목차

2020년 11월 11일 수요일 오후 3:19

107 데이터의 품질 높이기 4(CHECK)
108 데이터의 품질 높이기 5(FOREIGN KEY)
109 WITH절 사용하기 1(WITH ~ AS)
110 WITH절 사용하기 2(SUBQUERY FACTORING)
111 SQL로 알고리즘 문제 풀기 1(구구단 2단 출력)
112 SQL로 알고리즘 문제 풀기 2(구구단 1단 ~ 9단 출력)
113 SQL로 알고리즘 문제 풀기 3(직각삼각형 출력)
114 SQL로 알고리즘 문제 풀기 4(삼각형 출력)
115 SQL로 알고리즘 문제 풀기 5(마름모 출력)
116 SQL로 알고리즘 문제 풀기 6(사각형 출력)
117 SQL로 알고리즘 문제 풀기 7(1부터 10까지 숫자의 합)
118 SQL로 알고리즘 문제 풀기 8(1부터 10까지 숫자의 곱)
119 SQL로 알고리즘 문제 풀기 9(1부터 10까지 짝수만 출력)
120 SQL로 알고리즘 문제 풀기 10(1부터 10까지 소수만 출력)
121 SQL로 알고리즘 문제 풀기 11(최대 공약수)
122 SQL로 알고리즘 문제 풀기 12(최소 공배수)
123 SQL로 알고리즘 문제 풀기 13(피타고라스의 정리)
124 SQL로 알고리즘 문제 풀기 14(몬테카를로 알고리즘)
125 SQL로 알고리즘 문제 풀기 15(오일러 상수 자연상수 구하기)

20.10.20

2020년 10월 20일 화요일 오후 3:51

오라클 : database S/W

Database : Data 를 저장 및 관리 하는 저장소

Database의 종류

1. 오라클 : 중요한 데이터
2. Mssql : 주로 게임회사에서 사용
3. Mysql : 상대적으로 덜 중요한 데이터
4. Postgresql
5. Maria db

SQL : Stucture Query Language
데이터를 검색하고 조작하는 언어

SQL의 종류

1. Query문 : 데이터를 검색하는 언어
   select 문의 6가지 절
2. DML문 : Data Manipulation Language
   (조작)
   Insert : 데이터 입력언어
   Update : 데이터 수정언어
   Delete : 데이터 삭제언어
   Merge : 데이터 입력,수정,삭제를 한번에 수행
3. DDL문 : Date Definition Langauge
   crate(생성), alter(수정), drop (삭제), truncate(삭제),
   rename(이름변경)
4. DCL 문 : Data Control Langauge
   grant(데이터를 접근 할 수 있는 권한 부여),
   revoke(데이터를 접근 할 수 있는 권한 취소)
5. TCL문 : Transaction Control Language
   commit (현 상태의 데이터의 상태를 db에 영구 저장)
   rollback (지금까지의 작업을 모두 취소)
   savepoint (특정 지점까지 롤백하는 기능)

Oracle database 19c :
인공지능의 한 부분인 머신러닝 기능이 내장되어 있다.

- 컴퓨터가 스스로 데이터를 학습하여 공부하는 기능

설치시 주의사항 : 단계 4/8 - 컨테이너 데이터베이스로 생성 체크 해제
비밀번호 - oracle_4U
C 드라이브 밑에 oracle3 폴더 생성

오라클 접속 방법

1. 검색창에 cmd 검색 및 실행
2. Sqlplus "/as sysdba" 입력

- 도스창의 글씨크기와 배경색깔 변경방법
  마우스 우 클릭 후 속성 에서 변경 가능

Sys는 오라클의 최고 권한자이다.
- show user : 유저를 확인 할 수 있다.

Emp (사원) 테이블 컬럼 소개

Empno (사원번호)
Ename (사원이름)
Job (직업)
Mgr (관리자 번호)
Hiredate (입사일)
Sal (월급)
Comm (커미션)
Deptno (부서번호)

문제1. 이름과 월급과 직업을 출력하시오.

```
Select ename, sal, job
  from emp;
```

문제2. 이름과 입사일과 부서번호를 출력하시오.

```
Select ename, hiredate, deptno
  from emp;
```

20.10.21

2020년 10월 21일 수요일 오전 9:48

문제3. emp 테이블의 컬럼이 무엇이 있는지 확인하시오.

```
Describe emp
Desc emp
```

문제4. 이름, 월급, 커미션을 출력하시오.

```
Select ename, sal, comm
  from emp;
```

SQL 작성시 주의사항

1. SQL은 대소문자를 구분하지 않는다.
   ex) select * from emp ->가능
   ex) SELECT * FROM EMP ->가능
2. SQL을 한줄로 작성하지 않고
   절은 다음 라인으로 분리해서 작성한다.
   select 절
   from 절
3. 들여쓰기를 사용하여 SQL의 가동성을 높인다.
   select ename, sal
     from emp;

문제5. 이름, 직업, 입사일, 부서번호를 출력하시오.

```
Select ename, job, hiredate, deptno
  from emp;
```

문제6. emp 테이블에 모든 컬럼을 출력하시오.

```
Select *
  from emp;
```

SQL툴의 명령어

결과 화면의 가로 사이즈 조절하는 명령어 :
Set lines 300 (원하는 숫자)

결과 화면의 세로 사이즈 조절하는 명령어 :
Set pages 400 (원하는 숫자)

이전에 한 명령어를 한번더 수행하는 명령어 :
/ 엔터

메모장 열기 (이전에 한 명령어를 수정할 때 사용) :
Edit
ed

- *(asterisk)은 모든 컬럼을 모두 선택 할 때 사용

문제7. dept 테이블의 모든 컬럼을 출력하시오.

```
Select *
  from dept;
```

Dept (부서) 테이블 컬럼 소개

Deptno (부서번호)
Dname (부서명)
Loc (부서위치)

테이블은 컬럼과 로우로 구성되어 있다.

컬럼 별칭을 사용하여 출력되는 컬럼명 변경하기

- 컬럼별칭은 컬럼명 대신에 다른 컬럼명을 지정할 때 사용하는 문법
  ex) select ename as 이름, sal as 월급
  from emp;
- as 는 생략 가능

문제8. 이름과 입사일과 부서번호를 출력하는데 컬럼명이 한글로 이름, 입사일, 부서번호로 출력되게 하시오.

```
Select ename 이름, hiredate 입사일, deptno 부서번호
  from emp;
```

문제9. 이름과 월급을 출력하는데 컬럼명이 아래와 같이 출력되게 하시오.
Employee name, Salary

```
Select ename "Employee name", sal "Salary"
  from emp;
```

- 설명 : 컬럼 별칭에 대소문자를 구분하거나 공백 문자나 특수문자를 사용하려면
  양 쪽에 더블 쿼테이션 마크를 둘러줘야 한다.

연결 연산자 사용하기(||)

- 연결 연산자는 두 컬럼 데이터를 연결해서 출력하는 연산자

ex) select ename || sal
from emp;

ex) select ename || '의 월급은' || sal
from emp;

문제10. 아래와 같이 결과를 출력하시오.

KING 의 직업은 PRESIDENT 입니다.
SCOTT 의 직업은 ANALYST 입니다.

```
Select ename || ' 의 직업은 ' || job || ' 입니다.'
  from emp;
```

문제11. 위의 결과에서 출력되는 컬럼명을 사원정보라는 한글로 출력되게 하시오.

```
Select ename || ' 의 직업은 ' || job || ' 입니다.' 사원정보
  from emp;

Select ename || ' 의 직업은 ' || job || ' 입니다.' "사원 정보"
  from emp;
```

중복된 데이터를 제거해서 출력하기(DISTINCT)

- Distinct 키워드를 컬럼명 앞에 작성하고 실행하면 중복된 데이터를 제거하고 출력 가능

Ex) select job
from emp;
↓
select distinct job
from emp;

문제12. 부서번호를 출력하는데 중복을 제거해서 출력하시오.

```
Select distinct deptno
  from emp;
```

데이터를 정렬해서 출력하기(ORDER BY)

Order by 절은 데이터를 정렬하는 절이고 select 문장에 맨 마지막에 기술한다.

Ex) select ename, sal
from emp
order by sal asc;

asc : 올림차순(생략가능), desc : 내림차순

Order by 절에 여러개의 컬럼을 작성할 수 있다.

```
Ex) select ename, deptno, sal
      from emp
      order by deptno asc, sal desc;

    select ename, deptno, sal
      from emp
      order by 2 asc, 3 desc;
```

문제13. 이름과 월급을 출력하는데 월급이 높은 사원부터 출력하시오.

```
Select ename, sal
  from emp
  order by sal desc;
```

문제14. 이름과 입사일을 출력하는데 최근에 입사한 사원부터 출력하시오.

```
Select ename, hiredate
  from emp
  order by hiredate desc;
```

점심시간 문제.
이름과 월급과 부서번호를 출력하는데 부서번호가 10번, 20번, 30번 순으로 출력되게 하고 컬렴명이 한글로 이름, 월급, 부서번호로 출력되게 하시오

```
Select ename 이름, sal 월급, deptno 부서번호
  from emp
  order by deptno;
```

문제15. 이름과 직업과 입사일을 출력하는데 직업은 abc 순으로 정렬해서 출력하고 직업을 abc 순으로 정렬한 걸 기준으로 입사일을

먼저 입사한 사원부터 출력되게 하시오.

```
Select ename, job, hiredate
  from emp
  order by 2, 3;
```

Where절 배우기 1(숫자 데이터 검색)

Where 절을 사용하면 특정 조건에 대한 데이터만 선별해서 출력 가능

Ex) select ename, sal
    from emp
    where sal = 3000;

문제16. 사원번호가 7788인 사원의 사원번호와 이름과 월급을 출력하시오.

```
Select empno, ename, sal
  from emp
  where empno = 7788;
```

문제17. 30번 부서번호에서 근무하는 사원들의 이름과 월급과 부서번호를 출력하시오.

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

문제18. 위의 결과를 다시 출력하는데 월급이 높은사원 부터 출력하시오.

```
Select ename, sal, deptno
  from emp
  where deptno = 30
  order by sal desc;
```

문제19. 부서번호가 20번인 사원들의 이름과 입사일과 부서번호를 출력하는데 최근에 입사한 사원부터 출력하시오.

```
Select ename, hiredate, deptno
  from emp
  where deptno = 20
  order by hiredate desc;
```

Where절 배우기 2(문자와 날짜 검색)

Where 절로 데이터를 검색할 때 숫자와는 다르게 문자는 양쪽에 싱글 쿼테이션 마크를 둘러줘야 한다.

Ex) select ename, sal
from emp
where ename = 'SCOTT';

- SQL은 대소문자를 구분하지 않으나 데이터는 대소문자를 구분한다.
- 싱글 쿼테이션 마크 안에 있는 데이터는 문자 또는 날짜임을 오라클에 알려 준다.

문제20. 직업이 SALESMAN인 사원들의 이름과 월급과 직업을 출력하시오.

```
Select ename 이름, sal 월급, job 직업
  from emp
  where job = 'SALESMAN';
```

문제21. 81년 11월 17일에 입사한 사원의 이름과 입사일을 출력하시오.

```
Select ename, hiredate
  from emp
  where hiredate = '81.11.17';
```

- 날짜 검색을 할때는 년도/월/일 순으로 검색을 하지만 나라마다 순서가 다를수 있다. 미국이나 영국에서의 날짜 검색은 일/월/년도 순서 이다.

산술 연산자 배우기 (*,/,+,-)

Ex) select ename, sal, sal + 3000
from emp;

문제22. 이름과 연봉(sal*12) 을 출력하는데 컬럼명을 한글로 이름, 연봉으로 출력하시오.

```
Select ename 이름, sal*12 연봉
  from emp;
```

문제23. 위의 결과를 다시 출력하는데 연봉이 36000 인 사원들의 이름과 연봉을 출력하시오.

```
Select ename 이름, sal*12 연봉
  from emp
  where sal*12 = 36000;
```

3) Select ename 이름, sal*12 연봉
1) from emp
2) where 연봉 = 36000;

- 위의 SQL이 수행되지 않는 이유는 실행 순서 때문이다.
  코딩순서 : select -> from -> where
  실행순서 : from -> where -> select

문제24. 이름과 연봉을 출력하는데 연봉이 높은 사원부터 출력하시오.

```
Select ename 이름, sal*12 연봉
  from emp
  order by 연봉 desc;
```

- 코딩순서 : select -> from -> order by
  실행순서 : from -> select -> order by

문제25. 아래의 SQL에서 더하기가 먼저 실행되게 하시오

```
Select ename, sal + 200 * 2
   from emp;
```

```
Select ename, (sal + 200) * 2
  from emp;
```

비교 연산자 배우기 1(<>, <, >=,<=, =, !=,<>, ^=)

!=, <>, ^= : 같지 않다

문제26. 커미션이 150 이상인 사원들의 이름과 커미션을 출력하시오.

```
Select ename 이름, comm 커미션
  from emp
  where comm >= 150;
```

문제27. 월급이 3000 이상인 사원들의 이름과 월급을 출력하시오.

```
Select ename 이름, sal 월급
  from emp
  where sal >= 3000;
```

문제28. 직업이 SALESMAN이 아닌 사원들의 이름과 직업을 출력하시오.

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

문제29. 월급이 2400이하인 사원들의 이름과 월급을 출력하는데 월급이 높은 사원부터 출력하시오.

```
Select ename 이름, sal 월급
  from emp
  where sal <= 2400
  order by sal desc;
```

비교 연산자 배우기 2(BETWEEN AND)

문제30. 월급이 1000에서 3000 사이인 사원들의 이름과 월급을 출력하시오.

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

Select ename, sal
  from emp
  where sal >= 1000
  and sal <= 3000;
```

문제31. 월급이 1000에서 3000 사이가 아닌 사원들의 이름과 월급을 출력하시오.

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

문제32. 1981년도에 입사한 사원들의 이름과 입사일을 출력하시오.

```
Select ename, hiredate
  from emp
  where hiredate between '81.01.01' and '81.12.31';
```

비교 연산자 배우기 3(LIKE)

Ex) 이름의 첫 글자가 S로 시작하는 사원들의 이름을 출력하시오.

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

- Like 는 ~일 것 같은 이라는 영어 뜻처럼 이름의 첫번째 철자가 S로 시작할 것 같은과 같은 뜻이다.
  %는 와일드 카드로 이 자리에 뭐가 와도 관계없다.
  그 철자의 개수가 몇개가 되어도 관계가 없다는 뜻이다.

문제33. 이름의 끝 글자가 T로 끝나는 사원들의 이름을 출력하시오.

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

문제34. 81년도에 입사한 사원들의 이름과 입사일을 출력하는데 between and 를 사용하지 않고 like를 사용하시오.

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

SQL developer 설치
도스창을 이용하지 않고 오라클에 접속하여 메모장 같은 화면에서 편하게 SQL을 작성 할 수 있게 하는 툴

오라클에 접속을 하기 위해서는 접속 정보를 알아야한다.

1. 데이터 베이스 서버의 아이피 주소 : HOST=127.0.0.1
2. 포트번호 : PORT=1521
3. 오라클 인스턴스 이름 (SID) : orcl

위의 정보를 알아내는 명령어
도스창에 lsnrctl status

+ 클릭 후 name, 사용자 이름, 비밀번호, 포트 입력 후 테스트
(롤은 sysdba)

도구 -> 환경설정
설정 변경

문제35. 이름의 두번째 철자가 M인 사원들의 이름을 출력하시오.

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

- Like 와 같이 쓸 수 있는 키워드 2개
  % (와일드 카드) : 이 자리에 뭐가 와도 관계 없고 그 갯수가 몇 개가 되던 상관 없다.

  _ (언더 바) : 이 자리에 뭐가 와도 관계 없지만 자릿수는 한 개여야 한다.

문제36. 이름의 세번째 철자가 A인 사원들의 이름을 출력하시오.

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

20.10.22

2020년 10월 22일 목요일 오전 10:01

복습

기본 select 문 : select 절 (컬럼명), from (테이블명), where (검색조건), order by (정렬할 컬럼명)

코딩순서 : select -> from -> where -> order by
실행순서 : from -> where -> select -> order by

연결연산자(||) : 문자열로 결과를 출력하는 일을 수행

Ex) 김민수님의 통장 총 잔고는 39,000원 입니다.

컬럼 별칭 : 결과로 출력 될 컬럼의 이름을 변경하고 싶을 때 사용

Ex) ename -> 이름 으로 출력

비교 연산자 : >, <, >=, <=, !=

기타 비교 연산자 :

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

문제37. 이름의 첫 번째 철자가 A로 시작하는 사원들의 이름과 월급을 출력하시오.

```
Select ename, sal
  from emp
  where ename like 'A%';
```

- Like 대신 = 를 사용 할 경우 %가 와일드 카드로 인식되지 않기 때문에 Like를 사용한다.

- Like 연산자의 키워드 2가지
    - % : 와일드 카드
    - _ : 언더 바

문제38. 이름이 scott 인 사원의 이름과 월급과 직업을 출력하시오.

```
Select ename, sal, job
  from emp
  where ename = 'SCOTT';
```

문제39. 이름의 끝에서 두 번째 철자가 T 인 사원들의 이름과 월급을 출력하시오.

```
Select ename, sal
  from emp
  where ename like '%T_';
```

비교 연산자 배우기 4(IS NULL)

Null 값을 조회할 때 사용하는 연산자
Null - 데이터가 없는 상태 또는 알 수 없는 값 (unknown)

Ex) 이름과 월급과 커미션, 직업 을 출력하시오

```
Select ename, sal, comm, job
  from emp;
```

문제40. 커미션이 null인 사원들의 이름과 커미션을 출력하시오.

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

- Comm = null 로는 조회할 수 없다. 왜냐하면 null은 알 수 없는 값이므로 비교연산자인 = 로는 조회할 수 없다. 기타 비교 연산자인 is null을 사용하여야 한다.

문제41. 커미션이 null 이 아닌 사원들의 이름과 커미션을 출력하시오.

```
Select ename, comm
  from emp
```

```
where comm is not null;
```

문제42. mgr(관리자의 사원번호)이 null 사원의 이름과 직업을 출력하시오.

```
Select ename, job
  from emp
  where mgr is null;
```

- 사장은 관리자 번호가 없다.

문제43. 사원번호, 이름, 관리자 번호 를 출력하시오.

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

비교 연산자 배우기 5(IN)

Where 절의 검색조건에서 여러개의 행을 배교할 때는 in 을 사용해야 한다.

Ex) 사원번호가 7788, 7902인 사원들의 사원번호와 이름을 조회하시오.

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

```
Select empno, ename
  from emp
  where empno = 7788
  or empno = 7902;
```

- = 연산자는 하나의 값만 비교할 수 있다. 여러개의 값을 배교할 때는 in을 사용하여야 한다.

문제44. 직업이 SALESMAN, ANALYST인 사원들의 이름과 직업을 출력하시오.

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

문제45. 직업이 SALESMAN, ANALYST 가 아닌 사원들의 이름과 직업을 출력하시오.

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

- 우리반 데이터를 생성
  Insert into emp12
  Values('김정민', 28, '남', '메카트로닉스과', 'lg', 'rlawjdals113@naver.com',
  '경기도 시흥시 은행동')

문제46. 직업이 SALESMAN이 아닌 사원들의 이름과 월급과 직업을 출력하시오.

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

문제47. 위에 결과를 다시 출력하는데 월급이 높은 사원부터 출력하시오.

```
Select ename, sal, job
  from emp
  where job != 'SALESMAN'
  order by sal desc;
```

Emp12 테이블을 만든 scott 유저 창에서 emp 와 dept 테이블 생성하기

문제48. emp12 테이블에서 이름과 나이를 출력하는데 나이가 높은 학생부터 출력하시오.

```
Select ename, age
  from emp12
  order by age desc;
```

문제49. 이름과 나이와 주소를 출력하는데 30살 이상 학생들만 출력하시오.

```
Select ename, age, address
  from emp12
  where age >=30;
```

문제50. 성씨가 김씨인 학생들의 이름과 통신사를 출력하

시오.

```
Select ename, telecom
  from emp12
  where ename like '김%';
```

문제51. 전공에 통계를 포함하고 있는 학생들의 이름과 전공을 출력하시오.

```
Select ename, major
  from emp12
  where major like '%통계%';
```

- Like 연산자를 사용할 때 특정 단어를 포함하고 있는 데이터를 검색하려면 '%단어%' 라고 하면 된다.

문제52. 우리반에 gmail을 사용하는 학생들의 이름과 메일을 출력하시오.

```
Select ename, email
  from emp12
  where email like '%gmail%';
```

문제53. 나이가 27에서 34 사이인 학생들의 이름과 나이를 출력하시오.

```
Select ename, age
  from emp12
  where age between 27 and 34;
```

문제54. 나이가 27에서 34 사이가 아닌 학생들의 이름과 나이를 출력하시오.

```
Select ename, age
  from emp12
  where age not between 27 and 34;
```

문제55. 주소가 경기도인 학생들의 이름과 나이와 주소를 출력하시오.

```
Select ename, age, address
  from emp12
  where address like '%경기도%';
```

문제56. 통신사가 sk, lg인 학생들의 이름과 통신사를 출력하시오.

```
Select ename, telecom
  from emp12
  where telecom in ('sk','lg');
```

문제57. 서울에서 사는 학생들의 이름과 나이와 전공을 출력하는데 나이가 높은 학생부터 출력하시오.

```
Select ename, age, major
  from emp12
  where address like '%서울%'
  order by age desc;
```

문제58. 이메일이 gmail이 아닌 학생들의 이름과 이메일을 출력하시오.

```
Select ename, email
  from emp12
  where email not like '%gmail%';
```

문제59. 아래와 같이 결과가 출력되게 하시오.
김주원 학생의 나이는 44세 입니다.
권세원 학생의 나이는 36세 입니다.

```
Select ename || ' 학생의 나이는 ' || age || '세 입니다.'
  from emp12
  order by age desc;
```

논리 연산자 배우기(AND, OR, NOT)

오라클 연산자의 종류

1. 산술 연산자 : *, /, +, -
2. 비교 연산자 : <, >, <=, >=, !=
   기타 비교 연산자 : between and
   like
   is null

in

3. 논리 연산자 : and, or, not

Ex) 직업이 SALESMAN이고 월급이 1200 이상인 사원들의 이름과 월급과 직업을 출력하시오.

```
Select ename, sal, job
  from emp
  where job = 'SALESMAN' and sal >=1200;
```

True and true - true
False and true - false

```
Select ename, sal, job
  from emp
  where job = 'SALESMAN' or sal >=1200;
```

False or true - true

문제60. 직업이 SALESMAN 이거나 ANALYST인 사원들의 이름과 월급과 직업을 출력하시오.

```
Select ename, sal, job
  from emp
  where job = 'SALESMAN' or job = 'ANALYST';

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

문제61. 성씨가 김씨, 이씨가 아닌 학생들의 이름을 출력하시오.

```
Select ename
  from emp12
  where ename not like '김%' and ename not like '이%';
```

문제62. 이메일이 gmail과 naver가 아닌 학생들의 이름과 이메일을 출력하시오.

```
Select ename, email
  from emp12
  where email not like '%gmail%' and email not like '%naver%';
```

대소문자 변환 함수 배우기(UPPER, LOWER, INITCAP)

함수(function) : 어떤 특정 기능을 구현해 놓은 코드의 집합

입력값 -> 함수 -> 출력값

함수를 사용하는 이유 :
함수를 이용함면 좀 더 복잡한 데이터 검색을 할 수 있다.

Ex) 영화 겨울왕국에는 elsa가 많이 나올까 anna가 많이 나올까 ?
우리반 학생들이 제일 많이 쓰는 통신사는 어디인가 ?

함수의 종류

1. 단일행 함수
    a. 문자 : upper, lower, initcap
    b. 숫자
    c. 날짜
    d. 변환
    e. 일반
2. 복수행 함수
    a. Max
    b. Min
    c. Avg
    d. Sum
    e. Count
    f. Var
    g. stddev

Upper (대문자로 출력하는 함수), Lower (소문자로 출력하는 함수), Initcap (첫번째 철자는 대문자이고 나머지는 소문자로 출력하는 함수)

Ex)
```
Select upper (ename), lower (ename), initcap (ename)
  from emp;
```

문제63. emp12 테이블에서 통신사가 sk와 관련된 통신사이면 그 항색의 이름과 통신사를 출력하시오.

정확하게 데이터가 출력이 되어지게끔 sql을 작성하세요.

```
Select ename, telecom
  from emp12
  where upper (telecom) like ('SK%');
```

- Telecom 데이터를 전부 대문자로 변경

```
Select ename, telecom
  from emp12
  where lower (telecom) like ('sk%');
```

- Telecom 데이터를 전부 소문자로 변경

문자에서 특정 철자 추출하기(SUBSTR)

Ex)
```
Select ename, substr(ename,1,1)
  from emp12;
```

Ex) 성씨가 이씨인 학생들의 이름을 출력하시오 (in 과 substr 사용)

```
Select ename
  from emp12
  where substr(ename,1,1) in '이';
```

문제64. 성씨가 김, 이, 유씨인 학생들의 이름과 나이를 출력하는데 like 쓰지 말고 in과 substr을 이용하여 출력하시오.

```
Select ename, age
  from emp12
  where substr(ename, 1, 1) in ('김', '이', '유');
```

문자열의 길이를 출력하기(LENGTH)

- 철자의 길이를 출력하는 함수

Ex)
```
Select ename, length (ename)
  from emp;
```

문제65. emp12 테이블에서 이메일과 이메일의 철자의 길이를 출력하는데 이메일 철자의 길이가 가장 긴 것부터 출력하시오.

```
Select email, length (email)
```

```
from emp12
order by length (email) desc;
```

문제66. emp 테이블에서 ename을 출력하고 그 옆에 ename의 첫번째 철자를 출력하시오.

```
Select ename, substr (ename, 1, 1)
  from emp;
```

문제67. 위의 결과를 다시 출력하는데 이름의 첫번째 철자로 출력되는 부분을 소문자로 출력하시오.

```
Select ename, lower (substr (ename, 1, 1))
  from emp;
```

문제68. 아래의 결과를 initcap 쓰지 않고 upper, lowre, substr, || 를 사용하여 출력하시오.

```
Select initcap(ename)
  from emp;
```

```
Select upper(substr(ename,1,1)) || lower(substr(ename,2))
  from emp;
```

- Substr (첫번째 인자값, 두번째 인자값, 세번째 인자값)

Substr 작성시 세번째 인자값에 아무것도 작성하지 않으면 마지막 철자까지 인식된다.

```
Ex) select substr ('smith', 1, 3)
      from dual;
```

- Dual : select 절에 있는 함수의 값을 보기위한 가상의 테이블

문제69. 사원 테이블에서 이름을 출력하고 그 옆에 이름의 끝 철자를 출력하는데 끝 철자를 소문자로 출력하시오.

```
Select ename, lower(substr(ename,-1,1))
  from emp;
```

20.10.23

2020년 10월 22일 목요일 오후 6:16

문자에서 특정 철자의 위치 출력하기(INSTR)

특정 철자의 자릿수를 출력하는 함수

Ex)
```
Select instr('smith','m')
  from dual;
```

Ex)
Emp12 테이블에서 이메일을 출력하고 그 옆에 이메일에서 @가 몇번째 자리에 있는지 출력하시오.

```
Select email, instr(email,'@')
  from emp12;
```

Ex)
Emp12 테이블에서 이메일에서 @ 앞에까지의 철자를 잘라내시오.

```
Select substr(email, 1, instr(email,'@')-1)
  from emp12;
```

문제70. 이메일을 출력하고 그 옆에 이메일의 도메인을 출력하시오.

```
Select email, substr(email, instr(email,'@')+1, instr(email,'.') - instr(email,'@')-1)
  from emp12;
```

특정 철자를 다른 철자로 변경하기(REPLACE)

Ex)
```
Select replace('smitch','m','k')
  from dual;
```

문제71. 사원 테이블에서 이름과 월급을 출력하는데 월급을 출력할 때 숫자 0을 *로 출력하시오.

```
Select ename, replace(sal, '0','*')
  from emp;
```

- 문법 : replace (컬럼명, 대체 전 문자, 대체 후 문자)

문제72. emp12 테이블에서 이름을 출력하고 그 옆에 이름에두번째 철자를 출력하시오.

```
Select ename, substr (ename,2,1)
  from emp12;
```

문제73. 아산병원의 전광판을 구현하시오.

```
Select ename, replace (ename, substr (ename,2,1), '*')
  from emp12;
```

문제74. 남궁솔미 데이터를 입력하고 남궁*미로 출력 되도록 위의 SQL을 다시 작성하시오.

```
Select ename, replace (ename, substr(ename,-2,1),'*' )
  from emp12;
```

특정 철자를 N개 만큼 채우기(LPAD, RPAD)

항상 고정된 자릿수를 보장하기 위해서 필요한 함수

문법)
Lpad (컬럼명, 전체 자릿수, 채워넣을 값)

Ex)
```
Select sal, lpad(sal, 10, '*') , rpad(sal,10, '#')
  from emp;
```

설명 : lpad(sal,10,'*') 의 뜻은 월급을 출력하는데 전체 10자리 잡고 월급을 출력하고 남은 왼쪽 자리에 * 을 채워넣겠다라는 뜻

특정 철자 잘라내기(TRIM, RTRIM, LTRIM)

공백을 잘라낼 때 많이 사용하는 함수
공백 때문에 데이터 검색이 안되는 경우가 종종 있기 때문에 trim 함수를 사용한다.

설명 :
Ltrim : 왼쪽에 있는 공백을 자름
Rtrim : 오른쪽에 있는 공백을 자름
Trim : 양쪽에 있는 공백을 자름

문제75. 경기도에 사는 학생의 이름과 주소를 출력하시오. 와일드 카드를 양쪽에 사용하지 않고 한쪽에만 사용해서 출력하시오.

```
Select ename, address
  from emp12
  where ltrim (address) like '경기도%';
```

문제76. 정보통계학과가 전공인 학생의 이름과 나이와 전공을 출력하시오.

```
Select ename, age, major
 from emp12
 where trim (major) like '정보통계학과';
```

SQL

1. 기본 select 문
    a. Select : 컬럼명
    b. From : 테이블명
    c. Where : 검색조건
    d. Order by : 정렬할 컬럼명
2. 함수
    a. 단일행 함수 : 문자, 숫자, 날짜, 변환, 일반
        i. 문자 : upper, lower, initcap, substr, instr, replace, lpad/rpad, trim/rtrim/ltrim
        ii. 숫자 : round, trunc, mod
        iii. 날짜 : months_between, add_months, next_day, last_day
    b. 복수행 함수 : Max, min, avg, sum, count

현재 내가 접속한 세션의 날짜 형식을 확인

```
Select * from nls_session_parameters;
```
Nls : nationl language support

NLS_DATE_FORMAT : RR/MM/DD (년/월/일)

반올림해서 출력하기(ROUND)

숫자함수

1. Round : 반올림하는 함수
2. Trunc : 잘라내서 버리는 함수
3. Mod : 나눈 나머지 값을 출력하는 함수

Ex)
Select round (786.567, 2)
  from dual;

- 소수점 이후 두 번째 자리를 기준으로 두고 뒤에서 반올림 한다.

