

Simulazione della seconda prova d'esame

Materia: Informatica

Un terminal marittimo deve realizzare un sistema informatico per la gestione del traffico delle merci che arrivano ad esso via nave e che da qui vengono trasportate a destinazione via camion. Il terminal gestisce varie tipologie di merci.

Ogni nave, di cui si conosce il nome, effettua nel corso del tempo dei viaggi ad ognuno dei quali è caratterizzato da un porto e una data di partenza, un porto di arrivo e la data di allibramento corrispondente all'arrivo della merce in porto con conseguente scarico e immagazzinaggio delle merci nel terminal. Il porto di partenza determina la linea relativa alla direttrice del viaggio (Sudest Asiatico, Nord America, Sud America, Brasile, ecc) gestita principalmente a scopo statistico. Di ogni porto interessa osservare anche la nazionalità.

Per ogni viaggio di una nave, la merce trasportata afferisce ad un insieme di documenti dette polizze di carico. Per ogni polizza di carico si conoscono il porto di carico e quello di destinazione, la tipologia della merce trasportata, il peso (in kg) della medesima e il suo fornitore.

Il personale del terminal emette, su richiesta dei clienti che acquistano la merce, dei buoni di consegna necessari per il ritiro della medesima. Tali buoni vengono presentati al terminal da degli autotrasportatori incaricati dal cliente per il ritiro fisico della merce. Un buono di consegna fa riferimento, per un certo peso, alla merce di una specifica polizza di carico e la merce di una polizza può essere suddivisa in parti non necessariamente uguali per clienti diversi che la possono ritirare in più volte.

Il fornitore offre per ogni polizza un certo numero di giorni di magazzinaggio in franchigia, nell'arco dei quali il esso è gratuito, dopo tale periodo i clienti debbono pagare un tariffa giornaliera prefissata per tonnellata di merce ancora immagazzinata.

Si vuole mantenere un registro cronologico dei ritiri merce che tenga conto dei clienti, del peso ritirato, dei camion utilizzati (targa) e del nominativi dei conducenti. Un camion può essere guidato nel tempo da diversi autisti e viceversa.

Prima parte

Si consideri la realtà di riferimento sopra descritta e, una volta formulate opportune ipotesi aggiuntive, si realizzino:

1. La progettazione concettuale della realtà indicata attraverso la produzione di uno schema (ad esempio ER, *Entity-Relationship*) con gli attributi di ogni entità, il tipo di ogni relazione e i suoi eventuali attributi;
2. una traduzione dello schema concettuale realizzato in uno schema logico (ad esempio secondo uno schema relazionale);
3. le seguenti interrogazioni espresse in algebra relazionale e/o in linguaggio SQL:
 - a) produrre l'elenco dei viaggi con data di allibramento compresa tra i mesi di marzo e aprile del 2015 indicando nave, viaggio, data partenza, data allibramento e porto di partenza in ordine di arrivo;
 - b) produrre il traffico in termini di peso totale per ogni linea e tipologia di merce nel corso del 2014 ;
 - c) visualizzare il nome delle navi che non hanno mai trasportato un certo tipo di merce;
 - d) il nome del camionista che nel tempo ha ritirato la minore quantità (peso) di merce.

Seconda parte

Scegliere e sviluppare almeno due dei seguenti punti:

1. Fornire una porzione significativa della definizione della struttura della base di dati individuata mediante il linguaggio DDL ovvero del DB-Schema.
2. Argomentare adeguatamente gli aspetti relativi alla tecnologia ed alle relative implicazioni che la dirigenza del terminal dovrebbe valutare circa la realizzazione di un sistema web-based per la gestione delle operazioni di ingresso uscita delle merci nel terminal e la fatturazione dei costi di magazzinaggio per i vari clienti. Si discuta inoltre la possibilità di permettere l'accesso al sito a varie tipologie di utenti (operatori interni, clienti, fornitori, autotrasportatori)
3. Sviluppare il problema che segue descrivendone le problematiche e le soluzioni tecniche adottabili: il terminal vuole realizzare una funzionalità sul proprio sito Web per permettere di verificare per uno specifico fornitore la quantità di merce, suddivisa per polizza, relativa una certa coppia nave-viaggio e ancora giacente nei magazzini del terminal.
4. Data la relazione:

voli(ora_volo, linea, partenza, arrivo, aereo, posti)

