

SIMULAZIONE DELLA SECONDA PROVA SCRITTA DELL'ESAME DI STATO

Istituti Tecnici Settore Tecnologico Articolazione Informatica

SIMULAZIONE DEL TEMA: INFORMATICA E SISTEMI E RETI

*Simulazione e svolgimento a cura di Cesare Iacobelli, Marialaura Ajme, Elena Baldino,
Velia Marrone, Renato Rondano*

Il candidato (che potrà eventualmente avvalersi delle conoscenze e competenze maturate attraverso esperienze di alternanza scuola-lavoro, stage o formazione in azienda) svolga la prima parte della prova e due tra i quesiti proposti nella seconda parte.

PARTE PRIMA

Considerare la gestione di una società che si occupa della spedizione di missive (pacchi, raccomandate e plichi in genere) su tutto il territorio nazionale, per varie tipologie di clienti (privati cittadini, ditte, ...). La società possiede una serie di sedi in molte città italiane. Le sedi sono dotate di un magazzino, dove vengono ritirate le missive da spedire. Nella stessa sede è possibile anche ritirare le missive giunte dalle altre sedi. Ogni sede invia le missive da spedire ad un centro regionale di riferimento e, dallo stesso centro, riceve le missive da consegnare ai destinatari finali nella propria zona. Ad esempio, un cliente di Nichelino (Piemonte) chiede di consegnare una missiva ad un destinatario di Lanciano (Abruzzo): la missiva verrà presa in carico dalla sede di Nichelino che la inoltrerà alla sede regionale di Torino, che a sua volta la inoltrerà alla sede regionale di Pescara, che infine la inoltrerà alla sede di Lanciano per la consegna al domicilio del destinatario finale.

La società considerata vuole automatizzare il processo di raccolta, smistamento e consegna. In particolare, ai fini del tracciamento, vuole introdurre un sistema di identificazione delle missive che consenta all'azienda, ai clienti e ai destinatari di seguirne la posizione nel loro percorso fino a destinazione. Quando un cliente (mittente) vuole fare una spedizione, effettua una richiesta online: dalla sede più vicina, un trasportatore, recandosi presso il cliente per il ritiro, trasmette al sistema la presa in carico della spedizione. Un apposito algoritmo, già esistente, provvede a definire il percorso per la consegna. In ciascun passaggio del trasporto, la missiva viene prima presa in carico in ingresso e poi tracciata in uscita dai magazzinieri dei centri (locali e regionali) che la gestiscono. Il trasportatore che porta a termine una consegna raccoglie la firma del destinatario e la trasmette al sistema di tracciamento.

Il candidato, fatte le opportune ipotesi aggiuntive:

1. ipotizzi come potrà essere organizzata operativamente la nuova procedura di gestione informatizzata (acquisizione dei dati di mittente e destinatario, presa in carico dal mittente, metodi di identificazione e procedure operative di tracciamento in ciascuna sede fino alla consegna, tipologia delle informazioni raccolte, rilevamento dell'avvenuta consegna);
2. illustri il progetto dell'infrastruttura informatica necessaria per realizzarne la gestione automatizzata e consentirne la tracciabilità, motivando la soluzione scelta (interna o in Cloud) e dettagliando i dispositivi utilizzati e le modalità di comunicazione tra sistemi;
3. progetti a livello concettuale e a livello logico il database necessario per la rappresentazione e la gestione dei dati del sistema;
4. progetti inoltre un applicativo per la gestione via Web delle attività, tenendo conto delle seguenti richieste:
 - inserimento dei dati di una nuova missiva da inviare e modifica della sua posizione durante la spedizione;
 - possibilità da parte di un cliente, inserendo il codice della spedizione, di conoscere il percorso fatto dalla missiva fino a quel momento.

Sviluppi quindi in un linguaggio a piacere una parte significativa del codice.

PARTE SECONDA

QUESITO 1

Data la seguente tabella PRODOTTO (CodProdotto, DescrizioneProdotto, CodMagazzino, QuantitàProdotto, IndirizzoMagazzino, Magazziniere) dire se è in terza forma normale, e, in caso contrario, spiegarne i motivi. Trasformarla poi in un insieme di tabelle in terza forma normale.

QUESITO 2

Il servizio di Virtual Private Network (VPN) offre la possibilità di connettersi da remoto, in modo sicuro, alle risorse dell'azienda. Descrivere quali meccanismi di sicurezza sono tipicamente implementati in una VPN e fornire alcuni esempi di protocolli utilizzati per realizzare le VPN.

QUESITO 3

Date le due seguenti tabelle:

EDITORE (idEditore,nome,sede,città)

LIBRO (idLibro,titolo,autore,prezzo,genere,codEditore)

scrivere in SQL le query che permettano di:

1. visualizzare il numero di libri del genere “giallo” edito dalla casa editrice fornita in input;
2. visualizzare il titolo del libro scritto dall'autore fornito in input che ha il prezzo maggiore.

QUESITO 4

In relazione al tema proposto nella prima parte, si approfondiscano gli aspetti legati alla sicurezza delle strumentazioni, dei dati gestiti e del servizio offerto e si analizzino le misure utili a garantire la continuità del servizio.

SOLUZIONI DEI PUNTI DELLA PARTE PRIMA

Ipotesi aggiuntive

Ipotizziamo che la SOCIETÀ DI SPEDIZIONI (**SDP**) abbia fornito ad ogni trasportatore e ad ogni magazziniere un **mini tablet** con preinstallata un'apposita APP per comunicare al sistema la presa in carico della missiva.

La missiva viene univocamente identificata attraverso un **QR code** che il sistema produce e che il primo trasportatore applica alla missiva quando la ritira dal mittente.

Il mini tablet è dotato di fotocamera per la lettura del QR code e consente di acquisire la firma grafometrica del mittente e del destinatario.

L'avvenuta **presa in carico** viene comunicata (dal trasportatore o dal magazziniere) al sistema leggendo il QR code applicato alla missiva e inviando tutti i dati (codice_dipendente, data, ora, QR code) attraverso l'apposita APP della **SDP** installata sul mini tablet.

I dipendenti della **SDP** che vengono fisicamente a contatto con la missiva, siano essi magazzinieri o trasportatori, saranno i principali autori del tracciamento della spedizione.

PARTE PRIMA – SOLUZIONE DEL PUNTO 1

La nuova procedura di gestione informatizzata delle missive vede coinvolte diverse componenti che interagiscono tra loro. Le elenchiamo.

- a) **SOCIETÀ DI SPEDIZIONI**: è un'azienda che, attraverso una procedura informatizzata, vuole fornire al mittente e al destinatario un servizio di consegna e tracciabilità di una missiva in spedizione. Per fare questo utilizza due programmi: un'APP da installare sui mini tablet dei trasportatori e dei magazzinieri e un'applicazione centralizzata per la gestione di tutte le spedizioni. Nei propri uffici risiederà la centrale operativa.
- b) **MITTENTE**: è un privato o un'azienda che deve effettuare una o più spedizioni. Per farlo accede al servizio di spedizione online di SOCIETÀ DI SPEDIZIONI ed effettua la richiesta compilando un apposito form (se non è ancora cliente dovrà effettuare prima la registrazione) con i propri dati, quelli del destinatario e le caratteristiche della missiva.
- c) **DESTINATARIO**: è un privato o un'azienda che deve ricevere una o più spedizioni.
- d) **SEDE LOCALE (SL)**: è una qualsiasi sede locale della SOCIETÀ DI SPEDIZIONI dotata di magazzino e almeno un **MAGAZZINIERE**.
- e) **SEDA REGIONALE (SR)**: sede regionale dotata di un magazzino e di almeno un **MAGAZZINIERE**.
- f) **TRASPORTATORE**: è assegnato a una SL o a un SR ed è dotato di mini tablet per la presa in carico della spedizione.
- g) **MAGAZZINIERE**: è assegnato a una SL o a un SR ed è dotato di mini tablet per la presa in carico della spedizione.

Lo schema funzionale è rappresentato nella Figura 1.

