

CHospital.cpp

```
vector<CHospitalPtr> CHospitalDAO::GetListHospital()
{
    vector<CHospitalPtr> resultList;

    //SQL 구문 실행
    CRecordset rs(&m_db);
    rs.Open(CRecordset::forwardOnly, _T("select₩
        a.번호, a.인허가일자, d.영업상태명, a.상세영업상태코드, a.상세영업상태명, a.소재지전화,₩
        a.소재지우편번호, a.소재지전체주소, a.도로명전체주소, a.도로명우편번호, a.사업장명, a.업태구분명,₩
        a.의료기관종별명, a.의료인수, a.입원실수, a.병상수₩
        ,listagg(c.진료과목명, ',')₩
        ,listagg(b.진료과목코드, ',')₩
        from 병원 a, 병원진료과목목록 b, 병원진료과목 c, 병원영업상태 d₩
        where a.번호 = b.번호₩
        and a.영업상태명 = d.영업상태값₩
        and c.진료과목코드 = b.진료과목코드₩
        group by a.번호, a.인허가일자, d.영업상태명, a.상세영업상태코드, a.상세영업상태명, a.소재지전화,₩
        a.소재지우편번호, a.소재지전체주소, a.도로명전체주소, a.도로명우편번호, a.사업장명, a.업태구분명,₩
        a.의료기관종별명, a.의료인수, a.입원실수, a.병상수;"));

    //SQL 구문 실행 결과 얻기
    while (!rs.IsEOF()) {
        //스마트 포인터를 이용하여 객체 생성함
        CHospitalPtr pHospital = make_shared<CHospital>(); //new CHospital;
        if (pHospital == nullptr) return vector<CHospitalPtr>();

        //DB에서 조회된 Hospital 1건을 pHospital 객체로 설정하는 부분
        rs.GetFieldValue(_T("번호"), pHospital->strHospitalNo);
        rs.GetFieldValue(_T("인허가일자"), pHospital->strColDate);
        rs.GetFieldValue((short)2, pHospital->strColState);
        rs.GetFieldValue((short)3, pHospital->strColDetailCode);
        rs.GetFieldValue((short)4, pHospital->strColDetailName);
        rs.GetFieldValue((short)5, pHospital->strColPhone);
        rs.GetFieldValue((short)6, pHospital->nColPost);
    }
}
```

```

rs.GetFieldValue((short)7, pHospital->strColAddr);
rs.GetFieldValue((short)8, pHospital->strColRoadAddr);
rs.GetFieldValue((short)9, pHospital->nColRoadPost);
rs.GetFieldValue((short)10, pHospital->strColName);
rs.GetFieldValue((short)11, pHospital->strColCategory);
rs.GetFieldValue((short)12, pHospital->strColCategoryDetail);
rs.GetFieldValue((short)13, pHospital->nColNumOfPerson);
rs.GetFieldValue((short)14, pHospital->nColNumOfRoom);
rs.GetFieldValue((short)15, pHospital->nColNumOfBed);
rs.GetFieldValue((short)16, pHospital->strColTreatSubName);
rs.GetFieldValue((short)17, pHospital->strColTreatSubCode);

rs.MoveNext();

//배열에 스마트 포인터 객체(pHospital)를 추가한다
resultList.push_back(pHospital);
}
rs.Close();

return resultList;
}

CHospitalPtr CHospitalDAO::GetHospital(CString strHospitalNo)
{
    //스마트 포인터를 이용하여 객체생성함
    CHospitalPtr pHospital = make_shared<CHospital>();

    //SQL 구문 실행
    CRecordset rs(&m_db);
    rs.Open(CRecordset::forwardOnly, _T("select₩
a.번호, 인허가일자, b.영업상태명, 상세영업상태코드, 상세영업상태명,
소재지전화,₩
소재지우편번호, 소재지전체주소, 도로명전체주소, 도로명우편번호, 사업장명,
업태구분명,₩
의료기관종별명, 의료인수, 입원실수, 병상수, c.진료과목코드₩
from 병원 a, 병원영업상태 b,₩
(select e.번호, LISTAGG(g.진료과목코드, ',') 진료과목코드₩
from 병원 e, 병원영업상태 f, 병원진료과목목록 g₩
where e.영업상태명 = f.영업상태값₩

```

```

        and g.번호 = e.번호₩
        group by e.번호) c₩
        where a.영업상태명 = b.영업상태값₩
        and c.번호 = a.번호₩
        and a.번호 = ") + strHospitalNo);

```

//SQL 구문 실행 결과 얻기

```

if (!rs.IsEOF()) {

```

```

        rs.GetFieldValue((short)0, pHospital->strHospitalNo);
        rs.GetFieldValue((short)1, pHospital->strColDate);
        rs.GetFieldValue((short)2, pHospital->strColState);
        rs.GetFieldValue((short)3, pHospital->strColDetailCode);
        rs.GetFieldValue((short)4, pHospital->strColDetailName);
        rs.GetFieldValue((short)5, pHospital->strColPhone);
        rs.GetFieldValue((short)6, pHospital->nColPost);
        rs.GetFieldValue((short)7, pHospital->strColAddr);
        rs.GetFieldValue((short)8, pHospital->strColRoadAddr);
        rs.GetFieldValue((short)9, pHospital->nColRoadPost);
        rs.GetFieldValue((short)10, pHospital->strColName);
        rs.GetFieldValue((short)11, pHospital->strColCategory);
        rs.GetFieldValue((short)12, pHospital->strColCategoryDetail);
        rs.GetFieldValue((short)13, pHospital->nColNumOfPerson);
        rs.GetFieldValue((short)14, pHospital->nColNumOfRoom);
        rs.GetFieldValue((short)15, pHospital->nColNumOfBed);
        rs.GetFieldValue((short)16, pHospital->strColTreatSubName);

