

[정답 및 해설] [데이터 입출력 구현>논리 데이터저장소 확인]

논리 데이터저장소 확인 1. 논리 데이터 모델

- * 논리 데이터 모델
 - 개념적 데이터 모델을 컴퓨터가 이해할 수 있도록 표현
 - 목표 DBMS에 맞추어 논리적 모델로 설계 (관계형, 객체지향형, 객체관계형, 계층형, 망형 모델)
 - 정규화(Normalization) 과정 수행
 - 단순히 데이터 모델이라고 하면 논리적 데이터 모델을 의미

논리 데이터저장소 확인 2. ① 개체 ② 속성 ③ 관계

- * 구성 요소
 - 개체(Entity): 데이터베이스에 표현하려고 하는 현실 세계의 대상체
 - 속성(Attribute): 개체의 성질, 분류, 식별, 수량, 상태 등을 나타내는 논리적 단위
 - 관계(Relationship): 두 개체 간에 의미 있는 연결

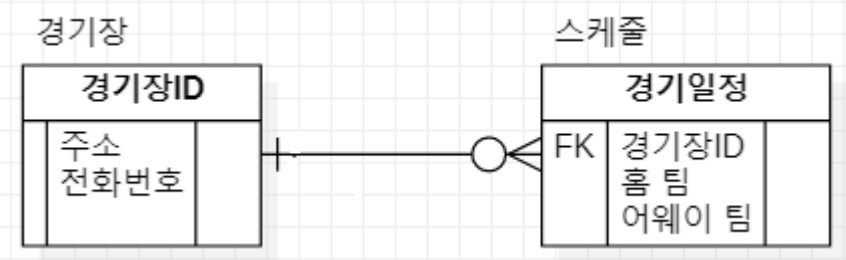
논리 데이터저장소 확인 3. 구조, 연산, 제약조건

- * 데이터 모델 구성요소
 - 구조(Structure): 개체들 간의 관계
 - 연산(Operation): 데이터 처리하는 방법
 - 제약조건(Constraint): 실제 데이터의 논리적인 제약조건

논리 데이터저장소 확인 4. ① (¬) ② (┌) ③ (┐)

- * 데이터 모델 종류
 - 개념적 데이터 모델: 현실세계를 추상적으로 표현 (E-R 모델)
 - 논리적 데이터 모델: 개념적 데이터 모델을 컴퓨터가 이해할 수 있도록 표현 (관계 데이터 모델)
 - 물리적 데이터 모델: 저장레코드 양식의 설계 및 물리적 구조 데이터 표현

논리 데이터저장소 확인 5



- * IE 표기법
 - 하나의 경기장은 여러 게임의 스케줄을 가질 수 있고, 스케줄이 없을 수도 있다. (선택, 다중)
 - 하나의 스케줄은 하나의 경기장에 꼭 배정된다. (필수)

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

[정답 및 해설] [데이터 입출력 구현>논리 데이터저장소 확인]

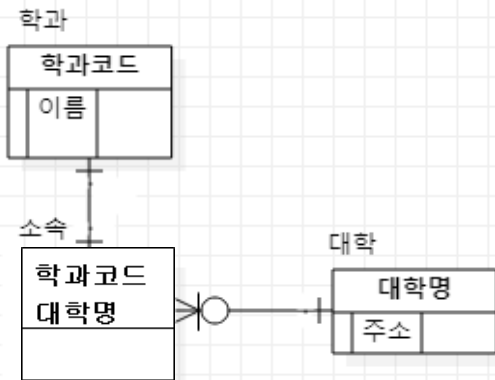
논리 데이터저장소 확인 6.



* 개념 데이터 모델 구성 요소

- 개체(Entity): 데이터베이스에 표현하려고 하는 현실 세계의 대상체
- 속성(Attribute): 개체의 성질, 분류, 식별, 수량, 상태 등을 나타내는 논리적 단위
- 관계(Relationship): 두 개체 간에 의미 있는 연결

논리 데이터저장소 확인 7.



