

## SOLUZIONE

La traccia propone una situazione di connessione client server gestibile via Internet con accorgimenti semplici

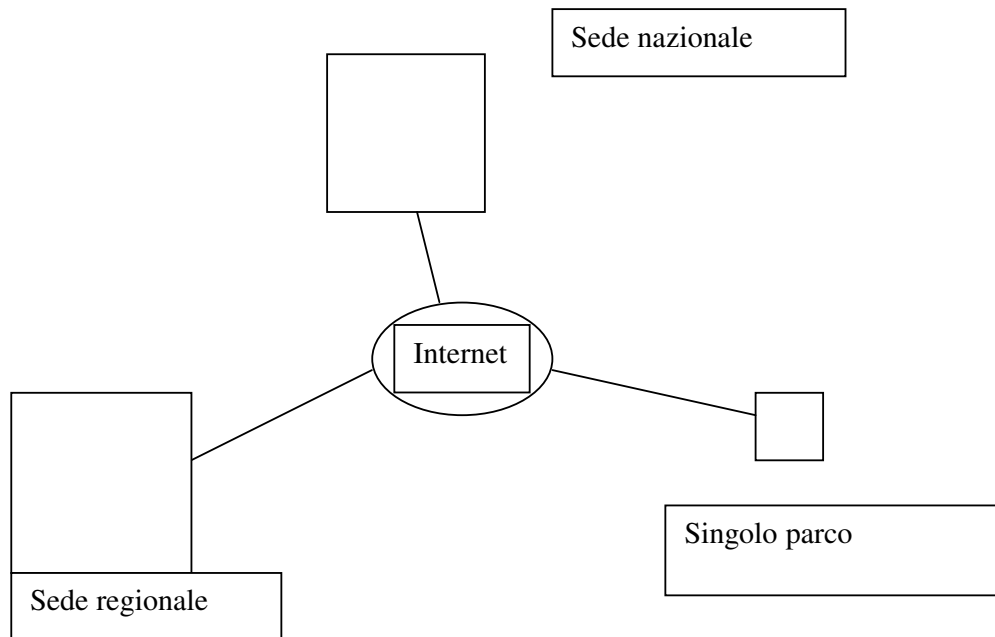
Sono identificabili tre ambiti, che chiameremo macro nodi.

Un nodo è rappresentato dalla postazione del parco: la realtà più semplice.

Un altro nodo, più complesso, è rappresentato dalla sede di raccordo regionale, collettore di dati e instradatore verso il server nazionale.

Terzo nodo è il centro nazionale, verso il quale convergono i vari nodi regionali.

In definitiva avremo tanti parchi connessi ai vari centri regionali. Questi ultimi, saranno connessi all'unico centro nazionale.



Consideriamo la postazione locale (Notebook)

Considerando il tipo di apparecchiatura, destinata ad un uso anche itinerante, e tenendo poi conto che è ormai consuetudine dotare questo tipo di computer di connessioni wi-fi, ipotizziamo di avvalerci di tale tipo di connessione.

L'accesso alla rete locale dovrebbe quindi avvenire con un access point, lasciando quello ad Internet alle funzionalità di un router di classe entry level, definito "router di Internet".

Nell'organizzazione logica della rete, il ruolo di questa postazione locale è tipicamente client.

Da ciò derivano alcune considerazioni:

- L'accesso ad Internet può essere effettuato tramite provider.
- Non occorre registrare il Notebook con un indirizzo pubblico.
- L'accesso tramite configurazione DHCP è sufficiente.
- Il computer remoto, che chiameremo Server, dovrà certamente appartenere ad un dominio, nell'ambito del quale dovrà essere previsto un account per il client.
- L'account implica la predisposizione di user name e PW.

Passiamo ai centri remoti.

Da un punto di vista strutturale, ma anche funzionale, le due realtà sono equivalenti, mentre varia la complessità dei ruoli.

Esploriamo le caratteristiche del nodo regionale.

