

Ipotesi di soluzione seconda prova di Informatica Mercurio 2013

Traccia della domanda teorica

Un'attività fondamentale della manutenzione dei sistemi informatici è la protezione dei sistemi stessi da eventi accidentali o da attacchi intenzionali, che possono comportare, ad esempio, il danneggiamento delle informazioni o la loro illecita sottrazione. L'esigenza di proteggere la sicurezza dei sistemi informatici è particolarmente attuale, anche a motivo dell'elevato livello di interconnessione dei sistemi mediante collegamenti in rete.

Dopo aver evidenziato i rischi concettualmente connessi a breccie nei sistemi di sicurezza di un sistema informatico o di una rete, il candidato esponga i principali meccanismi di attacco e le contromisure preventive che possono essere messe in atto.

Sviluppo della domanda teorica

Definizione di sicurezza informatica

E' il ramo dell'informatica che si occupa dell'analisi del rischio e della vulnerabilità dei sistemi informatici e delle azioni di protezione dell'integrità dei sistemi sia dal punto di vista hardware che software.

Analisi del rischio

Si può suddividere il danno sui dati in tre tipi di evento:

- **disponibilità dei dati:** fa riferimento alla perdita di informazioni
- **integrità dei dati:** fa riferimento alla perdita di coerenza delle informazioni tra loro
- **riservatezza dei dati:** fa riferimento ad accessi non autorizzati

Questi danni possono essere determinati da:

- **eventi accidentali:** si tratta di problemi tecnologici od errori umani
- **eventi indesiderati:** si tratta di accessi non autorizzati realizzati in modo fraudolento

Eventi Accidentali

Gli eventi accidentali possono essere:

- discontinuità dell'alimentazione
- difetti di esecuzione dei programmi
- errori umani nell'applicazione delle procedure
- eventi catastrofici

Contromisure per gli eventi accidentali

- Alimentazione in batteria tampone
- Backup dei dati
- Archiviazione in diversità di spazio

Eventi indesiderati

Gli eventi indesiderati si possono dividere in due categorie

- **Accessi non autorizzati:** si tratta di accessi fraudolenti ai dati effettuati da chi ha accesso alle macchine del sistema.
- **Hacking:** si tratta di attacchi al sistema attraverso la rete Internet sfruttando difetti dei programmi e dei sistemi.

Contromisure per gli eventi indesiderati

- **Sistemi di autenticazione:** sono limitazioni all'accesso ai dati basati su credenziali rilasciate agli utenti autorizzati
- **Firewall:** sono limitazioni all'accesso ai servizi interni della rete da parte di applicazioni che accedono dalla rete Internet
- **Crittografia:** sono tecniche di cifratura dei dati che ostacolano la lettura delle informazioni trasferite sulla rete Internet da parte di programmi non autorizzati. La tecnica più diffusa è la crittografia asimmetrica che prevede due distinte chiavi: una chiave pubblica per la cifratura ed una chiave privata per la de-cifratura.
- **Antivirus:** sono applicazioni che intercettano applicazioni dannose per il sistema arrivate attraverso al rete Internet oppure attraverso media rimovibili

ITIS Odone Belluzzi

Traccia dell'esercizio

Un grande museo deve realizzare un Sistema Informatico per la gestione della biglietteria online.

La biglietteria online consentirà di acquistare i biglietti sia per la visita del museo (biglietto base) sia per le singole esposizioni tematiche (biglietto evento). La visita e le esposizioni, sono caratterizzate da un codice identificativo, da un titolo, dalla tariffa ordinaria, dalla data di inizio e data di fine (queste ultime non valorizzate nel caso della visita).

Particolari categorie di visitatori hanno diritto ad una riduzione sulle tariffe ordinarie. Per ogni categoria è registrato il codice, la descrizione, il tipo di documento da esibire per avere diritto alla agevolazione e la percentuale di sconto.

