

기본 C#

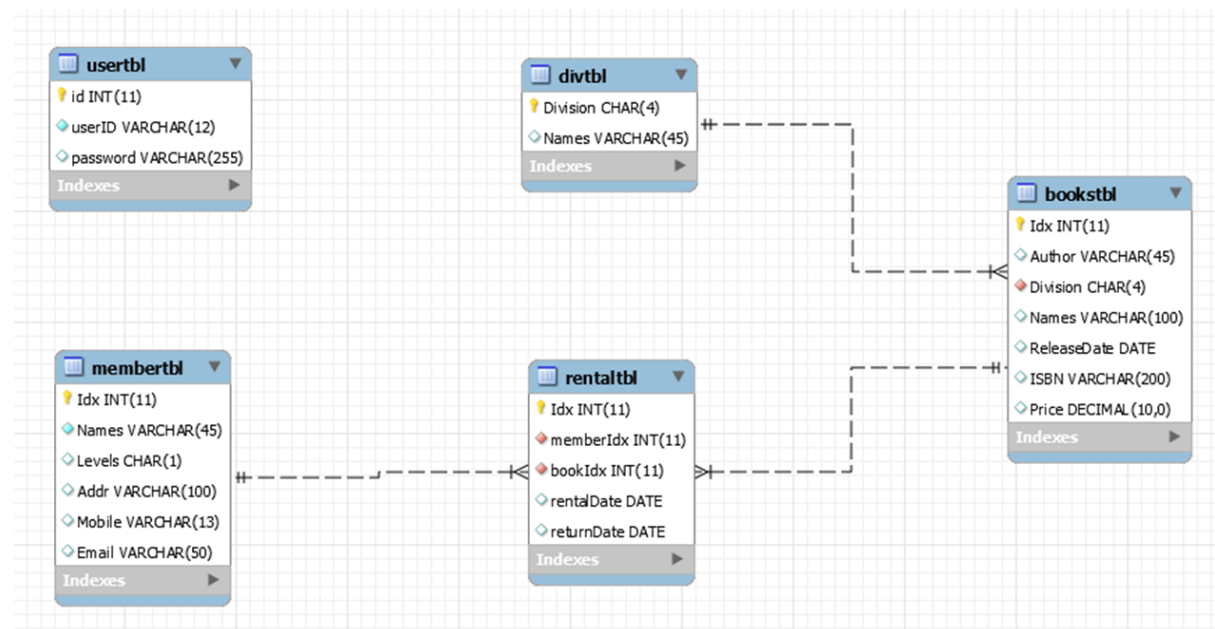
책대여점 앱 원품 만들기

개요

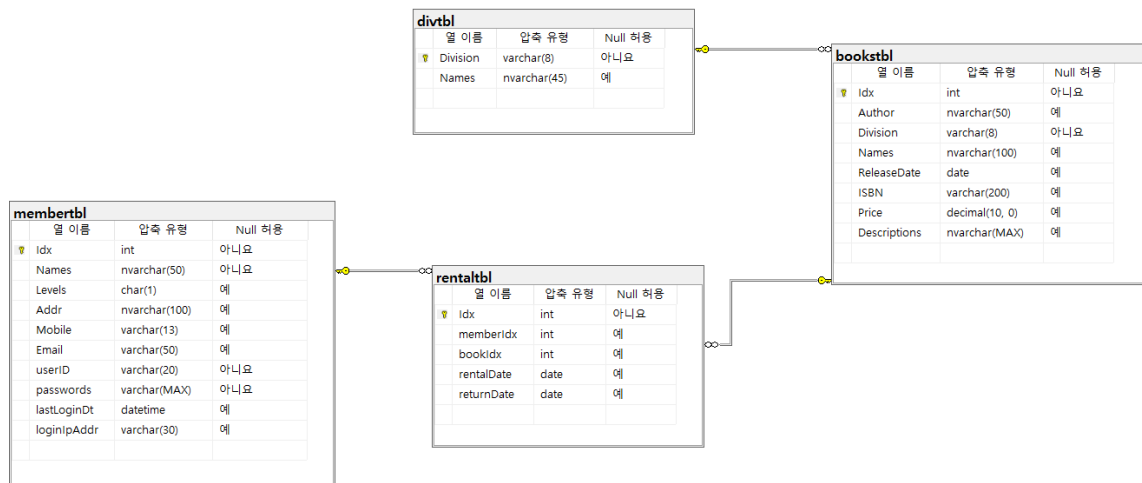
DataBase 강의 시 시험문제 MySQL DB(때에 따라 DB플랫폼은 변경될 수 있음)

BookRentalShop을 사용하여 원품 프로그램 개발을 도전(!)해 봅니다.

MySQL Reverse Engineering 다이어그램



SQL Server 다이어그램



SQL Server로 작업할 때는 User와 Member를 합쳐서 작업했습니다.

아래는 SQL Server의 DDL 쿼리문입니다.

```
USE [bookrentalshopDB]
GO

CREATE TABLE [dbo].[bookstbl](
    [Idx] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [Author] [nvarchar](50) NULL,
    [Division] [varchar](8) NOT NULL,
    [Names] [nvarchar](100) NULL,
    [ReleaseDate] [date] NULL,
    [ISBN] [varchar](200) NULL,
    [Price] [decimal](10, 0) NULL,
    [Descriptions] [nvarchar](max) NULL
)
GO

CREATE TABLE [dbo].[divtbl](
    [Division] [varchar](8) NOT NULL PRIMARY KEY,
    [Names] [nvarchar](45) NULL
)
GO

CREATE TABLE [dbo].[membertbl](
    [Idx] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [Names] [nvarchar](50) NOT NULL,
    [Levels] [char](1) NULL,
    [Addr] [nvarchar](100) NULL,
    [Mobile] [varchar](13) NULL,
    [Email] [varchar](50) NULL,
    [userID] [varchar](20) NOT NULL,
    [passwords] [varchar](max) NOT NULL,
    [lastLoginDt] [datetime] NULL,
    [loginIpAddr] [varchar](30) NULL
)
GO
```

```

GO

CREATE TABLE [dbo].[rentaltbl](
    [Idx] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [memberIdx] [int] NULL,
    [bookIdx] [int] NULL,
    [rentalDate] [date] NULL,
    [returnDate] [date] NULL
)
GO

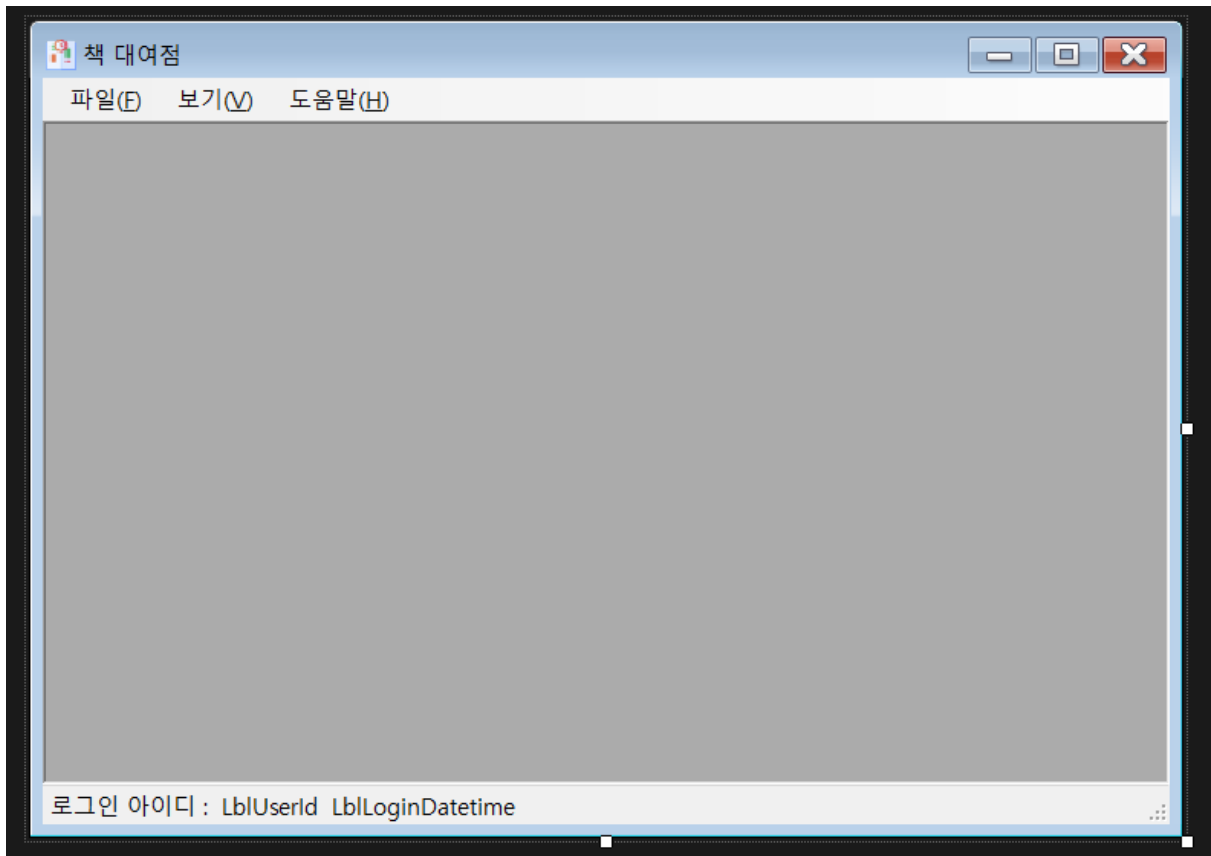
ALTER TABLE [dbo].[bookstbl] WITH CHECK ADD CONSTRAINT
[FK__bookstbl__Divtbl__267ABA7A] FOREIGN KEY([Division])
REFERENCES [dbo].[divtbl] ([Division])
GO
ALTER TABLE [dbo].[bookstbl] CHECK CONSTRAINT [FK__bookstbl__Divtbl__267ABA7A]
GO
ALTER TABLE [dbo].[rentaltbl] WITH CHECK ADD CONSTRAINT
[FK__rentaltbl__booktbl__2F10007B] FOREIGN KEY([bookIdx])
REFERENCES [dbo].[bookstbl] ([Idx])
GO
ALTER TABLE [dbo].[rentaltbl] CHECK CONSTRAINT [FK__rentaltbl__booktbl__2F10007B]
GO
ALTER TABLE [dbo].[rentaltbl] WITH CHECK ADD CONSTRAINT
[FK__rentaltbl__membertbl__2E1BDC42] FOREIGN KEY([memberIdx])
REFERENCES [dbo].[membertbl] ([Idx])
GO
ALTER TABLE [dbo].[rentaltbl] CHECK CONSTRAINT [FK__rentaltbl__membertbl__2E1BDC42]
GO

