1장. 데이터베이스 시스템의 개요

1. 다음 용어들을 설명하시오.

데이터베이스, DBMS, 데이터 모델, 데이터베이스 시스템

- **데이터베이스** : 조직체의 응용 시스템에서 공용으로 사용하는 데이터들을 구조적으로 통합하여 저장한 데이터 집합체이다. 컴퓨터 내부의 하드디스크에 저장된다.
- **DBMS** : 소프트웨어 프로그램으로 데이터베이스를 관리하는 기능을 하며, 사용자에게 편리하고 효율적인 데이터베이스 사용 환경을 제공한다. 컴퓨터 주기억 장치에 상주한다.
- **데이터 모델** : 데이터베이스 시스템에서 데이터를 저장하는 이론적 방법에 관한 것으로, 데이터베이스에 데이터가 어떻게 구조화되어 저장되는 지를 결정한다.
- **데이터베이스 시스템** : 데이터베이스 관리 시스템(DBMS), 데이터베이스, 데이터 모델 이 3가지로 구성된 시스템을 말한다.

2. 데이터베이스의 개념 네 가지를 설명하시오.

- **통합된 데이터** : 데이터의 중복을 최소화하면서 통합되는 데이터
- **저장된 데이터** : 디스크와 같은 매체에 저장된 데이터
- **운영 데이터** : 조직의 업무를 목적으로 사용되는 운영 데이터
- **공용 데이터** : 조직체의 모든 구성원이 공용으로 사용하는 공용 데이터

3. 데이터베이스의 특징 네 가지를 설명하시오.

- **실시간 접근성** : 사용자의 요청에 따라 실시간으로 서비스된다.
- **계속적인 변화** : 데이터베이스가 저장하는 내용은 특정 시점의 상태를 나타낸다. 즉 시간에 따라 데이터가 계속 변한다.
- **동시 공유** : 데이터베이스는 조직체의 공용정보로 여러 사용자 및 프로그램이 동시에 공유한다.
- **내용에 따른 참조** : 데이터베이스에 저장된 데이터는 데이터의 물리적인 위치가 아니라 데이터 값(내용)에 따라 참조된다.

4. 파일 시스템과 DBMS의 데이터 접근 방법의 차이를 설명하시오.

- **파일 시스템** : 데이터는 다양한 언어(c, java 등)로 작성되어 운영체제의 파일에 저장된다. 각각의 응용 프로그램마다 별도의 파일을 가지며 직접 저장된 파일에 접근한다.
- **DBMS** : 데이터베이스는 표준화된 형식으로 저장되며 여기에 대한 접근은 모두 응용프로그램에서 SQL를 이용하여 이루어진다.

5. 파일 시스템과 DBMS의 장단점을 비교하여 설명하시오.

	파일시스템	DBMS
장점	• 운영체제를 설치할 때 함께 설치되기 때문에 별도의 구입비용 없이 사용가능하다. • 보통 소규모 프로그램에서 사용해 주기억장치를 적게 사용하고, 속도가 빠르다.	• DBMS에서는 데이터중복을 최소화한 통합 데이터베이스를 구축한다. ✓ 프로그램-데이터가 독립적이어서 유지보수 비용 적게 든다. ✓ 여러 사용자가 통합된 데이터베이스를 공유하며 동시에 데이터베이스에 접근할 수 있어 데이터 공유에 용이하다. ✓ 모든 데이터를 하나의 데이터베이스에 통합하므로 데이터 중복과 불일치 감소한다. ✓ 중복제거로 데이터 일관성 유지한다. ✓ 데이터베이스를 통한 데이터의 중앙집중화로 보안조치가 용이하다. • 데이터에 대한 모든 접근은 데이터를 처리하는 DBMS가 담당한다. ✓ 데이터를 응용프로그램으로부터 분리시켜 데이터 독립성이 향상된다. ✓ 데이터 표준화 시행이 쉽다. ✓ 데이터 무결성이 유지된다. • 데이터베이스 접근 시 시스템이 다운되었을 경우 이전의 일관된 데이터베이스 상태 복구 가능하다.
단점	• 데이터 정의가 프로그램에 내포되어 있다. ✓ 프로그램-데이터 독립성이 없어 유지보수 비용 크다. ✓ 각 응용 프로그램마다 파일을 갖고 있으며, 응용 프로그램마다 각각 다른 언어로 작성되어 있어 데이터 공유가 어렵다(공유 역시 파일 단위로 수행). • 프로그램에서 데이터를 접근하고 조작하는 것 이외에 별도의 제어기능이 없다. ✓ 중복된 데이터 변경을 제어하기 어려워 각 파일에 데이터 중복 저장 가능하다. 그래서 중복된 데이터가 변경될 때 두 파일 모두 수정하지 않으면 데이터 불일치가 발생한다. ✓ 다수 사용자들을 위한 동시성 제어가 제공되지 않아 데이터 일관성이 훼손된다. ✓ 사용자 권한에 따른 접근제어 어려워 보안조치가 미흡하다. • 응용프로그램에서 파일 내 데이터 수정 시 시스템이 다운되었을 경우 회복기능이 없어 데이터 일관성 복구 어렵다. • 데이터의 의미와 데이터간의 상호관계를 나타내는 데이터 모델링 개념이 부족하여 프로그래머의 생산성 낮다.(하나하나 기능 정의)	• DBMS의 구입비용 및 추가적인 하드웨어 구입비용 등 초기 투자비용이 크다. • 데이터 관리의 오버헤드와 위험이 존재한다. • 응용프로그램이 단순하고, 데이터 변경이 자주 일어나지 않고, 다수 사용자의 접근이 필요하지 않을 때는 부적합하다.

6. 데이터 모델을 설명하고 종류를 나열하시오.

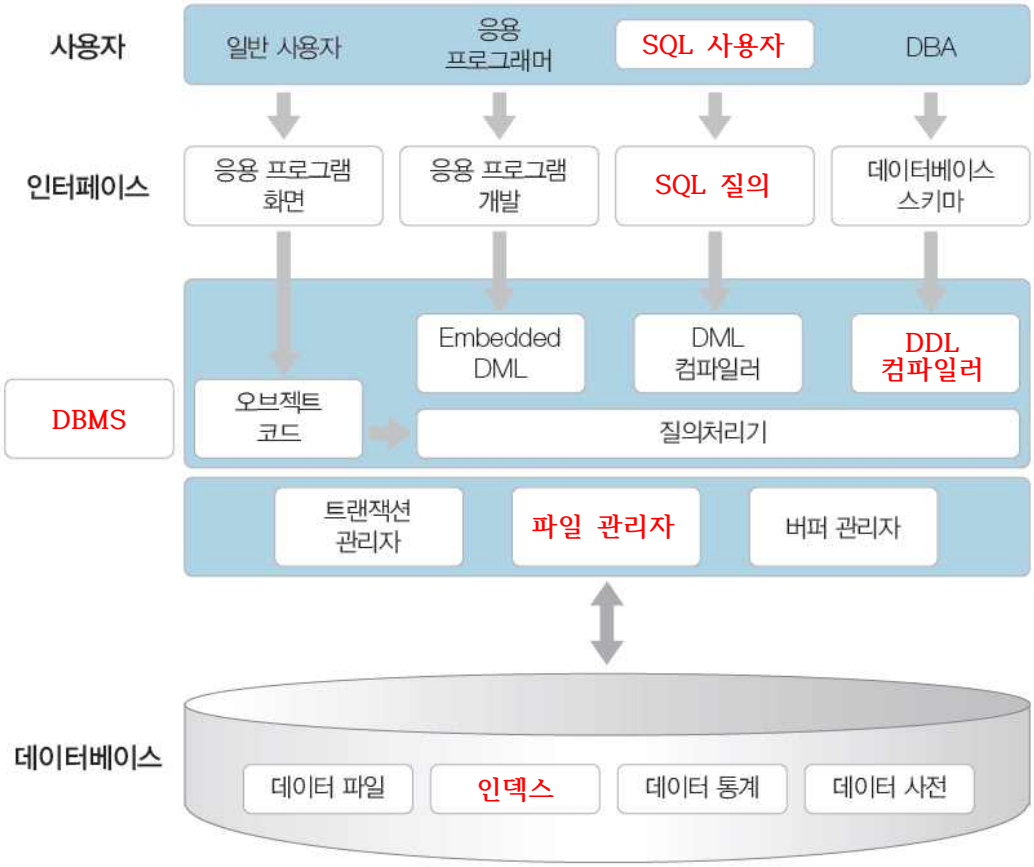
데이터 모델은 데이터베이스 시스템에서 데이터를 저장하는 이론적 방법에 관한 것으로, 데이터베이스에 데이터가 어떻게 구조화되어 저장되는 지를 결정한다. 데이터 모델을 구분하는 가장 큰 기준은 데이터들 간에 관계를 표현하는 방법인데 대표적으로 3가지가 있다.

- **포인터를 사용하여 데이터 간의 관계를 표현하는 모델**
  계층 데이터 모델, **네트워크 데이터 모델**이 여기에 해당한다. 포인터를 사용하여 데이터가 저장된 포인터 값을 다른 테이블 데이터에 저장하는 방법으로 데이터를 연결한다. 데이터를 직접 찾아가 프로그램 속도는 빠르지만, 실제적으로 포인터 연산 구현이 어려워 응용 프로그램 개발 속도가 더디다는 단점이 있다.

- **속성 값 사용하여 데이터 간의 관계를 표현하는 모델**
  관계 데이터 모델이 여기에 해당된다. 테이블의 <속성, 속성값>을 다른 테이블의 <속성, 속성값>에 직접 저장하는 방법이다. 데이터를 찾을 때 이 <속성, 속성값>을 이용하여 찾는다. 포인터를 사용하는 것보다 개념이 쉽고 프로그램 개발이 빠르지만 속도가 느리다는 단점이 있다.

- **객체 식별자를 사용하여 데이터 간의 관계를 표현하는 모델**
  모든 객체는 고유의 식별자를 갖고 있기 때문에 테이블을 객체 개념으로 보고 객체의 고유 식별자를 다른 테이블에 저장하는 방법이다. **객체지향 모델**이 여기에 속한다(객체지향 언어에서 개념을 도입한 것이다).

7. 다음 데이터베이스 시스템의 구성도를 보고, 빈 곳에 알맞는 용어를 써 넣으시오.



8. 다음 데이터베이스 사용자들의 역할을 설명하시오.

일반 사용자, 응용 프로그래머, SQL 사용자, DBA

- **일반 사용자** : 데이터를 다루는 업무를 맡은 사람으로 특별한 지식 없이 응용 프로그래머가 개발한 **프로그램**을 이용하여 데이터베이스에 접근해서 데이터를 사용한다.
- **응용 프로그래머** : 일반 사용자가 사용할 수 있는 프로그램을 개발하는 사람이다. 일반 프로그래밍 언어(C, Java, JSP 등)와 SQL 등을 사용하여 일반 사용자를 위한 데이터 관리 응용 로직을 개발한다.
- **SQL 사용자** : SQL을 사용하여 업무를 처리하는 IT부서의 업무 담당자이다. 응용프로그램으로 구현되어 있지 않은 업무를 SQL을 사용하여 처리하고 데이터를 모니터링하는 업무를 수행한다.
- **DBA** : 데이터베이스 운영 조직의 데이터베이스 시스템을 **총괄**하는 사람이다. 데이터베이스의 설계, 구현, 유지보수의 전 과정과 데이터베이스 사용자 통제, 보안, 성능 모니터링, 데이터 전체 파악 및 관리 등 제반 업무를 담당한다.

9. 데이터 독립성을 정의하고, 데이터베이스 시스템에서 그 중요성을 설명하시오.
데이터 독립성은 하위 단계의 내용을 추상화하여 상위 단계에서 그 세부 사항을 숨김으로써 하위 단계 내의 변경에 대해 상위 단계와 상호간 간섭이 없도록 하는 것을 의미한다. 데이터 독립성은 물리적 독립성과 논리적 독립성이 있는데 이런 데이터 독립성을 수행함으로써 데이터베이스 시스템의 운영과 관련 응용 프로그램의 유지보수가 용이해진다.

10. ANSI의 3단계 데이터베이스 구조에 대해 설명하시오.
ANSI-SPARC구조는 데이터베이스를 보는 관점을 기준으로 3개의 단계를 분리한 것으로, 데이터베이스를 3-Layer 스키마(외부, 내부, 개념 스키마)로 나누어 데이터베이스 구조를 단순화 시켜 물리적 데이터 독립성과 논리적 데이터 독립성을 유지시키는 것을 목적으로 한다.

첫 번째 층인 **외부 스키마**는 일반 사용자나 응용 프로그래머가 접근하는 계층으로 전체 데이터베이스 중에서 일반 사용자들에게 필요한 부분 스키마에 해당된다. 두 번째 층인 **개념 스키마**는 전체 데이터베이스를 일컫는 말로 조직별 하나만 존재하며 DBA가 관리한다. 개념 스키마는 데이터와의 관계, 제약사항, 무결성에 대한 내용을 포함하되, 저장장치에 독립적으로 기술한다. 마지막으로 **내부 스키마**는 DBMS의 관점에서 데이터베이스가 실제로 하드디스크에 저장되는 방법을 표현한 것으로 조직별 하나만 존재하여 인덱스, 데이터 레코드의 배치방법, 데이터 압축 등에 관한 사항을 포함한다.

DBMS는 세 가지 유형의 스키마 간의 사상을 책임진다. 총 2가지 사상이 있다. 먼저, 외부/개념 사상은 각 외부 스키마는 외부/개념 사상에 의해 개념스키마와 연관된다. 즉, 외부/개념 사상은 외부 스키마를 사용해서 입력된 사용자의 질의를 개념 단계의 스키마를 사용한 질의로 변환하는 과정을 수행한다. **논리적 데이터 독립성**을 지원한다. 두 번째로 개념/내부 사상이다. 개념/내부 사상은 위의 작업 결과를 다시 내부 단계의 스키마로 변환하여 디스크의 데이터베이스에 접근한다. 개념/내부 사상을 통해 DBMS가 개념 스키마 내의 한 논리적 레코드를 구성하는 물리적 저장장치 내의 실제 레코드를 찾을 수 있다. **물리적 데이터 독립성**을 지원한다.

11. 클라이언트/서버 구조를 설명하고, 2-tier, 3-tier 개념을 인터넷에서 찾아보시오.
클라이언트 서버 구조는 다수의 클라이언트와 하나의 서버로 구성된다. 서버는 실질적으로 작업을 수행하는 역할을 하고, 클라이언트는 서버에 작업을 요청하고 받아가는 역할을 한다. 클라이언트가 서버에 결과를 요청하면 서버가 이 요청을 받아 수행하고 결과를 반환하면 클라이언트가 이 결과 값을 받아간다.

