

## Chapter 1. 관계형 데이터베이스 개념

Oracle 데이터베이스를 배우기에 앞서 먼저 관계형 데이터베이스에 대한 개념을 간략하게 검토해 볼 필요가 있다. 그러나, 관계형 데이터베이스를 이해하기 위해서는 관계형 데이터 모델을 먼저 살펴봐야 하는데 이 책은 관계형 데이터 모델링의 방법론을 기술한 것이 아니므로 데이터 모델링에 대한 학문적 접근은 다루지 않지만, 오라클 데이터베이스를 사용하기 위해서는 필수적으로 관계형 데이터 모델의 구성요소들을 이해하고 개체관계도(ERD : Entity-Relational Diagram)를 분석 할 줄 알아야 한다. 관계형 데이터 모델링은 데이터베이스 시스템 구축에 있어서 기초가 되므로 잘못 설계된 데이터 모델은 추후 데이터베이스 전체를 재구축해야하는 엄청난 결과를 초래하게 될 수도 있다.

### 관계형 데이터 모델(Relational Data Model)

현재 상용 데이터베이스 제품의 대다수가 채용하고 있는 관계형 데이터 모델의 개념이 처음 언급된 것은 1970년대 E. F. Codd 박사의 "A Relational Model of Data for Large Shared Data Banks"라는 논문이었다. 그 당시, 사용되던 데이터 모델로는 계층형 데이터 모델, 네트워크형 데이터 모델이 있었지만 관계형 데이터 모델의 구조가 기존 데이터 모델에 비해 좀 더 유연하여 실세계를 좀 더 현실감 있게 반영할 수 있었기 때문에 많은 데이터베이스 시스템에 구현되었으며 이로 인하여 관계형 데이터 모델을 지원하는 관계형 데이터베이스 관리 시스템(RDBMS) 제품들이 데이터베이스 시장을 지배하게 되었다.

관계형 데이터 모델은 기본적으로 다음과 같은 핵심적인 3개의 구성요소로 이루어져 있으며, 실세계의 모든 업무체계를 아래의 3가지 구성요소로 모두 표현할 수 있다는 개념이다.

- 개체(Entity) : 시스템화하고자 하는 사건, 사물
- 관계(Relationship) : 개체간, 속성간의 연관성
- 속성(Attribute) : 개체, 관계성의 성질을 나타내는 더 이상 쪼갤 수 없는 정보의 단위

위에서 표현된 각 구성요소들이 결국에는 관계형 데이터베이스에 구현된다. 즉, 개체는 테이블이라는 2차원의 구조로 표현되며, 관계는 외래키(FK : Foreign Key), 속성은 테이블내의 컬럼으로 구축된다. 그러므로, 데이터베이스 시스템을 구축하기에 앞서 모델링하고자 하는 실세계를 개체, 관계, 속성으로 명확히 정의할 필요가 있다.

### 관계형 데이터베이스의 정의

관계형 데이터 모델을 전산화하여 논리적으로 구축해 놓은 것이 관계형 데이터베이스이다. 관계형 데이터베이스는 정보를 저장하기 위하여 아래와 같은 2차원의 테이블 구조를 사용한다. 예를 들어, 회사의 모든 사원 정보를 데이터베이스에 저장하고자 한다면 다음과 같은 사원 테이블이 필요하게 되는데, 이는 관계형 모델로 정의된 사원 개체가 관계형 데이터베이스 내에서 사원 테이블로 구현된 것이다.

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80/12/17	800		20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7566	JONES	MANAGER	7839	81/04/02	2975		20
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7698	BLAKE	MANAGER	7839	81/05/01	2850		30
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7788	SCOTT	ANALYST	7566	87/04/19	3000		20
7839	KING	PRESIDENT		81/11/17	5000		10
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
7876	ADAMS	CLERK	7788	87/05/23	1100		20
7900	JAMES	CLERK	7698	81/12/03	950		30
7902	FORD	ANALYST	7566	81/12/03	3000		20
7934	MILLER	CLERK	7782	82/01/23	1300		10

그림 1-1. EMP 테이블

## 관계형 데이터베이스의 용어

관계형 데이터베이스에서 모든 데이터는 2차원 테이블에 저장된다. 다음과 같이 사원 정보가 입력되어 있는 사원 테이블에서 각 번호가 의미하는 것은 다음과 같다.

