I DBMS sono ... efficienti

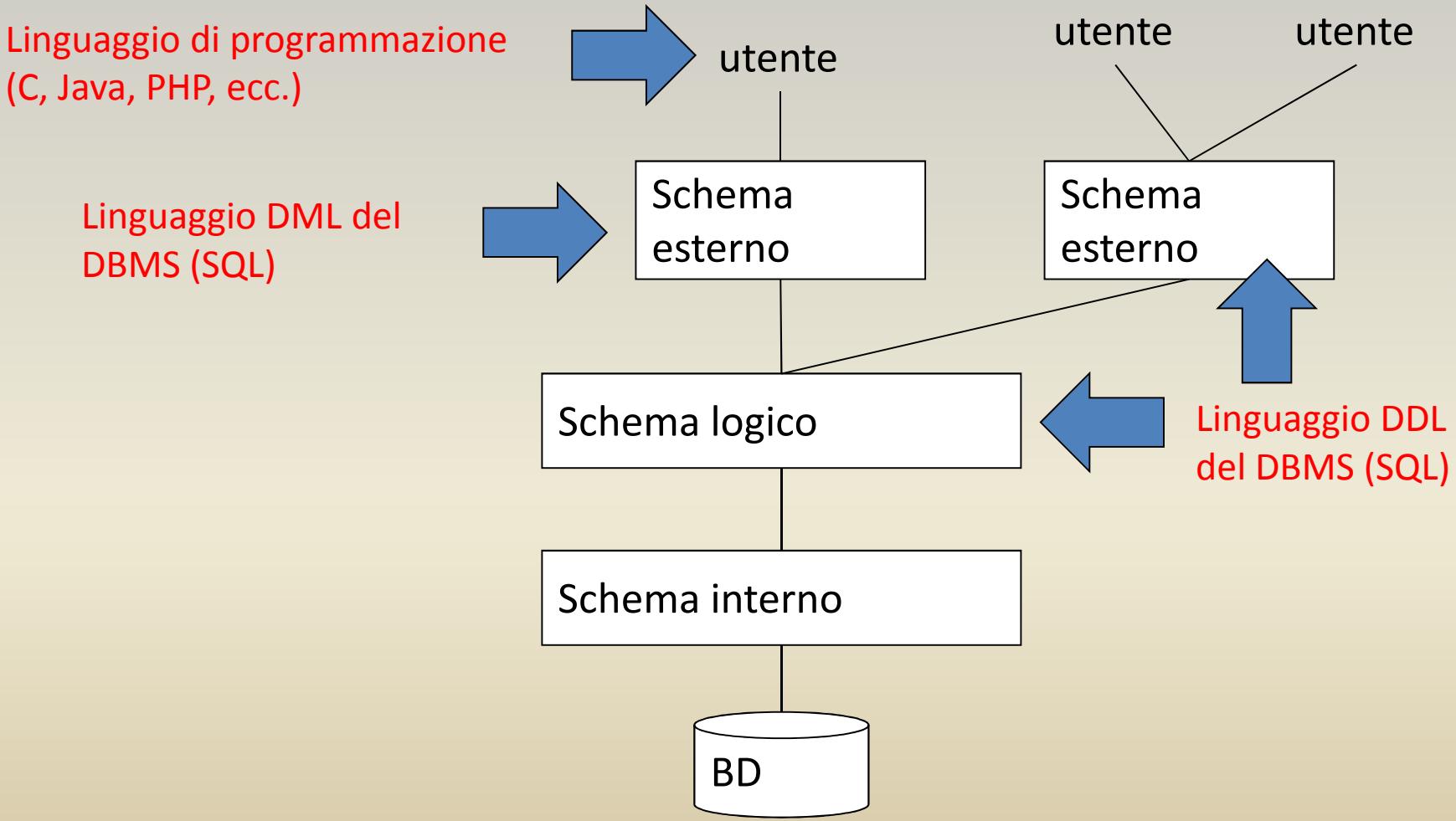
- Cercano di utilizzare al meglio le risorse di spazio di memoria (principale e secondaria) e tempo (**di esecuzione e di risposta**)
- Tecniche specializzate per l'implementazioni dei DBMS con investimenti e competizione

Perché (non) utilizzare MS Access?

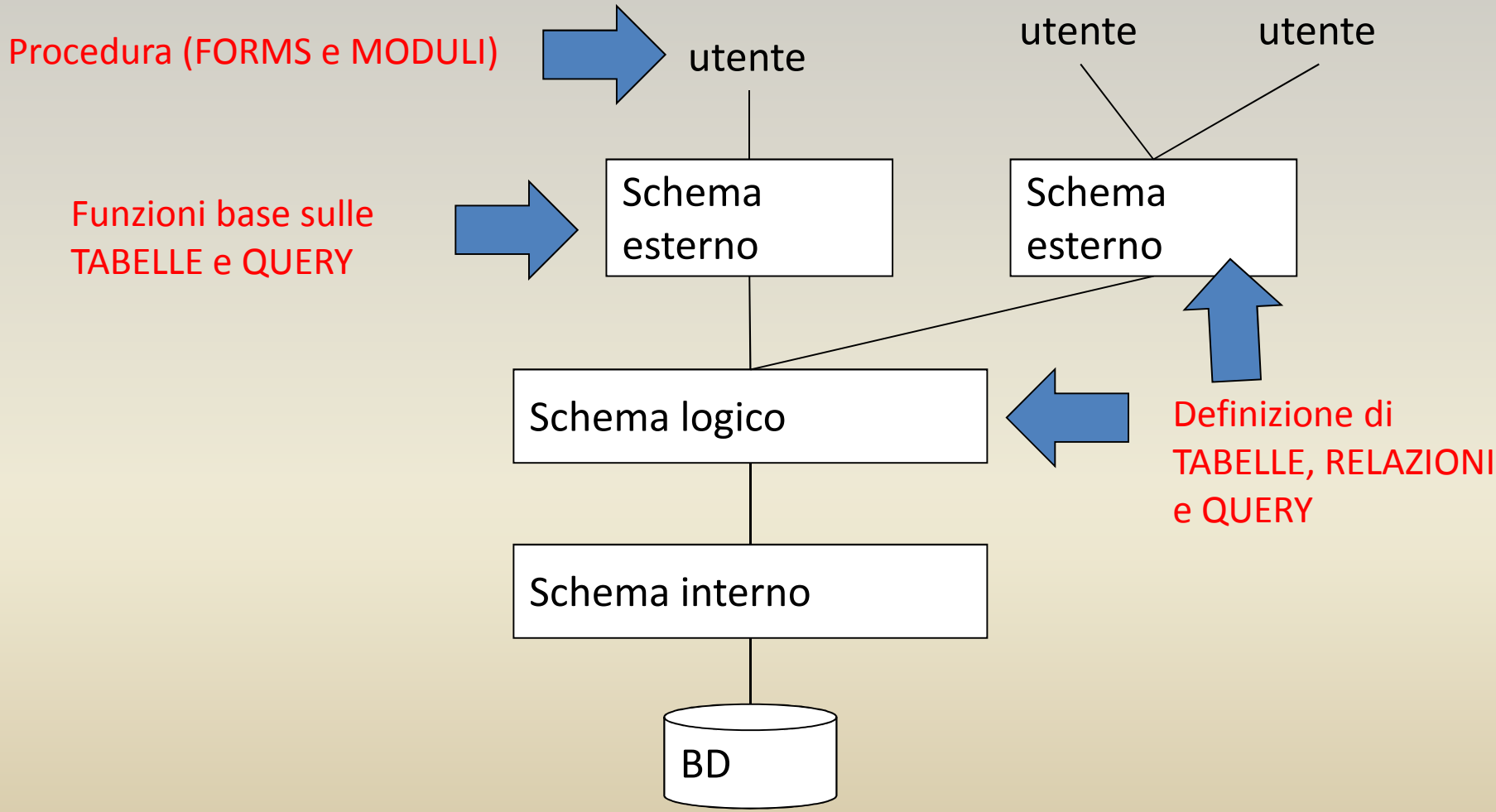
Vantaggi:
• potete creare facilmente il database utilizzando MS Office dal vostro pc, disponendo di una interfaccia facile e intuitiva
• oltre a dover configurare i permessi di scrittura sulla cartella che ospita il db e sul db stesso non avete bisogno di altro
• potete effettuare un back up del database quando desiderate semplicemente copiando il file .MDB

