

PRIMA PARTE

1.1 Analisi e ipotesi aggiuntive

Date le caratteristiche e la natura della società turistica che probabilmente non dispone di personale specializzato per la gestione di un server per ospitare il sito web, la soluzione più indicata è quella di acquistare un servizio di hosting del sito da un fornitore di servizi web (spesso è possibile acquistare il servizio di hosting del sito insieme alla registrazione del dominio che in questo caso potrebbe essere www.visiteguidate.it): questa soluzione è senz'altro la migliore tenuto conto che il sito deve essere operativo 24 ore su 24 e 7 giorni su 7, ed è anche economica. È necessario acquistare, oltre allo spazio web per ospitare il sito, l'integrazione dell'interprete PHP nel server web e la disponibilità di un database My-SQL per la gestione del database: si tratta in ogni caso di un pacchetto molto comune nell'offerta dei fornitori di servizi web.

Un aspetto importante per controllare gli accessi da parte delle varie tipologie di utenti sarà quello di prevedere una profilazione dei medesimi. In questo senso sarà necessario prevedere una tabella con username, password (opportunamente criptate o validate tramite una più sicura tecnica di strong authentication) e funzionalità abilitate per ognuno di essi in modo tale che questi possano accedere solo ad operazioni di loro competenza. La struttura delle funzionalità del sistema sarà quindi organizzata tramite menù che si configurano dinamicamente a seconda del profilo utente attivato (visitatori, gestori di visite/eventi, personale amministrativo, personale direttivo, amministratore di sistema, ecc.)

Dal momento il sistema dovrà necessariamente essere in multiutenza si suppone che venga sviluppato con una tecnologia transazionale aderente alle specifiche *ACID*.

Nell'elaborato esposto di seguito si è ipotizzato che:

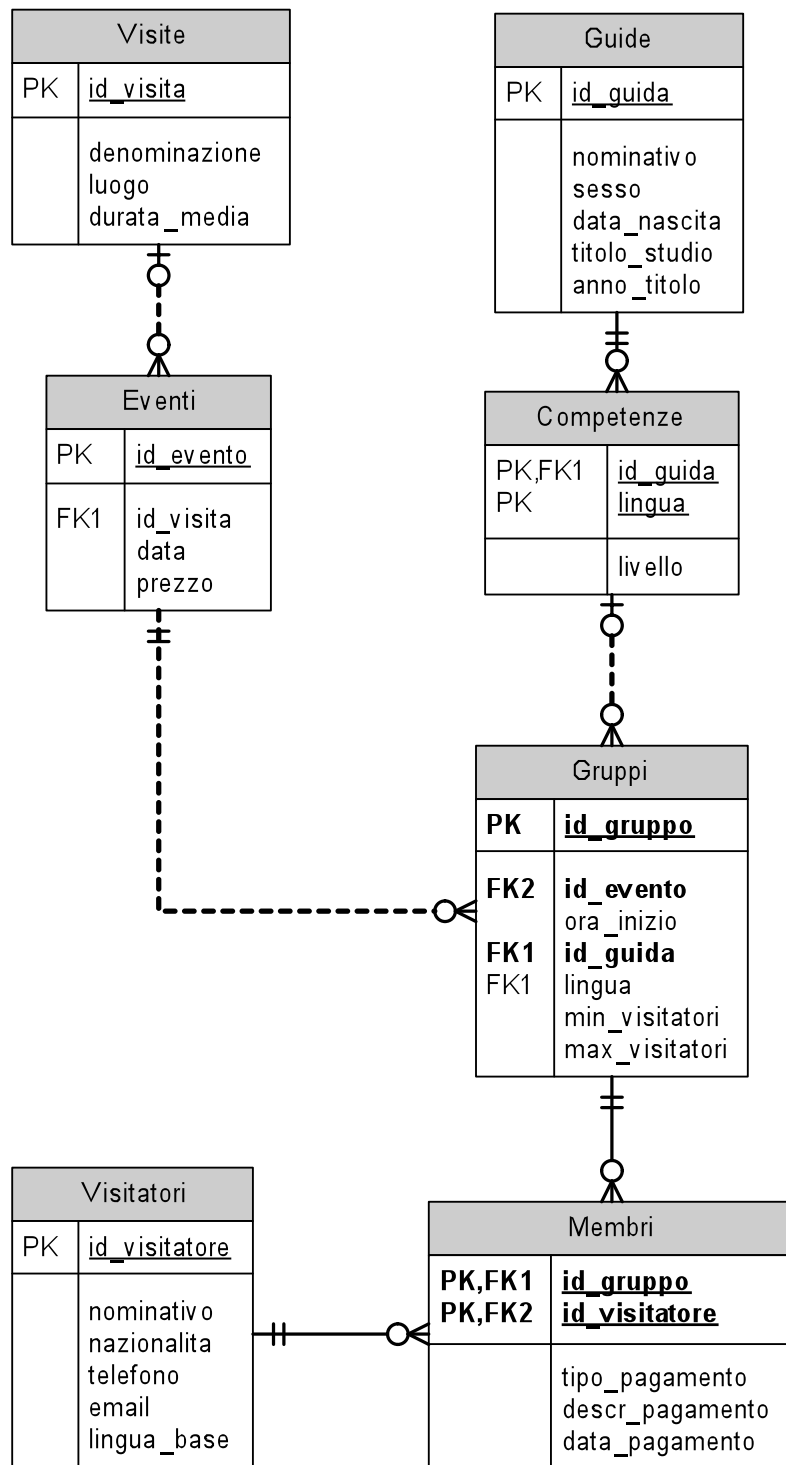
- ⊙ Il dato relativo alla durata media di una visita venga espresso in minuti;
- ⊙ per uno specifico evento di visita guidata possano essere formati anche più gruppi ognuno col proprio orario, accompagnatore e lingua;
- ⊙ i vari visitatori per potersi iscriversi ad uno o più eventi debbono registrarsi sul sito della società fornendo e-mail e recapito telefonico. La banca dati non prevede alcuna gestione relativamente agli utenti anonimi: essi possono operare solo per funzionalità limitate di interrogazione per vedere i dati degli eventi programmati;

