

초보자를 위한
Oracle SQL Database



고급 SQL 함수



한국기술교육대학교
온라인평생교육원

학습내용

- 집계 함수
- 그룹/분석 함수

학습목표

- **집계 함수**를 이해하고 사용할 수 있다.
- **그룹/분석 함수**를 이해하고 사용할 수 있다.

집계 함수



집계 함수란

집계 함수란

- 테이블의 하나 이상의 컬럼을 그룹으로 묶어 계산 결과를 출력하는 함수
- 집계함수는 NULL 값이 있는 컬럼은 조회에 포함시키지 않음
- 데이터(행)가 없는 테이블에 그룹함수 COUNT()를 사용 시 0이 출력되며 다른 집계 함수는 NULL 값이 출력됨
- COUNT(카운트), MAX(최대)와 MIN(최소) 함수는 문자, 숫자, 날짜 데이터 형에서 사용할 수 있으나, AVG(평균), SUM(합계), VARIANCE(분산), STDDEV(표준편차) 등은 NUMBER 형에서만 사용할 수 있음

집계 함수란

집계 함수란

- 단순 컬럼과 집계 함수는 같이 사용할 수 없음
(단, GROUP BY의 기준이 된 컬럼은 사용 가능)
- GROUP BY 절을 이용하여 그룹화하거나, HAVING 절을 이용하여 그룹에 대한 조건 비교를 할 수 있음



집계 함수

고급 SQL 함수

집계 함수

집계 함수의 종류

함수명	설명
AVG	평균을 계산
COUNT(*)	NULL 값을 포함한 모든 행의 수를 계산
COUNT(컬럼)	NULL 값을 제외한 행의 수를 계산
CUME_DIST	누적 분포치를 계산
DENSE_RANK	순위를 RANK()와 달리 동일 순위 다음의 순위는 동일 순위의 수와 상관없이 1 증가된 값으로 계산
FIRST	주어진 소트 지정에 대해서 FIRST 또는 LAST로서 순위를 주어서 행의 세트로부터 값의 세트에 운영하는 집계와 분석 함수
FIRST_VALUE	값의 정렬된 세트에서 첫 번째 값을 반환
GROUP_ID	지정된 GROUP BY 결과로부터 중복된 그룹을 구별
GROUPING	ROLLUP이나 CUBE 연산자와 함께 사용하여 연산이 되었는지를 보여 주는 함수 0: 사용, 1: 사용되지 않음
LAST	행을 서열화 시켜서 마지막 행을 추출

고급 SQL 함수

집계 함수

집계 함수의 종류

함수명	설명
MAX	최댓값을 계산
MIN	최솟값을 계산
PERCENT_RANK	그룹 수에 대한 값의 순위 퍼센트를 반환
PERCENTILE_CONT	연속된 분포 모델을 가정한 역 분포 함수(inverse distribution function)
PERCENTILE_DISC	이산 분포 모형을 가정하는 역 분포 함수
RANK	값의 그룹에서 값의 순위를 계산
STDDEV	표준 편차를 계산
SUM	합계를 계산
VARIAN	분산을 계산

집계 함수

집계 함수 예시

집계 함수 사용 예

- 오라클에 있는 Employees 테이블 사용
- 회사의 전체 사원수와 보너스 받는 사원의 수 구하기
- 보너스를 받는 사원은 수수료 commission_pct에 값이 있음

**SELECT COUNT(*) AS "전체 사원수",
COUNT(COMMISSION_PCT) AS "보너스 받는 사원수"**

FROM EMPLOYEES;