Select round (786.567, 0)
  from dual;

- 소수점 이전은 바로 그 자리에서 반올림한다.

문제77. emp12 테이블의 평균 나이값을 출력하시오.

```
Select avg(age)
  from emp12;
```

문제78. 위의 결과를 반올림해서 소수점 이후는 안나오게 하시오.

```
Select round(avg(age))
  from emp12;
```

숫자를 버리고 출력하기(TRUNC)

Select trunc (786.657, 2)
  from dual;

- Trunc 는 버리는 함수인데 소수점 이전은 자정된 자리를 포함해서 버리고, 소수점 이후는 지정된 자리 이후부터 버린다.

나눈 나머지 값 출력하기(MOD)

Ex)
Select mod(10, 3)
  from dual;

Select mod (24,2), mod (25, 2)
  from dual;

문제79. emp12 테이블에서 나이가 짝수인 학생들의 이름과 나이를 출력하시오.

```
Select ename, age
  from emp12
```

```
where mod(age,2) = 0;
```

날짜 간 개월 수 출력하기(MONTHS_BETWEN)

날짜 - 숫자 = 날짜
날짜 + 숫자 = 날짜
날짜 - 날짜 = 숫자

오늘 날짜를 확인하는 방법

```
Select sysdate
  from dual;

Select sysdate -1
  from dual;

Select sysdate +1
  from dual;

Select sysdate - hiredate
  from emp;
```

문제80. 위의 결과에서 소수점 이하는 나오지 않도록 반올림 하시오.

```
Select round(sysdate - hiredate)
  from emp;
```

문제81. 이름, 입사한 날짜부터 오늘까지 총 몇주 근무하였는지 출력하시오.(소수점 이하가 나오지 않도록 반올림하시오.)

```
Select ename, round (round(sysdate - hiredate)/7)
  from emp;
```

문제82. 이름, 입사한 날짜부터 오늘까지 총 몇 달 근무했는지 출력하시오.

```
Select ename, months_between (sysdate, hiredate)
  from emp;
```

- Month_between (최신날짜, 옛날날짜)
  날짜와 날짜 사이의 개월수를 출력

문제83. 아래와 같이 결과를 출력하시오.
KING 은 467 달을 근무했습니다.
BLAKE 은 464 달을 근무했습니다.

```
Select ename || ' 은 ' || round (months_between (sysdate, hiredate)) || ' 달을 근무했습니다.'
  from emp;
```

날짜 함수

1. Months_between
2. Add_months
3. Next_day
4. Last_day

개월 수 더한 날짜 출력하기(ADD_MONTHS)

Ex)
오늘날짜에서 100달뒤에 돌아오는 날짜가 몇일 인가?

```
Select add_months (sysdate, 100)
  from dual;
```

특정 날짜 뒤에 오는 요일 날짜 출력하기(NEXT_DAY)

Ex)
오늘부터 앞으로 바로 돌아올 월요일의 날짜를 출력하시오.

```
Select next_day (sysdate, '월요일')
  from dual;
```

문제84. 오늘날짜에서 100달 뒤에 돌아오는 목요일의 날짜를 출력하시오.

```
Select next_day(add_months(sysdate, 100), '목요일')
  from dual;
```

특정 날짜가 있는 달의 마지막 날짜 출력하기(LAST_DAY)

```
Select sysdate, last_day(sysdate)
  from dual;
```

문제85. 81/11/17일에 입사한 사원의 이름과 입사일을 출력하시오.

```
Select ename, hiredate
  from emp
  where hiredate = '81/11/17';
```

미국의 오라클 환경으로 날짜 형식을 변경한다.

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

```
Select ename, hiredate
  from emp
  where hiredate = '17/11/81';
```

날짜형식

년도 : RRRR, YYYY, RR, YY

RR : 81 (1981) - 현재 연도에서 가장 가까운 연도를 선택

YY : 81 (2081) - 현재 세기로 인식

월 : MM, MON

일 : DD

시간 : HH, HH24

분 : Mi

초 : SS

요일 : day, dy, d

```
Select ename, hiredate, to_char(hiredate, 'day'),
                        To_char(hiredate,'dy'),
                        To_char(hiredate,'d')
  from emp;
```

문자형으로 데이터 유형 변환하기(TO_CHAR)

숫자형 데이터나 날짜형 데이터를 문자형으로 변환 할 때 사용하는 함수

Ex)

오늘이 무슨 요일인지 출력하고 싶다면 ?

```
Select to_char(sysdate, 'day')
  from dual;
```

문제86. 이름, 입사일, 입사한 요일을 출력하시오.

```
Select ename, hiredate, to_char (hiredate,'day')
  from emp;
```

문제87. 수요일에 입사한 사원들의 이름과 압사일을 출력하시오.

```
Select ename, hiredate
   from emp
   where to_char(hiredate,'day') = '수요일';
```

문제88. 내가 무슨요일에 태어났는지 확인 하시오.

```
Select to_char (to_date('93/10/15','RR/MM/DD'), 'day')
   from dual;
```

To_char 함수를 날짜로 변환해주어야 한다.

문제89. 이름, 입사한 요일을 출력하는데 입사한 요일이 월화수목금토일 순으로 정렬되어서 출력되게 하시오.

```
Select ename, to_char (hiredate,'dy')
   from emp
   order by to_char (hiredate-1, 'd');
```

20.10.26

2020년 10월 26일 월요일 오전 9:34

복습

1. SQL을 배우는 이유
2. 기본 SELECT문
3. 함수
    a. 단일행 함수 : 문자, 숫자, 날짜, 변환, 일반
    b. 복수행 함수 : max, min, avg, sum, count

- 문자함수 : upper, lower, initcap, substr, instr, length, pad, trim, replace
- 숫자함수 : round, trunc, mod
- 날짜함수 : months_between, add_months, next_day, last_day
- 변환함수 : to_char, to_number, to_date

문제90. 직업이 SALESMAN인 사원들의 이름과 월급과 직업을 출력하는데 월급이 높은 사원부터 출력하시오.

```
Select ename, sal, job
  from emp
  where job = 'SALESMAN'
  order by sal desc;
```

설명 : order by절은 항상 마지막에 실행 된다.

문제91. 직업이 "SALESMAN'이 아닌 사람들의 이름과 입사일과 직업을 출력하는데 최근에 입사한 사원부터 출력하시오.

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

설명 :

3.Select 컬럼명

1.From 테이블명

2.Where 검색조건

4. Order by 정렬할 컬럼명

문제92. 월급이 1000에서 3000 사이인 사원들의 이름과 월급을 출력하시오.

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

문제93. 이름을 출력하고 그 옆에 이름의 첫번째 철자만 출력하는데 소문자로 출력하시오.

```
Select ename, lower(substr(ename,1,1))
  from emp;
```

문제94. emp12 테이블에서 이름과 이메일을 출력하고 그 옆에 이메일에서 @가 몇번째 철자인지 출력하시오.

```
Select ename, email, instr(email,'@')
  from emp12;
```

문제95. 이름과 입사일, 입사한 년도를 4자리로 출력하시오.

```
Select ename, hiredate, to_char (hiredate, 'YYYY')
  from emp;

Select ename, hiredate, to_char (hiredate, 'RRRR')
  from emp;
```

설명 :
년도 : RRRR, YYYY, RR, YY
월 : MM, MON
일 : DD
시간 : HH, HH24
분 : MI
초 : SS
요일 : DAY, DY, D

To_char : 날짜 -> 문자, 숫자 -> 문자

문제96. 11월에 입사한 사원들의 이름과 입사일을 출력하시

오.

```
Select ename, hiredate
   from emp
   where to_char (hiredate,'MM') = 11;
```

문제97. 문제96번을 to_char 을 사용하지 않고 substr을 사용하여 출력하시오.

```
Select ename, hiredate
   from emp
   where substr(hiredate,4,2) = 11;
```

변환함수 : to_date 함수

날짜로 형 변환하는 함수

Ex)

```
Select ename, hiredate
   from emp
   where hiredate = '81/11/17';
```

```
Select ename, hiredate
   from emp
   where hiredate = to_date ('81/11/17', 'RR/MM/DD');
```

Hiredate - 날짜형 데이터 유형

날짜형 데이터를 검색할 때는 반드시 to_date 함수를 사용할 것을 권장합니다.

문제98. 1981년도에 입사한 사원들의 이름과 입사일을 출력하시오.

```
Select ename, hiredate
   from emp
   where to_char(hiredate,'RRRR') = '1981';
```

```
Select ename, hiredate
   from emp
   where hiredate between to_date('81/01/01', 'RR/MM/DD') and
   to_date ('81/12/31', 'RR/MM/DD');
```

To_number 함수

숫자로 형변환하는 함수

Ex) 월급이 3000인 사원의 이름과 월급을 출력하시오.

```
Select ename, sal
   from emp
   where sal = 3000;
```

Varchar2 : 문자형

Number : 숫자형

Date : 날짜형

```
Select ename, sal
   from emp
   where sal (숫자형) = '3000'; (문자형)
```

내부적으로 문자형을 숫자형으로 변환 한다. (오라클)

단점 : 검색속도가 느려진다.

다른 database 소프트웨어에서는 에러가 나면서 실행이 되지 않는다.

(오라클에서는 실행 가능)

문제99. 1981년도에 입사한 사원들의 이름과 입사일과 입사한 년도를 출력하는데 가장 최근에 입사한 사원부터 출력하시오.

```
Select ename, hiredate, to_char (hiredate,'RRRR')
   from emp
   where to_char (hiredate,'RRRR') = 1981
   order by hiredate desc;
```

함수 :

단일행 함수 : 문자, 숫자, 날짜, 변환, 일반

복수행 함수 : max, min, avg, sum, count

변환 함수 :

1. To_char : 문자로 형 변환하는 함수
2. To_number : 숫자로 형 변환하는 함수
3. To_date : 날짜로 형 변환하는 함수

```
Create table emp100
(ename varchar(10),
Sal varchar(10));

Insert into emp100 values ('scott', '2000');
```

```
Insert into emp100 values ('smith', '3000');
Commit;

Set autot on;

Select *
   from emp100
   where sal = 2000;
```

조회 되는지 확인

- Where 절에 검색조건을 적을 때 주의할 사항은 문자 컬럼의 데이터를 검색할 때는 문자로 검색하고 숫자 컬럼의 데이터를 검색할 때는 숫자로 검색해야 한다. 만약 문자형에 숫자형으로 검색하거나 그 반대로 검색할 경우 오라클에서 에러는 나지 않지만 검색 성능이 느려진다. 그러므로 반드시 검색조건을 작성할 때 위의 사항을 지켜야 한다.

일반함수 :
Ex)전광판의 이름 : 유*수

1. Nvl 함수
2. Decode 함수
3. Case 함수

Nvl 함수 : 값 대신에 다른 값을 출력하고 싶을 때 사용하는 함수

Ex) 이름, 월급, 커미션, 월급+커미션을 출력하시오.

```
Select ename, sal, nvl(comm,0), nvl(comm,0)+sal
   from emp;
```

설명 - nvl함수로 다른값으로 대체해서 출력하는 것이지 실제로 테이블의 데이터가 0으로 변경 되는것은 아니다.

문제100. 이름, 커미션을 출력하시오.

```
Select ename, nvl(comm,0)
   from emp;
```

문제101. 이름, 커미션을 출력하는데 커미션이 null인 사원들은 no comm이라는 글씨로 출력되게 하시오.

```
Select ename, nvl(to_char(comm),'no comm')
  from emp;
```

설명 - 숫자형을 문자형으로 변환해서 데이터 타입을 서로 동일하게 맞춰주고 출력하면 된다.

문제102. comm이 null 인 사원의 이름과 커미션을 출력하시오
(is null을 사용하지 않고 nvl함수로 출력하시오)

```
Select ename, comm
  from emp
  where nvl(comm,-1) = -1;
```

Decode 함수
If 문을 SQL로 구현할 때 사용하는 함수

Ex)
```
Select ename, sal, deptno,
      decode (deptno,10, 5600,
                  20, 4500,
                  0 ) as 보너스
      from emp;
```

설명 : 부서번호가 10 번이면 5600을 출력하고, 부서번호가 20번이면 4500을 출력하고 나머지 부서번호는 0을 출력하라.

문제103. 이름, 월급, 직업, 보너스를 출력하는데 보너스가 직업이 SALESMAN이면 4500을 출력하고 직업이 ANALYST면 2400 을 출력하고 나머지 직업은 0을 출력하시오.

```
Select ename, sal, job,
      decode (job, 'SALESMAN', 4500,
                  'ANALYST', 2400,
                  0 ) 보너스
      From emp;
```

문제104. 이름, 입사한 년도 4자리로 출력하시오.

```
Select ename, to_char (hiredate,'RRRR')
      From emp;
```

문제105. 이름, 입사한 년도, 보너스를 출력하는데 보너스가 입사한 년도가 1980 년이면 5000, 1981 이면 4000, 나머지 년도는 0으로 출력하시오.

```
Select ename, to_char (hiredate,'RRRR'),
       Decode(to_char(hiredate,'RRRR'),1980,5000,
                                       1981,4000,
                                       0 ) 보너스
       From emp;
```

문제106. 이름, 월급, 보너스를 출력하는데 보너스가 월급이 4000이상이면 500을 출력하고 월급이 2000 이상이고 4000 보다 작으면 300을 출력하고 나머지 사원들은 0을 출력하시오.

Decode는 등호(=) 비교만 가능하기 때문에 부등호 비교를 하려면 case 문을 사용하여야 한다. (case문은 등호, 부등호 비교가 모두 가능하다.)

```
Select ename, sal,
            Case when sal >= 4000 then 500
                 When sal >= 2000 then 300
            Else 0 end 보너스
       From emp;
```

문제107. 이름, 월급, 부서번호, 보너스를 출력하는데 보너스는 부서번호가 10번이면 500을 출력하고 부서번호가 20번이면 300을 출력하고 나머지 부서번호면 0을 출력하시오.

```
Select ename, sal, deptno,
            Case when deptno = 10 then 500
                 When deptno = 20 then 300
                 Else 0 end 보너스
       From emp;
```

문제108. emp12 테이블에서 이름을 출력하고 그 옆에 보너스를 출력하는데 이름의 글자가 3글자이면 보너스를 7000을 출력하고 이름의 철자가 2글자이면 보너스를 5000을 출력하고 이름의 철자가 4글자이면 보너스를 4000을 출력하

시오.

```
Select ename,
      Case when length(ename) = 3 then 7000
           When length(ename) = 2 then 5000
           Else 4000 end 보너스
      From emp12;
```

문제109. emp12테이블에서 이름 세글자로만 이름의 가운데 글자를 *로 출력하시오.

```
Select replace (ename, substr(ename,2,1), '*')
      From emp12;
```

이름이 2,3 글자는 위의 SQL로 수행

```
Select replace (ename, substr(ename,-2,1),'*')
      From emp12;
```

이름이 4글자는 위의 SQL을 수행

문제110. emp12테이블의 이름의 철자 개수와 관계없이 일괄적으로 이름이 * 이 아래와 같이 출력되게 하시오.

결과
남궁*미
허*
김*비

```
Select case when length(ename) in (2, 3)
           Then replace (ename, substr (ename,2,1),'*')
           Else replace (ename, substr (ename,-2,1),'*') end
      From emp12;
```

문제111. emp테이블에서 이름을 출력하고 입사한 요일을 출력하는데 입사한 요일이 월화수목금토일 순으로 출력하시오.

```
Select ename, to_char(hiredate,'dy')
      From emp
```

```
Order by to_char(hiredate-1,'d');
```

Ex) 최대월급을 출력하시오.

```
Select max(sal)
    From emp;
```

문제112. 직업이 SALESMAN인 사원들 중에 최대 월급을 출력하시오.

```
Select max(sal)
    From emp
    Where job = 'SALESMAN';
```

문제113. emp12 테이블에서 최소 나이인 학생의 나이를 출력하시오.

```
Select min(age)
    From emp12;
```

문제114. 통신사가 sk인 학생들 중에서 최대 나이인 학생의 나이를 출력하시오.

```
Select max(age)
    From emp12
    Where trim(lower(telecom)) like 'sk%';
```

문제115. 30번 부서번호의 채대 월급을 출력하시오.

```
Select max(sal)
    From emp
    Where deptno = 30;
```

위의 SQL 문에는 deptno 와 max(sal) 이 동시에 나올 수 없다.

```
Select max(sal), deptno
    From emp
    Where deptno = 30
    Group by deptno;
```

Group by deptno 를 하면 여러 개 나오려는 deptno 를 grouping 해준다.

문제116. 직업, 직업별 최대월급을 출력하는데 직업이 SALESMAN만 출력하시오.

```
Select job, max(sal)
        From emp
        Where job = 'SALESMAN'
        Group by job;
```

과제 5문제

2020년 10월 28일 수요일 오후 3:35

문제1. 부서번호, 부서번호별 최대월급을 출력하는데 부서번호별 최대월급이 높은것 부터 출력하시오.

```
Select deptno, max(sal)
    From emp
    Where sal is not null
    Group by deptno
    Order by max(sal) desc;
```

문제2. 직업과 직업별 최대월급을 출력하는데, 직업이 SALESMAN은 제외하고 출력하시오.

```
Select job, max(sal)
    From emp
    Where job != 'SALESMAN'
    Group by job;
```

문제3. 입사한 년도 (4자리), 입사한 년도별 최소 월급을 출력하시오.

```
Select to_char (hiredate, 'RRRR'), min(sal)
    From emp
    Group by to_char (hiredate, 'RRRR');
```

문제4. 입사한 년도 (4자리), 입사한 년도별 최소 월급을 출력하는데 입사한 년도별 최소 월급이 높은 사원부터 출력하시오.

```
Select to_char (hiredate, 'RRRR'), min (sal)
    From emp
    Group by to_char(hiredate, 'RRRR')
    Order by min(sal) desc;
```

문제5. 통신사, 통신사별 최대 나이를 출력하시오.
(결과가 아래와 같이 출력 되어야 합니다.)

sk	36

lg	44
kt	31

```
Select decode (telecom, 'SK', 'sk',
                        'skt', 'sk',
                        telecom),
       max(age)
       From emp12
       Where telecom is not null
       Group by decode (telecom, 'SK', 'sk',
                                 'skt', 'sk',
                                 telecom);
```

20.10.29

2020년 10월 29일 목요일 오전 9:35

1. SQL을 배우는 이유
2. 기본 SELECT문
    a. Select
    b. From
    c. Where
    d. Group by
    e. Order by
- 실행 순서 : b, c, d, a, e

3. 함수
    a. 단일행 함수 : 문자, 숫자, 날짜, 변환, 일반
    b. 복수행 함수 : max, min, avg, sum, count

- 문자함수 : upper, lower, initcap, substr, instr, length, pad, trim, replace
- 숫자함수 : round, trunc, mod
- 날짜함수 : months_between, add_months, next_day, last_day
- 변환함수 : to_char, to_number, to_date

문제117. 직업이 ANALYST 인사원들의 최대 월급을 출력하시오.

```
Select max(sal)
    From emp
    Where job = 'ANALYST';
```

문제118. 직업과 직업별 최대 월급을 출력하시오.

```
Select job, max(sal)
    From emp
    Where job is not null
    Group by job;
```

문제119. 위의 결과를 다시 출력하는데 직업별 최대월급이 높은것부터 출력하시오.

```
Select job, max(sal)
    From emp
    Where job is not null
    Group by job
    Order by max(sal) desc;
```

문제120. 부서번호, 부서번호별 최대 월급을 출력하는데, 부서번호별 최대월급이 높은 것 부터 출력하시오.

```
Select deptno, max(sal)
        From emp
        Group by deptno
        Order by max(sal) desc nulls last;
```

설명 :

order by 절의 옵션

Nulls last : null을 맨 뒤로 보낸다.

Nulls first : null을 맨 앞으로 보낸다.

To_char 를 이용하여 숫자를 문자로 형변환 하기

Ex)

```
Select sal, to_char(sal,'999,999')
        From emp;
```

설명 : to_char (sal,'999,999') 의 의미는 sal 숫자를 문자로 출력하는데

to_char (숫자형 컬럼, '문자포멧')의 문법에 따라 문자 포멧에 맞는 문자로 출력하는 것

9는 자릿수를 의미하고 이 자리에 0~9 사이의 숫자중에 어떤 숫자가 와도 관계없지만

자릿수는 한자리여야 한다.

문제121. 이름과 연봉(sal*12) 을 출력하는데 연봉을 출력할 때 천단위와 백만단위가 표시되게 하시오.

```
Select ename, to_char(sal*12,'999,999,999')
        From emp;
```

문제122. 위의 결과를 다시 출력하는데 컬럼명을 한글로 연봉이라고 하고 연봉이 높은 사원부터 출력하시오.

```
Select ename, to_char (sal*12,'999,999,999') 연봉
        From emp
        Order by to_char (sal*12,'999,999,999') desc;
```

문제123. 직업, 직업별 최대월급을 출력하는데 직업별 최대월급을 출력할 때 천단위 콤마(,) 가 출력되게 하시오.

```
Select job, to_char(max(sal),'999,999')
        From emp
        Group by job;
```

문제124. 직업, 직업별 최소월급을 출력하시오.

```
Select job, min(sal)
        From emp
        Where job is not null
        Group by job;
```

문제125. 위의 결과를 다시 출력하는데 직업이 abc 순서대로 출력되게 하시오.

```
Select job, min(sal)
        From emp
        Where job is not null
        Group by job
        Order by job;
```

문제126. 위의 결과에서 직업이 SALESMAN 은 제외하고 출력하시오.

```
Select job, min(sal)
        From emp
        Where job != 'SALESMAN'
        Group by job
        Order by job;
```

문제127. emp12 테이블에서 최소나이를 출력하시오.

```
Select min(age)
        From emp12;
```

설명 : 만약 이름도 같이 출력하려면 지금까지 배운 내용으로는 할 수 없고 서브쿼리를 사용해야 한다.

문제128. 서울에서 사는 학생중에 최소 나이를 출력하시오.

```
Select min(age)
        From emp12
        Where address like '서울%';
```

설명 : 와일드 카드를 양쪽에 사용하게 되면 검색 성능이 느려진다.

와일드카드가 뒤쪽에 있는 경우는 상관이 없으나 앞에 있으면 검색성능이 느려진다.

문제129. 입사한 년도(4자리)를 출력하고 입사한 년도별 최소월급을 출력하시오.

```
Select to_char(hiredate,'RRRR'), min(sal)
	From emp
	Group by to_char(hiredate,'RRRR');
```

평균값 출력하기(AVG)

문제 130. emp12 테이블의 평균 나이를 구하시오.

```
Select avg(age)
	From emp12;
```

문제131. 사원테이블의 월급의 평균값을 출력하시오.

```
Select avg(sal)
	From emp;
```

설명 : 월급을 다 더한 후 14로 나눔

문제132. 커미션의 평균값을 출력하시오.

```
Select avg(comm)
	From emp;
```

설명 : 커미션을 다 더한 후 4로 나눔

- 그룹함수는 null값을 무시한다.

문제133. 위의 결과를 다시 출력하는데 4로 나누지 않고 14로 나누게 하시오.

```
Select avg(nvl(comm,'0'))
	From emp;
```

설명 : null값을 0으로 변경 후 평균값을 구했기 때문에 4로 나누지 않고 14로 나누게 되었다.

문제134. 위의 결과에서 소수점 이하는 나오지 않게 반올림 하시오.

```
Select round(avg(nvl(comm,'0')))
        From emp;
```

문제135. 직업, 직업별 평균 월급을 출력하시오.

```
Select job, avg(sal)
        From emp
        Where job is not null
        Group by job;
```

```
Delete from emp
Where job is null;

Commit;

Delete from emp12
Where telecom is null;

Commit;
```

문제136. emp12 테이블에서 통신사, 통신사별 평균나이를 출력하시오.

```
Select decode(telecom, 'skt', 'sk',
                              'SK','sk', telecom), avg(age)
        From emp12
        Group by decode(telecom, 'skt', 'sk',
                                      'SK','sk', telecom);


Select decode(lower(telecom), 'skt', 'sk', lower(telecom)), avg (age)
        From emp12
        Group by decode(lower(telecom), 'skt', 'sk', lower(telecom));
```

설명 : lower(telecom) 데이터가 skt면 sk 로 출력하고 나머지 통신사는 그대로 lower(telecom) 으로 출력한다.

문제137. 전공, 전공별 평균나이를 출력하는데 전공이 ㄱㄴㄷ 순으로 출력되게 하시오.

```
Select decode(trim(major),'통계학과','통계학과',trim(major)), avg(age)
        From emp12
        Group by decode(trim(major),'통계학과','통계학과',trim(major))
```

```
Order by decode(trim(major),'통계학과','통계학과',trim(major));
```

토탈값 출력하기(SUM)

토탈값을 출력하는 함수
Ex)
사원 테이블의 총 월급을 출력하시오.

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

문제138. 직업, 직업별 토탈월급을 출력하는데 직업별 토탈 월급이 높은것 부터 출력하시오.

```
Select job,sum(sal) 토탈
    From emp
    Group by job
    Order by 토탈 desc;
```

문제139. 위의 결과에서 직업이 SALESMAN은 제외하고 출력하시오.

```
Select job, sum(sal) 토탈
    From emp
    Where job != 'SALESMAN'
    Group by job
    Order by 토탈 desc;
```

문제140. 위의 결과를 다시 출력하는데 토탈 월급이 6000 이상인 것만 출력하시오.

```
Select job, sum(sal) 토탈
    From emp
    Where job != 'SALESMAN'
        And sum(sal) >= 6000
    Group by job
    Order by 토탈 desc;
```

```
Select job, sum(sal) 토탈
    From emp
    Where job != 'SALESMAN'
    Group by job
```

```
Having sum(sal) >= 6000
Order by 토탈 desc;
```

Group 함수로 조건을 줄 때는 where 절에 사용하면 안되고 having 절에 사용해야 한다.
Where 절에는 group 함수를 사용하지 않은 일반적인 검색조건을 줄 때만 사용한다.

Select 문의 6가지 절
코딩순서 : select -> from -> where -> group by -> having -> order by
실행순서 : from -> where -> group by -> having -> select -> order by

Select 검색할 컬럼명
From 검색할 테이블명
Where 검색조건
Group by 그룹핑할 컬럼
Having 그룹함수로 검색조건을 줄 때 사용
Order by 정렬할 컬럼명

문제141. 직업, 직업별 토탈월급을 출력하는데 직업이 SALESMAN 은 제외하고 출력하고 직업별 토탈월급이 6000 이상인 것만 출력하는데 직업별 토탈월급이 높은 것부터 출력하시오.

```
Select job, sum(sal)
        From emp
        Group by job
        Having sum(sal) >= 6000 and job != 'SALESMAN'
        Order by sum(sal) desc;
```

위의 SQL문이 실행 되기는 하나 검색속도가 느려진다.
일반적인 검색조건을 having 절에 사용하면 검색속도가 느려지므로 반드시 where 절에 작성해야 한다.

```
Select job, sum(sal)
        From emp
        Where job != 'SALESMAN'
        Group by job
        Having sum(sal) >= 6000
        Order by sum(sal) desc;
```

문제142. emp12 테이블에서 통신사, 통신사별 토탈 나이를 출력(skt 제외)하는데 통신사별 토탈 나이가 100 이상인 데이터만 출력하고 통신사별 토탈 나이가 높은 것부터 출력하시오.

```
Select lower(telecom), sum(age)
        From emp12
        Where telecom != 'skt'
        Group by lower(telecom)
        Having sum(age) >= 100
        Order by sum(age) desc;
```

문제143. 위의 문제를 다시 푸는데 skt 를 sk 에 포함시켜서 출력하시오.

```
Select decode(lower(telecom),'skt','sk',lower(telecom)), sum(age)
        From emp12
        Group by decode(lower(telecom),'skt','sk',lower(telecom))
        Having sum(age) >= 100
        Order by sum(age) desc;

Select substr(lower(telecom),1,2), sum(age)
        From emp12
        Group by substr(lower(telecom),1,2)
        Having sum(age) >=100
        Order by 2 desc;
```

문제144. 입사한 년도(4자리), 입사한 년도별 토탈월급을 출력하는데 토탈월급을 출력할 때 천단위 표시가 출력되게 하시오.

```
Select to_char(hiredate,'RRRR'), to_char(sum(sal),'999,999')
        From emp
        Group by to_char(hiredate,'RRRR');
```

건수 출력하기(COUNT)

건수를 출력하는 함수
Ex) 사원테이블의 인원수를 출력하시오.

```
Select count(empno)
        From emp;

Select count(*)
        From emp;

Select count(comm)
        From emp;
```

그룹함수는 null 값을 무시한다.

문제145. 직업,직업별 인원수를 출력하시오.

```
Select job, count(*)
        From emp
        Group by job;
```

설명 :

Select job, count(부모키 컬럼)
        From emp
        Group by job;

부모키 컬럼에는 * 을 사용하는 것이 가장 확실하다.

문제146. emp12 테이블에서 나이, 나이별 인원수를 출력하시오.

```
Select age, count(*)
        From emp12
        Group by age;
```

문제147. 위의 결과를 다시 출력하는데 나이별 인원수가 높은 것 부터 출력하시오.

```
Select age, count(*)
        From emp12
        Group by age
        Order by count(*) desc;
```

문제148. 위의 결과를 다시 출력하는데 나이별 인원수가 2명 이상인 것만 출력하시오.

```
Select age, count(*)
        From emp12
        Group by age
        Having count(*) >= 2
        Order by count(*) desc;
```

문제149. 통신사, 통신사별 인원수를 출력하시오.

```
Select substr(lower(telecom),1,2),count(*)
        From emp12
        Group by substr(lower(telecom),1,2);
```

- Decode로 출력 가능

문제150. 이름, 이메일의 도메인만 출력하시오.

```
Select ename, substr(email, instr(email,'@')+1, instr(email,'.',-1,1) - instr(email,'@')-1)
  from emp12;
```

문제151. 이메일 도메인, 이메일 도메인별 인원수를 출력하시오.

```
Select substr(email, instr(email,'@')+1, instr(email,'.',-1,1) - instr(email,'@')-1), count(*)
      from emp12
      Group by substr(email, instr(email,'@')+1, instr(email,'.',-1,1) - instr(email,'@')-1);
```

문제152. 주소, 주소별 인원수를 아래와 같이 출력하시오.

