

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#

10주차

원폼 C#과 Oracle 11g 연동(3)



ADOForm ver1에서 ver3까지 비교

■ ADOForm ver1에서 ver3까지 비교

ADOForm ver1 : 지역 변수
: 기능 => 전체보기, 추가, 수정, 삭제

ADOForm ver2 : 모듈 변수, 사용자 함수 정의
: 기능 => 전체보기, 추가, 수정, 삭제, 검색

ADOForm ver3 : 다중 폼 정보 전달, 모듈 변수, 사용자 함수 정의
: 기능 => 전체보기, 추가, 수정, 삭제, 검색

C#과 Oracle11g 연동 프로그램



ADOFrm(ver1)

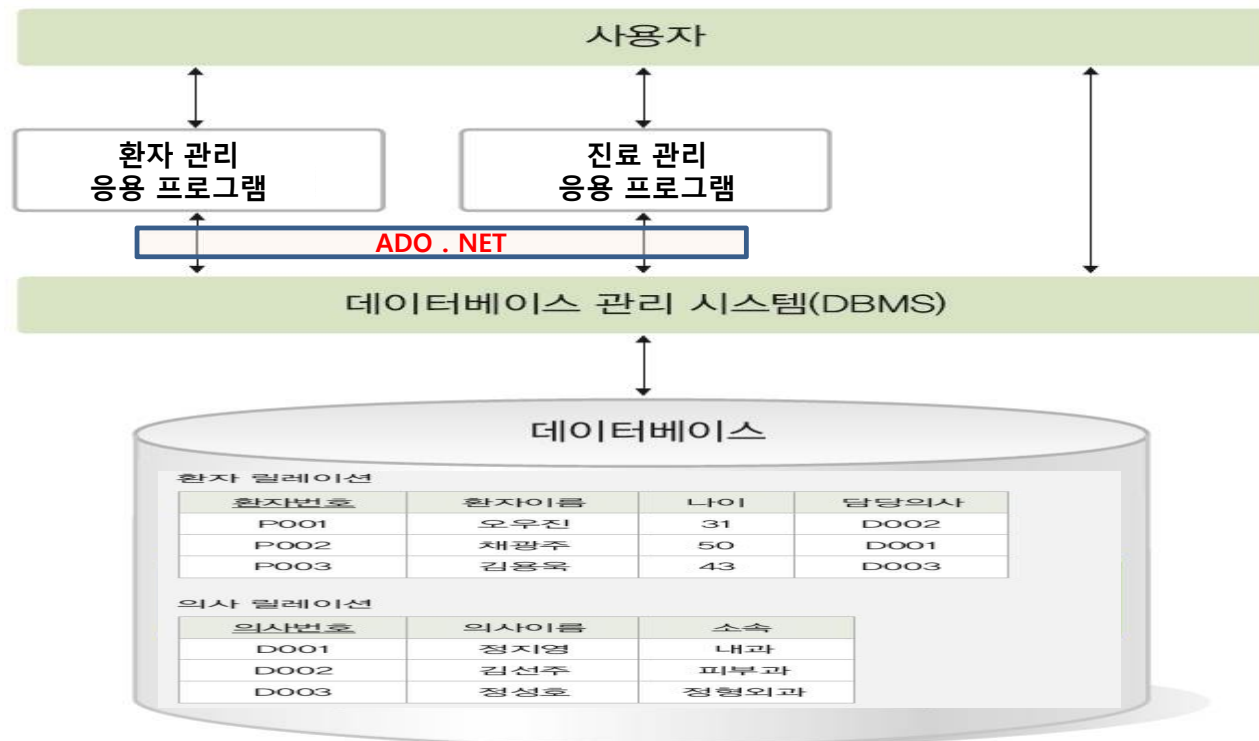
지역 변수

기능 => 전체보기, 추가, 수정, 삭제



데이터베이스 관리 시스템의 정의

■ 데이터베이스 관리 시스템에서의 데이터 관리

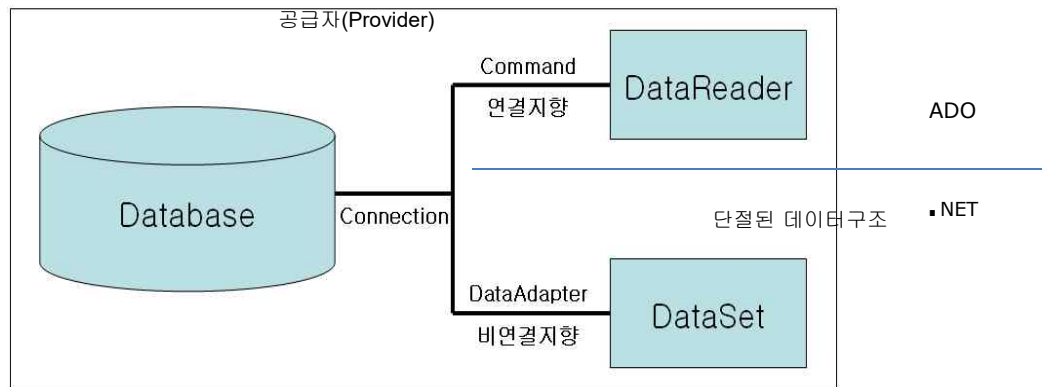
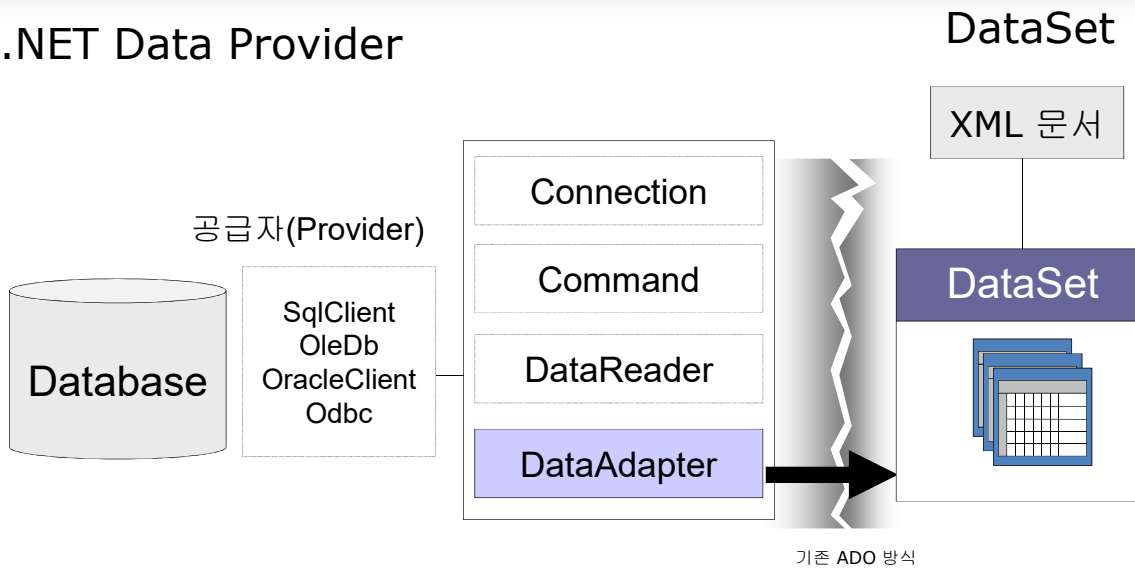


