

# いまさら聞けないVB2008 ADO.NET超入門

2008.05.31

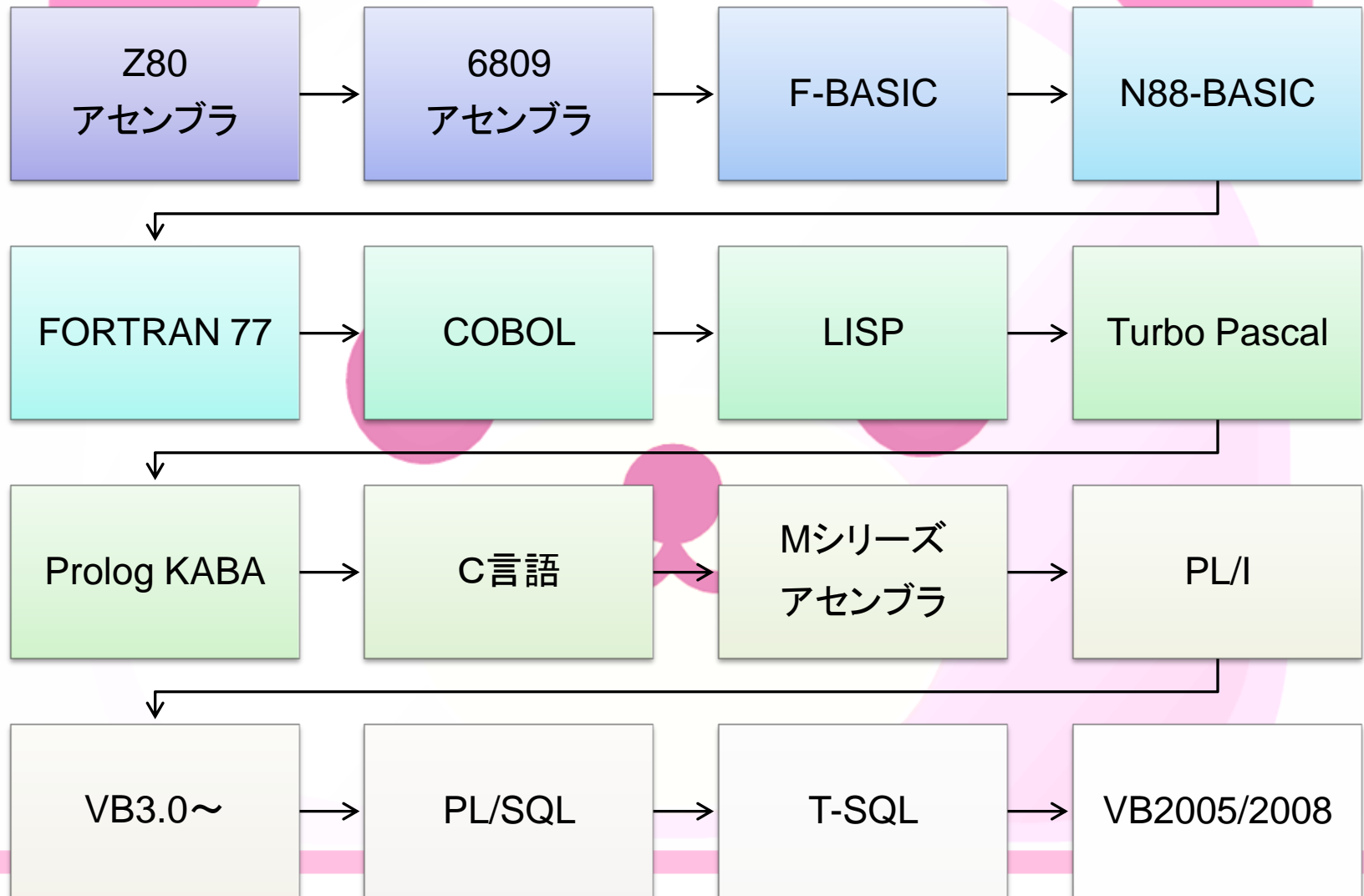
初音 玲



ぴんくま  
同盟

わんくま同盟 東京勉強会 #20 [ぴんくまDay]

