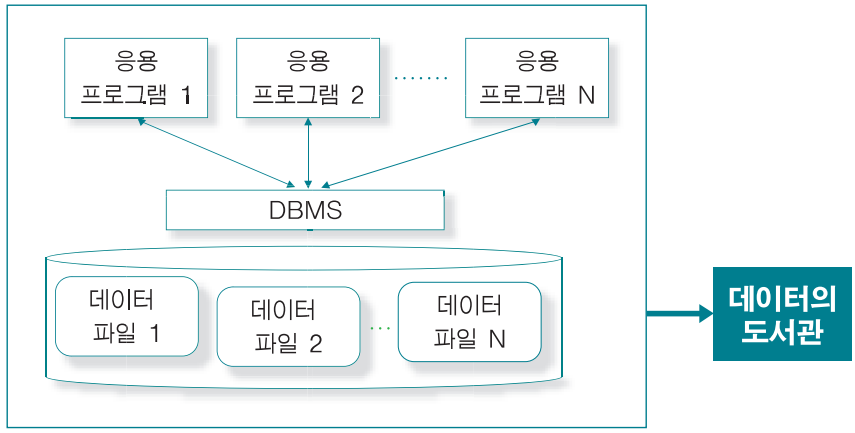
S/E/C/T/I/O/N

01 데이터베이스 실무 개요

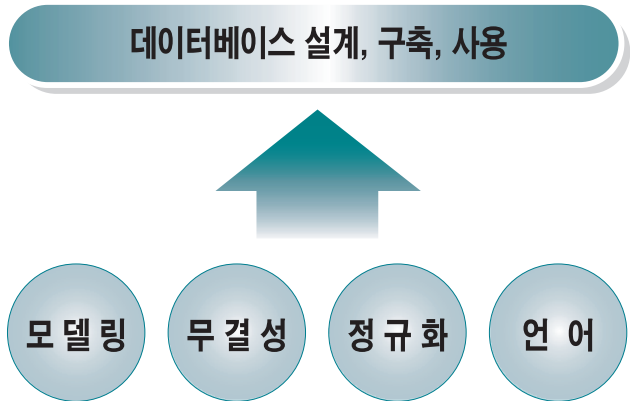
:01 데이터베이스의 개념

■ 데이터베이스의 정의

– 관련된 데이터들의 모임
– 조직체의 응용 시스템들이 공유해서 사용하는 운영 데이터들이 구조적으로 통합된 모임
– 운영, 통합, 저장, 공유

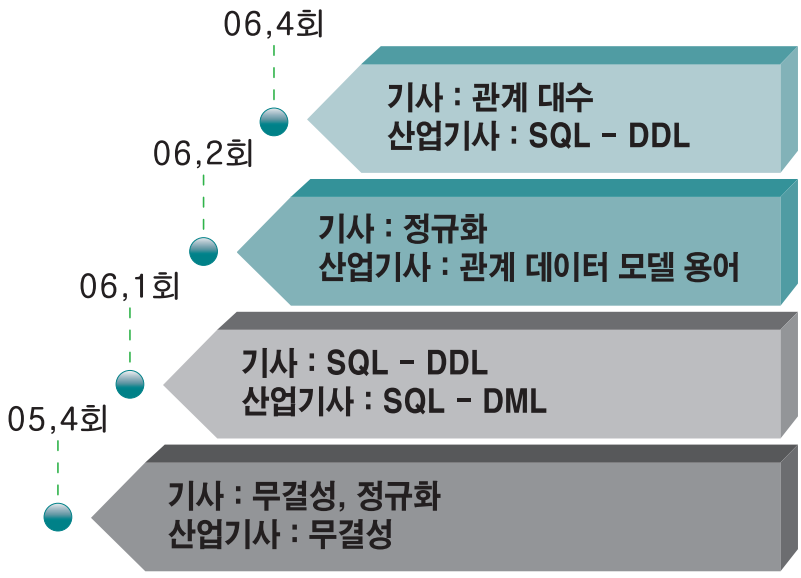


:02 시험 출제 영역




모델링
· 개념적 모델링
· 논리적 모델링
· 물리적 모델링
무결성
· 무결성 제약 조건
· 키
정규화
· 이상 현상
· 함수 종속성
· 정규화
언 어
· 관계 대수
· SQL

: 03 기출문제 분석


06,4회
기사 : 관계 대수
산업기사 : SQL - DDL
06,2회
기사 : 정규화
산업기사 : 관계 데이터 모델 용어
06,1회
기사 : SQL - DDL
산업기사 : SQL - DML
05,4회
기사 : 무결성, 정규화
산업기사 : 무결성

: 04 문제유형 분석 및 학습전략

1. 문제유형 I

먼저, 데이터베이스의 논리 데이터 모델과 물리적으로 실제 구현된 내역에 대해 상세하게 검토한 결과 시스템의 분석 및 설계 단계에서 작성한 논리 데이터 모델에는 개체들 간의 관계를 설정하여 하위 개체에 (❶)이(가) 설계되었이지만 물리적 구현 단계어서는 논리 데이터 모델 단계에서 정의하였던 (❶)(가) 물리 데이터베이스에 실제적으로 구현되어 있지 않아 결과적으로 테이블들이 고립(Isolate)되어 독립적인 상태로 존재하고 있음이 인지되었다. 데이터베이스로 구축되어 있는 물리 테이블 간에는 논리적인 상호 연관 관계가 설정되어 있어야 하며, 상위 개체의 주 키와 하위 개체에 존재하는 (❶)의 연결과 관련된 (❷)이(가) 확보되어 데이터의 입력, 수정 및 삭제 등이 정상적으로 이루어질 수 있도록 관리되어야 한다. 이러한 제약조건이 지켜지지 않는다면 데이터베이스내에 존재하는 데이터들의 일관성(Consistency)을 보장할 수 없게 된다. 그뿐만 아니라 데이터의 이상현상(Anomaly)도 발생하게 된다.

