

# SQL 목차

2018년 10월 13일 토요일    오후 5:58

- [SQL 활용 목차](#)
- [SQL 목차](#)
- [Oracle 설치 및 이용](#)
- [1. 컬럼과 로우](#)
- [2. 데이터 제한 및 정렬](#)
- [3. 함수\(단일행 함수, 복수행 함수\)](#)
- [4. 중첩 그룹함수](#)
- [5. Data 분석 함수](#)
- [6. 분석함수 문제](#)
- [7. 조인](#)
- [8. 조인 실습](#)
- [9. 집합 연산자](#)
- [10. 집합 연산자 사용시 주의 사항](#)
- [11. 서브쿼리란?](#)
- [12. Exists](#)
- [13. DML 문 insert, update, delete, merge](#)
- [14. Truncate 문](#)
- [15. 트랜잭션](#)
- [16. Lock 이란 무엇인가?](#)
- [17. 제약](#)
- [18. 테이블 \(뷰\)](#)
- [19. 시퀀스](#)
- [20. 인덱스\(index\)](#)
- [21. 시너님\(synonym\)](#)
- [22. 스칼라 서브쿼리](#)
- [23. 상호관련 서브쿼리](#)
- [24. With 절](#)
- [25. 대형 데이터 집합 조작](#)
- [26. Insert, update문에서 default 사용방법](#)
- [27. 다중 insert문 4가지](#)
- [28. 계층형 질의문](#)
- [29. 데이터 덱서너리를 사용한 객체관리](#)
- [30. 임시 테이블](#)
- [31. External 테이블](#)
- [32. 날짜함수](#)
- [33. 정규식 함수](#)

# Oracle 설치 및 이용

2018년 4월 17일 화요일     오후 5:36

oracle설치

- 1 - [otn.oracle.com](http://otn.oracle.com)
- 2 - setup설치
- 3 - password : oracle

접속

- 1 - 도스창열기(cmd)
- 2 - sqlplus sys/oracle as sysdba 입력  
sqlplus = 오라클 프로그램 sqlplus를 사용하겠다.  
sys = 유저이름  
/oracle = 패스워드  
as sysdba = 관리자로 접속

tip) 색깔 바꾸기 = 상단에 바 오른쪽마우스 -> 속성 -> 색 변경  
명령어

show user : 현재 접속한 유저가?

복/붙 = 바 오른쪽 클릭 -> 옵션 -> 빠른 편집 -> 오른쪽마우스 클릭하면 바로 붙여넣기됨.

1.sqlplus sys/oracle as sysdba -- 관리자 계정으로 접속

2. C:\Woraclexe\Wapp\Woracle\Wproduct\W11.2.0\server\Wrdbs\Wadmin\Wscott.sql --  
생성스크립트 실행

3.alter user scott identified by tiger;                      -- scott계정의 비밀번호를 tiger로  
변경

4.exit 후 sqlplus scott/tiger로 로그인 해본다.

- 실습용 데이터 만들기

SQL 검색문 예제

```
select ename, sal  
from emp;
```

tip) select = 검색

from = 어디서로부터  
emp 테이블의 컬럼명  
desc emp - emp테이블 검색  
set pages 숫자 = 한 화면에 숫자만큼 보이기  
sql문마다 한줄씩 작성하기(보기 수월함)

EMPNO     : 사원번호  
ENAME     : 사원이름  
JOB       : 직업  
MGR       : 관리자 사원번호  
HIREDATE : 입사일  
SAL       : 월급  
COMM      : 커미션  
DEPTNO    : 부서번호

문제1) 사원 이름, 월급, 직업 출력하시오.

```
select ENAME, SAL, JOB  
from emp;
```

문제2) 사원 번호, 이름, 월급, 부서번호 출력하시오.

```
select EMPNO, ENAME, SAL, EDPTNO  
from emp;
```

**문제3) 사원 이름, 입사일, 월급, 커미션, 관리자번호를 출력하시오.**

```
select ENAME, HIREDATE, SAL, COMM, MGR  
from emp;
```

- order by 절 : 데이터를 정렬하는 절  
ex) select ENAME, SAL  
from emp  
order by sal asc; -> SAL 을 낮은값부터 정렬  
tip) asc : 낮은값 -> 높은값  
desc : 높은값 -> 낮은값

**문제4) 이름과 입사일을 출력하는데 최근에 입사한 사원부터 출력하시오.**

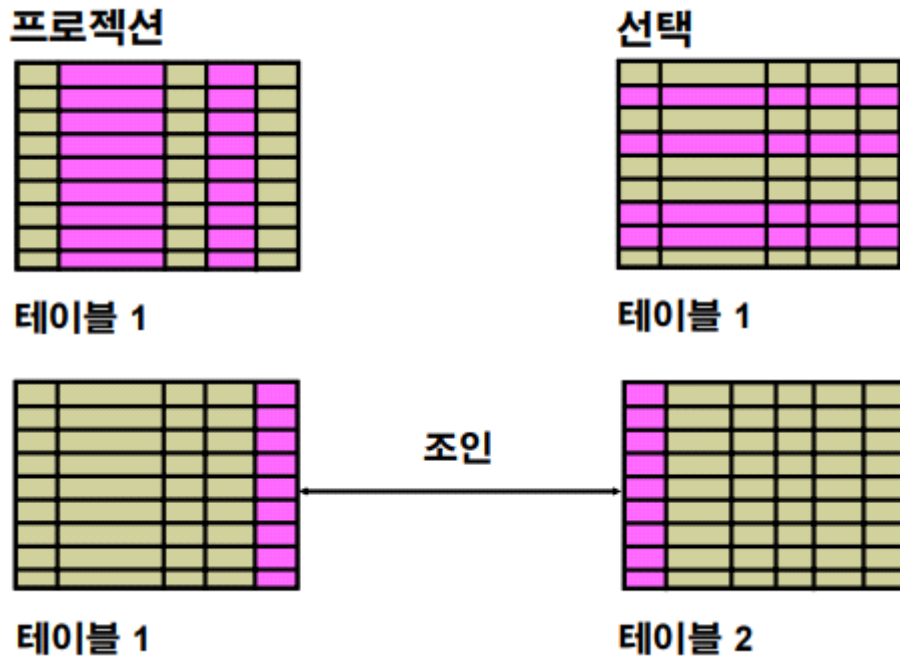
```
select ENAME, HIREDATE  
from emp  
order by HIREDATE desc;
```

# 1. 컬럼과 로우

2018년 4월 17일 화요일    오후 7:18

## 1. 컬럼과 로우

- 테이블은 데이터를 저장하는 논리적 저장소이고, **컬럼(column)**과 **로우(row)**로 구성되어 있다.



ex) dept 테이블로 확인.

문제5. 사원번호, 이름, 직업, 월급, 커미션을 출력하시오.

답) `select EMPNO, ENAME, JOB, SAL, COMM  
from emp;`

문제6. 모든 컬럼을 전부 조회하는 방법

답) `select *  
from emp;`

- 중복 데이터를 제거하는 키워드

```
select distinct job  
from emp;
```

- 예쁘게 테이블 출력하는방법

`show lines ->` 테이블 결과 출력 가로 사이즈  
`set lines 4000; ->` 가로사이즈 변경  
`set pages 5000; ->` 세로사이즈 변경

문제6. 부서번호를 출력하는데 중복을 제거해서 출력하시오.

답) Select distinct deptno

From emp;

DEPTNO
1
2
3

## • 연결 연산자 사용방법

오라클 연결 연산자 --> ||

Ex) select ename || '의 직업은' || job

From emp;

ENAME  '의직업은'  JOB
1 KING의 직업은PRESIDENT
2 BLAKE의 직업은MANAGER
3 CLARK의 직업은MANAGER
4 JONES의 직업은MANAGER
5 MARTIN의 직업은SALESMAN
6 ALLEN의 직업은SALESMAN
7 TURNER의 직업은SALESMAN
8 JAMES의 직업은CLERK
9 WARD의 직업은SALESMAN
10 FORD의 직업은ANALYST

문자열 은 ' '넣어 줘야함.

문제7. ename의 부서번호는 10 번 입니다. 라고 출력하시오.

답) Select ename || '의 부서번호는 ' || deptno || '입니다'

From emp;

문제8. 아래와 같이 결과가 출력되게하시오!

ENAME  '의월급은'  SAL  '입니다'
1 KING의 월급은5000 입니다
2 SCOTT의 월급은3000 입니다
3 FORD의 월급은3000 입니다
4 JONES의 월급은2975 입니다
5 BLAKE의 월급은2850 입니다
6 CLARK의 월급은2450 입니다
7 ALLEN의 월급은1600 입니다
8 TURNER의 월급은1500 입니다
9 MILLER의 월급은1300 입니다
10 WARD의 월급은1250 입니다

답) select ename || '의 월급은' || sal || ' 입니다'

FROM EMP

ORDER BY sal desc;

## • 컬럼 별칭 사용법

Select ename as "이름", sal as "월급"  
From emp;

	이름	월급
1	KING	5000
2	BLAKE	2850

문제9. 이름과 연봉을(sal\*12) 을 출력하는데 연봉이 높은 사원부터 출력하고,  
컬럼명을 한글로 이름, 연봉으로 출력되게 하시오.

답) Select ename as "이름", sal\*12 as "연봉"  
From emp  
Order by sal\*12 desc; // sal\*12 대신 연봉으로 쓸 수 있음

**오라클 실행순서 from -> select -> order by**

문제10. 이름, 월급, 직업을 출력하는데  
직업은 ABCD 순으로 출력되게 하고  
컬럼명을 한글로 이름, 월급, 직업으로 출력되게 하시오.

답) Select ename as "이름", sal as "월급", job as "직업"  
From emp  
Order by job asc;

문제11. 아래와 같이 출력하는데 최근에 입사한 사원부터 출력되게 하시오.

	사원정보
1	ADAMS의 입사일은15-JAN-83 입니다.
2	SCOTT의 입사일은22-DEC-82 입니다.
3	MILLER의 입사일은11-JAN-82 입니다.
4	FORD의 입사일은11-DEC-81 입니다.
5	JAMES의 입사일은11-DEC-81 입니다.
6	KING의 입사일은17-NOV-81 입니다.
7	MARTIN의 입사일은10-SEP-81 입니다.
8	TURNER의 입사일은21-AUG-81 입니다.
9	CLARK의 입사일은09-MAY-81 입니다.
10	BLAKE의 입사일은01-MAY-81 입니다.

답)

SELECT ename || '의 입사일은' || hiredate || ' 입니다.' AS "사원정보"  
FROM EMP  
ORDER BY hiredate DESC;

## • Null 값의 의미(0, 공백 아님)

1. 데이터가 없는 상태
2. 알수 없는 값

Ex) select ename, sal, comm  
from emp;

	ENAME	SAL	COMM
1	KING	5000	(null)
2	BLAKE	2850	(null)

	ENAME	SAL	COMM
1	KING	5000	(null)
2	BLAKE	2850	(null)
3	CLARK	2450	(null)
4	JONES	2975	(null)

Select ename, sal, comm, sal + comm  
From emp;

	ENAME	SAL	COMM	SAL + COMM
1	KING	5000	(null)	(null)
2	BLAKE	2850	(null)	(null)
3	CLARK	2450	(null)	(null)
4	JONES	2975	(null)	(null)

**Null 이 있으면 그룹함수를 사용할때  
연산이 빨라지는 장점이 있다.**

Ex) select sum(comm) from emp;  
select sum(nvl(comm,0)) from emp;

**문제 12. 이름, 월급 커미션을 출력하는데 커미션이 null인 사원들은 0으로 출력하시오.**

답) select ename, sal, nvl(comm,0)  
From emp;

	ENAME	SAL	NVL(COMM,0)
1	KING	5000	0
2	BLAKE	2850	0
3	CLARK	2450	0
4	JONES	2975	0

**문제 12-1. 이름, 월급, 커미션, 월급 + 커미션을 출력하는데 요번달 월급을 줄 수 있도록 출력하시오.**

답) select ename, sal, comm, sal + nvl(comm,0)  
From emp;

	ENAME	SAL	COMM	SAL + NVL(COMM,0)
1	KING	5000	(null)	5000
2	BLAKE	2850	(null)	2850
3	CLARK	2450	(null)	2450
4	JONES	2975	(null)	2975

## 2. 데이터 제한 및 정렬

2018년 4월 17일 화요일    오후 7:24

## 2장. 데이터 제한 및 정렬

1. Where 절 사용법
2. 비교 연산자 사용법
3. 논리 연산자 사용법
4. 연산자 우선순위

### 2-1. where 절 사용법

Ex) select 컬럼명  
From 테이블명  
Where 검색 조건

사원번호가 7788번인 사원의 사원번호와 이름을 출력하시오.

```
Select empno, ename  
From emp  
Where empno = 7788;
```

문제14. 월급이 3000인 사원의 이름과 월급을 출력하시오.

답) Select ename, sal  
From emp  
Where sal = 3000;

EMPNO	ENAME	SAL
1	FORD	3000
2	SCOTT	3000

문제15. 월급이 2500이상 사원들의 이름과 월급을 출력하시오.

답) Select ename, sal  
from emp  
where sal >= 2500;

EMPNO	ENAME	SAL
1	KING	5000
2	BLAKE	2850
3	JONES	2975
4	FORD	3000
5	SCOTT	3000

문제16. 연봉(sal\*12)가 36000이상인 사원들의 이름과 연봉을 출력하는데  
컬럼명을 이름과 연봉으로 출력



답) Select ename as "이름", sal \* 12 as "연봉"  
 from emp  
 where sal\*12 >= 36000; //sal\*12 대신 '연봉' 안됨 => 실행순서 때문  
 에

문제17. 직업이 SALESMAN인 직원들의 이름과 직업을 출력하시오.

답) Select ename, job  
 from emp  
 where job='SALESMAN'; => 문자와 날짜는 작은따옴표 있어야함.

## 2-2 기타 비교 연산자 사용법

- 1 . Between and
- 2 . Like
- 3 . In
- 4 . Is null

### 2-2-1. Between and

문제18. 월급이 1000에서 3000사이인 직원들의 이름과 월급을 출력하시오.

Select ename, sal  
 from emp  
 where sal between 1000 and 3000;

문제19. 월급이 1000에서 3000사이가 아닌 직원들의 이름과 월급을 출력하시오.

Select ename, sal  
 from emp  
 where sal not between 1000 and 3000;

문제20. 1981년 12월 11일에 입사한 이름과 입사일을 출력하시오.

6	NLS_CALENDAR	GREGORIAN
7	NLS_DATE_FORMAT	DD-MON-RR
8	NLS_DATE_LANGUAGE	AMERICAN

select ename, hiredate  
 from emp  
 where hiredate = '11-DEC-81';

-년도 데이터를 검색할 때에는 현재 내가 접속한 프로그램에 날짜 형식이 어떻게  
 되어있는지 확인을 하고 날짜 검색을 해야 제대로 검색할 수 있다.

(서양 날짜 형식 : 일/월/년도)

동양 날짜 형식 : 년도/월/일)

=> 확인방법 : select \*

From nls\_session\_parameters;

문제21. 1981년 입사한 직원들의 이름과 입사일을 출력하시오.

Select ename, hiredate  
 from emp  
 where hiredate = 'between 1 and 31 - between JEN and DEC-81';

문제22. 직업이 SALESMAN이 아닌 직원들의 이름과 직업을 출력하시오.

```
Select ename, hiredate
from emp
where job != 'SALESMAN';
```

문제23. 부서번호가 30번이 아닌 직원들의 이름과 부서번호를 출력하시오.

```
Select ename, deptno
from emp
where deptno != 30;
```

## 2-2-2. Like ( ~ 일 것 같은)

**%(like일때만) = wild card (이 자리에 뭐가 와도 상관없다.)**

**\_ : 이 자리에 어떻게 와도 상관없는데 자릿수는 한 자리여야한다.**

**ex) 이름의 첫번째 철자가 S 로 시작하는 직원들의 이름을 출력하시오.**

```
Select ename
From emp
Where ename like 'S%';
```

문제24. 이름의 끝글자가 T 로 끝나는 직원들의 이름을 출력하시오.

```
Select ename
from emp
where ename like '%T';
```

문제25. 이름의 두번째 철자가 M 인 직원들의 이름을 출력하시오.

```
Select ename
from emp
where ename like '_M%';
```

문제26. 이름의 세번째 철자가 L 인 직원들의 이름을 출력 하시오.

```
Select ename
from emp
where ename like '__L%';
```

문제27. 81년도에 입사한 직원들의 이름과 입사일을 출력하는데 **between** 쓰지 말고  
**like** 연산자로 출력하시오.

```
Select ename, hiredate
from emp
where hiredate like '%81';
```

아래의 insert 문을 실행해서 데이터를 입력하시오.

```
Insert into emp(empno, ename, sal)
values( 1234, 'A%B', 3500);
```

문제28. 이름의 두 번째 철자가 % 인 직원의 이름을 출력하시오.

```
Select ename
from emp
where ename like '_m%%' ESCAPE 'm'; -> m 바로 다음에 나오는 %는 wild card가 아니라 특수문자이다.(m 말고 다른단어 써도됨)
```

**문제29. 이름의 두 번째 철자와 세번째 철자가 % 인 사원의 이름을 출력하시오.**

```
Select ename
from emp
where ename like '_k%%%' escape 'k';
```

**문제30. 이름의 첫 번째 철자가 S 로 시작하지 않는 사원들의 이름을 출력하시오.**

```
Select ename
from emp
where ename not like 'S%';
```

### 2-2-3. In 연산자(여러 개 비교할 때), 하나 비교할 때는 =

**문제31. 직원번호가 7788, 7902, 7369 인 사원들의 직원번호와 이름을 출력하시오.**

```
Select ename, empno
from emp
where empno in(7788, 7902, 7369);
```

**문제32. 직업이 SALESMAN, ANALYST 인 사원들의 이름과 직업을 출력하시오.**

```
Select ename, job
from emp
where job in('SALESMAN', 'ANALYST');
```

**문제33. 직업이 SALESMAN, ANALYST 가 아닌 사원들의 이름과 직업을 출력하시오.**

```
Select ename, job
from emp
where job not in('SALESMAN', 'ANALYST');
```

### 2-2-4. Is null 연산자

**문제34. 커미션이 null 인 사원들의 이름과 커미션을 출력하시오.**

```
Select ename, comm
from emp
where comm is null;
```

**문제35. 커미션이 null 이 아니고 직업이 SALESMAN 인 사원들의 이름과 월급과 직업과 커미션을 출력하는데, 커미션이 높은 사원부터 출력하고, 컬럼명을 이름, 월급, 커미션 으로 출력되게 하시오.**

```
Select ename as "이름", sal as "월급", comm as "커미션"
From emp
where comm is not NULL or job = 'SALESMAN'
ORDER by comm desc;
```

## 반장자리에 데이터베이스 접속하는 방법

SQL gate 실행

새 연결 클릭 후 아래의 정보를 입력.

호스트 :192. 168. 19. 3

유저이름 : scott

패스워드 : tiger

서비스 이름 : xe

연결 모드 : normal

**문제36. 이름과 전공과 나이를 출력하는데 나이가 높은 학생부터 출력하시오.**

```
Select ename, major, age
From emp2
Order by age desc;
```

**문제37. 나이가 27에서 31 사이인 학생들의 이름과 나이와 전공과 주소를 출력하시오.**

```
Select ename, age, major, address
From emp2
Where age between 27 and 31;
```

% between .. And : >= and <= 이다.

**문제 38. 성이 김씨인 학생들의 이름과 나이를 출력하시오.**

```
SELECT ename, age
FROM emp2
WHERE ename LIKE '김%';
```

**문제 39. 전공이 컴퓨터 관련 학과가 아닌 학생들의 이름과 전공을 출력하시오.**

```
SELECT ename, major
FROM emp2
WHERE major NOT like '%컴퓨터%';
```

## 2-3. 논리 연산자 (and, or, not)

**문제40. 직업이 SALESMAN 이고 월급 1200 이상인 직원들의 이름과 월급과 직업을 출력하시오.**

```
SELECT ename, sal, job
FROM emp
WHERE job = 'SALESMAN' AND sal >= 1200;
```

% 진리 연산표

T & T = T	T or T = T
T & F = F	T or F = T
F & F = F	F or F = F

**문제41. 커미션이 null이고 부서번호가 20번인 직원들의 이름과 월급과 부서번호와 커미션을 출력하시오.**

```
SELECT ename, sal, deptno
FROM emp
WHERE COMM IS
AND deptno = 20;
```

**문제42. 전공이 컴퓨터 관련학과 이고, 나이가 20대인 학생들의 이름과 나이와 전공을 출력하시오.**

```
SELECT ename, age, major
FROM emp2
WHERE major LIKE '%컴퓨터%' AND age between 20 and 29;
```

% 가급적이면 문자는 like, 숫자는 between 쓰는게 좋다.

**문제43. 주소가 서울이고 통신사가 sk 인 학생들의 이름과 주소와 통신사를 출력하시오.**

```
SELECT ename, address, telecom
FROM emp2
WHERE address LIKE '서울%' AND telecom = 'sk';
```

```
SQL> create user scott
2 identified by tiger;

User created.

SQL> grant dba to scott;

Grant succeeded.
```

```
SQL> connect scott/tiger
Connected.
```

## 링크 걸기

```
SQL> create database link our_class_link
2 connect to scott
3 identified by tiger
4 using '192. 168. 19. 3 : 1521/xe';

Database link created.
```

**문제44. 전공이 컴퓨터 관련학과의 아닌 학생들의 이름과 나이와 전공을 출력하는데, 나이가 높은 학생부터 출력하고 컬럼명을 한글로 이름, 나이, 전공으로 출력되게 하시오.**

```
SELECT ename AS "이름", age AS "나이", major AS "전공"
FROM EMP2
WHERE major NOT LIKE '%컴퓨터%'
ORDER BY age DESC;
```

## 2-5. 연산자 우선순위

연산자	의미
1	산술 연산자
2	연결 연산자
3	비교 조건
4	IS [NOT] NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	같지 않음
7	NOT 논리 조건
8	AND 논리 조건
9	OR 논리 조건

% 산술 연산자에도 우선순위가 있다

Ex) select ename, sal + 10\*12  
From emp;

% 우선순위를 바꾸고 싶을때는 () 사용

Ex) select ename, (sal +10)\*12  
From emp;

### • 논리 연산자의 우선순위를 확인하는 SQL

Select ename, sal, job  
From emp  
Where job = 'SALESMAN'  
Or job = 'ANALYST'  
And sal > 1500;

and 먼저수행 후 or 수행

	ENAME	SAL	JOB
1	MARTIN	1250	SALESMAN
2	ALLEN	1600	SALESMAN
3	TURNER	1500	SALESMAN
4	WARD	1250	SALESMAN
5	FORD	3000	ANALYST
6	SCOTT	3000	ANALYST

문제45. 위의 SQL의 논리 연산자가 OR 부터 수행되게 쿼리를 변경하시오.

Select ename, sal, job  
From emp  
Where ( job = 'SALESMAN'  
Or job = 'ANALYST' )  
And sal >1500;

	ENAME	SAL	JOB
1	ALLEN	1600	SALESMAN
2	FORD	3000	ANALYST
3	SCOTT	3000	ANALYST

- 연산자의 종류 3가지

1. 산술 연산자 : \* / + -

2. 비교 연산자 : >, <, >=, <=, <>, !=, ^= <-- 같지 않다.

3. 논리 연산자 : and, or , not

### 3. 함수(단일행 함수, 복수행 함수)

2018년 4월 17일 화요일    오후 7:41

#### 3-1 단일행 함수

"데이터 검색을 할때 함수가 필요한 이유는 함수를 이용하면 더 다양하게 데이터를 검색할 수 있기 때문에 필요하다"

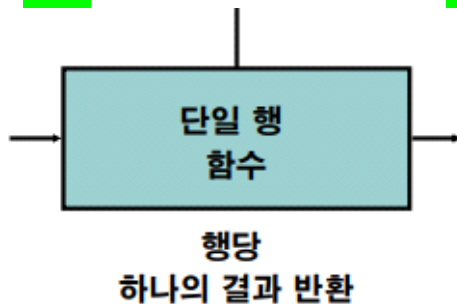
Ex) % 우리반에 네이버 메일을 사용하는 학생이 몇 명이나 되는지?

% 우리반 학생들이 어느 통신사를 가장 많이 사용 하는지 확인하고 싶다면?

- 함수의 종류 2가지

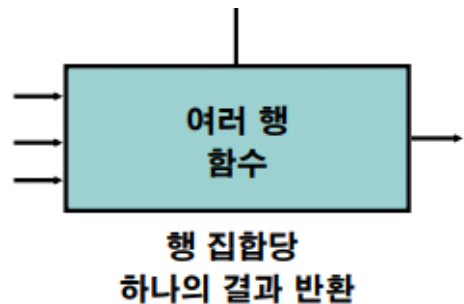
##### 1. 단일행 함수

한 개의 로우가 입력되고  
한 개의 로우가 출력됨



##### 2. 복수행 함수

여러 개의 로우가 입력되서  
한 개의 로우가 출력됨



##### 3-1) 단일 행 함수

- 문자 : upper, lower, initcap, substr, instr, replace, concat, lpad, rpad, ltrim, rtrim, trim
- 숫자 : round, trunc, mod
- 날짜 : months\_between, add\_months, next\_day, last\_day
- 변환 : to\_char, to\_number, to\_date
- 일반 : nvl, nvl2, decode, case

##### 1. 복수 행 함수

- Max, min, avg, sum, count

#### 문자함수

문제46. 아래의 SQL을 수행하고 결과를 보시오

```
Select upper(ename), lower(ename), initcap(ename)
From emp;
```

	UPPER(ENAME)	LOWER(ENAME)	INITCAP(ENAME)
1	KING	king	King
2	BLAKE	blake	Blake
3	CLARK	clark	Clark
4	JONES	jones	Jones
5	MARTIN	martin	Martin



- **Substr**: 단어에서 특정 철자만 잘라내는 함수

```
Select ename, substr(ename, 1, 3)  <= 1번째자리부터 3개를 가져와라
From emp;
```

문제47. 이름의 첫번째 철자만 출력하시오.

```
SELECT ename, substr(ename, 1, 1)
FROM EMP;
```

문제48. 우리반 테이블에서 성이 이씨 인 학생들의 이름과 나이를 출력하시오.(like x)

```
SELECT ename, age
FROM EMP2
WHERE substr(ename, 1, 1) = '이';
```

Ex) SELECT ename, substr(ename, -2, 2)  
FROM EMP;

~~A~~ B C ~~D~~

문제49. 우리반 테이블에서 주소를 아래와 같이 출력하시오.

□	ENAME	SUBSTR(ADDRESS, -3, 3)
1	김원섭	와부읍
2	김광록	하갈동
3	유혜린	방미동
4	이유진	사당동
5	윤진민	교문동
6	백광훈	공릉동
7	장은희	군자동

```
SELECT ename, substr(address, -3, 3)
FROM EMP2;
```

- **Instr** 함수 : 특정 철자가 단어에서 몇번째 철자인지 확인하는 함수(여러 개 이면 먼저나온거)

```
Ex) select ename, instr(ename, 'L')
From emp;
```

문제50. 이메일, 이메일에서 @가 몇번째 철자에 있는지 출력

```
SELECT ename, email, instr(email, '@')
FROM EMP2;
```

상)문제51. 아래와 같이 결과를 출력 하시오.

□	SUBSTR(EMAIL, INSTR(EMAIL, '@')+1)
1	naver.com
2	naver.com
3	gmail.com
4	gmail.com
5	gmail.com
6	naver.com

```
SELECT SUBSTR (email, instr(email, '@')+1)
FROM EMP2;
```

- **trim** 함수 : 양쪽 특정 철자를 잘라내는 함수 (**ltrim** : 왼쪽짜름, **rtrim** : 오른쪽짜름)

```
Ex) select ename, trim(ename, 'S')
From emp;
```

SCOTT	COTT
-------	------

문제52. 아래와 같이 결과를 출력하시오.

	RTRIM(SUBSTR(EMAIL, INSTR(EMAIL, '@') + 1), '.com')
1	naver
2	naver
3	gmail
4	gmail
5	gmail
6	naver
7	naver
8	naver
9	naver
10	naver

```
SELECT RTRIM(SUBSTR (email, instr(email, '@')+1), '.com')
FROM EMP2;
```

### % Trim 함수2

아래의 데이터를 입력하세요.

```
Insert into emp(empno, ename, sal)
Values(2035, 'JANE ', 4500);
```

문제53. 이름이 JANE인 사원의 이름과 월급을 출력하시오.

```
SELECT ename, sal
FROM EMP
WHERE TRIM(ename) = 'JANE';
```

<== 공백이있을때 trim을 써주면 나옴

### - replace 함수 : 특정 철자를 다른 철자로 변경하는 함수

Ex) select ename, replace(sal, 0, '\*')  
From emp;

	ENAME	REPLACE(SAL,0,'*')
1	KING	5***
2	BLAKE	285*
3	CLARK	245*

### - Regexp\_replace 함수 : 여러 개의 철자를 한번에 다른 철자로 변경하는 함수

Ex) select ename, regexp\_replace(sal, '[0-2]', '\*')  
From emp;

### - concat 함수 : 두개의 컬럼 데이터를 연결해서 출력하는 함수

Ex) select concat(ename, sal)  
From emp;

1	KING5000
2	BLAKE2850
3	CLARK2450
4	JONES2975

### - Lpad, rpad 함수 : 특정 철자를 컬럼의 데이터로 채워넣을 때 사용하는 함수

Rpad는 오른쪽에 채워넣음

Ex) select ename, lpad(sal, 10, '\*')  
From emp;

	ENAME	LPAD(SAL, 10, '*')
1	KING	*****5000
2	BLAKE	*****2850
3	CLARK	*****2450
4	JONES	*****2975
5	MARTIN	*****1250
6	ALLEN	*****1600
7	TURNER	*****1500
8	JAMES	*****950

- **Length** 함수 : 단어의 **철자의 갯수**를 세는 함수

Ex) select ename, length(ename)  
From emp;

문제54. 이름의 철자의 개수가 5개 이상인 직원들의 이름과 이름의 철자의 개수를 출력하시오.

```
SELECT ename, Length(ename)
FROM EMP
WHERE length(ename) >= 5;
```

	ENAME	LENGTH(ENAME)
1	BLAKE	5
2	CLARK	5
3	JONES	5
4	MARTIN	6
5	ALLEN	5
6	TURNER	6
7	JAMES	5
8	SMITH	5
9	SCOTT	5

문제55. 우리반 테이블에서 이름, 주소, 주소의 길이를 출력하는데 주소의 길이가 가장 긴 학생부터 출력하시오.

```
SELECT ename, address, LENGTH(address)
FROM EMP2
ORDER BY Length(address) DESC;
```

	ENAME	ADDRESS	LENGTH(ADDRESS)
1	한지윤	수원시 팔달구 월드컵로 206	16
2	김광록	경기도 용인시 기흥구 하갈동	15
3	신영근	경기도 안양시 만안구 석수동	15
4	미한새	경기도 용인시 기흥구 상하동	15
5	김영토	경기도 안양시 동안구 평촌동	15
6	미찬중	서울시 영등포구 영등포동	13
7	김동윤	서울시 영등포구 당산로	12
8	윤동환	경기도 부천시 역곡1동	12
9	김원섭	경기도 남양주시 와부읍	12
10	정인중	경기도 남양주시 별내면	12

## 숫자 함수

- **Round** : **반올림** 하는 함수

Ex) select 2567.56, round(2567.56, 1)     **2 5 6 7 . 5 6**

FROM dual;    <== dual : 결과를 보기 위한 가상의 테이블

% -1할 경우 -1을 반올림함

	2567.56	ROUND(2567.56,1)
1	2567.56	2567.6

**-trunc 함수 : 잘라내는 함수 (입력한 숫자까지 나오고 나머지 자름)**

Ex) select 2567.56, round(2567.56, 1) <= 1번째 까지 나오고 자름  
FROM dual;

**-mod 함수 : 나눈 나머지 값을 출력하는 함수**

Ex) select mod(10,3)  
From dual;

<input checked="" type="checkbox"/>	MOD(10,3)
1	1

**-power 함수 : n제곱 값을 출력하는 함수 ( power(2,n) )**

Ex) select power(2,3)  
From dual;

<input checked="" type="checkbox"/>	POWER(2,3)
1	8

**문제56. 아래와 같이 결과를 출력하시오.**

1	rladnjstjqek
2	kwangrok92
3	yohhr22
4	yjin1435
5	yjjmm92
6	bkh974
7	ja5772
8	dudxh1
9	songssmart
10	hoseong0701

SELECT substr(email, 1, instr(email, '@')-1)  
FROM EMP2;

### 날짜함수

- Months\_between,
- add\_months,
- next\_day,
- last\_day

날짜 - 날짜 = 숫자

날짜 - 숫자 = 날짜

날짜 + 숫자 = 날짜

- 오늘날짜를 보는 키워드

Select sysdate  
From dual;

**문제57. 이름, 입사한 날짜부터 오늘까지 총 몇일 근무했는지 출력하시오.**

SELECT ename, ROUND(SYSDATE - hiredate,0)  
FROM EMP;

### Months\_between

**문제58. 이름, 입사한 날짜부터 오늘까지 총 몇 달 근무했는지 출력하시오.**

SELECT ename, round(months\_between(SYSDATE, hiredate),0)  
FROM EMP;

### Add\_months

문제59. 오늘부터 100달 뒤에 돌아오는 날짜를 출력하시오.

```
SELECT ADD_MONTHS(SYSDATE,100)
FROM dual;
```

### Next\_day

문제60. 오늘부터 앞으로 돌아올 월요일의 날짜를 출력하시오.

```
SELECT SYSDATE, NEXT_DAY(SYSDATE, 'monday')
FROM dual;
```

문제61. 오늘부터 100달뒤에 돌아오는 월요일의 날짜를 출력하시오.

```
SELECT sysdate, NEXT_DAY(ADD_MONTHS(SYSDATE, 100), 'monday')
FROM dual;
```

### Last\_day

문제62. 요번달의 마지막 날짜를 출력하시오.

```
SELECT SYSDATE, LAST_DAY(sysdate)
FROM dual;
```

문제63. 오늘부터 요번달 말일까지 총 몇일 남았는지 출력하시오.

```
SELECT LAST_DAY(SYSDATE) - SYSDATE
FROM dual;
```

### 변환 함수

변환을 한다는 것은 무엇을 변환한다는 것인가?

- 형 변환을 하겠다는 것이다. <== 숫자형 > 문자형 > 날짜형

Ex1) where sal = '3000'; == where sal = 3000; 이러면 시간이 오래걸림

오라클이 내부적으로 실행한 SQL을 확인하고 싶으면,

**Set autot to (sql 게이트에서는 안됨, 게이트에서는 c+a+f7)**

Ex2)SELECT ename, sal  
FROM EMP

WHERE sal LIKE '30%'; 문자가 숫자로 바뀌어야하는데 30%라서 더느리게 바뀜

## - 데이터 유형(타입)의 변환 2가지

### 1. 암시적 형 변환

Ex) SELECT ename, sal  
FROM EMP  
WHERE sal LIKE '30%';

### 2. 명시적 형 변환- 우리가 임의로 데이터 유형을 변경하는 것

- To\_char : 문자형으로 변환하는 함수
- To\_number : 숫자 형으로 변환하는 함수
- To\_date : 날짜 형으로 변환하는 함수

### 숫자 -> 문자

Ex) select sal, to\_char(sal)  
From emp;

	SAL	TO_CHAR(SAL)
1	5000	5000
2	2850	2850
3	2450	2450

Ex2)select sal, to\_char(sal, '999,999') <- 문자 포맷

From emp;

	SAL	TO_CHAR(SAL, '999,999')
1	5000	5,000
2	2850	2,850
3	2450	2,450

**문제64. 이름, 월급 \* 2093045 를 출력하는데, 숫자 단위를 보기 편하게 천 단위도 표시하고 백만 단위로 표시해서 출력하시오.**

```
SELECT ename, TO_CHAR(sal * 2093045, '999,999,999,999')
FROM EMP;
```

날짜 -> 문자

**문제65. 오늘 요일을 출력하시오.**

```
Select to_char(sysdate, 'day') <- 문자 포맷
From dual;
```

**문제66. 내가 태어난 날짜의 요일을 출력하시오.**

```
SELECT ename, TO_CHAR(birth, 'day')
FROM EMP2
WHERE ename = '김원섭';
```

## - 날짜 포맷

1. 년도 : RRRR, RR, YYYY, YY
2. 달 : MM, MON
3. 일 : dd
4. 요일 : day, dy
5. 시간 : HH, HH24
6. 분 : MI
7. 초 : SS
8. 주 : WW, IW

**문제67. 토요일에 태어난 학생들의 이름과 생일과 주소를 출력하시오.**

```
SELECT ename, birth, address
FROM EMP2
WHERE TRIM(TO_CHAR(birth, 'day')) = 'saturday';
```

	ENAME	BIRTH	ADDRESS
1	이찬중	1991-10-26 오전 12:00:00	서울시 영등포구 영등포동
2	도웅	1991-01-12 오전 12:00:00	안양시 동안구 평촌동
3	김동윤	1991-06-22 오전 12:00:00	서울시 영등포구 당산로

문제68. 오늘부터 100 달 뒤에 돌아오는 날짜의 요일을 출력하시오.

```
SELECT SYSDATE, TO_CHAR(ADD_MONTHS(SYSDATE, 100), 'day')
FROM dual;
```

문제69. 이름, 커미션을 출력하시오.

```
SELECT ename, comm
FROM EMP;
```

문제70. 이름, 커미션을 출력하는데 커미션이 null 인 직원들은 0으로 출력

```
SELECT ename, nvl(comm, '0')
FROM EMP;
```

문제71. 아래와 같이 출력하시오.

	ENAME	NVL(TO_CHAR(COMM), 'NOCOMM')
1	KING	no comm
2	BLAKE	no comm
3	CLARK	no comm
4	JONES	no comm
5	MARTIN	1400
6	ALLEN	300

```
SELECT ename, NVL(to_char(comm), 'no comm')
FROM EMP;
```

문제72. 이름과 mgr(관리자번호)를 출력하는데 mgr이 null 인 직원들은 no manager  
란 글씨로 출력하시오.

```
SELECT ename, NVL(TO_CHAR(mgr), 'nomanager')
FROM EMP;
```

	ENAME	NVL(TO_CHAR(MGR), 'NOMANAGER')
1	KING	nomanager
2	BLAKE	7839
3	CLARK	7839
4	JONES	7839

문제73. 아래와 같이 결과를 출력하시오.

	ENAME    '의 커미션은'    NVL(COMM, 0)    '입니다'
1	MARTIN의 커미션은1400입니다
2	WARD의 커미션은500입니다
3	ALLEN의 커미션은300입니다
4	JONES의 커미션은0입니다
5	TURNER의 커미션은0입니다
6	JAMES의 커미션은0입니다
7	FORD의 커미션은0입니다
8	SMITH의 커미션은0입니다

```
SELECT ename || '의 커미션은' || NVL(comm, 0) || '입니다'
FROM EMP
ORDER BY NVL(comm, 0) desc;
```

## 정렬할때 null 을 조정하는 방법

문제74. 이름, 커미션을 출력하는데 커미션이 높은 직원부터 출력하시오.

```
SELECT ename, comm
FROM EMP
ORDER BY comm DESC NULLs last;
First;
```

문제75. 이름, 커미션을 출력하는데 커미션이 낮은 직원부터 출력하시오.

```
SELECT ename, comm
FROM EMP
ORDER BY COMM ASC nulls FIRST;
```

문제 76. 81년 9월 10일에 입사한 사원들의 이름과 입사일을 출력하시오.

- **현재 세션의 날짜 형식을 확인하고 변경하는 방법**

```
Alter session set nls_date_format = 'YY/MM/DD';
```

## 우변 가공하는방법 (속도빨라짐)

좌변을 변경한거라 성능이 안좋은.



## 일반함수 (nvl, nvl2, decode, case)

### Nvl2 함수

Select ename, sal, comm, nvl2(comm, sal+comm, sal) <= 커미션이 null이면 sal+comm  
From emp; null이 아니면 sal 출력

### Decode 함수 (등호 비교만 가능)

Ex) 이름, 월급, 부서번호, 보너스를 출력하시오.

(보너스가 부서번호가 10번이면 6000을 출력

부서번호가 20번이면 3400을 출력

나머지 부서번호는 0을 출력)

Select ename, sal, deptno, decode(deptno, 10, 6000,  
20, 3400, 0 ) 보너스

From emp;

	ENAME	SAL	DEPTNO	보너스
1	KING	5000	10	6000
2	BLAKE	2850	30	0
3	CLARK	2450	10	6000
4	JONES	2975	20	3400
5	MARTIN	1250	30	0
6	ALLEN	1600	30	0

문제78. 이름, 직업, 월급, 보너스를 출력하는데, 직업이 SALESMAN 이면 보너스를  
10000을 출력하고 나머지 직업은 다 0을 출력하시오.

SELECT ename, job, sal, DECODE(job, 'SALESMAN', 10000, 0) 보너스  
FROM EMP;

### Case 함수(부등호, 등호 가능)

Ex) 이름, 나이, 보너스를 출력하는데 나이가 30살 이상이면 보너스를

50000을 출력 나머지 나이는 0을 출력하시오.

SELECT ename, age,  
CASE WHEN age >= 30 THEN 50000  
ELSE 0 end AS "보너스"  
FROM EMP2;

27	이상민	26	0
28	정호진	32	50000
29	지윤철	28	0
30	김건태	28	0

문제79. 이름, 나이, 보너스를 출력하는데 나이가 25 이하면 6000을 보너스로 출력,  
나이가 26~29살 이면 8000을 출력, 나이가 30살 이상이면 9000을 출력

SELECT ename, age,  
CASE WHEN age <= 25 THEN 6000  
WHEN age BETWEEN 26 AND 29 THEN 8000  
else 9000 END AS "보너스"  
FROM EMP2;

## 3-2 복수행 함수

### 목차

3-2-1) 그룹함수가 무엇인가 ?

3-2-2) 그룹함수 max

3-2-3) 그룹함수 min

3-2-4) 그룹함수 avg

3-2-5) Having 절

3-2-6) 그룹함수 sum

3-2-7) 그룹함수 count

3-2-8) 그룹함수 중첩

### 3-2-1) 그룹함수가 무엇인가?

- 여러 개의 행 데이터를 집합 연산 수행하여 하나의 결과로 출력하는 함수 (기본적으로 null값을 무시함)

### 3-2-2) max

Ex) select max(sal) <= 최대 월급  
From emp;

문제80. 직업이 SALESMAN 인 사원들 중에서 최대 월급을 출력하시오.

```
SELECT MAX(sal)
FROM EMP
WHERE job = 'SALESMAN';
```

	MAX(SAL)
1	5000

### Group by

SELECT job, MAX(sal) <= job은 여러 개, max(sal)은 한 개라 오류가 떠서 그룹 바이절을 사용  
FROM EMP  
WHERE job = 'SALESMAN'  
GROUP BY job; <== job data를 그룹핑 하겠다

```
SELECT job, MAX(sal)
FROM EMP
GROUP BY job;
```

각 직업별로 최대월급

문제81. 부서번호, 부서번호별로 최대월급을 출력하시오.

```
SELECT deptno, max(sal)
FROM EMP
GROUP BY deptno;
```

	DEPTNO	MAX(SAL)
1	30	2850
2	20	3000
3	10	5000

문제82. 위의 결과에서 부서번호 30번은 제외하고 출력하시오.

```
SELECT deptno, max(sal)
FROM EMP
WHERE DEPTno != 30
GROUP BY deptno;
```

## ★코딩순서 select -> From -> Where -> Group by -> order by

### 실행순서

문제83. 직업, 직업별 최대월급을 출력하는데 직업이 SALESMAN 인 직원들을 제외하고 출력하고, 직업별 최대월급이 높은것부터 출력하시오.

```
SELECT job, MAX(sal)
FROM EMP
WHERE job != 'SALESMAN'
GROUP BY job
ORDER BY MAX(sal) DESC;
```

문제84. 통신사, 통신사별 최대 나이를 출력하시오

```
SELECT telecom, MAX(age)
FROM EMP2
GROUP BY telecom;
```

문제85. 전공, 전공별로 최대나이를 출력하시오.

```
SELECT major, MAX(age)
FROM EMP2
GROUP BY major;
```

### 3-2-3 min : 최소 값 출력하는 함수

문제86. 최소월급을 출력하시오.

```
SELECT MIN(sal)
FROM EMP;
```

문제87. 직업, 직업별 최소월급을 출력하시오.

```
SELECT job, MIN(sal)
FROM EMP
GROUP BY job;
```

문제88. 위에 결과를 최소월급이 높은것 부터 출력하시오.

```
SELECT job, MIN(sal)
FROM EMP
GROUP BY job
ORDER BY min(sal) DESC;
```

문제89. 위에 결과에서 직업이 SALESMAN 을 제외하시오.

```
SELECT job, MIN(sal)
FROM EMP
WHERE job != 'SALESMAN'
GROUP BY job
ORDER BY min(sal) DESC;
```

문제90. 우리반 테이블에서 나이, 나이별로 몇 명인지 출력하고, 높은 나이부터 출력되게 하고, 컬럼명을 나이, 건수 라고 출력되게 하시오.

```
SELECT age AS "나이", COUNT(age) AS "건수"
FROM EMP2
GROUP BY age
ORDER BY age desc;
```

문제91. 컴퓨터 관련 학과가 아닌 학생의 이름과 나이를 출력하는데, 나이가 높은 순서대로 출력하시오.

```
SELECT ename, age
FROM EMP2
WHERE major NOT LIKE '%컴퓨터%'
ORDER BY age DESC;
```

문제92. 부서번호, 부서번호별 최대월급을 출력하는데 20번 부서번호는 제외하고 출력하고, 부서번호별 최대월급이 높은 것 부터 출력하시오.

```
SELECT deptno, MAX(sal)
FROM EMP
WHERE deptno != 20
GROUP BY deptno
ORDER BY max(sal) DESC;
```

### 3-2-4 avg 그룹함수 : 평균 값

문제93. 이름, 커미션을 출력하시오

```
SELECT ename, comm
FROM EMP;
```

	ENAME	COMM
1	KING	(null)
2	BLAKE	(null)
3	CLARK	(null)
4	JONES	(null)
5	MARTIN	1400
6	ALLEN	300
7	TURNER	0
8	JAMES	(null)
9	WARD	500

문제94. 커미션이 null이 아닌 직원들의 직업을 출력하시오.

```
SELECT job
FROM EMP
WHERE comm IS NOT NULL;
```

문제94. 위에 결과를 다시 출력하는데 중복 제거 출력하시오.

```
SELECT DISTINCT job
FROM EMP
WHERE comm IS NOT NULL;
```

문제96. 커미션의 평균값을 출력하시오.

```
SELECT AVG(comm)
FROM EMP;
```

	AVG(COMM)
1	550

null을 무시함

문제 97. 직원테이블의 커미션의 평균값을 출력하는데 null값도 포함하여 나누어 출력하시오.

```
SELECT ROUND(AVG(NVL((comm), 0))) <--round = 반올림
FROM EMP;
```

문제98. 아래의 두개의 SQL의 결과는 같을까? 틀릴까?

```
Select sum(comm) from emp;
```

가음

문제99. 직업, 직업별 평균월급을 출력하는데 직업별 평균월급이 높은 것 부터 출력하시오.

