

# **MASTER IN TECNOLOGIE INTERNET PROGETTO GESTIONE DATI & TRANSAZIONI**

RELATORE:  
ING. STEFANO GHELARDINI

# INDICE

MASTER IN TECNOLOGIE INTERNET PROGETTO GESTIONE DATI & TRANSAZIONI ....	1
INDICE .....	2
TESTO DEL PROBLEMA.....	3
La coppa del mondo di sci .....	3
DIAGRAMMA E-R.....	4
SCHEMA RELAZIONALE .....	4
COME FUNZIONA IL PROGRAMMA.....	5
Diagramma che mostra l'interfaccia tra utente e programma .....	5
Programma: come viene usato dall'utente .....	6
inserimento di un nuovo atleta .....	6
inserimento di una nuova gara .....	6
inserimento di una nuova specialità .....	7
LEGGI DATI .....	8
leggiAtleta.....	8
INSERIMENTO DATI.....	9
Inserimento dati atleta.....	9
Inserisci a quale gara ha partecipato un atleta e con quale risultato .....	10
MODIFICA DATI .....	12
nuoveSpecialitaGara .....	12
CANCELLA DATI.....	15
Cancella specialità.....	15
cancellaDatiAtleta.....	17

## TESTO DEL PROBLEMA

### ***La coppa del mondo di sci***

La coppa del mondo di sci alpino prevede lo svolgimento di una serie di gare in ciascuna delle seguenti specialità: slalom speciale, slalom gigante, supergigante e discesa libera.

Gli atleti, di cui si conoscono nome, nazionalità, sesso, e specialità praticate, partecipano alle gare ottenendo punti.

Ai primi quindici classificati di ogni gara viene assegnato un punteggio decrescente dal primo al quindicesimo classificato. La somma dei punti conquistati nelle gare di ciascuna specialità costituisce il punteggio di un atleta in quella specialità.

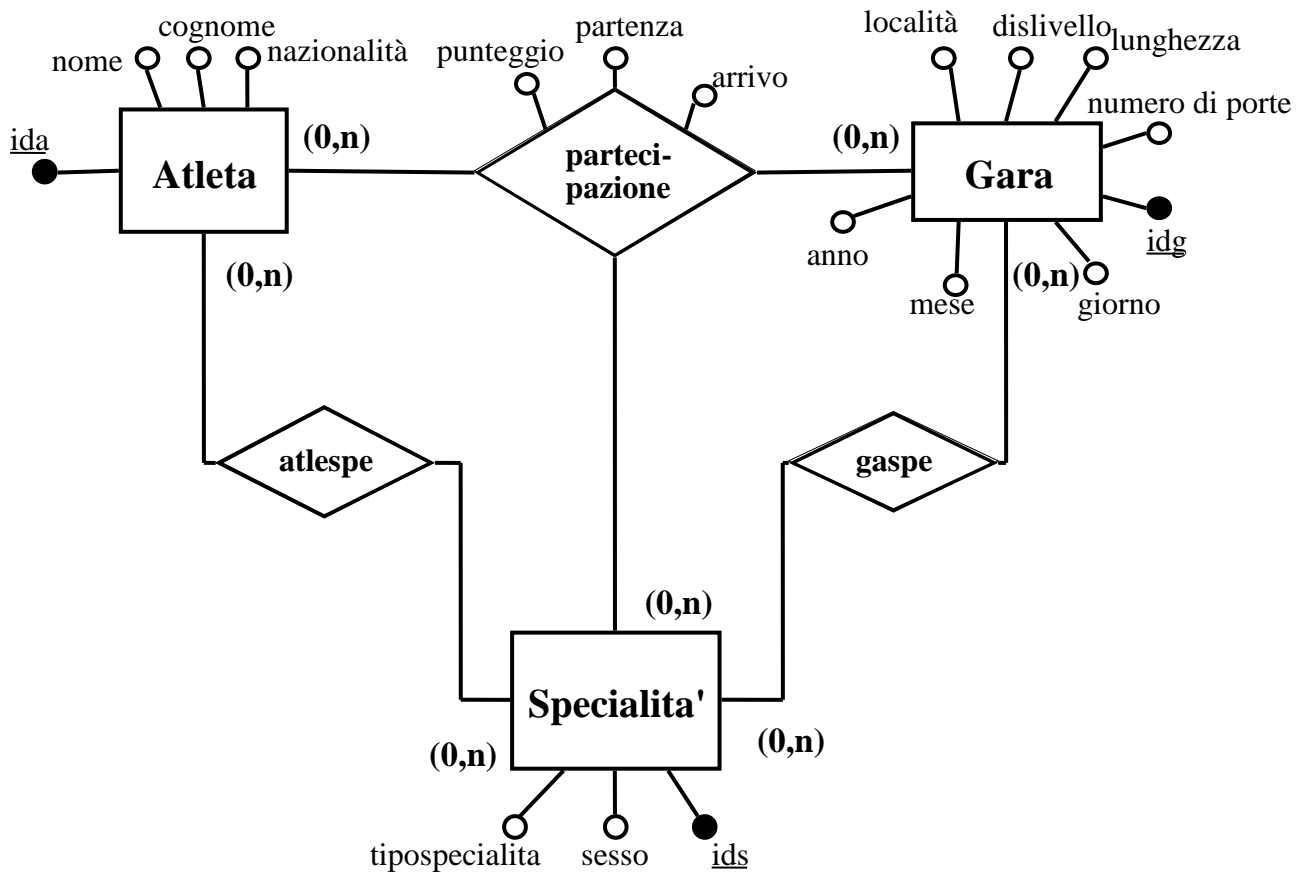
Di ogni gara si vogliono rappresentare il nome, la località, la data, la specialità, il sesso degli atleti impegnati, località, data in cui si svolge, dislivello di quota, lunghezza e numero di porte sul percorso. Inoltre, per ogni gara esiste un ordine di partenza ed uno di arrivo dei concorrenti.