Al biglietto può essere associato l'acquisto di servizi o prodotti accessori (ad es.: audio-guida, accompagnatore specializzato, catalogo, ecc.) dei quali viene memorizzato il codice, la descrizione e il prezzo unitario.

I biglietti acquistati dai clienti non sono nominativi. Essi hanno un codice identificativo e una data di validità.

Il candidato formuli ogni ipotesi accessoria che ritenga opportuna e proceda alla progettazione di una base di dati adeguata, formalizzando:

- lo schema concettuale;
- lo schema logico relazionale;
- i comandi necessari alla generazione di almeno due tabelle.

Il candidato indichi inoltre i comandi del linguaggio di query necessari a:

1. elencare i codici dei biglietti emessi per un determinato giorno;
2. elencare i titoli e le date delle esposizioni tematiche che si sono tenute nel periodo gennaio – 31 dicembre di un determinato anno;
3. indicare il numero di biglietti emessi per una determinata esposizione;
4. calcolare il ricavato della vendita dei biglietti di una determinata esposizione.

Il candidato descriva infine il codice necessario alla realizzazione di una porzione a sua scelta del sito del museo.

Analisi del problema ed ipotesi aggiuntive

La traccia richiede che il servizio sia online quindi l'acquisto deve poter effettuato, oltre che dagli sportelli della biglietteria anche attraverso la rete Internet. Questa specifica richiede che la banca dati di gestione sia collocata su un host accessibile dalla rete Internet e che l'utente si possa interfacciare con la banca dati attraverso una applicazione web.

L'ipotesi che il biglietto non sia nominativo esclude la presenza di sistemi di autenticazione e servizi di e-commerce quindi il sistema si limita all'emissione del biglietto non affrontando il problema dell'effettivo pagamento.

Entità

Si parte individuando le "Entità" cioè gli insiemi di informazioni la cui esistenza è indipendente dalla presenza di altri insiemi.

Entità Servizio: rappresenta l'insieme dei servizi principali offerti dal museo che si distingue in "visita" ed "esposizione" e quindi presenta una struttura di tipo IS-A. Le due sotto-entità condividono la Descrizione sotto forma di testo e la Tariffa espressa in valuta. La versione visita (parte permanente del museo) non ha altri attributi la versione "esposizione" (parte temporanea del museo) sono definite anche le date di inizio e fine. Il codice proposto dalle specifiche può essere considerato chiave candidata.

Entità Biglietto: rappresenta l'insieme dei biglietti effettivamente emessi per i vari servizi; secondo le specifiche è sicuramente indispensabile prevedere la data di validità e per semplificare si suppone che sia riferita ad un singolo giorno e non ad un periodo. Sebbene non sia esplicitamente richiesto dalla traccia può essere utile l'attributo data di emissione che può non coincidere con la data di validità. A causa dell'ipotesi di biglietto non nominativo non comprende informazioni sull'acquirente. Il codice proposto dalle specifiche può essere considerato chiave candidata.

Entità Accessorio: rappresenta l'insieme dei servizi accessori offerti dal museo ed è dotato degli attributi Descrizione sotto forma di testo e Prezzo Unitario espresso in valuta. Il codice

ITIS Odone Belluzzi

proposto dalle specifiche può essere considerato chiave candidata.

Entità Categoria: rappresenta l'insieme delle possibili agevolazioni riconosciute agli utenti. Ogni agevolazione è definita da una Descrizione sotto forma di testo, da una percentuale di sconto espressa da un intero e da un tipo di documento necessario per ottenere l'agevolazione. Per garantire la normalizzazione della banca dati il tipo di documento è descritto in una distinta entità. Per semplificare la struttura si ipotizza che per ogni categoria di esenzione sia valido un solo tipo di documento. Il codice proposto dalle specifiche può essere considerato chiave candidata.

