

## ANALISI DEL PROBLEMA

Da un'attenta lettura delle specifiche, si evidenzia che sono richieste le seguenti attività:

- Raccolta e pubblicazione dei dati relativi dell'orario settimanale dei collegamenti con le varie città europee in relazione alle esigenze dei passeggeri
- Raccolta dei dati relativi alle prenotazioni on-line dei passeggeri
- Raccolta dei dati relativi al personale viaggiante (autisti e assistenti di viaggio in servizio)
- Raccolta dei dati relativi alle ditte esterne a cui affidare il servizio di manutenzione autobus

La traccia richiede che il sistema di gestione delle prenotazioni sia online, quindi l'acquisto deve poter essere effettuato attraverso la rete Internet. Questa specifica, richiede che il database sia collocato su un host accessibile dalla rete Internet e che l'utente si possa interfacciare con il database attraverso una applicazione web.

Possiamo rilevare un primo elenco di entità :

**Città** : rappresenta l'insieme delle città raggiungibili dalla sede operativa della società City2City.

**Collegamento** : rappresenta l'insieme dei collegamenti offerti dalla società City2City in partenza dalla sede operativa della società City2City verso le altre città europee o viceversa. In essa è indicato il giorno della settimana e l'orario di partenza e di arrivo.

**Autobus** : rappresenta il parco autobus GT in dotazione della società City2City. Secondo le specifiche è sicuramente indispensabile prevedere alcuni attributi che indichino l'esistenza di confort (aria condizionata, wc, wifi, tv, etc). Inoltre è necessario che ogni istanza contenga una denominazione, una targa e il numero di posti massimo consentito.

**Viaggio** : rappresenta l'insieme dei viaggi effettuati in relazione ai collegamenti previsti. Ogni istanza di questa entità dovrà contenere la data, l'ora di partenza e di arrivo.

**Personale** : rappresenta l'insieme delle persone in servizio presso la società City2City. Ogni istanza di questa entità dovrà contenere i dati anagrafici e il ruolo espletato (autista, assistente).

**Ditta** : rappresenta l'insieme delle ditte che svolgono il servizio di manutenzione degli autobus.

**Confort** : rappresenta l'insieme dei servizi di confort di cui possono essere dotati gli autobus

### Associazioni

Dopo aver identificato le Entità si cercano le associazioni tra le entità indicando la molteplicità. Si cercano le associazioni 1:N ed N:N eseguendo le "regole di lettura" ed assegnando un nome alla associazione che in genere è un verbo.

**Città – Collegamento (direzione)** : una città può essere associata a N collegamenti mentre un collegamento può essere associato ad una città. Si tratta di un'associazione 1 a N.

**Collegamento – Viaggio (Percorso)** : un viaggio deve fare riferimento ad un singolo collegamento offerto mentre ad un collegamento possono corrispondere N viaggi eseguiti. Si tratta di un'associazione 1 a N.

**Autobus – Viaggio (Utilizzazione)** : un viaggio deve fare riferimento ad un singolo autobus mentre un autobus può essere utilizzato per eseguire N viaggi. Si tratta di un'associazione 1 a N.

**Autobus – Personale (Servizio)** : un viaggio può essere associato a N (due autisti più un assistente) persone mentre una persona può essere associata ad N viaggi. Si tratta di una associazione N a N.

**Autobus – Ditta (manutenzione)** : un autobus può essere mantenuto da N ditte mentre una ditta può mantenere N autobus. Si tratta di un'associazione N a N. Ogni istanza di questa associazione dovrà contenere alcuni dati essenziali relativi al servizio di manutenzione (data, elenco dei servizi etc).

**Autobus – Confort (Dotazione)** : un servizio di confort può essere presente su N autobus mentre un autobus può essere dotato di N tipi di Conforts. Si tratta di una associazione N a N. Ogni istanza di questa associazione indicherà la presenza di un determinato confort in un autobus.

