



CHAPTER 03 Oracle 전체 운영 실습

Contents

◦ CHAPTER 03 Oracle 전체 운영 실습

학습 목표

SECTION 01 요구사항 분석과 시스템 설계 그리고 모델링

- 1.1 정보시스템 구축 절차 요약
- 1.2 데이터베이스 모델링과 필수 용어

SECTION 02 Oracle을 이용한 스키마 구축 절차

- 2.1 스키마 생성
- 2.2 테이블 생성
- 2.3 데이터 입력
- 2.4 데이터 활용

Contents

SECTION 03 테이블 외의 데이터베이스 개체의 활용

- 3.1 인덱스
- 3.2 뷰
- 3.3 스토어드 프로시저
- 3.4 트리거

SECTION 04 데이터의 백업 및 복원

- 4.1 백업과 복원

SECTION 05 Oracle과 응용프로그램의 연결

이 장의 핵심 개념

- 3장에서는 이 책 전체에서 배울 내용을 미리 전체적으로 학습하는 것을 목표로 함.
 - 1. 데이터베이스 관련 용어는 데이터, 테이블, DB, DBMS, 열 등이 있음
 - 2. 데이터베이스 구축 절차는 데이터베이스 생성, 테이블 생성, 데이터 입력, 데이터 조회/활용의 순서로 진행됨.
 - 3. 테이블 외에 데이터베이스 개체로는 인덱스, 뷰, 스토어드 프로시저, 함수, 트리거, 커서 등이 있음.
 - 4. 백업은 현재의 데이터베이스를 다른 매체에 보관하는 작업을 말하며, 복원은 다른 매체에 백업된 데이터를 이용해서 원상태로 돌려놓는 작업을 말함.
 - 5. Oracle을 응용프로그램과 연동하는 것은 실무에서 많이 사용되는 방식임.

SECTION 01 요구사항 분석과 시스템 설계 그리고 모델링

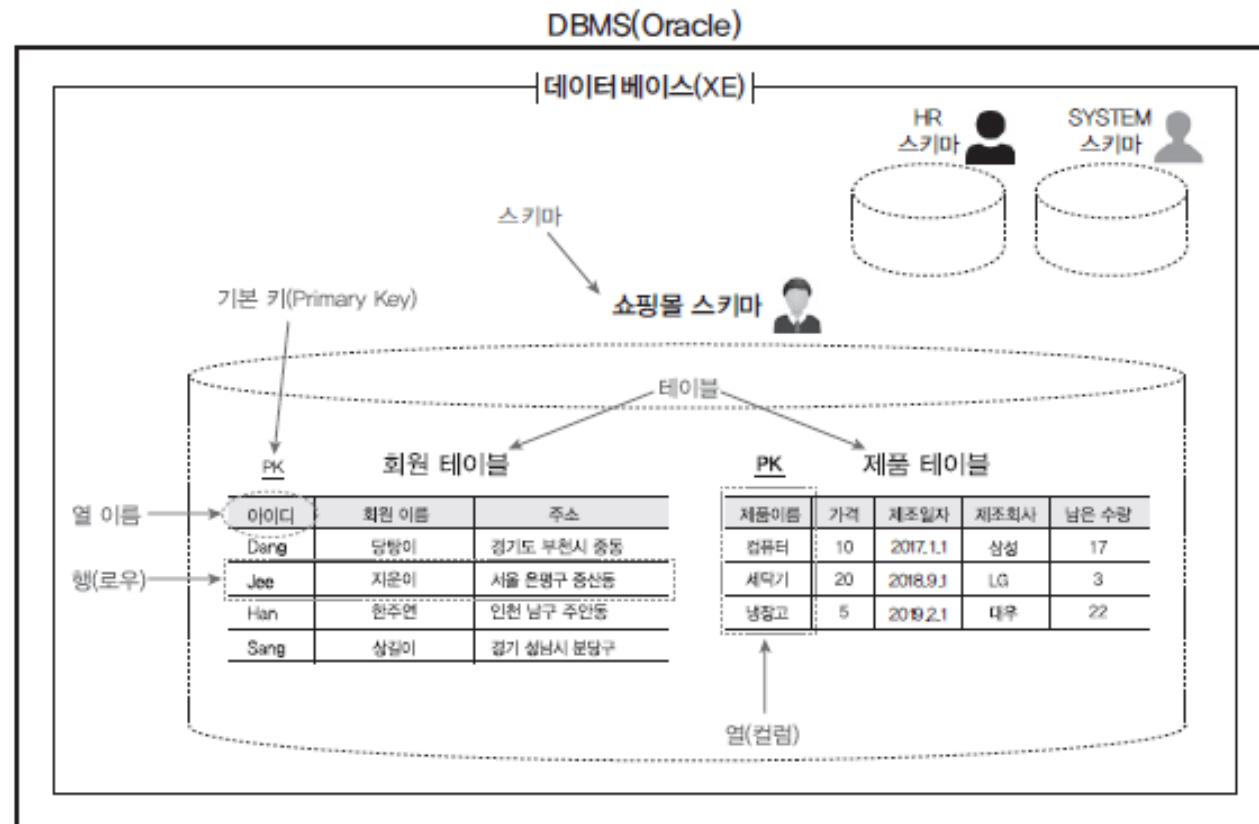
1.1 정보시스템 구축 절차 요약

- 정보시스템을 구축하기 위해서는 일반적으로 분석, 설계, 구현, 시험, 유지보수의 5가지 단계.
 - 분석 단계 : 시스템 분석 또는 요구사항 분석이라고 부름. 요구사항 분석은 현재 우리가 '무엇을' 할 것인지를 결정함.
 - 설계 단계 : 설계는 주로 시스템 설계 또는 프로그램 설계라는 용어로 부름. 구축하고자 하는 시스템을 '어떻게' 할 것인지를 결정함.
 - 시스템 설계가 끝나고 나면 그 결과 문서들을 프로그래머(또는 코더)에게 넘겨 주기만 하면 프로그래머는 설계서에 나온 그대로 프로그램을 작성함.
 - 시스템 설계가 끝나면 가장 큰 작업이 끝난 것으로 간주됨.

SECTION 01 요구사항 분석과 시스템 설계 그리고 모델링

1.2 데이터베이스 모델링과 필수 용어

- 분석과 설계 과정에서 가장 중요한 과정 중의 하나가 '데이터베이스 모델링' 임.
- 저장할 정보는 그냥 단편적으로 저장하는 것이 아니라 테이블이라는 형식에 맞춰서 넣어야 함.



[그림 3-1] 테이블의 구조와 관련 용어

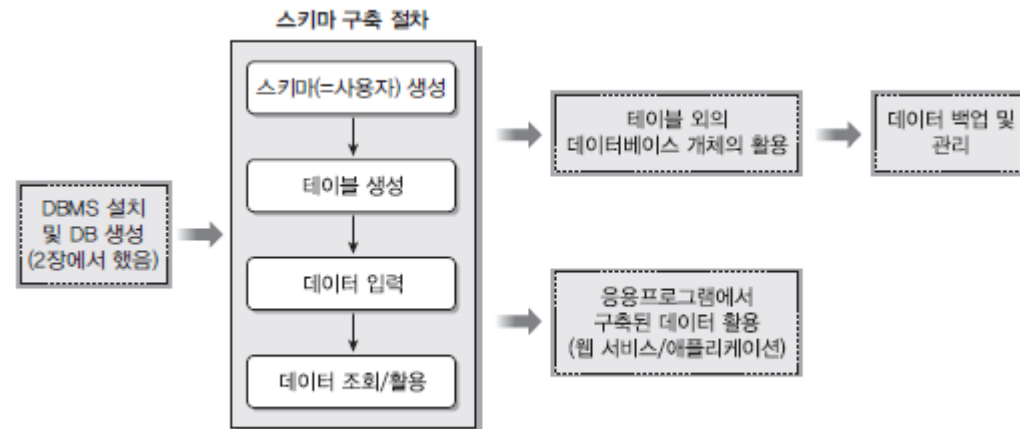
SECTION 01 요구사항 분석과 시스템 설계 그리고 모델링

- 데이터 : 정보는 있으나 아직 체계화 되지 못한 상태를 말함.
- 테이블 : 회원이나 제품의 데이터를 입력하기 위해 표 형태로 표현한 것을 말함.
- 스키마(Schema) : 테이블, 뷰 등이 저장되는 저장소를 말하거나 여러 테이블, 뷰 등의 묶음으로도 말함.
- 데이터베이스(Database, 줄여서 DB) : 여러 개의 스키마가 저장되는 저장 공간임.
- DBMS: DataBase Management System의 약자로 스키마를 관리하는 시스템 또는 소프트웨어를 말함.
- 열(=컬럼=필드) : 각 테이블은 열로 구성됨.
- 열 이름 : 각 열을 구분하기 위한 이름임.
- 데이터 유형 : 열의 데이터 유형을 말함.
- 행(=로우=레코드) : 실질적인 데이터를 말함.
- 기본 키Primary Key 열 : 기본 키(또는 주 키) 열은 각 행을 구분하는 유일한 열을 말함.
- 외래 키Foreign Key 필드 : 두 테이블의 관계를 맺어주는 키를 말함.
- SQLStructured Query Language : 구조화된 질의 언어. SQL은 사람과 DBMS가 소통하기 위한 말(언어)임.

SECTION 02 Oracle을 이용한 스키마 구축 절차

2.1 스키마 생성

- Oracle에서는 '사용자 이름 = 스키마 이름'으로 취급함.
- 스키마 구축/관리 및 활용의 전반적인 절차



[그림 3-3] 스키마 구축/관리 및 활용의 전반적인 절차

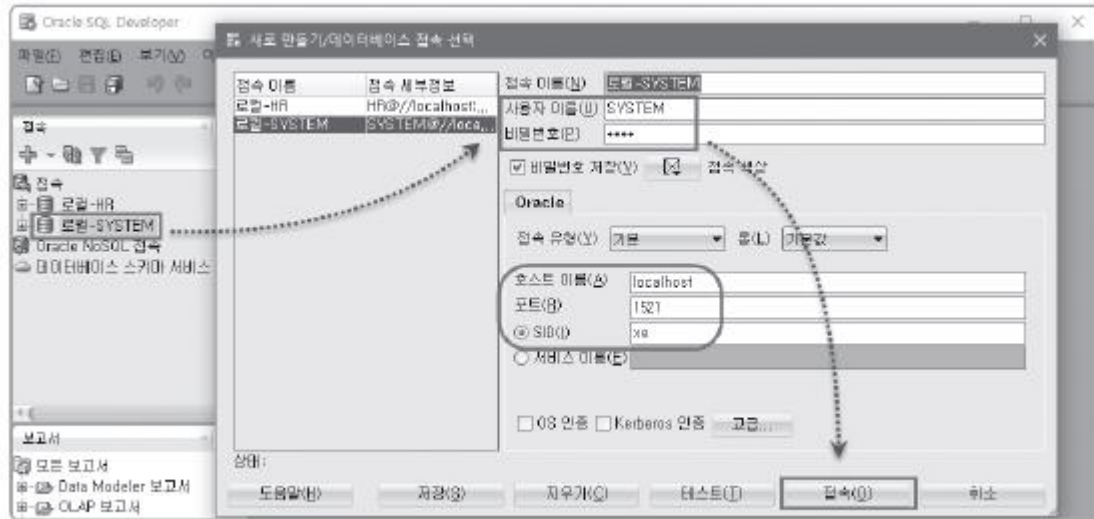
SECTION 02 Oracle을 이용한 스키마 구축 절차

실습1

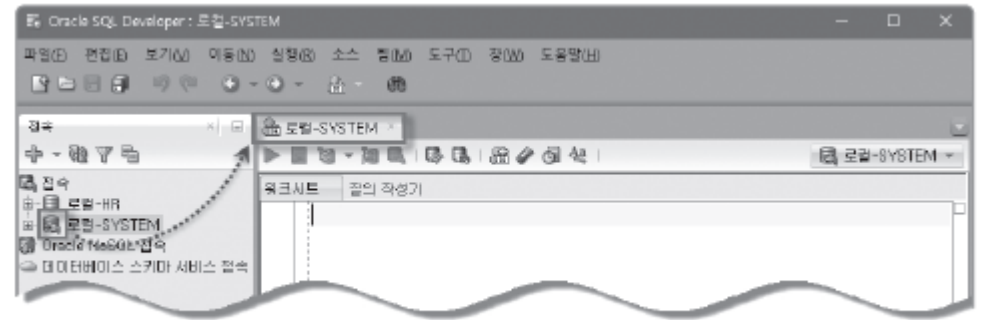
'쇼핑몰(Shop) 스키마를 생성해 보자.

step 0

SQL Developer를 실행하자.