클라이언트와 서버가 직접 연결되어 있는 모델을 **2-tier model**라 하고, 서버와 클라이언트 사이에 응용 서버가 추가된 모델을 **3-tier model**이라 한다. 3-tier model은 기존 2-tier model이 갖는 서버에 지나치게 의존적이라는 단점을 극복한 모델로 서비스의 요청이 클라이언트와 응용 서버 간에만 전송되므로 데이터베이스 시스템의 성능을 향상시킨다.

12. 우리나라 공공 데이터베이스 구축 현황을 공공데이터포털(www.data.go.kr)에서 확인해보시오. 세 개 이상의 데이터베이스를 찾아보고, 내용을 설명하시오.

- **서울도서관** : 학교도서관 개방 정보(서울시내 초, 중, 고등학교의 자치구명, 학교명, 학교주소, 면적, 연락처)
- **지하철정보** : 지하철 첫차와 막차 정보(호선, 요일, 상/하행선, 외부코드, 전철역코드, 전철역명, 첫차시간, 첫차출발역코드, 첫차출발역명, 첫차도착역코드, 첫차도착역명, 막차시간, 막차출발역코드, 막차출발역명, 막차도착역코드, 막차도착역명)
- **관광정보** : 국문 관광정보(관광지 이미지정보, 지역코드, 행사정보, 코스정보, 숙박정보, 지역기반 관광정보, 위치기반 관광정보)

13. 주요 DBMS 제조사(Oracle, IBM, Microsoft)의 국내 시장 점유율을 인터넷에서 찾아보시오.

출처 http://www.dt.co.kr/contents.html?article_no=2013012302011060746002

14. 위키피디아(www.wikipedia.org)에서 'database' 키워드를 입력하여 데이터베이스의 정의, 역사, 기술 동향을 찾아보시오.

● **정의**

여러 사람들이 공유하고 사용할 목적으로 통합 관리되는 정보의 집합이다. 논리적으로 연관된 하나 이상의 자료의 모음으로 그 내용을 고도로 구조화함으로써 검색과 갱신의 효율화를 꾀한 것이다. 즉, 몇 개의 자료 파일을 조직적으로 통합하여 자료 항목의 중복을 없애고 자료를 구조화하여 기억시켜 놓은 자료의 집합체라고 할 수 있다. 공동 자료로서 각 사용자는 같은 데이터라 할지라도 각자의 응용 목적에 따라 다르게 사용할 수 있다.

● **역사**

연도	역사
1963	데이터베이스라는 용어가 'Development and Management of Computer-Center Data Bases'라는 심포지움에서 처음 사용됨
1963	최초의 범용 DBMS 설계: GE에서 개발한 'Integerated DataStore'
1970	E.F.Codd가 관계형 데이터베이스 모델 제안
1976	Chen이 개체 관계(ER) 모델 제안
1980	개인용 컴퓨터를 위한 DBMS 개발(dBase,PARADOX 등)
1983	상용 관계 DBMS 등장(DB2, ORACLE 등)
1986	데이터베이스를 다루는 언어인 SQL이 관계형 데이터베이스 관리 시스템의 표준언어로 채택
1990~	상용 객체 지향 DBMS 등장

● **기술동향**

2012년 한국의 국내 DB산업은 DB구축 시장, DB컨설팅·솔루션 시장, DB서비스 시장 등 모든 분야에서 전년 대비 높은 성장세를 나타내고 있다. 특히 전 산업에서의 정보통신기술(ICT) 융합과 스마트 환경 확산, 빅 데이터 관련 수요가 증가하면서 향후 그 성장세는 계속될 것으로 내다봤다. 보고서에서는 DB산업의 성장을 내다보는 주요 요인으로 빅 데이터 분석·활용을 위한 기업의 신규 수요 증가, DB자산 가치 인식 증대로 인한 DB구축 투자 증가, 스마트 기반의 모바일 서비스 확산 등을 꼽고 있다.

15. 데이터베이스 전문가가 되기 위한 자격증에는 DBMS별 자격증과 일반 자격증이 있다. 각각 어떤 것들이 있는지 조사해보시오.

국내 자격증

- **DAP 데이터아키텍처 전문가(DAP, Data Architecture Professional)**
  효과적인 데이터아키텍처 구축을 위해 전사아키텍처와 데이터 품질관리에 대한 지식을 바탕으로 데이터 요건분석, 데이터 표준화, 데이터 모델링, 데이터베이스 설계와 이용 등의 직무를 수행하는 실무자를 말한다.

- **DAsP 데이터아키텍처 준전문가(DAsP, Data Architecture Semi-Professional)**
  DAP의 자격에 관한 기술 중 일부 기술을 인증하는 자격증이다.

- SQL-P
  데이터베이스와 데이터 모델링에 대한 지식을 바탕으로 데이터를 조작하고 추출하는데 있어서 정확하고 최적의 성능을 발휘하는 SQL을 작성할 수 있고, 이를 토대로 SQL을 내포하는 데이터베이스 프로그램이나 응용 소프트웨어의 성능을 최적화하거나, 이러한 성능 최적화를 지원할 수 있는 데이터베이스 개체(뷰, 인덱스 등)의 설계와 구현 등의 직무를 수행하는 전문가를 말한다.

- SQL-D
  데이터베이스와 데이터 모델링에 대한 지식을 바탕으로 응용소프트웨어를 개발하면서 데이터를 조작하고 추출하는데 있어서 정확하고 최적의 성능을 발휘하는 SQL을 작성할 수 있는 개발자를 말한다.

업체별 자격증

- **(오라클 DBMS) OCP, OCM**
  Oracle DBMS를 관리하기 위한 지식과 기술을 비롯한 유사시를 대비한 Backup 전략과 Recovery 방법, Server Side Tuning 방법과 데이터베이스 관리에 필요한 Utility 사용에 관한 인증이다.

- **(마이크로소프트 DBMS) MCIPT: DA(Database Administrator)**

2장. 관계 데이터 모델

1. 관계 데이터 모델의 릴레이션에 대한 설명 중 옳지 않은 것은?
② 릴레이션 스키마를 릴레이션 외연(~~extension~~)이라고 **내포(intension)**라고도 한다.

2. 릴레이션의 특징으로 알맞은 것은?
④ 모든 속성 값은 원자값이다.
→ 릴레이션에서 투플, 속성 간의 순서는 정해져 있지 않으며 중복 투플을 허용하지 않는다.

3. 하나의 속성이 가질 수 있는 값을 총칭하여 무엇이라 하는가?
③ 도메인

4. 외래키(FK, Foreign Key)에 대한 설명으로 옳은 것은?
① 릴레이션 R1에 속한 속성 집합 FK가 다른 릴레이션 R2의 기본키인 것을 말한다.
→ 외래키와 기본키가 정의된 도메인은 같아야 하고, 외래키는 NULL을 허용할 수도 있어야 한다. 또한 외래키는 후보키와 관련된 개념은 아니다.

5. 한 릴레이션의 기본키를 구성하는 어떠한 속성 값도 NULL 값이나 중복값을 가질 수 없다는 것을 의미하는 제약조건은?
① 개체 무결성 제약 조건

6. 릴레이션에서 특정 속성에 해당하는 열을 선택하는 데 사용하며, 릴레이션의 수직적 부분집합을 반환하는 관계대수 연산자는?
① projection

7. 릴레이션 C가 릴레이션 A와 B를 자연조인한 결과일 때 다음 중 맞는 설명을 모두 고르시오.
② C의 카디날리티는 A의 카디날리티보다 적다.
③ C의 차수는 A의 차수보다 많다.

8. 다음 용어를 설명하시오.
(1) **릴레이션**: 관계 데이터 모델에서 사용하는 2차원 테이블 형태의 데이터 구조이다.
(2) **스키마**: 구조라는 뜻으로 '데이터베이스 스키마', '릴레이션 스키마'로 자주 사용된다. 데이터베이스 스키마는 전체적인 데이터베이스 구조를 뜻하며 데이터베이스의 모든 가능한 상태를 미리 정의한다. 릴레이션 스키마는 릴레이션이 포함하는 에트리뷰트의 이름들을 담고 있다.
(3) **릴레이션 인스턴스**: 릴레이션에 어느 시점에 들어있는 투플들의 집단을 일컫는 말로 릴레이션 인스턴스는 정적이지 않고 데이터 조작연산에 따라 시시각각 변한다.
(4) **릴레이션 차수와 카디날리티**: 한 릴레이션에 들어있는 속성의 수를 차수라 하고, 한 릴레이션에 들어있는 투플의 수를 카디날리티라 한다.
(5) **도메인**: 한 속성에 나타날 수 있는 값들의 집합으로 프로그래밍 언어의 데이터 타입과 유

사하다.

(6) **투플**: 릴레이션에 있어서 행 부분으로, 릴레이션이 나타내는 엔티티의 한 인스턴스를 의미한다.

9. 릴레이션에 대한 다음 물음에 답하시오.

(1) 릴레이션 스키마와 릴레이션 인스턴스의 차이점을 설명하시오.

릴레이션은 관계 데이터 모델에서 사용하는 2차원 테이블 형태의 데이터 구조이고 릴레이션 스키마와 릴레이션 인스턴스로 구성되어 있다. 릴레이션 스키마는 열 단위의 릴레이션 속성들을 담고 있고, 릴레이션 인스턴스는 행 단위의 릴레이션 스키마에 실제로 저장된 데이터의 집합을 의미한다.

(2) 도메인 제약조건을 설명하시오.

도메인 제약조건은 릴레이션 내의 투플들은 릴레이션 스키마 선언이 정의된 각 속성의 도메인에 지정된 범위 내의 값만을 가져야 한다는 무결성 제약조건이다.

(3) 기본키 제약조건과 외래키 제약조건을 설명하시오.

기본키 제약조건은 기본키는 NULL값을 가져서는 안되며, 릴레이션 내에 오직 하나만 존재할 수 있다는 제약조건이고, 참조 무결성 제약조건은 릴레이션 간 참조관계를 선언하는 제약조건으로 자식 릴레이션에 외래키로 선언된 속성의 도메인 제약은 부모 릴레이션의 속성과동일하게 적용되며 자식 릴레이션의 값 변경 시 부모 릴레이션의 값을 참조해야한다는 제약조건이다.

(4) 참조 무결성 제약조건의 옵션 네 가지를 설명하시오.

	부모 릴레이션의 투플 (참조 받는 릴레이션)	**자식 릴레이션의 투플 (참조하는 릴레이션)**
삽입	(제약 없음)	**규칙 1**
삭제	**규칙 2**	(제약 없음)
수정	삭제 후 삽입과 같음	삭제 후 삽입과 같음

- **규칙 1 : 삽입(자식 릴레이션 투플 삽입)**

  부모, 자식 관계를 맺고 있는 두 릴레이션 중 자식 릴레이션에 투플을 삽입하는 경우 그 투플에 부모 릴레이션의 기본키가 존재하는 지 확인하고 없는 경우 삽입은 거부된다. 이때 투플에 기본키를 넣도록 수정하거나, 부모 릴레이션의 기본키 옵션에 NULL값을 허용하면 삽입이 가능하다.

- **규칙 2 : 삭제(부모 릴레이션 투플 삭제)**

  부모 릴레이션에서 투플을 삭제하는 경우 자식 릴레이션에서 참조하고 있는 값이 있으므로 DBMS는 응용 프로그래머에게 다음 같은 4가지 옵션 중 하나를 선택할 수 있도록 하며 선택 사항은 부모 릴레이션 생성(CRAETE) 문에서 선언한다.

명령어	의미
RESTRICTED (NO ACTION)	삭제작업 거부
CADSECADE	자식 릴레이션의 투플도 함께 삭제
DEFAULT	자식 릴레이션의 관련 투플을 미리 설정해둔 값으로 변경
NULL	자식 릴레이션의 관련 투플 값을 NULL로 설정

- **수정**
  수정은 삭제와 삽입명령이 연속해서 수행되는 것인데, 위에서 언급한 것과 마찬가지로 부모 릴레이션의 수정이 일어날 경우 삭제 방법에 따라 처리된 후, 삽입제약조건 확인에 따라 처리된다.

(5) 후보키와 기본키의 차이점을 설명하시오.
후보키와 기본키 모두 푸틀을 식별할 수 있는 최소 집합의 키이지만 기본키는 후보키보다 더 좁은 개념으로 후보키 중 선정된 하나를 의미한다.

10. 사원(주민등록번호, 사원번호, 사원이름, 주소, 생년월일) 릴레이션이 있다. 기본키는 (사원이름, 생년월일)이고, 그 밖의 대체키 1은 주민등록번호, 대체키 2는 사원번호이다. 다음 물음에 답하시오.

(1) (주민등록번호, 주소)는 후보키인가? 그 이유는 무엇인가?
후보키가 아니다. 주소 속성은 필요하지 않은 속성이다.
(2) 사원번호는 수퍼키인가? 그 이유는 무엇인가?
수퍼키이다. 투플을 유일하게 구분한다.
(3) 생년월일은 NULL 값을 가질 수 있는가?
가질 수 있다.
(4) 주소는 NULL 값을 가질 수 있는가?
가질 수 있다.

11. 다음 릴레이션에서 더 이상 삽입되는 데이터가 없다고 가정한다. 다음 물음에 답하시오.

(1) 릴레이션 R과 S의 후보키를 모두 보이시오.
- 릴레이션 R의 후보키: A
- 릴레이션 S의 후보키: (C,D) 혹은 (C,E)

(2) 릴레이션 R과 S의 기본키는 어떤 것이 좋을지 선택하시오.
- 릴레이션 R의 기본키: A
- 릴레이션 S의 기본키: (C,D) 혹은 (C,E) 중 하나를 선택

12. 다음 릴레이션에서 관계대수식의 결과를 작성하시오.

(1) $\sigma_{A=a2}(R)$

A	B	C
a2	b1	c1

(2) $\pi_{A,B}(R)$

A	B
a1	b1
a2	b1
a3	b1
a4	b2

(3) $R \bowtie_{R.c=S.c} S$

A	B	C	C	D	E
a1	b1	c1	c1	d2	e1
a1	b1	c1	c1	d1	e2
a2	b1	c1	c1	d2	e1
a2	b1	c1	c1	d1	e2
a3	b1	c2	c2	d3	e3
d4	b2	c3	c3	d3	e3

13. 다음 수강신청 관련 릴레이션에 대한 질의문을 관계대수식으로 표현하시오.

학생(학번, 이름, 전공, 학년)
수강(과목코드, 학번, 수강학기, 성적)
과목(과목코드, 과목이름, 강의실, 요일, 담당교수)

(1) 과목코드가 1234이고 성적이 A인 모든 학생의 학번을 보이시오.

$\pi_{학번}(\sigma_{과목코드=1234\ AND\ 성적='A'}(학생))$

(2) 과목코드가 1234인 과목을 등록한 학생의 이름과 전공을 보이시오.

$\pi_{이름,전공}(\sigma_{과목코드=1234}(학생 \bowtie_{학생.과목코드=수강.과목코드} 수강))$

(3) 모든 과목에 등록한 학생의 이름을 보이시오.

$\pi_{이름,과목코드}(학생 \bowtie_{학생.과목코드=수강.과목코드} 수강) \div \pi_{과목코드}(수강)$

(4) 과목 1234에 등록하지 않은 학생의 이름을 보이시오.