```

```

    }

```

```

    rs.Close();

```

```

    return pHospital;

```

```

}

```

```

BOOL CHospitalDAO::UpdateHospital(CHospitalPtr pHospital)

```

```

{

```

```

    //수정할 SQL 구문 생성

```

```

    CString strSQL;

```

```

    strSQL.Format(_T("update 병원 set ₩

```

```

        인허가일자='%s',   영업상태명='%s',   상세영업상태코드='%s',   상세영업상태명

```

```

='%s', 소재지전화='%s', ₩
        소재지우편번호='%s', 소재지전체주소='%s', 도로명전체주소='%s', 도로명우편
번호='%s', 사업장명='%s', ₩
        업태구분명='%s', 의료기관종별명='%s', 의료인수='%s', 입원실수='%s', 병상수
='%s' ₩

        where 번호='%s'"),
        pHospital->strColDate.GetBuffer(),
        pHospital->strColState.GetBuffer(),
        pHospital->strColDetailCode.GetBuffer(),
        pHospital->strColDetailName.GetBuffer(),
        pHospital->strColPhone.GetBuffer(),
        pHospital->nColPost.GetBuffer(),
        pHospital->strColAddr.GetBuffer(),
        pHospital->strColRoadAddr.GetBuffer(),
        pHospital->nColRoadPost.GetBuffer(),
        pHospital->strColName.GetBuffer(),
        pHospital->strColCategory.GetBuffer(),
        pHospital->strColCategoryDetail.GetBuffer(),
        pHospital->nColNumOfPerson.GetBuffer(),
        pHospital->nColNumOfRoom.GetBuffer(),
        pHospital->nColNumOfBed.GetBuffer(),
        pHospital->strHospitalNo.GetBuffer());

    try {
        m_db.BeginTrans();
        m_db.ExecuteNonQuery(strSQL.GetBuffer());
        m_db.CommitTrans();
    }
    catch (const CException* pEx) {
        m_db.Rollback();
        TCHAR szErr[100];
        pEx->GetErrorMessage(szErr, sizeof(szErr));
        AfxMessageBox(szErr);
        return FALSE;
    }
    return TRUE;
}

```

```

BOOL CHospitalDAO::InsertHospital(CHospitalPtr pHospital)

```

```

{
    //등록할 SQL 구문 생성
    CString strSQL;
    strSQL.Format(_T("insert into 병원 (₩
        번호, 인허가일자, 영업상태명, 상세영업상태코드, 상세영업상태명, 소재지전화,
₩
        소재지우편번호, 소재지전체주소, 도로명전체주소, 도로명우편번호, 사업장명,
업태구분명, ₩
        의료기관종별명, 의료인수, 입원실수, ₩
        병상수) values (₩
        '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s',
'%s')"),

        pHospital->strHospitalNo.GetBuffer(),
        pHospital->strColDate.GetBuffer(),
        pHospital->strColState.GetBuffer(),
        pHospital->strColDetailCode.GetBuffer(),
        pHospital->strColDetailName.GetBuffer(),
        pHospital->strColPhone.GetBuffer(),
        pHospital->nColPost.GetBuffer(),
        pHospital->strColAddr.GetBuffer(),
        pHospital->strColRoadAddr.GetBuffer(),
        pHospital->nColRoadPost.GetBuffer(),
        pHospital->strColName.GetBuffer(),
        pHospital->strColCategory.GetBuffer(),
        pHospital->strColCategoryDetail.GetBuffer(),
        pHospital->nColNumOfPerson.GetBuffer(),
        pHospital->nColNumOfRoom.GetBuffer(),
        pHospital->nColNumOfBed.GetBuffer());

    try {
        m_db.BeginTrans();
        m_db.ExecuteSQL(strSQL.GetBuffer());
        m_db.CommitTrans();
    }
    catch (const CException* pEx) {
        m_db.Rollback();
        TCHAR szErr[100];
        pEx->GetErrorMessage(szErr, sizeof(szErr));
        AfxMessageBox(szErr);
    }
}

```

```

        return FALSE;
    }
    return TRUE;
}

vector<CHospitalPtr>  CHospitalDAO::GetListHospitalFind(CString  strHospitalName,  CString
strHospitalPhone)
{
    vector<CHospitalPtr> resultList;

    //검색 SQL 구문 생성
    CString strSQL;

    strSQL.Format(_T("select₩
        a.번호, 인허가일자, b.영업상태명, 상세영업상태코드, 상세영업상태명, 소재지전
화,₩
        소재지우편번호, 소재지전체주소, 도로명전체주소, 도로명우편번호, 사업장명,
업태구분명,₩
        의료기관종별명, 의료인수, 입원실수, 병상수, c.진료과목코드₩
from 병원 a, 병원영업상태 b,₩
(select e.번호, LISTAGG(g.진료과목코드, ',') 진료과목코드₩
from 병원 e, 병원영업상태 f, 병원진료과목목록 g₩
where e.영업상태명 = f.영업상태값₩
and g.번호 = e.번호₩
group by e.번호) c₩
where a.영업상태명 = b.영업상태값₩
and c.번호 = a.번호₩
and a.사업장명 like '%s%%'                                ₩
and a.소재지전화 like '%s%%' ₩
"), strHospitalName.GetBuffer(), strHospitalPhone.GetBuffer());

    CRecordset rs(&m_db);

    rs.Open(CRecordset::forwardOnly, strSQL.GetBuffer());

    //SQL 구문 실행 결과 얻기
    while (!rs.IsEOF()) {
        //스마트 포인터를 이용하여 객체생성함
        CHospitalPtr pHospital = make_shared<CHospital>();
    }
}