# 自己紹介



index

接続

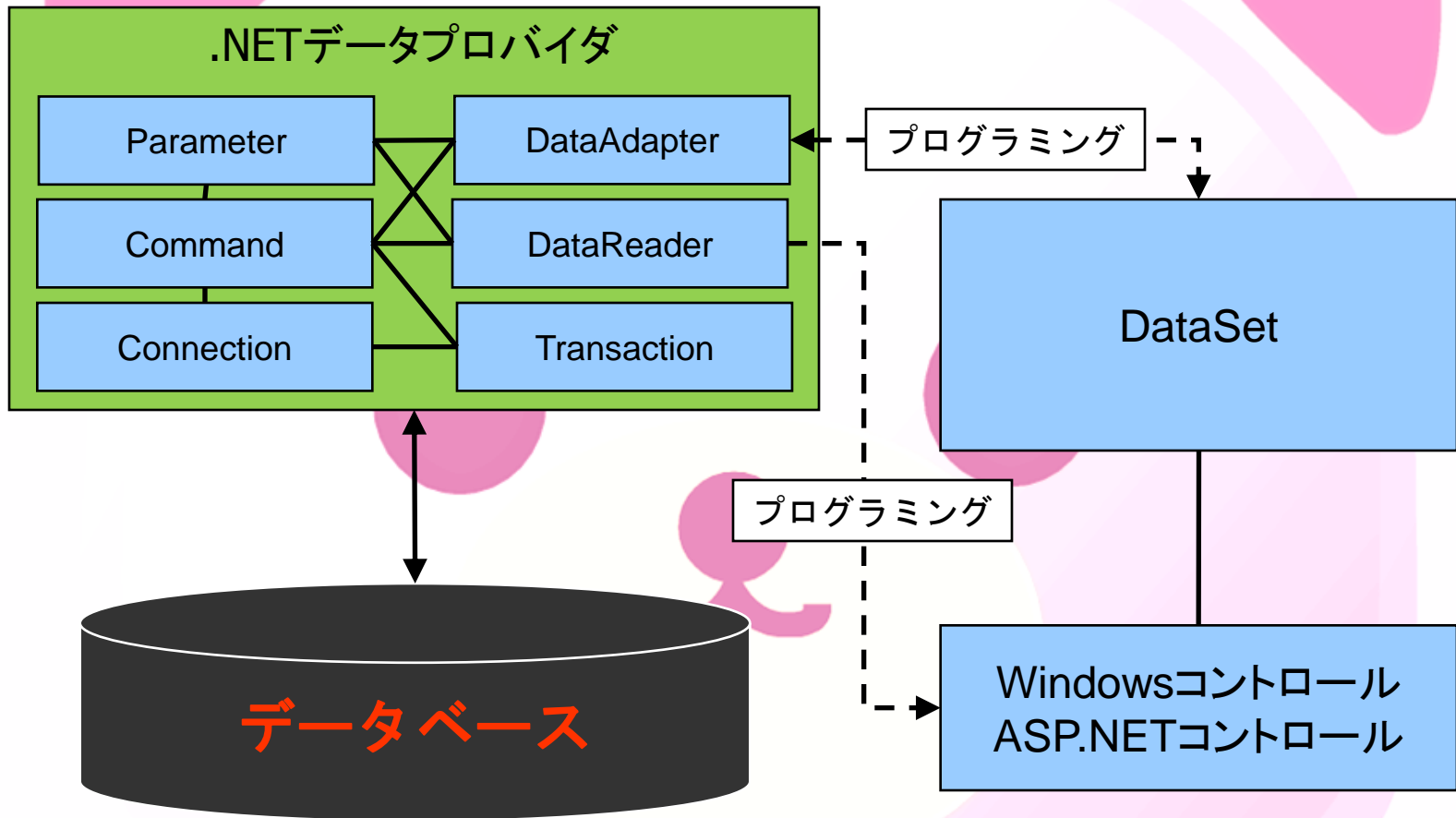
データ取得

データ更新

権限

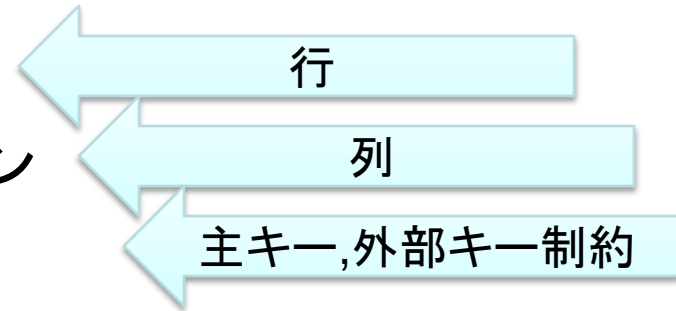


# ADO.NETの基本的な構造



## DataSetクラス

- メモリ上の仮想データベース
- DataTablesコレクション
  - DataTableクラス
    - DataRowコレクション
    - DataColumnコレクション
    - Constraintsコレクション
- DataRelationsコレクション
  - RDBMSのリレーション定義に相当
  - 親子関係を定義



## .NETデータプロバイダ

Connection  
オブジェクト

- 特定のデータソースへの接続を確立

Command  
オブジェクト

- データソースに対してコマンドを実行

DataReader  
オブジェクト

- データソースから前方向、読取専用でデータ取得

DataAdapter  
オブジェクト

- DataSetを設定し、データソースを使用して更新内容を解決



index

接続

データ取得

データ更新

権限



# Connectionオブジェクト

SQL Server

SqlConnection

- Dim cn As New **SqlConnection**()
- cn.ConnectionString = "User Id=sa;" & \_  
"Password=aU98rrx2;" & \_  
"Initial Catalog=pubs;" & \_  
"Data Source=servername;"
- cn.Open()

Oracle

OracleConnection

- Dim cn As New **OracleConnection**()
- cn.ConnectionString = "User Id=scott;" & \_  
"Password=tiger;" & \_  
"Data Source=orcl.world;"
- cn.Open()





## 接続文字列の設定タイミング

### 変数宣言時点

- Dim cn As New SqlConnection("User id=.....")
- cn.Open

### 任意

- Dim cn As New SqlConnection()  
:  
(中略)  
:
- cn.ConnectionString = "User id=....."
- cn.Open()



