

```
1 REM Author :
2 REM Date :
3 REM Objective : Chapter 2. Basic SELECT
4 REM Environment : CentOS 6.5, SQLGate 2010 for Oracle, Oracle 11g Enterprise
  Edition 11.2.0
5
6 REM SELECT의 기능
7 1. Selection : 조건검색, Row에 대한 필터링
8 2. Projection : column 에 대한 필터링
9 3. Join : 여러 테이블에서의 검색
10
11 REM SELECT Syntax
12
13     SELECT [DISTINCT | ALL] { * | column1, column2 [AS [alias]] | expr }
14     FROM table_name
15     WHERE condition
16     ORDER BY column [ASC | DESC];
17
18 1. SELECT 절과 FROM 절은 반드시 있어야 한다.
19 2. SELECT 절 다음에 질의하고 싶은 칼럼을 차례대로 나열한다. 이 때 여러 개의 칼럼 구분은 쉼표
  (,)로 한다.
20 3. FROM 절 다음에는 조회할 테이블 이름을 적는다.
21 4. * : 모든 칼럼을 조회한다.
22 5. ALL : 모든 결과 ROW를 보여준다.(기본값)
23 6. DISTINCT : 중복된 ROW 를 제외한 ROW를 보여준다.
24 7. expr : SQL 함수를 사용하거나, 수학 연산을 포함한 표현식
25 8. alias : 칼럼에 대한 별칭 사용.
26 9. Default Column Heading : column 명이 대문자로 Display 된다.
27 10. Default Data Justification : Number 값은 오른쪽 정렬, Character 와 Date 값은
  왼쪽 정렬된다.
28
29 REM 모든 열 선택
30     SELECT * FROM dept;
31
32     SELECT *
33     FROM emp;
34
35     SELECT * FROM emp;
36
37     SQL> SET pagesize 1000 -- 1000줄을 한 페이지로 설정하는 SQL*Plus 명령 실행
38     SQL> / -- SQL*Plus buffer 에 들어있는 SQL 문장을 다시 실행
39     SQL> COL[UMN] mgr FOR[MAT] 9999 --숫자 칼럼의 크기를 4자리로 조정하는
  SQL*Plus 명령 실행
40     SQL> COL ename FORMAT A8 -- 문자칼럼의 크기를 8자리로 조정하는 SQL*Plus 명령
  실행
41
42 REM 특정 열 선택
```

```
43 1. 각 열의 구분은 "," 로 한다.
44     SELECT empno, ename, sal
45     FROM emp;
46
47     SELECT empno, ename, job, mgr FROM emp;
48
49 REM 산술연산자 : 수학 연산 표현식
50 1. +, - : 음수, 혹은 양수를 나타내는 기호. 단항 연산자.
51 2. *(multiply), /(divide) : 곱하기, 나누기를 의미. 이항 연산자.
52 3. +(add), -(subtract) : 더하기, 빼기를 의미. 이항 연산자.
53 4. 연산자의 우선순위가 있다. 1 --> 2 --> 3
54 5. 우선순위가 높은 연산 먼저 수행하며, 같은 우선순위의 연산자들을 왼쪽에서 오른쪽으로 순서대로
    계산해 나간다.
55 6. 괄호를 사용하여 우선순위를 조절할 수 있다.
56
57     SELECT empno, ename, sal, sal + 100
58     FROM emp;
59
60     SELECT sal, -sal FROM emp;
61     SELECT sal, sal * 1.1 FROM emp;
62     SELECT sal, comm, sal + comm FROM emp;
63     SELECT sal, -sal + 100 * -2 FROM emp;
64     SELECT sal, (-sal + 100) * -2 FROM emp;
65     SELECT empno, ename, sal, sal * 12 FROM emp;
66     SELECT empno, ename, sal, sal * 12 + comm FROM emp;
67     SELECT empno, ename, sal, sal + comm * 12 FROM emp;
68     SELECT empno, ename, sal, (sal + comm) * 12 FROM emp;
69
70 REM NULL value
71 1. NULL 이란?