```

```

        if (pHospital == nullptr) return vector<CHospitalPtr>();

        rs.GetFieldValue((short)0, pHospital->strHospitalNo);
        rs.GetFieldValue((short)1, pHospital->strColDate);
        rs.GetFieldValue((short)2, pHospital->strColState);
        rs.GetFieldValue((short)3, pHospital->strColDetailCode);
        rs.GetFieldValue((short)4, pHospital->strColDetailName);
        rs.GetFieldValue((short)5, pHospital->strColPhone);
        rs.GetFieldValue((short)6, pHospital->nColPost);
        rs.GetFieldValue((short)7, pHospital->strColAddr);
        rs.GetFieldValue((short)8, pHospital->strColRoadAddr);
        rs.GetFieldValue((short)9, pHospital->nColRoadPost);
        rs.GetFieldValue((short)10, pHospital->strColName);
        rs.GetFieldValue((short)11, pHospital->strColCategory);
        rs.GetFieldValue((short)12, pHospital->strColCategoryDetail);
        rs.GetFieldValue((short)13, pHospital->nColNumOfPerson);
        rs.GetFieldValue((short)14, pHospital->nColNumOfRoom);
        rs.GetFieldValue((short)15, pHospital->nColNumOfBed);
        rs.GetFieldValue((short)16, pHospital->strColTreatSubName);

        rs.MoveNext();

        //배열에 스마트 포인터 객체를 추가한다
        resultList.push_back(pHospital);
    }
    rs.Close();

    return resultList;
}

vector<CHospitalSubjectPtr> CHospitalDAO::GetListHospitalSubject()
{
    vector<CHospitalSubjectPtr> resultList;

    //2. SQL 구문 실행
    CRecordset rs(&m_db);
    rs.Open(CRecordset::forwardOnly, _T("select * from 병원진료과목"));

    //3. SQL 구문 실행 결과 얻기

```

```

while (!rs.IsEOF()) {
    //스마트 포인터를 이용하여 객체생성함
    CHospitalSubjectPtr pHospitalSubject = make_shared<CHospitalSubject>();
    if (pHospitalSubject == nullptr) return vector<CHospitalSubjectPtr>();

    rs.GetFieldValue((short)0, pHospitalSubject->strSubjectCode);
    rs.GetFieldValue((short)1, pHospitalSubject->strSubjectName);

    rs.MoveNext();

    //배열에 스마트 포인터 객체를 추가한다
    resultList.push_back(pHospitalSubject);
}
rs.Close();

return resultList;
}

```

```

BOOL CHospitalDAO::InsertTreatList(CString strHospitalNo, CString pstrState)

```

```

{
    //등록할 SQL 구문 만든다
    CString strSQL;
    strSQL.Format(_T("insert into 병원진료과목목록 (₩
        번호, ₩
        진료과목코드 ₩
    ) values ( ₩
        '%s', %s)"), strHospitalNo, pstrState);

    try {
        m_db.BeginTrans();
        m_db.ExecuteSQL(strSQL.GetBuffer());
        m_db.CommitTrans();
    }
    catch (const CException* pEx) {
        m_db.Rollback();
        TCHAR szErr[100];
        pEx->GetErrorMessage(szErr, sizeof(szErr));
        AfxMessageBox(szErr);
        return FALSE;
    }
}

```



```

    }

    return TRUE;
}

BOOL CHospitalDAO::DeleteTreatList(CString strHospitalNo, CString pstrState)
{
    //삭제할 SQL 구문 만든다
    CString strSQL;
    strSQL.Format(_T("delete from 병원진료과목목록 ₩
        where 번호 = '%s' ₩
        and 진료과목코드 = '%s'"),strHospitalNo, pstrState);

    try {
        m_db.BeginTrans();
        m_db.ExecuteSQL(strSQL.GetBuffer());
        m_db.CommitTrans();
    }
    catch (const CException* pEx) {
        m_db.Rollback();
        TCHAR szErr[100];
        pEx->GetErrorMessage(szErr, sizeof(szErr));
        AfxMessageBox(szErr);
        return FALSE;
    }

    return TRUE;
}

```

```

BOOL CHospitalDAO::DeleteAllTreatList(CString strHospitalNo)
{
    //삭제할 SQL 구문 만든다
    CString strSQL;
    strSQL.Format(_T("delete from 병원진료과목목록 ₩
        where 번호 = '%s'"), strHospitalNo);

    try {
        m_db.BeginTrans();
    }
}

```

```

        m_db.ExecuteSQL(strSQL.GetBuffer());
        m_db.CommitTrans();
    }
    catch (const CException* pEx) {
        m_db.Rollback();
        TCHAR szErr[100];
        pEx->GetErrorMessage(szErr, sizeof(szErr));
        AfxMessageBox(szErr);
        return FALSE;
    }

    return TRUE;
}

