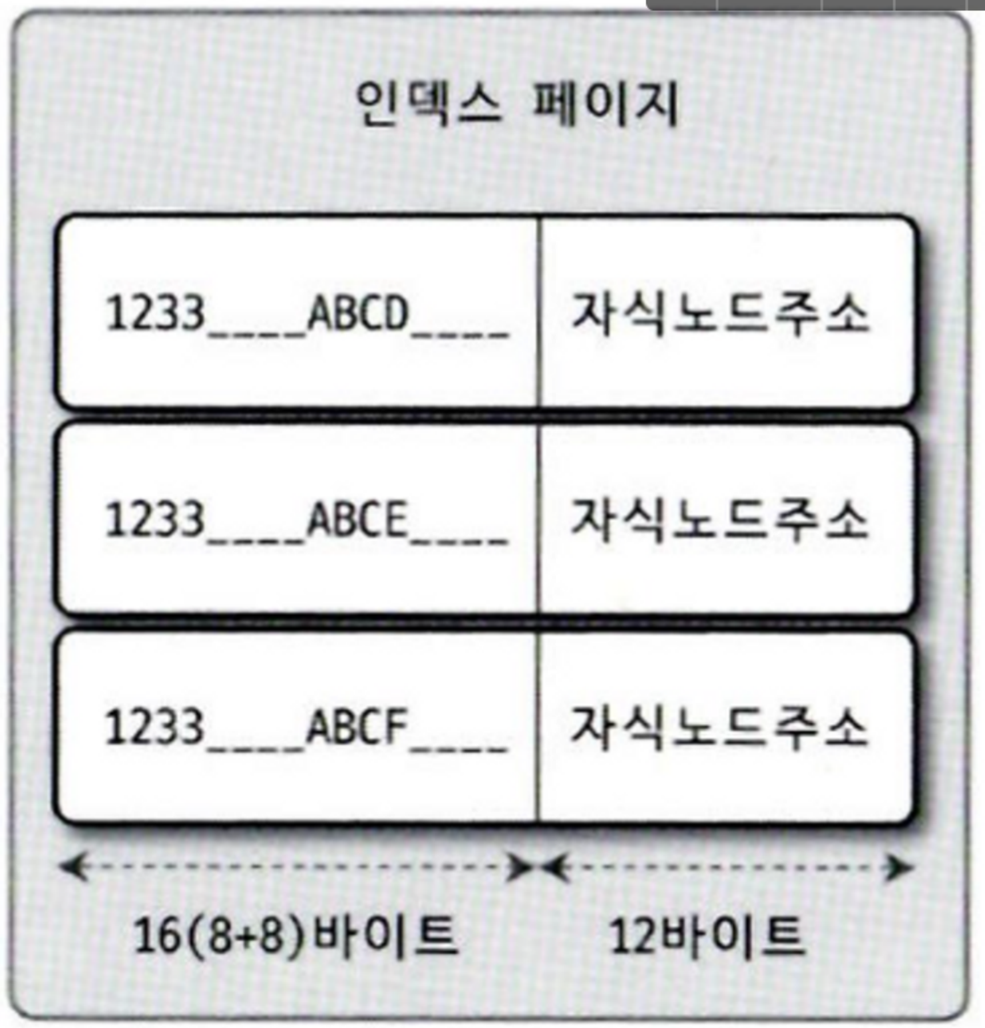


그림 8.7 인덱스 페이지의 구성

인덱스 스캔 방식

다양한 스캔 방식

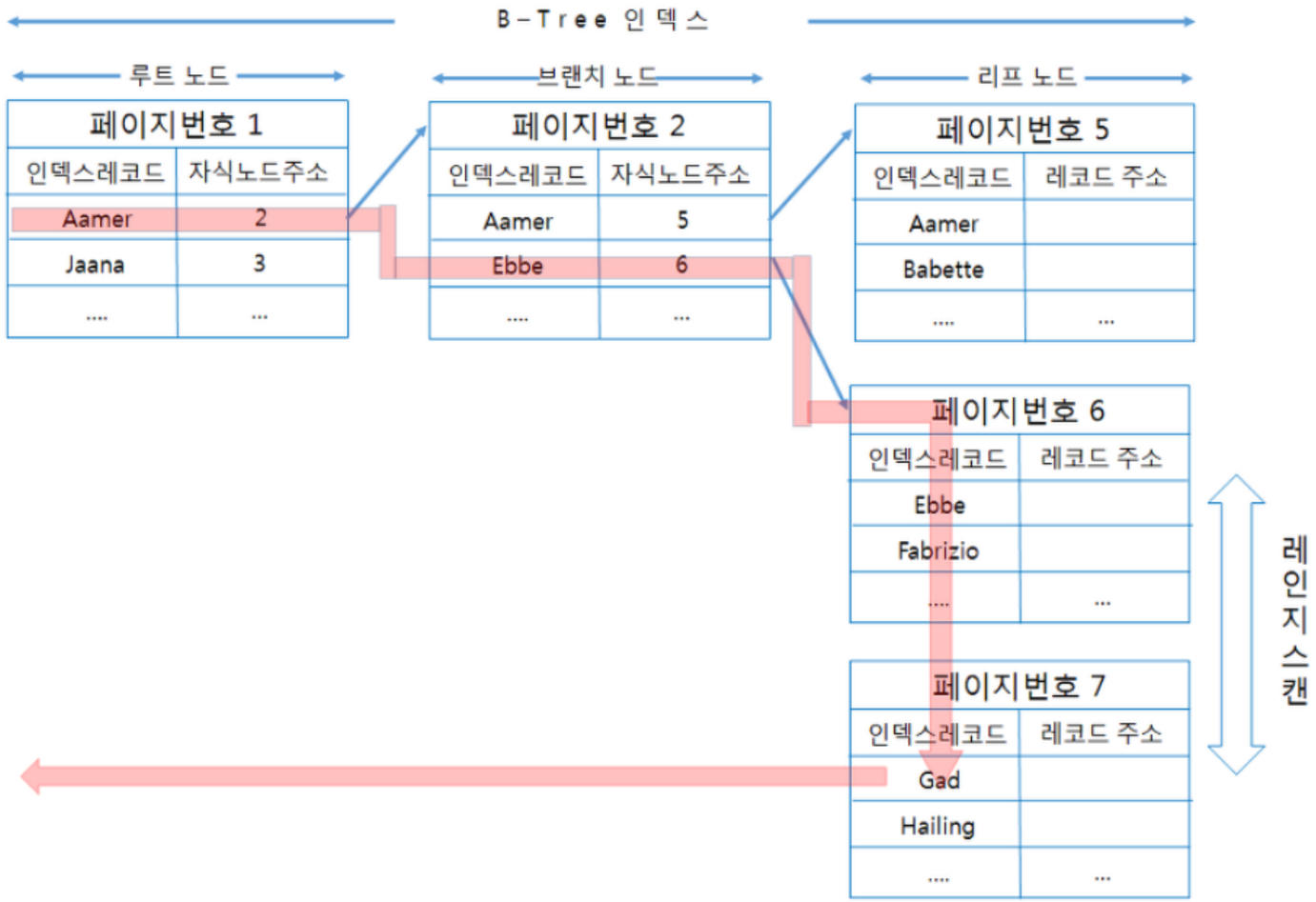
옵티마이저에 의해 결정

- 인덱스 레인지 스캔
- 인덱스 풀 스캔
- 루스 인덱스 스캔
- 인덱스 스킵 스캔

인덱스 레인지 스캔

매우 효율적

- 시작점 부터 쭉
- 연결 리스트
- 커버링 인덱스



인덱스 풀 스캔

