『AI 시대의 데이터베이스 개론』

연습문제·실전문제 정답 및 상세 해설

MCAIT 홈페이지 제공용 · v193 최종원본 대응

MySQL 8.4 LTS 기준 · 2026 년 9 월 검수

자료 이용 안내

이 자료는 『AI 시대의 데이터베이스 개론』 v193 최종원본의 Chapter 01~13 장 끝 연습문제와 실전문제에 대응한다.

객관식은 정답 번호와 핵심 근거를, 단답형·서술형은 정답 또는 예시 답안을 제공한다. 서술형과 실전문제는 표현이 하나로 고정되지 않으므로 교재의 개념과 검산 기준을 충족하는지를 중심으로 판단한다.

SQL 관련 수치와 결과는 Chapter 05 의 공통 데이터베이스 MCAIT_UNIVERSITY 초기 상태(DEPARTMENT 4, STUDENT 10, COURSE 8, SECTION 10, ENROLLMENT 18)를 기준으로 한다. 서버 버전·계정·EXPLAIN 비용처럼 환경에 따라 달라지는 값은 고정 정답으로 단정하지 않는다.

Chapter 01

객관식 정답 및 해설

문항	정답	해설
1	②	정보는 데이터를 목적과 맥락에 맞게 처리하여 의미를 드러낸 결과다.
2	③	판매 경향을 바탕으로 다음 준비량을 결정하는 것은 정보에 경험·규칙을 결합한 지식의 사례다.
3	①	같은 전화번호를 여러 파일에 반복 저장하면 중복이 생기고 일부만 수정할 때 불일치가 발생한다.
4	②	JSON 은 키와 값, 계층 구조처럼 의미를 구분할 표지가 있어 반정형 데이터에 해당한다.
5	③	AI 결과는 문장 자연스러움이 아니라 데이터 출처·기준 시점·결과 범위·접근 권한으로 검증해야 한다.

단답형·서술형 예시 답안

1. 정보(Information).

   데이터를 목적과 맥락에 맞게 처리하여 의미를 드러낸 결과가 정보다.

2. 반정형 데이터(Semi-Structured Data).

   JSON 은 고정 열 구조는 아니지만 키·값과 계층 구조라는 구조 표지가 있다.

3. 데이터 불일치(Data Inconsistency).

   같은 전화번호가 여러 파일에 중복 저장된 상태에서 일부 파일만 수정되어 서로 다른 값이 남은 문제다.

4. 예시: 도서번호·회원번호·대출시각의 개별 기록은 데이터다. 한 달 동안 도서 A 가 25 회 대출되었다는 집계는 정보다. 시험 기간에 특정 전공서 대출이 반복적으로 증가한다는 정보를 근거로 추가 비치를 결정한다면 행동에 활용하는 지식이 된다.

   데이터→정보→지식으로 갈수록 계산·맥락·경험·판단 규칙이 추가되는지 확인한다.

5. 중복 데이터는 학생번호 같은 공통 식별자를 기준으로 한 곳에서 관리하고 업무별 기록은 그 식별자로 연결한다. 파일마다 다른 열 이름·형식·코드 체계는 공통 스키마와 업무 규칙으로 통일하여 데이터 고립을 줄인다.

   통합은 모든 파일을 없애는 일이 아니라 공통 사실의 기준과 연결 규칙을 정하는 일이다.

실전문제 해설 및 예시 답안

동아리 회원 파일의 문제 찾기

- 가상 회원번호를 공통 식별자로 두고 회원 명단·회비 관리·참가 신청 파일의 이름, 학과, 전화번호를 비교한다. 같은 회원번호에서 값이 다르면 어느 파일이 최신인지 확인하고 불일치로 표시한다.
- 공통 데이터는 회원번호, 이름, 학과, 연락처처럼 여러 업무가 함께 사용하는 값으로 분리하고, 회비 납부·행사 참가처럼 업무별 사실은 별도 구조로 둔다. 연락처 수정은 권한 있는 한 경로에서 수행하고 변경 책임과 이력을 기록한다.
- 파일로 유지할 자료의 예는 원본 사진·문서처럼 파일 자체가 중요한 자료다. 검색·관계 연결·동시 수정·권한 통제가 필요한 회원·회비·참가 데이터는 데이터베이스 통합 대상으로 분류할 수 있다. 실제 개인정보 대신 가상 값만 사용한다.

Chapter 02

객관식 정답 및 해설

문항	정답	해설
1	②	실시간 접근성은 모든 요청이 0 초에 끝난다는 뜻이 아니라 업무가 요구하는 시간 안에 결과를 제공하는 성질이다.
2	③	스키마는 구조와 규칙, 인스턴스는 특정 시점에 실제 저장된 값의 집합이다.
3	②	저장 장치·파일 배치·인덱스 같은 내부 구조 변화가 상위 구조와 응용 프로그램에 미치는 영향을 줄이는 것은 물리적 데이터 독립성이다.
4	③	GRANT·REVOKE 와 같은 권한 부여·회수 명령은 이 책의 분류에서 DCL 에 속한다.
5	②	테이블·열·데이터 타입·제약조건·권한은 DBMS 가 관리하는 대표적인 메타데이터다.

