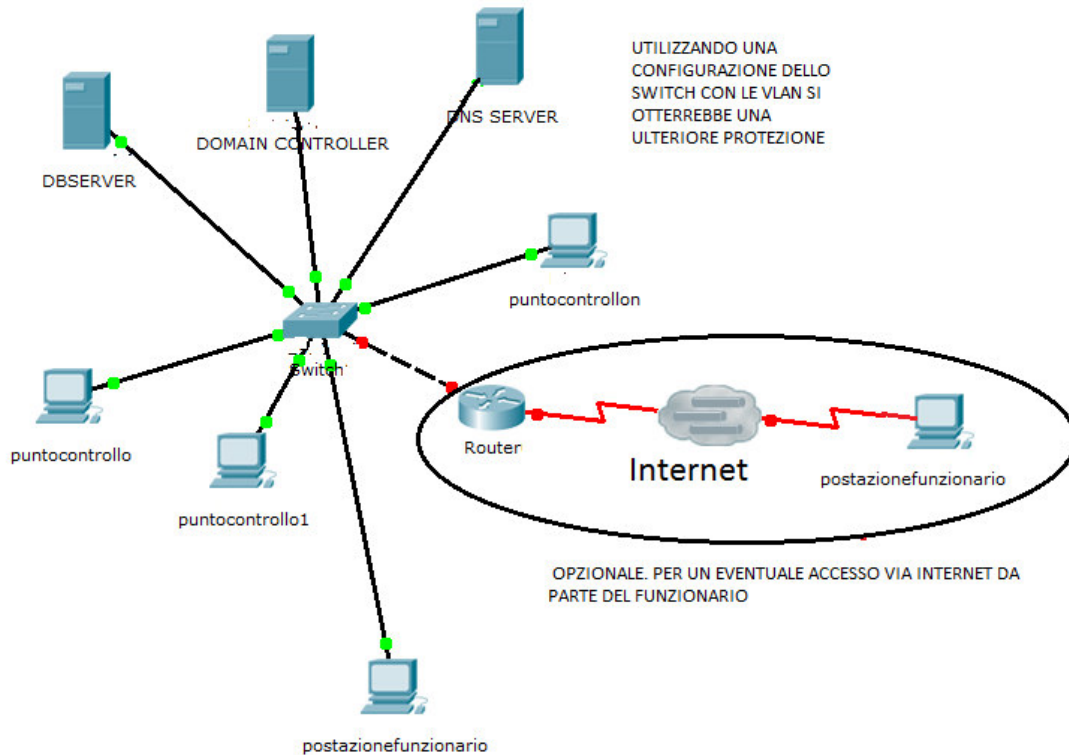


Il tema propone un classico problema di database condiviso mediante una rete locale. Infatti è logico ipotizzare, anche per evidenti motivi di sicurezza, che i vari punti di dogana siano collegati tra loro mediante una LAN dedicata in modo da ridurre al minimo la possibilità di intrusione da parte di estranei.



Ovviamente sarà a carico del responsabile della rete provvedere ad un controllo sistematico degli accessi e ad una revisione periodica delle password e delle user id dei singoli addetti e funzionari.

Ovviamente, nella creazione del database in Access bisogna prestare particolare attenzione ai permessi di accesso alle singole tabelle, maschere e query in maniera da assicurare il livello più alto di sicurezza.

Infine ipotizziamo che sia il funzionario mediante opportune procedure precostituite che vengono attivate manualmente a gestire la parte relativa all'apertura delle pratiche di fermo e all'inoltro dei dati alle autorità competenti.

**Una semplice osservazione:** La traccia non chiarisce fino in fondo il compito del funzionario e degli addetti alla dogana.

Sembra comunque alquanto improbabile che in una situazione reale l'addetto al controllo non possa interagire direttamente con il database per inserire personalmente i dati relativi al controllo effettuato, ma debba passare tutti i dati (in forma cartacea?) al funzionario che provvederà (difficilmente in tempo reale) a inserirli nel database. Si potrebbe quindi ipotizzare che l'addetto in prima battuta inserisca nel database i dati relativi al controllo effettuato e sia poi il funzionario a decidere l'azione da intraprendere.