[그림 3-4] Oracle XE 서버에 연결

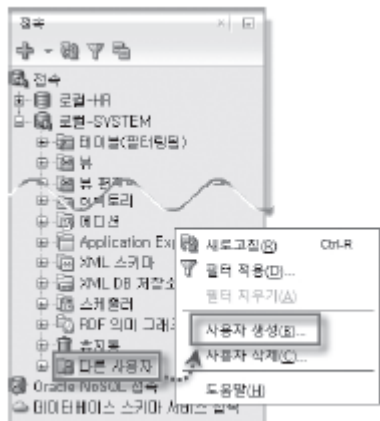


[그림 3-6] 연결된 접속 확인

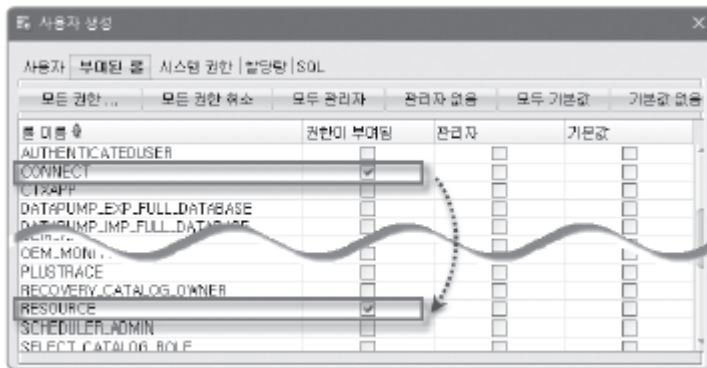
SECTION 02 Oracle을 이용한 스키마 구축 절차

step 1

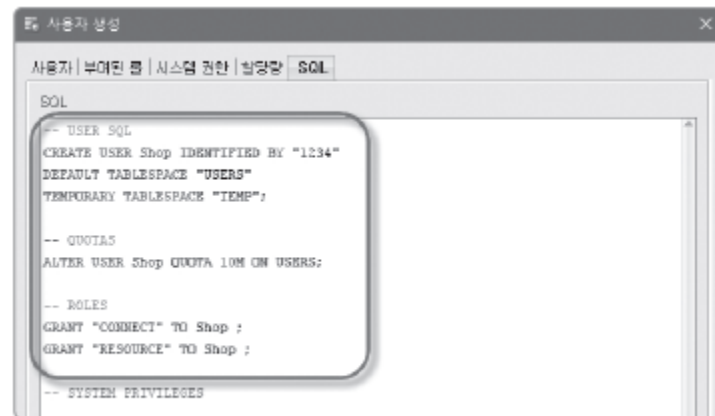
스키마(=사용자)를 생성하자. [그림 3-1]의 쇼핑몰(Shop) 스키마를 생성하자.



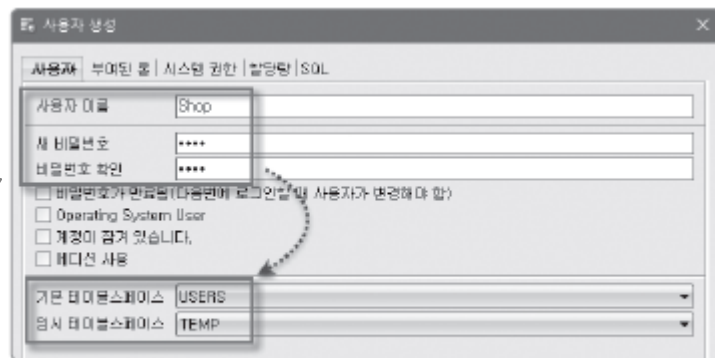
[그림 3-7] 스키마 생성 1



[그림 3-9] 스키마 생성 3



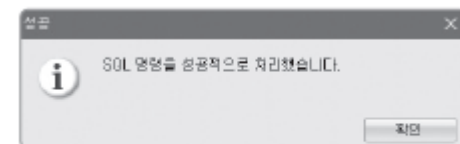
[그림 3-11] 스키마 생성 5



[그림 3-8] 스키마 생성 2



[그림 3-10] 스키마 생성 4

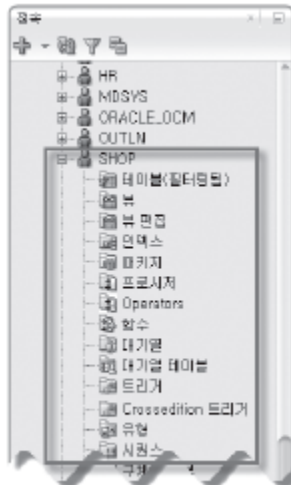


[그림 3-12] 스키마 생성 6

SECTION 02 Oracle을 이용한 스키마 구축 절차



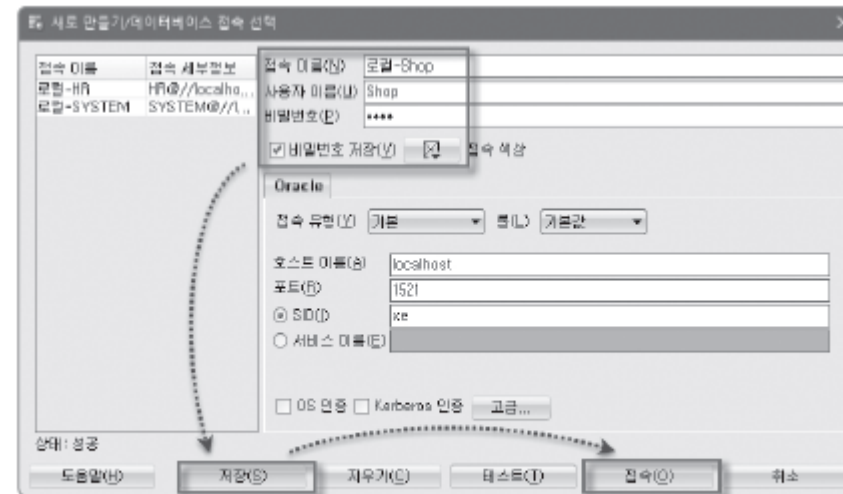
[그림 3-13] 스키마 생성 7



[그림 3-14] 스키마 생성 8

step 2

이제 새로운 스키마인 SHOP 스키마에 접속을 생성해 놓자.



[그림 3-15] 새로운 접속 생성



[그림 3-16] 기존 접속 해제

SECTION 02 Oracle을 이용한 스키마 구축 절차

2.2 테이블 생성

실습2

테이블을 생성하자.

step 0

테이블을 생성하기 전에 [그림 3-1]에는 나타나 있지 않은 각 열의 영문 이름 및 데이터 유형을 결정하자.
이 과정은 원래 데이터베이스 모델링 (특히 물리적 모델링) 시에 결정된다.

열 이름(한글)	영문 이름	데이터 유형	크기	널(Null) 아님
아이디	memberID	문자(CHAR)	8글자(영문)	O
회원 이름	memberName	문자(NCHAR)	5글자(한글)	O
주소	memberAddress	문자(NVARCHAR2)	20글자(한글)	X

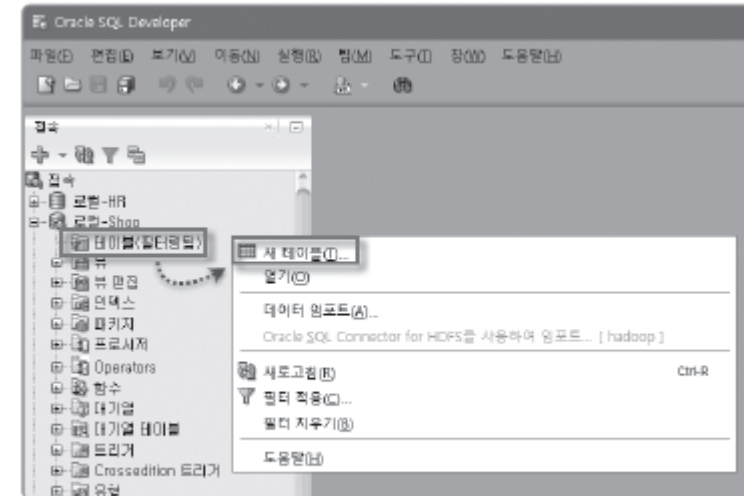
[표 3-1] 회원 테이블 정의

열 이름(한글)	영문 이름	데이터 유형	크기	널(Null) 아님
제품 이름	productName	문자(NCHAR)	4글자(한글)	O
가격	cost	숫자(NUMBER)	정수(최대 7자리)	O
제조일자	makeDate	날짜(DATE)	날짜형	X
제조회사	company	문자(NCHAR)	5글자(한글)	X
남은 수량	amount	숫자(NUMBER)	정수(최대 3자리)	O

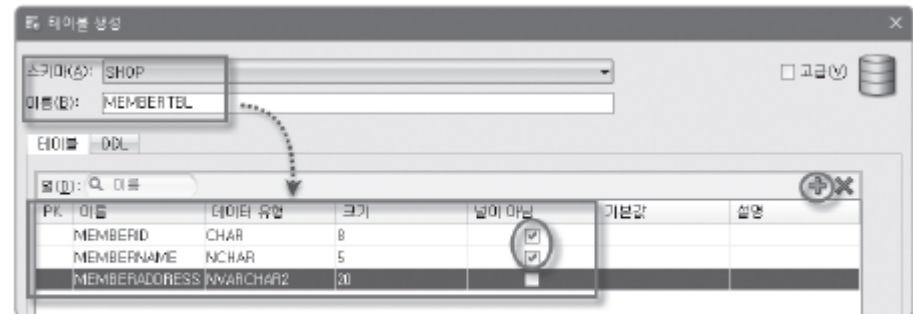
[표 3-2] 제품 테이블 정의

step 1

바로 앞 step 0 의 설계대로 회원 테이블(memberTBL)을 만들어 보자.



[그림 3-17] 테이블 생성 1



[그림 3-18] 테이블 생성 2

SECTION 02 Oracle을 이용한 스키마 구축 절차

PK	이름	데이터 유형	크기	Null이 아님	기본값	설명
<input checked="" type="checkbox"/>	MEMBERID	CHAR	8	<input checked="" type="checkbox"/>		
<input checked="" type="checkbox"/>	MEMBERNAME	NCHAR	5	<input checked="" type="checkbox"/>		
<input type="checkbox"/>	MEMBERADDRESS	NVARCHAR2	20	<input type="checkbox"/>		

[그림 3-19] 테이블 생성 3



[그림 3-20] 테이블 생성 확인

step 2

같은 방식으로 [표 3-2]의 제품 테이블(productTBL)을 독자가 직접 만들고 저장하자. (productName을 PK로 지정해야 한다.) 결과는 다음과 같다. <확인>을 클릭해서 테이블 생성 창을 닫는다.

