

INFORMATICA

Esame di Stato di Istituto Tecnico Commerciale Corso di Ordinamento Indirizzo: Programmatori

Andrea Pollini

TRACCIA MINISTERIALE

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.

Il candidato consideri inoltre il seguente scenario: 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.: audioguida, 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 1 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.

Il tema dell'Esame di Stato proposto nella sessione ordinaria di quest'anno è suddiviso in due sezioni distinte. Nella prima parte viene richiesto al candidato di sviluppare un pic-

colo tema riguardante l'ambito della sicurezza informatica. In particolare lo studente è invitato a evidenziare possibili rischi e relative contromisure per un sistema informatico o una rete.

Gli attacchi informatici possono essere di svariata natura. In particolare vi sono attacchi che mirano ad interrompere un servizio, oppure attacchi diretti alla sottrazione o alla cancellazione dei dati. Per quel che riguarda gli attacchi di rete in particolare, è possibile dover affrontare situazioni dove viene effettuato un accesso non autorizzato utilizzando bug presenti nei programmi oppure direttamente attraverso attacchi alle password degli utenti. Inoltre un sistema connesso in rete può essere soggetto di attacchi da parte di malware, virus e spyware in particolare, che mirano a distruggere dati, ad insediarsi nel computer per poi propagare l'infezione informatica. Altra tipologia di attacchi spesso portata avanti da macchine infette è quella dei Denial of Service, ove la macchina attaccata viene sovraccaricata di richieste fino a non poter più rispondere alle richieste degli utenti regolari. Le contromisure per queste classi di attacchi sono il mantenimento di una politica di gestione delle password sia per quel che riguarda la complessità che per quel che riguarda l'aging, ovvero il tempo che intercorre tra un cambio password ed il successivo, che va effettuato regolarmente. Una segmentazione rigida degli utenti in gruppi in relazione ai privilegi necessari per svolgere le loro mansioni nel sistema, secondo il principio per cui ogni utente ha i privilegi minimi necessari per il suo lavoro e l'eventuale implementazione sui sistemi di crittografia, consente di mitigare i rischi relativi agli attacchi alle password ed ai bug dei programmi. Efficienti strategie di backup dei dati possono essere una buona contromisura contro i virus di tipo distruttivo, e l'implementazione di firewall con regole stringenti consente il monitoraggio della rete del sistema alla ricerca di eventuali virus e spyware e la difesa del perimetro della rete da attacchi provenienti dall'esterno.

Per quel che riguarda la seconda parte del tema proposto, viene richiesto di realizzare un sistema informativo per la gestione delle prenotazioni in una biglietteria online di un grande museo. Si tratta di un problema abbastanza classico, simile ad altri già proposti in anni precedenti, tuttavia è necessaria un'attenta fase di analisi iniziale per realizzare un sistema che sia consistente con le richieste funzionali per l'applicazione.

Prima di poter definire una struttura per la base dati è necessaria l'analisi dei requisiti funzionali. Il tema, come solitamente avviene, indica al candidato la libertà di formulare ipotesi aggiuntive per poter definire poi in maniera consistente e non ambigua il modello concettuale e logico dell'applicazione che sta sviluppando.

Analizzando le richieste del problema è necessario decidere come memorizzare le date visto che nelle query da realizzare

si dovranno selezionare biglietti ed esposizioni relativi ad un certo intervallo temporale.

Come scelta semplice è a mio avviso consigliabile utilizzare una codifica numerica a 8 cifre dove le prime 4 saranno l'anno ed a seguire due cifre per il mese e due per il giorno. Ad esempio la data 2/6/2013 sarà codificata nel sistema come 20130602. In questo modo si preserva l'ordinamento temporale e sarà sufficiente un semplice confronto con minore e maggiore uguale per selezionare intervalli di date.

Per quel che riguarda le riduzioni la traccia non indica chiaramente se per eventi o situazioni particolari siano possibili riduzioni di tipo diverso. Ad esempio non viene definito se una mostra A, una mostra B ed una emissione per l'ingresso al museo possano o meno condividere le riduzioni di prezzo biglietto. In questa proposta di risoluzione della traccia la scelta fatta è quella di avere riduzioni diverse per ogni evento (sia esso un ingresso al museo o un ingresso ad un evento particolare) al fine di mantenere il sistema il più possibile flessibile ed adattabile alle esigenze del committente. Inoltre per ogni ingresso sarà possibile avere una sola riduzione.

Altro punto per il quale dobbiamo formulare delle ipotesi aggiuntive è quello relativo ai prodotti e servizi accessori in quanto nella traccia non viene infatti specificato se essi siano disponibili su tutte le tipologie di biglietto oppure no. Ragionevolmente possiamo ipotizzare che ogni evento avrà le proprie tipologie di servizi e prodotti accessori.