②	EMPNO	ENAME	JOB	MGR	HIREDATE	③	COMM	④	DEPTNO
	7369	SMITH	CLERK	7902	80/12/17	800			20
	7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	⑤	30
	7521	WARD	SALESMAN	7698	81/02/22	1250	500		30
	7566	JONES	MANAGER	7839	81/04/02	2975			20
	7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400		30
	7698	BLAKE	MANAGER	7839	81/05/01	2850			30
	7782	CLARK	MANAGER	7839	81/06/09	2450	⑥		10
	7788	SCOTT	ANALYST	7566	87/04/19	3000			20
	7839	KING	PRESIDENT		81/11/17	5000			10
①	7844	TURNER	SALESMAN	7698	81/09/08	1500	0		30
	7876	ADAMS	CLERK	7788	87/05/23	1100			20
	7900	JAMES	CLERK	7698	81/12/03	950			30
	7902	FORD	ANALYST	7566	81/12/03	3000			20
	7934	MILLER	CLERK	7782	82/01/23	1300			10

그림 1-2. 사원 테이블의 구조

- ① 행(Row) : 특정한 한 명의 사원을 표현하기 위해 필요한 모든 데이터이다. 테이블 내의 각 행은 반드시 기본키에 의하여 식별 가능해야 하며, 기본키의 값은 절대 중복되면 안 된다. 테이블내의 행들은 임의 순서로 저장되어 있으나, 검색시 정렬이 가능하다.
- ② 기본키(PK : Primay Key) : 사번(EMPNO) 컬럼은 절대 중복 될 수 없으며, 값이 반드시 지정되어야 한다. 그러므로, 테이블내의 모든 행들은 기본키에 의해 식별될 수 있다.
- ③ 컬럼(Column) : 급여(SAL) 컬럼은 기본키가 아니므로 특별한 제약조건이 지정되지 않는다

한, 값을 지정하지 않아도 되며 값이 중복되어도 상관 없다.

- ④ 외래키(FK : Foreign Key) : 외래키는 테이블간의 관계를 정의하는 컬럼이다. 부서코드(DEPTNO) 컬럼은 외래키로서 자기 자신 또는 다른 테이블의 기본키 또는 고유(Unique) 제약조건이 지정되어 있는 고유키 컬럼들을 참조한다. 여기서는 부서(DEPT) 테이블의 부서코드(DEPTNO) 컬럼을 참조하고 있다.
- ⑤ 필드(Field) : 행과 컬럼이 교차하는 지점을 필드(Field)라고 정의한다.
- ⑥ 널(Null) : 필드에 값이 지정되어 있지 않는 경우를 일반적으로 “NULL 값을 갖는다”라고 표현한다. NULL 값 자체도 다음의 두 가지 의미를 갖는데, “값이 아직 지정되지 않았다(Undefine)”는 의미와 “값을 아직 모른다(Unknown)”는 의미이다. 절대, 0(Zero)이나 공백(Space)과는 의미가 다르므로 혼동해서는 안 된다.

## 기본키(PK)와 외래키(FK)

ERD에서 표현된 개체는 추후 테이블로, 속성은 컬럼으로, 관계는 외래키로 구현된다고 기술한 바 있다. 여기서는 관계형 데이터베이스에서 아주 중요하게 다루어지는 기본키(PK : Primary Key)와 외래키(FK : Foreign Key)에 대하여 살펴본다. 기본키는 다음 그림과 같이 테이블내의 각 행을 유일하게 식별 할 수 있는 컬럼이며 외래키는 자기 자신 또는 다른 테이블의 기본키 또는 고유키들을 참조하는 컬럼이다.

PK      부서 테이블							
DEPTNO	DNAME	LOC					
10	ACCOUNTING	NEW YORK					
20	RESEARCH	DALLAS					
30	SALES	CHICAGO					
40	OPERATIONS	BOSTON					

PK      사원 테이블							FK
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80/12/17	800		20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7566	JONES	MANAGER	7839	81/04/02	2975		20
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7698	BLAKE	MANAGER	7839	81/05/01	2850		30
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7788	SCOTT	ANALYST	7566	87/04/19	3000		20
7839	KING	PRESIDENT		81/11/17	5000		10
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
7876	ADAMS	CLERK	7788	87/05/23	1100		20
7900	JAMES	CLERK	7698	81/12/03	950		30
7902	FORD	ANALYST	7566	81/12/03	3000		20
7934	MILLER	CLERK	7782	82/01/23	1300		10