Svantaggi:
• limitazione dichiarata da Microsoft sul numero di connessioni contemporanee al database, che è di 15 (teorico). In pratica con 3 siamo al limite.
• lentezza delle prestazioni in caso di “molti” record oppure multiutenza.

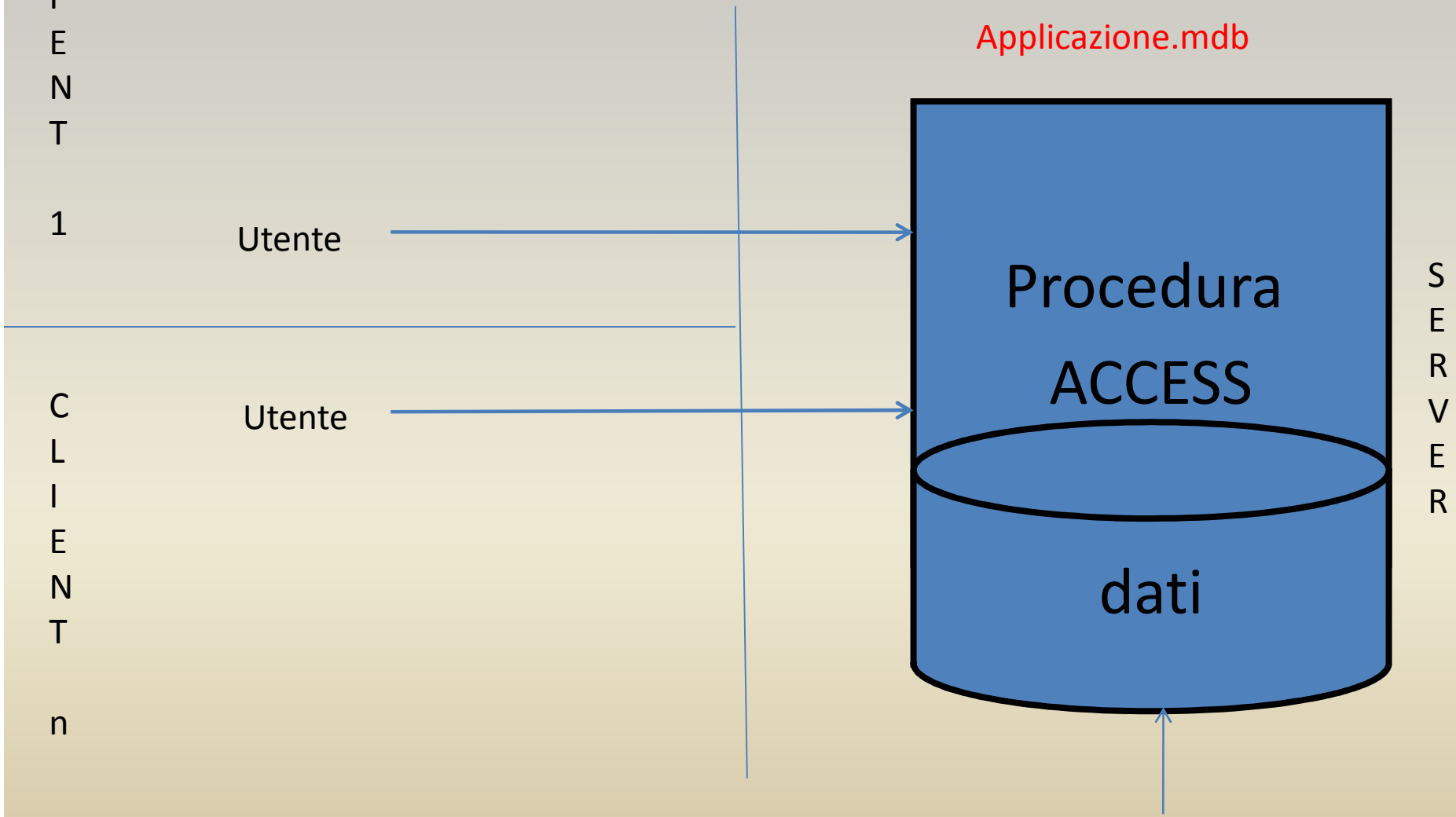
Architettura standard (ANSI/SPARC) a tre livelli per DBMS