[illegible]

### 3-2-5) having

문제 100. 직업, 직업별 평균월급을 출력하는데 직업별 평균월급이 4000이상인 것만 출력하시오

having AVG(sal) >= 4000 <---- 그룹함수에는 where절을 사용못함

문제101. 부서번호, 부서번호별 최소월급을 출력하는데 부서번호별 최소월급이 2000이상인 것만 출력하시오.

```
SELECT deptno, MIN(sal)
      FROM EMP
     GROUP BY deptno
    HAVING MIN(sal) >= 2000;
```

★코딩순서 select -> from -> where -> group by -> having -> order by

### 3-2-6)sum

문제102. 직업, 직업별 토탈월급을 출력하는데 직업이 SALESMAN인 사원은 제외하고 출력  
하고 직업별 토탈월급이 5000이상인 것만 출력하고 직업별 토탈월급이 높은것부터  
출력하시오.

```
SELECT job, SUM(sal)
  FROM EMP
 WHERE job != 'SALESMAN'
 GROUP BY job
 HAVING SUM(sal) >= 5000
 ORDER BY SUM(sal) DESC;
```

문제 103. 위에 결과를 다시 출력하는데 토탈월급이 출력될때에 천단위로 출력되게하시오.

```
SELECT job, TO_CHAR(SUM(sal), '999,999')
FROM EMP
WHERE job != 'SALESMAN'
GROUP BY job
HAVING SUM(sal) >= 5000
ORDER BY SUM(sal) DESC;
```

<-- where 절을 having절에 쓰면 속도 느려짐

문제104. 이름, 입사일, 입사한 년도(4자리) 를 출력하시오.

```
SELECT ename, hiredate, TO_CHAR(hiredate, 'RRRR')
FROM EMP;
```

% MM, MON, DD, DAY 다가능

**문제105. 입사년도(4자리), 입사한 년도별 토탈월급 출력하시오.**

```
SELECT TO_CHAR(hiredate, 'RRRR'), SUM(sal)
FROM EMP
GROUP BY TO_CHAR(hiredate, 'RRRR');
```

**문제106. 입사한 년도(4자리), 입사한 년도별 토탈월급, 입사한 년도별 평균월급, 입사한 년도별 최대 월급, 입사한 년도별 최소 월급을 출력하시오.**

### 3-2-7) Count 함수 : 테이블의 행의 건수를 세는 함수

Ex) select count(empno)

<= 만약 comm 쓰면 null값을 무시해서 4가나옴

From emp;

사원수를 셀때는 \* 쓰는게 좋음

	COUNT(EMPNO)
1	14

**문제107. 직업, 직업별 인원수를 출력하시오.**

```
SELECT job, COUNT(*)
FROM EMP
GROUP BY job;
```

**문제108. 직업, 직업별 인원수를 출력하는데, 직업별 인원수가 4명 이상인 것만 출력하시오.**

```
SELECT job, COUNT(*)
FROM EMP
GROUP BY job
HAVING COUNT(*) >= 4;
```

**문제109. 직업, 직업별 인원수를 출력하는데, 직업별 인원수가 4명 이상인 것만 출력하고, 직업별 인원수가 높은것부터 출력하시오.**

```
SELECT job, COUNT(*)
FROM EMP
GROUP BY job
HAVING COUNT(*) >= 4
ORDER BY COUNT(*) DESC;
```

**문제110. 위의 결과를 출력하는데 직업이 SALESMAN는 제외하고 출력하시오.**

```
SELECT job, COUNT(*)
FROM EMP
WHERE job != 'SALESMAN'
GROUP BY job
HAVING COUNT(*) >= 4
ORDER BY COUNT(*) DESC;
```

**문제111. 우리반에서 통계학과인 학생들이 몇 명인지 출력하시오.**

```
SELECT COUNT(*)
FROM EMP2
where major like '%통계%';
```

**문제112. 아래와 같이 결과를 출력하시오.**

	SUBSTR(ADDRESS,1,3)	COUNT(*)
1	서울시	15
2	경기도	11
3	인천시	2
4	관악구	1
5	수원시	1
6	안양시	1

```
SELECT SUBSTR(address,1,3), COUNT(*)  
  FROM EMP2  
 GROUP by substr(address,1,3)  
 ORDER BY COUNT(*) DESC;
```

## 4. 중첩 그룹함수

2018년 4월 17일 화요일    오후 7:59

### -그룹함수 중첩

문제113. 직업을 출력하는데 중복을 제거해서 출력하고 몇건이 나왔는지 카운트하시오.  
(회사에 직업이 몇가지 있는지 확인)

```
SELECT count(DISTINCT job)
FROM EMP;
```

문제114. 우리반에 나이대가 몇 개가 있는지 출력하시오.

```
SELECT COUNT(DISTINCT age)
FROM EMP2;
```

문제115. 직업과 직업별 평균월급을 출력하시오.

```
SELECT job, AVG(sal)
FROM EMP
GROUP BY job;
```

	AVG(SAL)
1	1400
2	1037.5
3	5000
4	2758.333333
5	3000

문제116. 위에 평균 월급중에서 최대값을 출력하시오.

```
SELECT MAX(AVG(sal))
FROM EMP
GROUP BY job;
```

	MAX(AVG(SAL))
1	5000

문제117. 직업별로 인원수가 가장 많은 직업의 인원수를 출력하시오.

```
SELECT MAX(COUNT(*))
FROM EMP
GROUP BY job;
```

	MAX(COUNT(*))
1	4

문제118. 위에 가장 인원수가 많은 직업이 무엇인지 출력하시오. (서브쿼리가 필요)

- 그룹함수의 결과를 세로--->가로로 출력하는 방법

문제119. 부서번호, 부서번호별 토달월급을 출력하시오.

```
SELECT SUM(DECODE(deptno, 10, sal, 0)) AS "10",
       SUM(DECODE(deptno, 20, sal, 0)) AS "20",
       SUM(DECODE(deptno, 30, sal, 0)) AS "30"
FROM EMP;
```

	10	20	30
1	8750	10875	9400

문제120. 부서번호가 30번인 직원들의 월급만 출력하시오.



```
SELECT sal
FROM EMP
WHERE deptno = '30';
```

### 직업별, 부서별 토탈월급

```
SELECT job,
       SUM(DECODE(deptno, 10, sal, 0)) AS "10",
       SUM(DECODE(deptno, 20, sal, 0)) AS "20",
       SUM(DECODE(deptno, 30, sal, 0)) AS "30"
FROM EMP
GROUP BY job;
```

	JOB	10	20	30
1	SALESMAN	0	0	5600
2	CLERK	1300	1900	950
3	PRESIDENT	5000	0	0
4	MANAGER	2450	2975	2850
5	ANALYST	0	6000	0

문제121. 아래의 쿼리를 수행하시오.

```
SELECT DECODE(deptno, 10, sal, 0)
from EMP;
```

<- 부서번호가 10번이면 월급을 출력하고 아니면 0을출력해라

	DECODE(DEPTNO,10,SAL,0)
1	5000
2	0
3	2450
4	0
5	0
6	0
7	0
8	0

문제122. 위의 결과를 다 sum한 값이 어떻게 되는가?

```
SELECT SUM(DECODE(deptno, 10, sal, 0))
FROM EMP;
```

	SUM(DECODE(DEPTNO,10,SAL,0))
1	8750

문제122. 직업, 직업별 토탈월급을 출력하는데 아래와 세로---->가로로 출력하시오.

```
SELECT SUM(DECODE(job,'SALESMAN', sal)) AS "SALESMAN",
       SUM(DECODE(job,'CLERK', sal)) AS "CLERK",
       SUM(DECODE(job,'PRESIDENT', sal)) AS "PRESIDENT",
       SUM(DECODE(job,'MANAGER', sal)) AS "MANAGER",
       SUM(DECODE(job,'ANALYST', sal)) AS "ANALYST"
FROM EMP;
```

	SALESMAN	CLERK	PRESIDENT	MANAGER	ANALYST
1	5600	4150	5000	8275	6000

문제123. 입사한 년도, 입사한 년도별 토탈월급을 출력하시오.(세로)

```
SELECT TO_CHAR(hiredate,'RRRR'), SUM(sal)
FROM EMP
GROUP BY TO_CHAR(hiredate,'RRRR');
```

	TO_CHAR(HIREDATE,'RRRR')	SUM(SAL)
1	1980	800
2	1983	1100
3	1982	4300
4	1981	22825

문제124. 위의 출력을 가로로 출력하시오.

```
SELECT sum(DECODE(TO_CHAR(hiredate,'RRRR'),'1980',sal)) AS "1980",
       sum(DECODE(TO_CHAR(hiredate,'RRRR'),'1983',sal)) AS "1983",
       sum(DECODE(TO_CHAR(hiredate,'RRRR'),'1982',sal)) AS "1982",
       sum(DECODE(TO_CHAR(hiredate,'RRRR'),'1981',sal)) AS "1981"
FROM EMP;
```

	1980	1983	1982	1981
1	800	1100	4300	22825

- 가로 ---> 세로, 세로 ---> 가로 로 출력하는 함수
  - 가로 ---> 세로 : pivot 함수(회전하다)
  - 세로 ---> 가로 : unpivot 함수

문제125. 부서번호, 부서번호별 토탈원급을 가로로 출력하시오.

```
SELECT *
FROM (SELECT deptno, sal FROM emp) <-- ()안에있는걸 메모리안에다가 집어넣고, 테이블
      처럼 사용(서브쿼리)
pivot( SUM(sal) FOR deptno IN (10,20,30)); <-- sum(sal)을위한 deptno의 10,20,30을 출력
```

	10	20	30
1	8750	10875	9400

문제126. 직업, 직업별 토탈월급을 출력하는데 pivot을 써서 가로로 출력하시오.

```
SELECT *
FROM( SELECT job, sal FROM EMP)
pivot( SUM(sal) FOR job IN ('SALESMAN','CLERK','PRESIDENT','MANAGER','ANALYST'));
```

	'SALESMAN'	'CLERK'	'PRESIDENT'	'MANAGER'	'ANALYST'
1	5600	4150	5000	8275	6000

직업을 다쓰기 힘들니까 애초에 만들때

```
SELECT distinct '||job||','
FROM EMP;
```

	'  JOB  ','
1	'SALESMAN',
2	'CLERK',
3	'PRESIDENT',
4	'MANAGER',
5	'ANALYST',

우리반 테이블 정제작업

- 통신사 데이터 정제

```
UPDATE EMP2
```

```
SET telecom = 'lg'
WHERE telecom IN('LG', 'uplus', 'LGU+', 'LGT');
```

```
UPDATE EMP2
```

```
SET telecom = 'kt'
WHERE telecom IN('KT');
```

```
UPDATE EMP2
SET telecom = 'cjh'
WHERE telecom IN('CJH');
```

#### 1. 전공 데이터 정제

```
UPDATE EMP2
SET major = '컴퓨터공학'
WHERE major = '컴퓨터 공학';
UPDATE EMP2
SET major = '경영정보학'
WHERE major = '경영정보';
UPDATE EMP2
SET major = '기술융합공학'
WHERE major = '기술융합공학과';
UPDATE EMP2
SET major = '시스템경영공학'
WHERE major = '시스템경영공학부';
```

문제127. 통신사, 통신사별 인원수를 출력하는데 통신사별 인원수가 높은것부터 출력하시오.

```
SELECT telecom, COUNT(*)
FROM EMP2
GROUP BY telecom
ORDER BY COUNT(*) DESC;
```

	TELECOM	COUNT(*)
1	kt	13
2	lg	9
3	sk	8
4	cjh	1

문제128. 위의 결과를 가로로 출력하시오.(pivot)

```
SELECT *
FROM (SELECT telecom FROM emp2)
pivot (COUNT(*) FOR telecom IN ('kt', 'lg', 'sk', 'cjh')); <-- telecom 별로 각각 count 하겠다!
```

	'kt'	'lg'	'sk'	'cjh'
1	13	9	8	1

'kt' 에서 "없애는 방법 : in('kt' as "kt")사용하면됨

```
SELECT *
FROM (SELECT major, telecom FROM emp2) <-- major를 넣으면 전공별로 나옴 (다른컬럼도가능)
pivot (COUNT(*) FOR telecom IN ('kt' AS "kt", 'lg' AS "lg", 'sk' AS "sk", 'cjh' AS "cjh"));
```

	MAJOR	kt	lg	sk	cjh
1	광고홍보학	0	1	0	0
2	문헌정보학	2	0	0	0
3	산업공학과	0	1	0	0
4	경영정보학	1	2	0	0
5	컴퓨터정보	0	0	1	0
6	컴퓨터공학	4	0	1	0
7	경영학	1	0	2	0

문제129. 아래의 결과를 출력하시오.

	MAJOR	24	25	26	27	28	29	32
1	광고홍보학	0	0	1	0	0	0	0
2	문헌정보학	0	0	0	2	0	0	0
3	산업공학과	0	0	0	0	1	0	0
4	경영정보학	1	0	0	1	0	1	0
5	컴퓨터정보	0	1	0	0	0	0	0
6	컴퓨터공학	1	1	3	0	0	0	0
7	경영학	1	1	0	0	1	0	0

SELECT \*

FROM (SELECT major, age FROM EMP2)

pivot (COUNT(\*) FOR age IN('24' AS "24", '25' AS "25", '26' AS "26", '27' AS "27", '28' AS "28", '29' AS "29", '32' AS "32"));

## 5. Data 분석 함수

2018년 4월 18일 수요일    오후 5:13

**DATA 분석 함수** : 기존 함수로는 구현하기 어려운 데이터 분석을 쉽게 도와주는 함수

### 종류

1. Rank -중복, Dense\_rank - 중복 없이
2. Ntile
3. 누적 데이터 출력함수
4. Listagg 함수
5. Lead, lag 함수
6. Pivot
7. unpivot

### 5-1. Rank, Dense\_Rank

#### : 순위를 출력하는 함수

Ex) select ename, sal,

Rank() over (order by sal desc) 순위    <--- over(확장하다) : () 안에있는 내용으로 출력

From emp;

문제130. 이름과 나이와 순위를 출력하는데, 순위가 나이가 높은 학생부터 출력하시오.

SELECT ename, age,

rank() OVER (order BY age desc) 순위

FROM EMP2;

	ENAME	AGE	순위
1	정호진	32	1
2	은해찬	29	2
3	이찬중	28	3
4	김건태	28	3
5	지윤철	28	3
6	김동윤	28	3
7	도웅	28	3
8	이근호	28	3
9	차호성	28	3
10	김영토	28	3
11	김지우	27	11

SELECT ename, age,

Dense\_rank() OVER (order BY age desc) 순위

FROM EMP2;

	ENAME	AGE	순위
1	정호진	32	1
2	은해찬	29	2
3	이찬중	28	3
4	김건태	28	3
5	지윤철	28	3
6	김동윤	28	3
7	도웅	28	3
8	이근호	28	3
9	차호성	28	3
10	김영토	28	3
11	김지우	27	4

문제131. 직업이 SALESMAN 인 직원들의 이름과 월급과 직업과 순위를 출력하는데  
순위가 월급이 높은직원부터 출력하시오.

```
SELECT ename, sal, job,
       RANK() OVER (ORDER BY sal desc) 순위
FROM EMP
WHERE job = 'SALESMAN';
```

	ENAME	SAL	JOB	순위
1	ALLEN	1600	SALESMAN	1
2	TURNER	1500	SALESMAN	2
3	WARD	1250	SALESMAN	3
4	MARTIN	1250	SALESMAN	3

문제132. 월급이 1000에서 3000사이인 직원들의 이름과 월급과 입사일과 순위를 출  
력하는데

순위가 먼저 입사한 사원부터 순위를 부여하시오.

```
SELECT ename, sal, hiredate,
       RANK() OVER(ORDER BY hiredate asc) 순위
FROM EMP
WHERE sal BETWEEN 1000 AND 3000;
```

	ENAME	SAL	HIREDATE	순위
1	ALLEN	1600	1981-02-11 오전 12:00:00	1
2	WARD	1250	1981-02-23 오전 12:00:00	2
3	JONES	2975	1981-04-01 오전 12:00:00	3
4	BLAKE	2850	1981-05-01 오전 12:00:00	4
5	CLARK	2450	1981-05-09 오전 12:00:00	5
6	TURNER	1500	1981-08-21 오전 12:00:00	6

문제133. 월급이 3000인 직원은 직원 테이블에서 월급의 순위가 어떻게 되는지 출력하  
시오.

```
SELECT
       RANK(3000) within group (ORDER BY sal desc) 순위
FROM EMP;
```

	순위
1	2

문제134. 나이가 25살인 학생은 우리반에서 나이의 순위가 어떻게 되는지 출력하시  
오.

```
SELECT
```

RANK(25) within GROUP (ORDER BY age desc) 순위

FROM EMP2;

순위
1
24

문제135. 통신사, 이름, 나이, 순위를 출력하는데, 순위가 **통신사 별로** 각각 나이가 **높은 순서대로**

순위를 출력하시오.

SELECT telecom, ename, age,

RANK() OVER (**PARTITION BY** telecom ORDER BY age DESC) 순위 <-- **partition by :**

**통신사 별로 각각**

FROM EMP2;

	TELECOM	ENAME	AGE	순위
1	cjh	이근호	28	1
2	kt	지윤철	28	1
3	kt	차호성	28	1
4	kt	김영토	28	1
5	kt	송윤호	27	4
6	kt	신현수	27	4
7	kt	김지우	27	4
8	kt	송윤호	27	4
9	kt	백광흠	26	8
10	kt	방승준	26	8
11	kt	이한새	25	10
12	kt	신영근	25	10
13	kt	김근아	24	12
14	kt	장은희	24	12
15	lg	정호진	32	1

문제136. 직업, 이름, 월급, 순위를 출력하는데 순위가 직업별로 각각 월급이 높은 순서대로 순위를

출력하시오.

SELECT job, ename, sal,

RANK() OVER (**PARTITION BY** job ORDER BY sal desc) 순위

FROM EMP;

	JOB	ENAME	SAL	순위
1	ANALYST	SCOTT	3000	1
2	ANALYST	FORD	3000	1
3	CLERK	MILLER	1300	1
4	CLERK	ADAMS	1100	2
5	CLERK	JAMES	950	3
6	CLERK	SMITH	800	4
7	MANAGER	JONES	2975	1
8	MANAGER	BLAKE	2850	2
9	MANAGER	CLARK	2450	3
10	PRESIDENT	KING	5000	1
11	SALESMAN	ALLEN	1600	1
12	SALESMAN	TURNER	1500	2

문제137. 전공, 이름, 나이, 순위를 출력하는데, 순위가 전공별로 각각 나이가 높은 학생부터

순위가 출력되게 하시오.

SELECT major, ename, age,

RANK() OVER (PARTITION BY major ORDER BY age desc) 순위

FROM EMP2;

	MAJOR	ENAME	AGE	순위
1	경영정보학	은해찬	29	1
2	경영정보학	김광록	27	2
3	경영정보학	장은희	24	3
4	경영학	김동윤	28	1
5	경영학	이한새	25	2
6	경영학	한지윤	24	3
7	경제학	차호성	28	1
8	경제학	윤진민	27	2
9	경제학	신현수	27	2
10	광고경영학	지윤철	28	1
11	광고홍보학	이상민	26	1
12	기술융합공	유혜린	26	1
13	도시계획학	김영토	28	1

문제138. 위의 결과에서 순위가 1위인 학생들만 출력하시오. (from 절에 서브쿼리) :

먼저 실행하게끔

SELECT \*

FROM( SELECT major, ename, age,

RANK() OVER (PARTITION BY major ORDER BY age desc) 순위

FROM EMP2)

WHERE 순위 = 1;

	MAJOR	ENAME	AGE	순위
1	경영정보학	은해찬	29	1
2	경영학	김동윤	28	1
3	경제학	차호성	28	1
4	광고경영학	지윤철	28	1
5	광고홍보학	이상민	26	1
6	기술융합공	유혜린	26	1
7	도시계획학	김영토	28	1
8	문헌정보학	송윤희	27	1
9	문헌정보학	송윤희	27	1
10	물리학	정호진	32	1
11	산업공학과	도웅	28	1

문제139. 직업, 이름, 월급, 순위를 출력하는데, 직업별로 각각 월급의 순위가 1위인 사원들만 출력.

SELECT \*

FROM (SELECT job, ename, sal,

RANK() over(PARTITION BY job ORDER BY sal desc) 순위

FROM EMP)

WHERE 순위 = 1;



	JOB	ENAME	SAL	순위
1	ANALYST	FORD	3000	1
2	ANALYST	SCOTT	3000	1
3	CLERK	MILLER	1300	1
4	MANAGER	JONES	2975	1
5	PRESIDENT	KING	5000	1
6	SALESMAN	ALLEN	1600	1

## 5-2. Ntile 분석 함수 (많이 사용함)

Ex) select

ENAME, sal,

NTILE(4) over (order by sal desc) 등급 <-- 월급을 4등급으로 나눠서 결과 출력

From emp;

문제140. 통신사, 이름, 나이, 등급을 출력하는데 등급을 통신사별로 각각 등급을 나누는데 4등급으로 등급을 나누시오.

SELECT telecom, ename, age,

NTILE(4) OVER(PARTITION BY telecom ORDER BY age desc) 등급

FROM EMP2;

	TELECOM	ENAME	AGE	등급
1	cjh	이근호	28	1
2	kt	지윤철	28	1
3	kt	차호성	28	1
4	kt	김영토	28	1
5	kt	송윤희	27	1
6	kt	신현수	27	2
7	kt	김지우	27	2
8	kt	송윤희	27	2
9	kt	백광훈	26	3
10	kt	방승준	26	3

문제141. 위의 결과에서 등급이 1등급인 학생들만 출력하시오.

SELECT \*

FROM (SELECT telecom, ename, age,

NTILE(4) OVER(PARTITION BY telecom ORDER BY age desc) 등급

FROM EMP2)

WHERE 등급 = 1;

	TELECOM	ENAME	AGE	등급
1	cjh	이근호	28	1
2	kt	지윤철	28	1
3	kt	차호성	28	1
4	kt	김영토	28	1
5	kt	송윤희	27	1
6	lg	정호진	32	1
7	lg	은해찬	29	1
8	lg	이찬중	28	1
9	sk	김건태	28	1
10	sk	김동윤	28	1

## 5-3. 누적 데이터 출력하는 분석 함수

## : 데이터를 누적해서 결과 출력 (많이 사용함)

Ex) select empno, ename, sal,

Sum(sal) over (order by empno) 누적치

From emp;

	EMPNO	ENAME	SAL	누적치
1	7369	SMITH	800	800
2	7499	ALLEN	1600	2400
3	7521	WARD	1250	3650
4	7566	JONES	2975	6625
5	7654	MARTIN	1250	7875
6	7698	BLAKE	2850	10725
7	7782	CLARK	2450	13175
8	7788	SCOTT	3000	16175
9	7839	KING	5000	21175
10	7844	TURNER	1500	22675

문제142. 부서번호, 사원번호, 이름, 월급, 월급의 누적치를 출력하는데, 월급의 누적치가 부서번호별로 각

각 월급의 누적치가 출력되게 하시오.

SELECT DEPTno, empno, ename, sal,

SUM(sal) OVER (PARTITION BY deptno ORDER BY sal ) 누적치

FROM EMP;

	DEPTNO	EMPNO	ENAME	SAL	누적치
1	10	7934	MILLER	1300	1300
2	10	7782	CLARK	2450	3750
3	10	7839	KING	5000	8750
4	20	7369	SMITH	800	800
5	20	7876	ADAMS	1100	1900
6	20	7566	JONES	2975	4875
7	20	7788	SCOTT	3000	10875
8	20	7902	FORD	3000	10875
9	30	7900	JAMES	950	950
10	30	7521	WARD	1250	3450
11	30	7654	MARTIN	1250	3450
12	30	7844	TURNER	1500	4950
13	30	7499	ALLEN	1600	6550
14	30	7698	BLAKE	2850	9400

## 5-4. Listagg 함수

### : 데이터를 가로로 출력하는 함수

Ex) select deptno,

Listagg(ename, ',') within group(order by ename asc) ',' : ,로 구분

From emp

Group by deptno;

	DEPTNO	LISTAGG(ENAME,',')WITHINGROUP(ORDERBYENAMI
1	10	CLARK,KING,MILLER
2	20	ADAMS,FORD,JONES,SCOTT,SMITH
3	30	ALLEN,BLAKE,JAMES,MARTIN,TURNER,WARD

문제143. 통신사, 이름을 출력하는데, 통신사 별로 각각 이름이 가로로 출력되게 하시오.

```
SELECT telecom,
       listagg(ename, ',') within GROUP(ORDER BY ename asc)
FROM EMP2
GROUP BY telecom;
```

	TELECOM	LISTAGG(ENAME,',')WITHINGROUP(ORDERBYENAMI
1	cjh	이근호
2	kt	김근아,김영토,김지우,방승준,백광훈,송윤호,송
3	lg	김광록,도웅,유혜린,은해찬,이광훈,이상민,이천
4	sk	김건태,김대경,김동윤,김원섭,윤동환,윤진민,이

**문제144. 아래와 같이 결과를 출력하시오.(중요)**

```
SELECT telecom,
       listagg(ename || '(' || age || ')', ',') within GROUP(ORDER BY age desc)
FROM EMP2
GROUP BY telecom;
```

	TELECOM	LISTAGG(ENAME  '('  AGE  ')',',')WITHINGROUP(ORI
1	cjh	이근호(28)
2	kt	김영토(28),지윤철(28),차호성(28),김지우(27),송
3	lg	정호진(32),은해찬(29),도웅(28),이찬중(28),김광
4	sk	김건태(28),김동윤(28),김대경(27),윤진민(27),김

## 5-5. Lead, lag 함수

**: 바로 전행 또는 바로 다음행을 다음라인에 출력하는 함수.**

Ex) select empno, ename, sal,

Lead(sal, 1) over (order by empno) lead\_sal,

1 = 그다음행, 2 = 다다음행

Lag(sal, 1) over (order by empno) lag\_sal

1 = 그전행, 2 = 전전행

From emp;

	EMPNO	ENAME	SAL	LEAC_SAL	LAG_SAL
1	7369	SMITH	800	1600	(null)
2	7499	ALLEN	1600	1250	800
3	7521	WARD	1250	2975	1600
4	7566	JONES	2975	1250	1250
5	7654	MARTIN	1250	2850	2975
6	7698	BLAKE	2850	2450	1250
7	7782	CLARK	2450	3000	2850

**문제145. 이름, 나이 나의 바로 전 나이, 나의 바로 다음나이를 출력하시오.**

```
SELECT ename, age,
       LAG(age, 1) OVER (ORDER BY age desc) lag_age,
       LEAD(age, 1) OVER (ORDER BY age desc) lead_age
FROM EMP2;
```

	ENAME	AGE	LAG_AGE	LEAD_AGE
1	정호진	32	(null)	29
2	은해찬	29	32	28
3	이찬중	28	29	28
4	김건태	28	28	28
5	지윤철	28	28	28
6	김동윤	28	28	28
7	도웅	28	28	28
8	이근호	28	28	28
9	차호성	28	28	28
10	김영토	28	28	27
11	김지우	27	28	27

문제146. 부서번호, 사원번호, 이름, 월급을 출력하고 그 옆에 부서번호별로 각각 바로 그 전 사원의 월급이 출력되게 하시오.

```
SELECT deptno, empno, ename, sal,
       lag(sal, 1) OVER (PARTITION BY deptno ORDER BY deptno asc) lag_sal
FROM EMP;
```

	DEPTNO	EMPNO	ENAME	SAL	LAG_SAL
1	10	7839	KING	5000	(null)
2	10	7782	CLARK	2450	5000
3	10	7934	MILLER	1300	2450
4	20	7876	ADAMS	1100	(null)
5	20	7788	SCOTT	3000	1100
6	20	7369	SMITH	800	3000
7	20	7902	FORD	3000	800
8	20	7566	JONES	2975	3000
9	30	7521	WARD	1250	(null)

## 5-6. Pivot

: 세로로 출력되는 결과를 가로로 출력하는 함수 (로우 -> 컬럼)

```
Ex) select *
      From (select deptno, sal from emp)
      Pivot ( sum(sal) for deptno in (10,20,30))
```

	10	20	30
1	8750	10875	9400

## 5-7- Unpivot

: 가로를 세로로 출력하는 함수 (컬럼 -> 로우)  
(엑셀로 데이터를 받아야할 경우)

	ENAME	BICYCLE	CAMERA	NOTEBOOK
1	SMITH	2	3	1
2	ALLEN	1	2	3
3	KING	3	2	2

```
Ex) select *
      From order2
      Unpivot (cnt for item in ( BICYCLE, CAMERA, NOTEBOOK)); cnt ,item = 아무단어
```

	ENAME	ITEM	CNT
1	SMITH	BICYCLE	2
2	SMITH	CAMERA	3
3	SMITH	NOTEBOOK	1
4	ALLEN	BICYCLE	1
5	ALLEN	CAMERA	2
6	ALLEN	NOTEBOOK	3
7	KING	BICYCLE	3
8	KING	CAMERA	2
9	KING	NOTEBOOK	2

문제147. 살인이 가장 많이 일어나는 장소가 어디인지 분석하시오.

구	CRIME_TYPE	아파트	집	고속도로	노상	상점	시장노점	숙박업소	병원	사무실	공장	공사장	창고	역대합실	지하철	교통	유흥접객업소	유원지	학교	금융기관	의료
1	절도	25389	37787	151	62560	29977	1239	9203	16053	8416	3540	4003	2329	881	737	2249	631	1532	3203	5888	
2	장물	88	130	1	483	1188	21	11	39	271	177	41	57	8	8	13	5	6	15	26	
3	손괴	3122	3172	32	22193	2054	142	907	3539	1183	158	217	67	100	41	101	45	115	239	176	
4	살인	242	312	1	280	23	5	43	87	40	15	4	4	8	1	9	1	13	8	0	

```

SELECT *
FROM(SELECT crime_type, c_loc, cnt
from crime_loc
unpivot(cnt for c_loc in( 아파트, 집, 고속도로, 노상, 상점, 시장노점, 숙박업소,
                        병원, 사무실, 공장, 공사장, 창고, 역대합실, 지하철, 교
                        통,
                        유흥접객업소, 유원지, 학교, 금융기관, 의료기관, 종교기관,
                        산야,해상, 부대, 구금장소, 공지, 기타 )) )
WHERE crime_type = '살인'
ORDER BY cnt DESC;

```

1. From 으로 데이터를 테이블화 시켜서
2. Where로 조건을 설정함

문제148. 절도가 많이 일어나는 장소가 어디인지 장소와 건수와 순위가 출력되게하시오.

```

SELECT crime_type, c_loc AS "장소", cnt,
       RANK() OVER(ORDER BY cnt desc) 순위
from crime_loc
unpivot(cnt for c_loc in( 아파트, 집, 고속도로, 노상, 상점, 시장노점, 숙박업소,
                        병원, 사무실, 공장, 공사장, 창고, 역대합실, 지하철, 교
                        통,
                        유흥접객업소, 유원지, 학교, 금융기관, 의료기관, 종교기관,
                        산야,해상, 부대, 구금장소, 공지, 기타 ))
WHERE crime_type = '절도'
ORDER BY cnt DESC;

```

	CRIME_TYPE	장소	CNT	순위
1	절도	노상	62560	1
2	절도	기타	44333	2
3	절도	집	37787	3
4	절도	상점	29977	4
5	절도	아파트	25389	5
6	절도	병원	16053	6
7	절도	숙박업소	9203	7
8	절도	사무실	8416	8
9	절도	금융기관	5888	9
10	절도	공사장	4003	10

문제149. 살인을 일으키는 가장 큰 원인이 무엇인지 범죄 원인, 건수, 순위를 출력하시오.

SELECT crime\_cause, cnt,

RANK() OVER(ORDER BY cnt desc) 순위

FROM crime\_cause

unpivot(cnt FOR crime\_cause IN(생계형, 유흥, 도박,허영심,복수,해고,징벌,가정불화,호기심,유혹,

사고,불만,부주의,기타))

WHERE crime\_type = '살인';

	CRIME_CAUSE	CNT	순위
1	사고	147	1
2	기타	118	2
3	가정불화	51	3
4	불만	15	4
5	유흥	6	5
6	복수	5	6

%% 이 문장을 테이블로 만들어서 하면 더편함

Create table crime\_cause2 as SELECT 문장~~

하고 select cause, con, rank() over (order by cnt desc)

select \* from crime\_cause2

Where crime\_type = '살인'; 을 쓰면 똑같은 결과가 나옴

## 6. 분석함수 문제

2018년 4월 18일 수요일    오후 5:30

문제150. 가정불화로 인해서 생기는 범죄 유형을 출력하시오.

```
SELECT *  
FROM crime_cause2  
WHERE cause = '가정불화';
```

	CRIME_TYPE	CAUSE	CNT
1	살인	가정불화	51
2	살인미수	가정불화	44
3	강도	가정불화	15
4	강간강제추행	가정불화	30

문제151. 가정불화로 인해서 생기는 범죄 유형, 건수와 순위를 출력하시오.

```
SELECT crime_type, cnt,  
       RANK() OVER (ORDER BY cnt desc) 순위  
FROM crime_cause2  
WHERE cause = '가정불화';
```

	CRIME_TYPE	CNT	순위
1	폭행	1724	1
2	상해	1666	2
3	폭력행위등	937	3
4	기타	455	4
5	손괴	346	5
6	교통범죄	174	6
7	협박	115	7

문제152. 방화를 일으키는 가장 큰 요인은 무엇인가?

```
SELECT cause, cnt,  
       RANK() OVER (ORDER BY cnt desc)  
from crime_cause2  
WHERE crime_type = '방화';
```

	CAUSE	CNT	RANK()OVER(ORDERBYCNTDESC)
1	사고	547	1
2	기타	339	2
3	불만	124	3
4	가정불화	97	4
5	호기심	62	5
6	부주의	40	6
7	생계형	6	7

위의 결과에서 순위가 1위인 것만 출력하시오.

```
SELECT *  
FROM(SELECT cause, cnt,  
       RANK() OVER (ORDER BY cnt desc) 순위  
from crime_cause2  
WHERE crime_type = '방화')  
WHERE 순위 = 1;
```

	CAUSE	CNT	순위
1	사고	547	1

%% 서울시 물가 데이터를 오라클에 입력

문제153. price 테이블의 전체 몇건인가?

```
SELECT COUNT(*)
FROM price;
```

	COUNT(*)
1	9517

```
Select a_name, a_price, a_unit, m_name
From price
Order by a_price desc;
```

문제154. 사과를 파는 판매장과 가격을 출력하는데 가격이 높은 곳 부터 출력하시오.

```
SELECT m_name, a_price
FROM price
WHERE a_name LIKE '%사과%'
ORDER BY a_price DESC;
```

	M_NAME	A_PRICE
1	롯데백화점 관악점	10000
2	롯데백화점 강남점	6400
3	롯데백화점 강남점	6400
4	롯데백화점 강남점	6400
5	현대백화점 신촌점	4750
6	이마트 용산점	4600
7	롯데백화점 청량리	4000
8	롯데백화점 청량리	4000
9	현대백화점 신촌점	3900
10	현대백화점 신촌점	3900
11	농협 하나로마트 용	3834
12	농협 하나로마트 용	3670

문제155. 위의 결과를 다시 출력하는데 가격의 순위도 같이 출력하시오.

```
SELECT m_name, a_price,
       Dense_RANK()OVER(ORDER BY a_price desc)순위
FROM price
WHERE a_name LIKE '%사과%'
ORDER BY a_price DESC;
```

	M_NAME	A_PRICE	순위
1	롯데백화점 관악점	10000	1
2	롯데백화점 강남점	6400	2
3	롯데백화점 강남점	6400	2
4	롯데백화점 강남점	6400	2
5	현대백화점 신촌점	4750	3
6	이마트 용산점	4600	4
7	롯데백화점 청량리	4000	5
8	롯데백화점 청량리	4000	5
9	현대백화점 신촌점	3900	6
10	현대백화점 신촌점	3900	6



문제156. 위의 결과에서 순위가 1부터 10까지 출력하시오.

```
SELECT *
FROM(SELECT m_name, a_price,
          DENSE_RANK()OVER(ORDER BY a_price desc)순위
FROM price
WHERE a_name LIKE '%사과%'
ORDER BY a_price DESC)
WHERE 순위 BETWEEN 1 AND 10;
```

	M_NAME	A_PRICE	순위
1	롯데백화점 관악점	10000	1
2	롯데백화점 강남점	6400	2
3	롯데백화점 강남점	6400	2
4	롯데백화점 강남점	6400	2
5	현대백화점 신촌점	4750	3
6	이마트 용산점	4600	4
7	롯데백화점 청량리	4000	5
8	롯데백화점 청량리	4000	5
9	현대백화점 신촌점	3900	6
10	현대백화점 신촌점	3900	6
11	농협 하나로마트 용	3834	7
12	농협 하나로마트 용	3670	8
13	이마트 창동점	3633	9
14	현대백화점 미아점	3600	10
15	현대백화점 미아점	3600	10
16	현대백화점 미아점	3600	10

%% 엑셀 데이터를 database에 테이블로 만드는 방법

도구 -> 데이터 가져오기

문제157. 2012년에 가장 자살을 많이한 나라명을 출력하시오.

```
SELECT *
FROM(
  SELECT cname,
        DENSE_RANK()OVER(ORDER BY Y2012 desc) 순위
  FROM suicide
  WHERE cname IS NOT NULL
)
WHERE 순위 = '1';
```

	CNAME	순위
1	한국	1

또는

```
SELECT *
FROM(
  SELECT cname,
        DENSE_RANK()OVER(ORDER BY nvl(Y2012,0) desc) 순위
  FROM suicide
)
WHERE 순위 = '1';
```

문제158. 직업, 이름, 입사일 순위를 출력하는데 순위가 직업별로 각각 먼저 입사한 직원순으

로 순위를 출력하시오.

```
SELECT job, ename, hiredate,  
       RANK() over(partition BY job ORDER BY hiredate asc) 순위  
FROM EMP;
```

	JOB	ENAME	HIREDATE	순위
1	ANALYST	FORD	1981-12-11 오전 12:00:00	1
2	ANALYST	SCOTT	1982-12-22 오전 12:00:00	2
3	CLERK	SMITH	1980-12-09 오전 12:00:00	1
4	CLERK	JAMES	1981-12-11 오전 12:00:00	2
5	CLERK	MILLER	1982-01-11 오전 12:00:00	3
6	CLERK	ADAMS	1983-01-15 오전 12:00:00	4
7	MANAGER	JONES	1981-04-01 오전 12:00:00	1
8	MANAGER	BLAKE	1981-05-01 오전 12:00:00	2
9	MANAGER	CLARK	1981-05-09 오전 12:00:00	3

문제159. 위의 결과에서 1등만 출력하시오.

```
SELECT *  
FROM(SELECT job, ename, hiredate,  
       RANK() over(partition BY job ORDER BY hiredate asc) 순위  
FROM EMP)  
WHERE 순위 = 1;
```

## 7. 조인

2018년 4월 3일 화요일 오전 9:47

### 조인 : 여러 개의 테이블의 데이터를 하나의 결과로 모아서 출력하는 SQL 문법.

Ex) 부서 테이블 전체를 조회하시오.

```
Select * from dept;
```

- 이름과 부서위치를 출력하시오.

```
SELECT ename, LOC  
FROM emp, DEPT
```

WHERE emp.deptno = dept.deptno; <-- 검색조건x, 조인 조건o (연관 증거) 양쪽 테이블에 서로 존재하는 컬럼

	ENAME	LOC
1	KING	NEW YORK
2	BLAKE	CHICAGO
3	CLARK	NEW YORK
4	JONES	DALLAS
5	MARTIN	CHICAGO
6	ALLEN	CHICAGO
7	TURNER	CHICAGO

- 조인 조건이 없으면 조건비교할때 하나하나 전부 대입해서 출력됨.

	ENAME	LOC
1	KING	NEW YORK
2	BLAKE	NEW YORK
3	CLARK	NEW YORK
4	JONES	NEW YORK
5	MARTIN	NEW YORK
6	ALLEN	NEW YORK
7	TURNER	NEW YORK
8	JAMES	NEW YORK
9	WARD	NEW YORK
10	FORD	NEW YORK
11	SMITH	NEW YORK
12	SCOTT	NEW YORK
13	ADAMS	NEW YORK
14	MILLER	NEW YORK
15	KING	DALLAS

### 조인 종류

#### 1. 오라클 조인문법

- EQUI 조인
- NON EQU 조인
- OUTER 조인 LEFT OUTER
- OUTER 조인 RIGHT OUTER
- SELF 조인 SELF OUTER

#### 2. 1999 ANSI 조인문법

- OUTER 조인 FULL OUTER

- **EQUI 조인** : 조인 하려는 테이블의 조인 조건의 연결고리가 이퀄 인 조인 문법

EX) select ename, loc  
 From emp, dept  
 Where emp.deptno = dept.deptno;

**문제160. 직원번호, 이름, 월급, 부서위치를 출력하시오.**

SELECT empno, ename, sal, loc  
 FROM emp, DEPT  
 WHERE emp.deptno = dept.deptno;

	EMPNO	ENAME	SAL	LOC
1	7839	KING	5000	NEW YORK
2	7698	BLAKE	2850	CHICAGO
3	7782	CLARK	2450	NEW YORK
4	7566	JONES	2975	DALLAS
5	7654	MARTIN	1250	CHICAGO
6	7499	ALLEN	1600	CHICAGO
7	7844	TURNER	1500	CHICAGO
8	7900	JAMES	950	CHICAGO
9	7521	WARD	1250	CHICAGO

**문제161. 위의 결과에서 부서번호도 출력하시오.**

SELECT empno, ename, sal, loc, emp.deptno  
 WHERE emp.deptno = dept.deptno;

<-- deptno만쓰면 emp인지 dept인지 어디서 가져오라는지 명확하지 않아서 오류가나옴.