Entità Documento: rappresenta l'insieme dei possibili documenti che danno diritto ad una esenzione. E' composto da una chiave artificiale di identificazione usata come chiave candidata e da una descrizione.

Associazioni

Dopo aver identificato le Entità si cercano le associazioni tra le entità identificando la molteplicità. Si cercano le associazioni 1:N ed N:M eseguendo le "regole di lettura" ed assegnando un nome alla associazione che in genere è un verbo per le associazioni 1:N ed un sostantivo per le associazioni N:M.

Associazione Biglietto-Servizio (valido_per): un biglietto emesso deve fare riferimento ad un singolo Servizio offerto mentre ad un servizio possono corrispondere 0 o molti biglietti emessi ma l'associazione è facoltativa. Si tratta di una associazione 1:N tra Biglietto e Servizio parziale dal lato servizio.

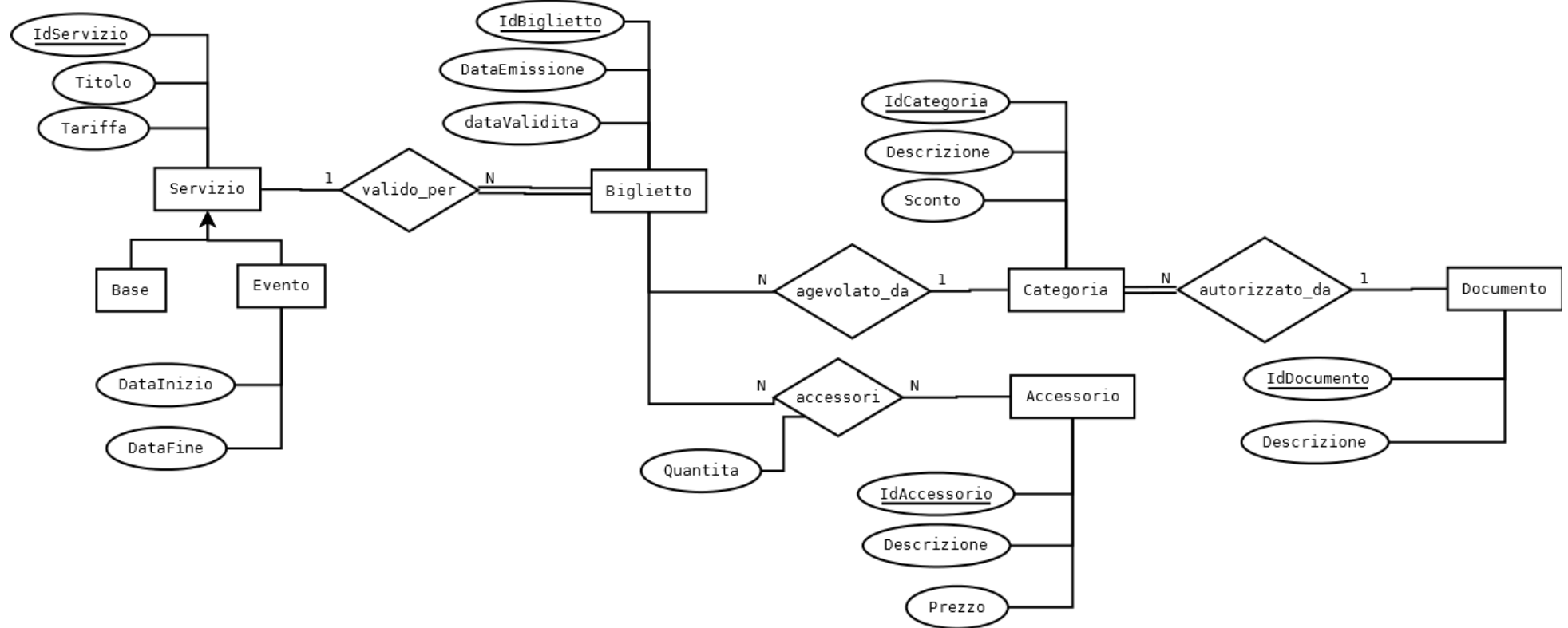
Associazione Biglietto-Categoria (agevolato_da): un biglietto emesso può fare riferimento ad una categoria di agevolazione mentre una categoria di agevolazione può essere applicata a 0 o molti biglietti emessi. Si tratta di una associazione 1:N tra Biglietto e Categoria parziale da entrambi i lati.