```

CHospitalDialog.cpp

```

CHospitalDialog::CHospitalDialog(vector<CHospitalPtr>& hospitalList,
vector<CMedicalClassPtr>& medicalclassList, vector<CHospitalStatePtr>& hospitalStateList,
CHospitalPtr pHospital, vector<CHospitalSubjectPtr>& hospitalSubjectList, CDatabase& db,
CWnd* pParent /*=nullptr*/)
    : CDialogEx(IDD_HOSPITAL_UPDATE, pParent)
    , m_hospitalList(hospitalList)
    , m_medicalclassList(medicalclassList)
    , m_hospitalStateList(hospitalStateList)
    , m_pHospital(pHospital)
    , m_hospitalSubjectList(hospitalSubjectList)
    , m_db(db)
    , m_strHospitalNo(_T(""))
    , m_strColDate(_T(""))
    , m_strColPhone(_T(""))
    , m_nColPost(0)
    , m_strColAddr(_T(""))
    , m_strColRoadAddr(_T(""))
    , m_nColRoadPost(0)
    , m_strColName(_T(""))
    , m_nColNumOfPerson(0)
    , m_nColNumOfRoom(0)
    , m_nColNumOfBed(0)
    , m_strColState(_T(""))
    , m_strColDetailCode(_T(""))

```

```

        , m_strColDetailName(_T(""))
        , m_strColCategory(_T(""))
        , m_strColCategoryDetail(_T(""))
        , m_strColSubject(_T(""))
    {
    }

    BOOL CHospitalDialog::OnInitDialog()
    {
        CDialogEx::OnInitDialog();

        if (m_pHospital->strHospitalNo.IsEmpty()) {
            ((CEdit*)GetDlgItem(IDC_EDIT_NO))->SetReadOnly(FALSE);
            GetDlgItem(IDC_BUTTON_OVERLAP)->ShowWindow(SW_SHOW);
        }

        m_treatListView.InsertColumn(0, _T("번호"), LVCFMT_LEFT, 100);
        m_treatListView.InsertColumn(1, _T("진료과목코드"), LVCFMT_LEFT, 100);

        DWORD dwExStyle = m_treatListView.GetExtendedStyle();
        m_treatListView.SetExtendedStyle(dwExStyle | LVS_EX_CHECKBOXES |
LVS_EX_BORDERSELECT | LVS_EX_GRIDLINES | LVS_EX_FULLROWSELECT);

        CMedicalClassDAO medicalClassDAO(m_db);
        GetDBAllMedicdalView(medicalClassDAO.GetListMedicalClass());

        m_strHospitalNo = m_pHospital->strHospitalNo;
        m_strColDate = m_pHospital->strColDate;
        m_strColPhone = m_pHospital->strColPhone;
        m_nColPost = _ttoi(m_pHospital->nColPost.GetBuffer());
        m_strColAddr = m_pHospital->strColAddr;
        m_strColRoadAddr = m_pHospital->strColRoadAddr;
        m_nColRoadPost = _ttoi(m_pHospital->nColRoadPost.GetBuffer());
        m_strColName = m_pHospital->strColName;
        m_nColNumOfPerson = _ttoi(m_pHospital->nColNumOfPerson.GetBuffer());
        m_nColNumOfRoom = _ttoi(m_pHospital->nColNumOfRoom.GetBuffer());
        m_nColNumOfBed = _ttoi(m_pHospital->nColNumOfBed.GetBuffer());
    }

```

```

m_strColState = m_pHospital->strColState;
m_strColDetailCode = m_pHospital->strColDetailCode;
m_strColDetailName = m_pHospital->strColDetailName;
m_strColCategory = m_pHospital->strColCategory;
m_strColCategoryDetail = m_pHospital->strColCategoryDetail;
m_strColSubject = m_pHospital->strColTreatSubName;

vector<CString> arr;

LPTSTR lpszCol16 = m_pHospital->strColTreatSubName.GetBuffer();
LPTSTR lpszToken = NULL;
LPTSTR lpszNextToken = NULL;
lpszToken = _tcstok_s(lpszCol16, _T(","), &lpszNextToken);
while (lpszToken) {
    arr.push_back(lpszToken);
    lpszToken = _tcstok_s(NULL, _T(","), &lpszNextToken);
}

const int nCount = m_treatlistView.GetItemCount();
m_nCount = m_treatlistView.GetItemCount();

for (const auto& parr : arr) {
    for (int i = 0; i < nCount; i++) {
        if (parr == m_medicalclassList[i]->strCode) {
            m_treatlistView.SetCheck(i);
        }
    }
}

int nIndex = 0;
for (const auto& pCHospitalState : m_hospitalStateList) {
    m_comboColState.InsertString(nIndex++, pCHospitalState->strName);
}
nIndex = 0;
for (const auto& pCHospital : m_hospitalList) {
    int index = m_comboColDetailCode.FindStringExact(-1, pCHospital-
>strColDetailCode);
    if (index == LB_ERR) {
        m_comboColDetailCode.InsertString(nIndex++, pCHospital-

