



# Week 4 강의

슬로우 쿼리 사례 5선

문영급 함정 · 2026-06-27

아 답답해 (고구마 100개)



# × QUEST 왜 이 주제인가



- 운영 백엔드가 가장 자주 받는 알림: "쿼리 느려요"
- 면접 단골: "이 쿼리 왜 느린지?"
- 5 가지 운영급 함정

"느린 쿼리를 고치는 사람보다 찾는 사람이 더 귀하다."

# 오늘 다룰 것

- 운영급 사례 5 개 — 학생도 본 적 있을 패턴
- 각 사례 **이게 왜?** 30 초 추측
- EXPLAIN + 운영 도구로 정답 확인
- 해결 + 이력서 카드

5 개 = 5 개 카드. [evidence/interview-cards/](https://www.notion.so/evidence/interview-cards/) 누적.

# 시작 전 - 용어 카드

용어	한 줄 정의
슬로우 쿼리	임계 시간(보통 1초+) 넘는 쿼리
슬로우 쿼리 로그	DB 가 자동으로 남기는 느린 쿼리 모음
EXPLAIN	쿼리 실행 전 옵티마이저 계획 보기
카디널리티	컬럼 값의 다양성 (낮으면 인덱스 효과 X)
옵티마이저	DB 가 쿼리 실행 계획 결정하는 두뇌

모르는 단어 나오면 이 표 다시.

# ✕ QUEST 사례 1 — deep page 8 초

```
SELECT * FROM order  
ORDER BY created DESC  
LIMIT 100 OFFSET 50000;
```

order: 100만 row  
created 인덱스 없음

🕒 page 1 = 10ms

🕒 page 500 = 8 초

🤔 왜 deep page 만 느릴까?

# 사례 1 — 정답 + 실무 패턴

OFFSET 50000

- DB 가 50,100 row 읽고 \_50,000 버림\_
- 페이지 깊을수록 \_더 느려짐\_

✓ 커서 기반 페이지네이션:

```
WHERE created < '2026-01-01 12:00:00'  
ORDER BY created DESC  
LIMIT 100;
```

- 8 초 → 10ms (페이지 깊이 무관)

페이스북·인스타 타임라인이 커서 인 이유.

# ✕ QUEST 사례 2 - 갑자기 느려짐

```
SELECT * FROM post
WHERE status = 'PUBLISHED'
      AND user_id = 5
ORDER BY created DESC;
```

어제까지: 50ms

오늘: 3 초

DB 변경 없음. 코드 변경 없음.

🤔 무엇이 바뀌었나?

# 사례 2 — 정답

status='PUBLISHED' 카디널리티 `_바뀜_`  
어제: 30% (적중) → 오늘: 95% (대다수)  
→ 옵티마이저가 `_풀스캔_` 선택

- ✓ ANALYZE TABLE post (통계 갱신)
- ✓ 복합 인덱스 (user\_id, status, created)
- ✓ USE INDEX 힌트 (마지막 수단)

데이터 분포가 옵티마이저를 좌우.

# ✕ QUEST 사례 3 — 평균만 8초

-- 단일 호출 측정: 80ms ✓

```
SELECT * FROM product WHERE id = 5;
```

APM 평균:            8 초

슬로우 쿼리 로그:    이 쿼리 \_다수\_ 보임

DB CPU:             20%

🤔 단일은 빠르는데 \_평균\_ 은 왜?

# 사례 3 — 정답: 락 대기

```
@Transactional
public void bigBatch() {
    product.findById(5); // row 락 잡음
    externalApi.call(); // 30 초 대기 (!)
    product.save();
}
// → 그 사이 같은 row 조회·수정 _대기_
```

- ✓ 트랜잭션 \_짧게\_ – 외부 호출은 Tx 밖
- ✓ innodb\_lock\_wait + 슬로우 쿼리 같이

# ✕ QUEST 사례 4 - TEXT 컬럼 함정

```
SELECT * FROM post WHERE id = 5;
```

어제까지: 5ms

오늘: 150ms

변경: post 에 content\_long (TEXT) 컬럼 추가

😬 인덱스로 가져오는 1 row 인데 30 배 느림?