데이터베이스 관리 시스템에서의 데이터 관리

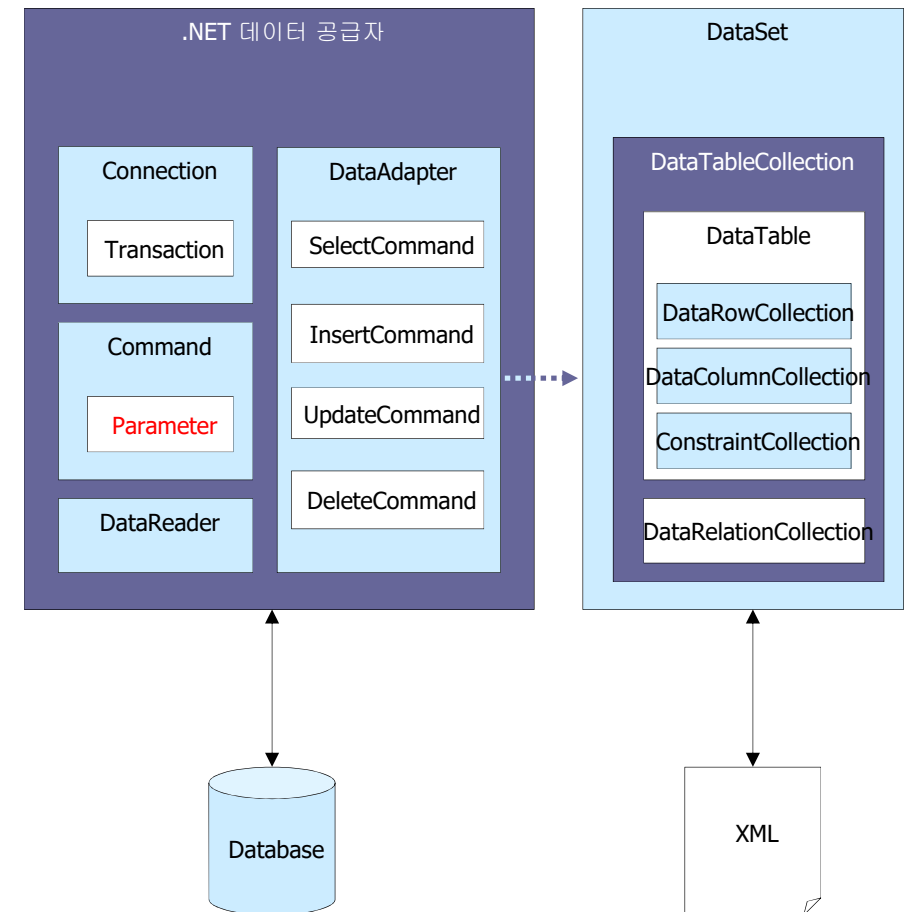


ADO.NET의 구조

.NET Data Provider



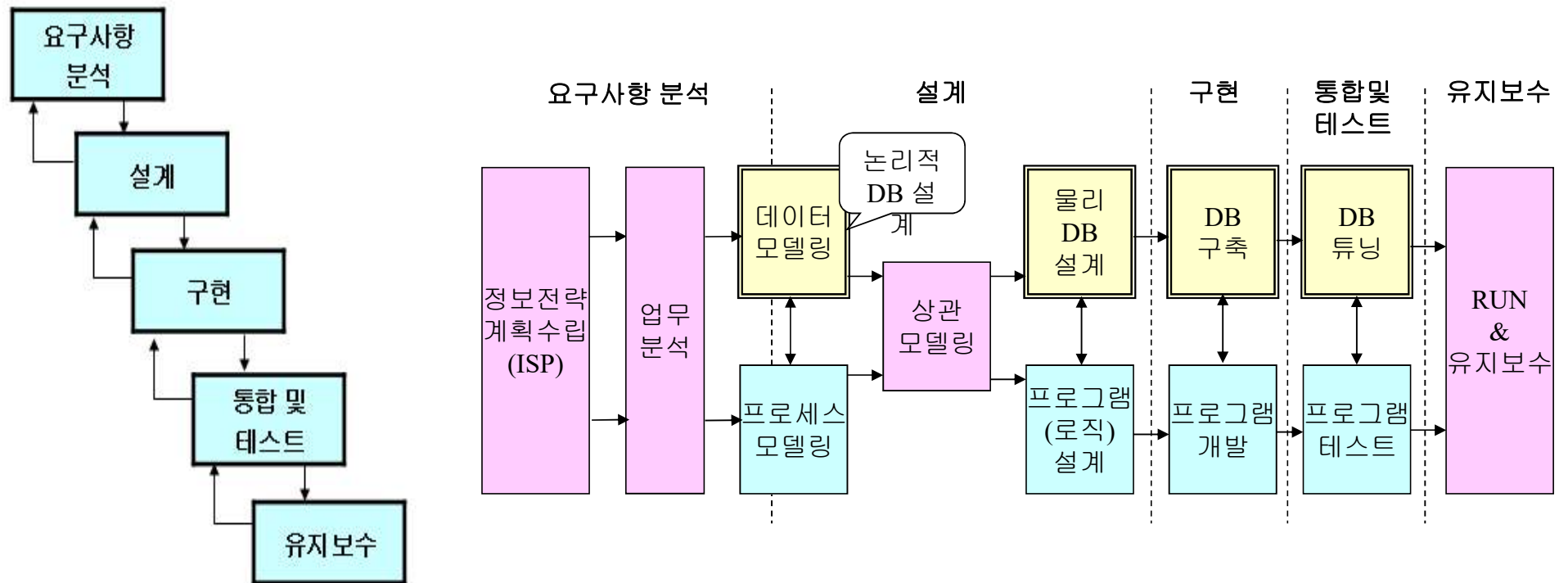
DataSet





소프트웨어 개발 과정

■ 일반적인 소프트웨어 개발 절차



소프트웨어 개발 절차 (waterfall 모델)



ADOForm(ver1) 프로그램 기획 구상

■ 실행 창

id 2

이름 박길동

전화번호 010-222-2222

E-Mail 222@bc.ac.kr

추가 수정 삭제

	ID	PNAME	PHONE	EMAIL
	1	홍길동	010-111-1111	111@bc.ac.kr
▶	2	박길동	010-222-2222	222@bc.ac.kr
	3	이길동	010-333-3333	333@bc.ac.kr
	4	김길동	010-444-4444	444@bc.ac.kr
*				

DAOpenBtn CCBtn DRBtn



ADOForm(ver1) 데이터베이스 구축 과정

■ 데이터베이스 구축 과정

1. DB 설계
2. DBMS 결정
3. 사용자 계정 설정과 권한 부여
4. 테이블 생성
5. 레코드 추가
6. 질의어로 업무에 필요한 정보 활용