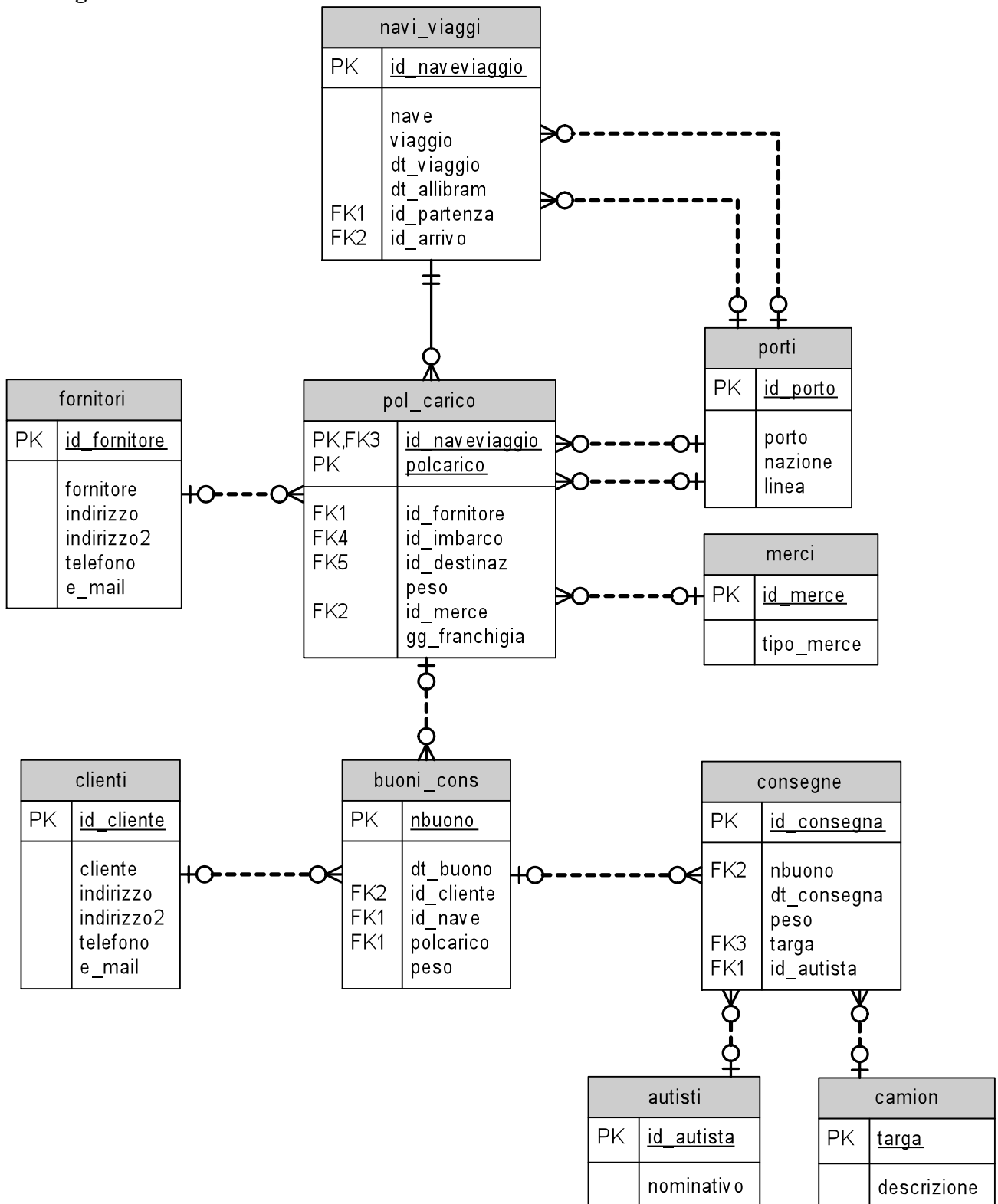
ora_volo	linea	partenza	arrivo	aereo	posti
7.00	1	Pisa	Milano	I-BIXN	80
7.00	2	Roma	Pisa	EI-CIKN	85
12.30	1	Pisa	Milano	I-ABXR	90
12.30	2	Roma	Pisa	I-BIXN	80
20.30	3	Roma	Milano	EI-DETS	90
20.30	1	Pisa	Milano	EI-CIKN	85