Associazione Categoria-Documento (autorizzato_da): una categoria di agevolazione deve fare riferimento ad un singolo tipo di documento mentre un tipo di documento può autorizzare 0 o molte categorie di esenzione(ad esempio tutte le categorie basate sull'età richiedono un documento di identità). Si tratta di una associazione 1:N tra Categoria e Documento parziale dal lato Documento.

Associazione Biglietto-Accessorio (accessori): un biglietto emesso può essere associato a 0 o molti servizi accessori e ogni possibile servizio accessorio può essere associato a 0 o molti biglietti emessi. Si tratta di una associazione N:M tra Biglietto ed Accessorio parziale da entrambi i lati. La specifica che indica come "prezzo unitario" il prezzo del servizio accessorio suggerisce che sia possibile associare ad uno stesso biglietto una quantità multipla di accessori uguali (ad esempio l'acquisto di più cataloghi) e questo richiede la definizione dell'attributo intero "quantità" nell'associazione.

Modello concettuale

Dall'analisi ed ipotesi aggiuntive deriva una rappresentazione grafica del modello concettuale secondo lo schema ER. Lo schema evidenzia entità, attributi, chiavi primarie ed associazioni tra entità, totalità o parzialità delle associazioni.



Note:

- I rettangoli rappresentano le entità, i rombi rappresentano le associazioni, gli ovali rappresentano gli attributi
- Le sottolineature rappresentano le chiavi primarie ottenute dalle chiavi candidate
- Le linee doppie rappresentano associazioni totali
- La frecce rappresentano associazioni IS-A

ITIS Odone Belluzzi

Modello logico

Dal modello ER si ricava il modello logico ipotizzando di scegliere uno schema di tipo relazionale.

Le regole di trasformazione sono:

- Le entità diventano relazioni
- Le chiavi primarie delle entità diventano chiavi primarie delle relazioni
- Gli attributi delle entità diventano attributi delle relazioni
- Le associazioni 1:N diventano chiavi esterne nella relazione dal lato 1 ottenuta esportando la chiave primaria del lato N
- Le associazioni N:M diventano relazioni con chiave primaria ottenuta combinando due chiavi esterne ottenute esportando le chiavi primarie dei due lati della associazione
- Gli attributi delle associazioni N:M diventano attributi della relazione definita nel punto precedente
- Le associazioni IS-A si possono risolvere con due strategie
 - un'unica relazione con attributi facoltativi conveniente nel caso in cui le parti della gerarchie differiscono poco tra di loro
 - una relazione principale con gli attributi comuni e relazioni secondarie associate 1:1 con la principale per le parti non comuni conveniente nel caso in cui le parti della gerarchie differiscono molto tra di loro

Definizione delle relazioni

Per ogni relazione definita con le regole precedenti viene descritto lo schema specificando i tipi di dato di ciascuna colonna.

Servizio(IdServizio:Integer,
Descrizione:String,
Tariffa: Decimal,
DataInizio: Date,
DataFine: Date)

Biglietto(IdBiglietto:Integer,
DataEmissione: Date,
DataValidita:Date,
IdServizio:Integer,
IdCategoria:Integer)

Categoria(IdCategoria: Integer,
Descrizione: String,
IdDocumento: Integer)

Documento(IdDocumento: Integer,
Descrizione: String)

Accessorio(IdAccessorio: Integer,
Descrizione: String
Prezzo: Decimal)

Accessori(IdBiglietto: Integer,
IdAccessorio:Integer,
Quantita: Integer)

ITIS Odone Belluzzi

Vincoli di integrità

I vincoli possono essere imposti sulle singole relazioni (intra-relazionali) oppure possono essere tra le relazioni (inter-relazionali).