	EMPNO	ENAME	SAL	LOC	DEPTNO
1	7839	KING	5000	NEW YORK	10
2	7698	BLAKE	2850	CHICAGO	30
3	7782	CLARK	2450	NEW YORK	10
4	7566	JONES	2975	DALLAS	20
5	7654	MARTIN	1250	CHICAGO	30

Select emp.empno, emp.ename, emp.sal, dept.loc, emp.deptno  
 From emp, dept  
 Where emp.deptno = dept.deptno;

=> 앞에다가 다 써주면 성능이 좋아짐

이렇게 하면 코딩이 길어지는데 ,

Select e.empno, e.ename, e.sal, d.loc, e.deptno  
 From emp e, dept d  
 Where e.deptno = d.deptno; 로 쓰면된다. <== 테이블 별칭

**문제162. 월급이 3000이상인 직원들의 이름과 월급과 부서위치를 출력하시오.**

SELECT e.ename, e.sal, d.loc  
 FROM EMP e, DEPT d  
 WHERE e.deptno = d.deptno      조인조건  
 And e.sal >= 3000;      검색조건

	ENAME	SAL	LOC
1	KING	5000	NEW YORK
2	SCOTT	3000	DALLAS
3	FORD	3000	DALLAS

**문제163. 이름의 첫글자가 S로 시작하는 직원들의 이름과 부서위치를 출력하시오.**

SELECT e.ename, d.loc  
 FROM EMP e, DEPT d  
 WHERE e.DEPTNO = d.DEPTNO  
 AND e.ename LIKE 'S%';

	ENAME	LOC
1	SCOTT	DALLAS

	ENAME	LOC
1	SCOTT	DALLAS
2	SMITH	DALLAS

문제164. 월급이 1000에서 3000인 직원들의 이름과 월급과 부서위치, 부서명을 출력하시오.

```
SELECT e.ename, e.sal, d.loc, d.dname
FROM EMP e, DEPT d
WHERE e.DEPTNO = d.DEPTNO
AND sal BETWEEN 1000 AND 3000;
```

	ENAME	SAL	LOC	DNAME
1	BLAKE	2850	CHICAGO	SALES
2	CLARK	2450	NEW YORK	ACCOUNTING
3	JONES	2975	DALLAS	RESEARCH
4	MARTIN	1250	CHICAGO	SALES
5	ALLEN	1600	CHICAGO	SALES
6	TURNER	1500	CHICAGO	SALES
7	WARD	1250	CHICAGO	SALES
8	FORD	3000	DALLAS	RESEARCH
9	SCOTT	3000	DALLAS	RESEARCH
10	ADAMS	1100	DALLAS	RESEARCH
11	MILLER	1300	NEW YORK	ACCOUNTING

문제165. DALLAS 에서 근무하는 직원들의 이름과 부서위치를 출력하시오.

```
SELECT e.ename, d.loc
FROM EMP e, DEPT d
WHERE e.DEPTNO = d.DEPTNO
AND d.loc = 'DALLAS';
```

	ENAME	LOC
1	JONES	DALLAS
2	FORD	DALLAS
3	SMITH	DALLAS
4	SCOTT	DALLAS
5	ADAMS	DALLAS

문제166. 아래와 같이 결과를 출력하시오.

	LOC	LISTAGG(ENAME,',')WITHINGROUP(ORDERBYE.ENA
1	CHICAGO	ALLEN,BLAKE,JAMES,MARTIN,TURNER,WARD
2	DALLAS	ADAMS,FORD,JONES,SCOTT,SMITH
3	NEW YORK	CLARK,KING,MILLER

```
SELECT d.loc,
listagg(ename, ',') within group(ORDER BY e.ename asc)
FROM EMP e, dept d
WHERE e.DEPTNO = d.deptno
GROUP BY d.loc;
```

문제167. 부서위치, 이름, 월급, 순위를 출력하는데 순위가 부서위치별로 각각 월급이 높은 순서대로 순위를 출력하시오.

```
SELECT d.loc, e.ename, e.sal,
dense_RANK() OVER (PARTITION BY d.loc orDER BY e.sal desc) 순위
FROM EMP e, DEPT d
WHERE e.DEPTNO = d.DEPTNO;
```

	LOC	ENAME	SAL	순위
1	CHICAGO	BLAKE	2850	1
2	CHICAGO	ALLEN	1600	2
3	CHICAGO	TURNER	1500	3

	LOC	ENAME	SAL	순위
1	CHICAGO	BLAKE	2850	1
2	CHICAGO	ALLEN	1600	2
3	CHICAGO	TURNER	1500	3
4	CHICAGO	WARD	1250	4
5	CHICAGO	MARTIN	1250	4
6	CHICAGO	JAMES	950	5
7	DALLAS	FORD	3000	1
8	DALLAS	SCOTT	3000	1
9	DALLAS	JONES	2975	2
10	DALLAS	ADAMS	1100	3
11	DALLAS	SMITH	800	4
12	NEW YORK	KING	5000	1
13	NEW YORK	CLARK	2450	2
14	NEW YORK	MILLER	1300	3

**Non equi join : 조인하려는 테이블 사이의 연결 고리가 이퀄 조건이 아닌 경우의 조인 문법.**

Ex) 급여 등급 테이블을 조회하시오.

Select \* from salgrade;

	GRADE	LOSAL	HISAL
1	1	700	1200
2	2	1201	1400
3	3	1401	2000
4	4	2001	3000
5	5	3001	9999

700~1200 은 1등급 1201~1400은 2등급 ....

-이름과 등급을 출력하시오.

```
SELECT e.ename, s.grade
FROM EMP e, salgrade s
WHERE e.sal BETWEEN s.losal and s.hisal;
```

	ENAME	GRADE
1	JAMES	1
2	SMITH	1
3	ADAMS	1
4	MARTIN	2
5	WARD	2
6	MILLER	2
7	ALLEN	3
8	TURNER	3
9	BLAKE	4
10	CLARK	4
11	JONES	4
12	FORD	4
13	SCOTT	4
14	KING	5

%% 오라클이 내부적으로 수행한 조인방법 : sort merge join

%% 오라클 SQL문을 실행하는 데몬 프로세서 : 옵티마이저

SQL문의 실행계획을 만들고 실행한다. (악성 SQL을 강화)

문제168. 위의 결과를 다시 출력하는데 등급이 5등급인 사원만 출력하시오.

```
SELECT e.ename, s.grade
      FROM EMP e, salgrade s
     WHERE e.sal BETWEEN s.losal and s.hisal AND s.grade = 5;
```

	ENAME	GRADE
1	KING	5

문제169. 아래와 같이 결과를 출력하시오.

	GRADE	LISTAGG(E.ENAME,',')WITHINGROUP(ORDERBYE.EN
1	1	ADAMS,JAMES,SMITH
2	2	MARTIN,MILLER,WARD
3	3	ALLEN,TURNER
4	4	BLAKE,CLARK,FORD,JONES,SCOTT
5	5	KING

```
SELECT s.grade,
       listagg(e.ename,',') within GROUP(ORDER BY e.ename asc)
  FROM EMP e, salgrade s
 WHERE e.sal BETWEEN s.losal AND s.hisal
 GROUP BY s.grade;
```

문제170. 아래와 같이 결과를 출력하시오.

	GRADE	LISTAGG(E.ENAME  '('  E.SAL  ')',')WITHINGROUP(
1	1	ADAMS(1100),JAMES(950),SMITH(800)
2	2	MARTIN(1250),MILLER(1300),WARD(1250)
3	3	ALLEN(1600),TURNER(1500)
4	4	BLAKE(2850),CLARK(2450),FORD(3000),JONES(297
5	5	KING(5000)

```
SELECT s.grade,
       listagg(e.ename || '(' || e.sal || ')',') within GROUP(ORDER BY e.ename asc)
  FROM EMP e, salgrade s
 WHERE e.sal BETWEEN s.losal AND s.hisal
 GROUP BY s.grade;
```

문제171. 아래와 같이 결과를 출력하시오.

	GRADE	LISTAGG(E.ENAME  '('  LTRIM(TO_CHAR(E.SAL,
1	1	ADAMS(1,100), JAMES(950), SMITH(800)
2	2	MARTIN(1,250), MILLER(1,300), WARD(1,250)
3	3	ALLEN(1,600), TURNER(1,500)
4	4	BLAKE(2,850), CLARK(2,450), FORD(3,000), JONES
5	5	KING(5,000)

```
SELECT s.grade,
       listagg(e.ename || '(' || LTRIM(TO_CHAR(e.sal,'999,999')) || ')',') within GROUP(ORDER BY e.ename asc)
  FROM EMP e, salgrade s
 WHERE e.sal BETWEEN s.losal AND s.hisal
 GROUP BY s.grade;
```

문제172. 이름과 월급과 등급을 출력하시오.

```
SELECT e.ename, e.sal, s.grade
      FROM EMP e, salgrade s
     WHERE e.sal BETWEEN s.losal AND s.hisal;
```

문제173. 아래와 같이 결과를 출력하시오.

	ENAME	SAL	GRADE
1	JAMES	950	F
2	SMITH	800	F
3	ADAMS	1100	F
4	MARTIN	1250	D
5	WARD	1250	D
6	MILLER	1300	D
7	ALLEN	1600	C
8	TURNER	1500	C
9	BLAKE	2850	B
10	CLARK	2450	B
11	JONES	2975	B
12	FORD	3000	B
13	SCOTT	3000	B
14	KING	5000	A

```
SELECT e.ename, e.sal, DECODE(s.grade, 1,'F',
                                2,'D',
                                3,'C',
                                4,'B',
                                5,'A') grade
FROM EMP e, salgrade s
WHERE e.sal BETWEEN s.losal AND s.hisal;
```

문제174. 위의 결과를 아래와 같이 출력하시오.

	grade	LISTAGG(E.ENAME,',')WITHINGROUP(ORDERBYE.EN
1	A	KING
2	B	BLAKE,CLARK,FORD,JONES,SCOTT
3	C	ALLEN,TURNER
4	D	MARTIN,MILLER,WARD
5	F	ADAMS,JAMES,SMITH

```
SELECT DECODE(s.grade, 1,'F',
                                2,'D',
                                3,'C',
                                4,'B',
                                5,'A') AS "grade",
listagg(e.ename,',')within GROUP(ORDER BY e.ename asc)
FROM EMP e, salgrade s
WHERE e.sal BETWEEN s.losal AND s.hisal
GROUP BY s.grade
ORDER BY grade desc;
```

**Outer join : equi join**으로는 볼 수 없는 결과를 볼때 사용하는 조인,  
**equi join**으로 조인 안된 데이터를 볼때 사용하는 문법.

Ex) 부서 테이블을 조회 하시오.

Select \* from dept;

	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	CHICAGO
4	40	OPERATIONS	BOSTON

이름과 부서위치를 출력하시오.

```
SELECT e.ename, d.LOC
```



```
FROM EMP e, DEPT d
WHERE e.DEPTNO = d.DEPTNO;
```

	ENAME	LOC
1	KING	NEW YORK
2	BLAKE	CHICAGO
3	CLARK	NEW YORK
4	JONES	DALLAS
5	MARTIN	CHICAGO
6	ALLEN	CHICAGO
7	TURNER	CHICAGO
8	JAMES	CHICAGO
9	WARD	CHICAGO
10	FORD	DALLAS
11	SMITH	DALLAS
12	SCOTT	DALLAS
13	ADAMS	DALLAS
14	MILLER	NEW YORK

%% 부서 테이블에 존재하는데 안나온 부서위치

- 위의 결과에서 보스턴도 출력하게하려면

```
SELECT e.ename, d.LOC
FROM EMP e, DEPT d
WHERE e.DEPTNO (+) = d.DEPTNO--(이쪽에도 사용가능);    (+) : outer join sign
```

13	ADAMS	DALLAS
14	MILLER	NEW YORK
15	(null)	BOSTON

문제175. 아래의 데이터를 입력하고 사원이름은 부서위치가 null 인 결과를 출력되게 하시오.

(사원 이름과 부서위치를 출력 하시오.)

```
INSERT INTO emp(empno, ename, sal, deptno)
VALUES(2034, 'JACK', 3400, 70);
```

```
SELECT e.ename, d.loc
FROM EMP e, DEPT d
WHERE e.deptno = d.DEPTNO (+);
```

13	MARTIN	CHICAGO
14	BLAKE	CHICAGO
15	JACK	(null)

**Full outer join(현업에서 많이 사용함 ★★★★★)**

문제176. 아래와 같이 출력하시오.

	ENAME	LOC
1	KING	NEW YORK
2	BLAKE	CHICAGO
3	CLARK	NEW YORK
4	JONES	DALLAS
5	MARTIN	CHICAGO
6	ALLEN	CHICAGO
7	TURNER	CHICAGO
8	JAMES	CHICAGO
9	WARD	CHICAGO
10	FORD	DALLAS
11	SMITH	DALLAS
12	SCOTT	DALLAS
13	ADAMS	DALLAS
14	MILLER	NEW YORK
15	JACK	(null)
16	(null)	BOSTON

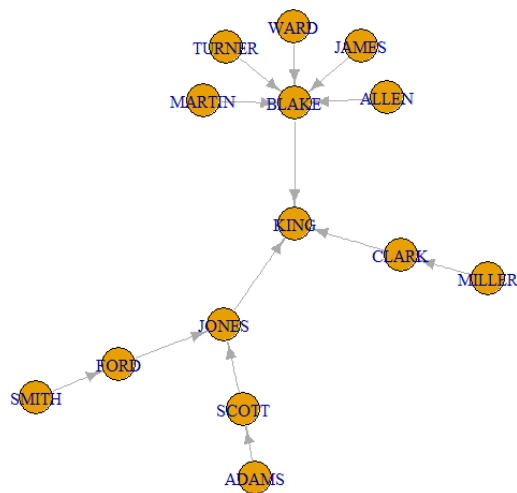
```
SELECT e.ename, d.loc
  FROM EMP e FULL outer join DEPT d
    ON e.deptno = d.DEPTNO;
```

## Self join : 자기 자신의 테이블과 조인하는 문법

Ex) 사원번호, 이름, 관리자의 사원번호를 출력

```
Select empno, ename, mgr
  From emp;
```

- Emp 테이블 mgr 시각화



문제177. 사원이름, 관리자의 이름(직속상사)을 출력하시오.

```
SELECT 사원.ename, 관리자.ename
  FROM EMP 사원, EMP 관리자
 WHERE 사원.mgr = 관리자.empno;
```

	ENAME	ENAME_1
1	JONES	KING
2	CLARK	KING
3	BLAKE	KING
4	WARD	BLAKE
5	JAMES	BLAKE
6	TURNER	BLAKE
7	ALLEN	BLAKE
8	MARTIN	BLAKE
9	MILLER	CLARK
10	SCOTT	JONES
11	FORD	JONES
12	SMITH	FORD
13	ADAMS	SCOTT

문제178. 사원이름쪽에 king도 나오게 하시오

```
SELECT 사원.ename, 관리자.ename
      FROM EMP 사원, EMP 관리자
     WHERE 사원.mgr = 관리자.empno (+);
```

11	FORD	JONES
12	SMITH	FORD
13	ADAMS	SCOTT
14	KING	(null)

문제179. 아래와 같이 결과를 출력하시오.

	ENAME	LISTAGG(사원.ENAME,',')WITHINGROUP(ORDERBY
1	BLAKE	ALLEN, JAMES, MARTIN, TURNER, WARD
2	CLARK	MILLER
3	FORD	SMITH
4	JONES	FORD, SCOTT
5	KING	BLAKE, CLARK, JONES
6	SCOTT	ADAMS

```
SELECT 관리자.ename, listagg(사원.ename,',')within GROUP(ORDER BY 사원.ename asc)
      FROM EMP 사원, EMP 관리자
     WHERE 사원.mgr = 관리자.empno
     GROUP BY 관리자.ename;
```

문제180. 부서위치, 월급, 순위를 출력하는데 순위가 부서위치별로 각각 월급이 높은 사원순으로 순위를 부여 하시오.

```
SELECT d.loc, e.ename, e.sal,
       dense_RANK() OVER(PARTITION by d.loc ORDER BY sal desc) 순위
      FROM EMP e, DEPT d
     WHERE e.DEPTNO = d.DEPTNO;
```

	LOC	ENAME	SAL	순위
1	CHICAGO	BLAKE	2850	1
2	CHICAGO	ALLEN	1600	2
3	CHICAGO	TURNER	1500	3
4	CHICAGO	WARD	1250	4
5	CHICAGO	MARTIN	1250	4
6	CHICAGO	JAMES	950	5
7	DALLAS	FORD	3000	1
8	DALLAS	SCOTT	3000	1
9	DALLAS	JONES	2975	2
10	DALLAS	ADAMS	1100	3
11	DALLAS	SMITH	800	4
12	NEW YORK	KING	5000	1
13	NEW YORK	CLARK	2450	2
14	NEW YORK	MILLER	1300	3

문제181. 통신사 테이블을 생성한다.

```
CREATE TABLE telecom_price
  (telecom VARCHAR2(10),
   t_price number(10),
   t_profit NUMBER(10) );
```

```
INSERT INTO telecom_price
VALUES('sk', 60000, 0);
```

```
INSERT INTO telecom_price
VALUES('lg', 65000, 3);
```

```
INSERT INTO telecom_price
VALUES('kt', 64000, 2);
```

문제182. 학생이름, 나이, 통신사, 통신사 이용요금을 출력하시오.

```
SELECT e2.ename, e2.age, e2.telecom, tp.t_price
FROM EMP2 e2, telecom_price tp
WHERE e2.TELECOM = tp.TELECOM;
```

	ENAME	AGE	TELECOM	T_PRICE
1	김원섭	26	sk	60000
2	김광록	27	lg	65000
3	유혜린	26	lg	65000
4	이유진	25	sk	60000
5	윤진민	27	sk	60000
6	백광훈	26	kt	64000
7	장은희	24	kt	64000
8	김영토	28	kt	64000
9	송윤호	27	kt	64000
10	차호성	28	kt	64000

문제183. 부서위치, 부서위치별 토탈월급을 출력하시오.

```
SELECT d.loc, SUM(e.sal)
FROM EMP e, DEPT d
WHERE e.DEPTNO = d.DEPTNO
GROUP BY d.loc;
```

	LOC	SUM(E.SAL)
1	NEW YORK	8750
2	CHICAGO	9400
3	DALLAS	10875

문제184. 통신사, 통신사별 토탈 금액을 출력하시오.

```
SELECT e2.telecom, SUM(tp.t_price)
FROM EMP2 e2, telecom_price tp
WHERE e2.telecom = tp.TELECOM
GROUP BY e2.telecom;
```

	TELECOM	SUM(TP.T_PRICE)
1	sk	480000
2	lg	585000
3	kt	832000

문제185. 이름, 나이, 통신사, 통신사 월납정액을 출력하는데 CJH 를 쓰는 이근호 학생이 출력되세 하시오.

```
SELECT e2.ename, e2.age, e2.telecom, tp.t_price
FROM EMP2 e2, telecom_price tp
WHERE e2.TELECOM = tp.TELECOM (+);
```

21	송윤희	27	kt	64000
22	신현수	27	kt	64000
23	신영근	25	kt	64000
24	미한새	25	kt	64000
25	방송준	26	kt	64000
26	차호성	28	kt	64000
27	송윤희	27	kt	64000
28	김영토	28	kt	64000
29	장은희	24	kt	64000
30	백광훈	26	kt	64000
31	이근호	28	cjh	(null)

Right/left outer join : 오라클 조인 문법의 outer join을 1999 ansi 문법으로 나타낸 SQL

Ex) 오라클 조인문법(where절에 조인조건, 검색조건 둘다씀)

```
select e.name, d.loc
  From emp e, dept d
  Where e.deptno (+) = d.deptno;          where절은 조인조건
```

1999 ansi 조인 문법 (on절에 조인조건, where절에는 검색조건)

```
Select e.ename, d.loc
  From emp e right outer join dept d
  On (e.deptno = d.deptno);              on조인조건
```

문제186. 아래의 오라클 조인 문법을 1999 ansi 문법으로 변경하시오.

```
Select e.ename, d.loc
  From emp e, dept d
  Where e.deptno = d.deptno (+);
```

```
-----
Select e.ename, d.loc
  From emp e left outer join dept d
  On(e.deptno = d.deptno);
```

문제187. emp2와 telecom\_price를 조인 해서 아래의 결과를 출력하는데 1999 ansi문법으로 출력하시오.

28	김영토	64000
29	장은희	64000
30	백광흠	64000
31	이근호	(null)

```
SELECT e2.ename, tp.t_price
FROM EMP2 e2 left outer JOIN TELECOM_PRICE tp
ON(e2.TELECOM = tp.TELECOM);
```

## On 절을 사용한 조인 : on절에다가 조인 조건을 주는 문법 (가독성이 좋아짐)

Ex) 오라클 조인 문법

```
Select e.ename, d.loc
From emp e, dept d
Where e.deptno = d.deptno;
```

1999 ansi 조인 문법

```
Select e.ename, d.loc
From emp e join dept d
On (e.deptno = d.deptno);
```

문제188. 이름과 통신사 월정액을 출력하는데 on절을 사용한 조인문법으로 출력하시오.

```
SELECT e2.ename, tp.t_price
FROM EMP2 e2 JOIN TELECOM_PRICE tp
ON(e2.TELECOM = tp.telecom);
```

	ENAME	T_PRICE
1	김원섭	60000
2	김광록	65000
3	유혜린	65000
4	미유진	60000
5	윤진민	60000
6	백광흠	64000
7	장은희	64000

%% 3개의 테이블 조인하는 방법

```
emp ----- dept ----- salgrade
연결조건    연결조건  <--조인조건
```

Ex) 이름, 월급, 부서위치, 급여등급(grade)를 출력하시오.

-오라클 조인 문법

```
Select e.ename, e.sal, d.loc, s.grade
From emp e, dept d, salgrade s
Where e.deptno = d.deptno and e.sal between s.losal and s.hisal;
```

=> 조인조건 2개를 쓰면 사용할 수 있다.

문제189. 아래의 보너스라는 테이블을 생성하고 이름과 월급과 부서위치와 보너스를 출력하시오.

```
Create table bonus as
Select empno, sal * 30 as bonus
From emp;
```

```
-----
Select e.ename, e.sal, d.loc, b.bonus
From emp e, dept d, bonus b
Where e.deptno = d.deptno and e.empno = b.empno;
```

	ENAME	SAL	LOC	BONUS
1	KING	5000	NEW YORK	150000
2	BLAKE	2850	CHICAGO	85500
3	CLARK	2450	NEW YORK	73500
4	JONES	2975	DALLAS	89250
5	MARTIN	1250	CHICAGO	37500
6	ALLEN	1600	CHICAGO	48000
7	TURNER	1500	CHICAGO	45000
8	JAMES	950	CHICAGO	28500
9	WARD	1250	CHICAGO	37500
10	FORD	3000	DALLAS	90000
11	SMITH	800	DALLAS	24000
12	SCOTT	3000	DALLAS	90000
13	ADAMS	1100	DALLAS	33000
14	MILLER	1300	NEW YORK	39000

문제190. 위의 SQL을 on절을 사용한 조인 문법으로 변경하시오.

```

Select e.ename, e.sal, d.loc, b.bonus
  From emp e join dept d
    On(e.deptno = d.deptno)
      join bonus b
        On(e.empno = b.empno);

```

	ENAME	SAL	LOC	BONUS
1	KING	5000	NEW YORK	150000
2	BLAKE	2850	CHICAGO	85500
3	CLARK	2450	NEW YORK	73500
4	JONES	2975	DALLAS	89250
5	MARTIN	1250	CHICAGO	37500
6	ALLEN	1600	CHICAGO	48000
7	TURNER	1500	CHICAGO	45000
8	JAMES	950	CHICAGO	28500
9	WARD	1250	CHICAGO	37500
10	FORD	3000	DALLAS	90000
11	SMITH	800	DALLAS	24000
12	SCOTT	3000	DALLAS	90000
13	ADAMS	1100	DALLAS	33000
14	MILLER	1300	NEW YORK	39000

문제191. 이름, 월급, 부서위치, 급여등급(grade)을 출력하시오 (on절로)

```

Select e.ename, e.sal, d.loc, s.grade
  From emp e join dept d
    On(e.deptno = d.deptno)
      Join salgrade s
        On(e.sal between s.losal and s.hisal);

```

	ENAME	SAL	LOC	GRADE
1	KING	5000	NEW YORK	5
2	BLAKE	2850	CHICAGO	4
3	CLARK	2450	NEW YORK	4
4	JONES	2975	DALLAS	4
5	MARTIN	1250	CHICAGO	2
6	ALLEN	1600	CHICAGO	3
7	TURNER	1500	CHICAGO	3
8	JAMES	950	CHICAGO	1
9	WARD	1250	CHICAGO	2
10	FORD	3000	DALLAS	4
11	SMITH	800	DALLAS	1
12	SCOTT	3000	DALLAS	4
13	ADAMS	1100	DALLAS	1
14	MILLER	1300	NEW YORK	2

문제192. 이름과 부서위치를 출력하는데 DALLAS 에서 근무하는 직원들만 출력하시오.

(oracle, 1999ansi 둘다)

-oracle

```

Select e.ename, d.loc
  From emp e, dept d
  Where e.deptno = d.deptno
        And d.loc = 'DALLAS';

```

-1999 ansi

```

Select e.ename, d.loc
  From emp e join dept d
    On(e.deptno = d.deptno)
  Where d.loc = 'DALLAS';

```

	ENAME	LOC
1	JONES	DALLAS
2	FORD	DALLAS
3	SMITH	DALLAS
4	SCOTT	DALLAS
5	ADAMS	DALLAS

## Using 절을 사용한 조인

Ex) 이름과 부서위치를 using 절을 사용한 조인으로 출력하시오.

```

Select e.ename, d.loc
  From emp e join dept d
    Using (deptno);

```

	ENAME	T_PRICE
1	김원섭	60000
2	김광록	65000
3	유혜린	65000
4	미유진	60000
5	윤진민	60000

문제193. 학생이름, 통신사 월정액을 출력하는데 using절을 사용한 조인으로 출력하시오.

```

Select e2.ename, tp.t_price
  From emp2 e2 join telecom_price tp
    Using(telecom);

```



<-- e.deptno 라고 테이블 별칭을 사용하면 안됨

**Cross join : where절은 하나 하나 조인하는데 cross join은 모두 조인 시킨다.**

(어쩌다가 사용함)

Ex) -오라클 조인

```
select e.ename, d.loc  
      From emp e, dept d;
```

- 1999 ansi 조인

```
Select e.ename, d.loc  
      From emp e cross join dept d;
```

<input type="checkbox"/>	ENAME	LOC
1	KING	NEW YORK
2	BLAKE	NEW YORK
3	CLARK	NEW YORK
4	JONES	NEW YORK
5	MARTIN	NEW YORK
6	ALLEN	NEW YORK
7	TURNER	NEW YORK
8	JAMES	NEW YORK
9	WARD	NEW YORK
10	FORD	NEW YORK
11	SMITH	NEW YORK
12	SCOTT	NEW YORK
13	ADAMS	NEW YORK
14	MILLER	NEW YORK
15	KING	DALLAS

50	JAMES	BOSTON
51	WARD	BOSTON
52	FORD	BOSTON
53	SMITH	BOSTON
54	SCOTT	BOSTON
55	ADAMS	BOSTON
56	MILLER	BOSTON

Ex) 이름과 부서위치를 출력하시오.

```
Select e.ename, d.loc  
      From emp e natural join dept d;
```

<input type="checkbox"/>	ENAME	LOC
1	KING	NEW YORK
2	BLAKE	CHICAGO
3	CLARK	NEW YORK
4	JONES	DALLAS
5	MARTIN	CHICAGO

오라클이 알아서 emp와 dept테이블의  
공통된 컬럼을 찾아서 연결시켜 조인해준다.

(공통된 컬럼이 없으면 안됨!!)



## 8. 조인 실습

2018년 4월 4일 수요일 오전 11:29

**문제194. 부서위치, 부서위치별 토탈월급, 최대월급, 최소월급, 인원수를 출력하시오.**

```
Select d.loc, sum(e.sal), max(e.sal), min(e.sal), count (*)
From emp e, dept d
Where e.deptno = d.deptno
Group by d.loc;
```

	LOC	SUM(E.SAL)	MAX(E.SAL)	MIN(E.SAL)	COUNT(*)
1	NEW YORK	8750	5000	1300	3
2	CHICAGO	9400	2850	950	6
3	DALLAS	10875	3000	800	5

**문제195. 부서위치, 부서위치별 인원수를 출력하는데 부서위치별 인원수가 3명 이상인것만 출력하고 부서위치별 인원수가 높은 것부터 출력하시오.**

```
Select d.loc, count(*)
From emp e,dept d
where e.deptno =d.deptno
Group by d.loc
having count(*) >= 3
Order by count(*) desc;
```

	LOC	COUNT(*)
1	CHICAGO	6
2	DALLAS	5
3	NEW YORK	3

**문제196. 통신사, 통신사별 월정액 토탈금액을 출력하는데 통신사별 월정액 토탈금액이 10만원 이상인 것만**

**출력하고 월정액 토탈금액이 높은것부터 출력하시오.**

```
Select e2.telecom, TO_CHAR(sum(tp.t_price),'999,999')
From emp2 e2, telecom_price tp
Where e2.telecom = tp.telecom
Having SUM(tp.t_price) >= 100000
GROUP BY e2.TELECOM
Order by SUM(tp.t_price) desc;
```

	TELECOM	TO_CHAR(SUM(TP.T_PRICE),'999,999')
1	kt	832,000
2	lg	585,000
3	sk	480,000

**문제197. 아래와 같이 출력하시오. (매우중요★★★★★)**

	JOB	DALLAS	CHICAGO	NEW YORK
1	SALESMAN	(null)	5600	(null)
2	CLERK	1900	950	1300
3	PRESIDENT	(null)	(null)	5000
4	MANAGER	2975	2850	2450
5	ANALYST	6000	(null)	(null)

```

SELECT *
FROM(
    SELECT e.job, d.loc, e.sal
    FROM EMP e, DEPT d
    WHERE e.DEPTNO = d.deptno
)
pivot(SUM(sal) FOR loc IN('DALLAS' AS "DALLAS", 'CHICAGO' AS "CHICAGO", 'NEW YORK'
AS "NEW YORK"));

```

문제198. 부서위치, 부서위치별 토탈월급을 세로로 출력하시오.

```

SELECT d.loc, SUM(e.sal)
FROM EMP e, DEPT d
WHERE e.DEPTNO = d.deptno
GROUP BY d.loc;

```

	JOB	LOC	SUM(E.SAL)
1	SALESMAN	CHICAGO	5600
2	CLERK	DALLAS	1900
3	MANAGER	NEW YORK	2450
4	ANALYST	DALLAS	6000
5	CLERK	NEW YORK	1300
6	MANAGER	CHICAGO	2850
7	CLERK	CHICAGO	950
8	PRESIDENT	NEW YORK	5000
9	MANAGER	DALLAS	2975

문제199. 부서위치, 부서위치별 토탈월급을 아래와 같이 가로로 출력하시오.

```

SELECT *
FROM(
    SELECT d.loc, e.sal
    FROM EMP e, DEPT d
    WHERE e.DEPTNO = d.deptno
)
pivot(SUM(sal) FOR loc IN('DALLAS', 'CHICAGO', 'NEW YORK'));

```

	'DALLAS'	'CHICAGO'	'NEW YORK'
1	10875	9400	8750

문제200. 아래와 같이 입사한 년도(4자리)별 토탈월급을 가로로 출력하시오.

	'1980'	'1981'	'1982'	'1983'
1	800	22825	4300	1100

```

SELECT *
FROM(
    Select to_char(hiredate, 'RRRR') AS hire, sal
    From emp e
)
Pivot(sum(sal) for hire in ('1980', '1981', '1982', '1983'));

```

Pivot 문의 from절 서브쿼리 안에 함수를 사용했으면 컬럼 별칭을 사용해야한다.

문제201. 아래와 같이 결과를 출력하시오.

	LOC	'1980'	'1981'	'1982'	'1983'
1	NEW YORK	(null)	7450	1300	(null)
2	CHICAGO	(null)	9400	(null)	(null)
3	DALLAS	800	5975	3000	1100

```
SELECT *
FROM(
    SELECT d.loc, TO_CHAR(hiredate, 'RRRR') AS hire, sal
    FROM EMP e, DEPT d
    WHERE e.deptno = d.deptno
)
pivot(SUM(sal) FOR hire IN ('1980', '1981', '1982', '1983'));
```

문제202. 아래와 같이 결과를 출력하시오.

	JOB	DALLAS	CHICAGO	NEW YORK
1	SALESMAN	0	4	0
2	CLERK	2	1	1
3	PRESIDENT	0	0	1
4	MANAGER	1	1	1
5	ANALYST	2	0	0

```
SELECT *
FROM(
    SELECT e.job, d.loc
    FROM EMP e, DEPT d
    WHERE e.DEPTNO = d.DEPTNO
)
pivot(COUNT(*) FOR LOC IN('DALLAS' AS "DALLAS", 'CHICAGO' AS "CHICAGO", 'NEW
YORK' AS "NEW
YORK"));
```

Count 함수는 count(\*) 를 사용해야 null 값으로 인해 데이터가 잘못 출력되는 현상을 막을 수 있다.

문제203. DALLAS 에서 근무하는 직원들의 월급중에 최대월급을 출력하시오.

```
Select max(e.sal)
From emp e, dept d
Where e.empno = d.deptno
And d.loc = 'DALLAS'
```

%% DALLAS에서 근무하는 직원들의 월급만 출력하시오.

```
Select max(e.sal)
From emp e, dept d
Where e.deptno = d.deptno
And d.loc = 'DALLAS';
```

	MAX(E.SAL)
1	3000

문제204. CHICAGO에서 근무하는 직원들중에 가장 늦게 입사한 직원의 입사일을 출력하시오.

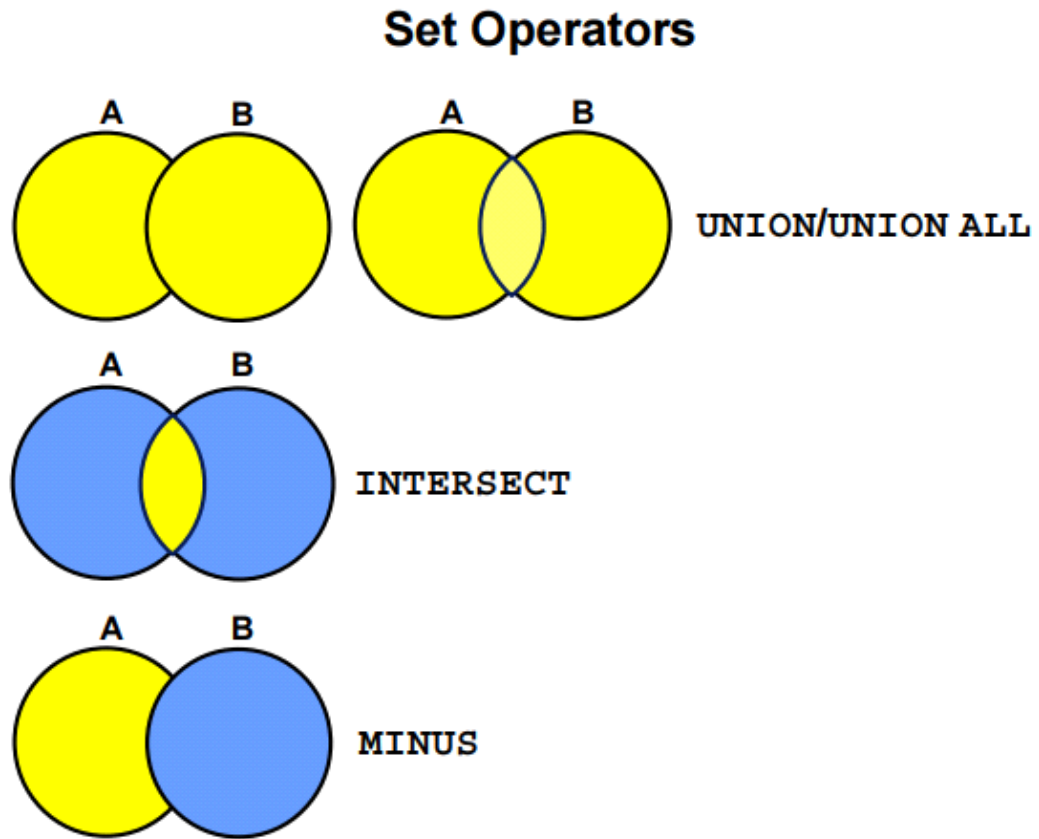
```
SELECT max(e.hiredate)
FROM EMP e, DEPT d
WHERE e.deptno = d.DEPTNO
AND d.loc = 'CHICAGO';
```

	MAX(E.HIREDATE)
1	1981-12-11 오전 12:00:00

## 9. 집합 연산자

2018년 4월 4일 수요일    오후 3:03

집합 연산자 : 두 집합 사이의 합집합과 교집합과 차집합을 출력하는 연산자



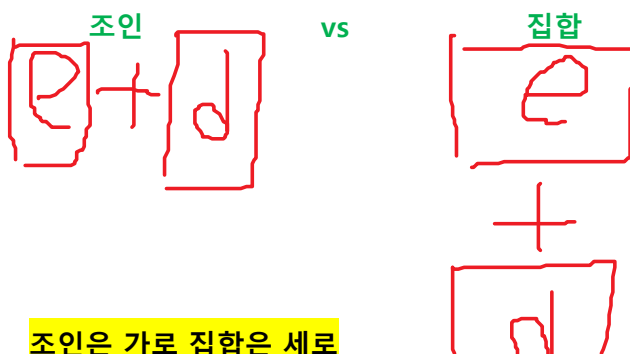
종류

1. Union all
2. Union
3. Intersect
4. minus

### 9-1. Union all

: 두 집합의 합 집합을 출력하는 연산자. (집합 = 결과 데이터,  
쿼리순서 = 아래부터)

\*조인과의 차이점



조인은 가로 집합은 세로



Ex) -직업, 직업별 토탈월급을 출력하시오

```

Select job, sum(sal)
From emp
Group by job;

```

	JOB	SUM(SAL)
1	SALESMAN	5600
2	CLERK	4150
3	PRESIDENT	5000
4	MANAGER	8275
5	ANALYST	6000

<-- 집합

-사원 테이블의 전체 토탈월급을 출력하시오.

```

Select sum(sal)
From emp;

```

<-- 집합

	SUM(SAL)
1	29025

-위의 두개의 결과 집합을 하나의 합 집합으로 출력하는게 합 집합.

%% 아래와 같이 결과를 출력하시오.

	JOB	SUM(SAL)
1	SALESMAN	5600
2	CLERK	4150
3	PRESIDENT	5000
4	MANAGER	8275
5	ANALYST	6000
6	전체토탈	29025

위에 컬럼의 개수가 2개면

아래 컬럼의 개수가 2개로 동일 해야한다.

```

SELECT job, SUM(sal)
FROM EMP
GROUP BY job

```

UNION ALL <--접착제 같이 붙여주는 역할

```

SELECT '전체토탈', SUM(sal)
FROM EMP;

```

문제205. 아래와 같이 결과를 출력하시오.

	JOB	TO_CHAR(SUM(SAL), '999,999')
1	SALESMAN	5,600
2	CLERK	4,150
3	PRESIDENT	5,000
4	MANAGER	8,275
5	ANALYST	6,000
6	전체토탈	29,025

```

SELECT job, TO_CHAR(SUM(sal), '999,999')
FROM EMP
GROUP BY job
UNION ALL

```

```

SELECT '전체토탈', TO_CHAR(SUM(sal), '999,999')
FROM EMP;

```

문제206. 아래와 같이 결과를 출력하시오.

	TO_CHAR(DEPTNO)	SUM(SAL)
1	20	10875
2	10	8750
3	30	9400
4	전체토탈	29025

```

Select TO_char(deptno), sum(sal)
  From EMP
  GROUP BY TO_CHAR(deptno)
  Union all
Select '전체토탈' as "전체토탈", sum(sal)
  FROM EMP;

```

문제207. 직업, 직업별 인원수를 출력하고, 맨 아래쪽에 전체 인원수를 출력하시오.

```

Select job, count(*)
  From emp
  Union all
Select '전체 인원수' as "전체 인원수", count(*)
  From emp

```

	JOB	COUNT(*)
1	SALESMAN	4
2	CLERK	4
3	PRESIDENT	1
4	MANAGER	3
5	ANALYST	2
6	전체 인원수	14

문제208. 부서번호, 부서번호별 평균월급을 출력하는데 맨 아래쪽에 전체 평균 월급을 출력하시오.

```

SELECT deptno, ROUND(avg(sal))
  FROM EMP
  GROUP BY deptno
  UNION ALL
SELECT null, ROUND(AVG(sal))
  FROM EMP;

```

	DEPTNO	ROUND(AVG(SAL))
1	30	1567
2	20	2175
3	10	2917
4	(null)	2073

문제209. 부서번호, 부서번호별 토탈월급을 출력하는데 아래와 같이 맨 아래쪽에 전체 토탈을 출력하시오.

```

Select deptno, sum(sal)
  From emp
  Group by deptno
  Union all
Select null, sum(sal)
  From emp;

```

	DEPTNO	SUM(SAL)
1	30	9400
2	20	10875
3	10	8750
4	(null)	29025

%%%%%%%%%%  
 레포팅함수 rollup : 자동 union



**cube  
grouping sets  
grouping**

튜닝후>> select deptno, sum(sal)  
From emp  
Group by **rollup**(deptno); 같은 결과나옴

**문제210. 직업, 직업별 토탈월급을 출력하는데 rollup을 써서 아래와 같이 출력하시오.**

<input type="checkbox"/>	NVL(JOB,'전체토탈')	TO_CHAR(SUM(SAL),'999,999')
1	ANALYST	6,000
2	CLERK	4,150
3	MANAGER	8,275
4	PRESIDENT	5,000
5	SALESMAN	5,600
6	전체토탈	29,025

Select NVL(job,'전체토탈'), to\_char(sum(sal),'999,999')  
From emp  
Group BY **rollup**(job);

**문제211. 입사한 년도(4자리), 입사한 년도별 토탈월급을 출력하시오.**

Select to\_char(hiredate, 'RRRR'), sum(sal)  
From emp  
Group by to\_char(hiredate, 'RRRR');

<input type="checkbox"/>	TO_CHAR(HIREDATE,'RRRR')	SUM(SAL)
1	1980	800
2	1983	1100
3	1982	4300
4	1981	22825

**문제212. 위의 결과에서 전체 토탈월급이 출력되게하시오.**

Select to\_char(hiredate, 'RRRR'), sum(sal)  
From emp  
Group by ROLLUP(to\_char(hiredate, 'RRRR'));

<input type="checkbox"/>	TO_CHAR(HIREDATE,'RRRR')	SUM(SAL)
1	1980	800
2	1981	22825
3	1982	4300
4	1983	1100
5	(null)	29025

**문제213. 부서위치, 부서위치별 토탈 월급을 출력하시오.**

Select d.loc, sum(e.sal)  
from EMP e, DEPT d  
WHERE e.DEPTNO = d.DEPTNO  
Group by d.loc;

<input type="checkbox"/>	LOC	SUM(E.SAL)
1	NEW YORK	8750
2	CHICAGO	9400
3	DALLAS	10875

**문제214. 위의 결과에서 맨 아래쪽에 전체 토탈 월급이 출력되게하시오.**

Select d.loc, sum(e.sal)  
from EMP e, DEPT d  
WHERE e.DEPTNO = d.DEPTNO

Group by ROLLUP(d.loc);

	LOC	SUM(E.SAL)
1	CHICAGO	9400
2	DALLAS	10875
3	NEW YORK	8750
4	(null)	29025

cube

문제215. 그런데 토달월급이 아래쪽에 출력되는게 아니라 위쪽에 출력되게 하시오.

```
Select d.loc, sum(e.sal)
from EMP e, DEPT d
WHERE e.DEPTNO = d.DEPTNO
Group by cube(d.loc);
```

	LOC	SUM(E.SAL)
1	(null)	29025
2	DALLAS	10875
3	CHICAGO	9400
4	NEW YORK	8750

문제216. 위의 결과를 다시출력하는데 큐브쓰지말고 union all 로 수행 해보시오.

```
SELECT NULL, SUM(sal)
FROM EMP
UNION all
SELECT d.loc, SUM(sal)
FROM EMP e, DEPT d
WHERE e.deptno = d.deptno
GROUP BY d.loc;
```

	NULL	SUM(SAL)
1	(null)	29025
2	NEW YORK	8750
3	CHICAGO	9400
4	DALLAS	10875

## 9-2. Union

: 두 집합의 합집합을 출력하는데 중복을 제거, 정렬 해서 출력한다.

```
Ex) select deptno
From emp
Union all
```

```
Select deptno
From emp;
```

	DEPTNO
1	10
2	30
3	10
4	20
5	30
6	30
7	30
8	30
9	30
10	20
11	20
12	20
13	20
14	10
15	10

```
select deptno
  From emp
Union
select deptno
  From emp;
```

	DEPTNO
1	10
2	20
3	30

문제216. 아래의 SQL의 결과와 똑같은 결과가 나오게 union all 로 작성해보시오.

```
Select deptno, sum(sal)
  From emp
  Group by deptno
Union
```

```
Select null, sum(sal)
  From emp;
```

위 컬럼명이라 아래 컬럼명이 같아야지 order by절을 사용가능

	DEPTNO	SUM(SAL)
1	10	8750
2	20	10875
3	30	9400
4	(null)	29025

```
Select deptno, sum(sal)
  From emp
  Group by deptno
UNION all
```

```
Select null as deptno, sum(sal)
  From EMP
```

ORDER BY deptno asc;      order by절은 맨 마지막 쿼리에 사용해야함.