Dallo schema si nota come siano due le componenti nevralgiche del sistema: la SOCIETÀ DI SPEDIZIONI e il TRASPORTATORE. La prima perché deve essere informata di ogni passaggio della missiva da una componente all'altra. Il secondo perché esegue fisicamente ogni passaggio della missiva tra le componenti.

In ogni fase del processo di spedizione una sola delle componenti del sistema ha in carico la **spedizione**, cioè è responsabile della missiva, e lo è fino a quando la missiva non viene presa in carico da un'altra componente.

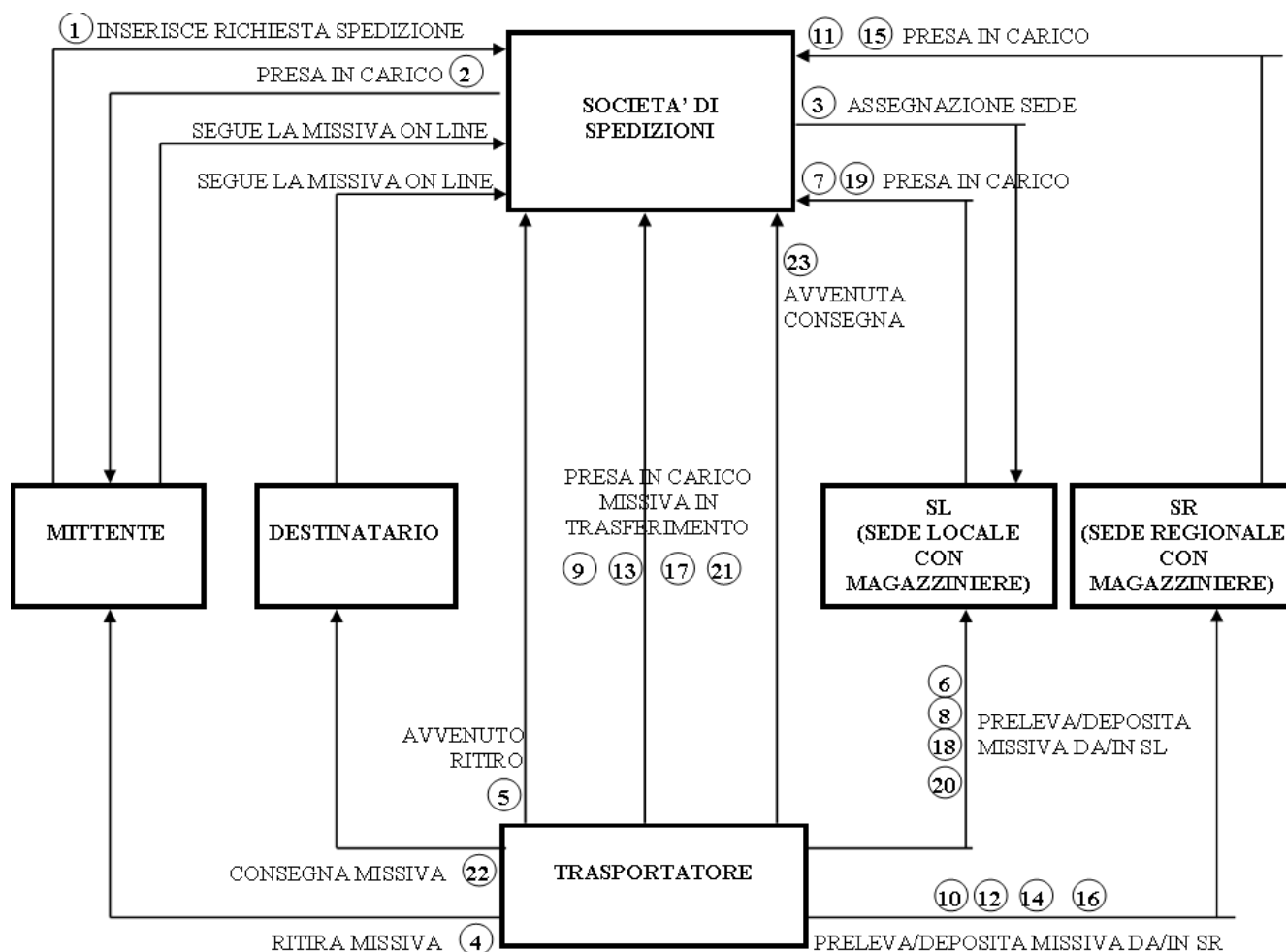


Figura 1 Schema funzionale del sistema di spedizioni automatizzato

Vediamo la sequenza dei 23 passi della nuova procedura automatizzata (figura 1).

1. Il **MITTENTE** accede al portale della **SOCIETÀ DI SPEDIZIONI**, effettua il login al servizio spedizioni e compila il form con la richiesta.
2. La **SOCIETÀ DI SPEDIZIONI** prende in carico la richiesta, la elabora e fornisce via mail a **MITTENTE** e **DESTINATARIO** il report sui tempi di consegna previsti.
3. La **SOCIETÀ DI SPEDIZIONI** assegna alla **SL** di competenza (l'algoritmo selezionerà la sede locale più vicina al **MITTENTE**) la spedizione e le fornisce il QR code per la missiva.
4. Il **TRASPORTATORE**, incaricato dalla **SL** di competenza, si reca dal **MITTENTE** munito di QR code, ritira la missiva, ne acquisisce la firma grafometrica, applica l'etichetta col QR code sulla missiva.
5. Il **TRASPORTATORE** comunica al sistema la presa in carico (avvenuto ritiro) inviando il proprio codice dipendente, la data, l'ora, il QR code e la firma acquisita.
6. Il **TRASPORTATORE** porta il pacco alla **SL** di competenza.

7. La **SL** di competenza, tramite un suo **MAGAZZINIERE**, prende in carico la missiva e la comunica al sistema.
8. Il **TRASPORTATORE** incaricato dalla SL di competenza preleva la missiva dal magazzino.
9. Il **TRASPORTATORE** comunica al sistema l'avvenuta presa in carico.
10. Il **TRASPORTATORE** porta la missiva alla SR della regione del **MITTENTE**.
11. La **SR** di competenza, tramite un suo **MAGAZZINIERE**, prende in carico la missiva e la comunica al sistema.
12. Il **TRASPORTATORE** incaricato dalla SR di competenza preleva la missiva dal magazzino.
13. Il **TRASPORTATORE** comunica al sistema l'avvenuta presa in carico.
14. Il **TRASPORTATORE** porta la missiva alla SR della regione del **DESTINATARIO**.
15. La **SR** di competenza, tramite un suo **MAGAZZINIERE**, prende in carico la missiva e la comunica al sistema.
16. Il **TRASPORTATORE** incaricato dalla SR di competenza preleva la missiva dal magazzino.
17. Il **TRASPORTATORE** comunica al sistema l'avvenuta presa in carico.
18. Il **TRASPORTATORE** porta la missiva alla SL di competenza (la più vicina al **DESTINATARIO**).
19. La **SL** di competenza, tramite un suo **MAGAZZINIERE**, prende in carico la missiva e la comunica al sistema.
20. Il **TRASPORTATORE** incaricato dalla SL di competenza preleva la missiva dal magazzino.
21. Il **TRASPORTATORE** comunica al sistema l'avvenuta presa in carico.
22. Il **TRASPORTATORE** provvede alla consegna della missiva presso il **DESTINATARIO**, acquisendone la firma grafometrica.
23. Il **TRASPORTATORE** comunica al sistema l'avvenuta consegna inviando, oltre ai soliti dati, anche la firma acquisita.

In qualsiasi momento, mittente e destinatario possono seguire lo stato della spedizione accedendo al servizio online della **SOCIETÀ DI SPEDIZIONI**.

PARTE PRIMA – SOLUZIONE DEL PUNTO 2

Nella Figura 2 viene illustrato il progetto dell'infrastruttura informatica necessaria a rendere operativo il sistema della **SOCIETÀ DI SPEDIZIONI** in una regione dotata di tre sedi locali distaccate (**SL**) e una sede regionale (**SR**). Tale scenario può essere replicato su scala nazionale.