- ⊙ per potersi iscrivere ad un gruppo di visita relativamente ad uno specifico evento, nei limiti della disponibilità di posti, ogni visitatore registrato effettui il pagamento tramite carta di credito (con codice della medesima), via PayPal (l'utente deve essere registrato a tale servizio), o tramite bonifico bancario di cui deve fornire gli estremi utilizzando il campo relativo alla descrizione del pagamento;

- ⊙ la società rimborserà ai visitatori l'importo pagato nel caso in cui non venga raggiunto il numero minimo di persone per la formazione di un gruppo (nel tema sviluppato di seguito non è comunque prevista la gestione esplicita di tale evento);

- ⊙ il prezzo di una visita sia comunque individuale e venga espresso a livello di evento in quanto suscettibile di variazioni nel tempo

1.2 Schema database



1.3 Tabelle database

Competenze

È la tabella relativa alle competenze linguistiche delle varie guide:

Campo	R	Tipo	Dim.	Descrizione
id_guida	PK	Numerico		Identificativo guida
lingua	PK	Carattere	15	Lingua conosciuta
livello		Carattere	15	Livello lingua

Chiave primaria: id_guida, lingua

Chiave esterna: id_guida → Guide.id_guida

Eventi

È la tabella relativa alla proramazine delle visite nel tempo:

Campo	R	Tipo	Dim.	Descrizione
id_evento	PK	Numerico		Identificativo evento
Id_visita	FK	Numerico		Identificativo visita
data		Data		Data visita
prezzo		Numerico		Prezzo biglietto individuale

Chiave Primaria: id_visita

Chiave Esterna.: id_visita → Visite.id_visita

Gruppi

È la tabella relativa alla formazione dei gruppi per le varie visite:

Campo	R	Tipo	Dim.	Descrizione
id_gruppo	PK	Numerico		Identificativo gruppo di visita
id_evento	FK	Numerico		Identificativo evento
id_guida	FK	Numerico		Identificativo guida
lingua	FK	Carattere	15	Lingua in cui viene illustrata la visita
ora_inizio		Ora		Orario inizio visita
min_visitatori		Numerico		Numero minimo partecipanti
max_visitatori		Numerico		Numero massimo partecipanti

Chiave Primaria: id_gruppo

Chiave Esterna.: id_guida, lingua → Competenze.id_guida, Competenze.lingua

Chiave Esterna.: id_evento → Eventi.id_evento

Guide

È la tabella relativa all'anagrafica delle guide di cui si avvale la società:

Campo	R	Tipo	Dim.	Descrizione
id_guida	PK	Numerico		Identificativo guida
nominativo		Carattere	30	Cognome e nome guida
sexo		Carattere	1	Sexo (M/F)
data_nascita		Data		Data di nascita
titolo_studio		Carattere	20	Titolo di studio (Laurea, Diploma, ecc.)
anno_titolo		Numerico		Anno conseguimento titolo di studio

Chiave Primaria: id_guida

Membri

È la tabella relativa ai membri che costituiscono i vari gruppi per ognuno dei quali viene riservata una visita:

Campo	R	Tipo	Dim.	Descrizione
id_gruppo	PK/FK	Numerico		Identificativo gruppo di visita
id_visitatore	PK/FK	Numerico		Identificativo visitatore
tipo_pagamento		Carattere	50	Tipologia di pagamento (Bonifico, Carta Credito, ecc.)
descr_pagamento		Carattere	200	Descrizione estremi del pagamento
data_pagamento		Data/ora		Data del pagamento

Chiave Primaria: id_gruppo, id_visitatore

Chiave Esterna.: id_gruppo → Gruppi.id_gruppo

Chiave Esterna.: id_visitatore → Visitatori.id_visitatore

Visitatori

È la tabella relativa all'anagrafica dei visitatori che si sono registrati nel tempo:

Campo	R	Tipo	Dim.	Descrizione
id_visitatore	PK	Numerico		Identificativo visitatore
nominativo		Carattere	30	Cognome e nome del visitatore
nazionalita		Carattere	20	Nazionalità
telefono		Carattere	15	Telefono
email		Carattere	25	Indirizzo e-mail
lingua_base		Carattere	15	Lingua base parlata

Chiave Primaria: id_visitatore

Visite

È la tabella relativa alle visite programmate/effettuate nel tempo:

Campo	R	Tipo	Dim.	Descrizione
id_visita	PK	Numerico		Identificativo visita
denominazione		Carattere	40	Denominazione (titolo) visita
luogo		Carattere	25	Località di riferimento
durata_media		Numerico		Durata media della visita in minuti

Chiave Primaria: id_visita

1.4 DB-schema

Di seguito si è riportato l'intero dbschema del database anche se al candidato viene richiesto di fornire la definizione in linguaggio SQL di almeno tre relazioni tra loro correlate.

```
CREATE TABLE Guide(
    id_guida INTEGER NOT NULL,
    nominativo VARCHAR(30),
    sesso VARCHAR(1),
    data_nascita DATE,
    titolo_studio VARCHAR(20),
    anno_titolo INTEGER,
    CONSTRAINT PrimaryKey PRIMARY KEY(id_guida)
);
CREATE TABLE Visitatori(
    id_visitatore INTEGER NOT NULL,
    nominativo VARCHAR(30),
    nazionalita VARCHAR(20),
    telefono VARCHAR(15),
    email VARCHAR(25),
    lingua_base VARCHAR(15),
    CONSTRAINT PrimaryKey PRIMARY KEY(id_visitatore)
);
CREATE TABLE Visite(
    id_visita INTEGER NOT NULL,
    denominazione VARCHAR(40),
    luogo VARCHAR(25),
    durata_media INTEGER,
    CONSTRAINT PrimaryKey PRIMARY KEY(id_visita)
);
CREATE TABLE Competenze(
    id_guida INTEGER NOT NULL,
    lingua VARCHAR(15) NOT NULL,
    livello VARCHAR(15),
    CONSTRAINT PrimaryKey PRIMARY KEY(id_guida,lingua),
    CONSTRAINT GuideCompetenze FOREIGN KEY(id_guida)
        REFERENCES Guide(id_guida)
);
CREATE TABLE Eventi(
    id_evento INTEGER NOT NULL,
    id_visita INTEGER NOT NULL,
    data DATE,
    prezzo DOUBLE,
    CONSTRAINT PrimaryKey PRIMARY KEY(id_evento),
    CONSTRAINT VisiteEventi FOREIGN KEY(id_visita)
        REFERENCES Visite(id_visita)
);
CREATE TABLE Gruppi(
    id_gruppo INTEGER NOT NULL,
    id_evento INTEGER NOT NULL,
    ora_inizio TIME,
    id_guida INTEGER NOT NULL,
    lingua VARCHAR(15) NOT NULL,
```

```

        min_visitatori INTEGER,
        max_visitatori INTEGER,
    CONSTRAINT PrimaryKey PRIMARY KEY(id_gruppo),
    CONSTRAINT CompetenzeGruppi
        FOREIGN KEY(id_guida,lingua)
        REFERENCES Competenze(id_guida,lingua),
    CONSTRAINT EventiGruppi FOREIGN KEY(id_evento)
        REFERENCES Eventi(id_evento)
    );
CREATE TABLE Membri(
    id_gruppo INTEGER NOT NULL,
    id_visitatore INTEGER NOT NULL,
    tipo_pagamento VARCHAR(50),
    descr_pagamento VARCHAR(200),
    data_pagamento DATETIME,
    CONSTRAINT PrimaryKey
        PRIMARY KEY(id_gruppo,id_visitatore),
    CONSTRAINT GruppiMembri FOREIGN KEY(id_gruppo)
        REFERENCES Gruppi(id_gruppo),
    CONSTRAINT VisitatoriMembri FOREIGN KEY(id_visitatore)
        REFERENCES Visitatori(id_visitatore)
    );

