**1. 기본적인 SELECT문 사용법**

SELECT문장을 이용하여 데이터베이스로부터 저장되어 있는 데이터를 검색하는 방법에 대하여 알아보기로 한다.

**1.1 SQL SELECT 문장의 성능**

1. Selection : 질의에 대해 RETURN하고자 하는 테이블의 행을 선택하기 위해 SQL의 Selection기능을 사용할 수 있습니다.
2. Projection : 질의에 대해 RETURN하고자 하는 테이블의 열을 선택하기 위해 SQL의 Projection 기능을 사용할 수 있습니다.
3. Join : 공유 테이블 양쪽의 열에 대해 링크를 생성하여 다른 테이블에 저장되어 있는 데이터를 함께 가져오기 위해 SQL의 join 기능을 사용할 수 있습니다.

**1.2 Syntax**

|  |
| --- |
| **SELECT** [**DISTINCT**] {\*, column [alias], . . .}  **FROM** table\_name  [**WHERE** condition]  [**ORDER BY** {column, expression} [**ASC** | **DESC**]]; |

DISTINCT 중복 행 제거 옵션

\* 테이블의 모든 column 출력

alias 해당 column에 대한 다른 이름 부여

table\_name 테이블명 질의 대상 테이블 이름

WHERE 조건을 만족하는 행들만 검색

condition column, 표현식, 상수 및 비교 연산자

ORDER BY 질의 결과 정렬을 위한 옵션(ASC:오름차순(Default),DESC내림차순)

**1.3 SQL 문장 작성법**

1. SQL 문장은 대소문자를 구별하지 않습니다.
2. SQL 문장은 한 줄 또는 여러 줄에 입력될 수 있습니다.
3. 하나의 명령어는 여러 줄에 나누거나 단축될 수 없습니다.
4. 절은 보통 읽고 편집하기 쉽게 줄을 나누도록 합니다.(권장)
5. 탭과 줄 넣기(들여쓰기)는 코드를 보다 읽기 쉽게 하기 위해 사용됩니다.(권장)
6. 일반적으로 키워드는 대문자로 입력합니다. 다른 모든 단어, 즉 테이블 이름, 열 이름은 소문자로 입력합니다.(권장)
7. SQL\*Plus에서SQL 문장은 SQL 프롬프트에 입력되며 1라인 이후의 라인은 라인 번호가 붙습니다. 가장 최근의 명령어가 1개가 SQL buffer에 저장됩니다.

**1.4 SQL 문장 실행**

1. 마지막 절의 끝에 “;”를 기술하여 명령의 끝을 표시한다.
2. 버퍼에서 마지막 라인에 슬래시를 넣습니다.(OS의 Editor사용시)
3. SQL프롬프트에 슬래시를 입력합니다.(SQL Buffer의 내용 실행)
4. SQL프롬프트에서 SQL\*Plus RUN 명령어를 실행합니다. (SQL Buffer의 내용 실행)

**1.5 모든 열 선택**

SELECT 키워드에 “\*“ 을 사용하여 테이블의 열 데이터 모두를 조회할 수 있습니다.

문제1) SCOTT이 소유하고 있는 EMP Table의 모든 자료를 출력하여라.

|  |
| --- |
| SQL> **SELECT \***  2 **FROM emp;**  EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO  ------ ---------- --------- --------- ---------- --------- --------- -------  7839 KING PRESIDENT 17-NOV-81 5000 10  7698 BLAKE MANAGER 7839 01-MAY-81 2850 30  7782 CLARK MANAGER 7839 09-JUN-81 2450 10  7566 JONES MANAGER 7839 02-APR-81 2975 20  . . . . . . . . . .  14 rows selected. |

**1.6 특정 Column 선택**

테이블의 특정 Column을 검색하고자 할 경우 Column이름을 “,”로 구분하여 명시함으로써 특정 Column을 출력할 수 있습니다. 출력 순서는 SELECT문 뒤에 기술한 Column의 순서대로 출력됩니다.

