

본 강의에서 수업자료로 이용되는 저작물은  
저작권법 제25조 수업목적 저작물 이용 보상금제도에 의거,  
한국복제전송저작권협회와 약정을 체결하고 적법하게 이용하고 있습니다.  
약정범위를 초과하는 사용은 저작권법에 저촉될 수 있으므로  
수업자료의 재 복제, 대중 공개·공유 및 수업 목적 외의 사용을 금지합니다.

2024. 8. 30.

부천대학교·한국복제전송저작권협회

**C#**

**ADO.Net 개념**

# C# 프로그래밍

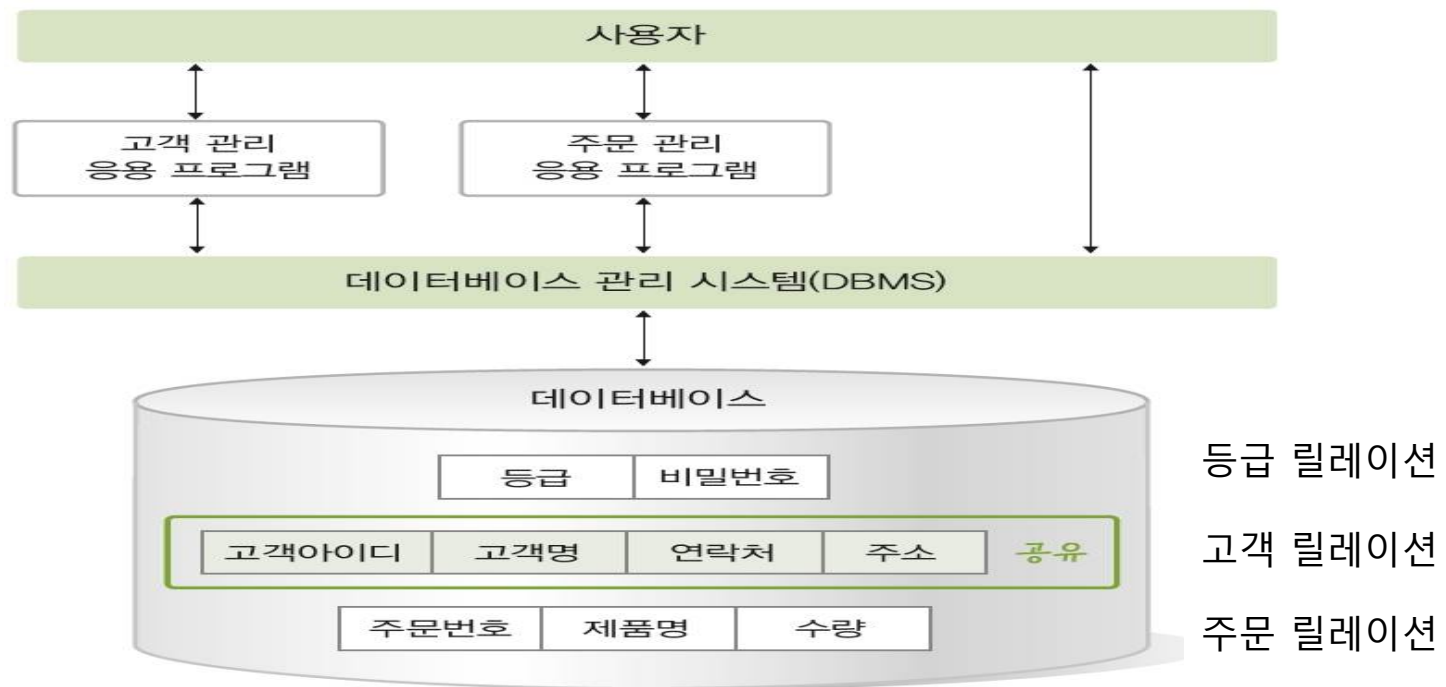


ADO.NET 개념



# 데이터베이스 관리 시스템의 정의

## ■ 데이터베이스 관리 시스템에서의 데이터 관리

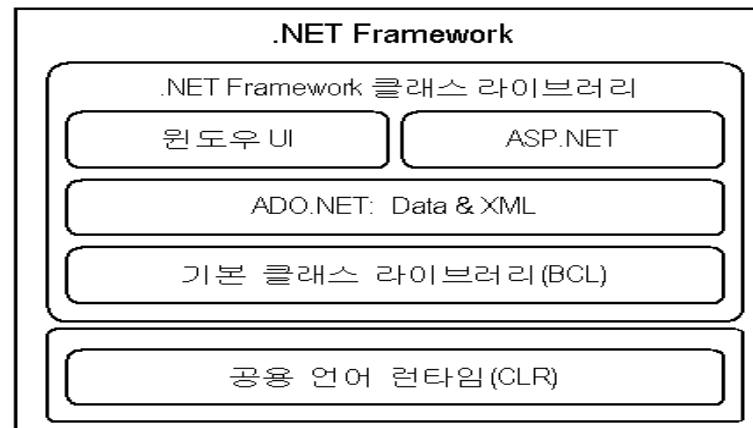


데이터베이스 관리 시스템에서의 데이터 관리



# 닷넷 프레임워크의 구성 요소

## ■ .NET Framework 클래스 라이브러리



- BCL : 클래스 라이브러리 중 가장 핵심적인 기능을 하는 클래스의 모음
- ADO.NET : 데이터베이스를 사용할 때 지원하는 클래스
- 윈도우 UI : 닷넷 언어로 윈도우 프로그램을 만들 때 지원하는 클래스
- ASP.NET : 닷넷 언어로 웹 프로그래밍을 할 때 지원하는 클래스



# ADO .NET의 개요

## ■ ADO .NET

- ADO(Active Data Object) + .NET