단답형·서술형 예시 답안

1. 스키마(Schema).

   데이터베이스의 구조와 제약조건을 정의한 설계도다.

2. 외부 스키마(External Schema).

   특정 사용자나 응용 프로그램에 필요한 데이터 관점을 정의한다.

3. 시스템 카탈로그(System Catalog).

   DBMS 가 테이블·열·키·제약조건·권한 등의 메타데이터를 저장하는 내부 저장소다.

4. 수강 신청 데이터베이스는 학생·강좌·신청 기록처럼 관리 대상인 데이터의 집합이다. DBMS 는 그 데이터베이스를 만들고 조회·수정하며 권한·동시성·회복을 관리하는 소프트웨어다.

   데이터베이스=관리 대상, DBMS=관리 소프트웨어로 구분한다.

5. 물리적 데이터 독립성은 내부 저장 구조 변경의 영향을 줄이는 성질이다. 예를 들어 인덱스를 추가하거나 저장 장치를 바꿔도 기존 학생 조회 화면을 유지한다. 논리적 데이터 독립성은 개념 스키마 변경의 영향을 줄이는 성질이다. 예를 들어 속성을 추가하거나 테이블 구조를 조정해도 기존 외부 화면을 가능한 한 유지한다.

   일반적으로 논리적 데이터 독립성이 더 확보하기 어렵다.

실전문제 해설 및 예시 답안

대학 도서관 메타데이터 점검

- 구조 메타데이터 예시는 테이블명, 열 이름, 데이터 타입, 기본키, 외래키, NULL 허용, 제약조건이다. 이러한 정보는 DBMS 의 시스템 카탈로그에서 상당 부분 확인할 수 있다.
- 업무 메타데이터 예시는 “전공 도서”의 분류 기준, “연체”의 기준일과 계산 규칙, “대출 가능”의 상태 정의다. 이 값들은 테이블 구조만으로 완전히 알기 어렵고 업무 담당자의 공식 정의가 필요하다.
- 각 업무 용어에는 정의뿐 아니라 담당 부서, 보존 기간, 열람 권한을 함께 기록한다. 구조 메타데이터와 업무 의미를 구분해 관리해야 시스템과 사람이 같은 용어를 같은 뜻으로 해석할 수 있다.

Chapter 03

객관식 정답 및 해설

문항	정답	해설
1	①	차수는 속성 수이므로 5, 카디널리티는 현재 튜플 수이므로 12 다.
2	②	후보키는 튜플을 유일하게 식별하면서 불필요한 속성이 없는 유일성과 최소성을 모두 만족해야 한다.
3	③	존재하지 않는 부모 키를 외래키가 참조하면 참조 무결성을 위반한다.
4	②	필요한 속성만 선택하는 관계 대수 연산은 프로젝션이다.
5	③	합집합·교집합·차집합은 속성 수가 같고 대응 속성의 도메인이 호환되는 합병 가능성이 필요하다.

단답형·서술형 예시 답안

1. 차수(Degree).

   릴레이션이 가진 속성의 개수다.

2. 후보키(Candidate Key).

   유일성과 최소성을 모두 만족하는 슈퍼키다.

3. 참조 무결성(Referential Integrity).

   외래키가 존재하지 않는 부모 키를 참조했기 때문에 위반된다.

4. 예: STUDENT 의 student_id 는 각 학생 튜플을 유일하게 식별하는 기본키다. ENROLLMENT 의 student_id 는 STUDENT.student_id 를 참조하는 외래키이고 section_id 는 SECTION.section_id 를 참조한다. 이 외래키들이 수강신청 행을 실제 학생·분반과 연결한다.

   기본키는 식별, 외래키는 다른 릴레이션과의 참조 연결 역할이다.

5. 먼저 STUDENT 에서 dept_id='D01'인 학생을 셀렉션한다. 그 결과를 ENROLLMENT 와 student_id 로 조인한다. 마지막에 필요한 section_id 만 프로젝션한다.

질문에 특정 신청 상태까지 요구되면 조인 결과에 status 조건을 추가로 셀렉션한다.

실전문제 해설 및 예시 답안

대학 수강 관리 릴레이션 분석

- STUDENT 의 한 튜플은 한 학생, SECTION 의 한 튜플은 한 분반, ENROLLMENT 의 한 튜플은 한 학생의 한 분반 수강신청 사실을 나타낸다.
- 대표 기본키는 STUDENT.학생번호, SECTION.분반 ID 이며 ENROLLMENT 는 (학생번호, 분반 ID) 조합을 기본키로 사용할 수 있다. ENROLLMENT 의 학생번호와 분반 ID 는 각각 STUDENT 와 SECTION 을 참조하는 외래키다.
- "CS01 학과 학생이 신청 상태인 분반 ID"는 STUDENT 에서 학과코드=CS01 을 셀렉션하고 ENROLLMENT 와 학생번호로 조인한 뒤 신청상태=신청을 셀렉션하고 마지막에 분반 ID 를 프로젝션한다. 각 단계에서 남는 속성과 튜플 수를 기록해 검산한다.

