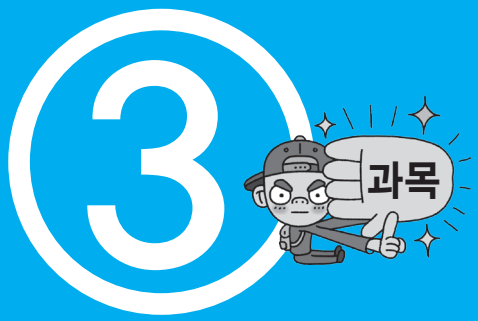


데이터베이스 구축

1장 / 논리 데이터베이스 설계

2장 / 물리 데이터베이스 설계

3장 / SQL 응용

4장 / SQL 활용

5장 / 데이터 전환

구독!
베리나이쓰~

1장

논리 데이터베이스 설계

070 데이터베이스 설계 A 등급
071 데이터 모델의 개념 B 등급
072 데이터 모델의 구성 요소 – 개체(Entity) A 등급
073 데이터 모델의 구성 요소 – 속성(Attribute) B 등급
074 데이터 모델의 구성 요소 – 관계(Relationship) B 등급
075 식별자(Identifier) C 등급
076 E-R(개체-관계) 모델 A 등급
077 관계형 데이터 모델 C 등급
078 관계형 데이터베이스의 구조 A 등급
079 관계형 데이터베이스의 제약 조건 – 키(Key) A 등급
080 관계형 데이터베이스의 제약 조건 – 무결성 B 등급
081 관계대수 및 관계해석 A 등급
082 정규화(Normalization) A 등급
083 반정규화(Denormalization) B 등급
084 시스템 카탈로그 A 등급

이 장에서 꼭 알아야 할 키워드 Best 10

1. 개체 **2.** 속성 **3.** 관계 **4.** 튜플 **5.** 도메인 **6.** 키 **7.** 무결성 **8.** 관계대수 **9.** 정규화 **10.** 시스템 카탈로그

SECTION 070 ● LIVE

데이터베이스 설계

전문가의 조언

데이터베이스 설계에서는 설계 순서와 각 단계별 특징이 중요합니다. 데이터베이스 설계 순서 및 각 단계에서 대표적으로 수행해야 할 작업을 중심으로 학습하세요.

1 데이터베이스 설계의 개념

데이터베이스 설계란 사용자의 요구를 분석하여 그것들을 컴퓨터에 저장할 수 있는 데이터베이스의 구조에 맞게 변형한 후 특정 DBMS로 데이터베이스를 구현하여 일반 사용자들이 사용하게 하는 것이다.

2 데이터베이스 설계 시 고려사항

- **무결성** : 삽입, 삭제, 갱신 등의 연산 후에도 데이터베이스에 저장된 데이터가 정해진 제약 조건을 항상 만족해야 한다.
- **일관성** : 데이터베이스에 저장된 데이터들 사이나, 특정 질의에 대한 응답이 처음부터 끝까지 변함없이 일정해야 한다.
- **회복** : 시스템에 장애가 발생했을 때 장애 발생 직전의 상태로 복구할 수 있어야 한다.
- **보안** : 불법적인 데이터의 노출 또는 변경이나 손실로부터 보호할 수 있어야 한다.
- **효율성** : 응답시간의 단축, 시스템의 생산성, 저장 공간의 최적화 등이 가능해야 한다.
- **데이터베이스 확장** : 데이터베이스 운영에 영향을 주지 않으면서 지속적으로 데이터를 추가할 수 있어야 한다.

3 데이터베이스 설계 순서

요구 조건 분석	요구 조건 명세서 작성
↓	
개념적 설계	개념 스키마, 트랜잭션 모델링, E-R 모델
↓	
논리적 설계	목표 DBMS에 맞는 논리 스키마 설계, 트랜잭션 인터페이스 설계
↓	
물리적 설계	목표 DBMS에 맞는 물리적 구조의 데이터로 변환
↓	
구현	목표 DBMS의 DDL(데이터 정의어)로 데이터베이스 생성, 트랜잭션 작성

4 요구 조건 분석

요구 조건 분석은 데이터베이스를 사용할 사람들로부터 필요한 용도를 파악하는 것이다.

- 데이터베이스 사용자에 따른 수행 업무와 필요한 데이터의 종류, 용도, 처리 형태, 흐름, 제약 조건 등을 수집한다.
- 수집된 정보를 바탕으로 요구 조건 명세를 작성한다.

5 개념적 설계(정보 모델링, 개념화)

개념적 설계란 정보의 구조를 얻기 위하여 현실 세계의 무한성과 계속성을 이해하고, 다른 사람과 통신하기 위하여 현실 세계에 대한 인식을 추상적 개념으로 표현하는 과정이다.

- 개념적 설계 단계에서는 개념 스키마 모델링과 트랜잭션 모델링을 병행 수행한다.
- 개념적 설계 단계에서는 요구 분석 단계에서 나온 결과인 요구 조건 명세를 DBMS에 독립적인 E-R 다이어그램으로 작성한다.
- DBMS에 독립적인 개념 스키마를 설계한다.

6 논리적 설계(데이터 모델링)

논리적 설계 단계란 현실 세계에서 발생하는 자료를 컴퓨터가 이해하고 처리할 수 있는 물리적 저장장치에 저장할 수 있도록 변환하기 위해 특정 DBMS가 지원하는 논리적 자료 구조로 변환시키는 과정이다.

- 개념 세계의 데이터를 필드로 기술된 데이터 타입과 이 데이터 타입들 간의 관계로 표현되는 논리적 구조의 데이터로 모델화한다.
- 개념적 설계가 개념 스키마를 설계하는 단계라면 논리적 설계에서는 개념 스키마를 평가 및 정재하고 DBMS에 따라 서로 다른 논리적 스키마를 설계하는 단계이다.
- 트랜잭션의 인터페이스를 설계한다.
- 관계형 데이터베이스라면 테이블을 설계하는 단계이다.

7 물리적 설계(데이터 구조화)

물리적 설계란 논리적 설계 단계에서 논리적 구조로 표현된 데이터를 디스크 등의 물리적 저장장치에 저장할 수 있는 물리적 구조의 데이터로 변환하는 과정이다.

- 물리적 설계 단계에서는 다양한 데이터베이스 응용에 대해 처리 성능을 얻기 위해 데이터베이스 파일의 저장 구조 및 액세스 경로를 결정한다.
- 저장 레코드의 형식, 순서, 접근 경로와 같은 정보를 사용하여 데이터가 컴퓨터에 저장되는 방법을 묘사한다.

8 데이터베이스 구현

데이터베이스 구현 단계란 논리적 설계 단계와 물리적 설계 단계에서 도출된 데이터베이스 스키마를 파일로 생성하는 과정이다.

- 사용하려는 특정 DBMS의 DDL(데이터 정의어)을 이용하여 데이터베이스 스키마를 기술한 후 컴파일하여 빈 데이터베이스 파일을 생성한다.
- 생성된 빈 데이터베이스 파일에 데이터를 입력한다.
- 응용 프로그램을 위한 트랜잭션을 작성한다.
- 데이터베이스 접근을 위한 응용 프로그램을 작성한다.

따라잡기

기출문제 따라잡기 Section 070

이전기출

1. 데이터베이스 설계 순서로 옳은 것은?

① 요구 조건 분석 → 개념적 설계 → 논리적 설계 → 물리적 설계 → 구현

② 요구 조건 분석 → 논리적 설계 → 개념적 설계 → 물리적 설계 → 구현

③ 요구 조건 분석 → 논리적 설계 → 물리적 설계 → 개념적 설계 → 구현

④ 요구 조건 분석 → 개념적 설계 → 물리적 설계 → 논리적 설계 → 구현

두말하면 잔소리죠! 반드시 암기하세요.

이전기출

2. 데이터베이스 설계 시 요구 분석 단계에서 나온 결과(명세)를 E-R 다이어그램과 같은 DBMS에 독립적이고 고차원적인 표현 기법으로 기술하는 것을 무엇이라고 하는가?

① 개념 스키마 모델링

② 트랜잭션 모델링

③ 물리적 설계

④ 계층 데이터베이스 모델링

개념 스키마는 현실 세계에 대한 인식을 추상적 개념으로 표현하는 과정으로, 특정 DBMS와는 무관합니다. 그리고 'E-R 모델링'하면 개념 설계라는 것도 잊지마세요.

이전기출

3. 데이터베이스 설계 단계와 그 단계에서 수행되는 결과의 연결이 잘못된 것은?

① 개념적 설계 단계 – 트랜잭션 모델링

② 물리적 설계 단계 – 목표 DBMS에 맞는 물리적 구조 설계

③ 논리적 설계 단계 – 목표 DBMS에 독립적인 논리 스키마 설계

④ 구현 단계 – 목표 DBMS DDL로 스키마 작성

논리적 설계 단계는 목표 DBMS 맞는, 즉 독립적인이 아니라 종속적인 논리 스키마를 설계하는 단계입니다.

이전기출

4. 데이터베이스 설계 과정 중 개념적 설계 단계에 대한 설명으로 틀린 것은?

① 산출물로 ER-D가 만들어진다.

② DBMS에 독립적인 개념 스키마를 설계한다.

③ 트랜잭션 인터페이스를 설계한다.

④ 논리적 설계 단계의 앞 단계에서 수행된다.

데이터베이스 설계 순서가 가장 중요하지만 각 단계를 구분할 수 있는 특징 정도는 알고 있어야 한다고 했죠? 트랜잭션 인터페이스를 설계하는 단계는 논리적 설계 단계입니다.

따라잡기

기출문제 따라잡기

Section 070

이전기출

5. 다음에서 설명하는 데이터베이스 설계 단계는?

- 데이터베이스 관리 시스템(DBMS)에 따라 서로 다른 논리적 스키마를 정의한다.
- 현실 세계를 표현하기 위한 데이터베이스의 논리적 구조, 즉 정규화 과정을 이용한 릴레이션의 속성을 결정하는 단계이다.

① 요구 사항 및 분석 단계
② 개념적 설계 단계
③ 논리적 설계 단계
④ 물리적 설계 단계

'논리적 스키마'하면, 논리적 설계가 생각나야 합니다.

이전기출

6. 데이터베이스 설계의 논리적 설계 단계에서 수행하는 작업이 아닌 것은?

① 논리적 데이터 모델로 변환
② 트랜잭션 인터페이스 설계
③ 스키마의 평가 및 정제
④ 트랜잭션 모델링

트랜잭션의 모델링은 개념적 설계 단계에서 수행합니다.

이전기출

7. 데이터베이스 설계 단계 중 목표 DBMS의 DDL로 스키마를 작성하여 데이터베이스에 등록하고 응용 프로그램을 위한 트랜잭션을 작성하는 단계는?

① 논리적 설계
② 물리적 설계
③ 구현
④ 운영

목표 DBMS의 DDL로 스키마를 작성하여 데이터베이스에 등록한다는 의미는 실제로 데이터베이스 파일을 생성한다는 의미입니다. 실제로 무엇인가가 컴퓨터 파일로 만들어지는 단계는 어느 단계일까요? 확인해 보세요.

이전기출

8. 데이터베이스 설계 시 고려 사항으로 적합하지 않은 것은?

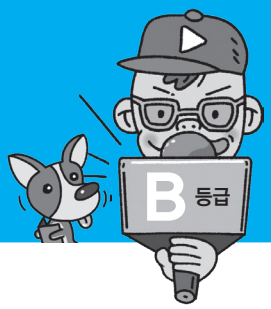
① 데이터 무결성 유지
② 데이터 일관성 유지
③ 데이터 보안성 유지
④ 데이터 종속성 유지

데이터베이스 설계 시에는 데이터의 종속성 유지가 아니라 데이터의 논리적, 물리적 독립성을 고려해야 합니다.

▶ 정답 : 1. ① 2. ① 3. ③ 4. ③ 5. ③ 6. ④ 7. ③ 8. ④

SECTION 071 ● LIVE

데이터 모델의 개념



전문가의 조언

데이터 모델에서는 개념이나 정의가 중요합니다. 데이터 모델의 정의를 숙지하고, 개념적 데이터 모델과 논리적 데이터 모델을 구분할 수 있도록 학습하세요.

1 데이터 모델의 정의

데이터 모델은 현실 세계의 정보들을 컴퓨터에 표현하기 위해서 단순화, 추상화하여 체계적으로 표현한 개념적 모형이다.

- 데이터 모델은 데이터, 데이터의 관계, 데이터의 의미 및 일관성, 제약조건 등을 기술하기 위한 개념적 도구들의 모임이다.
- 현실 세계를 데이터베이스에 표현하는 중간 과정, 즉 데이터베이스 설계 과정에서 데이터의 구조(Schema)를 논리적으로 표현하기 위해 사용되는 지능적 도구이다.
- **데이터 모델 구성 요소** : 개체, 속성, 관계
- **데이터 모델 종류** : 개념적 데이터 모델, 논리적 데이터 모델, 물리적 데이터 모델*
- **데이터 모델에 표시할 요소** : 구조, 연산, 제약 조건

물리적 데이터 모델에 대한 자세한 내용은 Section 085를 참조하세요.

전문가의 조언

데이터 모델의 구성 요소에는 개체, 속성, 관계가 있습니다. 개체, 속성, 관계는 데이터베이스의 기본 구성 요소로서 데이터베이스 과목을 공부하는 동안 항상 나오는 용어입니다. 반드시 숙지하고 넘어가세요.

2 데이터 모델의 구성 요소

- **개체(Entity)** : 개체는 데이터베이스에 표현하려는 것으로, 사람이 생각하는 개념이나 정보 단위 같은 현실 세계의 대상체이다.
- **속성(Attribute)** : 속성은 데이터의 가장 작은 논리적 단위로서 파일 구조상의 데이터 항목 또는 데이터 필드에 해당한다.
- **관계(Relationship)** : 관계는 개체 간의 관계 또는 속성 간의 논리적인 연결을 의미한다.

전문가의 조언

개념적 데이터 모델과 논리적 데이터 모델의 특징을 구분할 수 있도록 학습하세요.

3 개념적 데이터 모델

개념적 데이터 모델은 현실 세계에 대한 인간의 이해를 돕기 위해 현실 세계에 대한 인식을 추상적 개념으로 표현하는 과정이다.

- 개념적 데이터 모델은 속성들로 기술된 개체 타입과 이 개체 타입들 간의 관계를 이용하여 현실 세계를 표현한다.
- 개념적 데이터 모델은 현실 세계에 존재하는 개체를 인간이 이해할 수 있는 정보 구조로 표현하기 때문에 정보 모델이라고도 한다.
- 대표적인 개념적 데이터 모델로는 E-R 모델이 있다.

4 논리적 데이터 모델

논리적 데이터 모델은 개념적 모델링 과정에서 얻은 개념적 구조를 컴퓨터가 이해하

고 처리할 수 있는 컴퓨터 세계의 환경에 맞도록 변환하는 과정이다.

- 논리적 데이터 모델은 필드로 기술된 데이터 타입과 이 데이터 타입들 간의 관계를 이용하여 현실 세계를 표현한다.
- 단순히 데이터 모델이라고 하면 논리적 데이터 모델을 의미한다.
- 특정 DBMS는 특정 논리적 데이터 모델 하나만 선정하여 사용한다.
- 논리적 데이터 모델은 데이터 간의 관계를 어떻게 표현하느냐에 따라 관계 모델, 계층 모델, 네트워크 모델로 구분한다.

5 논리적 데이터 모델의 품질 검증

논리 데이터 모델 품질 검증은 완성된 논리 데이터 모델이 기업에 적합한지를 확인하기 위해 품질을 검증하는 것이다.

- 논리 데이터 모델의 품질은 논리 데이터 모델 품질 기준*에 따라 개체, 속성, 관계, 식별자, 모델 전반 등에 대하여 검토 체크리스트를 작성하고 체크리스트의 각 항목을 확인하는 방식으로 검증한다.
- **개체 품질 검증 항목** : 단수 명사 여부, 개체의 주 식별자, 개체 간 상호 배타성, 개체의 정규화 여부, 개체 상세 정의, 개체 관리 업무 기능, 개체에 2개 이상의 속성 존재 여부, 개체의 총 길이, 개체 동의어 여부, 개체 분산 요구 등
- **속성 품질 검증 항목** : 단수 명사 여부, 속성의 값 존재 여부 및 개수, 도메인 정의, 반복되는 속성, 그룹화 가능 속성, 주 식별자 및 비 식별자에 의존하는 속성, 다치 종속* 속성 등
- **관계 품질 검증 항목** : 관계의 명칭, 2개 이상의 노드와 관계 존재 여부, 노드의 기수성*과 선택성*, 필수적 관계, 유효한 관계, 중복된 관계, 외부식별자 존재 여부, 참조 무결성 여부 등
- **식별자 품질 검증 항목** : 식별자의 명칭, 정의, 구성, 정합성, 크기, 순서 등
- **전반적인 품질 검증 항목** : 주제 영역 구성의 적절성, 데이터 모델 상에 정규화 여부, 다대다 관계 해소 여부, 이력 관리 대상 선정 확인, 이력 관리 방법의 적절성 확인

6 데이터 모델에 표시할 요소

- **구조(Structure)** : 논리적으로 표현된 개체 타입들 간의 관계로서 데이터 구조 및 정적 성질을 표현한다.
- **연산(Operation)** : 데이터베이스에 저장된 실제 데이터를 처리하는 작업에 대한 명세로서 데이터베이스를 조작하는 기본 도구이다.
- **제약 조건(Constraint)** : 데이터베이스에 저장될 수 있는 실제 데이터의 논리적인 제약 조건이다.

논리 데이터 모델 품질 기준
논리 데이터 모델 품질 기준에는 정확성, 완전성, 준거성, 최신성, 일관성, 활용성 등이 있습니다.

다치 종속
A, B, C 3개의 속성을 가진 테이블 R에서 어떤 복합 속성(A, C)에 대응하는 B 값의 집합이 A 값에만 종속되고 C 값에는 무관하면, B는 A에 다치 종속이라 합니다.

기수성(Cardinality)
관계에 참여하는 각 개체에 대하여 가능한 관계의 수를 의미하며, 관계 차수라고도 합니다.
예 1:1, 1:M, M:N

선택성(Optionality)
관계가 필수인지 선택인지를 표현하는 것을 의미하며, 관계선택사양이라고도 합니다.
예 1:1, 1:0

전문가의 조언
정보처리 기사 필기 시험은 문제가 4지 선다형으로 출제되고 있기 때문에 데이터 모델에 표시할 요소는 출제하기 좋은 내용입니다. 구조, 연산, 제약 조건을 암기하세요.

이전기출
1. 데이터 모델(Data Model)의 개념으로 가장 적절한 것은?

① 현실 세계의 데이터 구조를 컴퓨터 세계의 데이터 구조로 기술하는 개념적인 도구이다.
② 컴퓨터 세계의 데이터 구조를 현실 세계의 데이터 구조로 기술하는 개념적인 도구이다.
③ 현실 세계의 특정한 한 부분의 표현이다.
④ 가상 세계의 데이터 구조를 현실 세계의 데이터 구조로 기술하는 개념적인 도구이다.

데이터 모델의 개념, 다시 한 번 볼까요? 데이터 모델은 현실 세계의 정보들을 컴퓨터에 표현하기 위해 단순화, 추상화하여 체계적으로 표현한 개념적 모형입니다.

이전기출
2. 데이터 모델의 구성 요소가 아닌 것은?

① 논리적으로 표현된 데이터 구조
② 구성 요소의 연산
③ 구성 요소의 제약 조건
④ 물리적 저장 구조

데이터 모델의 구성 요소라고 하기보다는 표현해야 할 요소라고 하는 것이 더 가깝습니다. 논리적 구조(Structure), 연산(Operation), 제약 조건(Constraint)! 기억하세요.

이전기출
3. 데이터의 가장 작은 논리적 단위로서 파일 구조상의 데이터 항목 또는 데이터 필드에 해당하는 것은?

① Tuple ② Relation
③ Domain ④ Attribute

개체와 속성의 구분을 명확히 해야 합니다. 개체는 한 개 이상의 속성(Attribute)으로 이루어집니다. 그리고 속성은 데이터베이스 구성의 가장 작은 논리적인 단위라는 것도 잊으면 안 됩니다.

이전기출
4. 데이터 모델에 표시할 요소 중 데이터베이스에 표현될 대상으로서의 개체 타입과 개체 타입들 간의 관계를 기술한 것은?

① Structure ② Operations
③ Constraints ④ Mapping

'구조, 연산, 제약조건'은 물론 각각의 의미도 기억해야 합니다.

이전기출
5. 데이터 모델에 표시할 요소 중 데이터베이스에 표현된 개체 인스턴스를 처리하는 작업에 대한 명세로서 데이터베이스를 조작하는 기본 도구에 해당하는 것은?

① Operation ② Constraint
③ Structure ④ Relationship

구조, 연산, 제약조건! 각각의 의미도 기억하라고 했죠?

이전기출
6. 데이터 모델에 대한 다음 설명 중 ( ) 안에 공통으로 들어갈 내용으로 가장 타당한 것은?

> 데이터 모델은 일반적으로 3가지 구성 요소를 포함하고 있다. 첫째, 논리적으로 표현된 데이터 구조, 둘째, 이 구조에서 허용 될 수 있는 ( ), 셋째, 이 구조와 ( )에서의 제약 조건에 대한 명세를 기술한 것이다.

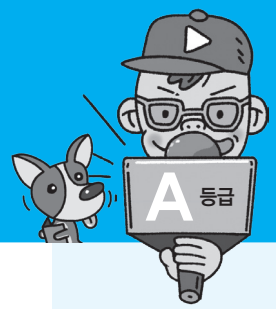
① 개체 ② 연산
③ 속성 ④ 도메인

이 문제도 사실 구조, 연산, 제약 조건만 외우고 있으면 맞힐 수 있는 문제죠.

▶ 정답 : 1. ① 2. ④ 3. ④ 4. ① 5. ① 6. ②

SECTION 072 ● LIVE

데이터 모델의 구성 요소 – 개체(Entity)



1 개체의 정의 및 특징

개체(Entity)는 데이터베이스에 표현하려는 것으로, 사람이 생각하는 개념이나 정보 단위 같은 현실 세계의 대상체이다.

- 개체는 실세계에 독립적으로 존재하는 유형, 무형의 정보*로서 서로 연관된 몇 개의 속성*으로 구성된다.
- 파일 시스템의 레코드에 대응하는 것으로 어떤 정보를 제공하는 역할을 수행한다.
- 영속적(Persistence)으로 존재하는 개체의 집합이다.
- 독립적으로 존재하거나 그 자체로서도 구별이 가능하다.
- 유일한 식별자(Unique Identifier)에 의해 식별이 가능하다.
- 개체는 업무 프로세스에 의해 이용된다.
- 다른 개체와 하나 이상의 관계(Relationship)가 있다.

예제 다음은 교수번호, 성명, 전공, 소속으로 구성된 교수 개체이다.

속성

교수번호	성명	전공	소속
144001	조성진	곤충	안산대
145002	박윤영	미생물	강원대
146001	강현준	식물	서울대

개체 타입(레코드 타입)

개체 세트

개체 인스턴스

교수 개체

해설

교수 개체의 구성 요소

- 속성 : 개체가 가지고 있는 특성, 교수번호, 성명, 전공, 소속
- 개체 타입 : 속성으로만 기술된 개체의 정의
- 개체 인스턴스 : 개체를 구성하고 있는 각 속성들이 값을 가져 하나의 개체를 나타내는 것으로 개체 어커런스(Occurence)라고도 한다.
- 개체 세트 : 개체 인스턴스의 집합

2 개체 선정 방법

- 업무 분석에 관한 내용을 구체적으로 설명한 업무 기술서를 이용한다.
- 실제 업무를 담당하고 있는 담당자와 인터뷰를 한다.

전문가의 조언

개체(Entity)는 데이터베이스의 기본 구성 요소로서 데이터베이스 과목을 공부하는 동안 항상 나오는 용어입니다. 반드시 숙지하고 넘어가세요.

유형, 무형의 정보

유형의 정보는 물리적으로 존재하는 사람, 자동차, 집 등을 말하고, 무형의 정보는 개념적으로 존재하는 여행, 음악, 취미 등을 말합니다.

속성에 대한 자세한 내용은 다음 섹션에서 학습합니다.

자료 흐름도(DFD; Data Flow Diagram)
자료 흐름도는 요구사항 분석에서 자료의 흐름 및 변환 과정과 기능을 도형을 중심으로 기술하는 방법으로, 자료 흐름 그래프 또는 버블 차트라고도 합니다.

자료 저장소(Data Store)
자료 저장소는 시스템에서의 자료 저장소(파일, 데이터베이스)를 의미합니다.

BPR(Business Process Reengineering, 업무 프로세스 재설계)
BPR은 기업이 경쟁에서 우위를 확보하기 위해 기업의 핵심 부문인 비용, 품질, 서비스, 스피드와 같은 요인들이 획기적인 향상을 이룰 수 있도록, 프로세스를 기반으로 비즈니스 시스템을 근본적으로 재설계하여 혁신적인 성과를 추구하는 것을 의미합니다.

- 업무 기술서와 인터뷰에서 확인하지 못한 정보가 있는지 실제 업무를 직접 견학하여 확인한다.
- 실제 업무에 사용되고 있는 장부와 전표를 이용한다.
- 이미 구축된 시스템이 있는 경우 해당 시스템의 산출물을 검토한다.
- 자료 흐름도(DFD)*를 통해 업무 분석을 수행했을 경우 자료 흐름도의 자료 저장소(Data Store)*를 이용한다.
- BPR(업무 프로세스 재설계)*에 의해 업무를 재정의한 경우 관련 개체를 찾는다.

3 개체명 지정 방법

- 일반적으로 해당 업무에서 사용하는 용어로 지정한다.
- 약어 사용은 되도록 제한한다.
- 가능하면 단수 명사를 사용한다.
- 모든 개체명은 유일해야 한다.
- 가능하면 개체가 생성되는 의미에 따라 이름을 부여한다.
  예 교수, 고객, 주문, 도시 등

기출문제 따라잡기 Section 072

출제예상
1. 다음 중 개체에 대한 설명으로 가장 옳지 않은 것은?
① 개체는 사람이 생각하는 개념이나 정보 단위 같은 현실 세계의 대상체를 말한다.
② 개체는 유일한 식별자에 의해 식별이 가능해야 한다.
③ 개체는 다른 개체와 하나 이상의 관계(Relationship)가 있어야 한다.
④ 개체에는 속성이 포함되어 있지 않아도 된다.

개체는 서로 연관된 몇 개의 속성으로 구성됩니다.

출제예상
2. 다음 중 개체 선정 방법에 대한 설명으로 옳지 않은 것은?
① 업무와 관련해서 설명한 업무 기술서를 이용한다.
② DFD에 의해 업무를 재정의한 경우 관련 개체를 찾는다.
③ 현업의 장표를 활용한다.
④ 현업 담당자와의 인터뷰를 활용한다.

자료 흐름도(DFD)를 통해 업무 분석을 수행했을 경우에는 자료 흐름도의 자료 저장소(Data Store)를 이용합니다.

출제예상
3. 다음 중 개체명 지정 방법에 대한 설명으로 가장 옳지 않은 것은?
① 일반적으로 현업에서 사용하는 용어를 사용한다.
② 모든 개체명은 유일해야 한다.
③ 가능하면 약어를 사용한다.
④ 가능하면 단수 명사를 사용한다.

개체명을 약어로 지정하면 사람들이 쉽게 이해할 수 있을까요?

이전기출
4. 데이터베이스에 관한 사항으로 다음에서 설명하는 것은?

> This is a "thing" in the real world an independent existence. It may be an object with a physical existence(a particular person, car, house, or employee) or an object with a conceptual existence(a company, a job, or a university course).

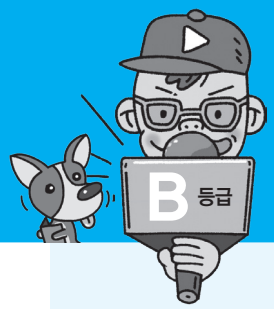
① entity ② view
③ value ④ relationship

이것은 실세계에서 독립적으로 존재하고, 물리적(개인, 차, 집, 종업원 등) 혹은 개념적(회사, 직업 , 대학 과정 등)으로 존재합니다.

▶ 정답 : 1. ④ 2. ② 3. ③ 4. ①

SECTION 073 ● LIVE

데이터 모델의 구성 요소 – 속성(Attribute)



1 속성의 정의 및 특징

- 속성(Attribute)은 데이터베이스를 구성하는 가장 작은 논리적 단위이다.
- 파일 구조상의 데이터 항목 또는 데이터 필드에 해당한다.
- 속성은 개체*를 구성하는 항목이다.
- 속성은 개체의 특성을 기술한다.
- 속성의 수를 디그리(Degree) 또는 차수라고 한다.

전문가의 조언

속성(Attribute)은 개체와 마찬가지로 데이터베이스의 기본 구성 요소이며, 데이터베이스 과목을 공부하는 동안 항상 나오는 용어이므로 확실히 숙지해 두세요.

개체에 대한 자세한 내용은 Section 072를 참조하세요.

2 속성의 종류

속성은 속성의 특성과 개체 구성 방식에 따라 분류할 수 있다.

속성의 특성에 따른 분류

기본 속성 (Basic Attribute)	• 업무 분석을 통해 정의한 속성이다. • 속성 중 가장 많고 일반적이다. • 업무로부터 분석한 속성이라도 업무상 코드로 정의한 속성은 기본 속성에서 제외된다.
설계 속성 (Designed Attribute)	• 원래 업무상 존재하지 않고 설계 과정에서 도출해내는 속성이다. • 업무에 필요한 데이터 외에 데이터 모델링을 위해 업무를 규칙화하려고 속성을 새로 만들거나 변형하여 정의하는 속성이다.
파생 속성 (Derived Attribute)	• 다른 속성으로부터 계산이나 변형 등의 영향을 받아 발생하는 속성이다. • 파생 속성은 되도록 적은 수를 정의하는 것이 좋다.

전문가의 조언

파생 속성은 다른 속성의 영향을 받는 만큼 프로세스 설계 시 정합성 유지를 위해 유의해야할 점이 많으므로 되도록 적게 정의하는 것이 좋습니다.

예제 속성의 특성에 따른 분류

기본 속성	설계 속성	파생 속성
	A01–세단 A02–SUV A03–트럭	총판매수량 총판매금액
자동차명, 제조일, 연비	자동차코드	계산값

해설

기본 속성인 '자동차명', '제조일', '연비'는 업무 분석을 통해 정의한 가장 일반적인 속성이고, 설계 속성인 '자동차코드'는 판매되는 자동차를 종류별로 구분하기 위해 업무에는 없지만 새롭게 정의한 속성이며, 파생 속성인 '총판매수량'과 '총판매금액'은 특정 기간동안 판매된 자동차의 수량과 금액의 합계 계산을 위해 정의한 속성이다.

기본키, 외래키에 대한 자세한 내용은 Section 079를 참조하세요.

전문가의 조언

속성명을 정확하게 지정하고 혼란을 방지하기 위해 용어사전이라는 업무사전을 사용하기도 합니다.
※ **용어사전** : 업무적으로 사용되는 명사형 단어를 체계적으로 정리한 사전

원시 속성
원시 속성은 다른 속성을 통해 다시 재현할 수 없는 속성을 의미합니다. 원시 속성을 버리면 해당 속성 정보가 완전히 소실되므로 버리면 안됩니다.

속성명을 서술형으로 지정하지 않는 이유
속성명을 수식어가 많은 서술형으로 지정하면 해당 속성명이 무엇을 의미하는지 파악하기 어렵고 다루기 힘들기 때문에 의미 파악이 쉽고 다루기 쉬운 명사형으로 지정해야 합니다.

개체 구성 방식에 따른 분류

기본키* 속성(Primary Key Attribute)	개체를 식별할 수 있는 속성이다.
외래키* 속성(Foreign Key Attribute)	다른 개체와의 관계에서 포함된 속성이다.
일반 속성	개체에 포함되어 있고 기본키, 외래키에 포함되지 않은 속성이다.

3 속성 후보 선정 원칙

속성으로 지정할 후보는 최대한 많이 선택하는 것이 좋으며 선정 원칙은 다음과 같다.

- 원시(Source) 속성*으로 판단되는 속성 후보는 버리지 않는다.
- 소그룹별로 속성 후보군을 만들고 가장 근접한 개체에 할당한다.

4 속성명 지정 원칙

- 속성명은 웹이나 클라이언트/서버(Client/Server) 등 어떠한 환경에서든 사용자 인터페이스에 나타나기 때문에 정확하고 혼란이 없도록 지정해야 한다.
- 속성명 지정 원칙
  - 해당 업무에서 사용하는 용어로 지정한다.
  - 서술형*으로 지정하지 않는다.
  - 가급적이면 약어의 사용은 제한한다.
  - 개체명은 속성명으로 사용할 수 없다.
  - 개체에서 유일하게 식별 가능하도록 지정한다.

따라잡기

기출문제 따라잡기

Section 073

출제예상

1. 다음 중 속성에 대한 설명으로 가장 옳지 않은 것은?

① 속성은 데이터베이스를 구성하는 가장 큰 논리적 단위이다.

② 속성은 개체를 구성하는 항목이다.

③ 하나의 속성은 하나의 개체에만 존재한다.

④ 속성의 수를 디그리(Degree) 또는 차수라고 한다.

속성은 데이터베이스를 구성하는 가장 작은 논리적 단위입니다.

출제예상

2. 다음 중 속성의 특성에 따른 분류에 대한 설명으로 옳지 않은 것은?

① 기본 속성은 업무로부터 추출한 모든 속성이다.

② 설계 속성은 데이터 모델링을 위해 업무를 규칙화하려고 속성을 새로 만들거나 변형하여 정의하는 속성이다.

③ 파생 속성은 다른 속성에 영향을 받아 발생하는 속성으로, 보통 계산된 값이 파생 속성에 해당된다.

④ 외래키 속성은 개체와의 관계에서 포함된 속성이다.

속성은 특성에 따라 기본 속성, 설계 속성, 파생 속성으로, 개체 구성 방식에 따라 기본키 속성, 외래키 속성, 일반 속성으로 분류됩니다.

출제예상

3. 속성을 개체 구성 방식에 따라 분류할 경우 포함되지 않는 것은?

① 기본키 속성 ② 외래키 속성

③ 일반 속성 ④ 파생 속성

2번 문제를 풀었으니 어렵지 않게 답을 찾을 수 있겠죠.

출제예상

4. 다음 중 속성명 지정 원칙에 대한 설명으로 가장 옳지 않은 것은?

① 해당 업무에서 사용하는 이름을 부여한다.

② 속성명은 서술식으로 지정해도 된다.

③ 약어 사용은 가급적 자제한다.

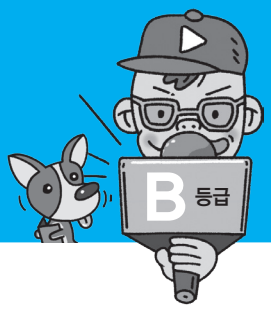
④ 개체명은 속성명으로 사용할 수 없다.

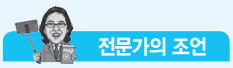
속성명을 수식어가 많은 서술형으로 지정할 경우 무엇을 의미하는지 파악하기 어렵지 않을까요?

▶ 정답 : 1. ① 2. ④ 3. ④ 4. ②

SECTION 074 ● LIVE

데이터 모델의 구성 요소 – 관계(Relationship)





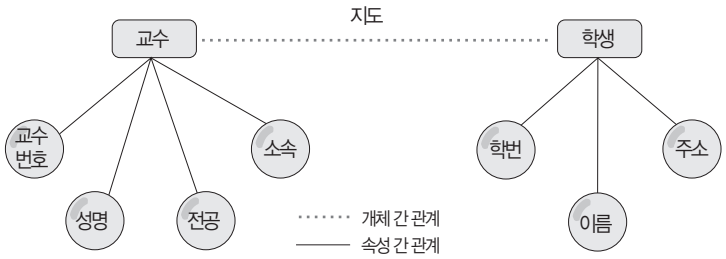
관계는 속성과 개체와 마찬가지로 데이터베이스의 기본 구성 요소로서 데이터베이스 과목을 공부하는 동안 항상 나오는 용어입니다. 관계의 형태와 종류에는 어떤 것이 있고, 각각의 특징이 무엇인지 학습하세요.

관계(Relationship)의 이용
강현준 교수의 전공을 알기 위해서는 속성 간의 관계를 이용하고, 강현준 교수가 가르치는 학생의 이름을 알기 위해서는 개체 간 관계를 이용합니다.

1 관계의 정의

관계는 개체와 개체 사이의 논리적인 연결을 의미한다.

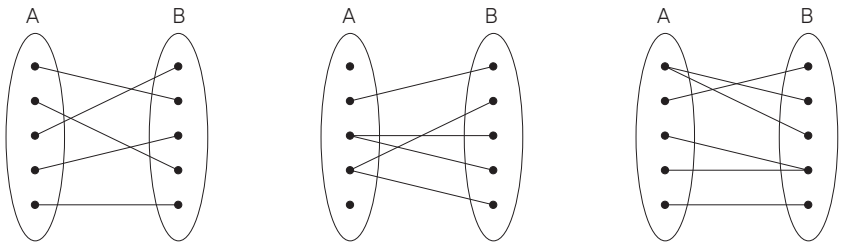
- 관계에는 개체 간의 관계와 속성 간의 관계가 있다.
  예 다음은 교수가 학생을 지도하는 관계이다.



2 관계의 형태

관계의 형태에는 일 대 일(1:1), 일 대 다(1:n), 다 대 다(n:m) 관계가 있다.

- **일 대 일(1:1)** : 개체 집합 A의 각 원소가 개체 집합 B의 원소 한 개와 대응하는 관계이다.
- **일 대 다(1:N)** : 개체 집합 A의 각 원소는 개체 집합 B의 원소 여러 개와 대응하고 있지만, 개체 집합 B의 각 원소는 개체 집합 A의 원소 한 개와 대응하는 관계이다.
- **다 대 다(N:M)** : 개체 집합 A의 각 원소는 개체 집합 B의 원소 여러 개와 대응하고, 개체 집합 B의 각 원소도 개체 집합 A의 원소 여러 개와 대응하는 관계이다.



일 대 일(1:1) 일 대 다(1:N) 다 대 다(N:M)

3 관계의 종류

관계는 처리하는 업무 형태에 따라 종속 관계, 중복 관계, 재귀 관계, 배타 관계 등이 있다.

- **종속 관계(Dependent Relationship)** : 두 개체 사이의 주 · 종 관계를 표현한 것으로, 식별 관계와 비식별 관계가 있다.
- **중복 관계(Redundant Relationship)** : 두 개체 사이에 2번 이상의 종속 관계가 발생하는 관계이다.
- **재귀 관계(Recursive Relationship)** : 개체가 자기 자신과 관계를 갖는 것으로, 순환 관계(Recursive Relationship)라고도 한다.
- **배타 관계(Exclusive Relationship)**
  - 개체의 속성이나 구분자를 기준으로 개체의 특성을 분할하는 관계로, 배타 AND 관계와 배타 OR 관계로 구분한다.
  - 배타 AND 관계는 하위 개체들 중 속성이나 구분자 조건에 따라 하나의 개체만을 선택할 수 있고, 배타 OR 관계는 하나 이상의 개체를 선택할 수 있다.

전문가의 조언

관계 데이터베이스 특성 상 관계가 복잡하면 최적화 작업이 어려우므로 중복 관계는 되도록 사용하지 않는 것이 좋습니다.

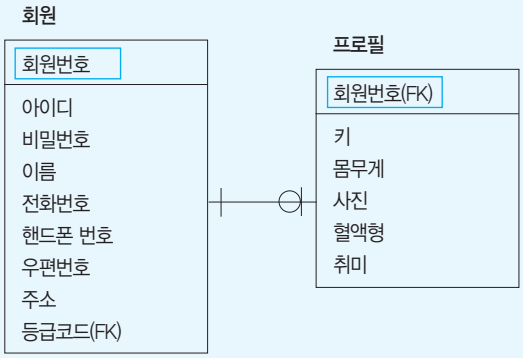
참깐만요

식별 관계(Identifying Relationship)/비식별 관계(Non-Identifying Relationship)

식별 관계(Identifying Relationship)

- 개체 A, B 사이의 관계에서 A 개체의 기본키*가 B 개체의 외래키*이면서 동시에 기본키가 되는 관계를 말합니다.
- B 개체의 존재 여부가 A 개체의 존재 여부에 의존적인 경우에 발생합니다.
- ER 도형에서 식별 관계는 실선으로 표시합니다.

예 회원 개체와 프로필 개체는 회원 개체의 기본키인 회원번호가 프로필 개체의 외래키이면서 동시에 기본키가 되기 때문에 서로 식별 관계에 있습니다.



※ 프로필 개체는 회원 개체의 회원번호에 의존적이기 때문에 회원 개체의 회원번호가 없으면 프로필 개체는 존재할 수 없습니다.

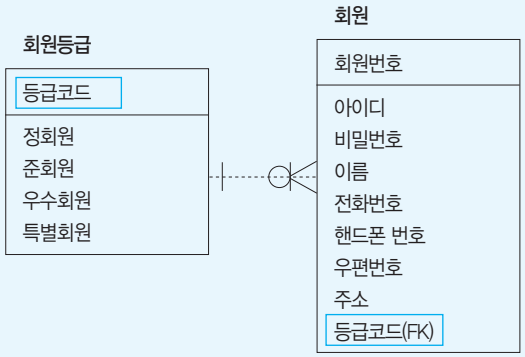
기본키(Primary Key)
기본키는 후보키 중에서 선택한 주키(Main Key)를 의미합니다.
※ **후보키(Candidate Key)** : 릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별할 수 있는 속성들의 부분 집합

외래키(Foreign Key)
외래키는 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미합니다.

비식별 관계(Non-Identifying Relationship)

- 개체 A, B 사이의 관계에서 A 개체의 기본키가 B 개체의 비기본키 영역에서 외래키가 되는 관계를 말합니다.
- B 개체의 존재 여부는 A 개체의 존재 여부와 관계없이 존재합니다.
- 일반적으로 두 개체는 비식별 관계로 존재하는 경우가 많습니다.
- ER 도형에서 비식별 관계는 점선으로 표시합니다.

예 회원등급 개체와 회원 개체는 회원등급 개체의 기본키인 등급코드가 회원 개체의 비기본키 영역에서 외래키가 되기 때문에 서로 비식별 관계에 있습니다.



※ 회원 개체는 회원등급 개체에 독립적이기 때문에 회원등급 개체의 등급코드에 관계없이 존재할 수 있습니다.

따라잡기

기출문제 따라잡기 Section 074

출제예상

1. 개체 집합 A의 각 원소는 개체 집합 B의 원소 여러 개와 대응하고 있지만, 개체 집합 B의 각 원소는 개체 집합 A의 원소 한 개와 대응하는 관계를 의미하는 것은?

① 1:1 ② 1:n
③ n:m ④ 1:0

생각자나지 않으면 310쪽의 그림을 다시 한 번 보고 오세요.

출제예상

2. 다음 중 관계의 종류에 대한 설명으로 가장 옳지 않은 것은?

① 종속 관계는 두 개체 간의 주/종 관계를 표현하는 것이다.
② 배타 관계는 개체가 둘 이상의 다른 개체의 합집합 또는 교집합과 관계를 갖는 것이다.
③ 중복 관계는 두 개체들 간에 두 번 이상의 종속 관계가 발생하는 것이다.
④ 재귀 관계는 하나의 개체가 여러 개체와 관계를 갖는 것이다.

재귀는 자기 자신을 참조하는 것을 의미합니다.

출제예상

3. 개체 집합 A의 각 원소는 개체 집합 B의 원소 여러 개와 대응하고, 개체 집합 B의 각 원소도 개체 집합 A의 원소 여러 개와 대응하는 관계를 의미하는 것은?

① 1:0 ② 1:1
③ 1:n ④ n:m

1번 문제를 풀었으니 어렵지 않게 답을 찾을 수 있겠죠.

출제예상

4. 다음 중 관계의 종류에 해당하지 않는 것은?

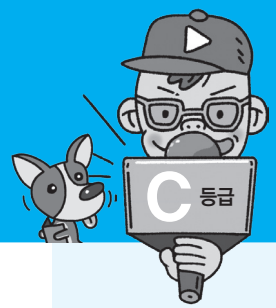
① 종속 관계 ② 재귀 관계
③ 단일 관계 ④ 중복 관계

관계의 종류에는 종속, 중복, 재귀, 배타 관계가 있다는 것! 잊지마세요.

▶ 정답 : 1. ② 2. ④ 3. ④ 4. ③

SECTION 075 LIVE

식별자(Identifier)



1 식별자의 정의 및 분류

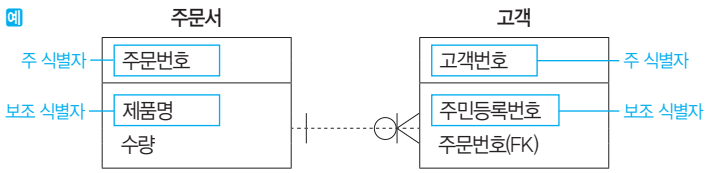
식별자(Identifier)는 하나의 개체 내에서 각각의 인스턴스*를 유일(Unique)하게 구분할 수 있는 구분자로, 모든 개체는 한 개 이상의 식별자를 반드시 가져야 한다.

- 식별자는 개체 내에서 대표성 여부, 스스로 생성 여부, 단일 속성 여부, 대체 여부에 따라 다음과 같이 분류한다.

분류	식별자
대표성 여부	주 식별자(Primary Identifier), 보조 식별자(Alternate Identifier)
스스로 생성 여부	내부 식별자(Internal Identifier), 외부 식별자(Foreign Identifier)
단일 속성 여부	단일 식별자(Single Identifier), 복합 식별자(Composit Identifier)
대체 여부	원조 식별자(Original Identifier), 대리 식별자(Surrogate Identifier)

2 주 식별자/보조 식별자

- 주 식별자(Primary Identifier)는 개체를 대표하는 유일한 식별자이다.
- 보조 식별자(Alternate Identifier)는 주 식별자를 대신하여 개체를 식별할 수 있는 속성이다.
- 두 식별자 모두 개체를 유일하게 식별할 수 있어야 한다.
- 하나의 개체에 주 식별자는 한 개만 존재하지만 보조 식별자는 한 개 이상 존재한다.
- 개체를 유일하게 식별할 수 있는 속성이 두 개 이상인 경우 업무에 가장 적합한 속성을 주 식별자로 지정하고, 나머지는 보조 식별자로 지정한다.
- 물리적 테이블에서 주 식별자는 기본키(Primary Key)로, 보조 식별자는 유니크 인덱스(Unique Index)*로 지정되어 사용된다.



전문가의 조언

식별자의 종류와 종류별 식별자의 특징을 구분할 수 있어야 합니다. 종류별 식별자의 특징은 명칭을 통해 쉽게 추측할 수 있으니 특징을 파악해 두세요.

전문가의 조언

식별자와 키(Key)는 같은 의미로 사용됩니다. 식별자는 논리 데이터 모델링 단계에서 사용되고, 키는 물리 데이터 모델링 단계에서 사용됩니다. 따라서 개체는 식별자를 갖고, 릴레이션은 키를 갖습니다. 키에 대한 자세한 내용은 Section 079에서 공부할 예정입니다.

인스턴스(Instance)

데이터 개체를 구성하고 있는 속성들에 데이터 타입이 정의되어 구체적인 데이터 값을 갖고 있는 것을 말합니다.

유니크 인덱스(Unique Index)

유니크 인덱스는 속성값에서 중복값이 없도록 하는 인덱스입니다.

전문가의 조언

'제품명'은 유일한 값들로 구성되는 식별자라는 전제에서만 '보조 식별자'가 될 수 있습니다.

잠깐만요

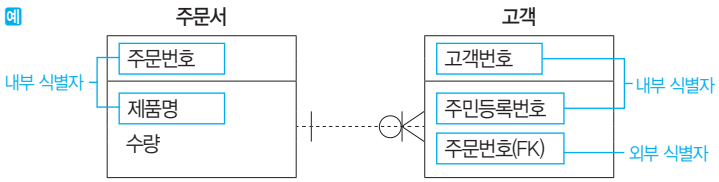
주 식별자의 4가지 특징

특징	설명
유일성	주 식별자에 의해 개체 내에 모든 인스턴스들이 유일하게 구분되어야 합니다.
최소성	주 식별자를 구성하는 속성의 수는 유일성을 만족하는 최소 수가 되어야 합니다.
불변성	주 식별자가 한 번 특정 개체에 지정되면 그 식별자는 변하지 않아야 합니다.
존재성	주 식별자가 지정되면 식별자 속성에 반드시 데이터 값이 존재해야 합니다.

3 내부 식별자/외부 식별자

- 내부 식별자(Internal Identifier)는 개체 내에서 스스로 만들어지는 식별자이다.
- 외부 식별자(Foreign Identifier)는 다른 개체와의 관계(Relationship)에 의해 외부 개체의 식별자를 가져와 사용하는 식별자이다.
- 외부 식별자는 자신의 개체에서 다른 개체를 찾아가는 연결자 역할을 한다.

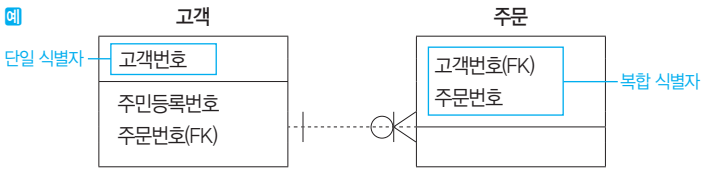
예



4 단일 식별자/복합 식별자

- 단일 식별자(Single Identifier)는 주 식별자가 한 가지 속성으로만 구성된 식별자이다.
- 복합 식별자(Composit Identifier)는 주 식별자가 두 개 이상의 속성으로 구성된 식별자이다.

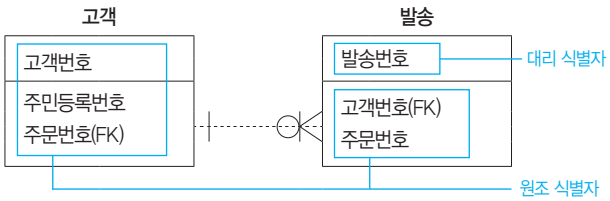
예



5 원조 식별자/대리 식별자

- 원조 식별자(Original Identifier)는 업무에 의해 만들어지는 가공되지 않은 원래의 식별자로, 본질 식별자라고도 한다.
- 대리 식별자(Surrogate Identifier)는 주 식별자의 속성이 두 개 이상인 경우 속성들을 하나의 속성으로 묶어 사용하는 식별자로, 인조 식별자라고도 한다.

예



6 대리 식별자의 조건

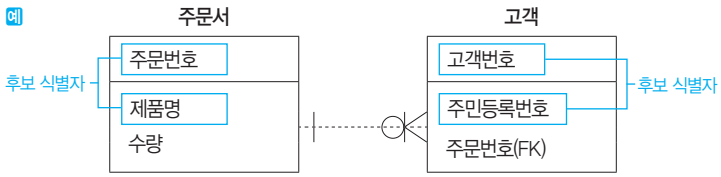
- 최대한 범용적인 값을 사용한다.
- 유일한 값을 만들기 위한 대리 식별자를 사용한다.
- 하나의 대리 식별자 속성으로 대체할 수 없는 경우를 주의한다.
- 편의성과 단순성, 의미의 체계화를 위한 대리 식별자를 사용할 수 있다.
- 시스템적인 필요성에 의해 내부적으로만 사용하는 대리 식별자를 사용할 수 있다.

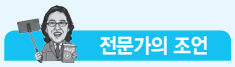
7 후보 식별자

후보 식별자는 개체에서 각 인스턴스를 유일하게 식별할 수 있는 속성 또는 속성 집합을 의미한다.

- 하나의 개체에는 한 개 이상의 후보 식별자가 있고, 이 중 개체의 대표성을 나타내는 식별자를 주 식별자로, 나머지는 보조 식별자로 지정한다.
- 후보 식별자 조건
  - 각 인스턴스를 유일하게 식별할 수 있어야 한다.
  - 속성들을 직접 식별할 수 있어야 한다.
  - 널 값(NULL Value)*이 될 수 없다.
  - 속성 집합은 후보 식별자로 지정한 경우 개념적으로 유일해야 한다.
  - 후보 식별자의 데이터는 자주 변경되지 않아야 한다.

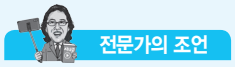
예





전문가의 조언

'발송번호'는 '고객번호'와 '주문번호' 속성을 하나로 묶어서 생성한 대리 식별자입니다.



전문가의 조언

후보 식별자의 경우 개념 및 후보 식별자가 되기 위한 조건에는 무엇이 있는지 학습하세요.

널 값(NULL Value)
데이터베이스에서 아직 알려지지 않거나 모르는 값으로서 '해당 없음' 등의 이유로 정보 부재를 나타내기 위해 사용하는, 이론적으로 아무것도 없는 특수한 데이터를 말합니다.

출제예상

1. 다음 중 식별자를 대표성 여부로 분류했을 경우 해당하는 것은?

① 내부 식별자, 외부 식별자
② 주 식별자, 보조 식별자
③ 단일 식별자, 복합 식별자
④ 원조 식별자, 대리 식별자

식별자를 대표성 여부로 분류하면 주와 보조 식별자, 스스로 생성 여부로 분류하면 내부와 외부 식별자, 단일 속성으로 분류하면 단일과 복합 식별자, 대체 여부로 분류하면 원조와 대리 식별자입니다.

출제예상

2. 다음 중 주 식별자와 보조 식별자에 대한 설명으로 가장 옳지 않은 것은?

① 주 식별자는 개체의 대표성을 나타내는 유일한 식별자이다.
② 보조 식별자는 주 식별자를 대신하여 보조적으로 개체를 식별할 수 있는 속성이다.
③ 주 식별자는 개체를 유일하게 식별하지만 보조 식별자는 식별하지 못한다.
④ 주 식별자는 개체 하나에 한 개인 반면 보조 식별자는 하나 이상일 수 있다.

보조 식별자는 주 식별자를 대신하는 것인데 개체를 식별할 수 없다면 의미가 있을까요?

출제예상

3. 다음 중 주 식별자의 특징이 아닌 것은?

① 존재성 ② 최대성
③ 불변성 ④ 유일성

주 식별자의 특징하면 '유일성, 최소성, 불변성, 존재성'이란 것을 기억해 두세요.

출제예상

4. 다음 중 식별자에 대한 설명으로 가장 옳지 않은 것은?

① 내부 식별자는 자신의 개체 타입 내에서 스스로 생성되어 존재하는 식별자이다.
② 외부 식별자는 다른 개체와의 관계에 의해 주 식별자 속성을 상속받아 자신의 속성에 포함되는 식별자이다.
③ 단일 식별자는 주 식별자의 구성이 한 가지 속성으로만 이루어진 것이다.
④ 다중 식별자는 두 개 이상의 속성으로 구성된 식별자이다.

두 개 이상의 속성으로 구성된 식별자는 복합 식별자라고 합니다.

출제예상

5. 다음 중 후보 식별자에 대한 설명으로 가장 옳지 않은 것은?

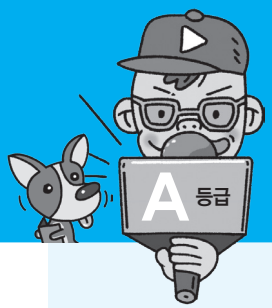
① 후보 식별자는 개체에서 각 인스턴스를 유일하게 식별할 수 있는 속성 또는 속성 집합이다.
② 하나의 개체에는 한 개 이상의 후보 식별자가 있다.
③ 후보 식별자는 널 값(NULL Value)이 될 수 있다.
④ 후보 식별자의 데이터는 자주 변경되지 않아야 한다.

널 값(NULL Value)은 아무 것도 없는 것을 의미합니다.

▶ 정답 : 1. ② 2. ③ 3. ② 4. ④ 5. ③

SECTION 076 ● LIVE

E-R(개체-관계) 모델



1 E-R(Entity-Relationship, 개체-관계) 모델의 개요

E-R 모델은 개념적 데이터 모델의 가장 대표적인 것으로, 1976년 피터 첸(Peter Chen)에 의해 제안되고 기본적인 구성 요소가 정립되었다.

- E-R 모델은 개체와 개체 간의 관계를 기본 요소로 이용하여 현실 세계의 무질서한 데이터를 개념적인 논리 데이터로 표현하기 위한 방법으로 많이 사용되고 있다.
- E-R 모델은 개체 타입(Entity Type)과 이들 간의 관계 타입(Relationship Type)을 이용해 현실 세계를 개념적으로 표현한다.
- E-R 모델에서는 데이터를 개체(Entity)*, 관계(Relationship)*, 속성(Attribute)*으로 묘사한다.
- E-R 모델은 특정 DBMS를 고려한 것은 아니다.
- E-R 다이어그램으로 표현하며, 1:1, 1:N, N:M 등의 관계 유형을 제한 없이 나타낼 수 있다.
- 최초에는 개체, 관계, 속성과 같은 개념들로 구성되었으나 나중에는 일반화 계층 같은 복잡한 개념들이 첨가되어 확장된 모델로 발전했다.

2 E-R 다이어그램

E-R 다이어그램은 E-R 모델의 기본 아이디어를 이해하기 쉽게 기호를 사용하여 시각적으로 표현한 그림이다.

- E-R 다이어그램은 실체 간의 관계는 물론 조직, 사용자, 프로그램, 데이터 등 시스템 내에서 역할을 가진 모든 실체들을 표현한다.
- E-R 다이어그램은 데이터에 대해 개발자, 관리자, 사용자들이 서로 다르게 인식하고 있는 뷰(View)들을 하나로 단일화시킨다.
- E-R 다이어그램 표기법에는 피터 첸 표기법, 정보 공학 표기법, 바커 표기법 등이 있다.

3 피터 첸 표기법

피터 첸 표기법(Peter Chen Notation)은 1976년 피터 첸(Peter Chen)이 개발하였다.

- 피터 첸 표기법은 개체, 속성, 기본키 속성, 관계 등을 다음과 같이 표현한다.

전문가의 조언

E-R 모델의 개념과 특징은 중요합니다. E-R 모델의 개념과 특징을 확실히 숙지하세요.

개체, 관계, 속성

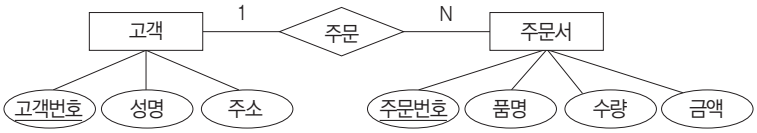
- **개체(Entity)** : 학생, 교수, 자동차 등과 같이 실세계에서 개념적 또는 물리적으로 존재하는 실제 사용을 의미합니다.
- **관계(Relationship)** : 교수 개체는 학생 개체를 지도하는 관계인 것처럼 다른 개체 타입에 속한 개체 사이의 관계를 표현합니다.
- **속성(Attribute)** : 학생의 이름, 주소 등과 같이 개체를 묘사하는 데 사용될 수 있는 특성을 의미합니다.

전문가의 조언

E-R 다이어그램에서는 피터 첸 표기법에서 사용되는 기호를 반드시 암기하고, 정보 공학 표기법과 바커 표기법은 두 표기법의 표현하는 방법을 구분할 수 있도록 정리하세요.

기호	기호 이름	의미
	사각형	개체(Entity) 타입
	마름모	관계(Relationship) 타입
	타원	속성(Attribute)
	이중 타원	다중값 속성(복합 속성)
	밑줄 타원	기본키 속성
	복수 타원	복합 속성 예 성명은 성과 이름으로 구성
n m	관계	1:1, 1:n, n:m 등의 개체 간 관계에 대한 대응수를 선 위에 기술함
	선, 링크	개체 타입과 속성을 연결

예제 다음은 피터 첸 표기법을 이용하여 고객과 주문서 간의 관계를 나타낸 E-R 다이어그램이다.



해설

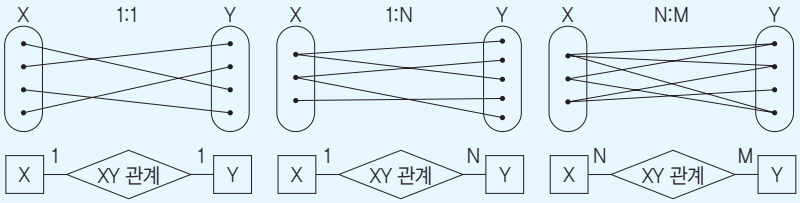
- **개체** : 고객, 주문서
- **속성**
  - 고객의 속성 : 고객번호, 성명, 주소
  - 주문서의 속성 : 주문번호, 품명, 수량, 금액
- **관계** : '고객'과 '주문서'의 '주문' 관계는 일 대 다의 관계, 즉 한 사람의 고객이 다수의 주문을 할 수 있고 주문서 1개는 특정인의 주문서로 되어 있다.
- 밑줄친 속성은 기본키*를 나타낸다.

기본키(Primary Key)
기본키는 개체 인스턴스들을 서로 구분할 수 있는 유일한 속성을 말합니다.

잠깐만요

E-R 다이어그램 대응수

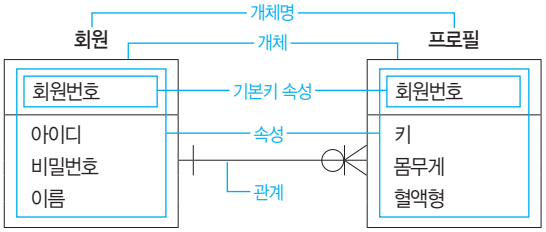
E-R 다이어그램 표현에서 대응수는 함수 관계도를 이용하여 쉽게 구할 수 있습니다.



4 정보 공학 표기법(Information Engineering Notation)

정보 공학 표기법은 1981년에 클리프 핀켈쉬타인(Clive Finkelstein)과 제임스 마틴(James Martin)이 공동 개발하였다.

- 정보 공학 표기법은 개체, 속성, 기본키 속성, 관계 등을 다음과 같이 표현한다.

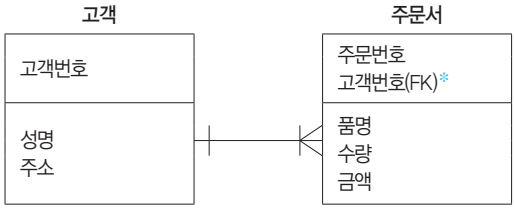


– 개체는 사각형 박스로 표시하고 개체명은 박스 바깥쪽 위에 표시한다.
– 속성은 기본키 속성과 일반 속성을 분리하여 표시한다.
– 관계는 관계 표기 기호를 사용하여 표시한다.

- **관계 표기 기호**

기호	의미
|	필수(Mandatory)
○	선택적(Optional)
<	다중(Multiple)

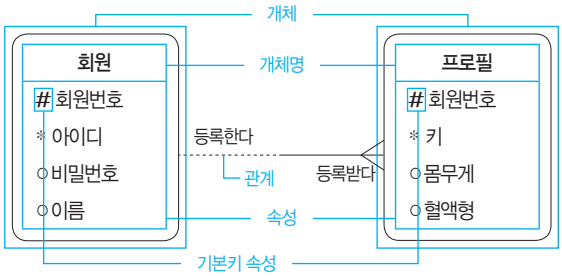
예제 318쪽의 **예제**를 정보 공학 표기법을 이용하여 E-R 다이어그램으로 나타내면 다음과 같다.



5 바커 표기법(Barker Notation)

바커 표기법은 영국 컨설팅 회사 CACI에서 개발하였고, 리차드 바커(Richard Barker)에 의해 정립되었다.

- 바커 표기법은 개체, 속성, 기본키 속성, 관계 등을 다음과 같이 표현한다.



전문가의 조언

관계는 양방향의 관계를 합쳐서 표현하므로 방향성에 따른 의미는 다음과 같습니다.

'회원' → '프로필' 관계 의미
각각의 회원은 하나 이상(Multiple)의 프로필을 갖거나 갖지 않을 수(Optional) 있습니다.

※ : 0 또는 1 또는 N

※ : 0 또는 1

'프로필' → '회원' 관계 의미
각각의 프로필은 하나의 회원을 반드시 가져야(Mandatory) 합니다.

고객번호(FK)
- '고객'과 '주문서'는 식별 관계이므로 주문서에 '고객번호'를 외래키(FK) 형태의 기본키로 추가해야 합니다.
- 논리 개체 관계도에서는 외래키에 해당하는 속성에 'FK'를 표기하여 외래키임을 나타냅니다.

전문가의 조언

두 개체를 연결하고 있는 실선과 점선은 관계선이며, 실선은 필수(Mandatory) 관계를, 점선은 선택적(Optional) 관계를 의미합니다. 관계선은 관계선이 붙어 있는 개체가 주어가 됐을 때 다른 개체를 바라보는 관계를 설명한 것입니다.

'회원' → '프로필' 관계 의미
'회원' 개체에 붙어 있는 관계선이 점선, 즉 선택적(Optional) 관계이므로 "각각의 회원은 하나 이상(Multiple)의 프로필을 갖거나 갖지 않을 수(Optional) 있다."는 의미입니다.

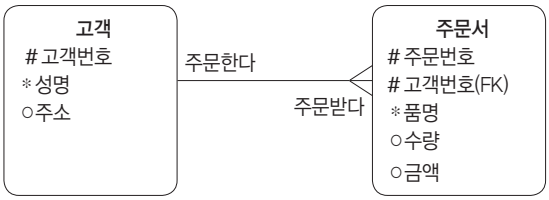
'프로필' → '회원' 관계 의미
'프로필' 개체에 붙어 있는 관계선이 실선, 즉 필수(Mandatory) 관계이므로 '각각의 프로필은 하나의 회원을 반드시 가져야(Mandatory) 한다.'는 의미입니다.

– 개체는 모서리가 둥근 박스로 표시하고 개체명은 박스 안 가장 위에 표시한다.
– 속성은 반드시 값이 저장되어야 하는 경우 *(Mandatory)를 표시하고, 값이 저장될 수도, 안될 수도 있는 경우 ○(Optional)을 표시한다.
– 관계는 관계 표기 기호를 사용하여 표시한 후 해당 개체의 역할을 동사적 단어로 입력한다.

- **관계 표기 기호**

기호	의미
————	필수(Mandatory)
- - - - - - - -	선택적(Optional)
<	다중(Multiple)

예제 318쪽의 **예제**를 바커 표기법을 이용하여 E-R 다이어그램으로 나타내면 다음과 같다.



관계의 표현 및 이해

관계	표현		의미
	정보 공학 표기법	바커 표기법	
1:1	┼──┼	————	양쪽에 반드시 1개씩 존재
1:0 또는 1:1	┼──○┤	- - - - ——	왼쪽에는 반드시 1개, 오른쪽에는 없거나 1개 존재
1:N	┼──<	——<	왼쪽에는 반드시 1개, 오른쪽에는 반드시 여러 개 존재
1:1 또는 1:N	┼──┤<	——<	왼쪽에는 반드시 1개, 오른쪽에는 1개 또는 여러 개 존재
1:0 또는 1:1 또는 1:N	┼──○<	- - - -—<	왼쪽에는 반드시 1개, 오른쪽에는 0개 또는 1개 또는 여러 개 존재

기출문제 따라잡기

Section 076

이전기출

1. 개체-관계(E-R) 모델에 대한 설명으로 잘못된 것은?

① E-R 다이어그램으로 표현하며 P. Chen이 제안했다.
② 일 대 일(1:1) 관계 유형만을 표현할 수 있다.
③ 개체 타입과 이들 간의 관계 타입을 이용해 현실 세계를 개념적으로 표현한 방법이다.
④ E-R 다이어그램은 E-R 모델을 그래프 방식으로 표현한 것이다.

E-R 모델은 특정 DBMS에 맞게 작성하는 것이 아니죠. 그러므로 나타낼 수 있는 관계의 유형은 일 대 일, 일 대 다, 다 대 다입니다.

이전기출

2. 개체-관계 모델(E-R Model)에 대한 설명으로 옳지 않은 것은?

① 개체와 개체 간의 관계를 도식화한다.
② 개체 집합을 사각형으로 표시한다.
③ 관계를 다이아몬드로 표시한다.
④ 오너-멤버(Owner-Member) 관계라고도 한다.

오너-멤버(Owner-Member) 관계는 망형 데이터 모델의 특징입니다.

이전기출

3. P.Chen이 제안한 것으로 현실 세계에 존재하는 객체들과 그들간의 관계를 사람이 이해하기 쉽게 표현한 모델은?

① 개체-관계(E-R) 모델
② 관계 데이터 모델
③ 네트워크 데이터 모델
④ 계층 데이터 모델

여기까지 와서 이 문제를 모른다고 하면 안됩니다.

이전기출

4. 다음 설명이 뜻하는 것은?

> In the design of information systems, a diagram that shows all the entities(organizations, departments, users, programs, and data) that play roles in the system, as well as the relationships between those entities.

① E-R Diagram ② Flow Chart
③ View ④ Normalization

정보 시스템 설계에서, 실체 간의 관계는 물론 시스템 내의 역할을 하는 모든 실체들, 즉 조직, 부서, 사용자, 프로그램, 데이터를 표시하는 다이어그램을 무엇이라 할까요?

이전기출

5. E-R 다이어그램의 구성 요소에 대한 표현이 틀리게 짝지어진 것은?

① 개체 집합 - 직사각형 ② 관계 집합 - 마름모꼴
③ 속성 - 원 ④ 링크 - 화살표

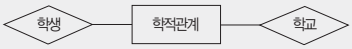
E-R 다이어그램! 이거 방법을 달리하여 자주 출제될 것으로 예상됩니다. 꼭 암기하세요.

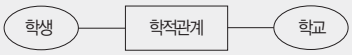
이전기출

6. 학생과 학교 개체 간의 학적 관계를 E-R 다이어그램으로 옳게 표현한 것은?

①

②

③


④


5번 문제와 같다고 봐도 되겠죠? 관계는 다이아몬드, 개체는 사각형!

이전기출

7. 사람과 도시 사이의 거주 관계에서 사람은 반드시 하나의 도시에 거주해야만 하며, 하나의 도시에는 다수의 사람이 거주한다고 할 때 이를 E-R 다이어그램으로 정확히 표현한 것은?

①

②

③

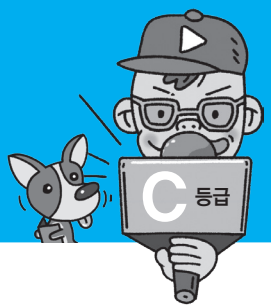
④

사람은 하나의 도시에, 도시에 사는 사람은 여럿, 그러면 도시와 사람의 관계는 1:N이 되겠네요. 이런 거 혼동되면 함수 관계도를 그려보세요. 오른쪽의 그림처럼 말이죠.

▶ 정답 : 1. ② 2. ④ 3. ① 4. ① 5. ④ 6. ① 7. ③

SECTION 077 ● LIVE

관계형 데이터 모델



전문가의 조언

관계형 데이터 모델이란 테이블을 이용해 데이터 상호 간의 관계를 표현하는 데이터 모델로, 데이터베이스에 저장된 데이터를 테이블의 형태로 표현한 것을 말합니다. 관계형 데이터 모델의 개념 및 특징을 정리하세요.

관계(Relation)와 관계(Relationship)
관계 모델에서는 테이블을 릴레이션(Relation)이라고 부르는데, 릴레이션을 우리말로 관계라고 해석하는 경우가 종종 있어 개체와 개체 간의 관계를 나타내는 Relationship과 혼동되는 경우가 있습니다. 대부분 Relation은 테이블 혹은 릴레이션이라 표기하고 Relationship은 관계라고 표시하지만 간혹 그렇지 않은 경우도 있으니 주의하기 바랍니다.

기본키와 외래키
기본키와 외래키에 대해서는 Section 079에서 학습합니다.

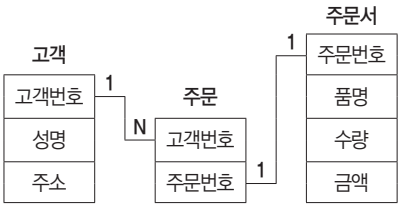
1 관계형 데이터 모델(Relational Data Model)의 개요

관계형 데이터 모델은 가장 널리 사용되는 데이터 모델로, 2차원적인 표(Table)를 이용해서 데이터 상호 관계를 정의하는 DB 구조를 말한다.

- 파일 구조처럼 구성한 테이블들을 하나의 DB로 묶어서 테이블 내에 있는 속성들 간의 관계(Relationship)*를 설정하거나 테이블 간의 관계를 설정하여 이용한다.
- 기본키(Primary Key)*와 이를 참조하는 외래키(Foreign Key)*로 데이터 간의 관계를 표현한다.
- 계층 모델과 망 모델의 복잡한 구조를 단순화시킨 모델이다.
- 관계형 모델의 대표적인 언어는 SQL이다.
- 1:1, 1:N, M:N 관계를 자유롭게 표현할 수 있다.

2 관계형 데이터 모델의 구성

예제 E-R 모델에서 설명한 고객과 주문서 그리고 주문 관계를 관계형 데이터 모델로 표시하면 다음과 같다.



해설

- '고객'과 '주문서'는 개체 테이블이고, '주문'은 관계(Relationship)로 만들어진 테이블이다.
- 위의 테이블에 데이터를 입력해 보면 좀 더 명확하게 이해할 수 있다.

고객

고객번호	성명	주소
001	박인식	수원시
002	김규정	안산시
003	이정훈	서울시
004	강현준	안양시

주문

고객번호	주문번호
001	101
001	102
002	103
002	104
003	105
003	106
003	107
003	108
004	109

주문서

주문번호	품명	수량	금액
101	01	2	300
102	02	3	200
103	03	4	100
104	01	7	200
105	02	5	300
106	05	2	200
107	01	3	100
108	02	1	200
109	03	5	200

- 〈고객〉 테이블과 〈주문서〉 테이블은 〈주문〉 테이블을 관계로 하여 연결되어 있다. 이때 〈고객〉 테이블에 있는 '고객번호'와 〈주문서〉 테이블에 있는 '주문번호'는 기본키이고, 〈주문〉 테이블에 있는 '고객번호'와 '주문번호'는 외래키이다.
- 〈고객〉과 〈주문〉의 관계는 1:N, 즉 한 명의 고객은 여러 개의 주문을 신청할 수 있다.
- 〈주문서〉와 〈주문〉의 관계는 1:1, 즉 '주문서'는 '주문번호' 1개에 대한 정보만을 가지고 있다.
- 〈주문〉 테이블의 내용을 가지고 주문 고객의 '성명'을 찾으려면 〈주문〉 테이블의 '고객번호'와 같은 '고객번호'를 〈고객〉 테이블에서 찾으면 되고, 주문 내역을 알려면 '주문번호'와 같은 '주문번호'를 〈주문서〉 테이블에서 찾으면 된다.

기출문제 따라잡기

Section 077

이전기출

1. 다음의 논리적인 데이터 모델에서 데이터 간의 관계를 기본키(Primary Key)와 이를 참조하는 외래키(Foreign Key)로 표현하는 데이터 모델은?

① 관계형 데이터 모델
② 네트워크 데이터 모델
③ 계층적 모델
④ 객체지향 데이터 모델

기본키와 외래키를 사용하는 관계형 데이터 모델, 아직 잊지 않았죠?

이전기출

2. 개체 집합에 대한 속성 관계를 표현하기 위해 개체를 테이블(Table)로 사용하고 개체 집합들 사이의 관계를 공통 속성으로 연결하는 독립된 형태의 데이터 모델은?

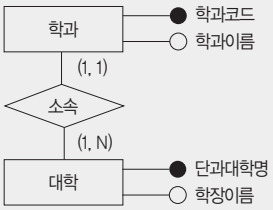
① 망 데이터 모델
② 계층 데이터 모델
③ 관계 데이터 모델
④ 객체지향 데이터 모델

개체를 테이블로 표시한다는 내용이 있으면 관계형 모델입니다.

▶ 정답 : 1. ① 2. ③

이전기출

3. 다음의 E-R 다이어그램은 대학교에서 어떤 학과가 어느 단과대학에 소속되는지의 관계를 모델링한 것이다. 아래의 다이어그램을 관계 모델의 릴레이션으로 표현한 것들 중 가장 적합한 것은?(단, 릴레이션 표현에서 밑줄 친 애트리뷰트는 기본키를 의미한다)



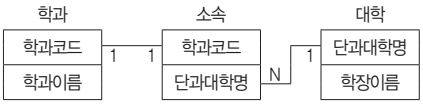
① 학과(학과코드, 학과이름, 단과대학명)
대학(단과대학명, 학장이름)

② 학과(학과코드, 학과이름)
대학(단과대학명, 학장이름, 소속학과)

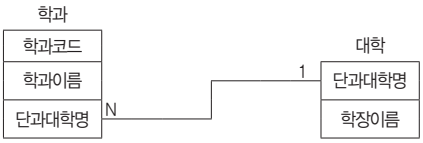
③ 학과(학과코드, 학과이름, 단과대학명, 학장이름)

④ 학과(학과코드, 학과이름)
대학(단과대학명, 학장이름)
소속(학과코드, 단과대학명)

관계 모델에서 릴레이션은 테이블과 같은 의미입니다.
1:N의 관계는 두 가지로 표현할 수 있다는 것은 알고 있죠? 어떤 방법이 효율적인지 두 가지를 모두 살펴보기로 하죠.
첫째, 1:N의 관계에서 〈소속〉 관계를 별도의 관계 릴레이션으로 구성합니다.



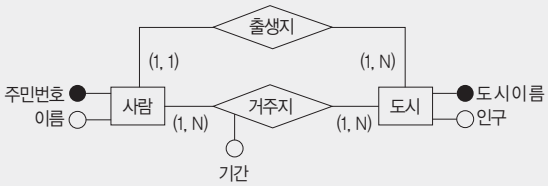
둘째, 소속 관계를 별도의 릴레이션으로 구성하지 않고 〈대학〉의 기본키를 〈학과〉의 외래키로 추가하여 만듭니다.



※ 일반적으로 두 번째와 같이 만드는 방법이 첫 번째에 나타나는 '학과코드'의 중복이 없으므로 효율적이라고 할 수 있습니다.

이전기출

4. 다음 E-R 다이어그램에 대한 설명 중 거리가 먼 것은?



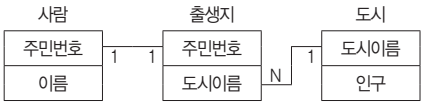
① 사람 개체의 기본키는 주민번호이다.

② 사람과 도시 사이의 거주지 관계는 다 대 다 관계이다.

③ 거주지 관계를 관계 데이터 모델로 표현할 때 별도의 릴레이션으로 모델링하는 것이 일반적이다.

④ 출생지 관계를 관계 데이터 모델로 표현할 때 별도의 릴레이션으로 모델링한다면 기본키는 주민번호와 도시이름이 되어야 한다.

④번이 왜 답인지 출생지 관계를 별도의 릴레이션으로 구성해서 알아보겠습니다. 나머지는 본문 내용을 참조하세요.

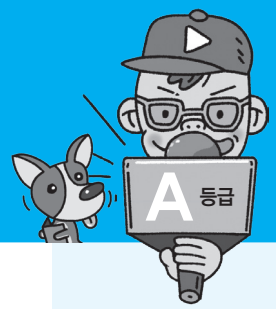


위의 관계 모델에서 〈사람〉과 〈출생지〉의 관계는 1:1이므로 〈출생지〉 릴레이션에는 같은 '주민번호'가 존재하지 않지만 〈도시〉와 〈출생지〉는 1:N의 관계이므로 〈출생지〉 릴레이션에는 같은 도시이름이 다수 존재할 수 있습니다. 즉 기본키는 개체를 구분할 수 있는 유일한 속성이어야 하므로 '주민번호'가 되어야 합니다. 이해가 안 되면 데이터가 들어 있는 앞 페이지의 〈고객〉과 〈주문서〉를 다시 한 번 읽어 보세요. 〈주문〉 릴레이션에서는 '고객번호'가 아닌 주문번호가 기본키가 될 수 있습니다.

▶ 정답 : 3. ① 4. ④

SECTION 078 ● LIVE

관계형 데이터베이스의 구조



1 관계형 데이터베이스의 개요

- 1970년 IBM에 근무하던 코드(E. F. Codd)에 의해 처음 제안되었다.
- 관계형 데이터베이스를 구성하는 개체(Entity)나 관계(Relationship)를 모두 릴레이션(Relation)이라는 표(Table)로 표현한다.
- 릴레이션은 개체를 표현하는 개체 릴레이션, 관계를 나타내는 관계 릴레이션으로 구분할 수 있다.
- **장점** : 간결하고 보기 편리하며, 다른 데이터베이스로의 변환이 용이하다.
- **단점** : 성능이 다소 떨어진다.

2 관계형 데이터베이스의 Relation 구조

릴레이션은 데이터들을 표(Table)의 형태로 표현한 것으로 구조를 나타내는 릴레이션 스키마와 실제 값들인 릴레이션 인스턴스* 로 구성된다.

〈학생〉 릴레이션

학번	이름	학년	신장	학과
19001	김예소	2	170	CD
20002	고강민	1	169	CD
19012	이향기	2	180	ID
17032	김동준	4	174	ED

튜플(Tuple)*

- 튜플은 릴레이션을 구성하는 각각의 행을 말한다.
- 튜플은 속성의 모임으로 구성된다.
- 파일 구조에서 레코드와 같은 의미이다.
- 튜플의 수를 카디널리티(Cardinality) 또는 기수, 대응수라고 한다.

속성(Attribute)*

- 속성은 데이터베이스를 구성하는 가장 작은 논리적 단위이다.
- 파일 구조상의 데이터 항목 또는 데이터 필드에 해당된다.
- 속성은 개체의 특성을 기술한다.
- 속성의 수를 디그리(Degree) 또는 차수라고 한다.

전문가의 조언

관계형 데이터베이스에서 릴레이션을 구성하는 용어들은 데이터베이스 과목을 공부하는 동안 자주 나오는 용어입니다. 반드시 숙지하고 넘어가세요.

용어
동일한 요소에 대해 Section 072 개체(Entity)에서 사용했던 것과 다르게 부릅니다. 비교해 보고 꼭 숙지하세요.

릴레이션 인스턴스
데이터 개체를 구성하고 있는 속성들에 데이터 타입이 정의되어 구체적인 데이터 값을 갖고 있는 것을 말합니다.
예 〈학생〉 릴레이션의 인스턴스

튜플
- 〈학생〉 릴레이션에서 카디널리티는 4입니다.
- 카디널리티 = 튜플의 수 = 기수 = 대응수

속성
- 〈학생〉 릴레이션에서 디그리는 5입니다.
- 디그리 = 속성의 수 = 차수

도메인
〈학생〉 릴레이션에서 '학년'의 도메인은 1~4입니다.

전문가의 조언

릴레이션의 특징을 무조건 암기하지 말고 주어진 예를 〈학생〉 릴레이션에 적용시켜 보세요. 쉽게 이해됩니다.

도메인(Domain)*

- 도메인은 하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자(Atomic)값들의 집합이다.
- 도메인은 실제 애트리뷰트 값이 나타날 때 그 값의 합법 여부를 시스템이 검사하는데에도 이용된다.
  - 예 성별 애트리뷰트의 도메인은 '남'과 '여'로, 그 외의 값은 입력될 수 없다.