- .NET에서 데이터베이스 조작에 관련된 .NET 클래스들의 집합
- 다양한 방법으로 데이터베이스를 검색, 수정, 업데이트 등의 작업 가능
- ADO의 발전 형태
  - Com InterOperation 서비스를 통해서 기존의 ADO를 사용
  - .NET의 추가적인 기능 포함
- ADO .NET이 ADO와 비교해 개선된 사항
  - .NET 기반의 다양한 언어 지원
  - XML 지원
    - Framework 상에서 일관된 포맷으로 지원
  - 단절된 데이터 구조를 표준으로 사용
    - connection이 끊어진 상태에서 작업, 소형 메모리 데이터베이스 모델 지원



# ADO .NET의 개요

## ■ ADO .NET과 관련된 네임스페이스

```
System.Data  
System.Data.Common  
System.Data.OleDb  
System.Data.SqlClient  
  
Oracle.DataAccess.Client;
```

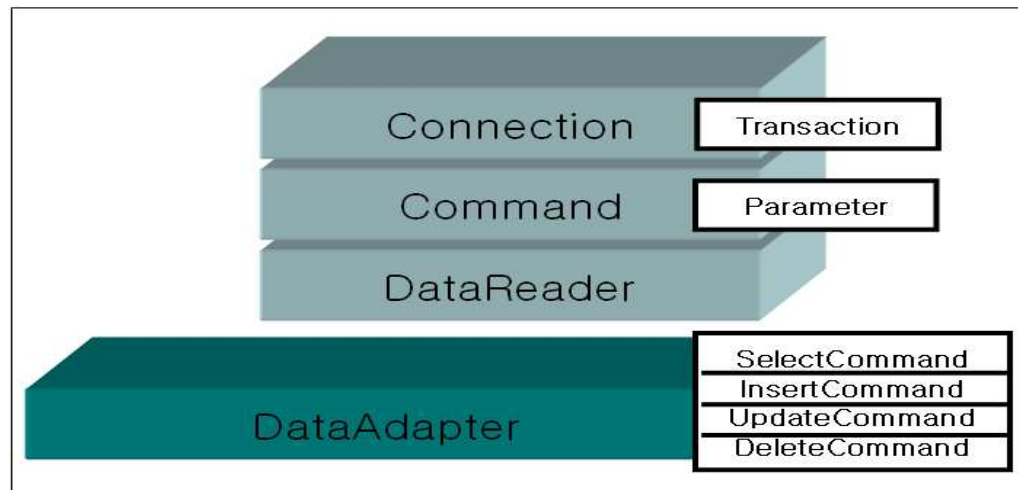
## ■ ADO .NET의 기본 구조

- 구조적으로 .NET Data Provider와 DataSet의 두 부분으로 나뉘어 짐
- .NET Data Provider
  - 데이터베이스에 연결하고, SQL문을 실행시키는 역할
- DataSet
  - 데이터베이스의 데이터를 DataSet으로 쉽게 이식 가능
  - 데이터베이스와 분리되어 DataSet만으로도 대부분의 작업 가능



