

## **M070 - ESAME DI STATO DI ISTITUTO TECNICO INDUSTRIALE**

CORSO DI ORDINAMENTO

**Indirizzo:** INFORMATICA

**Tema di:** INFORMATICA GENERALE E APPLICAZIONI TECNICO SCIENTIFICHE

Un negozio on line chiede che sia progettato e realizzato un database per l'organizzazione e la gestione di un portale per la vendita di libri su Internet.

Il negozio richiede che:

- il Catalogo dei libri sia organizzato per:
  - Reparti (in ordine alfabetico: *architettura e urbanistica, arte, classici greci e latini, cucina e casa, diritto, economia e management, fantascienza e fantasy, filosofia, fumetti, ...*)
- ciascun Reparto sia organizzato in libri:
  - "Da non perdere" caratterizzati da uno sconto del 20% sul prezzo di copertina
  - "I più venduti" caratterizzati dal maggior numero di copie vendute negli ultimi 30 giorni
  - "Ultimi arrivi" caratterizzati dai titoli aggiunti negli ultimi 30 giorni
  - "Offerte speciali" caratterizzate da uno sconto del 25% sul prezzo di copertina
  - "Remainders" caratterizzati da uno sconto del 50% sul prezzo di copertina
- ciascun libro in negozio sia archiviato con le seguenti ulteriori informazioni:

titolo, autore, prezzo di copertina, sconto (eventuale), dati (numero di pagine, rilegato, ...), anno di pubblicazione, editore, collana, immagine (foto della copertina), data di archiviazione
- gli utenti abilitati all'acquisto on line siano registrati con i seguenti dati:
  - nome e cognome, indirizzo, codice di avviamento postale, città, nazione, telefono, fax, e-mail, password, numero di carta di credito, tipo di carta di credito (Visa, CartaSi, Mastercard, ...), data di scadenza della carta di credito
- gli utenti possano chiedere, in fase di acquisto, che il negozio emetta fattura
- il Carrello Acquisti sia organizzato come segue:
  - numero di copie acquistate, titolo, autore, prezzo, disponibilità (giorni, settimane, ...), togli dal carrello (casella di spunta)
- il riepilogo degli ordini sia organizzato come segue:
  - data dell'ordine, numero d'ordine
  - dettagli dell'ordine (riferimento articolo, quantità, titolo, prezzo di copertina, prezzo scontato, stato dell'ordine, evasione prevista dalla data dell'ordine).

Il candidato, dopo aver fatto le eventuali ipotesi aggiuntive:

a. Fornisca:

1. lo schema concettuale e lo schema logico del database
2. la definizione delle relazioni in linguaggio SQL.

b. Implementi le seguenti query:

1. Ricerca per Reparto: scelto un Reparto, il numero di libri "Da non perdere" ed il numero di libri "I più venduti" con i relativi dettagli
2. Ordini: gli ordini in corso, con dettagli, di un dato utente.

c. Scriva in un linguaggio lato server, il codice di almeno una delle seguenti pagine del portale:

1. con accesso libero, la pagina utile a visualizzare i Reparti e, per ciascun reparto, la pagina che implementa la query n. 1;
2. con accesso riservato agli utenti registrati, la composizione degli ordini in corso.

Durata massima della prova: 6 ore.

È consentito l'uso di calcolatrici non programmabili.

Non è consentito lasciare l'Istituto prima che siano trascorse 3 ore dalla dettatura del tema.

# Proposta di soluzione

## 1 Analisi della realtà di interesse

### Quesito a1)

**Premessa:** Si tratta di un classico problema di progettazione ed implementazione di un database che richiede da parte del tecnico un'analisi puntuale dei bisogni in modo da ottimizzare non solo i tempi di accesso, ma anche l'occupazione di spazio sul supporto di memorizzazione.

In particolare risulta fondamentale analizzare lo schema logico del database in modo da ridurre al minimo la duplicazione dei dati, mantenendo però una struttura che, in caso di crash, consenta di recuperare, se non tutti, almeno buona parte dei dati.

Trattandosi di un archivio piuttosto complesso in cui gli elementi fondamentali, libri e clienti, risultano essere presenti in numero piuttosto consistente appare indispensabile prevedere per il sistema un sistema di backup automatico che consenta di ridurre al minimo le possibilità di perdite di dati (ad esempio prevedendo l'esistenza di un secondo server di supporto, su cui periodicamente viene aggiornata una copia del database con le variazioni apportate nell'ultimo periodo).

La traccia richiede progettare una base di dati per la gestione di una libreria on-line. La base di dati è amministrata dal **gestore della libreria** ed è acceduta via web da **clienti** e **visitatori**.