Due sono le tecnologie necessarie a realizzare il sistema: la rete **4G/LTE-A** per la presenza di lavoratori mobili (i trasportatori) e il **Cloud** (in alternativa alla soluzione interna all'azienda).

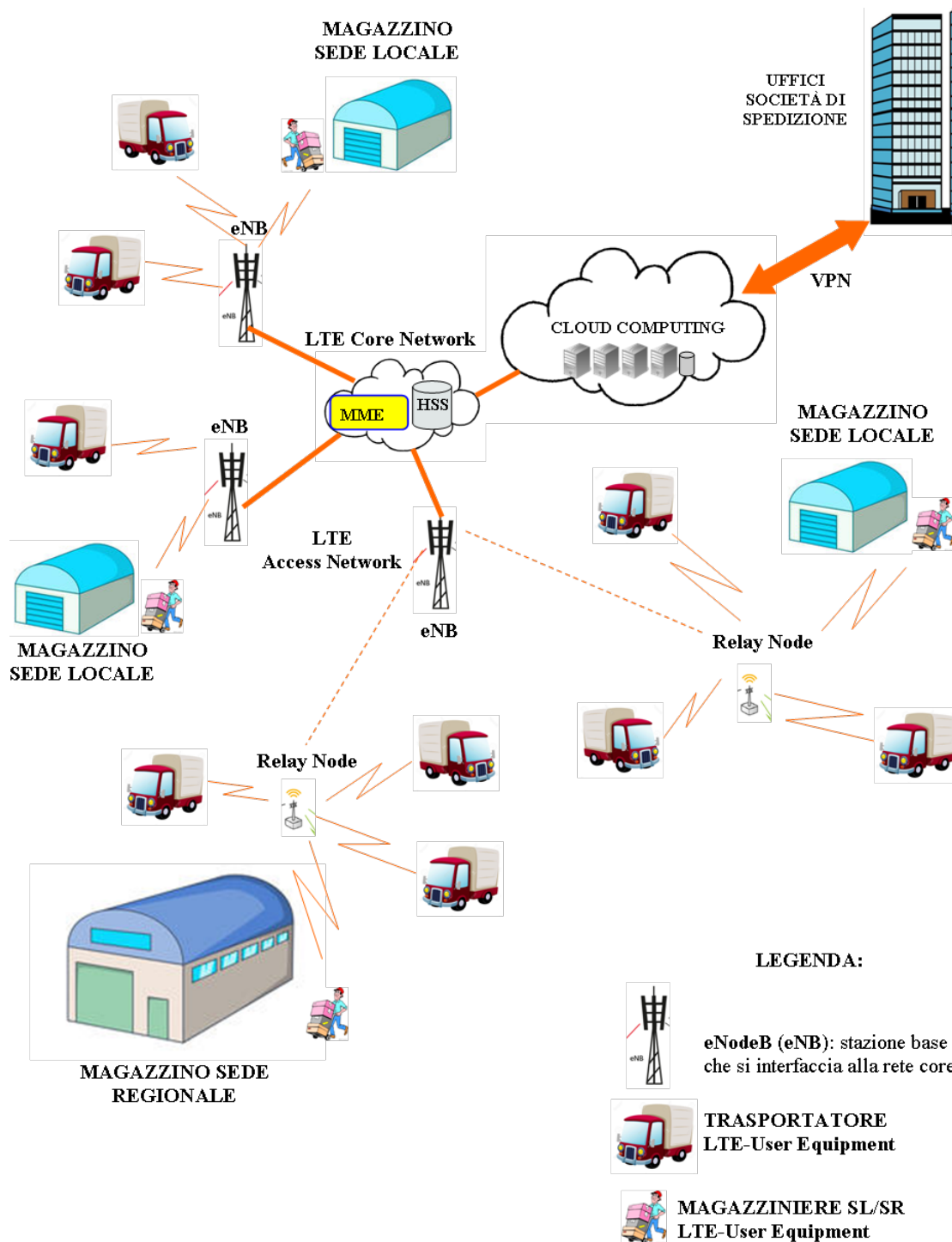


Figura 2 Infrastruttura informatica nello scenario proposto

I dispositivi utilizzati da trasportatori e magazzinieri possono essere dei **mini tablet** dotati di:

- scheda SIM/USIM (Universal Subscriber Identity Module, evoluzione della SIM) per garantire la connettività 4G/LTE-A necessaria ai trasportatori per comunicare in tempo reale i cambiamenti di stato della spedizione;
- scheda di rete Wi-Fi per garantire la connettività wireless in ambito locale, utile ai magazzinieri per comunicare in tempo reale i cambiamenti di stato della spedizione;
- APP per acquisire il QR code tramite la fotocamera digitale del mini tablet;
- GPS integrato per monitorare gli spostamenti degli automezzi in tempo reale;

- APP per acquisire, sul touchscreen del mini tablet, la firma autografa del mittente e del destinatario, nota come firma grafometrica o firma elettronica avanzata (da non confondersi con la firma digitale);
- APP per accedere al servizio in Cloud della SOCIETÀ DI SPEDIZIONI e comunicare i dati e i cambiamenti di stato della spedizione.

La comunicazione tra il sistema mobile dei trasportatori muniti di mini tablet e il Cloud della SOCIETÀ DI SPEDIZIONI avviene tramite la tecnologia **4G/LTE-A**.

Poiché i trasportatori sono in movimento e possono aver bisogno di collegarsi al servizio da qualsiasi luogo, la soluzione migliore è utilizzare la telefonia cellulare con tecnologia 4G/LTE-A.

I due componenti fondamentali di una rete LTE sono la **LTE Access Network** e la **LTE Core Network**.

La LTE Access Network è costituita dal **LTE User Equipment (LTE-UE)**, che è il dispositivo mobile dotato di smart card USIM che contiene i dati identificativi dell'utente utilizzati per l'autenticazione e l'accesso al servizio, e dalla **stazione base eNodeB (eNB)** che si interfaccia alla rete core e che svolge funzioni di sicurezza, tipicamente implementate con tecniche di tunneling e con servizi di autenticazione, integrità e cifratura dei dati.

La LTE Core Network ha invece come elemento principale il **Mobility Management Entity (MME)** che svolge funzioni di controllo, quali, ad esempio, l'autenticazione del dispositivo mobile LTE-UE tramite il database **HSS (Home Subscriber Server)**; si occupa anche della gestione della connessione e degli aspetti di mobilità.

L'evoluzione di LTE (**LTE-Advanced**) ha introdotto alcune importanti funzionalità. Tra queste l'impiego di ripetitori intelligenti, detti **Relay Node (RN)** che, compensando l'attenuazione di propagazione tra utente e stazione base, consentono di aumentare la capacità e la copertura.

L'utilizzo di questa tecnologia consentirà ai trasportatori non solo di comunicare i dati per la tracciatura della missiva (utile a mittente e destinatario) come richiesto nella prima parte del tema al Punto 1, ma anche di monitorare lo spostamento degli automezzi (utile alla SOCIETÀ DI SPEDIZIONI).

I magazzinieri all'interno delle SL e delle SR, anch'essi muniti di mini tablet, sfrutteranno il **Wi-Fi** della LAN aziendale per comunicare col Cloud della SOCIETÀ DI SPEDIZIONI. Per eventuali operazioni da svolgersi al di fuori dall'area coperta dal Wi-Fi, potranno anch'essi utilizzare la rete 4G/LTE-A.

Gli uffici della SOCIETÀ DI SPEDIZIONI saranno connessi in **VPN Site-to-site** col Cloud per garantire l'integrità e la riservatezza delle comunicazioni dei dati sensibili dell'azienda e dei suoi clienti.

Il tunnel VPN consente di creare un canale sicuro e dai percorsi garantiti tra gli uffici e l'infrastruttura Cloud creata in remoto.

Per quanto riguarda l'organizzazione dei server per la raccolta dei dati e l'offerta dei servizi web, abbiamo fin dall'inizio puntato molto per una **soluzione che contemplasse anche l'uso del Cloud**.

I servizi di cloud computing che servono alla SOCIETÀ DI SPEDIZIONI sono di tipo **SaaS** (Software as a Service).

