

Informatica

Andrea Pollini

Il tema d'esame propone la realizzazione di un sistema informatico per la gestione delle Olimpiadi dell'Informatica. Questa competizione prevede un sistema di gare in più fasi, dove gli atleti si scontrano raggruppati in squadre e singolarmente. Le richieste in termini di funzionalità ci devono servire per poter capire quali aspetti del problema reale il nostro sistema debba catturare e gestire. La base dati che ci accingiamo a realizzare sarà infatti un *modello* della realtà e si baserà su delle convenzioni che andremo via via a definire. Non dobbiamo effettuare una mappatura della realtà all'interno del nostro sistema, quanto piuttosto dovremo creare uno schema della realtà che permetta di soddisfare le richieste.

Progettazione dello schema concettuale

L'obiettivo della progettazione dello schema concettuale è quello di descrivere il modello della realtà che rappresenta il nostro dominio applicativo, individuando gli oggetti, siano essi concreti o astratti, che sono di pertinenza per il sistema informativo che stiamo progettando e le eventuali relazioni che li legano. Ad ogni entità vengono associati uno o più at-

tributi, che vanno a definirne le caratteristiche. Ogni attributo ha un proprio dominio, ovvero un insieme di valori che può assumere, ad esempio stringhe o valori numerici. Ad una entità, oltre ai suoi attributi, verranno anche assegnati dei vincoli. Un vincolo non è altro che una regola che l'entità deve soddisfare nel mondo reale.

Il passo successivo consiste nel redigere l'elenco delle dipendenze logiche, dette relazioni, di interesse per il nostro sistema informatico. Ogni relazione collega 2 entità e viene definita da un nome, che indica la funzione logica della relazione. Inoltre ogni entità possiede una molteplicità, che indica la cardinalità della relazione. Opzionalmente si possono aggiungere ulteriori attributi alla relazione, per specificarne le caratteristiche.

La realtà di cui stiamo creando un modello è composta da **2 entità ed una relazione**. Le entità saranno atleti e gare, mentre i risultati ottenuti da un atleta in una gara definiscono una relazione tra le due entità appena definite. Vediamo ora di definire attributi, domini e vincoli.

Ogni atleta, al minimo, può essere individuato mediante nome e cognome, tuttavia per soddisfare le richieste del testo ogni atleta deve possedere anche gli attributi aggiuntivi ri-

LA TRACCIA MINISTERIALE

Indirizzo Informatica

TEMA DI INFORMATICA GENERALE E APPLICAZIONI TECNICO-SCIENTIFICHE

(Testo valevole per i corsi di ordinamento e per i corsi sperimentali del progetto «SIRIO» - Informatica)

In occasioni delle Olimpiadi Internazionali di Informatica 2008, la società organizzatrice desidera realizzare un sistema informatico per la gestione delle gare e degli "atleti".

La base di dati deve consentire la memorizzazione delle informazioni

- degli atleti, che possono partecipare alle gare sia singolarmente sia raggruppati in squadre
- delle gare nelle varie fasi
- delle sedi di gara

Le Olimpiadi prevedono una fase scolastica (in ciascun istituto scolastico partecipante), una fase regionale, una finale nazionale e la gara internazionale che designerà il vincitore e la squadra vincitrice.

Il candidato, fatte le opportune ipotesi aggiuntive, realizzi:

1. un'analisi della realtà di riferimento individuando le possibili soluzioni e scelga quella che a suo motivato giudizio è la più idonea a rispondere alle specifiche indicate
2. uno schema concettuale della base di dati

3. uno schema logico della base di dati

4. la definizione delle relazioni della base di dati in linguaggio SQL

5. le seguenti interrogazioni espresse in linguaggio SQL:

- stampare l'elenco degli atleti raggruppati per squadre per ogni singola fase
- dato il nome di un atleta stampare i risultati ottenuti nelle diverse gare alle quali ha partecipato
- stampare il calendario delle gare
- stampare una scheda informativa (cognome, nome, istituto scolastico di provenienza, nazionalità) del vincitore e della squadra vincitrice
- stampare la classifica per ciascuna gara (a parità di punteggio vengono privilegiati gli atleti più giovani)
- aggiornare, per ciascuna fase (scolastica-regionale-nazionale-internazionale) gli eventuali punteggi record
- calcolare il punteggio medio ottenuto durante la prima selezione, per ciascun istituto scolastico
- stampare per ciascuna squadra il numero di "atleti" partecipanti e l'età media

6. l'interfaccia utente che il candidato intende proporre per interagire con la base di dati e codificare in un linguaggio di programmazione a scelta un segmento significativo del progetto realizzato

7. un sito Internet che presenti al pubblico le classifiche delle diverse gare.

guardanti l'istituto scolastico di provenienza e la nazionalità. Inoltre, vista la richiesta di creare le classifiche delle gare tenendo conto anche dell'età, ad ogni atleta deve essere associato anche un attributo che ne indichi l'età. Assumendo che questo archivio debba contenere solo i dati relativi alle Olimpiadi del 2008, è sufficiente che l'attributo età sia un intero positivo. Definiamo ora l'entità deputata alla definizione delle gare. Ogni gara deve possedere un attributo che indichi la data, il luogo in cui viene svolta a la fase di gara relativa. Quest'ultimo attributo ha come dominio una stringa corrispondente alle varie fasi di gara: scolastica, regionale, nazionale o internazionale.

Per ogni gara è necessario inoltre memorizzare per ogni giocatore i punteggi e la squadra di appartenenza. In prima battuta si potrebbe pensare di inserire l'informazione relativa alla squadra direttamente all'interno dell'entità atleti. Tuttavia possiamo verosimilmente presupporre che il raggruppamento in squadre sia diverso in base alla gara ed alla fase. A livello di istituto potremmo avere infatti squadre definite dalle classi di appartenenza, mentre a livello nazionale le squadre potrebbero essere le province di origine.

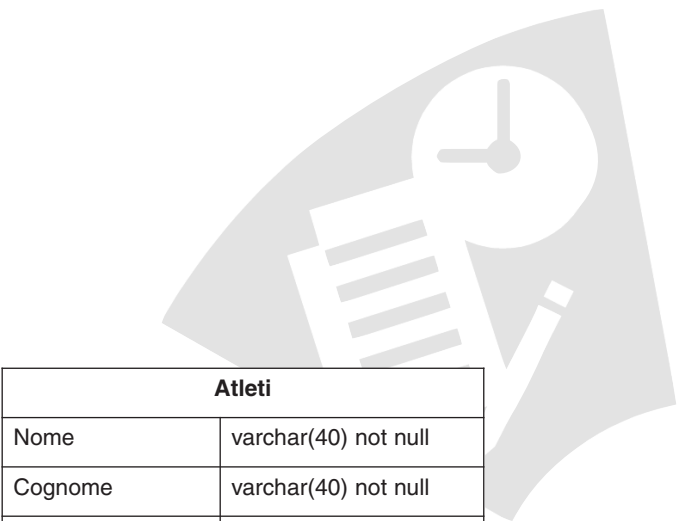
Per risolvere questi problemi definiamo una relazione tra le entità *atleti* e *gare*. Tale relazione, *RisultatiGare*, deve contenere anche tutti gli altri dati aggiuntivi appena individuati. La cardinalità della relazione è N-N, ovvero un atleta può partecipare a più gare, e ad ogni gara partecipano più atleti.

