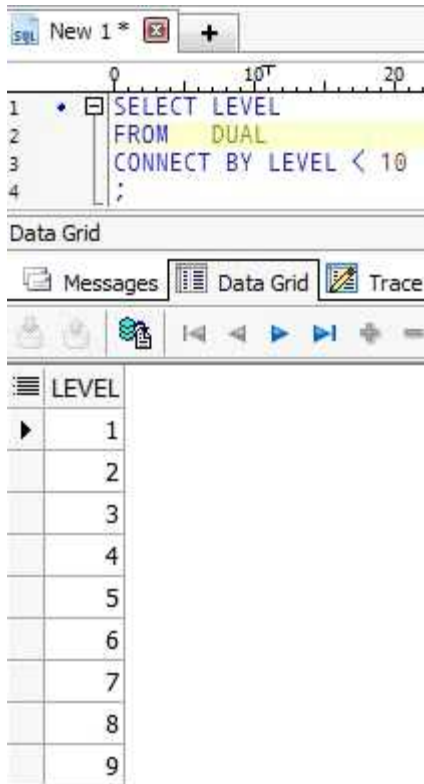


DO IT

17. Oracle의 CONNECT BY LEVEL < = : 변수 (< , = 사용 가능)

1부터 :변수 까지 1씩 루프를 돌린다.



The screenshot shows the Oracle SQL Developer interface. The top pane contains the following SQL query:

```
SELECT LEVEL  
FROM DUAL  
CONNECT BY LEVEL < 10  
;
```

The bottom pane shows the results in a 'Data Grid' tab. The results are as follows:

LEVEL
1
2
3
4
5
6
7
8
9

DO IT

DO IT

DO IT

2017년 기출 - 17번

DUAL TABLE connect by- level <=2 , 답 : 2

18. substring 문제

Oracle의 경우

DO IT

3번째 자리부터 쪽

```
SELECT substr('ABC한미희MH',3) FROM DUAL;  
C한미희MH
```

4번째 자리부터 2글자, 두 번째 인자 : 시작점 , 세 번째 인자 : 글자 갯수

```
SELECT substr('ABC한미희MH',4,2) FROM DUAL;  
한미
```

우측부터 추출하고 싶으면 - 기호를 사용

```
SELECT substr('ABC한미희MH',-2) FROM DUAL;  
MH
```

DO IT

DO IT

두 번째 인자 : 시작점 , 세 번째 인자 : 글자 갯수

```
SELECT substr('ABC한미희MH',-5,2) FROM DUAL;  
한미
```

2017년 기출 - 18번

substring문제 -> substr(칼럼,8,-2)

DO IT

19. lead/lag

게시판을 만들 때 목록에서 글을 선택한 후 상세 화면에서 내용을 본 후 아래로 내리면 다음글과 이전글을 링크해서 보여주는 경우가 있는데 이럴 때 다음글과 이전글을 SQL를 이용해 읽어 오는 방법이 있다.

lead(컬럼명,행수,null일 때 보여질 기본값)

: 특정 컬럼의 지정한 숫자의 다음 행의 데이터 반환

lag(컬럼명,행수,null일 때 보여질 기본값)

: 특정 컬럼의 지정한 숫자의 이전 행의 데이터 반환

DO IT

2017년 기출 - 19번

lead/lag - > lead (칼럼,2) 마지막에 0있으면 null이 아니라 0되서 없는 걸로 했습니다.

댓글에 0으로 채워져 있다고 제보가 왔으니 lead(칼럼,2,0)이 맞겠네요

20. rownum과 order by와의 관계

테이블에서 order by 로 소팅하고 원하는 상위 개수만 가져오고자 할 때 rownum을 쓰면된다.