```

1.5 Query SQL

a) *Elenco delle visite programmate nel mese corrente ordinate per luogo e data*

```

SELECT data, luogo, denominazione, prezzo
FROM visite, eventi
WHERE visite.id_visita = eventi.id_visita
    AND MONTH(data) = MONTH(CURDATE())
    AND YEAR(data) = YEAR(CURDATE())
ORDER BY luogo, data;

```

b) *Elenco delle guide laureate che sono madrelingua Inglese e che parlano anche il francese a livello avanzato*

```

SELECT guide.*
FROM guide, competenze
WHERE guide.id_guida = competenze.id_guida
    AND titolo_studio = 'Laurea'
    AND competenze.lingua = 'Inglese'
    AND competenze.livello = 'Madrelingua'
    AND guide.id_guida IN (SELECT DISTINCT id_guida
        FROM competenze
        WHERE competenze.lingua = 'Francese'
            AND competenze.livello = 'Avanzato')
);

```


c) Elenco dei visitatori che nel corso del 2016 hanno partecipato ad almeno tre visite guidate

```
SELECT visitatori.*
FROM visitatori, membri, gruppi, eventi
WHERE visitatori.id_visitatore = membri.id_visitatore
      AND membri.id_gruppo = gruppi.id_gruppo
      AND gruppi.id_evento = eventi.id_evento
      AND YEAR(data) = 2016
GROUP BY id_visitatore
HAVING COUNT(*) > 2;
```

d) Il tipo di visita (denominazione) che nel 2016 ha avuto in totale il maggior numero di visitatori

```
SELECT denominazione
FROM (SELECT denominazione, COUNT(*) AS n_partecipanti
      FROM membri, gruppi, eventi, visite
      WHERE membri.id_gruppo = gruppi.id_gruppo
            AND gruppi.id_evento = eventi.id_evento
            AND eventi.id_visita = visite.id_visita
            AND YEAR(data) = 2016
      GROUP BY denominazione
     ) AS T1
WHERE n_partecipanti = (SELECT MAX(n_partecipanti)
                      FROM (SELECT denominazione,
                                COUNT(*) AS n_partecipanti
                             FROM membri, gruppi, eventi, visite
                             WHERE membri.id_gruppo = gruppi.id_gruppo
                                   AND gruppi.id_evento = eventi.id_evento
                                   AND eventi.id_visita = visite.id_visita
                                   AND YEAR(data) = 2016
                             GROUP BY denominazione
                            ) AS T2
                      );
```

e) Dati dei visitatori che nel mese corrente hanno partecipato a gruppi in lingua diversa dalla propria lingua base.

```
SELECT visitatori.*
FROM visitatori, membri, gruppi, eventi
WHERE visitatori.id_visitatore = membri.id_visitatore
      AND membri.id_gruppo = gruppi.id_gruppo
      AND gruppi.id_evento = eventi.id_evento
      AND YEAR(CURDATE()) = YEAR(data)
      AND MONTH(CURDATE()) = MONTH(data)
      AND lingua <> lingua_base;
```

1.6 Esempio pagine web dinamiche

Come esempio di pagine web dinamiche previste dal sito della società di turismo presentiamo due semplici script PHP (il primo – *ini.php* – per inserire la richiesta ed il secondo – *risposta.php* – per visualizzare la relativa risposta) per la visualizzazione degli eventi previsti nel mese corrente di uno specifico luogo selezionato dall'utente tramite un *combo box*; selezionando l'opzione "Tutti" viene data la possibilità di vedere gli eventi programmati per tutti luoghi disponibili.

