

[정답 및 해설] [SQL 응용>SQL 기초]

SQL 기초 1. SQL

* SQL

- 1) 관계대수와 관계해석을 기초로 한 고급 데이터 언어
- 2) 이해하기 쉬운 형태
- 3) 대화식 질의어로 사용 가능
- 4) 데이터 정의, 조작, 제어 기능 제공
- 5) COBOL, C, PASCAL 등의 언어에 삽입 → 내장 SQL
- 6) DBMS에서 사용되는 비절차적 대화형 Language

SQL 기초 2.

```
CREATE TABLE 학생 (  
    학번 CHAR(17),  
    이름 CHAR(17),  
    학과코드 CHAR(20),  
    PRIMARY KEY(학번),  
    FOREIGN KEY(학과코드) REFERENCES  
    학과(학과코드)  
);
```

SQL 기초 3.

```
CREATE TABLE 회원 (  
    회원번호 CHAR(15),  
    이름 CHAR(5),  
    주민번호 CHAR(13) NOT NULL,  
    PRIMARY KEY(회원번호),  
    UNIQUE(주민번호)  
);
```

SQL 기초 4.

```
CREATE TABLE 회원 (  
    회원번호 CHAR(15),  
    이름 CHAR(5),  
    생년월일 DATE,  
    PRIMARY KEY(회원번호),  
    CHECK (생년월일 >= '1970-01-01')  
);
```

SQL 기초 5.

```
ALTER TABLE 학생  
    ADD 지역 CHAR(10);
```

SQL 기초 6.

```
ALTER TABLE 학생  
    DROP 성별;
```

[정답 및 해설] [SQL 응용>SQL 기초]

SQL 기초 7. CASCADE

CASCADE: 참조하는 테이블을 연쇄적으로 제거

SQL 기초 8.

- ① 정의어(DDL): CREATE, ALTER, DROP
- ② 조작어(DML): SELECT, INSERT, DELETE, UPDATE
- ③ 제어어(DCL): GRANT, REVOKE, COMMIT, ROLLBACK

SQL 기초 9.

SELECT 주소, 수강료 FROM 수강생;

SQL 기초 10.

SELECT * FROM 수강생;

SQL 기초 11.

INSERT INTO 수강생 VALUES ('최석현', 27, '엑셀', '중구', 100);

SQL 기초 12.

INSERT INTO 수강생(이름, 나이, 주소) VALUES ('이채희', 23, '서구');

SQL 기초 13.

INSERT INTO 정보수강생(이름, 과목, 수강료)
SELECT 이름, 과목, 수강료 FROM 수강생 WHERE 과목='정보';

SQL 기초 14.

DELETE FROM 수강생 WHERE 수강료=100;

SQL 기초 15.

DELETE FROM 수강생;

SQL 기초 16.

UPDATE 수강생 SET 과목='엑셀' WHERE 이름='김지현';

SQL 기초 17.

UPDATE 수강생 SET 수강료=수강료+20 WHERE 과목='컴활';

SQL 기초 18. DISTINCT

DISTINCT: 검색 결과에서 중복 배제

SQL 기초 19. IN

'=' 를 넣을 경우 SQL문이 실행되지 않는다.
'=' 은 한 가지 속성일 경우만 가능

[정답 및 해설] [SQL 응용>SQL 기초]

SQL 기초 20. OR

```
SELECT * FROM 수강생 WHERE 과목='정보' OR 과목='워드'
```

SQL 기초 21. <>

```
SELECT * FROM 수강생 WHERE 과목 <> '컴활';
```

SQL 기초 22. LIKE '김%'

```
SELECT FROM 수강생 WHERE 이름 LIKE '김%';
```

SQL 기초 23. BETWEEN 60 AND 100

```
SELECT * FROM 수강생 WHERE 수강료 BETWEEN 60 AND 100;
```

SQL 기초 24. IS NULL

```
SELECT * FROM 수강생 WHERE 나이 IS NULL;
```

SQL 기초 25. ORDER BY 이름 DESC;

```
SELECT * FROM 수강생 WHERE 과목='정보'  
ORDER BY 이름 DESC;
```

SQL 기초 26. GROUP BY 과목 HAVING COUNT(*)>=2;

```
SELECT 과목 FROM 수강생 WHERE 수강료>=100  
GROUP BY 과목 HAVING COUNT(*)>=2;
```

SQL 기초 27. (ㄱ)-(ㄴ)-(ㄷ)-(ㄹ)-(ㄺ)-(ㄻ)

* SQL 수행 순서

: FROM→WHERE→GROUP BY→HAVING→SELECT
→ ORDER BY

SQL 기초 28. WITH GRANT OPTION

GRANT: 권한 부여를 위한 명령어

SQL 기초 29. ① REVOKE ② CASCADE

REVOKE: 권한 취소를 위한 명령어

SQL 기초 30. ① Commit ② Roll Back

모든 작업이 성공한 경우 Commit을 해주고, 중간에 조금이라도 실수가 있었다면 Roll Back을 하게 된다.