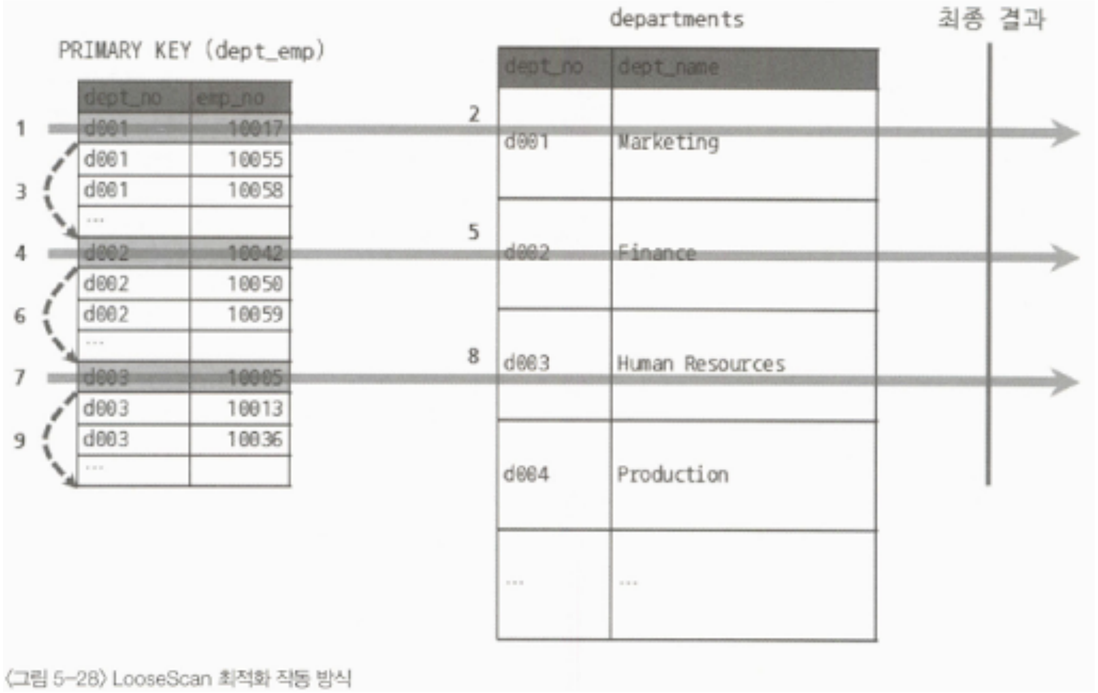
- (A, B, C) 순서 인데 (B, C)로 쿼리
- 인덱스 값 만으로 처리 가능

루즈 인덱스 스캔

- Group by, min, max
- 인덱스 값 만으로 연산이 될 때

예를 들어 (A,B) 순서로 인덱스가 걸려 있는데

```
SELECT A, min(B)
FROM table
WHERE A BETWEEN 'd002' AND 'd004'
GROUP BY A;
```