Vincoli intra-relazionali

Per la rappresentazione dell'associazione IS-A si sceglie la soluzione con un'unica relazione quindi i due attributi DataInizio e DataFine possono essere nulli

Se DataInizio e DataFine sono valorizzate (non nulle) DataInizio deve precedere o coincidere con DataFine.

La relazione tra Biglietto e Categoria è parziale quindi la chiave esportata IdCategoria in Biglietto può essere nulla (nessuno sconto)

Tutti gli altri attributi sono obbligatori (non nulli)

La data di emissione del biglietto deve essere antecedente o uguale alla data di validità.

Per la rappresentazione di Tariffe e Prezzi si sceglie il tipo Decimal con il vincolo che deve essere maggiore di 0.

Per la rappresentazione della quantità di servizi accessori uguali associati ad un biglietto si sceglie il tipo Integer con il vincolo che deve essere maggiore di 0

Vincoli inter-relazionali

Integrità referenziale

Biglietto.IdServizio contenuto in Servizio.IdServizio

Biglietto.IdCategoria contenuto in Categoria.IdCategoria oppure nullo

Categoria.IdDocumento contenuto in Documento.IdDocumento

Accessori.IdBiglietto contenuto in Biglietto.IdBiglietto

Accessori.IdAccessorio contenuto in Accessorio.IdAccessorio

Altri vincoli inter-relazionali

La data di validità di un biglietto per evento deve cadere all'interno della durata dell'evento.

Possibili alternative di progetto

Implementazione di dati storici

Il modello proposto fotografa la situazione corrente del museo senza tenere conto della storicità dei dati.

Potrebbe essere ad esempio necessario cambiare nel tempo le tariffe della visita base ed i prezzi dei servizi accessori. Con la soluzione proposta il cambiamento non può essere fatto direttamente sul preesistente record di Servizio e Accessorio perchè porterebbe ad una incongruenza "storica" (un biglietto emesso nel passato con una tariffa di un tempo successivo).

Si può risolvere il problema in due modi:

- Generazione di nuovi record per visita base e accessori al cambiamento del prezzo; questa scelta impone l'aggiunta di due attributi 'date' per definire il periodo di validità di una tariffa o prezzo e l'utilizzo per nuovi biglietti solo di prezzi e tariffe in corso di validità lasciando gli altri record come riferimento per gli acquisti passati
- Aggiunta dell'attributo tariffa applicata nell'entità biglietto come copia del valore attuale di Tariffa di Servizio e aggiunta dell'attributo prezzo applicato nella associazione Accessori come copia del valore attuale di Prezzo del servizio accessorio. Questi valori conservano la loro validità storica quando i valori attuali cambiano.

Autorizzazione con documenti multipli

Si può rimuovere l'ipotesi aggiuntiva di singolo tipo di documento per autorizzare l'agevolazione consentendo che una stessa agevolazione sia ottenuta a partire da più di un tipo di documento (ad esempio Carta d'Identità, Patente, Passaporto).

La modifica richiede la creazione nel modello ER di una associazione N:M tra Categoria e Documento che diventa nel modello relazionale una relazione con chiave primaria composta dalle due chiavi esterne esportate da Categoria e Documento

ITIS Odone Belluzzi

Generazione delle tabelle

Si ipotizza di usare per il modello Fisico un DBMS che rispetti lo standard SQL.