그림 1-3. 부서 테이블과 사원 테이블

예를 들어, 위 그림과 같이 관계형 데이터베이스에 부서 테이블과 사원 테이블이 구축되어

있다고 가정하자. 부서 테이블에는 부서 정보가 입력되어 있으며, 사원 테이블에는 사원들의 정보가 각각 입력되어 있다. 부서 테이블내에서 각각의 부서 정보를 유일하게 식별할 수 있는 컬럼을 기본키로 설정하게 되는데, 이 경우는 부서코드(DEPTNO)를 기본키로 설정했다. 그러나, 부서 테이블내에 입력된 실제 데이터를 보면 부서코드(DEPTNO) 컬럼 외에 부서명(DNAME), 위치(LOC) 컬럼 모두 기본키가 될 자격(모든 값이 고유하고 NULL 값이 존재하지 않음)은 충분히 있지만 부서명(DNAME)이나 위치(LOC) 컬럼은 추후 NULL 값이 입력되거나 데이터 값이 중복 입력될 소지가 있으므로 기본키의 후보로서 적절치 않다. 또한, 기본키는 해당 테이블의 대표 속성으로서의 의미를 가지기 때문에 부서코드(DEPTNO)가 기본키로서 적절함을 알 수 있다. 부서 테이블 또한 동일한 맥락에서 사원(EMPNO)이 기본키로 지정되었음을 확인할 수 있다.

외래키는 테이블간의 관계를 정의하는 컬럼으로 위 그림에서는 사원들의 소속부서 정보를 관리하고자 설정된 것이다. 만약, 사원의 소속부서 정보를 굳이 데이터베이스에 구축하여 관리하고자 할 필요가 없다면 이러한 관계 설정은 불필요한 것이 된다. 여기서는 부서 테이블의 기본키 컬럼을 사원 테이블의 외래키 컬럼으로 추가시켜 부서와 사원간의 관계를 상호 참조 할 수 있도록 지정하였다. 그러므로, 각각의 사원들이 어느 부서에 소속되어 있는지, 각각의 부서에 소속된 직원들은 누구인지 등의 정보를 검색할 수 있게 된다.

저자가 데이터 모델링을 강의하고 수강생들이 작성한 데이터 모델을 검토해보면 테이블간의 관계설정에 너무 부담을 갖고 불필요한 관계설정을 하게 되는 경우가 많은데, 반드시 기억해두어야 할 것은 이러한 관계 설정이 정말 필요한 것인지부터 먼저 규명을 해야 된다는 사실이다.

## 데이터 모델링의 4단계

이 책에서는 데이터 모델링에 대하여 자세히 다루지는 않지만, 개략적인 내용을 소개함으로써 ERD의 이해를 돕고자 한다. 데이터 모델링은 다음 그림과 같이 4단계에 걸쳐서 진행되며, 각 단계별 수행 작업과 산출물은 다음과 같다.



그림 1-4. 데이터 모델링의 4단계

## 1) 요구형성 및 분석

데이터베이스 시스템 구축을 위한 요구사항을 취합하여 분석한 후, 업무처리 규정을 도출하는 단계이다. 이 단계에서 필요한 자료는 업무분장표, 현행업무흐름도, 입출력 활용장표, 인터뷰 내용, 현 시스템분석도 등이 요구되며 이를 기반으로 최종 업무처리 규정이 산출된다.

예) 업무처리규정

- 각 회사원은 한 부서에 소속된다.
- 각 부서에는 여러 명의 사원이 배치된다.
- 회사원에 관해서는 사번, 이름, 주소 등의 정보가 유지된다.
- 부서에 관해서는 부서코드, 부서명에 대한 정보가 유지된다.

## 2) 개념적 설계

앞 단계에서 산출된 업무처리규정을 이용하여 중심 데이터인 개체를 도출하고 개체 관계도(ERD)를 작성하는 과정이다. 데이터베이스를 처음 다루는 입문자에게 개체라는 개념이 상당히 어렵게 느껴질 수도 있으나 관계형 데이터 모델이 최종적으로 관계형 데이터베이스에 구축될 때, 개체가 테이블로 변환된다는 생각을 하면 쉽게 개념을 잡을 수 있을 것이다. 저자가 현업 관리자들을 대상으로 강의를 하다보면 간혹 ERD에 대해 전혀 모르거나, ERD를 제대로 관리하지 않아서 데이터베이스에 어떤 데이터가 있는지 또는 어떤 데이터가 이제 더 이상 필요치 않은지를 파악하지 못하는 경우도 있었다. 그로 인하여, 더 이상 필요하지 않은 수백 메가 분량의 데이터를 많은 시간을 소비해 가면서 여전히 백업을 수행하고 있는 경우도 있었다. 데이터베이스 관리자라면 반드시 ERD를 정기적으로 관리해야 한다.