```

```

>strColDetailCode);
        arrDetailCode.push_back(pCHospital->strColDetailCode);
    }
}
nIndex = 0;
for (const auto& pCHospital : m_hospitalList) {
    int index = m_comboColDetailName.FindStringExact(-1, pCHospital-
>strColDetailName);
    if (index == LB_ERR) {
        m_comboColDetailName.InsertString(nIndex++, pCHospital-
>strColDetailName);
        arrDetailName.push_back(pCHospital->strColDetailName);
    }
}
nIndex = 0;
for (const auto& pCHospital : m_hospitalList) {
    int index = m_comboColCategory.FindStringExact(-1, pCHospital-
>strColCategory);
    if (index == LB_ERR) {
        m_comboColCategory.InsertString(nIndex++, pCHospital-
>strColCategory);
        arrCategory.push_back(pCHospital->strColCategory);
    }
}
nIndex = 0;
for (const auto& pCHospital : m_hospitalList) {
    int index = m_comboColCategoryDetail.FindStringExact(-1, pCHospital-
>strColCategoryDetail);
    if (index == LB_ERR && pCHospital->strColCategoryDetail != "") {
        m_comboColCategoryDetail.InsertString(nIndex++, pCHospital-
>strColCategoryDetail);
        //AfxMessageBox(pCHospital->strColCategoryDetail);
        arrCategoryDetail.push_back(pCHospital->strColCategoryDetail);
    }
}

UpdateData(FALSE);

return TRUE; // return TRUE unless you set the focus to a control

```

```

// 예외: OCX 속성 페이지는 FALSE를 반환해야 합니다.
}

void CHospitalDialog::OnBnClickedButtonOverlap()
{
    //사번을 읽어들인다.
    GetDlgItemText(IDC_EDIT_NO, m_strHospitalNo);
    if (m_strHospitalNo.IsEmpty()) {
        AfxMessageBox(_T("사번을 입력해주세요"));
        GetDlgItem(IDC_EDIT_NO)->SetFocus();
        return;
    }

    CMainFrame* pMainFrame = (CMainFrame*)AfxGetMainWnd();
    CProjectHospitalView* pView = (CProjectHospitalView*)pMainFrame->GetActiveView();

    CHospitalDAO hospitalDAO(pView->m_db);
    CHospitalPtr pHospital = hospitalDAO.GetHospital(m_strHospitalNo);
    if (pHospital->strHospitalNo == m_strHospitalNo) {
        AfxMessageBox(m_strHospitalNo + _T(" 사번은 중복 되었습니다"));
        SetDlgItemText(IDC_EDIT_NO, _T(""));
        GetDlgItem(IDC_EDIT_NO)->SetFocus();
        return;
    }
    else {
        AfxMessageBox(_T("사용하실 수 있습니다"));
    }
}

void CHospitalDialog::OnCbnSelendokComboState()
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    int nCurSel = m_comboColState.GetCurSel();
    if (nCurSel >= 0 && nCurSel < m_hospitalStateList.size()) {
        const CHospitalStatePtr pHospitalState = m_hospitalStateList[nCurSel];
        m_strColState = pHospitalState->strCode;
    }
}

```

```

void CHospitalDialog::OnCbnSelendokComboDetailCode()
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    int nCurSel = m_comboColDetailCode.GetCurSel();
    if (nCurSel >= 0 && nCurSel < m_hospitalStateList.size()) {
        m_pHospital->strColDetailCode = arrDetailCode[nCurSel];
    }
}

void CHospitalDialog::OnCbnSelendokComboDetailName()
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    int nCurSel = m_comboColDetailName.GetCurSel();
    if (nCurSel >= 0 && nCurSel < m_hospitalStateList.size()) {
        m_pHospital->strColDetailName = arrDetailName[nCurSel];
    }
}

void CHospitalDialog::OnCbnSelendokComboCategory()
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    int nCurSel = m_comboColCategory.GetCurSel();
    if (nCurSel >= 0 && nCurSel < m_hospitalStateList.size()) {
        m_pHospital->strColCategory = arrCategory[nCurSel];
    }
}

void CHospitalDialog::OnCbnSelendokComboCategoryDetail()
{
    int nCurSel = m_comboColCategoryDetail.GetCurSel();
    if (nCurSel >= 0 && nCurSel < m_hospitalStateList.size()) {
        m_pHospital->strColCategoryDetail = arrCategoryDetail[nCurSel];
    }
}

```

```

void CHospitalDialog::OnOK()
{
    // TODO: 여기에 특수화된 코드를 추가 및/또는 기본 클래스를 호출합니다.
    for (int i = m_nCount - 1; i >= 0; --i) {
        if (m_treatlistView.GetCheck(i)) {
            checklist.push_back(i);
        }
    }

    int nCurSel = m_comboColState.GetCurSel();
    if (nCurSel >= 0 && nCurSel < m_hospitalStateList.size()) {
        const CHospitalStatePtr pHospitalState = m_hospitalStateList[nCurSel];
        m_pHospital->strColState = pHospitalState->strCode;
    }

    CDialogEx::OnOK();
}

```

CHospitalFindDlg.cpp

```

void CHospitalFindDlg::OnBnClickedOk()
{
    UpdateData(TRUE);
    if (m_strHospitalName.IsEmpty() && m_strHospitalPhone.IsEmpty()) {
        AfxMessageBox(_T("병원명 또는 전화번호를 입력해주세요"));
        GetDlgItem(IDC_EDIT_FIND_NAME)->SetFocus();
        return;
    }
    CDialogEx::OnOK();
}