**IPOTESI AGGIUNTIVE**

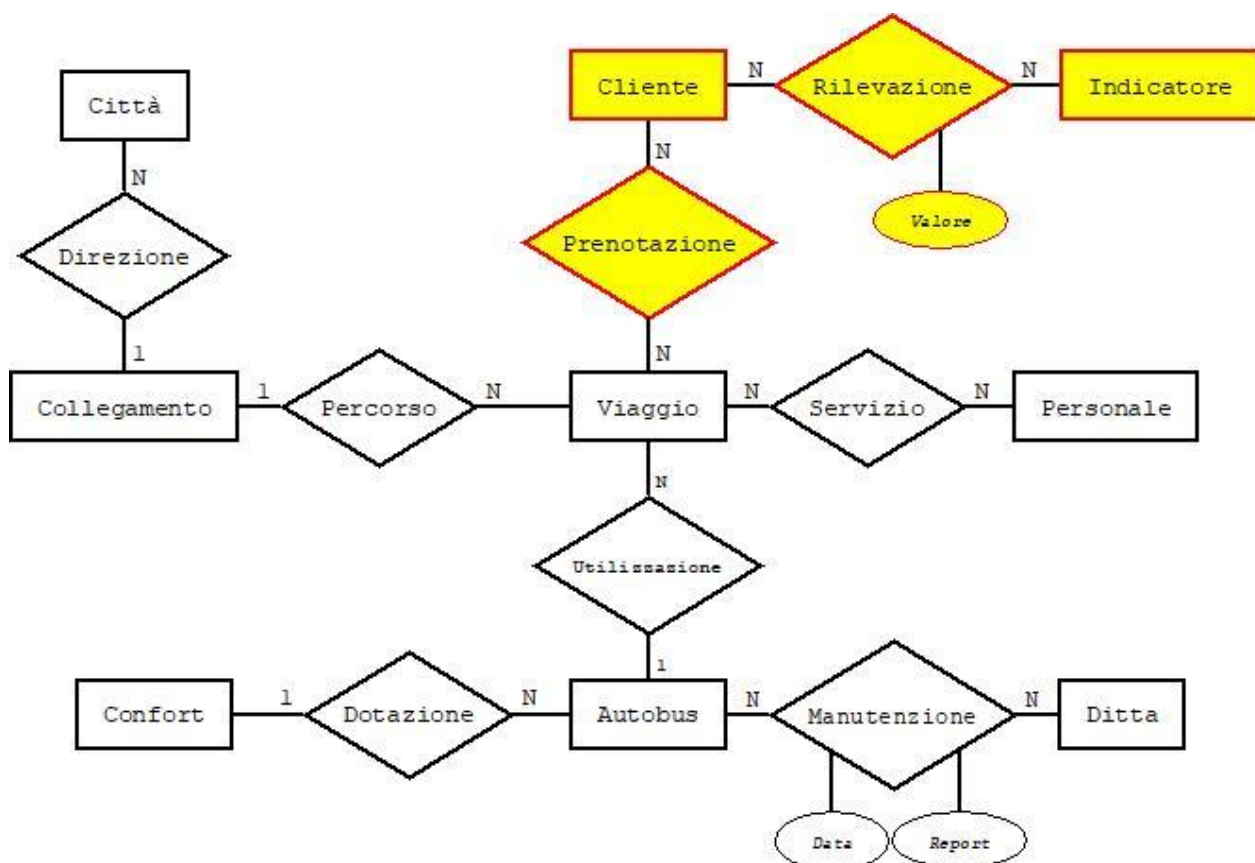
Poichè la compagnia ha interesse a fidelizzare i propri clienti mediante una tessera virtuale a punti, occorrerà associare ad ogni prenotazione il nominativo del cliente.

Pertanto, occorrerà aggiungere l'entità **CLIENTE** ed una associazione "Prenotazione" del tipo N a N con l'entità VIAGGIO.

Inoltre, poiché la compagnia intende attivare operazioni di marketing non convenzionale con i propri clienti, occorrerà aggiungere un'altra entità **INDICATORE** ed un'associazione "Rilevazione" di tipo N a N con l'entità CLIENTE