또한 일반적으로 개체 관계도에 표현되는 모든 개체들은 반드시 한번 이상의 관계를 다른 개체들과 유지해야 한다. 관계 설정이 없이 독립적으로 개체가 존재하다는 것은 바람직한 현상이 아니므로, 상위 개체에서는 주 키이면서 하위 테이블에서는 일반 (❸)으(로) 지정이 되어 있는 데이터 항목은 데이터 자체의 무결성 유지를 위해 ()(으)로 정의를 내려 사용하는 등 해당 시스템의 분석 및 설계 단계를 거치며 정의하였던 (❶)을(를) 적절한 수준에서 물리적으로 구현할 필요가 있다. 하지만, 해당 시스템에서는 (❷)이(가) 저해되어 데이터베이스가 특정 업무 수행을 위하여 개발자들에 의해 개발된 시스템을 의미하는 (❺)에 전적으로 의존하게 되는 등, 데이터 독립성이 저하됨으로써 개발자들에 의해 개발된 (❺)에서 오류가 발생하거나 데이터베이스 관리 도구에서 데이터를 직접 입력할 때에 잘못된 데이터에 대한 대처가 용이하지 않게 될 것으로 예상되었다.

한편, 〈표 1〉에서 일부 예를 나타내 바와 같이 (❹)에 위배된 사례도 다수 인지되었다. 이는 데이터베이스와 관련된 무결성 제약 조건 중에서 속성(칼럼)에 대하여 정의하는 물결성을 의미한다. 전체 시스템에 존재하는 45개의 테이블에서 총 389개의 (❸)을(를) 분석한 결과 명명규칙 측면에서 (❸) ID는 같으나 (❸)명이 다른 경우 또는 그 반대의 경우가 무척 많이 발견되었으며, 각 데이터 테이블 내의 속성 항목들을 검토한 결과 동일한 속성에 대하여 데이터 유형 및 데이터 길이가 다르게 정의되는 것과 같은 (❹)의 위배사례가 많이 발견되었다.

입력된 데이터가 일관성을 유지하도록 하기 위해서는 데이터베이스 설계 이전에 데이터베이스와 관련된 표준 명명 규칙을 제정하여 동일한 (❸)명에 대해서는 모든 테이블에서 동일한 데이터 유형이 유지되도록 하여 (❹)을(를) 확보할 수 있도록 해 주어야 한다. 특히 상용 DBMS내부에서 다른 데이터 유형으로 인식을 하는 (❸)명일 경우에는 데이터 JOIN 연산시에 Full Table Scan을 하게 될 수 있으므로 명칭을 일치시키는 것이 필요하다. 이는 데이터베이스 전체 성능과도 관련이 있는 문제이므로 설계서에서 정의하지 않은 값이 설정된 사례를 모두 조사하여 대응책을 강구하는 것이 요구되었다.

답항보기

1	DCL	2	개체무결성	3	이상 현상	4	관계	5	BCNF
6	대체 키	7	데이터공유성	8	카티션 프로덕트	9	연산 (operations)	10	데이터일관성
11	관계해석	12	데이터종속성	13	데이터중복성	14	데이터투명성	15	DBA
16	DML	17	데이터베이스 관리시스템	18	도메인무결성	19	레코드	20	뷰(view)
21	구조 (structure)	22	concurrency control	23	정규화	24	외래 키	25	P.Chen
26	관계대수	27	응용 시스템	28	DDL	29	응용 키	30	일관성
31	locking	32	참조무결성	33	칼럼	34	파일	35	트랜잭션
36	Query Processor	37	cardinality	38	후보 키	39	해싱	40	SQL 튜닝

(1) 암기가 아닌 개념 이해 위주의 학습 (필기와는 다른 접근이 필요)

[예]

- 필기 → 외래키 : 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합
- 실기 → 외래키를 통해 개체들이 연결되어 관계를 맺게 된다.
  외래키가 제대로 설정되지 않은 경우, 연결되어야 할 개체들이 독립적인 상태로 격리되어 참조 무결성이 위배될 수 있다.

용어를 다양하게 풀어서 설명
→ 단순 암기 불필요

먼저 이해하고,
그리고 꼭 필요한 것만 암기!

(1) 용어의 다양한 표현 – 동의어 정리

릴레이션 (❶)은 릴레이션의 구조를 나타내고 있습니다.
이 안에는 릴레이션의 이름, 각 속성의 이름과 타입, 그리고 속성 값의 도메인을 정의하는 릴레이션에 데이터를 넣을 수 있도록 하는 릴레이션 틀을 말한다.

2. 문제유형 II

- Student, Coures, Lnstructor 릴레이션의 id 속성은 기본키(primary key)이고, name 속성 값은 입력하지 않는 경우 오류가 발생한다. 또한 부모 릴레이션의 수정이 발생하면 자식 릴레이션의 외래키 값이 부모 릴레이션의 기본키 속성 값으로 연쇄적으로 수정된다.
- Student 릴레이션에서 ssn 속성의 값은 같은 행은 존재할 수 없고 major 속성은 Department 릴레이션을 참조한다.

```
CREATE TABLE Student {
        id          CHAR(8)       NOT NULL,
        name        CHAR(13)      NOT NULL,
        ssn         CHAR(10)      NOT NULL,
        major       CHAR(15),
UNIQUE(ssn)
PRIMARY KEY(②),
FOREIGN KEY(major) REFERENCE Department(①)
    ON DELE TE SET NULL ON UPDA TE ③
```

SQL과 관계 대수 문법 암기