```

CHospitalState.cpp

```

vector<CHospitalStatePtr> CHospitalStateDAO::GetListHospitalState()
{
    vector<CHospitalStatePtr> resultList;

    CRecordset rs(&m_db);
    rs.Open(CRecordset::forwardOnly, _T("select * from 병원영업상태"));
}

```



```

        while (!rs.IsEOF()) {
            CHospitalStatePtr pHospitalState = make_shared<CHospitalState>(); //new
CHospitalState;
            if (pHospitalState == nullptr) return vector<CHospitalStatePtr>();

            rs.GetFieldValue((short)0, pHospitalState->strCode);
            rs.GetFieldValue((short)1, pHospitalState->strName);

            rs.MoveNext();

            resultList.push_back(pHospitalState);
        }
        rs.Close();

        return resultList;
}

```

```

map<CString, CString> CHospitalStateDAO::GetMapHospitalState()
{
    map<CString, CString> resultMap;

    CRecordset rs(&m_db);
    rs.Open(CRecordset::forwardOnly, _T("select * from 병원영업상태"));

    CString strCode;
    CString strName;
    while (!rs.IsEOF()) {
        rs.GetFieldValue((short)0, strCode);
        rs.GetFieldValue((short)1, strName);

        rs.MoveNext();

        resultMap[strName] = strCode;
    }
    rs.Close();

    return resultMap;
}

```

```
}
```

CMedicalClass.cpp

```
vector<CMedicalClassPtr> CMedicalClassDAO::GetListMedicalClass()
{
    vector<CMedicalClassPtr> resultList;

    CRecordset rs(&m_db);
    rs.Open(CRecordset::forwardOnly, _T("select * from 병원진료과목"));

    while (!rs.IsEOF()) {
        CMedicalClassPtr pMedicalClass = make_shared<CMedicalClass>(); //new
CMedicalClass;
        if (pMedicalClass == nullptr) return vector<CMedicalClassPtr>();

        rs.GetFieldValue((short)0, pMedicalClass->strCode);
        rs.GetFieldValue((short)1, pMedicalClass->strName);

        rs.MoveNext();

        resultList.push_back(pMedicalClass);
    }
    rs.Close();

    return resultList;
}
```

ProjectHospitalView.cpp

```
void CProjectHospitalView::SetHospitalView(int nRow, const CHospitalPtr pHospital)
{
    m_listView.SetItemText(nRow, 1, pHospital->strColDate);
    m_listView.SetItemText(nRow, 2, pHospital->strColState);
    m_listView.SetItemText(nRow, 3, pHospital->strColDetailCode);
    m_listView.SetItemText(nRow, 4, pHospital->strColDetailName);
    m_listView.SetItemText(nRow, 5, pHospital->strColPhone);
    m_listView.SetItemText(nRow, 6, pHospital->nColPost);
    m_listView.SetItemText(nRow, 7, pHospital->strColAddr);
    m_listView.SetItemText(nRow, 8, pHospital->strColRoadAddr);
}
```

```

        m_listView.SetItemText(nRow, 9, pHospital->nColRoadPost);
        m_listView.SetItemText(nRow, 10, pHospital->strColName);
        m_listView.SetItemText(nRow, 11, pHospital->strColCategory);
        m_listView.SetItemText(nRow, 12, pHospital->strColCategoryDetail);
        m_listView.SetItemText(nRow, 13, pHospital->nColNumOfPerson);
        m_listView.SetItemText(nRow, 14, pHospital->nColNumOfRoom);
        m_listView.SetItemText(nRow, 15, pHospital->nColNumOfBed);
        m_listView.SetItemText(nRow, 16, pHospital->strColTreatSubName);
    }

void CProjectHospitalView::GetDBAllHospitalView(vector<CHospitalPtr> hospitalList)
{
    m_listView.DeleteAllItems();

    int nRow = 0;
    for (const auto& pHospital : hospitalList) {
        m_listView.InsertItem(nRow, pHospital->strHospitalNo, 0);
        SetHospitalView(nRow, pHospital);
        nRow++;
    }
}

void CProjectHospitalView::OnInitialUpdate()
{
    CFormView::OnInitialUpdate();
    GetParentFrame()->RecalcLayout();
    ResizeParentToFit();

    CFormView::OnInitialUpdate();
    GetParentFrame()->RecalcLayout();
    ResizeParentToFit();

    m_imageList.Create(48, 48, ILC_COLOR32, 5, 5);
    m_imageListSmall.Create(16, 16, ILC_COLOR32, 5, 5);

    m_imageList.Add(AfxGetApp()->LoadIcon(IDR_MAINFRAME));
    m_imageListSmall.Add(AfxGetApp()->LoadIcon(IDR_MAINFRAME));

    m_listView.SetImageList(&m_imageList, LVSIL_NORMAL);