$\pi_{이름}(\sigma_{과목코드 \neq 1234}(학생 \bowtie_{학생.과목코드=수강.과목코드} 수강))$

14. [극장 데이터베이스] 다음은 4개의 지점을 가진 극장 데이터베이스다. 밑줄 친 속성은 기본키이다.

극장(극장번호, 극장이름, 위치)
상영관(극장번호, 상영관번호, 영화제목, 가격, 좌석수)
예약(극장번호, 상영관번호, 고객번호, 좌석번호, 날짜)
고객(고객번호, 이름, 주소)

(1) 각 테이블에서 외래키를 찾아보시오.

상영관(극장번호), 예약(극장번호, 상영관번호, 고객번호)

(2) 각 테이블에 저장될 데이터를 세 개씩 적어보시오. 예를 들면 극장의 경우는 다음과 같다.

극장번호	극장이름	위치
1	대한	강남
2	씨티	강남
3	씨티	잠실

△ 극장 테이블

극장번호	상영관번호	영화제목	가격	좌석수
1	1	신세계	9,000	35
1	2	영웅호걸	9,000	40
3	1	러브레터	9,000	10

△ 상영관 테이블

극장번호	상영관번호	고객번호	좌석번호	날짜
1	1	1	30	2013-10-15
2	1	2	25	2013-10-15
3	1	1	25	2013-10-17

△ 예약 테이블

고객번호	이름	주소
1	장내윤	서울시 강동구
2	홍길동	서울시 도봉구
3	김유신	서울시 강남구

△ 고객 테이블

(3) 다음 관계대수식이 나타내는 릴레이션은 무엇인지 설명하시오.

① $\pi_{\text{극장번호}}(\sigma_{\text{가격}>6000}(\text{상영관}))$

영화 가격이 6,000원 이상인 상영관의 극장번호

② $\sigma_{\text{극장.극장번호=상영관.극장번호}}(\text{극장}\times\text{상영관})$

극장별 상영관(두 테이블 조인)

③ $\pi_{\text{극장이름}}(\text{극장}\bowtie_{\text{극장.극장번호=상영관.극장번호}}(\sigma_{\text{가격}>6000}(\text{상영관}))$

영화 가격이 6,000원 이상인 영화를 상영하는 극장이름

④ 고객 ⋈ ($\sigma_{\text{날짜} > '20140101'}$ (예약))

영화 예약 날짜가 2014년 1월 1일 이후인 고객의 정보와 예약내용(단, 예약이 없는 고객도 포함)

⑤ $\pi_{\text{고객이름, 극장번호}}$ (예약⋈$_{\text{예약.고객번호=고객.고객번호}}$ 고객) ÷ $\pi_{\text{극장번호}}$ ($\sigma_{\text{위치='강남'}}$ (극장))

강남에 위치한 극장을 모두 예약한 고객의 이름

(4) 다음 물음에 대하여 관계대수식을 작성하시오.

① 모든 극장의 이름과 위치를 보이시오.

$\pi_{\text{극장이름, 위치}}$ (극장)

② 가격이 7,000원 이하인 영화제목을 보이시오.

$\pi_{\text{영화제목}}$ ($\sigma_{\text{가격<=7000}}$ (상영관))

③ 모든 고객의 이름과 주소를 보이시오.

$\pi_{\text{이름, 주소}}$ (고객)

④ '강남'에 위치한 극장에서 상영 중인 영화제목을 보이시오.

$\pi_{\text{영화제목}}$ (($\sigma_{\text{위치='강남'}}$ 극장) ⋈ $_{\text{극장.극장번호=상영관.극장번호}}$ 상영관)

⑤ '강남'에 위치한 극장에 예약을 한 고객의 이름을 보이시오.

$\pi_{\text{고객이름}}$ ((($\sigma_{\text{위치=강남}}$ (극장)) ⋈$_{\text{극장.극장번호=예약.극장번호}}$ 예약) ⋈$_{\text{예약.고객번호=고객.고객번호}}$ 고객)

15. [판매원 데이터베이스] 다음 릴레이션을 보고 물음에 답하시오. Salesperson은 판매원, Order는 주문, Customer는 고객을 나타낸다. 밑줄 친 속성은 기본키이고 custname과 salesperson은 각각 Customer.name과 Salesperson.name을 참조하는 외래키이다.

Salesperson(<u>name</u>, age, salary) Order(<u>number</u>, custname, salesperson, amount) Customer(<u>name</u>, city, industrytype)

(1) 모든 판매원(Salesperson)의 이름을 보이시오.

π_{name} (Salesperson)

(2) 고객 '홍길동'의 주문을 수주한 판매원의 이름을 보이시오.

$\pi_{\text{salesperson}}$ ($\sigma_{\text{custname='홍길동'}}$ (Order))

(3) 주문이 있는 판매원의 이름을 보이시오.

$\pi_{\text{salesperson}}$ (Order)

(4) 주문이 없는 판매원의 이름을 보이시오.

π_{name} (Salesperson) - $\pi_{\text{salesperson}}$ (Order)

(5) 고객 '홍길동'의 주문을 수주한 판매원의 나이를 보이시오.

π_{age} (Salesperson $\bowtie_{Salesperson.name=Order.salesperson}$ ($\sigma_{custname='홍길동'}$ (Order)))

(6) 나이가 25살인 판매원에게 주문한 고객의 city 값을 보이시오.

π_{city} ((($\sigma_{age=25}$ Salesperson) $\bowtie$ Order) $\bowtie$ Customer)

(7) 판매원의 이름과 그 판매원에게 주문을 한 고객의 이름을 보이시오. 단 주문이 없는 판매원도 포함하여 구한다.

$\pi_{salesperson,\ custname}$ (Salesperson $\bowtie$ Order)

16. [기업 프로젝트 데이터베이스] 다음 릴레이션을 보고 물음에 답하시오. Employee는 사원, Department는 부서, Project는 프로젝트 내용, Works는 사원이 프로젝트에 참여한 내용을 나타낸다. 한 사원이 여러 프로젝트에서 일할 수 있고, 한 프로젝트에서 여러 사원이 일할 수 있다. hours-worked 속성은 각 사원이 각 프로젝트에서 일한 시간 수를 나타낸다.

Employee(empno, name, phoneno, address, sex, position, deptno) Department(deptno, deptname, manager) Project(projno, projname, deptno) Works(empno, projno, hours-worked)

(1) 각 릴레이션에서 기본키를 정하시오.

- Employee: empno
- Department: deptno
- Project: projno
- Work: empno, projno

(2) 릴레이션 간의 관계를 살펴보고 외래키를 찾아보시오.

- Employee : deptno
- Department : manager
- Project : deptno
- Works : (empno, projno)

(3) 다음 질문에 대하여 관계대수식을 작성하시오.

① 모든 직원의 이름을 보이시오.

π_{name} (Employee)

② 여자 직원의 이름을 보이시오.

π_{name} ($\sigma_{sex=female}$ (Employee))

③ 팀장(manager)의 이름과 주소를 보이시오.

$\pi_{name,\ address}$ (Employee $\bowtie$ Department)

④ IT 부서(Department)에서 일하는 직원의 이름과 주소를 보이시오.

$\pi_{name,address}$ ($\sigma_{deptname='IT부서'}$ (Employee ⋈ Department))

⑤ '미래' 프로젝트에서 일하는 직원의 이름을 보이시오.

π_{name} ($\sigma_{projname='미래'}$ (Project ⋈ Works ⋈ Employee))

⑥ '김철수' 팀장(manager) 부서에서 일하는 직원의 이름을 보이시오.

- 김철수 팀장의 부서 x : π_{deptno} ($\sigma_{name=김철수\ AND\ Department.manager=Employee.empno}$(Employee ⋈ Department))
- x 부서에서 일하는 직원 : π_{name} ($\sigma_{deptno=x}$ (Employee))

→ rename 연산자를 사용하여 다음과 같이 해결

π_{name} (Employee ⋈ $_{deptno=deptno2}$ ($\rho_{deptno/deptno2}$ (π_{deptno}

($\sigma_{name=김철수\ AND\ Department.manager=Employee.empno}$ (Employee ⋈ Department)))))

3장. SQL 기초

1. 마당서점의 고객이 요구하는 다음 질문에 대해 SQL 문을 작성하시오.

(1) 도서번호가 1인 도서의 이름

```
SELECT bookname FROM Book WHERE bookid=1;
```

(2) 가격이 20,000원 이상인 도서의 이름

```
SELECT bookname FROM Book WHERE price >= 20000;
```

(3) 박지성의 총 구매액

```
SELECT SUM(saleprice)
FROM Customer, Orders
WHERE Customer.custid=Orders.custid
        AND Customer.name LIKE '박지성';
```

(4) 박지성이 구매한 도서의 수

```
SELECT COUNT(*) FROM Customer, Orders
WHERE Customer.custid=Orders.custid
        AND Customer.name LIKE '박지성';
```

(5) 박지성이 구매한 도서의 출판사 수

```
SELECT COUNT(DISTINCT publisher)
FROM Customer, Orders, Book
WHERE Customer.custid=Orders.custid AND Orders.bookid=Book.bookid
         AND Customer.name LIKE '박지성';
```

(6) 박지성이 구매한 도서의 이름, 가격, 정가와 판매가격의 차이

```
SELECT bookname, price, price-saleprice
FROM Customer, Orders, Book
WHERE Customer.custid=Orders.custid AND Orders.bookid=Book.bookid
        AND Customer.name LIKE '박지성';
```

(7) 박지성이 구매하지 않은 도서의 이름

```
SELECT bookname FROM Book b1
WHERE NOT EXISTS
    (SELECT bookname FROM Customer, Orders
     WHERE Customer.custid=Orders.custid AND Orders.bookid=b1.bookid
             AND Customer.name LIKE '박지성');
```

2. 마당서점의 운영자와 경영자가 요구하는 다음 질문에 대해 SQL 문을 작성하시오.

(1) 마당서점 도서의 총 개수

```
SELECT count(*) FROM Book;
```

(2) 마당서점에 도서를 출고하는 출판사의 총 개수

```
SELECT COUNT(Distinct publisher)
FROM Book;
```

(3) 모든 고객의 이름, 주소

```
SELECT name, address
FROM Customer;
```

(4) 2014년 7월 4일~7월 7일 사이에 주문받은 도서의 주문번호

```
SELECT *
FROM Orders
WHERE orderdate BETWEEN '20140704' AND '20140707';
```

(5) 2014년 7월 4일~7월 7일 사이에 주문받은 도서를 제외한 도서의 주문번호

```
SELECT *
FROM Orders
WHERE orderdate NOT BETWEEN '20140704' AND '20140707';
```

(6) 성이 '김' 씨인 고객의 이름과 주소

```
SELECT name, address
FROM Customer
WHERE name LIKE '김%';
```

(7) 성이 '김' 씨이고 이름이 '아'로 끝나는 고객의 이름과 주소

```
SELECT name, address
FROM Customer
WHERE name LIKE '김%아';
```

(8) 주문하지 않은 고객의 이름(부속질의 사용)

```
SELECT name FROM Customer
WHERE name NOT IN
        (SELECT name
        FROM Orders, Customer
        WHERE Orders.custid=Customer.custid);
```

(9) 주문 금액의 총액과 주문의 평균 금액

```
SELECT SUM(saleprice), AVG(saleprice)
FROM Orders;
```

(10) 고객의 이름과 고객별 구매액

```
SELECT name, SUM(saleprice)
FROM  Orders, Customer
WHERE Orders.custid=Customer.custid
GROUP BY name;
```

(11) 고객의 이름과 고객이 구매한 도서 목록

```
SELECT name, bookname
FROM Book, Orders, Customer
WHERE Book.bookid=Orders.bookid
        AND Orders.custid=Customer.custid;
```

(12) 도서의 가격(Book 테이블)과 판매가격(Orders 테이블)의 차이가 가장 많은 주문

```
SELECT *
FROM Book, Orders
WHERE Book.bookid=Orders.bookid
      AND price-saleprice=
        (SELECT MAX(price-saleprice)
         FROM Book, Orders
         WHERE Book.bookid=Orders.bookid);
```

(13) 도서의 판매액 평균보다 자신의 구매액 평균이 더 높은 고객의 이름

```
SELECT name, AVG(saleprice)
FROM Customer, Orders
WHERE Customer.custid=Orders.custid
GROUP BY name
HAVING AVG(saleprice) >
          (SELECT AVG(saleprice) FROM Orders);
```

3. 마당서점에서 다음의 심화된 질문에 대해 SQL 문을 작성하시오.

(1) 박지성이 구매한 도서의 출판사와 같은 출판사에서 도서를 구매한 고객의 이름

```
SELECT name FROM Customer, Orders, Book
WHERE Customer.custid=Orders.custid
      AND Orders.bookid=Book.bookid AND name NOT LIKE '박지성'
        AND publisher IN
        (SELECT publisher FROM Customer, Orders, Book
```

```
WHERE Customer.custid=Orders.custid
      AND Orders.bookid=Book.bookid
      AND name LIKE '박지성');
```

(2) 두 개 이상의 서로 다른 출판사에서 도서를 구매한 고객의 이름

```
SELECT name FROM Customer c1
WHERE 2 >=
      (SELECT COUNT(DISTINCT publisher) FROM Customer, Orders, Book
       WHERE Customer.custid=Orders.custid
          AND Orders.bookid=Book.bookid AND (name LIKE c1.name));
```

(3) 전체 고객의 30% 이상이 구매한 도서

```
SELECT bookname FROM Book b1
WHERE  ( (SELECT COUNT(Book.bookid) FROM Book, Orders
          WHERE Book.bookid=Orders.bookid AND Book.bookid=b1.bookid)
           >= 0.3 * (SELECT COUNT(*) FROM Customer));
```

4. 다음 질의에 대해 DML 문을 작성하시오.

(1) 새로운 도서 (‘스포츠 세계’, ‘대한미디어’, 10000원)이 마당서점에 입고되었다. 삽입이 안 될 경우 필요한 데이터가 더 있는지 찾아보시오.

```
INSERT INTO BOOK VALUES(11, '스포츠 세계', '대한미디어', 10000);
```

(2) ‘삼성당’에서 출판한 도서를 삭제하시오.

```
DELETE FROM Book WHERE publisher LIKE '삼성당';
```

(3) ‘이상미디어’에서 출판한 도서를 삭제하시오. 삭제가 안 될 경우 원인을 생각해보시오.

오류 : DELETE 문이 REFERENCE 제약조건과 충돌했습니다.

외래키 제약조건에 위배된다.