Formulate queste ipotesi aggiuntive, è possibile analizzare il tema proposto per estrarre le informazioni utili relativamente alla definizione del modello concettuale. In questa tipologia di modello si focalizza l'attenzione sulle entità che compongono il sistema in esame. Come primo step il consiglio è sempre quello di cercare le entità reali che concorrono alla soluzione del problema, cioè gli oggetti che fisicamente sono coinvolti nel sistema che si sta modellizzando e successivamente aggiungere entità e relazioni funzioni rispetto all'applicazione. L'entità cuore del nostro sistema sarà il *biglietto*, che conterrà informazioni circa il codice identificativo che serve ad individuare l'evento a cui il biglietto si riferisce (mostra o visita singola) che sarà una stringa. Inoltre ci sarà un titolo per l'evento, la tariffa corrente e le date di inizio e fine, che saranno vuote per la visita singola. Una nota particolare merita l'informazione sulla tariffa corrente. Infatti questo campo potrà essere sovrascritto con il prezzo corrente nel caso di variazioni di prezzo per quel che riguarda la tariffa base e verrà copiato all'atto di creazione di un nuovo ingresso al museo (sia esso di base o per un evento) in un apposito campo dell'entità ingresso. In questo modo ogni volta nel-

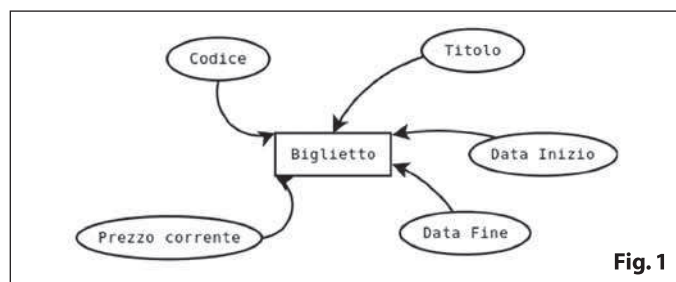


Fig. 1

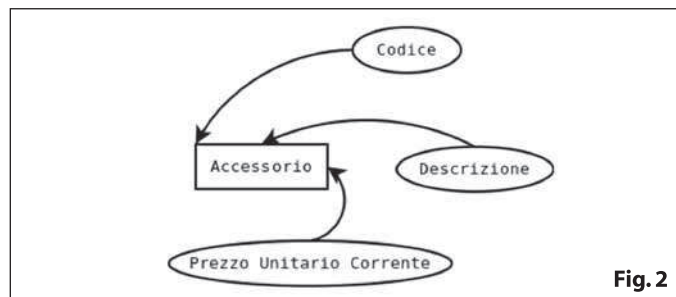


Fig. 2

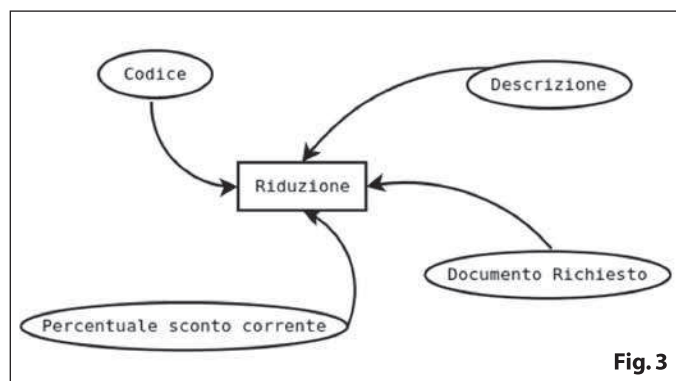


Fig. 3

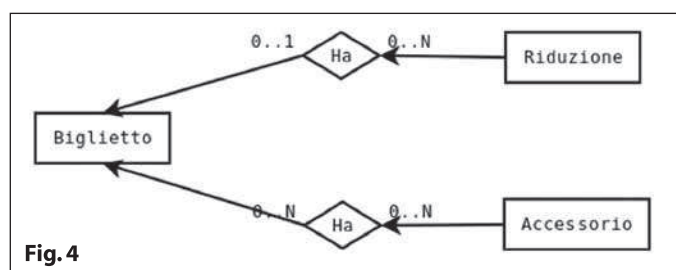


Fig. 4

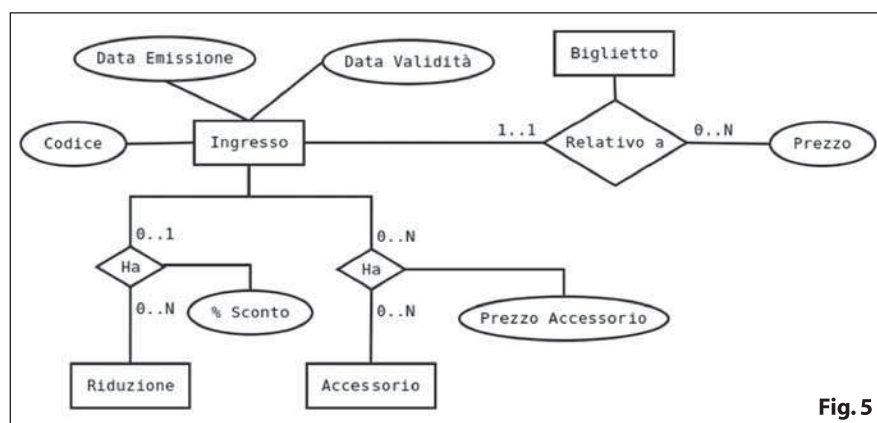


Fig. 5

l'entità *ingresso* avremo il prezzo pagato per quel particolare ingresso e da questa informazione potremo calcolare il ricavo per quell'ingresso. Non copiare il dato comporterebbe la perdita di consistenza dei dati nel caso di cambio del prezzo nell'entità *biglietto*, infatti portare ad esempio il prezzo del biglietto da 5 € a 7 € farebbe aumentare i ricavi di tutti gli ingressi precedenti. Sebbene non venga richiesto allo studente di gestire anche la parte di inserimento, mostrare di avere una visione d'insieme relativamente a quanto si sta realizzando è importante per mostrare competenze adeguate nella materia. L'entità *biglietto* avrà quindi la seguente struttura (Figura 1).

L'entità *accessorio* descrive, per ogni biglietto, quali siano gli accessori disponibili. Ogni accessorio è individuato da un codice, una descrizione ed un prezzo unitario corrente. La gestione del prezzo corrente è la medesima vista per il prezzo del biglietto (Figura 2).

