

[소프트웨어 개발>데이터 입출력 구현>논리 데이터저장소 확인]

[이전 기출 문제]

1. 비선형 구조와 선형 구조가 옳게 짝지어진 것은?

- | | |
|---------------|------------------------|
| ㄱ. 스택(Stack) | ㄴ. 큐(Queue) |
| ㄷ. 트리(Tree) | ㄹ. 연결 리스트(Linked List) |
| ㅁ. 그래프(Graph) | |

- ① 비선형 구조 : ㄱ, ㄴ, ㅁ / 선형 구조 : ㄷ, ㄹ
② 비선형 구조 : ㄷ, ㅁ / 선형 구조 : ㄱ, ㄴ, ㄹ
③ 비선형 구조 : ㄱ, ㄴ, ㄷ / 선형 구조 : ㄹ, ㅁ
④ 비선형 구조 : ㄷ / 선형 구조 : ㄱ, ㄴ, ㄹ, ㅁ

[이전 기출 문제]

2. 마지막 레벨을 제외하고 노드가 모두 채워져 있는 트리는?

- ① 포화 이진트리 ② 완전 이진트리
③ 불포화 이진트리 ④ 불완전 이진트리

[이전 기출 문제]

3. 연결 리스트(Linked List)에 대한 설명으로 거리가 먼 것은?

- ① 노드의 삽입이나 삭제가 쉽다.
② 노드들이 포인터로 연결되어 검색이 빠르다.
③ 연결을 해주는 포인터(Pointer)를 위한 추가 공간이 필요하다.
④ 연결 리스트 중에는 중간 노드 연결이 끊어지면 그 다음 노드를 찾기 힘들다.

[이전 기출 문제]

4. A, B, C, D의 순서로 정해진 입력 자료를 스택에 입력하였다가 출력한 결과가 될 수 없는 것은? (단, 왼쪽부터 먼저 출력된 순서이다.)

- ① C, B, A, D ② C, D, A, B
③ B, A, D, C ④ B, C, D, A

[이전 기출 문제]

5. 데크(deque)에 대한 옳은 설명으로만 짝지어진 것은?

- | |
|------------------------------|
| ㄱ. 양끝에서 노드의 삽입과 삭제가 모두 가능하다. |
| ㄴ. 하나의 포인터를 사용한다. |
| ㄷ. double ended queue의 약자이다. |
| ㄹ. 선형 구조이다. |

- ① ㄱ, ㄴ ② ㄱ, ㄴ, ㄷ
③ ㄱ, ㄷ, ㄹ ④ ㄱ, ㄹ

[이전 기출 문제]

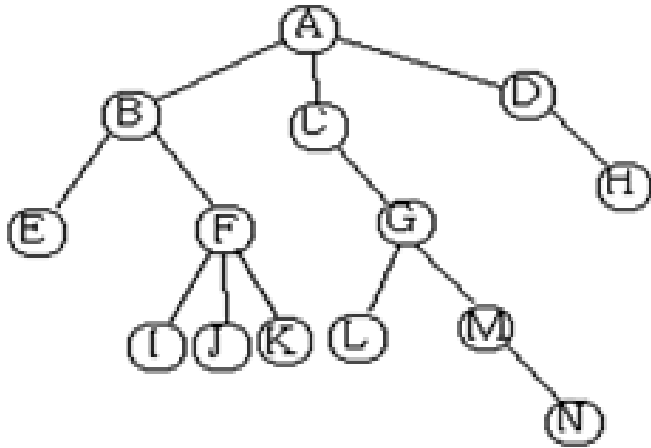
6. 자료구조에 관한 설명 중 옳지 않은 것은?

- ① 스택은 LIFO 구조로 복귀주소(return address) 등에 이용된다.
② 큐는 FIFO 구조로 작업 스케줄링 등에 이용된다.
③ 트리는 선형 구조이다.
④ 데크(Deque)는 서로 다른 방향에서 입.출력이 가능하다.

[소프트웨어 개발>데이터 입출력 구현>논리 데이터저장소 확인]

[이전 기출 문제]

7. 다음과 같은 트리(tree) 구조에서 기본 용어의 설명으로 맞는 것은?



- ① node는 10 이다.
- ② tree의 차수(degree of tree)는 4이다.
- ③ N노드의 레벨(level)은 5이다.
- ④ 근(root) node는 N이다.

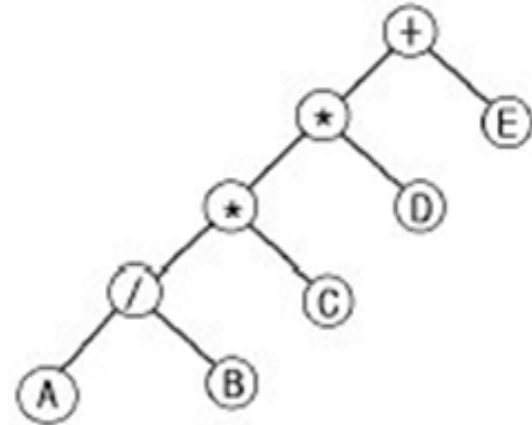
[이전 기출 문제]

8. 트리(tree)에서 자식이 없는 노드를 무엇이라고 하는가?

- ① Parent node
- ② Brother node
- ③ Leaf node
- ④ Children node

[이전 기출 문제]

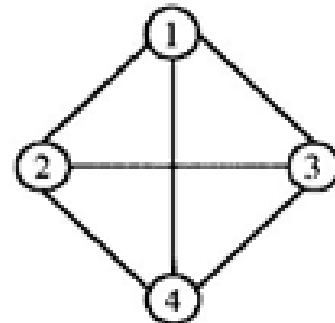
9. 다음 그림과 같은 이진 트리를 전위 순회(preorder-traversal)한 결과는?



- ① AB*CD*E/+
- ② *B+C*D/E
- ③ +**/ABCDE
- ④ A*B+CD*/E

[이전 기출 문제]

10. 다음과 같은 그래프에서 간선의 개수는?



- ① 2개
- ② 4개
- ③ 6개
- ④ 8개

[소프트웨어 개발>데이터 입출력 구현>논리 데이터저장소 확인]

[이전 기출 문제]

11. 역파일(inverted file)에 관한 설명으로 옳지 않은 것은?