(4) 출판사 ‘대한미디어’를 ‘대한출판사’로 이름을 바꾸시오.

```
UPDATE Book SET publisher='대한출판사' WHERE publisher LIKE '대한미디어';
```

(5) (테이블 생성) 출판사에 대한 정보를 저장하는 테이블 Bookcompany(name, address, begin)를 생성하고자 한다. name은 기본키며 VARCHAR(20), address는 VARCHAR(20), begin은 DATE 타입으로 선언하여 생성하시오.

```
CREATE TABLE Bookcompany (
 name          VARCHAR(20) PRIMARY KEY,
 address       VARCHAR(20),
 begin         DATE );
```

(6) (테이블 수정) Bookcompany 테이블에 인터넷 주소를 저장하는 webaddress 속성을 VARCHAR(30)으로 추가하시오.

```
ALTER TABLE Bookcompany ADD webaddress VARCHAR(30);
```

(7) Bookcompany 테이블에 임의의 투플 name=한빛아카데미, address=서울시 마포구, begin=1993-01-01, webaddress=http://hanbit.co.kr를 삽입하시오.

```
INSERT INTO Bookcompany
    VALUES ('한빛아카데미', '서울시 마포구', 1993-01-01, 'http://hanbit.co.kr');
```

5. 다음 EXISTS 질의의 결과를 보이시오.

(1) 질의의 결과는 무엇인가?

주문이 없는 고객

(2) NOT을 지우면 질의의 결과는 무엇인가?

주문이 있는 고객

6. [극장 데이터베이스] 다음은 4개의 지점을 가진 극장 데이터베이스다. 밑줄 친 속성은 기본키이다. 테이블의 구조를 만들고 데이터를 입력한 후 다음 질의에 대한 SQL 문을 작성하시오. 테이블의 구조를 만들 때 다음 제약조건을 반영하여 작성한다.

(1) 단순 질의

① 모든 극장의 이름과 위치를 보이시오.

```
SELECT 극장이름, 위치
FROM 극장;
```

② '잠실'에 있는 극장을 보이시오.

```
SELECT * FROM 극장
WHERE 위치 LIKE '잠실';
```

③ '잠실'에 사는 고객의 이름을 오름차순으로 보이시오.

```
SELECT 고객번호, 이름, 주소
FROM 고객
WHERE 주소 LIKE '잠실'
ORDER BY 이름;
```

④ 가격이 8,000원 이하인 영화의 극장번호, 상영관번호, 영화제목을 보이시오.

```
SELECT 극장번호, 상영관번호, 영화제목
FROM 상영관
WHERE 가격 <=8000;
```

⑤ 극장 위치와 고객의 주소가 같은 고객들을 보이시오.

```
SELECT 고객.이름, 극장.위치
FROM 고객, 극장
WHERE 고객.주소 LIKE 극장.위치;
```

(2) 집계질의

① 극장의 수는 몇 개인가?

```
SELECT COUNT(극장번호)
FROM 극장;
```

② 상영되는 영화의 평균 가격은 얼마인가?

```
SELECT AVG(가격)
FROM 상영관;
```

③ 2014년 9월 1일에 영화를 관람한 고객의 수는 얼마인가?

```
SELECT COUNT(이름)
FROM 고객, 예약
WHERE 예약.고객번호=고객.고객번호 AND 날짜 LIKE '2014-09-01';
```

(3) 부속질의와 조인

① '대한' 극장에서 상영된 영화제목을 보이시오.

```
SELECT 영화제목
FROM 극장, 상영관
WHERE 극장.극장번호=상영관.극장번호
        AND 극장이름 LIKE '대한';
```

② '대한' 극장에서 영화를 본 고객의 이름을 보이시오.

```
SELECT 고객.이름
FROM 극장, 예약, 고객
WHERE 극장.극장번호=예약.극장번호 AND 예약.고객번호=고객.고객번호
        AND 극장이름 LIKE '대한';
```

③ '대한' 극장의 전체 수입을 보이시오.

```
SELECT SUM(가격)
FROM 극장, 상영관, 예약
WHERE 극장.극장번호=상영관.극장번호 AND
        상영관.극장번호=예약.극장번호 AND 상영간.상영관번호=예약.상영관번호;
```

(4) 그룹질의

① 극장별 상영관 수를 보이시오.

```
SELECT 극장번호, COUNT(*)
```

```
FROM 상영관
GROUP BY 극장번호;
```

② '잠실'에 있는 극장의 상영관을 보이시오.

```
SELECT * FROM 극장, 상영관
WHERE 극장.극장번호=상영관.극장번호 AND 위치 LIKE '잠실';
```

③ 201년 9월 1일에 극장별 평균 관람 고객의 수를 보이시오.

```
SELECT 극장번호, COUNT(*)
FROM 예약
WHERE 날짜 LIKE '2014-09-01'
GROUP BY극장번호;
```

④ 2014년 9월 1일에 가장 많은 고객이 관람한 영화를 보이시오.

```
SELECT 영화제목
FROM 상영관, 예약
WHERE 상영관.극장번호=예약.극장번호 AND 상영관.상영관번호=예약.상영관번호
        AND 날짜 LIKE '2014-09-01'
GROUP BY 예약.극장번호, 예약.상영관번호
HAVING COUNT(*) = ( SELECT MAX(*)
                    FROM 상영관, 예약
                    WHERE 상영관.극장번호=예약.극장번호
                            AND 상영관.상영관번호=예약.상영관번호
                            AND 날짜 LIKE '2014-09-01'
                    GROUP BY 예약.극장번호, 예약.상영관번호);
```

(5) DML

① 각 테이블에 데이터를 삽입하는 INSERT 문들을 하나씩 보이시오.

```
(생략)
```

② 영화의 가격을 10% 인상하시오.

```
UPDATE 상영관
SET 가격 = 가격 *1.1;
```

7. [판매원 데이터베이스] 다음 릴레이션을 보고 물음에 답하시오. Salesperson은 판매원, Order는 주문, Customer는 고객을 나타낸다. 밑줄 친 속성은 기본키이고 custname과 salesperson은 각각 Customer.name과 Salesperson.name을 참조하는 외래키이다.

(1) 테이블을 생성하는 CREATE 문과 데이터를 삽입하는 INSERT 문을 작성하시오.

```
CREATE TABLE Order (
  number        PRIMARY KEY,
  custname      CHAR(10),
  salesperson   CHAR(10),
  amount        NUMBER,
  FOREIGN KEY(custname) REFERENCES Customer(name),
  FOREIGN KEY(salesperson) REFERENCES Salesperson(name));
  .. (이하 생략)
```

(2) 모든 판매원의 이름과 급여를 보이시오. 단, 중복 행은 제거한다.

```
SELECT   (DISTINCT) name, salary
FROM     Salesperson;
```

(3) 나이가 30세 미만인 판매원의 이름을 보이시오.

```
SELECT   name
FROM     Salesperson
WHERE    age < 30;
```

(4) 'S'로 끝나는 도시에 사는 고객의 이름을 보이시오.

```
SELECT   name
FROM     Customer
WHERE    city LIKE '%S';
```

(5) 주문을 한 고객의 수(서로 다른 고객만)를 보이시오.

```
SELECT   COUNT(DISTINCT custname)
FROM     Order;
```

(6) 판매원 각각에 대하여 주문의 수를 계산하시오.

```
SELECT     salesperson, COUNT(*)
FROM       Order
GROUP BY salesperson;
```

(7) 'LA'에 사는 고객으로부터 주문을 받은 판매원의 이름과 나이를 보이시오(부속질의를 사용).

```
SELECT name, age
FROM   Salesperson
WHERE name IN
        (SELECT salesperson FROM Order WHERE custname IN
                (SELECT name FROM Customer WHERE city LIKE 'LA') );
```

(8) 'LA'에 사는 고객으로부터 주문을 받은 판매원의 이름과 나이를 보이시오(조인을 사용).

```
SELECT   salesperson, age
FROM     Salesperson, Order, Customer
WHERE    Salesperson.name=Order.salesperson
         AND Order.custname=Customer.name AND city='LA';
```

(9) 두 번 이상 주문을 받은 판매원의 이름을 보이시오.

```
SELECT    Salesperson
FROM      Order
GROUP BY Salesperson
HAVING COUNT(*) > 1;
```

(10) 판매원 'TOM'의 봉급을 45,000원으로 변경하는 SQL 문을 작성하시오.

```
UPDATE Salesperson
SET       salary=45000
WHERE   name LIKE 'TOM';
```

8. [기업 프로젝트 데이터베이스] 다음 릴레이션을 보고 물음에 답하시오. Employee는 사원, Department는 부서, Project는 프로젝트, Works는 사원이 프로젝트에 참여한 내용을 나타낸다. 한 사원이 여러 프로젝트에서 일할 수 있고, 한 프로젝트에 여러 사원이 일할 수 있다. hours-worked 속성은 각 사원이 각 프로젝트에서 일한 시간 수를 나타낸다. 밑줄 친 속성은 기본키이다.

(1) 테이블을 생성하는 CREATE 문과 데이터를 삽입하는 INSERT 문을 작성하시오. 테이블의 데이터 타입은 임의로 정하고, 데이터는 아래 질의의 결과가 나오도록 삽입한다.

※ 테스용 스크립트

```
CREATE TABLE Department
(
        deptno number not null,
        deptname varchar(20),
        manager varchar(20),
        primary key(deptno)
)

CREATE TABLE Employee
(
        empno number not null,
        name varchar(20),
        phoneno number,
        address varchar(20),
        sex varchar(20),
        position varchar(20),
        deptno number,
        primary key(empno),
        foreign key(deptno) references Department(deptno)
```

```
)

CREATE TABLE Project
(
        projno number not null,
        projname varchar(20),
        deptno number,
        primary key(projno),
        foreign key(deptno) references Department(deptno)
)

CREATE TABLE Works
(
        projno number not null,
        empno number not null,
        hoursworked number,
        PRIMARY key(projno, empno),
        foreign key(projno) references Project(projno),
        foreign key(empno) references Emplyee(empno)
)

insert into Department values(1,'IT', '고남순')
insert into Department values(2,'Marketing', '홍길동')

insert into Emplyee values(1, '김덕성', 01012341232, '서울', '여', 'Programmer',1)
insert into Emplyee values(2, '이서울', 01012323122, '서울', '남', 'Programmer',1)
insert into Emplyee values(3, '박연세', 01076851231, '대전', '여', 'Salesperson',2)
insert into Emplyee values(4, '홍길동', 01012341546, '서울', '남', 'Manager',2)
insert into Emplyee values(5, '고남순', 01012311112, '서울', '여', 'Manager',1)

insert into Project values(1,'데이터베이스구축',1)
insert into Project values(2,'시장조사',2)

insert into Works values(1, 1, 3);
insert into Works values(1, 2, 1);
insert into Works values(2, 3, 1);
insert into Works values(2, 4, 5);
insert into Works values(1, 5, 1);
```

(2) 모든 사원의 이름을 보이시오.

```
SELECT name
FROM   Employee;
```

(3) 여자 사원의 이름을 보이시오.

```
SELECT name
FROM Employee
WHERE sex LIKE '여';
```

(4) 팀장(manager)의 이름을 보이시오.

```
SELECT name
FROM Employee
WHERE empno IN (SELECT manager FROM Department);
```

(5) 'IT' 부서에서 일하는 사원의 이름과 주소를 보이시오.

```
SELECT name, address
FROM Employee, Department
WHERE Emplyee.deptno=Department.deptno and deptname LIKE 'IT';
```

(6) '홍길동' 팀장(manager) 부서에서 일하는 사원의 수를 보이시오.

```
SELECT count(name)    --홍길동을 포함해서
FROM Employee
WHERE deptno IN (SELECT deptno FROM Employee, Department
                        WHERE Employee.empno=Department.manager
                        AND name LIKE '홍길동';
```

(7) 사원들이 일한 시간 수를 부서별, 사원 이름별 오름차순으로 보이시오.

```
SELECT deptno, name, SUM(hours-worked)
FROM Employee, Works
WHERE Emplyee.empno=Works.empno
ORDER BY deptno, name;
```

(8) 두 명 이상의 사원이 참여한 프로젝트의 번호, 이름, 사원의 수를 보이시오.

```
SELECT Project.projno, Project.projname, count(name) 사원수
FROM Employee, Project
WHERE Project.deptno=Employee.deptno
GROUP BY Project.projno, Project.projname
HAVING COUNT(*) >= 2;
```

(9) 세 명 이상의 사원이 있는 부서의 사원 이름을 보이시오.

```
SELECT name
FROM Employee, Department
WHERE Employee.deptno=Department.deptno and
        deptname = (SELECT deptname FROM Employee, Department
                WHERE Employee.deptno=Department.deptno
                GROUP BY deptname
                HAVING COUNT(name) >= 3);
```

9. [사원 데이터베이스] 다음은 scott 데이터베이스에 저장된 사원 데이터베이스다. 다음 질문에 대해 SQL 문을 작성하시오(부록 B의 scott 계정을 준비하였다면 예제 데이터베이스가 생성되어 있다).

Dept는 부서 테이블로 deptno(부서번호), dname(부서이름), loc(위치, location)으로 구성되어 있다. Emp는 사원 테이블로 empno(사원번호), ename(사원이름), job(업무), MGR(팀장번호, manager), hiredate(고용날짜), sal(급여, salary), comm(커미션금액, commission), deptno(부서번호)로 구성되어 있다. 밑줄 친 속성은 기본키이고 Emp의 deptno는 Dept의 deptno를 참조하는 외래키이다.

```
Dept(deptno NUMBER(2), dname VARCHAR2(14), loc VARCHAR2(13))
Emp(empno NUMBER(4), ename VARCHAR2(10), job VARCHAR2(9), mgr NUMBER(4),
    hiredate DATE, sal NUMBER(7,2), comm NUMBER(7,2), deptno NUMBER(2))
```

(1) 사원의 이름과 직위를 출력하시오. 단, 사원의 이름은 '사원이름', 직위는 '사원직위'머리글이 나오도록 출력한다.

```
SELECT ename AS '사원이름', job AS '사원직위'
FROM Emp;
```

(2) 30번 부서에 근무하는 모든 사원의 이름과 급여를 출력하시오.

```
SELECT ename, sal
FROM   Emp
WHERE deptno=30;
```

(3) 사원 번호와 이름, 현재 급여와 10% 인상된 급여(열 이름은 '인상된 급여')를 출력하시오. 단, 사원 번호순으로 출력한다. 증가된 급여분에 대한 열 이름은 '증가액'으로 한다.

```
SELECT empno, ename, sal, sal*0.1 '증가액', sal*1.1 as "인상된 급여"
FROM Emp
ORDER BY empno;
```

(4) 'S'로 시작하는 모든 사원과 부서번호를 출력하시오.

```
SELECT ename, deptno
FROM   Emp
WHERE ename LIKE 's%';
```

(5) 모든 사원의 최대 및 최소 급여, 합계 및 평균 급여를 출력하시오. 열 이름은 각각 MAX, MIN, SUM, AVG로 한다. 단, 소수점 이하는 반올림하여 정수로 출력한다.

```
SELECT MAX(sal) AS MAX, MIN(sal) AS MIN, sum(sal) AS SUM,
                ROUND(AVG(sal),0) AS AVG
FROM Emp;
```