	DEPTNO	SUM(SAL)
1	10	8750
2	20	10875
3	30	9400
4	(null)	29025

문제217. 아래의 SQL 결과를 union all로 수행하시오.

```
Select deptno, sum(sal)
  From emp
  Group by cube(deptno);
```

	DEPTNO	SUM(SAL)
1	(null)	29025
2	10	8750
3	20	10875
4	30	9400

```
SELECT NULL AS deptno, SUM(sal)
  FROM EMP
  UNION ALL
SELECT deptno, SUM(sal)
  FROM EMP
  GROUP BY deptno
  ORDER BY deptno nulls first;
```

Grouping set

문제218. 직업, 부서번호, 직업별 부서번호별 토탈월급을 출력하시오.

```
Select job, deptno, sum(sal)
  From emp
 Group by job, deptno
 ORDER BY deptno, job;
```

	JOB	DEPTNO	SUM(SAL)
1	CLERK	10	1300
2	MANAGER	10	2450
3	PRESIDENT	10	5000
4	ANALYST	20	6000
5	CLERK	20	1900
6	MANAGER	20	2975
7	CLERK	30	950
8	MANAGER	30	2850
9	SALESMAN	30	5600

## ROLLUP

문제219. 아래와 같이 결과를 출력하시오.

	JOB	DEPTNO	SUM(SAL)
1	CLERK	10	1300
2	MANAGER	10	2450
3	PRESIDENT	10	5000
4	ANALYST	20	6000
5	CLERK	20	1900
6	MANAGER	20	2975
7	CLERK	30	950
8	MANAGER	30	2850
9	SALESMAN	30	5600
10	(null)	(null)	29025

```
Select deptno, job sum(sal)
      From emp
```

Group by ROLLUP((deptno, job)) <- ()를 하나 더쓴이유 : 하나의 집합으로 만들어줘서 출력하고 이  
ORDER BY deptno, job; 집합이 없어지고 전체만나옴. 안쓰면job, deptno별로 따로 토탈월  
급이 나옴

## 안쓰면

	DEPTNO	JOB	SUM(SAL)
1	10	CLERK	1300
2	10	MANAGER	2450
3	<del>10</del>	<del>PRESIDENT</del>	<del>5000</del>
4	10	(null)	8750
5	20	CLERK	1900
6	20	ANALYST	6000
7	<del>20</del>	<del>MANAGER</del>	<del>2975</del>
8	20	(null)	10875
9	30	CLERK	950
10	30	MANAGER	2850
11	30	SALESMAN	5600
12	30	(null)	9400
13	(null)	(null)	29025

문제220. 텔레콤, 전공, 텔레콤별 전공별 인원수를 출력하시오.

```
Select telecom, major, count(*)
  From emp2
  Group by (telecom, major)
```

ORDER BY telecom;

	TELECOM	MAJOR	COUNT(*)
1	cjh	통계학	1
2	kt	경영정보학	1
3	kt	경영학	1
4	kt	경제학	2
5	kt	광고경영학	1

문제221. 위의 결과에서 전체 토탈 인원수가 나오게 출력하시오.

```
Select telecom, major, count(*)
From emp2
Group by rollup((telecom, major))
ORDER BY telecom;
```

20	sk	환경공학	1
21	sk	컴퓨터공학	1
22	sk	시스템경영	1
23	sk	컴퓨터정보	1
24	cjh	통계학	1
25	(null)	(null)	31

문제222. 위의 결과를 아래와 같이 중간중간 통신사별 인원수도 출력되게 하시오.

```
Select telecom, major, count(*)
From emp2
Group by rollup (telecom, major) <-- ()하나 빼기
ORDER BY telecom;
```

	TELECOM	MAJOR	COUNT(*)
1	kt	경영학	1
2	kt	경제학	2
3	kt	전자공학	1
4	kt	경영정보학	1
5	kt	광고경영학	1
6	kt	문헌정보학	2
7	kt	컴퓨터공학	4
8	kt	도시계획학	1
9	kt	(null)	13
10	lg	신학	1

10	lg	신학	1
11	lg	물리학	1
12	lg	경영정보학	2
13	lg	광고홍보학	1
14	lg	산업공학	1
15	lg	기술융합학	1
16	lg	정보통신학	1
17	lg	컴퓨터정보	1
18	lg	(null)	9

**Grouping sets : rollup 보다 grouping 된 결과를 더 잘 예상 할 수 있고 작성하기 편하다.**

(현업에서 많이쓰는 이유 : 내가 원하는 결과만 보고싶으면 그것만 쓰면됨.)

Ex) 문제222번을 rollup이 아니라 grouping sets로 변경하시오.

```
Select telecom, major, count(*)
From emp2
Group by grouping sets( (telecom, major), (telecom), () ); <- telecom별 인원수
```

문제223. 아래의 rollup의 결과를 grouping sets로 수행해보시오.

```

Select deptno, sum(sal)
From emp
Group by rollup(deptno);

```

<input type="checkbox"/>	DEPTNO	SUM(SAL)
1	10	8750
2	20	10875
3	30	9400
4	(null)	29025

```

Select deptno, sum(sal)
From emp
Group by grouping sets( (deptno, () );

```

<input type="checkbox"/>	DEPTNO	SUM(SAL)
1	10	8750
2	20	10875
3	30	9400
4	(null)	29025

**문제224. 아래의 rollup의 결과를 grouping sets로 변경하시오.**

```

Select deptno, job, sum(sal)
From emp
Group by rollup(deptno, job);

```

<input type="checkbox"/>	DEPTNO	JOB	SUM(SAL)
1	10	CLERK	1300
2	10	MANAGER	2450
3	10	PRESIDENT	5000
4	10	(null)	8750
5	20	CLERK	1900
6	20	ANALYST	6000
7	20	MANAGER	2975
8	20	(null)	10875
9	30	CLERK	950
10	30	MANAGER	2850
11	30	SALESMAN	5600
12	30	(null)	9400
13	(null)	(null)	29025

```

Select deptno, job, sum(sal)
From emp
Group by GROUPING sets((deptno,job),(deptno, ());

```

**문제225.아래와 같이 결과를 출력하시오.**

<input type="checkbox"/>	DEPTNO	NVL(JOB,'부서토탈:')	SUM(SAL)
1	10	CLERK	1300
2	10	MANAGER	2450
3	10	PRESIDENT	5000
4	10	부서토탈 :	8750
5	20	CLERK	1900
6	20	ANALYST	6000
7	20	MANAGER	2975
8	20	부서토탈 :	10875
9	30	CLERK	950
10	30	MANAGER	2850
11	30	SALESMAN	5600
12	30	부서토탈 :	9400
13	(null)	부서토탈 :	29025

```

Select deptno, NVL(job, '부서토탈 :'), sum(sal)
  From emp
  Group by GROUPING sets((deptno,job),(deptno), ());

```

문제226. 아래와 같이 결과를 출력하시오.

	DEPTNO	DECODE(DEPTNO,NULL,'전체토탈:',NVL(JOB,'부서토탈:'))	SUM(SAL)
1	10	CLERK	1300
2	10	MANAGER	2450
3	10	PRESIDENT	5000
4	10	부서토탈:	8750
5	20	CLERK	1900
6	20	ANALYST	6000
7	20	MANAGER	2975
8	20	부서토탈:	10875
9	30	CLERK	950
10	30	MANAGER	2850
11	30	SALESMAN	5600
12	30	부서토탈:	9400
13	(null)	전체토탈:	29025

```

Select deptno, DECODE(deptno, NULL, '전체토탈:', nvl(job, '부서토탈:')), sum(sal)
  From emp
  Group by GROUPING sets((deptno,job),(deptno), ());

```

==> 부서번호가 null이면 전체토탈, 그렇지 않으면 job이 출력되게하는데  
직업이 null 이면 부서토탈을 출력해라.

문제227. 아래와 같이 결과를 출력하시오.

	DEPTNO	ENAME	SUM(SAL)
1	10	KING	5000
2	10	CLARK	2450
3	10	MILLER	1300
4	20	FORD	3000
5	20	ADAMS	1100
6	20	JONES	2975
7	20	SCOTT	3000
8	20	SMITH	800
9	30	WARD	1250
10	30	ALLEN	1600
11	30	BLAKE	2850
12	30	JAMES	950
13	30	MARTIN	1250
14	30	TURNER	1500
15	(null)	(null)	29025

```

SELECT deptno, ename, SUM(sal)
  FROM EMP
  GROUP BY GROUPING SETS( (deptno, ename), ());

```

문제228. 위의 결과를 union all로 수행하시오.

```

SELECT deptno, ename, sal
  FROM EMP
  UNION all
SELECT NULL AS deptno, NULL AS ename, SUM(sal)
  FROM EMP
  ORDER BY deptno;

```

문제229. 아래와 같이 결과를 출력하시오. (rollup, grouping sets 둘다가능)

	NVL(JOB,'토탈값')	10	20	30
1	ANALYST	0	6000	0
2	CLERK	1300	1900	950
3	MANAGER	2450	2975	2850
4	PRESIDENT	5000	0	0
5	SALESMAN	0	0	5600
6	토탈 값	8750	10875	9400

```

Select NVL(job,'토탈 값'), sum(decode(deptno,10, sal, 0 ))"10",
        sum(decode(deptno,20, sal, 0 ))"20",
        sum(decode(deptno,30, sal, 0 ))"30"
From emp
Group by GROUPING sets( (job) ,( ) );

```

문제230. 나이별, 통신사별 인원수를 아래와 같이 출력하시오.

	나이	SUM(SK)	SUM(KT)	SUM(LG)	SUM(CJH)
1	24	1	2	0	0
2	25	2	2	1	0
3	26	1	2	2	0
4	27	2	4	2	0
5	28	2	3	2	1
6	29	0	0	1	0
7	32	0	0	1	0
8	토탈	8	13	9	1

%%pivot 문으로

```

SeLECT NVL(TO_CHAR(age), '토탈') 나이, SUM(sk), SUM(kt), SUM(lg), SUM(cjh)
FROM(SELECT age,telecom
      FROM EMP2
      )
pivot(COUNT(*) FOR telecom IN ( 'sk' AS sk, 'kt' AS kt, 'lg' AS lg, 'cjh' AS cjh ))
GROUP BY GROUPING sets( (age), ( ))
ORDER BY age;

```

%% sum(decode)로

```

SELECT NVL(TO_CHAR(age), '토탈') 나이, SUM(DECODE(telecom, 'sk', 1, 0 )) "sk",
        SUM(DECODE(telecom, 'kt', 1, 0 )) "kt",
        SUM(DECODE(telecom, 'lg', 1, 0 )) "lg",
        SUM(DECODE(telecom, 'cjh', 1, 0 )) "cjh"
FROM EMP2
GROUP BY GROUPING SETs( (age), ());

```

%%count(decode)로

```

SELECT NVL(TO_CHAR(age), '토탈') 나이, count(DECODE(telecom, 'sk', ename, null )) "sk",
        count(DECODE(telecom, 'kt', ename, null )) "kt",
        count(DECODE(telecom, 'lg', ename, null )) "lg",
        count(DECODE(telecom, 'cjh', ename, null )) "cjh",
FROM EMP2
GROUP BY GROUPING SETs( (age), ());

```

문제231. 아래와 같이 결과를 출력하시오.



	나이	sk	kt	lg	cjh	COUNT(*)
1	24	1	2	0	0	3
2	25	2	2	1	0	5
3	26	1	2	2	0	5
4	27	2	4	2	0	8
5	28	2	3	2	1	8
6	29	0	0	1	0	1
7	32	0	0	1	0	1
8	토탈	8	13	9	1	31

```

SELECT NVL(TO_CHAR(age), '토탈') 나이, count(DECODE(telecom, 'sk', ename, null ) ) "sk",
      count(DECODE(telecom, 'kt', ename, null ) ) "kt",
      count(DECODE(telecom, 'lg', ename, null ) ) "lg",
      count(DECODE(telecom, 'cjh', ename, null ) ) "cjh",
      count(*)
FROM EMP2
GROUP BY GROUPING SETS( (age), ());

```

**Grouping : null과 grouping 되는 결과를 보기위해서 어쩔수 없이 null로 출력되는 데이터를 구분해 주기 위해서 사용.**

```

Ex) select deptno, sum(sal), grouping(deptno)
From emp
Group by rollup(deptno);

```

```

UPDATE EMP
SET deptno = NULL
WHERE deptno = 10;

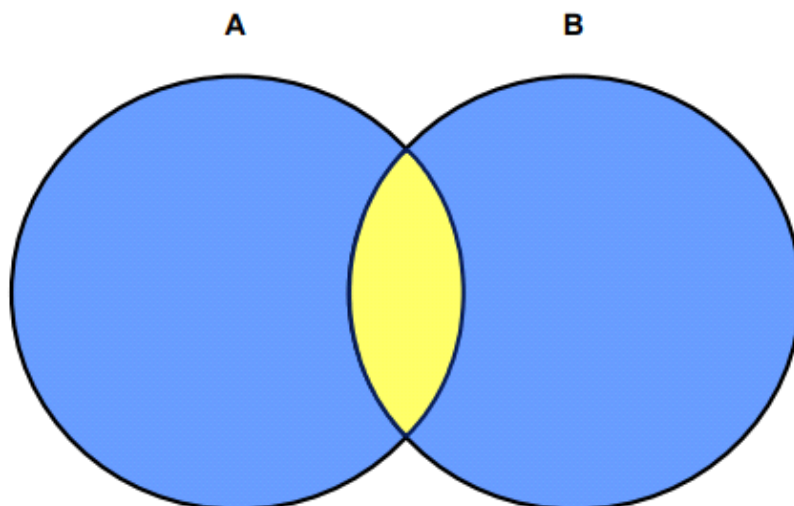
```

<- 바꾼 null  
<- 진짜 null

	DEPTNO	SUM(SAL)	GROUPING(DEPTNO)
1	20	10875	0
2	30	9400	0
3	(null)	8750	0
4	(null)	29025	1

### 9-3. Intersect

**: 교집합을 출력하는 연산자**



**INTERSECT 연산자는 양쪽 query에 공통되는 행을 반환합니다.**

Ex) 학생이름이 2건이 입력된 학생을 출력하시오.

```
Select ename, count(*)
From emp2
having count(ename) = 2;
```

	ENAME	COUNT(*)
1	송윤호	2

- 윤호의 중복된 데이터 1건을 삭제한다.

```
Select rowid, empno      rowid : 해당 row의 주소
From emp2
Where ename = '송윤호';
```

	ROWID	EMPNO
1	AAAE5wAABAAALDJAAl	2
2	AAAE5wAABAAALDJAAS	2

```
Delete from emp2
Where rowid = 'AAAE5wAABAAALDJAAl';
```

- 우리반에 전공이 경제학인 학생들의 데이터만 가지고 emp8 이라는 테이블을 생성하시오.

```
Create table emp8
As
select *
From emp2
Where major = '경제학';
```

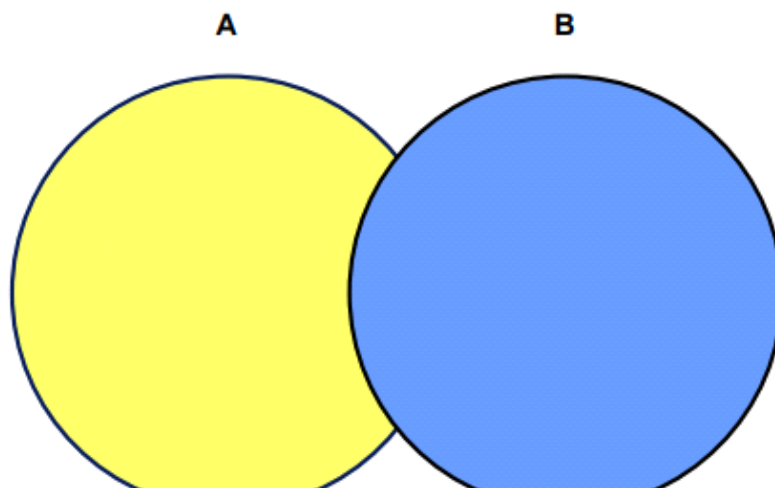
- Emp2테이블과 emp8 테이블의 **교집합 데이터**를 출력하시오.

```
Select *
From emp2
Intersect
Select *
From emp8;
```

	EMPNO	ENAME	AGE	BIRTH	MAJOR	EMAIL	MOBILE	ADDRESS	TELECOM
1	1	윤진민	27	1992-07-15 오전 12:00:00	경제학	yjmm92@gmail.com	010-4844-4224	경기도 구리시 교문동	sk
2	22	신현수	27	1992-05-04 오전 12:00:00	경제학	shyuns0504@gmail.com	010-2079-6714	서울시 관악구 삼성동	kt
3	28	차호성	28	1991-07-01 오전 12:00:00	경제학	hoseong0701@naver.com	010-4946-9144	서울시 관악구 신림동	kt

## 7-4. Minus

: 두집합 사이의 차집합을 출력하는 연산자





**MINUS 연산자는 첫번째 query에 의해 선택되지만 두번째 query 결과 집합에는 없는 모든 구분 행을 반환합니다.**

Ex) emp2와 emp8과의 차집합을 출력하시오.

```
Select *
  From emp2
 Minus
Select *
  From emp8;
```

EMPNO	ENAME	AGE	BIRTH	MAJOR	EMAIL	MOBILE	ADDRESS	TELECOM
1	2 송승호	27	1992-05-25 오전 12:00:00	문헌정보학	songssmart@naver.com	010-6859-3198	서울시 노원구 중계동	kt
2	3 은해찬	29	1990-07-18 오전 12:00:00	경영정보학	whitestar718@naver.com	01046142261	경기도 의왕시 오전동	lg
3	4 김영도	28	1991-10-13 오전 12:00:00	도시계획학 및 응용통계학	dudxh1@naver.com	010-3307-2231	경기도 안양시 동안구 평촌동	kt
4	5 정호진	32	1987-09-22 오전 12:00:00	물리학	tbop02@gmail.com	010-2985-9834	서울시 광진구 화양동	lg
5	6 김지우	27	1992-09-23 오전 12:00:00	전자공학	jiwoo00@naver.com	010-6435-0984	서울시 도봉구 쌍문동	kt
6	7 정인중	27	1992-09-11 오전 12:00:00	컴퓨터정보통신	jj911@naver.com	010-4177-3860	경기도 남양주시 별내면	lg
7	8 이유진	25	1994-12-06 오전 12:00:00	시스템경영공학	yjin1435@gmail.com	010-5920-3050	서울시 동작구 사당동	sk
8	9 백광훈	26	1993-06-18 오전 12:00:00	컴퓨터공학	bkh974@naver.com	010-7604-5060	서울시 노원구 공릉동	kt
9	10 김원섭	26	1994-02-08 오전 12:00:00	컴퓨터공학	rladnjstjgek@naver.com	010-3308-0492	경기도 남양주시 와부읍	sk
10	11 김광록	27	1992-12-16 오전 12:00:00	경영정보학	kwangrok92@naver.com	010-7554-5473	경기도 용인시 기흥구 하갈동	lg
11	12 장은희	24	1996-01-07 오전 12:00:00	경영정보학	ja5772@naver.com	010-5423-4327	서울시 광진구 군자동	kt
12	13 지윤철	28	1991-10-20 오전 12:00:00	광고경영학	hans546@naver.com	010-3021-8753	인천시 남동구 만수동	kt

경제학빼고 출력

## %% 데이터의 종류

1. Small data (business data)
2. Big data

Small 분석을 하기위해서 쿼리를 짤라고 한다면

개발서버-----운영서버

(테스트서버)

1억건

10억건

나의 자리

반장 자리

Emp2

emp2

- 반장 자리에 emp2 테이블을 조회하기 위해 db링크를 생성한다.

Create database link **dblink7**

Connect to scott

Identified by tiger

Using '192.168.19.3:1521/xe';

(반장) 아이피주소 : 포트번호 / db이름

- 반장 자리의 emp2 테이블을 조회한다.

Select \* from emp2@dblink7;

- 반장 자리의 emp2와 내자리의 emp2 데이터의 차이가 존재하는지 확인하시오.

Select from emp2@dblink7

Minus

Select \* from emp2;

**문제232. 짝공하고 나하고 emp2 테이블의 데이터의 차이가 존재하는지 확인하시오.**

나

짝공

Create database link dblink8

Connect to scott

Identified by tiger

Using '192.168.19.16:1521/xe';

Select from emp2@dblink8

Minus

Select \* from emp2;

## 10. 집합 연산자 사용시 주의 사항

2018년 4월 5일 목요일    오후 3:04

1. 집합 연산자 위아래의 쿼리문의 **컬럼의 개수와 데이터 타입이 동일** 해야한다.

Ex) select **ename, sal**  
      From sal  
      Union all  
      select **ename, sal**  
      From emp;

2. **Order by** 절은 맨 아래쪽 쿼리에만 사용할 수 있고, **order by** 절을 사용하려면 쿼리문들의 컬럼명이 전부 동일해야 한다.

Ex) select deptno, sum(sal)  
      From emp  
      Group by **deptno**  
      Union all  
      select **to\_number(null) as deptno**, sum(sal) <-- null 이더라도 맞춰줘야함.  
      From emp  
      **Order by deptno desc nulls first;**

# 11. 서브쿼리란?

2018년 4월 5일 목요일    오후 3:22

## 11-1. Single row subquery

Ex) JONES 보다 더 많은 월급을 받는 직원들의 이름과 월급을 출력하시오.

```
SELECT ename, sal      <-- 메인쿼리
FROM EMP
WHERE sal > (SELECT sal
              FROM EMP
              WHERE ename = 'JONES'); <-- 서브쿼리
```

	ENAME	SAL
1	KING	5000
2	FORD	3000
3	SCOTT	3000

문제233. SCOTT과 같은 월급을 받는 직원의 이름과 월급을 출력하시오.

```
Select ename, sal
From emp
Where sal = (select sal
              From emp
              Where ename='SCOTT');
```

	ENAME	SAL
1	FORD	3000
2	SCOTT	3000

문제234. 위의 결과에서 SCOTT을 제외해보시오.

```
Select ename, sal
From emp
Where sal = (select sal
              From emp
              Where ename = 'SCOTT') AND ename != 'SCOTT';
```

	ENAME	SAL
1	FORD	3000

문제235. ALLEN 보다 늦게 입사한 직원들의 이름과 입사일을 출력하시오.

```
Select ename, hiredate
From emp
Where hiredate > (select hiredate
                   From emp
                   where ename ='ALLEN')
```

	ENAME	HIREDATE
1	KING	1981-11-17 오전 12:00:00
2	BLAKE	1981-05-01 오전 12:00:00
3	CLARK	1981-05-09 오전 12:00:00
4	JONES	1981-04-01 오전 12:00:00
5	MARTIN	1981-09-10 오전 12:00:00
6	TURNER	1981-08-21 오전 12:00:00
7	JAMES	1981-12-11 오전 12:00:00
8	WARD	1981-02-23 오전 12:00:00
9	FORD	1981-12-11 오전 12:00:00
10	SCOTT	1982-12-22 오전 12:00:00
11	ADAMS	1983-01-15 오전 12:00:00
12	MILLER	1982-01-11 오전 12:00:00

문제236. 최대월급을 받는 사원의 이름과 월급을 출력하시오.

```

Select ename, sal
  From emp
 Where sal = (select max(sal)
              From emp);

```

	ENAME	SAL
1	KING	5000

문제237. 우리반에서 나이가 가장 어린 학생들의 이름과 나이를 출력하시오.

```

Select ename, age
  From emp2
 Where age = (select min(age)
              From emp2);

```

	ENAME	AGE
1	장은희	24
2	김근마	24
3	한지윤	24

- 실습 데이터 입력

```

create table university_tuition
(division varchar2 (20),
 type      varchar2(20 ),
 university varchar2(50),
 campus_type varchar2( 40),
 admission_fee number( 20),
 college_fee   number( 20),
 supporting_fee number( 20),
 tuition       number(20 ) ) ;

```

문제238. 자기가 다니거나 졸업한 학교의 모든 컬럼을 출력하시오.

```

Select *
  From university_tuition
 Where university = '고려대학교';

```

	DIVISION	TYPE	UNIVERSITY	CAMPUS_TYPE	ADMISSION_FEE	COLLEGE_FEE	SUPPORTING_FEE	TUITION
1	대학	사립	고려대학교	본교	1031	8208	0	8208

문제239. 전국에서 가장 등록금이 비싼 학교 이름을 출력하시오.

```

Select UNIVERSITY
  From university_tuition
 Where TUITION = (select max(TUITION)

```

From UNIVERSITY\_TUITION);

	UNIVERSITY
1	명지대학교 자연캠퍼스

- 테이블명 변경

Rename university\_tuition to univ; 기존이름 to 새로운 이름;

**문제240. 전국에서 가장 등록금이 낮은 학교이름과, 등록금을 출력하시오.**

```
Select university, tuition
From univ
Where tuition = (select min(tuition)
                  From univ
                  Where tuition != '0');
```

	UNIVERSITY	TUITION
1	한국방송통신대학교	703

**문제241. 전국에서 가장 비싼 생필품 이름과 가격과 파는곳을 출력하시오.**

```
SELECT A_NAME, A_PRICE, M_NAME
FROM PRICE
WHERE A_price = (SELECT max(A_Price)
                  FROM price);
```

	A_NAME	A_PRICE	M_NAME
1	조기(국산,생물)	81670	신세계백화점

**문제242. 살인의 가장 큰 원인이 무엇인지 알아내시오.(기타는 제외)**

```
SELECT cause
FROM CRIME_CAUSE2
WHERE cnt = (SELECT MAX(cnt)
              FROM CRIME_CAUSE2
              WHERE crime_type = '살인' and cause != '기타');
```

	CAUSE
1	사고

**문제243. 가정불화로 인해 생기는 가장 큰 범죄가 무엇인지 출력하시오.**

```
SELECT crime_type
FROM CRIME_CAUSE2
WHERE cnt = (SELECT max(cnt)
              FROM CRIME_CAUSE2
              WHERE cause = '가정불화');
```

	CRIME_TYPE
1	폭행

**문제244. JONES와 직업이 같은 직원들의 이름과 직업을 출력하시오.**

```
Select ename, job
From emp
Where job = (select job
              From emp
              Where ename = 'JONES');
```



문제245. 위의 결과에서 JONES는 제외하고 출력하시오.

```
Select ename, job
From emp
Where job = (select job
              From emp
              Where ename = 'JONES')
And ename != 'JONES';
```

## 11-2. Multiple row subquery : 서브쿼리 안에 2개 이상 값이 나오는 것.

문제246. 직업이 SALESMAN 인 직원들과 월급이 같은 직원들의 이름과 월급과 직업을 출력하시오.

```
Select ename, sal, job
From emp
Where sal in (select sal
              From emp
              Where job ='SALESMAN'); <-- 1250, 1250, 1600, 5000가 되어서 = 못씀
```

문제247. 전공이 경제학인 학생들과 통신사가 같은 학생들의 이름과 전공과 통신사를 출력하시오.

```
Select ename, major, telecom
From emp2
Where telecom in (select telecom
                  From emp2
                  Where major = '경제학');
```

	ENAME	SAL	JOB
1	WARD	1250	SALESMAN
2	MARTIN	1250	SALESMAN
3	ALLEN	1600	SALESMAN
4	TURNER	1500	SALESMAN

문제248. 위의 결과에서 전공이 경제학과인 학생들은 제외하고 출력하시오.

```
Select ename, major, telecom
From emp2
Where telecom in (select telecom
                  From emp2
                  Where major = '경제학')
And major != '경제학';
```

	ENAME	MAJOR	TELECOM
1	한지윤	경영학	sk
2	김건태	환경공학	sk
3	윤동환	컴퓨터정보시스템과	sk
4	김동윤	경영학	sk
5	김대경	통계학	sk
6	이유진	시스템경영공학	sk
7	김원섭	컴퓨터공학	sk
8	지윤철	광고경영학	kt
9	김근마	컴퓨터공학	kt
10	김지우	전자공학	kt
11	송윤호	문헌정보학	kt
12	신영근	컴퓨터공학	kt
13	미한새	경영학	kt

문제249. DALLAS에 있는 부서번호에서 근무하는 직원들의 이름과 월급을 출력하시오.

```
Select ename, sal
  From emp
 Where deptno = (select deptno
                  From dept
                  Where loc ='DALLAS'); <- 결과가 1개지만 in을써도 나오긴하는데 바꾸는데
                                       시간이 걸려서나옴
```

	ENAME	SAL
1	ADAMS	1100
2	SCOTT	3000
3	SMITH	800
4	FORD	3000
5	JONES	2975

문제250. king에게 보고하는 직원들의 이름과 월급을 출력하시오.

```
Select ename, sal
  From emp
 Where mgr = (select empno
               From emp
               Where ename = 'king');
```

	ENAME	SAL
1	BLAKE	2850
2	CLARK	2450
3	JONES	2975

### • 서브 쿼리시 사용하는 연산자

#### 1. Single row subquery

연산자 : =, <>, !=, ^=, >, <, >=, <=

#### 2. Multiple row subquery

연산자 : in, not in, >all, <all, >any, <any (in의 의미 : =any)

문제251. SCOTT과 직업이 같지 않은 직원들의 이름과 직업을 출력하시오.

```
Select ename, sal
  From emp
 Where job != (select job
                From emp
                Where ename = 'SCOTT');
```

	ENAME	SAL
1	KING	5000
2	BLAKE	2850
3	CLARK	2450
4	JONES	2975
5	MARTIN	1250
6	ALLEN	1600
7	TURNER	1500
8	JAMES	950
9	WARD	1250
10	SMITH	800
11	ADAMS	1100
12	MILLER	1300

문제252. 직업이 SALESMAN인 직원들과 월급이 같지 않은 직원들의 이름과 월급과 직업을 출력하시오.

```
Select ename, sal, job
```

```

From emp
Where sal not in (select sal
                  From emp
                  Where job = 'SALESMAN');

```

<input type="checkbox"/>	ENAME	SAL	JOB
1	SMITH	800	CLERK
2	SCOTT	3000	ANALYST
3	FORD	3000	ANALYST
4	BLAKE	2850	MANAGER
5	CLARK	2450	MANAGER
6	KING	5000	PRESIDENT
7	ADAMS	1100	CLERK
8	JAMES	950	CLERK
9	JONES	2975	MANAGER
10	MILLER	1300	CLERK

문제253. 관리자 번호만 출력하시오.

```

Select mgr
From emp;

```

문제254. 위의 결과를 중복 제거 하시오.

```

Select distinct mgr
From emp;

```

<input type="checkbox"/>	MGR
1	(null)
2	7839
3	7782
4	7698
5	7902
6	7566
7	7788

문제255. 관리자인 직원들의 이름과 월급을 출력하시오.

```

Select ename, sal
From emp
Where empno in (select mgr
                From emp);

```

<input type="checkbox"/>	ENAME	SAL
1	KING	5000
2	BLAKE	2850
3	JONES	2975
4	FORD	3000
5	SCOTT	3000
6	CLARK	2450

## 11-3. Multiple column subquery : 서브쿼리에서 쓰여지는 컬럼의 개수가 2개 이상인 경우의 서브쿼리

문제257. 직업이 SALESMAN인 직원들과 월급이 같고 커미션도 같은 직원들의 이름과 월급과 직업을 출력.

```

Select ename, sal, job
From emp
Where sal in (select sal

```

```

From emp
WHERE Job = 'SALESMAN')
and comm IN (select comm
From EMP
WHERE Job = 'SALESMAN');

```

	ENAME	SAL	JOB
1	WARD	1250	SALESMAN
2	MARTIN	1250	SALESMAN
3	ALLEN	1600	SALESMAN
4	TURNER	1500	SALESMAN

문제258. 통계학인 학생들과 통신사도 같고, 나이도 같은 학생들의 이름과 나이, 통신사, 전공을 출력.

```

Select ename, age, telecom, major
From emp2
Where telecom in (select telecom
From emp2
Where major = '통계학')

And age in (select age
From emp2
Where major = '통계학');

```

	ENAME	AGE	TELECOM	MAJOR
1	이근호	28	cjh	통계학
2	윤진민	27	sk	경제학
3	김대경	27	sk	통계학
4	김동윤	28	sk	경영학
5	김건태	28	sk	환경공학

코드 줄이는 방법

```

Select ename, age, telecom, major
From emp2
Where (telecom, age) in (select telecom, age
From emp2
Where major = '통계학');

```

	ENAME	AGE	TELECOM	MAJOR
1	이근호	28	cjh	통계학
2	김대경	27	sk	통계학
3	윤진민	27	sk	경제학

## • Multiple column subquery 종류 2가지

1. Non pair wise 방식 : cjh 이면서 28살, cjh 이면서 27살, sk 이면서 28살, sk 이면서 27살

Ex)

```

Select ename, age, telecom, major
From emp2
Where telecom in (select telecom
From emp2
Where major = '통계학')

And age in (select age
From emp2
Where major = '통계학');

```

2. Pair wise 방식 : cjh 이면서 28살, sk 이면서 27살

Ex)

```

Select ename, age, telecom, major
From emp2
Where (telecom, age) in (select telecom, age

```

```

From emp2
Where major = '통계학');

```

Oracle은 둘다 지원하는데 Mssql 은 pairwise 방식을 지원 안함

문제259. 직업, 이름, 입사일, 순위를 출력하는데 각 직업별로 가장 먼저 입사한 직원들만 출력하시오.

```

Select *
  From (SELECT job, ename, hiredate,
               RANK() OVER (PARTITION BY job ORDER BY hiredate) 순위
        FROM EMP
       )
 WHERE 순위 = 1;

```

	JOB	ENAME	HIREDATE	순위
1	ANALYST	FORD	1981-12-11 오전 12:00:00	1
2	CLERK	SMITH	1980-12-09 오전 12:00:00	1
3	MANAGER	JONES	1981-04-01 오전 12:00:00	1
4	PRESIDENT	KING	1981-11-17 오전 12:00:00	1
5	SALESMAN	ALLEN	1981-02-11 오전 12:00:00	1

## Multiple row subquery 연산자

문제260. 직업이 SALESMAN 인 직원들의 최대월급보다 더 많은 월급을 받는 직원들의 이름과 월급을 출력.

```

SELECT ename, sal
  FROM EMP
 WHERE sal > (select max(sal)
              FROM EMP
              WHERE job = 'SALESMAN');

```

	ENAME	SAL
1	KING	5000
2	BLAKE	2850
3	CLARK	2450
4	JONES	2975
5	FORD	3000
6	SCOTT	3000

```

SELECT ename, sal
  FROM EMP
 WHERE sal >all (select sal
                 FROM EMP
                 WHERE job = 'SALESMAN');

```

	ENAME	SAL
1	CLARK	2450
2	BLAKE	2850
3	JONES	2975
4	FORD	3000
5	SCOTT	3000
6	KING	5000

All을 쓰면 main 쿼리가 먼저 수행됨 (main쿼리먼저보다 서브 쿼리를 먼저 실행하는게 더 빠름)

문제261. 직업이 SALESMAN인 직원중에 가장 적은 월급보다 더 많은 월급을 받는 직원의 이름,월급을 출력

```

SELECT ename, sal
  FROM EMP
 WHERE sal >ALL (SELECT min(sal)

```

FROM EMP  
WHERE job = 'SALESMAN');

	ENAME	SAL
1	KING	5000
2	BLAKE	2850
3	CLARK	2450
4	JONES	2975
5	ALLEN	1600
6	TURNER	1500
7	FORD	3000
8	SCOTT	3000
9	MILLER	1300

SELECT ename, sal  
FROM EMP  
WHERE sal > any (SELECT sal  
FROM EMP  
WHERE job = 'SALESMAN');

	ENAME	SAL
1	KING	5000
2	SCOTT	3000
3	FORD	3000
4	JONES	2975
5	BLAKE	2850
6	CLARK	2450
7	ALLEN	1600
8	TURNER	1500
9	MILLER	1300

## 11-4. 서브쿼리에서 null 값

문제256. 관리자가 아닌 직원들의 이름과 월급을 출력하시오.

Select ename, sal  
From emp  
Where empno not in (select mgr  
From emp);      가 안되는 이유

Select ename, sal  
From emp  
Where empno not in  
(select NVL(mgr,0) In : =any 랑 같음 empno = (7839, 7566, ....., null)는  
From emp);      Empno = (7839 or 7566, ....., or null)이  
다.  
T or N = T이다.

	ENAME	SAL
1	TURNER	1500
2	WARD	1250
3	MARTIN	1250
4	ALLEN	1600
5	MILLER	1300
6	SMITH	800
7	ADAMS	1100
8	JAMES	950

Not in : !=any 는 empno = (7839, 7566, ....., null)는  
Empno = (7839 and ....., and null)이라서  
T and T = T  
F or F = F 이다.

★★★Subquery에서 not in 사용할 때 주의사항!!!!!!!★★★  
Null 처리를 해줘야한다.



## 12. Exists

2018년 4월 6일 금요일    오후 2:09

### Exists : 메인 쿼리에 존재하는 데이터가 서브쿼리에도 존재하는지 찾아보는 SQL

- 부서 테이블을 출력하시오.

Select \* from dept;

	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	CHICAGO
4	40	OPERATIONS	BOSTON

- 사원테이블의 부서번호를 조회하는데 부서테이블에는 존재하는데 사원 테이블에 없는 부서번호가 무엇인지 확인하시오.

SELECT deptno  
FROM EMP;

	DEPTNO
1	(null)
2	30
3	(null)
4	20
5	30
6	30
7	30
8	30
9	30
10	20
11	20
12	20
13	20
14	(null)

- 부서 테이블에서 부서번호를 출력하는데 사원테이블에 존재하는 부서번호만 출력하시오.

SELECT deptno  
FROM dept  
WHERE DEPTno IN (SELECT deptno FROM EMP);

	DEPTNO
1	30
2	20

- 부서 테이블에서 부서번호를 출력하는데 사원테이블에 존재하지 않는 부서번호만 출력하시오.

SELECT deptno  
FROM DEPT  
WHERE deptno NOT IN(SELECT NVL(deptno, 0) FROM emp);

	DEPTNO
1	40
2	10



**문제262. 아래의 SQL을 튜닝하시오.**

```
SELECT deptno
FROM dept
WHERE DEPTno IN (SELECT deptno FROM EMP);
```

튜닝후 :

```
Select deptno
From dept d
Where exists (select 'x'
              From emp e
              Where e.deptno = d.deptno);
```

x : 아무 단어써도됨.

DEPTNO	
1	30
2	20

==> 메인쿼리부터 실행하는데 in 보다 더 빠름 exists는 찾다가 존재하면 멈추고 다른 것을 수행함.(distinct도 가능한데 distinct는 데이터를 정렬하고 하는거라 느림)

**문제263. 이번에는 존재하지 않는 부서번호를 출력하시오.**

```
Select deptno
From dept d
Where NOT exists (select 'x'
                  From emp e
                  Where e.deptno = d.deptno);
```

DEPTNO	
1	40
2	10

**문제264. telecom\_price 테이블의 통신사를 출력하는데, 우리반 emp2에 존재하는 통신사만 출력하시오.**

```
SELECT telecom
FROM TELECOM_PRICE tp
WHERE EXISTS (SELECT 'x'
              FROM EMP2 e2
              WHERE e2.TELECOM = tp.telecom);
```

TELECOM	
1	sk
2	lg
3	kt

**문제265. telecom\_price 테이블의 통신사를 출력하는데, 존재하지 않는 통신사만 출력하시오.**

```
SELECT telecom
FROM TELECOM_PRICE tp
WHERE not EXISTS (SELECT 'x'
                  FROM EMP2 e2
                  WHERE e2.TELECOM = tp.telecom);
```

표시할 데이터가 없습니다.

## 13. DML 문 insert, update, delete, merge

2018년 4월 6일 금요일    오후 3:08

### 데이터 조작언어 (DML)

1. Insert
2. Update
3. Delete
4. 서브쿼리 를 이용한 DML
5. Merge --> insert, update, delete 를 한번에 수행하는 명령어

### 13-1. Insert문

<--맞춰줘야함

Ex) insert into emp (empno, ename, sal)  
Values(1234, 'JACK', 4000);

14	7934	MILLER	CLERK	7782	1982-01-11 오전 12:00:00	1300	(null)	(null)
15	1234	JACK	(null)	(null)	(null)	4000	(null)	(null)

commit 하지않고 취소를 하고 싶으면 rollback; 이라고 하면됨

문제266. 아래의 data를 입력하시오.

사원번호 3282

사원이름 JANE

월급 4500

입사일 오늘날짜

부서번호 20

INSERT INTO EMP (empno, ename, sal, hiredate, deptno)

VALUES(3282, 'JANE', 4500, sysdate, 20); <-- sysdate : 보다는 오늘날짜를 써야됨

시,분,초 가 나와서 검색할때 안나옴

INSERT INTO EMP (empno, ename, sal, hiredate, deptno)

VALUES(3282, 'JANE', 4500, TO\_DATE('2018/04/06','RRRR/MM/DD'), 20);

문제267. 오늘 입사한 사원의 이름과 입사일을 출력하시오.

SELECT ename, hiredate

FROM EMP

WHERE hiredate = TO\_DATE('2018/04/06','RRRR/MM/DD');

### • Null 값을 입력하는 방법

#### 1. 명시적으로 입력하는 방법

- Null
- "

#### 2. 암시적으로 입력하는 방법

- Insert into emp(empno, ename, sal) <-- 입력안한 컬럼들은 null로 들어감

Values(2911, 'aaa', 3400)

명시적 ex)

```
Insert into emp(empno, ename, sal)
Values(2911, null, 3400);
```

```
Insert into emp(empno, ename, sal)
Values(2945, '', 3500);
```

```
Insert into emp(empno, ename, sal)
Values(3924, ' ', 3600);
```

문제268. 이름이 null 이 아닌 직원들의 이름과 월급을 출력하는데 공백문자도 안나오게 하시오.

```
SELECT ename, sal
FROM EMP
WHERE TRIM(ename) IS not NULL;
```

	ENAME	SAL
1	KING	5000
2	BLAKE	2850
3	CLARK	2450
4	JONES	2975
5	MARTIN	1250
6	ALLEN	1600
7	TURNER	1500
8	JAMES	950
9	WARD	1250
10	FORD	3000
11	SMITH	800
12	SCOTT	3000
13	ADAMS	1100
14	MILLER	1300

## 13-2. Update

Ex) update emp

Set sal = 8000

Where ename = 'SCOTT';

==> 이름이 SCOTT인 직원의 월급을 8000으로 변경하겠다.

문제269. 직업이 SALESMAN인 직원들의 커미션을 5000으로 변경하시오.

```
UPDATE EMP
SET COMm = 5000
WHERE job = 'SALESMAN';
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	
1	7839	KING	PRESIDENT	(null)	1981-11-17 오전 12:00:00	5000	(null)	(r
2	7698	BLAKE	MANAGER	7839	1981-05-01 오전 12:00:00	2850	(null)	
3	7782	CLARK	MANAGER	7839	1981-05-09 오전 12:00:00	2450	(null)	(r
4	7566	JONES	MANAGER	7839	1981-04-01 오전 12:00:00	2975	(null)	
5	7654	MARTIN	SALESMAN	7698	1981-09-10 오전 12:00:00	1250	5000	
6	7499	ALLEN	SALESMAN	7698	1981-02-11 오전 12:00:00	1600	5000	
7	7844	TURNER	SALESMAN	7698	1981-08-21 오전 12:00:00	1500	5000	
8	7900	JAMES	CLERK	7698	1981-12-11 오전 12:00:00	950	(null)	
9	7521	WARD	SALESMAN	7698	1981-02-23 오전 12:00:00	1250	5000	

문제270. 부서번호가 30번인 직원들의 월급을 0 으로 변경하시오.

```
UPDATE EMP
SET sal = 0
WHERE deptno = 30;
```

5	7654	MARTIN	SALESMAN	7698	1981-09-10 오전 12:00:00	0	1400	30
6	7499	ALLEN	SALESMAN	7698	1981-02-11 오전 12:00:00	0	300	30
7	7844	TURNER	SALESMAN	7698	1981-08-21 오전 12:00:00	0	0	30
8	7900	JAMES	CLERK	7698	1981-12-11 오전 12:00:00	0 (null)		30
9	7521	WARD	SALESMAN	7698	1981-02-23 오전 12:00:00	0	500	30

%%만약 롤백안하고 커밋하면 되돌아가는 방법(15분안에해야됨)

```
Alter table emp enable row movement;
Flashback table emp to timestamp
To_timestamp('2018/04/06:15:40:00',
RRRR/MM/DD:HH24",MI:SS');
```

문제271. 월급이 3000 이하인 직원들의 커미션을 500으로 수정하시오.

```
UPDATE EMP
SET COMM = 500
WHERE sal <= 3000;
```

## 13-3. delete문

문제272. 직업이 SALESMAN 인 직원들을 삭제하시오.

```
Delete from emp
Where job = 'SALESMAN';
```

문제273. 월급이 3000이상인 직원들을 지우시오.

```
DELETE FROM EMP
WHERE sal >= 3000;
```

- 실수로 테이블을 지우고 exit 하면 저장하고 나가짐  
Exit : 커밋하고 나가기

### 13-4-1. 서브쿼리 delete

문제274. SCOTT 보다 많은 월급을 받는 직원들을 지우시오.

