

Dati storici sulle vendite per reparto

Si vuole realizzare un servizio che consenta di avere informazioni, tramite pagine Web, sull'andamento delle vendite nei diversi reparti di un grande centro commerciale.

I dati sono gestiti tramite un database che memorizza le vendite giornaliere nei reparti. Per ogni vendita occorre conoscere:

- nome del reparto,
- mese della vendita,
- giorno della vendita,
- importo totale della vendita giornaliera nel reparto.

Gli utenti hanno accesso al database delle vendite e possono richiedere l'andamento di un particolare reparto in un certo mese. L'applicazione risponde alle interrogazioni dell'utente presentando un'applet che visualizza graficamente l'andamento del reparto nel mese indicato.

La realizzazione di questa applicazione prevede l'utilizzo di varie tecnologie:

- **JDBC**, per l'accesso alla base di dati;
- **pagine JSP**, per la raccolta delle informazioni dall'utente e per attivare le interrogazioni alla base di dati;
- **applet**, per mostrare i dati numerici tramite dei grafici.

Viene riportata di seguito una traccia della soluzione, lasciando allo studente il compito di completarla e migliorarla.

L'applicazione si compone di due pagine Web.

La prima pagina è composta da un modulo tramite il quale l'utente può scegliere il nome del reparto e il mese. La seconda pagina contiene i risultati dell'interrogazione visualizzati all'interno di un'applet.

L'applet visualizza, usando un grafico a barre, le vendite di un certo reparto riferite a un periodo di un mese. I valori del reparto vengono passati all'applet tramite i parametri specificati dal tag `APPLET`. I parametri hanno i seguenti nomi:

- g1,
- g2,
- ...
- g31.

Ogni parametro si riferisce alle vendite del reparto relative a un giorno del mese. I giorni a cui non è associata la vendita, per esempio perché sono giorni festivi, non sono riportati come parametri. L'applet, quando legge i valori dei parametri, si accorge della mancanza di un parametro controllando il valore di ritorno del metodo `getParameter`. Se il metodo restituisce un valore uguale a `null` significa che il valore in quel giorno non è disponibile e quindi si preoccuperà di non visualizzare l'istogramma relativo a quel giorno.

L'applet legge i parametri usando il ciclo mostrato dal seguente frammento di codice:

```
private double vendite [] = new double[31];
```

```

public void init()
{
    String param;
    for(int i=0; i<31; i++)
    {
        param = getParameter("g"+(i+1));
        if (param == null)
        {
            // Il parametro e' mancante:
            // assegna il valore 0 alla vendita in quel giorno.
            vendite [i] = 0;
        }
        else
        {
            // Converti il parametro in numero
            try
            {
                vendite [i] = Double.valueOf(param).doubleValue();
            }
            catch(Exception e)
            {
                vendite[i] = 0;
            }
        }
    }
}

```

La prima pagina Web, contenente il modulo, richiama una pagina JSP che ha i seguenti compiti:

- interrogare il database per recuperare le vendite relative al mese e al reparto scelto dall'utente,
- creare una nuova pagina Web contenente l'applet,
- generare i parametri dell'applet con i valori letti dal database.

PAGINA WEB

```

<HTML>
<HEAD>
<TITLE>Vendite nei reparti </TITLE>
</HEAD>
<BODY>
<H1>Scegli il reparto e il mese:</H1>
<FORM ACTION="Vendite.jsp" METHOD="POST">
Codice reparto: <INPUT TYPE="TEXT" SIZE="20" NAME="reparto"><BR>
Mese:
<SELECT NAME="mese">
<OPTION VALUE="1" selected>Gennaio</OPTION>
<OPTION VALUE="2">Febbraio</OPTION>
<OPTION VALUE="3">Marzo</OPTION>
<OPTION VALUE="4">Aprile</OPTION>
<OPTION VALUE="5">Maggio</OPTION>
<OPTION VALUE="6">Giugno</OPTION>
<OPTION VALUE="7">Luglio</OPTION>
<OPTION VALUE="8">Agosto</OPTION>

```

```

<OPTION VALUE="9">Settembre</OPTION>
<OPTION VALUE="10">Ottobre</OPTION>
<OPTION VALUE="11">Novembre</OPTION>
<OPTION VALUE="12">Dicembre</OPTION>
</SELECT>
<BR>
<INPUT TYPE=SUBMIT NAME=submit VALUE="Mostra grafico">
</FORM>
</BODY>
</HTML>

```

La pagina Web precedente contiene il modulo che richiama la pagina *Vendite.jsp*. I campi del modulo sono il codice del reparto, realizzato come casella di testo, e il mese relativo alle vendite, realizzato con un elenco contenente i nomi dei mesi, ai quali sono associati i valori da 1 a 12.