**Clienti** sono coloro che hanno eseguito la registrazione memorizzando i propri dati nella base di dati

**Visitatori** sono coloro che accedono al sito web della libreria senza essere registrati.

L'**acquisto di libri** è riservato ai soli clienti e avviene come segue:

- 1) Si *inseriscono* i libri da acquistare in un **carrello virtuale**, specificando per ogni libro il numero di copie desiderate.
- 2) A partire dal carrello virtuale si *crea* quindi un **carrello reale** (da qui in poi chiamato semplicemente *carrello*) memorizzandolo nella base di dati. La creazione del carrello deve avvenire entro la stessa sessione di visita del sito web, e richiede l'immissione di username e password, assegnati durante la registrazione. Se non memorizzato un carrello virtuale scompare alla fine della sessione di visita del sito.
- 3) L'acquisto dei libri avviene emettendo un ordine per il carrello creato e ciò può essere fatto anche in una data successiva a quella di creazione del carrello). Per emettere un ordine si deve specificare una carta di credito e autorizzare l'acquisto dei libri presenti nel carrello che si vuole ordinare. Il costo addebitato al cliente è dato dalla somma dei prezzi dei libri più le spese di spedizione, che sono calcolate automaticamente dal sistema durante l'emissione dell'ordine.
- 4) Infine, l'ordine viene *evaso* dal gestore della libreria e ciò può avvenire in una data successiva a quella di emissione dell'ordine (ad esempio, se i libri ordinati non sono disponibili nel numero di copie desiderate e bisogna richiederli alla casa editrice).

Un visitatore può consultare il catalogo della libreria, effettuare la ricerca di un libro (secondo campi significativi), mettere dei libri in un carrello virtuale, ma non potrà procedere effettivamente all'acquisto, secondo la procedura sopra descritta, poiché deve eseguire prima la *registrazione*.

Un cliente, dunque, avendo già provveduto alla fase di registrazione, può effettuare tutto quanto previsto e descritto per un "semplice" visitatore con l'aggiunta fondamentale della memorizzazione nella base dati di un carrello e l'emissione di un ordine con relativa richiesta di fattura

Il gestore della base dati si occupa dell'inserimento-aggiornamento-cancellazione dei dati relativi a libri, autori, case editrici, clienti e dell'evasione di un ordine emesso da un cliente.

## **Descrizione dei dati**

L'organizzazione della base di dati della libreria on-line, in accordo a quanto indicato nella traccia, ruota attorno alle seguenti entità concettuali:

**Reparto** Il catalogo dei libri è organizzato per reparti tematici ordinati in ordine alfabetico come architettura ed urbanistica, arte, classici greci e latini, cucina e casa, diritto, economia, etc... Ciascun reparto è individuato da un codice identificativo univoco e da una descrizione obbligatoria (che ovviamente non può prevedere duplicati).

**SottoReparto** secondo la traccia, ciascun reparto può essere organizzato nei seguenti cinque settori:

- a) "Da non perdere": caratterizzati da uno sconto del 20% sul prezzo di copertina;
- b) "I più venduti": caratterizzati dal maggior numero di copie vendute nell'ultimo mese;
- c) "Ultimi arrivi": caratterizzati dai titoli aggiunti nell'ultimo mese;
- d) "Offerte speciali": caratterizzati da uno sconto del 25% sul prezzo di copertina;
- e) "Remainders": caratterizzati da uno sconto del 50% sul prezzo di copertina;

anch'essi caratterizzati da un un codice identificativo univoco e da una descrizione obbligatoria (che ovviamente non può prevedere duplicati) il cui valore è una delle cinque stringhe di caratteri sopra citate e dal campo sconto applicato che vale 0% (nei casi "I più venduti" ed "Ultimi Arrivi"), 20% (nel caso "Da non perdere"), 25% (nel caso "Offerte Speciali") oppure 50% (nel caso "Remainders").

**Libro** Il catalogo della libreria è costituito da diversi libri ciascuno dei quali identificato da un codice univoco (ad esempio il codice ISBN), dal titolo, dall'autore/i, dal prezzo di copertina, dallo sconto (eventuale), dall'anno di pubblicazione, dalla casa editrice, dalla collana, dall'immagine di copertina, dalla data di archiviazione, dal numero di pagine, dal tipo di rilegatura. Ovviamente la libreria dispone di 0 o più copie (0 significa che il libro non è al momento disponibile ma che lo sarà più avanti).

**Autore** Ogni (co)autore di un libro presente nel catalogo della libreria è memorizzato nella base di dati con i seguenti dati: un codice univoco, nome, cognome, data di nascita, scheda descrittiva.

**CasaEditrice** Ogni casa editrice di un libro presente nel catalogo della libreria è memorizzata nella base di dati con i seguenti dati: un codice univoco (ad esempio il numero di partita IVA), la denominazione, l'indirizzo, cap, città, nazione, il telefono, il fax, la e-mail, l'indirizzo del sito web.

**Cliente** Di un cliente vengono memorizzati, in fase di registrazione, i seguenti dati: un codice-cliente univoco, il nome, il cognome, l'indirizzo, il cap, la città, la nazione, il telefono, il fax, l'e-mail, lo username, la password, il numero di carta di credito, il tipo carta di credito, la data scadenza della carta di credito.

**CarrelloAcquisti** E' ottenuto dalla memorizzazione sulla base dati da parte di un cliente di un carrello virtuale ed è costituito da uno o più libri per ciascuno dei quali è specificato un numero (maggiore o uguale a uno) di copie.

Di un carrello vengono memorizzati i seguenti dati: un codice-carrello univoco, la data di registrazione, importo totale da pagare più una o più righe di dettaglio relative a ciascun libro presente nel carrello.