Il centro dovrà avere alcune caratteristiche strettamente legate al ruolo.

Tralasciando considerazioni sulla struttura della macchina server, preferiamo soffermarci sulla infrastruttura di rete.

Certamente si tratterà di una rete più complessa di quella del Notebook.

Potremo pensare ad un'unica struttura a stella con sezionamento asimmetrico, per riservare porte veloci dello Switch ai Servers ed al router.

Quest'ultimo non dovrebbe essere un entry level. Certamente la complessità di questo componente è vincolante, e va a favore di due aspetti non trascurabili: la gestibilità della rete e la sua sicurezza.

Passando all'organizzazione dell'infrastruttura, preferiamo parlare in termini di ruoli, piuttosto che di macchine.

Occorrerà un server di account, un server web e un server di DB. . Queste tre funzioni potrebbero sommarsi su una stessa macchina.

Prevedere poi un server di backup potrebbe essere una buona ed auspicabile scelta. In termini di organizzazione logica della rete nella sede regionale, dobbiamo pensare ad indirizzi pubblici per la porta del router verso Internet e per i server, per questioni di raggiungibilità e visibilità

La velocità della connessione deve essere un multiplo di quella con la quale i clients accedono.

Analoghe considerazioni valgono per il centro nazionale.

Per entrambe le realtà sono prevedibili anche postazioni di lavoro che trascuriamo.

Nodale è, come già si diceva, il ruolo dei routers. Spetta a loro, con regole d'accesso e operazioni di NAT e PAT ( che non approfondiamo considerando la tipologia della prova), proteggere rendendo però visibili, i vari servers.

Considerazioni sul sw da predisporre sulle macchine.

Sul Server abbiamo due possibili scelte: Window e Linux.

Dovendosi condividere un DB, considerando la connessione remota via WEB, si può optare per la seconda alternativa. Affidando alla duttilità di Apache il ruolo di ospite dell'ambiente web.

Rinviamo il discorso sul DBMS paragrafi successivi di questa soluzione.

Alcune considerazioni sulla sicurezza.

Dato che il database non prevede l'utilizzo di dati particolarmente sensibili e non interagisce con strutture più complesse possiamo evitare di prendere in considerazione strategie particolari che vadano oltre l'uso di Userid e Password.

## Ipotesi aggiuntive

Dopo un'attenta lettura della traccia appare sensato optare per una soluzione che preveda la presenza di un database locale installato con il software su ciascun notebook in dotazione ai responsabili dei vari parchi, la cui struttura verrà repplicata sul database centrale presente sul server della Regione. Per rendere tra loro compatibili le tabelle dei due database sarà sufficiente prevedere nelle tabelle degli esemplari della fauna e della flora un campo Parco che, in ambiente regionale, servirà per individuare i dati di ciascun parco, e in ambiente locale verrà presentato in automaytico dal software nelle varie maschere di acquisizione dati.

Le entità fondamentali che emergono dalle richieste della traccia sono ESEMPLARE ANIMALE e ESEMPLARE VEGETALE.

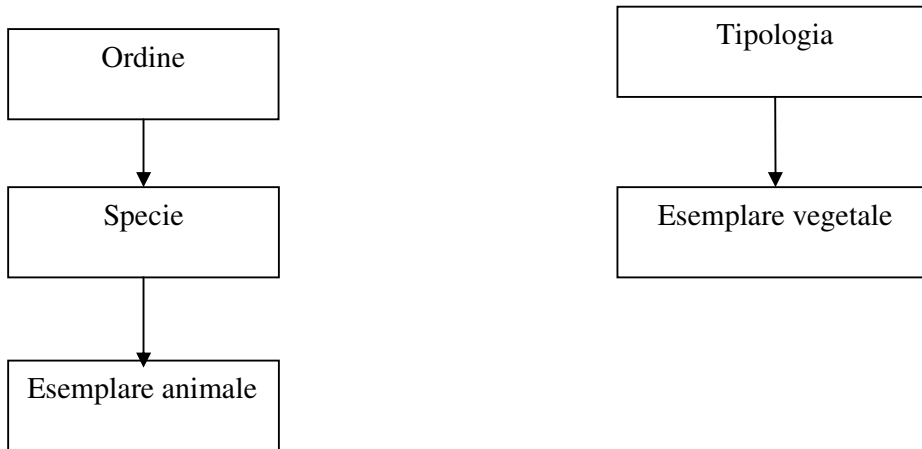
Nulla vieterebbe di realizzare il database utilizzando solo queste.

