/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

집계(Aggregation) 함수와 GROUP BY, HAVING

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

집계함수, 그룹함수, 단일행 함수

- 인수(argument)는 컬럼.

- sum(): 전체합계

- avg(): 평균

- min(): 최소값

- max(): 최대값

- stddev(): 표준편차

- variance(): 분산

- count(): 개수

- 인수:

- 컬럼명: null을 제외한 개수

- \*: 총 행수(null을 포함)

- count(\*)를 제외한 모든 집계함수들은 null을 제외하고 집계한다. (avg, stddev, variance는 주의)

- 문자형/DATE: max(), min(), count()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

--급여(salary)의 총합계, 평균, 최소값, 최대값, 표준편차, 분산, 총직원수를 조회

select sum(salary) 총합계 ,round(avg(salary),2) 평균, min(salary) 최소값, max(salary) 최대값,

round(stddev(salary),2) 표준편차, round(variance(salary),2) 분산,count(\*)

from emp;

select to\_char(sum(salary),'$9,999,999') 총합계 ,round(avg(salary),2) 평균, min(salary) 최소값, max(salary) 최대값,

round(stddev(salary),2) 표준편차, round(variance(salary),2) 분산,count(\*)

from emp;

-- 가장 최근 입사일과 가장 오래된 입사일을 조회

select max(hire\_date) "최근 입사일", min(hire\_date) "오래된 입사일" from emp;

select max(emp\_name), min(emp\_name) from emp; -- (MIN)숫자<대문자<소문자<한글(MAX)

-- emp 테이블에서 job 종류의 개수 조회

select count(distinct job) "job의 종류"from emp;

--TODO: 커미션 비율(comm\_pct)이 있는 직원의 수를 조회

select count(comm\_pct) from emp;

--TODO: 커미션 비율(comm\_pct)이 없는 직원의 수를 조회

select count(\*)-count(comm\_pct) from emp;

select count(\*) from emp where comm\_pct is null;

--TODO: 가장 큰 커미션비율(comm\_pct)과 과 가장 적은 커미션비율을 조회

select max(comm\_pct) "큰커미션비율" , min(comm\_pct)"적은 커미션비율" from emp;

--TODO: 커미션 비율(comm\_pct)의 평균을 조회.

--소수점 이하 2자리까지 출력

select round(avg(comm\_pct),2)"커미션평균" , --comm\_pct가 있는 직원의 평균

round(avg(nvl(comm\_pct,0)),2) "평균(0포함)"--전체 직원의 평균

from emp;

--TODO: 직원 이름(emp\_name) 중 사전식으로 정렬할때 가장 나중에 위치할 이름을 조회.

select max(emp\_name) from emp;

--TODO: 급여(salary)에서 최고 급여액과 최저 급여액의 차액을 출력

select max(salary)-min(salary) from emp;

--TODO: 가장 긴 이름(emp\_name)이 몇글자 인지 조회.

select max(length(emp\_name)) from emp;

--TODO: EMP 테이블의 업무(job) 종류가 몇개 있는 조회.

--교유값들의 개수

select count(distinct job) from emp;

--TODO: EMP 테이블의 부서(dept\_name)가 몇종류가 있는지 조회.

-- 고유값들의 개수

select count(distinct dept\_name) from emp;

select dept\_name, sum(salary) from emp; --이 경우 sum은 단일행 이름은 다중행이기 때문에 표가 구성이 되지 않는다.

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*

group by 절

- select의 where절 다음에 기술한다.

- 특정 컬럼(들)의 값별로 나눠 집계할 때 나누는 기준컬럼을 지정하는 구문.

- 구문: group by 컬럼명 [, 컬럼명]

- 컬럼: 분류형(범주형, 명목형) - 부서별 급여 평균, 성별 급여 합계

- select절에는 group by에서 선언한 컬럼들만 집계함수와 같이 올 수 있다.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

-- 업무(job)별 급여의 총합계, 평균, 최소값, 최대값, 표준편차, 분산, 직원수를 조회

select job, sum(salary), round(avg(salary),2), min(salary), max(salary),

round(stddev(salary),2), round(variance(salary),2), count(\*) from emp

group by job;

-- 입사연도 별 직원들의 급여 평균.

select to\_char(hire\_date,'yyyy') "입사연도",round(avg(salary),2)"평균" from emp

group by to\_char(hire\_date,'yyyy')

order by 1;

-- 급여(salary) 범위별 직원수를 출력. 급여 범위는 10000 미만, 10000이상 두 범주.

select case when salary>=10000 then '1000이상'

else '10000미만' end "등급", count(\*) 직원수 from emp

group by case when salary>=10000 then '1000이상'

else '10000미만' end;

select dept\_name,job, sum(salary) from emp --상위분류를 먼저 보여주는 것이 보기에 편하다

group by job, dept\_name

order by 1;