	전체 사원수	보너스 받는 사원수
1	107	35

- COUNT(컬럼)은 NULL 값은 제외한 행의 수를 계산

집계 함수 예시

집계 함수 사용 예

- 월급을 제일 많이 받는 금액과 제일 적은 금액, 합계와 평균을 구한 예

**SELECT MAX(SALARY), MIN(SALARY), SUM(SALARY), AVG(SALARY)
FROM EMPLOYEES;**

	MAX(SALARY)	MIN(SALARY)	SUM(SALARY)	AVG(SALARY)
1	24000	2100	691400	6461.682242990654205607476635514018691589

집계 함수

집계 함수 예시

집계 함수 사용 예

- 영업부(DEPARTMENT_ID=80)에서 영업부 매니저(JOB_ID=SA_MAN)의 평균 월급을 계산한 예
- 아래의 계산식을 보면
 - 영업부 매니저에 대한 AVG(SALARY)로 평균 월급을 계산
 - 합계를 영업부 인원($\text{SUM}(\text{SALARY}) / \text{COUNT}(\text{DEPARTMENT_ID})$)으로 나누었음
- 평균=합계를 인원수로 나눈 것이기 때문에 아래의 결과를 보면 두 값이 같음

계산식

```
SELECT
    AVG(SALARY),
    SUM(SALARY) / COUNT(DEPARTMENT_ID) AS 영업부매니저평균임금,
    COUNT(DEPARTMENT_ID) AS 사원수
FROM EMPLOYEES WHERE JOB_ID = 'SA_MAN';
```

AVG(SALARY)	영업부매니저평균임금	수
12200	12200	5

집계 함수 예시

집계 함수 사용 예

- 영업부 (DEPARTMENT_ID=80)에서 영업부 사원(JOB_ID=SA_REP)의 평균 월급을 계산한 예
- 아래의 계산식을 보면
 - 영업부 사원에 대한 AVG(SALARY)로 평균 월급을 계산
 - 합계를 영업부 인원($\text{SUM}() / \text{COUNT}()$)으로 나누었음
- 앞에서 영업부 매니저와 같은 식을 사용하였으나, 결과를 보면 두 값이 다르게 나옴

계산식

```
SELECT
    AVG(SALARY),
    SUM(SALARY) / COUNT(DEPARTMENT_ID) AS 영업부사원평균임금,
    COUNT(DEPARTMENT_ID) AS 사원수
FROM EMPLOYEES WHERE JOB_ID = 'SA_REP';
```

AVG(SALARY)	영업부사원평균임금
8350	8637.931034482758620689655172413793103448

집계 함수

고급 SQL 함수

집계 함수

집계 함수 예시

집계 함수 사용 예

```
SELECT AVG(SALARY), SUM(SALARY) / COUNT(DEPARTMENT_ID) AS 영업부매니저평균임금,
COUNT(DEPARTMENT_ID) AS 사원수 FROM EMPLOYEES WHERE JOB_ID = 'SA_MAN';

SELECT AVG(SALARY), SUM(SALARY) / COUNT(DEPARTMENT_ID) AS 영업부사원평균임금,
COUNT(DEPARTMENT_ID) AS 사원수 FROM EMPLOYEES WHERE JOB_ID = 'SA_REP';
```

- 두 SQL 문을 보면 WHERE 절의 JOB_ID의 값으로 사원과 매니저를 구분

고급 SQL 함수

집계 함수

집계 함수 예시

집계 함수 사용 예

- 영업부(DEPARTMENT_ID=80) 데이터를 보면 영업부 사원에 해당하는 직원 중 한 명이 DEPARTMENT_ID값이 NULL로 되어 있음
- 따라서, COUNT(DEPARTMENT_ID)에서 NULL 값을 제외한 행의 수로 계산이 되기 때문에 오류가 발생
- 집계함수에서 NULL은 계산에서 제외된다는 것에 주의

ST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER	DEPARTMENT_ID
Kingston	JLIVINGS	011.44.1644.429264	98/04/23	SA_REP	8400	0.2	1	80
Kent	KGRANT	011.44.1644.429263	99/05/24	SA_REP	7000	0.15	1	(null)
Brault	NCAMBRAU	011.44.1344.987668	98/12/09	SA_REP	7500	0.2	145	80
Colsen	COLSEN	011.44.1344.498718	98/03/30	SA_REP	8000	0.2	145	80