Architettura standard (ANSI/SPARC) MS-ACCESS

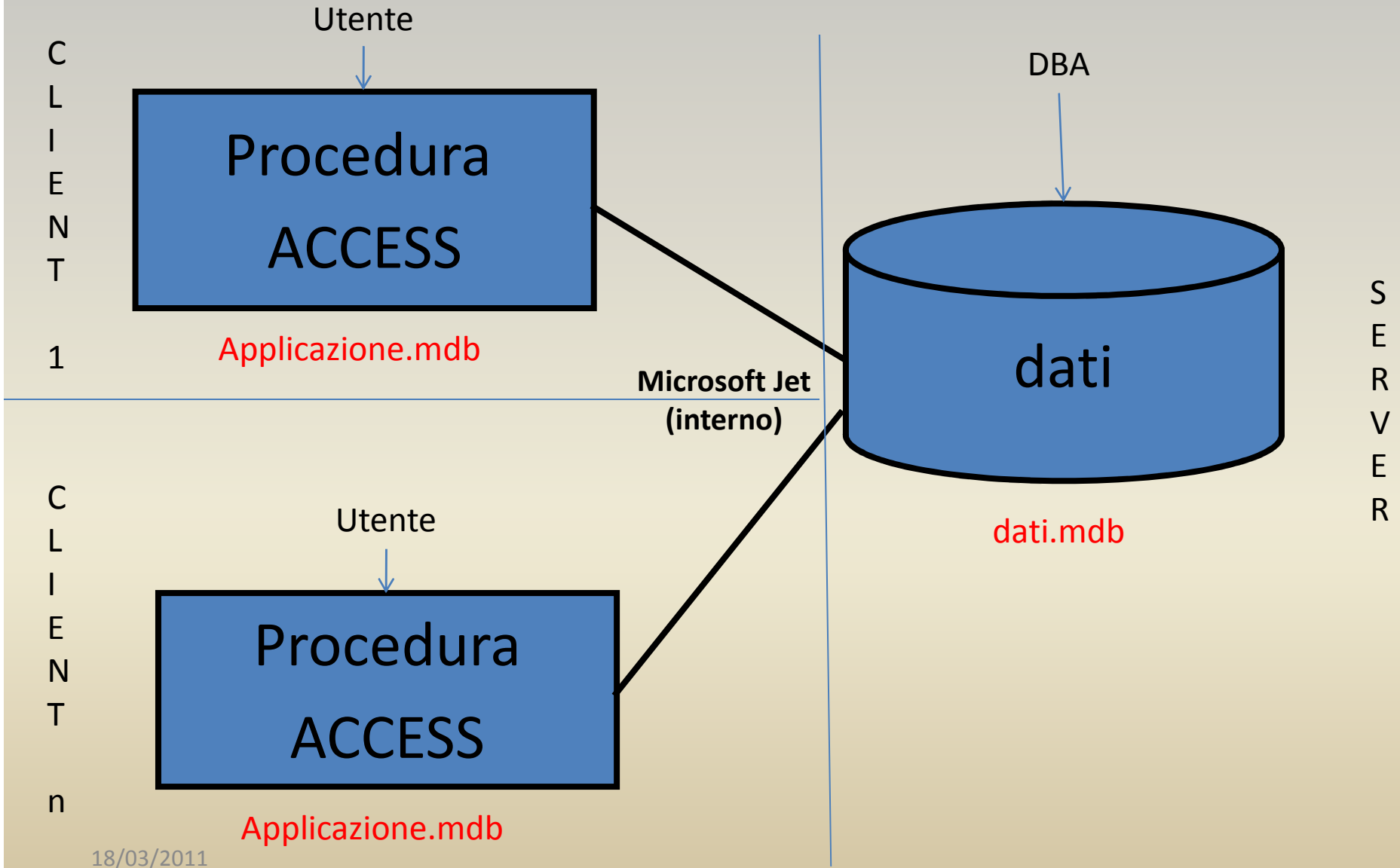


MULTIUTENZA

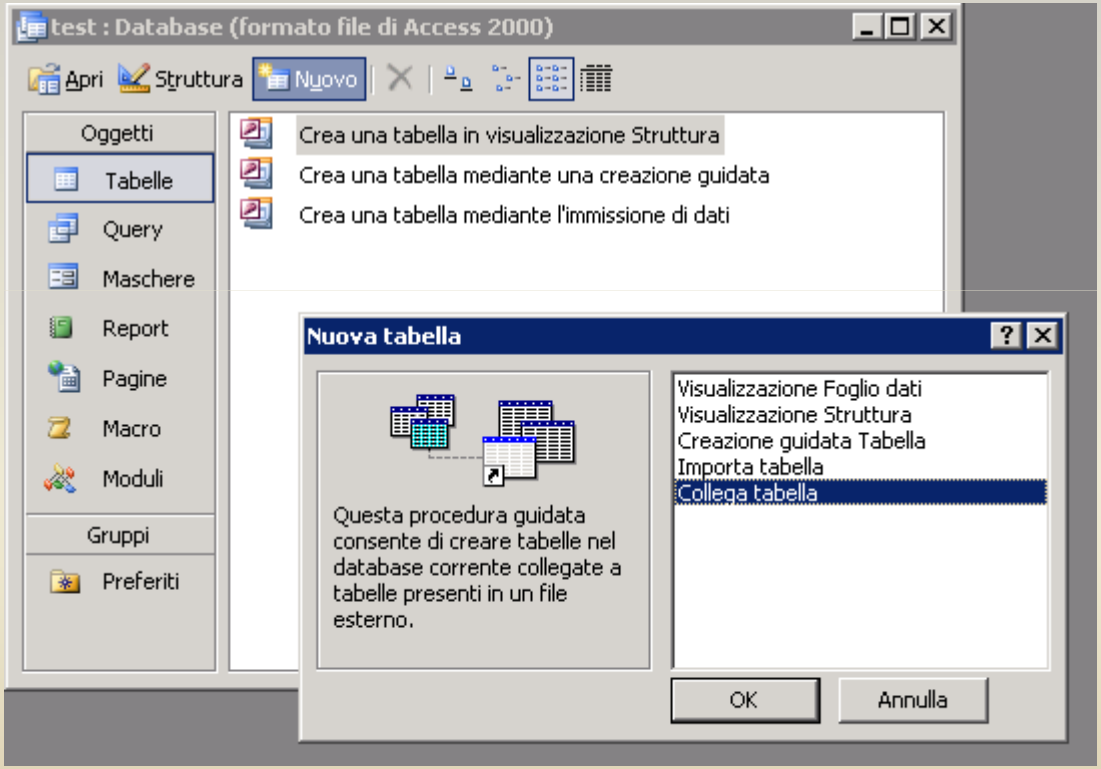


(altamente inefficiente)