ADOForm(ver1) DB 설계

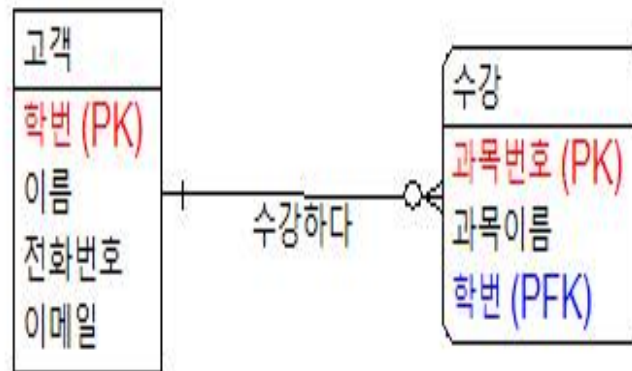
■ DB 설계 과정

논리적 설계 (데이터 모델링)	물리적 설계	데이터베이스
엔티티 (entity)	테이블 (table)	테이블
속성 (attribute)	컬럼 (column)	컬럼
주식별자 (primary identifier)	기본키 (primary key)	기본키
외래식별자 (foreign identifier)	외래키 (foreign key)	외래키
관계 (relationship)	관계 (relationship)	
관계의 카디널리티	관계의 카디널리티	-
관계의 참여도	관계의 참여도	-



ADOForm(ver1) DB 설계

■ 논리적 DB 설계 => 논리적 ERD



Entity

Entity Name: Table Name:

Key	Name	Column Name	Datatype	Not nu	Unique	Description
1	학번	id	Number	Not nu		
2	이름	pname	Varchar2(20)			
3	전화번호	phone	Varchar2(20)			
4	이메일	email	Varchar2(20)			

☒ Generate

Entity

Entity Name: Table Name:

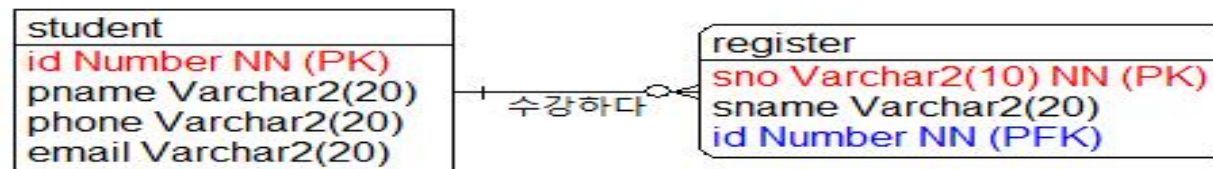
Key	Name	Column Name	Datatype	Not nu	Unique	Description
1	과목번호	sno	Varchar2(10)	Not nu		
2	과목이름	sname	Varchar2(20)			
3	학번	id	Number	Not nu		

☒ Generate



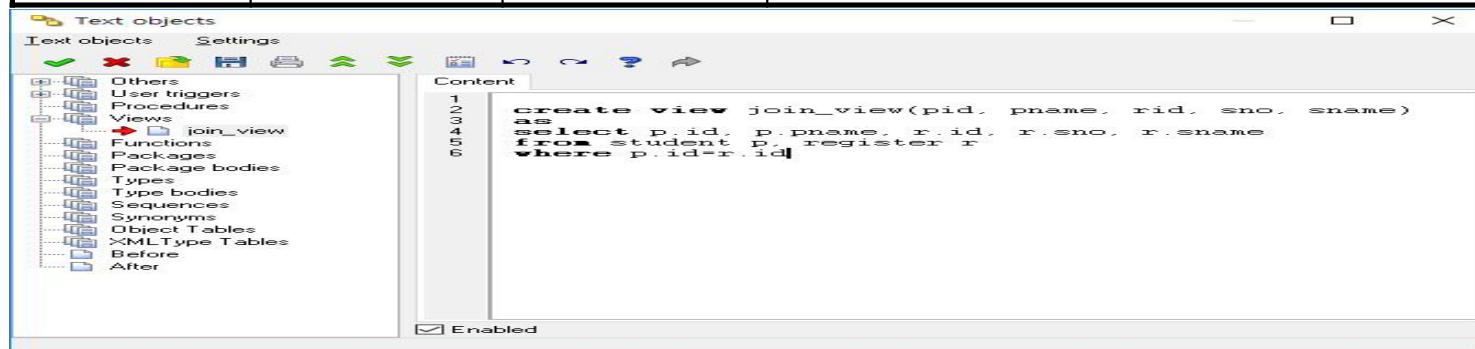
ADOForm(ver1) DB 설계

- 물리적 DB 설계 => 물리적 ERD



- 물리적 DB 설계 => 뷰 설계

뷰명	뷰 설명	관련테이블	SQL
Join_view	학생과 과목 수강 정보	student와 register	select p.id, p.pname, r.id, r.sno, r.sname from student p, register r where p.id=r.id





ADOForm(ver1) 구현 : DB 구축

■ Oracle 11g를 이용한 사용자 계정과 테이블 생성

```
c:\W>sqlplus
```

```
sql> conn system/system
```

```
sql> create user hong1 identified by 1111;
```

```
sql> grant create session to hong1;
```

```
sql> grant resource, create view, create table to hong1;
```

```
sql> conn hong1/1111
```

```
sql> @ c:\Whong\Whong1.sql
```

```
sql> commit;
```

■ 사용자 이름: hong1

■ 테이블 생성 : phone

필드 이름	데이터 형식
id	number
pname	varchar(20)
phone	varchar(20)
email	varchar(20)

■ hong1.sql작성

■ 메모장을 이용하여 저장

■ c:\Whong\Whong1.sql

```
drop table phone purge
/
create table phone(id number primary key, pname varchar(20), phone varchar(20), email varchar(20))
/
insert into phone values(1, '홍길동','010-111-1111', '111@bc.ac.kr')
/
insert into phone values(2, '박길동','010-222-2222', '222@bc.ac.kr')
/
insert into phone values(3, '이길동','010-333-3333', '333@bc.ac.kr')
/
insert into phone values(4, '김길동','010-444-4444', '444@bc.ac.kr')
/
```



ADOForm(ver1) 구현 : 폼 디자인

■ Form1 디자인

The screenshot shows a Windows Forms application window titled "Form1". The form contains the following elements:

- Input fields for "id", "이름", "전화번호", and "E-Mail".
- Text boxes labeled `txtid`, `txtName`, `txtPhone`, and `txtMail` are positioned over the input fields.
- Buttons labeled "추가", "수정", and "삭제" are positioned to the right of the input fields.
- Buttons labeled `AppendBtn`, `UpdateBtn`, and `DeleteBtn` are positioned below the "추가", "수정", and "삭제" buttons respectively.
- A large `DataGridView : DBGrid` is positioned in the center of the form.
- Buttons labeled `DAOOpenBtn`, `CCBtn`, and `DRBtn` are positioned at the bottom of the form.



