

2장 SQL의 기본

2.1 데이터 디렉터리 TAB

- 오라클을 설치하면 제공되는 사용자인 SCOTT은 학습을 위해서 테이블들이 제공된다.
- TAB은 TABLE의 약자로서 SCOTT 사용자가 소유하고 있는 테이블의 정보를 알려주는 데이터 디렉터리이다.

```
SQLPLUS SCOTT/TIGER
SELECT * FROM TAB;
```

2.2 테이블 구조를 살펴보기 위한 DESC

- 테이블에서 데이터를 조회하기 위해서는 테이블의 구조를 알아야 한다. 테이블의 구조를 확인하기 위한 명령어로는 DESCRIBE가 있다.
- DESC 명령어는 테이블의 컬럼 이름, 데이터 형, 길이와 NULL 허용 유무 등과 같은 특정 테이블의 정보를 알려준다.
- NOT NULL 제약 조건
 - NULL: 컬럼에 어떠한 값도 정해지지 않을 때 갖는 값이다. NULL은 할당받지 않은 값, 모르는 값, 정해지지 않을 값을 의미하여 0(숫자의 한 자리)이나, 스페이스(문자의 한 자리)와는 다른 값이다.
 - NOT NULL: 이 조건이 설정된 컬럼은 반드시 NULL값이 아닌 **확실한 정보가 필수적으로 입력되어야** 한다.
- 기본 키(Primary key) 제약 조건
 - 유일한 값을 저장하기 위한 조건으로 DESC 명령으로 확인할 수 없다.
- 외래 키(Foreign key) 제약 조건
 - 관계형 데이터베이스는 테이블 사이의 관계를 설정할 수 있다.
 - 다른 테이블의 컬럼을 참조하도록 연결하여 제약을 걸어 두기 위해서 사용한다.

```
-- DESC[RIBE] 테이블명
DESC DEPT
DESC EMP
```

2.3 오라클의 데이터형

2.3.1 NUMBER

- NUMBER 데이터 형은 **숫자 데이터를 저장하기** 위해서 제공된다.
- precision은 소수점을 포함한 전체 자리수를 의미하며 scale은 소수점 이하 자리수를 지정한다.

- scale을 생략한 채 precision만 지정하면 소수점 이하는 반올림되어 정수 값만 저장된다.
- precision과 scale을 모두 생략하면 입력한 데이터 값만큼 공간이 할당된다.

```
-- 형식
NUMBER(PRECISION, SCALE)

-- 예
NUMBER(4) -- 정수로 최대 4자리
NUMBER(8,2) -- 전체 8자리, 소수 이하 2자리의 실수
```

2.3.2 DATE

- DATE는 세기, 년, 월, 일, 시간, 분, 초의 날짜 및 시간 데이터를 저장하기 위한 데이터 형이다.
- 기본 날짜 형식은 "YY/MM/DD" 형식으로 "년/월/일"로 출력된다. 2005년 12월 14일은 "05/12/14"로 출력된다.

2.3.3 CHAR

- 문자 데이터를 저장하기 위한 자료형으로 CHAR가 있다. CHAR는 고정 길이 문자 데이터를 저장한다.
- 입력된 자료의 길이와는 상관없이 정해진 길이만큼 저장 영역 차지하며 최소 크기는 1이다.
- char(20)이라고 설정하고 'seoul'이라는 데이터를 저장하였다면 ...



- CHAR는 주어진 크기만큼 저장공간이 할당되므로 편차가 심한 데이터를 입력할 경우 위의 예와 같이 저장공간의 낭비를 초래한다.

2.3.4 VARCHAR2

- VARCHAR2 데이터 형은 가변적인 길이의 문자열을 저장하기 위해서 제공한다.
- address 란 컬럼의 데이터형을 VARCHAR2(20)이라고 설정하고, 'seoul' 이란 데이터를 저장하였다면



- VARCHAR2는 저장되는 데이터에 의해서 저장공간이 할당되므로 메모리 낭비를 줄일 수 있습니다.

[꿀팁] Boolean 타입

- 오라클 데이터베이스에서는 불리형(boolean type)을 지원하지 않으므로 아래와 같은 방법을 사용할 수 있다.

```
-- 컬럼에 체크 제약조건(check constraint)을 사용한다.
CREATE TABLE ITEMS (
  ITEM_NO NUMBER(3),
  ISACTIVE CHAR(1) CHECK (ISACTIVE IN ('Y','N'))
);

INSERT INTO ITEMS VALUES (101, 'Y');
```

2.4 데이터를 조회하기 위한 SELECT문

- SELECT 문은 데이터를 조회하기 위한 SQL 명령어이다.

```
-- 형식
SELECT [DISTINCT] {*, COLUMN[ALIAS], . . .}
FROM TABLE_NAME;

-- 예
SELECT * FROM DEPT;
SELECT * FROM EMP;
DESC EMP
-- EMP 테이블의 구조 살펴보기
SET LINESIZE 100
-- 라인당 출력될 문자 수 변경
SELECT EMPNO, ENAME FROM EMP;
```

