

## 제2장. 데이터 모델과 성능

31. ①

해설 : 분석/설계 단계에서 데이터베이스 처리 성능을 향상 시킬 수 있는 방법을 주도 면밀하게 고려해야 한다. 만약 어떤 트랜잭션이 해당 비즈니스 처리에 핵심적이고 사용자 업무처리에 있어 중요함을 가지고 있고 성능이 저하되면 안되는 특징을 가지고 있다면, 프로젝트 초기에 운영환경에 대비한 테스트 환경을 구현하고 그곳에 트랜잭션을 발생시켜 실제 성능을 테스트해 보아야 한다. 이때 데이터 모델의 구조도 변경하면서 어떠한 구조가 해당 사이트에 성능상 가장 적절한 구조인지를 검토하여 성능이 좋은 모습으로 디자인 하는 전략이 요구된다. 보기에서 문제 발생 시점의 SQL을 중심으로 집중 튜닝하는 것은 성능 데이터모델링과 무관한 내용이다.

32. 반정규화(역정규화)

해설 : 데이터 모델링 단계에서 성능을 충분히 고려하기 위한 성능 데이터 모델링 수행 절차에 대한 설명으로, 그 과정은 다음과 같다.

첫번째, 데이터모델링을 할 때 정규화를 정확하게 수행한다.

두번째, 데이터베이스 용량산정을 수행한다.

세번째, 데이터베이스에 발생하는 트랜잭션의 유형을 파악한다.

네번째, 용량과 트랜잭션의 유형에 따라 반정규화를 수행한다.

다섯번째, 이력모델의 조정, PK/FK조정, 슈퍼타입/서브타입 조정 등을 수행한다.

여섯번째, 성능관점에서 데이터모델을 검증한다.

33. ④

해설 : 성능을 고려한 데이터모델링은 정규화를 수행한 이후에 용량산정과 트랜잭션 유형을 파악하여 반정규화를 수행한다. 또한 PK/FK등을 조정하여 인덱스의 특징을 반영한 데이터모델로 만들고 이후에 데이터모델을 검증하는 방법으로 전개한다.

34. ①

해설 : 정규화가 항상 조회 성능을 저하시킨다는 것은 잘못된 생각이며 기본적으로 중복된 데이터를 제거함으로써 조회성능을 향상시킬 수 있음을 알아야 한다.

35. ③

해설 : 함수중속성의 규칙에 따라 {관서번호} → {관리점번호, 관서명, 상태, 관서등록일자}인 관서번호가 PK인 엔티티가 2차 정규화로 분리되어야 한다.

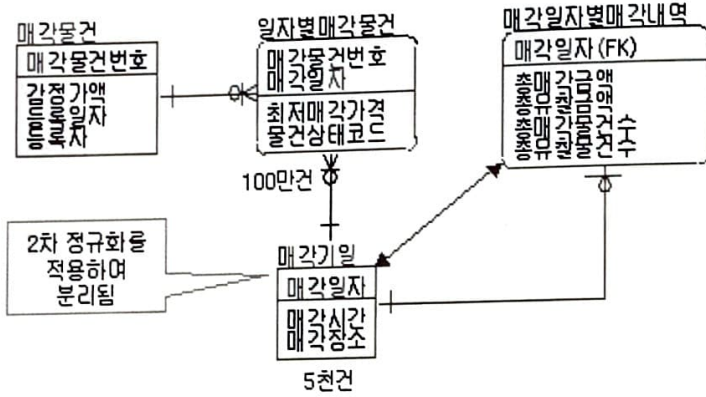
36. ③

해설 : 매각기일은 일자별로 매각이 시행되는 장소와 시간을 의미하는 것으로, 일자별매각물건 엔터티의 매각시간, 매각장소 속성은 두 개의 주식별자 속성 중 매각일자에만 종속되기 때문에 2차 정규화 대상이 된다.

그러므로 매각일자를 주식별자로 하고 매각시간과 매각장소 속성을 포함하는 매각기일 엔터티를 독립시킨다. 이때 매각기일 엔터티는 일자별매각물건의 주식별자 중 일부로부터 독립했기 때문에 매각기일과 일자별매각물건은 1:M 관계로 연결된다.

이와 같은 2차 정규화를 통해 특정 장소에서 이루어진 매각내역을 조회하고자 할 때 100만건의 일자별매각내역 데이터를 모두 읽어 원하는 장소에 해당하는 인스턴스들을 찾아 매각일자별로 그룹핑한 후 매각일자별매각내역과 조인할 필요가 없이 매우 적은 수의 매각기일 엔터티에서 특정 장소에 해당하는 매각일자들을 찾아 매각일자별매각내역과 1:1로 바로 조인하면 되기 때문에 I/O를 현저하게 감소시킬 수 있어 성능 향상 효과를 얻을 수 있다.