Chapter 04

객관식 정답 및 해설

문항	정답	해설
1	②	개념적 모델링은 DBMS·인덱스보다 무엇을 관리하고 대상이 어떤 관계를 가지는지 업무 의미를 표현하는 데 초점을 둔다.
2	①	다중값 속성은 한 개체가 같은 종류의 값을 여러 개 가질 수 있는 속성이다.
3	③	학생 한 명이 여러 분반, 분반 하나에 여러 학생이 연결될 수 있으므로 M:N 이다.
4	②	1:N 관계는 1 쪽 기본키를 N 쪽 릴레이션의 외래키로 둔다.
5	③	M:N 관계는 양쪽 기본키를 외래키로 가진 별도 관계 릴레이션으로 변환한다.

단답형·서술형 예시 답안

1. 다중값 속성(Multivalued Attribute).

   한 학생이 같은 종류의 전화번호를 여러 개 가질 수 있기 때문이다.

2. 1:N 관계.

   한 학과에는 여러 학생이 있을 수 있고 각 학생은 하나의 주전공 학과에 소속된다는 최대 카디널리티다.

3. 별도의 관계 릴레이션(연관 릴레이션), 예: ENROLLMENT.

   양쪽 개체의 기본키를 외래키로 포함하여 M:N 관계 한 건을 표현한다.

4. 카디널리티는 상대편 개체와 최대 몇 개까지 연결될 수 있는지를 나타낸다. 참여 제약조건은 최소 참여가 0 인지 1 인지처럼 관계 참여가 선택인지 필수인지 나타낸다. 예를 들어 한 학과에는 학생이 0 명 이상 있을 수 있지만 한 학생은 하나의 주전공 학과에 반드시 소속된다는 규칙으로 읽을 수 있다.

   최대 수와 최소 참여 수를 양쪽 방향으로 읽는다.

5. 다중값 속성을 한 셀이나 반복 열에 저장하면 개별 값 검색·중복 검사·수정이 어렵고 1NF 의 취지에도 맞지 않으므로 별도 릴레이션으로 분리한다. M:N 관계는 어느 한쪽에 외래키 하나만 두어 양쪽의 여러 연결을 표현할 수 없으므로 두 키를 가진 별도 관계 릴레이션이 필요하다.

   분리 뒤 한 행이 하나의 사실을 나타내고 원래 관계를 재구성할 수 있어야 한다.

실전문제 해설 및 예시 답안

축제 물품 대여 모델 확장

- 가능한 새 개체로 RESERVATION(예약)과 RETURN_INSPECTION(반납점검)을 둘 수 있다. 예약은 reservation_id, club_id, item_type_id 또는 asset_id, reserved_at, status 등을 가질 수 있고, 반납점검은 inspection_id, loan_id 또는 loan_item_id, inspected_at, condition_result 등을 가질 수 있다.
- 관계의 최소·최대 수는 실제 업무 규칙으로 정한다. 예를 들어 한 동아리는 여러 예약을 만들 수 있고 각 예약은 한 동아리에 속하도록 CLUB 1:N RESERVATION 으로 설계할 수 있다. 예약이 특정 개별자산을 미리 지정하는지 물품종류만 지정하는지는 업무 범위에 따라 결정한다.

- 반납점검을 대여항목당 정확히 한 번만 허용한다면 RETURN_INSPECTION.loan_item_id 에 UNIQUE 를 두는 1:1 구조를 검토한다. 여러 번 점검을 허용한다면 inspection_id 를 기본키로 두고 LOAN_ITEM 1:N RETURN_INSPECTION 으로 설계한다. 먼저 규칙을 문장으로 확정한 뒤 PK·FK 에 반영한다.

Chapter 05

객관식 정답 및 해설

문항	정답	해설
1	②	MySQL Server 가 데이터를 저장하고 SQL 을 실행하며 Workbench 는 서버 접속·SQL 편집·결과 확인을 돕는 클라이언트 도구다.
2	③	STUDENT.dept_id 외래키는 DEPARTMENT 에 존재하지 않는 학과코드를 참조하는 오류를 막는다.
3	②	정확한 소수 둘째 자리까지 보존해야 하는 점수에는 DECIMAL(5,2)이 적합하다.
4	③	TRUNCATE TABLE 은 구조를 남기고 모든 행을 제거하는 DDL 이며 MySQL 에서는 암시적 커밋이 발생한다.
5	③	안전한 변경은 사전 SELECT→예상 행 수→UPDATE→영향받은 행→사후 SELECT 순서로 검증한다.

단답형·서술형 예시 답안

1. DECIMAL(5,2).

   0.00~100.00 같은 점수를 정확한 소수 자릿수로 저장하는 데 적합하다.

2. SHOW CREATE TABLE 테이블명;

   열뿐 아니라 CHECK·외래키·문자 집합·저장 엔진 등 실제 생성 정의를 확인한다.

3. NULL.

   값이 아직 정해지지 않았거나 알 수 없음 등을 나타내며 숫자 0·빈 문자열과 다르다.

4. 외래키의 자식 값은 참조할 부모 키가 실제로 존재해야 한다. 따라서 부모 테이블을 먼저 만들고 부모 행을 먼저 입력해야 참조 무결성을 만족하며 자식 CREATE/INSERT 가 실패하지 않는다.

   외래키 의존 순서가 생성·입력 순서를 결정한다.