# ADO .NET의 개요

## ■ .NET Data Provider의 구성 요소



## ■ ADO .NET와 기존 ADO 연결 지향성

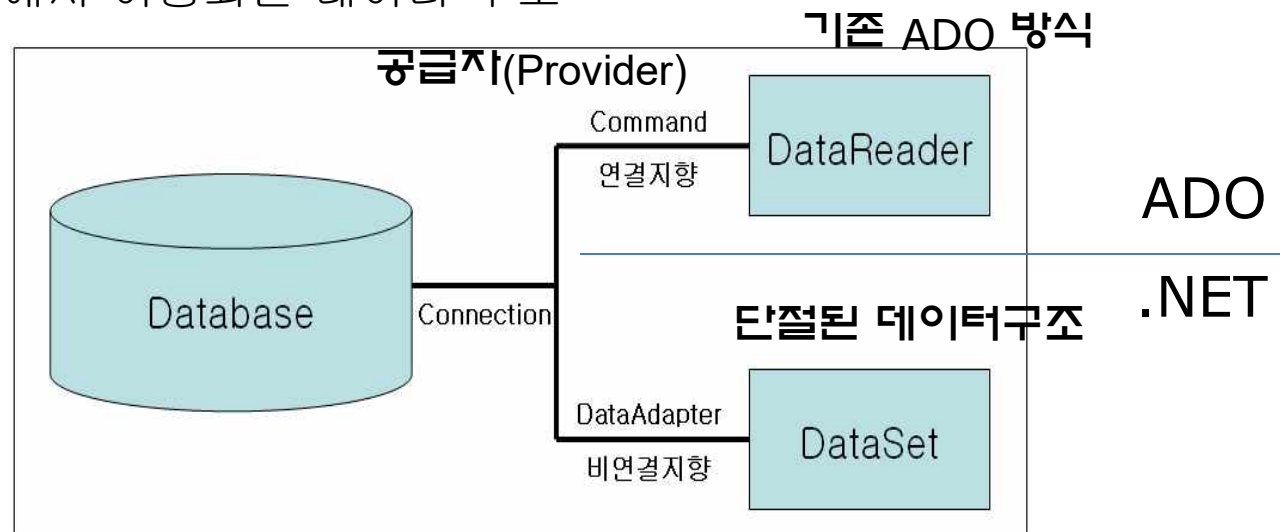
- 데이터베이스에 연결된 후 데이터베이스와 동적으로 작업을 수행
- Command 클래스와 관련이 깊음





# ADO .NET의 개요

- ADO .NET와 기존 ADO 비연결 지향성
  - 메모리 데이터베이스와 관련이 있음
  - 데이터베이스에 연결 설정 후 데이터베이스 형태로 보유하고 있으면서 메모리 데이터베이스만으로 작업을 수행
  - 데이터베이스와 동적으로 작업을 수행
- .NET에서 이용되는 데이터 구조





# ADO .NET의 개요

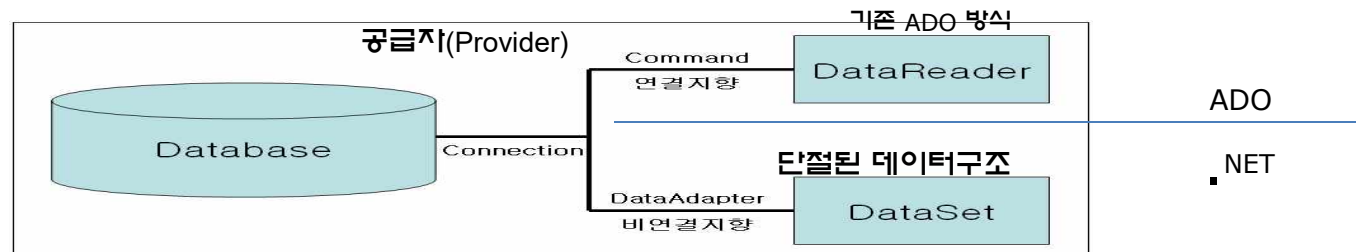
## ■ ADO .NET의 간략한 구조

### ■ DataSet

- 데이터 원본과 단절
- 데이터는 Memory In Cash 형태
- XML과의 데이터 교환

### ■ .NET 데이터 공급자 컴포넌트 (.NET Data Provider)

- Connection : 데이터 원본에 연결
- Command : 검색, 삽입, 수정, 삭제 등의 명령 수행
- DataReader : 순방향 읽기전용으로 연결된 결과 집합
- DataAdapter : 데이터베이스로부터 DataSet에 값을 채우거나 필요로 할 때 데이터베이스로 업데이트 가능