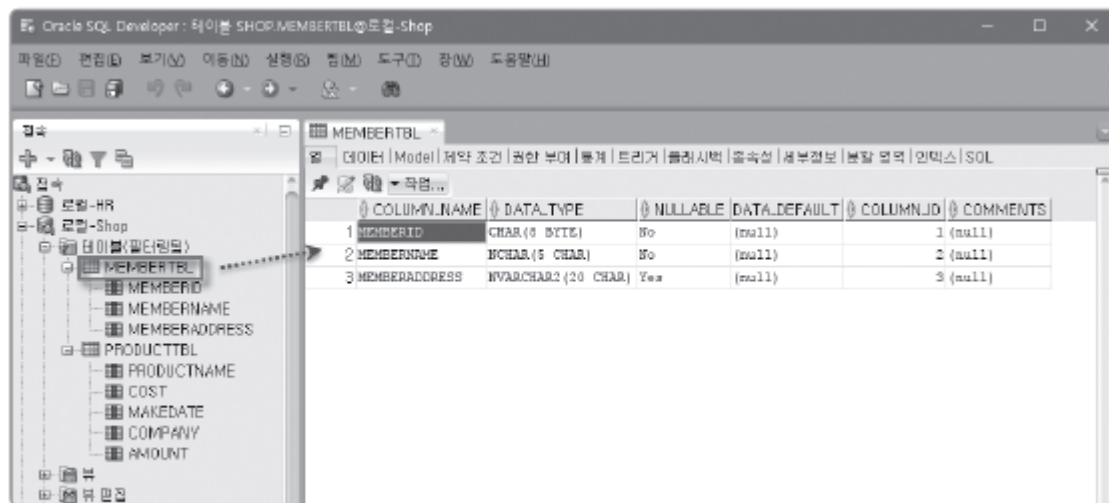
PK	이름	데이터 유형	크기	Null이 아님	기본값	설명
<input checked="" type="checkbox"/>	PRODUCTNAME	NCHAR	4	<input checked="" type="checkbox"/>		
<input checked="" type="checkbox"/>	COST	NUMBER	7	<input checked="" type="checkbox"/>		
<input type="checkbox"/>	MAKEDATE	DATE		<input type="checkbox"/>		
<input type="checkbox"/>	COMPANV	NCHAR	5	<input type="checkbox"/>		
<input checked="" type="checkbox"/>	AMOUNT	NUMBER	3	<input checked="" type="checkbox"/>		

[그림 3-21] 제품 테이블 생성

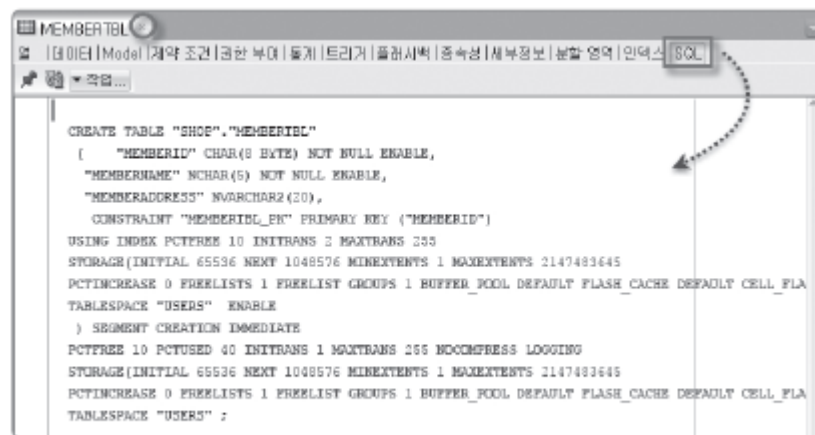
SECTION 02 Oracle을 이용한 스키마 구축 절차

step 3

생성한 테이블을 모두 확인해 보자.



[그림 3-22] 생성된 테이블 확인



[그림 3-23] SQL문 확인

SECTION 02 Oracle을 이용한 스키마 구축 절차

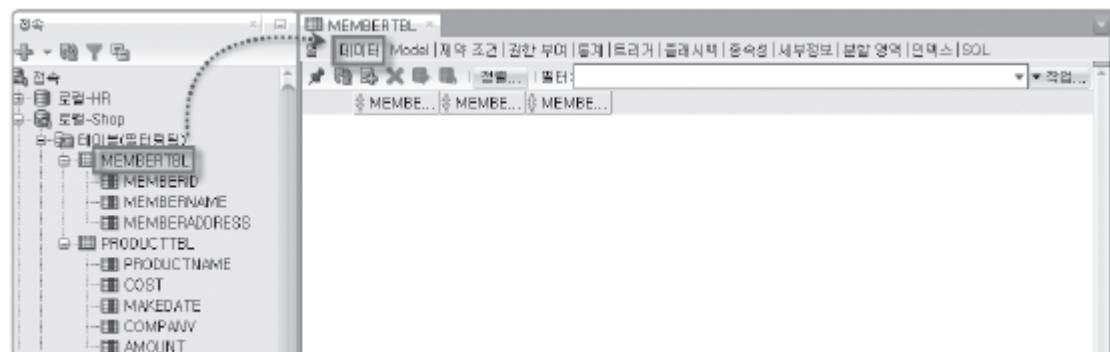
2.3 데이터 입력

실습3

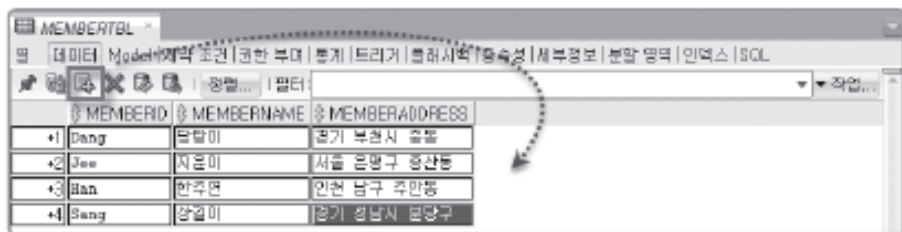
행 데이터를 입력하자.

step 1

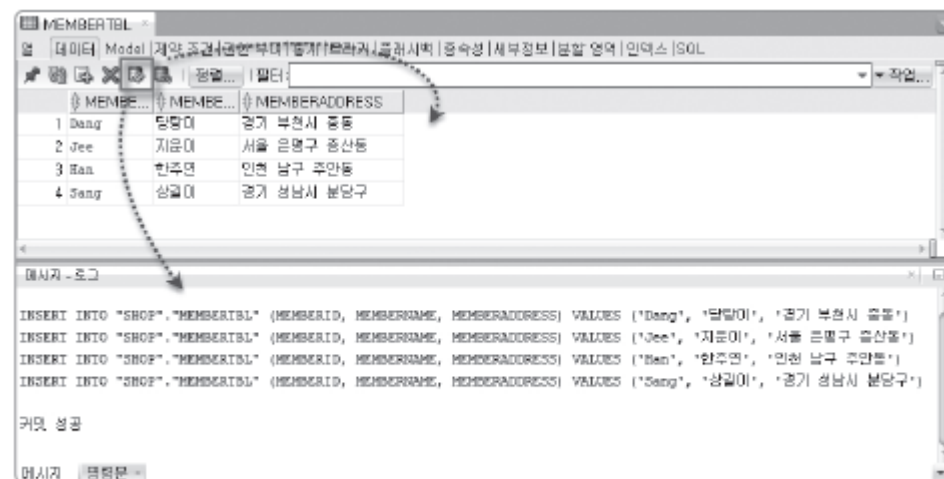
회원 테이블의 데이터를 입력하자.



[그림 3-24] 행 데이터 입력 1



[그림 3-25] 행 데이터 입력 2



[그림 3-26] 행 데이터 입력 3

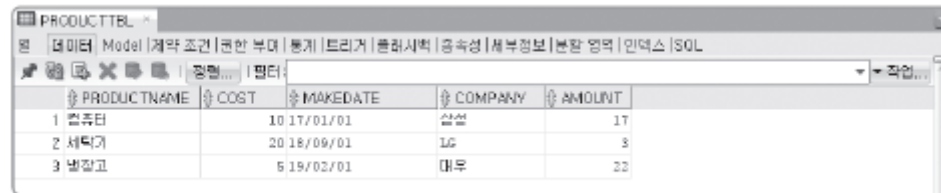


[그림 3-27] 행 데이터 삭제

SECTION 02 Oracle을 이용한 스키마 구축 절차

step 2

동일한 방식으로 제품 테이블 (productTBL)에 [그림 3-1]의 데이터를 입력한다. 입력된 결과는 다음과 같다. 변경내용을 커밋시킨다.



The screenshot shows the Oracle SQL Developer interface for the PRODUCTTBL table. The table has columns: PRODUCTNAME, COST, MAKEDATE, COMPANY, and AMOUNT. The data is as follows:

	PRODUCTNAME	COST	MAKEDATE	COMPANY	AMOUNT
1	컴퓨터	10	17/01/01	삼성	17
2	서탁기	20	18/09/01	LG	8
3	냉장고	5	19/02/01	대우	22

[그림 3-28] 행 데이터 입력 4

step 3

입력이 끝났으면 테이블을 켜고 [PRODUCTTBL] 창을 닫는다.

SECTION 02 Oracle을 이용한 스키마 구축 절차

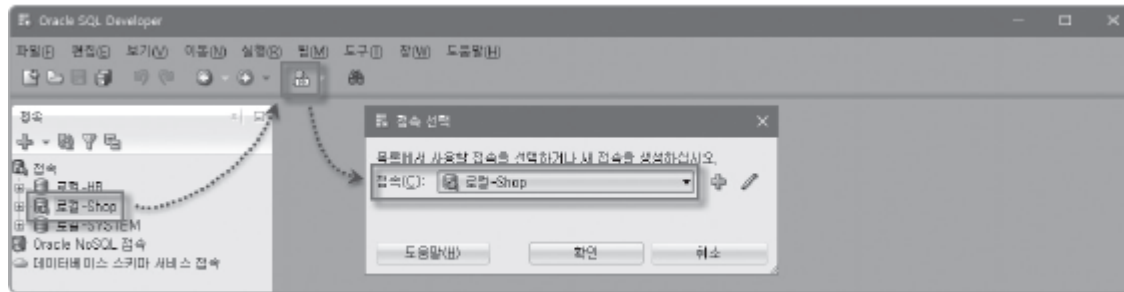
2.4 데이터 활용

실습4

데이터를 활용한다는 것은 주로 'SELECT'문을 사용한다는 의미이다. 이 SELECT는 앞으로 계속 나오게 될 것이며 특히, 6장과 7장에서 더 자세히 배우게 될 것이다.

step 0

SQL문을 직접 입력할 수 있는 워크시트를 열자.

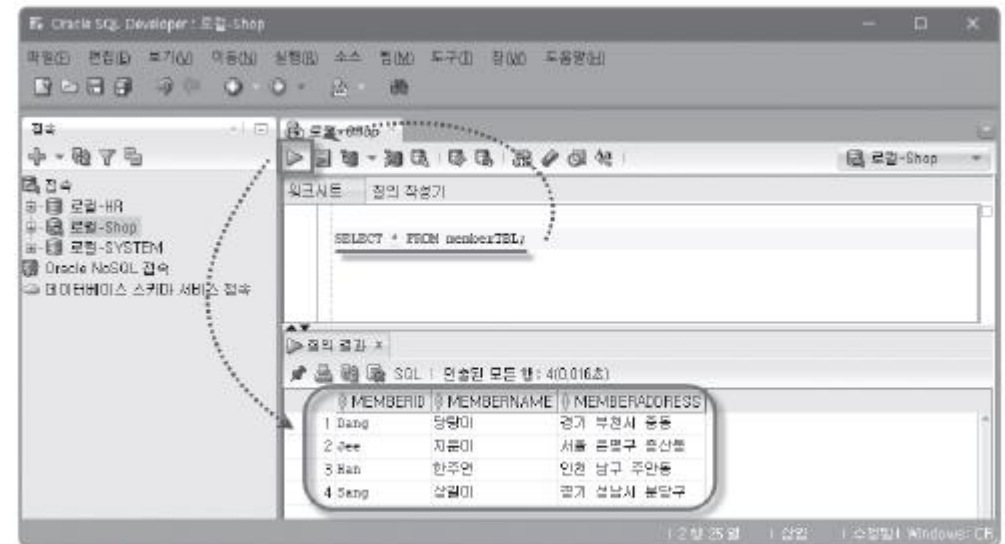


[그림 3-29] 새 워크시트 열기

step 1

기본적인 SELECT문을 몇 가지 사용해 보자.

