イースト
est.co.jp

専用アプリケーションとの違い

管理工数の削減と業務の簡素化

イースト株式会社

独自に専用アプリケーションを開発する場合

- 開発ツール（Visual Studio等）を使用した自由な構築が行なえます
- 業務に即したきめ細かな処理を組み込めます

だけど・・・

- プログラミングのスキルが必要
- データベースに関しての知識が必要
- データベースを使用する部分だけで、大量の処理記述を必要とする
- 当然実際に行ないたい業務処理部分の記述も必要となる
- データベースの変更・改修時に手間が掛かる
  （１から作り直しの場合もある）
- 業務処理の追加・変更時の改修に手間と工数が必要

データベース接続VB.NETコーディング例(1)

★クラス(OdbcDbIf.vb)

```
Imports System.Data.Odbc

Public Class OdbcDataBaseIf
    ' Connection
    Private _con As OdbcConnection = Nothing
    ' Transaction Object
    Private _trn As OdbcTransaction = Nothing
    '-------------
    ' DB接続 -
    '-------------
    ' データソース名
    ' データベース名
    ' ユーザーID
    ' パスワード
    ' タイムアウト値
    '
    Public Sub Connect(Optional ByVal dsn As String = "Data Source", _
                       Optional ByVal dbn As String = "Data Base Name", _
                       Optional ByVal uid As String = "User Name", _
                       Optional ByVal pas As String = "Password", _
                       Optional ByVal tot As Integer = -1)

        Try
            If _con Is Nothing Then
                _con = New OdbcConnection
            End If

            Dim cst As String = ""
            cst = cst & ";DSN=" & dsn
            cst = cst & ";Database=" & dbn
            cst = cst & ";UID=" & uid
            cst = cst & ";PWD=" & pas
            If tot > -1 Then
                '_con.ConnectionTimeout = tot
                cst = cst & ";Connect Timeout=" & tot.ToString
            End If
            _con.ConnectionString = cst

            _con.Open()
        Catch ex As Exception
            Throw New Exception("Connect Error", ex)
        End Try
    End Sub

    '-------------
    ' DB切断 -
    '-------------
    Public Sub Disconnect()
        Try
            _con.Close()
        Catch ex As Exception
            Throw New Exception("Disconnect Error", ex)
        End Try
    End Sub

    '----------------
    ' SQLの実行 -
    '----------------
    ' SQL文
    ' タイムアウト値
    '
    Public Function ExecSql(ByVal sql As String, _
                            Optional ByVal tot As Integer = -1) As DataTable
        Dim dt As New DataTable

        Try
            Dim sqlCommand As New OdbcCommand(sql, _con, _trn)

            If tot > -1 Then
                sqlCommand.CommandTimeout = tot
            End If

            Dim adapter As New OdbcDataAdapter(sqlCommand)

            adapter.Fill(dt)
            adapter.Dispose()
            sqlCommand.Dispose()
        Catch ex As Exception
            Throw New Exception("ExecuteSql Error", ex)
        End Try

        Return dt
    End Function
```

データベース接続VB.NETコーディング例(2)

```
    '-------------------------
    ' トランザクション開始 -
    '-------------------------
    Public Sub BeginTransaction()
        Try
            _trn = _con.BeginTransaction()
        Catch ex As Exception
            Throw New Exception("BeginTransaction Error", ex)
        End Try
    End Sub

    '-----------
    ' コミット -
    '-----------
    Public Sub CommitTransaction()
        Try
            If _trn Is Nothing = False Then
                _trn.Commit()
            End If
        Catch ex As Exception
            Throw New Exception("CommitTransaction Error", ex)
        Finally
            _trn = Nothing
        End Try
    End Sub

    '------------------
    ' ロールバック -
    '------------------
    Public Sub RollbackTransaction()
        Try
            If _trn Is Nothing = False Then
                _trn.Rollback()
            End If
        Catch ex As Exception
            Throw New Exception("RollbackTransaction Error", ex)
        Finally
            _trn = Nothing
        End Try
    End Sub
```