SQL 기초 31. UNION

UNION: 중복 없이 병합하는 명령어

SQL 기초 32.

```
CREATE VIEW 사무수강 _view(이름, 나이) AS  
SELECT 이름,나이  
FROM 학생  
WHERE 학과코드='사무';
```

[정답 및 해설] [SQL 응용>SQL 기초]

SQL 기초 33.

```
CREATE UNIQUE INDEX 학생_idx  
ON 학생(이름 ASC);
```

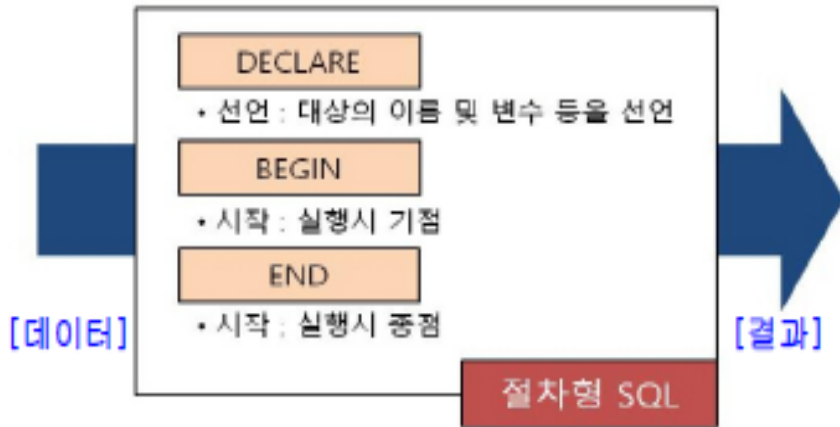
SQL 기초 34. ① DISTINCT ② IN

```
- SELECT DISTINCT name FROM Shop WHERE id  
IN (SELECT Shopid FROM Staff WHERE id = '20');  
- 'IN' 대신 '=' 도 가능
```

[정답 및 해설] [SQL응용>절차형 SQL 작성]

절차형 SQL 작성 1. ① BEGIN ② DECLARE ③ END

* 절차형 SQL 구성도



절차형 SQL 작성 2. ① IF SCORE >= 80 THEN ② ELSE ③ END IF;

* IF문: 조건이 '참'일 경우 문장1 실행, '거짓'일 경우 문장2 실행

```
IF 조건 THEN
    문장1;
ELSE
    문장2;
END IF;
```

절차형 SQL 작성 3. ① FOR ② 1..5 ③ SUM + i

- FOR i IN 1..5 : i 값은 1~5까지 1씩 증가하면서 LOOP문 수행
- SUM := SUM + i : 1~5까지 합계를 SUM 변수에 저장

```
FOR 변수 IN 초기값..종료값
LOOP
    문장;
END LOOP;
```

절차형 SQL 작성 4. ① 3 ② WHILE i <= 9 ③ i + 1

- number INT := 3 : number 변수의 초기값을 3으로 선언
- WHILE i <= 9 : i <= 9 인 경우에만 LOOP문 수행
- i := i + 1 : i를 1씩 증가

```
WHILE 조건
LOOP
    문장;
END LOOP;
```

절차형 SQL 작성 5. CONTINUE WHEN i=5;

* CONTINUE문: 조건이 '참'일 경우 CONTINUE 이후의 문장은 실행하지 않고 반복문 처음으로 이동

```
CONTINUE WHEN 조건;
```

[정답 및 해설] [SQL응용>절차형 SQL 작성]

절차형 SQL 작성 6. 1
2
1
2
3

* GOTO문: 원하는 레이블로 이동한다.

```
GOTO 레이블;  
:  
:  
<<레이블>>  
문장;
```

- IF i = 2 THEN

GOTO SECOND;

: i = 2일 경우, SECOND 레이블로 이동한다.

절차형 SQL 작성 7. ① 묵시적 커서(Implicit Cursor) ② 명시적 커서(Explicit Cursor)

* 커서(Cursor) 종류

- 묵시적 커서: 각 단계가 자동으로 수행
- 명시적 커서: 사용자가 직접 구현

절차형 SQL 작성 8. (⌋)→(≡)→(≡)→(⌋)

- 커서(Cursor) 처리 순서

: 커서 선언(CURSOR 커서명)→커서 열기(OPEN 커서명)→커서에서 데이터 가져오기(FETCH)→커서 닫기(CLOSE 커서명)

절차형 SQL 작성 9. 3

- UPDATE 급여 SET 기본급 = 기본급 * 1.05

WHERE 부서='인사'; : [급여]테이블에서 '부서' 속성이 "인사"인 튜플의 '기본급'을 수정한다.

- UPDATE 문의 결과로 반영되는 튜플은 1, 5, 6행 세 개이므로 3이 출력된다.

절차형 SQL 작성 10. 프로시저(Procedure)

프로시저(Procedure)에 대한 설명이다.

