

데이터베이스론

2017년도 국가공무원 5급(기술) 공개경쟁채용 제2차시험

응시번호 :                      성명 :

제 1 문. 비신장(unspanned) 레코드 구조를 사용하는 환경에서 디스크의 블록 크기가 4,096바이트이고, 블록 포인터와 레코드 포인터가 각각 4바이트라고 가정하자. 고객 파일에는 1,000,000개의 고정길이 고객 레코드가 들어 있다. 고객 레코드의 길이는 100바이트이고 구조는 다음과 같다. 물음에 답하시오. (총 20점)

고객번호	이름	주소	전화번호	성별	나이
30바이트	15바이트	40바이트	11바이트	1바이트	3바이트

- 1) 고객 파일에 대한 블로킹 인수(blocking factor)를 구하시오. (3점)
- 2) 고객 파일에 필요한 최소 데이터 블록 수를 구하시오. (3점)
- 3) 파일이 기본키인 고객번호 순서로 정렬되어 있다고 가정하자. 고객번호에 대하여 회소 기본 인덱스를 생성하려고 한다. 인덱스 블로킹 인수와 1단계 인덱스 블록 수를 구하시오. (5점)
- 4) 다단계 인덱스를 구축할 경우 필요한 총 단계의 수를 구하시오. (3점)
- 5) 다단계 인덱스가 필요로 하는 총 블록 수를 구하시오. (3점)
- 6) 다단계 인덱스를 사용하여 한 레코드를 검색할 때 필요한 디스크 블록 접근 횟수를 구하시오. (단, 마스터 인덱스는 메모리에 상주하는 것으로 가정한다) (3점)

제 2 문. 데이터베이스 항목 A의 초기값은 0이다. 아래의 트랜잭션 T1, T2, T3, T4가 동시에 수행된다. 다음 물음에 답하시오. (총 15점)

T1	T2	T3	T4
BEGIN READ(A); X = A; READ(A); Y = A; B = X + Y; WRITE(B); COMMIT;	BEGIN A = 1; WRITE(A); ABORT;	BEGIN A = 10; WRITE(A); COMMIT;	BEGIN A = 100; WRITE(A); COMMIT;

- 1) 모든 트랜잭션이 READ UNCOMMITTED 고립 수준(isolation level)으로 수행될 때, 트랜잭션 T1이 완료된 직후에 가능한 B의 값을 모두 나열하시오. (5점)
- 2) 모든 트랜잭션이 READ COMMITTED 고립 수준으로 수행될 때, 트랜잭션 T1이 완료된 직후에 가능한 B의 값을 모두 나열하시오. (5점)
- 3) 모든 트랜잭션이 REPEATABLE READ 고립 수준으로 수행될 때, 트랜잭션 T1이 완료된 직후에 가능한 B의 값을 모두 나열하시오. (5점)

제 3 문. 두 개의 릴레이션 deposit(id, name, amount, branch)과 loan(id, name, amount, branch)이 있을 때, 다음 물음에 답하시오.(단, id는 계좌번호, name은 이름, amount는 금액, branch는 지점명이다) (총 15점)

deposit			
id	name	amount	branch
1	John	100	상도
2	Smith	150	관악
3	John	150	상도
4	Maria	100	동작
5	Juliet	450	동작
6	Eagle	100	상도
7	Kite	200	방배

loan			
id	name	amount	branch
1	John	150	관악
2	Smith	200	상도
3	Storm	200	동작
4	Smith	250	동작
5	Kim	100	방배
6	Lee	300	상도

- 1) deposit 릴레이션에서 “예금액(amount)이 가장 적은 계좌의 이름(name)을 검색하라”는 질의를 관계대수식(relational algebra)으로 표현하시오. (단, 관계대수 연산자의 표기는 다음과 같으며, 그룹 연산자는 사용하지 않는다) (7점)

$\sigma$  : selection  
 $\Pi$  : projection  
 $\times$  : Cartesian product  
 $\rho$  또는  $\rightarrow$  : rename  
 $\cup$  : union  
 $-$  : set difference