## ADO.NETからのエラーの取得

### Try~Catch

- cn.OpenをTry~Catchで囲む
- CatchにはExceptionではなくSqlException

### System.Data.SqlClient.SqlException

- ex.Message
- ex.StackTrace
- ex.ErrorCode
- ex.State
- ex.Server



# ADO.NETからの切断

利用状況モニタ - localhost#SQLEXPRESS

更新 フィルタ... ヘルプ

17 項目 の合計から 2 項目 が表示されました。

	プロセス ID	ユーザー	データベース	状態	開い...	コマンド	アプリケーション
↔	51	sa	pubs	休止中	0	AWAITING ...	.Net SqlClient D.
↔	54	sa	tempdb	実行可能	2	SELECT IN...	Microsoft SQL S

接続のプロパティを表示します

進行状況

完了

閉じる(C)

- Closeメソッドの実行
- Connectionオブジェクトの破棄

本当？



index

接続

データ取得

データ更新

権限



# Commandオブジェクト

SQL Server

SqlCommand

- Using \_cmd As New **SqlCommand**()
- \_cmd.Connection = cn
- \_cmd.CommandText = "SELECT \* FROM employee"
- End Using

Oracle(ODP.NET)

OracleCommand

- Using \_cmd As New **OracleCommand**()
- \_cmd.Connection = cn
- \_cmd.CommandText = "SELECT \* FROM employee"
- End Using



# SELECT文設定タイミング

## 変数宣言時点

- Using \_cmd As New SqlCommand(sqlString, cn) ←
- :
- End Using

## 任意

- Using \_cmd As New SqlCommand()
- \_cmd.Connection = cn ←
- \_cmd.CommandText = sqlString
- :
- End Using

Connection

接続  
タイミング  
は？

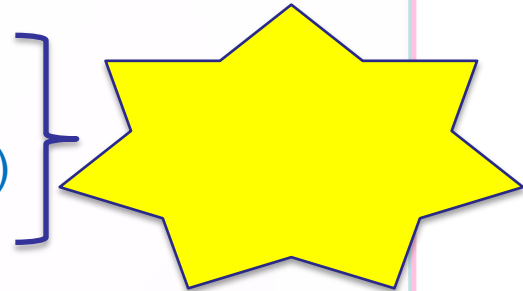


# DataReaderオブジェクト

SQL Server

SqlDataReader

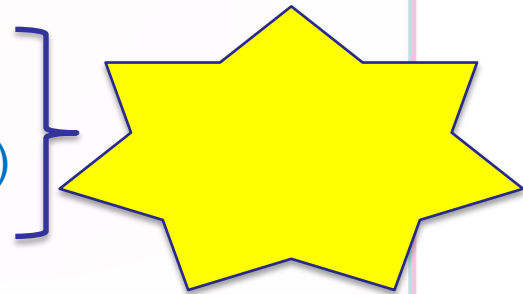
- Dim rd As **SqlDataReader**()
- rd = \_cmd.ExecuteReader
- Do While rd.Read
- Me.ResultList.Items.Add(rd.Item("fname").ToString)
- Loop
- rd.Close()



Oracle(ODP.NET)

OracleDataReader

- Dim rd As **OracleDataReader**()
- rd = \_cmd.ExecuteReader
- Do While rd.Read
- Me.ResultList.Items.Add(rd.Item("fname").ToString)
- Loop
- rd.Close()



## DataReaderを使う上での注意点

```
Using _cn As New SqlConnection(CnString)
    _cn.Open()
    Using _cmd As New SqlCommand(SqlString, _cn)
        Dim rd As SqlDataReader = Nothing
        rd = _cmd.ExecuteReader
        Do While rd.Read
            Me.ResultList.Items.Add(rd.Item("fname").ToString)
        Loop
    End Using
Using _cmd As New SqlCommand(SqlString, _cn)
    Dim rd As SqlDataReader = Nothing
    rd = _cmd.ExecuteReader
    Do While rd.Read
        Me.ResultList.Items.Add(rd.Item("fname").ToString)
    Loop
End Using
_cn.Close()
End Using
```

間違いは  
どこ？





## 列単位でデータを実取得する

```
rd = _cmd.ExecuteReader()
```

- rd.Readメソッドで1行分を読み込む
- 1行前の読みなおしは不可
- 行を飛ばして読み込むのも不可

```
rd = _cmd.ExecuteReader(_  
CommandBehavior.SequentialAccess)
```

- rd.Item(0)で1項目分を読み込む  
rd.GetBytes(0,stp,outSize,0,bufferSize)で分割読み込み
- 先頭項目から順番に取得
- 項目取得後に、それより前の項目取得不可
- 項目を飛ばして取得不可



# Parameterオブジェクト

SQL Server

SqlParameter

- sqlString = \_"SELECT \* FROM employee " & \_  
"WHERE fname=@fname AND lname=@lname"
- Using \_cmd As New **SqlCommand**(sqlString, \_cn)
- \_cmd.Parameters.Add(New **SqlParameter**("fname", Me.FName.Text))
- \_cmd.Parameters.Add(New **SqlParameter**("lname", Me.LName.Text))
- End Using