그룹/분석 함수

고급 SQL 함수

그룹/분석 함수

그룹 함수

그룹함수의 종류

- ROLLUP : 그룹의 소계 계산
- CUBE : GROUP BY 항목에 대한 다차원적인 소계 계산
- GROUPING SETS : 특정항목에 대한 소계 계산

고급 SQL 함수

그룹/분석 함수

그룹 함수

+ ROLLUP

ROLLUP

- ROLLUP에 지정된 그룹 컬럼의 목록은 Subtotal을 보여 줌
- ROLLUP에 적은 컬럼의 수와 순서가 바뀌게 되면 결과도 바뀌게 되므로 컬럼의 수와 순서에 주의하여야 함

그룹/분석 함수

고급 SQL 함수

그룹/분석 함수

그룹 함수

+ ROLLUP

학번	교과목코드	취득점수
1	2014172008 CSE132	89
2	2014172008 CSE242	74
3	2014172008 CSE244	84
4	2014172008 CSE341	98
5	2014172008 CSE430	87
6	2014172008 CSE431	93
7	2014172008 CSE440	92
8	2014172008 CSE441	52
9	2015136057 CSE132	64
10	2015136057 CSE242	94
11	2015136057 CSE244	72
12	2015136057 CSE341	86
13	2015136057 CSE430	53
14	2015136057 CSE431	87

```
SELECT 학번, 교과목코드, 취득점수
FROM 수강
ORDER BY 학번, 교과목코드;
```

수강테이블에 가서 학생테이블의 학번으로 정렬하여
신청한 교과목코드와 취득점수 검색

고급 SQL 함수

그룹/분석 함수

그룹 함수

+ ROLLUP

학번	교과목코드	취득점수
1	2014172008 CSE132	89
2	2014172008 CSE242	74
3	2014172008 CSE244	84
4	2014172008 CSE341	98
5	2014172008 CSE430	87
6	2014172008 CSE431	93
7	2014172008 CSE440	92
8	2014172008 CSE441	52
2014172008 (null)		669
10	2015136057 CSE132	64
11	2015136057 CSE242	94
12	2015136057 CSE244	72
13	2015136057 CSE341	86
14	2015136057 CSE430	53

```
SELECT 학번, 교과목코드,
       SUM(취득점수) AS 취득점수
FROM 수강
GROUP BY ROLLUP(학번, 교과목코드)
ORDER BY 학번, 교과목코드;
```

취득 점수의 소계

단순 컬럼과 집계 함수는 같이 사용할 수 없으나
GROUP BY의 기준이 된 컬럼은 사용이 가능

ROLLUP을 사용하면 자동으로 소계를 출력할 행이 추가됨

그룹/분석 함수

고급 SQL 함수

그룹/분석 함수

그룹 함수

+ ROLLUP

학번	취득점수합 타이틀 출력	점수
33 20161		73
34 2016174067	CSE132	65
35 2016174067	취득점수합	536
36 2018120034	ARB263	94
37 2018120034	CSE114	94
38 2018120034	CSE120	88
39 2018120034	CSE121	83
40 2018120034	CSE122	91
41 2018120034	CSE130	96
42 2018120034	CSE131	72
43 2018120034	CSE132	65
44 2018120034	취득점수합	683
45 전체신청과목수	취득점수합	3044

```
SELECT DECODE(GROUPING(학번),1,'전체신청과목수',학번) AS 학번,
       DECODE(GROUPING(교과목코드),1,'취득점수합',교과목코드)
AS 교과목코드,
       SUM(취득점수) AS 취득점수
FROM 수강
GROUP BY ROLLUP(학번, 교과목코드) -- 부분합 계산
ORDER BY 학번, 교과목코드;
```