* IE표기법

- 하나의 학과는 하나의 대학에 꼭 소속된다. (1:1)
- 하나의 대학에는 여러 개의 학과가 소속될 수 있다. (m:1)

논리 데이터저장소 확인 8. (\neg), (\sqsubset), (\equiv), (\sqsupset)

* 논리 데이터 모델의 품질 기준

- 정확성
- 완전성
- 준거성
- 최신성
- 일관성
- 활용성

논리 데이터저장소 확인 9. ① 완전성 ② 최신성 ③ 활용성 ④ 일관성 ⑤ 정확성 ⑥ 준거성

* 논리 데이터 모델의 품질 기준

- 정확성
- 완전성
- 준거성
- 최신성
- 일관성
- 활용성

논리 데이터저장소 확인 10. 개체, 속성, 관계, 모델 전반 등

* 논리 데이터 모델 품질 검토 체크리스트

: 개체, 속성, 관계, 모델 전반 등

논리 데이터저장소 확인 11. 정규화(Normalization)

정규화에 대한 설명이다.

논리 데이터저장소 확인 12. 제3정규형 또는 3NF

제3정규형(3NF)에 대한 설명이다.

[정답 및 해설] [데이터 입출력 구현>논리 데이터저장소 확인]

논리 데이터저장소 확인 13. ① 삽입 이상 ② 갱신 이상
③ 삭제 이상

구분	내용
갱신 이상	- 반복된 데이터 중에 일부만 수정하면 데이터의 불일치가 발생 예) '이상희'의 주소를 변경할 경우 모든 속성(칼럼)의 주소를 변경해야 한다. 만약 하나만 변경할 경우 데이터의 불일치가 발생한다.
삽입 이상	- 불필요한 정보를 함께 저장하지 않고는 어떤 정보를 저장하는 것이 불가능 예) 'MOS' 과목을 추가할 경우, 불필요한 회원정보까지 추가해야 한다.
삭제 이상	- 유용한 정보를 함께 삭제하지 않고는 어떤 정보를 삭제하는 것이 불가능 예) '이상희'의 데이터를 삭제하고자 할 경우, 정보처리라는 과목까지 삭제되어 버린다.

논리 데이터저장소 확인 14. ① 함수 종속 ② $X \rightarrow Y$

- * 논리 데이터 모델의 품질 기준
- 정확성
 - 완전성
 - 준거성
 - 최신성
 - 일관성
 - 활용성

논리 데이터저장소 확인 15. (\neg) , (\sqsubset)

- (\sqsubset) 제1정규형에 대한 설명이다.
- (\sqsupset) 제2정규형에 대한 설명이다.

논리 데이터저장소 확인 16. 제1정규형(1NF)

- 제1정규형은 테이블에 속한 모든 속성 값이 원자 값만으로 되어 있는 정규형이다.
- [회원정보] 테이블에서 한 명의 회원에 대해 여러 개의 수강 과목 정보가 발생하고 있으므로 제1정규형이 아니다. [회원정보] 테이블에서 반복되는 회원 관련 정보인 회원번호, 이름, 주소, 이메일, 주민번호를 분리하면 제1정규형인 [회원]테이블과 [수강]테이블이 만들어 진다.

[정답 및 해설] [데이터 입출력 구현>논리 데이터저장소 확인]

논리 데이터저장소 확인 17. ① 제3정규형(3NF) ② 이행적 함수적 종속 제거

- 제3정규형은 제2정규형이고, 이행적 함수적 종속 관계를 제거하여 비이행적 함수적 종속 관계를 만족하는 정규형이다.

1. 이행적 함수적 종속: 주문번호→연락처

- [주문]테이블에서 이행적 함수적 종속(주문번호→연락처)를 제거하여 무손실 분해함으로써 제3정규형인 [주문]테이블과 [고객]테이블이 생성된다.

논리 데이터저장소 확인 18. BCNF

- BCNF는 제3정규형이고, 테이블에서 모든 결정자가 후보키인 정규형이다.

1. 후보키: (학번, 과목), (학번, 교수)

2. 함수적 종속: (학번, 과목)→교수, 교수→과목

3. 결정자: (학번, 과목), 교수

- 교수 속성이 결정자이나 후보키가 아니므로 교수 속성을 분리하면 [수강] 테이블은 모든 결정자가 후보키므로 BCNF를 만족한다.

