



Episode 3: 시스템 수호국 — 데이터의 질서를 지키는 자

데이터의 안정성과 무결성을 지키는 헌법 수호자들

도시의 질서를 지키는 시스템 수호국은
데이터의 안정성과 보안을 책임졌어요.
그들은 보이지 않는 곳에서 도시의 기반을 지키며,
모든 정보가 정확하고 일관되게 유지되도록 힘썼답니다.

왜 필요한가?

데이터는 도시의 생명줄이자 가장 민감한 자산이에요.

- 잘못 다루면 손실, 오류, 보안 문제까지 생길 수 있어요.
- 시스템 수호국은 데이터의 무결성과 안정성을 지키는 마지막 방어선입니다.

이들은 단순한 관리자들이 아니라, 데이터 도시의 헌법을 집행하는 수호자들입니다.

트랜잭션의 4대 원칙 (ACID)

원칙	설명
Atomicity	작업은 전부 성공하거나 전부 실패해야 함
Consistency	데이터는 항상 일관된 상태 유지
Isolation	동시에 실행돼도 결과는 동일해야 함
Durability	커밋된 데이터는 영구 저장

 **알기 포인트:** 트랜잭션 흐름, COMMIT/ROLLBACK 시점, 격리 수준이 자주 출제돼요.

🔒 트랜잭션 격리 수준

격리 수준	설명	방지되는 현상
READ UNCOMMITTED	커밋되지 않은 데이터 읽기 가능	Dirty Read ❌
READ COMMITTED	커밋된 데이터만 읽음	Dirty Read 방지
REPEATABLE READ	같은 행 반복 조회 시 값 유지	Non-repeatable Read 방지
SERIALIZABLE	가장 엄격, 트랜잭션 직렬화	Phantom Read 방지

💡 **알기 포인트:** Dirty → Non-repeatable → Phantom 순으로 격리 강화

🔍 인덱스 감시탑

항목	설명
구조	B-Tree 기반
장점	검색 속도 향상, 정렬 최적화
단점	INSERT/UPDATE 시 성능 저하 가능성
종류	단일 인덱스, 복합 인덱스, 클러스터 인덱스

EXPLAIN SELECT * FROM 고객 WHERE 이름 = '홍길동';

💡 **알기 포인트:** 인덱스는 실행계획에서 사용 여부 확인 가능

📄 뷰(View) 창문

항목	설명
정의	SELECT 결과를 저장한 가상 테이블
장점	보안 강화, 복잡한 쿼리 단순화, 재사용
제한사항	일부 뷰는 수정 불가 (읽기 전용)
업데이트 가능 조건	단일 테이블 기반, 집계 함수 없음, GROUP BY 없음, JOIN 없음, DISTINCT 없음

💡 **알기 포인트:** 뷰의 정의, 장단점, 수정 가능 여부가 자주 출제돼요.

📄 제약조건 법령

제약조건	설명
PRIMARY KEY	고유 식별자, 중복/NULL 불가
FOREIGN KEY	다른 테이블 참조, 참조 무결성 유지
UNIQUE	중복 불가, NULL 허용
NOT NULL	반드시 값이 있어야 함
CHECK	조건 만족 여부 검사 (예: 급여 > 0)

```
CREATE TABLE 고객 (  
  고객 ID INT PRIMARY KEY,  
  이메일 VARCHAR(100) UNIQUE,  
  이름 VARCHAR(50) NOT NULL  
);
```

💡 **알기 포인트:** 제약조건은 CREATE 또는 ALTER로 설정

✂ 무결성의 수호자들

유형	설명	관련 제약조건
도메인 무결성	속성 값이 정의된 범위 내에 있어야 함	NOT NULL, CHECK
엔터티 무결성	기본키는 NULL이거나 중복되면 안 됨	PRIMARY KEY
참조 무결성	외래키는 반드시 부모 테이블의 값과 일치해야 함	FOREIGN KEY

💡 **알기 포인트:** 무결성은 데이터 품질을 지키는 헌법 수호자

📌 기억하기 쉬운 요약 정리

- 트랜잭션: 작업 묶음
- ACID: 안정성의 4대 원칙
- 격리 수준: Dirty → Non-repeatable → Phantom
- 인덱스: 검색 속도 향상
- 실행계획: 인덱스 사용 여부 확인
- 뷰(View): 가상 테이블
- 뷰 수정 조건: 단순 SELECT만 가능
- 제약조건: 데이터 규칙
- 무결성: 품질 보장
- 제약조건 적용 시점: CREATE 또는 ALTER

시스템 수호국 알기카드

카드 46

앞면: 트랜잭션이란?