Il database che gestisce le vendite è implementato in Access ed è contenuto nel file *CentroCommerciale.mdb*. Esso è associato tramite ODBC al nome *db_reparti*, specificato come nome della sorgente di dati (DSN).

Il database contiene la tabella dei *Reparti* (codice e descrizione di ogni reparto) e la tabella delle *Vendite* con campo ID (contatore), codice del reparto (chiave esterna), mese, giorno e importo della vendita.

La tabella *Vendite* ha la seguente struttura:

Nome	Tipo	Dimensione
ID	Contatore	
CodReparto	Testo	20
Mese	Numerico	Intero
Giorno	Numerico	Intero
Valore	Numerico	Precisione singola

L'interrogazione eseguita su questo database ha lo scopo di recuperare tutti i giorni e i valori associati a un reparto e a un mese specificati come parametri. La stringa SQL che identifica la query è la seguente:

```

String query = "SELECT * FROM Vendite"+
               "WHERE CodReparto = ? AND Mese = ? "+
               "ORDER BY Giorno";

```

I risultati della query sono ordinati in base al giorno; il valore dei parametri è indicato usando i punti interrogativi, secondo la notazione di Java.

Il valore dei parametri viene letto con il metodo **getParameter** e successivamente viene inserito nella stringa SQL usando il metodo **setString**.

```

String p1 = request.getParameter("reparto");
String p2 = request.getParameter("mese");
. . .
stmt.setString(1, p1);
stmt.setString(2, p2);

```

I risultati restituiti dall'esecuzione della query vengono memorizzati nella variabile *rs* di tipo *ResultSet*. Questi risultati, formati dai due campi giorno e valore, devono essere passati come parametri all'applet che li userà per creare l'istogramma. Il seguente frammento di codice realizza il ciclo per generare i parametri dell'applet:

```
<APPLET CODE=Istogramma.class WIDTH=300 HEIGHT=200>
<%
// ciclo per impostare i parametri dell'applet
while (rs.next())
{
    String giorno = rs.getString("giorno");
    String valore = rs.getString("valore");
%>
<PARAM NAME=g<%=giorno%> VALUE=<%=valore%> >
<%
} // fine ciclo while
%>
Applet non eseguibile.
</APPLET>
```

Per ogni riga presente nel *ResultSet* viene recuperato il campo *giorno* e il campo *valore* usando il metodo *getString*. Queste informazioni vengono aggiunte al tag PARAM usando i tag JSP di espressione `<%= %>`.

Un miglioramento che si può apportare alla precedente applicazione riguarda la prima pagina Web, quella contenente il modulo. Attualmente il codice del reparto viene richiesto all'utente tramite una casella di testo. Questa modalità ha lo svantaggio di costringere l'utente a ricordare i codici dei reparti, mentre sarebbe più conveniente individuare i reparti attraverso il loro nome.

In situazioni come queste, la soluzione è presentare all'utente un menu a tendina contenente l'elenco dei nomi dei reparti presenti nel database, mantenendone nascosto il codice. Per ottenere questo risultato, la pagina Web deve essere generata dinamicamente usando le informazioni lette dal database.

Questa nuova pagina JSP esegue un'interrogazione per recuperare tutti i nomi dei reparti nel seguente modo:

```
SELECT Codice, Descrizione FROM Reparti
```

Successivamente costruisce dinamicamente l'elenco in HTML (tag SELECT) inserendo i codici dei reparti come valore e i nomi come descrizione, nei singoli tag OPTION. In questo modo l'utente sceglie nell'elenco dei nomi di reparto e il form HTML passa come parametro alla pagina JSP il codice corrispondente.