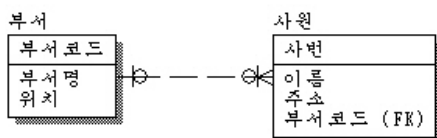


그림 1-5. 개체 관계도 예

## 3) 논리적 설계

전 단계에서 산출된 ERD로부터 개체를 테이블, 속성을 컬럼, 관계를 외래키로 적절히 변환하는 단계이다. 이 과정에서는 정규화라는 과정이 필수적으로 선행된다. 정규화란 자료의 손실이나 불필요한 정보의 도입 없이 데이터의 일관성, 최소한의 데이터 중복, 최대의 데이터 안정성 확보를 위한 안정적 자료구조로 변환하는 기법을 의미한다. 쉽게 설명하자면 동일한 테이블내에 같은 데이터들이 중복해서 입력이 되면 데이터가 입력, 수정, 삭제되는 조작 과정에서 여러 가지 이상현상이 발생하기 때문에 정규화는 이러한 일들이 발생되지 않도록 데이터 중복을 최소화하여 테이블을 여러 개로 분리시키는 일련의 과정을 말한다. 정규화에 관한 내용은 이 책의 범위를 벗어남으로 별도의 데이터 모델링 관련 서적을 참고하도록 한다.

예) 테이블로의 변환 예

부서 테이블(부서코드(PK), 부서명, 위치)

사원 테이블(사번(PK), 이름, 주소, 부서코드(FK))

#### 4) 물리적 설계

앞 단계에서 변환된 관계형 테이블을 실제 데이터베이스 서버에 구현할 수 있도록 테이블 차트를 작성한다. 또한, 이 단계에서는 서버의 디스크 구조에 따라 데이터 파일의 배치, 인덱스 최적 설계, 트랜잭션 처리 알고리즘 등을 결정하여, 구현된 데이터베이스가 최대한의 성능을 발휘할 수 있도록 구현하는 단계이다.

예) 테이블 차트의 예

테이블명 : DEPT

컬럼명	키	Null/ Unique	체크제약	FK 참조 테이블	FK 참조 컬럼	데이터 타입	길이
DEPTNO	PK	NN, U				VARCHAR2	3
DNAME		NN				VARCHAR2	10
LOC						VARCHAR2	10

테이블명 : EMP

컬럼명	키	Null/ Unique	체크제약	FK 참조 테이블	FK 참조 컬럼	데이터 타입	길이
EMPNO	PK	NN, U				VARCHAR2	5
ENAME		NN				VARCHAR2	10
ADDRESS						VARCHAR2	10
DEPTNO	FK			DEPT	DEPTNO	VARCHAR2	3

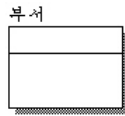
## ERD(Entity-Relationship Diagram)

ERD는 데이터베이스 모델링의 4단계 중에서 개념적 설계의 산출물로서 도출되어야 하는 결과이다. ERD는 수작업으로 작성할 수도 있으나, 최근의 데이터베이스들은 엄청난 수의 테이블을 포함하고 있기 때문에 ERD를 직접 손으로 작성한다는 것은 거의 불가능하다. 이를 위해서 상용 ERD 작성 툴을 많이 사용하는데, 그 대표적인 소프트웨어가 CA(Computer Associates)의 Erwin이라는 프로그램이다. 이 책에서는 Erwin을 사용하는 방법은 설명하지 않고, 기존의 Erwin으로 작성된 ERD를 분석하는 방법에 대하여 알아보도록 한다. Erwin의 사용방법을 학습하고자 하는 경우에는 Erwin 프로그램의 도움말 파일이나 CA의 홈페이지를 참고하도록 한다.

Erwin에서 관계형 데이터 모델의 기본이 되는 3개의 구성요소는 다음과 같이 표현된다.

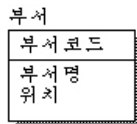
### ■ 개체(Entity)

Erwin에서 개체는 사각형으로 표현되며, 사각형의 상단에 개체의 이름이 기술된다.



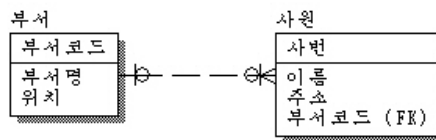
### ■ 속성(Attribute)

Erwin에서 개체의 속성은 사각형 내부에 기술하는데, 사각형의 위 칸에는 기본키 속성을, 아래 칸에는 일반 속성을 기술한다.



### ■ 관계(Relationship)

Erwin에서 관계는 개체 간에 실선 또는 점선으로 표시된다.