★フォームモジュール(Form1.vb)

```
Public Class Form1
    Inherits System.Windows.Forms.Form

#Region " Windows フォーム デザイナで生成されたコード "
' 省略
' 実行用ボタンを2つ追加
#End Region
    '-----------------
    ' ボタン処理 -
    '-----------------
    Private Sub Button1_Click( _
        ByVal sender As System.Object, ByVal e As System.EventArgs) _
            Handles Button1.Click
        Dim db  As New OdbcDataBaseIf
        Dim tb  As DataTable
        Dim i,j As Integer
        Dim str As String

        db.Connect()
        tb = db.ExecSql("select col1,col2,col3 from test")
        For i = 0 To tb.Rows.Count - 1
            For j = 0 To tb.Rows(i).ItemArray.Length -1
                str = str & CStr(tb.Rows(i).ItemArray(j))
                str = str & " : "
            Next
        Next
        MessageBox.Show(str)
        db.Disconnect()
    End Sub
    '-----------------
    ' ボタン処理 -
    '-----------------
    Private Sub Button2_Click( _
        ByVal sender As System.Object, ByVal e As System.EventArgs) _
            Handles Button2.Click
        Dim db As New OdbcDataBaseIf
        db.Connect()
        db.BeginTransaction()
        db.ExecSql( _
            "select name, zip, address, telno from employee")
        db.CommitTransaction()
        db.Disconnect()
    End Sub
End Class
```

