

Integrazione web-DBMS

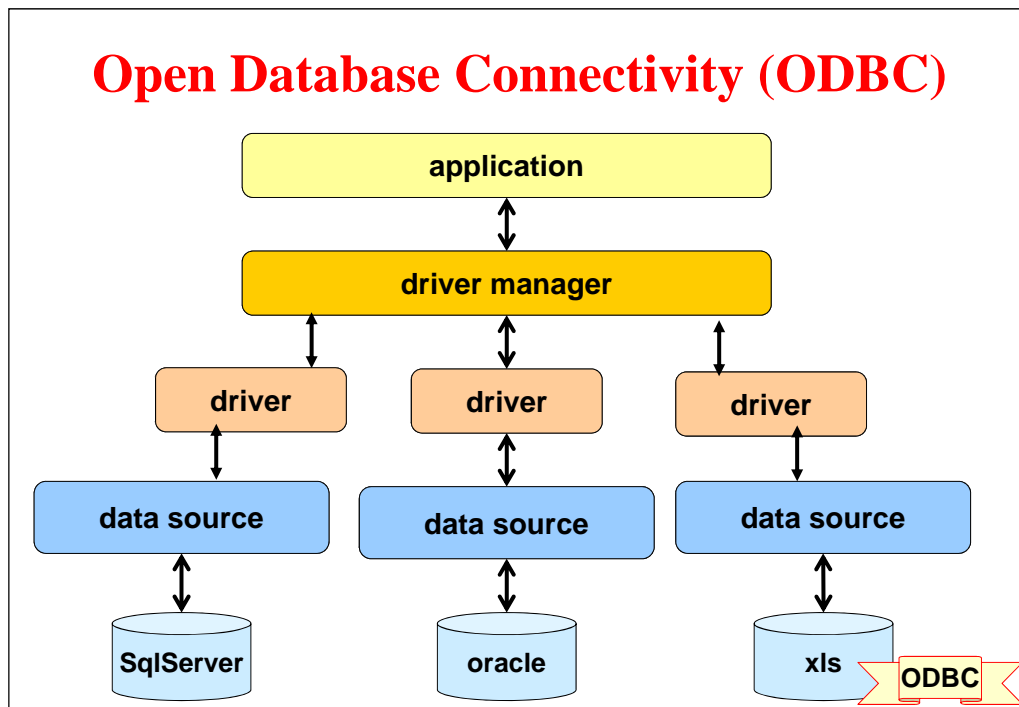
Gianluca Ramunno
< ramunno@polito.it >

Politecnico di Torino
Dip. Automatica e Informatica

Open Database Connectivity (ODBC)

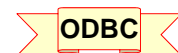
- una tecnologia parte di Microsoft Data Access Components (MDAC) per l'accesso ai DBMS
 - interfaccia a basso livello C++
- inclusa nei S.O. Microsoft
- permette l'accesso ai dati da qualsiasi applicazione indipendentemente dal DBMS
- inserimento di un middleware (il **database driver**) tra applicativo e DBMS
- usa SQL come linguaggio di manipolazione ed accesso ai dati
- usato da DAO (interfaccia a livello applicativo)

ODBC



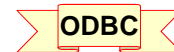
Open Database Connectivity (ODBC)

- **application**
- **driver manager (DLL)**
 - carica i driver su richiesta dell'applicazione
- **driver (DLL)**
 - elabora le chiamate ODBC ricevute dal Driver Manager
 - il driver manager e i driver appaiono all'applicazione come un unico blocco
- **data source**
 - DBMS, rete, SO su cui gira il DBMS

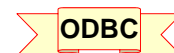
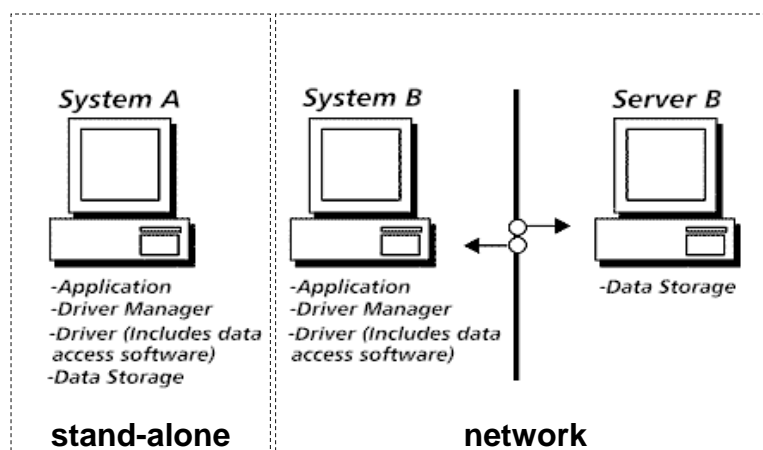