ADOForm(ver1) 구현 : 이벤트 처리기

■ DAOpenBtn_Click

private int SelectedRowIndex; //Form1생성자 위에 선언

```
private void DAOpenBtn_Click(object sender, EventArgs e)
{
    try
    {
        string connectionString = "User Id=hong1; Password=1111; Data Source=(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = xe) ) );";
        string commandString = "select * from phone";
        OracleDataAdapter DBAdapter = new OracleDataAdapter(commandString, connectionString);

        DataSet DS = new DataSet();
        DBAdapter.Fill(DS, "phone");
        DBGrid.DataSource = DS.Tables["phone"].DefaultView;
    }
    catch (DataException DE)
    {
        MessageBox.Show(DE.Message);
    }
    catch (Exception DE)
    {
        MessageBox.Show(DE.Message);
    }
}
```



ADOForm(ver1) 구현 : 이벤트 처리기

■ CCBtn_Click

```
■ private void CCBtn_Click(object sender, EventArgs e)
{ try
{
```

```
string connectionString = "User Id=hong1; Password=1111; Data Source=(DESCRIPTION = (ADDRESS = (PROTOCOL
= TCP)(HOST = localhost)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = xe) ) );";
```

```
OracleConnection myConnection = new OracleConnection(connectionString);
```

```
string commandString = "select * from Phone";
OracleCommand myCommand = new OracleCommand();
myCommand.Connection = myConnection;
myCommand.CommandText = commandString;
```

```
OracleDataAdapter DBAdapter = new OracleDataAdapter();
DBAdapter.SelectCommand = myCommand;
```

```
DataSet DS = new DataSet();
```

```
DBAdapter.Fill(DS, "Phone");
DBGrid.DataSource = DS.Tables["Phone"].DefaultView;
```

```
}
catch (DataException DE)
{
    MessageBox.Show(DE.Message);
}
catch (Exception DE)
{
    MessageBox.Show(DE.Message);
}
}
```

```
,
```




ADOForm(ver1) 구현 : 이벤트 처리기

■ DRBtn_Click

```
private void DRBtn_Click(object sender, EventArgs e)
{
    try
    {
        string connectionString = "User Id=hong1; Password=1111; Data Source=(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = xe) ) );";
        OracleConnection myConnection = new OracleConnection(connectionString);
        string commandString = "select * from phone";
        OracleCommand myCommand = new OracleCommand();
        myCommand.Connection = myConnection;
        myCommand.CommandText = commandString;
        myConnection.Open();
        OracleDataReader myReader;
        myReader = myCommand.ExecuteReader();
        string ResultMessage = "";
        while (myReader.Read())
        {
            ResultMessage = myReader.GetString(1) + ", " + myReader.GetString(2);
            MessageBox.Show(ResultMessage);
        }
        myReader.Close();
        myConnection.Close();
    }
    catch (DataException DE)
    {
        MessageBox.Show(DE.Message);
    }
    catch (Exception DE)
    {
        MessageBox.Show(DE.Message);
    }
}
```



ADOForm(ver1) 구현 : 이벤트 처리기

■ AppendBtn_Click

```
private void AppendBtn_Click(object sender, EventArgs e)
{
    try
    {
        string connectionString = "User Id=hong1; Password=1111; Data Source=(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = xe) ) );";
        string commandString = "select * from Phone";
        OracleDataAdapter DBAdapter= new OracleDataAdapter(commandString, connectionString);
        OracleCommandBuilder myCommandBuilder = new OracleCommandBuilder(DBAdapter);
        DataSet DS = new DataSet();
        DBAdapter.Fill(DS, "Phone");
        DataTable phoneTable = DS.Tables["Phone"];
        DataRow newRow = phoneTable.NewRow();
        newRow["id"] = Convert.ToInt32(txtid.Text);
        newRow["PName"] = txtName.Text;
        newRow["Phone"] = txtPhone.Text;
        newRow["Email"] = txtMail.Text;

        phoneTable.Rows.Add(newRow);
        DBAdapter.Update(DS, "Phone");
        DBGrid.DataSource = DS.Tables["Phone"].DefaultView;
    }
    catch (DataException DE)
    {
        MessageBox.Show(DE.Message);
    }
    catch (Exception DE)
    {
        MessageBox.Show(DE.Message);
    }
}
```



ADOForm(ver1) 구현 : 이벤트 처리기

■ UpdateBtn_Click

```
■ private void UpdateBtn_Click(object sender, EventArgs e)
{ try {
string connectionString = "User Id=hong1; Password=1111; Data Source=(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST =
localhost)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = xe) ) );";
string commandString = "select * from Phone";
OracleDataAdapter DBAdapter = new OracleDataAdapter(commandString, connectionString);
OracleCommandBuilder myCommandBuilder = new OracleCommandBuilder(DBAdapter);
DataSet DS = new DataSet("Phone");
DBAdapter.Fill(DS, "Phone");
DataTable phoneTable = DS.Tables["Phone"];
DataColumn[] PrimaryKey = new DataColumn[1];
PrimaryKey[0] = phoneTable.Columns["id"];
phoneTable.PrimaryKey = PrimaryKey;
DataRow currRow = phoneTable.Rows.Find(SelectedRowIndex);
currRow.BeginEdit();
currRow["id"] = txtid.Text;
currRow["PName"] = txtName.Text;
currRow["Phone"] = txtPhone.Text;
currRow["EMail"] = txtMail.Text;
currRow.EndEdit();
DataSet UpdatedSet = DS.GetChanges(DataRowState.Modified);
if (UpdatedSet.HasErrors)
{
    MessageBox.Show("변경된 데이터에 문제가 있습니다.");
}
else
{
    DBAdapter.Update(UpdatedSet, "Phone");
    DS.AcceptChanges();
    DBGrid.DataSource = DS.Tables["Phone"].DefaultView;
}
}
catch (DataException DE)
{
    MessageBox.Show(DE.Message);
}
catch (Exception DE)
{
    MessageBox.Show(DE.Message);
}
```