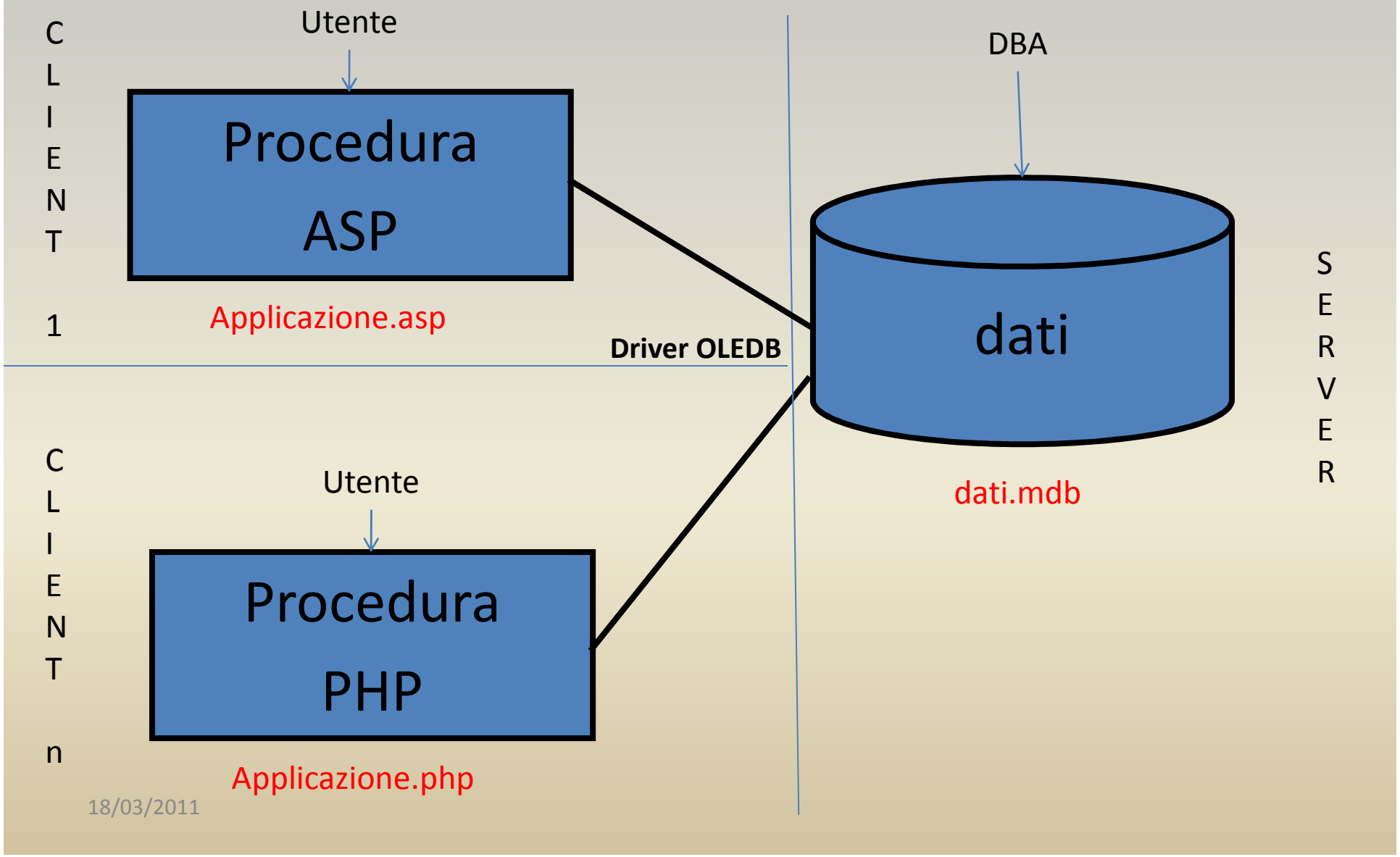
Uso di MS Access (DBMS + Appl.)



Uso di MS Access (DBMS + Appl.)



Uso di MS Access (solo DBMS)



Esempi di connessione ad un DB ACCESS

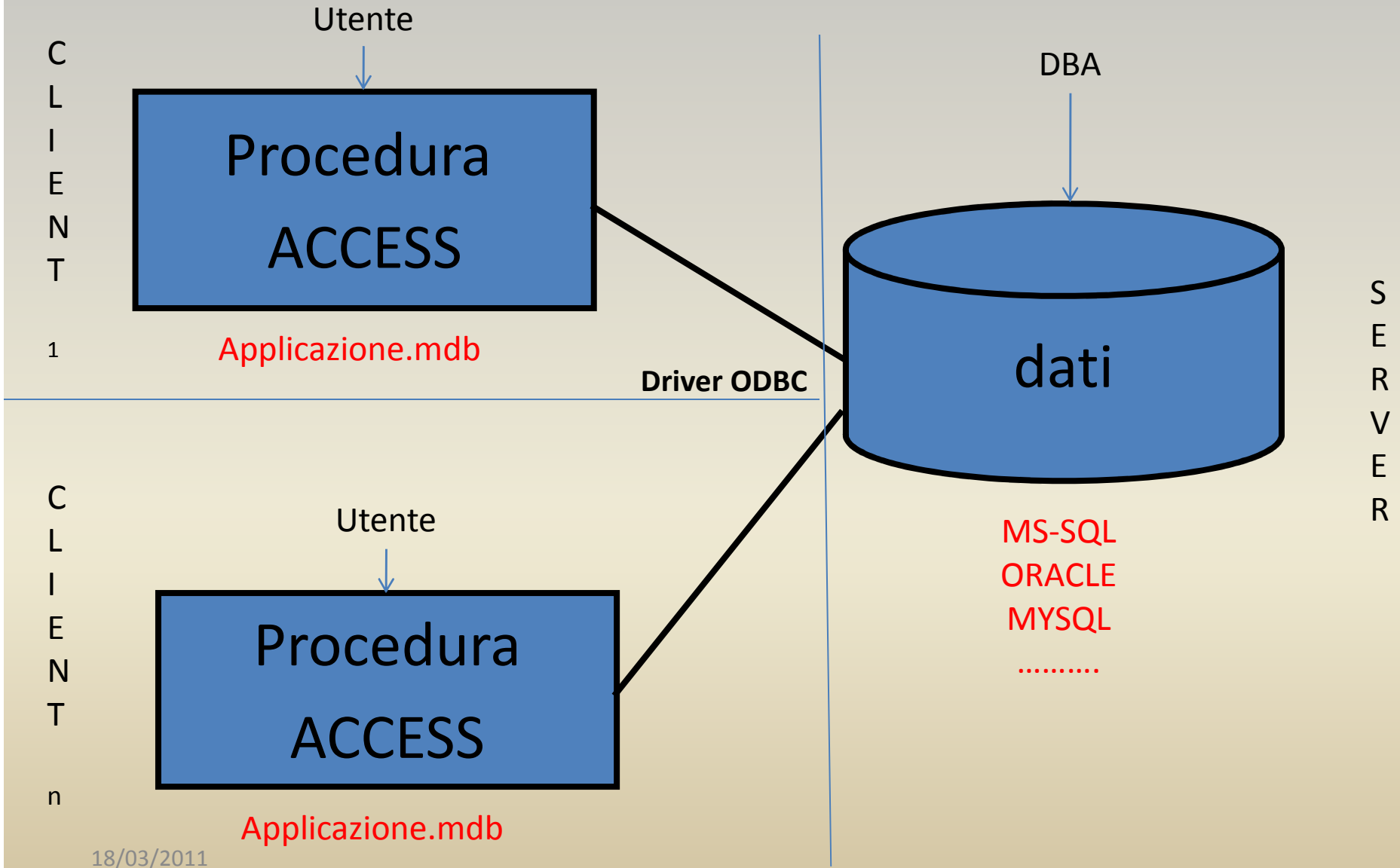
ASP

```
<%
Set conn = Server.CreateObject("ADODB.Connection")
conn.Open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" &
server.MapPath("/database/esempio.mdb")
sql = "SELECT * FROM Tabella1"
Set rs = Server.CreateObject("ADODB.Recordset")
rs.Open sql, conn
%>
```