하지만, 주의할점은 아래와 같이 sql문을 작성하면 rownum이 먼저 실행되고 order by가 나중에 실행되면서 원하는 결과가 나오지 않는다.

```
select * from 테이블명 where rownum <4 order by num desc
```

원하는 결과가 나오게 하려면 rownum과 order by를 나눠주면 된다.

```
select * from (select * from 테이블명 order by numsc) where rownum <4
```

2017년 기출 - 20번

order by 와 rownum 동작 순서를 묻는 문제

DO IT

DO IT

DO IT

21. rank와 dense_rank

```
select id, salary
```

```
-- 분석 함수의 order by 절을 사용하여 내림차순 정렬한 뒤 rank() 함수가 순위를 반환
```

```
, RANK() OVER (ORDER BY salary DESC) ranking1  
, DENSE_RANK() OVER (ORDER BY salary DESC) ranking2  
, ROW_NUMBER() OVER (ORDER BY salary DESC) ranking3
```

```
from (
```

```
select 100 id, 39000 salary from dual union all
```

```
select 101 id, 30000 salary from dual union all
```

```
select 102 id, 9000 salary from dual union all
```

```
select 103 id, 17000 salary from dual union all -- 중복
```

```
select 104 id, 60000 salary from dual union all
```

```
select 105 id, 9500 salary from dual union all
```

```
select 106 id, 17000 salary from dual union all -- 중복
```

```
select 107 id, 37500 salary from dual union all
```

```
select 108 id, 25000 salary from dual union all
```

```
select 109 id, 5000 salary from dual union all
```

```
select 110 id, 17000 salary from dual -- 중복
```

```
) ;
```

DO IT

```
-----  
ID  SALARY  RANKING1  RANKING2  RANKING3
```

```
-----  
104 60000   1         1         1
```

```
100 39000   2         2         2
```

```
107 37500   3         3         3
```

```
101 30000   4         4         4
```

```
108 25000   5         5         5
```

```
103 17000   6         6         6
```

```
110 17000   6         6         7
```

```
106 17000   6         6         8
```

```
105 9500    9         7         9
```

```
102 9000   10        8        10
```

```
109 5000   11        9        11  
-----
```