Lo schema concettuale della base di dati precedentemente tratteggiato dovrà essere eventualmente completato con attributi ragionevoli per ciascuna entità, identificando le possibili chiavi, eventuali gerarchie ISA, oggetti composti, vincoli di integrità e le principali associazioni significative.

Si richiede una relazione scritta composta almeno dai seguenti elementi:

- la definizione del diagramma ER della realtà specificata;
- la definizione dello schema relazionale corrispondente, comprensivo di tutti i vincoli di integrità con particolare attenzione alla definizione di quelli relativi all'integrità referenziale;
- il codice PostgreSQL per la definizione dello schema della base di dati relazionale;
- il codice PostgreSQL per popolare (inserire) la base di dati relazionale con un numero iniziale di tuple sufficientemente significativo;
- un'applicazione (o più applicazioni) JDBC che realizzi (no) varie transazioni in sola lettura – interrogazioni - e in lettura-scrittura - modifica e inserzione - delle informazioni mantenute nella base di dati con le seguenti caratteristiche:
  - a) le interrogazioni dovranno, fra l'altro, contemplare l'uso delle seguenti tipologie di costrutti: giunzioni fra almeno tre tabelle, raggruppamenti, interrogazioni nidificate con almeno due livelli di nidificazione, aggregazioni;
  - b) alcune modifiche e inserzioni dapprima dovranno volontariamente violare qualche vincolo di integrità definito, quindi rilevare la eccezione automaticamente generata dal SGBD al momento del commit fallimentare, e quindi porre rimedio all'errore completando correttamente la transazione in lettura-scrittura

## DIAGRAMMA E-R



## SCHEMA RELAZIONALE

Atleta (ida, nome, cognome, nazionalità)

Gara(idg, località, dislivello, lunghezza, numero\_di\_porte, giorno, mese, anno)

Specialità(ids, sesso, tipospecialita)

atlespe (idatleta\*, idspecialita\*)

vincolo di integrità referenziale tra idatleta di atlespe e ida di Atleta

idspecialita di atlespe e ids di Specialita

gaspe (idspec\*, idgar\*)

vincolo di integrità referenziale tra: idspec di gaspe e ids di Specialita

idgar di gaspe e idg di Gara

partecipazione(idgar\*, idatl\*, idsp\*, punteggio, arrivo, partenza)