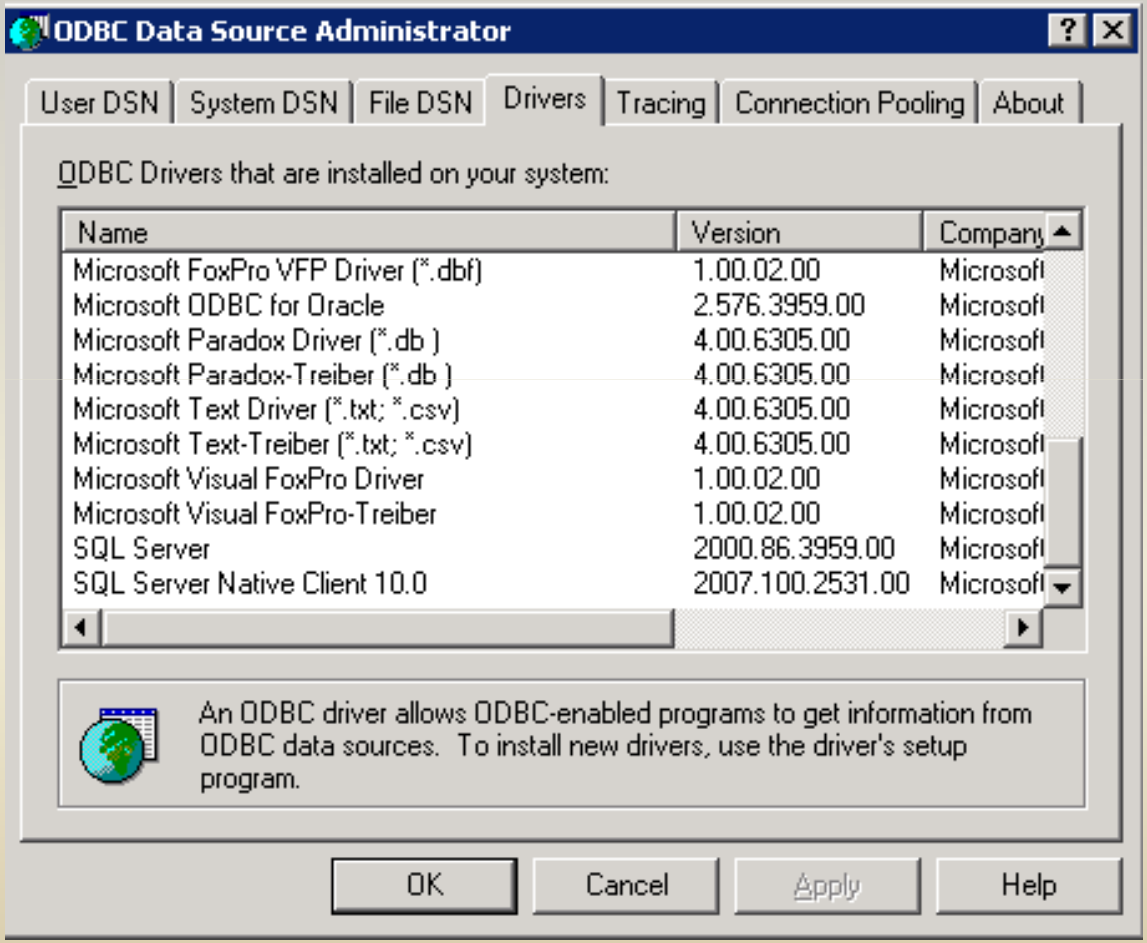
PHP

```
$connessione = new COM("ADODB.Connection");
$stringa_di_connessione = "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=".
realpath("testdb/testdb.mdb");
$connessione->Open($stringa_di_connessione);
$interrogazione = "select * from categorie";
$recordset = new COM("ADODB.Recordset");
```

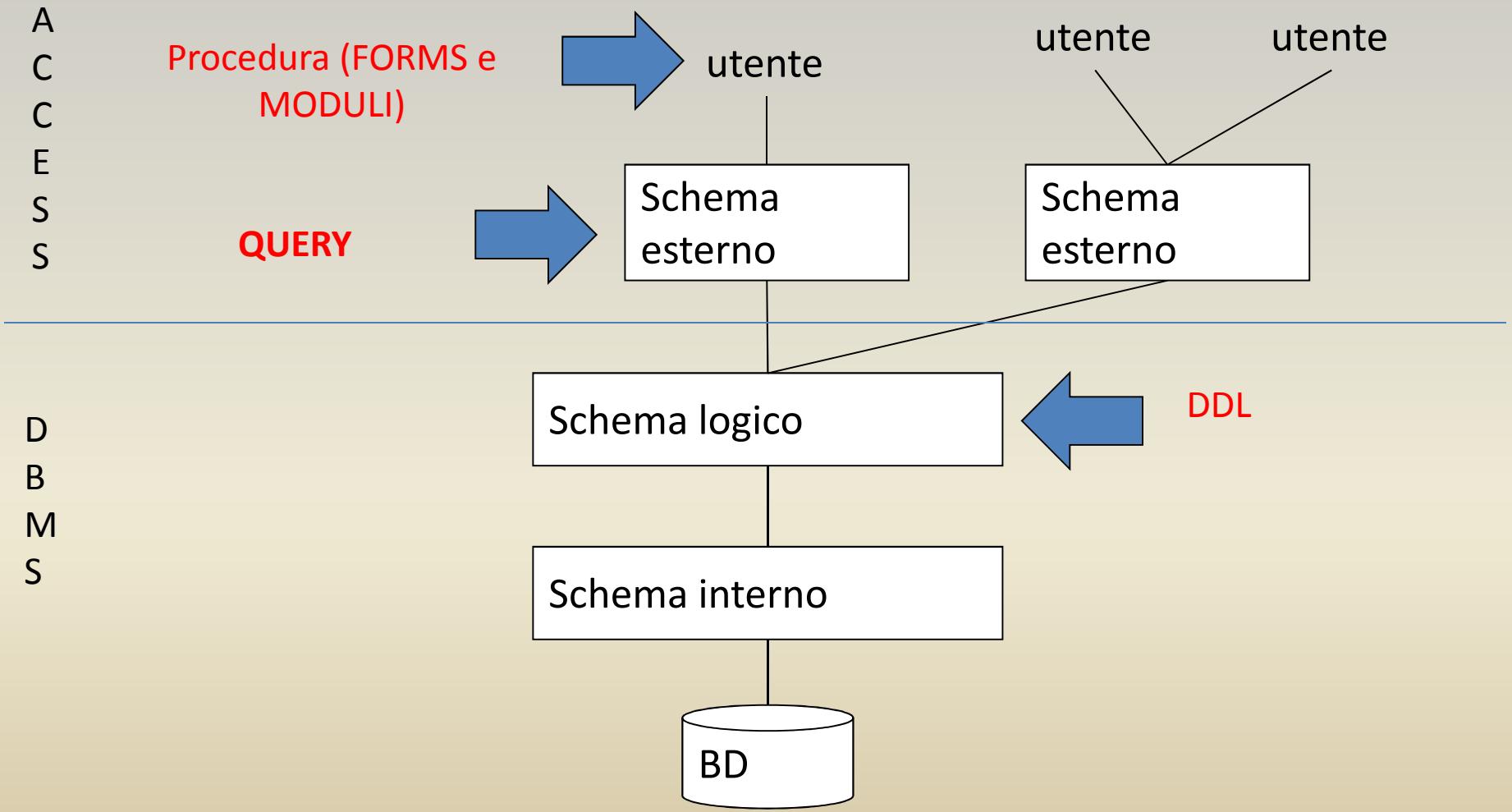
Uso di MS Access (Appl.)