5. 사전 SELECT 로 같은 WHERE 가 정확한 대상 행을 가리키는지 확인하고 예상 행 수를 정한다. UPDATE 뒤 Action Output 의 영향받은 행이 예상과 같은지 확인하며, 사후 SELECT 로 실제 값이 의도대로 바뀌었는지 검산한다.

   문법이 맞아도 WHERE 범위가 틀리면 여러 행을 잘못 바꿀 수 있기 때문이다.

실전문제 해설 및 예시 답안

AI 가 만든 변경 SQL 의 안전성 감사

- 학생번호 1005 한 행만 변경하는지 먼저 같은 WHERE 조건의 SELECT 로 확인하고 예상 행 수를 1 로 기록한다. 그다음 같은 WHERE 를 그대로 UPDATE 에 사용하고 영향받은 행 수와 사후 값을 확인한다. 마지막에는 원래 값으로 복원하거나 트랜잭션이면 ROLLBACK 한다.
- 안전성 감사에서 WHERE 없는 UPDATE, DROP, TRUNCATE, 존재하지 않는 열, 예상보다 넓은 조건이 포함되면 실행하지 않는다.

```
SELECT student_id, active
FROM STUDENT
WHERE student_id = 1005;
START TRANSACTION;
UPDATE STUDENT
SET active = TRUE
WHERE student_id = 1005;
SELECT ROW_COUNT() AS changed_rows;
SELECT student_id, active
FROM STUDENT
WHERE student_id = 1005;
ROLLBACK;
```

Chapter 06

객관식 정답 및 해설

문항	정답	해설
1	①	SELECT 절이 결과에 표시할 열과 표현식을 정한다.
2	②	AND 가 OR 보다 먼저 평가되므로 두 학과 조건을 괄호로 묶은 뒤 grade>=2 를 AND 로 적용해야 한다.
3	③	DISTINCT 는 SELECT 목록 전체인 dept_id 와 grade 의 값 조합이 같은 결과 행을 중복 제거한다.
4	③	OFFSET 3 은 앞 3 행을 건너뛰므로 네 번째 행부터 최대 3 행을 반환한다.
5	③	NULL 여부는 일반 등호가 아니라 IS NULL 또는 IS NOT NULL 로 검사한다.

단답형·서술형 예시 답안

1. `email IS NULL`

   NULL 은 `= NULL`로 비교하지 않는다.

2. DISTINCT

   `SELECT DISTINCT dept_id, grade ...`처럼 SELECT 목록 전체 조합의 중복을 제거한다.

3. `LIMIT 3 OFFSET 3`

   정렬을 먼저 고정한 뒤 앞의 3 행을 건너뛰고 최대 3 행을 가져온다.

4. SQL 에서는 AND 가 OR 보다 먼저 평가된다. 따라서 `(dept_id='D01' OR dept_id='D02') AND grade>=2`처럼 의미 단위를 괄호로 묶지 않으면 D01 조건과 D02+학년 조건이 서로 다르게 적용되어 의도하지 않은 행이 포함될 수 있다.

   괄호는 사람이 의도한 논리 그룹을 명확히 한다.

5. NULL 은 값이 없거나 알려지지 않은 상태를 나타내는 특별한 표시이며 일반 비교 결과가 UNKNOWN 이 될 수 있다. 0 은 실제 숫자 0 이고, 빈 문자열은 길이가 0 인 실제 문자열 값이다.

   NULL 여부는 IS NULL/IS NOT NULL 로 검사하고 0 과 빈 문자열은 각 자료형의 실제 값으로 비교한다.

실전문제 해설 및 예시 답안

공통 학생 데이터 조건 검색

- ① D01 학과 학생은 1001, 1002, 1008 의 3 명이다. ② 1~2 학년이면서 D01 또는 D03 인 학생은 1001, 1002, 1007 의 3 명이다. ③ email IS NULL 은 1003, 1008 의 2 행이고 IS NOT NULL 은 8 행이므로 합계가 전체 10 행과 일치한다.
- 각 SQL 은 실행 전에 예상 행 수를 적고, AND/OR 괄호가 의미 단위를 보존하는지와 NULL 을 IS NULL 로 검사했는지 확인한다.

```
SELECT student_id, student_name, grade
FROM STUDENT
WHERE dept_id = 'D01'
ORDER BY student_id;
SELECT student_id, student_name, dept_id, grade
FROM STUDENT
WHERE grade BETWEEN 1 AND 2
  AND (dept_id = 'D01' OR dept_id = 'D03')
ORDER BY student_id;
SELECT COUNT(*) AS null_email_rows
FROM STUDENT
WHERE email IS NULL;
SELECT COUNT(*) AS not_null_email_rows
FROM STUDENT
WHERE email IS NOT NULL;
```

Chapter 07

객관식 정답 및 해설

문항	정답	해설
1	②	CHAR_LENGTH 는 문자 개수를 반환하므로 한글 이름의 글자 수를 구하는 데 적합하다.
2	②	COUNT(score)는 NULL 이 아닌 score 만 세므로 9 다.
3	③	CASE 는 위에서부터 WHEN 을 평가하여 처음 TRUE 가 된 분기의 값을 반환한다.
4	①	GROUP BY 전에 원본 행을 제한하는 조건은 WHERE 에 둔다.
5	③	AVG(score)처럼 집계 결과에 대한 조건은 HAVING 에 둔다.