Driver in configurazione single-tier

- sullo stesso sistema risiedono:
 - application
 - driver manager
 - driver
 - software per accesso ai dati (data access software)
 - data storage
- esistono due configurazioni:
 - stand alone
 - network

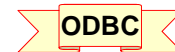


Driver in configurazione single-tier

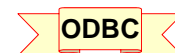
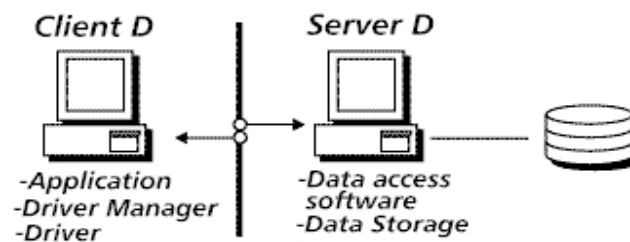


Driver in configurazione multiple-tier

- **sul sistema client risiedono:**
 - application
 - driver manager
 - driver
- **sul sistema server risiedono:**
 - software per accesso ai dati (**data access software**)
 - data storage

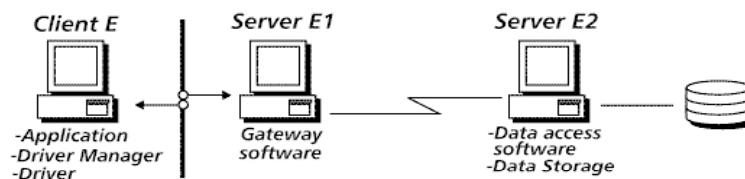


Driver in configurazione multiple-tier



Driver multiple-tier con gateway

- esiste un processo gateway
- instrada le richieste al data access software



OLE DB

- tecnologia Microsoft (MDAC) inclusa nei S.O.
 - evoluzione di ODBC
- interfaccia COM (Component Object Model) per fornire alle applicazioni un accesso uniforme a dati da fonti diverse
 - DBMS relazionali
 - DBMS non relazionali (novità)
 - file system, mail system, fogli di calcolo, directory, ...
- non limitato a SQL
- interfaccia a basso livello C++

OLE DB

OLE DB

■ consumer

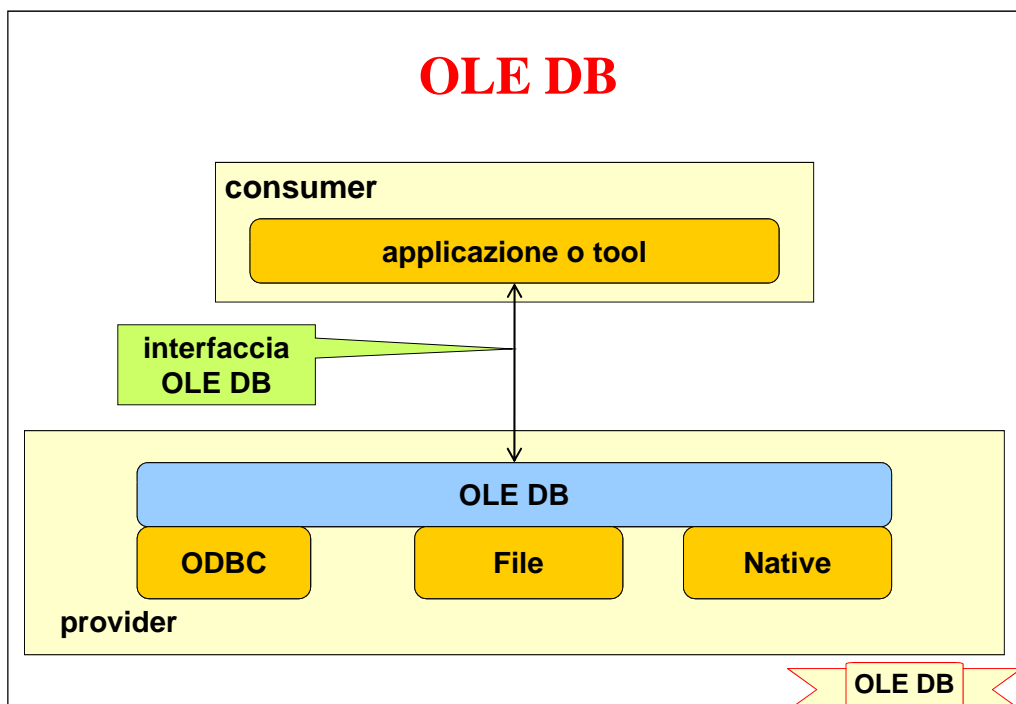
- applicazioni che attraverso l'interfaccia OLE DB manipola dei dati