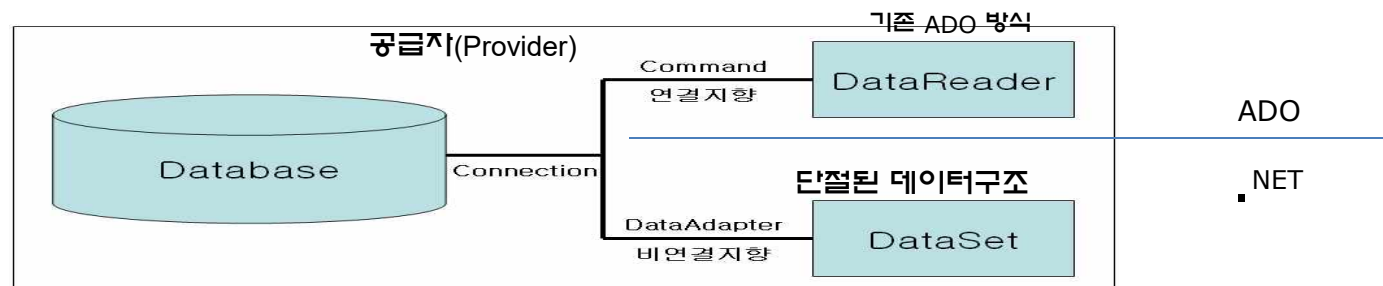
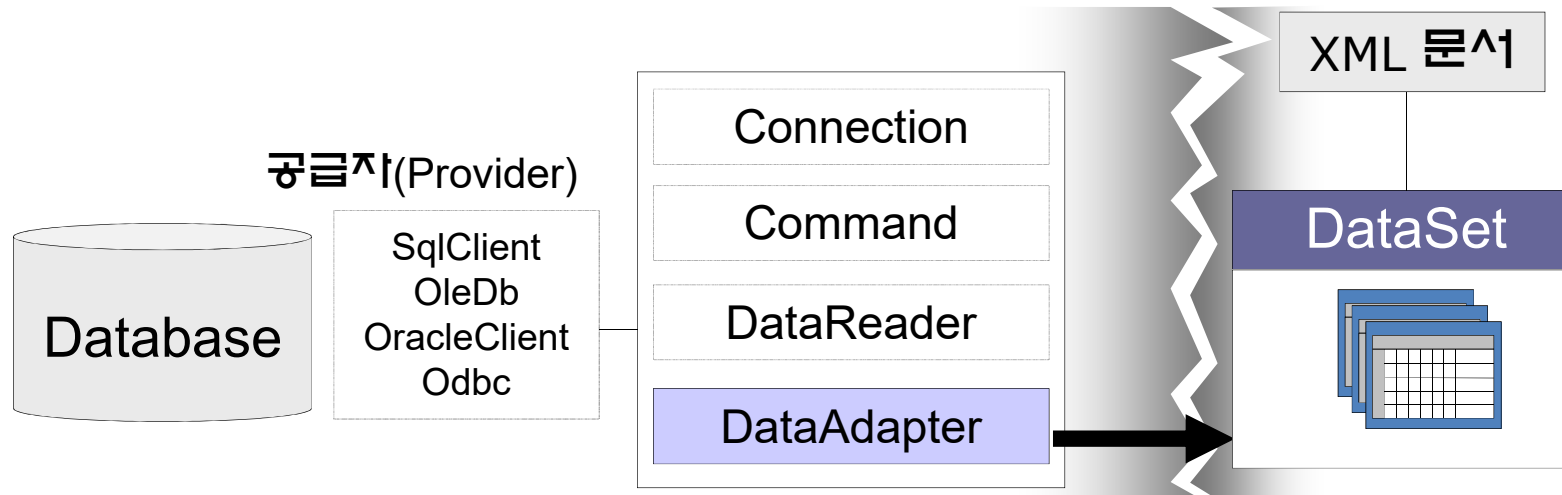