--TODO: 부서별(dept\_name) 직원수를 조회

select dept\_name, count(\*)"직원수" from emp

group by dept\_name;

--TODO: 업무별(job) 직원수를 조회. 직원수가 많은 것부터 정렬.

select job, count(\*)"직원수" from emp

group by job

order by 2 desc;

--TODO: 부서명(dept\_name), 업무(job)별 직원수, 최고급여(salary)를 조회. 부서이름으로 오름차순 정렬.

select dept\_name, job, count(\*) "직원수", max(salary)"최고 급여" from emp

group by dept\_name, job

order by 1;

--TODO: EMP 테이블에서 입사연도별(hire\_date) 총 급여(salary)의 합계을 조회.

--(급여 합계는 자리구분자 , 를 넣으시오. ex: 2,000,000)

select to\_char(hire\_date,'yyyy') 입사연도, to\_char(sum(salary), '9,999,999') from emp

group by to\_char(hire\_date,'yyyy');

--TODO: 업무(job)와 입사년도(hire\_date)별 평균 급여(salary)을 조회

--to\_char이용

select job, to\_char(hire\_date,'yyyy')"입사년도", ceil(avg(salary)) "평균급여" from emp

group by job, to\_char(hire\_date,'yyyy');

--extract이용

select job, extract(year from hire\_date) "입사년도", round(avg(salary),2) "평균급여" from emp

group by job, extract(year from hire\_date);

--TODO: 부서별(dept\_name) 직원수 조회하는데 부서명(dept\_name)이 null인 것은 제외하고 조회.

select dept\_name, count(\*) 직원수 from emp

where dept\_name is not null

group by dept\_name;

--TODO 급여 범위별 직원수를 출력. 급여 범위는 5000 미만, 5000이상 10000 미만, 10000이상 20000미만, 20000이상.

select case when salary<5000 then '5000미만'

when salary<10000 then '5000이상 10000미만'

when salary<20000 then '10000이상 20000미만'

when salary>=20000 then '20000이상' end "급여 범위" ,count(\*)"직원수"

from emp

group by case when salary<5000 then '5000미만'

when salary<10000 then '5000이상 10000미만'

when salary<20000 then '10000이상 20000미만'

when salary>=20000 then '20000이상' end

order by decode("급여 범위", '5000미만', 1, '5000이상 10000미만',2,'10000이상 20000미만',3,4) ;

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

having 절

- 집계결과에 대한 행 제약 조건

- group by 다음 order by 전에 온다.

- 구문

having 제약조건 : 연산자는 where절의 연산자를 사용한다. 피연산자는 집계함수의 결과

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

-- 직원수가 10 이상인 부서의 부서명(dept\_name)과 직원수를 조회

select dept\_name, count(\*) 직원수 from emp

group by dept\_name

having count(\*)>=10;

--having 직원수 >= 10;

--TODO: 15명 이상이 입사한 년도와 (그 해에) 입사한 직원수를 조회.

select extract(year from hire\_date) 입사년도, count(\*) from emp

group by extract(year from hire\_date)

having count(\*)>=15;

--TODO: 그 업무(job)을 담당하는 직원의 수가 10명 이상인 업무(job)명과 담당 직원수를 조회

select job, count(\*) from emp

group by job

having count(\*)>=10;

--TODO: 평균 급여가(salary) $5000이상인 부서의 이름(dept\_name)과 평균 급여(salary), 직원수를 조회

select dept\_name, round(avg(salary),2) 평균급여, count(\*) 직원수 from emp

group by dept\_name

having avg(salary)>=5000;

--TODO: 평균급여가 $5,000 이상이고 총급여가 $50,000 이상인 부서의 부서명(dept\_name), 평균급여와 총급여를 조회

select dept\_name, round(avg(salary),2) 평균급여, sum(salary) 총급여 from emp

group by dept\_name

having avg(salary)>=5000 and sum(salary)>=50000;

-- TODO 직원이 2명 이상인 부서들의 이름과 급여의 표준편차를 조회

select dept\_name, round(stddev(salary),2) 급여표준편차 from emp

group by dept\_name

having count(\*)>=2;

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

group by 의 확장 : rollup 과 cube

- rollup(): group by의 확장

-중간집계, 총집계를 group 별 부분집계에 추가해서 보려고 할 때 사용

-구문 : group by (컬럼명 [, 컬럼명])

- cube(): rollup의 확장

-group by 절의 컬럼에 모든 조합을 묶어서 집계.

-그룹으로 묶을 컬럼이 두개 이상일 때 사용.

grouping(), grouping\_id() 함수

-grouping(): group by에서 rollup이나 cube로 묶은 컬럼이 집계시 참여 했으면 0,

참여하지 않았으면 1을 반환한다.