ADOForm(ver1) 구현 : 이벤트 처리기

■ DeleteBtn_Click

```
private void DeleteBtn_Click(object sender, EventArgs e)
{
    try
    {
        string connectionString = "User Id=hong1; Password=1111; Data Source=(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = xe) ) );";
        string commandString = "select * from Phone";
        OracleDataAdapter DBAdapter = new OracleDataAdapter(commandString, connectionString);
        OracleCommandBuilder myCommandBuilder = new OracleCommandBuilder(DBAdapter);
        DataSet DS = new DataSet("Phone");
        DBAdapter.Fill(DS, "Phone");
        DataTable phoneTable = DS.Tables["Phone"];
        DataColumn[] PrimaryKey = new DataColumn[1];
        PrimaryKey[0] = phoneTable.Columns["id"];
        phoneTable.PrimaryKey = PrimaryKey;
        DataRow currRow = phoneTable.Rows.Find(SelectedRowIndex);
        currRow.Delete();
        DBAdapter.Update(DS.GetChanges(DataRowState.Deleted), "Phone");
        DBGrid.DataSource = DS.Tables["Phone"].DefaultView;
    }
    catch (DataException DE)
    {
        {
            MessageBox.Show(DE.Message);
        }
    }
    catch (Exception DE)
    {
        {
            MessageBox.Show(DE.Message);
        }
    }
}
```



ADOForm(ver1) 구현 : 이벤트 처리기

■ DBGrid_CellClick

```
private void DBGrid_CellClick(object sender, DataGridViewCellEventArgs e)
{ try
{
string connectionString = "User Id=hong1; Password=1111; Data Source=(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST =
localhost)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = xe) ) );";
string commandString = "select * from Phone";
OracleDataAdapter DBAdapter = new OracleDataAdapter(commandString, connectionString);
DataSet DS = new DataSet();
DBAdapter.Fill(DS, "Phone");
DataTable phoneTable = DS.Tables["Phone"];
if (e.RowIndex < 0)
{
return;
}
else if (e.RowIndex > phoneTable.Rows.Count - 1)
{
    MessageBox.Show("해당하는 데이터가 존재하지 않습니다.");
    return;
}
DataRow currRow = phoneTable.Rows[e.RowIndex];
txtid.Text = currRow["id"].ToString();
txtName.Text = currRow["PName"].ToString();
txtPhone.Text = currRow["Phone"].ToString();
txtMail.Text = currRow["EMail"].ToString();
SelectedRowIndex = Convert.ToInt32(currRow["id"]);
}
catch (DataException DE)
{
    MessageBox.Show(DE.Message);
}
catch (Exception DE)
{
    MessageBox.Show(DE.Message);
}
}
```

DataGridViewCellEventArgs.RowIndex 속성

: 이벤트가 발생하는 셀의 행 인덱스를 나타내는 값을 가져옴 0부터 시작



ADOForm(ver1) 프로그램 실행

■ 실행 창

id 2

이름 박길동

전화번호 010-222-2222

E-Mail 222@bc.ac.kr

추가 수정 삭제

	ID	PNAME	PHONE	EMAIL
	1	홍길동	010-111-1111	111@bc.ac.kr
▶	2	박길동	010-222-2222	222@bc.ac.kr
	3	이길동	010-333-3333	333@bc.ac.kr
	4	김길동	010-444-4444	444@bc.ac.kr
*				

DAOpenBtn

CCBtn DRBtn



C#과 Oracle11g 연동 프로그램

ADODForm(ver2)

모듈 변수, 사용자 함수 정의

기능 => 전체보기, 추가, 수정, 삭제, 검색



DataSet으로 작업하기 원폼 프로그램 만들기 ver2

■ 실행 창

Form1

id: 3

이름: 이길동

전화번호: 010-333-3333

E-Mail: 333@bc.ac.kr

추가 수정 삭제

ID	PNAME	PHONE	EMAIL
1	홍길동	010-111-1111	111@bc.ac.kr
2	박길동	010-222-2222	222@bc.ac.kr
3	이길동	010-333-3333	333@bc.ac.kr
4	김길동	010-444-4444	444@bc.ac.kr
*			

이름으로 찾기

txtSearch

찾기

SearchBtn

NameList :ListBox

위로

아래로

UpBtn

DownBtn

DAOOpenBtn



DataSet으로 작업하기 원폼 프로그램 만들기 ver2

■ 객체 변수 선언

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

```
public partial class Form1 : Form
```

```
{
```

```
    private int SelectedRowIndex;    //수정하거나 삭제하기 위해 선택된 행의 인덱스를 저장한다.
```

```
    OracleDataAdapter DBAdapter;    // Data Provider인 DBAdapter 입니다.
```

```
    DataSet DS; // DataSet 객체입니다.
```

```
    OracleCommandBuilder myCommandBuilder;
```

```
        // 추가, 수정, 삭제시에 필요한 명령문을 자동으로 작성해주는 객체입니다.
```

```
    DataTable phoneTable;           // DataTable 객체입니다.
```

```
    private int SelectedKeyValue; // 수정, 삭제할 때 필요한 레코드의 키값을 보관할 필드
```

```
    private void ClearTextBoxes()
```

```
{
```

```
        txtid.Clear();
```

```
        txtName.Clear();
```

```
        txtPhone.Clear();
```

```
        txtMail.Clear();
```

```
}
```