Costruzione dello schema logico

Partendo dallo schema concettuale appena definito procediamo ora alla creazione dello schema logico analizzando quali siano le tabelle da definire per descrivere lo schema concettuale in termini della base dati che stiamo costruendo. L'operazione di creazione dello schema logico avviene in più passi:

1. Ogni entità diventa una tabella dello schema logico. Ogni attributo dello schema concettuale diventa un attributo dello schema logico e il dominio ne individua il tipo. Vengono considerati ed inglobati eventuali vincoli.
2. Per ogni tabella viene definito un attributo o un insieme di attributi che possa identificare univocamente una singola riga. Se esso non esiste si aggiunge un attributo artificiale allo scopo, spesso un campo numerico univoco. Tale attributo è la chiave primaria.
3. Le relazioni diventano anche esse entità se come nel nostro caso sono relazioni di tipo N-N. Vengono definiti attributi e i riferimenti alle entità che fanno parte della relazione diventeranno dei vincoli di chiave esterna per la tabella della relazione.

Ecco i dati che definiscono la tabella atleti. Visto che nome e cognome non definiscono univocamente un atleta per la presenza di possibili omonimie, aggiungiamo un attributo artificiale *id* contenente un intero unico che individui il singolo atleta.



Atleti	
Nome	varchar(40) not null
Cognome	varchar(40) not null
Istituto	varchar(120) not null
Nazione	varchar(40) not null
Eta	integer not null
Id	integer not null
primary_key(id)	

La tabella gare descrive il modello concettuale dell'entità gare. Anche in questo caso è necessario aggiungere una chiave primaria rappresentata da un attributo artificiale (*id*) che avrà come dominio gli interi.

Gare	
Luogo	varchar(40) not null
Data	date not null
Fase	varchar(40)
Id	integer not null
primary_key(id)	

Per gestire la relazione tra *atleti* e *gare* definiamo la tabella *risultati_gare*. Tale tabella conterrà un riferimento ad entrambe le entità che formano la relazione, ovvero *atleti* e *gare*. Come

nomi per i campi della tabella *risultati_gare* utilizzeremo *gara_id* per la relazione con la tabella gare e *atleta_id* per la relazione con la tabella atleti. Tali campi saranno chiavi esterne per la tabella *risultati_gare*. Gli altri campi faranno riferimento agli attributi dell'entità *risultati_gare* prima descritti. Per la definizione della chiave primaria in questo caso basta considerare la coppia *<atleta_id,gara_id>*.

Risultati_gare	
atleta_id	integer not null
gara_id	integer not null
Squadra	varchar(40) not null
Punteggio	integer not null
primary_key(atleta_id,gara_id), foreign_key (gara_id) references gare(id), foreign_key(atleta_id) references atleti(id)	

Definizione in linguaggio SQL

Completata la fase dell'analisi concettuale possiamo agevolmente definire la struttura della nostra base dati utilizzando il linguaggio SQL. Come prima operazione creiamo il database con nome *olimpiadi*. Dopo di che definiamo per ogni tabella del modello concettuale il corrispondente comando di creazione in SQL.

```
CREATE DATABASE olimpiadi;

CREATE TABLE atleti (
  id integer NOT NULL,
  nome varchar(40) NOT NULL,
  cognome varchar(40) NOT NULL,
  istituto varchar(120) NOT NULL,
  nazione varchar(40) NOT NULL,
  eta integer NOT NULL,
  CONSTRAINT atleti_pk PRIMARY KEY(id)
);

CREATE TABLE gare (
  id integer NOT NULL,
  luogo varchar(120) NOT NULL,
  data date NOT NULL,
  fase varchar(40) NOT NULL,
  CONSTRAINT gare_pk PRIMARY KEY(id)
);

CREATE TABLE risultati_gare (
  atleta_id integer NOT NULL,
  gara_id integer NOT NULL,
  squadra varchar(40) NOT NULL,
  punteggio integer NOT NULL,
  CONSTRAINT risultati_gare_pk PRIMARY KEY (atleta_id,gara_id),
  FOREIGN KEY (gara_id) REFERENCES gare(id),
  FOREIGN KEY (atleta_id) REFERENCES atleti(id)
);
```

Costruzione delle interrogazioni SQL

Definita la nostra base dati possiamo procedere a costruire le query SQL che rispondono alle varie richieste del testo. Per ogni query è consigliabile procedere in maniera sistematica, cercando di affrontarla per gradi. Nel caso in cui vengano richiesti raggruppamenti o aggregati dei dati, come prima operazione si può produrre una versione preliminare della query che non effettui alcuna operazione, se non quella di stampare tutti i dati necessari per i raggruppamenti o le operazioni eventualmente richieste. Queste operazioni le andremo a realizzare nella versione definitiva, ottenuta modificando quella preliminare.