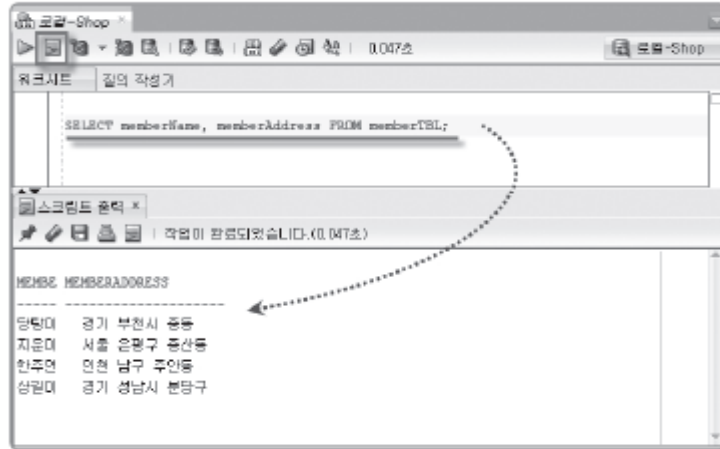
SELECT * FROM memberTBL;



[그림 3-30] SELECT 활용 1

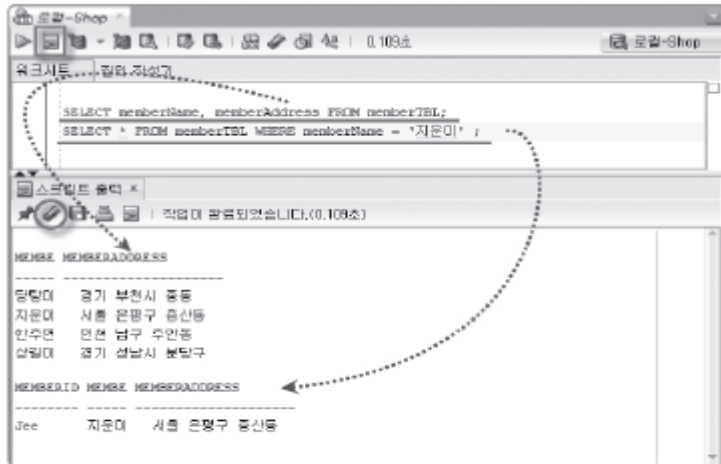
SECTION 02 Oracle을 이용한 스키마 구축 절차

SELECT memberName, memberAddress FROM memberTBL;



[그림 3-32] SELECT 활용 2

SELECT * FROM memberTBL WHERE memberName = '지운이' ;

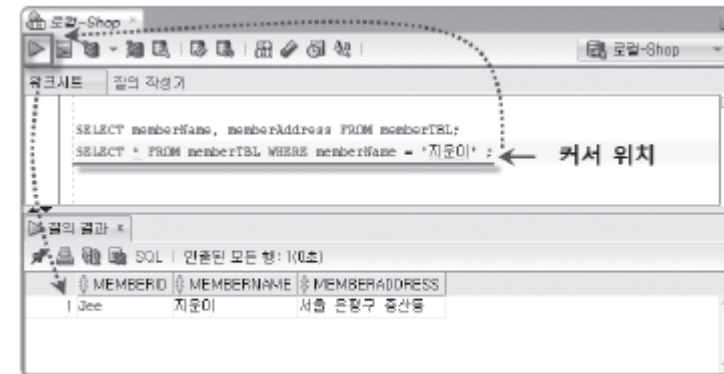


[그림 3-33] SELECT 활용 3

[[스크립트 출력] 창의 <지우기> 아이콘을 클릭 결과 지우기]



[그림 3-34] SELECT 활용 4



[그림 3-35] SELECT 활용 5

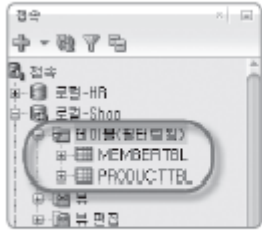
SECTION 02 Oracle을 이용한 스키마 구축 절차

step 2

SQL문으로 새로운 테이블을 하나 더 생성하자.

```
CREATE TABLE "my testTBL" (id NUMBER(3));
```

['my testTBL' 생성]



[그림 3-36] [접속] 창에는 적용이 안됨



[그림 3-37] 새로 고침

step 3

그런데, 자세히 보면 테이블에 영문 대소문자가 섞여 있다. Oracle은 기본적으로 테이블 등의 이름을 모두 대문자로 처리하지만, 큰 따옴표(")안에 이름을 넣을 경우에 그대로 이름을 사용한다.

```
DROP TABLE "MY TESTTBL";
```

[DROP TABLE문을 사용 삭제-오류(대소문자)]

```
DROP TABLE "my testTBL";
```

[DROP TABLE문을 사용 삭제-정상삭제(대소문자)]

SECTION 03 테이블 외의 데이터베이스 개체의 활용

3.1 인덱스

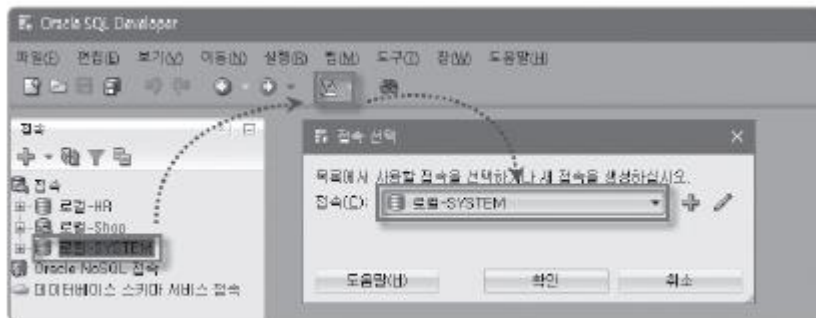
- 인덱스란 대부분의 책의 제일 뒤에 붙어 있는 '찾아보기'(또는 색인)와 같은 개념임.

실습5

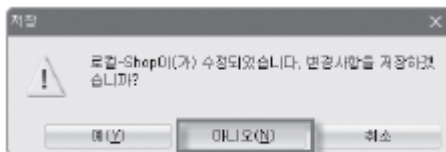
인덱스를 간단히 사용해 보자.

step 1

인덱스의 사용 여부를 확인하기 위해서는 데이터의 양이 어느 정도 있어야 의미가 있다. 적정량이 있는 테이블을 우선 생성하자.

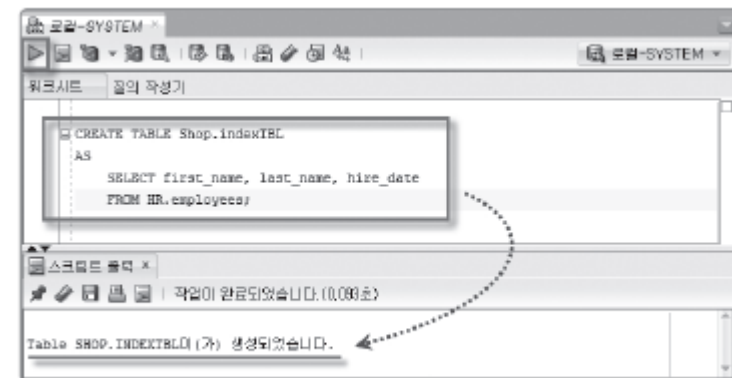


[그림 3-38] 새 워크시트 열기



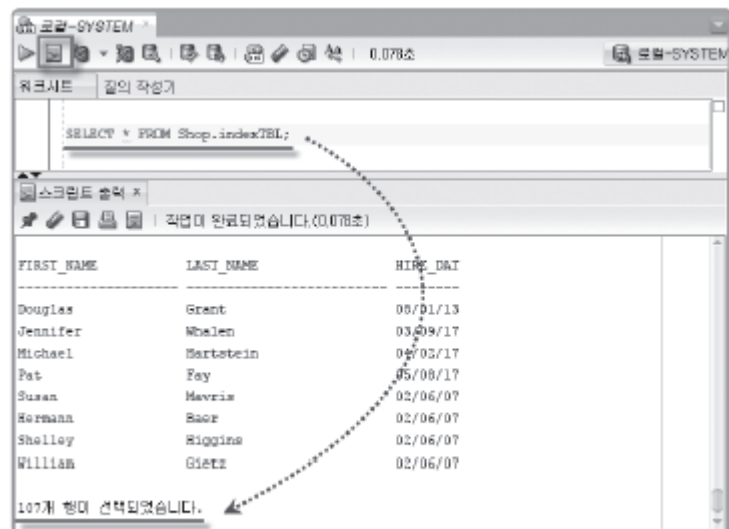
[그림 3-39] 쿼리 내용을 저장하지 않음

```
CREATE TABLE Shop.indexTBL
AS
SELECT first_name, last_name, hire_date
FROM HR.employees;
```



[그림 3-40] Shop 스키마에 indexTBL 생성

SECTION 03 테이블 외의 데이터베이스 개체의 활용



[그림 3-41] Shop 스키마의 indexTBL 확인

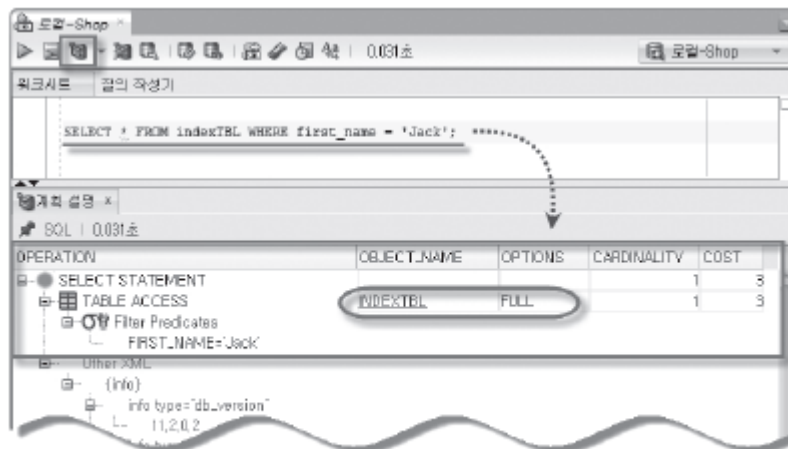
step 2

먼저 인덱스가 없는 상태에서 쿼리가 어떻게 작동하는지를 확인하자.

```
SELECT * FROM indexTBL WHERE first_name = 'Jack';
```