- 2) 다음 SQL문을 실행한 결과를 구하시오. (4점)

```

with deposit_total(name, value) as
    (select name, sum(amount)
     from deposit
     group by name),
loan_total(name, value) as
    (select name, sum(amount)
     from loan
     group by name)
select deposit_total.name
from deposit_total, loan_total
where deposit_total.name = loan_total.name and
      deposit_total.value > loan_total.value;
    
```

- 3) 다음과 같이 customer 뷰(view)가 정의되었다고 가정하자.

```

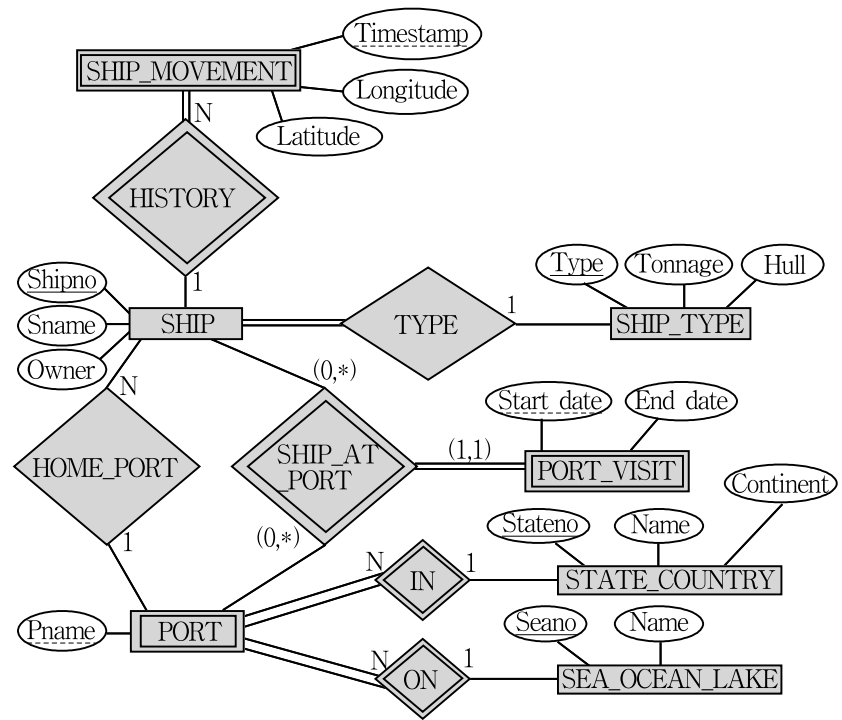
create view customer(name, branch) as
    select name, branch from deposit
    union all
    select name, branch from loan;
    
```

이 때, 다음 SQL문의 실행 결과를 구하시오. (4점)

```

select name, count(*) from customer
group by name
having count(*) >= 2;
    
```

제 4 문. 다음은 해양수산부에서 수송선(ship)의 위치 관리 데이터베이스 설계를 위한 개체-관계 다이어그램(Entity-Relationship Diagram)이다. 이 개체-관계 다이어그램을 관계 테이블로 모두 사상시키고 모든 개체와 관계(relationship)의 기본키(primary key)를 표시하시오. (단, 관계 테이블로 사상할 때 가장 효율적인 데이터베이스 스키마가 되도록 변환하며, 필요 시 기본키로 ‘일련번호 (Serialno)’와 같은 속성을 추가할 수 있고 기본키는 밑줄로 표시한다) (20점)



제 5 문. 데이터베이스에서는 성능향상 등을 위하여 정렬의 필요성이 있으나, 저장되어 있는 데이터의 양이 메모리의 용량을 초과하여 메모리 상주 정렬 알고리즘 (in-memory sorting algorithm)을 사용할 수 없는 경우가 빈번하며, 이러한 경우 하드디스크와 같은 대용량 저장장치에 기반한 외부 정렬-합병(external sort-merge) 알고리즘을 사용하게 된다. 이 때, 1회의 하드디스크 접근 시 동시에 읽거나 쓰는 데이터 크기의 단위를 블록이라고 하며, 메모리에서 상주가능한 최대 블록의 개수를 M으로 표기한다. 다음 물음에 답하시오. (총 10점)