La soluzione SaaS metterà a disposizione in remoto due programmi: uno per gestire i dati in arrivo dalle APP installate sui mini tablet di trasportatori e magazzinieri e metterli a disposizione degli uffici centrali, e uno per gestire il servizio online di registrazione dei clienti e l'inserimento delle richieste di spedizione.

Le motivazioni della scelta di appoggiarsi al cloud computing sono principalmente due:

1. economiche: poiché la SOCIETÀ DI SPEDIZIONI vuole offrire un nuovo servizio online, la soluzione Cloud consente minori investimenti iniziali sui server e sulle infrastrutture informatiche necessarie, assicurando manutenzione e aggiornamenti;
2. strategiche: poiché la SOCIETÀ DI SPEDIZIONI si occupa di spedizioni (trasportatori e magazzinieri), la soluzione Cloud libera risorse umane dell'IT (sistemisti, telecomunicazionisti, programmatori) che possono concentrarsi sul core business dell'azienda.

Nel data center interno all'azienda resterebbero solo i server per la gestione della LAN degli uffici (DHCP, DNS, web, FTP, mail,...) e i database relativi ai dipendenti e alle spedizioni. Sistemi che sicuramente la SOCIETÀ DI SPEDIZIONI aveva già "in casa" prima di decidere di passare alla nuova procedura di gestione informatizzata. Il backup in locale del sito e dei database in Cloud va comunque regolarmente previsto.

Qualora si optasse per una soluzione **totalmente interna all'azienda**, l'investimento, soprattutto quello iniziale, per automatizzare il servizio di tracciamento e gestire le richieste dei clienti, sarebbe oneroso. Inoltre, all'azienda servirebbero competenze e risorse umane che probabilmente non ha al proprio interno e che quindi dovrà procurarsi. Infine tutti gli aggiornamenti in termini di sistema e di sicurezza sarebbero a carico dell'azienda.

PARTE PRIMA – SOLUZIONE DEL PUNTO 3

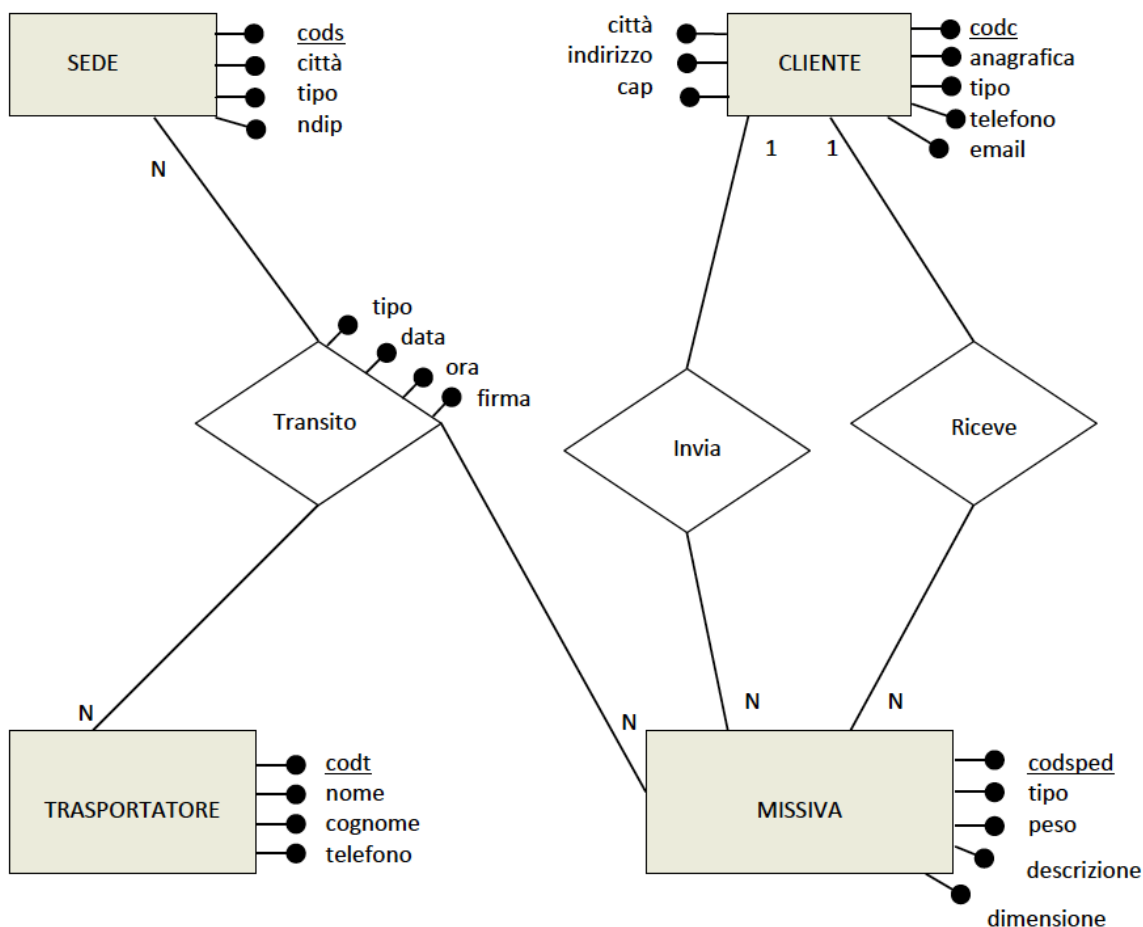


Figura 3 Schema concettuale

Un possibile schema concettuale che rappresenti la soluzione del problema proposto è quello mostrato nella figura 3.

Nello schema ER della figura sono stati riportati i dati relativi alle sedi, ai clienti, alle missive e ai trasportatori. Le associazioni Invia e Riceve tra CLIENTE e MISSIVA è di tipo 1:N, poiché un cliente può fare richiesta di invio (o può ricevere) 1 o più missive, mentre una missiva viene inviata (o ricevuta) da un solo cliente. L'associazione Transito è un'associazione tra tre entità diverse, perché rappresenta il transito che la missiva fa da una sede ad un'altra attraverso un dato trasportatore; l'associazione ha degli attributi, in particolare l'attributo tipo indica se si tratta di un transito in entrata o in uscita dalla sede associata.

Dizionario entità

| Nome entità | Descrizione | Attributi | Identificatore |
|---------------|--|---|----------------|
| SEDE | Sedi sia regionali che locali dove transitano o sono depositate le missive | cods città tipo (locale o regionale) ndip | cods |
| CLIENTE | Clienti della società di spedizione | codc tipo indirizzo città cap telefono email | codc |
| MISSIVA | Lista delle missive prese in carico dalla società di spedizione | codsped tipo descrizione peso dimensione (altezzaXprofonditàXlarghezza) | codsped |
| TRASPORTATORE | Lista dei trasportatori impiegati dalla società | codt nome cognome telefono | codt |

Dal modello concettuale passiamo al modello logico relazionale, applicando le regole di derivazione. Avremo 5 tabelle: 4 corrispondenti alle 4 entità presenti nello schema ER e una tabella TRANSITO corrispondente all'associazione ternaria con attributi. Nella tabella TRANSITO inseriamo gli attributi dell'associazione e le chiavi primarie delle tre entità associate, inseriamo inoltre l'attributo ntransito come chiave primaria. Per realizzare l'associazione Invia tra CLIENTE e MISSIVA di tipo 1:N facciamo migrare la chiave primaria di CLIENTE (codc) in MISSIVA, dove diventa chiave esterna (id_clienteM).

Facciamo lo stesso per realizzare l'associazione Riceve dove codc di cliente diventa chiave esterna (id_clienteD).

Lo schema relazionale ottenuto è il seguente:

SEDE (cods, città, tipo, ndip)

CLIENTE (codc, tipo, telefono, email, indirizzo, città, cap)

MISSIVA (codsped, tipo, peso, dimensione, id_clienteM, id_clienteD)

TRASPORTATORE (codt, nome, cognome, telefono)

TRANSITO (ntransito, data, ora, firma, tipo, id_sede, id_trasportatore, id_spedizione)

Osserviamo che le tabelle dello schema relazionale sono tutte in terza forma normale (3FN).