(6) 업무이름과 업무별로 동일한 업무를 하는 사원의 수를 출력하시오. 열 이름은 각각 '업무'와 '업무별 사원수'로 한다.

```
SELECT job 업무, COUNT(ename) "업무별 사원수"
FROM Emp
GROUP BY job;
```

(7) 사원의 최대 급여와 최소 급여의 차액을 출력하시오.

```
SELECT MAX(sal)-MIN(sal)
FROM Emp;
```

(8) 30번 부서의 구성원 수와 사원들 급여의 합계와 평균을 출력하시오.

```
SELECT COUNT(empno), SUM(sal), AVG(sal)
FROM Emp
WHERE deptno=30;
```

(9) 평균급여가 가장 높은 부서의 번호를 출력하시오.

```
SELECT AVG(sal), Dept.deptno
FROM Emp, Dept
WHERE Emp.deptno=Dept.deptno;
GROUP BY Dept.deptno
HAVING AVG(sal) >= ALL (SELECT (AVG(sal))
                        FROM Emp GROUP BY deptno);
```

(10) 세일즈맨을 제외하고, 각 업무별 사원들의 총 급여가 3000 이상인 각 업무에 대해서, 업무명과 각 업무별 평균 급여를 출력하되, 평균급여의 내림차순으로 출력하시오.

```
SELECT job, AVG(sal)
FROM emp
WHERE job NOT LIKE 'SALESMAN'
GROUP BY job
HAVING AVG(sal) >= 3000
ORDER BY AVG(sal) DESC;
```

(11) 전체 사원 가운데 직속상관이 있는 사원의 수를 출력하시오.

```
SELECT COUNT(empno)
FROM Emp
WHERE mgr IS NOT NULL;
```

(12) Emp 테이블에서 이름, 급여, 커미션 금액, 총액(sal + comm)을 구하여 총액이 많은 순서대로 출력하시오. 단, 커미션이 NULL인 사람은 제외한다.

```
SELECT ename, sal, comm, sal+comm
FROM Emp
EXCEPT
        SELECT ename, sal, comm, sal+comm
        FROM Emp
        WHERE comm IS NULL
ORDER BY sal+comm;
```

(13) 각 부서별로 같은 업무를 하는 사람의 인원수를 구하여 부서번호, 업무명, 인원수를 출력하시오.

```
SELECT job, COUNT(empno) 인원수, deptno
FROM emp
WHERE deptno IN (SELECT deptno FROM Emp)
GROUP BY job, deptno;
```

(14) 사원이 한 명도 없는 부서의 이름을 출력하시오.

```
SELECT Dept.deptno
FROM Dept
EXCEPT
        SELECT Dept. deptno
        FROM Dept, Emp
        WHERE Dept.deptno=Emp.deptno;
```

(15) 같은 업무를 하는 사람의 수가 4명 이상인 업무와 인원수를 출력하시오.

```
SELECT job, COUNT(empno)
FROM Emp
GROUP BY job
HAVING COUNT(empno) >= 4;
```

(16) 사원번호가 7400 이상 7600 이하인 사원의 이름을 출력하시오.

```
SELECT ename
FROM Emp
WHERE empno BETWEEN 7400 AND 7600;
```

(17) 사원의 이름과 사원의 부서를 출력하시오.

```
SELECT ename, dname, Dept.deptno
FROM Emp, Dept
WHERE Emp.deptno=Dept.deptno;
```

(18) 사원의 이름과 팀장의 이름을 출력하시오.

```
SELECT e1.ename, e2.ename
FROM Emp e1, Emp e2
WHERE e1.empno=e2.mgr;
```

(19) 사원 SCOTT보다 급여를 많이 받는 사람의 이름을 출력하시오.

```
SELECT ename
FROM Emp
WHERE sal > (SELECT sal FROM Emp WHERE ename LIKE 'SCOTT');
```

(20) 사원 SCOTT가 일하는 부서번호 혹은 DALLAS에 있는 부서번호를 출력하시오.

```
SELECT Dept.deptno
FROM Emp, Dept
WHERE Emp.deptno=Dept.deptno AND
        ename LIKE 'SCOTT' OR loc LIKE 'DALLAS';
```

10. [인사부서 데이터베이스] 다음은 어느 기업의 인사부서 데이터베이스다. 편의상 데이터베이스 이름을 'hr'이라고 부른다. 인사과(human resource, hr) 데이터베이스의 스키마는 아래와 같다. 다음 질문에 대해 SQL 문을 작성하시오.

(생략)

4장. SQL 고급

1. 다음 내장 함수의 결과를 적으시오.

(생략)

2. Mybook 테이블을 생성하고 NULL에 관한 다음 SQL 문에 답하시오. 질의의 결과를 보면서 NULL에 대한 개념도 정리해보시오.

(1) SELECT *
FROM Mybook;
→ 테이블 전체를 불러올 때 NULL은 결과에서 NULL로 명시된다.

(2) SELECT bookid, NVL(price, 0)
FROM Mybook;
→ NULL 값을 ISNULL() 함수를 통해 0이라는 값으로 대체했다.

(3) SELECT *
FROM Mybook
WHERE price IS NULL;
→ NULL 값을 비교할 EO는 =기호 대신 IS NULL 명령어를 통해 NULL 값을 가지는 투플을 검색한다.

(4) SELECT *
FROM Mybook
WHERE price ='';
→ NULL 값은 공백이 아니라 아직 입력하지 않은 값을 지칭하므로 ''로 검색했을 때 값이 나오지 않는다.

(5) SELECT bookid, price+100
FROM Mybook;
→ NULL에 대한 연산의 결과는 NULL로 나온다.

(6) SELECT SUM(price), AVG(price), COUNT(*)
FROM Mybook
WHERE bookid >= 4 ;
→ SUM(), AVG()는 WHERE 절의 조건에 따라 값이 없으므로 연산 결과 NULL 값을 결과로 갖고, COUNT(*)는 NULL 값을 1로 계산한다.

(7) SELECT COUNT(*), COUNT(price)
FROM Mybook;
→ COUNT(*)와 달리 COUNT(price)는 NULL을 인정하지 않으므로 NULL 투플을 제외한 개수를 셈한다.

(8) SELECT SUM(price), AVG(price)
FROM Mybook;
→ SUM(), AVG()은 NULL 값을 빼고 연산한다.

3. ROWNUM에 관한 다음 SQL 문에 답하시오. 데이터베이스는 마당서점 데이터베이스를 이용한다.

(1) SELECT *
FROM Book;
→ 투플 10개

(2) SELECT *
FROM Book
WHERE ROWNUM <= 5;
→ 투플 처음부터 5개

(3) SELECT *
FROM Book
WHERE ROWNUM <= 5
ORDER BY price;
→ 투플 처음부터 5개를 선택한 후 정렬

(4) SELECT *
FROM (SELECT * FROM Book) b
WHERE ROWNUM <= 5
ORDER BY price;
→ 투플 처음부터 5개를 선택한 후 정렬

(문제수정 예정)
SELECT *
FROM (SELECT * FROM Book ORDER BY price) b
WHERE ROWNUM <= 5;
→ 정렬한 후 처음부터 5개를 선택

(5) SELECT *
FROM (SELECT * FROM Book WHERE ROWNUM <=5) b
ORDER BY price;
→ 투플 처음부터 5개를 선택한 후 정렬

(6) SELECT *
FROM (SELECT * FROM Book WHERE ROWNUM <= 5 ORDER BY price) b;
→ 투플 처음부터 5개를 선택한 후 정렬

4. 부속질의에 관한 다음 SQL 문에 답하시오. 데이터베이스는 Madang 데이터베이스를 이용한다. 부속질의는 SELECT, FROM, WHERE 절에 각각 포함되어 있다.

(1) SELECT custid, (SELECT address
FROM Customer cs
WHERE cs.custid = od.custid) "address", SUM(saleprice) "total"
FROM Orders od
GROUP BY od.custid;
상관중첩질의이므로 외부질의부터 실행된다. 따라서 외부의 FROM과 GROUP BY를 통해 custid 별 고객을 얻고 이 고객들에 대해 custid, 내부질의에서 cs에 주소가 있는 고객을 선별한다. → 주문이 있는 고객에 대하여 고객별로 custid, address, 총주문액을 구한다.

(2) SELECT cs.name, s
FROM (SELECT custid, avg(saleprice) s
FROM Orders GROUP BY custid) od, Customer cs
WHERE cs.custid = od.custid;
INLINE 질의를 통해 고객별 평균 구매가격을 구하였고, 외부질의에서 주문에 대해서 내부질의에서 구한 custid와 같은 custid를 갖는 고객을 찾아 둘을 조인하였다. → 주문을 한 고객별 name, 평균 구매가격을 구한다.

(3) SELECT SUM(saleprice) "total"
FROM Orders od
WHERE EXISTS (SELECT *
FROM Customer cs
WHERE custid <= 3 AND cs.custid = od.custid);
상관중첩질의이므로 외부질의부터 시행된다. → 고객번호가 3보다 작은 고객들의 총 판매금액을 구한다.

5. 뷰의 장점과 단점을 설명하시오.

장점	• **편리성**: 미리 정의된 뷰를 일반 테이블처럼 사용할 수 있기 때문에 편리하다. 또 사용자가 필요한 정보만 요구에 맞게 가공하여 뷰로 만들어 쓸 수 있다. • **재사용성**: 자주 사용되는 질의를 뷰로 미리 정의해 놓을 수 있다. • **보안성**: 각 사용자별로 필요한 데이터만 선별하여 보여줄 수 있다. 중요한 질의의 경우 질의 내용을 암호화할 수 있다.
단점	• 실행 시 릴레이션으로부터 계산을 해야 하는 시간이 필요하다. • 데이터 조작(INSERT, DELETE, UPDATE)에 제한이 있다.

6. 다음에 해당하는 뷰를 작성하시오. 데이터는 마당서점 데이터베이스를 이용한다.

(1) 판매가격이 20,000원인 도서의 도서번호, 도서이름, 고객이름, 출판사, 판매가격을 보여주는 highorders 뷰를 생성하시오.

```
CREATE VIEW Highorders
AS   SELECT b.bookid, b.bookname, c.name, b.publisher, o.saleprice
     FROM Book b, Orders o, Customer c
     WHERE b.bookid=o.bookid AND o.custid=c.custid AND saleprice=20000;
```

(2) 생성한 뷰를 이용하여 판매된 도서의 이름과 고객의 이름을 출력하는 SQL 문을 작성하시오.

```
SELECT bookname, name FROM Highorders;
```

(3) highorders 뷰를 변경하고자 한다. 판매가격 속성을 삭제하는 명령을 수행하시오. 삭제 후 (2)번 SQL 문을 다시 수행하시오.

```
ALTER VIEW Highorders
AS   SELECT b.bookid, b.bookname, c.name, b.publisher
     FROM Book b, Orders o, Customer c
     WHERE b.bookid=o.bookid AND o.custid=c.custid AND saleprice=20000;
```

7. [사원 데이터베이스] 3장의 연습문제 9번 데이터베이스를 이용하여 다음 질의에 해당되는 SQL 문을 작성하시오.

(1) 팀장(MGR)이 없는 직원의 이름을 보이시오.

```
SELECT ename FROM Emp WHERE MGR IS NULL;
```

(2) 사원의 이름과 부서의 이름을 보이시오(조인/스칼라 부속질의 사용).

[조인]

```
SELECT ename, dname
FROM Emp, Dept WHERE Emp.deptno=Dept.deptno;
```

[스칼라 부속질의]

```
SELECT ename, (SELECT dname FROM Dept WHERE deptno=e.deptno)
FROM Emp e;
```

(3) 'CHICAGO'에 근무하는 사원의 이름을 보이시오(조인/인라인 뷰/중첩질의/EXISTS 사용).

[조인]

```
SELECT ename
FROM Emp, Dept WHERE Emp.deptno=Dept.deptno AND loc LIKE 'CHICAGO';
```

[인라인 뷰]

```
SELECT ename
FROM Emp, (SELECT deptno FROM Dept WHERE loc LIKE 'CHICAGO') d
WHERE Emp.deptno=d.deptno;
```

[중첩질의]

```
SELECT ename
FROM Emp
WHERE deptno IN
    (SELECT deptno FROM Dept WHERE loc LIKE 'CHICAGO');
```

[EXISTS]

```
SELECT ename
FROM Emp e
EXISTS
    (SELECT deptno FROM e.deptno=deptno AND loc LIKE 'CHICAGO')
```

(4) 평균보다 급여가 많은 직원의 이름을 보이시오.

```
SELECT name
FROM Emp
WHERE sal > (SELECT AVG(sal) FROM Emp);
```

(5) 자기 부서의 평균보다 급여가 많은 직원의 이름을 보이시오(상관 부속질의 사용).

```
SELECT name
FROM Emp e
WHERE sal > (SELECT AVG(sal) FROM Emp WHERE empno=e.empno);
```

8. [극장 데이터베이스 뷰] 다음은 4개의 지점을 가진 극장의 데이터베이스다. 밑줄 친 속성은 기본키이다. 아래 테이블을 보고 다음 뷰를 생성하시오.

(1) 극장이름과 고객이름을 저장하는 극장-고객 뷰를 생성하시오.

```
CREATE VIEW 극장-고객
AS      SELECT a.극장이름, c.고객이름
        FROM 극장 a, 예약 b, 고객 c
        WHERE a.극장번호=b.예약번호 AND b.고객번호=c.고객번호;
```

(2) '대한' 극장에 예약을 한 고객의 수를 날짜별로 저장하는 대한-고객수 뷰를 생성하시오.

```
CREATE VIEW 대한-고객수
AS      SELECT 날짜, COUNT(*)
        FROM 극장 a, 예약 b
        WHERE a.극장번호=b.예약번호 AND 극장이름 LIKE '대한'
        GROUP BY 날짜;
```

9. 8번의 영화관 데이터베이스에 대하여 다음과 같은 뷰를 생성하였다. 다음 질의에 대하여 의미가 있는지 판단하고 질의 결과가 어떤 내용인지 설명하시오.

(문제수정-빨간색)
8번의 극장 데이터베이스에 대하여 다음과 같은 뷰를 생성하였다. 다음 질의에 대하여 의미가 있는지 판단하고 질의 결과가 어떤 내용인지 설명하시오.

```
CREATE VIEW 극장예약(극장이름, 예약수)
AS SELECT A.극장이름, COUNT(*)
FROM 극장 A, 예약 B
WHERE A.극장번호=B.극장번호
GROUP BY A.극장이름;
```

```
(1) SELECT * FROM 극장예약;
(2) SELECT 예약수 FROM 극장예약 WHERE 극장이름='강남';
(3) SELECT MIN(예약수) FROM 극장예약;
(4) SELECT COUNT(*) FROM 극장예약;
(5) SELECT 극장이름 FROM 극장예약 WHERE 예약수>100;
(6) SELECT 극장이름 FROM 극장예약 ORDER BY 예약수;
```

(1) SELECT * FROM 극장예약;

극장이름과 예약 수를 검색

(2) SELECT 극장이름 FROM 극장예약 WHERE 극장이름='강남';

극장이름이 강남인 극장이름 검색

(2-문제수정) SELECT 예약수 FROM 극장예약 WHERE 극장이름='강남';
극장이름이 강남인 예약수 검색

(3) SELECT MIN(예약수) FROM 극장예약;
극장예약에서 가장 예약수가 수를 검색

(4) SELECT COUNT(*) FROM 극장예약;
예약이 있는 극장 수를 검색

(5) SELECT 극장이름 FROM 극장예약 WHERE 예약수 > 100;
예약 수가 100을 넘는 극장이름을 검색

(6) SELECT 극장이름 FROM 극장예약 ORDER BY 예약수
극장이름을 검색하되 예약수 순으로 오름차순 정렬하여 검색

10. 데이터베이스는 하드디스크에 저장된다. 하드디스크에서 데이터를 읽어 오는 데 걸리는 시간(액세스 시간)은 어떻게 구성되는지 설명하시오.