DO IT

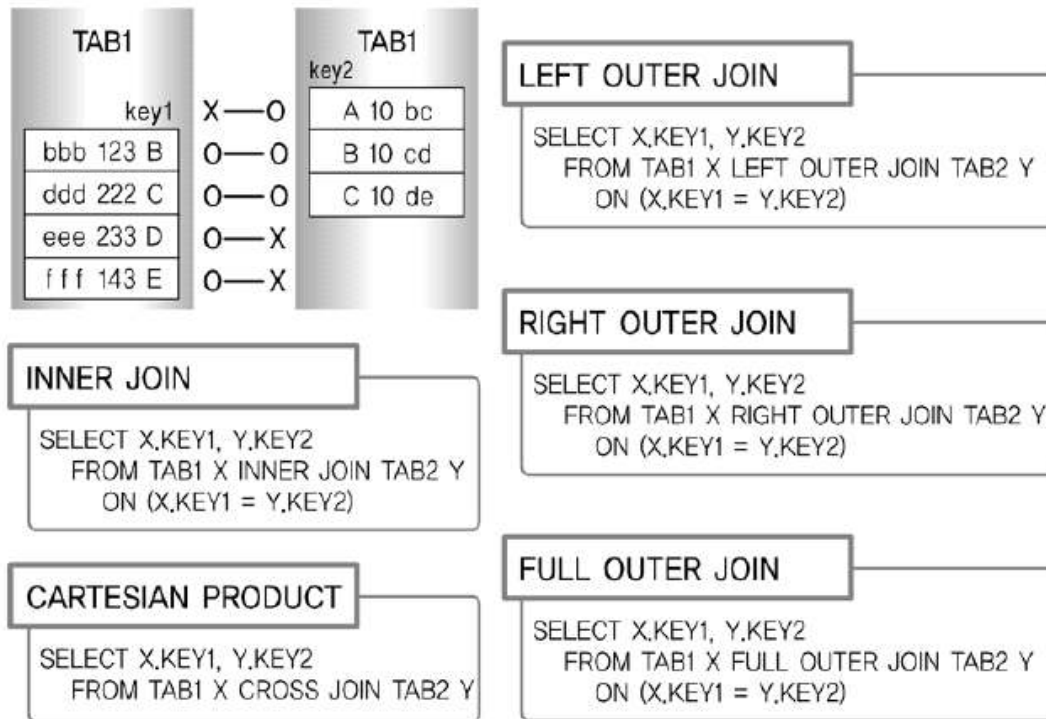
2017년 기출 - 21번

rank,dense_rank,rownum 묻는 것 -> 1,2,2,3 이런식이었기 때문에 dense_rank로 고름

DO IT

DO IT

22. 표준 join



[그림 II-2-4] INNER vs OUTER vs CROSS JOIN 문장 비교

1. INNER JOIN의 결과

B-B, C-C (2건)

DO IT

2. LEFT OUTER JOIN의 결과

B-B, C-C, D-NULL, E-NULL (4건)

3. RIGHT OUTER JOIN의 결과

NULL-A, B-B, C-C (3건)

4. FULL OUTER JOIN의 결과

NULL-A, B-B, C-C, D-NULL, E-NULL (5건)

DO IT

5. CROSS JOIN의 결과

4 * 3 = 12개의 결과

B-A, B-B, B-C, C-A, C-B, ...

2017년 기출 - 22번

full+inner+right+cross 개수 -> 5+1+2+8 = 16 , 문제가 left조인이면 18

DO IT

DO IT

23. 복원실패

// between and 묻는 문제 - > 테이블이 vip/vip row가 2개

24. 평균 구하는 문제.

COUNT , SUM , AVG는 NULL을 어떻게 처리할까?

-> NULL을 제외한 컬럼끼리만 계산을 한다. 즉 NULL != 0 아니다!

2017년 기출 - 24번

평균구하는 문제 avg(컬럼)으로 구하는 것과 sum(컬럼)/count(*) -> 105, 70

DO IT

25. TRIM , REPLACE

REPLACE : A문자열의 특정 문자를 다른 문자로 변환한다.

REPLACE(컬럼명, '변경할 문자', '변경 될 문자') FROM DUAL;

TRIM : 양쪽 특정한 문자를 제거 한다. 제거할 문자를 입력하지 않으면 기본적으로 공백이 제거 된다.

TRIM ([LEADING,TRAILING,BOTH] 제거할 문자 [FROM] 컬럼)

LEADING : 왼쪽에서 제거

TRAILING : 오른쪽에서 제거

BOTH : 양쪽에서 제거

LTRIM : 왼쪽부터 문자를 제거 한다. 제거할 문자를 입력하지 않으면 기본적으로 왼쪽 공백이 제거 된다.

RTRIM : 오른쪽부터 문자를 제거 한다. 제거할 문자를 입력하지 않으면 기본적으로 오른쪽 공백이 제거 된다.

LTRIM (컬럼 , 제거할 문자) | RTRIM (컬럼 , 제거할 문자)

제거할 문자가 없을시 기본적으로 공백

2017년 기출 - 25번 문제

TRIM ,REPLACE문제 -> REPLACE

DO IT

DO IT

DO IT

26. CAST

[표 II-1-32] 단일행 변환형 함수의 종류

변환형 함수 - Oracle	함수 설명
TO_NUMBER(문자열)	alphanumeric 문자열을 숫자로 변환한다.
TO_CHAR(숫자 날짜 [, FORMAT])	숫자나 날짜를 주어진 FORMAT 형태로 문자열 타입으로 변환한다.
TO_DATE(문자열 [, FORMAT])	문자열을 주어진 FORMAT 형태로 날짜 타입으로 변환한다.
변환형 함수 - SQL Server	함수 설명
CAST (expression AS data_type [(length)])	expression을 목표 데이터 유형으로 변환한다.
CONVERT (data_type [(length)], expression [, style])	expression을 목표 데이터 유형으로 변환한다.

2017 기출문제 - 26번

연산자 설명 틀린 것 -> CAST (형을 반환한다고 나와서 고름)

DO IT

27. group by , having

having절은 반드시 group by 절이 존재해야 한다.