vincolo di integrità referenziale tra: idgar di partecipazione e idg di Gara

idatl di partecipazione e ida di Atleta ,

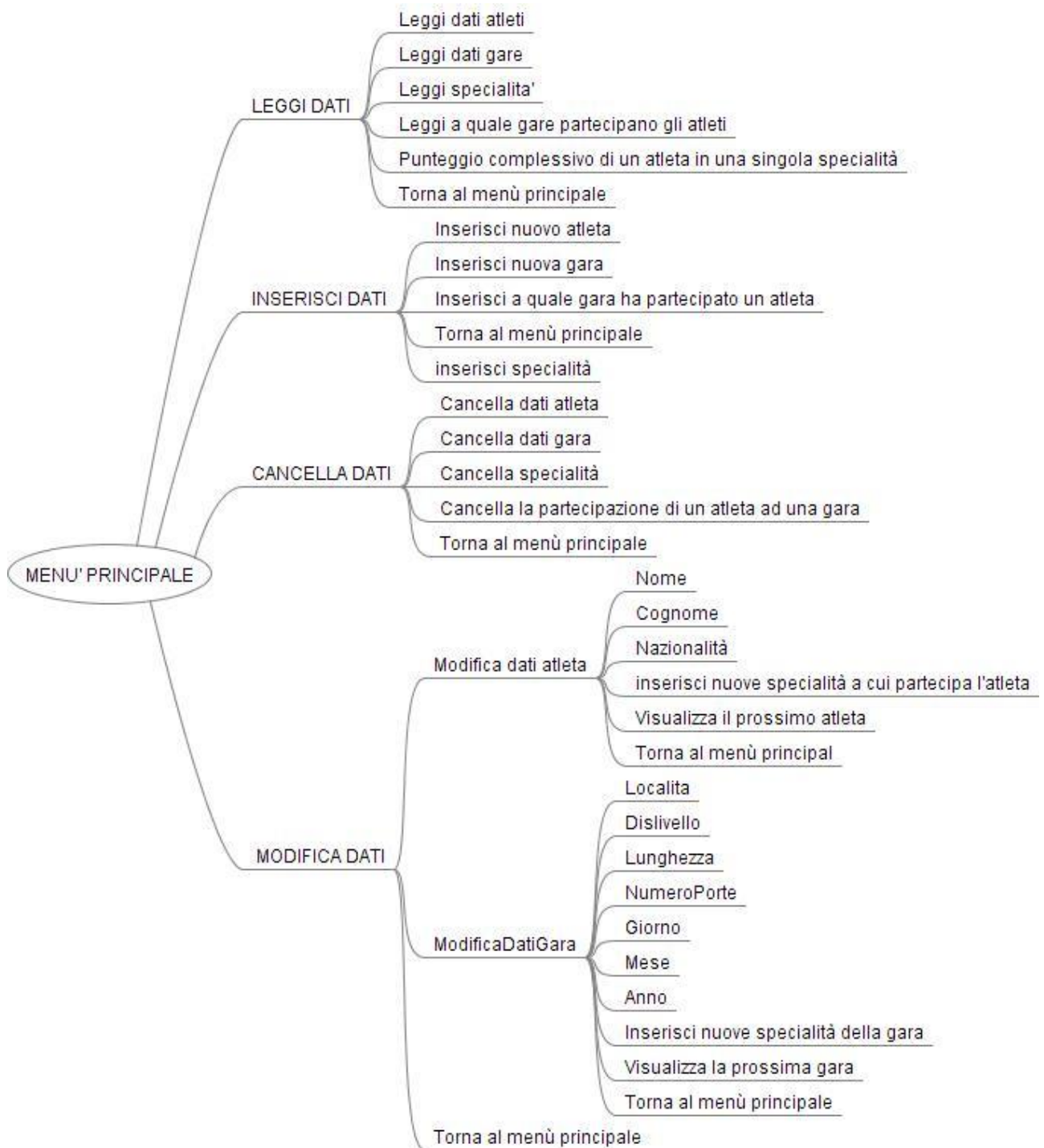
idsp di partecipazione e ids di Specialità

\_\_\_\_\_ chiave primaria

\* vincolo di integrità referenziale

# COME FUNZIONA IL PROGRAMMA

*Diagramma che mostra l'interfaccia tra utente e programma*



## ***Programma: come viene usato dall'utente***

Il programma opera nel seguente modo:

all'inizio l'utente acceda al menù principale, dal quale è possibile accedere a 4 sottomenù: leggi dati, inserisci dati, cancella dati, modifica dati (il menù principale ed i 4 sottomenù, sono realizzati in java con classi separate).