# ADO.NET의 구조

.NET Data Provider

DataSet





# Connection과 데이터 공급자

- SqlClient 공급자
  - SQL Server 공급자(SqlClient)는 OLE와 ODBC와 같은 중간계층 없이 MS-SQL 전용으로 직접 연결되기 때문에 간단하며 효율적으로 동작한다.
  - Microsoft SQL Server 버전 7.0 이상에 연결한다.
- OleDb 공급자
  - OLE DB 공급자는 OLE DB를 사용해서 노출된 데이터 소스 및 Microsoft SQL Server 버전 6.x 또는 그 이전 버전에 SQLOLEDB 즉 SQL Server용 OLE DB 공급자를 통해 연결한다.
- ODBC 공급자(.NET Framework 2.0에 추가된 공급자)
  - ODBC를 사용해서 노출된 데이터 소스에서 사용한다.
  - System.Data.Odbc
- Oracle 공급자(.NET Framework 2.0에 추가된 공급자)
  - 오라클 데이터 소스에서 사용한다.(오라클 클라이언트 소프트웨어 버전 8.1.7 이상에서 지원)
  - System.Data.OracleClient
- Oracle 공급자(.NET Framework 2.0에 추가된 공급자)=> ODP.Net(Oracle Data Provider for .NET)
  - 오라클 데이터 소스에서 사용한다.(오라클 클라이언트 소프트웨어 버전 8.1.7 이상에서 지원)
  - Oracle.DataAccess.Client



## Connection과 데이터 공급자

ADO.NET 데이터 프로바이더	네임스페이스	드라이버/프로바이더	사용빈도*	성능**
ODBC Data Provider	System.Data.Odbc	Microsoft ODBC for Oracle Driver	낮음	좋지 않음
		Oracle ODBC Driver	낮음	보통
		3 <sup>rd</sup> Party ODBC Driver	극히 낮음	?
OLEDB Data Provider	System.Data.OleDb	Microsoft Oracle OLEDB Provider	낮음	좋지 않음
		Oracle OLEDB Provider	보통	보통
		3 <sup>rd</sup> Party OLEDB Provider	?	?
Managed Data Provider for Oracle 1.0	System.Data.OracleClient	N/A (.NET Native)	낮음	보통
Managed Data Provider for Oracle 1.1	System.Data.OracleClient	N/A (.NET Native)	높음	좋음
ODP.NET Data Provider	Oracle.DataAccess.Client	N/A (.NET Native)	높음	좋음



# Connection과 데이터 공급자

## SQL-Server => SqlClient 공급자의 연결(Connection)

```
using System.Data.SqlClient;  
SqlConnection conn = new SqlConnection("Data Source=(local);Initial Catalog=HongDB; User ID=sa;Password=");  
conn.open();
```

## Oracle => ODBC 공급자의 연결(Connection)

```
using System.Data.Odbc;  
OdbcConnection conn= new OdbcConnection("Provider=MSDAORA; DSN=orads; Server = localhost; database=scott;  
UID=scott; PWD=tiger;");  
conn.Open();
```

## Access=>OleDb 공급자의 연결(Connection)

```
using System.Data.OleDb;  
OleDbConnection con = new OleDbConnection( @"Provider=Microsoft.Jet.OLEDB.4.0; Data Source=C:/NWind.mdb");  
conn.open();
```

## Oracle => **ODP.Net** 공급자의 연결(Connection)

```
using Oracle.DataAccess.Client;  
OracleConnection conn = new OracleConnection("User Id=scott; Password=scott; Data Source=(DESCRIPTION = (ADDRESS  
= (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED)  
(SERVICE_NAME = xe) ) );");  
conn.open();
```

# C# 프로그래밍



C#과 Oracle 11g 연동  
(ADO.NET 이용)



# 데이터베이스 다루기

## ■ 데이터베이스 공급자의 Connection 설정 - 1

### 1. 네임스페이스 명시

- Sql을 사용할 때

```
using System.Data.SqlClient
```

- Oracle를 사용할 때

```
using Oracle.DataAccess.Client;
```

```
using System.Data.Odbc;
```