〈그림 5-28〉 LooseScan 최적화 작동 방식

인덱스 스킵 스캔

- 인덱스 있는 컬럼 만으로 처리
- (A,B,C) 인덱스 (B,C) 쿼리
- A의 카디널리티 낮음

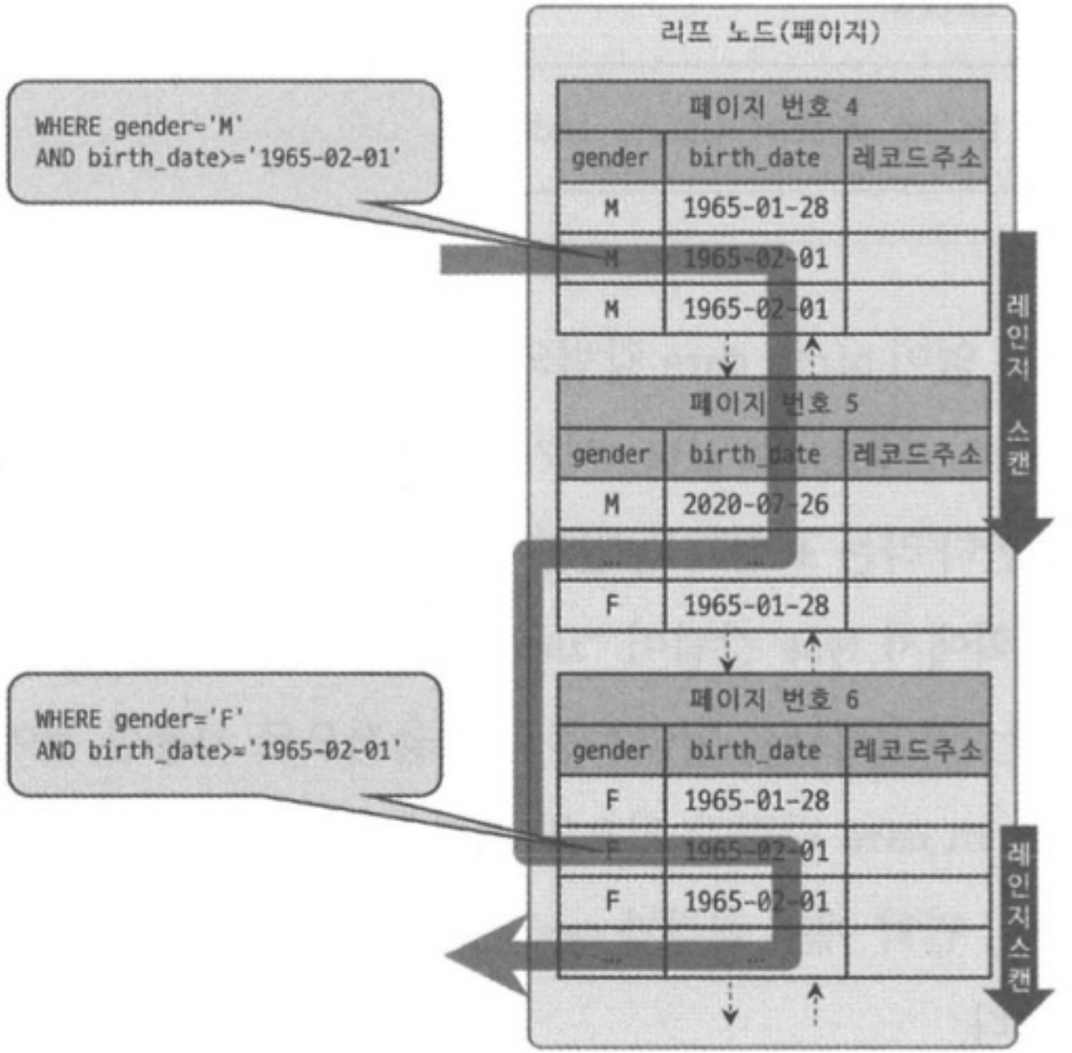


그림 8.12 인덱스 스킵 스캔