```

```

m_listView.SetImageList(&m_imageListSmall, LVSIL_SMALL);

//컬럼 정보 출력
m_listView.InsertColumn(0, _T("번호"), LVCFMT_LEFT, 100);
m_listView.InsertColumn(1, _T("인허가일자"), LVCFMT_LEFT, 100);
m_listView.InsertColumn(2, _T("영업상태명"), LVCFMT_LEFT, 100);
m_listView.InsertColumn(3, _T("상세영업상태코드"), LVCFMT_LEFT, 100);
m_listView.InsertColumn(4, _T("상세영업상태명"), LVCFMT_LEFT, 100);
m_listView.InsertColumn(5, _T("소재지전화"), LVCFMT_LEFT, 100);
m_listView.InsertColumn(6, _T("소재지우편번호"), LVCFMT_LEFT, 100);
m_listView.InsertColumn(7, _T("소재지전체주소"), LVCFMT_LEFT, 100);
m_listView.InsertColumn(8, _T("도로명전체주소"), LVCFMT_LEFT, 100);
m_listView.InsertColumn(9, _T("도로명우편번호"), LVCFMT_LEFT, 100);
m_listView.InsertColumn(10, _T("사업장명"), LVCFMT_LEFT, 100);
m_listView.InsertColumn(11, _T("업태구분명"), LVCFMT_LEFT, 100);
m_listView.InsertColumn(12, _T("의료기관종별명"), LVCFMT_LEFT, 100);
m_listView.InsertColumn(13, _T("의료인수"), LVCFMT_LEFT, 100);
m_listView.InsertColumn(14, _T("입원실수"), LVCFMT_LEFT, 100);
m_listView.InsertColumn(15, _T("병상수"), LVCFMT_LEFT, 100);
m_listView.InsertColumn(16, _T("진료과목내용명"), LVCFMT_LEFT, 300);

DWORD dwExStyle = m_listView.GetExtendedStyle();
m_listView.SetExtendedStyle(dwExStyle | LVS_EX_CHECKBOXES | LVS_EX_BORDERSELECT
| LVS_EX_GRIDLINES | LVS_EX_FULLROWSELECT);

CHospitalDAO hospitalDAO(m_db);
GetDBAllHospitalView(hospitalDAO.GetListHospital());
}

//삭제 버튼
void CProjectHospitalView::OnBnClickedButtonDelete()
{
    if (IDYES != AfxMessageBox(_T("삭제하시겠습니까?"), MB_YESNO))
    {
        return;
    }

    const int nCount = m_listView.GetItemCount();
    CString strHospitalNo;

```

```

CString strSQL;

CString strSqlParam;
CArray<int, int> arr;

for (int i = nCount - 1; i >= 0; --i) {
    if (m_listView.GetCheck(i)) {
        //삭제할 사원 번호를 얻는다
        strSqlHospitalNo = m_listView.GetItemText(i, 0);

        strSqlParam += strSqlHospitalNo + _T(",");

        //삭제할 사원의 위치를 배열에 추가
        arr.Add(i);
    }
}

try {
    if (!strSqlParameter.IsEmpty()) {
        //() 삭제
        strSqlParam.Delete(strSqlParameter.GetLength() - 1, 1);
        //SQL 삭제 구문을 생성한다
        strSQL = _T("delete from 병원 where 번호 in (") + strSqlParam +
        _T(")");

        //트랜잭션
        m_db.BeginTrans();
        m_db.ExecuteSQL(strSQL);//DML 구문 (insert, delete ,update, merge)
        m_db.CommitTrans();
        //리스트에서 삭제 항목을 제거
        for (int i = 0; i < arr.GetSize(); i++) {
            //목록에서 제거한다
            m_listView.DeleteItem(arr.GetAt(i));
        }
        AfxMessageBox(_T("삭제가 완료되었습니다.));
    }
}

catch (const CException* p) {
    //트랜잭션 복구 //DML

```

```

        m_db.Rollback();
        TCHAR szErr[500];
        p->GetErrorMessage(szErr, sizeof(szErr));
        AfxMessageBox(szErr);
    }
}

//수정 버튼
void CProjectHospitalView::OnBnClickedButtonChange()
{
    const int nCount = m_listView.GetItemCount();
    CString strHospitalNo;

    //선택된 사번을 얻는다
    int nRow;
    for (int i = nCount - 1; i >= 0; --i) {
        if (m_listView.GetCheck(i)) {
            //선택된 사원 번호를 얻는다
            strHospitalNo = m_listView.GetItemText(i, 0);
            nRow = i;
            break;
        }
    }
    if (strHospitalNo.IsEmpty()) {
        AfxMessageBox(_T("사원을 선택해 주세요"));
        return;
    }

    //상세코드, 상세영업상태명, 업태구분명, 의료기관정별명을 얻기 위해 병원 리스트를
전달
    CHospitalDAO hospitalDAO(m_db);
    vector<CHospitalPtr> hospitalList = hospitalDAO.GetListHospital();
    //진료과목명을 얻기 위해 병원진료과목 리스트를 전달
    CMedicalClassDAO medicalClassDAO(m_db);
    vector<CMedicalClassPtr> medicalclassList = medicalClassDAO.GetListMedicalClass();
    //영업상태명을 얻기 위해 병원영업상태 리스트를 전달
    CHospitalStateDAO hospitalStateDAO(m_db);
    vector<CHospitalStatePtr> hospitalStateList = hospitalStateDAO.GetListHospitalState();
    //선택한 병원을 전달