```

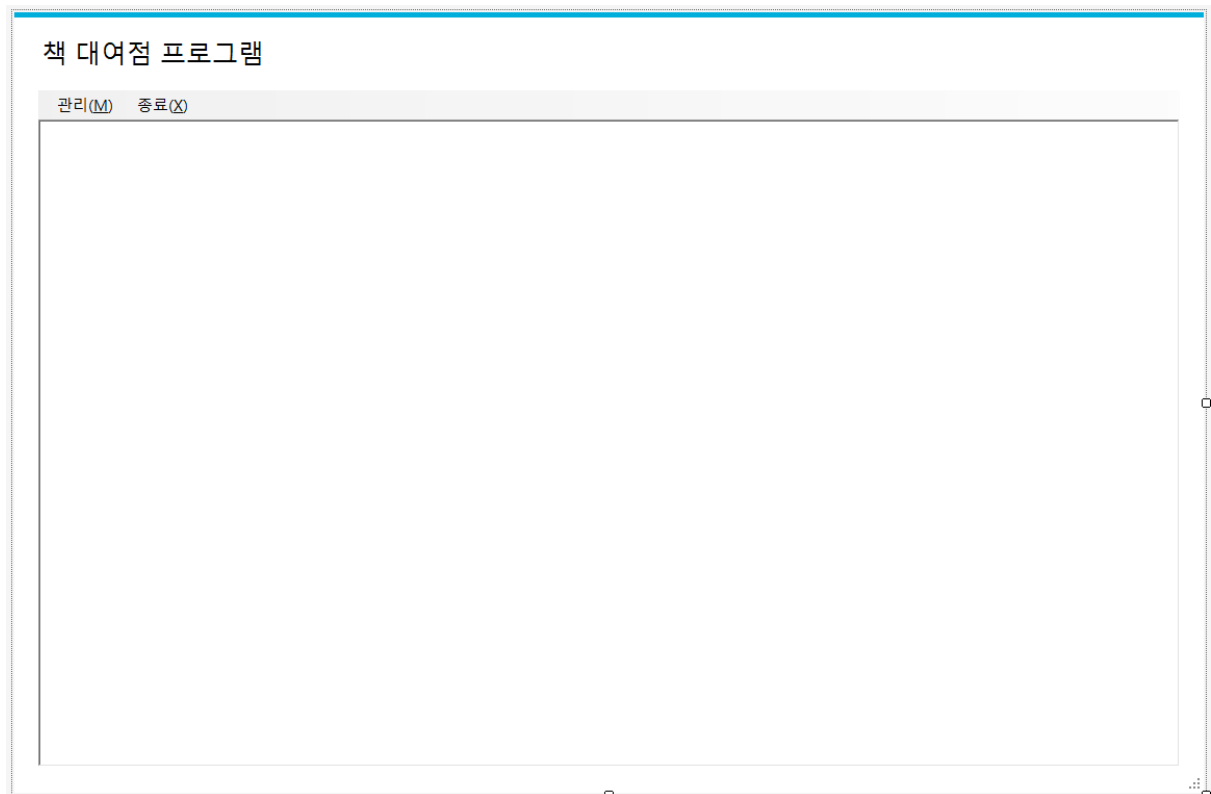
프로젝트 시작

메인폼 생성

MainForm.cs 입니다.



아래는 MetroUI 디자인을 적용한 화면입니다. 이 폼에서 가장 중요한 속성은 IsMdiContainer, True입니다.



기본 원품의 FrmMain 소스코드입니다.

```
using Org.BouncyCastle.Crypto.Tls;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using wf13_bookrentalshop.Helpers;

namespace wf13_bookrentalshop
{
    public partial class FrmMain : Form
    {
        #region < 각화면 품 >
        FrmGenre frmGenre = null; // 책장르관리 객체변수
        FrmBooks frmBooks = null; // 책정보관리
        #endregion

        #region < 생성자 >

        public FrmMain()
        {
            InitializeComponent();
        }

        #endregion

        #region < 이벤트 핸들러 영역 >

        private void FrmMain_Load(object sender, EventArgs e)
        {
            FrmLogin frm = new FrmLogin();
            frm.StartPosition = FormStartPosition.CenterScreen;
            frm.ShowDialog();

            LblUserId.Text = Commons.LoginID;
            LblLoginDatetime.Text = "/" + DateTime.Now.ToString();
        }

        private void FrmMain_FormClosing(object sender, FormClosingEventArgs e)
        {
            if (MessageBox.Show("종료하시겠습니까?", "확인", MessageBoxButtons.YesNo,
                MessageBoxIcon.Question) == DialogResult.Yes)
            {
                e.Cancel = false;
                Environment.Exit(0);
            }
            else
            {
                e.Cancel = true;
            }
        }
    }
}
```

```

private void ExitToolsStripMenuItem_Click(object sender, EventArgs e)
{
    Application.Exit(); // 전체 프로그램 종료!
}

private void MniGenre_Click(object sender, EventArgs e)
{
    //FrmGenre frm = new FrmGenre();
    //frm.TopLevel = false;
    //this.Controls.Add(frm);
    //frm.Show();
    frmGenre = ShowActiveForm(frmGenre, typeof(FrmGenre)) as FrmGenre;
}

#region < 임시 Collapse >

private void MniBookInfo_Click(object sender, EventArgs e)
{
    frmBooks = ShowActiveForm(frmBooks, typeof(FrmBooks)) as FrmBooks;
}

private void MniMember_Click(object sender, EventArgs e)
{
}

private void MniRental_Click(object sender, EventArgs e)
{
}

private void aboutToolStripMenuItem_Click(object sender, EventArgs e)
{
}

#endregion

#endregion

private Form ShowActiveForm(Form form, Type type)
{
    if (form == null) // 한번도 자식창을 안열었으면
    {
        form = (Form)Activator.CreateInstance(type); // 리플렉션으로 타입에 맞는
창을 새로 생성
        form.MdiParent = this; // FrmMain이 MDI부모
        form.WindowState = FormWindowState.Normal;
        form.Show();
    }
    else
    {
        if (form.IsDisposed) // 한번 닫혔다.
        {
            form = (Form)Activator.CreateInstance(type); // 리플렉션으로 타입에

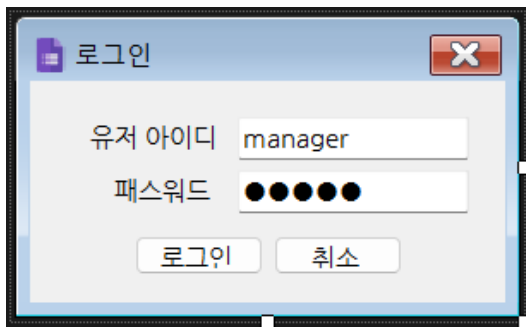
```