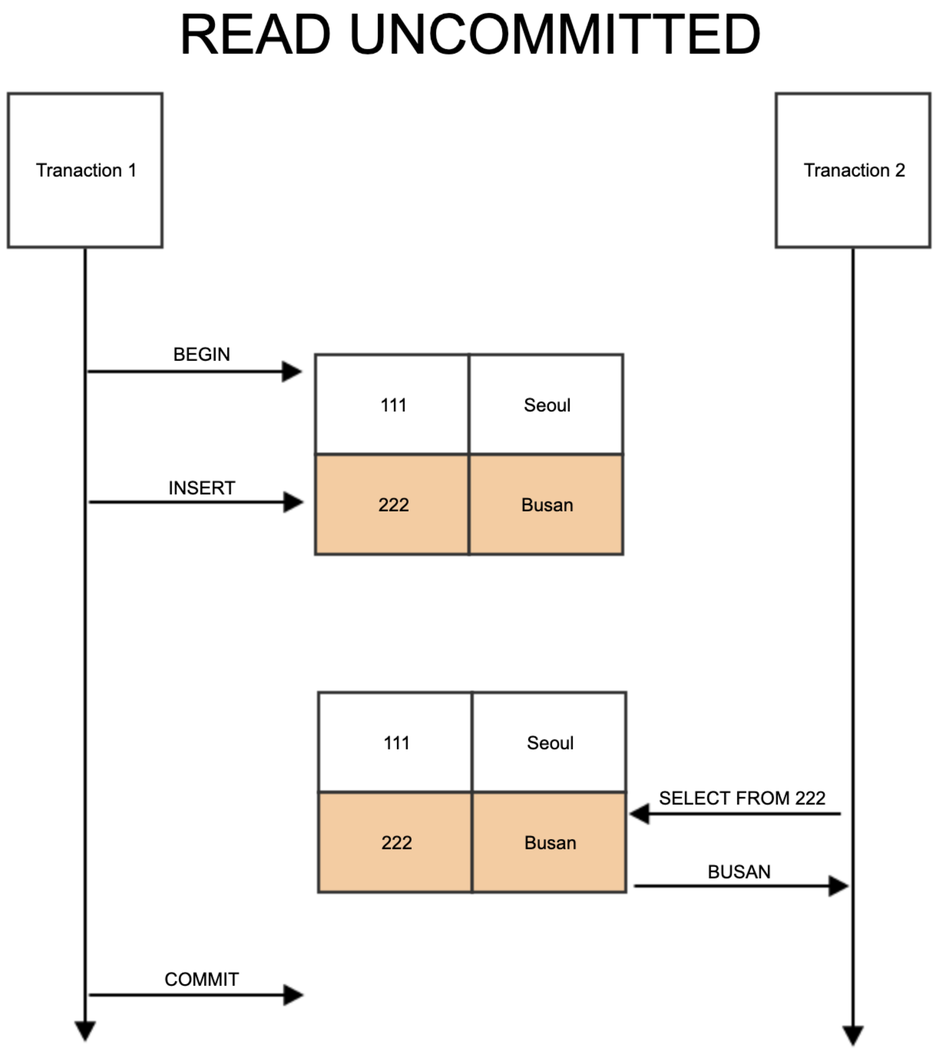
격리 수준

트랜잭션 격리 수준

- Read Uncommitted
- Read Committed
- Repeatable Read
- Serializable

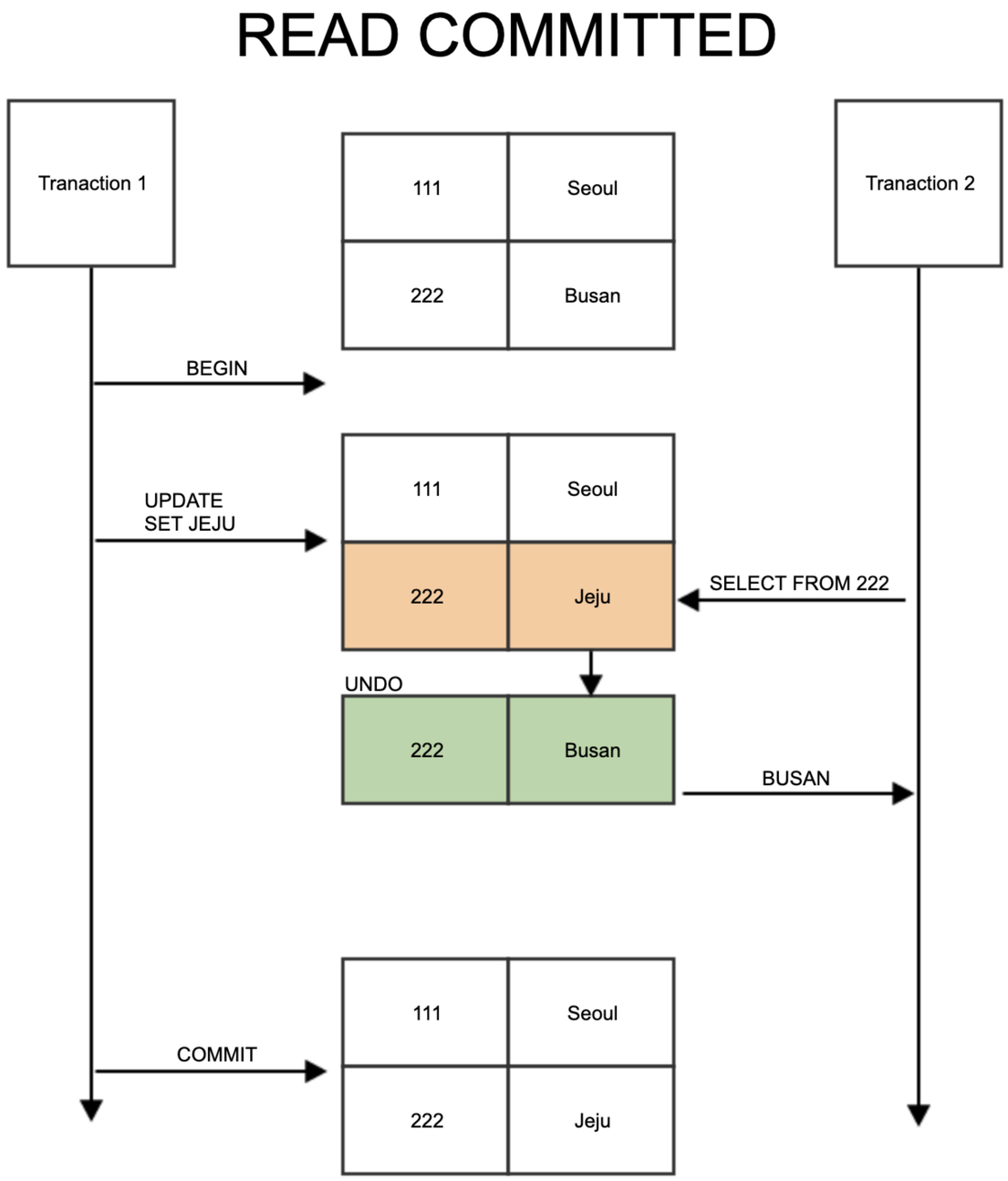
Read Uncommitted

