Si richiede un progetto di ampio respiro: servirebbe più tempo per l'elaborazione

Argomenti noti ai candidati, ma le richieste prevedono un'analisi approfondita. Una soluzione «semplificata»

Come avviene ormai da diversi anni la traccia ministeriale richiede l'analisi e la progettazione di un database molto articolato che deve essere gestito, per la consultazione e l'aggiornamento, sia in ambiente locale (la sede delle compagnia telefonica), sia via Internet.

Le richieste della traccia sono molto dettagliate e articolate e richiederebbero per una soluzione completa da parte dello studente un tempo molto maggiore di quello a disposizione.

Sicuramente il tema proposto, pur trattando argomenti che dovrebbero essere noti al candidato, sembra più adatto ad un progetto di ampio respiro da sviluppare in classe in un periodo di tempo piuttosto lungo che non per un tema di maturità.

Infatti le richieste spaziano da un'analisi approfondita delle tematiche legate alla sicurezza e alla riservatezza delle informazioni, ad un progetto completo di database in cui le modalità di salvaguardia delle informazioni nel tempo sono lasciate a carico del candidato. Inoltre vengono richieste diverse query che richiedono la conoscenza approfondita delle possibilità offerte dal linguaggio SQL.

Il punto numero 4, infine, allarga l'analisi alle tematiche relative alla gestione di un database remoto e alla realizzazione in Internet di un portale interattivo con accessi riservati, argomento che probabilmente da solo potrebbe rappresentare una parte cospicua della soluzione.

Per questo motivo proporremo una soluzione "semplificata" che pensiamo possa essere compatibile con il tempo a disposizione, segnalando in ogni fase di lavoro le problematiche trascurate e le modalità di approccio alla soluzione senza entrare nei dettagli.

Punto 1

Da un punto di vista generale il problema risulta essere molto articolato in quanto richiederebbe un'analisi approfondita delle tecniche da utilizzare per garantire da una parte la sicurezza e la coerenza dei dati nel tempo, dall'altra la riservatezza delle informazioni che dovrebbero essere accessibili solo alle persone autorizzate.

In particolare vogliamo accennare al fatto che, essendoci nel database informazioni riservate come le password, sarebbe opportuno registrare nel database dei dati crittografati definendo nel contempo l'algoritmo di codifica e decodifica.

Inoltre occorre distinguere tra le operazioni di competenza dell'Amministratore (aggiunta e modifica dei dati relativi ai tecnici, assegnazione di userid e password iniziale, validazione delle modifiche ai dati dei contatti, …) e quelle di competenza dei tecnici (modifica della propria password, consultazione e modifica dei dati relativi ai propri contatti, consultazione e/o modifica dei dati dei contatti degli altri componenti del proprio gruppo, …)

Infine occorre tener presente che nel lasso di tempo intercorrente tra l'accesso da parte di un tecnico per la modifica dei dati e la validazione della modifica stessa e della sua pubblicazione è indispensabile garantire l'integrità e la congruenza dei dati del database (un secondo tecnico non essendo a conoscenza delle modifiche apportate dal collega potrebbe tentare di apportarne altre che potrebbero risultare incompatibili con quelle del collega, dando origine ad un database inconsistente)

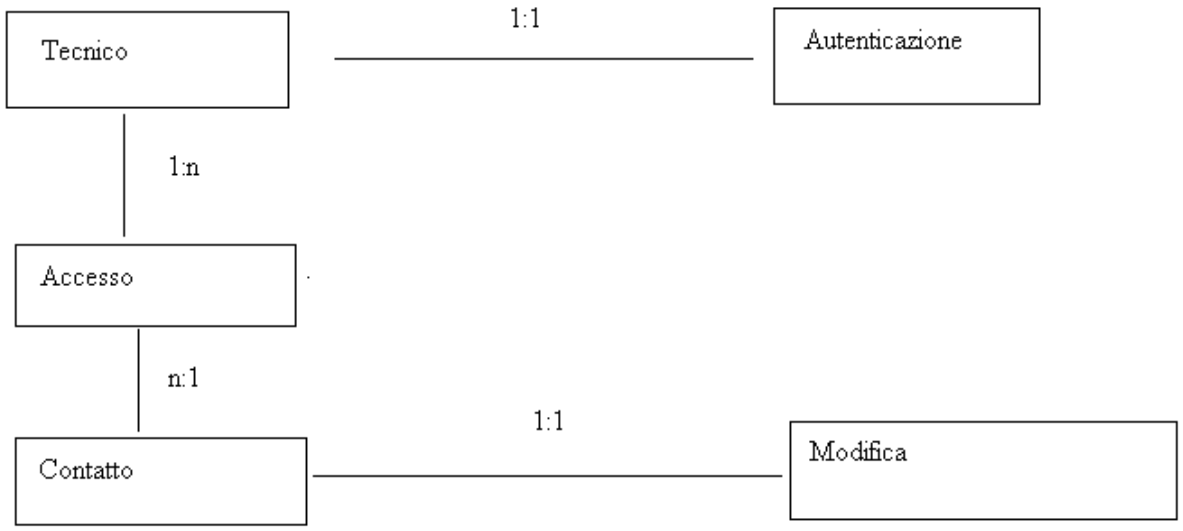
Tutto ciò richiederebbe la gestione di diverse tabelle con accessi separati e dei permessi per i singoli componenti.