맞는 창을 새로 생성

```
        form.MdiParent = this; // FrmMain01 MDI부모
        form.WindowState = FormWindowState.Normal;
        form.Show();
    }
    else // 창이 열려있으면
    {
        form.Activate(); // 화면이 있으면 그 화면을 활성화
    }
}

return form;
}
}
```

로그인 팝업폼



LoginForm.cs 입니다.

```
using System;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using wf13_bookrentalshop.Helpers;

namespace wf13_bookrentalshop
{
    public partial class FrmLogin : Form
    {
        private bool isLoggedIn = false; // 로그인 성공했는지 변수

        public FrmLogin()
        {
            InitializeComponent();
        }

        private void BtnLogin_Click(object sender, EventArgs e)
        {
            isLoggedIn = LoginProcess(); // 로그인을 성공해야만 true가 됨
        }
    }
}
```

```

        if (isLoggedIn) this.Close();
    }

    private void BtnCancel_Click(object sender, EventArgs e)
    {
        // Application.Exit();
        Environment.Exit(0); // 가장 완벽하게 프로그램 종료
    }

    // 이게 없으면 X버튼이나 Alt+F4로 했을때 메인폼이 나타남
    private void FrmLogin_FormClosed(object sender, FormClosedEventArgs e)
    {
        if (isLoggedIn != true)
        { // 로그인 안되었을때 창을 닫으면 프로그램 모두 종료
            Environment.Exit(0);
        }
    }

    private void TxtPassword_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (e.KeyChar == 13) // 엔터키 누르면
        {
            BtnLogin_Click(sender, e); // 버튼클릭 이벤트핸들러 호출
        }
    }

    // DB userTbl에서 정보확인 로그인처리
    private bool LoginProcess()
    {
        // Validation check(입력 검증)
        if (string.IsNullOrEmpty(TxtUserId.Text))
        {
            MessageBox.Show("유저아이디를 입력하세요", "오류", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
            return false;
        }
        if (string.IsNullOrEmpty(TxtPassword.Text))
        {
            MessageBox.Show("패스워드를 입력하세요", "오류", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
            return false;
        }

        string strUserId = "";
        string strPassword = "";

        try
        {
            // DB처리
            using (MySqlConnection conn = new
            MySqlConnection(Helpers.Commons.ConnString))
            {
                conn.Open();

                #region << DB쿼리를 위한 구성 >>
                string selQuery = @"SELECT userId

```

```

        , password
        FROM usertbl
        WHERE userId = @userId
        AND password = @password";
        MySqlCommand selCmd = new MySqlCommand(selQuery, conn);
        // @userId, @password 파라미터 할당
        MySqlParameter prmUserID = new MySqlParameter("@userId",
TxtUserId.Text);
        MySqlParameter prmPassword = new MySqlParameter("@password",
TxtPassword.Text);
        selCmd.Parameters.Add(prmUserID);
        selCmd.Parameters.Add(prmPassword);
        #endregion

        MySqlDataReader reader = selCmd.ExecuteReader(); // 실행한 다음에
        // userId, password
        if (reader.Read())
        {
            strUserId = reader["userId"] != null ?
reader["userId"].ToString() : "-";
            strPassword = reader["password"] != null ?
reader["password"].ToString() : "--";
            Commons.LoginID = strUserId; // 프로그램전체에서 사용
            return true;
        }
        else
        {
            MessageBox.Show($"로그인정보가 없습니다.", "오류",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            Commons.LoginID = string.Empty; // 비상용
            return false;
        }
    } // conn.Close();
    //MessageBox.Show($"{strUserId} / {strPassword}"); // 디버깅용
}
catch (Exception ex)
{
    MessageBox.Show($"비정상적 오류 : {ex.Message}", "오류",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    return false;
}
}

// 유저아이디 텍스트박스에서 엔터를 치면 패스워드 텍스트박스로 포커스 이동
private void TxtUserId_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == 13)
    {
        TxtPassword.Focus();
    }
}
}
}
}

```

공통클래스

Commons.cs 클래스 파일을 생성합니다.

```
public class Common {
    public static string ConnString = "Data Source=127.0.0.1;" +
        "Initial Catalog=bookrentalshopDB;" +
        "Persist Security Info=True;" +
        "User ID=sa;" +
        "Password=*****";

    public static string LoginUserId = string.Empty;

    /// <summary>
    /// 아이피주소 받아오는 메서드
    /// </summary>
    /// <returns></returns>
    internal static string GetLocalIp() {
        string localIP = "";
        IPEndPoint host = Dns.GetHostEntry(Dns.GetHostName());
        foreach (IPAddress ip in host.AddressList)
        {
            if (ip.AddressFamily == System.Net.Sockets.AddressFamily.InterNetwork)
            {
                localIP = ip.ToString();
                break;
            }
        }

        return localIP;
    }

    internal static string ReplaceCmdText(string strSource) {
        var result = strSource.Replace("'", " ' ");
        result = result.Replace("--", "");
        result = result.Replace("; ", "");

        return result;
    }
}
```

MySQL이라면 아래의 내용을 추가해야 합니다.

```
internal class Commons
{
    // 모든 프로그램상에서 다 사용가능
    // DB연결문자열은 여기서만 수정하면 됨
    public static readonly string ConnString = "Server=localhost;" +
        "Port=3306;" +
        "Database=bookrentalshop;" +
        "Uid=root;" +
        "Pwd=12345";
}
```

```
// 로그인 사용자 아이디 저장변수
// 프로그램전체에서 이 데이터를 공유
public static string LoginID = string.Empty;
}
```

구분코드화면

FrmGenre.cs 입니다.

```
using MySql.Data.MySqlClient;
using System;
using System.Data;
using System.Diagnostics;
using System.Windows.Forms;
using wf13_bookrentalshop.Helpers;

namespace wf13_bookrentalshop
{
    public partial class FrmGenre : Form
    {
        bool isNew = false; // false(UPDATE) / true(INSERT)

        #region < 생성자 >

        public FrmGenre()
        {

```

```

        InitializeComponent();
    }

    #endregion

    #region < 이벤트 핸들러 >

    private void FrmGenre_Load(object sender, EventArgs e)
    {
        isNew = true; // 신규부터 시작
        RefreshData();
    }

    private void BtnNew_Click(object sender, EventArgs e)
    {
        ClearInputs();
    }

    private void BtnSave_Click(object sender, EventArgs e)
    {
        if (CheckValidation() != true) return;

        SaveData(); // 데이터 저장/수정
        RefreshData(); // 데이터 재조회
        ClearInputs(); // 입력창 클리어
    }

    private void BtnDel_Click(object sender, EventArgs e)
    {
        if (isNew == true) // 신규
        {
            MessageBox.Show("삭제할 데이터를 선택하세요", "오류", MessageBoxButtons.OK,
                MessageBoxIcon.Error);
            return;
        }

        // FK제약조건으로 지울수없는 데이터인지 먼저 확인
        using (MySqlConnection conn = new MySqlConnection(Commons.ConnString))
        {
            if (conn.State == ConnectionState.Closed) conn.Open();

            string strChkQuery = "SELECT COUNT(*) FROM bookstbl WHERE Division =
@Division";

            MySqlCommand chkCmd = new MySqlCommand(strChkQuery, conn);
            MySqlParameter prmDivision = new MySqlParameter("@Division",
                TxtDivision.Text);
            chkCmd.Parameters.Add(prmDivision);

            var result = chkCmd.ExecuteScalar();

            if (result.ToString() != "0")
            {
                MessageBox.Show("이미 사용중인 코드입니다.", "삭제",
                    MessageBoxButtons.OK, MessageBoxIcon.Information);
                return;
            }
        }
    }

```

```

    }

    // 삭제여부를 물을때 아니오를 누르면 삭제진행 취소
    if (MessageBox.Show("삭제하시겠습니까?", "삭제", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question) == DialogResult.No) return;

    // Yes를 누르면 계속 진행 // SaveData() 에있는 로직 복사->수정
    DeleteData(); // 데이터 삭제처리
    RefreshData(); // 지우고나서 재조회
    ClearInputs(); // 입력창 데이터 지우기
}

// 그리드뷰 클릭하면 발생이벤트
private void DgvResult_CellClick(object sender, DataGridViewCellEventArgs e)
{
    if (e.RowIndex > -1) // 아무것도 선택안하면 -1
    {
        var selData = DgvResult.Rows[e.RowIndex];
        //Debug.WriteLine(selData.ToString());
        Debug.WriteLine(selData.Cells[0].Value);
        Debug.WriteLine(selData.Cells[1].Value);
        TxtDivision.Text = selData.Cells[0].Value.ToString();
        TxtNames.Text = selData.Cells[1].Value.ToString();
        TxtDivision.ReadOnly = true; // PK는 수정하면 안됨

        isNew = false; // 수정
    }
}

#endregion

#region < 커스텀 메서드 >

private void ClearInputs()
{
    TxtDivision.Text = TxtNames.Text = string.Empty;
    TxtDivision.ReadOnly = false; // 신규일땐 입력가능
    TxtDivision.Focus();
    isNew = true; // 신규
}

// 입력검증
private bool CheckValidation()
{
    var result = true;
    var errorMsg = string.Empty;

    if (string.IsNullOrEmpty(TxtDivision.Text))
    {
        result = false;
        errorMsg += "● 장르코드를 입력하세요.\r\n";
    }

    if (string.IsNullOrEmpty(TxtNames.Text))
    {
        result = false;
    }
}