FIRST_NAME	LAST_NAME	HIRE_DATE
Jack	Livingston	06/04/23

[그림 3-42] 조회 결과



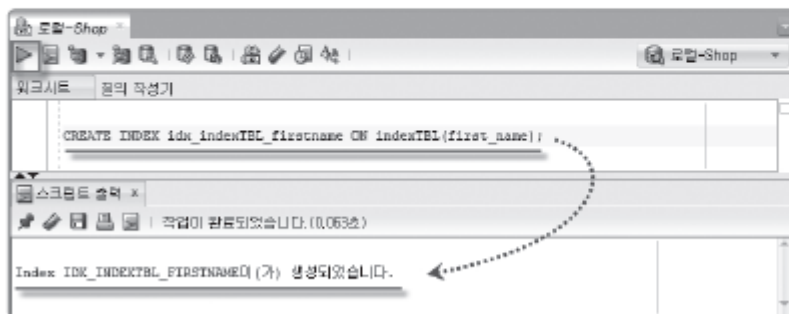
[그림 3-43] 인덱스 생성 전의 [계획 설명] 확인

SECTION 03 테이블 외의 데이터베이스 개체의 활용

step 3

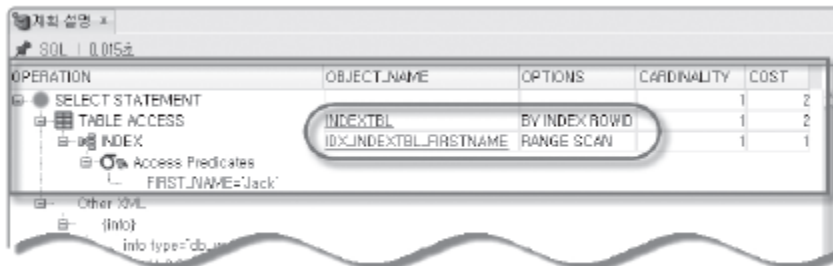
다음 SQL을 실행해서 테이블(indexTBL)의 이름(first_name) 열에 인덱스를 생성해 보자.

```
CREATE INDEX idx_indexTBL_firstname ON indexTBL(first_name);
```



[그림 3-44] 인덱스 생성

```
SELECT * FROM indexTBL WHERE first_name = 'Jack';
```

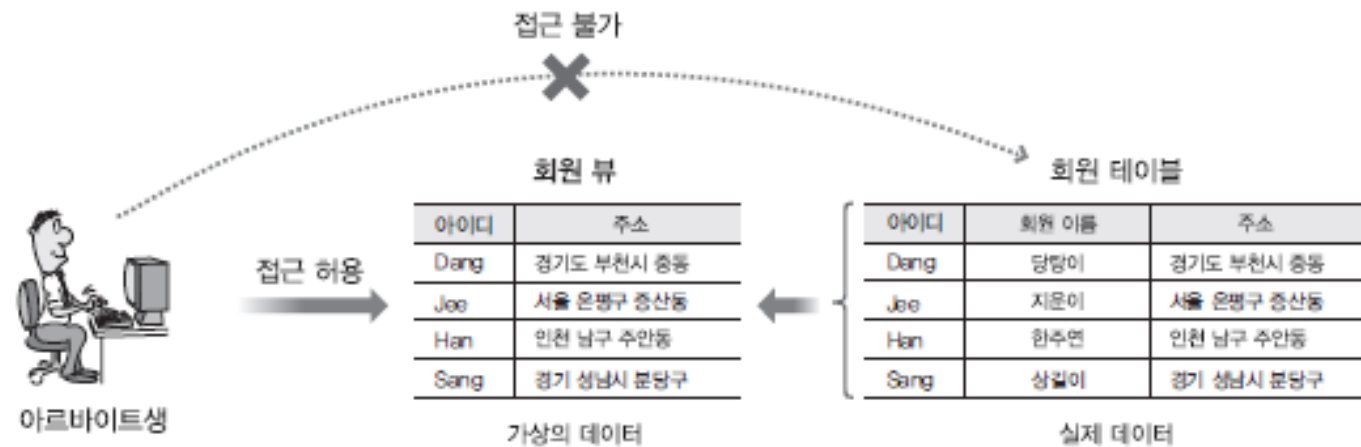


[그림 3-45] 인덱스 생성 후의 실행 계획 확인

SECTION 03 테이블 외의 데이터베이스 개체의 활용

3.2 뷰

- 뷰란 가상의 테이블이라고 생각하면 됨.



[그림 3-46] 뷰의 사용 예

SECTION 03 테이블 외의 데이터베이스 개체의 활용

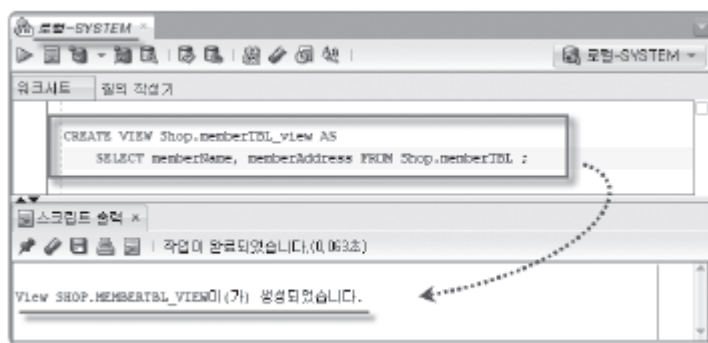
실습6

기본적인 뷰의 사용법을 실습하자.

step 1

회원 이름과 주소만 존재하는 뷰를 생성하자. 뷰 이름은 memberTBL_view로 주자.

```
CREATE VIEW Shop.memberTBL_view AS  
SELECT memberName, memberAddress FROM Shop.memberTBL ;
```

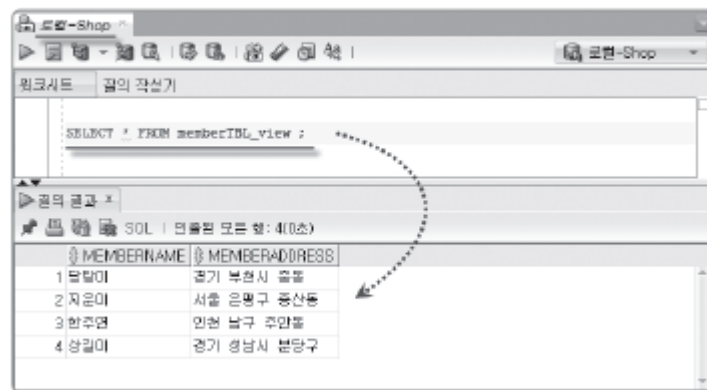


[그림 3-47] 뷰 생성

step 2

뷰를 조회해 보자.

```
SELECT * FROM memberTBL_view ;
```



[그림 3-48] 뷰의 조회

SECTION 03 테이블 외의 데이터베이스 개체의 활용

3.3 스토어드 프로시저

- 스토어드 프로시저란 Oracle에서 제공해주는 프로그래밍 기능을 말함.

실습7

간단한 스토어드 프로시저를 실습하자. 회원 테이블과 제품 테이블의 행 개수의 합계를 출력해 보자.

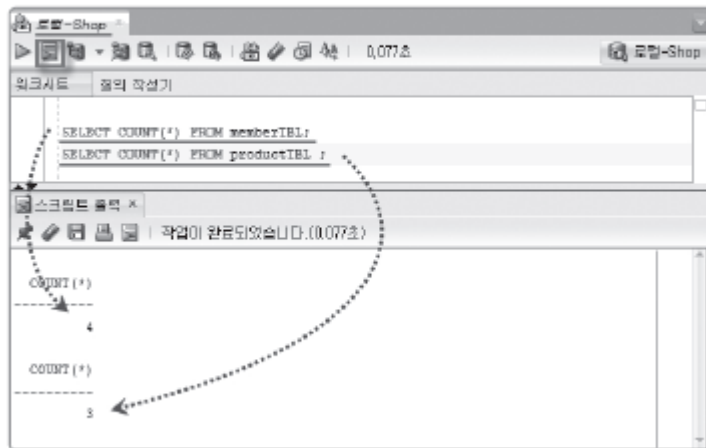
step 0

다시 기존의 워크시트를 닫고, [로컬-Shop]의 새 워크시트를 연다.

step 1

지금까지 배운 SQL문으로는 다음과 같이 동시에 수행하게 될 것이다. <스크립트 실행> 아이콘이나 **F5**를 눌러서 실행하자.

```
SELECT COUNT(*) FROM memberTBL;  
SELECT COUNT(*) FROM productTBL ;
```

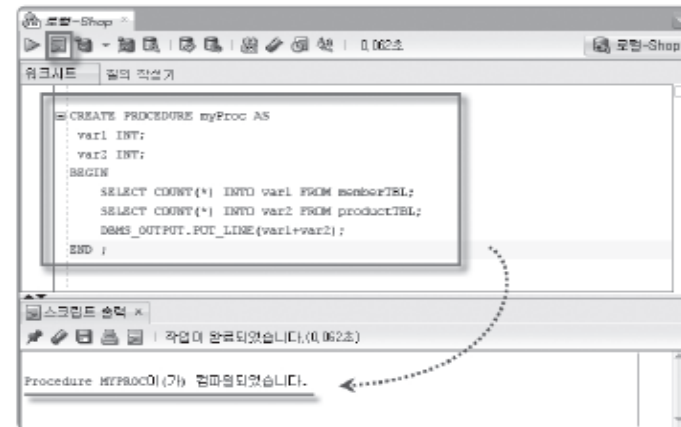


[그림 3-49] 여러 테이블의 조회

step2

이 두 쿼리를 하나의 스토어드 프로시저 안에서 계산까지 완료해 보자.

```
CREATE PROCEDURE Shop.myProc AS -- Shop 스키마에 생성함  
var1 INT;  
var2 INT;  
BEGIN  
    SELECT COUNT(*) INTO var1 FROM Shop.memberTBL;  
    SELECT COUNT(*) INTO var2 FROM Shop.productTBL;  
    DBMS_OUTPUT.PUT_LINE(var1+var2);  
END ;
```



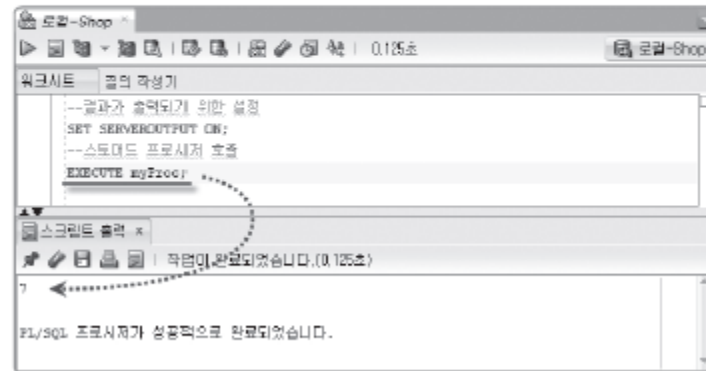
[그림 3-50] 스토어드 프로시저 생성

SECTION 03 테이블 외의 데이터베이스 개체의 활용

step 3

앞으로는 방금 생성한 스토어드 프로시저를 실행하기만 하면 된다.

```
-- 결과가 출력되기 위한 설정  
SET SERVEROUTPUT ON;  
-- 스토어드 프로시저 호출  
EXECUTE myProc;
```



[그림 3-51] 스토어드 프로시저 실행

SECTION 03 테이블 외의 데이터베이스 개체의 활용

3.4 트리거

- 트리거란 테이블에 부착되어서, 테이블에 INSERT나 UPDATE 또는 DELETE 작업이 발생되면 실행되는 코드를 말함.

