

Q1. 데이터베이스의 중요성을 설명하라.

A1. 데이터베이스는 데이터 중복성(redundancy)의 제어 및 중복의 최소화, 제어된 중복과 데이터 불일치, 보안 기능, 지속성 기억 공간 제공, 효율적 질의처리를 위한 저장 구조 제공, 백업(backup)과 회복(recovery)기능 제공, 다수의 사용자 인터페이스 제공, 데이터 간 복잡한 관계의 체계적 표현, 데이터베이스의 무결성(integrity) 제약 조건의 시행, 규칙을 사용한 추론과 수행, 동시성 제어 기능 제공, 데이터 독립성 등을 제공하기 때문에 데이터베이스는 중요하다.

Q2. 개체(entity)와 속성(attribute)을 구분해서 설명하라.

A2. 개체(entity)란 DB가 저장하는 유형, 무형의 정보 대상으로, 데이터베이스에서 단독으로 존재 가능하며 정보로서의 역할이 가능하다. 이 개체는 둘 이상의 속성으로 구성된다. 속성(attribute)이란 개체의 특성이나 상태를 나타내는 요소로, 이름을 가진 정보의 가장 작은 논리단위지만, 단독으로는 아무런 정보도 제공하지 못한다.

Q3. 데이터베이스에 대한 내용에 의한 참조와 주소에 의한 참조를 설명하고 일반 사용자의 입장에서 평가해보라.

A3. 내용에 의한 참조란 데이터간의 관계를 구성할 때, 데이터 내부의 내용을 기반으로 한다. 이는 데이터의 내용을 비교해야하기 때문에 일반적으로 처리 비용이 더 크고 성능에 영향을 줄 수 있다. 주소에 의한 참조란 데이터의 위치나 주소를 기반으로 데이터 간의 관계를 구성한다. 데이터의 위치 정보를 사용하기 때문에 내용 비교 없이 빠르게 데이터를 연결할 수 있으며, 성능에 이점을 제공한다.

일반 사용자의 입장에서 이런 참조 방식을 평가해보면, 내용에 의한 참조는 데이터를 더욱 정확하고 유용하게 검색할 수 있는 반면에 처리 시간이 필요하므로 대량의 데이터 처리를 할 때에는 성능에 영향을 줄 수 있다. 주소에 의한 참조는 데이터의 비교 없이 주소만으로 빠르게 연산이 가능하기 때문에 대량의 데이터 처리에도 효과적이다. 따라서 사용자는 빠르고 효율적인 데이터 액세스와 처리를 기대할 수 있다.

Q4. 3단계 데이터베이스 구조에 대해서 설명하라.

A4. 3단계 데이터베이스 구조는 외부 단계(사용자 관점), 개념 단계(조직 전체 관점), 내부 단계(저장장치 관점)으로 나뉜다. 외부 단계에서는 최종 사용자에게 보여지는 외부 스키마가 1...N개 만큼 있으며, 개별 사용자 관점에서 이해하고 표현한다.

개념 단계(조직 전체 관점)에서는 데이터베이스를 조직 전체의 관점에서 이해하고 표현한다. 데이터베이스 하나에 개념 스키마는 한개만 존재하며, 개념 단계에서 데이터베이스 전체의 논리적 구조를 정의한다. 개체 관계, 데이터 타입, 관계, 제약조건 등을 나타내는데 중점을 두고 있다.

내부 단계(저장장치 관점)에서는 데이터베이스를 저장장치 관점에서 이해하고 표현한다. 전체 데이터베이스가 저장 장치에 실제로 저장되는 방법을 정의하며, 레코드 구조, 필드 크기, 레코드 접근 경로 등의 물리적인 저장 구조를 나타낸다.

Q5. 개념적 데이터 모델과 논리적 데이터 모델, 그리고 물리적 데이터 구조와의 관계를 설명해 보라.