```
DELETE FROM EMP
WHERE sal > (SELECT sal
FROM EMP
WHERE ename = 'SCOTT');
```

문제275. ALLEN 보다 늦게 입사한 직원들을 삭제하시오.

```
DELETE FROM EMP
WHERE hiredate > (SELECT hiredate
FROM EMP
WHERE ename = 'ALLEN');
```

### 13-4-2. 서브쿼리 UPDATE

문제276. JONES 보다 많은 월급을 받는 직원들의 직업을 SALESMAN으로 변경하시오.

```
UPDATE EMP
  SET job = 'SALESMAN'
  WHERE sal > (SELECT sal
               FROM EMP
               WHERE ename = 'JONES');
```

문제277. JONES 보다 많은 월급을 받는 직원들의 직업을 KING의 직업으로 변경하시오.

```
UPDATE EMP
  SET job = (SELECT job
             FROM EMP
             WHERE ename = 'KING')
  WHERE sal > (SELECT sal
               FROM EMP
               WHERE ename = 'JONES');
```

문제278. 20번 부서번호인 직원들의 월급을 KING의 월급으로 변경하시오.

```
UPDATE EMP
  SET sal = (SELECT sal
             FROM EMP
             WHERE ename = 'KING')
  WHERE deptno = 20;
```

### 13-4-3. 서브쿼리 Insert

- Emp 테이블과 똑같은 구조의 테이블을 emp6 으로 생성하시오.

```
Create table emp6
As
Select *
From emp
Where 1 = 2;
```

<- 1 = 2 는 F여서 데이터는 안가져오고 테이블 구조만 가져온다.

문제279. emp 테이블의 데이터를 emp6에 입력하시오.

```
INSERT INTO emp6
  SELECT *
  FROM EMP;
```

문제280. 반장 자리의 emp2 테이블의 데이터를 나의 emp20 테이블에 입력하시오.

% 내가 가지고 있는 db link가 무엇이 있는지 조회

```
Select db_link
From dba_db_links;
```

	DB_LINK
1	DBMS_CLRDBLINK
2	DBLINK7
3	DBLINK8
4	OUR_CLASS_LINK
5	OUR_CLASS_LINK

```
Create table emp20
```

```
As
Select *
From emp2@dblink7;    <-- 반장 자리의 emp2테이블을 그대로 내자리의 emp20으로 생성
```

문제281. 반장 자리의 emp2데이터를 내 자리 emp20에 입력 하시오.

```
INSERT INTO emp20
SELECT *
FROM emp2@dblink7;
```

## 13-5. Merge 문

: insert, update, delete를 한번에 수행하는 명령어

Merge 문을 이용안하고 data 업데이트 하는방법

- Emp 테이블의 loc 컬럼을 추가하시오.

```
ALTER TABLE EMP
ADD loc VARCHAR2(20);
```

Update 문으로 emp 테이블에 loc 컬럼을 해당 사원의 부서위치로 값을 갱신하시오.

(하나씩 하나씩 결과를 변경해서 넘김 => 속도 저하)

```
UPDATE EMP e
SET loc = (SELECT LOC
           FROM DEPT d
           WHERE d.deptno = e.DEPTNO);
```

Emp 테이블의 크기를 백만건이상으로 생성

```
Insert into emp
Select *
From emp;
```

가로 그리드 1835008 개 행이 만들어졌습니다.

- emp 테이블에 loc 컬럼을 추가한다.

```
Alter table emp
Add loc varchar2(20);
```

- emp 테이블에 loc 컬럼의 데이터를 dept 테이블의 loc 컬럼의 데이터로 갱신하시오.

```
Update emp e
Set loc = (select loc
           From dept d
           Where d.deptno =e.deptno);
```

이러면 엄청오래걸림

튜닝을 한다.

```
Alter table emp
Add loc varchar2(20);
```

```
Merge into emp e
Using dept d
On (e.deptno = d.deptno)
When matched then
```

Update set e.loc = d.loc;

튜닝하면 훨씬 빨라진다.

% emp 테이블에 loc(부서위치) 컬럼을 추가하시오.

```
Alter table emp  
add loc varchar2(20);
```

	ENAME	LOC
1	KING	(null)
2	BLAKE	(null)
3	CLARK	(null)
4	JONES	(null)
5	MARTIN	(null)
6	ALLEN	(null)
7	TURNER	(null)
8	JAMES	(null)
9	WARD	(null)
10	FORD	(null)
11	SMITH	(null)
12	SCOTT	(null)
13	ADAMS	(null)
14	MILLER	(null)

% 사원 테이블과 부서테이블을 조인해서 사원이름과 부서위치를 출력하시오.

```
SELECT e.ename, d.loc  
FROM EMP e, DEPT d  
WHERE e.DEPTNO = d.DEPTNO;
```

조인을 사용하지 않고 한꺼번에 가져 올 수 있는게 **merge**이다.

```
Ex) merge into emp e  
Using dept d  
On (e.deptno = d.deptno)  
When matched then  
Update set e.loc = d.loc;
```

	ENAME	LOC
1	KING	NEW YORK
2	BLAKE	CHICAGO
3	CLARK	NEW YORK
4	JONES	DALLAS
5	MARTIN	CHICAGO
6	ALLEN	CHICAGO
7	TURNER	CHICAGO
8	JAMES	CHICAGO
9	WARD	CHICAGO
10	FORD	DALLAS
11	SMITH	DALLAS
12	SCOTT	DALLAS
13	ADAMS	DALLAS
14	MILLER	NEW YORK

문제282. 사원 테이블에 dname(부서명) 컬럼을 추가하고 dept 테이블의 dname으로 값을 변경하시오.

```
Alter table emp
```

```
add dname varchar2(20);
```

```
MERGE INTO EMP e
  USING DEPT d
  ON (e.deptno = d.deptno)
  WHEN matched THEN
    UPDATE SET e.dname = d.dname;
```

	DNAME	DNAME_1
1	ACCOUNTING	ACCOUNTING
2	SALES	SALES
3	ACCOUNTING	ACCOUNTING
4	RESEARCH	RESEARCH
5	SALES	SALES
6	SALES	SALES
7	SALES	SALES
8	SALES	SALES
9	SALES	SALES
10	RESEARCH	RESEARCH
11	RESEARCH	RESEARCH
12	RESEARCH	RESEARCH
13	RESEARCH	RESEARCH
14	ACCOUNTING	ACCOUNTING

%% emp 테이블의 loc 컬럼을 삭제

```
ALTER TABLE EMP
  DROP COLUMN loc;
```

%% emp 테이블의 dname 컬럼을 삭제

```
ALTER TABLE EMP
  DROP COLUMN dname;
```

**문제283. 사원 테이블에 grade 라는 컬럼을 추가하시오. (하기전에 284번 먼저)**

```
Alter table emp
  Add grade number(10);
```

**문제284. 사원이름, 월급, 급여등급을 출력하시오.(emp와 salgrade 조인)**

```
SELECT e.ename, e.sal, s.grade
  FROM EMP e, SALGRADE s
 WHERE e.sal BETWEEN s.losal AND s.hisal;
```

	ENAME	SAL	GRADE
1	JAMES	950	1
2	SMITH	800	1
3	ADAMS	1100	1
4	MARTIN	1250	2
5	WARD	1250	2
6	MILLER	1300	2
7	ALLEN	1600	3
8	TURNER	1500	3
9	BLAKE	2850	4
10	CLARK	2450	4
11	JONES	2975	4
12	FORD	3000	4
13	SCOTT	3000	4
14	KING	5000	5



**문제285. merge 문을 이용해서 emp 테이블의 grade를 해당 사원의 급여등급으로 값을 갱신.**

```

MERGE INTO EMP e
  USING SALGRADE s
  ON( e.sal BETWEEN s.losal AND s.hisal)
  WHEN matched THEN
    UPDATE SET e.grade = s.grade;

```

	ENAME	GRADE
1	KING	5
2	BLAKE	4
3	CLARK	4
4	JONES	4
5	MARTIN	2
6	ALLEN	3
7	TURNER	3
8	JAMES	1
9	WARD	2
10	FORD	4
11	SMITH	1
12	SCOTT	4
13	ADAMS	1
14	MILLER	2

%% emp 테이블의 grade 컬럼을 삭제하시오.

```

ALTER TABLE EMP
  DROP column grade;

```

**문제286. 부서번호, 부서번호별 토탈월급을 출력하시오.**

```

SELECT deptno, SUM(sal)
  FROM EMP
  GROUP by deptno;

```

	DEPTNO	SUM(SAL)
1	30	9400
2	20	10875
3	10	8750

%% dept 테이블의 sumsal 컬럼을 추가

```

Alter table dept
  Add sumsal number(10);

```

**문제287. dept 테이블에 sumsal 컬럼에 해당 부서번호 토탈 월급으로 값을 갱신하시오.**

```

MERGE INTO DEPT d
  USING (SELECT deptno, SUM(sal) sumsal FROM EMP GROUP BY deptno) e
  ON(e.deptno = d.deptno)
  WHEN matched THEN
    UPDATE SET d.sumsal = e.sumsal;

```

	DEPTNO	SUMSAL
1	10	8750
2	20	10875
3	30	9400
4	40	(null)



## 14. Truncate 문

2018년 4월 9일 월요일      오후 1:50

### Truncate 문 : 데이터를 삭제하는 SQL

- Data를 삭제하는 SQL 3가지

1. Delete from emp;
2. Truncate table emp;
3. Drop table emp;

### 중요!!!!!!

	Delete	truncate	drop
1 . Date	삭제	삭제	삭제
2 . 저장공간	유지	삭제	삭제
3 . 저장구조	유지	유지	삭제
4 . Rollback	가능	불가능	불가능
5 . Flashback	가능	불가능	가능

## 15. 트랜잭션

2018년 4월 9일 월요일    오후 2:08

**트랜잭션** : insert, update, delete, merge 문을 수행하기 시작했으면 하나의

트랜잭션이 시작된 것이다.

**Commit, rollback** 을 수행하면 하나의 트랜잭션이 종료가 된다.

암시적 commit

1. 정상 종료 (exit)
2. DDL 문장 수행      (create, alter, drop, truncate, rename)
3. DCL 문장 수행      (grant, revoke)

암시적 rollback

1. 시스템 정전 (비정상 종료)
2. 비정상적으로 sqlplus 같은 클라이언트 툴이 꺼졌을 때.

DDL 명령어 : create, alter, drop, truncate, rename)

테이블 생성 구분

```
Create table emp77
(empno number(10),
ename varchar(20),
sal number(10),
hiredate date);
```

```
Insert into (empno, ename, sal, hiredate)
Values(1, '원섭', 3333, 18/04/09);
```

```
Insert into (empno, ename, sal, hiredate)
Values(2, '백광', 2222, 18/04/09);
```

**문제289.** 아래의 데이터를 담을 수 있는 테이블을 생성하고, 데이터를 입력하시오.

이름: 김원섭

나이: 26

생일: 1994/02/08

전공: 컴퓨터공학

주소: 경기도 남양주시 와부읍

---

```
CREATE TABLE emp99
(ename varchar2(10),
age number(5),
birthday DATE,
```

```
major varchar2(20),
address varchar2(50));
```

	ENAME	AGE	BIRTHDAY	MAJOR	ADDRESS
1	김원섭	26	1994-02-08 오전 12:00:00	컴퓨터공학	경기도 남양주시 와부읍

## %% 컬럼의 길이를 늘리는 명령어

```
Alter table emp99
Modify address varchar2(100);
```

Desc emp99;

	COLUMN	NULL?	WIDTH
1	ENAME	(null)	VARCHAR2(10)
2	AGE	(null)	NUMBER(5)
3	BIRTHDAY	(null)	DATE
4	MAJOR	(null)	VARCHAR2(20)
5	ADDRESS	(null)	VARCHAR2(100)

## %% 컬럼 추가 명령어

```
Alter table emp99
Add email varchar2(50);
```

문제290. emp99 테이블에 **mobile** 이라는 컬럼을 추가하고 자신의 핸드폰 번호로 업데이트 하시오.

```
ALTER TABLE emp99
ADD mobile varchar2(20);
```

```
update emp99
SET mobile = '010-3308-0492'
WHERE ename = '김원섭';
```

	ENAME	AGE	BIRTHDAY	MAJOR	ADDRESS	EMAIL	MOBILE
1	김원섭	26	1994-02-08 오전 12:00:00	컴퓨터공학	경기도 남양주시 와부읍	(null)	010-3308-0492

## %% 컬럼 삭제

```
Alter table emp99
Drop column mobile;
```

	ENAME	AGE	BIRTHDAY	MAJOR	ADDRESS	EMAIL
1	김원섭	26	1994-02-08 오전 12:00:00	컴퓨터공학	경기도 남양주시 와부읍	(null)

## %% 컬럼 감추기

```
Alter table emp
Set unused column sal;
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	COMM	DEPTNO
1	7839	KING	PRESIDENT	(null)	1981-11-17 오전 12:00:00	(null)	10
2	7698	BLAKE	MANAGER	7839	1981-05-01 오전 12:00:00	(null)	30
3	7782	CLARK	MANAGER	7839	1981-05-09 오전 12:00:00	(null)	10
4	7566	JONES	MANAGER	7839	1981-04-01 오전 12:00:00	(null)	20
5	7654	MARTIN	SALESMAN	7698	1981-09-10 오전 12:00:00	1400	30
6	7499	ALLEN	SALESMAN	7698	1981-02-11 오전 12:00:00	300	30

## %% 감춰진 컬럼 drop 하는 방법

```
Alter table emp
```

## %% 테이블명 변경하는 명령어

RENAME EMP TO emp90;    <- 이전이름 to 변경할 이름

### %% 서브쿼리를 이용해서 테이블 생성하는 방법 (필요한 데이터, 컬럼만 가져와서 데이터 정제 할때 사용)

(ctas를 이용해서 테이블 생김)

```
CREATE TABLE emp90
AS
SELECT empno, ename, sal, deptno
FROM EMP;
```

문제291. 통신사, 통신사별 인원수를 출력하는 결과를 테이블로 생성하시오.(테이블명 : emp60)

```
CREATE TABLE emp60
AS
SELECT telecom, COUNT(*) AS "인원수"
FROM EMP2
GROUP BY telecom;
```

	TELECOM	인원수
1	sk	8
2	lg	9
3	qjh	1
4	kt	12

**Tip) ctas 구문에서 group함수를 사용하면 컬럼별칭을 사용해야한다.**

문제292. 아래의 결과를 emp500 으로 생성하시오.

```
CREATE TABLE emp500
AS
SELECT deptno AS "부서번호", SUM(sal) AS "토탈월급", AVG(sal) AS "평균월급",
COUNT(*) AS "인원수", MAX(sal) AS "최대월급"
FROM EMP
GROUP BY deptno;
```

구	부서번호	토달월급	평균월급	인원수	최대월급
1	30	9400	1566.6666666666666666666666666667	6	2850
2	20	10875	2175	5	3000
3	10	8750	2916.6666666666666666666666666667	3	5000

문제293. emp 테이블의 월급의 변화가 emp500 테이블에 토탈월급, 평균월급, 최대월급에 반영되게 하시오.

```
MERGE INTO emp500 e5
    USING (SELECT deptno, SUM(sal) 토탈월급,
                AVG(sal) 평균월급,
                MAX(sal) 최대월급
            FROM EMP
            GROUP BY deptno) e
    ON(e5.부서번호 = e.deptno)
```

```
UPDATE SET E5.토탈월급 = E.토탈월급,  
        E5.평균월급 = E.평균월급,  
        E5.최대월급 = E.최대월급;
```

문제294. 우리반 테이블에 birth\_day라는 컬럼을 추가하고, 그 학생이 태어난 요일로 값을 갱신하시오.

ADD birth\_day VARCHAR2(20); <-- 요일로 바꾸는 함수가 to\_char라서 date쓰면안  
됨.

```
UPDATE SET birth_day = TO_CHAR(birth, 'day');
```

SQL 활용 페이지 111

## 16. Rock 이란 무엇인가?

2018년 4월 9일 월요일    오후 4:00

**Rock : update를 할때 그 update하는 행을 잠그는 기능  
(다른 컬럼 변경해도 잠김, 행 단위로 걸림)**

1. Cmd를 두개 접속
2. 한쪽에서 update 명령을 하고 다른 한쪽에서 update를 하면 잠금.

푸는 방법

- Commit 하거나 rollback
- Sql 게이트에서 관리자 -> 세션 브라우저 -> 차단 잠금에서 오른쪽클릭후 종료



# 17. 제약

2018년 4월 16일 월요일 오전 10:05

## 제약이 필요한 이유?

- 데이터의 품질을 높이기 위해서

Ex) 중복된 data 입력 안되게 막는다.

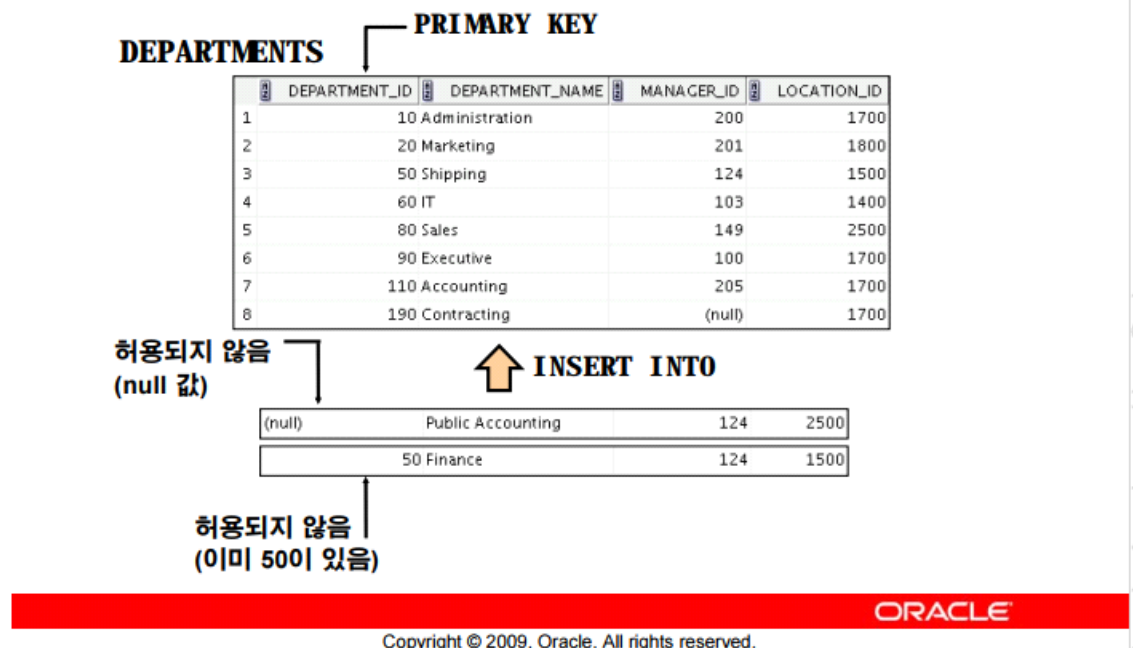
이메일 입력할 때 이메일 형식에 맞춰서 입력되게끔 제약을 줄 수 있다.

## 제약의 종류

1. Primary key : 중복된 data, null 값이 입력 안되게
2. Unique : 중복된 data 입력 안되게
3. Not null : null 값을 입력 안되게
4. Check : 미리 정의된 데이터만 입력되게
5. Foreign key : 참조 하는 컬럼에 거는 제약

## 17-1. primary key : null 값과 중복된 데이터가 입력이 안되게 하는 제약

### PRIMARY KEY 제약 조건



Ex) create table emp10

(empno number(10) primary key, ename varchar2(20));

INSERT INTO emp10 VALUES(1111, 'SCOTT'); <--- 입력됨

INSERT INTO EMP10 VALUES(NULL, 'SMITH'); <--- 입력 안됨

INSERT INTO emp10 VALUES(1111, 'ALLEN'); <--- 입력 안됨

**문제295. 만들어진 emp2 테이블 empno에 primary key 제약을 거시오.**

```
Alter table emp2
Add constraint emp2_empno_pk
primary key (empno);

Insert into emp2(empno, ename)
Values(6, '김지우'); <-- 이미있는거라 안들어감
```

**문제296. emp 테이블에 empno에 primary key 제약을 거시오.**

```
Alter table emp
Add constraint emp_empno_pk
Primary key(empno);
```

**%% emp2 테이블에 걸린 제약이 무엇인지 확인하는 방법**

```
Select table_name, constraint_name, constraint_type
From user_constraints
Where table_name = 'EMP2';
```

	TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE
1	EMP2	EMP2_EMPNO_PK	P

emp2 에 P(primary key)가 걸려있다.

**문제297. emp테이블에 어떠한 제약이 걸려 있는지 확인하시오.**

```
Select table_name, constraint_name, constraint_type
From user_constraints
Where table_name = 'EMP';
```

	TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE
1	EMP	SYS_C007014	C

emp에는 C(not null), P(primary key)가 걸려있다.

**%% 제약 삭제**

**문제298. emp 테이블에 empno에 걸린 primary key 제약을 삭제하시오.**

```
Alter table emp
Drop constraint emp_empno_pk; 제약이름
```

	TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE
1	EMP	SYS_C007014	C

**문제299. emp2 테이블에 걸린 primary 제약을 삭제하시오.**

```
ALTER TABLE EMP2
DROP CONSTRAINT emp2_empno_pk;
```

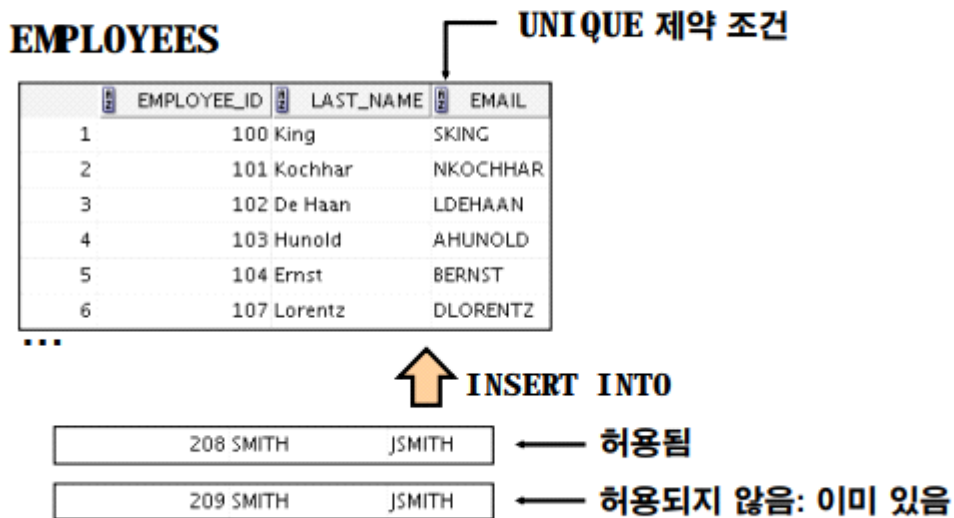
**문제300. dept 테이블의 deptno에 primary key제약을 거시오.**

```
ALTER TABLE dept
ADD CONSTRAINT dept_deptno_pk
PRIMARY KEY(deptno);
```

	TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE
1	DEPT	DEPT_DEPTNO_PK	P

**17-2. unique 제약 : 중복된 데이터를 허용하지 않겠다.**  
**(null(알 수 없는 값)은 중복해서 들어감)**

## UNIQUE 제약 조건



Ex) 테이블 생성시 unique 제약을 거는 방법

```
Create table emp15
(empno number(10),
ename varchar2(20) unique,
sal number(10) );
```

Insert into emp15 values(1111, 'SCOTT', 3000); <-- 입력됨

Insert into emp15 values(2222, 'SCOTT', 4000); <-- 입력안됨

**문제301. emp15 테이블에 걸린 제약을 확인하시오.**

```
Select table_name, constraint_name, constraint_type
From user_constraints
Where table_name = 'EMP15';
```

	TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE
1	EMP15	SYS_C007019	U

U(unique)

**문제302. emp16 테이블을 생성하는데 emp15와 똑같이 생성하고 제약이름을 주고 만드시오**

오.

```
Create table emp16
( empno number(10),
  ename varchar2(10) constraint emp16_ename_un unique, sal number(10) );
```

	TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE
1	EMP16	EMP16_ENAME_UN	U

문제303. emp16 테이블에 걸린 제약을 삭제하고 emp15 테이블에 걸린 제약도 삭제하시오.

```
ALTER TABLE emp15
  DROP CONSTRAINT SYS_C007019;
```

```
ALTER TABLE emp16
  DROP CONSTRAINT emp16_ename_un;
```

문제304. emp2 테이블에 ename에 unique 제약을 거시오.

```
ALTER TABLE EMP2
  ADD CONSTRAINT emp2_ename_un
    UNIQUE(ename);
```

	TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE
1	EMP2	EMP2_ENAME_UN	U

문제305. emp2 테이블에 empno에 unique 제약을 거시오.

```
ALTER TABLE EMP2
  ADD CONSTRAINT emp2_empno_un
    UNIQUE(empno);
```

	TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE
1	EMP2	EMP2_ENAME_UN	U
2	EMP2	EMP2_EMPNO_UN	U

## 17-3. not null : null 값이 입력 안되게 하는 제약

# NOT NULL 제약 조건

열에 null 값이 허용되지 않도록 보장합니다.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	COMMISSION_PCT	DEPARTMENT_ID	EMAIL	PHONE_NUMBER	HIRE_DATE
100	Steven	King	24000	(null)	90	SKING	515.123.4567	17-JUN-87
101	Neena	Kochhar	17000	(null)	90	NKOCHHAR	515.123.4568	21-SEP-89
102	Lex	De Haan	17000	(null)	90	LDEHAAN	515.123.4569	13-JAN-93
103	Alexander	Hunold	9000	(null)	60	AHUNOLD	590.423.4567	03-JAN-90
104	Bruce	Ernst	6000	(null)	60	BERNST	590.423.4568	21-MAY-91
107	Diana	Lorentz	4200	(null)	60	DLORENTZ	590.423.5567	07-FEB-99
124	Kevin	Mourgos	5800	(null)	50	KMOURGOS	650.123.5234	16-NOV-99
141	Trenna	Rajs	3500	(null)	50	TRAJS	650.121.8009	17-OCT-95
142	Curtis	Davies	3100	(null)	50	CDAVIES	650.121.2994	29-JAN-97
143	Randall	Matos	2600	(null)	50	RMATOS	650.121.2874	15-MAR-98
144	Peter	Vargas	2500	(null)	50	PVARGAS	650.121.2004	09-JUL-98
149	Eleni	Zlotkey	10500	0.2	80	EZLOTKEY	011.44.1344.429018	29-JAN-00
174	Ellen	Abel	11000	0.3	80	EABEL	011.44.1644.429267	11-MAY-96
176	Jonathon	Taylor	8600	0.2	80	JTAYLOR	011.44.1644.429265	24-MAR-98
178	Kimberely	Grant	7000	0.15	(null)	KGRANT	011.44.1644.429263	24-MAY-99
200	Jennifer	Whalen	4400	(null)	10	JWHALEN	515.123.4444	17-SEP-87
201	Michael	Hartstein	13000	(null)	20	MHARTSTE	515.123.5555	17-FEB-96
202	Pat	Fay	6000	(null)	20	PFAY	603.123.6666	17-AUG-97
205	Shelley	Higgins	12000	(null)	110	SHIGGINS	515.123.8080	07-JUN-94
206	William	Gietz	8300	(null)	110	WGIEZT	515.123.8181	07-JUN-94

NOT NULL 제약 조건  
(Primary Key는 NOT NULL  
제약 조건을 적용합니다.)

NOT NULL  
제약 조건

NOT NULL 제약 조건 없음(모든 행에서 이  
열에 대해 null 값을 포함할 수 있습니다.)

Ex) 테이블 생성시 not null 제약 거는 방법

```
Create table emp19
(empno number(10),
ename varchar2(10) constraint emp19_ename_nn not null,
sal number(10) );
```

INSERT INTO emp19 VALUES(1234, NULL, 3400); <-- 입력 안됨

문제306. emp2 테이블에 major에 not null 제약을 거시오.

```
Alter table emp2
modify major constraint emp2_major_nn not null;
```

3	EMP2	EMP2_MAJOR_NN	C
---	------	---------------	---

문제307. emp2 테이블 아래의 컬럼들의 not null 제약을 거시오.

Ename, address, mobile

```
Alter table emp2
Modify ename constraint emp2_ename_nn not NULL;
ALTER TABLE EMP2
MODIFY address constraint emp2_address_nn not NULL;
ALTER TABLE EMP2
MODIFY mobile constraint emp2_mobile_nn not null;
```

	TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE
1	EMP2	EMP2_ENAME_UN	U
2	EMP2	EMP2_EMPNO_UN	U
3	EMP2	EMP2_MAJOR_NN	C
4	EMP2	EMP2_ADDRESS_NN	C
5	EMP2	EMP2_MOBILE_NN	C

문제308. emp2테이블에 걸린 모든 제약을 빠르게 삭제할수 있는 스크립트를 쿼리문으로 생성.

(정제 작업시 필요한 스크립트)

```
SELECT 'ALTER TABLE EMP2 DROP constraint ' || CONSTRAINT_name || ';'
FROM user_constraints
WHERE table_name = 'EMP2';
```

## 17-4. Check 제약 : 미리 정의한 형식에 맞는 데이터만 입력되게끔 거는 제약

Ex) 이메일에 @가 있어야지만 테이블에 입력될 수 있게 하려면 체크 제약을 사용해야한다.

문제309. 사원테이블에 부서번호가 10, 20, 30 번만 입력 되게끔 체크제약을 거시오.

```
ALTER TABLE EMP
ADD CONSTRAINT emp_deptno_ck
check( deptno IN (10, 20, 30) );
```

```
INSERT INTO EMP(empno, ename, sal, deptno)
VALUES(1234,'JACK', 3400, 20);
```

<-- 들어감

```
INSERT INTO EMP(empno, ename, sal, deptno)
VALUES(2345,'JANE', 4500, 70);
```

<-- 안들어감

문제310. 위의 체크제약을 삭제하고 다시 부서번호가 70번이 입력될 수 있도록 체크제약을 거시오.

```
ALTER TABLE EMP
DROP CONSTRAINT emp_deptno_ck;

ALTER TABLE EMP
ADD CONSTRAINT emp_deptno_ck
check( deptno IN (10, 20, 30, 70) );
```

13	7876	ADAMS	CLERK	7788	1983-01-15 오전 12:00:00	1100	(null)	20
14	7934	MILLER	CLERK	7782	1982-01-11 오전 12:00:00	1300	(null)	10
15	2345	JANE	(null)	(null)	(null)	4500	(null)	70
16	1234	JACK	(null)	(null)	(null)	3400	(null)	20

문제311. 사원테이블에 월급이 0에서부터 6000 사이에 데이터만 입력되게끔 check제약을 거시오.

```
ALTER TABLE EMP
ADD CONSTRAINT emp_sal_ck
check( sal BETWEEN 0 AND 6000 );
```

문제312. 우리반 테이블에 email에 @ 가 있어야만 이메일이 입력되거나 수정될 수 있도록

### Check 제약을 거시오.

```
ALTER TABLE EMP2
ADD CONSTRAINT emp2_email_ck
CHECK (email LIKE '%@%');
```

문제313. 이메일에 걸린 기존 체크제약 삭제하고, 다시 체크 제약을 거는데 이메일에 @와 .을 포함하고 있는 데이터만 입력 또는 수정될 수 있도록 제약을 거시오.

```
ALTER TABLE EMP2
drop CONSTRAINT emp2_email_ck;

ALTER TABLE EMP2
ADD CONSTRAINT emp2_email_ck
CHECK (email LIKE '%@%.%');
```

문제314. 우리반에 입력된 이름의 성씨만 입력될 수 있도록 체크제약을 거시오.

```
SELECT DISTINCT '|| substr(ename, 1,1) || '||',
FROM EMP2;
```

을 써서

	SUBSTR(ENAME,1,1)  ','
1	'장,'
2	'이,'
3	'도,'
4	'유,'
5	'방,'
6	'한,'
7	'백,'
8	'정,'
9	'차,'
10	'신,'
11	'송,'
12	'지,'
13	'김,'
14	'윤,'
15	'은,'

복사후,

```
ALTER TABLE EMP2
ADD CONSTRAINT emp2_ename_ck
check( substr(ename, 1,1) IN ('장',
'이',
'도',
'유',
'방',
'한',
'백',
'정',
'차',
'신',
'송',
'지',
```

```
'김',
'윤',
'은') );
```

붙여넣기하면됨

문제315. 아래의 테이블을 생성하고 주민번호를 입력할 때 아래와 같은 형식으로만 입력되게 하시오.

941221-\*\*\*\*\* ( 중요 ★★★★★★ )

```
ALTER TABLE emp315
  ADD CONSTRAINT emp15_verify_no_ck
    CHECK ( regexp_like(verify_no, '^([0-9]{6}-[*]{7})' ) );
INSERT INTO emp315 VALUES('SCOTT', '123456-*****');
```

	ENAME	VERIFY_NO
1	SCOTT	123456-*****

문제316. 부서테이블에 부서위치에 NEW YORK, DALLAS, CHICAGO, BOSTON 만 입력될 수 있도록

체크 제약을 거시오.

```
ALTER TABLE dept
  ADD CONSTRAINT dept_loc_ck
    CHECK ( LOC IN ( 'NEW YORK', 'DALLAS', 'CHICAGO', 'BOSTON' ) );
```

## 17-5. Foreign key 제약 : 참조하는 컬럼에 거는 제약

Ex) emp 테이블

dept 테이블

Dept.Deptno에 p키 제약을 걸고, emp.dept 에는 foreign key 제약을 걸면,

Dept 테이블은 부모 테이블, emp 테이블은 자식 테이블

==> 부모테이블에 있는 제약이 자식에도 있다. 그렇다고 부모 데이터를 지울려 하는데 자식에

있는거면 함부로 지워지지 않는다.

%% dept 테이블 deptno에 primary key 제약을 건다.

```
ALTER TABLE DEPT
  ADD CONSTRAINT DEPT_deptno_pk PRIMARY KEY(deptno);
```

%%emp 테이블에 deptno에 foreign key 제약을 걸고

```
ALTER TABLE EMP
  ADD CONSTRAINT EMP_deptno_fk FOREIGN KEY(deptno)
    REFERENCES DEPT(deptno);
```

아래의 결과가 안됨-----

```
INSERT INTO emp(empno, ename, sal, deptno)
VALUES(2923, 'JACK', 4500, 80);
```

```
DELETE FROM DEPT WHERE deptno = 10;
```



문제316. dept 테이블에 deptno에 걸린 primary key를 삭제하시오.

```
ALTER TABLE DEPT
```

```
ADD CONSTRAINT dept_deptno_pk; <--- 먼저 삭제안됨.
```

```
%% alter table dept
```

```
Drop constraint dept_deptno_pk cascade; <-- 자식과 부모 동시에 삭제됨
```

- 제약을 중지시키고 다시 활성화 시키는 방법

문제317. 사원 테이블에 월급이 0에서 부터 9000 사이에 데이터만 입력되거나 수정 될 수 있도록

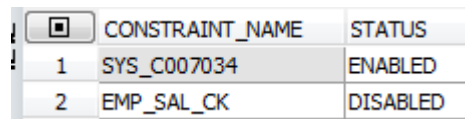
체크 제약을 거시오,

```
ALTER TABLE EMP
```

```
ADD CONSTRAINT emp_sal_ck  
CHECK(sal BETWEEN 0 AND 9000);
```

```
ALTER TABLE EMP
```

```
DISABLE CONSTRAINT emp_sal_ck; <== 중지
```



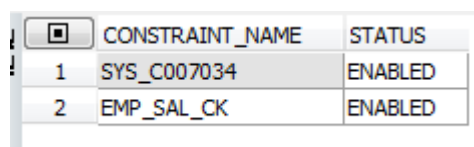
	CONSTRAINT_NAME	STATUS
1	SYS_C007034	ENABLED
2	EMP_SAL_CK	DISABLED

```
ALTER TABLE EMP
```

```
enable NOVALIDATE CONSTRAINT emp_sal_ck; <== 제약을 활성화 시키는데 기존에 들어있  
던
```

데이터 들에 제약검사를 하지말고  
제약을 활성화 시켜라.

```
SELECT CONSTRAINT_NAME,status  
FROM USER_CONSTRAINTS  
WHERE TABLE_NAME = 'EMP';
```



	CONSTRAINT_NAME	STATUS
1	SYS_C007034	ENABLED
2	EMP_SAL_CK	ENABLED

## 18. 테이블 (뷰)

2018년 4월 10일 화요일    오후 3:36

**View : 테이블 처럼 데이터를 저장하지는 않고 그냥 쿼리이다.**

### 데이터베이스 객체

객체	설명
테이블	기본 저장 단위이며 행으로 구성되어 있습니다.
뷰	하나 이상의 테이블에 있는 데이터의 부분 집합을 논리적으로 나타냅니다.
시퀀스	숫자 값을 생성합니다.
인덱스	데이터 검색 query의 성능을 향상시킵니다.
동의어	객체에 다른 이름을 부여합니다.

문제318. 부서번호, 이름, 월급, 순위를 출력하는데 순위가 부서번호별로 각각 월급이 높은 사원순으로 순위를 부여하시오.

```
Select deptno, ename, sal,  
dense_Rank() over(partition by deptno order by sal desc)  
From EMP;
```

	DEPTNO	ENAME	SAL	DENSE_RANK()OVER(PARTITIONBYDEPTNOORDERBYSALDESC)
1	10	KING	9500	1
2	10	CLARK	2450	2
3	10	MILLER	1300	3
4	20	SCOTT	3000	1
5	20	FORD	3000	1
6	20	JONES	2975	2
7	20	ADAMS	1100	3
8	20	SMITH	800	4
9	30	BLAKE	2850	1
10	30	ALLEN	1600	2
11	30	TURNER	1500	3
12	30	MARTIN	1250	4
13	30	WARD	1250	4
14	30	JAMES	950	5

문제319. 위의 결과에서 1등만 출력하시오.

```
SELECT *  
FROM (Select deptno, ename, sal,  
dense_Rank() over(partition by deptno order by sal desc) 순위  
From EMP)
```

WHERE 순위 = 1;

	DEPTNO	ENAME	SAL	순위
1	10	KING	9500	1
2	20	FORD	3000	1
3	20	SCOTT	3000	1
4	30	BLAKE	2850	1

%% 위에 결과를 맨날 봐야하는데 편하게 보기위해 뷰를 생성한다

```
CREATE OR REPLACE VIEW emp55
AS
Select deptno, ename, sal,
       dense_Rank() over(partition by deptno order by sal desc) 순위
From EMP;
```

생성하고

SELECT \* FROM emp55; 를 실행하면

	DEPTNO	ENAME	SAL	순위
1	10	KING	9500	1
2	10	CLARK	2450	2
3	10	MILLER	1300	3
4	20	SCOTT	3000	1
5	20	FORD	3000	1
6	20	JONES	2975	2
7	20	ADAMS	1100	3
8	20	SMITH	800	4
9	30	BLAKE	2850	1
10	30	ALLEN	1600	2
11	30	TURNER	1500	3
12	30	MARTIN	1250	4
13	30	WARD	1250	4
14	30	JAMES	950	5

SELECT \* FROM emp55 WHERE 순위 = 1;

	DEPTNO	ENAME	SAL	순위
1	10	KING	9500	1
2	20	FORD	3000	1
3	20	SCOTT	3000	1
4	30	BLAKE	2850	1

UPDATE EMP

SET sal = 300

WHERE ename = 'KING';

내용을 바꾸면 뷰안에있는 내용도 바뀐다.

바꾸고 SELECT \* FROM emp55 WHERE 순위 = 1; 를 실행하면

	DEPTNO	ENAME	SAL	순위
1	10	CLARK	2450	1
2	20	FORD	3000	1
3	20	SCOTT	3000	1
4	30	BLAKE	2850	1

문제320. 통신사, 이름, 나이, 순위를 출력하는데 통신사별로 각각 나이가 높은 순서대로

순위를 출력하는 쿼리를 뷰로 생성하시오. (view 이름 : emp320)

create OR REPLACE VIEW emp320

```
as
SELECT telecom, ename, age,
       dense_RANK() OVER(PARTITION BY telecom ORDER BY age desc) 순위
FROM EMP2;
```

문제321. 위의 view 의 결과에서 순위가 2등인 학생만 출력하시오.

```
SELECT *
FROM emp320
WHERE 순위 = 2;
```

	TELECOM	ENAME	AGE	순위
1	kt	신현수	27	2
2	kt	송윤호	27	2
3	kt	김지우	27	2
4	lg	은해찬	29	2
5	sk	윤진민	27	2
6	sk	김대경	27	2

문제322. 이름, 월급, 부서위치, 부서번호를 출력하는 view를 emp322로 생성하시오.

```
CREATE OR REPLACE VIEW emp322
as
SELECT e.ename, e.sal, d.loc, d.deptno
FROM EMP e, DEPT d
WHERE e.DEPTNO = d.DEPTNO;
```

	ENAME	SAL	LOC	DEPTNO
1	KING	300	NEW YORK	10
2	BLAKE	2850	CHICAGO	30
3	CLARK	2450	NEW YORK	10
4	JONES	2975	DALLAS	20
5	MARTIN	1250	CHICAGO	30
6	ALLEN	1600	CHICAGO	30
7	TURNER	1500	CHICAGO	30
8	JAMES	950	CHICAGO	30
9	WARD	1250	CHICAGO	30
10	FORD	3000	DALLAS	20
11	SMITH	800	DALLAS	20
12	SCOTT	3000	DALLAS	20
13	ADAMS	1100	DALLAS	20
14	MILLER	1300	NEW YORK	10

문제323. 위의 뷰의 결과에서 부서위치가 DALLAS만 출력하시오.

```
SELECT * FROM emp322 WHERE LOC = 'DALLAS';
```

	ENAME	SAL	LOC	DEPTNO
1	JONES	2975	DALLAS	20
2	FORD	3000	DALLAS	20
3	SMITH	800	DALLAS	20
4	SCOTT	3000	DALLAS	20
5	ADAMS	1100	DALLAS	20

문제324. 사원 테이블의 loc컬럼을 추가하고 아래의 뷰를 생성하시오.

```
ALTER TABLE EMP
ADD LOC VARCHAR2(20);
```

```
CREATE OR REPLACE VIEW emp324
AS
```

```
SELECT e.ename, e.loc AS emp_loc, d.loc AS dept_loc
FROM EMP e, DEPT d
WHERE e.DEPTNO = d.DEPTNO;
```

문제325. 사원 테이블에 sal2 라는 컬럼을 추가하고 ename, sal2, sal 을 조회하시오.

```
ALTER TABLE EMP
ADD sal2 number(10);
```

```
SELECT ename, sal2, sal
FROM EMP;
```

	ENAME	SAL2	SAL
1	KING	(null)	300
2	BLAKE	(null)	2850
3	CLARK	(null)	2450
4	JONES	(null)	2975
5	MARTIN	(null)	1250
6	ALLEN	(null)	1600
7	TURNER	(null)	1500
8	JAMES	(null)	950
9	WARD	(null)	1250
10	FORD	(null)	3000
11	SMITH	(null)	800
12	SCOTT	(null)	3000
13	ADAMS	(null)	1100
14	MILLER	(null)	1300

```
UPDATE EMP
SET sal2 = sal;
```

```
SELECT ename, sal2, sal
FROM EMP;
```

	ENAME	SAL2	SAL
1	KING	300	300
2	BLAKE	2850	2850
3	CLARK	2450	2450
4	JONES	2975	2975
5	MARTIN	1250	1250
6	ALLEN	1600	1600
7	TURNER	1500	1500
8	JAMES	950	950
9	WARD	1250	1250
10	FORD	3000	3000
11	SMITH	800	800
12	SCOTT	3000	3000
13	ADAMS	1100	1100
14	MILLER	1300	1300

문제326. emp324 view 를 업데이트하는데 dept\_loc의 데이터를 emp\_loc로 업데이트 하시오.

```
UPDATE emp324
SET emp_loc = dept_loc;
```

이렇게하면 안됨.

