의료데이터베이스 과제

Ch07. 데이터베이스언어 SQL 연습문제

202110475 김재하

23. 뷰를 생성하기 위한 기본 형식 중 WITH CHECK OPTION의 의미를 설명하시오.

뷰가 생성된 이후 뷰에 대한 삽입이나 수정 연산이 실행될 때 뷰 정의 조건을 위반하면 실행을 거부하도록 지시한다.

24. 다음의 관계 대수를 SQL 문으로 작성하시오

π도서명, 가격 (σ출판사=’한빛’^재고량>=50(도서))

SELECT 도서명, 가격 FROM 도서 WHERE 출판사=’한빛’ AND 재고량>=50;

25. 제품 테이블에 제조업체가 대한식품인 제품 50개, 한국푸드인 제품 30개, 한빛제과인 제품 50개에 대한 데이터가 저장되어 있을 때, 다음 SQL 문의 실행 결과 투플 수는 각각 얼마인가?

가. SELECT 제조업체 FROM 제품;

나. SELECT 제품명 FROM 제품 WHERE 제조업체=’대한식품’;

다. SELECT 제품명 FROM 제품 WHERE 제조업체=’대한식품’;

가: 130, 나: 3, 다: 50

26. 다음 S0Ｌ 문을 성공적으로 실행하여 부서 테이블과 사원 테이블이 생성되었다. 각 물음에 답하시오.

CREATE TABLE 부서 (

부서코드 CHAR(3) NOT NULL,

부서명 VARCHAR(10),

PRIMARY KEY(부서코드)

);

CREATE TABLE 사원 (

사원번호 INT NoT NULL,

소속부서 v"ARCHAR(10),

PRIMARY KEY(사원번호),

FOREIGN KEY(소속부서) REFERENCES 부서(부서코드) 0N DELETE CASCADE

);

INSERT INTO 부서 VALUES ('P1', '개발부);

INSERT INTO 부서 vA1UES ('P2', '홍보부');

INSERT INTO 사원 VALUES (100, 'P1’);

INSERT INTO 사원 VALUES (200, 'P2y);

INSERT INTO 사원 VALUES (300, 'P2);

(1) 부서 테이블과 사원 테이블의 카디널리티는 각각 얼마인가?

부서 테이블: 2

사원 테이블: 3

(2) 다음 SQR 문이 성공적으로 실행되었다면 부서 테이블과 사원 테이블의 카디널리티는 각각 얼마인가?

DELETE FROM 부서 WHERE 부서코드 = ‘P2’;

부서 테이블: 1

사원 테이블: 1

27. 다음 학생 테이블을 보고 각 물음에 답하시오.

|  |  |  |  |
| --- | --- | --- | --- |
| 학번 | 이름 | 학점 | 학과 |
| 101 | 이진아 | 3.5 | 컴퓨터공학과 |
| 102 | 양기섭 | 4.1 | 컴퓨터공학과 |
| 103 | 박수정 | 3.7 | 호텔관광경영학과 |
| 104 | 홍민호 | 2.8 | 호텔관광경영학과 |
| 105 | 오연주 | 3.2 | 호텔관광경영학과 |
| 106 | 김우리 | 2.5 | 건축학과 |
| 107 | 조유근 | 4.3 | 건축학과 |

(1) 다음 SQL 문을 실행했을 때 반환되는 결과 테이블의 모습을 예상해서 작성하시오

SELECT 학과, COUNT(\*) AS 결과

FROM 학생

GROUP BY 학과 HAVING count(\*) > 2;

|  |  |
| --- | --- |
| 학과 | 결과 |
| 경영학과 | 3 |

(2) 학생 테이블에 최대 20글자 가변 길이 문자열 타입의 연락처 속성을 추가하는 SQL 문을 작성하시오.

ALTER TABLE 학생 ADD 연락처 VARCHAR(20);

(3) 다음 SQL 문을 실행했을 때 반환되는 결과 테이블에서 가장 먼저 출력되는 학생은 누구인가?

SELECT 이름 FROM 학생 ORDER BY 학과 DESC, 학점 ASC;  
홍민호

(4) 김우리 학생과 같은 학과에 속한 학생들의 평균 학점을 검색하는 SQL문을 작성하시오. 단, 평균 학점이라는 이름으로 결과를 출력한다.