ERD에서 관계를 표시하는 방법은 관계의 카디널리티(Cardinality)에 따라 다르게 표현된다. 여기서, 카디널리티란 각 관계자에서 참여자의 수를 표현한 것으로, 쉽게 설명하면 한개 부서에 한명의 사원만이 배정되는 관계는 1:1이 되며, 한개 부서에 한명 이상의 사원이 배정되면 1:M(Many)이 되는 것이다. M:M의 경우는 한개의 부서에 한명 이상의 사원이 배정되며, 한명의 사원이 여러 부서에 동시에 소속될 수 있다는 의미가 된다. 예를 들면, 인사과에 홍길동, 경리과에 콩쥐 등 한개 부서에 한 사람만 배정되는 관계라면 1:1, 인사과에 홍길동, 콩쥐, 팔쥐 등 여러 명의 사원이 배정되는 관계라면 1:M, 인사과에 홍길동, 콩쥐, 팔쥐 등 여러 명의 사원이 배정되는 동시에 홍길동이 인사과, 총무과, 홍보과 등의 여러 부서에 소속되면 M:M 관계가 되는 것이다. 그러나, M:M 관계의 경우는 논리적인 테이블로 구현할 수 없기 때문에 M:M 관계를 테이블로 변환하여 두개의 1:M 관계로 반드시 변환하여야 한다는 점을 잊으면 안 된다.

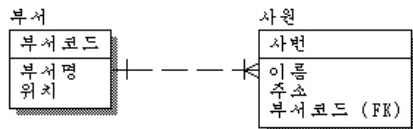
그러면 Erwin에서 이러한 카디널리티가 어떻게 표현되는지 살펴보자. Erwin에서 카디널리티가 1로 표현되는 개체 쪽은 실선으로 그리고, M으로 표현되는 개체 쪽은 까마귀 발(Crow's foot)과 같이 그린다. 위와 같은 경우는 한 개의 부서에 여러 명의 사원이 소속되는 관계를 표현한 것임을 알 수 있다.

또한, 관계로 표현된 점선의 양단에 ○ 기호를 확인 할 수 있는데, 이 기호는 선택도를 표시한 것이다. 즉, ○이 붙어 있는 경우를 선택(Optional), 그렇지 않은 경우를 필수(Mandatory)라고 표현하는데, 사원 개체 쪽에 ○이 붙어 있는 경우는 각 부서에 사원이 한명도 배정되지 않을 수도 있다는 의미이고 반대로, 부서 개체 쪽에 ○이 붙어 있는 경

우는 사원 중에서 어떠한 부서에도 소속되지 않은 사원이 있을 수 있음을 의미한다.

카디널리티와 선택도가 이해되었다면 다음과 같은 ERD를 각각 분석해보도록 하자. ERD를 읽을 때는 왼쪽 개체에서 오른쪽 개체로 한번 읽고, 그 반대로 한번 읽는다.

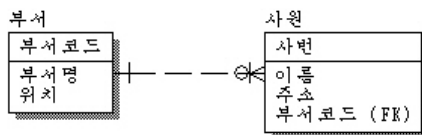
첫 번째 ERD는 부서와 사원 개체 간의 관계가 1:M이며 선택도를 표현하는 ○이 양쪽 모두 누락된 경우이다.



좌 → 우 : 각 부서는 한명 이상의 사원이 반드시 소속된다.(Mandatory)

우 → 좌 : 각 사원은 한개 부서에 반드시 배정된다.(Mandatory)

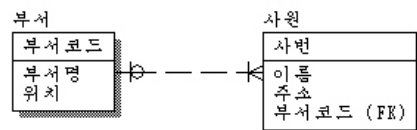
두 번째 ERD는 첫 번째 ERD에서 사원 개체에 ○이 추가된 경우이다.



좌 → 우 : 각 부서는 한명 이상의 사원이 소속될 수도 있다.(Optional)

우 → 좌 : 각 사원은 한개 부서에 반드시 배정된다.(Mandatory)

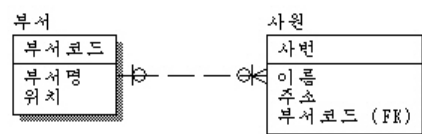
세 번째 ERD는 첫 번째 ERD에서 부서 개체에 ○이 추가된 경우이다.



좌 → 우 : 각 부서는 한명 이상의 사원이 반드시 소속된다.(Mandatory)

우 → 좌 : 각 사원은 한개 부서에 배정될 수도 있다.(Optional)

네 번째 ERD는 부서 개체와 사원 개체에 ○이 추가된 경우이다.



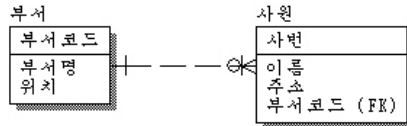
좌 → 우 : 각 부서는 한명 이상의 사원이 소속될 수도 있다.(Optional)

우 → 좌 : 각 사원은 한개 부서에 배정될 수도 있다.(Optional)

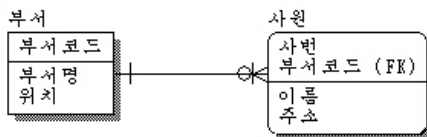
마지막으로 카디널리티와 선택도 이외에 한가지 더 알아두어야 할 것이 있는데 관계의