Descrizione attributi

SEDE

| Nome attributo | Descrizione | Tipo | Lunghezza | Vincoli |
|----------------|---|---------|-----------|------------------|
| cods | Codice della sede | stringa | 15 byte | obbligatorio |
| città | Nome della città dove è situata la sede | stringa | 20 byte | obbligatorio |
| tipo | Tipologia di sede | stringa | 10 byte | Locale/Regionale |
| ndip | Numero di dipendenti che vi lavorano | intero | | |

CLIENTE

| Nome attributo | Descrizione | Tipo | Lunghezza | Vincoli |
|----------------|-------------------------------------|---------|-----------|-------------------------------|
| codc | Codice cliente | stringa | 15 byte | obbligatorio |
| città | Nome della città di residenza | stringa | 20 byte | obbligatorio |
| anagrafico | Cognome e nome (o nome della ditta) | stringa | 30 byte | obbligatorio |
| indirizzo | Via e numero | stringa | 20 byte | obbligatorio |
| cap | Codice di avviamento postale | stringa | 5 byte | Solo cifre |
| tipo | Tipologia di cliente | stringa | 10 byte | Privati cittadini, ditte, ... |
| telefono | Numero di telefono | stringa | 15 byte | obbligatorio |
| email | Indirizzo mail | stringa | 30 Byte | obbligatorio |

TRASPORTATORE

| Nome attributo | Descrizione | Tipo | Lunghezza | Vincoli |
|----------------|-----------------------|---------|-----------|--------------|
| codt | Codice trasportatore | stringa | 15 byte | obbligatorio |
| nome | Nome trasportatore | stringa | 20 byte | obbligatorio |
| cognome | Cognome trasportatore | stringa | 20 byte | obbligatorio |
| telefono | Numero di telefono | stringa | 15 byte | obbligatorio |

MISSIVA

| Nome attributo | Descrizione | Tipo | Lunghezza | Vincoli |
|----------------|--|----------|-----------|--------------------|
| codsped | Codice spedizione | stringa | 15 byte | obbligatorio |
| id_clienteM | Codice cliente che effettua la spedizione (missiva) | stringa | 15 byte | Chiave esterna |
| id_clienteD | Codice cliente che riceve la spedizione (destinatario) | stringa | 15 byte | Chiave esterna |
| tipo | Tipologia di missiva | stringa | 10 byte | Lettera,pacco, ... |
| peso | Peso della missiva | numerico | | Espresso in grammi |
| descrizione | Descrizione della missiva | stringa | 30 Byte | |
| dimensione | Dimensione della missiva | stringa | 30 Byte | |

TRANSITO

| Nome attributo | Descrizione | Tipo | Lunghezza | Vincoli |
|------------------|---|----------------|-----------|----------------|
| ntransito | Codice progressivo del transito | Auto increment | | obbligatorio |
| id_sede | Codice sede di transito | stringa | 15 byte | Chiave esterna |
| id_trasportatore | Codice trasportatore della missiva | stringa | 15 byte | Chiave esterna |
| id_missiva | Codice della missiva | stringa | 15 byte | Chiave esterna |
| ora | Ora del trasporto | Data/ora | | |
| data | Data del trasporto | Data/ora | | |
| firma | Firma del trasportatore o del ricevente | stringa | 30 Byte | |
| tipo | Tipologia di transito | stringa | 10 byte | Entrata/Uscita |

PARTE PRIMA – SOLUZIONE DEL PUNTO 4

Progettare inoltre un applicativo per la gestione via Web delle attività tenendo conto delle seguenti richieste:

- richiesta di una nuova spedizione da parte del cliente;
- possibilità da parte di un cliente, inserendo il codice della spedizione, di conoscere il percorso fatto dalla missiva fino a quel momento.

Per la risoluzione di questo punto si può progettare la struttura dell'applicazione web nel suo insieme e implementare la parte relativa alla visualizzazione del percorso fatto dalla missiva scelta dall'utente. L'applicazione prevede una serie di pagine che si richiamano tra loro e che hanno tutte la stessa struttura composta dal logo e un bottone che permette di tornare alla pagina precedente o alla home page.

Il sito può avere la seguente struttura di massima.

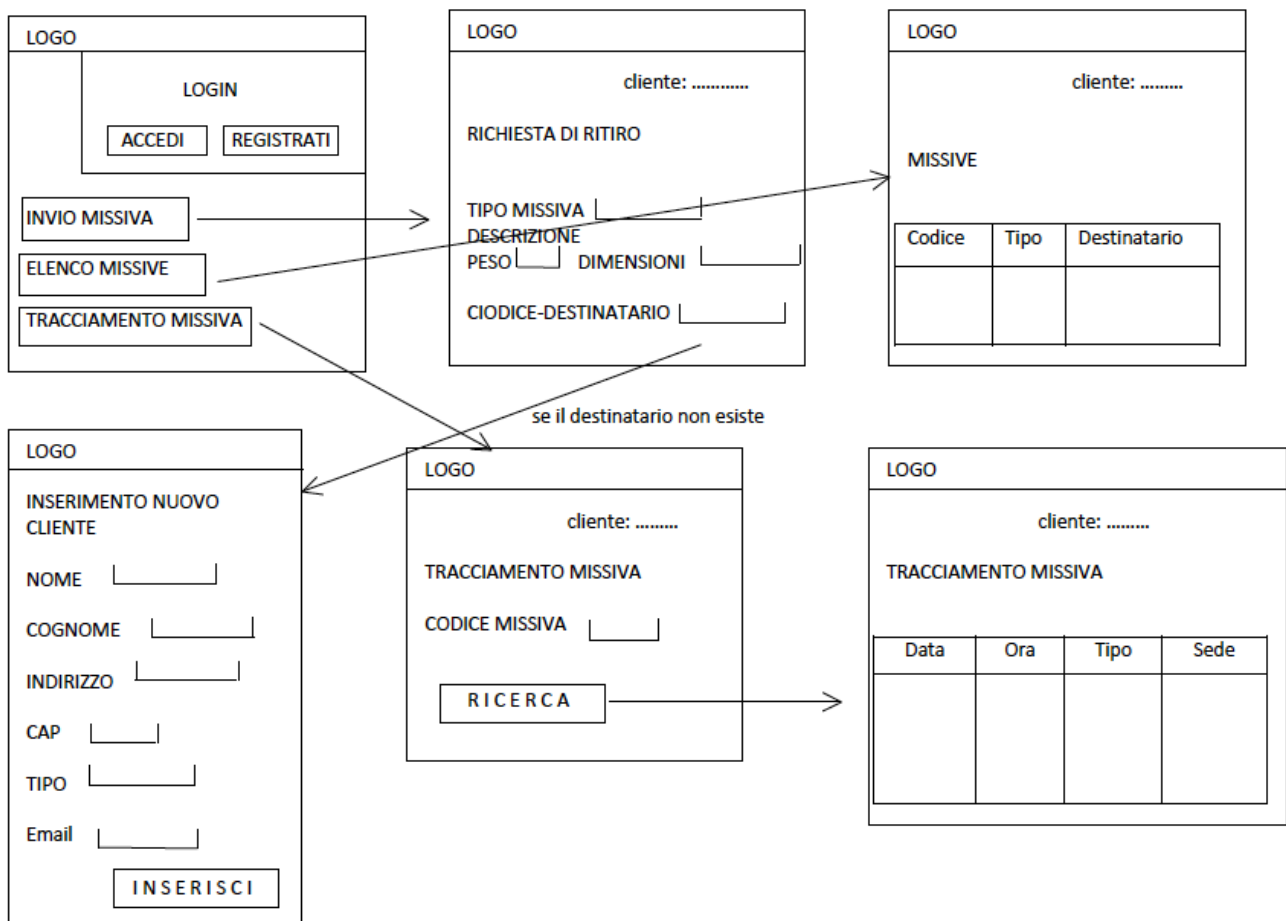


Figura 4 Struttura del sito

Poniamo come ipotesi aggiuntiva che il destinatario della missiva sia già cliente della società e quindi sia già presente nella base di dati. In caso contrario i suoi dati saranno inseriti nel sistema dall'utente che invia la missiva.