3 릴레이션의 특징

- 한 릴레이션에는 똑같은 튜플이 포함될 수 없으므로 릴레이션에 포함된 튜플들은 모두 상이하다.
  - 예 〈학생〉 릴레이션을 구성하는 홍길동 레코드는 홍길동에 대한 학적 사항을 나타내는 것으로 〈학생〉 릴레이션 내에서는 유일하다.
- 한 릴레이션에 포함된 튜플 사이에는 순서가 없다.
  - 예 〈학생〉 릴레이션에서 홍길동 레코드와 임꺽정 레코드의 위치가 바뀌어도 상관없다.
- 튜플들의 삽입, 삭제 등의 작업으로 인해 릴레이션은 시간에 따라 변한다.
  - 예 〈학생〉 릴레이션에 새로운 학생의 레코드를 삽입하거나 기존 학생에 대한 레코드를 삭제함으로써 테이블은 내용 면에서나 크기 면에서 변하게 된다.
- 릴레이션 스키마를 구성하는 속성들 간의 순서는 중요하지 않다.
  - 예 학번, 이름 등의 속성을 나열하는 순서가 이름, 학번 순으로 바뀌어도 데이터 처리에는 아무런 영향을 미치지 않는다.
- 속성의 유일한 식별을 위해 속성의 명칭은 유일해야 하지만, 속성을 구성하는 값은 동일한 값이 있을 수 있다.
  - 예 각 학생의 학년을 기술하는 속성인 '학년'은 다른 속성명들과 구분되어 유일해야 하지만 '학년' 속성에는 2, 1, 2, 4 등이 입력된 것처럼 동일한 값이 있을 수 있다.
- 릴레이션을 구성하는 튜플을 유일하게 식별하기 위해 속성들의 부분집합을 키(Key)로 설정한다.
  - 예 〈학생〉 릴레이션에서는 '학번'이나 '이름'이 튜플들을 구분하는 유일한 값인 키가 될 수 있다.
- 속성의 값은 논리적으로 더 이상 쪼갤 수 없는 원자값만을 저장한다.
  - 예 '학년'에 저장된 1, 2, 4 등은 더 이상 세분화할 수 없다.

기출문제 따라잡기

Section 078

이전기출

1. 데이터베이스에 관련된 용어의 설명으로 옳지 않은 것은?

① 튜플(Tuple) – 테이블에서 열에 해당된다.

② 애트리뷰트(Attribute) – 데이터의 가장 작은 논리적 단위로서 파일 구조상의 데이터 항목 또는 데이터 필드에 해당된다.

③ 릴레이션(Relation) – 릴레이션 스킴과 릴레이션 인스턴스로 구성된다.

④ 도메인(Domain) – 애트리뷰트가 취할 수 있는 값들의 집합이다.

튜플은 테이블에서 행에 해당합니다. 그럼 열에 해당하는 것은? 네, 애트리뷰트입니다.

이전기출

2. 릴레이션에서 속성의 수와 튜플의 수를 의미하는 것으로 순서대로 옳게 짝지어진 것은?

① Cardinality, Degree ② Domain, Degree

③ Degree, Cardinality ④ Degree, Domain

차수(Degree) = 속성(Attribute)의 수 = 열의 수, 기수(Cardinality) = 튜플(Tuple)의 수 = 행의 수! 아직 잊으면 안됩니다.

이전기출

3. 관계 데이터베이스에서 릴레이션을 구성하고 있는 각각의 속성(Attribute)에서 취할 수 있는 원자값들의 집합을 무엇이라 하는가?

① 튜플(Tuple)

② 도메인(Domain)

③ 개체 타입(Entity Type)

④ 개체 어커런스(Entity Occurrence)

도메인이란 학년은 1~4, 성별은 '남', '여'처럼 속성에 지정할 수 있는 값의 범위입니다.

이전기출

4. 다음은 무엇에 대한 설명인가?

> Each simple attribute of an entity type is associated with a value set, which specifies the set of values that may be assigned to that attribute for each individual entity.

① Domains ② Schemas

③ Attributes ④ Tuples

각각의 개체 타입에 대한 단순 속성은 어떤 값의 집합과 연관되어 있으며, 이러한 값의 집합이 각각 독립적인 개체의 속성에 할당되어 있는 일련의 값들을 정의하고 있는 것은 무엇일까요?

이전기출

5. 속성(Attribute)에 대한 설명으로 틀린 것은?

① 속성은 개체의 특성을 기술한다.

② 속성은 데이터베이스를 구성하는 가장 작은 논리적 단위이다.

③ 속성은 파일 구조상 데이터 항목 또는 데이터 필드에 해당된다.

④ 속성의 수를 "Cardinality"라고 한다.

2번 문제를 풀었다면 틀릴 수 없는 문제죠.

이전기출

6. What is the degree of a relation?

① the number of occurrences n of its relation schema

② the number of tables n of its relation schema

③ the number of attributes n of its relation schema

④ the number of key n of its relation schema

릴레이션은 테이블을 말합니다. 그리고 디그리(Degree)는 테이블에서 열에 해당하는 속성의 수를 말합니다.

이전기출

7. 다음 설명 중 릴레이션의 특징으로 옳은 내용 모두를 나열한 것은?

> ㉠ 모든 튜플은 서로 다른 값을 갖는다.
> ㉡ 각 속성은 릴레이션 내에서 유일한 이름을 가지며, 속성의 순서는 의미가 없다.
> ㉢ 하나의 릴레이션에서 튜플의 순서는 없다.
> ㉣ 모든 속성 값은 원자 값이다.

① ㉠

② ㉠, ㉡

③ ㉠, ㉡, ㉢

④ ㉠, ㉡, ㉢, ㉣

더 이상 말이 필요 없는 문제죠? 하나라도 이해되지 않는 내용이 있으면 앞 페이지를 다시 공부하고 오세요.

▶ 정답 : 1. ① 2. ③ 3. ② 4. ① 5. ④ 6. ③ 7. ④

SECTION 079 ● LIVE

관계형 데이터베이스의 제약 조건 – 키(Key)

제약 조건이란 데이터베이스에 저장되는 데이터의 정확성을 보장하기 위하여 키(Key)를 이용하여 입력되는 데이터에 제한을 주는 것으로 개체 무결성 제약, 참조 무결성 제약 등이 해당된다.

전문가의 조언

보기로 주어진 〈학생〉 릴레이션과 〈수강〉 릴레이션은 '학번'을 기준으로 '일 대 다(1:N)'의 관계를 맺고 있습니다. 〈학생〉 릴레이션과 〈수강〉 릴레이션을 통해 키의 개념과 종류를 확실하게 이해하세요.

1 키(Key)의 개념 및 종류

키는 데이터베이스에서 조건에 만족하는 튜플을 찾거나 순서대로 정렬할 때 튜플들을 서로 구분할 수 있는 기준이 되는 애트리뷰트를 말한다.

〈학생〉 릴레이션

학번	주민번호	성명
1001	010429–3******	김상욱
1002	000504–3******	임선호
1003	011215–3******	김한순
1004	001225–4******	이다해

〈수강〉 릴레이션

학번	과목명
1001	영어
1001	전산
1002	영어
1003	수학
1004	영어
1004	전산

- 키의 종류에는 후보키, 기본키, 대체키, 슈퍼키, 외래키 등이 있다.

2 후보키(Candidate Key)

후보키는 릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별하기 위해 사용하는 속성들의 부분집합, 즉 기본키로 사용할 수 있는 속성들을 말한다.

- 하나의 릴레이션내에서는 중복된 튜플들이 있을 수 없으므로 모든 릴레이션에는 반드시 하나 이상의 후보키가 존재한다.
- 후보키는 릴레이션에 있는 모든 튜플에 대해서 유일성과 최소성을 만족시켜야 한다.
  - 유일성(Unique) : 하나의 키 값으로 하나의 튜플만을 유일하게 식별할 수 있어야 한다.
  - 최소성(Minimality) : 모든 레코드들을 유일하게 식별하는 데 꼭 필요한 속성으로만 구성되어야 한다.

예 〈학생〉 릴레이션에서 '학번'이나 '주민번호'는 다른 레코드를 유일하게 구별할 수 있는 기본키로 사용할 수 있으므로 후보키이다.

3 기본키(Primary Key)

기본키는 후보키 중에서 특별히 선정된 주키(Main Key)로 중복된 값을 가질 수 없다.

- 한 릴레이션에서 특정 튜플을 유일하게 구별할 수 있는 속성이다.
- 기본키는 후보키의 성질을 갖는다. 즉, 유일성과 최소성을 가지며 튜플을 식별하기 위해 반드시 필요한 키이다.
- 기본키는 NULL 값*을 가질 수 없다. 즉 튜플에서 기본키로 설정된 속성에는 NULL 값이 있어서는 안 된다.

예 〈학생〉 릴레이션에서는 '학번'이나 '주민번호'가 기본키가 될 수 있고, 〈수강〉 릴레이션에서는 '학번'+'과목명'으로 조합해야 기본키가 만들어진다.

예 '학번'이 〈학생〉 릴레이션의 기본키로 정의되면 이미 입력된 '1001'은 다른 튜플의 '학번' 속성의 값으로 입력할 수 없다.

널 값(NULL Value)
데이터베이스에서 아직 알려지지 않거나 모르는 값으로서 '해당 없음' 등의 이유로 정보 부재를 나타내기 위해 사용하는, 이론적으로 아무것도 없는 특수한 데이터를 말합니다.

4 대체키(Alternate Key)

대체키는 후보키가 둘 이상일 때 기본키를 제외한 나머지 후보키를 의미한다.

- 보조키라고도 한다.

예 〈학생〉 릴레이션에서 '학번'을 기본키로 정의하면 '주민번호'는 대체키가 된다.

5 슈퍼키(Super Key)

슈퍼키는 한 릴레이션 내에 있는 속성들의 집합으로 구성된 키로서 릴레이션을 구성하는 모든 튜플들 중 슈퍼키로 구성된 속성의 집합과 동일한 값은 나타나지 않는다.

- 슈퍼키는 릴레이션을 구성하는 모든 튜플에 대해 유일성은 만족시키지만, 최소성*은 만족시키지 못한다.

예 〈학생〉 릴레이션에서는 '학번', '주민번호', '학번'+'주민번호', '주민번호'+'성명', '학번'+'주민번호'+'성명' 등으로 슈퍼키를 구성할 수 있다.

최소성
'학번'+'주민번호'를 사용하여 슈퍼키를 만들면 다른 튜플들과 구분할 수 있는 유일성은 만족하지만, '학번'이나 '주민번호' 하나만 가지고도 다른 튜플들을 구분할 수 있으므로 최소성은 만족시키지 못합니다.

6 외래키(Foreign Key)

외래키는 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미한다.

- 외래키는 참조되는 릴레이션*의 기본키와 대응되어 릴레이션 간에 참조 관계를 표현하는데 중요한 도구이다.
- 한 릴레이션에 속한 속성 A와 참조 릴레이션의 기본키인 B가 동일한 도메인 상에서 정의되었을 때의 속성 A를 외래키라고 한다.
- 외래키로 지정되면 참조 릴레이션의 기본키에 없는 값은 입력할 수 없다.

예 〈수강〉 릴레이션이 〈학생〉 릴레이션을 참조하고 있으므로 〈학생〉 릴레이션의 '학번'은 기본키이고, 〈수강〉 릴레이션의 '학번'은 외래키이다.

예 〈수강〉 릴레이션의 '학번'에는 〈학생〉 릴레이션의 '학번'에 없는 값은 입력할 수 없다.

참조 릴레이션
외래키를 포함하는 릴레이션이 참조하는 릴레이션이고, 대응되는 기본키를 포함하는 릴레이션이 참조 릴레이션입니다. 여기서는 〈수강〉 릴레이션이 참조하는 릴레이션이고, 〈학생〉 릴레이션이 참조 릴레이션입니다.

이전기출
1. 다음 문장의 ( ) 안 내용으로 공통 적용될 수 있는 가장 적절한 내용은 무엇인가?

> 관계형 데이터 모델에서 한 릴레이션의 ( )는 참조되는 릴레이션의 기본키와 대응되어 릴레이션 간에 참조 관계를 표현하는데 사용되는 중요한 도구이다. ( )를 포함하는 릴레이션이 참조하는 릴레이션이 되고, 대응되는 기본키를 포함하는 릴레이션이 참조 릴레이션이 된다.

① 참조 제약(Referential Constraint)
② 외래키(Foreign Key)
③ 기본키(Primary Key)
④ 참조 무결성(Referential Integrity)

참조되는 릴레이션의 기본키와 대응되어 릴레이션 간에 참조 관계를 표현하는데 사용되는 중요한 도구는 외래키입니다.

이전기출
2. 다음의 설명이 의미하는 것은?

> It is a single field or combination of fields that uniquely defines a database record. It is also called a major key. It is mainly used when records are searched or sorted.

① Foreign Key ② Alternative Key
③ Primary Key ④ Reference Key

주 키라고도 부르며, 레코드를 검색하거나 정렬할 때 주로 사용하는 키가 무엇일까요?

이전기출
3. 데이터베이스에서 사용되는 널 값(Null Value)에 대한 설명으로 옳지 않은 것은?

① 공백(Space) 또는 영(Zero)을 의미한다.
② 아직 알려지지 않거나 모르는 값이다.
③ 이론적으로 아무것도 없는 특수한 데이터를 의미한다.
④ 정보 부재를 나타내기 위해 사용한다.

널은 이론적으로 아무것도 없는 값을 의미하는데, 공백은 문자 데이터이고, 0은 숫자 데이터이니 널 값이 아니겠죠?

이전기출
4. 키는 개체 집합에서 고유하게 개체를 식별할 수 있는 속성이다. 데이터베이스에서 사용되는 키의 종류에 대한 설명 중 옳지 않은 것은?

① 후보키(Candidate Key) : 개체들을 고유하게 식별할 수 있는 속성
② 슈퍼키(Super Key) : 한 개 이상의 속성들의 집합으로 구성된 키
③ 외래키(Foreign Key) : 다른 테이블의 기본키로 사용되는 속성
④ 대체키(Alternative Key) : 후보키 중에서 대표로 선정된 키

대체키는 후보키들 중에서 기본키를 제외한 나머지 후보키들을 말합니다. 중요한 내용이죠? 혼동되면 한 번 더 공부하세요.

이전기출
5. 다음은 학생이라는 개체의 속성을 나타내고 있다. 여기서 '학과'를 기본 키로 사용하기 곤란한 이유로 가장 타당한 것은?

> 학생(학과, 성명, 학번, 세부전공, 주소, 우편번호)

① 학과는 기억하기 어렵다.
② 학과는 정렬하는데 많은 시간이 소요된다.
③ 학과는 기억 공간을 많이 필요로 한다.
④ 동일한 학과명을 가진 학생이 두 명 이상 존재할 수 있다.

기본키는 한 릴레이션에서 특정 튜플을 유일하게 구별하기 위한 속성이므로 기본키로 지정된 속성에는 동일한 값이 있어서는 안 됩니다.

이전기출
6. 다음은 무엇에 대한 설명인가?

> It is a minimal subset of attributes in a relation which uniquely identifies each tuple in the relation. It is designated as the primary.

① Super Key ② Foreign Key
③ Alternative Key ④ Candidate Key

릴레이션에서 각각의 튜플들을 유일하게 구분할 수 있는 속성들의 가장 작은 부분집합이며, 이 중 하나가 기본키로 지정되죠. 무슨 키 일까요?

이전기출
7. 릴레이션에 있는 모든 튜플에 대해 유일성은 만족시키지만 최소성은 만족시키지 못하는 키는?

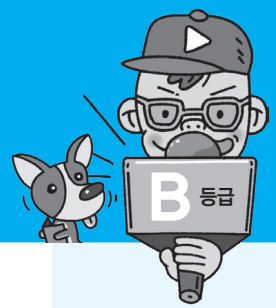
① 후보키 ② 슈퍼키
③ 기본키 ④ 외래키

유일성과 최소성을 모두 만족시키는 키는 후보키고 그 중에서 선택된 것이 기본키죠? 그럼 슈퍼키는?

▶ 정답 : 1. ② 2. ③ 3. ① 4. ④ 5. ④ 6. ④ 7. ②

SECTION 080 LIVE

관계형 데이터베이스의 제약 조건 – 무결성



1 무결성(Integrity)의 개념 및 종류

무결성이란 데이터베이스에 저장된 데이터 값과 그것이 표현하는 현실 세계의 실제 값이 일치하는 정확성을 의미한다.

- 무결성 제약 조건은 데이터베이스에 들어 있는 데이터의 정확성을 보장하기 위해 부정확한 자료가 데이터베이스 내에 저장되는 것을 방지하기 위한 제약 조건을 말한다.
- 무결성의 종류에는 개체 무결성, 도메인 무결성, 참조 무결성, 사용자 정의 무결성 등이 있다.

〈학생〉 릴레이션

학번	주민번호	성명
1001	010429–3******	김상욱
1002	000504–3******	임선호
1003	011215–3******	김한순
1004	001225–4******	이다해

〈수강〉 릴레이션

학번	과목명
1001	영어
1001	전산
1002	영어
1003	수학
1004	영어
1004	전산

2 개체 무결성(Entity Integrity, 실체 무결성)

개체 무결성은 기본 테이블의 기본키*를 구성하는 어떤 속성도 Null 값이나 중복값을 가질 수 없다는 규정이다.

예 〈학생〉 릴레이션에서 '학번'이 기본키로 정의되면 튜플을 추가할 때 '주민번호'나 '성명' 필드에는 값을 입력하지 않아도 되지만 '학번' 속성에는 반드시 값을 입력해야 한다. 또한 '학번' 속성에는 이미 한 번 입력한 속성값을 중복하여 입력할 수 없다.

3 도메인 무결성(Domain Integrity, 영역 무결성)

도메인 무결성은 주어진 속성 값이 정의된 도메인*에 속한 값이어야 한다는 규정이다.

예 〈수강〉 릴레이션의 '과목명' 속성에는 영어, 수학, 전산 세 가지만 입력되도록 유효값이 지정된 경우 반드시 해당 값만 입력해야 한다.

전문가의 조언

무결성이란 쉽게 말해 저장된 데이터베이스에 잘못된 데이터가 없다는 것을 의미합니다. 보기로 주어진 〈학생〉 릴레이션과 〈수강〉 릴레이션은 '학번'을 기준으로 일 대 다(1:N)의 관계를 맺고 있습니다. 〈학생〉 릴레이션과 〈수강〉 릴레이션을 통해 무결성 종류를 확실하게 이해하세요.

기본키(Primary Key)
기본키는 한 릴레이션에서 특정 튜플을 유일하게 구별할 수 있는 속성입니다.

도메인(Domain)
도메인은 하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자(Atomic)값들의 집합입니다.

4 참조 무결성(Referential Integrity)

참조 무결성은 외래키* 값은 Null이거나 참조 릴레이션의 기본키 값과 동일해야 한다. 즉 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다는 규정이다.

- 외래키와 참조하려는 테이블의 기본키는 도메인과 속성 개수가 같아야 한다.

예 〈수강〉 릴레이션의 '학번' 속성에는 〈학생〉 릴레이션의 '학번' 속성에 없는 값은 입력할 수 없다.

예 〈수강〉 릴레이션의 '학번'과 〈학생〉 릴레이션의 '학번' 속성에는 같은 종류의 데이가 입력되어 있어야 하며, 〈학생〉 릴레이션의 기본키가 '학번'+'이름'이었다면 〈수강〉 릴레이션의 외래키도 '학번'+'이름'으로 구성되어져야 한다.

외래키(Foreign Key)
외래키는 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미합니다.

5 사용자 정의 무결성

사용자 정의 무결성(User-Defined Integrity)은 속성 값들이 사용자가 정의한 제약 조건에 만족해야 한다는 규정이다.

전문가의 조언

데이터 품질을 확보하기 위해 애플리케이션, 데이터베이스 트리거, 제약 조건을 이용하여 데이터 무결성을 강화해야 합니다. 각 방법들의 개념과 장·단점에 대해 정리하세요.

6 데이터 무결성 강화

데이터 무결성은 데이터 품질에 직접적인 영향을 미치므로 데이터 특성에 맞는 적절한 무결성을 정의하고 강화해야 한다.

- 프로그램이 완성되고 데이터가 저장된 상태에서 무결성을 정의할 경우 많은 비용이 발생하므로 데이터베이스 구축 과정에서 정의한다.
- 데이터 무결성은 애플리케이션, 데이터베이스 트리거, 제약 조건을 이용하여 강화할 수 있다.

애플리케이션

- 데이터 생성, 수정, 삭제 시 무결성 조건을 검증하는 코드를 데이터를 조작하는 프로그램 내에 추가한다.
- 데이터베이스에서는 데이터 무결성 제약을 정의할 수 없으므로 복잡한 규칙 등은 애플리케이션 내에서 처리한다.
- **장점** : 사용자 정의 같은 복잡한 무결성 조건의 구현이 가능하다.
- **단점** : 소스 코드에 분산되어 있어 관리가 힘들고, 개별적인 시행으로 인해 적정성 검토가 어렵다.

데이터베이스 트리거*

- 트리거 이벤트*에 무결성 조건을 실행하는 절차형 SQL을 추가한다.
- **장점** : 통합 관리가 가능하고, 복잡한 요구 조건의 구현이 가능하다.
- **단점** : 운영 중 변경이 어렵고, 사용상 주의가 필요하다.

트리거(Trigger)
트리거는 데이터베이스 시스템에서 데이터의 입력, 갱신, 삭제 등의 이벤트(Event)가 발생할 때마다 자동적으로 수행되는 절차형 SQL입니다.

이벤트(Event)
이벤트는 시스템에 어떤 일이 발생한 것을 말하며, 트리거에서 이벤트는 데이터의 입력, 갱신, 삭제와 같이 데이터를 조작하는 작업이 발생했음을 의미합니다.

제약 조건

- 데이터베이스에 제약 조건을 설정하여 무결성을 유지한다.
- **장점** : 통합 관리 가능, 간단한 선언으로 구현 가능, 변경 용이, 오류 데이터 발생 방지 등이 있다.
- **단점** : 복잡한 제약 조건의 구현과 예외적인 처리가 불가능하다.

따라잡기

기출문제 따라잡기

Section 080

이전기출

1. 한 릴레이션의 기본키를 구성하는 어떠한 속성값도 널(NULL) 값이나 중복값을 가질 수 없다는 것을 의미하는 것은?

① 개체 무결성 제약 조건
② 참조 무결성 제약 조건
③ 보안 무결성 제약 조건
④ 정보 무결성 제약 조건

두 말할 필요 없는 문제죠. 꼭 숙지하세요.

이전기출

2. 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없음을 의미하는 제약 조건은?

① 개체 무결성 ② 참조 무결성
③ 보안 무결성 ④ 정보 무결성

1번 문제와 마찬가지입니다. 필수 암기 문제라고 할 수 있겠죠.

이전기출

3. 데이터 무결성과 가장 관계가 깊은 것은?

① 데이터의 안전성 ② 데이터의 공유성
③ 데이터의 중복성 ④ 데이터의 정확성

무결성은 결점이 없다는 뜻입니다. 즉 데이터가 정확하다는 뜻이죠.

이전기출

4. 데이터베이스 무결성에 관한 설명으로 옳지 않은 것은?

① 개체 무결성 규정은 한 릴레이션의 기본키를 구성하는 어떠한 속성값도 널(NULL) 값이나 중복값을 가질 수 없음을 규정하는 것이다.
② 무결성 규정에는 규정 이름, 검사 시기, 제약 조건 등을 명시한다.
③ 도메인 무결성 규정은 주어진 튜플의 값이 그 튜플이 정의된 도메인에 속한 값이어야 한다는 것을 규정하는 것이다.
④ 트리거는 트리거 조건이 만족되는 경우에 취해야 하는 조치를 명세한다.

도메인 무결성 규정은 주어진 튜플의 값에 대한 규정이 아니죠? 그렇습니다. 도메인 무결성 규정은 튜플을 구성하는 속성의 값이 그 속성이 정의된 도메인에 속한 값이어야 한다는 것을 말합니다.

출제예상

5. 외래키(Foreign Key)와 가장 직접적으로 관련된 제약 조건은 어느 것인가?

① 개체 무결성 ② 보안 무결성
③ 참조 무결성 ④ 정보 무결성

외래키는 참조되는 릴레이션의 기본키와 대응되는 키라는 것을 염두에 두고 생각해 보세요.

출제예상

6. 다음 중 속성 값들은 사용자가 정의한 제약 조건을 만족해야 한다는 규정을 의미하는 것은?

① 도메인 무결성 ② 사용자 정의 무결성
③ 참조 무결성 ④ 개체 무결성

문제 안에 답이 숨어 있네요. 문제를 다시 한 번 읽고 답을 찾아보세요!

출제예상

7. 다음 중 무결성 강화 방법과 가장 거리가 먼 것은?

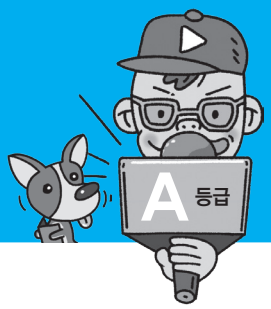
① 제약 조건
② 애플리케이션
③ 데이터베이스 트리거
④ 인덱스

무결성 강화 방법에는 '제약 조건, 애플리케이션, 데이터베이스 트리거'가 있다는 것을 기억해 두세요.

▶ 정답 : 1. ① 2. ② 3. ④ 4. ③ 5. ③ 6. ② 7. ④

SECTION 081 ● LIVE

관계대수 및 관계해석



전문가의 조언

관개대수는 관계해석과 관련하여 출제될 가능성이 높습니다. 관계해석과 비교하여 구분해 낼 수 있을 정도로 학습하세요.

1 관계대수의 개요

관계대수는 관계형 데이터베이스에서 원하는 정보와 그 정보를 검색하기 위해서 어떻게 유도하는가를 기술하는 절차적인 언어이다.

- 관계대수는 릴레이션을 처리하기 위해 연산자와 연산규칙을 제공하는 언어로 피연산자가 릴레이션이고, 결과도 릴레이션이다.
- 질의에 대한 해를 구하기 위해 수행해야 할 연산의 순서를 명시한다.
- 관계대수에는 관계 데이터베이스에 적용하기 위해 특별히 개발한 순수 관계 연산자와 수학적 집합 이론에서 사용하는 일반 집합 연산자가 있다.
- **순수 관계 연산자** : Select, Project, Join, Division
- **일반 집합 연산자** : UNION(합집합), INTERSECTION(교집합), DIFFERENCE(차집합), CARTESIAN PRODUCT(교차곱)

전문가의 조언

순수 관계 연산자는 관계 데이터베이스에 적용할 수 있도록 특별히 개발한 관계 연산자를 말합니다. 무슨 연산자를 설명하는지 알 수 있을 정도로 각 순수 관계 연산자의 기능을 학습하세요. 어렵지 않으니 예제를 통하여 기본적인 문장을 숙지하세요.

2 Select

Select는 릴레이션에 존재하는 튜플 중에서 선택 조건을 만족하는 튜플의 부분집합을 구하여 새로운 릴레이션을 만드는 연산이다.

- 릴레이션의 행(가로)에 해당하는 튜플을 구하는 것이므로 수평 연산이라고도 한다.
- 연산자의 기호는 그리스 문자 시그마(σ)를 사용한다.
- **표기 형식** : $\sigma_{\langle 조건 \rangle}(R)$
  - R은 릴레이션 이름이다.
  - 조건에서는 =, ≠, 〈, ≤, 〉, ≥ 등의 기호를 사용한 비교 연산이 허용되며, AND(∧), OR(∨), NOT(¬) 등의 논리 연산자를 사용하여 여러 개의 조건들을 하나의 조건으로 결합시킬 수도 있다.

전문가의 조언

=, ≠, 〈, ≤, 〉 ≥ 등의 비교 연산자를 θ로 일반화하여 $\sigma_{A\theta B}(R)$과 같이 표기하기도 합니다. A와 B는 릴레이션 R의 속성이고, θ는 비교 연산자입니다.

예 $\sigma_{국어>수학}$(성적)

〈성적〉 릴레이션에서 국어 점수가 수학 점수보다 큰 튜플들을 추출합니다.

예제 $\sigma_{Avg \geq 90}$ (성적) : 〈성적〉 릴레이션에서 '평균'이(Avg)이 90점 이상인 튜플들을 추출하시오.

〈성적〉

Name	Kor	Eng	Mat	Tot	Avg
고회식	100	90	100	290	96.6
김은소	80	80	100	260	86.6
최미경	100	70	80	250	83.3
김준용	90	100	90	280	93.3
윤정희	85	95	90	270	90.0

Name	Kor	Eng	Mat	Tot	Avg
고회식	100	90	100	290	96.6
김준용	90	100	90	280	93.3
윤정희	85	95	90	270	90.0

3 Project

Project는 주어진 릴레이션에서 속성 리스트(Attribute List)에 제시된 속성 값만을 추출하여 새로운 릴레이션을 만드는 연산이다. 단 연산 결과에 중복이 발생하면 중복이 제거된다.

- 릴레이션의 열(세로)에 해당하는 Attribute를 추출하는 것이므로 수직 연산자라고도 한다.
- 연산자의 기호는 그리스 문자 파이(π)를 사용한다.
- **표기 형식** : $\pi_{\langle 속성리스트 \rangle}(R)$
  - R은 릴레이션 이름이다.

예제 $\pi_{Name,\ Avg}$(성적) : 〈성적〉 릴레이션에서 'Name'과 'Avg' 속성을 추출하시오.

〈성적〉

Name	Kor	Eng	Mat	Tot	Avg
고회식	100	90	100	290	96.6
김은소	80	80	100	260	86.6
최미경	100	70	80	250	83.3
김준용	90	100	90	280	93.3
윤정희	85	95	90	270	90.0

➡

Name	Avg
고회식	96.6
김은소	86.6
최미경	83.3
김준용	93.3
윤정희	90.0

4 Join

Join은 공통 속성을 중심으로 두 개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 연산이다.

- Join의 결과로 만들어진 릴레이션의 차수는 조인된 두 릴레이션의 차수를 합한 것과 같다.
- Join의 결과는 Cartesian Product(교차곱)*를 수행한 다음 Select*를 수행한 것과 같다.
- 연산자의 기호는 ⋈를 사용한다.
- **표기 형식** : $R \bowtie_{키속성r=키속성s} S$
  - 키 속성 r은 릴레이션 R의 속성이고, 키 속성 s는 릴레이션 S의 속성이다.

Cartesian Product(교차곱)
Cartesian Product 연산은 두 릴레이션에 존재하는 모든 튜플들을 대응시켜 새로운 릴레이션을 만드는 연산으로, 연산의 결과 차수는 두 릴레이션의 차수를 합한 것과 같고 튜플은 두 릴레이션의 튜플 수를 곱한 것과 같습니다. 즉 Cartesian Product의 결과는 두 릴레이션을 연결하여 나타낼 수 있는 모든 튜플들을 표현할 수 있으므로 여기에서 필요한 튜플만 선별하는 Select 연산을 수행하면 Join 연산의 결과와 같아지는 것입니다.

예제 성적 $\bowtie_{No=No}$ 학적부 : 〈성적〉 릴레이션과 〈학적부〉 릴레이션을 'No' 속성을 기준으로 합치시오.

〈성적〉

No	Name	Kor	Eng	Mat	Tot	Avg
9801	고회식	100	90	100	290	96.6
9802	김은소	80	80	100	260	86.6
9803	최미경	100	70	80	250	83.39
9804	김준용	90	100	90	280	93.3
9805	윤정희	85	95	90	270	90.0

〈학적부〉

No	Addr
9801	망원동
9802	서교동
9803	성산동
9804	합정동
9805	공덕동

성적.No	Name	Kor	Eng	Mat	Tot	Avg	학적부.No	Addr
9801	고회식	100	90	100	290	96.6	9801	망원동
9802	김은소	80	80	100	260	86.6	9802	서교동
9803	최미경	100	70	80	250	83.39	9803	성산동
9804	김준용	90	100	90	280	93.3	9804	합정동
9805	윤정희	85	95	90	270	90.0	9805	공덕동

해설 위 그림에서와 같이 Join 조건이 '='일 때 동일한 속성이 2번 나타나게 된다.

잠깐만요

자연 조인(Natural Join)

- 조인 조건이 '='일 때 동일한 속성이 두 번 나타나게 되는데, 이중 중복된 속성을 제거하여 같은 속성을 한 번만 표기하는 방법을 자연(Natural) 조인이라고 합니다.
- 예를 들어, **예제**의 Join 연산 결과에는 조인에 사용된 'No' 속성이 두 번 표기되었는데 이것은 의미상 아무런 도움을 주지 않습니다. 이런 경우 다음과 같은 자연 조인 연산을 통해 'No' 속성이 한 번만 표기되게 할 수 있습니다.
  성적$\bowtie_{No}$ 학적부

※ 자연 조인이 성립되려면 두 릴레이션의 속성명과 도메인이 같아야 합니다.

5 Division

Division은 X⊃Y인 두 개의 릴레이션 R(X)와 S(Y)가 있을 때, R의 속성이 S의 속성값을 모두 가진 튜플에서 S가 가진 속성을 제외한 속성만을 구하는 연산이다.

- 연산자의 기호는 ÷를 사용한다.
- **표기 형식** : R [속성r ÷ 속성s] S
  - 속성 r은 릴레이션 R의 속성, 속성 s는 릴레이션 S의 속성, 속성 r과 속성 s는 동일 속성값을 가지는 속성이어야 한다.

〈구입자〉

번호	이름	구입품코드
1	고회식	A
1	고회식	B
2	김준용	A
2	김준용	B
2	김준용	C
4	윤정희	C

〈생산품1〉

생산품코드
A
B

〈생산품2〉

이름	생산품코드
김준용	A
김준용	B

예제 1 구입자[구입품코드 ÷ 생산품코드]생산품1의 결과는 다음과 같다.

번호	이름
1	고회식
2	김준용

해설 〈구입자〉 릴레이션에서 '구입품코드' 속성을 제외하고 '번호'와 '이름'을 추출한다. 단 〈생산품1〉 릴레이션의 모든 튜플과 연관되어 있는 튜플만 추출한다.

예제 2 구입자[이름, 구입품 코드 ÷ 이름, 생산품코드]생산품2의 결과는 다음과 같다.

번호
2

해설 〈구입자〉 릴레이션에서 '이름', '구입품코드' 속성을 제외하고 번호만 추출한다. 단 〈생산품2〉 릴레이션의 모든 튜플과 연관되어 있는 튜플만 추출한다.

6 일반 집합 연산자

일반 집합 연산자는 수학적 집합 이론에서 사용하는 연산자로서 릴레이션 연산에도 그대로 적용할 수 있다.

- 일반 집합 연산자 중 합집합(UNION), 교집합(INTERSECTION), 차집합(DIFFERENCE)을 처리하기 위해서는 합병 조건*을 만족해야 한다.
- 합병 가능한 두 릴레이션 R과 S가 있을 때 각 연산의 특징을 요약하면 다음과 같다.

연산자	기능 및 수학적 표현	카디널리티
합집합 UNION ∪	• 두 릴레이션에 존재하는 튜플의 합집합을 구하되, 결과로 생성된 릴레이션에서 중복되는 튜플은 제거되는 연산이다. • $R \cup S = \{t \mid t \in R \vee t \in S\}$ ※ t는 릴레이션 R 또는 S에 존재하는 튜플이다.	• $\lvert R \cup S \rvert \le \lvert R \rvert + \lvert S \rvert$ • 합집합의 카디널리티는 두 릴레 이션 카디널리티의 합보다 크지 않다.
교집합 INTERSECTION ∩	• 두 릴레이션에 존재하는 튜플의 교집합을 구하는 연산이다. • $R \cap S = \{t \mid t \in R \wedge t \in S\}$ ※ t는 릴레이션 R 그리고 S에 동시에 존재하는 튜플이다.	• $\lvert R \cap S \rvert \le MIN\{\lvert R \rvert, \lvert S \rvert\}$ • 교집합의 카디널리티는 두 릴레이션 중 카디널리티가 적은 릴레이션의 카디널리티보다 크지 않다.

전문가의 조언

일반 집합 연산자는 수학에서의 집합 연산자의 기능과 동일한 기능을 합니다. 연산이 가능하게 하는 합병 조건과 연산 후의 카디널리티(Cardinality)가 어떻게 변하는지 학습하세요.

합병 조건

합병 조건은 합병하려는 두 릴레이션 간에 속성의 수가 같고, 대응되는 속성별로 도메인이 같아야 합니다. 즉, 릴레이션 R과 S가 합병이 가능하다면, 릴레이션 R의 i번째 속성과 릴레이션 S의 i번째 속성의 도메인이 서로 같아야 합니다. 그러나 속성의 이름이 같아야 되는 것은 아닙니다.

차집합 DIFFERENCE −	• 두 릴레이션에 존재하는 튜플의 차집합을 구하는 연산이다. • $R - S = \{ t \mid t \in R \wedge t \notin S \}$ ※ t는 릴레이션 R에는 존재하고 S에 없는 튜플이다.	• $\lvert R-S \rvert \le \lvert R \rvert$ • 차집합의 카디널리티는 릴레이션 R의 카디널리티 보다 크지 않다.
교차곱 CARTESIAN PRODUCT ×	• 두 릴레이션에 있는 튜플들의 순서쌍을 구하는 연산이다. • $R \times S = \{ r \cdot s \mid r \in R \wedge s \in S \}$ ※ r은 R에 존재하는 튜플이고, s는 S에 존재하는 튜플이다.	• $\lvert R \times S \rvert = \lvert R \rvert \times \lvert S \rvert$ • 교차곱은 두 릴레이션의 카디널리티를 곱한 것과 같다.

〈사원〉

사번	이름
1	고회식
2	김준용

〈직원〉

사번	이름
2	김준용
3	윤정희

예제 1 $\pi_{이름}$(사원) ∪ $\pi_{이름}$(직원) : 〈사원〉 릴레이션과 〈직원〉 릴레이션에서 '이름'을 추출한 것의 합집합을 구한다.

이름
고회식
김준용
윤정희

예제 2 $\pi_{이름}$(사원) ∩ $\pi_{이름}$(직원) : 〈사원〉 릴레이션과 〈직원〉 릴레이션에서 '이름'을 추출한 것의 교집합을 구한다.

이름
김준용

예제 3 $\pi_{이름}$(사원) − $\pi_{이름}$(직원) : 〈사원〉 릴레이션과 〈직원〉 릴레이션에서 '이름'을 추출한 것의 차집합을 구한다.

이름
고회식

예제 4 $\pi_{이름}$(사원) × $\pi_{이름}$(직원) : 〈사원〉 릴레이션과 〈직원〉 릴레이션에서 '이름'을 추출한 것의 교차곱을 구한다.

사원.이름	직원.이름
고회식	김준용
고회식	윤정희
김준용	김준용
김준용	윤정희

7 관계해석(Relational Calculus)

관계해석은 관계 데이터 모델의 제안자인 코드(E. F. Codd)가 수학의 Predicate Calculus(술어 해석)에 기반을 두고 관계 데이터베이스를 위해 제안했다.

- 관계해석은 관계 데이터의 연산을 표현하는 방법으로, 원하는 정보를 정의할 때는 계산 수식을 사용한다.
- 관계해석은 원하는 정보가 무엇이라는 것만 정의하는 비절차적 특성을 지닌다.
- 튜플 관계해석과 도메인 관계해석이 있다.
- 기본적으로 관계해석과 관계대수는 관계 데이터베이스를 처리하는 기능과 능력면에서 동등하며, 관계대수로 표현한 식은 관계해석으로 표현할 수 있다.
- 질의어로 표현한다.

전문가의 조언

마찬가지로 관계대수와 구분할 수 있도록 학습하세요.

따라잡기

기출문제 따라잡기

Section 081

이전기출

1. 조건을 만족하는 릴레이션의 수평적 부분집합으로 구성하며, 연산자의 기호는 그리스 문자 시그마(σ)를 사용하는 관계대수 연산자는?

① Select 연산자 ② Project 연산자
③ Join 연산자 ④ Division 연산자

각 관계대수 연산자의 특징은 확실하게 구분할 수 있어야 합니다. 간단하게 다시 한 번 살펴볼까요?
- Select : 조건에 맞는 튜플을 구하는 수평적 연산
- Project : 리스트로 주어진 속성만 구하는 수직적 연산
- Join : 공통 속성을 중심으로 릴레이션의 병합
- Division : 같은 튜플들 중에서 비교 속성을 제외한 속성만 구하는 연산

이전기출

2. 관계형 대수의 연산자가 아닌 것은?

① JOIN ② PROJECT
③ PRODUCT ④ PART

순수 관계 연산자 네 가지와 각각을 표기하는 기호! 잊으면 안되겠죠.

이전기출

3. 테이블에서 특정 속성에 해당하는 열을 선택하는 데 사용되며 결과로는 릴레이션의 수직적 부분집합에 해당하는 관계대수 연산자는?

① Project 연산자 ② Join 연산자
③ Division 연산자 ④ Select 연산자

수평은 Select, 수직은 Project!

출제예상

4. 릴레이션 R의 차수(Degree)가 3, 카디널리티(Cardinality)가 3, 릴레이션 S의 차수가 4, 카디널리티가 4일 때, 두 릴레이션을 카티션 프로덕트(Cartesian Product)한 결과 릴레이션의 차수와 카디널리티는?

① 4, 4 ② 7, 7
③ 7, 12 ④ 12, 12

카티션 프로덕트(Cartesian Product), 즉 교차곱은 두 릴레이션의 차수(Degree, 속성의 수)는 더하고, 카디널리티(Cardinality, 튜플의 수)는 곱하면 됩니다. 즉 차수는 3 + 4 = 7이고, 카디널리티는 3 × 4 = 12입니다.

출제예상

5. 관계대수와 관계해석에 대한 설명으로 옳지 않은 것은?

① 관계대수는 원하는 정보가 무엇이라는 것만 정의하는 비절차적 특징을 가지고 있다.
② 기본적으로 관계대수와 관계해석은 관계 데이터베이스를 처리하는 기능과 능력면에서 동등하다.
③ 관계해석에는 튜플 관계해석과 도메인 관계해석이 있다.
④ 관계해석은 수학의 프레디킷 해석(Predicate Calculus)에 기반을 두고 있다.

①번은 관계해석에 대한 설명이죠? 관계대수는 원하는 그 정보를 검색하기 위해서 어떻게 유도하는가를 기술하는 절차적인 언어입니다.

▶ 정답 : 1. ① 2. ④ 3. ① 4. ③ 5. ①

이전기출
6. 다음의 관계대수 문장을 SQL로 표현한 것으로 옳은 것은?

$$\pi_{name,\ dept}(\sigma_{year\ =\ 3}(student))$$

① SELECT name, dept FROM student HAVING year = 3;
② SELECT name, dept FROM student WHERE year = 3;
③ SELECT student FROM name, dept WHERE year = 3;
④ SELECT student FROM name, dept HAVING year = 3;

관계대수에서 π는 열을 추출하는 Project 연산을 의미하고, σ는 행을 추출하는 Select 연산을 의미합니다. 각 연산을 분리해서 살펴보면 다음과 같습니다.
- $\sigma_{year\ =\ 3}$ (student)
  – Select * from student where year = 3;
  〈student〉 테이블에서 'year' 속성의 값이 3인 튜플들의 모든 속성을 추출하라.
- $\pi_{name,\ dept}$ ($\sigma_{year\ =\ 3}$ (student))
  – Select name, dept from student where year = 3;
  〈student〉 테이블에서 'year' 속성의 값이 3인 튜플의 'name', 'dept' 속성만 추출하라.

※ SQL 명령어는 3장 SQL 응용에서 자세하게 배웁니다.

이전기출
7. 다음 관계대수의 의미로 가장 타당한 것은?

$$\pi_{이름}(\sigma_{학과='컴퓨터'}(학생))$$

① 이름, 학과, 컴퓨터를 속성으로 하는 학생 테이블을 생성하라.
② 컴퓨터과 학생의 이름을 검색하라.
③ 컴퓨터과 학생의 이름을 삭제하라.
④ 학과의 이름을 컴퓨터로 변경하라.

이런 문제는 각각을 분리해서 이해하면 쉽습니다.
- $\pi_{이름}$: '이름'을 표시합니다.
- $\sigma_{학과='컴퓨터'}$: '학과'가 '컴퓨터'인 자료만을 대상으로 합니다.
- (학생) : 〈학생〉 테이블의 자료를 검색합니다.

결국 컴퓨터과 학생의 '이름'을 검색하라는 의미입니다.

이전기출
8. 관계해석(Relational Calculus)에 대한 설명으로 잘못된 것은?

① 튜플 관계해석과 도메인 관계해석이 있다.
② 원하는 정보와 그 정보를 어떻게 유도하는가를 기술하는 절차적인 특성을 가진다.
③ 기본적으로 관계해석과 관계대수는 관계 데이터베이스를 처리하는 기능과 능력 면에서 동등하다.
④ 수학의 Predicate Calculus에 기반을 두고 있다.

관계해석은 비절차적인 언어로 원하는 결과 정보만 기술해 줍니다. 절차적인 언어는 관계대수죠.

이전기출
9. 관계 데이터 연산에 관한 내용으로 적당하지 않는 것은?

① 관계대수는 원하는 정보와 그 정보를 어떻게 유도하는가를 기술하는 절차적인 방법이다.
② 관계해석은 원하는 정보가 무엇이라는 것만 정의하는 비절차적 특성을 지닌다.
③ 관계해석에는 튜플 관계해석(Tuple Relational Calculus)과 도메인 관계해석(Domain Relational Calculus)이 있다.
④ 관계해석으로 표현한 식은 관계대수로 표현할 수 없다.

관계해석으로 표현한 식은 관계대수로 표현할 수 있습니다.

이전기출
10. 관계대수의 조인 연산에서 결과가 동일한 애트리뷰트는 하나만 나타내는 것을 무엇이라고 하는가?

① 택일 조인 ② 자연 조인
③ 완전 조인 ④ 2차 조인

똑 같은 게 여럿이면 하나만 쓰는 게 자연스럽죠?

▶ 정답 : 6. ② 7. ② 8. ② 9. ④ 10. ②

SECTION 082 LIVE

정규화(Normalization)

1 정규화의 개요

정규화란 함수적 종속성 등의 종속성 이론을 이용하여 잘못 설계된 관계형 스키마를 더 작은 속성의 세트로 쪼개어 바람직한 스키마로 만들어 가는 과정이다.

- 하나의 종속성이 하나의 릴레이션에 표현될 수 있도록 분해해가는 과정이라 할 수 있다.
- 정규형에는 제1정규형, 제2정규형, 제3정규형, BCNF형, 제4정규형, 제5정규형이 있으며, 차수가 높아질수록 만족시켜야 할 제약 조건이 늘어난다.
- 정규화는 데이터베이스의 논리적 설계 단계에서 수행한다.
- 정규화는 논리적 처리 및 품질에 큰 영향을 미친다.
- 정규화된 데이터 모델은 일관성, 정확성, 단순성, 비중복성, 안정성 등을 보장한다.
- 정규화 수준이 높을수록 유연한 데이터 구축이 가능하고 데이터의 정확성이 높아지는 반면 물리적 접근이 복잡하고 너무 많은 조인으로 인해 조회 성능이 저하된다.

전문가의 조언

정규화는 데이터의 중복성을 최소화하고 일관성 등을 보장하여 데이터베이스의 품질을 보장하고 성능의 향상을 위해 수행합니다. 정규화의 개념이나 목적이 아닌 것을 구분해 낼 수 있을 정도로 학습하세요.

2 정규화의 목적

- 데이터 구조의 안정성 및 무결성을 유지한다.
- 어떠한 릴레이션이라도 데이터베이스 내에서 표현 가능하게 만든다.
- 효과적인 검색 알고리즘을 생성할 수 있다.
- 데이터 중복을 배제하여 이상(Anomaly)의 발생 방지 및 자료 저장 공간의 최소화가 가능하다.
- 데이터 삽입 시 릴레이션을 재구성할 필요성을 줄인다.
- 데이터 모형의 단순화가 가능하다.
- 속성의 배열 상태 검증이 가능하다.
- 개체와 속성의 누락 여부 확인이 가능하다.
- 자료 검색과 추출의 효율성을 추구한다.

3 이상(Anomaly)의 개념 및 종류

정규화를 거치지 않으면 데이터베이스 내에 데이터들이 불필요하게 중복되어 릴레이션 조작 시 예기치 못한 곤란한 현상이 발생하는데, 이를 이상(Anomaly)이라 하며 삽입 이상, 삭제 이상, 갱신 이상이 있다.

- **삽입 이상(Insertion Anomaly)** : 릴레이션에 데이터를 삽입할 때 의도와는 상관없이 원하지 않은 값들도 함께 삽입되는 현상이다.

전문가의 조언

이상이 발생하는 원인과 이상의 종류를 기억하세요. 각 이상의 의미는 용어 그대로 이므로 한 번만 읽어보면 바로 이해됩니다.

- **삭제 이상(Deletion Anomaly)** : 릴레이션에서 한 튜플을 삭제할 때 의도와는 상관없는 값들도 함께 삭제되는 연쇄 삭제 현상이 일어나는 현상이다.
- **갱신 이상(Update Anomaly)** : 릴레이션에서 튜플에 있는 속성값을 갱신할 때 일부 튜플의 정보만 갱신되어 정보에 모순이 생기는 현상이다.

4 정규화의 원칙

- 정보의 무손실 표현, 즉 하나의 스키마를 다른 스키마로 변환할 때 정보의 손실이 있어서는 안 된다.
- 분리의 원칙, 즉 하나의 독립된 관계성은 하나의 독립된 릴레이션으로 분리시켜 표현해야 한다.
- 데이터의 중복성이 감소되어야 한다.

전문가의 조언

정규화하는 과정을 생략하고 각 단계의 정규화 형태에 대한 정의만 수록했습니다. 각 단계별 특징을 구분할 수 있도록 정리하세요.

5 정규화 과정

1NF(제1정규형)

1NF는 릴레이션에 속한 모든 도메인(Domain)이 원자값(Atomic Value)만으로 되어 있는 정규형이다. 즉, 릴레이션의 모든 속성 값이 원자 값으로만 되어 있는 정규형이다.

- 릴레이션의 모든 속성이 단순 영역에서 정의된다.

2NF(제2정규형)

2NF는 릴레이션 R이 1NF이고, 기본키가 아닌 모든 속성이 기본키에 대하여 완전 함수적 종속을 만족하는 정규형이다.

함수적 종속 / 완전/부분 함수적 종속 및 이해

함수적 종속(Functional Dependency)

- 함수적 종속은 데이터들이 어떤 기준값에 의해 종속되는 것을 의미합니다.
- 예를 들어 〈수강〉 릴레이션이 (학번, 이름, 과목명)으로 되어 있을 때, '학번'이 결정되면 '과목명'에 상관없이 '학번'에는 항상 같은 '이름'이 대응됩니다. '학번'에 따라 '이름'이 결정될 때 '이름'을 '학번'에 함수 종속적이라고 하며 '학번 → 이름'과 같이 씁니다.

완전 함수적 종속

어떤 테이블 R에서 속성 A가 다른 속성 집합 B 전체에 대해 함수적 종속이지만 속성 집합 B의 어떠한 진부분 집합 C(즉, C ⊂ B)에는 함수적 종속이 아닐 때 속성 A는 속성 집합 B에 완전 함수적 종속이라고 합니다.

부분 함수적 종속

어떤 테이블 R에서 속성 A가 다른 속성 집합 B 전체에 대해 함수적 종속이면서 속성 집합 B의 어떠한 진부분 집합에도 함수적 종속일 때, 속성 A는 속성 집합 B에 부분 함수적 종속이라고 합니다.

완전/부분 함수적 종속의 이해

- 완전 함수적 종속은 어떤 속성이 기본키에 대해 완전히 종속적일 때를 말합니다.
- 예를 들어 〈수강〉 릴레이션이 (학번, 과목명, 성적, 학년)으로 되어 있고 (학번, 과목명)이 기본키일 때,

'성적'은 '학번'과 '과목명'이 같을 경우에는 항상 같은 '성적'이 옵니다. 즉 '성적'은 '학번'과 '과목명'에 의해서만 결정되므로 '성적'은 기본키(학번, 과목명)에 완전 함수적 종속이 되는 것입니다.

- 반면에 '학년'은 '과목명'에 관계없이 '학번'이 같으면 항상 같은 '학년'이 옵니다. 즉 기본키의 일부인 '학번'에 의해서 '학년'이 결정되므로 '학년'은 부분 함수적 종속이라고 합니다.

3NF(제3정규형)

3NF는 릴레이션 R이 2NF이고, 기본키가 아닌 모든 속성이 기본키에 대해 이행적 종속*을 만족하지 않는 정규형이다.

- 무손실 조인 또는 종속성 보존을 저해하지 않고도 항상 3NF 설계를 얻을 수 있다.

BCNF(Boyce-Codd 정규형)

BCNF는 릴레이션 R에서 결정자*가 모두 후보키(Candidate Key)인 정규형이다.

- 3NF에서 후보키가 여러 개 존재하고 서로 중첩되는 경우에 적용하는, 강한 제3정규형이라고도 한다.
- 모든 BCNF(Boyce-Codd Normal Form)가 종속성을 보존하는 것은 아니다.
- BCNF의 제약 조건
  - 키가 아닌 모든 속성은 각 키에 대하여 완전 종속해야 한다.
  - 키가 아닌 모든 속성은 그 자신이 부분적으로 들어가 있지 않은 모든 키에 대하여 완전 종속해야 한다.
  - 어떤 속성도 키가 아닌 속성에 대해서는 완전 종속할 수 없다.

4NF(제4정규형)

4NF는 릴레이션 R에 다치 종속* A ⟶ B가 성립하는 경우 R의 모든 속성이 A에 함수적 종속 관계를 만족하는 정규형이다.

5NF(제5정규형, PJ/NF)

5NF는 릴레이션 R의 모든 조인 종속*이 R의 후보키를 통해서만 성립되는 정규형이다.

잠깐만요

정규화 과정 정리

비정규 릴레이션
↓ 도메인이 원자값
1NF
↓ 부분적 함수 종속 제거
2NF
↓ 이행적 함수 종속 제거
3NF
↓ 결정자이면서 후보키가 아닌 것 제거
BCNF
↓ 다치 종속
4NF
↓ 조인 종속성 이용
5NF

정규화 단계 암기 요령

두부를 좋아하는 정규화가 두부가게에 가서 가게에 있는 두부를 다 달라고 말하니 주인이 깜짝 놀라며 말했다.

두부이걸다줘? ≒ 도부이결다조

도메인이 원자값
부분적 함수 종속 제거
이행적 함수 종속 제거
결정자이면서 후보키가 아닌 것 제거
다치 종속
조인 종속성 이용

이행적 종속(Transitive Dependency) 관계
A → B이고 B → C일 때 A → C를 만족하는 관계를 의미합니다.

결정자/종속자
결정자(Determinant)는 속성 간의 종속성을 규명할 때 기준이 되는 값이고, 종속자(Dependent)는 결정자의 값에 의해 정해지는 값을 의미합니다.
예 '학번'에 따라 '이름'이 결정되는 '학번 → 이름'일 때 '학번'을 결정자라하고, '이름'을 종속자라 합니다.

다치 종속(Multi Valued Dependency, 다가 종속)
A, B, C 3개의 속성을 가진 릴레이션 R에서 어떤 복합 속성(A, C)에 대응하는 B 값의 집합이 A 값에만 종속되고 C 값에는 무관하면, B는 A에 다치 종속이라 하고, 'A ⟶ B'로 표기합니다.

조인 종속(Join Dependency)
어떤 릴레이션 R의 속성에 대한 부분집합 A, B, …, C가 있다고 해봅시다. 이때 만일 릴레이션 R이 자신의 프로젝션(Projection) A, B, …, C를 모두 조인한 결과가 자신과 동일한 경우 릴레이션 R은 조인 종속(A, B, …, C)을 만족한다고 합니다.

이전기출
1. 정규화의 목적으로 틀린 것은?

① 어떠한 릴레이션이라도 데이터베이스 내에서 표현 가능하게 만든다.
② 데이터 삽입 시 릴레이션을 재구성할 필요성을 줄인다.
③ 중복을 배제하여 삽입, 삭제, 갱신 이상의 발생을 도모한다.
④ 효과적인 검색 알고리즘을 생성할 수 있다.

정규화의 목적 중 하나는 중복을 배제하여 이상(Anomaly)의 발생을 방지하는 것입니다.

이전기출
2. 관계 데이터베이스의 정규화에 대한 설명으로 옳지 않은 것은?

① 정규화를 거치지 않으면 여러 가지 상이한 종류의 정보를 하나의 릴레이션으로 표현하여 그 릴레이션을 조작할 때 이상(Anomaly) 현상이 발생할 수 있다.
② 정규화의 목적은 각 릴레이션에 분산된 종속성을 하나의 릴레이션에 통합하는 것이다.
③ 이상(Anomaly) 현상은 데이터들 간에 존재하는 함수 종속이 하나의 원인이 될 수 있다.
④ 정규화가 잘못되면 데이터의 불필요한 중복이 야기되어 릴레이션을 조작할 때 문제가 발생할 수 있다.

이상을 해결하기 위해 정규화를 수행하는데, 정규화의 목적은 각 릴레이션에 분산된 종속성을 하나의 릴레이션에 통합하는 것이 아니라 속성들 간의 종속 관계를 분석하여 한 개의 릴레이션을 여러 개의 릴레이션으로 분해하는 것입니다.

이전기출
3. 정규화 과정에서 발생하는 이상(Anomaly)에 관한 설명으로 옳지 않은 것은?

① 이상은 속성들 간에 존재하는 여러 종류의 종속 관계를 하나의 릴레이션에 표현할 때 발생한다.
② 정규화는 이상을 제거하기 위해서 중복성 및 종속성을 배제시키는 방법으로 사용한다.
③ 1NF의 이상을 해결하기 위해서는 프로젝션에 의해 릴레이션을 분해하여 부분 함수 종속을 제거해야 한다.
④ 속성들 간의 종속 관계를 분석하여 여러 개의 릴레이션을 하나로 결합하여 이상을 해결한다.

정규화는 속성들 간의 종속 관계를 분석하여 한 개의 릴레이션을 여러 개의 릴레이션으로 분해하여 이상을 해결합니다.

이전기출
4. 관계형 데이터베이스의 릴레이션을 조작할 때 발생하는 이상(Anomaly) 현상에 관한 설명으로 적절하지 않은 것은?

① 데이터의 종속으로 인해 발생하는 이상 현상에는 삭제 이상, 삽입 이상, 갱신 이상이 있다.
② 릴레이션의 한 튜플을 삭제함으로써 연쇄 삭제로 인해 정보의 손실을 발생시키는 현상이 삭제 이상이다.
③ 데이터를 삽입할 때 불필요한 데이터가 함께 삽입되는 현상을 삽입 이상이라 한다.
④ 튜플 중에서 일부 속성을 갱신함으로써 정보의 모순성이 발생하는 현상이 갱신 이상이다.

이상은 데이터의 종속으로 인해 발생하는 것이 아니고 속성(필드) 간 종속 혹은 데이터(튜플)의 중복으로 인해 발생하는 것입니다. ①번에서의 데이터의 종속이란 튜플 간의 종속을 말하는데, 튜플들 사이에는 종속이 존재하지 않습니다.

이전기출
5. 제 1정규형에서 제 2정규형 수행 시의 작업으로 옳은 것은?

① 이행적 함수 종속성 제거
② 다치 종속 제거
③ 모든 결정자가 후보키가 되도록 분해
④ 부분 함수 종속성 제거

'두부이결다줘 = 도부이결다조'에서 '부'에 해당합니다.

이전기출
6. 다음의 조건을 모두 만족하는 정규형은?

도메인은 원자 값이고 기본키가 아닌 모든 속성들이 기본키에 대해 완전 함수 종속적이며, 이행적 함수 종속 관계는 제거되었다.

① 제 1정규형
② 제 2정규형
③ 제 3정규형
④ 제 1정규형과 제2정규형

모든 도메인은 원자 값이므로 제 1정규형을 만족하고, 키가 아닌 모든 속성들이 기본키에 대해 완전 함수 종속이므로 제 2정규형을 만족합니다. 그리고 이행적 함수 종속 관계가 제거 되었으므로 제 3정규형을 만족합니다.

기출문제 따라잡기

Section 082

이전기출

7. 어떤 릴레이션 R에서 X와 Y를 각각 R의 애트리뷰트 집합의 부분집합이라고 할 경우 애트리뷰트 X의 값 각각에 대해 시간에 관계없이 항상 애트리뷰트 Y의 값이 오직 하나만 연관되어 있을 때 Y는 X에 함수 종속이라고 한다. 이 함수 종속의 표기로 옳은 것은?

① Y → X　　② Y ⊂ X

③ X → Y　　④ X ⊂ Y

어렵게 생각하지 마세요. 항상 X에 따라 Y가 결정될 때 Y를 X에 함수 종속적이라고 하며 X → Y와 같이 씁니다. 꼭 기억해 두세요.

이전기출

8. 다음 중 BCNF를 만족하기 위한 조건 모두로 옳게 짝지어진 것은?

㉠ 결정자이면서 후보키가 아닌 것 제외 ㉡ 이행적 함수 종속 제거 ㉢ 부분적 함수 종속 제거 ㉣ 도메인이 원자값

① ㉠　　② ㉠, ㉡, ㉢, ㉣

③ ㉡, ㉢, ㉣　　④ ㉠, ㉣

제1정규형 → 제2정규형 → 제3정규형을 거쳐 BCNF가 되므로 제1정규형(㉣), 제2정규형(㉢), 제3정규형(㉡), BCNF(㉠)가 되기 위한 조건을 모두 만족해야 합니다.

이전기출

9. 어떤 릴레이션 R에 존재하는 모든 조인 종속성이 릴레이션 R의 후보키를 통해서만 성립된다. 이 릴레이션 R은 어떤 정규형의 릴레이션인가?

① 제3정규형　　② 보이스-코드 정규형

③ 제4정규형　　④ 제5정규형

'두부이걸다줘 = 도부이결다조'에서 '조'에 해당합니다. 조인 종속성을 이용하는 정규화는 5NF입니다.

이전기출

10. 데이터의 중복으로 인해 릴레이션 조작 시 예상하지 못한 곤란한 현상이 발생한다. 이를 무엇이라고 하는가?

① Normalization　　② Degree

③ Cardinality　　④ Anomaly

이 문제 틀렸나요? 거 참 이상(Anomaly)하네요.

이전기출

11. 정규화에 대한 설명으로 옳지 않은 것은?

① 정규화 하는 것은 테이블을 결합하여 종속성을 증가시키는 것이다.

② 제 2정규형은 반드시 제 1정규형을 만족해야 한다.

③ 제 1정규형은 릴레이션에 속한 모든 도메인이 원자값만으로 되어 있는 릴레이션이다.

④ BCNF는 강한 제 3정규형이라고도 한다.

정규화하는 것은 테이블을 결합하여 종속성을 증가시키는 것이 아니라 더 작은 테이블로 분해해 가면서 종속성을 제거하는 것입니다.

이전기출

12. 이행적 함수 종속 관계를 의미하는 것은?

① A → B이고 B → C 일 때, A → C를 만족하는 관계

② A → B이고 B → C 일 때, C → A를 만족하는 관계

③ A → B이고 B → C 일 때, B → A를 만족하는 관계

④ A → B이고 B → C 일 때, C → B를 만족하는 관계

이것을 제거한 것이 제 3정규형입니다.

이전기출

13. 제 2정규형에서 제 3정규형이 되기 위한 조건은?

① 이행적 함수 종속 제거

② 부분적 함수 종속 제거

③ 다치 종속 제거

④ 결정자이면서 후보 키가 아닌 것 제거

12번 문제에서 뭐라고 했죠?

이전기출

14. 정규화를 거치지 않으면 릴레이션 조작 시 데이터 중복에 따른 예기치 못한 곤란한 현상이 발생할 수 있다. 이러한 이상(Anomaly) 현상의 종류에 해당하지 않는 것은?

① 삭제 이상　　② 삽입 이상

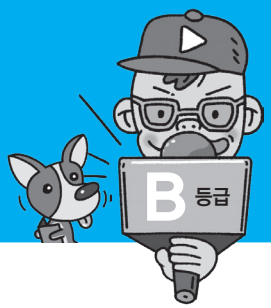
③ 갱신 이상　　④ 조회 이상

이 문제를 틀렸다면 앞의 문제들을 제대로 안 푼 거죠? 1, 4번 문제를 다시 읽어 보세요.

▶ 정답 : 1. ③ 2. ② 3. ④ 4. ① 5. ④ 6. ③ 7. ③ 8. ② 9 ④ 10. ④ 11. ① 12. ① 13. ① 14. ④

SECTION 083 ● LIVE

반정규화(Denormalization)



전문가의 조언

반정규화의 개념과 반정규화의 4가지 방법인 테이블 통합, 테이블 분할, 중복 테이블 추가, 중복 속성 추가의 특징에 대해 정리하세요.

정규화 원칙에 대한 자세한 내용은 342쪽을 참조하세요.

1 반정규화의 개념

반정규화란 시스템의 성능 향상, 개발 및 운영의 편의성 등을 위해 정규화된 데이터 모델을 통합, 중복, 분리하는 과정으로, 의도적으로 정규화 원칙*을 위배하는 행위이다.

- 반정규화를 수행하면 시스템의 성능이 향상되고 관리 효율성은 증가하지만 데이터의 일관성 및 정합성이 저하될 수 있다.
- 과도한 반정규화는 오히려 성능을 저하시킬 수 있다.
- 반정규화를 위해서는 사전에 데이터의 일관성과 무결성을 우선으로 할지, 데이터베이스의 성능과 단순화를 우선으로 할지를 결정해야 한다.
- 반정규화 방법에는 테이블 통합, 테이블 분할, 중복 테이블 추가, 중복 속성 추가 등이 있다.

2 테이블 통합

테이블 통합은 두 개의 테이블이 조인(Join)되는 경우가 많아 하나의 테이블로 합쳐 사용하는 것이 성능 향상에 도움이 될 경우 수행한다.

- 두 개의 테이블에서 발생하는 프로세스가 동일하게 자주 처리되는 경우, 두 개의 테이블을 이용하여 항상 조회를 수행하는 경우 테이블 통합을 고려한다.

학번	담당교수
201001	홍길동
201002	유관순
201003	윤봉길
201004	홍길동
201005	이순신
201006	유관순

담당교수	과목명
홍길동	정보처리
이순신	정보처리
윤봉길	인공지능
유관순	네트워크

학번	담당교수	과목명
201001	홍길동	정보처리
201002	유관순	네트워크
201003	윤봉길	인공지능
201004	홍길동	정보처리
201005	이순신	정보처리
201006	유관순	네트워크

〈테이블 통합〉

- 테이블 통합의 종류에는 1:1 관계 테이블 통합, 1:N 관계 테이블 통합, 슈퍼타입/서브타입* 테이블 통합이 있다.
- 테이블 통합 시 고려 사항
  - 데이터 검색은 간편하지만 레코드 증가로 인해 처리량이 증가한다.
  - 테이블 통합으로 인해 입력, 수정, 삭제 규칙이 복잡해질 수 있다.
  - Not Null*, Default*, Check* 등의 제약조건(Constraint)을 설계하기 어렵다.

슈퍼타입/서브타입
슈퍼타입은 상위 개체를, 서브타입은 하위 개체를 의미합니다.

- Not Null : 속성 값이 Null이 될 수 없음
- Default : 속성 값이 생략되면 기본값 설정
- Check : 속성 값의 범위나 조건을 설정하여 설정한 값만 허용

3 테이블 분할

테이블 분할은 테이블을 수직 또는 수평으로 분할하는 것이다.

〈수평 분할〉

학번	담당교수	과목명
201001	홍길동	정보처리
201002	유관순	네트워크
201003	윤봉길	인공지능
201004	홍길동	정보처리
201005	이순신	정보처리
201006	유관순	네트워크

학번	담당교수	과목명
201001	홍길동	정보처리
201002	유관순	네트워크
201003	윤봉길	인공지능

학번	담당교수	과목명
201004	홍길동	정보처리
201005	이순신	정보처리
201006	유관순	네트워크

〈수직 분할〉

학번	담당교수	과목명
201001	홍길동	정보처리
201002	유관순	네트워크
201003	윤봉길	인공지능
201004	홍길동	정보처리
201005	이순신	정보처리
201006	유관순	네트워크

학번	담당교수
201001	홍길동
201002	유관순
201003	윤봉길
201004	홍길동
201005	이순신
201006	유관순

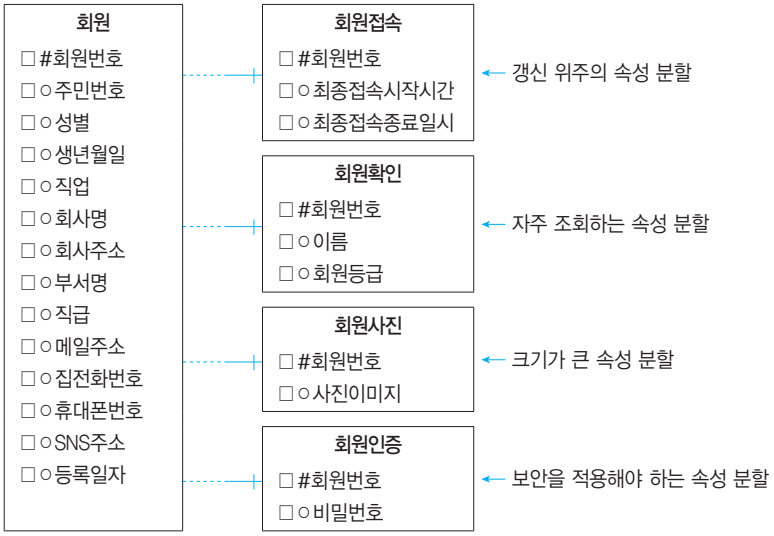
학번	과목명
201001	정보처리
201002	네트워크
201003	인공지능
201004	정보처리
201005	정보처리
201006	네트워크

- 수평 분할(Horizontal Partitioning)
  - 수평 분할은 레코드(Record)를 기준으로 테이블을 분할하는 것이다.
  - 레코드별로 사용 빈도의 차이가 큰 경우 사용 빈도에 따라 테이블을 분할한다.
- 수직 분할(Vertical Partitioning)
  - 수직 분할은 하나의 테이블에 속성이 너무 많을 경우 속성을 기준으로 테이블을 분할하는 것이다.
  - **갱신 위주의 속성 분할** : 데이터 갱신 시 레코드 잠금*으로 인해 다른 작업을 수행할 수 없으므로 갱신이 자주 일어나는 속성들을 수직 분할하여 사용한다.
  - **자주 조회되는 속성 분할** : 테이블에서 자주 조회되는 속성이 극히 일부일 경우 자주 사용되는 속성들을 수직 분할하여 사용한다.
  - **크기가 큰 속성 분할** : 이미지나 2GB 이상 저장될 수 있는 텍스트 형식 등으로 된 속성들을 수직 분할하여 사용한다.

레코드 잠금
레코드 잠금은 데이터의 무결성을 유지하기 위해 어떤 프로세스가 데이터 값을 변경하려고 하면 변경 작업이 완료될 때까지 다른 프로세스가 해당 데이터 값을 변경하지 못하도록 하는 것을 의미합니다.

– 보안을 적용해야 하는 속성 분할 : 테이블 내의 특정 속성에 대해 보안을 적용할 수 없으므로 보안을 적용해야 하는 속성들을 수직 분할하여 사용한다.

예 수직 분할 수행의 예



- 테이블 분할 시 고려 사항
  - 기본키의 유일성 관리가 어려워진다.
  - 데이터 양이 적거나 사용 빈도가 낮은 경우 테이블 분할이 필요한지를 고려해야 한다.
  - 분할된 테이블로 인해 수행 속도가 느려질 수 있다.
  - 데이터 검색에 중점을 두어 테이블 분할 여부를 결정해야 한다.

4 중복 테이블 추가

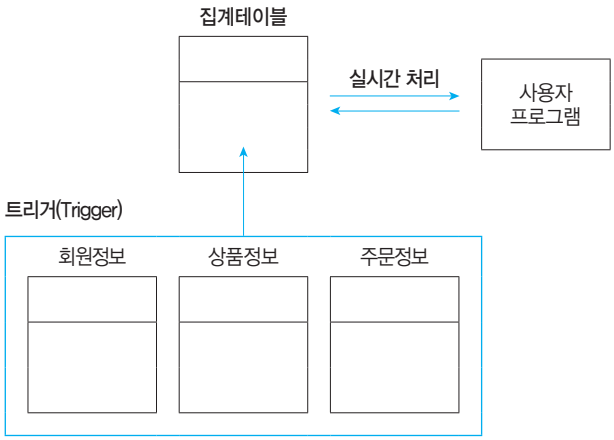
여러 테이블에서 데이터를 추출해서 사용해야 하거나 다른 서버에 저장된 테이블을 이용해야 하는 경우 중복 테이블을 추가하여 작업의 효율성을 향상시킬 수 있다.

- 중복 테이블을 추가하는 경우
  - 정규화로 인해 수행 속도가 느려지는 경우
  - 많은 범위의 데이터를 자주 처리해야 하는 경우
  - 특정 범위의 데이터만 자주 처리해야 하는 경우
  - 처리 범위를 줄이지 않고는 수행 속도를 개선할 수 없는 경우
- 중복 테이블을 추가하는 방법은 다음과 같다.
  - 집계 테이블의 추가 : 집계 데이터를 위한 테이블을 생성하고, 각 원본 테이블에 트리거(Trigger)*를 설정하여 사용하는 것으로, 트리거의 오버헤드(Overhead)에 유의해야 한다.
  - 진행 테이블의 추가 : 이력 관리* 등의 목적으로 추가하는 테이블로, 적절한 데이터 양의 유지와 활용도를 높이기 위해 기본키를 적절히 설정한다.

트리거(Trigger)
트리거는 데이터베이스 시스템에서 데이터의 입력, 갱신, 삭제 등의 이벤트(Event)가 발생할 때마다 자동적으로 수행되는 절차형 SQL입니다.
※ 이벤트(Event) : 시스템에 어떤 일이 발생한 것을 말하며, 트리거에서 이벤트는 데이터의 입력, 갱신, 삭제와 같이 데이터를 조작하는 작업이 발생했음을 의미함

이력 관리
이력 관리는 속성 값의 변화를 관리하기 위해 테이블에서 특정 속성 값이 변경될 때마다 변경되기 전의 속성 값을 저장하는 것을 말합니다.

– 특정 부분만을 포함하는 테이블의 추가 : 데이터가 많은 테이블의 특정 부분만을 사용하는 경우 해당 부분만으로 새로운 테이블을 생성한다.

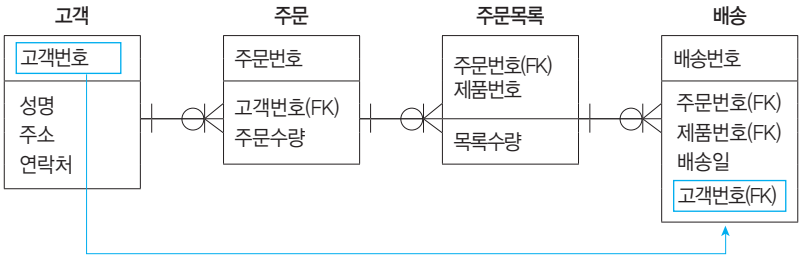


〈집계 테이블 추가〉

5 중복 속성 추가

중복 속성 추가는 조인해서 데이터를 처리할 때 데이터를 조회하는 경로를 단축하기 위해 자주 사용하는 속성을 하나 더 추가하는 것이다.

- 중복 속성을 추가하면 데이터의 무결성 확보가 어렵고, 디스크 공간이 추가로 필요하다.



〈중복 속성 추가〉

- 중복 속성을 추가하는 경우
  - 조인이 자주 발생하는 속성인 경우
  - 접근 경로가 복잡한 속성인 경우
  - 액세스의 조건으로 자주 사용되는 속성인 경우
  - 기본키의 형태가 적절하지 않거나 여러 개의 속성으로 구성된 경우
- 중복 속성 추가 시 고려 사항
  - 테이블 중복과 속성의 중복을 고려한다.
  - 데이터 일관성 및 무결성에 유의해야 한다.
  - SQL 그룹 함수를 이용하여 처리할 수 있어야 한다.
  - 저장 공간의 지나친 낭비를 고려한다.

전문가의 조언

〈회원정보〉, 〈상품정보〉, 〈주문정보〉 테이블의 데이터를 집계하는 집계 테이블을 추가하여 시스템 사용이 적은 시간에 배치 작업에 의해 원하는 데이터를 생성하여 사용합니다.

출제예상
1. 다음 중 반정규화에 대한 설명으로 가장 옳지 않은 것은?

① 반정규화는 의도적으로 정규화 원칙을 위배하는 행위이다.
② 반정규화를 수행하면 시스템의 성능이 저하된다.
③ 반정규화로 인해 데이터의 일관성 및 정합성이 저하될 수 있다.
④ 반정규화 방법에는 테이블 통합, 테이블 분할, 중복 테이블 및 중복 속성 추가 등이 있다.

반정규화를 수행하는 이유는 시스템 성능 향상과 관리 효율성 증가입니다.

출제예상
2. 다음 중 반정규화 방법 중 테이블 통합의 종류가 아닌 것은?

① 1:1 관계 테이블 통합
② 1:N 관계 테이블 통합
③ 슈퍼/서브타입 테이블 통합
④ N:M 관계 테이블 통합

테이블 통합하면 '1:1, 1:N, 슈퍼/서브타입'이란 것을 기억해 두세요!

출제예상
3. 다음 중 반정규화 시 중복 테이블을 추가해야 하는 경우가 아닌 것은?

① 많은 양의 범위를 자주 처리해야 하는 경우
② 특정 범위의 데이터만 자주 처리해야 하는 경우
③ 처리 범위를 늘리지 않고는 수행 속도를 개선할 수 없는 경우
④ 정규화로 인해 수행 속도가 느려지는 경우

수행 속도를 개선하려면 처리 범위를 줄여야 할까요? 늘려야 할까요?

출제예상
4. 다음의 내용은 반정규화의 어떤 유형에 대한 고려사항인가?

- 데이터 검색은 간편하지만 Row 수가 증가하여 처리량이 증가할 수 있음을 고려해야 한다.
- 입력, 수정, 삭제 규칙이 복잡해질 수 있음을 고려해야 한다.
- Not Null, Default, Check 등의 Constraint을 설계하기 어려운 점을 고려해야 한다.

① 테이블 통합 ② 테이블 분할
③ 중복 테이블 추가 ④ 중복 속성 추가

통합, 분할, 추가 중 어떤 경우에 열(Row)의 수가 증가할까요?

출제예상
5. 다음 중 반정규화 시 중복 속성을 추가해야 하는 경우가 아닌 것은?

① 기본키가 하나의 속성으로 구성된 경우
② 조인이 자주 발생하는 속성인 경우
③ 접근 경로가 복잡한 속성인 경우
④ 액세스의 조건으로 자주 사용되는 속성인 경우

기본키의 형태가 적절하지 않거나 여러 개의 속성으로 구성된 경우 중복 속성을 추가합니다.

출제예상
6. 다음 중 시스템의 성능 향상, 개발 및 운영의 편의성 등을 위해 정규화된 데이터 모델을 통합, 중복, 분리하는 과정을 의미하는 용어는?

① 통합정규화 ② 비정규화
③ 반정규화 ④ 강정규화

이것은 정규화에 반(反)하는 정규화를 의미합니다.

출제예상
7. 다음 중 반정규화 시 중복 테이블을 추가하는 방법에 해당하지 않는 것은?

① 진행 테이블의 추가
② 인증 테이블의 추가
③ 집계 테이블의 추가
④ 특정 부분만을 포함하는 테이블의 추가

중복 테이블을 추가하는 방법은 '진행, 집계, 특정 부분'이란 것! 잊지마세요.

출제예상
8. 다음 반정규화 방법 중 중복 속성 추가 시 고려할 사항으로 가장 거리가 먼 것은?

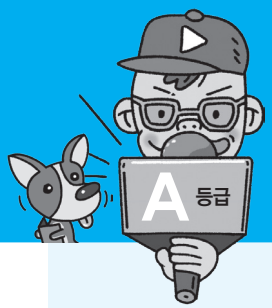
① 테이블 중복과 속성의 중복을 고려한다.
② 데이터 일관성 및 무결성에 유의해야 한다.
③ SQL Group Function을 이용하여 해결 가능한지 검토한다.
④ 기본키의 유일성 관리가 어려워짐을 고려한다.

기본키의 유일성 관리가 어려워짐을 고려하는 것은 테이블 분할 시 고려사항입니다.

▶ 정답 : 1. ② 2. ④ 3. ③ 4. ① 5. ① 6. ③ 7. ② 8. ④

SECTION 084 LIVE

시스템 카탈로그



1 시스템 카탈로그(System Catalog)의 의미

시스템 카탈로그는 시스템 그 자체에 관련이 있는 다양한 객체에 관한 정보를 포함하는 시스템 데이터베이스이다.

- 시스템 카탈로그 내의 각 테이블은 사용자를 포함하여 DBMS에서 지원하는 모든 데이터 객체에 대한 정의나 명세에 관한 정보를 유지 관리하는 시스템 테이블이다.
- 카탈로그들이 생성되면 데이터 사전(Data Dictionary)에 저장되기 때문에 좁은 의미로는 카탈로그를 데이터 사전이라고도 한다.

2 시스템 카탈로그 저장 정보

시스템 카탈로그에 저장된 정보를 메타 데이터(Meta-Data)라고 한다.

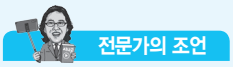
- **메타 데이터의 유형**
  - 데이터베이스 객체 정보 : 테이블(Table), 인덱스(Index), 뷰(View) 등의 구조 및 통계 정보
  - 사용자 정보 : 아이디, 패스워드, 접근 권한 등
  - 테이블의 무결성 제약 조건 정보 : 기본키, 외래키, NULL 값 허용 여부 등
  - 함수, 프로시저, 트리거 등에 대한 정보

3 카탈로그의 특징

- 카탈로그 자체도 시스템 테이블로 구성되어 있어 일반 이용자도 SQL을 이용하여 내용을 검색해 볼 수 있다.
- INSERT, DELETE, UPDATE문으로 카탈로그를 갱신하는 것은 허용되지 않는다.
- 데이터베이스 시스템에 따라 상이한 구조를 갖는다.
- 카탈로그는 DBMS가 스스로 생성하고 유지한다.
- **카탈로그의 갱신** : 사용자가 SQL문을 실행시켜 기본 테이블, 뷰, 인덱스 등에 변화를 주면 시스템이 자동으로 갱신한다.
- **분산 시스템에서의 카탈로그** : 보통의 릴레이션, 인덱스, 사용자 등의 정보를 포함할 뿐 아니라 위치 투명성 및 중복 투명성을 제공하기 위해 필요한 모든 제어 정보를 가져야 한다.