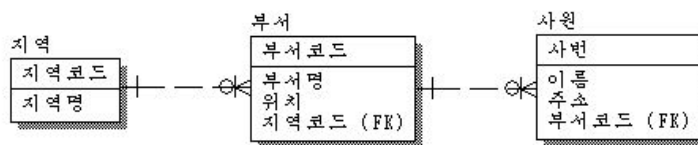
식별성이다. 지금까지 Erwin에서 표현되는 관계는 점선이었는데, 이 점선을 비식별(Non-Identifying) 관계라고 부른다. 비식별 관계로 표시되는 경우는 카디널리티가 1쪽인 개체의 기본키 속성이 M쪽인 사원 개체의 외래키로 구현되는 과정에서 일반 속성으로 추가된다는 의미이다. 즉, 아래 그림과 같이 부서 개체의 기본키인 부서코드 속성이 사원 개체의 외래키로 생성되며 일반 속성으로 추가된다는 의미이다.



반면에 다음과 같이 실선으로 표시되는 관계를 식별(Identifying) 관계라고 부르는데, 이와 같은 경우, 카디널리티가 1쪽인 개체의 기본키 속성이 M쪽인 사원 개체의 외래키로 구현되는 과정에서 사원의 기본키 속성으로 추가된다는 의미이다. 즉, 아래 그림과 같이 부서 개체의 기본키인 부서코드 속성이 사원 개체의 외래키로 생성되며 기본키 속성으로 추가된다는 의미이다.



식별/비식별 관계는 두개의 개체 간에서는 큰 의미가 없기 때문에 다음과 같은 세 개의 개체로 구성된 식별/비식별 관계를 살펴보도록 하자. 먼저, 지역 개체, 부서 개체, 사원 개체로 구성되어 있으며 각각 1:M 관계로 설정되었다. 즉, 지역 개체의 기본키인 지역코드가 부서 개체의 외래키(일반 속성)로 추가되었고, 부서 개체의 기본키인 부서코드가 사원 개체의 외래키(일반 속성)로 추가되었음을 확인할 수 있다. 이러한 ERD에서 사원의 이름을 검색조건으로 지역명을 검색하고자 한다면 3개의 개체를 모두 조인(Join) 하여야 한다.



반면, 아래와 같이 지역 개체와 부서 개체간의 식별관계로 구성된 경우는 사원 개체에 지역 개체와 부서 개체의 기본키가 각각 포함되므로 지역 개체와 사원 개체가 직접 조인(Join)이 가능해지기 때문에 위의 경우보다 수행성능이 향상될 것임을 예상할 수 있다.



식별/비식별 관계는 구현상 장단점이 있기 때문에 상황에 맞추어 선택해야 한다. 조인에

대한 설명은 뒷 부분에서 다룰 것이므로 당장 이 부분을 이해하지 못하더라도 상관은 없다. 나중에 조인에 대해 학습 한 후, 다시 한번 가볍게 읽어 볼 것을 권장한다.

## SQL 문장의 종류

SQL은 Structured Query Language의 줄임말로써 사용자는 SQL 문장을 이용하여 데이터베이스에 접근하고 데이터를 손쉽게 검색 및 조작할 수 있다. 또한, SQL 문장은 영어와 유사하기 때문에 쉽게 배울 수 있는 장점도 있다. Oracle 데이터베이스의 SQL은 산업 표준인 ANSI와 ISO를 준수하고 있기 때문에 다른 상용 데이터베이스 사용자라도 해당 데이터베이스가 관련 표준을 따르고 있다면 Oracle의 SQL 문장을 특별하게 수정할 필요 없이 원문 그대로 사용할 수 있다. 아래 표는 앞으로 설명하게 될 SQL 문장을 용도별로 분류한 것이다.

표 1-1. SQL 문장의 종류

문장	설명
SELECT	데이터베이스로부터 데이터를 검색
INSERT UPDATE DELETE MERGE	데이터베이스 내의 테이블에 새로운 행을 입력하거나, 기존의 행을 수정 또는 삭제하는 명령어로 데이터 조작어(DML : Data Manipulation Language)라고 함
CREATE ALTER DROP RENAME TRUNCATE	테이블을 생성, 변경, 삭제하는 명령어로 데이터 정의어(DDL : Data Definition Language)라고 함
COMMIT ROLLBACK SAVEPOINT	DML 문장에 의한 변경 사항을 관리하거나, 변경사항을 하나의 논리적 트랜잭션으로 포함시키는 명령어
GRANT REVOKE	데이터베이스와 데이터베이스를 구성하는 구조(테이블, 뷰 등)에 접근 권한을 부여하거나 회수하는 명령어로 데이터 제어어(DCL : Data Control Language)라고 함