- 1) 15개의 레코드를 가지고 있는 릴레이션의 사례에서 하나의 블록에는 하나의 레코드만이 저장될 수 있으며 M = 3일 경우, 첫번째 열의 오름차순을 기준으로 외부 정렬-합병 알고리즘을 수행하는 과정을 각 단계별로 모두 보이시오. (5점)

나	...
라	...
가	...
아	...
타	...
자	...
파	...
카	...
마	...
다	...
차	...
바	...
하	...
사	...
거	...

- 2) 만약 1)번 문제의 릴레이션 레코드가 3,072개이고 나머지 조건은 동일하다고 가정할 때 모든 정렬을 수행하기 위해 필요한 총 디스크 접근 횟수를 구하시오. (단, 최종적인 정렬의 결과를 하드디스크의 파일에 저장하는 것까지 모두 포함한다) (5점)

제 6 문. 다음은 책과 과목정보를 표현하기 위한 XML DTD이다. 책(Book)에 대해서는 제목(Title), 저자정보(Authors), 서평(Review), 각 책의 고유 식별자인 ISBN 번호(ISBN)가 포함되고, 과목(Course)에 대해서는 과목명(Title), 담당교수정보(Professor), 교재(Textbook)가 포함된다. 또한 저자(Author), 교수(Professor) 등의 인명 정보는 이름(Name), 생년월일(Birthdate), 주소(Address), 전화번호(Phone)를 포함한다. 다음 물음에 답하시오. (단, 아래의 DTD를 준수하는 XML 문서의 내용으로는 동일 제목의 서로 다른 책은 존재할 수 있으나, 동일 과목명의 서로 다른 과목은 존재할 수 없고, 동일한 생일을 가진 동명이인은 존재하지 않는다고 가정하고, 한 명의 저자는 다수의 책을 저술할 수 있으며, 한 명의 교수도 다수의 과목을 맡을 수 있다)

```
<!DOCTYPE DB [
<!ELEMENT DB (Book | Course)*>
<!ELEMENT Book (Title, Authors, Review?)>
<!ATTLIST Book ISBN ID #REQUIRED>
<!ELEMENT Course (Title, Professor)>
<!ATTLIST Course Textbook IDREF #REQUIRED>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Authors (Author+)>
<!ELEMENT Review (#PCDATA)>
<!ELEMENT Author (Name, Birthdate, Address, Phone)>
<!ELEMENT Professor (Name, Birthdate, Address, Phone)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Birthdate (#PCDATA)>
<!ELEMENT Address (#PCDATA)>
<!ELEMENT Phone (#PCDATA)>
]>
```

- 위 DTD를 준수하는 XML 문서를 관계형 데이터베이스에 저장하기 위하여 아래 스키마가 제시되었다고 하자. 각 테이블의 모든 컬럼의 자료형은 적절하게 설정된다고 가정하고, 밑줄( )은 각 테이블의 기본키(primary key)를 의미하고, 겹밑줄( )은 외래키(foreign key)를 의미한다고 하자. 이 때 이 스키마의 문제점을 모두 지적하시오. (8점)

Book(Title, Name, Birthdate, Address, Phone, Review, ISBN)

Course(Title, Name, Birthdate, Address, Phone, Textbook)

- 1)번에 주어진 스키마의 문제점을 해결할 수 있도록 SQL 문을 사용하여 새로운 스키마를 작성하시오. (단, 모든 자료형은 문자열 (varchar(n))을 사용하고, XML DTD와 주어진 조건에서 제시하고 있는 제약조건(constraint)을 최대한 반영한다) (12점)

## 인사혁신처 시험출제과장