뒷면: 작업의 논리적 단위

설명: 여러 작업을 하나로 묶어 처리

 암기 포인트: '트랜잭션 = 묶음 처리'

카드 47

앞면: ACID 중 Atomicity란?

뒷면: 전부 성공 또는 전부 실패

설명: 중간 실패 시 전체 취소

 암기 포인트: 'Atomic = 올 or Nothing'

카드 48

앞면: ACID 중 Consistency란?

뒷면: 데이터의 일관성 유지

설명: 규칙 위반 없이 상태 유지

 암기 포인트: 'Consistency = 규칙 지키기'

카드 49

앞면: ACID 중 Isolation이란?

뒷면: 동시에 실행돼도 결과 동일

설명: 트랜잭션 간 간섭 방지

 암기 포인트: 'Isolation = 독립 실행'

카드 50

앞면: ACID 중 Durability란?

뒷면: 커밋된 데이터는 영구 저장

설명: 장애 발생해도 유지됨


 암기 포인트: 'Durability = 저장 보장'

카드 51

앞면: 트랜잭션 격리 수준이란?

뒷면: 동시에 실행되는 트랜잭션 간 간섭 방지 단계

설명: READ COMMITTED, SERIALIZABLE 등


 암기 포인트: '격리 수준 = 충돌 방지 단계'

카드 52

앞면: COMMIT의 기능은?

뒷면: 변경 내용 저장

설명: 트랜잭션 완료

 암기 포인트: 'COMMIT = 저장'

카드 53

앞면: ROLLBACK의 기능은?

뒷면: 변경 내용 취소

설명: 트랜잭션 되돌리기

 암기 포인트: 'ROLLBACK = 취소'

카드 54

앞면: 인덱스의 주요 목적은?

뒷면: 검색 속도 향상

설명: B-Tree 구조로 빠른 탐색

 암기 포인트: '인덱스 = 검색 가속기'

카드 55

앞면: 인덱스의 단점은?

뒷면: 쓰기 성능 저하

설명: INSERT/UPDATE 시 부하 발생

 암기 포인트: '인덱스 = 읽기 빠름, 쓰기 느림'

카드 56

앞면: 실행계획(EXPLAIN)의 역할은?

뒷면: SQL 실행 흐름과 인덱스 사용 여부 확인

설명: 성능 분석에 활용


 암기 포인트: 'EXPLAIN = 실행 흐름 보기'

카드 57

앞면: 뷰(View)란?

뒷면: SELECT 결과를 저장한 가상 테이블

설명: 실제 데이터는 저장하지 않음

 암기 포인트: '뷰 = 가상의 창문'

카드 58

앞면: 뷰의 장점은?

뒷면: 보안 강화, 쿼리 단순화

설명: 복잡한 쿼리를 감추고 재사용 가능


 암기 포인트: '뷰 = 보안 + 재사용'

카드 59

앞면: 뷰(View)의 수정 가능 조건은?

뒷면: 단일 테이블 기반, 집계/JOIN 없음

설명: 복잡한 뷰는 읽기 전용


 암기 포인트: '뷰 수정 = 단순 SELECT만 가능'

카드 60

앞면: 제약조건은 언제 설정하나요?

뒷면: 테이블 생성 시 또는 ALTER로 추가

설명: CREATE TABLE 또는 ALTER TABLE 사용


 암기 포인트: '제약조건 = 생성 또는 수정 시 설정'

카드 61

앞면: PRIMARY KEY의 특징은?

