

PROGETTO BASI DI DATI
GESTIONE DELLE GARE DI ATLETICA LEGGERA

Luca Allegro

Matricola: 1120573

Marco Giollo

Matricola: 1122414

1. ABSTRACT

Si vuole realizzare una base di dati per la gestione delle prestazioni delle gare di atletica leggera.

Viene preso come modello di riferimento il database della “Federazione Italiana di Atletica Leggera” nel quale è possibile visualizzare tutte le prestazioni ottenute da ogni atleta. È possibile, inoltre, consultare le classifiche delle migliori prestazioni per ogni categoria e specialità di appartenenza degli atleti. Nel seguente modello viene anche aggiunta la gestione delle manifestazioni e delle relative edizioni nelle quali vengono svolte le gare.

Le competizioni differiscono per specialità e per categoria ammessa (se specificata). Ogni gara, inoltre, si svolge in diverse fasi (es. batteria, semifinale, finale) ciascuna con la relativa classifica e prestazione (es. tempo o misura in base alla specialità di gara). Nella base di dati deve comparire anche lo stadio dove vengono disputate le gare di una manifestazione e da quale società tale luogo viene gestito.

2. ANALISI DEI REQUISITI

Il progetto in esame vuole modellare una base di dati che gestisca le prestazioni delle gare di atletica leggera. La base di dati gestirà tutti gli aspetti riguardanti gli atleti, le gare, le associazioni e le manifestazioni in cui vengono disputate le competizioni.

Ciascun **atleta** è identificato dalle seguenti informazioni:

- Nome e cognome
- Data di nascita
- Matricola (univoca nella base di dati) che lo identifica

Ogni atleta è allenato da un **allenatore** del quale si vuole sapere:

- Nome e cognome
- Data di nascita
- Matricola (univoca nella base di dati) che lo identifica
- Numero di atleti allenati

Ciascun atleta e allenatore è annualmente tesserato con una **società** con:

- Numero di atleti tesserati
- Email
- Provincia
- Regione
- Nome
- Telefono
- Codice che la identifica

La società ha sede e gestisce uno **stadio** caratterizzato da:

- Nome
- Indirizzo (Via, Numero civico e Città)
- Dalla specifica indoor oppure outdoor (se è coperto o è all'aperto)

In uno stadio si può svolgere una **edizione** di una **manifestazione** dove vengono svolte le gare. Per ogni manifestazione si vuole sapere:

- Nome della manifestazione
- Il codice della manifestazione, univoca nella base di dati
- La data di inizio e di fine
- Il numero dell'edizione

In ogni manifestazione vengono svolte delle **gare** che si riferiscono ad una determinata **categoria** (a cui fanno parte per età gli atleti) e ad una **specialità**.

Le **categorie** sono identificate attraverso un codice univoco (tale codice distingue anche le categorie maschili da quelle femminili, es. AM è riferito ad “allievi maschili” e AF è riferito ad “Allievi femminili”) e suddividono gli atleti in base all'età:

- Allievi (età fino a 17)
- Juniores (età dai 18 ai 19)
- Seniores (età dai 20 ai 35)
- Master (a partire dal 1 Gennaio del 35esimo anno di età)

Gli allievi appartenenti ad una categoria possono gareggiare solo nelle competizioni dedicate alla loro fascia di età.

Per le **specialità**, identificate anche esse da un codice univoco, si vuole registrare la tipologia e l'unità di misura relativa.

Ogni atleta, infine ottiene una **prestazione** con una posizione di arrivo in una (o ciascuna) fase di una gara.

3. PROGETTAZIONE CONCETTUALE

3.1 ENTITA' E ATTRIBUTI E GERARCHIE

- **Persona:** modella una persona generica registrata nel database

ATTRIBUTI:

- Nome: string
- Cognome: string
- DataNascita: date
- Matricola: int (identificatore)

Le persone si suddividono in due sotto entità che identificano gli atleti e gli allenatori. Per quest'ultimi in particolare si ha un attributo in più:

- NumeroAtleti: int

Che rappresenta il numero di atleti allenati.

Rappresenta, quindi, una gerarchia totale ma non sovrapposta.

- **Società:** modella una società per cui una persona registrata nel database può essere tesserata.

ATTRIBUTI:

- Codice: int (identificatore)
- NumeroAtleti: int
- Provincia: string
- Regione: string
- Sito: string
- Telefono: string

- **Stadio:** modella le informazioni relative allo stadio sede di una società.

ATTRIBUTI:

- Nome: string
- Via: string
- NumeroCivico: int
- Città: string

Gli attributi "Nome" e "Città" sono gli identificatori di tale entità (in una città non ci possono essere più stadi con lo stesso nome, ma può avvenire tra città diverse).