단답형·서술형 예시 답안

1. CHAR_LENGTH().

   문자의 개수를 반환한다.

2. 9.

   NULL 이 아닌 score 행이 9 개이므로 COUNT(score)=9 다.

3. HAVING.

   GROUP BY 뒤의 집계 결과에 조건을 적용한다.

4. COUNT(*)는 NULL 여부와 관계없이 행을 센다. COUNT(score)는 score 가 NULL 이 아닌 행만 센다. COUNT(DISTINCT score)는 NULL 을 제외하고 서로 다른 score 값만 센다.

   COUNT 의 분모가 질문의 단위와 맞는지 확인한다.

5. WHERE 는 `status='완료'`처럼 집계 전에 원본 행을 먼저 제한한다. GROUP BY 로 학생별 그룹을 만든 뒤 HAVING 은 `AVG(score)>=90`처럼 계산된 집계 결과를 제한한다.

   WHERE 와 HAVING 의 위치가 바뀌면 평균의 대상 행과 결과 그룹이 달라질 수 있다.

실전문제 해설 및 예시 답안

AI 가 만든 상태별 집계의 분모 검증

- 상태별 COUNT(*)는 전체 수강 행 수, COUNT(score)는 점수가 입력된 행 수, AVG(score)는 점수 입력 행의 평균을 뜻한다. 공통 데이터 기준 완료 9/9/88.50, 수강 7/0/NULL, 신청 1/0/NULL, 취소 1/0/NULL 이다.
- WHERE score IS NOT NULL 을 GROUP BY 전에 넣으면 점수가 없는 상태 행이 아예 제거되어 질문의 분모가 달라진다. 상태별 전체 행을 보존하려면 GROUP BY 뒤 COUNT(score)와 AVG(score)로 NULL 의 영향을 확인한다.

```
SELECT status,
       COUNT(*) AS total_rows,
       COUNT(score) AS scored_rows,
       ROUND(AVG(score), 2) AS score_avg
FROM ENROLLMENT
GROUP BY status
ORDER BY status;
```

Chapter 08

객관식 정답 및 해설

문항	정답	해설
1	④	조인 조건 없는 카티션 곱의 후보 행 수는 10×18=180 이다.
2	②	완료 수강만 오른쪽에서 연결하면서 모든 학생을 보존하려면 완료 조건을 LEFT JOIN 의 ON 에 둔다.
3	①	관련 행이 하나라도 존재하는지만 검사할 때 EXISTS 가 적합하다.
4	③	첫 번째 결과에만 있는 행은 EXCEPT 로 구한다.
5	②	CTE 는 WITH 절에서 이름을 붙여 한 SQL 문장 범위에서 사용하는 중간 결과다.

단답형·서술형 예시 답안

1. 180 행.

   10×18=180 이다.

2. LEFT JOIN(LEFT OUTER JOIN).

   왼쪽 STUDENT 의 모든 행을 보존한다.

3. EXISTS.

   관련 행의 존재 여부만 검사할 때 적합하다.

4. 오른쪽 테이블 조건을 ON 에 두면 조건에 맞는 오른쪽 행만 연결하면서 왼쪽 행은 보존된다. 같은 조건을 WHERE 에 두면 조인 뒤 오른쪽 값이 NULL 인 보존 행이 제거되어 결과가 내부 조인처럼 줄어들 수 있다.

   LEFT JOIN 에서 조건 위치는 결과 보존 범위를 바꾼다.

5. 기본키·외래키의 실제 연결 경로를 확인하지 않으면 잘못된 열로 조인하거나 조건을 빠뜨려 카티션 곱, 중복 증폭, 누락이 생길 수 있다. 단계별 중간 행 수와 한 행의 의미를 기록하면 최종 결과가 질문 단위와 맞는지 검산할 수 있다.

   특히 여러 테이블 조인은 조인 하나가 틀려도 최종 결과가 그럴듯하게 보일 수 있다.

실전문제 해설 및 예시 답안

AI 생성 다중 조인 SQL 의 회귀 검증

- 결과 한 행의 의미를 “2026-2 학기의 완료 수강 한 건”으로 정의한다. 조인 경로는 ENROLLMENT→STUDENT→DEPARTMENT 와 ENROLLMENT→SECTION→COURSE 이며 각각 기본키·외래키로 연결한다.
- WHERE se.semester='2026-2' AND e.status='완료'를 적용하면 공통 데이터 기준 8 행이다. 2027-1 의 section_id=5010 완료 행은 제외된다. 중복 검산 키는 (student_id, section_id)다.

```
SELECT e.student_id, s.student_name, d.dept_name,
       c.course_name, e.score
FROM ENROLLMENT AS e
JOIN STUDENT AS s ON s.student_id = e.student_id
JOIN DEPARTMENT AS d ON d.dept_id = s.dept_id
JOIN SECTION AS se ON se.section_id = e.section_id
JOIN COURSE AS c ON c.course_id = se.course_id
WHERE se.semester = '2026-2'
  AND e.status = '완료'
ORDER BY e.student_id, e.section_id;
```

Chapter 09

객관식 정답 및 해설

문항	정답	해설
1	②	중복된 학과명 일부만 수정해 값이 달라지는 문제는 수정 이상이다.
2	②	복합키 전체가 아니라 student_id 일부만으로 student_name 이 결정되므로 부분 함수 종속이다.
3	②	1NF 는 반복 그룹을 제거하고 각 속성값을 하나의 원자적인 도메인 값으로 다룬다.
4	②	BCNF 는 모든 비자명 함수 종속에서 결정자가 슈퍼키여야 한다.
5	③	역정규화는 측정된 병목과 일관성 통제·동기화 책임이 있을 때 제한적으로 적용한다.