Per semplicità nella nostra soluzione ci atterremo alle seguenti ipotesi restrittive:

- Il database è relativo ad un unico gruppo di lavoro, per cui i tecnici hanno libero accesso a tutti i contatti;
- Non vengono prese in esame le problematiche legate alla crittografia dei dati e alla gestione delle password;
- Il sistema una volta accettata una modifica sui dati di un contatto ne impedisce l'accesso per modifica o cancellazione fino alla validazione da parte dell'Amministratore.

Punto 2

Sotto le ipotesi citate presentiamo lo schema concettuale del database:



Di conseguenza lo schema logico del database espresso in terza forma normale è:

Tecnico(ID, Nome, Cognome, …)

Autenticazione (IDTecnico, Userid, Password)

Accesso(IDTecnico, DataInizio, OraInizio, DataFine, OraFine, TipoOperazione, IDContatto)

Contatto(ID, Nome, …, Provincia)

Modifica (IDContatto, … copia campi di contatto …, TipoOperazione)

La tabella Tecnico viene gestita dall'Amministratore per le modifiche, le aggiunte e le cancellazioni e in consultazione dal programma di autenticazione per il permesso di accesso ai dati.

La tabella Accesso viene costruita sulla base della richiesta del tecnico e avvia il programma di consultazione sulla tabella Contatto o di modifica sulla tabella Modifica che contiene le modifiche temporanee in attesa di validazione.

A parte il campo TipoOperazione che limita la possibilità di inserire nuove modifiche su un record, le tabelle Contatto e Modifica sono speculari in quanto dopo la validazione i record per cui è stata concessa l'autorizzazione alla modifica vengono trasferiti uno a uno dalla tabella Modifica alla tabella Contatto, sovrascrivendo i record di quest'ultima tabella.

Al termine dell'esecuzione del programma di validazione le due tabelle saranno identiche e il campo TipoOperazione verrà azzerato in tutti i record.

Punto 3

a)

A titolo esemplificativo mostriamo la definizione della relazione SQL relativa alla tabella Tecnico:

```
CREATE TABLE Accesso
(       IDTecnico           varchar(5),
        DataInizio          date,
        OraInizio           time,
        DataFine            date,
        OraFine             time,
        TipoOperazione      integer,
        IdContatto          varchar(5),
        IdTecnico, DataOraInizio primary key);
```

TipoOperazione 0 = consultazione

1 = modifica

2 = cancellazione

3 = nuovo inserimento

b)

```
SELECT *
        FROM Contatto
        WHERE Provincia = [provincia richiesta]
        ORDER BY Nome;
```

c)

```
SELECT *
        FROM Accesso
        WHERE IdTecnico=[codice tecnico richiesto]
        ORDER BY DataInizio, OraInizio;
```

d)

```
SELECT DISTINCTROW Avg([Accesso Query].[Conteggio Di Accesso]) AS [Media Di Accesso]
FROM [Accesso Query];
```

ove Accesso Query ha la seguente struttura:

```
SELECT DISTINCTROW Accesso.DataInizio, Count(*) AS [Conteggio Di Accesso]
FROM Accesso
GROUP BY Accesso.DataInizio
HAVING (((Accesso.DataInizio) Between [data1] And [data2]));
```

e)

```
SELECT DISTINCTROW IdTecnico, Count(*) AS [Conteggio Di Accesso]
FROM Accesso
GROUP BY IDTecnico
HAVING TipoOperazione=3;
```

f)

```
SELECT DISTINCTROW *
FROM Accesso
ORDER BY IDTecnico
WHERE DataInizio=[data richiesta];
```

g)

SELECT DISTINCTROW *

From Contatto

WHERE ID=(SELECT (IDContatto FROM ConteggioAccesso ORDER BY N DESC LIMIT 1););

ove ConteggioAccesso ha la seguente struttura:

SELECT DISTINCTROW IDContatto, Count(*) AS [N]

FROM Accesso

GROUP BY IDContatto

HAVING (((Accesso.DataInizio) Between [data1] And [data2]));

Punto 4

Questo punto richiederebbe uno sviluppo molto articolato che però a nostro avviso va al di là delle possibilità fornite al candidato dal tempo a disposizione.

Ci limiteremo quindi ad indicare dei criteri di carattere generale che ciascuno potrà poi applicare al proprio caso specifico.

In primo luogo occorrerà progettare un'interfaccia web che consenta di collegarsi al database remoto tramite opportune procedure php attivabili interattivamente dall'utente.

Ciascuna procedura dovrà prevedere un controllo rigoroso dei dati inseriti e il loro invio al sistema remoto per verificare il possesso dei requisiti da parte di chi formula la richiesta.

Se la validazione ha successo verrà chiamata la funzione php mysql_connect con i valori di utente e password previsti per l'accesso di utenti registrati

$db = mysql_connect($host, $user, $password)

or die ("Impossibile connettersi al server $host Autorizzazione negata");

Appare evidente che nel contesto proposto risulta indispensabile imporre all'utente l'utilizzo di adeguati sistemi di protezione tramite le opzioni del browser e l'utilizzo di un proxy server che funga da interfaccia intelligente tra l'utente finale e la banca dati.

Infine va notato che per garantire un discreto margine di sicurezza, sarebbe opportuno far sì che le informazioni riservate, come ad esempio le password, non compaiano nel sorgente HTML della pagina, e che inoltre, tramite opportuni meccanismi, viaggino criptate attraverso la rete.