- ① 검색 속도가 빠르다.
- ② 데이터 파일에 접근하지 않아 질의 응답 시간이 줄어 들고, 처리가 비교적 쉽다.
- ③ 질의를 만족하는 레코드 검색시 한번씩만 접근하면 된다.
- ④ 색인의 각 항의 길이가 고정적이므로 기억 공간이 절약된다.

[기출 예상 문제]

12. 역(inverted) 파일과 다중 리스트 파일에 대한 비교, 설명 중 잘못된 것은?

- ① 역 파일은 각 필드마다 인덱스를 만들어 구현한다.
- ② 다중리스트 파일은 하나의 인덱스 값마다 하나의 데이터 레코드 리스트를 구축한다.
- ③ 역 파일은 인덱스 대신 데이터 레코드들 사이에 여러 개의 리스트를 구축하여 구현한다.
- ④ 다중리스트 파일은 동일한 키 값을 갖는 데이터 레코드 중 하나의 레코드에 대한 포인터만을 갖는다.

[이전 기출 문제]

13. 색인 순차 파일(Indexed Sequential File)에 대한 설명으로 옳지 않은 것은?

- ① 색인 영역은 트랙 색인 영역, 실린더 색인 영역, 오버플로우 색인 영역으로 구분할 수 있다.
- ② 랜덤(random) 및 순차(sequence)처리가 모두 가능하다.
- ③ 레코드의 삽입과 삭제가 용이하다.
- ④ 색인 및 오버플로우를 위한 공간이 필요하다.

[이전 기출 문제]

14. 직접 접근 방식(DAM : Directed Access Method)에 설명으로 거리가 먼 것은?

- ① 데이터의 입/출력이 빈번히 발생하는 곳에 응용하는 것이 좋다.
- ② 해싱 함수를 이용하여 레코드의 저장 위치를 결정한다.
- ③ 다른 레코드를 참조하지 않고 어떤 레코드를 접근할 수 있다.
- ④ 기억 공간의 효율성이 매우 좋다.

[소프트웨어 개발>데이터 입출력 구현>논리 데이터저장소 확인]

[기출 예상 문제]

15. 정적 인덱스 방법과 동적 인덱스 방법에 대한 설명으로 가장 적절한 것은?

- ① 정적 인덱스 방법에서 저장 공간이 부족할 때 오버플로 구역을 사용한다.
- ② 동적 인덱스의 레코드가 삭제될 때 인덱스 내용은 변하지만 인덱스의 구조 자체는 변경되지 않는다.
- ③ 정적 인덱스 방법에서 각 블록에는 추가로 삽입될 레코드를 감안하여 일정 비율 이상의 빈 공간을 미리 예비해 둔다.
- ④ 동적 인덱스의 방법에서 일정 수의 레코드를 유지하지 못하는 블록은 둘로 분할 시킨다.

[이전 기출 문제]

16. 데이터베이스 설계 순서를 바르게 나열한 것은?

- ㄱ. 요구조건의 분석 ㄴ. 물리적 설계
- ㄷ. 데이터베이스 구현 ㄹ. 개념적 설계
- ㅁ. 논리적 설계

- ① ㄱ-ㄴ-ㄷ-ㄹ-ㅁ
- ② ㄱ-ㄷ-ㄴ-ㄹ-ㅁ
- ③ ㄱ-ㄹ-ㅁ-ㄴ-ㄷ
- ④ ㄱ-ㄴ-ㄹ-ㄷ-ㅁ

[이전 기출 문제]

17. 다음에서 설명하는 데이터베이스 설계 단계는?

- 데이터베이스 관리시스템(DBMS)에 따라 서로 다른 논리적 스키마를 정의한다.
- 현실 세계를 표현하기 위한 데이터베이스의 논리적 구조 즉, 정규화 과정을 이용한 릴레이션의 속성을 결정하는 단계이다.

- ① 요구사항 및 분석단계
- ② 개념적 설계 단계
- ③ 논리적 설계 단계
- ④ 물리적 설계 단계

[이전 기출 문제]

18. 물리적 데이터베이스 설계를 수행할 때 결정할 사항으로 거리가 먼 것은?

- ① 어떤 인덱스를 만들 것인지에 대한 고려
- ② 성능 향상을 위한 개념 스키마의 변경 여부 검토
- ③ 빈번한 질의와 트랜잭션들의 수행 속도를 높이기 위한 고려
- ④ 개념스키마와 외부스키마 설계

[소프트웨어 개발>데이터 입출력 구현>논리 데이터저장소 확인]

[기출 예상 문제]

19. 논리 데이터저장소에 대한 설명으로 가장 거리가 먼 것은?

- ① 데이터 구조 등의 논리적인 자료구조로 만들어진 데이터 저장소이다.
- ② 논리 데이터저장소를 설계하는 과정이 개념 데이터 모델링이다.
- ③ 논리적 모델은 하드웨어나 소프트웨어에 독립적이다.
- ④ 모든 이해당사자들과 의사소통의 보조자료로서 ER 모델을 활용한다.

[기출 예상 문제]

20. 논리 데이터저장소 확인 절차로 가장 적절한 것은?

- ① 데이터 흐름 확인: 프로세스별 CRUD Matrix 확인
- ② 관계 확인: 계정별 접근 제어 확인
- ③ 백업정책 및 분산구조 확인: 정규화 수준 점검
- ④ 엔티티 및 속성 확인: 엔티티별 데이터 접근 권한 확인

[기출 예상 문제]

21. E-R모델에서 데이터를 묘사하는 개념으로 가장 거리가 먼 것은?

- ① 엔티티 ② 속성
- ③ 레코드 ④ 관계

[기출 예상 문제]

22. 논리 데이터저장소 확인에 대한 설명으로 가장 거리가 먼 것은?

- ① 데이터 프로세스와 엔티티 목록으로 CRUD Matrix를 작성한다.
- ② 엔티티 정의서에서 데이터 속성을 확인한다.
- ③ 엔티티 속성 중 PK 값은 주키에 관련된 값이다.
- ④ 자주 사용되는 코드성 데이터는 중복코드 대상이다.

[소프트웨어 개발>데이터 입출력 구현>물리 데이터저장소 설계]

[기출 예상 문제]

1. 다음은 무엇에 대한 설명인가?

- 전체 데이터베이스의 물리적 저장 형태를 기술한 것
- 데이터베이스 시스템 구조에서 데이터가 실제로 저장되는 구조를 기술

- ① 외부 스키마 ② 개념 스키마
- ③ 내부 스키마 ④ 물리 스키마

[기출 예상 문제]

2. 논리 데이터 모델에서 물리 데이터 모델 변환에 대한 설명으로 가장 거리가 먼 것은?