문제327. dept 테이블과 emp 테이블을 부모 자식관계로 만들고 다시 한번 업데이트 해보시오.

```
ALTER TABLE DEPT
ADD CONSTRAINT DEPT_deptno_pk PRIMARY KEY(deptno);
```

```
ALTER TABLE EMP
  ADD CONSTRAINT emp_deptno_fk FOREIGN KEY(deptno)
    REFERENCES DEPT(deptno);
```

```
UPDATE emp324
  SET emp_loc = dept_loc;
```

	ENAME	EMP_LOC	DEPT_LOC
1	KING	NEW YORK	NEW YORK
2	BLAKE	CHICAGO	CHICAGO
3	CLARK	NEW YORK	NEW YORK
4	JONES	DALLAS	DALLAS
5	MARTIN	CHICAGO	CHICAGO
6	ALLEN	CHICAGO	CHICAGO
7	TURNER	CHICAGO	CHICAGO
8	JAMES	CHICAGO	CHICAGO
9	WARD	CHICAGO	CHICAGO
10	FORD	DALLAS	DALLAS
11	SMITH	DALLAS	DALLAS
12	SCOTT	DALLAS	DALLAS
13	ADAMS	DALLAS	DALLAS
14	MILLER	NEW YORK	NEW YORK

%% 뷰를 안만들고 쿼리문을 다써서 할 수 있음(뷰를 만들권한이 없을때 쓰세요)

```
UPDATE (SELECT ename, e.loc AS emp_loc,
              d.loc AS dept_loc
        FROM EMP e, DEPT d
        WHERE e.deptno = d.deptno)
  SET emp_loc = dept_loc;
```

### %% View 의 종류

	단순 view	복합 view
Table 갯수	1개	2개 이상
그룹함수	미포함	포함
dml	가능	불가능 할 수 있음

### 단순 view

Ex) create view emp67

As

```
select empno, ename, sal
  From emp      <-1개
  Where job = 'SALSMAN';
```

```
SELECT * FROM emp67;
```

	EMPNO	ENAME	SAL
1	7654	MARTIN	1250
2	7499	ALLEN	1600
3	7844	TURNER	1500
4	7521	WARD	1250

%% ALLEN 의 sal을 3400으로 변경해보시오.

```
UPDATE emp67
  SET sal = 3400
```

WHERE ename = 'ALLEN';

	EMPNO	ENAME	SAL
1	7654	MARTIN	1250
2	7499	ALLEN	3400
3	7844	TURNER	1500
4	7521	WARD	1250

Emp67에서 변경되어도 emp에도 변경이 된다.

6	7499	ALLEN	SALESMAN	7698	1981-02-11 오전 12:00:00	3400
---	------	-------	----------	------	------------------------	------

### View가 필요한 이유?

1. 보안상의 이유로 특정 데이터를 조회 못하게 하고 싶은 경우.

Ex) emp 테이블에서 월급은 조회 안되게 하고 싶은 경우

Create view emp23

As

select empno, ename, job, mgr, hiredate, ...  
from emp;

2. 복잡한 쿼리를 간단하게 조회하고 싶은 경우

Ex) create view emp59

As

select e.ename, d.loc  
From emp e, dept d  
Where e.deptno = d.deptno;

Select \* from emp59;

문제328. king의 부서위치를 washington으로 바꾸어 보시오.

	ENAME	LOC
1	KING	NEW YORK
2	BLAKE	CHICAGO
3	CLARK	NEW YORK
4	JONES	DALLAS

UPDATE emp59

SET loc = 'washington'

WHERE ename = 'KING'; < -- 안됨

%% 조인 했을 때 emp 테이블 쪽은 변경이 되나 dept 테이블은 변경이 안된다.

문제329. emp 테이블에 dname 이라는 컬럼을 추가하고, dept 테이블의 dname 으로 값을 변경.

CREATE VIEW emp329

AS

SELECT e.ename, e.dname AS emp\_dname,  
d.dname AS dept\_dname

FROM EMP e, DEPT d

WHERE e.deptno = d.DEPTNO;

UPDATE emp329

SET emp\_dname = dept\_dname;

	ENAME	EMP_DNAME	DEPT_DNAME
1	KING	ACCOUNTING	ACCOUNTING
2	BLAKE	SALES	SALES
3	CLARK	ACCOUNTING	ACCOUNTING
4	JONES	RESEARCH	RESEARCH
5	MARTIN	SALES	SALES
6	ALLEN	SALES	SALES
7	TURNER	SALES	SALES
8	JAMES	SALES	SALES
9	WARD	SALES	SALES
10	FORD	RESEARCH	RESEARCH
11	SMITH	RESEARCH	RESEARCH
12	SCOTT	RESEARCH	RESEARCH
13	ADAMS	RESEARCH	RESEARCH
14	MILLER	ACCOUNTING	ACCOUNTING

이렇게 뷰를 생성해서 할 수 있고,

Emp 329 자리에 쓰기로도 할 수 있다.

Update (

```
SELECT e.ename, e.dname AS emp_dname,
       d.dname AS dept_dname
FROM EMP e, DEPT d
WHERE e.deptno = d.DEPTNO )
Set emp_dname = dept_dname;
```

**문제 330. 부서번호, 부서번호별 토탈월급을 가로로 출력하시오.**

```
SELECT SUM(DECODE(deptno, 10, sal, 0)) "10",
       SUM(DECODE(deptno, 20, sal, 0)) "20",
       SUM(DECODE(deptno, 30, sal, 0)) "30"
FROM EMP;
```

	10	20	30
1	4050	10875	11200

GROUP BY job;

	10	20	30
1	0	0	7400
2	1300	1900	950
3	300	0	0
4	2450	2975	2850
5	0	6000	0

%%위의 쿼리를 뷰로 만드시오.

Create view dept\_job\_sum

As (

```
SELECT SUM(DECODE(deptno, 10, sal, 0)) "10",
       SUM(DECODE(deptno, 20, sal, 0)) "20",
       SUM(DECODE(deptno, 30, sal, 0)) "30"
FROM EMP
Group by job;
```

Select \* from dept\_job\_sum;

<--- 테이블은 1개이지만 그룹함수를 포함하고 있어  
복합 view이다.

## view 의 옵션

1. With check option : where 절에서 기술한 조건에 위배되게끔 view 를 수정하지 못한다.
2. With read only : 아예 view 전체를 수정, 삭제, 입력이 불가능한 상태로 만들겠다.



문제331. 부서번호가 30번인 직원들의 이름과 월급과 부서번호를 출력하는 view를 생성하시오.

(view 이름은 emp331)

```
CREATE VIEW emp331
AS (
    SELECT ename, sal, deptno
    FROM EMP
    WHERE deptno = 30
);
```

SELECT \* FROM EMP331;

	ENAME	SAL	DEPTNO
1	BLAKE	2850	30
2	MARTIN	1250	30
3	ALLEN	3400	30
4	TURNER	1500	30
5	JAMES	950	30
6	WARD	1250	30

여기서 BLAKE의 부서번호를 10으로 바꾸어 보시오.

```
UPDATE emp331
SET deptno = 10
WHERE ename = 'BLAKE';
```

	ENAME	SAL	DEPTNO
1	MARTIN	1250	30
2	ALLEN	3400	30
3	TURNER	1500	30
4	JAMES	950	30
5	WARD	1250	30

%%

```
CREATE VIEW emp332
AS
SELECT ename, sal, deptno
FROM EMP
WHERE deptno = 30
WITH CHECK option;
```

이렇게 하면 where 절에있는 deptno를 수정할 수 없다.

문제332. 월급이 9000 이하인 직원들의 이름과 월급을 출력하는 view를 생성하는데, 월급을 9000보다 크게 수정할 수 없도록 view를 생성하시오.

```
CREATE VIEW emp333
as
SELECT ename, sal
FROM EMP
WHERE sal <= 9000
WITH CHECK OPTION;    => sal이 9천이상이면 변경이 안됨.
```

%% tip) create or replace view emp333 하면 없으면 생성, 있으면 변경

문제333. 우리반 테이블로 view를 생성하는데, 핸드폰 번호는 빼고 생성하고, 전체를 수정 못하게 생성.

(emp333)

```
CREATE OR REPLACE VIEW EMP333
AS
```

```
SELECT empno, ename, age, birth, major, email, address, telecom, birth_day
FROM EMP2
```

WITH READ only; <-- 전체가 수정 안됨

%% emp333이 테이블인지 뷰인지 확인하는 방법.

```
Select *
From user_views;
```

	VIEW_NAME	TEXT_LENGTH	TEXT
1	EMP55	118	Select deptno, ename, sal,
2	EMP320	112	SELECT telecom, ename, age,
3	EMP322	91	SELECT e.ename, e.sal,d.loc, d.deptno
4	EMP324	109	SELECT e.ename, e.loc AS emp_loc, d.loc AS dept_loc
5	EMP332	103	SELECT ename, sal, deptno
6	EMP67	64	select empno, ename, sal
7	EMP59	102	select e.ename, d.loc
8	EMP329	144	SELECT e.ename, e.dname AS emp_dname,
9	EMP331	96	(
10	EMP333	120	SELECT empno, ename, age, birth, major, email, address,

%% 내가 가지고 있는 테이블 리스트를 확인하시오.

```
Select *
from user_tables;
```

	TABLE_NAME	TABLESPACE_NAME
1	EMP05	SYSTEM
2	ORDER2	SYSTEM
3	CRIME_LOC	SYSTEM
4	CRIME_CAUSE	SYSTEM
5	EMP7	SYSTEM
6	SUCIDE	SYSTEM
7	CRIME_CAUSE2	SYSTEM

%% 내가 가지고 있는 view 들을 전부 삭제 하시오.

```
Select 'drop view ' || view_name || ';'
From user_views;
```

하고 복불(복구가 안되니 조심해서 사용해야함)

## 19. 시퀀스

2018년 4월 11일 수요일 오전 11:27

**Sequence : 숫자 값을 생성하는 db object**

**사용해야하는 이유?**

- 반드시 번호가 중복되지 않고 순서대로 입력되어야 하는 데이터의 경우 사용

Ex) 주식 매매, 수험표, 기계 시리얼 번호, 주문 번호

### 시퀀스

객체	설명
테이블	기본 저장 단위이며 행으로 구성되어 있습니다.
뷰	하나 이상의 테이블에 있는 데이터의 부분 집합을 논리적으로 나타냅니다.
시퀀스	숫자 값을 생성합니다.
인덱스	일부 query 성능을 향상시킵니다.
동의어	객체에 다른 이름을 부여합니다.

Ex2) create sequence seq1;

select seq1.nextval from dual; <- 계속 번호가 순서대로 올라가면서 생성됨 (은행 대기표)

Ex3) create sequence seq2;

```
create table order_product
(order_no number(10),
order_name varchar2(20),
order_price number(10));

Insert into order_product
Values(seq2.nextval, '키보드', 450000);

Insert into order_product
Values(seq2.nextval, '모니터', 800000);

SELECT * FROM order_product;
```

	ORDER_NO	ORDER_NAME	ORDER_PRICE
1	1	키보드	450000
2	2	모니터	800000

-----점심 문제-----

**문제334. 사원이름과 부서위치와 월급을 출력하는 뷰를 emp507로 생성하고 이뷰와 salgrade와**

**조인해서 급여등급이 3등급인 사원들의 이름, 월급, 부서위치, 급여등급을 출력 하시오.**

```
SELECT e507.ename, e507.sal, s.grade, e507.loc
FROM SALGRADE s, emp507 e507
WHERE e507.sal BETWEEN s.losal AND s.hisal
AND s.grade = 3;
```

```
SELECT * FROM SALGRADE;
```

	ENAME	SAL	GRADE	LOC
1	TURNER	1500	3	CHICAGO

## %% 시퀀스 생성구문

Create sequence seq3

START WITH 1           -- 시작값

INCREMENT BY 1       -- 증가치 (기본 1)

maxvalue 100           -- 최대값

MINVALUE -100       -- 최소값(최대값까지 갔다가 다시 순환할 때 -100부터 시작)

NOCYCLE               -- 순환여부

cache 20;             -- 메모리에 캐쉬( 미리 20을 뽑아서 메모리에 올려놓음)

**문제335. 시퀀스를 생성하시오.**

시퀀스 이름 : seq7

시작값       : 1

증가치       : 1

최대값       : 100,000

최소값       : -10

순환여부     : cycle

메모리 캐쉬 : 20

```
Create sequence seq7
```

```
start with 1
```

```
Maxvalue 100000
```

```
minvalue -10
```

```
cycle
```

```
cashe 20;
```

## %% 시퀀스 매개변수 값을 변경

```
Alter sequence seq1
```

```
maxvalue 1000000;
```

```
Select *
  from user_sequences;
```

[illegible]

## %% 시퀀스 삭제

```
Drop sequence seq1;
```

## 20. 인덱스(index)

2018년 4월 11일 수요일    오후 2:14

인덱스(index) : 검색 속도를 높이기 위한 db object

### 인덱스

객체	설명
테이블	기본 저장 단위이며 행으로 구성되어 있습니다.
뷰	하나 이상의 테이블에 있는 데이터의 부분 집합을 논리적으로 나타냅니다.
시퀀스	숫자 값을 생성합니다.
인덱스	일부 query 성능을 향상시킵니다.
동义词	객체에 다른 이름을 부여합니다.

문제336. 월급이 3000인 사원의 이름과 월급을 출력하시오.

```
Select ename, sal
from emp
where sal = 3000;
```

	ENAME	SAL
1	FORD	3000
2	SCOTT	3000

==월급에다가 인덱스 생성

Create index emp\_sal on emp(sal);    create index 인덱스이름 on 테이블(컬럼);

### %% 인덱스의 구조

컬럼값 + rowid

컬럼값 : 정렬(오름차순) 되어서 저장되어 있다.(인덱스에서 불러와서)

rowid : 행의 실제 물리적 주소(ex. 책의 페이지번호)

```
SELECT sal, ROWID
FROM EMP
WHERE sal > 0;
```

	SAL	ROWID
1	300	AAAE+ZAABAAALDBAAA
2	800	AAAE+ZAABAAALDBAAK
3	950	AAAE+ZAABAAALDBAAH
4	1100	AAAE+ZAABAAALDBAAM
5	1250	AAAE+ZAABAAALDBAAE
6	1250	AAAE+ZAABAAALDBAAI
7	1300	AAAE+ZAABAAALDBAAN
8	1500	AAAE+ZAABAAALDBAAG
9	2450	AAAE+ZAABAAALDBAAC
10	2850	AAAE+ZAABAAALDBAAB
11	2975	AAAE+ZAABAAALDBAAD
12	3000	AAAE+ZAABAAALDBAAJ
13	3000	AAAE+ZAABAAALDBAAL
14	3400	AAAE+ZAABAAALDBAAF

```

SELECT STATEMENT Optimizer Mode=ALL ROWS      14      350
  INDEX RANGE SCAN                             SCOTT.EMP_SAL  14      350

```

f7 눌러서 확인해보면 인덱스에서 불러옴

문제337. 이름에 인덱스를 생성하시오.

```
CREATE INDEX emp_ename ON emp(ename);
```

문제338. 이름과 rowid를 출력하는데, 인덱스에서 데이터를 가져오게 쿼리를 작성.

```

SELECT ename, ROWID
FROM EMP
WHERE ename > ' '; <-- 문자는 공백으로하면됨

```

문제339. 입사일의 인덱스를 생성하시오.

```
Create index emp_hiredate on emp(hiredate);
```

문제340. 입사일, rowid를 인덱스에서 데이터를 읽어와서 출력하시오.

```

SELECT hiredate, ROWID
FROM EMP
WHERE hiredate < TO_DATE('9999/12/30', 'RRRR/MM/DD'); <-- 아주 먼 미래의 날짜를 입력

```

&& 인덱스 컬럼을 where 절에 주어야 인덱스를 액세스 할 수 있는데,  
전체를 읽어와야 한다면 아래와 같이 검색조건을 준다.

1. 문자 > ' ' (공백)
2. 숫자 >= 0
3. 날짜 < to\_date('9999/12/31', 'RRRR/MM/DD')

정렬을 할 때 order by 절을 쓰게되면 성능이 안좋아진다.

문제341. 아래의 SQL을 튜닝하시오.

```

Select ename, sal
from emp
where sal >= 1200
order by sal asc;

```

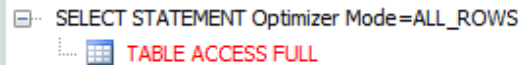
```
CREATE INDEX emp_sal ON EMP(sal);
```

```
SELECT ename, sal  
FROM EMP  
WHERE sal >= 1200;
```

**&& 인덱스 컬럼을 가공하게 되면 인덱스 액세스가 안된다.**

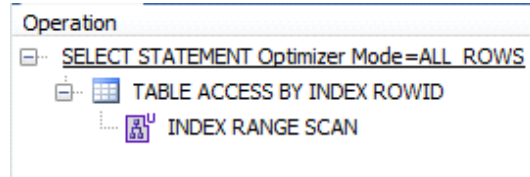
튜닝 전 : select ename, sal

```
from emp  
where sal*12 = 36000;
```



튜닝 후 :

```
SELECT ename, sal  
FROM EMP  
WHERE sal = 36000/12; <우변을 가공하면 인덱스 됨.
```



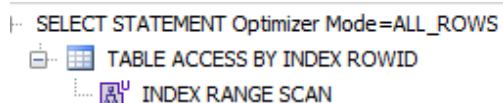
문제342. 아래의 SQL을 튜닝하시오.

```
Select ename, job  
From emp  
Where substr(job, 1, 5)= 'SALES';
```

-----  
Create index emp\_job on emp(job); <-- 직업에다가 인덱스 걸고

```
SELECT ename, job  
FROM EMP  
WHERE job LIKE 'SALES%';
```

	ENAME	JOB
1	MARTIN	SALESMAN
2	ALLEN	SALESMAN
3	TURNER	SALESMAN
4	WARD	SALESMAN



문제343. 아래의 SQL을 튜닝하시오.

```
Select ename, hiredate  
from emp  
where to_char(hiredate, 'RRRR/MM/DD') = '1981/12/11';
```

	ENAME	HIREDATE
1	JAMES	1981-12-11 오전 12:00:00
2	FORD	1981-12-11 오전 12:00:00



```
CREATE INDEX emp_hiredate ON EMP(hiredate);
```

```
SELECT ename, hiredate  
FROM EMP  
WHERE hiredate = to_date('1981/12/11', 'RRRR/MM/DD');
```

	ENAME	HIREDATE
1	JAMES	1981-12-11 오전 12:00:00
2	FORD	1981-12-11 오전 12:00:00

Operation  
SELECT STATEMENT Optimizer Mode=ALL\_ROWS  
TABLE ACCESS BY INDEX ROWID  
INDEX RANGE SCAN

문제344. 아래의 SQL을 튜닝하시오.

```
Insert into emp(empno, ename, sal)  
Values(3293, 'AAAA ', 34000);  
Select ename, sal  
From emp  
Where rtrim(ename) = 'AAAA';
```

```
-----  
SELECT ename, sal  
FROM EMP  
WHERE ename like 'AAAA%';
```

Operation  
SELECT STATEMENT Optimizer Mode=ALL\_ROWS  
TABLE ACCESS BY INDEX ROWID  
INDEX RANGE SCAN

```
-----  
INSERT INTO EMP (empno, ename, sal)  
VALUES(2922, ' JANE ', 45000);  
SELECT ename, sal  
FROM EMP  
WHERE TRIM(ename)='JANE';
```

```
-----  
SELECT ename, sal  
FROM EMP  
WHERE ename LIKE '%JANE%'; <-- 이렇게 하면 안됨
```

**함수기반을 만들어 줘야함 (트림까지 묶어서 만들기) = 함수 기반 인덱스**

```
Create index emp_ename_fum  
on emp(trim(ename) );
```

```
SELECT ename, sal  
FROM EMP  
WHERE trim(ename) = 'JANE';
```

	ENAME	SAL
1	JANE	45000

Operation  
SELECT STATEMENT Optimizer Mode=ALL\_ROWS  
TABLE ACCESS BY INDEX ROWID  
INDEX RANGE SCAN

**&& 인덱스 생성시 주의 사항**

1. 인덱스는 검색할 때 꼭 필요한 컬럼에만 인덱스를 걸어줘야 한다.
2. 인덱스가 많으면 테이블에 데이터 입력속도가 느려진다.

## 결합 컬럼 인덱스

### : 여러 개의 컬럼을 묶어서 인덱스를 생성

Ex) create index emp\_dept\_sal  
on emp(deptno, sal) ;

```
SELECT deptno, sal
FROM EMP
WHERE deptno > 0;
```

	DEPTNO	SAL
1	10	300
2	10	1300
3	10	2450
4	10	2850
5	20	800
6	20	1100
7	20	2975
8	20	3000
9	20	3000
10	30	950
11	30	1250
12	30	1250
13	30	1500
14	30	3400

부서별로 sal순으로 정렬된다.

문제345. 이름, 월급을 출력하는데, 월급이 높은 사원부터 출력하시오.

튜닝전: select ename, sal  
From emp  
Order by sal desc;

튜닝후 select /\*+ index\_desc(emp emp\_sal) \*/ ename, sal <<index\_desc=힌트  
From emp  
Where sal > 0;

힌트는 안줘도 되지만 더 좋게 하기 위해 쓴다.  
이미 좋은 쿼리에는 안써도 된다.

Operation
SELECT STATEMENT Optimizer Mode=ALL_ROWS
TABLE ACCESS BY INDEX ROWID
INDEX RANGE SCAN DESCENDING

문제346. 아래의 SQL을 튜닝하시오.

```
Select ename, hiredate
From emp
Order by hiredate desc;
```

CREATE INDEX emp\_hiredate ON EMP(hiredate);

```
SELECT /*+ index_desc(emp emp_hiredate) */ ename, hiredate
FROM EMP
WHERE hiredate < TO_DATE('9999/12/31', 'RRRR/MM/DD');
```

	ENAME	SAL
1	KING	5000
2	SCOTT	3000
3	FORD	3000
4	JONES	2975
5	BLAKE	2850
6	CLARK	2450
7	ALLEN	1600
8	TURNER	1500
9	MILLER	1300
10	WARD	1250
11	MARTIN	1250
12	ADAMS	1100
13	JAMES	950
14	SMITH	800

Operation  
 SELECT STATEMENT Optimizer Mode=ALL\_ROWS  
 TABLE ACCESS BY INDEX ROWID  
 INDEX RANGE SCAN DESCENDING

문제347. 아래의 SQL을 튜닝하시오.(max를 사용하지 않고 결과를 봐야함)

Select max(sal)  
 From emp;

-----  
 SELECT /\*+ index\_desc(emp emp\_sal) \*/ sal  
 FROM EMP  
 WHERE sal > 0 AND ROWNUM = 1;

	SAL
1	5000

Rownum : 결과를 출력시 넘버링하는 **쉐도우 컬럼**

SELECT ROWNUM, ename, sal  
 FROM EMP;

	ROWNUM	ENAME	SAL
1	1	KING	5000
2	2	BLAKE	2850
3	3	CLARK	2450
4	4	JONES	2975
5	5	MARTIN	1250
6	6	ALLEN	1600
7	7	TURNER	1500
8	8	JAMES	950
9	9	WARD	1250
10	10	FORD	3000
11	11	SMITH	800
12	12	SCOTT	3000
13	13	ADAMS	1100
14	14	MILLER	1300

문제348. 아래의 SQL을 튜닝하시오.

Select max(sal)  
 From emp  
 Where job = 'SALESMAN';

```

SELECT /*+ index_desc(emp emp_sal) */ sal
      FROM EMP
      WHERE job = 'SALESMAN'
      AND sal >= 0
      AND ROWNUM = 1;

```

	SAL
1	1600

문제349. 아래의 SQL을 그룹함수와 group by 쓰지 말고 튜닝하시오.(마지막 문제)

```

Select deptno, max(sal)
  From emp
  Where deptno = 20
  Group by deptno;

```

```

SELECT /*+ index_desc(emp emp_sal) */ deptno, sal
      FROM EMP
      WHERE deptno = 20
      AND sal >= 0
      AND ROWNUM = 1;

```

	DEPTNO	MAX(SAL)
1	20	3000

--Scott유저가 생성한 인덱스 리스트 확인

```

SELECT index_name
  FROM user_indexes;

```

	INDEX_NAME
1	EMP_SAL
2	EMP_HIREDATE
3	SYS_C007015

문제350. 직업이 SALESMAN인 직원들의 이름과 입사일을 출력하는데, 최근에 입사한 직원부터 출력하시오.

```

튜닝전: select ename, hiredate
  From emp
  Where job = 'SALESMAN'
  Oder by hiredate desc;

```

```

SELECT /*+ index_desc(emp emp_hiredate) */ ename, hiredate
      FROM EMP
      where job = 'SALESMAN'
      AND hiredate < TO_DATE('9999/12/31', 'RRRR/MM/DD');

```

	ENAME	HIREDATE
1	MARTIN	1981-09-10 오전 12:00:00
2	TURNER	1981-08-21 오전 12:00:00
3	WARD	1981-02-23 오전 12:00:00
4	ALLEN	1981-02-11 오전 12:00:00

문제351. 81년도에 입사한 직원들의 이름과 월급과 입사일을 출력하는데, 월급이 높은 사람부터 출력하시오.

```

SELECT /*+ index_desc(emp emp_sal) */ ename, sal, hiredate
      FROM EMP

```

WHERE hiredate BETWEEN TO\_date('1981-01-01', 'RRRR/MM/DD')  
AND TO\_date('1981-12-31', 'RRRR/MM/DD')  
AND sal >= 0;

	ENAME	SAL	HIREDATE
1	KING	5000	1981-11-17 오전 12:00:00
2	FORD	3000	1981-12-11 오전 12:00:00
3	JONES	2975	1981-04-01 오전 12:00:00
4	BLAKE	2850	1981-05-01 오전 12:00:00
5	CLARK	2450	1981-05-09 오전 12:00:00
6	ALLEN	1600	1981-02-11 오전 12:00:00
7	TURNER	1500	1981-08-21 오전 12:00:00
8	WARD	1250	1981-02-23 오전 12:00:00
9	MARTIN	1250	1981-09-10 오전 12:00:00
10	JAMES	950	1981-12-11 오전 12:00:00

Sal을 바로 가공하면 성능이 저하되지만,  
Hiredate는 가공해도 sal인덱스를 사용하기  
때문에 가공해도됨

문제352. 월급이 1000 에서 3000 사이인 직원들의 이름과 월급을 출력하는데 이름을 ABC순서대로 출력하시오.

```
SELECT /*+ index_asc(emp emp_ename) */ename, sal
FROM EMP
WHERE sal BETWEEN 1000 AND 3000
AND ename > ' ';
```

	ENAME	SAL
1	ADAMS	1100
2	ALLEN	1600
3	BLAKE	2850
4	CLARK	2450
5	FORD	3000
6	JONES	2975
7	MARTIN	1250
8	MILLER	1300
9	SCOTT	3000
10	TURNER	1500
11	WARD	1250

문제353. 부서번호가 10,20 번인 직원들의 이름과 직업을 출력하는데, 직업이 ABCD순서대로 출력하시오.

```
SELECT /*+ index_asc(emp emp_job) */ename, job
FROM EMP
WHERE DEPTno IN (10,20)
AND job > ' ';
```

	ENAME	JOB
1	FORD	ANALYST
2	SCOTT	ANALYST
3	SMITH	CLERK
4	ADAMS	CLERK
5	MILLER	CLERK
6	CLARK	MANAGER
7	JONES	MANAGER
8	KING	PRESIDENT

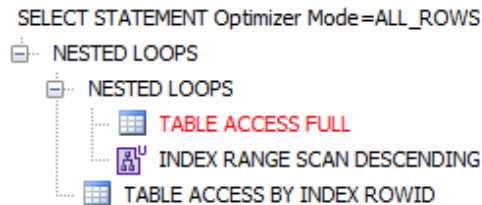
문제354. DALLAS에서 근무하는 직원들의 이름과 월급과 직업을 출력하는데 월급이 높은 직원부터 출력하시오.

```
SELECT /*+ index_desc(e emp_sal) */ e.ename, e.sal, d.loc
```

```
FROM EMP e, DEPT d
WHERE e.deptno = d.DEPTNO
AND d.LOC = 'DALLAS'
AND e.sal >= 0;
```

Emp가 e로 변경 되어서 힌트에도 e로 사용해야한다.

	ENAME	SAL	LOC
1	SCOTT	3000	DALLAS
2	FORD	3000	DALLAS
3	JONES	2975	DALLAS
4	ADAMS	1100	DALLAS
5	SMITH	800	DALLAS

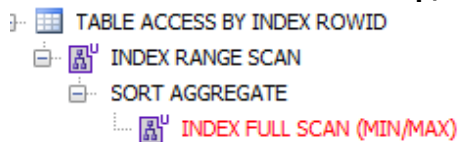


빨간색이 뜨긴뜨는데 dept에 뜨는거라

아직 안배워서 상관없음(나중에 조인 튜닝에서 배움)

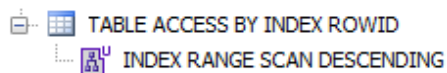
문제355. 아래의 SQL을 튜닝하시오.

```
Select ename, sal
From emp
Where sal = (select max(sal)
              From emp);
```



```
SELECT /*+ index_desc(emp emp_sal) */ ename, sal
FROM EMP
WHERE sal > 0
AND ROWNUM = 1;
```

	ENAME	SAL
1	KING	5000



문제356. 아래의 SQL을 튜닝하는데 dense\_rank 분석 함수를 이용해서 emp 테이블을 한번만 액세스 하게 작성하시오.(위의 결과에서 중복이 있을수 있다)

```
Select ename, sal
From emp
Where sal = (select max(sal)
              From emp);

%%
UPDATE EMP
SET sal = 5000
WHERE ename = 'SCOTT';    1등을 한명 더 넣어준다.
```

```
SELECT *
FROM (SELECT ename, sal,
             DENSE_RANK() over(ORDER BY sal desc) 순위
FROM EMP
```

WHERE sal > 0)

WHERE 순위 = 1;

	ENAME	SAL	순위
1	SCOTT	5000	1
2	KING	5000	1

WINDOW NOSORT STOPKEY  
TABLE ACCESS BY INDEX ROWID  
INDEX RANGE SCAN DESCENDING

## && 실행순서 확인하는방법

쿼리에 c+a f7 누르고, 실행통계 및 자동추적 ->

자동 추적		
	속성	값
1	recursive calls	0
2	db block gets	0
3	consistent gets	41
4	physical reads	0
5	redo size	0
6	bytes sent via SQL*Net	1102
7	bytes received via SQL*	1164
8	SQL*Net roundtrips to/fi	15
9	sorts (memory)	1
10	sorts (disk)	0

누르면 빨간동그라미를 누르고 다시 실행시킨다.

	속성	값
1	recursive calls	0
2	db block gets	0
3	consistent gets	41
4	physical reads	0

블록을 얼마나 읽었는지 총(여기는41) 개수가 많을수록 악성코드이다.  
위 결과는 튜닝전 SQL이다.

## 튜닝후 결과 SQL은

	속성	값
1	recursive calls	0
2	db block gets	0
3	consistent gets	2

문제357. 아래의 SQL을 튜닝하시오.

```
Select ename, hiredate
From emp
Where hiredate = (select max(hiredate)
                  From emp);
```

```
-----
SELECT *
FROM ( SELECT ename, hiredate,
              DENSE_RANK() over(ORDER BY hiredate desc) 순위
FROM EMP
WHERE hiredate < TO_DATE('9999/12/31','RRRR/MM/DD'))
```

WHERE 순위 = 1;

	ENAME	HIREDATE	순위
1	ADAMS	1983-01-15 오전 12:00:00	1

자동 추적		
	속성	값
1	recursive calls	0
2	db block gets	0
3	consistent gets	2
4	physical reads	0

## 인덱스를 통해서 테이블을 액세스 하는 방법

```

Select ename, hiredate
From emp
Where ename= 'SCOTT'
  
```

인덱스 이름 :Emp\_ename

인덱스는 책의 목차와 같아서  
데이터 검색을 할때 목차부터  
검색을 하면 속도가 빨라진다.

인덱스

	ENAME	ROWID
1	ADAMS	AAAE+7AABAAALDBAAM
2	ALLEN	AAAE+7AABAAALDBAAF
3	BLAKE	AAAE+7AABAAALDBAAB
4	CLARK	AAAE+7AABAAALDBAAC
5	FORD	AAAE+7AABAAALDBAAJ
6	JAMES	AAAE+7AABAAALDBAAH
7	JONES	AAAE+7AABAAALDBAAD
8	KING	AAAE+7AABAAALDBAAA
9	MARTIN	AAAE+7AABAAALDBAAE
10	MILLER	AAAE+7AABAAALDBAAN
11	SCOTT	AAAE+7AABAAALDBAAL
12	SMITH	AAAE+7AABAAALDBAAK
13	TURNER	AAAE+7AABAAALDBAAG
14	WARD	AAAE+7AABAAALDBAAI

테이블

	ENAME	HIREDATE	ROWID
1	SMITH	1980-12-09 오전 12:00:00	AAAE+7AABAAALDBAAK
2	ALLEN	1981-02-11 오전 12:00:00	AAAE+7AABAAALDBAAF
3	WARD	1981-02-23 오전 12:00:00	AAAE+7AABAAALDBAAI
4	JONES	1981-04-01 오전 12:00:00	AAAE+7AABAAALDBAAD
5	BLAKE	1981-05-01 오전 12:00:00	AAAE+7AABAAALDBAAB
6	CLARK	1981-05-09 오전 12:00:00	AAAE+7AABAAALDBAAC
7	TURNER	1981-08-21 오전 12:00:00	AAAE+7AABAAALDBAAG
8	MARTIN	1981-09-10 오전 12:00:00	AAAE+7AABAAALDBAAE
9	KING	1981-11-17 오전 12:00:00	AAAE+7AABAAALDBAAA
10	JAMES	1981-12-11 오전 12:00:00	AAAE+7AABAAALDBAAH
11	FORD	1981-12-11 오전 12:00:00	AAAE+7AABAAALDBAAJ
12	MILLER	1982-01-11 오전 12:00:00	AAAE+7AABAAALDBAAN
13	SCOTT	1982-12-22 오전 12:00:00	AAAE+7AABAAALDBAAL
14	ADAMS	1983-01-15 오전 12:00:00	AAAE+7AABAAALDBAAM

1. 인덱스에서 president를 찾아본다.
2. 찾았는데 원하는 값(hiredate)가 없으니 테이블로 가서 찾아온다.



## %% 실행계획보는 방법

Operation	Object Name	Rows
SELECT STATEMENT Optimizer Mode=ALL_ROWS		1
TABLE ACCESS BY INDEX ROWID	SCOTT.EMP	1
INDEX RANGE SCAN	SCOTT.EMP_ENAME	1

## %% 결합 컬럼 인덱스 생성

Create index emp\_ename\_hiredate  
On emp(ename, hiredate);

Select ename, hiredate  
From emp  
Where ename = 'SCOTT';

실행 계획은?

목차를 만들어줘서 table full이 아니라 인덱스로 된다.

문제358. 아래의 SQL은 어떻게 인덱스를 액세스해서 테이블을 액세스 하는지 그림으로 그리시오.

Select /\*+ index(emp emp\_job) \*/ ename, job, sal  
From emp  
Where job = 'PRESIDENT';

인덱스

테이블

	JOB	ROWID
1	ANALYST	AAAE+7AABAAALDBAAJ
2	ANALYST	AAAE+7AABAAALDBAAL
3	CLERK	AAAE+7AABAAALDBAAH
4	CLERK	AAAE+7AABAAALDBAAK
5	CLERK	AAAE+7AABAAALDBAAM
6	CLERK	AAAE+7AABAAALDBAAN
7	MANAGER	AAAE+7AABAAALDBAAB
8	MANAGER	AAAE+7AABAAALDBAAC
9	MANAGER	AAAE+7AABAAALDBAAD
10	PRESIDENT	AAAE+7AABAAALDBAAA
11	SALESMAN	AAAE+7AABAAALDBAAE
12	SALESMAN	AAAE+7AABAAALDBAAF
13	SALESMAN	AAAE+7AABAAALDBAAG
14	SALESMAN	AAAE+7AABAAALDBAAI

	ROWID	ENAME	JOB	SAL
1	AAAE+7AABAAALDBAAA	KING	PRESIDENT	5000
2	AAAE+7AABAAALDBAAB	BLAKE	MANAGER	2850
3	AAAE+7AABAAALDBAAC	CLARK	MANAGER	2450
4	AAAE+7AABAAALDBAAD	JONES	MANAGER	2975
5	AAAE+7AABAAALDBAAE	MARTIN	SALESMAN	1250
6	AAAE+7AABAAALDBAAF	ALLEN	SALESMAN	1600
7	AAAE+7AABAAALDBAAG	TURNER	SALESMAN	1500
8	AAAE+7AABAAALDBAAH	JAMES	CLERK	950
9	AAAE+7AABAAALDBAAI	WARD	SALESMAN	1250
10	AAAE+7AABAAALDBAAJ	FORD	ANALYST	3000
11	AAAE+7AABAAALDBAAK	SMITH	CLERK	800
12	AAAE+7AABAAALDBAAL	SCOTT	ANALYST	5000
13	AAAE+7AABAAALDBAAM	ADAMS	CLERK	1100
14	AAAE+7AABAAALDBAAN	MILLER	CLERK	1300

## 직업을 full table scan 해서 찾을때 방법

Select /\*+ full(emp) \*/ ename, job, sal  
From emp  
Where job = 'PRESIDENT';

	ENAME	JOB	SAL
--	-------	-----	-----

where job = 'PRESIDENT';

	ENAME	JOB	SAL
1	KING	PRESIDENT	5000
2	BLAKE	MANAGER	2850
3	CLARK	MANAGER	2450
4	JONES	MANAGER	2975
5	MARTIN	SALESMAN	1250
6	ALLEN	SALESMAN	1600
7	TURNER	SALESMAN	1500
8	JAMES	CLERK	950
9	WARD	SALESMAN	1250
10	FORD	ANALYST	3000
11	SMITH	CLERK	800
12	SCOTT	ANALYST	5000
13	ADAMS	CLERK	1100
14	MILLER	CLERK	1300



처음부터 끝까지 스캔한다 => 느려짐

문제359. 아래의 SQL이 테이블 액세스를 안하고 인덱스만 액세스 하게끔 인덱스를 생성하시오.

Select ename, sal, job, hiredate

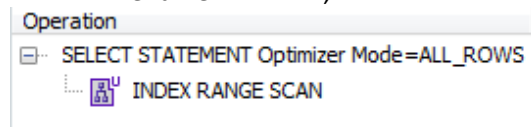
From emp

Where ename = 'ALLEN';

CREATE INDEX emp\_emp\_index2

ON EMP(ename, sal, job, hiredate);

-----  
SELECT /\*+ index(emp\_ename) \*/ ename, sal, job, hiredate  
FROM EMP  
WHERE ename = 'ALLEN';



문제360. 부서번호가 20번인 직원들의 이름과 월급과 입사일을 출력하는데 최근에 입사한 직원부터 출력하시오.

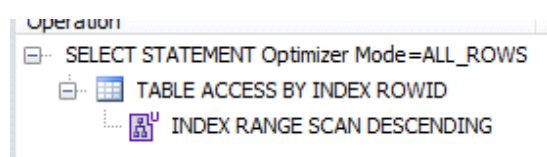
SELECT /\*+ index\_desc(emp emp\_hiredate) \*/ ename, sal, hiredate

FROM EMP

WHERE deptno = 20

AND hiredate < TO\_DATE('9999/12/31', 'RRRR/MM/DD');

	ENAME	SAL	HIREDATE
1	ADAMS	1100	1983-01-15 오전 12:00:00
2	SCOTT	5000	1982-12-22 오전 12:00:00
3	FORD	3000	1981-12-11 오전 12:00:00
4	JONES	2975	1981-04-01 오전 12:00:00
5	SMITH	800	1980-12-09 오전 12:00:00



## %% 인덱스 사용하는 SQL 작성 방법

1. Where 절에 검색에 사용되는 컬럼에 인덱스가 존재하는지 확인을 해야한다.
2. Where 절에 검색에 사용되는 컬럼이 가공이 되면 full table scan을 해서 느려짐.
3. Where 절에 검색에 사용되는 컬럼이 가공이 되지 않았는데도,  
인덱스를 액세스 하지 못하는 경우에 암시적 형 변환 여부를 확인 해봐야한다.

Like를 자주쓰면 숫자로 만들면 안된다.

그런데, like를 자주 쓰지 안 쓰지 예상하기 가 어려워

함수기반 인덱스를 사용해서 튜닝한다.

Ex)문제361. 아래의 SQL을 튜닝 하시오.

```
Select ename, sal, job, hiredate
FROM EMP
WHERE sal LIKE '30%';
```

Operation  
SELECT STATEMENT Optimizer Mode=ALL\_ROWS  
TABLE ACCESS FULL

Sal 은 숫자 이고 30%는 문자인데 원래는 문자가 바뀌어야하는데  
%가있어 sal이 문자로 바뀐다.

```
1 - filter(TO_CHAR("SAL") LIKE '30%')
```

함수 기반 인덱스를 생성한다.

```
Create index emp_sal_fun
On emp(to_char(sal));
```

생성 후에 다시 실행하면

Operation  
SELECT STATEMENT Optimizer Mode=ALL\_ROWS  
TABLE ACCESS BY INDEX ROWID  
INDEX RANGE SCAN

## %% 인덱스 삭제 방법

Drop index 인덱스이름;

## %% 인덱스 생성 방법 2가지

### 1. 수동으로 생성하는 방법

```
Create index emp_sal
On emp(sal);
```

### 2. 자동으로 생성하는 방법

Primary key 또는 unique 제약을 걸면 자동으로 그 컬럼에 인덱스가 생성됨.

제약을 제거하면 자동으로 생성된 인덱스도 제거된다.

## 21. 시너님(synonym)

2018년 4월 12일 목요일    오후 3:25

시너님(동의어) : 객체에 다른 이름을 부여할때 사용

### 동의어

객체	설명
테이블	기본 저장 단위이며 행으로 구성되어 있습니다.
뷰	하나 이상의 테이블에 있는 데이터의 부분 집합을 논리적으로 나타냅니다.
시퀀스	숫자 값을 생성합니다.
인덱스	일부 query 성능을 향상시킵니다.
동의어	객체에 다른 이름을 부여합니다.

### 생성 방법

```
Create synonym emp7000      emp7000 : 시너님 이름
For emp;
```

```
SELECT * FROM emp7000;
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESIDENT	(null)	1981-11-17 오전 12:00:00	5000	(null)	10
2	7698	BLAKE	MANAGER	7839	1981-05-01 오전 12:00:00	2850	(null)	30
3	7782	CLARK	MANAGER	7839	1981-05-09 오전 12:00:00	2450	(null)	10
4	7566	JONES	MANAGER	7839	1981-04-01 오전 12:00:00	2975	(null)	20
5	7654	MARTIN	SALESMAN	7698	1981-09-10 오전 12:00:00	1250	1400	30
6	7499	ALLEN	SALESMAN	7698	1981-02-11 오전 12:00:00	1600	300	30
7	7844	TURNER	SALESMAN	7698	1981-08-21 오전 12:00:00	1500	0	30
8	7900	JAMES	CLERK	7698	1981-12-11 오전 12:00:00	950	(null)	30
9	7521	WARD	SALESMAN	7698	1981-02-23 오전 12:00:00	1250	500	30
10	7902	FORD	ANALYST	7566	1981-12-11 오전 12:00:00	3000	(null)	20
11	7369	SMITH	CLERK	7902	1980-12-09 오전 12:00:00	800	(null)	20
12	7788	SCOTT	ANALYST	7566	1982-12-22 오전 12:00:00	5000	(null)	20
13	7876	ADAMS	CLERK	7788	1983-01-15 오전 12:00:00	1100	(null)	20
14	7934	MILLER	CLERK	7782	1982-01-11 오전 12:00:00	1300	(null)	10

만약

**Update emp7000**

**Set sal = 0**

**Where ename = 'SCOTT';** 을하면 emp 테이블에서 변경된다.

```
SELECT * FROM EMP;
```

<input type="checkbox"/>	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
1	7839	KING	PRESIDENT	(null)	1981-11-17 오전 12:00:00	5000
2	7698	BLAKE	MANAGER	7839	1981-05-01 오전 12:00:00	2850
3	7782	CLARK	MANAGER	7839	1981-05-09 오전 12:00:00	2450
4	7566	JONES	MANAGER	7839	1981-04-01 오전 12:00:00	2975
5	7654	MARTIN	SALESMAN	7698	1981-09-10 오전 12:00:00	1250
6	7499	ALLEN	SALESMAN	7698	1981-02-11 오전 12:00:00	1600
7	7844	TURNER	SALESMAN	7698	1981-08-21 오전 12:00:00	1500
8	7900	JAMES	CLERK	7698	1981-12-11 오전 12:00:00	950
9	7521	WARD	SALESMAN	7698	1981-02-23 오전 12:00:00	1250
10	7902	FORD	ANALYST	7566	1981-12-11 오전 12:00:00	3000
11	7369	SMITH	CLERK	7902	1980-12-09 오전 12:00:00	800
12	7788	SCOTT	ANALYST	7566	1982-12-22 오전 12:00:00	0

## 22. 스칼라 서브쿼리

2018년 4월 12일 목요일    오후 3:37

### 스칼라 서브쿼리

- Select 문에서 서브쿼리를 사용할 수 있는 절

서브쿼리

Select	가능	<- select에서 쓰이는 서브쿼리가 스칼라 서브쿼리
From	가능	<- 인라인뷰
Where	가능	<- 그냥 서브쿼리
Group by	불가능	
Having	가능	<- 그냥 서브쿼리
Order by	가능	<- order by도 스칼라 서브쿼리

Ex)

-최대월급을 출력하시오.

Select max(sal) from emp;

-최소월급을 출력하시오.

Select min(sal) from emp;

-이름과 월급을 출력하시오.

Select ename, sal from emp;

문제362. 이름, 월급, 최대월급, 최소월급을 출력하시오.

SELECT ename, sal, (Select max(sal) from emp) 최대,  
(Select min(sal) from emp) 최소

<--튜닝전 SQL

emp 테이블을 3번 select한다.

FROM EMP;

□	ENAME	SAL	최대	최소
1	KING	5000	5000	800
2	BLAKE	2850	5000	800
3	CLARK	2450	5000	800
4	JONES	2975	5000	800
5	MARTIN	1250	5000	800
6	ALLEN	1600	5000	800
7	TURNER	1500	5000	800
8	JAMES	950	5000	800
9	WARD	1250	5000	800
10	FORD	3000	5000	800
11	SMITH	800	5000	800
12	SCOTT	3000	5000	800
13	ADAMS	1100	5000	800
14	MILLER	1300	5000	800

튜닝 후

Select ename, sal,

Max(sal) over () 최대

() 안에 내용으로 확장해서 출력

튜닝 후

```
Select ename, sal,
      Max(sal) over () 최대,      () 안에 내용으로 확장해서 출력
      Min(sal) over () 최소
from emp;
```

문제363. 이름, 월급, 최대월급, 최소월급, 평균월급, 토탈월급, 전체 인원수를 출력하시오.