In questo caso la generazione delle tabelle viene realizzata usando i comandi del DDL (Data Description Language)

Creazione della tabella Servizio

```
CREATE TABLE Servizio (  
    IdServizio bigint(20) unsigned NOT NULL AUTO_INCREMENT,  
    Titolo varchar(255) NOT NULL,  
    Tariffa decimal(10,0) NOT NULL,  
    DataInizio date DEFAULT NULL,  
    DataFine date DEFAULT NULL,  
    PRIMARY KEY (IdServizio),  
    KEY Titolo(Titolo)  
)
```

Creazione della tabella Biglietto

```
CREATE TABLE Biglietto (  
    IdBiglietto bigint(20) unsigned NOT NULL AUTO_INCREMENT,  
    DataEmissione date NOT NULL,  
    DataValidita date NOT NULL,  
    IdServizio bigint(20) unsigned NOT NULL,  
    IdCategoria bigint(20) unsigned DEF AULT NULL,  
    PRIMARY KEY (IdEmesso),  
    KEY DataEmissione(DataEmissione)  
    KEY DataValidita(DataValidita),  
    KEY IdServizio(IdServizio),  
    KEY IdCategoria(IdCategoria),  
    FOREIGN KEY(IdServizio) REFERENCES Servizio(IdServizio)  
    FOREIGN KEY(IdCategoria)REFERENCES Categoria(IdCategoria);  
)
```

N.B. La presenza della chiave esterna IdCategoria nella tabella Biglietto richiede la preventiva realizzazione della tabella Categoria che a sua volta richiede la realizzazione della tabella Documento

Query

1. elencare i codici dei biglietti emessi per un determinato giorno;

```
$query=" SELECT      COUNT(IdBiglietto) AS contatore
        FROM        Biglietto
        WHERE       DataValidita=$dataValidita";
```

La query parametrica fa riferimento ad una ipotesi di embedding nel linguaggio ospite PHP dove le variabili vengono identificate dall'operatore \$ quindi si ipotizza che la query sia inserita in uno script che riceve l'informazione parametrica e la memorizza nella variabile \$dataValidita. Il risultato è una tabella avente come schema un'unica colonna di nome contatore e con un'unica istanza contenente il valore del conteggio.

Versione ASP

```
query=" SELECT      COUNT(IdBiglietto) AS contatore
        FROM        Biglietto
        WHERE       DataValidita="+dataValidita
```

2 .elencare i titoli e le date delle esposizioni tematiche che si sono tenute nel periodo gennaio – 31 dicembre di un determinato anno;

```
$query=" SELECT      Titolo,DataInizio,DataFine
        FROM        Servizio
        WHERE       DataInizio >= '$anno-01-01'
        AND        DataFine <= '$anno-12-31'
```

Anche in questo caso la query parametrica è inserita in uno script PHP che riceve come parametro l'informazione relativa all'anno richiesto e compone la data.

Il risultato è una tabella avente come schema Titolo,DataInizio e DataFine e una cardinalità pari agli eventi registrati nell'anno selezionato.

Con la scelta fatta vengono esclusi gli eventi che pur essendosi svolti nell'anno selezionato sono iniziati nell'anno precedente oppure sono terminati nell'anno successivo.

Versione ASP

```
query=" SELECT      Titolo,DataInizio,DataFine
        FROM        Servizio
        WHERE       YEAR(DataInizio)="+anno+"
        AND        YEAR(DataFine) "+anno
```

3. indicare il numero di biglietti emessi per una determinata esposizione;

```
$query=" SELECT      COUNT( t1.IdServizio ) AS contatore
        FROM        Servizio AS t1
                   JOIN
                   Biglietto AS t2
                   ON t1.IdServizio = t2.IdServizio
        WHERE       t1.IdServizio =$idServizio";
```

Anche in questo caso la query parametrica è inserita in uno script PHP che riceve come parametro l'informazione relativa all'id del servizio richiesto.

Il risultato è una tabella avente come schema un'unica colonna di nome contatore e con un'unica istanza contenente il valore del conteggio. La query è un giunzione naturale tra Servizio e Biglietto con una restrizione all'IdServizio richiesto.