- ① 컬럼(Column)을 속성(Attribute)로 변환한다.
- ② UID를 기본키(Primary Key)로 변환한다.
- ③ 관계(Relationship)를 외래키(Foreign Key)로 변환한다.
- ④ 컬럼 유형(Type)과 길이(Length)를 정의한다.

[기출 예상 문제]

3. 테이블 제약조건 중 삭제 제약 조건의 옵션으로 가장 거리가 먼 것은?

- ① Cascade ② Nullify
- ③ Restricted ④ Plus

[기출 예상 문제]

4. 인덱스 설계에 대한 설명으로 가장 거리가 먼 것은?

- ① 인덱스 컬럼은 되도록이면 수정이 빈번한 컬럼을 선정한다.
- ② 지나치게 많은 인덱스는 오버헤드로 작용하므로 주의한다.
- ③ 인덱스는 추가적인 저장공간이 필요함을 고려해야 한다.
- ④ 인덱스와 테이블 데이터의 저장 공간이 적절히 분리되도록 설계해야 한다.

[기출 예상 문제]

5. 뷰 속성에 대한 설명으로 가장 적절한 것은?

- ① REPLACE: Sub-Query 내의 조건을 만족하는 행만 변경한다.
- ② FORCE: 기본 테이블이 존재할 때만 뷰를 생성한다.
- ③ NOFORCE: 기본 테이블의 존재 여부에 관계 없이 뷰를 생성한다.
- ④ WITH READ ONLY: DML 작업이 불가능한 뷰를 생성한다.

[소프트웨어 개발>데이터 입출력 구현>물리 데이터저장소 설계]

[기출 예상 문제]

6. 클러스터 적용 기준과 가장 거리가 먼 것은?

- ① 분포도가 좁을수록 오히려 유리한 기법이다.
- ② 분포도가 넓은 테이블의 클러스터링은 저장 공간을 절약 가능하다.
- ③ 대량의 범위를 자주 액세스하는 경우 적용한다.
- ④ 여러 개의 테이블이 빈번히 조인을 일으킬 때 활용한다.

[기출 예상 문제]

7. 파티션에 대한 설명으로 가장 거리가 먼 것은?

- ① 범위분할: 지정한 열의 값을 기준으로 분할
- ② 논리분할: 논리적 설계에 따라 개체 기준으로 분할
- ③ 해시분할: 해시 함수에 따라 데이터를 분할
- ④ 조합 분할: 범위분할에 의해 데이터를 분할한 다음 해시 함수를 적용하여 다시 분할

[기출 예상 문제]

8. 테이블의 참조된 기본키의 값이 수정될 경우의 내용을 정의하는 제약 조건의 옵션에 대한 설명으로 가장 거리가 먼 것은?

- ① Cascade: 참조한 테이블에 있는 외부키와 일치하는 모든 Row가 수정된다.
- ② Restricted: 참조한 테이블에 있는 외부키에 없는 것만 수정 가능하다.
- ③ Insert: 참조한 테이블에 있는 외부키를 삽입하여 수정한다.
- ④ Nullify: 참조한 테이블에 정의된 외부키와 일치하는 것을 Null로 수정한다.

[기출 예상 문제]

9. 파티션의 장점으로 가장 거리가 먼 것은?

- ① 데이터 액세스 범위를 줄여서 성능이 향상된다.
- ② 각 분할 영역을 독립적으로 백업하고 복구 가능하다.
- ③ 전체 데이터의 훼손 가능성이 감소한다.
- ④ 파티션 키 값 변경에 대한 관리가 필요없어 테이블 관리에 용이하다.

[소프트웨어 개발>데이터 입출력 구현>물리 데이터저장소 설계]

[기출 예상 문제]

10. 다음은 무엇에 대한 설명인가?

관계형 데이터베이스와 객체 지향 프로그래밍 언어 간의 호환되지 않는 데이터를 변환하는 프로그래밍 기법이다. 객체 지향 언어에서 사용하는 객체를 관계형 데이터베이스로 변환하여 테이블을 구성하는데 활용된다.

- ① PL/SQL ② Operation
- ③ ORM ④ Function

[기출 예상 문제]

11. ORM 프레임워크에서 관계와 테이블 간의 변환 방법에 대한 설명으로 가장 거리가 먼 것은?

- ① 1:M 관계는 M 관계에 있는 테이블에 있는 주키를 1 관계 테이블에 외부키로 매핑한다.
- ② 1:1 관계는 두 개의 테이블을 합쳐서 하나의 테이블로 해도 되는지 확인한다.
- ③ 1:1 관계는 접근의 빈도수가 많은 테이블로 상대방 테이블의 주키를 외부키로 매핑한다.
- ④ M:M 관계는 새로운 관계 테이블을 만들고, 이 테이블에 양쪽 테이블의 주키를 갖도록 설계한다.

[기출 예상 문제]

12. ORM 기술에 대한 설명으로 가장 적절한 것은?

- ① SQL Mapping 기반 기술은 Full ORM이다.
- ② Partial ORM은 SQL을 특정 name으로 정의하여 사용한다.
- ③ OR Mapping 기반 기술의 매퍼는 SQL Mapper이다.
- ④ OR Mapping 기반 기술은 SQL Mapper 비해 덜 효율적인 매핑 방법이다.

[기출 예상 문제]

13. 객체 관계 중 Generalization의 테이블간 변환 방법에 대한 설명으로 가장 거리가 먼 것은?

- ① 상위 클래스와 각 하위 클래스를 각각 별도의 테이블로 변환한다.
- ② 상위 클래스가 하위 클래스의 모든 속성을 포함하여 단일 테이블로 변환한다.
- ③ 하위 클래스들이 상위 클래스의 속성을 상속받아 각각 하위 클래스들을 테이블로 매핑한다.
- ④ 하위 클래스의 테이블과 상위 클래스의 테이블을 병합한 테이블을 작성한다.

[소프트웨어 개발>데이터 입출력 구현>물리 데이터저장소 설계]

[이전 기출 문제]

14. 트랜잭션이 가져야 할 특성으로 거리가 먼 것은?