서울시	19
안산시	1

```
Select substr(trim(address),1,3), count(*)
      From emp12
      Group by substr(trim(address),1,3);
```

데이터 분석 함수로 순위 출력하기 1(RANK)

데이터 분석 함수 : 데이터 분석을 용이하게 하기 위해서 제공하는 함수

Rank는 순위를 출력하는 함수이다.

Ex)
이름, 월급, 월급에 대한 순위를 출력하시오.

```
Select ename, sal, rank() over (order by sal desc) as 순위
      From emp;
```

문제153. 이름, 나이, 순위를 출력하는데 나이가 높은 순서대로 출력하시오.

```
Select ename, age, rank() over(order by age desc) 순위
      From emp12;
```

문제154. 직업, 이름, 월급, 순위를 출력하는데 직업별로 각각 월급이 높은 순서대로 순위가 출력되게 하시오.

```
Select job, ename, sal, rank() over(partition by job
```

Order by sal desc) 순위
From emp;

rank() over(partition by job
Order by sal desc)
직업별로 파티션해서 월급을 높은 순서대로 순위를 출력하겠다.

설명 : partition by는 group by 와 혼동하면 안된다.
Partition by 데이터 분석함수에서 괄호 안에 쓰는 문법으로
Partition by job 이라고 하면 직업별로 각각 파티션해서 나누겠다는 뜻이다.

문제155. 부서번호, 이름, 입사일 순위를 출력하는데 순위가 부서번호별로 각각 먼저 입사한 사원 순으로 순위가 부여되게 하시오.

Select deptno, ename, hiredate, rank() over (partition by deptno
Order by hiredate) 순위
From emp;

문제156. 통신사, 이름, 나이, 순위를 출력하는데 통신사별로 각각 나이가 높은 학생순으로 순위를 부여하시오.

Select substr(lower(telecom),1,2), ename, age,
rank() over(partition by substr(lower(telecom),1,2)
Order by age desc) 순위
From emp12;

데이터 분석 함수로 순위 출력하기 2(DENSE_RANK)

Select substr(lower(telecom),1,2), ename, age,
Dense_rank() over(partition by substr(lower(telecom),1,2)
Order by age desc) 순위
From emp12;

문제157. 이름, 월급, 순위를 출력하는데 dense_rank를 사용하여 월급이 높은 순서에 대한 순위가 1,2,3,4 ... 등 전부 출력되도록 하시오.

Select ename, sal, dense_rank() over (order by sal desc) 순위
From emp;

문제158. 입사한 년도(4자리), 이름, 월급, 순위를 출력하는데 순위

가 입사한 년도별로 각가 월급이 높은 순서대로 순위를 출력하시오.

```
Select to_char(hiredate,'RRRR'), ename, sal,
Dense_rank() over(partition by to_char(hiredate,'RRRR')
                    Order by sal desc) 순위
      From emp;
```

월급이 같아서 같은 순위가 여러 개 나오면 다음 순위가 그 다음 순위가 중복된 수만큼 더해서 출력되는 것이 rank 이고 중복되었더라도 그냥 바로 그 다음 순위가 나오는게 dense_rack 이다.

문제159. 이메일 도메인, 이름, 나이, 순위를 출력하는데 순위가 이메일 도메인 별로 각각 나이가 높은 학생순으로 출력하시오.

```
Select substr(email,instr(email,'@')+1,instr(email,'.',-1,1)-instr(email,'@')-1) 도메인, ename,
age,
dense_rank() over(partition by substr(email,instr(email,'@')+1,instr(email,'.',-1,1)-
instr(email,'@')-1)
Order by age desc) 순위
      From emp12;
```

20.10.30//

2020년 10월 30일 금요일 오전 9:42

1. SQL을 배우는 이유
2. 기본 SELECT문
    a. Select : 검색할 컬럼명
    b. From : 검색할 테이블명
    c. Where : 검색 조건
    d. Group by : 그룹핑할 컬럼
    e. Having : 그룹함수로 만든 조건
    f. Order by : 정렬할 컬럼명

- 실행 순서 : b, c, d, e, a, f

3. 함수
    a. 단일행 함수 : 문자, 숫자, 날짜, 변환, 일반
        - 문자함수 : upper, lower, initcap, substr, instr, length, pad, trim, replace
        - 숫자함수 : round, trunc, mod
        - 날짜함수 : months_between, add_months, next_day, last_day
        - 변환함수 : to_char, to_number, to_date
        - 일반함수 : decode, case, nvl, nvl2

    b. 복수행 함수 (그룹함수) : max, min, avg, sum, count
4. 데이터 분석 함수

- 회사의 전산 시스템의 구조 (데이터 저장 부분에 대한 서버의 종류)
    - Oltp 서버 (OnLin Transaction Processing) : 실시간으로 봐야할 데이터들을 저장
    - Dw 서버 (DataWare house) : 과거 이력 데이터 (데이터 분석을 주로 사용)

    a. Rank

문제160. 이름, 입사일, 순위를 출력하는데 순위가 먼저 입사 사원 순으로 순위를 부여하시오.

```
Select ename, hiredate, dense_rank() over(order by hiredate) 순위
    From emp;
```

문제161. 직업, 이름, 입사일 순위를 출력하는데 순위가 직업별로 각각 먼저 입사한 사원순으로 순위를 부여하시오.

```
Select job, ename, hiredate,
        Dense_rank() over(partition by job order by hiredate) 순위
        From emp;
```

문제162. 이름, 월급, 순위를 출력하는데 순위가 월급이 높은 순서데로 순위를 부여하시오.

```
Select ename, sal, rank() over (order by sal desc)
        From emp;

Select ename, sal, dense_rank() over (order by sal desc)
        From emp;
```

두개의 차이 확인하기

문제163. 월급이 2975는 순위가 몇위 인가 ?

```
Select dense_rank(2975) within group (order by sal desc) 순위
        From emp;
```

문제164. 우리반에서 34 나이의 순위를 출력하시오.

```
Select dense_rank(34) within group (order by age desc) 순위
        From emp12;
```

문제165. 81년 11월 17일에 입사한 사원은 사원 테이블에서 몇 번째로 입사한 사원인가 ?

```
Select dense_rank(to_date('81/11/17','RR/MM/DD')) within group (order by hiredate) 순위
        From emp;
```

데이터 분석 함수로 등급 출력하기(NTILE)

등급을 출력하는 함수

Ex)

```
Select ename, sal, ntile(4) over(order by sal desc) 등급
        From emp;
```

설명 : 월급이 높은 순으로 정렬한 데이터를 4등급으로 나누겠다.

0 ~ 25% - 1 등급

25 ~ 50% - 2등급

50 ~ 75% - 3등급
75 ~ 100% - 4등급

문제166. 이름, 나이, 등급을 출력하는데 등급을 7등급으로 나눠서 출력하시오.

```
Select ename, age, ntile(7) over (order by age desc) 등급
    From emp12;
```

문제167. 직업, 이름, 월급 등급을 출력하는데 직업별 각각 등급이 3등급으로 나눠지게 하시오.

```
Select job, ename, sal, ntile(3) over (partition by job order by sal desc) 등급
    From emp;
```

데이터 분석 함수로 순위의 비율 출력하기(CUME_DIST)

순위의 비율을 출력하는 데이터 분석 함수

Ex)
```
Select ename, sal,
    Cume_dist() over (order by sal desc) 비율
    From emp;
```

문제168. 위의 결과에서 소수점 세번째 까지만 출력되게 반올림하시오.

```
Select ename, sal,
    Round(Cume_dist() over (order by sal desc) ,3) 비율
    From emp;
```

데이터 분석 함수로 데이터를 가로로 출력하기(LISTAGG)

데이터를 가로로 출력하는 함수

Ex)
```
Select deptno,
    listagg(ename,',') within group (order by ename) 이름
    From emp
    Group by deptno;
```

이름을 가로로 출력하는데 콤마 (,) 로 구분해서 출력하고 이름이 abc 순으로 정렬 되어서 출력한다.
Listagg 는 다른 분석함수와 다르게 group by 절이 필요하다.

문제169. 직업, 직업별로 해당하는 사원들의 이름을 가로로 출력하시오.

```
Select job, listagg(ename,', ') within group (order by ename) 이름
        From emp
        Group by job;
```

문제170. 아래의 결과물을 출력하시오.

ANALYST	FORD(3000), SCOTT(3000)
CLERK	ADAMS(1100), JAMES(950), MILLER(1300), SMITH(800)
MANAGER	BLAKE(2850), CLARK(2450), JONES(2975)
PRESIDENT	KING(5000)
SALESMAN	ALLEN(1600), MARTIN(1250), TURNER(1500), WARD(1250)

```
Select job, listagg(ename|| '(' || sal || ')',', ') within group (order by ename) 이름
        From emp
        Group by job;
```

문제171. 나이, 나이별로 해당하는 학생들의 이름을 가로로 출력하시오.

```
Select age, listagg(ename,', ') within group (order by ename) 이름
        From emp12
        Group by age;

Update emp12
        Set telecom = 'sk'
        Where telecom in ('skt', 'SK');

Commit;
```

문제172. 통신사를 출력하고 통신사 별로 해당하는 학생들의 이름을 출력하는데 이름 옆에 나이도 같이 출력되게 하고 나이가 높은 학생 순으로 출력되게 하시오.

```
Select telecom,
    listagg (ename || '(' || age || ')', ', ') within group (order by age desc) 이름
    From emp12
    Group by telecom;
```

데이터 분석 함수로 바로 전 행과 다음 행 출력하기(LAG, LEAD)

바로 전행을 옆에 나오게 하거나 바로 다음행을 옆에 나오게 할 때 사용하는 함수

Ex)

```
Select ename, sal, lag(sal,1) over (order by sal) 전행
    From emp;

Select ename, sal, lag(sal,1) over (order by sal) 전행,
    lead(sal,1) over (order by sal) 다음행
    From emp;
```

문제173. 이름, 입사일, 바로 전에 입사한 사원의 입사일, 바로 다음에 입사한 사원의 입사일을 출력하시오.

```
Select ename, hiredate,
    lag(hiredate,1) over (order by hiredate),
    Lead(hiredate,1) over (order by hiredate)
    From emp;
```

COLUMN을 ROW로 출력하기 1(SUM+DECODE)

회사에서 데이터를 저장하는 서버 종류

- Oltp 서버 (OnLin Transaction Processing) : 실시간으로 봐야할 데이터들을 저장
- Dw 서버 (DataWare house) : 과거 이력 데이터 (데이터 분석을 주로 사용)

Ex)

부서번호, 부서번호별 토탈월급을 출력하시오.

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

Ex)

부서번호, 부서번호가 10 번이면 월급이 출력되게 하고 아니면 0이 출력되게 하시오.

```
Select deptno, decode (deptno, 10, sal, 0)
    From emp;
```

Ex)

위의 결과에서 부서번호 컬럼은 나오지 않게 하시오.

```
Select decode (deptno, 10, sal, 0)
        From emp;
```

Ex)
위에서 출력된 14개의 데이터를 모두 sum 하시오.

```
Select sum(decode (deptno, 10, sal, 0))
        From emp;
```

Ex)
위의 컬럼명을 컬럼 별칭을 써서 숫자 10으로 변경하시오.

```
Select sum(decode (deptno, 10, sal, 0)) "10"
        From emp;
```

오라클에서 더블 쿼테이션 마크를 사용하는 경우

1. 컬럼 별칭 사용 할 때 특수문자, 공백문자, 대소문자, 구분, 숫자를 사용할 때

Ex)
위의 예제를 이용하여 20번과 30번도 그옆에 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;
```

문제174. emp12테이블에서 통신사, 통신사별 토탈 나이를 출력하시오.

```
Select telecom, sum(age)
        From emp12
        Group by telecom;
```

문제175. 위의 결과를 가로로 출력하시오.

```
Select sum(decode(telecom, 'sk', age, 0)) sk,
        sum(decode(telecom, 'lg', age, 0)) lg,
        sum(decode(telecom, 'kt', age, 0)) kt
        From emp12;
```

문제176. 아래의 SQL중에 성능이 좋은 SQL은 무엇인가 ?

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

Select sum(nvl(comm,0)) from emp;

그룹함수는 null 값을 무시한다.

Select sum(comm) from emp;
: null 값을 무시한 4건만 더한다.

Select sum(nvl(comm,0)) from emp;
: null 값을 0 으로 변경 하고 0을 연산에 포함시킨다.

설명 : Null값은 sum 연산에 포함되지 않기때문에 위의 SQL이 더 성능이 좋다.

문제177. 아래의 SQL을 튜닝하시오.

```
Select sum(decode(telecom, 'sk', age, 0)) sk,
       sum(decode(telecom, 'lg', age, 0)) lg,
       sum(decode(telecom, 'kt', age, 0)) kt
       From emp12;
```

```
Select sum(decode(telecom, 'sk', age, null)) sk,
       sum(decode(telecom, 'lg', age, null)) lg,
       sum(decode(telecom, 'kt', age, null)) kt
       From emp12;
```

문제178. 직업, 직업별 토탈월급을 출력하시오.

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

문제179. 직업, 직업별 토탈월급을 출력하시오.(가로출력)

```
Select sum(decode(job,'SALESMAN',sal)) salesman,
       sum(decode(job,'CLERK',sal)) clerk,
       sum(decode(job,'ANALYST',sal)) analyst,
       sum(decode(job,'MANAGER',sal)) manager,
       sum(decode(job,'PRESIDENT',sal)) president
       From emp;
```

COLUMN을 ROW로 출력하기 2(PIVOT)

세로를 가로로 출력하는 함수

Ex)
Select *

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

설명 :

Select * : pivot문을 사용할 때 *를 사용한다.

From (select deptno, sal from emp) : 사원테이블에서 deptno, sal만 가져온다.

Emp 테이블명만 쓸 수 없다. 결과를 보기위해서 필요한 컬럼만 선별해서 가져온다.

Emp 테이블명만 쓰면 에러가 발생한다.

Pivot (sum(sal) for deptno in (10, 20, 30)); : 부서번호가 각 10, 20, 30의 토탈 월급을 출력한다.

문제180. 통신사, 통신사별 토탈 나이를 가로로 출력하시오 (pivot 사용)

```
Select *
    From (select telecom, age from emp12)
    Pivot (sum(age) for telecom in ('sk' sk, 'lg' lg, 'kt' kt));
```

문제181. 위의 결과를 토탈나이가 아니라 평균 나이로 나오게 하시오.

```
Select *
    From (select telecom, age from emp12)
    Pivot (avg(age) for telecom in ('sk' sk, 'lg' lg, 'kt' kt));
```

데이터 분석 함수로 누적 데이터 출력하기(SUM OVER)

데이터를 누적해서 합계하는 데이터 분석 함수

Ex)
사원번호, 이름, 월급, 월급의 누적치를 출력하시오.

```
Select empno, ename, sal, sum(sal) over (order by empno) 누적치
    From emp;
```

문제182. 이름, 나이, 나이의 누적치를 출력하시오.

```
Select ename, age, sum(age) over (order by ename) 누적치
    From emp12;
```

문제183. 직업, 이름, 월급, 월급의 누적치를 출력하는데 직업별로

각각 월급의 누적치가 출력되게 하시오.

```
Select job, ename, sal,
        sum(sal) over (partition by job order by ename) 누적치
        From emp;
```

문제184. 통신사, 이름, 나이, 나이의 누적치를 출력하는데 나이의 누적치가 통신사별로 각각 누적되어서 출력되게 하시오.

```
Select telecom, ename, age,
        sum(age) over (partition by telecom order by ename) 누적치
        From emp12;
```

데이터 분석 함수로 비율 출력하기(RATIO_TO_REPORT)

Ex)
자기의 월급이 전체 월급 중에서의 비율이 어떻게 되는지 확인하는 함수

Ex)

```
Select ename, sal, round(ratio_to_report (sal) over (),2) 비율
        From emp
        Where job = 'SALESMAN';
```

데이터 분석 함수로 집계 결과 출력하기 1(ROLLUP)

집계한 결과를 맨 아래쪽에 출력하고 싶을 때 사용하는 함수

Ex)
부서번호, 부서번호별 토탈 월급을 출력하시오.

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


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

Ex)
직업과, 직업별 토탈월급을 출력하시오.

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

Ex)
위의 결과에서 직업별 토탈월급의 합계를 맨 아래 출력하시오.

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

데이터 분석 함수로 집계 결과 출력하기 2(CUBE)

집계한 결과를 맨 위쪽에 출력하고 싶을 때 사용하는 함수

Ex)

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

문제185. 통신사, 통신사별 토탈 나이를 출력하는데 맨 위에 전체 토탈나이가 출력되게 하시오.

```
Select telecom, sum(age)
        From emp12
        Group by cube(telecom);
```

문제186. 입사한 년도(4자리), 입사한 년도별 토탈 월급을 출력하는데 맨 위에 전체 토탈월급을 출력하시오.

```
Select to_char(hiredate,'RRRR'), sum(sal)
        From emp
        Group by cube(to_char(hiredate,'RRRR'));
```

문제187. 입사한년도(4자리), 입사한 년도별 토탈월급을 출력하시오. (세로, 가로 출력)

```
Select to_char(hiredate,'RRRR'), sum(sal)
        From emp
        Group by to_char(hiredate,'RRRR');

Select sum(decode(to_char(hiredate,'RRRR'),1981,sal)) "1981",
        sum(decode(to_char(hiredate,'RRRR'),1983,sal)) "1983",
        sum(decode(to_char(hiredate,'RRRR'),1980,sal)) "1980",
        sum(decode(to_char(hiredate,'RRRR'),1982,sal)) "1982"
        From emp;
```

20.11.02

2020년 11월 2일 월요일 오전 9:39

1. SQL을 배우는 이유
2. 기본 SELECT문
    a. Select : 검색할 컬럼명
    b. From : 검색할 테이블명
    c. Where : 검색 조건
    d. Group by : 그룹핑할 컬럼
    e. Havin : 그룹함수로 만든 조건
    f. Order by : 정렬할 컬럼명

- 실행 순서 : b, c, d, e, a, f

3. 함수
    a. 단일행 함수 : 문자, 숫자, 날짜, 변환, 일반
        - 문자함수 : upper, lower, initcap, substr, instr, length, pad, trim, replace
        - 숫자함수 : round, trunc, mod
        - 날짜함수 : months_between, add_months, next_day, last_day
        - 변환함수 : to_char, to_number, to_date
        - 일반함수 : decode, case, nvl, nvl2

    b. 복수행 함수 (그룹함수) : max, min, avg, sum, count
        - 그룹함수의 특징
            - null 값을 무시한다.
            - where 절의 조건이 거짓이어도 결과를 리턴한다.

            Ex) select sum(sal)
                From emp
                Where 1 = 2; 결과값 : (null)

4. 데이터 분석 함수
    a. Rank : 순위 출력
    b. Dense_rank
    c. Ntile : 등급출력
    d. Cume_dist : 비율출력
    e. Listagg : 데이터를 가로로 출력
    f. Sum (컬럼명) over (문법) : 누적데이터 출력
    g. Ratio_to_report : 비율출력
    h. Rollup : 집계 결과를 맨 아래에 출력
    i. Cube : 집계 결과를 맨 위에 출력

데이터 분석함수를 이용하면 우리가 현업에서 필수로 검색해야 하는 데이터를 긴 SQL로 작성하지 않고 간단한 함수를 이용해서 볼 수가 있다.

오라클 버전의 히스토리

8	8i	9i	10g	11g	12c	18c	19c

I : internet
G : grid (성능이 보통인 여러개의 컴퓨터를 여러대 붙여서 마치 성능이 아주 좋은 큰 서버처럼 운영하는 기술)
C : cloud

문제188. 직업, 직업별 토탈 월급을 출력하시오.

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

문제189. 그러면 위의 결과를 다시 출력하는데 맨 아래에 전체 토탈 월급이 출력되게 하시오.

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

문제190. 아래의 job 맨 아래에 null로 나온 부분에 '토탈값' 이라고 한글로 null 대신에 출력하시오.

```
ANALYST    6000
CLERK       4150
MANAGER  8275
PRESIDENT 5000
SALESMAN 5600
                29025
```

```
Select nvl(job,'토탈값'), sum(sal)
        From emp
        Group by rollup(job);
```

문제191. 위의 결과를 다시 출력하는데 컬럼명이 아래와 같이 출력되게 하시오.

JOB SUM(SAL)

ANALYST	6000
CLERK	4150
MANAGER	8275
PRESIDENT	5000
SALESMAN	5600
토탈값	29025

```
Select nvl(job,'토탈값') job, sum(sal)
    From emp
    Group by rollup(job);
```

문제192. 위의 결과를 다시 출력하는데 아래와 같이 토탈월급 부분에 천단위를 부여하시오.

```
Select nvl(job,'토탈값') job, to_char(sum(sal), '999,999')
    From emp
    Group by rollup(job);
```

데이터 분석 함수로 집계 결과 출력하기 3(GROUPING SETS)

집계결과를 출력하는 데이터 분석 함수

Ex)

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

전체 토탈값 출력 O

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

전체 토탈값 출력 X

문제193. 직업, 직업별 토탈월급을 출력하는데 맨 아래에 전체 토탈월급이 출력되게 하시오. (grouping sets 을 사용하시오)

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

문제194. 부서번호와 직업을 출력하고 그 옆에 부서번호별 직업별 토탈월급을 출력하시오.

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

Select 에 그룹함수와 함께 나열한 컬럼들은 반드시 group by 절에 명시해줘야 에러가 나지 않고 출력될 수 있다.

문제195. 부서번호와 직업을 출력하고 그 옆에 부서번호별 직업별 토탈월급을 출력하고 동시에 부서번호별 토탈월급도 중간중간 출력되게 하시오.

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

문제196. 위의 결과 맨 아래에 전체 토탈월급이 출력되게 하시오.

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

문제197. 사원번호, 사원이름, 월급을 출력하는데 맨 아래에 전체 토탈월급을 출력하시오.

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

사원번호는 중복되지 않기 때문에 grouping sets 함수의 ()안에 grouping 결과로 넣어도 그냥 자기 월급이 출력된다.

문제198. emp12 테이블에서 통신사, 이름, 나이를 출력하는데 중간중간 통신사별 토탈 나이가 출력되게 하시오.

```
Select telecom, ename, sum(age)
    From emp12
    Group by grouping sets ((telecom, ename),(telecom),());
```

문제199. 입사한 년도(4자리), 입사한 년도별 토탈월급을 출력하는데 전체 토탈월급이 맨 아래에 출력되게 하시오.

1. Rollup
```
Select to_char(hiredate,'RRRR'), sum(sal)
        From emp
        Group by rollup(to_char(hiredate,'RRRR'));
```

2. Grouping sets
```
Select to_char(hiredate,'RRRR'), sum(sal)
        From emp
        Group by grouping sets (to_char(hiredate,'RRRR'),());
```

문제200. 위의 결과를 다시 출력하는데 천단위를 부여해서 출력하시오.

```
Select to_char(hiredate,'RRRR'), to_char(sum(sal), '999,999')
        From emp
        Group by grouping sets (to_char(hiredate,'RRRR'),());
```

문제201. 입사한 년도(4자리), 부서번호, 입사한 년도별 부서번호별 토탈월급을 출력하시오.

```
Select to_char(hiredate,'RRRR'), deptno, sum(sal)
        From emp
        Group by to_char(hiredate,'RRRR'), deptno;
```

문제202. 입사한 년도(4자리), 부서번호, 입사한 년도(4자리) 별 부서번호별 토탈월급을 출력하는데 중간중간 입사한 년도(4자리) 별 토탈월급이 출력되게 하고 맨 아래에 전체 토탈월급이 출력되게 하시오.

```
Select to_char(hiredate,'RRRR'), deptno, sum(sal)
        From emp
        Group by grouping sets
            ((to_char(hiredate,'RRRR'),deptno),(to_char(hiredate,'RRRR')),());
```

데이터 분석 함수로 출력 결과 넘버링 하기(ROW_NUMBER)

Ex)
```
Select row_number() over(order by empno), empno, ename
        From emp;
```

Row_number() 함수를 사용하면 출력되는행의 필요한 행을 검색할 때 유용하다.

문제203. 직업이 SALESMAN인 사원들의 이름과 직업을 출력하는데 row_number 함수를 사용해서 맨 앞에 번호가 부여되게 하시오.

```
Select row_number() over (order by ename) 번호, ename, job
    From emp
    Where job = 'SALESMAN';
```

문제204. 직업이 SALESMAN인 사원들의 이름과 월급과 직업을 출력하는데 맨 앞에 row_number 함수를 사용하여 번호를 부여하시오.(월급이 높은순서대로 추력되게 하시오.)

```
Select row_number() over(order by sal desc) 번호, ename, sal, job
    From emp
    Where job = 'SALESMAN';
```

위의 결과에서 번호 1번만 출력하는 것은 인라인뷰를 배워야 수행 가능하다.

ROW를 COLUMN으로 출력하기(UNPIVOT)

Pivot문 : 세로를 가로로 출력
Unpivot문 : 가로를 세로로 출력

Order2라는 테이블을 생성하고 데이터를 3건 입력한다.

1. 테이블 생성 스크립트

```
create table order2
( ename  varchar2(10),
  bicycle  number(10),
  camera   number(10),
  notebook  number(10) );
```

2. 데이터 입력 스크립트

```
insert  into  order2  values('SMITH', 2,3,1);
insert  into  order2  values('ALLEN',1,2,3 );
insert  into  order2  values('KING',3,2,2 );

commit;
```

Ex)

```
Select *
    From order2
```

```
Unpivot (수량 for  물품 in (BICYCLE,CAMERA,NOTEBOOK));
```

가로로 되어 있던 컬럼을 세로로 변경

위의 SQL 에서 수량과 물품은 SQL 작성자가 임의로 명명하면 그대로 컬럼명으로 출력이 된다. 그리고 BICYCLE,CAMERA,NOTEBOOK 은 양쪽에 싱글 쿼테이션 마크를 둘러주지 않아도 된다.

출력되는 행 제한하기 1(ROWNUM)

Rownum 을 이용하면 출력되는 행의 갯수를 제한할 수 있다.

Ex)
```
Select rownum, empno, ename, sal
    From emp;
```

Ro_number 함수와는 다르게 인라인뷰를 사용하지 않고도 필요한 행을 검색할 수 있다.

Ex)
```
Select rownum, empno,ename, sal
    From emp
    Where rownum <= 3;

Select *
    From emp
    Where rownum <= 10;
```

대용량 테이블의 데이터가 있는 테이블의 내용을 보고 싶다면 rownum을 사용하여 몇건의 데이터만 보는게 좋다.

문제205. 직업이 SALESMAN인 사원들의 데이터중 2개만 출력하시오. (사원이름, 직업과 월급을 출력)

```
Select ename, job, sal
    From emp
    Where job = 'SALESMAN'
    And rownum <= 2;
```

문제206. 위의 결과에서 한건만 출력되게 하시오.

```
Select ename, job, sal
    From emp
    Where job = 'SALESMAN'
    And rownum = 1;
```

```
Select ename, job, sal
        From emp
        Where job = 'SALESMAN'
        And rownum = 2;
```

Rownum 은 부등호 비교와 같이 사용해야 하기 때문에 위의 SQL문은 결과물이 출력되지 않는다. (1은 제외)

문제207. emp12 테이블의 데이터를 가지고 오는데 위의 3건만 출력하시오.

```
Select *
        From emp12
        Where rownum <= 3;
```

출력되는 행 제한하기 2(Simple TOP-n Queries)

Order by 절까지 사용하여 검색하는 select 문의 출력되는 행의 일부를 가져올 때 사용하는 문법

Ex)
```
Select rownum, empno, ename, sal
        From emp
        Order by sal desc;
```

Order by가 수행되기 전에 rownum이 부여되어서 번호가 뒤죽박죽 석여서 출력된다.

Ex)
사원테이블에서 월급이 높은 사원4명만 출력하시오.
(이름과 월급을 출력하시오.)

```
Select ename, sal
        From emp
        Where rownum <= 4
        Order by sal desc;
```

위의 SQL문으로 검색하면 실행순서로 인해 결과값이 제대로 출력되지 않는다.
Emp 테이블에서 맨위의 4명만 가져와서 그 4명만으로 월급이 높은 순서대로 정렬한다.

위의 예제 문제를 제대로 출력하기 위해서는 아래와 같이 top n queries를 사용하여야 한다.

```
Select ename, sal
        From emp
        Order by sal desc
```

```
Fetch first 4 rows only;
```

문제208. emp12테이블에서 나이가 가장 많은 학생들의 이름과 나이를 출력하는데 5명만 출력하시오.

```
Select ename, age
    From emp12
    Order by age desc
    Fetch first 5 rows only;
```

문제209. 직업, 직업별 토탈월급을 출력하는데 직업별 토탈월급이 높은 것부터 출력하고 위의 2개의 행만 출력하시오.

```
Select job, sum(sal)
    From emp
    Group by job
    Order by sum(sal) desc
    Fetch first 2 rows only;
```

여러 테이블의 데이터를 조인해서 출력하기 1(EQUI JOIN)

조인을 이용하면 두개 이상의 테이블의 컬럼들 하나의 결과로 모아서 출력 할 수 있다.

Ex)

```
Select *
    From dept;
```

Deptno : 부서번호

Dname : 부서명

Loc : 부서위치

Ex)

이름, 부서위치를 출력하시오.

```
Select ename, loc
    From emp, dept;
```

14건 * 4건 = 56건에 대해서 모두 조인한 결과이다.

위의 문제를 제대로 출력하기 위해서는 아래와 같이 SQL문을 작성하여야 한다.

```
Select ename, loc
    From emp, dept
    Where emp.deptno = dept.deptno; (조인조건)
```

Where 절에 조인 조건을 기술해야 정확한 결과를 출력할 수 있다.

문제210. 직업이 SALESMAN인 사원들의 이름과 월급과 직업과 부서위치를 출력하시오.