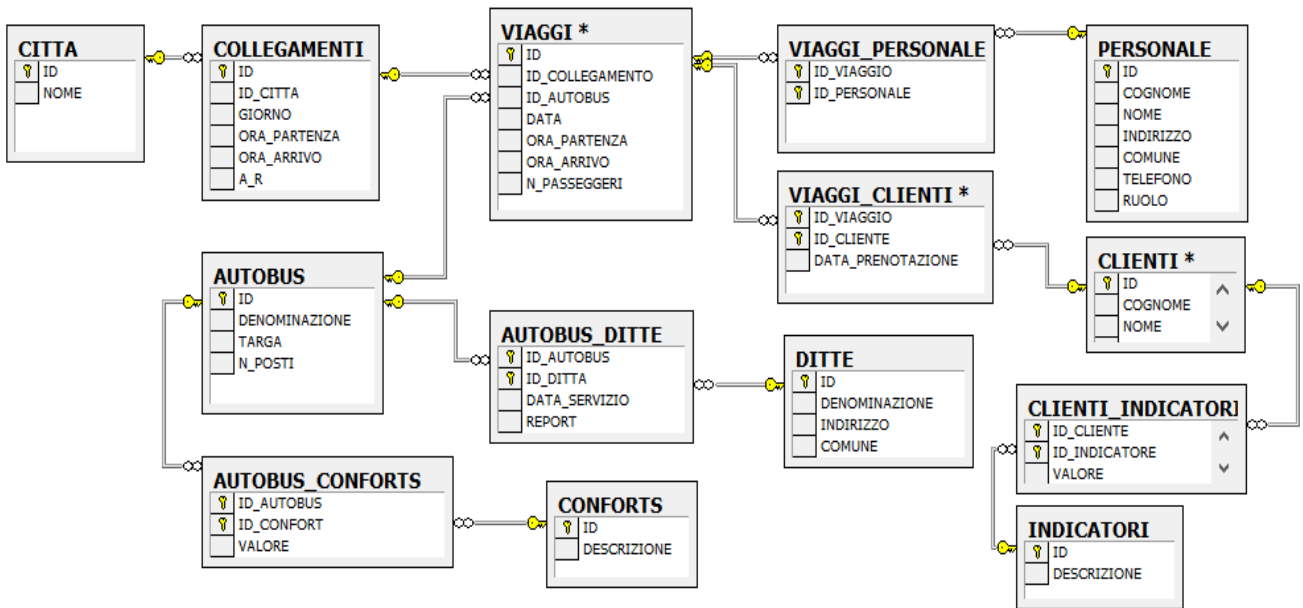
Qui di seguito lo schema concettuale :

**MODELLO E-R City2City**



Da cui si può mappare il seguente modello relazionale :

**MODELLO LOGICO**



**Vincoli di Integrità referenziale**

COLLEGAMENTO.ID_CITTA	contenuto in	CITTA.ID
AUTOBUS_DITTE.ID_DITTA	contenuto in	DITTE.ID
AUTOBUS_DITTE.ID_AUTOBUS	contenuto in	AUTOBUS.ID
VIAGGI.ID_COLLEGAMENTO	contenuto in	COLLEGAMENTI.ID
VIAGGI.ID_AUTOBUS	contenuto in	AUTOBUS.ID
VIAGGI_PERSONALE.ID_VIAGGIO	contenuto in	VIAGGI.ID
VIAGGI_PERSONALE.ID_PERSONALE	contenuto in	PERSONALE.ID
AUTOBUS_CONFORTS.ID_AUTOBUS	contenuto in	AUTOBUS.ID
AUTOBUS_CONFORTS.ID_CONFORT	contenuto in	CONFORTS.ID
VIAGGI_CLIENTI.ID_VIAGGIO	contenuto in	VIAGGI.ID
VIAGGI_CLIENTI.ID_CLIENTE	contenuto in	CLIENTI.ID
CLIENTI_INDICATORI.ID_CLIENTE	contenuto in	CLIENTI.ID
CLIENTI_INDICATORI.ID_INDICATORE	contenuto in	INDICATORI.ID

**SOLUZIONI TECNOLOGICHE**

Per quanto riguarda le soluzioni tecnologiche, occorre dimensionare le risorse hardware e software per l'implementazione della soluzione.

Si userà il DBMS SQLSERVER per la creazione del database.

Si ipotizza quindi di poter testare il funzionamento del software in ambiente locale che comprende una piattaforma integrata contenente:

- Sistema operativo Windows NT con
  - il server web con motore ASP.NET
  - il database SQLSERVER
  - l'interfaccia grafica (ad es. **SQL Management Studio**) per la creazione delle tabelle e il test delle query in SQL.