2017년 기출문제 - 27번

group by having 문제 틀린 것-> group by 없이 HAVING만 쓴거 고름

DO IT

DO IT

DO IT

28. ??

[표 II-1-15] 연산자의 종류

구분	연산자	연산자의 의미
비교 연산자	=	같다.
	>	보다 크다.
	>=	보다 크거나 같다.
	<	보다 작다.
	<=	보다 작거나 같다.
SQL 연산자	BETWEEN a AND b	a와 b의 값 사이에 있으면 된다.(a와 b 값이 포함됨)
	IN (list)	리스트에 있는 값 중에서 어느 하나라도 일치하면 된다.
	LIKE '비교문자열'	비교문자열과 형태가 일치하면 된다.(%, _ 사용)
	IS NULL	NULL 값인 경우
논리 연산자	AND	앞에 있는 조건과 뒤에 오는 조건이 참(TRUE)이 되면 결과도 참(TRUE)이 된다. 즉, 앞의 조건과 뒤의 조건을 동시에 만족해야 한다.
	OR	앞의 조건이 참(TRUE)이거나 뒤의 조건이 참(TRUE)이 되어야 결과도 참(TRUE)이 된다. 즉, 앞뒤의 조건 중 하나만 참(TRUE)이면 된다.
	NOT	뒤에 오는 조건에 반대되는 결과를 되돌려 준다.
부정 비교 연산자	!=	같지 않다.
	^=	같지 않다.
	<>	같지 않다.(ISO 표준, 모든 운영체제에서 사용 가능)
	NOT 칼럼명 =	~와 같지 않다.
	NOT 칼럼명 >	~보다 크지 않다.
부정 SQL 연산자	NOT BETWEEN a AND b	a와 b의 값 사이에 있지 않다. (a, b 값을 포함하지 않는다)
	NOT IN (list)	list 값과 일치하지 않는다.
	IS NOT NULL	NULL 값을 갖지 않는다.

2017년 기출문제 - 28번

a,b,c,null행이 있고 결과 다른 것-> 4번고름

DO IT

1. in(a,b)
2. a or b
3. is not null or <>c
4. not in (c,null) -> NOT IN

DO IT

DO IT

29. 복원 실패

결과 다른것 -> 괄호가 없어서 a=123123 and b=1 or b=2 or b=3 가 보기인 것 고름

30. 복원 실패

논문을 안쓴 회원 구하는것 -> 1,2 번이라고 하시네요 1,2번 고른 문제가있긴한데 이건진잘..

31. 복원 실패

DO IT

ADF어쩌구했떤 문제가 있었떤 것 같은데....

32. 연산자 우선 순위

[표 II-1-16] 연산자의 우선순위

연산 우선순위	설 명
1	괄호 ()
2	NOT 연산자
3	비교 연산자, SQL 비교 연산자
4	AND
5	OR

2017년 기출문제 - 32번

연산 순서 가장 마지막인것 -> 1.() , 2 not , 3.and ,4.or 이런식??? or고름

DO IT

DO IT

33. Oracle 계층형 질의

DO IT

```
SELECT...  
FROM 테이블  
WHERE condition AND condition...  
START WITH condition  
CONNECT BY [NOCYCLE] condition AND condition...  
[ORDER SIBLINGS BY column, column, ...]
```

[그림 II-2-7] 계층형 질의 구문

1) START WITH

계층 구조 전개에 시작 위치(루트)를 지정하는 구문. 즉 루트 데이터를 지정한다.
서브 쿼리를 사용할 수 있다.

2) CONNECT BY

부모행과 자식행의 관계를 규정. 다음에 전개될 자식 데이터를 지정하는 구문.
보통 PRIOR 연산자를 많이 사용한다.
자식 데이터는 CONNECT BY 절의 조건을 만족해야한다.

