

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

## 데이터 조작 프로시저 작성 1. (ㄴ)-(㉠)-(㉡)-(㉢)-(㉣)

### \* 데이터 조작 프로시저 개발 수행 순서

- 1단계: 데이터 저장소 연결
- 2단계: 데이터 저장소 정의
- 3단계: 데이터 조작 프로시저 작성
- 4단계: 데이터 검색 프로시저 작성
- 5단계: 절차형 데이터 조작 프로시저 작성

## 데이터 조작 프로시저 작성 2. SQL\*Plus

### \* SQL\* Plus

- Oracle DBMS는 데이터 조작 프로시저에 대한 테스트 환경으로 SQL\*Plus 라는 도구를 제공한다.
- SQL은 데이터를 조작하는 표준 언어인 반면 SQL\*Plus 는 이러한 SQL을 DBMS 서버에 전송하여 처리할 수 있도록 하는 Oracle에서 제공하는 도구인 것이 가장 큰 차이점이라 할 수 있다.

## 데이터 조작 프로시저 작성 3. SELECT, INSERT, UPDATE, DELETE

### \* DML의 유형

- SELECT: 테이블에서 튜플을 검색한다.
- INSERT: 테이블에 새로운 튜플을 삽입한다.
- DELETE: 테이블에서 튜플을 삭제한다.
- UPDATE: 테이블에서 튜플의 내용을 갱신한다.

## 데이터 조작 프로시저 작성 4. DDL

### \* 데이터 정의어(DDL)

- DB 구조 데이터 형식 접근 방식 등 DB를 구축하거나 수정할 목적으로 사용하는 언어이다.
- 번역한 결과가 데이터 사전 (Data Dictionary)이라는 특별한 파일에 여러 개의 테이블로 저장된다.
- CREATE, DROP, RENAME, ALTER, TRUNCATE 등이 있다.

## 데이터 조작 프로시저 작성 5. ① (ㄴ) ② (㉢) ③ (㉡) ④ (㉠)

### \* PL/SQL을 활용한 저장형 객체

- Stored Procedure (프로시저) : 보통 연속 실행 또는 구현이 복잡한 트랜잭션을 수행하는 PL/SQL블록을 DB에 저장하기 위해 생성한다.
- Stored Function (사용자 정의 함수) : PL/SQL블록 내에서 RETURN문을 통해서 반드시 값을 반환해야 한다.
- Stored Package : 패키지는 오라클 데이터베이스에 저장되어 있는 서로 관련 있는 PL/SQL 프로시저와 함수들의 집합이다.
- Trigger : INSERT, UPDATE, DELETE문이 TABLE에 대해 행해질 때 묵시적으로 수행되는 PROCEDURE이다.

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

### 데이터 조작 프로시저 작성 6. (ㄷ)-(ㄴ)-(ㄱ)-(ㄹ)

#### \* 데이터 저장소 연결 순서

- [1단계]드라이버 로딩: DB와 연결하기 위해 DBMS에서 제공하는 jar파일 드라이버를 메모리에 적재한다.

: oracle.jdbc.driver.OracleDriver;

- [2단계]Connection: 해당 드라이버를 사용하여 DB를 연결한다.

: conn=DriverManager.getConnection(url,"scott","tiger");

- [3단계]쿼리 전달: 쿼리를 DB로 전달하기 위해 Statement, PreparedStatement 객체를 생성한다.

: pstmt=conn.prepareStatement(sql);

- [4단계]결과 수신 : 전달된 쿼리의 수행으로 인한 반환 값을 수신한다.

: ResultSet rs=pstmt.executeQuery();

### 데이터 조작 프로시저 작성 7. ① SQL ② SQL\*Plus

SQL은 데이터를 조작하는 표준 언어인 반면 SQL\*Plus는 이러한 SQL을 DBMS 서버에 전송하여 처리할 수 있도록 하는 Oracle에서 제공하는 도구이다.

### 데이터 조작 프로시저 작성 8. EDIT 또는 ED

- ED[IT]: 버퍼의 내용을 편집기로 불러온다.

### 데이터 조작 프로시저 작성 9. SPOOL GISA.TXT 또는 SPO GISA.TXT

- SPO[OL] 파일명: 작업 내용을 파일에 기록한다. SPOOL 을 종료하려면 SPOOL OFF 명령을 실행한다.

### 데이터 조작 프로시저 작성 10. GET GISA.TXT

- GET 파일명: 파일의 내용을 SQL 버퍼로 읽어온다.