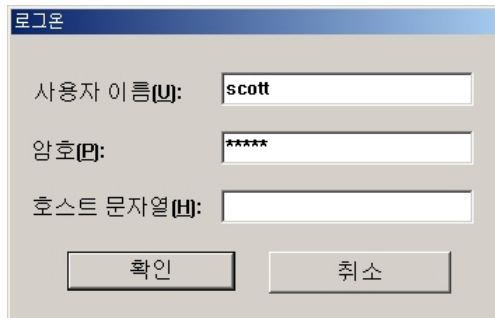
## 실습에 사용할 계정 및 테이블 목록

이 책 전체에서 다루어지는 모든 예제를 실습하려면 Oracle 데이터베이스에서 샘플로 제공되는 SCOTT 계정을 사용해야한다. SCOTT 계정의 암호는 TIGER이며, 다음의 두가지 방법으로 Oracle 데이터베이스에 접속할 수 있다. 첫 번째 방법은 기존 SQL\*Plus를 사용하는 방법이고, 두 번째 방법은 Oracle 9i 버전부터 제공되는 Apache 웹서버를 통하여 브라우저로 접속하는 방법이다.

### ■ SQL\*Plus에 의한 접속

1. [시작]-[프로그램]-[Oracle-OraHome92]-[Application Development]-[SQL Plus]를 차례로 클릭한다.
2. 로그인 대화상자에서 사용자 이름에는 SCOTT, 암호에는 TIGER를 입력하고, **확인**을

클릭한다.



A login dialog box titled '로그온' (Login). It contains three input fields: '사용자 이름(U):' (Username) with 'scott' entered, '암호(P):' (Password) with '\*\*\*\*\*' entered, and '호스트 문자열(H):' (Host string) which is empty. At the bottom are two buttons: '확인' (OK) and '취소' (Cancel).

그림 1-6. SQL\*Plus 로그인 대화상자

3. 사용자 이름과 암호가 올바르게 입력되었다면 오라클 데이터베이스에 정상적으로 로그인 된다.

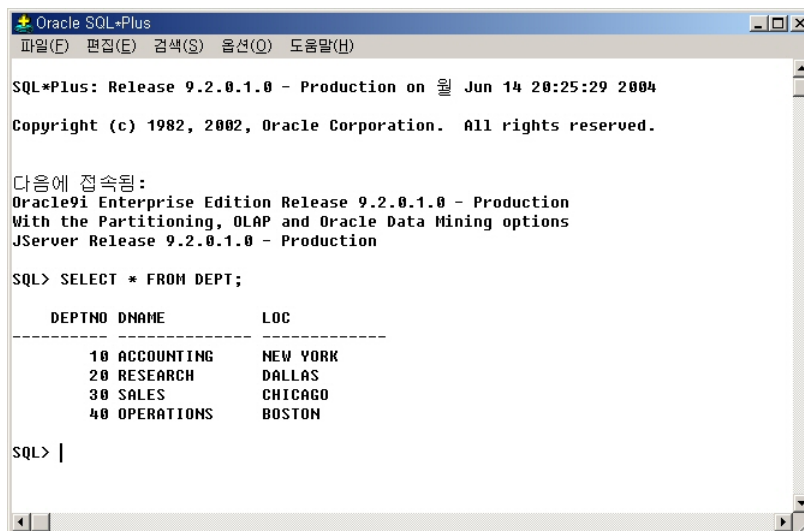


그림 1-7. SQL\*Plus 작업환경

#### ■ 브라우저에 의한 접속

1. 브라우저를 실행하고 주소에 **http://서버이름:포트번호/isqlplus**를 입력한다. 여기서, 서버이름은 Oracle 데이터베이스 서버가 설치된 컴퓨터의 호스트명 또는 IP 어드레스를 입력하며, 포트번호는 Oracle 데이터베이스와 함께 설치되는 아파치 서버의 리스닝 포트를 기술한다. 아파치 서버의 포트번호가 80번으로 구성되면, 포트번호를 생략해도 되지만, 80번 포트 이외의 번호로 구성되면 포트번호를 반드시 기술해야 한다. 아파치 서버의 포트를 확인하려면 C:\Woracle\Wora92\WApache\WApache\ports.ini 파일에서 확인 할 수 있다.
2. iSQL\*Plus 로그인 페이지에서 사용자 이름에 **SCOTT**, 암호에 **TIGER**를 입력하고 로그인 버튼을 클릭한다.

