



2020년 2회 정보처리기사 실기 시험 100% 합격전략집

1
일차

2
일차

3
일차

4
일차

5
일차

6
일차

7
일차

8
일차

9
일차

10
일차

11
일차

12
일차

13
일차

14
일차

15
일차

16
일차

17
일차

18
일차

19
일차

20
일차

8장 SQL 응용

핵심 113 DML – JOIN

핵심 114 DML – OUTER JOIN

핵심 115 프로시저(Procedure)

핵심 116 SQL – 트리거(Trigger)

핵심 117 SQL – 사용자 정의 함수

핵심 118 SQL – 커서(Cursor)



2020년 2회 정보처리기사 실기 대비용 핵심요약

2020 시나공 정보처리기사 필기 수록 내용

[핵심113] DML - JOIN

JOIN의 개념

- 2개의 테이블에 대해 연관된 튜플들을 결합하여, 하나의 새로운 릴레이션을 반환한다.
- JOIN은 크게 INNER JOIN과 OUTER JOIN으로 구분된다.
- JOIN은 일반적으로 FROM절에 기술하지만, 릴레이션이 사용되는 어느 곳에서나 사용할 수 있다.

INNER JOIN

- 가장 일반적인 JOIN의 형태로, 관계가 설정된 두 테이블에서 조인된 필드가 일치하는 행만을 표시한다.
- WHERE절을 이용한 표기 형식

```
SELECT [테이블명1.]속성명, [테이블명2.]속성명, ...
FROM 테이블명1, 테이블명2, ...
WHERE 테이블명1.속성명 = 테이블명2.속성명;
```

- NATURAL JOIN을 이용한 표기 형식

```
SELECT [테이블명1.]속성명, [테이블명2.]속성명, ...
FROM 테이블명1 NATURAL JOIN 테이블명2;
```

- JOIN ~ USING절을 이용한 표기 형식

```
SELECT [테이블명1.]속성명, [테이블명2.]속성명, ...
FROM 테이블명1 JOIN 테이블명2 USING(속성명);
```

<학생>

학번	이름	학과코드	선배	성적
15	고길동	com		83
16	이순신	han		96
17	김선달	com	15	95
19	아무개	han	16	75
37	박치민		17	55

<학과>

학과코드	학과명
com	컴퓨터
han	국어
eng	영어

예제 <학생> 테이블과 <학과> 테이블에서 학과코드 값이 같은 튜플을 JOIN하여 학번, 이름, 학과코드, 학과명을 출력하는 SQL문을 작성하시오.

- SELECT 학번, 이름, 학생.학과코드, 학과명
FROM 학생, 학과
WHERE 학생.학과코드 = 학과.학과코드;
- SELECT 학번, 이름, 학생.학과코드, 학과명
FROM 학생 NATURAL JOIN 학과;
- SELECT 학번, 이름, 학생.학과코드, 학과명
FROM 학생 JOIN 학과 USING(학과코드);

<결과>

학번	이름	학과코드	학과명
15	고길동	com	컴퓨터
16	이순신	han	국어
17	김선달	com	컴퓨터
19	아무개	han	국어

1. 다음은 <학생> 테이블과 <학과> 테이블에서 '학과코드' 속성의 값이 같은 자료 중 '이름'과 '학과명'을 검색하는 SQL문이다. 괄호(①~③)에 알맞은 명령어나 연산자를 넣어 SQL문을 완성하시오.

<학생>

학번	이름	학과코드	성적
15	고길동	han	73
16	이순신	com	96
37	전산오		83

<학과>

학과코드	학과명
com	컴퓨터
han	국문
eng	영어

<SQL문>

방법1	SELECT 이름, 학과명 AS 학과 FROM 학생, 학과 WHERE 학생.학과코드 (①) 학과.학과코드;
방법2	SELECT 이름, 학과명 AS 학과 FROM 학생 (②) 학과;
방법3	SELECT 이름, 학과명 AS 학과 FROM 학생 JOIN 학과 (③)(학과코드);



답

- ① :
- ② :
- ③ :

해설 1.

• 방법1

- SELECT 이름, 학과명 AS 학과 : '이름'과 '학과명'을 표시하되, '학과명'은 '학과'로 표시한다.
- FROM 학생, 학과 : <학생> 테이블과 <학과> 테이블을 대상으로 한다.
- WHERE 학생.학과코드 = 학과.학과코드; : <학생> 테이블의 '학과코드'와 <학과> 테이블의 '학과코드'가 같은 튜플을 JOIN한다.