Oracle(ODP.NET)

OracleParameter

- sqlString = "SELECT \* FROM employee " & \_  
"WHERE fname=:fname AND lname=:lname"
- Using \_cmd As New **OracleCommand**(sqlString, \_cn)
- \_cmd.BindByName = True
- \_cmd.Parameters.Add("fname", Me.FName.Text)
- \_cmd.Parameters.Add("lname", Me.LName.Text)
- End Using

なぜ違う？

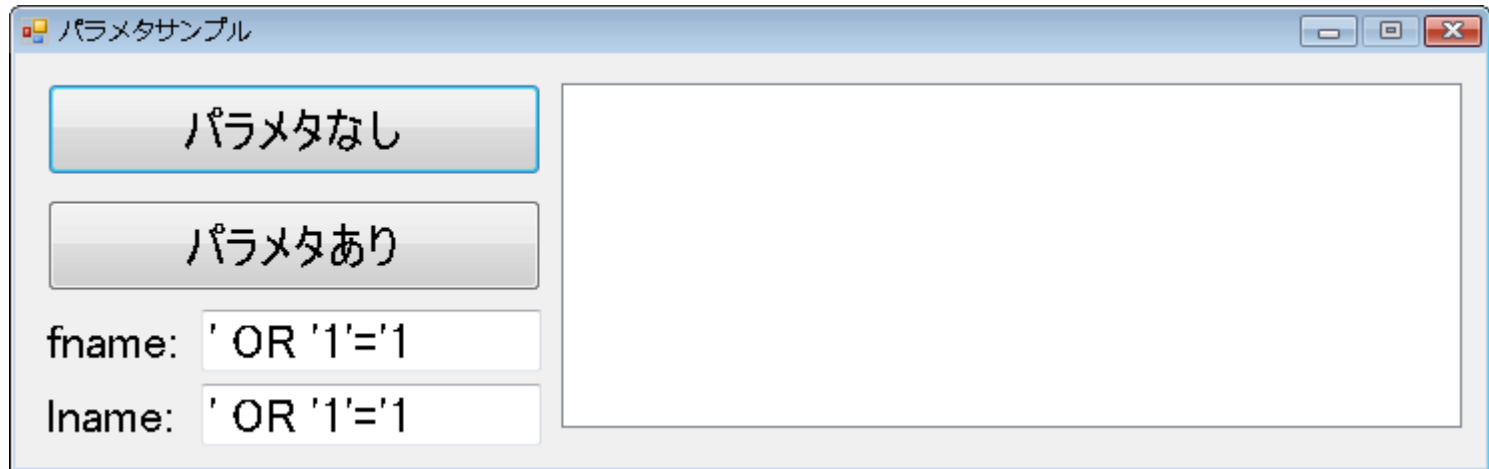


pinkuma  
同盟

わんくま同盟 東京勉強会 #20 [pinkumaDay]

# Parameterオブジェクト

以下の条件を与えたときの実行結果は？



パラメタサンプル

パラメタなし

パラメタあり

fname: ' OR '1'='1

lname: ' OR '1'='1

## ストアドプロシージャでデータ取得

```
Using _cmd As New SqlCommand("byroyalty", _cn)
  Dim rd As SqlDataReader = Nothing
  _cmd.CommandType = CommandType.StoredProcedure
  _cmd.Parameters.Add(New SqlParameter("percentage", 40))
  Try
    Me.ListBox1.Items.Clear()
    rd = _cmd.ExecuteReader
    Do While rd.Read
      Me.ResultList.Items.Add(rd.Item("au_id").ToString)
    Loop
  Catch ex As Exception
    MessageBox.Show(ex.Message, ".....")
  Finally
    If Not rd Is Nothing Then
      rd.Close()
    End If
  End Try
End Using
```

```
ALTER PROCEDURE [dbo].[byroyalty] @percentage int
AS
select au_id from titleauthor
where titleauthor.royaltyper = @percentage
```



index

接続

データ取得

データ更新

権限



# SQL文の直接実行(Commandオブジェクト)

SQL Server

SqlParameter

- sqlString = \_  
"UPDATE employee SET minit=@minit WHERE emp\_id=@emp\_id"
- Using \_cmd As New **SqlCommand**(sqlString, \_cn)
- \_cmd.Parameters.Add(New **SqlParameter**("emp\_id", Me.Emp\_Id.Text))
- \_cmd.Parameters.Add(New **SqlParameter**("minit", Me.Minit.Text))
- \_cmd.ExecuteNonQuery
- End Using

Oracle(ODP.NET)

OracleParameter

- sqlString = \_  
"UPDATE employee SET minit=:minit WHERE emp\_id=:emp\_id"
- Using \_cmd As New **OracleCommand**(sqlString, \_cn)
- \_cmd.BindByName = True
- \_cmd.Parameters.Add("emp\_id", Me.Emp\_Id.Text)
- \_cmd.Parameters.Add("minit", Me.Minit.Text)
- \_cmd.ExecuteNonQuery
- End Using