Riportiamo la codifica lato client e lato server della visualizzazione del percorso della missiva per la quale il cliente inserisce il codice. Il codice identificativo del cliente è già noto, poiché questa è solo una parte dell'applicazione e si arriva a questa pagina dopo aver fatto il login.

Codifica PHP

File index.html

```
<html>
<head></head>
<body>
<p align = "center"><font size = '5'><strong>Ricerca spedizione<br><br><br></p>
<form method = 'post' action = 'codice.php'>
<p align = "center"><font size = '5'>
Codice spedizione: <input type = 'text' name = 'codice' size = '10'>
<br><br><br>
<input type = 'submit' style="height:50px;" value = 'Visualizza'> &nbsp; &nbsp; &nbsp; &nbsp;
<input type = 'reset' style="height:50px;" value = 'Annulla'>
</p>
</form>
</body>
</html>
```

File codice.php

```
<?php
$hostname = "localhost";
$username = "root";
$password = "";
$dbname = "spedizioni";

//connessione al server sql
$conn = mysqli_connect($hostname, $username, $password,$dbname);
if(!$conn)
    die ("errore nella connessione");

//recupera i dati passati dal form html
$codice = $_POST['codice'];
$query = "SELECT tipo,peso,dimensione,descrizione FROM Missiva Where codsped = '$codice'";
$resultato= mysqli_query($conn,$query);
if (!$resultato)
{
    print "errore nel comando";
    exit();
}
$riga = mysqli_fetch_array($resultato);
if($riga){
    print "tipo: ".$riga['tipo']."<br>";
    print "peso: ".$riga['peso']."<br>";
    print "descrizione: ".$riga['descrizione']."<br>";
    print "dimensione: ".$riga['dimensione']."<br>";
    $query = "SELECT ntransito, data, ora, Transito.tipo, citta FROM Transito, Sede";
    $query = $query ." WHERE cods = id_sede and id_missiva = '$codice'";
    $resultato= mysqli_query($conn,$query);
    if (!$resultato){
        print "errore nel comando";
        exit();
    }
    $riga = mysqli_fetch_array($resultato);
```

```

        if($riga){
            print"<table border = '1'>";
            print " <tr> <td> Data </td> <td> Ora </td> <td> Tipo </td> <td> Sede </td> </tr>";
            while($riga) {
                print "<tr><td>".$riga['data']. "</td>";
                print "<td> ".$riga['ora']. "</td>";
                print "<td> ".$riga['tipo']. "</td>";
                print "<td> ".$riga['citta']. "</td></tr>";
                $riga = mysqli_fetch_array($risultato);
            }
            print "</table>";
        }
        else
            print "Attenzione!!! non risultano transiti per la spedizione $codice ";
    }
    else
        print "Attenzione!!! codice spedizione $codice non presente";
    mysqli_close($conn);
?>

```

Codifica C#.NET

Ricerca.aspx.cs (database Access)

```

using System;
using System.Data;
using System.Data.OleDb;

namespace Tracciamento
{
    public partial class Ricerca : System.Web.UI.Page
    {
        OleDbConnection connDb = new OleDbConnection();
        OleDbCommand command = new OleDbCommand();
        OleDbDataReader rs;

        protected void Page_Load(object sender, EventArgs e)
        {
            // Le seguenti variabili di sessione devono essere impostate
            // in fase di login
            Session["codc"] = "cc101";
            Session["anagrafica"] = "Rossi Carlo";
            lblCliente.Text = Session["anagrafica"].ToString();
            lblErrore.Text = " ";
        }
        protected void btnRicerca_Click(object sender, EventArgs e)
        {
            string codMissiva = TxtMissiva.Text;
            connDb.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;" +
                "Data Source=|DataDirectory|/missive.accdb";
            connDb.Open();
            string query = "SELECT codsped " +
                "FROM Missiva " +
                "where codsped = '" + codMissiva + "'";
            command.Connection = connDb;
            command.CommandText = query;

```

```

        rs = command.ExecuteReader();
        if (rs.Read())
        {
            Session["missiva"] = codMissiva;
            Response.Redirect("ElencoMissive.aspx");
        }
        else
        {
            lblErrore.Text = "Errore -- Missiva inesistente";
        }
    }
}

```

ElencoMissive.ASPX

```

using System;
using System.Data;
using System.Data.OleDb;

namespace Tracciamento
{
    public partial class ElencoMissive : System.Web.UI.Page
    {
        OleDbConnection connDb = new OleDbConnection();
        OleDbCommand command = new OleDbCommand();
        OleDbDataReader rs;
        string codMissiva;

        protected void Page_Load(object sender, EventArgs e)
        {
            codMissiva = Session["missiva"].ToString();
            lblCliente.Text = Session["anagrafica"].ToString();
            lblMissiva.Text = Session["missiva"].ToString();
            connDb.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;" +
                "Data Source=|DataDirectory|/missive.accdb";
            connDb.Open();
            string query = "SELECT tipo,peso,dimensione,descrizione " +
                "FROM Missiva " +
                "where codsped = '" + codMissiva + "'";
            command.Connection = connDb;
            command.CommandText = query;
            rs = command.ExecuteReader();
            if (rs.Read())
            {
                lblDescrizione.Text = rs["descrizione"].ToString();
                lblPeso.Text = rs["peso"].ToString();
                lblDimensione.Text = rs["dimensione"].ToString();
                lblTipo.Text = rs["tipo"].ToString();

                rs.Close();
                Visualizza_Tracciamenti();
            }
            else
            {
                lblErrore.Text = "Errore";
            }
        }

        private void Visualizza_Tracciamenti()
        {
            DataTable dt = new DataTable();
            string query = "SELECT ntransito,data,ora,Transito.tipo,citta " +
                "from Transito,Sede where cods = id_sede " +
                "and id_spedizione = '" + codMissiva + "'";

```



```

command.CommandText = query;
rs = command.ExecuteReader();
if (rs.HasRows)
{
    dt.Columns.Add("Transito");
    dt.Columns.Add("Data");
    dt.Columns.Add("Ora");
    dt.Columns.Add("Tipo");
    dt.Columns.Add("Citta");
    while (rs.Read())
    {
        DataRow row = dt.NewRow();
        row["Transito"] = rs["ntransito"];
        // Estrae il giorno dal campo data/ora
        DateTime dataT = Convert.ToDateTime(rs["data"].ToString());
        row["Data"] = dataT.ToString("d");
        //Estrae l'orario dal campo data/ora
        dataT = Convert.ToDateTime(rs["ora"].ToString());
        row["Ora"] = dataT.ToString("t");
        row["Tipo"] = rs["tipo"];
        row["Citta"] = rs["citta"];
        dt.Rows.Add(row);
    }
    GridView1.DataSource = dt;
    GridView1.DataBind();
    rs.Close();
    connDb.Close();
}
}
}
}

```

Ricerca.aspx.cs (database SQL Server)

```

using System;
using System.Data.SqlClient;
using System.Data;

public partial class Ricerca : System.Web.UI.Page
{
    namespace Tracciamento
    {
        SqlConnection connDb = new SqlConnection();
        SqlCommand command = new SqlCommand();
        SqlDataReader rs;

        protected void btnRicerca_Click(object sender, EventArgs e)
        {
            string codMissiva = TxtMissiva.Text;
            connDb.ConnectionString = "Data Source = (LocalDB)\\MSSQLLocalDB; " +
                "AttachDbFileName=|DataDirectory|missive.mdf; Initial Catalog = " +
                "Carpooling.mdf";
            connDb.Open();
            string query = "SELECT codsped " +
                "FROM Missiva " +
                "where codsped = '" + codMissiva + "'";
            command.Connection = connDb;
            command.CommandText = query;
            rs = command.ExecuteReader();
            if (rs.Read())
            {
                Session["missiva"] = codMissiva;
            }
        }
    }
}

```

```

        Response.Redirect("ElencoMissive.aspx");
    }
    else
        lblErrore.Text = "Errore -- Missiva inesistente";
}

}
}