```
Select ename, sal, job, loc
    From emp, dept
    Where job = 'SALESMAN'
    And emp.deptno = dept.deptno;
```

조인조건 : emp 테이블의 deptno는 dept 테이블의 deptno 와 같다라고 알려주면서 이 컬럼으로 두개의 테이블을 서로 조인한다.

문제211. 월급이 2000 이상인 사원들의 이름과 월급과 부서위치를 출력하시오.

```
Select ename, sal, loc
    From emp, dept
    Where sal >= 2000
    And emp. Deptno = dept. deptno;
```

문제212. 위의 결과에서 이름, 월급, 부서위치 옆에 부서번호도 같이 출력하시오.

```
Select emp. ename, emp. sal, dept. loc, emp. deptno
    From emp, dept
    Where sal >= 2000
    And emp. Deptno = dept. deptno;
```

Emp. Deptno 라고 작성함으로써 emp 테이블에 있는 부서번호를 가져오게 한다.
조인 문에서는 컬럼명 앞에 테이블명을 작성하면 검색 속도를 높일 수 있다.

```
Select e. ename, e. sal, d. loc, e. deptno
    From emp e, dept d
    Where sal >= 2000
    And e. Deptno = d. deptno;
```

위의 SQL문처럼 emp는 e, dept는 d 와 같이 별칭을 사용하여 코딩을 하면 좀 더 간결하게 조인문장을 작성할 수 있다.

문제213. 월급이 1000 에서 3000 사이인 사원들의 이름과 월급과 부서위치를 출력하시오.

```
Select e. ename, e. sal, d. loc
        From emp e, dept d
        Where sal between 1000 and 3000
        And e. deptno = d. deptno;
```

문제214. 사원번호가 7788, 7902, 7369번인 사원의 사원번호와 이름과 월급과 부서위치를 출력하시오.

```
Select e. empno, e. ename, e. sal, d. loc
        From emp e, dept d
        Where e. empno in(7788, 7902, 7369)
        And e. deptno = d. deptno;
```

문제215. 이름의 첫번째 철자가 S로 시작하는 사원의 이름과 월급과 부서위치를 출력하시오.

```
Select e. ename, e. sal, d. loc
        From emp e, dept d
        Where e. ename like 'S%'
        And e. deptno = d. deptno;
```

문제216. DALLAS에서 근무하는 사원들의 이름과 직업과 부서위치를 출력하시오.

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

문제217. 부서위치, 부서위치별 토탈월급을 출력하시오.

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

문제218. 부서위치, 부서위치별 평균 월급을 출력하는데 소수점 이하는 안나오게 반올림을 하고 부서위치별 평균월급이 높은 것부터 출력하고 부서위치별 평균월급이 출력될때 천단위를 부여하시오.

```
Select d. loc, to_char(round(avg(e. Sal)), '999,999')
```

```
From emp e, dept d
Where e. deptno = d. deptno
Group by d. loc
Order by to_char(round(avg(e. Sal)), '999,999') desc;
```

20.11.03

2020년 11월 3일 화요일 오전 9:23

1. SQL을 배우는 이유
2. 기본 SELECT문
    a. Select : 검색할 컬럼명
    b. From : 검색할 테이블명
    c. Where : 검색 조건
    d. Group by : 그룹핑할 컬럼
    e. Havin : 그룹함수로 만든 검색조건
    f. Order by : 정렬할 컬럼명
- 실행 순서 : b, c, d, e, a, f
3. 함수
    a. 단일행 함수 : 문자, 숫자, 날짜, 변환, 일반
        - 문자함수 : upper, lower, initcap, substr, instr, length, pad, trim, replace
        - 숫자함수 : round, trunc, mod
        - 날짜함수 : months_between, add_months, next_day, last_day
        - 변환함수 : to_char, to_number, to_date
        - 일반함수 : decode, case, nvl, nvl2

    b. 복수행 함수 (그룹함수) : max, min, avg, sum, count
        - 그룹함수의 특징
            - null 값을 무시한다.
            - where 절의 조건이 거짓이어도 결과를 리턴한다.

              Ex) select sum(sal)
                  From emp
                  Where 1 = 2; 결과값 : (null)

4. 데이터 분석 함수
    a. Rank : 순위 출력
    b. Dense_rank
    c. Ntile : 등급출력
    d. Cume_dist : 비율출력
    e. Listagg : 데이터를 가로로 출력
    f. Sum (컬럼명) over (문법) : 누적데이터 출력
    g. Ratio_to_report : 비율출력
    h. Lag, lead : 전행, 다음행 출력
    i. Rollup : 집계 결과를 맨 아래에 출력

j. Cube : 집계 결과를 맨 위에 출력
k. Grouping sets : 집계결과 출력

5. 조인(join) 문장
하나의 테이블에서 얻을 수 있는 정보가 아닌 여러개의 테이블에서 얻을 수 있는 정보를 하나의 결과로 보여주기 위해서 만든 문법

Ex)
```
Select e. ename, d. loc
        From emp e, dept d
        Where e. deptno = d. deptno and d. loc= 'DALLAS';
```

Select 컬럼명
From 테이블명
Where 조인조건 and 검색조건

문제219. 월급이 2700 이상인 사원들의 이름과 월급과 부서위치를 출력하시오.

```
Select e. ename, e. sal, d. loc
        From emp e, dept d
        Where e. deptno = d. deptno
        And e. sal >= 2700;
```

테이블 별칭을 사용해서 SQL 코딩을 심플하게 작성하고 조인문장 작성시 컬럼명 앞에 테이블 별칭을 사용하도록 한다.

테이블 별칭 사용시 이점
1. 검색속도가 빨라진다.
2. SQL 유지보수가 쉬워진다.

문제220. 이름의 끝글자가 T 로 끝나는 사원들의 이름과 월급과 부서위치와 부서명을 출력하시오.

```
Select e. ename, e. sal, d. loc, d. dname
        From emp e, dept d
        Where e. deptno = d. deptno
                and e. ename like '%T';
```

문제221. 직업이 SALESMAN이고 월급이 1200 이상인 사원들의 이름과 직업과 부서위치와 월급을 출력하시오.

```
Select e. ename, e. job, d. loc, e. sal
        From emp e, dept d
        Where e. deptno = d. deptno
                and e. job = 'SALESMAN'
                and e. sal >= 1200;
```

문제222. 부서위치, 부서위치별 토탈월급을 출력하는데 DALLAS는 제외하고 출력하시오.

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

문제223. 지금 출력된 결과를 다시 출하는데 토탈월급이 높은 것 부터 출력하시오.

```
Select d. loc, sum(e. sal)
        From emp e, dept d
        Where e. deptno = d. deptno
                And d. loc != 'DALLAS'
        Group by d. loc
        Order by sum(e. sal) desc;
```

여러 테이블의 데이터를 조인해서 출력하기 2(NON EQUI JOIN)

조인하려는 두개의 테이블 사이에 공통된 컬럼이 없을때 사용하는 조인 문법

Ex)
Salgrade 테이블 생성

Grade : 급여등급
Losal : 최소월급
Hisal : 최대월급

Ex)
이름, 월급, 급여등급을 출력하시오.

```
Select e. ename, e. sal, s. grade
        From emp e, salgrade s
        Where e. sal between s. losal and s. hisal;
```

문제224. 위의 결과에서 등급이 3등급인 사원들만 출력하시오.

```
Select e. ename, e. sal, s. grade
        From emp e, salgrade s
        Where e. sal between s. losal and s. hisal
                And s. grade = 3;
```

조인 문법의 종류 2가지

1. 오라클 조인 문법
   a. Equi join : 두개의 테이블 사이에 공통된 럼이 있을 때의 조인 방법
   b. Non equi join : 두개의 테이블 사이에 공통된 컬럼이 없을 때의 조인 방법
   c. Outer join : 두개의 테이블 사이에 공통된 컬럼은 있으나 조인하려는 컬럼의 데이터가 서로 일치 하지 않을 때 사용하는 조인 방법
   d. Self join : 자기 자신의 테이블과 조인
2. 1999 ansi 조인 분법 (american national standard institute)
   a. On절을 사용한 조인
   b. Using 절을 용한 조인
   c. Natural 조인
   d. Left/Right/Full outer 조인
   e. Cross 조인

문제225. 급여등급, 급여등급별로 해당하는 사원들의 이름을 가로로 출력하시오.

```
Select s. grade, listagg (e. ename, ', ') within group (order by e. ename) 이름
        From emp e, salgrade s
        Where e. sal between s. losal and s. hisal
        Group by s. grade;
```

문제226. 위의 결과에서 월급도 옆에 같이 나오게 하시오

```
Select s. grade, listagg (e. ename || '(' || e. sal || ')', ', ')
                within group (order by e. ename) 이름
        From emp e, salgrade s
        Where e. sal between s. losal and s. hisal
        Group by s. grade;
```

여러 테이블의 데이터를 조인해서 출력하기 3(OUTER JOIN)

조인하려는 두 테이블의 공통된 컬럼인 deptno의 데이터가 서로 일치 하지 않을 때 사용하는 조인 방법

Ex)
사원 테이블에서 부서번호를 출력하는데 중복제거해서 출력하시오.

```
Select distinct deptno
    from emp;
```

부서 테이블에서 부서번호를 출력하시오

```
Select deptno
    From dept;
```

문제227. emp와 dept를 서로 조인해서 이름과 부서위치와 부서번호를 출력하시오.

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

위의 결과를 보면 사원테이블에는 40부서번호가 없기 때문에 조인이 되지 않아서 결과로 출력되지 않는다.

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

Outer join sign (+) 는 결과로 출력될 때 데이터가 모자란 쪽에 붙여준다.
Emp 테이블에서는 부서번호가 10, 20, 30 이 있으나 40번이 없으나,
Dept 테이블에서는 부서번호가 10, 20, 30, 40 이 있다.

문제228. 부서위치, 부서위치별 토탈월급을 출력하시오.

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

위의 조인된 결과를 보면 부서위치(loc) 쪽에 BOSTON이 출력되지 않는다.

문제229. 위의 결과에서 BOSTON도 출력되도록 조인 문법을 변경하시오.

```
Select d. loc, sum (e. sal)
```

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

우리반 테이블과 조인하기 위한 테이블 생성
(통신사 기본 요금 테이블)

```
Create table telecom_price
(telecom varchar2(10),
Price number(10),
Service_cnt number(10));

Insert into telecom_price values('sk', 18500, 9);
Insert into telecom_price values('kt', 17000, 9);
Insert into telecom_price values('lg', 18000, 10);

Commit;
```

문제230. emp12 테이블과 telecom_price 테이블을 조인해서 이름, 나이, 성별, 통신사, 통신사 기본요금(price)을 출력하는데 나이가 27 이상인 학생들만 출력하시오.

```
Select e. ename, e. age, e. gender, e. telecom, t. price
    From emp12 e, telecom_price t
    Where e. telecom = t. telecom
        And e. age >= 27;
```

여러 테이블의 데이터를 조인해서 출력하기 4(SELF JOIN)

자기 자신의 테이블과 조인하는 조인 문법

Ex)
사원이름과 그 사원을 관리하는 관리자의 이름을 하나의 결과로 볼 수 있다.

사원번호, 사원이름, 관리자 번호를 출력하시오.

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

Ex)
사원이름, 해당 사원의 관리자의 이름을 출력하시오.

```
Select 사원.ename, 관리자.ename
    From emp 사원, emp 관리자
    Where 사원.mgr = 관리자.empno;
```

문제231. 위의 결과를 다시 출력하는데 컬럼명을 한글로 사원, 관리자 라고 출력되게 하시오.

```
Select 사원.ename 사원, 관리자.ename 관리자
    From emp 사원, emp 관리자
    Where 사원.mgr = 관리자.empno;
```

문제232. 사원이름, 사원월급, 관리자이름, 관리자의 월급을 출력하시오.

```
Select 사원.ename 사원, 사원.sal "사원 월급", 관리자.ename 관리자, 관리자.sal "관리자 월급"
    From emp 사원, emp 관리자
    Where 사원.mgr = 관리자.empno;
```

문제233. 위의 결과를 다시 출력하는데 사원의 월급이 관리자의 월급보다 더 큰 사원들만 출력하시오.

```
Select 사원.ename 사원, 사원.sal "사원 월급", 관리자.ename 관리자, 관리자.sal "관리자 월급"
    From emp 사원, emp 관리자
    Where 사원.mgr = 관리자.empno
        And 사원.sal > 관리자.sal;
```

문제234. 관리자보다 먼저 입사한 사원들의 사원 이름과 사원 입사일, 관리자 이름과 관리자 입사일을 출력하시오.

```
Select e1. ename 사원, e1. hiredate 사원입사일, e2. ename 관리자, e2. hiredate 관리자입사일
    From emp e1, emp e2
    Where e1.mgr = e2.empno
    And e1.hiredate < e2.hiredate;
```

문제235. 관리자이름을 출력하고 그 옆에 해당 관리자에 속한 사원들의 이름을 가로로 출력하시오.

```
Select e2.ename 관리자,
    listagg (e1.ename, ', ') within group (order by e1.ename) 사원
    From emp e1, emp e2
    Where e1.mgr = e2.empno
    Group by e2. ename;
```

문제236. 직업, 이름, 월급, 순위를 출력하는데 월급이 높은 순서데로 순위를 부여하시오.

```
Select job, ename, sal,
    dense_rank() over (order by sal desc) 순위
    From emp;
```

문제237. 위의 결과를 다시 출력하는데 직업별 각각 순위가 부여되게 하시오.

```
Select job, ename, sal,
    dense_rank() over (partition by job order by sal desc) 순위
    From emp;
```

문제238. 부서위치, 이름, 월급, 순위를 출력하는데 부서위치별로 각각 월급이 높은 순서대로 순위를 부여하시오.

```
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;
```

문제239. 부서위치, 부서위치별로 속한 사원들의 이름을 가로로 출력하시오.

```
Select d.loc, listagg(e.ename,', ') within group (order by e.ename) 사원
    From emp e, dept d
    Where e. deptno = d.deptno
    Group by d.loc;
```

문제240. 부서위치, 부서위치별 토탈월급을 출력하는데 맨 아래에 전체 토탈월급이 출력되게 하시오.

```
Select d.loc, sum(e.sal)
```

```
From emp e, dept d
Where e. deptno = d.deptno
Group by rollup(d.loc);
```

문제241. 직업, 직업 최대월급, 직업별 최소월급, 직업별 평균월급을 출력하시오.

```
Select job, max(sal), min(sal), avg(sal)
    From emp
    Group by job;
```

문제242. 부서위치, 부서위치별 최대월급, 부서위치별 최소월급, 부서위치별 인원수를 출력하시오.

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

여러 테이블의 데이터를 조인해서 출력하기 5(ON절)

Ex)

1. 오라클 equi join

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

2. On 을 사용한 조인

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

문제243. 이름과 월급과 부서위치를 출력하는데 월급이 2400 이상인 사원들만 출력하시오.(on절을 사용하여 출력하시오.)

```
Select e.ename, e.sal, d.loc
    From emp e join dept d
    On(e.deptno = d.deptno)
    where e.sal >= 2400;
```

On절을 사용한 조인 문법의 조인조건은 on 절에 주게 되어있고 검색조건은 where 절에 준다.

문제244. DALLAS에서 근무하는 사원들의 이름과 월급과 부서위치

와 직업을 출력하는데 on절을 사용한 조인문법으로 수행하시오.

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

문제245. emp테이블과 salgrade 테이블을 서로 조인해서 이름, 월급, 급여등급을 출력하는데 2등급만 출력되게 하고 on 절을 사용한 조인 문법으로 수행하시오.

```
Select e.ename, e.sal, s.grade
        From emp e join salgrade s
        On (e.sal between s.losal and s.hisal)
        Where s.grade = 2;
```

여러 테이블의 데이터를 조인해서 출력하기 5(USING절)

Ex)

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

Using() 괄호 안에 연결고리가 되는 컬럼명을 테이블 별칭 없이 써준다.

문제246. 직업이 SALESMAN인 사원들의 이름, 월급, 직업, 부서위치를 출력하는데 using절을 사용한 조인문법으로 수행하시오.

```
Select e.ename, e.sal, e.job, d.loc
        From emp e join dept d
        Using (deptno)
        Where e.job = 'SALESMAN';
```

여러 테이블의 데이터를 조인해서 출력하기 6(NATURAL JOIN)

Ex)

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

Where 절 없이 간단하게 조인하는 조인 문법
오라클 내부에서 두 테이블 사이에 공통된 컬럼이 있는지 확인하고 조인한다.

여러 테이블의 데이터를 조인해서 출력하기 7(LEFT/RIGHT OUTER JOIN)

Ex)

1. 오라클 조인 문법 (아우터 조인)

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

2. 1999 ansi 조인 문법

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

```
Insert into emp(empno, ename, sal, job, deptno)
        Values(1221, 'jack', 3500, 'SALESMAN', 70);

Commit;
```

1. 오라클 조인 문법 (아우터 조인)

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

Dept 테이블에는 70번 부서가 없으므로 조인할 때 equi 조인을 하게 되면 결과가 나오지 않고 outer join을 사용해야 한다.

문제247. 위의 결과를 1999 ansi 문법으로 구현하시오.

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

여러 테이블의 데이터를 조인해서 출력하기 8(FULL OUTER JOIN)

Ex)

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

위의 결과 출력을 가능하게 해주는 join 문법이 1999 ansi의 full outer join 이다.

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

Cross 조인

Where 절 없이 조인해서 전체를 모두 조인 하는 조인 문법이다.

1. 오라클 조인 문법

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

2. 1999 ansi 문법

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

조인 문법의 종류 2가지

1. 오라클 조인 문법
   a. Equi join : 두개의 테이블 사이에 공통된 럼이 있을 때의 조인 방법
   b. Non equi join : 두개의 테이블 사이에 공통된 컬럼이 없을 때의 조인 방법
   c. Outer join : 두개의 테이블 사이에 공통된 컬럼은 있으나 조인하려는 컬럼의 데이터가 서로 일치 하지 않을 때 사용하는 조인 방법
   d. Self join : 자기 자신의 테이블과 조인
2. 1999 ansi 조인 분법 (american national standard institute)
   a. On절을 사용한 조인
   b. Using 절을 용한 조인
   c. Natural 조인
   d. Left/Right/Full outer 조인
   e. Cross 조인

문제248. emp12과 telecom_price 을 조인해서 이름이 김정민 학생의 이름과 나이와 통신사와 통신 요금을(price) 을 출력하시오.

```
Select e.ename, e.age, e.telecom, t.price
    From emp12 e, telecom_price t
    Where e.telecom = t.telecom
    And e.ename = '김정민';
```

문제249. 나이가 28 이상인 학생들의 이름과 나이와 통신사와 통신 요금(price) 을 출력하시오.

```
Select e.ename, e.age, e.telecom, t.price
    From emp12 e join telecom_price t
    On (e.telecom = t.telecom)
```

```
Where e.age >= 28;
```

문제250. 부서위치, 부서위치별 토탈월급을 가로로 출력하시오.

```
Select sum(decode(d.loc,'NEW YORK',e.sal)) "NEW YORK",
       sum(decode(d.loc,'DALLAS',e.sal)) DALLAS,
       sum(decode(d.loc,'CHICAGO',e.sal)) CHICAGO
       From emp e, dept d
       Where e.deptno = d.deptno;
```

Pivot문

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

문제250 번의 결과를 보기 위해서 필요한 raw date는 부서위치와 월급이다.
따라서 pivot문의 from 절에 부서위치와 월급 컬럼만 가져오는 쿼리문을 작성한다.

```
Select *
       From (select d.loc, e.sal
              from emp e, dept d
              Where e.deptno = d.deptno)
       Pivot (sum(sal)
       for loc in ('NEW YORK' "NEW YORK",'DALLAS' DALLAS,'CHICAGO' CHICAGO));
```

From 절 괄호 안의 조인 쿼리문의 결과를 보기 위한 raw data이다.
From 절의 조인 쿼리문에서 d.loc 라고 작성 하였지만 실제로 출력되는 컬럼명은 loc 이므로 pivot문에서 d.loc 라고 작성하지 않고 loc라고 작성해야 한다.

20.11.04

2020년 11월 4일 수요일 오전 9:43

집합 연산자로 데이터를 위아래로 연결하기 1(UNION ALL)

데이터를 연결해서 출력하는 방법 2가지

1. 조인 (join) : 데이터를 양 옆에 연결해서 출력하는 방법
2. 집합 연산자 : 데이터를 위 아래로 연결해서 출력하는 방법

집합 연산자의 종류 4가지

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

Ex)
직업, 직업별 토탈월급을 출력하시오.

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

Ex)
전체 토탈월급을 출력하시오.

```
Select '전체토탈' job, sum(sal)
        From emp;
```

Ex)
위의 SQL을 하나로 합쳐서 데이터가 위아래로 출력되게 하시오.

```
Select job, sum(sal)
        From emp
        Group by job
Union all
Select '전체토탈' job, sum(sal)
        From emp;
```

Union all로 위의 쿼리문의 결과와 아래의 쿼리문의 결과를 하나로 합쳐서 출력한다.
집합 연산자를 사용할 때 주의 사항은 다음과 같다.

1. 집합연산자 위 아래의 쿼리문의 컬럼의 갯수가 동일해야 한다.
2. 집합연산자 위 아래의 쿼리문의 컬럼의 데이터 타입이 동일해야 한다.

3. 집합연산자 위 아래의 쿼리문의 칼럼의 컬럼명이 동일해야 한다.
4. Order by절은 맨아래의 쿼리에만 작성할 수 있다.

문제251. 부서번호, 부서번호별 토탈월급을 출력하는데 맨 아래의 전체 토탈월급이 출력되게 하시오 (union all 사용하시오.)

```
Select to_char(deptno), sum(sal)
        From emp
        Group by deptno
Union all
Select '전체토탈' deptno, sum(sal)
        From emp;
```

Deptno는 숫자형 데이터이고 '전체토탈' 은 문자형 데이터이기 때문에 오류가 발생하기 때문에 Deptno를 문자형 데이터로 변경해줘야 한다.

Delete from emp where hiredate is null;

Commit;

문제252. 입사한 년도(4자리), 입사한 년도별 토탈월급을 출력하시오.

```
Select to_char(hiredate, 'RRRR'), sum(sal)
        From emp
        Group by to_char(hiredate, 'RRRR');
```

문제253. 입사한 년도(4자리), 입사한 년도별 토탈월급을 출력하는데 맨 아래에 전체 토탈월급이 출력되게 하시오.

```
Select to_char (hiredate, 'RRRR') hire_year, sum(sal)
        From emp
        Group by to_char (hiredate,'RRRR')
Union all
Select '토탈' hire_year, sum(sal)
        From emp;
```

Union all 위 아래로 쿼리문의 컬럼의 갯수가 서로 동일해야 하며 컬럼명, 데이터 유형도 동일해야 한다.

문제254. emp12 테이블 에서 통신사, 통신사별 인원수를 출력하시

오.

```
Select telecom, count(*)
        From emp12
        Group by telecom;
```

문제255. 위 결과에서 전체인원수가 맨 아래에 출력되게 하시오.

```
Select telecom, count(*)
        From emp12
        Group by telecom
Union all
Select '전체' telecom, count(*)
        From emp12;
```

문제256. 문제253번의 결과를 정렬해서 출력하시오.

```
Select to_char (hiredate, 'RRRR') hire_year, sum(sal)
        From emp
        Group by to_char (hiredate,'RRRR')
Union all
Select '토탈' hire_year, sum(sal)
        From emp
        Order by hire_year;
```

Union all 에서 order by절은 항상 맨 아래의 쿼리문에만 사용해야 한다.

집합 연산자로 데이터를 위아래로 연결하기 2(UNION)

Union은 Union all과 같은 합집합 연산자이지만 union은 order by 절을 사용하지 않아도 정렬을 암시적으로 수행하며 중복된 데이터를 하나로 출력한다.

Ex)

```
Select to_char (hiredate, 'RRRR') hire_year, sum(sal)
        From emp
        Group by to_char (hiredate,'RRRR')
Union
Select '토탈' hire_year, sum(sal)
        From emp;
```

문제257. 부서번호, 부서번호별 토탈월급을 출력하는데 맨 아래에 전체 토탈 월급이 출력되게 하고 부서번호를 10, 20, 30 번 순으로

정렬해서 출력되게 하시오.

```
Select to_char(deptno) deptno, sum(sal)
        From emp
        Group by deptno
Union
Select '토탈월급' deptno, sum(sal)
        From emp;
```

Union은 암시적으로 정렬 작업을 수행하기 때문에 굳이 정렬된 결과를 볼 필요가 없다면 union all을 사용하는게 더 검색성능이 좋다.

문제258. 부서위치, 부서위치별 토탈월급을 출력하시오.

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

문제259. 위의 결과물에서 맨 아래에 전체 토탈월급도 출력되게 하시오.

```
Select d. loc, sum(e. sal)
        From emp e, dept d
        Where e.deptno = d.deptno
        Group by d. loc
Union all
Select '토탈월급' loc, sum(sal)
        From emp;
```

문제260. 위의 결과를 다시 출력하는데 부서위치를 abc 순서대로 정렬해서 출력되게 하시오.

```
Select d. loc, sum(e. sal)
        From emp e, dept d
        Where e.deptno = d.deptno
        Group by d. loc
Union
Select '토탈월급' loc, sum(sal)
        From emp;
```

문제261. 직업, 직업별 최대월급, 직업별 최소월급, 직업별 평균월급, 직업별 인원수를 출력하시오.

```
Select job, max(sal), min(sal), avg(sal), count(*)
	From emp
	Group by job;
```

문제262. 사원 테이블에서 최대 월급, 최소월급, 평균 월급, 전체 인원수를 출력하시오.

```
Select max(sal), min(sal), avg(sal), count(*)
	From emp;
```

문제263. 문제261의 결과와 문제262의 결과를 위아래로 연결해서 출력하시오.

```
Select job, max(sal), min(sal), avg(sal), count(*)
	From emp
	Group by job
Union all
Select '전체' job, max(sal), min(sal), avg(sal), count(*)
	From emp;
```

문제264. 위의 결과에서 숫자에 전부 천단위가 부여되게 하시오.

```
Select job, to_char(max(sal),'999,999') "MAX(SAL)", to_char(min(sal),'999,999') "MIN(SAL)",
To_char(avg(sal),'999,999') "AVG(SAL)", count(*)
	From emp
	Group by job
Union all
Select '전체' job, to_char(max(sal),'999,999') "MAX(SAL)", to_char(min(sal),'999,999')
"MIN(SAL)", To_char(avg(sal),'999,999') "AVG(SAL)", count(*)
	From emp;
```

집합 연산자로 데이터의 교집합을 출력하기(INTERSECT)

집합 연산자 :

1. 합집합 연산자 : union all, union
2. 교집합 연산자 : intersect
3. 차집합 연산자 : minus

Ex)
Emp12 테이블을 백업 하시오.

```
Create table emp12_backup
As
```

```
        Select *
                From emp12;

Select * from emp12_backup;
```

Emp12 테이블 백업2

```
Create table emp12_backup2
As
        Select *
                From emp12
                Where ename like '김%';

Select * from emp12_backup2;

Ex)
Select ename, age, telecom
        From emp12
Union all
Select ename, age, telecom
        From emp12_backup2;



Select ename, age, telecom
        From emp12
Union
Select ename, age, telecom
        From emp12_backup2;
```

위 2개의 SQL문 비교

```
Select ename, age, telecom
        From emp12
intersect
Select ename, age, telecom
        From emp12_backup2;
```

집합 연산자로 데이터의 차이를 출력하기(MINUS)

차집합을 구하는 집합 연산자

```
Select ename, age, telecom
        From emp12
Minus
Select ename, age, telecom
        From emp12_backup2;
```

Backup2 테이블에 있는 김씨의 성을 가진 사람들을 제외하고 전부 출력된다.

서브 쿼리 사용하기 1(단일행 서브쿼리)

Ex)
사원 테이블에서 최대월급을 받는 사원의 이름과 월급을 출력하시오.

```
Select ename, max(sal)
	From emp;
```

위의 SQL을 수행하려면 서브쿼리문을 사용해야 한다.

Ex)
JONES의 월급을 출력하시오.

```
Select sal
	From emp
	Where ename = 'JONES';
```

Ex)
JONES 의 월급보다 더 많은 월급을 받는 사원들의 이름과 월급을 출력하시오.

```
Select ename, sal
	From emp
	Where sal > 2975;
```

```
Select ename, sal	-> main query
	From emp
	Where sal > (select sal
			From emp
			Where ename = 'JONES');	-> sub query
```

쿼리를 두번 각각 실행하지 않고 위와 같이 한번에 수행

문제265. SCOTT과 같은 월급을 받는 사원들의 이름과 월급을 출력하시오. (SCOTT은 제외하고 출력하시오.)

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

문제266. SMITH와 직업이 같은 사원들의 이름과 직업을 출력하는

데 SMITH는 제외하고 출력하시오.

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

문제267. ALLEN보다 늦게 입사한 사원들의 이름과 입사일을 출력하시오.

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

문제268. 직업이 SALESMAN인 사원의 최대월급보다 더 많은 월급을 받는 사원들의 이름과 월급을 출력하시오.

```
Select ename, sal
	From emp
	Where sal > (select max(sal)
			From emp
			Where job = 'SALESMAN');
```

문제269. 최대월급을 받는 사원의 이름과 월급을 출력하시오.

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

문제270. 전공에 통계가 포함되어 있는 학생들 중에서의 최대 나이인 학생의 이름과 나이와 전공을 출력하시오.

```
Select ename, age, major
	From emp12
	Where age = (select max(age)
			From emp12
			Where major like '%통계%')
	And major like '%통계%';
```

문제271. KING 에게 보고하는 사원들의 이름을 출력하시오. (mgr 이 KING의 empno 인사원들)

```
Select ename
	From emp
	Where mgr = (select empno
				From emp
				Where ename = 'KING');
```

서브 쿼리 사용하기 2(다중 행 서브쿼리)

서브쿼리의 종류 3가지

1. 단일행 서브 쿼리 : 서브쿼리에서 메인쿼리로 하나의 값이 리턴되는 경우
   a. 연산자 : =, <, >, <=, >=, !=, <>, ^=
2. 다중행 서브 쿼리 : 서브쿼리에서 메인쿼리로 여러개의 값이 리턴되는 경우
   a. 연산자 : in, not in, >all, <all, >any, <any
3. 다중 컬럼 서브 쿼리 : 서브쿼리에서 메인쿼리로 여러개의 컬럼값이 리턴되는 경우

Ex)
직업이 SALESMAN 인 사원들과 월급이 같은 사원들의 이름과 월급을 출력하시오.