La prima richiesta è quella di «stampare l'elenco degli atleti raggruppati per squadre per ogni singola fase». La versione preliminare della query deve mostrare l'elenco delle quaterne nome,cognome,squadra e fase. Nome e cognome sono contenuti nella tabella atleti. La relazione *risultati_gare* ci consente di realizzare la versione preliminare della nostra query in quanto per ogni gara, ogni atleta farà parte di una squadra e questo ci consente di associare correttamente i dati utilizzando le chiavi esterne *atleta_id*, che collega i risultati all'atleta e la chiave *gara_id* che collega i risultati alla gara

in cui si sono svolti, consentendoci di recuperare la fase di gara. La query preliminare quindi è la seguente

```
SELECT
  atleti.nome,
  atleti.cognome,
  gare.fase,
  risultati_gare.squadra
FROM
  risultati_gare,gare,atleti
WHERE
  risultati_gare.atleta_id = atleti.id
AND
  risultati_gare.gara_id = gare.id
```

Ora possiamo ordinare i dati per squadra e per fase. Per fare questo dobbiamo aggiungere una clausola di tipo ORDER BY alla query che ordini i record per fase e nome squadra. La query in versione definitiva sarà quindi la seguente

```
SELECT
  risultati_gare.squadra,
  gare.fase,
  atleti.cognome,
  atleti.nome
FROM
  risultati_gare,gare,atleti
WHERE
  risultati_gare.atleta_id = atleti.id
AND
  risultati_gare.gara_id = gare.id
ORDER BY gare.fase,risultati_gare.squadra
```

La seconda richiesta è: «dato il nome di un atleta stampare i risultati ottenuti nelle diverse gare alle quali ha partecipato». La versione preliminare di questa query deve stampare a video luogo e data delle gare degli atleti, con i rispettivi punteggi. Per soddisfare la richiesta dobbiamo limitare i risultati a quelli di un singolo atleta, individuato dal nome e cognome. Per questo la query completa è la seguente:

```
SELECT
  risultati_gare.punteggio,
  gare.fase,
  gare.luogo,
  gare.data
FROM
  risultati_gare,gare,atleti
WHERE
  risultati_gare.atleta_id = atleti.id
AND
  risultati_gare.gara_id = gare.id
AND
  atleti.nome=[nome]
AND
  atleti.cognome = [cognome]
dove le diciture [nome] e [cognome] indicano i parametri.
```

La successiva richiesta è «stampare il calendario delle gare». Non necessitando di manipolare i dati in alcun modo, la versione definitiva e quella preliminare della query coincidono. L'unica accortezza è quella di aggiungere una clausola ORDER BY ... ASC per ordinare le date delle gare in ordine cronologico. Otteniamo quindi la query seguente (vd. a destra):

```
SELECT
  data,
  luogo
FROM
  gare
ORDER BY gare.data ASC
```