실습8

가장 일반적으로 사용되는 트리거의 용도를 실습해 보자.

step 0

먼저 데이터를 입력하고 수정하고 삭제하는 SQL문을 연습해보자. 앞의 <실습 3>과 같이 SQL Developer의 GUI 모드에서도 가능하지만, 실무적으로는 지금 배울 SQL을 훨씬 많이 사용하게 될 것이다. (상세한 것은 6장에서 나올 것이며 지금은 기본적인 것만 연습해 보자.) 먼저, 기존 워크시트를 닫고 [로컬-Shop]의 새 워크시트를 열자.

```
INSERT INTO memberTBL VALUES ('Figure', '연아', '경기도 군포시 당정동');
```

```
UPDATE memberTBL SET memberAddress = '서울 강남구 역삼동' WHERE memberName = '연아';
```

```
DELETE FROM memberTBL WHERE memberName = '연아';
```

step 1

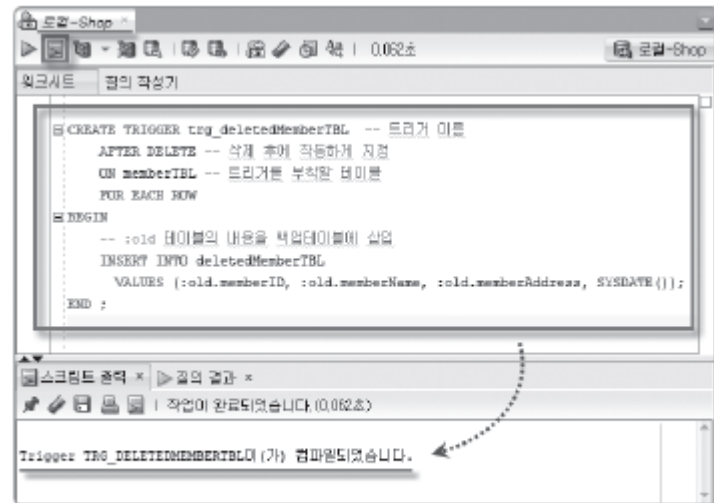
위와 같은 사례를 방지하기 위해서 회원 테이블에서 행 데이터를 삭제할 경우에 다른 테이블에 지워진 데이터와 더불어 지워진 날짜까지 기록해 보자.

```
CREATE TABLE deletedMemberTBL (  
    memberID char(8),  
    memberName nchar(5),  
    memberAddress nvarchar2(20),  
    deletedDate date -- 삭제한 날짜  
);
```

[지워진 데이터를 보관할
테이블(deletedMemberTBL)을 만들기]

SECTION 03 테이블 외의 데이터베이스 개체의 활용

```
CREATE TRIGGER trg_deletedMemberTBL -- 트리거 이름
AFTER DELETE -- 삭제 후에 작동하게 지정
ON memberTBL -- 트리거를 부착할 테이블
FOR EACH ROW -- 각 행마다 적용됨
BEGIN
    -- :old 테이블의 내용을 백업 테이블에 삽입
    INSERT INTO deletedMemberTBL
    VALUES (:old.memberID, :old.memberName, :old.memberAddress, SYSDATE());
END ;
```



[그림 3-52] 트리거 생성

step 2

회원 테이블의 데이터를 삭제해 보고 삭제된 데이터가 백업 테이블에 들어가는지 확인해 보자.

```
SELECT * FROM memberTBL;
```

```
DELETE FROM memberTBL WHERE memberName = '당탕이';
```

```
SELECT * FROM memberTBL;
```

MEMBERID	MEMBERNAME	MEMBERADDRESS
1	Jee	지문이
2	Ban	한주연
3	Sang	상권이

[그림 3-53] 데이터가 삭제된 것을 확인

```
SELECT * FROM deletedMemberTBL;
```

MEMBERID	MEMBERNAME	MEMBERADDRESS	DELETEDDATE
1	Dang	당탕이	17/09/22

[그림 3-54] 삭제된 데이터가 백업 테이블에 들어가 있음

SECTION 04 데이터의 백업 및 복원

4.1 백업과 복원

- 백업은 현재의 스키마를 다른 매체에 보관하는 작업을 말하며, 복원은 데이터베이스에 문제가 발생시에 다른 매체에 백업된 데이터를 이용해서 원상태로 돌려놓는 작업을 말함.

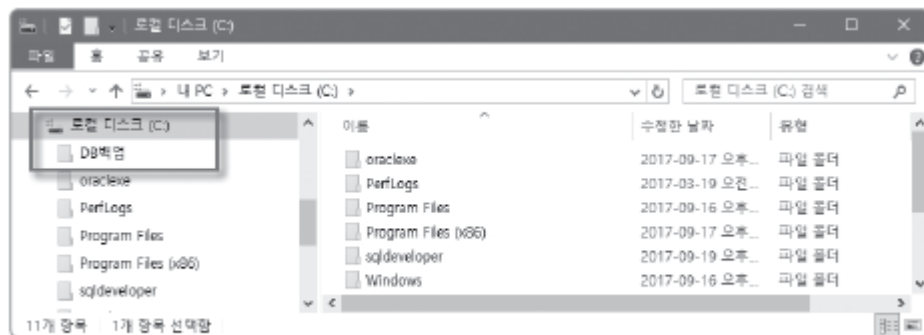
실습9

간단한 백업과 복원을 실습하자.

쇼핑몰 스키마를 백업 (Backup 또는 Export) 받은 후에, 실수로 데이터를 모두 삭제했을 경우에 원상태로 복원시켜 보자.

step 0

실제로 백업을 받는다면, 현재의 스키마가 저장된 디스크에 백업을 받는다는 것은 별 의미가 없다. 디스크가 깨진다면, 어차피 백업을 받은 것까지 다 날아가기 때문이다. 그러므로, 백업은 다른 디스크에 백업을 받아야만 의미가 있다.

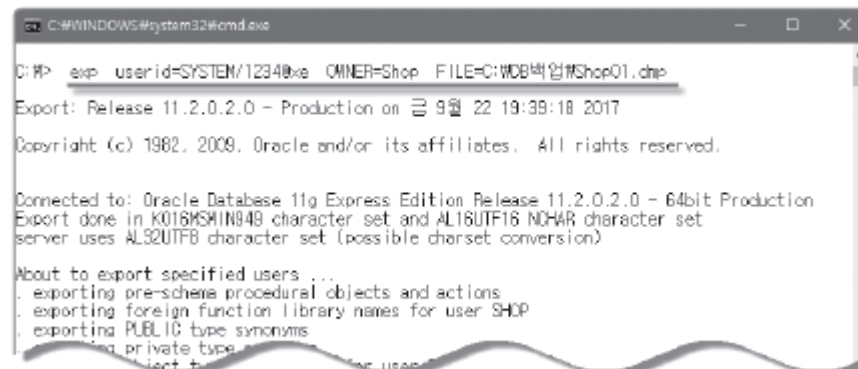


[그림 3-55] 백업용 폴더 생성

step 1

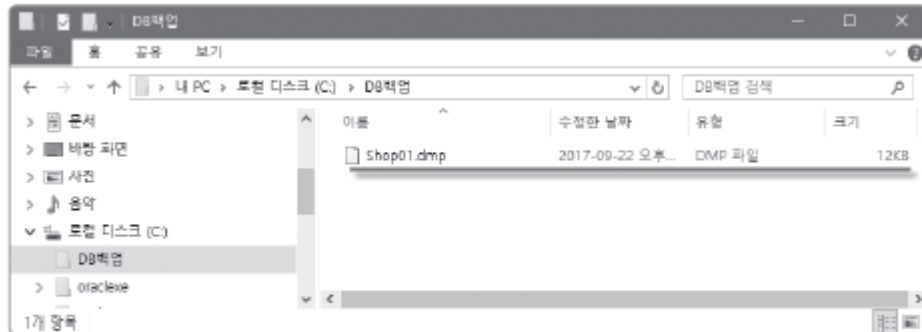
쇼핑몰 스키마(Shop)를 백업하자.

```
exp userid=SYSTEM/1234@xe OWNER=Shop FILE=C:\DB\Backup\Shop01.dmp
```



[그림 3-56] Shop 스키마 전체를 백업

SECTION 04 데이터의 백업 및 복원



[그림 3-57] Shop 스키마가 백업된 파일

step 4

Shop 스키마의 데이터를 다시 확인해 보자.

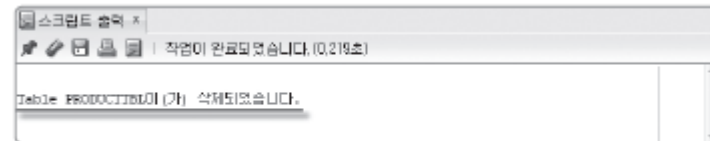
0 SQL Developer를 실행해서 [로컬-Shop]의 새 워크시트를 하나 연다.

1 **SELECT * FROM productTBL**문으로 데이터를 확인해 보자. 원래대로 데이터가 잘 보일 것이다.

2 SQL Developer 및 명령 프롬프트를 종료한다.

step 2

Shop 스키마의 데이터를 삭제하는 실수를 저질러 보자.

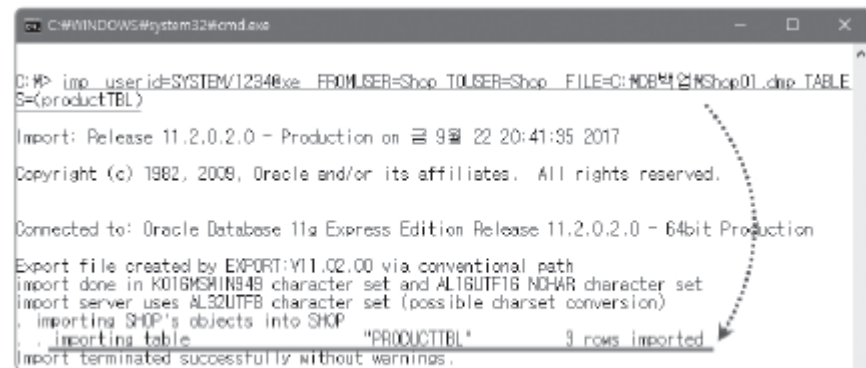


[그림 3-58] 테이블 삭제

step 3

백업한 파일로 테이블을 원상 복구해 보자.

```
imp userid=SYSTEM/1234@xe FROMUSER=Shop TOUSER=Shop FILE=C:\DB백업\Shop01.dmp  
TABLES=(productTBL)
```



[그림 3-59] Shop 스키마의 productTBL 테이블 복원

SECTION 05 Oracle과 응용프로그램의 연결

- 웹과 스키마가 연동되는 것을 실습.

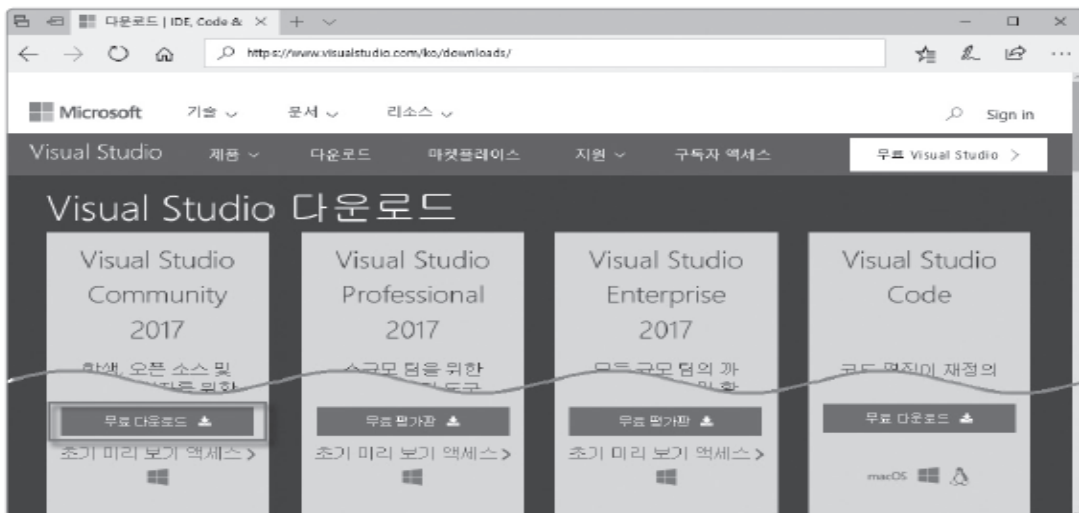
실습10

Oracle에서 구축한 쇼핑몰 데이터베이스를 웹에서 서비스하자.

step 1

개발 툴로 사용할 Microsoft Visual Studio Community 2017의 설치 파일을 다운로드하자.