전문가의 조언

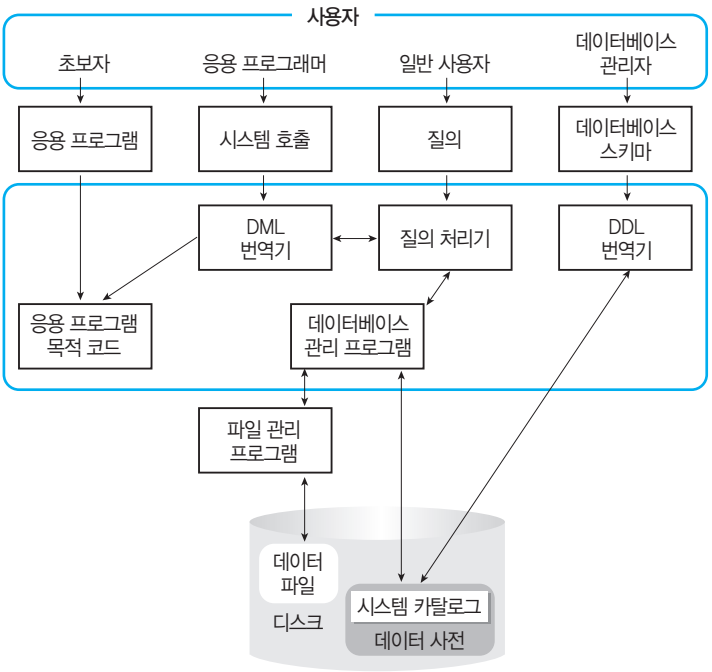
시스템 카탈로그의 의미와 특징을 전부 숙지하세요.



데이터 디렉터리의 개념을 명확히 하고 나머지는 어떤 역할을 하는지 다른 것들과 구분할 수 있을 정도로만 학습하세요.

4 카탈로그/데이터 사전을 참조하기 위한 DBMS 내의 모듈 시스템

- **데이터 정의어 번역기(DDL Compiler)** : DDL을 메타 데이터를 갖는 테이블(카탈로그)로 변환하여 데이터 사전에 저장시킨다.
- **데이터 조작어 번역기(DML Compiler)** : 응용 프로그램에 삽입된 DML문을 주 언어로 표현한 프로시저 호출로 변환하여 질의 처리기와 상호 통신한다.
- Data Directory
  - 데이터 사전에 수록된 데이터를 실제로 접근하는 데 필요한 정보를 관리 유지하는 시스템이다.
  - 시스템 카탈로그는 사용자와 시스템 모두 접근할 수 있지만 데이터 디렉터리는 시스템만 접근할 수 있다.
- **질의 최적화기** : 사용자의 요구를 효율적인 형태로 변환하고 질의를 처리하는 좋은 전략을 모색한다.
- **트랜잭션 처리기** : 복수 사용자 환경에서 평행으로 동시에 일어나는 트랜잭션 문제를 해결하여, 각각의 사용자가 데이터베이스 자원을 배타적으로 이용할 수 있도록 한다.



기출문제 따라잡기

Section 084

이전기출

1. 시스템 자신이 필요로 하는 여러 가지 객체에 관한 정보를 포함하고 있는 시스템 데이터베이스로서, 포함하고 있는 객체로는 테이블, 데이터베이스, 뷰, 접근 권한 등이 있는 것은?

① 인덱스(Index)
② 카탈로그(Catalog)
③ QBE(Query By Example)
④ SQL(Structure Query Language)

시스템 카탈로그에 대한 문제가 여러 가지 형태로 많이 출제됩니다. 의미를 명확히 숙지해야 합니다.

이전기출

2. 데이터베이스에 포함되는 모든 데이터 객체들에 대한 정의나 명세에 관한 정보를 유지 관리하는 시스템을 무엇이라 하는가?

① 데이터 디렉터리 ② 데이터 사전
③ 저장 시스템 ④ 메타 시스템

데이터 사전이 시스템 카탈로그를 의미한다는 것은 알고 있죠?

이전기출

3. 시스템 카탈로그에 대한 설명으로 옳지 않은 것은?

① 시스템 자신이 필요로 하는 여러 가지 개체에 대한 정보를 포함한 시스템 데이터베이스이다.
② 개체들로서는 기본 테이블, 뷰, 인덱스, 데이터베이스, 패키지, 접근 권한 등이 있다.
③ 카탈로그 자체도 시스템 테이블로 구성되어 있어 일반 이용자도 SQL을 이용하여 내용을 검색해 볼 수 있다.
④ 모든 데이터베이스 시스템에서 요구하는 정보는 동일하므로 데이터베이스 시스템의 종류에 관계없이 동일한 구조로 필요한 정보를 제공한다.

시스템 카탈로그는 기본 테이블, 뷰, 인덱스, 패키지, 접근 권한 등 시스템 그 자체에 관련이 있는 정보가 데이터 정의어(DDL)의 결과로 생성되는데, DDL은 사용하는 DBMS의 종류에 따라 다르게 생성됩니다.

이전기출

4. 시스템 카탈로그에 대한 설명으로 옳지 않은 것은?

① 데이터베이스의 무결성 유지를 위하여 사용자가 직접 시스템 카탈로그를 갱신해야 한다.
② 일반 질의어를 이용해 그 내용을 검색할 수 있다.
③ DBMS가 스스로 생성하고, 유지하는 데이터베이스 내의 특별한 테이블의 집합체이다.
④ 데이터베이스 스키마에 대한 정보를 제공한다.

사용자가 시스템 카탈로그를 직접 갱신할 수는 없습니다. 사용자가 SQL문을 실행시켜 기본 테이블, 뷰, 인덱스 등에 변화를 주면 시스템이 자동으로 갱신합니다.

이전기출

5. 시스템 카탈로그에 대한 설명으로 옳지 않은 것은?

① 시스템 자체에 관련 있는 다양한 객체에 관한 정보를 포함하는 시스템 데이터베이스이다.
② 데이터 사전이라고도 한다.
③ 기본 테이블, 뷰, 인덱스, 패키지, 접근 권한 등의 정보를 저장한다.
④ 시스템을 위한 정보를 포함하는 시스템 데이터베이스이므로 일반 사용자는 SQL을 이용하여 내용을 검색해 볼 수 없다.

시스템 카탈로그 자체도 시스템 테이블로 구성되어 있어 일반 사용자도 SQL을 이용하여 내용을 검색해 볼 수 있습니다. 단 수정은 불가능하다는 것! 잊지 않았죠?

이전기출

6. 다음 영문의 ( ) 안 내용으로 가장 적절한 것은?

> A(n) ( ) is a file that contains meta data that is, data about data. This file is consulted before actual data are read or modified in the database system.

① VIEW ② Index
③ ISAM File ④ Data Dictionary

문제의 지문에서 핵심 키워드는 Meta Data입니다. Meta Data가 저장되는 곳이 어디일까요?

이전기출

7. 데이터 디렉터리(Data Directory)에 관한 설명으로 옳은 것은?

① 시스템 카탈로그(System Catalog)라고도 한다.
② 여러 스키마와 이들 간의 사상(Mapping)에 관한 정보를 관리한다.
③ 시스템만 접근할 수 있으며, 데이터베이스에 수록된 데이터를 실제로 접근하는 데 필요한 정보를 관리한다.
④ 데이터베이스에 포함된 모든 데이터 객체들에 대한 정의나 명세에 관한 정보를 유지 관리한다.

③번에서 데이터베이스를 시스템 카탈로그나 시스템 데이터베이스로 수정해야 좀더 명확해집니다.
- 데이터 디렉터리는 시스템만 접근할 수 있으며, 시스템 카탈로그라는 시스템 데이터베이스에 수록된 데이터를 실제로 접근하는 데 필요한 정보를 관리합니다.
- 데이터 디렉터리는 다른 말로 사전 관리기라고 합니다.
- ①, ②, ④번은 모두 데이터 사전(시스템 카탈로그)에 대한 설명입니다. 데이터 사전과 데이터 디렉터리를 혼동하지 마세요.

▶ 정답 : 1. ② 2. ② 3. ④ 4. ① 5. ④ 6. ④ 7. ③

예·상·문·제·은·행

1. 데이터베이스 설계에 대한 설명으로 옳지 않은 것은?

① 요구 조건 분석 단계는 사용자의 요구 조건을 수집하고 분석하여 사용자가 의도하는 데이터베이스의 용도를 파악해야 한다.

② 개념적 설계 단계에서는 트랜잭션 인터페이스 설계, 스키마의 평가 및 정제 등의 작업을 수행한다.

③ 논리적 설계 단계에서는 개념적 설계 단계에서 만들어진 정보 구조로부터 특정 목표 DBMS가 처리할 수 있는 스키마를 생성한다.

④ 물리적 설계 단계에서는 저장 구조와 접근 경로 등을 결정한다.

2. 데이터베이스 설계 시 논리적 설계 단계에 대한 설명으로 옳지 않은 것은?

① 사용자의 요구에 대한 트랜잭션을 모델링한다.

② 트랜잭션 인터페이스를 설계한다.

③ 관계형 데이터베이스에서는 테이블을 설계하는 단계이다.

④ DBMS에 맞는 논리적 스키마를 설계한다.

3. 데이터베이스 설계 단계 중 물리적 설계에 해당하는 것은?

① 데이터 모형화와 사용자 뷰들을 통합한다.

② 사용자들의 요구사항을 확인하고, 메타 데이터를 수집, 기록한다.

③ 파일 조직 방법과 저장 방법, 그리고 파일 접근 방법 등을 선정한다.

④ 사용자들의 요구사항을 입력으로 하여 응용 프로그램의 골격인 스키마를 작성한다.

4. In the database design process, its result is a database schema in the implementation data model of DBMS. What is called this step?

① conceptual database design

② physical database design

③ transaction implementation design

④ logical database design

5. Which is the design step of database correctly?

① Requirement Formulation → Conceptual Schema → Physical Schema → Logical Schema

② Logical Schema → Requirement Formulation → Conceptual Schema → Physical Schema

③ Requirement Formulation → Conceptual Schema → Logical Schema → Physical Schema

④ Logical Schema → Requirement Formulation → Physical Schema → Conceptual Schema

6. 데이터베이스 설계에 있어 개념 스키마 모델링과 트랜잭션 모델링을 병행적으로 수행하는 단계는?

① 요구분석 설계

② 개념적 설계

③ 논리적 설계

④ 물리적 설계

7. 현실 세계에 존재하는 개체를 인간이 이해할 수 있는 정보 구조로 표현하는 과정을 무엇이라 하는가?

① 데이터 모델링(Data Modeling)

② 정보 모델링(Information Modeling)

③ 데이터 구조화(Data Structuring)

④ 정보 구조화(Information Structuring)

8. 데이터베이스의 구성 요소 중 개체(Entity)에 대한 설명으로 적합하지 않은 것은?

① 속성들이 가질 수 있는 모든 값들의 집합이다.

② 데이터베이스에 표현하려고 하는 현실 세계의 대상체이다.

③ 유형, 무형의 정보로서 서로 연관된 몇 개의 속성으로 구성된다.

④ 파일의 레코드에 대응하는 것으로 어떤 정보를 제공하는 역할을 수행한다.

9. What is the entity type definition correctly?

① a set of attributes that have the same entities

② a set of entities that have the same domains

③ a set of attributes that have the same domains

④ a set of entities that have the same attributes

10. 다음 설명이 의미하는 A와 B의 관계는?

An entity in an entity set A is associated with any number of entities in an entity set B, and an entity in B is associated with any number of entities in A.

① one to one ② one to many

③ many to one ④ many to many

해설은 358쪽에 있습니다.

11. 개체에 대한 설명 중 옳은 것은?

① 컴퓨터가 취급하는 파일의 레코드에 대응된다.
② 하나의 개체는 하나의 속성만을 가진다.
③ 한 속성이 취할 수 있는 모든 값을 의미한다.
④ 개체는 단독으로는 존재하지 못한다.

12. 다음 릴레이션의 Degree와 Cardinality는?

13001	홍길동	3학년	전기
13002	이순신	4학년	기계
13003	강감찬	2학년	컴퓨터

① Degree : 4, Cardinality : 3
② Degree : 3, Cardinality : 4
③ Degree : 3, Cardinality : 12
④ Degree : 12, Cardinality : 3

13. 개체-관계 모델에 대한 설명으로 옳지 않은 것은?

① 오너-멤버(Owner-Member) 관계라고도 한다.
② 개체 타입과 이들 간의 관계 타입을 기본 요소로 이용하여 현실 세계를 개념적으로 표현한다.
③ E-R 다이어그램에서 개체 타입은 사각형으로 나타낸다.
④ E-R 다이어그램에서 속성은 타원으로 나타낸다.

14. 개체-관계 모델의 E-R 다이어그램에서 사용되는 기호와 그 의미의 연결이 옳지 않은 것은?

① 사각형 – 개체 타입
② 삼각형 – 속성
③ 선 – 개체 타입과 속성을 연결
④ 마름모 – 관계 타입

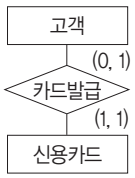
15. Which of the following is not a component of Entity-Relationship diagram?

① Rectangles, which represent entity sets
② Ellipses, which represent database operations
③ Diamond, which represent relationships among entity sets
④ Lines, which link attributes to entity sets and entity sets to relationships

16. 개체-관계(E-R) 모델의 설명으로 옳지 않은 것은?

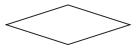
① 데이터를 엔티티, 관계, 속성으로 묘사한다.
② E-R 모델에서 엔티티는 실세계에서 개념적 또는 물리적으로 존재하는 실제 사용을 의미한다.
③ E-R 모델에서 속성은 엔티티를 묘사하는 데 사용될 수 있는 특성을 의미한다.
④ E-R 모델에서 관계는 속성들에 대한 관계 표현이다.

17. 다음은 어느 신용카드 회사에서 고객과 고객이 발급받은 신용카드 간의 관계를 나타내는 E-R 다이어그램이다. 이에 대한 설명 중 부적합한 것은?



① 신용카드를 발급받지 않은 고객은 존재할 수 없다.
② 고객이 없는 신용카드는 발급될 수 없다.
③ 개체 고객과 신용카드 간의 카드발급 관계는 일 대 일 관계이다.
④ 한 사람의 고객에는 반드시 하나의 신용카드만 발급된다.

18. E-R 모델에서 다중값 속성의 표기법은?

① 
②
③
④

19. 다음 영문의 괄호에 적합한 Database System은?

> Database management systems that process data from the perspective of a ( ) structure use a set of two dimensional tables to represent the logical relationships among the records of a user's files.

① Hierarchical Database
② Network Database
③ Relational Database
④ Object-oriented Database

20. 릴레이션의 특징으로 옳지 않은 것은?

① 한 릴레이션에 포함된 튜플 사이에는 순서가 없다.
② 속성의 값은 논리적으로 더 이상 쪼갤 수 없는 원자 값이다.
③ 한 릴레이션에 포함된 튜플들은 모두 상이하다.
④ 한 릴레이션을 구성하는 속성들 사이의 순서는 존재하며, 중요한 의미를 가진다.

▶ 정답 : 1. ② 2. ① 3. ③ 4. ④ 5. ③ 6. ② 7. ② 8. ① 9. ① 10. ④ 11. ① 12. ① 13. ① 14. ② 15. ② 16. ④ 17. ① 18. ③ 19. ③ 20. ④

21. 괄호 안 내용으로 옳게 짝지어진 것은?

> In relational database terminology, a row is called a(an) (㉠), a column is called a(an) (㉡), and the (㉢) is called a relation.

① ㉠ tuple – ㉡ table – ㉢ attribute
② ㉠ table – ㉡ attribute – ㉢ tuple
③ ㉠ tuple – ㉡ attribute – ㉢ table
④ ㉠ attribute – ㉡ tuple – ㉢ table

22. 다음은 관계형 데이터베이스의 키(Key)를 설명하고 있다. 해당되는 키는?

> 한 릴레이션 내의 속성들의 집합으로 구성된 키로서, 릴레이션을 구성하는 모든 튜플에 대한 유일성은 만족시키지만 최소성은 만족시키지 못한다.

① 후보키 ② 대체키
③ 슈퍼키 ④ 외래키

23. 다음 두 릴레이션 간의 관계에서 교수 릴레이션에 존재하는 외래키는? (단, 교수 릴레이션의 기본키는 교수번호이고 학과 릴레이션의 기본키는 학과번호이다)

> 교수(교수번호, 교수이름, 학과번호, 직급)
> 학과(학과번호, 학과이름, 학과장교수번호, 학생수)

① 교수이름 ② 학과번호
③ 학과장교수번호 ④ 학과이름

24. 다음은 관계 대수의 수학적 표현식이다. 해당되는 연산은?

> $R \times S = \{ r \cdot s \mid r \in R \wedge s \in S \}$
> $r = \langle a_1, a_2, \cdots, a_n \rangle$, $s = \langle b_1, b_2, \cdots, b_m \rangle$

① 합집합 ② 교집합
③ 차집합 ④ 카티션 프로덕트

25. 릴레이션 R의 차수가 4이고 카디널리티가 5이며, 릴레이션 S의 차수가 6이고 카디널리티가 7일 때, 두 개의 릴레이션을 카티션 프로덕트한 결과의 새로운 릴레이션의 차수와 카디널리티는 얼마인가?

① 24, 35 ② 24, 12
③ 10, 35 ④ 10, 12

26. 관계대수(Relational Algebra)의 연산자 중에서 두 릴레이션(Relation)의 교차 곱을 수행하기 때문에 두 릴레이션의 공통 튜플 수와 관계가 없는 것은?

① Union
② Intersection
③ Difference
④ Cartesian Product

27. 관계대수에 대한 설명으로 옳지 않은 것은?

① 릴레이션을 처리하기 위한 연산의 집합으로 피연산자가 릴레이션이고 결과도 릴레이션이다.
② 원하는 정보와 그 정보를 어떻게 유도하는가를 기술하는 절차적 특징을 가지고 있다.
③ 일반 집합 연산과 순수 관계 연산이 있다.
④ 수학의 Predicate Calculus에 기반을 두고 있다.

28. 관계 데이터베이스의 정규화에 관한 설명으로 옳지 않은 것은?

① 3NF는 무손실 조인 또는 종속성 보존을 저해하지 않고도 항상 3NF 설계를 얻을 수 있다.
② 3NF는 정규형에서 모든 이행(Transitive) 종속을 제거하지 못한 경우라도 정보의 중복에 대한 문제가 존재하지 않는 장점을 가지고 있다.
③ 모든 BCNF(Boyce–Codd Normal Form)가 종속성을 보존하는 것은 아니다.
④ 모든 BCNF 스키마는 3NF에 속하게 되며, 따라서 BCNF가 3NF보다 한정적 제한이 더 많다.

29. 관계 데이터 모델링 중 BCNF(Boyce–Codd Normal Form)에 대한 옳은 설명으로만 짝지어진 것은?

> ㉠ BCNF에 속하는 릴레이션은 반드시 제3정규형(Third Normal Form)에 속한다.
> ㉡ 제3정규형에 속하지만 BCNF에 속하지 않는 릴레이션이 있다.
> ㉢ 복합 속성을 허용하지 않는다.
> ㉣ 완전 함수적 종속성 개념에 기반을 두었다.

① ㉠, ㉢
② ㉠, ㉡
③ ㉠, ㉡, ㉢
④ ㉠, ㉡, ㉢, ㉣

해설은 359쪽에 있습니다.

30. 정규화에 관한 설명으로 옳지 않은 것은?

① 릴레이션 R의 도메인들의 값이 원자 값만을 가지면 릴레이션 R은 제1정규형에 해당된다.

② 릴레이션 R이 제1정규형을 만족하면서, 키가 아닌 모든 속성이 기본키에 완전 함수 종속이면 릴레이션 R은 제2정규형에 해당된다.

③ 정규형들은 차수가 높아질수록(제1정규형 → 제5정규형) 만족시켜야 할 제약 조건이 감소된다.

④ 릴레이션 R이 제2정규형을 만족하면서, 키가 아닌 모든 속성들이 기본키에 이행적으로 함수 종속되지 않으면 릴레이션 R은 제3정규형에 해당된다.

31. 정규화의 필요성으로 거리가 먼 것은?

① 데이터 구조의 안정성 최대화

② 중복 데이터의 활성화

③ 수정, 삭제 시 이상 현상의 최소화

④ 테이블 불일치 위험의 최소화

32. 시스템 카탈로그에 대한 설명으로 옳지 않은 것은?

① 시스템 카탈로그는 DBMS가 생성하고 유지하는 데이터베이스 내의 특별한 테이블들의 집합체이다.

② 일반 사용자도 시스템 카탈로그의 내용을 검색할 수 있다.

③ 시스템 카탈로그 내의 각 테이블은 DBMS에서 지원하는 개체들에 관한 정보를 포함한다.

④ 시스템 카탈로그에 대한 갱신은 데이터베이스의 무결성 유지를 위하여 사용자가 직접 갱신해야 한다.

33. 데이터 사전(Data Dictionary)에 대한 설명으로 부적합한 것은?

① 여러 가지 스키마와 이들 속에 포함된 사상들에 관한 정보도 컴파일되어 저장된다.

② 데이터베이스를 실제로 접근하는 데 필요한 정보를 유지, 관리하며 시스템만이 접근한다.

③ 사전 자체도 하나의 데이터베이스로 간주되며, 시스템 카탈로그(System Catalog)라고도 한다.

④ 데이터베이스가 취급하는 모든 데이터 객체들에 대한 정의나 명세에 관한 정보를 관리 유지한다.

34. 시스템 카탈로그에 대한 설명으로 틀린 것은?

① 데이터베이스에 포함된 다양한 데이터 객체에 대한 정보들을 유지, 관리하기 위한 시스템 데이터베이스이다.

② 시스템 카탈로그를 데이터 사전이라고도 한다.

③ 시스템 카탈로그에 저장된 정보를 메타 데이터라고도 한다.

④ 시스템 카탈로그는 시스템을 위한 정보를 포함하는 시스템 데이터베이스이므로 일반 사용자는 내용을 검색할 수 없다.

▶ 정답 : 21. ③ 22. ③ 23. ② 24. ④ 25. ③ 26. ④ 27. ④ 28. ② 29. ② 30. ③ 31. ② 32. ④ 33. ② 34. ④

예·상·문·제·은·행·해·설

1. Section 070

트랜잭션 인터페이스 설계, 스키마의 평가 및 정제 등의 작업은 논리적 설계 단계에서 수행한다.

2. Section 070

트랜잭션의 모델링은 개념적 설계 단계에서 수행한다.

3. Section 070

물리적 설계의 개념 및 특징

- 논리적 설계 단계에서 논리적 구조로 표현된 데이터를 디스크 등의 물리적 저장장치에 저장할 수 있는 물리적 구조의 데이터로 변환하는 과정이다.
- 물리적 설계 단계에서는 다양한 데이터베이스 응용에 대해서 처리 성능을 얻기 위해 데이터베이스 파일의 저장 구조 및 액세스 경로를 결정한다.
- 저장 레코드의 형식, 순서, 접근 경로와 같은 정보를 사용하여 데이터가 컴퓨터에 저장되는 방법을 묘사한다.
- 물리적 설계 단계에 꼭 포함시켜야 할 것은 저장 레코드의 양식 설계, 레코드 집중(Record Clustering)의 분석 및 설계, 접근 경로 설계 등이다.
- 물리적 데이터베이스 구조의 기본적인 데이터 단위는 저장 레코드(Stored Record)이다.
- 물리적 데이터베이스 구조는 여러 가지 타입의 저장 레코드 집합이라는 면에서 단순한 파일과 다르다.
- 물리적 데이터베이스 구조는 데이터베이스 시스템의 성능에 중대한 영향을 미친다.

4. Section 070

데이터베이스 설계 과정 중 DBMS의 모델에 맞게 구현된 데이터베이스 스키마가 결과로 산출되는 단계는 논리적 설계입니다.

5. Section 070

데이터베이스 설계 순서

요구 분석 → 개념적 설계 → 논리적 설계 → 물리적 설계 → 구현

6. Section 070

개념적 설계(정보 모델링, 개념화)

- 개념적 설계란 정보의 구조를 얻기 위하여 현실 세계의 무한성과 계속성을 이해하고, 다른 사람과 통신하기 위하여 현실 세계에 대한 인식을 추상적 개념으로 표현하는 과정이다.
- 개념적 설계 단계에서는 개념 스키마 모델링과 트랜잭션 모델링을 병행 수행한다.
- 개념적 설계 단계에서는 요구 분석 단계에서 나온 결과인 요구 조건 명세를 DBMS에 독립적인 E-R 다이어그램으로 작성한다.
- DBMS에 독립적인 개념 스키마를 설계한다.

8. Section 072

속성들이 가질 수 있는 모든 값들의 집합은 도메인(Domain)이라고 한다.

9. Section 072

개체(Entity)는 같은 개체를 갖는 속성들의 집합이다.

10. Section 074

개체 집합 A에 있는 하나의 개체가 개체 집합 B의 여러 개의 개체들과 관계를 맺고, 개체 집합 B에 있는 하나의 개체도 개체 집합 A의 여러 개의 개체들과 관계를 맺고 있는 관계의 종류는 다 대 다, 즉 Many to Many이다.

11. Section 072

② 하나의 개체는 하나 이상의 속성을 가진다.

③ 도메인(Domain)에 대한 설명이다.

④ 개체는 독립적으로 존재하거나 그 자체로서도 구별 가능하다.

12. Section 078

디그리(Degree)는 속성의 수(열의 수)를 말하고, 카디널리티(Cardinality)는 튜플의 수(행의 수)를 말한다.

13. Section 076

오너-멤버(Owner-Member) 관계라고도 불리는 데이터 모델은 논리적 데이터 모델 중 하나인 네트워크(망)형 데이터 모델이다.

14. Section 076

E-R 다이어그램에서 속성은 타원으로 표시한다.

15. Section 076

개체-관계 다이어그램의 요소로 잘못 설명된 것은?

① **사각형** : 개체 관계를 표현한다.

② **타원** : 데이터베이스 연산을 표현한다.

③ 마름모 : 개체 집합의 관계를 표현한다.

④ 선 : 개체와 속성 사이의 연결 그리고 개체 집합과 관계 사이의 연결을 표현한다.

16. Section 076

E-R 모델에서 관계는 속성들에 대한 관계 표현이 아니고 개체 간의 관계 표현이다.

17. Section 076

고객과 카드 발급 간의 대응이 (0, 1)이므로 최소 0, 최대 1개의 카드를 발급 받는다. 즉 고객 중에는 카드를 발급받지 않은 고객이 존재할 수 있다.

18. Section 076

- ◇ : 관계(Relationship) 타입
- □ : 개체(Entity) 타입
- ——— : 개체 타입과 속성 연결

19. Section 077

관계형 데이터베이스(Relational Database) 구조의 측면에서 데이터를 처리하는 데이터베이스 관리 시스템은 사용자의 파일에 있는 레코드들의 논리적 관계를 나타내는 일련의 2차원적 테이블(표)을 사용한다.

20. Section 078

릴레이션에서 각 속성은 유일한 이름을 가져야 하지만 속성의 순서는 의미가 없다.

21. Section 078

관계형 데이터베이스 관련 용어 중 행은 튜플(Tuple)이라고 부르며, 열은 속성(Attribute)이라고 부른다. 그리고 테이블(Table)은 릴레이션(Relation)이라고 부른다.

22. Section 079

- 후보키(Candidate Key) : 릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별하기 위해 사용하는 속성들의 부분집합, 즉 기본키로 사용할 수 있는 속성들을 말함
- 대체키(Alternate Key) : 후보키가 둘 이상일 때 기본키를 제외한 나머지 후보키를 의미함
- 외래키 : 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합

23. Section 079

한 학과에 여러 명의 교수가 있을 수 있지만 한 교수가 여러 학과에 소속될 수는 없다. 그러므로 위의 〈교수〉와 〈학과〉 릴레이션은 〈교수〉가 〈학과〉를 참조하는 N : 1의 관계이다. 〈학과〉 릴레이션의 '학과번호'는 기본키이고, 〈학과〉를 참조하는 〈교수〉 릴레이션의 '학과번호'는 외래키가 된다.

25. Section 081

카티션 프로덕트(Cartesian Product), 즉 교차곱은 두 릴레이션의 차수(Degree, 속성의 수)는 더하고, 카디널리티(튜플의 수)는 곱하면 된다. 즉 차수는 4 + 6 = 10이고, 카디널리티는 5 × 7 = 356이다.

26. Section 081

합집합(Union)은 두 릴레이션을 합친 후 공통적인 것 중 하나는 제거하고, 교집합은 두 릴레이션에서 공통적인 것만 구하고, 차집합은 두 릴레이션에서 공통적인 것을 제외한 나머지를 구하는 것이므로 모두 공통 튜플 수와 관계가 있다. 하지만 교차곱(Cartesian Product)은 두 릴레이션의 모든 카디널리티를 곱하는 연산이므로 두 릴레이션의 공통 튜플 수와 관계가 없다.

27. Section 081

수학의 Predicate Calculus에 기반을 두고 관계 데이터베이스를 위해 제안된 것은 관계해석이다.

28. Section 082

3NF는 정규형에서 모든 이행(Transitive) 종속을 제거하지 못한 경우 정보의 중복에 대한 문제가 존재한다.

29. Section 082

BCNF는 복합 속성을 허용하며, 릴레이션의 모든 결정자가 후보키라는 개념에 기본을 두고 있다.

30. Section 082

정규형들은 차수가 높아질수록 만족시켜야 할 제약 조건이 증가한다.

31. Section 082

정규화의 필요성은 중복 데이터의 활성화가 아니라 중복 데이터의 최소화이다.

32. Section 084

시스템 카탈로그는 데이터베이스에 변화가 있을 때마다 DBMS가 스스로 생성하고 유지한다. 사용자는 일반 질의어를 이용해 시스템 카탈로그의 내용을 검색할 수는 있지만 직접 갱신할 수는 없다.

33. Section 084

데이터 사전은 테이블(시스템 테이블)로 구성되어 있어 일반 사용자도 SQL을 이용하여 내용을 검색해 볼 수 있다. 하지만 데이터 사전의 내용을 변경할 수는 없다.

34. Section 084

카탈로그 자체도 시스템 테이블로 구성되어 있어 일반 사용자도 SQL을 이용하여 내용을 검색해 볼 수 있다.

2장

물리 데이터베이스 설계

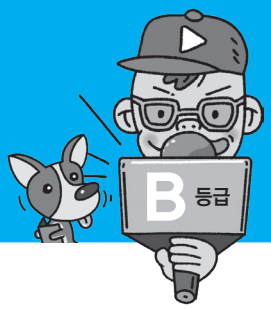
085 사전 조사 분석 B 등급
086 데이터베이스 저장 공간 설계 B 등급
087 트랜잭션 분석 / CRUD 분석 B 등급
088 인덱스 설계 A 등급
089 뷰(View) 설계 A 등급
090 클러스터 설계 A 등급
091 파티션 설계 A 등급
092 데이터베이스 용량 설계 C 등급
093 분산 데이터베이스 설계 A 등급
094 데이터베이스 이중화 / 서버 클러스터링 B 등급
095 데이터베이스 보안 / 암호화 B 등급
096 데이터베이스 보안 – 접근통제 A 등급
097 데이터베이스 백업 C 등급
098 스토리지 B 등급
099 논리 데이터 모델의 물리 데이터 모델 변환 A 등급
100 물리 데이터 모델 품질 검토 C 등급

이 장에서 꼭 알아야 할 키워드 Best 10

1. 물리 데이터베이스 설계 **2.** 트랜잭션 **3.** 인덱스 **4.** 뷰 **5.** 클러스터 **6.** 파티션 **7.** 분산 데이터베이스 **8.** 접근통제 **9.** 스토리지 **10.** 이중화

SECTION 085 ● LIVE

사전 조사 분석



전문가의 조언

물리적 데이터베이스 설계의 의미를 파악하고, 이 단계를 수행하기 전에 조사 및 분석할 내용에는 어떤 것들이 있는지 알아두세요.

1 물리 데이터베이스 설계

물리 데이터베이스 설계는 논리적 구조로 표현된 논리적 데이터베이스를 디스크 등의 물리적 저장장치에 저장할 수 있는 물리적 구조의 데이터로 변환하는 과정이다.

- 물리적 데이터베이스 구조의 기본적인 데이터 단위는 저장 레코드(Stored Record)이다.
- 물리적 설계 단계에 꼭 포함되어야 할 것은 저장 레코드의 양식 설계, 레코드 집중(Record Clustering)의 분석 및 설계, 접근 경로 설계 등이다.
- 물리적 데이터베이스 구조는 여러 가지 타입의 저장 레코드 집합이라는 면에서 단순한 파일과 다르다.
- 물리적 데이터베이스 구조는 데이터베이스 시스템의 성능에 중대한 영향을 미친다.
- 물리적 설계 시 고려 사항
  - 인덱스 구조
  - 레코드 크기
  - 파일에 존재하는 레코드 개수
  - 파일에 대한 트랜잭션*의 갱신과 참조 성향
  - 성능 향상을 위한 개념 스키마*의 변경 여부 검토
  - 빈번한 질의와 트랜잭션들의 수행속도를 높이기 위한 고려
  - 시스템 운용 시 파일 크기의 변화 가능성
- 물리적 설계 전에 기존 시스템을 분석하여 데이터 명명 규칙, 시스템 자원, 데이터베이스 관리 요소 등을 파악해야 한다.

트랜잭션(Transaction)
트랜잭션은 데이터베이스의 상태를 변환시키는 하나의 논리적 기능을 수행하기 위한 작업의 단위 또는 한꺼번에 모두 수행되어야 할 일련의 연산들을 의미합니다.

개념 스키마(Conceptual Schema)
개념 스키마는 데이터베이스의 전체적인 논리적 구조로서, 모든 응용 프로그램이나 사용자들이 필요로 하는 데이터를 종합한 조직 전체의 데이터베이스로 하나만 존재합니다.

잠깐만요

물리적 설계 옵션

물리적 설계 옵션이란 특정 DBMS에서 제공되는 것으로, 데이터베이스 파일에 대한 저장 구조와 접근 경로에 대한 다양한 옵션을 말합니다.

- **반응시간(Response Time)** : 트랜잭션 수행을 요구한 시점부터 처리 결과를 얻을때까지의 경과 시간
- **공간 활용도(Space Utilization)** : 데이터베이스 파일과 액세스 경로 구조에 의해 사용되는 저장 공간의 양
- **트랜잭션 처리량(Transaction Throughput)** : 단위시간 동안 데이터베이스 시스템에 의해 처리될 수 있는 트랜잭션의 평균 개수

2 데이터 명명 규칙 파악

데이터 명명 규칙은 물리 데이터 모델에 적용해야 하는 규칙으로, 조직마다 다를 수 있으므로 물리 데이터 모델의 설계 전에 파악해야 한다.

예 데이터 명명 규칙 예시

업무코드	• 업무코드 3자리, 세부업무코드 3자리로 구성된다. • 업무코드와 세부업무코드를 기준으로 거래코드, 프로그램 등이 명명되므로 표준을 준수해야 한다.
리소스 그룹	업무구분 3자리+세부업무구분 2자리로 명명한다.
거래코드	• 거래코드는 입력되는 데이터의 유형에 따라 부여된다. • 거래코드 명명규칙 : 서비스명과 동일 • 거래처코드와 서비스명은 1:1 관계로 명명한다.

- 데이터 명명 규칙은 데이터 표준화* 및 논리 데이터베이스 설계의 결과물 등을 통해 파악한다.
- 물리 데이터베이스 설계와 논리 데이터베이스 설계에 적용되는 명명 규칙은 서로 일관성을 유지해야 한다.
- 데이터 명명 규칙은 논리적 데이터 요소를 물리적 데이터 요소로 전환할 때 동일 명칭 부여의 근거로 사용된다.
- 데이터 명명 규칙을 통해 중복 구축 등을 방지할 수 있다.
- 명명 규칙을 파악하려면 도메인과 데이터 사전에 대한 지식이 필요하다.
- **도메인** : 객체(Entity)에 포함된 속성들의 데이터 타입, 크기 등을 표준화 규칙에 따라 일관성 있게 정의한 것을 의미한다.

예 도메인 예시

도메인 분류	도메인명	데이터 타입
코드	사원코드	VARCHAR2(10)
	매장코드	VARCHAR2(10)
	상품코드	VARCHAR2(10)
이름	사원이름	VARCHAR2(20)
	매장이름	VARCHAR2(20)
	상품이름	VARCHAR2(20)
수	사원수	NUMBER(10)
	상품수	NUMBER(10)

- **데이터 사전(데이터 용어 사전)**
  - 전체 프로젝트 과정에서 일관성 있는 데이터 이름과 인터페이스를 제공하기 위해 데이터 속성의 논리명(Logical Name), 물리명(Physical Name), 용어 정의(Definition)를 기술해 놓은 것이다.
  - 데이터 사전은 프로젝트에서 사용하는 명칭 부여의 근거로 사용된다.

3 시스템 자원 파악

시스템 자원은 데이터베이스 설치에 영향을 미칠 수 있는 물리적인 요소들로, 사전에 미리 파악해야 한다.

데이터 표준화
데이터 표준화는 데이터 정보 요소에 대한 명칭, 정의, 형식, 규칙에 대한 원칙을 수립하여 전체 시스템에 적용하는 것을 의미합니다.

DBMS(DataBase Management System)
DBMS는 사용자와 데이터베이스 사이에서 사용자의 요구에 따라 정보를 생성해 주고, 데이터베이스를 관리해 주는 소프트웨어입니다.

파라미터(Parameter, 매개 변수)
파라미터는 변수와 같이 값을 일시적으로 저장하는 것으로, DBMS에서는 사용자가 원하는 값을 프로그램에 전달하기 위해 사용됩니다.

옵티마이저(Optimizer)
SQL문을 수행할 가장 효율적이고 빠른 방법을 선택해주는 DBMS의 핵심 엔진입니다.

- 시스템 자원은 하드웨어 자원, 운영체제 및 DBMS* 버전, DBMS 파라미터* 정보 등으로 구분한다.
- **하드웨어 자원**

중앙처리장치	중앙처리장치(CPU)의 성능과 집중적인 부하 발생시간 등을 파악한다.
메모리	시스템의 전체 메모리 규모, 사용 중인 메모리 영역, 사용 가능한 메모리 영역 등 확보된 자원이나 실질적인 시스템 활용 정도 등을 파악한다.
디스크	전체 디스크의 크기, 확보된 디스크 자원, 디스크 분할 형태, 현재 디스크 활용률, 사용 가능한 디스크 공간 등을 파악한다.
I/O Controller	현행 시스템의 입 · 출력 컨트롤러의 성능, 운용의 적절성 등을 파악한다.
네트워크	네트워크의 처리량, 처리 속도, 집중적인 부하 발생 시간, 동시 접속 가능 정도 등을 파악한다.

- **운영체제 및 DBMS 버전** : 운영체제와 DBMS 버전은 데이터베이스 운영에 영향을 미칠 수 있으므로 관련 요소 등을 파악하고 적절하게 관리해야 한다.
- **DBMS 파라미터(Parameter) 정보**
  - DBMS 파라미터는 데이터베이스 관리 시스템별로 차이가 많고 관리 방법도 제각각이므로 시스템별 DBMS 파라미터의 종류 및 관리 대상 등을 파악한다.
  - DBMS의 저장 공간, 메모리 등에 대한 파라미터, 쿼리에서 활용하는 옵티마이저(Optimizer)*의 사용 방법 등을 파악한다.

4 데이터베이스 관리 요소 파악

데이터베이스 관리 요소는 데이터베이스 운영과 관련된 관리 요소로, 데이터베이스 시스템의 환경에 따라 달라질 수 있으므로 미리 파악해야 한다.

- 데이터베이스 관리 요소를 파악한 후 이를 기반으로 데이터베이스 시스템 조사 분석서를 작성한다.

예 데이터베이스 시스템 조사 분석서 예시

구분	환경 변수	중앙(Center)	지역(Local)
운영체제 환경	IP Address	192.254.250.20	192.254.250.20
	CPU	4G	4G
	DISK SPACE	50G	50G
데이터베이스	SGA	40M	40M
	MAX DATA FILES	50	50
	DB BLOCK SIZE	16K	16K
	DB FILES	40	40
	LOG BUFFER	45215	45215
	PROCESSES	40	40
	⋮	⋮	⋮

- 시스템 조사 분석서를 기반으로 데이터베이스 구조, 이중화 구성, 분산 데이터베이스, 접근제어/접근통제, DB암호화 등의 범위와 특성을 파악한다.

데이터베이스 구조	데이터베이스의 구조에 따라 문제 발생 시 대응 방법이 다르므로 서버와 데이터베이스의 구조 등을 파악한다.
이중화 구성	문제 발생에 대비하여 동일한 데이터베이스를 복제하여 관리하는 이중화 구성을 파악한다.
분산 데이터베이스	분산 데이터베이스는 물리적 재해 및 파손 등 데이터 유실을 최소화할 수 있고, 장애로 인한 데이터 유실 복구에 효과적이므로 데이터베이스의 분산 구조를 파악한다.
접근 제어/접근 통제	데이터베이스는 접근 가능한 사용자의 권한 남용으로 인한 정보 유출 및 변조가 가장 빈번하게 발생하므로 데이터베이스의 접근 제어 방법 등을 파악한다.
DB암호화	DB암호화는 데이터 암호화, 암호 키에 대한 인증 등을 통해 데이터 유출 시 데이터의 복호화를 어렵게 하므로 DB암호화의 특성을 파악한다.

전문가의 조언

이중화, 분산 데이터베이스, 접근통제, 암호화 등은 앞으로 자세히 공부합니다. 여기서는 데이터베이스 시스템 환경을 조사할 때는 이것들의 특성을 파악한다는 것 정도만 알아두세요.

따라잡기

기출문제 따라잡기

Section 085

출제예상

1. 데이터베이스 설계 시 물리적 설계 단계에서 수행하는 사항이 아닌 것은?

① 저장 레코드 양식 설계
② 레코드 집중의 분석 및 설계
③ 접근 경로 설계
④ 목표 DBMS에 맞는 스키마 설계

물리적 설계 단계에 꼭 포함되어야 할 세 가지 설계! 꼭 기억하세요. 목표 DBMS에 맞는 스키마 설계는 논리적 설계 단계에서 수행할 사항입니다.

출제예상

2. 물리적 데이터베이스 설계 수행 시 결정 사항으로 거리가 먼 것은?

① 어떤 인덱스를 만들 것인지에 대한 고려
② 성능 향상을 위한 개념 스키마의 변경 여부 검토
③ 빈번한 질의와 트랜잭션들의 수행 속도를 높이기 위한 고려
④ 개념 스키마와 외부 스키마 설계

개념 스키마는 개념적 설계 단계에서 수행하고, 외부 스키마는 데이터베이스를 작성한 후 운영중에 작성합니다.

출제예상

3. 데이터 명명 규칙에 대한 설명으로 틀린 것은?

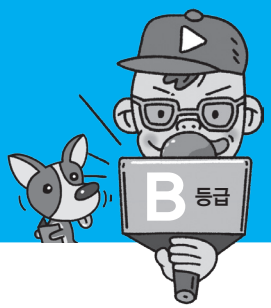
① 데이터 명명 규칙은 조직마다 다를 수 있으므로 설계 전에 미리 파악해야 한다.
② 물리 데이터베이스 설계와 논리 데이터베이스 설계의 명명 규칙은 각 단계의 특성에 따라 다양하게 지정해야 한다.
③ 데이터 명명 규칙은 물리 데이터베이스 설계 시 중복 구축을 방지하는데 도움을 준다.
④ 데이터 명명 규칙은 논리적 데이터 요소를 물리적 요소로 전환할 때 동일 명칭 부여의 근거로 사용된다.

물리 데이터베이스는 논리 데이터베이스를 변환하여 구성하는 것으로, 두 단계에 적용되는 명명 규칙은 일관성을 유지해야 합니다.

▶ 정답 : 1. ④ 2. ④ 3. ②

SECTION 086 ● LIVE

데이터베이스 저장 공간 설계



전문가의 조언

데이터베이스에 데이터를 저장하려면 테이블이나 컬럼 등 실제 데이터가 저장되는 공간을 정의해야 합니다. 테이블, 컬럼, 테이블스페이스의 개념과 특징을 숙지해 두세요.

1 테이블(Table)

테이블은 데이터베이스의 가장 기본적인 객체로 로우(Row, 행)와 컬럼(Column, 열)으로 구성되어 있다.

- 데이터베이스의 모든 데이터는 테이블에 저장된다.
- 테이블은 논리 설계 단계의 개체(Entity)에 대응하는 객체이다.
- DBMS 종류에 따라 테이블의 명칭과 기능 등은 약간씩 차이가 있다.
- 테이블의 종류에는 일반 테이블, 클러스터 인덱스 테이블, 파티셔닝 테이블, 외부 테이블, 임시 테이블 등이 있다.

전문가의 조언

일반 테이블은 데이터를 정렬하지 않고 가장 적절한 기억 장소에 저장한 후 임의의 방식으로 데이터를 관리하는 힙(Heap) 구조를 사용하므로 힙 구조 테이블(Heap-Organized Table)이라고도 합니다.

2 일반 테이블

일반 테이블은 현재 사용되는 대부분의 DBMS에서 표준 테이블로 사용되는 테이블 형태이다.

- 테이블에 저장되는 데이터의 로우(Row) 위치는 속성 값에 상관없이 데이터가 저장되는 순서에 따라 결정된다.

예 〈사원〉 테이블

사원번호	이름	입사일자	부서	직급	연봉
45S051	유순자	2018-05-09	영업부	사원	25,000,000
37S042	남민호	2019-01-30	기획부	과장	37,000,000
43S039	채진욱	2018-09-25	기획부	사원	28,000,000
36S201	박철수	2019-06-23	총무부	대리	35,000,000
32S122	안병수	2018-05-05	영업부	과장	41,000,000
32S219	임현희	2018-12-11	총무부	부장	55,000,000
42S235	오동근	2019-04-19	기획부	사원	29,000,000

데이터가 일정한 기준없이 입력되는 순서에 따라 테이블에 저장된다.

3 클러스터드 인덱스 테이블(Clustered Index Table)

클러스터드 인덱스 테이블은 기본키(Primary Key)나 인덱스키의 순서에 따라 데이터가 저장되는 테이블이다.

- 클러스터 인덱스 테이블은 일반적인 인덱스를 사용하는 테이블에 비해 접근 경로가 단축된다.

예 〈사원〉 테이블

사원번호	이름	입사일자	부서	직급	연봉
32S122	안병수	2018-05-05	영업부	과장	41,000,000
32S219	임현희	2018-12-11	총무부	부장	55,000,000
36S201	박철수	2019-06-23	총무부	대리	35,000,000
37S042	남민호	2019-01-30	기획부	과장	37,000,000
42S235	오동근	2019-04-19	기획부	사원	29,000,000
43S039	채진욱	2018-09-25	기획부	사원	28,000,000
45S051	유순자	2018-05-09	영업부	사원	25,000,000

데이터가 기본키인 '사원번호' 필드를 기준으로 정렬되어 테이블에 저장된다.

4 파티셔닝 테이블(Partitioning Table)

파티셔닝 테이블은 대용량의 테이블을 작은 논리적 단위인 파티션(Partition)으로 나눈 테이블이다.

- 파티셔닝 테이블은 대용량의 데이터를 효과적으로 관리할 수 있지만 파티션 키를 잘못 구성하면 성능 저하 등의 역효과를 초래할 수 있다.
- 파티셔닝 방식에 따라 범위 분할(Range Partitioning)*, 해시 분할(Hash Partitioning)*, 조합 분할(Composite Partitioning)* 등으로 나뉜다.

예 〈파트1〉 테이블

사원번호	이름	입사일자	부서	직급	연봉
32S122	안병수	2018-05-05	영업부	과장	41,000,000
45S051	유순자	2018-05-09	영업부	사원	25,000,000
43S039	채진욱	2018-09-25	기획부	사원	28,000,000
32S219	임현희	2018-12-11	총무부	부장	55,000,000

〈파트2〉 테이블

사원번호	이름	입사일자	부서	직급	연봉
37S042	남민호	2019-01-30	기획부	과장	37,000,000
42S235	오동근	2019-04-19	기획부	사원	29,000,000
36S201	박철수	2019-06-23	총무부	대리	35,000,000

'입사일자' 필드를 기준으로 연도별 파티셔닝 되었다.

전문가의 조언

파티셔닝에 대한 자세한 내용은 Section 091에서 공부합니다. 여기서는 파티셔닝의 개념 정도만 알아두세요.

- **범위 분할**(Range Partitioning) : 지정한 열의 값을 기준으로 분할
- **해시 분할**(Hash Partitioning) : 해시 함수를 적용한 결과 값에 따라 데이터를 분할
- **조합 분할**(Composite Partitioning) : 범위 분할로 분할한 다음 해시 함수를 적용하여 다시 분할하는 방식

5 외부 테이블(External Table)

외부 테이블은 데이터베이스에서 일반 테이블처럼 이용할 수 있는 외부 파일로, 데이터베이스 내에 객체로 존재한다.

- 외부 테이블은 데이터웨어하우스(Data Warehouse)*에서 ETL(Extraction, Transformation, Loading)* 등의 작업에 유용하게 사용된다.

6 임시 테이블(Temporary Table)

임시 테이블은 트랜잭션*이나 세션별로 데이터를 저장하고 처리할 수 있는 테이블이다.

- 임시 테이블에 저장된 데이터는 트랜잭션이 종료되면 삭제된다.
- 임시 테이블은 절차적인 처리를 위해 임시로 사용하는 테이블이다.

7 컬럼(Column)

컬럼은 테이블의 열을 구성하는 요소로 데이터 타입(Data Type), 길이(Length) 등으로 정의된다.

- 데이터 타입은 데이터의 일관성 유지를 위해 사용되는 가장 기본적인 것으로, 도메인을 정의한 경우 도메인에 따라 데이터의 타입과 길이가 정의된다.
- 두 컬럼을 비교하는 연산에서 두 컬럼의 데이터 타입이나 길이가 다르면 DBMS 내부적으로 데이터 타입을 변환한 후 비교 연산을 수행한다.
- 참조 관계인 컬럼들은 데이터 타입과 길이가 일치해야 한다.
- 데이터 타입과 길이 지정 시 고려 사항
  - 가변 길이 데이터 타입 : 예상되는 최대 길이로 정의
  - 고정 길이 데이터 타입 : 최소 길이로 지정
  - 소수점 이하 자릿수 : 소수점 이하 자릿수는 반올림되어 저장
- 데이터 타입에 따른 컬럼의 물리적인 순서
  - 고정 길이 컬럼이고 NOT Null인 컬럼 : 앞쪽
  - 가변 길이 컬럼 : 뒤쪽
  - Null 값이 많을 것으로 예상되는 컬럼 : 뒤쪽

8 테이블스페이스(Tablespace)

테이블스페이스는 테이블이 저장되는 논리적인 영역으로, 하나의 테이블스페이스에 하나 또는 그 이상의 테이블을 저장할 수 있다.

- 테이블을 저장하면 논리적으로는 테이블스페이스에 저장되고, 물리적으로는 해당 테이블스페이스와 연관된 데이터 파일(Data File)에 저장된다.

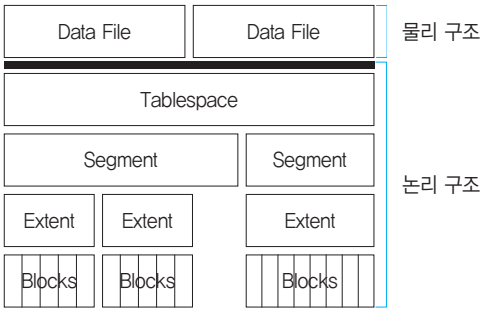
데이터웨어하우스(Data Warehouse)
데이터웨어하우스란 조직이나 기업체의 중심이 되는 주요 업무 시스템에서 추출되어 새로이 생성된 데이터베이스로서, 의사결정지원 시스템을 지원하는 주체적, 통합적, 시간적 데이터의 집합체를 말합니다.

ETL(Extraction, Transformation, Loading)
ETL은 데이터웨어하우스를 사용하여 추출(Extract), 변환(Transform), 적재(Load) 하는 일련의 모든 과정을 의미합니다.

트랜잭션(Transaction)
트랜잭션은 데이터베이스의 상태를 변환시키는 하나의 논리적 기능을 수행하기 위한 작업의 단위 또는 한꺼번에 모두 수행되어야 할 일련의 연산들을 의미합니다.

- 데이터베이스를 테이블, 테이블스페이스, 데이터 파일로 나눠 관리하면 논리적 구성이 물리적 구성에 종속되지 않아 투명성*이 보장된다.
- 테이블스페이스는 데이터베이스에 저장되는 내용에 따라 테이블, 인덱스, 임시(Temporary) 등의 용도로 구분하여 설계한다.
- 테이블스페이스 설계 시 고려사항
  - 테이블스페이스는 업무별로 구분하여 지정한다.
  - 대용량 테이블은 하나의 테이블스페이스에 독립적으로 저장한다.
  - 테이블과 인덱스는 분리하여 저장한다.
  - LOB(Large Object)* 타입의 데이터는 독립적인 공간으로 지정한다.

예 Oracle의 물리/논리 저장 구조 계층



- Oracle 프로그램은 데이터베이스를 Block, Extent, Segment, Tablespace, Data File로 나눠 관리한다. Block, Extent, Segment, Tablespace는 논리적인 영역이고, Data File은 물리적인 영역이다.
- Segment는 테이블, Extent는 데이터 블록의 모임, Block은 입 · 출력 단위에 해당한다.

투명성
투명성이란 어떠한 사실이 존재함에도 마치 투명하여 보이지 않는 것처럼, 사실의 존재 여부를 염두에 두지 않아도 되는 성질을 말합니다.

LOB(Large Object)
LOB(Large Object)는 대용량의 데이터를 저장할 수 있는 데이터 타입입니다.

기출문제 따라잡기 Section 086

출제예상

1. 데이터베이스의 모든 데이터가 저장되는 테이블에 대한 설명으로 틀린 것은?

① 테이블은 로우(Row)와 컬럼(Column)으로 구성되어 있다.

② 테이블은 논리 데이터 모델의 속성에 대응하는 객체이다.

③ 테이블의 종류에는 일반 유형 테이블, 클러스터 인덱스 테이블, 파티셔닝 테이블 등이 있다.

④ 테이블의 구성 요소인 컬럼은 데이터 타입과 길이로 정의된다.

테이블은 개체(Entity), 컬럼은 속성(Attribute)! 꼭 기억하세요.

출제예상

2. 다음 중 테이블에 대한 설명으로 가장 옳지 않은 것은?

① 일반 유형 테이블에 저장되는 데이터의 로우(Row) 위치는 속성 값에 상관없이 데이터가 저장되는 순서에 따라 결정된다.

② 클러스터 인덱스 테이블은 기본키나 인덱스 키의 순서에 따라 데이터가 저장된다.

③ 임시 테이블은 절차적인 처리를 위해 임시로 사용하는 테이블이다.

④ 외부 테이블은 트랜잭션별로 데이터를 저장하고 처리할 수 있는 테이블이다.

외부 테이블은 외부 파일을 데이터베이스 내의 테이블처럼 사용할 수 있는 것을 말합니다.

출제예상

3. 다음 중 컬럼에 대한 설명으로 잘못된 것은?

① 컬럼은 테이블을 구성하는 요소이다.

② 컬럼은 데이터 타입과 길이로 정의된다.

③ 데이터 타입과 길이가 다른 컬럼의 값은 비교할 수 없다.

④ 참조 관계인 컬럼들은 서로 데이터 타입과 길이가 동일해야 한다.

두 컬럼을 비교하는 연산에서 두 컬럼의 데이터 타입이나 길이가 다르면 DBMS 내부적으로 데이터 타입을 변환한 후 비교 연산을 수행한다.

출제예상

4. 다음 중 테이블스페이스에 대한 설명으로 틀린 것은?

① 테이블스페이스는 테이블이 저장되는 물리적인 영역이다.

② 하나의 테이블스페이스에 하나 또는 그 이상의 테이블을 저장할 수 있다.

③ 테이블스페이스는 하나 또는 여러 개의 데이터 파일을 가진다.

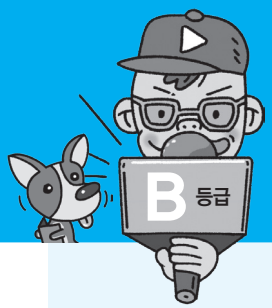
④ 테이블, 테이블스페이스, 데이터 파일로 나눠 관리하면 논리적 구성이 물리적 구성에 종속되지 않아 투명성이 보장된다.

테이블스페이스는 논리적인 영역이고, 데이터 파일이 물리적인 영역입니다.

▶ 정답 : 1. ② 2. ④ 3. ③ 4. ①

SECTION 087 ● LIVE

트랜잭션 분석 / CRUD 분석



1 트랜잭션(Transaction) 정의

트랜잭션은 데이터베이스의 상태를 변환시키는 하나의 논리적 기능을 수행하기 위한 작업의 단위 또는 한꺼번에 모두 수행되어야 할 일련의 연산들을 의미한다.

- 트랜잭션은 데이터베이스 시스템에서 병행 제어 및 회복 작업 시 처리되는 작업의 논리적 단위로 사용된다.
- 트랜잭션은 사용자가 시스템에 대한 서비스 요구 시 시스템이 응답하기 위한 상태 변환 과정의 작업 단위로 사용된다.

2 트랜잭션의 특성

다음은 데이터의 무결성(Integrity)을 보장하기 위하여 DBMS의 트랜잭션이 가져야 할 특성이다.

Atomicity(원자성)	• 트랜잭션의 연산은 데이터베이스에 모두 반영되도록 완료(Commit)되든지 아니면 전혀 반영되지 않도록 복구(Rollback)되어야 한다. • 트랜잭션 내의 모든 명령은 반드시 완벽히 수행되어야 하며, 모두가 완벽히 수행되지 않고 어느 하나라도 오류가 발생하면 트랜잭션 전부가 취소되어야 한다.
Consistency(일관성)	• 트랜잭션이 그 실행을 성공적으로 완료하면 언제나 일관성 있는 데이터베이스 상태로 변환한다. • 시스템이 가지고 있는 고정 요소는 트랜잭션 수행 전과 트랜잭션 수행 완료 후의 상태가 같아야 한다.
Isolation (독립성, 격리성, 순차성)	• 둘 이상의 트랜잭션이 동시에 병행 실행되는 경우 어느 하나의 트랜잭션 실행 중에 다른 트랜잭션의 연산이 끼어들 수 없다. • 수행중인 트랜잭션은 완전히 완료될 때까지 다른 트랜잭션에서 수행 결과를 참조할 수 없다.
Durability (영속성, 지속성)	성공적으로 완료된 트랜잭션의 결과는 시스템이 고장나더라도 영구적으로 반영되어야 한다.

3 CRUD 분석

CRUD는 '생성(Create), 읽기(Read), 갱신(Update), 삭제(Delete)'의 앞 글자만 모아서 만든 용어이며, CRUD 분석은 데이터베이스 테이블에 변화를 주는 트랜잭션의 CRUD 연산에 대해 CRUD 매트릭스를 작성하여 분석하는 것이다.

전문가의 조언

물리 데이터베이스를 설계하려면 데이터베이스에 어떤 트랜잭션이 얼마나 자주 발생하는지 분석하고 그에 따라 트랜잭션 처리 방법이나 데이터베이스 구조 등을 설계해야 합니다. 먼저 트랜잭션의 개념을 잘 이해하세요. 그리고 CRUD 분석을 기반으로 트랜잭션을 분석하는 방법과 이를 토대로 트랜잭션 분석서를 작성하는 방법을 정리하세요.

전문가의 조언

트랜잭션의 특성을 영어 앞글자만 모아서 ACID라고 합니다. 이 네 가지 종류와 각각의 의미를 꼭 숙지하세요.

전문가의 조언

CRUD 분석은 테이블에 발생하는 트랜잭션을 분석하여 물리적인 데이터베이스 설계 시 구조를 최적화하는데 목적이 있습니다. CRUD 분석에 사용되는 CRUD 매트릭스와 분석서가 어떻게 작성되는지 알아두세요.

- CRUD 분석으로 테이블에 발생되는 트랜잭션의 주기별 발생 횟수를 파악하고 연관된 테이블들을 분석하면 테이블에 저장되는 데이터의 양을 유추할 수 있다.
- CRUD 분석을 통해 많은 트랜잭션이 몰리는 테이블을 파악할 수 있으므로 디스크 구성 시 유용한 자료로 활용할 수 있다.
- CRUD 분석을 통해 외부 프로세스 트랜잭션의 부하가 집중되는 데이터베이스 채널을 파악하고 분산시킴으로써 연결 지연이나 타임아웃 오류를 방지할 수 있다.

4 CRUD 매트릭스

CRUD 매트릭스는 2차원 형태의 표로서, 열(Row)에는 프로세스를, 행(Column)에는 테이블을, 행과 열이 만나는 위치에는 프로세스가 테이블에 발생시키는 변화를 표시하는 업무 프로세스와 데이터 간 상관 분석표이다.

- CRUD 매트릭스를 통해 프로세스의 트랜잭션이 테이블에 수행하는 작업을 검증한다.
- CRUD 매트릭스의 각 셀에는 Create, Read, Update, Delete의 앞 글자가 들어가며, 복수의 변화를 줄 때는 기본적으로 'C 〉 D 〉 U 〉 R'의 우선순위를 적용하여 한 가지만 적지만, 활용 목적에 따라 모두 기록할 수 있다.

  예 '주문 변경' 프로세스를 실행하려면 테이블의 데이터를 읽은(Read) 다음 수정(Update) 해야 하므로 R(Read)과 U(Update)가 필요하지만 CRUD 매트릭스에는 우선순위가 높은 'U'만 표시한다.

- CRUD 매트릭스가 완성되었다면 C, R, U, D 중 어느 것도 적히지 않은 행이나 열, C나 R이 없는 행을 확인하여 불필요하거나 누락된 테이블 또는 프로세스를 찾는다.

  예 온라인 쇼핑몰의 CRUD 매트릭스 예시

테이블 / 프로세스	회원	상품	주문	주문목록	제조사
신규 회원 등록	C				
회원정보 변경	R, U				
주문 요청	R	R	C	C	
주문 변경			R	R, U	
주문 취소			R, D	R, D	
상품 등록		C			C, R
상품정보 변경		R, U			R, U

5 트랜잭션 분석

트랜잭션 분석의 목적은 CRUD 매트릭스를 기반으로 테이블에 발생하는 트랜잭션 양을 분석하여 테이블에 저장되는 데이터의 양을 유추*하고 이를 근거로 DB 용량을 산정하고 DB 구조를 최적화하는 것이다.

- 트랜잭션 분석은 업무 개발 담당자*가 수행한다.
- 트랜잭션 분석을 통해 프로세스가 과도하게 접근하는 테이블을 확인하여 여러 디스크에 배치함으로써 디스크 입 · 출력 분산을 통한 성능 향상을 가져올 수 있다.

6 트랜잭션 분석서

트랜잭션 분석서는 단위 프로세스와 CRUD 매트릭스를 이용하여 작성하며, 구성 요소에는 단위 프로세스, CRUD 연산, 테이블명, 컬럼명, 테이블 참조 횟수, 트랜잭션 수, 발생 주기 등이 있다.

- **단위 프로세스** : 업무를 발생시키는 가장 작은 단위의 프로세스
- **CRUD 연산*** : 프로세스의 트랜잭션이 데이터베이스 테이블에 영향을 주는 C, R, U, D의 4가지 연산
- **테이블명, 컬럼명** : 프로세스가 접근하는 데이터베이스의 테이블명을 기록한다. 필요한 경우 테이블의 컬럼명을 적는다. 컬럼명을 적을 때는 마침표로 연결하여 '테이블.컬럼명'과 같이 적는다.
- **테이블 참조 횟수** : 프로세스가 테이블을 참조하는 횟수
- **트랜잭션 수** : 주기별로 수행되는 트랜잭션 횟수
- **발생 주기** : 연, 분기, 월, 일, 시간 등 트랜잭션 횟수를 측정하기 위한 발생 주기

예 '주문요청' 프로세스에 대한 트랜잭션 분석서 예시

프로세스	CRUD	테이블명	컬럼명	참조 횟수	트랜잭션 수	주기
주문요청	R	회원	회원번호, 회원명, 주소	1	150	일
	R	상품	상품번호, 상품명, 재고량	1	150	
	C	주문	주문번호, 일자, 회원번호	3	450	
	C	주문목록	주문번호, 상품번호, 수량, 가격	5	750	

전문가의 조언

테이블(열)이나 프로세스(행)에 C, R, U, D 중 아무것도 입력된 게 없으면 해당 테이블과 프로세스는 사용되지 않는 것이니 삭제를 검토해야 합니다. C가 없는 테이블은 입력된 것이 없는데 사용한 것이고, R이 없으면 입력되었지만 사용되지 않은 것이므로 누락된 프로세스가 없는지 검토해야 합니다.

데이터 양 유추

직원이 10,000명이고, 직원 한 명당 발령은 2번, 상여금 변경은 20번 발생한다면 발령 테이블에는 10,000×2 = 20,000건, 상여금 테이블에는 10,000×20 = 200,000건의 데이터가 발생한다는 것을 유추할 수 있습니다.

※ 이 계산은 정확한 데이터 양을 계산한 것이 아니라 저장장치나 파일 등의 크기를 지정하기 위해 대략적으로 유추한 것입니다.

업무 개발 담당자

업무 개발 담당자는 애플리케이션을 분석 및 설계하는 일을 담당합니다.

CRUD 연산

트랜잭션은 데이터베이스의 상태를 변화시키는 논리적 기능을 수행합니다. 데이터베이스에 영향을 주는 데이터 처리 연산인 C, R, U, D가 바로 트랜잭션이 수행하는 연산에 해당합니다.

이전기출

1. 트랜잭션(Transaction)의 특성에 대한 설명으로 옳지 않은 것은?

① 원자성(Atomicity)은 트랜잭션의 일부만 수행된 상태로 종료될 수 있다는 특성을 의미한다.

② 일관성(Consistency)은 시스템의 고정 요소는 트랜잭션 수행 전과 수행 완료 후에 같아야 한다는 특성을 의미한다.

③ 격리성(Isolation)은 트랜잭션이 실행될 때마다 다른 트랜잭션의 간섭을 받지 않아야 한다는 성질을 의미한다.

④ 지속성(Durability)은 트랜잭션의 완료 결과가 데이터베이스에 영구히 기억되는 성질을 의미한다.

원자성은 All Or Nothing, 즉 전부가 아니면 아무것도 아니라는 뜻입니다. 완전히 수행하든지 하나도 수행하지 말든지…, 일부는 인정하지 않겠다는 뜻입니다.

출제예상

2. 데이터베이스에 영향을 주는 생성, 읽기, 갱신, 삭제 연산으로 프로세스와 테이블 간에 매트릭스를 만들어 트랜잭션을 분석하는 것은?

① 불일치 분석

② CASE 분석

③ CRUD 분석

④ 경계값 분석

생성(Create), 읽기(Read), 갱신(Update), 삭제(Delete)의 영문 앞 글자만 모아보세요. 이제 답이 보이죠?

출제예상

3. CRUD 분석에 대한 설명으로 옳지 않은 것은?

① CRUD 매트릭스를 작성하여 분석하는 기법이다.

② 생성 트랜잭션 규모를 파악하여 구축해야 할 스토리지의 규모를 예측할 수 있다.

③ CRUD 매트릭스에는 C〉R〉U〉D의 우선순위에 따라 한 가지만 적어야 한다.

④ 외부 프로세스의 트랜잭션이 집중되는 채널을 파악하여 분산시킬 수 있다.

CRUD 매트릭스에 기록하는 CRUD 연산의 우선순위는 'C〉D〉U〉R' 입니다. 또한 매트릭스의 활용 목적에 따라 4개의 연산을 모두 적을 수도 있습니다.

출제예상

4. CRUD 매트릭스에서 검토사항이 아닌 것은?

① 테이블에 C, R, U, D가 모두 없는 경우

② 테이블에 C 또는 R이 없는 경우

③ 프로세스에 C, R, U, D가 모두 없는 경우

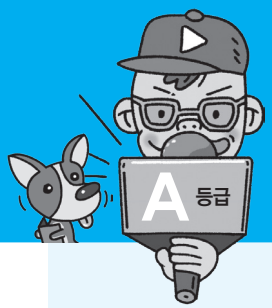
④ 프로세스에 C 또는 R이 없는 경우

프로세스는 생성(Create) 또는 읽기(Read)가 없을 수 있습니다. 예를 들어 신규 회원을 등록하는 프로세스는 읽기(Read) 연산 없이 회원으로부터 정보를 받아 〈회원〉 테이블에 데이터를 기록하는 생성(Create) 연산만 수행할 수 있습니다.

▶ 정답 : 1. ① 2. ③ 3. ③ 4. ④

SECTION 088 ● LIVE

인덱스 설계



1 인덱스(Index)의 개념

인덱스는 데이터 레코드를 빠르게 접근하기 위해 〈키 값, 포인터〉 쌍으로 구성되는 데이터 구조이다.

예 학생 릴레이션의 학번 속성에 대한 인덱스 예

〈인덱스〉

학번(키 값)	주소(포인터)
18032	300
19054	200
18065	100
17083	400

주소: 100, 400, 300, 200

〈학생〉

학번	이름	학년	학과
18065	강홍구	3	컴퓨터공학과
17083	김동오	4	경영학과
18032	정연중	3	영문과
19054	심희정	2	방송학과

위의 인덱스에서 키 값은 '학번'이고 포인터는 해당 '학번'이 저장된 레코드의 물리적인 주소이다. 키 값인 '학번'이 정렬되어 있기 때문에 인덱스를 통해 레코드를 빠르게 접근할 수 있다.

- 인덱스는 데이터가 저장된 물리적 구조와 밀접한 관계가 있다.
- 인덱스는 레코드가 저장된 물리적 구조에 접근하는 방법을 제공한다.
- 인덱스를 통해서 파일의 레코드에 대한 액세스를 빠르게 수행할 수 있다.
- 레코드의 삽입과 삭제가 수시로 일어나는 경우에는 인덱스의 개수를 최소로 하는 것이 효율적이다.
- 인덱스가 없으면 특정한 값을 찾기 위해 모든 데이터 페이지를 확인하는 TABLE SCAN*이 발생한다.
- 기본키를 위한 인덱스를 기본 인덱스라 하고, 기본 인덱스가 아닌 인덱스들을 보조 인덱스라고 한다. 대부분의 관계형 데이터베이스 관리 시스템에서는 모든 기본키에 대해서 자동적으로 기본 인덱스를 생성한다.
- 레코드의 물리적 순서가 인덱스의 엔트리 순서와 일치하게 유지되도록 구성되는 인덱스를 클러스터드(Clustered) 인덱스라고 한다.
- 인덱스는 인덱스를 구성하는 구조나 특징에 따라 트리 기반 인덱스, 비트맵 인덱스, 함수 기반 인덱스, 비트맵 조인 인덱스, 도메인 인덱스 등으로 분류된다.

전문가의 조언

인덱스의 개념과 각 인덱스의 특징을 알아두세요. 어떤 인덱스를 의미하는지는 이해할 수 있을 정도로는 알고 있어야 합니다.

TABLE SCAN

TABLE SCAN은 테이블에 있는 모든 레코드를 순차적으로 읽는 것으로, FULL TABLE SCAN이라고도 합니다. 일반적으로 적용 가능한 인덱스가 없거나 분포도가 넓은 데이터를 검색할 때는 FULL TABLE SCAN을 사용합니다.

잠깐만요

클러스터드 인덱스 / 넌클러스터드 인덱스

클러스터드 인덱스(Clustered Index)

- 인덱스 키의 순서에 따라 데이터가 정렬되어 저장되는 방식입니다.
- 실제 데이터가 순서대로 저장되어 있어 인덱스를 검색하지 않아도 원하는 데이터를 빠르게 찾을 수 있습니다.
- 데이터 삽입, 삭제 발생 시 순서를 유지하기 위해 데이터를 재정렬해야 합니다.
- 한 개의 릴레이션에 하나의 인덱스만 생성할 수 있습니다.

넌클러스터드 인덱스(Non-Clustered Index)

- 인덱스의 키 값만 정렬되어 있을 뿐 실제 데이터는 정렬되지 않는 방식입니다.
- 데이터를 검색하기 위해서는 먼저 인덱스를 검색하여 실제 데이터의 위치를 확인해야 하므로 클러스터드 인덱스에 비해 검색 속도가 떨어집니다.
- 한 개의 릴레이션에 여러 개의 인덱스를 만들 수 있습니다.

2 트리 기반 인덱스

트리 기반 인덱스는 인덱스를 저장하는 블록들이 트리 구조를 이루고 있는 것으로, 상용 DBMS에서는 트리 구조 기반의 B+ 트리 인덱스를 주로 활용한다.

- B 트리 인덱스
  - 일반적으로 사용되는 인덱스 방식으로, 루트 노드에서 하위 노드로 키 값의 크기를 비교해 나가면서 단말 노드에서 찾고자 하는 데이터를 검색한다.
  - 키 값과 레코드를 가리키는 포인터들이 트리 노드에 오름차순으로 저장된다.
  - 모든 리프 노드는 같은 레벨에 있다.
- B+ 트리 인덱스
  - B+ 트리는 B 트리의 변형으로 단말 노드가 아닌 노드로 구성된 인덱스 세트(Index Set)와 단말 노드로만 구성된 순차 세트(Sequence Set)로 구분된다.
  - 인덱스 세트에 있는 노드들은 단말 노드에 있는 키 값을 찾아갈 수 있는 경로로만 제공되며, 순차 세트에 있는 단말 노드가 해당 데이터 레코드의 주소를 가리킨다.
  - 인덱스 세트에 있는 모든 키 값이 단말 노드에 다시 나타나므로 단말 노드만을 이용한 순차 처리가 가능하다.

3 비트맵 인덱스

비트맵 인덱스는 인덱스 컬럼의 데이터를 Bit 값인 0 또는 1로 변환하여 인덱스 키로 사용하는 방법이다.

- 비트맵 인덱스의 목적은 키 값을 포함하는 로우(Row)의 주소*를 제공하는 것이다.
- 비트맵 인덱스는 분포도*가 좋은 컬럼에 적합하며, 성능 향상 효과를 얻을 수 있다.

로우(Row)의 주소

비트맵에서 비트의 위치는 테이블에서 로우(Row)의 상대적인 위치를 의미합니다. 해당 테이블이 시작되는 물리적인 주소를 기반으로 실제 로우의 물리적 위치를 계산할 수 있습니다.

분포도, 선택성(Selectivity)

- (조건에 맞는 레코드 수 / 전체 릴레이션 레코드 수) × 100
- 전체 레코드 중 조건에 맞는 레코드의 숫자가 적은 경우 분포도가 좋다고 합니다.
- 분포도가 10~15%인 경우 효율적인 인덱스 검색을 할 수 있습니다.
- 분포도를 선택성(Selectivity)이란 용어로 사용하기도 합니다.

- 데이터가 Bit로 구성되어 있기 때문에 효율적인 논리 연산이 가능하고 저장 공간이 작다.
- 비트맵 인덱스는 다중 조건을 만족하는 튜플의 개수 계산에 적합하다.
- 비트맵 인덱스는 동일한 값이 반복되는 경우가 많아 압축 효율이 좋다.

4 함수 기반 인덱스

함수 기반 인덱스는 컬럼의 값 대신 컬럼에 특정 함수(Function)나 수식(Expression)을 적용하여 산출된 값을 사용하는 것으로, B+ 트리 인덱스 또는 비트맵 인덱스를 생성하여 사용한다.

- 함수 기반 인덱스는 데이터를 입력하거나 수정할 때 함수를 적용해야 하므로 부하가 발생할 수 있다.
- 사용된 함수가 사용자 정의 함수일 경우 시스템 함수보다 부하가 더 크다.
- 함수 기반 인덱스는 대소문자, 띄어쓰기 등에 상관없이 조회할 때 유용하게 사용된다.
- **적용 가능한 함수의 종류** : 산술식(Arithmetic Expression), 사용자 정의 함수, PL/SQL Function, SQL Function, Package, C callout 등

5 비트맵 조인 인덱스

비트맵 조인 인덱스는 다수의 조인된 객체로 구성된 인덱스로, 단일 객체로 구성된 일반적인 인덱스와 액세스 방법이 다르다.

- 비트맵 조인 인덱스는 비트맵 인덱스와 물리적 구조가 동일하다.

6 도메인 인덱스*

도메인 인덱스는 개발자가 필요한 인덱스를 직접 만들어 사용하는 것으로, 확장형 인덱스(Extensible Index)라고도 한다.

- 개발자가 필요에 의해 만들었지만 프로그램에서 제공하는 인덱스처럼 사용할 수도 있다.

도메인 인덱스
도메인 인덱스는 오라클의 버전 8i에서부터 도입된 새로운 개념의 인덱스입니다.

7 인덱스 설계*

인덱스를 설계할 때는 분명하게 드러난 컬럼에 대해 기본적인 인덱스를 먼저 지정한 후 개발 단계에서 필요한 인덱스의 설계를 반복적으로 진행한다.

- 인덱스 설계 순서
  ❶ 인덱스의 대상 테이블이나 컬럼 등을 선정한다.
  ❷ 인덱스의 효율성을 검토하여 인덱스 최적화를 수행한다.
  ❸ 인덱스 정의서를 작성한다.

인덱스 설계
프로젝트 진행중에는 데이터나 프로세스 내용 등에 변경이 많기 때문에 처음부터 모든 인덱스를 한꺼번에 정의하는 것은 불가능합니다. 인덱스 설계 단계에서는 기본적인 인덱스를 지정하고 프로젝트를 진행하면서 SQL 문장 구조 등을 검토하여 반복적으로 인덱스 설계를 진행해야 합니다.

예 인덱스 정의서 예시

엔티티명	테이블명	인덱스명	컬럼명	타입	테이블 스페이스	인덱스 유형	정렬	구분
사원	EMP	PK_EMP	EMPNO	NUMBER(5)	ISTEXT01	UNIQUE	ASC	PK INDEX
		1_EMP2	DEPTNO	NUMBER(5)	ISTEXT01	NOT UNIQUE	ASC	UD INDEX
고객	CUST	PK_CUST	CUSTNO	NUMBER(5)	ISTEXT01	UNIQUE	ASC	PK INDEX
부서	DEPT	PK_DEPT	DEPTNO	NUMBER(3)	ISTEXT01	UNIQUE	ASC	PK INDEX
주문	ORDER	PK_ORDER	ORDERNO	NUMBER(6)	ISTEXT01	UNIQUE	ASC	PK INDEX

8 인덱스 대상 테이블 선정 기준

- MULTI BLOCK READ* 수에 따라 판단
  예 MULTI BLOCK READ가 16이면, 테이블의 크기가 16블록 이상일 경우 인덱스 필요
- 랜덤 액세스가 빈번한 테이블
- 특정 범위나 특정 순서로 데이터 조회가 필요한 테이블
- 다른 테이블과 순차적 조인이 발생되는 테이블

MULTI BLOCK READ
MULTI BLOCK READ란 테이블 액세스 시 메모리에 한 번에 읽어 들일 수 있는 블록의 수를 말합니다.

9 인덱스 대상 컬럼 선정 기준

- 인덱스 컬럼의 분포도가 10~15% 이내인 컬럼
  – 분포도 = (컬럼값의 평균 Row 수 / 테이블의 총 Row 수) × 100
- 분포도가 10~15% 이상이어도 부분 처리를 목적으로 하는 컬럼
- 입 · 출력 장표 등에서 조회 및 출력 조건으로 사용되는 컬럼
- 인덱스가 자동 생성되는 기본키와 Unique키 제약 조건을 사용한 컬럼
- 가능한 한 수정이 빈번하지 않은 컬럼
- ORDER BY, GROUP BY, UNION*이 빈번한 컬럼
- 분포도가 좁은 컬럼은 단독 인덱스로 생성
- 인덱스들이 자주 조합되어 사용되는 경우 하나의 결합 인덱스(Concatenate Index)*로 생성

ORDER BY, GROUP BY, UNION
ORDER BY는 정렬, GROUP BY는 그룹 지정, UNION은 통합 지정 시 사용하는 SQL 명령어입니다.

결합 인덱스
- 결합 인덱스란 한 릴레이션 내에 존재하는 여러 컬럼들을 묶어 하나의 인덱스로 만드는 것을 말합니다.
- 결합 인덱스는 컬럼 순서에 따라 액세스하는 범위가 달라질 수 있으므로 유의해야 합니다.
- 컬럼 순서 우선순위
  ① 항상 사용되는 컬럼
  ② '=' 연산이 되는 컬럼
  ② 분포도가 좋은 컬럼
  ③ 정렬이 자주 발생하는 컬럼

10 인덱스 설계 시 고려사항

- 새로 추가되는 인덱스는 기존 액세스 경로에 영향을 미칠 수 있다.
- 인덱스를 지나치게 많이 만들면 오버헤드가 발생한다.
- 넓은 범위를 인덱스로 처리하면 많은 오버헤드가 발생한다.
- 인덱스를 만들면 추가적인 저장 공간이 필요하다.
- 인덱스와 테이블 데이터의 저장 공간이 분리되도록 설계한다.*

인덱스와 테이블 분리
인덱스와 테이블을 분리하는 형태는 데이터베이스의 가장 일반적인 형태로, 데이터 저장 시 인덱스의 영향을 받지 않아 저장이 빠릅니다.

따라잡기

기출문제 따라잡기

Section 088

이전기출

1. 인덱스(Index)에 대한 설명으로 부적절한 것은?

① 인덱스는 데이터베이스의 물리적 구조와 밀접한 관계가 있다.

② 인덱스는 하나 이상의 필드로 만들어도 된다.

③ 레코드의 삽입과 삭제가 수시로 일어나는 경우는 인덱스의 개수를 최대한 많게 한다.

④ 인덱스를 통해서 테이블의 레코드에 대한 액세스를 빠르게 수행할 수 있다.

테이블에 레코드를 삽입 또는 삭제할 때는 인덱스도 변경해야 하는데, 인덱스의 개수가 많을 경우 삽입과 삭제 시 매 번 모든 인덱스를 갱신하느라 속도가 느려집니다. 삽입과 삭제가 수시로 일어나는 경우에는 인덱스의 개수를 최소로 하는 것이 좋습니다.

출제예상

2. 다음 중 함수 기반 인덱스에 대한 설명으로 옳지 않은 것은?

① 함수 기반 인덱스는 컬럼 값 자체가 아니라 컬럼에 특정 함수를 적용한 값이다.

② 시스템 함수를 사용하면 사용자 정의 함수를 사용할 때보다 더 많은 부하가 발생한다.

③ 함수 기반 인덱스는 데이터를 입력하거나 수정할 때 함수를 적용해야 하므로 부하가 발생할 수 있다.

④ 함수 기반 인덱스로 B+ 트리 인덱스 또는 비트맵 인덱스를 생성하여 사용한다.

시스템 함수는 DBMS 자체에서 제공하는 함수로, 사용자가 만든 사용자 정의 함수에 비해 부하가 덜 발생합니다.

출제예상

3. 다음 중 인덱스에 대한 설명으로 틀린 것은?

① 프로젝트 진행시 인덱스 설계는 처음에 완벽하게 설계해야 이후 작업에 오류가 발생하지 않는다.

② 인덱스를 많이 지정하면 오버헤드가 발생할 수 있으므로 적절히 지정한다.

③ 인덱스와 테이블 데이터의 저장 공간은 적절히 분리되도록 설계해야 한다.

④ 랜덤 액세스가 빈번히 발생하는 테이블을 인덱스 테이블로 선정한다.