Qui di seguito in la creazione del DATABASE

### MODELLO FISICO

```
CREATE DATABASE CITY2CITY
```

```
CREATE TABLE PERSONALE(  
    ID int NOT NULL PRIMARY KEY,  
    COGNOME nvarchar(50) NULL,  
    NOME nvarchar(50) NULL,  
    INDIRIZZO nvarchar(50) NULL,  
    COMUNE nvarchar(50) NULL,  
    TELEFONO nvarchar(50) NULL,  
    RUOLO nvarchar(50) NULL  
)
```

```
CREATE TABLE AUTOBUS(  
    ID int NOT NULL PRIMARY KEY,  
    DENOMINAZIONE nvarchar(30) NULL,  
    TARGA nvarchar(10) NOT NULL,  
    N_POSTI int NOT NULL  
)
```

```
CREATE TABLE CONFORTS(  
    ID int NOT NULL PRIMARY KEY,  
    DESCRIZIONE nvarchar(50) NOT NULL  
)
```

```
CREATE TABLE DITTE(  
    ID int NOT NULL PRIMARY KEY,  
    DENOMINAZIONE nvarchar(50) NOT NULL,  
    INDIRIZZO nvarchar(50) NULL,  
    COMUNE nvarchar(50) NULL  
)
```

```
CREATE TABLE CITTA(  
    ID int NOT NULL PRIMARY KEY,  
    NOME nvarchar(50) NOT NULL  
)
```

```
CREATE TABLE COLLEGAMENTI(  
    ID int NOT NULL PRIMARY KEY,  
    ID_CITTA int NOT NULL FOREIGN KEY REFERENCES CITTA (ID),  
    GIORNO int NOT NULL,  
    ORA_PARTENZA TIME NOT NULL,  
    ORA_ARRIVO TIME NOT NULL,  
    A_R char(1) NOT NULL  
)
```

```
CREATE TABLE VIAGGI(  
    ID int NOT NULL PRIMARY KEY,  
    ID_COLLEGAMENTO int NOT NULL FOREIGN KEY REFERENCES COLLEGAMENTI (ID),  
    ID_AUTOBUS int NOT NULL FOREIGN KEY REFERENCES AUTOBUS (ID),  
    DATA smalldatetime NOT NULL,  
    ORA_PARTENZA TIME NOT NULL,  
    ORA_ARRIVO TIME NOT NULL,  
    N_PASSEGGERI int NOT NULL,  
)
```

```
CREATE TABLE VIAGGI_PERSONALE (  
    ID_VIAGGIO int NOT NULL FOREIGN KEY REFERENCES VIAGGI (ID),  
    ID_PERSONALE int NOT NULL FOREIGN KEY REFERENCES PERSONALE (ID)  
    PRIMARY KEY (ID_VIAGGIO, ID_PERSONALE)  
)
```

```
CREATE TABLE AUTOBUS_DITTE (  
    ID_AUTOBUS int NOT NULL FOREIGN KEY REFERENCES AUTOBUS (ID),  
    ID_DITTA int NOT NULL FOREIGN KEY REFERENCES DITTE (ID),  
    DATA_SERVIZIO smalldatetime NOT NULL,  
    REPORT nvarchar(500) NOT NULL  
    PRIMARY KEY (ID_AUTOBUS, ID_DITTA)  
)
```

```
CREATE TABLE AUTOBUS_CONFORTS (  
    ID_AUTOBUS int NOT NULL FOREIGN KEY REFERENCES AUTOBUS (ID),  
    ID_CONFORT int NOT NULL FOREIGN KEY REFERENCES CONFORTS (ID),  
    VALORE int NOT NULL  
    PRIMARY KEY (ID_AUTOBUS, ID_CONFORT)  
)
```

```
CREATE TABLE CLIENTI (  
    ID int NOT NULL PRIMARY KEY,  
    COGNOME nvarchar(20) NOT NULL,  
    NOME nvarchar(30) NOT NULL,  
    NAZIONE nvarchar(2) NOT NULL,  
    EMAIL nvarchar(50) NOT NULL  
)
```

```
CREATE TABLE VIAGGI_CLIENTI (  
    ID_VIAGGIO int NOT NULL FOREIGN KEY REFERENCES VIAGGI (ID),  
    ID_CLIENTE int NOT NULL FOREIGN KEY REFERENCES CLIENTI (ID)  
    PRIMARY KEY (ID_VIAGGIO, ID_CLIENTE)  
)
```

```
CREATE TABLE INDICATORI (  
    ID int NOT NULL PRIMARY KEY,  
    DESCRIZIONE nvarchar(50) NOT NULL  
)
```

```
CREATE TABLE CLIENTI_INDICATORI (  
    ID_CLIENTE int NOT NULL FOREIGN KEY REFERENCES CLIENTI (ID),  
    ID_INDICATORE int NOT NULL FOREIGN KEY REFERENCES INDICATORI (ID),  
    VALORE int NOT NULL  
    PRIMARY KEY (ID_CLIENTE, ID_INDICATORE)  
)
```

