

Covering Index For Optimization

목표: 이 강의에서는 커버링 인덱스를 통한 성능 최적화 하는 방법을 배웁니다.

포인트: 커버링 인덱스가 무엇이고, 이를 언제 적용할 수 있는지에 대해 이해하는게 중요합니다.

▼ What are Covering Indexes?

커버링 인덱스는 쿼리 성능 최적화를 위해 사용되는 기법입니다.

일반적으로 인덱스를 사용하여 쿼리를 처리할 때, 인덱스에 있는 컬럼만으로 충분하지 않아 실제 테이블 레코드에 접근해야 할 경우가 있습니다.

하지만 커버링 인덱스를 사용하면 테이블 레코드에 접근하는 과정 없이 인덱스 수준에서만 쿼리를 처리할 수 있습니다. 이를 통해 읽기 성능이 향상됩니다.

▼ When to Use Covering Indexes

1. **특정 컬럼 자주 조회:** 특정 컬럼 때문에 테이블에 접근하는 것이 비용이 크게 느껴질 때 유용합니다.
2. **조인 연산 비용 줄이기:** 여러 테이블을 연결할 때 발생하는 비용을 줄일 수 있습니다.
3. **읽기 성능이 필요한 경우:** 인덱스 레벨에서만 필터링해서 읽기 성능이 필요한 경우에 적용할 수 있습니다.

▼ Things to Consider When Applying

- **인덱스 크기:** 인덱스가 커지면 인덱스 블록에 들어갈 수 있는 데이터의 수는 줄어듭니다.

- **쓰기 비용:** 인덱스에 추가된 컬럼이 자주 업데이트되는 경우, 추가적인 쓰기 비용이 발생할 수 있습니다.
- **컬럼 크기:** 큰 크기의 컬럼(VARCHAR(255) 등)은 인덱스로 추가하기 비효율적일 수 있습니다.
- **카디널리티:** 카디널리티가 낮은 값의 컬럼을 인덱스로 추가할 경우, 읽기 성능이 향상되지 않을 수 있습니다.

▼ Practice

Task: Trying Out a Covering Index

목적: 기존 인덱스를 커버링 인덱스로 변경해서 성능 향상을 이뤄내보자.

시나리오: 온라인 쇼핑몰의 주문 관리 시스템

테이블 구조: `orders`

- `order_id` (주문 ID, 기본 키)
- `customer_id` (고객 ID)
- `order_date` (주문 날짜)
- `quantity` (수량)

고객의 최근 주문 조회 쿼리

```
SELECT * FROM orders
WHERE customer_id = 8485731
ORDER BY order_date DESC
LIMIT 10;
```

1. 테이블 생성:

```
CREATE TABLE orders (
  id INT AUTO_INCREMENT PRIMARY KEY,
```

```
customer_id INT NOT NULL,  
order_date DATE NOT NULL,  
total_amount DECIMAL(10, 2) NOT NULL  
);
```

2. 테스트 데이터 삽입:

3. 인덱스 생성:

```
CREATE INDEX idx_customer_order_date ON orders(customer_id, order_date);
```

4. 실행 계획과 쿼리 소요 시간 분석:

```
EXPLAIN SELECT *  
FROM orders  
WHERE customer_id = 8485731  
ORDER BY order_date DESC  
LIMIT 10;
```

```
EXPLAIN ANALYZE SELECT *  
FROM orders  
WHERE customer_id = 8485731  
ORDER BY order_date DESC  
LIMIT 10;
```

5. 커버링 인덱스 생성:

```
CREATE INDEX idx_covering ON orders(customer_id, order_date, total_amount);
```

6. 실행 계획과 쿼리 소요 시간 분석:

```
EXPLAIN SELECT *  
FROM orders
```

```
WHERE customer_id = 25675  
ORDER BY order_date DESC  
LIMIT 10;
```

```
EXPLAIN ANALYZE SELECT *  
FROM orders  
WHERE customer_id = 25675  
ORDER BY order_date DESC  
LIMIT 10;
```