취득점수합 타이틀 출력

GROUPING으로 학번이 그룹 연산에 사용이 되었는지 체크하고,
DECODE로 타이틀을 보여줄지
학번(교과목코드)를 보여줄지를 결정

교과목코드에 표시되는 NULL을 보기 좋게 바꾸기 위하여 DECODE와 GROUPING을 사용하여 “취득점수합”, “전체신청과목수”를 표시

고급 SQL 함수

그룹/분석 함수

그룹 함수

+ CUBE

CUBE

- CUBE는 지정한 컬럼에 따라 가능한 모든 조합별로 집계한 결과를 반환

그룹/분석 함수

고급 SQL 함수

그룹/분석 함수

그룹 함수

+ CUBE

학번	교과목코드	취득점수합 타이틀 출력	수
42 20181	CSE132	취득점수합	72
43 2018120034	CSE132		65
44 2018120034		취득점수합	683
45 (null) ARB263			94
46 (null) CSE114			148
47 (null) CSE120			156
48 (null) CSE121			170
49 (null) CSE122			186
50 (null) CSE130			190
51 (null) CSE131			145
52 (null) CSE132			367
53 (null) CSE242			232
54 (null) CSE244			228

학번에 출력할 것이 없어 NULL,
교과목코드별 취득점수 합계 계산

```
SELECT 학번,
       DECODE(GROUPING(교과목코드),1,'취득점수합',교과목코드)
    AS 교과목코드,
       SUM(취득점수) AS 취득점수
   FROM 수강
 GROUP BY CUBE(학번, 교과목코드)
 ORDER BY 학번, 교과목코드;
```

CUBE를 이용하여 학번별 취득점수의 합과 교과목코드별 취득점수의 합을 계산

고급 SQL 함수

그룹/분석 함수

그룹 함수

+ CUBE

학번	교과목코드	취득점수합 타이틀 출력	수
42 2018120034	CSE131		263
43 2018120034	CSE132		114
44 2018120034		취득점수합	683
45 과목코드별 취득점수합			94
46 과목코드별 취득점수합			148
47 과목코드별 취득점수합			120
48 과목코드별 취득점수합			121
49 과목코드별 취득점수합			122
50 과목코드별 취득점수합			130
51 과목코드별 취득점수합			131
52 과목코드별 취득점수합			132
53 과목코드별 취득점수합			242
54 과목코드별 취득점수합			244
과목코드별 취득점수합			244

학번란에 NULL 대신
교과목코드별 취득점수 합계 계산

```
SELECT DECODE(GROUPING(학번),1,'과목코드별 취득점수합',학번) AS 학번,
       DECODE(GROUPING(교과목코드),1,'취득점수합',교과목코드) AS 교과목코드,
       SUM(취득점수) AS 취득점수
   FROM 수강
 GROUP BY CUBE(학번, 교과목코드)
 ORDER BY 학번, 교과목코드;
```

학번란에 표시되는 NULL을 보기 좋게 바꾸기 위하여 DECODE와
GROUPING을 추가하여 “과목코드별 취득점수합”이 표시되게 함

그룹/분석 함수

고급 SQL 함수

그룹/분석 함수

그룹 함수

+ GROUPING SETS

14172008 (null)	669
15136057 (null)	603
15136065 (null)	553
16174067 (null)	536
18120034 (null)	683
(null) ARB263	94
(null) CSE114	148
(null) CSE120	156
(null) CSE121	170
(null) CSE122	186
(null) CSE130	190
(null) CSE131	145
(null) CSE132	367

```
SELECT 학번, 교과목코드, SUM(취득점수)
AS 취득점수
FROM 수강
GROUP BY GROUPING SETS(학번,
교과목코드)
ORDER BY 학번, 교과목코드;
```

표시된 인수들 간에는 계층구조인
ROLLUP과는 달리 평등한 관계이므로
인수의 순서가 바뀌어도 결과는 같음

GROUPING SETS에 표시된 인수들에 대한 개별 집계

고급 SQL 함수

그룹/분석 함수

그룹 함수

+ GROUPING SETS

학번	취득점수합
1 2014172008	536
학번란에 NULL 대신 교과목코드별 취득점수 합계 계산	취득점수합
6 과목코드별 취득점수합	263
7 과목코드별 취득점수합	114
8 과목코드별 취득점수합	120
9 과목코드별 취득점수합	121
10 과목코드별 취득점수합	122
11 과목코드별 취득점수합	130
12 과목코드별 취득점수합	131
13 과목코드별 취득점수합	132