- これらの処理は、単にデータベースとの
  接続部分を記述しただけです！

- 実際に欲しい業務処理は、
  この後大量に記述する必要があります！

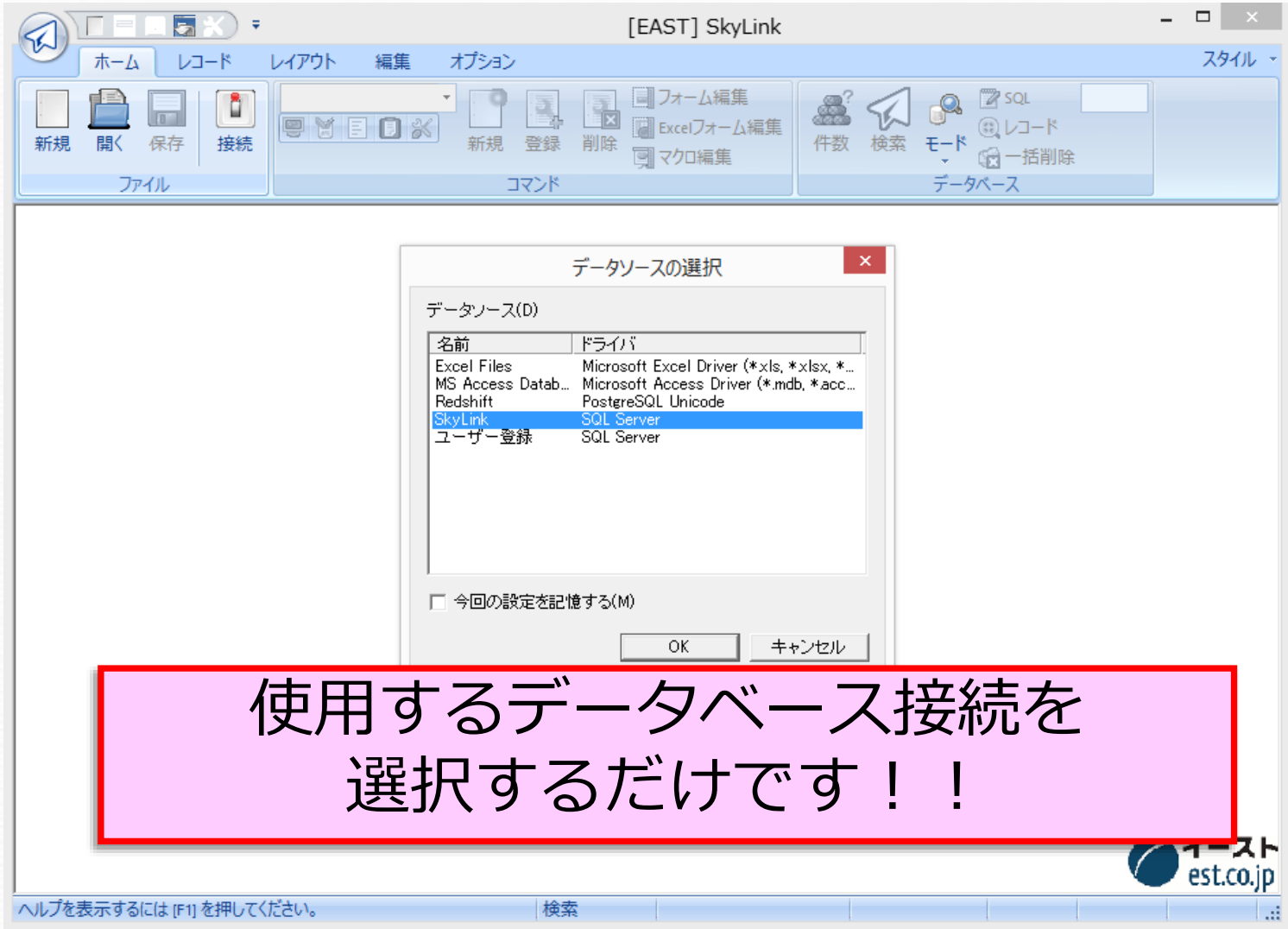
- 一度作成したものを、
  変更する場合には再び作成する必要があります！

SkyLinkを導入した場合

- 直感的な操作で使用可能
- 使用中のデータベースに接続するだけで使えます
  （DBミドルウェアを使用します）
- 抽出の条件設定も簡単です
- 業務で必要な項目は自由検索で選択するだけです
- データベースが切り替わっても各種データベースに対応しています
- ユーザーが自分で設定できるので改管理者の手間が掛かりません