```

```

        errorMsg += "● 장르명을 입력하세요.\r\n";
    }

    if (result == false)
    {
        MessageBox.Show(errorMsg, "오류", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
        return result;
    }
    else
    {
        return result;
    }
}

private void RefreshData()
{
    // DB divtbl 데이터 조회 DgvResult 뿌림
    try
    {
        using (MySqlConnection conn = new
        MySqlConnection(Helpers.Commons.ConnString))
        {
            if (conn.State == ConnectionState.Closed) conn.Open();

            // 쿼리 작성
            var selQuery = @"SELECT Division
                                , Names
                                FROM divtbl";
            MySqlDataAdapter adapter = new MySqlDataAdapter(selQuery, conn);
            DataSet ds = new DataSet();
            adapter.Fill(ds, "divtbl"); // divtbl으로 DataSet 접근가능

            DgvResult.DataSource = ds.Tables[0];
            //DgvResult.DataSource = ds;
            //DgvResult.DataMember = "divtbl";

            DgvResult.Columns[0].HeaderText = "장르코드";
            DgvResult.Columns[1].HeaderText = "장르명";
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"비정상적 오류 : {ex.Message}", "오류",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

// isNew = true INSERT / false UPDATE
private void SaveData()
{
    // INSERT부터 시작
    try
    {
        using (MySqlConnection conn = new
        MySqlConnection(Helpers.Commons.ConnString))

```

```

        {
            if (conn.State == ConnectionState.Closed) conn.Open();

            var query = "";

            if (isNew)
            {
                query = @"INSERT INTO divtbl
                           VALUES (@Division, @Names)";
            }
            else
            {
                query = @"UPDATE divtbl
                           SET Names = @Names
                           WHERE Division = @Division";
            }

            MySqlCommand cmd = new MySqlCommand(query, conn);
            MySqlParameter prmDivision = new MySqlParameter("@Division",
TxtDivision.Text);
            MySqlParameter prmNames = new MySqlParameter("@Names",
TxtNames.Text);

            cmd.Parameters.Add(prmDivision);
            cmd.Parameters.Add(prmNames);

            var result = cmd.ExecuteNonQuery(); // INSERT, UPDATE, DELETE

            if (result != 0)
            {
                // 저장성공
                MessageBox.Show("저장성공!!", "저장", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            }
            else
            {
                // 저장실패
                MessageBox.Show("저장실패!!", "저장", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"비정상적 오류 : {ex.Message}", "오류",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void DeleteData()
{
    try
    {
        using (MySqlConnection conn = new
MySqlConnection(Helpers.Commons.ConnString))
        {
            if (conn.State == ConnectionState.Closed) conn.Open();

```

```

        var query = @"DELETE FROM divtbl
                        WHERE Division = @Division";

        MySqlCommand cmd = new MySqlCommand(query, conn);
        MySqlParameter prmDivision = new MySqlParameter("@Division",
TxtDivision.Text);
        cmd.Parameters.Add(prmDivision);

        var result = cmd.ExecuteNonQuery(); // INSERT, UPDATE, DELETE

        if (result != 0)
        {
            // 저장성공
            MessageBox.Show("삭제 성공!!", "삭제", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        }
        else
        {
            // 저장실패
            MessageBox.Show("삭제 실패!!", "삭제-", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }
}
catch (Exception ex)
{
    MessageBox.Show($"비정상적 오류 : {ex.Message}", "오류",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

#endregion

}
}

```

회원관리화면

FrmMember.cs

회원 관리

상세

순번:

이름:

레벨:

주소:

핸드폰:

이메일:

아이디:

패스워드:

삭제

신규

저장

```

public partial class FrmMember : MetroForm {
    #region 전역변수 영역

    private bool isNew = false; // false (수정) / true (신규)

    #endregion

    #region 이벤트 영역

    public FrmMember() {
        InitializeComponent();
    }

    private void FrmDivCode_Load(object sender, EventArgs e) {
        isNew = true; // 신규 초기화
        RefreshData();
    }

    private void FrmDivCode_Resize(object sender, EventArgs e) {
        DgvData.Height = this.ClientRectangle.Height - 90;
        GrbDetail.Height = this.ClientRectangle.Height - 90;
    }

    private void DgvData_CellClick(object sender, DataGridViewCellEventArgs e) {
        if (e.RowIndex > -1) // 선택된 값이 존재하면
        {
            var selData = DgvData.Rows[e.RowIndex];
            TxtIdx.Text = selData.Cells[0].Value.ToString();
            TxtNames.Text = selData.Cells[1].Value.ToString();
            CboLevels.SelectedItem = selData.Cells[2].Value;

            TxtAddr.Text = selData.Cells[3].Value.ToString();
            TxtMobile.Text = selData.Cells[4].Value.ToString();
        }
    }

    #endregion
}

```

```

        TxtEmail.Text = selData.Cells[5].Value.ToString();
        TxtUserId.Text = selData.Cells[6].Value.ToString();

        TxtIdx.ReadOnly = true;

        isNew = false; // 수정
    }
}

private void BtnDelete_Click(object sender, EventArgs e) {
    // Validation
    if (CheckValidation() == false) return;
    // 삭제질문
    if (MetroMessageBox.Show(this, "삭제하시겠습니까?", "삭제",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.No) return;

    DeleteData();
    RefreshData();
    ClearInputs();
}

private void BtnNew_Click(object sender, EventArgs e) {
    ClearInputs();
}

private void BtnSave_Click(object sender, EventArgs e) {
    // Validation 체크
    if (CheckValidation() == false) return;

    SaveData();
    RefreshData();
    ClearInputs();
}

#endregion

#region 커스텀 메서드 영역

/// <summary>
/// 삭제처리 프로세스
/// </summary>
private void DeleteData() {
    try
    {
        using (SqlConnection conn = new SqlConnection(Helper.Common.ConnString))
        {
            if (conn.State == ConnectionState.Closed) conn.Open();
            SqlCommand cmd = new SqlCommand();
            cmd.Connection = conn;

            var query = "";

            query = "DELETE FROM [dbo].[membertbl] " +
                " WHERE [Idx] = @Idx ";
            cmd.CommandText = query;

```

```

        var pIdx = new SqlParameter("@Idx", SqlDbType.Int);
        pIdx.Value = TxtIdx.Text;
        cmd.Parameters.Add(pIdx);

        var result = cmd.ExecuteNonQuery();
        if (result == 1)
        {
            MetroMessageBox.Show(this, "삭제 성공", "삭제",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            MetroMessageBox.Show(this, "삭제 실패", "삭제",
                MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
    }
}
catch (Exception ex)
{
    MetroMessageBox.Show(this, $"예외발생 : {ex.Message}", "오류",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}
}

private void RefreshData() {
    try
    {
        using (SqlConnection conn = new SqlConnection(Helper.Common.ConnString))
        {
            if (conn.State == ConnectionState.Closed) conn.Open();

            var query = @"SELECT [Idx]
                           , [Names]
                           , [Levels]
                           , [Addr]
                           , [Mobile]
                           , [Email]
                           , [userID]
                           , [lastLoginDt]
                           , [loginIpAddr]
                           FROM [dbo].[membertbl] ";
            SqlDataAdapter adapter = new SqlDataAdapter(query, conn);
            DataSet ds = new DataSet();
            adapter.Fill(ds, "membertbl");

            DgvData.DataSource = ds;
            DgvData.DataMember = "membertbl";
        }
    }
    catch (Exception ex)
    {
        MetroMessageBox.Show(this, $"예외발생 : {ex.Message}", "오류",
            MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }
}

```

```

}

/// <summary>
/// 입력(수정)처리 프로세스
/// </summary>
private void SaveData() {
    try
    {
        using (SqlConnection conn = new SqlConnection(Helper.Common.ConnString))
        {
            if (conn.State == ConnectionState.Closed) conn.Open();
            SqlCommand cmd = new SqlCommand();
            cmd.Connection = conn;

            var query = "";

            if (isNew == true) // INSERT
            {
                query = @"INSERT INTO [dbo].[membertbl]
                        ([Names]
                        ,[Levels]
                        ,[Addr]
                        ,[Mobile]
                        ,[Email]
                        ,[userID]
                        ,[passwords])
                VALUES
                (@Names
                ,@Levels
                ,@Addr
                ,@Mobile
                ,@Email
                ,@userID
                ,@passwords) ";
            }
            else // UPDATE
            {
                query = @"UPDATE [dbo].[membertbl]
                SET [Names] = @Names
                    ,[Levels] = @Levels
                    ,[Addr] = @Addr
                    ,[Mobile] = @Mobile
                    ,[Email] = @Email
                    ,[userID] = @userID
                    ,[passwords] = @passwords
                WHERE Idx = @Idx ";
            }
            cmd.CommandText = query;

            var pNames = new SqlParameter("@Names", SqlDbType.NVarChar, 50);
            pNames.Value = TxtNames.Text;
            cmd.Parameters.Add(pNames);

            var pLevels = new SqlParameter("@Levels", SqlDbType.Char, 1);
            pLevels.Value = CboLevels.SelectedItem.ToString();
            cmd.Parameters.Add(pLevels);

