

6장 그룹 쿼리와 집합 연산자

6.1 기본 집계 함수

- 그룹 함수는 하나 이상의 행을 그룹으로 묶어 연산하여 총합, 평균 등 하나의 결과로 나타난다.
- 그룹 함수의 종류

구분	설명
SUM	그룹의 누적 합계를 반환한다.
AVG	그룹의 평균을 반환한다.
COUNT	그룹의 총 개수를 반환한다.
MAX	그룹의 최대값을 반환한다.
MIN	그룹의 최소값을 반환한다.
STDDEV	그룹의 표준편차를 반환한다.
VARIANCE	그룹의 분산을 반환한다.

6.1.1 합계를 구하는 SUM 함수

- SUM 함수는 해당 칼럼 값들에 대한 총합을 구하는 함수이다.
- 그룹 함수는 다른 연산자와는 달리 해당 칼럼 값이 NULL 인 것을 제외하고 계산한다.

```
SELECT SUM(SAL)
FROM EMP;

SELECT SUM(COMM)
FROM EMP; -- 그룹 함수와 NULL
```

6.1.2 평균 구하는 AVG 함수

- AVG 함수는 해당 칼럼 값들에 대해 평균을 구하는 함수이다.
- 해당 칼럼 값이 NULL 인 것에 대해서는 제외하고 계산한다.

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

6.1.3 최대값 구하는 MAX, 최소값 구하는 MIN 함수

- 지정한 칼럼 값들 중에서 최대값을 구하는 함수가 MAX이고, 최소값을 구하는 함수가 MIN이다.

```
SELECT MAX(SAL), MIN(SAL)
FROM EMP;
```

6.1.4 그룹 함수와 단순 컬럼

- SELECT문에 그룹 함수를 사용하는 경우, 그룹 함수를 적용하지 않은 단순 컬럼은 올 수 없다.

```
SELECT MAX(SAL)
FROM EMP;

SELECT ENAME, MAX(SAL)
FROM EMP; -- 오류 발생
```

6.1.5 로우 개수 구하는 COUNT 함수

- COUNT 함수는 테이블에서 조건을 만족하는 행의 개수를 반환하는 함수이다.

```
SELECT COUNT(COMM)
FROM EMP;

SELECT COUNT(*), COUNT(COMM)
FROM EMP;

SELECT COUNT(JOB) 업무수
FROM EMP;

SELECT COUNT(DISTINCT JOB) 업무수
FROM EMP;
```

6.2 GROUP BY절

- 그룹함수를 쓰되 어떤 컬럼 값을 기준으로 그룹함수를 적용할 경우 GROUP BY 절 뒤에 해당 컬럼을 기술하면 된다.
- 합계, 평균, 최대값이나, 최소값 등을 어떤 칼럼을 기준으로 그 칼럼의 값 별로 보고자 할 때 GROUP BY 절 뒤에 해당 칼럼을 기술하면 된다.
- GROUP BY 절을 사용할 때 주의할 점은 GROUP BY 절 다음에는 칼럼의 별칭을 사용할 수 없고, 반드시 칼럼명을 기술해야 한다.

```
-- 형식
SELECT 칼럼명, 그룹함수
FROM 테이블명
WHERE 조건 (연산자)
GROUP BY 칼럼명;

SELECT DEPTNO
FROM EMP
GROUP BY DEPTNO;

SELECT DEPTNO, AVG(SAL)
```

```

FROM EMP
GROUP BY DEPTNO;

SELECT DEPTNO, MAX(SAL), MIN(SAL)
FROM EMP
GROUP BY DEPTNO;

```

6.3 HAVING 조건

- SELECT 절에 조건을 사용하여 결과를 제한할 때는 WHERE 절을 사용하지만 그룹의 결과를 제한할 때는 HAVING 절을 사용한다.

```

SELECT DEPTNO, AVG(SAL)
FROM EMP
GROUP BY DEPTNO
HAVING AVG(SAL) >= 2000;

SELECT DEPTNO, MAX(SAL), MIN(SAL)
FROM EMP
GROUP BY DEPTNO
HAVING MAX(SAL) > 2900;