72     1) 특정 행, 특정 열에 대한 아직 값을 알 수 없는 상태, 의미가 없는 상태를 표현
73     2) 이용할 수 없거나, 지정되지 않거나, 알 수 없거나, 적용할 수 없는 값
74 2. 0 또는 공백과 다르다.
75 3. 연산의 대상에 포함되지 않는다.
76 4. NULL 값을 포함한 산술 연산 식의 결과는 언제나 NULL 이다.
77 5. NOT NULL 또는 Primary Key 제약조건이 걸린 칼럼에서는 NULL value가 나타날 수 없다.
78 6. Oracle DB에서는 NULL 인 칼럼은 Length 가 0 이므로 data를 위한 물리적 공간을 차지 하지
    않는다.
79
80     SELECT empno, job, comm FROM emp;
81     --NULL 인 값은 비어있는 것으로 표현된다. job이 salesman인 직원들에게만 커미션이
    적용되며, 사번 7844인 직원의 커미션은 0이다.
82
83     SELECT empno, ename, sal * 12 + comm
84     FROM emp;
85     --comm 값이 NULL 인 경우 연봉은 얼마인가? 연봉 계산한 수식의 column heading은
    어떻게 나타나는가?
```

```

86      --comm 값이 NULL 인 row 의 경우 (sal + comm) * 12를 하면 결과도 모두 NULL 이
      된다.
87      --또한, expression 전체가 column heading 으로 나타난다.
88
89  REM NVL function
90  1. NULL 값을 어떤 특정한 값으로 치환할 때 사용
91  2. 치환할 수 있는 값의 형태는 숫자형, 문자형, 날짜형 모두 가능
92  3. 치환된 값은 expr1컬럼의 데이터 타입과 일치해야 한다.
93  4. Syntax
94      NVL(expr1, expr2)
95      --expr1 : NULL
96      --expr2 : 치환값
97      --expr1값이 NULL 아니면 expr1 값을 그대로 사용
98      --만약 expr1 값이 NULL이면, expr2 값으로 대체
99  5. 예
100     NVL(comm, 0)
101     NVL(hiredate, '12/09/04')
102     NVL(job, 'No Job')
103
104     --위에서 연봉을 구하는 Query 를 NVL 함수를 사용하여 제대로 나올 수 있도록 고쳐보
105     자.
106     SELECT empno, ename, sal, comm, sal * 12 + NVL(comm, 0)
107     FROM emp;
108
109     SELECT empno, comm, NVL(comm, 0)
110     FROM emp;
111
112     --다음은 매니저가 없는, 즉 최고 직급의 직원인 경우 'No Manager'라고 출력되도록 하
113     는 문장이다. 실행하여 메시지를 적어보고, Error 가 나는 이유를 설명하시오.
114     SQL> SELECT NVL(mgr, 'No Manager') FROM emp;
115
116  REM Alias 별칭
117  1. column header 에 별칭을 부여 할 수 있다.
118  2. SELECT 절에 expression 을 사용할 때 도움이 된다.
119  3. 열 이름 바로 뒤에 기술한다. 또는 열이름과 별칭 사이에 AS를 사용할 수 있다.
120  4. 별칭에 공백이나 특수문자나 한글사용할 때, 대소문자를 기술할 때(기본값은 모두 대문자)에는 ""
121     로 기술한다.
122
123     SELECT empno 사번 FROM emp;
124     SELECT sal * 12 연봉 FROM emp;
125     SELECT sal * 12 annual_salary FROM emp;
126     SELECT sal * 12 Annual_Salary FROM emp;
127     SELECT sal * 12 Annual Salary FROM emp; --Error 발생
128     SELECT ename "Name" , sal AS "Salary", sal * 12 + NVL(comm, 0) AS
129     "Annual Salary"
130     FROM emp;

```

```
128 REM Concatenation Operator (연결 연산자)
129 1. 문자열 리터럴을 이을 때에는 '||' 를 사용한다.
130 2. 연결연산자(||)는 character string 들을 여결하여 하나의 결과 string 을 만들어 낸다.