descrivere il processo di riduzione in terza forma normale

1.0 Ipotesi aggiuntive

- ⊙ Nella presente soluzione si è preferito usare un'unica tabella per censire le coppie nave-viaggio piuttosto che usare due tabelle distinte in quanto per ogni nave si è ritenuto sufficiente osservarne solo il nome senza informazioni aggiuntive.
- ⊙ Ad ogni coppia nave-viaggio è stato associato un codice univoco. In alternativa era possibile usare la combinazione dei due attributi nave viaggio come chiave primaria. In quest'ultimo caso anche nella tabella delle polizze di carico avremmo dovuto usare la stessa coppia di attributi come chiave esterna.
- ⊙ Si è ragionevolmente supposto che ogni polizza faccia riferimento ad un'unica tipologia di merce.
- ⊙ Per i dati dei camionisti e degli autoveicoli sono state previste due apposite tabelle collegando poi queste con la tabella delle consegne tramite le corrispondenti chiavi esterne. Dal momento che vengono memorizzati di volta in volta solo la targa del camion e del relativo autista si poteva optare per la loro memorizzazione diretta nella tabella delle consegne. Questa soluzione pur non essendo è strettamente aderente ad un corretto processo di normalizzazione potrebbe essere probabilmente migliore da un punto di vista pratico.
- ⊙ Dal momento che si parla di una tariffa giornaliera per l'immagazzinaggio per tonnellata di merce, sarebbe utile memorizzare da qualche parte tale parametro. Anche se nella soluzione proposta questo punto non è stato analizzato, avremmo potuto prevedere una tabella in cui memorizzare tale valore associato ad una coppia di date che ne determina la validità temporale assumendo che nel tempo i costi di immagazzinaggio possano subire delle variazioni.

1.1 Diagramma database



1.2 Tabelle database

Autisti

È la tabella in cui sono censite i vari autisti che almeno una volta hanno ritirato della merce al terminal:

Campo	R	Tipo	Dim.	Descrizione
id_autista	PK	intero		Codice identificativo dell'autista
nominativo		carattere	20	Cognome e nome dell'autista

Chiave Primaria: Id_autista

Camion

È la tabella in cui sono censiti i vari autoveicoli che almeno una volta hanno ritirato della merce al terminal:

Campo	R	Tipo	Dim.	Descrizione
targa	PK	carattere		Targa dell'automezzo
descrizione		carattere	20	Descrizione dell'automezzo (marca e tipo)

Chiave Primaria: targa

Merci

È la tabella in cui sono censite le tipologie di merci che arrivano al terminal:

Campo	R	Tipo	Dim.	Descrizione
id_merce	PK	intero		Codice identificativo della tipologia di merce
tipo_merce		carattere	20	Descrizione categoria di della tipologia di merce

Chiave Primaria: id_merce

Clienti

È la tabella in cui sono censite i dati dei clienti che acquistano la merce:

Campo	R	Tipo	Dim.	Descrizione
id_cliente	PK	intero		Codice identificativo del cliente
cliente		carattere	20	Ragione sociale del cliente
indirizzo		carattere	30	Indirizzo
indirizzo2		carattere	30	Campo aggiuntivo per l'indirizzo
telefono		carattere	15	Numero recapito telefonico
e-mail		carattere	30	e-mail di riferimento

Chiave Primaria: id_cliente

Fornitori

È la tabella in cui sono censite i dati dei fornitori della merce:

Campo	R	Tipo	Dim.	Descrizione
id_fornitore	PK	intero		Codice identificativo del fornitore
cliente		carattere	20	Ragione sociale del fornitore
indirizzo		carattere	30	Indirizzo
indirizzo2		carattere	30	Campo aggiuntivo per l'indirizzo
telefono		carattere	15	Numero recapito telefonico
e-mail		carattere	30	e-mail di riferimento

Chiave Primaria: id_fornitore

Porti

È la tabella in cui sono censite i dati dei porti di partenza/destinazioni delle navi/merci:

Campo	R	Tipo	Dim.	Descrizione
id_porto	PK	intero		Codice identificativo del porto
porto		carattere	20	Denominazione del porto
nazione		carattere	30	Nazione del porto
linea		carattere	30	Direttrice dei flussi di traffico (determinata quando porto di partenza)

Chiave Primaria: id_porto

Navi_viaggi

È la tabella in cui sono memorizzati i dati delle navi e dei relativi viaggi:

Campo	R	Tipo	Dim.	Descrizione
id_naveviaggio	PK	intero		Codice identificativo della coppia nave/viaggio
nave		carattere	25	Nome della nave
viaggio		carattere	4	Sigla viaggio
dt_viaggio		data/ora		Data inizio viaggio
dt_allibram		data/ora		Data di allibramento/magazzinaggio merce
id_partenza	FK	intero		Porto partenza
id_arrivo	FK	intero		Porto arrivo

Chiave Primaria: id_naveviaggio

Chiave Esterna: id_partenza → porti.id_porto

Chiave Esterna: id_arrivo → porti.id_porto

Polcar

È la tabella in cui sono memorizzati i dati delle polizze di carico:

Campo	R	Tipo	Dim.	Descrizione
id_naveviaggio	PK	intero		Codice identificativo della coppia nave/viaggio
polcarico	PK	carattere	11	Sigla polizza
id_fornitore	FK	intero		Identificatore fornitore
id_imbarco	FK	intero		Codice porto imbarco polizza
id_destinaz	FK	intero		Codice porto destinazione polizza
peso		intero		Peso della merce in kg
id_merce	FK	intero		Codice merce
gg_franchigia		intero		Numero giorni immagazzinaggio gratuito