프로젝트를 진행하다보면 데이터나 프로세스 등이 계속해서 변경되는데 처음부터 완벽한 인덱스를 설계할 수 있을까요?

▶ 정답 : 1. ③ 2. ② 3. ①

SECTION 089 ● LIVE

뷰(View) 설계

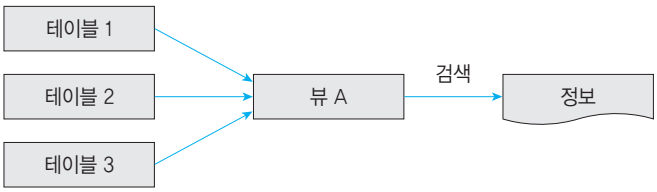
전문가의 조언

뷰의 의미, 특징, 장점과 단점 모두 중요합니다. 뷰의 의미를 충분히 이해하면 특징이나 장·단점도 쉽게 이해됩니다.

1 뷰(View)의 개요

뷰는 사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위해 하나 이상의 기본 테이블로부터 유도된, 이름을 가지는 가상 테이블이다.

- 뷰는 저장장치 내에 물리적으로 존재하지 않지만, 사용자에게는 있는 것처럼 간주된다.
- 뷰는 데이터 보정 작업, 처리 과정 시험 등 임시적인 작업을 위한 용도로 활용된다.
- 뷰는 조인문의 사용 최소화로 사용상의 편의성을 최대화한다.
- 뷰를 생성하면 뷰 정의가 시스템 내에 저장되었다가 생성된 뷰 이름을 질의어(예를 들면 SQL)에서 사용할 경우 질의어가 실행될 때 뷰에 정의된 기본 테이블로 대체되어 기본 테이블에 대해 실행된다.
- 다음 그림은 뷰 A가 테이블 1, 테이블 2, 테이블 3에서 유도되어 생성되며, 뷰 A를 통해 테이블 1, 테이블 2, 테이블 3에 대한 데이터에 접근할 수 있음을 나타낸 것이다.



2 뷰(View)의 특징

- 뷰는 기본 테이블로부터 유도된 테이블이기 때문에 기본 테이블과 같은 형태의 구조를 사용하며, 조작도 기본 테이블과 거의 같다.
- 뷰는 가상 테이블이기 때문에 물리적으로 구현되어 있지 않다.
- 데이터의 논리적 독립성을 제공할 수 있다.
- 필요한 데이터만 뷰로 정의*해서 처리할 수 있기 때문에 관리가 용이하고 명령문이 간단해진다.
- 뷰를 통해서만 데이터에 접근하게 하면 뷰에 나타나지 않는 데이터를 안전하게 보호하는 효율적인 기법으로 사용할 수 있다.
- 기본 테이블의 기본키를 포함한 속성(열) 집합으로 뷰를 구성해야만 삽입, 삭제, 갱신 연산이 가능하다.
- 일단 정의된 뷰는 다른 뷰의 정의에 기초가 될 수 있다.

정의 테이블

뷰는 하나 이상의 다른 테이블로부터 유도된 하나의 가상 테이블이며, 뷰를 만들기 위해 유도된 기본 테이블을 정의 테이블이라 합니다.

- 뷰가 정의된 기본 테이블이나 뷰를 삭제하면 그 테이블이나 뷰를 기초로 정의된 다른 뷰도 자동으로 삭제된다.

3 뷰(View)의 장 · 단점

장점	단점
• 논리적 데이터 독립성을 제공한다. • 동일 데이터에 대해 동시에 여러 사용자의 상이한 응용이나 요구를 지원해 준다. • 사용자의 데이터 관리를 간단하게 해준다. • 접근 제어를 통한 자동 보안이 제공된다.	• 독립적인 인덱스를 가질 수 없다. • 뷰의 정의를 변경할 수 없다. • 뷰로 구성된 내용에 대한 삽입, 삭제, 갱신 연산에 제약이 따른다.

4 뷰 설계 순서

❶ 대상 테이블을 선정한다.

– 외부 시스템과 인터페이스에 관여하는 테이블

– CRUD 매트릭스를 통해 여러 테이블이 동시에 자주 조인되어 접근되는 테이블

– SQL문 작성 시 거의 모든 문장에서 인라인 뷰* 방식으로 접근되는 테이블

❷ 대상 컬럼을 선정한다.

– 보안을 유지해야 하는 컬럼은 주의하여 선별한다.

❸ 정의서를 작성한다.

– 뷰 정의서 예시

뷰명	뷰 설명	관련 테이블	관련 컬럼명	데이터 타입
V_SELLITEM	판매와 판매목록 처리	SELL	SELLNO SELLNAME SELLDATE	VARCHAR(5) VARCHAR(40) DATE
		SELLITEM	ITEMNO PRICE	VARCHAR(5) NUMBER(10)

인라인 뷰(Inline View)
FROM절 안에 사용되는 서브 쿼리를 '인라인 뷰'라고 합니다.

예

```
select a.ban, a.name, a.exam
from student a, (select ban,
avg(exam) avgs from student
group by ban) b
where a.ban = b.ban and a.exam
> b.avgs;
```

인라인 뷰를 이용하여 반별로 'exam' 필드의 평균을 조회한 후 'exam'이 평균보다 큰 학생의 'ban', 'name', 'exam'를 표시합니다.

5 뷰 설계 시 고려 사항

- 테이블 구조가 단순화 될 수 있도록 반복적으로 조인을 설정하여 사용하거나 동일한 조건절을 사용하는 테이블을 뷰로 생성한다.

  예 〈주문〉 테이블과 〈거래처〉 테이블을 조인하여 사용하는 경우가 많다면 〈주문〉 테이블과 〈거래처〉 테이블에서 필요한 필드로 구성된 뷰를 생성한다.

- 동일한 테이블이라도 업무에 따라 테이블을 이용하는 부분이 달라질 수 있으므로 사용할 데이터를 다양한 관점에서 제시해야 한다.

  예 〈회원〉 테이블은 '회원번호', '이름', '주소', '전화번호', '총구매횟수', '총구매금액' 필드로 구성되어 있는데, 제품 발송 업무를 처리할 때는 '이름', '주소', '전화번호' 필드만 필요하므로 이 3개의 필드로 구성된 뷰를 설계한다.

- 데이터의 보안 유지를 고려하여 설계한다.

  예 〈회원〉 테이블의 '총구매횟수', '총구매금액' 등은 회사 차원에서 중요한 자료일 수 있으므로 발송 담당자가 볼 수 없도록 뷰를 설계한다.

따라잡기

기출문제 따라잡기 Section 089

이전기출

1. 데이터베이스 관리자가 기본 테이블에서 임의로 유도하여 만드는 테이블로서 사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위한 테이블을 무엇이라 하는가?

① 임시 테이블(Temporary Table)
② 뷰 테이블(View Table)
③ 색인 테이블(Index Table)
④ 기본 테이블(Base Table)

말이 필요없죠! 뷰(View), 잊으면 안 됩니다.

이전기출

2. SQL의 뷰(View)에 대한 장점으로 부적합한 것은?

① 논리적 데이터 독립성을 제공한다.
② 접근 제어를 통한 보안을 제공한다.
③ 나의 뷰를 제거해도 그 뷰를 기초로 정의된 다른 뷰는 제거되지 않는다.
④ 사용자의 데이터 관리를 간단하게 해준다.

하나의 뷰를 삭제하면 그 뷰를 기초로 정의된 다른 뷰도 자동으로 삭제됩니다.

이전기출

3. 데이터베이스의 뷰(View)에 관한 설명으로 옳지 않은 것은?

① 하나 이상의 테이블에서 유도되는 가상 테이블이다.
② 뷰는 별도의 디스크 공간을 차지하여 생성되는 실제적 테이블이다.
③ 뷰를 이용한 또 다른 뷰의 생성이 가능하다.
④ 삽입, 갱신, 삭제 연산에는 제약이 따른다.

뷰는 실제로 생성되는 물리적인 테이블이 아닌 가상 테이블입니다.

▶ 정답 : 1. ② 2. ③ 3. ②

SECTION 090 LIVE

클러스터 설계

1 클러스터(Cluster)의 개요

클러스터는 데이터 저장 시 데이터 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 방법이다.

- 클러스터링키*로 지정된 컬럼 값의 순서대로 저장되고, 여러 개의 테이블이 하나의 클러스터에 저장된다.

예 클러스터링 되지 않은 테이블

부서번호	부서
b001	기획부
b002	총무부
b003	영업부

〈부서〉 테이블

사원번호	이름	부서번호
1	홍길동	b001
2	장발장	b001
3	이예슬	b002
4	김강단	b003
5	정숙한	b002

〈사원〉 테이블

예 클러스터링된 테이블*

부서번호	부서
b001	기획부

사원번호	이름
1	홍길동
2	장발장

부서번호	부서
b002	총무부

사원번호	이름
3	이예슬
5	정숙한

부서번호	부서
b003	영업부

사원번호	이름
4	김강단

〈부서〉 테이블과 〈사원〉 테이블이 '부서번호' 필드를 기준으로 클러스터링 되었다. 이런 경우 '부서번호'를 클러스터링키라고 한다.

2 클러스터(Cluster)의 특징

- 클러스터링 된 테이블은 데이터 조회 속도는 향상시키지만 데이터 입력, 수정, 삭제에 대한 성능은 저하시킨다.
- 클러스터는 데이터의 분포도*가 넓을수록 유리하다.
- 데이터 분포도가 넓은 테이블을 클러스터링 하면 저장 공간을 절약할 수 있다.
- 클러스터링된 테이블은 클러스터링키 열을 공유하므로 저장 공간이 줄어든다.
- 대용량을 처리하는 트랜잭션은 전체 테이블을 스캔하는 일이 자주 발생하므로 클러스터링을 하지 않는 것이 좋다.

전문가의 조언

- 클러스터링은 비슷한 종류끼리 묶어준다는 의미로 테이블에서는 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 방법이고, 서버에서는 두 대 이상의 서버를 하나의 서버처럼 운영하기 위한 방법을 의미합니다. 이번 섹션에서는 테이블에서의 클러스터링에 대해 배웁니다. 클러스터의 개념과 특징에 대해 알아두세요.
- 서버에 대한 클러스터링은 Section 094에서 공부합니다.

클러스터링키

- 클러스터링된 테이블에서 각각의 행을 접근할 때 기준이 되는 열로, 데이터를 조회하면 클러스터링키로 지정된 필드에서 시작하여 클러스터링된 테이블의 데이터를 조회합니다.
- 클러스터링키인 '부서번호' 필드에서 조회를 시작하여 〈사원〉 테이블의 '사원번호' 필드 순으로 조회합니다.

분포도

인덱스는 분포도가 좁은 테이블이 좋지만 클러스터링은 분포도가 넓은 테이블에 유리합니다.

단일 테이블 클러스터링
단일 테이블 클러스터링은 여러 개의 테이블 뿐만 아니라 한 개의 테이블에 대해서도 클러스터링을 수행할 수 있습니다. 특정 컬럼의 동일한 값을 동일 블록이나 연속된 블록에 저장하므로 데이터 조회 성능이 향상됩니다.

파티셔닝
데이터베이스에서의 파티셔닝은 대용량의 테이블이나 인덱스를 작은 논리적 단위인 파티션으로 나누는 것입니다. Section 091에서 자세히 공부합니다.

ORDER BY, GROUP BY, UNION
ORDER BY는 정렬, GROUP BY는 그룹 지정, UNION은 통합 지정시 사용하는 SQL 명령어입니다.

- 처리 범위가 넓은 경우에는 단일 테이블 클러스터링*을, 조인이 많이 발생하는 경우에는 다중 테이블 클러스터링을 사용한다.
- 파티셔닝*된 테이블에는 클러스터링을 할 수 없다.
- 클러스터링을 하면 비슷한 데이터가 동일한 데이터 블록에 저장되기 때문에 디스크 I/O가 줄어든다.
- 클러스터링된 테이블에 클러스터드 인덱스를 생성하면 접근 성능이 향상된다.

3 클러스터 대상 테이블

- 분포도가 넓은 테이블
- 대량의 범위를 자주 조회하는 테이블
- 입력, 수정, 삭제가 자주 발생하지 않는 테이블
- 자주 조인되어 사용되는 테이블
- ORDER BY, GROUP BY, UNION*이 빈번한 테이블

기출문제 따라잡기 Section 090

출제예상
1. 다음 중 데이터베이스 테이블에서의 클러스터에 대한 설명으로 가장 옳지 않은 것은?
① 클러스터는 동일한 성격의 데이터를 물리적으로 동일한 데이터 블록에 저장하는 것이다.
② 클러스터는 데이터 조회 속도는 물론이고 데이터 입력, 수정, 삭제에 대한 성능도 향상시킨다.
③ 분포도가 넓은 테이블을 클러스터링 하면 저장 공간을 절약할 수 있다.
④ 파티셔닝된 테이블은 클러스터링을 할 수 없다.

클러스터는 데이터 조회 속도는 향상시켜주지만 데이터 입력, 수정, 삭제에 대한 성능은 저하시킵니다.

출제예상
2. 다음 중 클러스터링을 적용하기에 적당하지 않은 테이블은?
① 분포도가 좁은 테이블
② 수정이 거의 발생하지 않는 테이블
③ 자주 조인되어 사용되는 테이블
④ 대량의 범위를 자주 조회하는 테이블

클러스터링은 데이터의 분포도가 넓을수록 유리합니다.

▶ 정답 : 1. ② 2. ①

SECTION 091 ● LIVE

파티션 설계

1 파티션(Patition)의 개요

데이터베이스에서 파티션은 대용량의 테이블이나 인덱스를 작은 논리적 단위인 파티션으로 나누는 것을 말한다.

- 대용량 DB의 경우 중요한 몇 개의 테이블에만 집중되어 데이터가 증가되므로, 이런 테이블들을 작은 단위로 나눠 분산시키면 성능 저하를 방지할 뿐만 아니라 데이터 관리도 쉬워진다.
- 테이블이나 인덱스를 파티셔닝 하면 파티션키* 또는 인덱스키에 따라 물리적으로 별도의 공간에 데이터가 저장된다.
- 데이터 처리는 테이블 단위*로 이뤄지고, 데이터 저장은 파티션별로 수행된다.

2 파티션의 장 · 단점

장점	• 데이터 접근 시 액세스 범위를 줄여 쿼리 성능이 향상된다. • 파티션별로 데이터가 분산되어 저장되므로 디스크의 성능이 향상된다. • 파티션별로 백업 및 복구를 수행하므로 속도가 빠르다. • 시스템 장애 시 데이터 손상 정도를 최소화할 수 있다. • 데이터 가용성이 향상된다. • 파티션 단위로 입 · 출력을 분산시킬 수 있다.
단점	• 하나의 테이블을 세분화하여 관리하므로 세심한 관리가 요구된다. • 테이블간 조인에 대한 비용이 증가한다. • 용량이 작은 테이블에 파티셔닝을 수행하면 오히려 성능이 저하된다.

3 파티션의 종류

파티션의 종류는 파티셔닝 방식에 따라 범위 분할, 해시 분할, 조합 분할 등으로 나뉜다.

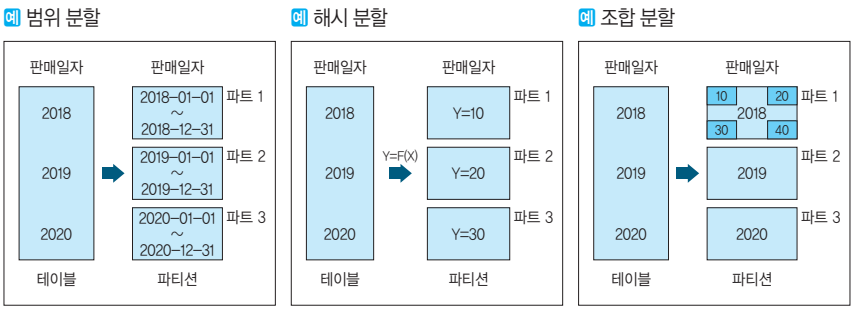
범위 분할 (Range Partitioning)	지정한 열의 값을 기준으로 분할한다. 예 일별, 월별, 분기별 등
해시 분할 (Hash Partitioning)	• 해시 함수를 적용한 결과 값에 따라 데이터를 분할한다. • 특정 파티션에 데이터가 집중되는 범위 분할의 단점을 보완한 것으로, 데이터를 고르게 분산할 때 유용하다. • 특정 데이터가 어디에 있는지 판단할 수 없다. • 고객번호, 주민번호 등과 같이 데이터가 고른 컬럼에 효과적이다.
조합 분할 (Composite Partitioning)	• 범위 분할로 분할한 다음 해시 함수를 적용하여 다시 분할하는 방식이다. • 범위 분할한 파티션이 너무 커서 관리가 어려울 때 유용하다.

전문가의 조언

파티션은 용량이 큰 테이블을 관리하기 쉽도록 작은 단위로 나눈 것을 말합니다. 파티션의 개요를 바탕으로 파티션의 장 · 단점, 종류, 인덱스 파티션의 종류 등을 파악하세요.

파티션키
파티션키는 파티션을 나누는 기준이 되는 열로, 파티션키로 지정된 열의 데이터 값에 따라 파티션이 나뉩니다.

테이블 단위로 처리
하나의 테이블이 여러 개의 파티션으로 나눠져 있어도 DB에 접근하는 애플리케이션은 테이블 단위로 데이터를 처리하기 때문에 파티션을 인식하지 못합니다.



- **범위 분할** : '판매일자' 필드를 기준으로 연도별로 파티션을 나눈다.
- **해시 분할** : 해시 함수(F(X))에 '판매일자' 필드를 인수로 입력하여 그 결과값에 따라 파티션을 나눈다.
- **조합 분할** : '판매일자' 필드를 기준으로 연도별로 파티션을 나눈 다음 하나의 파티션 안에서 다시 해시 함수를 적용하여 다시 파티션을 나눈다.

4 파티션키 선정 시 고려 사항

- 파티션키는 테이블 접근 유형에 따라 파티셔닝*이 이뤄지도록 선정한다.
- 데이터 관리의 용이성을 위해 이력성 데이터*는 파티션 생성주기와 소멸주기를 일치시켜야 한다.
- 매일 생성되는 날짜 컬럼, 백업의 기준이 되는 날짜 컬럼, 파티션 간 이동이 없는 컬럼, I/O 병목을 줄일 수 있는 데이터 분포가 양호한 컬럼 등을 파티션키로 선정한다.

접근 유형에 따른 파티셔닝
예를 들면, 분포도가 넓어 인덱스를 사용해도 조회 성능이 좋지 않은 테이블을 검색 범위와 파티션 단위를 동일하게 하여 파티셔닝을 수행하면 인덱스를 사용하지 않고도 데이터 조회 성능을 향상시킬 수 있습니다.

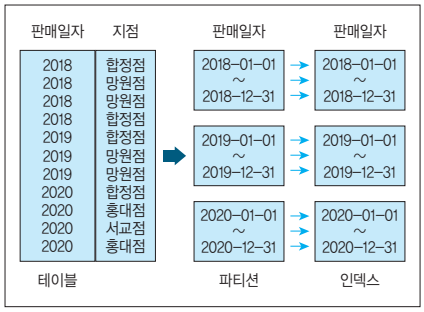
이력성 데이터
이력성 데이터는 수명이 다되어 데이터베이스에서는 삭제되었지만 데이터 관리 및 업무 규칙에 따라 별도의 저장장치에 보관되는 데이터로, 활용 가치에 따라 생성주기와 소멸주기가 결정됩니다. 생성주기와 소멸주기에 맞춰 데이터베이스를 정리해야 데이터 관리가 쉬워집니다.

5 인덱스 파티션

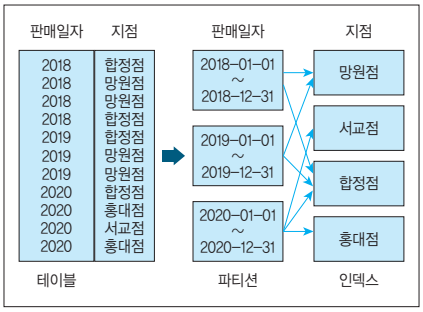
인덱스 파티션은 파티션된 테이블의 데이터를 관리하기 위해 인덱스를 나눈 것이다.

- 인덱스 파티션은 파티션된 테이블의 종속 여부에 따라 Local Partitioned Index와 Global Partitioned Index로 나뉜다.
  - Local Partitioned Index : 테이블 파티션과 인덱스 파티션이 1:1 대응되도록 파티셔닝한다.
  - Global Partitioned Index : 테이블 파티션과 인덱스 파티션이 독립적으로 구성되도록 파티셔닝한다.
  - Local Partitioned Index가 Global Partitioned Index에 비해 데이터 관리가 쉽다.

예 Local Partitioned Index



예 Global Partitioned Index



- Local Partitioned Index : 파티션과 인덱스를 모두 '판매일자' 필드를 기준으로 수행한다.
- Global Partitioned Index : 파티션은 '판매일자' 필드를 기준으로 수행하고, 인덱스는 '지점' 필드를 기준으로 수행한다.

- 인덱스 파티션은 인덱스 파티션키 컬럼의 위치에 따라 Prefixed Partitioned Index와 Non-prefixed Partitioned Index로 나뉜다.
  - Prefixed Partitioned Index : 인덱스 파티션키와 인덱스 첫 번째 컬럼이 같다.
  - Non-Prefixed Partitioned Index : 인덱스 파티션키와 인덱스 첫 번째 컬럼이 다르다.

예

파티션 인덱스 유형	인덱스 파티션키	인덱스키 컬럼
Prefixed Partitioned Index	**판매일자**	**판매일자**(+품목코드)
Non-Prefixed Partitioned Index	**판매일자**	**품목코드**(+거래일자)

- Local과 Global, Prefixed와 Nonprefixed를 조합하여 Local Prefixed Partitioned Index, Local Non-Prefixed Partitioned Index, Global Prefixed Partitioned Index 등으로 구성하여 사용한다.

전문가의 조언

Global Non-Prefixed Partitioned Index는 허용되지 않습니다.

기출문제 따라잡기

Section 091

출제예상

1. 다음 중 인덱스 파티션에 대한 설명으로 옳지 않은 것은?

① 테이블 파티션과 인덱스 파티션이 1:1 대응되는 파티셔닝은 Local Partitioned Index이다.

② 테이블 파티션과 인덱스 파티션이 독립적으로 구성된 파티셔닝은 Global Partitioned Index이다.

③ Local Partitioned Index보다 Global Partitioned Index가 데이터 관리가 수월하다.

④ 인덱스 파티션키와 인덱스 첫 번째 컬럼이 같은 파티셔닝은 Prefixed Partitioned Index이다.

테이블 파티션과 인덱스 파티션의 범위가 동일한 것이 관리하기 쉬울까요? 서로 다른 것이 쉬울까요?

출제예상

2. 다음 중 파티션에 대한 설명으로 틀린 것은?

① 파티션은 하나의 테이블을 작은 논리적 단위로 나눈 것이다.

② 파티셔닝된 테이블은 물리적으로 별도의 세그먼트에 저장된다.

③ 테이블 크기에 상관없이 파티셔닝을 수행하면 성능이 향상된다.

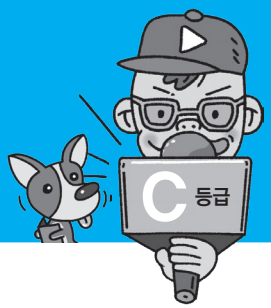
④ 파티셔닝을 수행하면 데이터 가용성이 향상된다.

대용량의 테이블을 파티셔닝 하면 성능이 향상되지만 저용량 테이블의 경우 오히려 성능이 저하됩니다.

▶ 정답 : 1. ③ 2. ③

SECTION 092 ● LIVE

데이터베이스 용량 설계



전문가의 조언

- 데이터베이스의 용량 설계는 데이터베이스 설계만큼이나 중요합니다. 용량이 필요할 때마다 용량을 증설한다면 데이터베이스의 전체적인 성능이 저하되고 비용이 계속 증가할 수 있습니다. 데이터베이스 용량 분석의 목적과 분석 절차에 대해 알아두세요.
- 테이블, 테이블스페이스, 인덱스 등의 용량과 점차 증가될 데이터 양을 파악해서 데이터베이스가 저장될 디스크나 스토리지의 용량 크기를 선택해야 합니다.

익스텐트(범위, Extent)
익스텐트는 기본적인 용량이 모두 찼을 경우 추가적으로 할당되는 공간을 의미합니다.

1 데이터베이스 용량 설계

데이터베이스 용량 설계는 데이터가 저장될 공간을 정의하는 것이다.

- 데이터베이스 용량을 설계할 때는 테이블에 저장할 데이터양과 인덱스, 클러스터 등이 차지하는 공간 등을 예측하여 반영해야 한다.

2 데이터베이스 용량 설계의 목적

- 데이터베이스의 용량을 정확히 산정하여 디스크의 저장 공간을 효과적으로 사용하고 확장성 및 가용성을 높인다.
- 디스크의 특성을 고려하여 설계함으로써 디스크의 입·출력 부하를 분산시키고 채널의 병목 현상을 최소화한다.
- 디스크에 대한 입·출력 경합이 최소화되도록 설계함으로써 데이터 접근성이 향상된다.
- 데이터 접근성을 향상시키는 설계 방법
  - 테이블의 테이블스페이스와 인덱스의 테이블스페이스를 분리하여 구성한다.
  - 테이블스페이스와 임시 테이블스페이스를 분리하여 구성한다.
  - 테이블을 마스터 테이블과 트랜잭션 테이블로 분류한다.
- 데이터베이스에 생성되는 오브젝트의 익스텐트* 발생을 최소화하여 성능을 향상시킨다.
- 데이터베이스 용량을 정확히 분석하여 테이블과 인덱스에 적합한 저장 옵션을 지정한다.

3 데이터베이스 용량 분석 절차

❶ 데이터 예상 건수, 로우(Row) 길이, 보존 기간, 증가율 등 기초 자료를 수집하여 용량을 분석한다.

예 기초 자료 수집 예시

NO	엔티티	테이블	Row 길이	보존 기간	초기건수	주기별 발생건수	발생 주기	년증가율(건)
001	주문	ODDER	1,000	2	19,250	80	일	500
002	상품	ITEM	430	2	34,502	200	일	300
003	판매자	SELLER	310	2	12,520	15	일	150

❷ 분석된 자료를 바탕으로 DBMS에 이용될 테이블, 인덱스 등 오브젝트별 용량을 산정한다.

❸ 테이블과 인덱스의 테이블스페이스 용량을 산정한다.

– 테이블스페이스 용량은 테이블스페이스에 생성되는 테이블 용량을 모두 더한 값에 약 40%* 정도를 추가하여 산정한다.

예 테이블 용량이 100M라면 테이블스페이스의 용량은 100 + (100 × 40%) = 140M 정도로 산정한다.

❹ 데이터베이스에 저장될 모든 데이터 용량과 데이터베이스 설치 및 관리를 위한 시스템 용량을 합해 디스크 용량을 산정한다.

예 디스크 용량 계산

NO	이름	용량(G)	사용용량(M)	사용비율	파일명	파일 용량
1	디스크1	3	2,700	90%	AB001.ABC01	1,700M
					AB002.ABC01	1,000M
2	디스크2	5	2,500	50%	AB001.ABC02	1,500M
					AB002.ABC02	1,000M

- **디스크1** : 디스크에 저장될 전체 파일의 용량이 2,700M이고 추후에 추가될 데이터가 많지 않아 300M의 여유 공간이 있는 3G 디스크를 선택하였다.
- **디스크2** : 디스크에 저장될 전체 파일의 용량이 2,500M이나 추후에 추가될 데이터 양이 많아 2,500M의 여유 공간이 있는 5G 디스크를 선택하였다.

약 40%
약 40%라는 값은 절대적인 값이 아닙니다. 업무에 따라 확장 가능성이 높은 것은 40% 정도, 확장 가능성이 낮은 것은 20% 정도로 지정하면 됩니다.

따라잡기

기출문제 따라잡기

Section 092

출제예상

1. 다음은 데이터베이스 용량 설계에 대한 내용이다. 가장 옳지 않은 것은?

① 데이터베이스 용량을 분석하려면 먼저 데이터 예상 건수, 로우 길이, 보존 기간, 증가율 등 기초 자료를 수집하여 분석해야 한다.

② 용량 설계를 정확히 산정하면 데이터의 확장성 및 가용성을 높일 수 있다.

③ 입 · 출력 경합을 최소화하여 데이터 접근성을 높이려면 테이블과 인덱스의 테이블스페이스를 같은 곳에 위치하도록 구성한다.

④ 데이터의 특성을 고려한 용량 설계를 통해 디스크 채널의 병목 현상을 최소화할 수 있다.

입 · 출력 경합을 최소화하려면 테이블을 분리해서 구성해야 합니다. 인덱스도 마찬가지입니다.

출제예상

2. 다음 중 데이터베이스 용량 설계에 대한 설명으로 틀린 것은?

① 데이터 용량 분석은 물리 데이터베이스 설계 과정에서 수행한다.

② 데이터 용량을 정확히 분석하면 디스크 공간을 효율적으로 사용할 수 있다.

③ 디스크 입 · 출력 경합을 최소화시켜 데이터 접근성이 향상되도록 설계해야 한다.

④ 데이터베이스에 생성되는 오브젝트의 익스텐트 발생이 최대화가 되도록 설계한다.

익스텐트는 기본적인 용량이 모두 찼을 경우 추가적으로 할당되는 공간입니다. 익스텐트가 자주 발생하는 게 좋을까요?

▶ 정답 : 1. ③ 2. ④

SECTION 093 ● LIVE

분산 데이터베이스 설계

전문가의 조언

분산 데이터베이스는 상식적인 내용들로 구성되었습니다. 가볍게 읽어보세요.

1 분산 데이터베이스 정의

분산 데이터베이스는 논리적으로는 하나의 시스템에 속하지만 물리적으로는 네트워크를 통해 연결된 여러 개의 컴퓨터 사이트(Site)에 분산되어 있는 데이터베이스를 말한다.

- 분산 데이터베이스는 데이터의 처리나 이용이 많은 지역에 데이터베이스를 위치시킴으로써 데이터의 처리가 가능한 해당 지역에서 해결될 수 있도록 한다.

2 분산 데이터베이스의 구성 요소

분산 처리기	자체적으로 처리 능력을 가지며, 지리적으로 분산되어 있는 컴퓨터 시스템을 말한다.
분산 데이터베이스	지리적으로 분산되어 있는 데이터베이스로서 해당 지역의 특성에 맞게 데이터베이스가 구성된다.
통신 네트워크	분산 처리기들을 통신망으로 연결하여 논리적으로 하나의 시스템처럼 작동할 수 있도록 하는 통신 네트워크를 말한다.

3 분산 데이터베이스 설계 시 고려 사항

- 작업부하(Work Load)의 노드별 분산 정책
- 지역의 자치성 보장 정책
- 데이터의 일관성 정책
- 사이트나 회선의 고장으로부터의 회복 기능
- 통신 네트워크를 통한 원격 접근 기능

전문가의 조언

분산 데이터베이스의 4대 목표를 영어 영문과 같이 정확하게 숙지하세요.

투명성(Transparency)
투명성이란 어떠한 사실이 존재함에도 마치 투명하여 보이지 않는 것처럼, 사실의 존재 여부를 염두에 두지 않아도 되는 성질을 말합니다.

4 분산 데이터베이스의 목표

- **위치 투명성***(Location Transparency) : 액세스하려는 데이터베이스의 실제 위치를 알 필요 없이 단지 데이터베이스의 논리적인 명칭만으로 액세스할 수 있다.
- **중복 투명성**(Replication Transparency) : 동일 데이터가 여러 곳에 중복되어 있더라도 사용자는 마치 하나의 데이터만 존재하는 것처럼 사용하고, 시스템은 자동으로 여러 자료에 대한 작업을 수행한다.
- **병행 투명성**(Concurrency Transparency) : 분산 데이터베이스와 관련된 다수의 트랜잭션들이 동시에 실현되더라도 그 트랜잭션의 결과는 영향을 받지 않는다.

- **장애 투명성(Failure Transparency)** : 트랜잭션, DBMS, 네트워크, 컴퓨터 장애에도 불구하고 트랜잭션을 정확하게 처리한다.

5 분산 데이터베이스의 장 · 단점

장점	단점
• 지역 자치성이 높다. • 자료의 공유성이 향상된다. • 분산 제어가 가능하다. • 시스템 성능이 향상된다. • 중앙 컴퓨터의 장애가 전체 시스템에 영향을 끼치지 않는다. • 효용성과 융통성이 높다. • 신뢰성 및 가용성이 높다. • 점진적 시스템 용량 확장이 용이하다.	• DBMS가 수행할 기능이 복잡하다. • 데이터베이스 설계가 어렵다. • 소프트웨어 개발 비용이 증가한다. • 처리 비용이 증가한다. • 잠재적 오류가 증가한다.

전문가의 조언

분산 데이터베이스의 정의와 목표를 숙지했다면 장점과 단점은 어렵지 않게 이해할 수 있습니다. 분산 데이터베이스의 장점이 아닌 것을 찾을 수 있을 정도로 알아두세요.

6 분산 데이터베이스 설계

분산 데이터베이스 설계는 애플리케이션이나 사용자가 분산되어 저장된 데이터에 접근하게 하는 것을 목적으로 한다.

- 잘못 설계된 분산 데이터베이스는 복잡성 증가, 응답 속도 저하, 비용 증가 등의 문제가 발생한다.
- 분산 데이터베이스의 설계는 전역 관계망을 논리적 측면에서 소규모 단위로 분할한 후, 분할된 결과를 복수의 노드에 할당하는 과정으로 진행된다. 노드에 할당된 소규모 단위를 분할(Fragment)이라 부른다.
- 분산 설계 방법에는 테이블 위치 분산, 분할(Fragmentation), 할당(Allocation)이 있다.

전문가의 조언

분산 데이터베이스의 설계 방법의 종류와 각각의 특징에 대해 숙지해 두세요.

7 테이블 위치 분산

테이블 위치 분산은 데이터베이스의 테이블을 각기 다른 서버에 분산시켜 배치하는 방법을 의미한다.

- 테이블 위치를 분산할 때는 테이블의 구조를 변경하지 않으며, 다른 데이터베이스의 테이블과 중복되지 않게 배치한다.
- 데이터베이스의 테이블을 각각 다른 위치에 배치하려면 해당 테이블들이 놓일 서버들을 미리 설정해야 한다.

예 테이블 위치 분산 예

위치 \ 테이블	사원	부서	상품	판매현황	거래업체
본사	0	0			0
지사			0	0	

8 분할(Fragment)

분할은 테이블의 데이터를 분할하여 분산시키는 것이다.

- **분할 규칙**
  - 완전성(Completeness) : 전체 데이터를 대상으로 분할해야 한다.
  - 재구성(Reconstruction) : 분할된 데이터는 관계 연산을 활용하여 본래의 데이터로 재구성할 수 있어야 한다.
  - 상호 중첩 배제(Dis-jointness) : 분할된 데이터는 서로 다른 분할의 항목에 속하지 않아야 한다.

    예 수평 분할한 경우 각각의 분할에 포함된 튜플들이 상호 중복되지 않아야 하고, 수직 분할한 경우는 각각의 분할에 포함된 속성들이 중복되지 않아야 한다.
- 주요 분할 방법
  - 수평 분할 : 특정 속성의 값을 기준으로 행(Row) 단위로 분할
  - 수직 분할 : 데이터 컬럼(속성) 단위로 분할

예 수평 분할

사원번호	사원이름	부서	입사일자	나이
S001	홍길동	영업부	2018.05.09	30
S002	장길산	영업부	2017.08.05	28
S003	김말자	영업부	2019.11.03	35
S004	이순자	총무부	2019.05.12	27
S005	양숙경	총무부	2016.02.24	28
S006	고진성	총무부	2017.09.12	30
S007	최진현	총무부	2019.11.25	29

예 수직 분할

사원번호	사원이름	부서	입사일자	나이
S001	홍길동	영업부	2018.05.09	30
S002	장길산	영업부	2017.08.05	28
S003	김말자	영업부	2019.11.03	35
S004	이순자	총무부	2019.05.12	27
S005	양숙경	총무부	2016.02.24	28
S006	고진성	총무부	2017.09.12	30
S007	최진현	총무부	2019.11.25	29

- **수평 분할** : '부서' 필드가 '영업부'인 테이블과 '총무부'인 테이블로 분할한다.
- **수직 분할** : '사원번호', '사원이름' 필드로 구성된 테이블과 '부서', '입사일자', '나이' 필드로 구성된 테이블로 분할한다.

9 할당(Allocation)

할당은 동일한 분할을 여러 개의 서버에 생성하는 분산 방법으로, 중복이 없는 할당(Allocation)과 중복이 있는 할당(Allocation)으로 나뉜다.

- 비중복 할당 방식
  - 최적의 노드를 선택해서 분산 데이터베이스의 단일 노드에서만 분할이 존재하도록 하는 방식이다.
  - 일반적으로 애플리케이션에는 릴레이션을 배타적 분할로 분리하기 힘든 요구가 포함되므로 분할된 테이블 간의 의존성은 무시되고 비용 증가, 성능 저하 등의 문제가 발생할 수 있다.

- **중복 할당 방식** : 동일한 테이블을 다른 서버에 복제하는 방식으로, 일부만 복제하는 부분 복제와 전체를 복제하는 완전 복제가 있다.

따라잡기

기출문제 따라잡기

Section 093

이전기출

1. 분산 데이터베이스의 장점으로 거리가 먼 것은?

① 지역 자치성이 높다.

② 효용성과 융통성이 높다.

③ 점진적 시스템 용량 확장이 용이하다.

④ 소프트웨어 개발 비용이 저렴하다.

분산 데이터베이스의 장점과 단점은 꼭 숙지해야 합니다. 객관식의 특성상 상대적으로 간단한 단점만 외우는 것이 문제를 푸는 데 편리할 수도 있습니다.

이전기출

2. 분산 DBMS의 4대 목표에 대한 설명으로 거리가 먼 것은?

① 위치 투명성(Location Transparency) : 트랜잭션은 특정 데이터 항목의 위치에 의존적이다.

② 중복 투명성(Replication Transparency) : 트랜잭션이 데이터의 중복 개수나 중복 사실을 모르고도 데이터 처리가 가능하다.

③ 병행 투명성(Concurrency Transparency) : 분산 데이터베이스와 관련된 다수의 트랜잭션들이 동시에 실현되더라도 그 트랜잭션의 결과는 영향을 안 받는다.

④ 장애 투명성(Failure Transparency) : 트랜잭션, DBMS, 네트워크, 컴퓨터 장애에도 불구하고 트랜잭션을 정확하게 처리한다.

위치 투명성은 어느 위치에 분산되어 있는가에 관계없이 사용할 수 있는 특성으로, 의존적이 아닌 독립적임 말합니다.

이전기출

3. 분산 데이터베이스 시스템의 장점으로 거리가 먼 것은?

① 사이트 간의 데이터들이 서로 오류가 발생할 가능성이 줄어든다.

② 데이터베이스 구축 이후에 점차적으로 새로운 사이트를 추가할 수 있다.

③ 특정한 사이트에서 장애가 발생하더라도 다른 사이트는 계속 운용할 수 있다.

④ 해당 지역에 필요한 데이터를 지역적으로 관리할 수 있다.

분산 데이터베이스 시스템은 중앙 집중식에 비해 사이트 간의 오류 발생률이 높은 단점이 있습니다. 그러나 일부 사이트에 장애가 발생하더라도 그 기능이 계속 수행될 수 있으므로, 전체가 가동하든지 아니면 전체가 고장나는 중앙 집중식에 비해 신뢰도가 높습니다.

이전기출

4. 분산 데이터베이스 설계 시 고려 사항으로 옳지 않은 것은?

① 작업부하(Work Load)의 노드별 분산 정책

② 지역의 자치성 보장 정책

③ 데이터의 일관성 정책

④ 분산 노드간 데이터의 중복성 보장과 가용성 감소

중복성 보장이 아니라, 동일 데이터가 여러 곳에 중복되어 있더라도 사용자는 마치 하나의 데이터만 존재하는 것처럼 사용하고, 시스템은 자동으로 여러 자료에 대한 작업을 수행할 수 있는 중복 투명성(Replication Transparency)이 제공되어야 하며, 가용성은 높여야 합니다.

이전기출

5. 분산 데이터베이스 시스템이 사용자에게 제공하는 네 가지 유형의 투명성(Transparency)에 속하지 않는 것은?

① 위치 투명성 ② 복제 투명성

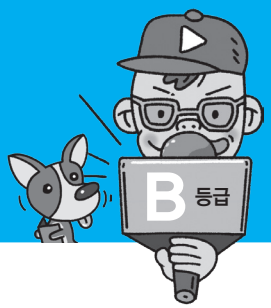
③ 수행 투명성 ④ 병행 투명성

네 가지 투명성 중에서 장애 투명성이 빠졌죠? 네 가지 투명성의 종류와 각각의 의미도 정확하게 알아두세요.

▶ 정답 : 1. ④ 2. ① 3. ① 4. ④ 5. ③

SECTION 094 ● LIVE

데이터베이스 이중화 / 서버 클러스터링



전문가의 조언

시스템의 예기치 못한 오류로 인한 데이터베이스의 서비스 중단을 막기 위한 방법으로 데이터베이스 이중화와 서버 클러스터링이 이용됩니다. 두 방법의 개념과 특징을 구분하여 알아두세요.

1 데이터베이스 이중화(Database Replication)

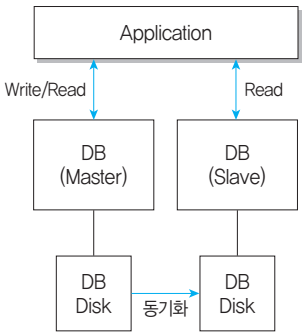
데이터베이스 이중화는 시스템 오류로 인한 데이터베이스 서비스 중단이나 물리적 손상 발생 시 이를 복구하기 위해 동일한 데이터베이스를 복제하여 관리하는 것이다.

- 데이터베이스 이중화를 수행하면 하나 이상의 데이터베이스가 항상 같은 상태를 유지하므로 데이터베이스에 문제가 발생하면 복제된 데이터베이스를 이용하여 즉시 문제를 해결할 수 있다.
- 데이터베이스 이중화는 여러 개의 데이터베이스를 동시에 관리하므로 사용자가 수행하는 작업이 데이터베이스 이중화 시스템에 연결된 다른 데이터베이스에도 동일하게 적용된다.
- 데이터베이스 이중화는 애플리케이션을 여러 개의 데이터베이스로 분산시켜 처리하므로 데이터베이스의 부하를 줄일 수 있다.
- 데이터베이스 이중화를 이용하면 손쉽게 백업 서버를 운영할 수 있다.

이중화 방식

데이터베이스 이중화 방식은 DBMS마다 여러 종류가 제공됩니다.

예 MySQL의 Replication 이중화 방식*



- 동일한 데이터베이스를 복제하여 관리하되, 데이터를 읽고 쓸 수 있는 마스터 DB와 읽기만 가능한 슬레이브 DB로 구분하여 사용한다.
- 슬레이브 DB는 마스터 DB의 로그를 이용하여 변경된 데이터를 동기화한다.

2 데이터베이스 이중화의 분류

데이터베이스 이중화는 변경 내용의 전달 방식에 따라 Eager 기법과 Lazy 기법으로 나뉜다.

Eager 기법	트랜잭션 수행 중 데이터 변경이 발생하면 이중화된 모든 데이터베이스에 즉시 전달하여 변경 내용이 즉시 적용되도록 하는 기법

Lazy 기법	트랜잭션의 수행이 종료되면 변경 사실을 새로운 트랜잭션에 작성하여 각 데이터베이스에 전달되는 기법으로, 데이터베이스마다 새로운 트랜잭션이 수행되는 것으로 간주된다.

3 데이터베이스 이중화 구성 방법

데이터베이스의 이중화 구성 방법에는 활동-대기(Active-Standby) 방법과 활동-활동(Active-Active) 방법이 있다.

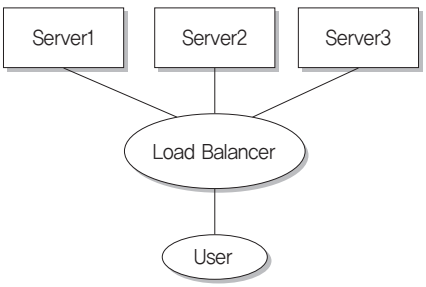
활동-대기(Active-Standby) 방법	• 한 DB가 활성 상태로 서비스하고 있으면 다른 DB는 대기하고 있다가 활성 DB에 장애가 발생하면 대기 상태에 있던 DB가 자동으로 모든 서비스를 대신 수행한다. • 구성 방법과 관리가 쉬워 많은 기업에서 이용된다.
활동-활동(Active-Active) 방법	• 두 개의 DB가 서로 다른 서비스를 제공하다가 둘 중 한쪽 DB에 문제가 발생하면 나머지 다른 DB가 서비스를 제공한다. • 두 DB가 모두 처리를 하기 때문에 처리율이 높지만 구성 방법 및 설정이 복잡하다.

4 클러스터링(Clustering)

클러스터링은 두 대 이상의 서버를 하나의 서버처럼 운영하는 기술이다.

- 클러스터링은 서버 이중화 및 공유 스토리지*를 사용하여 서버의 고가용성*을 제공한다.
- 클러스터링에는 고가용성 클러스터링과 병렬 처리 클러스터링이 있다.
  - 고가용성 클러스터링 : 하나의 서버에 장애가 발생하면 다른 노드(서버)가 받아 처리하여 서비스 중단을 방지하는 방식으로, 일반적으로 언급되는 클러스터링이 고가용성 클러스터링이다.
  - 병렬 처리 클러스터링 : 전체 처리율을 높이기 위해 하나의 작업을 여러 개의 서버에서 분산하여 처리하는 방식이다.

예 병렬 처리 클러스터링



사용자의 요청을 로드 밸런서(Load Balancer)*가 여러 대의 서버로 분산한다.

공유 스토리지(NAS; Network Attached Storage)
공유 스토리지는 데이터 저장소를 네트워크로 연결하여 파일 및 데이터를 공유하는 것으로, 다수의 사용자 또는 서버가 데이터를 안전하고 편리하게 공유할 수 있습니다.

고가용성(HA, High Availability)
고가용성은 시스템을 오랜 시간동안 계속해서 정상적으로 운영이 가능한 성질을 의미합니다.

로드 밸런서(Load Balancer)
로드 밸런서는 특정 서버에 집중되는 부하를 덜기 위해 여러 개의 서버로 부하를 분산시키는 네트워크 서비스입니다.

출제예상

1. 다음 중 데이터베이스 이중화에 대한 설명으로 틀린 것은?

① 데이터베이스 이중화는 동일한 데이터베이스를 복제하여 관리한다.

② 데이터베이스 이중화가 시행 중일 때 사용자가 하나의 데이터베이스를 수정하면 복제된 데이터베이스에도 동일하게 적용된다.

③ 데이터베이스 이중화가 시행 중일 때 데이터베이스에 문제가 발생하면 복제된 데이터베이스를 이용하여 서비스를 계속해서 제공할 수 있다.

④ 데이터 변경이 발생하면 즉시 이중화된 데이터베이스에 전달되어 변경 내용이 즉시 적용되는 기법은 Lazy 기법입니다.

변경 내용을 미루지 않고 바로 적용하는 기법인데, 게으른 Lazy 기법일까요?

출제예상

2. 다음 중 시스템의 예기치 못한 오류로 인한 데이터베이스의 서비스 중단을 막기 위한 데이터베이스 이중화에 대한 설명으로 옳지 않은 것은?

① 데이터베이스 이중화는 변경 내용 전달 방식에 따라 Eager 기법과 Lazy 기법으로 나뉜다.

② Eager 기법은 변경 내용이 발생하면 즉시 다른 데이터베이스에도 적용한다.

③ 활동-대기(Active-Standby) 방법은 활성 데이터베이스에 문제가 발생하면 대기 중이던 데이터베이스가 대신 서비스를 제공한다.

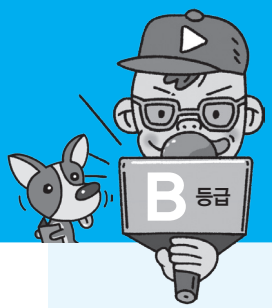
④ 활동-대기(Active-Standby) 방법은 활동-활동(Active-Active) 방법에 비해 구성 방법 및 설정이 복잡하다.

두 DB가 모두 처리를 하기 때문에 처리율은 높지만 구성 방법 및 설정이 복잡한 것은 활동-활동(Active-Active) 방법입니다.

▶ 정답 : 1. ④ 2. ④

SECTION 095 ● LIVE

데이터베이스 보안 / 암호화



1 데이터베이스 보안의 개요

데이터베이스 보안이란 데이터베이스의 일부분 또는 전체에 대해서 권한이 없는 사용자가 액세스하는 것을 금지하기 위해 사용되는 기술이다.

- 보안을 위한 데이터 단위는 테이블 전체로부터 특정 테이블의 특정한 행과 열 위치에 있는 특정한 데이터 값에 이르기까지 다양하다.
- 데이터베이스 사용자들은 일반적으로 서로 다른 객체에 대하여 다른 접근 권리 또는 권한을 갖게 된다.

2 암호화(Encryption)

암호화는 데이터를 보낼 때 송신자가 지정한 수신자 이외에는 그 내용을 알 수 없도록 평문을 암호문으로 변환하는 것이다.

- **암호화(Encryption) 과정** : 암호화되지 않은 평문을 정보 보호를 위해 암호문으로 바꾸는 과정
- **복호화(Decryption) 과정** : 암호문을 원래의 평문으로 바꾸는 과정
- 암호화 기법에는 개인키 암호 방식과 공개키 암호 방식이 있다.

3 개인키 암호 방식(Private Key Encryption) = 비밀키 암호 방식

비밀키 암호화 기법은 동일한 키로 데이터를 암호화하고 복호화한다.

- 비밀키 암호화 기법은 대칭 암호 방식 또는 단일키 암호화 기법이라고도 한다.
- 비밀키는 제3자에게는 노출시키지 않고 데이터베이스 사용 권한이 있는 사용자만 나누어 가진다.
- **종류** : 전위 기법, 대체 기법, 대수 기법, 합성 기법(DES*, LUCIFER)

4 공개키 암호 방식(Public Key Encryption)

공개키 암호화 기법은 서로 다른 키로 데이터를 암호화하고 복호화한다.

- 데이터를 암호화할 때 사용하는 키(공개키, Public Key)는 데이터베이스 사용자에게 공개하고, 복호화할 때의 키(비밀키, Secret Key)는 관리자가 비밀리에 관리하는 방법이다.
- 공개키 암호화 기법은 비대칭 암호 방식이라고도 하며, 대표적으로 RSA(Rivest Shamir Adleman)가 있다.

전문가의 조언

데이터베이스 보안의 의미, 그리고 보안과 무결성의 차이점을 명확히 구분해 두세요.

무결성(Integrity)과 보안(Security)

- 무결성은 권한이 있는 사용자로부터 데이터베이스를 보호하는 것이고, 보안은 권한이 없는 사용자로부터 데이터베이스를 보호하는 것입니다.
- 보안은 데이터베이스 사용자들이 데이터베이스를 사용하고자 할 때 언제든지 사용할 수 있도록 보장하는 것이고, 무결성은 정확하게 사용할 수 있도록 보장하는 것입니다.

전문가의 조언

개인키 암호 방식과 공개키 암호 방식을 확실하게 구분할 수 있어야 합니다.

DES 기법

DES 기법은 개인키 암호 방식의 대표적인 알고리즘으로, 64Bit의 평문 블록을 56Bit의 16개 키를 이용하여 16회의 암호 계산 단계를 거쳐 64Bit의 암호문을 얻습니다.

이전기출

1. 암호화 기법 중 암호화 알고리즘과 암호화키는 공개해서 누구든지 평문을 암호문으로 만들 수 있지만, 해독 알고리즘과 해독키는 비밀로 유지하는 기법을 무엇이라 하는가?

① DES(Data Encryption Standard) 기법

② 공중키(Public Key) 암호화 기법

③ 대체(Substitution) 암호화 기법

④ 전치(Transposed) 암호화 기법

공중키(Public Key) 암호화 기법과 개인키(Private Key) 암호화 기법은 명확하게 구분할 수 있어야 합니다. 암호화와 복호화에 한 개의 키를 사용하는 것은 개인키 암호화 기법이고, 암호화와 복호화에 서로 다른 두 개의 키를 사용하는 것은 공중키 암호화 기법입니다. 그리고 DES가 대표적인 개인키 암호화 기법이라는 것도 알아두세요.

이전기출

2. 데이터베이스 보안에 대한 설명으로 옳지 않은 것은?

① 보안을 위한 데이터 단위는 테이블 전체로부터 특정 테이블의 특정한 행과 열 위치에 있는 특정한 데이터 값에 이르기까지 다양하다.

② 각 사용자들은 일반적으로 서로 다른 객체에 대하여 다른 접근 권리 또는 권한을 갖게 된다.

③ 불법적인 데이터의 접근으로부터 데이터베이스를 보호하는 것이다.

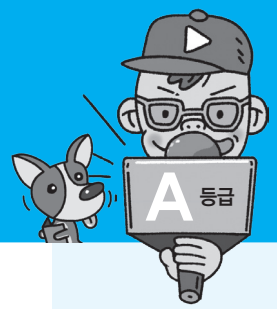
④ 보안을 위한 사용자들의 권한 부여는 관리자의 정책 결정보다는 DBMS가 자체 결정하여 제공한다.

데이터베이스 보안을 위한 사용자들의 권한 부여는 DBMS라는 소프트웨어 시스템이 자체 결정하는 것이 아니라, 사람인 관리자의 정책 결정에 의해서 정해집니다.

▶ 정답 : 1. ② 2. ④

SECTION 096 ● LIVE

데이터베이스 보안 – 접근통제



1 접근통제

접근통제는 데이터가 저장된 객체*와 이를 사용하려는 주체* 사이의 정보 흐름을 제한하는 것이다.

- 접근통제는 데이터에 대해 다음과 같은 통제를 함으로써 자원의 불법적인 접근 및 파괴를 예방한다.
  - 비인가된 사용자의 접근 감시
  - 접근 요구자의 사용자 식별
  - 접근 요구의 정당성 확인 및 기록
  - 보안 정책에 근거한 접근의 승인 및 거부 등
- 접근통제 기술에는 임의 접근통제(DAC), 강제 접근통제(MAC)가 있다.

임의 접근통제(DAC, Discretionary Access Control)	• 임의 접근통제는 데이터에 접근하는 사용자의 신원에 따라 접근 권한을 부여하는 방식이다. • 통제 권한이 주체에 있어 주체가 접근통제 권한을 지정하고 제어할 수 있다. • 일반적으로 특정 객체에 대한 조작 권한은 데이터베이스 관리 시스템으로부터 부여받지만 임의 접근통제에서는 객체를 생성한 사용자가 생성된 객체에 대한 모든 권한을 부여받고, 부여된 권한을 다른 사용자에게 허가할 수도 있다. • 임의 접근통제에 사용되는 SQL 명령어에는 GRANT*와 REVOKE*가 있다.
강제 접근통제(MAC, Mandatory Access Control)	• 강제 접근통제는 주체와 객체의 등급을 비교하여 접근 권한을 부여하는 방식이다. • 제3자가 접근통제 권한을 지정한다. • 데이터베이스 객체별로 보안 등급을 부여할 수 있고, 사용자별로 인가 등급을 부여할 수 있다. • 주체는 자신보다 보안 등급이 높은 객체에 대해 읽기, 수정, 등록이 모두 불가능하고, 보안 등급이 같은 객체에 대해서는 읽기, 수정, 등록이 가능하고, 보안 등급이 낮은 객체는 읽기가 가능하다.

- 접근통제의 3요소는 접근통제 정책, 접근통제 메커니즘, 접근통제 보안모델이다.

2 접근통제 정책

접근통제 정책은 어떤 주체가(Who)가 언제(When), 어디서(Where), 어떤 객체(What)에게, 어떤 행위(How)에 대한 허용 여부를 정의하는 것으로, 신분 기반 정책, 규칙 기반 정책, 역할 기반 정책이 있다.

전문가의 조언

접근통제는 말 그대로 데이터베이스에 대한 사용자들의 접근을 통제함으로써 데이터를 보호하는 방법입니다. 접근통제의 개념을 숙지하고 관련 기술의 종류와 개별적인 특징을 잘 정리하세요.

- **객체** : 테이블, 컬럼 등과 같은 데이터베이스 개체들을 의미합니다.
- **주체** : 일반적으로 객체를 사용하기 위해 접근을 시도하는 사용자를 의미합니다.

- GRANT : 객체에 대한 권한을 부여하는 명령입니다.
  예 GRANT RESOURCE TO KORA; : 사용자 ID가 KORA인 사람에게 데이터베이스 및 테이블을 생성할 수 있는 권한을 부여합니다.
- REVOKE : 객체에 부여된 권한을 취소하는 명령입니다.
  예 REVOKE UPDATE ON 고객 FROM JULIA; : 사용자 ID가 JULIA인 사람에게 부여한 권한 중 갱신(UPDATE) 권한을 취소합니다.

전문가의 조언

접근통제의 3요소인 접근통제 정책, 접근통제 메커니즘, 접근통제 보안모델의 개념 및 각각에 해당하는 종류를 구분하여 알아두세요.

신분 기반 정책	• 주체나 그룹의 신분에 근거하여 객체의 접근을 제한하는 방법으로, IBP와 GBP가 있다. • IBP(Individual-Based Policy) : 최소 권한 정책으로, 단일 주체에게 하나의 객체에 대한 허가를 부여한다. • GBP(Group-Based Policy) : 복수 주체에 하나의 객체에 대한 허가를 부여한다.
규칙 기반 정책	• 주체가 갖는 권한에 근거하여 객체의 접근을 제한하는 방법으로, MLP와 CBP가 있다. • MLP(Multi-Level Policy) : 사용자 및 객체별로 지정된 기밀 분류에 따른 정책 • CBP(Compartment-Based Policy) : 집단별로 지정된 기밀 허가에 따른 정책
역할 기반 정책	GBP의 변형된 정책으로, 주체의 신분이 아니라 주체가 맡은 역할에 근거하여 객체의 접근을 제한하는 방법이다. 예 인사 담당자, DBA 등

3 접근통제 매커니즘

접근통제 매커니즘은 정의된 접근통제 정책을 구현하는 기술적인 방법으로, 접근통제 목록, 능력 리스트, 보안 등급, 패스워드, 암호화 등이 있다.

- **접근통제 목록(Access Control List)** : 객체를 기준으로 특정 객체에 대해 어떤 주체가 어떤 행위를 할 수 있는지를 기록한 목록이다.
- **능력 리스트(Capability List)** : 주체를 기준으로 주체에게 허가된 자원 및 권한을 기록한 목록이다.
- **보안 등급(Security Label)** : 주체나 객체 등에 부여된 보안 속성의 집합으로, 이 등급을 기반으로 접근 승인 여부가 결정된다.
- **패스워드** : 주체가 자신임을 증명할 때 사용하는 인증 방법이다.
- **암호화** : 데이터를 보낼 때 지정된 수신자 이외에는 내용을 알 수 없도록 평문을 암호문으로 변환하는 것으로, 무단 도용을 방지하기 위해 주로 사용된다.

4 접근통제 보안 모델

접근통제 보안 모델은 보안 정책을 구현하기 위한 정형화된 모델로, 기밀성 모델, 무결성 모델, 접근통제 모델이 있다.

기밀성* 모델

기밀성 모델은 군사적인 목적으로 개발된 최초의 수학적 모델로, 기밀성 보장이 최우선인 모델이다.

- 기밀성 모델은 군대 시스템 등 특수 환경에서 주로 사용된다.
- 제약 조건
  - 단순 보안 규칙 : 주체는 자신보다 높은 등급의 객체를 읽을 수 없다.
  - ★(스타)-보안 규칙* : 주체는 자신보다 낮은 등급의 객체에 정보를 쓸 수 없다.
  - 강한 ★(스타) 보안 규칙 : 주체는 자신과 등급이 다른 객체를 읽거나 쓸 수 없다.

기밀성(Confidentiality)
기밀성은 시스템 내의 정보와 자원은 인가된 사용자에게만 접근이 허용되는 것으로, 정보가 전송 중에 노출되더라도 데이터를 읽을 수 없는 것을 의미합니다.

★(스타)-보안 규칙
★(스타)-보안 규칙은 높은 등급의 주체가 자신이 접근 가능한 정보를 낮은 등급으로 복사하여 유출하는 행위를 방지하기 위한 것으로, 정보의 기밀성 보호를 위한 제약입니다.

Level	단순 보안 규칙	★(스타)-보안 규칙	강한 ★(스타) 보안 규칙
	읽기 권한	쓰기 권한	읽기/쓰기 권한
높은 등급	통제	가능	통제
같은 등급	가능	가능	가능
낮은 등급	가능	통제	통제

무결성* 모델

무결성 모델은 기밀성 모델에서 발생하는 불법적인 정보 변경을 방지하기 위해 무결성을 기반으로 개발된 모델이다.

- 무결성 모델은 데이터의 일관성 유지에 중점을 두어 개발되었다.
- 무결성 모델은 기밀성 모델과 동일하게 주체 및 객체의 보안 등급을 기반으로 한다.
- 제약 조건
  - 단순 무결성 규칙 : 주체는 자신보다 낮은 등급의 객체를 읽을 수 없다.
  - ★(스타)-무결성 규칙 : 주체는 자신보다 높은 등급의 객체에 정보를 쓸 수 없다.

Level	단순 무결성 규칙	★(스타)-무결성 규칙
	읽기 권한	쓰기 권한
높은 등급	가능	통제
같은 등급	가능	가능
낮은 등급	통제	가능

무결성(Integrity)
무결성은 시스템 내의 정보는 인가된 사용자만 수정할 수 있는 것으로, 정보의 내용이 전송 중에 수정되지 않고 전달되는 것을 보장합니다.

접근통제 모델

접근통제 모델은 접근통제 메커니즘을 보안 모델로 발전시킨 것으로, 대표적으로 접근통제 행렬(Access Control Matrix)이 있다.

- **접근통제 행렬(Access Control Matrix)**

  임의적인 접근통제를 관리하기 위한 보안 모델로, 행은 주체, 열은 객체 즉, 행과 열로 주체와 객체의 권한 유형을 나타낸다.
  - 행 : 주체로서 객체에 접근을 시도하는 사용자이다.
  - 열 : 객체로서 접근통제가 이뤄지는 테이블, 컬럼, 뷰 등과 같은 데이터베이스의 개체이다.
  - 규칙 : 주체가 객체에 대하여 수행하는 입력, 수정, 삭제 등의 데이터베이스에 대한 조작이다.

예 접근통제 행렬 모델

이름	사원	부서	상품	판매현황	거래업체
김상민	ALL	ALL	R	R	R
우지경	R/W	R/W	R	R	R
하석민	R	R	R	R/W	R/W
양구성	R	R	R/W		

- ALL : 모든 권한
- R(Read) : 읽기 권한
- W(Write) : 쓰기 권한

5 접근통제 조건

접근통제 조건은 접근통제 매커니즘의 취약점을 보완하기 위해 접근통제 정책에 부가하여 적용할 수 있는 조건이다.

- **값 종속 통제(Value-Dependent Control)** : 일반적으로는 객체에 저장된 값에 상관없이 접근통제를 동일하게 허용하지만 객체에 저장된 값에 따라 다르게 접근통제를 허용해야 하는 경우에 사용한다.

  예 납입한 금액에 따라 보안 등급이 설정되고, 보안 등급에 따라 접근 여부가 결정되는 경우

- **다중 사용자 통제(Multi-User Control)** : 지정된 객체에 다수의 사용자가 동시에 접근을 요구하는 경우에 사용된다.

  예 여러 명으로 구성된 한 팀에서 다수결에 따라 접근 여부가 결정되는 경우

- **컨텍스트 기반 통제(Context-Based Control)** : 특정 시간, 네트워크 주소, 접근 경로, 인증 수준 등에 근거하여 접근을 제어하는 방법으로, 다른 보안 정책과 결합하여 보안 시스템의 취약점을 보완할 때 사용된다.

  예 근무시간, 즉 월요일~금요일 9:00~18:00 사이에만 접근 가능

6 감사 추적

감사 추적은 사용자나 애플리케이션이 데이터베이스에 접근하여 수행한 모든 활동을 기록하는 기능이다.

- 감사 추적은 오류가 발생한 데이터베이스를 복구하거나 부적절한 데이터 조작을 파악하기 위해 사용된다.
- 감사 추적 시 실행한 프로그램, 사용자, 날짜 및 시간, 접근한 데이터의 이전 값 및 이후 값 등이 저장된다.

따라잡기

기출문제 따라잡기

Section 096

출제예상

1. 다음 중 데이터 보안을 위한 접근통제에 대한 설명으로 틀린 것은?

① 접근통제는 객체와 이를 사용하려는 주체 사이의 정보 흐름을 제한하는 것이다.

② 접근통제는 허가된 사용자만이 객체에 접근하게 함으로써 불법적인 자원 접근 및 파괴를 예방한다.

③ 임의 접근통제는 데이터에 접근하는 사용자의 신원에 따라 접근 권한을 부여하는 것으로, 제3자가 접근통제 권한을 지정한다.

④ 강제 접근통제는 주체와 객체의 등급을 비교하여 접근 권한을 부여하는 것으로, 객체별 보안 등급과 사용자별 인가 등급을 부여할 수 있다.

임의 접근통제는 통제 권한을 가지고 있는 주체가 접근통제 권한을 지정합니다. 제3자가 접근통제 권한을 지정하는 것은 강제 접근통제입니다.

출제예상

2. 다음 중 접근통제의 3요소가 아닌 것은?

① 접근통제 정책

② 접근통제 보안 모델

③ 접근통제 조건

④ 접근통제 메커니즘

모두 앞에서 공부한 내용인데, 접근통제 3요소 혼동되죠? **정**(정책)**보**(보안 모델)의 중요성이 **커**(메커니즘)로 암기하세요.

출제예상

3. 다음 중 접근통제 매커니즘에 대한 설명으로 옳지 않은 것은?

① 접근통제 매커니즘은 보안 정책을 구현하기 위한 정형화된 모델이다.

② 패스워드는 주체가 자신임을 증명할 때 사용하는 인증 방법이다.

③ 접근통제 목록은 객체를 기준으로 특정 객체에 대해 어떤 주체가 어떤 행위를 할 수 있는지를 기록한 목록이다.

④ 능력 리스트는 주체를 기준으로 주체에게 허가된 자원 및 권한을 기록한 목록이다.

접근통제 매커니즘은 정의된 접근통제 정책을 구현하는 기술적인 방법입니다.

출제예상

4. 다음 중 접근통제 보안 모델에 대한 설명으로 틀린 것은?

① 기밀성 모델은 군사적인 목적으로 개발된 최초의 수학적 모델로, 기밀성 보장이 최우선인 모델이다.

② 무결성 모델은 기밀성 모델에서 발생하는 불법적인 정보 변경을 방지하기 위해 무결성을 기반으로 개발된 모델이다.

③ 기밀성은 시스템 내의 정보는 인가된 사용자만 수정할 수 있는 것으로, 정보가 전송 중에 수정되지 않고 전달되는 것을 보장한다.

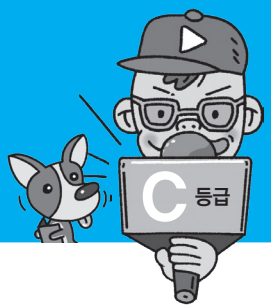
④ 무결성 모델은 데이터의 일관성 유지에 중점을 두어 개발되었다.

인가된 사용자만 접근할 수 있는 것은 기밀성, 인가된 사용자만 수정할 수 있는 것은 무결성! 혼동하지 마세요.

▶ 정답 : 1. ③ 2. ③ 3. ① 4. ③

SECTION 097 ● LIVE

데이터베이스 백업



전문가의 조언

백업은 상식적인 수준에서 생각해도 어떤 건지는 대충 알겠죠? 백업의 개념을 중심으로 장애 유형, 로그 파일, 복구 알고리즘 등을 정리해 두세요.

1 데이터베이스 백업

데이터베이스 백업은 전산 장비의 장애에 대비하여 데이터베이스에 저장된 데이터를 보호하고 복구하기 위한 작업으로, 치명적인 데이터 손실을 막기 위해서는 데이터베이스를 정기적으로 백업해야 한다.

- 데이터베이스 관리 시스템(DBMS)은 데이터베이스 파괴 및 실행 중단이 발생하면 이를 복구할 수 있는 기능을 제공한다.

2 데이터베이스 장애 유형

데이터베이스의 장애 유형을 정확히 파악하고 장애에 따른 백업 전략을 세워야 장애 발생 시 복구가 가능하다.

- **사용자 실수** : 사용자의 실수로 인해 테이블이 삭제되거나 잘못된 트랜잭션이 처리된 경우
- **미디어 장애** : CPU, 메모리, 디스크 등 하드웨어 장애나 데이터가 파손된 경우
- **구문 장애** : 프로그램 오류나 사용 공간의 부족으로 인해 발생하는 장애
- **사용자 프로세스 장애** : 프로그램이 비정상적으로 종료되거나 네트워크 이상으로 세션*이 종료되어 발생하는 오류
- **인스턴스 장애** : 하드웨어 장애, 정전, 시스템 파일 파손 등 비정상적인 요인으로 인해 메모리나 데이터베이스 서버의 프로세스가 중단된 경우

세션(Session)
세션은 사용자 또는 프로세스 간의 데이터 교환을 위한 논리적인 연결을 의미합니다.

3 로그 파일

로그 파일은 데이터베이스의 처리 내용이나 이용 상황 등 상태 변화를 시간의 흐름에 따라 모두 기록한 파일로, 데이터베이스의 복구를 위해 필요한 가장 기본적인 자료이다.

- 로그 파일을 기반으로 데이터베이스를 과거 상태로 복귀(UNDO)시키거나 현재 상태로 재생(REDO)시켜 데이터베이스 상태를 일관성 있게 유지할 수 있다.
- 로그 파일은 트랜잭션 시작 시점, Rollback* 시점, 데이터 입력, 수정 삭제 시점 등에서 기록된다.
- **로그 파일 내용** : 트랜잭션이 작업한 모든 내용, 트랜잭션 식별, 트랜잭션 레코드, 데이터 식별자, 갱신 이전 값(Before Image), 갱신 이후 값(After Image) 등

Rollback
데이터베이스 조작 작업이 비정상적으로 종료되었을 때 원래의 상태로 복구하는 명령어입니다.

4 데이터베이스 복구 알고리즘

데이터베이스 복구 알고리즘은 동기적/비동기적 갱신*에 따라 NO-UNDO/REDO*, UNDO/NO-REDO, UNDO/REDO, NO-UNDO/NO-REDO 방법으로 분류된다.

NO-UNDO/REDO	• 데이터베이스 버퍼의 내용을 비동기적으로 갱신한 경우의 복구 알고리즘 • NO-UNDO : 트랜잭션 완료 전에는 변경 내용이 데이터베이스에 기록되지 않으므로 취소할 필요가 없다. • REDO : 트랜잭션 완료 후 데이터베이스 버퍼에는 기록되어 있고, 저장매체에는 기록되지 않았으므로 트랜잭션 내용을 다시 실행해야 한다.
UNDO/NO-REDO	• 데이터베이스 버퍼의 내용을 동기적으로 갱신한 경우의 복구 알고리즘 • UNDO : 트랜잭션 완료 전에 시스템이 파손되었다면 변경된 내용을 취소한다. • NO-REDO : 트랜잭션 완료 전에 데이터베이스 버퍼 내용을 이미 저장 매체에 기록했으므로 트랜잭션 내용을 다시 실행할 필요가 없다.
UNDO/REDO	• 데이터베이스 버퍼의 내용을 동기/비동기적으로 갱신한 경우의 복구 알고리즘 • 데이터베이스 기록 전에 트랜잭션이 완료될 수 있으므로 완료된 트랜잭션이 데이터베이스에 기록되지 못했다면 다시 실행해야 한다.
NO-UNDO/NO-REDO	• 데이터베이스 버퍼의 내용을 동기적으로 저장 매체에 기록하지만 데이터베이스와는 다른 영역에 기록한 경우의 복구 알고리즘 • NO-UNDO : 변경 내용은 데이터베이스와 다른 영역에 기록되어 있으므로 취소할 필요가 없다. • NO-REDO : 다른 영역에 이미 기록되어 있으므로 트랜잭션을 다시 실행할 필요가 없다.

5 백업 종류

백업 종류는 복구 수준에 따라서 운영체제를 이용하는 물리 백업과 DBMS 유틸리티를 이용하는 논리 백업으로 나뉜다.

- **물리 백업** : 데이터베이스 파일을 백업하는 방법으로, 백업 속도가 빠르고 작업이 단순하지만 문제 발생 시 원인 파악 및 문제 해결이 어렵다.
- **논리 백업** : DB 내의 논리적 객체들을 백업하는 방법으로, 복원 시 데이터 손상을 막고 문제 발생 시 원인 파악 및 해결이 수월하지만 백업/복원 시 시간이 많이 소요된다.

구분	설명	복구 수준
물리 백업	로그 파일 백업 실시	완전 복구
	로그 파일 백업 없음	백업 시점까지 복구
논리 백업	DBMS 유틸리티	

• 동기적 갱신(Synchronous I/O) : 트랜잭션이 완료되기 전에 데이터베이스 버퍼 내용을 동시적으로 저장매체에 기록하는 것
• 비동기적 갱신(Asynchronous I/O) : 트랜잭션이 완료된 내용을 일정 주기나 작업량에 따라 시간 차이를 두고 저장매체에 기록하는 것
• UNDO : 기록한 내용을 취소함
• REDO : 트랜잭션 내용을 다시 실행함

따라잡기

기출문제 따라잡기

Section 097

출제예상

1. 다음 중 데이터베이스 백업에 대한 설명으로 틀린 것은?

① DBMS는 데이터베이스 파괴 시 복구하는 기능을 제공한다.

② 데이터베이스 백업은 정기적으로 수행하는 것이 좋다.

③ 데이터베이스의 상태를 일관성 있게 유지시키기 위해 로그 파일이 이용된다.

④ 로그 파일은 데이터베이스의 상태 변화에 대한 내용을 작업 순서에 따라 기록한 파일이다.

로그 파일은 작업 순서가 아닌 시간의 흐름에 따라 기록됩니다.

출제예상

2. 데이터 양이 방대해지면서 백업에 대한 중요성이 부각되고 있다. 다음 중 백업에 대한 설명으로 가장 옳지 않은 것은?

① 백업 종류에는 물리 백업과 논리 백업이 있다.

② 물리 백업은 운영체제를 이용한 것으로 데이터베이스 파일을 백업한다.

③ 논리 백업은 DBMS를 이용한 것으로 DB 내의 논리적 객체들을 백업한다.

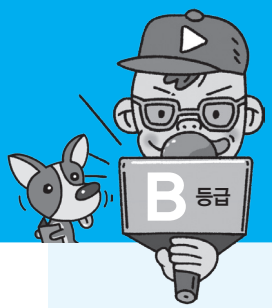
④ 논리 백업은 완전 복구가 가능하다.

논리 백업은 완전 복구가 아닌 백업 시점까지만 복구할 수 있습니다.

▶ 정답 : 1. ④ 2. ④

SECTION 098 LIVE

스토리지



1 스토리지(Storage)의 개요

스토리지는 단일 디스크로 처리할 수 없는 대용량의 데이터를 저장하기 위해 서버와 저장장치를 연결하는 기술이다.

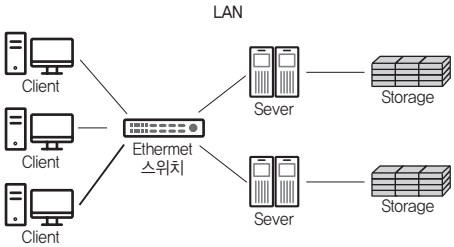
- 스토리지의 종류에는 DAS, NAS, SAN이 있다.

2 DAS(Direct Attached Storage)

DAS는 서버와 저장장치를 전용 케이블로 직접 연결하는 방식으로, 일반 가정에서 컴퓨터에 외장하드를 연결하는 것이 여기에 해당된다.

- 서버에서 저장장치를 관리한다.
- 저장장치를 직접 연결하므로 속도가 빠르고 설치 및 운영이 쉽다.
- 초기 구축 비용 및 유지보수 비용이 저렴하다.
- 직접 연결 방식이므로 다른 서버에서 접근할 수 없고 파일을 공유할 수 없다.
- 확장성 및 유연성이 상대적으로 떨어진다.
- 저장 데이터가 적고 공유가 필요 없는 환경에 적합하다.

예 DAS 구조



서버에 스토리지가 직접 연결되어 있어 속도는 빠르나 다른 서버에서 스토리지에 접근하여 사용할 수 없다.

3 NAS(Network Attached Storage)

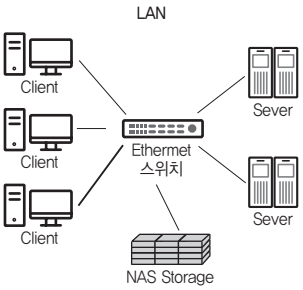
NAS는 서버와 저장장치를 네트워크를 통해 연결하는 방식이다.

- 별도의 파일 관리 기능이 있는 NAS Storage가 내장된 저장장치를 직접 관리한다.
- Ethernet 스위치를 통해 다른 서버에서도 스토리지에 접근할 수 있어 파일 공유가 가능하고, 장소에 구애받지 않고 저장장치에 쉽게 접근할 수 있다.
- DAS에 비해 확장성 및 유연성이 우수하다.
- 접속 증가 시 성능이 저하될 수 있다.

전문가의 조언

스토리지의 종류 세 가지인 DAS, NAS, SAN의 특징을 서로 구분할 수 있도록 정리하세요.

예 NAS 구조



파일 관리 기능이 있는 NAS Storage가 네트워크 상에 독립적으로 연결되어 있으므로 서버들이 자유롭게 스토리지에 접근하여 파일을 공유할 수 있다.

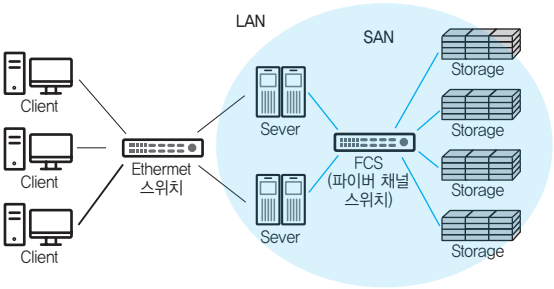
4 SAN(Storage Area Network)

SAN은 DAS의 빠른 처리와 NAS의 파일 공유 장점을 혼합한 방식으로, 서버와 저장장치를 연결하는 전용 네트워크를 별도로 구성하는 방식이다.

- 파이버 채널(FC)* 스위치를 이용하여 네트워크를 구성한다.
- 파이버 채널 스위치는 서버나 저장장치를 광케이블로 연결하므로 처리 속도가 빠르다.
- 서버들이 저장장치 및 파일을 공유할 수 있다.
- 확장성, 유연성, 가용성이 뛰어나다.
- 높은 트랜잭션 처리에 효과적이나 기존 시스템의 경우 장비의 업그레이드가 필요하고, 초기 설치 시에는 별도의 네트워크를 구축해야 하므로 비용이 많이 든다.

파이버 채널(Fibre Channel, FC)
파이버 채널은 컴퓨터 장치 간 데이터의 전송 속도를 기가바이트로 높이기 위한 네트워크 기술입니다. 초기에는 슈퍼컴퓨터 분야에서 주로 사용되었으나 최근에는 SAN의 표준 연결 방식에도 사용되고 있습니다.

예 SAN 구조



서버와 스토리지가 파이버 채널 스위치로 연결된 별도의 전용 스토리지 네트워크를 구성하는 방식으로 서버들이 저장장치 및 파일을 자유롭게 공유할 수 있다.

따라잡기

기출문제 따라잡기

Section 098

출제예상

1. 다음 보기는 무엇에 대한 설명인가?

- 저장장치와 서버를 연결하는 별도의 네트워크를 구성해 데이터를 관리하는 방식이다.
- 파이버 채널(FC) 스위치를 이용하여 네트워크를 구성한다.
- 높은 트랜잭션 처리에 효과적이나 기존 시스템 장비의 업그레이드가 필요하다.

① DAS(Direct Attached Storage)
② NAS(Network Attached Storage)
③ SAN(Storage Area Network)
④ SDS(Software Defined Storage)

파이버 채널 하면 SAN! 꼭 기억하세요.

출제예상

2. 다음 중 스토리지에 대한 설명으로 틀린 것은?

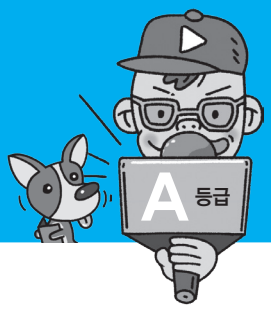
① 스토리지는 서버와 저장장치를 연결하는 하나의 기술이다.
② DAS는 서버와 저장장치를 전용 케이블로 직접 연결하는 방식으로, 확장성과 유연성이 상대적으로 우수하다.
③ NAS는 서버와 저장장치를 네트워크를 통해 연결하는 방식으로 접속 증가 시 성능이 저하될 수 있다.
④ SAN은 별도의 네트워크를 구성해 데이터를 관리하는 방식으로, 초기 별도의 네트워크를 구축해야 하므로 비용이 많이 든다.

DAS는 NAS나 SAN에 비해 확장성과 유연성이 떨어집니다.

▶ 정답 : 1. ③ 2. ②

SECTION 099 ● LIVE

논리 데이터 모델의 물리 데이터 모델 변환



전문가의 조언

논리 데이터 모델에서 설계한 엔티티(Entity), 속성, 주 식별자, 외부 식별자를 물리 데이터 모델의 테이블, 컬럼, 기본키, 외래키로 변환하는 과정 및 특징에 대해 알아두세요.

1 테이블(Table)

테이블은 데이터를 저장하는 데이터베이스의 가장 기본적인 오브젝트이다.

- 테이블은 컬럼(Column, 열)과 로우(Row, 행)로 구성되며, 컬럼에는 지정된 유형에 따라 데이터가 저장된다.
- 테이블의 구성 요소

로우(Row)	튜플, 인스턴스, 어커런스라고도 한다.
컬럼(Column)	각 속성 항목에 대한 값을 저장한다.
기본키 (Primary key)	• 기본키는 후보키 중에서 선택한 주키(Main Key)이다. • 한 릴레이션에서 특정 튜플을 유일하게 구별할 수 있는 속성이다.
외래키 (Foreign key)	• 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미한다. • 한 릴레이션에 속한 속성 A와 참조 릴레이션의 기본키인 B가 동일한 도메인 상에서 정의되었을 때의 속성 A를 외래키라고 한다.

Column

사원번호	이름	직급	연봉	부서번호
s001	최복선	사원	25,000,000	b02
s002	마정희	과장	37,000,000	b01
s003	이영호	사원	28,000,000	b03

Row

기본키 테이블의 구성 요소 외래키

2 엔티티(Entity)를 테이블로 변환

논리 데이터 모델에서 정의된 엔티티를 물리 데이터 모델의 테이블로 변환하는 것이다.

