2. 데이터 모델링과 성능

1. 아래 설명을 읽고 빈칸에 들어갈 단어를 작성하시오.

- 아래-
첫번째. 데이터모델링을 할 때 정규화를 정확하게 수행한다.
두번째. 데이터베이스 용량산정을 수행한다.
세번째. 데이터베이스에 발생되는 트랜잭션의 유형을 파악한다.
네번째. 용량과 트랜잭션의 유형에 따라 ( )을 수행한다.
다섯번째. 이력모델의 조정, PK/FK조정, 슈퍼타입/서브타입 조정 등을 수행한다.

2. 아래에서 성능을 고려한 데이터 모델링의 순서대로 나열하시오.

① 데이터모델링을 할 때 정규화를 정확하게 수행
② 용량과 트랜잭션의 유형에 따라 반정규화를 수행
③ 데이터베이스 용량산정을 수행
④ 데이터베이스에 발생되는 트랜잭션의 유형 파악
⑤ 성능관점에서 데이터 모델 검증
⑥ 이력모델의 조정, PK/FK조정, 슈퍼타입/서브타입 조정 등을 수행

3. 성능데이터 모델링을 할 때 고려사항으로 가장 부적절한 것은?

① 데이터 모델링의 정규화는 항상 조회 성능저하를 나타내므로 반정규화 관점에서만 성능을 고려하여 설계하도록 한다.
② 용량산정은 전체적인 데이터베이스에 발생되는 트랜잭션의 유형과 양을 분석하는 자료가 되므로 성능데이터 모델링을 할 때 중요한 작업이 될 수 있다.
③ 물리적인 데이터 모델링을 할 때 PK/FK의 칼럼의 순서조정, FK인덱스 생성 등은 성능 향상을 위한 데이터 모델링 작업에 중요한 요소가 된다.
④ 이력데이터는 시간에 따라 반복적으로 발생이 되기 때문에 대량 데이터일 가능성이 높아 특별히 성능을 고려하여 컬럼 등을 추가하도록 설계해야 한다.

4. 아래와 같은 보관금원장 엔터티에서 관서에 대한 정보가 반정규화 되어 있기 때문에 관서정보를 조회할 때 성능저하가 발생하고 있다. 이 엔터티에 대해 몇차 정규화가 필요한지와 분리된 스키마 구조를 가장 바르게 짝지은 것은?

- 아래 -
[보관금원장]

관서번호
납부자번호
관리점번호
관서명
상태
관서등록일자
직급명
통신번호

함수종속성(FD):
{관서번호, 납부자번호} → {직급명, 통신번호}
{관서번호} → {관리점번호, 관서명, 상태, 관서등록일자}

① 2차 정규화 - 정규화테이블(관서번호, 납부자번호, 관리점번호, 관서명, 상태, 관서등록일자}
② 3차 정규화 - 정규화테이블(관서번호, 납부자번호, 관리점번호, 관서명, 상태, 관서등록일자}
③ 2차 정규화 - 정규화테이블(관서번호, 관리점번호, 관서명, 상태, 관서등록일자}
④ 3차 정규화 - 정규화테이블(관서번호, 관리점번호, 관서명, 상태, 관서등록일자}

5. 다음 중 아래와 같이 수강지도 엔터티를 만들었을 때 이에 해당하는 정규형과 정규화의 대상으로 가장 바르게 짝지어진 것은?

- 아래 -
[수강지도]

학번
과목코드
성적
지도교수명
학과명

함수종속성(FD)
1. 학번 ‖ 과목번호 → 성적
2. 학번 → 지도교수명
3. 학번 → 학과명

① 1차 정규형 - 2차 정규화 대상
② 2차 정규형 - 3차 정규화 대상
③ 3차 정규형 - 보이스코드 정규화 대상
④ 보이스코드정규형 - 4차 정규화 대상

6. 다음 중 하나의 테이블의 전체 칼럼 중 자주 이용하는 집중화된 칼럼들이 있을 때 디스크 I/O를 줄이기 위해 해당 칼럼들을 별도로 모아놓는 반정규화 기법으로 가장 적절한 것은?

① 칼럼추가 - 부분칼럼추가
② 칼럼추가 - 중복칼럼추가
③ 테이블추가 - 중복테이블추가
④ 테이블추가 - 부분테이블추가

7. 다음 중 칼럼에 대한 반정규화 기법으로 가장 부적절한 것은?

① 중복칼럼추가 - 조인감소를 위해 여러 테이블에 동일한 칼럼을 갖도록 한다.
② 파생칼럼추가 - 조회 성능을 우수하게 하기 위해 미리 계산된 칼럼을 갖도록 한다.
③ FK에 대한 속성 추가 - FK관계에 해당하는 속성을 추가하여 조인성능을 높인다.
④ 이력테이블에 기능 칼럼을 추가 - 최신값을 처리하는 이력의 특성을 고려하여 기능성 칼럼 추가.

8. 아래 설명에서 데이터 액세스 성능을 향상시키기 위해 적용하는 방법에 대해서 괄호 안을 채우시오.

- 아래 -
하나의 테이블에 많은 양의 데이터가 저장되면 인텍스를 추가하고 테이블을 몇 개로 쪼개도 성능이 저하되는 경우가 있다. 이때 논리적으로는 하나의 테이블이지만 물리적으로는 여러 개의 테이블로 분리하여 데이터 액세스 성능도 향상시키고, 데이터 관리방법도 개선할 수 있도록 테이블에 적용하는 기법을 ( ) 이라고 한다.