Chiave Primaria: id_naveviaggio, polcarico

Chiave Esterna: id_fornitore → fornitori.id_fornitore

Chiave Esterna: id_merce → merci.id_merce

Chiave Esterna: id_naveviaggio → navi_viaggi.id_naveviaggio

Chiave Esterna.: id_imbarco → porti.id_porto

Chiave Esterna.: id_destinaz → porti.id_porto

Buoni_cons

È la tabella in cui sono memorizzati i dati dei buoni di ritiro merce:

Campo	R	Tipo	Dim.	Descrizione
nbuono	PK	carattere	9	Codice identificativo del buono di ritiro merce
dt_buono		data/ora		Data emissione buono consegna
id_cliente	FK	intero		Identificativo cliente
id_naveviaggio	FK	intero		Codice identificativo della coppia nave/viaggio
polcarico	FK	carattere	11	Sigla polizza
peso		intero		Peso di merce per il buono

Chiave Primaria: id_naveviaggio, polcarico

Chiave Esterna: id_naveviaggio, polcarico → polcar.id_naveviaggio, polcar.polcarico

Chiave Esterna: id_cliente → clienti.id_cliente

Consegne

È la tabella in cui sono memorizzati i dati relativi ai ritiri della merce:

Campo	R	Tipo	Dim.	Descrizione
id_consegna	PK	carattere	10	Codice identificativo della consegna merce
nbuono	FK	carattere	11	Riferimento buono ritiro merce
dt_consegna		data/ora		Data ritiro merce
peso		intero		Peso merce ritirata
id_autista	FK	Intero		Identificativo conducente automezzo che ritira la merce
targa	FK	intero		Identificativo automezzo usato per ritirare la merce

Chiave Primaria: id_consegna

Chiave Esterna: nbuono → buoni_cons.nbuono

Chiave Esterna: id_autista → autisti.id_autista

Chiave Esterna: targa → camion.targa

1.3 Query

a)

```
SELECT nave, viaggio, dt_viaggio, dt_allibram, porto AS porto_partenza
FROM navi_viaggi, porti
WHERE navi_viaggi.id_partenza=porti.id_porto
      AND YEAR(dt_allibram)=2015
      AND MONTH(dt_allibram)>=4
      AND MONTH(dt_allibram)<=6
ORDER BY dt_allibram;
```

b)

```
SELECT linea, tipo_merce, SUM(peso) AS peso_totale
FROM navi_viaggi, porti, pol_carico, merci
WHERE porti.id_porto = navi_viaggi.id_partenza
      AND navi_viaggi.id_naveviaggio = pol_carico.id_naveviaggio
      AND merci.id_merce = pol_carico.id_merce
GROUP BY linea, tipo_merce;
```

c)

```
SELECT DISTINCT navi_viaggi.nave
FROM navi_viaggi
WHERE id_naveviaggio NOT IN (
      SELECT id_naveviaggio
      FROM pol_carico, merci
      WHERE merci.id_merce = pol_carico.id_merce
            AND tipo_merce=[tipologia_merce]);
```

d)

```
SELECT id_autista, nominativo, PesoTotale
FROM (SELECT autisti.id_autista, nominativo, SUM(consegne.peso) AS PesoTotale
      FROM autisti, consegne
      WHERE autisti.id_autista = consegne.id_autista
      GROUP BY autisti.id_autista, nominativo) AS T1
WHERE PesoTotale=(
      SELECT MIN(PesoTotale)
      FROM (SELECT autisti.id_autista,nominativo,
                  SUM(consegne.peso) AS PesoTotale
            FROM autisti,consegne
            WHERE autisti.id_autista = consegne.id_autista
            GROUP BY autisti.id_autista,nominativo
            ) AS T2
      );
```