```

```

CHospitalPtr pHospital = hospitalDAO.GetHospital(strHospitalNo);
//해당병원의 진료과목내용명을 얻기위해 병원진료과목목록을 전달
CHospitalDAO hospitalSubjectDAO(m_db);
vector<CHospitalSubjectPtr> hospitalSubjectList =
hospitalSubjectDAO.GetListHospitalSubject();
//대화상자를 출력
CHospitalDialog dlg(hospitalList, medicalclassList, hospitalStateList, pHospital,
hospitalSubjectList, m_db);

if (dlg.DoModal() == IDOK) {
    pHospital->strColDate = dlg.m_strColDate;
    pHospital->strColPhone = dlg.m_strColPhone;
    pHospital->strColAddr = dlg.m_strColAddr;
    pHospital->strColRoadAddr = dlg.m_strColRoadAddr;
    pHospital->strColName = dlg.m_strColName;
    pHospital->strColTreatSubName = dlg.m_strColSubject;

    pHospital->nColPost.Format(_T("%d"), dlg.m_nColPost);
    pHospital->nColRoadPost.Format(_T("%d"), dlg.m_nColRoadPost);
    pHospital->nColNumOfPerson.Format(_T("%d"), dlg.m_nColNumOfPerson);
    pHospital->nColNumOfRoom.Format(_T("%d"), dlg.m_nColNumOfRoom);
    pHospital->nColNumOfBed.Format(_T("%d"), dlg.m_nColNumOfBed);

    CHospitalDAO hospitalDAO(m_db);

    if (hospitalDAO.UpdateHospital(pHospital)) {
        SetHospitalView(nRow, pHospital);
    }

    //목록 체크한거 db및 뷰에 적용
    hospitalDAO.DeleteAllTreatList(dlg.m_strHospitalNo);
    pHospital->strColTreatSubName = "";
    const int nCount = dlg.m_nCount;
    for (const int& pcheckboxlist : dlg.checkboxlist) {
        hospitalDAO.InsertTreatList(dlg.m_strHospitalNo,
dlg.m_medicalclassList[pcheckboxlist]->strCode);
        pHospital->strColTreatSubName = pHospital->strColTreatSubName +
"," + dlg.m_medicalclassList[pcheckboxlist]->strName;
    }
}

```

```

        pHospital->strColTreatSubName.Delete(0, 1);

        SetHospitalView(nRow, pHospital);

        AfxMessageBox(_T("수정이 완료되었습니다"));
    }
}

//등록 버튼
void CProjectHospitalView::OnBnClickedButtonInsert()
{
    CHospitalDAO hospitalDAO(m_db);
    vector<CHospitalPtr> hospitalList = hospitalDAO.GetListHospital();
    CMedicalClassDAO medicalClassDAO(m_db);
    vector<CMedicalClassPtr> medicalclassList = medicalClassDAO.GetListMedicalClass();
    CHospitalStateDAO hospitalStateDAO(m_db);
    vector<CHospitalStatePtr> hospitalStateList = hospitalStateDAO.GetListHospitalState();
    CHospitalDAO hospitalSubjectDAO(m_db);
    vector<CHospitalSubjectPtr> hospitalSubjectList =
hospitalSubjectDAO.GetListHospitalSubject();
    CHospitalPtr pHospital = make_shared<CHospital>();
    CHospitalDialog dlg(hospitalList, medicalclassList, hospitalStateList, pHospital,
hospitalSubjectList, m_db);

    int nRow;

    if (dlg.DoModal() == IDOK) {
        //SQL 삽입
        pHospital->strHospitalNo = dlg.m_strHospitalNo;
        pHospital->strColDate = dlg.m_strColDate;
        pHospital->strColPhone = dlg.m_strColPhone;
        pHospital->strColAddr = dlg.m_strColAddr;
        pHospital->strColRoadAddr = dlg.m_strColRoadAddr;
        pHospital->strColName = dlg.m_strColName;

        pHospital->nColPost.Format(_T("%d"), dlg.m_nColPost);
        pHospital->nColRoadPost.Format(_T("%d"), dlg.m_nColRoadPost);
        pHospital->nColNumOfPerson.Format(_T("%d"), dlg.m_nColNumOfPerson);
        pHospital->nColNumOfRoom.Format(_T("%d"), dlg.m_nColNumOfRoom);
    }
}

```



```

        pHospital->nColNumOfBed.Format(_T("%d"), dlg.m_nColNumOfBed);

        CHospitalDAO hospitalDAO(m_db);
        if (hospitalDAO.InsertHospital(pHospital)) {
            nRow = m_listView.GetItemCount();
            m_listView.InsertItem(nRow, pHospital->strHospitalNo, 0);
            SetHospitalView(nRow, pHospital);
        }

        hospitalDAO.DeleteAllTreatList(dlg.m_strHospitalNo);
        pHospital->strColTreatSubName = "";
        const int nCount = dlg.m_nCount;
        for (const int& pcheckboxlist : dlg.checkboxlist) {
            hospitalDAO.InsertTreatList(dlg.m_strHospitalNo,
dlg.m_medicalclassList[pcheckboxlist]->strCode);
            pHospital->strColTreatSubName = pHospital->strColTreatSubName +
            "," + dlg.m_medicalclassList[pcheckboxlist]->strName;
        }
        pHospital->strColTreatSubName.Delete(0, 1);

        SetHospitalView(nRow, pHospital);

        AfxMessageBox(_T("등록이 완료되었습니다"));
    }
}

//검색 버튼
void CProjectHospitalView::OnBnClickedButtonFind()
{
    CHospitalFindDlg dlg;
    if (dlg.DoModal() == IDOK) {
        CHospitalDAO hospitalDAO(m_db);
        GetDBAllHospitalView(hospitalDAO.GetListHospitalFind(dlg.m_strHospitalName,
dlg.m_strHospitalPhone));
    }
}

```