논리 데이터저장소 확인 19. ① (\neg) ② (\neg) ③ (\equiv) ④ (\supset)

* 정규화 절차

: 제1정규형→제2정규형→제3정규형→BCNF→제4정규형→제5정규형

- 제1정규형→제2정규형: 부분 함수적 종속 제거, 키가 아닌 모든 속성이 기본키 그룹에 완전하게 함수적 종속

- 제2정규형→제3정규형: 이행적 함수적 종속 제거, 키가 아닌 모든 속성이 기본키에 직접 종속(비이행적)

논리 데이터저장소 확인 20. ① (학번, 수강과목)→성적
② 학번→이름

* 완전 함수적 종속: [성적]테이블에서 (학번, 수강과목)이 기본키인데, 성적은 학번과 수강과목이 같을 경우에는 항상 같은 성적이다. 즉, 성적은 학번과 수강과목에 의해서만 결정되므로 성적은 기본키(학번, 과목번호)에 완전 함수적 종속이다.

* 부분 함수적 종속: 이름은 수강과목에 상관 없이 학번이 같으면 항상 같은 이름이다. 즉, 기본키(학번, 과목번호)의 일부인 학번에 의해서 학년이 결정되므로 부분 함수적 종속이다.

[정답 및 해설] [데이터 입출력 구현>논리 데이터저장소 확인]

논리 데이터저장소 확인 21. ① 제1정규형② 제2정규형

- 제2정규형 (2NF)

: 제1정규형이고, 부분함수적 종속을 제거하여 완전 (충분한) 함수적 종속을 만족하는 정규형.

[정답 및 해설] [데이터 입출력 구현>물리 데이터저장소 확인]

물리 데이터저장소 설계 1. 로우, 컬럼, 기본키, 외래키

* 테이블 구성 요소

- 로우
- 컬럼
- 기본키
- 외래키

물리 데이터저장소 설계 2. ① 컬럼 ② 로우 ③ 기본키

* 테이블 구성 요소

- 로우(Rows): 튜플, 인스턴스, 어커런스라고도 한다.
- 컬럼(Columns): 테이블의 관리 항목에 대한 Value를 저장
- 기본키(Primary keys): 하나의 컬럼 혹은 몇 개의 컬럼 조합으로 어떤 경우라도 테이블 내에 동일한 값을 갖는 튜플이 존재하지 않도록 한다.
- 외래키(Foreign keys): 외부 데이터 집합과의 관계(Relationship)를 구현한 구조

물리 데이터저장소 설계 3. ① 기본키 ② 외래키

* 논리 데이터 모델-물리 데이터 모델 변환 용어

논리 데이터 모델	물리 데이터 모델
개체(Entity)	테이블
속성(Attribute)	컬럼
주 식별자(Primary UID)	기본키
보조(대체) 식별자	대체키
관계(Relationship)	외래키

물리 데이터저장소 설계 4. 외래키(Foreign key)

- 외래키(Foreign keys)

: 외부 데이터 집합과의 관계(Relationship)를 구현한 구조

물리 데이터저장소 설계 5. (=)

단위 개체(Entity)를 테이블로 변환은 논리 데이터 모델에서 정의된 엔티티를 물리 데이터 모델의 테이블로 변환하는 것이다.

[정답 및 해설] [데이터 입출력 구현>물리 데이터저장소 확인]

물리 데이터저장소 설계 6. ① (ㄷ) ② (ㄱ) ③ (ㄴ)

- * 슈퍼타입/서브타입을 테이블로 변환
 - 슈퍼타입 기준 테이블 변환
 - 서브타입 기준 테이블 변환
 - 개별타입 기준 테이블 변환

물리 데이터저장소 설계 7. (ㄱ), (ㄴ)

- (ㄷ) 슈퍼타입 기준 테이블 변환의 장점이다.
- (ㄹ) UID(식별자)가 단위 테이블마다 존재하므로 유지관리가 어렵다.

물리 데이터저장소 설계 8. (ㄴ), (ㄹ)

- (ㄱ) 여러 테이블을 조인하지 않아도 되므로 수행 속도가 빨라진다.
- (ㄷ) 슈퍼타입 기준 테이블 변환은 서브타입을 슈퍼타입에 통합하여 하나의 테이블로 만드는 것이므로 여러 테이블을 통합한 뷰를 이용하지 않아도 된다. 슈퍼타입 기준 테이블 변환시 뷰를 이용하여 각각의 서브타입만을 액세스하거나 수정할 수 있는 장점이 있다.