- PRIOR

'PRIOR 자식 = 부모' 형태 : 부모데이터에서 자식데이터 방향으로 전개하는 순방향 전개
'PRIOR 부모 = 자식' 형태 : 자식데이터에서 부모데이터 방향으로 전개하는 역방향 전개

- NOCYCLE

데이터를 전개하면서 이미 나타났던 동일한 데이터가 전개 중에 다시 나타난다면 이것을 가리켜 사이클(Cycle)이 형성되었다라고 말한다. 사이클이 발생한 데이터는 런타임 오류가 발생한다.
그렇지만 NOCYCLE을 추가하면 사이클이 발생한 이후의 데이터는 전개하지 않는다.

6) ORDER SIBLINGS BY

형제 노드(동일 레벨) 사이에서 정렬을 수행한다.

DO IT

7) WHERE

모든 전개를 수행한 후에 지정된 조건을 만족하는 데이터만 추출한다.

2017년 기출문제 - 33번

계층형 문제 PRIOR 나오는 것 있었는데 선택지는 기억이...

DO IT

33번 계속...

DO IT

SQL SERVER 계층형질의

재귀적 CTE의 기본 형식은 다음과 같다.

WITH CTE_테이블이름(열이름)

AS

(
 <쿼리문1 : SELECT * FROM 테이블A >
 UNION ALL
 <쿼리문2 : SELECT * FROM 테이블A JOIN CTE_테이블이름>
)
SELECT * FROM CTE_테이블이름 ;

DO IT

위의 구문에서 <쿼리문1>을 앵커 멤버(Anchor Member:AM)이라고 부르고,
 <쿼리문2>를 재귀멤버(Recursive Member:RM)라고 지칭한다.

DO IT

- ① <쿼리문1>을 실행한다. 이것이 루틴의 최초 호출에 해당한다. 그리고 기본값은 0으로 초기화된다.
- ② <쿼리문2>를 실행한다. 기본값을 기본값+1로 증가시킨다. 그런데 SELECT의 결과가 빈것이 아니라면, 'CTE_테이블이름'을 다시 재귀적으로 호출한다
- ③ 계속 ②번을 반복한다. 단 SELECT의 결과가 아무것도 없다면 재귀적인 호출이 중단된다.
- ④ 외부의 SELECT 문을 실행해서 앞 단계에서의 누적된 결과(UNION ALL)를 가져온다.

DO IT

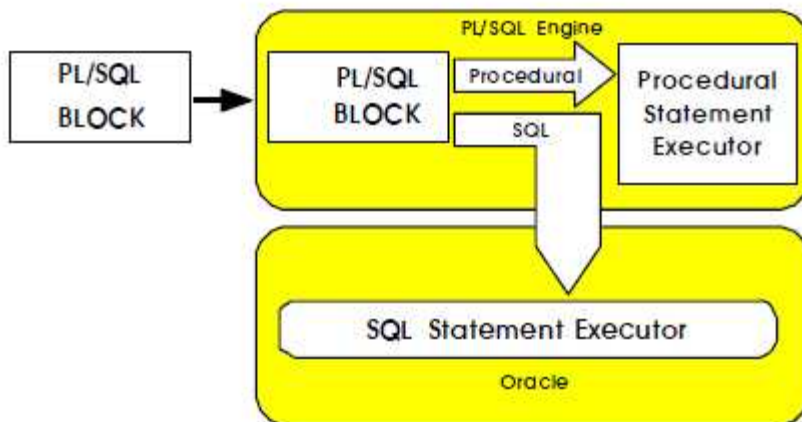
DO IT

34. PL / SQL

PL / SQL을 이용하여 다양한 저장모듈을 개발 할 수 있다. **저장모듈이란 PL / SQL 문장을 데이터베이스 서버에 저장하여 사용자와 애플리케이션 사이에 공유할 수 있도록 만든 일종의 SQL 컴포넌트 프로그램이며, 완전한 실행 프로그램이다.**

Oracle 저장 모듈에는 Procedure, User Defined Function, Trigger가 있다.