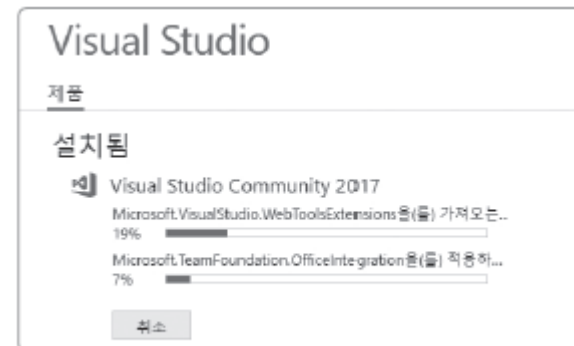
웹 브라우저 : <https://www.visualstudio.com/ko/downloads/>
책의 사이트 : (<http://cafe.naver.com/thisisOracle>)에서
Visual Studio Community 2017(vs_Community.exe, 1MB)을 다운로드



[그림 3-60] Visual Studio Community 2017 다운로드

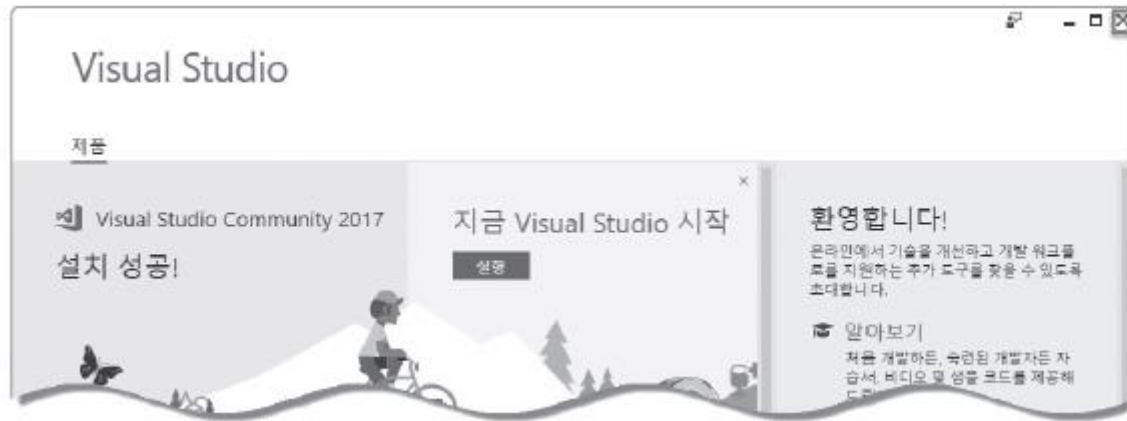


[그림 3-61] Visual Studio Community 설치 1



[그림 3-62] Visual Studio Community 설치 2

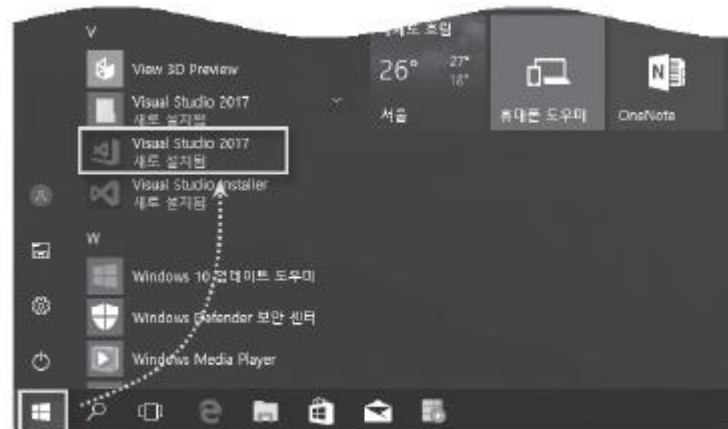
SECTION 05 Oracle과 응용프로그램의 연결



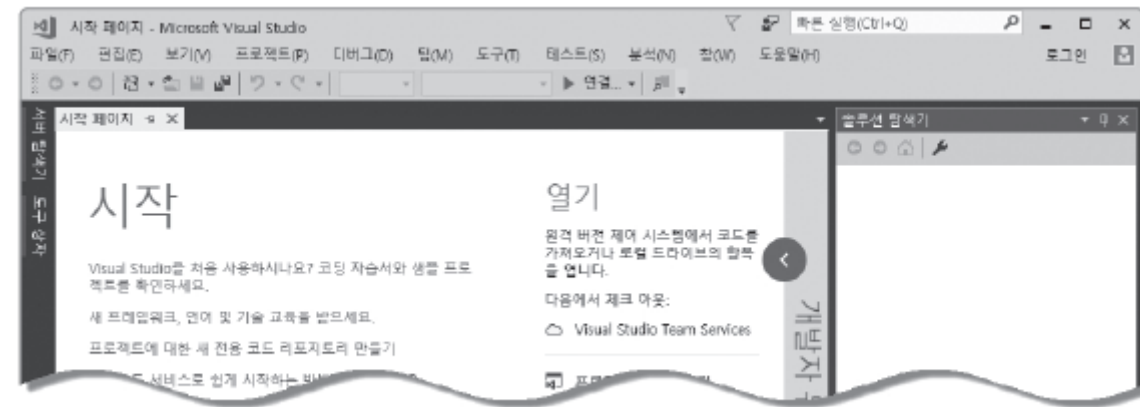
[그림 3-63] Visual Studio Community 설치 3



[그림 3-65] Visual Studio Community 실행 2



[그림 3-64] Visual Studio Community 실행 1



[그림 3-66] Visual Studio Community 실행 화면

SECTION 05 Oracle과 응용프로그램의 연결

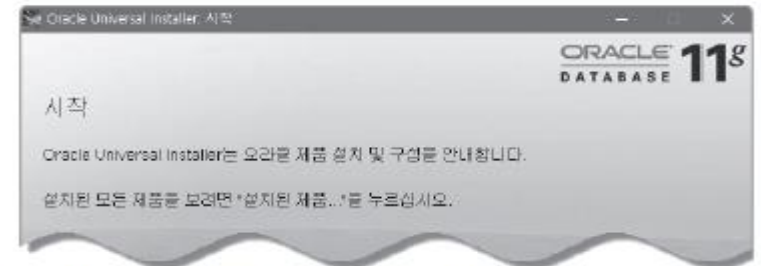
step 2

Visual Studio에서 Oracle에 연결하기 위해서는 Oracle Developer Tools를 설치해야 한다.

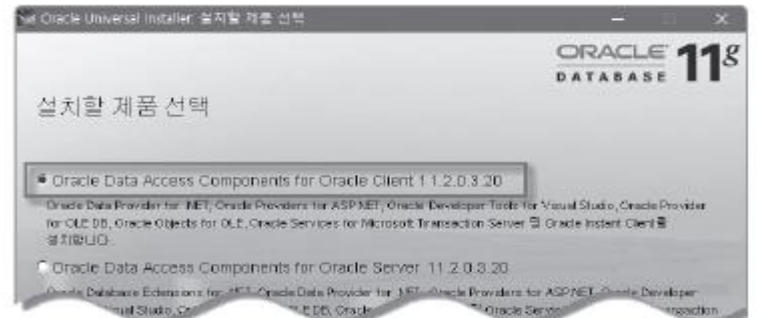
<http://www.oracle.com/technetwork/topics/dotnet/utilsoft-086879.html>
에서 'ODAC 11.2 Release 5 and Oracle Developer Tools for Visual Studio (11.2.0.3.20)' (파일명: ODTwithODAC1120320_32bit.zip, 224MB)를 다운로드



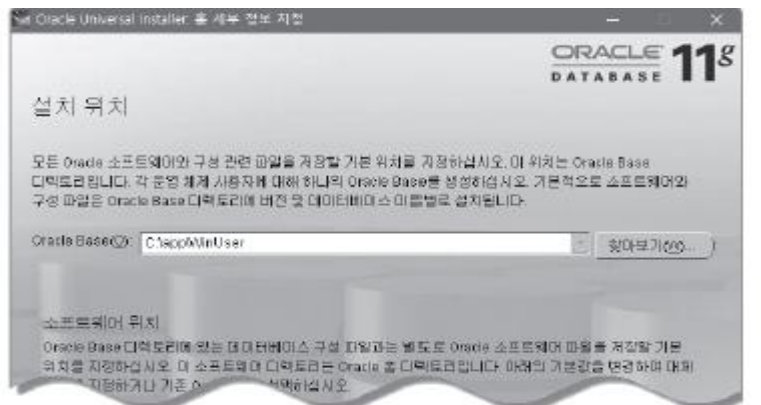
[그림 3-67] Oracle Developer Tools 다운로드



[그림 3-68] Oracle Developer Tools 설치 1

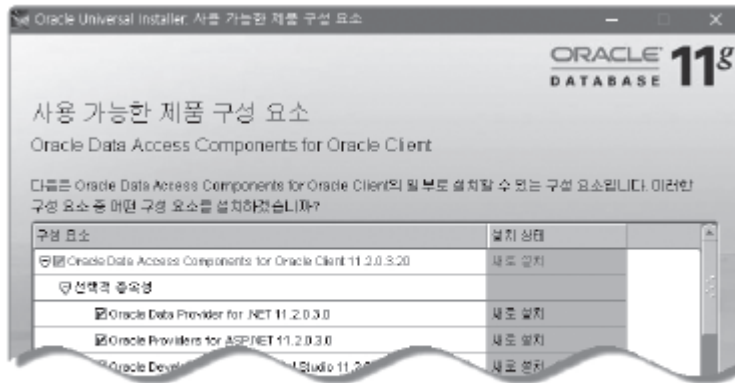


[그림 3-69] Oracle Developer Tools 설치 2



[그림 3-70] Oracle Developer Tools 설치 3

SECTION 05 Oracle과 응용프로그램의 연결



[그림 3-71] Oracle Developer Tools 설치 4



[그림 3-72] Oracle Developer Tools 설치 5

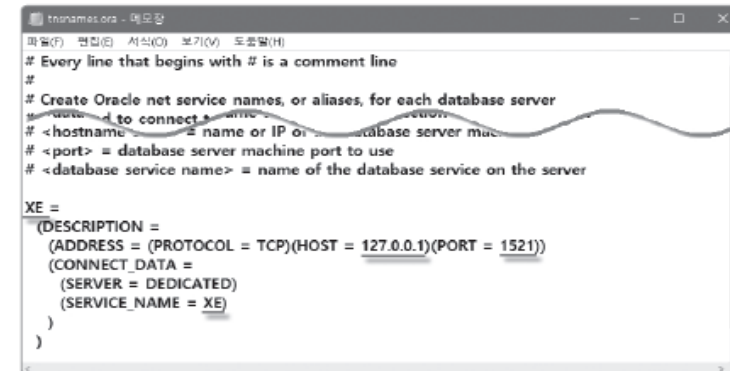
step3

서버와 접속하는 tnsnames.ora 파일을 설정해야 한다.