물리 데이터저장소 설계 9. (ㄷ)

- * 개별타입 기준 테이블 변환
 - 슈퍼타입과 서브타입을 각각 테이블로 변환한 경우이다.
 - 슈퍼타입과 서브타입 테이블 간에는 1:1 관계가 생성된다. (한 쪽을 모두 합치면 전체와 같게 된다.)

물리 데이터저장소 설계 10. ① (ㄱ) ② (ㄷ) ③ (ㄴ)

- * 속성을 칼럼으로 변환
 - 일반 속성 변환
 - Primary UID - 기본키(Primary Key) 변환
 - Secondary (Alternate) UID - Unique Key 변환

물리 데이터저장소 설계 11. ① 1:1 ② 1:N ③ N:M

- * 관계를 외래키(Foreign Key)로 변환
 - : n 관계에서 1 영역에 있는 기본키를 n 영역의 외래키로 선언한다.

논리 데이터저장소 확인 12. 반정규화

- * 반정규화
 - 논리 모델링에서 수행된 정규화 작업을 통해 데이터 모델은 데이터의 중복의 최소화와 데이터의 일관성, 정확성, 안정성 등을 보장하는 데이터 구조를 완성할 수 있다.
 - 반정규화는 정규화된 데이터 모델이 시스템의 성능 향상, 개발 과정의 편의성, 운영의 단순화를 목적으로 수행되는 의도적인 정규화 원칙 위배 행위를 의미한다.

[정답 및 해설] [데이터 입출력 구현>물리 데이터저장소 확인]

물리 데이터저장소 설계 13. 분할

테이블 분할은 테이블을 수직 또는 수평으로 분할하는 것이다.

물리 데이터저장소 설계 14. 테이블 분할, 중복 테이블 생성, 중복 속성 생성, 테이블 통합, 테이블 제거 등

* 반정규화 방법

- 테이블 분할
- 중복 속성 생성
- 테이블 제거
- 중복 테이블 생성
- 테이블 통합

물리 데이터저장소 설계 15. ① (=) ② (└) ③ (┐)

* 중복 테이블 추가(생성)

- 집계 테이블 추가
- 진행 테이블 추가
- 특정부분만을 포함하는 테이블 추가

물리 데이터저장소 설계 16. (=)

* 중복 속성 생성: 자주 사용하는 속성을 추가로 더 생성한다.

물리 데이터저장소 설계 17. ① (=) ② (└) ③ (┐) ④ (=)

* 테이블 수직 분할(Table Vertical Partitioning)

- 갱신 위주의 속성 분할
- 자주 조회되는 속성 분할
- 크기가 큰 속성 분할
- 보안을 적용해야 하는 속성 분할

물리 데이터저장소 설계 18. 테이블 통합

테이블 통합은 두 개의 테이블이 조인되는 경우가 많아 하나의 테이블로 합쳐 사용하는 것이 성능 향상에 도움이 될 경우 수행한다.

물리 데이터저장소 설계 19. Cascade

*삭제 제약조건>Delete Constraint)

- Cascade: 참조한 테이블에 있는 외래키와 일치하는 모든 Row가 삭제된다.

* 수정 제약조건(Update Constraint)

- Cascade: 참조한 테이블에 있는 외래키와 일치하는 모든 Row가 수정된다.

물리 데이터저장소 설계 20. 인덱스(Index)

인덱스(Index)에 대한 설명이다.

[정답 및 해설] [데이터 입출력 구현>물리 데이터저장소 확인]

물리 데이터저장소 설계 21. 분포도

- '성별'은 가질 수 있는 값이 '남' 또는 '여'이기 때문에 '성별' 컬럼으로 인덱스를 설계한다면 전체 튜플의 50% 정도가 검색 결과에 포함될 것이다. 이런 경우 높은 분포도를 가지므로 '성별'로 인덱스를 설계하지 않는 것이 좋다.
- '점수'는 가질 수 있는 값이 '0~100'이기 때문에 '점수' 컬럼으로 인덱스를 설계하면 좁은 분포도를 가질 수 있다. 만약 '50'을 검색할 경우 Table Scan(순차 접근)하지 않고 인덱스를 검색해서 빠르게 검색할 수 있다.