L'entità *riduzione* contiene le informazioni relative al codice della riduzione in oggetto, alla descrizione della riduzione ed alla tipologia di documento richiesta, oltre alla percentuale di sconto. Anche in questo caso si tratta di sconto corrente che potrà essere modificato, e si deve tener presente questo fatto nella definizione dello schema logico della base dati che stiamo modellizzando (Figura 3).

Le relazioni tra biglietto e accessorio e tra biglietto e riduzione sono senza attributi aggiuntivi e descrivono quali accessori e riduzioni sono eventualmente disponibili per il biglietto considerato. Per questa parte della base dati lo schema concettuale è il seguente (Figura 4).

L'ultima entità da modellizzare è l'ingresso, ovvero quel dato che descrive un visitatore del museo, sia che esso acceda per una visita singola, sia che lo faccia per un evento. Le informazioni contenute sono il riferimento al biglietto, la data di validità, quella di emissione, un codice identificativo, e relazioni con i dati dell'evento relativo, oltre ad eventuali riduzioni ed accessori acquistati. Il codice identificativo del biglietto è un numero progressivo con i dati relativi al prezzo del biglietto, degli eventuali accessori e dell'eventuale riduzione (Figura 5).

Progettazione della base dati

Realizzato il modello concettuale ora possiamo concentrarci sulla creazione della base dati. La traduzione del modello concettuale in quello logico e nella conseguente creazione del codice

ESAMI CONCLUSIVI

SQL per le tabelle che faranno parte della base dati è un'operazione abbastanza schematica. Entità e relazioni definite nel modello concettuale individuano le tabelle della base dati, e per le relazioni si analizza se esse debbano essere descritte da tabelle aggiuntive (nel caso di relazioni molti a molti) oppure possano essere inglobate in altre tabelle.

Nella creazione del modello logico partiamo da quanto ottenuto dal modello concettuale, ma introduciamo ulteriori informazioni quali ad esempio il tipo di dato (numerico, stringa) relativo ad ogni attributo ed eventuali attributi aggiuntivi che serviranno nella base dati per individuare univocamente una riga (chiavi primarie) nelle tabelle o effettuare il riferimento ad una riga di una tabella esterna (chiavi esterne) nelle relazioni ove serva.

Inoltre dovremo tenere conto che il modello concettuale deve essere normalizzato nella definizione del modello logico e delle corrispettive tabelle nel linguaggio SQL. In particolare le relazioni ingresso/accessorio, biglietto/accessorio e biglietto/riduzione hanno bisogno di una tabella intermedia, essendo relazioni di tipo molti a molti.