```

```

        var pAddr = new SqlParameter("@Addr", SqlDbType.NVarChar, 100);
        pAddr.Value = TxtAddr.Text;
        cmd.Parameters.Add(pAddr);

        var pMobile = new SqlParameter("@Mobile", SqlDbType.VarChar, 13);
        pMobile.Value = TxtMobile.Text;
        cmd.Parameters.Add(pMobile);

        var pEmail = new SqlParameter("@Email", SqlDbType.VarChar, 50);
        pEmail.Value = TxtEmail.Text;
        cmd.Parameters.Add(pEmail);

        var pUserId = new SqlParameter("@userId", SqlDbType.VarChar, 20);
        pUserId.Value = TxtUserId.Text;
        cmd.Parameters.Add(pUserId);

        var pPasswords = new SqlParameter("@passwords", SqlDbType.VarChar, 100);
        pPasswords.Value = TxtPasswords.Text;
        cmd.Parameters.Add(pPasswords);

        if (isNew == false) // Update 일때만 처리
        {
            var pIdx = new SqlParameter("@Idx", SqlDbType.Int);
            pIdx.Value = TxtIdx.Text;
            cmd.Parameters.Add(pIdx);
        }

        var result = cmd.ExecuteNonQuery();
        if (result == 1)
        {
            MetroMessageBox.Show(this, "저장 성공", "저장",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            MetroMessageBox.Show(this, "저장 실패", "저장",
                MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
    }
}
catch (Exception ex)
{
    MetroMessageBox.Show(this, $"예외발생 : {ex.Message}", "오류",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}
}

/// <summary>
/// 입력값 유효성 체크 메서드
/// </summary>
/// <returns></returns>
private bool CheckValidation() {
    if (string.IsNullOrEmpty(TxtNames.Text) ||
        string.IsNullOrEmpty(TxtAddr.Text) || string.IsNullOrEmpty(TxtMobile.Text)

```

```

||
    string.IsNullOrEmpty(TxtEmail.Text) || CboLevels.SelectedIndex == -1 ||
    string.IsNullOrEmpty(TxtUserId.Text))
{
    MetroMessageBox.Show(this, "빈값은 처리할 수 없습니다.", "경고",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    return false;
}

return true;
}

private void ClearInputs() {
    TxtIdx.Text = TxtNames.Text = "";
    TxtMobile.Text = TxtAddr.Text = TxtEmail.Text = "";
    TxtUserId.Text = TxtPasswords.Text = "";
    CboLevels.SelectedIndex = -1; // ?
    TxtIdx.ReadOnly = true;
    isNew = true;
}

#endregion
}

```

책관리화면

FrmBooks.cs

여기서 신경써야 할 것은 컨트롤의 TabIndex와 책장르 콤보박스, 출판일자 DateTimePicker, 책가격의 NumericUpDown 컨트롤 사용 등입니다.

```
using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.Data;
using System.Diagnostics;
using System.Windows.Forms;
using wf13_bookrentalshop.Helpers;

namespace wf13_bookrentalshop
{
    public partial class FrmBooks : Form
    {
        bool isNew = false; // false(UPDATE) / true(INSERT)

        #region < 생성자 >

        public FrmBooks()
        {
            InitializeComponent();
        }

        #endregion

        #region < 이벤트 핸들러 >

        private void FrmGenre_Load(object sender, EventArgs e)
        {
            isNew = true; // 신규부터 시작
            RefreshData();
            LoadCboData(); // 콤보박스에 들어갈 데이터 로드

            DtpReleaseDate.Format = DateTimePickerFormat.Custom;
            DtpReleaseDate.CustomFormat = "yyyy-MM-dd";
        }

        private void BtnNew_Click(object sender, EventArgs e)
        {
            ClearInputs();
        }

        private void BtnSave_Click(object sender, EventArgs e)
        {
            if (CheckValidation() != true) return;

            SaveData(); // 데이터 저장/수정
            RefreshData(); // 데이터 재조회
            ClearInputs(); // 입력창 클리어
        }

        private void BtnDel_Click(object sender, EventArgs e)
        {

```

```

        if (isNew == true) // 신규
        {
            MessageBox.Show("삭제할 데이터를 선택하세요", "오류", MessageBoxButtons.OK,
                MessageBoxIcon.Error);
            return;
        }

        // FK제약조건으로 지울수없는 데이터인지 먼저 확인
        using (MySqlConnection conn = new MySqlConnection(Commons.ConnString))
        {
            if (conn.State == ConnectionState.Closed) conn.Open();

            string strChkQuery = "SELECT COUNT(*) FROM rentaltbl WHERE bookIdx =
@bookIdx";

            MySqlCommand chkCmd = new MySqlCommand(strChkQuery, conn);
            MySqlParameter prmBookIdx = new MySqlParameter("@bookIdx",
                TxtBookIdx.Text);
            chkCmd.Parameters.Add(prmBookIdx);

            var result = chkCmd.ExecuteScalar();

            if (result.ToString() != "0")
            {
                MessageBox.Show("이미 대여중인 책입니다.", "삭제",
                    MessageBoxButtons.OK, MessageBoxIcon.Information);
                return;
            }
        }

        // 삭제여부를 물을때 아니오를 누르면 삭제진행 취소
        if (MessageBox.Show("삭제하시겠습니까?", "삭제", MessageBoxButtons.YesNo,
            MessageBoxIcon.Question) == DialogResult.No) return;

        // Yes를 누르면 계속 진행 // SaveData() 에있는 로직 복사->수정
        DeleteData(); // 데이터 삭제처리
        RefreshData(); // 지우고나서 재조회
        ClearInputs(); // 입력창 데이터 지우기
    }

    // 그리드뷰 클릭하면 발생이벤트
    private void DgvResult_CellClick(object sender, DataGridViewCellEventArgs e)
    {
        if (e.RowIndex > -1) // 아무것도 선택안하면 -1
        {
            var selData = DgvResult.Rows[e.RowIndex];
            //Debug.WriteLine(selData.ToString());
            Debug.WriteLine(selData.Cells[0].Value);
            Debug.WriteLine(selData.Cells[1].Value);
            TxtBookIdx.Text = selData.Cells[0].Value.ToString();
            TxtAuthor.Text = selData.Cells[1].Value.ToString();
            CboDivision.SelectedValue = selData.Cells[2].Value; // B001 == B001
            // selData.Cells[3] 사용안함
            TxtNames.Text = selData.Cells[4].Value.ToString();
            DtpReleaseDate.Value = (DateTime)selData.Cells[5].Value;
            TxtISBN.Text = selData.Cells[6].Value.ToString();
            NudPrice.Text = selData.Cells[7].Value.ToString();
        }
    }

```

```

        isNew = false; // 수정
    }
}

private void DgvResult_DataBindingComplete(object sender,
DataGridViewBindingCompleteEventArgs e)
{
    DgvResult.ClearSelection(); // 최초에 첫번째열 첫번째셀 선택되어있는것을 해제
}

#endregion

#region < 커스텀 메서드 >

private void ClearInputs()
{
    TxtBookIdx.Text = TxtAuthor.Text = string.Empty;
    TxtNames.Text = TxtISBN.Text = string.Empty;
    CboDivision.SelectedIndex = -1;
    DtpReleaseDate.Value = DateTime.Now; // 오늘날짜로 초기화
    NudPrice.Value = 0;
    TxtAuthor.Focus(); // 번호는 입력안함
    isNew = true; // 신규
}

// 입력검증
private bool CheckValidation()
{
    var result = true;
    var errorMsg = string.Empty;

    if (string.IsNullOrEmpty(TxtAuthor.Text))
    {
        result = false;
        errorMsg += "● 저자명을 입력하세요.\r\n";
    }

    if (CboDivision.SelectedIndex < 0)
    {
        result = false;
        errorMsg += "● 장르를 선택하세요.\r\n";
    }

    if (string.IsNullOrEmpty(TxtNames.Text))
    {
        result = false;
        errorMsg += "● 책제목을 입력하세요.\r\n";
    }

    if (DtpReleaseDate.Value == null)
    {
        result = false;
        errorMsg += "● 출판일자를 선택하세요.\r\n";
    }
}