• 방법2

- SELECT 이름, 학과명 AS 학과 : '이름'과 '학과명'을 표시하되, '학과명'은 '학과'로 표시한다.
- FROM 학생 NATURAL JOIN 학과; : <학생> 테이블과 <학과> 테이블에서 같은 속성을 기준으로 JOIN한다.

• 방법3

- SELECT 이름, 학과명 AS 학과 : '이름'과 '학과명'을 표시하되, '학과명'은 '학과'로 표시한다.
- FROM 학생 JOIN 학과 USING(학과코드); : <학생> 테이블과 <학과> 테이블에서 '학과코드'를 기준으로 JOIN한다.

정답 1. ① = ② NATURAL JOIN ③ USING

2020 시나공 정보처리기사 필기 수록 내용

[핵심114] DML - OUTER JOIN

- 릴레이션에서 JOIN 조건에 만족하지 않는 튜플도 결과로 출력하기 위한 JOIN 방법으로, LEFT OUTER JOIN, RIGHT OUTER JOIN 등이 있다.
- LEFT OUTER JOIN : INNER JOIN의 결과를 구한 후, 좌측 항 릴레이션의 어떤 튜플과도 맞지 않는 좌측 항의 릴레이션에 있는 튜플들에 NULL 값을 붙여서 INNER JOIN의 결과에 추가함
 - 표기 형식

```
SELECT [테이블명1].속성명, [테이블명2].속성명, ...
FROM 테이블명1 LEFT OUTER JOIN 테이블명2
ON 테이블명1.속성명 = 테이블명2.속성명;
```

- RIGHT OUTER JOIN : INNER JOIN의 결과를 구한 후, 좌측 항 릴레이션의 어떤 튜플과도 맞지 않는 우측 항의 릴레이션에 있는 튜플들에 NULL 값을 붙여서 INNER JOIN의 결과에 추가함

- 표기 형식

```
SELECT [테이블명1].속성명, [테이블명2].속성명, ...
FROM 테이블명1 RIGHT OUTER JOIN 테이블명2
ON 테이블명1.속성명 = 테이블명2.속성명;
```

<학생>

학번	이름	학과코드	선배	성적
15	고길동	com		83
16	이순신	han		96
17	김선달	com	15	95
19	아무개	han	16	75
37	박치민		17	55

<학과>

학과코드	학과명
com	컴퓨터
han	국어
eng	영어

예제 <학생> 테이블과 <학과> 테이블에서 학과코드 값이 같은 튜플을 JOIN하여 학번, 이름, 학과코드, 학과명을 출력하는 SQL문을 작성하시오. 이때, 학과코드가 입력되지 않은 학생도 출력하시오.

```
SELECT 학번, 이름, 학생.학과코드, 학과명
FROM 학생 LEFT OUTER JOIN 학과
ON 학생.학과코드 = 학과.학과코드;
```

해설 INNER JOIN을 하면 학과코드가 입력되지 않은 박치민은 출력되지 않는다. 그러므로 JOIN 구문을 기준으로 왼쪽 테이블, 즉 <학생>의 자료는 모두 출력되는 LEFT JOIN을 사용한 것이다. 다음과 같이 JOIN 구문을 기준으로 테이블의 위치를 교환하여 RIGHT JOIN을 사용해도 결과는 같다.

<결과>

학번	이름	학과코드	학과명
15	고길동	com	컴퓨터
16	이순신	han	국어
17	김선달	com	컴퓨터
19	아무개	han	국어
37	박치민		

2020년 2회 기능사 실기

1. 다음은 <topic> 테이블과 <auth> 테이블을 결합하여 검색하는 SQL문이다. 괄호에 알맞은 명령을 쓰시오.

<topic>

id	title1	title2	auth_id
1	sql	sql is	1
2	oracle	oracle is	1
3	sql-server	sql-server is	3
4	progress	progress is	2
5	mongoDB	mongoDB is	1

<auth>

id	name	job
1	kim	develope
2	shin	database administrator
3	park	scientist, develope

<SQL문>

```
SELECT * FROM topic LEFT ( ) auth ON topic.auth_id = auth.id;
```

<결과>

id	title1	title2	auth_id	id	name	job
1	sql	sql is	1	1	kim	develope
2	oracle	oracle is	1	1	kim	develope
3	sql-server	sql-server is	3	3	park	scientist, develope
4	progress	progress is	2	2	shin	database administrator
5	mongoDB	mongoDB is	1	1	kim	develope

답 :

해설 1.