문제2) SCOTT이 소유하고 있는 EMP Table에서 사원 번호, 이름, 급여, 담당업무를 출력하여라.

|  |
| --- |
| SQL> **DESC emp**  Name Null? Type  ------------------------------- -------- ----  EMPNO NOT NULL NUMBER(4)  ENAME VARCHAR2(10)  JOB VARCHAR2(9)  MGR NUMBER(4)  HIREDATE DATE  SAL NUMBER(7,2)  COMM NUMBER(7,2)  DEPTNO NOT NULL NUMBER(2) |
| SQL> **SELECT empno,ename,sal,job**  2 **FROM emp;**  EMPNO ENAME SAL JOB  --------- ---------- --------- ---------  7839 KING 5000 PRESIDENT  7698 BLAKE 2850 MANAGER  7782 CLARK 2450 MANAGER  . . . . . . . . . .  14 rows selected. |

**1.7 Column의 출력 형태**

날짜 열 헤딩과 데이터 뿐만 아니라 문자 열 헤딩과 데이터는 열 폭 내에서 좌측 정렬됩니다. 숫자 헤딩과 데이터는 우측 정렬입니다.

|  |
| --- |
| SQL> **SELECT empno,ename,hiredate**  2 **FROM emp;**  EMPNO ENAME HIREDATE  --------- ---------- -----------------  7839 KING 17-NOV-81  7698 BLAKE 01-MAY-81  . . . . . . . . . .  14 rows selected. |

**1.8 산술 표현식**

데이터가 출력 되는 방식을 수정하거나 계산을 수행하고자 할 때 산술 표현식을 사용한다. 산술 표현식은 열 이름, 숫자 상수, 문자 상수, 산술 연산자를 포함할 수 있으며 연산자는 +(Add), -(Subtract), \*(Multiply), /(Divide)을 사용합니다. SELECT문장에서는 FROM절을 제외한 SQL문장의 절에서 사용할 수 있습니다. 또한 산술 표현식이 하나 이상의 연산자를 포함한다면 일반적인 산술 연산자 우선 순위를 따른다.

문제3) 모든 종업원의 급여를 $300증가 시키기 위해 덧셈 연산자를 사용하고 결과에 SAL+300을 디스플레이 합니다.

|  |
| --- |
| SQL> **SELECT ename, sal, sal+300**  2 **FROM emp;**  ENAME SAL SAL+300  ---------- --------- ---------  KING 5000 5300  BLAKE 2850 3150  . . . . . . . . . . .  14 rows selected. |

♣ 참고

계산된 결과 열SAL+300은 EMP테이블의 새로운 열이 아님을 유의하십시오. 이것은 단지 디스플레이를 위한 것일 뿐입니다. 디폴트로 새로운 열의 이름 sal+300은 생성된 계산식으로부터 유래합니다. 또한 SQL\*Plus는 산술 연산자 앞뒤의 공백을 무시합니다.

**1.9 Null값의 처리**

행이 특정 열에 대한 데이터 값이 없다면, 값은 null이 됩니다. null값은 이용할 수 없거나 지정되지 않았거나, 알 수 없거나 또는 적용할 수 없는 값입니다. null값은 0이나 공백과는 다릅니다. 0은 숫자이며 공백은 문자입니다. 열이 NOT NULL로 정의되지 않았거나, 열이 생성될 때PRIMARY KEY로 정의되지 않았다면, 어떤 데이터형의 열은 null값을 포함할 수 있습니다. EMP 테이블의 COMM열에서 오직 SALESMAN만이 보너스를 받을 수 있음을 주목하십시오.

♣ 참고

1. NULL은 이용할 수 없고 할당되지 않고 알려져 있지않고 적용 불가한 값을 의미한다.
2. NULL이란 0나 공백(space)과 다르다.
3. 널 값을 포함한 산술 표현식 결과는 NULL이 된다.
4. column에 데이터 값이 없으면 그 값 자체가 널 또는 널 값을 포함하고 있다.
5. 널 값은 1바이트의 내부 저장 장치를 오버헤드로 사용하고 있으며 어떠한 datatype column들이라도 널 값을 포함할 수 있다.

문제4) EMP 테이블에서 사원번호, 이름, 급여, 보너스, 보너스 금액을 출력하여라

|  |
| --- |
| SQL> **SELECT empno,ename,sal,comm,sal+comm/100**  2 **FROM emp;**  EMPNO ENAME SAL COMM SAL+COMM/100  --------- ---------- --------- --------- ------------  7839 KING 5000  7698 BLAKE 2850  7782 CLARK 2450  7566 JONES 2975  7654 MARTIN 1250 1400 1264  7499 ALLEN 1600 300 1603  7844 TURNER 1500 0 1500  . . . . . . . . . .  14 rows selected. |