- 엔티티를 테이블로 변환한 후 테이블 목록 정의서를 작성한다.
  - 테이블 목록 정의서 : 전체 테이블을 목록으로 요약 관리하는 문서로, 테이블 목록이라고도 한다.

예 테이블 목록 정의서

테이블ID	테이블명	타입	분류	테이블 스페이스	파티션 여부	발생 주기	총건수	수정일
usr_item	제품	STANDARD						2019/02/20
sell_list	판매목록	STANDARD						2019/03/15
usr_name	사용자	STANDARD	key					2019/03/10
sell_name	판매자	STANDARD						2019/03/10

- 변환 규칙

논리적 설계(데이터 모델링)	물리적 설계
엔티티(Entity)	테이블(Table)
속성(Attribute)	컬럼(Column)
주 식별자(Primary Identifier)	기본키(Primary Key)
외부 식별자(Foreign Identifier)	외래키(Foreign Key)
관계(Relationship)	관계(Relationship)

예 엔티티를 테이블로 변환

〈주문목록〉과 〈주문조회〉 엔티티가 〈주문목록〉과 〈주문조회〉 테이블로 변환되었다.

- 변환 시 고려사항
  - 일반적으로 테이블과 엔티티 명칭은 동일하게 하는 것을 권고한다.
  - 엔티티는 주로 한글명을 사용하지만 테이블은 소스 코드의 가독성을 위해 영문명을 사용한다.
  - 메타 데이터 관리 시스템*에 표준화된 용어가 있을 때는 메타에 등록된 단어를 사용하여 명명한다.

메타 데이터 관리 시스템
메타 데이터 관리 시스템은 메타 데이터를 수집하거나 여러 사람이 메타 데이터를 편리하게 사용할 수 있도록 제공하는 시스템입니다.

3 슈퍼타입/서브타입을 테이블로 변환

슈퍼타입/서브타입은 논리 데이터 모델에서 이용되는 형태이므로 물리 데이터 모델을 설계할 때는 슈퍼타입/서브타입을 테이블로 변환해야 한다.

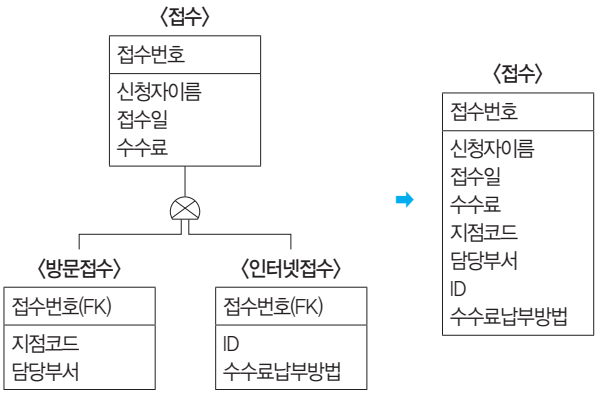
- 슈퍼타입/서브타입 모델을 테이블로 변환하는 방법에는 슈퍼타입 기준 테이블 변환, 서브타입 기준 테이블 변환, 개별타입 기준 테이블 변환이 있다.

슈퍼타입 기준 테이블 변환

슈퍼타입 기준의 테이블 변환은 서브타입을 슈퍼타입에 통합하여 하나의 테이블로 만드는 것이다.

- 서브타입에 속성이나 관계가 적을 경우에 적용하는 방법으로, 하나로 통합된 테이블에는 서브타입의 모든 속성이 포함되어야 한다.

예 슈퍼타입 기준 테이블 변환



서브타입의 〈방문접수〉 개체에 있는 '지점코드', '담당부서'와 〈인터넷접수〉 개체에 있는 'ID', '수수료납부방법'이 슈퍼타입인 〈접수〉 개체에 통합되어 〈접수〉 테이블로 변환된다.

- 장·단점

장점	• 데이터의 액세스가 상대적으로 용이하다. • 뷰를 이용하여 각각의 서브타입만을 액세스하거나 수정할 수 있다. • 서브타입 구분이 없는 임의 집합에 대한 처리가 용이하다. • 여러 테이블을 조인하지 않아도 되므로 수행 속도가 빨라진다. • SQL 문장 구성이 단순해진다.
단점	• 테이블의 컬럼이 증가하므로 디스크 저장 공간이 증가한다. • 처리마다 서브타입에 대한 구분(TYPE)이 필요한 경우가 많이 발생한다. • 인덱스 크기의 증가로 인덱스의 효율이 떨어진다.

서브타입 기준 테이블 변환

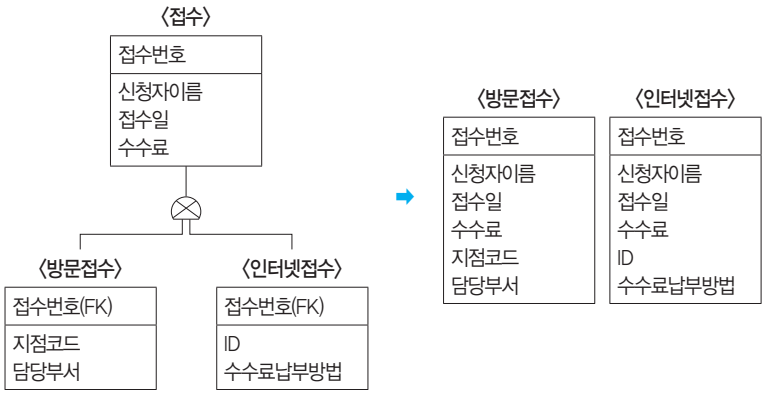
서브타입 기준의 테이블 변환은 슈퍼타입 속성들을 각각의 서브타입에 추가하여 서브타입들을 개별적인 테이블로 만드는 것이다.

- 서브타입에 속성이나 관계가 많이 포함된 경우 적용한다.
- 장·단점

장점	• 각 서브타입 속성들의 선택 사양이 명확한 경우에 유리하다. • 처리할 때마다 서브타입 유형을 구분할 필요가 없다. • 여러 개의 테이블로 통합하므로 테이블당 크기가 감소하여 전체 테이블 스캔*시 유리하다.
단점	• 수행 속도가 감소할 수 있다. • 복잡한 처리를 하는 SQL의 통합이 어렵다. • 부분 범위에 대한 처리가 곤란해진다. • 여러 테이블을 통합한 뷰는 조회만 가능하다. • UID(Unique Identifier, 식별자)의 유지 관리가 어렵다

스캔(SCAN) 방식
스캔 방식은 테이블의 데이터를 검색할 때 앞에서부터 순차적으로 행(Row)을 비교하여 원하는 데이터를 가져오는 접근 방법입니다. 전체 스캔(Full Scan)은 전체 테이블을, 범위 스캔(Range Scan)은 테이블의 특정 부분만을 검색하는 방법입니다.

예 서브타입 기준 테이블 변환



슈퍼타입인 〈접수〉 개체에 있는 '신청자이름', '접수일', '수수료'가 서브타입인 〈방문접수〉 개체와 〈인터넷접수〉 개체에 각각 추가되어 〈방문접수〉와 〈인터넷접수〉 테이블로 변환된다.

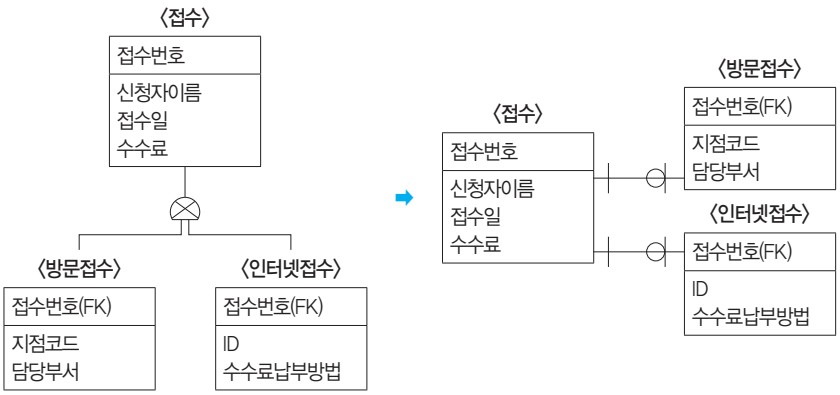
개별타입 기준 테이블 변환

개별타입 기준의 테이블 변환은 슈퍼타입과 서브타입들을 각각의 개별적인 테이블로 변환하는 것이다.

- 슈퍼타입과 서브타입 테이블들 사이에는 각각 1:1 관계가 형성된다.
- 개별타입 기준 테이블 변환을 적용하는 경우
  - 전체 데이터에 대한 처리가 빈번한 경우
  - 서브타입의 처리가 대부분 독립적으로 발생하는 경우
  - 통합하는 테이블의 컬럼 수가 많은 경우
  - 서브타입의 컬럼 수가 많은 경우
  - 트랜잭션이 주로 슈퍼타입에서 발생하는 경우
  - 슈퍼타입의 처리 범위가 넓고 빈번하게 발생하여 단일 테이블 클러스터링이 필요한 경우
- 장·단점

장점	• 저장 공간이 상대적으로 작다. • 슈퍼타입 또는 서브타입 각각의 테이블에 속한 정보만 조회하는 경우 문장 작성이 용이하다.
단점	슈퍼타입 또는 서브타입의 정보를 같이 처리하면 항상 조인이 발생하여 성능이 저하된다.

예 개별타입 기준 테이블 변환



슈퍼타입의 〈접수〉 개체와 서브타입의 〈방문접수〉, 〈인터넷접수〉 개체가 각각 〈접수〉, 〈방문접수〉, 〈인터넷접수〉 테이블로 변환된다.

4 속성을 컬럼으로 변환

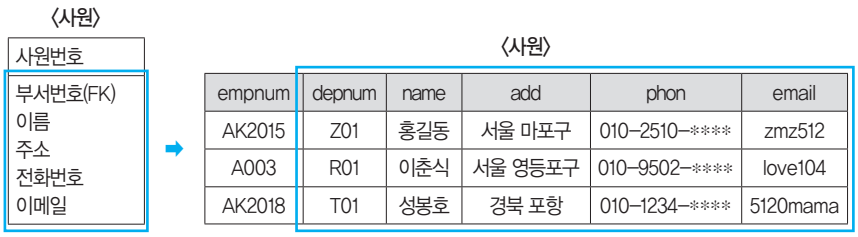
논리 데이터 모델에서 정의한 속성을 물리 데이터 모델의 컬럼으로 변환한다.

일반 속성 변환

- 속성과 컬럼은 명칭이 반드시 일치할 필요는 없으나, 개발자와 사용자 간 의사소통을 위하여 가능한 한 표준화된 약어*를 사용하여 일치시키는 것이 좋다.
- 컬럼명은 SQL의 예약어(Reserved Word) 사용을 피한다.
- 컬럼명은 SQL의 가독성을 높이기 위해 가능한 한 짧게 지정한다.
- 복합 단어를 컬럼명으로 사용할 때는 미리 정의된 표준을 따른다.
- 테이블의 컬럼을 정의한 후에는 한 로우(Row)에 해당하는 샘플 데이터를 작성하여 컬럼의 정합성을 검증한다.

표준화된 약어
데이터 표준화를 위해 정의된 표준 단어에는 기본적으로 영문명과 영문 약어명이 정의되어 있습니다.

예 일반 속성 변환



empnum	depnum	name	add	phon	email
AK2015	Z01	홍길동	서울 마포구	010-2510-****	zmz512
A003	R01	이춘식	서울 영등포구	010-9502-****	love104
AK2018	T01	성봉호	경북 포항	010-1234-****	5120mama

〈사원〉 엔티티의 '부서번호', '이름', '주소', '전화번호', '이메일' 속성이 〈사원〉 테이블의 각각의 컬럼으로 변환되었으며, 예시를 위한 데이터가 들어있다.

Primary UID*를 기본키로 변환

논리 데이터 모델에서의 Primary UID는 물리 데이터 모델의 기본키로 만든다.

Primary UID
UID(Unique IDentifier)는 식별자, Primary UID는 주 식별자를 의미합니다. 주 식별자에 대한 자세한 내용은 Section 075를 참조하세요.

Primary UID(관계의 UID Bar*)를 기본키로 변환

다른 엔티티와의 관계로 인해 생성된 Primary UID는 물리 데이터 모델의 기본키로 만든다.

Secondary(Alternate) UID를 유니크키*로 변환

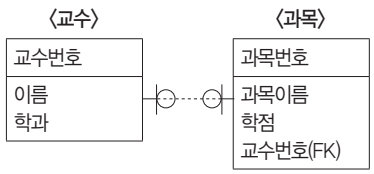
논리 모델링에서 정의된 Secondary UID 및 Alternate Key는 물리 모델에서 유니크키로 만든다.

5 관계를 외래키로 변환

논리 데이터 모델에서 정의된 관계는 기본키와 이를 참조하는 외래키로 변환한다.

- 다음은 개체 A, B로 이루어진 E-R 모델을 관계 테이블로 변환하는 방법이다.
  - 1:1 관계 : 개체 A의 기본키를 개체 B의 외래키로 추가하거나 개체 B의 기본키를 개체 A의 외래키로 추가하여 표현한다.

예 1:1 관계 변환



〈과목〉

과목번호	과목이름	학점	교수번호
B01	미생물	3	A001
B02	전자기	3	A002
B03	운영체제	2	A003
B04	미적분	2	A004

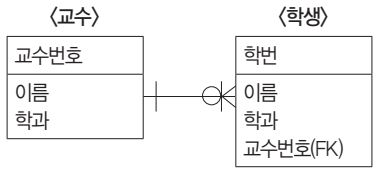
〈교수〉

교수번호	이름	학과
A001	이선경	생물학
A002	조신정	물리학
A003	조양희	전산학
A004	김석호	수학

〈교수〉 테이블의 기본키인 '교수번호' 필드를 〈과목〉 테이블의 외래키로 추가한다.

  - 1:M 관계 : 개체 A의 기본키를 개체 B의 외래키로 추가하여 표현하거나 별도의 테이블로 표현한다.

예 1:M 관계 변환



〈학생〉

학번	이름	학과	교수번호
B01	강민주	생물학	A001
B02	성형수	생물학	A002
B03	김덕인	생물학	A003
B04	하성진	수학	A004

〈교수〉

교수번호	이름	학과
A001	이선경	생물학
A002	조신정	물리학
A003	조양희	전산학
A004	김석호	수학

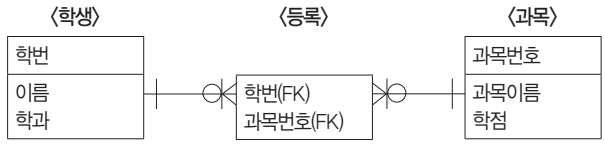
〈교수〉 테이블의 기본키인 '교수번호' 필드를 〈학생〉 테이블의 외래키로 추가한다.

UID Bar
UID Bar란 엔티티에 포함된 고유한 속성의 식별자(UID)가 아니라 다른 엔티티와의 관계로 인해 생성된 식별자(UID)를 의미합니다.

유니크키(Unique Key)
유니크키는 해당 속성에 입력된 값이 유일하다는 것을 보장하기 위한 제약 조건인 유니크(Unique) 속성이 설정한 키입니다.

– N:M 관계 : 릴레이션 A와 B의 기본키를 모두 포함한 별도의 릴레이션으로 표현한다. 이때 생성된 별도의 릴레이션을 교차 릴레이션(Intersection Relation) 또는 교차 엔티티(Intersection Entity)라고 한다.

예 N:M 관계 변환



〈학생〉

학번	이름	학과
B01	강민주	생물학
B02	성형수	생물학
B03	김덕인	생물학
B04	하성진	수학

〈등록〉

학번	과목번호
B01	K01
B01	K02
B01	K03
B02	K02
B02	K04
B04	K02
B04	K03

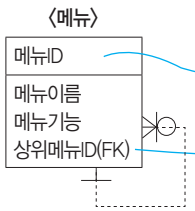
〈과목〉

과목번호	과목이름	학점
K01	미생물	3
K02	전자기	3
K03	운영체제	2
K04	미적분	2

N:M의 관계는 관계를 별도의 테이블로 구성해서 표시해야 한다. 〈학생〉 테이블의 기본키인 '학번' 필드와 〈과목〉 테이블의 기본키인 '과목번호' 필드를 이용하여 〈등록〉 테이블을 만든다.

– 1:M 순환 관계 : 개체 A에 개체 A의 기본키를 참조하는 외래키 컬럼을 추가하여 표현한다. 데이터의 계층 구조를 표현하기 위해 주로 사용된다.

예 1:M 순환 관계 변환



〈메뉴〉

메뉴ID	메뉴이름	메뉴	상위메뉴ID(FK)
ME001	수강관리	수강관리주메뉴	
ME002	수강신청	수강신청	ME001
ME003	수강취소	수강과목취소	ME001
ME004	권장과목	장과목표시	ME001
ME005	이수과목	이수과목표시	ME001

〈메뉴〉 테이블의 기본키인 '메뉴ID' 필드를 참조하는 외래키 '상위메뉴ID' 필드를 추가한다.

6 관리 목적의 테이블/컬럼 추가

논리 데이터 모델에는 존재하지 않는 테이블이나 컬럼을 데이터베이스의 관리 혹은 데이터베이스를 이용하는 프로그래밍의 수행 속도를 향상시키기 위해 물리 데이터 모델에 추가할 수 있다.

예 시스템 등록 일자, 시스템 번호 등

7 데이터 타입 선택

논리 데이터 모델에서 정의된 논리적인 데이터 타입을 물리적인 DBMS의 물리적 특성과 성능을 고려하여 최적의 데이터 타입과 데이터의 최대 길이를 선택한다.

- 주요 타입에는 문자 타입(Character Data Type), 숫자 타입(Numeric Type), 날짜 타입(Datetime Type)이 있다.
- Oracle에서 자주 사용되는 데이터 유형

CHAR	고정길이 문자열 Data 최대 2,000Byte까지 저장 가능
VARCHAR2	가변길이 문자열 Data 최대 4,000Byte까지 저장 가능
NUMBER	38자릿수의 숫자 저장 가능
DATE	날짜 저장

전문가의 조언

데이터 타입은 DBMS마다 많은 형식이 존재하므로, 주어진 상황에 맞는 적절한 데이터 타입을 설정해야 합니다.

따라잡기

기출문제 따라잡기 Section 099

출제예상

1. 다음 중 논리 데이터베이스에서 설계한 객체들을 물리 데이터베이스로 변환하는 과정에 대한 설명으로 틀린 것은?

① 엔티티는 테이블로 변환한다.
② 속성은 컬럼으로 변환한다.
③ 주 식별자는 외래키로 변환한다.
④ 테이블명은 영문으로 지정한다.

주 식별자는 기본키, 외부 식별자는 외래키로 변환합니다.

출제예상

2. 논리 데이터 모델을 물리 데이터 모델로 변환하는 과정에서 슈퍼타입과 서브타입을 테이블로 변환하는 방법이 아닌 것은?

① 슈퍼타입을 기준으로 하나의 테이블로 변환
② 서브타입을 기준으로 하나의 테이블로 변환
③ 슈퍼타입과 서브타입 각각의 테이블로 변환
④ 서브타입을 기준으로 여러 개의 테이블로 변환

하나의 테이블로 변환하는 것은 한 가지뿐 이었습니다. 잘 생각해 보세요.

출제예상

3. 다음 중 논리 데이터 모델을 물리 데이터 모델로 변환할 때 슈퍼타입/서브타입 데이터 모델의 변환 방법으로 가장 부적절한 것은?

① 슈퍼타입과 서브타입을 각각의 독립적인 테이블로 변환하여 테이블 간에 1:M 관계가 유지되도록 변환할 수 있다.
② 슈퍼타입과 서브타입을 통합하여 하나의 테이블로 변환할 수 있다.
③ 슈퍼타입 속성들을 서브타입에 각각 추가하여 서브타입마다 하나의 테이블로 변환할 수 있다.
④ 슈퍼타입을 기준으로 하나의 테이블로 변환하면 데이터 검색이 수월하고 조인이 감소한다.

테이블을 모두 독립적인 테이블로 변환하면 테이블 간에는 1:1 관계가 형성됩니다.

출제예상

4. 다음 중 논리 데이터 모델의 속성을 컬럼으로 변환하는 방법으로 가장 옳지 않은 것은?

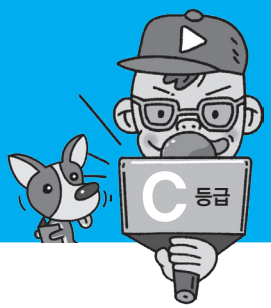
① SQL 예약어의 사용을 피한다.
② 가능한 한 표준화된 약어를 사용한다.
③ SQL의 가독성을 높이기 위해 컬럼명은 가능한 한 길게 지정한다.
④ 실제 데이터를 사용하여 컬럼의 정합성을 검증한다.

컬럼명이 20자나 30자라면 사용하기가 편할까요?

▶ 정답 : 1. ③ 2. ② 3. ① 4. ③

SECTION 100 ● LIVE

물리 데이터 모델 품질 검토



전문가의 조언

데이터 모델을 모두 작성한 다음에는 작성한 모델들이 품질 기준에 따라 정확히 작성되었는지 검토한 후 품질 검토 보고서를 작성해야 합니다. 물리 데이터 모델의 품질 기준에는 어떤 것이 있는지 기억하고 품질 검토 순서를 정리해 두세요.

1 물리 데이터 모델 품질 검토

물리 데이터 모델의 품질 검토는 물리 데이터 모델을 설계하고 데이터베이스 객체를 생성한 후 개발 단계로 넘어가기 전에 모델러와 이해관계자들이 모여 수행한다.

- 물리 데이터 모델은 시스템 성능에 직접적인 영향을 미치므로 향후 발생할 문제에 대해 면밀히 검토해야 한다.
- 물리 데이터 모델 품질 검토의 목적은 데이터베이스의 성능 향상과 오류 예방이다.
- 물리 데이터 모델을 검토하려면 모든 이해관계자가 동의하는 검토 기준이 필요하다.

2 물리 데이터 모델 품질 기준

대표적인 물리 데이터 모델의 품질 기준으로는 정확성, 완전성, 준거성, 최신성, 일관성, 활용성이 있다.

정확성	데이터 모델이 요구사항이나 업무 규칙, 표기법에 따라 정확하게 표현되었음을 의미한다.
완전성	데이터 모델이 데이터 모델의 구성 요소를 누락 없이 정의하고 요구사항이나 업무 영역을 누락 없이 반영하였음을 의미한다.
준거성	데이터 모델이 데이터 표준, 표준화 규칙, 법적 요건 등을 정확하게 준수하였음을 의미한다.
최신성	데이터 모델이 최근의 이슈나 현행 시스템을 반영하고 있음을 의미한다.
일관성	데이터 모델이 표현상의 일관성을 유지하고 있음을 의미한다.
활용성	작성된 모델과 설명을 사용자가 충분히 이해할 수 있고, 업무 변화에 따른 데이터 구조의 변경이 최소화될 수 있도록 설계되었음을 의미한다.

- 물리 데이터 모델의 품질 기준은 조직 혹은 업무 상황에 따라 가감하거나 변형하여 사용한다.

3 물리 데이터 모델 품질 검토 항목

물리 데이터 모델의 품질 검토 항목은 물리 데이터 모델의 특성을 반영한 품질 기준을 작성한 후 이를 기반으로 작성한다.

- 물리 데이터 모델에 정의된 테이블, 컬럼, 무결성 제약 조건 등 물리 데이터 모델의 주요 구성 요소와 반정규화, 인덱스, 스토리지 등 물리 데이터 모델의 전반적인 것을 검토 항목으로 작성한다.
- 체크리스트는 물리 데이터 모델의 품질 검토를 용이하게 수행할 수 있도록 한다.

예 데이터 품질 체크리스트

대상	검토항목	검토내용	평가	비고
테이블	명명	– 명명 규칙을 준수하였는가?		
		– 의미 전달이 명확한 명칭을 사용하였는가?		
		– 테이블 한글명은 엔티티 명칭과 일치하는가?		
	설명	– 데이터 집합 구성상의 특징이 설명되어 있는가?		
	정의	– 테이블 형태는 성능을 고려하여 결정되었는가?		
		– 테이블 생성 관련 파라미터들은 적절하고 충분하게 정의되었는가?		
	권한	– 테이블 생성/변경/삭제 시 메타 데이터 권한을 정의하였는가?		
		– 테이블에 대한 접근 권한을 정의하였는가?		

4 물리 데이터 모델의 품질 검토 순서

❶ 데이터 품질 정책 및 기준을 확인한다.
❷ 물리 데이터 품질의 특성에 따라 품질 기준을 작성한다.
❸ 데이터 품질 기준에 따라 체크리스트를 작성한다.
❹ 논리 데이터 모델과 물리 데이터 모델을 비교한다.
❺ 각 모델링 단계의 모델러와 이해관계자가 품질 검토를 수행한다.
❻ 모델러와 이해관계자가 작성한 체크리스트 내용을 종합하여 물리 데이터베이스 모델의 품질 검토 보고서를 작성한다.

따라잡기

기출문제 따라잡기 Section 100

출제예상
1. 다음 중 물리 데이터 모델의 품질 검토에 대한 내용으로 가장 옳지 않은 것은?

① 물리 데이터 모델 품질 검토의 목적은 데이터베이스의 성능 향상과 오류 예방이다.
② 물리 데이터 모델을 검토하려면 모든 이해관계자가 동의하는 검토 기준이 필요하다.
③ 물리 데이터 모델의 품질 기준에는 정확성, 완전성, 준거성 등이 있다.
④ 데이터 품질 기준 및 정책이 수립된 경우에는 물리 데이터 모델의 품질 기준을 변경하거나 수정해서는 안된다.

데이터 품질 기준 및 정책이 수립된 경우라도 조직이나 업무 상황에 따라 물리 데이터 모델의 품질 기준을 변경하거나 수정할 수 있습니다.

출제예상
2. 다음 중 물리 데이터 모델의 품질 기준에 대한 설명으로 틀린 것은?

① 완전성 : 데이터 모델의 구성 요소나 요구사항의 누락을 최소화함
② 정확성 : 데이터 표준, 표준화 규칙 등을 정확하게 준수함
③ 최신성 : 업무상의 변경이나 결정 사항 등이 시기 적절하게 반영됨
④ 활용성 : 업무 변화에 따른 데이터 구조의 변경이 최소화될 수 있도록 설계함

"표준"과 같은 의미로 "준거"라는 단어가 있습니다. 데이터 표준, 표준화 규칙 하면 준거성!! 꼭 기억해 두세요.

▶ 정답 : 1. ④ 2. ②

예·상·문·제·은·행

1. 데이터베이스 설계 단계 중 응답 시간, 저장 공간의 효율화, 트랜잭션 처리도와 가장 밀접한 관계가 있는 것은?

① 물리적 설계
② 논리적 설계
③ 개념적 설계
④ 요구조건 분석

2. 데이터베이스 설계 단계 중 물리적 설계에 대한 설명으로 옳지 않은 것은?

① 개념적 설계 단계에서 만들어진 정보 구조로부터 특정 목표 DBMS가 처리할 수 있는 스키마를 생성한다.
② 다양한 데이터베이스 응용에 대해서 처리 성능을 얻기 위해 데이터베이스 파일의 저장 구조 및 액세스 경로를 결정한다.
③ 물리적 저장장치에 저장할 수 있는 물리적 구조의 데이터로 변환하는 과정이다.
④ 물리적 설계에서 옵션 선택 시 응답 시간, 저장 공간의 효율화, 트랜잭션 처리율 등을 고려하여야 한다.

3. 물리적 데이터베이스 설계 시 고려 사항으로 가장 거리가 먼 것은?

① 레코드의 크기
② 파일에 대한 트랜잭션의 갱신과 참조 성향
③ 수행될 질의와 트랜잭션의 예상 빈도
④ 인덱스의 구조

4. 데이터베이스를 구성하는 테이블의 종류가 아닌 것은?

① 일반 유형 테이블
② 클러스터 인덱스 테이블
③ 인터럽트 테이블
④ 파티셔닝 테이블

5. 다음 중 물리 데이터베이스 모델의 구성 요소인 테이블에 대한 설명으로 틀린 것은?

① 테이블에는 데이터베이스의 모든 데이터가 저장된다.
② 파티셔닝 테이블은 대용량의 테이블을 물리적 단위로 나눈 것이다.
③ 외부 테이블은 외부 파일을 데이터베이스 내에 존재하는 일반 테이블처럼 이용할 수 있는 데이터베이스의 객체이다.
④ 클러스터 인덱스 테이블은 일반적인 인덱스를 사용하는 테이블에 비해 접근 경로가 단축된다.

6. 트랜잭션의 정의 및 특징이 아닌 것은?

① 한꺼번에 수행되어야 할 일련의 데이터베이스 연산집합
② 사용자의 시스템에 대한 서비스 요구 시 시스템의 상태 변환 과정의 작업 단위
③ 병행제어 및 회복 작업의 논리적 작업 단위
④ 트랜잭션의 연산이 데이터베이스에 모두 반영되지 않고 일부만 반영시키는 원자성의 성질

7. 트랜잭션(Transaction)은 보통 일련의 연산 작업이란 의미로 사용되며, 하나의 논리적 기능을 수행하는 작업의 단위이다. 트랜잭션(Transaction)이 가져야 할 특성에 해당하지 않는 것은?

① 원자성(Atomicity) ② 투명성(Transparency)
③ 일관성(Consistency) ④ 격리성(Isolation)

8. A 은행에서 B라는 사람이 고객 인증 절차를 거쳐 잔액을 조회한 후, 타인에게 송금하는 도중에 장애가 발생하였을 경우 문제가 발생한다. 이러한 경우의 부작용을 방지할 수 있는 트랜잭션의 특성은?

① 일관성(Consistency) ② 고립성(Isolation)
③ 지속성(Duration) ④ 원자성(Atomicity)

9. 트랜잭션 분석서를 작성하는 과정에서 필요 없는 항목에 해당하는 것은?

① 단위 프로세스 ② 연관 모듈
③ CRUD 연산 ④ 테이블 참조 횟수

10. 다음 중 비트맵 인덱스에 대한 설명으로 옳지 않은 것은?

① 비트맵 인덱스란 인덱스 컬럼의 데이터를 Bit 값인 0 또는 1로 변환하여 인덱스 키로 사용하는 방법이다.
② 비트맵 인덱스는 분포도가 넓은 컬럼에 적합하다.
③ 비트맵 인덱스의 목적은 키 값을 포함하는 로우의 주소를 제공하는 것이다.
④ 비트맵 인덱스는 동일한 값이 반복될 확률이 높아 압축 효율이 매우 좋다.

11. 다음 중 인덱스 설계 시 고려할 사항이 아닌 것은?

① 새로 추가되는 인덱스는 기존 액세스 경로에 영향을 미칠 수 있다.
② 결합 인덱스는 구성되는 컬럼 순서에 따라 성능이 차이날 수 있다.
③ 넓은 범위를 인덱스로 처리하면 많은 오버헤드가 발생한다.
④ 인덱스와 테이블 데이터는 같은 공간에 저장되도록 설계해야 한다.

해설은 423쪽에 있습니다.

12. 뷰(View)의 이점으로 거리가 먼 것은?

① 독자적인 인덱스를 가질 수 있으므로 관리가 편하다.
② 여러 사용자의 상이한 응용이나 요구를 지원해 준다.
③ 사용자의 데이터 관리를 간단하게 해준다.
④ 숨겨진 데이터를 위한 자동 보안이 제공된다.

13. 뷰(VIEW)에 관한 설명 중 잘못된 것은?

① 뷰는 가상 테이블이므로 물리적으로 구현되어 있지 않다.
② 뷰는 하나 이상의 기본 테이블로부터 유도되어 만들어지는 가상 테이블이다.
③ 뷰는 INSERT, DELETE, UPDATE 등을 이용한 삽입, 삭제, 갱신 연산이 항상 허용된다.
④ 필요한 데이터만 뷰로 정의해서 처리할 수 있기 때문에 관리가 용이하다.

14. 뷰(VIEW)의 특징으로 옳지 않은 것은?

① 뷰는 기본 테이블 검색 연산과 비교하여 제약이 따른다.
② DBA는 보안 측면에서 뷰를 활용할 수 있다.
③ 뷰 위에 또 다른 뷰를 정의할 수 있다.
④ 물리적으로 존재하지 않는 가상 테이블이다.

15. 다음 중 클러스터에 대한 설명으로 틀린 것은?

① 클러스터링을 하면 비슷한 데이터를 동일한 데이터 블록에 저장하기 때문에 디스크 I/O가 줄어든다.
② 단일 테이블 클러스터링은 조인이 많이 발생하는 경우, 다중 테이블 클러스터링은 처리 범위가 넓은 경우 주로 사용된다.
③ 클러스터링 된 테이블은 클러스터키 열을 공유하므로 저장 공간이 줄어든다.
④ 클러스터는 데이터의 액세스 효율을 향상시킨다.

16. 다음 중 파티셔닝의 장점으로 틀린 것은?

① 데이터 관리가 수월해진다.
② 시스템 장애 시 데이터 손상 정도를 최소화할 수 있다.
③ 파티션별로 백업 및 복구를 수행하므로 속도가 빠르다.
④ 테이블간 조인에 대한 비용이 감소한다.

17. 다음 중 테이블 파티셔닝의 종류에 대한 설명으로 틀린 것은?

① 범위 분할은 일별, 월별, 분기별과 같이 열의 값을 기준으로 분할한다.
② 해시 분할은 해시 함수를 적용한 결과 값에 따라 데이터를 분할한다.
③ 해시 분할은 특정 파티션에 데이터가 집중되는 단점이 있다.
④ 조합 분할은 범위 분할한 파티션이 너무 커서 관리가 어려울 때 유용하다.

18. 분산 데이터베이스의 장점으로 거리가 먼 것은?

① 데이터베이스의 설계와 개발이 용이
② 신뢰성(Reliability)과 가용성(Availability) 향상
③ 질의 처리(Query Processing) 시간의 단축
④ 데이터의 공유성 향상

19. 분산 데이터베이스에 대한 설명으로 잘못된 것은?

① 분산 데이터베이스 관리 시스템의 목적은 사용자들이 데이터가 어느 지역 데이터베이스에 위치하고 있는지를 알 수 있도록 하는 것이다.
② 분산 데이터베이스 관리 시스템의 형태로는 동질 분산 데이터베이스 관리 시스템과 이질 분산 데이터베이스 관리 시스템으로 구분할 수 있다.
③ 분산 데이터베이스에서의 수평 분할은 전역 테이블을 구성하는 튜플들을 부분 집합으로 분할하는 방법을 말한다.
④ 분산 데이터베이스는 데이터의 처리나 이용이 많은 지역에 데이터베이스를 위치시킴으로써 데이터의 처리가 가능한 해당 지역에서 해결될 수 있도록 하는 데이터베이스 시스템이다.

20. 분산 데이터베이스에서 사용자는 데이터가 물리적으로 저장되어 있는 곳을 알 필요 없이 논리적인 입장에서 데이터가 모두 자신의 사이트에 있는 것처럼 처리하는 특성을 무엇이라 하는가?

① 지역 자치성(Local Autonomy)
② 위치 투명성(Location Transparency)
③ 단편 투명성(Fragmentation Transparency)
④ 중복 투명성(Replication Transparency)

21. 다음 중 분산 데이터베이스 시스템의 목적이라 볼 수 없는 것은?

① 생산성의 향상
② 지역적 정보처리의 효율성 증진
③ 보안성 확보
④ 신뢰도와 가용성 증진

▶ 정답 : 1. ① 2. ① 3. ③ 4. ③ 5. ② 6. ④ 7. ② 8. ④ 9. ② 10. ② 11. ④ 12. ① 13. ③ 14. ① 15. ② 16. ④ 17. ③ 18. ① 19. ① 20. ② 21. ③

해설은 424쪽에 있습니다.

22. 다음 중 분산 데이터베이스 시스템의 구성 요소에 해당하지 않는 것은?

① 분산 처리기
② 통신 네트워크
③ 분산 데이터베이스
④ 분산 데이터베이스 컴퓨터

23. 다음 중 이중화와 클러스터링에 대한 설명으로 틀린 것은?

① 이중화와 클러스터링은 모두 고가용성을 위한 방법이다.
② 이중화의 활동-활동(Active-Active) 방법은 두 DB가 서로 다른 서비스를 제공하다가 둘 중 한쪽 DB에 문제가 발생하면 나머지 다른 DB가 서비스를 대신 수행한다.
③ 고가용성 클러스터링은 전체 처리율을 높이기 위해 하나의 작업을 여러 개의 서버에서 분산하여 처리한다.
④ 클러스터링은 두 대 이상의 서버를 하나의 서버처럼 운영하기 위한 기술이다.

24. 데이터베이스의 보안 기법이 아닌 것은?

① 패스워드
② 권한 부여
③ 데이터 암호화
④ 시스템 체크 포인트

25. 다음 중 접근통제 정책이 아닌 것은?

① 신분 기반 정책 ② 업무 기반 정책
③ 규칙 기반 정책 ④ 역할 기반 정책

26. 다음 중 접근통제 보안 모델에 대한 설명으로 틀린 것은?

① 접근통제 보안 모델은 보안 정책을 구현하기 위한 정형화된 모델이다.
② 무결성 모델은 군대 시스템 등 특수 환경에서 주로 사용된다.
③ 무결성 모델은 데이터의 일관성 유지에 중점을 두어 개발되었다.
④ 무결성은 시스템 내의 정보는 인가된 사용자만 수정할 수 있는 것을 의미한다.

27. 다음 중 단일 디스크로 처리할 수 없는 대용량의 데이터를 저장하기 위해 서버와 저장장치를 연결하는 스토리지가 아닌 것은?

① SSD ② NAS
③ SAN ④ DAS

28. 다음 중 슈퍼타입/서브타입 데이터 모델을 테이블로 변환하는 방법으로 가장 부적절한 것은?

① 슈퍼타입을 기준으로 하나의 테이블로 통합하면 여러 테이블을 조인하지 않아도 되므로 수행 속도가 빨라진다.
② 슈퍼타입과 서브타입이 하나로 통합된 테이블에는 모든 서브타입의 속성을 포함할 필요는 없다.
③ 슈퍼타입과 서브타입을 각각의 테이블로 변환하면 테이블 사이에는 1:1 관계가 형성된다.
④ 서브타입을 기준으로 서브타입들을 개별적인 테이블로 변환하면 처리할 때마다 서브타입 유형을 구분할 필요가 없다.

29. 논리 데이터 모델의 서브타입과 슈퍼타입은 물리 데이터 모델의 어떤 객체로 변환되는가?

① 테이블스페이스 ② 테이블
③ 데이터베이스 ④ 컬럼

30. 다음 중 논리 데이터 모델의 속성을 컬럼으로 변환하는 방법으로 틀린 것은?

① 가독성을 높이기 위해 SQL의 예약어를 컬럼명으로 사용한다.
② 컬럼명은 될 수 있으면 짧게 지정한다.
③ 복합 단어를 컬럼명으로 사용할 때는 미리 정의된 표준을 따른다.
④ 실제 테이블을 검증하기 위해 표본 데이터를 입력시킨다.

▶ 정답 : 22. ④ 23. ③ 24. ④ 25. ② 26. ② 27. ① 28. ② 29. ② 30. ①

예·상·문·제·은·행·해·설

1. Section 085

물리적 설계(데이터 구조화)

- 논리적 설계 단계에서 논리적 구조로 표현된 데이터를 디스크 등의 물리적 저장장치에 저장할 수 있는 물리적 구조의 데이터로 변환하는 과정이다.
- 물리적 설계 단계에서는 다양한 데이터베이스 응용에 대해서 처리 성능을 얻기 위해 데이터베이스 파일의 저장 구조 및 액세스 경로를 결정한다.
- 레코드의 형식, 순서, 접근 경로와 같은 정보를 사용하여 데이터가 컴퓨터에 저장되는 방법을 묘사한다.
- 기본적인 데이터 단위는 저장 파일이다.
- 데이터베이스 시스템의 성능에 중대한 영향을 미친다.

2. Section 085

①번의 내용은 논리적 설계 단계에 대한 설명이다.

3. Section 085

수행될 질의를 안다는 것은 요구사항 분석 이후의 내용이므로 개념적 설계 단계에서 고민할 문제이다.

4. Section 086

테이블의 종류에는 일반 유형 테이블, 클러스터 인덱스 테이블, 파티셔닝 테이블, 외부 테이블, 임시 테이블 등이 있다.

5. Section 086

파티셔닝 테이블은 대용량의 테이블을 논리적 단위인 파티션(Partition)으로 나눈 테이블이다.

6. Section 087

트랜잭션의 특징 중 원자성은 트랜잭션의 연산이 데이터베이스에 모두 반영되든지 아니면 전혀 반영되지 않아야 한다는 성질이다. 즉 일부만 반영되어서는 안 된다는 것이다.

7. Section 087

트랜잭션(Transaction)이 가져야 할 특성으로는 원자성(Atomicity), 일관성(Consistency), 독립성(Isolation, 격리성), 영속성(Durability, 지속성)이 있다.

8. Section 087

A 은행에서 B라는 사람이 고객 인증 절차를 거쳐 잔액을 조회한 후, 타인에게 송금하는 도중에 장애가 발생하였을 경우 트랜잭션이 완벽히 수행되지 않았으므로 트랜잭션이 취소되어야 한다. 이러한 트랜잭션의 특징을 원자성(Atomicity)이라고 한다.

9. Section 087

트랜잭션 분석서의 구성 항목은 단위 프로세스, CRUD 연산, 테이블명, 컬럼명, 테이블 참조 횟수, 트랜잭션 수, 발생 주기 등이다.

10. Section 088

비트맵 인덱스는 데이터 종류가 적고 동일한 데이터가 많은 경우, 즉 분포도가 낮은 경우 최적의 성능을 발휘한다.

11. Section 088

인덱스와 테이블 데이터의 저장 공간은 분리되도록 설계해야 한다.

12. Section 089

뷰(View)는 독자적인 인덱스를 가질 수 없다.

13. Section 089

뷰(View)에서 삽입, 삭제, 갱신 연산이 가능하게 하려면 기본 테이블의 기본키를 포함한 속성 집합으로 뷰를 구성해야 한다.

14. Section 089

뷰(View)는 삽입, 삭제, 갱신 연산에는 제한이 있지만 검색은 기본 테이블 검색 연산과 동일하다.

15. Section 090

단일 테이블 클러스터링은 처리 범위가 넓은 경우, 다중 테이블 클러스터링은 조인이 많이 발생하는 경우 주로 사용된다.

16. Section 091

파티션을 수행하면 하나의 테이블이 분리되어 저장되므로 테이블간 조인에 대한 비용이 증가한다.

17. Section 091

해시 분할은 데이터를 고르게 분산하여 분할한다. 특정 파티션에 데이터가 집중되는 것은 범위 분할이다.

18. Section 093

분산 데이터베이스는 여러 곳에 분산되어 있는 데이터베이스를 하나의 데이터베이스처럼 사용할 수 있도록 만들어야 하기 때문에 중앙 집중식에 비해 설계와 개발이 어렵고 개발 비용이 많이 든다.

19. Section 093

분산 데이터베이스 관리 시스템의 목적 중 한 가지가 사용자들이 데이터가 어느 지역 데이터베이스에 위치하고 있는지를 알 수 없도록 하는 것인데, 이를 위치 투명성이라고 한다.

20. Section 093

분산 데이터베이스의 목표

- 위치 투명성(Location Transparency) : 액세스하려는 데이터베이스의 실제 위치를 알 필요 없이 단지 데이터베이스의 논리적인 명칭만으로 액세스할 수 있음
- 중복 투명성(Replication Transparency) : 동일 데이터가 여러 곳에 중복되어 있더라도 사용자는 마치 하나의 데이터만 존재하는 것처럼 사용하고, 시스템은 자동으로 여러 자료에 대한 작업을 수행함
- 병행 투명성(Concurrency Transparency) : 분산 데이터베이스와 관련된 다수의 트랜잭션들이 동시에 실현되더라도 그 트랜잭션의 결과는 영향을 받지 않음
- 장애 투명성(Failure Transparency) : 트랜잭션, DBMS, 네트워크, 컴퓨터 장애에도 불구하고 트랜잭션을 정확하게 처리함

21. Section 093

분산 데이터베이스의 목적

- 원격 자원과 데이터를 사용함으로써 사용자의 생산성 향상과 시스템 가용성 증대
- 지역적 정보 처리의 효율성 증진
- 시스템의 확장성과 변경 용이성
- 신뢰도 증진

22. Section 093

분산 데이터베이스 시스템의 구성 요소

- 분산 처리기(Distributed Processors)
- 통신 네트워크(Communication Network)
- 분산 데이터베이스(Distributed Database)
- 분산 데이터베이스 관리 시스템(Distributed Database Management System)

23. Section 094

- 고가용성 클러스터링 : 하나의 서버에 장애가 발생하면 다른 노드(서버)가 받아 처리하여 서비스 중단을 방지하는 방식으로, 일반적으로 말하는 클러스터링이 바로 고가용성 클러스터링
- 병렬 처리 클러스터링 : 전체 처리율을 높이기 위해 하나의 작업을 여러 개의 서버에서 분산하여 처리하는 방식

24. Section 095

보안 기법의 종류

- 대조 확인(Authentication) : 패스워드, 키(key), 목소리나 지문으로 대조 확인
- 권한 부여 규정 : 사용자 프로필(User Profile)의 유지
- 암호화 : 암호키와 해독키 사용

※ 시스템 체크 포인트는 장애 발생 시 회복을 위해 갱신 내용이나 시스템 상황 등에 관한 정보를 보관하는 지점이다.

25. Section 096

접근통제 정책

- 신분 기반 정책 : 주체나 그룹의 신분에 근거하여 객체의 접근을 제한하는 방법
- 규칙 기반 정책 : 주체가 갖는 권한에 근거하여 객체의 접근을 제한하는 방법
- 역할 기반 정책 : 주체가 조직 내에서 맡은 역할에 근거하여 객체의 접근을 제한하는 방법

26. Section 096

군대 시스템 등 특수 환경에서 주로 사용되는 모델은 기밀성 모델이다.

27. Section 098

- 스토리지의 종류에는 DAS, NAS, SAN이 있다.
- SSD는 하드디스크 드라이브(HDD)와 비슷하게 동작하면서 HDD와는 달리 기계적 장치가 없는 반도체를 이용하여 정보를 저장하는 장치이다.

28. Section 099

슈퍼타입/서브타입 데이터 모델을 통합된 테이블로 변환할 때는 방법에 상관없이 슈퍼타입과 서브타입의 모든 속성을 포함해야 한다.

30. Section 099

SQL의 예약어는 될 수 있으면 컬럼명으로 사용하지 않아야 한다.

3 장

SQL 응용

101 SQL의 개념 B 등급

102 DDL C 등급

103 DCL C 등급

104 DML B 등급

105 DML-SELECT-1 A 등급

106 DML-SELECT-2 A 등급

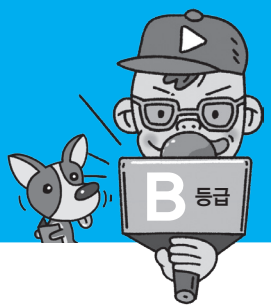
107 DML-JOIN B 등급

이 장에서 꼭 알아야 할 키워드 Best 10

1. SQL **2.** DDL **3.** DROP **4.** DCL **5.** GRANT **6.** DML **7.** DELETE **8.** SELECT **9.** DISTINCT **10.** JOIN

SECTION 101 ● LIVE

SQL의 개념



전문가의 조언

SQL은 관계형 데이터베이스의 표준 질의어로 정의, 조작, 제어 기능이 있습니다. 간단하게 개념을 숙지하세요.

질의어(Query Language)
질의어는 데이터베이스 파일과 범용 프로그래밍 언어를 정확히 알지 못하는 단말 사용자들이 단말기를 통해서 대화식으로 쉽게 DB를 이용할 수 있도록 되어 있는 비절차어의 일종입니다.

전문가의 조언

SQL은 사용 용도에 따라 DDL(데이터 정의어), DML(데이터 조작어), DCL(데이터 제어어)로 구분됩니다. 종류별로 어떠한 명령어들이 있는지 기억해 두세요. 각 명령어의 기능은 다음 섹션부터 배웁니다.

1 SQL(Structured Query Language)의 개요

- 1974년 IBM 연구소에서 개발한 SEQUEL에서 유래한다.
- 국제 표준 데이터베이스 언어이며, 많은 회사에서 관계형 데이터베이스(RDB)를 지원하는 언어로 채택하고 있다.
- 관계대수와 관계해석을 기초로 한 혼합 데이터 언어이다.
- 질의어*지만 질의 기능만 있는 것이 아니라 데이터 구조의 정의, 데이터 조작, 데이터 제어 기능을 모두 갖추고 있다.

2 SQL의 분류

DDL(Data Define Language, 데이터 정의어)

- DDL은 SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 정의하거나 변경 또는 삭제할 때 사용하는 언어이다.
- 논리적 데이터 구조와 물리적 데이터 구조의 사상을 정의한다.
- 데이터베이스 관리자나 데이터베이스 설계자가 사용한다.
- **DDL(데이터 정의어)의 세 가지 유형**

명령어	기능
CREATE	SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 정의한다.
ALTER	TABLE에 대한 정의를 변경하는 데 사용한다.
DROP	SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 삭제한다.

DML(Data Manipulation Language, 데이터 조작어)

- DML은 데이터베이스 사용자가 응용 프로그램이나 질의어를 통하여 저장된 데이터를 실질적으로 처리하는 데 사용되는 언어이다.
- 데이터베이스 사용자와 데이터베이스 관리 시스템 간의 인터페이스를 제공한다.
- **DML(데이터 조작어)의 네 가지 유형**

명령어	기능
SELECT	테이블에서 조건에 맞는 튜플을 검색한다.
INSERT	테이블에 새로운 튜플을 삽입한다.
DELETE	테이블에서 조건에 맞는 튜플을 삭제한다.
UPDATE	테이블에서 조건에 맞는 튜플의 내용을 변경한다.

DCL(Data Control Language, 데이터 제어어)

- DCL은 데이터의 보안, 무결성, 회복, 병행 수행 제어 등을 정의하는 데 사용되는 언어이다.
- 데이터베이스 관리자가 데이터 관리를 목적으로 사용한다.
- **DCL(데이터 제어어)의 종류**

명령어	기능
COMMIT	명령에 의해 수행된 결과를 실제 물리적 디스크로 저장하고, 데이터베이스 조작 작업이 정상적으로 완료되었음을 관리자에게 알려준다.
ROLLBACK	데이터베이스 조작 작업이 비정상적으로 종료되었을 때 원래의 상태로 복구한다.
GRANT	데이터베이스 사용자에게 사용 권한을 부여한다.
REVOKE	데이터베이스 사용자의 사용 권한을 취소한다.

따라잡기

기출문제 따라잡기

Section 101

이전기출

1. SQL의 명령은 사용 용도에 따라 DDL, DML, DCL로 구분할 수 있다. 다음 명령 중 그 성격이 나머지 셋과 다른 것은?

① SELECT ② UPDATE
③ INSERT ④ GRANT

GRANT는 데이터 제어어(DCL)인데, 나머지는? 맞아요! 데이터 조작어(DML)입니다.

이전기출

2. 다음 SQL 명령 중 데이터 정의어(DDL)에 포함되지 않는 명령어는?

① CREATE ② ALTER
③ SELECT ④ DROP

데이터 정의어(DDL) 세 가지 기억하세요? CREATE, ALTER, DROP! SELECT는 데이터 조작어(DML)입니다.

이전기출

3. SQL 언어의 데이터 제어어(DCL)에 해당하는 것은?

① SELECT문 ② INSERT문
③ UPDATE문 ④ GRANT문

데이터 제어어(DCL)의 종류 네 가지 기억하나요? COMMIT, ROLLBACK, GRANT, REVOKE, 잊지마세요.

이전기출

4. 스키마, 도메인, 테이블, 뷰를 정의할 때 사용되는 SQL 문은?

① SELECT ② CREATE
③ UPDATE ④ MAKE

SQL 명령을 모두 암기하지 못했더라도 '정의하다', '생성하다'의 의미를 갖는 보기를 찾으면 되겠죠?

이전기출

5. SQL 명령어로 수행된 결과를 실제 물리적 디스크로 저장하는 SQL 명령은?

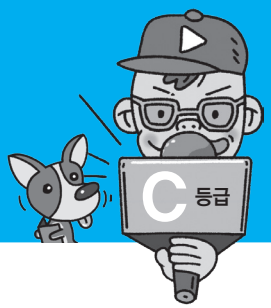
① ROLLBACK ② COMMIT
③ GRANT ④ REVOKE

데이터 제어어(DCL) 네 가지와 기능은 여기서 간단하게 정리하고 넘어가세요.

▶ 정답 : 1. ④ 2. ③ 3. ④ 4. ② 5. ②

SECTION 102 ● LIVE

DDL



전문가의 조언

DDL 구문을 모두 외울 필요는 없습니다. 각 명령어의 역할을 숙지하고, 구문을 보면 무엇을 의미하는지 이해할 수 있을 정도로만 정리해 두세요.

1 DDL(Data Define Language, 데이터 정의어)의 개념

DDL(데이터 정의어)는 DB 구조, 데이터 형식, 접근 방식 등 DB를 구축하거나 수정할 목적으로 사용하는 언어이다.

- DDL은 번역한 결과가 데이터 사전(Data Dictionary)이라는 특별한 파일에 여러 개의 테이블로서 저장된다.
- DDL에는 CREATE SCHEMA, CREATE DOMAIN, CREATE TABLE, CREATE VIEW, CREATE INDEX, ALTER TABLE, DROP 등이 있다.

2 CREATE SCHEMA

CREATE SCHEMA는 스키마*를 정의하는 명령문이다.

- 스키마의 식별을 위해 스키마 이름과 소유권자나 허가권자를 정의한다.

스키마(Schema)
스키마는 데이터베이스의 구조와 제약 조건에 관한 전반적인 명세(Specification)를 기술(Description)한 것으로 데이터 개체(Entity), 속성(Attribute), 관계(Relationship) 및 데이터 조작 시 데이터 값들이 갖는 제약 조건 등에 관해 전반적으로 정의합니다.

표기 형식

```
CREATE SCHEMA 스키마명 AUTHORIZATION 사용자_id;
```

예제 소유권자의 사용자 ID가 '홍길동'인 스키마 '대학교'를 정의하는 SQL문은 다음과 같다.

```
CREATE SCHEMA 대학교 AUTHORIZATION 홍길동;
```

3 CREATE DOMAIN

CREATE DOMAIN은 도메인*을 정의하는 명령문이다.

- 임의의 속성에서 취할 수 있는 값의 범위가 SQL에서 지원하는 전체 데이터 타입의 값이 아니고 일부분일 때, 사용자는 그 값의 범위를 도메인으로 정의할 수 있다.
- 정의된 도메인명은 일반적인 데이터 타입처럼 사용한다.

도메인(Domain)
도메인이란 하나의 속성이 취할 수 있는 동일한 유형의 원자값들의 집합을 의미합니다.
예를 들어 학년 속성의 데이터 타입이 정수형이고 해당 속성에서 취할 수 있는 값의 범위가 1~4까지라면, 1~4라는 범위는 해당 속성에 지정된 정수형의 모든 범위가 아니라 일부분이므로 사용자는 1~4까지의 범위를 해당 속성의 도메인으로 정의해서 사용할 수 있다는 의미입니다. 쉽게 말하면 도메인은 특정 속성에서 사용할 데이터의 범위를 사용자가 정의하는 사용자 정의 데이터 타입입니다.

구문에서 대괄호([ ])의 의미
SQL문에서 [AS] 처럼 대괄호로 묶은 명령어들은 생략이 가능하다는 의미입니다.

표기 형식

```
CREATE DOMAIN 도메인명 [AS]* 데이터_타입
        [DEFAULT 기본값]
        [CONSTRAINT 제약조건명 CHECK (범위값)];
```

- **데이터 타입** : SQL에서 지원하는 데이터 타입
- **기본값** : 데이터를 입력하지 않았을 때 자동으로 입력되는 값

예제 '성별'을 '남' 또는 '여'와 같이 정해진 1개의 문자로 표현되는 도메인 SEX를 정의하는 SQL문은 다음과 같다.

```
CREATE DOMAIN SEX CHAR(1)
    DEFAULT '남'
    CONSTRAINT VALID-SEX CHECK(VALUE IN ('남', '여'));
```

정의된 도메인은 이름이 'SEX'이며, 문자형이고 크기는 1자이다.

도메인 SEX를 지정한 속성의 기본값은 '남'이다.

SEX를 지정한 속성에는 '남', '여' 중 하나의 값만을 저장할 수 있다.

참깐만요

SQL에서 지원하는 기본 데이터 타입

- **정수(Integer)** : INTEGER(4Byte 정수), SMALLINT(2Byte 정수)
- **실수(Float)** : FLOAT, REAL, DOUBLE PRECISION
- **형식화된 숫자** : DEC(i, j), 단 i : 전체 자릿수, j : 소수부 자릿수
- **고정길이 문자** : CHAR(n), CHARACTER(n), 단 n : 문자수
- **가변길이 문자** : VARCHAR(n), CHARACTER VARYING(n), 단 n : 최대 문자수
- **고정길이 비트열(Bit String)** : BIT(n)
- **가변길이 비트열** : VARBIT(n)
- **날짜** : DATE
- **시간** : TIME

4 CREATE TABLE

CREATE TABLE은 테이블*을 정의하는 명령문이다.

표기 형식

```
CREATE TABLE 테이블명
        (속성명 데이터_타입 [DEFAULT 기본값] [NOT NULL], …
        [, PRIMARY KEY(기본키_속성명, …)]
        [, UNIQUE(대체키_속성명, …)]
        [, FOREIGN KEY(외래키_속성명, …)]
                REFERENCES 참조테이블(기본키_속성명, …)]
                [ON DELETE 옵션]
                [ON UPDATE 옵션]
        [, CONSTRAINT 제약조건명] [CHECK (조건식)]);
```

- 기본 테이블에 포함될 모든 속성에 대하여 속성명과 그 속성의 데이터 타입, 기본값, NOT NULL* 여부를 지정한다.
- **PRIMARY KEY** : 기본키로 사용할 속성 또는 속성의 집합을 지정한다.

테이블(Table)
테이블은 데이터베이스의 설계 단계에서는 테이블을 주로 릴레이션(Relation)이라 부르고, 조작이나 검색 시에는 테이블이라고 부릅니다. 그러나 대부분은 테이블과 릴레이션을 구분 없이 사용하니 두 의미가 같다는 것만 알아두세요.

NOT NULL
NULL이란 모르는 값 또는 적용할 수 없는 값을 의미하는 것으로, NOT NULL은 특정 속성이 데이터 없이 비어 있어서는 안 된다는 것을 지정할 때 사용합니다.

- UNIQUE : 대체키로 사용할 속성 또는 속성의 집합을 지정하는 것으로 UNIQUE로 지정한 속성은 중복된 값을 가질 수 없다.
- FOREIGN KEY ~ REFERENCES ~
  - 참조할 다른 테이블과 그 테이블을 참조할 때 사용할 외래키 속성을 지정한다.
  - 외래키가 지정되면 참조 무결성의 CASCADE 법칙*이 적용된다.
  - ON DELETE 옵션 : 참조 테이블의 튜플이 삭제되었을 때 기본 테이블에 취해야 할 사항을 지정한다. 옵션에는 NO ACTION, CASCADE, SET NULL, SET DEFAULT가 있다.
  - ON UPDATE 옵션 : 참조 테이블의 참조 속성 값이 변경되었을 때 기본 테이블에 취해야 할 사항을 지정한다. 옵션에는 NO ACTION, CASCADE, SET NULL, SET DEFAULT가 있다.
    - ▶ NO ACTION : 참조 테이블에 변화가 있어도 기본 테이블에는 아무런 조취를 취하지 않는다.
    - ▶ CASCADE : 참조 테이블의 튜플이 삭제되면 기본 테이블의 관련 튜플도 모두 삭제되고, 속성이 변경되면 관련 튜플의 속성 값도 모두 변경된다.
    - ▶ SET NULL : 참조 테이블에 변화가 있으면 기본 테이블의 관련 튜플의 속성 값을 NULL로 변경한다.
    - ▶ SET DEFAULT : 참조 테이블에 변화가 있으면 기본 테이블의 관련 튜플의 속성 값을 기본값으로 변경한다.
- CONSTRAINT : 제약 조건의 이름을 지정한다. 이름을 지정할 필요가 없으면 CHECK절만 사용하여 속성 값에 대한 제약 조건을 명시한다.
- CHECK : 속성 값에 대한 제약 조건을 정의한다.

참조 무결성의 CASCADE 법칙
참조 무결성 제약이 설정된 기본 테이블의 어떤 데이터를 삭제할 경우, 그 데이터와 밀접하게 연관되어 있는 다른 테이블의 데이터들도 도미노처럼 자동으로 삭제됩니다. 이러한 법칙을 '계단식', '연속성'이라는 사전적 의미를 가진 CASCADE 법칙이라고 합니다.

예제 '이름', '학번', '전공', '성별', '생년월일'로 구성된 〈학생〉 테이블을 정의하는 SQL 문을 작성하시오. 단, 제약 조건은 다음과 같다.

- '이름'은 NULL이 올 수 없고, '학번'은 기본키이다.
- '전공'은 〈학과〉 테이블의 '학과코드'를 참조하는 외래키로 사용된다.
- 〈학과〉 테이블에서 삭제가 일어나면 관련된 튜플들의 전공 값을 NULL로 만든다.
- 〈학과〉 테이블에서 '학과코드'가 변경되면 전공 값도 같은 값으로 변경한다.
- '생년월일'은 1980-01-01 이후의 데이터만 저장할 수 있다.
- 제약 조건의 이름은 '생년월일제약'으로 한다.
- 각 속성의 데이터 타입은 적당하게 지정한다. 단 '성별'은 도메인 'SEX'를 사용한다.

CREATE TABLE 학생	〈학생〉 테이블을 생성한다.
(이름 VARCHAR(15) NOT NULL,	'이름' 속성은 최대 문자 15자로 NULL 값을 갖지 않는다.
학번 CHAR(8),	'학번' 속성은 문자 8자이다.
전공 CHAR(5),	'전공' 속성은 문자 5자이다.
성별 SEX,	'성별' 속성은 'SEX' 도메인을 자료형으로 사용한다.
생년월일 DATE,	'생년월일' 속성은 DATE 자료형을 갖는다.
PRIMARY KEY(학번),	'학번'을 기본키로 정의한다.
FOREIGN KEY(전공) REFERENCES 학과(학과코드)	'전공' 속성은 〈학과〉 테이블의 '학과코드' 속성을 참조하는 외래키이다.
ON DELETE SET NULL	〈학과〉 테이블에서 튜플이 삭제되면 관련된 모든 튜플의 '전공' 속성의 값을 NULL로 변경한다.
ON UPDATE CASCADE,	〈학과〉 테이블에서 '학과코드'가 변경되면 관련된 모든 튜플의 '전공' 속성의 값도 같은 값으로 변경한다.
CONSTRAINT 생년월일제약 CHECK(생년월일)='1980-01-01'));	'생년월일' 속성에는 1980-01-01 이후의 값만을 저장할 수 있으며, 이 제약 조건의 이름은 '생년월일제약'이다.

CHAR과 VARCHAR
CHAR은 항상 지정된 크기만큼 기억 장소가 확보되고, VARCHAR은 기억 장소의 크기가 지정되어도 필드에 저장된 데이터만큼만 기억 장소가 확보됩니다. 예를 들어 '이름' 속성의 자료형을 CHAR(15)로 지정하면 '이름'에 한 글자가 저장되어도 항상 15바이트가 기억 장소로 확보되지만, VARCHAR(15)로 지정하면 저장된 한 글자 크기만큼만 기억 장소가 확보됩니다.

잠깐만요

다른 테이블을 이용한 테이블 정의

기존 테이블의 정보를 이용해 새로운 테이블을 정의할 수 있습니다.

표기 형식

CREATE TABLE 신규테이블명 AS SELECT 속성명[, 속성명, …] FROM 기존테이블명;

- 기존 테이블에서 추출되는 속성의 데이터 타입과 길이는 신규 테이블에 그대로 적용됩니다.
- 기존 테이블의 NOT NULL의 정의는 신규 테이블에 그대로 적용됩니다.
- 기존 테이블의 제약 조건은 신규 테이블에 적용되지 않으므로 신규 테이블을 정의한 후 ALTER TABLE 명령을 이용해 제약 조건을 추가해야 합니다.
- 기존 테이블의 일부 속성만 신규 테이블로 생성할 수 있으며, 기존 테이블의 모든 속성을 신규 테이블로 생성할 때는 속성명 부분에 '*'를 입력합니다.

예제 〈학생〉 테이블의 '학번', '이름', '학년' 속성을 이용하여 〈재학생〉 테이블을 정의하는 SQL문을 작성하시오.

```
CREATE TABLE 재학생 AS SELECT 학번, 이름, 학년 FROM 학생;
```

뷰(View)
뷰는 하나 이상의 기본 테이블로부터 유도되는 이름을 갖는 가상 테이블(Virtual Table)입니다. 테이블은 물리적으로 구현되어 실제로 데이터가 저장되지만, 뷰는 물리적으로 구현되지 않습니다. 즉 뷰를 생성하면 뷰 정의가 시스템 내에 저장되었다가 SQL 내에서 뷰 이름을 사용하면 실행 시간에 뷰 정의가 대체되어 수행됩니다.

SELECT
SELECT, ORDER BY, UNION 등은 Section 106과 107에서 자세하게 배웁니다.

서브 쿼리(Sub Query)
서브 쿼리는 조건절에 주어진 질의로서, 상위 질의에 앞서 실행되며 그 검색 결과는 상위 질의의 조건절의 피연산자로 사용됩니다.

인덱스(Index)
인덱스는 검색 시간을 단축시키기 위해 만든 보조적인 데이터 구조입니다.

전문가의 조언

대괄호([ ])는 생략할 수 있다는 것을 표시하고, 대괄호 안의 '|'는 선택을 의미합니다. 즉 'ASC | DESC'는 생략이 가능하지만, 생략하지 않을 경우에는 'ASC'와 'DESC' 중에서 하나만 선택할 수 있다는 의미입니다.

클러스터드 인덱스(Clustered Index)
인덱스 키의 순서에 따라 데이터가 정렬되어 저장되는 방식입니다. 실제 데이터가 순서대로 저장되어 있어 인덱스를 검색하지 않아도 원하는 데이터를 빠르게 찾을 수 있습니다. 하지만 데이터 삽입, 삭제 발생 시 순서를 유지하기 위해 데이터를 재정렬해야 합니다.

넌 클러스터드 인덱스(Non-Clustered Index)
인덱스의 키 값만 정렬되어 있을 뿐 실제 데이터는 정렬되지 않는 방식입니다. 데이터를 검색하기 위해서는 먼저 인덱스를 검색하여 실제 데이터의 위치를 확인해야 하므로 클러스터드 인덱스에 비해 검색 속도가 떨어집니다.

5 CREATE VIEW

CREATE VIEW는 뷰(View)*를 정의하는 명령문이다.

표기 형식

```
CREATE VIEW 뷰명[(속성명[, 속성명, …])]
AS SELECT문;
```

- SELECT*문을 서브 쿼리*로 사용하여 SELECT문의 결과로서 뷰를 생성한다.
- 서브 쿼리인 SELECT문에는 UNION*이나 ORDER BY*절을 사용할 수 없다.
- 속성명을 기술하지 않으면 SELECT문의 속성명이 자동으로 사용된다.

예제 〈고객〉 테이블에서 '주소'가 '안산시'인 고객들의 '성명'과 '전화번호'를 '안산고객'이라는 뷰로 정의하시오.

```
CREATE VIEW 안산고객(성명, 전화번호)
AS SELECT 성명, 전화번호
FROM 고객
WHERE 주소 = '안산시';
```

6 CREATE INDEX

CREATE INDEX는 인덱스*를 정의하는 명령문이다.

표기 형식

```
CREATE [UNIQUE] INDEX 인덱스명
ON 테이블명(속성명 [ASC | DESC]* [,속성명 [ASC | DESC]])
[CLUSTER];
```

- UNIQUE
  - 사용된 경우 : 중복 값이 없는 속성으로 인덱스를 생성한다.
  - 생략된 경우 : 중복 값을 허용하는 속성으로 인덱스를 생성한다.
- **정렬 여부 지정**
  - ASC : 오름차순 정렬
  - DESC : 내림차순 정렬
  - 생략된 경우 : 오름차순으로 정렬됨
- CLUSTER : 사용하면 인덱스가 클러스터드 인덱스로 설정됨*

예제 〈고객〉 테이블에서 UNIQUE한 특성을 갖는 '고객번호' 속성에 대해 내림차순으로 정렬하여 '고객번호_idx'라는 이름으로 인덱스를 정의하시오.

```
CREATE UNIQUE INDEX 고객번호_idx
ON 고객(고객번호 DESC);
```

7 ALTER TABLE

ALTER TABLE은 테이블에 대한 정의를 변경하는 명령문이다.

표기 형식

```
ALTER TABLE 테이블명 ADD 속성명 데이터_타입 [DEFAULT '기본값';
ALTER TABLE 테이블명 ALTER 속성명 [SET DEFAULT '기본값';
ALTER TABLE 테이블명 DROP COLUMN 속성명 [CASCADE];
```

- ADD : 새로운 속성(열)을 추가할 때 사용한다.
- ALTER : 특정 속성의 Default 값을 변경할 때 사용한다.
- DROP COLUMN : 특정 속성을 삭제할 때 사용한다.

예제 1 〈학생〉 테이블에 최대 3문자로 구성되는 '학년' 속성 추가하시오.

```
ALTER TABLE 학생 ADD 학년 VARCHAR(3);
```

예제 2 〈학생〉 테이블의 '학번' 필드의 데이터 타입과 크기를 VARCHAR(10)으로 하고 NULL 값이 입력되지 않도록 변경하시오.

```
ALTER TABLE 학생 ALTER 학번 VARCHAR(10) NOT NULL;
```

8 DROP

DROP은 스키마, 도메인, 기본 테이블, 뷰 테이블, 인덱스, 제약 조건 등을 제거하는 명령문이다.

표기 형식

```
DROP SCHEMA 스키마명 [CASCADE | RESTRICTED];
DROP DOMAIN 도메인명 [CASCADE | RESTRICTED];
DROP TABLE 테이블명 [CASCADE | RESTRICTED];
DROP VIEW 뷰명 [CASCADE | RESTRICTED];
DROP INDEX 인덱스명 [CASCADE | RESTRICTED];
DROP CONSTRAINT 제약조건명;
```

- DROP SCHEMA : 스키마를 제거한다.
- DROP DOMAIN : 도메인을 제거한다.
- DROP TABLE : 테이블을 제거한다.
- DROP VIEW : 뷰를 제거한다.
- DROP INDEX : 인덱스를 제거한다.
- DROP CONSTRAINT : 제약 조건을 제거한다.

전문가의 조언

DDL에서는 DROP 명령이 중요합니다. DROP 명령의 표기 형식을 기억하고, 특히 CASCADE와 RESTRICTED에 대해 정확하게 숙지하세요.

- CASCADE : 제거할 요소를 참조하는 다른 모든 개체를 함께 제거한다. 즉 주 테이블의 데이터 제거 시 각 외래키와 관계를 맺고 있는 모든 데이터를 제거하는 참조 무결성 제약 조건을 설정하기 위해 사용된다.
- RESTRICTED : 다른 개체가 제거할 요소를 참조중일 때는 제거를 취소한다.

예제 〈학생〉 테이블을 제거하되, 〈학생〉 테이블을 참조하는 모든 데이터를 함께 제거하시오.

```
DROP TABLE 학생 CASCADE;
```

따라잡기

기출문제 따라잡기 Section 102

이전기출

1. SQL 언어의 CREATE TABLE문에 포함될 수 없는 것은?

① 속성의 NOT NULL 제약 조건
② 속성의 타입 변경
③ 속성의 초기값 지정
④ CHECK 제약 조건의 정의

CREATE TABLE문은 기본 TABLE을 최초로 생성하기 위해 사용하는 명령문으로, 테이블을 변경하는 명령문은 사용할 수 없습니다.

이전기출

2. SQL 구문과 의미가 잘못 연결된 것은?

① CREATE – 테이블 생성
② DROP – 레코드 삭제
③ UPDATE – 자료 갱신
④ DESC – 내림차순 정렬

DROP은 테이블에 포함된 레코드를 삭제하는 것이 아니고 테이블 자체를 제거하는 것입니다. 아울러 DROP은 스키마, 도메인, 뷰, 인덱스의 제거에도 사용된다는 것도 알아두세요.

이전기출

3. 다음 SQL문의 실행 결과를 가장 올바르게 설명한 것은?

Drop Table 인사 Cascade;

① 인사 테이블을 제거한다.
② 인사 테이블을 참조하는 테이블과 인사 테이블을 제거한다.
③ 인사 테이블이 참조중이면 제거하지 않는다.
④ 인사 테이블을 제거할지의 여부를 사용자에게 다시 질의한다.

CASCADE는 삭제할 개체를 참조하고 있는 다른 개체도 모두 제거하는 옵션입니다. 다른 개체가 참조하고 있으면 삭제가 취소되는 옵션은 RESTRICTED입니다.

이전기출

4. SQL의 DROP문에 관한 설명 중 잘못된 것은?

① 해당 TABLE에 삽입된 TUPLE들도 없어진다.
② 해당 TABLE에 대해 만들어진 INDEX가 없어진다.
③ 해당 TABLE에 대해 만들어진 VIEW가 없어진다.
④ 해당 TABLE에 참조 관계가 있는 TABLE이 없어진다.

DROP 명령은 삭제하려는 테이블과 함께 그 테이블로부터 유도하여 만든 INDEX, VIEW도 모두 제거합니다. 그러나 참조하던 테이블은 해당 테이블로부터 유도된 테이블이 아니기 때문에 삭제되지 않습니다. 단, CASCADE 옵션을 사용하면 참조된 테이블도 삭제됩니다.

이전기출

5. 관계 데이터베이스에서 Main Table의 데이터를 삭제 시 각 외래키에 대해 부합되는 모든 데이터를 삭제하는 참조 무결성의 법칙은?

① RESTRICTED ② CASCADE
③ SET NULL ④ CUSTOMIZED

RESTRICTED와 CASCADE의 용도만 구분할 수 있다면 쉽게 풀 수 있는 문제입니다.

이전기출

6. 인사 테이블의 주소 필드에 대한 데이터 타입을 VARCHAR(10)으로 정의하였으나, 필드 길이가 부족하여 20바이트로 확장하고자 한다. 이에 적합한 SQL 명령은?

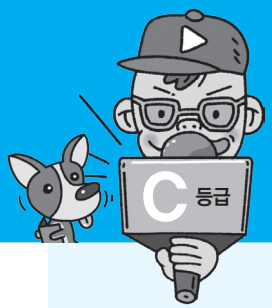
① MODIFY FIELD ② MODIFY TABLE
③ ALTER TABLE ④ ADD TABLE

테이블 생성은 CREATE, 테이블 삭제는 DROP, 테이블 수정은 ALTER! 잊지마세요.

▶ 정답 : 1. ② 2. ② 3. ② 4. ④ 5. ② 6. ③

SECTION 103 LIVE

DCL



1 DCL(Data Control Language, 데이터 제어어)의 개념

DCL(데이터 제어어)는 데이터의 보안, 무결성, 회복, 병행 제어 등을 정의하는 데 사용하는 언어이다.

- DCL은 데이터베이스 관리자(DBA)가 데이터 관리를 목적으로 사용한다.
- DCL에는 GRANT, REVOKE, COMMIT, ROLLBACK, SAVEPOINT 등이 있다.

2 GRANT / REVOKE

데이터베이스 관리자가 데이터베이스 사용자에게 권한을 부여하거나 취소하기 위한 명령어이다.

- GRANT : 권한 부여를 위한 명령어
- REVOKE : 권한 취소를 위한 명령어
- **사용자등급* 지정 및 해제**

– GRANT 사용자등급 TO 사용자_ID_리스트 [IDENTIFIED BY 암호];
– REVOKE 사용자등급 FROM 사용자_ID_리스트;

예제 1 사용자 ID가 "NABI"인 사람에게 데이터베이스 및 테이블을 생성할 수 있는 권한을 부여하는 SQL문을 작성하시오.

```
GRANT RESOURCE TO NABI;
```

예제 2 사용자 ID가 "STAR"인 사람에게 단순히 데이터베이스에 있는 정보를 검색할 수 있는 권한을 부여하는 SQL문을 작성하시오.

```
GRANT CONNECT TO STAR;
```

- **테이블 및 속성에 대한 권한 부여 및 취소**

– GRANT 권한_리스트 ON 개체 TO 사용자 [WITH GRANT OPTION];
– REVOKE [GRANT OPTION FOR] 권한_리스트 ON 개체 FROM 사용자 [CASCADE];

– 권한 종류 : ALL, SELECT, INSERT, DELETE, UPDATE, ALTER 등

– WITH GRANT OPTION : 부여받은 권한을 다른 사용자에게 다시 부여할 수 있는 권한을 부여함

전문가의 조언

DCL은 일반 사용자보다는 데이터베이스 관리자가 사용하는 명령입니다. GRANT, REVOKE, COMMIT, ROLLBACK, SAVEPOINT의 기능과 사용법을 숙지하세요.

사용자등급

- DBA : 데이터베이스 관리자
- RESOURCE : 데이터베이스 및 테이블 생성 가능자
- CONNECT : 단순 사용자

– GRANT OPTION FOR : 다른 사용자에게 권한을 부여할 수 있는 권한을 취소함
– CASCADE : 권한 취소 시 권한을 부여받았던 사용자가 다른 사용자에게 부여한 권한도 연쇄적으로 취소함

예제 3 사용자 ID가 "NABI"인 사람에게 〈고객〉 테이블에 대한 모든 권한과 다른 사람에게 권한을 부여할 수 있는 권한까지 부여하는 SQL문을 작성하시오.

```
GRANT ALL ON 고객 TO NABI WITH GRANT OPTION;
```

예제 4 사용자 ID가 "STAR"인 사람에게 부여한 〈고객〉 테이블에 대한 권한 중 UPDATE 권한을 다른 사람에게 부여할 수 있는 권한만 취소하는 SQL문을 작성하시오.

```
REVOKE GRANT OPTION FOR UPDATE ON 고객 FROM STAR;
```

3 COMMIT

트랜잭션*이 성공적으로 끝나면 데이터베이스가 새로운 일관성(Consistency) 상태를 가지기 위해 변경된 모든 내용을 데이터베이스에 반영하여야 하는데, 이때 사용하는 명령이 COMMIT이다.

- COMMIT 명령을 실행하지 않아도* DML문이 성공적으로 완료되면 자동으로 COMMIT되고, DML이 실패하면 자동으로 ROLLBACK이 되도록 Auto Commit* 기능을 설정할 수 있다.

4 ROLLBACK

ROLLBACK은 아직 COMMIT되지 않은 변경된 모든 내용들을 취소하고 데이터베이스를 이전 상태로 되돌리는 명령어이다.

- 트랜잭션 전체가 성공적으로 끝나지 못하면 일부 변경된 내용만 데이터베이스에 반영되는 비일관성(Inconsistency)인 상태를 가질 수 있기 때문에 일부분만 완료된 트랜잭션은 롤백(Rollback)되어야 한다.

5 SAVEPOINT

SAVEPOINT는 트랜잭션 내에 ROLLBACK 할 위치인 저장점을 지정하는 명령어이다.

- 저장점을 지정할 때는 이름을 부여하며, ROLLBACK 시 지정된 저장점까지의 트랜잭션 처리 내용이 취소된다.

〈사원〉

사원번호	이름	부서
10	김기획	기획부
20	박인사	인사부
30	최재무	재무부
40	오영업	영업부

전문가의 조언

COMMIT, ROLLBACK, SAVEPOINT는 트랜잭션을 제어하는 용도로 사용되므로 TCL(Transaction Control Language)로 분류하기도 합니다. 하지만 기능을 제어하는 명령이라는 공통점으로 DCL의 일부로 분류하기도 합니다.

트랜잭션(Transaction)

- 트랜잭션은 데이터베이스에서 하나의 논리적 기능을 수행하기 위한 일련의 연산 집합으로서 작업의 단위가 됩니다.
- 트랜잭션은 데이터베이스 관리 시스템에서 회복 및 병행 제어 시에 처리되는 작업의 논리적 단위입니다.
- 하나의 트랜잭션은 COMMIT 되거나 ROLLBACK 되어야 합니다.

COMMIT 명령 사용 여부

트랜잭션이 시작되면 데이터베이스의 데이터를 주기억장치에 올려 처리하다가 COMMIT 명령이 내려지면 그때서야 처리된 내용을 보조기억장치에 저장합니다. 그러니까 COMMIT 명령을 사용하지 않고 DBMS를 종료하면 그때까지 작업했던 모든 내용이 보조기억장치의 데이터베이스에 하나도 반영되지 않고 종료되는 것이지요. 이처럼 실수로 COMMIT 명령 없이 DBMS를 종료하는 것에 대비하여 대부분의 DBMS들은 Auto Commit 기능을 제공하고 있습니다.

Auto Commit 설정 명령

- Oracle
  - 설정 : set autocommit on;
  - 해제 : set autocommit off;
  - 확인 : show autocommit;
- MySQL
  - 설정 : set autocommit = true;
  - 해제 : set autocommit = false;
  - 확인 : select @@autocommit;

전문가의 조언

COMMIT과 SAVEPOINT 명령의 수행 시점에 따라 ROLLBACK 명령이 적용되는 범위가 달라집니다. 이와 같이 COMMIT, ROLLBACK, SAVEPOINT 명령은 서로 연관되어 사용되므로 한꺼번에 실습하여 결과를 확인할 수 있도록 하였습니다.

예제 1 〈사원〉 테이블에서 '사원번호'가 40인 사원의 정보를 삭제한 후 COMMIT을 수행하시오.

```
DELETE * FROM 사원 WHERE 사원번호 = 40;
COMMIT;
```

해설

DELETE 명령을 수행한 후 COMMIT 명령을 수행했으므로 DELETE 명령으로 삭제된 레코드는 이후 ROLLBACK 명령으로 되돌릴 수 없다.

〈사원〉 테이블 상태

사원번호	이름	부서
10	김기획	기획부
20	박인사	인사부
30	최재무	재무부

예제 2 '사원번호'가 30인 사원의 정보를 삭제하시오.

```
DELETE * FROM 사원 WHERE 사원번호 = 30;
```

해설

DELETE 명령을 수행한 후 COMMIT 명령을 수행하지 않았으므로 DELETE 명령으로 삭제된 레코드는 이후 ROLLBACK 명령으로 되돌릴 수 있다.

〈사원〉 테이블 상태

사원번호	이름	부서
10	김기획	기획부
20	박인사	인사부

예제 3 SAVEPOINT 'S1'을 설정하고 '사원번호'가 20인 사원의 정보를 삭제하시오.

```
SAVEPOINT S1;
DELETE * FROM 사원 WHERE 사원번호 = 20;
```

해설

〈사원〉 테이블 상태

사원번호	이름	부서
10	김기획	기획부

예제 4 SAVEPOINT 'S2'를 설정하고 '사원번호'가 10인 사원의 정보를 삭제하시오.

```
SAVEPOINT S2;
DELETE * FROM 사원 WHERE 사원번호 = 10;
```

해설

〈사원〉 테이블 상태

사원번호	이름	부서

DELETE문

DELETE문은 다음 섹션에서 자세히 공부합니다. 여기서는 DELETE문은 레코드를 삭제할 때 사용하는 명령어라는 것만 알아두세요.