2.1 DB-Schema

```
CREATE TABLE autisti( id_autista INTEGER,
                      nominativo VARCHAR(40),
                      CONSTRAINT PrimaryKey PRIMARY KEY(id_autista)
                      );
CREATE TABLE camion( targa VARCHAR(7),
                     descrizione VARCHAR(40),
                     CONSTRAINT PrimaryKey PRIMARY KEY(targa)
                     );
CREATE TABLE clienti( id_cliente INTEGER,
                      cliente VARCHAR(25),
                      indirizzo VARCHAR(30),
                      indirizzo2 VARCHAR(30),
                      telefono VARCHAR(15),
                      e_mail VARCHAR(40),
                      CONSTRAINT PrimaryKey PRIMARY KEY(id_cliente)
                      );
CREATE TABLE fornitori( id_fornitore VARCHAR(4),
                        fornitore VARCHAR(20),
                        indirizzo VARCHAR(30),
                        indirizzo2 VARCHAR(30),
                        telefono VARCHAR(15),
                        e_mail VARCHAR(40),
                        CONSTRAINT PrimaryKey PRIMARY KEY(id_fornitore)
                        );
CREATE TABLE merci( id_merce INTEGER,
                     tipo_merce VARCHAR(50),
                     CONSTRAINT PrimaryKey PRIMARY KEY(id_merce)
                     );
CREATE TABLE porti( id_porto INTEGER,
                    porto VARCHAR(20),
                    nazione VARCHAR(20),
                    linea VARCHAR(20),
                    CONSTRAINT PrimaryKey PRIMARY KEY(id_porto)
                    );
CREATE TABLE navi_viaggi( id_naveviaggio INTEGER,
                          nave VARCHAR(25),
                          viaggio VARCHAR(4),
                          dt_viaggio DATETIME,
                          dt_allibram DATETIME,
                          id_partenza INTEGER,
                          id_arrivo INTEGER,
                          CONSTRAINT portinavi_viaggi1
                          FOREIGN KEY(id_partenza)
                          REFERENCES Porti(id_porto),
                          CONSTRAINT portinavi_viaggi2
                          FOREIGN KEY(id_arrivo)
                          REFERENCES porti(id_porto),
                          CONSTRAINT PrimaryKey
                          PRIMARY KEY(id_naveviaggio)
                          );
CREATE TABLE pol_carico( id_naveviaggio INTEGER,
                        polcarico VARCHAR(11),
                        id_fornitore VARCHAR(4),
                        id_imbarco INTEGER,
                        id_destinaz INTEGER,
                        peso DOUBLE,
                        id_merce INTEGER,
                        gg_franchigia DOUBLE,
```

```

CONSTRAINT caricatoripol
    FOREIGN KEY(id_fornitore)
    REFERENCES fornitori(id_fornitore),
CONSTRAINT mercipol_carico
    FOREIGN KEY(id_merce)
    REFERENCES merci(id_merce),
CONSTRAINT navi_viaggipol_carico
    FOREIGN KEY(id_naveviaggio)
    REFERENCES navi_viaggi(id_naveviaggio),
CONSTRAINT portipol_caricol
    FOREIGN KEY(id_imbarco)
    REFERENCES Porti(id_porto),
CONSTRAINT portipol_carico2
    FOREIGN KEY(id_destinaz)
    REFERENCES porti(id_porto),
CONSTRAINT PrimaryKey
    PRIMARY KEY(id_naveviaggio,polcarico)
);
CREATE TABLE buoni_cons( nbuono VARCHAR(9),
    dt_buono DATETIME,
    id_cliente INTEGER,
    id_naveviaggio INTEGER,
    polcarico VARCHAR(11),
    peso INTEGER,
    CONSTRAINT polcar_buoniconsegna
        FOREIGN KEY(id_naveviaggio,polcarico)
        REFERENCES pol_carico(id_naveviaggio,polcarico),
    CONSTRAINT RicevitoriSchede
        FOREIGN KEY(id_cliente)
        REFERENCES clienti(id_cliente)
    CONSTRAINT PrimaryKey PRIMARY KEY(nbuono),
);
CREATE TABLE consegne( id_consegna VARCHAR(10),
    nbuono VARCHAR(9),
    dt_consegna DATETIME,
    peso DOUBLE,
    targa VARCHAR(7),
    id_autista INTEGER,
    CONSTRAINT AutistiConsegne
        FOREIGN KEY(id_autista)
        REFERENCES autisti(id_autista),
    CONSTRAINT BuoniConsegnaConsegne
        FOREIGN KEY(nbuono)
        REFERENCES buoni_cons(nbuono),
    CONSTRAINT CamionConsegne
        FOREIGN KEY(targa)
        REFERENCES camion(targa),
    CONSTRAINT PrimaryKey PRIMARY KEY(id_consegna)
);

```

2.2 Sito web

Date le caratteristiche del terminal che probabilmente non dispone di personale specializzato per la gestione di un server atto ad 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.term2016.com): 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 (ipotizzando che il sw venga sviluppato con questo tipo di linguaggio) nel server web e la disponibilità di un DBMS (es. My-SQL) per la gestione del database. Gli strumenti indicati costituiscono un pacchetto molto comune nell'offerta dei fornitori di servizi web.

Un aspetto di primaria importanza 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 dinamici che si configurano dinamicamente a seconda del profilo utente attivato (amministratore di sistema, personale operativo di magazzino, personale sottobordo, personale al gate di ingresso/uscita stradale, clienti, fornitori, ecc.)