SELECT AVG (학점) AS "평균 학점"

FROM 학생

WHERE 학과 = (SELECT 학과 FROM 학생 WHERE 이름 = ‘김우리’);

(5) 학번이 105인 학생의 학점을 4.5로 학과를 전자과로 변경하는 SQL문을 작성하시오.

UPDATE 학생

SET 학점 = 4.5, 학과 = ‘전자과’

WHERE 학번 = 105;

28. 다음 두 테이블을 보고 각 물음에 답하시오.

환자 테이블

|  |  |  |  |
| --- | --- | --- | --- |
| 환자번호 | 환자이름 | 나이 | 담당의사 |
| P001 | 오우진 | 31 | D002 |
| P002 | 채광주 | 50 | D001 |
| P003 | 김용욱 | 43 | D003 |

의사 테이블

|  |  |  |  |
| --- | --- | --- | --- |
| 의사번호 | 의사이름 | 소숙 | 근무연수 |
| D001 | 정지영 | 내과 | 5 |
| D002 | 김선주 | 피부과 | 10 |
| D003 | 정성호 | 정형외과 | 15 |

(1) 환자 테이블을 생성하는 SQＬ 문을 작성하시오. 단, 이름 속성은 널 값이 허용되지 않도록 지정하고, 담당의사 속성을 의사 테이블의 의사번호 속성을 참조하는 외래키로 지정하시오. (CREATE TABLE 환자 (

환자번호 CHARr(4) NOT NULL

환자이름 VARCHAR(10) NOT NULL,

나이 INT,

담당의사 CHAR(4),

PRIMARY KEY(환자번호),

FOREIGN KEY(담당의사) REFERENCES 의사(의사번호)

);

(2) 의사 테이블을 생성하는 SQL 문을 작성하시오. 단, 소속 속성의 값을 입력하지 않으면 자동으로 내과가 지정되도록 하고, 근무연수는 1년 이상 40년 이하 범위의 값을 가지도록 지정하시오.

CREATE TABE 의사(

의사번호 CHAR(4) NOT NULL,

의사이름 VARCHAR(10),

소속 VARCHAR(20) DEFAULT '내과,

근무연수 NT,

PRIMARY KEY(의사번호),

CHECK (근무연수 >= 1 AND 근무연수 <= 40)

);

(3) D001 의사가 담당하고 나이가 30세 이상인 환자의 환자번호와 환자이름을 검색하는 SQL 문을 작성 하시오.

SELECT 환자번호, 환자이름

FROM 환자

WHERE 담당의사 = ‘D001’ AND 나이 >= 30;

(4) 소속별로 의사의 수와 평균 근무연수를 검색하는 SQL 문을 작성하시오.

SELECT 소속, COUNT(\*) AS "의사 수", AVG(근무연수) 45 "평균 근무연수"

FROM 의사

GROUP BY 소속;

(5) 김용욱 환자를 담당하는 의사의 의사이름과 소속, 근무연수를 검색하는 SQL 문을 작성하시오.

SELECT 의사.의사이름, 의사.소속, 의사.근무연수

FROM 의사, 환자

WHERE 환자.환자이릉 = "김용욱 AND 의사.의사번호 = 환자.담담의사:

29. 다음 3개의 테이블 스키마를 보고 각 물음에 답하시오.

학생(학번, 이름, 학년)

과목(과목번호, 과목이름)

수강(학번, 과목번호, 중간성적, 기말성적, 학점)

(1) 수강 테이블에서 과목번호가 A로 시작하는 과목의 기말성적이 80점 이상인 학생의 이름과 중간성적을 검색하는 SQL 문을 작성하시오. 단, 이름을 기준으로 오름차순 정렬하고, 만약 이름이 같으면 중간성적을 기준으로 내림차순 정렬하시오.