131
132     SELECT empno || ename FROM emp;
133     SELECT empno || ename || hiredate FROM emp;
134     --number 이나 date값은 default 형태의 character 값으로 자동 변환한 후 연결된
135     다.
136 REM Literals (상수)
137 1. Literal 은 상수 값을 의미.
138 2. Character literal 은 작은 따옴표로 묶고, Number literal 은 따옴표 없이 표현한다.
139 3. Character literal을 작은 따옴표로 묶어 주어야 Oracle Server 는 keyword나 객체 이름
140     을 구별할 수 있다.
141
142     SELECT 'Emp# of ' || ename || ' is ' empno FROM emp;
143     SELECT dname || ' is located at ' || loc FROM dept;
144     SELECT ename || ' is a ' || job AS "Employee" FROM emp;
145     SELECT ename || ' ' || sal FROM emp;
146     SELECT ename || ' is working as a ' || job FROM emp;
147     SELECT 'Java is a language.' FROM emp; --14번 출력
148     SELECT 'Java is a language.' FROM dept; --4번 출력
149     SELECT 'Java is a language.' FROM dual;
150 REM Duplicate Values(중복 행 제거하기)
151 1. 일반 Query는 ALL 을 사용하기 때문에 중복된 행이 출력된다.
152 2. DISTINCT 를 사용하면 중복된 행의 값을 제거한다.
153 3. DISTINCT 는 SELECT 바로 뒤에 기술한다.
154 4. DISTINCT 다음에 나타나는 column은 모두 DISTINCT 에 영향을 받는다.
155
156     SELECT job FROM emp;
157     SELECT ALL job FROM emp;
158     SELECT DISTINCT job FROM emp;
159     SELECT deptno FROM emp;
160     SELECT DISTINCT deptno FROM emp;
161     SELECT deptno, job FROM emp;
162     SELECT DISTINCT deptno, job FROM emp;
163
164 REM WHERE 절
165 1. Syntax
166
167     SELECT column...
168     FROM table_name
169     WHERE conditions;
170
171 2. WHERE 절을 사용하지 않으면 FROM 절에 명시된 table의 모든 row를 조회하게 된다.
172 3. table내의 특정 row만 선택하고 싶을 때 WHERE 절에 조건식을 사용한다.
```

```
173 4. Oracle Server 는 table의 row를 하나씩 읽어 WHERE 절의 조건식을 평가하여 TRUE로 만족
    하는 것만을 선택한다.
174 5. condition을 평가한 결과는 TRUE, FALSE, NULL 중의 하나이다.
175 6. condition 내에서 character 와 date 값의 literal은 작은 따옴표를 사용하고, number
    값은 그대로 사용한다.
176 7. condition 에서 사용하는 character 값은 대소문자를 구별한다.
177     1) WHERE ename = 'JAMES';
178     2) WHERE ename = 'james';
179 8. date 타입은 현재 session 의 NLS_DATE_FORMAT 에 맞춰 표현한다.
180 9. WHERE 는 FROM 다음에 와야 한다.
181
182 /*
183     날짜 형식 바꾸기
184     ALTER SESSION SET NLS_DATE_FORMAT='YYYY-MM-DD';
185     SELECT value FROM NLS_SESSION_PARAMETERS
186     WHERE parameter = 'NLS_DATE_FORMAT';
187 */
188
189 REM 비교연산자
190 --<, >, <=, >=, =, !=, <>(같지 않다)
191
192     --직위가 CLERK 인 사원의 이름과 직위 및 부서번호를 출력하시오.
193     SELECT ename, job, deptno
194     FROM emp
195     WHERE job = 'CLERK';
196
197     SELECT empno, ename, job
198     FROM emp
199     WHERE empno = 7934;
200
201     SELECT empno, ename, job, hiredate
202     FROM emp
203     WHERE hiredate = '1981-12-03';
204
205     SELECT empno, ename
206     FROM emp
207     WHERE ename = 'JAMES';
208
209     SELECT empno, ename
210     FROM emp
211     WHERE ename = 'james';
212
213     SELECT dname
214     FROM dept
215     WHERE deptno = 30;
216
217     SELECT ename, sal
```

```
218      FROM emp
219      WHERE sal >= 1500;
220
221      --1983년 이후에 입사한 사원의 사번, 이름, 입사일을 출력하시오.