교과목코드에 NULL 대신
취득점수합 타이틀 출력

536

```
SELECT DECODE(GROUPING(학번),1,'과목코드별 취득점수합',학번) AS 학번,
DECODE(GROUPING(교과목코드),1,'취득점수합',교과목코드) AS
교과목코드,
SUM(취득점수) AS 취득점수
FROM 수강
GROUP BY GROUPING SETS(학번, 교과목코드)
ORDER BY 학번, 교과목코드;
```

학번과 교과목코드의 NULL을 보기 좋게 바꾸기 위하여 DECODE와 GROUPING을
추가하여 “과목코드별 취득점수합”, “취득점수합”이 표시되게 함

그룹/분석 함수

고급 SQL 함수

그룹/분석 함수

분석 함수

분석 함수란

- 테이블에 있는 행에 대해 특정 그룹별로 집계 값을 산출할 때 사용됨
- 분석 함수에서 사용하는 행별 그룹을 **윈도우**라고 함

```
SELECT
    분석함수 OVER
    ( [ PARTITION BY 컬럼1, 컬럼2, ... | ORDER BY
        컬럼1, 컬럼2 ... | WINDOW 절 ] )
FROM 테이블;
```

고급 SQL 함수

그룹/분석 함수

분석 함수

분석 함수의 종류

- 분석 함수 : 컬럼 값을 분석하려는 함수
 - ROW_NUMBER : 행 번호
 - CUME_DIST : 분산 값
 - RANK : 등수 계산(중복 등수 발생 시 다음 순위를 증가)
 - DENSE_RANK : 등수 계산(중복 등수가 발생해도 다음 등수 값 사용)
 - PERCENT_RANK : 백분율
 - NTILE(n) : 전체 데이터 분포를 n-Buckets으로 나누어 표시
 - FIRST_VALUE : 정렬된 값 중에서 첫 번째 값을 반환
 - LAST_VALUE : 정렬된 값 중에서 마지막 값을 반환
- PARTITION BY 절 : 컬럼을 세분하여 구분
- ORDER BY 절 : 정렬 순서를 지정
- WINDOW 절 : PARTITION을 더 세부 그룹으로 분할

그룹/분석 함수

고급 SQL 함수

그룹/분석 함수

분석 함수

학번	교과목코드	취득점수	전체취득점수
1	2018120034 ARB263	94	3044
2	2018120034 CSE114	94	3044
3	2018120034 CSE120	88	3044
4	2018120034 CSE121	83	3044
5	2018120034 CSE122	91	3044
6	2018120034 CSE130	96	3044
7	2018120034 CSE131	72	3044
8	2018120034 CSE132	65	3044
9	2014172008 CSE242	74	3044
10	2014172008 CSE244	84	3044
11	2014172008 CSE341	98	3044
12	2014172008 CSE430	87	3044
13	2014172008 CSE431	93	3044
14	2014172008 CSE440	92	3044
15	2014172008 CSE441	52	3044
16	2014172008 CSE132	89	3044

```
SELECT 학번, 교과목코드, 취득점수,
       SUM(취득점수) OVER() AS
    전체취득점수
   FROM 수강;
```

위의 예를 보면 단순 컬럼과 집계 함수를 GROUP BY 없이 사용하였음
OVER를 사용하여 SUM(취득점수)의 합계를 쉽게 구할 수 있음

고급 SQL 함수

그룹/분석 함수

분석 함수

학번	교과목코드	취득점수	취득
5	2014172008 CSE430	87	669
6	2014172008 CSE341	98	669
7	2014172008 CSE244	84	669
8	2014172008 CSE242	74	603
9	2015136057 CSE441	49	603
10	2015136057 CSE132	64	603
11	2015136057 CSE440	98	603
12	2015136057 CSE431	87	603
13	2015136057 CSE430	53	603
14	2015136057 CSE341	86	603
15	2015136057 CSE244	72	603
16	2015136057 CSE242	94	603
17	2015136065 CSE132	84	553
18	2015136065 CSE441	89	553
19	2015136065 CSE440	58	553
20	2015136065 CSE431	67	553

```
SELECT 학번, 교과목코드, 취득점수,
       SUM(취득점수)
      OVER(PARTITION BY 학번) AS
    전체취득점수
   FROM 수강;
```