```
SELECT ename, sal, max(sal) OVER () 최대,
       MIN(sal) OVER () 최소,
       AVG(sal) OVER () 평균,
       sum(sal) OVER () 토탈,
       COUNT(*) OVER () 전체 인원수
from EMP;
```

문제364. 부서번호, 부서번호별 최대월급을 출력하시오.

```
SELECT deptno, MAX(sal)
FROM EMP
GROUP BY deptno;
```

문제365. 이름, 월급, 부서번호, 부서번호별 최대 월급을 출력하시오.

```
SELECT ename, sal, deptno, (MAX(sal) OVER (PARTITION BY deptno) ) 최대
FROM EMP;      <-- 튜닝후
```

일련 번호	ENAME	SAL	DEPTNO	최대
1	KING	5000	10	5000
2	CLARK	2450	10	5000
3	MILLER	1300	10	5000
4	ADAMS	1100	20	3000
5	SCOTT	3000	20	3000
6	SMITH	800	20	3000
7	FORD	3000	20	3000
8	JONES	2975	20	3000
9	WARD	1250	30	2850
10	JAMES	950	30	2850
11	ALLEN	1600	30	2850
12	MARTIN	1250	30	2850
13	BLAKE	2850	30	2850
14	TURNER	1500	30	2850

튜닝 전을 작성해보자.

```
SELECT ename, sal, deptno, (SELECT max(sal) FROM EMP WHERE deptno = e.deptno) 최대
FROM EMP e;
```

일련 번호	ENAME	SAL	DEPTNO	최대
1	KING	5000	10	(null)
2	BLAKE	2850	30	(null)
3	CLARK	2450	10	(null)
4	JONES	2975	20	(null)
5	MARTIN	1250	30	(null)
6	ALLEN	1600	30	(null)
7	TURNER	1500	30	(null)

e.deptno에다가 하나씩 하나씩 넣어줌

	ENAME	SAL	DEPTNO	최대
1	KING	5000	10	(null)
2	BLAKE	2850	30	(null)
3	CLARK	2450	10	(null)
4	JONES	2975	20	(null)
5	MARTIN	1250	30	(null)
6	ALLEN	1600	30	(null)
7	TURNER	1500	30	(null)
8	JAMES	950	30	(null)
9	WARD	1250	30	(null)
10	FORD	3000	20	(null)
11	SMITH	800	20	(null)
12	SCOTT	3000	20	(null)
13	ADAMS	1100	20	(null)
14	MILLER	1300	10	(null)

e.deptno에다가 하나씩 하나씩 넣어줌

문제366. 이름, 월급, 부서번호, 자기가 속한 부서번호의 평균월급이 출력되게 하는데, 자기의 월급이 자기가 속한 부서번호의 평균월급보다 더 큰 사원들만 출력하시오.(마지막 문제)

```
SELECT *
FROM (
    SELECT ename, sal, deptno, round(avg(sal) OVER (PARTITION BY deptno)) 평균
    FROM EMP e
)
WHERE sal > 평균;
```

	ENAME	SAL	DEPTNO	평균
1	KING	5000	10	2917
2	SCOTT	3000	20	2175
3	FORD	3000	20	2175
4	JONES	2975	20	2175
5	ALLEN	1600	30	1567
6	BLAKE	2850	30	1567

문제367. 위의 SQL을 분석함수를 사용하지 말고(튜닝전) 작성하시오.

```
CREATE VIEW dept_avg
AS
SELECT deptno, AVG(sal) avgsal
FROM EMP
GROUP BY deptno;
```

<--- 먼저 뷰를 만들어준다.

```
SELECT e.ename, e.sal, e.deptno, v.avgsal
FROM EMP e, DEPT_avg v
WHERE e.DEPTNO = v.DEPTNO
AND e.sal > v.avgsal;
```

<--- 뷰를 만들고하려면 dba한테 요청해야 하는데 뷰를 만들지 말고 직접 쓰면된다.

```
SELECT e.ename, e.sal, e.deptno, v.avgsal
FROM EMP e, (select deptno, avg(sal) avgsal
from emp
```



```

group by deptno) v
WHERE e.DEPTNO = v.DEPTNO
AND e.sal > v.avgsal;

```

문제368. 이름, 월급, 직업, 직업별 평균 월급을 출력하는데 자기의 월급이 자기 직업의 평균월급보다 더 큰 직원들만 출력하시오.

```

SELECT *
FROM(SELECT ename, sal, job, ROUND(AVG(sal) OVER(PARTITION BY job) ) 직업별평균
FROM EMP)
WHERE 직업별평균 < sal;

```

	ENAME	SAL	JOB	직업별평균
1	MILLER	1300	CLERK	1038
2	ADAMS	1100	CLERK	1038
3	JONES	2975	MANAGER	2758
4	BLAKE	2850	MANAGER	2758
5	ALLEN	1600	SALESMAN	1400
6	TURNER	1500	SALESMAN	1400

자동 추적		
	속성	값
1	recursive calls	0
2	db block gets	0
3	consistent gets	3

튜닝전 :

```

SELECT e.ename, e.sal, e.job, v.avgsal
FROM EMP e, ( SELECT job, ROUND(AVG(sal)) avgsal
FROM EMP
GROUP BY job) v
WHERE e.job = v.job
AND e.sal > v.avgsal;

```

자동 추적		
	속성	값
1	recursive calls	19
2	db block gets	0
3	consistent gets	30

문제369. 직업, 직업별 인원수를 출력하시오.

```

SELECT job, COUNT(*)
FROM EMP
GROUP BY job;

```

	JOB	COUNT(*)
1	SALESMAN	4
2	CLERK	4
3	PRESIDENT	1
4	MANAGER	3
5	ANALYST	2

문제370. 이름, 직업, 직업별 인원수를 출력하는데, 자기 직업의 인원수가 3명 이상인 것만 출력하시오.

튜닝전

```

SELECT e.ename, e.job, v.cnt

```

```

FROM EMP e, (SELECT job, COUNT(*) cnt
              FROM EMP
              GROUP BY job) v
WHERE e.job = v.job
AND v.cnt >= 3;

```

	ENAME	JOB	CNT
1	WARD	SALESMAN	4
2	TURNER	SALESMAN	4
3	ALLEN	SALESMAN	4
4	MARTIN	SALESMAN	4
5	MILLER	CLERK	4
6	ADAMS	CLERK	4
7	SMITH	CLERK	4
8	JAMES	CLERK	4
9	JONES	MANAGER	3
10	CLARK	MANAGER	3
11	BLAKE	MANAGER	3

	속성	값
1	recursive calls	19
2	db block gets	0
3	consistent gets	30

### 튜닝후

```

SELECT *
FROM (SELECT ename, job, COUNT(*) OVER(PARTITION BY job) cnt
      FROM EMP)
WHERE cnt >= 3;

```

	ENAME	JOB	CNT
1	SMITH	CLERK	4
2	MILLER	CLERK	4
3	JAMES	CLERK	4
4	ADAMS	CLERK	4
5	JONES	MANAGER	3
6	CLARK	MANAGER	3
7	BLAKE	MANAGER	3
8	WARD	SALESMAN	4
9	ALLEN	SALESMAN	4
10	MARTIN	SALESMAN	4
11	TURNER	SALESMAN	4

	속성	값
1	recursive calls	4
2	db block gets	0
3	consistent gets	7

문제371. 이름, 부서위치, 부서위치별 인원수를 출력하는데, 자기 부서위치의 인원수가 4명 이상인 직원들만 출력하시오.

### 튜닝 후

```

SELECT *
FROM (SELECT e.ename, d.loc, COUNT(*) OVER(PARTITION BY d.loc) cnt
      FROM EMP e, DEPT d
      WHERE e.deptno = d.DEPTNO
      )

```

where cnt >= 4;

	ENAME	LOC	CNT
1	TURNER	CHICAGO	6
2	WARD	CHICAGO	6
3	JAMES	CHICAGO	6
4	ALLEN	CHICAGO	6
5	MARTIN	CHICAGO	6
6	BLAKE	CHICAGO	6
7	SCOTT	DALLAS	5
8	SMITH	DALLAS	5
9	FORD	DALLAS	5
10	JONES	DALLAS	5
11	ADAMS	DALLAS	5

	속성	값
1	recursive calls	7
2	db block gets	0
3	consistent gets	14

## Having 절의 서브쿼리

Ex) 직업, 직업별 최대 월급을 출력하시오.

```
SELECT job, MAX(sal)
FROM EMP
GROUP BY job;
```

위의 결과에서 SALESMAN의 최대월급보다 더 많은 것만 출력하시오.

```
SELECT job, MAX(sal)
FROM EMP
GROUP BY job
HAVING MAX(sal) > (SELECT MAX(sal) FROM EMP WHERE job = 'SALESMAN');
```

문제372. 부서위치, 부서위치별 인원수를 출력하는데, NEW YORK의 인원수 보다 더 많은 것만 출력하시오.

```
SELECT d.loc, COUNT(*)
FROM EMP e, DEPT d
WHERE e.DEPTNO = d.DEPTNO
GROUP BY d.loc
HAVING COUNT(*) > (SELECT COUNT(*) FROM EMP e, DEPT d
WHERE e.DEPTNO = d.DEPTNO
AND d.loc = 'NEW YORK'
);
```

	LOC	COUNT(*)
1	CHICAGO	6
2	DALLAS	5

	속성	값
1	recursive calls	14
2	db block gets	0
3	consistent gets	28

문제373. 이름, 월급, 부서번호, 자기가 속한 부서번호의 토탈월급,

```
SELECT ename, sal, deptno, (SELECT SUM(sal)
                             FROM EMP
                             WHERE deptno = outer.deptno) 토탈
FROM EMP outer;
```

	속성	값
1	recursive calls	8
2	db block gets	0
3	consistent gets	58

```
SELECT ename, sal deptno, (SELECT SUM(sal) FROM EMP WHERE deptno = outer.deptno) 토탈,  
      (SELECT MAX(sal) FROM EMP WHERE deptno = outer.deptno) 최대,  
      (SELECT min(sal) FROM EMP WHERE deptno = outer.deptno) 최소,  
      (SELECT avg(sal) FROM EMP WHERE deptno = outer.deptno) 평균  
FROM EMP outer;
```

SQL 활용 페이지 156

자동 추적		
	속성	값
1	recursive calls	17
2	db block gets	0
3	consistent gets	97

위의 결과는 분석함수로 바꿀수가 없다. WHY?

### (스칼라 서브쿼리의 특징)

문제375(점심시간문제) 문제375번의 결과를 튜닝하시오.

```
SELECT ename, sal deptno, (SELECT SUM(sal), MAX(sal), MIN(sal), AVG(sal)
                           FROM EMP
                           WHERE deptno = outer.deptno)
FROM EMP outer;
```

이렇게 하면 값의 수가 너무 많습니다 라는 오류가뜨는데

스칼라 서브쿼리는 결과값으로 하나의 값만 리턴하는 특징이 있다.

=> , 대신 || 를 붙여준다

```
SELECT ename, sal deptno, (SELECT RPAD(SUM(sal),10,' ') ||
                               RPAD(MAX(sal),10,' ') ||
                               RPAD(MIN(sal),10,' ') ||
                               RPAD(AVG(sal),10,' ')
                           FROM EMP
                           WHERE deptno = outer.deptno)
FROM EMP outer;
```

-----답

```
CREATE VIEW ttt
AS
SELECT ename, sal, deptno, (SELECT RPAD(SUM(sal),10,' ') ||
                               RPAD(MAX(sal),10,' ') ||
                               RPAD(MIN(sal),10,' ') ||
                               RPAD(AVG(sal),10,' ')
                           FROM EMP
                           WHERE deptno = outer.deptno) t123
FROM EMP outer;
```

뷰를 만들어서

```
SELECT ename, sal, deptno, substr(t123, 1, instr(t123, ' ')) 토탈,
                               substr(t123, 11, instr(t123, ' ')) 최대,
                               substr(t123, 21, instr(t123, ' ')) 최소,
                               substr(t123, 31, instr(t123, ' ')) 평균
FROM ttt; <- 뷰를 넣어준다.
```

뷰를 못만들면 뷰이름에 위의 as뒤에 SQL문을 넣어주면된다.

## 23. 상호관련 서브쿼리

2018년 4월 13일 금요일    오후 1:47

**상호관련 서브쿼리 : 서브쿼리와 메인쿼리가 상호 연관이 있게끔 작성한 쿼리**

Ex) select ename, sal

From emp main

Where sal > ( select avg(sal)

from emp sub

where deptno = main.deptno);

자신의 월급이 자신이 속한 부서번호의 평균 월급보다 더 큰 직원들을 출력

문제375. 이름, 직업, 출력하는데, 자기 직업의 인원수가 3명 이상인 직업만 출력되게 하시오.

(상호관련 서브쿼리로)

SELECT ename, job

FROM EMP main

WHERE 3 <= (SELECT COUNT(\*)

FROM EMP SUB

WHERE job = main.JOB);

	ENAME	JOB
1	BLAKE	MANAGER
2	CLARK	MANAGER
3	JONES	MANAGER
4	MARTIN	SALESMAN
5	ALLEN	SALESMAN
6	TURNER	SALESMAN
7	JAMES	CLERK
8	WARD	SALESMAN
9	SMITH	CLERK
10	ADAMS	CLERK
11	MILLER	CLERK

문제376. 부서테이블에서 부서위치를 출력하는데 부서의 인원이 3명 이상인 부서위치만 출력되게 하시오.

SELECT LOC

FROM dept d

WHERE 3 <= (SELECT COUNT(\*)

FROM emp

WHERE deptno= d.deptno);

조인 할 필요 없이 emp에서 가져오면된다.

문제377. 자기 전공의 인원수가 2명 이상인 전공을 갖는 학생이름과 전공을 출력하시오.

SELECT ename, major

FROM EMP2 main

WHERE 2 <= (SELECT COUNT(\*)

FROM EMP2

WHERE major = main.major);

	ENAME	MAJOR
1	김원섭	컴퓨터공학
2	김광록	경영정보학
3	윤진민	경제학
4	백광훈	컴퓨터공학
5	장은희	경영정보학
6	차호성	경제학
7	방승준	컴퓨터공학
8	미한새	경영학
9	이근호	통계학
10	신영근	컴퓨터공학
11	신현수	경제학
12	김대경	통계학

문제378. 통신사별 인원수가 2명 이하인 통신사를 갖는 학생의 이름과 통신사를 출력하시오.

```
SELECT ename, telecom
FROM EMP2 main
WHERE 2 >= (SELECT COUNT(*)
            FROM EMP2
            WHERE telecom = main.TELECOM);
```

	ENAME	TELECOM
1	이근호	cjh

**%% exists와 not exists : 메인쿼리를 먼저 수행하는데 메인쿼리를 수행할 때,  
서브쿼리에 데이터가 존재하는지 확인하는 쿼리문.**

Ex)

```
Select loc
From dept d
Where exists ( select 'x'
              From emp e
              Where e.deptno = d.deptno);
```

	LOC	DEPTNO
1	NEW YORK	10
2	DALLAS	20
3	CHICAGO	30
4	BOSTON	40

	DEPTNO
1	10
2	30
3	10
4	20
5	30
6	30

존재하면 멈춘다.

위부터 찾다가 존재하면 멈춤

문제379. 부서테이블에는 존재하는 부서번호인데, 사원테이블에는 존재하지 않는 부서번호에 대한

부서위치를 부서테이블에서 출력하시오.

```
SELECT d.deptno
FROM DEPT d
WHERE NOT EXISTS (SELECT 'x'
                  FROM emp e
                  WHERE e.deptno = d.deptno);
```

<input type="checkbox"/>	DEPTNO
1	40



## 24. With 절

2018년 4월 13일 금요일      오후 3:00

**With 절** : 쿼리의 결과를 temp 테이블 스페이스에 저장해서 쿼리의 성능을 높이는 SQL문 (from 절에 서브쿼리를 temp 테이블에 저장하는 것)

EX) 분석함수를 이용해서 이름, 월급, 부서번호, 자기가 속한 부서번호의 평균 월급을 출력 하시오.

```
SELECT ename, sal, deptno,
       AVG(sal) OVER(PARTITION BY deptno) 평균
FROM EMP;
```

	ENAME	SAL	DEPTNO	평균
1	KING	5000	10	2916.6666666666666666666666666667
2	CLARK	2450	10	2916.6666666666666666666666666667
3	MILLER	1300	10	2916.6666666666666666666666666667
4	ADAMS	1100	20	2175
5	SCOTT	3000	20	2175
6	SMITH	800	20	2175
7	FORD	3000	20	2175
8	JONES	2975	20	2175

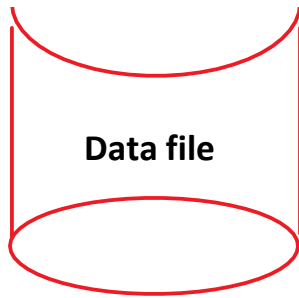
Ex2) 분석함수를 이용해서 이름, 월급, 부서번호, 자기가 속한 부서번호의 평균 월급을 출력하는데, 자기의 월급이 자기가 속한 부서번호의 평균 월급보다 더 큰 직원들만 출력 하시오.

```
SELECT *
      FROM (SELECT ename, sal, deptno,
                   AVG(sal) OVER(PARTITION BY deptno) 평균
              FROM EMP)                <-- from절에 서브쿼리를 쓰면
     WHERE 평균 < sal;
```

<input type="checkbox"/>	ENAME	SAL	DEPTNO	평균
1	KING	5000	10	2916.6666666666666666666666666667
2	SCOTT	3000	20	2175
3	FORD	3000	20	2175
4	JONES	2975	20	2175
5	ALLEN	1600	30	1566.6666666666666666666666666667
6	BLAKE	2850	30	1566.6666666666666666666666666667

## 메모리

From 절 서브쿼리를 메모리에 올려서  
메모리에 있는 결과를 가져온다.



With 절을 쓰면

From 절 서브쿼리는 data file에 올린다.

**With emp300 as** (SELECT /\*+ materialize \*/ ename, sal, deptno,  
AVG(sal) OVER(PARTITION BY deptno) 평균  
FROM EMP )

Select ename, sal, deptno, 평균  
From emp300  
Where sal > 평균;

Materialize 힌트를 주면 temp table 이 보인다.

```

SELECT STATEMENT Optimizer Mode=ALL_F
  TEMP TABLE TRANSFORMATION
    LOAD AS SELECT
      WINDOW SORT
        TABLE ACCESS FULL
      VIEW
        TABLE ACCESS FULL
  
```

Inline을 주면 일반 서브 쿼리로 돌아온다

-with 절 안 썼을 때

SELECT job, SUM(sal)  
FROM EMP  
GROUP BY job  
HAVING SUM(sal) > (SELECT AVG(SUM(sal))  
FROM EMP  
GROUP BY job);

JOB	SUM(SAL)
1 MANAGER	8275
2 ANALYST	6000

20분이걸린거를 두번해서 느리다

-with 절 썼을 때

WITH job\_sumsal AS ( SELECT job, SUM(sal) sumsal  
FROM EMP  
GROUP BY job )

SELECT job, sumsal  
FROM job\_sumsal  
WHERE sumsal > ( SELECT AVG(sumsal)  
FROM job\_sumsal );

JOB	SUMSAL
1 MANAGER	8275
2 ANALYST	6000

<--20분걸린 결과 값에서 가져오는거라 금방 가져온다

## with절 튜닝시 힌트 2가지(★★★★★)

1. `/*+ materialize */` = temp 테이블을 사용하겠다.
2. `/*+ inline */` = temp 테이블 사용안하고 서브쿼리로 수행하겠다.

## 25. 대형 데이터 집합 조작

2018년 3월 29일 목요일    오후 2:28

### 25-1.subquery를 사용한 insert문

: insert문에서 subquery를 사용할 수 있는 절

Ex) insert into emp(empno, ename, sal)

Values(1234, 'soctt', 3000);

<-- 둘다 가능하지만 (values 절은 values를 쓸 수 없음)

문제380. 아래와 같이 서브쿼리를 작성하시오.

Insert into (select empno, ename, sal, deptno

From emp

<- 서브쿼리

Where deptno = 20)

Values(1234, 'JACK', 3500, 20);

14	7934	MILLER	CLERK	7782	1982-01-11 오전 12:00:00	1300	(null)	10
15	1234	JACK	(null)	(null)	(null)	3500	(null)	20

설명 : 위의 insert문의 서브 쿼리에 with check option 조건을 추가하시오.

Insert into (select empno, ename, sal, deptno

From emp

Where deptno = 20

With check option)

Values(1234, 'JACK', 3500, 20);

<- where 절에서 기술한 조건에 위배되게끔 데이터를  
입력하려 하면 입력 할 수 없게 하겠다.

아래와 같이 30번을 넣으면 에러가 난다.

Insert into (select empno, ename, sal, deptno

From emp

Where deptno = 20

WITH CHECK option)

Values(1234, 'JACK', 3500, 30);

### 치환변수 (&) 사용 방법

: 동일한 컬럼을 데이터만 바꿔서 검색하거나 입력 해야 할 때  
치환변수를 사용하면 편리하다.

Ex) select empno, ename, sal, deptno

From emp

Where deptno = &부서번호;

```
SQL> select empno, ename, sal, deptno
2          From emp
3  Where deptno = &부서번호;
Enter value for 부서번호: 20
```

```

Enter value for 부서번호: 20
old 3: Where deptno = &부서번호
new 3: Where deptno = 20
Where deptno = 20
*
ERROR at line 3:
ORA-00933: SQL command not properly ended

```

문제381. 직업을 물어보게하고 직업을 입력하면 해당 직업인 직원들의 이름과 직업이 출력될 수 있도록 치환변수를 사용해서 SQL을 작성하시오.

```

SELECT ename, job
FROM EMP
WHERE job = &직업;

```

```

SQL> SELECT ename, job
2 FROM EMP
3 WHERE job = &직업;
Enter value for 직업: 'SALESMAN'
old 3: WHERE job = &직업
new 3: WHERE job = 'SALESMAN'

```

ENAME	JOB
MARTIN	SALESMAN
ALLEN	SALESMAN
TURNER	SALESMAN
WARD	SALESMAN

문제382. 치환변수(&)를 이용해서 직원번호, 이름, 월급, 부서번호를 각각 물어보게하고, emp 테이블에 입력될 수 있도록 하시오.

```

INSERT INTO emp(empno, ename, sal, deptno)
VALUES(&empno, '&ename', &sal, &deptno) <- 문자에는 ' ' 를붙여 줘야한다.

```

```

Enter value for empno: 3838
Enter value for ename: ALLEN
Enter value for sal: 3400
Enter value for deptno: 20
old 2: VALUES(&empno, '&ename', &sal, &deptno)
new 2: VALUES(3838, 'ALLEN', 3400, 20)

```

하고 / enter 를 치면 또 입력 할 수 있다.

문제383. 위의 insert 문에 insert 절에 서브쿼리를 사용해서 직원번호, 이름, 월급, 부서번호가 입력되게 하시오.

```

INSERT INTO (select empno, ename, sal, deptno
From emp
Where deptno = 20)
VALUES(&empno, '&ename', &sal, &deptno);

```

문제384. 위의 insert문으로 데이터를 입력할 때 부서번호가 20번 외에는 데이터가 입력되지 않도록 SQL문을 작성하시오.

```

INSERT INTO (select empno, ename, sal, deptno
From emp
Where deptno = 20
WITH CHECK option)

```

```

WITH CHECK option)
VALUES(&empno, '&ename', &sal, &deptno);

```

(insert할 때 부서번호 20번인거 이외는 막아버린다)

```

Enter value for empno: 2323
Enter value for ename: wonsub
Enter value for sal: 5000
Enter value for deptno: 31
old 5: VALUES(&empno, '&ename', &sal, &deptno)
new 5: VALUES(2323, 'wonsub', 5000, 31)
From emp
*
ERROR at line 2:
ORA-01402: view WITH CHECK OPTION where-clause violation

```

문제385. 직원번호, 이름, 월급, 부서번호를 각각 물어보게 하고 데이터를 입력하는 insert문을 생성하는데 월급이 9000이상으로 입력되지 못하게 insert문을 작성하시오.

```

INSERT INTO (SELECT empno, ename, sal, deptno
FROM EMP
WHERE sal < 9000
WITH CHECK option)
VALUES(&empno, '&ename', &sal, &deptno);

```

문제386. 우리반 테이블에 데이터를 입력하는 insert문을 작성하는데 학생번호, 이름, 나이, 통신사를 입력하게 하고 통신사를 입력할 때 sk, lg, kt, cjh만 입력될 수 있도록 insert문을 작성.

```

INSERT INTO (SELECT empno, ename, age, telecom
FROM EMP2
WHERE telecom IN('sk', 'lg', 'kt', 'cjh')
WITH CHECK option)
VALUES(&empno, '&ename', &age, &telecom);

```

```

.제38 Enter value for empno: 1234
Enter value for ename: wonsub
Enter value for age: 28
Enter value for telecom: qweqwe
old 5:      VALUES(&empno, &ename, &age, &telecom)
new 5:      VALUES(1234, wonsub, 28, qweqwe)
      VALUES(1234, wonsub, 28, qweqwe)
      *
ERROR at line 5:

```

문제387. 학생번호, 이름, 나이, 이메일을 입력하게 하는 insert문을 작성하는데, 이메일을 입력할 때 @와 .이 있어야만 insert 될 수 있도록 작성하시오.

```

INSERT INTO (SELECT empno, ename, age, email
FROM EMP2
WHERE email LIKE '%@%.%'
WITH CHECK option)
VALUES(&empno, '&ename', &age, &email);

```

```

Enter value for empno: 1234
Enter value for ename: wno
Enter value for age: 30
Enter value for email: rladnjs@qwrejnl
old 5:      VALUES(&empno, &ename, &age, &email)
new 5:      VALUES(1234, wno, 30, rladnjs@qwrejnl)
              WHERE email LIKE '%@%.%'
              *
ERROR at line 3:
ORA-00904: "EMAIL": invalid identifier

```

문제388. 학생번호, 이름, 나이, 전공을 입력하는데 insert문을 작성하는데 전공을 입력할 때 맨 뒤에 '학'이 입력되어야 입력될 수 있도록 insert문을 작성하시오.

```

insert INTO(SELECT empno, ename, age, major
            FROM EMP2
            WHERE major LIKE '%학'
            WITH CHECK option)
VALUES(&empno, '&ename', &age, '&major');

```

문제389. 학생번호, 이름, 나이, 주소를 입력하게 하는 insert 문을 작성하는데 주소를 입력할 때, xx시, xx구, xx동 으로 입력될 수 있도록 insert문을 작성하시오.(점심시간)

```

INSERT INTO(SELECT empno, ename, age, address
            FROM EMP2
            WHERE address LIKE '%시 %구 %동'
            WITH CHECK option)
VALUES(&empno, '&ename', &age, '&address');

```

또 다른 답:

```

INSERT INTO(SELECT empno, ename, age, address
            FROM EMP2
            WHERE regexp_like(address, '^.+시\s.+구\s.+동$')
            WITH CHECK option)
VALUES(&empno, '&ename', &age, '&address');

```

구문	설명
.	지원되는 character set에서 NULL을 제외한 임의의 문자와 일치
+	한 번 이상 발생 수 일치
?	0 또는 1번 발생 수 일치
*	선행 하위식의 0번 이상 발생 수 일치
{ m }	선행 표현식의 정확히 m번 발생 수 일치
{ m, }	선행 하위식과 최소 m번 이상 발생 수 일치
{ m, n }	선행 하위식의 최소 m번 이상, 최대 n번 이하 발생 수 일치
[ ... ]	괄호 안의 리스트에 있는 임의의 단일 문자와 일치
	여러 대안 중 하나와 일치
( ... )	괄호로 묶인 표현식을 한 단위로 취급합니다. 하위식은 리터럴의 문자열이나 연산자를 포함한 복잡한 표현식이 될 수 있습니다.

구문	설명
<code>^</code>	문자열 시작 부분과 일치
<code>\$</code>	문자열 끝부분과 일치
<code>\</code>	표현식에서 후속 메타 문자를 리터럴로 처리합니다.
<code>\n</code>	괄호 안에 그룹화된 <i>n</i> 번째 (1~9) 선행 하위식과 일치합니다. 괄호는 표현식이 기억되도록 만들고 backreference에서 표현식을 참조합니다.
<code>\d</code>	숫자 문자
<code>[ : class: ]</code>	지정된 POSIX 문자 클래스에 속한 임의의 문자와 일치
<code>[ ^: class: ]</code>	괄호 안의 리스트에 <u>없는</u> 임의의 단일 문자와 일치

문제390. 학생번호, 이름, 나이, 주소를 입력하게 하는 insert 문을 작성하는데 주소를 입력할 때, xx시, xx구, xx동 으로 입력될 수 있고, xx도 xx시 xx면으로도 입력될 수 있도록 insert문을 작성하시오.

```
INSERT INTO(SELECT empno, ename, age, address
FROM EMP2
WHERE regexp_like(address, '^.(시|도)\s.+(구|시)\s.+(동|면)$')
WITH CHECK option)
VALUES(&empno, '&ename', &age, '&address');
```



## 26. Insert, update문에서 default 사용방법

2018년 4월 16일 월요일    오후 2:06

Default 를 이용하면 특별한 값을 입력하지 않으면 default 값으로 데이터가 입력되게 설정된다.

Ex) alter table emp

Modify sal default 1200;

INSERT INTO emp(empno, ename)

VALUES (3921, 'JACK');    < - sal에 암시적 null을 입력한다.

18	ALLEN	3400
19	JACK	1200

==> default값으로 월급이 입력된다.(암시적일 때 만 되고, 명시적으로 null을 넣어주면 null이된다

문제391. 사원 테이블에 직업에 default값으로 SALESMAN으로 설정되게 하고 확인해보시오.

ALTER TABLE EMP

MODIFY job DEFAULT 'SALESMAN';

INSERT INTO emp(empno, ename)

VALUES (3412, 'WON');

20	WON	SALESMAN
----	-----	----------

## 27. 다중 insert문 4가지

2018년 4월 16일 월요일    오후 2:21

**다중 insert 문 : 데이터를 입력할 때 여러 개의 테이블에 동시에 데이터를 입력할 수 있게 해준다.**

1. 조건부 all insert
2. 조건부 first insert
3. 무조건 all insert
4. Pivoting insert

### 테스트 하기위한 테이블 생성

```
Create table emp2_sk
As
select * from emp2 where 1=2;
Create table emp2_lg
As
select * from emp2 where 1=2;
Create table emp2_kt
As
select * from emp2 where 1=2;
Create table emp2_cjh
As
select * from emp2 where 1=2;
```

1 = 2가 false이기 때문에 데이터를 안가져오고  
쓰겠다는 뜻

**문제392. 우리반 테이블에 텔레콤이 sk인 학생들의 data를 emp2\_sk테이블에 입력하시오.**

```
INSERT INTO emp2_sk(SELECT * FROM EMP2 WHERE telecom = 'sk');
```

```
SELECT * FROM emp2_sk;
```

EMPNO	ENAME	AGE	BIRTH	MAJOR	EMAIL	MOBILE	ADDRESS	TELECOM	BIRTH_DAY
10	김원섭	26	1994-02-08 오전 12:00:00	컴퓨터공학	rladnjstjek@naver.com	010-3308-0492	경기도 남양주시 와부읍	sk	tuesday
8	이유진	25	1994-12-06 오전 12:00:00	시스템경영공학부	yjin1435@gmail.com	010-5920-3050	서울시 동작구 사당동	sk	tuesday
1	윤진민	27	1992-07-15 오전 12:00:00	경제학	yjnm92@gmail.com	010-4844-4224	경기도 구리시 교문동	sk	wednesday
24	김대경	27	1992-01-21 오전 12:00:00	통계학	sbs120200@naver.com	010-7565-2243	경기도 김포시 장기동	sk	tuesday
25	김동윤	28	1991-06-22 오전 12:00:00	경영학	dy3696@naver.com	010-8981-3785	서울시 영등포구 당산로	sk	saturday
15	윤동환	25	1994-12-22 오전 12:00:00	컴퓨터정보시스템	ryeoha_@naver.com	010-6284-6069	경기도 부천시 역곡1동	sk	thursday
21	김건태	28	1991-10-28 오전 12:00:00	환경공학	kgt1028@gmail.com	010-9092-1210	서울시 강서구 등촌동	sk	monday
29	한지윤	24	1995-05-09 오전 12:00:00	경영학	wldbs3803@gmail.com	010-6222-3803	수원시 팔달구 월드컵로	sk	tuesday

### 27-1. 조건부 all insert

**문제393. 다중 insert문을 이용해서 동시에 4개의 테이블의 data가 입력되게 하시오.**

Emp2

Sk----->emp2\_sk

Lg----->emp2\_lg

Kt----->emp2\_kt

Cjh----->emp2\_cjh

INSERT ALL

WHEN telecom = 'sk' THEN INTO emp2\_sk

WHEN telecom = 'lg' THEN INTO emp2\_lg

WHEN telecom = 'kt' THEN INTO emp2\_kt

```

        WHEN telecom = 'cjh' THEN INTO emp2_cjh
SELECT * FROM EMP2;

```

문제394. 아래의 3개의 테이블을 생성하고 데이터를 한번에 입력하시오.

**Emp2\_30 (30대 이상)**

**Emp2\_27 (27~29살)**

**Emp2\_24 (24~26살)**

```

CREATE TABLE emp2_30
AS
SELECT * FROM EMP2 WHERE age >= 30;

CREATE TABLE emp2_27
AS
SELECT * FROM EMP2 WHERE age BETWEEN 27 AND 29;

CREATE TABLE emp2_24
AS
SELECT * FROM EMP2 WHERE age BETWEEN 24 AND 26;

INSERT ALL
    WHEN age >= 30
      THEN INTO emp2_30
    WHEN age BETWEEN 27 AND 29
      THEN INTO emp2_27
    WHEN age BETWEEN 24 AND 26
      THEN INTO emp2_24
SELECT * FROM EMP2;

```

## 27-2. 조건부 first insert

문제395. 위의 테이블들을 truncate 하고 다시 데이터를 입력하는데 나이가 30살 이상인 학생들은 emp2\_sk에 데이터를 입력하고 나머지 데이터인 나이가 30보다 작은 학생들중에 Lg 텔레콤이면 emp2\_lg에 입력하고 kt텔레콤이면 emp2\_kt에 입력하고 cjh이면 emp2\_cjh에 입력하시오.

```

INSERT FIRST
    WHEN age >= 30 THEN INTO emp2_sk
    WHEN telecom = 'lg' THEN INTO emp2_lg
    WHEN telecom = 'kt' THEN INTO EMP2_kt
    WHEN telecom = 'cjh' THEN INTO emp2_cjh
SELECT * FROM EMP2;

```

위에 조건에 먼저 걸려진 정보는 아래에 있는 정보에서 걸러 나온다.

## 27-3. 무조건 all insert문

문제396. emp2 테이블의 모든 데이터가 조건없이 아래의 테이블들에 동시에 입력되게 하시오.

Source table

Target table

```

Emp2 -----> emp2_나
                emp2_lg
                Emp2_kt
                Emp2_chj

```

```

INSERT ALL
    INTO emp2_sk
    INTO emp2_lg
    INTO emp2_kt
    INTO emp2_cjh
SELECT * FROM EMP2;

```

Insert all into 타겟 테이블이름

## 27-4. Pivoting insert 문

### : 컬럼 --> 데이터가 되어서 insert가 되는 구문

사용시기) 데이터 분석을 위해서 수집된 데이터가 엑셀 형태일때 이 엑셀 파일을 db에 저장해서 분석하고자 할때.

문제397. 부서번호, 부서번호별 토탈 월급을 출력 하시오.(가로로 출력하시오)

```
SELECT SUM(DECODE(deptno, 10, sal,0)) "10",  
       SUM(DECODE(deptno, 20, sal,0)) "20",  
       SUM(DECODE(deptno, 30, sal,0)) "30"  
FROM EMP;
```

	10	20	30
1	8750	10875	9400

문제398. 위의 결과를 ctas를 이용해서 dept\_sumsal 이라는 이름으로 테이블을 생성하시오.

Create table dept\_sumsal

As

```
SELECT SUM(DECODE(deptno, 10, sal,0)) "10",  
       SUM(DECODE(deptno, 20, sal,0)) "20",  
       SUM(DECODE(deptno, 30, sal,0)) "30"  
FROM EMP;
```

```
SELECT * FROM dept_sumsal;
```

	10	20	30
1	8750	10875	9400

문제399. 398번에서 만든 dept\_sumsal 테이블을 조회하는데 unpivot문을 이용해서 아래와 같이 결과가 출력되게하시오.

```
SELECT *
```

```
FROM DEPT_sumsal
```

unpivot (sumsal FOR deptno IN("10", "20", "30")); <-- 숫자 타입은 문자로 변환해야지 인식함

	DEPTNO	SUMSAL
1	10	8750
2	20	10875
3	30	9400

문제400. order2 테이블을 unpivot문을 이용해서 아래와 같이 결과를 출력하시오

```
SELECT * FROM order2;
```

	ENAME	BICYCLE	CAMERA	NOTEBOOK
1	SMITH	2	3	1
2	ALLEN	1	2	3
3	KING	3	2	2

||



	ENAME	ITEM	CNT
1	SMITH	BICYCLE	2
2	SMITH	CAMERA	3
3	SMITH	NOTEBOOK	1
4	ALLEN	BICYCLE	1
5	ALLEN	CAMERA	2
6	ALLEN	NOTEBOOK	3
7	KING	BICYCLE	3
8	KING	CAMERA	2
9	KING	NOTEBOOK	2

SELECT \*

FROM ORDER2

unpivot (cnt For item IN (BICYCLE, CAMERA, NOTEBOOK)); <-- 문자 타입은 문자로 바로 인식함  
(소문자로 써도됨)

문제401. 서울시에서 2014년에 살인사건이 가장 많이 일어난 지역과 그 순위를 아래와 같이 출력하시오.(crime\_age를 이용)

	LOCAL	SUM(SUM_TOT)	순위
1	영등포	14	1
2	동대문	12	2
3	종랑	11	3
4	송파	9	4
5	강서	8	5
6	강남	8	5
7	광진	8	5
8	관악	7	6
9	금천	7	6
10	도봉	6	7
11	성북	6	7
12	강북	6	7
13	강동	5	8

SELECT LOCAL, SUM(SUM\_tot), dense\_RANK() OVER (ORDER BY SUM(sum\_tot) desc) 순위  
FROM crime\_age  
WHERE TYPE = '살인' AND year ='2014'  
GROUP BY LOCAL; <-- 남,여가 나뉘져 있어서

문제402. crime\_age 테이블로 아래와 같이 결과를 출력하시오.

36	종로	폭력	789	남	UNDER_40	150
37	종로	폭력	789	남	UNDER_50	167
38	종로	폭력	789	남	UNDER_60	185
39	종로	폭력	789	남	OVER_60	107
40	종로	폭력	789	남	UNKONWN	0
41	남대문	살인	1	남	UNDER_6	0
42	남대문	살인	1	남	UNDER_12	0
43	남대문	살인	1	남	UNDER_15	0
44	남대문	살인	1	남	UNDER_20	0
45	남대문	살인	1	남	UNDER_30	0
46	남대문	살인	1	남	UNDER_40	0

SELECT LOCAL, type, sum\_tot, gender, age, cnt  
FROM crime\_age  
unpivot( cnt for age in (under\_6, under\_12,  
UNDER\_15, under\_20, under\_30,  
under\_40, under\_50, under\_60,  
over\_60, UNKonWN));

문제403. 2014년도에 살인을 일으킨 나이대, 나이대별 살인건수를 출력하시오.

```
SELECT age, SUM(cnt)
FROM (
SELECT LOCAL, type, sum_tot, gender, age, cnt
FROM crime_age
unpivot( cnt for age in (under_6, under_12,
                        UNDER_15, under_20, under_30,
                        under_40, under_50, under_60,
                        over_60, UNKonWN) )
WHERE TYPE = '살인' AND year = '2014'
)
GROUP BY age
ORDER BY SUM(cnt) desc;
```

	AGE	SUM(CNT)
1	UNDER_60	39
2	UNDER_50	37
3	OVER_60	32
4	UNDER_40	25
5	UNDER_30	16
6	UNDER_6	5
7	UNDER_20	2
8	UNDER_12	1
9	UNKONWN	1
10	UNDER_15	0

위의 결과를 pivoting insert를 쓰면

문제404. pivoting insert문을 이용해서 나이대에 대한 정보를 테이블의 data로 만드시오.

```
create table crime_age2
( local varchar2(100),
  type varchar2(20),
  sum_tot number(10),
  age_type varchar2(20),
  cnt number(10) );
```

테이블 생성 후 테이블 추가

```
insert all
into crime_age2 values(local, type, sum_tot, 'under_6', under_6)
into crime_age2 values(local, type, sum_tot, 'under_12', under_12)
into crime_age2 values(local, type, sum_tot, 'under_15', under_15)
into crime_age2 values(local, type, sum_tot, 'under_20', under_20)
into crime_age2 values(local, type, sum_tot, 'under_30', under_30)
into crime_age2 values(local, type, sum_tot, 'under_40', under_40)
into crime_age2 values(local, type, sum_tot, 'under_50', under_50)
into crime_age2 values(local, type, sum_tot, 'under_60', under_60)
into crime_age2 values(local, type, sum_tot, 'over_60', over_60)
select local, type, sum_tot, under_6, under_12, under_15,
      under_20, under_30, under_40, under_50,
      under_60, over_60
from crime_age;

SELECT * FROM crime_age2;
```

	LOCAL	TYPE	SUM_TOT	AGE_TYPE	CNT
1	중부	살인	1	under_50	0
2	중부	폭력	901	under_50	194
3	종로	살인	2	under_50	1
4	종로	폭력	789	under_50	167
5	남대문	살인	1	under_50	0
6	남대문	폭력	651	under_50	174
7	서대문	살인	3	under_50	0
8	서대문	폭력	1200	under_50	219
9	혜화	살인	0	under_50	0
10	혜화	폭력	688	under_50	170
11	용산	살인	1	under_50	0
12	용산	폭력	1322	under_50	319
13	성북	살인	4	under_50	1
14	성북	폭력	716	under_50	120

문제405. 서초에서 가장 많이 일어나는 범죄가 무엇인지 1~3위까지 출력하시오.