DataSet으로 작업하기 원폼 프로그램 만들기 ver2

■ DB연결

```
public partial class Form1 : Form
{
    private void DB_Open()
    {
        try
        {
            string connectionString = "User Id=hong1; Password=1111; Data Source=(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = xe) ) );";
            string commandString = "select * from Phone";
            DBAdapter = new OracleDataAdapter(commandString, connectionString);
            myCommandBuilder = new OracleCommandBuilder(DBAdapter);

            DS = new DataSet();
        }
        catch (DataException DE)
        {
            MessageBox.Show(DE.Message);
        }
    }

    public Form1()
    {
        InitializeComponent();
        DB_Open();
    }
}
```



DataSet으로 작업하기 원폼 프로그램 만들기 ver2

■ DB연결

```
public partial class Form1 : Form
{

    private void DAOpenBtn_Click(object sender, EventArgs e)
    {
        try
        {
            DS.Clear();/**/
            DBAdapter.Fill(DS, "phone");
            DBGrid.DataSource = DS.Tables["phone"].DefaultView;
        }
        catch (DataException DE)
        {
            MessageBox.Show(DE.Message);
        }
        catch (Exception DE)
        {
            MessageBox.Show(DE.Message);
        }
    }

}
```



DataSet으로 작업하기 원폼 프로그램 만들기 ver2

■ AppendBtn_Click

```
private void AppendBtn_Click(object sender, EventArgs e)
{
    try
    {
        MessageBox.Show("텍스트 상자에 모든 데이터 입력하셨으면 추가합니다!");
        DS.Clear();
        DBAdapter.Fill(DS, "Phone");

        phoneTable = DS.Tables["Phone"]; /**
        DataRow newRow = phoneTable.NewRow();

        newRow["id"] = Convert.ToInt32(txtid.Text);
        newRow["PName"] = txtName.Text;
        newRow["Phone"] = txtPhone.Text;
        newRow["Email"] = txtMail.Text;

        phoneTable.Rows.Add(newRow);

        DBAdapter.Update(DS, "Phone");
        DS.AcceptChanges();
        ClearTextBoxes();
        DBGrid.DataSource = DS.Tables["Phone"].DefaultView;

    }
    catch (DataException DE)
    {
        MessageBox.Show(DE.Message);
    }

    catch (Exception DE)
    {
        MessageBox.Show(DE.Message);
    }
}
```



DataSet으로 작업하기 원폼 프로그램 만들기 ver2

■ SearchBtn_Click

```
private void SearchBtn_Click(object sender, EventArgs e)
{
    DS.Clear();

    DBAdapter.Fill(DS, "Phone");

    phoneTable = DS.Tables["Phone"];

    DataRow[] ResultRows = phoneTable.Select("PName like '%" + txtSearch.Text + "%'");

    NameList.Items.Clear();

    foreach (DataRow currRow in ResultRows)
    {
        NameList.Items.Add(currRow["Id"].ToString() + " " + currRow["PName"].ToString());
    }
}
```



DataSet으로 작업하기 원폼 프로그램 만들기 ver2

■ NameList_SelectedIndexChanged

```
private void NameList_SelectedIndexChanged(object sender, EventArgs e)
{
    DS.Clear();
    DBAdapter.Fill(DS, "Phone");
    phoneTable = DS.Tables["Phone"];

    DataRow[] ResultRows = phoneTable.Select("PName like '%" + txtSearch.Text + "%'");

    DataColumn[] PrimaryKey = new DataColumn[1];
    PrimaryKey[0] = phoneTable.Columns["id"];
    phoneTable.PrimaryKey = PrimaryKey;

    DataRow currRow = phoneTable.Rows.Find(NameList.Text.Substring(0, 2));/*

    SelectedKeyValue = Convert.ToInt32(currRow["id"].ToString());
    txtid.Text = currRow["id"].ToString();
    txtName.Text = currRow["PName"].ToString();
    txtMail.Text = currRow["Email"].ToString();
    txtPhone.Text = currRow["Phone"].ToString();
}
```



DataSet으로 작업하기 원폼 프로그램 만들기 ver2

■ 위로, 아래로 버튼

```
private void UpBtn_Click(object sender, EventArgs e)
{
    if (NameList.SelectedIndex != 0)
    {
        NameList.SelectedIndex = NameList.SelectedIndex - 1;
    }
    else
    {
        MessageBox.Show("이곳은 레코드의 처음입니다.");
    }
}
```

```
private void DownBtn_Click(object sender, EventArgs e)
{
    if (NameList.SelectedIndex != NameList.Items.Count - 1)
    {
        NameList.SelectedIndex = NameList.SelectedIndex + 1;
    }
    else
    {
        MessageBox.Show("이곳은 레코드의 마지막입니다.");
    }
}
```

C#과 Oracle11g 연동 프로그램



ADODForm(ver3)