4. calcolare il ricavato della vendita dei biglietti di una determinata esposizione.

La complessità della query di calcolo del ricavato dipende dall'interpretazione che si dà della richiesta.

Una prima versione particolarmente semplice si ottiene con due ipotesi semplificative:

- Si calcola solo il ricavato dalla tariffa di biglietto escludendo gli accessori.
- Si calcola solo la tariffa intera escludendo gli sconti

La query che si ottiene con queste ipotesi è:

```
$query=" SELECT SUM( t1.Tariffa ) AS ricavato
        FROM Servizio AS t1
        JOIN
        Biglietto AS t2
        ON t1.IdServizio = t2.IdServizio
        WHERE t1.IdServizio =$idServizio";
```

Questa soluzione però è troppo semplicistica almeno per la parte riguardante gli sconti. E' quindi indispensabile applicare l'eventuale sconto. Si deve introdurre nella query anche la tabella Categoria per individuare e calcolare eventuali sconti. La scelta di ottenere lo sconto tramite una relazione parziale con Categoria crea il problema dell'esclusione dei biglietti privi di sconto dalla giunzione. La soluzione generale consiste nell'effettuare due query; in una query basata su un natural join si estraggono i biglietti scontati mentre nella seconda basata su un left join si estraggono i biglietti senza sconto.

Biglietti scontati

```
$query="SELECT SUM(t1.Tariffa-((t1.Tariffa /100)*t3.Sconto)) AS subtotale
        FROM Servizio AS t1
        JOIN
        Biglietto AS t2
        ON t1.IdServizio = t2.IdServizio
        JOIN
        Categoria AS t3
        ON t2.IdCategoria=t3.IdCategoria
        WHERE t1.IdServizio =$idServizio";
```

Questa query estrae solo i biglietti scontati perché i biglietti non scontati, che contengono NULL nel campo non trovano una corrispondenza tra chiave esterna IdCategoria e chiave primaria della tabella Categoria.

Poiché tutti i biglietti estratti sono scontati si può applicare il calcolo dello sconto nella proiezione.

Biglietti non scontati

```
$query=" SELECT SUM(t1.Tariffa) AS subtotale
        FROM Servizio AS t1
        JOIN
        Biglietto AS t2
        ON t1.IdServizio = t2.IdServizio
        LEFT JOIN
        Categoria AS t3
        ON t2.IdCategoria=t3.IdCategoria
        WHERE t1.IdServizio =$idServizio
        AND ISNULL( t3.IdCategoria );
```

Questa query estrae solo i biglietti non scontati perché i biglietti non scontati applica la

ITIS Odone Belluzzi

restrizione di nullità della chiave esterna al left join con Categoria escludendo quindi tutti i biglietti scontati.

Si ottengono in questo modo due subtotali che vanno sommati. Questa somma può essere effettuata per via procedurale attraverso uno script della applicazione ma può anche essere effettuata in modo non procedurale sfruttando l'operatore SQL di unione che si può applicare a tabelle aventi lo stesso schema.

Una possibile soluzione alternativa che semplifica la query consiste nel rendere obbligatoria la chiave esterna IdCategoria prevedendo una categoria fittizia priva di sconto con il valore di Sconto a 0. Questa soluzione consente di effettuare un'unica query ma trasferisce un vincolo di integrità nella gestione dell'interfaccia.

```
$query="
SELECT SUM(t4.subtotale) AS totale
FROM (SELECT SUM(t1.Tariffa - ((t1.Tariffa / 100) * t3.Sconto)) AS subtotale
      FROM Servizio AS t1
      JOIN Biglietto AS t2
      ON t1.IdServizio = t2.IdServizio
      JOIN Categoria AS t3
      ON t2.IdCategoria = t3.IdCategoria
      WHERE t1.IdServizio = $idServizio

UNION
SELECT SUM(t1.Tariffa) AS subtotale
FROM Servizio AS t1
JOIN Biglietto AS t2
ON t1.IdServizio = t2.IdServizio
LEFT JOIN Categoria AS t3
ON t2.IdCategoria = t3.IdCategoria
WHERE t1.IdServizio = $idServizio
AND ISNULL(t3.IdCategoria)
) AS t4";
```

Si tratta di una query annidata in cui le relazioni usate nella clausola FROM sono a loro volta ottenute da altre SELECT poste in UNION tra loro.