■ provider

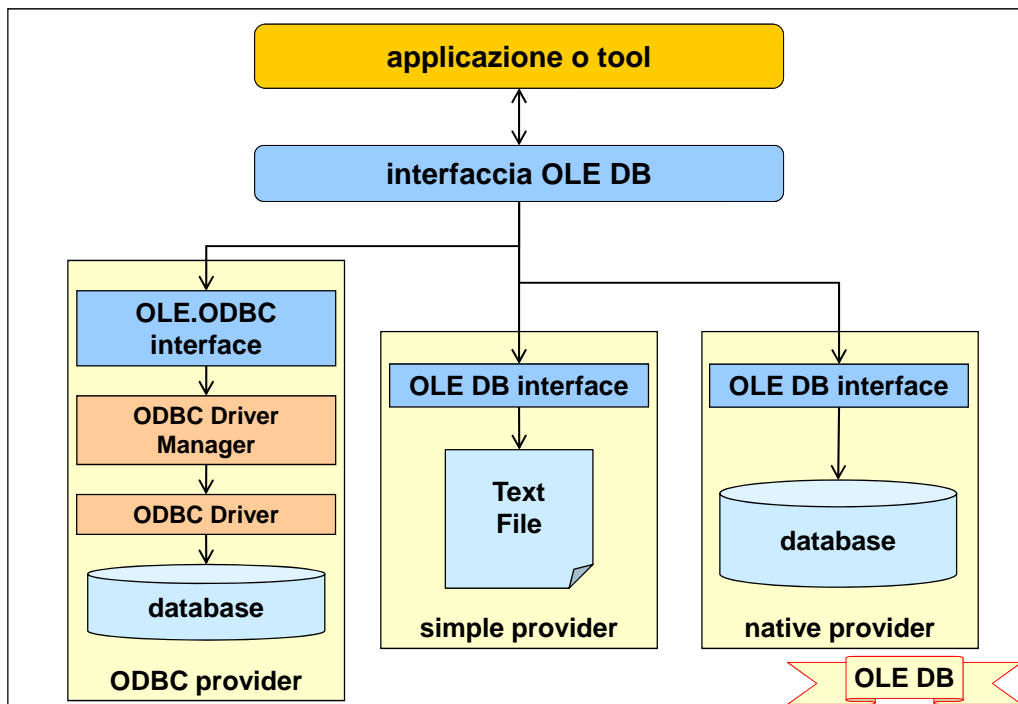
- intermediario che mette a disposizione i dati
- espone metodi OLE DB
- non tutti i provider hanno le stesse capacità, dipende da quali metodi OLE DB implementano