Un'altra interpretazione potrebbe invece portare a pensare che il funzionario non debba provvedere personalmente alla gestione dei singoli controlli, che rimarrebbe a carico dell'addetto alla dogana, mentre gestirebbe in tempo reale l'inserimento del data base dei parametri di riferimento che l'addetto dovrà poi utilizzare durante il proprio lavoro.

Per mantenerci il più possibile coerenti con la traccia, abbiamo fatto riferimento alla prima ipotesi inserendo forzatamente nel modello logico funzionale sia l'entità Funzionario che l'entità Addetto prevedendo però una relazione diretta solo tra il funzionario e il controllo, seguendo l'ipotesi che ogni funzionario operi come un capoufficio a cui ogni giorno vengono assegnati diversi addetti che a lui devono fare riferimento.

Ciò ci consente, tra l'altro, di rispondere adeguatamente ai 7 punti successivi

Partendo da queste ipotesi è possibile realizzare l'intero sistema software sfruttando le potenzialità di uno dei DBMS più diffusi nelle realtà scolastiche come Access.

In questo contesto le entità del modello logico concettuale del database possono essere definite nel modo seguente:

**Passeggero** (biglietto, cognome, nome, nazionalità, tipo documento, numero documento, provenienza, destinazione, motivo)

**Merce** (biglietto, codice progressivo, genere di appartenenza, descrizione, quantità)

**Controllo** (numero progressivo, punto di controllo, codice addetto, codice funzionario, data controllo, ora controllo, durata controllo, codice progressivo merce, esito, dazio dovuto, dazio incassato, note)

L'esito del controllo riporterà il valore 0 nel caso di nessuna contestazione, 1 nel caso di Passeggero in stato di fermo e così via.

**Genere** (codice, descrizione)

**Codice esito** (codice numerico, descrizione)

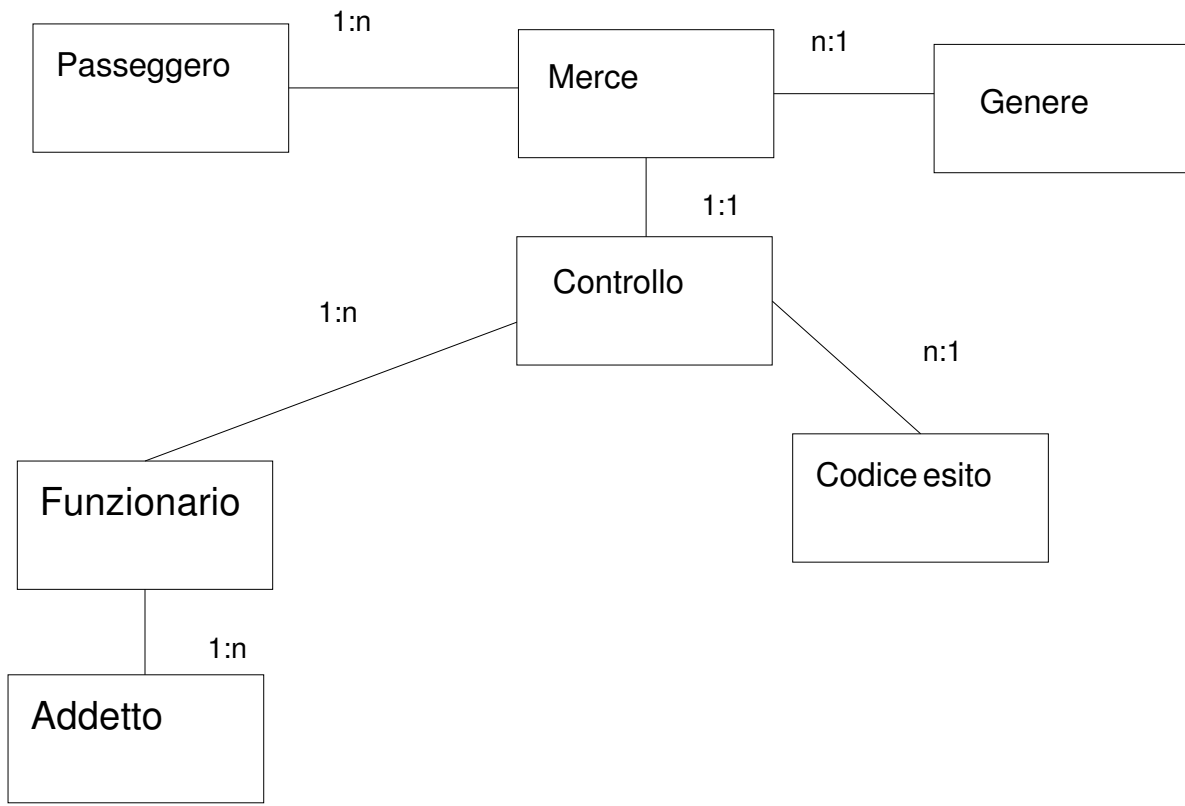
**Addetto** (codice identificativo, nominativo, abilitato, codice funzionario)