- ① 정확성(accuracy)
- ② 원자성(atomicity)
- ③ 일관성(consistency)
- ④ 고립성(isolation)

[이전 기출 문제]

15. A 은행에서 B 라는 사람이 고객 인증 절차를 거쳐 잔액을 조회한 후, 타인에게 송금하는 도중에 장애가 발생하였을 경우 문제가 발생한다. 이러한 경우의 부작용을 방지할 수 있는 트랜잭션의 특성은?

- ① 원자성(atomicity) ② 일관성(consistency)
- ③ 분리성(isolation) ④ 지속성(durability)

[이전 기출 문제]

16. 트랜잭션의 실행이 실패하였음을 알리는 연산자로 트랜잭션이 수행한 결과를 원래의 상태로 원상 복귀시키는 연산은?

- ① COMMIT 연산 ② BACKUP 연산
- ③ LOG 연산 ④ ROLLBACK 연산

[기출 예상 문제]

17. 트랜잭션이 수행 중에 다른 트랜잭션 연산에 끼어들 수 없음을 의미하는 트랜잭션의 특징은?

- ① atomicity ② consistency
- ③ isolation ④ durability

[기출 예상 문제]

18. 한 작업의 논리적 단위가 성공적으로 끝났고, 데이터베이스가 다시 일관된 상태에 있으며 이 트랜잭션이 행한 갱신 연산이 완료된 것을 트랜잭션 관리자에게 알려주는 연산은?

- ① ROLLBACK 연산 ② LOG 연산
- ③ COMMIT 연산 ④ BACKUP 연산

[소프트웨어 개발>데이터 입출력 구현>데이터 조작 프로시저 작성]

[이전 기출 문제]

1. 데이터베이스를 공용하기 위한 데이터 제어를 정의하고 기술하는 언어는?

- ① DDL(data definition language)
- ② DML(data manipulation language)
- ③ DCL(data control language)
- ④ DUL(data user language)

[기출 예상 문제]

2. 다음 SQL 명령어 중 유형이 다른 하나는?

- ① Insert ② Update
- ③ Delete ④ Create

[기출 예상 문제]

3. 다음은 무엇에 대한 설명인가?

현재 트랜잭션을 작게 분할하는 명령어로, 지정된 특정 지점까지 Rollback 할 수 있는 명령어이다.

- ① Savepoint ② Transaction
- ③ Commit ④ Redo

[기출 예상 문제]

4. Select 명령어의 옵션에 대한 설명으로 가장 거리가 적절한 것은?

- ① alias: 해당 column에 대해서 다른 이름을 부여할 때 사용
- ② having: 그룹핑하고자 하는 단위 지정
- ③ group by: 그룹핑한 결과값에 대한 조건 검색
- ④ *: 중복되는 행을 제거

[기출 예상 문제]

5. PL/SQL에 대한 설명으로 가장 거리가 먼 것은?

- ① Exception 처리 루틴을 이용하여 Error처리가 가능하다.
- ② If문, Loop 문 등 절차적 언어 구조로 된 프로그램 작성 가능하다.
- ③ Script 생성 및 변경 후 Compile과정을 거쳐 실행된다.
- ④ 변수, 상수 등 식별자를 선언하여 해당 식별자를 SQL과 절차적인 프로그램에서 사용할 수 있다.

[기출 예상 문제]

6. PL/SQL 구조가 아닌 것은?

- ① 선언부(Declare) ② 예외 처리부(Exception)
- ③ 실행부(Begin/End) ④ 에러부(Error)

[소프트웨어 개발>데이터 입출력 구현>데이터 조작 프로시저 작성]

[기출 예상 문제]

7. 다음은 무엇에 대한 설명인가?

- 보통 값을 계산하고 결과값을 반환하기 위해서 많이 사용한다.
- IN 파라미터만 사용할 수 있다.
- 반드시 반환될 값의 데이터 타입을 Return문에 선언해야 한다.
- PL/SQL블록 내에서 Return문을 통해서 반드시 값을 반환해야 한다.

- ① Stored Function ② Stored Procedure
- ③ Stored Package ④ Stored Class

[기출 예상 문제]

8. Trigger에 대한 설명으로 가장 거리가 먼 것은?

- ① VIEW에 대해서가 아니라 TABLE에 관해서만 정의될 수 있다.
- ② TABLE과 함께 데이터베이스 저장된다.
- ③ DML 명령어가 TABLE에 행해질 때 묵시적으로 수행되는 프로시저이다.
- ④ 'set serveroutput on'을 사용하여 출력한다.

[기출 예상 문제]

9. SQL과 SQL*Plus의 차이점으로 가장 적절한 것은?

- ① SQL*Plus는 SQL명령어를 서버에 전송하는 도구이다.
- ② SQL은 Oracle사 제공 도구이다.
- ③ SQL은 SQL buffer를 사용하지 않는다.
- ④ SQL*Plus는 키워드를 축약하여 사용할 수 없다.

[기출 예상 문제]

10. SQL*Plus 명령어 중 SQL script를 실행할 때 사용하는 명령어는?

- ① / ② @ ③ N ④ RUN

[기출 예상 문제]

11. SQL*Plus 명령어에 대한 설명으로 가장 거리가 먼 것은?

- ① GET {파일명} : 파일의 내용을 버퍼로 읽어온다.
- ② SPOOL {파일명}: 버퍼의 내용을 편집기로 불러온다.
- ③ SAVE {파일명}: 버퍼의 내용을 파일에 저장한다.
- ④ START {파일명}: 저장된 SQL script를 실행한다.

[기출 예상 문제]

12. SQL*Plus 명령어 중 버퍼에 새로운 행을 추가할 때 사용하는 명령어는?

- ① C ② L ③ I ④ A

[소프트웨어 개발>데이터 입출력 구현>데이터 조작 프로시저 작성]

[기출 예상 문제]

13. PL/SQL 처리 결과를 화면에 출력하기 위해 사용하는 패키지로 가장 적절한 것은?

- ① printf ② DBMS_OUTPUT
- ③ System.out.println ④ Show

[기출 예상 문제]

14. 저장 객체 테스트에 대한 설명으로 가장 적절한 것은?

- ① Stored Function: EXECUTE문을 이용해 함수를 실행하여 수행 결과를 확인 한다.
- ② Stored Procedure: DBMS_OUTPUT.PUT_LINE을 출력하여 수행 결과를 확인 한다.
- ③ Stored Package: RUN문을 이용해 함수 실행하여 수행 결과를 확인 한다.
- ④ Trigger: START문을 이용해 함수 실행하여 수행 결과를 확인 한다.