OLE DB

OLE DB

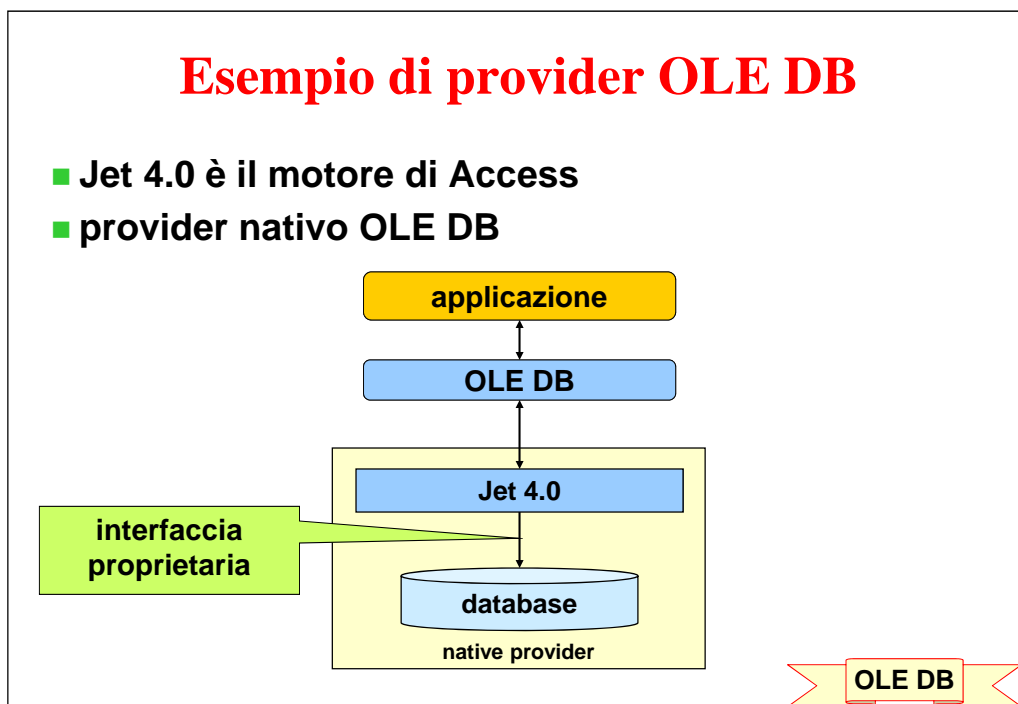


OLE DB



Esempio di provider OLE DB

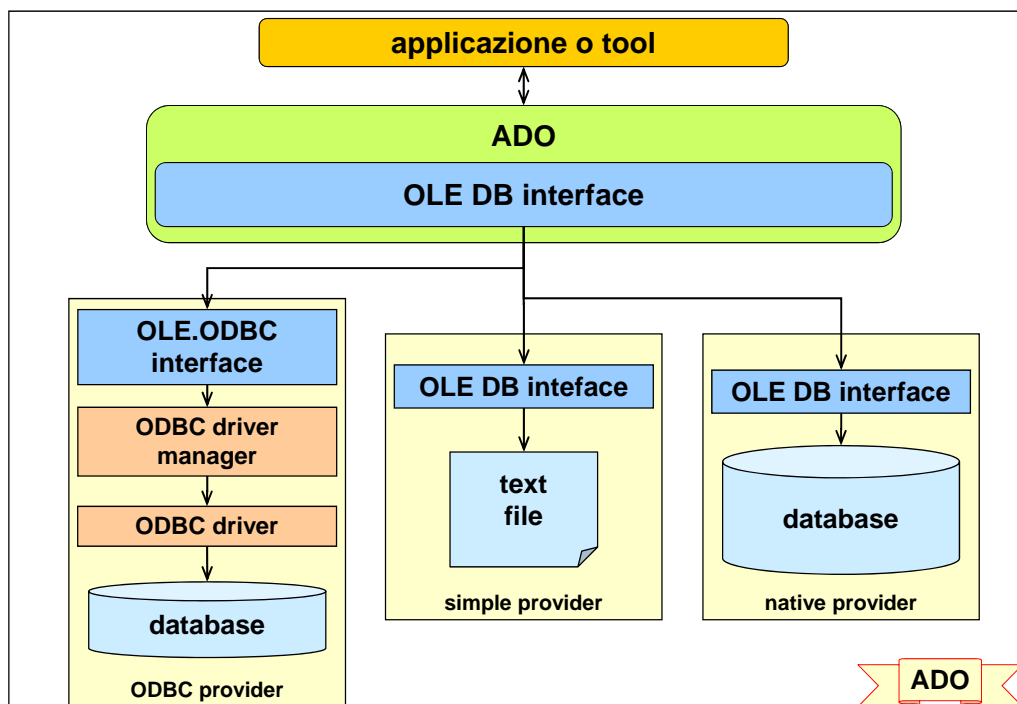
- Jet 4.0 è il motore di Access
- provider nativo OLE DB