Per l'entità riduzione avremo il seguente modello logico:

Tabella	<i>riduzioni</i>	
Campo	Tipo dato	vincoli
codice	VARCHAR(20)	PRIMARY KEY NOT NULL
descrizione	VARCHAR(120)	NOT NULL
tipo_documento	VARCHAR(120)	NOT NULL
p_sconto_attuale	REAL	NOT NULL

Tutte le colonne di questa tabella sono non nulle ed inoltre il codice deve definire in modo univoco la riduzione.

L'entità accessorio verrà modellizzata dalla tabella accessori con la seguente definizione:

Tabella	<i>accessori</i>	
Campo	Tipo dato	vincoli
codice	VARCHAR(20)	PRIMARY KEY NOT NULL
descrizione	VARCHAR(120)	NOT NULL
prezzo_attuale	REAL	NOT NULL

Per l'entità biglietto la descrizione logica sarà:

Tabella	<i>biglietti</i>	
Campo	Tipo dato	vincoli
codice	VARCHAR(20)	PRIMARY KEY NOT NULL
titolo	VARCHAR(120)	NOT NULL
data_inizio	INTEGER	
data_fine	INTEGER	
prezzo_attuale	REAL	NOT NULL

Come si può notare il codice rappresenta la chiave primaria per questa tabella, e i campi data_inizio e data_fine possono anche essere nulli, cosa che accade per la riga relativa ad un biglietto di base. Restano da descrivere a livello logico le due relazioni biglietto/riduzioni e biglietto/accessori. Essendo entrambe relazioni di tipo molti a molti, la rappresentazione normalizzata prevede una tabella intermedia due colonne che faranno riferimento alle chiavi primarie delle tabelle che mettono in relazione. Ovvero:

Tabella	<i>biglietti_accessori</i>	
Campo	Tipo dato	vincoli
codice_biglietto	VARCHAR(20)	NOT NULL
codice_accessorio	VARCHAR(20)	NOT NULL

Nel codice del programma o come vincolo aggiuntivo nel database si dovrà aggiungere che l'accoppiata (codice_biglietto, codice_accessorio) debba essere unica. Discorso simile per la relazione molti a molti biglietti/riduzioni che verrà modellizzata nella seguente tabella:

Tabella	<i>biglietti_riduzioni</i>	
Campo	Tipo dato	vincoli
codice_biglietto	VARCHAR(20)	NOT NULL
codice_riduzione	VARCHAR(20)	NOT NULL

Con il vincolo che la coppia (codice_biglietto, codice_riduzione) sia unica all'interno della tabella.

Per quel che riguarda l'entità ingresso serve tenere presente che la relazione sia di tipo uno ad uno con la tabella biglietti

e quindi è possibile inserire nella tabella ingressi il riferimento al biglietto, tramite il codice.

Tabella	ingressi	
Campo	Tipo dato	vincoli
codice	VARCHAR(20)	PRIMARY KEY NOT NULL
data_validita	INTEGER	NOT NULL
data_emissione	INTEGER	NOT NULL
codice_biglietto	VARCHAR(20)	NOT NULL
prezzo	REAL	NOT NULL
codice_riduzione	VARCHAR(20)	
p_riduzione	REAL	

Accanto a questa tabella ne serve una per definire la relazione molti a molti con la tabella degli accessori:

Tabella	ingressi_accessori	
Campo	Tipo dato	vincoli
codice_accessorio	VARCHAR(20)	NOT NULL
codice_ingresso	VARCHAR(20)	NOT NULL
prezzo	REAL	NOT NULL

Questa tabella ha il vincolo aggiuntivo che la coppia (codice_accessorio, codice_ingresso) deve essere unica nella tabella.

La creazione delle tabelle viene effettuata mediante il linguaggio SQL. Per le tabelle biglietti e ingressi il codice di creazione sarà il seguente:

```
CREATE TABLE biglietti (
  codice          VARCHAR(20)    NOT NULL PRIMARY KEY,
  titolo          VARCHAR 120    NOT NULL,
  data_inizio     INTEGER ,
  data_fine       INTEGER ,
  prezzo_attuale REAL    , NOT NULL
)
```

```
CREATE TABLE ingressi (
  codice          VARCHAR(20)    NOT NULL
                  PRIMARY KEY,
  data_validita   INTEGER NOT NULL,
```

```
  codice_biglietto VARCHAR(20)
                  REFERENCES(biglietti.codice) NOT
                  NULL ,
  prezzo          REAL NOT NULL,
  codice_riduzione VARCHAR(20)
                  REFERENCES(riduzioni.codice) NULL,
  p_riduzione     REAL DEFAULT 0,
)
```