단답형·서술형 예시 답안

1. 부분 함수 종속.

   복합 후보키 전체가 아닌 student_id 일부만으로 student_name 이 결정된다.

2. 보이스-코드 정규형(BCNF).

   모든 비자명 함수 종속의 결정자가 슈퍼키여야 한다.

3. 제 1 정규형(1NF).

   반복 그룹을 제거하고 각 속성값을 원자값으로 다룬다.

4. 부분 함수 종속은 복합 후보키의 일부가 비키 속성을 결정하는 경우다. 예를 들어 (student_id, section_id)가 키인데 student_id→student_name 이 성립하면 부분 종속이다. 이행 함수 종속은 키가 다른 비키 속성을 거쳐 비키 속성을 결정하는 경우다. 예를 들어 student_id→dept_id, dept_id→dept_name 이면 student_id 가 dept_name 을 간접적으로 결정한다.

   2NF 는 부분 종속, 3NF 는 이행 종속을 중심으로 제거한다.

5. 무손실 분해를 확인해야 분해한 릴레이션을 다시 조인했을 때 원래 업무 사실을 잃거나 가짜 튜플이 생기지 않는다. 종속성 보존도 함께 검토해야 원래의 중요한 업무 규칙을 복잡한 재조인 없이 각 릴레이션의 제약으로 검사할 수 있다.

   분해가 정규형만 만족하고 업무 규칙을 검증하기 어려워지는 상황을 피한다.

실전문제 해설 및 예시 답안

수강 요약 관계의 1NF·2NF·3NF 분해

- 후보키를 (student_id, section_id)로 두고 contact_list 처럼 한 셀에 여러 전화번호가 들어 있는 반복 그룹은 STUDENT_PHONE(student_id, phone_number, …)으로 분리하여 1NF 를 만족시킨다.
- 2NF 에서는 복합키 일부에만 의존하는 student_id→student_name, dept_id 와 section_id→course_id, semester 같은 부분 종속을 STUDENT·SECTION 등으로 분리한다. 3NF 에서는 dept_id→dept_name, course_id→course_name 처럼 비키 속성을 거쳐 발생하는 이행 종속을 DEPARTMENT·COURSE 등으로 분리한다.
- 최종 구조의 각 릴레이션에 PK·FK 를 지정하고 대표 student_id 와 section_id 를 재조인했을 때 원래 학생–분반 수강 사실과 필요한 속성이 복원되는지 확인한다. 원래보다 새로운 가짜 조합이 생기지 않아야 한다.

Chapter 10

객관식 정답 및 해설

문항	정답	해설
1	②	일반적인 뷰는 SELECT 정의에 이름을 붙여 기본 테이블처럼 조회하는 가상 테이블 성격의 객체다.
2	②	복합 인덱스 (status, score)의 첫 번째 왼쪽 접두부는 status 다.
3	①	뷰 생성 명령은 CREATE VIEW 다.
4	①	테이블 인덱스의 이름·열 순서 등을 확인하는 대표 명령은 SHOW INDEX 다.
5	③	SELECT 의 실행 계획은 EXPLAIN 으로 확인한다.

단답형·서술형 예시 답안

1. 뷰(View).

   자주 쓰는 SELECT 정의에 이름을 붙여 조회하는 객체다.

2. key.

   EXPLAIN 에서 옵티마이저가 실제 선택한 인덱스 이름을 나타낸다.

3. SHOW INDEX FROM 테이블명;

   인덱스 이름·유일성·열 순서 등을 확인한다.

4. 기본 테이블은 실제 행 데이터를 저장하지만 일반적인 뷰는 SELECT 정의를 저장해 결과를 보여 준다. `SELECT *`로 뷰를 만들면 기본 테이블에 열이 추가되거나 순서가 바뀔 때 뷰가 의도하지 않은 열을 노출하거나 의존성이 커질 수 있으므로 필요한 열을 명시하는 편이 안전하다.

   뷰의 열 범위는 보안·호환성에도 영향을 준다.

5. 인덱스는 검색 범위를 줄여 조회에 도움을 줄 수 있지만 INSERT·UPDATE·DELETE 때 해당 인덱스 항목도 함께 생성·수정·삭제해야 한다. 따라서 인덱스가 많을수록 쓰기 비용과 저장 공간, 유지 관리 비용이 증가한다.

   조회 이득과 쓰기 비용을 실제 워크로드로 비교해야 한다.