성능이 저하된 반정규화 사례 - 정규화를 통한 성능향상



특정 매각장소에 대해 매각일자들을 찾아 매각내역을 조회하려면 500건의 매각기일과 매각일자별매각내역과 조인이 된다.

37. ④

해설 : 컬럼에 의한 반복적인 속성값을 갖는 형태는 속성의 원자성을 위배한 제1차 정규화의 대상이 된다. 이와 같은 반복적인 속성 나열 형태에서는 각 속성에 대해 'or' 연산자로 연결된 조건들이 사용되는데, 이 때 어느 하나의 속성이라도 인덱스가 정의되어 있지 않게 되면 'or'로 연결된 모든 조건절들이 인덱스를 사용하지 않고 한 번의 전체 데이터 스캔으로 처리되게 되어 성능 저하가 나타날 수 있게 되며, 또한 모든 반복 속성에 인덱스를 생성하게 되면 검색 속도는 좋아지겠지만 반대급부적으로 너무 많은 인덱스로 인해 입력, 수정, 삭제의 성능이 저하되므로, 1차 정규화를 통해서 자연스럽게 문제가 해결될 수 있도록 해야 한다.

38. ①

해설 : 컬럼 단위에서 중복된 경우도 1차 정규화의 대상이 된다. 이에 대한 분리는 1:M의 관계로 두 개의 엔터티로 분리된다.

## 39. ①

해설 : PK에 대해 반복이 되는 그룹(Repeating)이 존재하지 않으므로 1차 정규형이라고 할 수 있으며, 부분 함수종속의 규칙을 가지고 있으므로 2차 정규형이라고 할 수 없음. 2차 정규화의 대상이 되는 엔터티임.

## 40. ①

해설 : • 다량 데이터 탐색의 경우 인덱스가 아닌 파티션 및 데이터 클러스터링 등의 다양한 물리 저장 기법을 활용하여 성능 개선을 유도할 수 있다. 다만, 하나의 결과셋을 추출하기 위해 다량의 데이터를 탐색하는 처리가 반복적으로 빈번하게 발생한다면 이때는 반정규화를 고려하는 것이 좋다.

- 이전 또는 이후 위치의 레코드에 대한 탐색은 window function으로 접근 가능하다.
- 집계 테이블 이외에도 다양한 유형(다수 테이블의 키 연결 테이블 등)에 대하여 반정규화 테이블 적용이 필요할 수 있다.

## 41. ④

해설 : 반정규화의 기법은 테이블, 속성, 관계에 대해서 반정규화를 적용할 수 있으며, 하나의 테이블의 전체 칼럼 중 자주 이용하는 집중화된 칼럼들이 있을 때 디스크 I/O를 줄이기 위해 해당 칼럼들을 별도로 모아 놓는 반 정규화 기법은 테이블추가 반정규화 기법 중에서 부분테이블 추가에 해당한다.

## 42. ③

해설 : ③ FK에 대한 속성 추가는 반정규화 기법이라기 보다는 데이터모델링에서 관계를 연결할 때 나타나는 자연스러운 현상이다.

## 43. ③

해설 : 제품 엔터티에 단가를 주문번호별로 합하는 것은 해당 제품이 여러 주문에 포함될 수 있기 때문에 특정 주문번호만의 단가 합계금액을 갖고 있을 수 없고, 주문목록 엔터티에 주문번호별 단가 합계 금액을 추가하게 되면 하나의 주문에 포함된 제품번호마다 동일한 합계 금액을 반복적으로 저장해야 해서 일관성 문제가 발생할 수 있다. 또한 제품 엔터티에 최근값 여부 칼럼을 추가하는 것은 단가 합계 금액을 빠르게 얻기 위한 반정규화와 무관한 조치이다. 그러므로 주문 엔터티에 전체를 통합한 계산된 칼럼을 추가하는 것이 한번에 데이터를 조회하는 방법이 되므로 가장 효과적인 반정규화 기법이다.

## 44. ①

해설 : 최근에 변경된 값만을 조회할 경우 과도한 조인으로 인해 성능이 저하되어 나타나게 된다.