**RigaCarrelloAcquisti** E' ottenuta dalla memorizzazione sulla base dati da parte di un cliente di un carrello virtuale ed è relativa ad uno o più libri per ciascuno dei quali è specificato un numero (maggiore o uguale a uno) di copie.

Nella singola riga di un carrello vengono memorizzati i seguenti dati: numero di copie acquistate, titolo, autore, casa editrice, prezzo effettivo, disponibilità (giorni, settimane..etc.).

**Ordine** E' emesso da un cliente per l'acquisto di un carrello. Di un ordine vengono memorizzati i seguenti dati: un codice-ordine univoco (ossia il numero d'ordine), la data di emissione, lo stato (i cui valori possono essere "In corso", "Evaso", "Annullato"), data di evasione prevista, importo totale da pagare più una o più righe di dettaglio dell'ordine relative a ciascun libro presente nel carrello.

**RigaOrdine** E' relativa ad un certo ordine per l'acquisto di un carrello richiesto da un cliente. E' caratterizzata dai seguenti dati: un codice-riga ordine univoco, il numero di copie totali, il titolo, l'autore, la casa editrice, il prezzo di copertina, il prezzo scontato, il prezzo totale dovuto

**Fattura** viene richiesta opzionalmente da un cliente all'atto dell'emissione di un ordine e viene rilasciata dal gestore della libreria on-line. Dell'oggetto "fattura" (che intendiamo in una sua versione semplificata) vengono memorizzati i seguenti dati: un codice-fattura univoco (ossia il numero di fattura), la data di emissione, l'ordine di riferimento e l'importo totale da corrispondere (che coincide ovviamente con l'importo totale dell'ordine)

### Descrizione degli utenti

Gli utenti della base di dati sono: il gestore della libreria, che amministra la base di dati; e clienti e visitatori, che accedono alla base di dati via web.

### Descrizione dei casi d'uso

I (principali) casi d'uso sono differenziati in base al tipo di utente.

**Gestore della libreria:** lettura/inserimento/modifica/cancellazione dei dati relativi a libri, autori, case editrici e clienti; evasione di un ordine.

**Cliente:** lettura/modifica dei propri dati di registrazione; consultazione del catalogo ricercando un libro tramite una qualsiasi combinazione di (autore/i, titolo, casa editrice, anno di pubblicazione, codice univoco); inserimento di un libro in un carrello virtuale; creazione di un carrello a partire da un carrello virtuale; modifica/cancellazione di un carrello; emissione di un ordine per un carrello creato in precedenza.

**Visitatore:** registrazione; consultazione del catalogo ricercando un libro tramite una qualsiasi combinazione di (autore/i, titolo, casa editrice, anno di pubblicazione, codice univoco); inserimento di un libro in un carrello virtuale.

### Glossario dei Termini

**Carrello** (reale) Un insieme di libri, ciascuno in una o più copie, creato da un cliente memorizzando nella base di dati un carrello virtuale.

**Carrello virtuale** Un insieme di libri, ciascuno in una o più copie, specificato da un visitatore/cliente durante la visita del sito della libreria.