```

```

        if (result == false)
        {
            MessageBox.Show(errorMsg, "오류", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            return result;
        }
        else
        {
            return result;
        }
    }

    private void RefreshData()
    {
        // DB divtbl 데이터 조회 DgvResult 뿌림
        try
        {
            using (MySqlConnection conn = new
MySqlConnection(Helpers.Commons.ConnString))
            {
                if (conn.State == ConnectionState.Closed) conn.Open();

                // 쿼리 작성
                var selQuery = @"SELECT b.bookIdx,
                                    b.Author,
                                    b.Division,
                                    d.Names AS DivNames,
                                    b.Names,
                                    b.ReleaseDate,
                                    b.ISBN,
                                    b.Price
                                FROM bookstbl AS b
                                INNER JOIN divtbl AS d
                                ON b.Division = d.Division
                                ORDER BY b.bookIdx ASC "; // 정렬때문에 추가
                MySqlDataAdapter adapter = new MySqlDataAdapter(selQuery, conn);
                DataSet ds = new DataSet();
                adapter.Fill(ds, "bookstbl"); // bookstbl DataSet 접근가능

                DgvResult.DataSource = ds.Tables[0];
                // 데이터그리드뷰 컬럼 헤더 제목
                DgvResult.Columns[0].HeaderText = "번호";
                DgvResult.Columns[1].HeaderText = "저자명";
                DgvResult.Columns[2].HeaderText = "책장르";
                DgvResult.Columns[3].HeaderText = "책장르";
                DgvResult.Columns[4].HeaderText = "책제목";
                DgvResult.Columns[5].HeaderText = "출판일자";
                DgvResult.Columns[6].HeaderText = "ISBN";
                DgvResult.Columns[7].HeaderText = "책가격";
                // 컬럼 넓이 또는 보이기
                DgvResult.Columns[0].Width = 55;
                DgvResult.Columns[2].Visible = false; // B001 코드영역 보일필요 없음
                DgvResult.Columns[5].Width = 78;
                DgvResult.Columns[7].Width = 80;
                // 컬럼 데이터 정렬
                DgvResult.Columns[0].DefaultCellStyle.Alignment =

```

```

DataGridViewContentAlignment.MiddleRight;
            DgvResult.Columns[5].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;
            DgvResult.Columns[7].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleRight;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"RefreshData() 비정상적 오류 : {ex.Message}", "오류",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void LoadCboData()
{
    try
    {
        using (MySqlConnection conn = new MySqlConnection(Commons.ConnString))
        {
            if (conn.State == ConnectionState.Closed) { conn.Open(); }
            var query = "SELECT Division, Names FROM divtbl";
            MySqlCommand cmd = new MySqlCommand(query, conn);
            MySqlDataReader reader = cmd.ExecuteReader();
            var temp = new Dictionary<string, string>();
            while (reader.Read())
            {
                temp.Add(reader[0].ToString(), reader[1].ToString()); //
                (Key)B001, (Value)공포/스릴러
            }

            // 콤보박스에 할당
            CboDivision.DataSource = new BindingSource(temp, null);
            CboDivision.DisplayMember = "Value";
            CboDivision.ValueMember = "Key";
            CboDivision.SelectedIndex = -1;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"LoadCboData() 비정상적 오류 : {ex.Message}", "오류",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

// isNew = true INSERT / false UPDATE
private void SaveData()
{
    // INSERT부터 시작
    try
    {
        using (MySqlConnection conn = new
        MySqlConnection(Helpers.Commons.ConnString))
        {
            if (conn.State == ConnectionState.Closed) conn.Open();

```

```

var query = "";

if (isNew)
{
    query = @"INSERT INTO bookstbl
                (Author,
                 Division,
                 Names,
                 ReleaseDate,
                 ISBN,
                 Price)
            VALUES
                (@Author,
                 @Division,
                 @Names,
                 @ReleaseDate,
                 @ISBN,
                 @Price)";
}
else
{
    query = @"UPDATE bookstbl
                SET Author = @Author,
                    Division = @Division,
                    Names = @Names,
                    ReleaseDate = @ReleaseDate,
                    ISBN = @ISBN,
                    Price = @Price
                WHERE bookIdx = @bookIdx";
}

MySQLCommand cmd = new MySQLCommand(query, conn);
MySQLParameter prmAuthor = new MySQLParameter("@Author",
TxtAuthor.Text);
MySQLParameter prmDivision = new MySQLParameter("@Division",
CboDivision.SelectedValue.ToString());
MySQLParameter prmNames = new MySQLParameter("@Names",
TxtNames.Text);
MySQLParameter prmReleaseDate = new MySQLParameter("@ReleaseDate",
DtpReleaseDate.Value);
MySQLParameter prmISBN = new MySQLParameter("@ISBN", TxtISBN.Text);
MySQLParameter prmPrice = new MySQLParameter("@Price",
NudPrice.Value);

cmd.Parameters.Add(prmAuthor);
cmd.Parameters.Add(prmDivision);
cmd.Parameters.Add(prmNames);
cmd.Parameters.Add(prmReleaseDate);
cmd.Parameters.Add(prmISBN);
cmd.Parameters.Add(prmPrice);

if (isNew == false) // update할 때는 bookIdx 파라미터를 추가!!
{
    MySQLParameter prmBookIdx = new MySQLParameter("@bookIdx",
TxtBookIdx.Text);

    cmd.Parameters.Add(prmBookIdx);
}

```

```

        var result = cmd.ExecuteNonQuery(); // INSERT, UPDATE, DELETE

        if (result != 0)
        {
            // 저장성공
            MessageBox.Show("저장성공!!", "저장", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        }
        else
        {
            // 저장실패
            MessageBox.Show("저장실패!!", "저장", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"비정상적 오류 : {ex.Message}", "오류",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void DeleteData()
{
    try
    {
        using (MySqlConnection conn = new
        MySqlConnection(Helpers.Commons.ConnString))
        {
            if (conn.State == ConnectionState.Closed) conn.Open();

            var query = @"DELETE FROM bookstbl
                           WHERE bookIdx = @bookIdx";

            MySqlCommand cmd = new MySqlCommand(query, conn);
            MySqlParameter prmBookIdx = new MySqlParameter("@bookIdx",
            TxtBookIdx.Text);
            cmd.Parameters.Add(prmBookIdx);

            var result = cmd.ExecuteNonQuery(); // INSERT, UPDATE, DELETE

            if (result != 0)
            {
                // 저장성공
                MessageBox.Show("삭제성공!!", "삭제", MessageBoxButtons.OK,
                MessageBoxIcon.Information);
            }
            else
            {
                // 저장실패
                MessageBox.Show("삭제실패!!", "삭제-", MessageBoxButtons.OK,
                MessageBoxIcon.Error);
            }
        }
    }
}

```

```

    }
    catch (Exception ex)
    {
        MessageBox.Show($"비정상적 오류 : {ex.Message}", "오류",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

#endregion

}
}

```

대여관리화면

FrmRental.cs

```

public partial class FrmRental : MetroForm {
    #region 전역변수 영역

    private bool isNew = false; // false (수정) / true (신규)

    #endregion

    #region 이벤트 영역

    public FrmRental()
    {
        InitializeComponent();
    }
}

```

```

}

private void FrmDivCode_Load(object sender, EventArgs e)
{
    isNew = true; // 신규 초기화
    InitCboData(); // 콤보박스 들어가는 데이터 초기화
    RefreshData(); // 테이블 조회

    DtpRentalDate.CustomFormat = "yyyy-MM-dd";
    DtpRentalDate.Format = DateTimePickerFormat.Custom;
}

private void FrmDivCode_Resize(object sender, EventArgs e)
{
    DgvData.Height = this.ClientRectangle.Height - 90;
    GrbDetail.Height = this.ClientRectangle.Height - 90;
}

private void DgvData_CellClick(object sender, DataGridViewCellEventArgs e)
{
    if (e.RowIndex > -1) // 선택된 값이 존재하면
    {
        var selData = DgvData.Rows[e.RowIndex];
        AssignToControls(selData);
        isNew = false; // 수정
        BtnSearchBook.Enabled = BtnSearchMember.Enabled = false;
        DtpRentalDate.Enabled = false;
    }
}

private void BtnDelete_Click(object sender, EventArgs e)
{
    // Validation
    if (CheckValidation() == false) return;
    // 삭제질문
    if (MetroMessageBox.Show(this, "삭제하시겠습니까?", "삭제",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.No) return;

    DeleteData();
    RefreshData();
    ClearInputs();
}

private void BtnNew_Click(object sender, EventArgs e)
{
    ClearInputs();
}

private void BtnSave_Click(object sender, EventArgs e)
{
    // Validation 체크
    if (CheckValidation() == false) return;

    SaveData();
    RefreshData();
    ClearInputs();
}