SkyLinkなら・・・

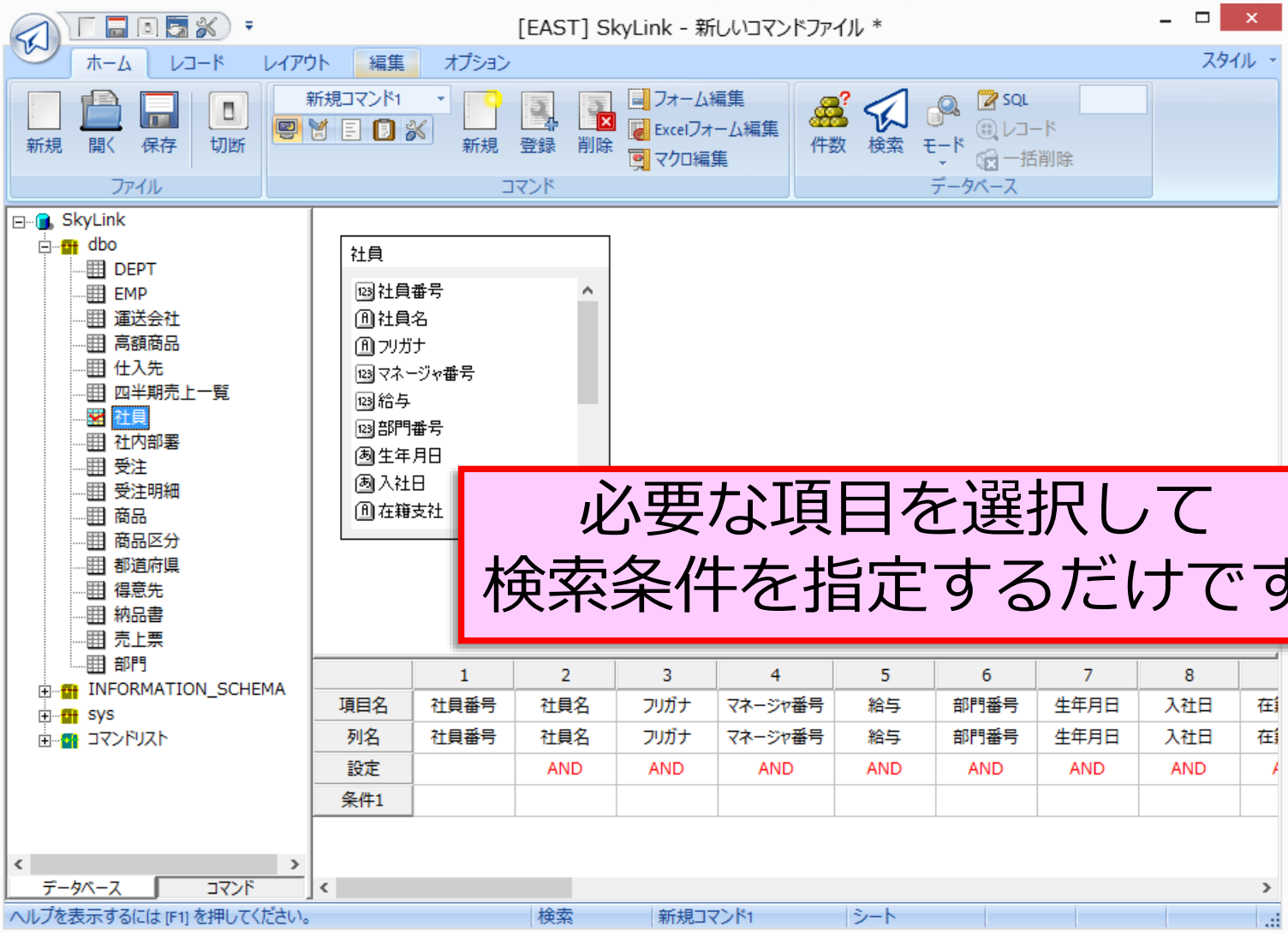
データベース接続



使用するデータベース接続を
選択するだけです！！

SkyLinkなら・・・

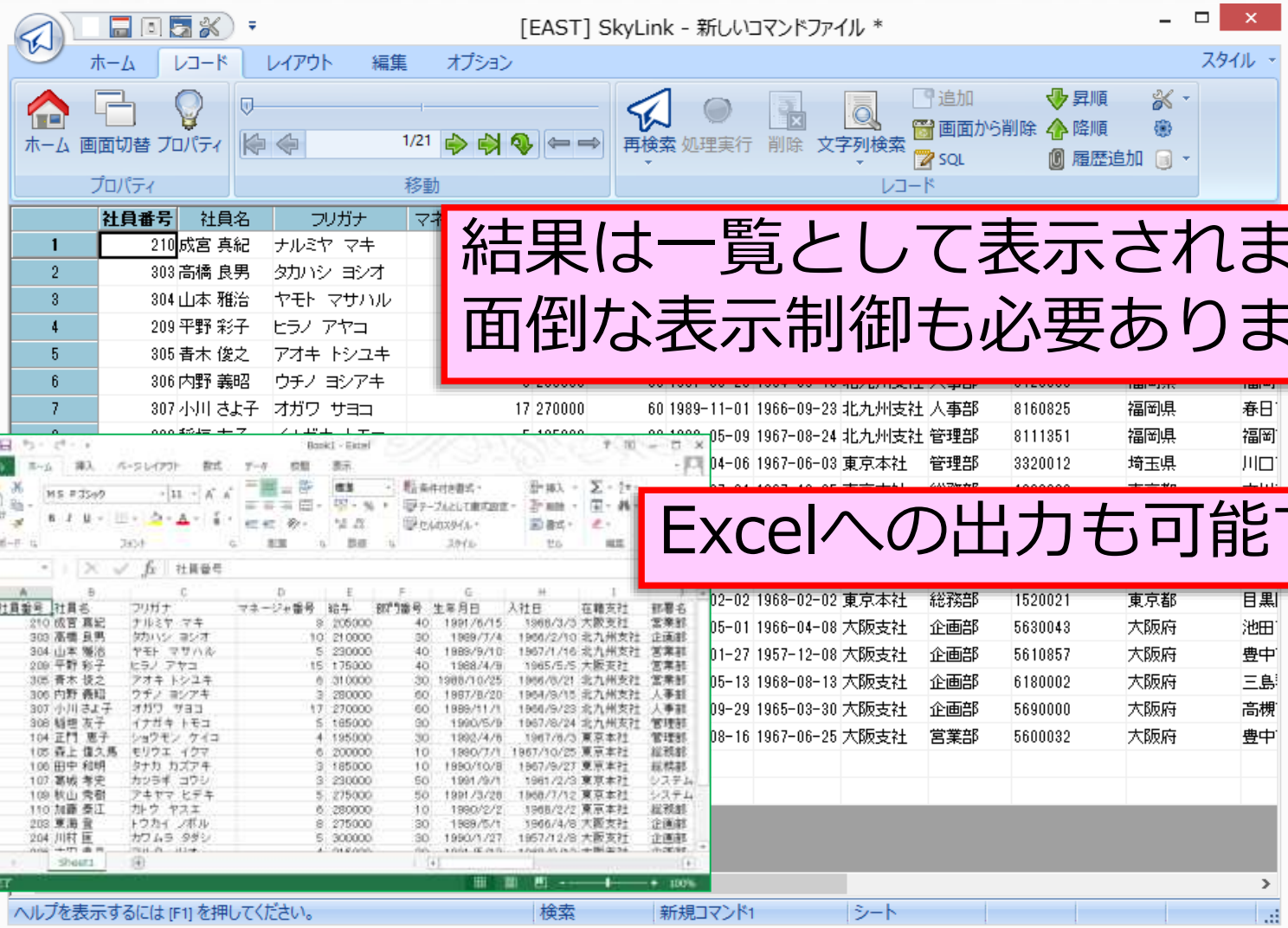
データ設定



イースト est.co.jp

SkyLinkなら・・・

結果表示



結果は一覧として表示されます。
面倒な表示制御も必要ありません。

Excelへの出力も可能です。