Si è ipotizzato che il server My-SQL sia in esecuzione sullo stesso computer del web server e che l'accesso al database "turismo" venga effettuato come utente "root" privo di password; inoltre le pagine web sono completamente prive di struttura grafica.

ini.php

```
<html>
<head>
<title>Cronologia eventi mese corrente per luogo e data</title>
</head>
<body>
<?php
    $connection = mysqli_connect("localhost", "root", "", "turismo");
    if (mysqli_connect_errno($connection))
    {
        echo "Errore di connessione al DBMS My-SQL." ;
        die();
    }
?>
<form action="risposta.php" method="GET">
Selezionare luogo visite (Tutti = qualsiasi luogo):
<select name="luogo">
<?php
    $query = "SELECT DISTINCT luogo
              FROM visite;";
    $result = mysqli_query($connection, $query);
    if (!$result)
    {
        echo "Errore esecuzione query SQL." ;
        die();
    }
    if (mysqli_num_rows($result) == 0)
    {
        echo "Nessun luogo disponibile." ;
        die();
    }
    $tutti="Tutti";
    echo "<option value=\"\$tutti\">$tutti</option>";
    while ($row = mysqli_fetch_array($result))
    {
        echo "<option value=\"\$row[0]\">$row[0]</option>";
    }

```

```

    mysqli_free_result($result);
    ?>
</select>
<br>
<br>
<input type="submit" value="Cerca eventi">
<br>
</form>
</body>
</html>

```

risposta.php

```

<html>
<head>
<title>Eventi programmati</title>
</head>
<body>
<?php
    $connection = mysqli_connect("localhost", "root", "", "turismo");
    if (mysqli_connect_errno($connection))
    {
        echo "Errore di connessione al DBMS My-SQL." ;
        die();
    }
    if ($_GET['luogo']== 'Tutti')
        $query = "SELECT data, luogo, denominazione, prezzo
                FROM visite, eventi
                WHERE visite.id_visita = eventi.id_visita
                AND MONTH(CURDATE()) = MONTH(data)
                AND YEAR(CURDATE()) = YEAR(data)
                ORDER BY data,luogo;";

    else
        $query = "SELECT data, luogo, denominazione, prezzo
                FROM visite, eventi
                WHERE visite.id_visita = eventi.id_visita
                AND MONTH(CURDATE()) = MONTH(data)
                AND YEAR(CURDATE()) = YEAR(data)
                AND luogo = '".$_GET['luogo']."'
                ORDER BY data;";

    $result = mysqli_query($connection, $query);
    if (!$result)
    {
        echo "Errore esecuzione query SQL." ;
        die();
    }
    if (mysqli_num_rows($result) == 0)
    {
        echo "Nessun evento trovato." ;
        die();
    }

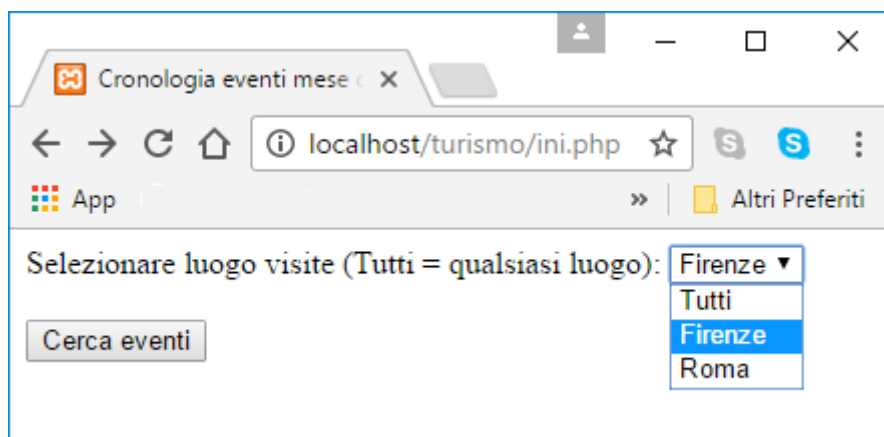
```

```

    }
?>
<table border>
  <caption><b>Eventi programmati</b></caption>
  <thead>
    <tr>
      <th>Data</th>
      <th>Luogo</th>
      <th>Denominazione</th>
      <th>Prezzo biglietto</th>
    </tr>
  </thead>
  <tbody>
    <?php
      while ($row = mysqli_fetch_assoc($result))
      {
?>
        <tr>
          <td><?php echo ($row['data']); ?></td>
          <td><?php echo ($row['luogo']); ?></td>
          <td><?php echo ($row['denominazione']); ?></td>
          <td><?php echo ($row['prezzo']); ?></td>
        </tr>
      <?php
      }
      mysqli_free_result($result);
      echo "</tbody>\n";
      echo "</table>\n";
      mysqli_close($connection);
?>
</body>
</html>