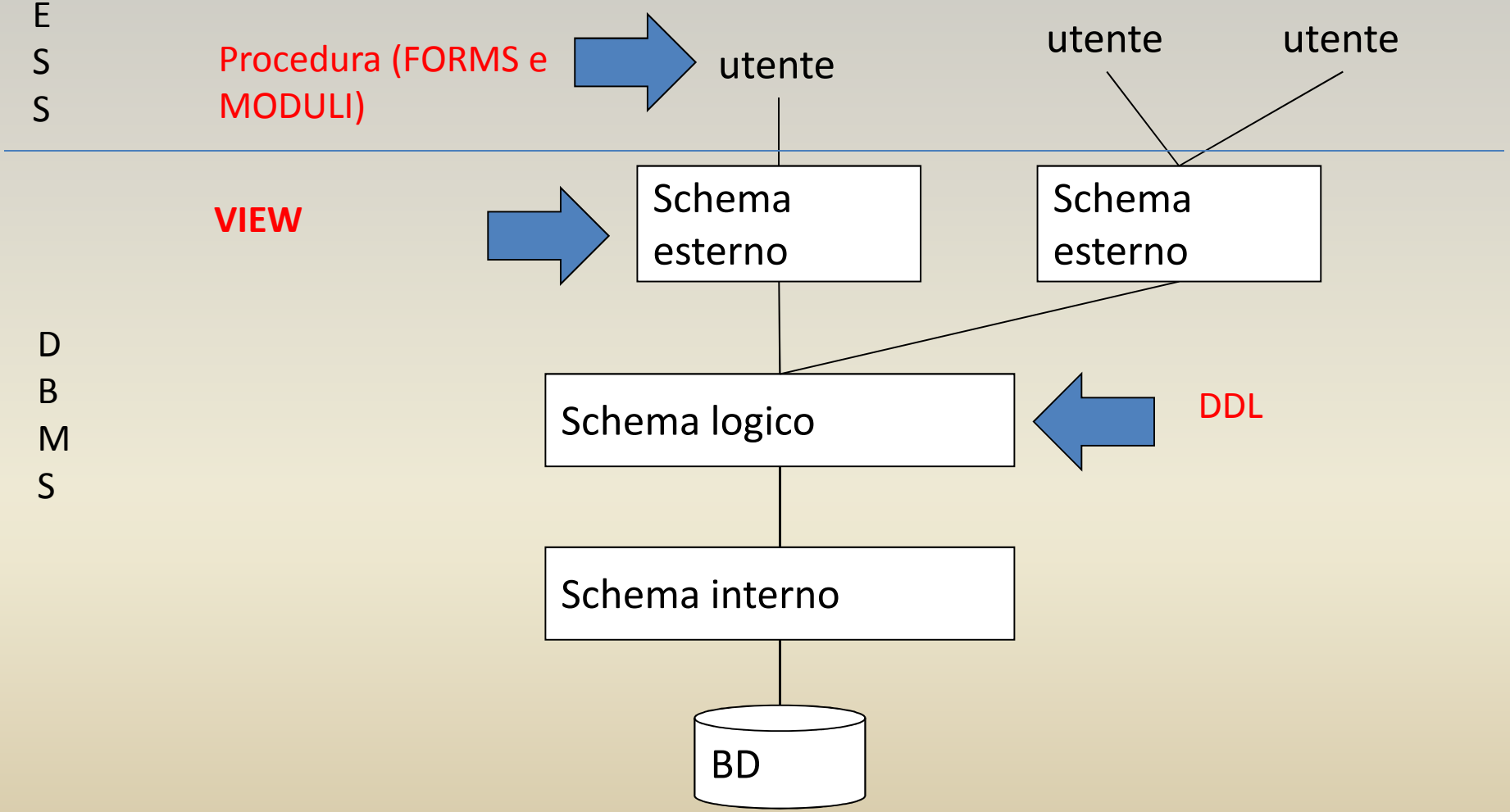
Uso di MS Access (Appl.)

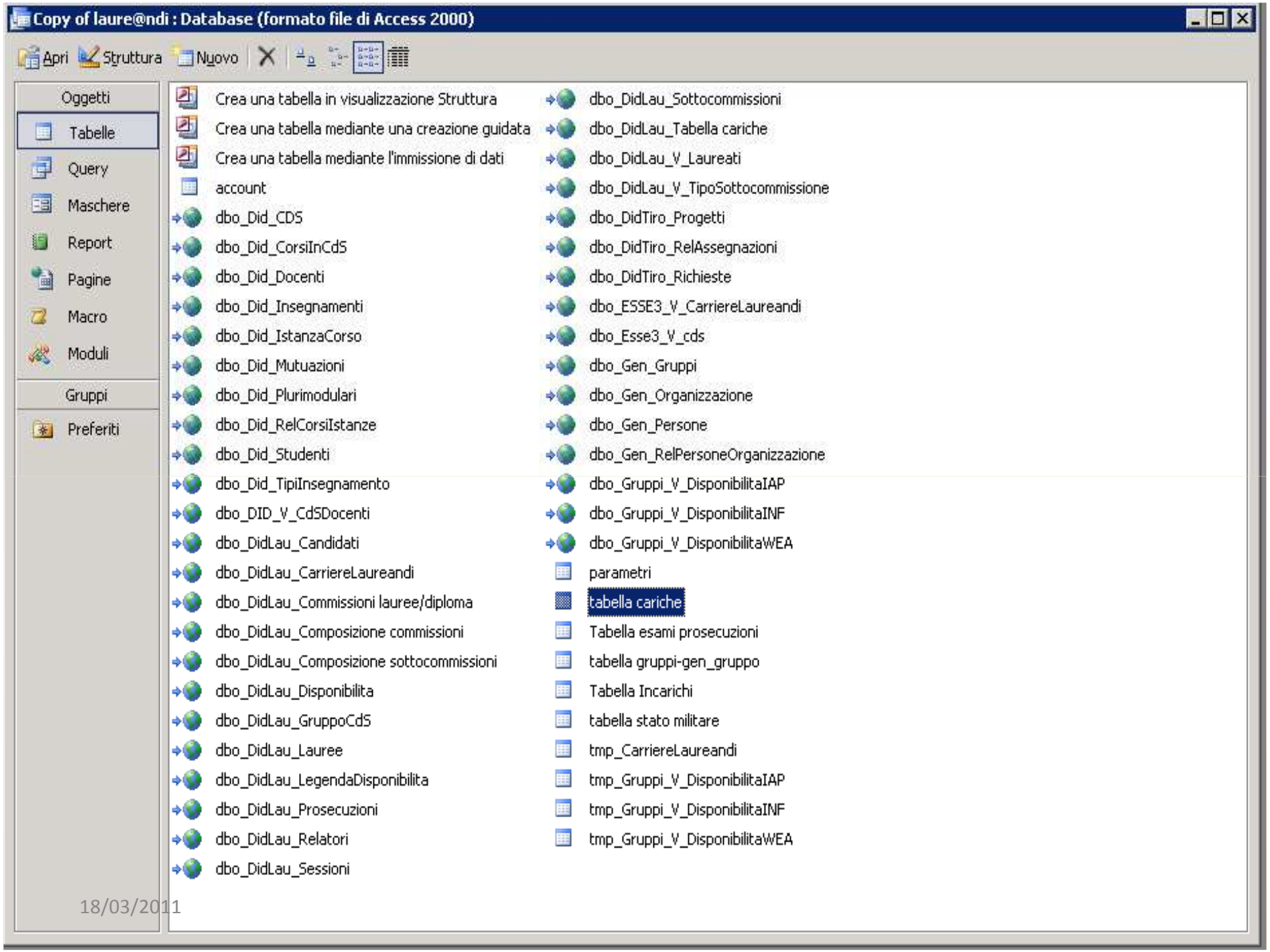


MS-ACCESS Appl.
(soluzione 1 – meno efficiente)