```

6.4 집합 연산자

6.4.1 UNION

- 집합의 합집합 개념
- 두 개 이상의 개별 SELECT 쿼리를 연결
- 개별 SELECT 쿼리 반환 결과가 중복될 경우 UNION 연산 결과는 한 로우만 반환됨

```

-- 집합 연산자
DROP TABLE EXP_GOODS_ASIA;
CREATE TABLE EXP_GOODS_ASIA (
    COUNTRY VARCHAR2(10),
    SEQ      NUMBER,
    GOODS    VARCHAR2(80));

INSERT INTO EXP_GOODS_ASIA VALUES ('한국', 1, '원유제외 석유류');
INSERT INTO EXP_GOODS_ASIA VALUES ('한국', 2, '자동차');
INSERT INTO EXP_GOODS_ASIA VALUES ('한국', 3, '전자집적회로');
INSERT INTO EXP_GOODS_ASIA VALUES ('한국', 4, '선박');
INSERT INTO EXP_GOODS_ASIA VALUES ('한국', 5, 'LCD');
INSERT INTO EXP_GOODS_ASIA VALUES ('한국', 6, '자동차부품');
INSERT INTO EXP_GOODS_ASIA VALUES ('한국', 7, '휴대전화');
INSERT INTO EXP_GOODS_ASIA VALUES ('한국', 8, '환식탄화수소');
INSERT INTO EXP_GOODS_ASIA VALUES ('한국', 9, '무선송신기 디스플레이 부속품');
INSERT INTO EXP_GOODS_ASIA VALUES ('한국', 10, '철 또는 비합금강');

INSERT INTO EXP_GOODS_ASIA VALUES ('일본', 1, '자동차');
INSERT INTO EXP_GOODS_ASIA VALUES ('일본', 2, '자동차부품');
INSERT INTO EXP_GOODS_ASIA VALUES ('일본', 3, '전자집적회로');

```

```

INSERT INTO EXP_GOODS_ASIA VALUES ('일본', 4, '선박');
INSERT INTO EXP_GOODS_ASIA VALUES ('일본', 5, '반도체웨이퍼');
INSERT INTO EXP_GOODS_ASIA VALUES ('일본', 6, '화물차');
INSERT INTO EXP_GOODS_ASIA VALUES ('일본', 7, '원유제외 석유류');
INSERT INTO EXP_GOODS_ASIA VALUES ('일본', 8, '건설기계');
INSERT INTO EXP_GOODS_ASIA VALUES ('일본', 9, '다이오드, 트랜지스터');
INSERT INTO EXP_GOODS_ASIA VALUES ('일본', 10, '기계류');

COMMIT;

SELECT GOODS
  FROM EXP_GOODS_ASIA
 WHERE COUNTRY = '한국'
 ORDER BY SEQ;

SELECT GOODS
  FROM EXP_GOODS_ASIA
 WHERE COUNTRY = '일본'
 ORDER BY SEQ;

-- UNION
SELECT GOODS
  FROM EXP_GOODS_ASIA
 WHERE COUNTRY = '한국'
UNION
SELECT GOODS
  FROM EXP_GOODS_ASIA
 WHERE COUNTRY = '일본';

```

6.4.2 UNION ALL

- UNION과 유사
- 개별 SELECT 쿼리 반환 결과가 중복될 경우, 중복되는 건까지 모두 반환

```

-- UNION ALL
SELECT GOODS
  FROM EXP_GOODS_ASIA
 WHERE COUNTRY = '한국'
UNION ALL
SELECT GOODS
  FROM EXP_GOODS_ASIA
 WHERE COUNTRY = '일본';

```

6.4.3 INTERSECT

- 집합의 교집합 개념
- 두 개 이상의 개별 SELECT 쿼리를 연결
- 개별 SELECT 쿼리 반환 결과 중 공통된 항목만 추출

```

-- INTERSECT
SELECT GOODS