222      SELECT empno, ename, hiredate
223      FROM emp
224      WHERE hiredate >= '1983-01-01';      //01-JAN-83
225
226      --급여가 보너스(comm) 이하인 사원의 이름, 급여 및 보너스를 출력하시오
227      SELECT ename, sal, comm
228      FROM emp
229      WHERE sal <= NVL(comm, 0);
230
231      --10번 부서의 모든 사람들에게 급여의 13%를 보너스로 지급하기로 했다. 이름, 급여, 보
232      --너스 금액, 부서번호를 출력하시오.
233      SELECT ename, sal, sal * 0.13, deptno
234      FROM emp
235      WHERE deptno = 10;
236
237      --30번 부서의 연봉을 계산하여, 이름, 부서번호, 급여, 연봉을 출력하라. 단, 연말에 급
238      --여의 150%를 보너스로 지급한다.
239      SELECT ename, deptno, sal, sal * 12 + NVL(comm, 0) + sal * 1.5 AS "년
240      봉"
241      FROM emp
242      WHERE deptno = 30;
243
244      --부서번호가 20인 부서의 시간당 임금을 계산하시오. 단, 1달의 근무일수는 12일이고, 1
245      --일 근무시간은 5시간이다. 출력양식은 이름, 급여, 시간당 임금을 출력하라.
246      SELECT ename, sal, sal / 12 / 5
247      FROM emp
248      WHERE deptno = 20;
249
250      --모든 사원의 실수령액을 계산하여 출력하시오. 단, 이름, 급여, 실수령액을 출력하시오.
251      --(실수령액은 급여에 대해 10%의 세금을 뺀 금액)
252      SELECT ename, sal, sal - sal * 0.1 AS "실수령액"
253      FROM emp;
254
255      --사번이 7788인 사원의 이름과 급여를 출력하시오.
256      --급여가 3000이 넘는 직종을 선택하시오.
257      --PRESIDENT를 제외한 직원들의 이름과 직종을 출력하시오.
258      --BOSTON 지역에 있는 부서이 번호와 이름을 출력하시오.
```

```
259      SELECT empno, ename, job, sal, deptno
260      FROM emp
261      WHERE sal >= 1000 AND deptno = 30;
262
263      --사원테이블에서 급여가 2000불이상이거나 담당업무가 매니저인 사원의 정보중 사원번호,
264      이름, 급여, 업무를 출력하시오.
265      SELECT empno, ename, sal, job
266      FROM emp
267      WHERE sal >= 2000 OR job = 'MANAGER';
268
269  REM SQL 연산자
270  1. BETWEEN A AND B : A보다 같거나 크고, B보다 작거나 같은
271  2. IN(list) : list 안에 있는 멤버들과 같은
272  3. ANY(list) : list에 있는 어느 한 멤버와 값을 비교, 반드시 =, !=, <, >, <=, >= 등과
273     함께 사용한다.
274  4. ALL(list) : list에 있는 모든 멤버와 값을 비교, 반드시 =, !=, <, >, <=, >= 등과 함
275     께 사용한다.
276  5. A LIKE B [ESCAPE 'C']: A가 B의 패턴과 일치하면 TRUE, 보통 %, _ 연산자와 같이 사용,
277     escape 을 사용하면 B의 패턴 중에서 C를 상수로 취급한다.
278  6. IS NULL / IS NOT NULL : NULL 여부를 테스트
279
280  1) BETWEEN A AND B
281      --사원테이블에서 월급이 1300불에서 1500불까지의 사원정보중 성명, 담당업무, 월급을
282      출력하시오.