```

```

}

#endregion

#region 커스텀 메서드 영역

private void InitCboData()
{
    try
    {
        var temp = new Dictionary<string, string>();
        temp.Add("R", "대여중");
        temp.Add("T", "반납");

        CboRentalState.DataSource = new BindingSource(temp, null);
        CboRentalState.DisplayMember = "Value";
        CboRentalState.ValueMember = "Key";
        CboRentalState.SelectedIndex = -1;
    }
    catch { }
}

private void AssignToControls(DataGridViewRow selData)
{
    TxtIdx.Text = selData.Cells[0].Value.ToString();
    selMemberIdx = (int)selData.Cells[1].Value;
    Debug.WriteLine($">>>> selMemberIdx : {selMemberIdx}");
    TxtMemberName.Text = selData.Cells[2].Value.ToString();
    selBookIdx = (int)selData.Cells[3].Value;
    Debug.WriteLine($">>>> selBookIdx : {selBookIdx}");
    TxtBookName.Text = selData.Cells[4].Value.ToString();
    DtpRentalDate.Value = (DateTime)selData.Cells[5].Value;
    TxtReturnDate.Text = selData.Cells[6].Value == null ? "" :
selData.Cells[6].Value.ToString();
    CboRentalState.SelectedValue = selData.Cells[7].Value;
}

/// <summary>
/// 삭제처리 프로세스
/// </summary>
private void DeleteData()
{
    try
    {
        using (SqlConnection conn = new SqlConnection(helper.Common.ConnString))
        {
            if (conn.State == ConnectionState.Closed) conn.Open();
            SqlCommand cmd = new SqlCommand();
            cmd.Connection = conn;

            var query = @"DELETE FROM [dbo].[bookstbl]
                        WHERE [Idx] = @Idx ";
            cmd.CommandText = query;

            var pIdx = new SqlParameter("@Idx", SqlDbType.Int);
            pIdx.Value = TxtIdx.Text;

```

```

cmd.Parameters.Add(pIdx);

var result = cmd.ExecuteNonQuery();
if (result == 1)
{
    MetroMessageBox.Show(this, "삭제 성공", "삭제",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
}
else
{
    MetroMessageBox.Show(this, "삭제 실패", "삭제",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
}
}
}
catch (Exception ex)
{
    MetroMessageBox.Show(this, $"예외발생 : {ex.Message}", "오류",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}
}

private void RefreshData()
{
    try
    {
        using (SqlConnection conn = new SqlConnection(Helper.Common.ConnString))
        {
            if (conn.State == ConnectionState.Closed) conn.Open();

            var query = @"SELECT r.Idx
                        ,r.memberIdx
                        ,m.Names as memberName
                        ,r.bookIdx
                        ,b.Names as bookName
                        ,r.rentalDate
                        ,r.returnDate
                        ,r.rentalState
                        ,case r.rentalState
                            when 'R' then '대여중'
                            when 'T' then '반납'
                            else '상태없음'
                        end as StateDesc
                        FROM dbo.rentaltbl as r
                        INNER JOIN dbo.membertbl as m
                        ON r.memberIdx = m.Idx
                        INNER JOIN dbo.bookstbl as b
                        ON r.bookIdx = b.Idx "; //

            SqlDataAdapter adapter = new SqlDataAdapter(query, conn);
            DataSet ds = new DataSet();
            adapter.Fill(ds, "rentaltbl");

            DgvData.DataSource = ds;
            DgvData.DataMember = "rentaltbl";
        }
    }
}

```

```

    }
    catch (Exception ex)
    {
        MetroMessageBox.Show(this, $"예외발생 : {ex.Message}", "오류",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }

    // 데이터그리드뷰 컬럼 화면에서 안보이게
    var column = DgvData.Columns[1]; // memberId 컬럼
    column.Visible = false;
    column = DgvData.Columns[3]; // bookIdx
    column.Visible = false;
    column = DgvData.Columns[7]; // rentalState
    column.Visible = false;

    column = DgvData.Columns[0]; // Idx
    column.DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleRight;
}

/// <summary>
/// 입력(수정)처리 프로세스
/// </summary>
private void SaveData()
{
    try
    {
        using (SqlConnection conn = new SqlConnection(Helper.Common.ConnString))
        {
            if (conn.State == ConnectionState.Closed) conn.Open();
            SqlCommand cmd = new SqlCommand();
            cmd.Connection = conn;

            var query = "";

            if (isNew == true) // ISNERT
            {
                query = @"INSERT INTO [dbo].[rentaltbl]
                        ([memberIdx]
                        ,[bookIdx]
                        ,[rentalDate]
                        ,[rentalState])
                        VALUES
                        (@memberIdx
                        ,@bookIdx
                        ,@rentalDate
                        ,@rentalState) ";
            }
            else // UPDATE
            {
                query = @"UPDATE [dbo].[rentaltbl]
                        SET [returnDate] = case @rentalState
                                when 'T' then GETDATE()
                                when 'R' then null end
                        ,[rentalState] = @rentalState
                        WHERE Idx = @Idx ";
            }
        }
    }
}

```

```

    }
    cmd.CommandText = query;

    if (isNew == true) // 신규
    {
        var pMemberIdx = new SqlParameter("@memberIdx", SqlDbType.Int);
        pMemberIdx.Value = selMemberIdx;
        cmd.Parameters.Add(pMemberIdx);

        var pBookIdx = new SqlParameter("@bookIdx", SqlDbType.Int);
        pBookIdx.Value = selBookIdx;
        cmd.Parameters.Add(pBookIdx);

        var pRentalDate = new SqlParameter("@rentalDate", SqlDbType.Date);
        pRentalDate.Value = DtpRentalDate.Value;
        cmd.Parameters.Add(pRentalDate);

        var pRentalState = new SqlParameter("@rentalState", SqlDbType.Char,
1);

        pRentalState.Value = CboRentalState.SelectedValue;
        cmd.Parameters.Add(pRentalState);
    }
    else // 업데이트일때
    {
        var pRentalState = new SqlParameter("@rentalState", SqlDbType.Char,
1);

        pRentalState.Value = CboRentalState.SelectedValue;
        cmd.Parameters.Add(pRentalState);

        var pIdx = new SqlParameter("@Idx", SqlDbType.Int);
        pIdx.Value = TxtIdx.Text;
        cmd.Parameters.Add(pIdx);
    }

    var result = cmd.ExecuteNonQuery();
    if (result == 1)
    {
        MetroMessageBox.Show(this, "저장 성공", "저장",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        MetroMessageBox.Show(this, "저장 실패", "저장",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
    }
}
catch (Exception ex)
{
    MetroMessageBox.Show(this, $"예외발생 : {ex.Message}", "오류",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}
}

/// <summary>

```

```

/// 입력값 유효성 체크 메서드
/// </summary>
/// <returns></returns>
private bool CheckValidation()
{
    if (string.IsNullOrEmpty(TxtMemberName.Text) ||
        string.IsNullOrEmpty(TxtBookName.Text) ||
        DtpRentalDate.Value == null ||
        CboRentalState.SelectedIndex < 0)
    {
        MetroMessageBox.Show(this, "빈값은 처리할 수 없습니다.", "경고",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return false;
    }

    return true;
}

private void ClearInputs()
{
    selMemberIdx = selBookIdx = 0;
    selMemberName = selBookName = "";
    TxtIdx.Text = TxtBookName.Text = TxtMemberName.Text = "";
    DtpRentalDate.Value = DateTime.Now; // 오늘 날짜로 초기화
    TxtReturnDate.Text = "";
    TxtIdx.ReadOnly = true;
    CboRentalState.SelectedIndex = -1;

    BtnSearchBook.Enabled = BtnSearchMember.Enabled = true;
    DtpRentalDate.Enabled = true;
    isNew = true;
}

#endregion

private int selMemberIdx = 0; // 선택된 회원번호
private string selMemberName = ""; // 선택된 회원이름

private void BtnSearchMember_Click(object sender, EventArgs e)
{
    FrmMemberPopup frm = new FrmMemberPopup();
    frm.StartPosition = FormStartPosition.CenterParent;
    if (frm.ShowDialog() == DialogResult.OK)
    {
        selMemberIdx = frm.SelIdx;
        TxtMemberName.Text = selMemberName = frm.SelName;
    }
}

private int selBookIdx = 0;
private string selBookName = "";

private void BtnSearchBook_Click(object sender, EventArgs e)
{
    FrmBooksPopup frm = new FrmBooksPopup();
    frm.StartPosition = FormStartPosition.CenterParent;

```

```

        if (frm.ShowDialog() == DialogResult.OK)
        {
            selBookIdx = frm.SelIdx;
            TxtBookName.Text = selBookName = frm.SelName;
        }
    }
}

```

암호화 추가

Commons.cs에 암호화 부분을 추가합니다.

```

public static string GetMd5Hash(MD5 md5Hash, string input)
{
    byte[] data = md5Hash.ComputeHash(Encoding.UTF8.GetBytes(input));
    StringBuilder sBuilder = new StringBuilder();
    for (int i = 0; i < data.Length; i++)
    {
        sBuilder.Append(data[i].ToString("x2"));
    }
    return sBuilder.ToString();
}

```

엑셀 내보내기 기능

엑셀 내보내기 기능을 Commons.cs에 추가합니다. NPOI 패키지를 추가하여 작업을 수행합니다.