Per evitare inutili duplicazioni, però, risulta conveniente prevedere a monte di esse le seguenti entità:

SPECIE ANIMALE  
ORDINE ANIMALE

TIPOLOGIA VEGETALE

Lo schema E/R del database risulta quindi essere il seguente:



Tutte le relazioni esistenti tra le varie entità sono di tipo 1:n e quindi sono facilmente implementabili con qualsiasi tipo di DBMS.

Da un punto di vista logico le entità individuate danno origine alle seguenti tabelle:

#### Ordine

Codice (chiave primaria)  
Denominazione

#### Specie

Cod (chiave primaria)  
Ordine di appartenenza (chiave esterna)  
Nome

#### Esemplare animale

Parco (parte della chiave primaria)  
Nome (parte della chiave primaria)  
Specie di appartenenza (chiave esterna)  
Numero individui adulti femmine  
Numero individui adulti maschi  
Numero cuccioli femmine  
Numero cuccioli maschi  
Livello di salute (1-10)  
Rischio di estinzione (1-10)  
Anno del primo avvistamento

#### Tipologia

Codice tipo (chiave primaria)  
Nome tipo

#### Esemplare vegetale

Parco (parte della chiave primaria)  
Nome pianta (parte della chiave primaria)  
Tipo di appartenenza (chiave esterna)  
Stagione di fioritura  
Note varie

Nella scelta degli attributi da associare ad ogni entità abbiamo fatto le seguenti ipotesi aggiuntive che sembrano essere in accordo con la logica seguita dall'estensore della traccia:

- Con il termine esemplare si fa riferimento non al singolo individuo ma a tutti gli individui appartenenti a una sottospecie omogenea (vedi query numero 3)
- Con il termine cucciolo si intende l'animale nato nell'ultimo anno di osservazione (qualora si volesse dettagliare maggiormente la situazione, si potrebbe prevedere un'ulteriore entità INDIVIDUO per la quale verrebbe registrata anche la data di nascita. Nel caso di animali allo stato libero sembra poco probabile che l'osservazione venga svolta in modo così capillare)
- Il rischio di estinzione e lo stato di salute di un esemplare o sottospecie viene valutato direttamente dal responsabile del parco sulla base di osservazioni oggettive e indicato con un valore da 1 a 10. Possiamo supporre che un esemplare sia considerato a alto rischio di estinzione se questo valore è maggiore o uguale a 8.

In base a quanto dello possiamo individuare il seguente modello relazionale espresso in modo formale:

Ordine(Codice, Denominazione)

Specie(Cod, Ordine di appartenenza, Nome)

Esemplare animale(Parco, Nome, Specie di appartenenza, Numero individui adulti femmine, Numero individui adulti maschi, Numero cuccioli femmine, Numero cuccioli maschi, Livello di salute, Rischio di estinzione, Anno del primo avvistamento)

Tipologia (Codice tipo, Nome tipo)

Esemplare vegetale(Parco, Nome pianta, Tipo di appartenenza, Numero individui, Stagione di fioritura, Note varie)

A titolo esemplificativo indichiamo la formulazione della relazione SQL che consente di creare la tabella Esemplare animale.

```
Create Table Esemplareanimale
(Parco char(5) primary key,
 Nome char(20) primary key,
 Specieapp not null references Specie(Cod),
 Adultifemmine integer,
 Adultimaschi integer,
 Cucciolifemmine integer,
 Cucciolimaschi integer,
 Salute smallint,
 Estinzione smallint,
 Anno integer);
```

Passiamo ora a sviluppare le query richieste dal testo.