**1.10 NVL 함수**

1. Null값을 어떤 특정한 값(실제 값)으로 변환하는데 사용한다.
2. 사용될 수 있는 데이터 타입은 날짜, 문자, 숫자입니다.
3. NVL 함수를 사용할 때 전환되는 값의 데이터 타입을 일치 시켜야 한다.

**1.10.1 Syntax**

|  |
| --- |
| **NVL(**expr1,expr2**)** |

expr1 Null 값을 포함하고 있는 Column이나 표현식

expr2 Null 변환을 위한 목표 값

**1.10.2 다양한 데이터형에 대한 NVL변형**

|  |  |
| --- | --- |
| 데이터형 | 변환 예 |
| NUMBER | NVL(comm, 0) |
| DATE | NVL(hiredate, ‘01-JAN-99’) |
| CHAR or VARCHAR2 | NVL(job, ‘업무없음’) |

문제5) EMP 테이블에서 이름, 급여, 보너스, 연봉을 출력하여라

|  |
| --- |
| SQL> **SELECT ename,sal,comm,sal\*12+NVL(comm,0)**  2 **FROM emp;**  ENAME SAL COMM SAL\*12+NVL(COMM,0)  ---------- --------- --------- ------------------  . . . . . . . . . .  JONES 2975 35700  MARTIN 1250 1400 16400  ALLEN 1600 300 19500  TURNER 1500 0 18000  . . . . . . . . . .  14 rows selected. |

**1.11 열에 별칭(Alias) 부여**

질의의 결과를 출력할 때 보통 SQL\*Plus는 열 Heading으로 선택된 열 이름을 사용합니다. 이 Heading은 때로 사용자가 이해하기가 어려운 경우가 있기 때문에 열 Heading을 변경하여 질의 결과를 출력하면 보다 쉽게 사용자가 이해할 수 있습니다.

**1.11.1 열 별칭(Alias) 정의**

1. 열 Heading이름을 변경 합니다.
2. 계산에 유용합니다.
3. 열 이름 바로 뒤에 사용합니다. 열 이름과 별칭 사이에 키워드 AS를 넣기도 합니다.
4. 공백이나 특수 문자 또는 대문자가 있으면 이중 인용부호(“ ”)가 필요 합니다.

문제6) EMP 테이블에서 ENAME를 NAME로 SAL을 SALARY로 출력하여라.

|  |
| --- |
| SQL> **SELECT ename AS name, sal salary**  2 **FROM emp;**  NAME SALARY  ---------- ---------  KING 5000  BLAKE 2850  CLARK 2450  . . . . . . . . . .  14 rows selected. |

문제7) EMP 테이블에서 ENAME를 Name로 SAL\*12를 Annual Salary로 출력하여라

|  |
| --- |
| SQL> **SELECT ename "Name", sal\*12 "Annual Salary"**  2 **FROM emp;**  Name Annual Salary  ---------- -------------  KING 60000  BLAKE 34200  . . . . . . . . . .  14 rows selected. |

문제8) EMP 테이블에서 ENAME를 Name로 SAL\*12를 Annual Salary로 출력하여라

|  |
| --- |
| SQL> **SELECT ename "성 명", sal "급 여"**  2  **FROM emp;**  성 명 급 여  ---------- ---------  KING 5000  BLAKE 2850  CLARK 2450  JONES 2975  . . . . . . . . . .  14 rows selected. |

**1.12 연결 연산자**

연결 연산자(||)를 사용하여 문자 표현식을 생성하기 위해 다른 열, 산술 표현식, 상수 값에 열을 연결 할 수 있습니다. 연결자의 왼쪽에 있는 열은 단일 결과 열을 만들기 위해 조합 됩니다.

1. 열이나 문자 STRING을 다른 열에 연결 합니다.
2. 두개의 “||”로 연결 합니다.
3. 문자 표현식의 결과 열을 생성 합니다.

문제9) EMP 테이블에서 이름과 업무를 연결하여 출력하여라.

|  |
| --- |
| SQL> **SELECT ename || ' ' || job AS "Employees"**  2 **FROM emp;**  Employees  --------------------  KING PRESIDENT  BLAKE MANAGER  . . . . . . . . . .  14 rows selected. |

**1.13 LITERAL 문자 STRING**

LITERAL은 열 이름이나 열 별칭이 아닌 SELECT목록에 포함되어 있는 문자, 표현식, 숫자입니다. 그것은 RETURN되는 각각의 행에 대해 출력됩니다. LITERAL과 STRING은 질의 결과에 포함될 수 있으며 SELECT목록에서 열과 똑같이 취급됩니다. 날짜와 문자 LITERAL은 단일 인용 부호(‘ ‘)를 사용하여야 하고 숫자 LITERAL은 사용하지 않습니다.