283      SELECT ename, job, sal
284      FROM emp
285      WHERE sal >= 1300 AND sal <= 1500;
286      WHERE sal BETWEEN 1300 AND 1500;
287
288      SELECT ename, job, sal
289      FROM emp
290      WHERE sal BETWEEN 1500 AND 1300;
291      --반드시 작은 값이 먼저 나와야 한다.
292
293      SELECT ename FROM emp
294      WHERE hiredate BETWEEN '1982-01-01' AND '1982-12-31';
295
296      --급여가 2000 에서 3000 사이인 사원을 출력하시오.
297      SELECT ename, job, sal FROM emp
298      WHERE sal BETWEEN 2000 AND 3000;
299
300  2) IN
301      --사원테이블에서 업무가 회사원, 매니저, 분석가인 사원의 이름, 업무를 출력하시오.
302      SELECT ename AS "이름", job
303      FROM emp
304      WHERE job = 'CLERK' OR job = 'MANAGER' OR job = 'ANALYST';
305      WHERE job IN('CLERK', 'MANAGER', 'ANALYST');
```

```
301
302      --관리자의 직원번호가 7902, 7566, 7788인 모든 사원의 직원번호, 이름, 급여 및 관
리자의 직원번호를 출력하시오.
303
304      SELECT dname FROM emp WHERE deptno IN(10,20);
305
306      --ANY와 ALL연산자의 앞에는 비교연산자가 반드시 함께 사용되어야 한다.
307      --IN 연산자는 =ANY와 같다.
308      --NOT IN 연산자는 <>ALL 연산자와 같은 결과이다.
309
310      SELECT ename FROM emp
311      WHERE job IN('ANALYST', 'CLERK')
312      WHERE job =ANY('ANALYST', 'CLERK');
313
314      SELECT ename FROM emp
315      WHERE sal <ALL(2000, 3000);
316
317      --BOSTON 이나 DALLAS 에 위치한 부서를 출력하시오.
318      SELECT dname, loc FROM dept
319      WHERE loc IN('BOSTON', 'DALLAS');
320      WHERE loc =ANY('BOSTON', 'DALLAS');
321
322      --30, 40번 부서에 속하지 않는 직원들을 출력하시오.
323      SELECT ename, deptno FROM emp
324      WHERE deptno NOT IN(30,40);
325      WHERE deptno <>ALL(30,40);
326
327      --DALLAS 의 20번 부서, 또는 CHICAGO의 30번 부서를 출력하시오.
328      SELECT * FROM dept
329      WHERE (deptno, loc) IN (20, 'DALLAS'), (30, 'CHICAGO'));
330
331 3) LIKE(%, _)
332      --Wildcard : %(0개 이상의 문자 대표), _ (1개의 문자 대표)
333      --Wildcard 문자를 일반 문자로 사용하고 싶을 때 ESCAPE 을 사용한다.
334      --ESCAPE 문자 바로 뒤에 사용된 Wildcard 문자는 일반 문자로 취급한다.
335
336      SELECT ename, job, hiredate FROM emp
337      WHERE hiredate LIKE '87%';
338      WHERE hiredate >= '87/01/01';
339
340      SELECT dname FROM dept
341      WHERE dname LIKE 'A%';
342
343      --이름이 A로 시작하는 직원을 출력하시오.
344      SELECT ename FROM emp
345      WHERE ename LIKE 'A%';
346
```



```
347      --사번이 8번으로 끝나는 사원을 출력하시오.
348      SELECT empno, ename FROM emp
349      WHERE empno LIKE '%8';
350
351      --1982에 입사한 사원을 출력하시오.
352      SELECT ename, hiredate FROM emp
353      WHERE hiredate LIKE '1982%';
354      WHERE hiredate BETWEEN '1982-01-01' AND '1982-12-31';
355      WHERE hiredate >= '1982-01-01' AND hiredate <= '1982-12-31';
356
357      SELECT empno, ename
358      FROM emp
359      WHERE ename LIKE 'MILLE_';
360
361      SELECT empno, ename
362      FROM emp
363      WHERE ename LIKE '%$_TEST' ESCAPE '$';
364
365      4) IS NULL / IS NOT NULL
366      --column 의 NULL 여부를 판단할 때에는 반드시 'IS NULL' 혹은 'IS NOT NULL' 연
      산자를 사용한다.