**Funzionario** (codice funzionario, nominativo, password assegnata)

In particolare:

- nell'entità Passeggero è stato inserito come chiave primaria il numero del biglietto di viaggio in quanto permette l'identificazione univoca del passeggero e può essere usato più facilmente come chiave esterna nelle relazioni tra le entità.
- Nelle entità Merce e Controllo si è optato per una chiave primaria progressiva che nel sistema può essere generata automaticamente in modo da evitare qualsiasi errore da parte dell'operatore
- Per ottimizzare lo spazio occupato si è inserita un'entità Genere e un'entità Codice esito in modo da non dover continuamente duplicare i dati
- Le entità Funzionario e Addetto permettono al responsabile di aggiornare, in tempo reale, l'elenco degli addetti ai punti di controllo abilitati ad interagire con il database.

Le relazioni esistenti tra le varie entità possono essere rappresentate dal seguente grafico:



Per quanto riguarda i quesiti veri e propri osserviamo che in generale si tratta di query a campi incrociati contenenti elementi di raggruppamento, di conteggio o di calcolo.

Riportiamo di seguito quelle che riteniamo maggiormente significative:

A titolo esemplificativo, indichiamo la formulazione della relazione SQL che consente di creare la tabella **Merce**.

Create Table Merce

```
(codice progressivo integer unsigned auto_increment primary key,  
biglietto char(20) foreign key references Passeggero  
on delete restrict  
on update cascade,  
genere di appartenenza char(20) not null,  
descrizione char(20) not null,  
quantita' double not null);
```

Passiamo ora a sviluppare alcune delle query richieste dal testo.

```
Select Passeggero.*  
from Passeggero, Merce, Controllo  
Where passeggero.biglietto=Merce.biglietto and  
Merce.codice progressivo=Controllo. codice progressivo merce and  
data controllo = CURRENT_DATE;
```

```
Select Controllo.punto di controllo, SUM(dazio incassato) from Controllo  
group by punto di controllo;
```

```
Select COUNT(codice progressivo), genere di appartenenza  
from Merce, Controllo  
where Merce.codice progressivo=Controllo. codice progressivo merce and  
data controllo >#1/1/2013# and esito=2  
group by genere di appartenenza;
```

```
Select AVG(durata controllo), punto di controllo  
from Controllo  
group by punto di controllo and  
data controllo = CURRENT_DATE;
```

```
Select Passeggero.*, nazionalita'  
from Passeggero, Merce, Controllo  
Where passeggero.biglietto=Merce.biglietto and  
Merce.codice progressivo=Controllo. codice progressivo merce and  
data controllo >#1/1/2013# and  
controllo.esito=1  
group by nazionalita'  
order by Passeggero.Cognome, Passeggero.Nome;
```

L'ultima richiesta della traccia, infine, riguarda lo sviluppo di una procedura significativa utilizzando un linguaggio di programmazione.

Escludendo gli aspetti legati alla gestione delle interfacce interne del database ci sembra che una delle questioni più importanti da gestire sia quella legata ai controlli relativi ai permessi di accesso.

Pur senza entrare nel merito dalle scelte che ogni candidato più effettuare circa la logica da seguire per il controllo e l'uso del linguaggio di programmazione vogliamo semplicemente segnalare che la soluzione può essere sviluppata sia in ambiente di database sfruttando le potenzialità degli oggetti messi a disposizione da Access e del linguaggio Visual Basic sia in ambiente Web utilizzando le interfacce HTML e script in Java.