뒷면: 중복/NULL 불가

설명: 테이블의 고유 식별자

 암기 포인트: 'PK = 고유 이름표'

카드 62

앞면: FOREIGN KEY의 역할은?

뒷면: 참조 무결성 유지

설명: 다른 테이블의 PK를 참조

 암기 포인트: 'FK = 연결 다리'

카드 63

앞면: UNIQUE 제약조건이란?

뒷면: 중복 불가, NULL 허용

설명: 이메일 등 고유값에 사용

 암기 포인트: 'UNIQUE = 중복 ❌, NULL 가능'

카드 64

앞면: NOT NULL 제약조건이란?

뒷면: 반드시 값이 있어야 함

설명: 필수 입력 필드에 사용


 암기 포인트: 'NOT NULL = 빈칸 금지'

카드 65

앞면: CHECK 제약조건이란?

뒷면: 조건 만족 여부 검사

설명: 급여 > 0 등


 암기 포인트: 'CHECK = 조건 필터'

카드 66

앞면: 도메인 무결성이란?

뒷면: 값이 정의된 범위 내에 있어야 함

설명: 데이터 형식과 범위 제한

 암기 포인트: '도메인 = 값의 규칙'

카드 67

앞면: 엔터티 무결성이란?

뒷면: 기본키는 NULL/중복 불가

설명: 고유 식별자 유지

 암기 포인트: '엔터티 = 고유성 유지'

카드 68

앞면: 참조 무결성이란?

뒷면: 외래키는 부모 테이블 값과 일치

설명: 연결된 데이터의 일관성

 암기 포인트: '참조 = 연결된 값 일치'

카드 69

앞면: 무결성과 제약조건의 연결은?

뒷면: 도메인/엔터티/참조 무결성 → 각 제약조건 대응

설명: 도메인 = CHECK, 엔터티 = PK, 참조 = FK

🔍 **알기 포인트:** '무결성 = 제약조건으로 구현'

대표 기출 문제

문제 1

트랜잭션의 원자성(Atomicity)이란?

① 작업을 순서대로 실행하는 것 ② 작업이 모두 성공하거나 모두 실패해야 함 ③ 작업 중 일부만 저장하는 것 ④ 작업을 병렬로 실행하는 것

✅ **정답:** ②

💡 **해설:** 트랜잭션은 하나의 단위로 처리되며, 중간 실패 시 전체 취소됩니다.

🔍 **알기 포인트:** 'Atomic = 올 or Nothing'

문제 2

인덱스를 사용하는 주요 목적은?

① 데이터 삭제 속도 향상 ② 검색 속도 향상 ③ 데이터 암호화 ④ 트랜잭션 관리

✅ **정답:** ②

💡 **해설:** 인덱스는 B-Tree 구조로 빠른 탐색을 가능하게 해줍니다.

🔍 **알기 포인트:** '인덱스 = 검색 가속기'

문제 3

뷰(View)의 특징으로 옳은 것은?

① 데이터를 직접 저장한다 ② 여러 테이블을 삭제한다 ③ SELECT 결과를 저장한 가상 테이블이다 ④ 트랜잭션을 관리한다

✅ **정답:** ③

💡 **해설:** 뷰는 실제 데이터를 저장하지 않고, SELECT 결과를 보여주는 창문 같은 역할을 합니다.

🔍 **알기 포인트:** '뷰 = 가상의 창문'

문제 4

다음 중 외래키(FK)의 역할은?

- ① 테이블의 고유값을 지정한다 ② 다른 테이블과 연결한다 ③ 속성의 이름을 바꾼다 ④ 테이블을 삭제한다

✓ 정답: ②

💡 해설: 외래키는 참조 무결성을 유지하며, 다른 테이블의 PK를 참조합니다.

🔍 알기 포인트: 'FK = 연결 다리'

문제 5

도메인 무결성(Domain Integrity)이란?

- ① 기본키가 중복되지 않도록 하는 것 ② 외래키가 부모 테이블과 일치하는 것 ③ 속성 값이 정의된 범위 내에 있어야 하는 것 ④ 트랜잭션을 모두 성공시키는 것

✓ 정답: ③

💡 해설: 도메인 무결성은 속성 값이 지정된 형식과 범위 내에 있어야 함을 의미합니다.