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

2.3 Pagine web interrogazione giacenza di magazzino

Come esempio di pagine web dinamiche presenti sul sito del terminal presentiamo 2 script PHP (il primo per inserire la richiesta ed il secondo per visualizzare la relativa risposta) che permettono al personale del terminal di verificare l'attuale giacenza di magazzino delle polizze di carico di un fornitore per una specifica nave-viaggio. Nel caso specifico, si è ipotizzato che il server My-SQL sia in esecuzione sullo stesso computer del web server e che l'accesso al database "terminal" venga effettuato come utente "root" privo di password; inoltre le pagine web sono completamente prive di struttura grafica.

ini.php

```
<html>
<head>
  <title>Selezione nave-viaggio e fornitore</title>
</head>
<body>
  <?php
    $connection = mysqli_connect("localhost", "root", "", "terminal");
    if (mysqli_connect_errno($connection))
    {
      echo "Errore di connessione al DBMS My-SQL." ;
      die();
    }
  ?>
  <form action="risposta.php" method="GET">
  Selezionare nave-viaggio:
  <select name="naveviaggio">
    <?php
      $query = "SELECT id_naveviaggio,nave,viaggio
                FROM navi_viaggi
                ORDER BY nave;";
      $result = mysqli_query($connection, $query);
      if (!$result)
      {
        echo "Errore esecuzione query SQL." ;
        die();
      }
      if (mysqli_num_rows($result) == 0)
      {
        echo "Nessuna nave-viaggio selezionata." ;
        die();
      }

      while ($row = mysqli_fetch_array($result))
      {
        echo "<option value=\"\$row[0]\">$row[1]-$row[2]</option>";
      }
      mysqli_free_result($result);
    ?>
  </select>
  <br>
  <br>
  <br>
  Selezionare fornitore:
```

```

<select name="fornitore">
  <?php

    $query = "SELECT id_fornitore, fornitore
              FROM fornitori
              ORDER BY fornitore;";
    $result = mysqli_query($connection, $query);
    if (!$result)
    {
        echo "Errore esecuzione query SQL.";
        die();
    }
    if (mysqli_num_rows($result) == 0)
    {
        echo "Nessun fornitore rilevato." ;
        die();
    }

    while ($row = mysqli_fetch_array($result))
    {
        echo "<option value=\"\$row[0]\">$row[1]</option>";
    }
    mysqli_free_result($result);
    mysqli_close($connection);
  ?>
</select>
<br>
<input type="submit" value="Cerca polizze">
<br>
</form>
</body>
</html>

```

risposta.php

```

<html>
<head>
  <title>Giacenza di magazzino per polizza</title>
</head>
<body>
  <?php
    $connection = mysqli_connect("localhost", "root", "", "terminal");
    if (mysqli_connect_errno($connection))
    {
        echo "Errore di connessione al DBMS My-SQL." ;
        die();
    }

    $query = "SELECT nave, viaggio, polcarico, peso AS pesoiniziale,
              pesoconsegnato, peso-pesoconsegnato AS rimanenza
              FROM( SELECT navi_viaggi.nave, navi_viaggi.viaggio,
                          pol_carico.polcarico, pol_carico.peso,
                          INT(SUM(consegne.peso)) AS pesoconsegnato
                    FROM navi_viaggi,pol_carico,buoni_cons,consegne
                    WHERE navi_viaggi.id_naveviaggio =
                          pol_carico.id_naveviaggio
                    AND pol_carico.polcarico = buoni_cons.polcarico
                    AND pol_carico.id_naveviaggio = buoni_cons.id_naveviaggio
                    AND buoni_cons.nbuono = consegne.nbuono
                    AND id_fornitore=$_GET[fornitore]