[기출 예상 문제]

15. 메시지 버퍼에 저장하는 명령어로 메시지의 마지막 라인 끝에 새로운 라인문자가 추가되는 명령어는?

- ① DBMS_OUTPUT.ENABLE
- ② DBMS_OUTPUT.PUT
- ③ DBMS_OUTPUT.PUT_LINE
- ④ DBMS_OUTPUT.GET_LINE

[소프트웨어 개발>데이터 입출력 구현>데이터 조작 프로시저 최적화]

[기출 예상 문제]

1. SQL처리 흐름으로 가장 거리가 먼 것은?

- ① 구문분석: 사용자가 요청한 SQL문이 실행계획이 있는지 확인한다.
- ② 실행: 실행 계획에 따라서 메모리 영역의 데이터베이스 버퍼 캐시 영역에 해당 데이터가 있는지 확인한다.
- ③ 실행: 실행계획이 있다면 과거에 사용된 문장이기 때문에 구문분석을 할 필요없이 실행한다.
- ④ 추출: 서버 프로세스는 데이터버퍼 캐시영역에서 관련 테이블 데이터를 읽어서 클라이언트로 보내준다.

[기출 예상 문제]

2. 쿼리 성능 최적화를 위한 고려사항으로 가장 거리가 먼 것은?

- ① 개발자는 SQL 특성을 충분히 이해하고 SQL문을 적절히 구사할 수 있어야 한다.
- ② 구문분석 단계에서 옵티마이저의 실행 계획에 따라서 실행 속도의 차이는 크게 날 수 있다.
- ③ 개발자는 SQL 작성 시 옵티마이저의 일련의 행위를 이해하고 있어야 한다.
- ④ 옵티마이저의 실행계획이 비정상 적이라면 Start 명령어를 사용하여 실행계획을 수정할 수 있다.

[기출 예상 문제]

3. 다음은 무엇에 대한 설명인가?

Oracle DBMS 에서 실행되는 SQL문장에서 분석정보를 제공하여 개발자가 특정 SQL문장을 어떻게 시용해야 할 것인지에 대한 가이드라인을 제공하는 도구이다.

- ① Practice SQL ② TKPROF
- ③ EXPLAIN PLAN ④ OPTIMIZER

[기출 예상 문제]

4. TKPROF 결과의 내용이 아닌 것은?

- ① Rows ② Call ③ Parse ④ Data

[기출 예상 문제]

5. TKPROF 를 활용하여 쿼리의 성능을 측정할 때 모든 SQL 수행에 대한 Trace 파일을 생성하는 유형은?

- ① Instance Level ② Session Level
- ③ Enable Level ④ Disable Level

[소프트웨어 개발>데이터 입출력 구현>데이터 조작 프로시저 최적화]

[기출 예상 문제]

6. EXPLAIN PLAN 실행 시 PLAN_TABLE 결과 항목의 의미로 가장 거리가 먼 것은?

- ① Recursive call: 재귀 호출의 횟수
- ② DB block gets: 현재의 블록이 요구된 횟수
- ③ Redo size: Redo 로그가 만들어진 크기
- ④ Row processed: 클라이언트로 보내진 바이트 수

[기출 예상 문제]

7. 다음은 무엇에 대한 설명인가?

사용자들이 SQL문의 액세스 경로를 확인하여 성능개선을 할 수 있도록 SQL문을 분석하고 해석하여 실행 계획을 수립하고, 관련 테이블(plan_table)에 저장하도록 지원해 주는 도구이다.

- ① Practice SQL ② TKPROF
- ③ EXPLAIN PLAN ④ OPTIMIZER

[기출 예상 문제]

8. Autotrace Mode 옵션 중 실행계획과 통계정보만을 제공하는 옵션으로 가장 적절한 것은?

- ① TRACEONLY ② ON
- ③ OFF ④ PLANONLY

[기출 예상 문제]

9. TKPROF 결과 항목의 의미로 가장 거리가 먼 것은?

- ① Parse: 커서를 파싱하고 실행계획을 생성하는 통계
- ② Count: 커서의 실행(insert/update/delete) 단계에 대한 통계
- ③ CPU: 현재 커서가 각 단계에서 사용한 CPU 시간(초)
- ④ Elapsed: 현재 커서가 각 단계의 시작에서 종료까지 총 경과 시간(초)

[기출 예상 문제]

10. SQL 성능 개선순서에 대한 설명으로 가장 거리가 먼 것은?

- ① 실행 계획은 개발자가 작성한 SQL을 가장 빠르고 효율적으로 수행할 최적의 처리 경로를 생성해 주는 데이터베이스 핵심 모듈이다.
- ② 가능한 한 where 절을 많이 써서 범위가 아닌 특정 값 지정으로 범위를 줄여 처리속도를 빠르게 하도록 한다.
- ③ 성능에 중요한 액세스 경로를 고려하여 인덱스를 재구성 한다.
- ④ 문제 있는 SQL을 식별하기 위해 Oracle의 경우 TKPROF 또는 SQL_Trace와 같은 유틸리티를 사용한다.

[소프트웨어 개발>데이터 입출력 구현>데이터 조작 프로시저 최적화]

[기출 예상 문제]

11. SQL 코드 인스펙션 대상으로 가장 거리가 먼 것은?

- ① 사용되지 않는 변수
- ② Null 값과 비교하는 코드
- ③ 과거의 데이터 타입을 사용
- ④ DBMS 용량

[기출 예상 문제]

12. SQL 코드 인스펙션 절차로 가장 적절한 것은?

- | | |
|------------------|--------|
| ㄱ. 추적(Follow-up) | ㄴ. 검사 |
| ㄷ. 준비 | ㄹ. 계획 |
| ㅁ. 개관(Overview) | ㅂ. 재작업 |

- ① ㄹ-ㄱ-ㄷ-ㄴ-ㅂ-ㄱ
- ② ㄷ-ㄹ-ㅂ-ㄴ-ㄱ-ㄱ
- ③ ㄷ-ㄱ-ㄹ-ㄴ-ㄱ-ㅂ
- ④ ㄷ-ㄱ-ㄹ-ㄱ-ㄴ-ㅂ

[기출 예상 문제]

13. SQL*Plus에서 Stored Procedure소스를 확인하는 방법은?

- ① Start 명령어를 사용한다.
- ② DML 명령어를 사용한다.
- ③ USER_SOURCE 데이터 사전을 이용한다.
- ④ 옵티마이저를 실행한다.