2.5 산술 연산자

- SQL은 다른 프로그래밍 언어와 같이 산술 연산자를 사용할 수 있다.

| 종류 | 예 |
|----|-----------------------------|
| + | SELECT SAL + COMM FROM EMP; |
| - | SELECT SAL - 100 FROM EMP; |
| * | SELECT SAL * 12 FROM EMP; |
| / | SELECT SAL / 2 FROM EMP; |

```
SELECT ENAME, SAL, SAL*12
FROM EMP;
```

2.6 NULL도 데이터!

- 오라클에서의 널은 매우 중요한 데이터이다. 왜냐하면 오라클에서는 컬럼에 널값이 저장되는

것을 허용하는데 널 값을 제대로 이해하지 못한 채 쿼리문을 사용하면 원하지 않는 결과를 얻을 수 있기 때문이다.

■ 널의 정의

- 0(zero)도 아니고 빈 공간도 아니다.
- 미확정(해당 사항 없음), 알 수 없는(unknown) 값을 의미한다.
- 어떤 값인지 알 수 없지만 어떤 값이 존재하고 있다.
- ? 혹은 ∞의 의미이므로 연산, 할당, 비교가 불가능하다.

$$\begin{array}{l} 100 + ? = ? \\ 100 + \infty = \infty \end{array}$$

```
SELECT ENAME, SAL, JOB, COMM, SAL*12, SAL*12+COMM
FROM EMP;
```

-- NVL(Null Value) 함수는 NULL을 0 또는 다른 값으로 변환하기 위해서 사용한다.

```
SELECT ENAME, SAL, JOB, COMM, SAL*12, SAL*12+NVL(COMM, 0)
FROM EMP;
```



2.7 컬럼 이름에 별칭 지정하기

- SQL에서 쿼리문의 결과가 출력될 때, 컬럼 이름이 컬럼에 대한 헤딩(heading)으로 출력된다.

2.7.1. AS로 컬럼에 별칭 부여하기

- 컬럼 이름 대신 별칭을 출력하고자 하면 컬럼을 기술한 바로 뒤에 AS 라는 키워드를 쓴 후 별칭을 기술한다.
- AS 키워드는 생략이 가능하다.

```
SELECT ENAME, SAL*12+NVL(COMM, 0) AS ANNSAL
FROM EMP;
```

-- AS 키워드는 생략이 가능하다.

```
SELECT ENAME, SAL*12+NVL(COMM, 0) ANNSAL
FROM EMP;
```

2.7.2. ""로 별칭 부여하기

- 위 예를 살펴보면 별칭을 부여 할 때에는 대소문자를 섞어서 기술하였는데 출력 결과를 보면 일괄적으로 대문자로 출력된 것을 확인할 수 있다.
- 대소문자를 구별하고 싶으면 " "을 사용한다.
- " "을 사용하여 별칭을 부여할 경우에는 별칭에 공백문자나 \$,_, #등 특수 문자를 포함시킬 수 있다.

```
SELECT ENAME, SAL*12+NVL(COMM, 0) "A N N S A L"
FROM EMP;
```

2.7.3. 별칭으로 한글 사용하기

- 오라클에서 한글을 지원하므로 별칭이 아닌 테이블을 생성할 때 컬럼을 설정하면서 컬럼 이름도 한글로 부여할 수 있다.

```
SELECT ENAME, SAL*12+NVL(COMM, 0) "연봉"
FROM EMP;
```

2.8 Concatenation 연산자의 정의와 사용

- Concatenation 의 사전적인 의미는 연결이다.
- 여러 개의 컬럼을 연결할 때 사용하는데 Concatenation 연산자로 "||" 수직바를 사용한다.

```
SELECT ENAME || ' IS A ' || JOB
FROM EMP;
```

2.9 DISTINCT 키워드

- 동일한 데이터 값들이 중복되지 않도록, 즉 한 번씩만 출력되도록 하기 위해서 사용한다.

```
SELECT DEPTNO
FROM EMP;

SELECT DISTINCT DEPTNO
FROM EMP;
```

[과제] 과제-02-01.TXT

1. emp 테이블을 구성하는 각 컬럼의 데이터 형태에 대해서 설명하세요.

```
SQL> DESC EMP
이름
-----
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              NUMBER(2)

SQL>
```

| 컬럼 이름 | 자료형 설명 |
|----------|------------------|
| EMPNO | 정수로 최대 4자리 |
| ENAME | 가변형 문자열로 최대 10자리 |
| JOB | |
| MGR | |
| HIREDATE | |
| SAL | |
| COMM | |
| DEPTNO | |