# データソースとDataSetクラスの対応付け

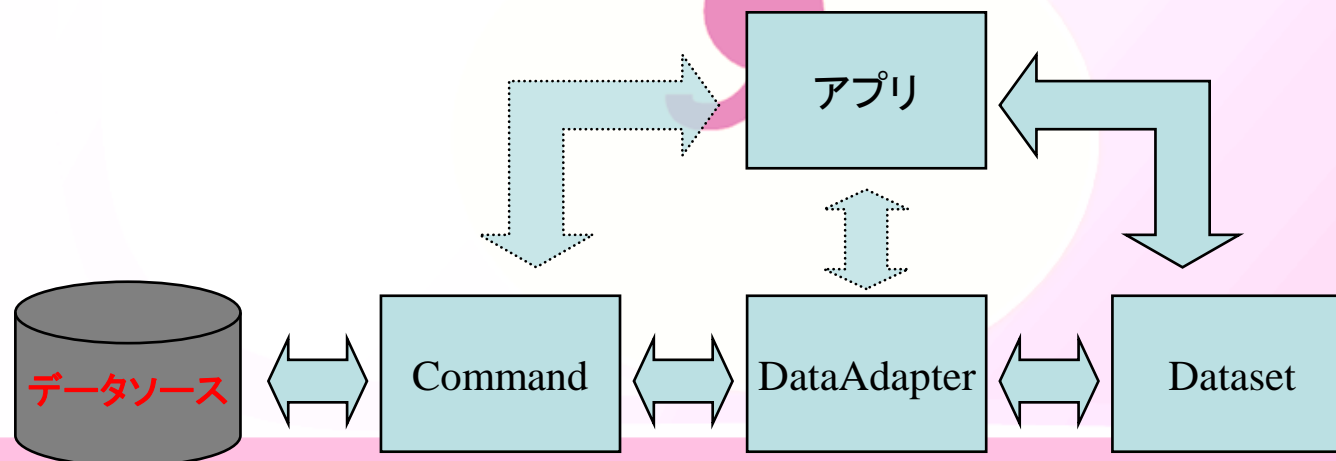
.NETデータプロバイダには、更新可能セットがない

DataSetは仮想的なデータベース

- もちろん更新も可能

.NETデータプロバイダとDataSetの相互乗り入れ

- DataSetによるデータソースの更新可能セットを実現



# DataAdapterオブジェクト

SQL Server

SqlDataAdapter

- Using \_cn As New **SqlConnection**(CnString)
- Using \_cmd As New **SqlCommand**("SELECT \* FROM employee", \_cn)
- Using \_da As New **SqlDataAdapter**(\_cmd)
- \_da.Fill(ds, "employee")
- Me.ResultGrid.DataSource = ds.Tables("employee")
- End Using
- End Using
- End Using

Oracle(ODP.NET)

OracleDataAdapter

- Using \_cn As New **OracleConnection**(CnString)
- Using \_cmd As New **OracleCommand**("SELECT \* FROM employee", \_cn)
- Using \_da As New **OracleDataAdapter**(\_cmd)
- \_da.Fill(ds, "employee")
- Me.ResultGrid.DataSource = ds.Tables("employee")
- End Using
- End Using
- End Using

Open  
タイミングは？



pinkuma  
同盟

わんくま同盟 東京勉強会 #20 [pinkumaDay]



# CommandBuilderでSQL作成

```
_cn.Open()
Using _tr As SqlTransaction = _cn.BeginTransaction()
    Using _cmd As New SqlCommand("SELECT * FROM employee ", _cn)
        _cmd.Transaction = _tr    '###重要###
        Using _da As New SqlDataAdapter(_cmd)
            Using _cb As New SqlCommandBuilder(_da)
                _da.UpdateCommand = _cb.GetUpdateCommand()
                _da.InsertCommand = _cb.GetInsertCommand()
                _da.DeleteCommand = _cb.GetDeleteCommand()
                Try
                    _da.Update(Ds, "employee")
                    _tr.Commit()
                Catch ex As Exception
                    MessageBox.Show(ex.Message, "...")
                    _tr.Rollback()
                End Try
            End Using
        End Using
    End Using
End Using
```



# CommandBuilderオブジェクト

## SetAllValuesプロパティ

- True:UPDATE SQLのSET句にすべてセット
- False:UPDATE SQLのSET句に変更値だけをセット
- 利用状況モニタで確認  
SET [minit] = @p1

## ConflictOptionプロパティ

- CompareAllSearchableValues  
WHERE (([emp\_id] = @p9) AND ([fname] = @p10) AND ((@p11 = 1  
AND [minit] IS NULL) OR ([minit] = @p12)) AND ([lname] = @p13)  
AND ([job\_id] = @p14) AND ((@p15 = 1 AND [job\_lvl] IS NULL) OR  
([job\_lvl] = @p16)) AND ([pub\_id] = @p17) AND ([hire\_date] = @p18))
- CompareRowVersion  
WHERE (([emp\_id] = @p9))
- OverwriteChanges  
WHERE (([emp\_id] = @p9))