- select 절에서 사용

- 구문: grouping(group by에서 사용한 컬럼명 1개)

-grouping\_id():

- 구문: grouping\_id(group by에서 사용한 컬럼명 [,컬럼명,..])

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

/\*

group by rollup(job)

1. job별 집계

2. job을 제거하고 집계

group by rollup(dept\_name, job) 아래 순으로 집계

1. (dept\_name,job) 집계

2. (dept\_name) 집계

3. () 전체집계

\*/

/\*

group by cube(dept\_name, job) 모든 경우의 수를 전부 집계

( )

(dept\_name)

(job)

(dept\_name, job)

\*/

-- EMP 테이블에서 업무(job) 별 급여(salary)의 평균과 평균의 총계도 같이나오도록 조회.

select job,ceil(avg(salary)) 평균급여

from emp

group by rollup(job);

select job,ceil(avg(salary)) 평균급여

from emp

group by cube(job);

select decode(grouping\_id(dept\_name),0,dept\_name, 1, '총평균'),--dept\_name이 그룹으로 묶일 때 사용된 값인가(T:0 F:1)

ceil(avg(salary)) 평균급여

from emp

group by rollup(dept\_name);

-- EMP 테이블에서 업무(JOB) 별 급여(salary)의 평균과 평균의 총계도 같이나오도록 조회.

-- 업무 컬럼에 소계나 총계이면 '총평균'을 일반 집계이면 업무(job)을 출력

select decode(grouping\_id(job),0,job,'총평균'), ceil(avg(salary)) 평균급여 from emp

group by rollup(job);

-- EMP 테이블에서 부서(dept\_name), 업무(job) 별 salary의 합계와 직원수를 소계와 총계가 나오도록 조회

select dept\_name, job, sum(salary) 총급여, count(\*) 직원수 from emp

group by rollup(dept\_name, job);

--grouping적용

select decode(grouping\_id(dept\_name),0,dept\_name,'총집계') dept\_name,

decode(grouping\_id(job),0,job,'중간집계') job, sum(salary) 총급여,

count(\*) 직원수 from emp

group by rollup(dept\_name, job);

select dept\_name, job, sum(salary) 총급여, count(\*) 직원수 from emp

group by cube(dept\_name, job);

--# 총계/소계 행의 경우 : 총계는 '총계', 중간집계는 '계' 로 출력

--TODO: 부서별(dept\_name) 별 최대 salary와 최소 salary를 조회

select dept\_name, max(salary), min(salary) from emp

group by dept\_name;

--TODO: 상사\_id(mgr\_id) 별 직원의 수와 총계를 조회하시오.

select decode(grouping\_id(mgr\_id),0,to\_char(mgr\_id),'총계')mgr\_id ,count(\*) 직원수 from emp

--decode(grouping\_id(mgr\_id),1,'총계',mgr\_id) : 처음 반환값으로 컬럼의 데이터형이 반환된다. 따라서 '총계'인 문자열 타입으로 결정되어 숫자도 문자열로 변환되어 실행된다.

--decode(grouping\_id(mgr\_id),0,mgr\_id,'총계') : 위와 달리 첫 반환값이 mgr\_id, 즉 숫자형이기 때문에 나중에 문자열 타입인 '총계'가 들어갈 수 없다.

--따라서 반드시 문자열이 먼저 나오도록 설계해야한다.

group by rollup(mgr\_id);

--TODO: 입사연도(hire\_date의 year)별 직원들의 수와 연봉 합계 그리고 총계가 같이 출력되도록 조회.

select decode(grouping\_id(to\_char(hire\_date,'yyyy')),0,to\_char(hire\_date,'yyyy'),'총계') hire\_date, count(\*) 직원수 , sum(salary) 연봉합계 from emp

group by rollup(to\_char(hire\_date,'yyyy'));

--TODO: 부서별(dept\_name), 입사년도별 평균 급여(salary) 조회. 부서별 집계와 총집계가 같이 나오도록 조회

select decode(grouping\_id(dept\_name),0,dept\_name,'총계')dept\_name, decode(grouping\_id(to\_char(hire\_date,'yyyy')),0,to\_char(hire\_date,'yyyy'),'계')to\_char, ceil(avg(salary)) from emp

group by rollup(dept\_name,to\_char(hire\_date,'yyyy'));

select grouping\_id(dept\_name)a, --group by에서 참여했으면 0 아니면 1반환

grouping\_id(to\_char(hire\_date,'yyyy'))b, --group by에서 참여했으면 0 아니면 1반환

grouping\_id(dept\_name, to\_char(hire\_date,'yyyy'))c, --두가지를 2진수로 집계하여 10진수로 반환(11:3, 01:1)

round(avg(salary))

from emp

group by rollup(dept\_name, to\_char(hire\_date,'yyyy'));