[정답] [소프트웨어 개발>데이터 입출력 구현]

1. 논리 데이터저장소 확인

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2	1	2	2	3	3	3	3	3	3	4	3	1	4	1	3	3	4	2	1
21	22																		
3	4																		

2. 물리 데이터저장소 설계

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18		
3	1	4	1	4	1	2	3	4	3	1	2	4	1	1	4	3	3		

3. 데이터 조작 프로시저 작성

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15					
3	4	1	1	3	4	1	2	1	2	2	3	2	1	3					

4. 데이터 조작 프로시저 최적화

1	2	3	4	5	6	7	8	9	10	11	12	13							
3	4	2	4	1	4	3	1	2	1	4	1	3							

[정답 및 해설] [소프트웨어 개발>데이터 입출력 구현>논리 데이터저장소 확인]

논리 데이터저장소 확인 1. ②

- 선형구조: 리스트(선형 리스트(Linear List), 연결 리스트(Linked List)), 스택(Stack), 큐(Queue), 덱(Deque)
- 비선형구조: 트리(Tree), 그래프(Graph)

논리 데이터저장소 확인 2. ②

- ② 완전 이진트리: 마지막 레벨을 제외하고 노드가 모두 채워져 있는 트리이다.

논리 데이터저장소 확인 3. ②

- ② 연결을 위한 포인터를 찾는 시간이 필요하기 때문에 검색 속도가 순차리스트에 비해 느리다.

논리 데이터저장소 확인 4. ②

- ② A가 B보다 먼저 출력될 수 없다.

논리 데이터저장소 확인 5. ③

- ㄴ. 두개의 포인트를 사용한다.

논리 데이터저장소 확인 6. ③

- ③ 트리(Tree)는 비선형구조이다.

논리 데이터저장소 확인 7. ③

- ① node는 14이다.
- ② tree의 차수는 3이다.
- ④ 근 node는 A다.

논리 데이터저장소 확인 8. ③

- 단말 노드(Leaf node)는 자식이 없는 노드를 의미한다.

논리 데이터저장소 확인 9. ③

- 전위 순회는 중간 노드를 방문하고 왼쪽 서브트리를 전위 순회한 후 오른쪽 서브트리를 전위 순회 한다.

논리 데이터저장소 확인 10. ③

- 간선(edge)은 정점(vertex)을 연결하는 선으로 6개이다.

논리 데이터저장소 확인 11. ④

- 역파일(inverted file)은 각 필드마다 인덱스를 만들어서 구현하므로 기억 공간이 늘어난다.

[정답 및 해설] [소프트웨어 개발>데이터 입출력 구현>논리 데이터저장소 확인]

논리 데이터저장소 확인 12. ③

③ 다중리스트 파일(multilist file)에 대한 설명이다.

논리 데이터저장소 확인 13. ①

① 색인 영역은 트랙 색인 영역, 실린더 색인 영역, 마스터 색인 영역으로 구분할 수 있다.

논리 데이터저장소 확인 14. ④

DAM은 해싱 함수를 계산해서 물리적 주소를 직접 접근하는 방법으로 순서에 관계 없이 저장하므로 기억 공간 효율이 저하된다.

논리 데이터저장소 확인 15. ①

- ② 정적 인덱스 방법에 대한 설명이다.
- ③ 동적 인덱스 방법에 대한 설명이다.
- ④ 동적 인덱스의 방법에서 일정 수를 레코드를 유지하지 못하는 블록은 다른 블록에 합병(merge)시킨다. 한 블록이 가득차게 될 때 이를 둘로 분할(split)시킨다.

논리 데이터저장소 확인 16. ③

- 데이터 모델링 절차
사용자 요구사항→개념 데이터 모델링→논리 데이터 모델링→물리 데이터 모델링→데이터베이스 구축

논리 데이터저장소 확인 17. ③

논리적 설계에서 목표 DBMS에 맞추어 논리적 모델로 설계하고 정규화 과정을 수행한다.

논리 데이터저장소 확인 18. ④

④ 데이터베이스 설계 단계 중 개념적 설계에서 개념 스키마를 설계한다.

논리 데이터저장소 확인 19. ②

② 논리 데이터저장소를 설계하는 과정이 논리 데이터 모델링이다.

논리 데이터저장소 확인 20. ①

- 논리 데이터저장소 확인 절차
- 엔티티 및 속성 확인: 업무처리를 위한 데이터 도출 확인, 엔티티 중복 최소화, 데이터 무결성 확인
 - 관계 확인: 엔티티 간의 관계 확인, 정규화 수준 점검
 - 데이터 흐름 확인: 프로세스별 CRUD Matrix 확인
 - 데이터 접근 권한 확인: 엔티티별 데이터 접근 권한 확인, 계정별 접근 제어 확인
 - 백업정책 및 분산구조 확인: 데이터 백업 및 복구 정책 확인, 계정별 접근 제어 확인

[정답 및 해설] [소프트웨어 개발>데이터 입출력 구현>논리 데이터저장소 확인]

논리 데이터저장소 확인 21. ③

- 엔티티: 관리할 대상이 되는 실체
- 속성: 관리할 정보의 구체적 항목
- 관계: 엔티티간의 대응관계

논리 데이터저장소 확인 22. ④

- ④ 자주 사용되는 코드성 데이터는 공통코드 대상이다.

[정답 및 해설] [소프트웨어 개발>데이터 입출력 구현>물리 데이터저장소 설계]

물리 데이터저장소 설계 1. ③

데이터베이스 스키마

- 외부(External) 스키마: 프로그래머나 사용자의 입장에서 데이터베이스의 모습으로 조직의 일부분을 정의한 것
- 개념(Conceptual) 스키마: 모든 응용 시스템과 사용자들이 필요로 하는 데이터를 통합한 조직 전체의 데이터베이스 구조를 논리적으로 정의한 것
- 내부(Internal) 스키마: 전체 데이터베이스의 물리적 저장 형태를 기술한 것

물리 데이터저장소 설계 2. ①

- ① 속성(Attribute)을 컬럼(Column)으로 변환한다.

