인덱스를 추가하기 위해,
조회쿼리 중 현재 콘서트 프로젝트에서 사용되는 쿼리들은 아래와 같습니다.

- Reservation
findByUserIdAndScheduleId(long userId, long scheduleId)
: 해당 userId 에 해당 scheduleId가 존재하는지 조회

- UserBalance
findByUserId(long userId) : 잔액 조회
findByUserIdWithLock(long userId) : 잔액 조회 (락 획득)

- Concert
findAll : 콘서트 목록 조회

- Schedule
findAllByConcertId(long concertId) : Concert ID 로 해당 콘서트의 일정을 조회
findScheduleIdWithLock(long concertId, String concertDateTime)
: concert id 와 concert datetime 으로 스케쥴id를 조회 (락 획득)
findById(long scheduleId) : Schedule ID 로 스케쥴 정보 조회

- Seat
findEmptySeat(long scheduleId) : 해당 schedule id의 좌석 중 상태가 'EMPTY' 인 좌석 조회
findSeatForReserveWithLock(long scheduleId, long seatNumber)
: 해당 schedule id 와 seatnumber, 상태값이 'EMPTY' 인 좌석 조회
findByIdAndStatus(long seatId, SeatType.TEMPORARILY) : 임시배정상태인 좌석 조회

- User
findById(long userId) : 유저 조회
findByIdWithLock(long userId) : 유저 조회 (락획득)

인덱스가 필요할만한 쿼리로는 (자주 조회 혹은 복잡한 쿼리)
- 콘서트 목록 조회
- 콘서트 일정 조회
로 판단했습니다.
하지만 콘서트 목록 조회는 단순히 id, title 만 있는 테이블이라 인덱스가 필요없다고 판단하여
콘서트 일정 조회 (Schedule) 테이블로만 진행하였습니다.

```
EXPLAIN
SELECT schedule.*
FROM Schedule schedule
         JOIN Concert concert ON schedule.concert_id = concert.id
WHERE concert.id = 1;
```

(schedule 테이블에는 약 500만건의 데이터가 있는 상태입니다)

- index 설정X

explain 결과

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	concert	<null>	const	PRIMARY	PRIMARY	8	const	1	100	Using index
1	SIMPLE	schedule	<null>	ALL	<null>	<null>	<null>	<null>	4856915	10	Using where

table 이 schedule 인 것을 기준으로

key : null (사용한 인덱스가 없음)

type : All

rows : 4856915

인 것을 보아 처음부터 끝까지 스캔해야한다는 것을 알 수 있습니다.

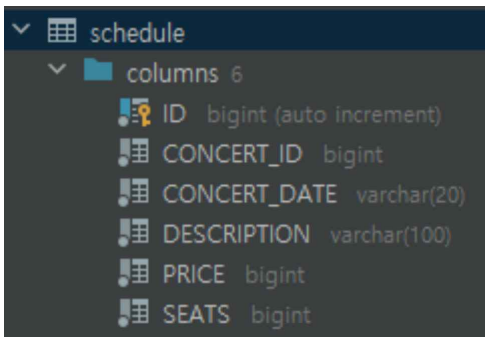
explain analyze 결과

```
EXPLAIN
-> Filter: (`schedule`.CONCERT_ID = 2)  (cost=510529 rows=485692) (actual time=0.362..2302 rows=2 loops=1)
    -> Table scan on schedule  (cost=510529 rows=4.86e+6) (actual time=0.36..2161 rows=5e+6 loops=1)
```

위의 결과로 table scan 할 때 실행시간이 0.36 초 부터 2161초까지 걸린걸 확인하였습니다.

인덱스 대상 컬럼은 높은 카디널리티(데이터 중복이 낮은)가 조건이 됩니다.

Schedule 테이블의 컬럼은 아래와 같습니다.



필요한 기능은 콘서트 일정 조회 이므로 concert_id (단일 인덱스) 로 걸면 될 것 같다고 판단하였습니다.

- concert_id (단일 인덱스) 설정

```
CREATE INDEX idx_concert_id ON Schedule(CONCERT_ID);
```

explain 결과

key : idx_concert_id (인덱스 적용됨)

type : ref

rows : 2428457

explain analyze 결과

```
EXPLAIN
-> Index lookup on schedule using idx_concert_id (CONCERT_ID=2)  (cost=2.18 rows=2) (actual time=0.682..0.684 rows=2 loops=1)
```

위의 결과로 table scan 할 때 실행시간이 0.682 초 부터 0.684초까지 걸린걸 확인하였습니다.

첫번째 행을 조회하는데의 시간은 약간 늘어났지만
마지막 행을 조회하는데의 시간이 3000배 넘게 줄은 것을 확인할 수 있었습니다.