**Cliente** Persona che ha eseguito la registrazione e i cui dati sono memorizzati nella base di dati.

**Emissione di un ordine** L'autorizzazione data da un cliente per l'acquisto di un carrello da lui creato.

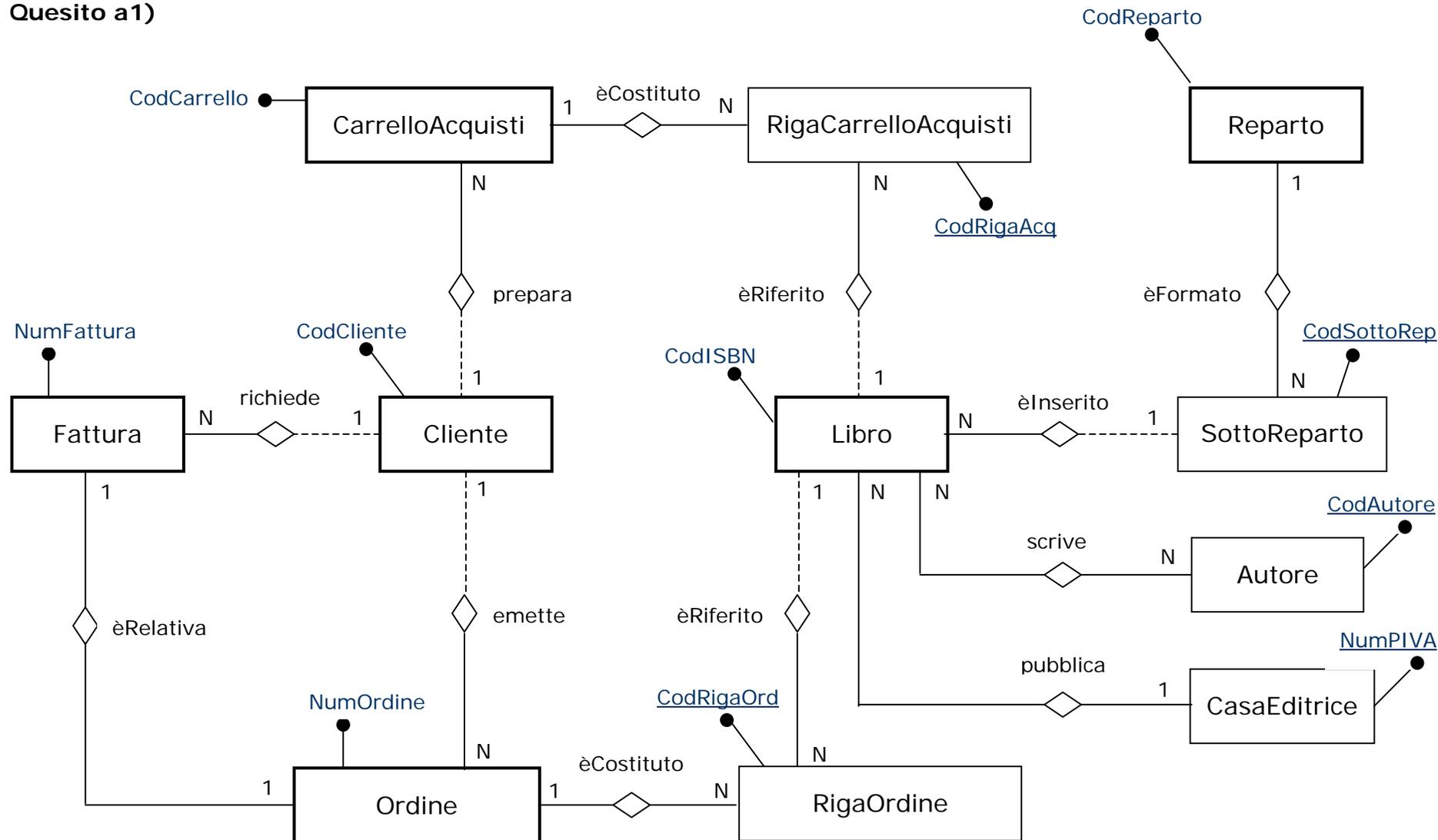
**Evasione di un ordine** Coincide con l'effettiva spedizione da parte del gestore della libreria dei libri richiesti da un cliente con l'emissione di un ordine.

**Registrazione** L'immissione dei propri dati da parte di un visitatore che diventa in questo modo un cliente.

**Visitatore** Persona che accede al sito web della libreria senza essere registrato e di cui non c'è traccia nella base di dati.

## 2 Diagramma ER (con indicazione per ragioni di spazio solo delle chiavi primarie e non di tutti gli attributi)

Quesito a1)



### 3 Schema logico relazionale

#### Quesito a1)

**Reparto** (CodRep, Descrizione)

**SottoReparto** (CodSottoRep, Descrizione, Sconto, CodRep1)

*Foreign Key: CodRep1 sulla relazione Reparto*

**Cliente** (CodCliente, Nome, Cognome, Indirizzo, Cap, Città, Nazione, Telefono, Fax, Email, Username, Password, NumCartaCredito, TipoCartaCredito, DataScadenza)

**Ordine** (NumOrdine, DataEmissione, Stato, DataEvasione, ImpTotaleDovuto, ImpTotaleDovuto, CodCliente1, NumFattura1)

*Foreign Key: CodCiente1 sulla relazione Cliente*

*Foreign Key: NumFattura1 sulla relazione Fattura*

**Fattura** (NumFattura, DataEmissione, ImpTotale, NumOrdine1)

*Foreign Key: NumOrdine1 sulla relazione Ordine*

**RigaOrdine** (CodRigaOrd, NumCopie, ImpTotaleDovuto, ImpTotaleScontato, CodLibro1)

*Foreign Key: CodLibro1 sulla relazione Libro*

**CarrelloAcquisti** (CodCarrello, DataRegistrazione, ImpTotaleDovuto, ImpTotaleDovuto, CodCliente2)

*Foreign Key: CodCiente2 sulla relazione Cliente*

**RigaCarrelloAcquisti** (CodRigaCarr, NumCopie, ImpTotaleDovuto, ImpTotaleScontato, Disponibilità, CodLibro2)

*Foreign Key: CodLibro2 sulla relazione Libro*

**CasaEditrice** (NumPIVA, Denominazione, Indirizzo, Cap, Città, Nazione, Telefono, Fax, Email, Web)

**Autore** (CodAutore, Nome, Cognome, DataNascita, SchedaDescr)

**Libro** (CodISBN, Titolo, PrezzoCop, AnnoPubbl, Collana, PathFileImg, FileImg, DataArchiviazione, NumPagine, TipoRilegatura, NumCopie, CodCasaEd1, CodSottoRep1, CodAutore1)