```

Vengono riportate di seguito le pagine web dinamiche create dai due script PHP:



The image shows a screenshot of a web browser window. The browser's address bar displays the URL 'localhost/turismo/risp'. The page title is 'Eventi programmati'. Below the browser interface, there is a table with the following data:

Data	Luogo	Denominazione	Prezzo biglietto
2017-03-23	Firenze	Galleria Uffizi	25
2017-03-27	Firenze	Galleria Uffizi	25
2017-03-30	Firenze	Museo del Bargello	25

SECONDA PARTE

2.1 In relazione al tema proposto nella prima parte il candidato discuta come intende gestire la registrazione degli utenti al sito e la protezione delle funzionalità del sito riservate ad utenti registrati.

Come già detto nella prima parte in merito ai requisiti di base che il sistema dovrà avere rispetto alle modalità di accesso offerte, saranno previsti meccanismi ad hoc per utenti di diversa tipologia sia interni (personale organizzativo, personale amministrativo, responsabili, guide, ecc.) che esterni (visitatori registrati e occasionali). Tali meccanismi avranno il compito di regolamentare e controllare gli accessi.

Per gli interni verrà prevista una profilazione della tipologia di utente e un meccanismo di menù dinamici che, a seconda della tipologia di appartenenza, visualizzi loro solo le funzionalità abilitate.

Per quanto riguarda i visitatori (utenti esterni) è necessario prevedere una procedura di iscrizione che richieda un indirizzo e-mail, una password (strutturata secondo regole di sicurezza in funzione del numero e del tipo di caratteri) e numero di telefono magari affiancata da meccanismi tipo captcha ("*completely automated public Turing test to tell computers and humans apart*"), o altro, per impedire tentativi di accesso di utenti non reali simulati da applicazioni informatiche (BOT) col fine di violare la sicurezza con operazioni di hacking (es. brute force) o spam. In alternativa si potrebbe pensare ad un sistema di autenticazione che sfrutti preventive iscrizioni a social network di larga diffusione (es. Facebook). Una volta effettuata la richiesta di iscrizione il sistema invierà una mail al richiedente con un link verso una opportuna funzionalità del sistema della società di turismo tramite il quale egli potrà confermare e completare l'iscrizione. I dati dei clienti (e-mail e password) verranno memorizzati in maniera criptata (eventualmente in un database esterno dedicato al controllo accessi del sistema). La procedura di registrazione permetterà ai visitatori di potersi iscrivere agli eventi programmati scegliendo le visite di loro interesse nella lingua desiderata e di effettuare contestualmente le operazioni di pagamento dei biglietti che saranno successivamente recapitati loro via mail. Per gli utenti registrati che lo desiderino, sarà inoltre possibile richiedere l'iscrizione ad una mailing list che li informi periodicamente delle attività programmate dalla società di turismo. Per utenti esterni che siano interessati alla sola consultazione degli eventi in programma il sistema offrirà la possibilità di libero accesso senza necessità di registrazione.

2.2 In relazione al tema proposto il candidato discuta come realizzare una nuova funzionalità del sito web della società che permetta alle guide di verificare la situazione delle prenotazioni delle visite di loro competenza.