액세스 시간 = 탐색시간(seek time, 액세스 헤드를 트랙에 이동시키는 시간)
+ 회전지연시간(rotational latency time, 섹터가 액세스 헤드에 접근하는 시간)
+ 데이터 전송시간(data transfer time, 데이터를 주기억장치로 읽어오는 시간)

11. B-tree는 균형잡힌 트리를 말한다. 차수가 3인 비어있는 B-tree에 1부터 9까지 삽입해보고 균형을 어떻게 유지하는지 설명하시오.
참고자료 1부터 7까지 삽입 과정(8,9는 생략)
http://upload.wikimedia.org/wikipedia/commons/3/33/B_tree_insertion_example.png

12. [사원 데이터베이스, 뷰] 3장의 연습문제 9번 데이터베이스를 이용하여 다음의 뷰를 생성하시오. 뷰를 이용하여 또 다른 뷰를 정의할 수 있는지 다음 SQL 문을 실행해보고 결과를 보이시오.
scott 계정에서 view를 생성하려면 view를 생성하는 권한이 필요하다. 다음과 같이 권한을 부여한 후 실행해본다.

```
/* create view 권한 부여하기 - 툴이 아닌 spl 창을 이용하여 권한 부여 */
/*   system 접속 : */
        conn system/비밀번호
        grant create view to scott
```

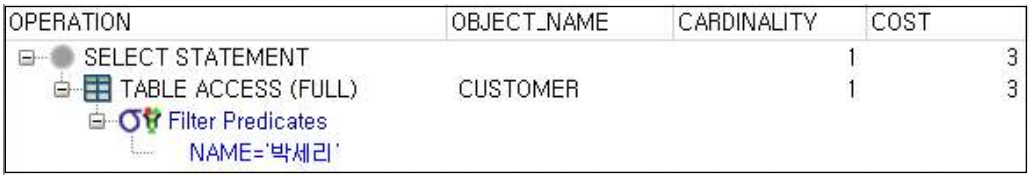
	EMPNO	ENAME	SAL	DNAME
1	7844	TURNER	1500	SALES
2	7499	ALLEN	1600	SALES

13. [마당서점 데이터베이스 인덱스] 마당서점 데이터베이스에서 다음 SQL 문을 수행하고 데이터베이스가 인덱스를 사용하는 과정을 확인해보시오.

(1) 다음 SQL 문을 수행해본다.

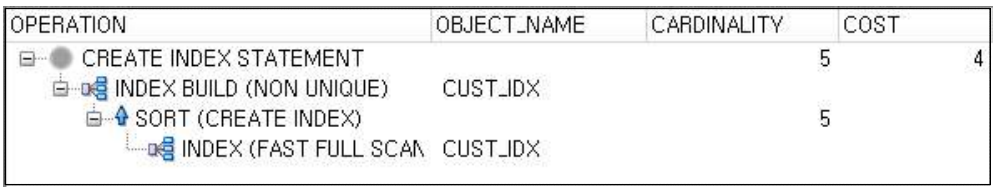
```
SELECT name FROM Customer WHERE name LIKE '박세리';
```

(2) 실행 계획을 살펴본다. 실행 계획은 [F10] 키를 누른 후 [계획 설명] 탭을 선택하면 표시된다(SQL Developer를 실행하고 SQL 문 위에서 오른쪽 마우스 클릭-[설명]-[계획 설명]).



(3) Customer 테이블에 name으로 인덱스를 생성하시오. 생성 후 (1)번의 SQL 문을 다시 수행하고 실행 계획을 살펴보시오.

```
CREATE INDEX cust_idx ON Customer(name);
```



(4) 같은 질의에 대한 두 가지 실행 계획을 비교해보시오.

인덱스 생성 후 인덱스를 먼저 검색하여 name LIKE '박세리' 조건 수행

(5) (3)번에서 생성한 인덱스를 삭제하시오.

```
DROP INDEX cust_idx;
```

5장. 데이터베이스 프로그래밍

1. PL/SQL에 대한 설명 중 가장 거리가 먼 것은?
① 기존 SQL 문과는 다른 것이다.

2. 저장 프로시저의 장점이 아닌 것은?
① SQL 질의 전체를 전송하는 대신 매개변수만 전달하여 네트워크 트래픽을 증가시킨다.

3. 다음 중 프로시저를 실행시키는 명령어는?
① EXEC

4. SQL에서 데이터베이스가 미리 정해놓은 조건을 만족하거나 어떤 동작이 수행되면 자동으로 동작하는 객체를 무엇이라고 하는가?
① 트리거

5. 다음 중 트리거를 만들기 위한 명령어는?
③ CREATE TRIGGER

6번-11번. (생략)

6장. 데이터 모델링

1. 데이터베이스 설계 순서로 옳은 것은?
② 요구사항 분석 → 개념적 모델링 → 논리적 모델링 → 물리적 모델링 → 데이터베이스 구현

2. ER 모델의 표현 방법으로 옳지 않은 것은?
③ 속성-오각형

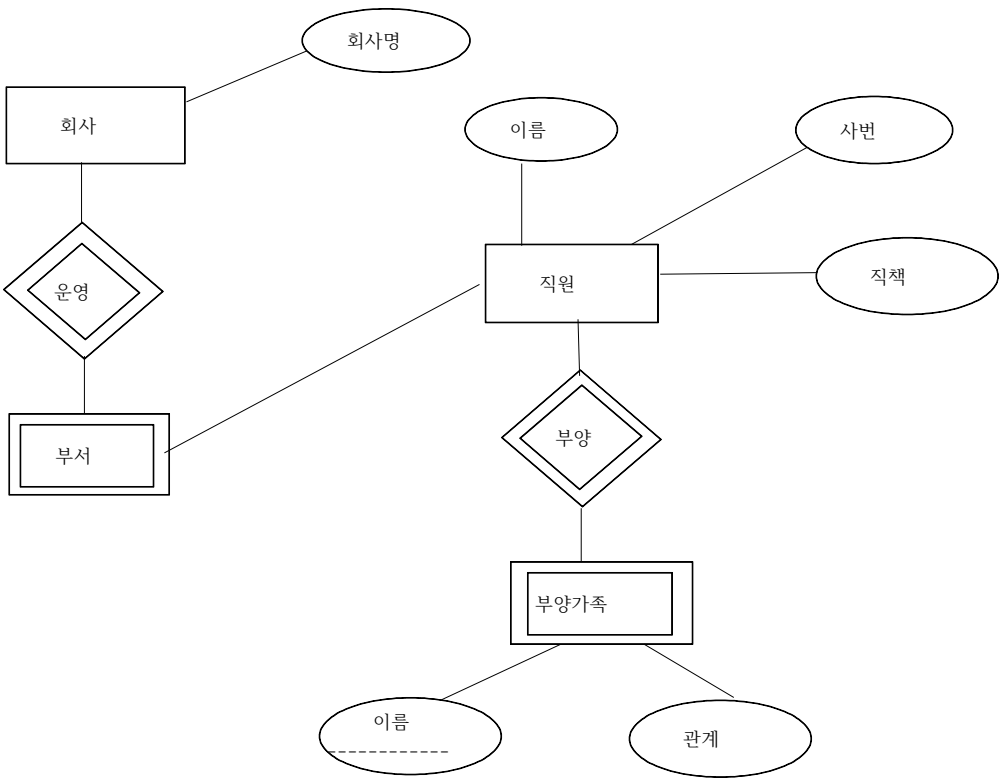
3. ER 모델에 대한 설명으로 옳지 않은 것은?
② 일대일(1:1) 관계 유형만 표현할 수 있다.

4. ER 표기법에 대한 설명 중 옳지 않은 것은?
② 약한 개체의 식별자

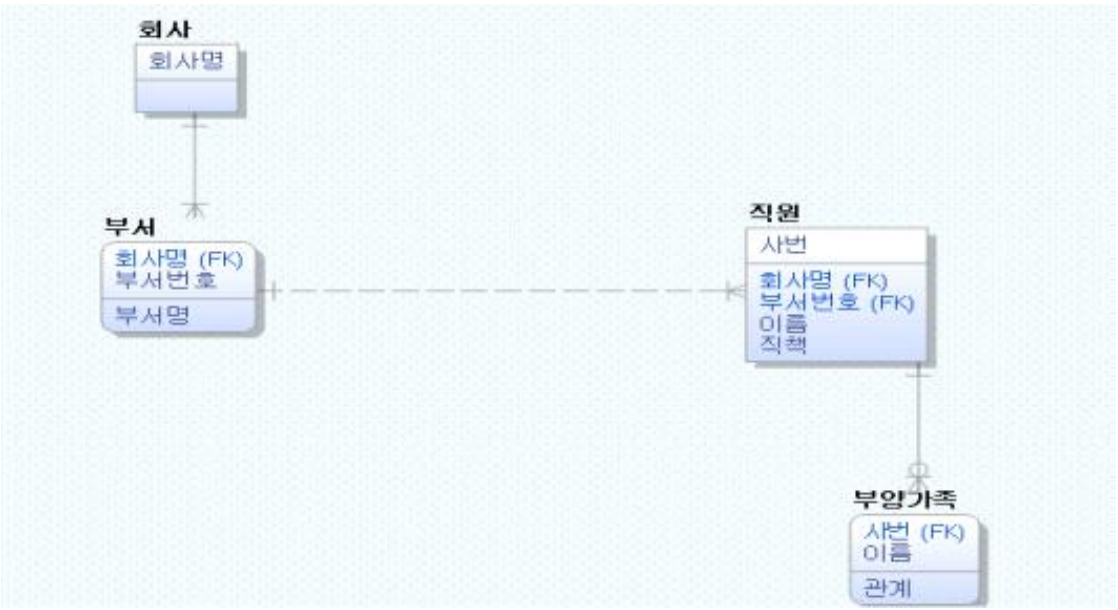
5 IE 표기법에 대한 설명으로 옳지 않은 것은?
④ 0(필수적 참여)

6. 다음 내용을 모두 포함하는 데이터베이스를 설계하시오. 필요한 경우 몇 가지 가정을 넣을 수 있다.
※ 정답은 설계 해석에 따라 달라질 수 있음(샘플 답, 정답은 여기에 가감이 필요함).

(1) ER 다이어그램을 그리시오.



(2) ER 다이어그램을 IE 표기법으로 변환하여 그리시오.



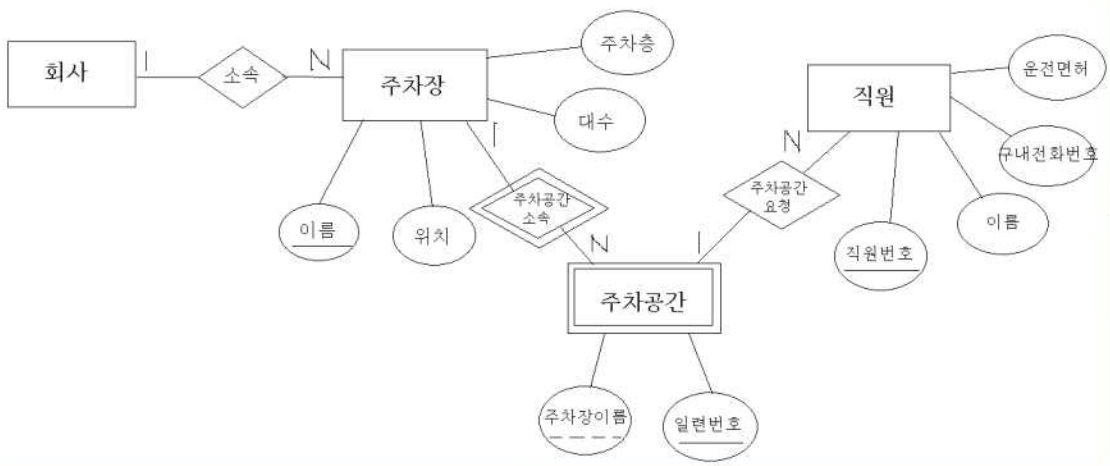
(3) ER 다이어그램을 테이블로 변환하시오.

회사(회사명)
부서(회사명, 부서번호, 부서명)
직원(사번, 회사명, 부서번호, 이름, 직책)
근무기록(사번, 부서번호, 기간, 직책)
부양가족(사번, 이름, 관계)

7. 다음 내용을 모두 포함하는 데이터베이스를 설계하시오. 필요한 경우 몇 가지 가정을 넣을 수 있다.

※ 정답은 설계 해석에 따라 달라질 수 있음(샘플 답, 정답은 여기에 가감이 필요함).

(1) ER 다이어그램을 그리시오.



(2) ER 다이어그램을 테이블로 변환하시오.

주차장(이름, 위치, 대수, 주차층) 주차공간(주차장이름, 일련번호) 직원(직원번호, 이름, 구내전화번호, 운전면허)

(3) ERwin을 이용하여 모델링하시오.

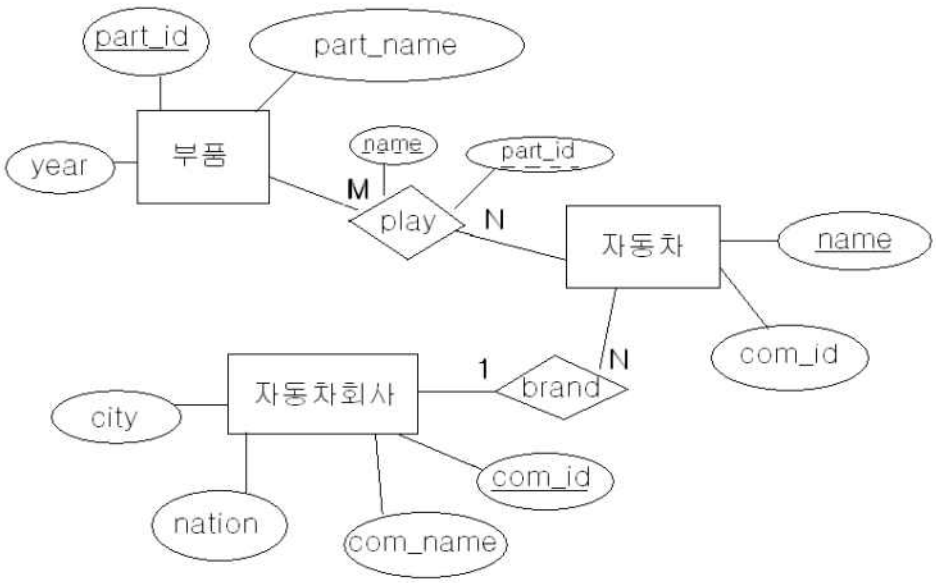
8. 다음은 고객과 주문에 관한 ER 다이어그램이다. 개체는 고객(Customer), 제품(Product), 주문(Invoice)으로 구성된다. Place 관계는 '주문한다'를, LineItem은 '주문 항목'을 의미한다. 그림에 해당하는 테이블을 작성하시오(변환된 테이블의 기본키는 밑줄 실선, 외래키는 밑줄 점선으로 표시한다. 기본키인 동시에 외래키일 경우에는 밑줄 실선으로 표시한다. 테이블 변환을 위하여 필요한 사항 중 설명되지 않은 것은 임의로 정하여 설계한다).