*Foreign Key: CodCasaEd1 sulla relazione CasaEditrice*

*Foreign Key: CodSottoRep1 sulla relazione SottoReparto*

*Foreign Key: CodAutore1 sulla relazione Scrive*

**Scrive** (CodAutore2, CodLibro3)

*Foreign Key: CodAutore2 sulla relazione Autore*

*Foreign Key: CodLibro3 sulla relazione Libro*

## 4 Definizione delle relazioni in linguaggio SQL

### Quesito a2)

Definiamo ora nei suoi aspetti principali, a titolo esemplificativo, una delle relazioni contenute nel database, dato che risulterebbe semplicemente nozionistica la definizione di tutte le tabelle. Abbiamo scelto una tabella che contiene anche chiavi esterne di correlazione in modo da evidenziarne le caratteristiche.

```
CREATE DATABASE libreria-on-line
```

.....

```
CREATE TABLE libro
```

```
(  
    CodISBN                CHAR(10) NOT NULL,  
    Titolo                 CHAR(10) NOT NULL,  
    PrezzoCop              FLOAT NOT NULL,  
    AnnoPub                INT NOT NULL,  
    Collana                FLOAT NOT NULL,  
    PathFileimg            CHAR(255) NOT NULL,  
    FileImg                CHAR(30) NOT NULL,  
    DataArchiviazione      DATE NOT NULL,  
    NumPagine              INT NOT NULL,  
    NumCopie                INT NOT NULL,  
    TipoRilegatura         CHAR (20),  
  
    CodCasaEd1             INT NOT NULL,  
    CodSottoRep1           INT NOT NULL,  
    CodAut1                INT NOT NULL,  
  
    PRIMARY KEY (CodISBN),  
    FOREIGN KEY (CodCasaEd1) REFERENCES CasaEditrice (NumPIVA),  
    FOREIGN KEY (CodSottoRep1) REFERENCES SottoReparto (CodSottoRep),  
    FOREIGN KEY (CodAut1) REFERENCES Autore(CodAutore),  
    CHECK (PrezzoCop > 0),  
    CHECK (AnnoPub > 0),  
    CHECK (NumPagine > 0),  
    CHECK (NumPagine >= 0),  
  
    ON DELETE RESTRICT,  
    ON UPDATE CASCADE  
);
```

.....

## 5 Definizione delle query (SQL ed algebra relazionale)

### Quesito b1)

**Q1:** Ricerca per Reparto: scelto un Reparto, il numero di libri "Da non perdere" ed il numero di libri "I più venduti" con i relativi dettagli

La query richiesta è un'interrogazione parametrica, nella quale i parametri dipendono dall'interpretazione che si può dare al quesito.

Supponiamo in prima ipotesi che l'utente fornisca la descrizione del reparto scelto (una tra quelle ammesse tipo *architettura ed urbanistica, arte, classici greci e latini, cucina e casa, diritto, economia*, etc) e non il codice del reparto (molto meno significativi per un visitatore o cliente)

Se si vogliono evidenziare, come appare logico supporre, tutti i dettagli dei libri che soddisfino i criteri sopra specificati l'istruzione SQL risulta essere del tipo:

#### Query 1 prima ipotesi

```
SELECT SottoReparto.Descrizione, Libro.Titolo, Libro.Prezzo, Libro.AnnoPubbl,  
Libro.Collana, Libro.DataArchiviazione, Libro.NumPagine, Libro.TipoRilegatura,  
Libro.NumCopie  
FROM Reparto, SottoReparto, Libro  
WHERE (Reparto.CodRep = SottoReparto.CodRep1)  
AND (SottoReparto.CodSottoRep = Libro.CodSottoRep1)  
AND ((SottoReparto.Descrizione = 'Da non perdere') OR (SottoReparto.Descrizione =  
'I più venduti'))  
AND Reparto.Descrizione = [inserire la descrizione del Reparto]  
ORDER BY SottoReparto.Descrizione;
```

dove i record risultanti sono mostrati per descrizione in ordine alfabetico.

Utilizzando gli operatori dell'algebra relazionale (trascurando la clausola ORDER BY) sarebbe

$$\Pi_A ( \sigma_P ( (Libro) \bowtie (Reparto \bowtie SottoReparto) ) )$$

CodSottoRep1 = CodSottoRep                      CodRep = CodRep1

con  $P = \{ Reparto.CodRep = SottoReparto.CodRep1 \}$  **AND** (SottoReparto.CodSottoRep = Libro.CodSottoRep1) **AND** ((SottoReparto.Descrizione = 'Da non perdere') **OR** (SottoReparto.Descrizione = 'I più venduti'))

ed  $A = \{ SottoReparto.Descrizione, Libro.Titolo, Libro.Prezzo, Libro.AnnoPubbl, Libro.Collana, Libro.DataArchiviazione, Libro.NumPagine, Libro.TipoRilegatura, Libro.NumCopie \}$

Nel caso si utilizzasse il codice del reparto anziché la sua descrizione la query risulterebbe meno complessa in quanto non sarebbe necessario il join tra le tabelle reparto e sottoreparto per ottenerne la descrizione.