# 데이터베이스 다루기

- 데이터베이스 공급자의 Connection 설정 – 2
  - 2. Connection 객체 생성 : 생성자에 연결 프로퍼티 설정
    - Sql을 사용할 때

```
SqlConnection conn;  
conn = new SqlConnection("Server=localhost;user id=sa;password; database=HongDB");
```

- Oracle를 사용할 때

```
OracleConnection conn;  
conn = new OracleConnection("("User Id=scott; Password=tiger; Data Source=(DESCRIPTION =  
(ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521)) (CONNECT_DATA =  
(SERVER = DEDICATED) (SERVICE_NAME = orcl.localdomain) ) );");
```

```
OdbcConnection conn;  
conn = new OdbcConnection("Provider=MSDAORA; DSN=orads; Server = localhost; database=scott;  
UID=scott; PWD=tiger;");
```



## 데이터베이스 다루기

### ■ 데이터베이스 공급자의 Connection 설정 - 3

#### 3. Connecton 연결

```
conn.Open();
```

#### 4. 작업수행( SQL 쿼리 작업)

#### 5. 연결 닫기

```
conn.Close();
```

# Oracle 공급자





# Oracle 공급자의 Connection 설정하기

## 예제 1 [w7\_dbex1\_xxxx]

### ■ SQL DB 연결 순서

```
■ using System;
■ using System.Data;
■ using Oracle.DataAccess.Client;

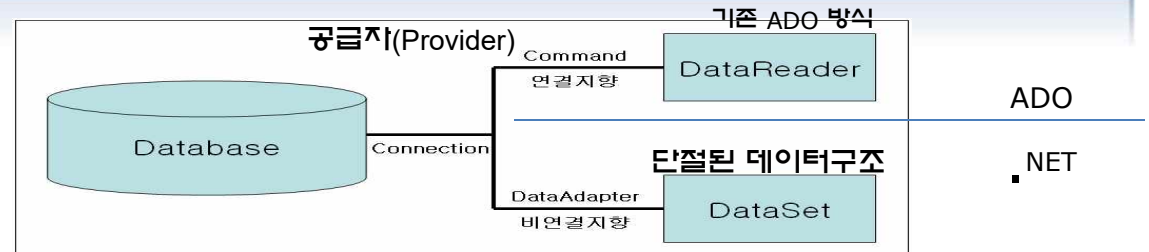
■ class ConnectionTest
■ {
■     public static void Main(string[] args)
■     {
■         String conninfo = "User Id=scott; Password=scott; Data Source=(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = xe) ) );";
■         OracleConnection conn = new OracleConnection(conninfo);
■         conn.Open();

■         Console.WriteLine("1. DataSource = " + conn.DataSource);
■         Console.WriteLine("2. DataServerVersion = " + conn.ServerVersion);
■         Console.WriteLine("3. State = " + conn.State);
■         conn.Close();// Close Connection
■         Console.WriteLine("4. ConnectionString = " + conn.ConnectionString);

■     } //main
■ } //class
```

### ■ Connection의 속성

```
■ conn.DataSource
■ conn.ServerVersion
■ conn.State
■ conn.ConnectionString
```



1. Connection 생성 및 열기

2. Connection 정보 출력

3. Connection 닫기

```
C:\WINDOWS\system32\cmd.exe
EDICATED>    <SERVICE_NAME = orcl.localdomain>    > >
2. DataServerVersion = 11.2.0.1.0
3. State = Open
4. ConnectionString = User Id=scott; Data Source=(DESCRIPTION =
  <ADDRESS = <PROTOCOL = TCP><HOST = localhost><PORT = 1521>>
  <CONNECT_DATA =    <SERVER = DEDICATED>    <SERVICE_NAME =
  orcl.localdomain>    > >);
계속하려면 아무 키나 누르십시오 . . .
```



## 데이터베이스 다루기

### ■ OracleCommand 클래스

- Connection 객체를 이용해 데이터베이스와 연결한 후, DB에 필요한 명령을 전달하거나 결과를 받아올 때 사용
- OracleCommand 클래스를 이용하여 수행할 수 있는 명령들
  - 단일 값 및 레코드 셋을 반환하는 SELECT, CREATE, ALERT, DROP 같은 DDL(Data Definition Language) 명령
  - GRANT, DENY, REVOKE 같은 DCL(Data Control Language) 명령
  - INSERT, UPDATE, DELETE 같은 DML(Data Modification Language) 명령
- 명령(Command)에 관련된 여러 가지 정보를 관리 가능
- 명령 수행을 위한 메서드들 포함