```

ElencoMissive.aspx.cs

```

using System;
using System.Data.SqlClient;
using System.Data;
namespace Tracciamento
{
    public partial class ElencoMissive : System.Web.UI.Page
    {
        SqlConnection connDb = new SqlConnection();
        SqlCommand command = new SqlCommand();
        SqlDataReader rs;
        string codMissiva;

        protected void Page_Load(object sender, EventArgs e)
        {
            codMissiva = Session["missive"].ToString();
            lblCliente.Text = Session["anagrafica"].ToString();
            lblMissiva.Text = Session["missive"].ToString();
            connDb.ConnectionString = "Data Source = (LocalDB)\\MSSQLLocalDB; " +
                "AttachDbFileName=|DataDirectory|missive.mdf; Initial Catalog = " +
                "Carpooling.mdf";
            connDb.Open();
            string query = "SELECT tipo,peso,dimensione,descrizione,id_clienteD " +
                "FROM Missiva " +
                "where codsped = '" + codMissiva + "'";
            command.Connection = connDb;
            command.CommandText = query;
            rs = command.ExecuteReader();
            if (rs.Read())
            {
                lblDescrizione.Text = rs["descrizione"].ToString();
                lblPeso.Text = rs["peso"].ToString();
                lblDimensione.Text = rs["dimensione"].ToString();
                lblTipo.Text = rs["tipo"].ToString();
                rs.Close();
                Visualizza_Tracciamenti();
            }
            else
                lblErrore.Text = "Errore";
        }

        private void Visualizza_Tracciamenti()
        {
            DataTable dt = new DataTable();
            string query = "SELECT data,ora,Transito.tipo,citta " +
                "from Transito,Sede where cods = id_sede " +
                "and id_spedizione = '" + codMissiva + "'";
            command.CommandText = query;
            rs = command.ExecuteReader();
            if (rs.HasRows)
            {

```

```

dt.Columns.Add("Data");
dt.Columns.Add("Ora");
dt.Columns.Add("Tipo");
dt.Columns.Add("Citta");

while (rs.Read())
{
    DataRow row = dt.NewRow();
    // Estrae il giorno dal campo data/ora
    DateTime dataT = Convert.ToDateTime(rs["data"].ToString());
    row["Data"] = dataT.ToString("d");
    //Estrae l'orario dal campo data/ora
    dataT = Convert.ToDateTime(rs["ora"].ToString());
    row["Ora"] = dataT.ToString("t");
    row["Tipo"] = rs["tipo"];
    row["Citta"] = rs["citta"];
    dt.Rows.Add(row);
}
GridView1.DataSource = dt;
GridView1.DataBind();
rs.Close();
connDb.Close();
}
}
}
}

```

```

public partial class Default : System.Web.UI.Page
{
    namespace Tracciamento
    {
        SqlConnection connDb = new SqlConnection();
        SqlCommand command = new SqlCommand();
        SqlDataReader rs;
        string cod_Missiva;

        protected void BtnElenco_Click(object sender, EventArgs e)
        {
            cod_Missiva = TxtMissiva.Text;
            connDb.ConnectionString = "Data Source = (LocalDB)\\MSSQLLocalDB; " +
                " AttachDbFileName =|DataDirectory|missive.mdf; Initial Catalog = " +
                " Carpooling.mdf";
            connDb.Open();
            string query = "SELECT tipo, peso, dimensione,descrizione,codc,email " +
                "FROM Missiva,Cliente " +
                "where codsped = '" + cod_Missiva + "'";
            command.Connection = connDb;
            command.CommandText = query;
            rs = command.ExecuteReader();
            if (rs.Read())
            {
                lblDescrizione.Text = rs["descrizione"].ToString();
                lblPeso.Text = rs["peso"].ToString() ;
            }
        }
    }
}

```

```

        lblDimensione.Text = rs["dimensione"].ToString();
        lblTipo.Text = rs["tipo"].ToString();
        lblCliente.Text = rs["codc"].ToString();
        lblMail.Text = rs["email"].ToString();

        rs.Close();
        Visualizza_Tracciamenti();
    }
    else
        lblErrore.Text = "Codice missiva inesistente";
}

private void Visualizza_Tracciamenti()
{
    DataTable dt = new DataTable();
    string query = "SELECT ntransito,data,ora,Transito.tipo,citta " +
        "FROM Transito,Sede where cods = id_sede " +
        " and id_spedizione = '" + cod_Missiva + "'";
    command.CommandText = query;
    rs = command.ExecuteReader();
    if (rs.HasRows)
    {
        dt.Columns.Add("Transito");
        dt.Columns.Add("Data");
        dt.Columns.Add("Ora");
        dt.Columns.Add("Tipo");
        dt.Columns.Add("Citta");
        while (rs.Read())
        {
            DataRow row = dt.NewRow();
            row["Transito"] = rs["ntransito"];
            // Estrae il giorno dal campo data/ora
            DateTime dataT = Convert.ToDateTime(rs["data"].ToString());
            row["Data"] = dataT.ToString("d");
            //Estrae l'orario dal campo data/ora
            dataT = Convert.ToDateTime(rs["ora"].ToString());
            row["Ora"] = dataT.ToString("t");
            row["Tipo"] = rs["tipo"];
            row["Citta"] = rs["citta"];
            dt.Rows.Add(row);
        }
        GridView1.DataSource = dt;
        GridView1.DataBind();
        rs.Close();
        connDb.Close();
    }
    else
        lblErrore.Text = "Errore ";
}
}
}

```

ESECUZIONE

TRACCIAMENTO MISSIVA cliente: Rossi Carlo

CODICE MISSIVA:

Ricerca

TRACCIAMENTO MISSIVA cliente: Rossi Carlo

missiva: 101

tipo missiva: pacco

descrizione: pacco fragile

peso: gr. 800

dimensione: cm 300 x 400

| Data | Ora | Tipo | Sede |
|------------|-------|---------|-----------|
| 02/03/2019 | 10:00 | entrata | Nichelino |
| 02/03/2019 | 14:00 | uscita | Nichelino |
| 04/03/2019 | 11:00 | entrata | Pescara |
| 06/03/2019 | 13:15 | uscita | Pescara |
| 08/03/2019 | 12:50 | entata | Lanciano |

SOLUZIONI DEI QUESITI DELLA PARTE SECONDA

PARTE SECONDA – SOLUZIONE DEL QUESITO 1

Quesito 1 – Data la seguente tabella PRODOTTO (CodProdotto, DescrizioneProdotto, CodMagazzino, QuantitàProdotto, IndirizzoMagazzino, Magazziniere) dire se è in terza forma normale, e, in caso contrario, spiegarne i motivi. Trasformarla poi in un insieme di tabelle in terza forma normale.

Verifichiamo se la tabella PRODOTTO è in terza forma normale. Una tabella del modello relazionale è in 3FN quando è in 2FN e gli attributi non chiave non dipendono transitivamente dalla chiave primaria.

La chiave primaria della tabella PRODOTTO è una chiave composta formata dagli attributi CodProdotto e CodMagazzino

PRODOTTO (CodProdotto, DescrizioneProdotto, CodMagazzino, QuantitàProdotto, IndirizzoMagazzino, Magazziniere)

Tale tabella non è in 2FN, poiché vi sono degli attributi che non dipendono dall'intera chiave, ma solo da una sua parte. In particolare l'attributo DescrizioneProdotto dipende solo da CodProdotto, gli attributi IndirizzoMagazzino e Magazziniere dipendono solo da CodMagazzino e l'attributo QuantitàProdotto dipende da tutta la chiave. Per rendere la relazione in 2FN dobbiamo spezzarla nelle seguenti relazioni:

PRODOTTO (CodProdotto, DescrizioneProdotto)

MAGAZZINO (CodMagazzino, IndirizzoMagazzino, Magazziniere)

PROMAG (CodProdotto, CodMagazzino, QuantitàProdotto)

Le tre relazioni sono in 3FN, perché sono in 2FN e non vi sono dipendenze transitive.