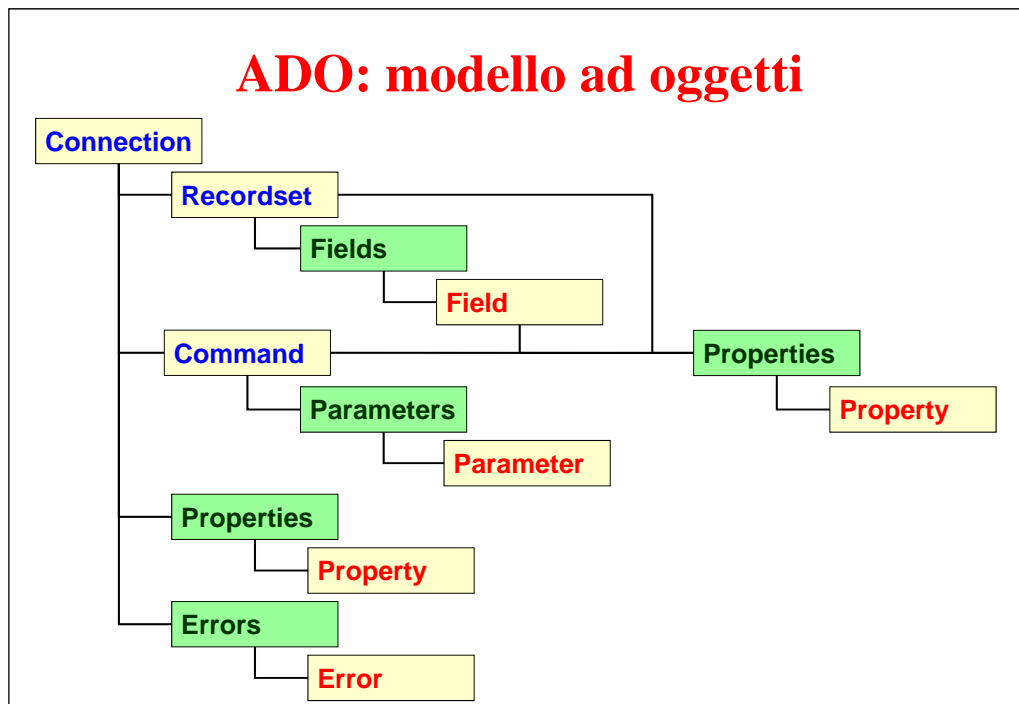
ADO (ActiveX Data Object)

- **middleware con interfaccia ad oggetti**
 - evoluzione di DAO
- **mette a disposizione l'interfaccia OLE DB a linguaggi diversi dal C++**
 - è un wrapper di OLE DB ad un livello applicativo
 - può essere usato con diversi linguaggi all'interno di
 - programmi utente compilati (VisualBasic)
 - macro in applicazioni come MS-Office (VisualBasic for Application - VBA)
 - WSH: script console (Jscript, VBscript)
 - pagine ASP in IIS (Jscript, VBscript)

ADO

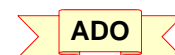


ADO



Oggetti ADO

- **Connection object**
 - rappresenta una sessione con una sorgente dati
- **Command object**
 - usato per definire un comando specifico come una query SQL verso una sorgente dati
- **Recordset object**
 - rappresenta un insieme di record di una tabella intera o del risultato di una query
 - composto da righe (record) e colonne (field)



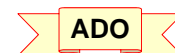
Oggetti ADO

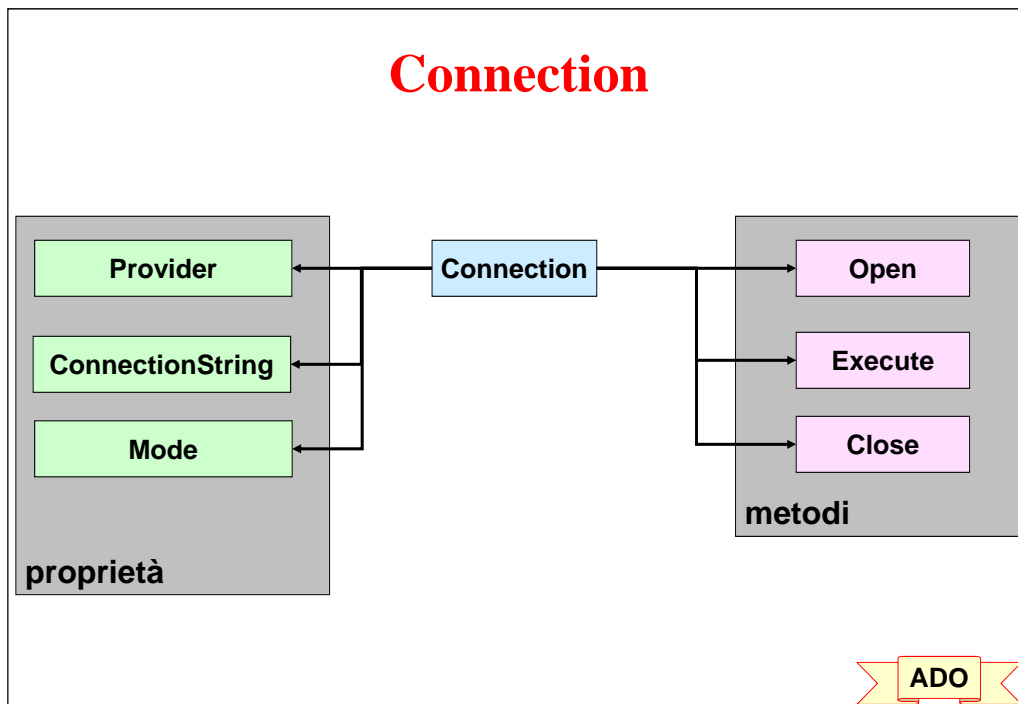
- **Record object**
 - rappresenta una riga di dati
 - corrisponde ad un singolo record di un recordset
- **Field object**
 - rappresenta una colonna di dati
 - corrisponde ad una colonna di un recordset
- **Parameter object**
 - rappresenta un parametro associato ad un oggetto command (es. parametri per stored procedure)