## 데이터베이스 다루기

### ■ OracleCommand의 주요 속성들

#### ■ CommandText

- 데이터 소스에서 실행할 SQL 문이나 저장 프로시저를 가져오거나 설정

#### ■ CommandType

- CommandText 속성이 해석될 방법을 나타내는 값을 가져오거나 설정

#### ■ Connection

- OracleCommand의 인스턴스에서 사용하는 OracleConnection을 가져오거나 설정

#### ■ Parameters

- OracleParameterCollection을 가져옴

#### ■ Transaction

- OracleCommand가 실행하는 트랜잭션을 가져오거나 설정



## 데이터베이스 다루기

### ■ OracleCommand의 주요 메서드

#### ■ ExecuteNonQuery()

- Connection에 대한 SQL 문을 실행하고 영향을 받는 행의 개수를 반환

#### ■ ExecuteReader()

- CommandText를 Connection에 보내고, OracleDataReader를 생성

#### ■ ExecuteScalar()

- 쿼리를 실행하고 쿼리에서 반환된 결과 집합의 첫 번째 행의 첫 번째 열 반환

#### ■ ExecuteXmlReader()

- CommandText를 Connection에 보내고, XmlReader 객체를 생성



## 데이터베이스 다루기

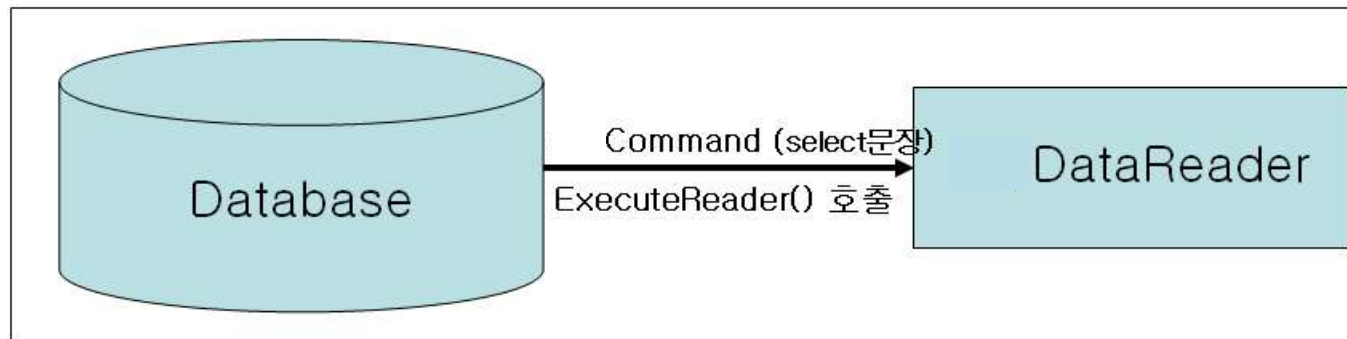
### ■ OracleCommand의 ExecuteNonQuery

#### ■ ExecuteNonQuery() 메서드

- OracleCommand를 이용하여 데이터베이스에 명령을 전달하고 이를 수행
- Select 명령을 제외한 대부분의 명령(DML)을 ExecuteNonQuery()를 이용하여 처리 가능

### ■ OracleCommand의 ExecuteReader

#### ■ Select 문장을 사용할 때 이용되는 메서드







# OracleCommand의 ExecuteReader 예제 2 [w7\_dbex2\_xxxx]

```
■ using System;
■ using System.Data;
■ using ???.DataAccess.Client;
■ public class CommandReader
■ {
■     public static void Main()
■     {
■         string ConStr = "User Id=scott; Password=scott; Data Source=(DESCRIPTION =
■         (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521)) (CONNECT_DATA =
■         (SERVER = DEDICATED) (SERVICE_NAME = xe) ) );";
■         string query = "select ename from emp";
■         ???Connection conn = new ???Connection(ConStr);
■         conn.Open();
■         ???Command comm = new ???Command(query, conn);
■         ???DataReader sr = comm.ExecuteReader();

■         while (sr.Read())
■         {
■             Console.WriteLine(sr.GetString(0) + "|");
■         }
■         sr.Close();
■         conn.Close();
■     }
■ }
```