```

SELECT *
FROM(
    SELECT TYPE, SUM(sum_tot), RANK() OVER(ORDER BY SUM(sum_tot) desc) 순위
    FROM crime_age2
    WHERE LOCAL = '서초'
    GROUP BY TYPE
)
WHERE 순위 BETWEEN 1 AND 3;

```

	TYPE	SUM(SUM_TOT)	순위
1	폭력	28872	1
2	살인	45	2

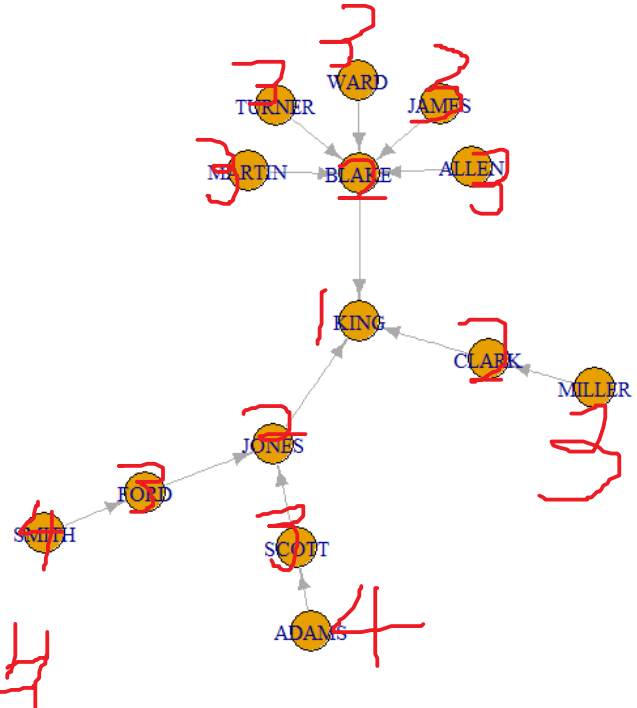
## 28. 계층형 질의문

2018년 4월 17일 화요일 오후 1:50

계층형 질의문 : 순위와 서열을 출력하는 SQL문 (짜궁 sibling, sys\_connect\_by\_path)

Ex) select level, empno, ename, mgr  
From emp  
Start with ename = 'KING'  
Connect by prior empno = mgr;

LEVEL	EMPNO	ENAME	MGR
1	7839	KING	(null)
2	7566	JONES	7839
3	7788	SCOTT	7566
4	7876	ADAMS	7788
5	7902	FORD	7566
6	7369	SMITH	7902
7	7698	BLAKE	7839
8	7499	ALLEN	7698
9	7521	WARD	7698
10	7654	MARTIN	7698
11	7844	TURNER	7698
12	7900	JAMES	7698
13	7782	CLARK	7839
14	7934	MILLER	7782



문제406. 위의 계층형 질의문을 서열에 맞춰서 더 잘 구분될 수 있도록 작성하시오.

```
select rpad(' ', level*2) || ename employee
From emp
Start with ename = 'KING'
Connect by prior empno = mgr;
```

<- 공백을 level\*2만큼 채워넣어라

EMPLOYEE
1 KING
2 JONES
3 SCOTT
4 ADAMS
5 FORD
6 SMITH
7 BLAKE
8 ALLEN
9 WARD
10 MARTIN
11 TURNER
12 JAMES
13 CLARK
14 MILLER

KING 밑에 JONES, BLAKE, CLARK  
그 밑에 SCOTT, FROD, ALLEN...

문제407. 위의 결과에서 BLAKE는 제외하고 출력하시오.

```
select rpad(' ', level*2) || ename employee
From emp
```



```
WHERE ename != 'BLAKE'
Start with ename = 'KING'
Connect by prior empno = mgr;
```

EMPLOYEE
1 KING
2 JONES
3 SCOTT
4 ADAMS
5 FORD
6 SMITH
7 ALLEN
8 WARD
9 MARTIN
10 TURNER
11 JAMES
12 CLARK
13 MILLER

문제408. 위의 결과에서 BLAKE 팀원들도 제외하시오.

```
select rpad(' ',level*2) || ename employee
From emp
Start with ename = 'KING'
Connect by prior empno = mgr
AND ename != 'BLAKE';
```

EMPLOYEE
1 KING
2 JONES
3 SCOTT
4 ADAMS
5 FORD
6 SMITH
7 CLARK
8 MILLER

문제409. 계층형 질의문으로 이름과 월급을 출력하는데 월급이 높은 사원부터 출력하시오.

```
select rpad(' ',level*2) || ename employee, sal
From emp
Start with ename = 'KING'
Connect by prior empno = mgr
ORDER BY sal desc;
```

	EMPLOYEE	SAL
1	KING	5000
2	SCOTT	3000
3	FORD	3000
4	JONES	2975
5	BLAKE	2850
6	CLARK	2450
7	ALLEN	1600
8	TURNER	1500
9	MILLER	1300
10	WARD	1250
11	MARTIN	1250
12	ADAMS	1100
13	JAMES	950
14	SMITH	800

==> 서열이 깨져서 계층형 질의문의 의미가 사라짐  
(누가 누구의 팀원인지 구별 못함)

문제410. 위의 문제에서 서열이 깨지지 않도록 출력하시오.

```
select rpad(' ',level*2) || ename employee, sal
From emp
Start with ename = 'KING'
Connect by prior empno = mgr
ORDER siblings BY sal desc;
```

	EMPLOYEE	SAL
1	KING	5000
2	JONES	2975
3	SCOTT	3000
4	ADAMS	1100
5	FORD	3000
6	SMITH	800
7	BLAKE	2850
8	ALLEN	1600
9	TURNER	1500
10	WARD	1250
11	MARTIN	1250
12	JAMES	950
13	CLARK	2450
14	MILLER	1300

서열중 에서 월급이 높은 사원들 먼저 나온다.

문제411. 계층형 질의문과 짝꿍 함수인 sys\_connect\_by\_path 를 이용해서 SQL문을 작성하시오.

```
SELECT ename,
       sys_connect_by_path(ename, '/') AS path
From emp
Start with ename = 'KING'
Connect by prior empno = mgr;
```

	ENAME	PATH
1	KING	/KING
2	JONES	/KING/JONES
3	SCOTT	/KING/JONES/SCOTT
4	ADAMS	/KING/JONES/SCOTT,
5	FORD	/KING/JONES/FORD
6	SMITH	/KING/JONES/FORD/
7	BLAKE	/KING/BLAKE

JONES은 KING밑에 ,  
SCOTT은 KING-JONES밑에 .....

<input type="checkbox"/>	ENAME	PATH
1	KING	/KING
2	JONES	/KING/JONES
3	SCOTT	/KING/JONES/SCOTT
4	ADAMS	/KING/JONES/SCOTT,
5	FORD	/KING/JONES/FORD
6	SMITH	/KING/JONES/FORD/
7	BLAKE	/KING/BLAKE
8	ALLEN	/KING/BLAKE/ALLEN
9	WARD	/KING/BLAKE/WARD
10	MARTIN	/KING/BLAKE/MARTIN
11	TURNER	/KING/BLAKE/TURNEF
12	JAMES	/KING/BLAKE/JAMES
13	CLARK	/KING/CLARK
14	MILLER	/KING/CLARK/MILLER

JONES은 KING밑에 ,  
SCOTT은 KING-JONES밑에 .....

문제412. 위의 결과에서 앞에 / 를 빼고 출력하시오.

```
SELECT ename,
       ITRIM(sys_connect_by_path(ename, '/'), '/') AS path
  From emp
 Start with ename = 'KING'
 Connect by prior empno = mgr;
```

<input type="checkbox"/>	ENAME	PATH
1	KING	KING
2	JONES	KING/JONES
3	SCOTT	KING/JONES/SCOTT
4	ADAMS	KING/JONES/SCOTT,
5	FORD	KING/JONES/FORD
6	SMITH	KING/JONES/FORD/
7	BLAKE	KING/BLAKE
8	ALLEN	KING/BLAKE/ALLEN
9	WARD	KING/BLAKE/WARD
10	MARTIN	KING/BLAKE/MARTIN
11	TURNER	KING/BLAKE/TURNEF
12	JAMES	KING/BLAKE/JAMES
13	CLARK	KING/CLARK
14	MILLER	KING/CLARK/MILLER

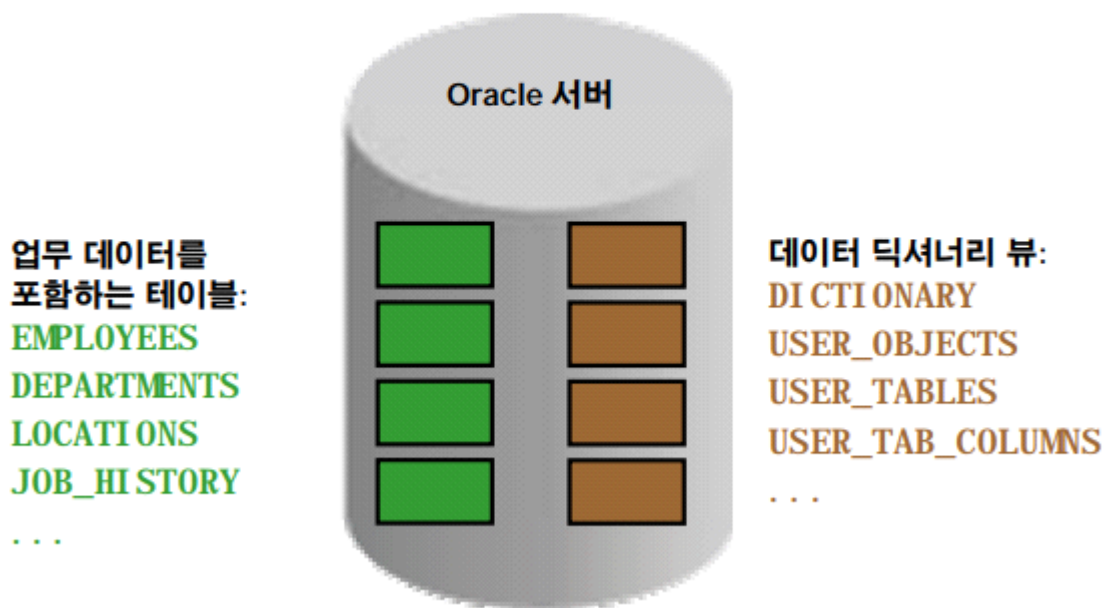
## 29. 데이터 디렉터리를 사용한 객체관리

2018년 4월 17일 화요일    오후 3:02

### 데이터 디렉터리

: 데이터 베이스의 오브젝트에 대한 정보가 들어있는 파일

### 데이터 디렉터리



문제413. 지금 내가 접속한 SCOTT 세션이 가지고 있는 테이블 리스트를 조회하시오.

Select \*

From user\_tables;

TABLE_NAME	TABLESPACE_NAME	CLUSTER_NAME
EMP05	SYSTEM	(null)
ORDER2	SYSTEM	(null)
CRIME_LOC	SYSTEM	(null)
CRIME_CAUSE	SYSTEM	(null)
EMP7	SYSTEM	(null)
SUCIDE	SYSTEM	(null)
CRIME_CAUSE2	SYSTEM	(null)
PRICE	SYSTEM	(null)
SALGRADE	SYSTEM	(null)
TELECOM_PRICE	SYSTEM	(null)
BONUS	SYSTEM	(null)
EMP8	SYSTEM	(null)
UNIV	SYSTEM	(null)
EMP20	SYSTEM	(null)

문제414. 내가 가지고 있는 테이블이 언제 만들어 졌는데 조회하시오.

SELECT object\_name, created

FROM user\_objects

□	OBJECT_NAME	CREATED
1	EMP05	2018-03-29 오후 3:03:53
2	ORDER2	2018-04-02 오후 2:09:50
3	CRIME_LOC	2018-04-02 오후 2:23:48
4	CRIME_CAUSE	2018-04-02 오후 3:01:19
5	EMP7	2018-04-02 오후 3:15:34
6	SUCIDE	2018-04-02 오후 4:33:38
7	CRIME_CAUSE2	2018-04-02 오후 3:31:23
8	PRICE	2018-04-02 오후 4:09:38
9	SALGRADE	2018-04-03 오후 1:54:41
10	TELECOM_PRICE	2018-04-03 오후 4:29:18
11	BONUS	2018-04-04 오전 10:26:0
12	EMP8	2018-04-05 오후 2:10:25
13	UNIV	2018-04-05 오후 4:08:42
14	EMP20	2018-04-09 오전 11:05:3

```
Select *
From user_views;
```

```
Select *
From user_indexes;
```

[illegible]

```
Select *
  From user_sequences;
```

```
SELECT *
FROM user_tab_comments;
```

SQL 활용 페이지 181

TABLE_NAME	TABLE_TYPE	COMMENTS
EMP05	TABLE	(null)
ORDER2	TABLE	(null)
CRIME_LOC	TABLE	(null)
CRIME_CAUSE	TABLE	(null)
EMP7	TABLE	(null)
SUCIDE	TABLE	(null)
CRIME_CAUSE2	TABLE	(null)
PRICE	TABLE	(null)
SALGRADE	TABLE	(null)
TELECOM_PRICE	TABLE	(null)
BONUS	TABLE	(null)
EMP8	TABLE	(null)

## %% 테이블 커멘트 다는 방법

Comment on TABLE emp is '사원에 대한 월급과 직업과 부서번호등에 대한 정보가 있는 테이블입니다';

TABLE_NAME	TABLE_TYPE	COMMENTS
CRIME_AGE2	TABLE	(null)
CRIME_AGE	TABLE	(null)
DEPT_SUMSAL	TABLE	(null)
CRIME_TIME	TABLE	(null)
EMP	TABLE	사원에 대한 월급과 직업과 부서번호등에 대한 정보가 있는 테이블입니다
DEPT	TABLE	(null)

문제420. dept 테이블에 대한 테이블 comment를 생성하시오.

COMMENT ON TABLE DEPT IS '부서번호, 부서이름, 부서위치의 정보가 있는 테이블입니다.;

TABLE_NAME	TABLE_TYPE	COMMENTS
CRIME_AGE2	TABLE	(null)
CRIME_AGE	TABLE	(null)
DEPT_SUMSAL	TABLE	(null)
CRIME_TIME	TABLE	(null)
EMP	TABLE	사원에 대한 월급과 직업과 부서번호등에 대한 정보가 있는 테이블
DEPT	TABLE	부서번호, 부서이름, 부서위치의 정보가 있는 테이블입니다.
FMP2 30	TARI F	(null)

## 데이터 디렉너리의 종류

뷰 접두어	목적
USER	유저의 뷰(유저의 스키마에 있는 내용, 유저가 소유한 내용)
ALL	확장된 유저의 뷰(유저가 액세스할 수 있는 내용)
DBA	데이터베이스 관리자의 뷰(모든 사람의 스키마에 있는 내용)
VS	성능 관련 데이터

User\_xxx : 내가 소유한 객체를 볼 수 있음

All\_xxx : 내가 소유한 객체 + 다른 유저가 권한을 부여한 객체를 볼 수 있음

Db\*\_xxx : 데이터 베이스에 모든 객체를 볼 수 있음

```
SELECT COUNT(*) FROM dba_tables;
```

	COUNT(*)
1	1728

## %%데이터 디렉터리를 조회할 수 있는 사전

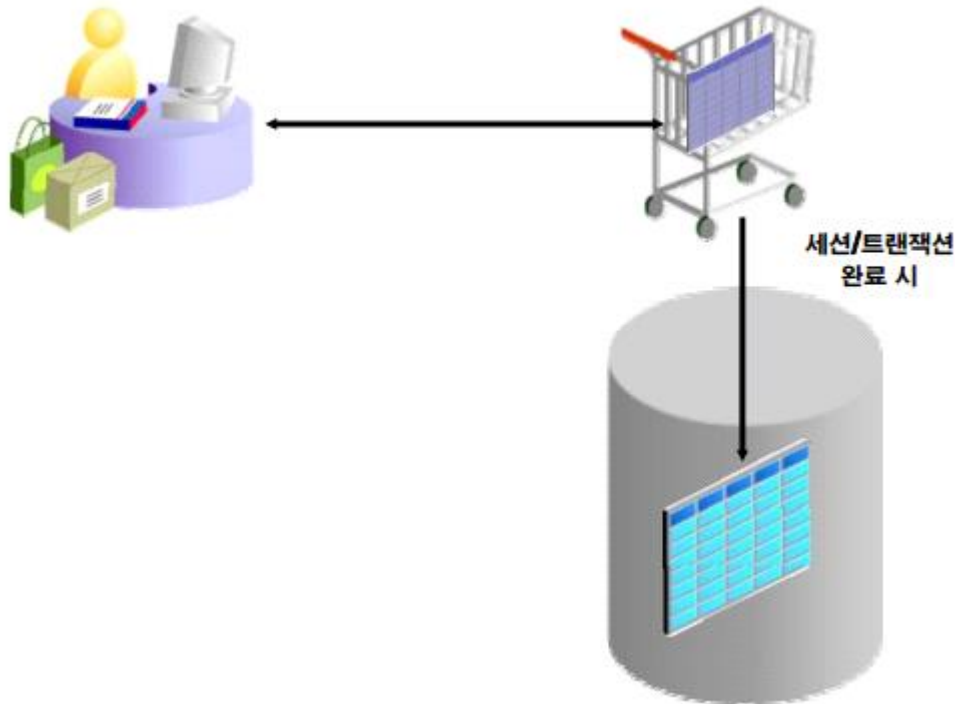
```
SELECT * FROM DICTIONARY;
```

	TABLE_NAME	COMMENTS
1	USER_CONS_COLUMNS	Information about accessible columns in constraint d
2	ALL_CONS_COLUMNS	Information about accessible columns in constraint d
3	DBA_CONS_COLUMNS	Information about accessible columns in constraint d
4	USER_LOG_GROUP_COLUMNS	Information about columns in log group definitions
5	ALL_LOG_GROUP_COLUMNS	Information about columns in log group definitions
6	DBA_LOG_GROUP_COLUMNS	Information about columns in log group definitions
7	USER_LOBS	Description of the user's own LOBs contained in the i
8	ALL_LOBS	Description of LOBs contained in tables accessible to
9	DBA_LOBS	Description of LOBs contained in all tables
10	USER_CATALOG	Tables, Views, Synonyms and Sequences owned by
11	ALL_CATALOG	All tables, views, synonyms, sequences accessible to
12	DBA_CATALOG	All database Tables, Views, Synonyms, Sequences
13	USER_CLUSTERS	Descriptions of user's own clusters

## 30. 임시 테이블

2018년 4월 17일 화요일    오후 4:03

**임시 테이블 : 데이터를 영구히 저장하는게 아니라 임시로 데이터를 저장하는 테이블**



### 임시 테이블의 종류 2가지

1. 세션이 종료 될 때까지 데이터를 저장  
On commit preserve rows
2. 트랜잭션이 종료 될 때까지 데이터를 저장  
On commit delete rows

### 30-1. 세션이 종료 될 때까지 데이터를 저장하는 임시테이블

Ex)

```
CREATE GLOBAL TEMPORARY TABLE emp_temp  
  (empno NUMBER(10),  
   ename VARCHAR2(10),  
   SAL NUMBER(10) )  
ON COMMIT preserve ROWS;
```

```
INSERT INTO emp_temp  
  SELECT empno, ename, sal  
  FROM EMP;
```

```
SELECT * FROM emp_temp;
```



	EMPNO	ENAME	SAL
1	7839	KING	5000
2	7698	BLAKE	2850
3	7782	CLARK	2450
4	7566	JONES	2975
5	7654	MARTIN	1250
6	7499	ALLEN	1600
7	7844	TURNER	1500
8	7900	JAMES	950
9	7521	WARD	1250
10	7902	FORD	3000

나갔다가 다시 들어와서 보면 데이터가 없음

## 30-2. 트랜잭션이 종료 될 때까지 데이터를 저장

문제421. commit을 하면 데이터가 사라지는 임시 테이블을 위의 테이블과 똑같이 생성하시오.

```
CREATE GLOBAL TEMPORARY TABLE emp_temp
(empno NUMBER(10),
ename VARCHAR2(10),
SAL NUMBER(10) )
ON COMMIT delete ROWS;
```

```
INSERT INTO emp_temp
SELECT empno, ename, sal
FROM EMP;
```

```
SELECT * FROM EMP_TEMP;
```

	EMPNO	ENAME	SAL
1	7839	KING	5000
2	7698	BLAKE	2850
3	7782	CLARK	2450
4	7566	JONES	2975
5	7654	MARTIN	1250
6	7499	ALLEN	1600
7	7844	TURNER	1500

Commit; 하고

```
SELECT * FROM EMP_TEMP;
```

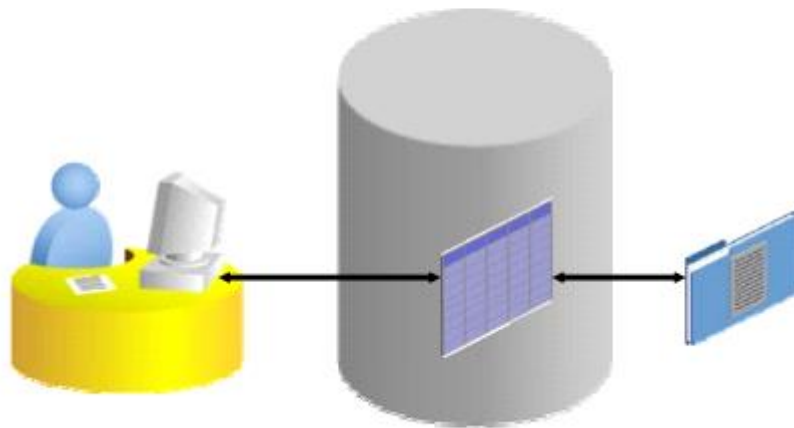
	EMPNO	ENAME	SAL
표시할 데이터가 없습니다.			



## 31. External 테이블

2018년 4월 17일 화요일    오후 4:23

**External 테이블** : db외부의 text나 csv 파일을 db에 입력하지 않고 링크만 걸어서 액세스하는 테이블



Ex) 1. **emp.txt**를 **D 드라이브** 밑에 둔다.

2. **emp.txt** 가 존재하는 위치가 어디다 라는 것을 알려줄 디렉터리를 생성한다.

```
Create DIRECTORY emp_dir  
As 'D:\';
```

3. **external** 테이블을 만든다.

```
create table ext_emp11  
( EMPNO NUMBER(10),  
  ENAME VARCHAR2(20),  
  JOB VARCHAR2(20),  
  MGR NUMBER(10),  
  HIREDATE DATE,  
  SAL NUMBER(10),  
  COMM NUMBER(10),  
  DEPTNO NUMBER(10))
```

**organization external**

(type oracle\_loader

**default directory emp\_dir**

access parameters

(records delimited by newline

외부 테이블 생성

Text 파일의 위치가 emp\_dir 이다

Text 파일의 라인과 라인은 엔터로 구분한다

access parameters  
 (records delimited by newline  
 fields terminated by ", "  
 (empno char,  
 ename char,  
 job char,  
 mgr char,  
 hiredate date "yyyy/mm/dd",  
 sal char,  
 comm char ,  
 deptno char ) )  
 location ('emp.txt') );

컬럼과 컬럼은 쉼표로 구분

Text file의 이름을 지정

SELECT \* FROM ext\_emp11;

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESIDENT	0	1981-11-17 오전 12:00:00	5000	0	10
2	7698	BLAKE	MANAGER	7839	1981-05-01 오전 12:00:00	2850	0	30
3	7782	CLARK	MANAGER	7839	1981-05-09 오전 12:00:00	2450	0	10
4	7566	JONES	MANAGER	7839	1981-04-01 오전 12:00:00	2975	0	20
5	7654	MARTIN	SALESMAN	7698	1981-09-10 오전 12:00:00	1250	1400	30
6	7499	ALLEN	SALESMAN	7698	1981-02-11 오전 12:00:00	1600	300	30
7	7844	TURNER	SALESMAN	7698	1981-08-21 오전 12:00:00	1500	0	30
8	7900	JAMES	CLERK	7698	1981-12-11 오전 12:00:00	950	0	30

문제422. 앞의 예제를 활용해서 dept.txt 를 생성하고 dept.txt를 액세스하는 외부 테이블을 Dept\_external로 생성하시오.(마지막 문제)

```
CREATE TABLE ext_DEPT11
  (DEPTNO number(10),
   DNAME VARCHAR2(20),
   LOC VARCHAR2(20) )
  organization external
  (type oracle_loader
   default directory dept_dir
   access parameters
   (records delimited by newline
    fields terminated by ", "
   (deptno CHAR,
    dname char,
    LOC char) )
   location ('dept.txt') );
```

SELECT \* FROM ext\_dept11;

	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	CHICAGO
4	40	OPERATIONS	BOSTON

## 32. 날짜함수

2018년 4월 18일 수요일 오전 9:49

날짜 함수 : 기본 날짜 함수 외에 다른 날짜 함수를 소개

다음 함수 사용:

- EXTRACT
- TZ\_OFFSET
- FROM\_TZ
- TO\_TIMESTAMP
- TO\_YMINTERVAL
- TO\_DSINTERVAL

%% 기본 날짜 함수의 종류

1. Months\_between
2. Add\_months
3. Next\_day
4. Last\_day

문제423. 이름, 입사일, 입사한 년도, 입사한 달을 출력하시오.

```
SELECT ename, hiredate, TO_CHAR(hiredate, 'RRRR'),  
       TO_CHAR(hiredate, 'MM')
```

FROM EMP;

	ENAME	HIREDATE	TO_CHAR(HIREDATE,'RRRR')	TO_CHAR(HIREDATE,'MM')
1	KING	1981-11-17 오전 12:00:00	1981	11
2	BLAKE	1981-05-01 오전 12:00:00	1981	05
3	CLARK	1981-05-09 오전 12:00:00	1981	05
4	JONES	1981-04-01 오전 12:00:00	1981	04
5	MARTIN	1981-09-10 오전 12:00:00	1981	09
6	ALLEN	1981-02-11 오전 12:00:00	1981	02
7	TURNER	1981-08-21 오전 12:00:00	1981	08

문제424. 입사한 년도, 입사한 년도별 토탈월급을 출력하시오.

```
SELECT TO_CHAR(hiredate, 'RRRR'), SUM(sal)  
FROM EMP  
GROUP BY TO_CHAR(hiredate, 'RRRR');
```

	TO_CHAR(HIREDATE,'RRRR')	SUM(SAL)
1	1980	800
2	1983	1100
3	1982	4300
4	1981	22825

- extract 함수

## : 날짜에서 연도나 달, 일 등을 추출하는 함수

Ex) SELECT **extract**(year FROM hiredate)  
FROM EMP;

	EXTRACT(YEARFROMHIREDATE)
1	1981
2	1981
3	1981
4	1981
5	1981
6	1981
7	1981
8	1981
9	1981
10	1981
11	1980
12	1982
13	1983
14	1982

문제425. 입사한 년도, 입사한 년도별 토달월급을 출력하시오.(extract함수로)

```
SELECT extract(year FROM hiredate), sum(sal)
FROM EMP
GROUP BY extract(year FROM hiredate);
```

	EXTRACT(YEARFROMHIREDATE)	SUM(SAL)
1	1982	4300
2	1983	1100
3	1980	800
4	1981	22825

문제426. 우리반 테이블에서 월별 생일자가 몇 명인지 출력하시오(월, 건수 출력)

```
SELECT extract(MONTH FROM birth) 월, COUNT(*) 건수
FROM EMP2
GROUP BY extract(month FROM birth);
```

	월	건수
1	1	6
2	(null)	4
3	6	2
4	2	2
5	4	3
6	5	3
7	7	3
8	12	4
9	10	4
10	9	3

문제427. extract 함수로 추출할 수 있는 날짜 포맷이 어떤 것인지 확인하시오.

```
SELECT extract(day FROM birth) 일, COUNT(*) 건수
FROM EMP2
GROUP BY extract(day FROM birth);
```

	일	건수
--	---	----

	일	건수
1	22	4
2	1	1
3	25	1
4	(null)	4
5	6	1
6	13	2
7	11	1
8	28	1
9	26	1

== year, month, day 까지 가능하다.

## - TZ\_OFFSET

: 나라별 시간대를 출력하는 함수 (영국 기준)

Ex) SELECT tz\_offset('europe/london')  
FROM dual;

	TZ_OFFSET('EUROPE/LONDON')
1	+01:00

문제428. 우리나라 시간대를 확인하시오.

참고 디렉터리 : select \* from v\$timezone\_names;

select \* from v\$timezone\_names  
WHERE LOWER(tzname) LIKE '%seoul%';

	TZNAME	TZABBREV
1	Asia/Seoul	LMT
2	Asia/Seoul	KST
3	Asia/Seoul	KDT

SELECT tz\_offset('asia/seoul')  
from dual;

	TZ_OFFSET('ASIA/SEOUL')
1	+09:00

문제429. 프랑스 파리와 우리나라와 몇시간 차이인지 확인하시오.

SELECT tz\_offset('europe/paris')  
FROM dual;

	TZ_OFFSET('EUROPE/PARIS')
1	+02:00

--> 7 시간 차이남

## - from\_tz

: 날짜를 특정 나라의 날짜 형식으로 출력하는데 시간대까지 같이 출력하는 함수

Ex)  
SELECT from\_tz(timestamp '2018-04-18 10:32:00', 'asia/seoul')  
FROM dual;

Ex)

```
SELECT from_tz(timestamp '2018-04-18 10:32:00', 'asia/seoul')
FROM dual;
```

	FROM_TZ(TIMESTAMP'2018-04-18 10:32:00','ASIA/SEOUL')
1	2018-04-18 오전 1:32:00

```
SELECT from_tz(timestamp '2018-04-18 10:32:00', 'europe/paris')
FROM dual;
```

	FROM_TZ(TIMESTAMP'2018-04-18 10:32:00','EUROPE/PARIS')
1	2018-04-18 오전 8:32:00

## - to\_timestamp 함수

: 시, 분, 초 까지 정교하게 검색을 할때 사용하는 함수

```
Ex) SELECT * FROM EMP
      AS OF timestamp
      to_timestamp('2018/04/18 10:45:00',
                  'RRRR/MM/DD HH24:MI:SS');
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESIDENT	(null)	1981-11-17 오전 12:00:00	5000	(null)	10
2	7698	BLAKE	MANAGER	7839	1981-05-01 오전 12:00:00	2850	(null)	30
3	7782	CLARK	MANAGER	7839	1981-05-09 오전 12:00:00	2450	(null)	10
4	7566	JONES	MANAGER	7839	1981-04-01 오전 12:00:00	2975	(null)	20
5	7654	MARTIN	SALESMAN	7698	1981-09-10 오전 12:00:00	1250	1400	30
6	7499	ALLEN	SALESMAN	7698	1981-02-11 오전 12:00:00	1600	300	30
7	7844	TURNER	SALESMAN	7698	1981-08-21 오전 12:00:00	1500	0	30
8	7900	JAMES	CLERK	7698	1981-12-11 오전 12:00:00	950	(null)	30
9	7521	WARD	SALESMAN	7698	1981-02-23 오전 12:00:00	1250	500	30
10	7902	FORD	ANALYST	7566	1981-12-11 오전 12:00:00	3000	(null)	20
11	7369	SMITH	CLERK	7902	1980-12-09 오전 12:00:00	800	(null)	20

==> 이전의 시간 데이터로 돌아간다.

## ## flashback query 테스트

```
DELETE FROM EMP;
```

COMMIT; 한 후의 위의 쿼리를 돌려본다.

```
SELECT * FROM EMP;
```

표시할 데이터가 없습니다.

```
SELECT * FROM EMP
```

AS OF timestamp <--flashback 문법

```
to_timestamp('2018/04/18 10:45:00',
            'RRRR/MM/DD HH24:MI:SS');
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESIDENT	(null)	1981-11-17 오전 12:00:00	5000	(null)	10
2	7698	BLAKE	MANAGER	7839	1981-05-01 오전 12:00:00	2850	(null)	30
3	7782	CLARK	MANAGER	7839	1981-05-09 오전 12:00:00	2450	(null)	10
4	7566	JONES	MANAGER	7839	1981-04-01 오전 12:00:00	2975	(null)	20
5	7654	MARTIN	SALESMAN	7698	1981-09-10 오전 12:00:00	1250	1400	30



	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESIDENT	(null)	1981-11-17 오전 12:00:00	5000	(null)	10
2	7698	BLAKE	MANAGER	7839	1981-05-01 오전 12:00:00	2850	(null)	30
3	7782	CLARK	MANAGER	7839	1981-05-09 오전 12:00:00	2450	(null)	10
4	7566	JONES	MANAGER	7839	1981-04-01 오전 12:00:00	2975	(null)	20
5	7654	MARTIN	SALESMAN	7698	1981-09-10 오전 12:00:00	1250	1400	30
6	7499	ALLEN	SALESMAN	7698	1981-02-11 오전 12:00:00	1600	300	30
7	7844	TURNER	SALESMAN	7698	1981-08-21 오전 12:00:00	1500	0	30
8	7900	JAMES	CLERK	7698	1981-12-11 오전 12:00:00	950	(null)	30
9	7521	WARD	SALESMAN	7698	1981-02-23 오전 12:00:00	1250	500	30
10	7902	FORD	ANALYST	7566	1981-12-11 오전 12:00:00	3000	(null)	20
11	7369	SMITH	CLERK	7902	1980-12-09 오전 12:00:00	800	(null)	20
12	7788	SCOTT	ANALYST	7566	1982-12-22 오전 12:00:00	3000	(null)	20
13	7876	ADAMS	CLERK	7788	1983-01-15 오전 12:00:00	1100	(null)	20
14	7934	MILLER	CLERK	7782	1982-01-11 오전 12:00:00	1300	(null)	10

문제430. flashback query를 이용해서 지워진 emp 데이터를 복구하시오.

INSERT into EMP

SELECT \* FROM EMP

AS OF timestamp

to\_timestamp('2018/04/18 10:45:00',

'RRRR/MM/DD HH24:MI:SS');

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESIDENT	(null)	1981-11-17 오전 12:00:00	5000	(null)	10
2	7698	BLAKE	MANAGER	7839	1981-05-01 오전 12:00:00	2850	(null)	30
3	7782	CLARK	MANAGER	7839	1981-05-09 오전 12:00:00	2450	(null)	10
4	7566	JONES	MANAGER	7839	1981-04-01 오전 12:00:00	2975	(null)	20
5	7654	MARTIN	SALESMAN	7698	1981-09-10 오전 12:00:00	1250	1400	30
6	7499	ALLEN	SALESMAN	7698	1981-02-11 오전 12:00:00	1600	300	30
7	7844	TURNER	SALESMAN	7698	1981-08-21 오전 12:00:00	1500	0	30
8	7900	JAMES	CLERK	7698	1981-12-11 오전 12:00:00	950	(null)	30
9	7521	WARD	SALESMAN	7698	1981-02-23 오전 12:00:00	1250	500	30
10	7902	FORD	ANALYST	7566	1981-12-11 오전 12:00:00	3000	(null)	20
11	7369	SMITH	CLERK	7902	1980-12-09 오전 12:00:00	800	(null)	20

## - To\_YMINTERVAL 함수

: add\_months 함수의 진화된 버전의 함수 (년, 월 제어)

Ex) 오늘 날짜에서 1년 2개월 후의 날짜가 어떻게 되는가?

```
SELECT SYSDATE + to_yinterval('01-02')    <-- 1년 2개월
FROM dual;
```

	SYSDATE+TO_YMINTERVAL('01-02')
1	2019-06-18 오전 11:15:12

## - TO\_DSINTERVAL 함수

: 시,분,초 까지 제어 할 수 있다.

Ex) 오늘 날짜에서 100일 10시간 후의 날짜와 시간이 어떻게 되는가?

```
SELECT SYSDATE + to_dsinterval('100 10:00:00')    <-- 100일 10시간
FROM dual;
```

	SYSDATE+TO_DSINTERVAL('100 10:00:00')
--	---------------------------------------

```
SELECT SYSDATE + to_dsinterval('100 10:00:00') <-- 100일 10시간
```

```
FROM dual;
```

	<input type="checkbox"/> SYSDATE+TO_DSINTERVAL('10010:00:00')
1	2018-07-27 오후 9:18:38

Sql gate 에서는 시분초가 나온다

문제431. 위의 SQL의 결과가 SQLPLUS 틀에서 시분초가 출력되게 하려면 어떻게 해야 되는가?

```
Select to_char(sysdate + to_dsinterval('100 10:00:00'),  
              'RRRR/MM/DD HH24:MI:SS')
```

```
From dual;
```

TO_CHAR(SYSDATE+TO_DSINTERVAL('10010:00:00'))
-----
2018/07/27 21:24:14

## 33. 정규식 함수

2018년 4월 18일 수요일 오전 11:37

**정규식 함수** : 정규식을 이용하면 문자열 데이터에서 복잡한 패턴의 검색을 할 수 있다.

### 정규식 함수 5가지

1. Regexp\_substr
2. Regexp\_count
3. Regexp\_like
4. Regexp\_replace
5. Regexp\_instr

#### 33-1. regexp\_substr

1) 테이블 생성 후, 겨울 왕국 대본을 오라클 데이터 베이스에 입력한다.

```
CREATE TABLE winter_kingdom  
(win_text VARCHAR2(100));
```

```
SELECT * FROM winter_kingdom;
```

WIN_TEXT
1 "Final Shooting Draft
2 by
3 Jennifer Lee
4 9/23/13
5 Property of the Walt Disney Animation Studios
6 OPEN ON: ICE.
7 We're underwater looking up at it. A saw cuts through,
8 heading right for us.
9 EXT. SNOW-CAPPED MOUNTAINS ? DUSK
10 ICE HARVESTERS, dressed in traditional Sami clothing, score a
11 frozen lake. They SING.

**문제432.** 겨울왕국 대본을 단어별로 잘라서 검색하시오.

```
SELECT regexp_substr(LOWER(win_text), '[^ ]+', 1, 1)  
FROM winter_kingdom;
```

REGEXP_SUBSTR(LOWER(WIN_TEXT), '[^ ]+', 1, 1)
1 "final
2 by
3 jennifer
4 9/23/13

1, 2를 하면 두번째 단어인 shooting이 나온다

query에 사용된 표현식 '[^ ]+'에 대한 설명입니다.

- [ 는 표현식을 시작합니다.
- ^는 NOT을 나타냅니다.
- 공백을 나타냅니다.

2	by
3	jennifer
4	9/23/13
5	property
6	open
7	we're
8	heading
9	ext.
10	ice
11	frozen
12	^the
13	ice

==> 첫 번째 단어들만 나온다

- [는 표현식을 시작합니다.
- ^는 NOT을 나타냅니다.
- 공백을 나타냅니다.
- ]는 표현식을 종료합니다.
- +는 1 이상을 나타냅니다.
- 공백을 나타냅니다.

==> 공백이아닌 단어들을 잘라내라

Create table winter7

as

```

SELECT regexp_substr(LOWER(win_text), '[^ ]+', 1, 1) aa1
  FROM winter_kingdom
  Union all
SELECT regexp_substr(LOWER(win_text), '[^ ]+', 1, 2) aa1
  FROM winter_kingdom
  Union all
SELECT regexp_substr(LOWER(win_text), '[^ ]+', 1, 3) aa1
  FROM winter_kingdom
  Union all
SELECT regexp_substr(LOWER(win_text), '[^ ]+', 1, 4) aa1
  FROM winter_kingdom
  Union all
SELECT regexp_substr(LOWER(win_text), '[^ ]+', 1, 5) aa1
  FROM winter_kingdom
  Union all
SELECT regexp_substr(LOWER(win_text), '[^ ]+', 1, 6) aa1
  FROM winter_kingdom
  Union all
SELECT regexp_substr(LOWER(win_text), '[^ ]+', 1, 7) aa1
  FROM winter_kingdom
  Union all
SELECT regexp_substr(LOWER(win_text), '[^ ]+', 1, 8) aa1
  FROM winter_kingdom
  Union all
SELECT regexp_substr(LOWER(win_text), '[^ ]+', 1, 9) aa1
  FROM winter_kingdom
  Union all
SELECT regexp_substr(LOWER(win_text), '[^ ]+', 1, 10) aa1
  FROM winter_kingdom
  Union all
SELECT regexp_substr(LOWER(win_text), '[^ ]+', 1, 11) aa1
  FROM winter_kingdom
  Union all
SELECT regexp_substr(LOWER(win_text), '[^ ]+', 1, 12) aa1
  FROM winter_kingdom
  Union all
SELECT regexp_substr(LOWER(win_text), '[^ ]+', 1, 13) aa1
  FROM winter_kingdom
  Union all
SELECT regexp_substr(LOWER(win_text), '[^ ]+', 1, 14) aa1
  FROM winter_kingdom;

```

문제433. 겨울왕국 대본의 단어, 건수를 출력하는데 건수가 높은 것부터 출력하시오.

```
SELECT aa1, COUNT(*)
FROM winter7
WHERE aa1 IS NOT null
GROUP BY aa1
ORDER BY COUNT(*) desc;
```

	AA1	COUNT(*)
1	the	879
2	anna	537
3	and	444
4	to	420
5	a	387
6	kristoff	303
7	of	254
8	elsa	234
9	i	218
10	you	207
11	(cont'd)	201
12	in	196
13	her	192

문제434. 겨울왕국 대본에서 anna는 몇번 나오고 elsa라는 단어는 몇번 나오는가?

```
SELECT aa1, COUNT(*)
FROM winter7
WHERE LOWER(aa1) IN ('anna', 'elsa')
GROUP BY aa1;
```

	AA1	COUNT(*)
1	anna	537
2	elsa	234

문제435. 영화 인터스텔라에서 가장 많이 나오는 단어와 그 건수와 순위를 출력하는데, 순위가 1위부터 10위까지만 출력하시오.

```
SELECT *
FROM(
    SELECT aa1, COUNT(*), RANK() OVER(ORDER BY COUNT(*) desc) 순위
    FROM interstella7
    WHERE aa1 IS NOT NULL
    GROUP BY aa1
)
WHERE 순위 BETWEEN 1 AND 10;
```

	AA1	COUNT(*)	순위
1	the	2733	1
2	a	879	2
3	to	841	3
4	cooper	817	4
5	of	669	5
6	and	500	6
7	is	436	7
8	in	356	8
9	it	331	9
10	brand	326	10

## 33-2. Regexp\_count

## : 문장 내에서 특정 단어가 몇번 나오는지 출력하는 함수

Ex)

```
SELECT regexp_count('오라클사에서 만든 오라클 데이터 베이스에서도 정규식  
오라클 SQL함수를 제공한다', '오라클') 오라클
```

```
FROM dual;
```

<input checked="" type="checkbox"/>	오라클
1	3

'오라클'이라는 단어가 3번나옴.

문제436. 겨울왕국 대본에 elsa가 몇번 나오는지 카운트 하시오.

```
SELECT sum(regexp_count(LOWER(aa1), 'elsa'))
```

```
FROM winter7;
```

<input checked="" type="checkbox"/>	SUM(REGEXP_COUNT(LOWER(AA1),'ELSA'))
1	329

더 많이 나오는 이유는 정교하게 잡아낸다. Ex) elsa's 도 포함시켜줌

### 33-3. Regexp\_like

: 일반 like 연산자 보다 더 복잡한 패턴의 데이터를 검색 할수 있는 정규식 함수

Ex)

```
REGEXP_LIKE(source_char, pattern [, match_parameter ])
```

```
SELECT first_name, last_name  
FROM employees  
WHERE REGEXP_LIKE (first_name, ' ^Ste(v|ph) en$' );
```

문제437. 이름에 EN 또는 IN 을 포함하고 있는 직원들의 이름과 월급을 출력하시오.(정규식)

```
SELECT ename, sal  
FROM EMP  
WHERE regexp_like(ename, '(EN|IN)');
```

<input checked="" type="checkbox"/>	ENAME	SAL
1	KING	5000
2	MARTIN	1250
3	ALLEN	1600

문제438. 우리반 테이블에서 전공이 경제, 경영, 통계 학인 학생들의 이름과 전공을 출력하시오.

```
SELECT ename, major  
FROM EMP2  
WHERE regexp_like(major, '(경제|경영|통계)');
```

	ENAME	MAJOR
1	김광록	경영정보학
2	미유진	시스템경영공학부
3	윤진민	경제학
4	장은희	경영정보학
5	차호성	경제학
6	미한새	경영학
7	이근호	통계학
8	신현수	경제학
9	김대경	통계학
10	김동윤	경영학
11	은해찬	경영정보학
12	지윤철	광고경영학
13	한지윤	경영학

### 33-4. regexp\_replace

: 일반 replace 문자 함수보다 더 복잡한 패턴의 데이터를  
다른 데이터로 대체할 때 유용한 정규식 함수

Ex) 이름과 월급을 출력하는데 월급을 출력 할때 0을 \*로 출력하시오.

```
SELECT ename, REPLACE(sal, 0, '*')
FROM EMP;
```

==> 일반 **replace**

[0-3] 쓸때 () 쓰면은 문자열로 입력되서

문제439. 이름과 월급을 출력하는데 월급을 출력할때 숫자 0, 1, 2, 3을 \*로 나오게 하시오.

```
SELECT ename, regexp_replace(sal, '[0-3]', '*')
FROM EMP;
```

	ENAME	REGEXP_REPLACE(SAL, '[0-3]', '*')
1	KING	5***
2	BLAKE	*85*
3	CLARK	*45*
4	JONES	*975
5	MARTIN	**5*
6	ALLEN	*6**
7	TURNER	*5**
8	JAMES	95*
9	WARD	**5*
10	FORD	****
11	SMITH	8**
12	SCOTT	****
13	ADAMS	****
14	MILLER	****

### 33-5. regexp\_instr

: 일반 instr 문자 함수보다 더 복잡한 패턴의 철자 또는 단어의  
자릿수를 찾을때 유용한 정규식 함수

Ex)

```
REGEXP_INSTR (source_char, pattern [, position [,  
occurrence [, return_option [, match_option]]])
```

```
SELECT street_address,  
REGEXP_INSTR(street_address, ' [[:alpha:]]') AS  
First_Alpha_Position  
FROM locations;
```

	STREET_ADDRESS	FIRST_ALPHA_POSITION
1	1297 Via Cola di Rie	6
2	93091 Calle della Testa	7
3	2017 Shinjuku-ku	6
4	9450 Kamiya-cho	6

알파벳으로 시작하는 자릿수가 어디인가?