Il sito della società dovrà prevedere la possibilità per le guide di verificare le situazione in termini di adesioni ai gruppi di visita a loro affidati. Ogni guida sarà dotata di username e password fornita dalla società che dovrà utilizzare per accedere ad una specifica funzionalità riservata prevista dall'interfaccia web del sito. Una volta effettuato l'accesso sarà possibile specificare un periodo compreso tra due date in risposta al quale il sistema mostrerà un rapporto dettagliato degli impegni previsti/effettuati. Il report fornito dal sistema prevederà

in maniera cronologica: data, ora di inizio, luogo, denominazione della visita, numero di partecipanti e lingua usata.

Supponendo di aver già fatto il login e inserito l'intervallo temporale di riferimento, uno script minimale per la visualizzazione del report, a cui vengono passati come parametri le due date e il codice della guida, potrebbe essere il seguente:

```
<html>
<head>
  <title>Eventi per guida</title>
</head>
<body>
  <?php
    $data1=$_GET['data_iniziale'];
    $data2=$_GET['data_finale'];
    $guida=$_GET['guida'];

    $connection = mysqli_connect("localhost", "root", "", "turismo");
    if (mysqli_connect_errno($connection))
    {
      echo "Errore di connessione al DBMS My-SQL." ;
      die();
    }

    $query = "SELECT data, ora_inizio, luogo, denominazione,
                COUNT(*) AS partecipanti,lingua
            FROM visite, eventi, gruppi, membri
            WHERE visite.id_visita = eventi.id_visita
                AND eventi.id_evento = gruppi.id_evento
                AND gruppi.id_gruppo = membri.id_gruppo
                AND data BETWEEN '". $data1.'" AND '". $data2.'"
                AND id_guida=".$guida."
            GROUP BY data,ora_inizio,luogo,denominazione,lingua;";

    $result = mysqli_query($connection, $query);
    if (!$result)
    {
      echo "Errore esecuzione query SQL." ;
      die();
    }
    if (mysqli_num_rows($result) == 0)
    {
      echo "Nessun gruppo trovato." ;
      die();
    }
  ?>
  <table border>
  <caption><b>Gruppi assegnati</b></caption>
  <thead>
  <tr>
    <th>Data</th>
    <th>Ora</th>
    <th>Luogo</th>
    <th>Denominazione</th>
```

```

    <th>Partecipanti</th>
    <th>Lingua</th>
</tr>
</thead>
<tbody>
<?php
    while ($row = mysqli_fetch_assoc($result))
    {
?>
        <tr>
            <td><?php echo ($row['data']); ?></td>
            <td><?php echo ($row['ora_inizio']); ?></td>
            <td><?php echo ($row['luogo']); ?></td>
            <td><?php echo ($row['denominazione']); ?></td>
            <td><?php echo ($row['partecipanti']); ?></td>
            <td><?php echo ($row['lingua']); ?></td>
        </tr>
    <?php
    }
    mysqli_free_result($result);
    echo "</tbody>\n";
    echo "</table>\n";
    mysqli_close($connection);
?>
</body>
</html>

```

2.3 Il candidato enunci le proprietà che rendono una base di dati in terza forma normale.

La normalizzazione di una base di dati relazionale è un procedimento che ha lo scopo di evitare le anomalie che possono verificarsi in una base di dati relazionale a fronte di operazioni di modifica dei dati (inserimento, aggiornamento e cancellazione). Essa è basata sulla decomposizione di una tabella in più tabelle e si articola principalmente su tre «forme normali» (1NF, 2NF e 3NF).

Prima forma normale (1NF). Una tabella è in 1NF se tutti i suoi attributi hanno domini atomici (valori elementari non ulteriormente scomponibili) ed esiste per essa una chiave primaria.

Definizione di “dipendenza funzionale”

Sia $X = \{x_1, x_2, \dots, x_N\}$ un insieme di attributi di una tabella T e Y un attributo di T . Si dice che Y dipende funzionalmente da X (o che X determina Y : $X \rightarrow Y$) se e solo se, per ogni possibile configurazione del contenuto delle righe di T , i valori degli attributi di X determinano univocamente il valore di Y .

(Se due o più righe hanno gli stessi valori per gli attributi x_1, x_2, \dots, x_N , allora devono necessariamente avere anche lo stesso valore dell’attributo Y ; ovvero l’insieme di attributi X rappresenta una chiave per Y)

Seconda forma normale (2NF). Una tabella è in 2NF se è in 1NF e non esistono tra i possibili insiemi di attributi dipendenze funzionali parziali, ma si hanno solo dipendenze funzionali

complete: ogni attributo A che non fa parte della chiave primaria dipende funzionalmente dalla chiave primaria e non da un sottoinsieme di essa.