Da questi sottomenù è possibile chiamare i metodi java (contenuti nelle varie classi) che effettivamente operano sul database.

Quanto scritto vale per tutti i sottomenù tranne modifica dati in cui è necessario utilizzare un ulteriore livello che permette la selezione tra atleti, gara, specialità.

Vediamo, di seguito, alcuni scenari di utilizzo del programma:

### **inserimento di un nuovo atleta**

Vado nel menù principale → (la freccia indica il sottomenù che devo selezionare) inserimento dati  
→ inserisci dati atleta

- 1) scrivo i dati dell'atleta (inseriti nella tabella Atleta)
- 2) scrivo il sesso e le specialità praticate dall'atleta (inserite con una serie di t-uple, una per ogni specialità, nella tabella atlespe)

a questo punto i dati dell'atleta sono memorizzati, devo inserire i dati della gara a cui ha partecipato;

vado al menù principale → inserimento dati → Inserisci a quale gara ha partecipato un atleta  
e con quale risultato

- 1)seleziono una gara
- 2)inserisco i risultati ottenuti da quell'atleta in quella gara.

Visualizzo i risultati: menù principale → leggi dati → leggi a quale gare partecipano gli atleti

### **inserimento di una nuova gara**

Vado nel menù principale → inserimento dati → Inserisci nuova gara

- 1) scrivo i dati della gara
- 2) scrivo sesso e specialità della gara

a questo punto i dati della gara sono memorizzati, devo inserire i dati degli atleti che hanno partecipato a questa gara;

vado al menù principale → inserimento dati → Inserisci a quale gara ha partecipato un atleta  
e con quale risultato

- 1)seleziono un atleta
- 2)inserisco i risultati ottenuti da quell'atleta in quella gara.

Visualizzo i risultati: menù principale → leggi dati → leggi a quale gare partecipano gli atleti

## **inserimento di una nuova specialità**

Vado nel menù principale → inserimento dati → Inserisci nuova specialita'

1) scrivo il sesso della specialità e il suo nome

a questo punto i dati della nuova specialità sono memorizzati,  
quindi devo modificare i dati di atleti e gare per vedere se partecipano a questa specialità

torno al menù principale → Modifica dati → Modifica dati atleta  
adesso il programma visualizza i dati del primo atleta e il seguente menù:

*Dell'atleta <nome atleta> cosa vuoi modificare?*

*1) Nome*

*2) Cognome*

*3) Nazionalità*

*4) inserisci nuove specialità a cui partecipa l'atleta*

*5) Visualizza il prossimo atleta*

*6) Torna al menù principale*

seleziono inserisci nuove specialità a cui partecipa l'atleta e rispondo sì o no quando il programma chiede se l'atleta partecipa oppure no a quella specialità.

torno al menù principale → Modifica dati → Modifica dati gara  
ripeto le operazioni viste per gli atleti

vado al menù principale → inserimento dati → Inserisci a quale gara ha partecipato un atleta  
e con quale risultato

1)seleziono una gara

2)inserisco i risultati ottenuti da quell'atleta in quella gara per la nuova specialità

Di seguito vediamo come alcuni metodi del programma java interagiscono con il database

## LEGGI DATI

### *leggiAtleta*

1) leggi ogni t-upla della tabella atleta

ATLETA				
	nome character var	cognome character var	nazionalita character var	ida [PK] integer
1	carlo	bianchi	italiana	1
2	giuseppe	verdi	francese	2
3	lisa	falck	americana	3
4	helen	cruck	tedesca	4
5	bianca	livingstone	inglese	6
*				

2) con l'ida di atleta, seleziona le t-uple della tabella atlespe che hanno idatleta=ida

ATLESPE		
	idatleta integer	idspecialita integer
1	1	1
2	1	2
3	1	3
4	1	4
5	2	1
6	2	2
7	3	5
8	3	6
*		

3) con gli idspecialita della tabella atlespe, seleziona le t-uple della tabella specialita.  
Queste t-uple rappresentano le specialità dell'atleta e verranno visualizzate insieme al suo nome  
rappresentano