### 데이터 조작 프로시저 작성 11. START GISA.TXT 또는 STA GISA.TXT 또는 @GISA.TXT

- STA[RT] 파일명: 파일에 저장된 내용을 실행한다. @와 동일하다.

### 데이터 조작 프로시저 작성 12. DBMS\_OUTPUT

- DBMS\_OUTPUT 패키지 활용

: 메시지를 버퍼에 저장하고 버퍼로부터 메시지를 읽어오기 위한 인터페이스를 제공하는 패키지인 DBMS\_OUTPUT을 코드에 포함하여야 한다.

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

### 데이터 조작 프로시저 작성 13. ENABLE

- \* DBMS\_OUTPUT 패키지 제공 메소드
- ENABLE: 메시지 버퍼 내용 할당 (화면 출력)

### 데이터 조작 프로시저 작성 14. SERVEROUTPUT

PL/SQL의 처리결과를 화면에 출력하기 위한 SERVEROUTPUT을 ON 시키고, 실행하고자 하는 PL/SQL 블록 또는 저장객체명을 호출한다.

### 데이터 조작 프로시저 작성 15. SHOW ERRORS

PL/SQL 실행 시 오류가 발생하면 'SHOW ERRORS' 명령어를 통해 오류내용을 확인하고 조치한다.

### 데이터 조작 프로시저 작성 16. ① (¬) ② (=) ③ (≡) ④ (⊂)

- \* 주요 SQL\*Plus 명령어
- L[IST]: 버퍼 내용 확인
- R[UN]: 버퍼 내용 실행
- A[PPEND] text: 버퍼의 끝에 text를 추가한다.
- CL[EAR] BUFF[ER]: 버퍼의 내용을 삭제한다.

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

## 데이터 조작 프로시저 최적화 1. APM

- \* APM(Application Performance Management (Monitoring))
  - 운영 중인 시스템에 대한 가용성 확보, 다운타임 최소화 등을 통해 안정적인 시스템 운영을 위하여, 부하량과 접속자 파악 및 장애진단 등을 목적으로 하는 성능 모니터링 도구이다.
  - 쿼리 성능을 최적화하기 전에 필요한 애플리케이션 성능 모니터링 도구로 최적화 할 쿼리를 선정해야 한다.

## 데이터 조작 프로시저 최적화 2. TKPROF, EXPLAIN PLAN

- \* TKPROF, EXPLAIN PLAN  
: 성능 모니터링 결과 문제시되는 SQL에 대한 처리흐름, 그리고 해당 SQL이 DBMS내에서 어떻게 동작하는지를 파악할 수 있는 쿼리 성능 측정 방법

## 데이터 조작 프로시저 최적화 3. ① 리소스 ②엔드 투 엔드(End to End)

- APM에는 크게 리소스 모니터링, 엔드 투 엔드 모니터링 방식이 있다.

## 데이터 조작 프로시저 최적화 4. 옵티마이저(Optimizer)

- \* 옵티마이저(Optimizer)
  - 옵티마이저는 SQL을 가장 빠르고 효율적으로 수행할 최적의 처리경로를 생성해 주는 DBMS 내부의 핵심 엔진이다. 이러한 최적의 실행 방법을 실행계획(Execution Plan)이라고 한다.
  - 최적의 실행 방법을 결정하는 방식에 따라 규칙기반 옵티마이저(우선순위)와 비용기반 옵티마이저(수행 시간)로 구분할 수 있다.

## 데이터 조작 프로시저 최적화 5. (┐)-(┐)-(┐)

- \* SQL 처리 단계
  - 1단계: 구문분석 단계(PARSE)
  - 2단계: 실행 단계(EXECUTE)
  - 3단계: 추출 단계(FETCH)

## 데이터 조작 프로시저 최적화 6. (┐)

- \* 구문분석 단계(PARSE)  
SQL문이 데이터베이스에서 처음 사용된 문장인지 이미 사용된 문장인지를 확인한다. 이미 사용된 문장이라면 구문분석(Parsing, SQL 문법 검사)이라는 작업을 할 필요가 없다.

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

데이터 조작 프로시저 최적화 7. 데이터베이스 버퍼 캐시(DB Buffer Cache)

### \* DB Buffer Cache

- 사용자가 입력한 데이터를 파일에 저장하고 이를 다시 읽는 과정에서 거쳐가는 캐시영역(메모리 공간)이다.
- 가장 최근에 사용된 데이터 블록을 저장하는 메모리 공간이다.