La quarta richiesta è «stampare una scheda informativa (cognome, nome, istituto scolastico di provenienza, nazionalità) del vincitore e della squadra vincitrice». Questa richiesta viene soddisfatta da due query distinte. La prima va ad estrarre, nella versione preliminare, i dati degli atleti e i punteggi ottenuti nella fase di gara 'INTERNAZIONALE'. Per cui

```
SELECT
  atleti.cognome,
  atleti.nome,
  atleti.istituto,
  atleti.nazione,
  risultati_gare.punteggio
FROM
  gare,risultati_gare,atleti
WHERE
  risultati_gare.atleta_id = atleti.id
  AND
  risultati_gare.gara_id = gare.id
  AND
  gare.fase = 'INTERNAZIONALE'
ORDER BY punteggio DESC
LIMIT 1
```

stampa l'atleta con il risultato più alto. Nel caso in cui vi siano più atleti con lo stesso punteggio potremmo agire come nella richiesta al caso successivo, dove a parità di punteggio viene privilegiato l'atleta con l'età minore. Per cui la query diventa

```
SELECT
  atleti.cognome,
  atleti.nome,
  atleti.eta,
  gare.luogo,
  gare.data,
  risultati_gare.punteggio
FROM
  gare,risultati_gare,atleti
WHERE
  risultati_gare.atleta_id = atleti.id
  AND
  risultati_gare.gara_id = gare.id
ORDER BY gare.data ASC, risultati_gare.punteggio DESC,atleti.eta ASC
```

```
SELECT
  atleti.cognome,
  atleti.nome,
  atleti.istituto,
  atleti.nazione,
  risultati_gare.punteggio
FROM
  gare,risultati_gare,atleti
WHERE
  risultati_gare.atleta_id = atleti.id
  AND
  risultati_gare.gara_id = gare.id
  AND
  gare.fase = 'INTERNAZIONALE'
ORDER BY punteggio DESC, eta ASC
LIMIT 1
```

Per recuperare invece la squadra con il punteggio maggiore dobbiamo nella versione preliminare mostrare a video tutti i punteggi degli elementi delle squadre. Inserendo i raggruppamenti per squadra e la somma dei punteggi otteniamo la query definitiva, che ritorna solo la squadra col punteggio maggiore grazie alla clausola LIMIT 1 e all'ordinamento per somma di punti

```
SELECT
  risultati_gare.squadra,
  SUM(risultati_gare.punteggio)
FROM
  gare,risultati_gare
WHERE
  risultati_gare.gara_id = gare.id
  AND
  gare.fase = 'INTERNAZIONALE'
GROUP BY risultati_gare.squadra
ORDER BY SUM(risultati_gare.punteggio) DESC
LIMIT 1
```

La richiesta successiva è «stampare la classifica per ciascuna gara (a parità di punteggio vengono privilegiati gli atleti più giovani)». Visto che non vengono menzionate le squadre risolviamo il quesito stampando solo la classifica degli atleti. Dobbiamo come prima cosa stampare a video i dati degli

atleti, i dati della gara e il punteggio ottenuto mediante una query preliminare, simile a quelle già viste. Le classifiche le possiamo ottenere semplicemente operando degli ordinamenti. Ordiniamo prima per data di gare, successivamente per punteggio e poi per età

Per rispondere al quesito «aggiornare, per ciascuna fase (scolastica-regionale-nazionale internazionale) gli eventuali punteggi record» possiamo definire una semplice query che potremmo utilizzare per creare una vista, che mantenga aggiornati i punteggi record. Raggruppando sul campo fase e prendendo il massimo del punteggio otteniamo quanto serve a soddisfare le richieste.

```
SELECT
  gare.fase,
  MAX(risultati_gare.punteggio)
FROM
  gare, risultati_gare
WHERE
  risultati_gare.gara_id = gare.id
GROUP BY gare.fase
```