예제 5 SAVEPOINT 'S2'까지 ROLLBACK을 수행하시오.

```
ROLLBACK TO S2;
```

해설

ROLLBACK이 적용되는 시점을 'S2'로 지정했기 때문에 예제 5의 ROLLBACK에 의해 〈사원〉 테이블의 상태는 예제 4의 작업을 수행하기 전으로 되돌려진다.

〈사원〉 테이블 상태

사원번호	이름	부서
10	김기획	기획부

예제 6 SAVEPOINT 'S1'까지 ROLLBACK을 수행하시오.

```
ROLLBACK TO S1;
```

해설

ROLLBACK이 적용되는 시점을 'S1'로 지정했기 때문에 예제 6의 ROLLBACK에 의해 〈사원〉 테이블의 상태는 예제 3의 작업을 수행하기 전으로 되돌려진다.

〈사원〉 테이블 상태

사원번호	이름	부서
10	김기획	기획부
20	박인사	인사부

예제 6 SAVEPOINT 없이 ROLLBACK을 수행하시오.

```
ROLLBACK;
```

해설

'사원번호'가 40인 사원의 정보를 삭제한 후 COMMIT을 수행했으므로 예제 7의 ROLLBACK이 적용되는 시점은 예제 1의 COMMIT 이후 새롭게 작업이 수행되는 예제 2의 작업부터이다.

〈사원〉 테이블 상태

사원번호	이름	부서
10	김기획	기획부
20	박인사	인사부
30	최재무	재무부

기출문제 따라잡기

Section 103

이전기출

1. SQL 언어의 데이터 제어어(DCL)에 해당하는 것은?

① SELECT문 ② INSERT문
③ UPDATE문 ④ GRANT문

데이터 제어어의 종류 기억하나요? COMMIT, ROLLBACK, GRANT, REVOKE, SAVEPOINT 잊지마세요.

이전기출

2. 한 작업의 논리적 단위가 성공적으로 끝났고, 데이터베이스가 다시 일관된 상태에 있으며, 이 트랜잭션이 행한 갱신 연산이 완료된 것을 트랜잭션 관리자에게 알려주는 연산은?

① Rollback 연산 ② Log 연산
③ Commit 연산 ④ Backup 연산

Commit은 작업의 성공, Rollback은 작업의 실패로 취소하는 연산입니다.

이전기출

3. 트랜잭션의 실행이 실패하였음을 알리는 연산으로 트랜잭션이 수행한 결과를 원래의 상태로 원상 복귀시키는 연산은?

① COMMIT 연산 ② BACKUP 연산
③ LOG 연산 ④ ROLLBACK 연산

Commit과 Rollback을 확실히 구분해서 기억하세요.

이전기출

4. 다음과 같은 일련의 권한 부여 SQL 명령어에 대한 설명 중 부적합한 것은?

```
DBA : GRANT SELECT ON STUDENT TO U1 WITH
      GRANT OPTION
U1 : GRANT SELECT ON STUDENT TO U2
DBA : REVOKE SELECT ON STUDENT FROM U1
      CASCADE;
```

① U1은 Student에 대한 검색 권한이 없다.
② DBA는 Student에 대한 검색 권한이 있다.
③ U2는 Student에 대한 검색 권한이 있다.
④ U2는 Student에 대한 검색 권한을 다른 사용자에게 부여할 수 없다.

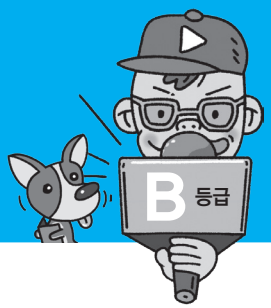
하나씩 살펴볼까요?

DBA : U1에게 STUDENT에 대한 검색 권한과 다른 사람에게 STUDENT를 검색할 수 있는 권한을 부여할 수 있는 권한을 줍니다.
U1 : U2에게 STUDENT를 검색할 수 있는 권한을 부여합니다.
DBA : U1에게서 STUDENT에 대한 검색 권한을 취소합니다. CASCADE 옵션이 있으므로 U1이 다른 사람에게 부여했던 권한도 모두 취소됩니다.

▶ 정답 : 1. ④ 2. ③ 3. ④ 4. ③

SECTION 104 ● LIVE

DML



전문가의 조언

DML의 네 가지 유형 중에서 INSERT, DELETE, UPDATE 명령문을 학습합니다. 세 가지 유형의 구문은 예제를 통해 사용법까지 꼭 암기하세요. SELECT 명령은 다음 섹션에서 학습합니다.

1 DML(Data Manipulation Language, 데이터 조작어)의 개념

DML(데이터 조작어)은 데이터베이스 사용자가 응용 프로그램이나 질의어를 통해 저장된 데이터를 실질적으로 관리하는데 사용되는 언어이다.

- DML은 데이터베이스 사용자와 데이터베이스 관리 시스템 간의 인터페이스를 제공한다.
- DML의 유형

명령문	기능
SELECT	테이블에서 튜플을 검색한다.
INSERT	테이블에 새로운 튜플을 삽입한다.
DELETE	테이블에서 튜플을 삭제한다.
UPDATE	테이블에서 튜플의 내용을 갱신한다.

2 삽입문(INSERT INTO~)

삽입문은 기본 테이블에 새로운 튜플을 삽입할 때 사용한다.

일반 형식

```
INSERT INTO 테이블명([속성명1, 속성명2,…])
VALUES (데이터1, 데이터2,… );
```

- 대응하는 속성과 데이터는 개수와 데이터 유형이 일치해야 한다.
- 기본 테이블의 모든 속성을 사용할 때는 속성명을 생략할 수 있다.
- SELECT문을 사용하여 다른 테이블의 검색 결과를 삽입할 수 있다.

〈사원〉

이름	부서	생일	주소	기본급
홍길동	기획	04/05/61	망원동	120
임꺽정	인터넷	01/09/69	성산동	80
황진이	편집	07/21/75	연희동	100
김선달	편집	10/22/73	망원동	90
성춘향	기획	02/20/64	망원동	100
장길산	편집	03/11/67	상암동	120
일지매	기획	04/29/78	합정동	110
강호동	인터넷	12/11/80		90

예제 1 〈사원〉 테이블에 (이름 – 홍승현, 부서 – 인터넷)을 삽입하시오.

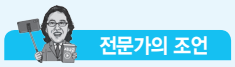
```
INSERT INTO 사원(이름, 부서) VALUES ('홍승현', '인터넷');
```

예제 2 〈사원〉 테이블에 (장보고, 기획, 05/03/73, 홍제동, 90)을 삽입하시오.

```
INSERT INTO 사원 VALUES ('장보고', '기획', #05/03/73#, '홍제동', 90);
```

예제 3 〈사원〉 테이블에 있는 편집부의 모든 튜플을 편집부원(이름, 생일, 주소, 기본급) 테이블에 삽입하시오.

```
INSERT INTO 편집부원(이름, 생일, 주소, 기본급)
SELECT 이름, 생일, 주소, 기본급
FROM 사원
WHERE 부서 = '편집';
```



전문가의 조언

날짜 데이터는 숫자로 취급하지만 ' ' 또는 # #으로 묶어줍니다.

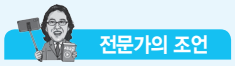
3 삭제문(DELETE FROM~)

삭제문은 기본 테이블에 있는 튜플들 중에서 특정 튜플(행)을 삭제할 때 사용한다.

일반 형식

```
DELETE
FROM 테이블명
[WHERE 조건];
```

- 모든 레코드를 삭제할 때는 WHERE절을 생략한다.
- 모든 레코드를 삭제하더라도 테이블 구조는 남아 있기 때문에 디스크에서 테이블을 완전히 제거하는 DROP과는 다르다.



전문가의 조언

DELETE문은 테이블 구조나 테이블 자체는 그대로 남겨 두고, 테이블 내의 튜플들만 삭제합니다. 테이블을 완전히 제거하기 위해서는 DROP문을 사용해야 합니다.

예제 1 〈사원〉 테이블에서 "임꺽정"에 대한 튜플을 삭제하시오.

```
DELETE
FROM 사원
WHERE 이름 = '임꺽정';
```

예제 2 〈사원〉 테이블에서 "인터넷" 부서에 대한 모든 튜플을 삭제하시오.

```
DELETE
FROM 사원
WHERE 부서 = '인터넷';
```

예제 3 〈사원〉 테이블의 모든 레코드를 삭제하시오.

```
DELETE
FROM 사원;
```

전문가의 조언

UPDATE~ SET~ WHERE입니다. 기억하세요.

4 갱신문(UPDATE~ SET~)

갱신문은 기본 테이블에 있는 튜플들 중에서 특정 튜플의 내용을 변경할 때 사용한다.

일반 형식

```
UPDATE 테이블명
SET 속성명 = 데이터[, 속성명=데이터, …]
[WHERE 조건];
```

예제 1 〈사원〉 테이블에서 "홍길동"의 '주소'를 "수색동"으로 수정하시오.

```
UPDATE 사원
SET 주소 = '수색동'
WHERE 이름 = '홍길동';
```

예제 2 〈사원〉 테이블에서 "황진이"의 '부서'를 "기획부"로 변경하고 '기본급'을 5만 원 인상시키시오.

```
UPDATE 사원
SET 부서 = '기획', 기본급 = 기본급 + 5
WHERE 이름 = '황진이';
```

잠깐만요

데이터 조작문의 네 가지 유형

- **SELECT(검색)** : SELECT~ FROM~ WHERE~
- **INSERT(삽입)** : INSERT INTO~ VALUES~
- **DELETE(삭제)** : DELETE~ FROM~ WHERE~
- **UPDATE(변경)** : UPDATE~ SET~ WHERE~

기출문제 따라잡기

Section 104

이전기출

1. SQL의 기술이 옳지 않은 것은?

① SELECT… FROM … WHERE…

② INSERT… ON… VALUES…

③ UPDATE… SET… WHERE…

④ DELETE… FROM… WHERE…

INSERT는 'INSERT INTO… VALUES'입니다. 나머지 세 가지도 확실하게 알아두세요.

이전기출

2. 다음 SQL 문장이 뜻하는 것은 무엇인가?

```
INSERT INTO 컴퓨터과테이블(학번, 이름, 학년)
SELECT 학번, 이름, 학년
FROM 학생테이블
WHERE 학과 = '컴퓨터';
```

① 학생테이블에서 학과가 컴퓨터인 사람의 학번, 이름, 학년을 검색하라.

② 학생테이블에 학과가 컴퓨터인 사람의 학번, 이름, 학년을 삽입하라.

③ 학생테이블에서 학과가 컴퓨터인 사람의 학번, 이름, 학년을 검색하여 컴퓨터과테이블에 삽입하라.

④ 컴퓨터과테이블에서 학과가 컴퓨터인 사람의 학번, 이름, 학년을 검색하여 학생테이블에 삽입하라.

하나씩 분해해서 살펴볼까요?

- **INSERT INTO 컴퓨터과테이블(학번, 이름, 학년)** : 컴퓨터과테이블의 학번, 이름, 학년 속성에 삽입하라.
- **SELECT 학번, 이름, 학년 FROM 학생테이블** : 학생테이블에서 학번, 이름, 학년 속성의 값을 검색하라.
- **WHERE 학과 = '컴퓨터'** : 학과 속성의 값이 '컴퓨터'인 튜플만을 대상으로 하라.

이전기출

3. SQL문에서 STUDENT(SNO, SNAME, YEAR, DEPT) 테이블에 학번 600, 성명 "홍길동", 학년 2학년인 학생 튜플을 삽입하는 명령으로 옳은 것은?(단, SNO는 학번, SNAME은 성명, YEAR는 학년, DEPT는 학생, 교수 구분 필드임)

① INSERT STUDENT INTO VALUES(600, '홍길동', 2);

② INSERT FROM STUDENT VALUES(600, '홍길동', 2);

③ INSERT INTO STUDENT(SNO, SNAME, YEAR) VALUES (600, '홍길동', 2);

④ INSERT TO STUDENT(SNO, SNAME, YEAR) VALUES(600, '홍길동', 2);

삽입문 기억하죠. INSERT INTO 테이블명(속성명) VALUES(값)

이전기출

4. SQL에서 DELETE 명령에 대한 설명으로 옳지 않은 것은?

① 테이블의 행을 삭제할 때 사용한다.

② WHERE 조건절이 없는 DELTE 명령을 수행하면 DROP TABLE 명령을 수행했을 때와 같은 효과를 얻을 수 있다.

③ SQL을 사용 용도에 따라 분류할 경우 DML에 해당한다.

④ 기본사용 형식은 "DELETE FROM 테이블 [WHERE 조건] ;"이다.

WHERE 조건절이 없는 DELETE 명령을 수행하면 테이블 내의 데이터만 모두 삭제됩니다. 테이블 구조까지 제거하는 DROP TABLE 명령과는 기능이 다릅니다.

이전기출

5. 학생 테이블에서 학번이 300인 학생의 학년을 3으로 수정하기 위한 SQL 질의어는?

```
학생(학번, 이름, 학년, 학과)
```

① UPDATE 학년=3 FROM 학생 WHERE 학번=300;

② UPDATE 학생 SET 학년=3 WHERE 학번=300;

③ UPDATE FROM 학생 SET 학년=3 WHERE 학번=300;

④ UPDATE 학년=3 SET 학생 WHEN 학번=300;

갱신문의 문법 기억하죠? 'UPDATE 테이블명 SET 속성명=데이터 WHERE 조건' 틀렸으면 다시 정리하고 넘어가세요.

▶ 정답 : 1. ② 2. ③ 3. ③ 4. ② 5. ②

SECTION 105 ● LIVE

DML – SELECT–1

전문가의 조언

- 실무에서 가장 많이 사용되는 SQL 명령이 SELECT입니다. 이번 섹션에서는 SELECT문의 일반 형식, 기본 검색, 조건 지정 검색, 정렬 검색, 하위 질의, 복수 테이블 질의에 대해 학습합니다. 각 절, 옵션의 기능까지 정확히 암기해 두세요. 나머지 중요한 요소는 그때그때 설명하겠습니다.
- SELECT문의 일반 형식에 포함된 내용이 길어 학습할 분량을 두 섹션으로 분리하였습니다. 흐리게 처리된 형식은 다음 섹션에서 학습할 내용입니다.

1 일반 형식

```
SELECT [PREDICATE] [테이블명.]속성명 [AS 별칭][, [테이블명.]속성명, …]
[, 그룹함수(속성명) [AS 별칭]]
[, Window함수 OVER (PARTITION BY 속성명1, 속성명2, …
                    ORDER BY 속성명3, 속성명4, …)]
FROM 테이블명[, 테이블명, …]
[WHERE 조건]
[GROUP BY 속성명, 속성명, …]
[HAVING 조건]
[ORDER BY 속성명 [ASC | DESC]];
```

- **SELECT절**
  - PREDICATE : 불러올 튜플 수를 제한할 명령어를 기술한다.
    - ▶ ALL : 모든 튜플을 검색할 때 지정하는 것으로, 주로 생략한다.
    - ▶ DISTINCT : 중복된 튜플이 있으면 그 중 첫 번째 한 개만 검색한다.
    - ▶ DISTINCTROW : 중복된 튜플을 제거하고 한 개만 검색하지만 선택된 속성의 값이 아닌, 튜플 전체를 대상으로 한다.
  - 속성명 : 검색하여 불러올 속성(열) 또는 속성을 이용한 수식을 지정한다.
    - ▶ 기본 테이블을 구성하는 모든 속성을 지정할 때는 '*'를 기술한다.
    - ▶ 두 개 이상의 테이블을 대상으로 검색할 때는 '테이블명.속성명'으로 표현한다.
  - AS : 속성 및 연산의 이름을 다른 제목으로 표시하기 위해 사용된다.
- **FROM절** : 질의에 의해 검색될 데이터들을 포함하는 테이블명을 기술한다.
- **WHERE절** : 검색할 조건을 기술한다.
- **ORDER BY절** : 특정 속성을 기준으로 정렬하여 검색할 때 사용한다.
  - 속성명 : 정렬의 기준이 되는 속성명을 기술한다.
  - [ASC | DESC] : 정렬 방식으로서 'ASC'는 오름차순, 'DESC'는 내림차순이다. 생략하면 오름차순으로 지정된다.

잠깐만요

조건 연산자 / 연산자 우선순위

조건 연산자

- 비교 연산자

연산자	=	〈〉	〉	〈	〉=	〈=
의미	같다	같지 않다	크다	작다	크거나 같다	작거나 같다

- **논리 연산자** : NOT, AND, OR
- **LIKE 연산자** : 대표 문자를 이용해 지정된 속성의 값이 문자 패턴과 일치하는 튜플을 검색하기 위해 사용됩니다.

대표 문자	%	_	#
의미	모든 문자를 대표함	문자 하나를 대표함	숫자 하나를 대표함

연산자 우선순위

종류	연산자	우선순위
산술 연산자	×, /, +, −	왼쪽에서 오른쪽으로 갈수록 낮아집니다.
관계 연산자	=, 〈〉, 〉, 〉=, 〈, 〈=	모두 같습니다.
논리 연산자	NOT, AND, OR	왼쪽에서 오른쪽으로 갈수록 낮아집니다.

※ 산술, 관계, 논리 연산자가 함께 사용되었을 때는 산술 〉 관계 〉 논리 연산자 순서로 연산자 우선순위가 정해집니다.

다음과 같은 기본 테이블에 대해 다음 예제의 결과를 확인하시오.

〈사원〉

이름	부서	생일	주소	기본급
홍길동	기획	04/05/61	망원동	120
임꺽정	인터넷	01/09/69	서교동	80
황진이	편집	07/21/75	합정동	100
김선달	편집	10/22/73	망원동	90
성춘향	기획	02/20/64	대흥동	100
장길산	편집	03/11/67	상암동	120
일지매	기획	04/29/78	연남동	110
강건달	인터넷	12/11/80		90

〈여가활동〉

이름	취미	경력
김선달	당구	10
성춘향	나이트댄스	5
일지매	태껸	15
임꺽정	씨름	8

전문가의 조언

지금부터는 주어진 릴레이션을 보고 예제의 결과를 꼭 확인하세요.

2 기본 검색

SELECT 절에 원하는 속성을 지정하여 검색한다.

예제 1 〈사원〉 테이블의 모든 튜플을 검색하시오.

```
• SELECT * FROM 사원;
• SELECT 사원.* FROM 사원;
• SELECT 이름, 부서, 생일, 주소, 기본급 FROM 사원;
• SELECT 사원.이름, 사원.부서, 사원.생일, 사원.주소, 사원.기본급 FROM 사원;
```

※ 위의 SQL은 모두 보기에 주어진 〈사원〉 테이블 전체를 그대로 출력한다.

〈결과〉

이름	부서	생일	주소	기본급
홍길동	기획	04/05/61	망원동	120
임꺽정	인터넷	01/09/69	서교동	80
황진이	편집	07/21/75	합정동	100
김선달	편집	10/22/73	망원동	90
성춘향	기획	02/20/64	대흥동	100
장길산	편집	03/11/67	상암동	120
일지매	기획	04/29/78	연남동	110
강건달	인터넷	12/11/80		90

전문가의 조언

중복을 제거하는 DISTINCT의 의미를 정확히 알아두세요.

예제 2 〈사원〉 테이블에서 '주소'만 검색하되 같은 '주소'는 한 번만 출력하시오.

```
SELECT DISTINCT 주소
FROM 사원;
```

〈결과〉

주소
대흥동
망원동
상암동
서교동
연남동
합정동

예제 3 〈사원〉 테이블에서 '기본급'에 특별수당 10을 더한 월급을 "XX부서의 XXX의 월급 XXX" 형태로 출력하시오.

```
SELECT 부서 + "부서의" AS 부서2, 이름 + "의 월급" AS 이름2, 기본급 + 10 AS 기본급2
FROM 사원;
```

부서+"부서의" AS 부서2

'부서'에 "부서의"를 연결하여 표시하되, '부서2'라는 속성명으로 표시합니다.

〈결과〉

부서2	이름2	기본급2
기획부서의	홍길동의 월급	130
인터넷부서의	임꺽정의 월급	90
편집부서의	황진이의 월급	110
편집부서의	김선달의 월급	100
기획부서의	성춘향의 월급	110
편집부서의	장길산의 월급	130
기획부서의	일지매의 월급	120
인터넷부서의	강건달의 월급	100

3 조건 지정 검색

WHERE 절에 조건을 지정하여 조건에 만족하는 튜플만 검색한다.

예제 1 〈사원〉 테이블에서 '기획'부의 모든 튜플을 검색하시오.

```
SELECT *
FROM 사원
WHERE 부서 = '기획';
```

〈결과〉

이름	부서	생일	주소	기본급
홍길동	기획	04/05/61	망원동	120
성춘향	기획	02/20/64	대흥동	100
일지매	기획	04/29/78	연남동	110

예제 2 〈사원〉 테이블에서 "기획" 부서에 근무하면서 "대흥동"에 사는 사람의 튜플을 검색하시오.

```
SELECT *
FROM 사원
WHERE 부서 = '기획' AND 주소 = '대흥동';
```

〈결과〉

이름	부서	생일	주소	기본급
성춘향	기획	02/20/64	대흥동	100

예제 3 〈사원〉 테이블에서 '부서'가 "기획"이거나 "인터넷"인 튜플을 검색하시오.

```
SELECT *
FROM 사원
WHERE 부서 = '기획' OR 부서 = '인터넷';
```

전문가의 조언

기본적인 내용입니다. WHERE절에서 사용되는 AND, OR, LIKE, BETWEEN의 기능을 확실하게 숙지하세요.

예제 3은 다음과 같이 검색해도 됩니다.

```
SELECT *
FROM 사원
WHERE 부서 IN ('기획', '인터넷');
```

〈결과〉

이름	부서	생일	주소	기본급
홍길동	기획	04/05/61	망원동	120
임꺽정	인터넷	01/09/69	서교동	80
성춘향	기획	02/20/64	대흥동	100
일지매	기획	04/29/78	연남동	110
강건달	인터넷	12/11/80		90

예제 4 〈사원〉 테이블에서 성이 '김'인 사람의 튜플을 검색하시오.

```
SELECT *
FROM 사원
WHERE 이름 LIKE "김%";
```

〈결과〉

이름	부서	생일	주소	기본급
김선달	편집	10/22/73	망원동	90

전문가의 조언

날짜 데이터는 숫자로 취급하지만 ' ' 또는 # #으로 묶어줍니다.

예제 5 〈사원〉 테이블에서 '생일'이 '01/01/69'에서 '12/31/73' 사이인 튜플을 검색하시오.

```
SELECT *
FROM 사원
WHERE 생일 BETWEEN #01/01/69# AND #12/31/73#;
```

〈결과〉

이름	부서	생일	주소	기본급
임꺽정	인터넷	01/09/69	서교동	80
김선달	편집	10/22/73	망원동	90

전문가의 조언

NULL이 아닌 값을 검색할 때는 IS NOT NULL을 사용합니다.

예 〈사원〉 테이블에서 '주소'가 NULL이 아닌 튜플 검색

```
SELECT *
FROM 사원
WHERE 주소 IS NOT NULL;
```

예제 6 〈사원〉 테이블에서 '주소'가 NULL인 튜플을 검색하시오.

```
SELECT *
FROM 사원
WHERE 주소 IS NULL;
```

〈결과〉

이름	부서	생일	주소	기본급
강건달	인터넷	12/11/80		90

전문가의 조언

ORDER BY절이 정렬에 사용된다는 것과 정렬 방식인 'ASC', 'DESC'의 의미를 꼭 이해하세요.

4 정렬 검색

ORDER BY 절에 특정 속성을 지정하여 지정된 속성으로 자료를 정렬하여 검색한다.

예제 1 〈사원〉 테이블에서 '주소'를 기준으로 내림차순 정렬시켜 상위 2개 튜플만 검색하시오.

```
SELECT TOP 2 *
FROM 사원
ORDER BY 주소 DESC;
```

〈결과〉

이름	부서	생일	주소	기본급
황진이	편집	07/21/75	대흥동	100
일지매	기획	04/29/78	연남동	110

예제 2 〈사원〉 테이블에서 '부서'를 기준으로 오름차순 정렬하고, 같은 '부서'에 대해서는 '이름'을 기준으로 내림차순 정렬시켜서 검색하시오.

```
SELECT *
FROM 사원
ORDER BY 부서 ASC, 이름 DESC;
```

〈결과〉

이름	부서	생일	주소	기본급
홍길동	기획	04/05/61	망원동	120
일지매	기획	04/29/78	연남동	110
성춘향	기획	02/20/64	대흥동	100
임꺽정	인터넷	01/09/69	서교동	80
강건달	인터넷	12/11/80		90
황진이	편집	07/21/75	합정동	100
장길산	편집	03/11/67	상암동	120
김선달	편집	10/22/73	망원동	90

5 하위 질의

하위 질의는 조건절에 주어진 질의를 먼저 수행하여 그 검색 결과를 조건절의 피연산자로 사용한다.

예제 1 '취미'가 "나이트댄스"인 사원의 '이름'과 '주소'를 검색하시오.

```
SELECT 이름, 주소
FROM 사원
WHERE 이름 = (SELECT 이름 FROM 여가활동 WHERE 취미 = '나이트댄스') ;
```

〈결과〉

이름	주소
성춘향	대흥동

먼저 "SELECT 이름 FROM 여가활동 WHERE 취미 = '나이트댄스'"를 수행하여 〈여가활동〉 테이블에서 '성춘향'을 찾습니다. 그런 다음 하위 질의에 해당하는 피연산자의 자리에 '성춘향'을 대입하면 질의문은 "SELECT 이름, 주소 FROM 사원 WHERE 이름 = '성춘향'"과 같습니다.

Not In ( )
Not In ( )은 포함되지 않는 데이터를 의미합니다. 즉 〈사원〉 테이블에서 모든 자료를 검색하는데, 〈여가활동〉 테이블에 '이름'이 있는 자료는 제외하고 검색합니다.

예제 2 취미활동을 하지 않는 사원들을 검색하시오.

```
SELECT *
FROM 사원
WHERE 이름 NOT IN (SELECT 이름 FROM 여가활동) ;
```

〈결과〉

이름	부서	생일	주소	기본급
홍길동	기획	04/05/61	망원동	120
황진이	편집	07/21/75	합정동	100
장길산	편집	03/11/67	상암동	120
강건달	인터넷	12/11/80		90

6 복수 테이블 검색

여러 테이블을 대상으로 검색을 수행한다.

예제 '경력'이 10년 이상인 사원의 '이름', '부서', '취미', '경력'을 검색하시오.

```
SELECT 사원.이름, 사원.부서, 여가활동.취미, 여가활동.경력
FROM 사원, 여가활동
WHERE 여가활동.경력 >= 10 AND 사원.이름 = 여가활동.이름 ;
```

〈결과〉

이름	부서	취미	경력
김선달	편집	당구	10
일지매	기획	태껸	15

따라잡기

기출문제 따라잡기 Section 105

이전기출
1. 다음 [ ]에 적당한 SQL 문장은?

```
SELECT 번호, 이름
FROM 학생테이블
[ ] 학과 = "컴퓨터";
```

① SET ② GROUP
③ WITH ④ WHERE

검색의 가장 기본이죠. SELECT ~ FROM ~ WHERE! 절대 잊으면 안 됩니다.

이전기출
2. 다음 SQL문에서 WHERE절의 조건이 의미하는 것은?

```
SELECT CNO,CNAME
FROM COURSE
WHERE CNO LIKE 'S_ _';
```

① S로 시작되는 3문자의 CNO를 검색한다.
② S로 시작되는 모든 문자 CNO를 검색한다.
③ 문자열로만 이루어진 모든 CNO를 검색한다.
④ S를 포함한 모든 CNO를 검색한다.

LIKE는 지정된 문자열의 패턴을 비교하여 속성의 값을 찾습니다. 'S'와 '_'이 두 개 있으므로 'S'로 시작하는 3문자가 됩니다.

기출문제 따라잡기

이전기출

3. 학생(STUDENT) 테이블에 전산과 학생이 50명, 전자과 학생이 100명, 기계과 학생이 50명 있다고 할 때, 다음 SQL문 ㉠, ㉡, ㉢의 실행 결과 튜플 수는 각각 얼마인가? (단, DEPT 필드는 학과명을 의미한다)

```
㉠ SELECT DEPT FROM STUDENT;
㉡ SELECT DISTINCT DEPT FROM STUDENT;
㉢ SELECT COUNT(DISTINCT DEPT) FROM STUDENT
  WHERE DEPT = '전산과';
```

① ㉠ 3 ㉡ 3 ㉢ 1　　② ㉠ 200 ㉡ 3 ㉢ 1

③ ㉠ 200 ㉡ 3 ㉢ 50　　④ ㉠ 200 ㉡ 200 ㉢ 50

㉠ STUDENT 테이블에서 DEPT를 검색합니다. 총 200개의 튜플이 들어 있고 검색 조건이 없으므로 튜플의 수는 200입니다.
㉡ STUDENT 테이블에서 DEPT를 검색하는 데 중복된 결과는 처음의 한 개만 검색에 포함시킵니다. 전산과 50개 튜플의 DEPT 속성의 값이 같으므로 1개, 전자과 100개 튜플의 DEPT 속성의 값이 같으므로 1개, 기계과 50개 튜플의 DEPT 속성의 값이 같으므로 1개를 검색에 포함시키므로 3개의 튜플이 검색됩니다.
㉢ STUDENT 테이블에서 DEPT 속성의 값이 '전산과'인 튜플의 중복을 제거하여 개수를 세므로 1이 검색 결과로 나타납니다.

이전기출

4. 다음 표와 같은 판매실적 테이블을 읽어 서울 지역에 한하여 판매액 내림차순으로 지점명과 판매액을 출력하고자 한다. 가장 적절한 SQL 구문은?

〈판매실적 테이블〉

도시	지점명	판매액
서울	강남지점	330
서울	강북지점	168
광주	광주지점	197
서울	강서지점	158
서울	강동지점	197
대전	대전지점	165

① SELECT 지점명, 판매액 FROM 판매실적 WHERE 도시 = '서울' ORDER BY 판매액 DESC;

② SELECT 지점명, 판매액 FROM 판매실적 ORDER BY 판매액 DESC;

③ SELECT 지점명, 판매액 FROM 판매실적 WHERE 도시 = '서울' ASC;

④ SELECT * FROM 판매실적 WHEN 도시 = '서울' ORDER BY 판매액 DESC;

조건은 WHERE, 그러므로 서울 지역에 대한 조건은 WHERE 도시 = '서울'이고, 정렬은 ORDER BY 그리고 내림차순은 DESC이므로, 판매액에 대한 내림차순은 ORDER BY 판매액 DESC입니다.

이전기출

5. 다음의 질의를 SQL문으로 가장 잘 변환한 것은?

3학년 이상의 전자계산과 학생들의 이름을 검색하시오.

① SELECT * FROM 학생 WHEN 학년 〉= 3 AND 학과 = "전자계산";

② SELECT 이름 FROM 학생 WHERE 학년 〉= 3 OR 학과 = "전자계산";

③ SELECT * FROM 학생 FOR 학년 〉= 3 AND 학과 = "전자계산";

④ SELECT 이름 FROM 학생 WHERE 학년 〉= 3 AND 학과 = "전자계산";

이름만 검색하는 것이므로 일단 ②, ④번 중 하나입니다. 그리고 3학년 이상의 전자계산과는 둘 다 포함되는 AND 조건입니다. 만약 OR 조건이라면 3학년이거나 전자계산과… 이렇게 되겠죠.

이전기출

6. 다음 표와 같은 두 테이블에서 성별이 "여"인 사원의 "성명", "나이", "직책"을 구하는 SQL문은?

〈테이블명 : 사원〉

사번	성명	나이	직책
11	홍길동	35	과장
32	안중근	44	부장
41	강동표	37	과장
31	송윤아	24	사원
45	이중건	26	사원
13	김순자	33	계장

〈테이블명 : 연락처〉

사번	성별	연락처
11	남	111-1111
32	남	222-2222
41	남	333-3333
31	여	444-4444

① SELECT * FROM 사원, 연락처 WHERE 성별 = "여";

② SELECT 성명, 나이, 직책 FROM 사원, 연락처 WHERE 성별 = "여";

③ SELECT 성명, 나이, 직책 FROM 사원, 연락처 WHERE 연락처.성별 = "여";

④ SELECT 성명, 나이, 직책 FROM 사원, 연락처 WHERE 연락처.성별 = "여" AND 사원.사번 = 연락처.사번;

"성별"은 〈연락처〉 테이블에 있으므로 〈연락처〉 테이블에서 "성별"이 "여"인 사원을 찾고, "성명", "나이", "직책"은 〈사원〉 테이블에 있으므로 〈연락처〉 테이블의 "사번"과 〈사원〉 테이블의 "사번"을 비교하여 검색합니다. 두 개의 연산이 AND 연산을 하므로 "성별"이 "여"인 사원만 〈사원〉 테이블에서 검색합니다.

▶ 정답 : 1. ④ 2. ① 3. ② 4. ① 5. ④ 6. ④

이전기출
7. 회사원이라는 테이블에서 사원명을 찾을 때, 연락번호가 Null 값이 아닌 사원명을 모두 찾을 때의 SQL 질의로 옳은 것은?

① SELECT 사원명 FROM 회사원 WHERE 연락번호! = NULL;
② SELECT 사원명 FROM 회사원 WHERE 연락번호 〈〉 NULL;
③ SELECT 사원명 FROM 회사원 WHERE 연락번호 IS NOT NULL;
④ SELECT 사원명 FROM 회사원 WHERE 연락번호 DON'T NULL;

NULL 값일 경우는 IS NULL, NULL 값이 아닐 경우는 IS NOT NULL입니다.

이전기출
8. 어떤 컬럼 데이터를 조건 검색하는 SQL문에서 틀린 부분은 몇 번째 line인가? (단, 테이블 : player, 컬럼 : player_name, team_id, height)

① SELECT player_name, height
② FROM player
③ WHERE team_id = 'Korea'
④ AND height BETWEEN 170 OR 180 ;

BETWEEN 연산자는 AND와 짝을 이루어야하므로 ④번을 'AND height BETWEEN 170 AND 180;'이라고 기술해야 합니다. 즉 ③번부터 조건문이므로 'team_id'가 "korea"이고 'height'가 170~180 사이의 데이터를 검색하는 SQL문이 되는 거죠.

이전기출
9. 관계 데이터베이스의 테이블인 수강(학번, 과목명, 중간성적, 기말성적)에서 과목명이 "DB"인 모든 튜플들을 성적에 의해 정렬된 형태로 검색하고자 한다. 이때 정렬 기준은 기말성적의 내림차순으로 정렬하고 기말성적이 같은 경우는 중간성적의 오름차순으로 정렬하고자 한다. 다음 SQL 질의문에서 ORDER BY 절의 밑줄 친 부분의 내용으로 옳은 것은?

```
SELECT * FROM 수강WHERE 과목명 = "DB"
ORDER BY _____;
```

① 중간성적 DESC, 기말성적 ASC
② 기말성적 DESC, 중간성적 ASC
③ 중간성적 DOWN, 기말성적 UP
④ 중간성적 (DESC), 기말성적 (ASC)

첫 번째 정렬 기준은 기말성적의 내림차순이므로 '기말성적 DESC'이고, 두 번째 정렬 기준은 중간성적의 오름차순이므로 '중간성적 ASC'입니다.

이전기출
10. 다음 SQL 문에서 DISTINCT의 의미는?

```
SELECT DISTINCT DEPT FROM STUDENT;
```

① 검색 결과에서 레코드의 중복을 제거하라.
② 모든 레코드를 검색하라.
③ 검색 결과를 순서대로 정렬하라.
④ DEPT의 처음 레코드만 검색하라.

DISTINCT는 별개의, 다른, 독특한 등의 의미가 있다는 걸 염두에 두고 찾아보세요.

이전기출
11. 다음 질의에 대한 SQL문은?

프로젝트번호(PNO) 1, 2, 3에서 일하는 사원의 주민등록번호(JUNO)를 검색하라. (단, 사원 테이블(WORKS)은 프로젝트번호(PNO), 주민등록번호(JUNO) 필드로 구성된다.)

① SELECT WORKS FROM JUNO WHERE PNO IN 1, 2, 3;
② SELECT WORKS FROM JUNO WHERE PNO ON 1, 2, 3;
③ SELECT JUNO FROM WORKS WHERE PNO IN (1, 2, 3);
④ SELECT JUNO FROM WORKS WHERE PNO ON (1, 2, 3);

이제 검색의 기본 구문인 SELECT~FROM~WHERE는 알죠? 조건절에 1, 2, 3 중 하나라도 포함하는 조건을 지정할 때는 WHERE PNO=1 OR PNO=2 OR PNO=3와 같이 입력하거나 WHERE PNO IN(1, 2, 3)와 같이 입력하면 됩니다.

이전기출
12. 다음 BETWEEN 연산의 의미와 동일한 것은?

```
SELECT *
FROM 성적
WHERE (점수 BETWEEN 90 AND 95)
AND 학과 = "컴퓨터공학과";
```

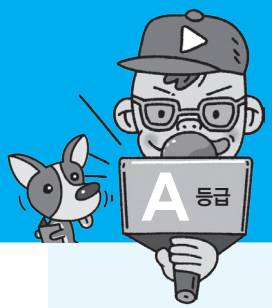
① 점수 〉= 90 AND 점수 〈= 95
② 점수 〉 90 AND 점수 〈 95
③ 점수 〉 90 AND 점수 〈= 95
④ 점수 〉= 90 AND 점수 〈 95

BETWEEN A AND B는 A와 B 사이의 값을 의미합니다.

▶ 정답 : 7. ③ 8. ④ 9. ② 10. ① 11. ③ 12. ①

SECTION 106 ● LIVE

DML – SELECT–2



1 일반 형식

```
SELECT [PREDICATE] [테이블명.]속성명 [AS 별칭][, [테이블명.]속성명, …]
[, 그룹함수(속성명) [AS 별칭]]
[, WINDOW함수 OVER (PARTITION BY 속성명1, 속성명2, …
                    ORDER BY 속성명3, 속성명4, …) [AS 별칭]]
FROM 테이블명[, 테이블명, …]
[WHERE 조건]
[GROUP BY 속성명, 속성명, …]
[HAVING 조건]
[ORDER BY 속성명 [ASC | DESC]];
```

- **그룹함수** : GROUP BY절에 지정된 그룹별로 속성의 값을 집계할 함수를 기술한다.
- **WINDOW 함수** : GROUP BY절을 이용하지 않고 속성의 값을 집계할 함수를 기술한다.
  - PARTITION BY : WINDOW 함수가 적용될 범위*로 사용할 속성을 지정한다.
  - ORDER BY : PARTITION 안에서 정렬 기준으로 사용할 속성을 지정한다.
- **GROUP BY절** : 특정 속성을 기준으로 그룹화하여 검색할 때 사용한다. 일반적으로 GROUP BY절은 그룹 함수와 함께 사용된다.
- **HAVING절** : GROUP BY와 함께 사용되며, 그룹에 대한 조건을 지정한다.

잠깐만요

그룹 함수 / WINDOW 함수

그룹 함수

GROUP BY절에 지정된 그룹별로 속성의 값을 집계할 때 사용됩니다.

- COUNT(속성명) : 그룹별 튜플 수를 구하는 함수
- SUM(속성명) : 그룹별 합계를 구하는 함수
- AVG(속성명) : 그룹별 평균을 구하는 함수
- MAX(속성명) : 그룹별 최대값을 구하는 함수
- MIN(속성명) : 그룹별 최소값을 구하는 함수
- STDDEV(속성명) : 그룹별 표준편차를 구하는 함수
- VAR(속성명) : 그룹별 분산을 구하는 함수

전문가의 조언

- 이번 섹션에서는 WINDOW 함수 이용 검색, 그룹 지정 검색, 집합 연산자를 이용한 통합질의에 대해 학습합니다. 각 절, 옵션의 기능까지 정확히 암기해 두세요. 나머지 중요한 요소는 그때그때 설명하겠습니다.
- SELECT문의 일반 형식에 포함된 내용이 길어 학습할 분량을 두 섹션으로 분리하였습니다. 흐리게 처리된 형식은 이전 섹션에서 학습한 내용입니다.

WINDOW 함수가 적용될 범위

GROUP BY 절에 지정한 속성이 그룹 함수의 범위로 사용되듯이 PARTITION BY 절에 지정한 속성이 WINDOW 함수의 범위로 사용됩니다.

- ROLLUP(속성명, 속성명, …)
  - 인수로 주어진 속성을 대상으로 그룹별 소계를 구하는 함수입니다.
  - 속성의 개수가 n개이면, n+1 레벨까지, 하위 레벨에서 상위 레벨 순으로 데이터가 집계됩니다.
- CUBE(속성명, 속성명, …)
  - ROLLUP과 유사한 형태이나 CUBE는 인수로 주어진 속성을 대상으로 모든 조합의 그룹별 소계를 구합니다.
  - 속성의 개수가 n개이면, n^2 레벨까지, 상위 레벨에서 하위 레벨 순으로 데이터가 집계됩니다.

WINDOW 함수

- GROUP BY절을 이용하지 않고 함수의 인수로 지정한 속성을 범위로 하여 속성의 값을 집계합니다.
- 함수의 인수로 지정한 속성이 대상 레코드의 범위가 되는데, 이를 윈도우(WINDOW)라고 부릅니다.
- WINDOW 함수
  - ROW_NUMBER( ) : 윈도우별로 각 레코드에 대한 일련 번호를 반환합니다.
  - RANK( ) : 윈도우별로 순위를 반환하며, 공동 순위를 반영합니다.
  - DENSE_RANK( ) : 윈도우별로 순위를 반환하며, 공동 순위를 무시하고 순위를 부여합니다.

2 WINDOW 함수 이용 검색

GROUP BY절을 이용하지 않고 함수의 인수로 지정한 속성을 범위로 하여 속성의 값을 집계한다.

〈상여금〉

부서	이름	상여내역	상여금
기획	홍길동	연장근무	100
기획	일지매	연장근무	100
기획	최준호	야간근무	120
기획	장길산	특별근무	90
인터넷	강건달	특별근무	90
인터넷	서국현	특별근무	90
인터넷	박인식	연장근무	30
편집	김선달	특별근무	80
편집	황종근	연장근무	40
편집	성춘향	야간근무	80
편집	임꺽정	야간근무	80
편집	황진이	야간근무	50

예제 1 〈상여금〉 테이블에서 '상여내역'별로 '상여금'에 대한 일련 번호를 구하시오. (단 순서는 내림차순이며 속성명은 'NO'로 할 것)

```
SELECT 상여내역, 상여금,
  ROW_NUMBER( ) OVER (PARTITION BY 상여내역 ORDER BY 상여금 DESC) AS NO
FROM 상여금;
```

〈결과〉

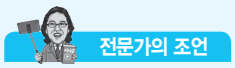
상여내역	상여금	NO
야간근무	120	1
야간근무	80	2
야간근무	80	3
야간근무	50	4
연장근무	100	1
연장근무	100	2
연장근무	40	3
연장근무	30	4
특별근무	90	1
특별근무	90	2
특별근무	90	3
특별근무	80	4

예제 2 〈상여금〉 테이블에서 '상여내역'별로 '상여금'에 대한 순위를 구하시오. (단, 순서는 내림차순이며, 속성명은 '상여금순위'로 하고, RANK() 함수를 이용할 것)

```
SELECT 상여내역, 상여금,
  RANK() OVER (PARTITION BY 상여내역 ORDER BY 상여금 DESC) AS 상여금순위
FROM 상여금;
```

〈결과〉

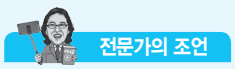
상여내역	상여금	상여금순위
야간근무	120	1
야간근무	80	2
야간근무	80	2
야간근무	50	4
연장근무	100	1
연장근무	100	1
연장근무	40	3
연장근무	30	4
특별근무	90	1
특별근무	90	1
특별근무	90	1
특별근무	80	4


전문가의 조언

RANK() 함수는 공동 순위가 있는 경우 공동 순위를 반영하여 다음 순위를 정합니다.

3 그룹 지정 검색

GROUP BY절에 지정한 속성을 기준으로 자료를 그룹화하여 검색한다.


전문가의 조언

GROUP BY절이 그룹을 지정한다는 것과, 그룹에 대한 조건을 지정할 때는 WHERE가 아닌 HAVING을 사용한다는 것을 기억해 두세요.

Avg(상여금) As 평균
'상여금' 속성에 있는 값들의 평균을 구하되 '평균'이라는 속성명으로 표시합니다.

예제 1 〈상여금〉 테이블에서 '부서'별 '상여금'의 평균을 구하시오.

```
SELECT 부서, AVG(상여금) AS 평균
FROM 상여금
GROUP BY 부서;
```

〈결과〉

부서	평균
기획	102.5
인터넷	70
편집	66

예제 2 〈상여금〉 테이블에서 부서별 튜플 수를 검색하시오.

```
SELECT 부서, COUNT(*) AS 사원수
FROM 상여금
GROUP BY 부서;
```

〈결과〉

부서	사원수
기획	4
인터넷	3
편집	5

- 'WHERE 상여금 >= 100' 절에 의해서 '상여금'이 100 이상인 자료만 검색 대상이 됩니다.
- 'GROUP BY 부서' 절에 의해서 '상여금'이 100 이상인 자료에 대해서만 '부서'별로 그룹을 지정합니다.
- 'HAVING COUNT(*) >= 2' 절에 의해서 '부서'의 인원이 2 이상인 '부서'의 인원만 검색합니다.

예제 3 〈상여금〉 테이블에서 '상여금'이 100 이상인 사원이 2명 이상인 '부서'의 튜플 수를 구하시오.

```
SELECT 부서, COUNT(*) AS 사원수
FROM 상여금
WHERE 상여금 >= 100
GROUP BY 부서
HAVING COUNT(*) >= 2;
```

〈결과〉

부서	사원수
기획	3

예제 4 〈상여금〉 테이블의 '부서', '상여내역', 그리고 '상여금'에 대해 부서별 상여내역별 소계와 전체 합계를 검색하시오. (단, 속성명은 '상여금합계'로 하고, ROLLUP 함수를 사용할 것)

```
SELECT 부서, 상여내역, SUM(상여금) AS 상여금합계
FROM 상여금
GROUP BY ROLLUP(부서, 상여내역);
```

〈결과〉

부서	상여내역	상여금합계	
기획	야간근무	120	3레벨(부서별, 상여내역별 '상여금'의 합계)
기획	연장근무	200	
기획	특별근무	90	
기획		410	→ 2레벨(부서별 '상여금'의 합계)
편집	야간근무	210	3레벨
편집	연장근무	40	
편집	특별근무	80	
편집		330	→ 2레벨
인터넷	연장근무	30	3레벨
인터넷	특별근무	180	
인터넷		210	→ 2레벨
		950	→ 1레벨(전체 '상여금'의 합계)

예제 5 〈상여금〉 테이블의 '부서', '상여내역', 그리고 '상여금'에 대해 부서별 상여내역별 소계와 전체 합계를 검색하시오. (단, 속성명은 '상여금합계'로 하고, CUBE 함수를 사용할 것)

```
SELECT 부서, 상여내역, SUM(상여금) AS 상여금합계
FROM 상여금
GROUP BY CUBE(부서, 상여내역);
```

〈결과〉

부서	상여내역	상여금합계	
		950	→ 1레벨(전체 '상여금'의 합계)
	야간근무	330	2레벨(상여내역별 '상여금'의 합계)
	연장근무	270	
	특별근무	350	
기획		410	→ 3레벨(부서별 '상여금'의 합계)
기획	야간근무	120	4레벨(부서별, 상여내역별 '상여금'의 합계)
기획	연장근무	200	
기획	특별근무	90	
편집		330	→ 3레벨
편집	야간근무	210	4레벨
편집	연장근무	40	
편집	특별근무	80	
인터넷		210	→ 3레벨
인터넷	연장근무	30	4레벨
인터넷	특별근무	180	

A B C D

전문가의 조언

ROLLUP 함수가 적용되는 속성이 2개이므로 집계되는 레벨 수는 2+1로 총 3레벨입니다. 가장 하위 레벨인 3레벨부터 표시됩니다. 3레벨은 부서별 상여내역별 '상여금'의 합계, 2레벨은 부서별 '상여금'의 합계, 1레벨은 전체 '상여금'의 합계가 표시됩니다. ROLLUP 함수는 표기된 속성의 순서에 따라 표시되는 집계 항목이 달라지므로 속성의 순서에 주의해야 합니다. ROLLUP(상여내역, 부서)로 지정하면 3레벨은 상여내역별 부서별 '상여금'의 합계, 2레벨은 상여내역별 '상여금'의 합계, 1레벨은 전체 '상여금'의 합계가 표시되므로 부서별 '상여금'의 합계는 확인할 수 없습니다.

전문가의 조언

CUBE 함수가 적용되는 속성이 2개이므로 집계되는 레벨 수는 2^2로 총 4레벨입니다. CUBE 함수는 가장 상위 레벨인 1레벨부터 표시됩니다. 1레벨은 전체 '상여금'의 합계, 2레벨은 상여내역별 '상여금'의 합계, 3레벨은 부서별 '상여금'의 합계, 4레벨은 부서별 상여내역별 '상여금'의 합계가 표시됩니다. CUBE 함수는 ROLLUP 함수와 달리 인수로 주어진 속성을 대상으로 결합 가능한 모든 집계를 표시하므로 인수로 주어진 속성의 순서가 바뀌어도 표시 순서만 달라질 뿐 표시되는 집계 항목은 동일합니다.

4 집합 연산자를 이용한 통합 질의

집합 연산자를 사용하여 2개 이상의 테이블의 데이터를 하나로 통합한다.

표기 형식

```
SELECT 속성명1, 속성명2, …
FROM 테이블명
UNION | UNION ALL | INTERSECT | EXCEPT
SELECT 속성명1, 속성명2, …
FROM 테이블명
[ORDER BY 속성명 [ASC | DESC]];
```

- 두 개의 SELECT문에 기술한 속성들은 개수와 데이터 유형이 서로 동일해야 한다.
- 집합 연산자의 종류(통합 질의의 종류)

집합 연산자	설명	집합 종류
UNION	• 두 SELECT문의 조회 결과를 통합하여 모두 출력한다. • 중복된 행은 한 번만 출력한다.	합집합
UNION ALL	• 두 SELECT문의 조회 결과를 통합하여 모두 출력한다. • 중복된 행도 그대로 출력한다.	합집합
INTERSECT	두 SELECT문의 조회 결과 중 공통된 행만 출력한다.	교집합
EXCEPT	첫 번째 SELECT문의 조회 결과에서 두 번째 SELECT문의 조회 결과를 제외한 행을 출력한다.	차집합

〈사원〉

사원	직급
김형석	대리
홍영선	과장
류기선	부장
김현천	이사

〈직원〉

사원	직급
신원섭	이사
이성호	대리
홍영선	과장
류기선	부장

예제 1 〈사원〉 테이블과 〈직원〉 테이블을 통합하는 질의문을 작성하시오. (단, 같은 레코드가 중복되어 나오지 않게 하시오.)

```
SELECT *
FROM 사원
UNION
SELECT *
FROM 직원;
```

〈결과〉

사원	직급
김현천	이사
김형석	대리
류기선	부장
신원섭	이사
이성호	대리
홍영선	과장

예제 2 〈사원〉 테이블과 〈직원〉 테이블에 공통으로 존재하는 레코드만 통합하는 질의문을 작성하시오.

```
SELECT *
FROM 사원
INTERSECT
SELECT *
FROM 직원;
```

〈결과〉

사원	직급
류기선	부장
홍영선	과장

기출문제 따라잡기 Section 106

이전기출

1. SQL 문장에서 GROUP BY절에 의해 선택된 그룹의 탐색 조건을 지정할 수 있는 것은?

① HAVING ② ORDER BY
③ UNION ④ JOIN

일반 튜플에 대한 조건은 WHERE, 지정된 그룹에 대한 조건은 HAVING!

이전기출

2. 관계 데이터베이스의 테이블 지점정보(지점코드, 소속도시, 매출액)에 대해 다음과 같은 SQL문이 실행되었다. 그 결과에 대한 설명으로 부적합한 것은?

```
SELECT 소속도시, AVG(매출액)
FROM 지점정보 WHERE 매출액 > 1000
GROUP BY 소속도시
HAVING COUNT(*) >= 3;
```

① WHERE절의 조건에 의해 해당 도시의 지점들의 매출액 평균이 1,000 이하인 경우는 출력에서 제외된다.
② 지점이 세 군데 이상 있는 도시에 대해 각 도시별로 그 도시에 있는 매출액 1,000 초과인 지점들의 평균 매출액을 구하는 질의이다.
③ SELECT절의 "AVG(매출액)"을 "MAX(매출액)"으로 변경하면 각 도시별로 가장 높은 매출을 올린 지점의 매출액을 구할 수 있다.
④ HAVING절에서 "COUNT(*) >= 3"을 "SUM(매출액) >= 5,000"으로 변경하면 어느 한 도시의 지점들의 매출액 합이 5,000 이상인 경우만 그 도시 지점들의 매출액 평균을 구할 수 있다.

질의문이 좀 복잡해 보입니다. 이럴 때는 순서대로 차분히 살펴보는 것이 좋습니다.
- WHERE절에 의해 전체 레코드 중에서 매출액이 1,000을 초과하는 레코드만 추출합니다. 이제 1,000 이하인 레코드는 검색에서 제외되는 거겠죠.
- GROUP BY절에 의해 1,000을 초과한 튜플만을 대상으로 소속도시를 기준으로 하여 그룹을 지정합니다.
- HAVING절에 의해 그룹의 지점 수가 세 개 이상인 소속도시 그룹만 추출됩니다.
- AVG(매출액) 함수에 의해 매출액 평균이 구해집니다.

※ ②번의 "지점이 세 군데 이상 있는 도시에 대해~"가 아니고, 매출액이 1000을 초과하는 지점이 세 개 이상인 도시의 소속도시별 매출액 평균을 구하는 것입니다.

이전기출

3. [상반기진급] 테이블과 [하반기진급] 테이블은 모두 '사번', '이름', '부서' 필드로 구성되어 있다. 다음 중 두 테이블의 레코드를 통합하려고 할 때 쿼리문으로 올바른 것은?

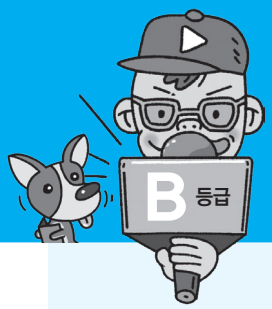
① Select 사번, 이름, 부서 From 상반기진급, 하반기진급
Where 상반기진급.사번 = 하반기진급.사번;
② Select 사번, 이름, 부서 From 상반기진급
JOIN Select 사번, 이름, 부서 From 하반기진급;
③ Select 사번, 이름, 부서 From 상반기진급
OR Select 사번, 이름, 부서 From 하반기진급;
④ Select 사번, 이름, 부서 From 상반기진급
UNION Select 사번, 이름, 부서 From 하반기진급;

성격이 유사한 두 개의 테이블 데이터를 통합하여 하나로 만들려면 합집합(UNION) 연산자를 사용하면 됩니다.

▶ 정답 : 1. ① 2. ② 3. ④

SECTION 107 LIVE

DML – JOIN



1 JOIN의 개념

JOIN(조인)은 2개의 테이블에 대해 연관된 튜플들을 결합하여, 하나의 새로운 릴레이션을 반환한다.

- JOIN은 크게 INNER JOIN과 OUTER JOIN으로 구분된다.
- JOIN은 일반적으로 FROM절에 기술하지만, 릴레이션이 사용되는 어느 곳에서나 사용할 수 있다.

2 INNER JOIN

INNER JOIN은 일반적으로 EQUI JOIN과 NON-EQUI JOIN으로 구분된다.

- 조건이 없는 INNER JOIN을 수행하면 CROSS JOIN*과 동일한 결과를 얻을 수 있다.
- EQUI JOIN
  - EQUI JOIN은 JOIN 대상 테이블에서 공통 속성을 기준으로 '='(equal) 비교에 의해 같은 값을 가지는 행을 연결하여 결과를 생성하는 JOIN 방법이다.
  - EQUI JOIN에서 JOIN 조건이 '='일 때 동일한 속성이 두 번 나타나게 되는데, 이 중 중복된 속성을 제거하여 같은 속성을 한 번만 표기하는 방법을 NATURAL JOIN이라고 한다.
  - EQUI JOIN에서 연결 고리가 되는 공통 속성을 JOIN 속성이라고 한다.
  - WHERE절을 이용한 EQUI JOIN의 표기 형식

```
SELECT [테이블명1.]속성명, [테이블명2.]속성명, …
FROM 테이블명1, 테이블명2, …
WHERE 테이블명1.속성명 = 테이블명2.속성명;
```

  - NATURAL JOIN절을 이용한 EQUI JOIN의 표기 형식

```
SELECT [테이블명1.]속성명, [테이블명2.]속성명, …
FROM 테이블명1 NATURAL JOIN 테이블명2;
```

  - JOIN ~ USING절을 이용한 EQUI JOIN의 표기 형식

```
SELECT [테이블명1.]속성명, [테이블명2.]속성명, …
FROM 테이블명1 JOIN 테이블명2 USING(속성명);
```

전문가의 조언

관계형 데이터베이스의 특성상 정규화 과정을 거치게 되면 여러 개의 테이블로 분리되는데, 이들을 합쳐 사용하기 위해서는 조인 기능이 많이 사용됩니다. 예제를 보며 이해하면 쉬우니, 조인의 의미를 이해하고 구문을 꼭 암기하세요.

CROSS JOIN(교차 조인)

- 교차 조인은 조인하는 두 테이블에 있는 튜플들의 순서쌍을 결과로 반환합니다.
- 교차 조인의 결과로 반환되는 테이블의 행의 수는 두 테이블의 행 수를 곱한 것과 같습니다.

전문가의 조언

실무에서 가장 흔히 사용되는 조인 형식은 WHERE절을 이용한 조인입니다. 교재에서도 특별한 경우를 제외하고는 WHERE절을 이용하여 조인하였습니다.

〈학생〉

학번	이름	학과코드	선배	성적
15	고길동	com		83
16	이순신	han		96
17	김선달	com	15	95
19	아무개	han	16	75
37	박치민		17	55

〈학과〉

학과코드	학과명
com	컴퓨터
han	국어
eng	영어

〈성적등급〉

등급	최저	최고
A	90	100
B	80	89
C	60	79
D	0	59

예제 1 〈학생〉 테이블과 〈학과〉 테이블에서 '학과코드' 값이 같은 튜플을 JOIN하여 '학번', '이름', '학과코드', '학과명'을 출력하는 SQL문을 작성하시오.

```
• SELECT 학번, 이름, 학생.학과코드, 학과명
  FROM 학생, 학과
  WHERE 학생.학과코드 = 학과.학과코드;

• SELECT 학번, 이름, 학생.학과코드, 학과명
  FROM 학생 NATURAL JOIN 학과;

• SELECT 학번, 이름, 학생.학과코드, 학과명
  FROM 학생 JOIN 학과 USING(학과코드);
```

〈결과〉

학번	이름	학과코드	학과명
15	고길동	com	컴퓨터
16	이순신	han	국어
17	김선달	com	컴퓨터
19	아무개	han	국어

- NON-EQUI JOIN
  - NON-EQUI JOIN은 JOIN 조건에 '=' 조건이 아닌 나머지 비교 연산자, 즉 〉, 〈, 〈〉, 〉=, 〈= 연산자를 사용하는 JOIN 방법이다.
  - 표기 형식

```
SELECT [테이블명1.]속성명, [테이블명2.]속성명, …
FROM 테이블명1, 테이블명2, …
WHERE (NON-EQUI JOIN 조건);
```

예제 2 〈학생〉 테이블과 〈성적등급〉 테이블을 JOIN하여 각 학생의 '학번', '이름', '성적', '등급'을 출력하는 SQL문을 작성하시오.

```
SELECT 학번, 이름, 성적, 등급
FROM 학생, 성적등급
WHERE 학생.성적 BETWEEN 성적등급.최저 AND 성적등급.최고;
```

전문가의 조언

두 테이블을 조인하여 사용할 때 한 테이블에만 있는 속성은 테이블명을 생략할 수 있지만 두 테이블에 모두 속해 있는 속성은 반드시 속성명을 테이블명과 함께 표시해야 합니다.

NATURAL JOIN

NATURAL JOIN은 조인할 속성을 지정하지 않기 때문에 조인하려는 두 테이블에는 이름과 도메인이 같은 속성이 반드시 존재해야 합니다. 〈학생〉 테이블과 〈학과〉 테이블에는 같은 이름의 속성과 범위가 같은 도메인을 갖는 '학과코드'가 있기 때문에 NATURAL JOIN이 가능한 것입니다.

전문가의 조언

데이터베이스 실무에서 주로 사용하는 대부분의 JOIN은 EQUI JOIN이고, NON-EQUI JOIN은 별로 사용하지 않습니다.

전문가의 조언

BETWEEN A AND B는 A에서 B 사이의 값을 말합니다. 예를 들어 〈성적등급〉 테이블의 '최저'가 80이고 '최고'가 89일 때, 'WHERE 학생.성적 BETWEEN 성적등급.최저 AND 성적등급.최고'는 〈학생〉 테이블의 '성적'이 80~89인 튜플을 〈성적등급〉 테이블의 '최저' 필드의 값이 80이고, '최고' 필드의 값이 89인 튜플과 조인하므로 등급은 B가 됩니다.

〈결과〉

학번	이름	성적	등급
15	고길동	83	B
16	이순신	96	A
17	김선달	95	A
19	아무개	75	C
37	박치민	55	D

3 OUTER JOIN

OUTER JOIN은 릴레이션에서 JOIN 조건에 만족하지 않는 튜플도 결과로 출력하기 위한 JOIN 방법으로, LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN이 있다.

- **LEFT OUTER JOIN** : INNER JOIN의 결과를 구한 후, 우측 항 릴레이션의 어떤 튜플과도 맞지 않는 좌측 항의 릴레이션에 있는 튜플들에 NULL 값을 붙여서 INNER JOIN의 결과에 추가한다.
  - 표기 형식

```
• SELECT [테이블명1.]속성명, [테이블명2.]속성명, …
  FROM 테이블명1 LEFT OUTER JOIN 테이블명2
  ON 테이블명1.속성명 = 테이블명2.속성명;

• SELECT [테이블명1.]속성명, [테이블명2.]속성명, …
  FROM 테이블명1, 테이블명2
  WHERE 테이블명1.속성명 = 테이블명2.속성명(+);
```

- **RIGHT OUTER JOIN** : INNER JOIN의 결과를 구한 후, 좌측 항 릴레이션의 어떤 튜플과도 맞지 않는 우측 항의 릴레이션에 있는 튜플들에 NULL 값을 붙여서 INNER JOIN의 결과에 추가한다.
  - 표기 형식

```
• SELECT [테이블명1.]속성명, [테이블명2.]속성명, …
  FROM 테이블명1 RIGHT OUTER JOIN 테이블명2
  ON 테이블명1.속성명 = 테이블명2.속성명;

• SELECT [테이블명1.]속성명, [테이블명2.]속성명, …
  FROM 테이블명1, 테이블명2
  WHERE 테이블명1.속성명(+) = 테이블명2.속성명;
```

전문가의 조언

INNER JOIN은 두 릴레이션에서 관련이 있는 튜플만 표시하고, LEFT OUTER JOIN은 좌측 릴레이션이 기준이 되어 좌측 릴레이션에 있는 튜플은 모두 표시하고 우측 릴레이션에서는 관련이 있는 튜플만 표시합니다. 반대로 RIGHT OUTER JOIN은 우측 릴레이션이 기준이 되어 우측 릴레이션에 있는 튜플은 모두 표시하고 좌측 릴레이션에서는 연관된 튜플만 표시합니다.

전문가의 조언

OUTER JOIN에서 '+'를 사용하면 INNER JOIN과 동일한 형식으로 사용할 수 있습니다. INNER JOIN 형식과 동일하게 작성하고 LEFT OUTER JOIN일 때는 조건문의 우측에, RIGHT OUTER JOIN일 때는 조건문의 좌측에 '+'를 붙입니다.

- FULL OUTER JOIN
  - LEFT OUTER JOIN과 RIGHT OUTER JOIN을 합쳐 놓은 것이다.
  - INNER JOIN의 결과를 구한 후, 좌측 항의 릴레이션의 튜플들에 대해 우측 항의 릴레이션의 어떤 튜플과도 맞지 않는 튜플들에 NULL 값을 붙여서 INNER JOIN의 결과에 추가한다. 그리고 유사하게 우측 항의 릴레이션의 튜플들에 대해 좌측 항의 릴레이션의 어떤 튜플과도 맞지 않는 튜플들에 NULL 값을 붙여서 INNER JOIN의 결과에 추가한다.
  - 표기 형식

```
SELECT [테이블명1.]속성명, [테이블명2.]속성명, …
FROM 테이블명1 FULL OUTER JOIN 테이블명2
ON 테이블명1.속성명 = 테이블명2.속성명;
```

예제 1 〈학생〉 테이블과 〈학과〉 테이블에서 '학과코드' 값이 같은 튜플을 JOIN하여 '학번', '이름', '학과코드', '학과명'을 출력하는 SQL문을 작성하시오. 이때, '학과코드'가 입력되지 않은 학생도 출력하시오.

```
• SELECT 학번, 이름, 학생.학과코드, 학과명
  FROM 학생 LEFT OUTER JOIN 학과
  ON 학생.학과코드 = 학과.학과코드;

• SELECT 학번, 이름, 학생.학과코드, 학과명
  FROM 학생, 학과
  WHERE 학생.학과코드 = 학과.학과코드(+);
```

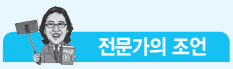
해설 INNER JOIN을 하면 '학과코드'가 입력되지 않은 "박치민"은 출력되지 않는다. 그러므로 JOIN 구문을 기준으로 왼쪽 테이블, 즉 〈학생〉의 자료는 모두 출력되는 LEFT JOIN을 사용한 것이다. 다음과 같이 JOIN 구문을 기준으로 테이블의 위치를 교환하여 RIGHT JOIN을 사용해도 결과는 같다.

```
• SELECT 학번, 이름, 학생.학과코드, 학과명
  FROM 학과 RIGHT OUTER JOIN 학생
  ON 학과.학과코드 = 학생.학과코드;

• SELECT 학번, 이름, 학생.학과코드, 학과명
  FROM 학과, 학생
  WHERE 학과.학과코드(+) = 학생.학과코드;
```

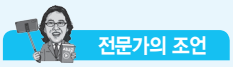
〈결과〉

학번	이름	학과코드	학과명
15	고길동	com	컴퓨터
16	이순신	han	국어
17	김선달	com	컴퓨터
19	아무개	han	국어
37	박치민		



전문가의 조언

예제 1을 462쪽에 있는 테이블을 참조하여 풀어보세요.



전문가의 조언

- LEFT OUTER JOIN은 좌측 릴레이션을 기준으로 좌측 릴레이션에 있는 튜플은 모두 표시하고, 우측 릴레이션에서는 관련 있는 튜플만 표시했으므로 "박치민"의 '학과명'이 빈 자리로 표시됩니다.
- RIGHT OUTER JOIN은 테이블의 위치를 변경했으므로 LEFT OUTER JOIN의 결과와 같은 결과가 표시됩니다.

예제 2 〈학생〉 테이블과 〈학과〉 테이블에서 '학과코드' 값이 같은 튜플을 JOIN하여 '학번', '이름', '학과코드', '학과명'을 출력하는 SQL문을 작성하시오. 이때, '학과코드'가 입력 안 된 학생이나 학생이 없는 '학과코드'도 모두 출력하시오.

```
SELECT 학번, 이름, 학과.학과코드, 학과명
FROM 학생 FULL OUTER JOIN 학과
ON 학생.학과코드 = 학과.학과코드;
```

해설 FULL OUTER JOIN을 하면 JOIN 구문으로 연결되지 않는 자료도 모두 출력된다. "박치민"은 '학과코드'가 없고, "eng"는 〈학생〉 테이블에 등록되지 않아서 연결고리가 없지만 FULL OUTER JOIN을 했으므로 모두 출력된다.

〈결과〉

학번	이름	학과코드	학과명
15	고길동	com	컴퓨터
16	이순신	han	국어
17	김선달	com	컴퓨터
19	아무개	han	국어
37	박치민		
		eng	영어

4 SELF JOIN

- SELF JOIN은 같은 테이블에서 2개의 속성을 연결하여 EQUI JOIN을 하는 JOIN 방법이다.
- 표기 형식

```
• SELECT [별칭1.]속성명, [별칭1.]속성명, …
  FROM 테이블명1 [AS] 별칭1 JOIN 테이블명1 [AS] 별칭2
  ON 별칭1.속성명 = 별칭2.속성명;

• SELECT [별칭1.]속성명, [별칭1.]속성명, …
  FROM 테이블명1 [AS] 별칭1, 테이블명1 [AS] 별칭2
  WHERE 별칭1.속성명 = 별칭2.속성명;
```

예제 〈학생〉 테이블을 SELF JOIN하여 선배가 있는 학생과 선배의 '이름'을 표시하는 SQL문을 작성하시오.

```
• SELECT A.학번, A.이름, B.이름 AS 선배
  FROM 학생 A JOIN 학생 B
  ON A.선배 = B.학번;

• SELECT A.학번, A.이름, B.이름 AS 선배
  FROM 학생 A , 학생 B
  WHERE A.선배 = B.학번;
```

전문가의 조언

- 'B.이름 AS 선배'는 〈B〉 테이블의 '이름'을 출력하되 필드명을 '선배'로 표시하라는 의미입니다.
- '학생 A' 대신 '학생 AS A'로 써도 됩니다.
- 예제를 462쪽의 테이블을 이용하여 풀어보세요.

〈결과〉

학번	이름	선배
17	김선달	고길동
19	아무개	이순신
37	박치민	김선달

잠깐만요

SELF 조인의 이해

SELF 조인은 1개의 테이블을 2개의 이름으로 사용하므로 종종 결과가 혼동됩니다. 이럴 때는 같은 테이블을 2개 그려서 생각하면 쉽게 결과를 알아낼 수 있습니다. '학번', '이름', '선배' 필드만 사용하므로 3개의 필드만 가지고 생각해 봅시다.

〈A〉

학번	이름	선배
15	고길동	
16	이순신	
17	김선달	15
19	아무개	16
37	박치민	17

〈B〉

학번	이름	선배
15	고길동	
16	이순신	
17	김선달	15
19	아무개	16
37	박치민	17

해설

〈A〉 테이블의 '선배'와 〈B〉 테이블의 '학번'이 같은 튜플을 조인하면 위 그림과 같이 연결됩니다. 여기서 두 테이블 간 조인된 튜플들만을 대상으로 〈A〉 테이블에서 '학번', '이름'을 표시하고, 〈B〉 테이블에서 이름을 출력하되 필드명을 '선배'로 하여 출력하면 위의 결과와 같이 됩니다.

따라잡기

기출문제 따라잡기

Section 107

출제예상

1. 다음 중 조인(Join)에 대한 설명으로 옳지 못한 것은?

① 두 개 이상의 테이블로부터 원하는 데이터를 검색하는 방법이다.

② 조인에 사용되는 기준 필드는 동일하거나 호환되는 데이터 형식을 가져야 한다.

③ 조인되는 두 테이블의 필드 수가 동일할 필요는 없다.

④ 같은 테이블에서 2개의 속성을 연결하여 EQUI JOIN을 하는 방법을 CROSS JOIN이라고 한다.

같은 테이블에서 조인을 수행한다는 것은 자기 자신(SELF)의 데이터를 이용한다는 것이죠.

출제예상

2. 다음 쿼리에서 두 테이블의 필드 값이 일치하는 레코드만 조인하기 위해 괄호 안에 넣어야 할 것으로 옳은 것은?

```
SELECT 필드목록
FROM 테이블1, 테이블2
ON 테이블1.필드 (      ) 테이블2.필드;
```

① =　　② JOIN

③ +　　④ −

조인된 필드의 값이 일치하는 행을 연결하여 결과를 생성하는 JOIN을 EQUI JOIN이라고 합니다.

따라잡기

기출문제 따라잡기

Section 107

출제예상

3. 관리(관리번호, 회원번호, 이름) 테이블과 회원(회원번호, 이름, 등급) 테이블을 다음과 같이 조인하여 질의한 결과로 옳은 것은?

〈관리〉

관리번호	회원번호	이름
G101	M101	최재균
G102	M104	김봉선
G105	M107	최준호

〈회원〉

회원번호	이름	등급
M101	최재균	일반
M104	김봉선	우수
M105	김삼순	VIP

〈질의문〉

```
SELECT 관리.*, 회원.*
FROM 관리 INNER JOIN 회원
ON 관리.회원번호 = 회원.회원번호;
```

①

관리번호	관리.회원번호	관리.이름	회원.회원번호	회원.이름	등급
G101	M101	최재균	M101	최재균	일반
G102	M104	김봉선	M104	김봉선	우수

②

관리번호	관리.회원번호	관리.이름	회원.회원번호	회원.이름	등급
G101	M101	최재균	M101	최재균	일반
G102	M104	김봉선	M104	김봉선	우수
G105	M107	최준호			

③

관리번호	관리.회원번호	관리.이름	회원.회원번호	회원.이름	등급
			M015	김삼순	VIP
G101	M101	최재균	M101	최재균	일반
G102	M104	김봉선	M104	김봉선	우수

④

관리번호	관리.회원번호	관리.이름	회원.회원번호	회원.이름	등급
			M015	김삼순	VIP
G101	M101	최재균	M101	최재균	일반
G102	M104	김봉선	M104	김봉선	우수
G105	M107	최준호			

질의문은 각 절을 분리하여 이해하면 쉽습니다.

- SELECT 관리.*, 회원.* : 〈관리〉 테이블의 모든 필드와 〈회원〉 테이블의 모든 필드를 검색합니다.
- FROM 관리 INNER JOIN 회원 : 〈관리〉 테이블과 〈회원〉 테이블에서 조인된 필드가 일치하는 행만 질의에 포함됩니다.
- ON 관리.회원번호 = 회원.회원번호 : 〈관리〉 테이블의 '회원번호'와 〈회원〉 테이블의 '회원번호' 값이 같은 레코드만 조인됩니다.

※ 〈관리〉 테이블의 '회원번호'(M101, M104, M107)와 〈회원〉 테이블의 '회원번호'(M101, M104, M105)의 값이 같은 레코드(M101, M104)만 조인됩니다.

관리번호	관리.회원번호	관리.이름	회원.회원번호	회원.이름	등급
G101	M101	최재균	M101	최재균	일반
G102	M104	김봉선	M104	김봉선	우수

이전기출

4. 다음 테이블 조인(JOIN)에 대한 설명으로 가장 적절한 것은?

- 가능한 모든 행들의 조합이 표시된다.
- 첫 번째 테이블의 모든 행들은 두 번째 테이블의 모든 행들과 조인된다.
- 첫 번째 테이블의 행수를 두 번째 테이블의 행수로 곱한 것만큼의 행을 반환한다.
- 조인 조건이 없는 조인이라고 할 수 있다.

① INNER JOIN ② LEFT JOIN
③ RIGHT JOIN ④ CROSS JOIN

두 테이블의 행수를 곱한 것만큼 행을 반환한다는 것은 두 테이블의 행을 서로 교차(CROSS)하여 조인을 수행한다는 의미입니다.

▶ 정답 : 1. ④ 2. ① 3. ① 4. ④

예·상·문·제·은·행

1. 다음 SQL에 대한 설명 중 틀린 것은?

① 1974년 IBM 연구소에서 개발한 SEQUEL(Structured English Query Language)에서 유래되었다.

② 관계대수와 관계해석을 기초로 한 혼합 데이터 언어이다.

③ IBM에서 개발한 시스템에만 한정적으로 사용된다.

④ 데이터 정의, 데이터 조작, 데이터 제어 기능을 모두 갖추고 있다.

2. SQL의 DDL에 의해 만들어지지 않고 질의문 처리 결과로 만들어지는 것은?

① 기본 테이블(Base Table)

② 임시 테이블(Temporary Table)

③ 가상 테이블(Virtual Table)

④ 뷰(View)

3. 스키마의 식별을 위해 스키마 이름과 소유권자나 허가권자를 정의하는 명령문은?

① DEFINE SCHEMA

② ALTER SCHEMA

③ CREATE SCHEMA

④ DROP SCHEMA

4. 임의의 애트리뷰트에서 취할 수 있는 원자값의 범위가 SQL에서 지원하는 data_type에 포함되는 전체 값이 아니고 일부분일 때, 사용자가 그 값의 범위를 사용자 정의 data_type으로 정의하는 명령문은?

① DEFINE SCHEMA

② CREATE SCHEMA

③ DEFINE DOMAIN

④ CREATE DOMAIN

5. SQL의 데이터형 중에서 4바이트 정수형을 나타내는 것은?

① DECIMAL ② INTEGER

③ SMALLINT ④ FLOAT

6. 학년 속성이 가질 수 있는 값의 범위를 1~4의 2바이트 정수로만 사용할 수 있는 도메인 SYEAR를 정의하는 명령문으로 올바른 것은? (단, 기본값은 1이며, NULL일 수 없음)

① CREATE DOMAIN SYEAR SMALLINT DEFAULT 1 CONSTRAINT VALID-SYEAR CHECK(VALUE IN (1, 2, 3, 4)) NOT NULL;

② DEFINE DOMAIN SYEAR SMALLINT DEFAULT 1 CONSTRAINT VALID-SYEAR CHECK(VALUE IN (1, 2, 3, 4)) NOT NULL;

③ CREATE DOMAIN SYEAR SMALLINT DEFAULT 1 CONSTRAINT VALID-SYEAR CHECK(VALUE IN (1~4)) NOT NULL;

④ DEFINE DOMAIN SYEAR SMALLINT DEFAULT 1 CONSTRAINT VALID-SYEAR CHECK(VALUE IN (1~4)) NOT NULL;

7. SQL에 존재하는 CREATE TABLE 명령어에 대한 설명으로 옳지 않은 것은?

① 일반적으로 테이블 생성 명령어에 스키마를 생성하지 않으면 현재 명령어가 실행되기 이전 환경에 있는 스키마에 명령이 적용된다.

② 테이블의 이름, 속성, 제약 사항을 명시한다.

③ 새로운 테이블을 생성하기 위해 사용한다.

④ 명령을 적용할 스키마를 명시할 수 있다.

8. CREATE TABLE에 대한 설명으로 틀린 것은?

① 테이블명 및 해당 테이블에 속하는 컬럼 이름, 데이터 타입 등을 명시한다.

② PRIMARY KEY 절에서는 기본키 속성을 지정한다.

③ CHECK 절은 인덱스에 대한 정보를 저장한다.

④ NOT NULL은 널 값을 허용하지 않을 때 지정한다.

9. 다음 명령문에 대한 설명으로 잘못된 것은?

```
CREATE TABLE 학생
  (이름 VARCHAR(15) Not Null,
  학번 VARCHAR(15) Not Null,
  전공 VARCHAR(20) Not Null,
  성별 SEX,
  생년월일 DATE,
  PRIMARY KEY (학번),
  FOREIGN KEY (전공) REFERENCES 학과(학과코드)
  CHECK (성별= '남' ) );
```

① 이름, 학번, 전공, 성별, 생년월일로 구성된 '학생' 테이블을 정의한다.

② 이름은 최대 15문자까지 입력할 수 있다.

③ 학과 릴레이션의 기본키인 학과코드를 학생 릴레이션이 참조할 때 전공 속성을 외래키로 이용한다.

④ 성별 속성의 data_type은 SQL에서 지원하는 기본 Type이다.

해설은 472쪽에 있습니다.

10. '고객' 테이블의 기본키인 '국가' 속성에 대해 오름차순 정렬하여 고객_INX 인덱스를 구성하기 위한 명령문은? (단, 동일 인덱스 값을 갖는 튜플들을 그룹으로 묶음)

① CREATE UNIQUE INDEX 고객_INX ON 고객(국가 ASC);
② CREATE UNIQUE INDEX 고객_INX ON 고객(국가 ASC) CLUSTER;
③ CREATE UNIQUE INDEX 고객_INX ON 고객(국가 DESC);
④ CREATE UNIQUE INDEX 고객_INX ON 고객(국가 DESC) CLUSTER;

11. 다음 SQL 명령어에 대해 바르게 설명한 것은?

```
ALTER TABLE STUDENT
ADD ADDRESS CHAR(20);
```

① STUDENT 테이블에서 ADDRESS행을 추가한다.
② STUDENT 테이블에서 ADDRESS열을 추가한다.
③ STUDENT 테이블을 ADDRESS 릴레이션으로 변경한다.
④ STUDENT 테이블에서 ADDRESS열을 삭제한다.

12. SQL의 DROP TABLE 명령어에 대한 설명으로 옳은 것은?

① 테이블을 수정하는 데 사용된다.
② CASCADE 옵션 하나만 존재한다.
③ 제거될 테이블을 참조하는 모든 제약과 뷰를 자동적으로 스키마로부터 삭제시키는 옵션이 존재한다.
④ 테이블이 제약이나 뷰로부터 참조되지 않는 경우에만 삭제되도록 하는 옵션은 존재하지 않는다.

13. SQL 언어의 질의 기능에 대한 설명 중 옳지 않은 것은?

① SELECT절은 질의 결과에 포함될 데이터 행들을 기술하며, 이는 데이터베이스로부터 데이터 행 또는 계산 행이 될 수 있다.
② FROM절은 질의에 의해 검색될 데이터들을 포함하는 테이블을 기술한다.
③ 복잡한 탐색 조건을 구성하기 위하여 단순 탐색 조건들을 AND, OR, NOT으로 결합할 수 있다.
④ ORDER BY절은 질의 결과가 한 개 또는 그 이상의 열 값을 기준으로 오름차순 또는 내림차순으로 정렬될 수 있도록 기술된다.

14. 다음 두 테이블 R과 S에 대한 아래 SQL문의 실행 결과로 옳은 것은?

```
SELECT A FROM R
UNION
SELECT A FROM S;
```

R

A	B
1	A
2	B
3	C

S

A	C
2	X
3	Y
4	Z

① 1, 2, 3　　② 2, 3, 4
③ 2, 3　　④ 1, 2, 3, 4

15. SQL문에서 HAVING을 사용할 수 있는 절은?

① LIKE절　　② WHERE절
③ GROUP BY절　　④ ORDER BY절

16. 입교 지원 현황을 조회하고자 할 때 다음 예시된 SQL 구문으로 알 수 없는 것은?

```
SELECT 지원, 지원학과, 전화번호
FROM 지원자
WHERE 점수 > 59
ORDER BY 지원학과, 점수 DESC;
```

① 지원자 테이블을 검색한다.
② 점수가 60점 이상인 지원자만을 검색한다.
③ 지원자 전체에 대해 점수 순(내림차순)으로 정렬된다.
④ 지원학과별 점수 순위를 알 수 있다.