Terza forma normale (3NF). Una tabella è in 3NF se è in 2NF e non esistono attributi non appartenenti alla chiave primaria dipendenti transitivamente dalla chiave primaria.

2.4 Il candidato discuta il diverso ruolo che hanno i linguaggi di sviluppo web lato client e lato server.

Nell'ambito delle architetture client-server si fa riferimento a linguaggi di scripting *client-side* e *server-side*.

I primi sono linguaggi che permettono l'incorporazione di comandi all'interno del codice HTML per l'implementazione di programmi che vengono eseguiti direttamente dal browser (es. JavaScript).

I secondi sono linguaggi che permettono di sviluppare programmi eseguiti direttamente sul server (es. PHP).

La programmazione lato server prevede tecniche che consentono di generare risorse in tempo reale e che tramite un server possono essere rese fruibili ai client senza che esista alcun file statico corrispondente a esse: in questo modo sono realizzati siti web di tipo dinamico. Questo risultato vien ottenuto generando su richiesta flussi di dati che producono output secondo formati tipici del web (HTML, XHTML, CSS, ecc.).

La programmazione lato server permette di espletare funzioni quali l'autenticazione degli utenti, l'accesso a basi di dati, ecc. In questo modo è possibile sviluppare applicazioni simili a quelle tradizionali anche se con alcune limitazione dovute all'interazione tra il client e il server. I vantaggi di questo tipo di approccio sono principalmente due:

- la possibilità di utilizzare le applicazioni praticamente in ogni luogo dove sia disponibile una connessione alla rete Internet;
- l'utente può utilizzare le applicazioni senza dover installare sul computer client alcun software: è sufficiente un browser standard.

Come accennato, in questo contesto si distingue tra pagine web statiche e pagine web dinamiche:

- le prime sono pagine contenute in file esistenti sul server codificate utilizzando i linguaggi HTML e CSS (più eventuali script in linguaggio JavaScript); sono immutabili nel tempo fino a quando non si decide di modificarle riscrivendo parte del loro codice;
- le seconde sono il prodotto di script eseguiti sul server, il cui output è formattato come codice HTML inviato al browser del client; dal momento che la sequenza delle istruzioni eseguite da un programma dipende dai dati che esso riceve in input e che l'output cambia di conseguenza, ne deriva che le pagine visualizzate lato client possono cambiare in maniera dinamica.

HTML. L'HyperText Markup Language è un linguaggio di markup di pubblico dominio finalizzato alla formattazione e impaginazione di documenti ipertestuali disponibili nel web. Esso è interpretato lato client direttamente dal browser. L'HTML permette l'inserimento di script (es. Javascript) e oggetti esterni (immagini, suoni, filmati, ecc.) ma non è un linguaggio di programmazione vero e proprio perché non prevede alcuna definizione di variabili, strutture dati, funzioni o strutture di controllo tipiche dei programmi. Oggi è utilizzato principalmente per scindere la struttura logica di una pagina web (definita dal markup) dalla sua rappresentazione, gestita tramite gli stili CSS che ne permettono l'adattamento alle nuove esigenze di comunicazione e pubblicazione in Internet.

PhP. Hypertext Preprocessor, è un linguaggio di scripting server side, con una sintassi C-like, utilizzato principalmente per sviluppare applicazioni web. L'esecuzione del codice PHP sul server produce codice HTML da inviare al browser dell'utente che ne fa richiesta per la realizzazione di pagine web dinamiche.

Javascript. È un linguaggio di scripting interpretato lato client direttamente dal browser. Esso è molto usato nella validazione degli input inseriti dall'utente che altrimenti dovrebbe avvenire lato server. Javascript permette quindi una notevole diminuzione del traffico di dati dal client verso il server e, nondimeno, una conseguente velocizzazione delle applicazioni web-based. Nelle ultime versioni di HTML sono stati introdotti nuovi tipi di campi di input per permettere la validazione lato client senza la necessità di impiegare JavaScript; spesso però queste nuove caratteristiche non sono pienamente supportate dai vari browser.