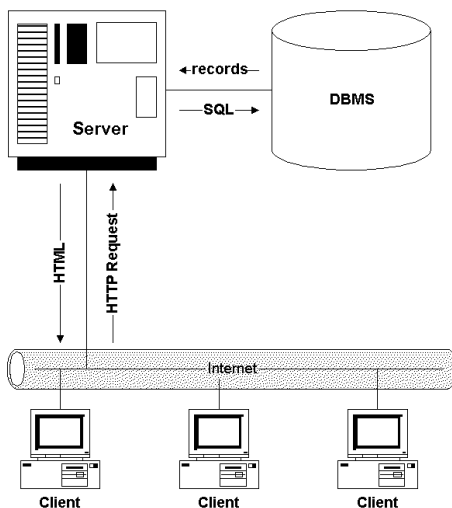


## Connessione e gestione DB in architettura C/S



In architettura C/S si prevede la possibilità di consultare, in base alle proprie esigenze, i contenuti di *data-base multimediali distribuiti* in Internet o semplicemente in Intranet.

Si analizzerà il caso di utilizzo di **oggetti ADO** che si integrano, lato server, alla tecnologia ASP.

- Oggetto [Connection](#)
- Oggetto [RecordSet](#)

### Connessione ad un database – Origine dei dati

Per connettere una pagina ad un database si possono usare diversi metodi, qui per brevità ne esamineremo soltanto i due principali a seconda del caso che si abbia o meno creato un **dsn** di sistema per il database.

### Connessione ad un database attraverso un DSN

Un DSN (Data Source Name cioè: "nome della sorgente dei dati" ) è un collegamento<sup>1</sup> al database con un nome scelto dall'utente.

Per effettuare tale collegamento occorre usare ODBC<sup>2</sup> (**Open Database Connectivity**) dal *pannello di controllo* del PC selezionando tra gli *Strumenti di amministrazione* l'opportuno percorso



Ad esempio, se si hanno i permessi, si scelga la Form **DSN di sistema** e l'opzione "Aggiungi". Si apre una maschera che chiede il tipo di base di dati e si crei il collegamento. Si selezioni "Driver da Microsoft Access (\*.mdb)" e si prosegua con "Fine". La maschera successiva chiede il nome del DSN (nome di origine dati). Si usi "dsn\_nomeDB" per ricordare dal nome che questo DSN è il collegamento al database *nomeDB.mdb*. Poi con click sul pulsante "Seleziona" si crei il collegamento al file "*nomeDB.mdb*" del database confermando poi con il pulsante "OK". Una volta creato il DSN, la stringa di connessione al database risulta brevissima (dsn\_nomeDB) ed ecco le poche righe da scrivere nel file ASP per collegarsi al database:

```
<%  
' creazione dell'oggetto Connection, lato Server  
  
Set DbConn = Server.CreateObject("ADODB.Connection")  
DbConn.Open "dsn_nomeDB"  
%>
```

Tale brevità è dovuta al fatto che **tipo di driver** e **percorso del database** sono stati già indicati nel file **dsn** configurato tramite il pannello di controllo. Questa soluzione potrebbe sembrare la più veloce ma in realtà si devono **creare file dsn per ogni database** che si utilizza.

<sup>1</sup> Con **dsn** si intende l' **insieme di informazioni identificate da un nome** che indicano ad un programma come connettersi ad una determinata fonte dati tramite un driver ODBC (Open DataBase Connectivity), quindi con dsn si fa riferimento al **path** di un file che sia una **fonte dati** ODBC.

<sup>2</sup> ODBC: componente Microsoft Windows ed in particolare di Windows Open Services Architecture. Tale API di Microsoft consente l'accesso a piattaforme DB da qualsiasi ambiente ed è standard universalmente diffuso per l'accesso ai dati in maniera trasparente, indipendentemente dalla piattaforma sulla quale risiedono. Una fonte dati ODBC è un **file** o archivio di **database** che consente l'accesso ai dati (consultazione e modifica) da parte di un qualsiasi programma Windows (videoscrittura, **foglio di calcolo**...). Gli archivi ODBC possono essere interrogati e modificati tramite il **linguaggio** SQL. Il programma (non database) che ne chiede la consultazione deve essere dotato di un apposito **driver** ODBC.

## Connessione ad un database senza un DSN (DSN-LESS)

Il codice risulta un pò allungato dato che in questo caso si dovrà specificare direttamente nella pagina il **percorso** del database ed il **tipo di driver** utilizzato:

```
<%  
Set OBJdbConnection = Server.CreateObject ("ADODB.Connection")  
OBJdbConnection.Open"driver={Microsoft Access Driver (*.mdb)}; dbq=" & server.mappath  
("\percorso\nomeDB.mdb")  
%>
```

In questo caso viene utilizzato un driver per Microsoft Access ed il database si trova nella sottocartella individuata dal *percorso* rispetto alla home page del sito web personale dato che il metodo **server.mappath** restituisce il percorso della home directory del sito web di default; tale percorso viene specificato durante l'installazione del server web. Se è noto, si può usare il percorso fisico<sup>3</sup> ad esempio:

```
<%  
Set OBJdbConnection = Server.CreateObject ("ADODB.Connection")  
OBJdbConnection.Open"driver={Microsoft Access Driver (*.mdb)}; dbq=c:\miosito\nomeDB.mdb"  
%>
```

Per maggior efficienza, se disponibile, si può far uso di **OLE-DB Provider<sup>4</sup> for Microsoft Jet** invece di DRIVER specifico

```
<%  
    OBJdbConnection.Open ("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & server.mappath  
    ("\percorso\nomeDB.mdb"))  
  
    ' Microsoft.Jet.OLEDB.x.y, dove x e y rappresentano la versione e la release del provider  
    ' nell'esempio la versione del provider per accedere ai database MS Access2000 4.0 e 2003  
%>
```

Istanziando un **oggetto Connection** si stabilisce la connessione e si possono aprire più **recordset** che prendono dati dalla connessione definita. L'**oggetto RecordSet** è un puntatore alla tabella creata al volo come risultato di un'interrogazione sulla base di dati.

```
<%  
Set RS = OBJdbConnection.execute(sSQL5)  
Set RS1 = OBJdbConnection.execute(sSQL)  
%>
```

In assenza di connessione aperta, ogni apertura del recordset creerebbe automaticamente una nuova connessione alla fonte dati

```
<%  
' creazione esplicita dell'oggetto Recordset  
  
Set RS = Server.CreateObject ("ADODB.RecordSet")  
  
' a connessione aperta, in sSQL memorizzata l'interrogazione in linguaggio SQL  
  
RS.open sSQL, OBJdbConnection  
%>
```

**E' importante ricordare una cosa:** se nel corso delle operazioni venisse modificata la struttura della fonte dati (ad esempio venisse creata una tabella o modificato il nome di un campo – recuperabile come *oggettoRecordSet.Fields(chiave)* - di un database) sarebbe necessario chiudere e riaprire la connessione alla fonte dati.

<sup>3</sup> Con web account free non si conosce il percorso fisico.

<sup>4</sup> Il termine **provider** indica in generale un elemento fornitore dal quale si ottengono servizi o dati.

<sup>5</sup> Comando SQL. Si ricordi che il linguaggio SQL permette due tipi di query:

- interrogazione **statica** (tipo *compile and store*)
- interrogazione **dinamica** (tipo *compile and go*) costruita in modo interattivo (visuale) come stringa ed eseguita “on – fly”.