17. 다음 질의어를 SQL 문장으로 바르게 나타낸 것은? (단, 학생 테이블에 학번, 이름, 학과의 열이 있다고 가정한다.)

```
학번이 100, 이름이 홍길동, 학과가 컴퓨터인 학생을 학생
테이블에 삽입하라.
```

① UPDATE 학생 SET 학번=100, 이름='홍길동', 학과='컴퓨터'
② INSERT INTO 학생 VALUES(100, '홍길동', '컴퓨터')
③ INSERT 학생 VALUE(100, '홍길동', '컴퓨터')
④ UPDATE 학생 SET(100, '홍길동', '컴퓨터')

▶ 정답 : 1. ③ 2. ② 3. ③ 4. ④ 5. ② 6. ① 7. ① 8. ③ 9. ④ 10. ② 11. ② 12. ③ 13. ① 14. ④ 15. ③ 16. ③ 17. ②

18. SQL에서 조건문으로 기술할 수 있는 구문은?

① SELECT ② FROM
③ DISTINCT ④ LIKE

19. 다음 SQL문을 올바르게 설명한 것은?

```
SELECT *
FROM STUDENT
WHERE SNAME LIKE '홍%';
```

① SNAME이 '홍'씨로 시작하면 삭제한다.
② SNAME이 '홍'씨로 시작되는 튜플을 찾는다.
③ SNAME이 '홍'씨로 시작하면 0으로 치환한다.
④ SNAME이 '홍'씨로 시작되는 튜플을 삭제한다.

20. 다음 SQL에서 사용되는 내장 함수 중 잘못 표현된 것은?

① SUM – 열에 있는 값들의 합계
② COUNT – 튜플의 개수
③ MAX – 열에서 최대값
④ AVR – 열에 있는 값들의 평균

21. 그룹화에 대한 설명으로 틀린 것은?

① 튜플이 어떤 속성값을 기준으로 분류된 그룹 단위에 내장 함수를 적용하고자 하는 경우에 사용한다.
② 튜플을 그룹으로 분류하는 데 사용되는 속성을 그룹화 속성(Grouping Attribute)이라 한다.
③ 그룹화를 위하여 GROUP AS절을 사용하여 그룹화 속성을 명시한다.
④ 함수를 적용시킨 결과를 그룹화 속성의 값과 같이 나타나도록 하기 위해서 SELECT절에 그룹화 속성을 같이 명시해야 한다.

22. 다음은 질의 수행 결과를 정렬하는 방법에 대한 설명이다. 옳지 않은 것은?

① ORDER BY절을 사용한다.
② 기본 정렬 방식은 내림차순이다.
③ ASC 키워드를 사용하여 사용자가 오름차순 정렬 방식을 지정할 수 있다.
④ DESC 키워드를 사용하여 사용자가 내림차순 정렬 방식을 지정할 수 있다.

23. 다음의 관계형 데이터베이스에서 "질의 : 제일은행(Company) 직원들의 이름(Name)과 그들이 사는 도시(City)를 찾아라."로 작성한 것의 ㉠, ㉡, ㉢의 위치에 적당한 내용은?

SQL : Select ㉠	lives(name, street, city)
From ㉡	works(name, company, salary)
Where ㉢	located(company, city);

① ㉠ name, lives.city
㉡ lives, located
㉢ company = '제일은행'
② ㉠ name, city
㉡ works, located
㉢ works.company = '제일은행' and located.com pany = '제일은행'
③ ㉠ works.name, city
㉡ lives, works
㉢ company = '제일은행' and lives.name = work.name
④ ㉠ lives.name, lives.city
㉡ lives, works, located
㉢ lives.name = works.name and works.company = '제일은행' and located company = '제일은행'

24. 다음 중 SQL의 INSERT 명령어에 대한 설명으로 옳지 않은 것은?

① 명령 하나로 동시에 두 개 이상의 테이블에 각각 삽입할 수 있다.
② 부속 질의어를 사용할 수 있다.
③ 하나의 INSERT 명령어를 사용하여 여러 개의 튜플을 삽입할 수도 있다.
④ NULL 값도 삽입할 수 있다.

25. 다음 문장을 만족하는 SQL 문장은?

학번이 1000번인 학생을 학생 테이블에서 삭제하시오.

① DELETE FROM 학생 WHERE 학번 = 1000;
② DELETE FROM 학생 IF 학번 = 1000;
③ SELECT * FROM 학생 WHERE 학번 = 1000;
④ SELECT * FROM 학생 CONDITION 학번 = 1000;

해설은 473쪽에 있습니다.

26. SQL의 UPDATE 명령어에 대한 설명으로 옳지 않은 것은?

① 하나 또는 그 이상의 튜플의 속성값을 변경하는 데 사용된다.

② 테이블에서 수정할 튜플을 선택하기 위해 WHERE절이 사용된다.

③ 변경할 속성과 그들의 새로운 값을 명시하기 위해 AS 절이 사용된다.

④ 참조 무결성 제약이 존재하는 경우에는 기본키 값을 변경하는 경우 그 변경이 외래키 값에 영향을 미칠 수 있다.

27. 다음 SQL문을 올바르게 설명한 것은?

```
UPDATE STUDENT
SET SCORE = SCORE + 10
WHERE SNAME = 'LEE' ;
```

① STUDENT 테이블에서 SNAME이 'LEE'인 모든 튜플의 SCORE 속성에 10을 더한다.

② STUDENT 테이블에서 SNAME이 'LEE'인 첫 번째 튜플의 SCORE 속성에 10을 더한다.

③ STUDENT 테이블에서 SNAME이 'LEE'인 모든 튜플의 SCORE 속성에 10을 더하여 삭제한다.

④ STUDENT 테이블에서 SNAME이 'LEE'인 모든 튜플의 SCORE 속성에 10을 곱한다.

28. 다음 SQL문 중에서 구문적 오류가 있는 것은?

① DELETE FROM STUDENT, ENROL WHERE SNO = 100;

② INSERT INTO STUDENT(SNO, SNAME, YEAR) VALUES(100, '홍길동', 4);

③ INSERT INTO COMPUTER(SNO, SNAME, YEAR) SELECT SNO, SNAME, YEAR FROM STUDENT WHERE DEPT='CE';

④ UPDATE STUDENT SET DEPT = (SELECT DEPT FROM COURSE WHERE CNO='C123') WHERE YEAR = 4;

29. 아래와 같은 결과를 만들어 내는 SQL문은?

[공급자 Table]

공급자번호	공급자명	위치
16	대신공업사	수원
27	삼진사	서울
39	삼양사	인천
62	진아공업사	대전
70	신촌상사	서울

[결과 Table]

공급자번호	공급자명	위치
16	대신공업사	수원
70	신촌상사	서울

① SELECT * FROM 공급자 WHERE 공급자명 LIKE '%신%';

② SELECT * FROM 공급자 WHERE 공급자명 LIKE '대%';

③ SELECT * FROM 공급자 WHERE 공급자명 LIKE '%사';

④ SELECT * FROM 공급자 WHERE 공급자명 LIKE '_사';

30. 다음 질의문 실행의 결과는?

```
SELECT 가격 FROM 도서가격 WHERE 책번호 =
(SELECT 책번호 FROM 도서 WHERE 책명 = '운영체제');
```

책번호	책명
1111	운영체제
2222	세계지도
3333	생활영어

[도서 테이블]

책번호	가격
1111	15000
2222	23000
3333	7000
4444	5000

[도서가격 테이블]

① 5000 ② 7000

③ 15000 ④ 23000

▶ 정답 : 18. ④ 19. ② 20. ④ 21. ③ 22. ② 23. ③ 24. ① 25. ① 26. ③ 27. ① 28. ① 29. ① 30. ③

예·상·문·제·은·행·해·설

1. Section 101

SQL(Structured Query Language)

- 1974년 IBM 연구소에서 개발한 SEQUEL에서 유래한다.
- IBM 외에도 많은 회사에서 관계형 데이터베이스를 지원하는 언어로 채택하고 있다.
- 관계대수와 관계해석을 기초로 한 혼합 데이터 언어이다.
- 질의어이지만, 질의 기능만 있는 것이 아니라 데이터 구조의 정의, 데이터 조작, 데이터 제어 기능을 모두 갖추고 있다.

2. Section 102

SQL에서 사용하는 테이블의 종류

- 기본 테이블(Base Table) : 이름을 가지고 있으며 독자적으로 존재함
- 뷰 테이블(View Table) : 독자적으로 존재하지 못하고, 기본 테이블로부터 유도된 이름을 가진 가상 테이블
- 임시 테이블(Temporary Table) : 질의문 처리 결과로 만들어진 테이블로서, 이름을 가지지 않음

3. Section 102

스키마 정의문 형식

```
CREATE SCHEMA 스키마명 AUTHORIZATION
사용자_ID;
```

4. Section 102

도메인(사용자 정의 Data_Type) 정의문

```
CREATE DOMAIN 도메인명 데이터_타입
[DEFAULT 기본값]
[CONSTRAINT 제약조건명 CHECK (범위값)];
```

5. Section 102

데이터 형(Data Type)

- DECIMAL(m,n) : 10진 소수
- INTEGER : 4Byte 정수
- SMALLINT : 2Byte 정수
- FLOAT : 부동 소수점 수
- CHAR(n) : 문자의 수가 n인 스트링
- VARCHAR(n) : 문자의 수가 최대 n인 스트링

6. Section 102

도메인(사용자 정의 Data_Type) 정의문

```
CREATE DOMAIN 도메인명 데이터_타입
[DEFAULT 기본값]
[CONSTRAINT 제약조건명 CHECK (범위값)];
```

7. Section 102

일반적으로 테이블 생성 명령어에 스키마를 생성하지 않으면 현재 명령어가 실행되는 환경에 있는 스키마에 명령이 적용된다.

CREATE TABLE 명령어

- 새로운 테이블을 생성하기 위해서 사용한다.
- 테이블의 이름, 속성, 제약을 명시한다.
- 명령을 적용할 스키마를 명시할 수 있다.

8. Section 102

CHECK 절은 제약 조건을 정의한다.

9. Section 102

기본 data_type에는 SEX가 없기 때문에 CREATE DOMAIN으로 사용자가 정의한 data_type이라고 보아야 한다.

10. Section 102

```
CREATE [UNIQUE] INDEX 인덱스명
        ON 테이블명(속성명 [ASC | DESC])
        [CLUSTER];
```

- 정렬 여부 지정
  - ASC : 오름차순 정렬
  - DESC : 내림차순 정렬
- CLUSTER : 동일 인덱스 값을 갖는 튜플들을 그룹으로 묶을 때 사용

11. Section 102

ALTER TABLE 일반 형식

```
ALTER TABLE 테이블명 ADD 속성명 데이터_타입
        [DEFAULT '기본값' ];
ALTER TABLE 테이블명 ALTER 속성명 [SET
        DEFAULT '기본값' ];
ALTER TABLE 테이블명 DROP COLUMN 속성명
        [CASCADE];
```

- ADD : 새로운 속성(열)을 추가할 때 사용
- ALTER : 특정 속성의 Default 값을 변경할 때 사용
- DROP COLUMN : 특정 속성을 삭제할 때 사용

※ 열은 속성을 말한다.

12. Section 102

DROP 명령어

- SCHEMA, DOMAIN, TABLE, VIEW를 제거하는 데 사용된다.
- CASCADE와 RESTRICT의 두 개의 옵션이 존재한다.
- CASCADE 옵션을 사용하면 제거될 테이블을 참조하는 모든 제약과 뷰도 자동으로 스키마로부터 삭제된다.
- RESTRICT 옵션이 사용되면 테이블이 제약이나 뷰로부터 참조되지 않는 경우에만 삭제된다.

13. Section 106

SELECT문의 일반 형식

```
SELECT PREDICATE [테이블명.]속성명1, [테이블명.]속성명2,…
FROM 테이블명1, 테이블명2,…
[WHERE 조건]
[GROUP BY 속성명1, 속성명2,…]
[HAVING 조건]
[ORDER BY 속성명 [ASC | DESC]];
```

※ SELECT절은 질의 결과에 포함될 데이터 열(속성)들을 기술하며, 이는 데이터베이스로부터 데이터 열 또는 계산 열이 될 수 있다.

14. Section 106

UNION은 중복을 제거하여 두 개의 테이블을 통합하는 명령으로, 두 개의 테이블에서 A 속성을 합치면 1, 2, 2, 3, 3, 4이지만 여기서 중복을 제거하면 1, 2, 3, 4가 된다.

15. Section 106

HAVING절은 GROUP BY와 함께 사용되는 것으로, 그룹에 대한 조건을 지정한다.

16. Section 105

정렬을 나타내는"ORDER BY 지원학과, 점수 DESC"를 통해 '지원학과' 순으로 오름차순 정렬되고, '지원학과'가 같은 경우는 '점수'를 기준으로 내림차순 정렬됨을 알수 있다. 또한 이 모든 정렬은 WHERE 점수 〉 59에 의해 '점수'가 60점 이상인 지원자만을 대상으로 함을 알 수 있다.

17. Section 104

INSERT의 일반 형식

```
INSERT INTO 테이블명(속성명1, 속성명2,… )
VALUES (데이터1, 데이터2,… );
```

※ 기본 테이블의 모든 속성을 사용하여 데이터를 삽입할 경우에는 속성명을 생략할 수 있다.

18. Section 105

LIKE

(속성 LIKE "부분문자%") 형식을 사용하여 지정된 속성에서 부분 문자가 들어 있는 튜플을 대상으로 검색시킬 조건을 표현할 때 사용한다.

19. Section 105

LIKE는 문자열의 패턴을 비교할 때 사용하는 연산자이고, '%'는 모든 문자를 의미하는 와일드카드 문자이므로 WHERE SNAME LIKE '홍%'은 'SNAME' 속성의 값이 "홍"으로 시작하는 모든 튜플을 의미한다.

20. Section 106

SQL의 내장 함수

- COUNT(속성명) : 튜플 수를 구하는 함수
- MAX(속성명) : 최대값을 구하는 함수
- MIN(속성명) : 최소값을 구하는 함수
- SUM(속성명) : 합계를 구하는 함수
- AVG(속성명) : 평균을 구하는 함수

21. Section 106

그룹화 속성을 명시하기 위해서는 GROUP BY절을 사용해야 한다.

22. Section 105

정렬

- 질의를 수행해서 얻은 튜플을 사용자가 정렬할 수 있도록 한다.
- ORDER BY절을 사용한다.
- 기본 정렬 방식은 오름차순이다.

- ASC, DESC 키워드를 사용하여 사용자가 각각 오름차순 정렬, 내림차순 정렬 방식을 지정할 수 있다.

23. Section 105

```
SELECT 〈속성 리스트〉
FROM 〈테이블 리스트〉
WHERE 〈조건〉
```

- **〈속성 리스트〉** : 직원의 '이름(name)'과 그들이 사는 '도시(city)'를 찾으라고 하였으므로 속성 'name'과 'city'를 기술한다. 단, 〈테이블 리스트〉에서 첫 번째로 기술한 테이블에 대한 속성은 소속 표시를 하지 않아도 된다.
- **〈테이블 리스트〉** : 직원들에 대한 테이블 'works'와 그들이 사는 위치에 대한 테이블 'lives'를 기술한다.
- **〈조건〉** : company가 "제일은행"인 튜플을 찾고, 찾은 튜플의 직원(name)이 사는 도시를 찾도록 조건을 지정한다. 즉 'works' 테이블에서 'company'가 "제일은행"이고 그 직원의 이름(name)과 일치하는 이름을 가진 튜플을 lives에서 찾도록 표현한다. 따라서 company = '제일은행' and works.name = lives.name으로 기술한다.

※ company 속성이 From 절에 기술된 두 테이블에 중복된 속성이 아니므로 테이블명을 생략할 수 있다.

24. Section 104

명령 하나로 한 개의 테이블에만 삽입시킬 수 있다.

25. Section 104

문제의 지문을 절별로 구분하면 SQL 문은 다음과 같다.

- **'학생' 테이블에서 삭제하시오.** : DELETE FROM 학생
- **'학번'이 1000번인 학생을 대상으로 하시오.** : WHERE 학번 = 1000;

26. Section 104

변경할 속성과 그들의 새로운 값을 명시하기 위해서는 SET 절이 사용된다.

UPDATE문의 일반 형식

```
UPDATE 테이블명
SET 속성명 = 데이터[, 속성명=데이터]
WHERE 조건;
```

27. Section 104

UPDATE 명령은 WHERE절에서 정의된 조건에 따라 SET절에서 정의된 식으로 갱신한다.

28. Section 104

한 개의 DELETE문에는 한 개의 테이블명만 사용할 수 있다.

29. Section 105

① LIKE '%신%' : 공급자명에 "신"이 포함된 레코드

공급자번호	공급자명	위치
16	대신공업사	수원
70	신촌상사	서울

② LIKE '대%' : 공급자명이 "대"로 시작하는 레코드

공급자번호	공급자명	위치
16	대신공업사	수원

③ LIKE '%사' : 공급자명이 "사"로 끝나는 레코드

공급자번호	공급자명	위치
16	대신공업사	수원
27	삼진사	서울
39	삼양사	인천
62	진아공업사	대전
70	신촌상사	서울

④ LIKE '_사' : 공급자명이 "사"로 끝나고 두 글자인 레코드

공급자번호	공급자명	위치

30. Section 105

문제의 질의문은 하위 질의가 있는 질의문이다. 먼저 WHERE 조건에 지정된 하위 질의의 SELECT문을 검색한 다음 그 검색 결과를 본 질의의 조건에 있는 '책번호' 속성과 비교한다.

❶ SELECT 책번호 FROM 도서 WHERE 책명 = '운영체제'; : 〈도서〉 테이블에서 '책명' 속성의 값이 "운영체제"와 같은 레코드의 '책번호' 속성의 값을 검색한다. 결과는 "1111"이다.

❷ SELECT 가격 FROM 도서가격 WHERE 책번호 = '1111'; : 〈도서가격〉 테이블에서 '책번호' 속성의 값이 "1111"과 같은 레코드의 '가격' 속성의 값을 검색한다. 결과는 15000이다.

4 장

SQL 활용

108 프로시저(Procedure) A 등급

109 트리거(Trigger) B 등급

110 사용자 정의 함수 B 등급

111 DBMS 접속 기술 B 등급

112 SQL 테스트 B 등급

113 ORM(Object-Relational Mapping) B 등급

114 쿼리 성능 최적화 C 등급

이 장에서 꼭 알아야 할 키워드 Best 10

1. 프로시저 **2.** 트리거 **3.** 웹 응용 시스템 **4.** JDBC **5.** ODBC **6.** 동적 SQL **7.** 사용자 정의 함수 **8.** ORM **9.** 쿼리 성능 최적화 **10.** 절차형 SQL 테스트

SECTION 108 ● LIVE

프로시저(Procedure)

전문가의 조언

실무에서는 응용 프로그램의 유지보수와 실행을 보다 쉽게 하기 위해 절차형 SQL을 많이 사용합니다. 대표적인 절차형 SQL인 프로시저의 생성, 실행, 제거 구문이 어떤 형태로 작성되는지 알아두세요.

절차형 SQL
절차형 SQL은 C, JAVA 등의 프로그래밍 언어와 같이 연속적인 실행이나, 분기, 반복 등의 제어가 가능한 SQL을 의미합니다.

트랜잭션 언어
트랜잭션 언어는 데이터베이스를 조작하고 트랜잭션을 처리하는 언어로, SQL과 TCL이 트랜잭션 언어에 속합니다.

1 프로시저(Procedure)의 개요

프로시저란 절차형 SQL*을 활용하여 특정 기능을 수행하는 일종의 트랜잭션 언어*로, 호출을 통해 실행되어 미리 저장해 놓은 SQL 작업을 수행한다.

- 프로시저를 만들어 데이터베이스에 저장하면 여러 프로그램에서 호출하여 사용할 수 있다.
- 프로시저는 데이터베이스에 저장되어 수행되기 때문에 스토어드(Stored) 프로시저라고도 불린다.
- 프로시저는 시스템의 일일 마감 작업, 일괄(Batch) 작업 등에 주로 사용된다.

프로시저의 구성도

데이터 ➡ 프로시저 ➡ 결과

```
프로시저
DECLARE (필수)
BEGIN (필수)
  • CONTROL
  • SQL
  • EXCEPTION
  • TRANSACTION
END (필수)
```

- **DECLARE** : 프로시저의 명칭, 변수, 인수, 데이터 타입을 정의하는 선언부이다.
- **BEGIN / END** : 프로시저의 시작과 종료를 의미한다.
- **CONTROL** : 조건문 또는 반복문이 삽입되어 순차적으로 처리된다.
- **SQL** : DML, DCL이 삽입되어 데이터 관리를 위한 조회, 추가, 수정, 삭제 작업을 수행한다.
- **EXCEPTION** : BEGIN ~ END 안의 구문 실행 시 예외가 발생하면 이를 처리하는 방법을 정의한다.
- **TRANSACTION** : 수행된 데이터 작업들을 DB에 적용할지 취소할지를 결정하는 처리부이다.

2 프로시저 생성

프로시저를 생성하기 위해서는 CREATE PROCEDURE 명령어를 사용한다.

표기 형식

```
CREATE [OR REPLACE] PROCEDURE 프로시저명(파라미터)
[지역변수 선언]
BEGIN
  프로시저 BODY;
END;
```

- **OR REPLACE** : 선택적인(Optional) 예약어이다. 이 예약어를 사용하면 동일한 프로시저 이름이 이미 존재하는 경우, 기존의 프로시저를 대체할 수 있다.
- **프로시저명** : 생성하려는 프로시저의 이름을 지정한다.
- **파라미터** : 프로시저 파라미터로는 다음과 같은 것들이 올 수 있다.
  - IN : 호출 프로그램이 프로시저에게 값을 전달할 때 지정한다.
  - OUT : 프로시저가 호출 프로그램에게 값을 반환할 때 지정한다.
  - INOUT : 호출 프로그램이 프로시저에게 값을 전달하고, 프로시저 실행 후 호출 프로그램에 값을 반환할 때 지정한다.
  - **매개변수명** : 호출 프로그램으로부터 전달받은 값을 저장할 변수의 이름을 지정한다.
  - **자료형** : 변수의 자료형을 지정한다.
- **프로시저 BODY**
  - 프로시저의 코드를 기록하는 부분이다.
  - BEGIN에서 시작하여 END로 끝나며, BEGIN과 END 사이에는 적어도 하나의 SQL문이 있어야 한다.

예제 '사원번호'를 입력받아 해당 사원의 '지급방식'을 "S"로 변경하는 프로시저를 생성하시오.

```
❶ CREATE OR REPLACE PROCEDURE emp_change_s(i_사원번호 IN INT)
❷ IS
❸ BEGIN
  ❹ UPDATE 급여 SET 지급방식 = 'S' WHERE 사원번호 = i_사원번호;
  ❺ EXCEPTION
        ❻ WHEN PROGRAM_ERROR THEN
              ❼ ROLLBACK;
  ❽ COMMIT;
❾ END;
```

해설

❶ 파라미터로 'i_사원번호'를 전달받는 프로시저 'emp_change_s'를 생성한다.
❷ 변수를 선언하는 예약어로, 변수를 사용하지 않으므로 예약어만 입력한다.
❸ 프로시저 BODY의 시작을 알리는 예약어로, ❹부터 ❽까지가 하나의 블록이 된다.
❹ 〈급여〉 테이블에서 '사원번호'가 'i_사원번호'로 받은 값과 같은 튜플의 '지급방식'을 "S"로 갱신한다.
❺ 예외처리의 시작을 알리는 예약어이다.

전문가의 조언

왼쪽의 예제는 Oracle로 작성된 프로시저입니다. DBMS마다 작성방법이 조금씩 다르지만 구성 요소는 동일하니 각 구성 요소가 어떤 역할을 하는지만 정확히 파악해 두세요.

예외의 조건
예외의 조건(WHEN~THEN)에는 DBMS가 내부 문제로 종료(PROGRAM_ERROR)되었을 때뿐만이 아니라, 데이터를 찾지 못했을 때, UNIQUE 옵션을 갖는 속성에 중복 데이터를 삽입할 때, 0으로 나누려 했을 때 등 여러 조건을 삽입할 수 있습니다.

❻ SQL이 DBMS 내부 문제로 종료*되었을 때 다음 문장을 수행한다.
❼ ERROR가 발생할 경우 수행되는 문장으로, ROLLBACK을 수행한다.
❽ ❹에서 변경한 내역을 데이터베이스에 반영하는 트랜잭션 명령어이다.
❾ 프로시저 BODY의 종료를 알리는 예약어이다.

3 프로시저 실행

프로시저를 실행하기 위해서는 EXECUTE 명령어 또는 CALL 명령어를 사용하며, EXECUTE 명령어를 줄여서 EXEC로 사용하기도 한다.

표기 형식

```
EXECUTE 프로시저명;
EXEC 프로시저명;
CALL 프로시저명;
```

예제 '사원번호' 32를 인수로 하여 위에서 생성된 emp_change_s 프로시저를 실행하시오.

```
EXECUTE emp_change_s(32);
```

4 프로시저 제거

프로시저를 제거하기 위해서는 DROP PROCEDURE 명령어를 사용한다.

표기 형식

```
DROP PROCEDURE 프로시저명;
```

예제 위에서 생성된 프로시저 emp_change_s를 제거하시오.

```
DROP PROCEDURE emp_change_s;
```

기출문제 따라잡기

Section 108

출제예상

1. 프로시저에 대한 설명으로 옳지 않은 것은?

① 절차형 SQL을 사용하는 트랜잭션 언어이다.

② 데이터베이스에 저장되어 수행되기 때문에 스토어드 프로시저라고도 부른다.

③ 프로시저는 다시 호출하여 사용할 수 없다.

④ 시스템의 일일 마감 작업, 배치 작업 등에 주로 사용된다.

프로시저는 데이터베이스에 저장되므로 어느 프로그램에서나 호출하여 반복 사용할 수 있습니다.

출제예상

2. 프로시저의 구성 요소에 대한 설명으로 거리가 먼 것은?

① DECLARE : 프로시저의 명칭이나 인수, 변수 등을 정의하는 부분

② CONTROL : 조건문이나 반복문을 삽입하여 데이터를 처리하는 부분

③ TRANSACTION : 앞에서 수행한 작업들에 대한 DB 적용 여부를 결정하는 부분

④ EXCEPTION : 파라미터에 오류가 발생했을 때 이를 처리하는 방법을 정의하는 부분

EXCEPTION은 BEGIN~END 영역 안에서 경고나 오류가 발생한 경우 이를 처리하는 방법을 정의하는 부분입니다.

출제예상

3. 데이터베이스에서 프로그래밍 언어와 같이 연속적인 실행이나, 분기, 반복 등의 제어가 가능한 SQL을 무엇이라 하는가?

① 함수 ② 인덱스

③ 절차형 SQL ④ SQL*PLUS

보기 중 이번 섹션에서 공부한 것은 하나밖에 없네요.

출제예상

4. DB에 저장된 'new_Test'라는 프로시저를 제거하고자 할 때 작성해야할 SQL문으로 옳은 것은?

① ALTER PROCEDURE new_Test;

② DROP PROCEDURE new_Test;

③ DELETE PROCEDURE new_Test;

④ ERASE PROCEDURE new_Test;

프로시저 생성은 CREATE, 실행은 EXECUTE, 제거는 DROP! 잊지마세요.

출제예상

5. 다음 DDL을 통해 생성된 프로시저 new_Procedure01에 대한 설명으로 옳지 않은 것은?

```
CREATE PROCEDURE new_Procedure01 (INOUT ind_
Num integer)
BEGIN
        .....
END
```

① new_Procedure01이라는 이름의 프로시저가 DB에 이미 존재하는 경우 새로 생성된 프로시저로 대체한다.

② 호출 프로그램으로부터 integer 자료형의 데이터를 입력받아 ind_Num에 저장한다.

③ 호출 프로그램에 ind_Num에 저장된 integer 자료형의 데이터를 반환한다.

④ 생성된 프로시저는 EXECUTE new_Procedure01([숫자]) 를 통해 실행할 수 있다.

동일한 이름의 프로시저가 존재할 때 이를 대체하여 생성하도록 하려면 CREATE문에 'OR REPLACE'를 추가하여 'CREATE OR REPLACE PROCEDURE'로 입력해야 합니다.

출제예상

6. 프로시저를 실행 또는 호출하기 위한 표기 형식에 해당하지 않는 것은?

① EXEC [프로시저명];

② EXECUTE [프로시저명];

③ LOAD [프로시저명];

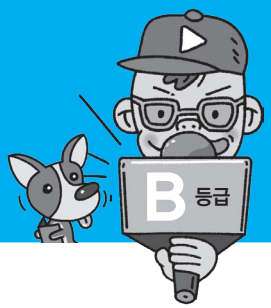
④ CALL [프로시저명];

DB에 저장된 프로시저를 실행하기 위해서는 실행이라는 의미의 EXECUTE와 그 약어 EXEC, 호출이라는 의미의 CALL이라는 예약어가 사용됩니다.

▶ 정답 : 1. ③ 2. ④ 3. ③ 4. ② 5. ① 6. ③

SECTION 109 ● LIVE

트리거(Trigger)



전문가의 조언

테이블에 어떤 데이터가 입력되었을 때 그 데이터와 관련된 작업을 자동으로 수행되게 하려면 트리거를 사용하여 간단하게 해결할 수 있습니다. 트리거의 용도와 사용 방법을 확실히 알아두세요.

이벤트(Event)
이벤트는 시스템에 어떤 일이 발생한 것을 말하며, 트리거에서 이벤트는 데이터의 삽입, 갱신, 삭제와 같이 데이터 조작 작업이 발생했음을 의미합니다.

무결성(Integrity)
무결성은 데이터베이스에 들어 있는 데이터의 정확성을 보장하기 위해 정확하지 않은 데이터가 데이터베이스 내에 저장되는 것을 방지하기 위한 제약 조건을 말합니다.

로그 메시지 출력
사용자가 컴퓨터에 요청한 명령이나 컴퓨터가 데이터를 처리하는 과정 및 결과 등을 기록으로 남긴 것을 로그(Log)라고 하며, 이것을 메시지로 출력할 때 트리거를 이용할 수 있습니다.

1 트리거(Trigger)의 개요

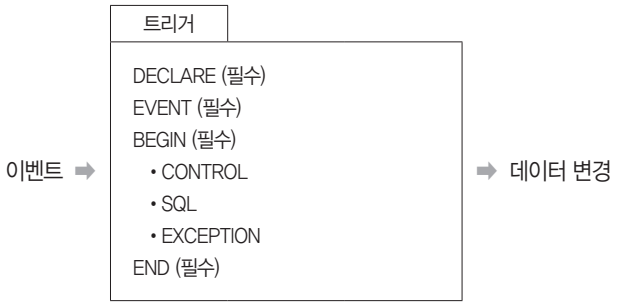
트리거는 데이터베이스 시스템에서 데이터의 삽입(Insert), 갱신(Update), 삭제(Delete) 등의 이벤트(Event)*가 발생할 때마다 관련 작업이 자동으로 수행되는 절차형 SQL이다.

- 트리거는 데이터베이스에 저장되며, 데이터 변경 및 무결성* 유지, 로그 메시지 출력* 등의 목적으로 사용된다.
- 트리거의 구문에는 DCL(데이터 제어어)을 사용할 수 없으며, DCL이 포함된 프로시저나 함수를 호출하는 경우에도 오류가 발생한다.
- 트리거에 오류가 있는 경우 트리거가 처리하는 데이터에도 영향을 미치므로 트리거를 생성할 때 세심한 주의가 필요하다.

2 트리거의 구성

트리거는 선언, 이벤트, 시작, 종료로 구성되며, 시작과 종료 구문 사이에는 제어(CONTROL), SQL, 예외(EXCEPTION)가 포함된다.

트리거 구성도



- **DECLARE** : 트리거의 명칭, 변수 및 상수, 데이터 타입을 정의하는 선언부이다.
- **EVENT** : 트리거가 실행되는 조건을 명시한다.
- **BEGIN / END** : 트리거의 시작과 종료를 의미한다.
- **CONTROL** : 조건문 또는 반복문이 삽입되어 순차적으로 처리된다.
- **SQL** : DML문이 삽입되어 데이터 관리를 위한 조회, 추가, 수정, 삭제 작업을 수행한다.
- **EXCEPTION** : BEGIN ~ END 안의 구문 실행 시 예외가 발생하면 이를 처리하는 방법을 정의한다.

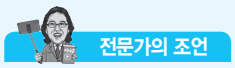
3 트리거의 생성

트리거를 생성하기 위해서는 CREATE TRIGGER 명령어를 사용한다.

표기 형식

```
CREATE [OR REPLACE] TRIGGER 트리거명 [동작시기 옵션][동작 옵션] ON 테이블명
REFERENCING [NEW | OLD] AS 테이블명
FOR EACH ROW
[WHEN 조건식]
BEGIN
  트리거 BODY;
END;
```

- **OR REPLACE** : 선택적인(Optional) 예약어이다. 이 예약어를 사용하면 동일한 트리거 이름이 이미 존재하는 경우, 기존의 프로시저를 대체할 수 있다.
- **동작시기 옵션** : 트리거가 실행될 때를 지정한다. 옵션에는 AFTER와 BEFORE가 있다.
  - AFTER : 테이블이 변경된 후에 트리거가 실행된다.
  - BEFORE : 테이블이 변경되기 전에 트리거가 실행된다.
- **동작 옵션** : 트리거가 실행되게 할 작업의 종류를 지정한다. 옵션에는 INSERT, DELETE, UPDATE가 있다.
  - INSERT : 테이블에 새로운 튜플을 삽입할 때 트리거가 실행된다.
  - DELETE : 테이블의 튜플을 삭제할 때 트리거가 실행된다.
  - UPDATE : 테이블의 튜플을 수정할 때 트리거가 실행된다.
- **NEW | OLD** : 트리거가 적용될 테이블의 별칭을 지정한다.
  - NEW : 추가되거나 수정에 참여할 튜플들의 집합(테이블)을 의미한다.
  - OLD : 수정되거나 삭제 전 대상이 되는 튜플들의 집합(테이블)을 의미한다.
- **FOR EACH ROW** : 각 튜플마다 트리거를 적용한다는 의미이다.
- **WHEN 조건식** : 선택적인(Optional) 예약어이다. 트리거를 적용할 튜플의 조건을 지정한다.
- **트리거 BODY**
  - 트리거의 본문 코드를 입력하는 부분이다.
  - BEGIN으로 시작해서 END로 끝나는데, 적어도 하나 이상의 SQL문이 있어야 한다. 그렇지 않으면 오류가 발생한다.



[동작시기 옵션]과 [동작 옵션]이 트리거가 실행되는 조건인 이벤트에 해당합니다.

전문가의 조언

오른쪽의 예제는 Oracle로 작성된 프로시저입니다. DBMS마다 작성 방법이 조금씩 다르지만 구성 요소는 동일하니 각 구성 요소가 어떤 역할을 하는지만 정확히 파악해 두세요.

전문가의 조언

'학년' 속성에는 '신입생', '1학년', '2학년', '3학년', '4학년' 값만이 올 수 있다는 도메인 무결성이 존재한다면 '학년정보_tri'를 이용하여 이러한 도메인 무결성을 위배하지 않고 튜플들을 삽입할 수 있습니다.

전문가의 조언

트리거는 데이터가 변경될 때 자동으로 수행되므로 호출문이 존재하지 않습니다.

예제 〈학생〉 테이블에 새로운 튜플이 삽입될 때, 삽입되는 튜플에 학년 정보가 누락됐으면 '학년' 필드에 "신입생"을 치환하는 트리거를 '학년정보_tri'라는 이름으로 정의하시오.

```
❶ CREATE TRIGGER 학년정보_tri BEFORE INSERT ON 학생
❷ REFERENCING NEW AS new_table
❸ FOR EACH ROW
❹ WHEN (new_table.학년 IS NULL)
   BEGIN
   ❺ :new_table.학년 := '신입생';
   END;
```

코드 해설

❶ 〈학생〉 테이블에 튜플을 삽입하기 전에 동작하는 트리거 '학년정보_tri'를 생성한다.
❷ 새로 추가될 튜플들의 집합 NEW의 별칭을 〈new_table〉로 명명한다.
❸ 모든 튜플을 대상으로 한다.
❹ 〈new_table〉에서 '학년' 속성이 NULL인 튜플에 '학년정보_tri'가 적용된다.
❺ 〈new_table〉의 '학년' 속성에 "신입생"을 치환한다.
– ❷에서 NEW 또는 OLD로 지정된 테이블 이름 앞에는 콜론(:)이 들어간다.
– A := B : A에 B를 치환하라는 의미로, '='가 아닌 ':='를 사용한다.

4 트리거의 제거

트리거를 제거하기 위해서는 DROP TRIGGER 명령어를 사용한다.

표기 형식

```
DROP TRIGGER 트리거명;
```

예제 '학년정보_tri'라는 트리거를 제거하는 SQL문을 작성하시오.

```
DROP TRIGGER 학년정보_tri;
```

기출문제 따라잡기

Section 109

출제예상

1. 절차형 SQL에서 트리거(Trigger)에 대한 설명으로 옳지 않은 것은?

① 다른 절차형 SQL과 다르게 외부에서 필드나 변수 데이터를 주고받지 않는다.

② DCL을 사용할 수 있을 뿐만 아니라, DCL이 포함된 다른 절차형 SQL을 호출하여 사용할 수도 있다.

③ 데이터베이스에 저장되어 데이터 변경 및 무결성 유지 등을 위해 사용된다.

④ 트리거에 문제가 발생하는 경우 DB에 저장된 데이터에도 영향을 주기 때문에 생성할 때 세심한 주의를 기울여야 한다.

트리거의 구문에는 DCL을 사용할 수 없습니다.

출제예상

2. 절차형 SQL 중 트리거(Trigger)의 목적에 해당하지 않는 것은?

① 데이터 변경

② 무결성 유지

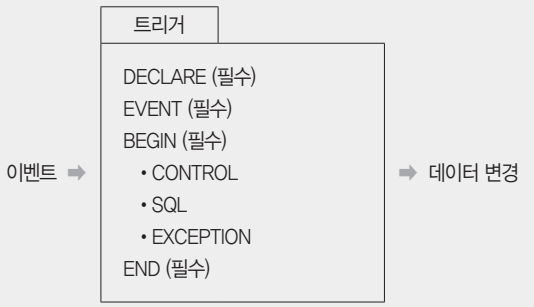
③ 로그 메시지 출력

④ 권한 변경

권한을 변경하려면 GRANT, REVOKE 등의 DCL을 사용해야 합니다.

출제예상

3. 다음 그림은 트리거의 구성도를 나타내고 있다. 구성 요소에 대한 설명으로 옳지 않은 것은?



① DECLARE는 트리거의 선언부로 모든 절차형 SQL에 포함된다.

② EVENT에는 테이블에 변화를 주는 CREATE, ALTER 등이 들어간다.

③ CONTROL은 조건문 또는 반복문이 삽입되어 실행 흐름을 제어한다.

④ EXCEPTION은 블록에서 예외가 발생하는 경우 이를 처리하기 위한 방법들을 정의한다.

트리거에서 이벤트(EVENT)는 데이터의 삽입, 갱신, 삭제와 같은 데이터 조작 작업의 발생을 의미합니다. 그렇다면 데이터의 삽입, 갱신, 삭제에 해당하는 SQL 명령문은 무엇일까요?

출제예상

4. 다음 Oracle로 작성된 SQL 문장에 대한 설명으로 잘못된 것은?

```
CREATE TRIGGER NI_Tri BEFORE INSERT ON
MEMBER
REFERENCING NEW AS New_MEMBER
FOR EACH ROW
WHEN (New_MEMBER.GRADE <> 0)
BEGIN
        New_MEMBER.NOTE := 'rejoin';
END;
```

① 트리거의 이름은 NI_Tri이고, 트리거가 수행되는 테이블은 〈MEMBER〉이다.

② 트리거는 〈MEMBER〉 테이블에 튜플이 삽입되기 전에 수행되며, 새로 추가되는 튜플들의 집합은 〈New_MEMBER〉로 명명한다.

③ 〈New_MEMBER〉의 'GRADE' 속성이 0인 튜플마다 트리거가 적용된다.

④ 〈New_MEMBER〉의 'NOTE' 속성에 "rejoin"을 치환한다.

관계 연산자 '<>'는 같지 않다는 의미입니다.

출제예상

5. 트리거(Trigger)를 제거하는 SQL 명령어는?

① DROP TRIGGER [트리거명];

② DELETE TRIGGER [트리거명];

③ ALTER TRIGGER [트리거명];

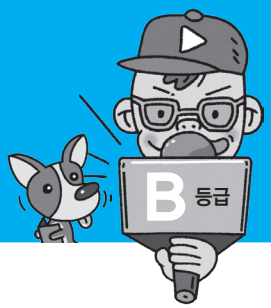
④ REMOVE TRIGGER [트리거명];

트리거 생성은 CREATE, 삭제는 DROP! 잊지마세요.

▶ 정답 : 1. ② 2. ④ 3. ② 4. ③ 5. ①

SECTION 110 ● LIVE

사용자 정의 함수



전문가의 조언

사용자 정의 함수와 프로시저의 차이점을 명확히 구분하여 이해하고, 사용자 정의 함수의 구성 요소와 사용 방법을 기억하세요.

1 사용자 정의 함수의 개요

사용자 정의 함수는 프로시저와 유사하게 SQL을 사용하여 일련의 작업을 연속적으로 처리하며, 종료 시 처리 결과를 단일값으로 반환하는 절차형 SQL이다.

- 사용자 정의 함수는 데이터베이스에 저장되어 SELECT, INSERT, DELETE, UPDATE 등 DML문의 호출에 의해 실행된다.
- 사용자 정의 함수는 예약어 RETURN을 통해 값을 반환하기 때문에 출력 파라미터가 없다.
- 사용자 정의 함수는 INSERT, DELETE, UPDATE를 통한 테이블 조작은 할 수 없고 SELECT를 통한 조회만 할 수 있다.
- 사용자 정의 함수는 프로시저를 호출하여 사용할 수 없다.
- 사용자 정의 함수는 SUM( ), AVG( ) 등의 내장 함수*처럼 DML문에서 반환값을 활용하기 위한 용도로 사용된다.

내장 함수

내장 함수는 DBMS에 기본적으로 포함되어 있는 함수들을 의미하며, 합계를 구하는 SUM( ), 평균을 구하는 AVG( ) 같은 그룹 함수가 여기에 속합니다.

잠깐만요

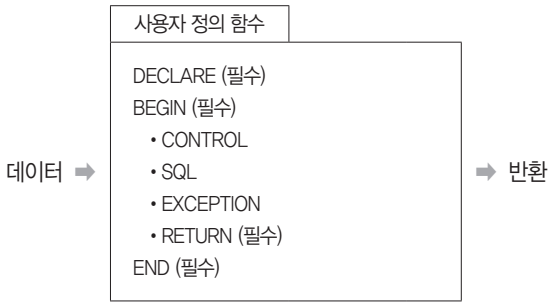
프로시저 vs 사용자 정의 함수

구분	프로시저	사용자 정의 함수
반환값	없거나 1개 이상 가능	1개
파라미터	입 · 출력 가능	입력만 가능
사용 가능 명령문	DML, DCL	SELECT
호출	프로시저, 사용자 정의 함수	사용자 정의 함수
사용 방법	실행문	DML에 포함

2 사용자 정의 함수의 구성

사용자 정의 함수의 구성은 프로시저와 유사하다. 프로시저의 구성에서 RETURN만 추가하면 된다.

사용자 정의 함수의 구성도



- **DECLARE** : 사용자 정의 함수의 명칭, 변수, 인수, 데이터 타입을 정의하는 선언부이다.
- **BEGIN / END** : 사용자 정의 함수의 시작과 종료를 의미한다.
- **CONTROL** : 조건문 또는 반복문이 삽입되어 순차적으로 처리된다.
- **SQL** : SELECT문이 삽입되어 데이터 조회 작업을 수행한다.
- **EXCEPTION** : BEGIN ~ END 안의 구문 실행 시 예외가 발생하면 이를 처리하는 방법을 정의한다.
- **RETURN** : 호출 프로그램에 반환할 값이나 변수를 정의한다.

3 사용자 정의 함수 생성

사용자 정의 함수를 생성하기 위해서는 CREATE FUNCTION 명령어를 사용한다.

표기 형식

```
CREATE [OR REPLACE] FUNCTION 사용자 정의 함수명(파라미터)
[지역변수 선언]
BEGIN
  사용자 정의 함수 BODY;
  RETURN 반환값;
END;
```

- **OR REPLACE** : 선택적인(Optional) 예약어이다. 이 예약어를 사용하면 동일한 사용자 정의 함수의 이름이 이미 존재하는 경우, 기존의 사용자 정의 함수를 대체할 수 있다.
- **파라미터** : 사용자 정의 함수의 파라미터로는 다음과 같은 것들이 올 수 있다.

– IN : 호출 프로그램이 사용자 정의 함수에게 값을 전달할 때 지정한다.

– 매개변수명 : 호출 프로그램으로부터 전달받은 값을 저장할 변수의 이름을 지정한다.

– 자료형 : 변수의 자료형을 지정한다.

- **사용자 정의 함수 BODY**
  – 사용자 정의 함수의 코드를 기록하는 부분이다.
  – BEGIN에서 시작하여 END로 끝나며, BEGIN과 END 사이에는 적어도 하나의 SQL문이 있어야 한다.
- **RETURN 반환값** : 반환할 값이나 반환할 값이 저장된 변수를 호출 프로그램으로 돌려준다.

예제 'i_성별코드'를 입력받아 1이면 "남자"를, 2면 "여자"를 반환하는 사용자 정의 함수를 'Get_S_성별'이라는 이름으로 정의하시오.

```
❶ CREATE FUNCTION Get_S_성별(i_성별코드 IN INT)
❷ RETURN VARCHAR2
❸ IS
BEGIN
  ❹ IF (i_성별코드 = 1) THEN
          RETURN '남자';
  ❺ ELSE
          RETURN '여자';
  ❻ END IF;
END;
```

해설

❶ 파라미터로 'i_성별코드'를 전달받는 사용자 정의 함수 'Get_S_성별'을 생성한다.
❷ 블록에서 리턴할 데이터의 자료형을 정의한다. 자료형의 크기는 입력할 필요 없다.
– 형식 : RETURN [자료형]
❸ 변수 선언을 위해 사용하는 예약어로, 변수를 사용하지 않으므로 예약어만 입력한다.
❹ 'i_성별코드'가 1이면 "남자"를 반환하고,
❺ 'i_성별코드'가 1이 아니면 "여자"를 반환한다.
❻ IF문의 끝

4 사용자 정의 함수 실행

사용자 정의 함수는 DML에서 속성명이나 값이 놓일 자리를 대체하여 사용된다.

표기 형식

```
SELECT 사용자 정의 함수명 FROM 테이블명;
INSERT INTO 테이블명(속성명) VALUES (사용자 정의 함수명);
DELETE FROM 테이블명 WHERE 속성명 = 사용자 정의 함수명;
UPDATE 테이블명 SET 속성명 = 사용자 정의 함수명;
```

전문가의 조언

오른쪽의 예제는 Oracle로 작성된 프로시저입니다. DBMS마다 작성 방법이 조금씩 다르지만 구성 요소는 동일하니 각 구성 요소가 어떤 역할을 하는지만 정확히 파악해 두세요.

전문가의 조언

Oracle에서는 가변길이 문자 자료형에 VARCHAR 외에 VARCHAR2가 추가로 있으며, Oracle 사는 개발자들에게 안정성을 위해 VARCHAR2를 주로 사용할 것을 권고합니다.

전문가의 조언

IF문은 [조건]에 따라 수행해야 하는 코드가 다를 경우 사용하는 명령어입니다. 사용 방법을 간단히 알아두세요.

```
IF ([조건]) THEN
  [조건이 참일 때 수행할 코드];
ELSE
  [조건이 거짓일 때 수행할 코드];
END IF;
```

예제 다음의 〈사원〉 테이블을 출력하되, '성별코드'는 앞에서 사용자 정의 함수 'Get_S_성별'에 값을 전달하여 반환받은 값으로 대체하여 출력하시오.

〈사원〉

이름	성별코드
김대진	1
이고을	2
최승규	1
송하나	2

```
SELECT 이름, Get_S_성별(성별코드) FROM 사원;
```

〈결과〉

이름	Get_S_성별(성별코드)
김대진	남자
이고을	여자
최승규	남자
송하나	여자

해설 〈사원〉 테이블에서 '이름' 속성과 앞에서 정의한 사용자 정의 함수 'Get_S_성별'에 '성별코드' 속성을 인수로 전달하고 반환받은 값을 결과로 출력한다. '성별코드'의 값이 1이면 "남자"가 출력되고, 1이 아니면 "여자"가 출력된다.

5 사용자 정의 함수 제거

사용자 정의 함수를 제거하기 위해서는 DROP FUNCTION 명령어를 사용한다.

표기 형식

```
DROP FUNCTION 사용자 정의 함수명;
```

예제 앞에서 생성된 사용자 정의 함수 'Get_S_성별'을 제거하시오.

```
DROP FUNCTION Get_S_성별;
```

출제예상

1. DBMS에서 사용자 정의 함수에 대한 설명으로 옳은 것은?

① 데이터베이스 시스템에 이벤트가 발생할 때 자동으로 수행되는 절차형 SQL이다.

② 블록 내에 다른 사용자 정의 함수나 프로시저를 호출하여 사용할 수 있다.

③ SELECT나 INSERT 등의 DML에 포함되어 실행된다.

④ 출력 파라미터를 이용하여 2개 이상의 값을 반환할 수 있다.

①번은 트리거(Trigger), ②, ④번은 프로시저(Procedure)에 대한 설명입니다.

출제예상

2. 프로시저와 사용자 정의 함수에 대한 설명으로 옳지 않은 것은?

① 프로시저와 사용자 정의 함수는 파라미터를 통해 값을 전달받거나 반환할 수 있다.

② 프로시저와 사용자 정의 함수는 절차형 SQL로서 단일 SQL 문장으로는 처리할 수 없는 연속적인 작업들을 수행하는데 도움을 준다.

③ 프로시저는 블록 내에 테이블을 조작하기 위해 SELECT, INSERT, DELETE 등 다양한 DML을 사용할 수 있지만, 사용자 정의 함수는 SELECT만 사용할 수 있다.

④ 별도로 실행하지 않고 DML에 포함하여 실행하는 방법을 사용하는 것은 사용자 정의 함수이다.

프로시저의 파라미터 옵션에는 IN, OUT, INOUT이 있지만, 사용자 정의 함수에는 IN만 있습니다.

출제예상

3. 직위와 연차를 이용하여 연봉을 계산하는 사용자 정의 함수 salary를 이용하여 사원(사원번호, 이름, 직위, 연차) 테이블에서 사원번호, 이름, 연봉을 조회하는 명령문으로 옳은 것은?

① SELECT 사원번호, 이름, salary AS 연봉 FROM 사원;

② SELECT 사원번호, 이름, salary(직위, 연차) AS 연봉 FROM 사원;

③ SELECT 사원번호, 이름, 연봉 AS salary(직위, 연차) FROM 사원;

④ SELECT 사원번호, 이름, 연봉 AS salary FROM 사원;

AS는 속성의 별칭을 지정하는 예약어로 '속성명 AS 별칭명'을 사용하며, 사용자 정의 함수는 속성명 자리에 사용할 수 있습니다.

출제예상

4. 다음은 태어난 해와 올해의 연도를 입력받아 현재 나이를 반환하는 사용자 정의 함수의 생성문이다. 빈 칸에 들어갈 단어로 알맞게 짝지어진 것은?

```
CREATE FUNCTION Get_Age (i_BirthYear IN INT, i_
NowYear IN INT)
RETURN INT
IS
        AGE INT;
(   )
        SET AGE := (i_NowYear - i_BirthYear) + 1;
        (   ) AGE;
END;
```

① OPEN, EXCEPTION

② OPEN, DECLARE

③ BEGIN, DECLARE

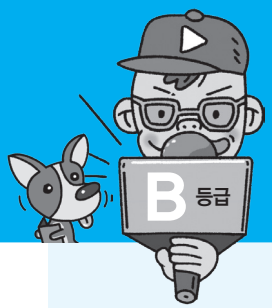
④ BEGIN, RETURN

사용자 정의 함수의 생성문에는 BEGIN, END, RETURN이 반드시 들어가야 합니다.

▶ 정답 : 1. ③ 2. ① 3. ② 4. ④

SECTION 111 LIVE

DBMS 접속 기술



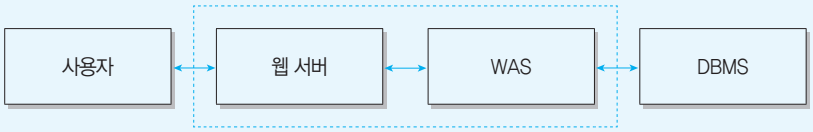
1 DBMS 접속의 개요

DBMS 접속은 사용자가 데이터를 사용하기 위해 응용 시스템을 이용하여 DBMS에 접근하는 것을 의미한다.

- 응용 시스템은 사용자로부터 매개 변수를 전달받아 SQL을 실행하고 DBMS로부터 전달받은 결과를 사용자에게 전달하는 매개체 역할을 수행한다.
- 인터넷을 통해 구동되는 웹 응용 프로그램은 웹 응용 시스템*을 통해 DBMS에 접근한다.
- 웹 응용 시스템은 웹 서버*와 웹 애플리케이션 서버(WAS*)로 구성되며, 서비스 규모가 작은 경우 웹 서버와 웹 애플리케이션 서버를 통합하여 하나의 서버만으로 운용할 수 있다.

잠깐만요

웹 응용 시스템의 구조



- 사용자는 웹 서버에 접속하여 데이터를 주고받습니다.
- 웹 서버는 많은 수의 서비스 요청을 처리하기 때문에 사용자가 대용량의 데이터를 요청하면 직접 처리하지 않고 WAS에게 해당 요청을 전달합니다.
- WAS는 수신한 요청을 트랜잭션 언어로 변환한 후 DBMS에 전달하여 데이터를 받습니다. 이렇게 받은 데이터는 처음 요청한 웹 서버로 다시 전달되어 사용자에게까지 도달하게 됩니다.

2 DBMS 접속 기술

DBMS 접속 기술은 DBMS에 접근하기 위해 사용하는 API* 또는 API의 사용을 편리하게 도와주는 프레임워크* 등을 의미한다.

JDBC(Java DataBase Connectivity)

JDBC는 Java 언어로 다양한 종류의 데이터베이스에 접속하고 SQL문을 수행할 때 사용되는 표준 API이다.

- 1997년 2월 썬 마이크로시스템에서 출시했다.

전문가의 조언

DBMS 접속에 대한 개념을 이해하고, DBMS 접속 기술의 종류와 동적 SQL에 대해 숙지하세요.

웹 응용 시스템

웹 응용 시스템은 이메일 사이트와 비교하면 이해가 쉽습니다. 예를 들어, 이메일을 확인하기 위해 사이트에 접속한다고 가정할 때, 사이트에 접속하기 위해 사용하는 웹브라우저는 '웹 응용 프로그램'이고, 접속한 사이트에서 보여주는 웹 페이지의 내용은 '웹 서버'에서 송출되는 것입니다. 로그인 후 이메일을 확인하기 위해 받은 편지함을 클릭하면 '웹 서버'는 받은 편지 목록에 대한 요청을 'WAS'에게 보내고, 'WAS'는 'DBMS'로부터 데이터를 가져와 '웹 서버'에 전달함으로써 사용자는 받은 편지 목록을 확인할 수 있습니다.

웹 서버/WAS

웹 서버는 웹 페이지, 저용량 자료 등의 처리를 담당하고, WAS는 DBMS와 통신하며 대용량 파일이나 복잡한 로직 등의 처리를 담당합니다.

API(Application Programming Interface)

API는 응용 프로그램 개발 시 운영체제나 DBMS 등을 이용할 수 있도록 규칙 등에 대해 정의해 놓은 인터페이스를 의미합니다.

프레임워크(Framework)

프레임워크는 본래 '뼈대', '골조'를 의미하는 용어로, 소프트웨어에서는 특정 기능을 수행하기 위해 필요한 클래스나 인터페이스 등을 모아둔 집합체를 가리킵니다.

- JDBC는 Java SE(Standard Edition)*에 포함되어 있으며, JDBC 클래스는 java.sql, javax.sql에 포함되어 있다.
- 접속하려는 DBMS에 대한 드라이버*가 필요하다.

ODBC(Open DataBase Connectivity)

ODBC는 데이터베이스에 접근하기 위한 표준 개방형 API로, 개발 언어에 관계없이 사용할 수 있다.

- 1992년 9월 마이크로소프트에서 출시했다.
- 프로그램 내 ODBC 문장을 사용하여 MS-Access, DBase, DB2, Excel, Text 등 다양한 데이터베이스에 접근할 수 있다.
- ODBC도 접속하려는 DBMS에 맞는 드라이버가 필요하지만, 접속하려는 DBMS의 인터페이스를 알지 못하더라도 ODBC 문장을 사용하여 SQL을 작성하면 ODBC에 포함된 드라이버 관리자가 해당 DBMS의 인터페이스에 맞게 연결해 주므로 DBMS의 종류를 몰라도 된다.

MyBatis

MyBatis는 JDBC 코드를 단순화하여 사용할 수 있는 SQL Mapping* 기반 오픈 소스 접속 프레임워크이다.

- JDBC로 데이터베이스에 접속하려면 다양한 메소드를 호출하고 해제해야 하는데, MyBatis는 이를 간소화 했고 접속 기능을 더욱 강화하였다.
- MyBatis는 SQL 문장을 분리하여 XML 파일을 만들고, Mapping을 통해 SQL을 실행한다.
- MyBatis는 SQL을 거의 그대로 사용할 수 있어 SQL 친화적인 국내 환경에 적합하여 많이 사용된다.

3 동적 SQL(Dynamic SQL)

동적 SQL은 개발 언어에 삽입되는 SQL 코드를 문자열 변수에 넣어 처리하는 것으로, 조건에 따라 SQL 구문을 동적으로 변경하여 처리할 수 있다.

- 동적 SQL은 사용자로부터 SQL문의 일부 또는 전부를 입력받아 실행할 수 있다.
- 동적 SQL은 값이 입력되지 않을 경우 사용하는 NVL 함수*를 사용할 필요가 없다.
- 동적 SQL은 응용 프로그램 수행 시 SQL이 변형될 수 있으므로 프리컴파일* 할 때 구문 분석, 접근 권한 확인 등을 할 수 없다.
- 동적 SQL은 정적 SQL에 비해 속도가 느리지만, 상황에 따라 다양한 조건을 첨가하는 등 유연한 개발이 가능하다.

Java SE
Java SE는 Java 표준안으로서, Java의 문법과 기능들을 정의하는 명세서입니다. 개발 도구인 JDK(Java Development Kit)에 포함되어 사용되며, JDBC의 기능들을 정의하는 클래스 파일들을 포함하고 있습니다.

드라이버
드라이버는 다른 장치나 시스템을 제어하는데 사용되는 프로그램을 의미합니다.

SQL Mapping
SQL Mapping은 SQL로 호출되는 테이블이나 열 데이터를 개발하려는 언어의 객체에 맞도록 변환하여 연결하는 것을 의미합니다.

전문가의 조언

동적 SQL은 쉽게 말해 사용자가 응용 프로그램을 실행 시킨 후 입력란에 SQL을 직접 입력하여 결과를 확인할 수 있는 것을 말합니다.

NVL 함수
NVL(A, B) 형태의 함수로, A가 NULL인 경우 B를 반환하고, 아니면 A를 반환합니다. 동적 SQL에서는 원하는 조건에 따라 자유롭게 SQL문을 바꿀 수 있으므로, NVL 함수 없이 SQL문을 구성하는 것이 가능합니다.

프리컴파일(Precompile)
프리컴파일은 고급언어를 기계어로 번역하는 컴파일(Compile) 전에 수행하는 작업으로, 필요한 라이브러리를 불러오거나 코드에 삽입된 SQL문을 DB와 연결하는 작업을 수행합니다.

잠깐만요

정적 SQL vs 동적 SQL

정적 SQL은 SQL 코드를 변수에 담지 않고 코드 사이에 직접 기술한 SQL문으로 동적 SQL과의 차이점은 다음과 같습니다.

	정적 SQL(Static SQL)	동적 SQL(Dynamic SQL)
SQL 구성	커서(Cursor)*를 통한 정적 처리	문자열 변수에 담아 동적 처리
개발 패턴	커서의 범위 안에서 반복문을 활용하여 SQL 작성	NVL 함수 없이 로직을 통해 SQL 작성
실행 속도	빠름	느림
사전 검사	가능	불가능

커서(Cursor)
커서는 SQL문의 실행 결과로 반환된 복수 개의 튜플들에 접근할 수 있도록 해주는 기능을 의미합니다.

따라잡기

기출문제 따라잡기

Section 111

출제예상

1. 사용자가 DBMS에 접속하는 과정에 대한 설명으로 옳지 않은 것은?

① 사용자는 응용 시스템을 매개체로 DBMS에 접속한다.
② 사용자가 PC나 스마트폰 등의 단말기를 통해 응용 시스템에 데이터를 요청하면 응용 시스템은 해당 요청을 정해진 로직에 따라 변환하여 DBMS에 전달한다.
③ 사용자가 웹 소프트웨어를 이용하는 경우 응용 시스템은 Web Server, WAS, DBMS로 구성된다.
④ 웹 응용 시스템의 경우 서비스 또는 시스템의 규모가 작으면 웹 서버와 WAS를 통합하여 사용할 수 있다.

웹 응용 시스템은 사용자의 요청을 받아 전달하는 웹 서버(Web Server)와 웹 서버의 요청을 받아 트랜잭션을 생성하여 DBMS에 전달하는 웹 애플리케이션 서버(WAS)로 구성됩니다.

출제예상

2. 웹 응용 프로그램에서 사용자가 데이터를 요청하고 결과를 받기까지의 순서로 옳은 것은?

① 사용자 ↔ 단말기 ↔ Web Server ↔ WAS ↔ DBMS
② 사용자 ↔ 단말기 ↔ WAS ↔ Web Server ↔ DBMS
③ 사용자 ↔ 단말기 ↔ WAS ↔ DBMS ↔ Web Server
④ 사용자 ↔ 단말기 ↔ DBMS ↔ WAS ↔ Web Server

사용자는 단말기를 통해 웹 응용 프로그램을 실행하며, 웹 응용 프로그램은 DBMS에 접근하기 위해 웹 서버와 WAS로 구성되는 웹 응용 시스템을 매개체로 사용합니다.

출제예상

3. JDBC에 대한 설명으로 옳지 않은 것은?

① JDBC는 C언어로 다양한 종류의 데이터베이스에 접속하기 위한 API이다.
② 썬 마이크로시스템에서 개발하였으며, 표준판에 포함되어 있다.
③ 접속하려는 DBMS의 벤더(Vendor)에 맞춰 드라이버를 설치해줘야 한다.
④ JDBC 클래스는 java.sql, javax.sql에 포함되어 있다.

JDBC는 Java DataBase Connectivity의 약어로, Java 언어와 관련이 있습니다.

출제예상

4. DBMS 접속 기술에서 Dynamic SQL에 대한 설명으로 옳지 않은 것은?

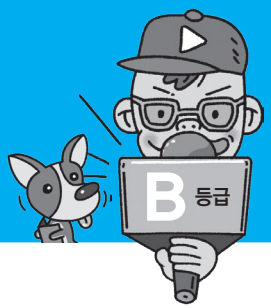
① String형 변수에 SQL문을 입력받아 처리한다.
② NULL 값을 처리할 때 주로 NVL 함수를 이용한다.
③ 사용자의 입력에 따라 SQL문이 변형될 수 있다.
④ Static SQL에 비해 유연한 로직의 개발이 가능하지만 실행 속도가 비교적 느리다.

동적(Dynamic) SQL은 SQL 문을 자유롭게 변경할 수 있어 NULL을 처리하는 함수를 사용할 필요가 없습니다.

▶ 정답 : 1. ③ 2. ① 3. ① 4. ②

SECTION 112 ● LIVE

SQL 테스트



전문가의 조언

사용하려는 SQL과 DBMS의 종류에 따라 테스트 방법 및 명령어들이 서로 다르니 잘 구분해서 알아두세요.

구문 오류(Syntax Error)
구문 오류는 잘못된 문법으로 작성된 SQL문을 실행하면 출력되는 오류를 의미합니다.

디버깅(Debugging)
디버깅은 오류를 잡기 위해 소스 코드를 한 줄 한 줄 추적해 가며 변수 값의 변화를 검사하는 작업입니다.

TCL (Transaction Control Language)
TCL은 트랜잭션을 제어하는 COMMIT, ROLLBACK, SAVEPOINT 명령을 가리키는 용어입니다.

DESCRIBE, DESC
DESCRIBE는 개체의 정보를 확인할 때 사용하는 명령어이고, DESC는 DESCRIBE의 약어로 대신 DESC를 사용해도 동일한 기능을 수행합니다.

전문가의 조언

시스템 권한을 확인할 때 MySQL에서는 DB서버의 위치를 가리키는 호스트(Host)를 적어야 합니다. 외부 DB서버에 접속해 있다면 해당 IP를 적고, PC에 설치된 DB서버라면 'localhost'를 적으면 됩니다.

1 SQL 테스트의 개요

SQL 테스트는 SQL이 작성 의도에 맞게 원하는 기능을 수행하는지 검증하는 과정이다.

- 단문 SQL은 SQL 코드를 직접 실행한 후 결과를 확인하는 것으로 간단히 테스트가 가능하다.
- 절차형 SQL은 테스트 전에 생성을 통해 구문 오류(Syntax Error)*나 참조 오류의 존재 여부를 확인한다.
- 정상적으로 생성된 절차형 SQL은 디버깅*을 통해 로직을 검증하고, 결과를 통해 최종적으로 확인한다.

2 단문 SQL 테스트

단문 SQL 테스트는 DDL, DML, DCL이 포함되어 있는 SQL과 TCL*을 테스트하는 것으로, 직접 실행하여 결과물을 확인한다.

- 실행 시 오류나 경고가 발생할 경우 메시지를 참조하여 문제를 해결한다.
- DESCRIBE* 명령어를 이용하면 DDL로 작성된 테이블이나 뷰의 속성, 자료형, 옵션들을 바로 확인할 수 있다.
  - DESC* [개체명];
- DML로 변경한 데이터는 SELECT문으로 데이터의 정상적인 변경 여부를 확인할 수 있다.
- DCL로 설정된 사용자 권한은 사용자 권한 정보가 저장된 테이블을 SELECT로 조회하거나, SHOW 명령어로 확인할 수 있다.
  - Oracle : SELECT * FROM DBA_ROLE_PRIVES WHERE GRANTEE = 사용자;
  - MySQL : SHOW GRANTS FOR 사용자@호스트;

예제 1 MySQL에서 〈학생〉 테이블의 정보를 확인하는 SQL문을 작성하시오.

```
DESC 학생;
```

〈결과〉

Field	Type	Null	Key	Default	Extra
이름	varchar(8)	No		noname	
학번	int(11)	No	PRI		
전공	char(8)	Yes			
주민번호	char(13)	No	UNI		

해설 Field는 속성명, Type은 자료형, Null은 Null 허용 여부, Key는 기본키(Primary Key)와 유니크(Unique) 옵션의 여부, Default는 기본값, Extra는 그 외의 기타 옵션을 의미한다.

예제 2 Oracle에서 'SINAGONG' 사용자의 시스템 권한을 확인하는 SQL문을 작성하시오.

```
SELECT * FROM DBA_SYS_PRIVS WHERE GRANTEE = 'SINAGONG';
```

〈결과〉

GRANTEE	PRIVILEGE	ADMIN_OPTION
SINAGONG	CREATE TABLE	NO
SINAGONG	CREATE VIEW	NO

해설 'SINAGONG'이 테이블과 뷰를 만들 권한(PRIVILEGE)을 갖고 있지만, 해당 권한을 다른 유저에게 부여할 수 있는 권한(ADMIN_OPTION)은 없다는 의미이다.

예제 3 MySQL에서 localhost 서버의 'SINAGONG' 사용자에 대한 권한을 확인하는 SQL문을 작성하시오.

```
SHOW GRANTS FOR 'SINAGONG'@'localhost';
```

〈결과〉

Grants for SINAGONG@localhost
GRANT SELECT ON *.* TO 'SINAGONG'@'localhost'

해설 localhost 서버에서 'SINAGONG' 사용자는 모든 스키마와 테이블에 대해 조회(SELECT) 권한이 있다는 의미이다.

3 절차형 SQL 테스트

프로시저, 사용자 정의 함수, 트리거 등의 절차형 SQL은 디버깅을 통해 기능의 적합성 여부를 검증하고, 실행을 통해 결과를 확인하는 테스트를 수행한다.

- 많은 코드로 구성된 절차형 SQL의 특성상 오류 및 경고 메시지가 상세히 출력되지 않으므로 SHOW 명령어를 통해 오류 내용을 확인하고 문제를 수정한다.
  - 형식 : SHOW ERRORS*;
- 데이터베이스에 변화를 줄 수 있는 SQL문은 주석*으로 처리하고, 출력문을 이용하여 화면에 출력하여 확인한다.
  - Oracle 출력 형식
    - ▶ DBMS_OUTPUT.ENABLE; : 화면에 출력하기 위해 DBMS_OUTPUT 패키지를 불러온다.
    - ▶ DBMS_OUTPUT.PUT_LINE(데이터); : '데이터'에 넣은 변수나 값을 화면에 출력한다.
  - MySQL 출력 형식
    - ▶ SELECT 데이터; : '데이터'에 넣은 변수나 값을 화면에 출력한다.

전문가의 조언

데이터베이스 전체 사용자 목록을 확인하려면 Oracle에서는 〈dba_users〉 테이블을, MySQL에서는 〈user〉 테이블을 조회하면 됩니다.

전문가의 조언

절차형 SQL 테스트는 절차형 SQL을 실행하기 전에 조작 대상이 되는 데이터 및 관련 데이터를 화면에 출력하여 정상적으로 수행되는지 확인하는 작업입니다.

Oracle 오류 확인 예

오류가 발생한 장소와 오류 코드, 오류 내용이 표시됩니다.

LINE/COL	ERROR
7/1	PLS-00363: 'N' 식은 피할당자로 사용될 수 없습니다.

주석(Comment)

설명을 위해 입력한 부분을 주석이라고 합니다. 즉 주석은 사람만 알아볼 수 있으며, 컴파일 되지도 않습니다. 때문에 잠시 사용하지 않을 SQL 코드를 주석으로 처리해 두면 지우지 않고도 해당 코드를 무시하고 SQL 문을 수행할 수 있습니다. DBMS의 종류에 따라 다음과 같이 다른 주석 기호를 사용합니다.

- # : MySQL에서 사용하는 주석 기호로, '#' 다음에 오는 한 줄을 주석으로 처리합니다.
- -- : Oracle에서 사용하는 주석 기호로, '--' 다음에 오는 한 줄을 주석으로 처리합니다.
- /* */ : DBMS에 관계없이 '/*'과 '*/' 사이에 있는 문장을 모두 주석으로 처리합니다.

전문가의 조언

Oracle은 DBMS_OUTPUT 패키지로 화면에 결과를 출력하려면 절차형 SQL을 실행하기 전에 'SET SERVEROUTPUT ON' 명령어로 환경 변수를 설정해야 합니다.

- 디버깅이 완료되면 출력문을 삭제하고, 주석 기호를 삭제한 후 절차형 SQL을 실행하여 결과를 검토한다.

예제 1 Oracle에서 변수 RESULT를 출력하는 SQL문을 작성하시오.

```
DBMS_OUTPUT.ENABLE;
DBMS_OUTPUT.PUT_LINE(RESULT);
```

해설 Oracle에서 화면에 변수나 텍스트 등을 출력하기 위해서는 DBMS_OUTPUT 패키지의 PUT_LINE 함수를 사용해야 한다. 첫 번째 행은 DBMS_OUTPUT 패키지를 사용하겠다는 의미이고, 두 번째 행은 실제 출력을 위해 PUT_LINE 함수를 사용한 것이다. 실행하면 RESULT에 기억된 값이 화면에 표시된다.

예제 2 MySQL에서 변수 RESULT를 출력하는 SQL문을 작성하시오.

```
SELECT RESULT;
```

해설 MySQL에서는 SELECT만으로 간편하게 화면에 변수나 텍스트 등을 출력할 수 있다. Oracle의 명령문을 사용하면 오류가 발생한다.

잠깐만요

Oracle 프로시저 디버깅 예

다음은 사용자에게 이름을 입력받아 〈student〉 테이블에서 같은 이름을 가진 학생의 국어, 수학, 영어 점수를 합하여 〈score〉 테이블에 삽입하는 프로시저를 디버깅하기 위해 수정한 것입니다. 진하게 강조한 코드와 주석 기호가 디버깅을 위해 추가한 코드입니다.

```
CREATE OR REPLACE PROCEDURE u_sum_avg(u_name IN CHAR)
IS
    ❶ v_sum INT;
    ❷ v_avg INT;
BEGIN
    ❸ DBMS_OUTPUT.ENABLE;
    ❹ SELECT kor+math+eng INTO v_sum FROM student WHERE name = u_name;
    ❺ v_avg := v_sum/3;
    ❻ DBMS_OUTPUT.PUT_LINE(v_sum);
    ❼ DBMS_OUTPUT.PUT_LINE(v_avg);
    ❽ -- INSERT INTO score VALUES(u_name, v_sum, v_avg);
END;
```

해설
❶ 정수를 저장할 변수 v_sum을 선언합니다.
❷ 정수를 저장할 변수 v_avg을 선언합니다.
❸ 결과를 화면에 출력하기 위해 DBMS_OUPUT 패키지를 사용 가능한 상태로 바꿉니다.
❹ 〈student〉 테이블의 name 속성에서 입력 파라미터 u_name과 같은 값을 가진 튜플을 찾아 kor, math, eng 속성의 값을 더해 변수 v_sum에 저장합니다.
❺ v_sum의 값을 3으로 나눠 v_avg에 저장합니다.
❻ v_sum의 값을 화면에 출력합니다.
❼ v_avg의 값을 화면에 출력합니다.
❽ u_name, v_sum, v_avg의 값을 〈score〉 테이블에 추가하는 SQL문이지만 주석 처리(– –)되었으므로 수행되지 않고 종료됩니다.

기출문제 따라잡기

Section 112

출제예상

1. SQL 테스트에 대한 내용으로 옳지 않은 것은?

① 작성한 SQL 코드의 기능이 의도대로 수행되는지 확인하는 과정을 말한다.

② SQL 코드를 직접 실행하여 에러 메시지를 확인하는 것도 SQL 테스트에 속한다.

③ 절차형 SQL은 생성 과정에서 구문 오류나, 참조 오류 등을 발견할 수 있다.

④ 절차형 SQL의 디버깅을 위해 SQL 코드를 수정할 필요는 없다.

절차형 SQL의 테스트를 위해서는 주석과 출력문을 삽입해야 합니다.

출제예상

2. 다음 중 CREATE 명령어를 통해 생성된 특정 개체의 속성, 자료형 등의 정보를 확인하기 위한 SQL 코드는 무엇인가?

① SHOW [개체명];

② SELECT [개체명];

③ DESC [개체명];

④ CHECK [개체명];

개체의 정보를 확인하는 예약어는 DESCRIBE이며, 약어로도 사용할 수 있습니다.

출제예상

3. 다음 SQL문을 수행하고 확인하는 방법으로 가장 올바른 것은?

```
ALTER TABLE 사원 ADD 입사일 DATE;
```

① DESCRIBE 사원;

② SHOW GRANTS FOR 사원;

③ SELECT 입사일 FROM 사원;

④ SELECT * FROM 사원;

〈사원〉 테이블의 '입사일' 속성을 추가하는 SQL문입니다. 테이블의 정보를 확인해보면 되겠네요.

출제예상

4. 프로시저를 생성하는 과정에서 발생한 오류(ERROR)를 확인하기 위해 사용하는 SQL 코드로 가장 올바른 것은?

① SHOW WARNINGS;

② SHOW ERRORS;

③ CHECK LOGSOURCE;

④ CHECK ERRORS;

절차형 SQL을 생성할 때 문제가 생기면 해당 문제를 확인하기 위해 SHOW 명령어를 사용하며, 확인하고자 하는 것은 오류입니다.

출제예상

5. 절차형 SQL 디버깅에 대한 설명으로 옳지 않은 것은?

① 절차형 SQL이 정상적으로 생성된 후 블록 내의 로직이 올바르게 수행되고 있는지 확인하기 위한 과정이다.

② 블록 내에 데이터베이스 개체의 데이터를 변경하는 SQL 코드가 있다면 주석 처리하고, 변경 내역은 화면에 출력하여 확인한다.

③ 화면에 출력하기 위해 Oracle에서는 DBMS_OUTPUT 패키지를, MySQL에서는 SELECT 명령어를 활용한다.

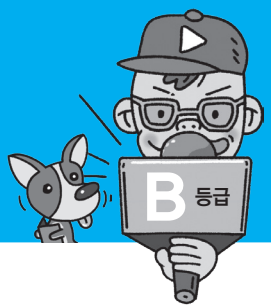
④ 여러 줄을 한 번에 주석 처리할 수 없으므로 매 행마다 SQL 코드 앞에 주석 기호를 추가해야 한다.

여러 행을 주석 처리할 때는 시작하는 부분에 '/*'를 입력하고, 끝나는 부분에 '*/'를 추가하면 됩니다.

▶ 정답 : 1. ④ 2. ③ 3. ① 4. ② 5. ④

SECTION 113 ● LIVE

ORM(Object-Relational Mapping)



전문가의 조언

ORM의 O는 객체(Object)를, R은 관계형 데이터베이스(Relational Database)를 의미한다는 것을 염두에 두고, ORM의 역할과 특징을 파악하세요.

객체지향 프로그래밍(OOP; Object Oriented Programming)
객체지향 프로그래밍은 현실 세계의 개체(Entity)를 기계의 부품처럼 하나의 객체(Object)로 만들어, 기계적인 부품들을 조립하여 제품을 만들듯이 소프트웨어를 개발할 때에도 객체들을 조립해서 작성할 수 있는 기법을 말합니다. 객체지향 프로그래밍에 대한 자세한 내용은 Section 134를 참고하세요.

1 ORM(Object-Relational Mapping)의 개요

ORM은 객체지향 프로그래밍*의 객체(Object)와 관계형 데이터베이스(Relational Database)의 데이터를 연결(Mapping)하는 기술을 의미한다.

- ORM은 객체지향 프로그래밍에서 사용할 수 있는 가상의 객체지향 데이터베이스를 만들어 프로그래밍 코드와 데이터를 연결한다.
- ORM으로 생성된 가상의 객체지향 데이터베이스는 프로그래밍 코드 또는 데이터베이스와 독립적이므로 재사용 및 유지보수가 용이하다.
- ORM은 SQL 코드를 직접 입력하지 않고 선언문이나 할당 같은 부수적인 코드가 생략되기 때문에 직관적이고 간단하게 데이터를 조작할 수 있다.

2 ORM 프레임워크

ORM 프레임워크는 ORM을 구현하기 위한 구조와 구현을 위해 필요한 여러 기능들을 제공하는 소프트웨어를 의미한다.

- ORM 프레임워크의 종류

JAVA	JPA, Hibernate, EclipseLink, DataNucleus, Ebean 등
C++	ODB, QxOrm 등
Python	Django, SQLAlchemy, Storm 등
iOS	DatabaseObjects, Core Data 등
.NET	NHibernate, DatabaseObjects, Dapper 등
PHP	Doctrine, Propel, RedBean 등

3 ORM의 한계

- ORM은 프레임워크가 자동으로 SQL을 작성하기 때문에 의도대로 SQL이 작성되었는지 확인할 필요가 있다.
- 객체지향적인 사용을 고려해야 하고, 설계된 데이터베이스가 아닌 경우 프로젝트가 크고 복잡해질수록 ORM 기술을 적용하기 어려워진다.
- 기존의 기업들은 ORM을 고려하지 않은 데이터베이스를 사용하고 있기 때문에 ORM에 적합하게 변환하려면 많은 시간과 노력이 필요하다.

기출문제 따라잡기 Section 113

출제예상

1. 데이터베이스 관련 기술 중 다음 설명에 해당하는 것은?

- OOP(Object-Oriented Programming)의 객체와 RDB(Relational Database)를 연결하는 기술이다.
- SQL 코드를 입력하지 않고 프로그래밍 코드와 데이터베이스를 연결하는 것이 가능하다.
- 중복되는 코드를 생략할 수 있어 코드를 간결하게 작성할 수 있다.

① DBMS ② ORM
③ DBA ④ Hadoop

객체(Object)와 관계형 데이터베이스(Relational Database)를 연결(Mapping)하는 기술은 무엇일까요?

출제예상

2. 다음 중 ORM의 기능으로 거리가 먼 것은?

① SQL 코드를 직접 입력하는 것에 비해 간단하게 객체와 데이터베이스를 연결할 수 있다.
② ORM으로 생성된 가상의 객체들은 데이터베이스와 독립적이다.
③ SQL 코드를 입력할 필요가 없어 별도로 SQL을 학습할 필요가 없다.
④ 객체들의 재사용과 유지보수가 용이하다.

ORM이 자동으로 생성해주는 SQL이 의도대로 수행되지 않는다면 직접 확인해야 합니다. SQL을 모르면 불가능하겠죠.

출제예상

3. 객체지향의 객체 모델을 데이터베이스와 연결시키기 위해 여러 가지 기능들을 제공하는 소프트웨어를 무엇이라 하는가?

① DBMS ② Transaction
③ ORM Framework ④ API

소프트웨어에서는 무언가를 구현하기 위해 필요한 구조 및 기능들을 제공하는 소프트웨어를 프레임워크라고 합니다.

출제예상

4. 다음 중 ORM(Object-Oriented Programming)을 구현하기 위한 프레임워크(Framework)에 해당하지 않는 것은?

① Eclipse ② Hibernate
③ Django ④ ODB

Eclipse는 통합 개발 환경(IDE, Integrated Development Environment)을 제공하는 소프트웨어로, ORM인 EclipseLink와는 구분해서 기억해야 합니다.

출제예상

5. 다음 중 ORM(Object-Oriented Programming)의 한계에 대한 설명으로 거리가 먼 것은?

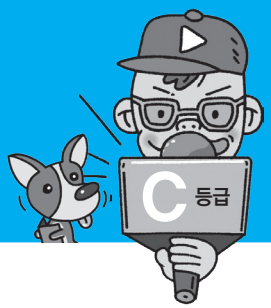
① SQL을 전혀 모르는 상태에서는 ORM을 적용하기 어렵다.
② 큰 프로젝트에서는 데이터베이스의 설계 단계부터 ORM을 고려하지 않으면 ORM을 적용하기가 어렵다.
③ 기존의 기업들은 ORM을 고려하지 않고 데이터 관리의 효율만을 고려하여 구축한 데이터베이스를 갖고 있기 때문에 새로운 애플리케이션을 개발하는데 ORM을 전격적으로 도입하는데 어려움이 있다.
④ ORM에서 생성된 가상의 데이터베이스는 객체와 데이터베이스에 의존적이어서 재사용과 유지보수가 어렵다.

2번 문제를 풀었다면 바로 맞힐 수 있는 문제네요. 답을 모르겠다면 2번 문제를 다시 풀어보세요.