다중 폼 정보 전달, 모듈 변수, 사용자 함수 정의
기능 => 전체보기, 추가, 수정, 삭제, 검색



DataSet으로 작업하기 원폼 프로그램 만들기 ver3

■ 실행 창

Form1

id: 3

이름: 이길동

전화번호: 010-333-3333

E-Mail: 333@bc.ac.kr

이름으로 찾기: 동 찾기

추가 수정 삭제

ID	PNAME	PHONE	EMAIL
1	홍길동	010-111-1111	111@bc.ac.kr
3	이길동	010-333-3333	333@bc.ac.kr
4	김길동	010-444-4444	444@bc.ac.kr
*			

DAOpenBtn

Form2

id: 3

이름: 이길동

전화번호: 010-333-3333

E-Mail: 333@bc.ac.kr

위로 아래로



DataSet으로 작업하기 원폼 프로그램 만들기 ver3

■ Form1 : SearchBtn_Click와 TxtS속성 정의

```
private void SearchBtn_Click(object sender, EventArgs e)
{
    Form2 frm2 = new Form2();
    frm2.Owner = this;
    frm2.ShowDialog();
    frm2.Dispose();
}

public String TxtS
{
    get { return txtSearch.Text; }
    set { txtSearch.Text = value.ToString(); }
}
```



DataSet으로 작업하기 원폼 프로그램 만들기 ver3

■ Form2 : Form2_Load

```
Form1 Parent; // Form2생성자 위에 선언
// ???   필요한 객체 선언
//수정하거나 삭제하기 위해 선택된 행의 인덱스를 저장한다.
    private int SelectedRowIndex;
    // Data Provider인 DBAdapter 입니다.
    OracleDataAdapter DBAdapter;
    // DataSet 객체입니다.
    DataSet DS;
    // 추가, 수정, 삭제시에 필요한 명령문을
    // 자동으로 작성해주는 객체입니다.
    OracleCommandBuilder myCommandBuilder;
    // ataTable 객체입니다.
    DataTable phoneTable;
    // 수정, 삭제할 때 필요한 레코드의 키값을 보관할 필드
    private int SelectedKeyValue;
private void Form2_Load(object sender, EventArgs e)
{
    DS.Clear();
    DBAdapter.Fill(DS, "Phone");
    Parent = (Form1)Owner;
    phoneTable = DS.Tables["Phone"];
    DataRow[] ResultRows = phoneTable.Select("PName like '%" + Parent.TxtS + "%'");
    NameList.Items.Clear();
    foreach (DataRow currRow in ResultRows)
    {
        NameList.Items.Add(currRow["Id"].ToString() + " " + currRow["PName"].ToString());
    }
}
```



DataSet으로 작업하기 원폼 프로그램 만들기 ver3

■ Form2 : NameList_SelectedIndexChanged

```
private void NameList_SelectedIndexChanged(object sender, EventArgs e)
{
    // DS.Clear();
    // DBAdapter.Fill(DS, "Phone");
    Parent = (Form1)Owner;
    phoneTable = DS.Tables["Phone"];

    DataRow[] ResultRows = phoneTable.Select("PName like '%" + Parent.TxtS + "%'");

    DataColumn[] PrimaryKey = new DataColumn[1];
    PrimaryKey[0] = phoneTable.Columns["id"];
    phoneTable.PrimaryKey = PrimaryKey;

    DataRow currRow = phoneTable.Rows.Find(NameList.Text.Substring(0, 2));

    SelectedKeyValue = Convert.ToInt32(currRow["id"].ToString());
    txtName.Text = currRow["PName"].ToString();
    txtMail.Text = currRow["Email"].ToString();
    txtPhone.Text = currRow["Phone"].ToString();
}
```



DataSet으로 작업하기 원폼 프로그램 만들기 ver3

■ 앞으로, 뒤로 버튼

```
private void PrevBtn_Click(object sender, EventArgs e)
{
    if (NameList.SelectedIndex != 0)
    {
        NameList.SelectedIndex = NameList.SelectedIndex - 1;
    }
    else
    {
        MessageBox.Show("이곳은 레코드의 처음입니다.");
    }
}
```

```
private void NextBtn_Click(object sender, EventArgs e)
{
    if (NameList.SelectedIndex != NameList.Items.Count - 1)
    {
        NameList.SelectedIndex = NameList.SelectedIndex + 1;
    }
    else
    {
        MessageBox.Show("이곳은 레코드의 마지막입니다.");
    }
}
```



ADOForm ver1에서 ver6까지 비교

■ ADOForm ver1에서 ver6까지 비교

ADOForm ver1 : 지역 변수
: 기능 => 전체보기, 추가, 수정, 삭제

ADOForm ver2 : 모듈 변수, 사용자 함수 정의
: 기능 => 전체보기, 추가, 수정, 삭제, 검색

ADOForm ver3 : 다중 폼 정보 전달, 모듈 변수, 사용자 함수 정의
: 기능 => 전체보기, 추가, 수정, 삭제, 검색

ADOForm ver4 : 다중 폼 정보 전달, 사용자 클래스 정의
: 기능 => 전체보기, 추가, 수정, 삭제, 검색

ADOForm ver5 : listview, DataReader, 외래키, 쿼리문(조인)

ADOForm ver6 : Parameters개체와 SQL의 매개변수(parameter) 활용
: 기능 => 검색