```

```

FROM EXP_GOODS_ASIA
WHERE COUNTRY = '한국'
INTERSECT
SELECT GOODS
FROM EXP_GOODS_ASIA
WHERE COUNTRY = '일본';

```

6.4.4 MINUS

- 집합의 차집합 개념
- 두 개 이상의 개별 SELECT 쿼리를 연결
- 개별 SELECT 쿼리 반환 결과 중 중복된 것을 제외한 선행 쿼리 결과 추출

```

-- MINUS
SELECT GOODS
FROM EXP_GOODS_ASIA
WHERE COUNTRY = '한국'
MINUS
SELECT GOODS
FROM EXP_GOODS_ASIA
WHERE COUNTRY = '일본';

SELECT GOODS
FROM EXP_GOODS_ASIA
WHERE COUNTRY = '일본'
MINUS
SELECT GOODS
FROM EXP_GOODS_ASIA
WHERE COUNTRY = '한국';

```

6.4.5 집합 연산자 제한사항

- 개별 SELECT 쿼리의 SELECT 리스트 개수와 데이터 타입이 일치해야 한다.
- ORDER BY 절은 맨 마지막 개별 SELECT 쿼리에만 명시 가능하다.
- BLOB, CLOB, BFILE 타입의 컬럼에 대해서는 집합 연산자를 사용할 수 없다.
- UNION, INTERSECT, MINUS 연산자는 LONG형 컬럼에는 사용할 수 없다.

```

-- 집합 연산자 제한사항
--1. 집합 연산자로 연결되는 각 SELECT문의 SELECT 리스트의 개수와 데이터 타입은 일치해야 한다.
SELECT GOODS
FROM EXP_GOODS_ASIA
WHERE COUNTRY = '한국'
UNION
SELECT SEQ, GOODS
FROM EXP_GOODS_ASIA
WHERE COUNTRY = '일본';
--ORA-01789: 질의 블록은 부정확한 수의 결과 열을 가지고 있습니다.
--01789. 00000 - "query block has incorrect number of result columns"

```

```

SELECT SEQ
  FROM EXP_GOODS_ASIA
 WHERE COUNTRY = '한국'
UNION
SELECT GOODS
  FROM EXP_GOODS_ASIA
 WHERE COUNTRY = '일본';
--ORA-01790: 대응하는 식과 같은 데이터 유형이어야 합니다
--01790. 00000 - "expression must have same datatype as corresponding expression"

--2. 집합 연산자로 SELECT 문을 연결할 때 ORDER BY절은 맨 마지막 문장에서만 사용할 수 있다.
SELECT GOODS
  FROM EXP_GOODS_ASIA
 WHERE COUNTRY = '한국'
 ORDER BY GOODS
UNION
SELECT GOODS
  FROM EXP_GOODS_ASIA
 WHERE COUNTRY = '일본';
--ORA-00933: SQL 명령어가 올바르게 종료되지 않았습니다
--00933. 00000 - "SQL command not properly ended"

SELECT GOODS
  FROM EXP_GOODS_ASIA
 WHERE COUNTRY = '한국'
UNION
SELECT GOODS
  FROM EXP_GOODS_ASIA
 WHERE COUNTRY = '일본'
 ORDER BY GOODS;

```

[과제] 과제-06-01.TXT

SQL> CONN ORA_USER/HONG 로 접속하여 SQL문을 작성하세요.

1. 직원테이블에서 입사년도별 직원수를 구하는 쿼리를 작성해보자.

<정답>

2. kor_loan_status 테이블에서 2012년도 월별, 지역별 대출 총 잔액을 구하는 쿼리를 작성하라.

<예시>

201211	전북	23762
201211	제주	6351.7
201211	충남	30933.2
201211	충북	18850.3
201212	강원	17449.7
201212	경기	281501.2
201212	경남	51368.5

<정답>