Uno stadio ha due sotto entità che identificano gli stadi indoor e quelli outdoor. Rappresenta, quindi una gerarchia totale ma non sovrapposta.

- **Manifestazione:** gestisce le informazioni generali relative ad una manifestazione

ATTRIBUTI:

- Codice: int (identificativo)
- Nome: string

- **Edizione Manifestazione:** identifica una specifica edizione della manifestazione, è quindi una entità che contiene le istanze dell'entità "manifestazione".

ATTRIBUTI:

- DataInizio: date
- DataFine: date
- Edizione: int
- CodiceManifestazione: int

L'identificatore è rappresentato dall'attributo "Edizione" e dal "codice" che identifica l'entità "Manifestazione". Sussiste, quindi un vincolo di integrità referenziale tra "CodiceManifestazione" e "Codice" dell'entità "Manifestazione"

- **Categoria:** rappresenta categorie in cui gli atleti vengono suddivisi.

ATTRIBUTI:

- Codice: int
- Nome: string

Non vengono specificati i sottoinsiemi elencati nell'analisi dei requisiti in quanto sono delle istanze di tale relazione.

- **Specialità:** Rappresenta in generale i vari tipi di specialità di gara

ATTRIBUTI:

- Codice: int
- Tipologia: string
- UnitàDiMisura: string

- **Gara:** Rappresenta le gare che vengono svolte nelle varie edizioni per determinate specialità e categorie.

ATTRIBUTI:

- Codice: int
- EdizioneManifestazione: int
- CodiceManifestazione: int
- SpecialitàDiRiferimento: int
- CategoriaDiRiferimento: int

Sussistono, in questo caso, anche dei vincoli di integrità referenziale:

- a) "EdizioneManifestazione" è vincolo di integrità referenziale con "Codice" di "Edizione Manifestazione".
- b) "NomeManifestazione" è vincolo di integrità referenziale con "CodiceManifestazione" di "Edizione Manifestazione".
- c) "SpecialitàDiRiferimento" è vincolo di integrità referenziale con "Codice" di "Specialità".
- d) "CategoriaDiRiferimento" è vincolo di integrità referenziale con "Codice" di "Categoria".

- **Prestazione:** Rappresenta le prestazioni che gli atleti possono sostenere nelle diverse fasi di una gara.

ATTRIBUTI:

- Posizione: int
- Atleta: int

E' presente in vincolo di integrità referenziale tra "Atleta" e "Matricola" dell'entità "Atleta".

La prestazione si specializza in due sotto entità:

- Corsa con attributi:
- Tempo: time
- Salto/Lancio, con attributi:
- Misura: double

3.2 DESCRIZIONE DELLE RELAZIONI E DELLE LORO PROPRIETA' STRUTTURALI

- **Persona – Società: Tesseramento**

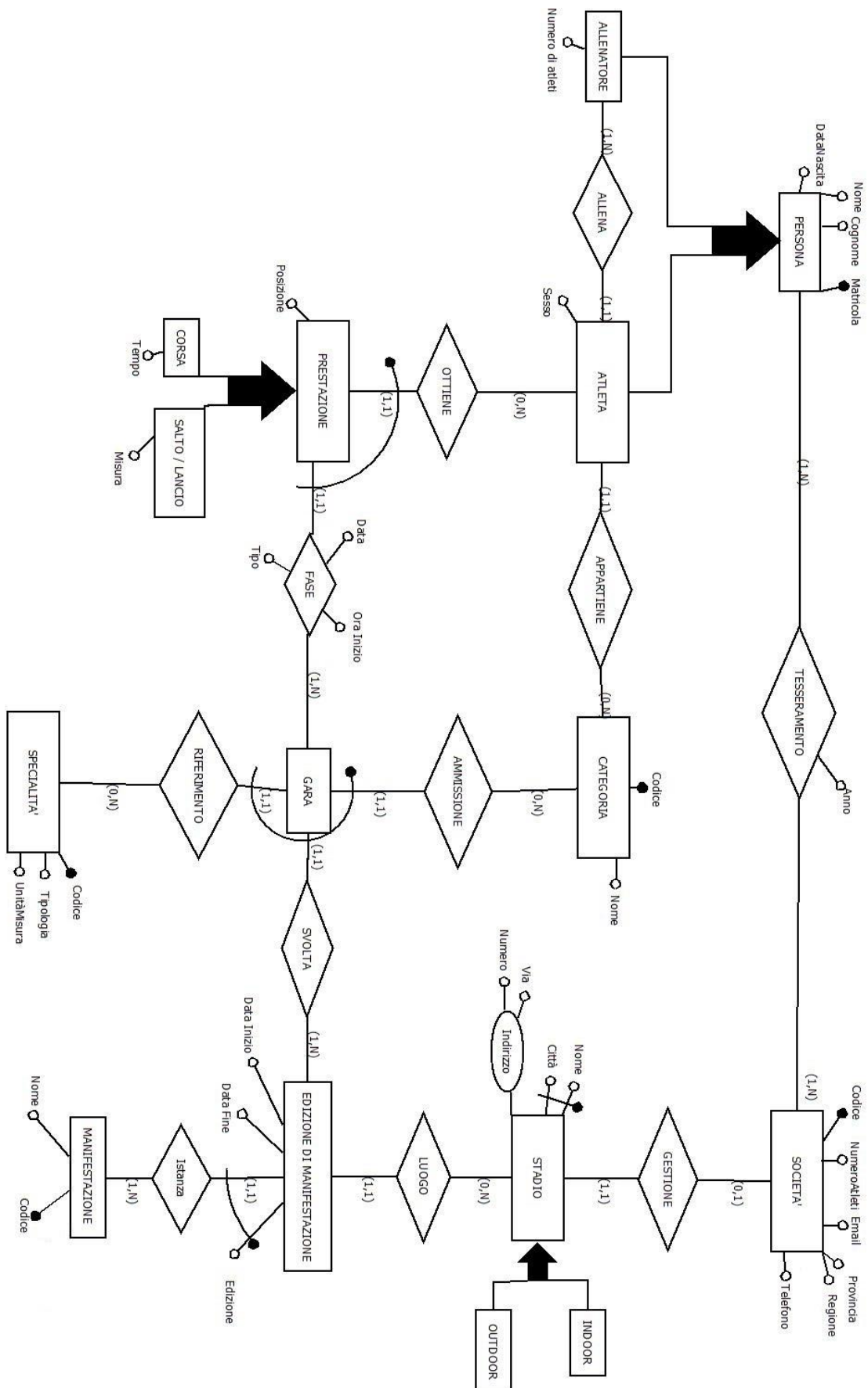
- Ogni persona è tesserata con una società annualmente.
- Una società ha uno o più persone tesserate.