물리 데이터저장소 설계 3. ④

삭제 제약 조건>Delete Constraint

- Cascade: 참조한 테이블에 있는 외부키와 일치하는 모든 Row가 삭제
- Restricted: 참조한 테이블에 있는 외부키에 없는 것만 삭제 가능
- Nullify: 참조한 테이블에 정의된 외부키와 일치하는 것을 Null로 수정

물리 데이터저장소 설계 4. ①

- ① 인덱스 컬럼은 가능한 한 수정이 빈번하지 않은 컬럼을 선정한다.

물리 데이터저장소 설계 5. ④

뷰 속성

- REPLACE: 뷰가 이미 존재하는 경우 재생성한다.
- FORCE: 기본 테이블의 존재 여부에 관계없이 뷰 생성을 한다.
- NOFORCE: 기본 테이블이 존재할 때만 뷰를 생성한다.
- WITH CHECK OPTION: Sub-Query 내의 조건을 만족하는 행만 변경한다.
- WITH READ ONLY: DML 작업이 불가능한 뷰를 생성한다.

물리 데이터저장소 설계 6. ①

- ① 분포도가 넓을수록 유리한 기법이다.(인덱스의 단점을 해결)

물리 데이터저장소 설계 7. ②

- 파티션 종류에는 범위 분할, 해시 분할, 조합 분할이 있다.

[정답 및 해설] [소프트웨어 개발>데이터 입출력 구현>물리 데이터저장소 설계]

물리 데이터저장소 설계 8. ③

수정 제약 조건(Update Constraint)

- Cascade: 참조한 테이블에 있는 외부키와 일치하는 모든 Row가 수정
- Restricted: 참조한 테이블에 있는 외부키에 없는것만수정 기능
- Nullify: 참조한 테이블에 정의된 외부키와 일치하는 것을 Null로 수정

물리 데이터저장소 설계 9. ④

④ 파티션 키 값 변경에 대한 별도 관리가 필요하다.

물리 데이터저장소 설계 10. ③

객체 관계 매핑(Object-Relational Mapping; ORM)에 대한 설명이다.

물리 데이터저장소 설계 11. ①

① 1:M 관계는 1 관계에 있는 테이블에 있는 주키를 M 관계 테이블에 외부키로 매핑한다.

물리 데이터저장소 설계 12. ②

- ① SQL Mapping 기반 기술은 Partial ORM이다. Full ORM은 OR Mapping 기반 기술이다.
- ③ OR Mapping 기반 기술의 매퍼는 Object Relation Mapper이다. SQL Mapper는 SQL Mapping 기반 기술의 매퍼이다.
- ④ OR Mapper가 SQL Mapper에 비해 더 효율적이다.

물리 데이터저장소 설계 13. ④

Generalization 관계와 테이블간 변환 방법

- 1안: 상위 클래스와 각 하위 클래스를 각각 별도의 테이블로 변환
- 2안: 상위 클래스가 하위 클래스의 모든속성을 포함하여 단일 테이블로 변환
- 3안: 하위 클래스들이 상위 클래스의 속성을 상속 받아 각각 하위 클래스들을 테이블로 매핑

물리 데이터저장소 설계 14. ①

데이터베이스 트랜잭션 특성에는 원자성(Atonomicity), 일관성(Consistency), 고립성(Isolation), 영속성(Durability)가 있다.

[정답 및 해설] [소프트웨어 개발>데이터 입출력 구현>물리 데이터저장소 설계]

물리 데이터저장소 설계 15. ①

원자성(Atonomicity)는 트랜잭션 연산을 데이터베이스 모두 반영되든지 아니면 전혀 반영 되지 않아야 하는 특성이다. (All or Nothing)

물리 데이터저장소 설계 16. ④

ROLLBACK 연산은 하나의 트랜잭션 처리가 비정상적 종료되어 해당 트랜잭션을 재시작 혹은 폐기하여 원래의 상태로 원상 복구하는 연산이다.

물리 데이터저장소 설계 17. ③

데이터베이스 트랜잭션 특징

- 원자성(Atonomicity): 트랜잭션 연산을 데이터베이스 모두 반영되든지 아니면 전혀 반영 되지 않아야 한다.
- 일관성(Consistency): 트랜잭션이 그 실행을 성공적으로 완료하면 언제나 일관성 있는 데이터베이스 상태로 변환된다.
- 독립성(Isolation): 둘 이상의 트랜잭션이 동시에 병행 실행되는 경우 어느 하나의 트랜잭션 실행 중에 다른 트랜잭션의 연산이 끼어들 수 없다.
- 영속성(Durability): 성공적으로 완료된 트랜잭션의 결과는 시스템이 고장나더라도 영구적으로 반영된다.

물리 데이터저장소 설계 18. ③

COMMIT 연산은 하나의 트랜잭션에 대한 작업이 성공적으로 끝나고 수행 결과를 관리자에게 알려주는 연산이다.

[정답 및 해설] [소프트웨어 개발>데이터 입출력 구현>데이터 조작 프로시저 작성]

데이터 조작 프로시저 작성 1. ③

- 데이터 정의어(DDL: Data Definition Language)
: 데이터를 저장하고 있는 테이블 등의 구조를 생성하고 변경하기 위하여 사용되는 명령어들을 의미한다.
- 데이터 조작어(DML: Data Manipulation Language)
: 데이터베이스에 있는 데이터를 변경하거나 검색하기 위하여 사용되는 명령어들을 의미한다.
- 데이터 제어어(DCL: Data Control Language)
: 사용자별로 데이터베이스에 접근할 수 있는 권한을 부여하거나 회수하는 명령어들을 말한다.

데이터 조작 프로시저 작성 2. ④

- ①, ②, ③ DML(Data Manipulation Language)이다.
- ④ DCL(Data Definition Language)이다.

데이터 조작 프로시저 작성 3. ①

Savepoint에 대한 설명이다. Savepoint A; 이렇게 명령할 경우 A지점까지 Rollback된다.

데이터 조작 프로시저 작성 4. ①

- ② having: 그룹핑한 결과값에 대한 조건 검색
- ③ group by: 그룹핑하고자 하는 단위 지정
- ④ *: 테이블의 모든 Column을 출력
- distinct: 중복되는 행을 제거

데이터 조작 프로시저 작성 5. ③

- ③ Compile이 필요 없어 Script 생성 및 변경 후 바로 실행이 가능하다.