실전문제 해설 및 예시 답안

뷰·인덱스·실행 계획 확인

- 활동 학생만 노출하는 V_ACTIVE_STUDENT 는 공통 데이터 기준 9 행이다. 뷰에는 필요한 열만 명시한다.
- ENROLLMENT(status, score) 인덱스를 만든 뒤 SHOW INDEX 로 열 순서를 확인하고, `status='완료' AND score>=90` 대표 SELECT 의 EXPLAIN 을 생성 전후로 비교한다. 실제 결과는 4 행이며 점수는 95, 93, 92, 91 이다.

- 실습 데이터가 작으면 옵티마이저가 인덱스를 선택하지 않을 수도 있다. 이 경우 특정 key 가 선택되지 않았다는 사실을 오류로 단정하지 말고 데이터 규모·선택도·통계 상태를 기록한다. 실습 후 자신이 만든 객체만 제거한다.

Chapter 11

객관식 정답 및 해설

문항	정답	해설
1	②	기본 자동 커밋 상태에서 명시적 트랜잭션 밖의 각 UPDATE 는 성공하면 문장 단위로 각각 커밋된다.
2	③	ROLLBACK TO SAVEPOINT 는 저장점 이후만 취소하고 현재 트랜잭션은 계속된다.
3	②	다른 트랜잭션의 아직 커밋되지 않은 값을 읽는 현상은 오손 읽기다.
4	②	같은 행을 다시 읽었을 때 다른 커밋 때문에 값이 달라지는 현상은 반복 불가능 읽기다.
5	②	MySQL 의 많은 DDL 은 실행 전후 암시적 커밋이 발생할 수 있다.

단답형·서술형 예시 답안

1. COMMIT.

   현재 트랜잭션의 변경을 확정한다.

2. ROLLBACK TO SAVEPOINT 저장점명;

   트랜잭션 전체가 아니라 지정 저장점 이후의 변경만 취소한다.

3. 오손 읽기(Dirty Read).

   다른 트랜잭션의 미커밋 값을 읽는 현상이다.

4. 수강 변경에서 기존 신청 취소와 새 신청 등록이 함께 성공해야 한다면 하나의 트랜잭션으로 묶는다. 원자성은 전부 성공하거나 전부 취소되는 성질, 일관성은 변경 전후에도 수강 규칙을 지키는 성질, 격리성은 동시에 실행되는 다른 수강 변경과의 간섭을 통제하는 성질, 지속성은 COMMIT 된 변경이 장애 뒤에도 유지되는 성질이다.

   ACID 는 각각 독립된 표어가 아니라 안전한 상태 전환을 함께 설명한다.

5. 격리 수준을 높이면 오손 읽기·반복 불가능 읽기·팬텀과 같은 이상을 줄일 수 있지만 잠금 대기, 충돌, 처리량 저하가 커질 수 있다. 업무가 반드시 지켜야 할 정확성 수준과 동시에 처리해야 할 요청량을 함께 고려해 필요한 격리 수준과 잠금 전략을 선택한다.

   가장 높은 격리 수준이 모든 업무에서 가장 좋은 것은 아니다.

실전문제 해설 및 예시 답안

수강 변경 트랜잭션과 부분 롤백 확인

- 1007 번 학생의 section_id=5006 초기 상태는 신청, score=NULL 이다. START TRANSACTION 뒤 첫 변경을 수행하고 SAVEPOINT 를 만든 뒤 두 번째 변경을 수행한다. ROLLBACK TO 는 저장점 이후 변경만 취소하고 트랜잭션은 계속된다.
- 마지막 전체 ROLLBACK 뒤에는 status 가 다시 신청이고 score 가 NULL 이어야 한다. 각 단계의 SELECT 결과와 예상 상태를 표로 비교한다.

```
SELECT student_id, section_id, status, score
FROM ENROLLMENT
WHERE student_id = 1007 AND section_id = 5006;
START TRANSACTION;
UPDATE ENROLLMENT
SET status = '수강'
WHERE student_id = 1007 AND section_id = 5006;
SAVEPOINT s1;
UPDATE ENROLLMENT
SET status = '완료', score = 100.00
WHERE student_id = 1007 AND section_id = 5006;
ROLLBACK TO s1;
SELECT status, score
FROM ENROLLMENT
WHERE student_id = 1007 AND section_id = 5006;
ROLLBACK;
```

Chapter 12

객관식 정답 및 해설

문항	정답	해설
1	②	인증된 주체가 어떤 작업을 수행할 수 있는지 결정하는 것은 인가다.
2	②	여러 권한을 직무별 묶음으로 관리하는 객체가 역할(Role)이다.
3	②	외부 입력값은 SQL 구조와 분리해 준비된 문장의 값 매개변수로 바인딩하는 것이 기본이다.
4	①	원본 값은 유지하고 화면 표시 일부만 가리는 처리는 마스킹이다.
5	③	백업이 실제로 쓸 수 있는지는 격리 환경에 복원하여 객체·행 수·업무 값을 검증해야 가장 직접적으로 확인된다.

단답형·서술형 예시 답안

1. 인증(Authentication).

   접속 주체가 누구인지 확인하는 과정이다.

2. 인가(Authorization).

   인증된 주체가 어떤 작업을 할 수 있는지 결정한다.

3. 마스킹(Masking).

   원본 저장값을 유지한 채 화면 표시의 일부를 가린다.