Per quel che riguarda le query per l'estrazione dei dati dal database creato procediamo per passi. Per prima cosa viene chiesto di «elencare i codici dei biglietti emessi per un determinato giorno». Avendo definito la codifica delle date tra le ipotesi aggiuntive questa query diventa facile. Infatti volendo estrarre tutti i codici dei biglietti emessi per il 7 agosto 2013, basta tenere conto che questa data viene codificata nell'intero 20130807 per cui la query diventa:

```
SELECT
  codice
FROM
  ingressi
WHERE
  data_emissione = 20130807
```

La seconda query richiesta è «elencare i titoli e le date delle esposizioni tematiche che si sono tenute nel periodo 1 gennaio – 31 dicembre di un determinato anno». Per ricavare titoli e date delle esposizioni tematiche ad esempio del 2013 è sufficiente dalla tabella biglietti selezionare le righe che hanno una data di inizio maggiore del 20130101 e fine inferiore a 20131231 e poi estrarre solo le tre colonne che servono.

```
SELECT
  titolo, data_inizio, data_fine
FROM
  biglietti
WHERE
  (data_inizio >= 20130101) AND (data_fine <= 20131231)
  AND
```

La terza query è «indicare il numero di biglietti emessi per una determinata esposizione».

Per ricavare gli ingressi di un'esposizione basta estrarre le righe che hanno il codice corrispondente alla mostra in questione, per contare quanti sono basta usare la funzione COUNT(). Per ricavare quelli della mostra 'MOS0101' si può procedere così:

```
SELECT
  COUNT(*)
FROM ingressi
WHERE codice_biglietto = 'MOS0101'
```

ESAMI CONCLUSIVI

La quarta ed ultima query è «calcolare il ricavato della vendita dei biglietti di una determinata esposizione». Questa è la richiesta più complessa del compito, e deve tenere conto sia dell'eventuale riduzione che degli eventuali accessori richiesti per ogni singolo ingresso.

```
SELECT
SUM(i.prezzo- i.prezzo*COALESCE(p_riduzione,0)/100.0+
 ( SELECT COALESCE(SUM(ia.prezzo),0)
 FROM
 ingressi_accessori ia
 WHERE i.codice=ia.codice_ingresso))
FROM
 ingressi i
WHERE
 i.codice_biglietto = 'MOS0101'
GROUP BY i.codice
```

Per quel che riguarda la progettazione di una porzione del sito del museo, il tema non indica se occorra sviluppare una parte dinamica oppure una parte statica. Di base è necessario ipotizzare una struttura generale del sito del museo, che potrebbe contenere sia una gallery di immagini, che l'elenco degli eventi e l'evento corrente in evidenza. Volendo presentare il codice relativo all'elenco degli eventi attivi utilizziamo un approccio MVC (Model View Controller). Il Model sarà un oggetto che esporrà un metodo che ritorni l'elenco degli eventi attivi (righe della tabella biglietti con data di inizio minore di quella attuale e data di fine maggiore). Per cui la query richiesta sarà:

```
SELECT * from biglietti
WHERE data_inizio <= <OGGI> AND data_fine >= <OGGI>
```

Questa lista verrà poi passata ad un template HTML in una

variabile che possiamo chiamare “eventi”. Il codice che mostrerà l'elenco potrebbe essere il seguente:

```
<html>
<head>
</head>
<body>
<h1>Elenco Eventi</h1>
<ul>
{% for evento in eventi %}
<li>{{evento.titolo}}- dal {{evento.data_inizio}} al {{evento.data_fine}}
(costo:{{evento.prezzo_attuale}}€)</li>
{% endfor %}
</ul>
</body>
</html>
```

Il tema proposto quest'anno non presenta particolari diversità rispetto a quelli proposti negli anni precedenti. La presenza di un quesito iniziale pone l'attenzione al fatto che l'informatica sia una disciplina non legata solo al fare quanto piuttosto anche all'accrescere il bagaglio culturale e teorico degli studenti. L'argomento teorico che viene chiesto di sviluppare infatti è molto vicino al mondo informatico vissuto dallo studente e va a testare quelle competenze che lo possano rendere un utilizzatore consapevole degli strumenti informatici di uso quotidiano.

Analizzando le richieste della seconda parte del quesito non vi sono domande particolarmente complesse se non l'ultima query. Dal punto di vista della modellizzazione della base dati invece va notato come venga richiesta una buona dimestichezza con le relazioni molti a molti con attributi aggiuntivi.

*Andrea Pollini
Università Cattolica, sede di Brescia*