데이터 조작 프로시저 작성 6. ④

PL/SQL구조

- 선언부(Declare): 실행부에서 참조할 모든 변수, 상수, Cursor, Exception을 선언한다.
- 실행부(Begin/End): Begin과 End사이에 기술되는 영역으로, 데이터베이스 데이터를 처리할 SQL문과 PL/SQL 블록을 기술한다.
- 예외 처리부(Exception): 실행부에서 에러가 발생했을 경우 수행될 문장을 기술한다.

데이터 조작 프로시저 작성 7. ①

Stored Function에 대한 설명이다.

데이터 조작 프로시저 작성 8. ②

- ② TABLE과는 별도로 데이터베이스에 저장된다.

데이터 조작 프로시저 작성 9. ①

- ② SQL*Plus에 대한 설명이다.
- ③ SQL은 SQL buffer를 사용한다.
- ④ SQL*Plus는 키워드를 축약할 수 있다.

[정답 및 해설] [소프트웨어 개발>데이터 입출력 구현>데이터 조작 프로시저 작성]

데이터 조작 프로시저 작성 10. ②

- ① /: 버퍼의 내용을 실행한다.
- ③ N(숫자): 현재 행을 출력한다.
- ④ RUN {파일명}: 버퍼의 내용을 실행한다.

데이터 조작 프로시저 작성 11. ②

- ② SPOOL {파일명}: 조회결과를 파일로 저장한다.

데이터 조작 프로시저 작성 12. ③

- ① C: 현재 행의 문자열을 치환한다.
- ② L: 버퍼의 전체 리스트를 출력한다.
- ④ A {문자스tring}: 현재 버퍼의 끝에 새로운 문자 스트링을 추가한다.

데이터 조작 프로시저 작성 13. ②

DBMS_OUTPUT 패키지는 메시지를 버퍼에 저장하고 버퍼로부터 메시지를 읽어오기 위한 인터페이스를 제공한다.

데이터 조작 프로시저 작성 14. ①

- ② Stored Procedure: EXECUTE문 이용
- ③ Stored Package: DBMS_OUTPUT.PUT_LINE 출력
- ④ Trigger: DBMS_OUTPUT.PUT_LINE 출력

데이터 조작 프로시저 작성 15. ③

- ① DBMS_OUTPUT.ENABLE: 메시지 버퍼 내용 할당
- ② DBMS_OUTPUT.PUT: 메시지 버퍼에 저장. 마지막 라인 끝에 새로운 라인 문자가 추가되지 않음.
- ④ DBMS_OUTPUT.GET_LINE: 한번 호출될 때 마다 하나의 라인만을 읽어옴

[정답 및 해설] [소프트웨어 개발>데이터 입출력 구현>데이터 조작 프로시저 최적화]

데이터 조작 프로시저 최적화 1. ③

③ 실행계획은 구문분석 단계에서 확인하며, 구문분석 단계에서 실행계획 확인 후 실행 계획이 있다면 구문분석하지 않고 바로 실행하고 실행계획이 없다면 옵티마이저가 실행계획을 찾는다.

데이터 조작 프로시저 최적화 2. ④

④ 옵티마이저의 실행계획이 비정상적이라면 개발자는 Hint같은 조건을 부여하여 실행계획을 수정할 수 있다.

데이터 조작 프로시저 최적화 3. ②

TKPROF(Trace Kernel PROFile)에 대한 설명이다.

데이터 조작 프로시저 최적화 4. ④

TKPROF 결과에는 call, count, cpu, elapsed, disk, query, current, rows 등이 있다.

데이터 조작 프로시저 최적화 5. ①

TKPROF를 활용한 Trace 유형

- Instance Level 추적: 모든 SQL 수행에 대한 Trace 파일을 생성하여 부하가 많다.
- Session Level 추적: 특정 프로세스별로 추적 파일을 생성한다.

데이터 조작 프로시저 최적화 6. ④

④ Row processed: 연산을 하는 동안 처리한 Row 수 / Byte sent via SQL *Net to client: 클라이언트로 보내진 바이트 수

데이터 조작 프로시저 최적화 7. ③

EXPLAIN PLAN에 대한 설명이다.

데이터 조작 프로시저 최적화 8. ①

Autotrace Mode

- On: SQL문의 실행 결과와 실행계획과 통계정보를 보여주는 옵션
- Off: AUTOTRACE를 해지하는 옵션
- TRACEONLY: 실행계획과 통계정보만을 제공하는 옵션

데이터 조작 프로시저 최적화 9. ②

② Execute: 커서의 실행(insert/update/delete) 단계에 대한 통계 / Count: Parse, Execute, Fetch 각 단계가 수행된 횟수

데이터 조작 프로시저 최적화 10. ①

① 옵티마이저(Optimizer)에 대한 설명이다.

[정답 및 해설] [소프트웨어 개발>데이터 입출력 구현>데이터 조작 프로시저 최적화]

데이터 조작 프로시저 최적화 11. ④

SQL 코드 인스펙션 대상

- 사용되지 않는 변수(unused variable): 프로시저에서 선언은 되었지만 본문에서는 전혀 사용되지 않는 변수
- 사용되지 않는 서브쿼리(unused subquery item): 컬럼이 선언은 되었지만 외부 쿼리 (outer query)에서 참조가 되지 않음
- Null 값과 비교 (Null comparison): Null 값과 비교하는 프로시저 소스가 있는 경우
- 과거의 데이터타입을 사용 (Deprecated type): 데이터 타입이 바뀌었지만 과거의 타입을 그대로 쓰는 소스가 있는 경우

데이터 조작 프로시저 최적화 12. ①

SQL 코드 인스펙션 절차

1. 계획: 문제되는 SQL 코드 선별, 문제점 인식 및 인스펙션 참여자 선정
2. 개관(OverView): SQL 코드 문제점 공유(실행시간, 자원사용량 등), 계획 및 방법 공유
3. 준비: 각자 SQL 소스코드분석, 문제점 확인
4. 검사: 공식적인 SQL 인스펙션 수행, 문제점 토의
5. 재작업: SQL 소스코드 수정 및 실행 시간 재측정
6. 추적(Follow-up): 개선 효과 분석

데이터 조작 프로시저 최적화 13. ③

SQL*Plus에서 Stored Procedure 소스 확인 방법

- USER_SOURCE 데이터 사전을 이용:
USER_SOURCE 데이터 사전을 이용하면 사용자가 만든 모든 프로시저를 보여준다.
- 프로시저 소스 확인: 프로시저 소스는 USER_SOURCE의 text 조회를 통하여 확인한다.