```
Select ename, sal
	From emp
	Where sal in (select sal
				From emp
				Where job in 'SALESMAN');
```

문제272. 통신사가 sk 인 학생들과 나이가 같은 학생들의 이름과 나이와 통신사를 출력하시오.

```
Select ename, age, telecom
	From emp12
	Where age in (select age
				From emp12
				Where telecom = 'sk');
```

문제273. 통신사가 sk 인 학생들과 나이가 같지 않은 학생들의 이름과 나이와 통신사를 출력하시오.

```
Select ename, age, telecom
	From emp12
	Where age not in (select age
```

```
From emp12
Where telecom = 'sk');
```

서브 쿼리 사용하기 3(NOT IN)

Ex)
관리자인 사원들의 이름을 출력하시오.
(자기 밑에 직속 부하가 한명이라도 있는 사람)

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

```
Select ename
      From emp
      Where empno in (7839, 7902, 7566, null);
```

```
Select ename
      From emp
      Where empno = 7839 or empno = 7902 or empno = 7566 or empno = null;
```

T or T or T or null

Where 절에 조건이 True 이면 결과가 잘 출력이 된다.

문제274. 관리자가 아닌 사원들의 이름을 출력하시오.

```
Select ename
      From emp
      Where empno not in (select mgr
                            From emp);
```

```
Select ename
      From emp
      Where empno not in (7839, 7902, 7566, null);
```

```
Select ename
      From emp
      Where empno != 7839 and empno != 7902 and empno != 7566 and empno != null;
```

T and T and T and null

Null 값으로 인해서 전체가 null이 되기 때문에 결과가 출력이 되지 않는다.

서브쿼리문에 not in 연산자 사용 시 주의사항 :
서브쿼리에서 null 값이 하나라도 리턴되면 결과가 출력되지 않는다.

문제275. 문제 274를 다시 출력하시오.

```
Select ename
	From emp
	Where empno not in (select nvl(mgr, 0)
				From emp);

Select ename
	From emp
	Where empno not in (select mgr
				From emp
				Where mgr is not null);
```

서브쿼리문 사용시 not in 연산자를 사용하면 반드시 서브쿼리에서 메인쿼리로 null 갑시 리턴되지 않도록 처리를 해줘야한다.

서브 쿼리 사용하기 4(EXISTS와 NOT EXISTS)

서브쿼리문에서 exists 와 not exists 를 사용해서 메인쿼리에 있는 데이터 중에 서브쿼리에 존재하는지 유무를 파악할 때 사용하는 SQL문법

Ex)
Emp 테이블에 권세원 데이터를 입력한다.

```
Insert into emp(empno, ename, sal, job, deptno)
	Values (9877, '권세원', 6000, 'ANALYST', 20);

Commit;
```

Ex)
Emp12 테이블에 있는 학생들 중에 사원 테이블에도 존재하는 학생이 있으면 출력하시오.

```
Select ename
	From emp12 e12
	Where exists ( select *
			From emp e
			Where e.ename = e12.ename);
```

Exists 문은 메인쿼리 컬럼이 서브쿼리 안으로 들어가면서 실행된다.
메인쿼리의 데이터 중에 서브쿼리에도 같은 데이터가 존재하는지 찾아본다.

문제276. 부서 테이블에서 부서번호와 부서위치를 출력하는데 부서

테이블에있는 부서번호중에 사원 테이블에도 존재하는 부서번호에 대한것만 출력하시오.

```
Select d.deptno, d.loc
        From dept d
        Where exists (select *
                        From emp e
                        Where e.deptno = d.deptno );
```

문제277. 위 문제를 존재하지 않는 부서번호와 부서위치를 출력하시오.

```
Select d.deptno, d.loc
        From dept d
        Where not exists (select *
                        From emp e
                        Where e.deptno = d.deptno );
```

문제278. 어제의 마지막 문제에 deptno 도 같이 출력되게 하시오.

```
Select e.deptno, sum(decode(d.loc,'NEW YORK',e.sal)) "NEW YORK",
        sum(decode(d.loc,'DALLAS',e.sal)) DALLAS,
        sum(decode(d.loc,'CHICAGO',e.sal)) CHICAGO
        From emp e, dept d
        Where e.deptno = d.deptno
        Group by e.deptno;
```

문제279. 위의 결과에 null 값 대신에 0으로 출력되게 하시오.

```
Select e.deptno, sum(decode(d.loc,'NEW YORK',e.sal, 0)) "NEW YORK",
        sum(decode(d.loc,'DALLAS',e.sal, 0)) DALLAS,
        sum(decode(d.loc,'CHICAGO',e.sal, 0)) CHICAGO
        From emp e, dept d
        Where e.deptno = d.deptno
        Group by e.deptno;
```

20.11.05

2020년 11월 5일 목요일 오전 9:36

서브 쿼리 사용하기 5(HAVING절의 서브 쿼리)

Select 문장에서 서브쿼리를 쓸 수 있는 절 : group by절을 제외하고 모두 사용가능

Select 서브쿼리 사용가능 (scalar subquery)
Form 서브쿼리 사용가능 (in line view)
Where 서브쿼리 사용가능
Group by
Having 서브쿼리 사용가능
Order by 서브쿼리 사용가능 (scalar subquery)

문제280. JAMES 보다 더 많은 월급을 받는 사원들의 이름과 월급을 출력하시오.

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

문제281. 직업, 직업별 토탈 월급을 출력하시오.

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

문제282. 위의 결과를 다시 출력하는데 직업이 SALESMAN의 토탈월급보다 더 큰 직업들만 출력하시오.

```
Select job, sum(sal)
      From emp
      Group by job
      Having sum(sal) > (select sum(sal)
                              From emp
                              Where job = 'SALESMAN');
```

그룹함수로 조건을 주는 절은 having 절이다.
Where 절에 작성하면 오류가 발생한다.

문제283. 부서번호와 부서번호별 인원수를 출력하는데 10번 부서번호의 인원수보다 더 큰것만 출력하시오.

```
Select deptno, count(*)
        From emp
        Group by deptno
        Having count(*) > (select count(*)
                                From emp
                                Where deptno = 10);
```

서브 쿼리 사용하기 6(FROM절의 서브 쿼리)

From 절에도 서브쿼리를 사용할 수 있는데, from 절의 서브쿼리는 in line view라고 한다.

Ex)
이름, 월급, 순위를 출력하는데 순위가 월급이 높은 사원순으로 순위를 부여하시오.

```
Select ename, sal, dense_rank() over (order by sal desc) 순위
        From emp;
```

Ex)
위의 결과에서 순위가 4등인 사원만 출력하시오.

```
Select ename, sal, dense_rank() over (order by sal desc) 순위
        From emp
        Where 순위 = 4;
```

위의 SQL문은 From절을 실행하고 where 절을 실행하기 때문에 emp테이블에는 순위라는 컬럼이 없기 때문에 에러가 발생한다.

순위가 4등인 사원의 이름과 월급과 순위를 출력하려면 from 절의 서브쿼리를 사용해야한다.

```
4.Select *
1.From(3.Select ename, sal, dense_rank() over (order by sal desc) 순위
        2.From emp
        );
```

From 절의 서브쿼리문의 결과가 마치 하나의 테이블 처럼 만들어져서 서브쿼리문의 결과 데이터가 메모리에 올라가게 된다.

```
5.Select *
        1.From(3.Select ename, sal, dense_rank() over (order by sal desc) 순위
                2.From emp
```

```
        )
    4.Where 순위 = 4;
```

문제284. 직업, 이름, 월급, 순위를 출력하는데 순위가 직업별로 각가 월급이 높은 순서대로 순위를 부여하시오.

```
Select job, ename, sal, dense_rank() over(partition by job order by sal desc) 순위
    From emp;
```

문제285. 위의 결과에서 순위가 1등인 사원들만 출력하시오.

```
Select *
    From (
        Select job, ename, sal,
        dense_rank() over(partition by job order by sal desc) 순위
        From emp
        )
Where 순위 = 1;
```

문제286. emp12 테이블에서 통신사별로 나이가 가장 많은 학생의 이름과 나이와 통신사와 나이의 순위를 출력하시오.

```
Select *
    From (
        Select telecom, ename, age,
            Dense_rank() over (partition by telecom order by age desc) 순위
        From emp12
        )
    Where 순위 = 1;
```

Price 테이블 소개

M_NAME : 마트 이름 또는 시장이름
A_NAME : 식료품 이름
A_PRICE : 식료품 가격

문제287. price 테이블의 전체 건수가 어떻게 되는지 확인하시오.

```
Select count(*)
    From price;
```

문제288. 서울시에서 가장 비싼 식료품의 이름과 가격과 파는 곳을 출력하시오.

```
Select *
    From (
        Select a_name, a_price, m_name,
            Dense_rank() over (order by a_price desc) 순위
        From price
        )
    Where 순위 = 1;
```

서브 쿼리 사용하기 7(SELECT절의 서브 쿼리)

Select 절에 서브쿼리를 사용할 수 있는데 select 절의 서브쿼리를 scalar subquery 라고 한다.

Ex)
이름, 월급, 사원테이블의 평균 월글을 출력하시오. (자기의 월급과 평균월급을 비교해 보기)

```
Select ename, sal, (select avg(sal)
                    From emp) 평균월급
    From emp;
```

문제289. 사원이름, 월급, 사원 테이블의 최대월급, 사원 테이블의 최소월급을 출력하시오.

```
Select ename, sal,
    (select max(sal) from emp) 최대월급,
    (select min(sal) from emp) 최소월급
    From emp;
```

Select 절의 서브쿼리인 스칼라 서브쿼리의 특징 :
스칼라 서브쿼리는 단 한개값만 리턴 할 수 있다.

문제290. emp12 테이블에서 이름, 나이, 우리반 나이의 평균나이를 출력하시오.

```
Select ename, age, (select avg(age) from emp12) 평균나이
    From emp12;
```

문제291. 위의 결과를 소수점 이하가 나오지 않게 반올림 하시오.

```
Select ename, age, round((select avg(age) from emp12)) 평균나이
	From emp12;

Select ename, age, (select round(avg(age)) from emp12) 평균나이
	From emp12;
```

문제292. 위의 결과에서 학생의 나이가 평균나이보다 더 큰 학생들의 이름과 나이와 평균나이를 출력하시오.

```
Select ename, age, round((select avg(age) from emp12)) 평균나이
	From emp12
	Where age > (select avg(age) from emp12);



Select *
	From (
		Select ename, age, (select round(avg(age))
					From emp12) 평균나이
			From emp12
		)
	Where age > 평균나이;
```

문제293. 사원 테이블에서 이름, 월급, 사원 테이블의 최대월급, 사원 테이블의 최소월급, 사원 테이블의 평균월급을 출력하시오.

튜닝 전

```
Select ename, sal, (select max(sal) from emp) 최대월급,
	(select min(sal) from emp) 최소월급,
	(select avg(sal) from emp) 평균월급
	From emp;
```

튜닝 후

```
Select ename, sal, max(sal) over () 최대월급,
			Min(sal) over () 최소월급,
			Avg(sal) over () 평균월급
	From emp;
```

문제294. emp12 테이블에서 이름, 나이, 최대나이, 최소나이, 평균나이, 인원수를 출력하시오.

```
Select ename, age, max(age) over () 최대나이,
            Min(age) over () 최소나이,
            Avg(age) over () 평균나이,
            Count(*) over () 인원수
       From emp12;
```

Select 절의 서브쿼리인 스칼라 서브쿼리가 성능이 느리므로 위와 같이 데이터 분석함수를 이용해서 튜닝을 하면 빠르게 대용량 데이터의 데이터를 검색 할 수 있다.

데이터 입력하기(INSERT)

테이블에 데이터를 입력하는 SQL 문장

Ex)
Insert into emp(empno, ename, sal) -> 데이터 입력할 컬럼들 기술
Values (1234, 'JACK', 4500); -> 위의 컬럼 순서대로 값을 기술

문제295. 사원 테이블에 아래의 데이터를 입력하시오.

사원번호 : 2939
서원이름 : JANE
월급 : 4700
입사일 : 오늘 날짜

```
Insert into emp(empno, ename, sal, hiredate)
       Values (2939, 'JANE', 4700, to_date('20/11/05', 'RR/MM/DD'));
```

날짜를 입력할 때는 to_date를 사용해서 입력해야 한다.
실패하지 않고 확실하게 날짜 데이터를 입력하는 방법이다.
위와 같이 날짜를 입력하면 날짜는 정확하게 2020년 11월 05일로 입력이 되고 시/분/초 는 00시 00분 00초로 입력이 된다.

```
Delete from emp where empno = 2939;
```

문제296. 오늘 입사한 사원의 이름과 입사일을 출력하시오.

```
Select ename, hiredate
       From emp
       Where hiredate = '20/11/05';
```

Hiredate = sysdate; 라고 작성하면 조회가 되지 않는다.
Sysdate 는 날짜 뿐만 아니라 시/분/초 도 출력하는데 입력할 때의 시/분/초 와 현재의 시/분/초가 다르기 때문에 조회 되지 않는다.

문제297. 2020년 11월 05일에 입사한 사원의 이름과 입사일을 출력하시오.

```
Select ename, hiredate
	From emp
	Where hiredate = to_date('2020/11/05', 'RRRR/MM/DD');
```

Where 절의 컬럼명쪽에 가급적 함수를 사용하지 않고 검색하는 것이 더 빠르게 데이터를 검색할 수 있는 방법이다.

문제298. 아래의 데이터를 사원 테이블에 입력하시오.

```
Insert into emp (empno, ename, sal, deptno)
	Values (4945, 'MIKE    ', 3000, 20);
```

문제299. 이름이 MIKE인 사원의 이름과 월급을 출력하시오.

튜닝 전

```
Select ename, sal
	From emp
	Where rtrim (ename) = 'MIKE';
```

튜닝 후

```
Select ename, sal
	From emp
	Where ename like 'MIKE%';
```

Where 절에 컬럼에는 함수를 가급적 사용하지 않아야 검색 성능이 좋아진다.

문제300. 아래의 데이터를 emp12 테이블에 입력하시오.

이름 : 김인호
성별 : 남
이메일 : abcd1234@gmail.com
주소 : 서울시 강남구 역삼동 삼원빌딩 4층

```
Insert into emp12(ename, gender, email, address)
        Values ('김인호', '남', 'abcd1234@gmail.com', '서울시 강남구 역상동 삼원빌딩 4층');
```

데이터 수정하기(UPDATE)

데이터를 수정하는 SQL문

Ex)

```
Update emp
        Set sal = 0
        Where ename = 'SCOTT';
```

SCOTT의 월급을 0으로 수정하는 업데이트 문장

```
Rollback;
```

문제301. king의 월급을 9000으로 변경하시오.

```
Update emp
        Set sal = 9000
        Where ename = 'KING';
```

Commit; -> 지금까지 변경한 모든 작업을 다 database에 영구히 저장한다.

Rollback; -> commit 이후에 작업한 모든 변경사항을 취소한다.

문제302. 직업이 SALESMAN인 사원들의 커미션을 9500으로 수정하시오.

```
Update emp
        Set comm = 9500
        Where job = 'SALESMAN';
```

데이터 삭제하기(DELETE, TRUNCATE, DROP)

오라클에서 데이터를 삭제하는 방법 3가지

1. Delete
2. Truncate
3. drop

Ex)

```
Delete from emp
Where empno = 7788;
```

사원번호가 7788번인 사원을 삭제 하겠다.

문제303. 직업이 SALESMAN인 사원들을 삭제하시오.

```
Delete from emp
Where job = 'SALESMAN';
```

```
Delete from emp;
Commit;
```

타임머신 기능을 이용해서 과거로 emp 테이블만 되돌린다.

1. Emp 테이블을 flachback이 가능한 상태로 변경한다.
   (타임머신 기능을 쓸 수 있도록 설정한다.)

   ```
   Alter table emp enable row movement;
   ```

2. 현재 시간에서 5분 전으로 emp 테이블을 되돌린다.

   ```
   Flashback table emp to timestamp
   (systimestamp - interval '5' minute);

   Commit;
   ```

   골든타임 시간 확인

   ```
   Show parameter undo_retention
   ```

Truncate 명령어 : 테이블 구조만 남기고 다 지운다.

1. 데이터 모두 삭제
2. Rollback 불가능
3. Flashback 불가능

```
Truncate table emp;
```

백업받은 데이터로 복구하는 수 밖에 없다.

Drop 명령어

1. 모든 데이터 삭제
2. 테이블 구조까지 모두 삭제
3. Rollback 불가능
4. Flashback 가능

```
Drop table emp;
```

Show recyclebin;
휴지통에 있는 데이터 확인

Flashback table emp to before drop;
복원하는 명령어

문제304. 부서번호, 부서번호별 토탈월급을 출력하는데 가로로 출력하시오.

(sum + decode 이용)

```
Select sum(decode(deptno, 10, sal)) "10",
       Sum(decode(deptno, 20, sal)) "20",
       Sum(decode(deptno, 30, sal)) "30"
From emp;
```

문제305. 위의 결과에서 집계결과를 아래와 같이 출력하시오.

JOB	10	20	30	토탈값
SALESMAN			5600	5600
CLERK	1300	1900	950	4150
ANALYST		6000		6000
MANAGER	2450	2975	2850	8275
PRESIDENT	5000			5000
전체토탈	8750	10875	8400	29025

```
Select job, sum(decode(deptno, 10, sal)) "10",
       Sum(decode(deptno, 20, sal)) "20",
       Sum(decode(deptno, 30, sal)) "30",
       Sum(sal) "토탈값"
From emp
Group by job
Union
Select '전체토탈' job, (select sum(sal) from emp where deptno = 10) "10",
       (select sum(sal) from emp where deptno = 20) "20",
       (select sum(sal) from emp where deptno = 30) "30",
       (select sum(sal) from emp)
       From emp;
```

20.11.06

2020년 11월 6일 금요일 오전 10:34

데이터 저장 및 취소하기(COMMIT, ROLLBACK)

Commit 은 지금까지 변경한 데이터베이스 작업들 (DML 문장) 을 데이터베이스에 영구히 반영하겠다 라는 TCL (Trasaction Control Language)문

Rollback은 지금까지 변경한 데이터 베이스 작업들 (DML 문장) 을 취소하는 명령어이다. 마지막으로 commit 한 이후의 작업한 DML 작업들을 취소한다.

SQL의 종류

1. Query 문 : select 문의 6가지 절, 조인, 서브쿼리
2. DML 문 : insert, update, delete, merge
3. DDL 문 : create, alter, drop, truncate, rename
4. DCL 문 : grant, revoke
5. TCL 문 : commit, rollback, savepoint

문제306. rollback이 마지막 commit 한 시점까지만 rollback이 된다는 것을 테스트 하기 위해 commit을 ㅣ금 수행하고 사원 테이블의 월급을 모두 0으로 변경하시오.

```
Update emp
      Set sal = 0;

Commit;

Delete from emp;

Rollback;
```

암시적 commit

1. 정상 종료 (exit)
2. DDL문 수행 : create, alter, drop, truncate, rename
3. DCL문 수행 : grant, revoke

Ex)
Emp 테이블을 생성하는 스크립트를 편하게 저장하고 수행하는 방법

1. 도스창을 열고 scott으로 접속한다.
   Sqlplus scott/tiger
2. Emp 테이블 생성하는 스크립트를 demo . Sql 로 저장한다
   Ed demo.Sql
   메모장이 나오면 카페에 있는 데이터 스크립트를 저장한다
   $dir 디렉터리 확인
3. Demo . Sql 스크립트를 수행한다.
   @demo.sql

Ex)
SQL> delete from emp;
SQL> create table emp902
(empno, number(10),
Ename, varchar2(10));

오라클은 DML 문장을 수행하고서 정상종료 또는 DDL문 수행 또는 DCL문을 수행하지 않고 명시적으로 COMMIT을 수행하지 않았다면 DML 작업을 취소(rollback) 할 수 있다.

MSSQL
Begin tran <- 앞에 이 명령어를 수행하지 않고 delete 문장을 수행하면 자동 commit 된다.

Delete from emp;

백업하고 중요 DML 작업을 수행한다.

암시적 rollback

1. Database 시스템이 비정상 종료 되었을 때 (정전)

데이터 입력, 수정, 삭제 한번에 하기(MERGE)

데이터 입력과 수정과 삭제를 한번에 수행하는 명령어 이고 SQL 튜닝을 위해서 자주 사용되는 SQL

Ex)
이름과, 부서위치를 출력하시오.

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

Ex)
사원 테이블에 부서위치 컬럼을 추가한다.

Alter table emp

```
Add loc varchar2(10);
```

Ex)
사원 테이블에 추가한 부서위치 컬럼에 데이터를 해당 사원의 부서위치로 값을 갱신하시오.

```
Merge into emp e
Using dept d
On (e.deptno = d.deptno)
When matched then
Update set e.loc = d.loc;
```

Merge into 다음에는 변경할 타깃(target) 테이블명을 작성한다.
Using 다음에는 소스(source) 테이블명을 작성한다.
On 다음에는 타깃과 소스 테이블간의 연결고리를 작성한다.
When matched then 다음에 변경할 update 문을 작성한다.

문제307. emp12 테이블에 telecom_price 컬럼을 추가하고 해당 telecom의 요금으로 값을 갱신하시오.

```
Alter table emp12
    Add telecom_price number(10);

Merge into emp12 e
Using telecom_price t
On (e.telecom = t.telecom)
When matched then
Update set e.telecom_price = t.price;
```

문제308. 부서위치, 부서위치별 토탈 월급을 출력하시오.

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

문제309. 부서번호, 부서번호별 토탈월급을 출력하시오.

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

문제310. 사원 테이블과 급여등급(salgrade) 을 조인해서 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;
```

문제311. 사원테이블에 급여 등급 컬럼을 추가하시오.

```
Alter table emp
    Add grade number(10);
```

문제312. 사원 테이블에 추가한 grade 컬럼에 해당 사원의 급여등급으로 값을 갱신하시오.

```
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;
```

문제313. 부서명(dname) 컬럼을 emp 테이블에 추가하고 해당 사원의 부서명으로 emp 테이블의 dname을 갱신하시오.

```
Alter table emp
    Add dname varchar2(10);

Merge into emp e
Using dept d
On (e.loc = d.loc)
When matched then
Update set e.dname = d.dname;
```

락(LOCK) 이해하기

특정 창(session) 에 접속한 유저(scott)가 KING 의 월급을 9000 으로 변경하고 있는 상태에서 다른 창(session)에 접속한 유저(scott)가 KING의 월급을 변경할 수 없도록 막는 기능을 lock라고 한다.

내가 변경한 데이터를 남이 보려면 내가 commit 해줘야 볼 수 있다.
내가 commit 하지 않으면 내가 변경한 데이터를 남이 볼 수 없다.

Update를 수행하면 update를 수행하고 있는 행(row)에 락(lock)을 건다.

Ex)
```
Update emp
Set sal = 8000
Where ename = 'KING';
```

위와 같이 update를 하면 KING 의 전체 행에 락(lock)을 건다.
다른 창 (session)에 접속한 유저가 KING의 데이터를 절대로 갱신할 수 없다.

Ex)
Update emp 도스창 A
Set sal = 8700
Where ename = 'KING';

Update emp 도스창 B
Set deptno = 30
Where ename = 'KING';

도스창 A 에서 KING 행에 lock을 걸었기 때문에 도스창 B와 서로 충돌한다.

Ex)
Update emp 도스창 A
Set sal = 9700
Where ename = 'JAMES';

Update emp 도스창 B
Set sal = 8000
Where ename = 'ALLEN';

도스창 A는 JAMES행에 lock을 걸었고 도스창 B는 ALLEN의 행에 lock을 걸어 서로 충돌하지 않는다.

오라클에 lock이 있어서 사용자들은 항상 일관된 데이터를 볼 수 있다.

SELECT FOR UPDATE절 이해하기

보통 lock은 update 문을 수행할 때 주로 걸리나 select 을 수행할 때는 lock이 걸리지 않는데 select 를 수행할 때도 lock을 걸고 싶으면 select for update 문을 이용하면 된다.

내가 어떤 데이터를 보고 있는 동안 그 누구도 이 데이터를 갱신하지 못하도록 막고 싶을 때 Select for update 문을 사용한다.

Ex)
코스트코 밤 9시에 문들 닫는데 매장에 진열된 상품의 갯수를 파악해서 모자란 상품을 주문해서 채워넣으려고 한다.

9시를 기준으로 현재 매장의 상품들의 상품 갯수를 파악하고 싶다.
항상 10개의 삼품이 있어야하는 커피 제품이 있다면 지금 남은게 3개면 7개를 주문하려고

한다.

그런데 상품의 갯수가 계속 변경되면 새로 주문을 넣을 때 혼란스러우므로 9시 현재를 기준으로 그 어떤 데이터도 갱신하지 못하도록 막아버린다.

Ex)
Select ename, sal 도스창 A
  From emp
  Where ename = 'BLAKE'
  For update;

Update emp 도스창 B
  Set sal = 0
  Where ename = 'BLAKE';

서브 쿼리를 사용하여 데이터 입력하기

지금까지 배운 insert 문장은 한번에 한건만 입력할 수 있었다.

Insert into emp(empno,ename,sal)
  Values(1234,'AAA',4000);

그런데 서브쿼리를 사용한 insert 문장을 이용하면 한번에 여러건의 데이터를 입력할 수 있게 된다.

Create table emp12_backup3
As
Select *
  From emp12;

Truncate table emp12;

Insert into emp12
Select *
  From emp12_backup3;

Commit;

문제314. dept 테이블을 백업 하시오.

Create table dept_backup
As
Select *
  From dept;

문제315. dept 테이블을 truncate 하시오.

```
Truncate table dept;
```

문제316. dept 테이블을 dept_backup 테이블을 이용해서 복구 하시오.

```
Insert into dept
Select *
	From dept_backup;

Commit;
```

서브 쿼리를 사용하여 데이터 수정하기

```
Update emp <- 서브쿼리 사용가능
	Set sal = 8900 <- 서브쿼리 사용가능
	Where ename = 'SCOTT'; <- 서브쿼리 사용가능
```

Ex)
SCOTT 보다 더 많은 월급을 받는 사원들의 직업을 SALESMAN으로 변경하시오.

```
Update emp
	Set job = 'SALESMAN'
	Where sal > (Select sal
			From emp
			Where ename = 'SCOTT');

Rollback;
```

문제317. ALLEN보다 더 늦게 입사한 사원들의 커미션을 9000으로 수정하시오.

```
Update emp
	Set comm = 9000
	Where hiredate > (select hiredate
				From emp
				Where ename = 'ALLEN');
```

문제318. SMITH와 같은 직업을 갖는 사원들의 월급을 9800으로 변경하시오.(SMITH는 제외)

```
Update emp
	Set sal = 9800
	Where job = (select job
```

```
                    From emp
                    Where ename = 'SMITH')
    And ename != 'SMITH';
```

문제319. ALLEN의 월급을 KING의 월급으로 변경하시오.

```
Update emp
    Set sal = (select sal
                    From emp
                    Where ename = 'KING')
    Where ename = 'ALLEN';
```

문제320. JONES보다 월급이 많은 사원들의 직업을 MARTIN의 직업으로 변경하시오.

```
Update emp
    Set job = (select job
                    From emp
                    Where ename = 'MARTIN')
    Where sal > (select sal
                    From emp
                    Where ename = 'JONES');
```

서브 쿼리를 사용하여 데이터 삭제하기

Ex)
SCOTT보다 월급을 많이 받는 사원들을 삭제하시오.

```
Delete from emp
Where sal > (select sal
                From emp
                Where ename = 'SCOTT');
```

문제321. ALLEN보다 늦게 입사한 사원들을 삭제하시오.

```
Delete from emp
Where hiredate > (select hiredate
                    From emp
                    Where ename = 'ALLEN');
```

문제322. JONES와 같은 부서번호에서 일하는 사운들을 삭제하시오.

```
Delete from emp
Where deptno = (select deptno
```

```
From emp
Where ename = 'JONES');
```

서브 쿼리를 사용하여 데이터 합치기

Merge 문에 서브쿼리를 사용하기

Ex)
부서번호, 부서번호별 토탈월급을 출력하시오.

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

Ex)
부서테이블에 sumsal 이라는 컬럼을 추가하시오.

```
Alter table dept
    Add sumsal number(10);
```

Ex)
위의 dept테이블에 추가한 sumsal 컬럼에 해당 부서번호의 토탈월급으로 값을 갱신하시오.

```
Merge into dept d
Using (select deptno, sum(sal) 토탈
        From emp
        Group by deptno) e
On (e.deptno = d.deptno)
When matched then
Update set d.sumsal = e.토탈
```

문제323. 부서번호, 부서번호별 인원수를 출력하시오.

```
Select deptno, count(*) 인원수
    From emp
    Group by deptno;
```

문제324. 부서테이블에 cnt라는 컬럼을 추가하시오.

```
Alter table dept
    Add cnt number(10);
```

문제325. dept테이블에 추가한 cnt 컬럼에 해당 부서번호에 인원수로 값을 갱신하시오.

```
Merge into dept d
Using (select deptno, count(*) 인원수
        From emp
        Group by deptno) e
On (d.deptno = e.deptno)
When matched then
Update set d.cnt = e.인원수;
```

Using 절에 사용한 서브쿼리문의 결과가 마치 테이블처럼 이 merge 문에서 사용되고 있다.

계층형 질의문으로 서열을 주고 데이터 출력하기 1

카카오 면접때 받았던 질문

알고리즘 문제 1)

1. 1부터 10까지의 합을 구하시오
   SQL로 수행 가능

```
Select 1+2+3+4+5+6+7+8+9+10
    From dual;
```

2. 1부터 100까지의 합을 구하시오
   계층형 질의문을 사용하면 쉽게 할 수 있다.

```
Select level
    From dual
    Connect by level <= 100;
```

1번부터 connect by 절의 level 다음에 나오는 숫자만큼 숫자가 증가해서 출력된다.

```
Select sum(level)
    From dual
    Connect by level <= 100;
```

알고리즘 문제2)

문제326. 1부터 10 사이의 숫자를 출력하는데 짝수만 출력하시오.

```
Select level
    From dual
    Where mod(level,2) = 0
    Connect by level <= 10;
```

문제327. 밑수가 자연상수 (e) 이고 진수가 10인 로그값을 출력하시

오.

```
Select ln(10) from dual;
```

문제328. 자연상수 (e) 의 10승을 구하시오.

```
Select exp(10) from dual;
```

알고리즘 문제3)

문제329. 1부터 10까지의 곱을 SQL로 출력하시오.