SELECT 학생.이름, 수강.중간성적

FROM 학생, 수강

WHERE 수강.과목번호 LIKE ‘A%’ and 수강.기말성적>=80 and 학생.학번=수강.학번

ORDER BY 학생.이름 ASC, 수강.기말성적 DESC;

(2) 수강 테이블에서 3명 이상의 학생이 수강하는 과목에 대해 과목별 등록 학생의 수와 기말성적의 평균을 검색하는 SQL 문을 작성하시오. 이때, 등록한 학생의 총 수는 ‘학생수’로, 기말성적의 평균은 ‘성적평균’으로 속성 이름을 새로 부여하시오.

SELECT 과목번호, COUNT(\*) AS 학생수, AVG(기말성적) AS 성적평균

FROM 수강

GROUP BY 과목번호 HAVING COUNT(\*)>=3;

(3) 수강 테이블에서 개설된 과목의 수를 검색하는 SQL 문을 작성하시오.

SELECT COUNT(DISTINCT 과목번호) FROM 수강;

(4) 과목번호가 A003인 과목을 수강하지 않는 학생의 이름과 학년을 검색하는 SQL 문을 작성하시오. 단, IN 연산자를 이용해 작성하시오.

SELECT 이름, 학년

FROM 학생

WHERE 학번 NOT IN (SELECT 학번 FROM 수강 WHERE 과목번호=‘A003’);

(5) 과목번호가 A003인 과목을 수강하지 않는 학생의 이름과 학년을 검색하는 SQL 문을 작성하시오. 단, EXISTS 연산자를 이용해 작성하시오.

SELECT 이름, 학년

FROM 학생

WHERE NOT EXISTS (SELECT\*

FROM 수강

WHERE 과목번호=‘A003’ AND 학생.학번=수강.학번);

30. 다음 네 개의 테이블 스키마를 보고 각 물음에 답하시오.

고객(고객번호, 고객이름, 거주도시, 할인율)

판매자(판매자번호, 판매자이름, 수수료, 판매지 역)

제품(제품번호, 제품명, 재고량, 가격)

주문(주문번호, 고객번호, 제품번호, 판매자번호, 주문수량)

(1) 고객 테이블에 고객번호가 005, 이름이 채희성, 거주도시가 대구이고, 할인율은 아직 결정되지 않은 고객의 데이터를 삽입하는 SQＬ 문을 작성하시오.

INSERT INTO 고객 VALUES ('C005", '채희성', '대구', NULL);

(2) 방지호 고객의 주문수량을 20% 증가시키는 SQＬ 문을 작성하시오.

UPDATE 주문

SET 주문수량 = 주문수량 \* 1,2

WHERE 고객번호 IN (SELECT 고객번호 FR0M 고객 WHERE 고객이름 = ‘방지호’);

(3) 재고량이 50개 미만인 제품을 모두 삭제하는 SQＬ 문을 작성하시오.

DELETE

FROM 제품

WHERE 재고량<50:

(4) 가격이 가장 최저가인 제품의 제품명을 중복 없이 검색하는 SQL 문을 작성하시오.

SELECT DISTINCT 제품명

FROM 제품

WHERE 가격 = (SELECT MIN(가격) FROM 제품)

(5) 제품명에 '부'가 포함된 제품을 주문한 고객의 이름을 검색하는 SQＬ 문을 작성하시오.

SELECT 고객.고객명

FROM 고객, 제품, 주문

WHERE 제품.제품명 LIKE '%부%' AND 고객.고객번호 = 주문.고객번호

AND 제품.제품번호 = 주문.제품번호:

(6) 주문수량의 합계가 300개 이상인 고객에 대해 고객별 주문 횟수와 주문수량의 합계를 검색하는 SQＬ 문을 작성하시오.

SELECT 고객번호, COUNT(\*) AS 주문횟수, SUM(주문수량) AS 주문수량합계

FROM 주문

GROUP BY 고객번호 HAVING SUM(주문수량) >= 300;

(7) 판먀지역이 서울인 모든 판매자의 수수료보다 비싼 수수료를 받는 판매자의 이름과 수수료, 판매지역을 검색하기 위한 SQL 문을 작성하싱오.

SELECT 판매자이름, 수수료, 판매지역

FROM 판매자

WHERE 수수료 > ALL (SELECT 수수료 FROM 판매자 WHERE 판매지역 = ‘서울’);