# 사례 4 — 정답: SELECT \*

TEXT/BLOB 컬럼은 별도 페이지 에 저장  
→ id 로 row 찾고도 TEXT 페이지 IO 추가  
→ 1 row 인데 IO 폭발

게다가 네트워크로 100KB 전송 → 클라 메모리 ↑

- ✓ SELECT id, title, summary (필요한 컬럼만)
- ✓ TEXT 는 별도 테이블 (PostBody 분리)
- ✓ 큰 컬럼은 요청 시 만 fetch (lazy column)

"백엔드는 **SELECT \*** 안 쓴다." — 면접 답변.

# ✕ QUEST 사례 5 - 인덱스 적중인데

```
SELECT * FROM order o
JOIN user u ON o.user_id = u.id
WHERE o.status = 'PENDING'
      AND u.tier = 'VIP';
```

order, user 모두 인덱스 \_많음\_  
EXPLAIN 결과: 모든 컬럼 인덱스 적중  
🕒 실측: 4 초  
😞 인덱스가 다 적중인데 왜 느릴까?

# 사례 5 — 정답: JOIN 순서

✓ 좋은 순서: user(VIP 100) → order JOIN

→ 100 × lookup = 빠름

✗ 잘못 고른 순서: order(50만) → user JOIN

→ 50만 × lookup = 느림

✓ ANALYZE TABLE (통계 정확)

✓ STRAIGHT\_JOIN 힌트 (마지막)

# 5 사례 한눈에

#	사례	핵심
1	deep page 8 초	커서 페이지네이션
2	갑자기 느려진 쿼리	통계 + 카디널리티
3	단일 빠르는데 평균 느림	트랜잭션 길이·락 대기
4	TEXT 추가 후 30배	SELECT * 안 쓰기
5	인덱스 적중인데 느림	JOIN 순서·통계

면접에서 이 5 개 패턴 짚을 수 있으면 운영 감각.

# 운영 도구 (실무 20)

```
# my.cnf
slow_query_log = 1
long_query_time = 1.0      # 1 초 이상
log_queries_not_using_indexes = 1
```

- ✓ Percona pt-query-digest - 자동 분석
- ✓ DataDog / NewRelic APM - UI 시각화
- ✓ pg\_stat\_statements (PG) - 누적 통계
- ✓ AWS RDS Performance Insights

면접에서 슬로우 쿼리 로그 + APM 언급 = 운영 감각 신호.

# 실무에서 — 슬로우 쿼리 자리

도메인	흔한 사례
페이지네이션	OFFSET deep page (사례 1)
관리자 통계	통계 노후 (사례 2)
결제 / 주문	트랜잭션 길이 (사례 3)
게시판 / CMS	TEXT 컬럼 (사례 4)
검색 / 필터	JOIN 순서 (사례 5)

5 개 도메인 무관 패턴. 어떤 회사 가도 이 5 개 봄.

# 이력서엔 — 슬로우 쿼리 카드

[관리자 페이지 deep page 8s → 10ms]

P (문제) /admin/orders page=500 응답 8s - 사용자 클레임

O (옵션) 캐시 / 인덱스 변경 / 커서 페이지네이션

D (결정) 커서 기반 - DB 부하 \_일정\_ (페이지 무관)

A (행동) created\_id 복합 커서 + API 변경 + frontend 협업

R (결과) page 500 8s → 10ms (-99%), DB CPU 60% → 5%

# AI 잘 쓰는 법

- **잘하는 것:** EXPLAIN 1 차 해석, 인덱스 후보 추천
- **자주 hallucinate:** 카디널리티 추측, 옵티마이저 예측
- **검증 루프:**
  1. 슬로우 쿼리 로그 **실제** 추출
  2. AI 에 EXPLAIN + 데이터 분포 함께 전달
  3. 본인 환경에서 EXPLAIN 직접
  4. before/after 측정 → evidence

AI 추천 인덱스가 **진짜로** 쓰이는지 EXPLAIN 으로 확인.

# 그럼 미션은?

- W4 미션 `05-week4-index` 통과 조건 = before/after EXPLAIN + latency
- 오늘 5 사례 중 본인 코드와 비슷한 1 개 골라 evidence
- `latency-comparison.md` 표 형식

막히면 → `{cohort}-질문` 채널 + 오피스아워 (화·목 `21:00`)

# 질문 ㄱㄱ ?

이번 주 = "내 쿼리에 5 패턴 있는지 점검"  
다음 면접 = "이 쿼리 왜 느릴까?" 답할 수 있게

다음 격주 강의(㉮6): **분산 추적의 세계.**