# DataSetの利用時の注意点

## SQL Serverの10進数型の扱い

- SQL Serverの10進数型の有効桁数
  - 38桁
- DataSet (CLR)の10進数型の有効桁数
  - 28桁

## 実行時エラー

- Fillメソッド実行時に10進数型の値が28桁を超えたとき

## 対策

- System.Data.SqlTypesを明示的に使用(ADO.NET 2.0以降)
- <http://msdn.microsoft.com/ja-jp/library/bh8kx08z.aspx>



## DataSetの利用時の注意点

Fillメソッドで全データ取得

selectメソッドで必要な情報取得

Fillメソッドで必要データのみ取得

表示データから選択データ抽出

selectメソッド利用



(おまけ)LINQ to SQL

LINQtoSQLクラスを追加(.dbml)

サーバエクスプローラからDrag&Drop

標準クエリ演算子を記述



index

接続

データ取得

データ更新

権限



## 権限

### 接続は、アプリ固有ユーザID

- コネクションプーリング
- アプリ側でアクセス制御
- 直接ツールで接続されたら？
- アプリ側にバグがあったら？

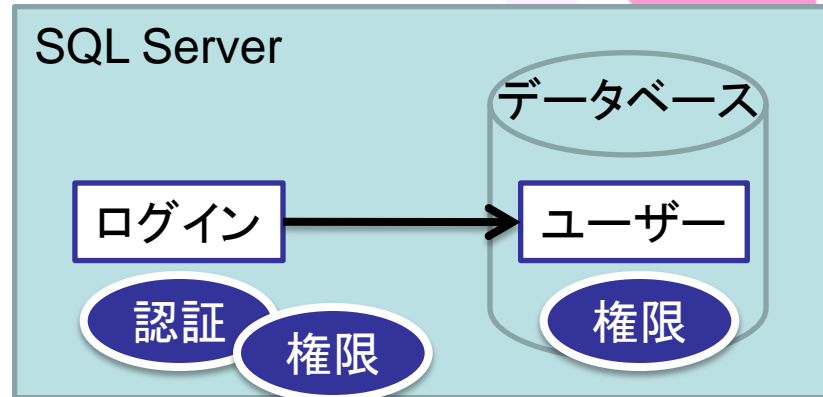
### 接続は、利用者固有ユーザID

- コネクションプーリング
- DBの設定でアクセス制御
- 直接ツールで接続されても安全
- アプリ側にバグがあっても安全

# SQL Serverにおけるユーザ管理

## SQL Server認証

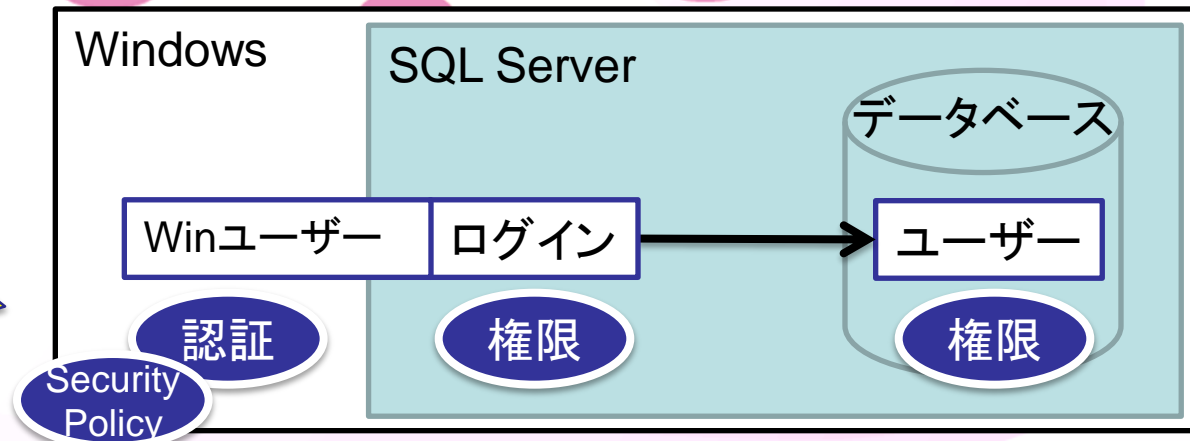
旧版との互換性  
のため？



<http://msdn.microsoft.com/ja-jp/library/aa905171.aspx>

## Windows認証

こちらが推奨？



ローカル認証だと4万人くらいが限界なのでAD認証も考慮



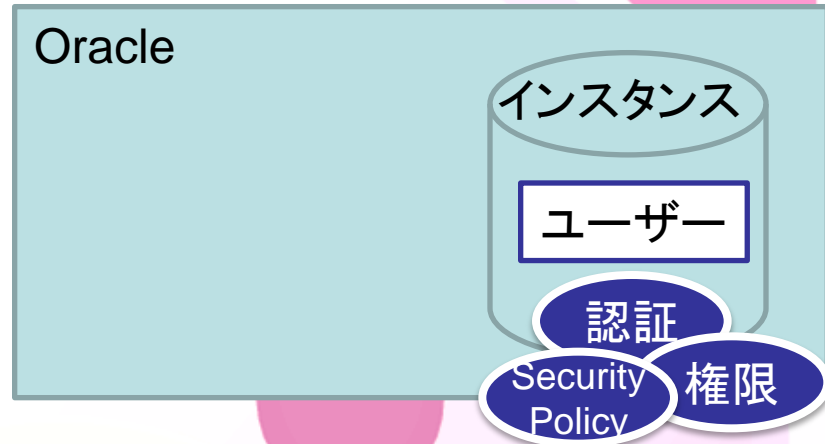
pinkuma  
同盟

わんくま同盟 東京勉強会 #20 [pinkumaDay]