Invoice(Invoice#, TotalPrderAmt, Date, Terms, ShipVia, Cus#(FK))
Customer(Cus#, CName, Street, City, Phone, Zip, State)
Product(Prod#, StandartPrice, Description)
LineItem(Invoice#(FK), Prod#(FK), SellPrice, Quantity)

9. [부품 데이터베이스] 글로벌 자동차 부품업체 A사는 자동차 회사에 부품을 공급하며 부품 공급에 대한 데이터베이스를 구축하고 있다. 요구사항은 아래와 같다. 이를 바탕으로 개념적 모델링 과정을 거쳐 ER 다이어그램을 작성하고 논리적 모델링 과정을 거쳐 관계 데이터 모델로 사상해보시오.

※ 정답은 설계 해석에 따라 달라질 수 있음(샘플 답, 정답은 여기에 가감이 필요함).

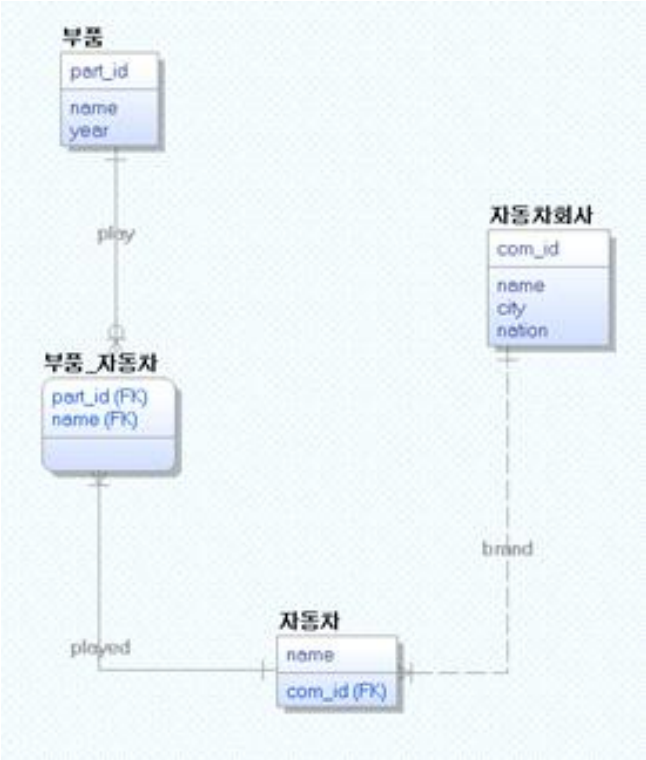
(1) ER 다이어그램을 그리시오.



(2) ER 다이어그램을 테이블로 변환하시오.

부품(고유번호(part_id), 이름(part_name), 제작년도(year))
자동차(고유번호(com_id), 이름(name), com_name(FK))
자동차회사(고유번호(com_id), 이름(com_name), 도시(city), 국가(nation))
조립(part_id, com_id, 고유번호(part_id), 이름(name))

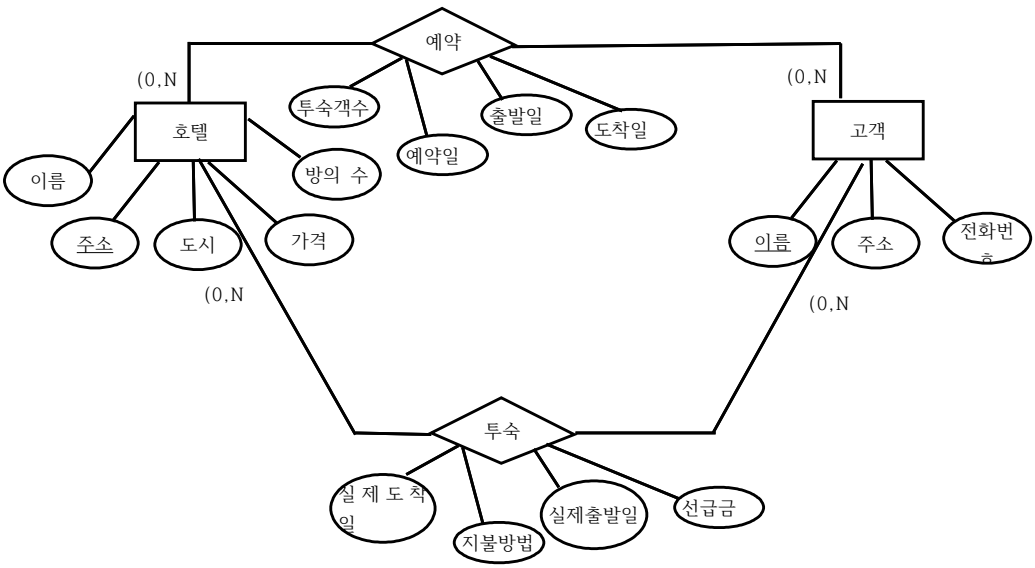
(3) ERwin을 이용하여 모델링하시오.



10. [호텔 데이터베이스] 여러 개의 지점을 가진 호텔을 데이터베이스로 구축하려고 한다. 다음 내용을 모두 포함하는 데이터베이스를 설계하시오. 필요한 경우 몇 가지 가정을 넣을 수 있다.

※ 정답은 설계 해석에 따라 달라질 수 있음(샘플 답, 정답은 여기에 가감이 필요함).

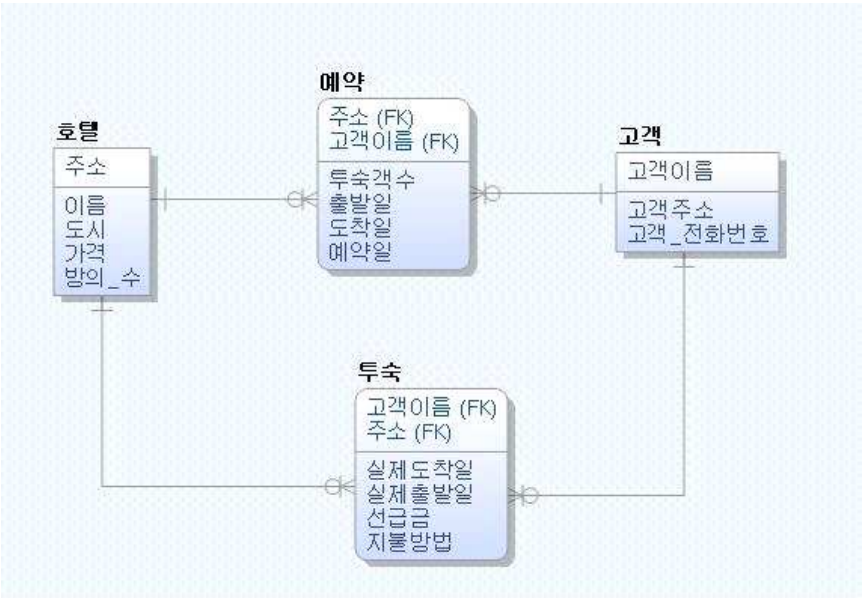
(1) ER 다이어그램을 그리시오.



(2) ER 다이어그램을 테이블로 변환하시오.

호텔(<u>주소</u>, 이름, 도시, 가격, 방의 수) 고객(<u>이름</u>, 주소, 전화번호) 예약(<u>호텔주소, 고객이름</u>, 투숙객수, 예약일, 출발일, 도착일) 투숙(<u>호텔주소</u>, <u>고객이름</u>, 실제도착일, 실제출발일, 선급금, 지불방법)

(3) ERwin을 이용하여 모델링하시오.



11. **(생략)**
12. **(생략)**

7장. 정규화

1. 정규화의 필요성으로 거리가 먼 것은?
② 중복 데이터의 활성화

2. 관계 데이터베이스의 정규화에 대한 설명으로 옳지 않은 것은?
② 정규화의 목적은 각 릴레이션에 분산된 종속성을 하나의 릴레이션에 통합하는 것

3. 정규화 과정에서 발생하는 이상현상에 관한 설명으로 옳지 않은 것은?
② 속성 간의 종속관계를 분석하여 여러 개의 릴레이션을 하나로 결합하여 이상현상을 해결한다.

4. 데이터의 중복으로 인해 릴레이션 조작 시 발생하는 이상현상에 관한 설명 중 옳지 않은 것은?
② 어떤 데이터를 삽입할 때 불필요하고 원하지 않는 데이터도 함께 삽입해야 되거나 삽입되지 않는 경우를 삽입이상이라고 한다.

5. 제 1정규형에서 제 2정규형이 되기 위한 조건은?
④ 키가 아닌 모든 속성이 기본키에 완전 함수 종속되어야 한다.

6. 제 2정규형에서 제 3정규형이 되기 위한 조건은?
① 이행적 함수 종속을 제거해야 한다.

7. 제 3정규형에서 보이스코드 정규형(BCNF)이 되기 위한 조건은?
④ 결정자가 후보키가 아닌 함수적 종속을 제거해야 한다.

8. 보이스코드 정규형(BCNF)에 대한 옳은 설명으로만 짝지어진 것은?
① ㉠, ㉡

9. 다음 중 보이스코드 정규형(BCNF)을 만족하기 위한 조건으로 옳게 짝지어진 것은?
② ㉠, ㉡, ㉢, ㉣

10. 다음과 같이 어떤 릴레이션 R과 그 릴레이션에 존재하는 종속성이 주어졌을 때 릴레이션 R은 몇 정규형인가?
③ 제 3정규형

11. 다음 릴레이션에서 함수 종속성과 키를 찾아보시오. 필요한 전제 사항을 포함하여 답하시오.
학생(학번), 강좌번호, 학기에 대한 성적 기록으로 볼 수 있다.
따라서 '(학번, 강좌번호, 학기) → 성적'이며 키는 (학번, 강좌번호, 학기)이다.

12. 다음 릴레이션에서 AB는 후보키가 될 수 있는가? 아니라면 후보키를 찾아보시오.

AB → C

ABD → E(AB → C, CD → E의 유사이행 규칙에 의하여)

ABD → CE

ABD → ABCDE 이므로 ABD 가 후보키이다.

13. 다음 릴레이션 R을 보고 오른쪽 함수 종속성 중에서 성립하는 것을 모두 고르시오.

B → D, C → A, C → B, C → D, D → B

14. 다음 릴레이션 R을 보고 아래 함수 종속성이 성립하는지 답하시오. 그 이유도 설명하시오.

(3),(5),(6)이 함수 종속성 성립

다른 값들은 결정자에 따라 유일한 속성을 갖지 않으므로 성립하지 않음

15. 아래의 릴레이션 R1(A, B, C)을 릴레이션 R2(A, B)와 R3(A, C)로 분해한 후, 속성 A를 사용하여 다시 조인하면 어떤 가짜 투플이 생기는가? 생긴다면 이유는 무엇인지 설명하시오.

무손실 분해 조건을 만족시키지 않아서 가짜 투플이 생성되었다. 무손실 분해의 공통속성이 있다는 조건은 성립하였으나, 이 공통속성이 분해한 릴레이션의 기본키가 된다는 속성은 성립하지 않았다. 가짜 투플은 다음과 같다.

A	B	C
1	1	2
1	2	1

16. 다음 릴레이션은 몇 정규형인지 말하고 보이스코드 정규형(BCNF)으로 정규화하시오.

제1정규형만 만족한다.

물품A(<u>물품번호, 행사번호</u>, 가격), 물품B(<u>물품번호</u>, 제조사, 스타일)

17. 다음 릴레이션 X에서 성립하는 정규형은 무엇인가?

(1) 제2정규형(이행적 종속성이 있음)

(2) 제1정규형(부분함수종속)

18. 다음은 배송(Shipping) 물품에 대한 릴레이션이다.

(1) 후보키를 찾으시오.

shipname, date

(2) 제 2정규형으로 정규화하시오.

S1(shipname, shiptype), S2(shipname, voyageID, cargo, port, date)

(3) 제 3정규형으로 정규화하시오.

S1(shipname, shiptype), S2(voyageID, cargo), S3(shipname, voyageID, port, date)

(4) BCNF로 정규화하시오.

S1(shipname, shiptype), S2(voyageID, cargo), S3(voyageID, port, date),
S4(voyageID, shipname)

19. 릴레이션 Book(booktitle, authorname, booktype, listprice, authorgroup, publisher)에서 함수 종속성은 다음과 같다.

(1) 릴레이션은 몇 정규형인가? 그 이유를 같이 설명하시오.

제1 정규형, 불완전 함수 종속

(2) 정규화를 수행하시오.

R1(booktitle, booktype, publisher, booktype)
R2(booktype, listprice)
R3(authorname, authorgroup)

20. 다음은 부품과 공급자에 대한 릴레이션 Part(partnumber, description, supplier, suppaddress, price)이다. 물음에 답하시오.

(1) 함수 종속성을 찾아보시오.

(partnumber, supplier) → price
partnumber → description
supplier → suppaddress

(2) 릴레이션 Part는 몇 정규형인가?

1정규형

(3) 다음과 같이 분해를 했을 때 각각의 릴레이션은 몇 정규형인가?

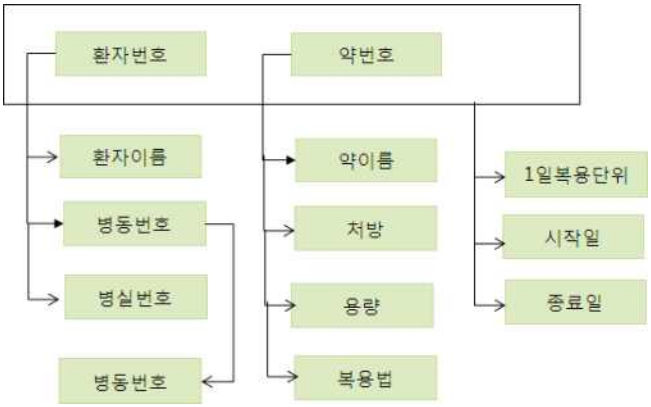
1정규형

(4) (3)번의 릴레이션에서 분해가 더 필요한가? 필요하면 분해를 수행하시오.

