

● index를 어느 column에 사용하는 것이 좋을까?

▶ index는 select where 절에서 자주 조회되고, 수정 빈도가 낮으면서, 데이터 중복이 적은 (카디널리티는 높고, 선택도가 낮은) column을 선택해서 설정하는 것이 가장 좋습니다.

▶ select where절에서 조회를 하기 위해서 index를 사용하는 것이기 때문에, 조회 활용도가 높을 수록 적합합니다. 또한 수정할 때마다 insert/update/delete 작업 시에 index를 매번 다시 정렬해야 하고, 이에 따른 부하가 발생하기 때문에, 수정 빈도가 낮은 column을 index로 설정하는 것이 좋습니다. 또한 카디널리티란 데이터가 중복되지 않는 정도를 뜻하는데, 중복되는 값이 없을 수록 cardinality가 높습니다. 데이터 중복이 높은 column은 index효과가 별로 없기 때문에, 카디널리티가 높을 때 index를 사용하는 것이 유리합니다. 반면에 선택도는 카디널리티와 반대로, 데이터에서 특정 값을 잘 골라낼 수 있는 정도를 뜻하는데, 특정 값을 잘 골라낼 수 있을 수록 선택도가 낮으며, 선택도가 1이면 모든 데이터가 unique함을 의미합니다. 즉, 카디널리티는 높고 선택도가 낮은 column을 선택해서 index를 설정하는 것이 좋습니다.

기준	적합성
카디널리티(Cardinality)	높을수록 적합 (데이터 중복이 적을수록 적합)
선택도(Selectivity)	낮을수록 적합
조회 활용도	높을수록 적합 (where 절에서 많이 사용되면 적합)
수정 빈도	낮을수록 적합

● Index 효과적으로 사용하는 방법

▶ 데이터의 양이 많을 수록 index로 인한 성능향상이 더 큽니다. 데이터 양이 적다면 index의 혜택보단 손해가 더 클 수 있습니다. Full table scan을 해도 많은 시간이 소요되지 않기 때문입니다.

▶ Join 조건으로 자주 사용되는 column의 경우 index를 생성하는 것이 좋습니다.

▶ 한 table에 index가 너무 많으면 데이터 수정시 소요되는 시간이 너무 길어질 수 있습니다. (그래서 보통 table당 4~5개 정도가 권장됩니다).

- SELECT WHERE절에 자주 사용되는 Column에 대해 index를 생성하는 것이 좋습니다. Select where절의 성능의 높이기 위한 것이 index이기 때문입니다.

- 데이터 수정 빈도가 낮을수록 적합합니다. insert / update / delete 작업 시, 데이터에 변화가 생기기 때문에 index에서는 매번 정렬을 다시 해야합니다. 이에 따른 부하가 발생하기 때문에 수정 빈도가 낮은 column을 index로 설정하면 좋습니다.

- 데이터의 중복이 높은 column은 index 효과가 별로 없습니다. 예를 들어 성별은 종류가 2 가지 밖에 없으므로 index를 생성하지 않는 것이 좋습니다. 즉, 선택도가 낮을 때, 보통 5~10%이내일 때 유리합니다.

● index를 쓰면 성능이 좋아지니까 모든 컬럼에 인덱스를 사용하면 성능이 더 좋겠네요?

▶ 그렇지 않습니다. 일단 index는 SELECT WHERE절에 대해서만 성능향상을 해줍니다. 이외에 데이터를 수정하게 되면 모든 index를 업데이트(+정렬)해야 하기 때문에 오히려 성능저하를 초래합니다. 또한 index를 생성할 때마다 저장공간도 차지하기 때문에 무분별하게 생성해서는 안됩니다.

● 우리 회사의 고객 DB에서 성별 column에 index를 걸어주는게 좋을까요?

▶ 좋지 않습니다. 성별처럼 남녀 두 종류로만 나뉘지는 경우에는 카디널리티가 매우 낮고, 선택도는 매우 높게 됩니다. 이 경우에는, index가 주는 이점이 매우 적고 오히려 저장공간 차지와 데이터 수정시 성능저하등을 고려하여 index를 생성하지 않는 것이 좋습니다.

● true 또는 false 값을 갖는 column에서, true 1%, false 99%의 비율로 구성된 상황에서는 index를 거는게 좋을까요?

▶ 아무리 1%, 99%로 비율에 차이가 있어도, true / false 두 종류로만 나뉘지는 경우에는 카디널리티가 매우 낮고, 선택도는 매우 높게 됩니다. 이 경우, index가 주는 이점이 매우 적고 오히려 저장공간 차지와 데이터 수정시 성능저하등을 고려하여 index를 생성하지 않는 것이 좋습니다.

Resources: inflearn 개발남 노씨