SPECIALITA'			
	sesso character(1)	tipospecialita character varying(30)	ids [PK] integer
1	M	slalom_speciale	1
2	M	slalom_gigante	2
3	M	supergigante	3
4	M	discesa_libera	4
5	F	slalom_speciale	5
6	F	slalom_gigante	6
7	F	supergigante	7
8	F	discesa_libera	8
*			



# INSERIMENTO DATI

## Inserimento dati atleta

1) inserimento di nome, cognome, nazionalità, ida, nella tabella atleta

ATLETA				
	nome character var	cognome character var	nazionalita character var	ida [PK] integer
1	carlo	bianchi	italiana	1
2	giuseppe	verdi	francese	2
3	lisa	falck	americana	3
4	helen	cruck	tedesca	4
5	bianca	livingstone	inglese	6
*				

```
insert into atleta (nome,cognome,nazionalita,ida )
values ( a.getNome(), a.getCognome(),
a.getNazionalita(), a.getIDa() )
a.getNome() rappresenta in java, il valore
dell'attributo nome della classe atleta
```

2) richiesta da parte del programma del sesso dell'atleta M (Maschile) ,F(Femminile);

3) supponiamo di rispondere sesso = M;

dalla tabella specialità sono selezionate tutte le t-uple in cui sesso è M.

SPECIALITA'			
	sesso character(1)	tipospecialita character varying(30)	ids [PK] integer
1	M	slalom_speciale	1
2	M	slalom_gigante	2
3	M	supergigante	3
4	M	discesa_libera	4
5	F	slalom_speciale	5
6	F	slalom_gigante	6
7	F	supergigante	7
8	F	discesa_libera	8
*			

```
select sesso,tipospecialita,ids
from specialita
```

4) a questo punto il programma chiede se l'atleta del punto 1) partecipa alle specialità del punto 3) ; se sì, l'ida dell'atleta e l'ids della specialità sono inserite nella tabella atlespe generando l'associazione tra atleta e specialità.

ATLESPE		
	idatleta integer	idspecialita integer
1	1	1
2	1	2
3	1	3
4	1	4
5	2	1
6	2	2
7	3	5
8	3	6
*		

```
insert into atlespe (idatleta,idspecialita)
values (a.getIDa(), sp.getIDs())
```

## Inserisci a quale gara ha partecipato un atleta e con quale risultato

1) seleziona ogni t-upla della tabella atleta.

ATLETA

	nome character var	cognome character var	nazionalita character var	ida [PK] integer
1	carlo	bianchi	italiana	1
2	giuseppe	verdi	francese	2
3	lisa	falck	americana	3
4	helen	cruck	tedesca	4
5	bianca	livingstone	inglese	6
*				

```
select nome,cognome,ida
from atleta
```

2) grazie all'ida di atleta, si selezionano dalla tabella specialità tutte le specialità a cui partecipa l'atleta;

per fare questo viene usata la tabella atlespe che, noto ida ( id della tabella atleta) permette di ricavare ids (id della tabella specialità)

ATLESPE

	idatleta integer	idspecialita integer
1	1	1
2	1	2
3	1	3
4	1	4
5	2	1
6	2	2
7	3	5
8	3	6
*		

```
select distinct specialita.sesso,specialita.tipospecialita, specialita.ids
from specialita, atleta, atlespe
where a.getIDa() =atlespe.idatleta and
atlespe.idspecialita = specialita.ids
```

SPECIALITA'

	sesso character(1)	tipospecialita character varying(30)	ids [PK] integer
1	M	slalom_speciale	1
2	M	slalom_gigante	2
3	M	supergigante	3
4	M	discesa_libera	4
5	F	slalom_speciale	5
6	F	slalom_gigante	6
7	F	supergigante	7
8	F	discesa_libera	8
*			

3) note le specialità fatte dall'atleta si ricava, tramite la tabella gaspe e la tabella gara , tutte le gare relative alle specialità a cui partecipa l'atleta.