PARTITION BY 학번 옵션에 따라
학번별로 SUM(취득점수)가 계산되었음

OVER 안에 PARTITION BY 절을 사용하여 합계를 학번별로 구해 보기

그룹/분석 함수

고급 SQL 함수

그룹/분석 함수

분석 함수

학번	교과목코드	취득점수	취득점수
5	2014172008 CSE132	89	386
6	2014172008 CSE440	92	478
7	2014172008 CSE431	93	571
8	2014172008 CSE341	98	669
9	2015136057 CSE441	49	49
10	2015136057 CSE430	53	102
11	2015136057 CSE132	64	166
12	2015136057 CSE244	72	238
13	2015136057 CSE341	86	324
14	2015136057 CSE431	87	411
15	2015136057 CSE242	94	505
16	2015136057 CSE440	98	603
17	2015136065 CSE341	46	46
18	2015136065 CSE440	58	104
19	2015136065 CSE242	64	168
20	2015136065 CSE431	67	235

```
SELECT 학번, 교과목코드, 취득점수,
       SUM(취득점수)
OVER(PARTITION BY 학번 ORDER
BY 취득점수) AS "취득점수 누적"
FROM 수강;
```

PARTITION BY 학번 ORDER
BY 취득점수 옵션에 따라
학번별로 취득점수가 누적됨

학번별 취득점수합을 수정해서 학번별로 취득점수를 누적

고급 SQL 함수

그룹/분석 함수

분석 함수

1	학번	교과목코드	취득점수
2	2014172008 CSE441	52	
3	2014172008 CSE242	74	
4	2014172008 CSE244	84	
5	2014172008 CSE430	87	
6	2014172008 CSE132	89	
7	2014172008 CSE440	92	
8	2014172008 CSE431	93	
9	2014172008 CSE341	98	
10	2015136057 CSE441	49	
11	2015136057 CSE430	53	
12	2015136057 CSE132	64	
13	2015136057 CSE244	72	
14	2015136057 CSE341	86	
15	2015136057 CSE431	87	
16	2015136057 CSE242	94	
17	2015136057 CSE440	98	

```
SELECT
       ROW_NUMBER()
OVER(PARTITION BY 학번 ORDER
BY 취득점수) AS "순번",
학번, 교과목코드, 취득점수
FROM 수강;
```

ROW_NUMBER() 함수로
학번별로 순번을 표시

SUM() 대신에 ROW_NUMBER()를 사용하여 학번별로 순번을 지정

그룹/분석 함수

고급 SQL 함수

그룹/분석 함수

분석 함수

등수	학번	교과목코드	취득점수
1	2014172008	CSE341	98
2	2014172008	CSE431	93
3	2014172008	CSE440	92
4	2014172008	CSE132	89
5	2014172008	CSE430	87
6	2014172008	CSE244	84
7	2014172008	CSE242	74
8	2014172008	CSE441	52
9	2015136057	CSE440	98
10	2015136057	CSE242	94
11	2015136057	CSE431	87
12	2015136057	CSE341	86
13	2015136057	CSE244	72
14	2015136057	CSE132	64
15	2015136057	CSE430	53
16	2015136057	CSE441	49

```
SELECT
    RANK() OVER(PARTITION BY 학번
    ORDER BY 취득점수 DESC) AS "등수",
    학번, 교과목코드, 취득점수
FROM 수강;
```