- **Società – Stadio: Gestione**

- Alcune società gestiscono uno stadio.
- Uno stadio gestito da una sola società.

- **Allenatore – Atleta: Allena**
 - Un allenatore può allenare diversi atleti.
 - Un atleta viene allenato da un solo allenatore.
- **Atleta – Categoria: Appartenere**
 - Un atleta per gareggiare deve appartenere ad una categoria.
 - Una categoria può avere diversi atleti appartenente ad essa.
- **Manifestazione – EdizioneManifestazione: Istanza**
 - Una manifestazione ha almeno una edizione
 - Una edizione di una manifestazione deve essere istanza di una manifestazione
- **Gara- EdizioneManifestazione: Svolta**
 - Una gara si svolge in una edizione di una manifestazione
 - In una edizione della manifestazione vengono svolte diverse gare
- **Gara-Categoria: Ammissione**
 - Ad una gara è ammessa una sola categoria
 - Ogni categoria può avere diverse gare
- **Gara-Specialità: Riferimento**
 - Una gara è riferita ad una specialità
 - Una competizione di una determinata specialità può essere svolta in diverse gare
- **Gara-Prestazione: Fase**
 - Ogni gara è composta da diverse fasi in cui gli atleti possono ottenere una prestazione (le fasi hanno una data, ora di inizio e un tipo che si riferisce alle batterie, alla semifinale e alla finale).
 - Ogni prestazione può essere ottenuta da un atleta solo in una fase di gara.
- **Atleta-Prestazione: Ottiene**
 - Un atleta può ottenere una prestazione
 - Ogni prestazione è ottenuta da un solo atleta

3.3 DIAGRAMMA ER



4. PROGETTAZIONE LOGICA

4.1 RISTRUTTURAZIONE DELLO SCHEMA ER

Per realizzare la traduzione verso il modello logico (nel nostro caso si applica il modello logico-relazionale), è necessario ristrutturare il diagramma ER precedentemente riportato. Per far ciò si applicano i seguenti passi:

- Eliminazione delle generalizzazioni:
 - a) L'entità "persona" viene eliminata e accorpata completamente nelle entità figlie in quanto nella maggior parte delle interrogazioni gli accessi ai figli sono separati. E' stato necessario, quindi, sdoppiare la relazione "tesseramento" in "tesseramento allenatore" e "tesseramento atleta".
 - b) L'entità "prestazione" viene anche essa accorpata completamente nelle entità figlie perché vengono accedute separatamente da parte delle interrogazioni. Sono state, quindi, create due nuove entità che rappresentano le prestazioni del salto/lancio e le prestazioni della corsa con ognuna la relazione che aveva il padre.
 - c) Nell'entità "stadio", invece, le entità figlie vengono accorpate nell'entità padre e aggiunto un attributo di tipo carattere per specificare se è indoor (i) oppure outdoor (o).