물리 데이터저장소 설계 22. ① Clustered Index ② Non-Clustered Index

- * Clustered Index
 - 물리적으로 데이터(테이블)를 정렬 시킴
 - 인덱스를 검색하지 않아도 데이터에서 빠르게 찾을 수 있음 (물리적으로 인접하므로)
 - 한 테이블에 하나만 만들 수 있음
 - 데이터 삽입, 삭제 발생시 재정렬
- * Non-Clustered Index
 - 인덱스 키 값만 정렬
 - 인덱스 검색 후 데이터를 찾음
 - 한 테이블에 여러 개 만들 수 있음

물리 데이터저장소 설계 23. B트리 인덱스, 비트맵 인덱스, 비트맵 조인 인덱스, 함수 기반 인덱스, 도메인 인덱스 등

* 인덱스 종류

- B트리 인덱스
- 비트맵 인덱스
- 비트맵 조인 인덱스
- 함수 기반 인덱스
- 도메인 인덱스

물리 데이터저장소 설계 24. ① (┐) ② (≡) ③ (=) ④ (┐)

* 인덱스 종류

- B트리 인덱스
- 비트맵 인덱스
- 비트맵 조인 인덱스
- 함수 기반 인덱스
- 도메인 인덱스

물리 데이터저장소 설계 25. ① (┐), (≡) ② (┐)

* B트리 인덱스: 루트 노드에서 하위 노드로 키 값의 크기를 비교해 나가면서 단말 노드에서 찾고자 하는 데이터를 검색한다.

* 비트맵 인덱스: 인덱스 컬럼의 데이터를 bit 값 0,1로 변환하여 인덱스 키로 사용한다.

[정답 및 해설] [데이터 입출력 구현>물리 데이터저장소 확인]

물리 데이터저장소 설계 26. 뷰(View)

뷰(View)에 대한 설명이다.

물리 데이터저장소 설계 27. $(\supset) \rightarrow (\subset) \rightarrow (\neg)$

* 뷰 설계 순서

- 1단계: 대상 테이블 선정
- 2단계: 대상 컬럼 선정
- 3단계: 뷰 정의서 작성

물리 데이터저장소 설계 28. ① 범위 분할 ② 해시 분할
③ 조합 분할

* 파티션 종류

- 범위 분할(Range Partitioning)
- 해시 분할(Hash Partitioning)
- 조합 분할(Composite Partitioning)

물리 데이터저장소 설계 29. 파티션(Partition)

파티션(Partition)에 대한 설명이다.

물리 데이터저장소 설계 30. $(\supset) \rightarrow (\subset) \rightarrow (\neg)$

* 파티션 순서

- 1단계: 파티션의 종류 결정
- 2단계: 파티션 키의 선정
- 3단계: 파티션 수의 결정

물리 데이터저장소 설계 31. ① Local ② Global

* Local Partitioned Index

- 테이블 파티션과 인덱스 파티션이 1:1 대응되도록 구성 → 동등 분할
- 테이블 파티션키 컬럼 기준으로 동일하게 인덱스 파티션이 구성됨
- 테이블 파티션의 구조가 변경되어도 인덱스 재생성이 필요 없으므로 관리가 쉬움 → 일반적으로 사용

* Global Partitioned Index

- 테이블 파티션과 인덱스 파티션이 독립적으로 구성

물리 데이터저장소 설계 32. 클러스터

* 클러스터

- 데이터 저장 시 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 기법이다.

물리 데이터저장소 설계 33. $(\neg) \rightarrow (\supset) \rightarrow (\supseteq) \rightarrow (\subset)$

* DB 용량 산정 순서

- 1단계: 용량 분석의 목적 이해하기
- 2단계: 기초 데이터 수집하기
- 3단계: 오브젝트(테이블, 인덱스 등)별로 용량 산정하기
- 4단계: 디스크 용량 산정

[정답 및 해설] [데이터 입출력 구현>물리 데이터저장소 확인]

물리 데이터저장소 설계 34. 디스크(Disk)

디스크 구성 설계는 테이블에 저장할 데이터 양과 인덱스, 클러스터 등이 차지하는 공간을 예측해서 설계하는 것이다.