데이터 조작 프로시저 최적화 8. (¬)

### \* 실행 단계(EXECUTE)

- 구문분석이 정상적으로 실행되면 서버 프로세스는 메모리 영역의 데이터베이스 버퍼 캐시영역을 검색하여 해당 테이블의 데이터가 다른 사용자의 다른 SQL 문에 의해 이미 데이터 버퍼 캐시영역에 존재하는지를 검색한다.

데이터 조작 프로시저 최적화 9. ① TKPROF ② EXPLAIN PLAN

- 모니터링 결과 문제시되는 SQL에 대한 처리흐름, 그리고 해당 SQL이 DBMS 내에서 어떻게 동작하는지를 파악할 수 있는 다양한 유틸리티(Oracle의 경우 TKPROF, EXPLAIN PLAN 등)를 활용해 그 결과를 분석하여 쿼리 성능을 개선할 수 있다.

데이터 조작 프로시저 최적화 10. ① (¬) ② (≡) ③ (¬) ④ (¬)

| 구분      | 내용  |
|---------|---|
| Call    | 커서 상태에 따라 Execute, Parse, Fetch 3개의 Call로 나뉨  |
| Count   | 문장이 분석되고 실행된 횟수. Query가 Parse, Execute, Fetch한 횟수   |
| Elapsed | 시작에서 종료까지 총 경과 시간(초)  |
| CPU     | CPU 소모 시간(초)  |
| Query   | 데이터 블록을 읽은 횟수   |
| Rows    | 읽은 총 행 수<br>- Execute 단계: INSERT, UPDATE, DELETE문에서 처리된 행수<br>- Fetch 단계: SELECT에 의해 질의된 행수 |

데이터 조작 프로시저 최적화 11. ① Instance ② Session

- Instance Level로 모든 SQL을 Trace하는 경우는 거의 없고, DB 응용프로그램에서 사용되는 특정 SQL에 대해서만 Trace하는 Session Level을 일반적으로 활용한다.

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

데이터 조작 프로시저 최적화 12. show parameter

\* 파라미터 설정 확인 순서

① Sys user(시스템 사용자 권한, sysdba)로 DB 접속  
: sqlplus "/ as sysdba"

② 파라미터 확인

: show parameter 또는 show parameter 파라미터명

데이터 조작 프로시저 최적화 13. ① PLAN Table ② PLUSTRACE

Autotrace 명령어를 사용하기 전에 반드시 PLAN TABLE이 생성되어야 하고, PLUSTRACE 권한을 가지고 있어야 한다.

데이터 조작 프로시저 최적화 14. ① (┐) ② (≡) ③ (⊃) ④ (⊂) ⑤ (┐)

\* PLAN\_TABLE 결과 항목 의미

- Recursive calls: 재귀 호출의 횟수 (반복 호출)

- DB block gets: 현재의 블록이 요구된 횟수

- Consistent gets: 한 블록에 대해 요구된 일관성 읽기 횟수

- Physical reads: 디스크로부터 읽어 들인 데이터 블록의 총 개수

- Row processed: 연산을 하는 동안 처리한 row 수

데이터 조작 프로시저 최적화 15. (┐)-(≡)-(⊃)-(┐)-(⊂)-(≡)

\* SQL 성능 개선 순서

- 1단계: 문제 있는 SQL 식별

- 2단계: 옵티마이저(Optimizer) 통계 확인

- 3단계: 실행계획 검토

- 4단계: SQL 문 재구성

- 5단계: 인덱스 재구성

- 6단계: 실행계획 유지관리

데이터 조작 프로시저 최적화 16. (┐)

\* SQL문 재구성

- 가능한 한 where= 을 많이 써서 범위가 아닌 특정 값 지정으로 인한 범위를 줄여 처리속도가 빠르도록 한다.

- where절의 칼럼에 연산자를 사용하여 칼럼변경이 발생하면 인덱스를 활용하지 못하게 됨을 이해하여 칼럼 변경 연산자를 쓰지 않도록 한다.

- 부분범위 처리의 경우 Sub-Query에 Exists 사용하여 불필요한 검색을 하지 않도록 한다.

- 옵티마이저가 비정상적인 실행계획을 수립하여 처리된다면, 힌트로서 옵티마이저의 액세스 경로 및 조인 순서를 제어할 수 있도록 한다.