A5. 개념적 데이터 모델은 현실 세계의 중요 데이터를 추출 후 개념 세계로 옮겨 모델링한 것을 말한다. 이때 중요 데이터는, 추상화(Abstraction)를 통해 DB에 저장하여 관리할 만한 가치가 있는 데이터를 말한다.

논리적 데이터 모델은 개념 세계의 데이터를 DB에 저장할 구조로 표현한 것이다. DB에 어떤 구조로 저장할지 결정한다.

물리적 데이터 구조는 디스크에 데이터가 저장될 수 있도록 논리적 데이터 모델을 물리적 데이터 구조로 변화시킨 것을 말한다.

개념적 데이터 모델과 논리적 데이터 모델은 현실세계의 정보를 데이터베이스에 저장할 수 있게 가공하는 과정과 관련이 있고, 논리적 데이터 모델과 물리적 데이터 구조는 데이터베이스에 저장된 데이터를 실제 물리적 서버에 저장하는 과정과 관련이 되어있어 3가지는 데이터베이스를 설계하고 데이터를 저장하는데 꼭 필요한 것들 이다.

Q6. SQL에서 뷰는 어떻게 정의되는가? 뷰를 갱신할 때 일어나는 문제점은 무엇인지 설명하라.

A6. SQL에서 뷰는 다른 테이블들에서 유도된 “가상”테이블로 정의된다. 따라서 실제로는 이곳에 데이터가 저장되지는 않는다. 뷰는 갱신(삽입, 삭제) 연산이 제한되는데, 물리적인 형태로 저장되지 않기 때문에 뷰는 일반적인 Alter문으로 변경할 수 없다.

Q7. 다음과 같은 릴레이션이 있을 때 아래 질의문을 SQL로 표현해보라.

학생(학번, 이름, 학년)

과목(과목 번호, 과목 이름, 교수)

등록(과목 번호, 학번, 점수)

1) 과목 'DBMS'를 수강한 4학년 학생의 이름을 검색하라.

A7-1. `SELECT `이름` FROM `학생` WHERE `학번` IN (SELECT `학번` FROM `등록` WHERE `과목 번호` IN (SELECT `과목 번호` FROM `과목` WHERE `과목 이름` = 'DBMS'));`

2) 'DBMS' 과목을 삭제하라.

A7-2. `DELETE FROM `과목` WHERE `과목 이름` = 'DBMS';`

3) 'AI' 과목을 과목 릴레이션에 삽입하라. (단, 과목 번호는 'C419'이다.)

A7-3. `INSERT INTO `과목` VALUES('C419', 'AI', '');`

4) 학번이 12345인 학생의 학년을 4로 변경하라.

A7-4. `UPDATE `학생` SET `학년` = 4 WHERE `학번` = 12345;`

Q8. 널값은 무엇이고 어떤 경우에 사용되는가?

A8. 널값은 알려지지 않은값, 이용할 수 없거나 보류해둔 값 또는 적용할 수 없는 속성인 경우에 사용된다.

Q9. $R \bowtie S$ 와 $S \bowtie R$ 은 같아야 되는가? 어떤 조건에서 $R \bowtie S = S \bowtie R$ 인가 설명하라.

A9. $R \bowtie S = A$, $S \bowtie R = B$ 라 할 때, 일반적으로 A와 B는 같지 않다. A는 R과 S를 조인하여 R의 속성과 S의 속성을 모두 포함하는 새로운 테이블을 생성하지만, B의 경우 S의 속성과 R의 속성을 모두 포함하는 새로운 테이블을 생성하기 때문에 R과 S의 속성의 순서가 다르면 결과 테이블이 달라지게 된다.

하지만, R과 S의 속성의 순서가 동일하다면, A와 B가 같아질 수 있다.

Q10. 참조 무결성을 구현하는 방법을 설명하라.

A10. 왜래 키 제약조건을 추가하면 참조 무결성을 강제로 구현할 수 있다.