- Eliminazione dell'attributo composto:

Viene eliminato l'attributo composto dell'entità "stadio" dividendolo nelle sue componenti nella stessa entità.

- Analisi delle ridondanze:

Nello schema sono presenti ridondanze:

- a) L'attributo "numeroAtleti" nell'entità "società"
- b) L'attributo "numeroAtleti" nell'entità "allenatore"

Si è deciso di togliere l'attributo da "allenatore" in quanto è maggiore il carico di lavoro per tenerlo aggiornato (dopo ogni aggiunta di un atleta) rispetto a calcolarlo ogni volta perché le interrogazioni non sono frequenti.

Invece, viene tenuto l'attributo di "società" perché viene utilizzato molto frequentemente e permette evitare l'onere di consultare ogni volta tutta la tabella atleti.

Nella tabella "Atleti" l'attributo categoria è ridondante perché sarebbe possibile calcolarla a partire dall'attributo "dataNascita" e "sesso" conoscendo l'anno corrente. E' stato scelto di mantenere tale ridondanza perché la maggior parte delle operazioni viene eseguita considerando la categoria dell'anno corrente e sarebbe troppo oneroso calcolarla ogni volta.

Si riporta l'analisi eseguita:

Tavola dei volumi:

Concetto	Tipo	Volume
Società	E	2000
Persona	E	220 000
Tesseramento	R	660 000

Tavola delle operazioni:

Operazione	Tipo	Frequenza
Inserimento nuova persona	I	2/giorno
Trovare il numero di atleti per ogni società	I	2/giorno
Rimuovere un atleta	I	2/giorno

Analisi:

Operazione 1:

Tabella accessi con ridondanza:

Concetto	Costr.	Acc.	Tipo
Persona	E	1	S
Tesseramento	R	1	S
Società	E	1	L
Società	E	1	S

Tabella accessi senza ridondanza:

Concetto	Costr.	Acc.	Tipo
Persona	E	1	S
Persona	R	1	S

Calcolando il costo con ridondanza:

1*(2) in lettura e 3*(4) accessi in scrittura per un totale di 18 accessi al giorno.

Calcolando il costo senza ridondanza:

2*(4) accessi in scrittura per un totale di 8 accessi al giorno.

Operazione 2:

Tabella accessi con ridondanza:

Concetto	Costr.	Acc.	Tipo
Società	E	1	L

Tabella accessi senza ridondanza:

Concetto	Costr.	Acc.	Tipo
Società	E	1	L
Tesseramento	R	1	L
Persona	E	1	L

Calcolando il costo con ridondanza:

1*(2) accessi in lettura per un totale di 2 accessi al giorno

Calcolando il costo senza ridondanza:

(1+100+100)*2 accessi in lettura per un totale di 402 accessi al giorno.

Operazione 3:

Tabella accessi con ridondanza:

Concetto	Costr.	Acc.	Tipo
Persona	E	1	S
Tesseramento	R	1	S
Società	E	1	L
Società	E	1	S

Tabella accessi senza ridondanza:

Concetto	Costr.	Acc.	Tipo
Persona	E	1	S
Persona	R	1	S

Calcolando il costo con ridondanza:

1*(2) in lettura e 3*(4) accessi in scrittura per un totale di 18 accessi al giorno.

Calcolando il costo senza ridondanza:

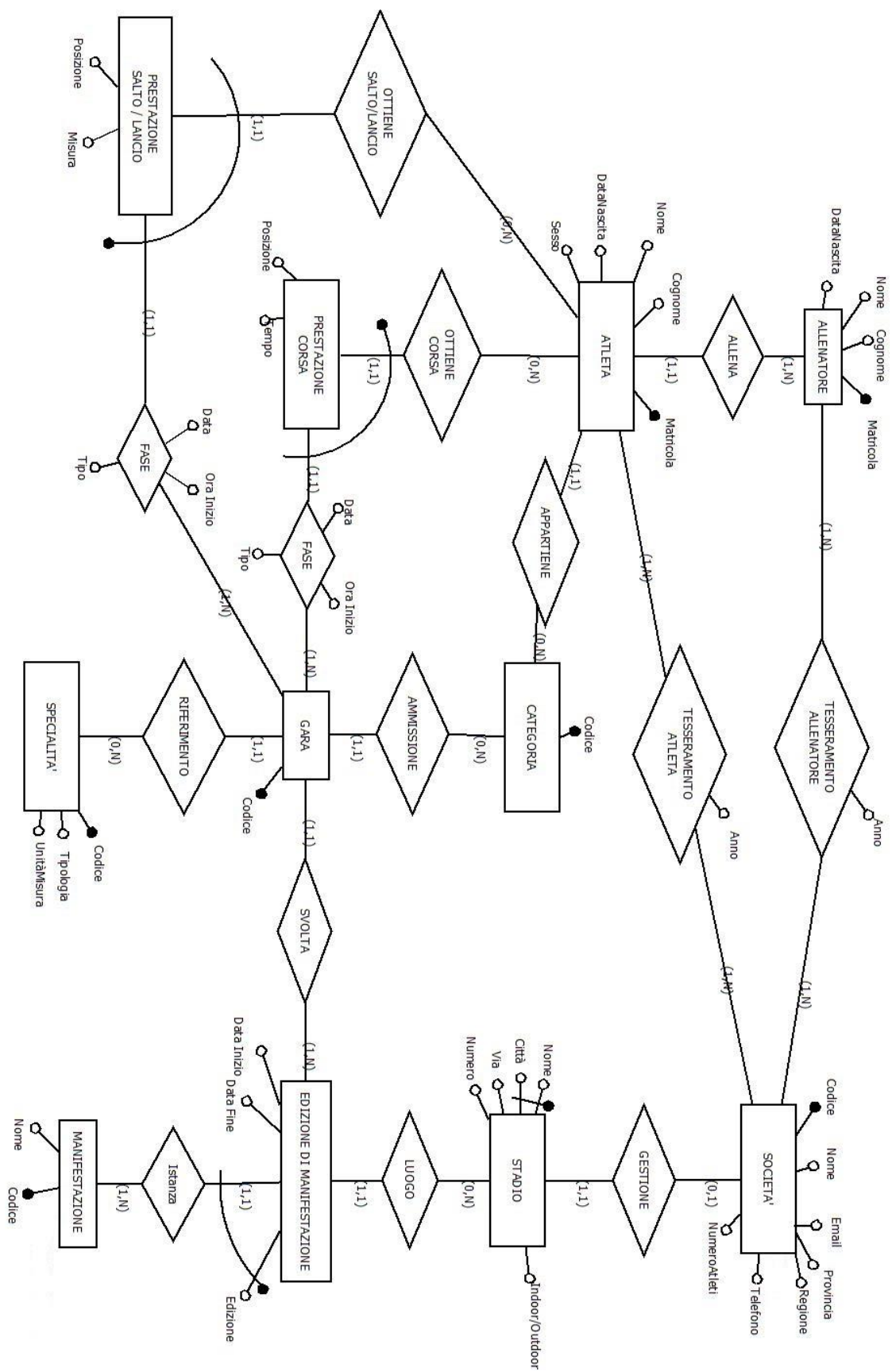
2*(4) accessi in scrittura per un totale di 8 accessi al giorno;

In conclusione mantenendo la ridondanza si hanno solo $18+18+2 = 38$ accessi al giorno in confronto ai $8+8+402 = 418$ accessi al giorno in caso di assenza di ridondanza. Si decide, quindi di mantenere la ridondanza.

Lo stesso identico principio lo si applica nel caso della ridondanza presente nell'entità "allenatore".

- Scelta degli identificatori principali:

Siccome per identificare una gara sarebbe necessario avere troppe chiavi esterne e quindi il volume delle tabelle "prestazione corsa" e "prestazione lancio" aumenterebbero senza motivo, è più conveniente inserire un codice identificativo univoco nella base di dati per ogni gara.



4.2 SCHEMA RELAZIONALE

SOCIETA (codice, numeroAtleti, email, provincia, regione, sito, telefono)
ALLENATORE (matricola, cognome, nome, dataNascita)
TESSERAMENTOALLENATORE (matricolaAllenatore, codiceSocieta, Anno)
ATLETA(matricola, cognome, nome, dataNascita, sesso, matricolaAllenatore, categoria)
TESSERAMENTOATLETA (matricolaAtleta, codiceSocieta, Anno)
STADIO (nome, via, numeroCivico, città, indoorOutdoor, societa)
MANIFESTAZIONE(codice, nome, descrizione)
EDIZIONEMANIFESTAZIONE(edizione, manifestazione, dataInizio, dataFine, stadio)
CATEGORIA (codice, nome)
SPECIALITA(codice, tipologia, unitaMisura)
GARA(id, edizione, manifestazione, categoria, specialita)
PRESTAZIONECORSA(idGara, atleta, tempo, posizione, tipoFase, data, orlnizio)
PRESTAZIONESL(idGara, atleta, misura, posizione, tipoFase, data, orlnizio)