367      SELECT ename FROM emp
368      WHERE comm IS NULL;
369      WHERE comm IS NOT NULL;
370
371      --comm 지급 대상인 사원을 출력하시오.
372      SELECT ename, comm FROM emp
373      WHERE comm IS NOT NULL;
374
375      SELECT ename, mgr
376      FROM emp
377      WHERE mgr IS NULL;
378
379      REM 연산자 우선순위
380      1. +, -, 괄호 ()
381      2. *, / : 산술연산자
382      3. +, - : 산술연산자, ||
383      4. =, <>, <, >, <=, >=, IS NULL, LIKE, BETWEEN, IN
384      5. NOT
385      6. AND
386      7. OR
387
388      SELECT ename, job FROM emp
389      WHERE NOT job = 'ANALYST';
390
391      SELECT ename, sal, deptno FROM emp
392      WHERE sal > 2500 AND deptno = 20;
```

```
393
394     SELECT deptno, dname FROM dept
395     WHERE deptno = 10 OR deptno = 20;
396
397     --업무가 SALESMAN 이거나 업무가 MANAGER 이고, 급여가 1300불이상인 사람의 사원번호, 이름, 업무, 급여를 출력하시오.
398     SELECT empno, ename, job, sal
399     FROM emp
400     WHERE (job LIKE 'S%' OR job LIKE 'M%') AND sal >= 1300;
401
402     --직종이 CLERK 인 사원 중에서 급여가 1000 이상인 사원을 출력하시오.
403     SELECT ename, job, sal FROM emp
404     WHERE job = 'CLERK' AND sal >= 1000;
405
406 REM ORDER BY
407 1. 기본적으로 오라클의 데이터는 정렬되지 않는다.
408 2. 같은 쿼리를 수행할 때마다 결과가 다르게 나올 수 있다.
409 3. 별칭을 사용할 수 있다.
410 4. Syntax
411
412     SELECT column_list
413     FROM table
414     [WHERE conditions]
415     [ORDER BY column[, column] {ASC | DESC}];
416
417 5. 특징
418     1) 기본적으로 오름차순정렬한다.
419         --숫자인경우 ( 1 --> 999)
420         --날짜인경우 (옛날 --> 최근)
421         --문자인경우 (알파벳순서, 유니코드순)
422     2) NULL 은 오름차순일 경우는 제일 마지막에, 내림차순인 경우에는 제일 처음에 출력
423 6. ORDER BY 절에 정렬의 기준이 되는 column 을 여러개 지정할 수 있다. 첫번째 column 으로 정렬한 다음, 그 column 값이 같은 row 들에 대해서는 두 번째 column 값으로 정렬한다.
424 7. 오름차순(ASC) 정렬이 기본이며, 내림차순으로 정렬하고자 할 때에는 DESC를 사용한다.
425 8. ORDER BY 절에 column 이름 대신 positional notation 을 사용할 수도 있다. Position number 는 SELECT 절에서의 column 순서를 의미한다.
426
427     --입사일자 순으로 정렬하여 사원번호, 이름, 입사일자를 출력하시오.
428     SELECT empno, ename, hiredate
429     FROM emp
430     ORDER BY hiredate DESC;
431
432     --부서번호가 20번인 사원의 연봉 오름차순으로 출력하시오.
433     SELECT empno, ename, sal, comm, sal * 12 + NVL(comm, 0) AS "Annual"
434     FROM emp WHERE deptno = 20
435     ORDER BY "Annual" ASC;
436
```

```
437      --부서번호로 정렬한 후, 부서번호가 같을 경우 급여가 많은 순으로 직원번호, 직원이름,  
438      업무, 부서번호, 급여를 출력하시오.  
439      SELECT empno, ename, job, deptno, sal  
440      FROM emp  
441      ORDER BY deptno ASC, sal DESC;
```