MS-ACCESS Appl. (soluzione 2 – più efficiente)




Copy of laure@ndi : Database (formato file di Access 2000)
Apri
Struttura
Nuovo
Oggetti
Tabelle
Query
Maschere
Report
Pagine
Macro
Moduli
Gruppi
Preferiti
Crea una tabella in visualizzazione Struttura
Crea una tabella mediante una creazione guidata
Crea una tabella mediante l'immissione di dati
account
dbo_Did_CDS
dbo_Did_CorsiInCdS
dbo_Did_Docenti
dbo_Did_Insegnamenti
dbo_Did_IstanzaCorso
dbo_Did_Mutuazioni
dbo_Did_Plurimodulari
dbo_Did_RelCorsiIstanze
dbo_Did_Studenti
dbo_Did_TipiInsegnamento
dbo_DID_V_CdSDocenti
dbo_DidLau_Candidati
dbo_DidLau_CarriereLaureandi
dbo_DidLau_Commissioni lauree/diploma
dbo_DidLau_Composizione commissioni
dbo_DidLau_Composizione sottocommissioni
dbo_DidLau_Disponibilita
dbo_DidLau_GruppoCdS
dbo_DidLau_Lauree
dbo_DidLau_LegendaDisponibilita
dbo_DidLau_Prosecuzioni
dbo_DidLau_Relatori
dbo_DidLau_Sessioni
dbo_DidLau_Sottocommissioni
dbo_DidLau_Tabella cariche
dbo_DidLau_V_Laureati
dbo_DidLau_V_TipoSottocommissione
dbo_DidTiro_Progetti
dbo_DidTiro_RelAssegnazioni
dbo_DidTiro_Richieste
dbo_ESSE3_V_CarriereLaureandi
dbo_Esse3_V_cds
dbo_Gen_Gruppi
dbo_Gen_Organizzazione
dbo_Gen_Persone
dbo_Gen_RelPersoneOrganizzazione
dbo_Gruppi_V_DisponibilitaIAP
dbo_Gruppi_V_DisponibilitaINF
dbo_Gruppi_V_DisponibilitaWEA
parametri
tabella cariche
Tabella esami prosecuzioni
tabella gruppi-gen_gruppo
Tabella Incarichi
tabella stato militare
tmp_CarriereLaureandi
tmp_Gruppi_V_DisponibilitaIAP
tmp_Gruppi_V_DisponibilitaINF
tmp_Gruppi_V_DisponibilitaWEA

Operazioni standard sui dati

- **Inserimento**
- **Ricerca**
- **Modifica**
- **Cancellazione**
- **Selezione**

USO DEGLI INDICI

Esempio: supponiamo di cercare un nome fra i seguenti

Giuseppe
Irene
Fabio
Carlo
Lola
Elena
Fabio

la procedura da fare è:

1. leggi il primo nome della lista
2. se nome trovato fermati
3. leggi prossimo nome torna a 2

Nel caso migliore leggo 1 volta. Nel caso peggiore leggo 7 volte.
Mediamente leggo $\frac{1+7}{2}$ = 4 volte

Se il dato non è presente si casca sempre nel caso peggiore.

USO DEGLI INDICI

Supponiamo ora di organizzare l'elenco precedente in ordine alfabetico:

Carlo
Elena
Fabio
Fabio
Giuseppe
Irene
Lola

Se la procedura è la solita:

1. leggi il primo nome della lista
2. se nome trovato fermati
3. leggi prossimo nome torna a 2

Si ha lo stesso numero di letture. Cambia però il caso del dato non presente che rientra nelle letture medie.

USO DEGLI INDICI

Supponiamo ora cambiare la procedura

Carlo
Elena
Fabio
Fabio
Giuseppe
Irene
Lola

La nuova procedura è:
1 leggi nome centrale
2. se nome trovato fermati
3. se nome > cercato considera la metà superiore del file
4. se nome < cercato considera la metà inferiore del file
5. vai a 1

Nel caso migliore leggo 1 volta. Nel caso peggiore leggo 3 volte.
Mediamente leggo $\frac{1+3}{2}$ = 2 volte

USO DEGLI INDICI

PROBLEMA:

Il caso si presenta più complicato quando aumenta il numero dei campi e il numero delle ricerche possibili.

1)	Giuseppe	Rossi
2)	Irene	Bianchi
3)	Fabio	Verdi
4)	Carlo	Bianchi
5)	Lola	Neri
6)	Elena	Bianchi
7)	Fabio	Rossi

Se si vuol ricercare sia per nome che per cognome non si può dare un'organizzazione unica ai dati e non si può neanche riordinarli prima di ciascuna ricerca.

USO DEGLI INDICI

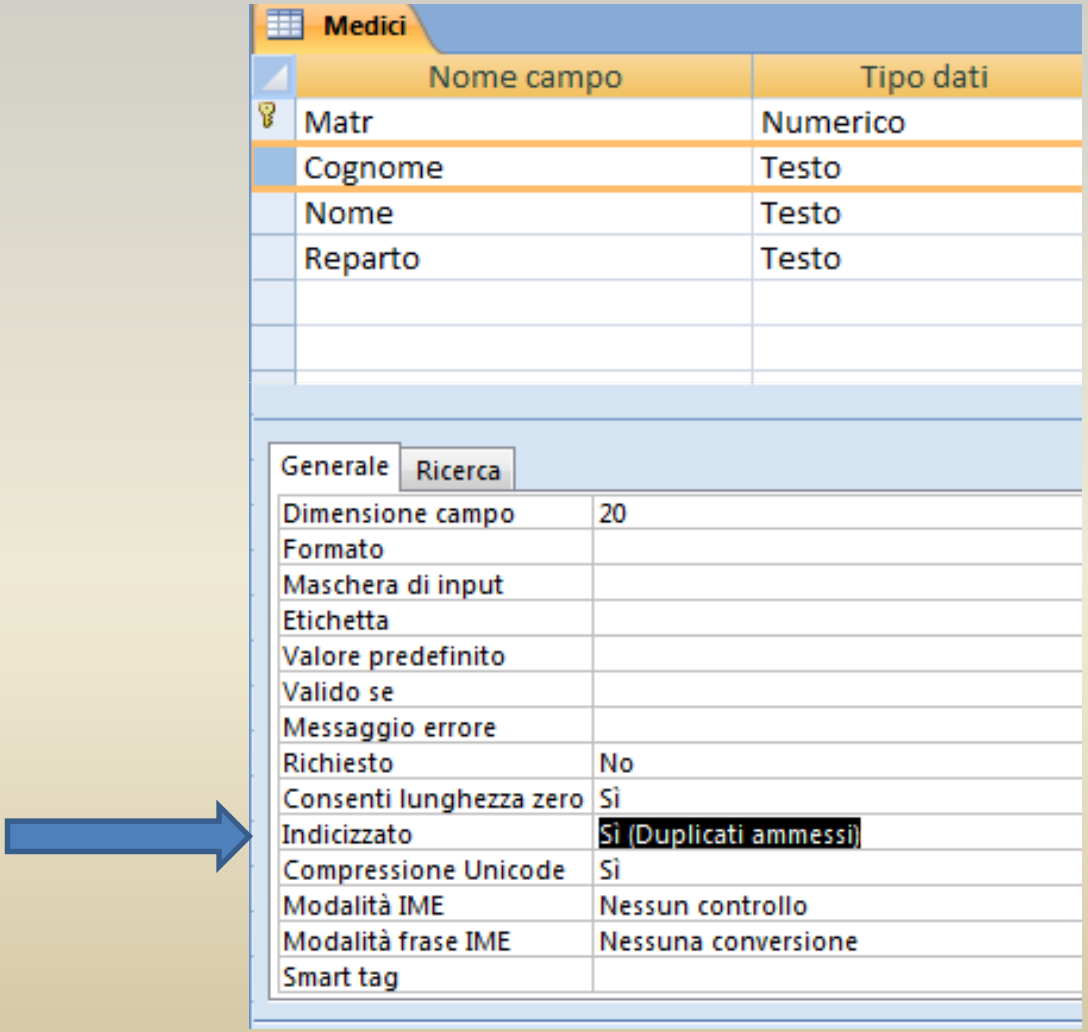
SOLUZIONE:

Si usano degli *indici*: archivi a parte nei quali si ha il riepilogo dei dati organizzati e la maniera di recuperare l'intero record nell'archivio principale che rimane inalterato

Nome	
Carlo	4
Elena	6
Fabio	3, 7
Giuseppe	1
Irene	2
Lola	5

Cognome	
Bianchi	2, 4, 6
Neri	5
Rossi	1, 7
Verdi	3

In ACCESS (struttura tabella)



SQL

- **DDL**
  - "Data Definition Language"
  - definizione degli oggetti dello schema
  - CREATE DATABASE
  - DROP DATABASE
  - CREATE TABLE
  - DROP TABLE
- **DCL**
  - "Data Control Language"
  - utenti e autorizzazioni
- **DML**
  - "Data Manipulation Language"
  - interrogazioni e aggiornamenti
  - INSERT, DELETE, UPDATE
  - SELECT

SQL - DDL

Esercizio 4.3

Dare le definizioni SQL delle tre tabelle

FONDISTA(Nome, Nazione, Età)
GAREGGIA(NomeFondista, NomeGara, Piazzamento)
GARA(Nome, Luogo, Nazione, Lunghezza)

rappresentando in particolare i vincoli di foreign key della tabella GAREGGIA.

Soluzione:

```
Create Table FONDISTA
(
Nome character(20) primary key,
Nazione character(30),
Età smallint
)
Create table GARA
(
Nome character(20) primary key,
Luogo character(20),
Nazione character(20),
Lunghezza integer
)
Create table GAREGGIA
(
NomeFondista character(20) references FONDISTA(Nome),
NomeGara character(20),
Piazzamento smallint,
primary key (NomeFondista, NomeGara),
foreign key (NomeGara) references GARA(Nome)
)
```

SQL - DDL

Esercizio 4.4

Dare le definizioni SQL delle tabelle

AUTORE (Nome, Cognome, DataNascita, Nazionalità)

LIBRO (TitoloLibro, NomeAutore, CognomeAutore, Lingua)
Per il vincolo *foreign key* specificare una politica di *cascade* sulla cancellazione e di *set null* sulle modifiche.

Soluzione:

```
Create table AUTORE
(
Nome character(20),
Cognome character(20),
DataNascita date,
Nazionalità character(20),
primary key(Nome, Cognome)
)
Create table LIBRO
(
TitoloLibro character(30) primary key,
NomeAutore character(20),
CognomeAutore character(20),
Lingua character(20),
foreign key (NomeAutore, CognomeAutore)
references AUTORE(Nome, Cognome)
on delete cascade
on update set NULL
)
```

Esercizio 4.5

Dato lo schema dell'esercizio precedente, spiegare cosa può capitare con l'esecuzione dei seguenti comandi di aggiornamento:

```
delete from AUTORE where Cognome = 'Rossi'

update LIBRO set NomeAutore= 'Umberto'  where CognomeAutore = 'Eco'

insert into AUTORE(Nome,Cognome) values('Antonio','Bianchi')

update AUTORE set Nome = 'Italo' where Cognome = 'Calvino'
```

Soluzione:

1. Il comando cancella dalla tabella AUTORE tutte le tuple con Cognome = 'Rossi'. A causa della politica cascade anche le tuple di LIBRO con CognomeAutore = 'Rossi' verranno eliminate.
2. Il comando non è corretto: Nome e Cognome sono attributi della tabella AUTORE e non della tabella LIBRO.
3. Il comando aggiunge una nuova tupla alla tabella AUTORE. Non ha alcun effetto sulla tabella LIBRO
4. Le tuple di AUTORE con Cognome = Calvino vengono aggiornate a Nome = Italo. A causa della politica set null gli attributi NomeAutore e CognomeAutore delle tuple di Libro con CognomeAutore = Calvino vengono posti a NULL.

Esercizio 4.7

Con riferimento ad una relazione PROFESSORI(CF, Nome, Età, Qualifica), scrivere le interrogazioni SQL che calcolano l'età media dei professori di ciascuna qualifica, nei due casi seguenti:

1. se l'età non è nota si usa per essa il valore nullo
2. se l'età non è nota si usa per essa il valore 0.

Soluzione:

1. Le funzioni aggregative escludono dalla valutazione le ennuple con valori nulli:

```
select Qualifica, avg(Eta) as EtaMedia
from Professori
group by Qualifica
```

2. E' necessario escludere esplicitamente dal calcolo della media le ennuple con il valore che denota l'informazione incompleta:

```
select Qualifica, avg(Eta) as EtaMedia
from Professori
where Eta <> 0
group by Qualifica.
```

Esercizio 4.9 (modificato)

Considerare le relazioni

IMPIEGATI (Matricola, Nome, Stipendio)
DIPARTIMENTI (Codice, Direttore)

e le due interrogazioni seguenti, specificare se e in quali casi esse possono produrre risultati diversi:

1. select avg(Stipendio) from Impiegati
where Matricola in (select Direttore from Dipartimenti)

2. select avg(Stipendio) from Impiegati I, Dipartimenti D
where I.Matricola = D.Direttore .

Soluzione:

Le due interrogazioni dovrebbero calcolare lo stipendio medio di un direttore di dipartimento.

Nel caso in cui esistono impiegati direttori di più dipartimenti la seconda query produce un risultato scorretto in quanto conta più occorrenze di questi impiegati nel computo della media.

La prima query produce il risultato corretto.