- 로그의 성질을 이용하여 풀 수 있다.

```
Select exp(sum(ln(level)))
        From dual
        Connect by level <= 10;
```

20.11.09

2020년 11월 9일 월요일 오전 9:31

계층형 질의문으로 서열을 주고 데이터 출력하기 2

순위와 서열을 출력하는 SQL 문

계층형 질의절은 WHERE 절 다음에 기술하며, from 절이 수행된 후 수행된다.
Start with 절과 connect by 절로 구성되며, start with 절이 수행된 후 connect by 절이 수행된다.
Start with 절은 생략이 가능하다.

Start with 절 : 루트 노드를 생성하며 한 번만 수행된다.
Connect by 절 : 루트 노드의 하위 노드를 생성하며 조회결과가 없을 때까지 반복 수행된다.

Ex)
1부터 10 까지의 숫자를 출력하시오.

```
Select level <- connect by 절을 사용해야지만 출력되는 컬럼
    From dual <- 결과를 보기 위한 가상의 테이블
    Connect by level <= 10; <- level이 10보다 작거나 같을때까지 계속해서 연결한다.
```

Ex)
사원 테이블의 서열순서를 출력하시오.

```
Select level, empno, ename, mgr
    From emp
    Start with ename = 'KING' <- KING 을 root 로 해서 시작한다.
    Connect by prior empno = mgr;
```

Select 절에 level 컬럼은 emp테이블에 존재하지 않는 컬럼이다. 그러나 connect by 절을 사용하였기 때문에 출력 될 수 있다.

문제330. emp 테이블의 서열을 SQL로 시작화 하시오.

```
Select rpad(' ', level*2) || ename as employee, level
    From emp
    Start with ename = 'KING'
    Connect by prior empno = mgr;
```

rpad(' ', level*2) 은 공백(' ') level*2 의 숫자 만큼 채워넣는다.

문제331. 위의 결과에서 BLAKE는 제외하고 출력하시오.

```
Select rpad(' ', level*2) || ename as employee, level
        From emp
        Where ename != 'BLAKE'
        Start with ename = 'KING'
        Connect by prior empno = mgr;
```

문제332. 위의 결과에서 BLAKE의 팀원들도 제외하고 출력하시오.

```
Select rpad(' ', level*2) || ename as employee, level
        From emp
        Start with ename = 'KING'
        Connect by prior empno = mgr
              And ename != 'BLAKE';
```

하위노드의 모든 데이터를 출력되지 않게 하려면 where 절을 사용하지 않고 connect by 절에 조건을 주면 된다.

문제333. BLAKE 와 BLAKE의 팀원들을 포함시킨 서열을 출력하는 SQL을 아래와 같이 실행하는데 월급이 높은 순서대로 출력하시오.

```
Select rpad(' ', level*2) || ename as employee, level, sal
        From emp
        Start with ename = 'KING'
        Connect by prior empno = mgr
        Order by sal desc;
```

위의 결과는 월급이 높은 순서대로 정렬이 되면서 서열로 정렬된 결과가 사라진다.

문제334. 위의 결과를 다시 서열로 정렬된 결과를 유지 하면서 월급이 높은 순서대로 출력하시오.

```
Select rpad(' ', level*2) || ename as employee, level, sal
        From emp
        Start with ename = 'KING'
        Connect by prior empno = mgr
        Order siblings by sal desc;
```

결과를 보면 같은 서열 내에서 월급이 높은 순서대로 정렬이 된다.
계층형 질의문을 사용할때 order by 절을 사용할 때는 siblings라는 키워드와 함께 사용하여야 한다.

문제335. 이름과 입사일과 서열 순위를 출력하는데 서열 순위를 유지하면서 먼저 입사한 사원순으로 정렬하여 출력하시오.

```
Select rpad(' ', level*2) || ename as employee, level, hiredate
    From emp
    Start with ename = 'KING'
    Connect by prior empno = mgr
    Order siblings by hiredate;
```

계층형 질의문으로 서열을 주고 데이터 출력하기 3

계층형 질의문과 짝꿍 함수인 sys_connect_by_path 를 이용하면 listagg를 사용한 것처럼 결과를 가로로 출력할 수 있다.

Ex)

```
Select ename, sys_connect_by_path(ename, '/ ') path
    From emp
    Start with ename = 'KING'
    Connect by prior empno = mgr;
```

문제336. 위의 예제 결과에서 앞의 / 를 잘라내시오.

```
Select ename, ltrim(sys_connect_by_path(ename, '/ '),'/ ') path
    From emp
    Start with ename = 'KING'
    Connect by prior empno = mgr;
```

계층형 질의문을 작성할 때 반드시 알아야하는 두 가지 키워드

1. Order by 절의 siblings
2. 서열을 가로로 출력하는 sys_connect_by_path 함수

계층형 질의문으로 서열을 주고 데이터 출력하기 4

계층형 질의문에서도 조인문장을 같이 사용 할 수 있다.

Ex)
이름, 서열, 월급, 부서위치를 출력하시오.

```
Select rpad(' ', level*2) || e. ename as employee, level, e. sal, d. loc
    From emp e, dept d
    Where e. deptno = d. deptno
    Start with ename = 'KING'
    Connect by prior empno = mgr;
```

문제337. 이름, 서열, 월급, 급여등급(grade) 을 출력하시오.

```
Select rpad(' ', level*2) || e. ename as employee, level, e.sal, s. grade
        From emp e, salgrade s
        Where e. sal between s. losal and s. hisal
        Start with ename = 'KING'
        Connect by prior empno = mgr;
```

문제338. 위의 결과를 다시 출력하는데 서열의 정렬을 유지 하면서 급여등급이 높은 사원부터 출력하시오.

```
Select rpad(' ', level*2) || e. ename as employee, level, e.sal, s. grade
        From emp e, salgrade s
        Where e. sal between s. losal and s. hisal
        Start with ename = 'KING'
        Connect by prior empno = mgr
        Order siblings by s. grade desc;
```

문제339. DALLAS에서 근무하는 사원들의 이름, 서열, 부서위치를 출력하시오.(서열은 전체 사원을 기준으로 서열순위를 부여하시오.)

```
Select rpad(' ', level*2) || e. ename as employee, level, d. loc
        From emp e, dept d
        Where e. deptno = d. deptno
                And d. loc = 'DALLAS'
        Start with ename = 'KING'
        Connect by prior empno = mgr;
```

알고리즘 문제4)

문제340. 계층형 질의문을 이용해서 구구단 2단을 출력하시오.

2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
.
.
.
2 x 9 = 18

```
Select '2 x ' || level || ' = ' || 2* level
        From dual
        Connect by level <= 9;
```

알고리즘 문제5)

문제341. 계층형 질의문을 이용해서 삼각형을 출력하시오.

```
Select lpad('❤', level, '❤')
       From dual
       Connect by level <= 10;
```

알고리즘 문제6)

문제342. 아래와 같이 결과를 출력하시오.

❤
❤❤
❤❤❤
❤❤❤❤
❤❤❤❤❤
❤❤❤❤
❤❤❤
❤❤
❤

```
Select lpad('❤', level, '❤')
       From dual
       Connect by level <= 5
Union all
Select lpad('❤❤❤❤', 5 - level)
       From dual
       Connect by level <= 4;
```

일반 테이블 생성하기(CREATE TABLE)

DDL문 (Data Definition Language) : create, alter, drop, truncate, rename

Ex)
겨울왕국 대본을 오라클에 입력하고 elsa가 많이 나오는지 anna가 많이 나오는지 또는 긍정단어가 많은지 부정단어가 많은지 등을 SQL로 검색하려면 DDL문을 이용해서 테이블을 생성할 수 있어야 한다.

Ex)
테이블 생성 스크립트

```
Create table emp500  -> 테이블명
( empno number (10),  -> 컬럼명  데이터유형(길이)
  ename varchar2 (20),
  sal number (10) );
```

데이터 유형 (길이)

1. 문자형 : varchar2, char
2. 숫자형 : number
3. 날짜형 : date

number (10) -> 숫자 10자리를 허용한다.
varchar2 (20) -> 영어철자 20자리를 허용한다.

문제343. emp500 테이블에 아래의 데이터를 입력하시오.

1111 scott 3000
2222 smith 2900

```
Create table emp500
( empno number (10),
  ename varchar2 (20),
  sal number (10) );

Insert into emp500 (empno, ename, sal)
        Values (1111, 'SCOTT', 3000);

Insert into emp500 (empno, ename, sal)
        Values (2222, 'SMITH', 2900);
```

문제344. 아래의 테이블의 이름을 501로 해서 생성하시오.

Empno
Ename
Sal
Hiredate
Deptno

```
Create table emp501
( empno number (10),
  ename varchar2 (20),
  sal number (10),
  hiredate date,
  deptno number (10) );
```

Datatype 의 유형

데이터 유형	설명
Char	고정길이 문자 데이터 유형이고 최대길이가 2000이다.
Varchar2	가변길이 문자 데이터 유형이고 최대길이가 4000이다.

Long	가변길이 문자 데이터 유형이고 최대 2GB 까지 데이터 허용
Clob	문자 데이터 유형이고 최대 4GB 까지 문자 데이터 허용
Blob	바이너리 데이터 유형이고 최대 4GB (사진, 동영상)
Number	숫자 데이터 유형이고 최대 38 까지 허용
Date	날짜 데이터 유형이고 기원전 4712년 1월 1일 부터 기원후 9999년 12월 31일 까지 날짜를 허용

Number (10, 2) : 숫자 데이터 10자리를 허용하는데 그중에 소수점 2자리를 허용한다.
Ex) 3500.23

문제345. emp501 테이블에 데이터를 2건 입력하시오.

```
Insert into emp501
      Values (7839, 'KING', 5000, to_date('81/11/17', 'RR/MM/DD'), 10);

Insert into emp501
      Values (7698, 'BLAKE', 2850, to_date('81/05/01', 'RR/MM/DD'), 30);
```

Ex)
Long 데이터 타입은 아주 긴 텍스트 데이터를 입력할 때 사용하는 데이터 유형이다.

```
Create table profile2
( ename varchar2 (20),
  self_intro long );

Insert into profile2 (ename, self_intro)
      Values ('김인호', '어렸을때 부터 우리집은 가난했었습니다.
                  그리고 어머니는 짜장면이 싫다고 하셨습니다.
                  야히 야히야');

Commit;
```

문제346. 겨울왕국 대본을 입력하기 위한 테이블을 winter_kingdom 이라는 이름으로 생성하시오.

```
Create table winter_kingdom
(win_text long);
```

문제347. 영화 겨울왕국 대본에서 elsa가 많이 나오는지 anna가 많이 나오는지 확인하시오.

```
Select regexp_count (lower(win_text), 'elsa') as cnt
```

```
From winter_kingdom;
```

Win_text를 전부 소문자로 변경하고 regexp_count를 이용해서 스크립트 한행 한행을 모두 살펴봐서 elsa라는 단어가 포함되어 있으면 카운트 한다.

```
Select sum (regexp_count (lower(win_text), 'elsa')) as cnt
    From winter_kingdom;
```

카운트 된 숫자들을 sum 함수를 이용해서 모두 더한다.

```
Select sum (regexp_count (lower(win_text), 'anna')) as cnt
    From winter_kingdom;
```

정규식 함수인 regexp_count 를 이용하면 쉽게 구현할 수 있다.

위의 SQL이 유용한 경우

1. 뉴스기사를 분석해서 회사 주식에 영향을 주는지 확인
2. 외국 캐냐의 사례중 은행 고객 대출 한도를 정할 때
   대출 신청자의 SNS 글을 분석해서 긍정적인 단어가 많은지 부정적인 단어가 많은지 분석해서 대출여부를 결정

문제348. 겨울왕국 테이블을 drop 하고 다시 생성하는데 데이터 타입을 long 말고 varchar2 (4000)으로 하시오.

```
Drop table winter_kingdom;

Create table winter_kingdom
( win_text varchar2(4000) );
```

문제349. 긍정단어를 저장하기 위한 positive라는 테이블을 아래와 같이 생성하시오.

```
Create table positive
( p_text varchar2(2000) );
```

문제350. 부정단어를 저장하기 위한 negative라는 테이블을 아래와 같이 생성하시오.

```
Create table negative
( n_text varchar2(2000) );
```

문제351. 긍정단어는 positive 테이블에 입력하고 부정단어는 negative 테이블에 입력하시오.

문제352. 겨울왕국 대본에는 긍정단어가 많은지 부정단어가 많은지 확인하시오. (나중에 다시 확인)

1. From 절 서브쿼리
2. 조인
3. Regexp_substr

```
select count(*) 긍정단어
 from (
          select regexp_substr( lower( win_text), '[^ ]+', 1, a ) word
            from winter_kingdom, ( select level a
                                   from dual
                                   connect by level <= 15 )
       )
 where word in ( select lower( p_text )
                 from positive );
```

문제353. 전국 대학교 등록금 현황 데이터를 저장할 테이블을 먼저 아래와 같이 생성하시오.

카페 data 게시판 32번 글 전국 대학 insert 문

테이블 소개
Division
Type
University
Campus_type
Admission_fee : 입학금
College_fee
Supporting_fee
Tuition : 등록금

문제354. 우리나라에서 대학 등록금이 가장 비싼 학교는 어디인지 출력하시오.

```
Select university
       From univ
       Where tuition = (select max(tuition)
                        From univ);
```

문제355. 데이터 게시판에 번죄 발생 지역 데이터를 내려받아 테이블을 생성하고 데이터를 입력하시오.

카페 data 게시판 34번 글 범죄 발생 지역 데이터

테이블 소개
Crime_type : 범죄유형
C_loc : 범죄장소
Cnt : 번죄건수

문제356. 범죄유형을 출력하는데 중복제거해서 출력하시오.

```
Select distinct(crime_type)
    From crime_loc;
```

문제357. 살인이 가장 많이 일어나는 장소가 어디인지 출력하시오.

```
Select c_loc
    From crime_loc
    Where crime_type = '살인'
    Order by cnt desc
    Fetch first 1rows only;
```

문제358. 절도가 가장 많이 일어나는 장소가 어디인지 1위부터 3위까지 출력하시오.

```
Select c_loc
    From crime_loc
    Where crime_type = '절도'
    Order by cnt desc
    Fetch first 3 rows only;
```

문제359. 데이터 게시판에서 범죄원인 데이터를 가져와서 오라클에 테이블로 구성하고 데이터를 입력하시오.

카페 data 게시판 6번 글 범죄원인

문제360. 살인을 일으키는 가장 큰 원인을 출력하시오.

데이터 -> 컬럼 : pivot 문
컬럼 -> 데이터 : unpivot 문

문제360번을 출력하기 위해서는 컬럼을 데이터로 넣어야 가능하다.

카페 data 게시판 6번 글 확인

```
create  table  crime_cause2
as
select *
from crime_cause
unpivot ( cnt for term in (생계형,
유흥,
도박,
허영심,
복수,
해고,
징벌,
가정불화,
호기심,
유혹,
사고,
불만,
부주의,
기타));
```

문제361. 동전을 던져서 앞면이 나오는지 뒷면이 나오는지 확인하시오.

(앞면 : 0, 뒷면 : 1)

```
Select round (dbms_random.value (0, 1))
       From dual;
```

0 ~ 1 사이의 실수를 랜덤으로 출력하는데 가우시안 분포를 따르는 데이터로 출력이 된다.

문제362. 동전을 10번 던지시오.

```
Select round (dbms_random.value (0, 1))
       From dual
       Connect by level <= 10;
```

알고리즘 문제7)

문제363. 동전을 10만번 던졌을 때 동전이 앞면이 나올 확률은 얼마인가. (From 절의 서브쿼리를 사용하면 쉽게 할 수 있다.)

```
Select count (*)/100000
    From (select round (dbms_random.value (0, 1)) 확률
            From dual
            Connect by level <= 100000)
    Where 확률 = 0;

Undefine 던진횟수
Select count (*)/&&던진횟수
    From (select round (dbms_random.value (0, 1)) 확률
            From dual
            Connect by level <= &던진횟수)
    Where 확률 = 0;
```

현업에서 계층형 질의문을 활용했을때의 장점

- 자바, C, 기타 프로그래밍 언어로 길게 작성해야 하는 코드를 SQL하나로 끝낼 수 있다.

치환변수 (&)

SQL문장을 수행할 때 매번 검색해야하는 데이터 값이 다른데 SQL문장이 같을 때 검색을 용이하게 하는 오라클 SQL 문법

Ex)

```
Select empno, ename, sal
    From emp
    Where empno = &사원번호;
```

20.11.10

2020년 11월 10일 화요일 오전 9:51

1. Select 문의 6가지 절
    a. Select
    b. From
    c. Where
    d. Group by
    e. Having
    f. Order by
2. 함수
    a. 단일행 함수 : 문자, 숫자, 날짜, 변환, 일반
    b. 복수행 함수 : max, min, avg, sum, count
3. 조인문장
    a. 오라클 조인문장
        i. Equi join
        ii. Non equi join
        iii. Outer join
        iv. Self join
    b. 1999 ansi 조인문장
        i. On절을 사용한 조인
        ii. Using 절을 사용한 조인
        iii. Natural 조인
        iv. Left/right/full outer 조인
        v. Cross 조인
4. 서브쿼리
    a. Single row subquery
    b. Multiple row subquery
    c. Multiple column subquery
5. 집합 연산자
    a. Union all
    b. Union
    c. Intersect
    d. Minus
6. 계층형 질의문 : 서열을 보여주는 SQL문

- Select level, ename
  From emp
  Start with ename = 'KING'
  Connect by prior empno(부모키) = mgr(자식키);

7. DML문장
    a. Insert
    b. Update
    c. Delete
    d. Merge

8. DDL문장
    a. Create
    b. Alter
    c. Drop
    d. Truncate
    e. rename

문제364. 아래의 SQL에서 count(*)/100000 에서 100000을 하드코딩하지 않고 인라인 뷰에서 시행한 동전 던진 전체 횟수가 들어가게 하시오.

```
select count(*) / 100000 as "동전이 앞면이 나올 확률"
  from ( select round( dbms_random.value(0, 1) ) as 동전
           from dual
           connect by level <= 100000)
  where 동전 = 0;
```

```
Select sum(동전) / count(*) as "동전이 앞면이 나올 확률"
      From ( select round ( dbms_random.value(0,1) ) as 동전
               from dual
             Connect by level <= 100000);
```

구구단 1단 부터 9단 까지 출력하기 위해 알아야 할 SQL

Ex)
숫자 1과 2를 출력하는 SQL문을 작성하시오.

```
Select level
      From dual
      Connect by level <= 2;
```

Ex)
위의 결과를 from 절의 서브쿼리로 만들고 emp 테이블과 조인하시오.

```
Select empno, ename, sal
      From emp e, (select level
                        From dual
                        Connect by level <= 2);
```

Emp 테이블의 모든 데이터가 숫자 1과 조인해서 14개 출력하고, 숫자 2와 조인해서 14개를

출력해서 총 28개가 출력된다.

Ex)
숫자를 1부터 9까지 출력하는 SQL문장을 작성하시오.

```
Select level
        From dual
        Connect by level <= 9;
```

Ex)
위의 숫자를 1부터 9까지 출력하는 SQL문을 아래와 같이 FROM 절의 서브쿼리로 2개로 만드시오.

```
Select *
        From ( select level num1
                    From dual
                    Connect by level <= 9 ) a,
              ( select level num2
                    From dual
                    Connect by level <= 9) b;
```

알고리즘 문제8)

문제365. 구구단 전체를 출력하시오.

```
Select num1 || ' x ' || num2 || ' = ' || num1*num2
        From ( select level num1
                    From dual
                    Connect by level <= 9 ) a,
              ( select level num2
                    From dual
                    Connect by level <= 9) b
        Where num1 != 1;
```

임시 테이블 생성하기(CREATE TEMPORAY TABLE)

데이터를 영구히 database에 저장하는게 아니라 임시로 저장하는 테이블

데이터 중에서 영구히 저장할 필요는 없고 지금 잠깐 테스트를 위해서 볼 데이터라던가 지금 현재만 필요하고 나중에는 필요하지 않는 데이터가 있는데 그 데이터를 잠깐 저장할 때 사용하는 테이블이 임시 테이블이다.

임시테이블의 종류 2가지

1. On commit delete rows 옵션 : 데이터를 commit 할때까지만 보관
2. On commit preserve rows 옵션 : 데이터를 접속한 유저가 로그아웃할때까지만 보관

Ex)
```
Create global temporary table emp700
( empno number(10),
  ename varchar2(10),
  sal number(10) )
On commit delete rows;

Insert into emp700
        Select empno, ename, sal
                From emp;

Select * from emp700;

Commit;

Select * from emp700;
```

Ex)
```
Create global temporary table emp800
( empno number(10),
  ename varchar2(10),
  sal number(10) )
On commit preserve rows;

Insert into emp800
        Select empno, ename, sal
                From emp;

Select * from emp800;

Commit;

Select * from emp800;
```

문제366. 구구단 문제를 풀기 위한 from 절의 서브쿼리문의 결과를 임시테이블로 각각 생성하시오.

테이블 생성1
```
Create global temporary table num1_9
( num1 number(10) )
On commit preserve rows;

Insert into num1_9
Select level
        From dual
        Connect by level <= 9;

        Select * from num1_9;
```

테이블 생성2

```
Create global temporary table num2_9
( num2 number(10) )
On commit preserve rows;

Insert into num2_9
Select level
        From dual
        Connect by level <= 9;

        Select * from num2_9;


Select a.num1, b.num2
        From num1_9 a, num2_9 b;

Select a.num1 || ' x ' || b.num2 || ' = ' || a.num1*b.num2
        From num1_9 a, num2_9 b
        Where a.num1 != 1;
```

알고리즘 문제9)

문제367. 주사위를 10만번 던져서 자사위의 눈이 6이 나올 확률을 구하시오.

```
Select count(*)/100000 "6이나올 확률"
        From (select round (dbms_random.value (0.5, 6.5)) 주사위
                        From dual
                        Connect by level <= 100000)
        Where 주사위 = 6;
```

알고리즘 문제10)

문제368. 두개의 주사위를 던져서 두개의 주사위의 눈의 합이 10이 되는 확률을 구하시오.

```
Select count(a.num1 + b.num2)/36
        From ( select level num1
                        From dual
                        Connect by level <= 6) a,
                ( select level num2
                        From dual
                        Connect by level <= 6) b
        Where a.num1 + b.num2 = 10;
```

복잡한 쿼리를 단순하게 하기 1(VIEW)

테이블은 아니고 데이터를 바라보는 쿼리문을 테이블처럼 하나의 object(객체)로 생성한 것을 말한다.

Ex)
```
Create table emp708
    As
        Select empno, ename, sal, deptno
            From emp;

Select * from emp708;
```

As 다음에 나오는 쿼리문의 결과대로 emp708 테이블이 생선이된다.
Emp테이블과는 별개의 또 다른 테이블이다.
Emp로 시작하는 내가 만든 테이블이 어떤 것들이 있는지 확인하는 방법

```
Select table_name
    From user_tables
    Where table_name like 'EMP%';
```

EMP는 반드시 대문자로 작성해야 한다.

문제369. DALLAS에서 근무하는 사원들의 이름, 부서위치를 출력하시오.

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

문제370. 위의 결과데이터를 담는 테이블 emp710을 생성하시오.

```
Create table emp710
    As
        Select e.ename, d.loc
            From emp e, dept d
            Where e.deptno = d.deptno
            And d.loc = 'DALLAS';
```

문제371. 숫자 1번부터 10번까지의 숫자를 담는 테이블을 emp705로 생성하시오.

```
Create table emp705
    As
        Select level as num1
            From dual
```

```
Connect by level <= 10;
```

컬럼별칭을 반드시 줘야한다.
위 SQL문에서 level은 SQL문과 같은 예약어이기 때문에 테이블의 컬럼며으로 생성할 수 없다.

문제372. emp705의 숫자 데이터중에 임의로 아무거나 하나를 지우시오.

```
Delete from emp705
	Where num1 =4;
```

알고리즘 문제11)

문제373. 1부터 10 사이에 숫자 중 하나가 없는데 그 수는 무엇인지 찾으시오.

```
Select (a.num0 - b.num2)
	From (select sum(level) num0
		From dual
		Connect by level <= 10) a,
	    (select sum(num1) num2
		From emp705) b;
```

View 생성

```
Create view emp701
	As
		Select empno, ename, sal, deptno
			From emp;
```

View는 table과는 다르게 별도로 데이터를 저장하고 있지 않고 그냥 그 테이블을 바라보는 쿼리문이다.

```
Update emp701
	Set sal = 0
	Where ename = 'SCOTT';
```

View를 업데이트하면 실제 테이블도 업데이트 된다.

데이터중에 공개하면 안되는 민감한 데이터(ex.월급)를 제외하고 view로 생성한다.

```
Create view emp809
	As
```

```
Select empno, ename, job, hiredate, mgr, deptno
    From emp;
```

문제374. 직업, 직업별 토탈월급을 출력하시오.

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

문제375. 위의 결과를 출력하는 view를 생성하시오.(view 이름은 emp403)

```
Create view emp403
    As
        Select job, sum(sal) 토탈
            From emp
            Group by job;
```

View 생성할 때 그룹함수를 사용하게 되면 컬럼별칭을 줘야한다.

문제376. 부서번호, 부서번호별 평균월급을 출력하는 view를 deptno_avg라는 이름으로 생성하시오.

```
Create view deptno_avg
    As
        Select deptno, avg(sal) 평균월급
            From emp
            Group by deptno;
```

문제377. emp 와 위에서 만든 deptno_avg view 와 조인하여 이름, 월급, 부서번호, 부서평균을 출력하시오.

```
Select e.ename, e.sal, e.deptno, d.평균월급
    From emp e, deptno_avg d
    Where e.deptno = d.deptno;
```

뷰의 장점 :

1. 민감한 컬럼을 감춰서 데이터를 제공할 수 있다.
2. 복잡한 쿼리문을 단순하게 만들 수 있다.

문제378. 이름, 월급, 부서번호, 부서평균을 출력하는데 월급이 부서평균보다 더 큰 사원들만 출력하시오.

```
Select e.ename, e.sal, e.deptno, d.평균월급
        From emp e, deptno_avg d
        Where e.deptno = d.deptno
        And e.sal > d.평균월급;
```

문제379. emp 테이블에서 퇴사할 것 같은 사원들을 예측하기 위해 자기의 월급이 자기가 속한 직업의 편균월급보다 더 작은 사원들의 이름과 월급과 직업과 직업평균을 출력하시오.

```
Create view emp1122
        As
                Select deptno, avg(sal) 직업평균
                        From emp
                        Group by deptno;

Select e.ename, e.sal, e.job, d.직업평균
        From emp e, emp1122 d
        Where e.deptno = d.deptno
        And e.sal < d.직업평균;
```

문제380. 직업, 이름, 월급, 순위를 출력하는데 순위가 직업별로 각각 월급이 높은 순서대로 순위를 부여하시오.

```
Select job, ename, sal, dense_rank() over (partition by job order by sal desc)
        From emp;
```

문제381. 위의 쿼리의 결과를 view로 만들고 view를 쿼리해서 순위가 1등인 사원들만 출력하시오.

```
Create view emp1234
        As
                Select job, ename, sal, dense_rank() over (partition by job order by sal desc)
                        From emp;

Select *
        From emp1234
        Where 순위 = 1;
```

View의 종류 2가지

	단순 view	복합 view
테이블의 갯수	1개	2개 이상

그룹함수	포함 안됨	포함 됨
DML 여부	가능	불가능 할 수도 있다.

문제382. 이름, 부서위치를 출력하는 view를 ename_loc라는 이름으로 생성하시오. (복합 view)

```
Create view ename_loc
    As
        Select e. ename, d. loc
            From emp e, dept d
            Where e. deptno = d. deptno;
```

문제383. ename_loc의 데이터 중에 SCOTT의 부서위치를 WASHINGTON 으로 변경하시오.

```
Update ename_loc
    Set loc = 'WASHINGTON'
    Where ename = 'SCOTT';
```

변경 불가

SCOTT의 부서위치가 DALLAS에서 WS로 변경이 된다면 실제로 dept 테이블의 부서위치가 DALLAS에서 WS로 변경이 되어야 하기때문에 SCOTT이 아닌 다른 사원들도 DALLSAS 에서 WS으로 변경되어지므로 변경이 되지 않는다.

문제375번에서 만든 emp403 view 를 쿼리하고 결과를 보시오.

```
Create view emp403
    As
        Select job, sum(sal) 토탈
            From emp
            Group by job;

Select * from emp403;

Update emp403
    Set 토탈 = 2000
    Where job = 'SALESMAN';
```

위와 같이 복합view는 데이터를 갱신할 수 없는데 만약 토탈월급이 갱신된다면 실제값도 갱신해야 하기 때문에 갱신할 수도 없고 해서도 안된다.

문제384. 내가 그동 view들이 무엇이 있는지 확인 하시오.

```
Select view_name, text
    From user_views;
```

위와 같이 조회하면 view를 생성했을 때 사용한 테이블명도 볼 수 있다. 외부에서 온 데이터 분석가들은 위의 쿼리를 조회할 수 있는 권한을 제한하는 경우가 많다.

View 삭제

```
Drop view emp809;
```

데이터 검색 속도를 높이기(INDEX)

오라클 데이터베이스의 객체(object)의 종류 5가지

1. Table : 데이터를 저장하는 기본 저장소
2. View : 데이터를 바라보는 쿼리문
3. Index : 검색 속도를 높이기 위한 db object
4. Sequence : 순서대로 번호를 생성하는 db object
5. Synonym : 테이블명에 대한 또 다른 이름

SQL튜닝을 위해 반드시 알아야 하는 db object

- 검색속도를 높이는 기술

Index는 책으로 치면 책 앞에 나오는 목차이다.
테이블이 책이면 인덱스는 책의 목차이다.

Ex)
이름이 SCOTT인 사원의 이름과 월급을 출력하시오.

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

Ename에 인덱스가 없기 때문에 scott을 emp 테이블에서 찾을 때 처음부터 끝까지 full table scan을 한다.

Ex)
Emp 테이블에 ename에 인덱스를 생성하시오.

```
Create index emp_ename
    On emp(ename);
```

Create index 인덱스 이름
    On 테이블명 (컬럼명)

책으로 치면 앞에 목차가 만들어지고 목차를 먼저 검색하기 때문에 훨씬 빠르게 원하는 데이터를 검색 할 수 있다.

인덱스(목차)의 구조
- 소제목 + 페이지 번호
- 소제목이 ㄱ,ㄴ,ㄷ, ... 순으로 구성되어 있다.