### GASPE

	idspec integer	idgar integer
1	1	1
2	2	1
3	3	1
4	4	1
5	5	2
6	6	2
7	7	2
8	8	2

```
select gara.localita, gara.idg
from gaspe, gara
where gaspe.idspec= spec.getIds() and
gara.idg=gaspe.idgar
```

### GARA

	localita character var	dislivello integer	lunghezza integer	numeroporte integer	giorno integer	mese integer	anno integer	idg [PK] integer
1	abetone mas	34	45	5	23	4	2010	1
2	abetone_fem	345	67	45	23	12	2010	2
*								

4) ora si conosce un atleta (ida) una specialità (ids), una gara (idg), quindi il programma valuta se tale terna è presente nella tabella partecipazione; questo controllo è effettuato tramite la funzione controllo(.....) che restituisce true se tali valori non esistono.

5) a questo punto il programma chiede se l'atleta selezionato in 1) ha partecipato alla gara (idg) selezionata in 3)

Se sì, vengono inseriti gli attributi dell'associazione partecipazione assieme alla terna ida (in idatl), ids (in idsp), idg(in idgar)

### PARTECIPAZIONE

	punteggio integer	ordineparten. integer	ordinearrivo integer	idgar integer	idatl integer	idsp integer
1	23	2	1	1	1	4
2	34	3	2	1	2	2
3	45	3	5	2	3	6
4	12	3	5	2	4	5
5	56	4	7	2	3	5
6	34	5	4	2	6	5

## MODIFICA DATI

### *nuoveSpecialitaGara*

vediamo come si arriva alla funzione nuoveSpecialitaGara:

il costruttore della classe MenuModificaDati chiama il metodo **modificaDatiGara()**;

questo metodo:

1) seleziona ogni t-upla della tabella gara e carica i valori così ottenuti in un oggetto java di tipo Gara

GARA								
	localita character var	dislivello integer	lunghezza integer	numeroporte integer	giorno integer	mese integer	anno integer	idg [PK] integer
1	abetone_mas	34	45	5	23	4	2010	1
2	abetone_fem	345	67	45	23	12	2010	2
*								

```
select localita,dislivello,lunghezza,numeroporte,giorno,mese,anno,idg
from gara
```

2) per ogni gara,seleziona le specialità di tale gara utilizzando la tabella gaspe

GASPE		
	idspec integer	idgar integer
1	1	1
2	2	1
3	3	1
4	4	1
5	5	2
6	6	2
7	7	2
8	8	2

```
select specialita.sesso,specialita.tipospecialita,specialita.ids
from specialita, gara, gaspe
where gara.idg = a.getIdg() and gara.idg=gaspe.idgar and gaspe.idspec = specialita.ids
```

3) visualizza la gara considerata al passo 1) il sesso e le specialità ottenute al passo 2)

4) a questo punto il metodo **modificaDatiGara()** chiama il metodo **MenuModificaDatiGara()** che visualizza il menu di modifica dei dati della gara;

supponiamo di voler modificare una specialità della gara: viene chiamato il metodo

**nuoveSpecialitaGara()**

a cui sono passati gli oggetti gara ,idg e sesso.

Il codice sql sotto, seleziona (per il certo sesso dell'atleta) il tipo di specialità presenti nella tabella specialità non ancora praticate dall'atleta.

```
select specialita.tipospecialita,specialita.sesso,specialita.ids
from specialita
where specialita.sesso=sesso and specialita.ids not in
      (select specialita.ids
       from gara,gaspe, specialita
       where gara.idg= idg and gara.idg=gaspe.idgar and
       gaspe.idspec=specialita.ids)
```

NOTA **not in** sta ad indicare che l'id di specialità deve essere diverso dall'ids che ottengo dalle tabelle gara, gaspe ,specialità

#### GARA

	localita character varying	dislivello integer	lunghezza integer	numeroporte integer	giorno integer	mese integer	anno integer	idg [PK] integer
1	abetone mas	34	45	5	23	4	2010	1
2	abetone_fem	345	67	45	23	12	2010	2
*								

#### GASPE

	idspec integer	idgar integer
1	1	1
2	2	1
3	3	1
4	4	1
5	5	2
6	6	2
7	7	2
8	8	2

#### SPECIALITA'

	sex character(1)	tipospecialita' character varying(30)	ids [PK] integer
1	M	slalom_speciale	1
2	M	slalom_gigante	2
3	M	supergigante	3
4	M	discesa_libera	4
5	F	slalom_speciale	5
6	F	slalom_gigante	6
7	F	supergigante	7
8	F	discesa_libera	8
*			

da idg =1 ottengo 4 specialità che il **not in** evita di considerare

# CANCELLA DATI

## Cancella specialità

1) seleziona ogni t-upla della tabella specialità.