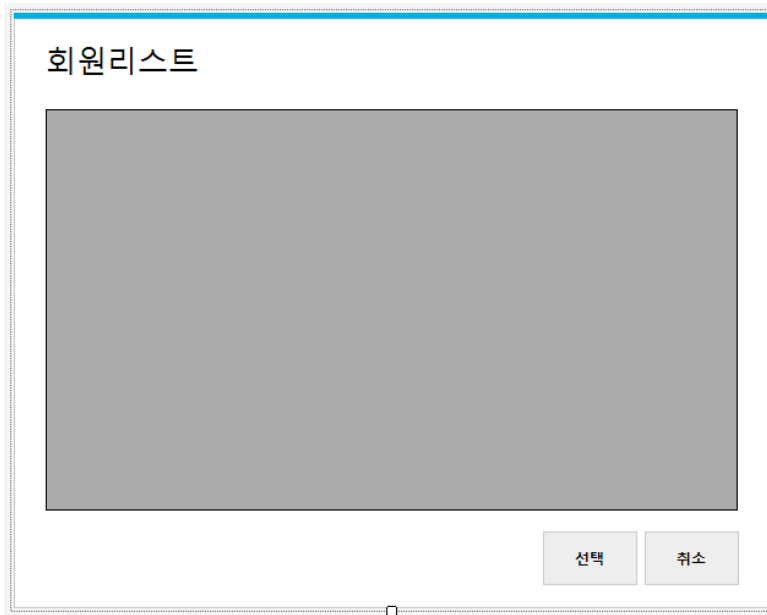
```

private void BtnExport_Click(object sender, EventArgs e) {
    IWorkbook workbook = new XSSFWorkbook(); // HSSFWorkbook
    ISheet sheet1 = workbook.CreateSheet("Sheet1");
    sheet1.CreateRow(0).CreateCell(0).SetCellValue("This is a Sample");
    int x = 1;
    DataSet dset = GrdRentalTbl.DataSource as DataSet;
    for (int i = 0; i < dset.Tables[0].Rows.Count; i++) { // row
        IRow row = sheet1.CreateRow(i);
        for (int j = 0; j < dset.Tables[0].Rows[i].ItemArray.Length; j++) {
            row.CreateCell(j).SetCellValue(dset.Tables[0].Rows[i].ItemArray[j].ToString());
        }
    }
    FileStream sw = File.Create("test.xlsx"); // xls
    workbook.Write(sw);
    sw.Close();
    MetroMessageBox.Show(this, "Export완료", "Export");
}

```

회원 팝업화면

FrmMemberPopup.cs



```
public partial class FrmMemberPopup : MetroForm
{
    #region 전역변수 영역

    public int SelIdx { get; set; }

    public string SelName { get; set; }

    #endregion

    #region 이벤트 영역

    public FrmMemberPopup()
    {
        InitializeComponent();
    }

    private void FrmDivCode_Load(object sender, EventArgs e)
    {
        RefreshData();
    }

    private void FrmDivCode_Resize(object sender, EventArgs e)
    {
    }

    #endregion

    #region 커스텀 메서드 영역
```

```

private void RefreshData()
{
    try
    {
        using (SqlConnection conn = new SqlConnection(Helper.Common.ConnString))
        {
            if (conn.State == ConnectionState.Closed) conn.Open();

            var query = @"SELECT [Idx]
                           , [Names]
                           , [Levels]
                           , [Addr]
                           , [Mobile]
                           , [Email]
                           FROM [dbo].[membertbl] ";
            SqlDataAdapter adapter = new SqlDataAdapter(query, conn);
            DataSet ds = new DataSet();
            adapter.Fill(ds, "membertbl");

            DgvData.DataSource = ds;
            DgvData.DataMember = "membertbl";
        }
    }
    catch (Exception ex)
    {
        MetroMessageBox.Show(this, $"예외발생 : {ex.Message}", "오류",
            MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }

    DgvData.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
}

#endregion

private void BtnCancel_Click(object sender, EventArgs e)
{
    this.DialogResult = DialogResult.Cancel;
    this.Close();
}

private void BtnSelect_Click(object sender, EventArgs e)
{
    if (DgvData.SelectedRows.Count == 0)
    {
        MetroMessageBox.Show(this, "데이터를 선택하세요", "경고",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    SelIdx = (int)DgvData.SelectedRows[0].Cells[0].Value;
    SelName = DgvData.SelectedRows[0].Cells[1].Value.ToString();

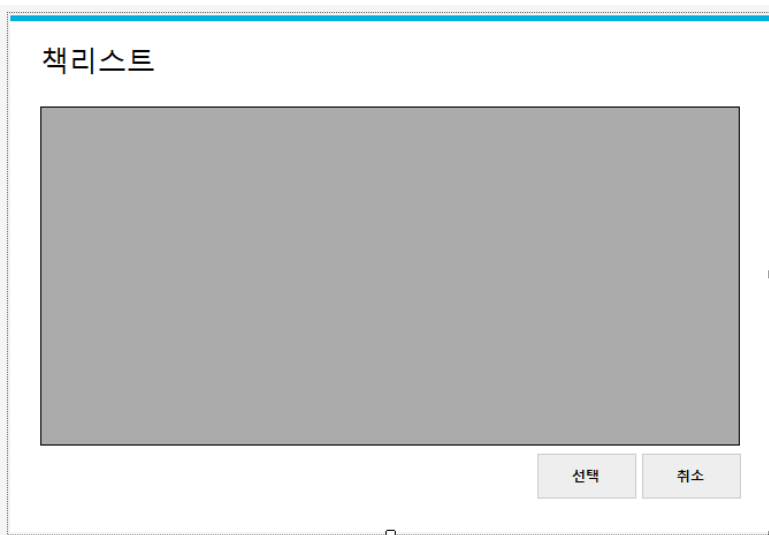
    this.DialogResult = DialogResult.OK;
    this.Close();
}

```

```
}
```

책 팝업화면

FrmBookPopup.cs



```
public partial class FrmBooksPopup : MetroForm
{
    #region 전역변수 영역

    public int SelIdx { get; set; }

    public string SelName { get; set; }

    #endregion

    #region 이벤트 영역

    public FrmBooksPopup()
    {
        InitializeComponent();
    }

    private void FrmDivCode_Load(object sender, EventArgs e)
    {
        RefreshData(); // 테이블 조회
    }

    private void FrmDivCode_Resize(object sender, EventArgs e)
    {
    }
}
```

```

private void DgvData_CellClick(object sender, DataGridViewCellEventArgs e)
{
}

#endregion

#region 커스텀 메서드 영역

private void RefreshData()
{
    try
    {
        using (SqlConnection conn = new SqlConnection(Helper.Common.ConnString))
        {
            if (conn.State == ConnectionState.Closed) conn.Open();

            var query = @"SELECT b.Idx
                        ,b.Author
                        ,b.Division
                        ,d.Names as DivName
                        ,b.Names
                        ,b.ReleaseDate
                        FROM dbo.bookstbl as b
                        INNER JOIN dbo.divtbl as d
                        ON b.Division = d.Division "; // 210318 Descriptions

            SqlDataAdapter adapter = new SqlDataAdapter(query, conn);
            DataSet ds = new DataSet();
            adapter.Fill(ds, "bookstbl");

            DgvData.DataSource = ds;
            DgvData.DataMember = "bookstbl";
        }
    }
    catch (Exception ex)
    {
        MetroMessageBox.Show(this, $"예외발생 : {ex.Message}", "오류",
            MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }

    // 데이터그리드뷰 컬럼 화면에서 안보이게
    var column = DgvData.Columns[2]; // Division 컬럼
    column.Visible = false;

    column = DgvData.Columns[4];
    column.Width = 250;
    column.HeaderText = "도서명";

    column = DgvData.Columns[0]; // Idx
    column.DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleRight;

    DgvData.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
}

#endregion

```

```

private void BtnCancel_Click(object sender, EventArgs e)
{
    this.DialogResult = DialogResult.Cancel;
    this.Close();
}

private void BtnSelect_Click(object sender, EventArgs e)
{
    if (DgvData.SelectedRows.Count == 0)
    {
        MetroMessageBox.Show(this, "데이터를 선택하세요", "경고",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    SelIdx = (int)DgvData.SelectedRows[0].Cells[0].Value;
    SelName = DgvData.SelectedRows[0].Cells[4].Value.ToString();

    this.DialogResult = DialogResult.OK;
    this.Close();
}
}

```

위의 소스만 잘 확인해도 대여점 앱을 만들 수 있습니다. 完