위와 같이 ename에 인덱스(목차)를 만들면 ename을 이용해서 만든 인덱스의 구조는 컬럼명 + rowid로 구성되어 있고 컬럼명은 a,b,c, ... 순으로 정렬되어 인덱스를 구성한다.

Rowid : 그 행을 대표하는 주소

```
Select rowid, ename, sal
    From emp;
```

Emp_ename의 인덱스의 구조를 보는 쿼리문

```
Select ename, rowid
    From emp
    Where ename > ' ';
```

```
Select ename, rowid
    From emp;
```

위 2개의 SQL 비교

문제385. 사원 테이블에 월급에 인덱스를 거시오.

```
Create index emp_sal
    On emp(sal);
```

문제386. emp 테이블의 sal의 인덱스인 emp_sal 의 구조를 확인하시오.

```
Select sal, rowid
    From emp
    Where sal >= 0;
```

인덱스에서 데이터를 읽어왔기 때문에 order by 절을 사용하지 않아도 월급이 정렬되어서 출력된다.

알고리즘 문제12)

문제387. 두개의 주사위를 동시에 던져서 주사위의 눈의 합이 짝수가 되는 확률이 어떻게 되는지 구하시오.

1.

```
Select count(mod((a.num1+b.num2),2))/36
        From (select level num1
                        From dual
                        Connect by level <= 6) a,
                (select level num2
                        From dual
                        Connect by level <= 6) b
        Where mod((a.num1 + b.num2), 2) = 0;
```

2.

```
Select count(mod((a+b),2))/10000
        From (select round (dbms_random.value (0.5, 6.5)) a
                        From dual
                        Connect by level <= 100),
                (select round (dbms_random.value (0.5, 6.5)) b
                        From dual
                        Connect by level <= 100)
        Where mod((a+b),2) = 0;
```

20.11.11

2020년 11월 11일 수요일 오전 9:30

알고리즘 문제13

문제388. 주사위 하나와 동전 한개를 동시에 던져서 주사위의 눈은 5가 나오고 동전은 앞면이 나올 확률을 구하시오.

```
Select count(*)/10000
    From (select round(dbms_random.value (0, 1)) a,
                round(dbms_random.value (0.5, 6.5)) b
            From dual
            Connect by level <= 10000)
    Where a = 0 and b = 5;
```

인덱스의 구조 : 컬럼명 + rowid (row 물리적 주소)

- 컬럼명이 오름차순으로 정렬이 되어있다.

Ex)

```
Drop index emp_ename;

Create index emp_ename
    On emp(ename);

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

Emp 테이블의 ename에 인덱스가 있으므로 ename의 인덱스인 emp_ename 을 통해서 테이블의 데이터를 검색하게 된다. 만약 index가 없다면 full table scan을 한다.

실행계획을 확인

```
Explain plan for
    Select ename, sal
        from emp
        Where ename = 'SCOTT';

Select * from table(dbms_xplan.display);
```

실행계획을 제어하는 명령어

```
Explain plan for
        Select /*+ index(emp emp_ename)*/ ename, sal
                from emp
                Where ename = 'SCOTT';

Select * from table(dbms_xplan.display);
```

/*+ index(emp emp_ename) */
- Emp 테이블 emp_ename 에 인덱스를 통해서 데이터 검색을 할 수 있게 한다.

실행계획이 full table scan으로 나오면 SCOTT을 검색하기 위해서 EMP테이블을 처음부터 끝까지 모두 스캔했다는 뜻이다.

문제389. 사원테이블에 월급의 인덱스를 생성하고 사원 테이블을 검색 하는데 월급이 3000인 사원들의 이름과 월급을 검색하시오.

```
Create index emp_sal
        On emp(sal);

Select ename, sal
        From emp
        Where sal = 3000;
```

문제390. 위의 SQL의 실행계획을 확인 하시오.

F10을 눌러서 실행계획을 확인할 수 있다.

```
Explain plan for
        Select ename, sal
                From emp
                Where sal = 3000;
Select * from table(dbms_xplan.display);
```

문제391. 위의 SQL이 emp_sal 의 인덱스를 통해서 데이터를 검색할 수 있도록 힌트를 주시오.

```
Explain plan for
Select /*+ index(emp emp_sal) */ ename, sal
        From emp
        Where sal = 3000;
Select * from table(dbms_xplan.display);
```

이름이 BLAKE인 사원의 이름과 월급을 출력하시오.

```
Select ename, sal
        From emp
        Where ename = 'BLAKE';
```

위 SQL이 인덱스를 통해서 테이블을 조회하는 방법

인덱스	테이블
select ename, rowid From emp Where ename > ' ';	Select rowid, ename, sal From emp;

문제392. 입사일에 인덱스를 생성하고 81년 11월 17일에 입사한 사원의 이름과 입사일을 조회하시오.

```
Create index emp_hiredate
        On emp(hiredate);

Select ename, hiredate
        From emp
        Where hiredate = to_date('81/11/17', 'RR/MM/DD');
```

문제393. 위의 SQL문에 실행계획을 확인하고 full table scan 을 했다면 index scan을 할 수 있도록 튜닝하시오.

```
Explain plan for
        Select ename, hiredate
                From emp
                Where hiredate = to_date ('81/11/17', 'RR/MM/DD');

Select * from tabel(dbms_xplan.display);



Explain plan for
        Select /*+ index (emp emp_hiredate) */ ename, hiredate
                From emp
                Where hiredate = to_date ('81/11/17', 'RR/MM/DD');

Select * from table(dbms_xplan.display);
```

인덱스의 구조를 전부 읽어오는 방법

1. 문자 > ' '
2. 숫자 >= 0

3. 날짜 < to_date('9999/12/31', 'RRRR/MM/DD')

위의 조건이 where 절에 있어야 인덱스 전체를 다 읽을 수 있다.

인덱스의 구조를 알면 SQL 튜닝을 할 수가 있는데 그 방법중에 하나가 order by절을 사용하지 않고 정렬된 결과를 볼 수 있다. Order by 절을 남발하면 검색성능이 느려진다.

Order by 절 사용하지 않고 인덱스를 통해서 정렬하는 방법

Ex)

1. Order by 절을 사용 했을 때

```
Select ename, sal
        From emp
        Order by sal;
```

2. 인덱스를 사용 했을 때

```
Select ename, sal
        From emp
        Where sal >= 0;

Select /*+ index(emp emp_sal) */ ename, sal
        From emp
        Where sal >= 0;
```

인덱스를 통해 정렬을 하는데 정렬이 되지 않으면 힌트를 주면 확실하게 정렬된 결과를 볼 수 있다.

힌트 :

- /*+ index(테이블명 인덱스이름) */
- /*+ index_desc(테이블명 인덱스이름) */

문제394. 이름과 월급을 출하는데 월급이 높은 사원부터 출력하시오. (order by절 사용 x)

```
Select /*+ index_desc (emp emp_sal) */ ename, sal
        From emp
        Where sal >= 0;
```

Where 절에 인덱스 컬럼의 조건을 주는 것은 필수이다.
Where 절 없이 힌트만 사용해서는 안된다.

문제395. 아래의 SQL을 튜닝하시오.

튜닝 전

Select ename, hiredate
From emp
Order by hiredate desc;

```
Select /*+ index_desc (emp emp_hiredate) */ ename, hiredate
    From emp
    Where hiredate < to_date('9999/12/31', 'RRRR/MM/DD');
```

인덱스를 사용한 SQL을 튜닝하는 방법

Ex)
튜닝 전 :

```
Explain plan for
Select /*+ index(emp emp_sal) */ ename, sal
    From emp
    Where sal * 12 = 36000;

Select * from table(dbms_xplan.display);
```

Emp테이블에 emp_sal 인덱스를 엑세스 하게 실행하여도 불구하고 where 절에 인덱스 컬럼이 가공되었기 때문에 실행계획이 full table scan을 한다.

튜닝 후 :

```
Explain plan for
Select /*+ index(emp emp_sal) */ ename, sal
    From emp
    Where sal = 36000/12;

Select * from table(dbms_xplan.display);
```

문제396. 사원 테이블의 직업에 인덱스를 생성하시오.

```
Create index emp_job
    On emp(job);
```

문제397. 아래의 SQL을 튜닝하시오.

튜닝 전

Select ename, job
From emp

```
Where substr(job,1,5) = 'SALES';

Explain plan for
Select /*+ index (emp emp_job) */ ename, job
        From emp
        Where job like 'SALES%';

Select * from table (dbms_xplan.display);
```

좌변에 있는 인덱스 컬을 가공하면 full table scan 하게 되므로 좌변을 가공하면 안된다.

문제398. 아래의 SQL을 튜닝하시오.

튜닝 전

```
Select ename, hiredate
        From emp
        Where to_char (hiredate, 'RRRR') = '1981';

Explain plan for
Select ename, hiredate
        From emp
        Where hiredate between to_date('1981/1/1', 'RRRR/MM/DD')
                and to_date('1981/12/31', 'RRRR/MM/DD');

Select * from table(dbms_xplan.display);
```

문제399. 직업이 SALESMAN인 사원들 이름과 월급과 직업을 출력하는데 월급이 높은 사원부터 출력하시오.

```
Select /*+ index_desc(emp emp_sal) */ ename, sal, job
        From emp
        Where job = 'SALESMAN'
                And sal >= 0;
```

생성한 index목록을 확인하는 방법

```
Select index_name
        From user_indexes
        Where index_name like 'EMP%';
```

인덱스 삭제하는 방법

```
Drop index emp_ename;
```

문제400. 나머지 인덱스들도 전부 drop 하시오.

```
Drop index emp_hiredate;
Drop index emp_job;
Drop index emp_sal;
```

절대로 중복되지 않는 번호 만들기(SEQUENE)

번호를 중복되지않게 순서대로 성하는 번호 생성기

Ex)
```
Create sequence seq1
        Start with 1
        Maxvalue 100;

Select seq1.nextval
        From dual;
```

시퀀스를 이용하면 번호를 중복되지 않고 일관되게 테이블에 입력할 수 있다.
Ex) 주식 테이블 매매번호, 사원 테이블 사원번호

Ex)
시퀀스 seq2 라고 만드시오. (1번 부터 1000번까지)

```
Create sequence seq2
        Start with 1
        Maxvalue 1000;

Select seq2.nextval
        From dual;
```

Ex)
아래의 컬럼을 담는 테이블을 emp534라는 이름으로 생성하시오.

```
Create table emp534
( empno number(10),
  ename varchar2(10),
  sal number(10) );
```

Ex)
위에서 만든 seq2를 이용해서 emp534의 empno에 일관된 번호가 입력되게 하시오.

```
Insert into emp534
        Values(seq2.nextval, 'SCOTT', 3000);

Select * from emp534;
```

```
Insert into emp534
      Values(seq2.nextval,'KING', 5000);

Select * from emp534;
```

시퀀스의 현재값 확인하는 방법

```
Select seq2.currval
      From dual;
```

생성한 sequence목록을 확인하는 방법

```
Select sequence_name
      From user_sequences;
```

Sequence 삭제하는 방법

```
Drop sequence seq2;
```

실수로 지운 데이터 복구하기 1(FLASHBACK QUERY)

타임머신 기능을 이용하여 과거의 데이터를 확인할 수 있고 테이블을 과거로 되돌릴 수 있다.

Flashback query는 과거의 데이터를 확인하는 기능이다.

Ex)

```
Delete from emp;

Commit;

Select *
      From emp as of timestamp (systimestamp - interval '5' minute);
```

5분 전의 emp 테이블의 상태를 확인할 수 있다.

```
Create table emp_backup_20201111
As
Select *
      From emp as of timestamp (systimestamp - interval '5' minute);

Select * from emp_backup_20201111;
```

문제401. 백업받은 emp_backup_20201111의 데이터를 emp테이블에 입력하시오.

```
Insert into emp
	Select *
		From emp_backup_20201111;
```

문제402. 아래와 같이 salgrade테이블 전부 delete 하고 commit 한 후에 복구하시오.

```
Delete from salgrade;

Commit;

Select *
	From salgrade as of timestamp (systimestamp - interval '1'minute);

Create table salgrade_backup
as
Select *
	From salgrade as of timestamp (systimestamp - interval '1'minute);

Insert into salgrade
Select *
	From salgrade_backup;

Commit;
```

실수로 지운 데이터 복구하기 2(FLASHBACK TABLE)

ex)

```
Delete from dept;

Commit;
```

1. Dept 테이블이 flashback 될 수 있도록 설정한다.

```
Alter table dept enable row movement;
```

2. Dept 테이블을 과거로 되돌린다.

```
Flashback table dept to timestamp
(systimestamp - interval '5' minute);
```

3. Dept 테이블 조회한다.

```
Select * from dept;
```

과거로 되돌리지 못하는 경우

1. Sys 유저에서 작업했을 경우
2. Delete 하고 commit 한 이후에 DDL 명령어를 수행한 경우

골든 타임 확인하는 방법
Show parameter undo_retention

Flashback table 령어를 수해서 과거로 돌렸으면 반드시 commit 을 해야 한다.

실수로 지운 데이터 복구하기 3(FLASHBACK DROP)

테이블을 drop 했을 경우에도 복구할 수 있는데 10g 버전 이후부터는 drop 한 ᅦ이블이 휴지통에 입력이 되기 때문에 휴지통에서 복구할 수 있다.

Drop table dept;

Show recyclebin;

Select *
From user_recyclebin;

Flashback table dept to before drop;

휴지통을 비우지 않는다면 계속해서 휴지통에 테이블이 있기 때문에 복구 할 수 있다.

Drop table dept;
Drop table salgrade;
Drop table emp;

Select *
From user_recyclebin;

휴지통 비우기 명령어

Purge recyclebin;

Select *
From user_recyclebin;

테이블 삭제할 때 휴지통에 넣지 않고 삭제하는 방법

Drop table emp800 purge;

실수로 지운 데이터 복구하기 4(FLASHBACK VERSION QUERY)

특정 테이블이 과거로부터 현재까지 어게 변경이 되어왔는지 그 이력정보를 확인하고자 할 때 사용하는 쿼리문

Ex)

1. 현재시간을 확인하시오.

```
Select systimestamp
    from dual;
```

20/11/11 15:49:09.258000000 +09:00

2. KING 의 이름과 월급과 부서번호를 조회하시오.

```
Select ename, sal, deptno
    From emp
    Where ename = 'KING';
```

3. KING의 월급을 8000으로 변경하고 commit 하시오.

```
Update emp
    Set sal = 8000
    Where ename = 'KING';

Commit;
```

4. KING의 부서번호를 20번으로 변경하고 commit 하시오.

```
Update emp
    Set deptno = 20
    Where ename = 'KING';

Commit;
```

5. 그동안 KING의 데이터가 어떻게 변경되어 왔는지 그 이력정보를 확인하시오.

```
Select ename, sal, deptno, versions_starttime, versions_endtime,
        Versions_operation
    From emp
    Versions between timestamp
        To_timestamp('20/11/11 15:49:09', 'RR/MM/DD HH24/MI/SS')
        And maxvalue
    Where ename = 'KING'
    Order by versions_endtime;
```

6. 위의 시간중에 하나를 확인해서 그 시간대에 emp 테이블의 상태를 확인하시오.

```
Select *
    From emp as of timestamp
    To_timestamp('20/11/11 15:52:56', 'RR/MM/DD HH24/MI/SS');
```

7. 20/11/11 15:52:56 시간으로 emp 테이블을 복구하시오.

Alter table emp enable row movement;

Flashback table emp to timestamp
To_timestamp ('20/11/11 15:52:56', 'RR/MM/DD HH24:MI:SS');

Select * from emp;

Commit;

일단 과거로 가면 현재로 돌아올 수 없다. 그래서 시간을 정확하게 확인해서 신중하게 되돌려야 한다.

실수로 지운 데이터 복구하기 5(FLASHBACK TRANSACTION QUERY)

Dba의 영역에 가깝기 때문에 따로 실습 x

그동안 작업했던 DML 문장을 확인해 볼 수 있다.
어떤 update 문장을 수행했고 어떤 delete 문장을 수행했고 어떤 insert 문장을 수행했는지 그 문장들을 반대로 수행하는 DML문장이 출력된다.

데이터의 품질 높이기 1(PRIMARY KEY)

데이터 분석을 하다보면 제일 많은 시간을 할애하는 작업이 데이터 전처리 이다.
품질이 높은 데이터를 처음부터 입력받게 강화 하면 데이터 전처리에 많은 시간을 들이지 않아도 된다.

그래서 데이터 품질을 높이기 위한 한 방법으로 제약을 사용한다.
처음부터 테이블에 데이터를 입력받을 때 부터 엄격한 기준으로 데이터를 입력받게 하려면 제약을 이용하면 된다.

제약의 종류
1. Primary key : 중복된 데이터와 null 값을 허용하지 않게 하는 제약
2. Uniqe : 중복된 데이터를 허용하지 않게 하는 제약
3. Not null : null 값을 허용하지 않게 하는 제약
4. Check : 특정 데이터 외에 다른 데이터는 입력되지 못하게 하는 제약
5. Foreign key : 참조하는 컬럼에 거는 제약

Primary key 제약을 생성해서 테이블 생성하기

Create table emp307
(empno number(10) primary key,

ename varchar2(20));

위와 같이 테이블을 생성하면 empno에는 중복된 데이터와 null값이 입력되지 않는다.

Insert into emp307 values (1111, 'SCOTT');
Insert into emp307 values (2222, 'SMITH');
Insert into emp307 values (1111, 'ALLEN');
- 중복된 data 입력불가

Insert into emp307 values (null, 'JONES');
- Null 값 입력불가

알고리즘 문제14)

문제403. factorial을 아래와 같이 출력되도록 SQL로 구현하시오. (치환변수를 사용하시오.)

5팩토리얼은 120 입니다.

```
Undefine 숫자
Select &&숫자 || ' 팩토리얼은 ' || 결과값 || ' 입니다.'
        From (select exp(sum(ln(level))) 결과값
                        From dual
                        Connect by level <= &숫자);

Undefine 숫자
Select &&숫자 || ' 팩토리얼은 ' || exp(sum(ln(level))) || ' 입니다.'
        From dual
        Connect by level <= &숫자
```

20.11.12

2020년 11월 12일 목요일 오전 9:33

데이터의 품질 높이기 2(UNIQUE)

데이터 품질을 높이기 위해서 중복 된 데이터가 테이블에 입력되지 못하게하는 제약

Ex)
Create table emp507
(empno number(10),
 ename varchar2(10) unique);

Ename 에 중복된 데이터 입력 불가

Insert into emp507
Values (1111, 'scott');

Insert into emp507
Values (2222, 'scott');

제약 삭제

1. 삭제할 제약 이름을 확인한다.

Select table_name, constraint_name
From user_constraints
Where table_name = 'EMP507';

2. EMP507의 unique 제약을 삭제한다.

Alter table emp507
Drop constraint SYS_C007459;

3. 중복된 데이터가 입력이 되는지 확인하시오.

Insert into emp507
Values (2222, 'scott');

- 위와 같이 제약이름이 SYS_C007459 와 같이 나와서 이름만 봐서는 제약이 어떤 제약인지 확인하기가 어려우니 제약을 처음 만들 때 부터 이름을 의미있게 부여해서 만들어주면 관리가 쉽다.

4. 제약 이름을 주고 테이블을 생성하는 방법

```
Create table emp508
( empno number(10),
  ename varchar2(10) constraint emp508_ename_un unique);
```

- emp508_ename_un (테이블명_컬럼명_제약축약)

5. Emp508 테이블의 ename에 걸린 unique 제약을 확인한다.

```
Select table_name, constraint_name
        From user_constraints
        Where table_name = 'EMP508';
```

문제404. emp508 테이블에 ename에 걸린 unique 제약을 삭제하시오.

```
Alter table emp508
        Drop constraint emp508_ename_un;
```

제약을 생성하는 방법 2가지

1. 테이블 생성할 때
2. 이미 만들어져 있는 테이블에 제약을 추가

이미 만들어 놓은 테이블에 제약을 추가하는 방법

1. Emp 이블에 ename에 unique 제약을 거시오.

```
Alter table emp
        Add constraint emp_ename_un unique(ename);
```

기존에 테이블에 중복된 데이터가 있다면 위의 명령어는 실행되지 않는다.

Database에서 제약을 사용해야 하는 이유

- 데이터의 품질을 높이기 위해서

문제405. emp12테이블의 이름에 unique 제약을 거시오.

```
Alter table emp12
        Add constraint emp12_ename_un unique(ename);
```

데이터의 품질 높이기 3(NOT NULL)

Null값을 입력하지 못하게 막는 제약

Ex)

1. 테이블을 생성할 때 제약거는 방법

```
Create table emp707
(empno number(10) constraint emp707_empno_nn not null,
 ename varchar2(10));

Insert into emp707
Values (1111,'smith');

Insert into emp707
Values (null,'scott');
```

2. 만들어진 테이블에 제약거는 방법

```
Alter table emp
        Modify ename constraint emp_ename_nn not null;
```

문제406. 사원테이블에 월급에 not null제약을 거시오.

```
Alter table emp
        Modify sal constraint emp_sal_nn not null;
```

문제407. 사원 테이블의 월급에 걸린 not null제약을 삭제하시오.

```
Alter table emp
        Drop constraint emp_sal_nn;
```

데이터의 품질 높이기 4(CHECK)

지정된 데이터만 입력되고 다른 데이터는 입력되지 못하게 막는 제약

Ex)
성별 : 남자, 여자

1. 테이블을 생성할 때 제약을 거는 방법

```
Create table emp745
(ename varchar2(10),
 loc varchar2(10) constraint emp745_loc_ck
        Check (loc in('DALLAS', 'CHICAGO', 'BOSTON')));
```

Loc에는 DALLAS와 CHICAGO, BOSTON 외에는 데이터가 입력되거나 수정되지 않는다.

2. 만들어진 테이블에 제약을 거는 방법

```
Alter table dept
        Add constraint dept_loc_ck
                Check (loc in ('NEW YORK', 'DALLAS', 'CHICAGO', 'BOSTON'));

Insert into dept(deptno, loc, dname)
        Values(50, 'SEOUL', 'RESEARCH');
```

문제408. 사원 테이블의 월급에 월급이 0 ~ 9500사이의 데이터만 입력되도록 check제약을 거시오.

```
Alter table emp
        Add constraint emp_sal_ck
                Check (sal between 0 and 9500);
```

문제409. emp12 테이블의 성별 컬럼에 남, 여만 입력 되도록 check 제약을 거시오.

```
Alter table emp12
        Add constraint emp12_gender_ck
                Check (gender in ('남', '여'));
```

데이터의 품질 높이기 5(FOREIGN KEY)

참조하는 컬럼에 거는 제약

부모테이블(Dept 테이블)에 primary key제약을 걸고 자식테이블(Emp 테이블)에 foreign key 제약을 걸면서 emp 테이블의 dept가 dept테이블의 deptno를 참조 한다.

Emp테이블에 deptno에 부서번호를 입력할 때 dept 테이블에있는 deptno인 10, 20, 30, 40 의 데이터만 입력할 수 있다. 그리고 dept 테이블의 deptno의 데이터를 지우려고 하면 emp테이블의 deptno가 참조하고 있으므로 지워지지 않는다.

Foreign key를 사용하는 이유

1. 데이터 품질을 높이기 위해서
2. 조인할 때 outer join을 남발하지 앟게 하기 위해서

Ex)
Emp와 dept테이블 생성스크립트를 수행한다.

1. Dept 테이블의 deptno에 primary key 제약을 건다.

```
Alter table dept
        Add constraint dept_deptno_pk primary key(deptno);
```

2. Emp 테이블의 deptno에 foreign key 제약을 걸면서 dept 테이블의 deptno를 참조한다.

```
Alter table emp
        Add constraint emp_deptno_fk foreign key(deptno)
        References dept(deptno);

Insert into emp(empno, ename, sal, deptno)
        Values (1234, 'JACK', 4500, 70);
```

WITH절 사용하기 1(WITH ~ AS)

복잡한 쿼리내에 동일한 쿼리가 두번 이상 발생한 경우 사용하면 좋은 성능을 보이는 SQL

테스트 테이블과 같이 임시로 사용하는 데이터를 가지고 테스트 할 때 유용한 SQL

Ex)
1부터 10 까지의 숫자를 출력하는 SQL을 작성하시오.

```
Select level num
        From dual
        Connect by level <= 10;
```

Ex)
위의 SQL을 with 절로 변경하시오.
With절로 내가 사용하는 SQL내에서만 임시로 테이블을 만든다.

```
With test_table as ( select level num
                        From dual
                        Connect by level <= 10 )
        Select num
                From test_table;
```

As 다음에 나오는 괄호 안의 쿼리문의 결과 데이터로 임시테이블을 생성하는데 그 테이블 이름이 test_table 이 된다. 그리고 그 아래의 쿼리에서 test_table 을 가지고 다양하게 쿼리문장을 만들어서 테스트 할 수 있다. 이 임시 테이블은 with 절이 끝나면 사라진다.

문제410. 위의 결과에서 짝수만 출력하시오.

```
With test_table as ( select level num
                        From dual
                        Where mod(level,2) = 0
                        Connect by level <= 10 )
        Select num
                From test_table;

With test_table as ( select level num
```

```
                    From dual
                    Connect by level <= 10 )
    Select num
        From test_table
        Where mod(num,2) = 0;
```

문제411. with절을 사용하여 구구단 2단을 출력하시오.

```
With test_table as ( select level num
                    From dual
                    Connect by level <= 9 )
    Select '2 x ' || num || ' = ' || num*2 구구단2단
        From test_table;
```

As다음에 나오는 괄호 안의 쿼리문의 결과가 하드 디스크에 저장이 되고 테이블명이 test_table이 된다.

문제412. with 절을 사용하여 구구단 전체를 출력하시오.

```
Select num1 || ' x ' || num2 || ' = ' || num1*num2
From ( select level num1
        From dual
        Connect by level <= 9 ) a,
        ( select level num2
        From dual
        Connect by level <= 9) b
Where num1 != 1;
```

```
With temp_table1 as (select level num1
                        From dual
                        Connect by level <= 9 ),
    Temp_table2 as (select level num2
                        From dual
                        Connect by level <= 9)
    Select num1 || ' x ' || num2 || ' = ' || num1*num2 구구단
    From temp_table1 a, temp_table2 b
    Where num1 != 1;
```

문제413. 아래의 인라인뷰 SQL을 with 절로 변경하시오.

```
Select e.ename, e.sal, e.deptno, v.부서평균
    From emp e, (select deptno, avg(sal) 부서평균
                        From emp
                        Group by deptno) v
    Where e.deptno = v.deptno
    And e.sal > v.부서평균;
```

```
With test_table as (select deptno, avg(sal) 부서평균
                    From emp
                    Group by deptno) v
    Select e.ename, e.sal, e.deptno, v.부서평균
    From emp e, test_table v
    Where e.deptno = v.deptno
    And e.sal > v.부서평균;
```

알고리즘 문제15)

문제414. with 절을 이용해서 직각 삼각형을 출력하시오.

```
With temp_table as (select level num1
                    From dual
                    Connect by level <= 9)
    Select lpad ('@', num1, '@')
        From temp_table;
```

문제415. 숫자를 물어보게하고 숫자를 입력하면 해당 숫자만큼 직각 삼각형이 만들어지게 하시오.

```
Undefine 숫자
With temp_table as (select level num1
                    From dual
                    Connect by level <= &숫자)
    Select lpad ('@', num1, '@')
        From temp_table;
```

알고리즘 문제16)

문제416. 이번에는 삼각형이 출력되게 하시오.

```
Undefine 숫자
With temp_table as (select level num1
                    From dual
                    Connect by level <= &&숫자)
    Select lpad(' ',(&숫자-num1) , ' ') || lpad ('★', num1, '★')
        From temp_table;
```

With 절의 장점

- 똑같은 SQL을 하나의 SQL내에서 여러 개 사용될 때 유용하다.

문제417. 직업, 직업별 토탈월급을 출력하시오.

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

문제418. 위의 집업별 토탈월급들 중에서의 평균값을 출력하시오.

```
Select avg(sum(sal))
        From emp
        Group by job;
```

문제419. 문제417과 문제418 을 이용하여 직업, 직업별 토탈월급을 출력하는데 직업별 토탈월급이 직업별 토탈 월급들의 평균값보다 더 큰 것만 출력하시오.

```
Select job, sum(sal)
        From emp
        Group by job
        having sum(sal) > (select avg(sum(sal))
                                From emp
                                Group by job);
```

문제420. 위의 SQL을 with 절로 구현하시오.

```
With job_sumsal as ( select job, sum(sal) 토탈
                                From emp
                                Group by job )
        Select job, 토탈
                From job_sumsal
                Where 토탈 > ( select avg(토탈)
                                From job_sumsal );
```

복잡한 쿼리내에 동일 쿼리블럭이 두번 이상 발생하는 경우에 사용하면 좋은 성능을 보이는 SQL이 with 절이다.

문제421. 위의 SQL의 실행계획을 확인해서 하드디스크에 temp테이블로 만들어지는지 확인하시오.

```
Explain plan for

With job_sumsal as ( select job, sum(sal) 토탈
                                From emp
                                Group by job )
        Select job, 토탈
```

```
        From job_sumsal
        Where 토탈 > ( select avg(토탈)
                        From job_sumsal );

Select * from table(dbms_xplan.display);
```

Accept 절 사용 방법

```
Select empno, ename, sal
    From emp
    Where empno = &사원번호;
```

입력 메시지 창에서 메세지를 변경하고 싶다면 accept를 사용하면 된다.

```
Accept p_num1 prompt '사원번호를 입력하시오.'
Select empno, ename, sal
    From emp
    Where empno = &p_num1;
```

문제422. 415문제를 다시 푸는데 아래와 같이 입력메세지 창에서 나오는 메세지가 '숫자를 입력하시오' 라고 나오게 하시오.

```
Accept p_num1 prompt '숫자를 입력하시오.'
With temp_table as (select level num1
                    From dual
                    Connect by level <= &p_num1)
    Select lpad ('@', num1, '@')
        From temp_table;
```

알고리즘 문제17)

문제423. 아래와 같이 숫자를 물어보게 하고 숫자를 입력하면 마름모가 출력되게하시오.