Il quesito successivo chiede di «calcolare il punteggio medio ottenuto durante la prima selezione, per ciascun istituto scolastico». Per fare questo dobbiamo stampare per prima cosa l'elenco dei punteggi, l'istituto di appartenenza dello studente che li ha ottenuti e fare in modo che i risultati riguardino solo gare della fase di «ISTITUTO». Poi raggruppiamo per istituto e aggregiamo i punteggi utilizzando la funzione SQL AVG() che torna la media dei valori. La query definitiva sarà quindi la seguente

```
SELECT
  atleti.istituto,
  AVG(risultati_gare.punteggio)
FROM
  gare,risultati_gare,atleti
WHERE
  risultati_gare.atleta_id = atleti.id
  AND
  risultati_gare.gara_id = gare.id
  AND
  gare.fase = 'ISTITUTO'
GROUP BY atleti.istituto
```

esami conclusivi

L'ultimo quesito chiede di «stampare per ciascuna squadra il numero di “atleti” partecipanti e l'età media». Per far questo dobbiamo stampare di ogni squadra l'età degli atleti e il nome della squadra. Per ottenere le informazioni che ci servono raggruppiamo sul campo squadra, applichiamo la funzione AVG() al campo *età* e aggiungiamo un aggregato con la funziona COUNT() che conta le righe raggruppate.

```
SELECT
  AVG(atleti.eta),
  risultati_gare.squadra,
  COUNT(risultati_gare.squadra)
FROM
  gare,risultati_gare,atleti
WHERE
  risultati_gare.atleta_id = atleti.id
  AND
  risultati_gare.gara_id = gare.id
GROUP BY risultati_gare.squadra
```

Definite tutte le query che soddisfano le richieste del testo, ci viene ora richiesto di progettare l'interfaccia utente del nostro progetto. L'interfaccia utente deve consentire di gestire le 3 tabelle che abbiamo definito, permettendo di realizzare le operazioni di CRUD (create, read, update, delete). Per far questo dobbiamo definire 3 viste, una per la visualizzazione dei dati già inseriti e la cancellazione che potrebbe avere un layout a tabella. Una per la creazione e la modifica. Per rispondere all'ultimo quesito, ovvero la realizzazione di un sito web che consenta di mostrare le classifiche delle varie gare, possiamo avvalerci di uno qualsiasi dei numerosi linguaggi utilizzabili per la realizzazione della logica lato server. Si potrebbe realizzare una pagi-

na web contenente l'elenco delle gare, utilizzando la query realizzata in precedenza per la stampa del calendario. Ogni gara visualizzata possiede poi un link ad una pagina web che viene composta tramite un parametro indicante l'id della gara passato nell'URL e la classifica viene costruita formattando secondo le esigenze del sito il risultato della query già realizzata per la stampa delle classifiche, modificata per recuperare i dati relativi ad una singola gara

```
SELECT
  atleti.cognome,
  atleti.nome,
  atleti.eta,
  gare.luogo,
  gare.data,
  risultati_gare.punteggio
FROM
  gare,risultati_gare,atleti
WHERE
  risultati_gare.atleta_id = atleti.id
  AND
  risultati_gare.gara_id = gare.id
  AND
  risultati_gare.gara_id = [gara_id]
ORDER BY gare.data ASC, risultati_gare.punteggio DESC,atleti.eta ASC
```

La prova d'esame affrontata propone numerose questioni. Se da un lato la progettazione della base dati non presenta particolari difficoltà, la costruzione delle query che soddisfano le richieste necessitano di particolare attenzione. La proposta di realizzare le query in due step successivi consente di razionalizzare il processo rendendolo più chiaro. Il processo di analisi del testo assegnato risulta essenziale per procedere poi allo sviluppo della traccia. Tener conto delle varie richieste nella fase di progettazione concettuale consente poi di sviluppare i quesiti in maniera abbastanza lineare.

Andrea Pollini
Università Cattolica del Sacro Cuore, Brescia

IN RETE

Nel sito riservato agli abbonati, nella pagina ESAMI DI STATO, il dossier **13 anni di esami**: i testi delle prove degli esami di Stato dal 1996 al 2008 divisi per materia con le relative soluzioni.

Come raggiungerci:
nel sito **www.lascuola.it** cliccare su
Nuova Secondaria (sulla sinistra della home page)
– viene richiesta la registrazione per la quale è necessario il numero di abbonamento
– vengono poi forniti **nome utente e password** che consentono l'accesso.