#### Query 1 seconda ipotesi

```
SELECT SottoReparto.Descrizione, Libro.Titolo, Libro.Prezzo, Libro.AnnoPubbl,  
Libro.Collana, Libro.DataArchiviazione, Libro.NumPagine, Libro.TipoRilegatura,  
Libro.NumCopie  
FROM SottoReparto, Libro  
WHERE (SottoReparto.CodSottoRep = Libro.CodSottoRep1)  
AND ( (SottoReparto.Descrizione = 'Da non perdere') OR (SottoReparto.Descrizione =  
'I più venduti') )  
AND SottoReparto.CodRep1 = [inserire il codice del Reparto]  
ORDER BY SottoReparto.Descrizione;
```

Supponiamo infine di optare per una soluzione più semplice, che si limita a fornire il numero dei libri appartenenti ad una certa categoria, l'istruzione SQL risulta essere del tipo:

### **Query 1 terza ipotesi**

```
SELECT COUNT (*) AS numero  
FROM Reparto, SottoReparto, Libro  
WHERE (Reparto.CodRep = SottoReparto.CodRep1)  
AND (SottoReparto.CodSottoRep = Libro.CodSottoRep1)  
AND (SottoReparto.Descrizione = 'Da non perdere')  
AND Reparto.Descrizione = [inserire la descrizione del Reparto];
```

per la prima tipologia di sottoreparto e

```
SELECT COUNT (*) AS numero  
FROM Reparto, SottoReparto, Libro  
WHERE (Reparto.CodRep = SottoReparto.CodRep1)  
AND (SottoReparto.CodSottoRep = Libro.CodSottoRep1)  
AND (SottoReparto.Descrizione = 'I più venduti')  
AND Reparto.Descrizione = [inserire la descrizione del Reparto];
```

per la seconda tipologia di sottoreparto.

### **Quesito b2)**

**Q2:** Ordini: gli ordini in corso, con dettagli, di un dato utente.

Anche questa query richiesta e' un'interrogazione parametrica e l'istruzione SQL risulta essere del tipo:

```
SELECT Ordine.NumOrdine, Ordine.DataEmissione, Ordine.Stato,  
Ordine.ImportoTotaleDovuto, Cliente. Cognome, Cliente.Nome, Cliente.Indirizzo,  
Cliente.Città, Cliente.Telefono, Cliente.Email  
FROM Cliente, Ordine  
WHERE (Cliente.CodCliente = Ordine.CodCliente1)  
AND (Ordine.Stato = 'In corso')  
WHERE Cliente.CodCliente = [inserire il codice del Cliente]  
ORDER BY Ordine.NumOrdine;
```

dove gli ordini risultanti sono mostrati per numeri d'ordine crescenti.

## 6 Struttura del portale e relativa codifica.

### Quesito c1)

Dato che viene richiesto un negozio on line, il database deve essere disponibile su un server on line. Per organizzare il sito di presentazione del negozio si può pensare di utilizzare uno dei tanti tool presenti sul mercato per generare la parte grafica e uno dei tanti linguaggi attualmente esistente per la realizzazione di pagine dinamiche per il codice.

Molte sono le opzioni disponibili per lo sviluppo della componente server-side. Nella soluzione che presentiamo, abbiamo deciso di utilizzare il linguaggio di programmazione PHP che consente di arricchire le pagine Web di codice script che sarà eseguito direttamente sul server.

In particolare, PHP consente di implementare un motore di scripting server side molto diffuso e multipiattaforma con un buon supporto della connettività verso database diversi (ad es. dBase, Oracle, MySQL) attraverso componenti standard.

Nello specifico ipotizzeremo di interfacciarsi verso un DBMS MySQL, anch'esso multipiattaforma e piuttosto semplice da utilizzare

Ovviamente, per potere utilizzare PHP è necessario aver installato sul proprio sistema un Web Server (come ad esempio Apache).

Riepilogando si è deciso di utilizzare quindi per la realizzazione del portale e del relativo servizio Web una piattaforma **WAMP** basata su

Sistema Operativo.: **Windows** 2000 o 2003 server

Web Server: **APACHE**

Database Server: **MYSQL**

Linguaggio di programmazione lato server: **PHP**

mentre si utilizzerà il **linguaggio SQL** per la creazione delle tabelle e per le interrogazioni.

Esistono piattaforme complete open-source assolutamente gratuite che contengono il web server APACHE, il linguaggio di scripting lato server PHP ed il database relazionale MYSQL, (Ad esempio EasyPHP, PHPMyAdmin oppure Apache2triad) sia per ambiente Windows sia per ambiente Linux.

Naturalmente l'uso di una piattaforma **LAMP** (ossia con un operativo server anch'esso open source come **Linux**) sebbene da un lato porti l'indiscutibile vantaggio della totale gratuità rispetto agli alti costi di gestione della Microsoft, dall'altro può creare a tutti quegli utenti "non addetti ai lavori" che decidano di farne uso e che non abbiano un grosso livello di competenza tecnica informatica specifica più di qualche problema di configurazione, di customizzazione e di utilizzo non sempre compatibili con le ragioni di efficienza aziendale