1. SELECT절에 포함된 LITERAL은 문자, 표현식, 숫자입니다.
2. 날짜와 문자 LITERAL 값은 단일 인용부호(‘ ‘) 안에 있어야 합니다.
3. 각각의 문자 STRING은 RETURN된 각 행에 대한 결과입니다.

문제10) EMP 테이블에서 이름과 업무를 “KING is a PRESIDENT” 형식으로 출력하여라.

|  |
| --- |
| SQL> **SELECT ename || ' ' || 'is a' || ' ' || job AS "Employees Details"**  2 **FROM emp;**  Employees Details  -------------------------  KING is a PRESIDENT  BLAKE is a MANAGER  . . . . . . . . . .  14 rows selected. |

문제11) EMP 테이블에서 이름과 연봉을 “KING: 1 Year salary = 60000” 형식으로 출력하여라.

|  |
| --- |
| SQL> **SELECT ename || ': 1 Year salary = ' || sal \* 12 Monthly**  2 **FROM emp;**  MONTHLY  --------------------------------------------------------------------  KING: 1 Year salary = 60000  . . . . . . . . . .  14 rows selected. |

**1.14 중복 행의 제거**

특별히 명시되지 않았다면, SQL\*Plus는 중복되지는 행을 제거하지 않고 Query 결과를 출력합니다. 결과에서 중복되는 행을 제거하기 위해서는 SELECT 키워드 바로 뒤에 DISTINCT를 기술한다.

문제12) EMP 테이블에서 JOB을 모두 출력하여하

|  |
| --- |
| SQL> **SELECT job**  2 **FROM emp;**  JOB  ---------  PRESIDENT  MANAGER  MANAGER  MANAGER  SALESMAN  SALESMAN  SALESMAN  CLERK  SALESMAN  ANALYST  CLERK  ANALYST  CLERK  CLERK  14 rows selected. |

문제13) EMP 테이블에서 담당하고 있는 업무의 종류를 출력하여라.

|  |
| --- |
| SQL> **SELECT DISTINCT job**  2 **FROM emp;**  JOB  ---------  ANALYST  CLERK  MANAGER  PRESIDENT  SALESMAN |

문제14) 부서별로 담당하는 업무를 한번씩 출력하여라.

|  |
| --- |
| SQL> **SELECT DISTINCT deptno,job**  2 **FROM emp;**  DEPTNO JOB  --------- ---------  10 CLERK  10 MANAGER  10 PRESIDENT  20 ANALYST  20 CLERK  20 MANAGER  30 CLERK  30 MANAGER  30 SALESMAN  9 rows selected. |

☞ Guidelines

1. DISTINCT라는 키워드는 항상 SELECT 바로 다음에 기술한다.
2. DISTINCT뒤에 나타나는 칼럼들은 모두 DISTINCT의 영향을 받는다.
3. DISTINCT뒤에 여러 개의 칼럼을 기술하였을 때 나타나는 행은 칼럼의 조합들이 중복되지 않게 나타난다.
4. DISTINCT를 사용하여 나타나는 결과는 기본적으로 오름차순 정렬된다.

**◈ 연 습 문 제 ◈**

1. 아래의 SELECT 문장이 성공적으로 수행 될까요? (참 / 거짓)

|  |
| --- |
| SQL> SELECT ename 이름,job 업무,sal 급여  2 FROM emp; |

2. 아래의 SELECT 문장이 성공적으로 수행 될까요? (참 / 거짓)

|  |
| --- |
| SQL> SELECT \*  2 FROM salgrade; |

3. 이 문장에 에러가 있습니다. 올바르게 작성하시오.

|  |
| --- |
| SQL> SELECT empno,ename,sal X 12 년 봉  2 FROM emp; |

4. EMP 테이블의 구조와 내용을 조회하여라.

5. EMP 테이블에서 중복되지 않는 부서번호를 출력하시오.

6. EMP 테이블의 이름과 업무를 연결하여 출력하여라.

7. DEPT 테이블의 부서명과 위치를 연결하여 출력하여라.

8. EMP 테이블의 업무와 급여를 연결하여 출력하여라.

9. 6,7,8번의 결과를 부석하여 설명하여라.