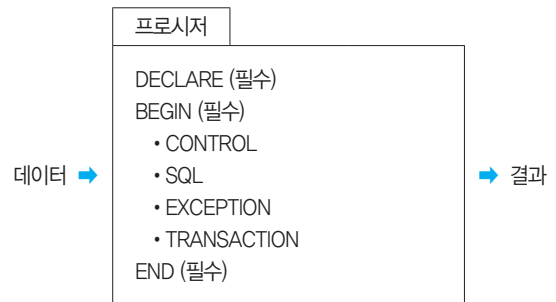
- SELECT *: 모든 속성을 표시한다.
- FROM topic LEFT OUTER JOIN auth : <topic> 테이블의 튜플들은 <auth> 테이블의 연관성과 관계없이 모두 표시하는 LEFT OUTER JOIN 이다.
- ON topic.auth_id = auth.id; : <topic> 테이블의 'auth_id'와 <auth> 테이블의 'id'로 JOIN한다.

정답 1. OUTER JOIN

[핵심115] 프로시저(Procedure)

- 절차형 SQL을 활용하여 특정 기능을 수행하는 일종의 트랜잭션 언어로, 호출을 통해 실행되어 미리 저장해 놓은 SQL 작업을 수행한다.
- 프로시저는 데이터베이스에 저장되어 수행되기 때문에 스토어드(Stored) 프로시저라고도 불린다.
- 프로시저는 시스템의 일일 마감 작업, 일괄(Batch) 작업 등에 주로 사용된다.

• 프로시저 구성도



프로시저 생성

```
CREATE [OR REPLACE] PROCEDURE 프로시저명(파라미터)
[지역변수 선언]
BEGIN
    프로시저 BODY;
END;
```

프로시저 실행

```
EXECUTE 프로시저명;
EXEC 프로시저명;
CALL 프로시저명;
```

프로시저 제거

```
DROP PROCEDURE 프로시저명;
```

1. DBMS에서 시스템의 주간 마감, 일일 마감 작업 등 주로 일괄 작업에 사용되며, 데이터베이스에 저장되어 수행된다. DECLARE, CONTROL, SQL, EXCEPTION 등의 구성 요소로 이루어져 있고, EXECUTE 또는 CALL 명령어로 실행되는 절차형 SQL은 무엇인지 쓰시오.

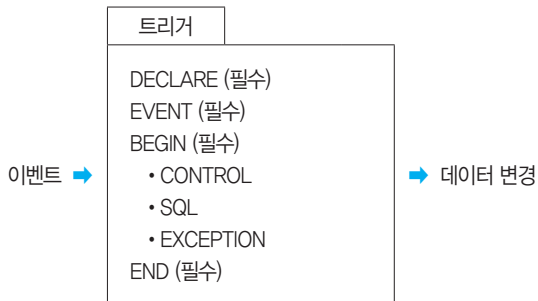
답 :

정답 1. 프로시저(Procedure) 또는 스토어드(Stored) 프로시저

[핵심116] SQL – 트리거(Trigger)

- 데이터베이스 시스템에서 데이터의 삽입(Insert), 갱신(Update), 삭제(Delete) 등의 이벤트(Event)가 발생할 때마다 관련 작업이 자동으로 수행되는 절차형 SQL이다.
- 트리거는 데이터베이스에 저장되며, 데이터 변경 및 무결성 유지, 로그 메시지 출력 등의 목적으로 사용된다.

트리거 구성도



트리거의 생성

```

CREATE [OR REPLACE] TRIGGER 트리거명 [동작시기 옵션]
    [동작 옵션] ON 테이블명
    REFERENCING [NEW | OLD] AS 테이블명
    FOR EACH ROW
    [WHEN 조건식]
    BEGIN
        트리거 BODY;
    END;
    
```

트리거의 제거

DROP TRIGGER 트리거명;

1. 데이터베이스 시스템에서 삽입, 갱신, 삭제 등의 이벤트가 발생할 때마다 관련 작업이 자동으로 수행되는 절차형 SQL을 쓰시오.