```
Accept p_num1 prompt '숫자를 입력하시오.'
With temp_table as (select level num1
                    From dual
                    Connect by level <= &&숫자)
    Select lpad(' ',(&숫자-num1) , ' ') || lpad ('★', num1, '★')
        From temp_table
Union all
    Select lpad(' ',num1 , ' ') || lpad ('★', (&숫자-num1)+1, '★')
        From temp_table;
```

WITH절 사용하기 2(SUBQUERY FACTORING)

With 절을 사용할 때 with 절 내에서 같은 SQL이 아래와 같이 두번 이상 사용되게 되면 temptable 에 자동으로 저장되고 한번만 사용하면 with 절의 as 다음에 나오는 쿼리문의 결과가 메모리에 저장된다.

With 절을 사용할 때 쓰는 유용한 힌트 2가지

1. /*+ materialize */ : temp table 을 하드 디스크에 생성
2. /*+ inline */ : temp table 을 만들지 말고 from 절의 서브쿼리인 in line view 에 작성

```
Explain plan for

With job_sumsal as ( select /*+ materialize */ job, sum(sal) 토탈
                              From emp
                              Group by job )
      Select job, 토탈
            From job_sumsal

Select * from table(dbms_xplan.display);
```

SQL로 알고리즘 문제 풀기 6(사각형 출력)

Ex)
★ 을 아래와 같이 5개를 출력하시오.

★★★★★

```
Select lpad('★', 5, '★')
      From dual;
```

Ex)
위의 ★ 5개가 아래로 6개가 나오게 하시오.

★★★★★
★★★★★
★★★★★
★★★★★
★★★★★
★★★★★

```
Select lpad('★', 5, '★')
      From dual
      Connect by level <= 6;
```

알고리즘 문제18)

문제424. 아래와 같이 두 숫자를 각각 입력받아 사각형을 출력하시오.

가로의 숫자를 입력하시오.
세로의 숫자를 입력하시오.

```
Accept p_num1 prompt '가로의 숫자를 입력하시오.'
Accept p_num2 prompt '세로의 숫자를 입력하시오.'
With temp_table as (Select level num1
                    From dual
                    Connect by level <= &p_num2)
    Select lpad ('★', &p_num1, '★')
    From temp_table;
```

SQL로 알고리즘 문제 풀기 11(최대 공약수)

약분을 빨리 하기 위해서 최대 공약수가 필요하다 .

16명의 인부들에게 24개의 빵을 공평하게 나눠주려면 1명당 몇개씩 주면 되는가

알고리즘 문제19)

문제425. SQL로 최대공약수를 구하시오.

첫번째 숫자를 입력하시오.
두번째 숫자를 입력하시오.

```
Accept p_num1 prompt '첫번째 숫자를 입력하시오.'
Accept p_num2 prompt '두번째 숫자를 입력하시오.'
Select max(공약수)
    From (select level 공약수
          From dual
          Where mod(&p_num1, level) = 0
          And mod(&p_num2, level) = 0
          Connect by level <= &p_num1);
```

20.11.13

2020년 11월 13일 금요일 오전 9:46

복습

1. 기본 SQL문장
2. 함수
3. 조인문
4. 집합연산자
5. 서브쿼리문
6. DML문
7. DDL문
8. TCL문
9. 계층형 질의문
10. With절
11. 제약

기타 SQL기능들과 수업때 다루지 않았던 주요 SQL기능들

3. 조인문
    a. 오라클 조인문법
        i. Equi join
        ii. Non dqui join
        iii. Outer join
        iv. Self join
    b. 1999ANSI 문법
        i. On 절을 사용한 조인
        ii. Using 절을 사용한 조인
        iii. Natural 조인
        iv. Left/right/full outer 조인
        v. Cross 조인

3개 이상의 테이블 조인

- 2개의 테이블 조인 : emp, dept 의 연결고리 1개 필요
- 3개의 테이블 조인 : dept, emp, salgrade의 연결고리 2개 필요

Where e.deptno = d.deptno and e.sal between s.losal and s.hisal

문제426. 이름, 부서위치 ,월급, 부서번호를 출력하시오.

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

문제427. emp와 dept와 salgrade 테이블을 조인하여 이름, 월급, 부서위치, 급여등급을 출력하시오.

```
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;
```

테이블이 3개 이면 2개의 연결고리를 where 절에 기술해줘야 한다.

문제428. 아래와 같이 bonus라는 테이블을 생성하시오.

```
Create table bonus
As
        Select empno, sal * 1.5 as comm2
                From emp;
```

문제429. 사원이름, 월급, 부서위치, comm2를 출력하시오.

```
Select e.ename, e.sal, d.loc, b.comm2
        From emp e, dept d, bonus b
        Where e.deptno = d.deptno
                And e.empno = b. empno;
```

문제430. 위의 결과에서 월급이 2000 이상인 사원들만 출력하시오.

```
Select e.ename, e.sal, d.loc, b.comm2
        From emp e, dept d, bonus b
        Where e.deptno = d.deptno
                And e.empno = b. empno
                And e.sal >= 2000;
```

SQL로 알고리즘 문제 풀기 13(피타고라스의 정리)

알고리즘 문제20)

문제431. 피타고라스의 정리를 구현하시오.

밑변을 입력하시오. 3

높이를 입력하시오. 4
빗변을 입력하시오. 5

직각 삼각형이 맞습니다. / 직각 삼각형이 아닙니다.

```
Accept p_num1 prompt '밑변을 입력하시오.'
Accept p_num2 prompt '높이를 입력하시오.'
Accept p_num3 prompt '빗변을 입력하시오.'
Select Case when power(&p_num1,2) + power(&p_num2,2) = power(&p_num3,2)
            Then '직각 삼각형이 맞습니다.'
            Else '직각 삼각형이 아닙니다.' end
      From dual;
```

함수

1. 단일행 함수
    a. 문자
    b. 숫자
        i. Round
        ii. Trunc
        iii. Mod
        iv. Power : select power(2,3)
                    From dual; -> 2의 3승
    c. 날짜
    d. 변환
    e. 일반
2. 복수행 함수

외부 테이블 (EXTERNAL)

오라클의 테이블의 종류

1. 일반 테이블 (heap table)
2. 임시 테이블 (temp table)
3. 외부 테이블 (external table)
4. 파티션 테이블 (partition table)

병렬처리 프로그래밍

- SQL 파티션 테이블을 생성하여 병렬 쿼리를 이용

외부 테이블 - database 외부에 있는 엑셀 파일이나 csv 파일, text 파일을 insert 문으로 만들어서 데이터베이스에 입력하지 않고 링크만 걸어서 해당 파일의 데이터를 select 할 수 있는 테이블

Ex)

1. Emp.txt를 c 드라이브 밑에 data라는 폴더를 만들고 그안에 emp.txt를 넣으시오.

2. C드라이브에 data 폴더를 오라클 디렉토리로 생성하시오.

```
Create directory emp_dir as 'c:₩data';
```

3. 외부 테이블을 생성하시오.

```
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 - external table 생성
  (type oracle_loader - data를 로드하는 엔진을 sql*loader을 사용
   default directory emp_dir - emp_dir 을 기본 디렉토리로 한다
   access parameters - text 파일의 내용을 테이블에 입력하기 위한 문법을 시작
   (records delimited by newline - 행과 행 사이는 엔터로 구분
    fields terminated by "," - 데이터와 데이터는 콤마(,) 로 구분
   (empno char, - 외부테이블의 컬럼들의 실제 데이터는 문자 (오라클에는 숫자)
    ename char,
    job char,
    mgr char,
    hiredate date "yyyy/mm/dd",
    sal char,
    comm char ,
    deptno  char  ) )
    location ('emp.txt') ); - 링크 걸 텍스트 파일 이름

    Select * from ext_emp11;
```

- 대용량 엑셀 파일이나 text 파일을 db에 넣으려면 시간이 많이 걸린다.

외부 테이블의 장점

1. Database의 저장공간이 필요없다.
2. 대용량 데이터인 경우 테이블에 입력하는 시간을 줄일 수 있다.

문제432. ext_emp11 테이블을 조회하여 직업이 SALESMAN인 사원들의 이름과 직업과 월급을 출력하시오.

```
Select ename, job, sal
      From ext_emp11
```

```
Where job = 'SALESMAN';
```

문제433. ext_emp11과 dept를 조인하여 이름과 부서위치를 출력하시오.

```
Select e.ename, d.loc
    From ext_emp11 e, dept d
    Where e.deptno = d.deptno;
```

External table의 단점

1. DML작업이 불가능 하고 SELECT만 가능하다.
2. 인덱스를 생성할 수 없다.

문제434. ext_emp11의 scott의 월급을 6000으로 갱신하시오.

```
Update ext_emp11
    Set sal = 6000
    Where ename = 'SCOTT';
```

문제435. dept.txt를 외부테이블로 생성해서 아래와 같이 select 할 수 있도록 하시오.

Select * from ext_dept;

1. Dept.txt 를 카페에서 내려받고 c 드라이브 밑에 data 폴더 밑에 넣는다.

2. Ext_emp11 테이블 스크립트를 가져와서 exp_dept 테이블을 만들 수 있도록 수정하시오.

```
create table ext_dept
    ( EMPNO   NUMBER(10),
     DNAME   VARCHAR2(14),
     LOC      VARCHAR2(13))
     organization external
    (type oracle_loader
     default directory emp_dir
     access parameters
     (records delimited by newline
      fields terminated by ","
     (empno char,
      dname char,
      loc  char  ) )
      location ('dept.txt') );
```

```
Select * from ext_dept;
```

외부 테이블 확인 및 삭제 방법

```
Select table_name, external
        From user_tables
        Where external = 'YES';

Drop table EXT_EMP11;
Drop table ext_dept;
```

정규식 함수 5가지

데이터 검색을 좀 더 상세하게 하려 할 때 기존의 함수로는 표현 할 수 없는 검색을 가능하게 해주는 함수

Regular expression (정규 표현식)

정규 표현식 코드는 오라클 뿐만 아니라 다른 언어에서도 공통적으로 사용하는 표현식 코드이다.

오라클 10.1 버전부터 정규표현식을 지원한다.

오라클 데이터 베이스는 정규표현식의 posix 연산자를 지원한다.

Posix 는 Portable Operation System Interface의 양자로 시스템간 호환성을 위해 미리 정의된 인터페이스를 의미한다.

Posix 연산자에는 기본 연산자, 앵커, 수량사, 서브표현식, 역참조, 문자리스트, posix문자 클래스 등이 있다.

1. 기본 연산자

연산자	영문	설명
.	Dot	모든 문자와 일치
|	Or	대체 문자를 구분
₩	Back slash	다음 문자를 일반 문자로 처리

Ex)

```
Select regexp_substr('aab', 'a.b') as c1,
                Regexp_substr('abb', 'a.b') as c2,
                Regexp_substr('acb', 'a.b') as c3,
                Regexp_substr('adc', 'a.b') as c4
```

```
From dual;
```

문제 436. 이름에 en 또는 in을 포함하고 있는 사원들의 이름과 월급을 출력하시오.

```
Select ename, sal
    From emp
    Where ename like '%EN%'
        Or ename like '%IN%';
```

문제437. 위의 SQL을 정규 표현식 함수인 regexp_like를 이용해서 출력하시오.

```
Select ename, sal
    From emp
    Where regexp_like (ename, 'EN|IN');
```

정규 표현식 함수

1. Regexp_substr
2. Regexp_like
3. Regexp_count
4. Regexp_instr
5. Regexp_replace

문제438. 우리반 테이블에서 전공이 통계, 수학, 컴퓨터, 전자가 포함된 학생들의 이름과 전공을 출력하시오.

```
Select ename, major
    From emp12
    Where major like '%통계%'
        Or major like '%수학%'
        Or major like '%컴퓨터%'
        Or major like '%전자%';

Select ename, major
    From emp12
    Where regexp_like (major, '통계|수학|컴퓨터|전자');
```

문제439. employees.csv 파일을 오라클의 테이블로 생성하시오.

```
Create table employees
(EMPLOYEE_ID        number(10),
 FIRST_NAME         varchar2(20),
```

```
LAST_NAME           varchar2(20),
EMAIL               varchar2(20),
PHONE_NUMBER        varchar2(30),
HIRE_DATE           date,
JOB_ID              varchar2(20),
SALARY              number(10,2),
COMMISSION_PCT      number(10,2),
MANAGER_ID          number(10),
DEPARTMENT_ID       number(10) );
```

Employees.csv 파일을 sqldeveloper 을 이용해서 employees 테이블에 로드한다.

데이터 임포트

```
Select count(*) from employees;
```

문제440. 이름의 첫 글자가 St로 시작하면서 끝 글자가 en으로 끝나는 사원들의 first_name을 출력하시오.

```
Select first_name
      From employees
      Where substr (first_name,1,2) = 'St'
      And first_name like '%en';
```

```
Select first_name
      From employees
      Where regexp_like (first_name, '^St(.)+en$');
```

'^St(.)+en$'

- ^ 는 시작 $ 는 끝을 나타낸다.
  시작이 St로 시작하면서 끝이 en으로 끝나는 first_name을 찾는다.
- (.)+ 는 점(.)은 한자리를 나타내는데 +가 여러개를 나타내므로 한자리가 여러 개인것을 표현
- St와 en 사이에 여러개의 철자가 와도 된다.

문제441. 우리반 테이블에서 성씨가 '김'씨 또는 '허'씨 이고 끝의 글자가 '민'인 학생들의 이름을 출력하시오.

```
Select ename
      From emp12
      Where regexp_like (ename, '^(김|허)(.)+민$');
```

서브표현식

- 서브표현식은 표현식을 소괄호로 묶은 표현식이다.

- 서브표현식은 하나의 단위로 처리된다.

연산자	설명
(표현식)	괄호 안의 표현식을 하나의 단위로 취급
+	1회 또는 그 이상의 횟수로 일치

Ex)
```
Select regexp_substr ('ababc', '(ab)+c') as c1,
       regexp_substr ('ababc', 'ab+c') as c2,
       regexp_substr ('abd', 'a(b|c)d') as c3,
       regexp_substr ('abd', 'ab|dc') as c4
From dual;
```

Regexp_count 함수
특정 단어나 철자가 문장에서 몇번 반복되어서 출력되는지 확인하는 함수

Count - 테이블의 행의 건수를 세는 함수
Regexp_count

Ex)

```
SELECT REGEXP_COUNT(
'ccacctttccctccactcctcacgttctcacctgtaaagcgtccctccctcatccccatgccccttaccctgcag
ggtagagtaggctagaaaccagagagctccaagctccatctgtggagaggtgccatccttgggctgcagagagaggag
aatttgccccaaagctgcctgcagagcttcaccacccttagtctcacaaagccttgagttcatagcatttcttgagtt
ttcaccctgcccagcaggacactgcagcacccaaagggcttcccaggagtagggttgccctcaagaggctcttgggtc
tgatggccacatcctggaattgttttcaagttgatggtcacagccctgaggcatgtaggggcgtggggatgcgctctg
ctctgctctcctctcctgaacccctgaacccctctggctaccccagagcacttagagccag',
'gtc') AS Count
FROM dual;
```

문제442. 겨울왕국(winter_kingdom) 에는 elsa가 몇번 나오는지 확인하시오.

```
Select sum(건수)
From (Select win_text, regexp_count (lower (win_text), 'elsa') 건수
            From winter_kingdom);
```

Regexp_replace 함수

Ex)
이름과 월급을 출력하는데 월급을 출력할 때 replace 함수를 이용해서 숫자 0 을 *로 출력하

시오.

```
Select ename, sal, replace (sal, 0, '*')
        From emp;
```

Ex)
이름과 월급을 출력하는데 월급 숫자 0~3까지를 * 로 출력하시오.

```
Select ename, sal, regexp_replace (sal,'[0-3]','*')
        From emp;
```

[0-3] 은 0 에서 3 까지를 의미한다.

문제443. emp12 테이블에서 이름과 나이를 출력하는데 나이를 출력할 때 앞의 숫자를 *로 출력되게 하시오.

```
Select ename, replace (age, substr(age,1,1), '*')
        From emp12;
```

다중 insert 문

```
Insert into emp(empno, ename, sal)
        Values (1234,'SCOTT',3000);
```

그동안 배웠던 insert 문장은 한번에 한건만 입력 할 수 있었다.
다중 insert 문을 이용하면 여러개의 테이블에 동시에 같은 데이터를 여러 개 입력할 수 있다.

다중 insert 문의 종류 4가지

1. 무조건 all insert 문
2. 조건부 all insert 문
3. 조건부 first insert문
4. Pivoting insert문

무조건 all insert 문

- 여러개의 테이블에 조건 없이 한번에 데이터를 입력하는 것

Ex)

```
Create table target_a
As
Select *
        From emp
        Where 1 = 2;
```

```
Create table target_b
As
Select *
	From emp
	Where 1 = 2;

Create table target_c
As
Select *
	From emp
	Where 1 = 2;
```

Where 절의 1 = 2 가 거짓이므로 emp 테이블을 가져와서 target_c 테이블을 생성 할 때 데이터는 가져오지 못하고 구조만 가져와서 만든다.

```
Insert all into target_a
	Into target_b
	Into target_c
Select *
	From emp;
```

문제 444. emp12 테이블을 emp12_a, emp12_b, emp12_c 로 구조만 가져와서 만들고 무조건 all insert문으로 emp12테이블의 데이터를 동시에 입력하시오.

```
Create table emp12_a
As
	Select *
		From emp12
		Where 1 = 2;

Create table emp12_b
As
	Select *
		From emp12
		Where 1 = 2;

Create table emp12_c
As
	Select *
		From emp12
		Where 1 = 2;

Insert all into emp12_a
		Into emp12_b
		Into emp12_c
	Select *
```

```
        From emp12;

Commit;

Truncate table target_a;
Truncate table target_b;
Truncate table target_c;
Truncate table emp12_a;
Truncate table emp12_b;
Truncate table emp12_c;
```

조건부 all insert 문

- 조건에 맞는 데이터만 입력되게 조건을 주는 입력문

ex)
Target_a 테이블에는 comm을 받는 사원들만 입력하고
Target_b 테이블에는 comm을 받지 않는 사원들만 입력하시오.

```
Insert all
    When comm is not null then
    Into target_a (empno, ename, sal, comm)
    When comm is null then
    Into target_b (empno, ename, sal, comm)
Select empno, ename, sal, comm
    From emp;
```

문제445. emp12 테이블의 데이터를 아래의 3개의 테이블에 입력하는데 통신사가 sk면 emp12_sk에 입력하고 lg면 emp12_lg에 입력하고 kt면 emp12_kt에 입력하시오.

```
Create table emp12_sk As Select * From emp12 where 1 = 2;
Create table emp12_lg As Select * From emp12 where 1 = 2;
Create table emp12_kt As Select * From emp12 where 1 = 2;

Insert all
    When telecom = 'sk' then
    Into emp12_sk
    When telecom = 'lg' then
    Into emp12_lg
    When telecom = 'kt' then
    Into emp12_kt
Select *
    From emp12;
```

조건부 first insert 문

조건에 맞는 데이터가 첫번째 테이블에 입력이 되고 남은 나머지 데이터를 가지고 새로운 조건에 맞춰서 두번째 또는 세번째에 입력하는 insert 문

Ex)
부서번호가 20번인 사원들은 target_a에 입력하고 남은 나머지 부서번호인 사원들의 월급중에서 월급이 1200 이상은 target_b 에 입력하고 1200보다 작은 사원은 target_c에 입력하시오.

```
Truncate table target_a;
Truncate table target_b;
Truncate table target_c;

Insert first
        When deptno = 20 then into target_a
        When sal >= 1200 then into target_b
        When sal < 1200 then into target_c
Select *
        From emp;
```

SQL로 알고리즘 문제 풀기 14(몬테카를로 알고리즘)

몬테카를로 알고리즘은 수많은 노가다를 통해서 정답을 알아내는 것을 말한다.

알고리즘 문제21)

문제446. 테카를로 알고리즘으로 원주율을 구하시오.

```
Select count(*)/10000*4
        From ( select round(dbms_random.value (0,1),1) x,
                        round(dbms_random.value (0,1),1) y
                From dual
                Connect by level <= 10000 )
        Where (power(x,2)+power(y,2)) <= 1
```

20.11.16

2020년 11월 16일 월요일 오전 9:59

Pivoting insert 문

Pivot문으로 작성하지 않고 pivoting insert문으로 데이터를 입력한 후에 pivot 문을 사용하지 않은 간단한 SQL로 결과를 보게 할 때 필요한 insert문

Ex)
```
Create table week_sum_sal
(empno number(10),
 ename varchar2(10),
 mon number(10),
 tues number(10),
 wed number(10),
 thur number(10),
 fri number(10));

Insert into week_sum_sal
        Select empno, ename, sal*0.1, sal*0.2, sal*0.3, sal*0.4, sal*0.5
                From emp;

Commit;
```

문제446. week_sum_sal 에서 이름, 일당을 다 합친 주급을 출력하시오.

```
Select ename, mon+tues+wed+thur+fri
        From week_sum_sal;
```

위와 같이 sum 그룹함수를 이용하지 못하고 컬럼의 데이터를 모두 일일히 더해서 SQL문장을 작성하면 번거로운 코딩이 된다.

Pivoting insert 문으로 구현

```
Create table sales_info
(ename varchar2(10),
 sal number(10));

Insert all into sales_info values (ename, mon)
                into sales_info values (ename, tues)
                into sales_info values (ename, wed)
```

```
            into sales_info values (ename, thur)
            into sales_info values (ename, fri)
Select ename, mon, tues, wed, thur, fri
      From week_sum_sal;

Select ename, sum(sal)
      From sales_info
      Group by ename;
```

SQL 복습

1. 기본 select 문
2. 함수
    a. 단일행 함수
    b. 복수행 함수
    c. 데이터 분석 함수
3. 조인
4. 집합 연산자
5. 서브쿼리문
6. DML 문장
7. DDL 문장
8. TCL 문장
9. 계층형 질의문
10. With절
11. 제약
12. Database object 5가지
13. Flashback 기능 5가지
14. 다중 insert문 4가지

금요일 시험문제 유형

유형1)

문제447. 직업, 직업별 토탈월급을 출력하는데 직업이 SALESMAN은 제외하고 출력하고 직업별 토탈 월급이 4000이상인 것만 출력하고 직업별 토탈월급이 높은 것 부터 출력하시오.

```
Select job, sum(sal)
      From emp
      Where job != 'SALESMAN'
      Group by job
      Having sum(sal) >= 4000
      Order by sum(sal) desc;
```

유형2)

문제448. 이름, 직업, 월급을 출력하는데 직업은 abcd 순서대로 출력하고 직업을 abcd 순으로 출력한 것을 기준으로 월급을 높은 것부터 출력하시오.

```
Select ename, job, sal
    From emp
    Order by job, sal desc;
```

유형2)

문제449. 이름, 부서번호, 입사일을 출력하는데 부서번호를 ascending 하게 출력하고 부서번호를 ascending 하게 출력된 것을 기준으로 입사일을 descending 하게 출력하시오.

```
Select ename, deptno, hiredate
    From emp
    Order by deptno, hiredate desc;
```

유형3)

문제450. 직업, 직업별 토탈월급을 출력하는데 맨 아래에 전체 토탈월급이 출력되게 하시오.

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

유형3)

문제451. 위의 결과를 grouping sets로 출력하시오.

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

유형3)

문제452. 아래의 SQL의 결과를 union all로 변경하시오.

(rollup, cube, grouping sets 와 같은 레포팅 함수 사용 x)

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

Select to_number(null) deptno, job, sum(sal)
        From emp
        Group by job
Union all
Select deptno, to_char(null) job, sum(sal)
        From emp
        Group by deptno;
```

Union all 위, 아래의 쿼리의 컬럼 개수, 데이터 타입, 이름이 동일해야 한다.

유형3)

문제453. 아래의 SQL을 union all로 변경하시오.

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

Select deptno, to_char(null) job, sum(sal)
        From emp
        Group by deptno
Union all
Select to_number(null) deptno, job, sum(sal)
        From emp
        Group by rollup(job);
```

유형4)

문제454. 부서위치, 부서위치별 토탈월급을 출력하는데 부서위치별 토탈월급을 출력할 때 천단위 콤마를 부여해서 출력하시오.

```
Select d.loc, to_char(sum(e.sal), '999,999')
        From emp e, dept d
        Where e.deptno = d.deptno
        Group by d.loc;
```

유형5)

문제455. 부서위치, 부서위치별 토탈월급을 출력하는데 가로로 출력하시오.

```
Select sum(decode(d.loc, 'NEW YORK', e.sal)) "NEW YORK",
        sum(decode(d.loc, 'DALLAS', e.sal)) "DALLAS",
```

```
sum(decode(d.loc, 'CHICAGO', e.sal)) "CHICAGO"
From emp e, dept d
Where e.deptno = d.deptno;
```

유형6)

문제456. ALLEN보다 더 늦게 인사한 사원들의 이름과 입사일을 출력하는데 최근에 입사한 사원부터 출력하시오.

```
Select ename, hiredate
    From emp
    Where hiredate > (select hiredate
                      From emp
                      Where ename = 'ALLEN')
    Order by hiredate desc;
```

유형7)

문제457. 관리자가 아닌 사원들의 이름을 출력하시오.

```
Select ename
    From emp
    Where empno not in (select mgr
                        From emp
                        Where mgr is not null);
```

유형8)

뮨재458. 직업, 이름, 월급, 순위를 출력하는데 순위가 직업별로 각각 월급이 높은 순서대로 순위를 부여하시오.

```
Select job, ename, sal, dense_rank() over(partition by job
                                          Order by sal desc) 순위
    From emp;
```

유형8)

문제459. 위의 결과를 다시 출력하는데 순위가 1등인 사원들만 출력하시오.

```
1.
Select *
    From (select job, ename, sal, dense_rank() over(partition by job
                                                    Order by sal desc) 순위
          From emp)
```

```
        Where 순위 = 1;

2.
With temp_table as (select job, ename, sal,
                dense_rank() over(partition by job
                                        Order by sal desc) 순위
                                From emp)
Select *
        From temp_table
        Where 순위 = 1;
```

유형9)

문제460. 아래의 테이블을 생성하고 empno에 primary key 제약을 거시오.

(테이블명 : emp 460

컬럼명 : empno, ename, sal, hiredate)

```
1.
Create table emp460

(empno number(10),
 ename varchar2(10),
 sal number(10),
 hiredate date);

Alter table emp460
        Add constraint emp460_empno_pr primary key(empno);

2.
Create table emp460

(empno number(10) constraint emp460_empno_pk primary key,
 ename varchar2(10),
 sal number(10),
 hiredate date);
```

유형9)

문제461. 아래의 데이터를 담는 테이블을 emp461로 생성하고 아래의 데이터를 입력하고 ename에 unique 제약을 거시오.

Empno	Ename	Sal	Hiredate
1111	Scott	3000	81/11/17

2222	Smith	4000	82/12/21
3333	Allen	5000	83/05/02

```
Create table emp461

(empno number(10),
 ename varchar2(10),
 sal number(10),
 hiredate date);

Alter table emp461
        Add constraint emp461_ename_un unique(ename);

Insert into emp461
Values (1111, 'scott', 3000, to_date('81/11/17', 'RR/MM/DD'));
Insert into emp461
Values (2222, 'smith', 4000, to_date('82/12/21', 'RR/MM/DD'));
Insert into emp461
Values (3333, 'allen', 5000, to_date('83/05/02', 'RR/MM/DD'));
```

유형10)

문제462. 이름, 월급, 부서번호, 자기가 속한 부서번호의 평균 월급을 출력하시오.

```
Select e.ename, e.sal, e.deptno, v.부서평균
        From emp e, (select deptno, avg(sal) 부서평균
                                From emp
                                Group by deptno) v
        Where e.deptno = v.deptno;
```

유형10)

문제463. 위의 SQL을 with 절로 구현하시오.

```
With temp_table as (select deptno, avg(sal) 부서평균
                                From emp
                                Group by deptno)

Select e.ename, e.sal, e.deptno, t.부서평균
        From emp e, temp_table t
        Where e.deptno = t.deptno;
```

유형11)

문제464. 1부터 10 까지의 숫자중에서 홀수만 출력하시오.

1.
```
Select level
        From dual
        Where mod(level,2) = 1
        Connect by level <= 10;
```

2.
```
With num_table as (select level num1
                        From dual
                        Connect by level <= 10)

Select num1
        From num_table
        Where mod(num1,2) = 1;
```

DCL 문장

SQL문장의 종류

1. 쿼리문 : select 문의 6가지 절
2. DML문 : insert, update, delete, merge
3. DDL문 : create, alter, drop, truncate, rename
4. DCL문 : grant, revokr
5. TCL문 : commit, rollback, savepoint

유저생성하는 방법

```
Create user king
        Identified by tiger;
```

접속할 수 있는 권한을 부여하는 방법

```
Grant connect to king;
```

도스창을 열고 king으로 접속한다.

```
Sqlplus king/tiger

Show user
```

문제465. allen 이라는 유저를 패스워드 tiger1234로 생성하고 allen으로 접속할 수 있도록 접속 권한을 부여하고 접속하시오.

```
Exit

Create user allen
```

Identified by tiger1234;

Grant connect to allen;

Sqlplus allen/tiger

Show user

문제466. allen 유저에서 아래의 테이블을 생성하시오.

테이블명 : emp466
컬럼명 : empno, ename, sal

Create table emp466

(empno number(10),
 ename varchar2(10),
 sal number(10));

권한이 불충분합니다. 라는 에러가 발생한다.
Create table 권한이 있어야 테이블을 생성 할 수 있다.

Create tabel 권한을 allen 유저에게 부여하는 방법

Exit

Sys 유저로 접속한다.
Sqlplus "/as sysdba"

Sys 유저인지 확인한다.
Show user

Create table 권한을 allen에게 부여한다.
Grant create table to allen;

Allen에 다시 접속 후 테이블을 생성한다.
Exit
Sqlplus allen/tiger

Create table emp466

(empno number(10),
 ename varchar2(10),
 sal number(10));

내가 가지고 있는 시스템 권한 확인하는 방법
Select * from session_privs;

Set container
Create table : 테이블 생성 권한
Create session : 접속 할 수 있는 권한

오라클에 존재하는 권한 리스트를 확인하는 방법

Sys 유저로 접속한다.

Select * from session_privs;

Datebase에 있는 모든 테이블들을 다 볼 수 있는 권한이
Select any table 권한이다.

권한 취소하기 (revoke)

Sys 유저에서 allen 에게 주었던 create table 권한을 취소하기
Sys 유저로 접속한다.

Revoke create table from allen;

유저 삭제하는 방법

Sys 유저에서 아래와 같이 작업

Drop user allen cascade;

문제467. King 유저를 삭제하시오.

Drop user king cascade;

Database에 있는 유저목록 확인하는 방법

Select username
From dba_users;