#### 45. ③

해설 : 한 테이블에 많은 칼럼들이 존재할 경우 데이터가 물리적으로 저장되는 디스크 상에 넓게 분포할 가능성이 커지게 되어 디스크 I/O가 대량으로 발생할 수 있고, 이로 인해 성능이 저하될 수 있음. 따라서 트랜잭션이 접근하는 칼럼유형을 분석해서 자주 접근하는 칼럼들과 상대적으로 접근 빈도가 낮은 칼럼들을 구분하여 1:1로 테이블을 분리하면 디스크 I/O가 줄어들어 성능을 향상 시킬 수 있다. 테이블 내에서 칼럼의 위치를 조정하는 것은 데이터 주로 채워지는 칼럼을 앞 쪽에 위치시키고, 데이터가 채워지지 않고 주로 NULL 상태로 존재하는 칼럼들을 뒤쪽에 모아둠으로써 로우의 길이를 어느 정도 감소시킬 수 있으나, NULL 상태이던 칼럼에 나중에 데이터가 채워지게 될 경우 더 많은 로우 체인이 발생할 수도 있기 때문에 바람직한 해결책이라고 보기에 부족하며, 무엇보다도 데이터가 채워지지 않고 NULL 상태로 존재하게 되는 칼럼들이 많이 나타나는 경우는, 너무 많은 엔터티들에 무리하게 동일성을 부여하여 통합을 수행했거나, 예측하기 어려운 미래 시점을 겨냥하여 과도하게 의욕적으로 속성을 확장한 경우 등에서 주로 나타나기 때문에, 자주 사용되는 칼럼들이나 현시점에서 주로 사용되는 칼럼들을 한데 모으고, 사용빈도가 낮은 칼럼들이나 미래 시점에 사용될 것으로 예상되는 나머지 칼럼들을 한데 모아 별도의 1:1 관계 엔터티로 분리하는 등의 데이터모델 설계 수정을 고려해 보는 것이 좋다.

#### 46. 파티셔닝(Partitioning)

#### 47. ②

해설 : 개별 테이블을 모두 조회하는 트랜잭션이 대부분이라는 가정이 있으므로 UNION/UNION ALL할 경우 개별조회에 따른 시간소요와 이것을 조합하는 성능저하가 발생된다. 따라서 하나의 테이블로 통합하도록 하고 대신 PK체계나 일반속성에 각 사건을 구분할 수 있도록 구분자를 부여한다.

#### 48. ④

해설 : 트랜잭션은 항상 전체를 통합하여 분석 처리하는데 슈퍼-서브타입이 하나의 테이블로 통합되어 있으면 하나의 테이블에 집적된 데이터만 읽어 처리할 수 있기 때문에 다른 형식에 비해 더 성능이 우수하다. (조인 감소)

#### 49. ④

해설 : 인덱스는 값의 범위에 따라 일정하게 정렬이 되어 있으므로 상수값으로 EQUAL 조건으로 조회되는 칼럼이 가장 앞으로 나오고 범위조회 하는 유형의 칼럼이 그 다음에 오도록 하는 것이 인덱스 액세스 범위를 좁힐 수 있는 가장 좋은 방법이 됨

#### 50. ②

해설 : '='로 들어온 조건에 해당하는 칼럼이 인덱스의 가장 앞쪽에 위치할 때 인덱스의 이용 효율성이 가장 높다고 할 수 있다.

## 51. ②, ④

해설 : 엔터티 간에 논리적 관계가 있을 경우 즉, 엔터티 간에 관계(Relationship)를 정의하여 관련 엔터티 상호간에 업무적인 연관성이 있음을 표현한 경우에는, 이 데이터들이 업무적으로 밀접하게 연결되어 상호간에 조인이 자주 발생한다는 것을 의미하는 것이기 때문에, 데이터베이스 상에서 DBMS가 제공하는 FK Constraints를 생성했는지 여부와 상관없이 조인 성능을 향상시키기 위한 인덱스를 생성해주는 것이 좋다. 그러므로 수강신청테이블의 학사기준번호에 인덱스가 필요하다.

데이터베이스에 생성하는 FK Constraints는 데이터 모델 상에 표현된 논리적 관계에 따라 관련 인스턴스 간에 일관성을 보장하기 위해 설계된 제약조건을 구현할 수 있도록 DBMS가 제공하는 하나의 '지원 기능'으로 이해될 수 있다.

## 52. ④

해설 : Global Single Instance(GSI)는 통합된 한 개의 인스턴스 즉, 통합 데이터베이스 구조를 의미하므로, 분산데이터베이스와는 대치되는 개념이다.

공통코드, 기준정보 등과 같은 마스터 데이터를 한 곳에 두고 운영하는 경우 원격지에서의 접근이 빈번할수록 실시간 업무처리에 대해 좋은 성능을 얻기가 어려울 수 있기 때문에 분산 환경에 복제분산을 하는 방법으로 분산데이터베이스를 구성할 수 있다. 또한 백업 사이트 구성에 대해서도 분산 환경으로 구성하여 적용할 수 있다.