🔍 알기 포인트: '도메인 = 값의 규칙'

문제 6

트랜잭션을 취소하고 이전 상태로 되돌리는 명령은?

- ① COMMIT ② ROLLBACK ③ DELETE ④ ALTER

✓ 정답: ②

💡 해설: ROLLBACK은 트랜잭션 중 변경된 내용을 취소하고 원래 상태로 되돌립니다.

🔍 알기 포인트: 'ROLLBACK = 취소'

문제 7

다음 중 제약조건에 해당하지 않는 것은?

- ① PRIMARY KEY ② FOREIGN KEY ③ SELECT ④ CHECK

✓ 정답: ③

💡 해설: SELECT는 데이터를 조회하는 SQL 명령어이며, 제약조건이 아닙니다.

🔍 알기 포인트: '제약조건 = PK, FK, UNIQUE, NOT NULL, CHECK'

문제 8

UNIQUE 제약조건의 특징은?

- ① 중복 허용, NULL 불가 ② 중복 불가, NULL 허용 ③ 중복 불가, NULL 불가 ④ 중복 허용, NULL 허용

✓ 정답: ②

💡 해설: UNIQUE는 중복을 허용하지 않지만, NULL은 허용합니다.

🔍 알기 포인트: 'UNIQUE = 중복 ❌, NULL 가능'

문제 9

트랜잭션의 일관성(Consistency)이란?

- ① 작업이 빠르게 끝나는 것 ② 데이터가 항상 유효한 상태를 유지하는 것 ③ 작업을 병렬로 실행하는 것 ④ 데이터를 암호화하는 것

✓ 정답: ②

💡 해설: 일관성은 트랜잭션 전후에 데이터가 규칙을 만족하는 상태를 유지하는 것을 의미합니다.

🔍 알기 포인트: 'Consistency = 규칙 지키기'

문제 10

CHECK 제약조건의 용도는?

- ① 테이블 삭제 ② 속성 값이 조건을 만족하는지 검사 ③ 속성 이름 변경 ④ 트랜잭션 저장

✓ 정답: ②

💡 해설: CHECK는 속성 값이 특정 조건을 만족하는지 확인하는 제약조건입니다.

🔍 알기 포인트: 'CHECK = 조건 필터'

문제 11

다음 중 트랜잭션 격리 수준 중 가장 엄격한 것은?

- ① READ COMMITTED ② REPEATABLE READ ③ SERIALIZABLE ④ READ UNCOMMITTED

✓ 정답: ③

💡 해설: SERIALIZABLE은 모든 트랜잭션을 직렬화하여 가장 높은 격리 수준을 제공합니다.

🔍 알기 포인트: 'SERIALIZABLE = 완전 격리'

문제 12

다음 중 실행계획(EXPLAIN)의 주요 목적은?

- ① 테이블 삭제 ② 인덱스 생성 ③ SQL 실행 흐름 분석 ④ 트랜잭션 저장

✓ 정답: ③

💡 해설: EXPLAIN은 SQL이 어떻게 실행되는지 분석하고 인덱스 사용 여부를 확인합니다.

🔍 알기 포인트: 'EXPLAIN = 실행 흐름 보기'

문제 13

다음 중 뷰(View)를 수정할 수 있는 조건으로 옳은 것은?

- ① GROUP BY 포함 ② 단일 테이블 기반 ③ 집계 함수 포함 ④ JOIN 포함

✓ 정답: ②

💡 해설: 뷰가 수정 가능하려면 단일 테이블 기반이어야 하며, 집계 함수나 JOIN, GROUP BY가 없어야 합니다.

🔍 알기 포인트: '뷰 수정 = 단순 SELECT만 가능'

마무리: SQLD는 이 도시의 시민증

이 도시를 이해하고, 설계하고, 탐험하고,
그리고 지킬 수 있는 사람만이
SQLD 시민증(자격증)을 받을 수 있어요.