- 다른 트랜잭션에서 커밋 안한거도 보임



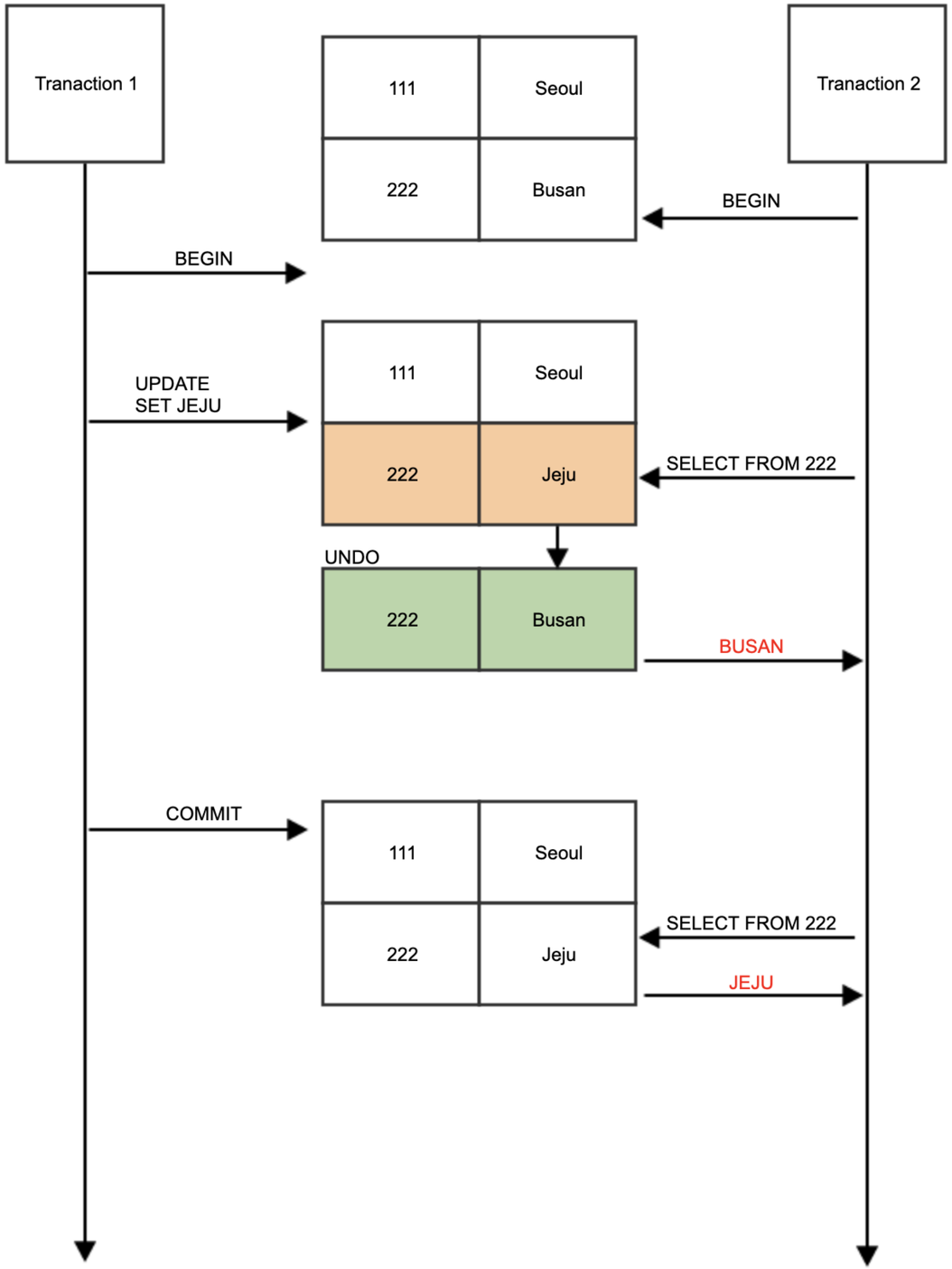
Read Committed

- 커밋 한 거만 보임
- Undo 영역 활동



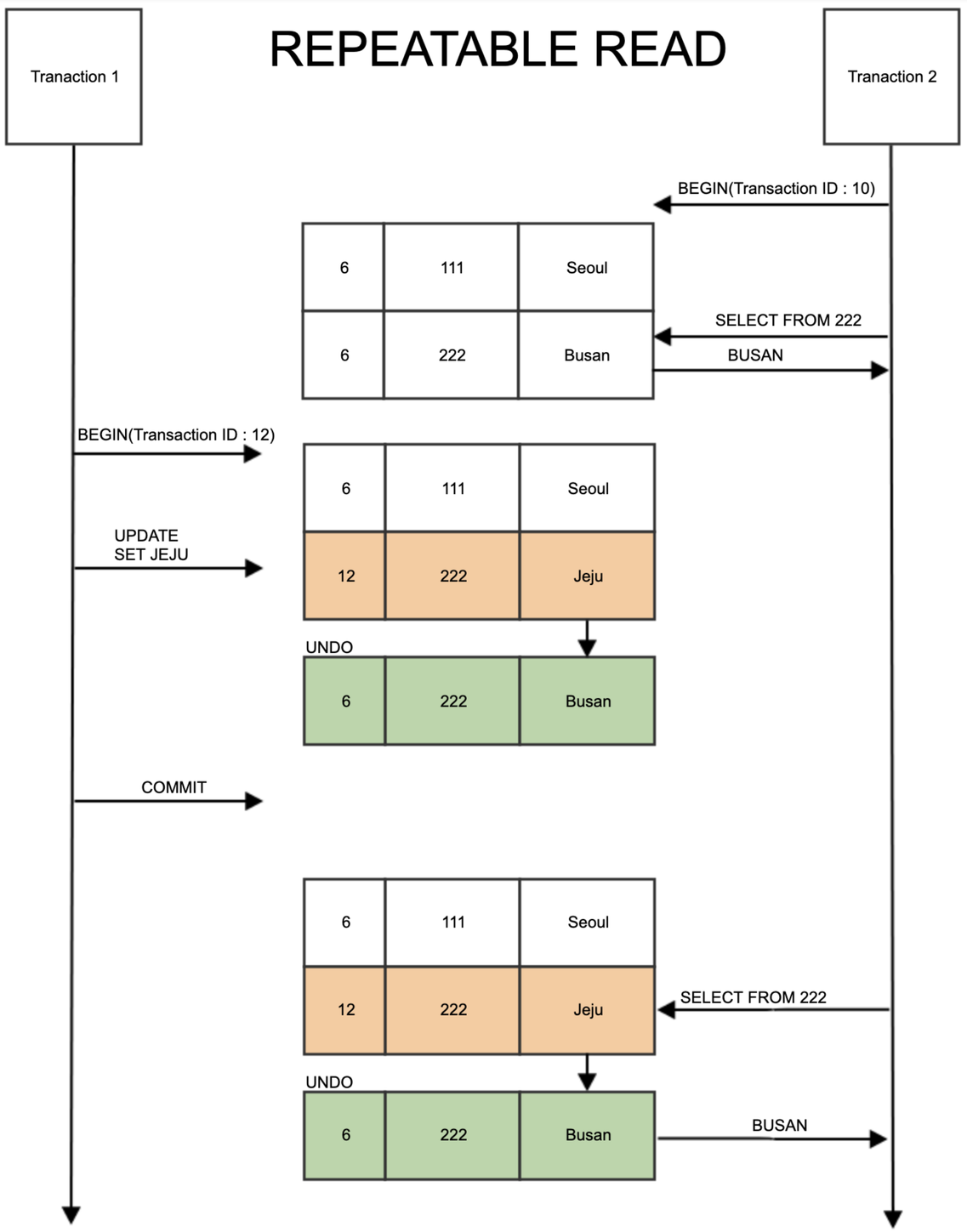
Read Committed