La UNION produce una nuova relazione che rappresenta l'unione delle due relazioni.

Ricavato con accessori

Per aggiungere gli accessori si può procedere con la stessa tecnica usata per differenziare i biglietti scontati da quelli non scontati aggiungendo una terza sub-query che generi il subtotale degli accessori per l'evento considerato.

```
$query="SELECT SUM(t6.subtotale) AS totale
FROM (SELECT ... ricavato biglietti scontati (vedi sopra)...
      UNION
      SELECT ... ricavato biglietti non scontati (vedi sopra)...
      UNION
      SELECT SUM(t4.prezzo * t5.quantita) AS subtotale
      FROM biglietto AS t2
      JOIN accessori AS t4 ON t2.IdBiglietto = t4.IdBiglietto
      JOIN accessorio AS t5 ON t4.IdAccessorio = t5.IdAccessorio
      WHERE t1.IdBiglietto = $idBiglietto
) AS t6";
```

ITIS Odone Belluzzi

Porzione di codice

La web-application prevede un grande numero di pagine web attive che svolgono funzioni di consultazione dell'offerta, registrazione dei biglietti, manutenzione del sistema.

Consultazione dell'offerta di Eventi

Versione PHP

Nella versione PHP si ipotizza che la banca dati sia gestita da un DBMS MySQL.

```
<?php
    //crea una connessione al database MySQL
    $connection = new mysqli('localhost', 'nobody', '', 'museo');
    //crea la stringa di query: tutti gli eventi attuali e futuri
    $query="SELECT * FROM Servizio WHERE DataFine >=Date(NOW())";
    //effettua la query per ottenere i dati
    $result = $connection->query($query);
    //disegna una tabella
    echo"<table>";
    //estrae i record come tabella associativa
    while ($row = $result->fetch_assoc()) {
        echo"<tr><td>".$row['Titolo']."</td>";
        echo"<td>".$row['Tariffa']."</td>";
        echo"<td>".$row['DataInizio']."</td>";
        echo"<td>".$row['DataFine']."</td></tr>";
    }
    echo"</table>";
    $connection->close();
?>
```

Versione ASP

Nella versione ASP si ipotizza che la banca dati sia gestita da un DBMS MS-SQL.

```
<%
    'crea una connessione al database
    Set connection = Server.CreateObject("ADODB.Connection")
    connection.Provider="Microsoft.SQLSERVER.CE.OLEDB.3.5"
    connection.Open "museo.sdf"
    'crea la stringa di query: tutti gli eventi attuali e futuri
    query="SELECT * FROM Servizio WHERE DataFine >=Date(NOW())"
    'effettua la query per ottenere i dati
    Set Recordset = Server.CreateObject("ADODB.Recordset")
    Recordset.Open query,connection
    'disegna una tabella
    Response.Write("<table>")
    Do While not Recordset.EOF
        'estrae i dati dal recordset
        Response.Write ("<tr><td>"+Recordset("Titolo")+"</td>")
        Response.Write ("<tr><td>"+Recordset("Tariffa")+"</td>")
        Response.Write ("<tr><td>"+Recordset("DataInizio")+"</td>")
        Response.Write ("<tr><td>"+Recordset("DataFine")+"</td></tr>")
        Recordset.MoveNext
    Loop
    Response.Write("</table>")
    Recordset.Close
    connection.Close
%>
```