## FILE: **ricerca-reparto.php**

```
<html>
<head>
<title> Visualizzazione Reparti libreria on-line </title>
</head>
<body>

<?php
$host = 'localhost'; //ipotizzando di accedere ad un server locale altrimenti
                //IP Address del server remoto
$user='visitatore' //un'unica tipologiadi user per tutti i visitatori
$database = 'libri';
$pswd = 'visitatore' // password per l'accesso al db uguale per tutti i visitatori

$connessione = mysql_connect($host, $ user, $ pswd) or die ("Impossibile connettersi
al server $host");

mysql_select_db($database, [$connessione]) or die ("Impossibile connettersi al
database $database");

$query = "select * from reparto";
$resultato = mysql_query($query, [$connessione]);
$numrighe=mysql_num_rows($resultato);

//INTESTAZIONE TABELLA DI OUTPUT
echo "<table>";
echo "<tr>";
echo "<td><b>Codice</b></td>";
echo "<td><b>Descrizione</b></td>";
echo "<\tr>";

for ($i=0; $i<$numrighe; $i+ +)
{
//REPERIMENTO INFORMAZIONI DALLA QUERY
$codreparto=mysql_result($resultato,$i,'codreparto');
$descrizione=mysql_result($resultato,$i,'descrizione');
//COSTRUZIONE i-ESIMA RIGA DELLA TABELLA
echo "<tr>";
echo "<td><b>" . $descrizione . "</b><\td>"; //prima colonna la descrizione
echo "<td>";
echo "
<form action='query1-reparto.php' method='POST'>
  <input type='hidden' name='CODREPARTO' value='$codreparto'>
  <input type='hidden' name='DESCREPARTO' value='$descrizione'>
  <input type='submit' value='Esegui'>
</form>
";
echo "<\td>"; //seconda colonna il tasto di esecuzione della query
echo "<\tr>"; //seconda colonna il tasto di esecuzione della query

}
echo "</table>";

mysql_free_result($resultato);
mysql_close($connessione);
?>
</body>
</html>
```

## FILE query1-reparto.php

```
<html>
<head>
<title> Dettaglio query su reparto </title>
</head>
<body>

<?php
$host = 'localhost'; //ipotizzando di accedere ad un server locale altrimenti
                //IP Address del server remoto
$user='visitatore' //un'unica tipologia di user per tutti i visitatori
$dbase = 'libri';
$pswd = visitatore" // password per l'accesso al db uguale per tutti i visitatori

$connessione = mysql_connect($host, $ user, $ pswd) or die ("Impossibile connettersi
al server $host");

mysql_select_db($dbase, [$connessione]) or die ("Impossibile connettersi al
database $dbase");

$codreparto = $_POST["CODREPARTO"] // prelev il codice del reparto passato
                // dal file ricerca-reparto.php
$descrizione = $_POST["DESCREPARTO"] // prelevo la descrizione passata dal
                // file ricerca-reparto.php

//query 1 prima ipotesi
$query = "SELECT SottoReparto.Descrizione, Libro.Titolo, Libro.Prezzo,
Libro.AnnoPubbl, Libro.Collana, Libro.DataArchiviazione, Libro.NumPagine,
Libro.TipoRilegatura, Libro.NumCopie
FROM Reparto, SottoReparto, Libro
WHERE (Reparto.CodRep = SottoReparto.CodRep1)
AND (SottoReparto.CodSottoRep = Libro.CodSottoRep1)
AND ((SottoReparto.Descrizione = 'Da non perdere') OR (SottoReparto.Descrizione = 'I
più venduti'))
AND Reparto.Descrizione = " . $descrizione . " ORDER BY SottoReparto.Descrizione;"

$risultato = mysql_query($query, [$connessione]);

$numrighe=mysql_num_rows($risultato);

//INTESTAZIONE TABELLA DI OUTPUT
echo "<table>";
echo "<tr>";
echo "<td><b>campo1-query1</b></td>";
echo "<td><b>campo2-query1</b></td>";
.....
echo "<td><b>campoN-query1</b></td>";
echo "<\tr>";

for ($i=0; $i<$numrighe; $i++)
{
//REPERIMENTO INFORMAZIONI DALLA QUERY
    $valcampo1=mysql_result($risultato,$i, 'campo1-query1');
    $valcampo2=mysql_result($risultato,$i, 'campo2-query1');
.....
    $valcampo1=mysql_result($risultato,$i, 'campoN-query1');

//COSTRUZIONE i-ESIMA RIGA DELLA TABELLA
```

```
    echo "<tr>";
    echo "<td>". $valcampo1."<\td>";
    echo "<td>". $valcampo2."<\td>";
    ....
    echo "<td>". $valcampoN."<\td>";
    echo "<tr>";

}
echo "</table>";

//IMPOSTO IL COLLEGAMENTO ALLA PAGINA CHIAMANTE
echo "<a href='ricerca-reparto.php'>Torna Dietro <\a>";

mysql_free_result($risultato);
mysql_close($connessione);
?>
</body>
</html>
```

Nel primo script (*ricerca-reparto.php*) abbiamo utilizzato la funzione che riceve come parametro il nome dell'host, il nome dell'utente e la password.

*\$connessione* = **mysql\_connect**(\$host, \$ user, \$ passwd)

per effettuare il collegamento con un host su cui gira un'istanza di MySQL.

Questa funzione restituisce un oggetto che rappresenta integralmente il nostro host e con il quale è possibile utilizzare la funzione

**mysql\_select\_db**(\$database, [ \$connessione])

per selezionare l'istanza corretta del database (essenziale in quanto l'istanza del database server può contenere più di un database).