4. 값 매개변수 바인딩은 외부 값을 SQL 구조와 분리해 값으로만 해석되게 한다. 테이블명·정렬 열 같은 식별자는 값 매개변수로 바인딩할 수 없으므로 허용 목록으로 제한한다. 최소 권한은 방어가 우회되거나 SQL 이 잘못 생성되더라도 읽기·변경 가능한 범위를 줄인다.

   세 방어는 서로 다른 공격 경로를 막기 때문에 함께 사용한다.

5. 백업 명령의 성공 메시지와 파일 존재만으로는 파일이 손상되지 않았는지, 필요한 객체·행·권한을 실제로 복원할 수 있는지 알 수 없다. 격리 환경에 복원해 구조, 행 수, 주요 합계, 업무 시나리오를 검증해야 복구 가능한 백업임을 증명할 수 있다.

   복원 시험은 백업 절차의 일부다.

실전문제 해설 및 예시 답안

읽기 역할과 최소 권한의 허용·거부 시험

- 폐기 가능한 로컬 실습 서버에서 V_ACTIVE_STUDENT 의 정의와 보안 속성을 확인한다. `Chapter_12.sql`은 02 초기화 뒤에도 실습할 수 있도록 이 뷰를 준비하는 안내를 포함한다.
- 보고서 계정과 읽기 역할에는 V_ACTIVE_STUDENT 의 SELECT 만 부여한다. CURRENT_ROLE()과 SHOW GRANTS ... USING 으로 유효 권한을 확인한다. 뷰 SELECT 는 성공해야 하고 기본 STUDENT 직접 SELECT, UPDATE, DDL 은 권한 오류로 거부되어야 한다.
- 실습 종료 뒤 역할 부여와 역할 권한을 회수하고 실습 사용자와 역할을 제거한다. 잔존 계정/역할 수가 0 인지 확인한다. 운영 서버나 실제 계정에서는 이 실습을 수행하지 않는다.

Chapter 13

객관식 정답 및 해설

문항	정답	해설
1	④	노드와 관계, 경로 탐색이 중심인 데이터는 그래프 데이터베이스가 적합하다.
2	①	정답 레이블이 있는 과거 데이터를 학습해 새로운 사례의 범주를 예측하는 작업은 분류다.
3	①	임베딩 검색은 벡터 사이의 유사도를 비교해 의미적으로 가까운 후보를 찾는다.
4	②	RAG는 검색 실패와 생성 실패의 원인·개선 방법이 다르므로 검색 결과와 생성 답변을 분리해 평가해야 한다.

문항	정답	해설
5	②	AI 생성 UPDATE는 대상·영향·트랜잭션·되돌리기와 사람 승인을 확인한 뒤 실행해야 한다.

단답형·서술형 예시 답안

1. 그래프 데이터베이스(Graph Database).

   노드와 관계를 저장하고 관계 경로 탐색을 중심으로 사용한다.

2. 분류(Classification).

   정답 레이블이 있는 과거 데이터를 학습해 새로운 사례의 범주를 예측한다.

3. 유사도(Similarity).

   코사인 유사도 등 선택한 지표로 임베딩 벡터의 의미적 가까움을 비교한다.

4. 정답 문서가 검색되지 않은 검색 실패와 정답 문서를 찾았지만 다른 답을 만든 생성 실패는 원인과 수정 대상이 다르다. 검색 후보·점수와 생성 문장·인용을 분리해 보면 검색기, 분할·임베딩, 재순위화, 생성 프롬프트 중 어디를 고쳐야 하는지 판단할 수 있다.

   권한 실패도 품질 실패와 별도로 분리해야 한다.

5. Text-to-SQL은 자연어를 그럴듯한 SQL로 바꿀 수 있지만 실제 스키마에 없는 열, 잘못된 조인·필터, 과도한 범위, DML·DDL을 만들 수 있다. 따라서 승인 스키마와 권한, 대상 행·예상 결과, EXPLAIN·비용, 트랜잭션·되돌리기와 사람 승인을 확인해 실행 위험을 통제해야 한다.

   생성과 실행 인터페이스를 분리하는 것이 안전하다.

실전문제 해설 및 예시 답안

문서 검색과 구조화 조회의 처리 경로 설계

- 정확한 현재 값을 묻는 질문은 SQL/API 경로, 규정·설명처럼 문서 근거가 필요한 질문은 RAG 경로, 현재 값과 규정을 함께 묻는 질문은 SQL+RAG 혼합 경로로 분류한다.
- 예: “현재 완료한 고유 학생은 몇 명인가?”는 읽기 전용 SQL에서 COUNT(DISTINCT student_id)로 계산하고 기준 시점을 표시한다. “수강 철회 규정은 무엇인가?”는 출처·버전·접근 권한이 검증된 현재 문서를 검색해 문단 근거를 제시한다. “현재 완료 학생 수와 철회 규정을 함께 설명해 달라”는 두 경로의 결과를 각각 검증한 뒤 하나의 응답에서 근거와 시점을 분리해 표시한다.
- RAG 문서에는 source_id, version, 권한과 검색 시각을 기록한다. 관련 문서가 없거나 최신 문서가 충돌하면 추측하지 않고 근거 부족/확인 필요로 처리한다.