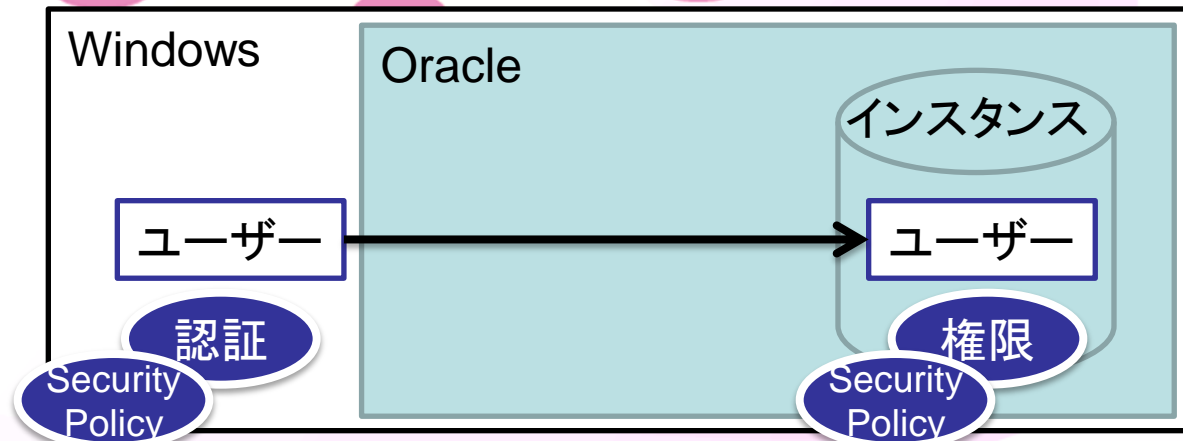


# Oracleにおけるユーザ管理

Oracle認証



Windows認証



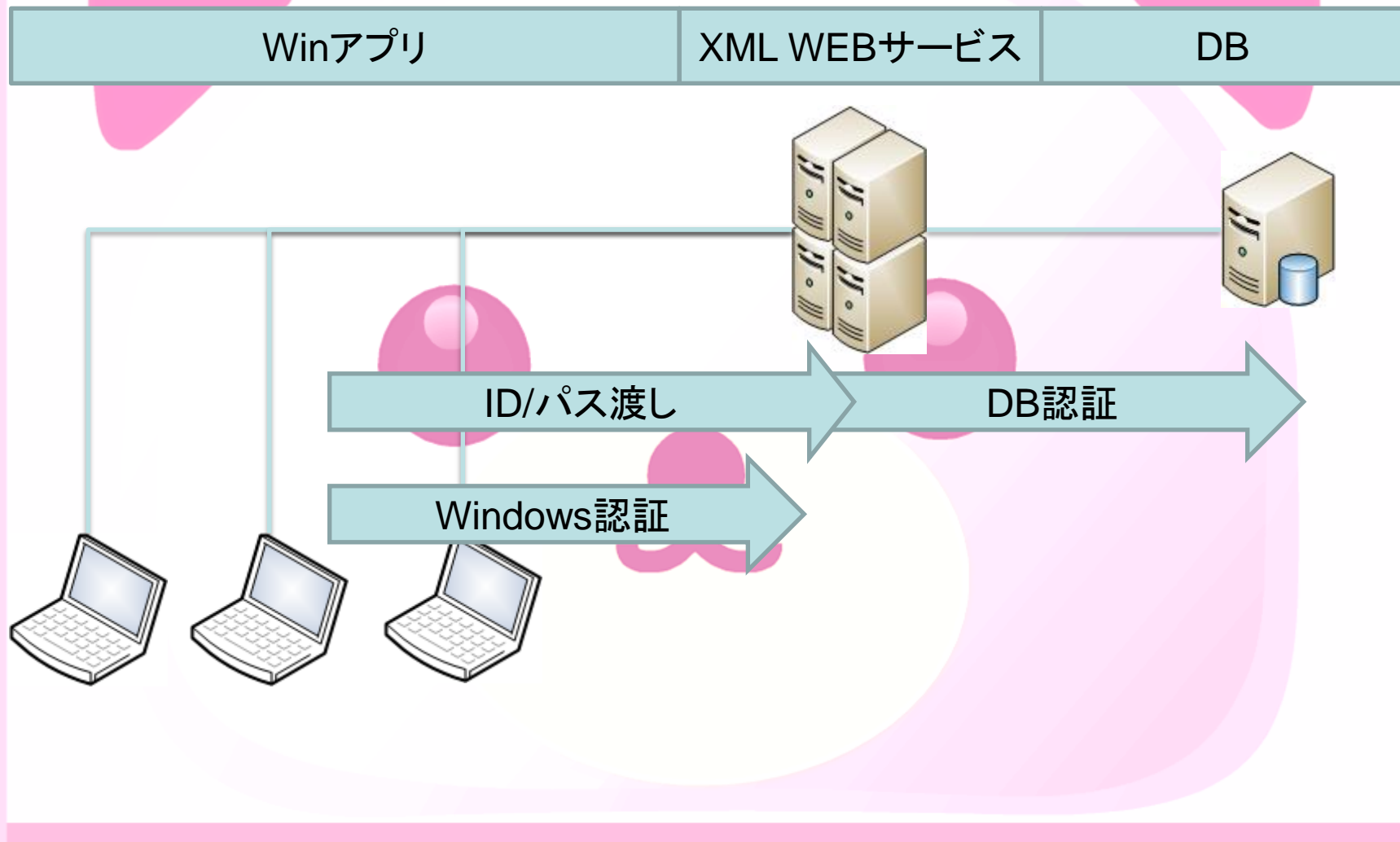
ローカル認証だと4万人くらいが限界なのでAD認証も考慮