- Unepeatable read 문제 발생



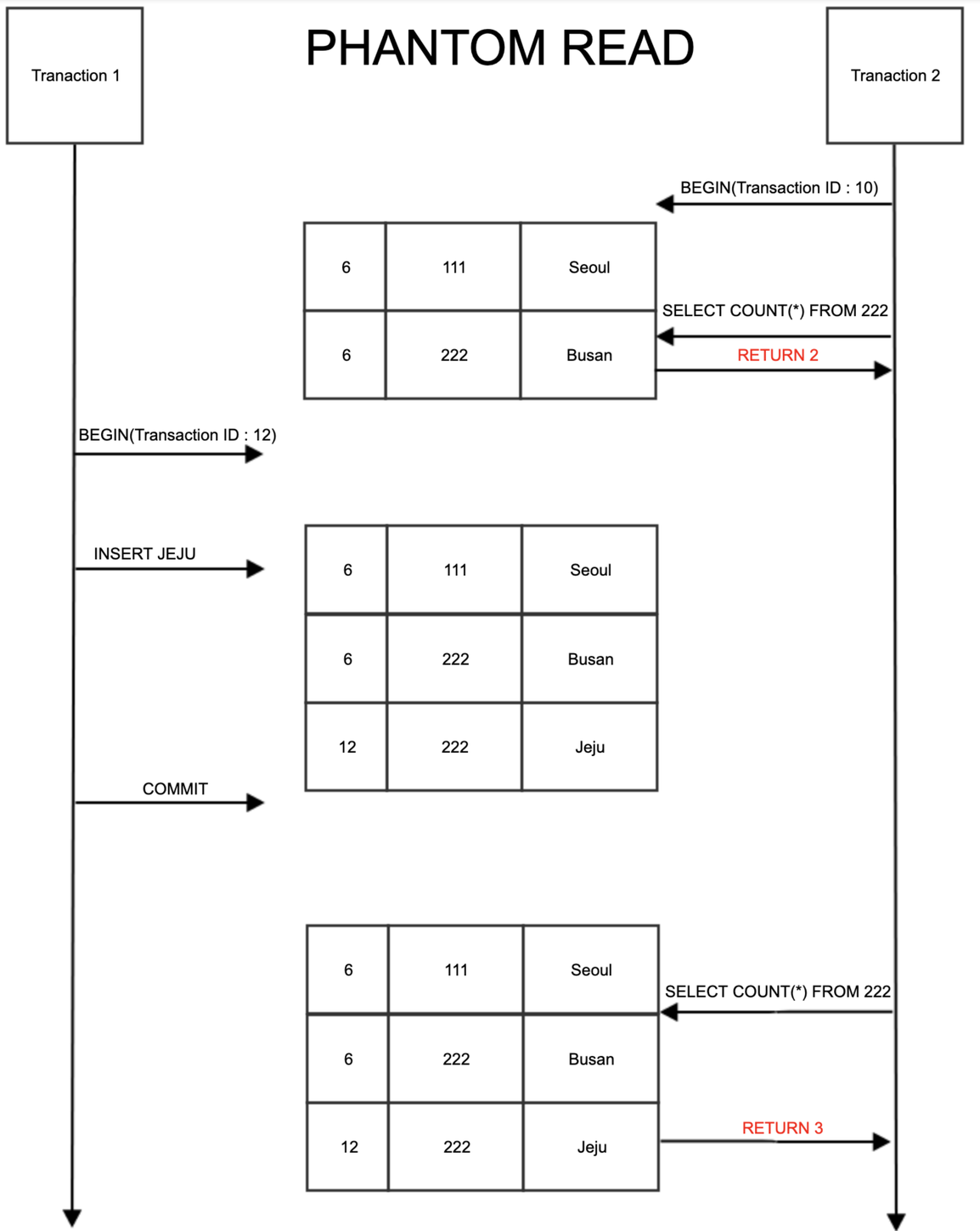
Repeatable Read

- MVCC 메커니즘 활용
- 일반적인 DB에서는 phantom read 발생



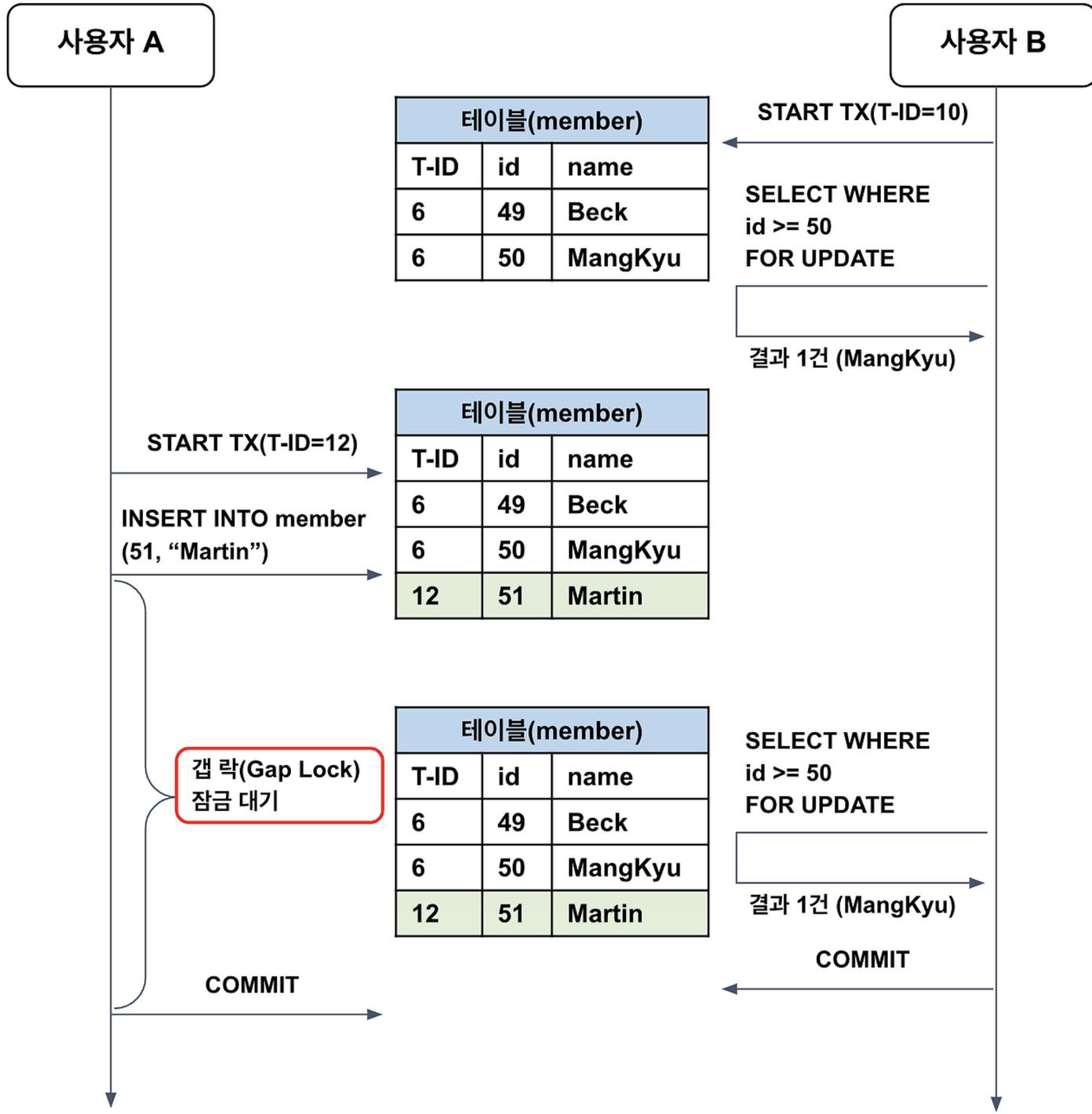