```

```

        AND navi_viaggi.id_naveviaggio=$_GET[naveviaggio]
        GROUP BY navi_viaggi.nave, navi_viaggi.viaggio,
                pol_carico.polcarico, pol_carico.peso
    ) AS T;";

$result = mysqli_query($connection, $query);
if (!$result)
{
    echo "Errore esecuzione query SQL." ;
    die();
}
if (mysqli_num_rows($result) == 0)
{
    echo "Nessuna polizza trovata." ;
    die();
}
?>
<table border>
<caption><b>Stato polizze</b></caption>
<thead>
<tr>
<th>Nave</th>
<th>Viaggio</th>
<th>Polizza Carico</th>
<th>Peso iniziale</th>
<th>Peso consegnato</th>
<th>Rimanenza</th>
</tr>
</thead>
<tbody>
<?php
    while ($row = mysqli_fetch_assoc($result))
    {
        ?>
        <tr>
            <td><?php echo ($row['nave']); ?></td>
            <td><?php echo ($row['viaggio']); ?></td>
            <td><?php echo ($row['polcarico']); ?></td>
            <td><?php echo ($row['pesoiniziale']); ?></td>
            <td><?php echo ($row['pesoconsegnato']); ?></td>
            <td><?php echo ($row['rimanenza']); ?></td>
        </tr>
    <?php
    }
    mysqli_free_result($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:

Selezionare nave-viaggio: STAR KILIMANJARO-613

Selezionare fornitore: INTERNATIONAL PAPER

Cerca polizze

- CELLULOSA INC.
- CELLULOSE NIPO BRAS
- CENTRAL NATIONAL
- CHETWIND
- CMPC CELLULOSA
- DOMTAR INDUSTRIEST
- ELDORADO
- FIBRE SOURCE INTERN
- FIBRIA
- GEORGIA PACIFIC
- GRIEG STAR
- INTER PAPIER
- INTERNATIONAL FOREST
- INTERNATIONAL PAPER**
- RESOLUTE FP US INC
- ROCKTENN CP
- SILK CELLULOSA INC.
- SUZANO PAPEL
- TRADERS SRL
- UPM S.A. URUGUAY

Giacenza di magazzino p X

localhost/terminal/risposta.php?naveviaggio=1006&fornit ☆

Stato polizze

Nave	Viaggio	Polizza Carico	Peso iniziale	Peso consegnato	Rimanenza
STAR KILIMANJARO	613	13ATL1401A	120484	120483	1
STAR KILIMANJARO	613	13ATL1401B	238382	210249	28133
STAR KILIMANJARO	613	13ATL1401C	146865	146864	1
STAR KILIMANJARO	613	13ATL1401D	75653	75652	1
STAR KILIMANJARO	613	13ATL1401E	1115350	1115349	1
STAR KILIMANJARO	613	13ATL1401F	150911	150910	1
STAR KILIMANJARO	613	13ATL1401G	162714	162714	0
STAR KILIMANJARO	613	13ATL1407A	1452354	1254305	198049
STAR KILIMANJARO	613	13ATL1407B	285940	225101	60839
STAR KILIMANJARO	613	13ATL1407C	175089	57356	117733
STAR KILIMANJARO	613	13ATL1414A	1828950	1828950	0
STAR KILIMANJARO	613	13ATL1414B	152413	152413	0
STAR KILIMANJARO	613	13ATL1414C	457238	457237	1
STAR KILIMANJARO	613	13ATL1414D	152413	152413	0
STAR KILIMANJARO	613	13ATL1414E	365791	365791	0
STAR KILIMANJARO	613	13ATL1414F	304826	304826	0
STAR KILIMANJARO	613	13ATL1414G	426758	426757	1
STAR KILIMANJARO	613	13ATL1414H	121931	121931	0
STAR KILIMANJARO	613	13ATL1415A	115868	115867	1
STAR KILIMANJARO	613	13ATL1415B	14553	14553	0
STAR KILIMANJARO	613	13ATL1415C	145422	145421	1
STAR KILIMANJARO	613	13ATL1415D	836499	836498	1
STAR KILIMANJARO	613	13ATL1415E	147405	147405	0
STAR KILIMANJARO	613	13ATL1415F	30792	30792	0
STAR KILIMANJARO	613	13ATL1415G	1139980	1034307	105673

Soluzione alternativa utilizzando la tecnologia PDO

connection.php

```
<?php
$host = 'localhost';
$user = 'root';
$password = '';
$database = 'terminal';
$dsn = "mysql:dbname=$database;host=$host";

try {
    $pdo = new PDO($dsn, $user, $password);
    $pdo->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
} catch (PDOException $e) {
    printf('Errore di connessione al DBMS My-SQL. (%s)', $e->getMessage());
    exit();
}
```

ini.php

```
<?php
require __DIR__.'/'connection.php';
?>
<html xmlns="http://www.w3.org/1999/html">
  <head>
    <title>Selezione nave-viaggio e fornitore</title>
  </head>
  <body>
    <form action="risposta.php" method="GET">
      Selezionare nave-viaggio:
      <select name="naveviaggio">
        <?php
          foreach ($pdo->query('SELECT id_naveviaggio, nave, viaggio
                                FROM navi_viaggi ORDER BY nave') as $row) {
            printf('<option value="%s">%s - %s</option>',
                  $row['id_naveviaggio'], $row['nave'], $row['viaggio']);
          }
        ?>
      </select>
      <br>
      <br>
      Selezionare fornitore:
      <select name="fornitore">
        <?php
          foreach ($pdo->query('SELECT id_fornitore, fornitore
                                FROM fornitori ORDER BY fornitore') as $row) {
            printf('<option value="%s">%s</option>', $row['id_fornitore'],
                  $row['fornitore']);
          }
        ?>
      </select>
      <br>
      <br>
      <input type="submit" value="Cerca polizze">
      <br>
    </form>
  </body>
</html>
```

risposta.php

```
<?php
require __DIR__.' /connection.php';

if (empty($_GET['fornitore']) || empty($_GET['naveviaggio'])) {
    header('Location: ./ini.php');
    exit();
}