Oggetti & collection ADO

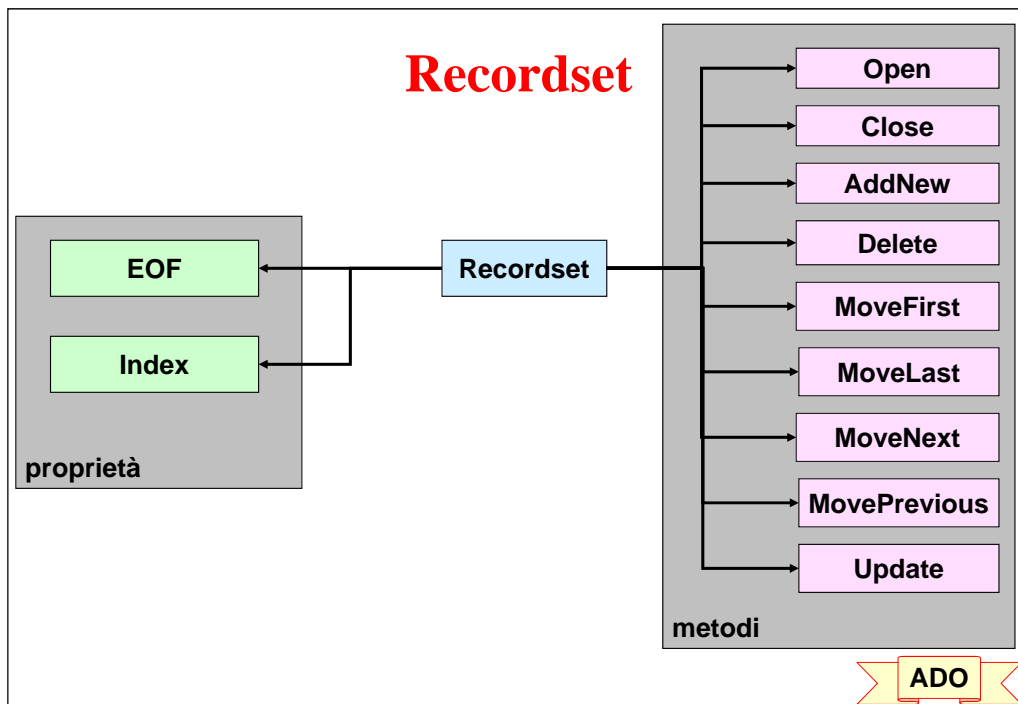
- **Error object**
 - dettagli di un eventuale errore avvenuto in un operazione col provider
- **Fields collection**
 - contiene tutti gli oggetti **Field** di un recordset
- **Errors collection**
 - contiene tutti gli oggetti **Error** generati da un operazione non riuscita col un provider
- **Parameters collection**
 - contiene tutti gli oggetti **Parameter** di un command





Connection

- **Mode**
 - modalità di apertura del DB
 - read-only è il default
 - per avere read-write con.Mode = adModeReadWrite
- **Execute (stringSQL)**
 - restituisce un RS read-only con cursore forward-only
 - se si vogliono altre proprietà, non usare con.Execute ma rs.Open