- Block 구조로 되어있어 기능별로 모듈화 가능
- 변수, 상수 등을 선언하여 SQL 문장 간 값 교환 가능
- IF, LOOP 등의 절차형 언어를 사용하여 절차적인 프로그램이 가능하도록 함
- DBMS 정의 에러나 사용자 정의 에러를 정의하여 사용 가능
- PL/SQL은 Oracle에 내장되어있어서, 이를 지원하는 어떤 서버로도 프로그램을 옮길 수 있음
- PL / SQL은 응용 프로그램의 성능을 향상시킴
- 이는 SQL 문장을 여러 블록으로 묶고 한 번에 블록 전부를 서버로 보내기 때문에 통신량을 줄일 수 있음



DO IT

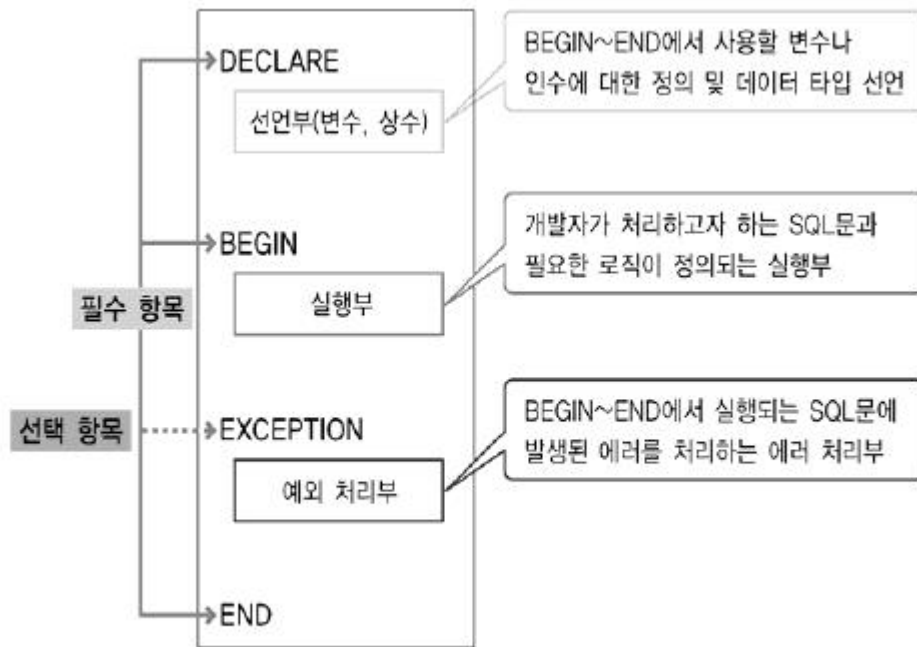
[그림 II-2-18] Oracle의 PL/SQL 엔진

- PL/SQL 엔진 : 프로그램 문장 처리
- SQL Statement Executor : SQL 문장 처리

DO IT

나. PL / SQL 구조

DO IT



[그림 II-2-19] PL/SQL 블록 구조

DO IT

다. PL / SQL 기본 문법

```
CREATE [OR REPLACE] Procedure [Procedure_name]
( argument1 [mode] data_type1, argument2 [mode] data_type2,...)
IS [AS]
...
BEGIN
...
EXCEPTION
...
END;
/
```

DO IT

다음은 생성된 프로시저를 삭제하는 명령어이다.

```
DROP Procedure [Procedure_name];
```

DO IT

DO IT

- 프로시저는 절차적인 언어를 이용하여 작성한 프로그램 모듈, 필요할 때 호출하여 실행 가능
- OR REPLACE는 디비내에 같은 이름의 프로시저가 있을 경우, 기존 것을 무시하고 대체하는 것
- Argument는 프로시저가 호출될 때 처리한 결과 값을 운영체제로 리턴시킬 매개변수를 지정할 때 사용
- [mode]에 가능한 값은 3가지
 - IN : 운영체제에서 프로시저로 전달
 - OUT : 프로시저에 운영체제로 전달
 - INOUT : 둘다 가능
- / : 데이터베이스에게 프로시저를 컴파일 하라는 명령

DO IT

2017년 기출문제 - 34번

PL/SQL 틀린 것 -> 테이블 생성할 수 없다.

DO IT