5. QUERY, FUNZIONI, TRIGGER E PROCEDURE

5.1 FUNZIONI

- a) Si vuole creare una funzione che passati come parametri la data di nascita dell'atleta e un anno, ritorni la categoria associata all'atleta in tale anno.

```
DELIMITER $$
CREATE FUNCTION Fcategoria (mat INTEGER, anno YEAR)
RETURNS CHAR(2)
BEGIN
DECLARE annoNascita INTEGER;
DECLARE eta INTEGER;
DECLARE codiceCat CHAR(1);
DECLARE codiceSesso CHAR(1);
SELECT YEAR(dataNascita) INTO annoNascita FROM Atleta WHERE matricola=mat;
SELECT sesso INTO codiceSesso FROM Atleta WHERE matricola=mat;
SET eta = anno-annoNascita;
IF (eta<18) THEN SET codiceCat = 'A'; END IF;
IF (eta=18 OR eta=19) THEN SET codiceCat = 'J'; END IF;
IF (eta>19 AND eta<35) THEN SET codiceCat= 'S'; END IF;
IF (eta>34) THEN SET codiceCat='M'; END IF;
CONCAT(codiceCat,codiceSesso);
END$$
DELIMITER ;
```

- b) Si vuole creare una funzione che passati come parametri l'anno di nascita e il sesso di un atleta ritorni la categoria dell'atleta in tale anno.

```
DROP FUNCTION IF EXISTS FcategoriaS;
DELIMITER $$
CREATE FUNCTION FcategoriaS (sesso char(1), dataN DATE, anno YEAR)
RETURNS CHAR(2)
BEGIN
DECLARE eta INTEGER;
DECLARE codiceCat CHAR(1);
SET eta = anno-YEAR(dataN);
IF (eta<18) THEN SET codiceCat = 'A'; END IF;
IF (eta=18 OR eta=19) THEN SET codiceCat = 'J'; END IF;
```

```

IF (eta>19 AND eta<35) THEN SET codiceCat= 'S'; END IF;
IF (eta>34) THEN SET codiceCat='M'; END IF;
RETURN CONCAT(codiceCat, sesso);
END$$
DELIMITER ;

```

Viene utilizzata la seconda funzione perché è più efficiente in quanto nella prima al suo interno vengono eseguite due interrogazioni sulla tabella “atleta”, mentre nella seconda non viene fatta alcuna query.

5.2 TIGGER

- a) Si vuole creare un trigger che prima di inserire un atleta calcoli, attraverso l'apposita funzione, la categoria associata e la inserisca correttamente nella tabella.

```

DROP TRIGGER IF EXISTS TcatAtleta;
DELIMITER $$
CREATE TRIGGER TcatAtleta
BEFORE INSERT ON Atleta
FOR EACH ROW
BEGIN
SET new.categoria=FcategoriaS(new.sesso,new.dataNascita,YEAR(CURDATE()));
END$$
DELIMITER ;

```

- b) Si vuole creare un trigger che prima di inserire una prestazione per l'atleta in una determinata gara verifichi se l'atleta abbia potuto partecipare a tale gara. In caso negativo ritorni un messaggio di errore.

```

DROP TRIGGER IF EXISTS controllaCategoria;
DELIMITER $$
CREATE TRIGGER controllaCategoria
BEFORE INSERT ON prestazioneCorsa
FOR EACH ROW
BEGIN
DECLARE catAtleta CHAR(2);
DECLARE catGara CHAR(2);
DECLARE strErrore VARCHAR(80);
SET catAtleta=Fcategoria(new.atleta, YEAR(new.data));
SELECT categoria INTO catGara FROM Gara WHERE id=new.idGara;
IF(catAtleta <> catGara)
THEN
SET strErrore = CONCAT('L atleta con matricola ', new.atleta, ' non ha partecipato alla gara con codice ', new.idGara);
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = strErrore;
END IF;
END$$
DELIMITER ;

```

5.3 QUERY

- a) Le 10 migliori prestazioni della categoria allievi nel 2015 nei 1000 metri. Si vuole sapere, inoltre il nome, cognome, l'anno di nascita, data della gara e la società con cui sono tesserati gli atleti nell'anno in cui hanno ottenuto la prestazione ordinandoli per miglior tempo.

```

SELECT a.cognome,a.nome, YEAR(a.dataNascita) AS annoDiNascita, soc.nome AS societa,
pc.tempo, pc.data

```

```

FROM Atleta a JOIN prestazioneCorsa pc ON (a.matricola=pc.atleta)
JOIN Gara g ON (pc.idGara=g.id)
JOIN Specialita s ON (g.specialita=s.codice)
JOIN tesseramentoAtleta ta ON (a.matricola=ta.matricolaAtleta AND ta.anno='2015')
JOIN Societa soc ON (ta.codiceSocieta=soc.codice),
Categoria c
WHERE c.nome='Allievo Maschile'
AND g.categoria=c.codice
AND s.nome='1000 metri'
AND YEAR(pc.data)='2015'
ORDER BY pc.tempo ASC
LIMIT 10;