[그림 3-73] tnsnames.ora 파일 편집 1

수정 전	수정 후
<data source alias>	XE
<hostname or IP>	127.0.0.1
<port>	1521
<data service name>	XE



[그림 3-74] tnsnames.ora 파일 편집 2

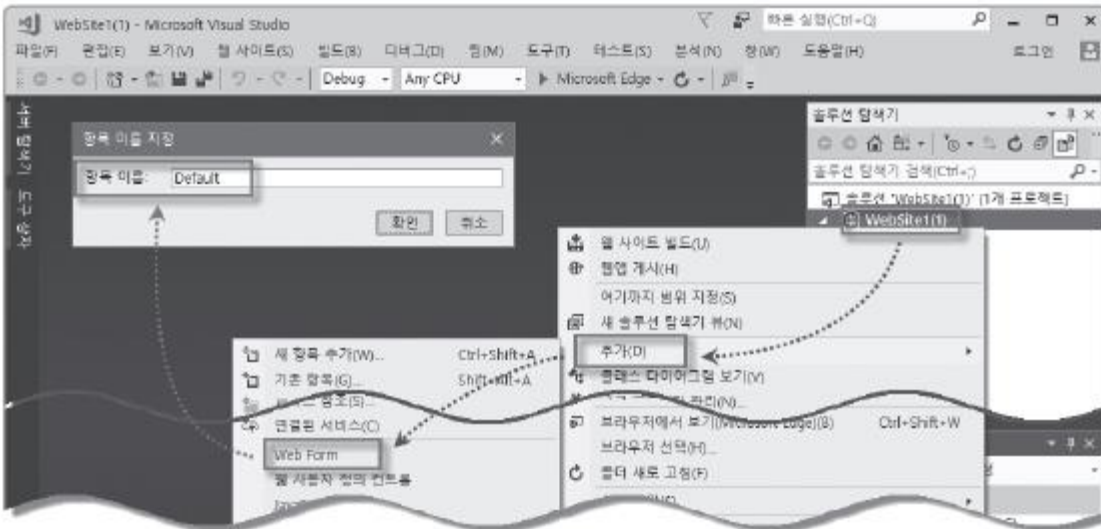
SECTION 05 Oracle과 응용프로그램의 연결

step 4

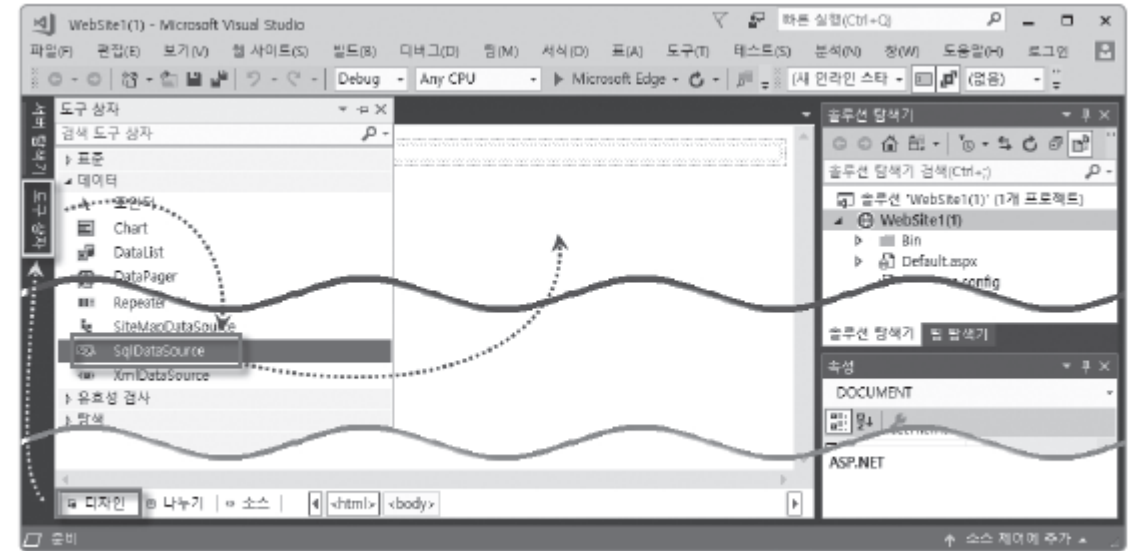
ASP.NET 웹 응용프로그램을 작성하자.



[그림 3-75] Microsoft Visual Studio의 웹 사이트 구축 1



[그림 3-76] Microsoft Visual Studio의 웹 사이트 구축 2

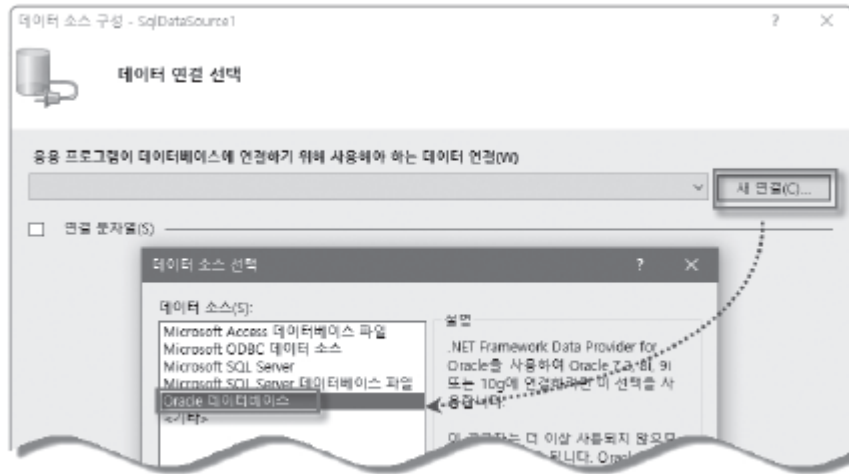


[그림 3-77] Microsoft Visual Studio의 웹 사이트 구축 3

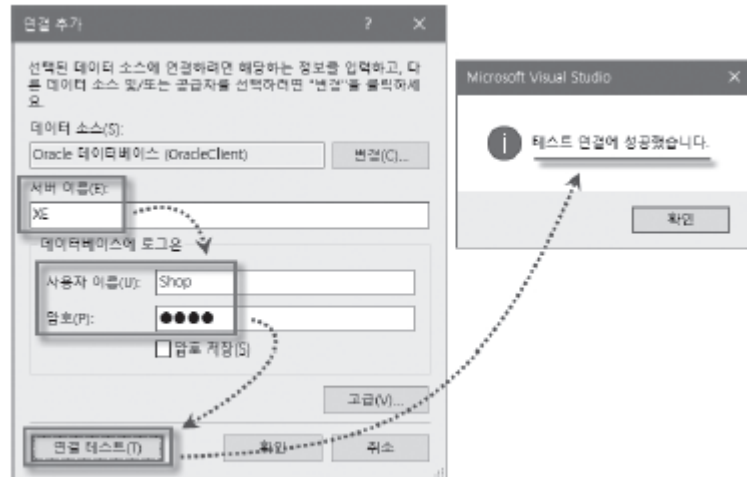


[그림 3-78] Microsoft Visual Studio의 웹 사이트 구축 4

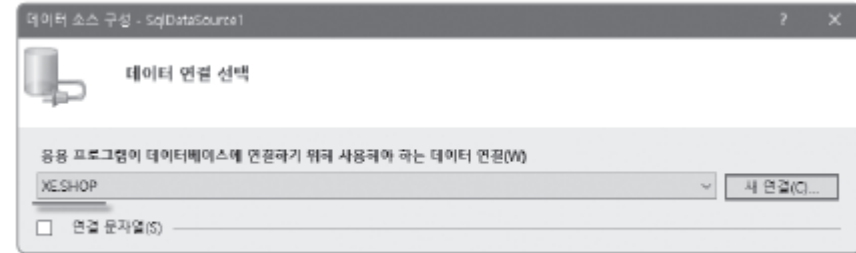
SECTION 05 Oracle과 응용프로그램의 연결



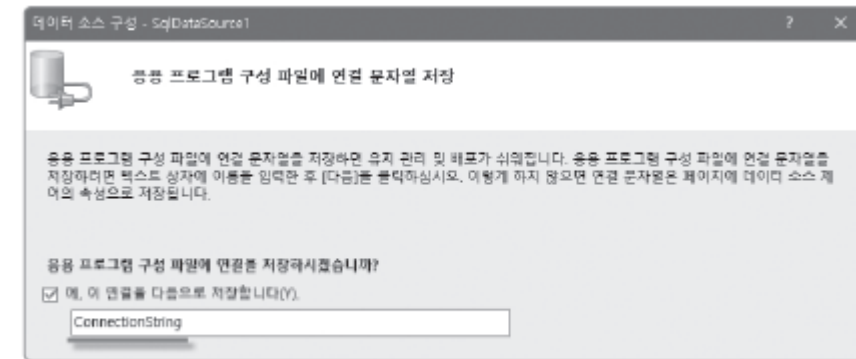
[그림 3-79] Microsoft Visual Studio의 웹 사이트 구축 5



[그림 3-80] Microsoft Visual Studio의 웹 사이트 구축 6



[그림 3-81] Microsoft Visual Studio의 웹 사이트 구축 7



[그림 3-82] Microsoft Visual Studio의 웹 사이트 구축 8



[그림 3-83] Microsoft Visual Studio의 웹 사이트 구축 9

SECTION 05 Oracle과 응용프로그램의 연결



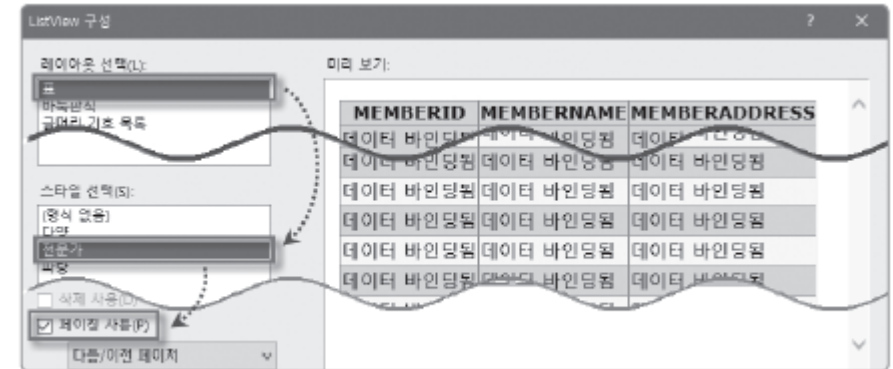
[그림 3-84] Microsoft Visual Studio의 웹 사이트 구축 10



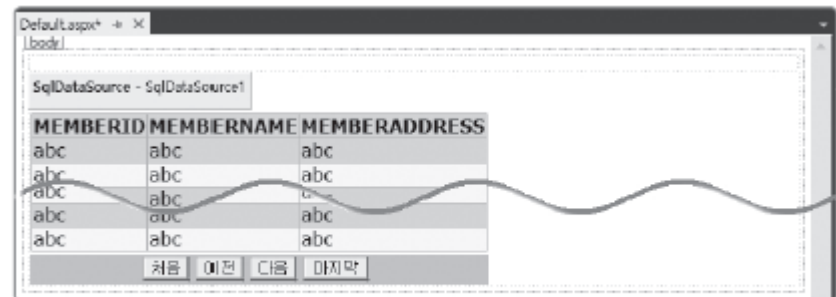
[그림 3-85] Microsoft Visual Studio의 웹 사이트 구축 11



[그림 3-86] Microsoft Visual Studio의 웹 사이트 구축 12



[그림 3-87] Microsoft Visual Studio의 웹 사이트 구축 13



[그림 3-88] Microsoft Visual Studio의 웹 사이트 구축 14

SECTION 05 Oracle과 응용프로그램의 연결

step 5

실제 웹 서비스가 되는 것을 확인해 보자. 메뉴의 [파일] >> [브라우저에서 보기]를 선택하고 잠시 기다리면 웹 브라우저가 실행되고, Oracle의 데이터들이 웹 브라우저에서 보이게 될 것이다.



The screenshot shows a web browser window with the address bar displaying 'localhost:48727/Default.aspx'. The main content area displays a table with three columns: MEMBERID, MEMBERNAME, and MEMBERADDRESS. The table contains three rows of data. Below the table, there are four buttons: '처음' (First), '이전' (Previous), '다음' (Next), and '마지막' (Last).

MEMBERID	MEMBERNAME	MEMBERADDRESS
Jee	지운이	서울 은평구 증산동
Han	한주연	인천 남구 주안동
Sang	상길이	경기 성남시 분당구

처음 이전 다음 마지막

[그림 3-89] 웹 브라우저에서 Oracle 데이터 조회