Dopo aver selezionato il database, è possibile effettuare query su qualsiasi tabella tramite la funzione

*\$risultato* = **mysql\_query**(\$query, [ \$connessione]);

A questo punto l'oggetto *\$risultato* contiene il risultato della query.

Per l'estrazione dei dati dalla select abbiamo utilizzato la funzione

**mysql\_result**(\$risultato, \$i, \$nomecampo)

che restituisce i contenuti di una cella da un risultato MySQL dove l'argomento campo può essere l'indice della riga contenente tutti i dati. Per ottenere solo numeri di riga validi si è utilizzata la funzione

**mysql\_num\_rows** ( \$risultato )

che restituisce il numero di righe in un risultato. Questo comando è valido solo per le istruzioni SELECT.

Questa query ha estratto tutti i record della tabella reparti e con le opportune istruzioni HTML li abbiamo visualizzati all'interno di una tabella.

Per terminare correttamente una sessione di interrogazione ad MYSQL dobbiamo utilizzare le funzioni

**mysql\_free\_result** (\$risultato);

**mysql\_close** (\$connessione);

che rispettivamente libera tutta la memoria associata all'identificativo del risultato *\$risultato* e chiude la connessione al server MySQL associata all'identificativo di connessione *\$connessione* specificato permettendo eventualmente ad un altro client di utilizzarla.

Il secondo script (*query1-reparto.php*) si occupa di ottenere tutte le informazioni di dettaglio della query numero 1 (qui intesa nella sua prima versione) eseguita per ciascun reparto cliccando sull'apposito tasto "Esegui".

Utilizza la stessa logica di connessione e di reperimento delle informazioni dello script chiamante con la particolarità di dettagliare nella sua stringa di query il parametro ricevuto in modo nascosto nella variabile CODREPARTO attraverso il metodo POST presente nel FORM in esso contenuto.

Le variabili di tipo `$_POST` vengono impostate automaticamente dal PHP e fanno parte di un array globale di variabili passati allo script corrente attraverso il metodo POST del protocollo http e non visibili sulla barra di indirizzamento del browser. Se usassimo nel FORM il metodo GET le informazioni ricavate dal nostro form si troverebbero invece nell'array globale di variabili `$_GET` e sarebbero visibili sulla barra di indirizzamento del browser

### **Quesito c2)**

Relativamente alla richiesta di implementare una pagina del portale con l'accesso riservato agli utenti registrati dove poter visualizzare la composizione degli ordini in corso a lui relativi, è necessario predisporre a monte una pagina di autenticazione dell'utente, attraverso la quale (tramite un normale form html), l'utente introdurrà il proprio username e la propria password, che saranno validate lato server (ad esempio da uno script di autenticazione php). Se la validazione ha successo, verrà chiamata la funzione *mysql\_connect* con i valori di utente e password previsti per l'accesso di utenti registrati

*\$connessione = mysql\_connect(\$host, \$user, \$password) or die ("Impossibile connettersi al server \$host");*

Per garantire un discreto margine di sicurezza, sarebbe opportuno far sì che le informazioni riservate, come ad esempio le password, non compaiano nel sorgente html della pagina, e che inoltre, tramite opportuni meccanismi, viaggino criptate attraverso la rete.

Per eseguire query diverse da quella che, nello script proposto, produce l'elenco dei reparti è sufficiente inserire il codice nella query che deve essere eseguita e, quindi

*\$query = " select ... ";*

Inoltre, per questioni di sicurezza e per garantire la consistenza dei dati, sarebbe opportuno definire diverse tipologie di utenti del negozio on line (ad esempio utenti semplici, utenti registrati, gestori del magazzino libri e così via), definendo appropriate modalità di accesso ai dati. Determinare i livelli di accesso e i gruppi di utenti riconosciuti è compito dell'amministratore della base dati; la regola di massima è quella di concedere i permessi di accesso strettamente necessari. Di norma, tali privilegi vengono assegnati attraverso il comando GRANT, disponibile in MySQL.

Tramite questo comando

```
GRANT SELECT ON libri.* TO pippo@localhost IDENTIFIED BY 'password' WITH GRANT OPTION
```

Si concede all'utente l'autorizzazione all'esecuzione di query di estrazione dati, ma non la possibilità di modificarli. Il permesso di inserire nuovi record viene, ad esempio, concesso, invece, tramite il comando GRANT INSERT.

## **7 CONCLUSIONI FINALI.**

Ovviamente la soluzione presentata può risultare poco efficace e non ottimizzata, ma la mia scelta è stata quella di proporre una soluzione realmente fattibile nel tempo assegnato (6 ore)

Di conseguenza ci siamo limitati a prendere in considerazione gli aspetti essenziali del testo, tralasciando i problemi di connettività con gli uffici interni del negozio e con le procedure di aggiornamento delle tabelle che, per quanto riguarda i libri e i clienti, risultano ovviamente essere per il cliente a sola lettura. Sarà compito di altre parti del programma interagire attivamente con queste tabelle aggiornando la giacenza di magazzino, l'evasione degli ordini, la modifica delle categorie di appartenenza dei singoli titoli, l'aggiornamento dei dati del cliente, e così via.