$sql = "SELECT
        nave,
        viaggio,
        polcarico,
        peso                AS pesoiniziale,
        pesoconsegnato,
        peso - pesoconsegnato AS rimanenza
FROM
        (
            SELECT
                navi_viaggi.nave,
                navi_viaggi.viaggio,
                pol_carico.polcarico,
                pol_carico.peso,
                FLOOR(SUM(consegne.peso)) AS pesoconsegnato
            FROM navi_viaggi, pol_carico, buoni_cons, consegne
            WHERE navi_viaggi.id_naveviaggio =
                pol_carico.id_naveviaggio
                AND pol_carico.polcarico = buoni_cons.polcarico
                AND pol_carico.id_naveviaggio =
                    buoni_cons.id_naveviaggio
                AND buoni_cons.nbuono = consegne.nbuono
                AND id_fornitore = :fornitore
                AND navi_viaggi.id_naveviaggio = :naveviaggio
            GROUP BY navi_viaggi.nave, navi_viaggi.viaggio,
                pol_carico.polcarico, pol_carico.peso
        ) AS T";

?>
<html>
<head>
<title>Giacenza di magazzino per polizza</title>
</head>
<body>
<?php
    $statement = $pdo->prepare($sql);
    $statement->execute(
        array (
            ':fornitore' => $_GET['fornitore'],
            ':naveviaggio' => $_GET['naveviaggio'],
        )
    );
?>
<table border>
<caption><b>Stato polizze</b></caption>
<thead>
<tr>
<th>Nave</th>
<th>Viaggio</th>
<th>Polizza Carico</th>
<th>Peso iniziale</th>
<th>Peso consegnato</th>

```

```

        <th>Rimanenza</th>
    </tr>
</thead>
<tbody>
    <?php foreach ($statement->fetchAll() as $row) { ?>
        <tr>
            <td><?php echo($row['nave']); ?></td>
            <td><?php echo($row['viaggio']); ?></td>
            <td><?php echo($row['polcarico']); ?></td>
            <td><?php echo($row['pesoiniziale']); ?></td>
            <td><?php echo($row['pesoconsegnato']); ?></td>
            <td><?php echo($row['rimanenza']); ?></td>
        </tr>
    <?php } ?>
</tbody>
</table>
</body>
</html>

```

Anche in questo caso gli output prodotti sono gli stessi visti in precedenza.

2.4 Normalizzazione tabella voli

Consideriamo la relazione:

voli(ora_volo, linea, partenza, arrivo, aereo, posti)

ora_volo	tratta	partenza	arrivo	aereo	posti
7.00	1	Pisa	Milano	I-BIXN	80
7.00	2	Roma	Pisa	EI-CIKN	85
12.30	1	Pisa	Milano	I-ABXR	90
12.30	2	Roma	Pisa	I-BIXN	80
20.30	3	Roma	Milano	EI-DETS	90
20.30	1	Pisa	Milano	EI-CIKN	85

Prima forma normale. Una relazione è detta in 1NF quando tutti i domini sono atomici ed è stata definita la chiave primaria.

Nel nostro caso i domini non sono ulteriormente scomponibili e la chiave primaria potrebbe essere costituita dalla coppia ora_volo, tratta.

Si possono quindi osservare le seguenti dipendenze funzionali:

- 1) tratta → partenza (ogni tratta ha uniche partenza e arrivo)
- 2) tratta → arrivo
- 3) ora_volo, tratta → aereo: ad ogni volo è associato un aeromobile
- 4) aereo → posti: ogni aeromobile ha una sua capienza massima.

Procediamo ora con la normalizzazione.

Seconda forma normale. Una relazione è detta in 2FN quando non esistono dipendenze funzionali parziali ma si hanno solo dipendenze funzionali complete.

La relazione voli non è in 2FN perché partenza e arrivo dipendono solo da tratta, quindi è necessario scomporre la relazione voli in:

voli2(ora_volo, tratta, aereo, posti) e **tratte**(tratta, partenza, arrivo)

Le relazioni in seconda forma normale possono ancora presentare delle ridondanze e di conseguenza delle anomalie. Questo accade quando esistono attributi non chiave che dipendono da altri attributi non chiave.

Nel nostro caso abbiamo che:

- aereo → posti

ed anche che:

- (ora_volo, tratta) → aereo → posti.

Quindi posti dipende dalla chiave in modo transitivo tramite aereo.

Terza forma normale. Diremo che una relazione R è in 3FN quando per ogni possibile chiave si ha:

- 1) R è in 2FN cioè non si hanno dipendenze funzionali parziali
- 2) non esistono attributi dipendenti transitivamente dalla chiave

Quindi si ha la necessità di una ulteriore scomposizione della relazione voli2 fino ad ottenere:

voli3(ora_volo, tratta, aereo)

aerei(aereo, posti)

tratte(tratta, partenza, arrivo)