등수는 큰 점수순으로 표시돼야 해서
ORDER BY를 DESC 순으로 지정

ROW_NUMBER() 대신에 RANK()를 사용하여 취득점수별 등수를 구함

고급 SQL 함수

그룹/분석 함수

분석 함수

등수	학번	교과목코드	취득점수
1	2016174067	CSE114	54
2	2018120034	CSE130	96
2	2018120034	CSE114	94
2	2018120034	ARB263	94
3	2018120034	CSE122	91
4	2018120034	CSE120	83
5	2018120034	CSE121	72
6	2018120034	CSE131	72

```
SELECT
    RANK() OVER(PARTITION BY 학번
    ORDER BY 취득점수 DESC) AS "등수1",
    DENSE_RANK() OVER(PARTITION BY 학번
    ORDER BY 취득점수 DESC) AS "등수2",
    학번, 교과목코드, 취득점수
FROM 수강;
```

동점자

RANK()와 DENSE_RANK()의 차이점 확인

실습하기



고급 SQL 함수 실습하기

Ac remote

소스 템(M) 도구(I) 창(W) 도움말(H)

작업 시작 페이지 19c remote EMPLOYEES

워크시트 질의 작성기

1. 집계 함수

- 테이블의 하나 이상의 컬럼을 그룹으로 묶어 계산 결과를 출력하는 함수를 뜻합니다.
- 집계함수는 NULL값이 있는 컬럼은 조회에 포함시키지 않습니다.
- 데이터(행)가 없는 테이블에 그룹함수 COUNT()를 사용 시 0이 출력되며 다른 집계 함수는 NULL 값이 출력됩니다.
- COUNT(카운트), MAX(최대), MIN(최소) 함수는 문자, 숫자, 날짜 데이터 형에서 사용할 수 있으나, AVG(평균), STDN(합계), VARIANCE(분산), STDEV(표준편차) 등은 NUMBER 형에서만 사용할 수 있습니다.
- 단순 컬럼과 집계 함수는 같이 사용할 수 있습니다.
- 단, GROUP BY의 기준이 된 컬럼은 사용 가능).
- GROUP BY를 이용하여 그룹화하거나, HAVING 절을 이용하여 그룹에 대한 조건 비교를 할 수 있습니다.
- 집계 함수 (Aggregate Function)
- 예 1

스크립트 출력 질의 결과

SQl | 인출된 모든 행: 1(0,003초)

	Avg(Salary)	영업부사원평균임금	사원수
1	8350	8637.931034482758620689655172413793103448	29

SQL 내역

메시지 - 로그

19c remote*: 2022. 8. 22 오후 8:18:27; 커밋 성공
19c remote*: 2022. 8. 22 오후 8:22:10; 커밋 성공

집계 함수

- + Oracle Database 19c 버전 사용
- + 최신 버전인 Oracle Database 21c는 현재 cloud에서만 사용 가능하며, 지원 기간은 2년

실습단계

집계 함수

집계함수는 NULL값은 조회에 포함시키지 않음

그룹함수 COUNT()는 NULL일 경우 0을, 다른 집계 함수는 NULL 값이 출력

집계 함수 예 1

집계 함수 예 2

집계 함수 예 3

집계 함수 예 4

그룹 함수

ROLLUP

ROLLUP에 지정된 그룹 컬럼의 목록은 부분합(Subtotal)을 보여줌

ROLLUP에 적은 컬럼의 수와 순서에 따라 결과가 바뀌게 됨

CUBE

지정한 컬럼에 대한 모든 조합별로 집계 결과를 반환

실습하기

실습단계

GROUPING SETS

인수들에 대한 개별 집계 결과를 반환

분석함수

OVER를 사용하여 SUM(취득점수)로 합계 계산

PARTITION BY 학번 옵션에 따라 학번별로 SUM(취득점수)을 계산

PARTITION BY 학번 ORDER BY 취득점수 옵션에 따라 학번별 누적 점수 계산

RANK()와 DENSE_RANK()의 차이점

RANK()는 2등이 2명이면 그 다음 등수는 4등

DENSE_RANK()는 2등이 2명이 있어도 그 다음 등수는 3등