SPECIALITA'			
	sexo character(1)	tipospecialita character varying(30)	ids [PK] integer
1	M	slalom_speciale	1
2	M	slalom_gigante	2
3	M	supergigante	3
4	M	discesa_libera	4
5	F	slalom_speciale	5
6	F	slalom_gigante	6
7	F	supergigante	7
8	F	discesa_libera	8
*			

```
select sesso,tipospecialita, ids  
from specialita
```

2)seleziono l'ids della t-upla che voglio cancellare

NOTA: per poter cancellare una t-upla della tabella specialità è necessario:

- cancellare tutte le righe, presenti nelle tabelle atlespe e gaspe, riferirite da quella t-upla tramite i vincoli di integrità referenziali
- le t-uple della tabella partecipazione riferite a gare e atleti che partecipano a quella specialità.

3) parto con la tabella partecipazione e cancello tutte le t-uple in cui ids è uguale a quella che ho inserito

PARTECIPAZIONE						
	punteggio integer	idparten. integer	ordinearrivo integer	idgar integer	idatl integer	idsp integer
1	23	2	1	1	1	4
2	34	3	2	1	2	2
3	45	3	5	2	3	6
4	12	3	5	2	4	5
5	56	4	7	2	3	5
6	34	5	4	2	6	5

```
delete from partecipazione  
where partecipazione.idatl in  
    (select atlespe.idatleta  
     from atlespe  
     where idspecialita = Console.leggiIntero())
```

operatore IN  
(memo: IN vale o è uguale a)  
  
partecipazione.idatl è **uguale**  
**a** tutti i valori atlespe.idatleta  
ottenuti nel select tra le  
parentesi

In particolare nel codice sql precedente, sono cancellati dalla tabella partecipazione tutte le t-uple in cui idatl è uguale agli idatleta selezionati nella tabella atlespe quando idspecialita è uguale al valore inserito da programma (Console.leggiIntero())

ATLESPE		
	idatleta integer	idspecialita integer
1	1	1
2	1	2
3	1	3
4	1	4
5	2	1
6	2	2
7	3	5
8	3	6

4) cancello da atlespe le t-uple in cui idspecialità è uguale al valore inserito da programma

```
delete from atlespe
where idspecialita= Console.leggiIntero()
```

5) cancello da gaspe le t-uple in cui idspecialità è uguale al valore inserito da programma

```
delete from gaspe
where idspec= Console.leggiIntero()
```

6) a questo punto posso cancellare la t-upla della tabella specialità in cui idspecialità è uguale al valore inserito da programma



### *cancellaDatiAtleta*

1) leggi tutte le t-upla della tabella Atleta.

ATLETA				
	nome character var	cognome character var	nazionalita character var	ida [PK] integer
1	carlo	bianchi	italiana	1
2	giuseppe	verdi	francese	2
3	lisa	falck	americana	3
4	helen	cruck	tedesca	4
5	bianca	livingstone	inglese	6
*				

```
select nome,cognome,nazionalita, ida
from atleta
```

2)associa ad ogni atleta le sue specialità

```
select specialita.sesso,specialita.tipospecialita
from specialita, atleta, atlespe
where atleta.ida = a.getIDa() and atleta.ida=atlespe.idatleta and atlespe.idspecialita =
specialita.ids
```

3)L'utente seleziona l'ida dell'atleta che vuole cancellare

4)cancello tutte le t-uple della tabella partecipazione in cui idatl =ida

```
delete from partecipazione
where idatl= ida dell'utente
```

5)cancello tutte le t-uple della tabella atlespe in cui idatleta =ida

```
delete from atlespe
where idatl= ida dell'utente
```

6)tolti tutti i vincoli referenziali che il mio atleta ha nelle tabelle atlespe e partecipazione, cancello l'atleta dalla relativa tabella

```
delete from atleta
where idatl= ida dell'utente
```