PARTE SECONDA – SOLUZIONE DEL QUESITO 2

Quesito 2 – Il servizio di Virtual Private Network (VPN) offre la possibilità di connettersi da remoto, in modo sicuro, alle risorse dell'azienda. Descrivere quali meccanismi di sicurezza sono tipicamente implementati in una VPN e fornire alcuni esempi di protocolli utilizzati per realizzare le VPN.

Una **VPN** (Virtual Private Network) è una rete privata creata all'interno di un'infrastruttura di rete pubblica. Una **Remote-access VPN** consente ai singoli utenti di stabilire connessioni sicure con la intranet aziendale, collegandosi da casa, dalla sede di un cliente o fornitore. Gli utenti possono accedere alle risorse protette della rete locale, come se fossero direttamente collegati ai server della rete.

Dal momento che una VPN utilizza una rete pubblica, si devono affrontare alcuni importanti problemi legati alla trasmissione dei dati e alla loro riservatezza:

1. la variabilità del tempo di trasferimento (traffico, congestione, latenza, velocità variabili, jitter, perdita di pacchetti, ecc.): l'impiego di tecniche di QoS (Quality of Service) consente di trattare il traffico in rete in modo da disciplinare il flusso dei pacchetti;

2. il controllo degli accessi alla VPN e la sicurezza delle trasmissioni: questi problemi vengono affrontati impiegando una combinazione di tre fattori: **autenticazione, cifratura e tunneling**.

L'autenticazione consente solo alle persone autorizzate di accedere alla VPN; solitamente oltre al classico login con username e password si richiede di immettere un pin generato da un key fob oppure si usano applicazioni per la lettura di un QR code sulla pagina di autenticazione.

La cifratura si occupa di rendere illeggibili i dati, così da garantirne la segretezza. Le VPN usano vari algoritmi di cifratura (3-DES, IDEA, ...). Vengono utilizzati dei protocolli di sicurezza per lo scambio delle chiavi di cifratura: un esempio è IKE (Internet Key Exchange) della suite di protocolli IPsec.

Il tunneling è una tecnica che consiste nell'incapsulare i pacchetti da trasmettere sulla VPN in altri pacchetti, in modo tale che il pacchetto esterno protegge tutto il contenuto del pacchetto interno, che potrà essere completamente cifrato (inclusi i dati dell'header).

Tipicamente per realizzare le VPN si usano i protocolli:

- **IPsec** (IP security): è un'architettura di sicurezza a livello Network, molto usata dalle aziende, che si compone di più protocolli: AH (Authentication Header), che garantisce l'autenticazione e l'integrità del messaggio, ma non la confidenzialità; ESP (Encapsulating Security Payload), che fornisce autenticazione, confidenzialità e integrità del messaggio; IKE (Internet Key Exchange) usato per lo scambio delle chiavi di crittografia;
- **SSL/TLS** (Secure Sockets Layer/Transport Layer Security) di tipo Client/Server, che hanno lo scopo di autenticare il server da parte del client e di creare un canale cifrato, sicuro, di comunicazione tra i due. Si tratta di protocolli che operano a livello Transport, usati per proteggere le comunicazioni tra due applicazioni. Infatti, sono tipicamente impiegati in ambito web (HTTPS), dove l'identità nelle connessioni può essere autenticata usando la crittografia asimmetrica (esempio: RSA). È prevista la certificazione del server e, opzionalmente, quella del client. In questo contesto il certificato digitale è il documento elettronico che associa l'identità del server alla chiave pubblica;
- **MPLS** (Multi-Protocol Label Switching): questa modalità di creare le VPN è tipica degli ISP (Internet Service Provider) ed è offerta come servizio gestito in ambito di reti IP. Questo perché MPLS necessita della rete circoscritta di un service provider che ne fornisca l'infrastruttura di supporto.

PARTE SECONDA – SOLUZIONE DEL QUESITO 3

Quesito 3 – Date le due seguenti tabelle:

EDITORE (idEditore, nome, sede, città)

LIBRO (idLibro, titolo, autore, prezzo, genere, codEditore)

scrivere in SQL le query che permettano di:

1. visualizzare il numero di libri del genere "giallo" edito dalla casa editrice fornita in Input;
2. visualizzare il titolo del libro scritto dall'autore fornito in Input che ha il prezzo maggiore.

Query 1)

Nel codice delle query le parentesi quadre indicano i valori forniti dall'utente.

```
SELECT count(*) as numero
FROM libro, editore
```

```
WHERE libro.codEditore = editore.idEditore  
and genere = "giallo" and nome = [nome editore]
```

Query2)

```
SELECT titolo, prezzo  
FROM libro  
WHERE prezzo = (select max(prezzo)  
                from libro  
                where autore = [autore])  
and autore = [autore]
```

PARTE SECONDA – SOLUZIONE DEL QUESITO 4

Quesito 4 – In relazione al tema proposto nella prima parte, si approfondiscano gli aspetti legati alla sicurezza delle strumentazioni, dei dati gestiti e del servizio offerto e si analizzino le misure utili a garantire la continuità del servizio.

La **sicurezza delle strumentazioni**, in particolare i mini tablet forniti ai trasportatori e ai magazzinieri, viene garantita impostando l'accesso al dispositivo con password o impronta digitale.

Anche l'accesso ai server in Cloud viene strettamente controllato. L'utilizzo di un server AAA risponde allo scopo. L'acronimo indica i tre compiti: il server AAA conferma chi sei (authentication), identifica ciò a cui ti è permesso accedere tramite la connessione (authorization) e tiene traccia di ciò che fai mentre sei loggato (accounting).

La **sicurezza dei dati e la tolleranza ai guasti** che possono comportare una perdita di dati, è garantita dal fornitore del servizio Cloud, acquistando il backup automatico del sito e del database. Una buona pratica che la società di spedizioni metterà in atto sarà accedere regolarmente al Cloud in remoto per fare una copia in locale di sito e database.

Per quanto riguarda le misure utili per garantire la **continuità del servizio** online della società di spedizioni (tracciamento delle spedizioni e gestione delle richieste dei clienti), la soluzione Cloud, affidata ad aziende specializzate in cloud computing, può dare ampie garanzie (servizi di hosting a pagamento garantiscono la continuità del servizio).

In ogni caso, qualunque sia la soluzione scelta, l'interruzione di un servizio offerto online può essere dovuta a diversi fattori da cui ci si deve proteggere:

- interruzione dell'alimentazione: si previene fornendo la rete di gruppi di continuità;
- guasto alle macchine server: si riesce a limitare la durata dell'interruzione utilizzando server di back up ridondanti o, se i server sono virtualizzati, clonandoli;
- interruzione della connessione Internet: si previene mediante una seconda linea gestita, eventualmente, da un ISP (Internet Service Provider) diverso dal gestore della linea principale;
- eccesso di traffico con punti di congestione e successiva perdita di pacchetti: si tiene sotto controllo il traffico mediante una serie di strumenti di network management che permettono di avere una supervisione dell'intera rete; sono possibili SLA (Service-Level Agreement) con gli ISP per garantirsi una voluta QoS (Quality of Service) in termini di larghezza di banda e throughput.

- server sovraccarico di richieste con rallentamento e successivo blocco del servizio (*system crash*) in certi orari: anche in questo caso sono i Network Management Tools che permettono di monitorare gli accessi al server e in generale alle risorse di rete e quindi permettono di intervenire tempestivamente. Se il rallentamento si verifica di frequente, una soluzione può essere decidere per un server dedicato (non condiviso) da configurare su una macchina server dalle elevate prestazioni. Se invece il rallentamento si verifica occasionalmente, è sufficiente l'intervento dell'amministratore di rete che può, ad esempio, diminuire il TTL (Time To Live) dei pacchetti in attesa. Fondamentale è anche il controllo del throughput della banda di trasmissione da parte dell'amministratore della rete (*bandwidth management*);
- attacchi informatici: gli attacchi tipo DoS (Denial of Service), che mirano a bombardare il server di richieste fino a bloccarlo completamente, si possono anch'essi limitare mediante l'utilizzo dei Network Management Tools. Il filtraggio dei pacchetti in arrivo (campionamento e analisi) consente di individuare in tempo utile i flussi da bloccare.