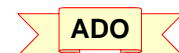


Recordset: metodo open

- **apertura di un recordset**

```
ObjRS.Open ( Source, ActiveConnection,
             CursorType, LockType, option );
```

- **Source**: comando SQL, nome di tabella, stored procedure
- **ActiveConnection**: un oggetto di tipo Connection, oppure una stringa di connessione
- **CursorType**: tipo di cursore sul RS risultante
- **Locktype**: tipo di lock sui dati
- **Option**: interpretazione di Source



CursorType

adOpenUnspecified	Unspecified type of cursor.
adOpenForwardOnly (default)	This improves performance when only one pass through a RS is needed.
adOpenKeyset	Like a dynamic cursor, except that you can't see records that other users add, although records that other users delete are inaccessible from your RS. Data changes by other users are still visible.
adOpenDynamic	Additions, changes, and deletions by other users are visible, and all types of movement through the RS are allowed.
adOpenStatic	A static copy of a set of records that you can use to find data or generate reports. Additions, changes, or deletions by other users are not visible.

Recordset: metodo AddNew

■ aggiunta di un record

```
// se la tabella non e' già accessibile ...
objRS.Open("utenti", conString,
           adOpenDynamic, adLockPessimistic);
// aggiunta del record
objRS.AddNew();
objRS("Nome") = "Alberto";
objRS("Cognome") = "Rossi";
objRS.Update();
```

ADO

ADO modello di operazione (query)

- creare un oggetto ADO connessione
- aprire la connessione verso un DBMS
- creare un oggetto ADO recordset
- aprire il recordset
- manipolare i dati nel recordset
- chiudere il recordset
- chiudere la connessione
- rimuovere gli oggetti

ADO

ADO: esempio ASP/JScript (query)

- creazione ed apertura connessione

```
<% @LANGUAGE="JScript" %>
<!--#include file="adojavas.inc"-->
<%
var con;
var conString;
con = Server.CreateObject("ADODB.Connection");
conString = "Provider=Microsoft.Jet.OLEDB.4.0; "
           + "Data Source= c:\\webdata\\mydb.mdb ";
con.Open(conString);
%>
```

ADO

ADO: esempio ASP/Jscript (query)

■ creazione ed apertura recordset

```
<%  
var rst;  
var sSQLString;  
rst = Server.CreateObject("ADODB.Recordset");  
sSQLString = "Select * from Customers";  
rst = con.Execute(sSQLString);  
%>
```

ADO

ADO: esempio ASP/JScript (query)

■ estraggo i dati dal recordset **rst**

```
<%  
while (!rst.EOF) {  
    Response.write(rst("Name") + "<BR>");  
    rst.MoveNext( );  
}  
%>
```

ADO

ADO: esempio ASP/JScript (query)

■ chiudo recordset & connessione

```
<%  
rst.Close( );  
con.Close( );  
rst = null;  
con = null;  
%>
```

ADO

Accesso a DB con path relativo

■ col metodo Server.MapPath

```
<% @LANGUAGE="JScript" %>  
<%  
var con;  
var conString;  
  
con = Server.CreateObject("ADODB.Connection");  
conString = "Provider = Microsoft.Jet.OLEDB.4.0; "  
+ "Data Source = " + Server.MapPath("./mydb.mdb");  
con.Open(conString);  
... .
```

ADO: esempio di update tramite SQL

■ creazione ed apertura connessione

```
<% @LANGUAGE="JScript" %>
<!--#include file="adojavas.inc"-->
<%
var con;
var conString;
con = Server.CreateObject("ADODB.Connection");
conString = "Provider=Microsoft.Jet.OLEDB.4.0; "
           + "Data Source= c:\\webdata\\mydb.mdb ";
con.Open(conString);
%>
```

ADO

ADO: esempio di update tramite SQL

- aggiunta record al DB (tramite SQL)
- chiusura connessione

```
<%
var sSQLString =
  "INSERT INTO tBooks (Title, Description, Category)
  VALUES ('DB Design','How to design a database', 3);"
con.Execute(sSQLString);

con.Close( );
con = null;
%>
```

ADO

Parole riservate (keyword)

- **si possono verificare errori se i nomi delle tabelle o delle colonne coincidono con keyword di ASP o del linguaggio di scripting usato (es. value)**
- **per evitare questi errori basta racchiudere il nome tra parentesi quadre (es. [value])**