R1(partnumber, supplier, price)
R2(supplier, suppaddress)
R3(partnumber, description)

21. [병원 데이터베이스] 다음은 마당병원의 환자 처방 폼의 일부다.

(1) 처방 폼에 있는 속성을 보고 함수 종속성을 찾아내고 함수 종속성 다이어그램을 그리시오. 그리고 키를 정하시오. 필요한 경우 속성이나 데이터 값에 조건을 붙여도 좋다.



(2) 함수 종속성 다이어그램을 보고 정규화를 수행하시오. 최소 제 3정규형이 되도록 만드시오.

(3) 정규화 후 분해된 테이블에 대하여 기본키와 외래키를 표시하시오.

환자(환자번호, 환자이름, 병동번호, 병실번호)
병동(병동번호, 병동이름)
약(약번호, 약이름, 처방, 용량, 복용법)
처방(환자번호(FK), 약번호(FK), 1일복용단위, 시작일, 종료일)

22. (생략)

8장. 트랜잭션, 동시성 제어, 회복

1. 트랜잭션(Transaction)은 일련의 연산 집합이란 의미로, 하나의 논리적인 기능을 수행하는 작업의 단위다. 트랜잭션이 가져야 할 성질로 거리가 먼 것은?
④ 병행성(Concurrency)

2. 트랜잭션에 대한 설명으로 옳지 않은 것은?
④ 트랜잭션 연산이 데이터베이스에 모두 반영되지 않고 일부만 반영되는 것을 원자성이라고 한다.

3. 트랜잭션의 성질에 대한 설명으로 옳지 않은 것은?
③ 어느 하나의 트랜잭션 실행 중에 다른 트랜잭션이 동시에 실행될 수 없다.

4. 트랜잭션의 동시성에 관한 설명 중 옳지 않은 것은?
④ 2단계 락킹을 사용하면 데드락 현상은 발생하지 않는다.

5. 병행제어의 락킹(locking) 단위에 대한 설명으로 옳지 않은 것은?
① 락킹 단위가 작아지면 병행성 수준이 낮아진다.

6. 병행제어에 영향을 주는 요소로 한 번에 락(lock)되어야 할 데이터의 크기를 락킹 단위(locking granularity)라고 한다. 이 단위가 클 경우에 대한 설명으로 옳지 않은 것은?
① 병행성 수준이 높아진다.

7. 2단계 락킹에 대한 설명으로 옳지 않은 것은?
④ 각 트랜잭션의 락 요청과 해제 요청을 2단계로 실시한다.

8. 하나의 트랜잭션 수행이 실패한 후 회복되기 전에 다른 트랜잭션이 실패한 갱신 결과를 참조하는 현상은?
① Lost Update

9. 회복을 위한 로그 기록 방법 중 지연 갱신(deferred update)에서 사용하는 방법은 어느것인가?
③ NO-UNDO/REDO

10. 오라클의 기본 고립 수준은 무엇인가?
② READ COMMITTED

11. 트랜잭션이 다음과 같이 동시에 수행될 때 결과를 구하시오. read_item()은 버퍼에 있는 데이터베이스를 읽는 함수이고, write_item()은 버퍼에 데이터베이스를 기록하는 함수이다. A1, A2는 트랜잭션 T1의 지역변수이고, B1, B2는 트랜잭션 T2의 지역변수이다. 트랜잭션이 시작하기 전의 데이터베이스에서 X와 Y의 값은 각각 1000이다.

(1) [스케줄 1]과 같이 수행하였을 때, 수행 후 A와 B의 값을 구하시오.

A:990, B:1010

(2) [스케줄 2]와 같이 수행하였을 때, 수행 후 A와 B의 값을 구하시오.

A:1100, B:1100

(3) [스케줄 2]와 같이 트랜잭션이 동시에 수행되면 문제가 있는지 없는지 판단하시오. 그리고 그 이유를 설명하시오.

T2가 갱신한 값 X가 T1에 의해서 덮어씌워져서 갱신손실이 발생한다.
또 T1이 갱신한 값 Y가 T2에 의해서 덮어씌워져서 갱신손실이 발생한다.

(4) 이 문제를 해결하는 방법을 설명하시오. 이 경우 [스케줄 2]가 어떻게 진행되어야 하는지 설명하시오.

각각의 트랜잭션에 락을 걸어서 수정 중인 데이터를 읽지 않도록 한다.

12. 다음 트랜잭션 T1, T2가 동시에 실행될 때 공유락과 배타락을 사용한다. 트랜잭션 T1의 첫 번째 질의의 수행결과는 20이고,두 번째 질의의 수행결과는 21이다.

(1) 이러한 현상이 일어나는 이유와 현상의 이름을 설명하시오.

트랜잭션이 데이터를 읽는 동안 다른 트랜잭션이 수정을 했다.

(2) (1)의 현상을 방지하기 위한 방법을 설명하시오.

트랜잭션 수행 중에 다른 트랜잭션이 변경하지 않도록 LOCK을 건다.

(3) (1)의 현상을 방지하기 위한 SQL 문을 작성하시오.

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;

13. 다음 트랜잭션 T1, T2가 동시에 실행될 때 결과에 대하여 다음 물음에 답하시오.

(1) 두 트랜잭션을 수행하기 전 ①번의 수행 결과가 10000이라면, 두 트랜잭션을 위의 순서대로 실행했을 때 ②번의 수행 결과는 무엇인가?

10100

(2) ①번과 ②번의 결과가 트랜잭션 T2의 영향을 받지 않으려면 어떻게 해야 하는가?

트랜잭션 분리 수준을 높인다. - REPEATABLE READ;

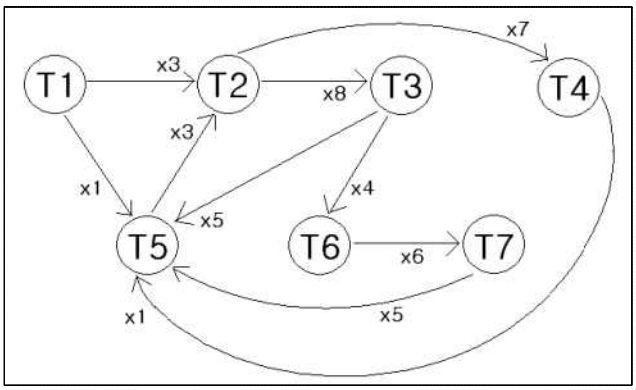
(3) ③번의 COMMIT 문을 ④번으로 옮겨 다시 실행하면 어떻게 되는가?

트랜잭션 T1이 대기상태에 빠진다.

(4) (3)번과 같이 실행하면서 (2)번의 실행결과를 나타나게 하려면 어떻게 해야 하는가?

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED 문 추가

14. 데드락은 대기 그래프(wait-for graph)를 그려보면 발생 여부를 판단할 수 있다. 다음 표를 보고 대기 그래프를 그리고, 데드락의 발생 여부를 판단하시오(대기 그래프에서 사이클이 존재하면 데드락이 발생한 것이다. 대기 그래프는 트랜잭션을 노드로, 락 요청을 화살표로 표현한다).



T2→T3→T5→T2, T2→T4→T5→T2 에 사이클이 생겨 데드락이 발생한다.

15. 다음은 어느 트랜잭션 T1, T2, T3의 로그 기록이다. 지연 갱신 방법을 사용한다고 가정하고 15번 로그에서 시스템 장애가 일어났을 때 각 트랜잭션 T1, T2, T3를 복구하는 방법에 대하여 설명하시오.

T1은 checkpoint가 있으므로 아무 일도 하지 않고,
T2는 COMMIT이 있으므로 REDO 작업을 한다.
T3은 COMMIT이 없으므로 아무 일도 하지 않는다.

9장. 데이터베이스 보안과 관리

1. 오라클을 사용하기 위해 접속하는 것을 무엇이라고 하는가?
④ 로그인

2. 다음 중 데이터베이스 관리 업무가 아닌 것은?
④ 판매 자료 입력 및 관리

<1~4쇄>
3. 다음 중 GRANT 문에서 열 단위로 부여할 수 있는 권한은 무엇인가?
③ SELECT, UPDATE

<5쇄>
3. 다음과 같은 일련의 권한 부여 SQL 명령에 대한 설명 중 옳지 않은 것은?
① U1은 STUDENT에 대한 검색 권한이 없다.

---

4. 데이터베이스 보안에 대한 설명 중 옳지 않은 것은?
③ 보안을 위해 사용자에게 권한을 부여할 때는 관리자의 정책 결정보다 DBMS가 자체적으로 결정하여 제공한다.

5. 허가받은 권한을 다른 사용자에게 부여할 때 사용하는 명령은 무엇인가?
① WITH GRANT OPTION

6. 사용자별로 권한을 관리하는 어려움을 해결하기 위해 서로 연관된 권한을 그룹으로 정의하는 개념은 무엇인가?
③ ROLE

7. 다음은 어떤 파일에 대한 설명인가?

데이터베이스 내에서 일어나는 모든 트랜잭션 정보를 저장하는 파일이다. 이 파일은 특정 시점까지의 자료를 복원하는 데 사용할 수 있다.

④ Archive Log File
* Redo log 는 정보를 기록, Acrhive Log 는 Redo log를 복사하여 생성되며 복구 시 사용

8. 데이터베이스 전체를 백업하는 것을 무엇이라고 하는가?
③ 전체 백업

9 다음 중 오라클에서 사용할 수 있는 백업 방법이 아닌 것은 무엇인가

② SNAPSHOT

10. 다음 중 차등 백업의 특징이 아닌 것은?

③ 데이터베이스 개체, 시스템 테이블, 데이터 등 데이터베이스 전체를 백업한다.

11. GRANT와 REVOKE 명령어는 객체의 소유자가 권한을 부여하고, 다시 회수할 때 사용한다. 두 명령어의 기능을 상세하게 설명하시오.

[GRANT : 권한 부여]

```
GRANT { ALL [ PRIVILEGES ] }
      | permission [ ( column [ ,...n ] ) ] [ ,...n ]
      [ ON [ class :: ] securable ] TO principal [ ,...n ]
      [ WITH GRANT OPTION ] [ A
```

[REVOKE : 권한 제거]

```
REVOKE [ GRANT OPTION FOR ]
      {
        [ ALL [ PRIVILEGES ] ]
        |
                permission [ ( column [ ,...n ] ) ] [ ,...n ]
      }
      [ ON [ class :: ] securable ]
      { TO | FROM } principal [ ,...n ]
      [ CASCADE] [ AS principal ]
```

테이블 대상 사용가능 권한은 SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER, DROP 등이 있다.

12. 다음 각 조건에 맞게 권한을 부여하는 정의문을 완성하시오.

(1) madang 사용자가 WITH GRANT OPTION 없이 mdguest에게 Order 테이블의 INSERT 권한을 부여

madang 사용자) GRANT INSERT ON Order TO mdguest;

(2) madang 사용자가 WITH GRANT OPTION과 함께 mdguest2에게 Order 테이블의 SELECT, DELETE 권한을 부여

```
madang 사용자) GRANT SELECT, DELETE ON Order TO mdguest2
                    WITH GRANT OPTION;
```

(3) madang 사용자가 WITH GRANT OPTION 없이 mdguest에게 Book 테이블의 name에 대해 UPDATE 권한을 부여

madang 사용자) GRANT UPDATE (name) ON Book TO mdguest;

(4) madang 사용자가 자신이 소유한 Customer의 기본키인 custid에 대한 REFERENCES 권한을 사용자 mdguest2에게 부여

madang 사용자) GRANT REFERENCES (custid) ON Customer TO mdguest2;

13 오라클 시스템 데이터 파일의 종류와 각각의 목적 및 기능을 설명하시오.

- **제어 파일(control files)** : 데이터베이스를 구성하는 모든 파일의 정보를 가짐
- **데이터 파일(data files)** : 데이터베이스 내에 저장된 실제 데이터를 저장(인덱스, 데이터 등)
- **리두 로그 파일(redo log files)** : 운영 중 발생되는 트랜잭션과 저장된 결과에 대한 기록을 저장
- **아카이브 로그 파일(archive log files)** : 계속 증가하는 리두 로그 파일을 기록한 것으로 데이터베이스의 복구 시 사용

14 오라클 system 계정과 일반 사용자 계정의 차이점을 설명하시오.

- **sys 계정** : 데이터베이스 전체에 대한(startup, shutdown 포함) 관리 계정
- **system 계정** : sys 계정에 의해 부여된 데이터베이스에 대한 관리자 계정, 사용자 생성, 테이블 생성 등의 관리자 계정
- **사용자 계정** : 데이터베이스의 사용자로서 관리자에게서 부여된 권한 내에서만 처리할 수 있는 계정

15 오라클의 인증 방식을 알아보고 설명하시오.

- **OS 인증** : OS의 특정 그룹에 속한 유저로 로그인된 사용자는 모두 sys 계정으로 인정하는 방법
- **Password 인증** : password 파일에 기술된 암호를 통해 관리자로 접속하는 방법

16 오라클에서 사용하는 TNS와 NLS에 대해서 조사하고 설명하시오.

- **TNS(Transparent Network Substrate)** : 오라클에서 사용하는 표준 프로토콜을 위한 공통 인터페이스로 서버와 클라이언트간의 데이터 송수신 등의 통신처리 방법
- **NLS(National Language Support)** : 오라클 데이터베이스의 기본 국가 정보 즉, 문자규격, 날짜, 숫자, 통화방법 등을 표현하는 파라미터

17 오라클의 STARTUP 4단계에 대해 설명하시오.

- **CLOSE** : Database, Instance 모두 닫힌 상태
- **NOMOUNT** : 파라미터 내용으로 instance를 구성하는 단계
- **MOUNT** : Instance와 Database가 맞춰지는 단계, 제어 파일 정보를 읽음, DB 구조변경 및 복구 작업 수행
- **OPEN** : 실제 데이터 파일, 리두 로그 파일 등 물리적인 파일을 열어 모든 액세스가 사용 가능한 상태

18. 전체 백업과 차등 백업의 차이점을 설명하시오.

- **전체 백업:** 데이터베이스 개체, 시스템 테이블, 데이터 등 데이터베이스 전체를 백업한다. 전체 복구, 대량 로그 복구, 단순 복구 모델에서 모두 사용가능하다. 전체 백업은 최초에 데이터베이스를 생성하였을 때나 데이터베이스에 변경이 있을 때 수행하는 것이 좋다. 백업을 수행할 때마다 수행 시점의 모든 데이터를 백업하기 때문에 여러 번 하면 각 백업 파일에 데이터가 중복 저장된다. 또한 데이터의 양이 많을 경우 백업을 수행할 때마다 많은 시간이 소요된다. 복원 시 전체 백업을 이용하면 전체 데이터를 복원한다.
- **차등 백업:** 차등 백업은 전체 백업을 수행한 이후 변경된 데이터만 저장한다.