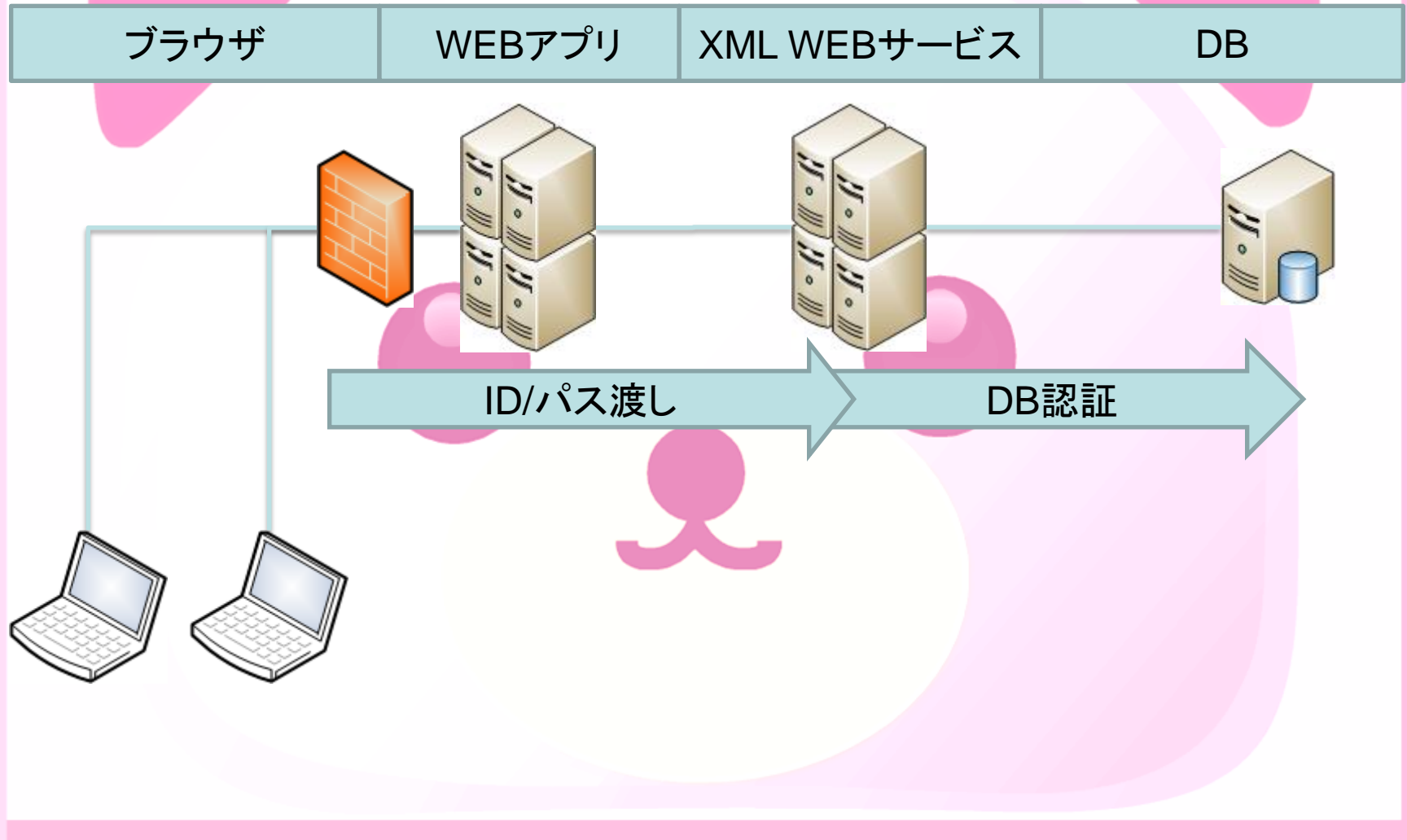
ぴんくま  
同盟


わんくま同盟 東京勉強会 #20 [ぴんくまDay]

# Windowsアプリにおけるお勧め認証構造



# WEBアプリにおけるお勧め認証構造





ありがとうございました



ぴんくま  
同盟

わんくま同盟 東京勉強会 #20 [ぴんくまDay]