1. Select Esemplareanimale.Nome, Esemplareanimale.Parco, Specie.Nome from Esemplareanimale, Specie  
Where Esemplareanimale.Specieapp=Specie.Cod  
order by Esemplareanimale.Specieapp;
2. Select Nome, Parco from Esemplareanimale  
where Estinzione>7;
3. select sum(Cucciolifemmine + Cucciolimaschi) as nascite, Parco  
from Esemplareanimale  
where Nome=[inserire il nome dell'esemplare desiderato];
4. supponendo di aver utilizzato nella tabella Tipologia il codice 1 per gli arbusti otteniamo:  
select count(Nomepianta) as numerospecie  
from Esemplarevegetale  
group by Parco  
where Tipoapp=1;

5. 

```
select count(Nomepianta) from Esemplarevegetale
Where Nomepianta like '%pino%'
group by Parco;
```
6. La tecnica più semplice per risolvere il problema è quella di realizzare due query di cui la seconda fa riferimento a una tabella transitoria costruita dalla prima.

```
create view cuccioli as
select Specieapp, parco, sum(Cucciolimaschi+Cucciolifemmine) as Cucc
from Esemplareanimale
group by Specieapp, parco;
```

```
select specieapp, avg(Cucc)
from cuccioli
group by specieapp;
```

7. Anche in questo caso abbiamo optato per una soluzione simile a quella del punto precedente.

```
create view vecchio as
select Specieapp, Parco, min(anno) as eta from Esemplareanimale
group by Specieapp, Parco;
```

```
select Esemplareanimale.nome, Esemplareanimale.Specieapp, vecchio.eta
from Esemplareanimale, vecchio
where Esemplareanimale.specieapp=vecchio.specieapp
and
Esemplareanimale.anno=vecchio.eta
and
Esemplareanimale.parco=[inserire il codice del parco];
```

Passiamo ora all'ultimo quesito della traccia

La traccia fa riferimento al trasferimento di informazioni periodiche da nodi remoti a server di accumulo.

Gran parte del lavoro di gestione dell'accesso al DB è affidata a MYSQL ed alla potenzialità del PHP, o dei vari ambienti di sviluppo per applicazioni web.

Certamente, però, sviluppare il frazionamento delle informazioni, tenendo anche conto delle caratteristiche operative delle reti fisiche, con l'ausilio di moduli ad hoc, garantisce maggior snellezza.

Lo scenario è certamente la programmazione di rete, unitamente all'uso dello stack TCP/IP. Molti sono gli ambienti di lavoro, dato che i linguaggi per operare partendo da questi strumenti, (tcp/ip ) sono molteplici.

Considerando che certamente tutti gli allievi hanno conoscenza del linguaggio C, ci riferiremo, per le nostre considerazioni, a questo linguaggio.

Uno strumento semplice da usare in questo contesto è la programmazione con i socket.

Le molteplici funzioni di libreria disponibili consentono di frazionare lo scambio file in un limitato numero di operazioni(funzioni).

Senza scendere nel dettaglio del loro specifico utilizzo, ci limitiamo a indicare le funzioni principali.

Dall'apertura del canale con la Create, che restituisce l'handle al Socket, si passa alla Bind che collega il canale fisico al socket creato.

Considerando che si voglia sviluppare il software lato client, basterà utilizzare la funzione Connect per avviare il link col Server e le funzioni Send e Receive per lo scambio dei files.

Da tener presente che, con queste modalità, è trasparente la frammentazione in pacchetti prima ed in frames dopo.

Ricordare, infine, di utilizzare la funzione Closesocket per abbattere il Link.

*Antonio Garavaglia*  
*Alessandra Valpiani*  
(ITIS Lagrange – Milano)