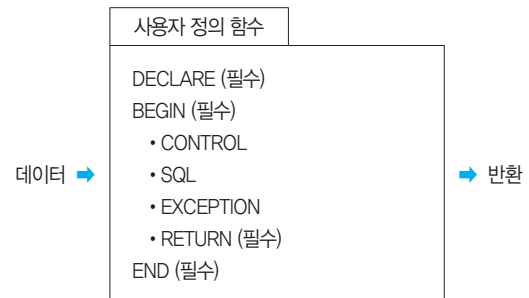
답 :

정답 1. 트리거(Trigger)

[핵심117] SQL – 사용자 정의 함수

- 프로시저와 유사하게 SQL을 사용하여 일련의 작업을 연속적으로 처리하며, 종료 시 처리 결과를 단일 값으로 반환하는 절차형 SQL이다.
- 사용자 정의 함수는 데이터베이스에 저장되어 SELECT, INSERT, DELETE, UPDATE 등 DML문의 호출에 의해 실행된다.
- 사용자 정의 함수는 예약어 RETURN을 통해 값을 반환하기 때문에 출력 파라미터가 없다.

사용자 정의 함수의 구성도



사용자 정의 함수 생성

```

CREATE [OR REPLACE] FUNCTION 사용자 정의 함수명(파라미터)
    [지역변수 선언]
    BEGIN
        사용자 정의 함수 BODY;
        RETURN 반환값;
    END;
    
```

사용자 정의 함수 실행

SELECT 사용자 정의 함수명 FROM 테이블명;
INSERT INTO 테이블명(속성명) VALUES (사용자 정의 함수명);
DELETE FROM 테이블명 WHERE 속성명 = 사용자 정의 함수명;
UPDATE 테이블명 SET 속성명 = 사용자 정의 함수명;

사용자 정의 함수 제거

DROP FUNCTION 사용자 정의 함수명;

1. 다음 설명에 가장 부합하는 절차형 SQL의 종류를 쓰시오.

- 일련의 작업을 연속적으로 처리하며, 종료 시 처리 결과를 단일값으로 반환한다.
- 데이터베이스에 저장되어 DML문의 호출에 의해 실행된다.
- RETURN을 통해 값을 반환하기 때문에 출력 파라미터가 존재하지 않는다.

답 :

정답 1. 사용자 정의 함수

[핵심 118] SQL – 커서(Cursor)

커서(Cursor)

- 쿼리문의 처리 결과가 저장되어 있는 메모리 공간을 가리키는 포인터(Pointer)이다.
- 커서는 내부에서 자동으로 생성되어 사용되는 묵시적 커서와, 사용자가 직접 정의해서 사용하는 명시적 커서가 있다.
- 커서의 수행은 열기(Open), 패치(Fetch), 닫기(Close)의 세 단계로 진행된다.

묵시적 커서(Implicit Cursor)

- DBMS 자체적으로 열리고(Open) 패치(Fetch)되어 사용이 끝나면 닫히지만(Close) 커서의 속성을 조회하여 사용된 쿼리 정보를 열람하는 것이 가능하다.

• 커서의 속성

SQL%FOUND	쿼리 수행의 결과로 패치(Fetch)된 튜플 수가 1개 이상이면 TRUE
SQL%NOTFOUND	쿼리 수행의 결과로 패치(Fetch)된 튜플 수가 0개이면 TRUE
SQL%ROWCOUNT	쿼리 수행의 결과로 패치(Fetch)된 튜플 수를 반환
SQL%ISOPEN	<ul style="list-style-type: none"> • 커서가 열린(Open) 상태이면 TRUE • 묵시적 커서는 자동으로 생성된 후 자동으로 닫히기 때문에 항상 FALSE

명시적 커서(Explicit Cursor)

- 사용자가 직접 정의해서 사용하는 커서로, 주로 절차형 SQL에서 SELECT문의 결과로 반환되는 여러 튜플들을 제어하기 위해 사용된다.
- 커서는 기본적으로 '열기(Open) – 패치(Fetch) – 닫기(Close)' 순으로 이루어지며, 명시적 커서로 사용하기 위해서는 열기 전에 선언(Declare)을 해야 한다.

1. 다음 괄호에 공통적으로 들어갈 가장 적합한 용어를 쓰시오.

- ()는 쿼리문의 처리 결과가 저장되어 있는 메모리 공간을 가리키는 포인터(Pointer)이다.
- ()는 묵시적 ()와 명시적 ()가 있다.
- ()의 수행은 열기(Open), 패치(Fetch), 닫기(Close)의 세 단계로 진행된다.

답 :

2. 절차형 SQL에서 튜플들에 순차적으로 접근하는데 사용되는 커서(Curosr) 중 명시적 커서와 묵시적 커서에 대해 간략히 서술하시오.

답

① 명시적 커서 :

② 묵시적 커서 :

정답 1. 커서(Cursor)

2. ① 사용자가 직접 정의해서 사용하는 커서이다. ② 내부에서 자동으로 생성하여 사용하는 커서이다.