Repeatable Read

- MVCC 메커니즘 활용
- 일반적인 DB에서는 phantom read 발생



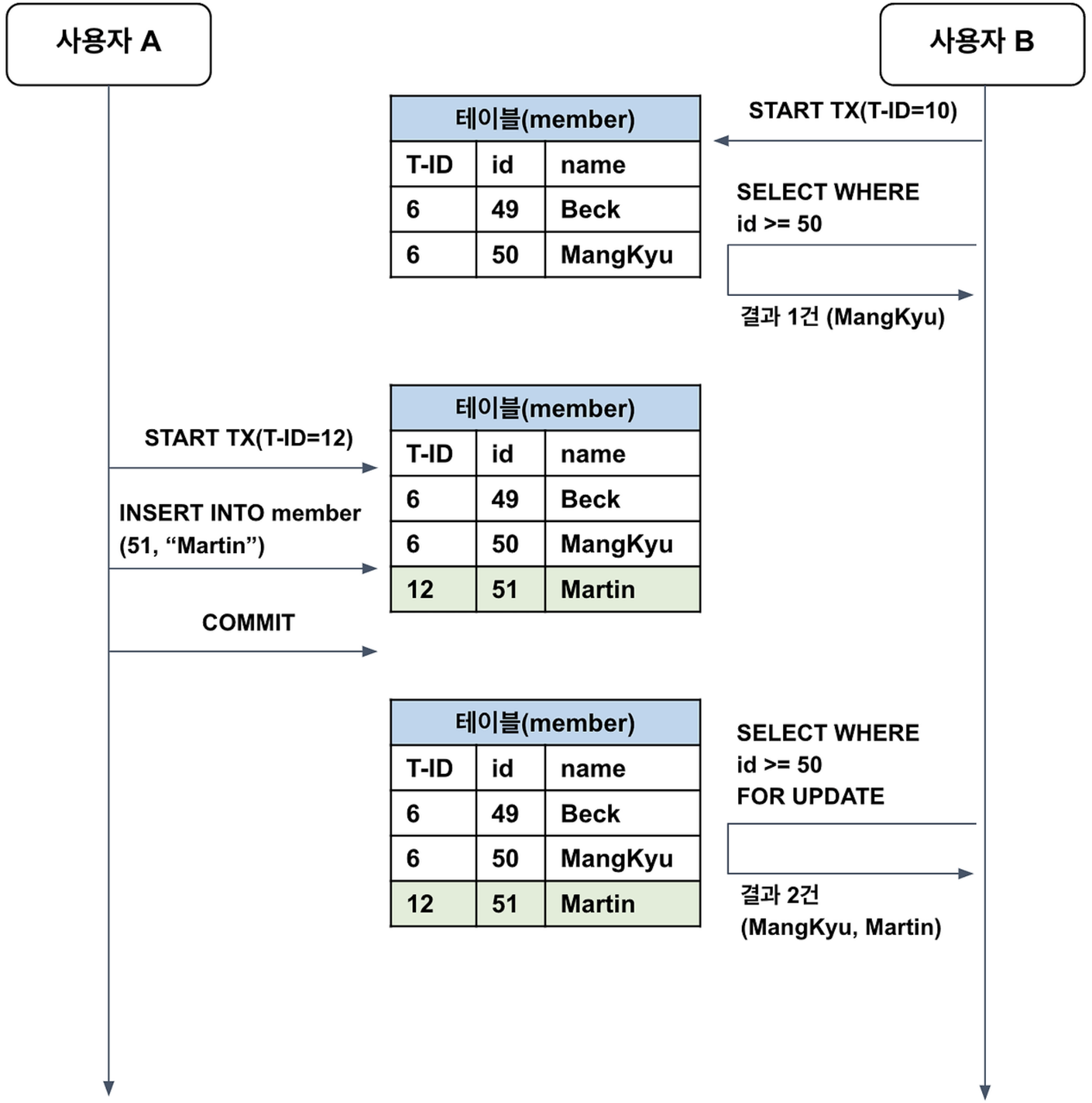
Repeatable Read

- My Sql에서는 Gap Lock을 활용해서
- 일반적인 상황에서는 발생하지 않음



Repeatable Read

- My Sql에서
- Phantom Read 발생하는 상황
- Undo 영역에는 락을 못검



Serializable

- 마치 하나씩 처리하 듯이
- 모든 SELECT문에 공유락 획득하도록 함
- 따라서 읽을 때는 절대 수정 못하게 막음

감사합니다.