## QUERY

- a) elenco dei viaggi di andata svolti verso una determinata città, in un intervallo di date fornite in input;

```
SELECT VIAGGI.*
FROM VIAGGI INNER JOIN COLLEGAMENTI ON VIAGGI.ID_COLLEGAMENTO = COLLEGAMENTI.ID
INNER JOIN CITTA ON COLLEGAMENTI.ID_CITTA = CITTA.ID
WHERE CITTA.NOME = "MILANO"
AND VIAGGI.DATA BETWEEN '01/01/2016' AND '20/06/2016'
```

- b) per ciascuna città collegata, calcolare la media del tempo di percorrenza dei viaggi di andata effettuati nel mese di agosto dell'anno corrente.

```
SELECT CITTA.NOME, AVG (ORA_ARRIVO - ORA_PARTENZA) AS TEMPO_MEDIO
FROM VIAGGI INNER JOIN COLLEGAMENTI ON VIAGGI.ID_COLLEGAMENTO = COLLEGAMENTI.ID
INNER JOIN CITTA ON COLLEGAMENTI.ID_CITTA = CITTA.ID
WHERE MONTH(VIAGGI.DATA) = 08 AND YEAR(VIAGGI.DATA) = 2016
GROUP BY CITTA.NOME
```

## PAGINA WEB IN AMBIENTE ASP.NET

### DISEGNO

# ANAGRAFICA CLIENTI

Cognome	<input type="text"/>
Nome	<input type="text"/>
Nazione	<input type="text"/>
Email	<input type="text"/>
<input type="button" value="SALVA"/>	<input type="button" value="ANNULLA"/>

### CODICE HTML CON ELEMENTI ASP.NET INTEGRATI

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="CLIENTI.aspx.vb" Inherits="CLIENTI"%>
```

```
<html>
<head runat="server">
  <title>ANAGRAFICA CLIENTI</title>
</head>
<body>
  <form id="FORM_CLIENTI" runat="server">
    <h1>ANAGRAFICA CLIENTI</h1>
    <table>
      <tr>
        <td>
          <asp:Label ID="Label1" runat="server" Text="Cognome"></asp:Label>
        </td>
        <td>
          <asp:TextBox ID="TB_COGNOME" runat="server"></asp:TextBox>
        </td>
      </tr>
      <tr>
        <td>
          <asp:Label ID="Label2" runat="server" Text="Nome"></asp:Label>
        </td>
        <td>
          <asp:TextBox ID="TB_NOME" runat="server"></asp:TextBox>
        </td>
      </tr>
      <tr>
        <td>
          <asp:Label ID="Label3" runat="server" Text="Nazione"></asp:Label>
        </td>
        <td>
          <asp:TextBox ID="TB_NAZIONE" runat="server"></asp:TextBox>
        </td>
      </tr>
      <tr>
        <td>
          <asp:Label ID="Label4" runat="server" Text="Email"></asp:Label>
        </td>
      </tr>
    </table>
  </form>
</body>
</html>
```

```

        <td>
            <asp:TextBox ID="TB_EMAIL" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>
            <asp:Button runat="server" ID="BT_SALVA" Text="SALVA" />
        </td>
        <td>
            <asp:Button runat="server" ID="BT_ANNULLA" Text="ANNULLA"/>
        </td>
    </tr>
    <tr>
        <td>
            <asp:Label ID="LB_MESSAGGIO" runat="server" Text=" "></asp:Label>
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

## CODICE VB.NET

```

Imports System.Data.SqlClient
Partial Class CLIENTI
    Inherits System.Web.UI.Page
    Const STRINGA As String = "DATA SOURCE = city2city.SDF;File Mode=Shared Read;Persist
Security Info=False"

    Dim CONNESSIONE As New SqlConnection
    Dim COMANDO As New SqlCommand

    Sub SALVA(sender As Object, e As EventArgs) Handles BT_SALVA.Click

        Dim CONNESSIONE As New SqlConnection
        Dim COMANDO As New SqlCommand
        CONNESSIONE.ConnectionString = STRINGA
        COMANDO.CommandText = "INSERT INTO clienti (COGNOME,NOME, NAZIONE,EMAIL) VALUES ('" &
            TB_COGNOME.Text & "','" &
            TB_NOME.Text & "','" &
            TB_NAZIONE.Text & "','" &
            TB_EMAIL.Text & "')"

        Try
            CONNESSIONE.Open()
            COMANDO.Connection = CONNESSIONE
            COMANDO.ExecuteNonQuery()
            LB_MESSAGGIO.Text = "insert eseguita !"
        Catch ex As Exception
            LB_MESSAGGIO.Text = Err.Description
        Finally
            CONNESSIONE.Close()
        End Try
    End Sub

End Class

```