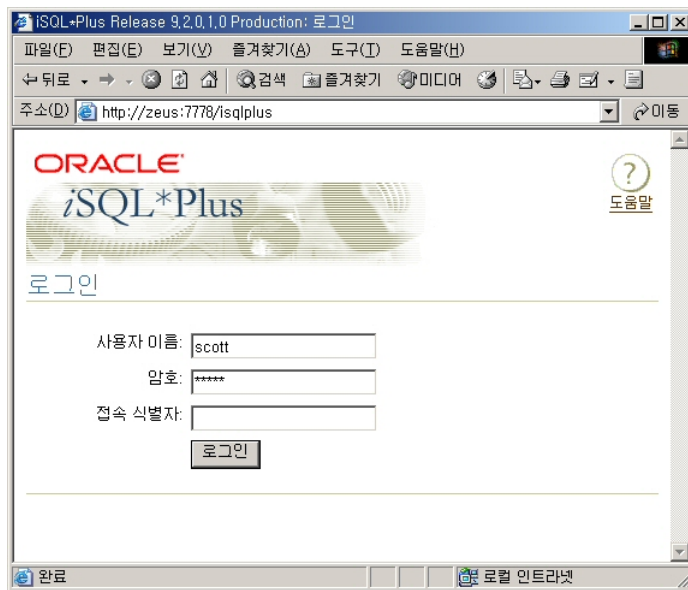


그림 1-8. iSQL\*Plus 로그인 대화상자

3. 사용자 이름과 암호가 올바르게 입력되었다면 오라클 데이터베이스에 정상적으로 로그인 된다.

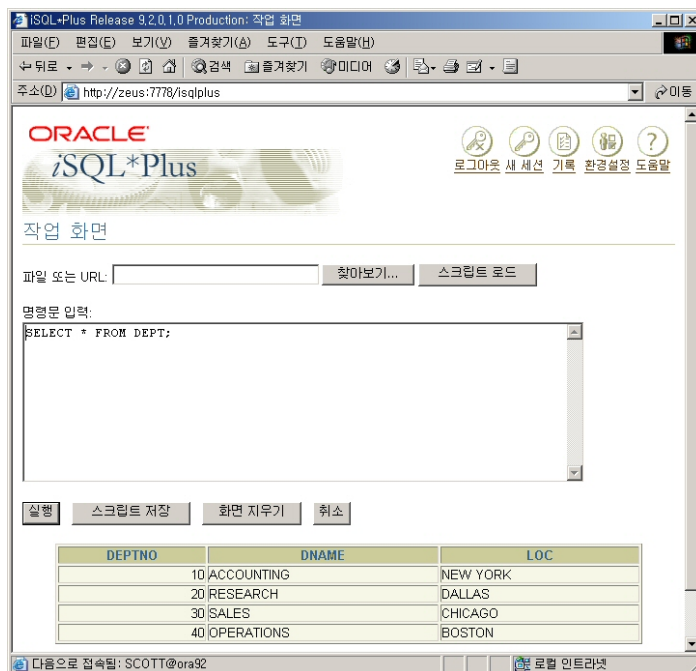


그림 1-9. iSQL\*Plus 작업환경

이 책에서 실습을 위해 다루고 있는 대표적인 테이블은 다음과 같다.

#### ■ 부서(DEPT) 테이블

부서 테이블은 부서 정보가 입력되어 있는 테이블로서 부서코드(DEPTNO), 부서명(DNAME), 부서위치(LOC)로 구성되어 있다.

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

그림 1-10. DEPT 테이블

#### ■ 사원(EMP) 테이블

사원 테이블은 사원 정보가 입력되어 있는 테이블로서 사번(EMPNO), 사원명(ENAME), 업무(JOB), 관리자사번(MGR), 입사일(HIREDATE), 급여(SAL), 커미션(COMM), 부서코드(DEPTNO)로 구성되어 있다.

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80/12/17	800		20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7566	JONES	MANAGER	7839	81/04/02	2975		20
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7698	BLAKE	MANAGER	7839	81/05/01	2850		30
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7788	SCOTT	ANALYST	7566	87/04/19	3000		20
7839	KING	PRESIDENT		81/11/17	5000		10
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
7876	ADAMS	CLERK	7788	87/05/23	1100		20
7900	JAMES	CLERK	7698	81/12/03	950		30
7902	FORD	ANALYST	7566	81/12/03	3000		20
7934	MILLER	CLERK	7782	82/01/23	1300		10

그림 1-11. EMP 테이블

#### ■ 급여등급(SALGRADE) 테이블

급여등급 테이블은 급여등급이 입력되어 있는 테이블로서 등급(GRADE), 하한값(LOSAL), 상한값(HISAL)으로 구성되어 있다.

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

그림 1-12. SALGRADE 테이블

## 복습

1. 관계형 데이터 모델의 기본적인 3가지 구성요소는 무엇인가?
2. 데이터 모델링의 4단계는 무엇인가?
3. 개체관계도(ERD)란 무엇인가?
4. 기본키(Primary Key)와 외래키(Foreign Key)를 각각 설명하시오.