▶ 정답 : 1. ② 2. ③ 3. ③ 4. ① 5. ④

SECTION 114 ● LIVE

쿼리 성능 최적화



전문가의 조언

쿼리 성능의 최적화가 어떤 과정을 거쳐 이루어지는지 이해하고, 최적화를 위해 무엇을 해야 하는지 확실히 파악하고 넘어가세요.

APM(Application Performance Management/Monitoring)

APM은 애플리케이션의 성능 관리를 위해 접속자, 자원 현황, 트랜잭션 수행 내역, 장애 진단 등 다양한 모니터링 기능을 제공하는 도구입니다. APM은 리소스 방식과 엔드투엔드(End-to-End) 방식이 있으며 종류는 다음과 같습니다.

- **리소스 방식** : Nagios, Zabbix, Cacti 등
- **엔드투엔드 방식** : VisualVM, 제니퍼, 스카우터 등

옵티마이저(Optimizer)

- 옵티마이저는 작성된 SQL이 가장 효율적으로 수행되도록 최적의 경로를 찾아 주는 모듈로, RBO와 CBO 두 종류가 있습니다. 실무에서는 주로 CBO가 사용됩니다.
- CBO 옵티마이저는 입·출력 속도, CPU 사용량, 쿼리의 블록 개수, 쿼리에 사용되는 개체의 속성, 튜플의 개수 등을 종합하여 각 DBMS마다 고유의 알고리즘에 따라 산출되는 '비용'을 계산합니다. 그러므로 개체나 DBMS의 버전이 변경되어 알고리즘에 변화가 생기면 실행 계획을 다시 확인해야 합니다.

전문가의 조언

옵티마이저는 사용자가 쿼리를 실행하면 SQL문을 분석(Parsing)하여 SQL에 대한 실행 계획(Execution Plan)을 작성한 후, 실행 계획에 따라 데이터를 조작합니다.

1 쿼리 성능 최적화의 개요

쿼리 성능 최적화는 데이터 입·출력 애플리케이션의 성능 향상을 위해 SQL 코드를 최적화하는 것이다.

- 쿼리 성능을 최적화하기 전에 성능 측정 도구인 APM*을 사용하여 최적화 할 쿼리를 선정해야 한다.
- 최적화 할 쿼리에 대해 옵티마이저*가 수립한 실행 계획을 검토하고 SQL 코드와 인덱스를 재구성한다.

잠깐만요

RBO vs CBO

RBO(Rule Based Optimizer)는 규칙 기반 옵티마이저이고, CBO(Cost Based Optimizer)는 비용 기반 옵티마이저로서 다음과 같은 차이점이 있습니다.

	RBO	CBO
최적화 기준	규칙에 정의된 우선순위	액세스 비용
성능 기준	개발자의 SQL 숙련도	옵티마이저의 예측 성능
특징	실행 계획 예측이 쉬움	성능 통계치 정보 활용, 예측이 복잡함
고려사항	개발자의 규칙 이해도, 규칙의 효율성	비용 산출 공식의 정확성

2 실행 계획(Execution Plan)

실행 계획은 DBMS의 옵티마이저가 수립한 SQL 코드의 실행 절차와 방법을 의미한다.

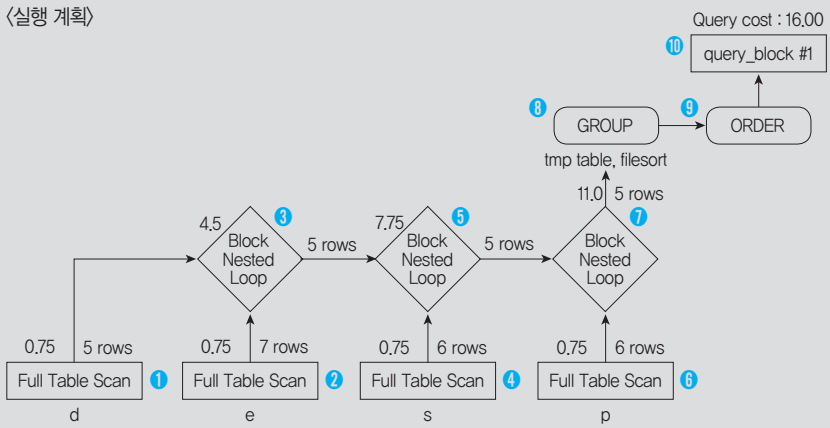
- 실행 계획은 EXPLAIN 명령어를 통해 확인할 수 있으며, 그래픽이나 텍스트로 표현된다.
- 실행 계획에는 요구사항들을 처리하기 위한 연산 순서가 적혀있으며, 연산에는 조인, 테이블 검색, 필터, 정렬 등이 있다.

예제 1 다음은 MySQL에서 4개의 테이블을 조인하는 〈SQL〉의 〈실행 계획〉을 그래픽으로 나타낸 것이다.

〈SQL〉

```
    SELECT
⑩       e.id, e.name, d.dept, SUM(s.num*p.price) as sale
    FROM
①       department AS d,
②       employee AS e,
④       sales AS s,
⑥       product AS p
    WHERE
③       e.dept_id = d.id AND
⑤       s.emp_chg = e.id AND
⑦       p.id = s.goods
⑧   GROUP BY e.id
⑨   ORDER BY e.id;
```

〈실행 계획〉



해설

① d 테이블에서 5개의 튜플을 대상으로 Full Table Scan*을 수행하며, 이 작업에는 0.75의 cost가 발생한다.
② e 테이블에서 7개의 튜플을 대상으로 Full Table Scan을 수행하며, 이 작업에는 0.75의 cost가 발생한다.
③ Block Nested Loop(BNL)* 방식으로 d, e 테이블을 조인하며, 이 작업까지 총 4.5의 cost가 발생하고 5개의 튜플이 반환된다.
④ s 테이블에서 6개의 튜플을 대상으로 Full Table Scan을 수행하며, 이 작업에는 0.75의 cost가 발생한다.
⑤ Block Nested Loop(BNL) 방식으로 ③에서 반환된 5개의 튜플과 s 테이블을 조인하며, 이 작업까지 총 7.75의 cost가 발생하고 5개의 튜플들이 반환된다.
⑥ p 테이블에서 6개의 튜플을 대상으로 Full Table Scan을 수행하며, 이 작업에는 0.75의 cost가 발생한다.
⑦ Block Nested Loop(BNL) 방식으로 ⑤에서 반환된 5개의 튜플과 p 테이블을 조인하며, 이 작업까지 총 11.0의 cost가 발생하였고 그 결과로 5개의 튜플들이 반환된다.
⑧ ⑦에서 반환된 5개의 튜플을 임시 테이블(tmp table)로 만들어 'filesort'* 방식을 이용해 그룹(GROUP)으로 분류(Sort)한다.
⑨ 정렬 작업을 수행한다.
⑩ 'query_block'은 옵티마이저가 한 번에 수행하는 블록 단위이며, '#1'에서 실행 계획이 종료된 것이다. 〈SQL〉 코드가 모두 수행되는데 총 16.00의 cost가 발생한다.

전문가의 조언

'AS 별칭'은 속성뿐만 아니라 테이블에도 사용할 수 있습니다. 'department AS d'는 'department'를 'd'로 사용한다는 의미입니다.

전문가의 조언

왼쪽의 그림은 MySQL Workbench 8.0에서 〈SQL〉을 실행한 후 나타나는 결과 탭의 'Execution Plan'을 클릭하면 확인할 수 있습니다. 단, 실행 계획은 DBMS의 버전이나 실행 환경에 따라 달라질 수 있다는 점을 유념해 두세요.

Full Table Scan
Full Table Scan은 전체 테이블을 대상으로 첫 번째 튜플부터 차례대로 데이터를 조회하는 방식을 의미합니다.

NL, BNL
A 테이블과 B 테이블을 조인했을 때, A 테이블의 튜플만큼 B 테이블의 튜플을 반복하여 조회해 나가는 방식을 NL(Nested Loop)이라고 하고, 두 테이블 모두 인덱스가 없는 경우 조회 성능 개선을 위해 한 테이블의 일정 양의 튜플을 버퍼 메모리에 저장한 후 NL방식으로 조회하는 것을 BNL(Block Nested Loop)이라고 합니다.

filesort
filesort는 분류(sort) 작업을 수행하려는 테이블에 인덱스가 없는 경우 메모리 공간 절약을 위해 디스크에 파일 형식으로 테이블을 저장한 후 분류하는 방식을 의미합니다.

3 쿼리 성능 최적화

쿼리 성능 최적화는 실행 계획에 표시된 연산 순서, 조인 방식, 테이블 조회 방법 등을 참고하여 SQL문이 더 빠르고 효율적으로 작동하도록 SQL 코드와 인덱스를 재구성하는 것을 의미한다.

SQL 코드 재구성

- WHERE 절을 추가하여 일부 레코드만 조회하게 함으로써 조회에 들어가는 비용을 줄인다.
- WHERE 절에 연산자가 포함되면 INDEX를 활용하지 못하므로 가능한 한 연산자 사용을 자제한다.
- 서브 쿼리에 특정 데이터가 존재하는지 확인할 때는 IN보다 EXISTS*를 활용한다.
- 옵티마이저의 실행 계획이 잘못되었다고 판단되는 경우 힌트*를 활용하여 실행 계획의 액세스 경로 및 조인 순서를 변경한다.

인덱스 재구성

- SQL 코드에서 조회되는 속성과 조건들을 고려하여 인덱스를 구성한다.
- 실행 계획을 참고하여 인덱스를 추가하거나 기존 인덱스의 열 순서를 변경한다.
- 인덱스의 추가 및 변경은 해당 테이블을 참조하는 다른 SQL문에도 영향을 줄 수 있으므로 신중히 결정한다.
- 단일 인덱스로 쓰기나 수정 없이 읽기로만 사용되는 테이블의 경우 IOT(Index-Organized Table)*로 구성하는 것을 고려한다.
- 불필요한 인덱스를 제거한다.

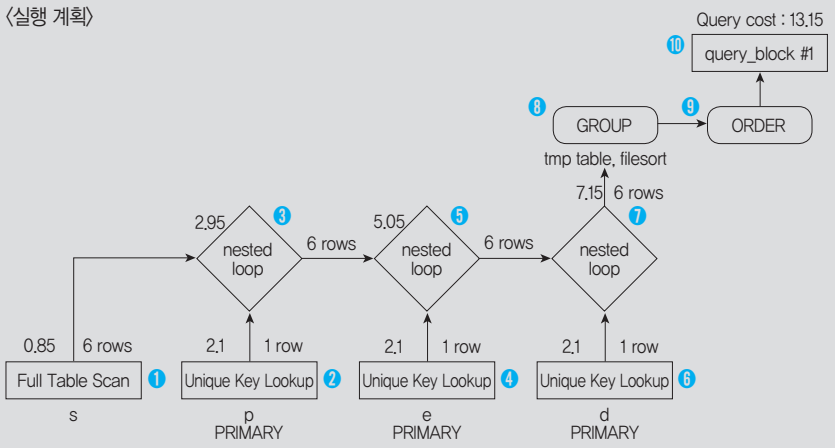
EXISTS
EXISTS는 서브 쿼리의 모든 데이터를 확인하는 IN과 달리 데이터의 존재여부가 확인되면 검색이 종료되므로 IN보다 처리 속도가 빠릅니다.

힌트(Hint)
힌트는 SQL문에 추가되어 테이블 접근 순서를 변경하거나, 인덱스 사용을 강제하는 등의 실행 계획에 영향을 줄 수 있는 문장을 말합니다.

IOT(Index-Organized Table)
일반적으로 인덱스가 있는 테이블을 조회할 때, 인덱스를 검색하여 주소를 얻으면 주소를 다시 찾아가는 과정을 거칩니다. 반면 IOT는 인덱스 안에 테이블 데이터를 직접 삽입하여 저장함으로써 주소를 얻는 과정이 생략되어 더욱 빠른 조회가 가능합니다.

예제 2 다음은 예제 1의 〈SQL〉의 각 테이블에서 조인에 활용되는 속성들을 인덱스로 설정하고, 힌트를 통해 조회 순서를 변경하는 최적화를 수행한 후의 〈실행 계획〉이다.

〈실행 계획〉



해설

① 조인 작업을 수행할 때 시작 지점의 테이블은 무조건 Full Table Scan을 수행한다. s 테이블은 6개의 튜플을 대상으로 Full Table Scan을 수행하며, 이 작업에는 0.85의 cost가 발생한다.

❷ p 테이블은 유니크 키(Unique Key)를 활용한 검색이 이루어지며, 검색에 활용되는 키는 기본키(Primary Key)이다. 인덱스를 통한 검색은 1개의 튜플을 조회하는 것으로 표시된다. 이 작업에는 2.1의 cost가 발생한다.
❸ Nested Loop(NL) 방식으로 s, p 테이블을 조인하며, 이 작업까지 2.95의 cost가 발생하고 6개의 튜플이 반환된다.
❹ e 테이블은 유니크 키(Unique Key)를 활용한 검색이 이루어지며, 검색에 활용되는 키는 기본키(Primary Key)이다. 이 작업에는 2.1의 cost가 발생한다.
❺ Nested Loop(NL) 방식으로 ❸에서 반환된 6개의 튜플과 e 테이블을 조인하며, 이 작업까지 5.05의 cost가 발생하고 6개의 튜플이 반환된다.
❻ d 테이블은 유니크 키(Unique Key)를 활용한 검색이 이루어지며, 검색에 활용되는 키는 기본키(Primary Key)이다. 이 작업에는 2.1의 cost가 발생한다.
❼ Nested Loop(NL) 방식으로 ❺에서 반환된 6개의 튜플과 d 테이블을 조인하며, 이 작업까지 총 7.15의 cost가 발생하고 6개의 튜플이 반환된다.
❽ ❼에서 반환된 6개의 튜플을 임시 테이블(tmp table)로 만들어 'filesort' 방식을 이용해 그룹(GROUP)으로 분류(Sort)한다.
❾ 정렬 작업을 수행한다.
❿ 하나의 블록(query_block #1)으로 처리되었으며, 종료된 〈SQL〉 코드는 총 13.15의 cost가 발생한다.

따라잡기

기출문제 따라잡기 Section 114

출제예상

1. DBMS에서 작성한 쿼리를 수행하는데 시간이 너무 오래 걸려 최적화 작업을 수행하고자 한다. 이와 관련된 내용으로 옳지 않은 것은?

① 먼저 APM 도구를 사용하여 어떤 쿼리를 최적화할지 특정한다.
② 선정된 쿼리에서 옵티마이저가 수립한 실행 계획을 검토한다.
③ DBMS가 규칙 기반 옵티마이저(RBO)를 사용한다면 사전에 정의해 놓은 규칙을 수정하여 실행 계획을 변경한다.
④ 비용 기반 옵티마이저(CBO)의 경우 개발자의 숙련도에 따라 성능이 좌우되므로 쿼리를 충분히 이해하고 최적화 작업을 수행한다.

비용 기반 옵티마이저(CBO)는 옵티마이저의 예측 알고리즘이 얼마나 뛰어나냐에 따라 성능이 좌우됩니다.

출제예상

2. 규칙 기반 옵티마이저(RBO)와 비용 기반 옵티마이저(CBO)에 대한 설명 중 잘못된 것은?

① 규칙 기반 옵티마이저는 개발자가 사전에 정의해 놓은 규칙에 의해 실행 계획이 작성된다.
② 비용 기반 옵티마이저는 정해진 알고리즘에 따라 비용을 계산하여 실행 계획을 작성된다.
③ 규칙 기반 옵티마이저는 DBMS의 버전이나 테이블이 변경되면 접근 경로 등이 변경될 수 있으므로, 변화가 있을 때마다 실행 계획을 재검토해야 한다.
④ 실무에서 주로 사용되는 것은 비용 기반 옵티마이저이다.

DBMS의 버전이 바뀐다고 사람이 정해놓은 규칙이 변경될까요?

출제예상

3. DBMS의 실행 계획에 관한 설명 중 옳지 않은 것은?

① DBMS의 옵티마이저가 수립한 쿼리의 실행 절차와 방법을 의미한다.
② 실행 계획은 DESCRIBE 명령어를 통해 문서로 작성하고 확인할 수 있다.
③ 실행 계획에는 SQL에서 요구하는 사항들을 처리하기 위한 연산 순서가 적혀 있다.
④ 실행 계획에 표시되는 연산에는 조인 기법, 액세스 기법, 필터, 정렬 등이 있다.

실행 계획은 EXPLAIN 명령어를 통해 확인할 수 있습니다.

출제예상

4. 쿼리 성능을 위한 최적화 방법으로 옳지 않은 것은?

① 실행 계획을 바탕으로 SQL 코드 또는 인덱스를 재구성한다.
② WHERE 절을 활용하여 검색 범위를 좁힌다.
③ 단일 인덱스로 조회만 이루어지는 테이블은 IOT로 구성한다.
④ 데이터의 유무 판단을 위해 EXISTS보다 IN을 사용한다.

데이터가 존재하는지(EXISTS)를 확인할 때 사용하는 명령어로 적합한 것은 무엇일까요?

▶ 정답 : 1. ④ 2. ③ 3. ② 4. ④

예·상·문·제·은·행

1. 절차형 SQL의 특징에 대한 설명으로 옳지 않은 것은?

① 프로시저, 사용자 정의 함수, 트리거가 여기에 속한다.
② 프로그래밍 언어와 같은 절차적이고 연속적인 명령문의 실행이 가능하다.
③ 모듈화가 어려워 재사용이 어렵다.
④ DBMS 내부에 저장되어 처리된다.

2. 다음 중 데이터베이스에 저장되어 여러 프로그램에서 호출하여 사용할 수 있으며, 일일 마감 작업 및 배치 작업에 주로 사용되는 절차형 SQL은 무엇인가?

① 프로시저(Procedure)
② 트리거(Trigger)
③ 사용자 정의 함수(User-defined Fuction)
④ 인덱스(Index)

3. 저장 프로시저(Stored Procedure)를 작성하고자 할 때 반드시 작성해야 하는 부분은?

① 트랜잭션(TRANSACTION)
② 선언(DECLARE)
③ 예외(EXCEPTION)
④ 반환(RETURN)

4. 절차형 SQL 중 프로시저의 파라미터(Parameter)에 대한 설명으로 옳지 않은 것은?

① 매개 변수를 의미하며, 프로시저와 호출문 사이에서 값을 전달하는 역할을 수행한다.
② IN은 호출문으로부터 프로시저의 매개 변수에 값을 전달하겠다는 의미이다.
③ OUT은 프로시저로부터 호출문에 매개 변수에 값을 전달하겠다는 의미이다.
④ 파라미터에는 어떤 값이 올지 알 수 없으므로 자료형을 생략한다.

5. 다음 〈실무사례〉에서 사용하기에 가장 적합한 절차형 SQL은?

〈실무사례〉

> 길벗학교에서는 학적을 관리하는 〈학생부〉 테이블을 만들어서 관리하고 있지만, 많은 사용자들이 데이터베이스를 사용하고 있어 잘못된 수정이 발생하는 경우 책임관계가 명확하지 않은 문제가 발생하였다. 이를 해결하기 위해 데이터베이스에 자료가 변경될 때마다 로그 메시지가 관리자에게 출력되도록 만들고자 한다.

① Procedure
② Trigger
③ User-defined Function
④ Message Event

6. 트리거의 이벤트(Event)에 대한 설명으로 옳지 않은 것은?

① 트리거에서는 테이블에 데이터의 조작 행위가 발생한 것이 이벤트에 해당한다.
② 트리거를 생성할 때는 동작시기 옵션, 동작 옵션 설정을 통해 이벤트를 정의한다.
③ 동작 옵션에는 삽입, 삭제, 수정이 있다.
④ 동작시기 옵션에는 테이블이 변경되기 전에 수행하는 First와 나중에 수행하는 Late가 있다.

7. 절차형 SQL인 프로시저, 트리거, 사용자 정의 함수의 각 구성요소 중에서 공통적인 요소에 해당하지 않는 것은?

① DECLARE
② EVENT
③ SQL
④ CONTROL

8. 〈회원〉 테이블에 데이터가 변경된 후에 수행되는 트리거 'ud_tri'를 생성하고자 할 때, 선언부에 가장 적합한 문장은?

① CREATE TRIGGER ud_tri BEFORE INSERT ON 〈회원〉
② CREATE TRIGGER ud_tri AFTER INSERT ON 〈회원〉
③ CREATE TRIGGER ud_tri BEFORE UPDATE ON 〈회원〉
④ CREATE TRIGGER ud_tri AFTER UPDATE ON 〈회원〉

9. 사용자 정의 함수에 대한 설명으로 옳지 않은 것은?

① 프로시저와 동일하게 데이터베이스에 저장되어 실행되는 절차형 SQL이다.
② RETURN 명령어를 통해 값을 반환하는 것이 특징이다.
③ 내장 함수처럼 DML문에서 반환값을 활용하기 위한 용도로 사용된다.
④ 프로시저를 호출하여 사용할 수 있다.

해설은 505쪽에 있습니다.

10. 사용자 정의 함수를 실행하는 방법으로 옳지 않은 것은?

① 사용자 정의 함수는 DML에 포함되어 속성명 또는 값이 놓일 자리에 위치하여 실행된다.

② 사용자 정의 함수는 SELECT문의 조회할 속성 또는 WHERE 조건문에 위치할 수 있다.

③ 사용자 정의 함수는 UPDATE문의 속성명과 속성이 변경될 값의 자리에 위치할 수 있다.

④ 사용자 정의 함수는 DELETE문의 WHERE 조건문에만 위치할 수 있다.

11. 다음 중 응용 시스템에 대한 설명으로 가장 옳지 않은 것은?

① 사용자가 DBMS에 접근할 수 있도록 매개체 역할을 수행한다.

② 웹 응용 시스템의 경우 웹 서버와 WAS로 구분된다.

③ 정적 데이터를 처리하는 웹 서버와 동적 데이터를 처리하는 WAS는 반드시 분리하여 운용해야 한다.

④ 사용자가 단말기를 통해 웹 서버에 데이터를 요청하면 웹 서버는 WAS에 요청을 넘기고 WAS가 트랜잭션을 생성하여 DBMS로부터 데이터를 받아오는 구조이다.

12. DBMS에 접근하기 위해 사용되는 프레임워크인 ODBC에 대한 설명으로 잘못된 것은?

① 개발 언어에 관계없이 사용할 수 있는 표준 개방형 API이다.

② SQL 문장을 분리하여 XML 파일을 만들어 데이터베이스 연결 작업을 수행한다.

③ 마이크로소프트 사에서 개발하였으며, DB2, Access, DBase, Excel, Text 등의 데이터베이스에 접근이 가능하다.

④ 접속하려는 DBMS의 인터페이스를 알지 못해도 ODBC 문장을 통해 사용이 가능하다.

13. 다음 중 MyBatis의 특징에 대한 설명으로 가장 옳지 않은 것은?

① 객체지향의 객체와 데이터베이스를 매핑해주는 ORM에 해당한다.

② JDBC의 코드를 단순화하여 사용할 수 있다.

③ XML 파일을 생성하여 매핑을 통해 SQL을 수행한다.

④ SQL문을 그대로 사용할 수 있어 SQL 친화적인 국내 환경에 적합하다.

14. 정적 SQL과 동적 SQL의 특징에 대한 설명으로 옳지 않은 것은?

① SQL 구성 시 커서(Crusor)를 통해 처리하는 것은 정적 SQL이다.

② 동적 SQL은 String 변수에 SQL문을 담아 처리한다.

③ 정적 SQL은 동적 SQL에 비해 느려 잘 사용되지 않는다.

④ 동적 SQL은 SQL 코드가 상황에 따라 수시로 변화하므로 프리컴파일(Precompile)이 불가능하다.

15. DBMS 접속 기술인 ODBC와 JDBC에 대해 올바르게 설명한 것은?

① ODBC와 JDBC 모두 DBMS에 접속하기 위한 표준 API로, 다양한 언어로 사용이 가능하다.

② ODBC는 JDBC와 다르게 접속하려는 DBMS에 맞는 드라이버가 필요 없다.

③ ODBC는 마이크로소프트에서 개발하였으며, C 언어 또는 .NET을 기반으로 작성된다.

④ JDBC는 MyBatis라는 프레임워크를 통해 더 단순화한 코드로 효율적인 이용이 가능하다.

16. SQL 테스트에 대한 설명으로 잘못된 것은?

① SQL 테스트 중 발생하는 구문 오류(Syntax Error)는 문법이 잘못된 경우 발생한다.

② 참조 오류(Reference Error)는 SQL문이 잘못된 개체를 참조할 경우 발생한다.

③ 절차형 SQL은 디버깅을 통해 검증하는 작업을 수행한다.

④ 단문 SQL은 주석을 통해 결과를 검증한다.

17. 절차형 SQL의 테스트 과정에 대한 설명으로 옳지 않은 것은?

① 먼저 절차형 SQL의 생성을 통해 구문 및 참조 오류가 있는지 확인한다.

② 오류 및 경고 메시지의 상세 확인을 위해 DESC 명령어를 사용한다.

③ 데이터베이스에 변화를 줄 수 있는 SQL문을 주석 처리하고, 출력문을 이용하여 SQL문이 의도에 맞게 수행되는지 확인한다.

④ DBMS에 따라 변수 또는 값의 화면 출력을 위한 과정이 다르므로, 충분히 숙지한 후 출력문을 이용한다.

▶ 정답 : 1. ③ 2. ① 3. ② 4. ④ 5. ② 6. ④ 7. ② 8. ④ 9. ④ 10. ③ 11. ③ 12. ② 13. ① 14. ③ 15. ④ 16. ④ 17. ②

해설은 506쪽에 있습니다.

18. MySQL에서 호스트 주소가 'www.gilbutaws.co.kr'이고, ID가 'gilbut'인 사용자의 권한을 확인하는 SQL문은?

① SHOW GRANTS FOR 'gilbut'@'www.gilbutaws.co.kr';

② SHOW GRANTS FOR 'www.gilbutaws.co.kr'@'gilbut';

③ SHOW AUTHORITY FOR 'gilbut'@'www.gilbutaws.co.kr';

④ SHOW AUTHORITY FOR 'www.gilbutaws.co.kr'@'gilbut';

19. ORM(Object-Relational Mapping)의 특징으로 옳지 않은 것은?

① 객체와 관계형 데이터베이스를 연결하는 기술을 의미한다.

② 가상의 객체 데이터베이스를 활용한다.

③ 코드 및 데이터베이스와 독립적이므로 재사용 및 유지보수가 용이하다.

④ 프로그래밍 코드에서 SQL문을 분리하여 XML에 저장함으로써 SQL문 관리가 수월해진다.

20. 다음 중 개발 환경과 해당 환경이 지원하는 ORM 프레임워크의 연결이 잘못된 것은?

① JAVA – Hibernate

② C++ – ODB

③ .NET – Core Data

④ Python – Django

21. APM(Application Performance Monitoring)에 대한 설명으로 옳은 것은?

① 애플리케이션의 성능 관리 및 진단을 위해 사용되는 도구이다.

② 데이터베이스에서 작성된 SQL이 효율적으로 수행되도록 최적의 경로를 찾아주는 도구이다.

③ DBMS의 옵티마이저가 수립한 SQL 코드의 실행 절차와 방법을 의미한다.

④ 개발자가 정한 규칙에 따라 성능이 좌우된다.

22. 쿼리 성능 최적화를 위한 작업으로 잘못된 것은?

① 힌트를 활용하여 쿼리 내부 명령문들의 실행 순서를 변경해본다.

② 하나의 테이블에 가능한 한 다양한 인덱스를 생성하여 효율적으로 검색될 수 있도록 한다.

③ 단일 인덱스로 읽기만 하는 경우 IOT(Index-Organized Table)로 구성한다.

④ WHERE절을 사용하여 조회 범위를 제한한다.

▶ 정답 : 18. ① 19. ④ 20. ③ 21. ① 22. ②

예·상·문·제·은·행·해·설

1. Section 108

절차형 SQL은 BEGIN~END 형식으로 작성되는 블록(Block) 구조로 되어 있기 때문에 기능별 모듈화가 가능하다.

3. Section 108

프로시저의 구성 요소

- DECLARE(필수) : 프로시저의 명칭, 변수, 인수, 데이터 타입을 정의하는 선언부이다.
- BEGIN(필수) / END(필수) : 프로시저의 시작과 종료를 의미한다.
- CONTROL : 조건문 또는 반복문이 삽입되어 순차적으로 처리된다.
- SQL : DML, DCL이 삽입되어 데이터 관리를 위한 조회, 추가, 수정, 삭제 작업을 수행한다.
- EXCEPTION : BEGIN ~ END 안의 구문 실행 시 예외가 발생하면 이를 처리하는 방법을 정의한다.
- TRANSACTION : 수행된 데이터 작업들을 DB에 적용할지 취소할지를 결정하는 처리부이다.

4. Section 108

자료형은 컴퓨터가 데이터를 취급하는 방법에 대해 정의하는 것으로, 자료형이 없는 경우 데이터를 메모리로 읽어 들이는 것조차 불가능하다.

5. Section 109

데이터 조작 작업이 발생할 때마다 자동으로 수행되고, 로그 메시지 출력의 목적으로 사용되는 절차형 SQL은 트리거(Trigger)이다.

6. Section 109

동작시기 옵션에는 변경 전에 트리거를 수행하는 BEFORE와 변경 후에 트리거를 수행하는 AFTER가 있다.

7. Section 108, 109, 110

이벤트(EVENT)는 트리거에만 존재하는 구성 요소로, 트리거가 생성된 테이블에 데이터의 삽입, 갱신, 삭제와 같은 데이터 조작 작업이 발생했음을 의미한다.

8. Section 109

테이블이 변경된 후에 트리거를 실행시키는 옵션은 AFTER, 테이블에 변경이 있을 때 실행시키는 옵션은 UPDATE이다.

9. Section 110

사용자 정의 함수는 다른 사용자 정의 함수는 호출할 수 있지만 프로시저는 호출하지 못한다.

10. Section 110

사용자 정의 함수는 UPDATE문의 속성이 변경될 값의 자리에는 위치할 수 있지만 속성명 자리에는 위치할 수 없다.

11. Section 111

웹 응용 시스템은 웹 서버와 웹 어플리케이션 서버(WAS)로 구성되며, 서비스 규모가 작은 경우 웹 서버와 웹 어플리케이션 서버를 통합하여 하나의 서버만으로 운용할 수 있다.

12. Section 111

②번의 내용은 MyBatis에 대한 설명이다.

13. Section 111

MyBatis는 JDBC 코드를 단순화하여 사용할 수 있는 SQL Mapping 기반 오픈 소스 접속 프레임워크이다.

14. Section 111

정적 SQL(Static SQL)은 동적 SQL(Dynamic SQL)에 비해 유연한 대응은 불가능하지만 처리 속도가 빠른 장점이 있다.

15. Section 111

- ①, ③ ODBC는 다양한 언어를 기반으로 하지만 JDBC는 JAVA 언어를 기반으로 한다.
- ② ODBC와 JDBC 모두 드라이버 없이는 정상적인 사용이 불가능하다.

16. Section 112

단문 SQL은 DDL, DML, DCL이 포함되어 있는 SQL과 TCL을 통해 결과를 검증하며, 주석을 통해 결과를 검증하는 것은 절차형 SQL이다.

17. Section 112

오류 및 경고 메시지 확인을 위해 사용하는 명령어는 SHOW이며, DESC는 데이터베이스 개체의 정보를 확인할 때 사용하는 명령어이다.

18. Section 112

MySQL에서 사용자의 권한 정보를 확인하는 SQL문의 형식은 'SHOW GRANTS FOR 사용자@호스트;' 이다.

19. Section 113

④번의 내용은 MyBatis에 대한 설명이다.

20. Section 113

Core Data는 iOS에서 ORM을 구현하는 프레임워크이다.

ORM 프레임워크의 종류

- JAVA : JPA, Hibernate, EclipseLink, DataNucleus, Ebean 등
- C++ : ODB, QxOrm 등
- Python : Django, SQLAlchemy, Storm 등
- iOS : DatabaseObjects, Core Data 등
- .NET : NHibernate, DatabaseObjects, Dapper 등
- PHP : Doctrine, Propel, RedBean 등

21. Section 114

②번은 옵티마이저(Optimizer), ③번은 DBMS의 실행 계획(Execution Plan), ④번은 규칙 기반 옵티마이저인 RBO(Rule Based Optimizer)에 대한 설명이다.

22. Section 114

인덱스의 추가 및 변경은 해당 테이블을 참조하는 다른 SQL문에도 영향을 줄 수 있으므로 신중히 결정해야 한다.

5장

데이터 전환

115 데이터 전환 B등급

116 데이터 전환 계획서 작성 C등급

117 데이터 전환 방안 C등급

118 데이터 검증 C등급

119 오류 데이터 측정 및 정제 C등급

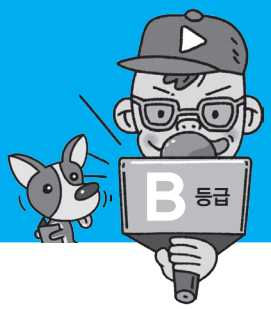
120 데이터 정제요청서 및 정제보고서 C등급

이 장에서 꼭 알아야 할 키워드 Best 10

1. 데이터 전환 **2.** 데이터 전환 계획서 **3.** 데이터 전환 환경 구성 **4.** 데이터 전환 방안 **5.** 데이터 검증
6. 로그 검증 **7.** 오류 데이터 측정 **8.** 오류 데이터 정제 **9.** 데이터 정제 요청서 **10.** 데이터 정제 보고서

SECTION 115 ● LIVE

데이터 전환



전문가의 조언

현재 운영 중인 시스템의 데이터를 추출하고 이를 변환한 후 새로 개발할 시스템에서 사용할 수 있도록 적재하는 과정을 데이터 전환이라고 합니다. 데이터 전환을 성공적으로 수행하려면 데이터 전환 계획서를 작성하여 이를 토대로 진행하면 됩니다. 이번 섹션에서는 데이터 전환의 정의와 데이터 전환 계획서에 작성되는 항목에 대해 알아보겠습니다. 이후 섹션에서는 데이터 전환 계획서의 주요 항목에 작성되는 내용들을 좀 더 자세하게 학습하도록 하겠습니다.

새로 개발할 정보 시스템을 목적 시스템 또는 To-Be 시스템이라고도 하며, 운용 중인 기존 시스템을 원천 시스템 또는 As-Is 시스템이라고도 합니다.

데이터 정비 방안
데이터 정비 방안은 데이터 정비 대상 및 방법을 정의하고 데이터 정비 일정 및 조직에 대한 계획을 수립하는 것을 의미합니다.

비상 계획
비상 계획은 데이터 전환 당일에 구성하게 될 종합상황실에 대한 계획을 수립하고 데이터 전환 시 종합상황실과 전환 관리 그룹과의 실시간 정보 공유 방법에 대한 계획을 수립하는 것을 의미합니다.

데이터 복구 대책
데이터 복구 대책은 데이터 전환 과정에서 오류가 발생하거나 데이터 전환 자체가 실패할 경우에 대비한 데이터의 백업 및 복구 방안을 수립하는 것을 의미합니다.

1 데이터 전환의 정의

데이터 전환이란 운영 중인 기존 정보 시스템*에 축적되어 있는 데이터를 추출(Extraction)하여 새로 개발할 정보 시스템*에서 운영 가능하도록 변환(Transformation)한 후, 적재(Loading)하는 일련의 과정을 말한다.

- 데이터 전환을 ETL(Extraction, Transformation, Load), 즉 추출, 변환, 적재 과정이라고 한다.
- 데이터 전환을 데이터 이행(Data Migration) 또는 데이터 이관이라고도 한다.

2 데이터 전환 계획서

데이터 전환 계획서는 데이터 전환이 필요한 대상을 분석하여 데이터 전환 작업에 필요한 모든 계획을 기록하는 문서로 주요 항목은 다음과 같다.

항목	세부 항목
데이터 전환 개요	데이터 전환 목표 주요 성공 요인 전제조건 및 제약조건
데이터 전환 대상 및 범위	
데이터 전환 환경 구성	원천 시스템 구성도 목적 시스템 구성도 전환 단계별 DISK 사용량
데이터 전환 조직 및 역할	데이터 전환 조직도 조직별 역할
데이터 전환 일정	
데이터 전환 방안	데이터 전환 규칙 데이터 전환 절차 데이터 전환 방법 데이터 전환 설계 전환 프로그램 개발 및 테스트 계획 데이터 전환 계획 데이터 검증 방안
데이터 정비 방안*	데이터 정비 대상 및 방법 데이터 정비 일정 및 조직
비상 계획*	종합상황실 및 의사소통 체계
데이터 복구 대책*	

따라잡기

기출문제 따라잡기

Section 115

출제예상

1. 다음이 설명하는 것은 무엇인가?

> 운영 중인 기존 정보 시스템에 축적되어 있는 데이터를 추출(Extraction)하여 새로 개발할 정보 시스템에서 운영 가능하도록 변환(Transformation)한 후, 적재(Loading)하는 일련의 과정을 말한다.

① 데이터 변환
② 데이터 전송
③ 데이터 전환
④ 데이터 암호화

이것을 ETL, 데이터 이행, 데이터 이관이라고도 하죠.

출제예상

2. 다음이 설명하는 것은 무엇인가?

> 데이터 전환이 필요한 대상을 분석하여 데이터 전환 작업에 필요한 모든 계획을 기록하는 문서로 주요 항목은 데이터 전환 개요, 데이터 전환 대상 및 범위, 데이터 전환 환경 구성 등이 있다.

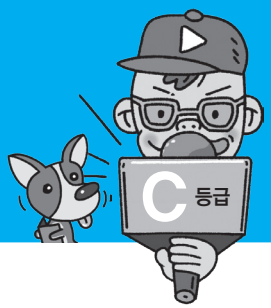
① 데이터 전환 정의서
② 데이터 전환 계획서
③ 데이터 전환 매뉴얼
④ 데이터 전환 요청서

데이터 전환에 필요한 모든 계획을 기록하는 문서는 무엇일까요?

▶ 정답 : 1. ③ 2. ②

SECTION 116 ● LIVE

데이터 전환 계획서 작성



전문가의 조언

이번 섹션에서는 데이터 전환 계획서의 주요 항목 중 데이터 전환 개요, 데이터 전환 대상 및 범위, 데이터 전환 환경 구성, 데이터 전환 조직 및 역할, 데이터 전환 일정을 작성하는 방법에 대해 학습합니다. 각 항목별로 어떤 내용들이 포함되는지를 가볍게 읽으면서 정리하세요.

성공적인 데이터 전환을 위한 주요 요인

성공적인 데이터 전환을 위한 주요 요인에는 '프로젝트 일정 계획에 따른 품질관리 활동을 실시', '데이터 전환 및 검증을 위한 데이터 전환 전문가들을 별도의 팀으로 구성' 등과 같이 데이터 전환을 성공적으로 완수하기 위해 필요한 주요 요인을 기술합니다.

1 데이터 전환 개요

데이터 전환 개요 항목에는 데이터 전환의 목표, 성공적인 데이터 전환을 위한 주요 요인*, 데이터 전환 작업을 위한 전제 조건 및 제약 사항을 기술한다.

- 데이터 전환 목표는 간단하고 명료하게 정의한다.
- 데이터 전환 작업을 위한 전제 조건이나 제약 사항을 정확히 기술하여 데이터 전환 참여자들이 작업을 진행하는 과정에서 혼선이 빚어지지 않도록 한다.

2 데이터 전환 대상 및 범위

데이터 전환 대상 및 범위에는 단위 업무별로 데이터 전환 대상 정보, 해당 업무에 사용되는 Table의 수, 데이터 크기를 기술한다.

예 데이터 전환 대상 및 범위

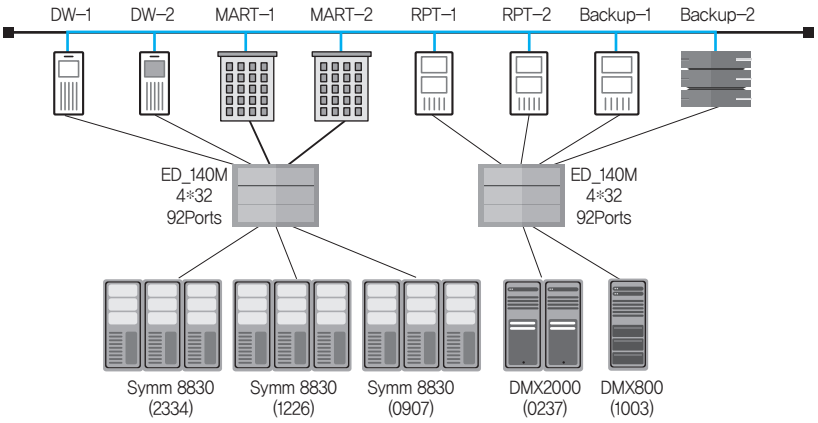
NO	분류 구분	전환 대상	Table 수	크기(GB)
01	제품정보	제품마스터	200	500
02	거래처정보	거래처마스터, 거래내역	100	250
03	조직정보	조직도, 부서정보	5	50
04	인사정보	인사마스터, 학력, 경력, 포상내역, 발령내역 등	150	400

3 데이터 전환 환경 구성

데이터 전환 환경 구성 항목에는 원천 시스템과 목적 시스템의 구성도, 전환 단계별 DISK 사용량을 기술한다.

원천 시스템 구성도

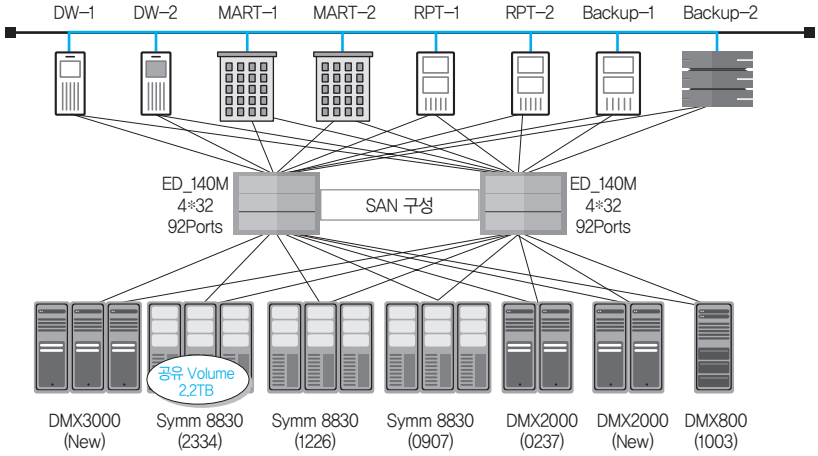
원천 시스템의 서버, 스토리지, 네트워크 등을 포함한 구성도를 작성한다.



원천 시스템 구성도 예

목적 시스템 구성도

목적 시스템의 서버, 스토리지, 네트워크 등을 포함한 구성도를 작성한다.



목적 시스템 구성도 예

전환 단계별 DISK 용량 산정

전환 검증, 시험 단계, 본 전환 단계별로 요구되는 File 공간과 DB 공간을 산정하여 기술한다.

예 전환 단계별 DISK 용량 산정

전환 단계	시스템	File 공간	DB 공간	합계
전환 검증 및 시험	제품정보	12.0 TB	4.0 TB	16.0 TB
	거래처정보	30.0 TB	10.0 TB	40.0 TB
	인사정보	2.4 TB	0.8 TB	3.2 TB
	합계	44.4 TB	14.8 TB	59.2 TB

리허설 및 본 전환	제품정보	10.0 TB	4.0 TB	14.0 TB
	거래처정보	25.0 TB	10.0 TB	35.0 TB
	인사정보	2.0 TB	0.8 TB	2.8 TB
	합계	37.0 TB	14.8 TB	51.8 TB

4 데이터 전환 조직 및 역할 작성

데이터 전환 조직 및 역할에는 데이터 전환을 수행하고 결과를 검증할 작업자와 작업자별 역할을 최대한 상세히 기술한다.

예 데이터 전환 조직도

조직	작업자	주요 역할
사업 관리팀	황종근 부장	• 각 업무운영 팀의 협조 필요 시 중계 지원 • 통합정보센터의 협조 필요 시 중계 지원
각 업무 운영팀	제품관리 박인식 부장 거래처관리 홍종돌 차장 인사관리 김학건 차장	• 현 운영 데이터 확인 필요 시 지원 • 현 시스템 검증 자료 제공
통합정보센터	박정호 부장	• 인프라 전환 팀에서 운영 전환 환경 구축 시 지원 • 데이터 전환 시 작업 환경 지원
프로젝트 총괄	윤태성 PM	• 데이터 전환 총괄 • 타 팀과 업무 협조 사항 지원
데이터 전환팀	신동석 수석	• 데이터 전환 상세 계획 수립 • 데이터 전환 검증 계획 수립 • 데이터 전환 프로그램 개발 • 데이터 전환 및 검증
인프라 전환팀	류세열 수석 최준호 책임	• 데이터 전환 전 서버, 네트워크 환경 구성 및 지원 • 각종 관리도구, DBMS 등 SW 설치 및 지원 • 데이터 전환 환경 장애 발생 시 지원
장비 사업자	영업 대표 송준호 영업 대표 김영진	데이터 전환 작업에 필요한 하드웨어/소프트웨어/네트워크 장비 지원
각 업무 개발팀	제품관리 정범석 부장 거래처관리 서국현 차장 인사관리 송성수 차장	• 데이터 매핑 정의서 설계 – 데이터 매핑 정의서(테이블) – 데이터 매핑 정의서(코드) – 데이터 매핑 정의서(칼럼) • 각 업무별 검증 자료 준비 • 데이터 전환 결과 최종 점검

5 데이터 전환 일정 작성

데이터 전환 일정에는 데이터 전환 및 검증 작업별로 상세하게 일정을 수립하여 작성한다.

- 데이터 전환 관계자들 사이의 보고나 의사소통 등 효과적인 일정 관리를 위해 도식화된 일정도 작성하여 포함한다.

따라잡기

기출문제 따라잡기

Section 116

출제예상

1. 데이터 전환 계획서 작성에 대한 설명 중 가장 옳지 않은 것은?

① 데이터 전환 목표는 구체적이고 자세하게 정의한다.

② 데이터 전환 환경과 관련해서는 원천 시스템과 목적 시스템의 구성도, 전환 단계별 DISK 사용량을 기술한다.

③ 데이터 전환 조직도 작성 시 데이터 전환을 수행하고 결과를 검증할 작업자와 작업자별 역할은 최대한 상세히 정리한다.

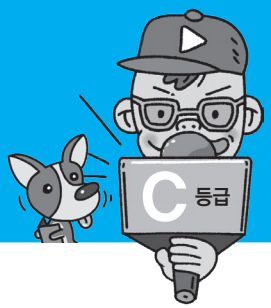
④ 데이터 전환 일정은 데이터 전환 및 검증 작업별로 상세하게 일정을 수립하여 작성한다.

데이터 전환 목표는 간단하고 명료하게 정의합니다.

▶ 정답 : 1. ①

SECTION 117 ● LIVE

데이터 전환 방안



전문가의 조언

이번 섹션에서는 데이터 전환 계획서의 주요 항목 중 데이터 전환 방안을 작성하는 방법에 대해 학습합니다. 데이터 전환 방안에는 어떤 내용들이 포함되는지를 가볍게 읽으면서 정리하세요.

1 데이터 전환 방안

데이터 전환 방안 항목에는 데이터 전환 규칙, 데이터 전환 절차, 데이터 전환 방법, 데이터 전환 설계, 전환 프로그램 개발 및 테스트 계획, 데이터 전환 계획, 데이터 검증 방안이 있다.

2 데이터 전환 규칙

데이터 전환 규칙 항목에는 데이터 전환 과정에서 공통적으로 적용해야 할 규칙들을 기술한다.

예 데이터 전환 규칙

공통 규칙	규칙 설명
본 전환 시 동시 수행 프로세스 수	CPU 수 × 1.5 : 동시 수행될 전환 프로세스 수
DDL 수행 성능 강화	DW* 초기 적재 : 2차 수행 후 Index 생성
DW 초기 적재 시 분산 처리 성능 강화	전환될 추출 데이터를 개발 서버 2대와 운영 서버에 분산하여 전송
선 전환 시 운영 서버 부하 최소화	선 전환 시 운영 서버 부하를 최소화할 수 있는 시간 대에 작업을 수행
운영 서버 여유 공간	여유 공간이 미 확보된 경우에는 개발 서버 NFS(Network File System)* 공간을 활용

3 데이터 전환 절차

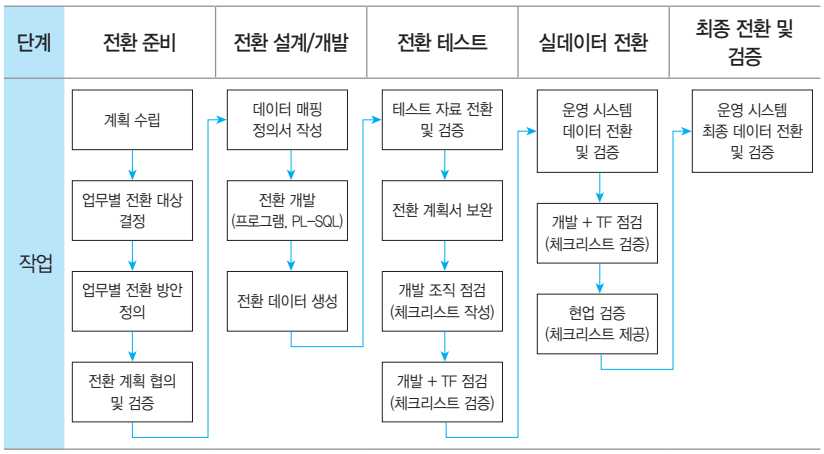
데이터 전환 절차 항목에는 전환 준비, 전환 설계/개발, 전환 테스트, 실데이터 전환, 최종 전환 및 검증의 데이터 전환 절차를 체계적이고 상세하게 기술한다.

- 데이터 전환 절차 수립 시 작업의 이해를 위해 데이터 흐름도를 작성한다.

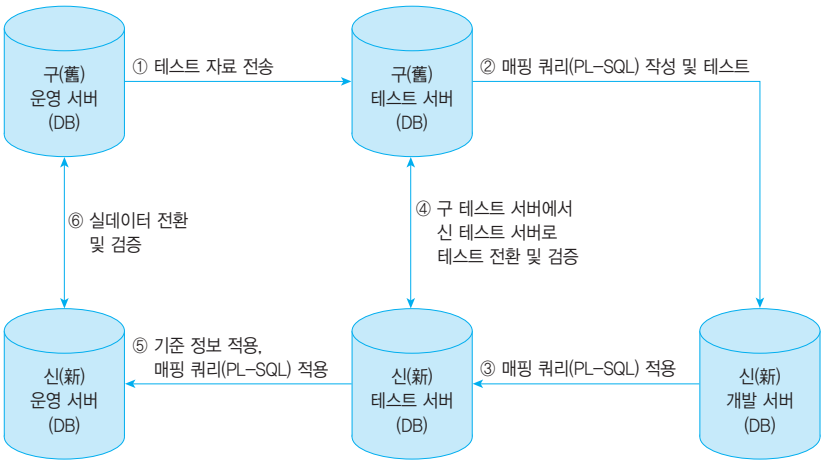
DW(Data Warehouse, 데이터웨어하우스)
DW는 조직이나 기업체의 중심이 되는 주요 업무 시스템에서 추출되어 새로이 생성된 데이터베이스로서 의사결정지원 시스템을 지원하는 주체적, 통합적, 시간적 데이터의 집합체를 말합니다. 쉽게 말하면 정보 시스템을 운영하기 위해 업무 중 발생한 다양한 데이터를 한데 모아놓은 것을 말합니다.

NFS(Network File System)
NFS는 선(SUN) 마이크로시스템스 사에 의해 개발된 것으로, 근거리 통신망 또는 원격 통신망을 통해 파일에 접근하기 위해 구현되고 구체화된 시스템입니다. 독립된 파일 시스템을 가진 서로 다른 워크스테이션의 모임이 서로 연결된 형태입니다.

예 데이터 전환 절차



예 데이터 전환 절차상의 데이터 흐름도



4 데이터 전환 방법

데이터 전환 방법 항목에는 단위 업무별로 데이터 전환 방법을 기술하되, 데이터 전환 시 업무별로 요구되는 전제 조건도 함께 기술한다.

예 데이터 전환 방법

ERP 및 그룹웨어	전환 방안	전제 조건
DB동기화 방식 전환 방법	• Oracle DB와 SQL Server 간 DB 동기 • 목적 시스템에서 필요한 정보에 대한 전환 프로시저를 작성(Oracle DB As_Is 테이블 → SQL Server DB To_Be 테이블 전환)	• DB 간 동기화는 전환 기간 동안 유지할 것 • Oracle DB 테이블에 대한 접근 권한을 확보할 것
DB비동기화 방식 전환 방법	• Oracle DB와 SQL Server 간 DB 비동기 • 목적 시스템에서 필요한 정보에 대한 전환 프로시저를 작성(Oracle DB As_Is 테이블 → SQL Server DB To_Be 테이블 전환)	• DB 간 동기화는 전환 기간 중 필요한 경우에만 유지할 것 • Oracle DB 테이블에 대한 접근 권한을 확보할 것

기타 방법	• 입력 자료의 양이 적은 겨우 ERP 입력 메뉴를 이용해 직접 입력 • As-Is 데이터와 To-Be 데이터 간 전환이 어려운 경우 View 형태 제공 등의 방법을 선정	• ERP 입력 메뉴 제공 • 전환 데이터 검증

5 데이터 전환 설계

데이터 전환 설계 항목에는 업무별로 전환 대상과 전환 제외 대상을 기술하고 원천 시스템 테이블과 목적 시스템 테이블의 매핑 정의서를 작성한다.

예 전환 대상 및 전환 제외 대상

업무명	To-Be Table 수		
	전체	전환 제외	전환 대상
제품관리	322	15	307
거래처관리	151	7	144
인사관리	254	0	254
평가관리	127	0	127
근태관리	85	1	84
급여관리	262	1	261
합계	1,201	24	1,177

예 테이블 매핑 정의서

업무 영역	1:1	1:N	N:1	N:M	To-Be에서 삭제
제품관리	302	4		1	15
거래처관리	132		12		7
인사관리	244	8		2	0
평가관리	120		6	1	0
근태관리	80	3		1	1
급여관리	252	9			1
합계	1,130	24	18	5	24

6 전환 프로그램 개발 및 테스트 계획

전환 프로그램 개발 및 테스트 계획 항목에는 전환 프로그램 개발 계획과 전환 프로그램 테스트 계획을 수립한 후 관련 내용을 기술한다.

- 전환 프로그램은 목록별로 프로그램 입력 정보, 중간 생성 정보, 출력 정보, 프로그램 위치, 담당자 등을 포함하여 작성한다.

- 원천 시스템에서 사용하던 데이터를 목적 시스템으로 전환하기 위한 세부 데이터 항목을 전환 프로그램 목록별로 작성한다.
- 전환 프로그램 테스트는 체크리스트를 기반으로 실시하고 실시 결과를 전환 시나리오에 반영하도록 계획한다.

7 데이터 전환 계획

데이터 전환 계획 항목에는 데이터 전환 시간을 단축하기 위해 선 전환, 본 전환, 후 전환으로 분리하여 계획을 수립한 후 관련 내용을 기술한다.

- 데이터 전환 시간을 단축하기 위해 일자별 거래 내역, 일자별 근태 내역과 같은 대량의 데이터 테이블은 사전에 전환한다.
- 본 전환에 대한 세부 절차는 시간대별로 상세하게 작성한다.
- 본 전환 이후에 전환을 수행해도 되는 대상을 정의한 다음, 후 전환 계획을 수립한다.
- 전환을 위한 준비부터 선 전환, 본 전환, 전환 점검, 시스템 오픈, 후 전환 등 작업별로 전환 시간, 전환 담당자, 관리자 등을 지정한 전환 시나리오를 작성한다.

8 데이터 검증 방안

데이터 검증 방안 항목에는 데이터 전환 이후 전환 데이터의 정합성*을 검증하고 전환 과정에서 발생할 수 있는 문제에 대응할 수 있도록 단계별 데이터 전환 검증 방안을 수립한 후 관련 내용을 기술합니다.

예 데이터 전환 검증 수행 방안

단계	검증 내용	역할
전환 작업	• 데이터 추출 시간 Log 생성 여부 • 데이터 변환 시간 Log 생성 여부 • 데이터 오류 발생 시 Log 생성 여부 • 적재 SAM 파일*이 목적 테이블에 적재될 때 Log 생성 여부	데이터 전환팀
1단계 검증 (전환 검증)	• 원천 시스템 DB vs 목적 시스템 DB 테이블의 검증 • 숫자 칼럼이 있는 테이블의 Group By SUM 비교	데이터 전환팀 메타 관리팀
2단계 검증 (업무 검증)	① 원천 시스템에서 작업 • 각 업무별 조회 화면을 확보 • 보고서 프로그램 실행 후 자료 확보 • 주요 테이블에 대하여 업무적으로 중요한 데이터를 산출한 자료 확보	업무 운영팀
	② 목적 시스템에서 작업 • 각 업무별 조회 화면을 확보 • 보고서 프로그램 실행 후 자료 확보 • 주요 테이블에 대하여 업무적으로 중요한 데이터를 산출한 자료 확보	업무 개발팀
	③ 원천 시스템 vs 목적 시스템 검증 자료 확인	업무 운영팀 업무 개발팀

정합성
정합성은 데이터의 값이 서로 모순 없이 일관되게 일치하는 정도를 말합니다.

SAM(Sequential Access Method) 파일
SAM 파일은 레코드를 논리적인 처리 순서에 따라 연속된 물리적 저장 공간에 기록한 것으로, 판독할 때도 순차적으로 접근하기 때문에 순차 접근 방식의 파일이라고 합니다.

출제예상

1. 데이터 전환 방안 작성에 대한 설명 중 가장 옳지 않은 것은?

① 데이터 전환 절차 수립 시 작업의 이해를 위해 데이터 흐름도를 작성한다.

② 데이터 전환 시간을 단축하기 위해 일자별 거래 내역, 일자별 근태 내역과 같은 대량의 데이터 테이블은 사후에 전환한다.

③ 전환 프로그램은 목록별로 프로그램 입력 정보, 중간 생성 정보, 출력 정보, 프로그램 위치, 담당자 등을 작성한다.

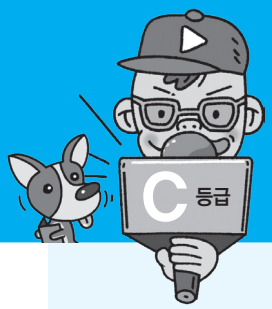
④ 단위 업무별로 데이터 전환 방법을 기술하되, 데이터 전환 시 업무별로 요구되는 전제 조건도 함께 기술한다.

전환에 많은 시간이 소요되는 작업들은 미리 진행해야 전체적인 전환 시간을 단축할 수 있습니다.

▶ 정답 : 1. ②

SECTION 118 LIVE

데이터 검증



1 데이터 검증

데이터 검증이란 원천 시스템의 데이터를 목적 시스템의 데이터로 전환하는 과정이 정상적으로 수행되었는지 여부를 확인하는 과정을 말한다.

- 데이터 전환 검증은 검증 방법과 검증 단계에 따라 분류할 수 있다.

2 검증 방법에 따른 분류

데이터 검증은 검증 방법에 따라 로그 검증, 기본 항목 검증, 응용 프로그램 검증, 응용 데이터 검증, 값 검증으로 분류할 수 있다.

로그 검증	데이터 전환 과정에서 작성하는 추출, 전환, 적재 로그를 검증한다.
기본 항목 검증	로그 검증 외에 별도로 요청된 검증 항목에 대해 검증한다.
응용 프로그램 검증	응용 프로그램을 통한 데이터 전환의 정합성을 검증한다.
응용 데이터 검증	사전에 정의된 업무 규칙을 기준으로 데이터 전환의 정합성을 검증한다.
값 검증	숫자 항목의 합계 검증, 코드 데이터의 범위 검증, 속성 변경에 따른 값 검증을 수행한다.

3 검증 단계에 따른 분류

원천 데이터를 추출하는 시점부터 전환 시점, DB 적재 시점, DB 적재 후 시점, 전환 완료 후 시점별로 그 목적과 검증 방법을 달리하여 데이터 전환의 정합성 검증을 수행할 수 있다.

검증 단계	목적	검증 방법
추출	원천 시스템 데이터에 대한 정합성 확인	로그 검증
전환	• 매핑 정의서에 정의된 내용이 정확히 반영되었는지 확인 • 매핑 정의서 오류 여부 확인	로그 검증
DB 적재	SAM 파일을 적재하는 과정에서 발생할 수 있는 오류나 데이터 누락 여부 등 확인	로그 검증
DB 적재 후	적재 완료 후 정합성 확인	기본 항목 검증
전환 완료 후	데이터 전환 완료 후 추가 검증 과정을 통해 데이터 전환의 정합성 검증	응용 프로그램 검증 응용 데이터 검증

전문가의 조언

이번 섹션에서는 데이터 전환이 정상적으로 수행되었는지 여부를 확인하는 데이터 검증 방법에 대해 학습합니다. 먼저 데이터 검증에 대한 개념을 명확히 잡으세요. 그리고 검증 방법에는 어떤 것들이 있는지 파악해 두세요.

출제예상

1. 다음 중 데이터 전환 검증에 사용되는 검증 방법이 아닌 것은?

① 로그 검증
② 기본 항목 검증
③ 정합성 검증
④ 값 검증

검증 방법에 따른 분류에는 보기에 제시된 것 외에 응용 프로그램 검증과 응용 데이터 검증이 있습니다.

출제예상

2. 다음 지문에 제시된 내용이 설명하는 것은 무엇인가?

원천 시스템의 데이터를 목적 시스템의 데이터로 생성하는 과정이 정상적으로 수행되었는지 여부를 확인하는 과정이다.

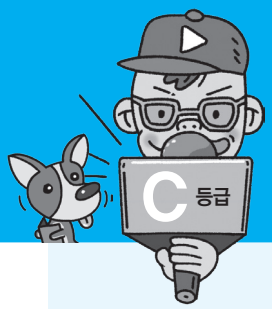
① 데이터 추출
② 데이터 검증
③ 데이터 적재
④ 데이터 변환

보기 중에서 '확인하는 것'과 관련된 것을 찾아보세요.

▶ 정답 : 1. ③ 2. ②

SECTION 119 LIVE

오류 데이터 측정 및 정제



1 오류 데이터 측정 및 정제의 개요

오류 데이터 측정 및 정제는 고품질의 데이터를 운영 및 관리하기 위해 수행한다.

- 오류 데이터 측정 및 정제는 '데이터 품질 분석 → 오류 데이터 측정 → 오류 데이터 정제' 순으로 진행한다.

2 데이터 품질 분석

데이터 품질 분석은 오류 데이터를 찾기 위해 원천 및 목적 시스템 데이터의 정합성* 여부를 확인하는 작업이다.

- 대표적인 원천 시스템 데이터의 정합성 항목
  - 필수 항목에 해당하는 데이터가 모두 있는가?
  - 문자형, 숫자형, 날짜형 등 데이터의 유형은 올바르게 관리되는가?
  - 날짜는 유효한 날짜 형태인가?
  - 금액은 유효한 값의 범위인가?
  - 계좌번호 또는 코드값은 정해진 규칙에 맞는 번호인가?
  - 영업점은 실제 존재하는가?
  - 업무 규칙에 위배되는 잘못된 정보는 없는가?
  - 보고서값과 실제 데이터값이 일치하는가?
  - 잔액은 회계 정보와 일치하는가?
- 대표적인 목적 시스템 데이터의 정합성 항목
  - 과목별 좌수* 및 잔액
  - 특정 기준으로 분류된 좌수 및 잔액
  - 보고서 항목 또는 통계 수치
  - 계좌 및 고객을 샘플링하여 해당사항 확인
  - 특수 관계의 고객에 대한 데이터 확인

3 오류 데이터 측정

오류 데이터 측정은 데이터 품질 분석을 기반으로 정상 데이터와 오류 데이터의 수를 측정하여 오류 관리 목록을 작성하는 것이다.

- 정상 데이터는 전환 대상 범위의 데이터를 업무 영역별, 테이블별로 구분하여 수량을 측정 및 기재한다.
- 오류 데이터는 업무별로 오류 위치 및 유형을 확인하여 수량을 측정 및 기재한다.

전문가의 조언

데이터의 품질을 관리하기 위해서는 원천 및 전환된 데이터의 품질을 분석한 후 오류 데이터를 측정하고 데이터 정제 작업을 수행해야 합니다. 데이터 품질 분석, 오류 데이터 측정, 오류 데이터 정제에 대해 학습하세요.

정합성

정합성은 데이터의 값이 서로 모순 없이 일관되게 일치하는 정도를 말합니다.

좌수

좌수는 펀드의 기준 단위, 즉 펀드를 세는 단위입니다.

4 오류 데이터 정제

오류 데이터 정제는 오류 관리 목록의 각 항목을 분석하여 원천 데이터를 정제하거나 전환 프로그램을 수정하는 것이다.

오류 데이터 분석

- 오류 관리 목록의 오류 데이터를 분석하여 오류 상태, 심각도, 해결 방안을 확인 및 기재한다.
- 상태

Open	오류가 보고만 되고 분석되지 않은 상태
Assigned	오류의 영향 분석 및 수정을 위해 개발자에게 오류를 전달한 상태
Fixed	개발자가 오류를 수정한 상태
Closed	수정된 오류에 대해 테스트를 다시 했을 때 오류가 발견되지 않은 상태
Deferred	오류 수정을 연기한 상태
Classified	보고된 오류를 관련자들이 확인했을 때 오류가 아니라고 확인된 상태

- 심각도

상	데이터 전환을 수행할 수 없는 오류
중	데이터 전환 전반에 영향을 미치는 오류
하	데이터 전환에 직접적인 영향을 미치지는 않으나 상황에 맞지 않는 용도 및 배치 오류

- **해결 방안** : 수정 가능한 오류의 경우 오류의 해결 방안을, 수정 불가능한 오류의 경우 고객과 협의하여 협의 결과를 기재한다.

오류 데이터 정제

확인된 오류 데이터 분석을 통해 원천 데이터를 정제하거나 전환 프로그램을 수정한다.

예 오류 관리 목록

프로젝트	거래처관리 시스템 재구축 프로젝트
시스템	고객관리 시스템

순번	오류ID	정제ID	오류원인	해결방안	심각도	상태	발생일자	예상 해결일자	담당자
1	ER001	2019-123	고객 코드 미존재	코드 정보에 기업체 추가	상	Open	2019-10-25 08:30	2019-11-01 10:00	김선길
2	ER002	2019-124	기본값 없음	NULL 데이터의 초기값 부여	상	Fixed	2019-10-26 15:00	2019-11-02 12:00	임선호

따라잡기

기출문제 따라잡기

Section 119

출제예상

1. 다음 중 오류 데이터 측정 및 정제에 대한 설명으로 옳지 않은 것은?

① 원천 데이터를 분석하여 정합성 여부를 확인하고 오류 데이터의 유형과 건수를 측정한다.

② '오류 데이터 측정 → 데이터 품질 분석 → 오류 데이터 정제' 순으로 진행한다.

③ 오류 데이터 측정 시 정상 데이터와 오류 데이터를 구분하여 측정한다.

④ 오류 데이터 정제는 원천 데이터를 정제하거나 전환 프로그램을 수정하는 것이다.

오류 데이터하면 '분석 → 측정 → 정제'라는 것을 기억해 두세요.

출제예상

2. 다음 중 원천 데이터의 품질 검증을 위한 정합성 항목과 가장 거리가 먼 것은?

① 필수 항목의 데이터가 모두 존재하는가?

② 데이터의 유형이 정확하게 관리되고 있는가?

③ 계좌 번호 등이 정해진 법칙에 따른 번호인가?

④ 과목별 좌수 및 잔액은 정확한가?

과목별 좌수 및 잔액은 목적 데이터에 대한 정합성 검증 항목에 해당합니다.

출제예상

3. 다음 중 오류 데이터 상태에 대한 설명으로 가장 옳지 않은 것은?

① Open은 오류가 보고되었지만 아직 분석되지 않은 상태를 의미한다.

② Assigned는 오류의 영향 분석 및 수정을 위해 개발자에게 오류를 할당한 상태를 의미한다.

③ Fixed는 개발자가 오류를 수정한 상태를 의미한다.

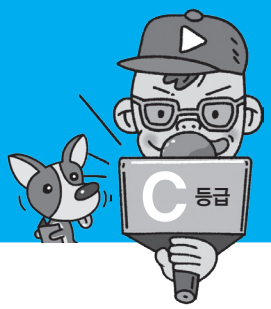
④ Classified는 수정이 불가능한 오류 상태를 의미한다.

Classified는 보고된 오류를 관련자들이 확인했을 때 오류가 아니라고 확인된 상태를 의미합니다.

▶ 정답 : 1. ② 2. ④ 3. ④

SECTION 120 ● LIVE

데이터 정제요청서 및 정제보고서



전문가의 조언

데이터 정제요청서와 정제보고서는 말그대로 데이터 정제가 필요한 데이터들의 정제를 요청하는 문서와 정제된 데이터의 검증 결과를 기록한 문서입니다. 각각의 문서를 작성하는 방법에 대해 학습하세요.

오류 관리 목록에 대한 자세한 내용은 521쪽을 참고하세요.

1 데이터 정제요청서의 개요

데이터 정제요청서는 원천 데이터의 정제와 전환 프로그램의 수정을 위해 요청사항 및 조치사항 등 데이터 정제와 관련된 전반적인 내용을 문서로 작성한 것이다.

- 오류 관리 목록*을 기반으로 데이터 정제 요건 목록을 작성하고 이 목록의 항목별로 데이터 정제요청서를 작성한다.

2 데이터 정제 요건 목록 작성

데이터 정제 요건 목록은 오류 관리 목록의 각 항목에 대해 정제 유형을 분류하고 현재의 정제 상태를 정의한 것이다.

예 데이터 정제 요건 목록

순번	정제ID	정제제목	정제유형	정제방법	상태	정제요청		정제조치		
						일자	이름	검토자	조치자	확인자
1	2019-123	고객 코드 오류	일치성	원천	1차 확인	10/26	김선길	조광희		
2	2019-124	기본값 오류	기타	전환	1차 확인	10/27	임선호	김준용		

- **정제 유형**
  - 완전성 : 업무상 반드시 필요한 자료가 누락된 경우
    예 고객 정보에 주민등록번호 등이 누락된 경우
  - 유효성 : 항목의 값이 유효하지 않은 경우
    예 생년월일이 현재 날짜 이후인 경우
  - 일치성 : 상호 관련이 있는 항목이 서로 다른 경우
    예 성별이 여자이면서 주민등록번호 뒤 7자리 중 첫 번째가 1 또는 3인 경우
  - 유일성 : 서로 상이해야 하는 항목의 값이 동일한 경우
    예 서로 다른 고객의 주민등록번호가 동일한 경우
  - 기타 : 그 외의 정제 유형
- **정제 방법**
  - 원천 : 원천 데이터의 정제가 필요한 경우
  - 전환 : 전환 프로그램의 수정이 필요한 경우
  - 모두 : 원천 데이터의 정제와 전환 프로그램의 수정이 모두 필요한 경우
- **상태** : 요건 제기, 1~3차 검토/조치/확인 등 진행 상태를 기재한다.

3 데이터 정제요청서 작성

데이터 정제요청서에는 데이터 전환 시 발생한 오류의 수정을 위한 정제 요청의 전반적인 내용들을 작성하며, 데이터 정제 검토 시 신속한 의사 결정을 위해 오류사항의 해결 방안도 포함시킨다.

예 데이터 정제요청서

<table>
<tr><td rowspan="4">정제ID</td><td>정제ID</td><td>2019-123</td><td>상태</td><td>1차 확인</td></tr>
<tr><td>정제제목</td><td>고객 코드 오류</td><td>정제유형</td><td>일치성</td></tr>
<tr><td>시스템</td><td>고객관리 시스템</td><td>서브시스템</td><td>고객예약상담 시스템</td></tr>
<tr><td>관련 테이블</td><td>BL_FIELD, BL_SECTION</td><td>예상처리건수</td><td>25,436</td></tr>
<tr><td rowspan="3">데이터 정제 요청</td><td>요건 요청팀</td><td colspan="3">CRM01</td></tr>
<tr><td>요청일자</td><td>2019-10-26</td><td>요청자</td><td>김선길</td></tr>
<tr><td colspan="4">• BL_FIELD의 2010년 이전 정보와 BL_SECTION의 2010년 이후 정보를 TBL_SECTION으로 병합함
• 고객코드는 대문자로 표시하고 공백은 제거함</td></tr>
<tr><td rowspan="6">데이터 정제 검토</td><td>1차 검토부서</td><td>DEC1</td><td>1차 검토자</td><td>조광희</td><td>1차 검토일자</td><td>2019-10-27</td></tr>
<tr><td colspan="5">소문자로 표시된 고객코드 존재함</td></tr>
<tr><td>2차 검토부서</td><td>DEC2</td><td>2차 검토자</td><td>김한순</td><td>2차 검토일자</td><td>2019-10-27</td></tr>
<tr><td colspan="5">공백을 제거한 후 동일한 고객으로 인식함</td></tr>
<tr><td>3차 검토부서</td><td>DEC3</td><td>3차 검토자</td><td>강준용</td><td>3차 검토일자</td><td>2019-10-28</td></tr>
<tr><td colspan="5">특이사항 없음</td></tr>
</table>

4 데이터 정제보고서의 개요 및 작성

데이터 정제보고서는 데이터 정제요청서를 통해 정제된 원천 데이터가 정상적으로 정제되었는지 확인한 결과를 문서로 작성한 것이다.

- 정제 요청 데이터와 정제된 데이터 항목을 육안으로 직접 비교하여 확인한다.
- 정제된 데이터를 데이터 전환 프로그램을 이용하여 전환한 후 오류의 발생 여부를 확인하고 목적 데이터베이스에서 전환된 데이터를 확인한다.
- 정제된 데이터의 확인 결과를 반영한 데이터 정제보고서를 정제ID별로 작성한다.
- 데이터 정제보고서에는 데이터 전환 결과 외에도 오류 데이터의 원인, 실제 데이터 정제 건수, 향후 대응 방안 등을 포함한다.

출제예상

1. 다음이 설명하는 것은?

> 정제가 필요한 원천 데이터와 오류를 생성하는 원천 프로그램을 수정하기 위해 데이터 정제와 관련된 전반적인 내용을 문서로 작성한 것이다.

① 데이터 정제요청서
② 데이터 정제보고서
③ 데이터 정제분석서
④ 데이터 정제확인서

문제의 지문에 제시된 문서는 데이터의 정제를 요청하는 문서입니다.

출제예상

2. 다음 중 데이터 정제보고서에 대한 설명으로 가장 옳지 않은 것은?

① 데이터 정제 결과를 확인한 후 정제된 결과를 반영한 보고서이다.
② 데이터 정제보고서는 요청일자별로 묶어서 작성한다.
③ 정제를 요청한 원천 데이터와 정제된 데이터 항목을 육안으로 일일이 비교한다.
④ 데이터 정제보고서는 원천 데이터의 오류 데이터 원인과 향후 대응 방안을 함께 작성한다.

데이터 정제보고서는 정제ID별로 작성합니다.

출제예상

3. 다음 중 데이터 정제 요건 목록의 정제 유형과 그 예가 잘못된 것은?

① 유효성 : 생년월일이 현재 일자보다 큰 경우
② 완전성 : 고객 정보에 주민등록번호 또는 사업자등록번호가 없는 경우
③ 일치성 : 서로 다른 사람의 주민등록번호가 동일한 경우
④ 기타 : 주요 유형 이외의 정제 유형

일치성은 성별이 여자이면서 주민등록번호 뒤 7자리 중 첫 번째가 1 또는 3인 경우처럼 상호 관련 있는 항목이 서로 다른 경우를 의미합니다.

▶ 정답 : 1. ① 2. ② 3. ③

예 · 상 · 문 · 제 · 은 · 행

해설은 529쪽에 있습니다.

1. 운영 중인 기존 정보 시스템에 축적되어 있는 데이터를 추출하여 새로 개발할 정보 시스템에서 운영 가능하도록 변환한 후 적재하는 일련의 과정을 무엇이라고 하는가?

① 데이터 보안
② 데이터 변환
③ 데이터 전환
④ 데이터 복원

2. 다음 중 데이터 전환 계획서에 포함되는 항목이 아닌 것은?

① 데이터 전환 개요
② 데이터 전환 구성도
③ 데이터 전환 방안
④ 데이터 정비 방안

3. 데이터 전환 계획서 작성에 대한 설명으로 잘못된 것은?

① 데이터 전환 참여자들의 빠른 의사소통을 할 수 있도록 데이터 전환 작업을 위한 전제 조건이나 제약 사항은 간략히 기술한다.
② 데이터 전환 대상 및 범위에는 단위 업무별로 데이터 전환 대상 정보, 해당 업무에 사용되는 Table의 수, 데이터 크기를 기술한다.
③ 전환 단계별 DISK 용량은 전환 검증, 시험 단계, 본 전환 단계별로 요구되는 File 공간과 DB 공간을 산정하여 기술한다.
④ 데이터 전환 일정 작성 시 데이터 전환 관계자들 사이의 보고나 의사소통 등 효과적인 일정 관리를 위해 도식화된 일정도 작성하여 포함한다.

4. 데이터 전환 계획서의 데이터 전환 방안 항목에 대한 설명 중 잘못된 것은?

① 데이터 전환 계획 항목에는 데이터 전환 시간을 단축하기 위해 선 전환, 본 전환, 후 전환으로 분리하여 계획을 수립한 후 관련 내용을 기술한다.
② 데이터 전환 규칙 항목에는 데이터 전환 과정에서 공통적으로 적용해야 할 규칙들을 기술한다.
③ 데이터 전환 방법 항목에는 단위 업무별로 데이터 전환 방법을 기술하되, 데이터 전환 시 업무별로 요구되는 전제 조건은 제외한다.
④ 데이터 전환 설계 항목에는 업무별로 전환 대상과 전환 제외 대상을 기술하고 원천 시스템 테이블과 목적 시스템 테이블의 매핑 정의서를 작성한다.

5. 검증 방법에 따른 데이터 검증에 대한 설명으로 잘못된 것은?

① 로그 검증 : 데이터 전환 과정에서 작성하는 추출, 전환, 적재 로그를 검증함
② 기본 항목 검증 : 로그 검증 외에 별도로 요청된 검증 항목에 대해 검증함
③ 정합성 검증 : 사전에 정의된 업무 규칙을 기준으로 데이터 전환의 정합성을 검증함
④ 값 검증 : 숫자 항목의 합계 검증, 코드 데이터의 범위 검증, 속성 변경에 따른 값 검증을 수행함

6. 다음 중 데이터 검증 방법 중 로그 검증이 사용되는 검증 단계가 아닌 것은?

① 추출
② 전환
③ DB 적재
④ 전환 완료 후

7. 원천 시스템의 데이터를 목적 시스템의 데이터로 전환하는 과정이 정상적으로 수행되었는지 여부를 확인하는 과정은 무엇인가?

① 데이터 측정
② 데이터 변환
③ 데이터 정제
④ 데이터 검증

8. 다음 중 오류 데이터를 찾기 위해 원천 및 목적 시스템 데이터의 정합성 여부를 확인하는 작업은?

① 오류 데이터 측정
② 데이터 품질 분석
③ 오류 데이터 정제
④ 데이터 보안 검증

9. 다음 중 오류 데이터 분석 시 오류 상태에 대한 설명으로 가장 옳지 않은 것은?

① Deferred : 오류 수정을 연기한 상태
② Fixed : 오류를 수정한 상태
③ Closed : 오류 수정을 실패한 상태
④ Classified : 오류가 아니라고 판단된 상태

▶ 정답 : 1. ③ 2. ② 3. ① 4. ③ 5. ③ 6. ④ 7. ④ 8. ② 9. ③

해설은 529쪽에 있습니다.

10. 다음 중 목적 시스템 데이터의 품질 검증을 위한 정합성 항목과 가장 거리가 먼 것은?

① 보고서 항목 또는 통계 수치

② 영업점의 실제 존재 여부 확인

③ 특수 관계의 고객에 대한 데이터 확인

④ 특정 기준으로 분류된 좌수 및 잔액

11. 다음 중 데이터 정제요청서에 대한 설명으로 가장 옳지 않은 것은?

① 데이터 정제요청서의 정제 유형에는 완전성, 유효성, 일치성 등이 있다.

② 데이터 정제요청서의 상태에는 요건 제기, 1~3차 검토, 조치, 확인 등이 있다.

③ 데이터 정제요청서의 정제 방법에는 상, 중, 하 등이 있다.

④ 데이터 정제요청서에는 정제 검토 시 빠른 의사 결정을 내릴 수 있도록 해결 방안을 같이 작성한다.

12. 다음 중 데이터 정제요청서에 대한 설명으로 옳은 것은?

① 요청사항 및 조치사항 등 데이터 정제와 관련된 전반적인 내용을 작성한 문서이다.

② 정제된 원천 데이터가 정상적으로 정제되었는지를 확인한 결과를 작성한 문서이다.

③ 오류 관리 목록의 각 항목에 대해 정제 유형을 분류하고 현재의 정제 상태를 작성한 문서이다.

④ 데이터 품질 분석을 기반으로 오류 데이터와 정상 데이터의 수를 측정하여 작성한 문서이다.

▶ 정답 : 10. ② 11. ③ 12. ①

2. Section 116

데이터 전환 계획서의 주요 항목

데이터 전환 개요, 데이터 전환 대상 및 범위, 데이터 전환 환경 구성, 데이터 전환 조직 및 역할, 데이터 전환 일정, 데이터 전환 방안, 데이터 정비 방안, 비상 계획, 데이터 복구 대책

3. Section 116

데이터 전환 작업을 위한 전제 조건이나 제약 사항을 정확히 기술하여 데이터 전환 참여자들이 작업을 진행하는 과정에서 혼선이 빚어지지 않도록 해야 한다.

4. Section 117

데이터 전환 방법 항목에는 단위 업무별로 데이터 전환 방법을 기술하되, 데이터 전환 시 업무별로 요구되는 전제 조건도 함께 기술한다.

5. Section 118

③번의 내용은 응용 데이터 검증에 대한 설명이다.

6. Section 118

전환 완료 후에는 응용 프로그램 검증이나 응용 데이터 검증이 수행된다.

8. Section 119

- **오류 데이터 측정** : 데이터 품질 분석을 기반으로 정상 데이터와 오류 테이블의 수를 측정하여 오류 관리 목록을 작성하는 것
- **오류 데이터 정제** : 오류 관리 목록의 각 항목을 분석하여 원천 데이터를 정제하거나 전환 프로그램을 수정하는 것

9. Section 119

Closed는 수정된 오류에 대해 테스트를 다시 했을 때 오류가 발견되지 않은 상태를 의미한다.

10. Section 119

영업점의 실제 존재 여부 확인은 원천 시스템 데이터의 품질 검증을 위한 정합성 항목에 해당한다.

11. Section 120

데이터 정제요청서의 정제 방법에는 원천, 전환, 모두 등이 있다.

12. Section 120

②번은 데이터 정제보고서, ③번은 데이터 정제 요건 목록, ④번은 오류 관리 목록에 대한 설명이다.