```

cognome	nome	annoDiNascita	societa	tempo	data
Giordani	Luca	2000	POLISPORTIVA AURORA 76 ASD	00:02:35.20	2015-06-01
Todescato	Matteo	1998	ATLETICA VICENTINA	00:02:35.40	2015-09-06
Smith	Marco	1999	POLISPORTIVA AURORA 76 ASD	00:02:35.89	2015-08-18
Griggio	Tommaso	2000	C.U.S. PADOVA	00:02:36.01	2015-08-18
Griggio	Tommaso	2000	C.U.S. PADOVA	00:02:36.02	2015-06-01
Todescato	Matteo	1998	ATLETICA VICENTINA	00:02:36.51	2015-06-01
Rackik	Yassin	1999	G.S.A. VENEZIA	00:02:37.67	2015-10-22
Smith	Marco	1999	POLISPORTIVA AURORA 76 ASD	00:02:37.68	2015-10-22
Giordani	Luca	2000	POLISPORTIVA AURORA 76 ASD	00:02:37.90	2015-09-06
Griggio	Tommaso	2000	C.U.S. PADOVA	00:02:38.10	2015-10-22

- b) La migliore prestazione dell'atleta nelle gare di corsa per le specialità in cui ha gareggiato per ogni anno.

```

DROP VIEW IF EXISTS migliorePrestazioneCorsa;
CREATE VIEW migliorePrestazioneCorsa
AS
SELECT p1.idGara AS idGara, p1.atleta AS atleta, p1.tempo AS tempo, p1.posizione AS
posizione, p1.tipoFase AS tipoFase, p1.data AS data, p1.oraInizio AS oraInizio
FROM prestazioneCorsa p1
JOIN Gara g1 ON (p1.idGara=g1.id)
WHERE p1.tempo IN (SELECT MIN(p2.tempo)
FROM prestazioneCorsa p2 JOIN Gara g2 ON (p2.idGara=g2.id)
WHERE p1.atleta=p2.atleta AND YEAR(p1.data)=YEAR(p2.data)
AND g1.specialita=g2.specialita);

```

```

SELECT *
FROM migliorePrestazioneCorsa;

```

idGara	atleta	tempo	posizione	tipoFase	data	oraInizio
1	2	00:02:35.40	1	F	2015-09-06	11:40:00
1	11	00:02:38.50	3	F	2015-09-06	11:40:00
2	6	00:02:37.67	1	F	2015-10-22	16:40:00
2	7	00:02:39.20	4	F	2015-10-22	16:40:00
3	15	00:02:35.20	1	F	2015-06-01	20:00:00
4	3	00:02:35.89	1	F	2015-08-18	18:00:00
4	9	00:02:36.01	2	F	2015-08-18	18:00:00
5	6	00:02:35.30	2	F	2016-06-25	15:30:00
5	10	00:02:35.45	3	F	2016-06-25	15:30:00
5	15	00:02:34.99	1	F	2016-06-25	15:30:00
6	12	00:02:33.21	2	F	2015-06-01	20:45:00
6	13	00:02:32.67	1	F	2015-06-01	20:45:00
7	7	00:00:11.40	2	F	2015-10-21	21:00:00
7	11	00:00:11.02	1	F	2015-10-21	21:00:00
8	4	00:02:39.23	2	F	2015-08-18	19:00:00
8	5	00:02:36.00	1	F	2015-08-18	19:00:00
8	8	00:02:39.45	3	F	2015-08-18	19:00:00

- c) Nome, cognome, società e misura ottenuta dai 5 atleti distinti appartenenti alla categoria allievi maschili che nel 2015 hanno ottenuto le migliori prestazioni in stadi outdoor nei 1000 metri ordinati per tempo.

```

SELECT a.nome, a.cognome, soc.nome AS societa, pc.tempo

```

```

FROM Atleta a JOIN migliorePrestazioneCorsa pc ON (a.matricola=pc.atleta)
JOIN Gara g ON (pc.idGara=g.id)
JOIN Specialita sp ON (g.specialita=sp.codice)
JOIN tesseramentoAtleta ta ON (a.matricola=ta.matricolaAtleta AND ta.anno=2015)
JOIN Societa soc ON (ta.codiceSocieta=soc.codice)
JOIN edizioneManifestazione em ON (g.edizione=em.edizione AND g.manifestazione=em.manifestazione)
JOIN Stadio st ON (em.nomeStadio=st.nome AND em.cittaStadio=st.citta), Categoria c
WHERE c.nome='Allievo Maschile'
AND g.categoria=c.codice
AND sp.nome='1000 metri'
AND st.indoorOutdoor='O'
AND YEAR(pc.data)='2015'
ORDER BY pc.tempo ASC
LIMIT 5;

```

nome	cognome	societa	tempo
Matteo	Todescato	ATLETICA VICENTINA	00:02:35.40
Marco	Smith	POLISPORTIVA AURORA 76 ASD	00:02:35.89
Tommaso	Griggio	C.U.S. PADOVA	00:02:36.01
Yassin	Rackik	G.S.A. VENEZIA	00:02:37.67
Roberto	Sirio	C.U.S. PADOVA	00:02:38.50