절차형 SQL 작성 11. EXECUTE 또는 EXEC 또는 CALL

* 프로시저 호출 명령어: EXECUTE, EXEC, CALL

[정답 및 해설] [SQL응용>절차형 SQL 작성]

절차형 SQL 작성 12. ① DECLARE ② BEGIN ③ CONTROL ④ SQL ⑤ EXCEPTION ⑥ TRANSACTION ⑦ END

★ 프로시저 구성요소

- DECLARE: 프로시저의 명칭, 변수와 인수 그리고 그에 대한 데이터 타입을 정의하는 선언부
- BEGIN / END: 프로시저의 시작과 종료를 표현
- CONTROL: 조건문 또는 반복문이 삽입되어 기본적으로는 순차적으로 처리
- SQL: DML, DCL이 삽입되어 데이터 관리를 위한 조회, 추가, 수정, 삭제 작업을 수행
- EXCEPTION: BEGIN ~ END절에서 실행되는 SQL문이 실행될 때 예외 발생 시 예외 처리 방법을 정의
- TRANSACTION: 프로시저에서 수행된 DML 수행 내역의 DBMS의 적용 또는 취소 여부를 결정하는 처리부

절차형 SQL 작성 13. ① CREATE ② BEGIN ③ UPDATE 회원 SET 수강기간 = '20200607' WHERE 회원번호 = pm_회원번호; ④ END

- CREATE : DBMS 내에 프로시저를 생성(프로시저명: change_date)
- BEGIN : 프로시저 시작을 알림
- UPDATE 회원 SET 수강기간 = '20200607' WHERE 회원번호 = pm_회원번호
: 회원 테이블의 회원번호가 프로시저의 매개변수로 받은 pm_회원번호와 같으면 해당 회원의 수강기간을 '20200607'로 변경
- END : 프로시저 끝을 알림

절차형 SQL 작성 14. (⌋), (⌌), (⌍)

- (⌋), (⌌) : 프로시저와의 차이점이다.
 - 프로시저는 반환값이 없거나 1개 이상 가능하고, 파라미터에는 입·출력 모두 가능하다.
- (⌍) : 프로시저와 마찬가지로 호출에 의해 실행되며, 사용자 정의함수는 주로 SELECT문에 포함되어 실행된다.

절차형 SQL 작성 15. RETURN

- RETURN 반환값 : 사용자 정의 함수 종료 시 반환한 값을 호출한 쿼리에 반환한다.

[정답 및 해설] [SQL응용>절차형 SQL 작성]

절차형 SQL 작성 16. 트리거(Trigger)

트리거(Trigger)에 대한 설명이다.

절차형 SQL 작성 17. AFTER, BEFORE

- * 트리거 동작시기 옵션
 - AFTER: 테이블 변경 후 실행
 - BEFORE: 테이블 변경되기 전 실행

절차형 SQL 작성 18. INSERT, DELETE, UPDATE

- * 트리거 동작 옵션
 - INSERT: 테이블에 튜플이 삽입될 때 실행
 - DELETE: 테이블의 튜플이 삭제될 때 실행
 - UPDATE: 테이블의 튜플이 수정될 때 실행

절차형 SQL 작성 19. ① INSERT ② FOR EACH ROW

- INSERT ON 사원: '사원'테이블이 입력될 때
- FOR EACH ROW: 입력되는 모든 행에 대해 트리거 실행

[정답 및 해설] [SQL응용>응용 SQL 작성]

응용 SQL 작성 1. ① MAX ② SUM ③ AVG

* 집계 함수 분류

- COUNT
- SUM
- AVG
- MAX
- MIN
- STDDEV
- VARIAN

응용 SQL 작성 2. CUBE

- * CUBE : 결합 가능한 모든 값에 대해 다차원 집계를 생성하는 그룹 함수

응용 SQL 작성 3. ① 집계 함수 ② 그룹 함수 ③ 윈도우 함수

* 데이터 분석 함수 종류

- 집계 함수
- 그룹 함수
- 윈도우 함수

응용 SQL 작성 4. GROUP BY GROUPING SETS

- * GROUPING SETS: 다양한 소계 집합을 만들 수 있다. 집계 대상 컬럼들에 대한 개별 집계를 구할 수 있다.

응용 SQL 작성 5. ROLLUP

- * ROLLUP: 지정된 컬럼은 소계 등 중간 집계값을 산출하기 위해 사용된다.

응용 SQL 작성 6. ROW_NUMBER()

* ROW_NUMBER

: 동일 순위의 값이 존재해도 이와 무관하게 연속 번호를 부여한다.

응용 SQL 작성 7. RANK()

* RANK

: 동일 순위의 레코드 존재 시 후순위는 넘어간다. (ex. 1위, 2위, 2위, 4위)

응용 SQL 작성 8. DENSE_RANK()

* DENSE_RANK

: 동일 순위의 레코드 존재 시에도 후순위를 넘어가지 않는다. (ex. 1위, 2위, 2위, 3위)

응용 SQL 작성 9. MyBatis

MyBatis에 대한 설명이다.

응용 SQL 작성 10. 사용자 그룹

다수의 사용자가 접속하는 사용자를 사용자 그룹이라고 한다.