- d) Gli allenatori che attualmente allenano atleti che hanno vinto una gara(tipoFase='f'). Per tale allenatore si vuole sapere nome, cognome, numero di atleti vincenti e società per cui sono attualmente tesserati.

```

SELECT al.Cognome, al.Nome,s.Nome AS Societa, COUNT(atl.matricola) AS atletiVincenti
FROM Allenatore al JOIN tesseramentoAllenatore ta ON
(al.matricola=ta.matricolaAllenatore AND ta.anno=YEAR(CURDATE()))
JOIN Societa s ON (ta.codiceSocieta=s.codice)
JOIN Atleta atl ON (al.matricola=atl.matricolaAllenatore)
WHERE EXISTS( ( SELECT atl.matricola
FROM prestazioneCorsa pc
WHERE atl.matricola=pc.atleta AND pc.posizione=1 AND pc.tipoFase='F')
UNION
(SELECT atl.matricola
FROM prestazioneSL psl
WHERE atl.matricola=psl.atleta
AND psl.posizione=1 AND psl.tipoFase='F' ))
GROUP BY (al.matricola)
ORDER BY COUNT(atl.matricola) DESC;

```

Cognome	Nome	Societa	atletiVincenti
Tomasi	Diana	ATLETICA VICENTINA	2
Cestonaro	Giuseppe	ATLETICA VICENTINA	2
Matteazzi	Matteo	G.S.A. VENEZIA	1
Zanon	Silvia	POLISPORTIVA AURORA 76 ASD	1
Grosso	Marco	ATLETICA VIS ABANO	1

- e) Nome, cognome degli atleti, nome della manifestazione, data della gara e posizione di arrivo degli sportivi che ha raggiunto il podio in una gara svoltasi nello stadio gestito dalla società per cui era stato tesserato nell'anno in cui si è svolta la gara.

```

DROP VIEW IF EXISTS risultati;
CREATE VIEW risultati

```

```

AS (SELECT p1.idGara AS idGara, p1.atleta AS atleta, p1.posizione AS posizione, p1.tipoFase AS tipo-
Fase, p1.data AS data
FROM prestazioneCorsa p1)
UNION
(SELECT p1.idGara AS idGara, p1.atleta AS atleta, p1.posizione AS posizione,
p1.tipoFase AS tipoFase, p1.data AS data
FROM prestazioneSL p1);

SELECT atl.cognome, atl.nome, r.posizione, r.data, s.nome AS societa, st.nome AS
stadio, m.nome AS manifestazione
FROM Atleta atl JOIN risultati r ON (atl.matricola=r.atleta)
JOIN Gara g ON (r.idGara=g.id)
JOIN edizioneManifestazione em ON (g.edizione=em.edizione AND g.manifestazione=em.manifesta-
zione)
JOIN Manifestazione m ON (em.manifestazione=m.codice)
JOIN Stadio st ON (em.nomeStadio=st.nome AND em.cittaStadio=st.citta)
JOIN tesseramentoAtleta ta ON (atl.matricola=ta.matricolaAtleta AND ta.anno=YEAR(r.data))
JOIN Societa s ON (ta.codiceSocieta=s.codice)
WHERE st.societa=s.codice AND r.tipoFase='F' AND r.posizione<=3;

```

cognome	nome	posizione	data	societa	stadio	manifestazione
Smith	Marco	2	2015-10-22	POLISPORTIVA AURORA 76 ASD	Dal Maso	Memorial Mennea
Lora	Filippo	3	2016-08-22	S.A.F. BOLZANO	Brichi	Campionati Italiani

- f) Le provincie delle società per cui erano tesserati gli atleti che hanno ottenuto le peggiori prestazioni di corsa differenziate per specialità e categoria, indicandone il numero.

```

SELECT s.provincia, COUNT(*) AS peggioriPrestazioni
FROM Gara g JOIN prestazioneCorsa pc ON (g.id=pc.idGara)
JOIN Atleta atl ON (pc.atleta=atl.matricola)
JOIN tesseramentoAtleta ta ON (atl.matricola=ta.matricolaAtleta AND
YEAR(pc.data)=ta.anno)
JOIN Societa s ON (ta.codiceSocieta=s.codice)
WHERE pc.tempo IN (SELECT MAX(tempo) FROM prestazioneCorsa pc2 JOIN Gara g2 ON
(pc2.idGara=g2.id) WHERE g2.categoria=g.categoria AND g2.specialita=g.specialita)
GROUP BY s.provincia
ORDER BY peggioriPrestazioni DESC;

```

provincia	peggioriPrestazioni
Bolzano	2
Venezia	1
Padova	1